

**DEVELOPMENT OF A PERSONALIZED WEB-BASED
DATA MANAGEMENT SYSTEM FOR PHYSIOLOGICAL
AND METABOLOMIC DATA INTEGRATION IN
SYSTEMS BIOLOGY RESEARCH**

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA

BY

Priyaranjan Tokachichu

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Stephen T. Parente, Ph.D., Advisor;
Terrence J. Adam, Ph.D., Co-Advisor

November, 2010

Acknowledgements

First and foremost I offer my sincere gratitude to my supervisor and mentor Dr. Gregory Beilman without whom this thesis never would have been possible. I am deeply grateful for the tremendous confidence he showed in me and the support he gave me in completing this thesis. I am thankful to Stephen Parente, Terrence Adam, Stuart Speedie, William Thomas and David Radosevich for their advice and support during this project.

I would like to acknowledge and appreciate the editing efforts of Daniel Lexcen and his support in understanding metabolomics research. I appreciate the help and advice of surgical critical care laboratory members Jeffrey Chipman, Kristine Mulier, Elizabeth Lusczek, Nancy Witowski, Joseph Greenberg and Eileen Eggenberger in the development of this project. I am thankful to Minnesota Supercomputing Institute and their support personnel for hosting and supporting the project.

I am grateful to my wife Eileen Schneider and my daughter Nyneishia Schneider for their love that guided me through difficult times during the project development. I am grateful to my mom Rangamani Tokachichu and my dad Anjaneya Raju Tokachichu for the sacrifices they made for my education. I am thankful to my brother Gourinath Tokachichu for his support.

I would like to thank The Office of Naval Research for supporting this project; Grants N00014-09-1-0323 and N00014-05-1-0344.

I would like to extend my sincere gratitude to all the people who either directly or indirectly supported this project but not listed here.

Dedication

This thesis is dedicated to my mentor Gregory J. Beilman, my wife Eileen M. B. Schneider and my daughter Nyneishia P. M. K. Schneider without whom I never could have made it this far.

Abstract

The Surgical Critical Care laboratory at the University of Minnesota conducts pre-clinical research that involves integration of physiological and metabolomic data. Managing the data using flat file data management systems has become increasingly difficult as the research progressed. Diversity of research groups and differences in data generation timeline makes data sharing between research groups difficult. This project explains the development of a data management system that can fulfill the data needs of the Surgical Critical Care laboratory at the University of Minnesota, and other research with similar data needs.

Each systems biology research project has unique data needs and requires a personalized data management system. A web-based data management system would be independent of any computer operating system and makes data access and sharing easier. A relational database management system manages enormous amounts of data more effectively than a flat file data management system. These are some of the factors that influenced the development of a personalized web-based data management system with a back end relational database management system. Multi-disciplinary teams' involvement, short-term goals, need for continuous changes/development, pre-clinical research, etc., are some of the factors that have driven the hybridization of open-source/commercial software products, and the usage of a health informatics model in combination with an agile software development model, in developing this software product named c-Surge.

The c-Surge software was tested for user satisfaction using a survey, and to find any differences in the real-time data collected by the software and the manually collected

data. Out of 12 members who used the software or participated in its development, 10 members have participated in the survey. Most of the users were satisfied/extremely satisfied with several functions of the software, or the overall software. A paired t-test was conducted to compare the real-time and manual data of each subject, and there was no statistically significant difference between the data types. The results show that home grown applications like c-Surge data management system are useful in fulfilling the data needs of a systems biology research project.

The c-Surge data management system's development process shows easier ways to develop a home grown application that can fulfill unique data needs of a systems biology research project within a reasonable economic expense.

Table of Contents

Acknowledgements	i
Dedication	ii
Abstract.....	iii
Table of Contents	v
List of Tables	viii
List of Figures.....	ix
Introduction.....	1
Statement of the problem	1
Background and Need	3
Rationale.....	4
Purpose of the Project.....	5
Hypothesis	5
Methods	6
Limitations.....	7
Review of Literature.....	9
Materials and Methods.....	16
Review of Problems in Data Management.....	18
Standalone vs. Web-based Application Development	25
Technological choices and hosting services.....	26

Database Development.....	29
Website Development	32
Website Menu.....	34
Data Extraction.....	44
Methods Followed to Improve Data Analysis Procedure	55
Development and Production Machine Maintenance	57
Automated Data Backup.....	57
Managing Web Development.....	58
Final Product Functionality Validation	62
Results	65
Survey Questions and Responses	65
Results of comparison of manual and real-time data	81
Discussion.....	82
Conclusions.....	84
References.....	85
Appendices.....	91
Appendix A. Website code.....	91
Appendix B. Database tables' structure	574
Appendix C. Manual and Real-time data comparison	592

Appendix D. Automated web programming file modification on development machine and synchronization with production machine	621
Appendix E. Real-time data upload	652
Appendix F. Real-time data extraction.....	658

List of Tables

Table 1 Vitals and Blood Gas variables collected during experiment.....	18
Table 2 Laboratory variables collected during experiment.	20
Table 3 Real-time variables collected.....	21
Table 4 List of software used.....	29

List of Figures

Figure 1 Extreme Programming.....	17
Figure 2 S-file with real-time data	23
Figure 3 H-file with real-time data	23
Figure 4 G-file with real-time data	24
Figure 5 Data flow with Standalone and Web-based technology usage.....	26
Figure 6 Database Structure.....	31
Figure 7 Directory tree of the website on the production server	33
Figure 8 Web-forms example – Vitals.....	35
Figure 9 Simple Export webpage.....	37
Figure 10 Advanced Export variable selection webpage.....	38
Figure 11 Advanced Export query results pop-up	39
Figure 12 Advanced Export query save box.....	40
Figure 13 My Queries	41
Figure 14 Shared Queries.....	42
Figure 15 File Upload	42
Figure 16 User Account Administration.....	43
Figure 17 TEG raw data file	45
Figure 18 TEG processed data file.....	45
Figure 19 Binning raw data file	46
Figure 20 HNMR raw data file	47
Figure 21 PNMR raw data file.....	48
Figure 22 Real-time data files.....	50

Figure 23 Real-time raw data file	50
Figure 24 Real-time data with removed blank lines and unwanted data lines.....	51
Figure 25 Real-time data after removing irrelevant characters	52
Figure 26 Real-time data fully processed	54
Figure 27 Real-time data final directories	54
Figure 28 File backup, automated code processing and upload	61
Figure 29 Survey embedded in cSurge	64
Figure 30 Question 1 responses	65
Figure 31 Question 2 responses	66
Figure 32 Question 3 responses	67
Figure 33 Question 4 responses	68
Figure 34 Question 5 responses	69
Figure 35 Question 6 responses	70
Figure 36 Question 7 responses	71
Figure 37 Question 8 responses	72
Figure 38 Question 9 responses	73
Figure 39 Question 10 responses	74
Figure 40 Question 11 responses	75
Figure 41 Question 12 responses	76
Figure 42 Question 13 responses	77
Figure 43 Question 14 responses	78
Figure 44 Question 15 responses	79
Figure 45 Question 16 responses	80

Figure 46 Question 17 responses	81
Figure 47 htdocs folder structure	91
Figure 48 Files in images directory	92
Figure 49 List of files in js directory	93
Figure 50 List of files in stylesheets directory	241
Figure 51 List of files in SCCL directory in cgi-bin.....	292
Figure 52 List of files in cgi-bin directory on development machine.....	621
Figure 53 List of files used for silent real-time data upload	652

Introduction

Statement of the problem

Systems biology research often faces data management problems and need a uniquely built software solution that will fulfill data management needs. Balancing the budgetary issues, time spent in development, and long lasting product support are a few of the challenges faced by developers of the system.

The Surgical Critical Care (SCC) laboratory, University of Minnesota, conducts metabolomic experiments under the guidance of Dr. Beilman. To understand the need for a data management tool to manage this research, it is necessary to have a basic understanding about systems biology and metabolomic research. Lisacek & Appel, 2007 quoted in their editorial that “There is currently no consensus for the definition of Systems Biology” and it appears to be right even in 2010 due to a variety of definitions floating around in various journals, articles and websites. But, as per the following definitions, it is clear on what systems biology is intended to do. According to Kitano, Computational systems biology, 2002 “To understand biology at the system level, we must examine the structure and dynamics of cellular and organismal function, rather than the characteristics of isolated parts of a cell or organism”. Molecular systems biology journal of the Nature Publishing Groups, 2010 defines molecular systems biology as “...an integrative discipline that seeks to explain the properties and behavior of complex biological systems in terms of their molecular components and their interactions”.

Several “-omics” research including the newly emerging¹ metabolomics research are considered as systems biology research (Loewe, 2009). Metabolome is defined by Fiehn O. , 2002 as “...the set of metabolites synthesized by an organism...” based on Oliver, et al., 1998 publication about “Systematic Functional Analysis of Yeast Genome”. Oliver Fiehn also explained different analytical approaches for metabolome and metabolic data analyses which are target analysis, metabolite profiling, metabolomics and metabolic fingerprinting. A huge amount of metabolome data is generated in a typical metabolomic experiment (Goodacre, 2005). Goodacre in the same publication also briefly mentioned about formation of several global metabolomics-based research initiatives and The Metabolomics Society (metabolomicsociety.org) whose mission is to “...promote the growth and development of metabolomics internationally”. Information about metabolomic data standards², metabolomic databases³, software available⁴ for various metabolomic analyses are some of the services offered by Metabolomics Society via their website⁵.

All of the literature here points towards the recent advancements in systems biology especially metabolomics and huge amounts of data generated in metabolomics research. The same is true about the data generated from metabolomics research in the SCC laboratory at University of Minnesota. Simple flat file data management system will not be able to handle these enormous data.

¹ <http://www.metabolomicsociety.org/>

² <http://www.metabolomicsociety.org/standardization.html>

³ <http://www.metabolomicsociety.org/database.html>

⁴ <http://www.metabolomicsociety.org/software.html>

⁵ <http://www.metabolomicsociety.org/>

Background and Need

A pre-clinical research protocol (termed MONR for convenience) at the Surgical Critical Care laboratory aims "...to identify the changes occurring in the metabolome during the course of traumatic shock and resuscitation with the objective to identify biomarkers in clinically relevant biofluids (urine, plasma) that are most associated with risk of multiple organ failure and death" (Beilman, 2008). This protocol is a modification of a previously well-established protocol of hemorrhagic shock/resuscitation and recovery period developed in the same laboratory. Materials and methods of the MONR protocol closely relate to that of the hemorrhagic shock protocol that was published in the journal of Trauma (George, et al., 2010). The difference between these protocols is that the recent protocol involves a simulated pulmonary contusion before induced hemorrhagic shock, and an induced liver crush injury before limited resuscitation. The MONR study also proposes that an organism's metabolome is an indicator of biological system's response to external stimuli. This data analysis proposed closely reflects the metabolite profiling data analysis method described by Fiehn, 2002.

The MONR study generates huge amounts of physiological, biochemical and metabolomic data. Following are the general problems faced by researchers and laboratory personnel of the Surgical Critical Care laboratory in data management of the MONR study.

1. Enormity of data
2. Amount of time spent in data collection and data entry
3. Need for streamlining data collection, integration, display, exchange and export
4. Real-time data management

5. Lag between data collection and analyses

Rationale

The above situation closely reflects many pre-clinical research environments in terms of data management needs (Mayer, 2009). They need a better database management system, a better user interface and a well distributed application. The three primary approaches considered to address this issue are

- Developing home-grown proprietary software
- Using open-source software available
- Using commercial proprietary software (Mayer, 2009)

According to Mayer, 2009 developing home-grown software is the most expensive of all of the choices available; The inexpensive solution is open-source software, but a perfect match to the data needs is hard to find and difficult to use; Commercial software come with standards and good service but lack the flexibility to suite the changing needs of innovative research. Many of the institutions opting for home grown applications have to accumulate sufficient information technology to support the home grown application but probably will not be used for any other purpose. In such a case all the maintenance costs associated with the infrastructure are just associated with one home grown application which will be expensive on expense per product basis. In case of small research groups, there is a lack of information systems and support personnel (Leon-Chisen, 2004); Universities might have better information systems intended to serve a group of research projects but the information technology personnel might not be supporting individual projects. Each choice alone could not satisfy the data management needs of metabolomic

research, so the reasonable choice would be to build a hybrid utilizing open-source and commercial software products.

Purpose of the Project

Systems biology research needs a data management system that will address issues like

- Enormity of data
- Time spent on data collection
- Sharing data, real-time data management
- Data integration
- Lag between data collection and analysis

Examples of these kinds of systems include SBEAMS (Marzolf, et al., 2006), AMEN (Chalmel & Primig, 2008), data management products of pharmaceutical companies (Ficenec, et al., 2003) etc. (Mayer, 2009). These systems concentrate on collecting and integrating data pertaining to specific research area but not including physiological data. The purpose of this project is to build a data management system that not only addresses the issues mentioned above but also will integrate physiological data with other research data to allow exploratory data analyses.

Hypothesis

Systems biology projects need personalized home-grown applications due to unique data management and functionality needs of specific research. A hybridization of open-source and commercial software products will reduce the expenses involved in building a home-grown application from scratch and will still be equally capable of addressing data management needs as other home-grown or commercial applications.

Methods

A combination of the Extreme programming (XP) which is a type agile software development model (Lindstrom & Jeffries, 2004) and health informatics model with three basic components data, information and knowledge (DIK) is used in developing the cSurge software. A thorough review of data management needs at SCC laboratory, technological choices, and development plan are the core building blocks for this software.

The project will first be developed on a development machine, tested and then moved to a production machine. After initial prototype has been launched, updates will be performed regularly and will be pushed to the production machine from development machine. The update process will be automated to avoid any human generated mistakes. Development machine will remain on site at Surgical Critical Care laboratory and production machine is hosted by Minnesota Supercomputing Institute (MSI), University of Minnesota. The application is deployed via internet (web-based). Open source software programs and open-source computer languages are used for web programming and designing. Real-time data collection and data analysis are performed by commercial software products. File hosting, database hosting and data analysis software services are provided by MSI. All of the MSI services are available for free of charge for research as per University of Minnesota's policy.

Personal and group interviews with researchers and lab personnel will be performed to assess the data flow. Personal visits to operating room, NMR facility, MSI facility will be performed to understand the systems. Meetings with MSI support personnel will be

scheduled to discuss the possibility of changes in their systems to accommodate the project needs.

Research into various open-source software products and open-source computer languages will be performed in order to select them for the project. Alternatives to commercial software packages will be reviewed in order to replace them with open-source products in future.

Comparison manual and real-time data collected will be performed in order to assess the importance of real-time data collection. Time motion assessment of data management before and after the implementation of the project will be performed. New data integration methods will be created and the prototype will be tested for future use. Modifications to the system will be performed based on the conclusions derived from weekly laboratory meetings over the period of development.

At the end of the project user satisfaction survey will be conducted to assess the quality of the project and user satisfaction. Real-time and manual data is compared statistically to assess the accuracy of the system in collecting data.

Limitations

Metabolomics, as a part of systems biology research, will generate enormous amounts of data with diverse data sources and file formats that are unique for a research project or a research group. Any data management system that is built to address the needs of this kind of situation might not be cross compatible with other research groups. Out of the software choices made for hybridization some are due to superior qualities, better support, and some are just due to the ease of availability, personal proficiency in the

software. A multitude of combination of these software products is possible for the same project. This will cause serious problems for continued support of this data management system unless it is well documented with quality tutorials. New programmers/system administrators need the same or better proficiency of programming languages and software used in the system, for its continuity and usability.

Review of Literature

Metabolomics which is considered as a part of systems biology research is a new field of research compared to proteomics and genomics. Human Metabolome Database (HMDB) was first published in 2007 (Wishart, et al., 2007; Wishart, Human Metabolome Database: completing the `human parts list, 2007) and is very recent compared to the Protein Data Bank which has over 36 years of history (Berman, 2008). Rapid development of metabolomics field is attributable to technological advancements (Fernie, et al., 2004).

Data integration and management are the major challenges faced by systems biology research projects. There are several working examples of data management and integration in proteomics (Shah, et al., 2007). Metabolomics field faces similar data integration and management challenges. Goodman, et al., 1998 mentions complexity of datasets generated in large scale biology projects demand a flexible database management system that can meet changing project-requirements. Authors developed an application to manage biological laboratory data that also provides work flow management. Kitano in Systems biology: A brief overview, 2002 discusses the need for good software infrastructure, collaboration and standards in order to be able to integrate analysis and computing packages to create simulation software in aiding systems biology research. Shah, et al., Goodman, et al., and Kitano provide an insight on the basic requirements to successfully manage data in systems biology projects that are applicable to metabolomics profiling.

As the systems biology research advances there is a need for advanced data integration techniques as the traditional techniques become too expensive to handle the new requirements (Brown, et al., 2005). Brown, et al., also suggested xml as the communication or data exchange language for better data integration and data search. Boyle, et al., 2009; Hoschek, et al., 2001; Tatarinov, et al., 2003; Wang, et al., 1997 proposed different data management and data sharing strategies with example applications they developed. All of these projects emphasize the importance of data management, integration and sharing in systems biology research. Advanced data management and integrations methods also come with the need for advanced computational programming and a need for dedicated programmer. But, it would be beneficial to researchers if the user interfaces are built that no programmer intervention is needed for maintenance of the project (Maier, et al., 2009).

The diversity and magnitude of data generated in systems biology needs web-based technologies to ensure proper integration and sharing (Deus, et al., 2008). Deus, et al., developed a prototype software product that is open-source that can be used in shared and distributed data management in integrative biomedical informatics. Cruz-Correia, et al., 2007 reviewed data integration of patient data in terms of evolution of systems in practice to meet patient needs. Cruz-Correia, et al., emphasized the importance of web-based technologies and messaging technologies in supporting data integration projects. Duncan, et al., 2001 developed a clinical data repository using internet protocols. This project involves rapid development of the application in a multi-disciplinary environment. They have also discussed about incorporating secure remote access to the data repository and diverse clinical systems over the internet. McCusker, et al., 2009 developed semantic data

warehouse called Corvus for caGrid^{6 7}. caGrid is the service oriented architecture that underlies caBig⁸. McCusker, et al., 2009 explained that the generalizable and widely usable Corvus system uses semantic integration to integrate data from distributed web services.

Mayer, 2009 in his literature review of data management in systems biology concludes that every project needs a home grown application as each and every project is unique based on the protocol, standards or systems used. He also concluded that web-based technologies have better advantages in multi-disciplinary environments like systems biology experiments. Bourne, et al., 2004 while talking about protein data bank (PDB) mentions that, users download PDB files and access them remotely using their own client application. This suggests that users still need a home grown application to process the data from community developed database or application.

The following studies concentrate on development of data integration and management applications using available open-source and commercial software products. Delaney, et al., 1997 discusses the use of relational database management system for creating automated, electronic progress notes from existing data sources which were not being utilized. Authors mention that relational databases are useful in rapid development of applications. They also suggest that using readily available products in developing data management solutions will save a lot of development time. Clayton, et al., 1996 expressed similar opinion in their study called “The economic motivations for clinical information systems”. While mentioning that economic factors control the development

⁶ <http://cagrid.org/display/cagridhome/Home>

⁷ <https://cabig.nci.nih.gov/workspaces/Architecture/caGrid>

⁸ <https://cabig.nci.nih.gov/>

of information systems in healthcare, they suggested that developing information systems from various components and vendors will reduce the time in development. U.S. Department of Energy Office of Science, 2009 announced on their website that the guiding principles of their systems biology knowledge base are open access, open source and open development. This suggests increasing importance of open-source software in collaborative projects and systems biology.

The following papers discuss how process automation will improve decision support, data management and process flow. Drummond, et al., 2004 automated data collection in ICU/critical care setting that help prevent paper flow sheets, time-consuming manual searches which they claimed in turn reduced information overload and errors. Vawdrey, et al., 2007 while assessing the data quality in manual entry of ventilator settings concluded that automated data collection in ICU setting improves charting efficiency, eliminate delays and reduce errors. Aukburg, et al., 1989 have experimented automated data collection in post anesthesia period to find that it increased sensitivity of adverse physiological event detection allowing better care. Hernandez, et al., 1999 found information management for critical care monitoring to be very difficult due to overwhelming amounts of data produced by a variety of monitoring devices and lack of automation. They developed automated atrial and ventricular activity detection using multi-sensor and multi-data fusion schemes. The application of these schemes were quantitatively evaluated and found to be better compared with current methods.

Systems biology not only depends on integration of heterogeneous data sources but also knowledge sources. Tu, et al., 2007 explained integration of heterogeneous knowledge sources using Standards-Based Active Guideline Environment (SAGE) system which

faces difficulties for wide adoption due to lack of standards in clinical information systems. Data management system development planned for the current project might face similar issues when it comes to wide adoption.

As most of the home grown applications are unique most of the times they cannot be compared to other systems for evaluation. In this situation, a satisfaction survey or an evaluation questionnaire would be useful to evaluate if the implemented system is better than the old system. Higgins, et al., 1991 built a graphical ICU workstation in 1991 and used questionnaires to evaluate it.

Almost all of the technologically advanced systems introduced will face some degree of rejection from users. Campbell, et al., 2006 tried to list the reasons for such a reaction from users. They called these reasons as unintended adverse consequences and tried to explain them in the context of Computerized Provider Order Entry (CPOE) system they have implemented. They grouped these consequences into 9 categories in the order of decreasing frequency with more/new work occupying top of the list and overdependence on the technology occupying the bottom. Even though it did not top the list, negative emotions is the unintended consequence more noticeable by the programmers in small research groups.

The reviewed literature supports several decisions made in building the current project and also points out the difficulties in user acceptance, wide spread usage and evaluation. The literature suggests that introduction of advanced technologies opens doors to better data management and integration. As almost all of the systems biology projects are multi-disciplinary, a web-based system would be more beneficial in user communication and

sharing data. A flat file database like excel would not benefit rapid application development or huge datasets. Use of relational database management system accelerates the application development process and also provides numerous advantages over flat file database systems. There are several community developed web-based software applications available for metabolomics but they are useful only after several data management tasks have been performed at the project level. So, a home-grown application is necessary to manage these projects. But, developing a home-grown application from scratch is not economical and takes lot of time. The literature suggests that developing these applications from pre-existing software will reduce the development time without compromising quality of the product. The literature points out the importance of open-source software products in these projects. They also suggest that making these projects available as open-source will lead to further development of the product. Usually critical care environments are full of data collection monitors whose data is not being utilized for research either due to the lack of proper data integration mechanism or difficulty in managing enormous amounts of data. Human interaction in utilizing this data might lead to errors causing problems in proper decision making. The literature suggests that an automated system that can collect and integrate data from these monitors with added programmatic decision support will aid in better care and research. These projects face several problems like unintended adverse consequences, lack of proper evaluation mechanism, lack of standards etc. The literature mentions these drawbacks and suggests possible alternatives in overcoming these difficulties.

The current project follows the suggestions of previous literature in many aspects including but not limited to, building web-based technologies, using relational database

management systems, using a hybridization of open-source and commercial software products, electronic form based data entry, automated data collection and integration, decision support, advanced data query mechanisms, data export, and data analysis.

Materials and Methods

The SCC laboratory consists of multidisciplinary team conducting pre-clinical research utilizing flat file database like Microsoft Excel. The University of Minnesota was a good platform for this project as it has huge information technology support. The information technology groups that support research at the University of Minnesota are Academic Health Center, Office of Information Technology and Minnesota Supercomputing Institute. Out of these, Minnesota Supercomputing Institute is dedicated to provide services to researchers and their teams free of charge. That was a great economic incentive to try and develop a home grown data management application using open-source and commercial software products supported by MSI.

Some of the researchers at SCC laboratory were in the process of developing a data management strategy while others started data collection. There was a need for data interchange and a continuous modification of the data management system according to the evolving needs of researchers. The software development model that suits these needs is called “Extreme programming” (Figure 1) which is a type of agile software development (Lindstrom & Jeffries, 2004).

Even though the software was developed for pre-clinical research, the research itself was aimed at better healthcare outcomes. Hence it also needs a health informatics model with three key elements, data, information and knowledge (DIK). DIK model is most commonly represented in the form of a pyramid with data occupying the bottom, information the middle and knowledge the top of the pyramid. A combination of health informatics model and extreme programming model is used to develop this project.

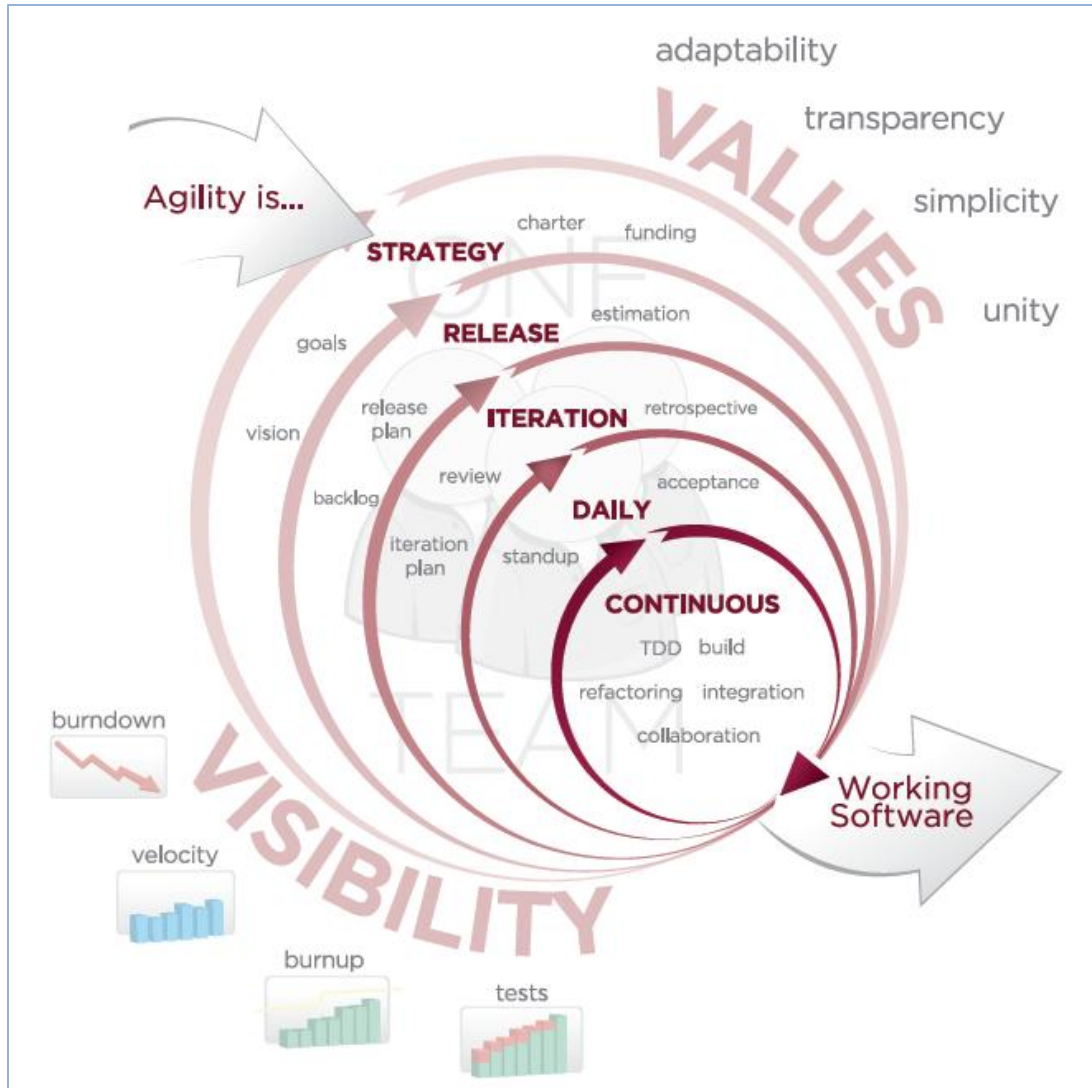


Figure 1 Extreme Programming⁹

Extreme programming takes part in each of the key elements of health informatics model. This project’s development starts with reviewing problems in data management followed by technological choices, component development, data integration, and ends with data analysis. This process follows DIK model of health informatics.

⁹ Used with permission from “VersionOne” <http://www.versionone.com>

Review of Problems in Data Management

A review of data management problems faced by researchers and laboratory personnel of the Surgical Critical Care laboratory was performed.

1. Enormity of data

Each experiment of each research protocol lasts for few hours to 72 hours. The data generated can be divided into vitals and blood gases, labs, Glasgow Coma Scale (GCS) scores, Thromboelastography (TEG), binning and nuclear magnetic resonance (NMR) data groups.

Some vitals data are collected every 10 min and all of the vitals and blood gases data are collected every one hour during the experiment (Table 1).

A total of 13 laboratory variables are collected 9 times over a full period of experiment (Table 2).

Glasgow Coma Scale (GCS) scores were collected 4 times over the entire period of an experiment.

Table 1 Vitals and Blood Gas variables collected during experiment.

S. No	Variable Name	Collected at 10 min intervals	Collected at one hour intervals
1	Arterial Diastolic Pressure	✓	✓
2	Arterial Oxygen Saturation	✗	✓
3	Arterial Partial Pressure Of Carbon Dioxide	✗	✓
4	Arterial Partial Pressure Of Oxygen	✗	✓
5	Arterial pH	✗	✓
6	Arterial Systolic Pressure	✓	✓

S. No	Variable Name	Collected at 10 min intervals	Collected at one hour intervals
7	Base Deficit	✗	✓
8	Bispectral Index	✓	✓
9	Bladder Pressure	✗	✓
10	Calcium	✗	✓
11	Cardiac Output	✗	✓
12	Chest Tube Output	✗	✓
13	Fraction Of Inspired Oxygen	✓	✓
14	Glucose	✗	✓
15	Heart Rate	✓	✓
16	Hemoglobin	✗	✓
17	Lactate	✗	✓
18	Potassium	✗	✓
19	Propofol given	✓	✓
20	Pulmonary Artery Diastolic Pressure	✓	✓
21	Pulmonary Artery Systolic Pressure	✓	✓
22	Sodium	✗	✓
23	Temperature	✓	✓
24	Tissue Hemoglobin Index 15 mm	✓	✓
25	Tissue Hemoglobin Index Multi-Depth	✓	✓
26	Tissue Oxygen Saturation 15 mm	✓	✓
27	Tissue Oxygen Saturation Multi-Depth	✓	✓
28	Urine Output	✗	✓
29	Venous Oxygen Saturation	✗	✓
30	Venous Partial Pressure Of Carbon Dioxide	✗	✓
31	Venous Partial Pressure Of Oxygen	✗	✓
32	Wedge Pressure	✗	✓

Table 2 Laboratory variables collected during experiment.

S. No	Variable Name
1	Alanine Aminotransferase
2	Albumin
3	Alkaline Phosphatase
4	Aspartate Aminotransferase
5	Creatinine
6	Creatinine Kinase
7	Lactate Dehydrogenase
8	Platelet Count
9	Total Bilirubin
10	Total Protein
11	Urea Nitrogen
12	Urine Creatinine
13	Urine Osmolality

Thromboelastography (TEG) data is generated in the form of text file which will contain variables named R min, K min, Angle and MA, a total of 4 collected 9 times over the entire period of an experiment.

Binning data is generated as a text file containing a maximum of 250 variables (average binning range values) collected six times over the entire period of an experiment. There are deviations in binning data collection of different tissue samples and so should be treated separately.

NMR (Nuclear Magnetic Resonance) spectroscopy data is collected with two different techniques named Phosphorous NMR and Proton NMR. The variables are compounds of interest selected by the researchers and are different for different tissue

samples. They are about 20-50 without a defined minimum and maximum number of variables collected 6 times over the entire period of an experiment.

Real-time data includes three text files named with starting letter of the commercial name of the collecting monitor followed by the date and time the file was created. For the sake of identification, they will be mentioned as S-file, H-file and G-file. The variables stored in each of these files are listed in Table 3.

Table 3 Real-time variables collected.

S. No	H-file variables collected at two second intervals	S-file variables collected at five second intervals	G-file variables collected at one hour intervals
1	Tissue Hemoglobin Index 15 mm	Cardiac Output ¹⁰	Arterial Oxygen Saturation
2	Tissue Oxygen Saturation 15 mm	Diastolic Arterial Pressure	Arterial Partial Pressure of Carbon dioxide
3		Diastolic Pulmonary Arterial Pressure	Arterial Partial Pressure of Oxygen
4		Oxygen Saturation	Arterial pH
5		Systolic Pulmonary Arterial Pressure	Calcium
6		Systolic Arterial Pressure	Glucose
7			Hemoglobin
8			Lactate
9			Potassium
10			Sodium
11			Venous Oxygen Saturation
12			Venous Partial Pressure of Oxygen

¹⁰ Cardiac output is collected at one hour intervals

A flat file database like MS Excel will not be able to handle such huge amounts of data effectively.

2. Amount of time spent in data collection and data entry

A considerable amount of time is spent on each experiment collecting data manually on a paper and entering the data into Excel files. Even though some of the data is in the form of well-formatted text files and easier to import into Excel, combining the data of different subjects is time consuming. Usual number of experiments per month will range from two to eight. When the number of experiments per month reach the maximum, it will be very hard to finish data entry without a lag.

3. Need for streamlining data collection, integration, display, exchange and export

The laboratory personnel of SCC come from diverse backgrounds and were assigned specific sub-group of data management or analysis duties. At times, the data is concentrated in the laboratory sub-groups and causes confusion and difficulty in data integration, exchange and regular update. To display or export a pre-defined set of variables users have to go through the entire flat file database deleting unwanted columns and rows and modifying the data which needs some amount of MS Excel application knowledge and time to spare.

4. Real-time data management

Real-time data has not been used for data analyses due to the unstructured raw text files (Figure 2, Figure 3 and Figure 4) and enormity of data. In order to utilize this data a new data management strategy is needed.

```

2132 10:39:47 123/ 73 93 26/ 14 20
2133 10:39:52 122/ 72 92 26/ 14 20
2134 10:39:57 125/ 74 93 26/ 14 20
2135 10:40:02 126/ 75 94 26/ 15 20
2136 10:40:07 118/ 70 91 26/ 14 20
2137 10:40:12 119/ 70 91 25/ 13 18
2138 10:40:17 126/ 75 93 26/ 11 18
2139 10:40:22 124/ 73 93 26/ 12 18
2140 10:40:27 124/ 74 93 27/ 13 19
2141 10:40:32 126/ 75 94 26/ 13 19
2142 10:40:37 119/ 71 92 28/ 14 19
2143 10:40:42 119/ 70 91 28/ 13 20
2144 10:40:47 126/ 75 94 27/ 13 20
2145 10:40:52 122/ 72 92 27/ 13 20
2146 10:40:57 123/ 73 92 27/ 14 20
2147 10:41:02 121/ 72 92 27/ 13 20
2148 10:41:07 120/ 72 92 26/ 13 20
2149
2150          SPACELABS MEDICAL PATIENT DATA LOGGER
2151 PATIENT NAME                               Bed # SL001 DATE 19 OCT 2009
2152
2153          ART mmHg          PA mmHg          CO
2154 TIME          SYS/DIA MEAN  SYS/DIA MEAN  TIME LPM
2155 *****          *****          *****          *****
2156 10:41:08 120/ 72 92 26/ 13 20 10:40 2.3
2157 10:41:13 127/ 76 94 26/ 15 20 10:40 2.3
2158 10:41:18 127/ 76 93 26/ 13 21 10:40 2.3
2159 10:41:23 121/ 71 92 26/ 14 20 10:40 2.3
2160 10:41:28 123/ 72 92 26/ 14 20 10:40 2.3
2161 10:41:33 122/ 72 92 27/ 15 20 10:40 2.3
2162 10:41:38 121/ 72 92 27/ 15 20 10:40 2.3
2163 10:41:43 127/ 76 94 27/ 14 21 10:40 2.3
2164 10:41:48 127/ 75 94 29/ 15 21 10:40 2.3
2165 10:41:53 121/ 71 92 28/ 12 21 10:40 2.3
2166 10:41:58 121/ 71 92 28/ 13 21 10:40 2.3
2167 10:42:03 124/ 74 93 27/ 15 20 10:40 2.3
2168 10:42:08 123/ 73 93 27/ 15 20 10:40 2.3
2169 10:42:13 126/ 74 93 28/ 14 21 10:40 2.3
2170 10:42:18 120/ 71 93 27/ 14 20 10:40 2.3

```

Normal text file 960098 chars 1001138 bytes 20521 lines

Figure 2 S-file with real-time data

```

3040 2009-10-19, 10:39:40, 0, --, --
3041 2009-10-19, 10:39:42, 0, --, --
3042 2009-10-19, 10:39:44, 0, --, --
3043 2009-10-19, 10:39:46, 0, --, --
3044 2009-10-19, 10:39:48, 0, --, --
3045 2009-10-19, 10:39:50, 0, --, --
3046 2009-10-19, 10:39:52, 0, --, --
3047 2009-10-19, 10:39:54, 0, --, --
3048 2009-10-19, 10:39:56, 0, --, --
3049 2009-10-19, 10:39:58, 0, --, --
3050 2009-10-19, 10:40:00, 0, --, --
3051 2009-10-19, 10:40:02, 0, --, --
3052 2009-10-19, 10:40:04, 0, --, --
3053 2009-10-19, 10:40:06, 0, --, --
3054 2009-10-19, 10:40:08, 0, --, --
3055 2009-10-19, 10:40:10, 1, 93, 7.9
3056 2009-10-19, 10:40:12, 1, 91, 6.9
3057 2009-10-19, 10:40:14, 1, 90, 7.3
3058 2009-10-19, 10:40:16, 1, 90, 6.7
3059 2009-10-19, 10:40:18, 1, 88, 5.9
3060 2009-10-19, 10:40:20, 1, 88, 6.6
3061 2009-10-19, 10:40:22, 1, 87, 6.6
3062 2009-10-19, 10:40:24, 1, 83, 6.6
3063 2009-10-19, 10:40:26, 1, 82, 6.6
3064 2009-10-19, 10:40:28, 1, 82, 6.6
3065 2009-10-19, 10:40:30, 1, 82, 6.7
3066 2009-10-19, 10:40:32, 1, 81, 6.8
3067 2009-10-19, 10:40:34, 1, 82, 7.1
3068 2009-10-19, 10:40:36, 1, 83, 7.3
3069 2009-10-19, 10:40:38, 1, 84, 7.0
3070 2009-10-19, 10:40:40, 1, 84, 6.7
3071 2009-10-19, 10:40:42, 1, 84, 6.9
3072 2009-10-19, 10:40:44, 1, 84, 7.0
3073 2009-10-19, 10:40:46, 1, 85, 6.8
3074 2009-10-19, 10:40:48, 1, 85, 6.9
3075 2009-10-19, 10:40:50, 1, 86, 7.2
3076 2009-10-19, 10:40:52, 1, 86, 7.1
3077 2009-10-19, 10:40:54, 1, 87, 7.3
3078 2009-10-19, 10:40:56, 1, 87, 7.3

```

Normal text file 1424780 chars 1512776 bytes 43999 lines

Figure 3 H-file with real-time data

```

100 R 11 ^^^Na+DriftB -2 mmol/L
101 R 12 ^^^Na+MeasuredB 139 mmol/L
102 R 13 ^^^pCO2DriftB 0 mmm3mHg
103 R 14 ^^^pCO2MeasuredB 32 mmHg
104 R 15 ^^^pHDriftB 0.00
105 R 16 ^^^pHMeasuredB 7.40
106 R 17 ^^^pO2DriftB 2 mmHg
107 R 18 ^^^pO2MeasuredB 176 mmHg
108 L 1
109
110 XXXXXXXXXX GEM 3000^5.5.3.US1 ^13137^^972480^2.4 20091015085124
111 P 1 MONR101209
112 O 1 56 A
113 C 1 I PR 48 G
114 R 1 ^^^BE (B) 7.4 mmol/L 20091015085041
115 R 2 ^^^Ca++ 1.29 mmol/L
116 R 3 ^^^Glu 96 mg/dL
117 R 4 ^^^HCO3- 31.1 mmol/L
118 R 5 ^^^Hct 19 %
119 R 6 ^^^K+ 3.3 mmol/L
120 R 7 ^^^Lac 0.8 mmol/L
121 R 8 ^^^Na+ 141 mmol/L
122 R 9 ^^^SO2c 97 %
123 R 10 ^^^THbc 5.9 g/dL
124 R 11 ^^^pCO2 39 mmHg
125 R 12 ^^^pH 7.51
126 R 13 ^^^pO2 87 mmHg
127 R 14 ^^^Temp 37.0 C
128 L 1
129
130 XXXXXXXXXX GEM 3000^5.5.3.US1 ^13137^^972480^2.4 20091019095353
131 P 1 MONR101909
132 O 1 57 A
133 C 1 I PRE-BASELINE G
134 R 1 ^^^BE (B) 7.4 mmol/L 20091019095351
135 R 2 ^^^Ca++ 1.35 mmol/L
136 R 3 ^^^Glu 133 mg/dL
137 R 4 ^^^HCO3- 31.2 mmol/L
138 R 5 ^^^Hct 24 %

```

Normal text file 27409 chars 29927 bytes 1260 lines

Figure 4 G-file with real-time data

5. Lag between data collection and analyses

There are two reasons for the lag between data collection and analysis. One reason is the time consumed in data collection, entry and integration. This delays the submission of the final dataset to the biostatistician to perform statistical analyses. The other reason is varying levels of biostatistical software and data analysis expertise of laboratory personnel, which makes it hard for them to perform statistics on the data.

After reviewing these data management issues and data needs of SCC laboratory, a decision has to be made to develop a standalone or web-based application before making technological choices.

Standalone vs. Web-based Application Development

The first step towards the application development is to determine whether it is going to be a standalone application or a web-based. Standalone and web-based technologies have their own advantages and disadvantages. Debates continue on determining the best technology¹¹ in various situations. The best answer for the question will depend on the needs of the client.

The SCC laboratory needs a research management application that is easy to use, manage the data, store the data in a location accessible to all the laboratory users, export the data in a variety of formats, and perform statistical analyses. All of these tasks can be achieved by both standalone and web-based technologies except that as per definition, standalone application cannot be used over the internet and automated real-time data cleaning and extraction is not possible with web-based application due to limited access of web browsers to client machines. For statistical analyses using a standard, well accepted, standalone technology would be more appropriate. There are no specific set of rules for building a graphical user interface in case of standalone technologies but there are standards set (W3C web standards)¹² and followed by many users for web-based user interface programming. To address these needs a better choice would be using both standalone and web-based technologies. Figure 5 shows data flow with the use of standalone and web-based technologies. Software icons used in the figure represent the current standard icons used by the software vendors.

¹¹ <http://www.nairaland.com/nigeria/topic-15881.0.html>

¹² <http://www.w3.org/>

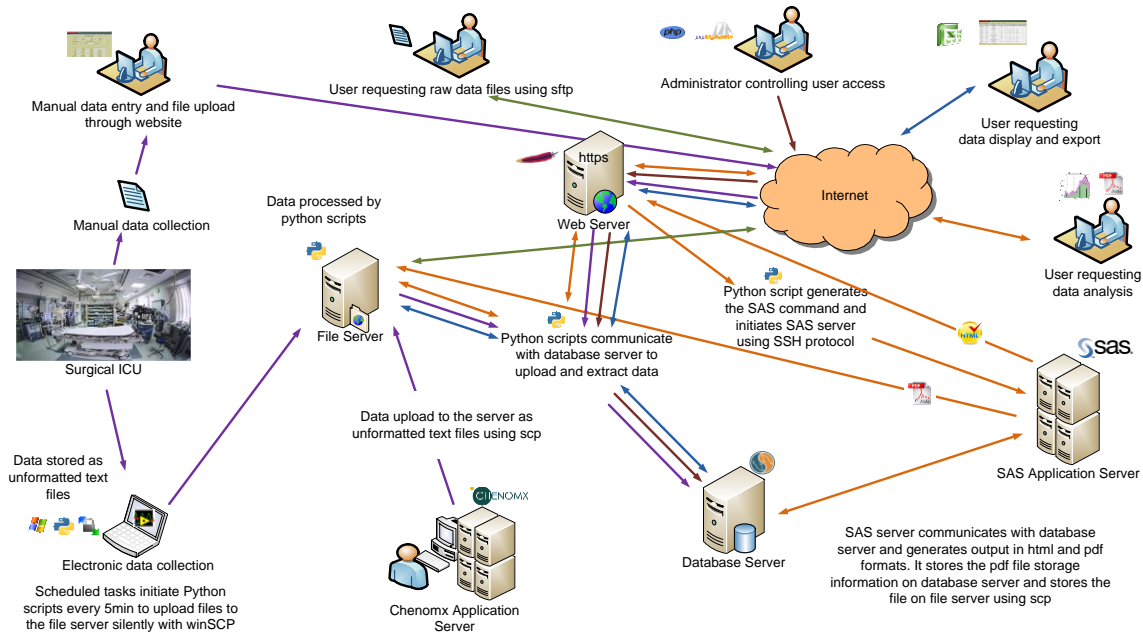


Figure 5 Data flow with Standalone and Web-based technology usage

Technological choices and hosting services

The advantage of web-based technologies is that components required for building an application are readily available where many of them are open-source with general public license, and have been proven to be dependable and flexible. Some of the technological choices are limited to commercial software due their efficiency and acceptance. So, a hybridization of open-source and commercial software technologies was planned to achieve the desired result.

Office of Information Technology (OIT) at the University of Minnesota offers free web hosting services for departments at the University. At first, it was considered as the project host but the lack of database back-end forced finding a different project host¹³. Minnesota Supercomputing Institute (MSI) provides supercomputing resources and

¹³ <http://www.oit.umn.edu/services-systems/index.htm>

support to faculty and their research groups without a fee for services¹⁴. MSI in general does not support web hosting to the extent of offering other supercomputing services but supports the individual user profile, which allows installation of certain server side software. The MSI follows a multi-tier system where its web server, database server and SAS application server are on different physical partitions. MSI offered Apache web server, MySQL/Oracle database server, support for a variety of languages like Python, Perl and PHP, file server with sftp capability, ssh tunneling, and remote login to the SAS application server.

Python¹⁵ was chosen as server side programming language for a variety of advantages¹⁶¹⁷, compared to other object-oriented programming languages and also familiarity with the language. Even though the latest Python version available was 3.1.2, version 2.6 was used due to its compatibility with MySQL database management system version 5.x, MS Excel compatible modules and extensive user community support. phpMyAdmin¹⁸ was used for administration of MySQL database over the internet and it was written in PHP¹⁹ language. SAS application version 9.2 was provided by MSI, which is the latest version of SAS available. R statistical software is also considered for the project but planned only for future implementation due to SAS's standards and acceptance. Apache²⁰ is the open-source web server provided by MSI for web hosting which competes equally in quality and the number of users, with commercial web server products²¹. MySQL is selected as

¹⁴ <https://www.msi.umn.edu/about/>

¹⁵ <http://www.Python.org/>

¹⁶ <http://wiki.Python.org/moin/LanguageComparisons>

¹⁷ http://en.wikipedia.org/wiki/Comparison_of_programming_languages

¹⁸ http://www.phpmyadmin.net/home_page/index.php

¹⁹ <http://php.net/index.php>

²⁰ <http://www.apache.org/>

²¹ http://en.wikipedia.org/wiki/Comparison_of_web_server_software

the relational database management system of choice after considering its similarities in features and open-source nature compared to Oracle database management system²².

Client-side web programming follows W3C standards for XHTML 1.0 transitional and utf-8 character set. XHTML, CSS, JavaScript, XML, XSLT, RSS and a combination of these, which is famously called AJAX (Asynchronous JavaScript and XML) technology, is used. Flash programming was not used, as the browsers in the University of Minnesota computers did not support it.

A combination of Python, Visual basic, WinSCP²³ an SFTP client were used to upload the real-time data text files generated on a MS Windows XP Professional operated laptop to the server. The task is driven by Windows Task Scheduler, which initiates the process every 3 minutes.

The Python programming language is also used for server side processing of excel files, text files, SSH logins, initiating SAS server and several other processes not related to the web programming.

Table 4 shows the role of these software products in data management. It lists most of the software used on development and production machines for project development.

Several other software products like PuTTY, Xming, MySQL Workbench, Firefox web browser and its plugins like Firebug, Colorzilla, MeasureIt etc., Internet Explorer, Safari and Opera web browsers, and Adobe Captivate also contributed towards the project development.

²² http://download.oracle.com/docs/cd/E12151_01/doc.150/e12155/oracle_mysql_compared.htm

²³ <http://winscp.net/eng/index.php>

Table 4 List of software used.

S. No	Software used for development	Software provided by MSI on the server	Software installed on the server not supported by MSI
1	7zip		7zip
2	Apache	Apache	
3	jQuery		jQuery
4	jQuery gui tools		jQuery gui tools
5	LabView ²⁴		
6	Microsoft Office ²⁴		
7	MS Visual Studio ²⁴		
8	MySQL v5	MySQL v5	
9	Notepad plus plus		
10	PHP	PHP	
11	phpMyAdmin		phpMyAdmin
12	Putty	Bash shell	
13	Python add-on for Visual Studio		
14	Python v 2.6	Python v 2.4	Python v 2.6
15	R		R
16	SAS v 9.2 ²⁴	SAS v 9.2 ²⁴	
17	Windows 7 OS, Windows XP ²⁴	Suse Linux	
18	WinSCP		
19	Xming		

Database Development

MSI provides a shared MySQL database server, which leaves the users with one database administration account with all privileges within the database. The database was provided with default MyISAM type table generation with utf8 character set and utf8_general_ci collation. Even though MyISAM table type allows ease of administration, it only allows table locking. InnoDB table type allows row locking while updating the data, allowing users access to the rest of the table rows²⁵. Hence, InnoDB table type is forced for all the tables created in the database with character set utf-8 and collation utf8_bin. The

²⁴ These are commercial software and the rest are free software with most of them being open-source

²⁵ http://tag1consulting.com/MySQL_Engines_MyISAM_vs_InnoDB

collation utf8_bin is binary and uses case sensitivity while utf8_general_ci is case insensitive.

Database design shows one-to-many relationships between the tables. Authentication table is for user account management for website login. Identifier table is the subject identification information for research protocols. All the tables with subject information in the database have subject id as the foreign key pointing to the subject id in the identifier table where it is primary key with options on update cascade and on delete cascade, which makes sure the changes made to the identification table will reflect in rest of the tables. This also denies the wrongful entry of the data into database without an existing subject id in the identifier table.

For user training and experimentation purposes a mirror image of the database tables is created for use with a test website.

A detailed “create” syntax for each table is listed in Appendix. Binning, HNMR, PNMR data tables are created automatically when a user uploads the respective file. All the other tables were created manually before the data were entered. Figure 6 illustrates the database structure. Each table displays the number of variables in it in the form of “n more” where n is the number of variables in the table. Data tables with no relation to other tables like user login info, studies, monr_event_definitions, monr_filerowupdate realtime and monr_temp_sql_query_storage were not included in the figure.

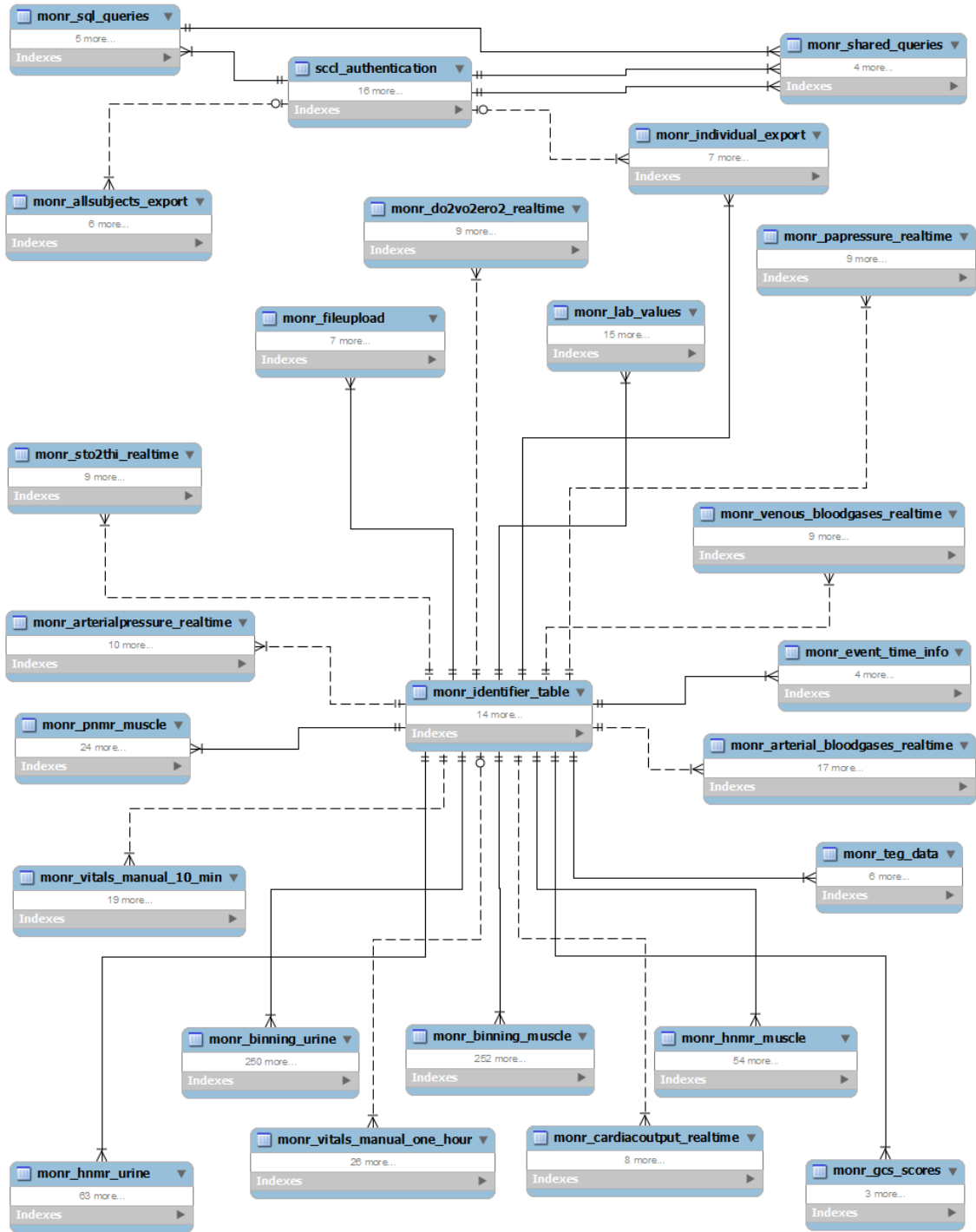


Figure 6 Database Structure²⁶

²⁶ For explanation of entity relationships visit the link <http://www2.cs.uregina.ca/~bernatja/crowsfoot.html>

Website Development

MSI is a shared resource environment where one web server will be serving multiple virtual servers. This is a positive feature for the hosting service provider as it uses fewer resources but presents a security threat to the users. The web folders are on a Suse Linux machine that allows users to define varying folder permissions. Any of the files or folders should not be given write permissions to the global (other) user so that they cannot change the file content or place their own code in those folders.

The root folder named www was given permissions²⁷ 751(rwxr-x--x) restricting the access of global users to the root folder's contents. The documents folder is named htdocs and generally used for html, images, style sheets, JavaScript etc. where there is no security risk if the global (other) user is able to read those files. The cgi folder named cgi-bin hosts server scripts and will not serve any files other than server scripts. However, to enhance the security of the website html files were placed in cgi-bin directory and server scripts load the appropriate html files based on user request. The website is served with secure http protocol with web address <https://sccl.umn.edu>.

Figure 7 shows directory tree of the website on the production server. Cgi-bin directory consists of three sub-directories, SCCL directory hosting the files for main website, SCCL_test hosting the files for test website, rss directory hosts files and directories for a prototype of rss feed of pubmed articles to embed in the website in future. Htdocs directory hosts files and directories like images, js, stylesheets for the main website, test directory in htdocs hosts directories and files for test website, dbmanager directory hosts files and directories of phpMyAdmin software, Admin and testupload directories are test

²⁷ <http://www.zzee.com/solutions/linux-permissions.shtml>

directories that cannot be accessed over the website, and rss is the directory hosting files and folders for an rss prototype.

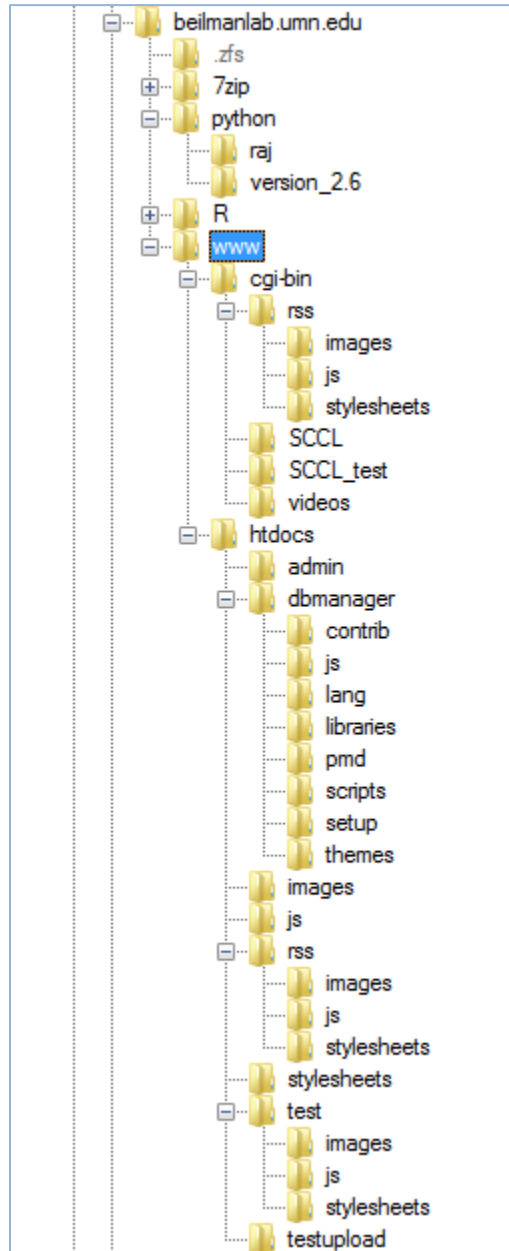


Figure 7 Directory tree of the website on the production server

Website Menu

1. Data Entry Forms

Several data entry forms were created for manual data entry of Vitals and Gases, Labs and GCS data. All the forms are typical web forms with added decision support aiding the user not making mistakes while entering the data. The forms communicate with server scripts (cgi scripts) using Ajax²⁸. The advantage of Ajax communication compared general communication is that in Ajax only the data requested by the user is transmitted without the entire webpage being sent by the server reducing the bandwidth usage and avoids page refreshes. This technology comes with some disadvantages like the browsing history not being stored in the browser, no page refresh and hence no feedback to the user about data being sent/received etc. This technology is effective where there is no use of maintaining the state of the webpage or storing the browsing history.

The difference between common data entry forms and these web forms is that these forms have a data display window that is updated whenever the data is entered. This gives the user feedback that the data has been properly entered in the database. Users can see if the data is not being represented properly in the database and inform the database administrator and web programmer. The data display mimics a regular spreadsheet with a static headers row. When a user makes a mistake or an auditor finds a mistake in the data entry that can be corrected directly on the data display spreadsheet by double-clicking on the field and modifying the value. Instead of using a data entry form, a spreadsheet-like form could have been implemented as in Google documents. But, as the number of

²⁸ <http://www.w3schools.com/Ajax/Default.Asp>

variables of the data entry forms usually won't fit into one screen users can't see their mistakes while entering the data unless they scroll the webpage horizontally (Figure 8).

The screenshot shows a web form titled "Vitals" with various input fields for medical data. Below the form is a table displaying a list of vitals records. The table has the following columns: Date, Time, Event, HR, Art BP, Mean Art Pres, PA Pres, Wedge Pres, and Urir Outp. The data in the table is as follows:

Date	Time	Event	HR	Art BP	Mean Art Pres	PA Pres	Wedge Pres	Urir Outp
02/03/2009	10:37:00	FR-20	23	108/53	71.0	30/19	12	43
02/03/2009	10:27:00		99	108/55	72.0	30/18		
02/03/2009	10:17:00		99	102/51	68.0	31/18		
02/03/2009	10:07:00	Towel On	99	100/49	66.0	29/17		55
02/03/2009	9:57:00		99	105/52	69.0	30/17		
02/03/2009	9:47:00		99	99/51	67.0	30/18		
02/03/2009	9:37:00	FR-19	22	101/51	67.0	31/17	12	59
02/03/2009	9:27:00		99	111/54	73.0	30/19	11	
02/03/2009	9:17:00		99	114/60	78.0			
02/03/2009	9:07:00		99	105/51	69.0			62
02/03/2009	8:57:00		99	97/48	64.0			
02/03/2009	8:47:00		99	103/50	67.0			
		FR-18, 5mL Bolus of						

Figure 8 Web-forms example – Vitals

The approach in this website towards having regular web forms, with Ajax communication and data display reduced the chances of errors and improved the ease of data entry.

2. Data

Data menu of the webpage consists of sub-menus Simple Export, Advanced Export, My Queries and Shared Queries. They help the researcher to extract the data from the database and store the queries.

a) **Simple Export**

Simple export consists of a variety of pre-defined SQL queries that will extract the data from the database and create excel files for download. A new Python module was written using “xlwt” Python module and was named “raj_excel_export”. Administrators are given access to update the excel files so that they are update after each experiment. The advantage of Simple Export page is that it is simple to use and mimics the spreadsheets the team has been using for a while. But, it is not ideal to use Simple Export page for real-time data. Every time the “Update Excel Files” button is pushed, a series of SQL queries are executed and excel files are created. This will cause a lot of strain over the database server and webserver and will keep the user waiting until the files are updated. Individual excel file update is not ideal as the researchers could miss updating one of the files. So, a better data export system is needed in order to handle large amounts of data and get up-to-date data from the database without a manual update. Figure 9 shows screenshot of simple export webpage.

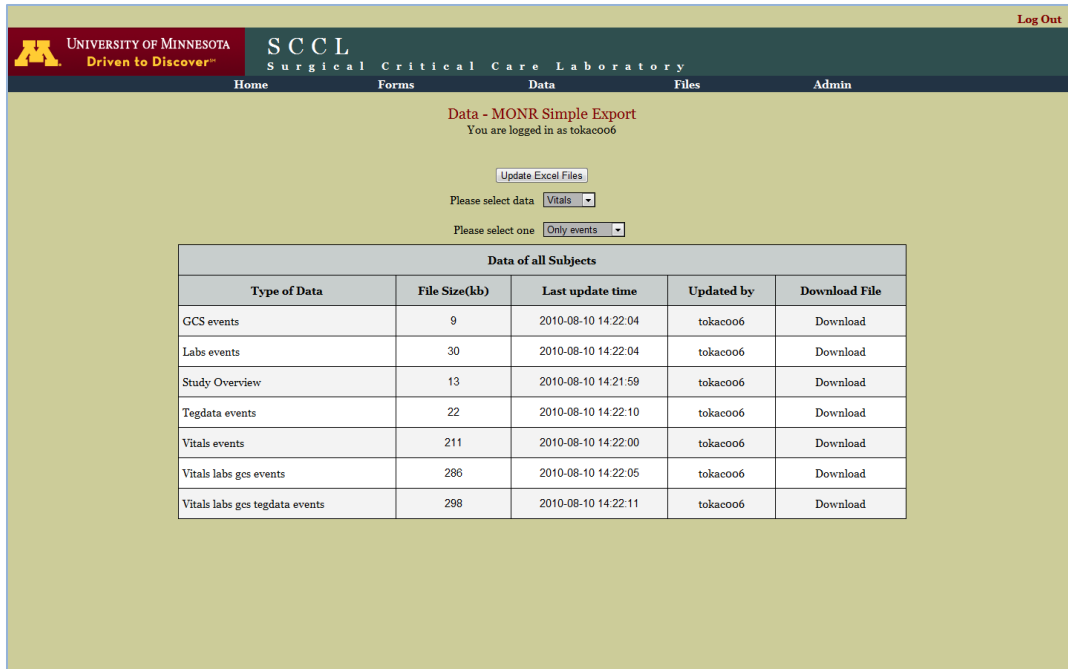


Figure 9 Simple Export webpage

b) Advanced Export

“Advanced Export” page was developed to satisfy the data needs where Simple Export page could not. This page consists of a series of data selection boxes appearing based on the user’s selection and loading the variable names from the database using Ajax technology. jQuery²⁹ was used for a better user experience. Figure 10 shows screenshot of “Advanced Export” variable selection page.

²⁹ <http://jquery.org/about>

Figure 10 Advanced Export variable selection webpage

After the selection of variable and the frequency of variables, users should submit the form which will open a pop-up window with the display of results with the latest data available in the database. Figure 11 shows part of the query results pop-up page. Webpage provides options to print the output (if the data fits the browser window horizontally else browser print options need to be changed) or export the data in the form of an Excel or CSV file. The CSV file option is provided as there are limitations for the maximum number of rows and columns that an Excel file can hold depending

on its version³⁰. “raj_excel_export” module was updated to serve the Advanced Export page to output Excel and CSV files. 97-2003 Excel version was used in order to maintain compatibility with the Python module “xlwt” on which “raj_excel_export” is dependent on. Inner-Join is used between MySQL tables to export the data. Researchers wanted to have data output that is similar to the spreadsheets they used to use to have a better data understanding. This task needs Uni-Join procedure which is not available in MySQL. There was an alternative proposed for Uni-Join over several webpages but they need a lot of Left-Joins and Unions between tables which might slow the query considerably.

identifier table Pig_ID	vitals Event_name	vitals Art_Dia	vitals Art_Sys	vitals Arterial_O2_sat
020209	1	60	104	99
020209	2	24	46	99
020209	3	40	74	96
020209	4	53	99	98
020209	5	56	105	97
020209	6	74	124	97
020209	7	60	122	97
020209	8	63	127	96
020209	9	113	122	97
020209	10	53	108	97
020209	11	49	110	97
020209	12	42	89	99
020209	13	47	92	98
020209	14	45	99	96

Figure 11 Advanced Export query results pop-up

The pop-up webpage displaying the data of “Advanced Export” query has a button called “Save query”. On click, this slide opens a box (Figure 12) asking the user to

³⁰ <http://office.microsoft.com/en-us/excel-help/row-height-and-column-width-limits-in-excel-HA010137545.aspx>

save the query by giving it a name, description and mention other users of the database with whom the user would like to share the query. After clicking on the “Save” button, the user will get a feedback that the query was saved if it was successful. These saved queries show up in the user’s “My Queries” page and “Shared Queries” page of other users with whom it was shared.

Share with	
All users	<input checked="" type="checkbox"/>
<i>User Name (User ID)</i>	
Andrew H. Nguyen (nguy0833)	<input checked="" type="checkbox"/>
Daniel R. Lexcen (lexc0003)	<input checked="" type="checkbox"/>
Drew Scribner (scri0022)	<input checked="" type="checkbox"/>
Duncan N. Nyangau (mako0030)	<input checked="" type="checkbox"/>
Eileen E. Jackson (ejackson)	<input checked="" type="checkbox"/>
Elizabeth R. Lusczek (lusc0006)	<input checked="" type="checkbox"/>

Figure 12 Advanced Export query save box

c) **My Queries**

“My Queries” page lists all the queries saved by the user with information of date and time of the saved query, name of the person it was shared with or number of people it was shared with in case of more than one. It also provides options to edit the query information and sharing or, delete it. JQuery is used for better user experience and

sorting of the queries based on name and date. Clicking on the name of the query will open a pop-up window with the results of the query avoiding the repeated selection of data variables using “Advanced Export” page by the user. Figure 13 shows a screenshot of the “My Queries” page.





Query Name	Saved on	Shared with	Description	
Survival	2010-04-25 20:57:13	18 users	Date_of_experiment and Time_of_experiment columns represent Date and time at Baseline (Event 1) respectively. Experiment_end_date column represents 'Date of death' in case of subjects that died.	 
Test	2010-09-24 06:36:50	18 users	No description entered	 

Figure 13 My Queries

d) Shared Queries

“Shared Queries” page is almost similar to “My Queries” page except for a couple options. Every user can save the queries shared with them into their own “My Queries” page by clicking on a save button. This will allow the users to save the queries they think are useful on a long-term basis. The delete button will remove the query sharing for the user but will not delete the query itself. Figure 14 shows a screenshot of the Shared Queries page.



Query Name	Saved on	Shared by	Description	
Heart Rate All Events	2010-04-29 13:59:45	test0002 test0002 (test0002)	No description entered	 

Figure 14 Shared Queries

3. Files

The “Files” menu holds “File Upload” sub-menu in order to upload a variety of files to the file server. Excel files created before the creation of web forms were uploaded for maintenance of source and audit purposes. TEG Data, Binning, HNMR and PNMR file options are created to upload text files generated by a variety of tasks including but not limited to automated machine collected data, software generated data and spectroscopy data. This webpage also displays the files that have been already uploaded to the server below the “File Upload” form using Ajax technology. Figure 15 shows a screenshot of the webpage “File Upload”.

Files - MONR File Upload
You are logged in as tokacoo6

File Upload

Select Pig ID:

Please select the file type you are uploading

Please select the sample origin

Enter tissue weight in milli grams:

Enter lyophilized tissue weight in milli grams:

Please select the time point

Type of Data	File Name	File Type	File Size(kb)	Time of Upload	Download File
HNMR Muscle 1	Mo20209MB_1D_1H_HCN.txt	Text	1	2010-02-10 15:53:07	Download
HNMR Muscle 2	Mo20209MS45_1D_1H_HCN.txt	Text	1	2010-03-11 13:32:35	Download
HNMR Muscle 5	Mo20209MFR2_1D_1H_HCN.txt	Text	1	2010-03-11 13:32:35	Download
HNMR Muscle 11	Mo20209MFR8_1D_1H_HCN.txt	Text	1	2010-03-11 13:32:35	Download
HNMR Muscle 23	Mo20209MFR20_1D_1H_HCN.txt	Text	1	2010-03-11 13:32:35	Download
HNMR Muscle 25	Mo20209MPR48_1D_1H_HCN.txt	Text	1	2010-03-11 13:32:35	Download

Figure 15 File Upload

4. Admin

The “Admin” menu holds “Mange Users” sub-menu and is only visible to Administrators. This menu is not available on the test website. This webpage consists of two parts, “Add Users” form and a display of current users’ information. The form lets the administrators assign the new user to a specific group, grant access to specific studies, and allow access to only the test website. The added user’s information will immediately be displayed in the current users’ information. The current users’ information can be edited or deleted by using the “Edit” button. More advanced algorithms are planned to manage the users’ accounts more effectively. Figure 16 shows a screenshot of “Admin” page.

Admin - Manage Users

You are logged in as tokac006

Add Users

¹ User ID*: <input type="text"/>	² Password*: <input type="password"/>	Re-enter Password*: <input type="password"/>
³ First Name*: <input type="text"/>	³ Last Name*: <input type="text"/>	⁴ Middle Initial*: <input type="text"/>
Email address*: <input type="text"/>	Re-Enter Email address*: <input type="text"/>	Group assigned*: <input type="text" value="Select"/>

⁵Test-only access: Projects assigned*: All
 MONR
 PONR
 DARPA

	User ID	Email	Name	Group Name	Acess Type	Studies Assigned	Account Creation Date	Last Login Time
Edit	tokac006	tokac006 @umn.edu	Priyaranjan Tokachichu	Administrator	Full	All	2009-06-06	2010-09-24 20:15:40
Edit	groeh001	groeh001 @umn.edu	Kristine E Mulier	Administrator	Full	All	2009-06-15	2010-09-17 08:53:29
Edit	witow001	witow001 @umn.edu	Nancy Witowski	Lab User	Full	All	2009-06-18	2010-07-12 11:07:21
Edit	conwa066	conwa066 @umn.edu	Tyler M Conway	Lab User	Full	All	2009-06-22	2009-06-22 15:17:34
Edit	lusco006	lusco006 @umn.edu	Elizabeth R Lusczek	Lab User	Full	All	2009-06-22	2010-09-22 14:42:28
Edit	ejackson	ejackson @umn.edu	Eileen E Jackson	Lab User	Full	All	2009-06-26	2010-09-24 09:12:38
Edit	lexco003	lexco003 @umn.edu	Daniel R Lexcen	Lab User	Full	All	2009-06-29	2010-09-02 13:33:41
Edit	nguyo833	nguyo833 @umn.edu	Andrew H Nguyen	Lab User	Full	All	2009-06-29	2009-11-02 11:09:26
Edit	beilm001	beilm001 @umn.edu	Gregory J Beilman	Administrator	Full	All	2009-06-30	2009-07-07 16:52:02
Edit	chipm001	chipm001 @umn.edu	Jeffrey G Chipman	Administrator	Full	All	2009-06-30	None
Edit	gros055	gros055 @umn.edu	Joseph J	Lab User	Full	All	2009-06-30	2010-09-20

Figure 16 User Account Administration

Data Extraction

Data from the Excel files created before the development of project management website and, text files generated under varying circumstances should be extracted and uploaded to the database to make them readily available for data query and export. Python programming language was used for this purpose. “xlrd” Python module was used in reading excel files and “MySQLdb” module was used in writing data to the database. Data extraction programs for text files were written based on the data arrangements in those files.

1. TEG data extraction

It is one of the more organized dataset. The TEG raw data text file consists of all the subjects’ data stored in it. Previously the laboratory users used to check the text file carefully to get the right subject’s data and input it into an Excel file. An automated data extraction was planned to avoid the time spent in performing this task.

The TEG raw data text file is a well-organized tab delimited file. A Python program was created to manage the data extraction from this file when users upload this file over the website. This Python program reads the file, extracts only the data related to particular study based on the study field, extracts only the data related to a particular subject, creates and stores a text file with that data on the file server, and uploads that data to the database. Figure 17 shows screenshot of part of the TEG raw data file and Figure 18 shows screenshot of processed TEG data file for one subject from which the data is extracted to the database.

1	VERSION	PATIENTID	FULLNAME	PROCEDURENAME	GENDER	BIRTHDATE	AGE	WEIGHT	SAMPLEDESCRIPTION	SITEID	OPERATORID	SAMPLETYPE	CHANNEL	ONTIME	OFFTIME	SP(min)	R(min)	K(min)	Angle(deg)	N
2																				
3	4.2.101	DARPA021109	DARPA021109	DARPA021109			FR 8		Temporary Operator	K	2	2/12/2009 8:27:19 AM	2/12/2009 9:12:14 AM	4.1	4.6	1.2	79.7	68.1	19.2	10.67
4																				
5	4.2.101	DARPA021109	DARPA021109	DARPA021109			FR 24		Temporary Operator	K	1	2/19/2009 10:59:29 AM	2/19/2009 11:21:04 AM	6.8	7.3	1.3	79.7	73.2	16.8	13.67
6																				
7	4.2.101	DARPA021109	DARPA021109	DARPA021109			FR 20		Temporary Operator	K	2	2/12/2009 9:49:46 AM	2/12/2009 10:14:06 AM	4.4	5.0	1.2	74.0	72.4	18.1	13.12
8																				
9	4.2.101	DARPA021109	DARPA021109	DARPA021109			FR 16		Temporary Operator	K	1	2/12/2009 9:13:37 AM	2/12/2009 9:50:32 AM	5.2	5.8	1.1	75.0	72.2	15.0	12.96
10																				
11	4.2.101	DARPA021109	DARPA021109	DARPA021109			Baseline		Temporary Operator	K	1	2/11/2009 12:22:02 PM	2/11/2009 12:59:47 PM	6.3	9.2	10.0		21.7	27.3	30.4
12																				
13	4.2.101	DARPA021109	DARPA021109	DARPA021109			Shock 60		Temporary Operator	K	2	2/11/2009 1:41:29 PM	2/11/2009 2:39:59 PM	8.4	11.2	10.2		21.2	37.4	4
14																				
15	4.2.101	DARPA021109	DARPA021109	DARPA021109			FR 4		Temporary Operator	K	1	2/12/2009 8:26:36 AM	2/12/2009 9:12:21 AM	3.4	3.8	1.6	69.0	62.8	19.3	8.456
16																				
17	4.2.101	Biosyn072507	Biosyn072507Control	Control			2 Hours PR		Temporary Operator	K	2	7/26/2007 1:09:18 PM	7/26/2007 1:38:03 PM	1.8	2.2	1.2	74.7	70.8	17.5	
18																				
19	4.2.101	Biosyn072507	Biosyn072507Control	Control			8 Hours PR		Temporary Operator	K	2	7/26/2007 3:03:22 PM	7/26/2007 3:44:22 PM	2.7	3.3	1.2	74.3	77.5	20.3	
20																				
21	4.2.101	Biosyn072507	Biosyn072507Control	Control			Baseline		Temporary Operator	K	1	7/26/2007 11:40:02 AM	7/26/2007 12:02:47 PM	2.9	3.3	1.0	76.7	74.9	18.2	
22																				
23	4.2.101	Biosyn072507	Biosyn072507Control	Control			1 hour PR		Temporary Operator	K	1	7/26/2007 1:09:17 PM	7/26/2007 1:37:57 PM	2.4	2.8	1.3	72.9	72.9	22.5	
24																				
25	4.2.101	Biosyn072507	Biosyn072507Control	Control			6 Hours PR		Temporary Operator	K	1	7/26/2007 3:03:21 PM	7/26/2007 3:44:11 PM	2.8	3.3	1.1	75.4	76.0	20.4	
26																				
27	4.2.101	Biosyn072507	Biosyn072507Control	Control			Shock 30		Temporary Operator	K	2	7/26/2007 11:40:03 AM	7/26/2007 12:02:38 PM	3.0	3.3	1.1	76.2	73.8	16.7	
28																				
29	4.2.101	Biosyn072507	Biosyn072507Control	Control			Shock 60		Temporary Operator	K	1	7/26/2007 12:34:24 PM	7/26/2007 12:56:54 PM	3.1	3.4	1.1	76.0	74.5	17.4	
30																				
31	4.2.101	Biosyn072507	Biosyn072507Control	Control			3 Hours PR		Temporary Operator	K	1	7/26/2007 1:49:24 PM	7/26/2007 2:24:04 PM	2.8	3.1	1.0	76.5	71.6	17.2	
32																				
33	4.2.101	Biosyn072507	Biosyn072507Control	Control			4 Hours PR		Temporary Operator	K	2	7/26/2007 1:49:28 PM	7/26/2007 2:28:28 PM	2.4	2.8	1.2	72.5	67.3	19.2	
34																				
35	4.2.101	Biosyn072507	Biosyn072507Control	Control			Shock 90		Temporary Operator	K	2	7/26/2007 12:34:25 PM	7/26/2007 12:59:20 PM	3.6	4.2	1.3	71.6	-74.6	-23.2	
36																				
37	4.2.101	DARPA061509	DARPA061509	DARPA061509			FR 16		Temporary Operator	K	1	6/16/2009 11:46:33 AM	6/16/2009 12:08:13 PM	2.8	3.0	0.8	81.1	77.9	13.6	17,614.8
38																				
39	4.2.101	DARPA061509	DARPA061509	DARPA061509			FR 20		Temporary Operator	K	1	6/16/2009 1:26:11 PM	6/16/2009 2:03:46 PM	2.2	2.4	0.8	81.2	78.2	14.6	17,930.8
40																				
41	4.2.101	DARPA061509	DARPA061509	DARPA061509			Baseline		Temporary Operator	K	1	6/15/2009 4:24:35 PM	6/15/2009 5:09:25 PM	4.9	5.4	1.0	76.3	73.4	17.5	13,79
42																				
43	4.2.101	DARPA061509	DARPA061509	DARPA061509			FR 8		Temporary Operator	K	2	6/16/2009 11:22:52 AM	6/16/2009 11:40:02 AM	0.3	2.0	0.8	74.1	86.1	4.4	6,394.0 127.9

Figure 17 TEG raw data file

1	PATIENTID	SAMPLEDESCRIPTION	R(min)	K(min)	Angle(deg)	MA(mm)
2	020209	11	4.5	1.0	76.2	76.4
3	020209	2	7.7	3.2	44.0	42.8
4	020209	7	2.4	0.9	77.9	71.8
5	020209	24	9.8	1.8	67.3	78.4
6	020209	19	5.0	0.8	79.1	77.6
7	020209	23	6.8	1.1	75.7	77.0
8	020209	25	-24.3			
9	020209	1	11.6	7.3	27.1	40.6
10	020209	5	3.1	2.2	62.2	50.5

Figure 18 TEG processed data file

2. Binning data extraction

Binning raw data text files are another well-organized tab delimited text files. While the data extraction from these files was being planned, researchers did not finalize the number of bins and their values they are going to use for a variety of tissue samples. So, a dynamic table creation algorithm was written so that when a researcher uploads the first file, Python program reads the file and creates a table with as many columns as the

number of bins and, the values are taken from the text file. Figure 19 shows a screenshot of Binning raw data file from which data is extracted to the database.

File	9.98000	9.94000	9.90000	9.86000	9.82000	9.78000	9.74000	9.70000	9.66000	9.62000	9.58000	9.54000	9.50000	9.46000
M020209MB_1D_1H_HCN.cnx	0.000078768	0.000079646	0.00019019	0.00018706	0.00027301	0.00025361	0.00025972	0.00028833						

Figure 19 Binning raw data file

3. HNMR and PNMR data extraction

Similar to Binning files, HNMR and PNMR files are examples of well-organized tab delimited text files. But, these files contain utf-8 characters which needed the database tables to be utf-8 character set and the data extraction session needs to be initiated with utf-8 character set to force the characters to appear the same way when they are downloaded. The database tables for holding HNMR and PNMR data were created dynamically once the first file was uploaded by a researcher over the website. Python program reads the file and creates a table with the compound names as column names. Different tissue samples have different compound sets and so are the tables created for them. Figure 20 and Figure 21 show HNMR and PNMR raw data files respectively and all the HNMR and PNMR files are similar to the respective formats. Data is extracted from these files and stored in the database.

1	Concentration Units mM
2	Export Date Thu Apr 09 14:49:16 CDT 2009
3	m032009b_x1.cnx
4	1-Methylnicotinamide 0.0246
5	2-Methylglutarate 0.0684
6	2-Oxoglutarate 0.1173
7	3-Hydroxyisovalerate 0.0712
8	4-Hydroxybenzoate 0.1029
9	Acetate 0.0949
10	Alanine 0.1191
11	Allantoin 2.8204
12	Ascorbate 0.3016
13	Choline 0.0421
14	Citrate 0.4892
15	Creatine 0.4610
16	Creatinine 4.4130
17	Dimethylamine 0.3921
18	Formate 0.1375
19	Glucose 0.8479
20	Glutamate 0.4046
21	Glycine 0.4609
22	Hippurate 3.1965
23	Hypoxanthine 0.0385
24	Lactate 0.2432
25	Mannitol 2.2866
26	Mannose 0.1778
27	Methanol 0.0850
28	N,N-Dimethylglycine 0.1514
29	Niacinamide 0.0161
30	Phenylacetyl-glycine 1.4220
31	Pyruvate 0.0118
32	Quinolate 0.2141
33	Succinate 0.0471
34	Taurine 0.8066
35	Trigonelline 0.0907
36	Trimethylamine 0.0110
37	Trimethylamine N-oxide 0.0810
38	Tryptophan 0.0700
39	Tyrosine 0.4854

Figure 20 HNMR raw data file

```

1
2 M081009MB
3
4 6002
5
6 BB 5mm
7 1D P
8 148.1 mg
9 150nmoles DSS
10 150nmoles TMP
11 pH 7.389
12
13 9/10/09
14
15
16 region      start (ppm)    end            integral
17 1           2.41908       2.33995       -0.0343154
18 2           1.95616       1.89285       -0.0116893
19 3           1.84537       1.76228       0.0613039
20 4           1.6515        1.51302       0.0797541
21 5           1.43389       1.34684       0.027432
22 6           1.28353       0.504085     -0.0841101
23 7           0.488259      0.41704       0.0967083
24 8           0.0728164     -0.0537946    1
25 9           -2.14785      -2.22505     -0.0103739
26 10          -2.54832      -2.64481     0.00684201
27 11          -3.08388      -3.19485     -0.0356526
28 12          -5.60728      -5.76168     1.48599
29 13          -8.2          -8.49797     0.529845
30 14          -8.52933      -8.71752     0.0686845
31 15          -12.9675      -13.187      0.0395221
32 16          -13.3909      -13.6732     0.466566
33 17          -13.7359      -14.0496     0.0275116
34 18          -23.9609      -24.2746     0.476059
35

```

Figure 21 PNMR raw data file

4. Real-Time data extraction

a) Data File Upload to the Server

To limit unnecessary manual laboratory management a Python program was written and stored on the laptop that is used for collecting the real-time data. The machine operated with Windows XP operating system. Windows Task Scheduler was scheduled to execute a Visual Basic code every 3 min once the computer is turned on and the Visual Basic

code executes the Python program on the background without causing any interference to the programs being executed on the computer or, to the user performing specific tasks on that computer. The Python program checks for the latest files (created in the last 48 hours) stored on the computer in a specific folder, and upload them to the file server by initiating a WinSCP software program. WinSCP program will execute a Python program over the file server after finishing the file upload. The Python program on the file server is meant to extract data from the uploaded files and store them in a MySQL database. As the Windows Task Scheduler initiates the Python program after 3min again, it will only read the files from the latest rows of the files after previous file upload. This reduces the stress over the computer collecting the data.

b) Data Extraction

As mentioned earlier, the files were named after the machines collecting the data and are called S-file, H-file and G-file for convenience purpose. To clean and format the data it is important to visualize how the output should look like. Figure 22 to Figure 26 below show screenshots of the output files generated in the process flow explained below. S-file is used to show the process flow changes.

Data cleaning and extraction in these files involves a series of tasks including but not limited to the following

- Determining the Subject ID based on the file name

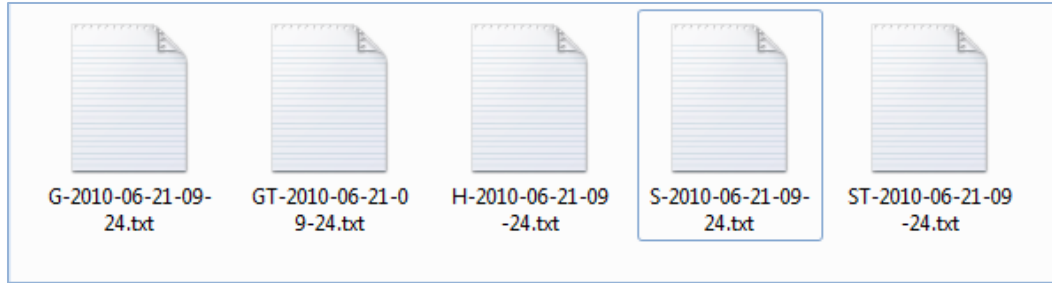


Figure 22 Real-time data files

- Removing rows that are without data or with unwanted data

S-2010-06-21-09-24.txt

1	08:32:00	0/ 0	0	1/ 1	0
2	08:32:05	0/ 0	0	1/ 1	0
3	08:32:10	0/ 0	0	1/ 1	0
4	08:32:15	0/ 0	0	1/ 1	0
5	08:32:20	0/ 0	0	1/ 1	1
6	08:32:25	0/ 0	0	1/ 1	1
7	08:32:30	0/ 0	0	1/ 1	1
8	08:32:35	1/ 0	0	1/ 1	1
9	08:32:40	0/ 0	0	1/ 1	1
10	08:32:45	0/ 0	0	1/ 1	1
11	08:32:50	1/ 0	0	1/ 1	1
12	08:32:55	0/ 0	0	0/ 0	0
13	08:33:00	0/ 0	0	1/ 1	1
14	08:33:05	1/ 0	0	1/ 1	1
15	08:33:10	0/ 0	0	1/ 1	1
16	08:33:15	0/ 0	0	1/ 1	1
17	08:33:20	1/ 0	0	1/ 1	1
18	08:33:25	0/ 0	0	1/ 1	1
19	FF				
20					
21					
22					
23					
24					
25					
26	08:33:30	1/ 0	0	1/ 1	1
27	08:33:35	1/ 0	0	1/ 1	0

SPACELABS MEDICAL PATIENT DATA LOGGER
 Bed # SL001 DATE 21 JUN 2010

TIME	ART mmHg SYS/DIA MEAN	PA mmHg SYS/DIA MEAN
08:33:30	1/ 0 0	1/ 1 1
08:33:35	1/ 0 0	1/ 1 0

Figure 23 Real-time raw data file

firststage.bt							
1	08:32:00	0/	0	0	1/	1	0
2	08:32:05	0/	0	0	1/	1	0
3	08:32:10	0/	0	0	1/	1	0
4	08:32:15	0/	0	0	1/	1	0
5	08:32:20	0/	0	0	1/	1	1
6	08:32:25	0/	0	0	1/	1	1
7	08:32:30	0/	0	0	1/	1	1
8	08:32:35	1/	0	0	1/	1	1
9	08:32:40	0/	0	0	1/	1	1
10	08:32:45	0/	0	0	1/	1	1
11	08:32:50	1/	0	0	1/	1	1
12	08:32:55	0/	0	0	0/	0	0
13	08:33:00	0/	0	0	1/	1	1
14	08:33:05	1/	0	0	1/	1	1
15	08:33:10	0/	0	0	1/	1	1
16	08:33:15	0/	0	0	1/	1	1
17	08:33:20	1/	0	0	1/	1	1
18	08:33:25	0/	0	0	1/	1	1
19	08:33:30	1/	0	0	1/	1	1
20	08:33:35	1/	0	0	1/	1	0
21	08:33:40	0/	0	0	1/	1	0
22	08:33:45	0/	0	0	1/	1	1
23	08:33:50	1/	0	0	1/	1	1
24	08:33:55	1/	0	0	1/	1	1
25	08:34:00	0/	0	0	1/	1	1
26	08:34:05	0/	0	0	1/	1	1
27	08:34:10	0/	0	0	1/	1	1
28	08:34:15	0/	-1	0	1/	1	1
29	08:34:20	0/	-1	0	1/	1	1
30	08:34:25	0/	0	0	1/	1	1

Figure 24 Real-time data with removed blank lines and unwanted data lines

- Removing irrelevant characters

secondstage.txt							
1	08:32:00	0	0	0	1	1	0
2	08:32:05	0	0	0	1	1	0
3	08:32:10	0	0	0	1	1	0
4	08:32:15	0	0	0	1	1	0
5	08:32:20	0	0	0	1	1	1
6	08:32:25	0	0	0	1	1	1
7	08:32:30	0	0	0	1	1	1
8	08:32:35	1	0	0	1	1	1
9	08:32:40	0	0	0	1	1	1
10	08:32:45	0	0	0	1	1	1
11	08:32:50	1	0	0	1	1	1
12	08:32:55	0	0	0	0	0	0
13	08:33:00	0	0	0	1	1	1
14	08:33:05	1	0	0	1	1	1
15	08:33:10	0	0	0	1	1	1
16	08:33:15	0	0	0	1	1	1
17	08:33:20	1	0	0	1	1	1
18	08:33:25	0	0	0	1	1	1
19	08:33:30	1	0	0	1	1	1
20	08:33:35	1	0	0	1	1	0
21	08:33:40	0	0	0	1	1	0
22	08:33:45	0	0	0	1	1	1
23	08:33:50	1	0	0	1	1	1
24	08:33:55	1	0	0	1	1	1
25	08:34:00	0	0	0	1	1	1
26	08:34:05	0	0	0	1	1	1
27	08:34:10	0	0	0	1	1	1
28	08:34:15	0	-1	0	1	1	1

Figure 25 Real-time data after removing irrelevant characters

- Separating and grouping the data of different variables

Figure 25 shows only arterial pressure and PA pressure. As the data file is scrolled down, more columns appear that are separated to arterial pressure, PA pressure, cardiac output and SpO2 files.

- Removing duplicate data points

This is a precaution taken even though there is no solid evidence of duplicate records.

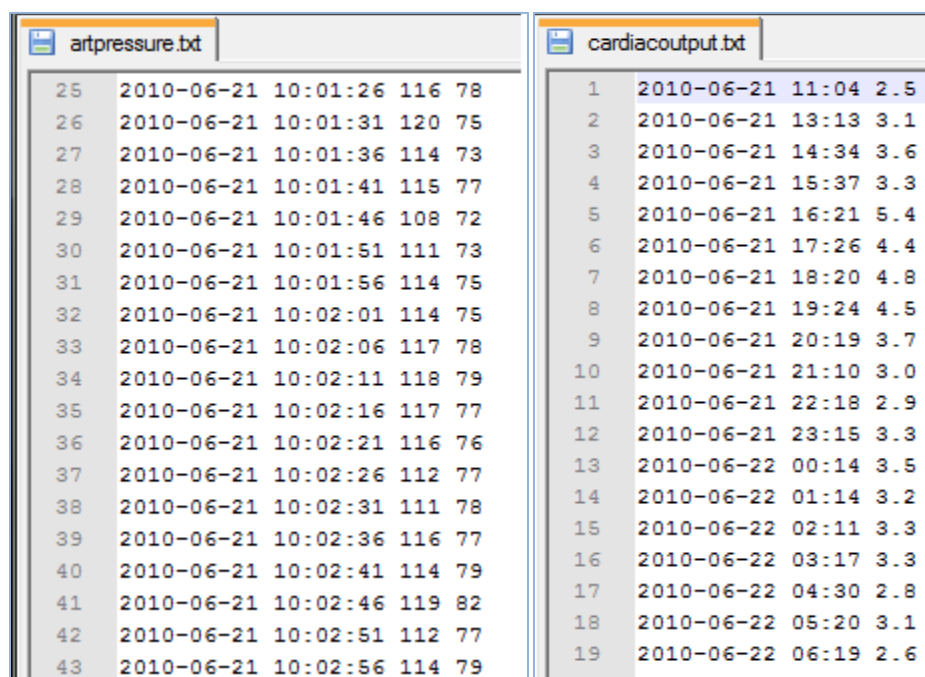
- Removing data that is physiologically not possible

Values like zero or one are not in physiological range of arterial systolic or diastolic pressure.

- Accounting for unexpected errors

If the file is suddenly out of regular format, the data extraction process stops. Accounting for unexpected errors acknowledges the error, stores it and continues to process the rest of the file.

- Accounting for time differences between machines and standardizing for laptop time (synchronized with internet time). All the machines collecting the data have a pre-set time and have no way of synchronizing it with the local standard time. Some machines account for day light saving and some don't. This makes these machines show a difference from few minutes to more than an hour from the internet standard time of the laptop where the data is stored. This time difference is calculated by noting the time in the raw file and the computer when the data collection starts and it is used to modify the time points printed in raw files.



Line	File	Timestamp	Systolic Pressure	Diastolic Pressure	Cardiac Output
25	artpressure.txt	2010-06-21 10:01:26	116	78	
26	artpressure.txt	2010-06-21 10:01:31	120	75	
27	artpressure.txt	2010-06-21 10:01:36	114	73	
28	artpressure.txt	2010-06-21 10:01:41	115	77	
29	artpressure.txt	2010-06-21 10:01:46	108	72	
30	artpressure.txt	2010-06-21 10:01:51	111	73	
31	artpressure.txt	2010-06-21 10:01:56	114	75	
32	artpressure.txt	2010-06-21 10:02:01	114	75	
33	artpressure.txt	2010-06-21 10:02:06	117	78	
34	artpressure.txt	2010-06-21 10:02:11	118	79	
35	artpressure.txt	2010-06-21 10:02:16	117	77	
36	artpressure.txt	2010-06-21 10:02:21	116	76	
37	artpressure.txt	2010-06-21 10:02:26	112	77	
38	artpressure.txt	2010-06-21 10:02:31	111	78	
39	artpressure.txt	2010-06-21 10:02:36	116	77	
40	artpressure.txt	2010-06-21 10:02:41	114	79	
41	artpressure.txt	2010-06-21 10:02:46	119	82	
42	artpressure.txt	2010-06-21 10:02:51	112	77	
43	artpressure.txt	2010-06-21 10:02:56	114	79	
1	cardiacoutput.txt	2010-06-21 11:04			2.5
2	cardiacoutput.txt	2010-06-21 13:13			3.1
3	cardiacoutput.txt	2010-06-21 14:34			3.6
4	cardiacoutput.txt	2010-06-21 15:37			3.3
5	cardiacoutput.txt	2010-06-21 16:21			5.4
6	cardiacoutput.txt	2010-06-21 17:26			4.4
7	cardiacoutput.txt	2010-06-21 18:20			4.8
8	cardiacoutput.txt	2010-06-21 19:24			4.5
9	cardiacoutput.txt	2010-06-21 20:19			3.7
10	cardiacoutput.txt	2010-06-21 21:10			3.0
11	cardiacoutput.txt	2010-06-21 22:18			2.9
12	cardiacoutput.txt	2010-06-21 23:15			3.3
13	cardiacoutput.txt	2010-06-22 00:14			3.5
14	cardiacoutput.txt	2010-06-22 01:14			3.2
15	cardiacoutput.txt	2010-06-22 02:11			3.3
16	cardiacoutput.txt	2010-06-22 03:17			3.3
17	cardiacoutput.txt	2010-06-22 04:30			2.8
18	cardiacoutput.txt	2010-06-22 05:20			3.1
19	cardiacoutput.txt	2010-06-22 06:19			2.6

	2010-06-21 09:57:00	28	13
1	2010-06-21 09:57:00	28	13
2	2010-06-21 09:57:10	30	30
3	2010-06-21 09:57:15	29	29
4	2010-06-21 09:59:35	32	7
5	2010-06-21 09:59:55	22	4
6	2010-06-21 10:00:00	20	9
7	2010-06-21 10:00:05	12	5
8	2010-06-21 10:00:21	13	6
9	2010-06-21 10:00:26	15	3
10	2010-06-21 10:00:36	10	4
11	2010-06-21 10:00:41	18	9
12	2010-06-21 10:00:46	21	12
13	2010-06-21 10:00:51	24	13
14	2010-06-21 10:00:56	21	12
15	2010-06-21 10:01:01	23	11
16	2010-06-21 10:01:06	23	15
17	2010-06-21 10:01:11	24	13
18	2010-06-21 10:01:16	24	12

Figure 26 Real-time data fully processed

The parent raw-data file was missing SpO2 data

- Uploading data to MySQL database

This is done by Python language module for MySQL called MySQLdb.

- Backing up all the removed data files, and files created at every stage of data processing for auditing purpose (Figure 27)

This is done automatically with Python code.

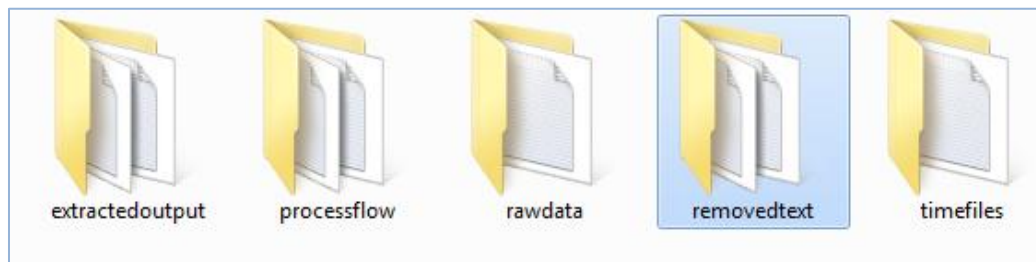


Figure 27 Real-time data final directories

- Deleting temporary files created during the process

- Adjusting the real-time data event times to match the event times entered manually by the laboratory staff in order to be able to use real-time data in data analysis along with manual data

Manually entered data is accurate up to minutes but the real-time data is accurate up to seconds. This will cause problems when manually entered data for example collected at a time point called Shock 45 is noted as 11:45, the real-time data point could be 11:45:23. To note this time point as Shock 45 a Python program was written that will compare the data points of real-time data to manually entered data and insert the time points that match very closely in the real-time data table. This enables the comparison of real-time data to manually entered data.

Methods Followed to Improve Data Analysis Procedure

The usual methods followed by researchers before the development of this research management system is to rearrange the data in an Excel spreadsheet or in an SPSS software application in order to perform specific statistical analyses or produce graphs. The principal biostatistician of the study was provided with an Excel spreadsheet with cumulative data. The biostatistician used SAS software application to rearrange the data and perform statistical analyses. These methods were very ineffective due to duplication of the data several times and time spent in obtaining desired results. As the “Advanced Export” mechanism over the website allows the users to export the desired data into an Excel spreadsheet, it reduced data manipulation and rearrangement by researchers. But, the biostatistician is still provided with a cumulative data Excel spreadsheet produced by “Advanced Export”. An alternative approach to reduce the data duplication, manipulation

and time spent on those processes by the biostatistician would be to use SAS SQL procedure.

SAS SQL procedure which is used as “PROC SQL” in SAS programs is able to execute SQL queries along with some other pre-defined procedures. SAS is able to connect to a MySQL database and retrieve data into a temporary dataset or a permanent dataset as per the user request, and perform statistical analyses. To perform this task there is a need to know SQL and the specific database architecture which generally is not plausible for a biostatistician. Writing complex SQL queries to perform statistical analyses will defeat the whole purpose of the research management system. As the research management system already has the ability to store SQL queries, this feature can be used to submit those queries to SAS server using Python programming language. This procedure was tested and planned for implementation. After this implementation users will be able to use stored SQL queries and perform statistics over the website with options to view and store the results. Advanced users will be able to download SQL queries to further modify and use them over SAS server hosted by MSI.

A Python wrapper program³¹ is used to write dynamic SAS programs, and is considerably more effective than the complex Macros used in SAS. SAS Macro language is considered a separate language by many programmers due to its complexity and dissimilarity compared to general SAS programs. Due to this, python wrapper is a better choice than SAS macros in web-based data analysis. Any object oriented programming language can perform similar tasks as Python does in this scenario. R³² software package

³¹ A Python program that executes SAS commands as part of its code

³² <http://www.r-project.org/>

would be used in future in place of SAS due to its high quality graphics and better programming features.

Development and Production Machine Maintenance

Usual programming environment consists of separate development and production machines. Development machine is where the programmer tests the website by making changes and updates the production machine with the finalized files. While the development continues at the development machine, production machine will only be updated intermittently when a stable and complete webpage is ready. Having the development server and production server on the same machine or using the same server for both development and production involves several security risks especially in a shared resource environment arising from incomplete coding. A variety of automated and manual data backups were implemented to maintain safety and integrity of the website.

Automated Data Backup

Automated file backup of individual user accounts is performed by MSI at a frequency of once in a week and MySQL database backups once in a month. In a matter of system failure, programmers can go back to previous backed up files and restore the website to a previous stage and MySQL database can be restored to a previous image with the help of MSI. MySQL commands that are directly typed into the MSI server bash shell are stored in a log file. This is useful for restoring the database or for auditing purposes. But, MySQL commands sent from Python or other programming languages are not written to the log file which makes it difficult for restoring the database and auditing purposes. A

cron job³³ was created to automatically backup MySQL database once every day at 3:10 AM. A proposal is in place to change it to a weekly backup to accommodate huge real-time datasets. The cron job will also delete the backups older than 30 days. The folder containing backed up database files is in synchrony with a database backup folder on the development machine which allows backups on two different machines minimizing the data loss due to system failure. A Python program present on those machines will inform the programmer about the successful transfer by sending out an email. Python program on the development machine attaches the download log of WinSCP while sending the email. This will help the programmer find the cause of the error in case of download failures.

Managing Web Development

Due to operating system differences an exact replication of production system paths was not possible on the development system. This required changes in several html, JavaScript, CSS and Python programming files before uploading to the server. There was also a need to replicate the website into “test” and “official” websites which also need changes in the code. File permissions need to be managed effectively on the production system each time files are uploaded. Manually performing these tasks is time consuming, and the process is prone to errors. Well planned and organized automation of these processes will improve project efficiency and will reduce errors. A Python application, with a one-click web interface, was developed for processes like backup, text modification, upload, and download of files. This application also carries out minor security measures preventing unauthorized access to the source files.

³³ <http://www.scrounge.org/linux/cron.html>

1. Source Code Backup

In case of both “Backup and Upload” and “Backup and Download” scenarios, the first click starts the file backup on the development machine. Various directories and the files in the directories are copied into a backup location into a directory named with date of the backup that was created by Python command `os.makedirs([directoryname])`. The directory structure is maintained while backing up the files.

2. Source Code Modification and Feedback

Source code modification is the next step in the “Backup and Upload” scenario. This is performed by multiple Ajax responses to mimic command line output. A JavaScript request is made to the server to change the text of the latest programming files and copy them to respective folders in order to upload the files to the server. As the feedback to the user is only possible when all the processes are over, request is made to modify one file at a time. At the end of the file modification server sends out a callback JavaScript code to the browser that shows that the file has been modified and also executes the next Ajax request. This process continues until all the files have been modified. Different types of files are represented by different colors that are chosen randomly to differentiate them from each other. While modifying the files, windows end of line characters (`\r\n`) were modified to (`\n`) so that Python code will not have problems on the server machine which is Linux based.

3. Python Code Conversion to Binaries

As the production system is a shared environment, it is important to hide the source code of Python files to avoid accidental or intentional manipulation. So, server side Python

files were converted into binary files and were stored in the Python installation directory for Python modules. These binary files are called as modules by Python files in the web directory which is a less secure location due to certain mandatory file permissions.

4. Compression and Password Protection of Backed Up Files

Even though reverse compilation of the binary files is possible, it is tedious and time consuming. So, there is a need for the Python source files to be stored and backed up over the production machine in order to be able to recover them when needed. Even though the backup location is more secure than web folders, it is better to add another layer of security to prevent accidental or intentional hacking. Open-source software called 7-Zip³⁴ was used for this purpose. The reason for the choice other than being open-source and free software is compatibility with a variety of operating systems including Windows and UNIX and extensive command line support. After the completing of all the processes mentioned in previous sections, python code calls 7-Zip software to compress the required folders and encrypt them with a password. In order to be able to handle these files over the server if required, 7-Zip software was installed in the local account provided by the MSI.

5. Automated File Exchange

A series of WinSCP commands were written to synchronize files and folders between development machine and production machine. These commands were stored in external files that are called by Python programs at the end of “Backup and Upload” or “Backup and Download” processes. In case of “Backup and Upload” process, files are uploaded to

³⁴ About 7-Zip

the respective web folders, binary files are uploaded to the Python program installation folders, compressed files uploaded to the backup location. Files with the same name are replaced by the uploaded files during this process. In case of “Backup and Download” process, backed up compressed files are downloaded to a directory, decompressed and the files and folders are copied to respective folders as programmed.

This automated process enabled the freedom of project development from multiple machines by synchronizing them with each other. Figure 28 shows screenshot of web programming code backup, code processing, and upload of the programming files to the production server along with password protected zipped files for backup.

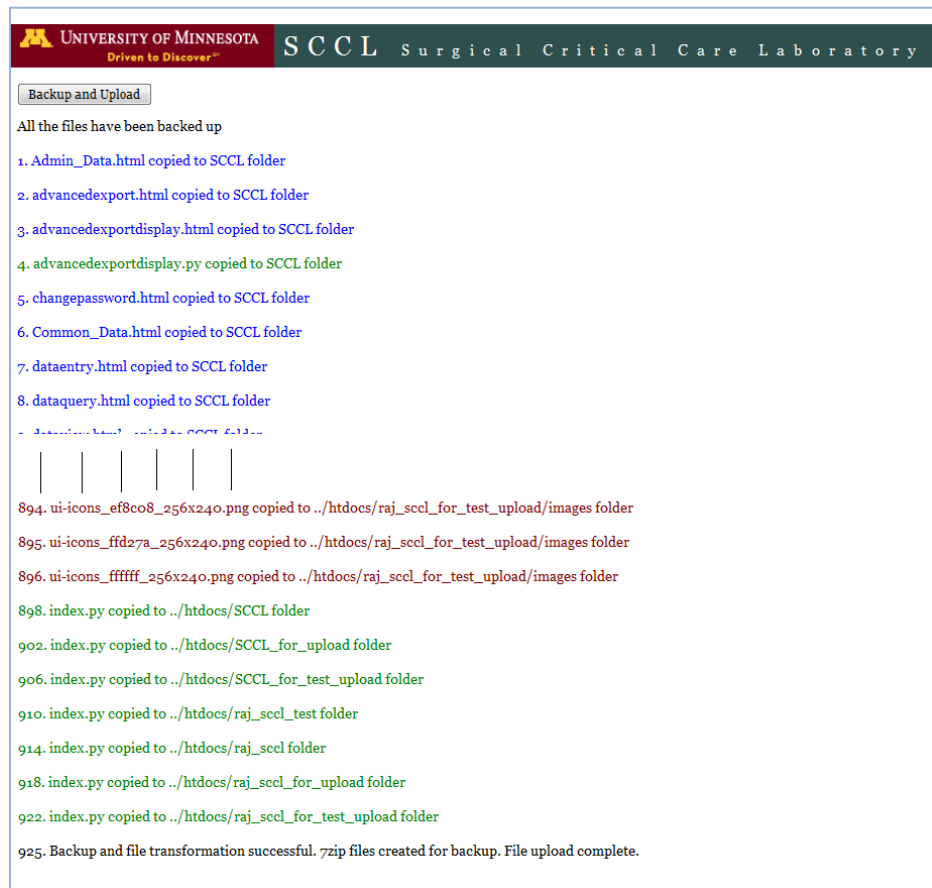


Figure 28 File backup, automated code processing and upload

Final Product Functionality Validation

- A flat file database has been replaced with a relational database making it easier to modify and query data. This is a frequent challenge in systematic data management of research data where simple flat file solutions provide a starting point for data collection which frequently becomes difficult to manage as the size of the data set grows. Systematic informatics modeling at the beginning can provide a mechanism to avoid a need for re-engineering and help with maintenance, reusability and integration of research data (Adam, 2010).
- Multiple users can access and perform data entry and export at the same time.
- Real-time data management was made possible and data is ready for analysis.
- Web-based data management made the contribution and sharing easier.
- Data export was made simpler by introducing data export solutions over the web page.
- Time spent on data cleaning and data import into flat file database was decreased tremendously by implementing automated data cleaning and extraction over the website.
- Auditing data modifications is made possible.
- Better data backup mechanisms in place.
- User account management was performed over the website.
- Automated project development synchronization reduces the risk of error and time spent on synchronization.

- **Product Evaluation Survey**

A product evaluation survey was developed and was modeled using Survey Monkey web application. The survey was then embedded onto the cSurge website and an email was sent out to all the registered users of the cSurge application (Figure 29). Out of all the registered users 12 users are identified as the users who used cSurge web application or participated in training or development of the project to some extent. The survey was made anonymous and user computers' IPs are not collected. This made the tracking of people who completed the survey difficult and was solved by providing an option to the user to submit the completion of the survey to the cSurge database along with Survey Monkey database. The cSurge database will not be able to link the user to survey but will only register whether the user has completed the survey or not. It will also register the survey completion time which allows the comparison of the survey completion time noted on the Survey Monkey database. The survey was posted on 10/10/2010 and collection of completed surveys was closed on 10/21/2010. A total of 10 completed surveys were collected out of which three members deviated from the instructions on registering the completion of surveys in cSurge database. Two of the three members were identified by contacting all the registered users and one was not accounted for so far.

The evaluation survey consists of 17 questions out of which four are general questions regarding the user overall experience and satisfaction and the rest are specific to the functionalities of cSurge software. As most of the users do not use all the features of cSurge, the entire questions specific to the functionality of cSurge were provided with an option "Not applicable". All the questions require an answer in

order to complete the survey, and were also provided with an optional comments field.

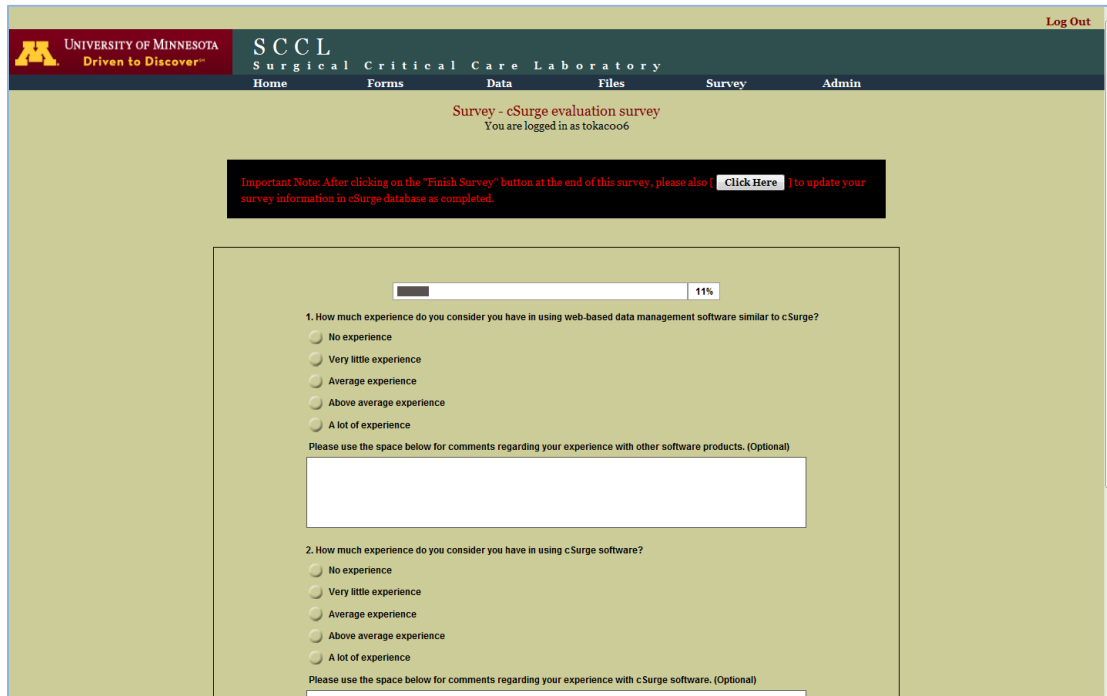


Figure 29 Survey embedded in cSurge

Evaluation questionnaire and its responses are presented in Results section.

Comparison of manual and real-time data

A paired t-test analysis was done between manually collected and real-time data of the test subjects. Real-time data is not supposed to be significantly different from manually collected data proving the accuracy of real-time data collection. SAS biostatistical software was used to achieve this task. Multiple “proc sql” commands were issued to extract data from the MySQL database, and statistics were performed over the SAS datasets created.

Results

Survey Questions and Responses

All the percentages discussed here are relative to the number of users who answered the survey. Users who answered the choice “Not applicable” are not counted towards the percentage for that particular question in the discussion of results but are counted towards percentage in the figures. Question 1 (Figure 30) was targeted at the experience of users with software similar to cSurge. 90% of the users claimed that they have no experience or very little experience with similar software.

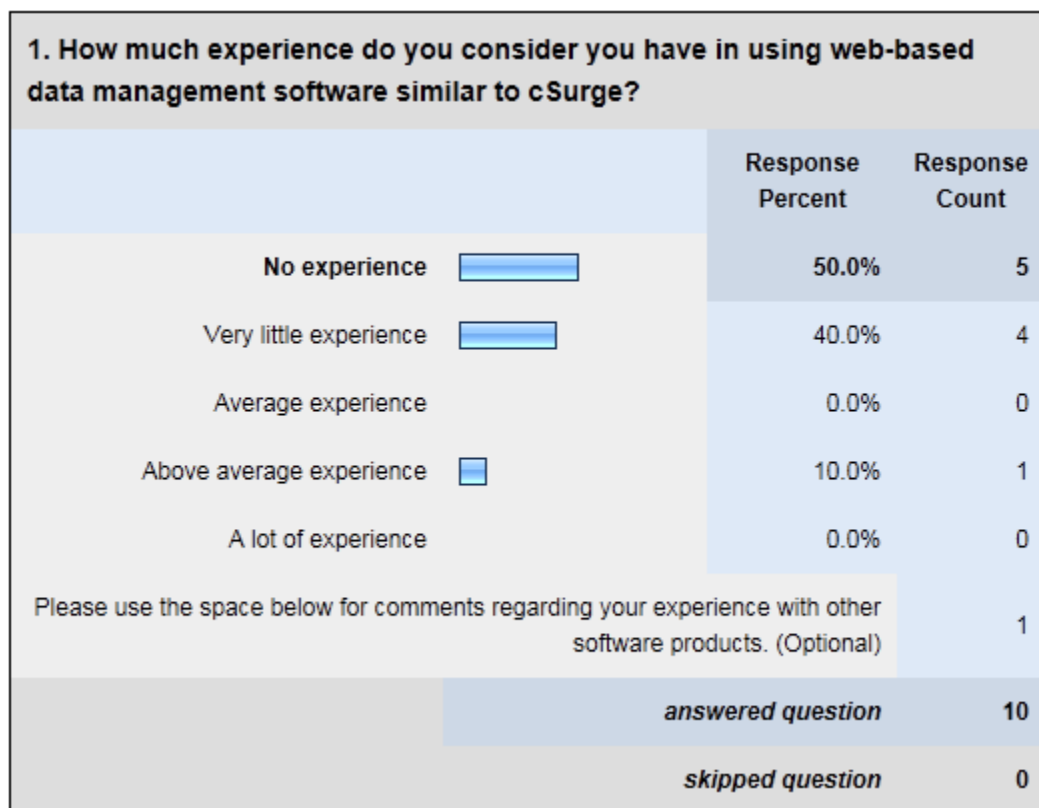


Figure 30 Question 1 responses

Question 2 (Figure 31) was targeted at user experience with cSurge software. 90% of the users have some amount of experience with cSurge software and one user responded as having no experience. The user could have been involved in training or development of cSurge software. The responses show that majority of the users have a very little experience with cSurge software that could be due to their use of only certain features of it.

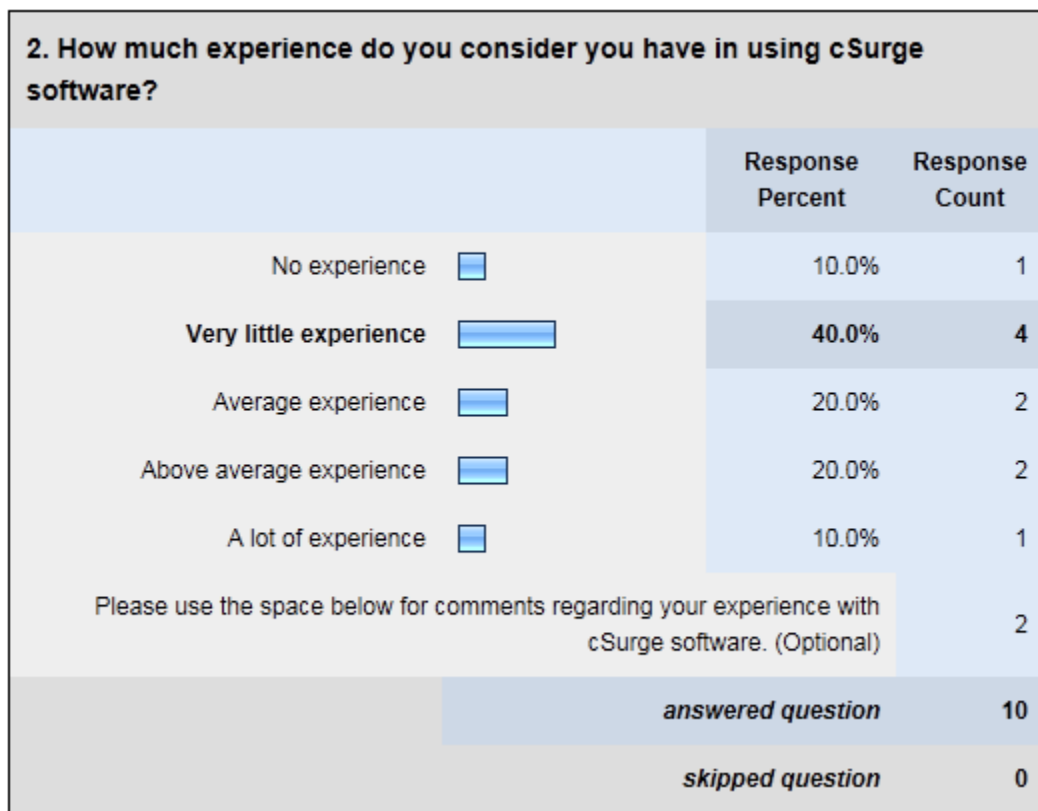


Figure 31 Question 2 responses

Question 3 (Figure 32) was targeted at the ease of navigation of the website. 70% of the respondent users considered that it was very easy to navigate in their experience. None of the users considered that it was difficult to navigate.

3. How would you rate cSurge in ease of navigation in your experience?		
	Response Percent	Response Count
Very difficult	0.0%	0
Somewhat difficult	0.0%	0
Neither easy nor difficult <input type="checkbox"/>	20.0%	2
Somewhat easy <input type="checkbox"/>	10.0%	1
Very easy <input checked="" type="checkbox"/>	70.0%	7
Not applicable	0.0%	0
Please use the space below for comments regarding the navigation in cSurge. Please include both positive and negative perceptions. (Optional)		2
	answered question	10
	skipped question	0

Figure 32 Question 3 responses

Question 4 (Figure 33) was targeted at the ease of data entry. Out of all the users that ever used data entry feature 83% considered that it was very easy.

4. How would you rate cSurge in ease of data entry in your experience?		
	Response Percent	Response Count
Very difficult	0.0%	0
Somewhat difficult	0.0%	0
Neither easy nor difficult	0.0%	0
Somewhat easy <input type="checkbox"/>	10.0%	1
Very easy <input checked="" type="checkbox"/>	50.0%	5
Not applicable <input type="checkbox"/>	40.0%	4
Please use the space below for comments regarding the data entry in cSurge. Please include both positive and negative perceptions. (Optional)		2
	<i>answered question</i>	10
	<i>skipped question</i>	0

Figure 33 Question 4 responses

Question 5 (Figure 34) was targeted at the ease of generating reports the feature which was used by most of the users. 50% users considered that it was very easy and the rest considered that it was somewhat easy.




5. How would you rate cSurge in ease of generating reports in your experience?		
	Response Percent	Response Count
Very difficult	0.0%	0
Somewhat difficult	0.0%	0
Neither easy nor difficult	0.0%	0
Somewhat easy 	40.0%	4
Very easy 	40.0%	4
Not applicable 	20.0%	2
Please use the space below for comments regarding the generation of reports in cSurge. Please include both positive and negative perceptions. (Optional)		1
	<i>answered question</i>	10
	<i>skipped question</i>	0

Figure 34 Question 5 responses

Question 6 (Figure 35) was targeted at the confidence of users in the accuracy of generated results. Mixed responses were received for this question. 75% of the respondents were very confident at the generated reports. 12.5 % are somewhat confident and 12.5% are somewhat not confident at the generated reports. This is majorly due to miscommunication of the process of report generation to the users.

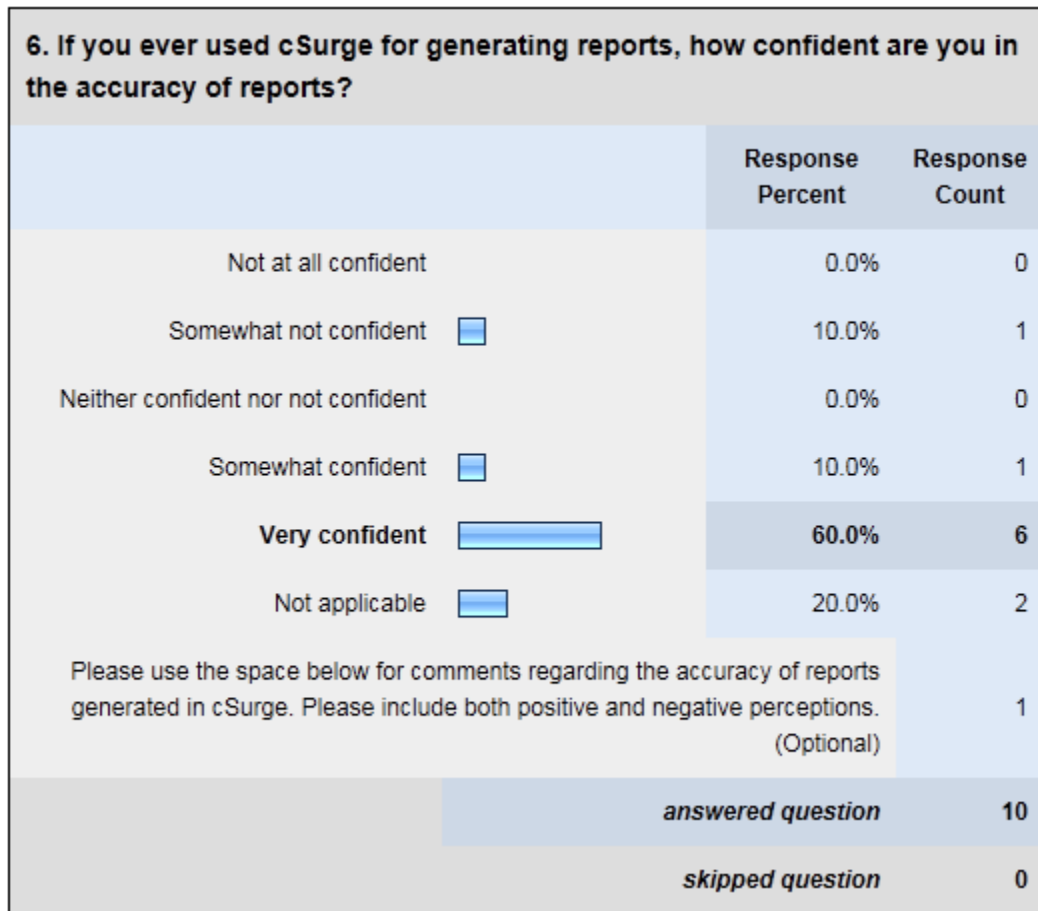


Figure 35 Question 6 responses

Question 7 (Figure 36) was targeted at the ease of file upload. All the users of this feature considered that it was very easy.


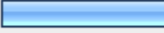
7. How would you rate cSurge in ease of file upload in your experience?		
	Response Percent	Response Count
Very difficult	0.0%	0
Somewhat difficult	0.0%	0
Neither easy nor difficult	0.0%	0
Somewhat easy	0.0%	0
Very easy 	30.0%	3
Not applicable 	70.0%	7
Please use the space below for comments regarding the file upload in cSurge. Please include both positive and negative perceptions. (Optional)		1
answered question		10
skipped question		0

Figure 36 Question 7 responses

Question 8 (Figure 37) was targeted at the usefulness of the automatic data extraction from the uploaded files. This is a follow-up question for the question 7 but surprisingly received more user input than the question 7. 80% of the users considered that this feature is very useful and one user considered that this feature is neither useful nor not useful.

8. cSurge software comes with the ability to automatically extract data from files uploaded on the file upload page. How would you rate the usefulness of this feature?		
	Response Percent	Response Count
Not at all useful	0.0%	0
Somewhat less useful	0.0%	0
Neither useful nor not useful <input type="checkbox"/>	10.0%	1
Somewhat useful	0.0%	0
Very useful <input type="checkbox"/>	40.0%	4
Not applicable <input type="checkbox"/>	50.0%	5
Please use the space below for comments regarding the data extraction from files in cSurge. Please include both positive and negative perceptions. (Optional)		1
	answered question	10
	skipped question	0

Figure 37 Question 8 responses

Question 9 (Figure 38) was targeted at the usefulness of query saving and sharing capabilities of cSurge software. 40% of the users of these features considered that they are very useful and the rest of them considered that they are somewhat useful.



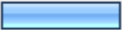
9. cSurge offers a functionality to save and share the report generating queries. If you ever used it, how would you rate the usefulness of this feature based on your experience?		
	Response Percent	Response Count
Not at all useful	0.0%	0
Somewhat not useful	0.0%	0
Neither useful nor not useful	0.0%	0
Somewhat useful 	30.0%	3
Very useful 	20.0%	2
Not applicable 	50.0%	5
Please use the space below for comments regarding saving and sharing report generating queries in cSurge. Please include both positive and negative perceptions. (Optional)		2
	answered question	10
	skipped question	0

Figure 38 Question 9 responses

Question 10 (Figure 39) was targeted at the ease of data export into a MS Excel and Comma Separated Value (CSV) formats. 62.5% of the users considered that it was easy and the rest of the users considered that it was somewhat easy.

10. How would you rate cSurge in ease of exporting data in the form of Microsoft Excel or Comma Separated Value (CSV) format in your experience?		
	Response Percent	Response Count
Very difficult	0.0%	0
Somewhat difficult	0.0%	0
Neither easy nor difficult	0.0%	0
Somewhat easy	30.0%	3
Very easy	50.0%	5
Not applicable	20.0%	2
Please use the space below for comments regarding the data export in cSurge. Please include both positive and negative perceptions. (Optional)		0
	answered question	10
	skipped question	0

Figure 39 Question 10 responses

Question 11 (Figure 40) was targeted at the user account management by the administrators. There were three administrators but it seems they have never used this feature.

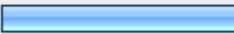
11. If your account on cSurge was assigned administrative privileges, you will have access to a page called manage users. If you have ever used it, how would you rate the ease of user account management?		
	Response Percent	Response Count
Very difficult	0.0%	0
Somewhat difficult	0.0%	0
Neither easy nor difficult	0.0%	0
Somewhat easy	0.0%	0
Very easy	0.0%	0
Not applicable 	100.0%	10
Please use the space below for comments regarding managing user accounts in cSurge. Please include both positive and negative perceptions. (Optional)		1
	<i>answered question</i>	10
	<i>skipped question</i>	0

Figure 40 Question 11 responses

Question 12 (Figure 41) targets the flexibility of the cSurge system for intermittent changes. 60% of the users experienced very quick changes in the software when they requested and 20% observed somewhat quick response and the rest did not consider the response as either quick or slow.

12. If you have ever needed and requested a change in the cSurge structure or functionality, can you please rate the response time?		
	Response Percent	Response Count
Very late	0.0%	0
Somewhat late	0.0%	0
Neither late nor quick <input type="checkbox"/>	10.0%	1
Somewhat quick <input type="checkbox"/>	20.0%	2
Very quick <input type="checkbox"/>	30.0%	3
Not applicable <input type="checkbox"/>	40.0%	4
Please use the space below for comments regarding the response time for making changes in cSurge. Please include both positive and negative perceptions. (Optional)		0
<i>answered question</i>		10
<i>skipped question</i>		0

Figure 41 Question 12 responses

Question 13 (Figure 42) was targeted at the extent of training needed by the users to use the features of cSurge software. 50% of the users considered the training needed was somewhat less extensive, 40% considered that the training needed was very less extensive and one user considered the training needed was neither extensive nor less extensive.


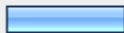

13. What is the extent of training you needed to be able to use cSurge?		
	Response Percent	Response Count
Very extensive	0.0%	0
Somewhat extensive	0.0%	0
Neither extensive nor less 	10.0%	1
Somewhat less 	50.0%	5
Very less 	40.0%	4
Not applicable	0.0%	0
Please use the space below for comments regarding the training for using cSurge. Please include both positive and negative perceptions. (Optional)		2
	answered question	10
	skipped question	0

Figure 42 Question 13 responses

Question 14 (Figure 43) was targeted at cSurge addressing data sharing and exchange issues. Only two users responded that they had issues with data sharing and exchange before cSurge and they were efficiently addressed by cSurge.

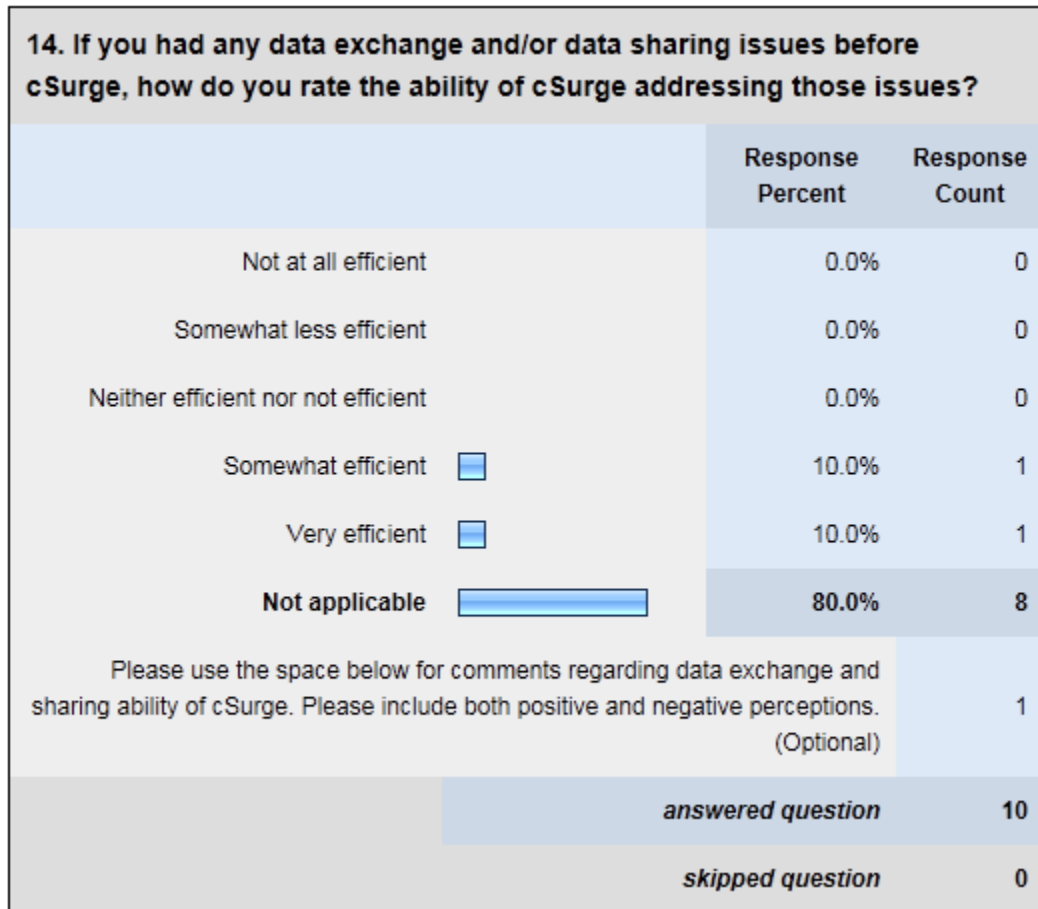


Figure 43 Question 14 responses

Question 15 (Figure 44) was targeted at the usefulness of the real-time data collection of the cSurge software if the users are ever aware of it. 50% of the respondents considered it very useful, 25% considered it somewhat useful and the rest remained neutral about the usefulness. The reason for the last 25% in doubt of the usefulness could be because the real-time data are not yet being used for statistical purposes in the SCC laboratory.





15. cSurge automatically processes the real-time data collected during experiments and uploads them to the database. If you are aware of this feature, can you please rate the usefulness of it?		
	Response Percent	Response Count
Not at all useful	0.0%	0
Somewhat less useful	0.0%	0
Neither useful nor not useful 	20.0%	2
Somewhat useful 	20.0%	2
Very useful 	40.0%	4
Not applicable 	20.0%	2
Please use the space below for comments regarding the data extraction from real-time data files in cSurge. Please include both positive and and negative perceptions. (Optional)		2
	answered question	10
	skipped question	0

Figure 44 Question 15 responses

Question 16 (Figure 45) and Question 17 (Figure 46) were targeted at the user satisfaction with the overall quality and overall usefulness of the cSurge software respectively. 80% of the users were very satisfied at the overall quality of the software. 70% of the users were very satisfied at the overall usefulness of the software. These two questions did not receive any responses that suggest that any user is not satisfied with the overall quality or usefulness of cSurge software.

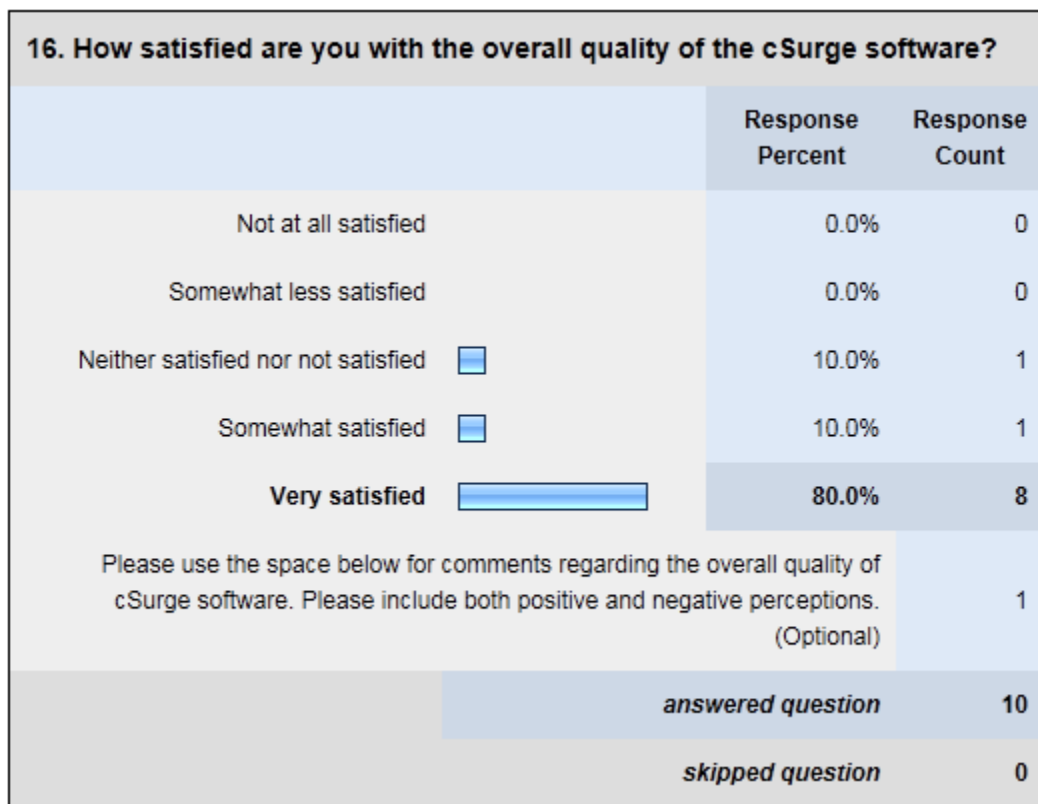


Figure 45 Question 16 responses

17. How satisfied are you with the overall usefulness of the cSurge software?		
	Response Percent	Response Count
Not at all satisfied	0.0%	0
Somewhat less satisfied	0.0%	0
Neither satisfied nor not satisfied	0.0%	0
Somewhat satisfied	30.0%	3
Very satisfied	70.0%	7
Please use the space below for comments regarding the overall usefulness of the cSurge software. Please include both positive and negative perceptions. (Optional)		0
	<i>answered question</i>	10
	<i>skipped question</i>	0

Figure 46 Question 17 responses

Results of comparison of manual and real-time data

Paired t-test statistics were performed on all the subjects in the study so far and no statistically significant difference was found between manual and real-time data. Statistical results of one of the subjects were included in Appendix C along with the SAS code.

Discussion

Systems biology research produces huge amounts of unique data which is difficult to manage without a personalized data management system. This project focuses on developing such a system for Surgical Critical Care laboratory at University of Minnesota. The project development followed a combination of a health informatics data model, and agile software development model. The project evaluation approaches followed were, an evaluation survey, and comparison of the manual and real-time data.

The survey conducted to evaluate this project showed higher percentages of user satisfaction but results of question 1 and question 2 indicate that most of the cSurge users are not experienced in cSurge or similar software. This means that their responses to rest of the questions do not compare cSurge software to any other software but with their personal experience. Not all the users were experienced in all the features of cSurge software, and if they were the results could be different.

Though the survey results are very convincing that the cSurge software prevails in all the features offered by it, it can only be applied to SCC laboratory research, and to research groups that does similar research in similar conditions as SCC laboratory. The software is used by a small group of 12 users out of whom 10 users answered the survey. Due to this small number a statistically significant conclusion cannot be made based on these results. However, the results are very promising supporting the idea that systems biology research needs homegrown applications like cSurge in order to efficiently manage and use enormous amounts of data generated.

The paired t-test between manual and real-time data did not show any statistically significant difference. This means that the real-time data is as reliable as manually collected data and can be used in place of manual data which can save huge amounts of time spent on data collection and data entry.

Based on the results cSurge, the web-based data management system developed for metabolomics research at Surgical Critical Care laboratory is successful in fulfilling the data management and integration needs. The project exemplifies the development process needed to build a home grown application using open-source and commercial software products while using the information technology infrastructure available at the educational institution. Even in the absence of such infrastructure it is possible to build the project within a reasonable expense as most of the software products are open-source, or free. Since the project is web-based, it can be incorporated into any other web-based system and at the same time can accommodate other web-based systems.

The future development of this project aims at replacing the manual data entry with real-time data, adding decision support as per the study protocol during the experiment, real-time graphical display of subject vitals over the website, web-based advanced statistical analysis tool, extensive logging of user activity, accommodating more research groups etc. This project also opens doors to web-based electronic ICU management system development using various algorithms and decision support.

Major drawback of agile development in general and this project in particular is lapse in product documentation and tutorials. This leads to difficulty in project maintenance for new programmers joining the team.

From time to time there have been efforts to build home grown applications and some of them end up becoming community driven open-source projects. This project might not become a community driven open-source project as it is personalized to the data needs of SCC laboratory but, the development process will be the same for systems biology research projects building home grown data management applications. Once the data management and data integration needs of any project are finalized, this development process can be used to build a personalized web-based data management system. This will help reduce time spent in data collection and data entry. It will also reduce the lag between data collection and analysis speeding up the research process.

Conclusions

Based on the results, it can be concluded that developing a personalized web-based data management system for systems biology projects using open-source and commercial software products yield favorable results by integrating data effectively and reducing the lag between data collection and analyses. A health informatics data model and agile software development model are useful in developing data management solutions for systems biology research.

References

- Adam, T. (2010, October). Informatics Modeling. (P. Tokachichu, Interviewer)
- Aukburg, S., Ketikidis, P., Kitz, D., Mavrides, T., & Matschinsky, B. (1989). Automation of Physiologic Data Presentation and Alarms in the Post Anesthesia Care Unit. *Annual Symposium on Computer Application in Medical Care* (pp. 580-582). American Medical Informatics Association.
- Beilman, G. (2008, November 1). Identification of Biomarkers using Metabolomics in Porcine Hemorrhagic Shock. Minneapolis, MN, United States of America.
- Berman, H. M. (2008). The Protein Data Bank: a historical perspective. *Acta Crystallographica Section A*, 64, 88-95.
- Bourne, P. E., Westbrook, J., & Berman, H. M. (2004). The Protein Data Bank and lessons in data management. *Briefings in Bioinformatics*, 5, 23-30.
- Boyle, J., Rovira, H., Cavnor, C., Burdick, D., Killcoyne, S., & Shmulevich, I. (2009). Adaptable data management for systems biology investigations. *Bmc Bioinformatics*, 10.
- Brown, P., Haas, P., Myllymaki, J., Pirahesh, H., Reinwald, B., & Sismanis, Y. (2005). Toward automated large-scale information integration and discovery. *3551*, pp. 161-180.
- Campbell, E. M., Sittig, D. F., Ash, J. S., Guappone, K. P., & Dykstra, R. H. (2006, SEP-OCT). Types of unintended consequences related to computerized provider order

entry. *JOURNAL OF THE AMERICAN MEDICAL INFORMATICS ASSOCIATION*, 13, 547-556.

Chalmel, F., & Primig, M. (2008, FEB 6). The Annotation, Mapping, Expression and Network (AMEN) suite of tools for molecular systems biology. *BMC BIOINFORMATICS*, 9.

Clayton, P. D., & van Mulligen, E. (1996). The economic motivations for clinical information systems. *Proc AMIA Annu Fall Symp*, 660-8.

Cruz-Correia, R. J., Vieira-Marques, P. M., Ferreira, A. M., Almeida, F. C., Wyatt, J. C., & Costa-Pereira, A. M. (2007, JUN 12). Reviewing the integration of patient data: how systems are evolving in practice to meet patient needs. *BMC MEDICAL INFORMATICS AND DECISION MAKING*, 7.

Delaney, D., Zibrak, J., Samore, M., & Peterson, M. (1997). Use of relational database management system by clinicians to create automated MICU progress note from existent data sources. *JOURNAL OF THE AMERICAN MEDICAL INFORMATICS ASSOCIATION*, 7-11.

Deus, H. F., Stanislaus, R., Veiga, D. F., Behrens, C., Wistuba, I., Minna, J. D., et al. (2008). A Semantic Web Management Model for Integrative Biomedical Informatics. *Plos One*, 3.

Drummond, W., Carnes, T., Coons, S., Birkett, K., Xian, J., & Roys, R. (2004, APR). CentraView (CV): An automated real-time ICU patient data integration and presentation system. *PEDIATRIC RESEARCH*, 55, 372.

- Duncan, R., Saperia, D., Dulbandzhyan, R., Shabot, M., Polaschek, J., & Jones, D. (2001). Integrated Web-based viewing and secure remote access to a clinical data repository and diverse clinical systems. *JOURNAL OF THE AMERICAN MEDICAL INFORMATICS ASSOCIATION*, 149-153.
- Fernie, A. R., Trethewey, R. N., Krotzky, A. J., & Willmitzer, L. (2004). Innovation - Metabolite profiling: from diagnostics to systems biology. *Nature Reviews Molecular Cell Biology*, 5, 763-769.
- Ficenec, D., Osborne, M., Pradines, J., Richards, D., Felciano, R., Cho, R. J., et al. (2003). Computational knowledge integration in biopharmaceutical research. *Briefings in Bioinformatics*, 4, 260-278.
- Fiehn, O. (2002, JAN). Metabolomics - the link between genotypes and phenotypes. *PLANT MOLECULAR BIOLOGY*, 48, 155-171.
- George, M. E., Mulier, K. E., & Beilman, G. J. (2010, MAR). Hypothermia Is Associated With Improved Outcomes in a Porcine Model of Hemorrhagic Shock. *JOURNAL OF TRAUMA-INJURY INFECTION AND CRITICAL CARE*, 68, 662-668.
- Goodacre, R. (2005, MAR). Metabolomics - the way forward. *METABOLOMICS*, 1, 1-2.
- Goodman, N., Rozen, S., Stein, L. D., & Smith, A. G. (1998). The LabBase system for data management in large scale biology research laboratories. *Bioinformatics*, 14, 562-574.
- Hernandez, A., Carrault, G., Mora, F., Thoraval, L., Passariello, G., & Schleich, J. (1999, OCT). Multisensor fusion for atrial and ventricular activity detection in coronary

care monitoring. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 46, 1186-1190.

Higgins, S. B., Jiang, K., Swindell, B. B., & Bernard, G. R. (1991). A Graphical ICU Workstation. *Annual Symposium on Computer Application in Medical Care* (pp. 783-787). American Medical Informatics Association.

Hoschek, W., Jaen-Martinez, J., Samar, A., Stockinger, H., & Stockinger, K. (2001). Data management in an International Data Grid project.

Kitano, H. (2002). Computational systems biology. *Nature*, 420, 206-210.

Kitano, H. (2002). Systems biology: A brief overview. *Science*, 295, 1662-1664.

Leon-Chisen, N. (2004, August 12). *hospitalconnect.com*. Retrieved October 13, 2010, from http://www.hhnmag.com/hhnmag_app/hospitalconnect/search/article.jsp?dcrpath=HHNMAG/PubsNewsArticle/data/040817HHN_Online_LeonChizen&domain=HHNMAG

Lindstrom, L., & Jeffries, R. (2004, SUM). Extreme programming and agile software development methodologies. *INFORMATION SYSTEMS MANAGEMENT*, 21, 41-52.

Lisacek, F., & Appel, R. D. (2007). Systems biology - A loose definition. *Proteomics*, 7, 825-827.

Loewe, L. (2009, FEB 24). A framework for evolutionary systems biology. *BMC SYSTEMS BIOLOGY*, 3.

- Maier, C. W., Long, J. G., Hemminger, B. M., & Giddings, M. C. (2009). Ultra-Structure database design methodology for managing systems biology data and analyses. *Bmc Bioinformatics, 10*.
- Marzolf, B., Deutsch, E. W., Moss, P., Campbell, D., Johnson, M. H., & Galitski, T. (2006, JUN 6). SBEAMS-microarray: Database software supporting genomic expression analyses for systems biology. *BMC BIOINFORMATICS, 7*.
- Mayer, G. (2009, aug). Data management in systems biology I - Overview and bibliography. *ArXiv e-prints*.
- McCusker, J. P., Phillips, J. A., Beltran, A. G., Finkelstein, A., & Krauthammer, M. (2009). Semantic web data warehousing for caGrid. *BMC BIOINFORMATICS, 10*.
- Nature Publishing Groups. (2010). *Molecular Systems Biology: About the journal*. Retrieved 08 25, 2010, from Nature: http://www.nature.com/msb/about/about_site.html
- Oliver, S., Winson, M., Kell, D., & Baganz, F. (1998, SEP). Systematic functional analysis of the yeast genome. *TRENDS IN BIOTECHNOLOGY, 16*, 373-378.
- Shah, A. R., Singhal, M., Klicker, K. R., Stephan, E. G., Wiley, H. S., & Waters, K. M. (2007). Enabling high-throughput data management for systems biology: The bioinformatics resource manager. *Bioinformatics, 23*, 906-909.
- Tatarinov, I., Ives, Z., Madhavan, J., Halevy, A., Suci, D., Dalvi, N., et al. (2003). The Piazza peer data management project. *Sigmod Record, 32*, 47-52.

- Tu, S. W., Campbell, J. R., Glasgow, J., Nyman, M. A., McClure, R., McClay, J., et al. (2007, SEP-OCT). The SAGE guideline model: Achievements and overview. *JOURNAL OF THE AMERICAN MEDICAL INFORMATICS ASSOCIATION*, 14, 589-598.
- U.S. Department of Energy Office of Science. (2009, 11). *DOE Systems Biology Knowledgebase for a New Era in Biology*. Retrieved 07 01, 2010, from Genomic Science Program, energy.gov: <http://genomicscience.energy.gov/compbio/>
- Vawdrey, D. K., Gardner, R. M., Evans, R. S., Orme, J., Clemmer, T. P., Greenway, L., et al. (2007, MAY-JUN). Assessing data quality in manual entry of ventilator settings. *JOURNAL OF THE AMERICAN MEDICAL INFORMATICS ASSOCIATION*, 14, 295-303.
- Wang, K., van Wingerde, F., Bradshaw, K., Szolovits, P., & Kohane, I. (1997). A Java-based multi-institutional medical information retrieval system. *JOURNAL OF THE AMERICAN MEDICAL INFORMATICS ASSOCIATION*, 538-542.
- Wishart, D. S. (2007, JUL). Human Metabolome Database: completing the `human parts list. *PHARMACOGENOMICS*, 8, 683-686.
- Wishart, D. S., Tzur, D., Knox, C., Eisner, R., Guo, A. C., Young, N., et al. (2007, JAN). HMDB: the human metabolome database. *NUCLEIC ACIDS RESEARCH*, 35, D521-D526.

Appendices

Appendix A. Website code

The code below is the final functional code that is generated and stored on the production machine. This will continuously change based on project needs. For the current files and their code, please contact the author.

Most of the webserver host files in two major directories, cgi-bin and htdocs.

The htdocs directory hosting files and folders required for the main website are shown in Figure 47.

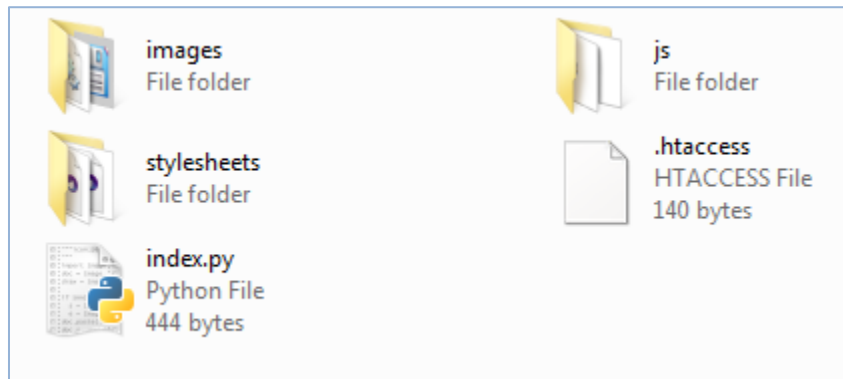


Figure 47 htdocs folder structure

Code written for index.py

```
#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import os
#####
directory = "../cgi-bin/SCCL"
os.chdir(directory)
#####
execfile('start.py')
```

Code written for .htaccess

DirectoryIndex index.py index.html index.htm index.xml index.php index.php3 index.pl
index.cgi index.shtml default.shtml index.xhtml

List of files in the images directory are shown in Figure 48.

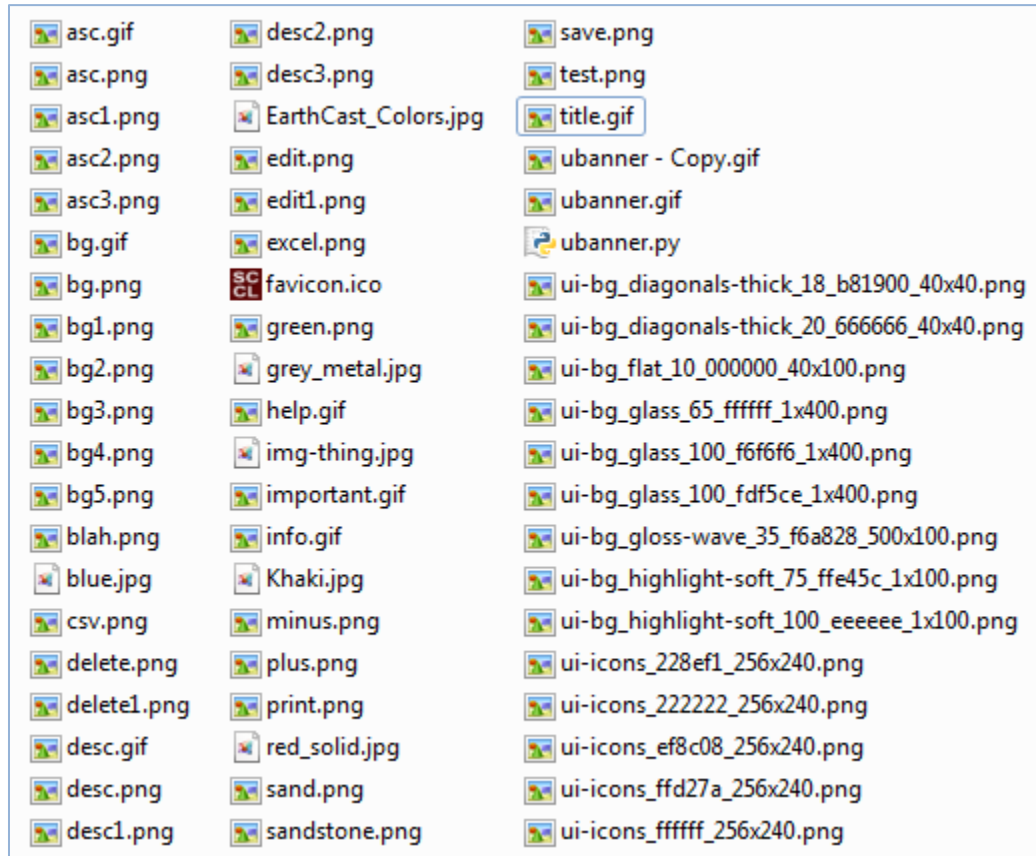


Figure 48 Files in images directory

List of files in js directory are shown in Figure 49.

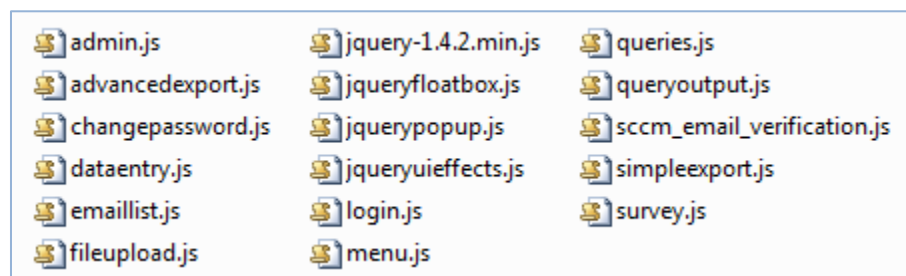


Figure 49 List of files in js directory

Code written for each of the files in js directory is shown below except for emailist.js, jquery files and sccm_email_verification files. This is because they are either not essential for the project or code from an external source.

Code written for admin.js

```
//Inserting a new row after a particular row code start
/*
function insertAfter(newElement,targetElement)
{
    var parent = targetElement.parentNode;
    if(parent.lastchild == targetElement)
    {
        parent.appendChild(newElement);
    }
    else{
        parent.insertBefore(newElement, targetElement.nextSibling);
    }
}
*/
//Inserting a new row after a particular row code end
//Submit form function start
function selectAll(abc)
{
    if(abc=="project")
    {
        if(document.adduserform.allprojects.checked)
        {
            document.adduserform.project_MONR.checked=true;
            document.adduserform.project_PONR.checked=true;
            document.adduserform.project_DARPA.checked=true;
        }
        else if(!document.adduserform.allprojects.checked)
        {
            document.adduserform.project_MONR.checked=false;
            document.adduserform.project_PONR.checked=false;
            document.adduserform.project_DARPA.checked=false;
        }
    }
    else{}
    if(document.edituserform)
    {
        if(document.edituserform.allprojects.checked)
        {
            document.edituserform.project_MONR.checked=true;
            document.edituserform.project_PONR.checked=true;
            document.edituserform.project_DARPA.checked=true;
        }
        else if(!document.edituserform.allprojects.checked)
        {
            document.edituserform.project_MONR.checked=false;
        }
    }
}
```

```

        document.edituserform.project_PONR.checked=false;
        document.edituserform.project_DARPA.checked=false;
    }
    }
    else{}
}
else{}
}
function selectOne(abc,dbc)
{
    if(abc=="project")
    {
        if(document.adduserform.project_MONR.checked==false ||
document.adduserform.project_PONR.checked==false ||
document.adduserform.project_DARPA.checked==false)
        {
            document.adduserform.allprojects.checked = false;
        }
        else if(document.adduserform.project_MONR.checked==true &&
document.adduserform.project_PONR.checked==true &&
document.adduserform.project_DARPA.checked==true)
        {
            document.adduserform.allprojects.checked = true;
        }
        else{}
        if(document.edituserform)
        {
            if(document.edituserform.project_MONR.checked==false ||
document.edituserform.project_PONR.checked==false ||
document.edituserform.project_DARPA.checked==false)
            {
                document.edituserform.allprojects.checked = false;
            }
            else if(document.edituserform.project_MONR.checked==true &&
document.edituserform.project_PONR.checked==true &&
document.edituserform.project_DARPA.checked==true)
            {
                document.edituserform.allprojects.checked = true;
            }
        }
        else{}
    }
    else{}
}
}
function Admin_submitForm(a,b){
    if (a=="adduser"){
        document.getElementById('message1').innerHTML='';
        document.adduserform.user_id.style.color = "Black";
        document.adduserform.password1.style.color = "Black";
        document.adduserform.password2.style.color = "Black";
        document.adduserform.firstname.style.color = "Black";
        document.adduserform.lastname.style.color = "Black";
        document.adduserform.middleinitial.style.color = "Black";
        document.adduserform.email1.style.color = "Black";
        document.adduserform.email2.style.color = "Black";
        document.adduserform.groupname.style.color = "Black";
        document.getElementById('message1').style.color = "Red";
    }
}

```

```

        //document.getElementById('message1').innerHTML='Please wait while your
request is being processed. If the wait time is more than 10sec, submit the form
again';
        if(!((document.adduserform.user_id.value).length> 7) ||
!((document.adduserform.user_id.value).match(/^[A-Za-z]+_[A-Za-z0-9]*$/)))
        {
            document.getElementById('message1').innerHTML= "Error in User ID.
Please check the fine print";
            document.adduserform.user_id.focus();
            document.adduserform.user_id.style.color = "Red";
        }
        //else if(!((document.adduserform.password1.value).length>6) ||
!((document.adduserform.password1.value).match(/[A-Za-z0-9!_@#\$%\&\(\)]+/g)) || ((document.adduserform.password1.value).match(/^[A-Za-
z!_@#\$%\&\(\)]+/g)) ||
!((document.adduserform.password1.value).match(/[A-Za-z]+/g)) ||
!((document.adduserform.password1.value).match(/[0-9]+/g)))
        //else if(!((document.adduserform.password1.value).length>6) ||
((document.adduserform.password1.value).match(/[\+\=\'\\"\\]+/g)) ||
!((document.adduserform.password1.value).match(/[A-Za-z]+/g)) ||
!((document.adduserform.password1.value).match(/[0-9]+/g)))
        else if(!((document.adduserform.password1.value).length>6) ||
((document.adduserform.password1.value).match(/^[A-Za-z0-9!@#\$%^&*\(\)_-\
\?\<\>,\.\|\/\;\:\|]+/g)) || !((document.adduserform.password1.value).match(/[A-Za-
z]+/g)) || !((document.adduserform.password1.value).match(/[0-9]+/g)))
        {
            document.getElementById('message1').innerHTML= "Error in Password.
Please check the fine print";
            document.adduserform.password1.focus();
            document.adduserform.password1.style.color = "Red";
        }
        else if((document.adduserform.password2.value) !=
(document.adduserform.password1.value))
        {
            document.getElementById('message1').innerHTML= "Error in re-
entered password. It did not match the password";
            document.adduserform.password2.focus();
            document.adduserform.password2.style.color = "Red";
        }
        else if((document.adduserform.firstname.value=='') ||
!((document.adduserform.firstname.value).match(/[A-Za-z]+/)))
        {
            document.getElementById('message1').innerHTML= "Error in first
name. Please check the fine print";
            document.adduserform.firstname.focus();
            document.adduserform.firstname.style.color = "Red";
        }
        else if((document.adduserform.lastname.value!='') &&
!((document.adduserform.lastname.value).match(/[A-Za-z]+/)))
        {
            document.getElementById('message1').innerHTML= "Error in last
name. Please check the fine print";
            document.adduserform.lastname.focus();
            document.adduserform.lastname.style.color = "Red";
        }
        else if((document.adduserform.middleinitial.value!='') &&
!((document.adduserform.middleinitial.value).match(/[A-Za-z]+/)))
        {

```

```

        document.getElementById('message1').innerHTML= "Error in middle
initial. Please check the fine print";
        document.adduserform.middleinitial.focus();
        document.adduserform.middleinitial.style.color = "Red";
    }
    else if(!(document.adduserform.email1.value).match(/^([A-Za-z]+_?[0-9]*\.[A-Za-z0-9]+@[A-Za-z]+[0-9]*\.[A-Za-z]*\.[A-Za-z]{2,})$/g))
    {
        document.getElementById('message1').innerHTML= "Error in email.
The email is not in the right format";
        document.adduserform.email1.focus();
        document.adduserform.email1.style.color = "Red";
    }
    else
    if((document.adduserform.email2.value)!=(document.adduserform.email1.value))
    {
        document.getElementById('message1').innerHTML= "Error in re-
entered email. It did not match the email";
        document.adduserform.email2.focus();
        document.adduserform.email2.style.color = "Red";
    }
    else if(document.adduserform.groupname.value=='Not_selected')
    {
        document.getElementById('message1').innerHTML= "One of the groups
should be selected";
        document.adduserform.groupname.focus();
        document.adduserform.groupname.style.color = "Red";
    }
    else if(!document.adduserform.allprojects.checked &&
!document.adduserform.project_MONR.checked &&
!document.adduserform.project_PONR.checked &&
!document.adduserform.project_DARPA.checked)
    {
        document.getElementById('message1').innerHTML= "At least one
of the studies should be checked";
    }

    else{
        document.getElementById('message1').innerHTML='';
        document.getElementById('message1').style.color = "Black";
        document.getElementById('message1').innerHTML='Please wait while your
request is being processed. If the wait time is more than 10sec, submit the form
again';

        var TestOnly="No";
        var MONR="No";
        var PONR="No";
        var DARPA="No";
        //When more studies are added, add them here
        var All_studies="No";
        var theDate = new Date;
        var Account_creation_day;
        var Account_creation_month;
        var Account_creation_year = (theDate.getFullYear()).toString();
        if(((theDate.getMonth()+1).toString()).length==1)
        {
            Account_creation_month =
"0"+(theDate.getMonth()+1).toString();
        }
    }

```

```

else{
    Account_creation_month = (theDate.getMonth()+1).toString();
}
if((theDate.getDate()).toString().length==1)
{
    Account_creation_day = "0" + theDate.getDate().toString();
}
else
{
    Account_creation_day = theDate.getDate().toString();
}
var Account_creation_date = Account_creation_year + "-" +
Account_creation_month + "-" + Account_creation_day;
if(document.adduserform.testonly.checked)
{
    TestOnly="Yes"
}
else{}
if(document.adduserform.allprojects.checked)
{
    All_studies="Yes"
}
else{
    if(document.adduserform.project_MONR.checked)
    {
        MONR="Yes"
    }
    else{}
    if(document.adduserform.project_PONR.checked)
    {
        PONR="Yes"
    }
    else{}
    if(document.adduserform.project_DARPA.checked)
    {
        DARPA="Yes"
    }
    else{}
    //When more studies are added, add them here
}

```

```

var parameters = "which=" + escape("adduser")+
"&User_id=" +
escape(document.adduserform.user_id.value)+
"&Password=" +
escape(document.adduserform.password1.value)+
"&Email=" +
escape(document.adduserform.email1.value)+
"&User_first_name=" +
escape(document.adduserform.firstname.value)+
"&User_last_name=" +
escape(document.adduserform.lastname.value)+
"&User_middle_initial=" +
escape(document.adduserform.middleinitial.value)+
"&Group_name=" +
escape(document.adduserform.groupname.value)+
"&TestOnly=" + escape(TestOnly)+

```

```

        "&MONR=" + escape(MONR)+
        "&PONR=" + escape(PONR)+
        "&DARPA=" + escape(DARPA)+
        "&All_studies=" + escape(All_studies)+
        "&Account_creation_date=" +
escape(Account_creation_date);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
}
else if(a=="edituser")
{
    oldUser_id = b;
    document.getElementById('message2').innerHTML='';
    document.edituserform.user_id.style.color = "Black";
    document.edituserform.password1.style.color = "Black";
    document.edituserform.password2.style.color = "Black";
    document.edituserform.firstname.style.color = "Black";
    document.edituserform.lastname.style.color = "Black";
    document.edituserform.middleinitial.style.color = "Black";
    document.edituserform.email1.style.color = "Black";
    document.edituserform.email2.style.color = "Black";
    document.edituserform.groupname.style.color = "Black";
    document.getElementById('message2').style.color = "Red";
    //document.getElementById('message2').innerHTML='Please wait while your
request is being processed. If the wait time is more than 10sec, submit the form
again';
    if(!((document.edituserform.user_id.value).length> 7) ||
!((document.edituserform.user_id.value).match(/^[A-Za-z]+_[A-Za-z0-9]*$/)))
    {
        document.getElementById('message2').innerHTML= "Error in User ID.
Please check the fine print";
        document.edituserform.user_id.focus();
        document.edituserform.user_id.style.color = "Red";
    }
    //else
if(!((document.edituserform.password1.value).length==0)&&!((document.edituserform
.password2.value).length==0)&&!((document.edituserform.password1.value).length>6
) || !((document.edituserform.password1.value).match(/[A-Za-z0-9,_*\.\.-
@#\$\%\&\(\)]+/g)) || ((document.edituserform.password1.value).match(/^[A-Za-
z_\*\.\.-@#\$\%\&\(\),0-9]+/g)) ||
!((document.edituserform.password1.value).match(/[A-Za-z]+/g)) ||
!((document.edituserform.password1.value).match(/[0-9]+/g))))
    //else
if(!((document.edituserform.password1.value).length==0)&&!((document.edituserform
.password2.value).length==0)&&!((document.edituserform.password1.value).length>6
) || ((document.edituserform.password1.value).match(/[\+=\=\'"\|]+/g)) ||
!((document.edituserform.password1.value).match(/[A-Za-z]+/g)) ||
!((document.edituserform.password1.value).match(/[0-9]+/g))))
    else
if(!((document.edituserform.password1.value).length==0)&&!((document.edituserform
.password2.value).length==0)&&!((document.edituserform.password1.value).length>6
) || ((document.edituserform.password1.value).match(/^[A-Za-z0-9!@#\$^&*\(\)_\.-

```

```

\?\<\>,\.\\/\;\:\|]+/g)) || !((document.edituserform.password1.value).match(/[A-Za-z]+/g)) || !((document.edituserform.password1.value).match(/[0-9]+/g)))
    {
        document.getElementById('message2').innerHTML= "Error in Password.
Please check the fine print";
        document.edituserform.password1.focus();
        document.edituserform.password1.style.color = "Red";
    }
    else if((document.edituserform.password2.value) !=
(document.edituserform.password1.value))
    {
        document.getElementById('message2').innerHTML= "Error in re-
entered password. It did not match the password";
        document.edituserform.password2.focus();
        document.edituserform.password2.style.color = "Red";
    }
    else if((document.edituserform.firstname.value=='') ||
!((document.edituserform.firstname.value).match(/[A-Za-z]+/)))
    {
        document.getElementById('message2').innerHTML= "Error in first
name. Please check the fine print";
        document.edituserform.firstname.focus();
        document.edituserform.firstname.style.color = "Red";
    }
    else if((document.edituserform.lastname.value!='') &&
!((document.edituserform.lastname.value).match(/[A-Za-z]+/)))
    {
        document.getElementById('message2').innerHTML= "Error in last
name. Please check the fine print";
        document.edituserform.lastname.focus();
        document.edituserform.lastname.style.color = "Red";
    }
    else if((document.edituserform.middleinitial.value!='') &&
!((document.edituserform.middleinitial.value).match(/[A-Za-z]+/)))
    {
        document.getElementById('message2').innerHTML= "Error in middle
initial. Please check the fine print";
        document.edituserform.middleinitial.focus();
        document.edituserform.middleinitial.style.color = "Red";
    }
    else if(!((document.edituserform.email1.value).match(/^([A-Za-z]+_?[0-
9]*\.[A-Za-z0-9]+@[A-Za-z]+[0-9]*\.[A-Za-z]*\.[A-Za-z]{2,})$/g))
    {
        document.getElementById('message2').innerHTML= "Error in email.
The email is not in the right format";
        document.edituserform.email1.focus();
        document.edituserform.email1.style.color = "Red";
    }
    else
if((document.edituserform.email2.value)!=((document.edituserform.email1.value))
    {
        document.getElementById('message2').innerHTML= "Error in re-
entered email. It did not match the email";
        document.edituserform.email2.focus();
        document.edituserform.email2.style.color = "Red";
    }
    else if(document.edituserform.groupname.value=='Not_selected')
    {

```



```

        document.getElementById('message2').innerHTML= "One of the groups
should be selected";
        document.edituserform.groupname.focus();
        document.edituserform.groupname.style.color = "Red";
    }
    else if(!document.edituserform.allprojects.checked &&
        !document.edituserform.project_MONR.checked &&
        !document.edituserform.project_PONR.checked &&
        !document.edituserform.project_DARPA.checked)
    {
        document.getElementById('message2').innerHTML= "At least one
of the studies should be checked";
    }

    else{
        document.getElementById('message2').innerHTML='';
        document.getElementById('message2').style.color = "Black";
        document.getElementById('message2').innerHTML='Please wait while your
request is being processed. If the wait time is more than 10sec, submit the form
again';

        var TestOnly="No";
        var MONR="No";
        var PONR="No";
        var DARPA="No";
        //When more studies are added, add them here
        var All_studies="No";
        var Password = "None";
        var theDate = new Date;
        var Account_creation_day;
        var Account_creation_month;
        var Account_creation_year = (theDate.getFullYear()).toString();
        if(!((document.edituserform.password1.value).length==0))
        {
            Password = document.edituserform.password1.value;
        }
        else{}
        if((theDate.getMonth()+1).toString().length==1)
        {
            Account_creation_month =
"0"+(theDate.getMonth()+1).toString();
        }
        else{
            Account_creation_month = (theDate.getMonth()+1).toString();
        }
        if((theDate.getDate()).toString().length==1)
        {
            Account_creation_day = "0" + theDate.getDate().toString();
        }
        else
        {
            Account_creation_day = theDate.getDate().toString();
        }
        var Account_creation_date = Account_creation_year + "-" +
Account_creation_month + "-" + Account_creation_day;
        if(document.edituserform.testonly.checked)
        {
            TestOnly="Yes"
        }
    }

```

```

else{}
if(document.edituserform.allprojects.checked)
{
    All_studies="Yes"
}
else{
    if(document.edituserform.project_MONR.checked)
    {
        MONR="Yes"
    }
    else{}
    if(document.edituserform.project_PONR.checked)
    {
        PONR="Yes"
    }
    else{}
    if(document.edituserform.project_DARPA.checked)
    {
        DARPA="Yes"
    }
    else{}
    //When more studies are added, add them here
}

var parameters = "which=" + escape("edituser")+
"&oldUser_id=" + escape(oldUser_id)+
"&User_id=" +
escape(document.edituserform.user_id.value)+
"&Password=" + escape>Password)+
"&Email=" +
escape(document.edituserform.email1.value)+
"&User_first_name=" +
escape(document.edituserform.firstname.value)+
"&User_last_name=" +
escape(document.edituserform.lastname.value)+
"&User_middle_initial=" +
escape(document.edituserform.middleinitial.value)+
"&Group_name=" +
escape(document.edituserform.groupname.value)+
"&TestOnly=" + escape(TestOnly)+
"&MONR=" + escape(MONR)+
"&PONR=" + escape(PONR)+
"&DARPA=" + escape(DARPA)+
"&All_studies=" + escape(All_studies);
var method="POST";
var submitto = "../cgi-bin/SCCL/formsubmit.py";
var contenttype = "application/x-www-form-urlencoded";
var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
}
else if(a=="deleteuser")
{
    User_id = b;
    var parameters = "which=" + escape("deleteuser")+ "&User_id="+
escape(User_id);

```

```

        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype ="application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    else{}
}
function clearFields()
{
    document.adduserform.user_id.value = "";
    document.adduserform.password1.value = "";
    document.adduserform.password2.value = "";
    document.adduserform.firstname.value = "";
    document.adduserform.lastname.value = "";
    document.adduserform.middleinitial.value = "";
    document.adduserform.email1.value = "";
    document.adduserform.email2.value = "";
    document.adduserform.groupname.value = "Not_selected";
    document.adduserform.testonly.checked=false;
    document.adduserform.allprojects.checked=false;
    document.adduserform.project_MONR.checked=false;
    document.adduserform.project_PONR.checked=false;
    document.adduserform.project_DARPA.checked=false;
}
//window.onload=loadUserData;
function loadUserData()
{
    var parameters = "which=" + escape("getuserdata");
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function editUserData(a)
{
    if(document.getElementById("userdataedittable"))
    {
        alert("Please complete the previously initiated User Data Edit")
    }
    else
    {
        var User_id = a;
        document.getElementById('message1').innerHTML = '';
        document.getElementById(User_id).innerHTML = '<form
id="edituserform" name="edituserform" action="" method="post"
enctype="application/x-www-form-urlencoded" onsubmit="javascript:return
false"><table>
<tbody
id="userdataedittable">
<tr>
<td
class="descriptor1">
for="' +User_id+'user_id">

```

```

<sup>1</sup>User ID*:\
                                </label>\
                                </td>\
                                <td class="field1"
title="Enter User ID">\
                                <input type="text"
name="user_id" id="' + User_id + 'user_id\" maxlength="30" size="20" tabindex="16"/>\
                                </td>\
                                <td
class="descriptor1">\
                                <label
for="' + User_id + 'password1\">\
<sup>2</sup>Password*:\
                                </label>\
                                </td>\
                                <td class="field1"
title="Enter Password">\
                                <input
type="password" name="password1" id="' + User_id + 'password1\" maxlength="30"
size="20" tabindex="17"/>\
                                </td>\
                                <td
class="descriptor1">\
                                <label
for="' + User_id + 'password2\">\
                                Re-enter
Password*:\
                                </label>\
                                </td>\
                                <td class="field1"
title="Re-enter Password">\
                                <input
type="password" name="password2" id="' + User_id + 'password2\" maxlength="30"
size="20" tabindex="18"/>\
                                </td>\
                                </tr>\
                                <tr>\
                                <td
class="descriptor1">\
                                <label
for="' + User_id + 'firstname\">\
<sup>3</sup>First Name*:\
                                </label>\
                                </td>\
                                <td class="field1"
title="Enter First Name of the User">\
                                <input type="text"
name="firstname" id="' + User_id + 'firstname\" maxlength="30" size="20"
tabindex="19"/>\
                                </td>\
                                <td
class="descriptor1">\
                                <label
for="' + User_id + 'lastname\">\

```

```

<sup>3</sup>Last Name:\
                                </label>\
                                </td>\
                                <td class="field1"
title="Enter Last Name of the User">\
                                <input type="text"
name="lastname" id="' +User_id+'lastname\" maxlength="30" size="20"
tabindex="20"/>\
                                </td>\
                                <td
class="descriptor1">\
                                <label
for="' +User_id+'middleinitial\">\
<sup>4</sup>Middle Initial:\
                                </label>\
                                </td>\
                                <td class="field1"
title="Enter Middle Initial of the User">\
                                <input type="text"
name="middleinitial" id="' +User_id+'middleinitial\" maxlength="2" size="20"
tabindex="21"/>\
                                </td>\
                                </tr>\
                                <tr>\
                                <td
class="descriptor1">\
                                <label
for="' +User_id+'email1\">\
                                Email
address*:\
                                </label>\
                                </td>\
                                <td class="field1"
title="Enter user email address">\
                                <input type="text"
name="email1" id="' +User_id+'email1\" maxlength="40" size="20" tabindex="22"/>\
                                </td>\
                                <td
class="descriptor1">\
                                <label
for="' +User_id+'email2\">\
                                Re-Enter Email
address*:\
                                </label>\
                                </td>\
                                <td class="field1"
title="Re-Enter user email address">\
                                <input type="text"
name="email2" id="' +User_id+'email2\" maxlength="40" size="20" tabindex="23"/>\
                                </td>\
                                <td
class="descriptor1">\
                                <label
for="' +User_id+'groupname\">\
                                Group
assigned*:\

```

```


```

<pre> </pre>	<pre> </pre>
<pre> </pre>	<pre> </pre>
<pre> </pre>	<pre> </pre>
<pre> </pre>	<pre> </pre>

```

name="project_DARPA" id="' +User_id+'project_DARPA"
onclick="javascript:selectOne('\project\',this);" tabindex="28"/>
</td>
<td
class="descriptor1">
<button
class="buttons" type="submit" name="edituser" id="edituser" value="edituser"
title="Submit the add user form"
onclick="javascript:Admin_submitForm('\edituser\',\'+User_id+\');return false"
tabindex="29">
Submit</button>
</td>
<td class="field1"
style="text-align:center">
<button
class="buttons" type="button" name="deleteuser" id="deleteuser"
value="deleteuser"
title="Submit the add user form"
onclick="javascript:Admin_submitForm('\deleteuser\',\'+User_id+\');return
false" tabindex="30">
Delete
User</button>
<button
class="buttons" type="button" name="canceledit" id="canceledit"
value="canceledit"
title="Submit the add user form" onclick="javascript:loadUserData();"
tabindex="31">
Cancel</button>
</td>
</tr>
<tr>
<td id="message2"
colspan="6">
&nbsp;
</td>
</tr>
</tbody>
</table>
</form>
var parameters = "which=" +
escape("loadedituserdata")+ "&User_id="+ escape(User_id);
var method="POST";
var submitto = "../cgi-bin/SCCL/formsubmit.py";
var contenttype = "application/x-www-form-urlencoded";
var runthisonstatechange = showContents;
makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
}
function loadEditUserData(a,b,c,d,e,f,g,h,i,j,k)
{
document.edituserform.user_id.value = a;
document.edituserform.firstname.value = c;

```

```

document.edituserform.lastname.value = d;
document.edituserform.middleinitial.value = e;
document.edituserform.email1.value = b;
document.edituserform.email2.value = b;
document.edituserform.groupname.value = f;
if(g=="Yes")
{
    document.edituserform.testonly.checked=true;
}
else{}
if(h=="Yes")
{
    document.edituserform.project_MONR.checked=true;
    document.edituserform.allprojects.checked=false;
}
else{}
if(i=="Yes")
{
    document.edituserform.project_PONR.checked=true;
    document.edituserform.allprojects.checked=false;
}
else{}
if(j=="Yes")
{
    document.edituserform.project_DARPA.checked=true;
    document.edituserform.allprojects.checked=false;
}
else{}
if(k=="Yes")
{
    document.edituserform.allprojects.checked=true;
    document.edituserform.project_MONR.checked=true;
    document.edituserform.project_PONR.checked=true;
    document.edituserform.project_DARPA.checked=true;
}
else{}
}

```

Code written for advancedexport.js

```

var removeboxes=new Array;
var removecontent=new Array;
var showboxes=new Array;
var allForms=new Array;
var datarequestForms=new Array;
var datarequestVar=new Array;
var datalogicForms=new Array;
var datalogicVar=new Array;
var datadeterminationForms=new Array;
var datadeterminationVar=new Array;
var tempdirectoryname='null';
var firstmsg='<div class="innerdiv">\
    <table class="wrappertable"><tr><td class="wrappercell">\
        <table class="selection">\
            <tbody>\
                <tr class="oddRow">\

```



```

        <td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> Please select</span></td>\
        </tr>\
    </tbody>\
</table>\
</td></tr></table>\
</div>'
var eventtmsg='<div class="innerdiv">\
    <table class="wrappertable"><tr><td class="wrappercell">\
        <table class="selection">\
            <tbody>\
                <tr class="oddRow">\
                    <td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> Please select \"Selected
events\" option to populate the events</span></td>\
                    </tr>\
                </tbody>\
            </table>\
        </td></tr></table>\
    </div>'
var waitmsg='<div class="innerdiv">\
    <table class="wrappertable"><tr><td class="wrappercell">\
        <table class="selection">\
            <tbody>\
                <tr class="oddRow">\
                    <td class="inform"><span class="arrowmsg"> Please
wait.....</span></td>\
                    </tr>\
                </tbody>\
            </table>\
        </td></tr></table>\
    </div>'
var selectGroups='<div class="innerdiv">\
    <table class="wrappertable"><tr><td class="wrappercell">\
        <table class="selection">\
            <tbody>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="allgroups">All Groups</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
onclick="javascript:selectAll('\groups\');" id="allgroups" name="allgroups"></td>\
                    </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="Group_1">Group-1 (C1: Instrumentation/No shock)</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
onclick="javascript:selectOne('\groups\');" id="Group_1" name="Group_1"></td>\
                    </tr>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="Group_2">Group-2 (C2: Shock - No resuscitation)</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
onclick="javascript:selectOne('\groups\');" id="Group_2" name="Group_2"></td>\
                    </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="Group_3">Group-3 (E1: Shock - Resuscitation)</label></td>\

```

```

                                <td class="inputcolumn"><input type="checkbox"
onclick="javascript:selectOne('\groups\');" id="Group_3" name="Group_3"></td>\
                                </tr>\
                                <tr class="oddRow">\
                                    <td class="optionscolumn"><label
for="Group_4">Group-4 (E2: Pre feed)</label></td>\
                                    <td class="inputcolumn"><input type="checkbox"
onclick="javascript:selectOne('\groups\');" id="Group_4" name="Group_4"></td>\
                                    </tr>\
                                </tbody>\
                            </table>\
                        </td></tr></table>\
                    </div>'
var selectVitals='<div class="innerdiv">\
    <table class="wrappertable"><tr><td class="wrappercell">\
        <table class="selection">\
            <tbody>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="allvitals">Select All</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="allvitals" id="allvitals" onclick="javascript:selectAll('\vitals\');"
/></td>\
                    </tr>\
                    <tr class="evenRow">\
                        <td class="optionscolumn"><label
for="monr_vitals_manual_one_hour-monr_vitals_manual_10_min">Vitals and
Gases</label></td>\
                        <td class="inputcolumn"><input type="checkbox"
name="monr_vitals_manual_10_min-monr_vitals_manual_one_hour"
id="monr_vitals_manual_one_hour-monr_vitals_manual_10_min"
onclick="javascript:selectOne('\vitals\');" /></td>\
                        </tr>\
                        <tr class="oddRow">\
                            <td class="optionscolumn"><label
for="monr_lab_values">Lab values</label></td>\
                            <td class="inputcolumn"><input type="checkbox"
name="monr_lab_values" id="monr_lab_values"
onclick="javascript:selectOne('\vitals\');" /></td>\
                            </tr>\
                            <tr class="evenRow">\
                                <td class="optionscolumn"><label
for="monr_gcs_scores">GCS scores</label></td>\
                                <td class="inputcolumn"><input type="checkbox"
name="monr_gcs_scores" id="monr_gcs_scores"
onclick="javascript:selectOne('\vitals\');" /></td>\
                                </tr>\
                                <tr class="oddRow">\
                                    <td class="optionscolumn"><label
for="monr_teg_data">Teg data</label></td>\
                                    <td class="inputcolumn"><input type="checkbox"
name="monr_teg_data" id="monr_teg_data"
onclick="javascript:selectOne('\vitals\');" /></td>\
                                    </tr>\
                                </tbody>\
                            </table>\
                        </td></tr></table>\
                    </div>'

```

```

var selectHNMR='<div class="innerdiv">\
    <table class="wrappertable"><tr><td class="wrappercell">\
        <table class="selection">\
            <tbody>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="allhnmr">Select All</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="allhnmr" id="allhnmr" onclick="javascript:selectAll('\hnmr\');" /></td>\
                </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="monr_hnmr_muscle|No">HNMR Muscle Raw data</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_muscle|No" id="monr_hnmr_muscle|No"
onclick="javascript:selectOne('\hnmr\');" /></td>\
                </tr>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="monr_hnmr_muscle|Wet_tissue">HNMR Muscle Wet-tissue data</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_muscle|Wet_tissue" id="monr_hnmr_muscle|Wet_tissue"
onclick="javascript:selectOne('\hnmr\');" /></td>\
                </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="monr_hnmr_muscle|Lyophilized_tissue">HNMR Muscle Lyophilized-tissue
data</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_muscle|Lyophilized_tissue"
id="monr_hnmr_muscle|Lyophilized_tissue" onclick="javascript:selectOne('\hnmr\');"
/></td>\
                </tr>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="monr_hnmr_liver|No">HNMR Liver Raw data</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_liver|No" id="monr_hnmr_liver|No"
onclick="javascript:selectOne('\hnmr\');" /></td>\
                </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="monr_hnmr_liver|Wet_tissue">HNMR Liver Wet-tissue data</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_liver|Wet_tissue" id="monr_hnmr_liver|Wet_tissue"
onclick="javascript:selectOne('\hnmr\');" /></td>\
                </tr>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="monr_hnmr_liver|Lyophilized_tissue">HNMR Liver Lyophilized-tissue
data</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_liver|Lyophilized_tissue" id="monr_hnmr_liver|Lyophilized_tissue"
onclick="javascript:selectOne('\hnmr\');" /></td>\
                </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="monr_hnmr_urine|No">HNMR Urine Raw data</label></td>\

```

```

                <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_urine|No" id="monr_hnmr_urine|No"
onclick="javascript:selectOne('\hnmr\');" /></td>\
            </tr>\
            <tr class="oddRow">\
                <td class="optionscolumn"><label
for="monr_hnmr_urine|Yes">HNMR Urine Normalized data</label></td>\
                <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_urine|Yes" id="monr_hnmr_urine|Yes"
onclick="javascript:selectOne('\hnmr\');" /></td>\
            </tr>\
            <tr class="evenRow">\
                <td class="optionscolumn"><label
for="monr_hnmr_serum|No">HNMR Serum Raw data</label></td>\
                <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_serum|No" id="monr_hnmr_serum|No"
onclick="javascript:selectOne('\hnmr\');" /></td>\
            </tr>\
            <tr class="oddRow">\
                <td class="optionscolumn"><label
for="monr_hnmr_serum|Yes">HNMR Serum Normalized data</label></td>\
                <td class="inputcolumn"><input type="checkbox"
name="monr_hnmr_serum|Yes" id="monr_hnmr_serum|Yes"
onclick="javascript:selectOne('\hnmr\');" /></td>\
            </tr>\
        </tbody>\
    </table>\
</td></tr></table>\
</div>'
var selectPNMR='<div class="innerdiv">\
    <table class="wrappertable"><tr><td class="wrappercell">\
        <table class="selection">\
            <tbody>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="allpnmr">Select All</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="allpnmr" id="allpnmr" onclick="javascript:selectAll('\pnmr\');" /></td>\
                </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="monr_pnmr_muscle|No">PNMR Muscle Raw data</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_pnmr_muscle|No" id="monr_pnmr_muscle|No"
onclick="javascript:selectOne('\pnmr\');" /></td>\
                </tr>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="monr_pnmr_muscle|Wet_tissue">PNMR Muscle Wet-tissue data</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_pnmr_muscle|Wet_tissue" id="monr_pnmr_muscle|Wet_tissue"
onclick="javascript:selectOne('\pnmr\');" /></td>\
                </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="monr_pnmr_muscle|Lyophilized_tissue">PNMR Muscle Lyophilized-tissue
data</label></td>\

```

```

                <td class="inputcolumn"><input type="checkbox"
name="monr_pnmr_muscle|Lyophilized_tissue"
id="monr_pnmr_muscle|Lyophilized_tissue" onclick="javascript:selectOne('\pnmr\');"
/></td>\
            </tr>\
            <tr class="oddRow">\
                <td class="optionscolumn"><label
for="monr_pnmr_liver|No">PNMR Liver Raw data</label></td>\
                <td class="inputcolumn"><input type="checkbox"
name="monr_pnmr_liver|No" id="monr_pnmr_liver|No"
onclick="javascript:selectOne('\pnmr\');" /></td>\
            </tr>\
            <tr class="evenRow">\
                <td class="optionscolumn"><label
for="monr_pnmr_liver|Wet_tissue">PNMR Liver Wet-tissue data</label></td>\
                <td class="inputcolumn"><input type="checkbox"
name="monr_pnmr_liver|Wet_tissue" id="monr_pnmr_liver|Wet_tissue"
onclick="javascript:selectOne('\pnmr\');" /></td>\
            </tr>\
            <tr class="oddRow">\
                <td class="optionscolumn"><label
for="monr_pnmr_liver|Lyophilized_tissue">PNMR Liver Lyophilized-tissue
data</label></td>\
                <td class="inputcolumn"><input type="checkbox"
name="monr_pnmr_liver|Lyophilized_tissue" id="monr_pnmr_liver|Lyophilized_tissue"
onclick="javascript:selectOne('\pnmr\');" /></td>\
            </tr>\
        </tbody>\
    </table>\
</td></tr></table>\
</div>'
var selectBinning='<div class="innerdiv">\
    <table class="wrappertable"><tr><td class="wrappercell">\
        <table class="selection">\
            <tbody>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="allbinning">Select All</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="allbinning" id="allbinning" onclick="javascript:selectAll('\binning\');"
/></td>\
                </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="monr_binning_muscle">Binning Muscle</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_binning_muscle" id="monr_binning_muscle"
onclick="javascript:selectOne('\binning\');" /></td>\
                </tr>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="monr_binning_liver">Binning Liver</label></td>\
                    <td class="inputcolumn"><input type="checkbox"
name="monr_binning_liver" id="monr_binning_liver"
onclick="javascript:selectOne('\binning\');" /></td>\
                </tr>\
            <tr class="evenRow">\

```

```

                <td class="optionscolumn"><label
for="monr_binning_urine">Binning Urine</label></td>\
                <td class="inputcolumn"><input type="checkbox"
name="monr_binning_urine" id="monr_binning_urine"
onclick="javascript:selectOne('\binning\');" /></td>\
            </tr>\
            <tr class="oddRow">\
                <td class="optionscolumn"><label
for="monr_binning_serum">Binning Serum</label></td>\
                <td class="inputcolumn"><input type="checkbox"
name="monr_binning_serum" id="monr_binning_serum"
onclick="javascript:selectOne('\binning\');" /></td>\
            </tr>\
        </tbody>\
    </table>\
</td></tr></table>\
</div>'
var selectDatapointFrequency='<div class="innerdiv">\
    <table class="wrappertable"><tr><td
class="wrappercell">\
        <table class="selection">\
            <tbody>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="allevents">All events</label></td>\
                    <td class="inputcolumn"><input
type="radio" name="datapointfrequency" id="allevents" value="allevents"
checked="checked" onclick="javascript:showEventsAdvancedExport(this.value);"
/></td>\
                </tr>\
                <tr class="evenRow">\
                    <td class="optionscolumn"><label
for="selecedevents">Selected events</label></td>\
                    <td class="inputcolumn"><input
type="radio" name="datapointfrequency" id="selecedevents" value="selecedevents"
onclick="javascript:showEventsAdvancedExport(this.value);" /></td>\
                </tr>\
                <tr class="oddRow">\
                    <td class="optionscolumn"><label
for="alldatapoints">All data points</label></td>\
                    <td class="inputcolumn"><input
type="radio" name="datapointfrequency" id="alldatapoints" value="alldatapoints"
onclick="javascript:showEventsAdvancedExport(this.value);" /></td>\
                </tr>\
            </tbody>\
        </table>\
    </td></tr></table>\
</div>'
var queryform='<div id="querywrapper" class="wrapform">\
    <div>\
        <div class="descriptor" style="padding-bottom:10px">Name
of the query</div>\
        <div style="width:468px;height:24px"><input type="text"
name="queryname" id="queryname" maxlength="100" /></div>\
    </div>\
    <div class="descriptor" style="padding-
bottom:10px">Description of the query</div>\

```

```

        <div style="width:468px;height:72px"><textarea
name="querydescription" id="querydescription"
onfocus="javascript:this.select();">No description entered</textarea></div>\
    </div>\
    <div>\
        <div class="descriptor">Share with</div>\
        <div class="box">\
            <div id="userid" class="outerdiv">&nbsp;</div>\
        </div>\
    </div>\
    <div style="text-align:right">\
        <button type="button" id="cancelsavequery"
name="cancelsavequery" value="cancelsavequery" class="buttons" title="Cancel
Saving Query" onclick="javascript:opener.closeSaveQueryForm();">Cancel</button>\
        <button type="submit" id="savequery" name="savequery"
value="savequery" class="buttons" title="Save query"
onclick="javascript:opener.saveQuery();return false">Save</button>\
    </div>\
</div>'
function loadGroupsAdvancedExport()
{
    document.getElementById('groups').innerHTML=selectGroups;
}
function loadSubjectsAdvancedExport(abc)
{
    var parameters = "which=" +
escape("loadSubjectsAdvancedExport")+ "&groups="+escape(abc);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function loadVitalsVariablesAdvancedExport(abc)
{
    var parameters = "which=" +
escape("loadVitalsVariablesAdvancedExport")+ "&vitals_tables="+escape(abc);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function loadHNMRVariablesAdvancedExport(abc)
{
    var parameters = "which=" +
escape("loadHNMRVariablesAdvancedExport")+ "&hnmr_tables="+escape(abc);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function loadPNMRVariablesAdvancedExport(abc)
{
    var parameters = "which=" +
escape("loadPNMRVariablesAdvancedExport")+ "&pnmr_tables="+escape(abc);
    var method="POST";

```

```

        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype = "application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
function loadBinningVariablesAdvancedExport(abc)
{
    var parameters = "which=" +
escape("loadBinningVariablesAdvancedExport")+ "&binning_tables="+escape(abc);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function loadEvIndVarAdvancedExport()
{
    var parameters = "which=" + escape("loadEvIndVarAdvancedExport");
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function showEventsAdvancedExport(abc)
{
    document.getElementById('eventchoice').innerHTML=waitmsg;
    if (abc=="selectedevents")
    {
        var parameters = "which=" + escape("showEventsAdvancedExport");
        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype = "application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    else
    {
        document.getElementById('eventchoice').innerHTML=eventtmsg;
    }
}
var lowerlimit=0;
//var numberofdatapoints=250;
function queryAdvancedExport(numberofdatapoints)
{
    var
formstosend=['groupsform','subjectsform','vitalsdataselectform','vitalsvariablesfo
rm','hnmrdataselectform','hnmrvariablesform','pnmrdataselectform','pnmrvariablesfo
rm','binningdataselectform','binningvariablesform','eventindependentvarselectform'
,'datapointsselectform','eventchoiceform']
    var requestedvariables='';
    var firstvar=0;
    try
    {
        for(var i=0;i<document.forms.length;i++)
        {
            for (var x in formstosend)

```



```

        {
            if (document.forms[i].name==formstosend[x])
            {
                for(var
j=0;j<document.forms[i].elements.length;j++)
                {

//if(document.forms[i].elements[j].type=="checkbox")

if((document.forms[i].elements[j].type=="checkbox" ||
document.forms[i].elements[j].type=="radio")&&
document.forms[i].elements[j].checked==true)
                {

                    if (firstvar==0)
                    {

requestedvariables=requestedvariables+document.forms[i].name+'|'+document.forms[i]
.elements[j].id;

firstvar++;

                    }
                    else
                    {

requestedvariables=requestedvariables+'|'+document.forms[i].name+'|'+document.form
s[i].elements[j].id;

                    }
                }
            }
        }
    }
    }
}
else{}

}

}
else{}

}

}

}
var parameters = "which=" + escape("queryAdvancedExport")+

"&requestedvariables="+encodeURIComponent(requestedvariables)+
"&lowerlimit=" + escape(lowerlimit)+
"&numberofdatapoints=" +
escape(numberofdatapoints)+
"&tempdirectoryname=" + escape(tempdirectoryname);
var method="POST";
var submitto = "../cgi-bin/SCCL/formsubmit.py";
var contenttype = "application/x-www-form-urlencoded";
var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
}
catch(e)
{
    alert(e);
}
}
}
function displayQueryOutput(numberofdatapoints)
{
    var parameters = "which=" + escape("displayQueryOutput") +
"&lowerlimit=" + escape(lowerlimit)+

```

```

        "&numberofdatapoints=" + escape(numberofdatapoints) +
"&advancedexport_total_data_points="+escape(advancedexport_total_data_points)+
        "&thesqlqueryrandomkey=" +
escape(thesqlqueryrandomkey);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function selectAll(abc)
{
    if(abc=="groups")
    {

setTimeout("document.getElementById('subjects').innerHTML=waitmsg;",10);
        if(document.groupsform.allgroups.checked)
            {
                var groupnames='';
                for(var i=0;i<document.groupsform.elements.length;i++)
                    {

if(document.groupsform.elements[i].type=="checkbox")
                    {

document.groupsform.elements[i].checked=true;

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#D1EE
EE";

if((document.groupsform.elements[i].name!="allgroups") &&
(document.groupsform.elements[i].checked==true))
                    {
                        if(groupnames=='')
                            {

groupnames=document.groupsform.elements[i].name
                            }
                        else
                            {
                                groupnames=groupnames+"-
"+document.groupsform.elements[i].name
                            }
                    }
                }
            }
        }
    }
}

removeboxes=['vitalsdataselctform','vitalsvariablesform','hnmrdataselctform','hnmr
mrvariablesform','pnmrdataselctform','pnmrvariablesform','binningdataselctform',
'binningvariablesform','eventindependentvarselctform','datapointsselctform','eve
ntchoiceform','submitrequestform']

removecontent=['subjects','vitalsselect','vitalsvariablesselect','hnmrselect','hnm

```

```

rvariablesselect', 'pnmrselect', 'pnmrvariablesselect', 'binningselect', 'binningvaria
blesselect', 'eventindependentvarselect', 'datapointsselect']
    for(var i=0;i<removeboxes.length;i++)
    {
document.getElementById(removeboxes[i]).style.display="none";
    }
    for(var i=0;i<removecontent.length;i++)
    {
document.getElementById(removecontent[i]).innerHTML=firstmsg;
    }
    loadSubjectsAdvancedExport(groupnames);
    }
    else if(!document.groupsform.allgroups.checked)
    {
    for(var i=0;i<document.groupsform.elements.length;i++)
    {
        if(document.groupsform.elements[i].type=="checkbox")
        {
            document.groupsform.elements[i].checked=false;

if(document.groupsform.elements[i].parentNode.parentNode.className=="oddRow")
            {

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3
F3";

                }
                else
                {

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFF
FF";

                    }
                }
            }
        }
    }
}

removeboxes=['vitalsdataselectform', 'vitalsvariablesform', 'hnmrdataselectform', 'hnm
rvariablesform', 'pnmrdataselectform', 'pnmrvariablesform', 'binningdataselectform',
'binningvariablesform', 'eventindependentvarselectform', 'datapointsselectform', 'eve
ntchoiceform', 'submitrequestform']

removecontent=['subjects', 'vitalsselect', 'vitalsvariablesselect', 'hnmrselect', 'hnm
rvariablesselect', 'pnmrselect', 'pnmrvariablesselect', 'binningselect', 'binningvaria
blesselect', 'eventindependentvarselect', 'datapointsselect']
    for(var i=0;i<removeboxes.length;i++)
    {
document.getElementById(removeboxes[i]).style.display="none";
    }
    for(var i=0;i<removecontent.length;i++)
    {
document.getElementById(removecontent[i]).innerHTML=firstmsg;
    }
    }
    else{}

```

```

    }
    else if(abc=="subjects")
    {
        if(document.subjectsform.allsubjects.checked)
        {
            for(var i=0;i<document.subjectsform.elements.length;i++)
            {
                if(document.subjectsform.elements[i].type=="checkbox")
                {
document.subjectsform.elements[i].checked=true;

document.subjectsform.elements[i].parentNode.parentNode.style.backgroundColor="#D1
EEEE";

                }
                else{}
            }
        }

showboxes=['vitalsdataselctform','vitalsvariablesform','hnmrdataselctform','hnmr
variablesform','pnmrdataselctform','pnmrvariablesform','binningdataselctform','b
inningvariablesform','eventindependentvarselectform','datapointsselectform','event
choiceform','submitrequestform']
        for(var i=0;i<showboxes.length;i++)
        {

document.getElementById(showboxes[i]).style.display="block";

        }

document.getElementById('vitalsselect').innerHTML=selectVitals;
        document.getElementById('hnmrselect').innerHTML=selectHNMR;
        document.getElementById('pnmrselect').innerHTML=selectPNMR;

document.getElementById('binningselect').innerHTML=selectBinning;

//document.getElementById('eventindependentvarselect').innerHTML=selectEventIndepe
ndentVar;
        loadEvIndVarAdvancedExport();

document.getElementById('datapointsselect').innerHTML=selectDatapointFrequency;
        }
        else if(!document.subjectsform.allsubjects.checked)
        {
            for(var i=0;i<document.subjectsform.elements.length;i++)
            {
                if(document.subjectsform.elements[i].type=="checkbox")
                {

document.subjectsform.elements[i].checked=false;

if(document.subjectsform.elements[i].parentNode.parentNode.className=="oddRow")
                {

document.subjectsform.elements[i].parentNode.parentNode.style.backgroundColor="#F3
F3F3";

                }
                else
                {

```

```

document.subjectsform.elements[i].parentNode.parentNode.style.backgroundColor="#FF
FFFF";
    }
    }
    else{}
}

removeboxes=['vitalsdataselectform','vitalsvariablesform','hnmrdataselectform','hn
mrvariablesform','pnmrdataselectform','pnmrvariablesform','binningdataselectform',
'binningvariablesform','eventindependentvarselectform','datapointsselectform','eve
ntchoiceform','submitrequestform']

removecontent=['vitalsselect','vitalsvariablesselect','hnmrselect','hnmrvariables
elect','pnmrselect','pnmrvariablesselect','binningselect','binningvariablesselect'
,'eventindependentvarselect','datapointsselect']
    for(var i=0;i<removeboxes.length;i++)
    {
document.getElementById(removeboxes[i]).style.display="none";
    }
    for(var i=0;i<removecontent.length;i++)
    {
        //alert(removeboxes[i]);

document.getElementById(removecontent[i]).innerHTML=firstmsg;
    }
    }
    else{}
}
    else if(abc=="vitals")
    {

document.getElementById('vitalsvariablesselect').innerHTML=waitmsg;
    if(document.vitalsdataselectform.allvitals.checked)
    {
        var vitalsoptions=''
        for(var
i=0;i<document.vitalsdataselectform.elements.length;i++)
        {

if(document.vitalsdataselectform.elements[i].type=="checkbox")
        {

document.vitalsdataselectform.elements[i].checked=true;

document.vitalsdataselectform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#D1EEEE";

if((document.vitalsdataselectform.elements[i].name!="allvitals") &&
(document.vitalsdataselectform.elements[i].checked==true))
        {
            if(vitalsoptions=='')
            {

vitalsoptions=document.vitalsdataselectform.elements[i].name
            }
            else

```

```

        {
            vidualoptions=vidualoptions+"-
"+document.vidualsdatabselectform.elements[i].name
        }
    }
    else{}
}
else{}
}
loadVidualsVariablesAdvancedExport(vidualoptions);
}
else if(!document.vidualsdatabselectform.allviduals.checked)
{
    for(var
i=0;i<document.vidualsdatabselectform.elements.length;i++)
    {
        if(document.vidualsdatabselectform.elements[i].type=="checkbox")
        {
            document.vidualsdatabselectform.elements[i].checked=false;

            if(document.vidualsdatabselectform.elements[i].parentNode.parentNode.className=="odd
Row")
                {

                    document.vidualsdatabselectform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#F3F3F3";
                }
                else
                {

                    document.vidualsdatabselectform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#FFFFFF";
                }
            }
            else{}
        }
        removeboxes=['vidualsvariablesform']
        removecontent=['vidualsvariablesselect']
        for(var i=0;i<removecontent.length;i++)
        {
            //alert(removeboxes[i]);
        }
        document.getElementById(removecontent[i]).innerHTML=firstmsg;
    }
}
else{}
}
else if(abc=="vidualsvariables")
{
    if(document.vidualsvariablesform.allvidualsvariables.checked)
    {
        for(var
i=0;i<document.vidualsvariablesform.elements.length;i++)
        {
            if(document.vidualsvariablesform.elements[i].type=="checkbox")

```

```

        {
document.vitalsvariablesform.elements[i].checked=true;

document.vitalsvariablesform.elements[i].parentNode.parentNode.style.backgroundCol
or="#D1EEEE";
        }
        else{}
    }
}
else if(!document.vitalsvariablesform.allvitalsvariables.checked)
{
    for(var
i=0;i<document.vitalsvariablesform.elements.length;i++)
    {

if(document.vitalsvariablesform.elements[i].type=="checkbox")
    {

document.vitalsvariablesform.elements[i].checked=false;

if(document.vitalsvariablesform.elements[i].parentNode.parentNode.className=="oddR
ow")
        {

document.vitalsvariablesform.elements[i].parentNode.parentNode.style.backgroundCol
or="#F3F3F3";
        }
        else
        {

document.vitalsvariablesform.elements[i].parentNode.parentNode.style.backgroundCol
or="#FFFFFF";
        }
        }
        else{}
    }
}
else if(abc=="hnmr")
{
    document.getElementById('hnmrvariablesselect').innerHTML=waitmsg;
    if(document.hnmrdatasetform.allhnmr.checked)
    {
        hnmroptions='';
        for(var i=0;i<document.hnmrdatasetform.elements.length;i++)
        {

if(document.hnmrdatasetform.elements[i].type=="checkbox")
    {

document.hnmrdatasetform.elements[i].checked=true;

document.hnmrdatasetform.elements[i].parentNode.parentNode.style.backgroundColo
r="#D1EEEE";

if((document.hnmrdatasetform.elements[i].name!="allhnmr") &&
(document.hnmrdatasetform.elements[i].checked==true))

```

```

        {
            if(hnmroptions=='')
            {
                hnmroptions=document.hnmrdataselectform.elements[i].name
            }
            else
            {
                hnmroptions=hnmroptions+"-
"+document.hnmrdataselectform.elements[i].name
            }
        }
        else{}
    }
}
loadHNMRVariablesAdvancedExport(hnmroptions);
}
else if(!document.hnmrdataselectform.allhnmr.checked)
{
    for(var i=0;i<document.hnmrdataselectform.elements.length;i++)
    {
        if(document.hnmrdataselectform.elements[i].type=="checkbox")
        {
            document.hnmrdataselectform.elements[i].checked=false;

            if(document.hnmrdataselectform.elements[i].parentNode.parentNode.className=="oddRow")
            {
                document.hnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3F3";
            }
            else
            {
                document.hnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFFFF";
            }
        }
        else{}
    }
    removeboxes=['hnmrvariablesform']
    removecontent=['hnmrvariablesselect']
    for(var i=0;i<removecontent.length;i++)
    {
        //alert(removeboxes[i]);

        document.getElementById(removecontent[i]).innerHTML=firstmsg;
    }
}
else{}
}
else if(abc=="hnmrvariables")
{
    if(document.hnmrvariablesform.allhnmrvariables.checked)
    {

```



```

        for(var i=0;i<document.hnmrvariablesform.elements.length;i++)
        {
if(document.hnmrvariablesform.elements[i].type=="checkbox")
        {
document.hnmrvariablesform.elements[i].checked=true;

document.hnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#D1EEEE";
        }
        else{}
        }
}
else if(!document.hnmrvariablesform.allhnmrvariables.checked)
{
    for(var i=0;i<document.hnmrvariablesform.elements.length;i++)
    {

if(document.hnmrvariablesform.elements[i].type=="checkbox")
        {

document.hnmrvariablesform.elements[i].checked=false;

if(document.hnmrvariablesform.elements[i].parentNode.parentNode.className=="oddRow
")
        {

document.hnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#F3F3F3";
        }
        else
        {

document.hnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#FFFFFF";
        }
        }
        else{}
        }
}
else{}
}
else if(abc=="pnmr")
{
    document.getElementById('pnmrvariablesselect').innerHTML=waitmsg;
    if(document.pnmrdataselectform.allpnmr.checked)
    {
        pnmoptions='';
        for(var i=0;i<document.pnmrdataselectform.elements.length;i++)
        {

if(document.pnmrdataselectform.elements[i].type=="checkbox")
        {

document.pnmrdataselectform.elements[i].checked=true;

```

```

document.pnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColo
r="#D1EEEE";

if((document.pnmrdataselectform.elements[i].name!="allpnmr") &&
(document.pnmrdataselectform.elements[i].checked==true))
    {
        if(pnmroptions=='')
            {
                pnmroptions=document.pnmrdataselectform.elements[i].name
            }
        else
            {
                pnmroptions=pnmroptions+"-
"+document.pnmrdataselectform.elements[i].name
            }
    }
    }
    loadPNMRVariablesAdvancedExport(pnmroptions);
}
else if(!document.pnmrdataselectform.allpnmr.checked)
{
    for(var i=0;i<document.pnmrdataselectform.elements.length;i++)
    {
        if(document.pnmrdataselectform.elements[i].type=="checkbox")
            {
                document.pnmrdataselectform.elements[i].checked=false;

                if(document.pnmrdataselectform.elements[i].parentNode.parentNode.className=="oddRo
w")
                    {
                        document.pnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColo
r="#F3F3F3";
                    }
                else
                {
                    document.pnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColo
r="#FFFFFF";
                }
            }
        else{}
    }
    removeboxes=['pnmrvariablesform']
    removecontent=['pnmrvariablesselect']
    for(var i=0;i<removecontent.length;i++)
    {
        //alert(removeboxes[i]);
    }
    document.getElementById(removecontent[i]).innerHTML=firstmsg;
}
else{}

```

```

    }
    else if(abc=="pnmrvariables")
    {
        if(document.pnmrvariablesform.allpnmrvariables.checked)
        {
            for(var i=0;i<document.pnmrvariablesform.elements.length;i++)
            {

if(document.pnmrvariablesform.elements[i].type=="checkbox")
                {

document.pnmrvariablesform.elements[i].checked=true;

document.pnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#D1EEEE";

                    }
                    else{}
                }
            }
        else if(!document.pnmrvariablesform.allpnmrvariables.checked)
        {
            for(var i=0;i<document.pnmrvariablesform.elements.length;i++)
            {

if(document.pnmrvariablesform.elements[i].type=="checkbox")
                {

document.pnmrvariablesform.elements[i].checked=false;

if(document.pnmrvariablesform.elements[i].parentNode.parentNode.className=="oddRow
")
                    {

document.pnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#F3F3F3";

                            }
                            else
                            {

document.pnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#FFFFFF";

                                    }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        else if(abc=="binning")
        {

document.getElementById('binningvariablesselect').innerHTML=waitmsg;
        if(document.binningdataselectform.allbinning.checked)
        {
            binningoptions='';
            for(var
i=0;i<document.binningdataselectform.elements.length;i++)
            {

```

```

if(document.binningdataselectform.elements[i].type=="checkbox")
    {

document.binningdataselectform.elements[i].checked=true;

document.binningdataselectform.elements[i].parentNode.parentNode.style.backgroundC
olor="#D1EEEE";

if((document.binningdataselectform.elements[i].name!="allbinning") &&
(document.binningdataselectform.elements[i].checked==true))
    {
        if(binningoptions=='')
            {

binningoptions=document.binningdataselectform.elements[i].name
                }
            else
                {

binningoptions=binningoptions+"-"+document.binningdataselectform.elements[i].name
                }
            }
        }
        loadBinningVariablesAdvancedExport(binningoptions);
    }
    else if(!document.binningdataselectform.allbinning.checked)
    {
        for(var
i=0;i<document.binningdataselectform.elements.length;i++)
        {

if(document.binningdataselectform.elements[i].type=="checkbox")
            {

document.binningdataselectform.elements[i].checked=false;

if(document.binningdataselectform.elements[i].parentNode.parentNode.className=="od
dRow")
                {

document.binningdataselectform.elements[i].parentNode.parentNode.style.backgroundC
olor="#F3F3F3";
                }
            else
                {

document.binningdataselectform.elements[i].parentNode.parentNode.style.backgroundC
olor="#FFFFFF";
                }
            }
        }
        else{}
    }
    }
    removeboxes=['binningvariablesform']
    removecontent=['binningvariablesselect']
    for(var i=0;i<removecontent.length;i++)
    {

```

```

        //alert(removeboxes[i]);
document.getElementById(removecontent[i]).innerHTML=firstmsg;
    }
    }
    else{}
}
else if(abc=="binningvariables")
{
    if(document.binningvariablesform.allbinningvariables.checked)
    {
        binningoptions='';
        for(var
i=0;i<document.binningvariablesform.elements.length;i++)
        {

if(document.binningvariablesform.elements[i].type=="checkbox")
        {

document.binningvariablesform.elements[i].checked=true;

document.binningvariablesform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#D1EEEE";
        }
        else{}
        }
    }
    else
if(!document.binningvariablesform.allbinningvariables.checked)
    {
        for(var
i=0;i<document.binningvariablesform.elements.length;i++)
        {

if(document.binningvariablesform.elements[i].type=="checkbox")
        {

document.binningvariablesform.elements[i].checked=false;

if(document.binningvariablesform.elements[i].parentNode.parentNode.className=="odd
Row")
        {

document.binningvariablesform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#F3F3F3";
        }
        else
        {

document.binningvariablesform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#FFFFFF";
        }
    }
    }
    else{}
}
}
else{}
}
}

```

```

else if(abc=="eventindependent")
{
if(document.eventindependentvselectform.alleventindependent.checked)
{
eventindependentoptions='';
for(var
i=0;i<document.eventindependentvselectform.elements.length;i++)
{
if(document.eventindependentvselectform.elements[i].type=="checkbox")
{
document.eventindependentvselectform.elements[i].checked=true;

document.eventindependentvselectform.elements[i].parentNode.parentNode.style.backgroundColor="#D1EEEE";
}
else{}
}
}
else
if(!document.eventindependentvselectform.alleventindependent.checked)
{
for(var
i=0;i<document.eventindependentvselectform.elements.length;i++)
{
if(document.eventindependentvselectform.elements[i].type=="checkbox")
{
document.eventindependentvselectform.elements[i].checked=false;

if(document.eventindependentvselectform.elements[i].parentNode.parentNode.className=="oddRow")
{
document.eventindependentvselectform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3F3";
}
else
{
document.eventindependentvselectform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFFFF";
}
}
else{}
}
}
}
else if(abc=="users")
{
if(alladvancedExportPopup.document.savequeryform.allusers.checked)
{
eventindependentoptions='';

```

```

        for(var
i=0;i<alladvancedExportPopup.document.savequeryform.elements.length;i++)
        {

if(alladvancedExportPopup.document.savequeryform.elements[i].type=="checkbox")
        {

alladvancedExportPopup.document.savequeryform.elements[i].checked=true;

alladvancedExportPopup.document.savequeryform.elements[i].parentNode.parentNode.st
yle.backgroundColor="#D1EEEE";
        }
        else{}
        }
    }
    else
if(!alladvancedExportPopup.document.savequeryform.allusers.checked)
    {
        for(var
i=0;i<alladvancedExportPopup.document.savequeryform.elements.length;i++)
        {

if(alladvancedExportPopup.document.savequeryform.elements[i].type=="checkbox")
        {

alladvancedExportPopup.document.savequeryform.elements[i].checked=false;

if(alladvancedExportPopup.document.savequeryform.elements[i].parentNode.parentNode
.className=="oddRow")
                {

alladvancedExportPopup.document.savequeryform.elements[i].parentNode.parentNode.st
yle.backgroundColor="#F3F3F3";
                }
                else
                {

alladvancedExportPopup.document.savequeryform.elements[i].parentNode.parentNode.st
yle.backgroundColor="#FFFFFF";
                }
        }
        else{}
        }
    }
    else{}
}
}
function selectOne(abc)
{
    if(abc=="groups")
    {
        document.getElementById('subjects').innerHTML=waitmsg;
        var checkboxes=0;
        var totalcheckboxes=0;
        var groupnames=''
        for(var i=0;i<document.groupsform.elements.length;i++)
        {

```

```

        if(document.groupsform.elements[i].type=="checkbox")
        {

if((document.groupsform.elements[i].name!="allgroups") &&
(document.groupsform.elements[i].checked==true))
        {
            checkboxes++;

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#D1EE
EE";

                if(groupnames=='')
                {

groupnames=document.groupsform.elements[i].name
                }
                else
                {
                    groupnames=groupnames+"-
"+document.groupsform.elements[i].name
                }
                else
                {

if(document.groupsform.elements[i].parentNode.parentNode.className=="oddRow")
                {

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3
F3";

                }
                else
                {

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFF
FF";

                }
                }
            totalcheckboxes++;
        }
        else{}
    }
    if (checkboxes<(totalcheckboxes-1))
    {
        document.groupsform.allgroups.checked=false;

if(document.groupsform.allgroups.parentNode.parentNode.className=="oddRow")
        {

document.groupsform.allgroups.parentNode.parentNode.style.backgroundColor="#F3F3F3
";

        }
        else
        {

document.groupsform.allgroups.parentNode.parentNode.style.backgroundColor="#FFFFFF
";

        }
    }
}

```



```

        else if ((checkboxes+1)== totalcheckboxes)
        {
            document.groupsform.allgroups.checked=true;
document.groupsform.allgroups.parentNode.parentNode.style.backgroundColor="#D1EEEE
";
        }
        else{}
        if (checkboxes>0)
        {
            loadSubjectsAdvancedExport(groupnames);

removeboxes=['vitalsdataselctform','vitalsvariablesform','hnmrdataselctform','hnmrvariablesform','pnmrdataselctform','pnmrvariablesform','binningdataselctform','binningvariablesform','eventindependentvarselctform','datapointsselectform','eventchoiceform','submitrequestform']

removecontent=['vitalsselect','vitalsvariablesselect','hnmrselect','hnmrvariablesselect','pnmrselect','pnmrvariablesselect','binningselect','binningvariablesselect','eventindependentvarselect','datapointsselect']
            for(var i=0;i<removeboxes.length;i++)
            {
                //alert(removeboxes[i]);

//document.getElementById(removecontent[i]).innerHTML=firstmsg;
document.getElementById(removeboxes[i]).style.display="none";
            }
            for(var i=0;i<removecontent.length;i++)
            {
                //alert(removeboxes[i]);

document.getElementById(removecontent[i]).innerHTML=firstmsg;

//document.getElementById(removeboxes[i]).style.display="none";
            }
        }
        else
        {

removeboxes=['vitalsdataselctform','vitalsvariablesform','hnmrdataselctform','hnmrvariablesform','pnmrdataselctform','pnmrvariablesform','binningdataselctform','binningvariablesform','eventindependentvarselctform','datapointsselectform','eventchoiceform','submitrequestform']

removecontent=['subjects','vitalsselect','vitalsvariablesselect','hnmrselect','hnmrvariablesselect','pnmrselect','pnmrvariablesselect','binningselect','binningvariablesselect','eventindependentvarselect','datapointsselect']
            for(var i=0;i<removeboxes.length;i++)
            {

document.getElementById(removeboxes[i]).style.display="none";
            }
            for(var i=0;i<removecontent.length;i++)
            {
                //alert(removeboxes[i]);

document.getElementById(removecontent[i]).innerHTML=firstmsg;

```

```

        }
    }
}
else if(abc=="subjects")
{
    var checkboxes=0;
    var totalcheckboxes=0;
    var subjectnames=''
    for(var i=0;i<document.subjectsform.elements.length;i++)
    {
        if(document.subjectsform.elements[i].type=="checkbox")
        {
            if((document.subjectsform.elements[i].name!="allsubjects") &&
            (document.subjectsform.elements[i].checked==true))
            {
                checkboxes++;

                document.subjectsform.elements[i].parentNode.parentNode.style.backgroundColor="#D1
                EEEE";

                if(subjectnames=='')
                {
                    subjectnames=document.subjectsform.elements[i].name
                }
                else
                {
                    subjectnames=subjectnames+"-
                "+document.subjectsform.elements[i].name
                }
            }
            else
            {
                if(document.subjectsform.elements[i].parentNode.parentNode.className=="oddRow")
                {
                    document.subjectsform.elements[i].parentNode.parentNode.style.backgroundColor="#F3
                    F3F3";
                }
                else
                {
                    document.subjectsform.elements[i].parentNode.parentNode.style.backgroundColor="#FF
                    FFFF";
                }
            }
            totalcheckboxes++;
        }
        else{}
    }
    if (checkboxes<(totalcheckboxes-1))
    {
        document.subjectsform.allsubjects.checked=false;

        if(document.subjectsform.allsubjects.parentNode.parentNode.className=="oddRow")
        {

```

```

document.subjectsform.allsubjects.parentNode.parentNode.style.backgroundColor="#F3
F3F3";
        }
        else
        {

document.subjectsform.allsubjects.parentNode.parentNode.style.backgroundColor="#FF
FFFF";
                }
        }
        else if ((checkboxes+1)== totalcheckboxes)
        {
            document.subjectsform.allsubjects.checked=true;

document.subjectsform.allsubjects.parentNode.parentNode.style.backgroundColor="#D1
EEEE";
        }
        else{}
        if(checkboxes>0)
        {

showboxes=['vitalsdataselctform','vitalsvariablesform','hnmrdataselctform','hnmr
variablesform','pnmrdataselctform','pnmrvariablesform','binningdataselctform','b
inningvariablesform','eventindependentvarselctform','datapointsselectform','event
choiceform','submitrequestform']
            for(var i=0;i<showboxes.length;i++)
            {

document.getElementById(showboxes[i]).style.display="block";
                }

document.getElementById('vitalsselect').innerHTML=selectVitals;
document.getElementById('hnmrselect').innerHTML=selectHNMR;
document.getElementById('pnmrselect').innerHTML=selectPNMR;
document.getElementById('binningselect').innerHTML=selectBinning;
                loadEvIndVarAdvancedExport();

document.getElementById('datapointsselect').innerHTML=selectDatapointFrequency;

removeboxes=['vitalsvariablesform','hnmrvariablesform','pnmrvariablesform','binnin
gvariablesform']

removecontent=['vitalsvariablesselect','hnmrvariablesselect','pnmrvariablesselect'
,'binningvariablesselect']
            for(var i=0;i<removecontent.length;i++)
            {
                //alert(removeboxes[i]);

document.getElementById(removecontent[i]).innerHTML=firstmsg;

//document.getElementById(removeboxes[i]).style.display="none";
                }
        }
        else

```

```

        {

removeboxes=['vitalsdataselctform','vitalsvariablesform','hnmrdataselctform','hnmrvariablesform','pnmrdataselctform','pnmrvariablesform','binningdataselctform','binningvariablesform','eventindependentvarselctform','datapointsselectform','eventchoiceform','submitrequestform']

removecontent=['vitalsselect','vitalsvariablesselect','hnmrselect','hnmrvariablesselect','pnmrselect','pnmrvariablesselect','binningselect','binningvariablesselect','eventindependentvarselect','datapointsselect']
        for(var i=0;i<removeboxes.length;i++)
        {
            //alert(removeboxes[i]);

//document.getElementById(removecontent[i]).innerHTML=firstmsg;

document.getElementById(removeboxes[i]).style.display="none";
        }
        for(var i=0;i<removecontent.length;i++)
        {
            //alert(removeboxes[i]);

document.getElementById(removecontent[i]).innerHTML=firstmsg;

//document.getElementById(removeboxes[i]).style.display="none";
        }
    }
    else if(abc=="vitals")
    {

document.getElementById('vitalsvariablesselect').innerHTML=waitmsg;
        var checkboxes=0;
        var totalcheckboxes=0;
        var vitalsoptions=''
        for(var i=0;i<document.vitalsdataselctform.elements.length;i++)
        {

if(document.vitalsdataselctform.elements[i].type=="checkbox")
        {

if((document.vitalsdataselctform.elements[i].name!="allvitals") &&
(document.vitalsdataselctform.elements[i].checked==true))
            {
                checkboxes++;

document.vitalsdataselctform.elements[i].parentNode.parentNode.style.backgroundColor="#D1EEEE";

                if(vitalsoptions=='')
                {

vitalsoptions=document.vitalsdataselctform.elements[i].name
                }
                else
                {
                    vitalsoptions=vitalsoptions+"-
"+document.vitalsdataselctform.elements[i].name
                }
            }
        }
    }
}

```

```

        }
        else
        {

if(document.vitalsdataselectform.elements[i].parentNode.parentNode.className=="odd
Row")

        {

document.vitalsdataselectform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#F3F3F3";

        }
        else
        {

document.vitalsdataselectform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#FFFFFF";

        }
        }
        totalcheckboxes++;
    }
    else{}
}
if (checkboxes<(totalcheckboxes-1))
{
    document.vitalsdataselectform.allvitals.checked=false;

if(document.vitalsdataselectform.allvitals.parentNode.parentNode.className=="oddRo
w")

    {

document.vitalsdataselectform.allvitals.parentNode.parentNode.style.backgroundColo
r="#F3F3F3";

    }
    else
    {

document.vitalsdataselectform.allvitals.parentNode.parentNode.style.backgroundColo
r="#FFFFFF";

    }
    }
    else if ((checkboxes+1)== totalcheckboxes)
    {
        document.vitalsdataselectform.allvitals.checked=true;

document.vitalsdataselectform.allvitals.parentNode.parentNode.style.backgroundColo
r="#D1EEEE";

    }
    else{}
    if(checkboxes>0)
    {

//document.getElementById('vitalsvariablesform').style.display="block";
loadVitalsVariablesAdvancedExport(vitalsoptions);

    }
    else
    {
        removeboxes=['vitalsvariablesform']
        removecontent=['vitalsvariablesselect']

```

```

        for(var i=0;i<removecontent.length;i++)
        {
            //alert(removeboxes[i]);
document.getElementById(removecontent[i]).innerHTML=firstmsg;
//document.getElementById(removeboxes[i]).style.display="none";
        }
    }
    else if(abc=="vitalsvariables")
    {
        var checkboxes=0;
        var totalcheckboxes=0;
        var vitalsoptions=''
        for(var i=0;i<document.vitalsvariablesform.elements.length;i++)
        {
            if(document.vitalsvariablesform.elements[i].type=="checkbox")
            {
                if((document.vitalsvariablesform.elements[i].name!="allvitalsvariables") &&
                (document.vitalsvariablesform.elements[i].checked==true))
                {
                    checkboxes++;

document.vitalsvariablesform.elements[i].parentNode.parentNode.style.backgroundCol
or="#D1EEEE";

                    if(vitalsoptions=='')
                    {
                        vitalsoptions=document.vitalsvariablesform.elements[i].name
                    }
                    else
                    {
                        vitalsoptions=vitalsoptions+"-
"+document.vitalsvariablesform.elements[i].name
                    }
                }
                else
                {

            if(document.vitalsvariablesform.elements[i].parentNode.parentNode.className=="oddR
ow")

                {

document.vitalsvariablesform.elements[i].parentNode.parentNode.style.backgroundCol
or="#F3F3F3";

                    }
                    else
                    {

document.vitalsvariablesform.elements[i].parentNode.parentNode.style.backgroundCol
or="#FFFFFF";

                }
            }
            totalcheckboxes++;
        }
    }
}

```

```

        else{}
    }
    if (checkboxes<(totalcheckboxes-1))
    {

document.vitalsvariablesform.allvitalsvariables.checked=false;

if(document.vitalsvariablesform.allvitalsvariables.parentNode.parentNode.className
=="oddRow")
    {

document.vitalsvariablesform.allvitalsvariables.parentNode.parentNode.style.backgr
oundColor="#F3F3F3";
    }
    else
    {

document.vitalsvariablesform.allvitalsvariables.parentNode.parentNode.style.backgr
oundColor="#FFFFFF";
    }
    }
    else if ((checkboxes+1)== totalcheckboxes)
    {

document.vitalsvariablesform.allvitalsvariables.checked=true;

document.vitalsvariablesform.allvitalsvariables.parentNode.parentNode.style.backgr
oundColor="#D1EEEE";
    }
    else{}
}
else if(abc=="hnmr")
{
    document.getElementById('hnmrvariablesselect').innerHTML=waitmsg;
    var checkboxes=0;
    var totalcheckboxes=0;
    var hnmroptions=''
    for(var i=0;i<document.hnmrdataselectform.elements.length;i++)
    {

if(document.hnmrdataselectform.elements[i].type=="checkbox")
    {

if((document.hnmrdataselectform.elements[i].name!="allhnmr") &&
(document.hnmrdataselectform.elements[i].checked==true))
        {
            checkboxes++;

document.hnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColo
r="#D1EEEE";

            if(hnmroptions=='')
            {

hnmroptions=document.hnmrdataselectform.elements[i].name
                }
            else
            {

```

```

                                hnmroptions=hnmroptions+"-
"+document.hnmrdataselectform.elements[i].name
                                }
                                }
                                else
                                {

if(document.hnmrdataselectform.elements[i].parentNode.parentNode.className=="oddRow")
                                {

document.hnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3F3";
                                }
                                else
                                {

document.hnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFFFF";
                                }
                                }
                                totalcheckboxes++;
                                }
                                else{}
                                }
                                if (checkboxes<(totalcheckboxes-1))
                                {
                                document.hnmrdataselectform.allhnmr.checked=false;

if(document.hnmrdataselectform.allhnmr.parentNode.parentNode.className=="oddRow")
                                {

document.hnmrdataselectform.allhnmr.parentNode.parentNode.style.backgroundColor="#F3F3F3";
                                }
                                else
                                {

document.hnmrdataselectform.allhnmr.parentNode.parentNode.style.backgroundColor="#FFFFFF";
                                }
                                }
                                else if ((checkboxes+1)== totalcheckboxes)
                                {
                                document.hnmrdataselectform.allhnmr.checked=true;

document.hnmrdataselectform.allhnmr.parentNode.parentNode.style.backgroundColor="#D1EEEE";
                                }
                                else{}
                                if(checkboxes>0)
                                {
                                loadHNMRVariablesAdvancedExport(hnmroptions);
                                }
                                else
                                {
                                removeboxes=['hnmrvariablesform']
                                removecontent=['hnmrvariablesselect']

```



```

        for(var i=0;i<removecontent.length;i++)
        {
            //alert(removeboxes[i]);
document.getElementById(removecontent[i]).innerHTML=firstmsg;
        }
    }
    else if(abc=="hnmrvariables")
    {
        var checkboxes=0;
        var totalcheckboxes=0;
        var hnmroptions=''
        for(var i=0;i<document.hnmrvariablesform.elements.length;i++)
        {
            if(document.hnmrvariablesform.elements[i].type=="checkbox")
            {
                if((document.hnmrvariablesform.elements[i].name!="allhnmrvariables") &&
                (document.hnmrvariablesform.elements[i].checked==true))
                {
                    checkboxes++;

document.hnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#D1EEEE";
                    if(hnmroptions=='')
                    {
hnmroptions=document.hnmrvariablesform.elements[i].name
                    }
                    else
                    {
                        hnmroptions=hnmroptions+"-
"+document.hnmrvariablesform.elements[i].name
                    }
                }
            }
            else
            {
                if(document.hnmrvariablesform.elements[i].parentNode.parentNode.className=="oddRow
                ")
                {
document.hnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#F3F3F3";
                }
                else
                {
document.hnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#FFFFFF";
                }
            }
        }
        totalcheckboxes++;
    }
}
else{}
}

```

```

        if (checkboxes<(totalcheckboxes-1))
        {
            document.hnmrvariablesform.allhnmrvariables.checked=false;

if(document.hnmrvariablesform.allhnmrvariables.parentNode.parentNode.className=="o
ddRow")
            {

document.hnmrvariablesform.allhnmrvariables.parentNode.parentNode.style.background
Color="#F3F3F3";
            }
            else
            {

document.hnmrvariablesform.allhnmrvariables.parentNode.parentNode.style.background
Color="#FFFFFF";
            }
        }
        else if ((checkboxes+1)== totalcheckboxes)
        {
            document.hnmrvariablesform.allhnmrvariables.checked=true;

document.hnmrvariablesform.allhnmrvariables.parentNode.parentNode.style.background
Color="#D1EEEE";
        }
        else{}
    }
    else if(abc=="pnmr")
    {
        document.getElementById('pnmrvariablesselect').innerHTML=waitmsg;
        var checkboxes=0;
        var totalcheckboxes=0;
        var pnmroptions=''
        for(var i=0;i<document.pnmrdataselectform.elements.length;i++)
        {

if(document.pnmrdataselectform.elements[i].type=="checkbox")
            {

if((document.pnmrdataselectform.elements[i].name!="allpnmr") &&
(document.pnmrdataselectform.elements[i].checked==true))
                {
                    checkboxes++;

document.pnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColo
r="#D1EEEE";

                    if(pnmroptions==' ')
                    {

pnmroptions=document.pnmrdataselectform.elements[i].name
                    }
                    else
                    {
                        pnmroptions=pnmroptions+"-
"+document.pnmrdataselectform.elements[i].name
                    }
                }
            }
        }
    }
    else

```

```

        {
if(document.pnmrdataselectform.elements[i].parentNode.parentNode.className=="oddRow")
        {
document.pnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3F3";
        }
        else
        {
document.pnmrdataselectform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFFFF";
        }
        }
        totalcheckboxes++;
    }
    else{}
}
if (checkboxes<(totalcheckboxes-1))
{
    document.pnmrdataselectform.allpnmr.checked=false;

if(document.pnmrdataselectform.allpnmr.parentNode.parentNode.className=="oddRow")
    {

document.pnmrdataselectform.allpnmr.parentNode.parentNode.style.backgroundColor="#F3F3F3";
    }
    else
    {

document.pnmrdataselectform.allpnmr.parentNode.parentNode.style.backgroundColor="#FFFFFF";
    }
    }
    else if ((checkboxes+1)== totalcheckboxes)
    {
        document.pnmrdataselectform.allpnmr.checked=true;

document.pnmrdataselectform.allpnmr.parentNode.parentNode.style.backgroundColor="#D1EEEE";
    }
    else{}
if(checkboxes>0)
    {
        loadPNMRVariablesAdvancedExport(pnmroptions);
    }
    else
    {
        removeboxes=['pnmrvariablesform']
        removecontent=['pnmrvariablesselect']
        for(var i=0;i<removecontent.length;i++)
        {
            //alert(removeboxes[i]);

document.getElementById(removecontent[i]).innerHTML=firstmsg;

```

```

    }
    }
}
else if(abc=="pnmrvariables")
{
    var checkboxes=0;
    var totalcheckboxes=0;
    var pnmroptions=''
    for(var i=0;i<document.pnmrvariablesform.elements.length;i++)
    {
if(document.pnmrvariablesform.elements[i].type=="checkbox")
    {

if((document.pnmrvariablesform.elements[i].name!="allpnmrvariables") &&
(document.pnmrvariablesform.elements[i].checked==true))
    {
        checkboxes++;

document.pnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#D1EEEE";
                if(pnmroptions=='')
                {

pnmroptions=document.pnmrvariablesform.elements[i].name
                    }
                else
                {
                    pnmroptions=pnmroptions+"-
"+document.pnmrvariablesform.elements[i].name
                }
                }
            else
            {

if(document.pnmrvariablesform.elements[i].parentNode.parentNode.className=="oddRow
")
                {

document.pnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#F3F3F3";
                    }
                else
                {

document.pnmrvariablesform.elements[i].parentNode.parentNode.style.backgroundColor
="#FFFFFF";
                    }
                }
            totalcheckboxes++;
        }
    else{}
    }
}
if (checkboxes<(totalcheckboxes-1))
{
    document.pnmrvariablesform.allpnmrvariables.checked=false;
}

```

```

if(document.pnmrvariablesform.allpnmrvariables.parentNode.parentNode.className=="o
ddRow")
    {
document.pnmrvariablesform.allpnmrvariables.parentNode.parentNode.style.background
Color="#F3F3F3";
    }
    else
    {
document.pnmrvariablesform.allpnmrvariables.parentNode.parentNode.style.background
Color="#FFFFFF";
    }
    }
    else if ((checkboxes+1)== totalcheckboxes)
    {
        document.pnmrvariablesform.allpnmrvariables.checked=true;
document.pnmrvariablesform.allpnmrvariables.parentNode.parentNode.style.background
Color="#D1EEEE";
    }
    else{}
}
else if(abc=="binning")
{
document.getElementById('binningvariablesselect').innerHTML=waitmsg;
    var checkboxes=0;
    var totalcheckboxes=0;
    var binningoptions=''
    for(var i=0;i<document.binningdataselectform.elements.length;i++)
    {
if(document.binningdataselectform.elements[i].type=="checkbox")
    {
if((document.binningdataselectform.elements[i].name!="allbinning") &&
(document.binningdataselectform.elements[i].checked==true))
        {
            checkboxes++;
document.binningdataselectform.elements[i].parentNode.parentNode.style.backgroundC
olor="#D1EEEE";
                if(binningoptions=='')
                {
binningoptions=document.binningdataselectform.elements[i].name
                }
                else
                {
                    binningoptions=binningoptions+"-
"+document.binningdataselectform.elements[i].name
                }
                }
            else
            {

```

```

if(document.binningdataselectform.elements[i].parentNode.parentNode.className=="oddRow")
    {

document.binningdataselectform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3F3";
    }
    else
    {

document.binningdataselectform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFFFF";
    }
    }
    totalcheckboxes++;
    }
    else{}
    }
    if (checkboxes<(totalcheckboxes-1))
    {
        document.binningdataselectform.allbinning.checked=false;

if(document.binningdataselectform.allbinning.parentNode.parentNode.className=="oddRow")
    {

document.binningdataselectform.allbinning.parentNode.parentNode.style.backgroundColor="#F3F3F3";
    }
    else
    {

document.binningdataselectform.allbinning.parentNode.parentNode.style.backgroundColor="#FFFFFF";
    }
    }
    else if ((checkboxes+1)== totalcheckboxes)
    {
        document.binningdataselectform.allbinning.checked=true;

document.binningdataselectform.allbinning.parentNode.parentNode.style.backgroundColor="#D1EEEE";
    }
    else{}
    if(checkboxes>0)
    {

//document.getElementById('binningvariablesform').style.display="block";
        loadBinningVariablesAdvancedExport(binnoptions);
    }
    else
    {
        removeboxes=['binningvariablesform']
        removecontent=['binningvariablesselect']
        for(var i=0;i<removecontent.length;i++)
        {
            //alert(removeboxes[i]);

```

```

document.getElementById(removecontent[i]).innerHTML=firstmsg;

//document.getElementById(removeboxes[i]).style.display="none";
    }
}
else if(abc=="binningvariables")
{
    var checkboxes=0;
    var totalcheckboxes=0;
    var binningoptions=''
    for(var i=0;i<document.binningvariablesform.elements.length;i++)
    {

if(document.binningvariablesform.elements[i].type=="checkbox")
    {

if((document.binningvariablesform.elements[i].name!="allbinningvariables") &&
(document.binningvariablesform.elements[i].checked==true))
    {
        checkboxes++;

document.binningvariablesform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#D1EEEE";

        if(binningoptions=='')
        {

binningoptions=document.binningvariablesform.elements[i].name
        }
        else
        {
            binningoptions=binningoptions+"-
"+document.binningvariablesform.elements[i].name
        }
        }
        else
        {

if(document.binningvariablesform.elements[i].parentNode.parentNode.className=="odd
Row")
    {

document.binningvariablesform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#F3F3F3";

    }
    else
    {

document.binningvariablesform.elements[i].parentNode.parentNode.style.backgroundCo
lor="#FFFFFF";

    }
    }
    totalcheckboxes++;
    }
    else{}
}
if (checkboxes<(totalcheckboxes-1))

```

```

        {
document.binningvariablesform.allbinningvariables.checked=false;

if(document.binningvariablesform.allbinningvariables.parentNode.parentNode.className=="oddRow")
        {
document.binningvariablesform.allbinningvariables.parentNode.parentNode.style.backgroundColor="#F3F3F3";
        }
        else
        {
document.binningvariablesform.allbinningvariables.parentNode.parentNode.style.backgroundColor="#FFFFFF";
        }
        }
        else if ((checkboxes+1)== totalcheckboxes)
        {
document.binningvariablesform.allbinningvariables.checked=true;

document.binningvariablesform.allbinningvariables.parentNode.parentNode.style.backgroundColor="#D1EEEE";
        }
        else{}
    }
    else if(abc=="eventindependent")
    {
        var checkboxes=0;
        var totalcheckboxes=0;
        var eventindependentoptions=''
        for(var
i=0;i<document.eventindependentvarselectform.elements.length;i++)
        {

if(document.eventindependentvarselectform.elements[i].type=="checkbox")
        {

if((document.eventindependentvarselectform.elements[i].name!="alleventindependent"
) && (document.eventindependentvarselectform.elements[i].checked==true))
        {
            checkboxes++;

document.eventindependentvarselectform.elements[i].parentNode.parentNode.style.backgroundColor="#D1EEEE";

            if(eventindependentoptions=='')
            {

eventindependentoptions=document.eventindependentvarselectform.elements[i].name
            }
            else
            {

eventindependentoptions=eventindependentoptions+"-
"+document.eventindependentvarselectform.elements[i].name
            }

```



```

        }
        else
        {

if(document.eventindependentvselectform.elements[i].parentNode.parentNode.className=="oddRow")
        {

document.eventindependentvselectform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3F3";
        }
        else
        {

document.eventindependentvselectform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFFFF";
        }
        }
        totalcheckboxes++;
    }
    else{}
}
if (checkboxes<(totalcheckboxes-1))
{

document.eventindependentvselectform.alleventindependent.checked=false;

if(document.eventindependentvselectform.alleventindependent.parentNode.parentNode.className=="oddRow")
    {

document.eventindependentvselectform.alleventindependent.parentNode.parentNode.style.backgroundColor="#F3F3F3";
    }
    else
    {

document.eventindependentvselectform.alleventindependent.parentNode.parentNode.style.backgroundColor="#FFFFFF";
    }
    }
    else if ((checkboxes+1)== totalcheckboxes)
    {

document.eventindependentvselectform.alleventindependent.checked=true;

document.eventindependentvselectform.alleventindependent.parentNode.parentNode.style.backgroundColor="#D1EEEE";
    }
    else{}
}
//eventnames
else if(abc=="eventnames")
{
    var checkboxes=0;
    var totalcheckboxes=0;
    var eventoptions=''
    for(var i=0;i<document.eventchoiceform.elements.length;i++)

```

```

        {
            if(document.eventchoiceform.elements[i].type=="checkbox")
            {
                if(document.eventchoiceform.elements[i].checked==true)
                {
                    checkboxes++;

                    document.eventchoiceform.elements[i].parentNode.parentNode.style.backgroundColor="
                    #D1EEEE";

                    if(eventoptions=='')
                    {

                        eventoptions=document.eventchoiceform.elements[i].name
                    }
                    else
                    {
                        eventoptions=eventoptions+"-
                    "+document.eventchoiceform.elements[i].name
                    }
                }
                else
                {

                    if(document.eventchoiceform.elements[i].parentNode.parentNode.className=="oddRow")
                    {

                        document.eventchoiceform.elements[i].parentNode.parentNode.style.backgroundColor="
                        #F3F3F3";

                    }
                    else
                    {

                        document.eventchoiceform.elements[i].parentNode.parentNode.style.backgroundColor="
                        #FFFFFF";

                    }
                }
                totalcheckboxes++;
            }
            else{}
        }
    }
    else if(abc=="users")
    {
        var checkboxes=0;
        var totalcheckboxes=0;
        var usersoptions=''
        for(var
i=0;i<alladvancedExportPopup.document.savequeryform.elements.length;i++)
        {

            if(alladvancedExportPopup.document.savequeryform.elements[i].type=="checkbox")
            {

                if((alladvancedExportPopup.document.savequeryform.elements[i].name!="allusers") &&
                (alladvancedExportPopup.document.savequeryform.elements[i].checked==true))
                {
                    checkboxes++;

```

```

alladvancedExportPopup.document.savequeryform.elements[i].parentNode.parentNode.st
yle.backgroundColor="#D1EEEE";
        if(usersoptions=='')
        {
usersoptions=alladvancedExportPopup.document.savequeryform.elements[i].name
        }
        else
        {
            usersoptions=usersoptions+"-
"+alladvancedExportPopup.document.savequeryform.elements[i].name
        }
    }
    else
    {

if(alladvancedExportPopup.document.savequeryform.elements[i].parentNode.parentNode
.className=="oddRow")
        {

alladvancedExportPopup.document.savequeryform.elements[i].parentNode.parentNode.st
yle.backgroundColor="#F3F3F3";
        }
        else
        {

alladvancedExportPopup.document.savequeryform.elements[i].parentNode.parentNode.st
yle.backgroundColor="#FFFFFF";
        }
    }
    totalcheckboxes++;
}
else{}
}
if (checkboxes<(totalcheckboxes-1))
{

alladvancedExportPopup.document.savequeryform.allusers.checked=false;

if(alladvancedExportPopup.document.savequeryform.allusers.parentNode.parentNode.cl
assName=="oddRow")
    {

alladvancedExportPopup.document.savequeryform.allusers.parentNode.parentNode.style
.backgroundColor="#F3F3F3";
    }
    else
    {

alladvancedExportPopup.document.savequeryform.allusers.parentNode.parentNode.style
.backgroundColor="#FFFFFF";
    }
    }
    else if ((checkboxes+1)== totalcheckboxes)
    {

alladvancedExportPopup.document.savequeryform.allusers.checked=true;

```

```

alladvancedExportPopup.document.savequeryform.allusers.parentNode.parentNode.style
.backgroundColor="#D1EEEE";
    }
    else{}
    }
    else{}
}
function advancedExportWindow(){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    if(!window["alladvancedExportPopup"] ||
window["alladvancedExportPopup"].closed){
        alladvancedExportPopup = window.open('../cgi-
bin/SCCL/advancedexportdisplay.py','advancedExportwindow','channelmode=no,fullscre
en=no,toolbar=no,directories=no,status=no,menubar=no,scrollbars=yes,copyhistory=no
,location=no,resizable=yes');
        alladvancedExportPopup.focus();
        alladvancedExportPopup.onload=function(){
            alladvancedExportPopup.document.getElementById('message1').innerHTML =
'Please wait...';
            alladvancedExportPopup.document.getElementById('datatable').innerHTML=
'Please wait...';
        }
    }
    else {
        alladvancedExportPopup.document.getElementById('message1').innerHTML =
'Please wait...';
        alladvancedExportPopup.document.getElementById('datatable').innerHTML=
'Please wait...';
    }
}
queryAdvancedExport(alladvancedExportPopup.document.advanceexportdisplayform.numbe
rofdatapoints.value);
    alladvancedExportPopup.focus();
}
}
function openSaveQueryForm()
{
    $(document).ready(
        function(){
            var
toggleelement=alladvancedExportPopup.document.getElementById("querysaveform");
            //alladvancedExportPopup.jAlert(toggleelement.id)
            $(toggleelement).show("slow");
        });
}
function closeSaveQueryForm()
{
    $(document).ready(
        function(){
            var
toggleelement=alladvancedExportPopup.document.getElementById("querysaveform");
            //alladvancedExportPopup.jAlert(toggleelement.id)
            $(toggleelement).hide('slow');
        });
}
function alladvancedExportPopupShowmsg(elementid,message)
{

```

```

alladvancedExportPopup.document.getElementById(elementid).style.display='block';

alladvancedExportPopup.document.getElementById(elementid).innerHTML=message;
}
function alladvancedExportPopupClearmsg(elementid)
{

alladvancedExportPopup.document.getElementById(elementid).innerHTML='&nbsp;';

alladvancedExportPopup.document.getElementById(elementid).style.display='none';
}
function clearQuerySaveMsg(secs)
{
    var totaltime=secs*1000;
    //setTimeout("alladvancedExportPopupClearmsg('querysavemsg')",totaltime);
    setTimeout("closeSaveQueryForm()",totaltime);
}
function showQueryFormAndIDs()
{
    //var
    thequeryform=alladvancedExportPopup.document.getElementById('showqueryform').inner
HTML;

alladvancedExportPopup.document.getElementById('querysaveform').innerHTML=queryfor
m;

alladvancedExportPopup.document.getElementById("userids").innerHTML=waitmsg;
alladvancedExportPopup.document.savequeryform.queryname.value='';
var parameters = "which=" + escape("loaduserids")+
"&formname="+escape("savequery");
var method="POST";
var submitto = "../cgi-bin/SCCL/formsubmit.py";
var contenttype = "application/x-www-form-urlencoded";
var runthisonstatechange = showContents;
makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
openSaveQueryForm();
alladvancedExportPopup.document.savequeryform.queryname.focus();
}
var thesqlqueryrandomkey='';
function saveQuery()
{
    var
    nameofthequery=alladvancedExportPopup.document.savequeryform.queryname.value;
    var
    descriptionofthequery=alladvancedExportPopup.document.savequeryform.querydescripti
on.value;
    //validateinput();
    if (nameofthequery.length<3)
    {
        alladvancedExportPopup.alert("Name of the query should be atleast
3 characters length");
        return false;
    }
    else{}
    var userids='';
    for(var
j=0;j<alladvancedExportPopup.document.savequeryform.elements.length;j++)

```

```

    {
    if(alladvancedExportPopup.document.savequeryform.elements[j].type=="checkbox" &&
    alladvancedExportPopup.document.savequeryform.elements[j].checked==true)
        {
    if(alladvancedExportPopup.document.savequeryform.elements[j].name!="allusers")
        {
            if (userids=='')
            {
    userids=userids+alladvancedExportPopup.document.savequeryform.elements[j].id;
            }
            else
            {
    userids=userids+':'+alladvancedExportPopup.document.savequeryform.elements[j].id;
            }
            }
            else{}
        }
        else{}
    }
    var parameters = "which=" + escape("savequery") +
        "&formtype="+escape("savequery")+
        "&nameofthequery="+escape(nameofthequery)+
        "&descriptionofthequery="+escape(descriptionofthequery)+
        "&thesqueryrandomkey="+escape(thesqueryrandomkey)+
        "&userids="+escape(userids);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
//Pagination code start
var advancedexport_total_data_points = 0;
var advancedexport_total_pages=0;
var advancedexport_totals_groups=0;
var advancedexport_page_group=0;
var advancedexport_page_number=1;
var advancedexport_previous=0;
var advancedexport_next=0;
function getadvancedexportCount()
{
    var parameters = "which=" + escape("checkadvancedexportCount")+
        "&pig_id=" + escape(document.form1.pig_id.value);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function pagination()
{
    if(window["alladvancedExportPopup"] ||
    !window["alladvancedExportPopup"].closed)

```

```

        {
            var max_number_of_datapoints =
window["alladvancedExportPopup"].document.advanceexportdisplayform.numberofdatapoints.value;
        }
        else{}
        var max_pages_to_show = 5;
        advancedexport_total_pages =
parseInt(advancedexport_total_data_points/max_number_of_datapoints);
        if ((advancedexport_total_data_points%max_number_of_datapoints)>0)
        {
            advancedexport_total_pages = advancedexport_total_pages+1;
        }
        else{}
        advancedexport_totals_groups =
parseInt(advancedexport_total_pages/max_pages_to_show);
        if ((advancedexport_total_pages%max_pages_to_show)>0)
        {
            advancedexport_totals_groups = advancedexport_totals_groups+1;
        }
        else{}
        advancedexport_page_group =
parseInt(advancedexport_page_number/max_pages_to_show);
        if((advancedexport_page_number%max_pages_to_show)==0)
        {
            advancedexport_page_group = advancedexport_page_group-1;
        }
        else{}
        var pageNumbersHtml = '';
        advancedexport_previous = parseInt(advancedexport_page_number)-1;

        if (advancedexport_previous==0)
        {
            advancedexport_previous = 'end';
        }
        else
        {
            var group_of_prev =
parseInt(advancedexport_previous/max_pages_to_show);
            if((advancedexport_previous%max_pages_to_show)==0)
            {
                group_of_prev = group_of_prev-1;
            }
            else{}
        }
        advancedexport_next = parseInt(advancedexport_page_number)+1;
        if (advancedexport_next>advancedexport_total_pages)
        {
            advancedexport_next = 'end';
        }
        else
        {
            var group_of_next =
parseInt(advancedexport_next/max_pages_to_show);
            if((advancedexport_next%max_pages_to_show)==0)
            {
                group_of_next = group_of_next-1;
            }
        }
    }

```

```

        else{}
    }
    if(advancedexport_previous!='end')
    {
        if(window["alladvancedExportPopup"] ||
!window["alladvancedExportPopup"].closed)
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id="\page'+advancedexport_previous+'_group'+group_of_prev+'_prev\'
onclick="javascript:opener.advancedexportchangePage(this.id,this.innerHTML)"
class="advancedexportchangePage">Prev&nbsp;&laquo;&nbsp;&nbsp;&nbsp;&nbsp;</span>&nbsp;&nbsp;&nbsp;&nbsp;'
        }
        else{}
    }
}
else{}
var pageList = new Array;
var page_lower_limit = (advancedexport_page_group*max_pages_to_show)+1;
var page_upper_limit = (advancedexport_page_group+1)*max_pages_to_show;
if (page_upper_limit>advancedexport_total_pages)
{
    page_upper_limit = advancedexport_total_pages;
}
else{}
for(i=page_lower_limit; i<(page_upper_limit+1);i++)
{
    if(i==advancedexport_page_number)
    {
        pageNumbersHtml = pageNumbersHtml+'<span
id="\advancedexport_page'+i+'\'
class="advancedexportselectedPage">'+i+'</span>&nbsp;&nbsp;&nbsp;&nbsp;'
    }
    else
    {
        if(window["alladvancedExportPopup"] ||
!window["alladvancedExportPopup"].closed)
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id="\advancedexport_page'+i+'\'
onclick="javascript:opener.advancedexportchangePage(this.id,this.innerHTML)"
class="advancedexportchangePage">'+i+'</span>&nbsp;&nbsp;&nbsp;&nbsp;'
        }
        else{}
    }
}
if(advancedexport_next!='end')
{
    if(window["alladvancedExportPopup"] ||
!window["alladvancedExportPopup"].closed)
    {
        pageNumbersHtml = pageNumbersHtml+'<span
id="\page'+advancedexport_next+'_group'+group_of_next+'_next\'
onclick="javascript:opener.advancedexportchangePage(this.id,this.innerHTML)"
class="advancedexportchangePage">&raquo;&nbsp;&nbsp;&nbsp;&nbsp;Next</span>'
    }
    else{}
}
}
else{}
}
else{}
}

```



```

var gotopageinnerHtml = '';
var gotopageoptions='';
for(var j=1;j<=advancedexport_total_pages;j++)
{
    if(j==advancedexport_page_number)
    {
        gotopageoptions=gotopageoptions+'<option value=\"'+j+'\"
selected="selected">'+j+'</option>';
    }
    else
    {
        gotopageoptions=gotopageoptions+'<option
value=\"'+j+'\">'+j+'</option>';
    }
}
if(window["alladvancedExportPopup"] ||
!window["alladvancedExportPopup"].closed)
{
    gotopageinnerHtml='Go to Page: <select
id="advancedexport_page_choose" name="advancedexport_page_choose"
onchange="javascript:opener.advancedexportchangePage(this.id,this.value)">'+gotopa
geoptions+'</select>';;

window["alladvancedExportPopup"].document.getElementById('advancedexportOptions').
innerHTML=pageNumbersHtml;

window["alladvancedExportPopup"].document.getElementById('advancedexport_totalrows
').innerHTML='Total Rows: '+advancedexport_total_data_points;

window["alladvancedExportPopup"].document.getElementById('advancedexport_totalpage
s').innerHTML='Total Pages: '+advancedexport_total_pages;

window["alladvancedExportPopup"].document.getElementById('advancedexport_go_to_pag
e').innerHTML=gotopageinnerHtml;
}
else{}
}

function advancedexportchangePage(elementid,elementvalue)
{
    if(window["alladvancedExportPopup"] ||
!window["alladvancedExportPopup"].closed)
    {
        window["alladvancedExportPopup"].document.getElementById('message1').innerHTML =
        'Please wait.....';
        var numberofdatapoints =
window["alladvancedExportPopup"].document.advancedexportdisplayform.numberofdatapoi
nts.value;
    }
    else
    {}
    if (!parseInt(elementvalue))
    {
        var tempsplit = elementid.split("_");
        var abc = tempsplit[0];
        advancedexport_page_number = abc.match(/[\\d]+/g);
    }
}

```

```

        lowerlimit = ((parseInt(advancedexport_page_number))-
1)*numberofdatapoints;
    }
    else
    {
        advancedexport_page_number = parseInt(elementvalue);
        lowerlimit = ((parseInt(elementvalue))-1)*numberofdatapoints;
    }
    displayQueryOutput(numberofdatapoints);
}
//Pagination code end

$(document).ready(
    function(){
        $('.togglediv').toggle(
            function(){ // you can add as much here as you'd like
                //$('#subjects').hide('slow');

                //alert(this.parentNode.parentNode.parentNode.parentNode.className);
                var
toggleelement=$(this).parent().parent().children('.outerdiv1');
                var
togglepairelement=$(this).parent().parent().parent().next().children().children('.
outerdiv1');
                var imageelement=$(this).children()
                $(toggleelement).show('slow');
                $(togglepairelement).show('slow');
                $(imageelement).attr('src', '/images/minus.png');

            }, function() { // same here
                var
toggleelement=$(this).parent().parent().children('.outerdiv1');
                var
togglepairelement=$(this).parent().parent().parent().next().children().children('.
outerdiv1');
                var imageelement=$(this).children()
                $(toggleelement).hide('slow');
                $(togglepairelement).hide('slow');
                $(imageelement).attr('src', '/images/plus.png');

            });
    });

$(document).ready(
    function(){
        $('.togglediv1').toggle(
            function(){ // you can add as much here as you'd like
                //$('#subjects').hide('slow');
                var
toggleelement=$(this).parent().parent().children('.outerdiv1');
                var imageelement=$(this).children()
                $(toggleelement).show('slow');
                $(imageelement).attr('src', '/images/minus.png');

            }, function() { // same here
                var
toggleelement=$(this).parent().parent().children('.outerdiv1');
                var imageelement=$(this).children()

```

```

        $(toggleelement).hide('slow');
        $(imageelement).attr('src', '/images/plus.png');
    });
});
function downloadFile(which,studyname,File_name)
{
    alladvancedExportPopup.location = "/cgi-
bin/SCCL/filedownload.py?which="+which+"&studyname="+studyname+"&File_name="+File_
name+"&tempdirectoryname="+tempdirectoryname;
}
function closePopupWindows(){
    closePopup("alladvancedExportPopup");
}
function closePopup(thePopup){
    if (window[thePopup] && !window[thePopup].closed){
        window[thePopup].close();
    }
}
window.onunload = closePopupWindows;

```

Code written for changepassword.js

```

//Validate and allow submit code start
function validateForm()
{
    document.getElementById('message1').innerHTML='';
    document.getElementById('message1').style.color = "Red";
    document.changepasswordform.newpassword.style.color = "Black";
    document.changepasswordform.reenterpassword.style.color = "Black";
    if(!((document.changepasswordform.newpassword.value).length>6) ||
((document.changepasswordform.newpassword.value).match(/^[^A-Za-z0-9!@#$$%^*\\(\)_-
\?\\<\>,\.\.\/\;\\:\|]+/g)) ||
!((document.changepasswordform.newpassword.value).match(/[A-Za-z]+/g)) ||
!((document.changepasswordform.newpassword.value).match(/[0-9]+/g)))
    {
        document.getElementById('message1').innerHTML= "Error in new
password entered. Please check the fine print";
        document.changepasswordform.newpassword.focus();
        document.changepasswordform.newpassword.style.color = "Red";
        document.cookie = 'jsnewpass=; path=/';
        document.cookie = 'page=changepassword; path=/';
        return false;
    }
    else if((document.changepasswordform.reenterpassword.value) !=
(document.changepasswordform.newpassword.value))
    {
        document.getElementById('message1').innerHTML= "Error in re-
entered password. It did not match the new password field";
        document.changepasswordform.reenterpassword.focus();
        document.changepasswordform.reenterpassword.color = "Red";
        document.cookie = 'jsnewpass=; path=/';
        document.cookie = 'page=changepassword; path=/';
        return false;
    }
}
else

```

```

        {
            document.getElementById('message1').innerHTML='';
            document.getElementById('message1').style.color = "Black";
            document.getElementById('message1').innerHTML='Please wait while
your request is being processed. If the wait time is more than 10sec, submit the
form again';
            document.cookie = 'jsnewpass=Valid; path=/';
            document.cookie = 'page=changepassword; path=/';
            return true;
        }
    }
//Validate and allow submit code end

```

Code written for dataentry.js

```

//Submit form function start
var subjectsArray = new Array;
function onLoadScripts()
{
    storePigIDs();
    document.form1.pig_id.focus();
}
function submitForm(a){
    if (a=="mainform"){
        document.getElementById('message1').innerHTML='Please wait while your
request is being processed. If the wait time is more than 10sec, submit the form
again';
        if((document.form1.pig_id.value).length==6 &&
            document.form1.group.value==' ' &&
            document.form1.weight.value==' '){
            var parameters = "which=" + escape("retrievedata")+
                "&pig_id=" + escape(document.form1.pig_id.value);
            var method="POST";
            var submitto = "../cgi-bin/SCCL/formsubmit.py";
            var contenttype = "application/x-www-form-urlencoded";
            var runthisonstatechange = showContents;
            document.getElementById('pig_id').style.color = 'Black';

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
            //setTimeout("dateofExpday(1)",300);
        }
        else if((document.form1.pig_id.value).length<6 ||
            document.form1.weight.value==' ' ||
            document.form1.group.value==''){
            errorMessage1 = document.getElementById("message1");
            errorMessage1.innerHTML = "Form cannot be submitted if ID, Group
and/or Weight are not entered";
            return false;
        }
        else {
            if (document.form1.lockdata.checked == true)
            {
                lockdata="Not finished"
            }
            else { lockdata="Finished" }
            var parameters = "which=" + escape("mainformsubmit")+

```

```

        "&pig_id=" + escape(document.form1.pig_id.value)+
        "&group=" + escape(document.form1.group.value)+
        "&weight=" + escape(document.form1.weight.value)+
        "&bloodremoved=" +
escape(document.form1.bloodremoved.value)+
        "&orfluids=" + escape(document.form1.orfluids.value)+
        "&bloodgiven=" +
escape(document.form1.bloodgiven.value)+
        "&lrgiven=" + escape(document.form1.lrgiven.value)+
        "&totalvolume=" +
escape(document.form1.totalvolume.value)+
        "&start_month=" +
escape(document.form1.start_month.value)+
        "&start_day=" +
escape(document.form1.start_day.value)+
        "&start_year=" +
escape(document.form1.start_year.value)+
        "&died=" + escape(document.form1.died.value)+
        "&died_hours=" +
escape(document.form1.died_hours.value)+
        "&died_minutes=" +
escape(document.form1.died_minutes.value)+
        "&causeofdeath=" +
encodeURIComponent(document.form1.causeofdeath.value)+
        "&end_month=" +
escape(document.form1.end_month.value)+
        "&end_day=" + escape(document.form1.end_day.value)+
        "&end_year=" + escape(document.form1.end_year.value)+
        "&overallcomments=" +
encodeURIComponent(document.form1.overallcomments.value)+
        "&lockdata=" + encodeURIComponent(lockdata);

    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;

    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    else if(a=="mainform1"){
        document.getElementById('message1').innerHTML = 'Please wait while
your request is being processed. If the wait time is more than 10sec, submit the
form again';
        document.getElementById('message2').innerHTML = '';
        document.getElementById('message3').innerHTML = '';
        document.getElementById('message4').innerHTML = '';
        if(document.form1.pig_id.value==' ' ||
document.form1.weight.value==' ' ||
document.form1.start_month.value==' ' ||
document.form1.start_day.value==' ' ||
document.form1.start_year.value==''){
            errorMessage1 = document.getElementById("message1");
            errorMessage1.innerHTML = "Form cannot be submitted if ID, Weight or
Experiment Start Date are not entered";
        }
        else{
            if (document.form1.lockdata.checked == true)

```

```

        {
            lockdata = "Not finished"
        }
    else { lockdata = "Finished" }
    var parameters = "which=" + escape("mainform1submit")+
        "&pig_id=" + escape(document.form1.pig_id.value)+
        "&group=" + escape(document.form1.group.value)+
        "&weight=" + escape(document.form1.weight.value)+
        "&bloodremoved="+
escape(document.form1.bloodremoved.value)+
        "&orfluids="+
escape(document.form1.orfluids.value)+
        "&bloodgiven="+
escape(document.form1.bloodgiven.value)+
        "&lrgiven=" + escape(document.form1.lrgiven.value)+
        "&totalvolume="+
escape(document.form1.totalvolume.value)+
        "&start_month=" +
escape(document.form1.start_month.value)+
        "&start_day=" +
escape(document.form1.start_day.value)+
        "&start_year=" +
escape(document.form1.start_year.value)+
        "&died=" + escape(document.form1.died.value)+
        "&died_hours=" +
escape(document.form1.died_hours.value)+
        "&died_minutes=" +
escape(document.form1.died_minutes.value)+
        "&causeofdeath=" +
encodeURIComponent(document.form1.causeofdeath.value)+
        "&end_month=" +
escape(document.form1.end_month.value)+
        "&end_day=" +
escape(document.form1.end_day.value)+
        "&end_year=" +
escape(document.form1.end_year.value)+
        "&overallcomments=" +
encodeURIComponent(document.form1.overallcomments.value)+
        "&lockdata=" + encodeURIComponent(lockdata);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;

    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
}
else if(a=="vitalsform"){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = 'Please wait while
your request is being processed. If the wait time is more than 10sec, submit the
form again';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    vitals_page_group=0;
    vitals_page_number=1;
    vitals_previous=0;
    vitals_next=0;
}

```

```

        if(!window["allVitalsPopup"] || window["allVitalsPopup"].closed)
        {
            document.getElementById('message2').innerHTML = 'Please
wait.....';
            document.getElementById("vitalsdataorder").value="Descending";
        }
        else
        {
            window["allVitalsPopup"].document.getElementById('message2').innerHTML = 'Please
wait.....';

            window["allVitalsPopup"].document.getElementById("vitalsdataorder").value="Descend
ing";
        }
        if (document.form1.submitidentifiers.value=='mainform' ||
            document.form2.hours.value==' ' ||
            document.form2.minutes.value==''){
            document.getElementById('message2').innerHTML="Document cannot be
submitted without time and without the primary identifier values";
        }
        else{
            document.getElementById('message2').innerHTML='Please wait while
your request is being processed. If the wait time is more than 10sec, submit the
form again';

            var exp_year = document.form1.start_year.value;
            var exp_month = document.form1.start_month.value;
            var exp_day = document.form1.start_day.value;
            var the_date1 = new Date(exp_year,exp_month-1,exp_day);
            var num_of_days = (parseInt(document.form2.day.value))-1;
            the_date1.setDate(the_date1.getDate()+num_of_days);
            var year_exp = the_date1.getFullYear();
            var month_exp = the_date1.getMonth()+1;
            var day_exp = the_date1.getDate();
            var date_exp = year_exp.toString()+"-"+month_exp.toString()+"-
"+day_exp.toString();
            var checkfields = 0;
            var onehourvitals = 0;
            if(document.form2.heartrate.value!=''){
                checkfields++
            }
            else{}
            if(document.form2.artsys.value!=''){
                checkfields++;
            }
            else{}
            if(document.form2.artdia.value!=''){
                checkfields++;
            }
            else{}
            if(document.form2.pasys.value!=''){
                checkfields++;
            }
            else{}
            if(document.form2.padia.value!=''){
                checkfields++;
            }
            else{}
        }
    }
}

```

```

if(document.form2.wedgepressure.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.urineoutput.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.temperature.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.cardiacoutput.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.sto2_multidepth.value!=''){
    checkfields++;
}
else{}
if(document.form2.thi_multidepth.value!=''){
    checkfields++;
}
else{}
if(document.form2.fio2.value!=''){
    checkfields++;
}
else{}
if(document.form2.propofol.value!=''){
    checkfields++;
}
else{}
if(document.form2.bis.value!=''){
    checkfields++;
}
else{}
if(document.form2.chesttubeoutput.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.sto2_15mm.value!=''){
    checkfields++;
}
else{}
if(document.form2.thi_15mm.value!=''){
    checkfields++;
}
else{}
if(document.form2.arterialph.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}

```



```

if(document.form2.art_pco2.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.art_po2.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.art_o2_sat.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.ven_po2.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.ven_o2_sat.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.hemoglobin.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.sodium.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.potassium.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.calcium.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.lactate.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.glucose.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.basedeficit.value!=''){
    checkfields++;
}
}

```

```

else{}
if(document.form2.bladderpres.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.eventcomments.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if(document.form2.specificgravity.value!=''){
    checkfields++;
    onehourvitals++;
}
else{}
if (checkfields > 0){
    var parameters = "which=" + escape("vitalsformsubmit")+
        "&pig_id=" +
escape(document.form1.pig_id.value)+
        "&onehourvitals=" + escape(onehourvitals)+
        "&day=" + escape(document.form2.day.value)+
        "&date=" + escape(date_exp)+
        "&hours=" +
escape(document.form2.hours.value)+
        "&minutes=" +
escape(document.form2.minutes.value)+
        "&event=" +
escape(document.form2.event.value)+
        "&heartrate=" +
escape(document.form2.heartrate.value)+
        "&artsys=" +
escape(document.form2.artsys.value)+
        "&artdia=" +
escape(document.form2.artdia.value)+
        "&pasys=" +
escape(document.form2.pasys.value)+
        "&padia=" +
escape(document.form2.padia.value)+
        "&wedgepressure=" +
escape(document.form2.wedgepressure.value)+
        "&urineoutput=" +
escape(document.form2.urineoutput.value)+
        "&temperature=" +
escape(document.form2.temperature.value)+
        "&cardiacoutput=" +
escape(document.form2.cardiacoutput.value)+
        "&sto2_multidepth=" +
escape(document.form2.sto2_multidepth.value)+
        "&thi_multidepth=" +
escape(document.form2.thi_multidepth.value)+
        "&fio2=" + escape(document.form2.fio2.value)+
        "&propofol=" +
escape(document.form2.propofol.value)+
        "&bis=" + escape(document.form2.bis.value)+
        "&chesttubeoutput=" +
escape(document.form2.chesttubeoutput.value)+

```

```

                                "&sto2_15mm=" +
escape(document.form2.sto2_15mm.value)+
                                "&thi_15mm=" +
escape(document.form2.thi_15mm.value)+
                                "&arterialph=" +
escape(document.form2.arterialph.value)+
                                "&art_pco2=" +
escape(document.form2.art_pco2.value)+
                                "&art_po2=" +
escape(document.form2.art_po2.value)+
                                "&art_o2_sat=" +
escape(document.form2.art_o2_sat.value)+
                                "&ven_po2=" +
escape(document.form2.ven_po2.value)+
                                "&ven_o2_sat=" +
escape(document.form2.ven_o2_sat.value)+
                                "&hemoglobin=" +
escape(document.form2.hemoglobin.value)+
                                "&sodium=" +
escape(document.form2.sodium.value)+
                                "&potassium=" +
escape(document.form2.potassium.value)+
                                "&calcium=" +
escape(document.form2.calcium.value)+
                                "&lactate=" +
escape(document.form2.lactate.value)+
                                "&glucose=" +
escape(document.form2.glucose.value)+
                                "&basedeficit=" +
escape(document.form2.basedeficit.value)+
                                "&bladderpres=" +
escape(document.form2.bladderpres.value)+
                                "&eventcomments=" +
encodeURIComponent(document.form2.eventcomments.value)+
                                "&specificgravity=" +
encodeURIComponent(document.form2.specificgravity.value);
                                var method="POST";
                                var submitto = "../cgi-bin/SCCL/formssubmit.py";
                                var contenttype = "application/x-www-form-urlencoded";
                                var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
                                }
                                else{
                                document.getElementById('message2').innerHTML = "Fill at least
one field after event field before submitting the data"
                                }
                                }
                                }
                                else if(a=="labsform")
                                {
                                document.getElementById('message1').innerHTML = '';
                                document.getElementById('message2').innerHTML = '';
                                document.getElementById('message3').innerHTML = 'Please wait while
your request is being processed. If the wait time is more than 10sec, submit the
form again';
                                document.getElementById('message4').innerHTML = '';
                                if (document.form1.submitidentifiers.value=='mainform'){

```

```

        document.getElementById('message3').innerHTML="Document cannot be
submitted without the primary identifier values";
    }
    else{
        document.getElementById('message3').innerHTML='Please wait while
your request is being processed. If the wait time is more than 10sec, submit the
form again';
        var Pig_ID = document.form1.pig_id.value;
        var Event_name = document.form4.labsevent.value;
        var Bilirubin_total = document.form4.Bilirubin_total.value;
        var Albumin = document.form4.Albumin.value;
        var Protein_total = document.form4.Protein_total.value;
        var Alk_phosphatase = document.form4.Alk_phosphatase.value;
        var ALT = document.form4.ALT.value;
        var AST = document.form4.AST.value;
        var Urea_nitrogen = document.form4.Urea_nitrogen.value;
        var CK = document.form4.CK.value;
        var Creatinine = document.form4.Creatinine.value;
        var LD = document.form4.LD.value;
        var Platelet_count = document.form4.Platelet_count.value;
        var Urine_osmolality = document.form4.Urine_osmolality.value;
        var Urine_creatinine = document.form4.Urine_creatinine.value;
        var checklabfields = 0;
        if(Bilirubin_total!=''){
            checklabfields++
        }
        else{}
        if(Albumin!=''){
            checklabfields++;
        }
        else{}
        if(Protein_total!=''){
            checklabfields++;
        }
        else{}
        if(Alk_phosphatase!=''){
            checklabfields++;
        }
        else{}
        if(ALT!=''){
            checklabfields++;
        }
        else{}
        if(AST!=''){
            checklabfields++;
        }
        else{}
        if(Urea_nitrogen!=''){
            checklabfields++;
        }
        else{}
        if(CK!=''){
            checklabfields++;
        }
        else{}
        if(Creatinine!=''){
            checklabfields++;
        }
    }

```

```

else{}
if(LD!=''){
    checklabfields++;
}
else{}
if(Platelet_count!=''){
    checklabfields++;
}
else{}
if(Urine_osmolality!=''){
    checklabfields++;
}
else{}
if(Urine_creatinine!=''){
    checklabfields++;
}
else{}
if (checklabfields > 0){
    var parameters = "which=" + escape("labsformsubmit")+
        "&Pig_ID=" + escape(Pig_ID)+
        "&Event_name=" + escape(Event_name)+
        "&Bilirubin_total="+ escape(Bilirubin_total)+
        "&Albumin=" + escape(Albumin)+
        "&Protein_total=" + escape(Protein_total)+
        "&Alk_phosphatase=" + escape(Alk_phosphatase)+
        "&ALT=" + escape(ALT)+
        "&AST=" + escape(AST)+
        "&Urea_nitrogen=" + escape(Urea_nitrogen)+
        "&CK=" + escape(CK)+
        "&Creatinine=" + escape(Creatinine)+
        "&LD=" + escape(LD)+
        "&Platelet_count=" + escape(Platelet_count)+
        "&Urine_osmolality=" +
escape(Urine_osmolality)+
        "&Urine_creatinine=" +
escape(Urine_creatinine);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
else{
    document.getElementById('message3').innerHTML = "Fill at least
one field after event field before submitting the data"
}
}
}
else if(a=="gcsscoresform")
{
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = 'Please wait while
your request is being processed. If the wait time is more than 10sec, submit the
form again';
    if (document.form1.submitidentifiers.value=='mainform'){

```

```

        document.getElementById('message4').innerHTML="Document cannot be
submitted without the primary identifier values";
    }
    else{
        var Pig_ID = document.form1.pig_id.value;
        var gcs1 = document.form6.gcs1.value;
        var gcs23 = document.form6.gcs23.value;
        var gcs24 = document.form6.gcs24.value;
        var gcs25 = document.form6.gcs25.value;
        var checkgcsfields = 0;
        if(gcs1!=''){
            checkgcsfields++
        }
        else{}
        if(gcs23!=''){
            checkgcsfields++;
        }
        else{}
        if(gcs24!=''){
            checkgcsfields++;
        }
        else{}
        if(gcs25!=''){
            checkgcsfields++;
        }
        else{}
        if (checkgcsfields > 0){
            var parameters = "which=" + escape("gcsscoresformsubmit")+
                "&Pig_ID=" + escape(Pig_ID)+
                "&gcs1=" + escape(gcs1)+
                "&gcs23="+ escape(gcs23)+
                "&gcs24=" + escape(gcs24)+
                "&gcs25=" + escape(gcs25);
            var method="POST";
            var submitto = "../cgi-bin/SCCL/formsubmit.py";
            var contenttype ="application/x-www-form-urlencoded";
            var runthisonstatechange = showContents;

            makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
        }
        else{
            document.getElementById('message4').innerHTML = "Fill at least
one field before submitting the data"
        }
    }
}
else
{
    document.getElementById('message1').innerHTML="There was an error.
Please contact datamanager";
}
}
//Submit form function end
function editform(){
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    var vitalsdisplay = document.getElementById("vitalsform").style;

```

```

vitalsdisplay.display = "none";
var allvitalsdisplay = document.getElementById("vitalsdataholder").style;
allvitalsdisplay.display = "none";
document.form1.selectform.value="none";
disableSelectform();
var givealert = document.getElementById("message1");
givealert.innerHTML = "Please be aware that ID and Experiment Start Date
cannot be edited.\
                                If you want to change any of those, please contact
data manager.\
                                If you have entered the ID incorrectly, you can
continue adding the data to it.\
                                This can be changed by your datamanger later without
any loss of data";
if(document.form1.group.getAttribute('disabled')){
    document.form1.group.removeAttribute('disabled');
}
else{}
if(document.form1.weight.getAttribute('disabled')){
    document.form1.weight.removeAttribute('disabled');
}
else{}
if(document.form1.bloodremoved.getAttribute('disabled')){
    document.form1.bloodremoved.removeAttribute('disabled');
}
else{}
if(document.form1.orfluids.getAttribute('disabled')){
    document.form1.orfluids.removeAttribute('disabled');
}
else{}
if(document.form1.bloodgiven.getAttribute('disabled')){
    document.form1.bloodgiven.removeAttribute('disabled');
}
else{}
if(document.form1.lrgiven.getAttribute('disabled')){
    document.form1.lrgiven.removeAttribute('disabled');
}
else{}
if(document.form1.totalvolume.getAttribute('disabled')){
    document.form1.totalvolume.removeAttribute('disabled');
}
else{}
if(document.form1.died.getAttribute('disabled')){
    document.form1.died.removeAttribute('disabled');
}
else{}
if(document.form1.end_month.getAttribute('disabled')){
    document.form1.end_month.removeAttribute('disabled');
}
else{}
if(document.form1.end_day.getAttribute('disabled')){
    document.form1.end_day.removeAttribute('disabled');
}
else{}
if(document.form1.end_year.getAttribute('disabled')){
    document.form1.end_year.removeAttribute('disabled');
}
else{}

```

```

if(document.form1.died_hours.getAttribute('disabled')){
    document.form1.died_hours.removeAttribute('disabled');
}
else{}
if(document.form1.died_minutes.getAttribute('disabled')){
    document.form1.died_minutes.removeAttribute('disabled');
}
else{}
if(document.form1.causeofdeath.getAttribute('disabled')){
    document.form1.causeofdeath.removeAttribute('disabled');
}
else{}
if(document.form1.overallcomments.getAttribute('disabled')){
    document.form1.overallcomments.removeAttribute('disabled');
}
else { }
var Usergroup = '';
checkGroupEnableLock();
}
function checkGroupEnableLock()
{
    var parameters = "which=" + escape("checkgroupenabledatalock");
    var method = "POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters, method, submitto, contenttype,
runthisonstatechange);
}
function idcheck(theElement,theid,fieldname){
    //checkInt(theElement,theid,fieldname);
    if((document.form1.pig_id.value).length==6 &&
checkInt(theElement,theid,fieldname)==true){
        document.getElementById('message1').innerHTML='';
        var parameters = "which=" + escape("idcheck")+
            "&pig_id=" + escape(document.form1.pig_id.value);
        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype ="application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
        setTimeout("document.getElementById('holdsuggestions').innerHTML =
'';",1600);
    }
    else
    {
        document.getElementById('message1').innerHTML = 'The ID field must be
an integer and of 6 digits length';
        setTimeout("document.getElementById('holdsuggestions').innerHTML =
'';",1600);
    }
}
function duplicateid(){
    document.getElementById('message1').innerHTML='Identifier with similar ID
already exists. Click submit to view the data or else contact data manager for any
queries';
    document.form1.start_month.value='';
    document.form1.start_day.value='';
}

```



```

        document.form1.start_year.value='';
    }
function startDatecreate(){
    var Pig_ID = document.form1.pig_id.value;
    var start_month = Pig_ID.substring(0,2);
    var start_day = Pig_ID.substring(2,4);
    var start_year = "20"+Pig_ID.substring(4,6);
    document.form1.start_month.value = start_month;
    document.form1.start_day.value = start_day;
    document.form1.start_year.value = start_year;
}
function pageReload(){
    window.location.reload(true);
}
function disableform1(){
    document.form1.pig_id.setAttribute('disabled','disabled');
    document.form1.group.setAttribute('disabled','disabled');
    document.form1.weight.setAttribute('disabled','disabled');
    document.form1.bloodremoved.setAttribute('disabled','disabled');
    document.form1.orfluids.setAttribute('disabled','disabled');
    document.form1.bloodgiven.setAttribute('disabled','disabled');
    document.form1.lrgiven.setAttribute('disabled','disabled');
    document.form1.totalvolume.setAttribute('disabled','disabled');
    document.form1.start_month.setAttribute('disabled','disabled');
    document.form1.start_day.setAttribute('disabled','disabled');
    document.form1.start_year.setAttribute('disabled','disabled');
    document.form1.end_month.setAttribute('disabled','disabled');
    document.form1.end_day.setAttribute('disabled','disabled');
    document.form1.end_year.setAttribute('disabled','disabled');
    document.form1.died.setAttribute('disabled','disabled');
    document.form1.died_hours.setAttribute('disabled','disabled');
    document.form1.died_minutes.setAttribute('disabled','disabled');
    document.form1.causeofdeath.setAttribute('disabled','disabled');
    document.form1.overallcomments.setAttribute('disabled','disabled');
    document.form1.lockdata.setAttribute('disabled','disabled');
}
function changeOnClick()
{
    var test = document.getElementById("buttonsholder").innerHTML;
    var newattribute = test.replace(/mainform/g,"mainform1");
    document.getElementById("buttonsholder").innerHTML = newattribute;
}
function formSelect(b)
{
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    if (b=="vitals")
    {
        var vitalsdisplay = document.getElementById("vitalsform").style;
        vitalsdisplay.display = "inline";
        var labsdisplay = document.getElementById("labsform").style;
        labsdisplay.display = "none";
        var alllabsdisplay =
document.getElementById("labsdataholder").style;
        alllabsdisplay.display = "none";
        var gcsscoresdisplay = document.getElementById("gcsscores").style;

```

```

        gcsscoresdisplay.display = "none";
        var allgcsscoresdisplay =
document.getElementById("gcsdataholder").style;
        allgcsscoresdisplay.display = "none";
    }
    else if (b=="labs")
    {
        var labsdisplay = document.getElementById("labsform").style;
        labsdisplay.display = "inline";
        var vitalsdisplay = document.getElementById("vitalsform").style;
        vitalsdisplay.display = "none";
        var allvitalsdisplay =
document.getElementById("vitalsdataholder").style;
        allvitalsdisplay.display = "none";
        var gcsscoresdisplay = document.getElementById("gcsscores").style;
        gcsscoresdisplay.display = "none";
        var allgcsscoresdisplay =
document.getElementById("gcsdataholder").style;
        allgcsscoresdisplay.display = "none";
    }
    else if (b=="gcsscores")
    {
        var gcsscoresdisplay = document.getElementById("gcsscores").style;
        gcsscoresdisplay.display = "inline";
        var vitalsdisplay = document.getElementById("vitalsform").style;
        vitalsdisplay.display = "none";
        var allvitalsdisplay =
document.getElementById("vitalsdataholder").style;
        allvitalsdisplay.display = "none";
        var labsdisplay = document.getElementById("labsform").style;
        labsdisplay.display = "none";
        var alllabsdisplay =
document.getElementById("labsdataholder").style;
        alllabsdisplay.display = "none";
    }
    else if (b=="all")
    {
        var vitalsdisplay = document.getElementById("vitalsform").style;
        vitalsdisplay.display = "inline";
        var labsdisplay = document.getElementById("labsform").style;
        labsdisplay.display = "inline";
        var gcsscoresdisplay = document.getElementById("gcsscores").style;
        gcsscoresdisplay.display = "inline";
    }
    else if (b=="none")
    {
        var vitalsdisplay = document.getElementById("vitalsform").style;
        vitalsdisplay.display = "none";
        var allvitalsdisplay =
document.getElementById("vitalsdataholder").style;
        allvitalsdisplay.display = "none";
        var labsdisplay = document.getElementById("labsform").style;
        labsdisplay.display = "none";
        var alllabsdisplay =
document.getElementById("labsdataholder").style;
        alllabsdisplay.display = "none";
        var gcsscoresdisplay = document.getElementById("gcsscores").style;
        gcsscoresdisplay.display = "none";
    }

```

```

        var allgcsscoresdisplay =
document.getElementById("gcsdataholder").style;
        allgcsscoresdisplay.display = "none";
    }
}
function enableSelectform(){
    if(document.form1.selectform.getAttribute('disabled')){
        document.form1.selectform.removeAttribute('disabled');
    }
    else{}
}
function disableSelectform(){
    if(document.form1.selectform.getAttribute('disabled')){

    }
    else{
        document.form1.selectform.setAttribute('disabled','disabled');
    }
}

```

```

/*****
*****/

```

Form element numbering in Safari browser is different from others. It won't count the fieldset as an element

So, when you try to loop using form elements for the below mentioned purpose, you will land in problem

So, try to use the names of the elements and write all them down even though it takes time, to make sure that

your code is cross browser compatible. Wherever you see this kind of flat coding, the only reason for that is

to make it compatible with Safari. Hope they will change it soon.

```

*****/
*****/

```

```

function vitalsRefresh(){
    document.form2.event.value = '99';
    document.form2.hours.value = '';
    document.form2.minutes.value = '';
    document.form2.heartrate.value = '';
    document.form2.artsys.value = '';
    document.form2.artdia.value = '';
    document.form2.pasys.value = '';
    document.form2.padia.value = '';
    document.form2.wedgepressure.value = '';
    document.form2.urineoutput.value = '';
    document.form2.temperature.value = '';
    document.form2.cardiacoutput.value = '';
    document.form2.sto2_multidepth.value = '';
    document.form2.thi_multidepth.value = '';
    document.form2.fio2.value = '';
    document.form2.propofol.value = '';
    document.form2.bis.value = '';
    document.form2.chesttubeoutput.value = '';
    document.form2.sto2_15mm.value = '';
    document.form2.thi_15mm.value = '';
    document.form2.arterialph.value = '';
    document.form2.art_pco2.value = '';
    document.form2.art_po2.value = '';
}

```

```

document.form2.art_02_sat.value = '';
document.form2.ven_po2.value = '';
document.form2.ven_o2_sat.value = '';
document.form2.hemoglobin.value = '';
document.form2.sodium.value = '';
document.form2.potassium.value = '';
document.form2.calcium.value = '';
document.form2.lactate.value = '';
document.form2.glucose.value = '';
document.form2.basedeficit.value = '';
document.form2.bladderpres.value = '';
document.form2.eventcomments.value = '';
document.form2.specificgravity.value = '';
//document.form2.day.focus();
//Use the above in case of future problems with date was entered by mistake
because of this selection
document.form2.hours.focus();
}
function labsRefresh(){
document.form4.Bilirubin_total.value = '';
document.form4.Albumin.value = '';
document.form4.Protein_total.value = '';
document.form4.Alk_phosphatase.value = '';
document.form4.ALT.value = '';
document.form4.AST.value = '';
document.form4.Urea_nitrogen.value = '';
document.form4.CK.value = '';
document.form4.Creatinine.value = '';
document.form4.LD.value = '';
document.form4.Platelet_count.value = '';
document.form4.Urine_osmolality.value = '';
document.form4.Urine_creatinine.value = '';
document.form4.labsevent.focus();
}
function gcscscoresRefresh(){
document.form6.gcs1.value = '';
document.form6.gcs23.value = '';
document.form6.gcs24.value = '';
document.form6.gcs25.value = '';
}
function checkInt(theElement,theid,fieldname){
var elementVal = theElement.value;
if(elementVal.match(/^[^0-9]+/g)){
alert("The field "+fieldname+" only accepts integers");
setTimeout("document.getElementById(\""+theid+"\").focus();",1);
theElement.style.color = "Red";
return false;
}
else{
document.getElementById(theid).style.color = "Black";
return true;
}
}
function requestAllVitalsNewWindow(){
if(!window["allVitalsPopup"] || window["allVitalsPopup"].closed){}
else{
window["allVitalsPopup"].document.getElementById("message2").innerHTML='';
vitals_page_group=0;
}
}

```

```

        vitals_page_number=1;
        vitals_previous=0;
        vitals_next=0;
        var vitalsdataorder =
window["allVitalsPopup"].document.getElementById("vitalsdataorder").value;
        numberofdatapoints =
window["allVitalsPopup"].document.getElementById('numberofdatapoints').value;
        var lowerlimit = 0;
        var parameters = "which=" + escape("viewallvitalsnewwindow")+"&pig_id=" +
escape(document.form1.pig_id.value)+
                "&vitalsdataorder=" + escape(vitalsdataorder)+
                "&lowerlimit=" + escape((lowerlimit))+
                "&numberofdatapoints=" + escape(numberofdatapoints);
        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype ="application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
}
function requestAllLabsNewWindow(){
    if(!window["allLabsPopup"] || window["allLabsPopup"].closed){}
    else{
        window["allLabsPopup"].document.getElementById("message2").innerHTML='';
        window["allLabsPopup"].document.getElementById("message3").innerHTML='';
        var parameters = "which=" + escape("viewalllabsnewwindow")+
                "&pig_id=" + escape(document.form1.pig_id.value);
        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype ="application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
}
function requestAllVitals(){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = 'Please wait....';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    vitals_page_group=0;
    vitals_page_number=1;
    vitals_previous=0;
    vitals_next=0;
    numberofdatapoints = document.form3.numberofdatapoints.value;
    var vitalsdataorder=document.form3.vitalsdataorder.value;
    var lowerlimit = 0;
    if(!window["allVitalsPopup"] || window["allVitalsPopup"].closed){}
    else{window["allVitalsPopup"].close();}
    var allvitalsdisplay = document.getElementById("vitalsdataholder").style;
    if(allvitalsdisplay.display=="inline"){
        //var parameters = "which=" + escape("viewallvitals")+"&pig_id=" +
escape(document.form1.pig_id.value);
        allvitalsdisplay.display="none";
    }
    else{
        allvitalsdisplay.display = "inline";
        var parameters = "which=" + escape("viewallvitals")+
                "&pig_id=" + escape(document.form1.pig_id.value)+

```

```

        "&vitalsdataorder=" + escape(vitalsdataorder)+
        "&lowerlimit=" + escape((lowerlimit))+
        "&numberofdatapoints=" + escape(numberofdatapoints);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
}
function showAllVitals(){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    vitals_page_group=0;
    vitals_page_number=1;
    vitals_previous=0;
    vitals_next=0;
    numberofdatapoints = document.getElementById('numberofdatapoints').value;
    var vitalsdataorder = document.getElementById("vitalsdataorder").value;
    var lowerlimit = 0;
    var parameters = "which=" + escape("viewallvitals")+
        "&pig_id=" + escape(document.form1.pig_id.value)+
        "&vitalsdataorder=" + escape(vitalsdataorder)+
        "&lowerlimit=" + escape((lowerlimit))+
        "&numberofdatapoints=" + escape(numberofdatapoints);

    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function requestAllLabs(){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    if(!window["allLabsPopup"] || window["allLabsPopup"].closed){}
    else{window["allLabsPopup"].close();}
    var alllabsdisplay = document.getElementById("labsdataholder").style;
    if(alllabsdisplay.display=="inline"){
        //var parameters = "which=" + escape("viewalllabs")+ "&pig_id=" +
    escape(document.form1.pig_id.value);
        alllabsdisplay.display="none";
    }
    else{
        alllabsdisplay.display = "inline";
        var parameters = "which=" + escape("viewalllabs")+ "&pig_id=" +
    escape(document.form1.pig_id.value);
        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype ="application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
}
function showAllLabs(){

```

```

document.getElementById('message1').innerHTML = '';
document.getElementById('message2').innerHTML = '';
document.getElementById('message3').innerHTML = '';
document.getElementById('message4').innerHTML = '';
var parameters = "which=" + escape("viewalllabs")+"&pig_id=" +
escape(document.form1.pig_id.value);
var method="POST";
var submitto = "../cgi-bin/SCCL/formsubmit.py";
var contenttype ="application/x-www-form-urlencoded";
var runthisonstatechange = showContents;
makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function requestAllGcsscores(){
document.getElementById('message1').innerHTML = '';
document.getElementById('message2').innerHTML = '';
document.getElementById('message3').innerHTML = '';
document.getElementById('message4').innerHTML = '';
var allgcsscoresdisplay = document.getElementById("gcsdataholder").style;
if(allgcsscoresdisplay.display=="inline"){
    allgcsscoresdisplay.display="none";
}
else{
    allgcsscoresdisplay.display = "inline";
    var parameters = "which=" + escape("viewallgcsscores")+"&pig_id=" +
escape(document.form1.pig_id.value);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
}
function showAllGcsscores(){
document.getElementById('message1').innerHTML = '';
document.getElementById('message2').innerHTML = '';
document.getElementById('message3').innerHTML = '';
document.getElementById('message4').innerHTML = '';
var allgcsscoresdisplay = document.getElementById("gcsdataholder").style;
if(allgcsscoresdisplay.display=="inline"){
    var parameters = "which=" + escape("viewallgcsscores")+"&pig_id=" +
escape(document.form1.pig_id.value);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
else{}
}
var allVitalsPopup;
function allVitalsWindow(){
document.getElementById('message1').innerHTML = '';
document.getElementById('message2').innerHTML = '';
document.getElementById('message3').innerHTML = '';
document.getElementById('message4').innerHTML = '';
var allvitalsdisplay = document.getElementById("vitalsdataholder").style;
if(allvitalsdisplay.display=="inline"){
    allvitalsdisplay.display = "none";
}
}

```

```

    }
    else{}
    if(!window["allVitalsPopup"] || window["allVitalsPopup"].closed){
        allVitalsPopup = window.open('../cgi-
bin/SCCL/dataview.py', 'allVitalswindow', 'channelmode=no,fullscreen=no,toolbar=no,d
irectories=no,status=no,menubar=no,scrollbars=yes,copyhistory=no,location=yes,resi
zable=yes');
        allVitalsPopup.focus();
    }
    else {
        allVitalsPopup.focus();
    }
}
var allLabsPopup;
function allLabsWindow(){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    var alllabsdisplay = document.getElementById("labsdataholder").style;
    if(alllabsdisplay.display=="inline"){
        alllabsdisplay.display = "none";
    }
    else{}
    if(!window["allLabsPopup"] || window["allLabsPopup"].closed){
        allLabsPopup = window.open('../cgi-
bin/SCCL/labsdataview.py', 'allLabswindow', 'channelmode=no,fullscreen=no,toolbar=no
,directories=no,status=no,menubar=no,scrollbars=yes,copyhistory=no,location=yes,re
sizable=yes');
        allLabsPopup.focus();
    }
    else {
        allLabsPopup.focus();
    }
}
function editVitalsElement1(abc){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    var x = abc.innerHTML;
    if(x.match(/^</)){}
    else{
        if(x!="&nbsp;"){
            abc.innerHTML = "<input type='text' class='editVitalsInput1' onBlur
='javascript:vitalsMakeChange(this);' value='"+x+"' />";
        }
        else{
            abc.innerHTML = "<input type='text' class='editVitalsInput1' onBlur
='javascript:vitalsMakeChange(this);' value='' />";
        }
    }
    abc.firstChild.focus();
}
function editLabsElement1(abc){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';

```



```

document.getElementById('message4').innerHTML = '';
var x = abc.innerHTML;
if(x.match(/^</)){}
else{
    if(x!="&nbsp;"){
        abc.innerHTML = "<input type='text' class='editLabsInput1' onblur
='javascript:labsMakeChange(this);' value=\"'+x+'\"/>";
    }
    else{
        abc.innerHTML = "<input type='text' class='editLabsInput1' onblur
='javascript:labsMakeChange(this);' value='\"/>";
    }
}
abc.firstChild.focus();
}
function editGcscsscoresElement(abc){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    var x = abc.innerHTML;
    if(x.match(/^</)){}
    else{
        if(x!="&nbsp;"){
            abc.innerHTML = "<input type='text' class='editGcscsscoresInput' onblur
='javascript:gcscsscoresMakeChange(this);' value=\"'+x+'\"/>";
        }
        else{
            abc.innerHTML = "<input type='text' class='editGcscsscoresInput' onblur
='javascript:gcscsscoresMakeChange(this);' value='\"/>";
        }
    }
    abc.firstChild.focus();
}
function editVitalsElement2(abc){
    allVitalsPopup.document.getElementById('message1').innerHTML = '';
    allVitalsPopup.document.getElementById('message2').innerHTML = '';
    var x = abc.innerHTML;
    if(x.match(/^</)){}
    else{
        if(x!="&nbsp;"){
            abc.innerHTML = "<input type='text' class='editVitalsInput1' onblur
='javascript:opener.vitalsMakeChange(this);' value=\"'+x+'\"/>";
        }
        else{
            abc.innerHTML = "<input type='text' class='editVitalsInput1' onblur
='javascript:opener.vitalsMakeChange(this);' value='\"/>";
        }
    }
    abc.firstChild.focus();
}
function editLabsElement2(abc){
    allLabsPopup.document.getElementById('message1').innerHTML = '';
    allLabsPopup.document.getElementById('message2').innerHTML = '';
    allLabsPopup.document.getElementById('message3').innerHTML = 'Please wait...';
    var x = abc.innerHTML;
    if(x.match(/^</)){}
    else{

```

```

        if(x!="&nbsp;"){
            abc.innerHTML = "<input type='text' class='editLabsInput1' onblur
='javascript:opener.labsMakeChange(this);' value=\"'+x+'\"/>";
        }
        else{
            abc.innerHTML = "<input type='text' class='editLabsInput1' onblur
='javascript:opener.labsMakeChange(this);' value='\"/>";
        }
    }
    abc.firstChild.focus();
}
function vitalsMakeChange(abc){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    var a = abc.parentNode.id;
    var b = a.split('_');
    var pig_id = b[0];
    var form_name = b[1];
    var exp_date = b[2];
    var split_date = b[2].split('/');
    var exp_date_formatted = split_date[2]+"-"+split_date[0]+"-"+split_date[1];
    var exp_time = b[3];
    var field_name = b[4];
    var field_value = abc.value;
    if (field_value == ""){
        field_value = "nullify"
    }
    //alert(field_name);
    //alert(field_value);
    if(form_name=="Vitals"){
        if(field_name=="eventcomments"){
            var parameters = "which=" + escape("editvitals")+
                "&elementid=" + escape(a)+
                "&pig_id=" + escape(pig_id)+
                "&exp_date=" + escape(exp_date_formatted)+
                "&exp_time=" + escape(exp_time)+
                "&"+field_name+"=" +
encodeURIComponent(field_value);
            var method="POST";
            var submitto = "../cgi-bin/SCCL/formsubmit.py";
            var contenttype = "application/x-www-form-urlencoded";
            var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
        }
        else{
            var parameters = "which=" + escape("editvitals")+
                "&elementid=" + escape(a)+
                "&pig_id=" + escape(pig_id)+
                "&exp_date=" + escape(exp_date_formatted)+
                "&exp_time=" + escape(exp_time)+
                "&"+field_name+"=" + escape(field_value);

            var method="POST";
            var submitto = "../cgi-bin/SCCL/formsubmit.py";
            var contenttype = "application/x-www-form-urlencoded";
            var runthisonstatechange = showContents;

```

```

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    }
    else{}
    //var x = abc.value;
    //abc.parentNode.innerHTML = x;
    }
function labsMakeChange(abc){
    document.getElementById('message1').innerHTML = '';
    document.getElementById('message2').innerHTML = '';
    document.getElementById('message3').innerHTML = '';
    document.getElementById('message4').innerHTML = '';
    var a = abc.parentNode.id;
    var b = a.split('_');
    var Pig_ID = b[0];
    var form_name = b[1];
    var Event_name = b[2];
    var field_name = b[3];
    var field_value = abc.value;
    if (field_value == ""){
        field_value = "nullify"
    }
    else{}
    if(form_name=="Labs"){
        var parameters = "which=" + escape("editlabs")+
            "&elementid=" + escape(a)+
            "&Pig_ID=" + escape(Pig_ID)+
            "&Event_name=" + escape(Event_name)+
            "&"+field_name+"=" + escape(field_value);

        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype ="application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    else{}
}
function gcsscoresMakeChange(abc){
    var a = abc.parentNode.id;
    var b = a.split('_');
    var Pig_ID = b[0];
    var Event_name = b[1];
    var field_value = abc.value;
    if (field_value == ""){
        field_value = "nullify"
    }
    else{}
    var parameters = "which=" + escape("editgcsscores")+
        "&elementid=" + escape(a)+
        "&Pig_ID=" + escape(Pig_ID)+
        "&Event_name=" + escape(Event_name)+
        "&field_value=" + escape(field_value);

    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}

```

```

}
function vitalsEmptyContent1(abc){
    document.getElementById(abc).innerHTML = "&nbsp;";
}
function labsEmptyContent1(abc){
    document.getElementById(abc).innerHTML = "&nbsp;";
}
function gcsscoresEmptyContent(abc){
    var allgcsscoresdisplay = document.getElementById('gcsdataholder').style;
    if(allgcsscoresdisplay.display=='inline'){
        var givealert4 = document.getElementById('message4');
        givealert4.innerHTML = 'GCS score successfully updated';
        document.getElementById(abc).innerHTML = "&nbsp;";
    }
    else{}
}
function vitalsEmptyContent2(abc){
    allVitalsPopup.document.getElementById(abc).innerHTML = "&nbsp;";
}
function labsEmptyContent2(abc){
    allLabsPopup.document.getElementById(abc).innerHTML = "&nbsp;";
}
function vitalsEmptyContent(abc){
    var allvitalsdisplay = document.getElementById('vitalsdataholder').style;
    if(window['allVitalsPopup'] && !window['allVitalsPopup'].closed){
        var givealert2 = allVitalsPopup.document.getElementById('message2');
        givealert2.innerHTML = 'Vitals successfully updated';
        vitalsEmptyContent2(abc);
    }
    else if(allvitalsdisplay.display=='inline'){
        var givealert2 = document.getElementById('message2');
        givealert2.innerHTML = 'Vitals successfully updated';
        vitalsEmptyContent1(abc);
    }
    else{}
}
function labsEmptyContent(abc){
    var alllabsdisplay = document.getElementById('labsdataholder').style;
    if(window['allLabsPopup'] && !window['allLabsPopup'].closed){
        var givealert3 = allLabsPopup.document.getElementById('message3');
        givealert3.innerHTML = 'Labs successfully updated';
        labsEmptyContent2(abc);
    }
    else if(alllabsdisplay.display=='inline'){
        var givealert3 = document.getElementById('message3');
        givealert3.innerHTML = 'Labs successfully updated';
        labsEmptyContent1(abc);
    }
    else{}
}
function vitalsEditSuccess1(abc){
    var x = document.getElementById(abc).firstChild.value;
    if (x==''){
        x = "&nbsp;";
    }
    document.getElementById(abc).innerHTML = x;
}
function labsEditSuccess1(abc){

```

```

var x = document.getElementById(abc).firstChild.value;
if (x==''){
    x = "&nbsp;";
}
document.getElementById(abc).innerHTML = x;
}
function gcsscoresEditSuccess(abc){
var allgcsscoresdisplay = document.getElementById('gcsdataholder').style;
if(allgcsscoresdisplay.display=='inline'){
var givealert4 = document.getElementById('message4');
givealert4.innerHTML = 'GCS score successfully updated';
var x = document.getElementById(abc).firstChild.value;
if (x==''){
    x = "&nbsp;";
}
else{}
document.getElementById(abc).innerHTML = x;
}
else{}
}
function vitalsEditSuccess2(abc){
var x = allVitalsPopup.document.getElementById(abc).firstChild.value;
if (x==''){
    x = "&nbsp;";
}
else{}
allVitalsPopup.document.getElementById(abc).innerHTML = x;
}
function labsEditSuccess2(abc){
var x = allLabsPopup.document.getElementById(abc).firstChild.value;
if (x==''){
    x = "&nbsp;";
}
else{}
allLabsPopup.document.getElementById(abc).innerHTML = x;
}
function vitalsEditSuccess(abc)
{
var allvitalsdisplay = document.getElementById('vitalsdataholder').style;
if(window['allVitalsPopup'] && !window['allVitalsPopup'].closed)
{
var givealert2 =
allVitalsPopup.document.getElementById('message2');
givealert2.innerHTML = 'Vitals successfully updated';
vitalsEditSuccess2(abc);
}
else if(allvitalsdisplay.display=='inline')
{
var givealert2 = document.getElementById('message2');
givealert2.innerHTML = 'Vitals successfully updated';
vitalsEditSuccess1(abc);
}
else{}
}
function labsEditSuccess(abc)
{
var alllabsdisplay = document.getElementById('labsdataholder').style;
if(window['allLabsPopup'] && !window['allLabsPopup'].closed)

```

```

        {
            var givealert3 = allLabsPopup.document.getElementById('message3');
            givealert3.innerHTML = 'Labs successfully updated';
            labsEditSuccess2(abc);
        }
    else if(alllabsdisplay.display=='inline')
    {
        var givealert3 = document.getElementById('message3');
        givealert3.innerHTML = 'Labs successfully updated';
        labsEditSuccess1(abc);
    }
}
else{}
}
function vitalsEditFailure(abc)
{
    requestAllVitalsWindowIndependent(abc);
    var allvitalsdisplay = document.getElementById('vitalsdataholder').style;
    if(window['allVitalsPopup'] && !window['allVitalsPopup'].closed)
    {
        var givealert2 =
allVitalsPopup.document.getElementById('message2');
        givealert2.innerHTML = 'There was an error in the data entry.
Please check your data and submit again';
    }
    else if(allvitalsdisplay.display=='inline')
    {
        var givealert2 = document.getElementById('message2');
        givealert2.innerHTML = 'There was an error in the data entry.
Please check your data and submit again';
    }
}
else{}
}
function labsEditFailure(abc)
{
    requestAllLabsWindowIndependent(abc);
    var alllabsdisplay = document.getElementById('labsdataholder').style;
    if(window['allLabsPopup'] && !window['allLabsPopup'].closed)
    {
        var givealert3 = allLabsPopup.document.getElementById('message3');
        givealert3.innerHTML = 'There was an error in the data entry.
Please check your data and submit again';
    }
    else if(alllabsdisplay.display=='inline')
    {
        var givealert3 = document.getElementById('message3');
        givealert3.innerHTML = 'There was an error in the data entry.
Please check your data and submit again';
    }
}
else{}
}
function gcsscoresEditFailure(abc){
    showAllGcsscores(abc);
    var allgcsscoresdisplay = document.getElementById('gcsdataholder').style;
    if(allgcsscoresdisplay.display=='inline'){
        var givealert4 = document.getElementById('message4');
        givealert4.innerHTML = 'There was an error in the data entry. Please check
your data and submit again';
    }
}

```

```

    else{}
  }
function displayNoneD02V02ER02(abc){
  var allvitalsdisplay = document.getElementById('vitalsdataholder').style;
  if(window['allVitalsPopup'] && !window['allVitalsPopup'].closed){
    var givealert2 = allVitalsPopup.document.getElementById('message2');
    givealert2.innerHTML = 'Vitals successfully updated';
    vitalsEditSuccess2(abc);
    gettoElement =
allVitalsPopup.document.getElementById(abc).parentNode.parentNode;
    //getD02Element = gettoElement.childNodes[28].firstChild;
    //getV02Element = gettoElement.childNodes[29].firstChild;
    //getER02Element = gettoElement.childNodes[30].firstChild;
    //To make sure the code works in both IE and firefox and other browsers in
the same way, following code is used instead of the above 3 lines
    getD02Element =
gettoElement.getElementsByTagName("td")[28].getElementsByTagName("div")[0];
    getV02Element =
gettoElement.getElementsByTagName("td")[29].getElementsByTagName("div")[0];
    getER02Element =
gettoElement.getElementsByTagName("td")[30].getElementsByTagName("div")[0];
    getD02Element.innerHTML = "&nbsp;";
    getV02Element.innerHTML = "&nbsp;";
    getER02Element.innerHTML = "&nbsp;";
  }
  else if(allvitalsdisplay.display=='inline'){
    var givealert2 = document.getElementById('message2');
    givealert2.innerHTML = 'Vitals successfully updated';
    vitalsEditSuccess1(abc);
    gettoElement = document.getElementById(abc).parentNode.parentNode;
    //getD02Element = gettoElement.childNodes[28].firstChild;
    //getV02Element = gettoElement.childNodes[29].firstChild;
    //getER02Element = gettoElement.childNodes[30].firstChild;
    //To make sure the code works in both IE and firefox and other browsers in
the same way, following code is used instead of the above 3 lines
    getD02Element =
gettoElement.getElementsByTagName("td")[28].getElementsByTagName("div")[0];
    getV02Element =
gettoElement.getElementsByTagName("td")[29].getElementsByTagName("div")[0];
    getER02Element =
gettoElement.getElementsByTagName("td")[30].getElementsByTagName("div")[0];
    getD02Element.innerHTML = "&nbsp;";
    getV02Element.innerHTML = "&nbsp;";
    getER02Element.innerHTML = "&nbsp;";
  }
  else{}
}
function displayAllD02V02ER02(abc,DO,VO,ERO){
  var allvitalsdisplay = document.getElementById('vitalsdataholder').style;
  if(window['allVitalsPopup'] && !window['allVitalsPopup'].closed){
    var givealert2 = allVitalsPopup.document.getElementById('message2');
    givealert2.innerHTML = 'Vitals successfully updated';
    vitalsEditSuccess(abc);
    gettoElement =
allVitalsPopup.document.getElementById(abc).parentNode.parentNode;
    //getD02Element = gettoElement.childNodes[28].firstChild;
    //getV02Element = gettoElement.childNodes[29].firstChild;
    //getER02Element = gettoElement.childNodes[30].firstChild;

```

```

        //To make sure the code works in both IE and firefox and other browsers in
        the same way, following code is used instead of the above 3 lines
        getDO2Element =
    gettoElement.getElementsByTagName("td")[28].getElementsByTagName("div")[0];
        getVO2Element =
    gettoElement.getElementsByTagName("td")[29].getElementsByTagName("div")[0];
        getERO2Element =
    gettoElement.getElementsByTagName("td")[30].getElementsByTagName("div")[0];
        getDO2Element.innerHTML = DO;
        getVO2Element.innerHTML = VO;
        getERO2Element.innerHTML = ERO;
    }
    else if(allvitalsdisplay.display=='inline'){
        var givealert2 = document.getElementById('message2');
        givealert2.innerHTML = 'Vitals successfully updated';
        vitalsEditSuccess(abc);
        gettoElement = document.getElementById(abc).parentNode.parentNode;
        //getDO2Element = gettoElement.childNodes[28].firstChild;
        //getVO2Element = gettoElement.childNodes[29].firstChild;
        //getERO2Element = gettoElement.childNodes[30].firstChild;
        //To make sure the code works in both IE and firefox and other browsers in
        the same way, following code is used instead of the above 3 lines
        getDO2Element =
    gettoElement.getElementsByTagName("td")[28].getElementsByTagName("div")[0];
        getVO2Element =
    gettoElement.getElementsByTagName("td")[29].getElementsByTagName("div")[0];
        getERO2Element =
    gettoElement.getElementsByTagName("td")[30].getElementsByTagName("div")[0];
        getDO2Element.innerHTML = DO;
        getVO2Element.innerHTML = VO;
        getERO2Element.innerHTML = ERO;
    }
    else{}
}
function displayOneDO2(abc,DO){
    var allvitalsdisplay = document.getElementById('vitalsdataholder').style;
    if(window['allVitalsPopup'] && !window['allVitalsPopup'].closed){
        var givealert2 = allVitalsPopup.document.getElementById('message2');
        givealert2.innerHTML = 'Vitals successfully updated';
        vitalsEditSuccess(abc);
        gettoElement =
    allVitalsPopup.document.getElementById(abc).parentNode.parentNode;
        //getDO2Element = gettoElement.childNodes[28].firstChild;
        //getVO2Element = gettoElement.childNodes[29].firstChild;
        //getERO2Element = gettoElement.childNodes[30].firstChild;
        //To make sure the code works in both IE and firefox and other browsers in the
        same way, following code is used instead of the above 3 lines
        getDO2Element =
    gettoElement.getElementsByTagName("td")[28].getElementsByTagName("div")[0];
        getVO2Element =
    gettoElement.getElementsByTagName("td")[29].getElementsByTagName("div")[0];
        getERO2Element =
    gettoElement.getElementsByTagName("td")[30].getElementsByTagName("div")[0];
        getDO2Element.innerHTML = DO;
        getVO2Element.innerHTML = "&nbsp;";
        getERO2Element.innerHTML = "&nbsp;";
    }
    else if(allvitalsdisplay.display=='inline'){

```



```

var givealert2 = document.getElementById('message2');
givealert2.innerHTML = 'Vitals successfully updated';
vitalsEditSuccess(abc);
gettoElement = document.getElementById(abc).parentNode.parentNode;
//getDO2Element = gettoElement.childNodes[28].firstChild;
//getVO2Element = gettoElement.childNodes[29].firstChild;
//getERO2Element = gettoElement.childNodes[30].firstChild;
//To make sure the code works in both IE and firefox and other browsers in the
same way, following code is used instead of the above 3 lines
getDO2Element =
gettoElement.getElementsByTagName("td")[28].getElementsByTagName("div")[0];
getVO2Element =
gettoElement.getElementsByTagName("td")[29].getElementsByTagName("div")[0];
getERO2Element =
gettoElement.getElementsByTagName("td")[30].getElementsByTagName("div")[0];
getDO2Element.innerHTML = DO;
getVO2Element.innerHTML = "&nbsp;";
getERO2Element.innerHTML = "&nbsp;";
}
else{}
}
function requestAllVitalsWindowIndependent()
{
var allvitalsdisplay = document.getElementById('vitalsdataholder').style;
if(window['allVitalsPopup'] && !window['allVitalsPopup'].closed)
{
var givealert2 =
allVitalsPopup.document.getElementById('message2');
requestAllVitalsNewWindow();
}
else if(allvitalsdisplay.display=='inline')
{
var givealert2 = document.getElementById('message2');
showAllVitals();
}
else{}
}
function requestAllLabsWindowIndependent()
{
var alllabsdisplay = document.getElementById('labsdataholder').style;
if(window['allLabsPopup'] && !window['allLabsPopup'].closed)
{
var givealert3 = allLabsPopup.document.getElementById('message3');
requestAllLabsNewWindow();
}
else if(alllabsdisplay.display=='inline')
{
var givealert3 = document.getElementById('message3');
showAllLabs();
}
else{}
}
function storePigIDs()
{
var parameters = "which=" + escape("storepigids");
var method="POST";
var submitto = "../cgi-bin/SCCL/formsubmit.py";
var contenttype ="application/x-www-form-urlencoded";

```

```

        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
function getSuggestions()
{
    var enteredID = document.getElementById("pig_id").value;
    if(enteredID!='')
        {
            document.getElementById("holdsuggestions").innerHTML = "";
            for (var i=1; i<subjectsArray.length; i++)
                {
                    var thisID = subjectsArray[i];
                    if ((thisID.indexOf(enteredID) == 0) &&
(thisID!=enteredID))
                        {
                            var tempDiv =
document.createElement("div");
                            tempDiv.innerHTML = thisID;
                            tempDiv.className = "suggestions";
                            //tempDiv.onclick = makeChoice;
                            //Above line won't let me pass values
to the function. I have to use makeChoice(evt) to capture event and the element
(evt reserved for event)
                            //tempDiv.setAttribute('onclick','javascript:makeChoice(this);')
                            //Above line does not work in IE7. That
is why use the line below
                            tempDiv.onclick = new
Function('javascript:makeChoice(this.innerHTML)');
                            document.getElementById("holdsuggestions").appendChild(tempDiv);
                        }
                    else if(thisID==enteredID)
                        {
                            document.getElementById("holdsuggestions").innerHTML = "";
                        }
                    else{}
                }
        }
    else
    {
        document.getElementById("holdsuggestions").innerHTML = "";
    }
}
function makeChoice(abc)
{
    document.getElementById("pig_id").value = abc;
    document.getElementById("holdsuggestions").innerHTML = "";
    submitForm('mainform');
}
var vitals_total_data_points = 0;
var vitals_total_pages=0;
var vitals_totals_groups=0;
var vitals_page_group=0;
var vitals_page_number=1;
var vitals_previous=0;
var vitals_next=0;

```

```

function getVitalsCount()
{
    var parameters = "which=" + escape("checkVitalsCount")+
                    "&pig_id=" + escape(document.form1.pig_id.value);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function pagination()
{
    if(!window["allVitalsPopup"] || window["allVitalsPopup"].closed)
    {
        var max_number_of_datapoints =
document.form3.numberofdatapoints.value;
    }
    else
    {
        var max_number_of_datapoints =
window["allVitalsPopup"].document.form3.numberofdatapoints.value;
    }
    var max_pages_to_show = 5;
    vitals_total_pages =
parseInt(vitals_total_data_points/max_number_of_datapoints);
    if ((vitals_total_data_points%max_number_of_datapoints)>0)
    {
        vitals_total_pages = vitals_total_pages+1;
    }
    else{}
    vitals_totals_groups = parseInt(vitals_total_pages/max_pages_to_show);
    if ((vitals_total_pages%max_pages_to_show)>0)
    {
        vitals_totals_groups = vitals_totals_groups+1;
    }
    else{}
    vitals_page_group = parseInt(vitals_page_number/max_pages_to_show);
    if((vitals_page_number%max_pages_to_show)==0)
    {
        vitals_page_group = vitals_page_group-1;
    }
    else{}
    var pageNumbersHtml = '';
    vitals_previous = parseInt(vitals_page_number)-1;

    if (vitals_previous==0)
    {
        vitals_previous = 'end';
    }
    else
    {
        var group_of_prev = parseInt(vitals_previous/max_pages_to_show);
        if((vitals_previous%max_pages_to_show)==0)
        {
            group_of_prev = group_of_prev-1;
        }
        else{}
    }
}

```

```

vitals_next = parseInt(vitals_page_number)+1;
if (vitals_next>vitals_total_pages)
{
    vitals_next = 'end';
}
else
{
    var group_of_next = parseInt(vitals_next/max_pages_to_show);
    if((vitals_next%max_pages_to_show)==0)
    {
        group_of_next = group_of_next-1;
    }
    else{}
}
if(vitals_previous!='end')
{
    if(!window["allVitalsPopup"] || window["allVitalsPopup"].closed)
    {
        pageNumbersHtml = pageNumbersHtml+'<span
id=\"page'+vitals_previous+'_group'+group_of_prev+'_prev\"
onclick=\"javascript:vitalschangePage(this.id,this.innerHTML)\"
class=\"vitalschangePage\">Prev&nbsp;&laquo;&nbsp;&nbsp;&nbsp;</span>&nbsp;&nbsp;&nbsp;';
    }
    else
    {
        pageNumbersHtml = pageNumbersHtml+'<span
id=\"page'+vitals_previous+'_group'+group_of_prev+'_prev\"
onclick=\"javascript:opener.vitalschangePage(this.id,this.innerHTML)\"
class=\"vitalschangePage\">Prev&nbsp;&laquo;&nbsp;&nbsp;&nbsp;</span>&nbsp;&nbsp;&nbsp;';
    }
}
else{}
var pageList = new Array;
var page_lower_limit = (vitals_page_group*max_pages_to_show)+1;
var page_upper_limit = (vitals_page_group+1)*max_pages_to_show;
if (page_upper_limit>vitals_total_pages)
{
    page_upper_limit = vitals_total_pages;
}
else{}
for(i=page_lower_limit; i<(page_upper_limit+1);i++)
{
    if(i==vitals_page_number)
    {
        pageNumbersHtml = pageNumbersHtml+'<span
id=\"vitals_page'+i+'\" class=\"vitalsselectedPage\">'+i+'</span>&nbsp;&nbsp;&nbsp;';
    }
    else
    {
        if(!window["allVitalsPopup"] ||
window["allVitalsPopup"].closed)
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id=\"vitals_page'+i+'\"
onclick=\"javascript:vitalschangePage(this.id,this.innerHTML)\"
class=\"vitalschangePage\">'+i+'</span>&nbsp;&nbsp;&nbsp;';
        }
    }
}

```

```

        else
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id=\"vitals_page'+i+'\"
onclick=\"javascript:opener.vitalschangePage(this.id,this.innerHTML)\"
class=\"vitalschangePage\">'+i+'</span>&nbsp;&nbsp;&nbsp;';
        }
    }
    if(vitals_next!='end')
    {
        if(!window[\"allVitalsPopup\"] || window[\"allVitalsPopup\"].closed)
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id=\"page'+vitals_next+'_group'+group_of_next+'_next\"
onclick=\"javascript:vitalschangePage(this.id,this.innerHTML)\"
class=\"vitalschangePage\">&raquo;&nbsp;&nbsp;&nbsp;Next</span>'
        }
        else
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id=\"page'+vitals_next+'_group'+group_of_next+'_next\"
onclick=\"javascript:opener.vitalschangePage(this.id,this.innerHTML)\"
class=\"vitalschangePage\">&raquo;&nbsp;&nbsp;&nbsp;Next</span>'
        }
    }
    else{}
    var gotopageinnerHtml = '';
    var gotopageoptions='';
    for(var j=1;j<=vitals_total_pages;j++)
    {
        if(j==vitals_page_number)
        {
            gotopageoptions=gotopageoptions+'<option value=\"'+j+'\"
selected=\"selected\">'+j+'</option>';
        }
        else
        {
            gotopageoptions=gotopageoptions+'<option
value=\"'+j+'\">'+j+'</option>';
        }
    }
    if(!window[\"allVitalsPopup\"] || window[\"allVitalsPopup\"].closed)
    {
        gotopageinnerHtml='Go to Page: <select id=\"vitals_page_choose\"
name=\"vitals_page_choose\"
onchange=\"javascript:vitalschangePage(this.id,this.value)\">'+gotopageoptions+'</se
lect>';
    }

    document.getElementById('vitalsOptions').innerHTML=pageNumbersHtml;
    document.getElementById('vitals_totalrows').innerHTML='Total Rows:
'+vitals_total_data_points;
    document.getElementById('vitals_totalpages').innerHTML='Total
Pages: '+vitals_total_pages;

    document.getElementById('vitals_go_to_page').innerHTML=gotopageinnerHtml;
}
else

```

```

        {
            gotopageinnerHtml='Go to Page: <select id="vitals_page_choose"
name="vitals_page_choose"
onchange="javascript:opener.vitalschangePage(this.id,this.value)">'+gotopageoption
s+'</select>';

window["allVitalsPopup"].document.getElementById('vitalsOptions').innerHTML=pageNu
mbersHtml;

window["allVitalsPopup"].document.getElementById('vitals_totalrows').innerHTML='To
tal Rows: '+vitals_total_data_points;

window["allVitalsPopup"].document.getElementById('vitals_totalpages').innerHTML='T
otal Pages: '+vitals_total_pages;

window["allVitalsPopup"].document.getElementById('vitals_go_to_page').innerHTML=go
topageinnerHtml;
        }
    }
function vitalschangePage(elementid,elementvalue)
    {
        if(!window["allVitalsPopup"] || window["allVitalsPopup"].closed)
            {
                document.getElementById('message2').innerHTML = 'Please
wait.....';
                var numberofdatapoints = document.form3.numberofdatapoints.value;
                var vitalsdataorder = document.form3.vitalsdataorder.value;
                var showVitalsdata = "viewallvitals";
            }
        else
            {
                window["allVitalsPopup"].document.getElementById('message2').innerHTML = 'Please
wait.....';
                var numberofdatapoints =
window["allVitalsPopup"].document.form3.numberofdatapoints.value;
                var vitalsdataorder =
window["allVitalsPopup"].document.form3.vitalsdataorder.value;
                var showVitalsdata = "viewallvitalsnewwindow";
            }
        if (!parseInt(elementvalue))
            {
                var tempsplit = elementid.split("_");
                var abc = tempsplit[0];
                vitals_page_number = abc.match(/[0-9]+/g);
                var lowerlimit = ((parseInt(vitals_page_number))-
1)*numberofdatapoints;
            }
        else
            {
                vitals_page_number = parseInt(elementvalue);
                var lowerlimit = ((parseInt(elementvalue))-1)*numberofdatapoints;
            }
        var parameters = "which=" + escape(showVitalsdata)+
"&pig_id=" + escape(document.form1.pig_id.value)+
"&vitalsdataorder=" + escape(vitalsdataorder)+
"&lowerlimit=" + escape(lowerlimit)+
"&numberofdatapoints=" + escape(numberofdatapoints);
    }

```

```

        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype ="application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
function closePopupWindows(){
    closePopup("allVitalsPopup");
    closePopup("allLabsPopup");
}
function closePopup(thePopup){
    if (window[thePopup] && !window[thePopup].closed){
        window[thePopup].close();
    }
}
window.onunload = closePopupWindows;

```

Code written for fileupload.js

```

function loadPigIds()
{
    //document.getElementById("message1").innerHTML = '';
    var parameters = "which=" + escape("loadpigids");
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function selectFileType(a)
{
    if(a=="Select" || a=="")
    {
        document.getElementById('filesholder').innerHTML='';
        document.getElementById("selectfiletype").style.display = "none";
        document.getElementById("allowupload").style.display = "none";
        document.getElementById("selecttissue").style.display =
"none";
        document.getElementById("selectevent").style.display = "none";
        document.getElementById("entertime").style.display = "none";
        document.getElementById("enterrawwt").style.display = "none";
        document.getElementById("enterlyowt").style.display = "none";
        document.getElementById("enternormalization").style.display =
"none";
    }
    else
    {
        document.getElementById('filesholder').innerHTML='';
        document.getElementById("allowupload").style.display = "none";
        document.getElementById("selecttissue").style.display =
"none";
        document.getElementById("selectevent").style.display = "none";
        document.getElementById("entertime").style.display = "none";
        document.getElementById("enterrawwt").style.display = "none";
        document.getElementById("enterlyowt").style.display = "none";
    }
}

```

```

        document.getElementById("enternormalization").style.display =
"none";
        document.getElementById("selectfiletype").style.display = "table";
        document.getElementById("dynamicoptions").innerHTML='<select
name="typeoffile" id="typeoffile"
onchange="javascript:selectTissueType(this.value);">\
                                                    <option
selected="selected" value="Select">Select</option>\
                                                    <option
value="Vitals">Vitals</option>\
                                                    <option
value="Lab_Values">Lab Values</option>\
                                                    <option
value="GCS_Scores">GCS Scores</option>\
                                                    <option
value="Teg_data">Teg Data</option>\
                                                    <option
value="Binning">Binning</option>\
                                                    <option
value="HNMR">HNMR</option>\
                                                    <option
value="PNMR">PNMR</option>\
                                                    </select>';
        var filetype=readCookie("fileupload_filetype");
        if ((filetype!="Select") && (filetype!=""))
        {
            document.uploadfileform.typeoffile.value=filetype;
            selectTissueType(filetype);
        }
        else{}
    }
}
function selectTissueType(a)
{
    document.getElementById("message1").innerHTML = '';
    var Pig_ID = document.uploadfileform.pigids.value;
    var File_type = document.uploadfileform.typeoffile.value;
    if(a=="Select")
    {
        document.getElementById("selecttissuetype").style.display =
"none";
        document.getElementById("selectevent").style.display = "none";
        document.getElementById("entertime").style.display = "none";
        document.getElementById("allowupload").style.display = "none";
        document.getElementById("enterrawwt").style.display = "none";
        document.getElementById("enterlyowt").style.display = "none";
        document.getElementById("enternormalization").style.display =
"none";
        document.getElementById('filesholder').innerHTML='';
    }
    else
    {
        var parameters = "which=" + escape("loadfiles")+
"&Pig_ID=" + escape(Pig_ID)+
"&File_type=" + escape(File_type);
        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype = "application/x-www-form-urlencoded";
    }
}

```



```

        var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
        if(a=="Vitals" || a=="Lab_Values" || a=="GCS_Scores" ||
a=="Teg_data")
        {
            document.getElementById("selecttissue").style.display
= "none";
            document.getElementById("selectevent").style.display =
"none";
            document.getElementById("entertime").style.display =
"none";
            document.getElementById("enterrawwt").style.display =
"none";
            document.getElementById("enterlyowt").style.display =
"none";

document.getElementById("enternormalization").style.display = "none";
            document.getElementById("allowupload").style.display =
"table";
        }
        else if (a=="PNMR")
        {
            document.getElementById("selecttissue").style.display
= "table";

document.getElementById("dynamicoptions1").innerHTML='<select name="typeoftissue"
id="typeoftissue" onchange="javascript:selectEvent(this.value);">\
<option selected="selected" value="Select">Select</option>\
<option value="Liver">Liver</option>\
<option value="Muscle">Muscle</option>\
</select>';
            document.getElementById("allowupload").style.display =
"none";
            document.getElementById("selectevent").style.display =
"none";
            document.getElementById("enterrawwt").style.display =
"none";
            document.getElementById("enterlyowt").style.display =
"none";

document.getElementById("enternormalization").style.display = "none";
            document.getElementById("entertime").style.display =
"none";
        }
        else
        {
            document.getElementById("selecttissue").style.display
= "table";

document.getElementById("dynamicoptions1").innerHTML='<select name="typeoftissue"
id="typeoftissue" onchange="javascript:selectEvent(this.value);">\

```

```

<option selected="selected" value="Select">Select</option>\
<option value="Liver">Liver</option>\
<option value="Muscle">Muscle</option>\
<option value="Serum" id="Serum">Serum</option>\
<option value="Urine" id="Urine">Urine</option>\
</select>';
        document.getElementById("allowupload").style.display =
"none";
        document.getElementById("selectevent").style.display =
"none";
        document.getElementById("enterrawwt").style.display =
"none";
        document.getElementById("enterlyowt").style.display =
"none";
document.getElementById("enternormalization").style.display = "none";
        document.getElementById("entertime").style.display =
"none";
    }
}
function selectEvent(a)
{
    document.getElementById("message1").innerHTML = '';
    if(a=="Select")
    {
        document.getElementById("selectevent").style.display = "none";
        document.getElementById("entertime").style.display = "none";
        document.getElementById("allowupload").style.display = "none";
        document.getElementById("enterrawwt").style.display = "none";
        document.getElementById("enterlyowt").style.display = "none";
        document.getElementById("enternormalization").style.display =
"none";
    }
    else if(document.uploadfileform.typeoffile.value!="HNMR" &&
document.uploadfileform.typeoffile.value!="PNMR")
    {
        document.getElementById("selectevent").style.display = "table";
        document.getElementById("enterrawwt").style.display = "none";
        document.getElementById("enterlyowt").style.display = "none";
        document.getElementById("enternormalization").style.display =
"none";
        document.getElementById("dynamicoptions2").innerHTML='<select
name="event" id="event" onchange="javascript:selectTime(this.value);">\
selected="selected" value="Select">Select</option>\
value="0">0 (Pre-Baseline)</option>\
value="1">1 (Baseline)</option>\
value="2">2 (Shock45)</option>\

```

```

value="5">5 (FR2)</option>\
                                                                    <option
value="11">11 (FR8)</option>\
                                                                    <option
value="23">23 (FR20)</option>\
                                                                    <option
value="25">25 (PR48)</option>\
                                                                    </select>';
    }
    else if(a=="Muscle" || a=="Liver")
    {
        document.getElementById("selectevent").style.display = "table";
        document.getElementById("enterrawwt").style.display = "table";
        document.getElementById("enterlyowt").style.display = "table";
        document.getElementById("enternormalization").style.display =
"none";
        document.getElementById("dynamicoptions2").innerHTML='<select
name="event" id="event" onchange="javascript:selectTime(this.value);">\
                                                                    <option
selected="selected" value="Select">Select</option>\
                                                                    <option
value="1">1 (Baseline)</option>\
                                                                    <option
value="2">2 (Shock45)</option>\
                                                                    <option
value="5">5 (FR2)</option>\
                                                                    <option
value="11">11 (FR8)</option>\
                                                                    <option
value="23">23 (FR20)</option>\
                                                                    <option
value="25">25 (PR48)</option>\
                                                                    </select>';
    }
    else if(a=="Serum" || a=="Urine")
    {
        document.getElementById("selectevent").style.display = "table";
        document.getElementById("enterrawwt").style.display = "none";
        document.getElementById("enterlyowt").style.display = "none";
        document.getElementById("enternormalization").style.display =
"table";
        document.getElementById("dynamicoptions2").innerHTML='<select
name="event" id="event" onchange="javascript:selectTime(this.value);">\
                                                                    <option
selected="selected" value="Select">Select</option>\
                                                                    <option
value="0">0 (Pre-Baseline)</option>\
                                                                    <option
value="1">1 (Baseline)</option>\
                                                                    <option
value="2">2 (Shock45)</option>\
                                                                    <option
value="5">5 (FR2)</option>\
                                                                    <option
value="11">11 (FR8)</option>\

```

```

value="23">23 (FR20)</option>\
value="25">25 (PR48)</option>\
</select>';
    }
    else
    {
    }
}
function selectTime(a)
{
    document.getElementById("message1").innerHTML = '';
    if(a=="Select")
    {
        document.getElementById("allowupload").style.display = "none";
    }
    else if(a=="0")
    {
        document.getElementById("entertime").style.display = "table";
        document.getElementById("allowupload").style.display = "table";
    }
    else
    {
        document.getElementById("entertime").style.display = "none";
        document.getElementById("allowupload").style.display = "table";
    }
}
}
function validateForm()
{
    var filename = (document.uploadfileform.fileupload.value).split("/");
    filename = filename[(filename.length)-1];
    var dobi = filename;
    var geebi=filename.split("\\")
    geebi = geebi[(geebi.length)-1]
    filename = geebi
    if(document.uploadfileform.pigids.value=="Select")
    {
        document.cookie = 'validate_filename=; path=/';
        document.getElementById("message1").innerHTML = "Error in
selection of ID.Please check the data and try again";
        return false;
    }
    else
    {
        if(document.uploadfileform.typeoffile.value=="Select")
        {
            document.cookie = 'validate_filename=; path=/';
            document.getElementById("message1").innerHTML = "Error in
selection of File type. Please check the data and try again";
            return false;
        }
        else if(document.uploadfileform.typeoffile.value=="Vitals" ||
document.uploadfileform.typeoffile.value=="Lab_Values" ||
document.uploadfileform.typeoffile.value=="GCS_Scores" ||
document.uploadfileform.typeoffile.value=="Teg_data")

```

```

    {
        if(!((filename).length)>0)
        {
            document.cookie = 'validate_filename=; path=/';
            document.getElementById("message1").innerHTML =
"Error. Please select a file for upload and try again";
            return false;
        }
        else if((filename).match(/^[A-Za-z0-9_\-\s\.\.]+/g) ||
!((filename).match(/[\.\.]+/g)))
        {
            document.cookie = 'validate_filename=; path=/';
            document.getElementById("message1").innerHTML = "Error
in the type of file or file name. Please check the fine print for details.";
            return false;
        }
        else
        {
            document.cookie = 'validate_filename=Valid; path=/';
            document.getElementById("message1").innerHTML =
"Please wait...";
            return true;
        }
    }
else
{
    if(document.uploadfileform.typeoftissue.value=="Select")
    {
        document.cookie = 'validate_filename=; path=/';
        document.getElementById("message1").innerHTML =
"Error. Please select sample origin and try again";
        return false;
    }
    else if((document.uploadfileform.typeoftissue.value=="Muscle"
|| document.uploadfileform.typeoftissue.value=="Liver") &&
(document.uploadfileform.typeoffile.value=="HNMR" ||
document.uploadfileform.typeoffile.value=="PNMR"))
    {
        if(document.uploadfileform.tissuewt.value=='')
        {
            document.cookie = 'validate_filename=;
path=/';
            document.getElementById("message1").innerHTML
= "Error. Please complete tissue weight fields and try again";
            return false;
        }
        else
        if((document.uploadfileform.tissuewt.value).match(/^[^0-9+.\?0-
9*]/g)||parseFloat(document.uploadfileform.tissuewt.value)<0||
(document.uploadfileform.lyowt.value).match(/^[^0-9+.\?0-
9*]/g)||parseFloat(document.uploadfileform.lyowt.value)<0)
        {
            document.cookie = 'validate_filename=;
path=/';
            document.getElementById("message1").innerHTML
= "Error. Check the data entered in tissue weight fields and try again";
            return false;
        }
    }
}

```

```

else
{
if(document.uploadfileform.event.value=="Select")
{
document.cookie = 'validate_filename=;
path=/';

document.getElementById("message1").innerHTML = "Error. Please select a file for
upload and try again";

return false;
}
else
{
if(document.uploadfileform.event.value=="0")
{
if
{
document.cookie =

'validate_filename=; path=/';

document.getElementById("message1").innerHTML = "Error. Please complete hours and
minutes fields and try again";

return false;
}
else
if((document.uploadfileform.hours.value).match(/^[^0-
9+]/g)||parseInt(document.uploadfileform.hours.value)>23||parseInt(document.upload
fileform.hours.value)<0||(document.uploadfileform.minutes.value).match(/^[^0-
9+]/g)||parseInt(document.uploadfileform.minutes.value)>59||parseInt(document.uplo
adfileform.minutes.value)<0)
{
document.cookie =

'validate_filename=; path=/';

document.getElementById("message1").innerHTML = "Error. Check the data entered in
hours and minutes fields and try again";

return false;
}
else
{

if(!((filename).length)>0)

{

document.cookie = 'validate_filename=; path=/';

document.getElementById("message1").innerHTML = "Error. Please select a file for
upload and try again";

return false;
}
else
if((filename).match(/^[^A-Za-z0-9_\-\s\.\.]+/g) || !((filename).match(/[\.\.]+/g)))
{

```

```

document.cookie = 'validate_filename=; path=/';

document.getElementById("message1").innerHTML = "Error in the type of file or file
name. Please check the fine print for details.";
return false;
}
else
{

document.cookie = 'validate_filename=Valid; path=/';

document.getElementById("message1").innerHTML = "Please wait...";
return true;
}
}
else
{
if(!((filename).length)>0)
{
document.cookie =
'validate_filename=; path=/';

document.getElementById("message1").innerHTML = "Error. Please select a file for
upload and try again";
return false;
}
else if((filename).match(/^[A-
Za-z0-9_-\s\.\.]+/g) || !((filename).match(/[\.\.]+/g)))
{
document.cookie =
'validate_filename=; path=/';

document.getElementById("message1").innerHTML = "Error in the type of file or file
name. Please check the fine print for details.";
return false;
}
else
{
document.cookie =
'validate_filename=Valid; path=/';

document.getElementById("message1").innerHTML = "Please wait...";
return true;
}
}
}
}
else if((document.uploadfileform.typeoftissue.value=="Serum"
|| document.uploadfileform.typeoftissue.value=="Urine") &&
(document.uploadfileform.typeoffile.value=="HNMR" ||
document.uploadfileform.typeoffile.value=="PNMR"))
{
if(document.uploadfileform.normfact.value=='')
{

```

```

        document.cookie = 'validate_filename=';
path='/';
        document.getElementById("message1").innerHTML
= "Error. Please complete normalization factor field and try again";
        return false;
    }
    else
if((document.uploadfileform.normfact.value).match(/^[^0-9+.\?0-
9*]/g)||parseFloat(document.uploadfileform.normfact.value)<0)
    {
        document.cookie = 'validate_filename=';
path='/';
        document.getElementById("message1").innerHTML
= "Error. Check the data entered in normalization factor field and try again";
        return false;
    }
    else
    {
if(document.uploadfileform.event.value=="Select")
    {
        document.cookie = 'validate_filename=';
path='/';
        document.getElementById("message1").innerHTML = "Error. Please select a file for
upload and try again";
        return false;
    }
    else
    {
if(document.uploadfileform.event.value=="0")
    {
        if
        (document.uploadfileform.hours.value=="' ||
document.uploadfileform.minutes.value=="' )
            {
                document.cookie =
'validate_filename='; path='/';
        document.getElementById("message1").innerHTML = "Error. Please complete hours and
minutes fields and try again";
                return false;
            }
        else
if((document.uploadfileform.hours.value).match(/^[^0-
9+]/g)||parseInt(document.uploadfileform.hours.value)>23||parseInt(document.upload
fileform.hours.value)<0||((document.uploadfileform.minutes.value).match(/^[^0-
9+]/g)||parseInt(document.uploadfileform.minutes.value)>59||parseInt(document.uplo
adfileform.minutes.value)<0)
            {
                document.cookie =
'validate_filename='; path='/';
        document.getElementById("message1").innerHTML = "Error. Check the data entered in
hours and minutes fields and try again";
                return false;
            }
    }
    }
    }
    }

```



```

else
{
if(!((filename).length)>0)
{
document.cookie = 'validate_filename=; path='/';
document.getElementById("message1").innerHTML = "Error. Please select a file for
upload and try again";
return false;
}
else
if((filename).match(/^[^A-Za-z0-9_\-\s\.\.]+/g) || !((filename).match(/[\.\.]+/g)))
{
document.cookie = 'validate_filename=; path='/';
document.getElementById("message1").innerHTML = "Error in the type of file or file
name. Please check the fine print for details.";
return false;
}
else
{
document.cookie = 'validate_filename=Valid; path='/';
document.getElementById("message1").innerHTML = "Please wait...";
return true;
}
}
else
{
if(!((filename).length)>0)
{
document.cookie =
'validate_filename=; path='/';
document.getElementById("message1").innerHTML = "Error. Please select a file for
upload and try again";
return false;
}
else if((filename).match(/^[^A-
Za-z0-9_\-\s\.\.]+/g) || !((filename).match(/[\.\.]+/g)))
{
document.cookie =
'validate_filename=; path='/';
document.getElementById("message1").innerHTML = "Error in the type of file or file
name. Please check the fine print for details.";
return false;
}
else
{
document.cookie =
'validate_filename=Valid; path='/';

```

```

document.getElementById("message1").innerHTML = "Please wait...";
                                return true;
                                }
                                }
                                }
else
{
    if(document.uploadfileform.event.value=="Select")
    {
        document.cookie = 'validate_filename=;
path=/';
        document.getElementById("message1").innerHTML
= "Error. Please select a file for upload and try again";
        return false;
    }
    else
    {
        if(document.uploadfileform.event.value=="0")
        {
            if
(document.uploadfileform.hours.value==' ' ||
document.uploadfileform.minutes.value==' ')
            {
                document.cookie =
'validate_filename=; path=/';
                document.getElementById("message1").innerHTML = "Error. Please complete hours and
minutes fields and try again";
                return false;
            }
            else
if((document.uploadfileform.hours.value).match(/^[^0-
9+]/g)||parseInt(document.uploadfileform.hours.value)>23||parseInt(document.upload
fileform.hours.value)<0||(document.uploadfileform.minutes.value).match(/^[^0-
9+]/g)||parseInt(document.uploadfileform.minutes.value)>59||parseInt(document.uplo
adfileform.minutes.value)<0)
            {
                document.cookie =
'validate_filename=; path=/';
                document.getElementById("message1").innerHTML = "Error. Check the data entered in
hours and minutes fields and try again";
                return false;
            }
            else
            {
                if(!((filename).length)>0)
                {
                    document.cookie =
'validate_filename=; path=/';
                    document.getElementById("message1").innerHTML = "Error. Please select a file for
upload and try again";
                    return false;
                }
            }
        }
    }
}

```

```

        else if((filename).match(/^[^A-
Za-z0-9_\-\.]+/g) || !((filename).match(/[\.\.]+/g)))
        {
            document.cookie =
'validate_filename=; path=/';

document.getElementById("message1").innerHTML = "Error in the type of file or file
name. Please check the fine print for details.";
            return false;
        }
        else
        {
            document.cookie =
'validate_filename=Valid; path=/';

document.getElementById("message1").innerHTML = "Please wait...";
            return true;
        }
    }
else
{
    if(!((filename).length)>0)
    {
        document.cookie =
'validate_filename=; path=/';

document.getElementById("message1").innerHTML = "Error. Please select a file for
upload and try again";
            return false;
        }
        else if((filename).match(/^[^A-Za-z0-
9_\-\.]+/g) || !((filename).match(/[\.\.]+/g)))
        {
            document.cookie =
'validate_filename=; path=/';

document.getElementById("message1").innerHTML = "Error in the type of file or file
name. Please check the fine print for details.";
            return false;
        }
        else
        {
            document.cookie =
'validate_filename=Valid; path=/';

document.getElementById("message1").innerHTML = "Please wait...";
            return true;
        }
    }
}
}
}
}
}
}
function deleteFile(a,b)
{
    var parameters = "which=" + escape("deletefile")+

```

```

        "&Pig_ID=" + escape(a)+
        "&Data_type=" + escape(b);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function dataTypeChange(a,b,c)
{
    var parameters = "which=" + escape("changedatatype")+
        "&Pig_ID=" + escape(a)+
        "&old_Data_type=" + escape(b)+
        "&new_Data_type=" + escape(c);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
function downloadFile(a,b,c)
{
    window.location = "../cgi-
bin/SCCL/filedownload.py?which=normaldownload&studname="+a+"&Pig_ID="+b+"&Data_ty
pe="+c;
}

```

Code written for login.js

```

//Validate and allow submit code start
function validateForm()
{
    document.getElementById('message1').innerHTML='';
    document.getElementById('message1').style.color = "Red";
    document.loginform.user_id.style.color = "Black";
    document.loginform.password.style.color = "Black";
    if(!((document.loginform.user_id.value).length> 7) ||
!((document.loginform.user_id.value).match(/^[A-Za-z]+_[A-Za-z0-9]*$/)))
    {
        document.getElementById('message1').innerHTML= "Error in User ID.
Please check the fine print";
        document.loginform.user_id.focus();
        document.loginform.user_id.style.color = "Red";
        document.cookie = 'jslogin=Invalid; path=/';
        return false;
    }
    //else if(!((document.changepasswordform.newpassword.value).length>6) ||
!((document.changepasswordform.newpassword.value).match(/[A-Za-z0-9,_*\.\_-
@#\$\%\&\(\)]+/g)) || ((document.changepasswordform.newpassword.value).match(/^[A-
Za-z_\*\.\_-@#\$\%\&\(\),0-9]+/g)) ||
!((document.changepasswordform.newpassword.value).match(/[A-Za-z]+/g)) ||
!((document.changepasswordform.newpassword.value).match(/[0-9]+/g)))
    //else if(!((document.loginform.password.value).length>6) ||
((document.loginform.password.value).match(/[\+=\ \'\\"\\]+/g)) ||

```

```

!((document.loginform.password.value).match(/[A-Za-z]+/g)) ||
!((document.loginform.password.value).match(/[0-9]+/g)))
    else if(!((document.loginform.password.value).length>6) ||
((document.loginform.password.value).match(/^[A-Za-z0-9!@#$$%^&*\\(\)_\-\
\?\\<\>,\.\.\/\;\:\|]+/g)) || !((document.loginform.password.value).match(/[A-Za-
z]+/g)) || !((document.loginform.password.value).match(/[0-9]+/g)))
    {
        document.getElementById('message1').innerHTML= "Error in password
entered. Please check the fine print";
        document.loginform.password.focus();
        document.loginform.password.style.color = "Red";
        document.cookie = 'jslogin=Invalid; path=/';
        return false;
    }
else
    {
        document.getElementById('message1').innerHTML='';
        document.getElementById('message1').style.color = "Black";
        document.getElementById('message1').innerHTML='Please wait while
your request is being processed. If the wait time is more than 10sec, submit the
form again';
        document.cookie = 'jslogin=Valid; path=/';
        return true;
    }
}
//Validate and allow submit code end

```

Code written for menu.js

```

//Code below is for Ajax requests and cookie handling. This is included here in
order to avoid redundancies in every js file
var xhr = false;
function
makeRequest(parameters,method,submitto,contenttype,runthisonstatechange) {
    if (window.XMLHttpRequest) {
        try{
            xhr = new XMLHttpRequest();
        }
        catch (e) { }
    }
    else if(window.ActiveXObject){
        try {
            xhr = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(e){
            try{
                xhr = new ActiveXObject("Msxml2.XMLHTTP");
            }
            catch (e) { }
        }
    }
    if (xhr) {
        xhr.onreadystatechange = runthisonstatechange;
    }
}

```

```

        xhr.open(method,submitto, true);//Change the file to which it is
being submitted//
        xhr.setRequestHeader("Content-Type",contenttype);
        xhr.setRequestHeader("Connection", "close");
        xhr.send(parameters);
    }
    else {
        document.getElementById("message1").innerHTML = "Sorry, but I
couldn't create an XMLHttpRequest";
    }
}
function showContents() {
    try{
        if (xhr.readyState == 4){
            if (xhr.status ==200){
                eval(xhr.responseText);
            }
        }
    }
    catch(e){
        document.getElementById("message1").innerHTML = e+'There
was a problem with the request. Ready State = ' + xhr.readyState + 'Status = ' +
xhr.status;
    }
}

//Read Cookies function start
function readCookie(name){
    var ca = document.cookie.split(';');
    for(var i=0;i < ca.length;i++) {
        var c = ca[i];
        while (c.charAt(0)==' '){
            c = c.substring(1,c.length);
        }
        var d = c.split(':');
        for (var j=0;j<d.length;j++){
            var e = d[j].split('=');
            if((e[0])==name){
                return e[1]
            }
        }
    }
    return null
}

//Read Cookies function end
//Browser detect code start
var BrowserDetect = {
    init: function () {
        this.browser = this.searchString(this.dataBrowser) || "An unknown
browser";
        this.version = this.searchVersion(navigator.userAgent)
            || this.searchVersion(navigator.appVersion)
            || "an unknown version";
        this.OS = this.searchString(this.dataOS) || "an unknown OS";
    },
    searchString: function (data) {
        for (var i=0;i<data.length;i++) {
            var dataString = data[i].string;

```

```

        var dataProp = data[i].prop;
        this.versionSearchString = data[i].versionSearch ||
data[i].identity;
        if (dataString) {
            if (dataString.indexOf(data[i].subString) != -1)
                return data[i].identity;
        }
        else if (dataProp)
            return data[i].identity;
    }
},
searchVersion: function (dataString) {
    var index = dataString.indexOf(this.versionSearchString);
    if (index == -1) return;
    return
parseFloat(dataString.substring(index+this.versionSearchString.length+1));
},
dataBrowser: [
    {
        string: navigator.userAgent,
        subString: "Chrome",
        identity: "Chrome"
    },
    {
        string: navigator.userAgent,
        subString: "OmniWeb",
        versionSearch: "OmniWeb/",
        identity: "OmniWeb"
    },
    {
        string: navigator.vendor,
        subString: "Apple",
        identity: "Safari",
        versionSearch: "Version"
    },
    {
        prop: window.opera,
        identity: "Opera"
    },
    {
        string: navigator.vendor,
        subString: "iCab",
        identity: "iCab"
    },
    {
        string: navigator.vendor,
        subString: "KDE",
        identity: "Konqueror"
    },
    {
        string: navigator.userAgent,
        subString: "Firefox",
        identity: "Firefox"
    },
    {
        string: navigator.vendor,
        subString: "Camino",
        identity: "Camino"
    },
},

```

```

        {
            // for newer Netscapes (6+)
            string: navigator.userAgent,
            subString: "Netscape",
            identity: "Netscape"
        },
        {
            string: navigator.userAgent,
            subString: "MSIE",
            identity: "Explorer",
            versionSearch: "MSIE"
        },
        {
            string: navigator.userAgent,
            subString: "Gecko",
            identity: "Mozilla",
            versionSearch: "rv"
        },
        {
            // for older Netscapes (4-)
            string: navigator.userAgent,
            subString: "Mozilla",
            identity: "Netscape",
            versionSearch: "Mozilla"
        }
    ],
    dataOS : [
        {
            string: navigator.platform,
            subString: "Win",
            identity: "Windows"
        },
        {
            string: navigator.platform,
            subString: "Mac",
            identity: "Mac"
        },
        {
            string: navigator.userAgent,
            subString: "iPhone",
            identity: "iPhone/iPod"
        },
        {
            string: navigator.platform,
            subString: "Linux",
            identity: "Linux"
        }
    ]
];
BrowserDetect.init();
//Browser name: BrowserDetect.browser
//Browser version: BrowserDetect.version
//OS name: BrowserDetect.OS
//Browser detect code end
//Start of dynamic menu code
function expand(s)
{
    //offsetWidth works for width for most of the browsers as well as (getOffsetTop
+ document.body.offsetTop) for height

```



```

var menubarWidth=document.getElementById('menubarholder').offsetWidth;
var
totalMenuItems=document.getElementById('holdnavbaritems').getElementsByTagName('table');
for (var i=0;i<totalMenuItems.length;i++)
{

//totalMenuItems[i].style.width=(980/totalMenuItems.length).toString()+ 'px';

totalMenuItems[i].style.width=(menubarWidth/totalMenuItems.length).toString()+ 'px'
;
}
var td = s;
var d = td.getElementsByTagName("div").item(0);
if (BrowserDetect.browser=="Explorer"){
td.className= "menuVisibleHoverExplorer";
}
else{
td.className = "menuVisibleHover";
}
d.className = "menuHover";
}

function collapse(s)
{
var td = s;
var d = td.getElementsByTagName("div").item(0);

td.className = "menuVisible";
d.className = "menuNormal";
}
//End of dynamic menu code
//Open menu links code start
function openPage(a)
{
if(a!="filedownload")
{
document.cookie = 'page='+a+'; path=/';
/*
if(a!="fileupload")
{
document.cookie = 'session_Pig_ID=; path=/';
}
else{}
*/
}
else{}
window.location = '/'
}
//Open menu links code end
//Highlight and Un-highlight menu code start
function highlight(a)
{
a.style.backgroundColor = "#2F4F4F";
}
function unhighlight(a)
{
//a.style.backgroundColor = "#223344";
}

```

```

    a.style.backgroundColor = "#747BAD";
}
//jquery code start (skipped)
//jquery code end

```

Code written for queries.js

```

function makeTableSortable()
{
    $(document).ready(function()
    {
        $("table").tablesorter({
            headers:{
                2:{
                    sorter:false
                },
                3:{
                    sorter:false
                }
            }
        });
    });
}

function loadMyQueries()
{
    var parameters = "which=" + escape("loadmyqueries");
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}

function loadSharedQueries()
{
    var parameters = "which=" + escape("loadsharedqueries");
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}

var lowerlimit=0;
var tempdirectoryname='null';
function displayQueryResults(numberofdatapoints)
{
    var queryname=document.getElementById("queryname").innerHTML;

    try
    {
        var parameters = "which=" + escape("displayqueryresults")+
            "&queryname="+encodeURIComponent(queryname)+
            "&lowerlimit=" + escape(lowerlimit)+
            "&numberofdatapoints=" +
            escape(numberofdatapoints)+
            "&tempdirectoryname=" + escape(tempdirectoryname);
        var method="POST";
    }
}

```

```

        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype = "application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    catch(e)
    {
        alert(e);
    }
}
function selectAll(abc)
{
    if(abc=="groups")
    {
        document.getElementById('subjects').innerHTML=waitmsg;
        if(document.groupsform.allgroups.checked)
        {
            var groupnames='';
            for(var i=0;i<document.groupsform.elements.length;i++)
            {

if(document.groupsform.elements[i].type=="checkbox")
            {

document.groupsform.elements[i].checked=true;

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#D1EE
EE";

if((document.groupsform.elements[i].name!="allgroups") &&
(document.groupsform.elements[i].checked==true))
            {
                if(groupnames=='')
                {

groupnames=document.groupsform.elements[i].name
                }
                else
                {
                    groupnames=groupnames+"-
"+document.groupsform.elements[i].name
                }
            }
            else{}
        }
        else{}
    }
}

removeboxes=['vitalsdataselectform','vitalsvariablesform','hnmrdataselectform','hn
mrvariablesform','pnmrdataselectform','pnmrvariablesform','binningdataselectform',
'binningvariablesform','eventindependentvselectform','datapointsselectform','eve
ntchoiceform','submitrequestform']

removecontent=['subjects','vitalsselect','vitalsvariablesselect','hnmrselect','hnm
rvariablesselect','pnmrselect','pnmrvariablesselect','binningselect','binningvaria
blesselect','eventindependentvselect','datapointsselect']
    for(var i=0;i<removeboxes.length;i++)

```

```

        {
document.getElementById(removeboxes[i]).style.display="none";
        }
        for(var i=0;i<removecontent.length;i++)
        {
document.getElementById(removecontent[i]).innerHTML=firstmsg;
        }
        loadSubjectsqueryresult(groupnames);
    }
    else if(!document.groupsform.allgroups.checked)
    {
        for(var i=0;i<document.groupsform.elements.length;i++)
        {
            if(document.groupsform.elements[i].type=="checkbox")
            {
                document.groupsform.elements[i].checked=false;

if(document.groupsform.elements[i].parentNode.parentNode.className=="oddRow")
                {

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3
F3";

                    }
                    else
                    {

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFF
FF";

                        }

                            }
                            else{}
                        }
                    }
}

removeboxes=['vitalsdataselectform','vitalsvariablesform','hnmrdataselectform','hnmrvariablesform','pnmrdataselectform','pnmrvariablesform','binningdataselectform','binningvariablesform','eventindependentvarselectform','datapointsselectform','eventchoiceform','submitrequestform']

removecontent=['subjects','vitalsselect','vitalsvariablesselect','hnmrselect','hnmrvariablesselect','pnmrselect','pnmrvariablesselect','binningselect','binningvariablesselect','eventindependentvarselect','datapointsselect']
        for(var i=0;i<removeboxes.length;i++)
        {
document.getElementById(removeboxes[i]).style.display="none";
        }
        for(var i=0;i<removecontent.length;i++)
        {
document.getElementById(removecontent[i]).innerHTML=firstmsg;
        }
    }
    else{}
}
else{}
}
}
}

```

```

function selectOne(abc)
{
    if(abc=="groups")
    {
        document.getElementById('subjects').innerHTML=waitmsg;
        var checkboxes=0;
        var totalcheckboxes=0;
        var groupnames=''
        for(var i=0;i<document.groupsform.elements.length;i++)
        {
            if(document.groupsform.elements[i].type=="checkbox")
            {
                if((document.groupsform.elements[i].name!="allgroups") &&
                (document.groupsform.elements[i].checked==true))
                {
                    checkboxes++;

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#D1EE
EE";

                    if(groupnames=='')
                    {
groupnames=document.groupsform.elements[i].name
                    }
                    else
                    {
                        groupnames=groupnames+"-
"+document.groupsform.elements[i].name
                    }
                    else
                    {

if(document.groupsform.elements[i].parentNode.parentNode.className=="oddRow")
                    {

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#F3F3
F3";

                    }
                    else
                    {

document.groupsform.elements[i].parentNode.parentNode.style.backgroundColor="#FFFF
FF";

                    }
                }
                totalcheckboxes++;
            }
            else{}
        }
        if (checkboxes<(totalcheckboxes-1))
        {
            document.groupsform.allgroups.checked=false;

if(document.groupsform.allgroups.parentNode.parentNode.className=="oddRow")
            {

```

```

document.groupsform.allgroups.parentNode.parentNode.style.backgroundColor="#F3F3F3
";
        }
        else
        {
document.groupsform.allgroups.parentNode.parentNode.style.backgroundColor="#FFFFFF
";
        }
    }
    else if ((checkboxes+1)== totalcheckboxes)
    {
        document.groupsform.allgroups.checked=true;

document.groupsform.allgroups.parentNode.parentNode.style.backgroundColor="#D1EEEE
";
    }
    else{}
    if (checkboxes>0)
    {
        loadSubjectsqueryresult(groupnames);

removeboxes=['vitalsdataselectform','vitalsvariablesform','hnmrdataselectform','hnmrvariablesform','pnmrdataselectform','pnmrvariablesform','binningdataselectform','binningvariablesform','eventindependentvarselectform','datapointsselectform','eventchoiceform','submitrequestform']

removecontent=['vitalssselect','vitalsvariablesselect','hnmrselect','hnmrvariablesselect','pnmrselect','pnmrvariablesselect','binningselect','binningvariablesselect','eventindependentvarselect','datapointsselect']
        for(var i=0;i<removeboxes.length;i++)
        {
document.getElementById(removeboxes[i]).style.display="none";
        }
        for(var i=0;i<removecontent.length;i++)
        {

document.getElementById(removecontent[i]).innerHTML=firstmsg;
        }
    }
    else
    {

removeboxes=['vitalsdataselectform','vitalsvariablesform','hnmrdataselectform','hnmrvariablesform','pnmrdataselectform','pnmrvariablesform','binningdataselectform','binningvariablesform','eventindependentvarselectform','datapointsselectform','eventchoiceform','submitrequestform']

removecontent=['subjects','vitalssselect','vitalsvariablesselect','hnmrselect','hnmrvariablesselect','pnmrselect','pnmrvariablesselect','binningselect','binningvariablesselect','eventindependentvarselect','datapointsselect']
        for(var i=0;i<removeboxes.length;i++)
        {

document.getElementById(removeboxes[i]).style.display="none";
        }

```

```

        for(var i=0;i<removecontent.length;i++)
        {
document.getElementById(removecontent[i]).innerHTML=firstmsg;
        }
    }
    else{}
}
function queryResultWindow(){
    document.getElementById('message1').innerHTML = '';
    if(!window["queryResultPopup"] || window["queryResultPopup"].closed){
        queryResultPopup = window.open('../cgi-
bin/SCCL/queryresultdisplay.py', 'queryResultWindow', 'channelmode=no,fullscreen=no,
toolbar=no,directories=no,status=no,menubar=no,scrollbars=yes,copyhistory=no,locat
ion=no,resizable=yes');
        queryResultPopup.focus();
        queryResultPopup.onload=function(){
            queryResultPopup.document.getElementById('message1').innerHTML =
'Please wait....';
            queryResultPopup.document.getElementById('datatable').innerHTML=
'Please wait....';
        }
    }
    else {
        queryResultPopup.document.getElementById('message1').innerHTML = 'Please
wait....';
        queryResultPopup.document.getElementById('datatable').innerHTML= 'Please
wait....';

displayQueryResults(queryResultPopup.document.queryresultdisplayform.numberofdatap
oints.value);
        queryResultPopup.focus();
    }
}
function openSaveQueryForm()
{
    $(document).ready(
        function(){
            var
toggleelement=queryResultPopup.document.getElementById("querysaveform");
            $(toggleelement).show("slow");
        }
    );
}
function closeSaveQueryForm()
{
    $(document).ready(
        function(){
            var
toggleelement=queryResultPopup.document.getElementById("querysaveform");
            $(toggleelement).hide('slow');
        }
    );
}
function queryResultPopupShowmsg(elementid,message)
{
    queryResultPopup.document.getElementById(elementid).style.display='block';
    queryResultPopup.document.getElementById(elementid).innerHTML=message;
}

```

```

function queryResultPopupClearmsg(elementid)
{
    queryResultPopup.document.getElementById(elementid).innerHTML='&nbsp;';
    queryResultPopup.document.getElementById(elementid).style.display='none';
}
function clearQuerySaveMsg(secs)
{
    var totaltime=secs*1000;
    setTimeout("closeSaveQueryForm()",totaltime);
}
function showQueryFormAndIDs()
{
    var
thequeryform=queryResultPopup.document.getElementById('showqueryform').innerHTML;

queryResultPopup.document.getElementById('querysaveform').innerHTML=queryform;
queryResultPopup.document.getElementById("userids").innerHTML=waitmsg;
queryResultPopup.document.savequeryform.queryname.value='';
var parameters = "which=" + escape("loaduserids");
var method="POST";
var submitto = "../cgi-bin/SCCL/formsubmit.py";
var contenttype ="application/x-www-form-urlencoded";
var runthisonstatechange = showContents;
makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
openSaveQueryForm();
queryResultPopup.document.savequeryform.queryname.focus();
}
function saveQuery()
{
    var
nameofthequery=queryResultPopup.document.savequeryform.queryname.value;
    if (nameofthequery.length<3)
    {
        queryResultPopup.alert("Name of the query should be atleast 3
characters length");
        return false;
    }
    else{}
    var userids='';
    var firstvar=0;
    for(var j=0;j<queryResultPopup.document.savequeryform.elements.length;j++)
    {

if(queryResultPopup.document.savequeryform.elements[j].type=="checkbox" &&
queryResultPopup.document.savequeryform.elements[j].checked==true)
    {
        if (firstvar==0)
        {

userids=userids+queryResultPopup.document.savequeryform.elements[j].id;
            firstvar++;
        }
        else if (firstvar==1)
        {
            if (userids=="allusers")
            {
                break;
            }
        }
    }
}
}

```



```

                else
                {
usersids=usersids+':'+queryResultPopup.document.savequeryform.elements[j].id;
                    firstvar++;
                }
            }
            else
            {
usersids=usersids+':'+queryResultPopup.document.savequeryform.elements[j].id;
            }
        }
        else{}
    }
    var parameters = "which=" + escape("savesqlquery")+
                    "&nameofthequery="+escape(nameofthequery)+
                    "&thesqlqueryrandomkey="+escape(thesqlqueryrandomkey)+
                    "&usersids="+escape(usersids);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype = "application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
//Pagination code start
var queryresult_total_data_points = 0;
var queryresult_total_pages=0;
var queryresult_totals_groups=0;
var queryresult_page_group=0;
var queryresult_page_number=1;
var queryresult_previous=0;
var queryresult_next=0;
function pagination()
{
    if(window["queryResultPopup"] || !window["queryResultPopup"].closed)
    {
        var max_number_of_datapoints =
window["queryResultPopup"].document.queryresultdisplayform.numberofdatapoints.valu
e;
    }
    else{}
    var max_pages_to_show = 5;
    queryresult_total_pages =
parseInt(queryresult_total_data_points/max_number_of_datapoints);
    if ((queryresult_total_data_points%max_number_of_datapoints)>0)
    {
        queryresult_total_pages = queryresult_total_pages+1;
    }
    else{}
    queryresult_totals_groups =
parseInt(queryresult_total_pages/max_pages_to_show);
    if ((queryresult_total_pages%max_pages_to_show)>0)
    {
        queryresult_totals_groups = queryresult_totals_groups+1;
    }
    else{}
}

```

```

        queryresult_page_group =
parseInt(queryresult_page_number/max_pages_to_show);
        if((queryresult_page_number%max_pages_to_show)==0)
        {
            queryresult_page_group = queryresult_page_group-1;
        }
    else{}
    var pageNumbersHtml = '';
    queryresult_previous = parseInt(queryresult_page_number)-1;

    if (queryresult_previous==0)
    {
        queryresult_previous = 'end';
    }
    else
    {
        var group_of_prev =
parseInt(queryresult_previous/max_pages_to_show);
        if((queryresult_previous%max_pages_to_show)==0)
        {
            group_of_prev = group_of_prev-1;
        }
        else{}
    }
    queryresult_next = parseInt(queryresult_page_number)+1;
    if (queryresult_next>queryresult_total_pages)
    {
        queryresult_next = 'end';
    }
    else
    {
        var group_of_next = parseInt(queryresult_next/max_pages_to_show);
        if((queryresult_next%max_pages_to_show)==0)
        {
            group_of_next = group_of_next-1;
        }
        else{}
    }
    if(queryresult_previous!='end')
    {
        if(window["queryResultPopup"] ||
!window["queryResultPopup"].closed)
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id=\"page'+queryresult_previous+'_group'+group_of_prev+'_prev\"
onclick=\"javascript:opener.queryresultchangePage(this.id,this.innerHTML)\"
class=\"queryresultchangePage\">Prev&nbsp;&laquo;&nbsp;&nbsp;&nbsp;&nbsp;</span>&nbsp;&nbsp;&nbsp;';
        }
        else{}
    }
    else{}
    var pageList = new Array;
    var page_lower_limit = (queryresult_page_group*max_pages_to_show)+1;
    var page_upper_limit = (queryresult_page_group+1)*max_pages_to_show;
    if (page_upper_limit>queryresult_total_pages)
    {
        page_upper_limit = queryresult_total_pages;
    }

```

```

    }
    else{}
    for(i=page_lower_limit; i<(page_upper_limit+1);i++)
    {
        if(i==queryresult_page_number)
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id="queryresult_page'+i+'\"
class="queryresultselectedPage">'+i+'</span>&nbsp;&nbsp;&nbsp;';
        }
        else
        {
            if(window["queryResultPopup"] ||
!window["queryResultPopup"].closed)
            {
                pageNumbersHtml = pageNumbersHtml+'<span
id="queryresult_page'+i+'\"
onclick="javascript:opener.queryresultchangePage(this.id,this.innerHTML)"
class="queryresultchangePage">'+i+'</span>&nbsp;&nbsp;&nbsp;';
            }
            else{}
        }
    }
    if(queryresult_next!='end')
    {
        if(window["queryResultPopup"] ||
!window["queryResultPopup"].closed)
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id="page'+queryresult_next+'_group'+group_of_next+'_next\"
onclick="javascript:opener.queryresultchangePage(this.id,this.innerHTML)"
class="queryresultchangePage">&raquo;&nbsp;&nbsp;&nbsp;Next</span>'
        }
        else{}
    }
    else{}
    var gotopageinnerHtml = '';
    var gotopageoptions='';
    for(var j=1;j<=queryresult_total_pages;j++)
    {
        if(j==queryresult_page_number)
        {
            gotopageoptions=gotopageoptions+'<option value=\"'+j+'\"
selected="selected">'+j+'</option>';
        }
        else
        {
            gotopageoptions=gotopageoptions+'<option
value=\"'+j+'\">'+j+'</option>';
        }
    }
    if(window["queryResultPopup"] || !window["queryResultPopup"].closed)
    {
        gotopageinnerHtml='Go to Page: <select
id="queryresult_page_choose" name="queryresult_page_choose"
onchange="javascript:opener.queryresultchangePage(this.id,this.value)">'+gotopageo
ptions+'</select>';;
    }

```

```

window["queryResultPopup"].document.getElementById('queryresultOptions').innerHTML
=pageNumbersHtml;

window["queryResultPopup"].document.getElementById('queryresult_totalrows').innerH
TML='Total Rows: '+queryresult_total_data_points;

window["queryResultPopup"].document.getElementById('queryresult_totalpages').inner
HTML='Total Pages: '+queryresult_total_pages;

window["queryResultPopup"].document.getElementById('queryresult_go_to_page').inner
HTML=gotopageinnerHtml;
    }
    else{}
}

function queryresultchangePage(elementid,elementvalue)
{
    if(window["queryResultPopup"] || !window["queryResultPopup"].closed)
    {

window["queryResultPopup"].document.getElementById('message1').innerHTML = 'Please
wait.....';
        var numberofdatapoints =
window["queryResultPopup"].document.queryresultdisplayform.numberofdatapoints.valu
e;
        }
        else
        {}
        if (!parseInt(elementvalue))
        {
            var tempsplit = elementid.split("_");
            var abc = tempsplit[0];
            queryresult_page_number = abc.match(/[0-9]+/g);
            lowerlimit = ((parseInt(queryresult_page_number))-
1)*numberofdatapoints;
        }
        else
        {
            queryresult_page_number = parseInt(elementvalue);
            lowerlimit = ((parseInt(elementvalue))-1)*numberofdatapoints;
        }
        queryqueryresult(numberofdatapoints);
    }
}
//Pagination code end
$(document).ready(
function(){
    $('.togglediv').toggle(
function(){ // you can add as much here as you'd like
        var
toggleelement=$(this).parent().parent().children('.outerdiv1');
        var imageelement=$(this).children()
        $(toggleelement).show('slow');
        $(imageelement).attr('src','/images/minus.png');

    }, function() {
        var
toggleelement=$(this).parent().parent().children('.outerdiv1');

```

```

        var imageelement=$(this).children()
        $(toggleelement).hide('slow');
        $(imageelement).attr('src', '/images/plus.png');

    });
});
function downloadFile(which,studyname,File_name)
{
    queryResultPopup.location = "/cgi-
bin/SCCL/filedownload.py?which="+which+"&studyname="+studyname+"&File_name="+File_
name+"&tempdirectoryname="+tempdirectoryname;
}
function closePopupWindows(){
    closePopup("queryResultPopup");
}
function closePopup(thePopup){
    if (window[thePopup] && !window[thePopup].closed){
        window[thePopup].close();
    }
}
window.onunload = closePopupWindows;

```

Code written for queryoutput.js

```

var queryform='<form id="editqueryform" name="editqueryform" autocomplete="off"
action="." method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">\
    <div id="queryeditform">\
        <div id="querywrapper" class="wrapform">\
            <div id="quersavemessage">\
                <div>\
                    <div class="descriptor" style="padding-
bottom:10px">Name of the query</div>\
                    <div style="width:468px;height:24px"><input
type="text" name="editqueryname" id="editqueryname" maxlength="100" /></div>\
                    </div>\
                    <div class="descriptor" style="padding-
bottom:10px">Description of the query</div>\
                    <div style="width:468px;height:72px"><textarea
name="editquerydescription" id="editquerydescription"
onfocus="javascript:this.select();">No description entered</textarea></div>\
                    </div>\
                    <div style="padding-bottom:10px">\
                        <div class="descriptor">Share with</div>\
                        <div class="box">\
                            <div id="edituserid"
class="editouterdiv">&nbsp;</div>\
                        </div>\
                    </div>\
                    <div style="text-align:right">\
                        <button type="button" class="close-floatbox"
id="canceleditquery" name="canceleditquery" value="canceleditquery" title="Cancel
Saving Query">Cancel</button>\

```

```

                                <button type="submit" id="editquery"
name="savequery" value="savequery" class="buttons" title="edit query"
onclick="javascript:saveQuery('\saveeditedsqlquery\');return false">Save</button>
                                </div>\
                                </div>\
                                <div class="close-floatbox" id="closebox">Click here
to close this box</div>\
                                </div>\
                                </div>\
                                </form>'
var sharedqueryform='<form id="editqueryform" name="editqueryform"
autocomplete="off" action="." method="post" enctype="application/x-www-form-
urlencoded" onsubmit="javascript:return false">\
                                <div id="queryeditform">\
                                <div id="querywrapper" class="wrapform">\
                                <div id="querysavemessage">\
                                <div>\
                                <div class="descriptor" style="padding-
bottom:10px">Name of the query</div>\
                                <div style="width:468px;height:24px"><input
type="text" name="editqueryname" id="editqueryname" maxlength="100" /></div>\
                                </div>\
                                <div class="descriptor" style="padding-
bottom:10px">Description of the query</div>\
                                <div style="width:468px;height:72px"><textarea
name="editquerydescription" id="editquerydescription"
onfocus="javascript:this.select();">No description entered</textarea></div>\
                                </div>\
                                <div class="descriptor">Share with</div>\
                                <div class="box">\
                                <div id="edituserids"
class="editouterdiv">&nbsp;  </div>\
                                </div>\
                                </div>\
                                <div style="text-align:right">\
                                <button type="button" class="close-floatbox"
id="canceleditquery" name="canceleditquery" value="canceleditquery" title="Cancel
Saving Query">Cancel</button>\
                                <button type="submit" id="editquery"
name="savequery" value="savequery" class="buttons" title="edit query"
onclick="javascript:saveQuery('\savesharedsqlquery\');return false">Save</button>
                                </div>\
                                </div>\
                                <div class="close-floatbox" id="closebox">Click
here to close this box</div>\
                                </div>\
                                </div>\
                                </form>'
var waitmsg='<div class="innerdiv">\
                                <table class="wrappertable"><tr><td class="wrappercell">\
                                <table class="selection">\
                                <tbody>\
                                <tr class="oddRow">\
                                <td class="inform"><span class="arrowmsg"> Please
wait.....</span></td>\
                                </tr>\

```

```

                </tbody>\
            </table>\
        </td></tr></table>\
    </div>'
/*The code below doesn't work in ie so discarded
function saveQuerytest()
{
    var clickevent=document.createEvent("MouseEvents");
    clickevent.initEvent("click", true, true);
    document.getElementById("canceleditquery").dispatchEvent(clickevent);
}
*/
$(document).ready(function()
{
    $(".entirepagecontainer").height(($("window").height()-16);
});
function makeMyqueriesTableSortable()
{
    $(document).ready(function()
    {
        $("table").tablesorter({
            headers:{
                2:{
                    sorter:false
                },
                3:{
                    sorter:false
                },
                4:{
                    sorter:false
                },
                5:{
                    sorter:false
                }
            },
            sortList:[[0,0]]
        });
    });
}
function makeSharedQueriesTableSortable()
{
    $(document).ready(function()
    {
        $("table").tablesorter({
            headers:{
                3:{
                    sorter:false
                },
                4:{
                    sorter:false
                }
            },
            sortList:[[0,0]]
        });
    });
}
function loadMyQueries()

```

```

    {
        var parameters = "which=" + escape("loadmyqueries");
        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype ="application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
function loadSharedQueries()
{
    var parameters = "which=" + escape("loadsharedqueries");
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}
var lowerlimit=0;
var tempdirectoryname='null';
var queryname='';
var createduser='';
function displayQueryResults(numberofdatapoints)
{
    try
    {
        var parameters = "which=" + escape("displayqueryresults")+
            "&queryname="+encodeURIComponent(queryname)+
            "&createduser="+escape(createduser)+
            "&lowerlimit=" + escape(lowerlimit)+
            "&numberofdatapoints=" +
escape(numberofdatapoints)+
            "&tempdirectoryname=" + escape(tempdirectoryname);
        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype ="application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;

        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    catch(e)
    {
        alert(e);
    }
}
var old_query_name=new Array;
var old_user_string=new Array;
var userids=new Array;
function selectAll(abc)
{
    if(abc=="users")
    {
        if(document.editqueryform.allusers.checked)
        {
            eventindependentoptions='';
            for(var i=0;i<document.editqueryform.elements.length;i++)
            {
if(document.editqueryform.elements[i].type=="checkbox")

```



```

        {
document.editqueryform.elements[i].checked=true;

document.editqueryform.elements[i].parentNode.parentNode.style.backgroundColor="#D
1EEEE";
        }
        else{}
    }
}
else if(!document.editqueryform.allusers.checked)
{
    for(var i=0;i<document.editqueryform.elements.length;i++)
    {

if(document.editqueryform.elements[i].type=="checkbox")
    {

document.editqueryform.elements[i].checked=false;

if(document.editqueryform.elements[i].parentNode.parentNode.className=="oddRow")
    {

document.editqueryform.elements[i].parentNode.parentNode.style.backgroundColor="#F
3F3F3";
        }
        else
        {

document.editqueryform.elements[i].parentNode.parentNode.style.backgroundColor="#F
FFFF";
        }
    }
}
else{}
    }
}
else{}
}
}
function selectOne(abc)
{
    if(abc=="users")
    {
        var checkboxes=0;
        var totalcheckboxes=0;
        var usersoptions=''
        for(var i=0;i<document.editqueryform.elements.length;i++)
        {
            if(document.editqueryform.elements[i].type=="checkbox")
            {

if((document.editqueryform.elements[i].name!="allusers") &&
(document.editqueryform.elements[i].checked==true))
            {
                checkboxes++;
            }
        }
    }
}
}

```

```

document.editqueryform.elements[i].parentNode.parentNode.style.backgroundColor="#D
1EEEE";
                                if(usersoptions=='')
                                {
usersoptions=document.editqueryform.elements[i].name
                                }
                                else
                                {
                                    usersoptions=usersoptions+"-
"+document.editqueryform.elements[i].name
                                }
                                }
                                else
                                {

if(document.editqueryform.elements[i].parentNode.parentNode.className=="oddRow")
                                {

document.editqueryform.elements[i].parentNode.parentNode.style.backgroundColor="#F
3F3F3";
                                }
                                else
                                {

document.editqueryform.elements[i].parentNode.parentNode.style.backgroundColor="#F
FFFF";
                                }
                                }
                                totalcheckboxes++;
                                }
                                else{}
                                }
if (checkboxes<(totalcheckboxes-1))
{
    document.editqueryform.allusers.checked=false;

if(document.editqueryform.allusers.parentNode.parentNode.className=="oddRow")
    {

document.editqueryform.allusers.parentNode.parentNode.style.backgroundColor="#F3F3
F3";
    }
    else
    {

document.editqueryform.allusers.parentNode.parentNode.style.backgroundColor="#FFFF
FF";
    }
    }
else if ((checkboxes+1)== totalcheckboxes)
{
    document.editqueryform.allusers.checked=true;

document.editqueryform.allusers.parentNode.parentNode.style.backgroundColor="#D1EE
EE";
    }
}

```

```

        else{}
    }
    else{}
}
function queryResultWindow(nameofthequeryanduser){
    queryname=nameofthequeryanduser.split('|')[0];
    createduser=nameofthequeryanduser.split('|')[1];
    document.getElementById('message1').innerHTML = '';
    if(!window["queryResultPopup"] || window["queryResultPopup"].closed){
        queryResultPopup = window.open('../cgi-bin/SCCL/queryresultdisplay.py', 'queryResultWindow', 'channelmode=no,fullscreen=no,toolbar=no,directories=no,status=no,menubar=no,scrollbars=yes,copyhistory=no,location=no,resizable=yes');
        queryResultPopup.focus();
        queryResultPopup.onload=function(){
            queryResultPopup.document.getElementById('message1').innerHTML = 'Please wait....';
            queryResultPopup.document.getElementById('datatable').innerHTML='Please wait....';
        }
    }
    else {
        queryResultPopup.document.getElementById('message1').innerHTML = 'Please wait....';
        queryResultPopup.document.getElementById('datatable').innerHTML= 'Please wait....';
    }
    displayQueryResults(queryResultPopup.document.queryresultdisplayform.numberofdatapoints.value);
    queryResultPopup.focus();
}
function openeditQueryForm()
{
    $(document).ready(
        function(){
            var toggleelement=document.getElementById("queryeditform");
            $(toggleelement).show("slow");
        }
    );
}
function closeeditQueryForm()
{
    $(document).ready(
        function(){
            var toggleelement=document.getElementById("queryeditform");
            $(toggleelement).hide('slow');
        }
    );
}
function queryResultPopupShowmsg(elementid,message)
{
    document.getElementById(elementid).style.display='block';
    document.getElementById(elementid).innerHTML=message;
}
function queryResultPopupClearmsg(elementid)
{
    document.getElementById(elementid).innerHTML='&nbsp;';
    document.getElementById(elementid).style.display='none';
}

```

```

function clearQueryeditMsg(secs)
{
    var totaltime=secs*1000;
    setTimeout("closeeditQueryForm()",totaltime);
}
/*
function foreditQuery()
{
    $(document).ready(function(){
        $("#queryeditform").CenterIt({ignorechildren:false});
    });
}
*/

function editQuery()
{
    $(document).ready(function(){
        $(".editquery").click(function () {
            $.floatbox({
                content: queryform,
                fade: true
            });
            document.getElementById("edituserids").innerHTML=waitmsg;
            var
query_name=$(this).parent().parent().children('.queryname').html();
            old_query_name=query_name;
            var
querydescription=$(this).parent().parent().children('.description').html();
            //var
querydescription=the_parent_node.getElementsByTagName("td")[3].innerHTML;
            document.editqueryform.editqueryname.value=query_name;

            document.editqueryform.editquerydescription.value=querydescription;
            //document.editqueryform.editqueryname.focus();
            var parameters = "which=" + escape("loaduserids")+
                "&formname="+escape("editquery")+
                "&query_name="+escape(query_name);
            var method="POST";
            var submitto = "../cgi-bin/SCCL/formsubmit.py";
            var contenttype = "application/x-www-form-urlencoded";
            var runthisonstatechange = showContents;

            makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
            //openeditQueryForm();
        });
    });
}

function deleteQuery()
{
    $(document).ready(function(){
        $(".deletequery").click(function () {
            //if (deleteConfirmation())
            var getTargetElement=$(this);
            jConfirm("Are you sure that you want to delete this
query?", "confirmation dialogue", function(r)
            {
                if(r)
                {

```

```

document.getElementById('message1').innerHTML="Please wait.....";
    var
query_name=getTargetElement.parent().parent().children('.queryname').html();
    var parameters = "which=" + escape("deletequery")+
    "&query_name="+escape(query_name);
    var method="POST";
    var submitto = "../cgi-bin/SCCL/formsubmit.py";
    var contenttype ="application/x-www-form-
urlencoded";
    var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    else{}
});
});
});
}
function saveToMyQueries()
{
    $(document).ready(function(){
        $(".savequerytomyqueries").click(function () {
            $.floatbox({
                content: sharedqueryform,
                fade: true
            });
            document.getElementById("edituserids").innerHTML=waitmsg;
            var
query_name=$(this).parent().parent().children('.queryname').html();
            old_query_name=query_name;
            var
querydescription=$(this).parent().parent().children('.description').html();
            var
shared_by=$(this).parent().parent().children('.user').attr("id");
            createduser=shared_by;
            //var
querydescription=the_parent_node.getElementsByTagName("td")[3].innerHTML;
            document.editqueryform.editqueryname.value=query_name;

            document.editqueryform.editquerydescription.value=querydescription;
            //document.editqueryform.editqueryname.focus();
            var parameters = "which=" + escape("loaduserids")+
            "&formname="+escape("editsharedquery")+
            "&shared_by="+escape(shared_by)+
            "&query_name="+escape(query_name);
            var method="POST";
            var submitto = "../cgi-bin/SCCL/formsubmit.py";
            var contenttype ="application/x-www-form-urlencoded";
            var runthisonstatechange = showContents;

            makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
            //openeditQueryForm();
        });
    });
}
function removeSharedQuery()
{

```

```

$(document).ready(function(){
    $(".removequery").click(function () {
        //if(removeConfirmation())
        var getTargetElement=$(this);
        jConfirm("Are you sure that you want to delete this
query?","confirmation dialogue",function(r){
            if(r)
            {
                document.getElementById('message1').innerHTML="Please
wait.....";
                var
query_name=getTargetElement.parent().parent().children('.queryname').html();
                var
shared_by=getTargetElement.parent().parent().children('.user').attr("id");
                var parameters = "which=" +
escape("removesharedquery")+
                    "&shared_by="+escape(shared_by)+
                    "&query_name="+escape(query_name);
                var method="POST";
                var submitto = "../cgi-bin/SCCL/formsubmit.py";
                var contenttype = "application/x-www-form-urlencoded";
                var runthisonstatechange = showContents;

makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
            }
            else{}
        });
    });
});
}
function markSelectedUsers()
{
    if (old_user_string!='')
    {
        var userlist=old_user_string.split(':');
        for (var i=0;i<userlist.length;i++)
        {
            document.editqueryform[userlist[i]].checked=true;
            //alert(document.editqueryform[userlist[i]].value);
        }
        selectOne("users")
    }
    else{}
}
function saveQuery(formtype)
{
    var nameofthequery=document.editqueryform.editqueryname.value;
    var
descriptionofthequery=document.editqueryform.editquerydescription.value;
    //validateinput();
    if (nameofthequery.length<3)
    {
        alert("Name of the query should be atleast 3 characters length");
        return false;
    }
    else{}
    userids='';
    for(var j=0;j<document.editqueryform.elements.length;j++)

```

```

        {
            if(document.editqueryform.elements[j].type=="checkbox" &&
document.editqueryform.elements[j].checked==true)
            {
                if(document.editqueryform.elements[j].name!="allusers")
                {
                    if (userids=='')
                    {
                        usersids=usersids+document.editqueryform.elements[j].id;
                    }
                    else
                    {
                        usersids=usersids+':'+document.editqueryform.elements[j].id;
                    }
                }
                else{}
            }
            else{}
        }
        //alert(old_user_string);
        //alert(userids);
        var parameters = "which=" + escape("savesqlquery")+
            "&formtype="+escape(formtype)+
            "&old_query_name="+escape(old_query_name)+
            "&nameofthequery="+escape(nameofthequery)+
            "&descriptionofthequery="+escape(descriptionofthequery)+
            "&old_user_string="+escape(old_user_string)+
            "&shared_by="+escape(createduser)+
            "&userids="+escape(userids);
        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype = "application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    //Pagination code start
    var queryresult_total_data_points = 0;
    var queryresult_total_pages=0;
    var queryresult_totals_groups=0;
    var queryresult_page_group=0;
    var queryresult_page_number=1;
    var queryresult_previous=0;
    var queryresult_next=0;
    function pagination()
    {
        if(window["queryResultPopup"] || !window["queryResultPopup"].closed)
        {
            var max_number_of_datapoints =
window["queryResultPopup"].document.queryresultdisplayform.numberofdatapoints.valu
e;
        }
        else{}
        var max_pages_to_show = 5;
        queryresult_total_pages =
parseInt(queryresult_total_data_points/max_number_of_datapoints);
    }

```

```

if ((queryresult_total_data_points%max_number_of_datapoints)>0)
{
    queryresult_total_pages = queryresult_total_pages+1;
}
else{}
queryresult_totals_groups =
parseInt(queryresult_total_pages/max_pages_to_show);
if ((queryresult_total_pages%max_pages_to_show)>0)
{
    queryresult_totals_groups = queryresult_totals_groups+1;
}
else{}
queryresult_page_group =
parseInt(queryresult_page_number/max_pages_to_show);
if((queryresult_page_number%max_pages_to_show)==0)
{
    queryresult_page_group = queryresult_page_group-1;
}
else{}
var pageNumbersHtml = '';
queryresult_previous = parseInt(queryresult_page_number)-1;

if (queryresult_previous==0)
{
    queryresult_previous = 'end';
}
else
{
    var group_of_prev =
parseInt(queryresult_previous/max_pages_to_show);
    if((queryresult_previous%max_pages_to_show)==0)
    {
        group_of_prev = group_of_prev-1;
    }
    else{}
}
queryresult_next = parseInt(queryresult_page_number)+1;
if (queryresult_next>queryresult_total_pages)
{
    queryresult_next = 'end';
}
else
{
    var group_of_next = parseInt(queryresult_next/max_pages_to_show);
    if((queryresult_next%max_pages_to_show)==0)
    {
        group_of_next = group_of_next-1;
    }
    else{}
}
if(queryresult_previous!='end')
{
    if(window["queryResultPopup"] ||
!window["queryResultPopup"].closed)
    {
        pageNumbersHtml = pageNumbersHtml+'<span
id=\"page'+queryresult_previous+'_group'+group_of_prev+'_prev\"

```



```

onclick="javascript:opener.queryresultchangePage(this.id,this.innerHTML)"
class="queryresultchangePage">Prev&nbsp;&laquo;&nbsp;&nbsp;&nbsp;</span>&nbsp;&nbsp;&nbsp;'
    }
    else{}
}
else{}
var pageList = new Array;
var page_lower_limit = (queryresult_page_group*max_pages_to_show)+1;
var page_upper_limit = (queryresult_page_group+1)*max_pages_to_show;
if (page_upper_limit>queryresult_total_pages)
{
    page_upper_limit = queryresult_total_pages;
}
else{}
for(i=page_lower_limit; i<(page_upper_limit+1);i++)
{
    if(i==queryresult_page_number)
    {
        pageNumbersHtml = pageNumbersHtml+'<span
id="queryresult_page'+i+'\"
class="queryresultselectedPage">'+i+'</span>&nbsp;&nbsp;&nbsp;';
    }
    else
    {
        if(window["queryResultPopup"] ||
!window["queryResultPopup"].closed)
        {
            pageNumbersHtml = pageNumbersHtml+'<span
id="queryresult_page'+i+'\"
onclick="javascript:opener.queryresultchangePage(this.id,this.innerHTML)"
class="queryresultchangePage">'+i+'</span>&nbsp;&nbsp;&nbsp;';
        }
        else{}
    }
}
if(queryresult_next!='end')
{
    if(window["queryResultPopup"] ||
!window["queryResultPopup"].closed)
    {
        pageNumbersHtml = pageNumbersHtml+'<span
id="page'+queryresult_next+'_group'+group_of_next+'_next\"
onclick="javascript:opener.queryresultchangePage(this.id,this.innerHTML)"
class="queryresultchangePage">&raquo;&nbsp;&nbsp;&nbsp;Next</span>'
    }
    else{}
}
else{}
var gotopageinnerHtml = '';
var gotopageoptions='';
for(var j=1;j<=queryresult_total_pages;j++)
{
    if(j==queryresult_page_number)
    {
        gotopageoptions=gotopageoptions+'<option value="'+j+'\"
selected="selected">'+j+'</option>';
    }
}

```

```

        else
        {
            gotopageoptions=gotopageoptions+'<option
value=\"'+j+'\">'+j+'</option>';
        }
    }
    if(window["queryResultPopup"] || !window["queryResultPopup"].closed)
    {
        gotopageinnerHTML='Go to Page: <select
id="queryresult_page_choose" name="queryresult_page_choose"
onchange="javascript:opener.queryresultchangePage(this.id,this.value)">'+gotopageoptions+'</select>';

window["queryResultPopup"].document.getElementById('queryresultOptions').innerHTML
=pageNumbersHtml;

window["queryResultPopup"].document.getElementById('queryresult_totalrows').innerH
TML='Total Rows: '+queryresult_total_data_points;

window["queryResultPopup"].document.getElementById('queryresult_totalpages').inner
HTML='Total Pages: '+queryresult_total_pages;

window["queryResultPopup"].document.getElementById('queryresult_go_to_page').inner
HTML=gotopageinnerHTML;
    }
    else{}
}

function queryresultchangePage(elementid,elementvalue)
{
    if(window["queryResultPopup"] || !window["queryResultPopup"].closed)
    {

window["queryResultPopup"].document.getElementById('message1').innerHTML = 'Please
wait.....';
        var numberofdatapoints =
window["queryResultPopup"].document.queryresultdisplayform.numberofdatapoints.valu
e;
    }
    else
    {}
    if (!parseInt(elementvalue))
    {
        var tempsplit = elementid.split("_");
        var abc = tempsplit[0];
        queryresult_page_number = abc.match(/[0-9]+/g);
        lowerlimit = ((parseInt(queryresult_page_number))-
1)*numberofdatapoints;
    }
    else
    {
        queryresult_page_number = parseInt(elementvalue);
        lowerlimit = ((parseInt(elementvalue))-1)*numberofdatapoints;
    }
    displayQueryResults(numberofdatapoints);
}
//Pagination code end
function downloadFile(which,studyname,File_name)

```

```

    {
        queryResultPopup.location = "/cgi-
bin/SCCL/filedownload.py?which="+which+"&studynome="+studynome+"&File_name="+File_
name+"&tempdirectoryname="+tempdirectoryname;
    }
function closePopupWindows(){
    closePopup("queryResultPopup");
}
function closePopup(thePopup){
    if (window[thePopup] && !window[thePopup].closed){
        window[thePopup].close();
    }
}
window.onunload = closePopupWindows;

```

Code written for simpleexport.js

```

function loadDataType()
{
    document.getElementById("selectdatatype").style.display = "table";
    document.getElementById("dynamicoptions").innerHTML='<select
name="datatype" id="datatype"
onchange="javascript:selectMoreOptions(this.value);">\
                                <option
selected="selected" value="Select">Select</option>\
                                <option
value="Vitals">Vitals</option>\
                                <option
value="Binning">Binning</option>\
                                <option
value="HNMR">HNMR</option>\
                                <option
value="PNMR">PNMR</option>\
                                </select>';
}
function selectMoreOptions(a)
{
    document.getElementById("message1").innerHTML = '';
    document.getElementById("allexcelfileholder").innerHTML = '';
    if(a=="Select")
    {
        document.getElementById("more_options").style.display = "none";
    }
    else
    {
        if(a=="Vitals")
        {
            document.getElementById("more_options").style.display =
"table";
        }
    }
    document.getElementById("dynamicoptions1").innerHTML='<select name="dataoption"
id="dataoption" onchange="javascript:loadAllExcelFiles();">\
<option selected="selected" value="Select">Select</option>\

```

```

<option value="all_data">All data points</option>\
<option value="only_events">Only events</option>\
</select>';
    }
    else if(a=="HNMR" || a=="Binning")
    {
        document.getElementById("more_options").style.display =
"table";

document.getElementById("dynamicoptions1").innerHTML='<select name="dataoption"
id="dataoption" onchange="javascript:loadAllExcelFiles();">\
<option selected="selected" value="Select">Select</option>\
<option value="Liver">Liver</option>\
<option value="Muscle">Muscle</option>\
<option value="Serum">Serum</option>\
<option value="Urine">Urine</option>\
</select>';

    }
    else
    {
        document.getElementById("more_options").style.display =
"table";

document.getElementById("dynamicoptions1").innerHTML='<select name="dataoption"
id="dataoption" onchange="javascript:loadAllExcelFiles();">\
<option selected="selected" value="Select">Select</option>\
<option value="Liver">Liver</option>\
<option value="Muscle">Muscle</option>\
</select>';

    }
}
function updateExcelFiles()
{
    document.getElementById('message1').innerHTML='Please wait while your
request is being processed. This might take a few minutes';
    var parameters = "which=" + escape("update_all_excel_files")
    var method="POST";
    var submitto = "../cgi-bin/SCCL/update_excel_files.py";
    var contenttype ="application/x-www-form-urlencoded";
    var runthisonstatechange = showContents;
    makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
}

```

```

function loadAllExcelFiles()
{
    document.getElementById("allexcelfileholder").innerHTML = 'Please
wait.....';
    var datatype=document.getElementById('datatype').value;
    if(document.getElementById('dataoption'))
    {
        var dataoption=document.getElementById('dataoption').value;
    }
    else
    {
        var dataoption='';
    }
    if (datatype!='Select' && dataoption!='Select' && dataoption!='')
    {
        var parameters = "which=" + escape("excel_export_load_overall")+
                        "&datatype="+escape(datatype)+
                        "&dataoption="+escape(dataoption);

        var method="POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype = "application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;

        makeRequest(parameters,method,submitto,contenttype,runthisonstatechange);
    }
    else
    {
        document.getElementById("allexcelfileholder").innerHTML = '';
    }
}
function downloadFile(which,studynome,whichfile,subjectid)
{
    window.location = "../cgi-
bin/SCCL/data_download_excel.py?which="+which+"&studynome="+studynome+"&Data_type=
"+whichfile+"&Pig_ID="+subjectid;
}

```

Code written for survey.js

```

function SurveyInfoUpdate()
{
    response=confirm("If you have finished the survey by clicking on 'Finish
Survey' button at the end of the survey (not the one on this page), then click
'OK' to continue or else click 'Cancel' and finish the survey");
    if (response) {
        document.getElementById("message2").innerHTML = "Please wait .....";
        var parameters = "which=" + escape("survey_info");
        var method = "POST";
        var submitto = "../cgi-bin/SCCL/formsubmit.py";
        var contenttype = "application/x-www-form-urlencoded";
        var runthisonstatechange = showContents;
        makeRequest(parameters, method, submitto, contenttype,
runthisonstatechange);
    }
}

```

```

    else {}
}

```

List of files in stylesheets directory are shown in Figure 50.

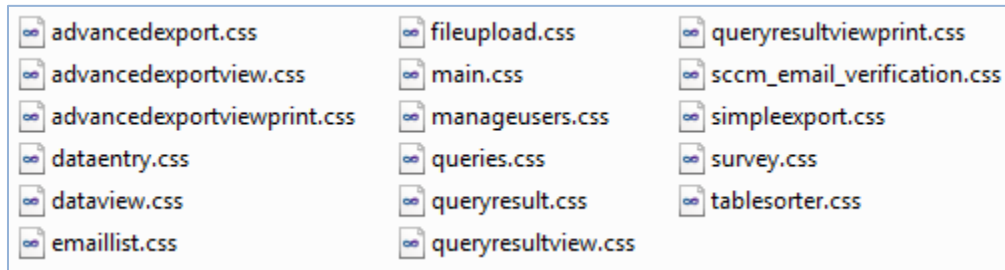


Figure 50 List of files in stylesheets directory

Code written for all the files in stylesheets directory is shown below except for emailist.css, sccm_email_verification.css as they are not essential for the project.

Code written for advancedexport.css

```

form,div,table,td,th,input
{
    border:none;
}
img
{
    cursor:pointer;
    border:0;
    margin:0;
    padding:0;
}
.descriptor
{
    text-align:right;
    vertical-align:middle;
    height:40px;
    margin:0;
    padding-right:10px;
}
.field
{
    text-align:left;
    vertical-align:middle;
    height:40px;
    margin:0;
    padding-right:10px;
}

```

```

}
div
{
    margin-left:auto;
    margin-right:auto;
    text-align:center;
}
.holdHeaders td
{
    background-color:#C8CECD;
    text-align:center;
    font-weight:bold;
    vertical-align:middle;
    font-family:Georgia, "Times New Roman", Times, serif, Arial;
}
.clickoption
{
    cursor:pointer;
}
.buttons
{
    cursor:pointer;
    font-size:17px;
    width:auto;
}
/*#subjectsform,#vitalsdataselctform,#vitalsvariablesform,#hnmrdataselctform,#hnmrvariablesform,#pnmrdataselctform,#pnmrvariablesform,#binningdataselctform,#binningvariablesform*/
#vitalsdataselctform,#vitalsvariablesform,#hnmrdataselctform,#hnmrvariablesform,#pnmrdataselctform,#pnmrvariablesform,#binningdataselctform,#binningvariablesform,#eventindependentvarselectform,#datapointsselectform,#eventchoiceform,#submitrequestform
{
    display:none;
}
.box,.box1
{
    width:466px;
    /*float:left;*/
    margin-top:10px;
    margin-bottom:10px;
    margin-left:10px;
    margin-right:10px;
    /*
    border-width:1px;
    border-color:Black;
    border-style:outset;
    */
    border:solid 1px White;
    background-color: #DAEEDD;
    /*position:relative;*/
}
.box2,.box3
{
    width:304px;
    /*float:left;*/
    margin-top:10px;
    margin-bottom:10px;
}

```

```

margin-left:10px;
margin-right:10px;
/*
border-width:1px;
border-color:Black;
border-style:outset;
*/
border:solid 1px White;
background-color: #DADEDD;
/*position:relative;*/
}
/*
.box1,.box2
{
clear:left;
position:relative;
}
*/
.box4
{
text-align:right;
}
.formbox,.formbox1,.formbox2,.formbox3
{
float:left;
position:relative;
}
.formbox1,.formbox2
{
clear:left;
position:relative;
}
.formbox4
{
float:right;
}
/*
.datagroupselect
{
width:22%;
float:left;
margin-top:10px;
margin-bottom:10px;
margin-left:10px;
margin-right:10px;
border-width:1px;
border-color:Black;
border-style:outset;
background-color: #DADEDD;
}
*/
.caption
{
width:100%;
/*
border-width:1px;
border-color:Gray;
border-style:outset;

```



```

    */
    height:20px;
    padding-top:4px;
    padding-bottom:4px;
    background-color: #747BAD;
    font-size:17px;
    font-weight:bold;
    color:White;
    vertical-align:middle;
}
.outerdiv,.outerdiv1
{
    width:100%;
    /*height:200px;*/
    /*overflow-x:hidden;
    overflow-y:scroll;*/
    /*background-color: #FFFFFF;*/
    display:block;
}
/*
#check
{
    display:none;
}
*/
.innerdiv
{
    position:relative;
    margin-left:auto;
    margin-right:auto;
    text-align:center;
    /*display:table-cell;*/
    vertical-align:middle;
    height:200px;
    width:100%;
    overflow-x:hidden;
    overflow-y:scroll;
    /*padding-right:10px;*/
}
.selection th
{
    padding-top:3px;
    padding-bottom:3px;
}
.selection td
{
    vertical-align:middle;
    padding-top:3px;
    padding-bottom:3px;
    font-family:Verdana, "Times New Roman", Times, serif, Arial;
    border-width: 1px;
    border-spacing: 2px;
    border-style: outset;
    border-color:#E0C5A0;
    /*border-color:Gray;*/
    border-collapse: separate;
    /*text-align:left;*/
    height:20px;
}

```

```

}
.optionscolumn
{
    text-align:left;
    padding-left:50px;
    width:80%;
}
.inputcolumn
{
    text-align:center;
    width:15%;
}
.subheading
{
    /*font-size:18px;
    font-weight:bold;*/
    font-style:italic;
    font-family:Verdana, "Times New Roman", Times, serif, Arial;
    background-color:#E0C5A0;
}
.inform
{
    height:45px;
}
.arrow
{
    font-size:30px;
    font-weight:bold;
}
.arrowmsg
{
    font-size:17px;
}
.alignleft
{
    float: left;
    padding-left:10px;
}
.alignright
{
    float: right;
    margin:0;
    border:solid 1px black;
    padding:0;
    position:relative;
    /*padding-right:10px;*/
    /*position:relative;*/
}
.outendiv1
{
    display:none;
}
.wrappertable
{
    vertical-align:middle;
    width:100%;
    height:100%;
}

```

```

.wrappercell
{
    vertical-align:middle;
}
.togglediv,.togglediv1
{
    position:relative;
    right:0;
    float:right;
    border:none;
    margin:4px;
    padding:0;
}

```

Code written for advancedexportview.css

```

.datatable table
{
    border:solid 1px Black;
    width:auto;
}
.datatable
{
    margin:20px;
    padding:0;
    float:left;
    clear:left;
}
.datatable td
{
    vertical-align:middle;
    padding-top:3px;
    padding-bottom:3px;
    font-family:Verdana, "Times New Roman", Times, serif, Arial;
    border-width: 1px;
    border-spacing: 2px;
    border-style: outset;
    border-color:#E0C5A0;
    /*border-color:Gray;*/
    border-collapse: separate;
}
.datatable th
{
    font-size:17px;
    font-weight:bold;
    color:White;
    padding-top:4px;
    padding-bottom:4px;
    background-color: #747BAD;
    border:solid 1px black;
}
.datatable div
{
    width:200px;
}

```

```

        text-align:center;
    }
    .formheading
    {
        padding-left:20px;
    }
    .message
    {
        padding-left:20px;
        font-size:18px;
        font-family:Verdana, "Times New Roman", Times, serif, Arial;
    }
    .downloadicons
    {
        width: 27px;
        height: 27px;
        cursor:pointer;
    }
    #downloadexcel
    {
        background:#CCCC99 url(../images/excel.png) no-repeat center;
    }
    #downloadcsv
    {
        background:#CCCC99 url(../images/csv.png) no-repeat center;
    }
    #printtable
    {
        background:#CCCC99 url(../images/print.png) no-repeat center;
    }
    #message1
    {
        float:left;
        clear:left;
        padding-left:20px;
        padding-bottom:20px;
        height:2px;
        display:block;
    }
    #savequeryform
    {
        float:left;
        clear:left;
    }
    #querysaveform
    {
        padding-left:20px;
        padding-right:20px;
        padding-bottom:20px;
        display:block;
        float:left;
        clear:left;
    }
    #querysavemsg
    {
        padding:20px;
        float:left;
        clear:left;
    }

```

```

        width:468px;
        background-color:Gray;
        color:White;
    }
    .printonly,hr,.pagebreakafter,#querysaveform,#querysavemsg
    {
        display:none;
    }
    .descriptor
    {
        width:400px;
        text-align:left;
        color:White;
        font-size:17px;
        height:24px;
        padding-top:10px;
    }
    #querysaveform .wrapform
    {
        background-color:Gray;
    }
    .wrapform
    {
        padding-left:15px;
        padding-right:15px;
        padding-bottom:10px;
        height:468px;
        width:492px;
    }
    .inlinequerymessage
    {
        color:White;
        font-size:30px;
    }
    #queryname,#querydescription
    {
        width:464px;
        border:solid 1px White;
        background-color:#E0C5A0;
        font-size:18px;
    }
    #querydescription
    {
        height:68px;
    }
    #holdbuttons
    {
        clear:left;
        padding-left:20px;
    }
    /*This code is useful when a image should be showed as background but still in img
    tags*/
    /*.excelicon
    {
        position: relative;
        height:100%;
        width:100%;
        top: 0;
    }

```

```

        left: 0
    }/*
    /*Advanced Export Pagination code start*/
    #pagination
    {
        float:left;
        clear:left;
    }
    .advancedexport_pagination_bar
    {
        margin-left:20px;
        border:0;
        padding:0;
        width:1000px;
    }
    .advancedexportchangePage
    {
        cursor:pointer;
        font-weight:bold;
    }
    .advancedexportchangePage:hover
    {
        background-color:Gray;
    }
    .advancedexportselectedPage
    {
        background-color:Gray;
    }
    .advancedexport_pagination_bar td
    {
        font-family:Verdana, "Times New Roman", Times, serif, Arial;
        text-align:left;
        font-size:17px;
    }
    .advancedexport_pagination_bar select
    {
        font-size:17px;
    }
    td.advancedexportoptions
    {
        text-align:left;
        font-family:Verdana, "Times New Roman", Times, serif, Arial;
        font-weight:bold;
        padding-right:10px;
    }
    /*Advanced Export Pagination code end*/

```

Code written for advancedexportviewprint.css

```

table
{
    border-collapse: collapse;
    border: 0;
    margin: 0;

```

```

padding: 0;
width: auto;
table-layout:fixed;
}
tr
{
text-align: center;
}
img
{
border: 0;
}
td
{
border: 0;
margin: 0;
padding: 0;
font-size: 15px;
font-family: Georgia, "Times New Roman" , Times, serif, Arial;
color: #000000;
text-align: center;
vertical-align: middle;
}
th
{
font-family: Georgia, "Times New Roman" , Times, serif, Arial;
background-color: #C8CECD;
border-right: solid 1px #000000;
border-bottom: solid 1px #000000;
border-left: none;
border-top: none;
text-align: center;
vertical-align: middle;
margin: 0;
padding: 0;
}
.datatable table
{
border:solid 2px Black;
}
.datatable td
{
vertical-align:middle;
font-family:Verdana, "Times New Roman", Times, serif, Arial;
border-width: 1px;
border-spacing: 2px;
border-style: outset;
border-color:#000000;
/*border-color:Gray;*/
border-collapse: separate;
}
.datatable th
{
font-size:17px;
font-weight:bold;
padding-top:4px;
padding-bottom:4px;
background-color: #FFFFFF;

```

```

}
.datatable div
{
    width:200px;
    text-align:center;
}
.pagebreakafter
{
    page-break-after:always;
}
#banner, .formheading, .noprint, .message, .box
{
    display:none;
}

```

Code written for dataentry.css

```

/*Start of data different from main.css*/
input
{
    text-align: center;
}
tr
{
    padding-top:20px;
    padding-bottom:20px;
}
/*End of data different from main.css*/
legend.elements
{
    font-size: 15px;
    color: #000000;
    padding: 0;
}
#causeofdeath, #overallcomments
{
    width: 170px;
}
#vitalsform, #labsform, #gcsscores
{
    border: 0;
    padding: 0;
    margin: 0;
    display: none;
}
.selection
{
    width: 110px;
    font-size: 15px;
}
.descriptor1
{
    text-align: right;
    vertical-align: middle;
    padding-right: 10px;
    width: 114px;
}

```



```

        height: 40px;
        margin: 0;
        border: 0;
    }
    .descriptor2
    {
        text-align: right;
        vertical-align: middle;
        padding-right: 6px;
        width: 100px;
        height: 40px;
        margin: 0;
        border: 0;
    }
    .field1
    {
        text-align: left;
        vertical-align: middle;
        width: 90px;
        height: 40px;
        margin: 0;
        border: 0;
        padding-right: 10px;
    }
    .field2
    {
        text-align: left;
        vertical-align: middle;
        width: 130px;
        height: 40px;
        margin: 0;
        border: 0;
        padding: 0;
    }
    .dategroup div
    {
        padding-top: 12px;
        text-align: center;
        vertical-align: middle;
        padding-bottom: 10px;
    }
    .startdateAdjust, .enddateAdjust
    {
        padding-top: 10px;
        padding-bottom: 10px;
    }
    .daymonth
    {
        width: 30px;
    }
    .year
    {
        width: 60px;
    }
    .mddy
    {
        padding-right: 2px;
        padding-left: 6px;
    }

```

```

}
.datarows
{
    text-align: center;
    padding-top: 10px;
    padding-bottom: 10px;
    vertical-align: middle;
}
.submitbuttons
{
    text-align: center;
    padding-top: 10px;
    padding-bottom: 10px;
}
.formsubmit
{
    text-align: right;
}
.innerfields
{
    margin: 0;
    width: 280px;
}
.vitalsdataholder, .labsdataholder, .gcsdataholder
{
    display: none;
}
.vitalsdataholder td, .labsdataholder td, .gcsdataholder td
{
    border-right: solid 1px #000000;
    border-bottom: solid 1px #000000;
    border-left: none;
    border-top: none;
    text-align: center;
    vertical-align: middle;
}
.holdHeaders table, .holdLabsHeaders table, .holdGcscscoresTable
{
    border-top: solid 1px #000000;
}
.holdHeaders, .holdLabsHeaders, .holdGcscscoresHeaders
{
    height: 57px;
    overflow-x: hidden;
    overflow-y: scroll;
    padding-top: 1px;
    border: 0;
    margin: 0;
    padding-bottom: 0;
}
.innerDiv
{
    height: 380px;
    overflow-x: hidden;
    overflow-y: scroll;
    border: 0;
    margin: 0;
    padding: 0;
}

```



```

padding: 0;
background-color: #F8FEEE;
}
.entireVitals, .entireLabs, .entireGcsscores
{
width: 976px;
overflow-x: scroll;
overflow-y: hidden;
margin: 0;
padding: 0;
}
.vitalsDate, .vitalsTime
{
width: 100px;
margin-left: auto;
margin-right: auto;
}
.vitalsEvent
{
width: 120px;
margin-left: auto;
margin-right: auto;
}
.vitalsColWidth
{
width: 70px;
margin-left: auto;
margin-right: auto;
}
.vitalsSpecificGravity
{
width: 80px;
margin-left: auto;
margin-right: auto;
}
.labsColWidth
{
width: 120px;
}
.gcsscoresColWidth
{
width: 240px;
}
.vitalsComments
{
width: 200px;
margin-left: auto;
margin-right: auto;
}
.vitalsDate input, .vitalsTime input
{
width: 98px;
}
.vitalsEvent input
{
width: 118px;
}
.vitalsColWidth input, .gcsscoresColWidth input, .vitalsSpecificGravity input

```

```

{
    width: 68px;
}
.labsColWidth input
{
    width: 50px;
}
.vitalsComments textarea
{
    width: 198px;
}
.vitalsDate, .vitalsTime, .vitalsEvent, .vitalsColWidth, .vitalsComments,
.labsColWidth, .gcsscoresColWidth, .vitalsSpecificGravity
{
    text-align: center;
    vertical-align: middle;
}
.holdHeaders td, .holdLabsHeaders td, .holdGcsscoresHeaders td
{
    height: 56px;
    background-color: #C8CECD;
    text-align: center;
    font-weight: bold;
    vertical-align: middle;
    font-family: Georgia, "Times New Roman" , Times, serif, Arial;
}
.innerDiv td
{
    text-align: left;
    vertical-align: middle;
    font-family: Arial, Helvetica, sans-serif;
    height: 30px;
    width: 70px;
}
.labsinnerDiv td
{
    text-align: left;
    vertical-align: middle;
    font-family: Arial, Helvetica, sans-serif;
    height: 30px;
    width: 50px;
}
.gcsscoresinnerDiv td
{
    text-align: left;
    vertical-align: middle;
    font-family: Arial, Helvetica, sans-serif;
    height: 30px;
    width: 70px;
}
.evenRow
{
    background-color: #FFFFFF;
}
.oddRow
{
    background-color: #F3F3F3;
}

```

```

/*ID suggestions box code start*/
#holdsuggestions
{
    position: absolute;
    width: 90px;
}
div.suggestions
{
    background-color: #747BAD;
    padding: 2px 6px;
    border: 1px solid #FFFFFF;
    width: 96px;
    text-align: center;
    font-family: Arial, Helvetica, sans-serif;
    color: #FFFFFF;
    cursor: pointer;
}

div.suggestions:hover
{
    background-color: #2F4F4F;
}
/*ID suggestions box code end*/
/*Vitals Pagination code start*/
.vitalschangePage
{
    cursor:pointer;
    font-weight:bold;
}
.vitalschangePage:hover
{
    background-color:Gray;
}
.vitalsselectedPage
{
    background-color:Gray;
}
.vitals_pagination_bar
{
    margin:0;
    border:0;
    padding:0;
}
.vitals_pagination_bar td
{
    margin:0;
    border:0;
    padding:0;
    font-family:Times New Roman;
}
td.vitalsoptions
{
    text-align:right;
    font-family:Times New Roman;
    font-weight:bold;
    padding-right:10px;
}
/*Vitals Pagination code end*/

```

Code written for dataview.css

```
/*Start of data different from main.css*/
input
{
    text-align:center;
}
/*End of data different from main.css*/
.vitalsdataholder,.labsdataholder
{
    margin:0;
    border:0;
    padding:0;
}
.vitalsdataholder td,.labsdataholder td
{
    border-right:solid 1px #000000;
    border-bottom:solid 1px #000000;
    border-left:none;
    border-top:none;
    text-align:center;
    vertical-align:middle;
}
.holdVitalsTable
{
    width:3780px;
    margin:0;
    padding:0;
    background-color:#F8FEEE;
}
.holdLabsTable
{
    width:1720px;
    margin:0;
    padding:0;
    background-color:#F8FEEE;
}
.holdHeaders table,.holdLabsHeaders table
{
    border-top:solid 1px #000000;
}
.holdHeaders,.holdLabsHeaders
{
    padding-top:1px;
    height:58px;
    overflow-x:hidden;
    overflow-y:scroll;
}
.innerDiv
{
    height:480px;
    overflow-x:hidden;
    overflow-y:scroll;
}
```

```

}
.labsinnerDiv
{
    height:300px;
    overflow-x:hidden;
    overflow-y:scroll;
}
.innerDiv input,.labsinnerDiv input
{
    margin:0;
    padding:0;
    text-align:center;
    vertical-align:middle;
    display:inline;
}
.vitalsDate,.vitalsTime
{
    width:100px;
    text-align:center;
    vertical-align:middle;
    margin-left:auto;
    margin-right:auto;
}
.vitalsEvent
{
    width:120px;
    text-align:center;
    vertical-align:middle;
    margin-left:auto;
    margin-right:auto;
}
.vitalsColWidth
{
    width:70px;
    text-align:center;
    vertical-align:middle;
    margin-left:auto;
    margin-right:auto;
}
.vitalsSpecificGravity
{
    width:80px;
    margin-left:auto;
    margin-right:auto;
}
.labsColWidth
{
    width:120px;
}
.vitalsComments
{
    width:200px;
    text-align:center;
    vertical-align:middle;
    margin-left:auto;
    margin-right:auto;
}
.vitalsDate input,.vitalsTime input

```



```

{
    width:98px;
}
.vitalsEvent input
{
    width:118px;
}
.vitalsColWidth input,.labsColWidth input,.vitalsSpecificGravity input
{
    width:68px;
}
.vitalsComments textarea
{
    width:318px;
}
.vitalsDate,.vitalsTime,.vitalsEvent,.vitalsColWidth,.vitalsComments,.labsColWidth
,.vitalsSpecificGravity
{
    text-align:center;
    vertical-align:middle;
}
.holdHeaders td,.holdLabsHeaders td
{
    height:56px;
    background-color:#C8CECD;
    text-align:center;
    font-weight:bold;
    vertical-align:middle;
    font-family:Georgia, "Times New Roman", Times, serif, Arial;
}
.innerDiv td
{
    text-align:left;
    vertical-align:middle;
    font-family:Arial, Helvetica, sans-serif;
    /*width:70px;*/
    height:30px;
}
.labsinnerDiv td
{
    text-align:left;
    vertical-align:middle;
    font-family:Arial, Helvetica, sans-serif;
    width:50px;
    height:30px;
}
.evenRow
{
    background-color:#FFFFFF;
}
.oddRow
{
    background-color:#F3F3F3;
}
/*Vitals Pagination code start*/
.vitalschangePage
{
    cursor:pointer;
}

```

```

        font-weight:bold;
    }
    .vitalschangePage:hover
    {
        background-color:Gray;
    }
    .vitalsselectedPage
    {
        background-color:Gray;
    }
    .vitals_pagination_bar
    {
        margin:0;
        border:0;
        padding:0;
    }
    .vitals_pagination_bar td
    {
        margin:0;
        border:0;
        padding:0;
        font-family:Times New Roman;
        text-align:left;
    }
    td.vitalsoptions
    {
        text-align:left;
        font-family:Times New Roman;
        font-weight:bold;
        padding-right:10px;
    }
    /*Vitals Pagination code end*/

```

Code written for fileupload.css

```

.descriptor
{
    text-align:right;
    vertical-align:middle;
    height:40px;
    margin:0;
    border:0;
    padding-right:10px;
}
.field
{
    text-align:left;
    vertical-align:middle;
    height:40px;
    margin:0;
    border:0;
    padding-right:10px;
}
.allrows

```

```

{
    text-align:center;
    vertical-align:middle;
    height:40px;
    margin:0;
    border:0;
    padding-right:10px;
    display:none;
}
.uploadedfiles
{
    border-top:solid 1px #000000;
    border-left:solid 1px #000000;
    padding:0;
}
.uploadedfiles td
{
    border-right:solid 1px #000000;
    border-bottom:solid 1px #000000;
    border-left:none;
    border-top:none;
    text-align:center;
    vertical-align:middle;
    padding-left:5px;
    padding-right:5px;
    height:40px;
}
div
{
    text-align:center;
}
.evenRow
{
    background-color:#FFFFFF;
}
.oddRow
{
    background-color:#F3F3F3;
}
.holdHeaders td
{
    background-color:#C8CECD;
    text-align:center;
    font-weight:bold;
    vertical-align:middle;
    font-family:Georgia, "Times New Roman", Times, serif, Arial;
}
.clickoption
{
    cursor:pointer;
}
.numbers
{
    font-family:Arial;
}

```

Code written for main.css

```
body
{
    margin-left: auto;
    margin-right: auto;
    padding: 0;
    background-color: #CCCC99; /*background-color: #9C8D67;*/ /*background:
#6A0015 url(../images/sandstone.png) repeat;*/
    width: 100%;
    height: 100%;
}
table
{
    border-collapse: collapse;
    border: 0;
    margin: 0;
    padding: 0;
    width: 100%;
}
tr
{
    text-align: center;
}
img
{
    border: 0;
}
td
{
    border: 0;
    margin: 0;
    padding: 0;
    font-size: 15px;
    font-family: Georgia, "Times New Roman" , Times, serif, Arial;
    color: #000000;
    text-align: center;
    vertical-align: middle;
}
th
{
    font-family: Georgia, "Times New Roman" , Times, serif, Arial;
    background-color: #C8CECD;
    border-right: solid 1px #000000;
    border-bottom: solid 1px #000000;
    border-left: none;
    border-top: none;
    text-align: center;
    vertical-align: middle;
    margin: 0;
    padding: 0;
}
input
{
    font-size: 15px;
    font-family: Arial, Helvetica, sans-serif;
```

```

        color: #000000;
        text-align: left;
        vertical-align: top; /*background-color: #DAEDED;*/
        background-color: #B5B5B4;
    }
    select
    {
        /*background-color: #DAEDED;*/
        background-color: #B5B5B4;
    }
    ul
    {
        list-style: none;
        padding: 0;
    }
    form
    {
        border: 0;
        margin: 0;
        padding: 0;
    }
    fieldset
    {
        /*border: solid 1px #708090;*/
        border: solid 1px #2B2F4B;
        height: auto;
        margin-left: 10px;
        margin-right: 10px;
        margin-bottom: 0;
        margin-top: 0;
    }
    legend
    {
        /*color: #2F4F4F;*/
        color: #2B2F4B;
        font-size: 18px;
        padding-top: 10px;
    }
    textarea
    {
        font-size: 14px;
        height: 30px;
        width: 320px; /*background-color: #DAEDED;*/
        background-color: #B5B5B4;
    }
    div
    {
        border: 0;
        margin: 0;
        padding: 0;
        text-align: left;
    }
    .entirepagecontainer
    {
        overflow: visible;
        margin: 0;
        border: 0;
        padding: 0;
    }

```

```

}
#entirepage
{
    width: 100%; /*background-color: #CCCC99;*/ /*margin-top: 20px;*/
}
#banner
{
    width: 100%; /*height: 47px;*/
    height: 62px;
    background-color: #2F4F4F;
}
#pageidentifier
{
    border: 0;
    padding: 0;
    margin: 0;
    width: 100%;
}
#ubanner
{
    background: #6A0015 url(../images/ubanner.gif) no-repeat left; /*width:
290px;*/
    width: 320px;
}
#mainform
{
    margin-top: 8px;
}
#mainformfield
{
    border: solid 1px #000000;
}
#message1, #message2, #message3
{
    font-family: Sans-Serif;
}
.buttons
{
    width:auto;
    cursor: pointer;
}
.aligncenter
{
    width: 980px;
    padding: 0;
    margin: 0 auto;
}
/*Drop down menu style start*/
/* ----- */

.menubar
{
    /*background-color: #747BAD;*/
    background-color: #223344; /*-moz-border-radius*/
    height: 22px;
}
p

```

```

{
    margin: 0px;
    font-weight: bold;
    text-align: center;
    vertical-align: middle;
    cursor: pointer; /*border:solid 1px white;*/
}

table.navbar
{
    /*background-color: #747BAD;*/
    background-color: #223344;
    font-size: 15px;
    margin: 0px;
    padding: 0px;
    border: 0px;
    width: 980px;
}

table.menu
{
    font-size: 12px;
    margin: 0px;
    padding: 0px;
}

.menuVisible
{
    padding: 1px;
    color: #FFFFFF;
    vertical-align: middle;
    width: 194px; /*background-color:#747BAD;*/
}

.menuVisibleHover
{
    padding: 1px;
    color: #FFFFFF;
    vertical-align: middle; /*background-color: #6699FF;*/
    width: 194px;
}

.menuVisibleHoverExplorer
{
    padding: 1px;
    color: #FFFFFF;
    vertical-align: middle; /*background-color: #6699FF;*/
    width: 194px;
    text-align: left;
}

.menuNormal td
{
    padding: 1px;
    vertical-align: middle; /*remove the following line for drop-down menu with
images*/ /*background-color: #747BAD;*/
}

.menuHover td
{

```

```

        border: 1px solid #000000;
        vertical-align: middle;
        text-align: left;
        height: 22px; /*remove the following line for drop-down menu with images*/
/*background-color: #223344;*/
        background-color: #747BAD;
        color: #FFFFFF;
        cursor: pointer;
    }
div.menuNormal
{
    display: none;
    position: absolute;
    margin: 0px;
    border: 0px;
}
div.menuHover
{
    display: block;
    position: absolute;
    margin: 0px;
    text-align: center;
}
/*Drop down menu style end*/
.smallcaption
{
    font-family: Georgia, "Times New Roman" , Times, serif;
    font-size: 32px;
    color: #FFFFFF;
    letter-spacing: 10px;
    vertical-align: middle;
    text-align: left;
    padding-left: 30px;
    padding-bottom: 5px;
    padding-top: 5px;
}
.formheading
{
    text-align: center;
    font-size: 20px;
    height: 20px;
    vertical-align: middle;
    width: 100%;
    padding-top: 17px;
    color: Maroon;
}
.tableholder, form
{
    margin: 0;
    border: 0;
    padding: 0;
}
.message
{
    text-align: center;
    height: 40px;
}
.testwebsite

```



```

{
    padding: 0;
    border: 0;
    margin: 0;
    color: Navy;
    height: 24px;
    font-weight: bold;
    font-size: 20px;
    letter-spacing: 20px;
}
.logout
{
    padding-right: 20px;
    border: 0;
    margin: 0;
    text-align: right;
    cursor: pointer;
    color: Maroon;
    height: 24px;
    width: 100px;
    font-weight: bold;
}
.newwindowlink
{
    color: Maroon;
    font-weight: bold;
    font-size: 15px;
}
.evenRow, .evenRow td
{
    background-color: #FFFFFF;
}
.oddRow, .oddRow td
{
    background-color: #F3F3F3; /*background-color: #E0C5A0;*/
}
/*
* jQuery UI CSS Framework
* Copyright (c) 2010 AUTHORS.txt (http://jqueryui.com/about)
* Dual licensed under the MIT (MIT-LICENSE.txt) and GPL (GPL-LICENSE.txt)
licenses.
*/

/* Layout helpers
-----*/
.ui-helper-hidden
{
    display: none;
}
.ui-helper-hidden-accessible
{
    position: absolute;
    left: -9999999px;
}
.ui-helper-reset
{
    margin: 0;
    padding: 0;
}

```

```

border: 0;
outline: 0;
line-height: 1.3;
text-decoration: none;
font-size: 100%;
list-style: none;
}
.ui-helper-clearfix:after
{
content: ".";
display: block;
height: 0;
clear: both;
visibility: hidden;
}
.ui-helper-clearfix
{
display: inline-block;
}
/* required comment for clearfix to work in Opera */
* html .ui-helper-clearfix
{
height: 1%;
}
.ui-helper-clearfix
{
display: block;
}
/* end clearfix */
.ui-helper-zfix
{
width: 100%;
height: 100%;
top: 0;
left: 0;
position: absolute;
opacity: 0;
filter: Alpha(Opacity=0);
}

/* Interaction Cues
-----*/
.ui-state-disabled
{
cursor: default !important;
}

/* Icons
-----*/

/* states and images */
.ui-icon
{
display: block;
text-indent: -9999px;
overflow: hidden;

```

```

    background-repeat: no-repeat;
}

/* Misc visuals
-----*/

/* Overlays */
.ui-widget-overlay
{
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
}

/*jQuery UI CSS Framework (skipped)*/

/*jquery jAlert css start (skipped)*/
/*jquery jAlert css end*/

```

Code written for manageusers.css

```

#adduserform td
{
height:40px;
}
.descriptor1
{
    text-align:right;
    vertical-align:middle;
    padding-right:10px;
    width:200px;
    margin:0;
    border:0;
}
.field1
{
    text-align:left;
    vertical-align:middle;
    padding-right:10px;
    width:120px;
    margin:0;
    border:0;
}
.userdata
{
    border-top:solid 1px #000000;
    border-left:solid 1px #000000;
    padding:0;
}
.userdata td
{
    border-right:solid 1px #000000;
}

```

```

        border-bottom:solid 1px #000000;
        border-left:none;
        border-top:none;
        text-align:center;
        vertical-align:middle;
        padding-left:5px;
        padding-right:5px;
    }
    div
    {
        text-align:center;
    }
    .evenRow
    {
        background-color:#FFFFFF;
    }
    .oddRow
    {
        background-color:#F3F3F3;
    }
    .holdHeaders td
    {
        background-color:#C8CECD;
        text-align:center;
        font-weight:bold;
        vertical-align:middle;
        font-family:Georgia, "Times New Roman", Times, serif, Arial;
    }
    .clickoption
    {
        cursor:pointer;
        height:30px;
    }
    .numbers
    {
        font-family:Arial;
    }
    .buttons
    {
        width:auto;
    }

```

Code written for queries.css

```

/*table sorter code start along with some other elements' code*/
#myqueriestable .header {
    background-image: url(../images/bg.png);
    background-repeat: no-repeat;
    background-position:2% 50%;
}
#myqueriestable .headerSortUp {
    background-image: url(../images/asc.png);
    background-repeat: no-repeat;
}

```

```

        background-position:2% 50%;
    }
    #myqueriestable .headerSortDown {
        background-image: url(../images/desc.png);
        background-repeat: no-repeat;
        background-position:2% 50%;
    }
    #myqueriestable {
        border: solid 1px White;
        height:100%;
    }
    #myqueriestable th {
        text-align: center;
        cursor:pointer;
        font-size:16px;
        /*padding: 5px;*/
        /*background-color: #909090;*/
        background-color:#747BAD;
        color:White;
        border:solid 1px White;
        height:30px;
    }
    #myqueriestable td
    {
        color: Black;
        padding: 5px;
        border:solid 1px #E0C5A0;
        font-size:16px;
        width:auto;
    }
    #myqueriestable .queryname
    {
        color:Blue;
        text-decoration:underline;
        cursor:pointer;
    }
    #myqueriestable .savedat
    {
        font-family:Times New Roman Arial Verdana Garamond;
    }
    /*This class is from jquery popup plugin*/
    .close-floatbox
    {
        cursor:pointer;
        font-size:17px;
    }
    #closebox
    {
        display:none;
        text-align:center;
        color:White;
        font-size:26px;
    }
    /*Table sorter code along with some other style ended here*/
    .buttons
    {
        cursor:pointer;
        font-size:17px;
    }

```

```

        width:auto;
    }
    .datatable
    {
        margin:20px;
        padding:0;
    }
    .formheading
    {
        padding-left:20px;
    }
    .downloadicons
    {
        width: 27px;
        height: 27px;
        cursor:pointer;
    }
    #downloadexcel
    {
        background:#CCCC99 url(../images/excel.png) no-repeat center;
    }
    #downloadcsv
    {
        background:#CCCC99 url(../images/csv.png) no-repeat center;
    }
    #printtable
    {
        background:#CCCC99 url(../images/print.png) no-repeat center;
    }
    #queryeditform
    {
        padding-left:20px;
        padding-right:20px;
        padding-bottom:20px;
        display:block;
    }
    #queryeditmsg
    {
        padding:20px;
        float:left;
        clear:left;
        /*width:468px;*/
        width:100%;
        background-color:Gray;
        color:White;
    }
    .printonly,hr,.pagebreakafter /*,#queryeditform,#queryeditmsg*/
    {
        display:none;
    }
    .descriptor
    {
        /*width:400px;*/
        width:100%;
        text-align:left;
        color:White;
        font-size:17px;
        height:24px;
    }

```

```

padding-top:10px;
}
#queryeditform .wrapform
{
background-color:Gray;
}
#editqueryform
{
float:left;
clear:left;
position:fixed;
/*top:10%;*/
/*left:29%;*/
/*height:52%;*/
/*width:40%;*/
width:511px;
}
.wrapform
{
padding-left:24px;
padding-right:24px;
padding-bottom:15px;
/*height:468px;
width:492px;*/
height:100%;
width:100%;
}
.inlinequerymessage
{
color:White;
font-size:30px;
}
input
{
border:solid 1px White;
background-color:#E0C5A0;
font-size:18px;
}
#holdquerytable
{
text-align:center;
height:100%;
}
#holdquerytable .queryname,#holdquerytable .editqueryname,#holdquerytable
.savedat,#holdquerytable .editdat,#holdquerytable .user
{
width:20%;
}
#holdquerytable .holdicons
{
width:53px;
}
#holdquerytable .editquery,#holdquerytable .deletequery,#holdquerytable
.removequery,#holdquerytable .savequerytomyqueries
{
cursor:pointer;
width:20px;
float:right;
}

```

```

        height:24px;
        padding:2px;
    }
    .box
    {
        width:466px;
        border:solid 1px White;
        background-color: #DAEEDD;
    }
    .caption
    {
        width:100%;
        height:20px;
        padding-top:4px;
        padding-bottom:4px;
        background-color: #747BAD;
        font-size:17px;
        font-weight:bold;
        color:White;
        vertical-align:middle;
    }
    .outertdiv
    {
        width:100%;
        display:block;
    }
    #editqueryname,#editquerydescription
    {
        width:464px;
        border:solid 1px White;
        background-color:#E0C5A0;
        font-size:18px;
    }
    #editquerydescription
    {
        height:68px;
    }
    /*
    #check
    {
        display:none;
    }
    */
    .innerdiv
    {
        position:relative;
        margin-left:auto;
        margin-right:auto;
        text-align:center;
        /*display:table-cell;*/
        vertical-align:middle;
        height:200px;
        width:100%;
        overflow-x:hidden;
        overflow-y:scroll;
        /*padding-right:10px;*/
    }
    .selection th

```



```

{
padding-top:3px;
padding-bottom:3px;
}
.selection td
{
vertical-align:middle;
padding-top:3px;
padding-bottom:3px;
font-family:Verdana, "Times New Roman", Times, serif, Arial;
border-width: 1px;
border-spacing: 2px;
border-style: outset;
border-color:#E0C5A0;
/*border-color:Gray;*/
border-collapse: separate;
/*text-align:left;*/
height:20px;
}
.optionscolumn
{
text-align:left;
padding-left:70px;
width:80%;
}
.inputcolumn
{
text-align:center;
width:15%;
}
.subheading
{
/*font-size:18px;
font-weight:bold;*/
font-style:italic;
font-family:Verdana, "Times New Roman", Times, serif, Arial;
background-color:#E0C5A0;
}
.inform
{
height:45px;
}
.wrappertable
{
vertical-align:middle;
width:100%;
height:100%;
}
.wrappercell
{
vertical-align:middle;
}

.editquery
{
background:url(../images/edit.png) no-repeat center;
}
.deletequery,.removequery

```

```

{
    background:url(../images/delete.png) no-repeat center;
}
.savequerytomyqueries
{
    background:url(../images/save.png) no-repeat center;
}
/*Query Result Pagination code start*/
.queryresult_pagination_bar
{
    margin-left:20px;
    border:0;
    padding:0;
    width:1000px;
}
.queryresultchangePage
{
    cursor:pointer;
    font-weight:bold;
}
.queryresultchangePage:hover
{
    background-color:Gray;
}
.queryresultselectedPage
{
    background-color:Gray;
}
.queryresult_pagination_bar td
{
    font-family:Verdana, "Times New Roman", Times, serif, Arial;
    text-align:left;
    font-size:17px;
}
.queryresult_pagination_bar select
{
    font-size:17px;
}
td.queryresultoptions
{
    text-align:left;
    font-family:Verdana, "Times New Roman", Times, serif, Arial;
    font-weight:bold;
    padding-right:10px;
}
/*Query Result Pagination code end*/

```

Code written for queryresult.css

```

form,div,table,td,th,input
{
    border:none;
}
img

```

```

{
  cursor:pointer;
  border:0;
  margin:0;
  padding:0;
}
.descriptor
{
  text-align:right;
  vertical-align:middle;
  height:40px;
  margin:0;
  padding-right:10px;
}
.field
{
  text-align:left;
  vertical-align:middle;
  height:40px;
  margin:0;
  padding-right:10px;
}
div
{
  margin-left:auto;
  margin-right:auto;
  text-align:center;
}
.holdHeaders td
{
  background-color:#C8CECD;
  text-align:center;
  font-weight:bold;
  vertical-align:middle;
  font-family:Georgia, "Times New Roman", Times, serif, Arial;
}
.clickoption
{
  cursor:pointer;
}
.buttons
{
  cursor:pointer;
  font-size:17px;
  width:auto;
}
#vitalsdataselctform,#vitalsvariablesform,#hnmrdataselctform,#hnmrvariablesform,
#pnmrdataselctform,#pnmrvariablesform,#binningdataselctform,#binningvariablesfor
m,#eventindependentvarselectform,#datapointsselctform,#eventchoiceform,#submitreq
uestform
{
  display:none;
}
.box,.box1
{
  width:466px;
  margin-top:10px;
  margin-bottom:10px;
}

```

```

        margin-left:10px;
        margin-right:10px;
        border:solid 1px White;
        background-color: #DADEDD;
    }
    .box2,.box3
    {
        width:304px;
        margin-top:10px;
        margin-bottom:10px;
        margin-left:10px;
        margin-right:10px;
        border:solid 1px White;
        background-color: #DADEDD;
    }
    .box4
    {
        text-align:right;
    }
    .formbox,.formbox1,.formbox2,.formbox3
    {
        float:left;
        position:relative;
    }
    .formbox1,.formbox2
    {
        clear:left;
        position:relative;
    }
    .formbox4
    {
        float:right;
    }
    .caption
    {
        width:100%;
        height:20px;
        padding-top:4px;
        padding-bottom:4px;
        background-color: #747BAD;
        font-size:17px;
        font-weight:bold;
        color:White;
        vertical-align:middle;
    }
    .outerdiv,.outerdiv1
    {
        width:100%;
        display:block;
    }
    .innerdiv
    {
        position:relative;
        margin-left:auto;
        margin-right:auto;
        text-align:center;
        /*display:table-cell;*/
        vertical-align:middle;
    }

```

```

        height:200px;
        width:100%;
        overflow-x:hidden;
        overflow-y:scroll;
        /*padding-right:10px;*/
    }
    .selection th
    {
        padding-top:3px;
        padding-bottom:3px;
    }
    .selection td
    {
        vertical-align:middle;
        padding-top:3px;
        padding-bottom:3px;
        font-family:Verdana, "Times New Roman", Times, serif, Arial;
        border-width: 1px;
        border-spacing: 2px;
        border-style: outset;
        border-color:#E0C5A0;
        /*border-color:Gray;*/
        border-collapse: separate;
        /*text-align:left;*/
        height:20px;
    }
    .optionscolumn
    {
        text-align:left;
        padding-left:70px;
        width:80%;
    }
    .inputcolumn
    {
        text-align:center;
        width:15%;
    }
    .subheading
    {
        /*font-size:18px;
        font-weight:bold;*/
        font-style:italic;
        font-family:Verdana, "Times New Roman", Times, serif, Arial;
        background-color:#E0C5A0;
    }
    .inform
    {
        height:45px;
    }
    .arrow
    {
        font-size:30px;
        font-weight:bold;
    }
    .arrowmsg
    {
        font-size:17px;
    }
}

```

```

.alignleft
{
    float: left;
    padding-left:10px;
}
.alignright
{
    float: right;
    margin:0;
    border:solid 1px black;
    padding:0;
    position:relative;
    /*padding-right:10px;*/
    /*position:relative;*/
}
.outterdiv1
{
    display:none;
}
.wrappertable
{
    vertical-align:middle;
    width:100%;
    height:100%;
}
.wrappercell
{
    vertical-align:middle;
}
.togglediv, .togglediv1
{
    position:relative;
    right:0;
    float:right;
    border:none;
    margin:4px;
    padding:0;
}

```

Code written for queryresultview.css

```

form,div,table,td,th,input
{
    border:none;
}
img
{
    cursor:pointer;
    border:0;
    margin:0;
    padding:0;
}
.descriptor
{

```

```

        text-align:right;
        vertical-align:middle;
        height:40px;
        margin:0;
        padding-right:10px;
    }
    .field
    {
        text-align:left;
        vertical-align:middle;
        height:40px;
        margin:0;
        padding-right:10px;
    }
    div
    {
        margin-left:auto;
        margin-right:auto;
        text-align:center;
    }
    .holdHeaders td
    {
        background-color:#C8CECD;
        text-align:center;
        font-weight:bold;
        vertical-align:middle;
        font-family:Georgia, "Times New Roman", Times, serif, Arial;
    }
    .clickoption
    {
        cursor:pointer;
    }
    .buttons
    {
        cursor:pointer;
        font-size:17px;
        width:auto;
    }
    #vitalsdataselctform,#vitalsvariablesform,#hnmrdataselctform,#hnmrvariablesform,
    #pnmrdataselctform,#pnmrvariablesform,#binningdataselctform,#binningvariablesfor
    m,#eventindependentvarselectform,#datapointsselectform,#eventchoiceform,#submitreq
    uestform
    {
        display:none;
    }
    .box,.box1
    {
        width:466px;
        /*float:left;*/
        margin-top:10px;
        margin-bottom:10px;
        margin-left:10px;
        margin-right:10px;
        border:solid 1px White;
        background-color: #DADEDD;
        /*position:relative;*/
    }
    .box2,.box3

```

```

{
    width:304px;
    /*float:left;*/
    margin-top:10px;
    margin-bottom:10px;
    margin-left:10px;
    margin-right:10px;
    border:solid 1px White;
    background-color: #DAEEDD;
    /*position:relative;*/
}
.box4
{
    text-align:right;
}
.formbox, .formbox1, .formbox2, .formbox3
{
    float:left;
    position:relative;
}
.formbox1, .formbox2
{
    clear:left;
    position:relative;
}
.formbox4
{
    float:right;
}
.caption
{
    width:100%;
    height:20px;
    padding-top:4px;
    padding-bottom:4px;
    background-color: #747BAD;
    font-size:17px;
    font-weight:bold;
    color:White;
    vertical-align:middle;
}
.outerdiv, .outerdiv1
{
    width:100%;
    display:block;
}
.innerdiv
{
    position:relative;
    margin-left:auto;
    margin-right:auto;
    text-align:center;
    /*display:table-cell;*/
    vertical-align:middle;
    height:200px;
    width:100%;
    overflow-x:hidden;
    overflow-y:scroll;
}

```



```

        /*padding-right:10px;*/
    }
    .selection th
    {
        padding-top:3px;
        padding-bottom:3px;
    }
    .selection td
    {
        vertical-align:middle;
        padding-top:3px;
        padding-bottom:3px;
        font-family:Verdana, "Times New Roman", Times, serif, Arial;
        border-width: 1px;
        border-spacing: 2px;
        border-style: outset;
        border-color:#E0C5A0;
        /*border-color:Gray;*/
        border-collapse: separate;
        /*text-align:left;*/
        height:20px;
    }
    .optionscolumn
    {
        text-align:left;
        padding-left:70px;
        width:80%;
    }
    .inputcolumn
    {
        text-align:center;
        width:15%;
    }
    .subheading
    {
        font-style:italic;
        font-family:Verdana, "Times New Roman", Times, serif, Arial;
        background-color:#E0C5A0;
    }
    .inform
    {
        height:45px;
    }
    .arrow
    {
        font-size:30px;
        font-weight:bold;
    }
    .arrowmsg
    {
        font-size:17px;
    }
    .alignleft
    {
        float: left;
        padding-left:10px;
    }
    .alignright

```

```

{
    float:right;
    margin:0;
    border:solid 1px black;
    padding:0;
    position:relative;
}
.outertdiv1
{
    display:none;
}
.wrappertable
{
    vertical-align:middle;
    width:100%;
    height:100%;
}
.wrappercell
{
    vertical-align:middle;
}
.togglediv,.togglediv1
{
    position:relative;
    right:0;
    float:right;
    border:none;
    margin:4px;
    padding:0;
}
.datatable table
{
    border:solid 1px White;
    width:auto;
}
.datatable
{
    margin:20px;
    padding:0;
}
.datatable td
{
    vertical-align:middle;
    padding-top:3px;
    padding-bottom:3px;
    font-family:Verdana, "Times New Roman", Times, serif, Arial;
    border-width: 1px;
    border-spacing: 2px;
    border-style: outset;
    border-color:#E0C5A0;
    /*border-color:Gray;*/
    border-collapse: separate;
}
.datatable th
{
    font-size:17px;
    font-weight:bold;
    color:White;
}

```

```

padding-top:4px;
padding-bottom:4px;
background-color: #747BAD;
border:solid 1px White;
}
.datatable div
{
width:200px;
text-align:center;
}
.formheading
{
padding-left:20px;
}
.message
{
padding-left:20px;
font-size:18px;
font-family:Verdana, "Times New Roman", Times, serif, Arial;
}
.downloadicons
{
width: 27px;
height: 27px;
cursor:pointer;
}
#downloadexcel
{
background:#CCCC99 url(../images/excel.png) no-repeat center;
}
#downloadcsv
{
background:#CCCC99 url(../images/csv.png) no-repeat center;
}
#printtable
{
background:#CCCC99 url(../images/print.png) no-repeat center;
}
#message1
{
float:left;
clear:left;
padding-left:20px;
padding-bottom:20px;
height:2px;
display:block;
}
.printonly,hr,.pagebreakafter,#querysaveform,#querysavemsg
{
display:none;
}
#holdbuttons
{
clear:left;
padding-left:20px;
}
/*query Export Pagination code start*/
.queryresult_pagination_bar

```

```

{
    margin-left:20px;
    border:0;
    padding:0;
    width:1000px;
}
.queryresultchangePage
{
    cursor:pointer;
    font-weight:bold;
}
.queryresultchangePage:hover
{
    background-color:Gray;
}
.queryresultselectedPage
{
    background-color:Gray;
}
.queryresult_pagination_bar td
{
    font-family:Verdana, "Times New Roman", Times, serif, Arial;
    text-align:left;
    font-size:17px;
}
.queryresult_pagination_bar select
{
    font-size:17px;
}
td.queryresultoptions
{
    text-align:left;
    font-family:Verdana, "Times New Roman", Times, serif, Arial;
    font-weight:bold;
    padding-right:10px;
}
/*query Export Pagination code end*/

```

Code written for queryresultviewprint.css

```

table
{
    border-collapse: collapse;
    border: 0;
    margin: 0;
    padding: 0;
    width: auto;
    table-layout:fixed;
}
tr
{
    text-align: center;
}
img

```

```

{
border: 0;
}
td
{
border: 0;
margin: 0;
padding: 0;
font-size: 15px;
font-family: Georgia, "Times New Roman" , Times, serif, Arial;
color: #000000;
text-align: center;
vertical-align: middle;
}
th
{
font-family: Georgia, "Times New Roman" , Times, serif, Arial;
background-color: #C8CECD;
border-right: solid 1px #000000;
border-bottom: solid 1px #000000;
border-left: none;
border-top: none;
text-align: center;
vertical-align: middle;
margin: 0;
padding: 0;
}
.datatable table
{
border:solid 2px Black;
}
.datatable td
{
vertical-align:middle;
font-family:Verdana, "Times New Roman", Times, serif, Arial;
border-width: 1px;
border-spacing: 2px;
border-style: outset;
border-color:#000000;
/*border-color:Gray;*/
border-collapse: separate;
}
.datatable th
{
font-size:17px;
font-weight:bold;
padding-top:4px;
padding-bottom:4px;
background-color: #FFFFFF;
}
.datatable div
{
width:200px;
text-align:center;
}
.pagebreakafter
{
page-break-after:always;
}

```

```

}
#banner, .formheading, .noprint, .message, .box
{
    display:none;
}

```

Code written for simpleexport.css

```

.descriptor
{
    text-align:right;
    vertical-align:middle;
    height:40px;
    margin:0;
    border:0;
    padding-right:10px;
}
.field
{
    text-align:left;
    vertical-align:middle;
    height:40px;
    margin:0;
    border:0;
    padding-right:10px;
}
.excelfiles
{
    border-top:solid 1px #000000;
    border-left:solid 1px #000000;
    padding:0;
}
.excelfiles td
{
    border-right:solid 1px #000000;
    border-bottom:solid 1px #000000;
    border-left:none;
    border-top:none;
    text-align:center;
    vertical-align:middle;
    padding-left:5px;
    padding-right:5px;
    height:40px;
}
div
{
    text-align:center;
}
.evenRow
{
    background-color:#FFFFFF;
}
.oddRow
{
    background-color:#F3F3F3;
}

```

```

.holdHeaders td
{
    background-color:#C8CECD;
    text-align:center;
    font-weight:bold;
    vertical-align:middle;
    font-family:Georgia, "Times New Roman", Times, serif, Arial;
}
.clickoption
{
    cursor:pointer;
}
.numbers
{
    font-family:Arial;
}
#more_options
{
    display:none;
}

```

Code written for survey.css

```

#surveymessage
{
    background-color:Black;
    width:900px;
    padding:20px;
    margin:0 auto;
}
.surveynotfinished
{
    text-align:left;
    color:Red;
}
.surveyfinished
{
    text-align:center;
    color:Green;
}
.buttons
{
    font-weight:bold;
    font-family:Georgia,"Times New Roman",Times,serif,Arial;
}
.surveyholder
{
    border: 1px solid black;
    padding:40px;
}

```

Code written for tablesorter.css

```
/*
#myqueriestable .evenRow {
    background-color: Gray;
}
#myqueriestable .oddRow {
    background-color: #6E6E6E;
}
*/
#myqueriestable .header {
    background-image: url(../images/bg.png);
    background-repeat: no-repeat;
    background-position:2% 50%;
}
#myqueriestable .headerSortUp {
    background-image: url(../images/asc.png);
    background-repeat: no-repeat;
    background-position:2% 50%;
}
#myqueriestable .headerSortDown {
    background-image: url(../images/desc.png);
    background-repeat: no-repeat;
    background-position:2% 50%;
}

#myqueriestable {
    border: solid 1px White;
    height:100%;
}
#myqueriestable th {
    text-align: center;
    cursor:pointer;
    font-size:16px;
    /*padding: 5px;*/
    /*background-color: #909090;*/
    background-color:#747BAD;
    color:White;
    border:solid 1px White;
    height:30px;
}
#myqueriestable .editquery
{
    border-right:hidden;
}
.editquery,.deletequery,.removequery
{
    cursor:pointer;
}
.editquery div,.deletequery div,.removequery div
{
    width:10px;
}
#myqueriestable td
{
```



```

        color: Black;
        padding: 5px;
        border:solid 1px #E0C5A0;
        font-size:16px;
    }
#myqueriestable .queryname
{
    color:Blue;
    text-decoration:underline;
    cursor:pointer;
}
#myqueriestable .savedat
{
    font-family:Times New Roman Arial Verdana Garamond;
}

```

The SCCL directory in cgi-bin hosts the python and html files for the website and they are listed in Figure 51.

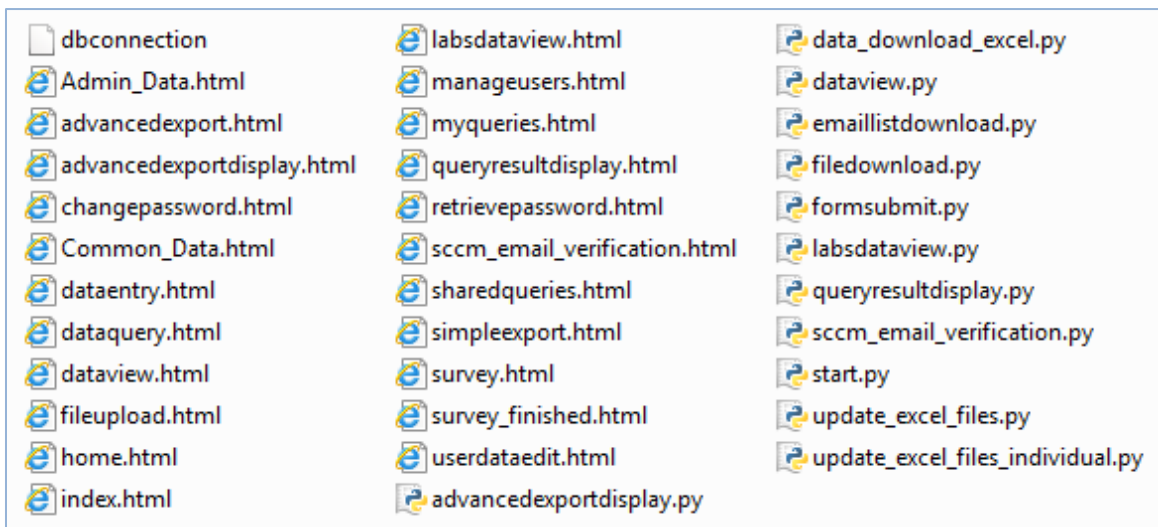


Figure 51 List of files in SCCL directory in cgi-bin

Code written for all the files listed follows except for the retrievepassword.html, sccm_email_verification.html, emaillistdownload.py and sccm_email_verification.py as they are either incomplete or do not contribute to the project. Python files import the actual code from binary files stored in secure directory as modules. The code of the python file before conversion to binary file is presented here. Password data encryption

code is removed from the python file for security purposes. Python code is very specific about tabs and spaces which can't be displayed properly in width limited documents.

Please contact author if you need original files or anything else related to the code.

Contents of dbconnection

```
HostAddress  
DatabasePort  
UserIDforDatabase  
UserPasswordforDatabase  
DatabaseName
```

Code written for Admin_Data.html

```
<!-- Don't change anything below this until next comment -->  
    <table id="entirepage">  
        <tbody>  
            <tr>  
                <td class="tableholder">  
                    <table>  
                        <tbody>  
                            <tr>  
                                <td class="testwebsite">&nbsp;</td>  
                                <td  
onclick="javascript:openPage('logout')" class="logout">  
                                    Log Out  
                                </td>  
                            </tr>  
                        </tbody>  
                    </table>  
                </td>  
            </tr>  
            <tr>  
                <td class="tableholder">  
                    <table id="banner" summary="This table defines the  
banner">  
                        <tbody>  
                            <tr>  
                                <td id="ubanner">  
                                    &nbsp;<br /><span style="font-  
size:14px;font-weight:bold;float:left;clear:left">Surgical Critical Care  
Laboratory</span>  
                                </td>  
                            </tr>  
                        </tbody>  
                    </table>  
                </td>  
            </tr>  
        </tbody>  
    </table>
```

```

        </table>
    </td>
</tr>
<!-- Changes made -->
<tr>
    <td class="menubar">
        <div class="aligncenter" id="menubarholder">
            <table class="navbar">
                <tbody>
                    <tr id="holdnavbaritems">
                        <td id="home" class="menuVisible"
onmouseover="expand(this);" onmouseout="collapse(this);">
                            <p>Home</p>
                            <div class="menuNormal">
                                <table class="menu">
                                    <tr>
                                        <td
class="menuNormal" onclick="javascript:openPage('index');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
                                            <center>Home
Page</center>
                                        </td>
                                    </tr>
                                </table>
                            </div>
                        </td>
                    </tr>
                </tbody>
            </table>
        </div>
    </td>
    <td id="forms" class="menuVisible"
onmouseover="expand(this);" onmouseout="collapse(this);">
        <p>Forms</p>
        <div class="menuNormal">
            <table class="menu">
                <tr><td
class="menuNormal" onclick="javascript:openPage('dataentry');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
                    <center>Data
Entry Forms</center>
                </td>
            </tr>
        </table>
        </div>
    </td>
    <td id="hello" class="menuVisible"
onmouseover="expand(this);" onmouseout="collapse(this);">
        <p>Data</p>
        <div class="menuNormal">
            <table class="menu">
                <tr>

```

```

class="menuNormal" onclick="javascript:openPage('simpleexport');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>Simple Export</center>
</td>
</tr>
<tr>
<td
class="menuNormal" onclick="javascript:openPage('advancedexport');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>Advanced Export</center>
</td>
</tr>
<tr>
<td
class="menuNormal" onclick="javascript:openPage('myqueries');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>My
Queries</center>
</td>
</tr>
<tr>
<td
class="menuNormal" onclick="javascript:openPage('sharedqueries');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>Shared Queries</center>
</td>
</tr>
</table>
</div>
</td>
<td id="files" class="menuVisible"
onmouseover="expand(this);" onmouseout="collapse(this);">
<p>Files</p>
<div class="menuNormal">
<table class="menu">
<tr>
<td
class="menuNormal" onclick="javascript:openPage('fileupload');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>File Upload</center>
</td>
</tr>
</table>
</div>
</td>
<td id="survey"
class="menuVisible" onmouseover="expand(this);" onmouseout="collapse(this);">
<p>Survey</p>

```

```

        <div class="menuNormal">
            <table class="menu">
                <tr>
                    <td
class="menuNormal" onclick="javascript:openPage('survey');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>cSurge Evaluation Survey</center>
                    </td>
                </tr>
            </table>
        </div>
    </td>
    <td id="admin" class="menuVisible"
onmouseover="expand(this);" onmouseout="collapse(this);">
        <p>Admin</p>
        <div class="menuNormal">
            <table class="menu">
                <tr>
                    <td
class="menuNormal" onclick="javascript:openPage('manageusers');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>Manage Users</center>
                </td>
            </tr>
        </table>
    </div>
    </td>
</tr>
</tbody>
</table>
</div>
</td>
</tr>
<!-- Don't change anything above this -->

```

Code written for advancedexport.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Advanced Data Export</title>
    <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
    <link rel="stylesheet" type="text/css" href="stylesheets/advancedexport.css"
media="screen" />

```

```

    <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
    <script language="javascript" type="text/javascript"
src="js/advancedexport.js"></script>
</head>
<body onload="javascript:loadGroupsAdvancedExport();">
#Common_Data#
    <tr>
        <td class="formheading">
            Data - MONR Advanced Export</td>
    </tr>
    <tr>
        <td>
            <div class="aligncenter">
                <table>
                    <tbody>
                        <tr>
                            <td>#greetings#</td>
                        </tr>
                        <tr>
                            <td id="message1" class="message">&nbsp;</td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </td>
    </tr>
</tbody>
</table>
<div class="aligncenter">
    <div id="message2" style="overflow:auto">&nbsp;</div>
    <form class="formbox" id="groupsform" name="groupsform" action="."
method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
        <div class="box">
            <div class="caption">Select Groups</div>
            <div id="groups" class="outerdiv">
                &nbsp;<input type="text" value="<input type="text" value="
            </div>
        </div>
    </form>
    <form class="formbox" id="subjectsform" name="subjectsform" action="."
method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
        <div class="box">
            <div class="caption">Select Subjects</div>
            <div id="subjects" class="outerdiv">
                <div class="innerdiv">
                    <table class="wrappertable"><tr><td class="wrappercell">
                        <table class="selection">
                            <tbody>
                                <tr class="oddRow">
                                    <td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> Please select</span></td>
                                </tr>
                            </tbody>
                        </table>
                    </td></tr></table>
                </div>
            </div>
        </div>
    </form>

```

```

        </div>
    </div>
</div>
</form>
<form class="formbox1" id="vitalsdataselectform"
name="vitalsdataselectform" action="." method="post" enctype="application/x-www-
form-urlencoded" onsubmit="javascript:return false">
    <div class="box1">
        <div class="caption"><div class="togglediv"></div><div>Select Vitals</div></div>
        <div id="vitalsselect" class="outerdiv1">
            &nbsp;
        </div>
    </div>
</form>
<form class="formbox" id="vitalsvariablesform" name="vitalsvariablesform"
action="." method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
    <div class="box">
        <div class="caption">Select Vitals variables</div>
        <div id="vitalsvariablesselect" class="outerdiv1">
            <div class="innerdiv">
                <table class="wrappertable"><tr><td class="wrappercell">
                <table class="selection">
                    <tbody>
                        <tr class="oddRow">
                            <td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> Please select</span></td>
                            </tr>
                    </tbody>
                </table>
                </td></tr></table>
            </div>
        </div>
    </form>
<form class="formbox1" id="hnmrdataselectform" name="hnmrdataselectform"
action="." method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
    <div class="box1">
        <div class="caption"><div class="togglediv"></div><div>Select HNMR data</div></div>
        <div id="hnmrselect" class="outerdiv1">
            &nbsp;
        </div>
    </div>
</form>
<form class="formbox" id="hnmrvariablesform" name="hnmrvariablesform"
action="." method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
    <div class="box">
        <div class="caption">Select HNMR variables</div>
        <div id="hnmrvariablesselect" class="outerdiv1">
            <div class="innerdiv">
                <table class="wrappertable"><tr><td class="wrappercell">
                <table class="selection">
                    <tbody>
                        <tr class="oddRow">

```

```

                <td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> Please select</span></td>
                </tr>
            </tbody>
        </table>
    </td></tr></table>
</div>
</div>
</div>
</form>
<form class="formbox1" id="pnmrdataselectform" name="pnmrdataselectform"
action="." method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
    <div class="box1">
        <div class="caption"><div class="togglediv"></div><div>Select PNMR data</div></div>
        <div id="pnmrselect" class="outerdiv1">
            &nbsp;
        </div>
    </div>
</form>
<form class="formbox" id="pnmrvariablesform" name="pnmrvariablesform"
action="." method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
    <div class="box">
        <div class="caption">Select PNMR variables</div>
        <div id="pnmrvariablesselect" class="outerdiv1">
            <div class="innerdiv">
                <table class="wrappertable"><tr><td class="wrappercell">
                    <table class="selection">
                        <tbody>
                            <tr class="oddRow">
                                <td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> Please select</span></td>
                                </tr>
                        </tbody>
                    </table>
                </td></tr></table>
            </div>
        </div>
    </form>
<form class="formbox1" id="binningdataselectform"
name="binningdataselectform" action="." method="post" enctype="application/x-www-
form-urlencoded" onsubmit="javascript:return false">
    <div class="box1">
        <div class="caption"><div class="togglediv"></div><div>Select Binning
data</div></div>
        <div id="binningselect" class="outerdiv1">
            &nbsp;
        </div>
    </div>
</form>
<form class="formbox" id="binningvariablesform"
name="binningvariablesform" action="." method="post" enctype="application/x-www-
form-urlencoded" onsubmit="javascript:return false">
    <div class="box">

```



```

<div class="caption">Select Binning variables</div>
<div id="binningvariablesselect" class="outerdiv1">
  <div class="innerdiv">
    <table class="wrappertable"><tr><td class="wrappercell">
      <table class="selection">
        <tbody>
          <tr class="oddRow">
            <td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> Please select</span></td>
          </tr>
        </tbody>
      </table>
    </td></tr></table>
  </div>
</div>
</form>
<form class="formbox2" id="eventindependentvarselectform"
name="eventindependentvarselectform" action="." method="post"
enctype="application/x-www-form-urlencoded" onsubmit="javascript:return false">
  <div class="box2">
    <div class="caption"><div class="toggediv1"></div><div>Select event-independent
variables</div></div>
    <div id="eventindependentvarselect" class="outerdiv1">
      &nbsp;
    </div>
  </div>
</form>
<form class="formbox3" id="datapointsselectform"
name="datapointsselectform" action="." method="post" enctype="application/x-www-
form-urlencoded" onsubmit="javascript:return false">
  <div class="box3">
    <div class="caption"><div class="toggediv"></div><div>Select data points'
frequency</div></div>
    <div id="datapointsselect" class="outerdiv1">
      &nbsp;
    </div>
  </div>
</form>
<form class="formbox3" id="eventchoiceform" name="eventchoiceform"
action="." method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
  <div class="box3">
    <div class="caption">Select events</div>
    <div id="eventchoice" class="outerdiv1">
      <div class="innerdiv">
        <table class="wrappertable"><tr><td class="wrappercell">
          <table class="selection">
            <tbody>
              <tr class="oddRow">
                <td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> Please select "Selected events"
option to populate the events</span></td>
              </tr>
            </tbody>
          </table>
        </td></tr></table>
      </div>
    </div>
  </div>
</form>

```

```

                </td></tr></table>
            </div>
        </div>
    </div>
</form>
<form class="formbox4" id="submitrequestform" name="submitrequestform"
action="." method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
    <div class="box4">
        <button type="button" name="submitrequest" id="submitrequest"
value="Submit" onclick="javascript:advancedExportWindow()"
class="buttons">Submit</button>
    </div>
</form>
</div>
</body>
</html>

```

Code written for advancedexportdisplay.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Data Export</title>
    <link rel="icon" type="image/x-icon" href="/images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon" href="/images/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="/stylesheets/main.css"
media="screen" />
    <link rel="stylesheet" type="text/css" href="/stylesheets/advancedexport.css"
media="screen" />
    <link rel="stylesheet" type="text/css"
href="/stylesheets/advancedexportview.css" media="screen" />
    <link rel="stylesheet" type="text/css"
href="/stylesheets/advancedexportviewprint.css" media="print" />
</head>
<body
onload="javascript:opener.queryAdvancedExport(document.advancedexportdisplayform.nu
mberofdatapoints.value);">
    <table id="entirepage">
        <tbody>
            <tr>
                <td class="tableholder">
                    <table id="banner" summary="This table defines the banner">
                        <tbody>
                            <tr>
                                <td id="ubanner">
                                    &nbsp;
                                </td>
                                <td class="smallcaption">
                                    SCCL<br /><span style="font-size:14px;font-
weight:bold">Surgical Critical Care Laboratory</span>
                                </td>
                            </tr>
                        </tbody>
                    </table>
                </td>
            </tr>
        </tbody>
    </table>

```

```

        </tr>
      </tbody>
    </table>
  </td>
</tr>
<tr>
  <td class="formheading" style="text-align:left">Data - MONR
Advanced Export Data View</td>
</tr>
<tr>
  <td id="advancedexportdataholder"
class="advancedexportdataholder">
    <div id="message1" class="message">Please wait.....</div>
    <div id="holdbuttons" class="noprint">
      <div style="float:left;clear:left;padding-
bottom:10px;padding-top:10px">
        <button class="buttons" title="Switch to data
selection page to change your selection" type="button" name="showparent"
id="showparent" value="showparent" onclick="javascript:opener.focus()">Switch to
main page</button>
        <button class="buttons" title="Close this window
and go back to data selection page" type="button" name="closethis" id="closethis"
value="closethis"
onclick="javascript:opener.alladvancedExportPopup.close();opener.focus();">Close
this window</button>
        <button class="buttons" title="Save your query"
type="button" name="showqueryform" id="showqueryform" value="showqueryform"
onclick='javascript:opener.showQueryFormAndIDs();'>Save query</button>
      </div>
      <div class="downloadicons" style="float:left;padding-
left:5px;padding-bottom:10px;padding-top:10px" id="printtable" title="Print
visible data" onclick="javascript>window.print();return false;">
        &nbsp;
      </div>
      <div id="holddownloadicons" style="float:left">
        &nbsp;
      </div>
    </div>
    <!--
    <div id="querysavemsg">
      &nbsp;
    </div>
    -->
    <form id="savequeryform" name="savequeryform"
autocomplete="off" action="." method="post" enctype="application/x-www-form-
urlencoded" onsubmit="javascript:return false">
      <div id="querysaveform" class="noprint">
        <!-- <div id="querywrapper" class="wrapform">
          <div>
            <div class="descriptor" style="padding-
bottom:10px">Name of the query</div>
            <div
style="width:468px;height:24px"><input type="text" name="queryname" id="queryname"
maxlength="100" /></div>
          </div>
          <div>
            <div class="descriptor" style="padding-
bottom:10px">Description of the query</div>

```

```

                <div
style="width:468px;height:72px"><textarea name="querydescription"
id="querydescription" onfocus="javascript:this.select();">No description
entered<</textarea></div>
            </div>
            <div>
                <div class="descriptor">Share with</div>
                <div class="box">
                    <div id="userid"
class="outerdiv">&nbsp;</div>
                </div>
            </div>
            <div style="text-align:right">
                <button type='button' id='cancelsavequery'
name='cancelsavequery' value="cancelsavequery" class='buttons' title='Cancel
Saving Query' onclick='javascript:opener.closeSaveQueryForm();'>Cancel</button>
                <button type='submit' id='savequery'
name='savequery' value="savequery" class='buttons' title='Save query'
onclick='javascript:opener.saveQuery();return false'>Save</button>
            </div>
        </div> -->
    </div>
</form>
<form id="advancedexportdisplayform"
name="advancedexportdisplayform" autocomplete="off" action="." method="post"
enctype="application/x-www-form-urlencoded" onsubmit="javascript:return false">
    <div id="pagination" class="noprint">
        <table class="advancedexport_pagination_bar"
style="clear:left">
            <tr>
                <td id="datasort">
                    <select id="numberofdatapoints"
name="numberofdatapoints"
onchange='javascript:opener.advancedexportchangePage(this.id,1)'>
                        <option value="1000">1000</option>
                        <option value="750">750</option>
                        <option value="500">500</option>
                        <option value="250"
selected="selected">250</option>
                        <option value="100">100</option>
                        <option value="50">50</option>
                        <option value="10">10</option>
                    </select>
                    rows/page
                </td>
                <td>
                    <span
id="advancedexport_totalrows"></span>
                </td>
                <td>
                    <span
id="advancedexport_totalpages"></span>
                </td>
                <td id="advancedexport_go_to_page">
                </td>
                <td id="advancedexportOptions"
class="advancedexportoptions">
                </td>
            </tr>
        </table>
    </div>
</form>

```

```

                </tr>
            </table>
        </div>
    </form>
    <div id="datatable" class="datatable"><span style="font-
size:30px">Please wait...</span></div>
    <div id="message3" class="message"
style="height:0px">&nbsp;</div>
        </td>
    </tr>
</tbody>
</table>
<div class="pagebreakafter"></div>
<hr/>
<p class="printonly">
    Surgical Critical Care laboratory, University of Minnesota
</p>
<p class="printonly">
    If you are not authorized to view this data, please destroy the document
immediately
</p>
</body>
</html>

```

Code written for changepassword.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Home</title>
    <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
    <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
    <script language="javascript" type="text/javascript"
src="js/changepassword.js"></script>
</head>
<body onload="javascript:document.changepasswordform.newpassword.focus()">
    <table id="entirepage">
        <tbody>
            <tr>
                <td class="tableholder">
                    <table>
                        <tbody>
                            <tr>
                                <td>&nbsp;</td>
                                <td onclick="javascript:openPage('logout')"
class="logout">
                                    Log Out
                                </td>
                            </tr>
                        </tbody>
                    </table>
                </td>
            </tr>
        </tbody>
    </table>

```

```

        </tbody>
    </table>
</td>
</tr>
<tr>
    <td class="tableholder">
        <table id="banner" summary="This table defines the banner">
            <tbody>
                <tr>
                    <td id="ubanner">
                        &nbsp;
                    </td>
                    <td class="smallcaption">
                        SCCL <span style="font-size:16px;font-
weight:bold">Surgical Critical Care Laboratory</span>
                    </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
<tr>
    <td class="formheading">Please choose a new password</td>
</tr>
<tr>
    <td>
        <table>
            <tbody>
                <tr>
                    <td>#greetings#</td>
                </tr>
                <tr>
                    <td class="tableholder">
                        <div class="aligncenter"
style="width:750px;text-align:center">
                            <form id="changepasswordform"
name="changepasswordform" action="." method="post" enctype="application/x-www-
form-urlencoded" onsubmit="javascript:if(!validateForm()){return false;}">
                                <fieldset class="loginform">
                                    <legend class="mainform">Create new
password</legend>
                                    <table summary="User login form">
                                        <tr>
                                            <td>
                                                <label for="newpassword">
                                                    New Password:
                                                </label>
                                            </td>
                                            <td>
                                                <input type="password"
name="newpassword" id="newpassword" maxlength="30" size="20" tabindex="1"
title="Enter User ID"/>
                                            </td>
                                        </tr>
                                        <tr>
                                            <td>
                                                <label
for="reenterpassword">
                                                    Re-enter Password:
                                                </label>

```

```
                </td>
                <td>
                    <input type="password"
name="reenterpassword" id="reenterpassword" maxlength="30" size="20" tabindex="2"
title="Enter Password"/>
                </td>
                <td>
                    <button class="buttons"
type="submit" name="submitpasswords" id="submitpasswords" value="submitpasswords"
                    tabindex="20"
title="Click here after entering the passwords">
                        Submit
                    </button>
                    <input type="hidden"
name="changepassword" id="changepassword" value="changepassword" />
                </td>
            </tr>
        </table>
    </fieldset>
</form>
</div>
</td>
</tr>
<tr>
<td class="tableholder">
    <table>
    <tbody>
        <tr>
            <td id="message1" class=message">
                &nbsp;
            </td>
        </tr>
    </tbody>
    </table>
</td>
</tr>
</tbody>
</table>
</td>
</tr>
</tbody>
</table>
</body>
</html>
```

Code written for Common_data.html

```
<!-- Don't change anything below this until next comment -->
    <table id="entirepage">
        <tbody>
            <tr>
                <td class="tableholder">
                    <table>
```

```

        <tbody>
          <tr>
            <td class="testwebsite">&nbsp;</td>
            <td
onclick="javascript:openPage('logout')" class="logout">
              Log Out
            </td>
          </tr>
        </tbody>
      </table>
    </td>
  </tr>
  <tr>
    <td class="tableholder">
      <table id="banner" summary="This table defines the
banner">
        <tbody>
          <tr>
            <td id="ubanner">
              &nbsp;</td>
            <td class="smallcaption">
              SCCL<br /><span style="font-
size:14px;font-weight:bold;float:left;clear:left">Surgical Critical Care
Laboratory</span>
            </td>
          </tr>
        </tbody>
      </table>
    </td>
  </tr>
  <!-- Changes made -->
  <tr>
    <td class="menubar">
      <div class="aligncenter" id="menubarholder">
        <table class="navbar">
          <tbody>
            <tr id="holdnavbaritems">
              <td id="home" class="menuVisible"
onmouseover="expand(this);" onmouseout="collapse(this);">
                <p>Home</p>
                <div class="menuNormal">
                  <table class="menu">
                    <tr>
                      <td
class="menuNormal" onclick="javascript:openPage('index');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">

```



```

Account</center>
</td>
</tr>
</table>
</div>
</td>
<td id="forms" class="menuVisible"
onmouseover="expand(this);" onmouseout="collapse(this);">
<p>Forms</p>
<div class="menuNormal">
<table class="menu">
<tr><td
class="menuNormal" onclick="javascript:openPage('dataentry');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>Data
Entry Forms</center>
</td>
</tr>
</table>
</div>
</td>
<td id="hello" class="menuVisible"
onmouseover="expand(this);" onmouseout="collapse(this);">
<p>Data</p>
<div class="menuNormal">
<table class="menu">
<tr>
<td
class="menuNormal" onclick="javascript:openPage('simpleexport');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>Simple Export</center>
</td>
</tr>
<tr>
<td
class="menuNormal" onclick="javascript:openPage('advancedexport');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>Advanced Export</center>
</td>
</tr>
<tr>
<td
class="menuNormal" onclick="javascript:openPage('myqueries');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">
<center>My
Queries</center>
</td>
</tr>
<tr>
<td
class="menuNormal" onclick="javascript:openPage('sharedqueries');">

```

```

onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">

<center>Shared Queries</center>
</td>
</tr>
</table>
</div>
</td>
<td id="files" class="menuVisible"
onmouseover="expand(this);" onmouseout="collapse(this);">
<p>Files</p>
<div class="menuNormal">
<table class="menu">
<tr>
<td
class="menuNormal" onclick="javascript:openPage('fileupload');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">

<center>File Upload</center>
</td>
</tr>
</table>
</div>
</td>
<td id="survey"
class="menuVisible" onmouseover="expand(this);" onmouseout="collapse(this);">
<p>Survey</p>
<div class="menuNormal">
<table class="menu">
<tr>
<td
class="menuNormal" onclick="javascript:openPage('survey');"
onmouseover="javascript:highlight(this);"
onmouseout="javascript:unhighlight(this);">

<center>cSurge Evaluation Survey</center>
</td>
</tr>
</table>
</div>
</td>
</tr>
</tbody>
</table>
</div>
</td>
</tr>
<!-- Don't change anything above this -->

```

Code written for dataentry.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Data Entry</title>
  <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
  <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
  <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
  <link rel="stylesheet" type="text/css" href="stylesheets/dataentry.css"
media="screen" />
  <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
  <script language="javascript" type="text/javascript"
src="js/dataentry.js"></script>

</head>
<body onload="javascript: onLoadScripts();">
#Common_Data#
  <tr>
    <td class="formheading">Forms - MONR Data Entry Form</td>
  </tr>
  <tr>
    <td>
      <div class="aligncenter">
        <table><tbody>
          <tr>
            <td>#greetings#</td>
          </tr>
          <tr>
            <td class="tableholder">
              <form id="form1" name="form1" action="." method="post"
autocomplete="off" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
                <fieldset>
                  <legend>Subject Identifiers</legend>
                  <table summary="This table holds MONR identifier form"
id="mainform">
                    <tr>
                      <td class="tableholder">
                        <table>
                          <tbody>
                            <tr>
                              <td class="descriptor1"
title="ID number of the MONR subject">
                                <label for="pig_id">
                                  ID:
                                </label>
                              </td>
                              <td class="field1" title="ID
number of the MONR subject">
                                <input type="text"
name="pig_id" id="pig_id" maxlength="6" size="10"
onkeyup="javascript:getSuggestions();"
onblur="javascript:idcheck(this,this.id,'Pig ID');" /><br />
                                <div
id="holdsuggestions"></div>

```

```

        </td>
        <td class="descriptor1"
            title="Group the MONR subject belongs to">
            <label for="group">
                Group:
            </label>
        </td>
        <td class="field1"
            title="Group the MONR subject belongs to">
            <select name="group">
                <option value=""
                    selected="selected">Select</option>
                <option value="1">1
                <option value="2">2
                <option value="3">3
                <option value="4">4
            </select>
        </td>
        <td class="descriptor1"
            title="Weight of the MONR subject">
            <label for="weight">
                Weight:
            </label>
        </td>
        <td class="field1"
            title="Weight of the MONR subject">
            <input type="text"
                name="weight" id="weight" maxlength="4" size="10" />
        </td>
        <td class="descriptor1"
            title="Total volume of blood removed">
            <label for="bloodremoved">
                Blood Removed:
            </label>
        </td>
        <td class="field1"
            title="Total volume of blood removed">
            <input type="text"
                name="bloodremoved" id="bloodremoved" maxlength="4" size="10" />
        </td>
    </tr>
    <tr>
        <td class="descriptor1"
            title="Total volume of O.R fluids given">
            <label for="orfluids">
                O.R. Fluids:
            </label>
        </td>
        <td class="field1"
            title="Total volume of O.R fluids given">
            <input type="text"
                name="orfluids" id="orfluids" maxlength="4" size="10" />
        </td>
    </tr>

```

```

        <td class="descriptor1"
title="Total volume of blood given">
        <label for="bloodgiven">
            Blood Given:
        </label>
    </td>
    <td class="field1"
title="Total volume of blood given">
        <input type="text"
name="bloodgiven" id="bloodgiven" maxlength="4" size="10" />
    </td>
    <td class="descriptor1"
title="Total volume of Lactate Ringers given">
        <label for="lrgiven">
            Lactated Ringers:
        </label>
    </td>
    <td class="field1"
title="Total volume of Lactate Ringers given">
        <input type="text"
name="lrgiven" id="lrgiven" maxlength="4" size="10" />
    </td>
    <td class="descriptor1"
title="Total volume of fluids given">
        <label for="totalvolume">
            Total volume:
        </label>
    </td>
    <td class="field1"
title="Total volume of fluids given">
        <input type="text"
name="totalvolume" id="totalvolume" maxlength="4" size="10" />
    </td>
</tr>
</tbody>
</table>
</td>
</tr>
<tr>
    <td class="tableholder">
        <table>
            <tbody>
                <tr>
                    <td title="Date the experiment
started" class="startdateAdjust">
                        <fieldset
class="innerfields">
                            <legend
class="elements">Experiment Start Date</legend>
                            <div id="startdate"
class="dategroup">
                                <label>
                                    <span
class="mmdyy">MM </span>
                                    <input
type="text" class="daymonth" name="start_month" id="start_month" maxlength="2"
disabled="disabled"/></label>

```

```

class="mddyy">DD </span>
type="text" class="daymonth" name="start_day" id="start_day" maxlength="2"
disabled="disabled" /></label>
class="mddyy">YYYY </span>
type="text" class="year" name="start_year" id="start_year" maxlength="4"
disabled="disabled" /></label>
</div>
</fieldset>
</td>
<td title="Date the experiment
ended" class="enddateAdjust">
class="innerfields">
class="elements">Experiment End Date</legend>
class="dategroup">
class="mddyy">MM </span>
type="text" class="daymonth" name="end_month" id="end_month" maxlength="2"
/></label>
class="mddyy">DD </span>
type="text" class="daymonth" name="end_day" id="end_day" maxlength="2" /></label>
class="mddyy">YYYY </span>
type="text" class="year" name="end_year" id="end_year" maxlength="4" /></label>
</div>
</fieldset>
</td>
<td title="MONR subject died
during the experiment or not">
class="selection">
selected="selected">Select</option>
value="NO">No</option>
value="YES">Yes</option>
</td>

```

```

        </select>
    </td>
</tr>
</tbody>
</table>
</td>
</tr>
<tr>
    <td class="tableholder" style="padding-
bottom:10px">
        <table>
            <tbody>
                <tr>
                    <td>
                        <label>
                            Time of Death:
                        </label>
                    </td>
                    <td>
                        HH
                        <input type="text"
name="died_hours" id="died_hours" maxlength="2" size="2"
onblur="javascript:checkInt(this,this.id,'Died Hours');" />
                        MM
                        <input type="text"
name="died_minutes" id="died_minutes" maxlength="2" size="2" />
                    </td>
                    <td title="Cause of death of
the MONR subject">
                        <label for="causeofdeath">
                            Cause of Death:
                        </label>
                    </td>
                    <td>
                        <textarea
name="causeofdeath" id="causeofdeath" cols="12" rows="3"></textarea>
                    </td>
                    <td title="Overall comments
about the experiment">
                        <label
for="overallcomments">
                            Overall Comments:
                        </label>
                    </td>
                    <td>
                        <textarea
name="overallcomments" id="overallcomments" cols="12" rows="3"></textarea>
                    </td>
                </tr>
                <tr title="Only an Administrator
can use this feature">
                    <td colspan="8">
                        <div style="padding-
top:15px;text-align:center">
                            <input type="checkbox"
checked="checked" id="lockdata" name="lockdata" value="lockdata"

```

```

disabled="disabled"/> <label for="lockdata">Lock Data Export (Administrator
Only)</label>
</div>
</td>
</tr>
</tbody>
</table>
</td>
</tr>
</table>
</fieldset>
<table>
<tr>
<td class="tableholder">
<table>
<tbody>
<tr>
<td style="text-align:right"
id="buttonsholder">
<select id="selectform"
name="selectform" class="selectform" onchange="javascript:formSelect(this.value);"
disabled="disabled">
<option
selected="selected" value="none">Select Form</option>
<option
value="all">All</option>
<option
value="vitals">Vitals</option>
<option
value="labs">Labs</option>
<option
value="gcscscores">GCS Scores</option>
</select>
<button type='button'
id='editidentifiers' name='editidentifiers' class='buttons' title='Edit the MONR
subject identifier form' onclick='javascript:editform();'>
Edit
</button>
<button class="buttons"
type="button" name="reloadpage" id="reloadpage" title="Reset the MONR subject
identifier form" onclick="javascript:pageReload();">
New Form
</button>
<button class="buttons"
type="submit" name="submitidentifiers" id="submitidentifiers" value='mainform'
title="Submit the MONR
subject identifier form" onclick="javascript:submitForm('mainform');return false">
Submit
</button>
</td>
</tr>
</tbody>
</table>
<table>
<tbody>
<tr>
<td id="message1"
class="message">&nbsp;  </td>

```



```

        </tr>
      </tbody>
    </table>
  </td>
</tr>
</table>
</form>
</td>
</tr>
<!--
End of Main form and Start of Vitals form
-->
<tr>
  <td id="vitalsform" class="tableholder">
    <form id="form2" name="form2" action="." method="post"
autocomplete="off" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
      <fieldset>
        <legend>Vitals</legend>
        <table summary="This table holds MONR subject vitals
form">
          <tr>
            <td class="tableholder">
              <table>
                <tbody>
                  <tr>
                    <td class="descriptor2" title="Day of the event">
                      <label for="day">
                        Day:
                      </label>
                    </td>
                    <td class="field2" title="Day of the event">
                      <select name="day" id="day" class="selection">
                        <option value="1"
selected="selected">1</option>
                        <option value="2">2</option>
                        <option value="3">3</option>
                        <option value="4">4</option>
                      </select>
                    </td>
                    <td class="descriptor2" title="Time of the event">
                      <label for="hours">
                        Time:<br />(HH MM)
                      </label>
                    </td>
                    <td class="field2" title="Time of the event">
                      <input type="text" name="hours" id="hours"
maxlength="2" size="2" />
                      <input type="text" name="minutes" id="minutes"
maxlength="2" size="2" />
                    </td>
                    <td class="descriptor2" title="Event Comments if
any">
                      <label for="eventcomments">
                        Event<br />Comments:
                      </label>
                    </td>
                    <td title="Event Comments if any" colspan="3">

```

```

                                <textarea name="eventcomments"
id="eventcomments" cols="40" rows="2"></textarea>
                                </td>
</tr>
<tr>
    <td class="descriptor2" title="Event Name">
        <label for="event">
            Event:
        </label>
    </td>
    <td class="field2" title="Event Name">
        <select name="event" id="event"
class="selection">
            <option value="99" selected="selected">99
            (Other)</option>
                                <option value="1">1 (Baseline)</option>
                                <option value="2">2 (Shock45)</option>
                                <option value="3">3 (LR1)</option>
                                <option value="4">4 (FR1)</option>
                                <option value="5">5 (FR2)</option>
                                <option value="6">6 (FR3)</option>
                                <option value="7">7 (FR4)</option>
                                <option value="8">8 (FR5)</option>
                                <option value="9">9 (FR6)</option>
                                <option value="10">10 (FR7)</option>
                                <option value="11">11 (FR8)</option>
                                <option value="12">12 (FR9)</option>
                                <option value="13">13 (FR10)</option>
                                <option value="14">14 (FR11)</option>
                                <option value="15">15 (FR12)</option>
                                <option value="16">16 (FR13)</option>
                                <option value="17">17 (FR14)</option>
                                <option value="18">18 (FR15)</option>
                                <option value="19">19 (FR16)</option>
                                <option value="20">20 (FR17)</option>
                                <option value="21">21 (FR18)</option>
                                <option value="22">22 (FR19)</option>
                                <option value="23">23 (FR20)</option>
                                <option value="24">24 (PR24)</option>
                                <option value="25">25 (PR48)</option>
                                </select>
        </td>
    <td class="descriptor2" title="Heart Rate">
        <label for="heartrate">
            HR:
        </label>
    </td>
    <td class="field2" title="Heart Rate">
        <input type="text" name="heartrate"
id="heartrate" maxlength="3" size="10"/>
    </td>
    <td class="descriptor2" title="Arterial Systolic
Pressure">
        <label for="artsys">
            Art Sys:
        </label>
    </td>

```

```

Pressure">
    <td class="field2" title="Arterial Systolic
        <input type="text" name="artsys" id="artsys"
maxlength="3" size="10" />
    </td>
    <td class="descriptor2" title="Arterial Diastolic
        <label for="artdia">
            Art Dia:
        </label>
    </td>
    <td class="field2">
        <input type="text" name="artdia" id="artdia"
maxlength="3" size="10" />
    </td>
</tr>
<tr>
    <td class="descriptor2" title="PA Systolic
        <label for="pasys">
            PA Sys:
        </label>
    </td>
    <td class="field2" title="PA Systolic Pressure">
        <input type="text" name="pasys" id="pasys"
maxlength="3" size="10" />
    </td>
    <td class="descriptor2" title="PA Diastolic
        <label for="padia">
            PA Dia:
        </label>
    </td>
    <td class="field2">
        <input type="text" name="padia" id="padia"
maxlength="3" size="10" />
    </td>
    <td class="descriptor2" title="Wedge Pressure">
        <label for="wedgepressure">
            Wedge Press:
        </label>
    </td>
    <td class="field2" title="Wedge Pressure">
        <input type="text" name="wedgepressure"
id="wedgepressure" maxlength="3" size="10" />
    </td>
    <td class="descriptor2" title="Urine Output">
        <label for="urineoutput">
            Urine Output:
        </label>
    </td>
    <td class="field2" title="Urine Output">
        <input type="text" name="urineoutput"
id="urineoutput" maxlength="4" size="10" />
    </td>
</tr>
<tr>
    <td class="descriptor2" title="Bladder Pressure">

```

```

        <label for="bladderpres">
            Bladder Pressure:
        </label>
    </td>
    <td class="field2" title="Bladder Pressure">
        <input type="text" name="bladderpres"
id="bladderpres" maxlength="4" size="10" />
    </td>
    <td class="descriptor2" title="Temperature">
        <label for="temperature">
            Temperature:
        </label>
    </td>
    <td class="field2" title="Temperature">
        <input type="text" name="temperature"
id="temperature" maxlength="4" size="10" />
    </td>
    <td class="descriptor2" title="Cardiac Output">
        <label for="cardiacoutput">
            Cardiac Output:
        </label>
    </td>
    <td class="field2" title="Cardiac Output">
        <input type="text" name="cardiacoutput"
id="cardiacoutput" maxlength="6" size="10" />
    </td>
    <td class="descriptor2" title="Propofol">
        <label for="propofol">
            Propofol:
        </label>
    </td>
    <td class="field2" title="Propofol">
        <input type="text" name="propofol"
id="propofol" maxlength="2" size="10" />
    </td>
</tr>
<tr>
    <td class="descriptor2" title="BIS">
        <label for="bis">
            BIS:
        </label>
    </td>
    <td class="field2" title="BIS">
        <input type="text" name="bis" id="bis"
maxlength="2" size="10" />
    </td>
    <td class="descriptor2" title="FiO2">
        <label for="fio2">
            FiO2:
        </label>
    </td>
    <td class="field2" title="FiO2">
        <input type="text" name="fio2" id="fio2"
maxlength="4" size="10" />
    </td>
    <td class="descriptor2" title="StO2 (15mm)">
        <label for="sto2_15mm">
            StO2 (15mm):

```

```

        </label>
    </td>
    <td class="field2" title="StO2 (15mm)">
        <input type="text" name="sto2_15mm"
id="sto2_15mm" maxlength="3" size="10" />
    </td>
    <td class="descriptor2" title="THI (15mm)">
        <label for="thi_15mm">
            THI (15mm):
        </label>
    </td>
    <td class="field2" title="THI (15mm)">
        <input type="text" name="thi_15mm"
id="thi_15mm" maxlength="4" size="10" />
    </td>
</tr>
<tr>
    <td class="descriptor2" title="StO2 (Multi
Depth)">
        <label for="sto2_multidepth">
            StO2:<br />(Multi Depth)
        </label>
    </td>
    <td class="field2" title="StO2 (Multi Depth)">
        <input type="text" name="sto2_multidepth"
id="sto2_multidepth" maxlength="3" size="10" />
    </td>
    <td class="descriptor2" title="THI (Multi Depth)">
        <label for="thi_multidepth">
            THI:<br />(Multi Depth)
        </label>
    </td>
    <td class="field2" title="THI (Multi Depth)">
        <input type="text" name="thi_multidepth"
id="thi_multidepth" maxlength="4" size="10" />
    </td>
    <td class="descriptor2" title="Chest Tube Output">
        <label for="chesttubeoutput">
            Chest tube<br />output:
        </label>
    </td>
    <td class="field2" title="Chest Tube Output">
        <input type="text" name="chesttubeoutput"
id="chesttubeoutput" maxlength="5" size="10" />
    </td>
    <td class="descriptor2" title="Arterial pH">
        <label for="arterialph">
            Arterial pH:
        </label>
    </td>
    <td class="field2" title="Arterial pH">
        <input type="text" name="arterialph"
id="arterialph" maxlength="4" size="10" />
    </td>
</tr>
<tr>
    <td class="descriptor2" title="Arterial PCO2">
        <label for="art_pco2">

```

```

        Arterial PCO2:
    </label>
</td>
<td class="field2" title="Arterial PCO2">
    <input type="text" name="art_pco2"
id="art_pco2" maxlength="3" size="10" />
</td>
<td class="descriptor2" title="Arterial PO2">
    <label for="art_po2">
        Arterial PO2:
    </label>
</td>
<td class="field2" title="Arterial PO2">
    <input type="text" name="art_po2" id="art_po2"
maxlength="3" size="10" />
</td>
<td class="descriptor2" title="Arterial O2
Saturated">
    <label for="art_o2_sat">
        Arterial O2<br />Saturated:
    </label>
</td>
<td class="field2" title="Arterial O2 Saturated">
    <input type="text" name="art_o2_sat"
id="art_o2_sat" maxlength="3" size="10" />
</td>
<td class="descriptor2" title="Venous PO2">
    <label for="ven_po2">
        Venous PO2:
    </label>
</td>
<td class="field2" title="Venous PO2">
    <input type="text" name="ven_po2" id="ven_po2"
maxlength="3" size="10" />
</td>
</tr>
<tr>
<td class="descriptor2" title="Venous O2
Saturated">
    <label for="ven_o2_sat">
        Venous O2 Saturated:
    </label>
</td>
<td class="field2" title="Venous O2 Saturated">
    <input type="text" name="ven_o2_sat"
id="ven_o2_sat" maxlength="3" size="10" />
</td>
<td class="descriptor2" title="Hemoglobin">
    <label for="hemoglobin">
        Hemoglobin:
    </label>
</td>
<td class="field2">
    <input type="text" name="hemoglobin"
id="hemoglobin" maxlength="4" size="10" />
</td>
<td class="descriptor2" title="Lactate">
    <label for="lactate">

```

```

        Lactate:
    </label>
</td>
<td class="field2" title="Lactate">
    <input type="text" name="lactate" id="lactate"
maxlength="4" size="10" />
</td>

<td class="descriptor2" title="Base deficit">
    <label for="basedeficit">
        Base deficit:
    </label>
</td>
<td class="field2" title="Base deficit">
    <input type="text" name="basedeficit"
id="basedeficit" maxlength="5" size="10" />
</td>
</tr>
<tr>
    <td class="descriptor2" title="Specificgravity of
urine">
        <label for="specificgravity">
            Specificgravity:
        </label>
</td>
<td class="field2" title="Base deficit">
    <input type="text" name="specificgravity"
id="specificgravity" maxlength="5" size="10" />
</td>
<td class="descriptor2" title="Sodium">
    <label for="sodium">
        Sodium:
    </label>
</td>
<td class="field2" title="Sodium">
    <input type="text" name="sodium" id="sodium"
maxlength="3" size="10" />
</td>
<td class="descriptor2" title="Potassium">
    <label for="potassium">
        Potassium:
    </label>
</td>
<td class="field2" title="Potassium">
    <input type="text" name="potassium"
id="potassium" maxlength="4" size="10" />
</td>
<td class="descriptor2" title="Calcium">
    <label for="calcium">
        Calcium:
    </label>
</td>
<td class="field2" title="Calcium">
    <input type="text" name="calcium" id="calcium"
maxlength="4" size="10" />
</td>
</tr>
<tr>

```

```

        <td class="descriptor2" title="Glucose">
            <label for="glucose">
                Glucose:
            </label>
        </td>
        <td class="field2" title="Glucose">
            <input type="text" name="glucose" id="glucose"
maxlength="3" size="10" />
        </td>
    </tr>
</tbody>
</table>
</td>
</tr>
</table>
</fieldset>
<table summary="This table is used to hold user
addition form">
    <tr>
        <td class="tableholder">
            <table>
                <tbody>
                    <tr>
                        <td style="text-align:right">
                            <button type="button"
name="viewvitalsout" id="viewvitalsout" value="viewvitalsinnewwindow"
                            title="View all
vitals data entered in a new window" onclick="javascript:allVitalsWindow();">
                                View Vitals in New
                                Window
                            </button>
                        <button type="button"
name="viewvitals" id="viewvitals" value="viewvitals"
                            title="View all
vitals data entered" onclick="javascript:requestAllVitals();">
                                View Vitals
                            </button>
                        <button class="buttons"
type="reset" name="resetvitals" value="resetvitals"
                            title="Reset the MONR
subject vitals form">
                                Reset
                            </button>
                        <button class="buttons"
type="submit" name="submitvitals" value="submitvitals"
                            title="Submit the MONR
subject vitals form" onclick="javascript:submitForm('vitalsform');return false">
                                Submit
                            </button>
                    </td>
                </tr>
                </tbody>
            </table>
            <table>
                <tbody>
                    <tr>
                        <td id="message2"
class="message">&nbsp;  </td>

```



```

        </tr>
    </tbody>
</table>
</td>
</tr>
</table>
</form>
</td>
</tr>
<!-- All vitals show code start -->
<tr>
    <td id="vitalsdataholder" class="vitalsdataholder">
        <form id="form3" name="form3" action="." method="post"
autocomplete="off" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
            <table class="vitals_pagination_bar">
                <tr>
                    <td id="datasort" style="text-align:left"
width="30%">Order:
                        <select id="vitalsdataorder"
name="vitalsdataorder" onchange='javascript:vitalschangePage(this.id,1)''>
                            <option value="Descending"
selected="selected">Descending</option>
                            <option
value="Ascending">Ascending</option>
                        </select>
                        Data rows:
                        <select id="numberofdatapoints"
name="numberofdatapoints" onchange='javascript:vitalschangePage(this.id,1)''>
                            <option value="1000">1000</option>
                            <option value="750">750</option>
                            <option value="500">500</option>
                            <option value="250">250</option>
                            <option value="100"
selected="selected">100</option>
                            <option value="50">50</option>
                            <option value="10">10</option>
                        </select>
                    </td>
                    <td width="12%" style="text-align:left">
                        <span id="vitals_totalrows"></span>
                    </td>
                    <td width="12%" style="text-align:left">
                        <span id="vitals_totalpages"></span>
                    </td>
                    <td id="vitals_go_to_page">
                    </td>
                    <td id="vitalsOptions" class="vitalsoptions">
                    </td>
                </tr>
            </table>
            <div id="entireVitals" class="entireVitals">
                <div class="holdVitalsTable">
                    <div id="holdHeaders" class="holdHeaders">
                        <table><tr><td><div
class="vitalsDate">Date</div></td><td><div
class="vitalsTime">Time</div></td><td><div class="vitalsComments">Event<br
/>Comments</div></td><td><div class="vitalsEvent">Event</div></td><td><div

```

```

class="vitalsColWidth">HR</div></td><td><div class="vitalsColWidth">Art
BP</div></td><td><div class="vitalsColWidth">Mean<br />Art Pres</div></td><td><div
class="vitalsColWidth">PA Pres</div></td><td><div class="vitalsColWidth">Wedge<br
/>Pres</div></td><td><div class="vitalsColWidth">Urine<br
/>Output</div></td><td><div class="vitalsColWidth">Bladder<br
/>Pres</div></td><td><div class="vitalsColWidth">Temp</div></td><td><div
class="vitalsColWidth">Cardiac<br />Output</div></td><td><div
class="vitalsColWidth">Propofol</div></td><td><div
class="vitalsColWidth">BIS</div></td><td><div
class="vitalsColWidth">FiO2</div></td><td><div class="vitalsColWidth">StO2<br
/>(15mm)</div></td><td><div class="vitalsColWidth">THI<br
/>(15mm)</div></td><td><div class="vitalsColWidth">StO2<br />(Multi
Depth)</div></td><td><div class="vitalsColWidth">THI<br />(Multi
Depth)</div></td><td><div class="vitalsColWidth">Chest<br />Tube<br
/>Output</div></td><td><div class="vitalsColWidth">Art pH</div></td><td><div
class="vitalsColWidth">Art PCO2</div></td><td><div class="vitalsColWidth">Art
PO2</div></td><td><div class="vitalsColWidth">Art O2 sat</div></td><td><div
class="vitalsColWidth">Ven PO2</div></td><td><div class="vitalsColWidth">Ven O2
sat</div></td><td><div class="vitalsColWidth">Hb</div></td><td><div
class="vitalsColWidth">DO2</div></td><td><div
class="vitalsColWidth">VO2</div></td><td><div
class="vitalsColWidth">ERO2</div></td><td><div
class="vitalsColWidth">Lactate</div></td><td><div class="vitalsColWidth">Base<br
/>deficit</div></td><td><div class="vitalsSpecificGravity">Specific gravity<br
/>of Urine</div></td><td><div class="vitalsColWidth">Na+</div></td><td><div
class="vitalsColWidth">K+</div></td><td><div
class="vitalsColWidth">Ca++</div></td><td><div
class="vitalsColWidth">Glucose</div></td></tr></table>
</div>
<div id="innerDiv" class="innerDiv">
<h2>Please wait while we process your
request</h2>
</div>
</div>
</div>
</div>
</form>
</td>
</tr>
<!-- All vitals show code end. Labs form code start -->
<tr>
<td id="labsform" class="tableholder">
<form id="form4" name="form4" action="." method="post"
autocomplete="off" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
<fieldset>
<legend>Labs</legend>
<table summary="This table holds MONR labs form">
<tr>
<td class="tableholder">
<table>
<tbody>
<tr>
<td class="descriptor2" title="Event Name">
<label for="labsevent">
Event:
</label>
</td>
<td class="field2" title="Event Name">

```

```

class="selection">
    <select name="labsevent" id="labsevent"
        <option value="1">1 (Baseline)</option>
        <option value="2">2 (Shock45)</option>
        <option value="5">5 (FR2)</option>
        <option value="7">7 (FR4)</option>
        <option value="11">11 (FR8)</option>
        <option value="19">19 (FR16)</option>
        <option value="23">23 (FR20)</option>
        <option value="24">24 (PR24)</option>
        <option value="25">25 (PR48)</option>
    </select>
</td>
<td class="descriptor2" title="Total bilirubin">
    <label for="Bilirubin_total">
        Bilirubin-Total:
    </label>
</td>
<td class="field2" title="Total bilirubin">
    <input type="text" name="Bilirubin_total"
id="Bilirubin_total" maxlength="5" size="10" />
</td>
<td class="descriptor2" title="Albumin">
    <label for="Albumin">
        Albumin:
    </label>
</td>
<td class="field2" title="Albumin">
    <input type="text" name="Albumin" id="Albumin"
maxlength="4" size="10" />
</td>
<td class="descriptor2" title="Total Protein">
    <label for="Protein_total">
        Protein-Total:
    </label>
</td>
<td class="field2" title="Total Protein">
    <input type="text" name="Protein_total"
id="Protein_total" maxlength="4" size="10" />
</td>
</tr>
<tr>
<td class="descriptor2" title="Alkaline Phosphatase">
    <label for="Alk_phosphatase">
        ALK Phosphatase:
    </label>
</td>
<td class="field2" title="Alkaline Phosphatase">
    <input type="text" name="Alk_phosphatase"
id="Alk_phosphatase" maxlength="3" size="10" />
</td>
<td class="descriptor2" title="Alanine
aminotransferase ">
    <label for="ALT">
        ALT:
    </label>
</td>
<td class="field2" title="Alanine aminotransferase ">

```

```

        <input type="text" name="ALT" id="ALT"
maxLength="3" size="10" />
    </td>
    <td class="descriptor2" title="Aspartate
aminotransferase">
        <label for="AST">
            AST:
        </label>
    </td>
    <td class="field2" title="Aspartate aminotransferase">
        <input type="text" name="AST" id="AST"
maxLength="3" size="10" />
    </td>
    <td class="descriptor2" title="Urea Nitrogen">
        <label for="Urea_nitrogen">
            Urea Nitrogen:
        </label>
    </td>
    <td class="field2" title="Urea Nitrogen">
        <input type="text" name="Urea_nitrogen"
id="Urea_nitrogen" maxlength="2" size="10" />
    </td>
</tr>
<tr>
    <td class="descriptor2" title="Creatine kinase ">
        <label for="CK">
            CK:
        </label>
    </td>
    <td class="field2" title="Creatine kinase ">
        <input type="text" name="CK" id="CK" maxlength="5"
size="10" />
    </td>
    <td class="descriptor2" title="Creatinine">
        <label for="Creatinine">
            Creatinine:
        </label>
    </td>
    <td class="field2" title="Creatinine">
        <input type="text" name="Creatinine"
id="Creatinine" maxlength="5" size="10" />
    </td>
    <td class="descriptor2" title="Lactate dehydrogenase">
        <label for="LD">
            LD:
        </label>
    </td>
    <td class="field2" title="Lactate dehydrogenase">
        <input type="text" name="LD" id="LD" maxlength="5"
size="10" />
    </td>
    <td class="descriptor2" title="Platelet Count">
        <label for="Platelet_count">
            Platelet Count:
        </label>
    </td>
    <td class="field2" title="Platelet Count">

```

```

        <input type="text" name="Platelet_count"
id="Platelet_count" maxlength="4" size="10" />
    </td>
</tr>
<tr>
<td class="descriptor2" title="Urine Osmolality">
    <label for="Urine_osmolality">
        Urine Osmolality:
    </label>
</td>
<td class="field2" title="Urine Osmolality">
    <input type="text" name="Urine_osmolality"
id="Urine_osmolality" maxlength="6" size="10" />
</td>
<td class="descriptor2" title="Urine Creatinine">
    <label for="Urine_creatinine">
        Urine Creatinine:
    </label>
</td>
<td class="field2" title="Urine Creatinine">
    <input type="text" name="Urine_creatinine"
id="Urine_creatinine" maxlength="6" size="10" />
</td>
<td class="descriptor2">
    &nbsp;
</td>
<td class="field2">
    &nbsp;
</td>
<td class="descriptor2">
    &nbsp;
</td>
<td class="field2">
    &nbsp;
</td>
</tr>
</tbody>
</table>
</td>
</tr>
</table>
</fieldset>
<table summary="This table is used to hold user
addition form">
    <tr>
        <td class="tableholder">
            <table>
                <tbody>
                    <tr>
                        <td style="text-align:right">
                            <button type="button"
name="viewlabsout" id="viewlabsout" value="viewlabsinnewwindow"
                                title="View all
labs data entered in a new window" onclick="javascript:allLabsWindow();">
                                    View Labs in New
                                </button>
                            </td>
                    </tr>
                </tbody>
            </table>
        </td>
    </tr>
</table>

```

```

name="viewlabs" id="viewlabs" value="viewlabs"
                                <button type="button"
                                title="View all
                                View Labs
                                </button>
                                <button class="buttons"
                                title="Reset the MONR
                                Reset
                                </button>
                                <button class="buttons"
                                title="Submit the MONR
                                Submit
                                </button>
                                </td>
                                </tr>
                                </tbody>
                                </table>
                                <table>
                                <tbody>
                                <tr>
                                <td id="message3"
                                class="message">&nbsp;</td>
                                </tr>
                                </tbody>
                                </table>
                                </td>
                                </tr>
                                </table>
                                </form>
                                </td>
                                </tr>
                                <!-- All labs show code start -->
                                <tr>
                                <td id="labsdataholder" class="labsdataholder">
                                <form id="form5" name="form5" action="." method="post"
                                autocomplete="off" enctype="application/x-www-form-urlencoded"
                                onsubmit="javascript:return false">
                                <div id="entireLabs" class="entireLabs">
                                <div class="holdLabsTable">
                                <div id="holdLabsHeaders"
                                class="holdLabsHeaders">
                                <table><tr><td><div
                                class="labsColWidth">Event</div></td><td><div class="labsColWidth">Total<br
                                />Bilirubin</div></td><td><div class="labsColWidth">Albumin</div></td><td><div
                                class="labsColWidth">Total<br />Protein</div></td><td><div
                                class="labsColWidth">ALK<br />Phosphatase</div></td><td><div
                                class="labsColWidth">ALT</div></td><td><div
                                class="labsColWidth">AST</div></td><td><div class="labsColWidth">Urea<br
                                />Nitrogen</div></td><td><div class="labsColWidth">CK</div></td><td><div
                                class="labsColWidth">Creatinine</div></td><td><div
                                class="labsColWidth">LD</div></td><td><div class="labsColWidth">Platelet<br
                                />Count</div></td><td><div class="labsColWidth">Urine<br

```

```

/>Osmolality</div></td><td><div class="labsColWidth">Urine<br
/>Creatinine</div></td></tr></table>
        </div>
        <div id="labsinnerDiv"
class="labsinnerDiv">
        <h2>Please wait while we process your
request</h2>
        </div>
    </div>
</div>
</form>
</td>
</tr>
<!-- All labs show code end. GCS scores form code start -->
<tr>
    <td id="gcscscores" class="tableholder">
        <form id="form6" name="form6" action="." method="post"
autocomplete="off" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
            <fieldset>
            <legend>GCS</legend>
            <table summary="This table holds MONR GCS scores
form">
                <tr>
                <td class="tableholder">
                <table>
                <tbody>
                <tr>
                <td class="descriptor2" title="GCS score at Baseline">
                    <label for="gcs1">
                        GCS at 1 (Baseline):
                    </label>
                </td>
                <td class="field2" title="GCS score at event 1">
                    <input type="text" name="gcs1" id="gcs1"
maxlength="2" size="10" />
                </td>
                <td class="descriptor2" title="GCS score at 21 hours
after shock">
                    <label for="gcs23">
                        GCS at 23(FR20):
                    </label>
                </td>
                <td class="field2" title="GCS score at event 23">
                    <input type="text" name="gcs23" id="gcs23"
maxlength="2" size="10" />
                </td>
                <td class="descriptor2" title="GCS score at 45 hours
after shock">
                    <label for="gcs24">
                        GCS at 24(PR24):
                    </label>
                </td>
                <td class="field2" title="GCS score at event 24">
                    <input type="text" name="gcs24" id="gcs24"
maxlength="2" size="10" />
                </td>
            </tbody>
            </table>
            </form>
        </td>
    </tr>

```

```

after shock">
    <td class="descriptor2" title="GCS score at 69hours
        <label for="gcs25">
            GCS at 25(PR48):
        </label>
    </td>
    <td class="field2" title="GCS score at event 25">
        <input type="text" name="gcs25" id="gcs25"
maxlength="2" size="10" />
    </td>
</tr>
</tbody>
</table>
</td>
</tr>
</table>
</fieldset>
<table summary="This table is used to hold user
addition form">
    <tr>
        <td class="tableholder">
            <table>
                <tbody>
                    <tr>
                        <td style="text-align:right">
                            <button type="button"
name="viewgcsscores" id="viewgcsscores" value="viewgcsscores"
                                title="View all
GCS scores entered" onclick="javascript:requestAllGcscscores();">
                                    View GCS
                                </button>
                            <button class="buttons"
type="reset" name="resetgcscscores" value="resetgcscscores"
                                title="Reset the GCS
scores form">
                                    Reset
                                </button>
                            <button class="buttons"
type="submit" name="submitgcscscores" value="submitgcscscores"
                                title="Submit the GCS
scores form" onclick="javascript:submitForm('gcscscoresform');return false">
                                    Submit
                                </button>
                        </td>
                    </tr>
                </tbody>
            </table>
        </td>
        <td id="message4"
class="message">&nbsp;</td>
    </tr>
</tbody>
</table>
</td>
</tr>
</table>

```



```

        </form>
    </td>
</tr>
<!-- Show GCS score code start -->
<tr>
    <td id="gcsdataholder" class="gcsdataholder">
        <form id="form7" name="form7" action="." method="post"
autocomplete="off" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
            <div id="entireGcscsscores" class="entireGcscsscores">
                <div class="holdGcscsscoresTable">
                    <div id="holdGcscsscoresHeaders"
class="holdGcscsscoresHeaders">
                        <table><tr><td><div
class="gcscsscoresColWidth">At 1(Baseline)</div></td><td><div
class="gcscsscoresColWidth">At 23(FR20)</div></td><td><div
class="gcscsscoresColWidth">At 24(PR24)</div></td><td><div
class="gcscsscoresColWidth">At 25(PR48)</div></td></tr></table>
                    </div>
                    <div id="gcscsscoresinnerDiv"
class="gcscsscoresinnerDiv">
                        <h2>Please wait while we process your
request</h2>
                    </div>
                </div>
            </div>
        </form>
    </td>
</tr>
</tbody>
</table>
</div>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for dataquery.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Data Query</title>
    <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
    <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
</head>

```

```

<body>
#Common_Data#
  <tr>
    <td class="formheading">
      Data - MONR Data Query Form</td>
  </tr>
  <tr>
    <td>#greetings#</td>
  </tr>
  <tr>
    <td>Page under construction</td>
  </tr>
</tbody>
</table>
</body>
</html>

```

Code written for dataview.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Data View</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico"/>
  <link rel="shortcut icon" type="image/x-icon" href="/images/favicon.ico"/>
  <link rel="stylesheet" type="text/css" href="/stylesheets/main.css"
media="screen" />
  <link rel="stylesheet" type="text/css" href="/stylesheets/dataview.css"
media="screen" />
</head>
<body onload="javascript:opener.requestAllVitalsNewWindow();">
  <table id="entirepage">
    <tbody>
      <tr>
        <td class="tableholder">
          <table id="banner" summary="This table defines the banner">
            <tbody>
              <tr>
                <td id="ubanner">
                  &nbsp;
                </td>
                <td class="smallcaption">
                  SCCL<br /><span style="font-size:14px;font-
weight:bold">Surgical Critical Care Laboratory</span>
                </td>
              </tr>
            </tbody>
          </table>
        </td>
      </tr>
    </tbody>
  </table>
</tr>
</tr>

```

```

        <td class="formheading" style="text-align:left">Forms - MONR
Vitals Data View</td>
    </tr>
    <tr>
        <td id="vitalsdataholder" class="vitalsdataholder">
            <form id="form3" name="form3" action="." method="post"
enctype="application/x-www-form-urlencoded" onsubmit="javascript:return false">
                <table class="vitals_pagination_bar">
                    <tr>
                        <td id="datasort" width="10%">Order:
                            <select id="vitalsdataorder"
onchange='javascript:opener.vitalschangePage(this.id,1)'+
                                <option
value="Ascending" selected="selected">Ascending</option>
                                <option
value="Descending">Descending</option>
                            </select>
                            Data rows:
                            <select id="numberofdatapoints"
onchange='javascript:opener.vitalschangePage(this.id,1)'+
                                <option value="1000">1000</option>
                                <option value="750">750</option>
                                <option value="500">500</option>
                                <option value="250">250</option>
                                <option value="100"
selected="selected">100</option>
                                <option value="50">50</option>
                                <option value="10">10</option>
                            </select>
                        </td>
                        <td width="4%">
                            <span id="vitals_totalrows"></span>
                        </td>
                        <td width="4%">
                            <span id="vitals_totalpages"></span>
                        </td>
                        <td id="vitals_go_to_page" width="5%">
                        </td>
                        <td id="vitalsOptions" class="vitalsoptions">
                        </td>
                    </tr>
                </table>
                <div id="entireVitals" class="entireVitals">
                    <div class="holdVitalsTable">
                        <div id="holdHeaders" class="holdHeaders">
                            <table><tr><td><div
class="vitalsDate">Date</div></td><td><div
class="vitalsTime">Time</div></td><td><div class="vitalsComments">Event<br
/>Comments</div></td><td><div class="vitalsEvent">Event</div></td><td><div
class="vitalsColWidth">HR</div></td><td><div class="vitalsColWidth">Art
BP</div></td><td><div class="vitalsColWidth">Mean<br />Art Pres</div></td><td><div
class="vitalsColWidth">PA Pres</div></td><td><div class="vitalsColWidth">Wedge<br
/>Pres</div></td><td><div class="vitalsColWidth">Urine<br
/>Output</div></td><td><div class="vitalsColWidth">Bladder<br
/>Pres</div></td><td><div class="vitalsColWidth">Temp</div></td><td><div
class="vitalsColWidth">Cardiac<br />Output</div></td><td><div
class="vitalsColWidth">Propofol</div></td><td><div
class="vitalsColWidth">BIS</div></td><td><div

```

```

class="vitalsColWidth">FiO2</div></td><td><div class="vitalsColWidth">StO2<br
/>(15mm)</div></td><td><div class="vitalsColWidth">THI<br
/>(15mm)</div></td><td><div class="vitalsColWidth">StO2<br />(Multi
Depth)</div></td><td><div class="vitalsColWidth">THI<br />(Multi
Depth)</div></td><td><div class="vitalsColWidth">Chest<br />Tube<br
/>Output</div></td><td><div class="vitalsColWidth">Art pH</div></td><td><div
class="vitalsColWidth">Art PCO2</div></td><td><div class="vitalsColWidth">Art
PO2</div></td><td><div class="vitalsColWidth">Art O2 sat</div></td><td><div
class="vitalsColWidth">Ven PO2</div></td><td><div class="vitalsColWidth">Ven O2
sat</div></td><td><div class="vitalsColWidth">Hb</div></td><td><div
class="vitalsColWidth">D02</div></td><td><div
class="vitalsColWidth">V02</div></td><td><div
class="vitalsColWidth">ER02</div></td><td><div
class="vitalsColWidth">Lactate</div></td><td><div class="vitalsColWidth">Base<br
/>deficit</div></td><td><div class="vitalsSpecificGravity">Specific<br
/>gravity<br />of Urine</div></td><td><div
class="vitalsColWidth">Na+</div></td><td><div
class="vitalsColWidth">K+</div></td><td><div
class="vitalsColWidth">Ca++</div></td><td><div
class="vitalsColWidth">Glucose</div></td></tr></table>
</div>
<div id="innerDiv" class="innerDiv">
<h2>Please wait while we process your
request</h2>
</div>
</div>
</div>
<div id="message1" class="message"
style="height:2px">&nbsp;</div>
<div id="message2" class="message" style="text-
align:left">&nbsp;</div>
<div id="message3" class="message"
style="height:0px">&nbsp;</div>
</form>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for fileupload.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>File Upload</title>
<link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
<link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
<link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />

```

```

    <link rel="stylesheet" type="text/css" href="stylesheets/fileupload.css"
media="screen" />
    <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
    <script language="javascript" type="text/javascript"
src="js/fileupload.js"></script>
</head>
<body onload="javascript:loadPigIds();">
#Common_Data#
    <tr>
        <td class="formheading">Files - MONR File Upload</td>
    </tr>
    <tr>
        <td>
            <div class="aligncenter">
                <table>
                    <tbody>
                        <tr>
                            <td>#greetings#</td>
                        </tr>
                        <tr>
                            <td class="tableholder">
                                <form id="uploadfileform"
name="uploadfileform" action="." method="post" enctype="multipart/form-data"
onsubmit="javascript:if(!validateForm()){return false;};">
                                    <fieldset>
                                        <legend class="mainform">File
Upload</legend>
                                        <table summary="This table holds
file upload">
                                            <tr>
                                                <td class="tableholder">
                                                    <table>
                                                        <tbody>
                                                            <tr>
                                                                <td
class="descriptor" width="50%">
                                                                    Select
                                                                    Pig ID:
                                                                    </td>
                                                                <td
id="holdpigids" class="field" width="50%">
                                                                    <!--
                                                                    <select name="pigids" id="pigids"
onchange="javascript:selectFileType(this.value);">
                                                                    Bug in
                                                                    IE prevents using inner html property of select element. So, instead inner HTML of
                                                                    td is used to load values
                                                                    </select> -->
                                                                </td>
                                                            </tr>
                                                        </tbody>
                                                    </table>
                                                </td>
                                            </tr>
                                        </table>
                                    </td>
                                </tr>
                            <td class="tableholder">

```

```

id="selectfiletype" class="allrows">

class="descriptor" width="50%">
select the file type you are uploading

class="field" id="dynamicoptions" width="50%">

```

```

<table
  <tbody>
    <tr>
      <td
        Please
      </td>
      <td
        &nbsp;
      </td>
    </tr>
  </tbody>
</table>
<table

```

```

id="selecttissuetype" class="allrows">

class="descriptor" width="50%">
select the sample origin

class="field" id="dynamicoptions1" width="50%">

```

```

  <tbody>
    <tr>
      <td
        Please
      </td>
      <td
        &nbsp;
      </td>
    </tr>
  </tbody>
</table>
<table id="enterrawwt"

```

```

class="allrows">

class="descriptor" id="descriptor3" title="Time of the event" width="50%">
for="tissuewt">
Enter tissue weight in milli grams:
</label>

```

```

  <tbody>
    <tr>
      <td
        <input
      </td>
      <td
        <label
      </td>
    </tr>
  </tbody>
</table>
<table id="enterlyowt"

```

```

class="allrows">

```

```

  <tbody>
    <tr>

```

```

class="descriptor" id="descriptor4" title="Time of the event" width="50%">
</td>
<label
for="lyowt">
Enter lyophilized tissue weight in milli grams:
</label>
</td>
<td
class="field" id="field4" title="Lyophilized tissue weight in mg" width="50%">
<input
type="text" name="lyowt" id="lyowt" maxlength="9" size="9" />
</td>
</tr>
</tbody>
</table>
<table
id="enternormalization" class="allrows">
<tbody>
<tr>
<td
class="descriptor" id="descriptor5" title="Time of the event" width="50%">
<label
for="normfact">
Enter normalization constant:
</label>
</td>
<td
class="field" id="field5" title="Normalization constant" width="50%">
<input
type="text" name="normfact" id="normfact" maxlength="9" size="9" />
</td>
</tr>
</tbody>
</table>
<table
id="selectevent" class="allrows">
<tbody>
<tr>
<td
class="descriptor" width="50%">
Please
select the time point
</td>
<td
class="field" id="dynamicoptions2" width="50%">
<input type="text" />
</td>
</tr>
</tbody>
</table>
<table id="entertime"
class="allrows">
<tbody>
<tr>

```

```

class="descriptor" id="descriptor2" title="Time of the event" width="50%">
for="hours">
Enter exact time of collection:<br />(HH MM)
</label>
class="field" id="field2" title="Time of the event" width="50%">
type="text" name="hours" id="hours" maxlength="2" size="2" />
type="text" name="minutes" id="minutes" maxlength="2" size="2" />
id="allowupload" class="allrows">
type="file" id="fileupload" name="fileupload" class="fileupload" size="50"/>
type="submit" value="Upload" />
type="hidden" name="fileuploadpage" id="fileuploadpage" value="fileupload" />
type="hidden" name="studyname" id="studyname" value="MONR" />
id="message1"></td>

```



```

                </tr>
                <tr>
                    <td class="tableholder" id="filesholder"></td>
                </tr>
            </tbody>
        </table>
    </div>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for home.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Home</title>
    <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
    <link rel="stylesheet" type="text/css" href="stylesheets/emaillist.css"
media="screen" />
    <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
    <script language="javascript" type="text/javascript"
src="js/emaillist.js"></script>
</head>
<body><!-- onload="loadEmailList();" -->
#Common_Data#
    <tr>
        <td class="formheading">SCCL - Home Page</td>
    </tr>
    <tr>
        <td>
            <div class="aligncenter">
                <table>
                    <tbody>
                        <tr>
                            <td>#greetings#</td>
                        </tr>
                        <tr>
                            <td class="tableholder">
                                Please choose the appropriate menu link.
                                To logout, click on Log Out link at the right top corner or just close the
                                browser.
                            </td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </td>
    </tr>

```

```

        <td id="message1" class="message">&nbsp;</td>
    </tr>
    <tr>
        <td id="message2" class="message">&nbsp;</td>
    </tr>
    <!--
    <tr>
        <td class="tableholder" id="emaillist">Please
wait.....</td>
    </tr>
    -->
</tbody>
</table>
</div>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for index.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Home</title>
    <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
    <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
    <script language="javascript" type="text/javascript"
src="js/login.js"></script>
</head>
<body onload="document.loginform.user_id.focus()">
    <table id="entirepage">
        <tbody>
            <tr>
                <td class="tableholder">
                    <table>
                        <tbody>
                            <tr>
                                <td class="testwebsite">&nbsp;</td>
                                <td>&nbsp;</td>
                            </tr>
                        </tbody>
                    </table>
                </td>
            </tr>
        </tbody>
    </table>

```

```

<tr>
  <td class="tableholder">
    <table id="banner" summary="This table defines the banner">
      <tbody>
        <tr>
          <td id="ubanner">
            &nbsp;
          </td>
          <td class="smallcaption">
            SCCL<br /><span style="font-size:14px;font-
weight:bold;float:left;clear:left">Surgical Critical Care Laboratory</span>
          </td>
        </tr>
      </tbody>
    </table>
  </td>
</tr>
<tr>
<tr>
  <td class="formheading">SCCL - Login Page</td>
</tr>
<tr>
<tr>
  <td>
    <div class="aligncenter">
      <table>
        <tbody>
          <tr>
            <td>#greetings#</td>
          </tr>
          <tr>
            <td class="tableholder">
              <!-- <div class="aligncenter"
style="width:650px;"> -->
                <form id="loginform" name="loginform"
action="." method="post" autocomplete="off" enctype="application/x-www-form-
urlencoded" onsubmit="javascript:if(!validateForm()){return false;};">
                  <fieldset class="loginform">
                    <legend
class="mainform">Login</legend>
                    <table summary="User login form">
                      <tr>
                        <td style="text-
align:right;padding-right:10px;padding-top:20px" width="45%">
                          <label for="user_id">
                            User ID:
                          </label>
                        </td>
                        <td style="text-
align:left;padding-top:20px">
                          <input type="text"
name="user_id" id="user_id" maxlength="30" size="20" tabindex="1" title="Enter
User ID"/>
                        </td>
                      </tr>
                      <tr>
                        <td style="text-
align:right;padding-right:10px;padding-top:20px" width="45%">
                          <label for="password">
                            Password:

```

```

        </label>
    </td>
    <td style="text-align: left; padding-top: 20px">
        <input type="password"
        name="password" id="password" maxlength="30" size="20" tabindex="2" title="Enter
        Password"/>
    </td>
</tr>
<tr>
    <td style="text-align: left; padding-left: 46px; padding-top: 10px">
        <button class="button"
        type="submit" name="login" id="login" value="login"
        title="Click here after entering User ID and Password">
            Submit
        </button>
    </td>
    <td colspan="2" style="padding-bottom: 10px">
        Forgot password? <span
        style="text-decoration: underline; cursor: pointer"
        onclick="javascript:openPage('retrievepassword')">Click Here</span> to request a
        temporary password
    </td>
</tr>
</table>
</fieldset>
</form>
<!-- </div> -->
</td>
</tr>
<tr>
    <td class="tableholder">
        <table>
            <tbody>
                <tr>
                    <td>
                        <div id="message1"
                        class="message">&nbsp;</div>
                    </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
</tbody>
</table>
</div>
</td>
</tr>
</tbody>
</table>
</body>

```



```

class="labsColWidth">ALK<br />Phosphatase</div></td><td><div
class="labsColWidth">ALT</div></td><td><div
class="labsColWidth">AST</div></td><td><div class="labsColWidth">Urea<br
/>Nitrogen</div></td><td><div class="labsColWidth">CK</div></td><td><div
class="labsColWidth">Creatinine</div></td><td><div
class="labsColWidth">LD</div></td><td><div class="labsColWidth">Platelet<br
/>Count</div></td><td><div class="labsColWidth">Urine<br
/>Osmolality</div></td><td><div class="labsColWidth">Urine<br
/>Creatinine</div></td></tr></table>
</div>
<div id="labsinnerDiv" class="labsinnerDiv">
<h2>Please wait while we process your
request</h2>
</div>
</div>
</div>
</div>
</form>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for manageusers.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>Manage Users - Add user</title>
<link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
<link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
<link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
<link rel="stylesheet" type="text/css" href="stylesheets/manageusers.css"
media="screen" />
<script language="javascript" type="text/javascript"
src="js/menu.js"></script>
<script language="javascript" type="text/javascript"
src="js/admin.js"></script>
</head>
<body onload="document.adduserform.user_id.focus();loadUserData();">
#Common_Data#
<tr>
<td class="formheading">Admin - Manage Users</td>
</tr>
<tr>
<td>
<div class="aligncenter">
<table>
<tbody>

```

```

        <tr>
            <td>#greetings#</td>
        </tr>
        <tr>
            <td class="tableholder">
                <form id="adduserform" name="adduserform"
action="." method="post" autocomplete="off" enctype="application/x-www-form-
urlencoded" onsubmit="javascript:return false">
                    <fieldset>
                        <legend class="mainform">Add
Users</legend>
                        <table summary="This table is used to
hold user addition form">
                            <tr>
                                <td class="tableholder">
                                    <table>
                                        <tbody>
                                            <tr>
                                                <td
class="descriptor1">
                                                    <label
for="user_id">
<sup>1</sup>User ID*:
                                                    </label>
                                                </td>
                                                <td
class="field1" title="Enter User ID">
                                                    <input
type="text" name="user_id" id="user_id" maxlength="30" size="20" tabindex="1" />
                                                    </td>
                                                <td
class="descriptor1">
                                                    <label
for="password1">
<sup>2</sup>Password*:
                                                    </label>
                                                </td>
                                                <td
class="field1" title="Enter Password">
                                                    <input
type="password" name="password1" id="password1" maxlength="30" size="20"
tabindex="2" />
                                                    </td>
                                                <td
class="descriptor1">
                                                    <label
for="password2">
Re-
enter Password*:
                                                    </label>
                                                </td>
                                                <td
class="field1" title="Re-enter Password">
                                                    <input
type="password" name="password2" id="password2" maxlength="30" size="20"
tabindex="3" />

```

```

        </td>
      </tr>
      <tr>
        <td>
          <label
            class="descriptor1"
            for="firstname">
            <sup>3</sup>First Name*:
          </label>
        </td>
        <td>
          <input
            class="field1" title="Enter First Name of the User"
            type="text" name="firstname" id="firstname" maxlength="30" size="20" tabindex="4"
            />
        </td>
      </tr>
      <tr>
        <td>
          <label
            class="descriptor1"
            for="lastname">
            <sup>3</sup>Last Name:
          </label>
        </td>
        <td>
          <input
            class="field1" title="Enter Last Name of the User"
            type="text" name="lastname" id="lastname" maxlength="30" size="20" tabindex="5" />
        </td>
      </tr>
      <tr>
        <td>
          <label
            class="descriptor1"
            for="middleinitial">
            <sup>4</sup>Middle Initial:
          </label>
        </td>
        <td>
          <input
            class="field1" title="Enter Middle Initial of the User"
            type="text" name="middleinitial" id="middleinitial" maxlength="2" size="20"
            tabindex="6" />
        </td>
      </tr>
      <tr>
        <td>
          <label
            class="descriptor1"
            for="email1">
            address*:
          </label>
        </td>
        <td>
          Email
          <input
            class="field1" title="Enter user email address"
            type="text" name="email1" id="email1" maxlength="40" size="20" tabindex="7" />
        </td>
      </tr>

```



```

</td>
<td
class="descriptor1">
for="email2">
Enter Email address*:
class="field1" title="Re-Enter user email address">
type="text" name="email2" id="email2" maxlength="40" size="20" tabindex="8" />
class="descriptor1">
for="groupname">
assigned*:
class="field1" title="Select the projects assigned. Press ctrl key to select
multiple studies">
name="groupname" id="groupname" style="width:100%" tabindex="9">
<option selected="selected" value="Not_selected">Select</option>
<option value="Lab_user">Lab User</option>
<option value="Guest">Guest</option>
<option value="Admin">Administrator</option>
<option value="Special">Special Access</option>
class="descriptor1">
for="testonly">
<sup>5</sup>Test-only access:
class="field1" style="text-align:center">
type="checkbox" name="testonly" id="testonly" tabindex="10" />
class="descriptor1">
for="allprojects">

```

Projects assigned*:

```

                                                                 </label>
                                                                 </td>
                                                                 <td>
class="field1" title="" style="text-align:right;font-size:12px;padding-
right:68px">
                                                                 <label
for="allprojects">All </label><input type="checkbox" name="allprojects"
id="allprojects" onclick="javascript:selectAll('project');" tabindex="11" /><br />
                                                                 <label
for="project_MONR">MONR </label><input type="checkbox" name="project_MONR"
id="project_MONR" onclick="javascript:selectOne('project',this);" tabindex="12"
/><br />
                                                                 <label
for="project_PONR">PONR </label><input type="checkbox" name="project_PONR"
id="project_PONR" onclick="javascript:selectOne('project',this);" tabindex="13"
/><br />
                                                                 <label
for="project_DARPA">DARPA </label><input type="checkbox" name="project_DARPA"
id="project_DARPA" onclick="javascript:selectOne('project',this);" tabindex="14"
/>
                                                                 </td>
                                                                 <td>
class="descriptor1">
                                                                 </td>
                                                                 <td>
class="field1" style="text-align:center">
                                                                 </td>
                                                                 </tr>
                                                                 </tbody>
                                                                 </table>
                                                                 </td>
                                                                 </tr>
                                                                 </table>
</fieldset>
<table summary="This table is used to hold
user addition form">
  <tr>
    <td class="tableholder">
      <table>
        <tbody>
          <tr>
            <td style="text-
align:left">
              fields
            </td>
            <td style="text-
align:right">
              *Required
            </td>
          <tr>
            <td colspan="2" style="text-align:center">
              <button
class="buttons" type="reset">
                Reset</button>
              <button
class="buttons" type="submit" name="adduser" id="Button1" value="adduser">
                Add User</button>
            </td>
          </tr>
        </tbody>
      </table>
    </td>
  </tr>
</table>

```

```

                                                    tabindex="15"
title="Submit the add user form"
onclick="javascript:Admin_submitForm('adduser','');return false">
                                                    Submit</button>
                                                    </td>
                                                    </tr>
                                                    </tbody>
                                                    </table>
                                                    </td>
                                                    </tr>
                                                    </table>
                                                    </form>
                                                    </td>
</tr>
<tr>
    <td class="tableholder">
        <table>
            <tbody>
                <tr>
                    <td id="message1" class="message">
                        &nbsp;
                    </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
<tr>
    <td class="tableholder" id="addedusers">
        &nbsp;
    </td>
</tr>
</tbody>
</table>
</div>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for myqueries.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>My saved queries</title>
    <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />

```

```

    <link rel="stylesheet" type="text/css" href="stylesheets/queries.css"
media="screen" />
    <link rel="stylesheet" type="text/css" href="stylesheets/queriesprint.css"
media="print" />
    <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
    <script language="javascript" type="text/javascript"
src="js/jqueryfloatbox.js"></script>
    <script language="javascript" type="text/javascript"
src="js/queryoutput.js"></script>
</head>
<body onload="javascript:loadMyQueries()">
    <div class="entirepagecontainer">
        #Common_Data#
        <tr>
            <td class="formheading">
                Data - MONR "My Queries"</td>
            </tr>
            <tr>
                <td>
                    <div class="aligncenter">
                        <table>
                            <tbody>
                                <tr>
                                    <td>#greetings#</td>
                                </tr>
                                <tr>
                                    <td id="message1"
class="message">&nbsp;</td>
                                </tr>
                            </tbody>
                        </table>
                    </div>
                </td>
            </tr>
        </tbody>
    </table>
    <div class="datatable" id="holdalldata">
        <div id="holdquerytable">Please wait....</div>
        <!--
        <form id="editqueryform" name="editqueryform" autocomplete="off"
action="." method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
            <div id="queryeditform"></div>
        </form>
        -->
        <div id="footer">
            <div class="pagebreakafter"></div>
            <hr/>
            <p class="printonly">
                Surgical Critical Care laboratory, University of Minnesota
            </p>
            <p class="printonly">
                If you are not authorized to view this data, please destroy
the document immediately
            </p>
        </div>
    </div>

```

```

    </div>
</body>
</html>

```

Code written for queryresultdisplay.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Query results display and export</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico"/>
  <link rel="shortcut icon" type="image/x-icon" href="/images/favicon.ico"/>
  <link rel="stylesheet" type="text/css" href="/stylesheets/main.css"
media="screen" />
  <link rel="stylesheet" type="text/css" href="/stylesheets/queryresultview.css"
media="screen" />
  <link rel="stylesheet" type="text/css"
href="/stylesheets/queryresultviewprint.css" media="print" />
</head>
<body
onload="javascript:opener.displayQueryResults(document.queryresultdisplayform.numb
erofdatapoints.value);">
  <table id="entirepage">
    <tbody>
      <tr>
        <td class="tableholder">
          <table id="banner" summary="This table defines the banner">
            <tbody>
              <tr>
                <td id="ubanner">
                  &nbsp;
                </td>
                <td class="smallcaption">
                  SCCL<br /><span style="font-size:14px;font-
weight:bold">Surgical Critical Care Laboratory</span>
                </td>
              </tr>
            </tbody>
          </table>
        </td>
      </tr>
      <tr>
        <td class="formheading" style="text-align:left">Data - MONR "Query
results view"</td>
      </tr>
      <tr>
        <td id="queryresultdataholder" class="queryresultdataholder">
          <div id="message1" class="message">Please wait.....</div>
          <div id="holdbuttons" class="noprint">
            <div style="float:left;clear:left;padding-
bottom:10px;padding-top:10px">

```

```

        <button class="buttons" title="Switch to data
selection page to change your selection" type="button" name="showparent"
id="showparent" value="showparent" onclick="javascript:opener.focus()">Switch to
main page</button>
        <button class="buttons" title="Close this window
and go back to data selection page" type="button" name="closethis" id="closethis"
value="closethis"
onclick="javascript:opener.queryResultPopup.close();opener.focus();">Close this
window</button>
    </div>
    <div class="downloadicons" style="float:left;padding-
left:5px;padding-bottom:10px;padding-top:10px" id="printtable" title="Print
visible data" onclick="javascript>window.print();return false;">
        &nbsp;
    </div>
    <div id="holddownloadicons" style="float:left">
        &nbsp;
    </div>
</div>
<form id="queryresultdisplayform"
name="queryresultdisplayform" autocomplete="off" action="." method="post"
enctype="application/x-www-form-urlencoded" onsubmit="javascript:return false">
    <div id="pagination" class="noprint">
        <table class="queryresult_pagination_bar"
style="clear:left">
            <tr>
                <td id="datasort">
                    <select id="numberofdatapoints"
name="numberofdatapoints"
onchange='javascript:opener.queryresultchangePage(this.id,1)'\>
                        <option value="1000">1000</option>
                        <option value="750">750</option>
                        <option value="500">500</option>
                        <option value="250">250</option>
                        <option value="100">100</option>
                        <option value="50">50</option>
                        <option value="10">10</option>
                    </select>
                    rows/page
                </td>
                <td>
                    <span
id="queryresult_totalrows"></span>
                </td>
                <td>
                    <span
id="queryresult_totalpages"></span>
                </td>
                <td id="queryresult_go_to_page">
                </td>
                <td id="queryresultOptions"
class="queryresultoptions">
                </td>
            </tr>
        </table>
    </div>
</form>

```

```

                <center><div id="datatable" class="datatable"><span
style="font-size:30px">Please wait...</span></div></center>
                <div id="message3" class="message"
style="height:0px">&nbsp;</div>

                </td>
            </tr>
        </tbody>
    </table>
    <div class="pagebreakafter"></div>
    <hr/>
    <p class="printonly">
        Surgical Critical Care laboratory, University of Minnesota
    </p>
    <p class="printonly">
        If you are not authorized to view this data, please destroy the document
immediately
    </p>
</body>
</html>

```

Code written for sharedqueries.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Queries shared by others</title>
    <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
    <link rel="stylesheet" type="text/css" href="stylesheets/queries.css"
media="screen" />
    <link rel="stylesheet" type="text/css" href="stylesheets/queriesprint.css"
media="print" />
    <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
    <script language="javascript" type="text/javascript"
src="js/jqueryfloatbox.js"></script>
    <script language="javascript" type="text/javascript"
src="js/queryoutput.js"></script>
</head>
<body onload="javascript:loadSharedQueries()">
    <div class="entirepagecontainer">
        #Common_Data#
        <tr>
            <td class="formheading">
                Data - MONR "Shared Queries"</td>
        </tr>
        <tr>
            <td>
                <div class="aligncenter">

```

```

        <table>
          <tbody>
            <tr>
              <td>#greetings#</td>
            </tr>
            <tr>
              <td id="message1"
class="message">&nbsp;</td>
            </tr>
          </tbody>
        </table>
      </div>
    </td>
  </tr>
</tbody>
</table>
<div class="datatable" id="holdalldata">
  <div id="holdquerytable">Please wait....</div>
  <div class="pagebreakafter"></div>
  <hr/>
  <p class="printonly">
    Surgical Critical Care laboratory, University of Minnesota
  </p>
  <p class="printonly">
    If you are not authorized to view this data, please destroy the
document immediately
  </p>
</div>
</div>
</body>
</html>

```

Code written for simpleexport.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Data Query</title>
  <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
  <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
  <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
  <link rel="stylesheet" type="text/css" href="stylesheets/simpleexport.css"
media="screen" />
  <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
  <script language="javascript" type="text/javascript"
src="js/simpleexport.js"></script>
</head>
<body onload="javascript:loadDataType();">
#Common_Data#
  <tr>

```



```

        <td class="formheading">
Data - MONR Simple Export</td>
</tr>
<tr>
<td>
<div class="aligncenter">
<table>
<tbody>
<tr>
<td>#greetings#</td>
</tr>
<!--
<tr>
<td class="tableholder">
<form id="excelexportform"
name="excelexportform" action="." method="post" enctype="multipart/form-data">
<fieldset>
<legend class="mainform">Excel
Export</legend>
<table summary="This table holds
file upload">
<tr>
<td class="tableholder">
<table>
<tbody>
<tr>
<td
class="descriptor1" width="50%">
Pig ID:
id="holdpigids" class="field1" width="50%"> -->
<select name="pigids" id="pigids"
onchange="javascript:selectFileType(this.value);">
IE prevents using inner html property of select element. So, instead inner HTML of
td is used to load values
</select> -->
<!-- </td>
</tr>
</tbody>
</table>
</td>
</tr>
</table>
</fieldset>
</form>
</td>
</tr>
-->
<tr>
<td id="message1" class="message">&nbsp;</td>
</tr>
<!--
<tr>

```

```

        <td class="tableholder"
id="individualexcelfileholder"></td>
        </tr>
        -->
        <tr>
            <td style="text-align:center"
id="excelbutton">
                <button type="button"
id="update_excel_files" name="update_excel_files" title="Update all excel files"
onclick="javascript:updateExcelFiles();">
                    Update Excel Files
                </button>
            </td>
        </tr>
        <tr>
            <td class="tableholder">
                <table id="selectdatatype"
class="allrows">
                    <tbody>
                        <tr>
                            <td class="descriptor"
width="50%">
                                Please select data
                            </td>
                            <td class="field"
id="dynamicoptions" width="50%">
                                &nbsp;
                            </td>
                        </tr>
                    </tbody>
                </table>
                <table id="more_options" class="allrows">
                    <tbody>
                        <tr>
                            <td class="descriptor"
width="50%">
                                Please select one
                            </td>
                            <td class="field"
id="dynamicoptions1" width="50%">
                                &nbsp;
                            </td>
                        </tr>
                    </tbody>
                </table>
            </td>
        </tr>
        <tr>
            <td class="tableholder"
id="allexcelfileholder"></td>
        </tr>
    </tbody>
</table>
</div>
</td>
</tr>
</tbody>
</table>

```

```
</body>
</html>
```

Code written for survey.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Survey</title>
  <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
  <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
  <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
  <link rel="stylesheet" type="text/css" href="stylesheets/survey.css"
media="screen" />
  <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
  <script language="javascript" type="text/javascript"
src="js/survey.js"></script>
</head>
<body>
#Common_Data#
  <tr>
    <td class="formheading">Survey - cSurge evaluation survey</td>
  </tr>
  <tr>
    <td>
      <div class="aligncenter">
        <table>
          <tbody>
            <tr>
              <td>#greetings#</td>
            </tr>
            <tr>
              <td id="message1" class="message">&nbsp;</td>
            </tr>
            <tr>
              <td id="surveycompletionmessage">
                <div id="surveymessage"
class="surveynotfinished">
                  Important Note: After clicking on the
"Finish Survey" button at the end of this survey, please also
                  [ <button class="buttons"
onclick="javascript:SurveyInfoUpdate();">
                    Click Here
                  </button> ]
                  to update your survey information in
cSurge database as completed.
                </div>
              </td>
            </tr>
          </tbody>
        </table>
      </div>
    </td>
  </tr>
</body>
```

```

        <td id="message2" class="message">&nbsp;</td>
    </tr>
    <tr>
        <td id="surveyholder">
            <div style="margin:0 auto;text-align:center" class="surveyholder"><script type="text/javascript"
src="https://www.surveymonkey.com/jsEmbed.aspx?sm=gRuIzq5s8SRqTbMghIS5dg_3d_3d"></script></div>
        </td>
    </tr>
</tbody>
</table>
</div>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for survey_finished.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Survey</title>
    <link rel="icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="stylesheets/main.css"
media="screen" />
    <link rel="stylesheet" type="text/css" href="stylesheets/survey.css"
media="screen" />
    <script language="javascript" type="text/javascript"
src="js/menu.js"></script>
    <script language="javascript" type="text/javascript"
src="js/survey.js"></script>
</head>
<body>
#Common_Data#
    <tr>
        <td class="formheading">Survey - cSurge evaluation survey</td>
    </tr>
    <tr>
        <td>
            <div class="aligncenter">
                <table>
                    <tbody>
                        <tr>
                            <td>#greetings#</td>
                        </tr>
                        <tr>
                            <td id="message1" class="message">&nbsp;</td>

```

```

        </tr>
        <tr>
            <td id="surveycompletionmessage">
                <div id="surveymessage"
class="surveyfinished">
                    You have successfully finished the
survey. If you think this is an error, please contact cSurge administrator.
                </div>
            </td>
        </tr>
        <tr>
            <td id="message2" class="message">&nbsp;</td>
        </tr>
        <tr>
            <td id="surveyholder">
                &nbsp;</td>
        </tr>
    </tbody>
</table>
</div>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for userdataedit.html

```

<table>
  <tbody>
    <tr>
      <td class="descriptor1">
        <label for="User_iduser_id">
          <sup>1</sup>User ID*:
        </label>
      </td>
      <td class="field1" title="Enter User ID">
        <input type="text" name="User_iduser_id" id="User_iduser_id"
maxlength="30" size="20" tabindex="1" />
      </td>
      <td class="descriptor1">
        <label for="User_idpassword1">
          <sup>2</sup>Password*:
        </label>
      </td>
      <td class="field1" title="Enter Password">
        <input type="password" name="User_idpassword1"
id="User_idpassword1" maxlength="30" size="20" tabindex="2" />
      </td>
      <td class="descriptor1">
        <label for="User_idpassword2">
          Re-enter Password*:

```

```

        </label>
    </td>
    <td class="field1" title="Re-enter Password">
        <input type="password" name="User_idpassword2"
id="User_idpassword2" maxlength="30" size="20" tabindex="3" />
    </td>
</tr>
<tr>
    <td class="descriptor1">
        <label for="User_idfirstname">
            <sup>3</sup>First Name*:
        </label>
    </td>
    <td class="field1" title="Enter First Name of the User">
        <input type="text" name="User_idfirstname" id="User_idfirstname"
maxlength="30" size="20" tabindex="4" />
    </td>
    <td class="descriptor1">
        <label for="User_idlastname">
            <sup>3</sup>Last Name:
        </label>
    </td>
    <td class="field1" title="Enter Last Name of the User">
        <input type="text" name="User_idlastname" id="User_idlastname"
maxlength="30" size="20" tabindex="5" />
    </td>
    <td class="descriptor1">
        <label for="User_idmiddleinitial">
            <sup>4</sup>Middle Initial:
        </label>
    </td>
    <td class="field1" title="Enter Middle Initial of the User">
        <input type="text" name="User_idmiddleinitial"
id="User_idmiddleinitial" maxlength="2" size="20" tabindex="6" />
    </td>
</tr>
<tr>
    <td class="descriptor1">
        <label for="User_idemail1">
            Email address*:
        </label>
    </td>
    <td class="field1" title="Enter user email address">
        <input type="text" name="User_idemail1" id="User_idemail1"
maxlength="40" size="20" tabindex="7" />
    </td>
    <td class="descriptor1">
        <label for="User_idemail2">
            Re-Enter Email address*:
        </label>
    </td>
    <td class="field1" title="Re-Enter user email address">
        <input type="text" name="User_idemail2" id="User_idemail2"
maxlength="40" size="20" tabindex="8" />
    </td>
    <td class="descriptor1">
        <label for="User_idgroupname">
            Group assigned*:

```

```

        </label>
    </td>
    <td class="field1" title="Select the projects assigned. Press ctrl key
to select multiple studies">
        <select name="User_idgroupname" id="User_idgroupname"
style="width:100%" tabindex="9">
            <option selected="selected"
value="Not_selected">Select</option>
            <option value="Lab_user">Lab User</option>
            <option value="Guest">Guest</option>
            <option value="Admin">Administrator</option>
            <option value="Special">Special Access</option>
        </select>
    </td>
</tr>
<tr>
    <td class="descriptor1">
        <label for="User_idtestonly">
            <sup>5</sup>Test-only access:
        </label>
    </td>
    <td class="field1" style="text-align:center">
        <input type="checkbox" name="User_idtestonly" id="User_idtestonly"
tabindex="10" />
    </td>
    <td class="descriptor1">
        <label for="User_idallprojects">
            Projects assigned*:
        </label>
    </td>
    <td class="field1" title="" style="text-align:right;font-
size:12px;padding-right:68px">
        <label for="User_idallprojects">All </label><input type="checkbox"
name="User_idallprojects" id="User_idallprojects"
onclick="javascript:selectAll('\project\');" tabindex="11" /><br />
        <label for="User_idproject_MONR">MONR </label><input
type="checkbox" name="User_idproject_MONR" id="User_idproject_MONR"
onclick="javascript:selectOne('\project\',this);" tabindex="12" /><br />
        <label for="User_idproject_PONR">PONR </label><input
type="checkbox" name="User_idproject_PONR" id="User_idproject_PONR"
onclick="javascript:selectOne('\project\',this);" tabindex="13" /><br />
        <label for="User_idproject_DARPA">DARPA </label><input
type="checkbox" name="User_idproject_DARPA" id="User_idproject_DARPA"
onclick="javascript:selectOne('\project\',this);" tabindex="14" />
    </td>
    <td class="descriptor1">

    </td>
    <td class="field1" style="text-align:center">

    </td>
</tr>
</tbody>
</table>

```

Code written for advancedexportdisplay.py

```
#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sccl_advancedexportdisplay import *
```

Code written for the imported sccl_advancedexportdisplay module

```
#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql
import cgi
import cgi
#import cgi
#cgi.enable()
import _mysql
import re
import urllib
import string
import shutil
import os
import operator
import time
import datetime
import random
import bz2
import urllib2
import cookielib
import Cookie
from urllib2 import urlopen, Request
import httplib
#####
#####
connectparam = []
dbconnection = open('dbconnection', 'r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
dbconnection.close()
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
#####
#####
abc = open("Common_Data.html", 'r')
commondata = abc.read()
action=''
form = cgi.FieldStorage()
if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", "") )
    try:
        bloatvalue = a_cookie["bloat"].value
        connection = sql.connect(host=lhost, port=lport, user=luser,
passwd=lpasswd, db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SELECT User_id FROM sccl_authentication WHERE
Temp_key='%s'""%(bloatvalue))
```



```

output = cursor.fetchall()
connection.commit()
cursor.close()
connection.close()
if (len(output)==1):
    print 'Content-Type: text/html\n\n'
    filehandle = urllib.urlopen('advancedexportdisplay.html')
    for lines in filehandle.readlines( ):
        s = lines
        print s
    filehandle.close()
else:
    print ""Set-Cookie: bloat=; Path=""
    print ""Set-Cookie: page=index; Path=""
    print ""Set-Cookie: fileupload_Pig_ID=; Path=""
    print ""Set-Cookie: jsnewpass=; Path=""
    print ""Set-Cookie: jslogin=; Path=""
    print ""Set-Cookie: validate_filename=; Path=""
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'
except:
    print ""Set-Cookie: bloat=; Path=""
    print ""Set-Cookie: page=index; Path=""
    print ""Set-Cookie: fileupload_Pig_ID=; Path=""
    print ""Set-Cookie: jsnewpass=; Path=""
    print ""Set-Cookie: jslogin=; Path=""
    print ""Set-Cookie: validate_filename=; Path=""
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'
else:
    print ""Set-Cookie: bloat=; Path=""
    print ""Set-Cookie: page=index; Path=""
    print ""Set-Cookie: fileupload_Pig_ID=; Path=""
    print ""Set-Cookie: jsnewpass=; Path=""
    print ""Set-Cookie: jslogin=; Path=""
    print ""Set-Cookie: validate_filename=; Path=""
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'

```

Code written for datadownloadexcel.py

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sccl_data_download_excel import *

```

Code written for the imported sccl_data_download_excel module

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql
import cgi
import cgitb
#cgitb.enable()
import _mysql
import re
import urllib

```

```

import string
import shutil
import os
import operator
import time
import datetime
import random
import bz2
import urllib2
import cookielib
import Cookie
from urllib2 import urlopen, Request
#####
#####
alphabet =
'abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
uvwxyzabcdefghijklmnopqrstuvwxyz'
min = 24
max = 32
buscopan=''
for x in random.sample(alphabet,random.randint(min,max)):
    buscopan+=x
theKey = buscopan
#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
dbconnection.close()
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
#####
#####
abc = open("Common_Data.html",'r')
commondata = abc.read()
action=''
form = cgi.FieldStorage()
if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", "") )
    try:
        bloatvalue = a_cookie["bloat"].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SELECT User_id FROM sccl_authentication WHERE
Temp_key='%s'""%(bloatvalue))
        output = cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        if (len(output)==1):
            username=''
            for line in output:
                for i in line:

```

```

        username = i
    if(form.has_key('which')):
        action = form['which'].value
        if (action=='all_excel'):
            Data_type = form['Data_type'].value
            studyname= form['studyname'].value
            try:
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
                cursor = connection.cursor()
                cursor.execute("""SELECT
File_storage_path,File_name,File_size FROM monr_allsubjects_export WHERE
Data_type='%s'"""%(Data_type))
                output1 = cursor.fetchall()
                filepath = output1[0][0]+output1[0][1]
                filename = output1[0][1]
                filesize = output1[0][2]
                if not os.path.exists(filepath):
                    try:
                        os.remove(filepath)
                        os.removedirs(output[0][0])
                    except:
                        pass
                cursor.execute("""DELETE FROM monr_allsubjects_export
WHERE Data_type='%s'""%(Data_type))
            else:
                try:
                    filestring = open(filepath,'rb')
                    filedata = filestring.read()
                except:
                    pass
                connection.commit()
                cursor.close()
                connection.close()
                print 'Pragma: public'
                print 'Expires:0'
                print 'Cache-Control:must-revalidate,post-check=0,pre-
check=0'

                print 'Content-length:%d'%(filesize)
                print 'Content-Disposition:
attachment;filename=%s'%(filename)
                print 'Content-Description: database files'
                print 'Content-Transfer-Encoding:binary'
                print 'Content-Type: application/octet-stream\n'
                print """"%s""""%(filedata)
            except:
                pass
        elif (action=='individual_excel'):
            Pig_ID = form['Pig_ID'].value
            Data_type = form['Data_type'].value
            studyname= form['studyname'].value
            try:
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
                cursor = connection.cursor()
                cursor.execute("""SELECT
File_storage_path,File_name,File_size FROM monr_individual_export WHERE
Pig_ID='%s' AND Data_type='%s'""%(Pig_ID,Data_type))

```

```

output1 = cursor.fetchall()
filepath = output1[0][0]+output1[0][1]
filename = output1[0][1]
filesize = output1[0][2]
if not os.path.exists(filepath):
    try:
        os.remove(filepath)
        os.removedirs(output[0][0])
    except:
        pass
    cursor.execute("""DELETE FROM monr_individual_export
WHERE Pig_ID='%s' AND Data_type='%s'""%(Pig_ID,Data_type))
else:
    try:
        filestring = open(filepath,'rb')
        filedata = filestring.read()
        filestring.close()
    except:
        pass
connection.commit()
cursor.close()
connection.close()
print 'Pragma: public'
print 'Expires:0'
print 'Cache-Control:must-revalidate,post-check=0,pre-
check=0'

print 'Content-length:%d'%(filesize)
print 'Content-Disposition:
attachment;filename=%s'%(filename)
print 'Content-Description: database files'
print 'Content-Transfer-Encoding:binary'
print 'Content-Type: application/octet-stream\n'
print """"%s""""%(filedata)
except:
    pass
else:
    pass
else:
    pass
else:
    pass
except:
    pass
else:
    pass

```

Code written for dataview.py

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sccl_dataview import *

```

Code written for the imported sccl_dataview module

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql

```

```

import cgi
#import cgi.tb
#cgi.tb.enable()
import _mysql
import re
import urllib
import string
import shutil
import os
import operator
import time
import datetime
import random
import bz2
import urllib2
import cookielib
import Cookie
from urllib2 import urlopen, Request
import httplib
#####
#####
alphabet =
'abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz'
min = 24
max = 32
buscopan=''
for x in random.sample(alphabet,random.randint(min,max)):
    buscopan+=x
theKey = buscopan
#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
dbconnection.close()
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
#####
#####
abc = open("Common_Data.html",'r')
commondata = abc.read()
action=''
form = cgi.FieldStorage()
if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", "") )
    try:
        bloatvalue = a_cookie["bloat"].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SELECT User_id FROM sccl_authentication WHERE
Temp_key='%s'""%(bloatvalue))
        output = cursor.fetchall()

```

```

connection.commit()
cursor.close()
connection.close()
if (len(output)==1):
    print 'Content-Type: text/html\n\n'
    filehandle = urllib.urlopen('dataview.html')
    for lines in filehandle.readlines( ):
        s = lines
        print s
    filehandle.close()
else:
    print ""Set-Cookie: bloat=; Path="/"
    print ""Set-Cookie: page=index; Path="/"
    print ""Set-Cookie: fileupload_Pig_ID=; Path="/"
    print ""Set-Cookie: jsnewpass=; Path="/"
    print ""Set-Cookie: jslogin=; Path="/"
    print ""Set-Cookie: validate_filename=; Path="/"
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'
except:
    print ""Set-Cookie: bloat=; Path="/"
    print ""Set-Cookie: page=index; Path="/"
    print ""Set-Cookie: fileupload_Pig_ID=; Path="/"
    print ""Set-Cookie: jsnewpass=; Path="/"
    print ""Set-Cookie: jslogin=; Path="/"
    print ""Set-Cookie: validate_filename=; Path="/"
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'
else:
    print ""Set-Cookie: bloat=; Path="/"
    print ""Set-Cookie: page=index; Path="/"
    print ""Set-Cookie: fileupload_Pig_ID=; Path="/"
    print ""Set-Cookie: jsnewpass=; Path="/"
    print ""Set-Cookie: jslogin=; Path="/"
    print ""Set-Cookie: validate_filename=; Path="/"
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'

```

Code written for filedownload.py

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sccl_filedownload import *

```

Code written for imported sccl_filedownload module

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql
import cgi
#import cgitb
#cgitb.enable()
import re
import os
import random
import Cookie

```

```

#####
#####
alphabet =
'abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
uvvwxyzabcdefghijklmnopqrstuvwxyz'
min = 24
max = 32
buscopan=''
for x in random.sample(alphabet,random.randint(min,max)):
    buscopan+=x
theKey = buscopan
#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
dbconnection.close()
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
#####
#####
abc = open("Common_Data.html",'r')
commondata = abc.read()
action=''
form = cgi.FieldStorage()
if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", "") )
    try:
        bloatvalue = a_cookie["bloat"].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SELECT User_id FROM sccl_authentication WHERE
Temp_key='%s'""%(bloatvalue))
        output = cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        if (len(output)==1):
            username=''
            for line in output:
                for i in line:
                    username = i
            action=''
            form = cgi.FieldStorage()
            if(form.has_key('which')):
                action = form['which'].value
                if(action=='tempfiledownload'):
                    File_name = form['File_name'].value
                    Dir_name = form['tempdirectoryname'].value
                    try:
                        #connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=ldb)
                        #cursor = connection.cursor()

```

```

        #cursor.execute("""SELECT File_storage_path,Show_name FROM
monr_tempfilestorage WHERE File_name='%s' AND
Dir_name='%s'""%(File_name,Dir_name))
        #output1 = cursor.fetchall()
        #filepath,filename=output1[0]
        filepath="""/project/beilmanlab.umn.edu-
upload/MONR/temp_files/"""+Dir_name+'/' +File_name
        if re.search(r'.xls$',File_name):
            filename="Data_export.xls"
        else:
            filename="Data_export.csv"
        try:
            filestring = open(filepath,'rb')
            filedata = filestring.read()
            filesize = len(filedata)
        except:
            pass
            raise
        #connection.commit()
        #cursor.close()
        #connection.close()
        print 'Pragma: public'
        print 'Expires:0'
        print 'Cache-Control:must-revalidate,post-check=0,pre-
check=0'

        print 'Content-length:%d'%(filesize)
        print 'Content-Disposition:
attachment;filename=%s'%(filename)
        print 'Content-Description: database files'
        print 'Content-Transfer-Encoding:binary'
        print 'Content-Type: application/octet-stream\n'
        print """"%s""""%(filedata)
    except:
        pass
        raise
    else:
        if(form.has_key('studyname') and form.has_key('Pig_ID') and
form.has_key('Data_type')):
            Pig_ID = form['Pig_ID'].value
            Data_type = form['Data_type'].value
            try:
                connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=lhb)
                cursor = connection.cursor()
                cursor.execute("""SELECT
File_storage_path,File_name,File_size FROM monr_fileupload WHERE Pig_ID='%s' AND
Data_type='%s'""%(Pig_ID,Data_type))
                output1 = cursor.fetchall()
                filepath = output1[0][0]+"""/"+output1[0][1]
                filename = output1[0][1]
                filesize = output1[0][2]
            try:
                filestring = open(filepath,'rb')
                filedata = filestring.read()
            except:
                pass
            connection.commit()
            cursor.close()

```



```

        connection.close()
        print 'Pragma: public'
        print 'Expires:0'
        print 'Cache-Control:must-revalidate,post-check=0,pre-
check=0'

        print 'Content-length:%d'%(filesize)
        print 'Content-Disposition:
attachment;filename=%s'%(filename)
        print 'Content-Description: database files'
        print 'Content-Transfer-Encoding:binary'
        print 'Content-Type: application/octet-stream\n'
        print """"%s""""%(filedata)
    except:
        pass
    else:
        pass
    else:
        pass
    else:
        pass
    except:
        pass
        raise
else:
    pass

```

Code written for formsubmit.py

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sctl_formsubmit import *

```

Code written for imported sctl_formsubmit module

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql
import cgi
#import cgitb
#cgitb.enable()
import _mysql
import re
import urllib
import string
import shutil
import os
import operator
import time
import datetime
import random
import bz2
import urllib2
import cookielib
import Cookie
from raj_excel_export import *
import raj_excel_export
import sys

```

```

from tempfile import *
from urllib2 import urlopen,Request
try: # Windows needs stdio set for binary mode.
    import msvcrt
    msvcrt.setmode (0, os.O_BINARY) # stdin = 0
    msvcrt.setmode (1, os.O_BINARY) # stdout = 1
except ImportError:
    pass
#####
#####
def randkey(min,max):
    alphabet =
'abcdefghijklmnopqrstuvwxyz32434stuvwssxyzabcde23432465fghijklmnopq7rstu7wxyzrstu6764evwx
yzabcd2efghijklmno9523pqrstuv32wx4353yzabcdefgghijklmn79opqrstuvwxyz23rstuvwx'
    if min>=len(alphabet):
        min=len(alphabet)-1
    else:
        pass
    if max>=len(alphabet):
        max=len(alphabet)
    else:
        pass
    buscopan=''
    for x in random.sample(alphabet,random.randint(min,max)):
        buscopan+=x
    theKey = buscopan
    return theKey
#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
#####
#####
#####
def connect_to_db_and_execute(the_query):
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
    cursor = connection.cursor()
    cursor.execute("""%s"""%the_query)
    connection.commit()
    cursor.close()
    connection.close()
#####
#####
#####
def connect_to_db_and_return_result(the_query):
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
    cursor = connection.cursor()
    cursor.execute("""%s"""%the_query)
    queryresult=cursor.fetchall()

```

```

        connection.commit()
        cursor.close()
        connection.close()
        return queryresult
#####
#####
#####
studyname = "MONR"
if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", "") )
    #try:
        bloatvalue = a_cookie["bloat"].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SELECT User_id, Group_name FROM scc1_authentication WHERE
Temp_key='%s'""%(bloatvalue))
        output100 = cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        if (len(output100)==1):
            current_user_id = output100[0][0]
            current_user_group = output100[0][1]
            action=''
            form = cgi.FieldStorage()
            if(form.has_key('which')):
                action = form['which'].value
                if (action == 'alertGroup'):
                    try:
                        if (current_user_group!='Admin'):
                            print 'Content-Type: text/javascript\n'
                            print ""alert('Only an Administrator can use this
feature');""
                    else:
                        print 'Content-Type: text/javascript\n'
                    except:
                        print 'Content-Type: text/javascript\n'
                        #raise
                elif (action=='checkgroupenabledatalock'):
                    try:
                        if (current_user_group)=='Admin':
                            print 'Content-Type: text/javascript\n'
                            print ""if
(document.form1.lockdata.getAttribute('disabled')){document.form1.lockdata.removeA
ttribute('disabled');}else { }""
                    else:
                        print 'Content-Type: text/javascript\n'
                    except:
                        print 'Content-Type: text/javascript\n'
                        #raise

                elif (action == 'idcheck'):
                    try:
                        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
                        cursor = connection.cursor()
                        Pig_ID = form['pig_id'].value

```

```

        cursor.execute("""SELECT Pig_ID FROM monr_identifier_table
WHERE Pig_ID='%s'""%(Pig_ID))
        output = cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        if (len(output)>0):
            print 'Content-Type: text/javascript\n'
            print ""duplicateid();""
        else:
            print 'Content-Type: text/javascript\n'
            print ""startDatecreate();""
    except:
        print 'Content-Type: text/javascript\n'
        print ""document.getElementById('message1').innerHTML='There
was an error in the ID. Please contact the data manager.';""
        #raise
    elif(action == 'mainformsubmit'):
        try:
            a=''
            b=''
            c=''
            d=''
            e=''
            f=''
            g=''
            h=''
            i=''
            j=''
            k=''
            Pig_ID = form['pig_id'].value
            Pig_group = form['group'].value
            Weight = float(form['weight'].value)
            Start_day = str(form['start_day'].value)
            Start_month = str(form['start_month'].value)
            Start_year = str(form['start_year'].value)
            Start_date = (Start_year+"-"+Start_month+"-"+Start_day)
            connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
            cursor = connection.cursor()
            try:
                cursor.execute("""INSERT INTO
monr_identifier_table(Pig_ID,Pig_group,Weight,Start_date) VALUES
("%s", "%s", %f, "%s")""%(Pig_ID,Pig_group,Weight,Start_date))
            try:
                if (form.has_key('bloodremoved')):
                    Blood_removed = form['bloodremoved'].value
                    cursor.execute("""UPDATE monr_identifier_table SET
Blood_removed = "%s" WHERE Pig_ID='%s'""%(Blood_removed,Pig_ID))
                    a=
                    ""document.form1.bloodremoved.value="%s";document.form1.bloodremoved.setAttribute
('disabled', 'disabled');""%(Blood_removed)
                else:
                    pass
                if (form.has_key('orfluids')):
                    OR_fluids = form['orfluids'].value
                    cursor.execute("""UPDATE monr_identifier_table SET
OR_fluids = "%s" WHERE Pig_ID='%s'""%(OR_fluids,Pig_ID))

```

```

        b=
        """document.form1.orfluids.value="%s";document.form1.orfluids.setAttribute('disabled', 'disabled');"""%(OR_fluids)
        else:
            pass
            if (form.has_key('bloodgiven')):
                Blood_given = form['bloodgiven'].value
                cursor.execute("""UPDATE monr_identifier_table SET
Blood_given = "%s" WHERE Pig_ID='%s'"""%(Blood_given,Pig_ID))
            c=
            """document.form1.bloodgiven.value="%s";document.form1.bloodgiven.setAttribute('disabled', 'disabled');"""%(Blood_given)
            else:
                pass
                if (form.has_key('lrgiven')):
                    LR_given = form['lrgiven'].value
                    cursor.execute("""UPDATE monr_identifier_table SET
LR_given = "%s" WHERE Pig_ID='%s'"""%(LR_given,Pig_ID))
            d=
            """document.form1.lrgiven.value="%s";document.form1.lrgiven.setAttribute('disabled', 'disabled');"""%(LR_given)
            else:
                pass
                if (form.has_key('totalvolume')):
                    Total_volume_given = form['totalvolume'].value
                    cursor.execute("""UPDATE monr_identifier_table SET
Total_volume_given = "%s" WHERE Pig_ID='%s'"""%(Total_volume_given,Pig_ID))
            e=
            """document.form1.totalvolume.value="%s";document.form1.totalvolume.setAttribute('disabled', 'disabled');"""%(Total_volume_given)
            else:
                pass
                if (form.has_key('end_year') and
form.has_key('end_day') and form.has_key('end_month')):
                    End_day = str(form['end_day'].value)
                    End_month = str(form['end_month'].value)
                    End_year = str(form['end_year'].value)
                    End_date = (End_year+"-"+End_month+"-"+End_day)
                    cursor.execute("""UPDATE monr_identifier_table SET
End_date = "%s" WHERE Pig_ID='%s'"""%(End_date,Pig_ID))
                    f= """document.form1.end_month.value= "%s";
                    document.form1.end_day.value= "%s";
                    document.form1.end_year.value= "%s";

                    document.form1.end_month.setAttribute('disabled', 'disabled');

                    document.form1.end_day.setAttribute('disabled', 'disabled');

                    document.form1.end_year.setAttribute('disabled', 'disabled');"""%(End_month,End_day
                    ,End_year)
                else:
                    pass
                    if (form.has_key('died')):
                        Died = form['died'].value
                        cursor.execute("""UPDATE monr_identifier_table SET
Died = "%s" WHERE Pig_ID='%s'"""%(Died,Pig_ID))

```

```

g=
"""document.form1.died.value="%s";document.form1.died.setAttribute('disabled','dis
abled');"""%(Died)
    else:
        pass
    if (form.has_key('died_hours') and
form.has_key('died_minutes')):
        Hour_of_death = str(form['died_hours'].value)
        Minute_of_death = str(form['died_minutes'].value)
        Time_of_death = (Hour_of_death+ ":" +
Minute_of_death + ":" + "00")
        cursor.execute("""UPDATE monr_identifier_table SET
Time_of_death = "%s" WHERE Pig_ID='%s'"""%(Time_of_death,Pig_ID))
        h= """document.form1.died_hours.value="%s";
document.form1.died_minutes.value="%s";

document.form1.died_hours.setAttribute('disabled','disabled');

document.form1.died_minutes.setAttribute('disabled','disabled');"""%(Hour_of_death
,Minute_of_death)
    else:
        pass
    if (form.has_key('causeofdeath')):
        Cause_of_death = form['causeofdeath'].value
        cursor.execute("""UPDATE monr_identifier_table SET
Cause_of_death = "%s" WHERE
Pig_ID='%s'"""%(sql.escape_string(Cause_of_death),Pig_ID))
        i=
        """document.form1.causeofdeath.value="%s";document.form1.causeofdeath.setAttribute
('disabled','disabled');"""%(Cause_of_death)
    else:
        pass
    if (form.has_key('overallcomments')):
        Comments = form['overallcomments'].value
        Comments = Comments.replace("\n"," ")
        Comments = Comments.replace("\r"," ")
        cursor.execute("""UPDATE monr_identifier_table SET
Comments = "%s" WHERE Pig_ID='%s'"""%(sql.escape_string(Comments),Pig_ID))
        j=
        """document.form1.overallcomments.value="%s";document.form1.overallcomments.setAtt
ribute('disabled','disabled');"""%(Comments)
    else:
        pass
    k=
    """document.form1.pig_id.setAttribute('disabled','disabled');

document.form1.group.setAttribute('disabled','disabled');

document.form1.weight.setAttribute('disabled','disabled');

document.form1.start_month.setAttribute('disabled','disabled');

document.form1.start_day.setAttribute('disabled','disabled');

document.form1.start_year.setAttribute('disabled','disabled');"""
    if (form.has_key('lockdata')):
        Datainput_status = form['lockdata'].value

```

```

        cursor.execute("""UPDATE monr_identifier_table SET
Datainput_status = "%s" WHERE Pig_ID='%s'"""%(Datainput_status,Pig_ID))
        l=
        """document.form1.lockdata.setAttribute('disabled','disabled');"""
        else:
            pass
        abc = a+b+c+d+e+f+g+h+i+j+k+l
        dbc = """document.form1.pig_id.value='%s';
        document.form1.group.value="%s";
        document.form1.weight.value="%.1f";
        document.form1.start_month.value="%s";
        document.form1.start_day.value="%s";
        document.form1.start_year.value="%s";
        changeOnClick();
        var givealert1 =

document.getElementById('message1');
        givealert1.innerHTML = 'Identifiers
successfully
initiated';"""%(Pig_ID,Pig_group,Weight,Start_month,Start_day,Start_year)
        final = (abc+dbc).replace('\n','')
        print 'Content-Type: text/javascript\n'
        print final
        print """enableSelectform();"""
    except:
        try:
            connection.rollback()
        except:
            pass
            #raise
        print 'Content-Type: text/javascript\n'
        print
        """document.getElementById('message1').innerHTML='There was an error. Please
contact the data manager.';
        setTimeout("pageReload()",1250);"""
        #raise
    except:
        try:
            cursor.execute("""SELECT * FROM monr_identifier_table
WHERE Pig_ID='%s'"""%(Pig_ID))
            output = cursor.fetchall()
            if (len(output)>0):
                Pig_ID=output[0][0]
                start_date=str(output[0][1]).split("-")
                Start_year=start_date[0]
                Start_month=start_date[1]
                Start_day=start_date[2]
                if not (output[0][2]==None):
                    end_date=str(output[0][2]).split("-")
                    End_year=end_date[0]
                    End_month=end_date[1]
                    End_day=end_date[2]
                else:
                    End_year=''
                    End_month=''
                    End_day=''
                Weight=output[0][3]
                Pig_group=output[0][4]
                if not (output[0][5]==None):

```

```

        Blood_removed=output[0][5]
else:
    Blood_removed=''
if not (output[0][6]==None):
    OR_fluids=output[0][6]
else:
    OR_fluids=''
if not (output[0][7]==None):
    Blood_given=output[0][7]
else:
    Blood_given=''
if not (output[0][8]==None):
    LR_given=output[0][8]
else:
    LR_given=''
if not (output[0][9]==None):
    Total_volume_given=output[0][9]
else:
    Total_volume_given=''
if not (output[0][10]==None):
    Died=output[0][10]
else:
    Died=''
if not (output[0][11]==None):
    died_time=str(output[0][11]).split(':')
    Hour_of_death=died_time[0]
    Minute_of_death=died_time[1]
else:
    Hour_of_death=''
    Minute_of_death=''
if not (output[0][12]==None):
    Cause_of_death=output[0][12]

Cause_of_death=Cause_of_death.replace("\'", "\\\'")
Cause_of_death=Cause_of_death.replace("\\"", "\\\\"")
Cause_of_death=Cause_of_death.replace("\n", "<br />")
else:
    Cause_of_death=''
if not (output[0][13]==None):
    Comments=output[0][13]
    Comments=Comments.replace("\'", "\\\'")
    Comments=Comments.replace("\\"", "\\\\"")
    Comments=Comments.replace("\n", "<br />")
else:
    Comments=''
if not (output[0][14]==None):
    Datainput=output[0][14]
else:
    Datainput='Not finished'
print 'Content-Type: text/javascript\n'
print ""document.form1.group.value="%s";
    document.form1.weight.value="%.1f";
    document.form1.start_month.value="%s";
    document.form1.start_day.value="%s";
    document.form1.start_year.value="%s";
    document.form1.end_month.value= "%s";

```



```

document.form1.end_day.value= "%s";
document.form1.end_year.value= "%s";
document.form1.died.value="%s";
document.form1.died_hours.value="%s";
document.form1.died_minutes.value="%s";
document.form1.causeofdeath.value="%s";

document.form1.overallcomments.value="%s";
document.form1.bloodremoved.value="%s";
document.form1.orfluids.value="%s";
document.form1.bloodgiven.value="%s";
document.form1.lrgiven.value="%s";
document.form1.totalvolume.value="%s";
changeOnClick();
enableSelectform();
var Datainput="%s";
if
(Datainput=="Finished"){document.form1.lockdata.checked=false;}else
{document.form1.lockdata.checked=true;}

document.getElementById('message1');
givealert1.innerHTML = 'Pig ID already
exists. Showing previously entered data. Click on edit to edit data';

disableform1();""%(Pig_group,Weight,Start_month,Start_day,Start_year,End_month,En
d_day,End_year,Died,Hour_of_death,Minute_of_death,Cause_of_death,Comments,Blood_re
moved,OR_fluids,Blood_given,LR_given>Total_volume_given,Datainput)
else:
print 'Content-Type: text/javascript\n'
print ""var givealert1 =
document.getElementById('message1');
givealert1.innerHTML = 'The ID you have
entered does not exist in the database. To create a new entry, you must complete
ID, Group and Weight fields';
startDatecreate();""
except:
print 'Content-Type: text/javascript\n'
print ""var givealert1 =
document.getElementById('message1');
givealert1.innerHTML = 'The ID you have
entered does not exist in the database. To create a new entry, you must complete
ID, Group and Weight fields';
startDatecreate();""
#raise
#raise
except:
try:
connection.rollback()
except:
pass
#raise
print 'Content-Type: text/javascript\n'
print ""document.getElementById('message1').innerHTML='There
was an error. Please contact the data manager.';
setTimeout("pageReload()",1250);""
#raise
if(connection):
connection.commit()

```

```

else:
    pass
if(cursor):
    cursor.close()
else:
    pass
if(connection):
    connection.close()
else:
    pass
elif(action == "mainform1submit"):
    try:
        a=''
        b=''
        c=''
        d=''
        e=''
        f=''
        g=''
        h=''
        i=''
        j=''
        k=''
        Pig_ID = form['pig_id'].value
        Pig_group = form['group'].value
        Weight = float(form['weight'].value)
        Start_day = str(form['start_day'].value)
        Start_month = str(form['start_month'].value)
        Start_year = str(form['start_year'].value)
        Start_date = (Start_year+"-"+Start_month+"-"+Start_day)
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
        cursor = connection.cursor()
        cursor.execute("""UPDATE monr_identifier_table SET
Pig_group=%s,Weight=%f,Start_date=%s" WHERE
Pig_ID=%s'""%(Pig_group,Weight,Start_date,Pig_ID))
        connection.commit()
        cursor.close()
        connection.close()
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
        cursor = connection.cursor()
        if (form.has_key('bloodremoved')):
            Blood_removed = form['bloodremoved'].value
            cursor.execute("""UPDATE monr_identifier_table SET
Blood_removed = %s" WHERE Pig_ID=%s'""%(Blood_removed,Pig_ID))
            a=
            """document.form1.bloodremoved.value=%s";document.form1.bloodremoved.setAttribute
('disabled','disabled');""%(Blood_removed)
        else:
            pass
            if (form.has_key('orfluids')):
                OR_fluids = form['orfluids'].value
                cursor.execute("""UPDATE monr_identifier_table SET
OR_fluids = %s" WHERE Pig_ID=%s'""%(OR_fluids,Pig_ID))
                b=
                """document.form1.orfluids.value=%s";document.form1.orfluids.setAttribute('disabl
ed','disabled');""%(OR_fluids)

```

```

else:
    pass
    if (form.has_key('bloodgiven')):
        Blood_given = form['bloodgiven'].value
        cursor.execute("""UPDATE monr_identifier_table SET
Blood_given = "%s" WHERE Pig_ID='%s'"""%(Blood_given,Pig_ID))
        c=
        """document.form1.bloodgiven.value="%s";document.form1.bloodgiven.setAttribute('di
sabled','disabled');"""%(Blood_given)
    else:
        pass
        if (form.has_key('lrgiven')):
            LR_given = form['lrgiven'].value
            cursor.execute("""UPDATE monr_identifier_table SET
LR_given = "%s" WHERE Pig_ID='%s'"""%(LR_given,Pig_ID))
            d=
            """document.form1.lrgiven.value="%s";document.form1.lrgiven.setAttribute('disabled
','disabled');"""%(LR_given)
        else:
            pass
            if (form.has_key('totalvolume')):
                Total_volume_given = form['totalvolume'].value
                cursor.execute("""UPDATE monr_identifier_table SET
Total_volume_given = "%s" WHERE Pig_ID='%s'"""%(Total_volume_given,Pig_ID))
                e=
                """document.form1.totalvolume.value="%s";document.form1.totalvolume.setAttribute('
disabled','disabled');"""%(Total_volume_given)
            else:
                pass
                if (form.has_key('end_year') and form.has_key('end_day') and
form.has_key('end_month')):
                    End_day = str(form['end_day'].value)
                    End_month = str(form['end_month'].value)
                    End_year = str(form['end_year'].value)
                    End_date = (End_year+"-"+End_month+"-"+End_day)
                    cursor.execute("""UPDATE monr_identifier_table SET
End_date = "%s" WHERE Pig_ID='%s'"""%(End_date,Pig_ID))
                    f= """document.form1.end_month.value= "%s";
                        document.form1.end_day.value= "%s";
                        document.form1.end_year.value= "%s";

document.form1.end_month.setAttribute('disabled','disabled');

document.form1.end_day.setAttribute('disabled','disabled');

document.form1.end_year.setAttribute('disabled','disabled');"""%(End_month,End_day
,End_year)
                else:
                    pass
                    if (form.has_key('died')):
                        Died = form['died'].value
                        cursor.execute("""UPDATE monr_identifier_table SET Died =
%s" WHERE Pig_ID='%s'"""%(Died,Pig_ID))
                        g=
                        """document.form1.died.value="%s";document.form1.died.setAttribute('dis
abled','disabled');"""%(Died)
                    else:
                        pass

```

```

        if (form.has_key('died_hours') and
form.has_key('died_minutes')):
            Hour_of_death = str(form['died_hours'].value)
            Minute_of_death = str(form['died_minutes'].value)
            Time_of_death = (Hour_of_death+ ":" + Minute_of_death +
":" + "00")

            cursor.execute("""UPDATE monr_identifier_table SET
Time_of_death = "%s" WHERE Pig_ID='%s'"""%(Time_of_death,Pig_ID))
            h= """document.form1.died_hours.value="%s";
            document.form1.died_minutes.value="%s";

document.form1.died_hours.setAttribute('disabled','disabled');

document.form1.died_minutes.setAttribute('disabled','disabled');"""%(Hour_of_death
,Minute_of_death)
        else:
            pass
        if (form.has_key('causeofdeath')):
            Cause_of_death = form['causeofdeath'].value
            Cause_of_death = Cause_of_death.replace("\n"," ")
            Cause_of_death = Cause_of_death.replace("\r"," ")
            cursor.execute("""UPDATE monr_identifier_table SET
Cause_of_death = "%s" WHERE
Pig_ID='%s'"""%(sql.escape_string(Cause_of_death),Pig_ID))
            i=
            """document.form1.causeofdeath.value="%s";document.form1.causeofdeath.setAttribute
('disabled','disabled');"""%(Cause_of_death)
        else:
            pass
        if (form.has_key('overallcomments')):
            Comments = form['overallcomments'].value
            Comments = Comments.replace("\n"," ")
            Comments = Comments.replace("\r"," ")
            cursor.execute("""UPDATE monr_identifier_table SET
Comments = "%s" WHERE Pig_ID='%s'"""%(sql.escape_string(Comments),Pig_ID))
            j=
            """document.form1.overallcomments.value="%s";document.form1.overallcomments.setAtt
ribute('disabled','disabled');"""%(Comments)
        else:
            pass
        k=
        """document.form1.group.setAttribute('disabled','disabled');

document.form1.weight.setAttribute('disabled','disabled');

document.form1.start_month.setAttribute('disabled','disabled');

document.form1.start_day.setAttribute('disabled','disabled');

document.form1.start_year.setAttribute('disabled','disabled');"""
        if (form.has_key('lockdata')):
            Datainput_status = form['lockdata'].value
            cursor.execute("""UPDATE monr_identifier_table SET
Datainput_status = "%s" WHERE Pig_ID='%s'"""%(Datainput_status,Pig_ID))
            l=
            """document.form1.lockdata.setAttribute('disabled','disabled');"""
        else:
            pass

```

```

abc = a+b+c+d+e+f+g+h+i+j+k+l
dbc = """document.form1.pig_id.value='%s';
        document.form1.group.value="%s";
        document.form1.weight.value="%.1f";
        document.form1.start_month.value="%s";
        document.form1.start_day.value="%s";
        document.form1.start_year.value="%s";
        var givealert1 = document.getElementById('message1');
        givealert1.innerHTML = 'Identifiers successfully
updated';""%(Pig_ID,Pig_group,Weight,Start_month,Start_day,Start_year)
final = (abc+dbc).replace('\n', '')
print 'Content-Type: text/javascript\n'
print final
print """enableSelectform();"""
except:
try:
    connection.rollback()
except:
    pass
    #raise
print 'Content-Type: text/javascript\n'
print """document.getElementById('message1').innerHTML='There
was an error. Please contact the data manager';"""
print """enableSelectform();"""
print """disableform1();"""
#raise
if(connection):
    connection.commit()
else:
    pass
if(cursor):
    cursor.close()
else:
    pass
if(connection):
    connection.close()
else:
    pass
elif(action == "retrievedata"):
try:
    Pig_ID = form['pig_id'].value
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
    cursor = connection.cursor()
    cursor.execute("""SELECT * FROM monr_identifier_table WHERE
Pig_ID='%s'""%(Pig_ID))
    output = cursor.fetchall()
    connection.commit()
    cursor.close()
    connection.close()
    if (len(output)>0):
        Pig_ID=output[0][0]
        start_date=str(output[0][1]).split("-")
        Start_year=start_date[0]
        Start_month=start_date[1]
        Start_day=start_date[2]
        if not (output[0][2]==None):
            end_date=str(output[0][2]).split("-")

```

```

        End_year=end_date[0]
        End_month=end_date[1]
        End_day=end_date[2]
else:
    End_year=''
    End_month=''
    End_day=''
Weight=output[0][3]
Pig_group=output[0][4]
if not (output[0][5]==None):
    Blood_removed=output[0][5]
else:
    Blood_removed=''
if not (output[0][6]==None):
    OR_fluids=output[0][6]
else:
    OR_fluids=''
if not (output[0][7]==None):
    Blood_given=output[0][7]
else:
    Blood_given=''
if not (output[0][8]==None):
    LR_given=output[0][8]
else:
    LR_given=''
if not (output[0][9]==None):
    Total_volume_given=output[0][9]
else:
    Total_volume_given=''
if not (output[0][10]==None):
    Died=output[0][10]
else:
    Died=''
if not (output[0][11]==None):
    died_time=str(output[0][11]).split(':')
    Hour_of_death=died_time[0]
    Minute_of_death=died_time[1]
else:
    Hour_of_death=''
    Minute_of_death=''
if not (output[0][12]==None):
    Cause_of_death=output[0][12]
    Cause_of_death=Cause_of_death.replace("\'","\\\'")
    Cause_of_death=Cause_of_death.replace("\", "\\")
    Cause_of_death=Cause_of_death.replace("\n", "<br />")
else:
    Cause_of_death=''
if not (output[0][13]==None):
    Comments=output[0][13]
    Comments=Comments.replace("\'","\\\'")
    Comments=Comments.replace("\", "\\")
    Comments=Comments.replace("\n", "<br />")
else:
    Comments=''
if not (output[0][14]==None):
    Datainput=output[0][14]
else:
    Datainput='Not finished'

```

```

print 'Content-Type: text/javascript\n'
print """document.form1.group.value="%s";
        document.form1.weight.value="%1f";
        document.form1.start_month.value="%s";
        document.form1.start_day.value="%s";
        document.form1.start_year.value="%s";
        document.form1.end_month.value= "%s";
        document.form1.end_day.value= "%s";
        document.form1.end_year.value= "%s";
        document.form1.died.value="%s";
        document.form1.died_hours.value="%s";
        document.form1.died_minutes.value="%s";
        document.form1.causeofdeath.value="%s";
        document.form1.overallcomments.value="%s";
        document.form1.bloodremoved.value="%s";
        document.form1.orfluids.value="%s";
        document.form1.bloodgiven.value="%s";
        document.form1.lrgiven.value="%s";
        document.form1.totalvolume.value="%s";
        changeOnClick();
        enableSelectform();
        var Datainput="%s";
        if
(Datainput=="Finished"){document.form1.lockdata.checked=false;}else
{document.form1.lockdata.checked=true;}
        var givealert1 =
document.getElementById('message1');
        givealert1.innerHTML = 'Showing previously
entered data';

disableform1());"""% (Pig_group,Weight,Start_month,Start_day,Start_year,End_month,En
d_day,End_year,Died,Hour_of_death,Minute_of_death,Cause_of_death,Comments,Blood_re
moved,OR_fluids,Blood_given,LR_given,Total_volume_given,Datainput)
        else:
            print 'Content-Type: text/javascript\n'
            print """var givealert1 =
document.getElementById('message1');
                givealert1.innerHTML = 'The ID you have entered
does not exist in the database. To create a new entry, you must complete ID, Group
and Weight fields';
                    startDatecreate();"""
            except:
                print 'Content-Type: text/javascript\n'
                print """var givealert1 = document.getElementById('message1');
                    givealert1.innerHTML = 'The ID you have entered does
not exist in the database. To create a new entry, you must complete ID, Group and
Weight fields';
                        startDatecreate();"""
                #raise

#####
#####
#####
elif(action=="checkVitalsCount"):
    try:
        Pig_ID = form['pig_id'].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lodb)

```

```

        cursor = connection.cursor()
        # The below request selects the complex join query as an alias
        table and selects the count from it. This is useful for pagination
        cursor.execute("""SELECT COUNT(*) FROM(SELECT
`monr_vitals_manual_10_min`.Date_of_experiment,`monr_vitals_manual_10_min`.Time_of
_experiment,`monr_vitals_manual_10_min`.Heart_Rate,`monr_vitals_manual_10_min`.Art
_Sys,`monr_vitals_manual_10_min`.Art_Dia,`monr_vitals_manual_10_min`.MeanArtPres,`
monr_vitals_manual_10_min`.PA_Sys,`monr_vitals_manual_10_min`.PA_Dia,`monr_vitals_
manual_10_min`.sto2_multidepth,`monr_vitals_manual_10_min`.thi_multidepth,`monr_vi
tals_manual_10_min`.StO2_15mm,`monr_vitals_manual_10_min`.THI_15mm,`monr_vitals_ma
nual_10_min`.FiO2,`monr_vitals_manual_10_min`.Propofol,`monr_vitals_manual_10_min`.
.BIS,`monr_vitals_manual_10_min`.Event_name,`monr_vitals_manual_one_hour`.Wedge_pr
essure,`monr_vitals_manual_one_hour`.Urine_output,`monr_vitals_manual_10_min`.Temp
erature,`monr_vitals_manual_one_hour`.Cardiac_output,`monr_vitals_manual_one_hour`.
Arterial_pH,`monr_vitals_manual_one_hour`.Arterial_PCO2,`monr_vitals_manual_one_h
our`.Arterial_PO2,`monr_vitals_manual_one_hour`.Arterial_O2_sat,`monr_vitals_manua
l_one_hour`.Venous_PO2,`monr_vitals_manual_one_hour`.Venous_O2_sat,`monr_vitals_ma
nual_one_hour`.Hb,`monr_vitals_manual_one_hour`.DO2,`monr_vitals_manual_one_hour`.
VO2,`monr_vitals_manual_one_hour`.ERO2,`monr_vitals_manual_one_hour`.Lactate,`monr
_vitals_manual_one_hour`.Base_deficit,`monr_vitals_manual_one_hour`.Bladder_pressu
re,`monr_vitals_manual_one_hour`.Chesttube_output,`monr_vitals_manual_10_min`.Even
t_comments,`monr_vitals_manual_one_hour`.Specificgravity_urine FROM
`monr_vitals_manual_10_min` LEFT JOIN `monr_vitals_manual_one_hour` ON
`monr_vitals_manual_10_min`.Date_of_experiment=`monr_vitals_manual_one_hour`.Date_
of_experiment AND
`monr_vitals_manual_10_min`.Time_of_experiment=`monr_vitals_manual_one_hour`.Time_
of_experiment WHERE `monr_vitals_manual_10_min`.Pig_ID='%s' ORDER BY
Date_of_experiment,Time_of_experiment) AS table_alias"""%(Pig_ID))
        output3 = cursor.fetchall()
        vitals_total_data_points = output3[0][0]
        connection.commit()
        cursor.close()
        connection.close()
    except:
        print "There was an error"
        #raise

#####
#####
#####
        elif (action == "viewallvitals") or (action ==
"viewallvitalsnewwindow"):
            #try:
            Pig_ID = form['pig_id'].value
            connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
            cursor = connection.cursor()
            cursor.execute("""SELECT * FROM monr_identifier_table WHERE
Pig_ID='%s'""%(Pig_ID))
            output = cursor.fetchall()
            vitalsdataorder = str(form['vitalsdataorder'].value)
            if(vitalsdataorder=='Descending'):
                vitalsdataorder='DESC'
            else:
                vitalsdataorder='ASC'
            lowerlimit = int(form['lowerlimit'].value)
            starting_row = lowerlimit+1
            numberofdatapoints = int(form['numberofdatapoints'].value)

```



```

ending_row = lowerlimit+numberofdatapoints
if(action == "viewallvitals"):
    cursor.execute("""SELECT
`monr_vitals_manual_10_min`.Date_of_experiment,`monr_vitals_manual_10_min`.Time_of
_experiment,`monr_vitals_manual_10_min`.Heart_Rate,`monr_vitals_manual_10_min`.Art
_Sys,`monr_vitals_manual_10_min`.Art_Dia,`monr_vitals_manual_10_min`.MeanArtPres,`
monr_vitals_manual_10_min`.PA_Sys,`monr_vitals_manual_10_min`.PA_Dia,`monr_vitals_
manual_10_min`.sto2_multidepth,`monr_vitals_manual_10_min`.thi_multidepth,`monr_vi
tals_manual_10_min`.StO2_15mm,`monr_vitals_manual_10_min`.THI_15mm,`monr_vitals_ma
nual_10_min`.FiO2,`monr_vitals_manual_10_min`.Propofol,`monr_vitals_manual_10_min`
.BIS,`monr_vitals_manual_10_min`.Event_name,`monr_vitals_manual_one_hour`.Wedge_pr
essure,`monr_vitals_manual_one_hour`.Urine_output,`monr_vitals_manual_10_min`.Temp
erature,`monr_vitals_manual_one_hour`.Cardiac_output,`monr_vitals_manual_one_hour`
.Arterial_pH,`monr_vitals_manual_one_hour`.Arterial_PC02,`monr_vitals_manual_one_h
our`.Arterial_PO2,`monr_vitals_manual_one_hour`.Arterial_O2_sat,`monr_vitals_manua
l_one_hour`.Venous_PO2,`monr_vitals_manual_one_hour`.Venous_O2_sat,`monr_vitals_ma
nual_one_hour`.Hb,`monr_vitals_manual_one_hour`.DO2,`monr_vitals_manual_one_hour`.
VO2,`monr_vitals_manual_one_hour`.ERO2,`monr_vitals_manual_one_hour`.Lactate,`monr
_vitals_manual_one_hour`.Base_deficit,`monr_vitals_manual_one_hour`.Bladder_pressu
re,`monr_vitals_manual_one_hour`.Chesttube_output,`monr_vitals_manual_10_min`.Even
t_comments,`monr_vitals_manual_one_hour`.Specificgravity_urine,`monr_vitals_manual
_one_hour`.Sodium,`monr_vitals_manual_one_hour`.Potassium,`monr_vitals_manual_one
_hour`.Calcium,`monr_vitals_manual_one_hour`.Glucose FROM
`monr_vitals_manual_10_min` LEFT JOIN `monr_vitals_manual_one_hour` ON
`monr_vitals_manual_10_min`.Date_of_experiment=`monr_vitals_manual_one_hour`.Date_
of_experiment AND
`monr_vitals_manual_10_min`.Time_of_experiment=`monr_vitals_manual_one_hour`.Time_
of_experiment WHERE `monr_vitals_manual_10_min`.Pig_ID='%s' ORDER BY
Date_of_experiment %s,Time_of_experiment %s LIMIT
%d,%d"""%(Pig_ID,vitalsdataorder,vitalsdataorder,lowerlimit,numberofdatapoints))
else:
    cursor.execute("""SELECT
`monr_vitals_manual_10_min`.Date_of_experiment,`monr_vitals_manual_10_min`.Time_of
_experiment,`monr_vitals_manual_10_min`.Heart_Rate,`monr_vitals_manual_10_min`.Art
_Sys,`monr_vitals_manual_10_min`.Art_Dia,`monr_vitals_manual_10_min`.MeanArtPres,`
monr_vitals_manual_10_min`.PA_Sys,`monr_vitals_manual_10_min`.PA_Dia,`monr_vitals_
manual_10_min`.sto2_multidepth,`monr_vitals_manual_10_min`.thi_multidepth,`monr_vi
tals_manual_10_min`.StO2_15mm,`monr_vitals_manual_10_min`.THI_15mm,`monr_vitals_ma
nual_10_min`.FiO2,`monr_vitals_manual_10_min`.Propofol,`monr_vitals_manual_10_min`
.BIS,`monr_vitals_manual_10_min`.Event_name,`monr_vitals_manual_one_hour`.Wedge_pr
essure,`monr_vitals_manual_one_hour`.Urine_output,`monr_vitals_manual_10_min`.Temp
erature,`monr_vitals_manual_one_hour`.Cardiac_output,`monr_vitals_manual_one_hour`
.Arterial_pH,`monr_vitals_manual_one_hour`.Arterial_PC02,`monr_vitals_manual_one_h
our`.Arterial_PO2,`monr_vitals_manual_one_hour`.Arterial_O2_sat,`monr_vitals_manua
l_one_hour`.Venous_PO2,`monr_vitals_manual_one_hour`.Venous_O2_sat,`monr_vitals_ma
nual_one_hour`.Hb,`monr_vitals_manual_one_hour`.DO2,`monr_vitals_manual_one_hour`.
VO2,`monr_vitals_manual_one_hour`.ERO2,`monr_vitals_manual_one_hour`.Lactate,`monr
_vitals_manual_one_hour`.Base_deficit,`monr_vitals_manual_one_hour`.Bladder_pressu
re,`monr_vitals_manual_one_hour`.Chesttube_output,`monr_vitals_manual_10_min`.Even
t_comments,`monr_vitals_manual_one_hour`.Specificgravity_urine,`monr_vitals_manual
_one_hour`.Sodium,`monr_vitals_manual_one_hour`.Potassium,`monr_vitals_manual_one
_hour`.Calcium,`monr_vitals_manual_one_hour`.Glucose FROM
`monr_vitals_manual_10_min` LEFT JOIN `monr_vitals_manual_one_hour` ON
`monr_vitals_manual_10_min`.Date_of_experiment=`monr_vitals_manual_one_hour`.Date_
of_experiment AND
`monr_vitals_manual_10_min`.Time_of_experiment=`monr_vitals_manual_one_hour`.Time_
of_experiment WHERE `monr_vitals_manual_10_min`.Pig_ID='%s' ORDER BY

```

```

Date_of_experiment %s,Time_of_experiment %s LIMIT
%d,%d""""%(Pig_ID,vitalsdataorder,vitalsdataorder,lowerlimit,numberofdatapoints))
    output1 = cursor.fetchall()
    cursor.execute("""SELECT COUNT(*) FROM(SELECT
`monr_vitals_manual_10_min`.Date_of_experiment,`monr_vitals_manual_10_min`.Time_of
_experiment,`monr_vitals_manual_10_min`.Heart_Rate,`monr_vitals_manual_10_min`.Art
_Sys,`monr_vitals_manual_10_min`.Art_Dia,`monr_vitals_manual_10_min`.MeanArtPres,`
monr_vitals_manual_10_min`.PA_Sys,`monr_vitals_manual_10_min`.PA_Dia,`monr_vitals_
manual_10_min`.sto2_multidepth,`monr_vitals_manual_10_min`.thi_multidepth,`monr_vi
tals_manual_10_min`.StO2_15mm,`monr_vitals_manual_10_min`.THI_15mm,`monr_vitals_ma
nual_10_min`.FiO2,`monr_vitals_manual_10_min`.Propofol,`monr_vitals_manual_10_min`.
.BIS,`monr_vitals_manual_10_min`.Event_name,`monr_vitals_manual_one_hour`.Wedge_pr
essure,`monr_vitals_manual_one_hour`.Urine_output,`monr_vitals_manual_10_min`.Temp
erature,`monr_vitals_manual_one_hour`.Cardiac_output,`monr_vitals_manual_one_hour`.
Arterial_pH,`monr_vitals_manual_one_hour`.Arterial_PC02,`monr_vitals_manual_one_h
our`.Arterial_PO2,`monr_vitals_manual_one_hour`.Arterial_O2_sat,`monr_vitals_manua
l_one_hour`.Venous_PO2,`monr_vitals_manual_one_hour`.Venous_O2_sat,`monr_vitals_ma
nual_one_hour`.Hb,`monr_vitals_manual_one_hour`.DO2,`monr_vitals_manual_one_hour`.
VO2,`monr_vitals_manual_one_hour`.ER02,`monr_vitals_manual_one_hour`.Lactate,`monr
_vitals_manual_one_hour`.Base_deficit,`monr_vitals_manual_one_hour`.Bladder_pressu
re,`monr_vitals_manual_one_hour`.Chesttube_output,`monr_vitals_manual_10_min`.Even
t_comments,`monr_vitals_manual_one_hour`.Specificgravity_urine,`monr_vitals_manual
_one_hour`.Sodium,`monr_vitals_manual_one_hour`.Potassium,`monr_vitals_manual_one
_hour`.Calcium,`monr_vitals_manual_one_hour`.Glucose FROM
`monr_vitals_manual_10_min` LEFT JOIN `monr_vitals_manual_one_hour` ON
`monr_vitals_manual_10_min`.Date_of_experiment=`monr_vitals_manual_one_hour`.Date_
of_experiment AND
`monr_vitals_manual_10_min`.Time_of_experiment=`monr_vitals_manual_one_hour`.Time_
of_experiment WHERE `monr_vitals_manual_10_min`.Pig_ID='%s' ORDER BY
Date_of_experiment,Time_of_experiment) AS table_alias""""%(Pig_ID))
    output2 = cursor.fetchall()
    thelength = len(output1)
    if(numberofdatapoints > thelength):
        ending_row = lowerlimit+thelength
    else:
        pass
    vitals_total_data_points = int(output2[0][0])
    connection.commit()
    cursor.close()
    connection.close()
    if (len(output)>0):
        if(len(output1)>0):
            datatable="""<table>"""
            evenRow = "True"
            for i in range(len(output1)):
                Date_of_event = str(output1[i][0]).split('-')
                Day_of_event = Date_of_event[2]
                Month_of_event = Date_of_event[1]
                Year_of_event = Date_of_event[0]
                Event_Date =
Month_of_event+"/"+Day_of_event+"/"+Year_of_event
                Time_of_event=str(output1[i][1])
                Event_name = str(output1[i][15])
                if not (output1[i][2]==None):
                    Heart_Rate = str(output1[i][2])
                else:
                    Heart_Rate = '&nbsp;'
                if not (output1[i][3]==None):

```

```

        Art_Sys = str(output1[i][3])
    else:
        Art_Sys = '&nbsp;';
    if not (output1[i][4]==None):
        Art_Dia = str(output1[i][4])
    else:
        Art_Dia = '&nbsp;';
    if not(output1[i][3]==None) and not
(output1[i][4]==None):
        Art_BP = Art_Sys+"/"+Art_Dia
    else:
        Art_BP = '&nbsp;';
    if not (output1[i][5]==None):
        MeanArtPres = str(output1[i][5])
    else:
        MeanArtPres = '&nbsp;';
    if not (output1[i][6]==None):
        PA_Sys = str(output1[i][6])
    else:
        PA_Sys = '&nbsp;';
    if not (output1[i][7]==None):
        PA_Dia = str(output1[i][7])
    else:
        PA_Dia = '&nbsp;';
    if not (output1[i][6]==None) and not
(output1[i][7]==None):
        PA_Pres = PA_Sys+"/"+PA_Dia
    else:
        PA_Pres = '&nbsp;';
    if not (output1[i][8]==None):
        sto2_multidepth = str(output1[i][8])
    else:
        sto2_multidepth = '&nbsp;';
    if not (output1[i][9]==None):
        thi_multidepth = str(output1[i][9])
    else:
        thi_multidepth = '&nbsp;';
    if not (output1[i][10]==None):
        StO2_15mm = str(output1[i][10])
    else:
        StO2_15mm = '&nbsp;';
    if not (output1[i][11]==None):
        THI_15mm = str(output1[i][11])
    else:
        THI_15mm = '&nbsp;';
    if not (output1[i][12]==None):
        FiO2 = str(output1[i][12])
    else:
        FiO2 = '&nbsp;';
    if not (output1[i][13]==None):
        Propofol = str(output1[i][13])
    else:
        Propofol = '&nbsp;';
    if not (output1[i][14]==None):
        BIS = str(output1[i][14])
    else:
        BIS = '&nbsp;';
    if not (output1[i][16]==None):

```

```

        Wedge_pressure = str(output1[i][16])
else:
    Wedge_pressure = '&nbsp;';
if not (output1[i][17]==None):
    Urine_output = output1[i][17]
else:
    Urine_output = '&nbsp;';
if not (output1[i][18]==None):
    Temperature = str(output1[i][18])
else:
    Temperature = '&nbsp;';
if not (output1[i][19]==None):
    Cardiac_output = str(output1[i][19])
else:
    Cardiac_output = '&nbsp;';
if not (output1[i][20]==None):
    Arterial_pH = str(output1[i][20])
else:
    Arterial_pH = '&nbsp;';
if not (output1[i][21]==None):
    Arterial_PCO2 = str(output1[i][21])
else:
    Arterial_PCO2 = '&nbsp;';
if not (output1[i][22]==None):
    Arterial_PO2 = str(output1[i][22])
else:
    Arterial_PO2 = '&nbsp;';
if not (output1[i][23]==None):
    Arterial_O2_sat = str(output1[i][23])
else:
    Arterial_O2_sat = '&nbsp;';
if not (output1[i][24]==None):
    Venous_PO2 = str(output1[i][24])
else:
    Venous_PO2 = '&nbsp;';
if not (output1[i][25]==None):
    Venous_O2_sat = str(output1[i][25])
else:
    Venous_O2_sat = '&nbsp;';
if not (output1[i][26]==None):
    Hb = str(output1[i][26])
else:
    Hb = '&nbsp;';
if not (output1[i][27]==None):
    D02 = str(output1[i][27])
else:
    D02 = '&nbsp;';
if not (output1[i][28]==None):
    V02 = str(output1[i][28])
else:
    V02 = '&nbsp;';
if not (output1[i][29]==None):
    ERO2 = str(output1[i][29])
else:
    ERO2 = '&nbsp;';
if not (output1[i][30]==None):
    Lactate = str(output1[i][30])
else:

```

```

        Lactate = '&nbsp;';
    if not (output1[i][31]==None):
        Base_deficit = str(output1[i][31])
    else:
        Base_deficit = '&nbsp;';
    if not (output1[i][32]==None):
        Bladder_pressure = str(output1[i][32])
    else:
        Bladder_pressure = '&nbsp;';
    if not (output1[i][33]==None):
        Chesttube_output = str(output1[i][33])
    else:
        Chesttube_output = '&nbsp;';
    if not (output1[i][34]==None):
        Event_comments = str(output1[i][34])
        Event_comments =
Event_comments.replace("\'","\\\'")
        Event_comments = Event_comments.replace("\n","<br
/>")
    else:
        Event_comments = '&nbsp;';
    if not (output1[i][35]==None):
        Specificgravity_urine = str(output1[i][35])
    else:
        Specificgravity_urine = '&nbsp;';
    if not (output1[i][36]==None):
        Sodium = str(output1[i][36])
    else:
        Sodium = '&nbsp;';
    if not (output1[i][37]==None):
        Potassium = str(output1[i][37])
    else:
        Potassium = '&nbsp;';
    if not (output1[i][38]==None):
        Calcium = str(output1[i][38])
    else:
        Calcium = '&nbsp;';
    if not (output1[i][39]==None):
        Glucose = str(output1[i][39])
    else:
        Glucose = '&nbsp;';
    elementID =
Pig_ID+"_"+Vitals+"_"+Event_Date+"_"+Time_of_event+"_"
    Date_ID = elementID+"neweventdate"
    Time_ID = elementID+"neweventtime"
    Event_ID = elementID+"event"
    HR_ID = elementID+"heartrate"
    ArtBP_ID = elementID+"artbp"
    PAPres_ID = elementID+"papres"
    Wedge_ID = elementID+"wedgepressure"
    Urine_ID = elementID+"urineoutput"
    Temp_ID = elementID+"temperature"
    CAop_ID = elementID+"cardiacoutput"
    Sto2multidepth_ID = elementID+"sto2-multidepth"
    Thimultidepth_ID = elementID+"thi-multidepth"
    FiO2_ID = elementID+"fio2"
    Propofol_ID = elementID+"propofol"
    BIS_ID = elementID+"bis"

```



```
class="vitalsColWidth">%s</div></td></tr>""%(Date_ID,Event_Date,Time_ID,Time_of_e
vent,Comments_ID,Event_comments,Event_ID,Event_name,HR_ID,Heart_Rate,ArtBP_ID,Art
BP,MeanArtPres,PAPres_ID,PA_Pres,Wedge_ID,Wedge_pressure,Urine_ID,Urine_output,Bla
dderPres_ID,Bladder_pressure,Temp_ID,Temperature,CAop_ID,Cardiac_output,Propofol_I
D,Propofol,BIS_ID,BIS,FiO2_ID,FiO2,Sto215mm_ID,StO2_15mm,Thi15mm_ID,THI_15mm,Sto2m
ultidepth_ID,sto2_multidepth,Thimultidepth_ID,thi_multidepth,Chest_ID,Chesttube_ou
tput,ArtpH_ID,Arterial_pH,Artpco2_ID,Arterial_PC02,Artpo2_ID,Arterial_PO2,ArtO2sat
_ID,Arterial_O2_sat,Venpo2_ID,Venous_PO2,VenO2sat_ID,Venous_O2_sat,Hb_ID,Hb,D02,VO
2,ERO2,Lactate_ID,Lactate,Basedeficit_ID,Base_deficit,Specificgravity_ID,Specificg
ravity_urine,Sodium_ID,Sodium,Potassium_ID,Potassium,Calcium_ID,Calcium,Glucose_ID
,Glucose)
```

```
evenRow = "False"
```

```
else:
```

```
datatable = datatable + ""<tr
```

```
class="oddRow"><td><div id="%s" class="vitalsDate">%s</div></td><td><div id="%s"
```

```
class="vitalsTime">%s</div></td><td><div id="%s"
```

```
class="vitalsComments">%s</div></td><td><div id="%s"
```

```
class="vitalsEvent">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```
class="vitalsColWidth">%s</div></td><td><div id="%s"
```

```

_ID,Arterial_O2_sat,Venpo2_ID,Venous_PO2,VenO2sat_ID,Venous_O2_sat,Hb_ID,Hb,D02,VO
2,ERO2,Lactate_ID,Lactate,Basedeficit_ID,Base_deficit,Specificgravity_ID,Specific
ravity_urine,Sodium_ID,Sodium,Potassium_ID,Potassium,Calcium_ID,Calcium,Glucose_ID
,Glucose)
        evenRow = "True"
        #datatable = datatable+"""/table><div
id="theend">&nbsp;</div>""
        #<div id="theend">&nbsp;</div>
        print 'Content-Type: text/javascript\n'
        if(action == "viewallvitals"):
            datatable = datatable+"""/table>""
            datatable = datatable.replace('<div id','<div
ondblclick="javascript:editVitalsElement1(this);" id')
            print """/document.getElementById('innerDiv').innerHTML
=
's';vitals_total_data_points=parseInt(%d);pagination();document.getElementById('m
essage2').innerHTML='Showing rows from %d to
%d';""%(datatable,vitals_total_data_points,starting_row,ending_row)
            else:
                datatable = datatable+"""/table>""
                datatable = datatable.replace('<div id','<div
ondblclick="javascript:opener.editVitalsElement2(this);" id')
                print
                """/allVitalsPopup.document.getElementById('innerDiv').innerHTML =
's';vitals_total_data_points=parseInt(%d);pagination();allVitalsPopup.document.ge
tElementById('message2').innerHTML='Showing rows from %d to
%d';""%(datatable,vitals_total_data_points,starting_row,ending_row)
                #window["allVitalsPopup"].location='dataview.py#theend';
            else:
                print 'Content-Type: text/javascript\n'
                if(action == "viewallvitals"):
                    print """/var givealert1 =
document.getElementById('innerDiv');
                    givealert1.innerHTML = '<h2>No vitals data
entered yet</h2>';""
                else:
                    print """/var givealert1 =
allVitalsPopup.document.getElementById('innerDiv');
                    givealert1.innerHTML = '<h2>No vitals data
entered yet</h2>';""
                else:
                    print 'Content-Type: text/javascript\n'
                    print """/var givealert1 = document.getElementById('message1');
                    givealert1.innerHTML = 'The ID you have entered does
not exist in the database. To create a new entry, you must complete ID, Group and
Weight fields';""
                    #except:
                    #print 'Content-Type: text/javascript\n'
                    #print """/var givealert1 =
document.getElementById('message1');
                    #givealert1.innerHTML = 'The ID you have entered does
not exist in the database. To create a new entry, you must complete ID, Group and
Weight fields';""
#####
#####
#####
elif(action=="vitalsformsubmit"):

```



```

try:
    Pig_ID = str(form['pig_id'].value)
    day_of_exp = int(form['day'].value)
    Date_of_experiment = form['date'].value
    Event_name = form['event'].value
    if Event_name=='Other':
        Event_name='99'
    else:
        pass
    Event_name = Event_name.replace(' ','')
    hours = str(form['hours'].value)
    minutes = str(form['minutes'].value)
    Time_of_experiment = (hours+ ":" + minutes + ":" + "00")
    #vitalsdataorder = str(form['vitalsdataorder'].value)
    #if(vitalsdataorder=='Descending'):
        #vitalsdataorder='DESC'
    #else:
        #vitalsdataorder='ASC'
    #numberofdatapoints = int(form['numberofdatapoints'].value)
    connection = sql.connect(host=lhost,port=lport,user=user,
passwd=lpasswd,db=ldb)
    cursor = connection.cursor()
    onehourvitals = int(form['onehourvitals'].value)
    if(onehourvitals > 0):
        cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

    else:
        pass
        cursor.execute("""INSERT INTO
monr_vitals_manual_10_min(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

    #####
    D02 = ''
    V02 = ''
    if (form.has_key('heartrate')):
        Heart_Rate = int(form['heartrate'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Heart_Rate = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Heart_Rate,Pig_ID,Date_of_experiment,Time_of_experiment))

    else:
        pass
    if (form.has_key('artsys')):
        Art_Sys = int(form['artsys'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Art_Sys = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Art_Sys,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
    if (form.has_key('artdia')):
        Art_Dia = int(form['artdia'].value)

```

```

        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Art_Dia = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Art_Dia,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
    if (form.has_key('artsys')) and (form.has_key('artdia')):
        Art_Sys = float(form['artsys'].value)
        Art_Dia = float(form['artdia'].value)
        MeanArtPres = Art_Dia + ((Art_Sys - Art_Dia)/3)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
MeanArtPres = "%.2f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(MeanArtPres,Pig_ID,Date_of_experiment,Time_of_experime
nt))
    else:
        pass
    if (form.has_key('pasys')):
        PA_Sys = int(form['pasys'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
PA_Sys = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(PA_Sys,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
    if (form.has_key('padia')):
        PA_Dia = int(form['padia'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
PA_Dia = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(PA_Dia,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
    if (form.has_key('sto2_multidepth')):
        sto2_multidepth = int(form['sto2_multidepth'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
sto2_multidepth = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(sto2_multidepth,Pig_ID,Date_of_experiment,Time_of_expe
riment))
    else:
        pass
    if (form.has_key('thi_multidepth')):
        thi_multidepth = float(form['thi_multidepth'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
thi_multidepth = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(thi_multidepth,Pig_ID,Date_of_experiment,Time_of_exper
iment))
    else:
        pass
    if (form.has_key('sto2_15mm')):
        StO2_15mm = int(form['sto2_15mm'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
StO2_15mm = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(StO2_15mm,Pig_ID,Date_of_experiment,Time_of_experiment
))
    else:
        pass
    if (form.has_key('thi_15mm')):
        THI_15mm = float(form['thi_15mm'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
THI_15mm = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND

```

```

Time_of_experiment='%s'""%(THI_15mm,Pig_ID,Date_of_experiment,Time_of_experiment)
)
    else:
        pass
    if (form.has_key('fio2')):
        Fio2 = float(form['fio2'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Fio2 = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Fio2,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
    if (form.has_key('propofol')):
        Propofol = int(form['propofol'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Propofol = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Propofol,Pig_ID,Date_of_experiment,Time_of_experiment)
)
    else:
        pass
    if (form.has_key('bis')):
        BIS = int(form['bis'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET BIS
= "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(BIS,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
    if (form.has_key('wedgepressure')):
        Wedge_pressure = int(form['wedgepressure'].value)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Wedge_pressure = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Wedge_pressure,Pig_ID,Date_of_experiment,Time_of_exper
iment))
    else:
        pass
    if (form.has_key('urineoutput')):
        Urine_output = int(form['urineoutput'].value)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Urine_output = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Urine_output,Pig_ID,Date_of_experiment,Time_of_experim
ent))
    else:
        pass
    if (form.has_key('temperature')):
        Temperature = float(form['temperature'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Temperature = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Temperature,Pig_ID,Date_of_experiment,Time_of_experime
nt))
    else:
        pass
    if (form.has_key('cardiacoutput')):
        Cardiac_output = float(form['cardiacoutput'].value)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Cardiac_output = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Cardiac_output,Pig_ID,Date_of_experiment,Time_of_exper
iment))
    else:
        pass

```

```

        if (form.has_key('arterialph')):
            Arterial_pH = float(form['arterialph'].value)
            cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Arterial_pH = "%.2f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Arterial_pH,Pig_ID,Date_of_experiment,Time_of_experime
nt))
        else:
            pass
        if (form.has_key('art_pco2')):
            Arterial_PCO2 = int(form['art_pco2'].value)
            cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Arterial_PCO2 = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Arterial_PCO2,Pig_ID,Date_of_experiment,Time_of_experi
ment))
        else:
            pass
        if (form.has_key('art_po2')):
            Arterial_PO2 = int(form['art_po2'].value)
            cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Arterial_PO2 = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Arterial_PO2,Pig_ID,Date_of_experiment,Time_of_experim
ent))
        else:
            pass
        if (form.has_key('art_o2_sat')):
            Arterial_O2_sat = int(form['art_o2_sat'].value)
            cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Arterial_O2_sat = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Arterial_O2_sat,Pig_ID,Date_of_experiment,Time_of_expe
riment))
        else:
            pass
        if (form.has_key('ven_po2')):
            Venous_PO2 = int(form['ven_po2'].value)
            cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Venous_PO2 = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Venous_PO2,Pig_ID,Date_of_experiment,Time_of_experimen
t))
        else:
            pass
        if (form.has_key('ven_o2_sat')):
            Venous_O2_sat = int(form['ven_o2_sat'].value)
            cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Venous_O2_sat = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Venous_O2_sat,Pig_ID,Date_of_experiment,Time_of_experi
ment))
        else:
            pass
        if (form.has_key('hemoglobin')):
            Hb = float(form['hemoglobin'].value)
            cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Hb = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Hb,Pig_ID,Date_of_experiment,Time_of_experiment))
        else:
            pass
        if (form.has_key('cardiacoutput')) and
(form.has_key('hemoglobin')) and (form.has_key('art_o2_sat')):
            Cardiac_output = float(form['cardiacoutput'].value)

```

```

        Hb = float(form['hemoglobin'].value)
        Arterial_O2_sat = float(form['art_o2_sat'].value)
        DO2 = (Cardiac_output*Hb*13.4*Arterial_O2_sat)/(1940)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
D02 = "%.3f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(D02,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
        if (form.has_key('cardiacoutput')) and
(form.has_key('hemoglobin')) and (form.has_key('art_o2_sat')) and
(form.has_key('ven_o2_sat')):
            Cardiac_output = float(form['cardiacoutput'].value)
            Hb = float(form['hemoglobin'].value)
            Arterial_O2_sat = float(form['art_o2_sat'].value)
            Venous_O2_sat = float(form['ven_o2_sat'].value)
            V02 = ((Cardiac_output*Hb*13.4)*(Arterial_O2_sat-
Venous_O2_sat))/(1940)
            ERO2 = (V02*100)/(D02)
            cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
V02 = "%.3f",ERO2 = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(V02,ERO2,Pig_ID,Date_of_experiment,Time_of_experiment)
)
        else:
            pass
            if (form.has_key('lactate')):
                Lactate = float(form['lactate'].value)
                cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Lactate = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Lactate,Pig_ID,Date_of_experiment,Time_of_experiment))
            else:
                pass
                if (form.has_key('basedeficit')):
                    Base_deficit = float(form['basedeficit'].value)
                    cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Base_deficit = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Base_deficit,Pig_ID,Date_of_experiment,Time_of_experim
ent))
                else:
                    pass
                    if (form.has_key('bladderpres')):
                        Bladder_pressure = float(form['bladderpres'].value)
                        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Bladder_pressure = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Bladder_pressure,Pig_ID,Date_of_experiment,Time_of_exp
eriment))
                    else:
                        pass
                        if (form.has_key('chesttubeoutput')):
                            Chesttube_output = float(form['chesttubeoutput'].value)
                            cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Chesttube_output = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Chesttube_output,Pig_ID,Date_of_experiment,Time_of_exp
eriment))
                    else:
                        pass
                        if (form.has_key('eventcomments')):
                            Event_comments = form['eventcomments'].value

```

```

        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Event_comments = "%s" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(sql.escape_string(Event_comments),Pig_ID,Date_of_exper
iment,Time_of_experiment))
    else:
        pass
    if (form.has_key('specificgravity')):
        Specificgravity_urine =
float(form['specificgravity'].value)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Specificgravity_urine = "%.3f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Specificgravity_urine,Pig_ID,Date_of_experiment,Time_o
f_experiment))
    else:
        pass
    if (form.has_key('sodium')):
        Sodium = int(form['sodium'].value)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Sodium = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Sodium,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
    if (form.has_key('potassium')):
        Potassium = float(form['potassium'].value)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Potassium = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Potassium,Pig_ID,Date_of_experiment,Time_of_experiment
))
    else:
        pass
    if (form.has_key('calcium')):
        Calcium = float(form['calcium'].value)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Calcium = "%.2f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Calcium,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
    if (form.has_key('glucose')):
        Glucose = int(form['glucose'].value)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour SET
Glucose = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Glucose,Pig_ID,Date_of_experiment,Time_of_experiment))
    else:
        pass
    connection.commit()
    cursor.close()
    connection.close()
    print 'Content-Type: text/javascript\n'
    print """var givealert2 = document.getElementById('message2');
var allvitalsdisplay =
document.getElementById('vitalsdataholder').style;
vitalsRefresh();
if(window['allVitalsPopup'] &&
!window['allVitalsPopup'].closed){
    requestAllVitalsNewWindow();
}
else if(allvitalsdisplay.display=='inline'){
    showAllVitals();
}

```

```

        }
        else{};
        givealert2.innerHTML = 'Vitals successfully
updated';""

#####
#####
except:
    try:
        connection.rollback()
    except:
        pass
        #raise
    print 'Content-Type: text/javascript\n'
    print ""var givealert2 = document.getElementById('message2');
        givealert2.innerHTML = 'There was an error in the
data entry. Please check your data and submit again';""
    #raise

#####
#####
elif(action=="editvitals"):
    try:
        elementID = form['elementid'].value
        Pig_ID = str(form['pig_id'].value)
        Date_of_experiment = form['exp_date'].value
        Time_of_experiment = form['exp_time'].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
        cursor = connection.cursor()
        Event_name = ''
        Vitalsoneshour = 0
        cursor.execute("""SELECT * FROM monr_vitals_manual_one_hour
WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        checkVitalsoneshour = cursor.fetchall()
        if (len(checkVitalsoneshour)>0):
            Vitalsoneshour = 1
        else:
            cursor.execute("""SELECT Event_name FROM
monr_vitals_manual_10_min WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
            has_event = cursor.fetchall()
            Event_name = str(has_event[0][0])
            Event_name = Event_name.replace(' ','')
#####
            D02 = ''
            V02 = ''
            ER02 = ''
            if (form.has_key('neweventdate')):
                if not (form['neweventdate'].value=="nullify"):
                    New_Date_of_experiment = form['neweventdate'].value
                    newdatesplit = New_Date_of_experiment.split('/')
                    newyear = newdatesplit[2]
                    newmonth = newdatesplit[0]
                    newday = newdatesplit[1]
                    newdatesplit = newyear+"-"+newmonth+"-"+newday
                    if(Vitalsoneshour == 1):

```

```

        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Date_of_experiment = "%s" WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(newdatesplit,Pig_ID,Date_of_experiment,Time_of_experim
ent))

        cursor.execute("""UPDATE monr_vitals_manual_10_min
SET Date_of_experiment = "%s" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(newdatesplit,Pig_ID,Date_of_experiment,Time_of_experim
ent))

    else:
        cursor.execute("""UPDATE monr_vitals_manual_10_min
SET Date_of_experiment = "%s" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(newdatesplit,Pig_ID,Date_of_experiment,Time_of_experim
ent))

        print 'Content-Type: text/javascript\n'
        print """"requestAllVitalsWindowIndependent();""""
    else:
        print 'Content-Type: text/javascript\n'
        print """"vitalsEditFailure("%s");""""%(elementID)
else:
    pass
if (form.has_key('neweventtime')):
    if not (form['neweventtime'].value=="nullify"):
        New_Time_of_experiment =
str(form['neweventtime'].value)
        if(Vitalsonehour == 1):
            cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Time_of_experiment = "%s" WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(New_Time_of_experiment,Pig_ID,Date_of_experiment,Time_
of_experiment))

            cursor.execute("""UPDATE monr_vitals_manual_10_min
SET Time_of_experiment = "%s" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(New_Time_of_experiment,Pig_ID,Date_of_experiment,Time_
of_experiment))

        else:
            cursor.execute("""UPDATE monr_vitals_manual_10_min
SET Time_of_experiment = "%s" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(New_Time_of_experiment,Pig_ID,Date_of_experiment,Time_
of_experiment))

            print 'Content-Type: text/javascript\n'
            print """"requestAllVitalsWindowIndependent();""""
        else:
            print 'Content-Type: text/javascript\n'
            print """"vitalsEditFailure("%s");""""%(elementID)
else:
    pass
if (form.has_key('heartrate')):
    if not (form['heartrate'].value=="nullify"):
        Heart_Rate = int(form['heartrate'].value)
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Heart_Rate = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Heart_Rate,Pig_ID,Date_of_experiment,Time_of_experimen
t))

        print 'Content-Type: text/javascript\n'
        print """"vitalsEditSuccess("%s");""""%(elementID)
    else:

```



```

        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Heart_Rate = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print """"vitalsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('artbp')):
        if not (form['artbp'].value=="nullify"):
            Art_BP = form['artbp'].value
            split_Art_BP = Art_BP.split('/')
            Art_Sys = int(split_Art_BP[0])
            Art_Dia = int(split_Art_BP[1])
            cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Art_Sys = "%d",Art_Dia = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Art_Sys,Art_Dia,Pig_ID,Date_of_experiment,Time_of_expe
riment))

            Art_Sys = float(Art_Sys)
            Art_Dia = float(Art_Dia)
            MeanArtPres = Art_Dia + ((Art_Sys - Art_Dia)/3)
            cursor.execute("""UPDATE monr_vitals_manual_10_min SET
MeanArtPres = "%.2f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(MeanArtPres,Pig_ID,Date_of_experiment,Time_of_experime
nt))

            print 'Content-Type: text/javascript\n'
            print """"var allvitalsdisplay =
document.getElementById('vitalsdataholder').style;
                if(window['allVitalsPopup'] &&
!window['allVitalsPopup'].closed){
                    var givealert2 =
allVitalsPopup.document.getElementById('message2');
                    givealert2.innerHTML = 'Vitals
successfully updated';

                    vitalsEditSuccess("%s");
                    gettoElement =
allVitalsPopup.document.getElementById("%s").parentNode;
                    getnextElement =
gettoElement.nextSibling.firstChild;
                    getnextElement.innerHTML = "%.2f";
                }
                else if(allvitalsdisplay.display=='inline'){
                    var givealert2 =
document.getElementById('message2');
                    givealert2.innerHTML = 'Vitals
successfully updated';

                    vitalsEditSuccess("%s");
                    gettoElement =
document.getElementById("%s").parentNode;
                    getnextElement =
gettoElement.nextSibling.firstChild;
                    getnextElement.innerHTML = "%.2f";
                }

            else{""%(elementID,elementID,MeanArtPres,elementID,elementID,MeanArtPres)
                else:
                    cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Art_Sys = NULL,Art_Dia = NULL,MeanArtPres = NULL WHERE Pig_ID='%s' AND

```

```

Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
    print 'Content-Type: text/javascript\n'
    print ""var allvitalsdisplay =
document.getElementById('vitalsdataholder').style;
    if(window['allVitalsPopup'] &&
!window['allVitalsPopup'].closed){
    var givealert2 =
allVitalsPopup.document.getElementById('message2');
    givealert2.innerHTML = 'Vitals
successfully updated';
    vitalsEmptyContent2("%s");
    gettoElement =
allVitalsPopup.document.getElementById("%s").parentNode;
    getnextElement =
gettoElement.nextSibling.firstChild;
    getnextElement.innerHTML = "";
    }
    else if(allvitalsdisplay.display=='inline'){
    var givealert2 =
document.getElementById('message2');
    givealert2.innerHTML = 'Vitals
successfully updated';
    vitalsEmptyContent1("%s");
    gettoElement =
document.getElementById("%s").parentNode;
    getnextElement =
gettoElement.nextSibling.firstChild;
    getnextElement.innerHTML = "";
    }
}""%(elementID,elementID,elementID,elementID)
    else:
    pass
    if (form.has_key('papres')):
    if not (form['papres'].value=="nullify"):
    PA_Pres = form['papres'].value
    split_PA_Pres = PA_Pres.split('/')
    PA_Sys = int(split_PA_Pres[0])
    PA_Dia = int(split_PA_Pres[1])
    cursor.execute("""UPDATE monr_vitals_manual_10_min SET
PA_Sys = "%d",PA_Dia = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(PA_Sys,PA_Dia,Pig_ID,Date_of_experiment,Time_of_experi
ment))
    print 'Content-Type: text/javascript\n'
    print ""vitalsEditSuccess("%s");""%(elementID)
    else:
    cursor.execute("""UPDATE monr_vitals_manual_10_min SET
PA_Sys = NULL,PA_Dia = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
    print 'Content-Type: text/javascript\n'
    print ""vitalsEmptyContent("%s");""%(elementID)
    else:
    pass
    if (form.has_key('sto2-multidepth')):
    if not (form['sto2-multidepth'].value=="nullify"):
    StO2_multidepth = int(form['sto2-multidepth'].value)

```

```

        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
StO2_multidepth = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(StO2_multidepth,Pig_ID,Date_of_experiment,Time_of_expe
riment))

        print 'Content-Type: text/javascript\n'
        print """"vitalsEditSuccess("%s");""%(elementID)
    else:
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
StO2_multidepth = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print """"vitalsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('event')):
        Event_name = form['event'].value
        if Event_name=='Other':
            Event_name='99'
        else:
            pass
        Event_name = Event_name.replace(' ','')
        if not (Event_name=="nullify"):
            cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Event_name = "%s" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Event_name,Pig_ID,Date_of_experiment,Time_of_experimen
t))

            if (Vitalsonehour == 1):
                cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Event_name = "%s" WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Event_name,Pig_ID,Date_of_experiment,Time_of_experimen
t))

            else:
                pass
            print 'Content-Type: text/javascript\n'
            print """"vitalsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"vitalsEditFailure("%s");""%(elementID)
    else:
        pass
    if (form.has_key('thi-multidepth')):
        if not (form['thi-multidepth'].value=="nullify"):
            THI_multidepth = float(form['thi-multidepth'].value)
            cursor.execute("""UPDATE monr_vitals_manual_10_min SET
THI_multidepth = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(THI_multidepth,Pig_ID,Date_of_experiment,Time_of_exper
iment))

            print 'Content-Type: text/javascript\n'
            print """"vitalsEditSuccess("%s");""%(elementID)
        else:
            cursor.execute("""UPDATE monr_vitals_manual_10_min SET
THI_multidepth = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
            print 'Content-Type: text/javascript\n'
            print """"vitalsEmptyContent("%s");""%(elementID)
    else:
        pass

```

```

        if (form.has_key('sto2-15mm')):
            if not (form['sto2-15mm'].value=="nullify"):
                StO2_15mm = int(form['sto2-15mm'].value)
                cursor.execute("""UPDATE monr_vitals_manual_10_min SET
StO2_15mm = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(StO2_15mm,Pig_ID,Date_of_experiment,Time_of_experiment
))

                print 'Content-Type: text/javascript\n'
                print """"vitalsEditSuccess("%s");""%(elementID)
            else:
                cursor.execute("""UPDATE monr_vitals_manual_10_min SET
StO2_15mm = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                print 'Content-Type: text/javascript\n'
                print """"vitalsEmptyContent("%s");""%(elementID)
        else:
            pass
        if (form.has_key('thi-15mm')):
            if not (form['thi-15mm'].value=="nullify"):
                THI_15mm = float(form['thi-15mm'].value)
                cursor.execute("""UPDATE monr_vitals_manual_10_min SET
THI_15mm = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(THI_15mm,Pig_ID,Date_of_experiment,Time_of_experiment)
)

                print 'Content-Type: text/javascript\n'
                print """"vitalsEditSuccess("%s");""%(elementID)
            else:
                cursor.execute("""UPDATE monr_vitals_manual_10_min SET
THI_15mm = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                print 'Content-Type: text/javascript\n'
                print """"vitalsEmptyContent("%s");""%(elementID)
        else:
            pass
        if (form.has_key('fio2')):
            if not (form['fio2'].value=="nullify"):
                FiO2 = float(form['fio2'].value)
                cursor.execute("""UPDATE monr_vitals_manual_10_min SET
FiO2 = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(FiO2,Pig_ID,Date_of_experiment,Time_of_experiment))
                print 'Content-Type: text/javascript\n'
                print """"vitalsEditSuccess("%s");""%(elementID)
            else:
                cursor.execute("""UPDATE monr_vitals_manual_10_min SET
FiO2 = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                print 'Content-Type: text/javascript\n'
                print """"vitalsEmptyContent("%s");""%(elementID)
        else:
            pass
        if (form.has_key('propofol')):
            if not (form['propofol'].value=="nullify"):
                Propofol = int(form['propofol'].value)
                cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Propofol = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Propofol,Pig_ID,Date_of_experiment,Time_of_experiment)
)

                print 'Content-Type: text/javascript\n'

```

```

        print ""vitalsEditSuccess("%s");""%(elementID)
    else:
        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Propofol = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print ""vitalsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('bis')):
        if not (form['bis'].value=="nullify"):
            BIS = int(form['bis'].value)
            cursor.execute("""UPDATE monr_vitals_manual_10_min SET
BIS = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(BIS,Pig_ID,Date_of_experiment,Time_of_experiment))
            print 'Content-Type: text/javascript\n'
            print ""vitalsEditSuccess("%s");""%(elementID)
        else:
            cursor.execute("""UPDATE monr_vitals_manual_10_min SET
BIS = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
            print 'Content-Type: text/javascript\n'
            print ""vitalsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('wedgepressure')):
        if not (form['wedgepressure'].value=="nullify"):
            if(Vitalsonehour==0):
                cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_nam
e) VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
            else:
                pass
                Wedge_pressure = int(form['wedgepressure'].value)
                cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Wedge_pressure = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Wedge_pressure,Pig_ID,Date_of_experiment,Time_of_exper
iment))
                print 'Content-Type: text/javascript\n'
                print ""vitalsEditSuccess("%s");""%(elementID)
            else:
                if(Vitalsonehour==1):
                    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Wedge_pressure = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                    print 'Content-Type: text/javascript\n'
                    print ""vitalsEmptyContent("%s");""%(elementID)
                else:
                    print 'Content-Type: text/javascript\n'
                    print ""vitalsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('urineoutput')):
        if not (form['urineoutput'].value=="nullify"):
            if(Vitalsonehour==0):

```

```

        cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

        else:
            pass
            Urine_output = int(form['urineoutput'].value)
            cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Urine_output = %d WHERE Pig_ID=%s' AND Date_of_experiment=%s' AND
Time_of_experiment=%s'""%(Urine_output,Pig_ID,Date_of_experiment,Time_of_experiment))

            print 'Content-Type: text/javascript\n'
            print ""vitalsEditSuccess("%s");""%(elementID)
        else:
            if(Vitalsonehour==1):
                cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Urine_output = NULL WHERE Pig_ID=%s' AND
Date_of_experiment=%s' AND
Time_of_experiment=%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                print 'Content-Type: text/javascript\n'
                print ""vitalsEmptyContent("%s");""%(elementID)
            else:
                print 'Content-Type: text/javascript\n'
                print ""vitalsEmptyContent("%s");""%(elementID)
    else:
        pass
        if (form.has_key('temperature')):
            if not (form['temperature'].value=="nullify"):
                #if(Vitalsonehour==0):
                    cursor.execute("""INSERT IGNORE INTO
monr_vitals_manual_10_min(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

                    #else:
                        #pass
                        Temperature = float(form['temperature'].value)
                        cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Temperature = %.1f" WHERE Pig_ID=%s' AND Date_of_experiment=%s' AND
Time_of_experiment=%s'""%(Temperature,Pig_ID,Date_of_experiment,Time_of_experiment))

                        print 'Content-Type: text/javascript\n'
                        print ""vitalsEditSuccess("%s");""%(elementID)
                    else:
                        #if(Vitalsonehour==1):
                            cursor.execute("""INSERT IGNORE INTO
monr_vitals_manual_10_min(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

                            cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Temperature = NULL WHERE Pig_ID=%s' AND Date_of_experiment=%s' AND
Time_of_experiment=%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                            print 'Content-Type: text/javascript\n'
                            print ""vitalsEmptyContent("%s");""%(elementID)
                        #else:
                            #print 'Content-Type: text/javascript\n'

```

```

                                #print ""vitalsEmptyContent("%s");""%(elementID)
else:
    pass
    if (form.has_key('arterialph')):
        if not (form['arterialph'].value=="nullify"):
            if(Vitalsonehour==0):
                cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
                else:
                    pass
                    Arterial_pH = float(form['arterialph'].value)
                    cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Arterial_pH = "%.2f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Arterial_pH,Pig_ID,Date_of_experiment,Time_of_experiment))
                    print 'Content-Type: text/javascript\n'
                    print ""vitalsEditSuccess("%s");""%(elementID)
            else:
                if(Vitalsonehour==1):
                    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Arterial_pH = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                    print 'Content-Type: text/javascript\n'
                    print ""vitalsEmptyContent("%s");""%(elementID)
                else:
                    print 'Content-Type: text/javascript\n'
                    print ""vitalsEmptyContent("%s");""%(elementID)
        else:
            pass
            if (form.has_key('art-pco2')):
                if not (form['art-pco2'].value=="nullify"):
                    if(Vitalsonehour==0):
                        cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
                        else:
                            pass
                            Arterial_PC02 = int(form['art-pco2'].value)
                            cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Arterial_PC02 = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Arterial_PC02,Pig_ID,Date_of_experiment,Time_of_experiment))
                            print 'Content-Type: text/javascript\n'
                            print ""vitalsEditSuccess("%s");""%(elementID)
                    else:
                        if(Vitalsonehour==1):
                            cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Arterial_PC02 = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                            print 'Content-Type: text/javascript\n'
                            print ""vitalsEmptyContent("%s");""%(elementID)

```

```

else:
    print 'Content-Type: text/javascript\n'
    print ""vitalsEmptyContent("%s");""%(elementID)
else:
    pass
if (form.has_key('art-po2')):
    if not (form['art-po2'].value=="nullify"):
        if(Vitalsonehour==0):
            cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
else:
    pass
    Arterial_PO2 = int(form['art-po2'].value)
    cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Arterial_PO2 = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Arterial_PO2,Pig_ID,Date_of_experiment,Time_of_experiment))
    print 'Content-Type: text/javascript\n'
    print ""vitalsEditSuccess("%s");""%(elementID)
else:
    if(Vitalsonehour==1):
        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Arterial_PO2 = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print ""vitalsEmptyContent("%s");""%(elementID)
    else:
        print 'Content-Type: text/javascript\n'
        print ""vitalsEmptyContent("%s");""%(elementID)
else:
    pass
if (form.has_key('ven-po2')):
    if not (form['ven-po2'].value=="nullify"):
        if(Vitalsonehour==0):
            cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
else:
    pass
    Venous_PO2 = int(form['ven-po2'].value)
    cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Venous_PO2 = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Venous_PO2,Pig_ID,Date_of_experiment,Time_of_experiment))
    print 'Content-Type: text/javascript\n'
    print ""vitalsEditSuccess("%s");""%(elementID)
else:
    if(Vitalsonehour==1):
        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Venous_PO2 = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))

```



```

        print 'Content-Type: text/javascript\n'
        print ""vitalsEmptyContent("%s");""%(elementID)
    else:
        print 'Content-Type: text/javascript\n'
        print ""vitalsEmptyContent("%s");""%(elementID)
    else:
        pass
        if (form.has_key('cardiacoutput')) or
(form.has_key('hemoglobin')) or (form.has_key('art-o2-sat')) or
(form.has_key('ven-o2-sat')):
            D02=''
            V02=''
            ER02=''
            if (form.has_key('ven-o2-sat')):
                if not (form['ven-o2-sat'].value=="nullify"):
                    if(Vitalsonehour==0):
                        cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_nam
e) VALUES
("%s", "%s", "%s", "%s)""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
                    else:
                        pass
                        Venous_O2_sat = int(form['ven-o2-sat'].value)
                        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Venous_O2_sat = "%d" WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Venous_O2_sat,Pig_ID,Date_of_experiment,Time_of_experi
ment))
                        Venous_O2_sat = float(form['ven-o2-sat'].value)
                        cursor.execute("""SELECT
Hb,Cardiac_output,Arterial_O2_sat FROM monr_vitals_manual_one_hour WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                        thevalues = cursor.fetchall()
                        if (len(thevalues)>0):
                            if(len(thevalues[0])>2):
                                if not (thevalues[0][0]==None) and not
(thevalues[0][1]==None) and not (thevalues[0][2]==None):
                                    Hb = float(thevalues[0][0])
                                    Cardiac_output =
float(thevalues[0][1])
                                    Arterial_O2_sat =
float(thevalues[0][2])
                                    D02 =
(Cardiac_output*Hb*13.4*Arterial_O2_sat)/(1940)
                                    V02 =
((Cardiac_output*Hb*13.4)*(Arterial_O2_sat-Venous_O2_sat))/(1940)
                                    ER02 = (V02*100)/(D02)
                                    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = "%.3f",V02 = "%.3f",ER02 = "%.1f" WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(D02,V02,ER02,Pig_ID,Date_of_experiment,Time_of_experim
ent))
                                print 'Content-Type:
text/javascript\n'
                                print
""displayAllD02V02ER02("%s", "%.3f", "%.3f", "%.3f");""%(elementID,D02,V02,ER02)

```

```

else:
    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
    print 'Content-Type:
text/javascript\n'
    print
    """displayNoneD02V02ER02("%s");""%(elementID)
    else:
        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print
    """displayNoneD02V02ER02("%s");""%(elementID)
    else:
        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print
    """displayNoneD02V02ER02("%s");""%(elementID)
    else:
        if(Vitalsonehour==1):
            cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Venous_O2_sat = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
            cursor.execute("""SELECT
Hb,Cardiac_output,Arterial_O2_sat FROM monr_vitals_manual_one_hour WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
            thevalues = cursor.fetchall()
            if(len(thevalues)>0):
                if(len(thevalues[0])>2):
                    if not (thevalues[0][0]==None) and not
(thevalues[0][1]==None) and not (thevalues[0][2]==None):
                        Hb = float(thevalues[0][0])
                        Cardiac_output =
float(thevalues[0][1])
                        Arterial_O2_sat =
float(thevalues[0][2])
                        D02 =
(Cardiac_output*Hb*13.4*Arterial_O2_sat)/(1940)
                        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = "%.3f",V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(D02,Pig_ID,Date_of_experiment,Time_of_experiment))
                        print 'Content-Type:
text/javascript\n'
                        print
                    """displayOneD02("%s", "%.3f");""%(elementID,D02)
                    else:
                        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ERO2 = NULL WHERE

```

```

Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type:
text/javascript\n'
        print
        ""displayNoneD02VO2ER02("%s");""%(elementID)
        else:
            cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,VO2 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type:
text/javascript\n'
        print
        ""displayNoneD02VO2ER02("%s");""%(elementID)
        else:
            cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,VO2 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print
        ""displayNoneD02VO2ER02("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print
            ""vitalsEmptyContent("%s");""%(elementID)
            elif (form.has_key('cardiacoutput')):
                if not (form['cardiacoutput'].value=="nullify"):
                    if(Vitalsonehour==0):
                        cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_nam
e) VALUES
("%s", "%s", "%s", "%s)""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
                    else:
                        pass
                        Cardiac_output =
float(form['cardiacoutput'].value)
                        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Cardiac_output = "%.1f" WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Cardiac_output,Pig_ID,Date_of_experiment,Time_of_exper
iment))
                        cursor.execute("""SELECT
Hb,Arterial_O2_sat,Venous_O2_sat FROM monr_vitals_manual_one_hour WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                        thevalues = cursor.fetchall()
                        if(len(thevalues)>0):
                            if(len(thevalues[0])>2):
                                if not (thevalues[0][0]==None) and not
(thevalues[0][1]==None):
                                    Hb = float(thevalues[0][0])
                                    Arterial_O2_sat =
float(thevalues[0][1])
                                    D02 =
(Cardiac_output*Hb*13.4*Arterial_O2_sat)/(1940)

```

```

float(thevalues[0][2])
if not (thevalues[0][2]==None):
    Venous_O2_sat =
    V02 =
    ((Cardiac_output*Hb*13.4)*(Arterial_O2_sat-Venous_O2_sat))/(1940)
    ERO2 = (V02*100)/(D02)
else:
    V02="nullify"
    ERO2="nullify"
else:
    D02 = "nullify"
if (D02=="nullify"):
    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type:
text/javascript\n'
print
"""displayNoneD02V02ER02("%s");""%(elementID)
elif (V02=="nullify"):
    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = "%.3F",V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(D02,Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type:
text/javascript\n'
print
"""displayOneD02("%s", "%.3f");""%(elementID,D02)
else:
    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = "%.3F",V02 = "%.3F",ERO2 = "%.3F" WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(D02,V02,ERO2,Pig_ID,Date_of_experiment,Time_of_experim
ent))
print 'Content-Type:
text/javascript\n'
print
"""displayAllD02V02ER02("%s", "%.3f", "%.3f", "%.3f");""%(elementID,D02,V02,ER02)
else:
    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type: text/javascript\n'
print
"""displayNoneD02V02ER02("%s");""%(elementID)
else:
    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type: text/javascript\n'
print
"""displayNoneD02V02ER02("%s");""%(elementID)
else:
    if(Vitalsonehour==1):

```

```

        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Cardiac_output = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print
        """displayNoneD02V02ERO2("%s");"""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print
            """vitalsEmptyContent("%s");"""%(elementID)
            elif (form.has_key('hemoglobin')):
                if not (form['hemoglobin'].value=="nullify"):
                    if(Vitalsonehour==0):
                        cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_nam
e) VALUES
("%s", "%s", "%s", "%s)"""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
                        else:
                            pass
                            Hb = float(form['hemoglobin'].value)
                            cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Hb = "%.1f" WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Hb,Pig_ID,Date_of_experiment,Time_of_experiment))
                            cursor.execute("""SELECT
Cardiac_output,Arterial_O2_sat,Venous_O2_sat FROM monr_vitals_manual_one_hour
WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                            thevalues = cursor.fetchall()
                            if(len(thevalues)>0):
                                if(len(thevalues[0])>2):
                                    if not (thevalues[0][0]==None) and not
(thevalues[0][1]==None):
                                        Cardiac_output =
float(thevalues[0][0])
                                        Arterial_O2_sat =
float(thevalues[0][1])
                                        D02 =
(Cardiac_output*Hb*13.4*Arterial_O2_sat)/(1940)
                                        if not (thevalues[0][2]==None):
                                            Venous_O2_sat =
float(thevalues[0][2])
                                        V02 =
((Cardiac_output*Hb*13.4)*(Arterial_O2_sat-Venous_O2_sat))/(1940)
                                        ERO2 = (V02*100)/(D02)
                                    else:
                                        V02="nullify"
                                        ERO2="nullify"
                                else:
                                    D02 = "nullify"
                                    if (D02=="nullify"):

```

```

cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type:
text/javascript\n'
print
"""displayNoneD02V02ER02("%s");""%(elementID)
elif (V02=="nullify"):
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = "%.3f",V02 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(D02,Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type:
text/javascript\n'
print
"""displayOneD02("%s", "%.3f");""%(elementID,D02)
else:
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = "%.3f",V02 = "%.3f",ER02 = "%.3f" WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(D02,V02,ER02,Pig_ID,Date_of_experiment,Time_of_experim
ent))
print 'Content-Type:
text/javascript\n'
print
"""displayAllD02V02ER02("%s", "%.3f", "%.3f", "%.3f");""%(elementID,D02,V02,ER02)
else:
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type: text/javascript\n'
print
"""displayNoneD02V02ER02("%s");""%(elementID)
else:
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type: text/javascript\n'
print
"""displayNoneD02V02ER02("%s");""%(elementID)
else:
if(Vitalsonehour==1):
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Hb = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type: text/javascript\n'
print
"""displayNoneD02V02ER02("%s");""%(elementID)
else:
print 'Content-Type: text/javascript\n'

```

```

        print
        """vitalsEmptyContent("%s");"""%(elementID)
        elif (form.has_key('art-o2-sat')):
            if not (form['art-o2-sat'].value=="nullify"):
                if(Vitalsonehour==0):
                    cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
VALUES
("%s","%s","%s","%s")"""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
                else:
                    pass
                    Arterial_O2_sat = int(form['art-o2-sat'].value)
                    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Arterial_O2_sat = "%d" WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Arterial_O2_sat,Pig_ID,Date_of_experiment,Time_of_experiment))
                    Arterial_O2_sat = float(form['art-o2-sat'].value)
                    cursor.execute("""SELECT
Cardiac_output,Hb,Venous_O2_sat FROM monr_vitals_manual_one_hour WHERE Pig_ID='%s'
AND Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                    thevalues = cursor.fetchall()
                    if(len(thevalues)>0):
                        if(len(thevalues[0])>2):
                            if not (thevalues[0][0]==None) and not
(thevalues[0][1]==None):
                                Cardiac_output =
float(thevalues[0][0])
                                Hb = float(thevalues[0][1])
                                D02 =
(Cardiac_output*Hb*13.4*Arterial_O2_sat)/(1940)
                                if not (thevalues[0][2]==None):
                                    Venous_O2_sat =
float(thevalues[0][2])
                                    V02 =
((Cardiac_output*Hb*13.4)*(Arterial_O2_sat-Venous_O2_sat))/(1940)
                                    ERO2 = (V02*100)/(D02)
                                else:
                                    V02="nullify"
                                    ERO2="nullify"
                                else:
                                    D02 = "nullify"
                                if (D02=="nullify"):
                                    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                                    print 'Content-Type:
text/javascript\n'
                                    print
                                    """displayNoneD02V02ERO2("%s");"""%(elementID)
                                    elif (V02=="nullify"):
                                        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = "%.3F",V02 = NULL,ERO2 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(D02,Pig_ID,Date_of_experiment,Time_of_experiment))

```

```

print 'Content-Type:
text/javascript\n'
print
"""displayOneD02("%s", "%.3f");"""%(elementID,D02)
else:
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = "%.3F",V02 = "%.3F",ER02 = "%.3F" WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(D02,V02,ER02,Pig_ID,Date_of_experiment,Time_of_experim
ent))
print 'Content-Type:
text/javascript\n'
print
"""displayAllD02V02ER02("%s", "%.3f", "%.3f", "%.3f");"""%(elementID,D02,V02,ER02)
else:
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type: text/javascript\n'
print
"""displayNoneD02V02ER02("%s");"""%(elementID)
else:
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type: text/javascript\n'
print
"""displayNoneD02V02ER02("%s");"""%(elementID)
else:
if(Vitalsonehour==1):
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Arterial_O2_sat = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Pig_ID,Date_of_experiment,Time_of_experiment))
cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET D02 = NULL,V02 = NULL,ER02 = NULL WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Pig_ID,Date_of_experiment,Time_of_experiment))
print 'Content-Type: text/javascript\n'
print
"""displayNoneD02V02ER02("%s");"""%(elementID)
else:
print 'Content-Type: text/javascript\n'
print
"""vitalsEmptyContent("%s");"""%(elementID)
else:
pass
if (form.has_key('lactate')):
if not (form['lactate'].value=="nullify"):
if(Vitalsonehour==0):
cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_nam
e) VALUES
("%s", "%s", "%s", "%s)"""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
else:

```



```

        pass
        Lactate = float(form['lactate'].value)
        cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Lactate = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Lactate,Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print """"vitalsEditSuccess("%s");""""%(elementID)
    else:
        if(Vitalsonehour==1):
            cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Lactate = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
            print 'Content-Type: text/javascript\n'
            print """"vitalsEmptyContent("%s");""""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"vitalsEmptyContent("%s");""""%(elementID)
    else:
        pass
        if (form.has_key('basedeficit')):
            if not (form['basedeficit'].value=="nullify"):
                if(Vitalsonehour==0):
                    cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s", "%s", "%s", "%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)
                else:
                    pass
                    Base_deficit = float(form['basedeficit'].value)
                    cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Base_deficit = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Base_deficit,Pig_ID,Date_of_experiment,Time_of_experiment))
                    print 'Content-Type: text/javascript\n'
                    print """"vitalsEditSuccess("%s");""""%(elementID)
                else:
                    if(Vitalsonehour==1):
                        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Base_deficit = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                        print 'Content-Type: text/javascript\n'
                        print """"vitalsEmptyContent("%s");""""%(elementID)
                    else:
                        print 'Content-Type: text/javascript\n'
                        print """"vitalsEmptyContent("%s");""""%(elementID)
            else:
                pass
                if (form.has_key('bladderpres')):
                    if not (form['bladderpres'].value=="nullify"):
                        if(Vitalsonehour==0):
                            cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s", "%s", "%s", "%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

```

```

else:
    pass
    Bladder_pressure = float(form['bladderpres'].value)
    cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Bladder_pressure = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Bladder_pressure,Pig_ID,Date_of_experiment,Time_of_exp
eriment))

    print 'Content-Type: text/javascript\n'
    print """"vitalsEditSuccess("%s");""%(elementID)
else:
    if(Vitalsonehour==1):
        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Bladder_pressure = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print """"vitalsEmptyContent("%s");""%(elementID)
    else:
        print 'Content-Type: text/javascript\n'
        print """"vitalsEmptyContent("%s");""%(elementID)
else:
    pass
    if (form.has_key('chesttubeoutput')):
        if not (form['chesttubeoutput'].value=="nullify"):
            if(Vitalsonehour==0):
                cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_nam
e) VALUES
("%s", "%s", "%s", "%s)""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name
)
                )
            else:
                pass
                Chesttube_output =
float(form['chesttubeoutput'].value)
                cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Chesttube_output = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Chesttube_output,Pig_ID,Date_of_experiment,Time_of_exp
eriment))

                print 'Content-Type: text/javascript\n'
                print """"vitalsEditSuccess("%s");""%(elementID)
            else:
                if(Vitalsonehour==1):
                    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Chesttube_output = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                    print 'Content-Type: text/javascript\n'
                    print """"vitalsEmptyContent("%s");""%(elementID)
                else:
                    print 'Content-Type: text/javascript\n'
                    print """"vitalsEmptyContent("%s");""%(elementID)
else:
    pass
    if (form.has_key('eventcomments')):
        if not (form['eventcomments'].value=="nullify"):
            if(Vitalsonehour==0):
                cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_nam

```

```

e) VALUES
("%s", "%s", "%s", "%s")""%(Pig_ID, Date_of_experiment, Time_of_experiment, Event_name)
)

else:
    pass
    Event_comments = form['eventcomments'].value
    cursor.execute("""UPDATE monr_vitals_manual_10_min SET
Event_comments = "%s" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(sql.escape_string(Event_comments), Pig_ID, Date_of_exper
iment, Time_of_experiment))
    print 'Content-Type: text/javascript\n'
    print ""vitalsEditSuccess("%s");""%(elementID)
else:
    if(Vitalsonehour==1):
        cursor.execute("""UPDATE monr_vitals_manual_10_min
SET Event_comments = NULL WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID, Date_of_experiment, Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print ""vitalsEmptyContent("%s");""%(elementID)
    else:
        print 'Content-Type: text/javascript\n'
        print ""vitalsEmptyContent("%s");""%(elementID)
else:
    pass
    if (form.has_key('specificgravity')):
        if not (form['specificgravity'].value=="nullify"):
            if(Vitalsonehour==0):
                cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID, Date_of_experiment, Time_of_experiment, Event_nam
e) VALUES
("%s", "%s", "%s", "%s")""%(Pig_ID, Date_of_experiment, Time_of_experiment, Event_name)
)

else:
    pass
    Specificgravity_urine =
float(form['specificgravity'].value)
    cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Specificgravity_urine = "%.3f" WHERE Pig_ID='%s' AND Date_of_experiment='%s'
AND
Time_of_experiment='%s'""%(Specificgravity_urine, Pig_ID, Date_of_experiment, Time_o
f_experiment))
    print 'Content-Type: text/javascript\n'
    print ""vitalsEditSuccess("%s");""%(elementID)
else:
    if(Vitalsonehour==1):
        cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Specificgravity_urine = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID, Date_of_experiment, Time_of_experiment))
        print 'Content-Type: text/javascript\n'
        print ""vitalsEmptyContent("%s");""%(elementID)
    else:
        print 'Content-Type: text/javascript\n'
        print ""vitalsEmptyContent("%s");""%(elementID)
else:
    pass
    if (form.has_key('sodium')):
        if not (form['sodium'].value=="nullify"):

```

```

        if(Vitalsonehour==0):
            cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s", "%s", "%s", "%s")"""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

            else:
                pass
                Sodium = int(form['sodium'].value)
                cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Sodium = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Sodium,Pig_ID,Date_of_experiment,Time_of_experiment))
                print 'Content-Type: text/javascript\n'
                print """"vitalsEditSuccess("%s");""""%(elementID)
            else:
                if(Vitalsonehour==1):
                    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Sodium = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                    print 'Content-Type: text/javascript\n'
                    print """"vitalsEmptyContent("%s");""""%(elementID)
                else:
                    print 'Content-Type: text/javascript\n'
                    print """"vitalsEmptyContent("%s");""""%(elementID)
        else:
            pass
            if (form.has_key('potassium')):
                if not (form['potassium'].value=="nullify"):
                    if(Vitalsonehour==0):
                        cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s", "%s", "%s", "%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

                        else:
                            pass
                            Potassium = float(form['potassium'].value)
                            cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Potassium = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Potassium,Pig_ID,Date_of_experiment,Time_of_experiment
))

                            print 'Content-Type: text/javascript\n'
                            print """"vitalsEditSuccess("%s");""""%(elementID)
                        else:
                            if(Vitalsonehour==1):
                                cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Potassium = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                                print 'Content-Type: text/javascript\n'
                                print """"vitalsEmptyContent("%s");""""%(elementID)
                            else:
                                print 'Content-Type: text/javascript\n'
                                print """"vitalsEmptyContent("%s");""""%(elementID)
                    else:
                        pass
                        if (form.has_key('calcium')):

```

```

        if not (form['calcium'].value=="nullify"):
            if(Vitalsonehour==0):
                cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s", "%s", "%s", "%s")"""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

                else:
                    pass
                    Calcium = float(form['calcium'].value)
                    cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Calcium = "%.2f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Calcium,Pig_ID,Date_of_experiment,Time_of_experiment))
                    print 'Content-Type: text/javascript\n'
                    print """"vitalsEditSuccess("%s");""%(elementID)
            else:
                if(Vitalsonehour==1):
                    cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Calcium = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                    print 'Content-Type: text/javascript\n'
                    print """"vitalsEmptyContent("%s");""%(elementID)
                else:
                    print 'Content-Type: text/javascript\n'
                    print """"vitalsEmptyContent("%s");""%(elementID)
        else:
            pass
            if (form.has_key('glucose')):
                if not (form['glucose'].value=="nullify"):
                    if(Vitalsonehour==0):
                        cursor.execute("""INSERT INTO
monr_vitals_manual_one_hour(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name) VALUES
("%s", "%s", "%s", "%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Event_name)
)

                        else:
                            pass
                            Glucose = int(form['glucose'].value)
                            cursor.execute("""UPDATE monr_vitals_manual_one_hour
SET Glucose = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Glucose,Pig_ID,Date_of_experiment,Time_of_experiment))
                            print 'Content-Type: text/javascript\n'
                            print """"vitalsEditSuccess("%s");""%(elementID)
                        else:
                            if(Vitalsonehour==1):
                                cursor.execute("""UPDATE
monr_vitals_manual_one_hour SET Glucose = NULL WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Pig_ID,Date_of_experiment,Time_of_experiment))
                                print 'Content-Type: text/javascript\n'
                                print """"vitalsEmptyContent("%s");""%(elementID)
                            else:
                                print 'Content-Type: text/javascript\n'
                                print """"vitalsEmptyContent("%s");""%(elementID)
                    else:
                        pass
            connection.commit()

```

```

        cursor.close()
        connection.close()
    except:
        try:
            connection.rollback()
        except:
            pass
            #raise
    print 'Content-Type: text/javascript\n'
    print ""vitalsEditFailure("%s");""%(elementID)
    #raise

#####
#####
#####
    elif (action == "viewalllabs") or (action == "viewalllabsnewwindow"):
        try:
            Pig_ID = form['pig_id'].value
            connection = sql.connect(host=lhost,port=lport,user=luser,
            passwd=lpasswd,db=lhb)
            cursor = connection.cursor()
            cursor.execute("""SELECT * FROM monr_identifier_table WHERE
            Pig_ID='%s'""%(Pig_ID))
            output = cursor.fetchall()
            if (len(output)>0):
                cursor.execute("""SELECT * FROM monr_lab_values WHERE
            Pig_ID='%s' ORDER BY Event_name""%(Pig_ID))
                output1 = cursor.fetchall()
                if(len(output1)>0):
                    datatable=""<table>""
                    evenRow = "True"
                    for i in range(len(output1)):
                        Event_name = str(output1[i][1])
                        if not (output1[i][2]==None):
                            Bilirubin_total = str(output1[i][2])
                        else:
                            Bilirubin_total = '&nbsp;';
                        if not (output1[i][3]==None):
                            Albumin = str(output1[i][3])
                        else:
                            Albumin = '&nbsp;';
                        if not (output1[i][4]==None):
                            Protein_total = str(output1[i][4])
                        else:
                            Protein_total = '&nbsp;';
                        if not (output1[i][5]==None):
                            Alk_phosphatase = str(output1[i][5])
                        else:
                            Alk_phosphatase = '&nbsp;';
                        if not (output1[i][6]==None):
                            ALT = str(output1[i][6])
                        else:
                            ALT = '&nbsp;';
                        if not (output1[i][7]==None):
                            AST = str(output1[i][7])
                        else:
                            AST = '&nbsp;';
                        if not (output1[i][8]==None):

```

```

        Urea_nitrogen = str(output1[i][8])
    else:
        Urea_nitrogen = '&nbsp;';
    if not (output1[i][9]==None):
        CK = str(output1[i][9])
    else:
        CK = '&nbsp;';
    if not (output1[i][10]==None):
        Creatinine = str(output1[i][10])
    else:
        Creatinine = '&nbsp;';
    if not (output1[i][11]==None):
        LD = str(output1[i][11])
    else:
        LD = '&nbsp;';
    if not (output1[i][12]==None):
        Platelet_count = str(output1[i][12])
    else:
        Platelet_count = '&nbsp;';
    if not (output1[i][13]==None):
        Urine_osmolality = str(output1[i][13])
    else:
        Urine_osmolality = '&nbsp;';
    if not (output1[i][14]==None):
        Urine_creatinine = str(output1[i][14])
    else:
        Urine_creatinine = '&nbsp;';
    elementID = Pig_ID+"_"+"Labs"+"_"+"Event_name"+"_"
    Event_ID = elementID+"newevent"
    Bilirubin_total_ID = elementID+"Bilirubintotal"
    Albumin_ID = elementID+"Albumin"
    Protein_total_ID = elementID+"Proteintotal"
    Alk_phosphatase_ID = elementID+"Alkphosphatase"
    ALT_ID = elementID+"ALT"
    AST_ID = elementID+"AST"
    Urea_nitrogen_ID = elementID+"Ureanitrogen"
    CK_ID = elementID+"CK"
    Creatinine_ID = elementID+"Creatinine"
    LD_ID = elementID+"LD"
    Platelet_count_ID = elementID+"Plateletcount"
    Urine_osmolality_ID = elementID+"Urineosmolality"
    Urine_creatinine_ID = elementID+"Urinecreatinine"
    if (evenRow=="True"):
        datatable = datatable + ""<tr
class="evenRow"><td><div id="%s" class="labsColWidth">%s</div></td><td><div
id="%s" class="labsColWidth">%s</div></td><td><div id="%s"
class="labsColWidth">%s</div></td><td><div id="%s"
class="labsColWidth">%s</div></td><td><div id="%s"
class="labsColWidth">%s</div></td><td><div id="%s"
class="labsColWidth">%s</div></td><td><div id="%s"
class="labsColWidth">%s</div></td><td><div id="%s"
class="labsColWidth">%s</div></td><td><div id="%s"
class="labsColWidth">%s</div></td><td><div id="%s"
class="labsColWidth">%s</div></td><td><div id="%s"
class="labsColWidth">%s</div></td></tr>""%(Event_ID,Event_name,Bilirubin_total_ID

```



```

        print """var givealert1 =
document.getElementById('message1');
        givealert1.innerHTML = 'The Subject ID you have
entered does not exist in the database. To create a new entry, you must complete
ID, Group and Weight fields';"""
        connection.commit()
        cursor.close()
        connection.close()
    except:
        print 'Content-Type: text/javascript\n'
        print """var givealert1 = document.getElementById('message1');
        givealert1.innerHTML = 'The ID you have entered does
not exist in the database. To create a new entry, you must complete ID, Group and
Weight fields';"""
        #raise

#####
#####
#####
        elif(action=="labsformsubmit"):
            try:
                Pig_ID = str(form['Pig_ID'].value)
                Event_name = form['Event_name'].value
                if Event_name=='Other':
                    Event_name='99'
                else:
                    pass
                Event_name = Event_name.replace(' ', '')
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
                cursor = connection.cursor()
                cursor.execute("""INSERT IGNORE INTO
monr_lab_values(Pig_ID,Event_name) VALUES ("%s", "%s")"""%(Pig_ID,Event_name))
                #####
                if (form.has_key('Bilirubin_total')):
                    Bilirubin_total = float(form['Bilirubin_total'].value)
                    cursor.execute("""UPDATE monr_lab_values SET
Bilirubin_total = "%f" WHERE Pig_ID='%s' AND
Event_name='%s'"""%(Bilirubin_total,Pig_ID,Event_name))
                else:
                    pass
                if (form.has_key('Albumin')):
                    Albumin = float(form['Albumin'].value)
                    cursor.execute("""UPDATE monr_lab_values SET Albumin =
"%f" WHERE Pig_ID='%s' AND Event_name='%s'""%(Albumin,Pig_ID,Event_name))
                else:
                    pass
                if (form.has_key('Protein_total')):
                    Protein_total = float(form['Protein_total'].value)
                    cursor.execute("""UPDATE monr_lab_values SET Protein_total
= "%f" WHERE Pig_ID='%s' AND Event_name='%s'""%(Protein_total,Pig_ID,Event_name))
                else:
                    pass
                if (form.has_key('Alk_phosphatase')):
                    Alk_phosphatase = int(form['Alk_phosphatase'].value)
                    cursor.execute("""UPDATE monr_lab_values SET
Alk_phosphatase = "%d" WHERE Pig_ID='%s' AND
Event_name='%s'""%(Alk_phosphatase,Pig_ID,Event_name))

```

```

else:
    pass
    if (form.has_key('ALT')):
        ALT = int(form['ALT'].value)
        cursor.execute("""UPDATE monr_lab_values SET ALT = "%d"
WHERE Pig_ID='%s' AND Event_name='%s'""%(ALT,Pig_ID,Event_name))
    else:
        pass
        if (form.has_key('AST')):
            AST = int(form['AST'].value)
            cursor.execute("""UPDATE monr_lab_values SET AST = "%d"
WHERE Pig_ID='%s' AND Event_name='%s'""%(AST,Pig_ID,Event_name))
        else:
            pass
            if (form.has_key('Urea_nitrogen')):
                Urea_nitrogen = int(form['Urea_nitrogen'].value)
                cursor.execute("""UPDATE monr_lab_values SET Urea_nitrogen
= "%d" WHERE Pig_ID='%s' AND Event_name='%s'""%(Urea_nitrogen,Pig_ID,Event_name))
            else:
                pass
                if (form.has_key('CK')):
                    CK = int(form['CK'].value)
                    cursor.execute("""UPDATE monr_lab_values SET CK = "%d"
WHERE Pig_ID='%s' AND Event_name='%s'""%(CK,Pig_ID,Event_name))
                else:
                    pass
                    if (form.has_key('Creatinine')):
                        Creatinine = float(form['Creatinine'].value)
                        cursor.execute("""UPDATE monr_lab_values SET Creatinine =
"%f" WHERE Pig_ID='%s' AND Event_name='%s'""%(Creatinine,Pig_ID,Event_name))
                    else:
                        pass
                        if (form.has_key('LD')):
                            LD = int(form['LD'].value)
                            cursor.execute("""UPDATE monr_lab_values SET LD = "%d"
WHERE Pig_ID='%s' AND Event_name='%s'""%(LD,Pig_ID,Event_name))
                        else:
                            pass
                            if (form.has_key('Platelet_count')):
                                Platelet_count = int(form['Platelet_count'].value)
                                cursor.execute("""UPDATE monr_lab_values SET
Platelet_count = "%d" WHERE Pig_ID='%s' AND
Event_name='%s'""%(Platelet_count,Pig_ID,Event_name))
                            else:
                                pass
                                if (form.has_key('Urine_osmolality')):
                                    Urine_osmolality = int(form['Urine_osmolality'].value)
                                    cursor.execute("""UPDATE monr_lab_values SET
Urine_osmolality = "%d" WHERE Pig_ID='%s' AND
Event_name='%s'""%(Urine_osmolality,Pig_ID,Event_name))
                                else:
                                    pass
                                    if (form.has_key('Urine_creatinine')):
                                        Urine_creatinine = int(form['Urine_creatinine'].value)
                                        cursor.execute("""UPDATE monr_lab_values SET
Urine_creatinine = "%d" WHERE Pig_ID='%s' AND
Event_name='%s'""%(Urine_creatinine,Pig_ID,Event_name))
                                else:

```

```

        pass
        connection.commit()
        cursor.close()
        connection.close()
        print 'Content-Type: text/javascript\n'
        print """var givealert3 = document.getElementById('message3');
                givealert3.innerHTML = 'Labs successfully updated';
                var alllabsdisplay =
document.getElementById('labsdataholder').style;
                labsRefresh();
                if(window['allLabsPopup'] &&
!window['allLabsPopup'].closed){
                    requestAllLabsNewWindow();
                }
                else if(alllabsdisplay.display=='inline'){
                    showAllLabs();
                }
                else{}"""
    except:
        try:
            connection.rollback()
        except:
            pass
            #raise
            print 'Content-Type: text/javascript\n'
            print """var givealert3 = document.getElementById('message3');
                    givealert3.innerHTML = 'There was an error in the
data entry. Please check your data and submit again';"""
            #raise

#####
#####
        elif(action=="editlabs"):
            try:
                elementID = form['elementid'].value
                Pig_ID = str(form['Pig_ID'].value)
                Event_name = form['Event_name'].value
                if Event_name=='Other':
                    Event_name='99'
                else:
                    pass
                Event_name = Event_name.replace(' ','')
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
                cursor = connection.cursor()
                #####
                if (form.has_key('newevent')):
                    if not((form['newevent'].value=="nullify")):
                        newEvent_name = form['newevent'].value
                        newEvent_name = newEvent_name.replace(' ','')
                        cursor.execute("""UPDATE monr_lab_values SET
Event_name='%s' WHERE Pig_ID='%s' AND
Event_name='%s'"""%(newEvent_name,Pig_ID,Event_name))
                        print 'Content-Type: text/javascript\n'
                        print
                        """requestAllLabsWindowIndependent();labsEditSuccess("%s");"""%(elementID)
                    else:
                        print 'Content-Type: text/javascript\n'

```

```

        print ""labsEditFailure("%s");""%(elementID)
    else:
        pass
    if (form.has_key('Bilirubintotal')):
        if not (form['Bilirubintotal'].value=="nullify"):
            Bilirubin_total = float(form['Bilirubintotal'].value)
            cursor.execute("""UPDATE monr_lab_values SET
Bilirubin_total=%f' WHERE Pig_ID=%s' AND
Event_name=%s'""%(Bilirubin_total,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print ""labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print ""labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('Albumin')):
        if not (form['Albumin'].value=="nullify"):
            Albumin = float(form['Albumin'].value)
            cursor.execute("""UPDATE monr_lab_values SET
Albumin=%f' WHERE Pig_ID=%s' AND Event_name=%s'""%(Albumin,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print ""labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print ""labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('Proteintotal')):
        if not (form['Proteintotal'].value=="nullify"):
            Protein_total = float(form['Proteintotal'].value)
            cursor.execute("""UPDATE monr_lab_values SET
Protein_total=%f' WHERE Pig_ID=%s' AND
Event_name=%s'""%(Protein_total,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print ""labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print ""labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('Alkphosphatase')):
        if not (form['Alkphosphatase'].value=="nullify"):
            Alk_phosphatase = int(form['Alkphosphatase'].value)
            cursor.execute("""UPDATE monr_lab_values SET
Alk_phosphatase=%d' WHERE Pig_ID=%s' AND
Event_name=%s'""%(Alk_phosphatase,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print ""labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print ""labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('ALT')):
        if not (form['ALT'].value=="nullify"):
            ALT = int(form['ALT'].value)

```

```

        cursor.execute("""UPDATE monr_lab_values SET ALT='%d'
WHERE Pig_ID='%s' AND Event_name='%s'""%(ALT,Pig_ID,Event_name))
        print 'Content-Type: text/javascript\n'
        print """"labsEditSuccess("%s");""%(elementID)
    else:
        print 'Content-Type: text/javascript\n'
        print """"labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('AST')):
        if not (form['AST'].value=="nullify"):
            AST = int(form['AST'].value)
            cursor.execute("""UPDATE monr_lab_values SET AST='%d'
WHERE Pig_ID='%s' AND Event_name='%s'""%(AST,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print """"labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('Ureanitrogen')):
        if not (form['Ureanitrogen'].value=="nullify"):
            Urea_nitrogen = int(form['Ureanitrogen'].value)
            cursor.execute("""UPDATE monr_lab_values SET
Urea_nitrogen='%d' WHERE Pig_ID='%s' AND
Event_name='%s'""%(Urea_nitrogen,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print """"labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('CK')):
        if not (form['CK'].value=="nullify"):
            CK = int(form['CK'].value)
            cursor.execute("""UPDATE monr_lab_values SET CK='%d'
WHERE Pig_ID='%s' AND Event_name='%s'""%(CK,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print """"labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('Creatinine')):
        if not (form['Creatinine'].value=="nullify"):
            Creatinine = float(form['Creatinine'].value)
            cursor.execute("""UPDATE monr_lab_values SET
Creatinine='%f' WHERE Pig_ID='%s' AND
Event_name='%s'""%(Creatinine,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print """"labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"labsEmptyContent("%s");""%(elementID)
    else:

```

```

        pass
    if (form.has_key('LD')):
        if not (form['LD'].value=="nullify"):
            LD = int(form['LD'].value)
            cursor.execute("""UPDATE monr_lab_values SET LD='%d'
WHERE Pig_ID='%s' AND Event_name='%s'""%(LD,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print """"labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('Plateletcount')):
        if not (form['Plateletcount'].value=="nullify"):
            Platelet_count = int(form['Plateletcount'].value)
            cursor.execute("""UPDATE monr_lab_values SET
Platelet_count='%d' WHERE Pig_ID='%s' AND
Event_name='%s'""%(Platelet_count,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print """"labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('Urineosmolality')):
        if not (form['Urineosmolality'].value=="nullify"):
            Urine_osmolality = int(form['Urineosmolality'].value)
            cursor.execute("""UPDATE monr_lab_values SET
Urine_osmolality='%d' WHERE Pig_ID='%s' AND
Event_name='%s'""%(Urine_osmolality,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print """"labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"labsEmptyContent("%s");""%(elementID)
    else:
        pass
    if (form.has_key('Urinecreatinine')):
        if not (form['Urinecreatinine'].value=="nullify"):
            Urine_creatinine = int(form['Urinecreatinine'].value)
            cursor.execute("""UPDATE monr_lab_values SET
Urine_creatinine='%d' WHERE Pig_ID='%s' AND
Event_name='%s'""%(Urine_creatinine,Pig_ID,Event_name))
            print 'Content-Type: text/javascript\n'
            print """"labsEditSuccess("%s");""%(elementID)
        else:
            print 'Content-Type: text/javascript\n'
            print """"labsEmptyContent("%s");""%(elementID)
    else:
        pass
    connection.commit()
    cursor.close()
    connection.close()
except:
    try:
        connection.rollback()

```

```

except:
    pass
    #raise
print 'Content-Type: text/javascript\n'
print ""labsEditFailure("%s");""%(elementID)
#raise

#####
#####
#####
elif (action == "viewallgcscscores"):
    try:
        Pig_ID = form['pig_id'].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
        cursor = connection.cursor()
        cursor.execute("""SELECT * FROM monr_identifier_table WHERE
Pig_ID='%s'""%(Pig_ID))
        output = cursor.fetchall()
        if (len(output)>0):
            cursor.execute("""SELECT * FROM monr_gcs_scores WHERE
Pig_ID='%s' ORDER BY Event_name""%(Pig_ID))
            output1 = cursor.fetchall()
            if (len(output1)>0):
                datatable=""<table>""
                evenRow = "True"
                Baseline = "&nbsp;"
                FR20 = "&nbsp;"
                PR24 = "&nbsp;"
                PR48 = "&nbsp;"
                for i in range(len(output1)):
                    if (output1[i][1]==1):
                        if not (output1[i][2]==None):
                            Baseline = str(output1[i][2])
                        else:
                            pass
                    elif (output1[i][1]==23):
                        if not (output1[i][2]==None):
                            FR20 = str(output1[i][2])
                        else:
                            pass
                    elif (output1[i][1]==24):
                        if not (output1[i][2]==None):
                            PR24 = str(output1[i][2])
                        else:
                            pass
                    elif (output1[i][1]==25):
                        if not (output1[i][2]==None):
                            PR48 = str(output1[i][2])
                        else:
                            pass
                BaselineID = Pig_ID+"_"+"Baseline"
                FR20ID = Pig_ID+"_"+"FR20"
                PR24ID = Pig_ID+"_"+"PR24"
                PR48ID = Pig_ID+"_"+"PR48"
                if (evenRow=="True"):
                    datatable = datatable + ""<tr
class="evenRow"><td><div id="%s" class="gcscscoresColWidth">%s</div></td><td><div

```

```

id="%s" class="gcsscoresColWidth">%s</div></td><td><div id="%s"
class="gcsscoresColWidth">%s</div><td><div id="%s"
class="gcsscoresColWidth">%s</div></td></tr>""%(BaselineID,Baseline,FR20ID,FR20,P
R24ID,PR24,PR48ID,PR48)
        evenRow = "False"
    else:
        datatable = datatable + ""<tr
class="oddRow"><td><div id="%s" class="gcsscoresColWidth">%s</div></td><td><div
id="%s" class="gcsscoresColWidth">%s</div></td><td><div id="%s"
class="gcsscoresColWidth">%s</div></td><td><div id="%s"
class="gcsscoresColWidth">%s</div></td></tr>""%(BaselineID,Baseline,FR20ID,FR20,P
R24ID,PR24,PR48ID,PR48)
        evenRow = "True"
        print 'Content-Type: text/javascript\n'
        datatable = datatable+""</table>""
        datatable = datatable.replace('<div id','<div
onclick="javascript:editGcssscoresElement(this);" id')
        print
        ""document.getElementById('gcsscoresinnerDiv').innerHTML = '%s';""%datatable
    else:
        print 'Content-Type: text/javascript\n'
        print ""var givealert5 =
document.getElementById('gcsscoresinnerDiv');
        givealert5.innerHTML = '<h2>No GCS scores data
entered yet</h2>';""
    else:
        print 'Content-Type: text/javascript\n'
        print ""var givealert1 =
document.getElementById('message1');
        givealert1.innerHTML = 'The ID you have entered
does not exist in the database. To create a new entry, you must complete ID, Group
and Weight fields';""
        connection.commit()
        cursor.close()
        connection.close()
    except:
        print 'Content-Type: text/javascript\n'
        print ""var givealert1 = document.getElementById('message1');
        givealert1.innerHTML = 'The ID you have entered does
not exist in the database. To create a new entry, you must complete ID, Group and
Weight fields';""
        #raise

#####
#####
#####
elif(action=="gcsscoresformsubmit"):
    try:
        Pig_ID = str(form['Pig_ID'].value)
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lodb)
        cursor = connection.cursor()
        #####
        if (form.has_key('gcs1')):
            Event_name = "1"
            gcs1 = int(form['gcs1'].value)
            cursor.execute("""INSERT IGNORE INTO
monr_gcs_scores(Pig_ID,Event_name) VALUES ('%s','%s')""%(Pig_ID,Event_name))

```



```

        cursor.execute("""UPDATE monr_gcs_scores SET GCS_score=%d
WHERE Pig_ID='%s' AND Event_name='%s'"""%(gcs1,Pig_ID,Event_name))
    else:
        pass
    if (form.has_key('gcs23')):
        Event_name = "23"
        gcs23 = int(form['gcs23'].value)
        cursor.execute("""INSERT IGNORE INTO
monr_gcs_scores(Pig_ID,Event_name) VALUES ("%s","%s")""%(Pig_ID,Event_name))
        cursor.execute("""UPDATE monr_gcs_scores SET GCS_score=%d
WHERE Pig_ID='%s' AND Event_name='%s'""%(gcs23,Pig_ID,Event_name))
    else:
        pass
    if (form.has_key('gcs24')):
        Event_name = "24"
        gcs24 = int(form['gcs24'].value)
        cursor.execute("""INSERT IGNORE INTO
monr_gcs_scores(Pig_ID,Event_name) VALUES ("%s","%s")""%(Pig_ID,Event_name))
        cursor.execute("""UPDATE monr_gcs_scores SET GCS_score=%d
WHERE Pig_ID='%s' AND Event_name='%s'""%(gcs24,Pig_ID,Event_name))
    else:
        pass
    if (form.has_key('gcs25')):
        Event_name = "25"
        gcs25 = int(form['gcs25'].value)
        cursor.execute("""INSERT IGNORE INTO
monr_gcs_scores(Pig_ID,Event_name) VALUES ("%s","%s")""%(Pig_ID,Event_name))
        cursor.execute("""UPDATE monr_gcs_scores SET GCS_score=%d
WHERE Pig_ID='%s' AND Event_name='%s'""%(gcs25,Pig_ID,Event_name))
    else:
        pass
    connection.commit()
    cursor.close()
    connection.close()
    print 'Content-Type: text/javascript\n'
    print """gcsscoresRefresh();showAllGcsscores();
        var givealert4 = document.getElementById('message4');
        givealert4.innerHTML = 'GCS scores successfully
updated';
        """
except:
    try:
        connection.rollback()
    except:
        pass
        #raise
    print 'Content-Type: text/javascript\n'
    print """var givealert4 = document.getElementById('message4');
        givealert4.innerHTML = 'There was an error in the
data entry. Please check your data and submit again';"""
        #raise

#####
#####
elif(action=="editgcsscores"):
    try:
        elementID = form['elementid'].value
        Pig_ID = str(form['Pig_ID'].value)

```

```

Event_name = form['Event_name'].value
if Event_name=='Other':
    Event_name='99'
else:
    pass
Event_name = Event_name.replace(' ', '')
if (Event_name=='Baseline'):
    Event_name='1'
elif (Event_name=='FR20'):
    Event_name='23'
elif(Event_name=='PR24'):
    Event_name='24'
elif(Event_name=='PR48'):
    Event_name='25'
else:
    pass
connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
cursor = connection.cursor()
#####
if (form.has_key('field_value')):
    if not((form['field_value'].value=="nullify")):
        field_value = int(form['field_value'].value)
        cursor.execute("""UPDATE monr_gcs_scores SET
GCS_score='%d' WHERE Pig_ID='%s' AND
Event_name='%s'""%(field_value,Pig_ID,Event_name))
        print 'Content-Type: text/javascript\n'
        print ""gcsscoresEditSuccess("%s");""%(elementID)
    else:
        cursor.execute("""UPDATE monr_gcs_scores SET
GCS_score=NULL WHERE Pig_ID='%s' AND Event_name='%s'""%(Pig_ID,Event_name))
        print 'Content-Type: text/javascript\n'
        print ""gcsscoresEmptyContent("%s");""%(elementID)
else:
    pass
connection.commit()
cursor.close()
connection.close()
except:
    try:
        connection.rollback()
    except:
        pass
        #raise
print 'Content-Type: text/javascript\n'
print ""gcsscoresEditFailure("%s");""%(elementID)
#raise

#####
#####
#####
elif(action=="adduser"):
    User_id = form["User_id"].value
    Password = form["Password"].value
    Email = form["Email"].value
    User_first_name = form["User_first_name"].value
    if(form.has_key('User_last_name')):
        User_last_name = form["User_last_name"].value

```

```

else:
    User_last_name = ''
if(form.has_key('User_middle_initial')):
    User_middle_initial = form["User_middle_initial"].value
else:
    User_middle_initial = ''
Group_name = form["Group_name"].value
TestOnly = form["TestOnly"].value
MONR = form["MONR"].value
PONR = form["PONR"].value
DARPA = form["DARPA"].value
All_studies = form["All_studies"].value
Account_creation_date = form["Account_creation_date"].value
try:
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
    cursor = connection.cursor()
    cursor.execute("""SELECT * FROM sccl_authentication WHERE
User_id='%s'""%(User_id))
    output=cursor.fetchall()
    if ((len(output))>0):
        print 'Content-Type: text/javascript\n'
        print
        """document.getElementById('message1').innerHTML='User ID already exists. Please
check the User IDs below';"""
    else:
        cursor.execute("""INSERT INTO
sccltest_authentication(User_id>Password>Email>User_first_name>User_last_name>User
_middle_initial>Group_name>TestOnly>MONR>PONR>DARPA>All_studies>Account_creation_d
ate) VALUES
("%s",md5(sha1("%s")),"%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s", now())""%(
User_id>Password>Email>User_first_name>User_last_name>User_middle_initial>Group_n
ame>TestOnly>MONR>PONR>DARPA>All_studies))#Ignore_this###
        cursor.execute("""INSERT INTO
sccl_authentication(User_id>Password>Email>User_first_name>User_last_name>User_mid
dle_initial>Group_name>TestOnly>MONR>PONR>DARPA>All_studies>Account_creation_date)
VALUES
("%s",md5(sha1("%s")),"%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s", now())""%(
User_id>Password>Email>User_first_name>User_last_name>User_middle_initial>Group_n
ame>TestOnly>MONR>PONR>DARPA>All_studies))
        print 'Content-Type: text/javascript\n'
        print
        """document.getElementById('message1').innerHTML='User added
successfully';clearFields();loadUserData();"""
        connection.commit()
        cursor.close()
        connection.close()
except:
    print 'Content-Type: text/javascript\n'
    print """document.getElementById('message1').innerHTML='An
error occurred. Please contact data manager';"""
    #raise
elif(action=="getuserdata"):
    try:
        Studies = ''
        Access_type = ''
        individualuserdata = ''
        row = "odd"

```

```

totaluserdata = '<table><tbody><tr><td><table
class="userdata"><tbody><tr class="holdHeaders"><td><div
style="width:40px">&nbsp;</div></td><td><div style="width:80px">User
ID</div></td>\
                                <td><div
style="width:190px">Email</div></td><td><div
style="width:120px">Name</div></td><td><div style="width:100px">Group
Name</div></td>\
                                <td><div style="width:90px">Access
Type</div><div style="width:80px">Studies Assigned</div></td>\
                                <td><div style="width:80px">Account
Creation Date</div></td><td><div style="width:80px">Last Login
Time</div></td></tr></tbody></table></td></tr>'
connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
cursor = connection.cursor()
cursor.execute("""SELECT * FROM sccl_authentication ORDER BY
Account_creation_date""")
output = cursor.fetchall()
for i in range(len(output)):
    User_id = output[i][0]
    Email = output[i][3].split("@")
    Email = Email[0]+"@"+Email[1]
    User_first_name = output[i][4]+" "
    User_last_name = output[i][5]
    User_middle_initial = output[i][6]+" "
    User_name =
User_first_name+User_middle_initial+User_last_name
    Group_name = output[i][7]
    if(Group_name=="Lab_user"):
        Group_name="Lab User"
    elif(Group_name=="Admin"):
        Group_name = "Administrator"
    elif(Group_name=="Special"):
        Group_name = "Special Access"
    else:
        pass
    TestOnly = output[i][8]
    if(TestOnly=="Yes"):
        Access_type = "Test Only"
    else:
        Access_type = "Full"
    MONR = output[i][9]
    PONR = output[i][10]
    DARPA = output[i][11]
    All_studies = output[i][12]
    if (All_studies=="Yes"):
        Studies = "All"
    else:
        if(output[i][9]=="Yes"):
            MONR = "MONR<br/>"
        else:
            MONR = ''
        if(output[i][10]=="Yes"):
            PONR = "PONR<br/>"
        else:
            PONR = ''
        if(output[i][11]=="Yes"):

```

```

        DARPA = "DARPA"
    else:
        DARPA = ''
    Studies = MONR+PONR+DARPA
    Account_creation_date = str(output[i][13])
    Last_login_time = str(output[i][14])
    if (row=="odd"):
        individualuserdata = individualuserdata+'<tr
class="oddRow">'
        row = "even"
    else:
        individualuserdata = individualuserdata+'<tr
class="evenRow">'
        row = "odd"
        individualuserdata = individualuserdata+""<td
id="%s"><table class="userdata"><tbody><tr><td class="clickoption"><div
style="width:40px" onclick="javascript:editUserData(\\'%s\\');">Edit</div></td>\\
        <td><div
style="width:80px">%s</div></td>\\
        <td><div
style="width:190px">%s</div></td>\\
        <td><div
style="width:120px">%s</div></td>\\
        <td><div
style="width:100px">%s</div></td>\\
        <td><div
style="width:90px">%s</div></td>\\
        <td><div
style="width:80px">%s</div></td>\\
        <td class="numbers"><div
style="width:80px">%s</div></td>\\
        <td class="numbers"><div
style="width:80px">%s</div></td></tr></tbody></table></td></tr>""%(User_id,User_i
d,User_id,Email,User_name,Group_name,Access_type,Studies,Account_creation_date,Las
t_login_time)
    totaluserdata =
totaluserdata+individualuserdata+""</tbody></table>""
    connection.commit()
    cursor.close()
    connection.close()
    print 'Content-Type: text/javascript\\n'
    print
    ""document.getElementById('addedusers').innerHTML='%s';""%(totaluserdata)
    except:
        print 'Content-Type: text/javascript\\n'
        print ""document.getElementById('addedusers').innerHTML='An
error occurred. Please contact data manager';""
        #raise
    elif(action=="loadedituserdata"):
        User_id = form["User_id"].value
        userdata = ""
        htmlfile = ''
        try:
            connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
            cursor = connection.cursor()
            cursor.execute("""SELECT * FROM sccl_authentication WHERE
User_id='%s'""%(User_id))

```

```

output = cursor.fetchall()
for i in range(len(output)):
    User_id = output[i][0]
    Email = output[i][3]
    User_first_name = output[i][4]
    User_last_name = output[i][5]
    User_middle_initial = output[i][6]
    Group_name = output[i][7]
    TestOnly = output[i][8]
    MONR = output[i][9]
    PONR = output[i][10]
    DARPA = output[i][11]
    All_studies = output[i][12]
connection.commit()
cursor.close()
connection.close()
print 'Content-Type: text/javascript\n'
print
"""loadEditUserData('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s');"""%(U
ser_id,Email,User_first_name,User_last_name,User_middle_initial,Group_name,TestOnl
y,MONR,PONR,DARPA,All_studies)
except:
    print 'Content-Type: text/javascript\n'
    print """document.getElementById('message1').innerHTML='There
was an error';"""
    #raise
elif(action=="edituser"):
    try:
        oldUser_id = form["oldUser_id"].value
        User_id = form["User_id"].value
        Password = form["Password"].value
        Email = form["Email"].value
        User_first_name = form["User_first_name"].value
        if(form.has_key('User_last_name')):
            User_last_name = form["User_last_name"].value
        else:
            User_last_name = ''
        if(form.has_key('User_middle_initial')):
            User_middle_initial = form["User_middle_initial"].value
        else:
            User_middle_initial = ''
        Group_name = form["Group_name"].value
        TestOnly = form["TestOnly"].value
        MONR = form["MONR"].value
        PONR = form["PONR"].value
        DARPA = form["DARPA"].value
        All_studies = form["All_studies"].value
        connection = sql.connect(host=lhost,port=lport,user=user,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        if not (Password=="None"):
            cursor.execute("""UPDATE sccl_authentication SET
User_id='%s',Password=md5(sha1('%s')),Email='%s',User_first_name='%s',User_last_na
me='%s',User_middle_initial='%s',Group_name='%s',TestOnly='%s',MONR='%s',PONR='%s'
,DARPA='%s',All_studies='%s' WHERE
User_id='%s'"""%(User_id>Password,Email,User_first_name,User_last_name,User_middle
_initial,Group_name,TestOnly,MONR,PONR,DARPA,All_studies,oldUser_id))
        else:

```

```

        cursor.execute("""UPDATE sccl_authentication SET
User_id='%s',Email='%s',User_first_name='%s',User_last_name='%s',User_middle_initi
al='%s',Group_name='%s',TestOnly='%s',MONR='%s',PONR='%s',DARPA='%s',All_studies='
%s' WHERE
User_id='%s'""%(User_id,Email,User_first_name,User_last_name,User_middle_initial,
Group_name,TestOnly,MONR,PONR,DARPA,All_studies,oldUser_id))
        connection.commit()
        cursor.close()
        connection.close()
        print 'Content-Type: text/javascript\n'
        print ""document.getElementById('message1').innerHTML='User
data updated successfully';loadUserData();""
    except:
        print 'Content-Type: text/javascript\n'
        print ""document.getElementById('message2').innerHTML='An
error occurred. Please contact data manager';""
        #raise
    elif(action=="deleteuser"):
        User_id = form["User_id"].value
        try:
            connection = sql.connect(host=lhost,port=lport,user=user,
passwd=lpasswd,db=ldb)
            cursor = connection.cursor()
            cursor.execute("""DELETE FROM sccltest_authentication WHERE
User_id='%s'""%(User_id))#Ignore_this###
            cursor.execute("""DELETE FROM sccl_authentication WHERE
User_id='%s'""%(User_id))
            connection.commit()
            cursor.close()
            connection.close()
            print 'Content-Type: text/javascript\n'
            print ""document.getElementById('message1').innerHTML='User
%s is removed from the user list';loadUserData();""%(User_id)
        except:
            print 'Content-Type: text/javascript\n'
            print ""document.getElementById('message1').innerHTML='There
was an error. User %s is not removed from the user list. Please contact data
manager.';loadUserData();""%(User_id)
            #raise
    elif(action=="storepigids"):
        #suggestionbox = ""<?xml version="1.0"?><choices
xml:lang="EN">""
        #suggestion = ""
        suggestions = ""
        try:
            connection = sql.connect(host=lhost,port=lport,user=user,
passwd=lpasswd,db=ldb)
            cursor = connection.cursor()
            cursor.execute("""SELECT Pig_ID FROM monr_identifier_table
ORDER BY Start_date""")
            output = cursor.fetchall()
            for i in range(len(output)):
                for j in range(len(output[i])):
                    #suggestion =
suggestion+'<suggestion>'+output[i][j]+'</suggestion>'
                    suggestions = suggestions+'_'+output[i][j]
                    #suggestionbox=suggestionbox+suggestion+""</choices>""
            connection.commit()

```

```

        cursor.close()
        connection.close()
        print 'Content-Type: text/javascript\n'
        #print """"Pig_ID_suggestions='%s';""""%(suggestionbox)
        print """"var
suggestions='%s';subjectsArray=suggestions.split('_');""""%(suggestions)
    except:
        print 'Content-Type: text/javascript\n'
        print """"document.getElementById('message1').innerHTML='There
was an error. Pig IDs for the study %s could not be loaded.';""""%(studyname)
        raise
    elif(action=="loadpigids"):
        session_Pig_ID = ''
        #fileupload_filetype=''
        SelectNode = """"<select name="pigids" id="pigids"
onchange="javascript:selectFileType(this.value);">""""
        #onchange="javascript:loadFiles();" #This should have been inside
the SelectNode, but the code has been changed to file type
        try:
            try:
                session_Pig_ID = a_cookie["session_Pig_ID"].value
                #fileupload_filetype=a_cookie["fileupload_filetype"].value
                options = '<option value="Select">Select</option>'
            except:
                session_Pig_ID = "Select"
                #fileupload_filetype="Select"
                options = '<option selected="selected"
value="Select">Select</option>'
                #raise
            connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
            cursor = connection.cursor()
            cursor.execute("""SELECT Pig_ID FROM monr_identifier_table
ORDER BY Start_date""")
            output = cursor.fetchall()
            for i in range(len(output)):
                for j in range(len(output[i])):
                    if(output[i][j]==session_Pig_ID):
                        options = options+'<option selected="selected"
value=\''+output[i][j]+'\'>'+output[i][j]+'</option>'
                    else:
                        options = options+'<option
value=\''+output[i][j]+'\'>'+output[i][j]+'</option>'
            SelectNode=SelectNode+options+""""</select>""""
            connection.commit()
            cursor.close()
            connection.close()
            print 'Content-Type: text/javascript\n'
            print
            """"document.getElementById('holdpigids').innerHTML='%s';selectFileType(document.up
loadfileform.pigids.value);/*loadFiles();*/""""%(SelectNode)
        except:
            print 'Content-Type: text/javascript\n'
            print """"document.getElementById('message1').innerHTML='There
was an error. Pig IDs for the study %s could not be loaded.';""""%(studyname)
            raise
    elif(action=="loadfiles"):
        Pig_ID = form["Pig_ID"].value

```



```

File_type = form["File_type"].value
singlefile = ''
allfiles = ''
uniqueid = ''
File_size = 0
try:
    row = "odd"
    allfiles = ""<table class="uploadedfiles"><tbody><tr
class="holdHeaders"><td>Type of Data</td><td>File Name</td>\
    <td>File Type</td><td>File
Size(kb)</td><td>Time of Upload</td><!-- <td>Edit Data Type</td> -->\
    <!-- <td>Delete File</td> --><td>Download
File</td></tr>""
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
    cursor = connection.cursor()
    cursor.execute("""SELECT
Data_type,File_name,Upload_time,File_type,File_size FROM monr_fileupload WHERE
Pig_ID='%s' AND Data_type LIKE '%s%' ORDER BY CASE
    WHEN Data_type LIKE 'Vitals' THEN 1
    WHEN Data_type LIKE 'Lab_Values' THEN 2
    WHEN Data_type LIKE 'GCS_Scores' THEN 3
    WHEN Data_type LIKE 'Binning_Liver_0%'
THEN 4
    WHEN Data_type LIKE 'Binning_Liver_1' THEN
5
    WHEN Data_type LIKE 'Binning_Liver_2' THEN
6
    WHEN Data_type LIKE 'Binning_Liver_5' THEN
7
    WHEN Data_type LIKE 'Binning_Liver_11'
THEN 8
    WHEN Data_type LIKE 'Binning_Liver_23'
THEN 9
    WHEN Data_type LIKE 'Binning_Liver_25'
THEN 10
    WHEN Data_type LIKE 'Binning_Muscle_0%'
THEN 11
    WHEN Data_type LIKE 'Binning_Muscle_1'
THEN 12
    WHEN Data_type LIKE 'Binning_Muscle_2'
THEN 13
    WHEN Data_type LIKE 'Binning_Muscle_5'
THEN 14
    WHEN Data_type LIKE 'Binning_Muscle_11'
THEN 15
    WHEN Data_type LIKE 'Binning_Muscle_23'
THEN 16
    WHEN Data_type LIKE 'Binning_Muscle_25'
THEN 17
    WHEN Data_type LIKE 'Binning_Serum_0%'
THEN 18
    WHEN Data_type LIKE 'Binning_Serum_1' THEN
19
    WHEN Data_type LIKE 'Binning_Serum_2' THEN
20
    WHEN Data_type LIKE 'Binning_Serum_5' THEN
21

```

```

THEN 22      WHEN Data_type LIKE 'Binning_Serum_11'
THEN 23      WHEN Data_type LIKE 'Binning_Serum_23'
THEN 24      WHEN Data_type LIKE 'Binning_Serum_25'
THEN 25      WHEN Data_type LIKE 'Binning_Urine_0%%'
26           WHEN Data_type LIKE 'Binning_Urine_1' THEN
27           WHEN Data_type LIKE 'Binning_Urine_2' THEN
28           WHEN Data_type LIKE 'Binning_Urine_5' THEN
THEN 29      WHEN Data_type LIKE 'Binning_Urine_11'
THEN 30      WHEN Data_type LIKE 'Binning_Urine_23'
THEN 31      WHEN Data_type LIKE 'Binning_Urine_25'
32           WHEN Data_type LIKE 'HNMR_Liver_0%%' THEN
33           WHEN Data_type LIKE 'HNMR_Liver_1' THEN 33
34           WHEN Data_type LIKE 'HNMR_Liver_2' THEN 34
35           WHEN Data_type LIKE 'HNMR_Liver_5' THEN 35
36           WHEN Data_type LIKE 'HNMR_Liver_11' THEN
37           WHEN Data_type LIKE 'HNMR_Liver_23' THEN
38           WHEN Data_type LIKE 'HNMR_Liver_25' THEN
39           WHEN Data_type LIKE 'HNMR_Muscle_0%%' THEN
40           WHEN Data_type LIKE 'HNMR_Muscle_1' THEN
41           WHEN Data_type LIKE 'HNMR_Muscle_2' THEN
42           WHEN Data_type LIKE 'HNMR_Muscle_5' THEN
43           WHEN Data_type LIKE 'HNMR_Muscle_11' THEN
44           WHEN Data_type LIKE 'HNMR_Muscle_23' THEN
45           WHEN Data_type LIKE 'HNMR_Muscle_25' THEN
46           WHEN Data_type LIKE 'HNMR_Serum_0%%' THEN
47           WHEN Data_type LIKE 'HNMR_Serum_1' THEN 47
48           WHEN Data_type LIKE 'HNMR_Serum_2' THEN 48
49           WHEN Data_type LIKE 'HNMR_Serum_5' THEN 49
50           WHEN Data_type LIKE 'HNMR_Serum_11' THEN
51           WHEN Data_type LIKE 'HNMR_Serum_23' THEN
52           WHEN Data_type LIKE 'HNMR_Serum_25' THEN
53           WHEN Data_type LIKE 'HNMR_Urine_0%%' THEN

```

```

57         WHEN Data_type LIKE 'HNMR_Urine_1' THEN 54
58         WHEN Data_type LIKE 'HNMR_Urine_2' THEN 55
59         WHEN Data_type LIKE 'HNMR_Urine_5' THEN 56
60         WHEN Data_type LIKE 'HNMR_Urine_11' THEN
61
62         WHEN Data_type LIKE 'HNMR_Urine_23' THEN
63
64         WHEN Data_type LIKE 'HNMR_Urine_25' THEN
65
66         WHEN Data_type LIKE 'PNMR_Liver_0%%' THEN
67
68         WHEN Data_type LIKE 'PNMR_Liver_1' THEN 61
69         WHEN Data_type LIKE 'PNMR_Liver_2' THEN 62
70         WHEN Data_type LIKE 'PNMR_Liver_5' THEN 63
71         WHEN Data_type LIKE 'PNMR_Liver_11' THEN
72
73         WHEN Data_type LIKE 'PNMR_Liver_23' THEN
74
75         WHEN Data_type LIKE 'PNMR_Liver_25' THEN
76
77         WHEN Data_type LIKE 'PNMR_Muscle_0%%' THEN
78
79         WHEN Data_type LIKE 'PNMR_Muscle_1' THEN
80
81         WHEN Data_type LIKE 'PNMR_Muscle_2' THEN
82
83         WHEN Data_type LIKE 'PNMR_Muscle_5' THEN
84
85         WHEN Data_type LIKE 'PNMR_Muscle_11' THEN
86
87         WHEN Data_type LIKE 'PNMR_Muscle_23' THEN
88
89         WHEN Data_type LIKE 'PNMR_Muscle_25' THEN
90
91         END""%(Pig_ID,File_type))
92
93 output = cursor.fetchall()
94 if(len(output)>0):
95     for i in range(len(output)):
96         Data_type = output[i][0]
97         Data_type1 = re.sub("_", " ",Data_type)
98         File_name = output[i][1]
99         Upload_time = str(output[i][2])
100        File_type = output[i][3]
101        File_size = output[i][4]
102        File_size = int(File_size/1000)
103        if(File_size<1):
104            File_size=1
105        else:
106            pass
107        if (row=="odd"):
108            singlefile = singlefile+'<tr class="oddRow">'
109            row = "even"
110        else:
111            singlefile = singlefile+'<tr class="evenRow">'
112            row = "odd"
113        uniqueid = Pig_ID+Data_type
114        singlefile = singlefile+""""<td>%s</td>\
115            <td>%s</td>\

```

```

                                <td>%s</td>\
                                <td class="numbers">%d</td>\
                                <td class="numbers">%s</td>\
                                <!-- <td class="clickoption"
onclick="javascript:editDataType(\\'%s\\',\\'%s\\',this);">Edit</td> -->\
                                <!-- <td class="clickoption"
onclick="javascript:deleteFile(\\'%s\\',\\'%s\\');">Delete</td> -->\
                                <td class="clickoption"
onclick="javascript:downloadFile(\\'%s\\',\\'%s\\',\\'%s\\');">Download</td></tr>"
""%(Data_type1,File_name,File_type,File_size,Upload_time,Pig_ID,Data_type,Pig_ID,D
ata_type,studyname,Pig_ID,Data_type)
                                allfiles = allfiles+singlefile+""</tbody></table>""
                                else:
                                    allfiles = ''
                                    connection.commit()
                                    cursor.close()
                                    connection.close()
                                    print 'Content-Type: text/javascript\n'
                                    print
                                ""document.getElementById('filesholder').innerHTML='%s';""%(allfiles)
                                except:
                                    print 'Content-Type: text/javascript\n'
                                    print ""document.getElementById('message1').innerHTML='An
error occurred. Please contact data manager';""
                                    #raise
                                    ##### This part of the code below has been removed
#####
#####
#####
                                #elif(action=="deletefile"):
                                    ""
                                    Pig_ID = form['Pig_ID'].value
                                    Data_type = form['Data_type'].value
                                    Data_type1 = re.sub("_"," ",Data_type)
                                    try:
                                        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
                                        cursor = connection.cursor()
                                    ""
                                    #cursor.execute("""SELECT File_storage_path,File_name FROM
monr_fileupload WHERE Pig_ID='%s' AND Data_type='%s'""%(Pig_ID,Data_type))
                                    ""
                                    output = cursor.fetchall()
                                    filepath = output[0][0]+"/"+output[0][1]
                                    try:
                                        os.remove(filepath)
                                        os.removedirs(output[0][0])
                                    except:
                                        pass
                                    ""
                                    #cursor.execute("""DELETE FROM monr_fileupload WHERE
Pig_ID='%s' AND Data_type='%s'""%(Pig_ID,Data_type))
                                    ""
                                    connection.commit()
                                    cursor.close()
                                    connection.close()
                                    print 'Content-Type: text/javascript\n'
                                    ""

```

```

        #print
        """document.getElementById('message1').innerHTML=\'"%s\'" file of Pig ID \'"%s\'"
        deleted successfully';loadFiles();"""%(Data_type1,Pig_ID)
        #except:
            #print 'Content-Type: text/javascript\n'
            #print """document.getElementById('message1').innerHTML=\'"%s\'"
file of Pig ID "%s" cannot be deleted at this moment. Please try again later or
contact data manager.';loadFiles();"""%(Data_type,Pig_ID)
            ##### This part of the code above has been
removed
#####
#####
#####
            ##### This part of the code below has been removed
#####
#####
#####
            #elif(action=="changedatatype"):#This part of the code has been
removed
            """
            Pig_ID = form['Pig_ID'].value
            old_Data_type = form['old_Data_type'].value
            Data_type1 = re.sub("_", " ",old_Data_type)
            new_Data_type = form['new_Data_type'].value
            Data_type2 = re.sub("_", " ",new_Data_type)
            output=''
            """
            #try:
            """
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
                cursor = connection.cursor()
            """
            #cursor.execute("""SELECT File_storage_path,File_name FROM
monr_fileupload WHERE Pig_ID='%s' AND Data_type='%s'"""%(Pig_ID,old_Data_type))
            """
            output = cursor.fetchall()
            oldfilepath = output[0][0]+"/"+output[0][1]
            File_storage_path = "/project/beilmanlab.umn.edu-
upload/"+studynome+"/Files/"+Pig_ID+"/"+new_Data_type
            newfilepath = File_storage_path+"/"+output[0][1]
            try:
                #Make sure the path exists
                if not os.path.exists(File_storage_path):
                    os.umask(0)
                    os.makedirs(File_storage_path,0771)
                else:
                    pass
            except:
                pass
            shutil.move(oldfilepath,newfilepath)
            try:
                os.removedirs(output[0][0])
            except:
                pass
            """

```

```

        #cursor.execute("""UPDATE monr_fileupload SET
Data_type='%s',File_storage_path='%s' WHERE Pig_ID='%s' AND
Data_type='%s'""%(new_Data_type,File_storage_path,Pig_ID,old_Data_type))
        """
        connection.commit()
        cursor.close()
        connection.close()
        print 'Content-Type: text/javascript\n'
        """
        #print """document.getElementById('message1').innerHTML='Data
type changed from \"%s\" to \"%s\" for Pig ID \"%s\"
successfully';loadFiles();""%(Data_type1,Data_type2,Pig_ID)
        #except:
        """
        try:
            connection.rollback()
            try:
                #Make sure the path exists
                if not os.path.exists(output[0][0]):
                    os.makedirs(output[0][0])
                else:
                    pass
            except:
                pass
            shutil.move(newfilepath,oldfilepath)
            os.removedirs(File_storage_path)
        except:
            pass
        print 'Content-Type: text/javascript\n'
        """
        #print """document.getElementById('message1').innerHTML='Data
type could not be changed from \"%s\" to \"%s\" for Pig ID \"%s\". Please contact
data manager.';loadFiles();""%(Data_type1,Data_type2,Pig_ID)
        ##### This part of the code above has been
removed
#####
#####
##### This part of the code below has been
removed
#####
#####
        #elif(action=="excel_export_load_ids"):
            #SelectNode = """<select name="pigids" id="pigids"
onchange="javascript:loadIndividualExcelFiles(this.value);">"""
            """
            try:
                options = '<option selected="selected"
value="Select">Select</option>'
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lodb)
                cursor = connection.cursor()
                """
                #cursor.execute("""SELECT Pig_ID FROM monr_identifier_table
ORDER BY Start_date""")
                """
                output = cursor.fetchall()

```

```

        for i in range(len(output)):
            for j in range(len(output[i])):
                options = options+'<option
value=\"'+output[i][j]+'\">'+output[i][j]+'</option>'
                """
            #SelectNode=SelectNode+options+"""/select>""
            """
            connection.commit()
            cursor.close()
            connection.close()
            print 'Content-Type: text/javascript\n'
            """
            #print
            """document.getElementById('holdpigids').innerHTML='%s';""%(SelectNode)
            """
            except:
                print 'Content-Type: text/javascript\n'
                """
                #print """document.getElementById('message1').innerHTML='There
was an error. Pig IDs for the study %s could not be loaded.';""%(studyname)
                ##### This part of the code above has been
removed
#####
#####
#####

#####
#####
#####
elif(action=="excel_export_load_overall"):
    singlefile = ''
    allfiles = ''
    uniqueid = ''
    File_size = 0
    try:
        datatype=form['datatype'].value
        dataoption=form['dataoption'].value
        row = "odd"
        allfiles = """<table class="excelfiles"><tbody><tr
class="holdHeaders"><td colspan="5">Data of all Subjects</td></tr><tr
class="holdHeaders"><td>Type of Data</td><!-- <td>File Name</td> -->\
                <td>File Size(kb)</td><td>Last update
time</td><td>Updated by</td>\
                <td>Download File</td></tr>""
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
        cursor = connection.cursor()
        if datatype=='Vitals':
            if dataoption=='all_data':
                dataoption='NOT LIKE'
            else:
                dataoption='LIKE'
            cursor.execute("""SELECT
Data_type,File_name,File_size,Last_update_time,Updated_by FROM
monr_allsubjects_export WHERE (Data_type='Study_Overview' OR Data_type LIKE
'Vitals%' OR Data_type LIKE 'Labs%' OR Data_type LIKE 'GCS%' OR Data_type LIKE
'Tegdata%') AND (Data_type='Study_Overview' OR Data_type %s '%events') ORDER BY
Data_type""%(dataoption))

```

```

else:
    datatype=datatype.lower()
    dataoption=dataoption.lower()
    cursor.execute("""SELECT
Data_type,File_name,File_size,Last_update_time,Updated_by FROM
monr_allsubjects_export WHERE Data_type='Study_Overview' OR Data_type LIKE
'%s_%s%' ORDER BY Data_type"""%(datatype,dataoption))
    #cursor.execute("""SELECT
Data_type,File_name,File_size,Last_update_time,Updated_by FROM
monr_allsubjects_export ORDER BY Data_type""")
    #cursor.execute("""SELECT
Data_type,File_name,File_size,Last_update_time,Updated_by FROM
monr_allsubjects_export WHERE (Data_type='Study_Overview' OR Data_type LIKE
'Vitals%' OR Data_type LIKE 'Labs%' OR Data_type LIKE 'GCS%' OR Data_type LIKE
'Tegdata%') AND Data_type LIKE '%events' ORDER BY Data_type""")
    #cursor.execute("""SELECT
Data_type,File_name,File_size,Last_update_time,Updated_by FROM
monr_allsubjects_export WHERE (Data_type='Study_Overview' OR Data_type LIKE
'Vitals%' OR Data_type LIKE 'Labs%' OR Data_type LIKE 'GCS%' OR Data_type LIKE
'Tegdata%') AND Data_type NOT LIKE '%events' ORDER BY Data_type""")
    #cursor.execute("""SELECT
Data_type,File_name,File_size,Last_update_time,Updated_by FROM
monr_allsubjects_export WHERE Data_type='Study_Overview' OR Data_type LIKE
'%s_%s%' ORDER BY Data_type""%( 'hnmr','urine'))
    output1 = cursor.fetchall()
    connection.commit()
    cursor.close()
    connection.close()
    if(len(output1)>0):
        for i in range(len(output1)):
            Data_type = output1[i][0]
            Data_type1 = re.sub("_", " ",Data_type)
            File_name = output1[i][1]
            File_size = output1[i][2]
            File_size = int(File_size/1000)
            if(File_size<1):
                File_size=1
            else:
                pass
            Last_update_time = str(output1[i][3])
            Updated_by = output1[i][4]
            if (row=="odd"):
                singlefile = singlefile+'<tr class="oddRow">'
                row = "even"
            else:
                singlefile = singlefile+'<tr class="evenRow">'
                row = "odd"
            uniqueid = Data_type
            singlefile = singlefile+"""<td style="text-
align:left">%s</td>\
                                <!-- <td style="text-
align:left">%s</td> -->\
                                <td class="numbers">%d</td>\
                                <td class="numbers">%s</td>\
                                <td>%s</td>\
                                <td class="clickoption"
onclick="javascript:downloadFile(\\'all_excel\\',\\'%s\\',\\'%s\\',\\'\\'\\');">Downl

```



```

oad</td></tr>""%(Data_type1,File_name,File_size,Last_update_time,Updated_by,study
name,Data_type)
        allfiles = allfiles+singlefile+""</tbody></table>""
    else:
        allfiles = ''
        print 'Content-Type: text/javascript\n'
        print
        ""document.getElementById('allexcelfileholder').innerHTML='%s';/*loadIDs();*/""%(
allfiles)
    except:
        print 'Content-Type: text/javascript\n'
        print ""document.getElementById('message1').innerHTML='An
error occurred. Please contact data manager';""
        #raise

#####
#####
#####
##### This part of the code below has been
removed
#####
#####
#####
        #elif(action=="excel_export_load_individual"):
            ""
            Pig_ID = str(form["Pig_ID"].value)
            singlefile = ''
            allfiles = ''
            uniqueid = ''
            File_size = 0
            try:
                row = "odd"
            ""
            #allfiles = ""<table class="excelfiles"><tbody><tr
class="holdHeaders"><td colspan="6">Data of Subject ID: %s</td></tr><tr
class="holdHeaders"><td>Type of Data</td><td>File Name</td>\
            #<td>File Size(kb)</td><td>Last update
time</td><td>Updated by</td>\
            #<td>Download File</td></tr>""%(Pig_ID)
            ""
            connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
            cursor = connection.cursor()
            ""
            #cursor.execute("""SELECT
Data_type,File_name,File_size,Last_update_time,Updated_by FROM
monr_individual_export WHERE Pig_ID='%s' ORDER BY Data_type""%(Pig_ID))
            ""
            output1 = cursor.fetchall()
            connection.commit()
            cursor.close()
            connection.close()
            if(len(output1)>0):
                for i in range(len(output1)):
                    Data_type = output1[i][0]
                    Data_type1 = re.sub("_"," ",Data_type)
                    File_name = output1[i][1]
                    File_size = output1[i][2]

```

```

        File_size = int(File_size/1000)
        if(File_size<1):
            File_size=1
        else:
            pass
        Last_update_time = str(output1[i][3])
        Updated_by = output1[i][4]
        if (row=="odd"):
            singlefile = singlefile+'<tr class="oddRow">'
            row = "even"
        else:
            singlefile = singlefile+'<tr class="evenRow">'
            row = "odd"
        uniqueid = Data_type
    """
    #singlefile = singlefile+""<td style="text-
align:left">%s</td>\
                                #<td style="text-
align:left">%s</td>\
                                #<td class="numbers">%d</td>\
                                #<td class="numbers">%s</td>\
                                #<td>%s</td>\
                                #<td class="clickoption"
onclick="javascript:downloadFile(\\'individual_excel\\',\\'%s\\',\\'%s\\',\\'%s\\'
);">Download</td></tr>""%(Data_type1,File_name,File_size,Last_update_time,Updated
_by,studyname,Data_type,Pig_ID)
    #allfiles = allfiles+singlefile+""</tbody></table>""
    """
    else:
        allfiles = ''
    print 'Content-Type: text/javascript\n'
    """
    #print
    """document.getElementById('individualexcelfileholder').innerHTML='%s';""%(allfil
es)
    """
    except:
        print 'Content-Type: text/javascript\n'
    """
    #print """document.getElementById('message3').innerHTML='An
error occurred. Please contact data manager';""
    ##### This part of the code above has been
removed
#####
#####
##### This part of the code below has been
removed
#####
#####
#####
    #elif(action=="loadGroupsAdvancedExport"):
    """
    try:
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
        cursor = connection.cursor()
    """

```

```

        #cursor.execute("""SELECT DISTINCT Pig_group FROM
monr_identifier_table ORDER BY Pig_group""")
        """
        output=cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        row="oddRow"
        """
        #display_data="""<tr class="%s">td
class="optionscolumn"><label for="allgroups">All Groups</label></td><td
class="inputcolumn"><input type="checkbox" name="allgroups" id="allgroups"
onclick="javascript:selectAll(\\'groups\\');" /></td></tr>""%(row)
        """
        for i in output:
            Pig_group=i[0]
            name_and_id="Group_%s"%(Pig_group)
            if Pig_group==1:
                Display="Group-1 (C1)"
            elif Pig_group==2:
                Display="Group-2 (C2)"
            elif Pig_group==3:
                Display="Group-3 (E1)"
            elif Pig_group==4:
                Display="Group-4 (E2)"
            else:
                Display="Group-%s"%(Pig_group)
            if (row=="oddRow"):
                row = "evenRow"
            else:
                row = "oddRow"
        """
        #display_data=display_data+""<tr class="%s">td
class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne(\\'groups\\');"
/></td></tr>""%(row,name_and_id,Display,name_and_id)
        #display_table="""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><!-- <tr><th colspan="2">Select Groups</th></tr> --
>%s</tbody></table></td></tr></table></div>""%(display_data)
        #print 'Content-Type: text/javascript\n'
        #print
        """document.getElementById('groups').innerHTML='%s';""%(display_table)
        #except:
            #print 'Content-Type: text/javascript\n'
            #print """document.getElementById('subjects').innerHTML='An
error occured. Please contact data manager.';""
            ##raise
            ##### This part of the code above has been
removed
#####
#####
#####

#####
#####
#####

```

```

elif(action=="loadSubjectsAdvancedExport"):
    try:
        groups=form['groups'].value
        request_from_groups=''
        request_from_groups=groups.replace('Group_', 'Pig_group=')
        request_from_groups=request_from_groups.replace('-', ' OR ')
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        set_character_set=""
        cursor.execute(set_character_set)
        cursor.execute("""SELECT DISTINCT Pig_ID,Pig_group FROM
monr_identifier_table WHERE (%s) AND Datainput_status='Finished' ORDER BY
Pig_group,Start_date""%(request_from_groups))
        output=cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        row="oddRow"
        display_data=""
        class="optionscolumn"><label for="allsubjects">All Subjects</label></td><td
class="inputcolumn"><input type="checkbox" name="allsubjects" id="allsubjects"
onclick="javascript:selectAll(\\'subjects\\');" /></td></tr>""%(row)
        #display_data=display_data+""
        colspan="2">%s</td></tr>""%(data_requested)
        previous_group=''
        for i in output:
            Pig_ID=i[0]
            Pig_group=i[1]
            Pig_group="Group-"+str(Pig_group)
            if not (previous_group==Pig_group):
                display_data=display_data+""
                class="subheading"><td colspan="2">%s</td></tr>""%(Pig_group)
            else:
                pass
            previous_group=Pig_group
            name_and_id="subjectid_%s"%(Pig_ID)
            Display="Subject-%s"%(Pig_ID)
            if (row=="oddRow"):
                row = "evenRow"
            else:
                row = "oddRow"
            display_data=display_data+""
            class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne(\\'subjects\\');"
/></td></tr>""%(row,Pig_group,name_and_id,Display,name_and_id,name_and_id)
            display_table=""
            class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><!-- <tr><th colspan="2">Select Subjects</th></tr> --
>%s</tbody></table></td></tr></table></td></tr></table></div>""%(display_data)
            print 'Content-Type: text/javascript\n'
            print
            ""document.getElementById('subjects').innerHTML='%s';""%(display_table)
    except:
        print 'Content-Type: text/javascript\n'
        print ""document.getElementById('subjects').innerHTML='An
error occured. Please contact data manager.';""

```

```

#raise

#####
#####
#####
elif(action=="loadVitalsVariablesAdvancedExport"):
    try:
        vitals_data_tables=form['vitals_tables'].value
        request_from_tables=[]
        vitals_columns_tables=[]
        vitals_columns=[]
        request_from_tables=vitals_data_tables.split('-')
        avoid_columns=['Pig_ID', 'Event_name']
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
        cursor = connection.cursor()
        set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""
        cursor.execute(set_character_set)
        row="oddRow"
        display_data=""<tr class="%s"><td
class="optionscolumn"><label for="allvitalsvariables">All vitals
variables</label></td><td class="inputcolumn"><input type="checkbox"
name="allvitalsvariables" id="allvitalsvariables"
onclick="javascript:selectAll(\\'vitalsvariables\\');" /></td></tr>""%(row)
        for requestedtable in request_from_tables:
            cursor.execute("""SELECT column_name FROM
information_schema.columns WHERE table_name='%s';""%(requestedtable))
            output=cursor.fetchall()
            if len(output)>0:
                if requestedtable=='monr_vitals_manual_10_min':
                    data_requested='Vitals and Gases'
                    display_data=display_data+""<tr
class="subheading"><td colspan="2">%s</td></tr>""%(data_requested)
                    elif not
(requestedtable=='monr_vitals_manual_one_hour' or
requestedtable=='monr_vitals_manual_10_min'):
#data_requested=(requestedtable.split('_')[1]).capitalize()+
'+(requestedtable.split('_')[2])
                    data_requested=(
".join(requestedtable.split('_')).replace('test ', '')
                    data_requested=data_requested.replace('monr ', '')
                    data_requested=data_requested.capitalize()
                    display_data=display_data+""<tr
class="subheading"><td colspan="2">%s</td></tr>""%(data_requested)
                    else:
                        data_requested='Vitals and Gases'
                        other_columns=[]
                        for i in range(len(output)):
                            column_name=output[i][0]
                            if not (column_name in avoid_columns):
                                vitals_columns_tables.append(column_name+'|'+requestedtable)
                                avoid_columns.append(column_name)
                                if
(requestedtable=='monr_vitals_manual_one_hour' or
requestedtable=='monr_vitals_manual_10_min'):

```

```

vitals_columns.append(column_name+'|'+requestedtable)
    else:
        other_columns.append(column_name)
    else:
        pass
if(requestedtable=='monr_vitals_manual_one_hour'):
    vitals_columns.sort()
    for column_table_stored in vitals_columns:
        if (row=="oddRow"):
            row = "evenRow"
        else:
            row = "oddRow"
        Display=column_table_stored.split('|')[0]
        Display=Display.replace('_', ' ')
        data_name=data_requested

name_and_id=column_table_stored.split('|')[0]+'|'+column_table_stored.split('|')[1
]
        display_data=display_data+""<tr
class="%s"><td title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne(\\'vitalsvariables\\');"
/></td></tr>""%(row,data_name,name_and_id,Display,name_and_id,name_and_id)
        elif
not(requestedtable=='monr_vitals_manual_one_hour'):
    other_columns.sort()
    for column_stored in other_columns:
        if (row=="oddRow"):
            row = "evenRow"
        else:
            row = "oddRow"
        Display=column_stored
        Display=Display.replace('_', ' ')
        data_name=data_requested
        name_and_id=column_stored+'|'+requestedtable
        display_data=display_data+""<tr
class="%s"><td title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne(\\'vitalsvariables\\');"
/></td></tr>""%(row,data_name,name_and_id,Display,name_and_id,name_and_id)
        else:
            pass
    else:
        pass
    #vitals_columns_tables.sort()
    connection.commit()
    cursor.close()
    connection.close()
    if not (len(vitals_columns_tables))<1:
        display_table=""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><!-- <tr><th colspan="2">Select Subjects</th></tr> --
>%s</tbody></table></td></tr></table></div>""%(display_data)
        print 'Content-Type: text/javascript\n'
        print
        ""document.getElementById('vitalsvariablesselect').innerHTML='%s';""%(display_ta
ble)

```

```

else:
    print 'Content-Type: text/javascript\n'
    print
    """document.getElementById('vitalsvariablesselect').innerHTML='<div
class="innerdiv"><table class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><tr class="oddRow"><td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> No data are available for the
selection</span></td></tr></tbody></table></td></tr></table></div>';"""
    except:
        print 'Content-Type: text/javascript\n'
        print
        """document.getElementById('vitalsvariablesselect').innerHTML='An error occured.
Please contact data manager.';"""
        #raise

#####
#####
#####
elif(action=="loadHNMRVariablesAdvancedExport"):
    try:
        datatype=''
        hnmr_data_tables=form['hnmr_tables'].value
        request_from_tables=[]
        hnmr_columns_tables=[]
        request_from_tables=hnmr_data_tables.split('-')
        avoid_columns=['Pig_ID', 'Event_name', 'Normalization']
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
        cursor = connection.cursor()
        set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'"""
        cursor.execute(set_character_set)
        row="oddRow"
        display_data=""<tr class="%s"><td
class="optionscolumn"><label for="allhnmrvariables">All hnmr
variables</label></td><td class="inputcolumn"><input type="checkbox"
name="allhnmrvariables" id="allhnmrvariables"
onclick="javascript:selectAll('\\hnmrvariables\\');" /></td></tr>""%(row)
        output=''
        alltables=[]
        for i in range(len(request_from_tables)):
            requestedtable=request_from_tables[i].split('|')[0]
            alltables.append(requestedtable)
        alltables=set(alltables)
        for the_table in alltables:
            datatype=''
            totaldatatypes=0
            for table_and_datatype in request_from_tables:
                if re.search(the_table,table_and_datatype):
                    totaldatatypes=totaldatatypes+1
                    if totaldatatypes==1:
                        datatype=datatype+table_and_datatype.split('|')[1]
                        else:
                            datatype=datatype+'+'+table_and_datatype.split('|')[1]
                        else:
                            pass
                            requestedtable=the_table

```

```

        #data_requested=("
".join(requestedtable.split('_')).capitalize()
        data_requested=("
".join(requestedtable.split('_')).replace('test ',''))
        data_requested=data_requested.replace('monr ','')
        data_requested=data_requested.capitalize()
        datatype_compile=re.compile('No')
        datatype_requested=datatype_compile.sub('Raw
Data',datatype,count=1)
        datatype_compile=re.compile('Yes')
        datatype_requested=datatype_compile.sub('Normalized
Data',datatype_requested,count=1)
        datatype_compile=re.compile('_')
        datatype_requested=datatype_compile.sub('
',datatype_requested)
        cursor.execute("""SELECT column_name FROM
information_schema.columns WHERE table_name='%s';"""%(requestedtable))
        output=cursor.fetchall()
        if len(output)>0:
            display_data=display_data+""<tr
class="subheading"><td colspan="2">%s |
%s</td></tr>""%(data_requested,datatype_requested)
            for i in range(len(output)):
                column_name=output[i][0]
                if not (column_name in avoid_columns):
                    hnmr_columns_tables.append(column_name+'|'+requestedtable+'|'+datatype)
                    #avoid_columns.append(column_name)
                    if (row=="oddRow"):
                        row = "evenRow"
                    else:
                        row = "oddRow"
                    Display=column_name
                    Display=Display.replace('_', ' ')
                    data_name_datatype="%s |
%s"%(data_requested,datatype_requested)
                    name_and_id=column_name+'|'+requestedtable+'|'+datatype
                    display_data=display_data+""<tr
class="%s"><td title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne(\\'hnmrvariables\\');"
/></td></tr>""%(row,data_name_datatype,name_and_id,Display,name_and_id,name_and_i
d)
                    else:
                        pass
                    else:
                        pass
                    #else:
                    #pass
                    #hnmr_columns_tables.sort()
                    connection.commit()
                    cursor.close()
                    connection.close()
                    if not (len(hnmr_columns_tables))<1:
                        display_table=""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table

```



```

class="selection"><tbody><!-- <tr><th colspan="2">Select Subjects</th></tr> --
>%s</tbody></table></td></tr></table></div>""%(display_data)
        print 'Content-Type: text/javascript\n'
        print
"""document.getElementById('hnmrvariablesselect').innerHTML='%s';""%(display_tabl
e)
        else:
            print 'Content-Type: text/javascript\n'
            print
"""document.getElementById('hnmrvariablesselect').innerHTML='<div
class="innerdiv"><table class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><tr class="oddRow"><td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> No data are available for the
selection</span></td></tr></tbody></table></td></tr></table></div>';""
        except:
            print 'Content-Type: text/javascript\n'
            print
"""document.getElementById('hnmrvariablesselect').innerHTML='An error ocured.
Please contact data manager.';""
            #raise

#####
#####
#####
        elif(action=="loadPNMRVariablesAdvancedExport"):
            try:
                datatype=''
                pnmr_data_tables=form['pnmr_tables'].value
                request_from_tables=[]
                pnmr_columns_tables=[]
                request_from_tables=pnmr_data_tables.split('-')

avoid_columns=['Pig_ID','Event_name','Normalization','Value_type']
                connection = sql.connect(host=lhost,port=lport,user=user,
passwd=lpasswd,db=ldb)
                cursor = connection.cursor()
                set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""
                cursor.execute(set_character_set)
                row="oddRow"
                display_data=""<tr class="%s"><td
class="optionscolumn"><label for="allpnmrvariables">All pnmr
variables</label></td><td class="inputcolumn"><input type="checkbox"
name="allpnmrvariables" id="allpnmrvariables"
onclick="javascript:selectAll('\pnmrvariables\');" /></td></tr>""%(row)
                output=''
                alltables=[]
                for i in range(len(request_from_tables)):
                    requestedtable=request_from_tables[i].split('|')[0]
                    alltables.append(requestedtable)
                alltables=set(alltables)
                for the_table in alltables:
                    datatype=''
                    totaldatatypes=0
                    for table_and_datatype in request_from_tables:
                        if re.search(the_table,table_and_datatype):
                            totaldatatypes=totaldatatypes+1
                            if totaldatatypes==1:

```

```

datatype=datatype+table_and_datatype.split('|')[1]
    else:

datatype=datatype+' '+table_and_datatype.split('|')[1]
    else:
        pass
        requestedtable=the_table
        #data_requested=("
".join(requestedtable.split('_')).capitalize()
        data_requested=("
".join(requestedtable.split('_')).replace('test ', ''))
        data_requested=data_requested.replace('monr ', '')
        data_requested=data_requested.capitalize()
        datatype_compile=re.compile('No')
        datatype_requested=datatype_compile.sub('Raw
Data', datatype, count=1)
        datatype_compile=re.compile('Yes')
        datatype_requested=datatype_compile.sub('Normalized
Data', datatype_requested, count=1)
        datatype_compile=re.compile('_')
        datatype_requested=datatype_compile.sub('
', datatype_requested)
        cursor.execute("""SELECT column_name FROM
information_schema.columns WHERE table_name='%s';"""%(requestedtable))
        output=cursor.fetchall()
        if len(output)>0:
            display_data=display_data+"""<tr
class="subheading"><td colspan="2">%s |
%s</td></tr>""%(data_requested, datatype_requested)
            for i in range(len(output)):
                column_name=output[i][0]
                if not (column_name in avoid_columns):
                    pnmr_columns_tables.append(column_name+' '+requestedtable+' '+datatype)
                    #avoid_columns.append(column_name)
                    if (row=="oddRow"):
                        row = "evenRow"
                    else:
                        row = "oddRow"
                    if re.search(r'\d+', column_name):
                        Display="Region "+column_name+" integral"
                    else:
                        Display=column_name
                    Display=Display.replace('_', ' ')
                    data_name_datatype="%s |
%s""%(data_requested, datatype_requested)

                    name_and_id=column_name+' '+requestedtable+' '+datatype
                    display_data=display_data+"""<tr
class="%s"><td title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne(\\'pnmrvariables\\');"
/></td></tr>""%(row, data_name_datatype, name_and_id, Display, name_and_id, name_and_i
d)
                else:
                    pass
            else:

```

```

        pass
        #pnmr_columns_tables.sort()
        connection.commit()
        cursor.close()
        connection.close()
        if not (len(pnmr_columns_tables))<1:
            display_table="""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><!-- <tr><th colspan="2">Select Subjects</th></tr> --
>%s</tbody></table></td></tr></table></div>""%(display_data)
            print 'Content-Type: text/javascript\n'
            print
            """document.getElementById('pnmrvariablesselect').innerHTML='%s';""%(display_table)
        else:
            print 'Content-Type: text/javascript\n'
            print
            """document.getElementById('pnmrvariablesselect').innerHTML='<div
class="innerdiv"><table class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><tr class="oddRow"><td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> No data are available for the
selection</span></td></tr></tbody></table></td></tr></table></div>';"""
            except:
                print 'Content-Type: text/javascript\n'
                print
                """document.getElementById('pnmrvariablesselect').innerHTML='An error ocured.
Please contact data manager.';"""
                #raise

#####
#####
#####
            elif(action=="loadBinningVariablesAdvancedExport"):
                try:
                    binning_data_tables=form['binning_tables'].value
                    request_from_tables=[]
                    binning_columns_tables=[]
                    request_from_tables=binning_data_tables.split('-')
                    avoid_columns=['Pig_ID', 'Event_name']
                    connection = sql.connect(host=lhost,port=lport,user=luser,
                    passwd=lpasswd,db=lhb)
                    cursor = connection.cursor()
                    set_character_set="""SET NAMES 'utf8' COLLATE 'utf8_bin'"""
                    cursor.execute(set_character_set)
                    row="oddRow"
                    display_data="""<tr class="%s"><td
class="optionscolumn"><label for="allbinningvariables">All binning
variables</label></td><td class="inputcolumn"><input type="checkbox"
name="allbinningvariables" id="allbinningvariables"
onclick="javascript:selectAll(\\'binningvariables\\');" /></td></tr>""%(row)
                    for requestedtable in request_from_tables:
                        cursor.execute("""SELECT column_name FROM
information_schema.columns WHERE table_name='%s';""%(requestedtable))
                        output=cursor.fetchall()

#data_requested=(requestedtable.split('_')[1]).capitalize()+
'+(requestedtable.split('_')[2]).capitalize()

```

```

        #data_requested=("
".join(requestedtable.split('_')).capitalize()
        data_requested=("
".join(requestedtable.split('_')).replace('test ', ''))
        data_requested=data_requested.replace('monr ', '')
        data_requested=data_requested.capitalize()
        if len(output)>0:
            display_data=display_data+""<tr
class="subheading"><td colspan="2">%s</td></tr>""%(data_requested)
            for i in range(len(output)):
                column_name=output[i][0]
                if not (column_name in avoid_columns):

binning_columns_tables.append(column_name+'|'+requestedtable)
                #avoid_columns.append(column_name)
                if (row=="oddRow"):
                    row = "evenRow"
                else:
                    row = "oddRow"
                Display=column_name
                Display=Display.replace('_', ' ')
                data_name=data_requested
                name_and_id=column_name+'|'+requestedtable
                display_data=display_data+""<tr
class="%s"><td title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne(\\'binningvariables\\');"
/></td></tr>""%(row,data_name,name_and_id,Display,name_and_id,name_and_id)
                    else:
                        pass
                else:
                    pass
                #binning_columns_tables.sort()
                connection.commit()
                cursor.close()
                connection.close()
                if not (len(binning_columns_tables)<1:
                    display_table=""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><!-- <tr><th colspan="2">Select Subjects</th></tr> --
>%s</tbody></table></td></tr></table></div>""%(display_data)
                    print 'Content-Type: text/javascript\n'
                    print
                    ""document.getElementById('binningvariablesselect').innerHTML='%s';""%(display_t
able)
                else:
                    print 'Content-Type: text/javascript\n'
                    print
                    ""document.getElementById('binningvariablesselect').innerHTML='<div
class="innerdiv"><table class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><tr class="oddRow"><td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> No data are available for the
selection</span></td></tr></tbody></table></td></tr></table></div>';""
                    except:
                        print 'Content-Type: text/javascript\n'
                        print
                        ""document.getElementById('binningvariablesselect').innerHTML='An error ocured.
Please contact data manager.';""

```

```

#raise

#####
#####
#####
elif(action=="loadEvIndVarAdvancedExport"):
    try:
        identifier_columns=[]
        request_from_tables=['monr_identifier_table']

#avoid_columns=['Pig_ID','Event_name','Pig_group']#, 'OR_fluids','LR_given']
        avoid_columns=['Pig_ID','Event_name']#, 'OR_fluids','LR_given']

add_columns=['Crystalloids_given','Time_till_death_from_baseline']
        connection = sql.connect(host=lhost,port=lport,user=luser,
        passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""
        cursor.execute(set_character_set)
        row="oddRow"
        display_data=""<tr class="%s"><td
class="optionscolumn"><label for="alleventindependent">All
variables</label></td><td class="inputcolumn"><input type="checkbox"
name="alleventindependent" id="alleventindependent"
onclick="javascript:selectAll(\\'eventindependent\\');" /></td></tr>""%(row)
        for requestedtable in request_from_tables:
            cursor.execute("""SELECT column_name FROM
information_schema.columns WHERE table_name='%s';""%(requestedtable))
            output=cursor.fetchall()
            #data_requested=(
".join(requestedtable.split('_')).capitalize()
            data_requested=(
".join(requestedtable.split('_')).replace('test ','')
            data_requested=data_requested.replace('monr ','')
            data_requested=data_requested.capitalize()
            if len(output)>0:
                display_data=display_data+""<tr
class="subheading"><td colspan="2">%s</td></tr>""%(data_requested)
                for i in range(len(output)):
                    column_name=output[i][0]
                    if not (column_name in avoid_columns):
                        identifier_columns.append(column_name+'|'+requestedtable)
                        avoid_columns.append(column_name)
                    else:
                        pass
                #for j in range(len(add_columns)):
                #column_name=add_columns[j]
                #if not (column_name in avoid_columns):
#identifier_columns.append(column_name+'|'+requestedtable)
                #avoid_columns.append(column_name)
                #else:
                #pass
            identifier_columns.sort()
            for column_table_stored in identifier_columns:
                if (row=="oddRow"):
                    row = "evenRow"

```

```

else:
    row = "oddRow"
    column_name=column_table_stored.split('|')[0]
    Display=(column_name.replace('_', '
')).capitalize()

    if re.search(r'Died',Display):
        Display='Died (Yes or No)'
    elif re.search(r'Start date',Display):
        Display='Experiment start date'
    elif re.search(r'End date',Display):
        Display='Experiment end date'
    elif re.search(r'Comments',Display):
        Display='Overall comments'
    data_name=data_requested
    name_and_id=column_name+'|'+requestedtable
    display_data=display_data+""<tr class="%s"><td
title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne('\eventindependent\');"
/></td></tr>""%(row,data_name,name_and_id,Display,name_and_id,name_and_id)
    else:
        pass
        connection.commit()
        cursor.close()
        connection.close()
        if not (len(identifier_columns))<1:
            display_table=""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody>%s</tbody></table></td></tr></table></div>""%(display_da
ta)

            print 'Content-Type: text/javascript\n'
            print
            ""document.getElementById('eventindependentvartable').innerHTML='%s';""%(displa
y_table)

        else:
            print 'Content-Type: text/javascript\n'
            print
            ""document.getElementById('eventindependentvartable').innerHTML='<div
class="innerdiv"><table class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><tr class="oddRow"><td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> No data are available for the
selection</span></td></tr></tbody></table></td></tr></table></div>';""

            except:
                print 'Content-Type: text/javascript\n'
                print
                ""document.getElementById('eventindependentvartable').innerHTML='An error
occured. Please contact data manager.';""
                #raise

#####
#####
#####
elif(action=="showEventsAdvancedExport"):
    try:
        request_from_tables=['monr_event_definitions']
        Event_names=[]
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)

```

```

        cursor = connection.cursor()
        set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""
        cursor.execute(set_character_set)
        row="evenRow"
        display_data=''
        for requestedtable in request_from_tables:
            cursor.execute("""SELECT Event_name,Old_Event_name FROM
`s` WHERE Event_name!=99"""%(requestedtable))
            output=cursor.fetchall()
            data_requested='Events'
            if len(output)>0:
                display_data=display_data+""<tr
class="subheading"><td colspan="2">%s</td></tr>""%(data_requested)
                for eventname,oldeventname in output:
                    column_name=str(eventname)
                    Event_names.append(column_name+'|'+requestedtable)
                    #avoid_columns.append(column_name)
                    if (row=="oddRow"):
                        row = "evenRow"
                    else:
                        row = "oddRow"
                    Display=column_name+' ('+oldeventname+)'
                    Display=Display.replace('_', ' ')
                    data_name=data_requested
                    name_and_id=column_name+'|'+requestedtable
                    display_data=display_data+""<tr class="%s"><td
title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne(\\'eventnames\\');"
/></td></tr>""%(row,data_name,name_and_id,Display,name_and_id,name_and_id)
                else:
                    pass
            #binning_columns_tables.sort()
            connection.commit()
            cursor.close()
            connection.close()
            if not (len(Event_names))<1:
                display_table=""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody>%s</tbody></table></td></tr></table></div>""%(display_da
ta)
                print 'Content-Type: text/javascript\n'
                print
                ""document.getElementById('eventchoice').innerHTML='%s';""%(display_table)
            else:
                print 'Content-Type: text/javascript\n'
                print
                ""document.getElementById('eventchoice').innerHTML='<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody><tr class="oddRow"><td class="inform"><span
class="arrow">&larr;</span><span class="arrowmsg"> No data are available for the
selection</span></td></tr></tbody></table></td></tr></table></div>';""
            except:
                print 'Content-Type: text/javascript\n'
                print ""document.getElementById('eventchoice').innerHTML='An
error occured. Please contact data manager.';""
                #raise

```

```

#####
#####
#####
elif(action=="queryAdvancedExport"):
    try:
        requestedvariables=form['requestedvariables'].value
        # Sorting Dictionaries
        # (IMHO) the simplest approach:
        def sortedDictValues1(adict):
            items = adict.items()
            items.sort()
            return [value for key, value in items]

        # an alternative implementation, which
        # happens to run a bit faster for large
        # dictionaries on my machine:
        def sortedDictValues2(adict):
            keys = adict.keys()
            keys.sort()
            return [dict[key] for key in keys]

        # a further slight speed-up on my box
        # is to map a bound-method:
        def sortedDictValues3(adict):
            keys = adict.keys()
            keys.sort()
            return map(adict.get, keys)

        #The def below makes a dictionary for each table in the query
and stores the temp table values like t1, t2 etc, useful for mysql queries
        tablenumber=0
        def returnnewtablename(thetable):
            global tablenumber,newtablename,maptables,match_tables
            if not thetable in maptables:
                tablenumber=tablenumber+1
                newtablename='`t'+str(tablenumber)+'`'
                maptables[thetable]=newtablename
                match_tables[newtablename]=thetable
            return newtablename
        else:
            return maptables[thetable]

#####
#####
        connection = sql.connect(host=lhost,port=lport,user=user,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SET NAMES 'utf8' COLLATE 'utf8_bin'""")

#requestedvariables='groupsform|Group_1:groupsform|Group_2:subjectsform|allsubject
s:subjectsform|subjectid_051809:subjectsform|subjectid_111609:subjectsform|subject
id_120709:subjectsform|subjectid_110909:vitalsdataselectform|monr_vitals_manual_on
e_hour-
monr_vitals_manual_10_min:vitalsdataselectform|monr_lab_values:vitalsvariablesform
|allvitalsvariables:vitalsvariablesform|Art_Dia|monr_vitals_manual_10_min:vitalsva
riablesform|Art_Sys|monr_vitals_manual_10_min:vitalsvariablesform|Arterial_O2_sat|
monr_vitals_manual_one_hour:vitalsvariablesform|Arterial_PC02|monr_vitals_manual_o
ne_hour:vitalsvariablesform|Arterial_P02|monr_vitals_manual_one_hour:vitalsvariabl

```



```

esform|Arterial_pH|monr_vitals_manual_one_hour:vitalsvariablesform|BIS|monr_vitals
_manual_10_min:vitalsvariablesform|Base_deficit|monr_vitals_manual_one_hour:vitals
variablesform|Bladder_pressure|monr_vitals_manual_one_hour:vitalsvariablesform|Cal
cium|monr_vitals_manual_one_hour:vitalsvariablesform|Cardiac_output|monr_vitals_ma
nual_one_hour:vitalsvariablesform|Chesttube_output|monr_vitals_manual_one_hour:vit
alsvariablesform|DO2|monr_vitals_manual_one_hour:vitalsvariablesform|Date_of_exper
iment|monr_vitals_manual_one_hour:vitalsvariablesform|ERO2|monr_vitals_manual_one
hour:vitalsvariablesform|Event_comments|monr_vitals_manual_10_min:vitalsvariablesf
orm|FiO2|monr_vitals_manual_10_min:vitalsvariablesform|Glucose|monr_vitals_manual
_one_hour:vitalsvariablesform|Hb|monr_vitals_manual_one_hour:vitalsvariablesform|He
art_Rate|monr_vitals_manual_10_min:vitalsvariablesform|Lactate|monr_vitals_manual
_one_hour:vitalsvariablesform|MeanArtPres|monr_vitals_manual_10_min:vitalsvariable
sform|PA_Dia|monr_vitals_manual_10_min:vitalsvariablesform|PA_Sys|monr_vitals_manua
l_10_min:vitalsvariablesform|Potassium|monr_vitals_manual_one_hour:vitalsvariables
form|Propofol|monr_vitals_manual_10_min:vitalsvariablesform|Sodium|monr_vitals_ma
nual_one_hour:vitalsvariablesform|Specificgravity_urine|monr_vitals_manual_one_hou
r:vitalsvariablesform|StO2_15mm|monr_vitals_manual_10_min:vitalsvariablesform|StO2
_multidepth|monr_vitals_manual_10_min:vitalsvariablesform|THI_15mm|monr_vitals_manu
al_10_min:vitalsvariablesform|THI_multidepth|monr_vitals_manual_10_min:vitalsvaria
blesform|Temperature|monr_vitals_manual_one_hour:vitalsvariablesform|Time_of_exper
iment|monr_vitals_manual_one_hour:vitalsvariablesform|Urine_output|monr_vitals_ma
nual_one_hour:vitalsvariablesform|VO2|monr_vitals_manual_one_hour:vitalsvariablesfo
rm|Venous_O2_sat|monr_vitals_manual_one_hour:vitalsvariablesform|Venous_PO2|monr_v
itals_manual_one_hour:vitalsvariablesform|Wedge_pressure|monr_vitals_manual_one_ho
ur:vitalsvariablesform|ALT|monr_lab_values:vitalsvariablesform|AST|monr_lab_values
:vitalsvariablesform|Albumin|monr_lab_values:vitalsvariablesform|Alk_phosphatase|m
onr_lab_values:vitalsvariablesform|Bilirubin_total|monr_lab_values:vitalsvariables
form|CK|monr_lab_values:vitalsvariablesform|Creatinine|monr_lab_values:vitalsvaria
blesform|LD|monr_lab_values:vitalsvariablesform|Platelet_count|monr_lab_values:vit
alsvariablesform|Protein_total|monr_lab_values:vitalsvariablesform|Urea_nitrogen|m
onr_lab_values:vitalsvariablesform|Urine_creatinine|monr_lab_values:vitalsvariable
sform|Urine_osmolality|monr_lab_values:datapointsselectform|allevents'
varlist=requestedvariables.split(':')
formnvarmap={}

```

```

varformpairs={'vitalsvariablesform':'vitalsdataselectform','hnmrvariablesform':'hnm
rdataselectform','pnmrvariablesform':'pnmrdataselectform','binningvariablesform':
'binningdataselectform'}

```

```

whereformpairs={'subjectsform':'groupsform','datapointsselectform':'eventchoicefor
m'}

```

```

Pig_groups='(`t1`.`Pig_group`)'
Pig_IDs='(`t1`.`Pig_ID`)'
identiervariables=''
allvariables=[]
alltables=[]
maptables={}
match_tables={}
newtablename=''
vitalsvariables=''
hnmrvariables=''
hnmrnormalization=''
pnmrvariables=''
pnmrnormalization=''
pnmrintegralonly=''
binningvariables=''
eventindependentvariables=''

```

```

#formpairs=[['vitalsvariablesform','vitalsdataselctform'],['subjectsform','groups
form']]
        #print
sorted(formpairs,key=operator.itemgetter(1),reverse=True)
        #print map(operator.itemgetter(1),formpairs)
        #for a in formpairs:
            #print a
        for thestring in varlist:
            formname=''
            queryvarstring=[]
            formnvariablelist=thestring.split('|')
            formname=formnvariablelist[0]
            for i in range(1,len(formnvariablelist)):
                queryvarstring.append(formnvariablelist[i])
            queryvarstring='|'.join(queryvarstring)
            if formname in formnvarmap:

formnvarmap[formname]=formnvarmap[formname]+':'+queryvarstring
            else:
                formnvarmap[formname]=queryvarstring
            #print formnvarmap
            #print sortedDictValues3(formnvarmap)
            #for theform in whichformpairs:
            if ('subjectsform' in formnvarmap) and (len(formnvarmap)==3):
                if re.search(r'all',formnvarmap['subjectsform']):
                    if re.search(r'all',formnvarmap['groupsform']):
                        pass
                    else:

Pig_groups=(formnvarmap['groupsform']).replace('Group_', '')
                Pig_groups_list=Pig_groups.split(':')
                temp_Pig_groups_list=[]
                for Pig_group in Pig_groups_list:

temp_Pig_groups_list.append("`t1`.`Pig_group`="+Pig_group)
                Pig_groups=' OR '.join(temp_Pig_groups_list)
                Pig_groups=' ('+Pig_groups+' ) '
            else:

Pig_IDs=(formnvarmap['subjectsform']).replace('subjectid_', '')
                Pig_IDs_list=Pig_IDs.split(':')
                temp_Pig_IDs_list=[]
                for Pig_ID in Pig_IDs_list:

temp_Pig_IDs_list.append("`t1`.`Pig_ID`=\ '"+Pig_ID+"\`")
                Pig_IDs=' OR '.join(temp_Pig_IDs_list)
                Pig_IDs=' ('+Pig_IDs+' ) '
                allvariables.append('Pig_ID')
                allvariables.append('Pig_group')
                alltables.append('monr_identifier_table')
                identifiervariables=('`t1`.`Pig_ID`,`t1`.`Pig_group`)

tempsubjecttable=returnnewtablename('monr_identifier_table')
            elif ('subjectsform' in formnvarmap):
                if re.search(r'all',formnvarmap['subjectsform']):
                    if re.search(r'all',formnvarmap['groupsform']):

```

```

        pass
    else:
        Pig_groups=(formnvarmap['groupsform']).replace('Group_', '')
        Pig_groups_list=Pig_groups.split(':')
        temp_Pig_groups_list=[]
        for Pig_group in Pig_groups_list:
            temp_Pig_groups_list.append("`t1`.`Pig_group`="+Pig_group)
            Pig_groups=' OR '.join(temp_Pig_groups_list)
            Pig_groups=' ('+Pig_groups+' ) '
        else:
            Pig_IDs=(formnvarmap['subjectsform']).replace('subjectid_', '')
            Pig_IDs_list=Pig_IDs.split(':')
            temp_Pig_IDs_list=[]
            for Pig_ID in Pig_IDs_list:
                temp_Pig_IDs_list.append("`t1`.`Pig_ID`='"+Pig_ID+"'")
                Pig_IDs=' OR '.join(temp_Pig_IDs_list)
                Pig_IDs=' ('+Pig_IDs+' ) '
                allvariables.append('Pig_ID')
                #allvariables.append('Pig_group')
                alltables.append('monr_identifier_table')
                #identifiervariables=('`t1`.`Pig_ID`,`t1`.`Pig_group`')
                identifiervariables=('`t1`.`Pig_ID`')

            tempsubjecttable=returnnewtablename('monr_identifier_table')
            else:
                pass
                #print identifiervariables
                #print Pig_groups
                #print Pig_IDs
                if 'eventindependentvartable' in formnvarmap:
                    if
re.search(r'^alleventindependent', formnvarmap['eventindependentvartable']):
eventindependentvariables=formnvarmap['eventindependentvartable']
eventindependentvariables=eventindependentvariables.split(':')
temp_eventindependentvariables_list=[]
previoustable=''
currenttable=''
tempnewtablename=''
for i in range(1,len(eventindependentvariables)):
currenttable=(eventindependentvariables[i].split('|'))[1]
if currenttable==previoustable:
tempnewtablename=maptables[currenttable]
else:
tempnewtablename=returnnewtablename(currenttable)
temp_eventindependentvariables_list.append(tempnewtablename+'.`'+(eventindependent
variables[i].split('|'))[0]+'`)')
allvariables.append(eventindependentvariables[i].split('|')[0])

```

```

alltables.append(eventindependentvariables[i].split('|')[1])
    previoustable=currenttable

eventindependentvariables=','.join(temp_eventindependentvariables_list)
    else:

eventindependentvariables=formnvarmap['eventindependentvarselectform']

eventindependentvariables=eventindependentvariables.split(':')
    temp_eventindependentvariables_list=[]
    previoustable=''
    currenttable=''
    tempnewtablename=''
    for i in range(len(eventindependentvariables)):

currenttable=(eventindependentvariables[i].split('|'))[1]
    if currenttable==previoustable:
        tempnewtablename=maptables[currenttable]
    else:

tempnewtablename=returnnewtablename(currenttable)

temp_eventindependentvariables_list.append(tempnewtablename+'.'+(eventindependent
variables[i].split('|'))[0]+'`)

allvariables.append(eventindependentvariables[i].split('|')[0])

alltables.append(eventindependentvariables[i].split('|')[1])
    previoustable=currenttable

eventindependentvariables=','.join(temp_eventindependentvariables_list)
    else:
        pass
    #print eventindependentvariables
    if 'vitalsvariablesform' in formnvarmap:
        if
re.search(r'^allvitalsvariables',formnvarmap['vitalsvariablesform']):
    vitalsvariables=formnvarmap['vitalsvariablesform']
    vitalsvariables=vitalsvariables.split(':')
    temp_vitalsvariables_list=[]
    previoustable=''
    currenttable=''
    tempnewtablename=''
    for i in range(1,len(vitalsvariables)):
        currenttable=(vitalsvariables[i].split('|'))[1]
        if currenttable==previoustable:
            tempnewtablename=maptables[currenttable]
        else:

tempnewtablename=returnnewtablename(currenttable)

temp_vitalsvariables_list.append(tempnewtablename+'.'+(vitalsvariables[i].split('
|'))[0]+'`)

allvariables.append(vitalsvariables[i].split('|')[0])
    alltables.append(vitalsvariables[i].split('|')[1])
    previoustable=currenttable

```

```

        vidualsvariables=','.join(temp_vidualsvariables_list)
    else:
        vidualsvariables=formnvarmap['vidualsvariablesform']
        vidualsvariables=vidualsvariables.split(':')
        temp_vidualsvariables_list=[]
        previoustable=''
        currenttable=''
        tempnewtablename=''
        for i in range(len(vidualsvariables)):
            currenttable=(vidualsvariables[i].split('|'))[1]
            if currenttable==previoustable:
                tempnewtablename=maptables[currenttable]
            else:

tempnewtablename=returnnewtablename(currenttable)

temp_vidualsvariables_list.append(tempnewtablename+'.`'+(vidualsvariables[i].split('
|'))[0]+'`')

allvariables.append(vidualsvariables[i].split('|')[0])
        alltables.append(vidualsvariables[i].split('|')[1])
        previoustable=currenttable
        vidualsvariables=','.join(temp_vidualsvariables_list)
    else:
        pass
    #print vidualsvariables
    if 'hnmrvariablesform' in formnvarmap:
        if
re.search(r'^allhnmrvariables',formnvarmap['hnmrvariablesform']):
        hnmrvariables=formnvarmap['hnmrvariablesform']
        hnmrvariables=hnmrvariables.split(':')
        temp_hnmrvariables_list=[]
        previoustable=''
        currenttable=''
        tempnewtablename=''
        tempnormalization={}
        for i in range(1,len(hnmrvariables)):
            currenttable=(hnmrvariables[i].split('|'))[1]
            if currenttable==previoustable:
                tempnewtablename=maptables[currenttable]
            else:

tempnewtablename=returnnewtablename(currenttable)

temp_hnmrvariables_list.append(tempnewtablename+'.`'+(hnmrvariables[i].split('|'))
[0]+'`')
                if not tempnewtablename in tempnormalization:

tempnormalization[tempnewtablename]=((hnmrvariables[i].split('|'))[2]).split('+')

temp_hnmrvariables_list.append(tempnewtablename+'.`Normalization`')
                allvariables.append('Normalization')
            else:
                pass

allvariables.append(hnmrvariables[i].split('|')[0])
        alltables.append(currenttable)
        previoustable=currenttable

```

```

        hnmrvariables=','.join(temp_hnmrvariables_list)
    else:
        hnmrvariables=formnvarmap['hnmrvariablesform']
        hnmrvariables=hnmrvariables.split(':')
        temp_hnmrvariables_list=[]
        previoustable=''
        currenttable=''
        tempnewtablename=''
        tempnormalization={}
        for i in range(len(hnmrvariables)):
            currenttable=(hnmrvariables[i].split('|'))[1]
            if currenttable==previoustable:
                tempnewtablename=maptables[currenttable]
            else:
                tempnewtablename=returnnewtablename(currenttable)

        temp_hnmrvariables_list.append(tempnewtablename+'.`'+(hnmrvariables[i].split('|'))
        [0]+'`')
            if not tempnewtablename in tempnormalization:
                tempnormalization[tempnewtablename]=((hnmrvariables[i].split('|'))[2]).split('+')
        temp_hnmrvariables_list.append(tempnewtablename+'.`Normalization`')
            allvariables.append('Normalization')
        else:
            pass

        allvariables.append(hnmrvariables[i].split('|')[0])
            alltables.append(currenttable)
            previoustable=currenttable
            hnmrvariables=','.join(temp_hnmrvariables_list)
        for temptable in tempnormalization:
            modifytext="\` OR "+temptable+".`Normalization`=\`"
            captureonetablenorm="
            (" +temptable+".`Normalization`=\`"+modifytext.join(tempnormalization[temptable])+
            "\`)"
            if hnmrnormalization=='':
                hnmrnormalization=captureonetablenorm
            else:
                hnmrnormalization=hnmrnormalization+"AND"+captureonetablenorm
        else:
            pass
        #print hnmrtempnormal
        #print hnmrvariables
        #print hnmrnormalization
        if 'pnmrvariablesform' in formnvarmap:
            if
re.search(r'^allpnmrvariables',formnvarmap['pnmrvariablesform']):
        pnmrvariables=formnvarmap['pnmrvariablesform']
        pnmrvariables=pnmrvariables.split(':')
        temp_pnmrvariables_list=[]
        previoustable=''
        currenttable=''
        tempnewtablename=''
        tempnormalization={}
        for i in range(1,len(pnmrvariables)):

```

```

        currenttable=(pnmrvariables[i].split('|'))[1]
        if currenttable==previousstable:
            tempnewtablename=maptables[currenttable]
        else:

tempnewtablename=returnnewtablename(currenttable)

temp_pnmrvariables_list.append(tempnewtablename+'.'+(pnmrvariables[i].split('|'))
[0]+'`)

        if not tempnewtablename in tempnormalization:

tempnormalization[tempnewtablename]=((pnmrvariables[i].split('|'))[2]).split('+')

temp_pnmrvariables_list.append(tempnewtablename+'.`Normalization`')
        allvariables.append('Normalization')
        else:
            pass

allvariables.append(pnmrvariables[i].split('|')[0])
        alltables.append(currenttable)
        previousstable=currenttable
        pnmrvariables=', '.join(temp_pnmrvariables_list)
    else:
        pnmrvariables=formnvarmap['pnmrvariablesform']
        pnmrvariables=pnmrvariables.split(':')
        temp_pnmrvariables_list=[]
        previousstable=''
        currenttable=''
        tempnewtablename=''
        tempnormalization={}
        for i in range(len(pnmrvariables)):
            currenttable=(pnmrvariables[i].split('|'))[1]
            if currenttable==previousstable:
                tempnewtablename=maptables[currenttable]
            else:

tempnewtablename=returnnewtablename(currenttable)

temp_pnmrvariables_list.append(tempnewtablename+'.'+(pnmrvariables[i].split('|'))
[0]+'`)

                if not tempnewtablename in tempnormalization:

tempnormalization[tempnewtablename]=((pnmrvariables[i].split('|'))[2]).split('+')

temp_pnmrvariables_list.append(tempnewtablename+'.`Normalization`')
                    allvariables.append('Normalization')
                    else:
                        pass

allvariables.append(pnmrvariables[i].split('|')[0])
                    alltables.append(currenttable)
                    previousstable=currenttable
                    pnmrvariables=', '.join(temp_pnmrvariables_list)
                for temptable in tempnormalization:
                    modifytext="\` OR "+temptable+".`Normalization`="
                    captureonetablenorm="
(+temptable+".`Normalization`="+modifytext.join(tempnormalization[temptable])+
\) "

```

```

        if pnmrnormalization=='':
            pnmrnormalization=captureonetablenorm
            pnmrintegralonly="
("+temptable+".`Value_type`='Integral') "
            else:

pnmrnormalization=pnmrnormalization+"AND"+captureonetablenorm
            pnmrintegralonly=pnmrintegralonly+"AND"+"
("+temptable+".`Value_type`='Integral') "

        else:
            pass
        #print pnmrtempnormal
        #print pnmrvariables
        #print pnmrnormalization
        """
        if 'pnmrvariablesform' in formnvarmap:
            if
re.search(r'^allpnmrvariables',formnvarmap['pnmrvariablesform']):
                pnmrvariables=formnvarmap['pnmrvariablesform']
                pnmrvariables=pnmrvariables.split(':')
                temp_pnmrvariables_list=[]
                previoustable=''
                currenttable=''
                tempnewtablename=''
                tempnormalization={}
                for i in range(1,len(pnmrvariables)):
                    currenttable=(pnmrvariables[i].split('|'))[1]
                    if currenttable==previoustable:
                        tempnewtablename=maptables[currenttable]
                    else:

tempnewtablename=returnnewtablename(currenttable)

temp_pnmrvariables_list.append(tempnewtablename+'.`'+(pnmrvariables[i].split('|'))
[0]+'`')

                    if not tempnewtablename in tempnormalization:

tempnormalization[tempnewtablename]=((pnmrvariables[i].split('|'))[2]).split('+')

temp_pnmrvariables_list.append(tempnewtablename+'.`Normalization`')
                        allvariables.append('Normalization')
                    else:
                        pass

allvariables.append(pnmrvariables[i].split('|')[0])
                        alltables.append(pnmrvariables[i].split('|')[1])
                        previoustable=currenttable
                        pnmrvariables=', '.join(temp_pnmrvariables_list)
            else:
                pnmrvariables=formnvarmap['pnmrvariablesform']
                pnmrvariables=pnmrvariables.split(':')
                temp_pnmrvariables_list=[]
                previoustable=''
                currenttable=''
                tempnewtablename=''
                tempnormalization={}
                for i in range(len(pnmrvariables)):

```



```

        currenttable=(pnmrvariables[i].split('|'))[1]
        if currenttable==previousstable:
            tempnewtablename=maptables[currenttable]
        else:

tempnewtablename=returnnewtablename(currenttable)

temp_pnmrvariables_list.append(tempnewtablename+'.'+(pnmrvariables[i].split('|'))
[0]+'`)

            if not tempnewtablename in tempnormalization:

tempnormalization[tempnewtablename]=((pnmrvariables[i].split('|'))[2]).split('+')

temp_pnmrvariables_list.append(tempnewtablename+'.`Normalization`')
            allvariables.append('Normalization')
        else:
            pass

allvariables.append(pnmrvariables[i].split('|')[0])
            alltables.append(pnmrvariables[i].split('|')[1])
            previousstable=currenttable
            pnmrvariables=','.join(temp_pnmrvariables_list)
            for temptable in tempnormalization:
                modifytext="\` OR "+temptable+".`Normalization`=\`"
                captureonetablenorm="
("+temptable+".`Normalization`=\`"+modifytext.join(tempnormalization[temptable])+
\`) AND (" +temptable+".`Value_type`='Integral') "
                if pnmrnormalization=='':
                    pnmrnormalization=captureonetablenorm
                else:

pnmrnormalization=pnmrnormalization+"AND"+captureonetablenorm
            else:
                pass
            #print pnmrtempnormal
            #print pnmrvariables
            #print pnmrnormalization
            """
            if 'binningvariablesform' in formnvarmap:
                if
re.search(r'^allbinningvariables',formnvarmap['binningvariablesform']):
                binningvariables=formnvarmap['binningvariablesform']
                binningvariables=binningvariables.split(':')
                temp_binningvariables_list=[]
                previousstable=''
                currenttable=''
                tempnewtablename=''
                for i in range(1,len(binningvariables)):
                    currenttable=(binningvariables[i].split('|'))[1]
                    if currenttable==previousstable:
                        tempnewtablename=maptables[currenttable]
                    else:

tempnewtablename=returnnewtablename(currenttable)

temp_binningvariables_list.append(tempnewtablename+'.'+(binningvariables[i].split
('|'))[0]+'`)

```

```

allvariables.append(binningvariables[i].split('|')[0])

alltables.append(binningvariables[i].split('|')[1])
    previoustable=currenttable
    binningvariables=', '.join(temp_binningvariables_list)
else:
    binningvariables=formnvarmap['binningvariablesform']
    binningvariables=binningvariables.split(':')
    temp_binningvariables_list=[]
    previoustable=''
    currenttable=''
    tempnewtablename=''
    for i in range(len(binningvariables)):
        currenttable=(binningvariables[i].split('|'))[1]
        if currenttable==previoustable:
            tempnewtablename=maptables[currenttable]
        else:

tempnewtablename=returnnewtablename(currenttable)

temp_binningvariables_list.append(tempnewtablename+'`'+(binningvariables[i].split
('|'))[0]+'`')

allvariables.append(binningvariables[i].split('|')[0])

alltables.append(binningvariables[i].split('|')[1])
    previoustable=currenttable
    binningvariables=', '.join(temp_binningvariables_list)
else:
    pass
#print binningvariables
alltables=set(alltables)
Datapoints=''
datapointvariables=''
if ('datapointsselectform' in formnvarmap) and
(len(alltables)>1) :
    if
re.search(r'allevents',formnvarmap['datapointsselectform']):
        Datapoints=' (`t2`.`Event_name`!=99) '
    elif
re.search(r'alldatapoints',formnvarmap['datapointsselectform']):
        Datapoints=' (`t2`.`Event_name`) '
    else:
        if 'eventchoiceform' in formnvarmap:
            Datapoints=re.sub(r'^[0-
9:]', '', formnvarmap['eventchoiceform'])
            Datapoints=Datapoints.split(':')
            temp_Datapoints_list=[]
            for Datapoint in Datapoints:

temp_Datapoints_list.append("`t2`.`Event_name`="+Datapoint+"")
            Datapoints=' OR '.join(temp_Datapoints_list)
            Datapoints=' ('+Datapoints+') '
        else:
            Datapoints=' (`t2`.`Event_name`!=99) '
            allvariables.append('Event_name')
            datapointvariables=('`t2`.`Event_name`)

```

```

else:
    pass
#print datapointvariables
#print Datapoints
#print tablename
#print newtablename
#print allvariables

tablesorderforleftjoin=['monr_vitals_manual_10_min','monr_vitals_manual_one_hour',
'monr_lab_values','monr_gcs_scores','monr_teg_data','monr_hnmr_muscle','monr_hnmr_liver',
'monr_hnmr_urine','monr_hnmr_serum','monr_pnmr_muscle','monr_pnmr_liver','monr_binning_muscle',
'monr_binning_liver','monr_binning_urine','monr_binning_serum',
'monr_identifier_table']
sortedalltables=[]
for thetable in tablesorderforleftjoin:
    if thetable in alltables:
        sortedalltables.append(thetable)
    else:
        pass
#print alltables
#print sortedalltables
the_var='SELECT DISTINCT '
theleftjoion=' LEFT JOIN '
theinnerjoin=' INNER JOIN '
theas=' AS '
thefrom=' FROM '
theon=' ON '
onetime=0
jointables=' FROM '
the_where=" WHERE (`t1`.`Datainput_status`='Finished') AND "
the_and='#'
for i in range(len(sortedalltables)):
    if (len(sortedalltables)>1):

joinonpigid=maptables[sortedalltables[0]]+'.`Pig_ID`='+maptables[sortedalltables[i]]+'.`Pig_ID`'

joinoneventname=maptables[sortedalltables[0]]+'.`Event_name`='+maptables[sortedalltables[i]]+'.`Event_name`'

joinondateofexp=maptables[sortedalltables[0]]+'.`Date_of_experiment`='+maptables[sortedalltables[i]]+'.`Date_of_experiment`'

joinontimeofexp=maptables[sortedalltables[0]]+'.`Time_of_experiment`='+maptables[sortedalltables[i]]+'.`Time_of_experiment`'
        joinonnormalizations=maptables[sortedalltables[i-1]]+'.`Normalization`='+maptables[sortedalltables[i]]+'.`Normalization`'
        else:
            pass
            if i==0:

jointables=jointables+'`'+sortedalltables[i]+'`'+theas+maptables[sortedalltables[i]]
        else:
            if
(sortedalltables[i]=='monr_vitals_manual_one_hour') and (sortedalltables[i-1]=='monr_vitals_manual_10_min'):
                #if i==1:

```

```

jointables=jointables+theleftjoion+'`'+sortedalltables[i]+'`'+theas+maptables[sort
edalltables[i]]+theon+joinondateofexp+' AND '+joinontimeofexp
    #else:

#jointables=jointables+theleftjoion+'`'+sortedalltables[i]+'`'+theas+maptables[sort
edalltables[i]]+theon+joinonpigid+' AND '+joinoneventname
    elif (sortedalltables[i]=='monr_identifier_table'):

jointables=jointables+theleftjoion+'`'+sortedalltables[i]+'`'+theas+maptables[sort
edalltables[i]]+theon+joinonpigid
    elif (sortedalltables[i]=='monr_hnmr_liver' and
sortedalltables[i-1]=='monr_hnmr_muscle'):

jointables=jointables+theleftjoion+'`'+sortedalltables[i]+'`'+theas+maptables[sort
edalltables[i]]+theon+joinonpigid+' AND '+joinoneventname+' AND
'+joinonnormalizations
    elif (sortedalltables[i]=='monr_hnmr_serum' and
sortedalltables[i-1]=='monr_hnmr_urine'):

jointables=jointables+theleftjoion+'`'+sortedalltables[i]+'`'+theas+maptables[sort
edalltables[i]]+theon+joinonpigid+' AND '+joinoneventname+' AND
'+joinonnormalizations
    elif (sortedalltables[i]=='monr_pnmr_muscle' and
(sortedalltables[i-1]=='monr_hnmr_muscle' or sortedalltables[i-
1]=='monr_hnmr_liver')):

jointables=jointables+theleftjoion+'`'+sortedalltables[i]+'`'+theas+maptables[sort
edalltables[i]]+theon+joinonpigid+' AND '+joinoneventname+' AND
'+joinonnormalizations
    elif (sortedalltables[i]=='monr_pnmr_liver' and
(sortedalltables[i-1]=='monr_hnmr_muscle' or sortedalltables[i-
1]=='monr_hnmr_liver' or sortedalltables[i-1]=='monr_pnmr_muscle')):

jointables=jointables+theleftjoion+'`'+sortedalltables[i]+'`'+theas+maptables[sort
edalltables[i]]+theon+joinonpigid+' AND '+joinoneventname+' AND
'+joinonnormalizations
    else:

jointables=jointables+theleftjoion+'`'+sortedalltables[i]+'`'+theas+maptables[sort
edalltables[i]]+theon+joinonpigid+' AND '+joinoneventname
    #jointables=re.sub('###','#',jointables)
    #jointables=jointables.strip('#')
    #jointables=re.sub('#',' AND ',jointables)

the_var=the_var+identifiervariables+', '+datapointvariables+', '+eventindependentvar
iables+', '+vitalsvariables+', '+hnmrvariables+', '+pnmrvariables+', '+binningvariable
s
    the_var=re.sub(',*,',',',the_var)
    the_var=the_var.strip(',')
    the_var_and_join=the_var+jointables

the_where=the_where+Pig_groups+the_and+Pig_IDs+the_and+Datapoints+the_and+hnmrnorm
alization+the_and+pnmrnormalization+the_and+pnmrintegralonly
    the_where=re.sub('###','#',the_where)
    the_where=the_where.strip('#')
    the_where=re.sub('#',' AND ',the_where)

```

```

        if not (re.search(r'=',(the_where).lower()) or
re.search(r'=',(the_where).lower()) or re.search(r'=',the_where) or
re.search(r'=',(the_where).lower()) or re.search(r'is',(the_where).lower()) or
re.search(r'like',(the_where).lower())):
            the_where=''
        else:
            pass
        #print the_where
        vitals_10min_table='monr_vitals_manual_10_min'
        vitals_one_hour_table='monr_vitals_manual_one_hour'
        labs_table='monr_lab_values'
        gcs_table='monr_gcs_scores'

hnmr_pnmr_tables=['monr_hnmr_muscle','monr_hnmr_liver','monr_hnmr_urine','monr_hnm
r_serum','monr_pnmr_muscle','monr_pnmr_liver']
        the_order=" ORDER BY "
        if(len(maptables)>1):
            if (vitals_10min_table in maptables) or
(vitals_one_hour_table in maptables):

the_order=the_order+"`t1`.`Start_date`,`t2`.`Date_of_experiment`,`t2`.`Time_of_exp
eriment`"
                #finalquery=the_var_and_join+the_where+" ORDER BY
`t1`.`Start_date`,`t2`.`Date_of_experiment`,`t2`.`Time_of_experiment`"
            else:

the_order=the_order+"`t1`.`Start_date`,`t2`.`Event_name`"
                #finalquery=the_var_and_join+the_where+" ORDER BY
`t1`.`Start_date`,`t2`.`Event_name`"
                for the_table in hnmr_pnmr_tables:
                    if the_table in maptables:

the_order=the_order+', '+maptables[the_table]+'Normalization'
                    #Ordering table as below without alias infront
will #raise mysql error
                    #the_order=the_order+',Normalization'
                    break
                else:
                    pass
            else:
                the_order=the_order+"`Start_date`"
                #finalquery=the_var_and_join+the_where+" ORDER BY
`Start_date`"

        finalquery=the_var_and_join+the_where+the_order
        #print finalquery
        lowerlimit = int(form['lowerlimit'].value)
        starting_row = lowerlimit+1
        numberofdatapoints = int(form['numberofdatapoints'].value)
        ending_row = lowerlimit+numberofdatapoints
        #The commented approach below to find the total rows of the
query has been abandoned as SELECT * won't allow duplicate names for columns
        #countquery="SELECT COUNT(*) FROM("+finalquery+") AS
table_alias"

        #cursor.execute(countquery)
        #output2 = cursor.fetchall()
        #advancedexport_total_data_points = int(output2[0][0])
        #cursor.execute(finalquery)
        #output2 = cursor.fetchall()

```

```

#advancedexport_total_data_points = len(output2)
nolimitquery=finalquery
#####Temporarily store sql
query#####
thesqlqueryrandomkey=''
try:
    thesqlqueryrandomkey=randkey(30,40)
    cursor.execute("""DELETE FROM monr_temp_sql_query_storage
WHERE Saved_time<=SUBTIME(CURRENT_TIMESTAMP, '02:00:00')""")
    cursor.execute("""INSERT INTO
monr_temp_sql_query_storage(Random_string,Query_string,Saved_time)
VALUES("%s", "%s",CURRENT_TIMESTAMP)""%(thesqlqueryrandomkey,sql.escape_string(nolimitquery)))
    except:#This repetition is to account for an extreme case
where the Random_string matches the one already in the table
    thesqlqueryrandomkey=randkey(30,40)
    cursor.execute("""DELETE FROM monr_temp_sql_query_storage
WHERE Saved_time<=SUBTIME(CURRENT_TIMESTAMP, '02:00:00')""")
    cursor.execute("""INSERT INTO
monr_temp_sql_query_storage(Random_string,Query_string,Saved_time)
VALUES("%s", "%s",CURRENT_TIMESTAMP)""%(thesqlqueryrandomkey,sql.escape_string(nolimitquery)))

#####
#####
    finalquery=finalquery+" LIMIT
%d,%d"%(lowerlimit,numberofdatapoints)
    cursor.execute(finalquery)
    output=cursor.fetchall()
    thelength = len(output)
    #thelength = cursor.rowcount
    if(numberofdatapoints > thelength):
        ending_row = lowerlimit+thelength
    else:
        pass
    if ending_row==0:
        starting_row=0
    else:
        pass
    connection.commit()
    cursor.close()
    connection.close()
    #print output
    #print the_var,'\n'
    #print allvariables

#####
#####
    row="oddRow"
    display_data=''
    headers=re.sub('SELECT DISTINCT ','',the_var)
    headers=re.sub('`,`','@',headers)
    #headers=re.sub('`t\d+`.`','',headers)
    #headers=re.sub('`,`','',headers)
    headers=headers.split('@')
    headercolumn=''
    datarows=''
    temp_headers=[]

```

```

#####
#####
    for header in headers:
        header=header.replace('.', '@')
        header_table=header.split('@')[0]
        variable_name=header.split('@')[-1]
        variable_name=variable_name.replace('.', '')
        if header_table in match_tables:
            thetable=match_tables[header_table]
            thetable=thetable.replace('monr_', '')
            thetable=thetable.replace('_manual_one_hour', '')
            thetable=thetable.replace('_manual_10_min', '')
            thetable=thetable.replace('_', ' ')
            thetable=thetable+' | '
            temp_headers.append(thetable+variable_name)
        else:
            temp_headers.append("unknown|" +variable_name)
#####Excel
Creation#####

raj_excel_export.ldb=ldb;raj_excel_export.lhost=lhost;raj_excel_export.luser=luser
;raj_excel_export.lpasswd=lpasswd;raj_excel_export.lport=lport;raj_excel_export.st
udyname=studyname
        tempdirectoryname=form['tempdirectoryname'].value
        tempbasedir='/project/beilmanlab.umn.edu-
upload/MONR/temp_files/'
        if tempdirectoryname=='null':
            try:
                for the_folder in os.listdir(tempbasedir):
                    if re.search(current_user_id,the_folder):
                        for the_file in
os.listdir(tempbasedir+the_folder):
os.remove(tempbasedir+the_folder+'/' +the_file)
                    os.rmdir(tempbasedir+the_folder)
                    #shutil.rmtree(tempbasedir+'/' +the_folder)
                else:
                    pass
            except:
                pass
                #raise
        tempdirectoryname=current_user_id+randkey(7,10)
        store_save_dir=tempbasedir+tempdirectoryname+'/'
        try:
            #Make sure the path exists
            if not os.path.exists(tempbasedir):
                os.umask(0)
                os.makedirs(store_save_dir,0771)
            else:
                pass
            if not os.path.exists(store_save_dir):
                os.umask(0)
                os.makedirs(store_save_dir,0771)
            else:
                pass
        except:
            pass

```

```

        ##raise
        store_save_dir=(store_save_dir.replace('\', '/'))
    else:
        store_save_dir=tempbasedir+tempdirectoryname+'/'
        raj_excel_export.excel_save_dir=store_save_dir
        raj_excel_export.csv_save_dir=store_save_dir

#tmpfd,excel_file_path=mkstemp(suffix='.xls',prefix='raj',dir=store_save_dir)
store_file_name=current_user_id+randkey(7,10)
excel_file_name=store_file_name+'.xls'
csv_file_name=store_file_name+'.csv'
excel_file_path=store_save_dir+excel_file_name
csv_file_path=store_save_dir+csv_file_name
#store_file_name=excel_file_path.split('/')[ -1]
try:

create_and_save_excel_and_csv(studyname,store_file_name,nolimitquery,match_tables)
        #excel_download=""<button type="button"
class="downloadicons" id="downloadexcel" title="Download this data as excel file"
onclick="javascript:opener.downloadFile('\tempfiledownload\','\%s\','\%s\');
"></button>""%(studyname,excel_file_name)
        #csv_download=""<button type="button"
class="downloadicons" id="downloadcsv" title="Download this data as csv file"
onclick="javascript:opener.downloadFile('\tempfiledownload\','\%s\','\%s\');
"></button>""%(studyname,csv_file_name)
        excel_and_csv=""<div class="downloadicons"
style="float:left;padding-left:5px;padding-bottom:10px;padding-top:10px"
id="downloadexcel" title="Download this data as excel file"
onclick="javascript:opener.downloadFile('\tempfiledownload\','\%s\','\%s\');
">&nbsp;</div><div class="downloadicons" style="float:left;padding-
left:5px;padding-bottom:10px;padding-top:10px" id="downloadcsv" title="Download
this data as csv file"
onclick="javascript:opener.downloadFile('\tempfiledownload\','\%s\','\%s\');
">&nbsp;</div>""%(studyname,excel_file_name,studyname,csv_file_name)
    except:
        #excel_download='Error'
        #csv_download='Error'
        excel_and_csv=""Error in creating Excel and/or CSV
files""

advancedexport_total_data_points=raj_excel_export.output_row_count

#####
#####
#####
        #errors_found=0
        #def
update_tempfile_database(File_name,Dir_name,File_storage_path,Show_name):
        #global errors_found,Updated_by
        #try:
            #if(os.path.exists(File_storage_path)):
                #insert_info=""INSERT IGNORE INTO
monr_tempfilestorage(File_name,Dir_name,File_storage_path,Show_name)
VALUES('%s','%s','%s','%s')""%(File_name,Dir_name,File_storage_path,Show_name)
                #connect_to_db_and_execute(insert_info)
            #else:
                #errors_found=errors_found+1
        #except:

```



```

#errors_found=errors_found+1

#####
#####
#####

#update_tempfile_database(excel_file_name,tempdirectoryname,excel_file_path,"Data_
export.xls")

#####
#####
#####

    for header in temp_headers:

headercolumn=headercolumn+""<th><center><div>%s</div></center></th>""%(str(heade
r))

    headerrow=""<tr class="%s">%s</tr>""%(row,headercolumn)
    for i in range(len(output)):
        if row=="oddRow":
            row="evenRow"
        else:
            row="oddRow"
        datafields=''
        for j in range(len(output[i])):
            dataoutput=str(output[i][j])
            if dataoutput=='None':
                dataoutput=''
            else:
                pass
            dataoutput=dataoutput.replace('\n','')
            datafields=datafields+""<td
title="%s"><center><div>%s</div></center></td>""%(temp_headers[j],dataoutput)
            datarows=datarows+""<tr
class="%s">%s</tr>""%(row,datafields)
            display_data=headerrow+datarows

display_table=""<table><tbody>%s</tbody></table>""%(display_data)
    if len(output)==0:
        display_table="No data generated for the query. Please
modify the query and try again. If the problem persits, contact data manager"
    else:
        pass

#####
#####

        print 'Content-Type: text/javascript\n'
        print
        ""alladvancedExportPopup.document.getElementById('message1').innerHTML='';""
        print
        ""alladvancedExportPopup.document.getElementById('datatable').innerHTML =
's';""%(display_table)
        print
        ""advancedexport_total_data_points=%d;pagination();""%(advancedexport_total_data
_points)
        print
        ""alladvancedExportPopup.document.getElementById('message1').innerHTML='Showing
rows from %d to %d';""%(starting_row,ending_row)
        print ""tempdirectoryname='%s';""%(tempdirectoryname)

```

```

        #print
        """alladvancedExportPopup.document.getElementById('excel_download').innerHTML='%s'
        """%(excel_download)
        #print
        """alladvancedExportPopup.document.getElementById('csv_download').innerHTML='%s'"""
        %(csv_download)
        print
        """alladvancedExportPopup.document.getElementById('holddownloadicons').innerHTML='
        %s'"""%(excel_and_csv)
        print """thesqlqueryrandomkey='%s'"""%(thesqlqueryrandomkey)
        #print """alert(thesqlqueryrandomkey)"""
        #print
        """document.getElementById('message2').innerHTML="%s";"""%(finalquery)
        except:
            print 'Content-Type: text/javascript\n'
            print
            """alladvancedExportPopup.document.getElementById('message1').innerHTML='There was
            an error. Please contact data manager.';"""
            print
            """alladvancedExportPopup.document.getElementById('datatable').innerHTML='There
            was an error. Please contact data manager.';"""
            #raise

#####
#####
#####
        elif(action=="displayQueryOutput"):
            try:
                thesqlqueryrandomkey=form['thesqlqueryrandomkey'].value
                lowerlimit=int(form['lowerlimit'].value)
                starting_row = lowerlimit+1
                numberofdatapoints=int(form['numberofdatapoints'].value)
                ending_row = lowerlimit+numberofdatapoints
                connection = sql.connect(host=lhost,port=lport,user=luser,
                passwd=lpasswd,db=lodb)
                cursor = connection.cursor()
                set_character_set="""SET NAMES 'utf8' COLLATE 'utf8_bin'"""
                cursor.execute(set_character_set)
                cursor.execute("""SELECT Query_string FROM
                monr_temp_sql_query_storage WHERE Random_string='%s'"""%(thesqlqueryrandomkey))
                output=cursor.fetchall()
                finalquery=output[0][0]
                finalquery=finalquery+" LIMIT
                %d,%d"%(lowerlimit,numberofdatapoints)
                cursor.execute(finalquery)
                output1=cursor.fetchall()
                thelength = len(output1)
                #thelength = cursor.rowcount
                if(numberofdatapoints > thelength):
                    ending_row = lowerlimit+thelength
                else:
                    pass
                if ending_row==0:
                    starting_row=0
                else:
                    pass
                connection.commit()
                cursor.close()

```

```

connection.close()

#####
#####
        row="oddRow"
        display_data=''
        headers=[]
        headercolumn=''
        datarows=''

#####
#####
        the_query=finalquery.split('FROM')
        the_query=(the_query[0]).replace('SELECT ','')
        the_query=the_query.replace('.', '@')
        the_query=the_query.replace(',', '#')
        the_query=the_query.split('#')
        column_array=[]
        for i in range(len(the_query)):
            table_column=the_query[i].split('@')
            column_name=table_column[-1]
            column_name=column_name.replace('.', '')
            headers.append(column_name)
        for header in headers:

headercolumn=headercolumn+""<th><center><div>%s</div></center></th>""%(str(header))

        headerrow=""<tr class="%s">%s</tr>""%(row, headercolumn)
        for i in range(len(output1)):
            if row=="oddRow":
                row="evenRow"
            else:
                row="oddRow"
            datafields=''
            for j in range(len(output1[i])):
                dataoutput=str(output1[i][j])
                if dataoutput=='None':
                    dataoutput=''
                else:
                    pass
                dataoutput=dataoutput.replace("\'", "\\'\")
                dataoutput=dataoutput.replace('\n', '<br />')
                datafields=datafields+""<td
title="%s"><center><div>%s</div></center></td>""%(headers[j], dataoutput)
                datarows=datarows+""<tr
class="%s">%s</tr>""%(row, datafields)
                display_data=headerrow+datarows

display_table=""<table><tbody>%s</tbody></table>""%(display_data)
        if len(output)==0:
            display_table="No data generated for the query. Please
modify the query and try again. If the problem persits, contact data manager"
        else:
            pass

#####
#####
        print 'Content-Type: text/javascript\n'

```

```

        print
        """alladvancedExportPopup.document.getElementById('message1').innerHTML='';"""
        print
        """alladvancedExportPopup.document.getElementById('datatable').innerHTML =
        '%s';"""%(display_table)
        print """pagination();"""
        print
        """alladvancedExportPopup.document.getElementById('message1').innerHTML='Showing
        rows from %d to %d';"""%(starting_row,ending_row)
        except:
            print 'Content-Type: text/javascript\n'
            print
            """alladvancedExportPopup.document.getElementById('message1').innerHTML='There was
            an error. Please contact data manager.';"""
            print
            """alladvancedExportPopup.document.getElementById('datatable').innerHTML='There
            was an error. Please contact data manager.';"""
            #raise

#####
#####
#####
        elif(action=="loaduserids"):
            formname=form['formname'].value
            if formname=='savequery':
                try:
                    request_from_tables=['sccl_authentication']
                    connection = sql.connect(host=lhost,port=lport,user=luser,
                    passwd=lpasswd,db=ldb)
                    cursor = connection.cursor()
                    set_character_set="""SET NAMES 'utf8' COLLATE
                    'utf8_bin'"""

                    cursor.execute(set_character_set)
                    row="evenRow"
                    userids=[]
                    display_data="""<tr class="%s"><td
                    class="optionscolumn"><label for="allusers">All users</label></td><td
                    class="inputcolumn"><input type="checkbox" name="allusers" id="allusers"
                    onclick="javascript:opener.selectAll('\`users\`');" /></td></tr>"""%(row)
                    for requestedtable in request_from_tables:
                        cursor.execute("""SELECT
                        User_first_name,User_middle_initial,User_last_name,User_id FROM `%s` WHERE
                        User_id!='%s'"""%(requestedtable,current_user_id))
                        output=cursor.fetchall()
                        data_requested='User Name (User ID)'
                        if len(output)>0:
                            display_data=display_data+"""<tr
                            class="subheading"><td colspan="2">%s</td></tr>"""%(data_requested)
                            for firstname,middleinitial,lastname,userid in
                            output:
                                if not (middleinitial==None or
                                middleinitial==' ' or len(middleinitial)<1):
                                    middleinitial=' '+middleinitial+'.'
                                else:
                                    middleinitial=''
                                lastname=' '+lastname
                                display=firstname+middleinitial+lastname+'
                                ('+userid+')'

```

```

        userids.append(display)
    else:
        pass
userids.sort()
for userdata in userids:
    if (row=="oddRow"):
        row = "evenRow"
    else:
        row = "oddRow"
    Display=userdata
    data_name=data_requested
    userid=userdata.split('(')[1]
    userid=userid.replace(')','')
    name_and_id=userid
    display_data=display_data+""<tr class="%s"><td
title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:opener.selectOne(\\'users\\');"
/></td></tr>""%(row,data_name,name_and_id,Display,name_and_id,name_and_id)
    display_table=""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody>%s</tbody></table></td></tr></table></div>""%(display_da
ta)

    print 'Content-Type: text/javascript\n'
    print
    ""alladvancedExportPopup.document.getElementById('userids').innerHTML='%s';""%(d
isplay_table)

    connection.commit()
    cursor.close()
    connection.close()
except:
    print 'Content-Type: text/javascript\n'
    print
    ""alladvancedExportPopup.document.getElementById('userids').innerHTML='An error
occured. Please contact data manager.';""
    #raise

#####
elif formname=='editquery':
    try:
        request_from_tables=['sccl_authentication']
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        set_character_set=""SET NAMES 'utf8' COLLATE
'utf8_bin'""

        cursor.execute(set_character_set)
        row="evenRow"
        userids=[]
        display_data=""<tr class="%s"><td
class="optionscolumn"><label for="allusers">All users</label></td><td
class="inputcolumn"><input type="checkbox" name="allusers" id="allusers"
onclick="javascript:selectAll(\\'users\\');" /></td></tr>""%(row)
        for requestedtable in request_from_tables:
            cursor.execute("""SELECT
User_first_name,User_middle_initial,User_last_name,User_id FROM `%s` WHERE
User_id!='%s'""%(requestedtable,current_user_id))
            output=cursor.fetchall()

```

```

        data_requested='User Name (User ID)'
        if len(output)>0:
            display_data=display_data+""<tr
class="subheading"><td colspan="2">%s</td></tr>""%(data_requested)
            for firstname,middleinitial,lastname,userid in
output:
                if not (middleinitial==None or
middleinitial==' ' or len(middleinitial)<1):
                    middleinitial=' '+middleinitial+'.'
                else:
                    middleinitial=''
                    lastname=' '+lastname
                    display=firstname+middleinitial+lastname+'
('+userid+)'
                    userids.append(display)
            else:
                pass
        userids.sort()
        for userdata in userids:
            if (row=="oddRow"):
                row = "evenRow"
            else:
                row = "oddRow"
            Display=userdata
            data_name=data_requested
            userid=userdata.split('(')[1]
            userid=userid.replace(')','')
            name_and_id=userid
            display_data=display_data+""<tr class="%s"><td
title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne('\\'users\\');"
/></td></tr>""%(row,data_name,name_and_id,Display,name_and_id,name_and_id)
            display_table=""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody>%s</tbody></table></td></tr></table></div>""%(display_da
ta)
            Query_name=form['query_name'].value
            cursor.execute("""SELECT Shared_with FROM
monr_shared_queries WHERE Query_name='%s' AND
Shared_by='%s'""%(Query_name,current_user_id))
            output1=cursor.fetchall()
            userstring=''
            print 'Content-Type: text/javascript\n'
            print
            ""document.getElementById('edituserids').innerHTML='%s';""%(display_table)
            if len(output1)>0:
                for i in range(len(output1)):
                    if i==0:
                        userstring=userstring+output1[i][0]
                    else:
                        userstring=userstring+' '+output1[i][0]
                print ""old_user_string='%s'""%(userstring)
                print ""markSelectedUsers();""
            else:
                pass
        connection.commit()
        cursor.close()

```

```

        connection.close()
    except:
        print 'Content-Type: text/javascript\n'
        print
        """document.getElementById('edituserid').innerHTML='An error occured. Please
        contact data manager.';"""
        #raise

#####
        elif formname=='editsharedquery':
            try:
                request_from_tables=['sccl_authentication']
                connection = sql.connect(host=lhost,port=lport,user=luser,
                passwd=lpasswd,db=lhb)
                cursor = connection.cursor()
                set_character_set="""SET NAMES 'utf8' COLLATE
                'utf8_bin'"""

                cursor.execute(set_character_set)
                row="evenRow"
                userids=[]
                display_data="""<tr class="%s"><td
                class="optionscolumn"><label for="allusers">All users</label></td><td
                class="inputcolumn"><input type="checkbox" name="allusers" id="allusers"
                onclick="javascript:selectAll(\\'users\\');" /></td></tr>"""%(row)
                for requestedtable in request_from_tables:
                    cursor.execute("""SELECT
                    User_first_name,User_middle_initial,User_last_name,User_id FROM `%s` WHERE
                    User_id!='%s'"""%(requestedtable,current_user_id))
                    output=cursor.fetchall()
                    data_requested='User Name (User ID)'
                    if len(output)>0:
                        display_data=display_data+"""<tr
                        class="subheading"><td colspan="2">%s</td></tr>"""%(data_requested)
                        for firstname,middleinitial,lastname,userid in
                        output:
                            if not (middleinitial==None or
                            middleinitial==' ' or len(middleinitial)<1):
                                middleinitial=' '+middleinitial+'.'
                            else:
                                middleinitial=''
                            lastname=' '+lastname
                            display=firstname+middleinitial+lastname+'
                            ('+userid+')'

                        userids.append(display)
                    else:
                        pass
                userids.sort()
                for userdata in userids:
                    if (row=="oddRow"):
                        row = "evenRow"
                    else:
                        row = "oddRow"
                    Display=userdata
                    data_name=data_requested
                    userid=userdata.split('(')[1]
                    userid=userid.replace(')','')
                    name_and_id=userid

```

```

                display_data=display_data+""<tr class="%s"><td
title="%s" class="optionscolumn"><label for="%s">%s</label></td><td
class="inputcolumn"><input type="checkbox" name="%s" id="%s"
onclick="javascript:selectOne(\\'users\\');"
/></td></tr>""%(row,data_name,name_and_id,Display,name_and_id,name_and_id)
        display_table=""<div class="innerdiv"><table
class="wrappertable"><tr><td class="wrappercell"><table
class="selection"><tbody>%s</tbody></table></td></tr></table></div>""%(display_da
ta)

        Query_name=form['query_name'].value
        userid=current_user_id
        if formname=='editsharedquery':
            userid=form['shared_by'].value
        else:
            pass
        cursor.execute("""SELECT Shared_with FROM
monr_shared_queries WHERE Query_name='%s' AND Shared_by='%s' AND
Shared_with!='%s'""%(Query_name,userid,current_user_id))
        output1=cursor.fetchall()
        userstring=''
        print 'Content-Type: text/javascript\n'
        print
        ""document.getElementById('edituserid').innerHTML='%s';""%(display_table)
        if len(output1)>0:
            for i in range(len(output1)):
                if i==0:
                    userstring=userstring+output1[i][0]
                else:
                    userstring=userstring+':'+output1[i][0]
            print ""old_user_string='%s'""%(userstring)
            print ""markSelectedUsers();""
        else:
            pass
        connection.commit()
        cursor.close()
        connection.close()
    except:
        print 'Content-Type: text/javascript\n'
        print
        ""document.getElementById('edituserid').innerHTML='An error occured. Please
contact data manager.';""
        #raise

#####
#####
#####
        elif(action=="savesqlquery"):
            typeofform=form['formtype'].value
            try:
                if typeofform=='savesqlquery':
                    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
                    cursor = connection.cursor()
                    set_character_set=""SET NAMES 'utf8' COLLATE
'utf8_bin'""

                    cursor.execute(set_character_set)
                    nameofthequery=form['nameofthequery'].value
                    descriptionofthequery=form['descriptionofthequery'].value

```



```

thesqlqueryrandomkey=form['thesqlqueryrandomkey'].value
userids=''
thesqlquery=''
if form.has_key('userids'):
    userids=form['userids'].value
    userids=userids.split(':')
else:
    pass
cursor.execute("""SELECT Query_string FROM
monr_temp_sql_query_storage WHERE Random_string='%s'"""%(thesqlqueryrandomkey))
output=cursor.fetchall()
thesqlquery=output[0][0]
cursor.execute("""INSERT IGNORE INTO
monr_sql_queries(User_id,Query_name)
VALUES('%s','%s')"""%(current_user_id,nameofthequery))
cursor.execute("""UPDATE monr_sql_queries SET
Query_string='%s',Query_description='%s',Saved_time=CURRENT_TIMESTAMP WHERE
User_id='%s' AND
Query_name='%s'"""%(sql.escape_string(thesqlquery),sql.escape_string(descriptionof
thequery),current_user_id,nameofthequery))
connection.commit()
cursor.close()
connection.close()
connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
cursor = connection.cursor()
set_character_set="""SET NAMES 'utf8' COLLATE
'utf8_bin'"""
cursor.execute(set_character_set)
for i in range(len(userids)):
    cursor.execute("""INSERT IGNORE INTO
monr_shared_queries(Shared_with,Shared_by,Query_name)
VALUES('%s','%s','%s')"""%(userids[i],current_user_id,nameofthequery))
connection.commit()
cursor.close()
connection.close()
print 'Content-Type: text/javascript\n'
print
"""alladvancedExportPopup.document.getElementById('querywrapper').innerHTML='<table
style="width:100%;height:100%"><tr><td class="inlinequerymessage">Query
Saved</td></tr></table>';"""
#print """closeSaveQueryForm(1000);"""
#print
"""alladvancedExportPopupShowmsg('quersavemsg','Query Saved');"""
print """clearQuerySaveMsg(2);"""
elif typeofform=='saveeditedsqlquery':
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
cursor = connection.cursor()
set_character_set="""SET NAMES 'utf8' COLLATE
'utf8_bin'"""
cursor.execute(set_character_set)
old_query_name=form['old_query_name'].value
nameofthequery=form['nameofthequery'].value
descriptionofthequery=form['descriptionofthequery'].value
userids=''
olduserids=''
if form.has_key('old_user_string'):

```

```

        olduserids=form['old_user_string'].value
        olduserids=olduserids.split(':')
    else:
        pass
    if form.has_key('userids'):
        userids=form['userids'].value
        userids=userids.split(':')
    else:
        pass
    cursor.execute("""UPDATE monr_sql_queries SET
Query_name='%s',Query_description='%s' WHERE User_id='%s' AND
Query_name='%s'""%(nameofthequery,sql.escape_string(descriptionofthequery),current
t_user_id,old_query_name))
    connection.commit()
    cursor.close()
    connection.close()
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)

    cursor = connection.cursor()
    set_character_set="""SET NAMES 'utf8' COLLATE
'utf8_bin'"""

    cursor.execute(set_character_set)
    for userid in olduserids:
        if not userid in userids:
            cursor.execute("""DELETE FROM monr_shared_queries
WHERE Shared_with='%s' AND Shared_by='%s' AND
Query_name='%s'""%(userid,current_user_id,nameofthequery))
        else:
            pass
    for userid in userids:
        if not userid in olduserids:
            cursor.execute("""INSERT IGNORE INTO
monr_shared_queries(Shared_with,Shared_by,Query_name)
VALUES('%s','%s','%s')""%(userid,current_user_id,nameofthequery))
        else:
            pass
    connection.commit()
    cursor.close()
    connection.close()
    print 'Content-Type: text/javascript\n'
    print
    """document.getElementById('quersavemessage').innerHTML='<table
style="width:100%;height:100%"><tr><td class="inlinequerymessage">Query
Saved</td></tr></table>';"""
    print """loadMyQueries();"""
    print
    """document.getElementById('quersavemessage').style.height='300px';"""
    print
    """document.getElementById('closebox').style.height='158px';"""
    print
    """document.getElementById('closebox').style.display='block';"""
    #print """closeSaveQueryForm(1000);"""
    #print
    """alladvancedExportPopupShowmsg('quersavemsg','Query Saved');"""
    #print """clearQueryeditMsg(2);"""
    elif typeofform=='savesharedsqlquery':
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)

```

```

        cursor = connection.cursor()
        set_character_set="""SET NAMES 'utf8' COLLATE
'utf8_bin'"""

        cursor.execute(set_character_set)
        old_query_name=form['old_query_name'].value
        nameofthequery=form['nameofthequery'].value
        descriptionofthequery=form['descriptionofthequery'].value
        createduser=form['shared_by'].value
        userids=''
        olduserids=[]
        thesqlquery=''
        cursor.execute("""SELECT Query_string FROM
monr_sql_queries WHERE User_id='%s' AND
Query_name='%s'"""%(createduser,nameofthequery))
        output=cursor.fetchall()
        thesqlquery=output[0][0]
        if form.has_key('userids'):
            userids=form['userids'].value
            userids=userids.split(':')
        else:
            pass
        cursor.execute("""INSERT IGNORE INTO
monr_sql_queries(User_id,Query_name)
VALUES('%s','%s')""%(current_user_id,nameofthequery))
        cursor.execute("""UPDATE monr_sql_queries SET
Query_string='%s',Query_description='%s',Saved_time=CURRENT_TIMESTAMP WHERE
User_id='%s' AND
Query_name='%s'""%(sql.escape_string(thesqlquery),sql.escape_string(descriptionof
thequery),current_user_id,nameofthequery))
        connection.commit()
        cursor.close()
        connection.close()
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)

        cursor = connection.cursor()
        set_character_set="""SET NAMES 'utf8' COLLATE
'utf8_bin'"""

        cursor.execute(set_character_set)
        cursor.execute("""SELECT Shared_with FROM
monr_shared_queries WHERE Query_name='%s' AND
Shared_by='%s'""%(nameofthequery,current_user_id))
        output1=cursor.fetchall()
        for i in range(len(output1)):
            olduserids.append(output1[i][0])
        for userid in olduserids:
            if not userid in userids:
                cursor.execute("""DELETE FROM monr_shared_queries
WHERE Shared_with='%s' AND Shared_by='%s' AND
Query_name='%s'""%(userid,current_user_id,nameofthequery))
            else:
                pass
        for userid in userids:
            if not userid in olduserids:
                cursor.execute("""INSERT IGNORE INTO
monr_shared_queries(Shared_with,Shared_by,Query_name)
VALUES('%s','%s','%s')""%(userid,current_user_id,nameofthequery))
            else:
                pass

```

```

        connection.commit()
        cursor.close()
        connection.close()
        print 'Content-Type: text/javascript\n'
        print
        """document.getElementById('quersavemessage').innerHTML=<table
style="width:100%;height:100%"><tr><td class="inlinequerymessage">Query
Saved</td></tr></table>";"""
        print """loadSharedQueries();"""
        print
        """document.getElementById('quersavemessage').style.height='300px';"""
        print
        """document.getElementById('closebox').style.height='158px';"""
        print
        """document.getElementById('closebox').style.display='block';"""
        #print """closeSaveQueryForm(1000);"""
        #print
        """alladvancedExportPopupShowmsg('quersavemsg','Query Saved');"""
        #print """clearQueryeditMsg(2);"""
    else:
        pass
    except:
        if (typeofform=='saveeditedsqlquery' or
typeofform=='savessharedsqlquery'):
            print 'Content-Type: text/javascript\n'
            print
            """document.getElementById('querywrapper').innerHTML='An error occured. Please
contact data manager';"""
            #print """clearQueryeditMsg(4);"""
            elif typeofform=='savesqlquery':
                print 'Content-Type: text/javascript\n'
                print
                """alladvancedExportPopup.document.getElementById('querywrapper').innerHTML='An
error occured. Please contact data manager';"""
                print """clearQuerySaveMsg(4);"""
            #raise

#####
#####
#####
        elif(action=='deletequery'):
            try:
                Query_name=form['query_name'].value
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
                cursor = connection.cursor()
                set_character_set="""SET NAMES 'utf8' COLLATE 'utf8_bin'"""
                cursor.execute(set_character_set)
                cursor.execute("""DELETE FROM monr_sql_queries WHERE
User_id='%s' AND Query_name='%s'"""%(current_user_id,Query_name))
                connection.commit()
                cursor.close()
                connection.close()
                print 'Content-Type: text/javascript\n'
                print """document.getElementById('message1').innerHTML='Query
successfully deleted';"""
                print """loadMyQueries();"""
            except:

```

```

        print 'Content-Type: text/javascript\n'
        print ""document.getElementById('message1').innerHTML='An
error occured. Please contact data manager';""
        #raise

#####
#####
#####
        elif(action=='removesharedquery'):
            try:
                Query_name=form['query_name'].value
                Shared_by=form['shared_by'].value
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
                cursor = connection.cursor()
                set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""
                cursor.execute(set_character_set)
                cursor.execute("""DELETE FROM monr_shared_queries WHERE
Shared_with='%s' AND Shared_by='%s' AND
Query_name='%s'""%(current_user_id,Shared_by,Query_name))
                connection.commit()
                cursor.close()
                connection.close()
                print 'Content-Type: text/javascript\n'
                print ""document.getElementById('message1').innerHTML='Query
successfully removed from your list';""
                print ""loadSharedQueries();""
            except:
                print 'Content-Type: text/javascript\n'
                print ""document.getElementById('message1').innerHTML='An
error occured. Please contact data manager';""
                #raise

#####
#####
#####
        elif(action=="loadmyqueries"):
            try:
                querytable=""<table id="myqueriestable"><thead><tr><th
title="Click here to sort by Query Name">Query Name</th><th title="Click here to
sort by query saving time">Saved on</th><th style="cursor:default">Shared
with</th><th style="cursor:default">Description</th><th
style="cursor:default;width:auto"></th></tr></thead><tbody>""
                row="evenRow"
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
                cursor = connection.cursor()
                set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""
                cursor.execute(set_character_set)
                cursor.execute("""SELECT
Query_name,Saved_time,Query_description FROM monr_sql_queries WHERE
User_id='%s'""%(current_user_id))
                output=cursor.fetchall()
                for i in range(len(output)):
                    if (row=="oddRow"):
                        row = "evenRow"
                    else:
                        row = "oddRow"

```

```

        Query_name=output[i][0]
        Saved_time=output[i][1]
        Query_description=output[i][2]
        cursor.execute("""SELECT Shared_with FROM
monr_shared_queries WHERE Shared_by='%s' AND
Query_name='%s'"""%(current_user_id,Query_name))
        output2=cursor.fetchall()
        Shared_with="None"
        if len(output2)>1:
            Shared_with=str(len(output2))+ " users"
        elif len(output2)==1:
            Shared_with=output2[0][0]
            requestedtable="sccl_authentication"
            cursor.execute("""SELECT
User_first_name,User_middle_initial,User_last_name FROM `s` WHERE
User_id='%s'""%(requestedtable,Shared_with))
            output1=cursor.fetchall()
            if len(output1)>0:
                for firstname,middleinitial,lastname in output1:
                    if not (middleinitial==None or
middleinitial==' ' or len(middleinitial)<1):
                        middleinitial=' '+middleinitial+'.'
                    else:
                        middleinitial=''
                    lastname=' '+lastname
                    Shared_with=firstname+middleinitial+lastname+'
('+Shared_with+')'
                else:
                    pass
            else:
                pass
        querytable=querytable+""<tr class="%s"><td id="%s|%s"
class="queryname" onclick="javascript:queryResultWindow(this.id);">%s</td><td
class="savedat">%s</td><td class="user">%s</td><td class="description">%s</td><td
class="holdicons"><div class="deletequery" title="Delete this
query">&nbsp;</div><div class="editquery" title="Edit the details of this
query">&nbsp;</div></td></tr>""%(row,Query_name,current_user_id,Query_name,Saved_
time,Shared_with,Query_description)
        querytable=querytable+""</tbody></table>""
        connection.commit()
        cursor.close()
        connection.close()
        querytable=querytable.replace("\'","\\\'")
        querytable=querytable.replace("\n", "<br />")
        if len(output)>0:
            print 'Content-Type: text/javascript\n'
            print
            ""document.getElementById('holdquerytable').innerHTML='%s';""%(querytable)
            print ""makeMyqueriesTableSortable();""
            print ""editQuery();""
            print ""deleteQuery();""
        else:
            print 'Content-Type: text/javascript\n'
            #print
            ""document.getElementById('holdquerytable').style.textAlign='center';""
            print
            ""document.getElementById('holdquerytable').innerHTML='No queries saved yet';""
    except:

```

```

        print 'Content-Type: text/javascript\n'
        print ""document.getElementById('message1').innerHTML='An
error occured. Please contact data manager';""
        #raise

#####
#####
#####
        elif(action=="loadsharedqueries"):
            try:
                querytable=""<table id="myqueriestable"><thead><tr><th
title="Click here to sort by Query Name">Query Name</th><th title="Click here to
sort by query saving time">Saved on</th><th>Shared by</th><th
style="cursor:default">Description</th><th
style="cursor:default"></th></tr></thead><tbody>""
                row="evenRow"
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
                cursor = connection.cursor()
                set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""
                cursor.execute(set_character_set)
                #cursor.execute("""SELECT
Query_name,Saved_time,Query_description,User_id FROM monr_sql_queries WHERE
(Shared_with LIKE('%%s%%') OR Shared_with LIKE('allusers')) AND
User_id!='s'""%(current_user_id,current_user_id))
                cursor.execute("""SELECT Query_name,Shared_by,Shared_time FROM
monr_shared_queries WHERE Shared_with='s'""%(current_user_id))
                output=cursor.fetchall()
                for Query_name,Shared_by,Shared_time in output:
                    if (row=="oddRow"):
                        row = "evenRow"
                    else:
                        row = "oddRow"
                    cursor.execute("""SELECT Query_description FROM
monr_sql_queries WHERE User_id='s' AND Query_name='s'""%(Shared_by,Query_name))
                    output1=cursor.fetchall()
                    Query_description=output1[0][0]
                    requestedtable="sccl_authentication"
                    Shared_by_id=Shared_by
                    cursor.execute("""SELECT
User_first_name,User_middle_initial,User_last_name FROM `s` WHERE
User_id='s'""%(requestedtable,Shared_by))
                    output2=cursor.fetchall()
                    if len(output2)>0:
                        for firstname,middleinitial,lastname in output2:
                            if not (middleinitial==None or middleinitial=='
or len(middleinitial)<1):
                                middleinitial=' '+middleinitial+'.'
                            else:
                                middleinitial=''
                                lastname=' '+lastname
                                Shared_by=firstname+middleinitial+lastname+'
('+Shared_by+')'
                    else:
                        pass
                querytable=querytable+""<tr class="%s"><td id="%s|%s"
class="queryname" onclick="javascript:queryResultWindow(this.id);">%s</td><td
class="savedat">%s</td><td class="user" id="%s">%s</td><td

```

```

class="description">%s</td><td class="holdicons"><div class="removequery"
title="Remove the shared query">&nbsp;</div><div class="savequerytomyqueries"
title="Save this query to your 'My Queries'
section">&nbsp;</div></td></tr>""%(row,Query_name,Shared_by_id,Query_name,Shared_
time,Shared_by_id,Shared_by,Query_description)
querytable=querytable+""</tbody></table>""
connection.commit()
cursor.close()
connection.close()
querytable=querytable.replace("\'","\\'\")
querytable=querytable.replace("\n","<br />")
if len(output)>0:
    print 'Content-Type: text/javascript\n'
    print
""document.getElementById('holdquerytable').innerHTML='%s';""%(querytable)
    print ""makeSharedQueriesTableSortable();""
    print ""saveToMyQueries();""
    print ""removeSharedQuery();""
else:
    print 'Content-Type: text/javascript\n'
    #print
""document.getElementById('holdquerytable').style.textAlign='center';""
    print
""document.getElementById('holdquerytable').innerHTML='None of the users shared
their queries yet';""
except:
    print 'Content-Type: text/javascript\n'
    print ""document.getElementById('message1').innerHTML='An
error occured. Please contact data manager';""
    #raise

#####
#####
#####
elif(action=="displayqueryresults"):
    try:
        Query_name=form['queryname'].value
        createduser=form['createduser'].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
        cursor = connection.cursor()
        set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""
        cursor.execute(set_character_set)
        cursor.execute("""SELECT Query_string FROM monr_sql_queries
WHERE User_id='%s' AND Query_name='%s'""%(createduser,Query_name))
        output=cursor.fetchall()
        lowerlimit = int(form['lowerlimit'].value)
        starting_row = lowerlimit+1
        numberofdatapoints = int(form['numberofdatapoints'].value)
        ending_row = lowerlimit+numberofdatapoints
        finalquery=output[0][0]
        nolimitquery=finalquery
        finalquery=finalquery+" LIMIT
%d,%d"%(lowerlimit,numberofdatapoints)
        cursor.execute(finalquery)
        output=cursor.fetchall()
        thelength = len(output)
        if(numberofdatapoints > thelength):

```



```

        ending_row = lowerlimit+thelength
else:
    pass
if ending_row==0:
    starting_row=0
else:
    pass
connection.commit()
cursor.close()
connection.close()

#####
#####

row="oddRow"
display_data=''
headers=[]
headercolumn=''
datarows=''

#####
#####

the_query=nolimitquery.split('FROM')
the_query=(the_query[0]).replace('SELECT ','')
the_query=the_query.replace('.', '@')
the_query=the_query.replace(',','#')
the_query=the_query.split('#')
column_array=[]
for i in range(len(the_query)):
    table_column=the_query[i].split('@')
    column_name=table_column[-1]
    column_name=column_name.replace(' ','')
    headers.append(column_name)

#####
#####

raj_excel_export.ldb=ldb;raj_excel_export.lhost=lhost;raj_excel_export.luser=luser
;raj_excel_export.lpasswd=lpasswd;raj_excel_export.lport=lport;raj_excel_export.st
udyname=studyname
    tempdirectoryname=form['tempdirectoryname'].value
    tempbasedir='/project/beilmanlab.umn.edu-
upload/MONR/temp_files/'
    if tempdirectoryname=='null':
        try:
            for the_folder in os.listdir(tempbasedir):
                if re.search(current_user_id,the_folder):
                    for the_file in
os.listdir(tempbasedir+the_folder):
os.remove(tempbasedir+the_folder+'/'+the_file)
    os.rmdir(tempbasedir+the_folder)
        else:
            pass
    except:
        pass
        #raise
tempdirectoryname=current_user_id+randkey(7,10)
store_save_dir=tempbasedir+tempdirectoryname+'/'

```

```

try:
    #Make sure the path exists
    if not os.path.exists(tempbasedir):
        os.umask(0)
        os.makedirs(store_save_dir,0771)
    else:
        pass
    if not os.path.exists(store_save_dir):
        os.umask(0)
        os.makedirs(store_save_dir,0771)
    else:
        pass
except:
    pass
    ##raise
store_save_dir=(store_save_dir.replace('\\', '/'))
else:
    store_save_dir=tempbasedir+tempdirectoryname+'/'
    raj_excel_export.excel_save_dir=store_save_dir
    raj_excel_export.csv_save_dir=store_save_dir
    store_file_name=current_user_id+randkey(7,10)
    excel_file_name=store_file_name+'.xls'
    csv_file_name=store_file_name+'.csv'
    excel_file_path=store_save_dir+excel_file_name
    csv_file_path=store_save_dir+csv_file_name
    #try:
    match_tables='#It was used before in advanced export code but
there is no need for this anymore. So, assigned null

create_and_save_excel_and_csv(studname,store_file_name,nolimitquery,match_tables)
    excel_and_csv="""<div class="downloadicons"
style="float:left;padding-left:5px;padding-bottom:10px;padding-top:10px"
id="downloadexcel" title="Download this data as excel file"
onclick="javascript:opener.downloadFile('\\tempfiledownload\\',\\'%s\\',\\'%s\\');
">&nbsp;</div><div class="downloadicons" style="float:left;padding-
left:5px;padding-bottom:10px;padding-top:10px" id="downloadcsv" title="Download
this data as csv file"
onclick="javascript:opener.downloadFile('\\tempfiledownload\\',\\'%s\\',\\'%s\\');
">&nbsp;</div>""%(studname,excel_file_name,studname,csv_file_name)
    #except:
    #excel_and_csv="""Error in creating Excel and/or CSV
files""

queryresult_total_data_points=raj_excel_export.output_row_count
    for header in headers:

headercolumn=headercolumn+""<th><center><div>%s</div></center></th>""%(str(heade
r))

headerrow="""<tr class="%s">%s</tr>""%(row,headercolumn)
    for i in range(len(output)):
        if row=="oddRow":
            row="evenRow"
        else:
            row="oddRow"
        datafields=''
        for j in range(len(output[i])):
            dataoutput=str(output[i][j])
            if dataoutput=='None':

```

```

        dataoutput=''
    else:
        pass
        dataoutput=dataoutput.replace("\'","\\\'")
        dataoutput=dataoutput.replace('\n','<br />')
        datafields=datafields+""<td
title="%s"><center><div>%s</div></center></td>""%(headers[j],dataoutput)
        datarows=datarows+""<tr
class="%s">%s</tr>""%(row,datafields)
        display_data=headerrow+datarows

display_table=""<table><tbody>%s</tbody></table>""%(display_data)
    if len(output)==0:
        display_table="No data generated for the query. Please
modify the query and try again. If the problem persits, contact data manager"
    else:
        pass

#####
#####
        print 'Content-Type: text/javascript\n'
        print
        ""queryResultPopup.document.getElementById('message1').innerHTML='';""
        print
        ""queryResultPopup.document.getElementById('datatable').innerHTML =
's';""%(display_table)
        print
        ""queryresult_total_data_points=%d;pagination();""%(queryresult_total_data_point
s)
        print
        ""queryResultPopup.document.getElementById('message1').innerHTML='Showing rows
from %d to %d';""%(starting_row,ending_row)
        print ""tempdirectoryname='%s';""%(tempdirectoryname)
        print
        ""queryResultPopup.document.getElementById('holddownloadicons').innerHTML='%s'""
%(excel_and_csv)
    except:
        print 'Content-Type: text/javascript\n'
        print
        ""queryResultPopup.document.getElementById('message1').innerHTML='There was an
error. Please contact data manager.';""
        print
        ""queryResultPopup.document.getElementById('datatable').innerHTML='There was an
error. Please contact data manager.';""
        #raise

#####
#####
#####
        elif(action=="drbeilman_emailist"):
            try:
                if current_user_id=="beilm001" or current_user_id=="tokac006"
or current_user_id=="groeh001":
                    raj_excel_export.excel_base_dir="/project/beilmanlab.umn.edu-
upload/"+studynome+"/excel_files/"

                    raj_excel_export.excel_save_dir=raj_excel_export.excel_base_dir

```

```

create_paths=[]
create_paths.append(excel_save_dir)
try:
    #Make sure the path exists
    for the_dir in create_paths:
        if not os.path.exists(the_dir):
            os.umask(0)
            os.makedirs(the_dir,0771)
        else:
            pass
except:
    pass
    #raise

#####

raj_excel_export.ldb=ldb;raj_excel_export.lhost=lhost;raj_excel_export.luser=luser
;raj_excel_export.lpasswd=lpasswd;raj_excel_export.lport=lport;raj_excel_export.st
udyname=studyname

    excel_file_name="sccmaillist.xls"
    tables_match={'`t1`':'monr_sccm_email_confirmation'}
    the_query="""SELECT
`t1`.firstname,`t1`.lastname,`t1`.email_1,`t1`.email_2,`t1`.response,`t1`.date_of_
response,`t1`.primary_email_option FROM `monr_sccm_email_confirmation` AS `t1`
ORDER BY response ASC,firstname,lastname"""

create_and_save_excel(studyname,excel_file_name,the_query,tables_match)
    file_path=excel_save_dir+excel_file_name

#####

connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
row = "odd"
cursor = connection.cursor()
cursor.execute("""SELECT COUNT(*) FROM
monr_sccm_email_confirmation""")
total_members=cursor.fetchall()
total_members_number=total_members[0][0]
cursor.execute("""SELECT COUNT(*) FROM
monr_sccm_email_confirmation WHERE response='Yes'""")
output=cursor.fetchall()
responded_number=output[0][0]
emaillist_headers = """<table class="emaillist"><tbody><tr
class="holdHeaders"><td colspan="7">Dr. Beilman\\'s email list responses visible
to users beilm001, tokac006, groeh001 and dicki002 only</td></tr>
<tr class="holdHeaders"><td
colspan="7">Out of %d members, %d have responded till today. <span
onclick="javascript:downloadFile(\\'emaillistdownload\\',\\'%s\\',\\'%s\\',\\'%s\\
');" class="clickoption">Download Data</span></td></tr>
<tr class="holdHeaders"><td>First
Name</td><td>Last Name</td>
<td>Email 1</td><td>Responded</td>
<td>Responded Date</td><td>Primary
email

```

```

choice</td></tr>""%(total_members_number,responded_number,studyname,file_path,exc
el_file_name)

        emallistdata=''
        cursor.execute("""SELECT
firstname,lastname,email_1,email_2,response,date_of_response,primary_email_option
FROM monr_sccm_email_confirmation ORDER BY response ASC,firstname,lastname""")
        output1 = cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        for i in range(len(output1)):
            response='No'
            date_of_response='&nbsp;'
            primary_email_option='&nbsp;'
            firstname = output1[i][0]
            lastname = output1[i][1]
            email1 = output1[i][2]
            email2 = output1[i][3]
            if output1[i][4]=='Yes':
                response = 'Yes'
                date_of_response = output1[i][5]
                primary_email_option = output1[i][6]
            else:
                pass
            if (row=="odd"):
                emallistdata = emallistdata+'<tr
class="oddRow">'

                row = "even"
            else:
                emallistdata = emallistdata+'<tr
class="evenRow">'

                row = "odd"
            emallistdata = emallistdata+""<td style="text-
align:left">%s</td>\
                                <td style="text-
align:left">%s</td>\
                                <td style="text-
align:left">%s</td>\
                                <td style="text-
align:left">%s</td>\
                                <td style="text-
align:left">%s</td>\
                                <td style="text-
align:left">%s</td></tr>""%(firstname,lastname,email1,response,date_of_response,p
rimary_email_option)
            emallist =
emallist_headers+emallistdata+""</tbody></table>""
            print 'Content-Type: text/javascript\n'
            print
            ""document.getElementById('emallist').innerHTML='%s';""%(emallist)
            else:
            print 'Content-Type: text/javascript\n'
            print
            ""document.getElementById('emallist').innerHTML='&nbsp;';""
        except:
            print 'Content-Type: text/javascript\n'
            print ""document.getElementById('message2').innerHTML='An
error occurred. Please contact data manager';""

```

```

#raise

#####
#####
#####
elif(action=="survey_info"):
    try:
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        try:
            cursor.execute("""DESCRIBE monr_user_survey_info""")
        except:
            cursor.execute("""CREATE TABLE IF NOT EXISTS
`monr_user_survey_info` (
                                                                    `User_id`
varchar(15) NOT NULL,
                                                                    PRIMARY KEY
`Completed_survey` enum('Yes','No') NOT NULL default 'No',
                                                                    INDEX
(`User_id`),
                                                                    FOREIGN KEY
(`User_id`) REFERENCES `scc1_authentication` (`User_id`)
                                                                    ON DELETE
NO ACTION ON UPDATE NO ACTION
                                                                    )
ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
                                                                    """)
            cursor.execute("""SELECT User_id FROM
scc1_authentication""")
            userlist=cursor.fetchall()
            for i in range(len(userlist)):
                cursor.execute("""INSERT IGNORE INTO
monr_user_survey_info (User_id) VALUES('%s')"""%(userlist[i][0]))
            try:
                cursor.execute("""DESCRIBE monr_survey_completion_time""")
            except:
                cursor.execute("""CREATE TABLE IF NOT EXISTS
`monr_survey_completion_time` (
                                                                    PRIMARY KEY
`Serialnumber` MEDIUMINT NOT NULL AUTO_INCREMENT,
                                                                    )
`Survey_completion_time` datetime NOT NULL,
                                                                    )
ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;""")
            cursor.execute("""UPDATE monr_user_survey_info SET
Completed_survey='Yes' WHERE User_id='%s'""%(current_user_id))
            cursor.execute("""INSERT IGNORE INTO
monr_survey_completion_time (Survey_completion_time) VALUES (NOW())""")
            connection.commit()
            cursor.close()
            connection.close()
            print 'Content-Type: text/javascript\n'
            print ""document.getElementById("message2").innerHTML='''

```

```

        print """alert('Thank you for updating the information that
you have completed the survey. If you have done this in error, please contact
cSurge administrator.');
```

```

        print
        """document.getElementById('surveyholder').innerHTML='&nbsp;';"""
        print
        """document.getElementById('surveycompletionmessage').innerHTML='<div
id="surveymessage" class="surveyfinished">You have successfully finished the
survey. If you think this is an error, please contact cSurge
administrator.</div>'''
    except:
        if cursor:
            cursor.close()
        else:
            pass
        if connection:
            connection.close()
        else:
            pass
        print 'Content-Type: text/javascript\n'
        print """document.getElementById('message2').innerHTML='An
error occurred. Please contact data manager';"""
        #raise

#####
#####
#####
#####
        else:
            print 'Content-Type: text/javascript\n'
            print """document.getElementById('message1').innerHTML='Error in
AJAX script. Error No: f1';"""

#####
#####
#####
#####
        else:
            print 'Content-Type: text/javascript\n'
            print """document.getElementById('message1').innerHTML='Error in AJAX
script. Error No: f2';"""
        else:
            print 'Content-Type: text/javascript\n'
            print """window.location = '.'"""
        #except:
            #print 'Content-Type: text/javascript\n'
            #print """window.location = '.'"""
            #print sys.exc_info()[1]
    else:
        print 'Content-Type: text/javascript\n'
        print """window.location = '.'"""

```

Code written for labsdataview.py

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sccl_labsdataview import *

```

Code written for imported sctl_labsdataview module

```
#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql
import cgi
import cgi
#cgi.enable()
import _mysql
import re
import urllib
import string
import shutil
import os
import operator
import time
import datetime
import random
import bz2
import urllib2
import cookielib
import Cookie
from urllib2 import urlopen, Request
import httplib
#####
alphabet =
'abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz'
min = 24
max = 32
buscopan=''
for x in random.sample(alphabet,random.randint(min,max)):
    buscopan+=x
theKey = buscopan
#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
dbconnection.close()
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
#####
#####
abc = open("Common_Data.html",'r')
commondata = abc.read()
action=''
form = cgi.FieldStorage()
if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", "" ) )
    try:
        bloatvalue = a_cookie["bloat"].value
```



```

        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SELECT User_id FROM sccl_authentication WHERE
Temp_key='%s'""%(bloatvalue))
        output = cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        if (len(output)==1):
            print 'Content-Type: text/html\n\n'
            filehandle = urllib.urlopen('labsdataview.html')
            for lines in filehandle.readlines( ):
                s = lines
                print s
            filehandle.close()
        else:
            print ""Set-Cookie: bloat=; Path=/"
            print ""Set-Cookie: page=index; Path=/"
            print ""Set-Cookie: fileupload_Pig_ID=; Path=/"
            print ""Set-Cookie: jsnewpass=; Path=/"
            print ""Set-Cookie: jslogin=; Path=/"
            print ""Set-Cookie: validate_filename=; Path=/"
            print 'Status: 301 Moved Permanently'
            print 'Location: http://sccl.umn.edu\n\n'
    except:
        print ""Set-Cookie: bloat=; Path=/"
        print ""Set-Cookie: page=index; Path=/"
        print ""Set-Cookie: fileupload_Pig_ID=; Path=/"
        print ""Set-Cookie: jsnewpass=; Path=/"
        print ""Set-Cookie: jslogin=; Path=/"
        print ""Set-Cookie: validate_filename=; Path=/"
        print 'Status: 301 Moved Permanently'
        print 'Location: http://sccl.umn.edu\n\n'
else:
    print ""Set-Cookie: bloat=; Path=/"
    print ""Set-Cookie: page=index; Path=/"
    print ""Set-Cookie: fileupload_Pig_ID=; Path=/"
    print ""Set-Cookie: jsnewpass=; Path=/"
    print ""Set-Cookie: jslogin=; Path=/"
    print ""Set-Cookie: validate_filename=; Path=/"
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'

```

Code written for queryresultdisplay.py

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sccl_queryresultdisplay import *

```

Code written for imported sccl_queryresultdisplay module

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql
import cgi

```

```

import cgi
import cgi
import _mysql
import re
import urllib
import string
import shutil
import os
import operator
import time
import datetime
import random
import bz2
import urllib2
import cookielib
import Cookie
from urllib2 import urlopen, Request
import http

#####
#####
alphabet =
'abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz'
min = 24
max = 32
buscopan=''
for x in random.sample(alphabet,random.randint(min,max)):
    buscopan+=x
theKey = buscopan
#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
dbconnection.close()
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
#####
#####
abc = open("Common_Data.html",'r')
commondata = abc.read()
action=''
form = cgi.FieldStorage()
if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", ""))
    try:
        bloatvalue = a_cookie["bloat"].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SELECT User_id FROM sccl_authentication WHERE
Temp_key='%s'""%(bloatvalue))
        output = cursor.fetchall()
        connection.commit()

```

```

cursor.close()
connection.close()
if (len(output)==1):
    print 'Content-Type: text/html\n\n'
    filehandle = urllib.urlopen('queryresultdisplay.html')
    for lines in filehandle.readlines( ):
        s = lines
        print s
    filehandle.close()
else:
    print ""Set-Cookie: bloat=; Path="/"
    print ""Set-Cookie: page=index; Path="/"
    print ""Set-Cookie: fileupload_Pig_ID=; Path="/"
    print ""Set-Cookie: jsnewpass=; Path="/"
    print ""Set-Cookie: jslogin=; Path="/"
    print ""Set-Cookie: validate_filename=; Path="/"
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'
except:
    print ""Set-Cookie: bloat=; Path="/"
    print ""Set-Cookie: page=index; Path="/"
    print ""Set-Cookie: fileupload_Pig_ID=; Path="/"
    print ""Set-Cookie: jsnewpass=; Path="/"
    print ""Set-Cookie: jslogin=; Path="/"
    print ""Set-Cookie: validate_filename=; Path="/"
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'
else:
    print ""Set-Cookie: bloat=; Path="/"
    print ""Set-Cookie: page=index; Path="/"
    print ""Set-Cookie: fileupload_Pig_ID=; Path="/"
    print ""Set-Cookie: jsnewpass=; Path="/"
    print ""Set-Cookie: jslogin=; Path="/"
    print ""Set-Cookie: validate_filename=; Path="/"
    print 'Status: 301 Moved Permanently'
    print 'Location: http://sccl.umn.edu\n\n'

```

Code written for start.py

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sccl_start import *

```

Code written for imported sccl_start module

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql
import cgi
#import cgitb
#cgitb.enable()
import _mysql
import re
import urllib
import string
import shutil

```

```

import os
import operator
import time
import datetime
import random
import bz2
import urllib2
import cookielib
import Cookie
from urllib2 import urlopen, Request
#####
#####
def replacemachine(newfile,sourceText, replaceText):
    if (len(sourceText)==len(replaceText)):
        filex = open(newfile, "r") #Opens the file in read-mode
        text = filex.read() #Reads the file and assigns the value to a variable
        filex.close() #Closes the file (read session)
        filey = open(newfile, "w") #Opens the file again, this time in write-mode
        #filey.write(text.replace(sourceText[i], replaceText[i])) #replaces all
instances of our keyword
        for i in range(len(sourceText)):
            text=text.replace(sourceText[i], replaceText[i])
            filey.write(text)
        # and writes the whole output when done, wiping over the old contents of
the file
        filey.close() #Closes the file (write session)
    else:
        print 'Check the data entered for the text to be replaced'
#####
#####
#####
#directory = "../..//cgi-bin/SCCL_test"
#os.chdir(directory)
#####
#####
alphabet =
'abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz'
min = 24
max = 32
buscopan=''
for x in random.sample(alphabet,random.randint(min,max)):
    buscopan+=x
theKey = buscopan
#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
#####
#####
try: # Windows needs stdio set for binary mode.

```

```

import msvcrt
msvcrt.setmode (0, os.O_BINARY) # stdin = 0
msvcrt.setmode (1, os.O_BINARY) # stdout = 1
except ImportError:
    pass
#####
#####
def connect_to_db_and_return_result(the_query):
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
    cursor = connection.cursor()
    cursor.execute("""%s""the_query)
    queryresult=cursor.fetchall()
    connection.commit()
    cursor.close()
    connection.close()
    return queryresult
#####
#####
def which_survey_page(User_id):
    check_survey_completion_status="""SELECT Completed_survey FROM
monr_user_survey_info WHERE User_id='%s'""%(User_id)
    page='survey.html'
    try:

queryresult=connect_to_db_and_return_result(check_survey_completion_status)
        if (queryresult[0][0])=='Yes':
            page='survey_finished.html'
        else:
            pass
    except:
        pass
    return page
#####
#####
abc = open("Common_Data.html", 'r')
commondata = abc.read()
abc.close()
bbc = open("Admin_Data.html", 'r')
admindata = bbc.read()
bbc.close()
action=''
form = cgi.FieldStorage()
page = "index.html"
if (form.has_key('login')):
    if (form.has_key('user_id')) and (form.has_key('password')):
        User_id = form['user_id'].value
        User_id = User_id.lower()
        Password = form['password'].value
        try:
            connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lhb)
            cursor = connection.cursor()
            cursor.execute("""SELECT
User_id,Password,First_login,Group_name,TestOnly FROM sccl_authentication WHERE
User_id='%s' AND Password=(md5(sha1('%s')))"""%(User_id,Password))
            output = cursor.fetchall()
            connection.commit()

```

```

cursor.close()
connection.close()
try:
    if (len(output)>0):
        for c,d,e,f,g in output:
            if (c==User_id) and not (g=="Yes"):
                connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=ldb)
cursor = connection.cursor()
cursor.execute("""UPDATE sccl_authentication SET
Last_login_time=now(),Temp_key='%s' WHERE User_id='%s'"""%(theKey,User_id))
connection.commit()
cursor.close()
connection.close()
if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie(
os.environ.get("HTTP_COOKIE", ""))
    if (a_cookie['jslogin'].value=="Valid"):
        print ""Set-Cookie: bloat=%s;

Path=/"""%(theKey)

        print ""Set-Cookie: session_Pig_ID=;

Path=/"""

        print ""Set-Cookie: fileupload_filetype=;

try:
    page = a_cookie["page"].value+ ".html"
    if e=="Yes":
        page="changepassword.html"
        filehandle = urllib.urlopen(page)
        print ""Set-Cookie:

page=changepassword; Path=/"""

        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#greetings#",User_id+"
This is your first login. Please choose a new password ",s)
            print n
            filehandle.close()
    else:
        if(f=="Admin"):
            commondata=admindata
        else:
            pass
        if(page=="index.html"):
            page="home.html"
        elif(page=="manageusers.html"):
            if f=="Admin":
                page="manageusers.html"
            else:
                page="home.html"
        elif(page=="survey.html"):
            page=which_survey_page(User_id)
        else:
            pass
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines

```

```

re.sub("#Common_Data#",commondata,s)
logged in as "+User_id,n)

n =
n = re.sub("#greetings#", "You are

print n
filehandle.close()
except:
page="home.html"
if(f=="Admin"):
commondata=admindata
else:
pass
if(page=="manageusers.html"):
if f=="Admin":
page="manageusers.html"
else:
page="home.html"
else:
pass
if e=="Yes":
page="changepassword.html"
filehandle = urllib.urlopen(page)
print 'Set-Cookie:

print 'Content-Type: text/html\n\n'
for lines in filehandle.readlines( ):
s = lines
n =

n =
re.sub("#greetings#",User_id+", this is your first login. Please choose a new
password ",n)

print n
filehandle.close()
else:
filehandle = urllib.urlopen(page)
print 'Content-Type: text/html\n\n'
for lines in filehandle.readlines( ):
s = lines
n =

n = re.sub("#greetings#", "You are

print n
filehandle.close()
"""
filehandle = urllib.urlopen(page)
print 'Content-Type: text/html\n\n'
for lines in filehandle.readlines( ):
s = lines
n =

n = re.sub("#greetings#", "You are

print n
filehandle.close()
"""
#raise

```

```

else:
    filehandle = urllib.urlopen(page)
    print 'Content-Type: text/html\n\n'
    for lines in filehandle.readlines( ):
        s = lines
        n = re.sub("#Common_Data#",commondata,s)
        n = re.sub("#greetings#", "Javascript needs
to be enabled in order to login",n)
        print n
    filehandle.close()
else:
    filehandle = urllib.urlopen(page)
    print 'Content-Type: text/html\n\n'
    for lines in filehandle.readlines( ):
        s = lines
        n = re.sub("#Common_Data#",commondata,s)
        n = re.sub("#greetings#", "Browser Cookies and
javascript need to be enabled in order to login",n)
        print n
    filehandle.close()
else:
    filehandle = urllib.urlopen(page)
    print 'Content-Type: text/html\n\n'
    for lines in filehandle.readlines( ):
        s = lines
        n = re.sub("#Common_Data#",commondata,s)
        n = re.sub("#greetings#", "Login error. Please try
again or contact administrator",n)
        print n
    filehandle.close()
else:
    filehandle = urllib.urlopen(page)
    print 'Content-Type: text/html\n\n'
    for lines in filehandle.readlines( ):
        s = lines
        n = re.sub("#Common_Data#",commondata,s)
        n = re.sub("#greetings#", "Login error. Please try again or
contact administrator",n)
        print n
    filehandle.close()
except:
    filehandle = urllib.urlopen(page)
    print 'Content-Type: text/html\n\n'
    for lines in filehandle.readlines( ):
        s = lines
        n = re.sub("#Common_Data#",commondata,s)
        n = re.sub("#greetings#", "Login error. Please try again or
contact administrator",n)
        print n
    filehandle.close()
    #raise
except:
    filehandle = urllib.urlopen(page)
    print 'Content-Type: text/html\n\n'
    for lines in filehandle.readlines( ):
        s = lines
        n = re.sub("#Common_Data#",commondata,s)

```



```

        n = re.sub("#greetings#", "Login error. Please try again or contact
administrator",n)
        print n
        filehandle.close()
        #raise
    else:
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "Login error. Please try again or contact
administrator",n)
            print n
            filehandle.close()
    else:
        if (os.environ.has_key('HTTP_COOKIE')):
            a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", "") )
            try:
                bloatvalue = a_cookie["bloat"].value
                connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
                cursor = connection.cursor()
                cursor.execute("""SELECT User_id,First_login,Group_name,TestOnly FROM
scc1_authentication WHERE Temp_key='%s'""%(bloatvalue))
                output = cursor.fetchall()
                connection.commit()
                cursor.close()
                connection.close()
                if (len(output)>0):
                    User_id=''
                    for a,b,c,d in output:
                        User_id = a
                        if not (User_id=='') and not (d=="Yes"):
                            try:
                                page = a_cookie["page"].value+ ".html"
                                if(c=="Admin"):
                                    commondata=admindata
                                else:
                                    pass
                                if(page=="index.html"):
                                    page="home.html"
                                elif(page=="manageusers.html"):
                                    if (c=="Admin"):
                                        page="manageusers.html"
                                    else:
                                        page="home.html"
                                else:
                                    pass
                                if(page=="logout.html"):
                                    page="index.html"
                                filehandle = urllib.urlopen(page)
                                print ""Set-Cookie: bloat=; Path=/""
                                print ""Set-Cookie: page=index; Path=/""
                                print ""Set-Cookie: session_Pig_ID=; Path=/""
                                print ""Set-Cookie: fileupload_filetype=;

Path=/""

                                print ""Set-Cookie: jsnewpass=; Path=/""

```

```

        print ""Set-Cookie: jslogin=; Path="/"
        print ""Set-Cookie: validate_filename=; Path="/"
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#",User_id+", you have
successfully logged out",n)
            print n
            filehandle.close()
        elif(page=="changepassword.html"):
            if(form.has_key('changepassword')):
                try:
                    newpassword = form["newpassword"].value
                    javascriptverify =
a_cookie["jsnewpass"].value
                    if (javascriptverify=="Valid"):
                        connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=lhb)
                        cursor = connection.cursor()
                        cursor.execute("""SELECT First_login
FROM sccl_authentication WHERE User_id='%s'""%(User_id))
                        result=cursor.fetchall()
                        for value in result[0]:
                            if value=="Yes":
                                cursor.execute("""UPDATE
sccl_authentication SET Password=md5(sha1('%s')),First_login='No' WHERE
User_id='%s'""%(newpassword,User_id))
                                connection.commit()
                                cursor.close()
                                connection.close()
                                page = "home.html"
                                filehandle =
urllib.urlopen(page)
                                print ""Set-Cookie:
                                print ""Set-Cookie:
                                print 'Content-Type:
                                for lines in
                                    s = lines
                                    n =
re.sub("#Common_Data#",commondata,s)
                                    n =
re.sub("#greetings#", "Password change successful. You are logged in as
"+User_id,n)
                                    print n
                                    filehandle.close()
                                else:
                                    page="index.html"
                                    filehandle =
urllib.urlopen(page)
                                    print ""Set-Cookie:
                                    print ""Set-Cookie:
                                page=index; Path="/"

```

```

print 'Content-Type:
text/html\n\n'
filehandle.readlines( ):
    s = lines
    n =
re.sub("#greetings#",User_id+", could not change password. Please check the fine
print. If the error repeats, please contact administrator.",s)
    print n
    filehandle.close()
else:
    filehandle = urllib.urlopen(page)
    print ""Set-Cookie: jsnewpass=;
Path="/""
    print 'Content-Type: text/html\n\n'
    for lines in filehandle.readlines( ):
        s = lines
        n =
re.sub("#greetings#",User_id+", could not change password. Please check the fine
print. If the error repeats, please contact administrator.",s)
        print n
        filehandle.close()
except:
    filehandle = urllib.urlopen(page)
    print ""Set-Cookie: jsnewpass=; Path="/""
    print 'Content-Type: text/html\n\n'
    for lines in filehandle.readlines( ):
        s = lines
        n = re.sub("#greetings#",User_id+",
could not change password. Please check the fine print. If the error repeats,
please contact administrator.",s)
        print n
        filehandle.close()
    #raise
else:
    try:
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#greetings#",User_id+",
this is your first login. Please choose a new password",s)
            print n
            filehandle.close()
    except:
        page="home.html"
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n =
re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "You are
logged in as "+User_id,n)
            print n
            filehandle.close()
        #raise
elif(page=="survey.html"):

```

```

page=which_survey_page(User_id)
filehandle = urllib.urlopen(page)
print 'Content-Type: text/html\n\n'
for lines in filehandle.readlines( ):
    s = lines
    n = re.sub("#Common_Data#",commondata,s)
    n = re.sub("#greetings#", "You are logged in as

"+User_id,n)

    print n
    filehandle.close()
elif(page=="fileupload.html"):
    message2=''
    try:
        Pig_ID=a_cookie["session_Pig_ID"].value
        filetype=a_cookie["fileupload_filetype"].value
    except:
        Pig_ID=''
        filetype=''
        #raise
    if(form.has_key('fileuploadpage')):
        Pig_ID = form['pigids'].value
        filetype=form['typeoffile'].value
    try:

if(a_cookie["validate_filename"].value=="Valid"):
    # A nested FieldStorage instance holds
the file
        uploaded_file = form['fileupload']
        if(form["typeoffile"].value=="Vitals"
or form["typeoffile"].value=="Lab_Values" or
form["typeoffile"].value=="GCS_Scores" or form["typeoffile"].value=="Teg_data"):
            Data_type =
form["typeoffile"].value
        elif not(form["event"].value=="0"):
            Data_type =
form["typeoffile"].value+"_"+form["typeoftissue"].value+"_"+form["event"].value
        else:
            Data_type =
form["typeoffile"].value+"_"+form["typeoftissue"].value+"_"+form["event"].value+"_"
"+form["hours"].value+"_"+form["minutes"].value
            studyname = form['studyname'].value
            # Test if the file was uploaded
            File_name = ''
            File_type = ''
            if (uploaded_file.filename):
                # strip leading path from file
name to avoid directory traversal attacks
                File_name =
os.path.basename(uploaded_file.filename)
                File_name = File_name.replace("
", "")
                File_name =
                File_name =
                File_name =
                File_name =
                File_name.replace("(", "")
                File_name.replace(";", "")

```

```

File_name.replace(":", "")
File_name.replace("=", "")
File_name.replace("\'", "")
File_name.replace("\\"", "")
File_name.replace("%", "")
(File_name.split("\\\\"))[-1]
(form['typeoffile'].value=="Teg_data"):
store_buffer=uploaded_file.file.read()

File_name.split(".")[1]
or file_extension == "xlsx"):
"txt"):
"doc" or file_extension == "docx"):

file_extension

"/project/beilmanlab.umn.edu-upload/"+studynome+"/Files/"+Pig_ID+"/"+Data_type
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=ldb)
'utf8' COLLATE 'utf8_bin'")
IGNORE INTO monr_fileupload(Pig_ID,Data_type)
VALUES('%s', '%s')""%(Pig_ID,Data_type))

monr_fileupload SET
File_name='%s',File_storage_path='%s',Upload_time=CURRENT_TIMESTAMP,File_type='%s'
,File_size=%d WHERE Pig_ID='%s' AND
Data_type='%s'""%(File_name,File_storage_path,File_type,File_size,Pig_ID,Data_typ
e))

os.path.exists(File_storage_path):

os.makedirs(File_storage_path,0771)

File_name =
File_name =
File_name =
File_name =
File_name =
File_name =
File_name =
if not

try:
file_extension =
if(file_extension == "xls"
File_type = "MS Excel"
elif(file_extension ==
File_type = "Text"
elif(file_extension ==
File_type = "MS Word"
else:
File_type =

except:
File_type = "Un Known"
#raise
File_size = len(store_buffer)
File_storage_path =
connection =
cursor = connection.cursor()
#cursor.execute("""SET NAMES

cursor.execute("""INSERT

cursor.execute("""UPDATE

try:
#Make sure the path exists
if not
os.umask(0)

```

```

else:
    pass
except:
    pass
    #raise
for the_file in

try:

except:
    pass
    #raise
dest_file =

dest_file.write(store_buffer)
dest_file.close()
os.umask(0)

os.listdir(File_storage_path):

os.remove(File_storage_path+"/"+the_file)

open(File_storage_path+"/"+File_name,'wb')

os.chmod(File_storage_path+"/"+File_name,0644)
if(form['typeoffile'].value=="HNMR"):
    if
    (form['typeoftissue'].value=="Muscle" or form['typeoftissue'].value=="Liver"):
        for times in
        range(1,3):
            if times==1:
                table_type=''

set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""
else:

table_type='sas_'
set_character_set=""SET NAMES 'latin1'""
cursor.execute(set_character_set)
Event_name=form['event'].value
Event_name=='Other':
    if
        Event_name=99
    else:
        pass

typeoffile=(form['typeoffile'].value).lower()
typeoftissue=(form['typeoftissue'].value).lower()
data1=open(File_storage_path+"/"+File_name,'r').readlines()
data2=[]
data3=dict()
datalines=0
for line in data1:

datalines=datalines+1
if

(datalines>3) and not (len(line.split('\t'))<2):

```

```

data2.append(line)

data3[line.split('\t')[0]]=line.split('\t')[1]

else:
    pass
file_columns=[]
for line in data2:

file_columns.append(line.split('\t')[0])

cursor.execute("""CREATE TABLE IF NOT EXISTS `monr_%s%s_%s` (
`Pig_ID` varchar(6) NOT NULL,
`Event_name` INT( 2 ) NOT NULL,
`Normalization` ENUM('No','Wet_tissue','Lyophilized_tissue') DEFAULT NULL,
`Weight_of_tissue` FLOAT(8,4) DEFAULT NULL,
`Normalization_factor` FLOAT(8,4) DEFAULT NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`,`Normalization`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_%s%s_%s_ibfk_1` FOREIGN KEY (`Pig_ID`)
REFERENCES `monr_identifier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
)
ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin""%(table_type,typeoffile,typeoftissue,table_type,typeoffile,type
oftissue))

cursor.execute("""SELECT COLUMN_NAME FROM information_schema.columns WHERE
TABLE_NAME='monr_%s%s_%s'""%(table_type,typeoffile,typeoftissue))

output=cursor.fetchall()

db_columns=[]
for line in
    for column in
        if not
            (column=='Pig_ID') and not (column=='Event_name') and not
            (column=='Normalization') and not (column=='Weight_of_tissue') and not
            (column=='Normalization_factor'):
                db_columns.append(column)
        else:
            pass

#cursor.execute("""SHOW VARIABLES LIKE 'character_set%""")

#output1=cursor.fetchall()

#print output1

```

```

file_columns:
    for column in
        if not (column
in db_columns)and not(len(db_columns)>243):
            cursor.execute("""ALTER TABLE `monr_%s%s_%s` ADD COLUMN `%s` FLOAT(8,4) default
            NULL""%(table_type,typeoffile,typeoftissue,column))
            else:
                pass
            #for column in
            #if not column
db_columns:
            #if not column
in file_columns:
            #cursor.execute("""ALTER TABLE `monr_%s%s_%s` DROP COLUMN
            `%s`""%(table_type,typeoffile,typeoftissue,column))
            #else:
            #pass

if(Event_name=="0"):
    Event_name=str(Event_name)+"_"+form["hours"].value+"_"+form["minutes"].value
    else:
        pass
    for type in
range(0,3):
        if type==0:

Normalization="No"

    cursor.execute("""INSERT IGNORE INTO `monr_%s%s_%s`
    (Pig_ID,Event_name,Normalization)
    VALUES('%s','%s','%s')""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_name,No
    rmalization))
    for
metabolite in data3:
        try:

metabolite_value=float(data3[metabolite])

except:

metabolite_value=''

#raise
        if not
metabolite_value=='':

    cursor.execute("""UPDATE `monr_%s%s_%s` SET `%s`=%s WHERE Pig_ID='%s' AND
    Event_name='%s' AND
    Normalization='%s'""%(table_type,typeoffile,typeoftissue,metabolite,metabolite_va
    lue,Pig_ID,Event_name,Normalization))
        else:

pass
        elif type==1:

Normalization="Wet_tissue"

```



```

Weight_of_tissue=float(form['tissuewt'].value)

Normalization_factor=1000/Weight_of_tissue

cursor.execute("""INSERT IGNORE INTO `monr_%s%s_%s`
(Pig_ID,Event_name,Normalization)
VALUES('%s','%s','%s')""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_name,Normali
zation))

cursor.execute("""UPDATE `monr_%s%s_%s` SET
Weight_of_tissue=%.4f,Normalization_factor=%.4f WHERE Pig_ID='%s' AND
Event_name='%s' AND
Normalization='%s'""%(table_type,typeoffile,typeoftissue,Weight_of_tissue,Normali
zation_factor,Pig_ID,Event_name,Normalization))

metabolite in data3:
try:

metabolite_value=float(data3[metabolite])

metabolite_value_normalized=metabolite_value*Normalization_factor

except:

metabolite_value=''

metabolite_value_normalized=''

#raise
if not

metabolite_value_normalized=='':

cursor.execute("""UPDATE `monr_%s%s_%s` SET `s`=%.4f WHERE Pig_ID='%s' AND
Event_name='%s' AND
Normalization='%s'""%(table_type,typeoffile,typeoftissue,metabolite,metabolite_va
lue_normalized,Pig_ID,Event_name,Normalization))

else:

pass

elif type==2:
if

form.has_key('lyowt') and not form['lyowt'].value=='':

Normalization="Lyophilized_tissue"

Weight_of_tissue=float(form['lyowt'].value)

Normalization_factor=1000/Weight_of_tissue

cursor.execute("""INSERT IGNORE INTO `monr_%s%s_%s`
(Pig_ID,Event_name,Normalization)
VALUES('%s','%s','%s')""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_name,Normali
zation))

cursor.execute("""UPDATE `monr_%s%s_%s` SET
Weight_of_tissue=%.4f,Normalization_factor=%.4f WHERE Pig_ID='%s' AND
Event_name='%s' AND

```

```

Normalization='%s'""%(table_type,typeoffile,typeoftissue,Weight_of_tissue,Normali
zation_factor,Pig_ID,Event_name,Normalization))
                                                                 for
metabolite in data3:

try:

metabolite_value=float(data3[metabolite])

metabolite_value_normalized=metabolite_value*Normalization_factor

except:

metabolite_value=''

metabolite_value_normalized=''

#raise
                                                                 if
not metabolite_value_normalized=='':

cursor.execute("""UPDATE `monr_%s%s_%s` SET `s`=%s WHERE Pig_ID='%s' AND
Event_name='%s' AND
Normalization='%s'""%(table_type,typeoffile,typeoftissue,metabolite,metabolite_va
lue_normalized,Pig_ID,Event_name,Normalization))

else:

pass
                                                                 else:
                                                                 pass

elif(form['typeoftissue'].value=="Urine" or form['typeoftissue'].value=="Serum"):
                                                                 for times in
range(1,3):
                                                                 if times==1:
                                                                 table_type=''

set_character_set=""SET NAMES 'utf8' COLLATE 'utf8_bin'""

                                                                 else:

table_type='sas_'

set_character_set=""SET NAMES 'latin1'""

cursor.execute(set_character_set)

Event_name=form['event'].value
                                                                 if
Event_name=='Other':
                                                                 Event_name=99
                                                                 else:
                                                                 pass

typeoffile=(form['typeoffile'].value).lower()

typeoftissue=(form['typeoftissue'].value).lower()

```

```

Normalization_factor=float(form['normfact'].value)

data1=open(File_storage_path+"/"+File_name,'r').readlines()

data2=[]
data3=dict()
datalines=0
for line in data1:

    datalines=datalines+1

    if
        (datalines>3) and not (len(line.split('\t'))<2):
            data2.append(line)
            data3[line.split('\t')[0]]=line.split('\t')[1]
        else:
            pass
            file_columns=[]
            for line in data2:

file_columns.append(line.split('\t')[0])

cursor.execute("""CREATE TABLE IF NOT EXISTS `monr_%s_%s_%s` (
`Pig_ID` varchar(6) NOT NULL,
`Event_name` INT( 2 ) NOT NULL,
`Normalization` ENUM('No','Yes') DEFAULT NULL,
`Normalization_factor` FLOAT(8,4) DEFAULT NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`,`Normalization`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_%s_%s_%s_ibfk_1` FOREIGN KEY (`Pig_ID`)
REFERENCES `monr_identifier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
)
ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin""%(table_type,typeoffile,typeoftissue,table_type,typeoffile,type
oftissue))

cursor.execute("""SELECT COLUMN_NAME FROM information_schema.columns WHERE
TABLE_NAME='monr_%s_%s_%s'""%(table_type,typeoffile,typeoftissue))

output=cursor.fetchall()

db_columns=[]
for line in
    for column in
        if not
            (column=='Pig_ID') and not (column=='Event_name') and not
            (column=='Normalization') and not (column=='Normalization_factor'):

```

```

db_columns.append(column)

else:
    pass

#cursor.execute("""SHOW VARIABLES LIKE 'character_set%'""")
#output1=cursor.fetchall()

#print output1
for column in

file_columns:
    if not (column
in db_columns) and not(len(db_columns)>243):

cursor.execute("""ALTER TABLE `monr_%s_%s_%s` ADD COLUMN `%s` FLOAT(8,4) default
NULL"""%(table_type,typeoffile,typeoftissue,column))

else:
    pass
#for column in

db_columns:
    #if not column

in file_columns:

#cursor.execute("""ALTER TABLE `monr_%s_%s_%s` DROP COLUMN
`s`""%(table_type,typeoffile,typeoftissue,column))

#else:
    #pass

if(Event_name=="0"):

Event_name=str(Event_name)+"_"+form["hours"].value+"_"+form["minutes"].value
else:
    pass
for type in

range(0,2):
    if type==0:

Normalization="No"

cursor.execute("""INSERT IGNORE INTO `monr_%s_%s_%s`
(Pig_ID,Event_name,Normalization)
VALUES('%s','%s','%s')""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_name,No
rmalization))

for

metabolite in data3:
    try:

metabolite_value=float(data3[metabolite])

except:

metabolite_value=''

#raise

if not

metabolite_value=='':

cursor.execute("""UPDATE `monr_%s_%s_%s` SET `%s`=%s WHERE Pig_ID='%s' AND

```

```

Event_name='%s' AND
Normalization='%s'""""%(table_type,typeoffile,typeoftissue,metabolite,metabolite_val
ue,Pig_ID,Event_name,Normalization))
else:

pass

elif type==1:

Normalization="Yes"

cursor.execute("""INSERT IGNORE INTO `monr_%s%s_%s`
(Pig_ID,Event_name,Normalization)
VALUES('%s','%s','%s')""""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_name,No
rmalization))

cursor.execute("""UPDATE `monr_%s%s_%s` SET Normalization_factor=%.4f WHERE
Pig_ID='%s' AND Event_name='%s' AND
Normalization='%s'""""%(table_type,typeoffile,typeoftissue,Normalization_factor,Pig
_ID,Event_name,Normalization))

for
metabolite in data3:
try:

metabolite_value=float(data3[metabolite])

metabolite_value_normalized=metabolite_value*Normalization_factor

except:

metabolite_value=''

metabolite_value_normalized=''

#raise
if not
metabolite_value_normalized=='':

cursor.execute("""UPDATE `monr_%s%s_%s` SET `%s`=%.4f WHERE Pig_ID='%s' AND
Event_name='%s' AND
Normalization='%s'""""%(table_type,typeoffile,typeoftissue,metabolite,metabolite_val
ue_normalized,Pig_ID,Event_name,Normalization))
else:

pass

else:
pass

elif(form['typeoffile'].value=="PNMR"):
#if
(form['typeoftissue'].value=="Muscle" or form['typeoftissue'].value=="Liver"):
for times in range(1,3):
if times==1:
table_type=''

set_character_set="""SET NAMES 'utf8' COLLATE 'utf8_bin'""""
else:
table_type='sas_'

```

```

set_character_set=""SET NAMES 'latin1'""
cursor.execute(set_character_set)
Event_name=form['event'].value
typeoffile=(form['typeoffile'].value).lower()
typeoftissue=(form['typeoftissue'].value).lower()
data1=open(File_storage_path+"/"+File_name,'r').readlines()
data2=[]
data3=dict()
data4=dict()
data5=dict()
datalines=0
for line in data1:
    if
re.search(r'\s\d+\s',line):
data2.append(line)
datalines=datalines+1
data3[line.split()[0]]=line.split()[3]
data4[line.split()[0]]=line.split()[1]
data5[line.split()[0]]=line.split()[2]
else:
    pass
file_columns=[]
for line in data2:
file_columns.append(line.split()[0])
cursor.execute("""CREATE TABLE IF NOT EXISTS `monr_%s_%s_%s` (
`Pig_ID` varchar(6) NOT NULL,
`Event_name` INT( 2 ) NOT NULL,
`Normalization` ENUM('No','Wet_tissue','Lyophilized_tissue') DEFAULT NULL,
`Weight_of_tissue` FLOAT(8,4) DEFAULT NULL,
`Normalization_factor` FLOAT(8,4) DEFAULT NULL,
`Value_type` ENUM('Start_ppm','End_ppm','Integral') DEFAULT NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`,`Normalization`,`Value_type`),
`Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_%s_%s_%s_ibfk_1` FOREIGN KEY (`Pig_ID`)
REFERENCES `monr_identifiaer_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE

```

KEY

```

ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin""%(table_type,typeoffile,typeoftissue,table_type,typeoffile,type
oftissue))

cursor.execute("""SELECT COLUMN_NAME FROM information_schema.columns WHERE
TABLE_NAME='monr_%s%s_%s'""%(table_type,typeoffile,typeoftissue))

output=cursor.fetchall()

base_columns=['Pig_ID','Event_name','Normalization','Weight_of_tissue','Normalizat
ion_factor','Value_type']

db_columns=[]
for line in output:
    for column in
        if not column
            else:
                pass

line:
in base_columns:
db_columns.append(column)

#cursor.execute("""SHOW VARIABLES LIKE 'character_set%'""")
#output1=cursor.fetchall()
#print output1
for column in
    if not (column in
db_columns)and not(len(db_columns)>243):
cursor.execute("""ALTER TABLE `monr_%s%s_%s` ADD COLUMN `%s` DOUBLE(14,8) default
NULL""%(table_type,typeoffile,typeoftissue,column))
    else:
        pass
#for column in
db_columns:
    #if not column in
file_columns:
#cursor.execute("""ALTER TABLE `monr_%s%s_%s` DROP COLUMN
`s`""%(table_type,typeoffile,typeoftissue,column))
    #else:
        #pass
if(Event_name=="0"):
Event_name=str(Event_name)+"_"+form["hours"].value+"_"+form["minutes"].value
    else:
        pass

Value_types=['Start_ppm','End_ppm','Integral']
for type in
range(0,3):
    if type==0:

Normalization="No"

```

```

                                                                    for valuetype
in Value_types:

cursor.execute("""INSERT IGNORE INTO `monr_%s%s_%s`
(Pig_ID,Event_name,Normalization,Value_type)
VALUES('%s','%s','%s','%s')"""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_na
me,Normalization,valuetype))

                                                                    for peak in
data3:
                                                                    try:

Integral=float(data3[peak])

Start_ppm=float(data4[peak])

End_ppm=float(data5[peak])

                                                                    except:

Integral=''

Start_ppm=''

End_ppm=''

                                                                    #raise
                                                                    if not

(Integral==' or Start_ppm==' or End_ppm==' ):

#cursor.execute("""INSERT IGNORE INTO `monr_%s%s_%s`
(Pig_ID,Event_name,Normalization) VALUES('%s','%s','%s') WHERE Pig_ID='%s' AND
Event_name='%s' AND
Normalization='%s'""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_name,Normal
ization))

cursor.execute("""UPDATE `monr_%s%s_%s` SET `%s`=%f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='Integral'""%(table_type,typeoffile,typeoftissue,peak,Integral,Pig_ID,
Event_name,Normalization))

cursor.execute("""UPDATE `monr_%s%s_%s` SET `%s`=%f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='Start_ppm'""%(table_type,typeoffile,typeoftissue,peak,Start_ppm,Pig_I
D,Event_name,Normalization))

cursor.execute("""UPDATE `monr_%s%s_%s` SET `%s`=%f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='End_ppm'""%(table_type,typeoffile,typeoftissue,peak,End_ppm,Pig_ID,Ev
ent_name,Normalization))

                                                                    else:
                                                                    pass
elif type==1:

Normalization="Wet_tissue"

Weight_of_tissue=float(form['tissuewt'].value)

Normalization_factor=1000/Weight_of_tissue

#cursor.execute("""INSERT IGNORE INTO `monr_%s%s_%s`

```



```

(Pig_ID,Event_name,Normalization)
VALUES('%s','%s','%s')""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_name,Normalization))
                                                                    for valuetype
in Value_types:

cursor.execute("""INSERT IGNORE INTO `monr_%s_%s_%s`
(Pig_ID,Event_name,Normalization,Value_type)
VALUES('%s','%s','%s','%s')""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_name,Normalization,valuetype))

cursor.execute("""UPDATE `monr_%s_%s_%s` SET
Weight_of_tissue=%.4f,Normalization_factor=%.4f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='%s'""%(table_type,typeoffile,typeoftissue,Weight_of_tissue,Normalization_factor,Pig_ID,Event_name,Normalization,valuetype))
                                                                    for peak in
data3:
                                                                    try:

Integral=float(data3[peak])*Normalization_factor

Start_ppm=float(data4[peak])

End_ppm=float(data5[peak])
                                                                    except:

Integral=''

Start_ppm=''

End_ppm=''
                                                                    #raise
                                                                    if not
(Integral==' or Start_ppm==' or End_ppm==' ):

#cursor.execute("""UPDATE `monr_%s_%s_%s` SET `s`=%.7f WHERE Pig_ID='%s' AND
Event_name='%s' AND
Normalization='%s'""%(table_type,typeoffile,typeoftissue,peak,peak_value_normalized,Pig_ID,Event_name,Normalization))

cursor.execute("""UPDATE `monr_%s_%s_%s` SET `s`=%f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='Integral'""%(table_type,typeoffile,typeoftissue,peak,Integral,Pig_ID,Event_name,Normalization))

cursor.execute("""UPDATE `monr_%s_%s_%s` SET `s`=%f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='Start_ppm'""%(table_type,typeoffile,typeoftissue,peak,Start_ppm,Pig_ID,Event_name,Normalization))

cursor.execute("""UPDATE `monr_%s_%s_%s` SET `s`=%f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='End_ppm'""%(table_type,typeoffile,typeoftissue,peak,End_ppm,Pig_ID,Event_name,Normalization))
                                                                    else:
                                                                    pass
                                                                    elif type==2:

```

```

                                                                    if
form.has_key('lyowt') and not form['lyowt'].value=='':

Normalization="Lyophilized_tissue"

Weight_of_tissue=float(form['lyowt'].value)

Normalization_factor=1000/Weight_of_tissue

#cursor.execute("""INSERT IGNORE INTO `monr_%s%s_%s`
(Pig_ID,Event_name,Normalization)
VALUES('%s','%s','%s')"""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_name,No
rmalization))

#cursor.execute("""UPDATE `monr_%s%s_%s` SET
Weight_of_tissue=%.4f,Normalization_factor=%.4f WHERE Pig_ID='%s' AND
Event_name='%s' AND
Normalization='%s'""%(table_type,typeoffile,typeoftissue,Weight_of_tissue,Normali
zation_factor,Pig_ID,Event_name,Normalization))
                                                                    for
valuetype in Value_types:

cursor.execute("""INSERT IGNORE INTO `monr_%s%s_%s`
(Pig_ID,Event_name,Normalization,Value_type)
VALUES('%s','%s','%s','%s')""%(table_type,typeoffile,typeoftissue,Pig_ID,Event_na
me,Normalization,valuetype))

cursor.execute("""UPDATE `monr_%s%s_%s` SET
Weight_of_tissue=%.4f,Normalization_factor=%.4f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='%s'""%(table_type,typeoffile,typeoftissue,Weight_of_tissue,Normalizat
ion_factor,Pig_ID,Event_name,Normalization,valuetype))
                                                                    for peak
in data3:
                                                                    try:

Integral=float(data3[peak])*Normalization_factor

Start_ppm=float(data4[peak])

End_ppm=float(data5[peak])

except:

Integral=''

Start_ppm=''

End_ppm=''

#raise
                                                                    if not
(Integral==' or Start_ppm==' or End_ppm==' ):

#cursor.execute("""UPDATE `monr_%s%s_%s` SET `s`=%.7f WHERE Pig_ID='%s' AND
Event_name='%s' AND
Normalization='%s'""%(table_type,typeoffile,typeoftissue,peak,peak_value_normaliz
ed,Pig_ID,Event_name,Normalization))

```

```

cursor.execute("""UPDATE `monr_%s%s_%s` SET `%s`=%f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='Integral'""%(table_type,typeoffile,typeoftissue,peak,Integral,Pig_ID,
Event_name,Normalization))

cursor.execute("""UPDATE `monr_%s%s_%s` SET `%s`=%f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='Start_ppm'""%(table_type,typeoffile,typeoftissue,peak,Start_ppm,Pig_I
D,Event_name,Normalization))

cursor.execute("""UPDATE `monr_%s%s_%s` SET `%s`=%f WHERE Pig_ID='%s' AND
Event_name='%s' AND Normalization='%s' AND
Value_type='End_ppm'""%(table_type,typeoffile,typeoftissue,peak,End_ppm,Pig_ID,Ev
ent_name,Normalization))

else:

pass

else:
pass

elif(form['typeoffile'].value=="Binning"):
set_character_set=""SET
NAMES 'utf8' COLLATE 'utf8_bin'""

cursor.execute(set_character_set)

Event_name=form['event'].value

message2=''
if Event_name=='Other':
Event_name=99
else:
pass

typeoftissue=(form['typeoftissue'].value).lower()

data1=open(File_storage_path+"/"+File_name,'r').readlines()
line1_split=[]
line2_split=[]
file_columns=[]
data3=[]
data4=dict()
datalines=0
for line in data1:
if datalines==0:

line1_split=line.split()
for a in
range(1,len(line1_split)):
file_columns.append(line1_split[a])
datalines=1
else:

line2_split=line.split()
for a in
range(1,len(line2_split)):

```

```

data3.append(line2_split[a])
range(len(file_columns)):
data4[file_columns[a]]=data3[a]
TABLE IF NOT EXISTS `monr_binning_%s` (
    varchar(6) NOT NULL,
    `Event_name` INT( 2 ) NOT NULL,
    KEY (`Pig_ID`,`Event_name`),
    `Pig_ID` (`Pig_ID`),
    `monr_binning_%s_ibfk_1` FOREIGN KEY (`Pig_ID`)
    `monr_identifier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin""%(typeoftissue,typeoftissue))
COLUMN_NAME FROM information_schema.columns WHERE
TABLE_NAME='monr_binning_%s'""%(typeoftissue))

(column=='Pig_ID') and not (column=='Event_name'):
db_columns.append(column)

new_column_count=len(db_columns)
file_columns:
db_columns:
new_column_count=new_column_count+1

new_column_count>253:
file_columns:
db_columns:
cursor.execute("""ALTER TABLE `monr_binning_%s` ADD COLUMN `%s` DOUBLE(10,9)
default NULL""%(typeoftissue,column))
for a in
cursor.execute("""CREATE
`Pig_ID`
PRIMARY
KEY
CONSTRAINT
REFERENCES
)
cursor.execute("""SELECT
output=cursor.fetchall()
db_columns=[]
for line in output:
    for column in line:
        if not
            else:
                pass
for column in
    if not column in
        else:
            pass
#if not
for column in
    if not column in
        else:
            pass
#for column in db_columns:

```

```

                                                                    #if not column in
file_columns:
#cursor.execute("""ALTER TABLE `monr_binning_%s` DROP COLUMN
`s` """"%(typeoftissue,column))
                                                                    #else:
                                                                    #pass
                                                                    if(Event_name=="0"):
Event_name=str(Event_name)+"_"+form["hours"].value+"_"+form["minutes"].value
                                                                    else:
                                                                    pass
                                                                    cursor.execute("""INSERT
IGNORE INTO `monr_binning_%s` (Pig_ID,Event_name)
VALUES('%s','%s')""""%(typeoftissue,Pig_ID,Event_name))
                                                                    for column in data4:
                                                                    try:
cursor.execute("""UPDATE `monr_binning_%s` SET `s`=%0.9f WHERE Pig_ID='%s' AND
Event_name='%s'""""%(typeoftissue,column,float(data4[column]),Pig_ID,Event_name))
                                                                    except:
                                                                    pass
                                                                    #else:
                                                                    #message2='. It looks
like you are trying to upload a binning file with different set of definitions
than the previous ones in the database. Please contact the data manager in order
to make the changes. The data has not been uploaded'
                                                                    connection.commit()
                                                                    cursor.close()
                                                                    connection.close()
                                                                    print """"Set-Cookie:
validate_filename=; Path=/""""
                                                                    print """"Set-Cookie:
session_Pig_ID="""+Pig_ID+"""; Path=/""""
                                                                    print """"Set-Cookie:
fileupload_filetype="""+filetype+"""; Path=/""""
                                                                    print 'Content-Type:
text/html\n\n'
                                                                    message = 'The file \'' +
File_name + '\" was uploaded successfully'
                                                                    filehandle =
urllib.urlopen("fileupload.html")
                                                                    for lines in
filehandle.readlines( ):
                                                                    s = lines
                                                                    n =
                                                                    n =
                                                                    n =
                                                                    n = re.sub('<td
class="message" id="message1">', '<td class="message"
id="message1">'+message+message2,n)
                                                                    print n
                                                                    filehandle.close()
                                                                    else:#Teg data upload
                                                                    set_character_set=""""SET NAMES
'utf8' COLLATE 'utf8_bin'""""

```

```

store_buffer=uploaded_file.file.readlines()

File_type = "Text"
#File_size = len(store_buffer)
File_storage_path =
"/project/beilmanlab.umn.edu-upload/"+studynome+"/Files/"+Pig_ID+"/"+Data_type
connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=lhb)
cursor = connection.cursor()

cursor.execute(set_character_set)

#cursor.execute("""INSERT INTO
monr_fileupload(File_name,File_storage_path,Upload_time,Data_type,File_type,File_s
ize,Pig_ID)
VALUES('%s','%s',CURRENT_TIMESTAMP,'%s','%s',%d,'%s')""%(File_name,File_storage_p
ath,Data_type,File_type,File_size,Pig_ID))

try:
    #Make sure the path exists
    if not
        os.path.exists(File_storage_path):
            os.umask(0)

        else:
            pass
    except:
        pass
    dest_file =
        number=0
    for line in store_buffer:
        if number==0:
            #print
            ding=line.split('\t')
            temp=[]

            temp.append((ding[1]).replace(' ',''))
            temp.append((ding[8]).replace(' ',''))
            temp.append((ding[16]).replace(' ',''))
            temp.append((ding[17]).replace(' ',''))
            temp.append((ding[18]).replace(' ',''))
            temp.append((ding[19]).replace(' ',''))

            dest_file.write('
            number=1
        else:
            pass

        if len(line)>1 and
Pig_ID_pattern=re.compile(r'%s'%Pig_ID)
re.search(r'MONR',line.split('\t')[1]) and
re.search(Pig_ID_pattern,line.split('\t')[1]):

```

```

len(line.split('\t'))

temp.append((ding[1]).replace('MONR', ''))
temp.append((ding[8]).replace(' ', ''))
temp.append((ding[16]).replace(' ', ''))
temp.append((ding[17]).replace(' ', ''))
temp.append((ding[18]).replace(' ', ''))
temp.append((ding[19]).replace(' ', ''))

'.join(temp)+'\n')

dest_file.write('

else:
    pass
dest_file.close()
os.umask(0)

os.chmod(File_storage_path+"/"+File_name,0644)

sourceText1=['Baseline', 'Shock45', 'LR1', 'FR20', 'FR3', 'FR4', 'FR5', 'FR6', 'FR7', 'FR8',
'FR9', 'FR10', 'FR11', 'FR12', 'FR13', 'FR14', 'FR15', 'FR16', 'FR17', 'FR18', 'FR19', 'FR1',
'FR2', 'PR24', 'PR48']

replaceText1=['1', '2', '3', '23', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16',
'17', '18', '19', '20', '21', '22', '4', '5', '24', '25']

replacemachine(File_storage_path+"/"+File_name, sourceText1, replaceText1)
File_size =
len(open(File_storage_path+"/"+File_name, 'r').read())
cursor.execute("""INSERT
IGNORE INTO monr_fileupload(Pig_ID,Data_type)
VALUES('%s', '%s')""%(Pig_ID,Data_type))

cursor.execute("""UPDATE
monr_fileupload SET
File_name='%s',File_storage_path='%s',Upload_time=CURRENT_TIMESTAMP,File_type='%s',
File_size=%d WHERE Pig_ID='%s' AND
Data_type='%s'""%(File_name,File_storage_path,File_type,File_size,Pig_ID,Data_typ
e))

cbc=open(File_storage_path+"/"+File_name, 'r').readlines()
for line in cbc:
    if number==1:
        number=2
    elif number==2:
        #print line
        #if

(len(line.split())>0):
#Pig_ID=line.split()[0]

#else:
#Pig_ID='

```

```

(len(line.split())>1):
Event_name=line.split()[1]

(len(line.split())>2):
R_min=line.split()[2]

(len(line.split())>3):
K_min=line.split()[3]

(len(line.split())>4):
Angle_deg=line.split()[4]

(len(line.split())>5):
MA_mm=line.split()[5]

not Event_name==' and not R_min==' :
R_min=float(line.split()[2])

cursor.execute("""INSERT IGNORE INTO monr_teg_data (Pig_ID,Event_name)
VALUES('%s','%s')"""%(Pig_ID,Event_name))

cursor.execute("""INSERT IGNORE INTO monr_teg_data (Pig_ID,Event_name)
VALUES('%s','%s')""%(Pig_ID,Event_name))

cursor.execute("""UPDATE monr_teg_data SET R_min=%.1f WHERE Pig_ID='%s' AND
Event_name='%s'""%(R_min,Pig_ID,Event_name))

cursor.execute("""UPDATE monr_teg_data SET R_min=%.1f WHERE Pig_ID='%s' AND
Event_name='%s'""%(R_min,Pig_ID,Event_name))

not Event_name==' and not K_min==' :
K_min=float(line.split()[3])

cursor.execute("""INSERT IGNORE INTO monr_teg_data (Pig_ID,Event_name)
VALUES('%s','%s')""%(Pig_ID,Event_name))

cursor.execute("""INSERT IGNORE INTO monr_teg_data (Pig_ID,Event_name)
VALUES('%s','%s')""%(Pig_ID,Event_name))

```

```

if
else:
    Event_name=''
if
else:
    R_min=''
if
else:
    K_min=''
if
else:
    Angle_deg=''
if
else:
    MA_mm=''
if not Pig_ID==' and
else:
    pass
if not Pig_ID==' and

```



```

cursor.execute("""UPDATE monr_teg_data SET K_min=%.1f WHERE Pig_ID='%s' AND
Event_name='%s'"""%(K_min,Pig_ID,Event_name))

cursor.execute("""UPDATE monr_teg_data SET K_min=%.1f WHERE Pig_ID='%s' AND
Event_name='%s'"""%(K_min,Pig_ID,Event_name))

else:
    pass
if not Pig_ID==' and

not Event_name==' and not Angle_deg=='':

Angle_deg=float(line.split()[4])

cursor.execute("""INSERT IGNORE INTO monr_teg_data (Pig_ID,Event_name)
VALUES('%s','%s')"""%(Pig_ID,Event_name))

cursor.execute("""INSERT IGNORE INTO monr_teg_data (Pig_ID,Event_name)
VALUES('%s','%s')"""%(Pig_ID,Event_name))

cursor.execute("""UPDATE monr_teg_data SET Angle_deg=%.1f WHERE Pig_ID='%s' AND
Event_name='%s'"""%(Angle_deg,Pig_ID,Event_name))

cursor.execute("""UPDATE monr_teg_data SET Angle_deg=%.1f WHERE Pig_ID='%s' AND
Event_name='%s'"""%(Angle_deg,Pig_ID,Event_name))

else:
    pass
if not Pig_ID==' and

not Event_name==' and not MA_mm=='':

MA_mm=float(line.split()[5])

cursor.execute("""INSERT IGNORE INTO monr_teg_data (Pig_ID,Event_name)
VALUES('%s','%s')"""%(Pig_ID,Event_name))

cursor.execute("""INSERT IGNORE INTO monr_teg_data (Pig_ID,Event_name)
VALUES('%s','%s')"""%(Pig_ID,Event_name))

cursor.execute("""UPDATE monr_teg_data SET MA_mm=%.1f WHERE Pig_ID='%s' AND
Event_name='%s'"""%(MA_mm,Pig_ID,Event_name))

cursor.execute("""UPDATE monr_teg_data SET MA_mm=%.1f WHERE Pig_ID='%s' AND
Event_name='%s'"""%(MA_mm,Pig_ID,Event_name))

else:
    pass
connection.commit()
cursor.close()
connection.close()
print ""Set-Cookie:

print ""Set-Cookie:

print ""Set-Cookie:

print 'Content-Type:

message = 'The file \'' +

validate_filename=; Path="/""
session_Pig_ID=""+Pig_ID+""; Path="/""
fileupload_filetype=""+filetype+""; Path="/""
text/html\n\n'
File_name + '\" was uploaded successfully'

```

```

urllib.urlopen("fileupload.html")
filehandle.readlines( ):
re.sub("#Common_Data#",commondata,s)
re.sub("#greetings#", "You are logged in as "+User_id,n)
class="message" id="message1">', '<td class="message" id="message1">'+message,n)
print n
filehandle.close()
else:
message = 'Please select a file to
upload'
Pig_ID = form['pigids'].value
filehandle =
urllib.urlopen("fileupload.html")
validate_filename=; Path="/"
session_Pig_ID=""+Pig_ID+""; Path="/"
fileupload_filetype=""+filetype+""; Path="/"
text/html\n\n'
):
re.sub("#Common_Data#",commondata,s)
are logged in as "+User_id,n)
n = re.sub("#greetings#", "You
n = re.sub('<td
class="message" id="message1">', '<td class="message" id="message1">'+message,n)
print n
filehandle.close()
else:
message = 'Javascript should be
enabled to upload files.'
filehandle =
urllib.urlopen("fileupload.html")
validate_filename=; Path="/"
session_Pig_ID=""+Pig_ID+""; Path="/"
fileupload_filetype=""+filetype+""; Path="/"
print ""Set-Cookie:
print ""Set-Cookie:
print ""Set-Cookie:
print 'Content-Type:
for lines in filehandle.readlines( ):
s = lines
n =
n = re.sub("#greetings#", "You are
n = re.sub('<td class="message"
id="message1">', '<td class="message" id="message1">'+message,n)

```

```

        print n
        filehandle.close()
    except:
        message="Either File upload or File data
upload did not complete. Invalid file/file name or file already exists in the
database. It could also be a programming error. Please contact the data manager."
        #try:
            #connection.rollback()
        #except:
            #pass
            ##raise
        filehandle =
urllib.urlopen("fileupload.html")
        print ""Set-Cookie: validate_filename=;
Path="/"
        print ""Set-Cookie:
session_Pig_ID=""+Pig_ID+""; Path="/"
        print ""Set-Cookie:
fileupload_filetype=""+filetype+""; Path="/"
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n =
re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "You are
logged in as "+User_id,n)
            n = re.sub('<td class="message"
id="message1">', '<td class="message" id="message1">'+message,n)
            print n
            filehandle.close()
            #raise
    else:
        filehandle = urllib.urlopen(page)
        print ""Set-Cookie: validate_filename=;
Path="/"
        print ""Set-Cookie:
session_Pig_ID=""+Pig_ID+""; Path="/"
        print ""Set-Cookie:
fileupload_filetype=""+filetype+""; Path="/"
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "You are logged
in as "+User_id,n)
            print n
            filehandle.close()
    else:
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "You are logged in as
"+User_id,n)
            print n
            filehandle.close()
except:

```

```

        page="home.html"
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "You are logged in as
"+User_id,n)

            print n
            filehandle.close()
            #raise
    else:
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "Please login to continue",n)
            print n
            filehandle.close()
    else:
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "Please login to continue",n)
            print n
            filehandle.close()
    except:
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "Please login to continue",n)
            print n
            filehandle.close()
            #raise
    else:
        filehandle = urllib.urlopen(page)
        print 'Content-Type: text/html\n\n'
        for lines in filehandle.readlines( ):
            s = lines
            n = re.sub("#Common_Data#",commondata,s)
            n = re.sub("#greetings#", "Please login to continue",n)
            print n
            filehandle.close()

```

Code written for update_excel_files.py

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sctl_update_excel_files import *

```

Code written for imported sctl_update_excel_files module

```

#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql
import cgi
#import cgi
#cgi.enable()
import urllib
import string
import shutil
import operator
import time
import datetime
import random
import os
import bz2
import urllib2
import cookielib
import Cookie
from urllib2 import urlopen, Request
import httplib
from xlwt import *
import xlwt as pyxel
import raj_excel_export
#####
#####
#####
alphabet =
'abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz'
min = 24
max = 32
buscopan=''
for x in random.sample(alphabet,random.randint(min,max)):
    buscopan+=x
theKey = buscopan
#####
#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
dbconnection.close()
create_paths=[]
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
User_id=''
#Pig_ID=''
action=''
errors_found=0
files_not_found=0
Updated_by=''
#####
#####
#####
def connect_to_db_and_execute(the_query):

```

```

connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
cursor = connection.cursor()
cursor.execute("""%s""""the_query)
connection.commit()
cursor.close()
connection.close()
#####
#####
#####
def
update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
):
    global errors_found,Updated_by
    try:
        if(os.path.exists(File_path)):
            File_contents=open(File_path,'rb').read()
            File_size=len(File_contents)
            insert_info="""INSERT IGNORE INTO monr_allsubjects_export(Data_type)
VALUES('%s')""""(Data_type)
            connect_to_db_and_execute(insert_info)
            insert_info="""UPDATE monr_allsubjects_export SET
File_name='%s',File_storage_path='%s',File_size=%d,Last_update_time=CURRENT_TIMEST
AMP,Updated_by='%s' WHERE
Data_type='%s'""""(File_name,File_storage_path,File_size,Updated_by,Data_type)
            connect_to_db_and_execute(insert_info)
        else:
            files_not_found=files_not_found+1
    except:
        errors_found=errors_found+1
#####
#####
#####
def
update_individual_export_database(Data_type,File_name,File_storage_path,File_path)
:
    global errors_found,Pig_ID,Updated_by
    try:
        if(os.path.exists(File_path)):
            File_contents=open(File_path,'rb').read()
            File_size=len(File_contents)
            insert_info="""INSERT IGNORE INTO
monr_individual_export(Pig_ID,Data_type) VALUES('%s','%s')""""(Pig_ID,Data_type)
            connect_to_db_and_execute(insert_info)
            insert_info="""UPDATE monr_individual_export SET
File_name='%s',File_storage_path='%s',File_size=%d,Last_update_time=CURRENT_TIMEST
AMP,Updated_by='%s' WHERE Pig_ID='%s' and
Data_type='%s'""""(File_name,File_storage_path,File_size,Updated_by,Pig_ID,Data_ty
pe)
            connect_to_db_and_execute(insert_info)
        else:
            files_not_found=files_not_found+1
    except:
        errors_found=errors_found+1
#####Individual#####
#####
#####
print 'Content-Type: text/javascript\n'

```

```

if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", "") )
    try:
        bloatvalue = a_cookie["bloat"].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SELECT User_id,Group_name FROM sccl_authentication WHERE
Temp_key='%s'"""%(bloatvalue))
        output = cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        if (len(output)>0):
            for a,b in output:
                User_id = a
                User_group = b
                if(User_group=='Admin'):
                    form = cgi.FieldStorage()
                    if(form.has_key('which')):
                        action = form['which'].value
                        if (action == 'update_all_excel_files'):
                            #Pig_ID = form['Pig_ID'].value
                            raj_excel_export.lhost = lhost
                            raj_excel_export.lport = lport
                            raj_excel_export.luser = luser
                            raj_excel_export.lpasswd = lpasswd
                            raj_excel_export.ldb = ldb
                            raj_excel_export.studyname="MONR"

raj_excel_export.excel_base_dir="/project/beilmanlab.umn.edu-
upload/"+raj_excel_export.studyname+"/excel_files/"

raj_excel_export.excel_save_dir=raj_excel_export.excel_base_dir
Updated_by=User_id
create_paths.append(raj_excel_export.excel_save_dir)
try:
    #Make sure the path exists
    for the_dir in create_paths:
        if not os.path.exists(the_dir):
            os.umask(0)
            os.makedirs(the_dir,0771)
        else:
            pass
    except:
        pass

#####Overall#####
#####

raj_excel_export.excel_file_name="monr_identifier_table.xls"

raj_excel_export.tables_match={'`t1`: 'monr_identifier_table'}
raj_excel_export.the_query="""SELECT
`t1`.Pig_ID,`t1`.Start_date,`t1`.End_date,`t1`.Weight,`t1`.Pig_group,`t1`.Blood_re
moved,`t1`.LR_given,`t1`.Total_volume_given,`t1`.Died,`t1`.Time_of_death,`t1`.Caus

```

```
e_of_death,`t1`.Comments FROM `monr_identifier_table` AS `t1` ORDER BY
`t1`.Start_date"""
```

```
raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Study_Overview"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir
```

```
File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name
```

```
update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
```

```
#####Overall#####
#####
#####
```

```
#raj_excel_export.excel_file_name="monr_all_vitals_labs_gcs_only_events.xls"
```

```
#raj_excel_export.tables_match={'`t1`':'monr_vitals_manual_10_min`,`t2`':'monr_vitals_manual_one_hour`,`t3`':'monr_lab_values`,`t4`':'monr_gcs_scores`,`t5`':'monr_identifier_table'}
```

```
    #raj_excel_export.the_query=""SELECT
`t1`.Pig_ID,`t5`.Pig_group,`t1`.Date_of_experiment,`t1`.Time_of_experiment,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanArtPres,`t1`.PA_Sys,`t1`.PA_Dia,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t1`.StO2_15mm,`t1`.THI_15mm,`t1`.FiO2,`t1`.Propofol,`t1`.BIS,`t2`.Wedge_pressure,`t2`.Urine_output,`t1`.temperature,`t2`.Cardiac_output,`t2`.Arterial_pH,`t2`.Arterial_PCO2,`t2`.Arterial_PO2,`t2`.Arterial_O2_sat,`t2`.Venous_PO2,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.DO2,`t2`.VO2,`t2`.ERO2,`t2`.Lactate,`t2`.Base_deficit,`t2`.Bladder_pressure,`t2`.Chesttube_output,`t1`.Event_comments,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Potassium,`t2`.Calcium,`t2`.Glucose,`t3`.Bilirubin_total,`t3`.Albumin,`t3`.Protein_total,`t3`.Alk_phosphatase,`t3`.ALT,`t3`.AST,`t3`.Urea_nitrogen,`t3`.CK,`t3`.Creatinine,`t3`.LD,`t3`.Platelet_count,`t3`.Urine_creatinine,`t4`.GCS_score FROM
`monr_vitals_manual_10_min` AS `t1` LEFT JOIN `monr_vitals_manual_one_hour` AS
`t2` ON `t1`.Date_of_experiment=`t2`.Date_of_experiment AND
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_lab_values` AS
`t3` ON `t1`.Event_name=`t3`.Event_name AND `t1`.Pig_ID=`t3`.Pig_ID LEFT JOIN
`monr_gcs_scores` AS `t4` ON `t1`.Event_name=`t4`.Event_name AND
`t1`.Pig_ID=`t4`.Pig_ID LEFT JOIN `monr_identifier_table` AS `t5` ON
`t1`.Pig_ID=`t5`.Pig_ID WHERE `t1`.Event_name<>99 ORDER BY
Pig_ID,Date_of_experiment,Time_of_experiment"""
```

```
#raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    #Data_type="Vitals_Labs_GCS_events"
    #File_name=raj_excel_export.excel_file_name
    #File_storage_path=raj_excel_export.excel_save_dir
```

```
#File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name
```

```
#update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
h)
```

```
#####Overall#####
#####
#####
```



```

#raj_excel_export.excel_file_name="monr_all_vitals_labs_gcs.xls"

#raj_excel_export.tables_match={'`t1`':'monr_vitals_manual_10_min',`t2`':'monr_vitals_manual_one_hour',`t3`':'monr_lab_values',`t4`':'monr_gcs_scores',`t5`':'monr_idenfier_table'}

#raj_excel_export.the_query=""SELECT
`t1`.Pig_ID,`t5`.Pig_group,`t1`.Date_of_experiment,`t1`.Time_of_experiment,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanArtPres,`t1`.PA_Sys,`t1`.PA_Dia,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t1`.StO2_15mm,`t1`.THI_15mm,`t1`.FiO2,`t1`.Propofol,`t1`.BIS,`t2`.Wedge_pressure,`t2`.Urine_output,`t1`.temperature,`t2`.Cardiac_output,`t2`.Arterial_pH,`t2`.Arterial_PC02,`t2`.Arterial_PO2,`t2`.Arterial_O2_sat,`t2`.Venous_PO2,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.DO2,`t2`.VO2,`t2`.ER02,`t2`.Lactate,`t2`.Base_deficit,`t2`.Bladder_pressure,`t2`.Chesttube_output,`t1`.Event_comments,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Potassium,`t2`.Calcium,`t2`.Glucose,`t3`.Bilirubin_total,`t3`.Albumin,`t3`.Protein_total,`t3`.Alk_phosphatase,`t3`.ALT,`t3`.AST,`t3`.Urea_nitrogen,`t3`.CK,`t3`.Creatinine,`t3`.LD,`t3`.Platelet_count,`t3`.Urine_creatinine,`t4`.GCS_score FROM
`monr_vitals_manual_10_min` AS `t1` LEFT JOIN `monr_vitals_manual_one_hour` AS `t2` ON `t1`.Date_of_experiment=`t2`.Date_of_experiment AND
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_lab_values` AS `t3` ON `t1`.Event_name=`t3`.Event_name AND `t1`.Pig_ID=`t3`.Pig_ID LEFT JOIN `monr_gcs_scores` AS `t4` ON `t1`.Event_name=`t4`.Event_name AND
`t1`.Pig_ID=`t4`.Pig_ID LEFT JOIN `monr_idenfier_table` AS `t5` ON
`t1`.Pig_ID=`t5`.Pig_ID ORDER BY Pig_ID,Date_of_experiment,Time_of_experiment""

#raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
#Data_type="Vitals_Labs_GCS"
#File_name=raj_excel_export.excel_file_name
#File_storage_path=raj_excel_export.excel_save_dir

#File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

#update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)

#####Overall#####
#####
#####
raj_excel_export.excel_file_name="monr_vitals_events.xls"

raj_excel_export.tables_match={'`t1`':'monr_vitals_manual_10_min',`t2`':'monr_vitals_manual_one_hour',`t3`':'monr_idenfier_table'}
raj_excel_export.the_query=""SELECT
`t1`.Pig_ID,`t3`.Pig_group,`t1`.Date_of_experiment,`t1`.Time_of_experiment,`t1`.Event_comments,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanArtPres,`t1`.PA_Sys,`t1`.PA_Dia,`t2`.Wedge_pressure,`t2`.Urine_output,`t2`.Bladder_pressure,`t1`.temperature,`t2`.Cardiac_output,`t1`.Propofol,`t1`.BIS,`t1`.FiO2,`t1`.StO2_15mm,`t1`.THI_15mm,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t2`.Chesttube_output,`t2`.Arterial_pH,`t2`.Arterial_PC02,`t2`.Arterial_PO2,`t2`.Arterial_O2_sat,`t2`.Venous_PO2,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.DO2,`t2`.VO2,`t2`.ER02,`t2`.Lactate,`t2`.Base_deficit,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Potassium,`t2`.Calcium,`t2`.Glucose FROM `monr_vitals_manual_10_min` AS `t1` LEFT JOIN
`monr_vitals_manual_one_hour` AS `t2` ON
`t1`.Date_of_experiment=`t2`.Date_of_experiment AND
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_idenfier_table`

```

```
AS `t3` ON `t1`.Pig_ID=`t3`.Pig_ID WHERE `t1`.Event_name<>'99' ORDER BY
`t1`.Date_of_experiment,`t1`.Time_of_experiment"""
```

```
raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Vitals_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir
```

```
File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name
```

```
update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
```

```
#####Overall#####
#####
#####
```

```
raj_excel_export.excel_file_name="monr_vitals.xls"
```

```
raj_excel_export.tables_match={'`t1`':'monr_vitals_manual_10_min`,`t2`':'monr_vit
als_manual_one_hour`,`t3`':'monr_identifer_table'}
```

```
raj_excel_export.the_query="""SELECT
`t1`.Pig_ID,`t3`.Pig_group,`t1`.Date_of_experiment,`t1`.Time_of_experiment,`t1`.Ev
ent_comments,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanAr
tPres,`t1`.PA_Sys,`t1`.PA_Dia,`t2`.Wedge_pressure,`t2`.Urine_output,`t2`.Bladder_p
ressure,`t1`.temperature,`t2`.Cardiac_output,`t1`.Propofol,`t1`.BIS,`t1`.FiO2,`t1`
.StO2_15mm,`t1`.THI_15mm,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t2`.Chesttube_o
utput,`t2`.Arterial_pH,`t2`.Arterial_PC02,`t2`.Arterial_PO2,`t2`.Arterial_O2_sat,`
t2`.Venous_PO2,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.DO2,`t2`.VO2,`t2`.ER02,`t2`.Lactate
,`t2`.Base_deficit,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Potassium,`t2`.Calc
ium,`t2`.Glucose FROM `monr_vitals_manual_10_min` AS `t1` LEFT JOIN
`monr_vitals_manual_one_hour` AS `t2` ON
`t1`.Date_of_experiment=`t2`.Date_of_experiment AND
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_identifer_table`
AS `t3` ON `t1`.Pig_ID=`t3`.Pig_ID ORDER BY
`t1`.Date_of_experiment,`t1`.Time_of_experiment"""
```

```
raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Vitals"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir
```

```
File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name
```

```
update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
```

```
#####Overall#####
#####
#####
```

```
raj_excel_export.excel_file_name="monr_labs_events.xls"
```

```
raj_excel_export.tables_match={'`t1`':'monr_lab_values`,`t2`':'monr_identifer_ta
ble'}
```

```
raj_excel_export.the_query="""SELECT
`t1`.Pig_ID,`t2`.Pig_group,`t1`.Event_name,`t1`.Bilirubin_total,`t1`.Albumin,`t1`.
Protein_total,`t1`.Alk_phosphatase,`t1`.ALT,`t1`.AST,`t1`.Urea_nitrogen,`t1`.CK,`t
```

```
1`.Creatinine,`t1`.LD,`t1`.Platelet_count,`t1`.Urine_creatinine FROM
`monr_lab_values` AS `t1` LEFT JOIN `monr_identifer_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN
'Baseline' THEN 1 WHEN 'Shock45' THEN 2 WHEN '3' THEN 3 WHEN '5' THEN 4 WHEN '9'
THEN 5 WHEN '17' THEN 6 WHEN '21' THEN 7 WHEN '45' THEN 8 WHEN '69' THEN 9 END"""
```

```
raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Labs_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir
```

```
File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name
```

```
update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
```

```
#####Overall#####
#####
#####
```

```
raj_excel_export.excel_file_name="monr_gcs_events.xls"
```

```
raj_excel_export.tables_match={'`t1`: 'monr_gcs_scores', '`t2`: 'monr_identifer_ta
ble'}
```

```
raj_excel_export.the_query=""SELECT
`t1`.Pig_ID,`t2`.Pig_group,`t1`.Event_name,`t1`.GCS_score FROM `monr_gcs_scores`
AS `t1` LEFT JOIN `monr_identifer_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER
BY `t2`.Start_date,CASE Event_name WHEN 'Baseline' THEN 1 WHEN '21' THEN 2 WHEN
'45' THEN 3 WHEN '69' THEN 4 END"""
```

```
raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="GCS_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir
```

```
File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name
```

```
update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
```

```
#####Overall#####
#####
#####
```

```
raj_excel_export.excel_file_name="monr_vitals_labs_gcs_events.xls"
```

```
raj_excel_export.tables_match={'`t1`: 'monr_vitals_manual_10_min', '`t2`: 'monr_vit
als_manual_one_hour', '`t3`: 'monr_lab_values', '`t4`: 'monr_gcs_scores', '`t5`: 'mon
r_identifer_table'}
```

```
raj_excel_export.the_query=""SELECT
`t1`.Pig_ID,`t5`.Weight,`t5`.Pig_group,`t5`.Start_date,`t5`.Died,`t5`.End_date,`t5
`.Time_of_death,`t5`.Cause_of_death,`t5`.OR_fluids,`t5`.LR_given,`t5`.Blood_given,
`t5`.Blood_removed,`t5`.Total_volume_given,`t1`.Date_of_experiment,`t1`.Time_of_ex
periment,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanArtPre
s,`t1`.PA_Sys,`t1`.PA_Dia,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t1`.StO2_15mm,
`t1`.THI_15mm,`t1`.FiO2,`t1`.Propofol,`t1`.BIS,`t2`.Wedge_pressure,`t2`.Urine_outp
ut,`t1`.temperature,`t2`.Cardiac_output,`t2`.Arterial_pH,`t2`.Arterial_PC02,`t2`.A
```

```

arterial_P02,`t2`.Arterial_O2_sat,`t2`.Venous_P02,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.D
O2,`t2`.VO2,`t2`.ERO2,`t2`.Lactate,`t2`.Base_deficit,`t2`.Bladder_pressure,`t2`.Ch
esttube_output,`t1`.Event_comments,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Pot
assium,`t2`.Calcium,`t2`.Glucose,`t3`.Bilirubin_total,`t3`.Albumin,`t3`.Protein_to
tal,`t3`.Alk_phosphatase,`t3`.ALT,`t3`.AST,`t3`.Urea_nitrogen,`t3`.CK,`t3`.Creatin
ine,`t3`.LD,`t3`.Platelet_count,`t3`.Urine_creatinine,`t4`.GCS_score FROM
`monr_vitals_manual_10_min` AS `t1` LEFT JOIN `monr_vitals_manual_one_hour` AS
`t2` ON `t1`.Date_of_experiment=`t2`.Date_of_experiment AND
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_lab_values` AS
`t3` ON `t1`.Event_name=`t3`.Event_name AND `t1`.Pig_ID=`t3`.Pig_ID LEFT JOIN
`monr_gcs_scores` AS `t4` ON `t1`.Event_name=`t4`.Event_name AND
`t1`.Pig_ID=`t4`.Pig_ID LEFT JOIN `monr_identifiler_table` AS `t5` ON
`t1`.Pig_ID=`t5`.Pig_ID WHERE `t1`.Event_name<>99 ORDER BY
`t5`.Start_date,Date_of_experiment,Time_of_experiment""

```

```

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Vitals_labs_gcs_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

```

```
File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name
```

```
update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
```

```

#####Overall#####
#####
#####

```

```
raj_excel_export.excel_file_name="monr_vitals_labs_gcs.xls"
```

```

raj_excel_export.tables_match={'`t1`':'monr_vitals_manual_10_min`,`t2`':'monr_vit
als_manual_one_hour`,`t3`':'monr_lab_values`,`t4`':'monr_gcs_scores`,`t5`':'mon
r_identifiler_table'}

```

```

raj_excel_export.the_query=""SELECT
`t1`.Pig_ID,`t5`.Weight,`t5`.Pig_group,`t5`.Start_date,`t5`.Died,`t5`.End_date,`t5
`.Time_of_death,`t5`.Cause_of_death,`t5`.OR_fluids,`t5`.LR_given,`t5`.Blood_given,
`t5`.Blood_removed,`t5`.Total_volume_given,`t1`.Date_of_experiment,`t1`.Time_of_ex
periment,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanArtPre
s,`t1`.PA_Sys,`t1`.PA_Dia,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t1`.StO2_15mm,
`t1`.THI_15mm,`t1`.FiO2,`t1`.Propofol,`t1`.BIS,`t2`.Wedge_pressure,`t2`.Urine_outp
ut,`t1`.temperature,`t2`.Cardiac_output,`t2`.Arterial_pH,`t2`.Arterial_PC02,`t2`.A
rterial_P02,`t2`.Arterial_O2_sat,`t2`.Venous_P02,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.D
O2,`t2`.VO2,`t2`.ERO2,`t2`.Lactate,`t2`.Base_deficit,`t2`.Bladder_pressure,`t2`.Ch
esttube_output,`t1`.Event_comments,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Pot
assium,`t2`.Calcium,`t2`.Glucose,`t3`.Bilirubin_total,`t3`.Albumin,`t3`.Protein_to
tal,`t3`.Alk_phosphatase,`t3`.ALT,`t3`.AST,`t3`.Urea_nitrogen,`t3`.CK,`t3`.Creatin
ine,`t3`.LD,`t3`.Platelet_count,`t3`.Urine_creatinine,`t4`.GCS_score FROM
`monr_vitals_manual_10_min` AS `t1` LEFT JOIN `monr_vitals_manual_one_hour` AS
`t2` ON `t1`.Date_of_experiment=`t2`.Date_of_experiment AND
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_lab_values` AS
`t3` ON `t1`.Event_name=`t3`.Event_name AND `t1`.Pig_ID=`t3`.Pig_ID LEFT JOIN
`monr_gcs_scores` AS `t4` ON `t1`.Event_name=`t4`.Event_name AND
`t1`.Pig_ID=`t4`.Pig_ID LEFT JOIN `monr_identifiler_table` AS `t5` ON
`t1`.Pig_ID=`t5`.Pig_ID ORDER BY
`t5`.Start_date,Date_of_experiment,Time_of_experiment""

```

```

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Vitals_labs_gcs"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

#####Overall#####
#####
#####
    raj_excel_export.excel_file_name="monr_tegdata_events.xls"

raj_excel_export.tables_match={'`t1`':'monr_teg_data`,`t2`':'monr_identifier_table'}

    raj_excel_export.the_query="""SELECT
`t1`.Pig_ID,`t2`.Pig_group,`t1`.Event_name,`t1`.R_min,`t1`.K_min,`t1`.Angle_deg,`t
1`.MA_mm FROM `monr_teg_data` AS `t1` LEFT JOIN `monr_identifier_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN
'Baseline' THEN 1 WHEN 'Shock45' THEN 2 WHEN '3' THEN 3 WHEN '5' THEN 4 WHEN '9'
THEN 5 WHEN '17' THEN 6 WHEN '21' THEN 7 WHEN '45' THEN 8 WHEN '69' THEN 9 END"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Tegdata_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

#####Overall#####
#####
#####

raj_excel_export.excel_file_name="monr_vitals_labs_gcs_tegdata_events.xls"

raj_excel_export.tables_match={'`t1`':'monr_vitals_manual_10_min`,`t2`':'monr_vit
als_manual_one_hour`,`t3`':'monr_lab_values`,`t4`':'monr_gcs_scores`,`t5`':'mon
r_identifier_table`,`t6`':'monr_teg_data'}

    raj_excel_export.the_query="""SELECT
`t1`.Pig_ID,`t5`.Weight,`t5`.Pig_group,`t5`.Start_date,`t5`.Died,`t5`.End_date,`t5
`.Time_of_death,`t5`.Cause_of_death,`t5`.OR_fluids,`t5`.LR_given,`t5`.Blood_given,
`t5`.Blood_removed,`t5`.Total_volume_given,`t1`.Date_of_experiment,`t1`.Time_of_ex
periment,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanArtPre
s,`t1`.PA_Sys,`t1`.PA_Dia,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t1`.StO2_15mm,
`t1`.THI_15mm,`t1`.FiO2,`t1`.Propofol,`t1`.BIS,`t2`.Wedge_pressure,`t2`.Urine_outp
ut,`t1`.temperature,`t2`.Cardiac_output,`t2`.Arterial_pH,`t2`.Arterial_PC02,`t2`.A
rterial_PO2,`t2`.Arterial_O2_sat,`t2`.Venous_PO2,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.D
O2,`t2`.VO2,`t2`.ERO2,`t2`.Lactate,`t2`.Base_deficit,`t2`.Bladder_pressure,`t2`.Ch
esttube_output,`t1`.Event_comments,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Pot
assium,`t2`.Calcium,`t2`.Glucose,`t3`.Bilirubin_total,`t3`.Albumin,`t3`.Protein_to

```

```

tal,`t3`.Alk_phosphatase,`t3`.ALT,`t3`.AST,`t3`.Urea_nitrogen,`t3`.CK,`t3`.Creatin
ine,`t3`.LD,`t3`.Platelet_count,`t3`.Urine_creatinine,`t4`.GCS_score,`t6`.R_min,`t
6`.K_min,`t6`.Angle_deg,`t6`.MA_mm FROM `monr_vitals_manual_10_min` AS `t1` LEFT
JOIN `monr_vitals_manual_one_hour` AS `t2` ON
`t1`.Date_of_experiment=`t2`.Date_of_experiment AND
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_lab_values` AS
`t3` ON `t1`.Event_name=`t3`.Event_name AND `t1`.Pig_ID=`t3`.Pig_ID LEFT JOIN
`monr_gcs_scores` AS `t4` ON `t1`.Event_name=`t4`.Event_name AND
`t1`.Pig_ID=`t4`.Pig_ID LEFT JOIN `monr_teg_data` AS `t6` ON
`t1`.Event_name=`t6`.Event_name AND `t1`.Pig_ID=`t6`.Pig_ID LEFT JOIN
`monr_identifider_table` AS `t5` ON `t1`.Pig_ID=`t5`.Pig_ID WHERE
`t1`.Event_name<>99 ORDER BY
`t5`.Start_date,Date_of_experiment,Time_of_experiment""

```

```

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Vitals_labs_gcs_tegdata_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

```

```
File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name
```

```
update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
```

```

#####Overall#####
#####
#####

```

```
raj_excel_export.excel_file_name="monr_vitals_labs_gcs_tegdata.xls"
```

```

raj_excel_export.tables_match={'`t1`':'monr_vitals_manual_10_min',`t2`':'monr_vit
als_manual_one_hour',`t3`':'monr_lab_values',`t4`':'monr_gcs_scores',`t5`':'mon
r_identifider_table',`t6`':'monr_teg_data'}

```

```

raj_excel_export.the_query=""SELECT
`t1`.Pig_ID,`t5`.Weight,`t5`.Pig_group,`t5`.Start_date,`t5`.Died,`t5`.End_date,`t5
`.Time_of_death,`t5`.Cause_of_death,`t5`.OR_fluids,`t5`.LR_given,`t5`.Blood_given,
`t5`.Blood_removed,`t5`.Total_volume_given,`t1`.Date_of_experiment,`t1`.Time_of_ex
periment,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanArtPre
s,`t1`.PA_Sys,`t1`.PA_Dia,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t1`.StO2_15mm,
`t1`.THI_15mm,`t1`.FiO2,`t1`.Propofol,`t1`.BIS,`t2`.Wedge_pressure,`t2`.Urine_outp
ut,`t1`.temperature,`t2`.Cardiac_output,`t2`.Arterial_pH,`t2`.Arterial_PCO2,`t2`.A
rterial_PO2,`t2`.Arterial_O2_sat,`t2`.Venous_PO2,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.D
O2,`t2`.VO2,`t2`.ERO2,`t2`.Lactate,`t2`.Base_deficit,`t2`.Bladder_pressure,`t2`.Ch
esttube_output,`t1`.Event_comments,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Pot
assium,`t2`.Calcium,`t2`.Glucose,`t3`.Bilirubin_total,`t3`.Albumin,`t3`.Protein_to
tal,`t3`.Alk_phosphatase,`t3`.ALT,`t3`.AST,`t3`.Urea_nitrogen,`t3`.CK,`t3`.Creatin
ine,`t3`.LD,`t3`.Platelet_count,`t3`.Urine_creatinine,`t4`.GCS_score,`t6`.R_min,`t
6`.K_min,`t6`.Angle_deg,`t6`.MA_mm FROM `monr_vitals_manual_10_min` AS `t1` LEFT
JOIN `monr_vitals_manual_one_hour` AS `t2` ON
`t1`.Date_of_experiment=`t2`.Date_of_experiment AND
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_lab_values` AS
`t3` ON `t1`.Event_name=`t3`.Event_name AND `t1`.Pig_ID=`t3`.Pig_ID LEFT JOIN
`monr_gcs_scores` AS `t4` ON `t1`.Event_name=`t4`.Event_name AND
`t1`.Pig_ID=`t4`.Pig_ID LEFT JOIN `monr_teg_data` AS `t6` ON
`t1`.Event_name=`t6`.Event_name AND `t1`.Pig_ID=`t6`.Pig_ID LEFT JOIN
`monr_identifider_table` AS `t5` ON `t1`.Pig_ID=`t5`.Pig_ID ORDER BY
`t5`.Start_date,Date_of_experiment,Time_of_experiment""

```

```

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Vitals_labs_gcs_tegdata"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

#####Overall#####
#####

    try:

raj_excel_export.excel_file_name="monr_binning_muscle_events.xls"

raj_excel_export.tables_match={'`t1`':'monr_binning_muscle`,`t2`':'monr_identifie
r_table'}

    raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Pig_group FROM `monr_binning_muscle` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="binning_muscle_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

    except:
        errors_found=errors_found+1

#####Overall#####
#####

    try:

raj_excel_export.excel_file_name="monr_binning_serum_events.xls"

raj_excel_export.tables_match={'`t1`':'monr_binning_serum`,`t2`':'monr_identifier
_table'}

    raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Pig_group FROM `monr_binning_serum` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="binning_serum_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

```

```

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
    except:
        errors_found=errors_found+1

#####Overall#####
#####

    try:

raj_excel_export.excel_file_name="monr_binning_liver_events.xls"

raj_excel_export.tables_match={'`t1`: 'monr_binning_liver', '`t2`: 'monr_identifie
r_table'}

        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Pig_group FROM `monr_binning_liver` AS `t1` LEFT JOIN
`monr_identifie
r_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
        Data_type="binning_liver_events"
        File_name=raj_excel_export.excel_file_name
        File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
    except:
        errors_found=errors_found+1

#####Overall#####
#####

    try:

raj_excel_export.excel_file_name="monr_binning_urine_events.xls"

raj_excel_export.tables_match={'`t1`: 'monr_binning_urine', '`t2`: 'monr_identifie
r_table'}

        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Pig_group FROM `monr_binning_urine` AS `t1` LEFT JOIN
`monr_identifie
r_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
        Data_type="binning_urine_events"
        File_name=raj_excel_export.excel_file_name
        File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

```



```

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
    except:
        errors_found=errors_found+1

#####Overall#####
#####
#####
    try:

raj_excel_export.excel_file_name="monr_hnmr_muscle_rawdata.xls"

raj_excel_export.tables_match={'`t1`: 'monr_hnmr_muscle', '`t2`: 'monr_identifia
ble'}

        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_muscle` AS `t1` LEFT JOIN
`monr_identifiaer_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='No' ORDER BY `t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
        Data_type="hnmr_muscle_rawdata"
        File_name=raj_excel_export.excel_file_name
        File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
    except:
        errors_found=errors_found+1
        #raise

#####Overall#####
#####
#####
    try:

raj_excel_export.excel_file_name="monr_hnmr_muscle_wettissue.xls"

raj_excel_export.tables_match={'`t1`: 'monr_hnmr_muscle', '`t2`: 'monr_identifia
ble'}

        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_muscle` AS `t1` LEFT JOIN
`monr_identifiaer_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Wet_tissue' ORDER BY `t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
        Data_type="hnmr_muscle_wettissue"
        File_name=raj_excel_export.excel_file_name
        File_storage_path=raj_excel_export.excel_save_dir

```

```

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
    except:
        errors_found=errors_found+1
        #raise

#####Overall#####
#####

    try:

raj_excel_export.excel_file_name="monr_hnmr_muscle_lyophilized.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_muscle`,`t2`':'monr_identifiaer_t
able'}

        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_muscle` AS `t1` LEFT JOIN
`monr_identifiaer_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Lyophilized_tissue' ORDER BY
`t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
        Data_type="hnmr_muscle_lyophilized"
        File_name=raj_excel_export.excel_file_name
        File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
    except:
        errors_found=errors_found+1
        #raise

#####Overall#####
#####

    try:

raj_excel_export.excel_file_name="monr_hnmr_muscle_alltissue.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_muscle`,`t2`':'monr_identifiaer_t
able'}

        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_muscle` AS `t1` LEFT JOIN
`monr_identifiaer_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)

```

```

Data_type="hnmr_muscle_alltissue"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

except:
    errors_found=errors_found+1
    #raise

#####Overall#####
#####

try:

raj_excel_export.excel_file_name="monr_hnmr_liver_rawdata.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_liver`,`t2`':'monr_identifiaer_ta
ble`}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_liver` AS `t1` LEFT JOIN
`monr_identifiaer_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='No' ORDER BY `t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="hnmr_liver_rawdata"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

except:
    errors_found=errors_found+1

#####Overall#####
#####

try:

raj_excel_export.excel_file_name="monr_hnmr_liver_wettissue.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_liver`,`t2`':'monr_identifiaer_ta
ble`}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_liver` AS `t1` LEFT JOIN
`monr_identifiaer_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Wet_tissue' ORDER BY `t2`.Start_date,`t1`.Event_name"""

```

```

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="hnmr_liver_wettissue"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
    except:
        errors_found=errors_found+1

#####Overall#####
#####

    try:

raj_excel_export.excel_file_name="monr_hnmr_liver_lyophilized.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_liver`,`t2`':'monr_identifier_ta
ble`}
        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_liver` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Lyophilized_tissue' ORDER BY
`t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="hnmr_liver_lyophilized"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)
    except:
        errors_found=errors_found+1

#####Overall#####
#####

    try:

raj_excel_export.excel_file_name="monr_hnmr_liver_alltissue.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_liver`,`t2`':'monr_identifier_ta
ble`}
        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_liver` AS `t1` LEFT JOIN

```

```

`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="hnmr_liver_alltissue"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

    except:
        errors_found=errors_found+1

#####Overall#####
#####
#####

    try:

raj_excel_export.excel_file_name="monr_pnmr_muscle_rawdata.xls"

raj_excel_export.tables_match={'`t1`':'monr_pnmr_muscle`,`t2`':'monr_identifier_t
able'}

    raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_pnmr_muscle` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='No' ORDER BY
`t2`.Start_date,`t1`.Event_name,`t1`.Value_type""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="pnmr_muscle_rawdata"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

    except:
        errors_found=errors_found+1

#####Overall#####
#####
#####

    try:

raj_excel_export.excel_file_name="monr_pnmr_muscle_wettissue.xls"

raj_excel_export.tables_match={'`t1`':'monr_pnmr_muscle`,`t2`':'monr_identifier_t
able'}

    raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.

```

```

Blood_removed,`t2`.Total_volume_given FROM `monr_pnmr_muscle` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Wet_tissue' ORDER BY
`t2`.Start_date,`t1`.Event_name,`t1`.Value_type""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="pnmr_muscle_wettissue"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

    except:
        errors_found=errors_found+1

#####Overall#####
#####
#####
    try:

raj_excel_export.excel_file_name="monr_pnmr_muscle_lyophilized.xls"

raj_excel_export.tables_match={'`t1`:':`monr_pnmr_muscle`,`t2`:':`monr_identifier_t
able`}

    raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_pnmr_muscle` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Lyophilized_tissue' ORDER BY
`t2`.Start_date,`t1`.Event_name,`t1`.Value_type""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="pnmr_muscle_lyophilized"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

    except:
        errors_found=errors_found+1

#####Overall#####
#####
#####
    try:

raj_excel_export.excel_file_name="monr_pnmr_muscle_alltissue.xls"

raj_excel_export.tables_match={'`t1`:':`monr_pnmr_muscle`,`t2`:':`monr_identifier_t
able`}

```

```

        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_pnmr_muscle` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name,`t1`.Normalization,`t1`.Value_type"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
        Data_type="pnmr_muscle_alltissue"
        File_name=raj_excel_export.excel_file_name
        File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

        except:
            errors_found=errors_found+1

#####Overall#####
#####
#####
        try:

raj_excel_export.excel_file_name="monr_pnmr_liver_rawdata.xls"

raj_excel_export.tables_match={'`t1`':'monr_pnmr_liver`,`t2`':'monr_identifier_ta
ble`}

        raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Tim
e_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.
Blood_removed,`t2`.Total_volume_given FROM `monr_pnmr_liver` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='No' ORDER BY
`t2`.Start_date,`t1`.Event_name,`t1`.Value_type"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
        Data_type="pnmr_liver_rawdata"
        File_name=raj_excel_export.excel_file_name
        File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
)

        except:
            errors_found=errors_found+1

#####Overall#####
#####
#####
        try:

raj_excel_export.excel_file_name="monr_pnmr_liver_wettissue.xls"

```

```

raj_excel_export.tables_match={'t1':'monr_pnmr_liver','t2':'monr_identifier_table'}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM `monr_pnmr_liver` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Wet_tissue' ORDER BY
`t2`.Start_date,`t1`.Event_name,`t1`.Value_type"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="pnmr_liver_wettissue"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)
except:
errors_found=errors_found+1

#####Overall#####
#####

try:

raj_excel_export.excel_file_name="monr_pnmr_liver_lyophilized.xls"

raj_excel_export.tables_match={'t1':'monr_pnmr_liver','t2':'monr_identifier_table'}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM `monr_pnmr_liver` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Lyophilized_tissue' ORDER BY
`t2`.Start_date,`t1`.Event_name,`t1`.Value_type"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="pnmr_liver_lyophilized"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)
except:
errors_found=errors_found+1

#####Overall#####
#####

try:

```



```

raj_excel_export.excel_file_name="monr_pnmr_liver_alltissue.xls"

raj_excel_export.tables_match={'t1':'monr_pnmr_liver','t2':'monr_identifiaer_table'}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM `monr_pnmr_liver` AS `t1` LEFT JOIN
`monr_identifiaer_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name,`t1`.Normalization,`t1`.Value_type"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="pnmr_liver_alltissue"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)

except:
errors_found=errors_found+1

#####Overall#####
#####

try:

raj_excel_export.excel_file_name="monr_hnmr_urine_rawdata.xls"

raj_excel_export.tables_match={'t1':'monr_hnmr_urine','t2':'monr_identifiaer_table'}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_urine` AS `t1` LEFT JOIN
`monr_identifiaer_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='No' ORDER BY `t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="hnmr_urine_rawdata"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)

except:
errors_found=errors_found+1

#####Overall#####
#####

try:

```

```

raj_excel_export.excel_file_name="monr_hnmr_urine_normalized.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_urine','`t2`':'monr_identifier_table'}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_urine` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Yes' ORDER BY `t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="hnmr_urine_normalized"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)

except:
errors_found=errors_found+1

#####Overall#####
#####

try:

raj_excel_export.excel_file_name="monr_hnmr_urine_alltissue.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_urine','`t2`':'monr_identifier_table'}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_urine` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="hnmr_urine_alltissue"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)

except:
errors_found=errors_found+1

#####Overall#####
#####

try:

```

```

raj_excel_export.excel_file_name="monr_hnmr_serum_rawdata.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_serum`,`t2`':'monr_identifier_table'}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_serum` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='No' ORDER BY `t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="hnmr_serum_rawdata"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)

except:
errors_found=errors_found+1

#####Overall#####
#####

try:

raj_excel_export.excel_file_name="monr_hnmr_serum_normalized.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_serum`,`t2`':'monr_identifier_table'}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_serum` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Normalization='Yes' ORDER BY `t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="hnmr_serum_normalized"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)

except:
errors_found=errors_found+1

#####Overall#####
#####

try:

```

```

raj_excel_export.excel_file_name="monr_hnmr_serum_alltissue.xls"

raj_excel_export.tables_match={'`t1`':'monr_hnmr_serum`,`t2`':'monr_identifier_table'}

raj_excel_export.the_query="""SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_date,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM `monr_hnmr_serum` AS `t1` LEFT JOIN
`monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID ORDER BY
`t2`.Start_date,`t1`.Event_name"""

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
Data_type="hnmr_serum_alltissue"
File_name=raj_excel_export.excel_file_name
File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path)

except:
errors_found=errors_found+1

#####
#####
#####

print
"""document.getElementById('message1').innerHTML='Excel files successfully updated. Files not found = %d, and Errors found = %d.<br />If any of these numbers are not zero, inform data manager';loadAllExcelFiles();"""%(files_not_found,errors_found)
else:
print """document.getElementById('message1').innerHTML='An error occurred. Please contact data manager. Files not found = %d, and Errors found = %d';"""%(files_not_found,errors_found)
else:
print """document.getElementById('message1').innerHTML='An error occurred. Please contact data manager. Files not found = %d, and Errors found = %d';"""%(files_not_found,errors_found)
else:
print """document.getElementById('message1').innerHTML='Sorry, only an Administrator or a Lab User can update excel data.';"""
else:
print """document.getElementById('message1').innerHTML='An error occurred. Please contact data manager. Error no:1';"""
except:
print """document.getElementById('message1').innerHTML='An error occurred. Please contact data manager. Error no:2';"""
#raise
else:
print """document.getElementById('message1').innerHTML='An error occurred. Please contact data manager. Error no:3';"""

```

Code written for update_excel_files_individual.py

```
#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
from sccl_update_excel_files_individual import *
```

Code written for imported sccl_update_excel_files_individual.py

```
#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import MySQLdb as sql
import cgi
#import cgi
#cgi.enable()
import urllib
import string
import shutil
import operator
import time
import datetime
import random
import os
import bz2
import urllib2
import cookielib
import Cookie
from urllib2 import urlopen, Request
import httplib
from xlwt import *
import xlwt as pycel
import raj_excel_export
#####
#####
#####
alphabet =
'abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz'
uvwxyzabcdefghijklmnopqrstuvwxyz'
min = 24
max = 32
buscopan=''
for x in random.sample(alphabet,random.randint(min,max)):
    buscopan+=x
theKey = buscopan
#####
#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
dbconnection.close()
create_paths=[]
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
User_id=''
Pig_ID=''
action=''
```

```

errors_found=0
files_not_found=0
Updated_by= ''
#####
#####
#####
def connect_to_db_and_execute(the_query):
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
    cursor = connection.cursor()
    cursor.execute("""%s"""%the_query)
    connection.commit()
    cursor.close()
    connection.close()
#####
#####
#####
def
update_allsubjects_export_database(Data_type,File_name,File_storage_path,File_path
):
    global errors_found,Updated_by
    try:
        if(os.path.exists(File_path)):
            File_contents=open(File_path,'rb').read()
            File_size=len(File_contents)
            insert_info="""INSERT IGNORE INTO monr_allsubjects_export(Data_type)
VALUES('%s')"""%(Data_type)
            connect_to_db_and_execute(insert_info)
            insert_info="""UPDATE monr_allsubjects_export SET
File_name='%s',File_storage_path='%s',File_size=%d,Last_update_time=CURRENT_TIMEST
AMP,Updated_by='%s' WHERE
Data_type='%s'"""%(File_name,File_storage_path,File_size,Updated_by,Data_type)
            connect_to_db_and_execute(insert_info)
        else:
            files_not_found=files_not_found+1
    except:
        errors_found=errors_found+1
#####
#####
#####
def
update_individual_export_database(Data_type,File_name,File_storage_path,File_path)
:
    global errors_found,Pig_ID,Updated_by
    try:
        if(os.path.exists(File_path)):
            File_contents=open(File_path,'rb').read()
            File_size=len(File_contents)
            insert_info="""INSERT IGNORE INTO
monr_individual_export(Pig_ID,Data_type) VALUES('%s','%s')"""%(Pig_ID,Data_type)
            connect_to_db_and_execute(insert_info)
            insert_info="""UPDATE monr_individual_export SET
File_name='%s',File_storage_path='%s',File_size=%d,Last_update_time=CURRENT_TIMEST
AMP,Updated_by='%s' WHERE Pig_ID='%s' and
Data_type='%s'"""%(File_name,File_storage_path,File_size,Updated_by,Pig_ID,Data_ty
pe)
            connect_to_db_and_execute(insert_info)
        else:

```

```

        files_not_found=files_not_found+1
    except:
        errors_found=errors_found+1
#####Individual#####
#####
#####
print 'Content-Type: text/javascript\n'
if (os.environ.has_key('HTTP_COOKIE')):
    a_cookie = Cookie.SimpleCookie( os.environ.get("HTTP_COOKIE", "") )
    try:
        bloatvalue = a_cookie["bloat"].value
        connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=ldb)
        cursor = connection.cursor()
        cursor.execute("""SELECT User_id,Group_name FROM sccl_authentication WHERE
Temp_key='%s'""%(bloatvalue))
        output = cursor.fetchall()
        connection.commit()
        cursor.close()
        connection.close()
        if (len(output)>0):
            for a,b in output:
                User_id = a
                User_group = b
            if(User_group=='Admin'):
                form = cgi.FieldStorage()
                if(form.has_key('which')):
                    action = form['which'].value
                    if (action == 'update_all_excel_files'):
                        Pig_ID = form['Pig_ID'].value
                        raj_excel_export.lhost = lhost
                        raj_excel_export.lport = lport
                        raj_excel_export.luser = luser
                        raj_excel_export.lpasswd = lpasswd
                        raj_excel_export.ldb = ldb
                        raj_excel_export.studyname="MONR"

raj_excel_export.excel_base_dir="/project/beilmanlab.umn.edu-
upload/"+raj_excel_export.studyname+"/excel_files/"

#raj_excel_export.excel_save_dir=raj_excel_export.excel_base_dir

raj_excel_export.excel_save_dir=raj_excel_export.excel_base_dir+Pig_ID+'/'
Updated_by=User_id
create_paths.append(raj_excel_export.excel_save_dir)
try:
    #Make sure the path exists
    for the_dir in create_paths:
        if not os.path.exists(the_dir):
            os.umask(0)
            os.makedirs(the_dir,0771)
        else:
            pass
    except:
        pass

#####Individual#####

```

```
#####  
#####
```

```
raj_excel_export.excel_file_name="monr_vitals_and_gases_individual_only_events.xls"  
"
```

```
raj_excel_export.tables_match={'t1':'monr_vitals_manual_10_min','t2':'monr_vitals_manual_one_hour','t3':'monr_identifier_table'}
```

```
raj_excel_export.the_query="""SELECT  
`t1`.Pig_ID,`t3`.Pig_group,`t1`.Date_of_experiment,`t1`.Time_of_experiment,`t1`.Event_comments,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanAr  
tPres,`t1`.PA_Sys,`t1`.PA_Dia,`t2`.Wedge_pressure,`t2`.Urine_output,`t2`.Bladder_p  
ressure,`t2`.Temperature,`t2`.Cardiac_output,`t1`.Propofol,`t1`.BIS,`t1`.FiO2,`t1`.StO2_15mm,`t1`.THI_15mm,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t2`.Chesttube_o  
utput,`t2`.Arterial_pH,`t2`.Arterial_PC02,`t2`.Arterial_PO2,`t2`.Arterial_O2_sat,`  
t2`.Venous_PO2,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.DO2,`t2`.VO2,`t2`.ER02,`t2`.Lactate  
,`t2`.Base_deficit,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Potassium,`t2`.Calc  
ium,`t2`.Glucose FROM `monr_vitals_manual_10_min` AS `t1` LEFT JOIN  
`monr_vitals_manual_one_hour` AS `t2` ON  
`t1`.Date_of_experiment=`t2`.Date_of_experiment AND  
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_identifier_table`  
AS `t3` ON `t1`.Pig_ID=`t3`.Pig_ID WHERE `t1`.Pig_ID='%s' AND  
`t1`.Event_name<>'Other' ORDER BY  
`t1`.Date_of_experiment,`t1`.Time_of_experiment""%(Pig_ID)
```

```
raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export  
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)  
Data_type="Vitals_Gases_events"  
File_name=raj_excel_export.excel_file_name  
File_storage_path=raj_excel_export.excel_save_dir
```

```
File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name
```

```
update_individual_export_database(Data_type,File_name,File_storage_path,File_path)
```

```
#####Individual#####  
#####  
#####
```

```
raj_excel_export.excel_file_name="monr_vitals_and_gases_individual.xls"
```

```
raj_excel_export.tables_match={'t1':'monr_vitals_manual_10_min','t2':'monr_vitals_manual_one_hour','t3':'monr_identifier_table'}
```

```
raj_excel_export.the_query="""SELECT  
`t1`.Pig_ID,`t3`.Pig_group,`t1`.Date_of_experiment,`t1`.Time_of_experiment,`t1`.Event_comments,`t1`.Event_name,`t1`.Heart_Rate,`t1`.Art_Sys,`t1`.Art_Dia,`t1`.MeanAr  
tPres,`t1`.PA_Sys,`t1`.PA_Dia,`t2`.Wedge_pressure,`t2`.Urine_output,`t2`.Bladder_p  
ressure,`t2`.Temperature,`t2`.Cardiac_output,`t1`.Propofol,`t1`.BIS,`t1`.FiO2,`t1`.StO2_15mm,`t1`.THI_15mm,`t1`.sto2_multidepth,`t1`.thi_multidepth,`t2`.Chesttube_o  
utput,`t2`.Arterial_pH,`t2`.Arterial_PC02,`t2`.Arterial_PO2,`t2`.Arterial_O2_sat,`  
t2`.Venous_PO2,`t2`.Venous_O2_sat,`t2`.Hb,`t2`.DO2,`t2`.VO2,`t2`.ER02,`t2`.Lactate  
,`t2`.Base_deficit,`t2`.Specificgravity_urine,`t2`.Sodium,`t2`.Potassium,`t2`.Calc  
ium,`t2`.Glucose FROM `monr_vitals_manual_10_min` AS `t1` LEFT JOIN  
`monr_vitals_manual_one_hour` AS `t2` ON  
`t1`.Date_of_experiment=`t2`.Date_of_experiment AND  
`t1`.Time_of_experiment=`t2`.Time_of_experiment LEFT JOIN `monr_identifier_table`  
AS `t3` ON `t1`.Pig_ID=`t3`.Pig_ID WHERE `t1`.Pig_ID='%s' ORDER BY  
`t1`.Date_of_experiment,`t1`.Time_of_experiment""%(Pig_ID)
```



```

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Vitals_Gases"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_individual_export_database(Data_type,File_name,File_storage_path,File_path)

#####Individual#####
#####

raj_excel_export.excel_file_name="monr_labs_individual_only_events.xls"

raj_excel_export.tables_match={'t1':'monr_lab_values','t2':'monr_identifier_table'
}

    raj_excel_export.the_query=""SELECT
`t1`.Pig_ID,`t2`.Pig_group,`t1`.Event_name,`t1`.Bilirubin_total,`t1`.Albumin,`t1`.
Protein_total,`t1`.Alk_phosphatase,`t1`.ALT,`t1`.AST,`t1`.Urea_nitrogen,`t1`.CK,`t
1`.Creatinine,`t1`.LD,`t1`.Platelet_count,`t1`.Urine_creatinine FROM
`monr_lab_values` AS `t1` LEFT JOIN `monr_identifier_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Pig_ID='%s' ORDER BY `t1`.Pig_ID,CASE
`t1`.Event_name WHEN 'Baseline' THEN 1 WHEN 'Shock45' THEN 2 WHEN '3' THEN 3 WHEN
'5' THEN 4 WHEN '9' THEN 5 WHEN '17' THEN 6 WHEN '21' THEN 7 WHEN '45' THEN 8 WHEN
'69' THEN 9 END""%(Pig_ID)

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="Labs_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_individual_export_database(Data_type,File_name,File_storage_path,File_path)

#####Individual#####
#####

raj_excel_export.excel_file_name="monr_gcs_individual_only_events.xls"

raj_excel_export.tables_match={'t1':'monr_gcs_scores','t2':'monr_identifier_table'
}

    raj_excel_export.the_query=""SELECT
`t1`.Pig_ID,`t2`.Pig_group,`t1`.Event_name,`t1`.GCS_score FROM `monr_gcs_scores`
AS `t1` LEFT JOIN `monr_identifier_table` AS `t2` ON `t1`.Pig_ID=`t2`.Pig_ID WHERE
`t1`.Pig_ID='%s' ORDER BY Pig_ID,CASE Event_name WHEN 'Baseline' THEN 1 WHEN '21'
THEN 2 WHEN '45' THEN 3 WHEN '69' THEN 4 END""%(Pig_ID)

raj_excel_export.create_and_save_excel(raj_excel_export.studyname,raj_excel_export
.excel_file_name,raj_excel_export.the_query,raj_excel_export.tables_match)
    Data_type="GCS_events"
    File_name=raj_excel_export.excel_file_name
    File_storage_path=raj_excel_export.excel_save_dir

```

```

File_path=raj_excel_export.excel_save_dir+raj_excel_export.excel_file_name

update_individual_export_database(Data_type,File_name,File_storage_path,File_path)

#####
#####
#####
        print
"""document.getElementById('message1').innerHTML='Excel files successfully
updated. Files not found = %d, and Errors found = %d.<br />If any of these numbers
are not zero, inform data manager';"""%(files_not_found,errors_found)
        else:
            print """document.getElementById('message1').innerHTML='An
error occurred. Please contact data manager. Files not found = %d, and Errors
found = %d';"""%(files_not_found,errors_found)
            else:
                print """document.getElementById('message1').innerHTML='An
error occurred. Please contact data manager. Files not found = %d, and Errors
found = %d';"""%(files_not_found,errors_found)
                else:
                    print """document.getElementById('message1').innerHTML='Sorry,
only an Administrator or a Lab User can update excel data.';"""
                    else:
                        print """document.getElementById('message1').innerHTML='An error
occurred. Please contact data manager. Error no:1';"""
                        except:
                            print """document.getElementById('message1').innerHTML='An error occurred.
Please contact data manager. Error no:2';"""
                            else:
                                print """document.getElementById('message1').innerHTML='An error occurred.
Please contact data manager. Error no:3';"""

```

Appendix B. Database tables' structure

The following MySQL code represents only the database tables that are currently present.

These tables and their structure will frequently change based on the project needs. For the current database tables, please contact the author.

```
/* The MySQL code below can be used to reproduce the database for the SCC
laboratory. Some of the tables are created automatically when related files are
uploaded to the database. */
```

```
DROP TABLE IF EXISTS `sccl_authentication`;
CREATE TABLE `sccl_authentication` (
  `User_id` varchar(15) collate utf8_bin NOT NULL,
  `Password` varchar(140) collate utf8_bin NOT NULL,
  `Temp_key` varchar(140) collate utf8_bin default NULL,
  `Email` varchar(50) collate utf8_bin NOT NULL,
  `User_first_name` varchar(15) collate utf8_bin NOT NULL,
  `User_last_name` varchar(15) collate utf8_bin default NULL,
  `User_middle_initial` varchar(2) collate utf8_bin default NULL,
  `Group_name` enum('Admin','Lab_user','Guest','Special') collate utf8_bin NOT
NULL default 'Lab_user',
  `TestOnly` enum('Yes','No') collate utf8_bin NOT NULL default 'No',
  `MONR` enum('Yes','No') collate utf8_bin NOT NULL default 'No',
  `PONR` enum('Yes','No') collate utf8_bin NOT NULL default 'No',
  `DARPA` enum('Yes','No') collate utf8_bin NOT NULL default 'No',
  `All_studies` enum('Yes','No') collate utf8_bin NOT NULL default 'No',
  `Account_creation_date` date NOT NULL,
  `Last_login_time` timestamp NOT NULL default '0000-00-00 00:00:00',
  `First_login` enum('Yes','No') collate utf8_bin NOT NULL default 'Yes',
  PRIMARY KEY (`User_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_identifier_table`;
CREATE TABLE `monr_identifier_table` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Start_date` date NOT NULL,
  `End_date` date default NULL,
  `Weight` float NOT NULL,
  `Pig_group` int(1) NOT NULL,
  `Blood_removed` int(4) default NULL,
  `OR_fluids` int(4) default NULL,
  `Blood_given` int(4) default NULL,
  `LR_given` int(4) default NULL,
  `Total_volume_given` int(4) default NULL,
  `Died` enum('YES','NO') collate utf8_bin default NULL,
  `Time_of_death` time default NULL,
  `Cause_of_death` varchar(30) collate utf8_bin default NULL,
  `Comments` varchar(1000) collate utf8_bin default NULL,
  `Datainput_status` enum('Finished','Not finished') collate utf8_bin NOT NULL
default 'Not finished',
  PRIMARY KEY (`Pig_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

```

DROP TABLE IF EXISTS `monr_allsubjects_export`;
CREATE TABLE `monr_allsubjects_export` (
  `Data_type` varchar(50) collate utf8_bin NOT NULL,
  `File_name` varchar(150) collate utf8_bin default NULL,
  `File_storage_path` varchar(500) collate utf8_bin default NULL,
  `File_size` int(10) default NULL,
  `Last_update_time` timestamp NOT NULL default '0000-00-00 00:00:00' on update
CURRENT_TIMESTAMP,
  `Updated_by` varchar(15) collate utf8_bin default NULL,
  PRIMARY KEY (`Data_type`),
  KEY `Updated_by` (`Updated_by`),
  CONSTRAINT `monr_allsubjects_export_ibfk_1` FOREIGN KEY (`Updated_by`)
REFERENCES `sccl_authentication` (`User_id`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

/* The table below is automatically generated by a Python code after file upload for the first time */

```

DROP TABLE IF EXISTS `monr_binning_muscle`;
CREATE TABLE `monr_binning_muscle` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `9.98000` double(11,9) default NULL,
  `9.94000` double(11,9) default NULL,
  `9.90000` double(11,9) default NULL,
  `9.86000` double(11,9) default NULL,
  `9.82000` double(11,9) default NULL,
  | | | | | |
  | | | | | |
  `0.18000` double(11,9) default NULL,
  `0.14000` double(11,9) default NULL,
  `0.10000` double(11,9) default NULL,
  `0.06000` double(11,9) default NULL,
  `0.02000` double(11,9) default NULL,
  PRIMARY KEY (`Pig_ID`,`Event_name`),
  KEY `Pig_ID` (`Pig_ID`),
  CONSTRAINT `monr_binning_muscle_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

/* The table below is automatically generated by a Python code after file upload for the first time */

```

DROP TABLE IF EXISTS `monr_binning_urine`;
CREATE TABLE `monr_binning_urine` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `9.98000` double(11,9) default NULL,
  `9.94000` double(11,9) default NULL,
  `9.90000` double(11,9) default NULL,
  `9.86000` double(11,9) default NULL,
  `9.82000` double(11,9) default NULL,
  | | | | | |
  | | | | | |
  `0.18000` double(11,9) default NULL,
  `0.14000` double(11,9) default NULL,
  `0.10000` double(11,9) default NULL,
  `0.06000` double(11,9) default NULL,

```

```

`0.02000` double(11,9) default NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_binning_urine_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identififier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_event_definitions`;
CREATE TABLE `monr_event_definitions` (
  `Old_Event_name` varchar(20) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `Event_Instance` mediumtext collate utf8_bin,
  PRIMARY KEY (`Event_name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_fileupload`;
CREATE TABLE `monr_fileupload` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Data_type` varchar(35) collate utf8_bin NOT NULL,
  `File_name` varchar(150) collate utf8_bin default NULL,
  `File_storage_path` varchar(500) collate utf8_bin default NULL,
  `Upload_time` timestamp NOT NULL default '0000-00-00 00:00:00' on update
CURRENT_TIMESTAMP,
  `File_type` varchar(10) collate utf8_bin default NULL,
  `File_size` int(10) default NULL,
  PRIMARY KEY (`Data_type`,`Pig_ID`),
  KEY `monr_fileupload_ibfk_1` (`Pig_ID`),
  CONSTRAINT `monr_fileupload_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identififier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_gcs_scores`;
CREATE TABLE `monr_gcs_scores` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `GCS_score` int(2) default NULL,
  PRIMARY KEY (`Pig_ID`,`Event_name`),
  KEY `Pig_ID` (`Pig_ID`),
  CONSTRAINT `monr_gcs_scores_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identififier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

/* The table below is automatically generated by a Python code after file upload
for the first time */

DROP TABLE IF EXISTS `monr_hnmr_liver`;
CREATE TABLE `monr_hnmr_liver` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `Normalization` enum('No','Wet_tissue','Lyophilized_tissue') collate utf8_bin
NOT NULL default 'No',
  `Weight_of_tissue` float(8,4) default NULL,
  `Normalization_factor` float(8,4) default NULL,
  `2-Aminobutyrate` float(8,4) default NULL,
  `3-Hydroxybutyrate` float(8,4) default NULL,
  `3-Hydroxyisovalerate` float(8,4) default NULL,
  `ADP` float(8,4) default NULL,
  `AMP` float(8,4) default NULL,

```

```

`ATP` float(8,4) default NULL,
`Acetate` float(8,4) default NULL,
`Adenosine` float(8,4) default NULL,
`Alanine` float(8,4) default NULL,
`Allantoin` float(8,4) default NULL,
`Alloisoleucine` float(8,4) default NULL,
`Asparagine` float(8,4) default NULL,
`Aspartate` float(8,4) default NULL,
`Betaine` float(8,4) default NULL,
`Carnosine` float(8,4) default NULL,
`Choline` float(8,4) default NULL,
`Citrate` float(8,4) default NULL,
`Creatine` float(8,4) default NULL,
`Creatinine` float(8,4) default NULL,
`DSS (Chemical Shape Indicator)` float(8,4) default NULL,
`Formate` float(8,4) default NULL,
`Fumarate` float(8,4) default NULL,
`Glucose` float(8,4) default NULL,
`Glutamate` float(8,4) default NULL,
`Glutamine` float(8,4) default NULL,
`Glutathione` float(8,4) default NULL,
`Glycine` float(8,4) default NULL,
`Glycolate` float(8,4) default NULL,
`Histidine` float(8,4) default NULL,
`Hypoxanthine` float(8,4) default NULL,
`Isoleucine` float(8,4) default NULL,
`Isovalerate` float(8,4) default NULL,
`Lactate` float(8,4) default NULL,
`Leucine` float(8,4) default NULL,
`Lysine` float(8,4) default NULL,
`Methionine` float(8,4) default NULL,
`NAD+` float(8,4) default NULL,
`Niacinamide` float(8,4) default NULL,
`O-Phosphocholine` float(8,4) default NULL,
`Oxalacetate` float(8,4) default NULL,
`Phenylalanine` float(8,4) default NULL,
`Pyroglutamate` float(8,4) default NULL,
`Pyruvate` float(8,4) default NULL,
`Serine` float(8,4) default NULL,
`Succinate` float(8,4) default NULL,
`Taurine` float(8,4) default NULL,
`Threonine` float(8,4) default NULL,
`Tyrosine` float(8,4) default NULL,
`Valine` float(8,4) default NULL,
`sn-Glycero-3-phosphocholine` float(8,4) default NULL,
`3-Aminoisobutyrate` float(8,4) default NULL,
`Arginine` float(8,4) default NULL,
`Benzoate` float(8,4) default NULL,
`Proline` float(8,4) default NULL,
`S-Adenosylhomocysteine` float(8,4) default NULL,
`Xanthine` float(8,4) default NULL,
`Dimethylamine` float(8,4) default NULL,
`UDP-glucose` float(8,4) default NULL,
`Guanosine` float(8,4) default NULL,
`Maltose` float(8,4) default NULL,
`Sucrose` float(8,4) default NULL,
`NADP+` float(8,4) default NULL,
PRIMARY KEY (`Pig_ID`, `Event_name`, `Normalization`),

```

```

KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_hnmr_liver_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifiaer_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

```

/* The table below is automatically generated by a Python code after file upload
for the first time */

```

```

DROP TABLE IF EXISTS `monr_hnmr_muscle`;
CREATE TABLE `monr_hnmr_muscle` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `Normalization` enum('No', 'Wet_tissue', 'Lyophilized_tissue') collate utf8_bin
NOT NULL default 'No',
  `Weight_of_tissue` float(8,4) default NULL,
  `Normalization_factor` float(8,4) default NULL,
  `2-Oxoglutarate` float(8,4) default NULL,
  `3-Hydroxybutyrate` float(8,4) default NULL,
  `3-Hydroxyisovalerate` float(8,4) default NULL,
  `3-Indoxylsulfate` float(8,4) default NULL,
  `5,6-Dihydrouracil` float(8,4) default NULL,
  `AMP` float(8,4) default NULL,
  `ATP` float(8,4) default NULL,
  `Acetate` float(8,4) default NULL,
  `Alanine` float(8,4) default NULL,
  `Allantoin` float(8,4) default NULL,
  `Betaine` float(8,4) default NULL,
  `Carnitine` float(8,4) default NULL,
  `Carnosine` float(8,4) default NULL,
  `Choline` float(8,4) default NULL,
  `Citrate` float(8,4) default NULL,
  `Creatine` float(8,4) default NULL,
  `Creatine phosphate` float(8,4) default NULL,
  `Formate` float(8,4) default NULL,
  `Fumarate` float(8,4) default NULL,
  `Glucose` float(8,4) default NULL,
  `Glutamate` float(8,4) default NULL,
  `Glutamine` float(8,4) default NULL,
  `Glycine` float(8,4) default NULL,
  `Histamine` float(8,4) default NULL,
  `Histidine` float(8,4) default NULL,
  `Imidazole` float(8,4) default NULL,
  `Isoleucine` float(8,4) default NULL,
  `Lactate` float(8,4) default NULL,
  `Leucine` float(8,4) default NULL,
  `Malate` float(8,4) default NULL,
  `Mannose` float(8,4) default NULL,
  `Methylmalonate` float(8,4) default NULL,
  `NAD+` float(8,4) default NULL,
  `Niacinamide` float(8,4) default NULL,
  `O-Phosphocholine` float(8,4) default NULL,
  `Pantothenate` float(8,4) default NULL,
  `Phenylalanine` float(8,4) default NULL,
  `Pyruvate` float(8,4) default NULL,
  `Ribose` float(8,4) default NULL,
  `Succinate` float(8,4) default NULL,
  `Taurine` float(8,4) default NULL,
  `Trimethylamine` float(8,4) default NULL,

```

```

`Trimethylamine N-oxide` float(8,4) default NULL,
`Tyramine` float(8,4) default NULL,
`Valine` float(8,4) default NULL,
`myo-Inositol` float(8,4) default NULL,
`sn-Glycero-3-phosphocholine` float(8,4) default NULL,
`π-Methylhistidine` float(8,4) default NULL,
`τ-Methylhistidine` float(8,4) default NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`,`Normalization`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_hnmr_muscle_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifrier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

/* The table below is automatically generated by a Python code after file upload for the first time */

```

DROP TABLE IF EXISTS `monr_hnmr_urine`;
CREATE TABLE `monr_hnmr_urine` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `Normalization` enum('No','Yes') collate utf8_bin NOT NULL default 'No',
  `Normalization_factor` float(8,4) default NULL,
  `1-Methylnicotinamide` float(8,4) default NULL,
  `2-Methylglutarate` float(8,4) default NULL,
  `2-Oxoglutarate` float(8,4) default NULL,
  `Acetate` float(8,4) default NULL,
  `Alanine` float(8,4) default NULL,
  `Allantoin` float(8,4) default NULL,
  `Ascorbate` float(8,4) default NULL,
  `Betaine` float(8,4) default NULL,
  `Choline` float(8,4) default NULL,
  `Citrate` float(8,4) default NULL,
  `Creatine` float(8,4) default NULL,
  `Creatinine` float(8,4) default NULL,
  `Dimethylamine` float(8,4) default NULL,
  `Formate` float(8,4) default NULL,
  `Glucose` float(8,4) default NULL,
  `Glutamate` float(8,4) default NULL,
  `Glycine` float(8,4) default NULL,
  `Hippurate` float(8,4) default NULL,
  `Hypoxanthine` float(8,4) default NULL,
  `Lactate` float(8,4) default NULL,
  `Mannose` float(8,4) default NULL,
  `N,N-Dimethylglycine` float(8,4) default NULL,
  `Niacinamide` float(8,4) default NULL,
  `Phenylacetyl glycine` float(8,4) default NULL,
  `Pyruvate` float(8,4) default NULL,
  `Quinolate` float(8,4) default NULL,
  `Succinate` float(8,4) default NULL,
  `Trigonelline` float(8,4) default NULL,
  `Trimethylamine` float(8,4) default NULL,
  `Trimethylamine N-oxide` float(8,4) default NULL,
  `Tryptophan` float(8,4) default NULL,
  `Tyramine` float(8,4) default NULL,
  `Tyrosine` float(8,4) default NULL,
  `1,6-Anhydro-β-D-glucose` float(8,4) default NULL,
  `4-Hydroxybenzoate` float(8,4) default NULL,
  `Adenosine` float(8,4) default NULL,

```



```

`Fumarate` float(8,4) default NULL,
`Glucitol` float(8,4) default NULL,
`Glutamine` float(8,4) default NULL,
`Glutathione` float(8,4) default NULL,
`Glycerol` float(8,4) default NULL,
`Glycylproline` float(8,4) default NULL,
`Homogentisate` float(8,4) default NULL,
`Inosine` float(8,4) default NULL,
`Isoleucine` float(8,4) default NULL,
`Lysine` float(8,4) default NULL,
`Mannitol` float(8,4) default NULL,
`Methylguanidine` float(8,4) default NULL,
`N-Isovaleroylglycine` float(8,4) default NULL,
`Oxypurinol` float(8,4) default NULL,
`Proline` float(8,4) default NULL,
`Taurine` float(8,4) default NULL,
`Urea` float(8,4) default NULL,
`Valine` float(8,4) default NULL,
`Xanthine` float(8,4) default NULL,
`Xanthosine` float(8,4) default NULL,
`cis-Aconitate` float(8,4) default NULL,
`3-Hydroxyisovalerate` float(8,4) default NULL,
`Methanol` float(8,4) default NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`,`Normalization`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_hnmr_urine_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifiler_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_individual_export`;
CREATE TABLE `monr_individual_export` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Data_type` varchar(50) collate utf8_bin NOT NULL,
  `File_name` varchar(150) collate utf8_bin default NULL,
  `File_storage_path` varchar(500) collate utf8_bin default NULL,
  `File_size` int(10) default NULL,
  `Last_update_time` timestamp NOT NULL default '0000-00-00 00:00:00' on update
CURRENT_TIMESTAMP,
  `Updated_by` varchar(15) collate utf8_bin default NULL,
  PRIMARY KEY (`Data_type`,`Pig_ID`),
  KEY `Pig_ID` (`Pig_ID`),
  KEY `Updated_by` (`Updated_by`),
  CONSTRAINT `monr_individual_export_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifiler_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `monr_individual_export_ibfk_2` FOREIGN KEY (`Updated_by`) REFERENCES
`scc1_authentication` (`User_id`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_lab_values`;
CREATE TABLE `monr_lab_values` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `Bilirubin_total` float(4,2) default NULL,
  `Albumin` float(3,1) default NULL,
  `Protein_total` float(3,1) default NULL,
  `Alk_phosphatase` int(3) default NULL,
  `ALT` int(3) default NULL,
  `AST` int(3) default NULL,

```

```

`Urea_nitrogen` int(2) default NULL,
`CK` int(5) default NULL,
`Creatinine` float(4,2) default NULL,
`LD` int(5) default NULL,
`Platelet_count` int(4) default NULL,
`Urine_osmolality` int(3) default NULL,
`Urine_creatinine` int(3) default NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_lab_values_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifrier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

/* The table below is automatically generated by a Python code after file upload for the first time */

```

DROP TABLE IF EXISTS `monr_pnmr_muscle`;
CREATE TABLE `monr_pnmr_muscle` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `Normalization` enum('No','Wet_tissue','Lyophilized_tissue') collate utf8_bin
NOT NULL default 'No',
  `Weight_of_tissue` float(8,4) default NULL,
  `Normalization_factor` float(8,4) default NULL,
  `Value_type` enum('Start_ppm','End_ppm','Integral') collate utf8_bin NOT NULL
default 'Start_ppm',
  `1` double(14,8) default NULL,
  `2` double(14,8) default NULL,
  `3` double(14,8) default NULL,
  `4` double(14,8) default NULL,
  `5` double(14,8) default NULL,
  `6` double(14,8) default NULL,
  `7` double(14,8) default NULL,
  `8` double(14,8) default NULL,
  `9` double(14,8) default NULL,
  `10` double(14,8) default NULL,
  `11` double(14,8) default NULL,
  `12` double(14,8) default NULL,
  `13` double(14,8) default NULL,
  `14` double(14,8) default NULL,
  `15` double(14,8) default NULL,
  `16` double(14,8) default NULL,
  `17` double(14,8) default NULL,
  `18` double(14,8) default NULL,
  PRIMARY KEY (`Pig_ID`,`Event_name`,`Normalization`,`Value_type`),
  KEY `Pig_ID` (`Pig_ID`),
  CONSTRAINT `monr_pnmr_muscle_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifrier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

/* The table below is automatically generated by a Python code after file upload for the first time. This table is generated as a backup for monr_hnmr_liver so that the utf-8 column headings won't cause problems during SAS data analyses if proc sql is not used for analyses. */

```

DROP TABLE IF EXISTS `monr_sas_hnmr_liver`;
CREATE TABLE `monr_sas_hnmr_liver` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,

```

```

`Event_name` int(2) NOT NULL,
`Normalization` enum('No', 'Wet_tissue', 'Lyophilized_tissue') collate utf8_bin
NOT NULL default 'No',
`Weight_of_tissue` float(8,4) default NULL,
`Normalization_factor` float(8,4) default NULL,
`2-Aminobutyrate` float(8,4) default NULL,
`3-Hydroxybutyrate` float(8,4) default NULL,
`3-Hydroxyisovalerate` float(8,4) default NULL,
`ADP` float(8,4) default NULL,
`AMP` float(8,4) default NULL,
`ATP` float(8,4) default NULL,
`Acetate` float(8,4) default NULL,
`Adenosine` float(8,4) default NULL,
`Alanine` float(8,4) default NULL,
`Allantoin` float(8,4) default NULL,
`Alloisoleucine` float(8,4) default NULL,
`Asparagine` float(8,4) default NULL,
`Aspartate` float(8,4) default NULL,
`Betaine` float(8,4) default NULL,
`Carnosine` float(8,4) default NULL,
`Choline` float(8,4) default NULL,
`Citrate` float(8,4) default NULL,
`Creatine` float(8,4) default NULL,
`Creatinine` float(8,4) default NULL,
`DSS (Chemical Shape Indicator)` float(8,4) default NULL,
`Formate` float(8,4) default NULL,
`Fumarate` float(8,4) default NULL,
`Glucose` float(8,4) default NULL,
`Glutamate` float(8,4) default NULL,
`Glutamine` float(8,4) default NULL,
`Glutathione` float(8,4) default NULL,
`Glycine` float(8,4) default NULL,
`Glycolate` float(8,4) default NULL,
`Histidine` float(8,4) default NULL,
`Hypoxanthine` float(8,4) default NULL,
`Isoleucine` float(8,4) default NULL,
`Isovalerate` float(8,4) default NULL,
`Lactate` float(8,4) default NULL,
`Leucine` float(8,4) default NULL,
`Lysine` float(8,4) default NULL,
`Methionine` float(8,4) default NULL,
`NAD+` float(8,4) default NULL,
`Niacinamide` float(8,4) default NULL,
`O-Phosphocholine` float(8,4) default NULL,
`Oxalacetate` float(8,4) default NULL,
`Phenylalanine` float(8,4) default NULL,
`Pyroglutamate` float(8,4) default NULL,
`Pyruvate` float(8,4) default NULL,
`Serine` float(8,4) default NULL,
`Succinate` float(8,4) default NULL,
`Taurine` float(8,4) default NULL,
`Threonine` float(8,4) default NULL,
`Tyrosine` float(8,4) default NULL,
`Valine` float(8,4) default NULL,
`sn-Glycero-3-phosphocholine` float(8,4) default NULL,
`3-Aminoisobutyrate` float(8,4) default NULL,
`Arginine` float(8,4) default NULL,
`Benzoate` float(8,4) default NULL,

```

```

`Proline` float(8,4) default NULL,
`S-Adenosylhomocysteine` float(8,4) default NULL,
`Xanthine` float(8,4) default NULL,
`Dimethylamine` float(8,4) default NULL,
`UDP-glucose` float(8,4) default NULL,
`Guanosine` float(8,4) default NULL,
`Maltose` float(8,4) default NULL,
`Sucrose` float(8,4) default NULL,
`NADP+` float(8,4) default NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`,`Normalization`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_sas_hnmr_liver_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifer_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

/* The table below is automatically generated by a Python code after file upload for the first time. This table is generated as a backup for monr_hnmr_muscle so that the utf-8 column headings won't cause problems during SAS data analyses if proc sql is not used for analyses. */

```

DROP TABLE IF EXISTS `monr_sas_hnmr_muscle`;
CREATE TABLE `monr_sas_hnmr_muscle` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `Normalization` enum('No','Wet_tissue','Lyophilized_tissue') collate utf8_bin
NOT NULL default 'No',
  `Weight_of_tissue` float(8,4) default NULL,
  `Normalization_factor` float(8,4) default NULL,
  `2-Oxoglutarate` float(8,4) default NULL,
  `3-Hydroxybutyrate` float(8,4) default NULL,
  `3-Hydroxyisovalerate` float(8,4) default NULL,
  `3-Indoxylsulfate` float(8,4) default NULL,
  `5,6-Dihydrouracil` float(8,4) default NULL,
  `AMP` float(8,4) default NULL,
  `ATP` float(8,4) default NULL,
  `Acetate` float(8,4) default NULL,
  `Alanine` float(8,4) default NULL,
  `Allantoin` float(8,4) default NULL,
  `Betaine` float(8,4) default NULL,
  `Carnitine` float(8,4) default NULL,
  `Carnosine` float(8,4) default NULL,
  `Choline` float(8,4) default NULL,
  `Citrate` float(8,4) default NULL,
  `Creatine` float(8,4) default NULL,
  `Creatine phosphate` float(8,4) default NULL,
  `Formate` float(8,4) default NULL,
  `Fumarate` float(8,4) default NULL,
  `Glucose` float(8,4) default NULL,
  `Glutamate` float(8,4) default NULL,
  `Glutamine` float(8,4) default NULL,
  `Glycine` float(8,4) default NULL,
  `Histamine` float(8,4) default NULL,
  `Histidine` float(8,4) default NULL,
  `Imidazole` float(8,4) default NULL,
  `Isoleucine` float(8,4) default NULL,
  `Lactate` float(8,4) default NULL,
  `Leucine` float(8,4) default NULL,
  `Malate` float(8,4) default NULL,

```

```

`Mannose` float(8,4) default NULL,
`Methylmalonate` float(8,4) default NULL,
`NAD+` float(8,4) default NULL,
`Niacinamide` float(8,4) default NULL,
`O-Phosphocholine` float(8,4) default NULL,
`Pantothenate` float(8,4) default NULL,
`Phenylalanine` float(8,4) default NULL,
`Pyruvate` float(8,4) default NULL,
`Ribose` float(8,4) default NULL,
`Succinate` float(8,4) default NULL,
`Taurine` float(8,4) default NULL,
`Trimethylamine` float(8,4) default NULL,
`Trimethylamine N-oxide` float(8,4) default NULL,
`Tyramine` float(8,4) default NULL,
`Valine` float(8,4) default NULL,
`myo-Inositol` float(8,4) default NULL,
`sn-Glycero-3-phosphocholine` float(8,4) default NULL,
`L-Methylhistidine` float(8,4) default NULL,
`D,-Methylhistidine` float(8,4) default NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`,`Normalization`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_sas_hnmr_muscle_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifrier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

/* The table below is automatically generated by a Python code after file upload for the first time. This table is generated as a backup for monr_hnmr_urine so that the utf-8 column headings won't cause problems during SAS data analyses if proc sql is not used for analyses. */

```

DROP TABLE IF EXISTS `monr_sas_hnmr_urine`;
CREATE TABLE `monr_sas_hnmr_urine` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `Normalization` enum('No','Yes') collate utf8_bin NOT NULL default 'No',
  `Normalization_factor` float(8,4) default NULL,
  `1-Methylnicotinamide` float(8,4) default NULL,
  `2-Methylglutarate` float(8,4) default NULL,
  `2-Oxoglutarate` float(8,4) default NULL,
  `3-Hydroxyisovalerate` float(8,4) default NULL,
  `4-Hydroxyphenylacetate` float(8,4) default NULL,
  `Acetate` float(8,4) default NULL,
  `Alanine` float(8,4) default NULL,
  `Allantoin` float(8,4) default NULL,
  `Ascorbate` float(8,4) default NULL,
  `Betaine` float(8,4) default NULL,
  `Choline` float(8,4) default NULL,
  `Citrate` float(8,4) default NULL,
  `Citrulline` float(8,4) default NULL,
  `Creatine` float(8,4) default NULL,
  `Creatinine` float(8,4) default NULL,
  `Dimethylamine` float(8,4) default NULL,
  `Formate` float(8,4) default NULL,
  `Glucose` float(8,4) default NULL,
  `Glutamate` float(8,4) default NULL,
  `Glycine` float(8,4) default NULL,
  `Hippurate` float(8,4) default NULL,
  `Homocitrulline` float(8,4) default NULL,

```

```

`Hypoxanthine` float(8,4) default NULL,
`Lactate` float(8,4) default NULL,
`Mannose` float(8,4) default NULL,
`N,N-Dimethylglycine` float(8,4) default NULL,
`Niacinamide` float(8,4) default NULL,
`Phenylacetylglycine` float(8,4) default NULL,
`Pyruvate` float(8,4) default NULL,
`Quinolate` float(8,4) default NULL,
`Succinate` float(8,4) default NULL,
`Trigonelline` float(8,4) default NULL,
`Trimethylamine` float(8,4) default NULL,
`Trimethylamine N-oxide` float(8,4) default NULL,
`Tryptophan` float(8,4) default NULL,
`Tyramine` float(8,4) default NULL,
`Tyrosine` float(8,4) default NULL,
`1,6-Anhydro-Î²-D-glucose` float(8,4) default NULL,
`4-Hydroxybenzoate` float(8,4) default NULL,
`Adenosine` float(8,4) default NULL,
`Fumarate` float(8,4) default NULL,
`Glucitol` float(8,4) default NULL,
`Glutamine` float(8,4) default NULL,
`Glutathione` float(8,4) default NULL,
`Glycerol` float(8,4) default NULL,
`Glycylproline` float(8,4) default NULL,
`Homogentisate` float(8,4) default NULL,
`Inosine` float(8,4) default NULL,
`Isoleucine` float(8,4) default NULL,
`Lysine` float(8,4) default NULL,
`Mannitol` float(8,4) default NULL,
`Methylguanidine` float(8,4) default NULL,
`N-Isovalerylglycine` float(8,4) default NULL,
`Oxypurinol` float(8,4) default NULL,
`Proline` float(8,4) default NULL,
`Taurine` float(8,4) default NULL,
`Urea` float(8,4) default NULL,
`Valine` float(8,4) default NULL,
`Xanthine` float(8,4) default NULL,
`Xanthosine` float(8,4) default NULL,
`cis-Aconitate` float(8,4) default NULL,
`Methanol` float(8,4) default NULL,
PRIMARY KEY (`Pig_ID`, `Event_name`, `Normalization`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_sas_hnmr_urine_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifider_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

/* The table below is automatically generated by a Python code after file upload for the first time. This table is generated as a backup for monr_pnmr_muscle so that the utf-8 column headings won't cause problems during SAS data analyses if proc sql is not used for analyses. */

```

DROP TABLE IF EXISTS `monr_sas_pnmr_muscle`;
CREATE TABLE `monr_sas_pnmr_muscle` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `Normalization` enum('No', 'Wet_tissue', 'Lyophilized_tissue') collate utf8_bin
  NOT NULL default 'No',
  `Weight_of_tissue` float(8,4) default NULL,

```

```

`Normalization_factor` float(8,4) default NULL,
`Value_type` enum('Start_ppm','End_ppm','Integral') collate utf8_bin NOT NULL
default 'Start_ppm',
`1` double(14,8) default NULL,
`2` double(14,8) default NULL,
`3` double(14,8) default NULL,
`4` double(14,8) default NULL,
`5` double(14,8) default NULL,
`6` double(14,8) default NULL,
`7` double(14,8) default NULL,
`8` double(14,8) default NULL,
`9` double(14,8) default NULL,
`10` double(14,8) default NULL,
`11` double(14,8) default NULL,
`12` double(14,8) default NULL,
`13` double(14,8) default NULL,
`14` double(14,8) default NULL,
`15` double(14,8) default NULL,
`16` double(14,8) default NULL,
`17` double(14,8) default NULL,
`18` double(14,8) default NULL,
PRIMARY KEY (`Pig_ID`,`Event_name`,`Normalization`,`Value_type`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_sas_pnmr_muscle_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_shared_queries`;
CREATE TABLE `monr_shared_queries` (
  `Shared_with` varchar(15) collate utf8_bin NOT NULL,
  `Shared_by` varchar(15) collate utf8_bin NOT NULL,
  `Query_name` varchar(53) collate utf8_bin NOT NULL,
  `Shared_time` timestamp NOT NULL default CURRENT_TIMESTAMP on update
CURRENT_TIMESTAMP,
  PRIMARY KEY (`Shared_with`,`Shared_by`,`Query_name`),
  KEY `Shared_by` (`Shared_by`,`Query_name`),
  CONSTRAINT `monr_shared_queries_ibfk_1` FOREIGN KEY (`Shared_with`) REFERENCES
`scc1_authentication` (`User_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `monr_shared_queries_ibfk_2` FOREIGN KEY (`Shared_by`) REFERENCES
`scc1_authentication` (`User_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `monr_shared_queries_ibfk_3` FOREIGN KEY (`Shared_by`,`Query_name`)
REFERENCES `monr_sql_queries` (`User_id`,`Query_name`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_sql_queries`;
CREATE TABLE `monr_sql_queries` (
  `User_id` varchar(15) collate utf8_bin NOT NULL,
  `Query_name` varchar(53) collate utf8_bin NOT NULL,
  `Query_string` longtext collate utf8_bin,
  `Query_description` longtext collate utf8_bin,
  `Saved_time` timestamp NOT NULL default CURRENT_TIMESTAMP on update
CURRENT_TIMESTAMP,
  PRIMARY KEY (`User_id`,`Query_name`),
  CONSTRAINT `monr_sql_queries_ibfk_1` FOREIGN KEY (`User_id`) REFERENCES
`scc1_authentication` (`User_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

```

DROP TABLE IF EXISTS `monr_survey_completion_time`;
CREATE TABLE `monr_survey_completion_time` (
  `Serialnumber` mediumint(9) NOT NULL auto_increment,
  `Survey_completion_time` datetime NOT NULL,
  PRIMARY KEY (`Serialnumber`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_teg_data`;
CREATE TABLE `monr_teg_data` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,
  `R_min` float(3,1) default NULL,
  `K_min` float(3,1) default NULL,
  `Angle_deg` float(4,1) default NULL,
  `MA_mm` float(4,1) default NULL,
  PRIMARY KEY (`Pig_ID`,`Event_name`),
  KEY `Pig_ID` (`Pig_ID`),
  CONSTRAINT `monr_teg_data_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_temp_sql_query_storage`;
CREATE TABLE `monr_temp_sql_query_storage` (
  `Random_string` varchar(100) collate utf8_bin NOT NULL,
  `Query_string` longtext collate utf8_bin,
  `Saved_time` timestamp NOT NULL default CURRENT_TIMESTAMP on update
CURRENT_TIMESTAMP,
  PRIMARY KEY (`Random_string`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_user_survey_info`;
CREATE TABLE `monr_user_survey_info` (
  `User_id` varchar(15) collate utf8_bin NOT NULL,
  `Completed_survey` enum('Yes','No') collate utf8_bin NOT NULL default 'No',
  PRIMARY KEY (`User_id`),
  KEY `User_id` (`User_id`),
  CONSTRAINT `monr_user_survey_info_ibfk_1` FOREIGN KEY (`User_id`) REFERENCES
`scc1_authentication` (`User_id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

DROP TABLE IF EXISTS `monr_vitals_manual_10_min`;
CREATE TABLE `monr_vitals_manual_10_min` (
  `Date_of_experiment` date NOT NULL,
  `Time_of_experiment` time NOT NULL,
  `Heart_Rate` int(3) default NULL,
  `Art_Sys` int(3) default NULL,
  `Art_Dia` int(3) default NULL,
  `MeanArtPres` float(5,2) default NULL,
  `PA_Sys` int(3) default NULL,
  `PA_Dia` int(3) default NULL,
  `StO2_multidepth` int(3) default NULL,
  `THI_multidepth` float(3,1) default NULL,
  `StO2_15mm` int(3) default NULL,
  `THI_15mm` float(3,1) default NULL,
  `FiO2` float(4,1) default NULL,
  `Propofol` int(2) default NULL,
  `BIS` int(3) default NULL,
  `Temperature` float(3,1) default NULL,

```



```

`Event_name` int(2) NOT NULL,
`Event_comments` varchar(1000) collate utf8_bin default NULL,
`Pig_ID` varchar(6) collate utf8_bin NOT NULL,
PRIMARY KEY (`Date_of_experiment`,`Time_of_experiment`),
KEY `Pig_ID` (`Pig_ID`),
CONSTRAINT `monr_vitals_manual_10_min_ibfk_1` FOREIGN KEY (`Pig_ID`) REFERENCES
`monr_identifrier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

```

DROP TABLE IF EXISTS `monr_vitals_manual_one_hour`;
CREATE TABLE `monr_vitals_manual_one_hour` (
  `Date_of_experiment` date NOT NULL,
  `Time_of_experiment` time NOT NULL,
  `Wedge_pressure` int(2) default NULL,
  `Urine_output` int(3) default NULL,
  `Cardiac_output` float(3,1) default NULL,
  `Arterial_pH` float(4,2) default NULL,
  `Arterial_PC02` int(3) default NULL,
  `Arterial_P02` int(3) default NULL,
  `Arterial_O2_sat` int(3) default NULL,
  `Venous_P02` int(3) default NULL,
  `Venous_O2_sat` int(3) default NULL,
  `Hb` float(3,1) default NULL,
  `D02` float(5,3) default NULL,
  `V02` float(5,3) default NULL,
  `ER02` float(5,3) default NULL,
  `Lactate` float(3,1) default NULL,
  `Base_deficit` float(3,1) default NULL,
  `Bladder_pressure` float(4,1) default NULL,
  `Chesttube_output` float(4,1) default NULL,
  `Sodium` int(3) default NULL,
  `Potassium` float(3,1) default NULL,
  `Calcium` float(3,2) default NULL,
  `Glucose` int(3) default NULL,
  `Event_name` int(2) NOT NULL,
  `Pig_ID` varchar(6) collate utf8_bin default NULL,
  `Specificgravity_urine` float(4,3) default NULL,
  PRIMARY KEY (`Date_of_experiment`,`Time_of_experiment`),
  KEY `Pig_ID` (`Pig_ID`),
  CONSTRAINT `monr_vitals_manual_one_hour_ibfk_1` FOREIGN KEY (`Pig_ID`)
REFERENCES `monr_identifrier_table` (`Pig_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

```

DROP TABLE IF EXISTS `user_login_info`;
CREATE TABLE `user_login_info` (
  `User_id` varchar(15) collate utf8_bin NOT NULL,
  `User_ip` varchar(15) collate utf8_bin default NULL,
  `Login_time` datetime default NULL,
  PRIMARY KEY (`User_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

```

CREATE TABLE IF NOT EXISTS `monr_arterialpressure_realtime` (
  `Date_of_experiment` date NOT NULL,
  `Time_of_experiment` time NOT NULL,
  `Original_date` date NOT NULL,
  `Original_time` time NOT NULL,
  `Time_diff` time NOT NULL default '00:00:00',
  `Art_Sys` int(3) default NULL,

```

```

`Art_Dia` int(3) default NULL,
`MeanArtPres` float(5,2) default NULL,
`Event_name` int(2) NOT NULL default 99,
`Pig_ID` varchar(6) collate utf8_bin NOT NULL,
PRIMARY KEY (`Date_of_experiment`,`Time_of_experiment`),
INDEX (`Pig_ID`),
FOREIGN KEY (`Pig_ID`) REFERENCES `monr_identifier_table` (`Pig_ID`)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE IF NOT EXISTS `monr_papressure_realtime` (
  `Date_of_experiment` date NOT NULL,
  `Time_of_experiment` time NOT NULL,
  `Original_date` date NOT NULL,
  `Original_time` time NOT NULL,
  `Time_diff` time NOT NULL default '00:00:00',
  `PA_Sys` int(3) default NULL,
  `PA_Dia` int(3) default NULL,
  `Event_name` int(2) NOT NULL default 99,
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
PRIMARY KEY (`Date_of_experiment`,`Time_of_experiment`),
INDEX (`Pig_ID`),
FOREIGN KEY (`Pig_ID`) REFERENCES `monr_identifier_table` (`Pig_ID`)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE IF NOT EXISTS `monr_cardiacoutput_realtime` (
  `Date_of_experiment` date NOT NULL,
  `Time_of_experiment` time NOT NULL,
  `Original_date` date NOT NULL,
  `Original_time` time NOT NULL,
  `Time_diff` time NOT NULL default '00:00:00',
  `Cardiac_output` float(3,1) default NULL,
  `Event_name` int(2) NOT NULL default 99,
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
PRIMARY KEY (`Date_of_experiment`,`Time_of_experiment`),
INDEX (`Pig_ID`),
FOREIGN KEY (`Pig_ID`) REFERENCES `monr_identifier_table` (`Pig_ID`)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE IF NOT EXISTS `monr_sto2thi_realtime` (
  `Date_of_experiment` date NOT NULL,
  `Time_of_experiment` time NOT NULL,
  `Original_date` date NOT NULL,
  `Original_time` time NOT NULL,
  `Time_diff` time NOT NULL default '00:00:00',
  `StO2_15mm` int(3) default NULL,
  `THI_15mm` float(3,1) default NULL,
  `Event_name` int(2) NOT NULL default 99,
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
PRIMARY KEY (`Date_of_experiment`,`Time_of_experiment`),
INDEX (`Pig_ID`),
FOREIGN KEY (`Pig_ID`) REFERENCES `monr_identifier_table` (`Pig_ID`)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE IF NOT EXISTS `monr_arterial_bloodgases_realtime` (

```

```

`Date_of_experiment` date NOT NULL,
`Time_of_experiment` time NOT NULL,
`Original_date` date NOT NULL,
`Original_time` time NOT NULL,
`Time_diff` time NOT NULL default '00:00:00',
`Calcium` float(3,2) default NULL,
`Glucose` int(3) default NULL,
`Potassium` float(3,1) default NULL,
`Lactate` float(3,1) default NULL,
`Sodium` int(3) default NULL,
`Hb` float(3,1) default NULL,
`Arterial_pH` float(4,2) default NULL,
`Arterial_PC02` int(3) default NULL,
`Arterial_PO2` int(3) default NULL,
`Arterial_O2_sat` int(3) default NULL,
`Event_name` int(2) NOT NULL default 99,
`Pig_ID` varchar(6) collate utf8_bin NOT NULL,
PRIMARY KEY (`Date_of_experiment`,`Time_of_experiment`),
INDEX (`Pig_ID`),
FOREIGN KEY (`Pig_ID`) REFERENCES `monr_identifier_table` (`Pig_ID`)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE IF NOT EXISTS `monr_venous_bloodgases_realtime` (
  `Date_of_experiment` date NOT NULL,
  `Time_of_experiment` time NOT NULL,
  `Original_date` date NOT NULL,
  `Original_time` time NOT NULL,
  `Time_diff` time NOT NULL default '00:00:00',
  `Venous_PO2` int(3) default NULL,
  `Venous_O2_sat` int(3) default NULL,
  `Event_name` int(2) NOT NULL default 99,
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  PRIMARY KEY (`Date_of_experiment`,`Time_of_experiment`),
  INDEX (`Pig_ID`),
  FOREIGN KEY (`Pig_ID`) REFERENCES `monr_identifier_table` (`Pig_ID`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE IF NOT EXISTS `monr_do2vo2ero2_realtime` (
  `Date_of_experiment` date NOT NULL,
  `Time_of_experiment` time NOT NULL,
  `Original_date` date NOT NULL,
  `Original_time` time NOT NULL,
  `Time_diff` time NOT NULL default '00:00:00',
  `DO2` float(5,3) default NULL,
  `VO2` float(5,3) default NULL,
  `ERO2` float(5,3) default NULL,
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  PRIMARY KEY (`Date_of_experiment`,`Time_of_experiment`),
  INDEX (`Pig_ID`),
  FOREIGN KEY (`Pig_ID`) REFERENCES `monr_identifier_table` (`Pig_ID`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE IF NOT EXISTS `monr_event_time_info` (
  `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
  `Event_name` int(2) NOT NULL,

```

```

    `Date_of_experiment` date default NULL,
    `Time_of_experiment` time default NULL,
    PRIMARY KEY (`Pig_ID`,`Event_name`),
    INDEX (`Pig_ID`),
    FOREIGN KEY (`Pig_ID`) REFERENCES `monr_identifier_table` (`Pig_ID`)
    ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE IF NOT EXISTS `monr_time_modified_realtime_data` (
    `Date_collected` date NOT NULL,
    `Time_collected` time NOT NULL,
    `Table_name` varchar(100) NOT NULL,
    `Pig_ID` varchar(6) collate utf8_bin NOT NULL,
    `Date_modified_to` date default NULL,
    `Time_modified_to` time default NULL,
    `Time_difference` varchar(50) default NULL,
    PRIMARY KEY (`Date_collected`,`Time_collected`,`Table_name`,`Pig_ID`),
    INDEX (`Pig_ID`),
    FOREIGN KEY (`Pig_ID`) REFERENCES `monr_identifier_table` (`Pig_ID`)
    ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

Appendix C. Manual and Real-time data comparison

SAS code and results for one subject

SAS code

```
options nonumber nodate orientation=portrait;
*libname priyaran mysql user=sfs password='dfdsfdsf*' database=sdfdsfdf
server=dsfsfsdf port=42343;
*proc univariate data=priyaran.monr_binning_muscle;
/*The above code won't work if the column names of the database table
won't conform with SAS naming conventions.
    So, it is better to use proc sql where SAS will allow strange
names*/
ods graphics on;
ods pdf
file='C:\Desktopfiles\Project_related\SAS_code_for_comparision\reg_outp
ut\012510.pdf';
proc format;
    picture dbdate
        other = '%Y-%0m-%0d:%0H:%0M:%0S' (datatype=datetime);
run;

proc sql noprint;
    libname priyaran "C:\Desktopfiles\sastest\";
    select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
    libname priyaran "C:\Desktopfiles\sastest\";
    connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
    create table priyaran.temp as
    select *
        from connection to mysql
        /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
        /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifier_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/
        (select t1.Event_name,t1.Art_Sys as
Art_Sys_realtime,t2.Art_Sys as Art_Sys_manual from
test_arterialpressure_realtime as t1 inner join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
```

```

t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510');
      /*(select
t1.Date_of_experiment,t1.Time_of_experiment,t1.Art_Sys as
Art_Sys_realtime,t2.Art_Sys as Art_Sys_manual from
test_arterialpressure_realtime as t1 left join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');*/
disconnect from mysql;
quit;

proc print data=priyaran.temp;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
  cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=r1 c=blue h=0.5 l=1;
symbol2 v=dot i=r1 c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "Arterial Pressure-Systolic")
minor=(n=4);
axis2 label=("Event Name") minor=none order=(0to 25 by 1) offset=(2,2);
legend1 label=none value=(j=left "Arterial Systolic Manual" j=right
"Arterial Systolic Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF ARTERIAL SYSTOLIC
OF REAL-TIME AND MANUAL DATA ONLY EVENTS";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp;
  plot (Art_Sys_manual Art_Sys_realtime)*Event_name/overlay
vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp;
  paired Art_Sys_manual*Art_Sys_realtime;
run;

proc sql noprint;
  libname priyaran "C:\Desktopfiles\sastest\";
  select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
  libname priyaran "C:\Desktopfiles\sastest\";
  connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
  create table priyaran.temp as
  select *
  from connection to mysql
      /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
      /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da

```

```

te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifier_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/

/*(select t1.Event_name,t1.Art_Sys as
Art_Sys_realtime,t2.Art_Sys as Art_Sys_manual from
test_arterialpressure_realtime as t1 inner join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510'); */
(select
t1.Date_of_experiment,t1.Time_of_experiment,t1.Art_Sys as
Art_Sys_realtime,t2.Art_Sys as Art_Sys_manual from
test_arterialpressure_realtime as t1 left join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');
disconnect from mysql;
quit;

data priyaran.temp1;
set priyaran.temp;
Date_and_Time=dhms(Date_of_experiment,hour(Time_of_experiment),mi
nute(Time_of_experiment),second(Time_of_experiment));
format Date_and_Time dbdate.;
*put Date_and_Time=dbdate.;

run;

proc print data=priyaran.temp1;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=rl c=blue h=0.5 l=1;
symbol2 v=dot i=rl c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "Arterial Pressure-Systolic")
minor=(n=4);
axis2 label=("Date and Time") minor=none offset=(2,2);
legend1 label=none value=(j=left "Arterial Systolic Manual" j=right
"Arterial Systolic Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF ARTERIAL SYSTOLIC
OF REAL-TIME AND MANUAL DATA";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp1;
plot (Art_Sys_manual Art_Sys_realtime)*Date_and_Time/overlay
vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 grid;
run;

```

```

proc ttest data=priyaran.temp1;
    paired Art_Sys_manual*Art_Sys_realtime;
run;
/*#####
#####*/
proc sql noprint;
    libname priyaran "C:\Desktopfiles\sastest\";
    select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
    libname priyaran "C:\Desktopfiles\sastest\";
    connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
    create table priyaran.temp as
        select *
            from connection to mysql
                /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
                /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifier_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/
            (select t1.Event_name,t1.Art_Dia as
Art_Dia_realtime,t2.Art_Dia as Art_Dia_manual from
test_arterialpressure_realtime as t1 inner join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510');
                /*(select
t1.Date_of_experiment,t1.Time_of_experiment,t1.Art_Dia as
Art_Dia_realtime,t2.Art_Dia as Art_Dia_manual from
test_arterialpressure_realtime as t1 left join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');*/
        disconnect from mysql;
quit;

proc print data=priyaran.temp;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
    cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;

```



```

symbol1 v=dot i=rl c=blue h=0.5 l=1;
symbol2 v=dot i=rl c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "Arterial Pressure-Diastolic")
minor=(n=4);
axis2 label=("Event Name") minor=none order=(0to 25 by 1) offset=(2,2);
legend1 label=none value=(j=left "Arterial Diastolic Manual" j=right
"Arterial Diastolic Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF ARTERIAL DIASTOLIC
OF REAL-TIME AND MANUAL DATA ONLY EVENTS";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp;
    plot (Art_Dia_manual Art_Dia_realtime)*Event_name/overlay
vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp;
    paired Art_Dia_manual*Art_Dia_realtime;
run;

proc sql noprint;
    libname priyaran "C:\Desktopfiles\sastest\";
    select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
    libname priyaran "C:\Desktopfiles\sastest\";
    connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
    create table priyaran.temp as
        select *
            from connection to mysql
                /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
                /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifer_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/
                /*(select t1.Event_name,t1.Art_Dia as
Art_Dia_realtime,t2.Art_Dia as Art_Dia_manual from
test_arterialpressure_realtime as t1 inner join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510'); */
                (select
t1.Date_of_experiment,t1.Time_of_experiment,t1.Art_Dia as
Art_Dia_realtime,t2.Art_Dia as Art_Dia_manual from
test_arterialpressure_realtime as t1 left join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and

```

```

t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');
disconnect from mysql;
quit;

data priyaran.temp1;
set priyaran.temp;
Date_and_Time=dhms(Date_of_experiment, hour(Time_of_experiment), minute(Time_of_experiment), second(Time_of_experiment));
format Date_and_Time dbdate.;
*put Date_and_Time=dbdate.;
run;

proc print data=priyaran.temp1;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=r1 c=blue h=0.5 l=1;
symbol2 v=dot i=r1 c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "Arterial Pressure-Diastolic")
minor=(n=4);
axis2 label=("Date and Time") minor=none offset=(2,2);
legend1 label=none value=(j=left "Arterial Diastolic Manual" j=right
"Arterial Diastolic Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF ARTERIAL DIASTOLIC
OF REAL-TIME AND MANUAL DATA";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp1;
plot (Art_Dia_manual Art_Dia_realtime)*Date_and_Time/overlay
vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp1;
paired Art_Dia_manual*Art_Dia_realtime;
run;

/*#####
#####*/
proc sql noprint;
libname priyaran "C:\Desktopfiles\sastest\";
select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
libname priyaran "C:\Desktopfiles\sastest\";
connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
create table priyaran.temp as
select *
from connection to mysql

```

```

        /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
        /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifer_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/

(select t1.Event_name,t1.MeanArtPres as
MeanArtPres_realtime,t2.MeanArtPres as MeanArtPres_manual from
test_arterialpressure_realtime as t1 inner join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510');
        /*(select
t1.Date_of_experiment,t1.Time_of_experiment,t1.MeanArtPres as
MeanArtPres_realtime,t2.MeanArtPres as MeanArtPres_manual from
test_arterialpressure_realtime as t1 left join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');*/
disconnect from mysql;
quit;

proc print data=priyaran.temp;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
    cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=r1 c=blue h=0.5 l=1;
symbol2 v=dot i=r1 c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "Mean Arterial Pressure") minor=(n=4);
axis2 label=("Event Name") minor=none order=(0to 25 by 1) offset=(2,2);
legend1 label=none value=(j=left "Mean Arterial Pressure Manual"
j=right "Mean Arterial Pressure Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF MEAN ARTERIAL
PRESSURE OF REAL-TIME AND MANUAL DATA ONLY EVENTS";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp;
    plot (MeanArtPres_manual MeanArtPres_realtime)*Event_name/overlay
vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp;
    paired MeanArtPres_manual*MeanArtPres_realtime;
run;

```

```

proc sql noprint;
    libname priyaran "C:\Desktopfiles\sastest\";
    select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
    libname priyaran "C:\Desktopfiles\sastest\";
    connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
    create table priyaran.temp as
        select *
            from connection to mysql
                /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
                /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifer_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/
                /*(select t1.Event_name,t1.MeanArtPres as
MeanArtPres_realtime,t2.MeanArtPres as MeanArtPres_manual from
test_arterialpressure_realtime as t1 inner join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510'); */
                (select
t1.Date_of_experiment,t1.Time_of_experiment,t1.MeanArtPres as
MeanArtPres_realtime,t2.MeanArtPres as MeanArtPres_manual from
test_arterialpressure_realtime as t1 left join
test_vitals_manual_10_min as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');
            disconnect from mysql;
quit;

data priyaran.temp1;
    set priyaran.temp;
    Date_and_Time=dhms(Date_of_experiment,hour(Time_of_experiment),mi
nute(Time_of_experiment),second(Time_of_experiment));
    format Date_and_Time dbdate.;
    *put Date_and_Time=dbdate.;
run;

proc print data=priyaran.temp1;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in

```

```

cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=r1 c=blue h=0.5 l=1;
symbol2 v=dot i=r1 c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "Mean Arterial Pressure") minor=(n=4);
axis2 label=("Date and Time") minor=none offset=(2,2);
legend1 label=none value=(j=left "Mean Arterial Pressure Manual"
j=right "Mean Arterial Pressure Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF MEAN ARTERIAL
PRESSURE OF REAL-TIME AND MANUAL DATA";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp1;
    plot (MeanArtPres_manual
MeanArtPres_realtime)*Date_and_Time/overlay vaxis=axis1 haxis=axis2
caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp1;
    paired MeanArtPres_manual*MeanArtPres_realtime;
run;

/*#####
#####*/
proc sql noprint;
    libname priyaran "C:\Desktopfiles\sastest\";
    select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
    libname priyaran "C:\Desktopfiles\sastest\";
    connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
    create table priyaran.temp as
    select *
        from connection to mysql
            /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
            /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifier_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/
        (select t1.Event_name,t1.StO2_15mm as
StO2_15mm_realtime,t2.StO2_15mm as StO2_15mm_manual from
test_sto2thi_realtime as t1 inner join test_vitals_manual_10_min as t2
on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510');
            /*(select
t1.Date_of_experiment,t1.Time_of_experiment,t1.StO2_15mm as
StO2_15mm_realtime,t2.StO2_15mm as StO2_15mm_manual from
test_sto2thi_realtime as t1 left join test_vitals_manual_10_min as t2

```

```

on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');*/
disconnect from mysql;
quit;

proc print data=priyaran.temp;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
  cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=r1 c=blue h=0.5 l=1;
symbol2 v=dot i=r1 c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "StO2-15mm") minor=(n=4);
axis2 label=("Event Name") minor=none order=(0to 25 by 1) offset=(2,2);
legend1 label=none value=(j=left "StO2-15mm Manual" j=right "StO2-15mm
Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF STO2-15MM OF REAL-
TIME AND MANUAL DATA ONLY EVENTS";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp;
  plot (StO2_15mm_manual StO2_15mm_realtime)*Event_name/overlay
vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp;
  paired StO2_15mm_manual*StO2_15mm_realtime;
run;

proc sql noprint;
  libname priyaran "C:\Desktopfiles\sastest\";
  select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
  libname priyaran "C:\Desktopfiles\sastest\";
  connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
  create table priyaran.temp as
  select *
  from connection to mysql
  /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
  /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifer_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/

```

```

                /*(select t1.Event_name,t1.StO2_15mm as
StO2_15mm_realtime,t2.StO2_15mm as StO2_15mm_manual from
test_sto2thi_realtime as t1 inner join test_vitals_manual_10_min as t2
on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510'); */
                (select
t1.Date_of_experiment,t1.Time_of_experiment,t1.StO2_15mm as
StO2_15mm_realtime,t2.StO2_15mm as StO2_15mm_manual from
test_sto2thi_realtime as t1 left join test_vitals_manual_10_min as t2
on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');
disconnect from mysql;
quit;

data priyaran.temp1;
    set priyaran.temp;
        Date_and_Time=dhms(Date_of_experiment,hour(Time_of_experiment),mi
nute(Time_of_experiment),second(Time_of_experiment));
        format Date_and_Time dbdate.;
        *put Date_and_Time=dbdate.;
run;

proc print data=priyaran.temp1;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
cctx=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
    cback=white cctx=green
fctx="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=rl c=blue h=0.5 l=1;
symbol2 v=dot i=rl c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "StO2-15mm") minor=(n=4);
axis2 label=("Date and Time") minor=none offset=(2,2);
legend1 label=none value=(j=left "StO2-15mm Manual" j=right "StO2-15mm
Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF STO2-15MM OF REAL-
TIME AND MANUAL DATA";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp1;
    plot (StO2_15mm_manual StO2_15mm_realtime)*Date_and_Time/overlay
vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp1;
    paired StO2_15mm_manual*StO2_15mm_realtime;
run;

/*****
*****/
proc sql noprint;
    libname priyaran "C:\Desktopfiles\sastest\";
    select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;

```

```

quit;
proc sql;
  libname priyaran "C:\Desktopfiles\sastest\";
  connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
  create table priyaran.temp as
  select *
  from connection to mysql
  /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
  /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifier_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/

  (select t1.Event_name,t1.THI_15mm as
THI_15mm_realtime,t2.THI_15mm as THI_15mm_manual from
test_sto2thi_realtime as t1 inner join test_vitals_manual_10_min as t2
on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510');
  /*(select
t1.Date_of_experiment,t1.Time_of_experiment,t1.THI_15mm as
THI_15mm_realtime,t2.THI_15mm as THI_15mm_manual from
test_sto2thi_realtime as t1 left join test_vitals_manual_10_min as t2
on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');*/
disconnect from mysql;
quit;

proc print data=priyaran.temp;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
  cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=rl c=blue h=0.5 l=1;
symbol2 v=dot i=rl c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "THI-15mm") minor=(n=4);
axis2 label=("Event Name") minor=none order=(0to 25 by 1) offset=(2,2);
legend1 label=none value=(j=left "THI-15mm Manual" j=right "THI-15mm
Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF THI-15mm OF REAL-
TIME AND MANUAL DATA ONLY EVENTS";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp;
  plot (THI_15mm_manual THI_15mm_realtime)*Event_name/overlay
vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 grid;

```



```

run;

proc ttest data=priyaran.temp;
    paired THI_15mm_manual*THI_15mm_realtime;
run;

proc sql noprint;
    libname priyaran "C:\Desktopfiles\sastest\";
    select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
    libname priyaran "C:\Desktopfiles\sastest\";
    connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
    create table priyaran.temp as
        select *
            from connection to mysql
                /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
                /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifier_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/
                /*(select t1.Event_name,t1.THI_15mm as
THI_15mm_realtime,t2.THI_15mm as THI_15mm_manual from
test_sto2thi_realtime as t1 inner join test_vitals_manual_10_min as t2
on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510'); */
                (select
t1.Date_of_experiment,t1.Time_of_experiment,t1.THI_15mm as
THI_15mm_realtime,t2.THI_15mm as THI_15mm_manual from
test_sto2thi_realtime as t1 left join test_vitals_manual_10_min as t2
on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');
    disconnect from mysql;
quit;

data priyaran.temp1;
    set priyaran.temp;
    Date_and_Time=dhms(Date_of_experiment,hour(Time_of_experiment),mi
nute(Time_of_experiment),second(Time_of_experiment));
    format Date_and_Time dbdate.;
    *put Date_and_Time=dbdate.;
run;

proc print data=priyaran.temp1;
run cancel;

```

```

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
  cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=r1 c=blue h=0.5 l=1;
symbol2 v=dot i=r1 c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "THI-15mm") minor=(n=4);
axis2 label=("Date and Time") minor=none offset=(2,2);
legend1 label=none value=(j=left "THI-15mm Manual" j=right "THI-15mm
Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF THI-15mm OF REAL-
TIME AND MANUAL DATA";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp1;
  plot (THI_15mm_manual THI_15mm_realtime)*Date_and_Time/overlay
vaxis=axis1 haxis=axis2 caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp1;
  paired THI_15mm_manual*THI_15mm_realtime;
run;

/*#####
#####*/
proc sql noprint;
  libname priyaran "C:\Desktopfiles\sastest\";
  select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
  libname priyaran "C:\Desktopfiles\sastest\";
  connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
  create table priyaran.temp as
  select *
  from connection to mysql
  /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
  /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifier_table` AS `t2` ON
`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/
  (select t1.Event_name,t1.Cardiac_output as
Cardiac_output_realtime,t2.Cardiac_output as Cardiac_output_manual from
test_cardiacoutput_realtime as t1 inner join
test_vitals_manual_one_hour as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510');

```

```

                /*(select
t1.Date_of_experiment,t1.Time_of_experiment,t1.Cardiac_output as
Cardiac_output_realtime,t2.Cardiac_output as Cardiac_output_manual from
test_cardiacoutput_realtime as t1 left join test_vitals_manual_one_hour
as t2 on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');*/
disconnect from mysql;
quit;

proc print data=priyaran.temp;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
c text=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
  cback=white ctext=green
f title="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=rl c=blue h=0.5 l=1;
symbol2 v=dot i=rl c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "Cardiac Output") minor=(n=4);
axis2 label=("Event Name") minor=none order=(0 to 25 by 1) offset=(2,2);
legend1 label=none value=(j=left "Cardiac Output Manual" j=right
"Cardiac Output Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF CARDIAC OUTPUT OF
REAL-TIME AND MANUAL DATA ONLY EVENTS";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp;
  plot (Cardiac_output_manual
Cardiac_output_realtime)*Event_name/overlay vaxis=axis1 haxis=axis2
caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp;
  paired Cardiac_output_manual*Cardiac_output_realtime;
run;

proc sql noprint;
  libname priyaran "C:\Desktopfiles\sastest\";
  select distinct users,password,server,database,port
into:userid,:password,:server,:database,:port from priyaran.login;
quit;
proc sql;
  libname priyaran "C:\Desktopfiles\sastest\";
  connect to mysql (user=&userid password=&password server=&server
database=&database port=&port);
  create table priyaran.temp as
  select *
  from connection to mysql
  /*(select * from test_hnmr_muscle where Event_name=2 and
Normalization='Wet_tissue'); */
  /*(SELECT
`t1`.*,`t2`.Weight,`t2`.Pig_group,`t2`.Start_date,`t2`.Died,`t2`.End_da
te,`t2`.Time_of_death,`t2`.Cause_of_death,`t2`.OR_fluids,`t2`.LR_given,
`t2`.Blood_given,`t2`.Blood_removed,`t2`.Total_volume_given FROM
`test_hnmr_muscle` AS `t1` LEFT JOIN `test_identifer_table` AS `t2` ON

```

```

`t1`.Pig_ID=`t2`.Pig_ID WHERE `t1`.Normalization='Wet_tissue' AND
`t1`.Event_name='2' AND (`t2`.Pig_group='2' OR `t2`.Pig_group='1')
ORDER BY `t2`.Start_date,CASE `t1`.Event_name WHEN '1' THEN 1 WHEN '2'
THEN 2 WHEN '5' THEN 3 WHEN '11' THEN 4 WHEN '23' THEN 5 WHEN '25' THEN
6 END);*/

/*(select t1.Event_name,t1.Cardiac_output as
Cardiac_output_realtime,t2.Cardiac_output as Cardiac_output_manual from
test_cardiacoutput_realtime as t1 inner join
test_vitals_manual_one_hour as t2 on
t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Event_name!=99 and t1.Pig_ID='012510'); */
(select
t1.Date_of_experiment,t1.Time_of_experiment,t1.Cardiac_output as
Cardiac_output_realtime,t2.Cardiac_output as Cardiac_output_manual from
test_cardiacoutput_realtime as t1 left join test_vitals_manual_one_hour
as t2 on t1.Date_of_experiment=t2.Date_of_experiment and
t1.Time_of_experiment=t2.Time_of_experiment and t1.Pig_ID=t2.Pig_ID
where t1.Pig_ID='012510');
disconnect from mysql;
quit;

data priyaran.temp1;
set priyaran.temp;
Date_and_Time=dhms(Date_of_experiment,hour(Time_of_experiment),mi
nute(Time_of_experiment),second(Time_of_experiment));
format Date_and_Time dbdate.;
*put Date_and_Time=dbdate.;

run;

proc print data=priyaran.temp1;
run cancel;

*goptions reset=all autosize=off ftext='Arial' htext=1 gunit=pct
ctext=green;
goptions reset=global gunit=pct hsize=10 in vsize=8 in hsize=10 in
vsize=8 in
cback=white ctext=green
ftitle="TimesRomanBold" ftext="TimesRoman" htitle=3 htext=2;
symbol1 v=dot i=r1 c=blue h=0.5 l=1;
symbol2 v=dot i=r1 c=brown h=0.5 l=3;
axis1 label=(angle=90 rotate=0 "Cardiac Output") minor=(n=4);
axis2 label=("Date and Time") minor=none offset=(2,2);
legend1 label=none value=(j=left "Cardiac Output Manual" j=right
"Cardiac Output Real-time") shape=line(10);
title h=2 f='Arial/bo' "Subject 012510 COMPARISON OF CARDIAC OUTPUT OF
REAL-TIME AND MANUAL DATA";
*footnote j=right "Source: Health-United States-2003";
proc gplot data=priyaran.temp1;
plot (Cardiac_output_manual
Cardiac_output_realtime)*Date_and_Time/overlay vaxis=axis1 haxis=axis2
caxis=blue legend=legend1 grid;
run;

proc ttest data=priyaran.temp1;
paired Cardiac_output_manual*Cardiac_output_realtime;
run;

```

```

/*
proc sort data=priyaran.temp;
    by Pig_group;
run cancel;
proc contents data=priyaran.temp;
run cancel;
proc ttest data=priyaran.temp;
    class Pig_group;
run cancel;

proc nparlway data = priyaran.temp;
    class Pig_group;
    var Lactate;
run cancel;
*/

/*
proc univariate data=priyaran.stattest_hnmr_muscle;
run;
*/

/*
The following example connects to MySQL and sends it two EXECUTE
statements to
process:
*/

ods _all_ close;
ods graphics off;

```

SAS output (removed unwanted output)

Subject 012510 COMPARISON OF CARDIAC OUTPUT OF REAL-TIME AND MANUAL DATA
--

The TTEST Procedure

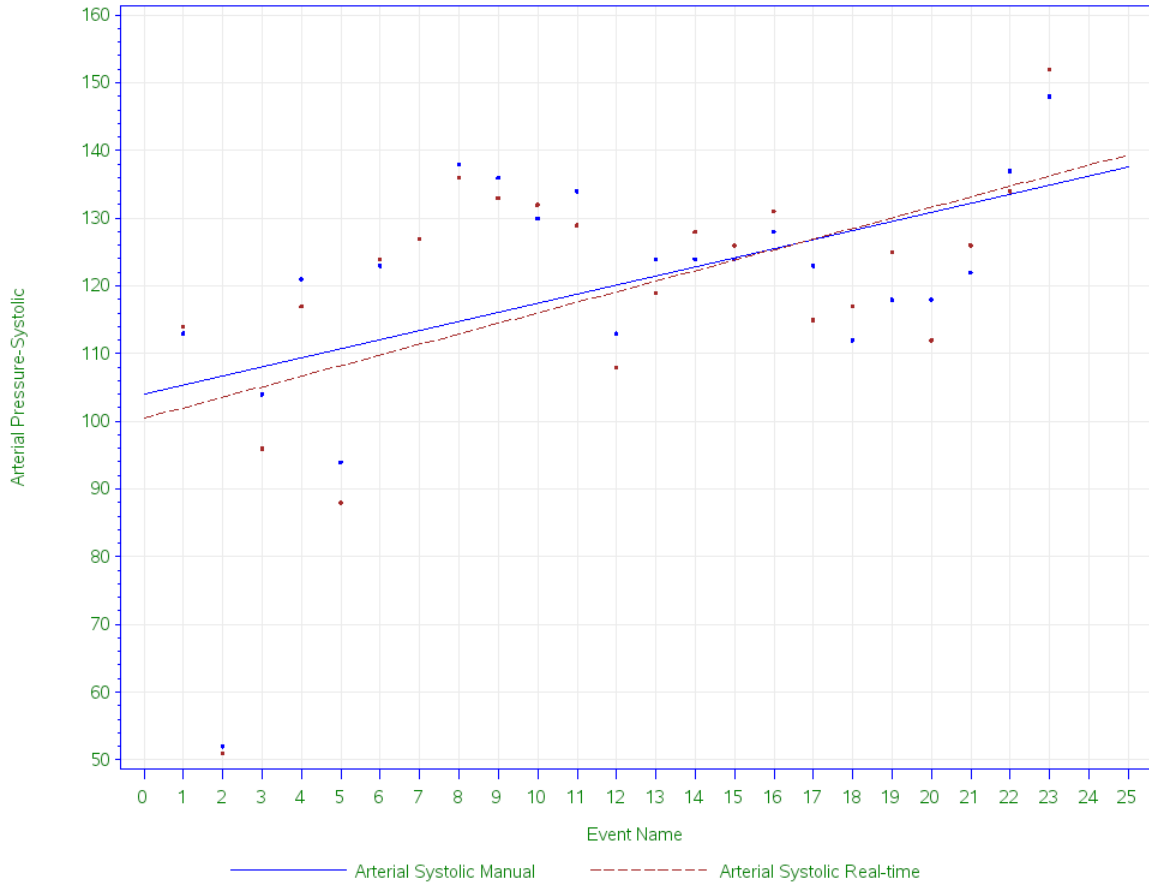
Difference: Art_Sys_manual - Art_Sys_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
23	1.0000	4.4313	0.9240	-7.0000	8.0000

Mean	95% CL Mean	Std Dev	95% CL Std Dev
1.0000	-0.9162	2.9162	4.4313
			3.4271
			6.2718

DF	t Value	Pr > t
22	1.08	0.2909

Subject 012510 COMPARISON OF ARTERIAL SYSTOLIC OF REAL-TIME AND MANUAL DATA ONLY EVENTS



Subject 012510 COMPARISON OF ARTERIAL SYSTOLIC OF REAL-TIME AND MANUAL DATA ONLY EVENTS

The TTEST Procedure

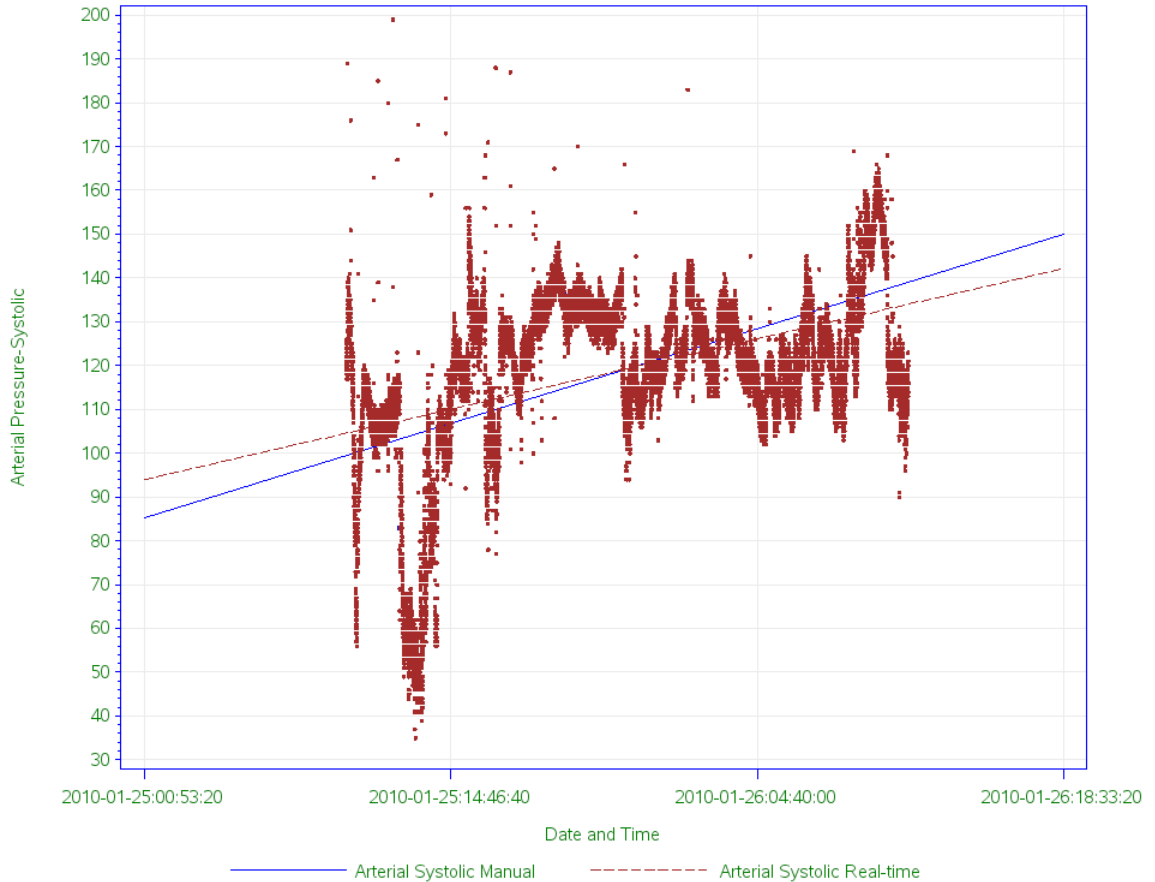
Difference: Art_Sys_manual - Art_Sys_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
47	-0.2766	6.5763	0.9593	-31.0000	13.0000

Mean	95% CL Mean	Std Dev	95% CL Std Dev	
-0.2766	-2.2075	1.6543	6.5763	
			5.4648	8.2598

DF	t Value	Pr > t
46	-0.29	0.7744

Subject 012510 COMPARISON OF ARTERIAL SYSTOLIC OF REAL-TIME AND MANUAL DATA



Subject 012510 COMPARISON OF ARTERIAL SYSTOLIC OF REAL-TIME AND MANUAL DATA

The TTEST Procedure

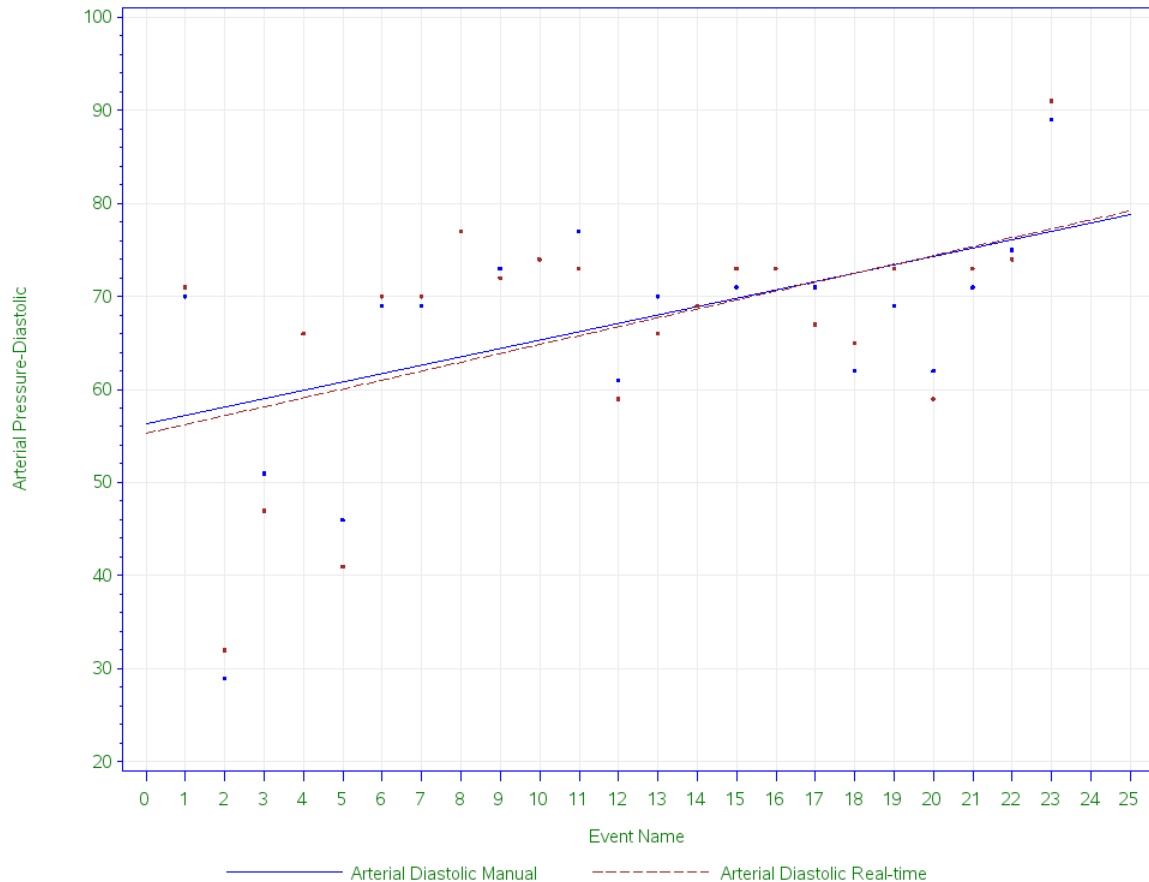
Difference: Art_Dia_manual - Art_Dia_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
23	0.3913	2.6066	0.5435	-4.0000	5.0000

Mean	95% CL Mean	Std Dev	95% CL Std Dev
0.3913	-0.7359	1.5185	2.6066

DF	t Value	Pr > t
22	0.72	0.4791

Subject 012510 COMPARISON OF ARTERIAL DIASTOLIC OF REAL-TIME AND MANUAL DATA ONLY EVENTS



Subject 012510 COMPARISON OF ARTERIAL DIASTOLIC OF REAL-TIME AND MANUAL DATA ONLY EVENTS

The TTEST Procedure

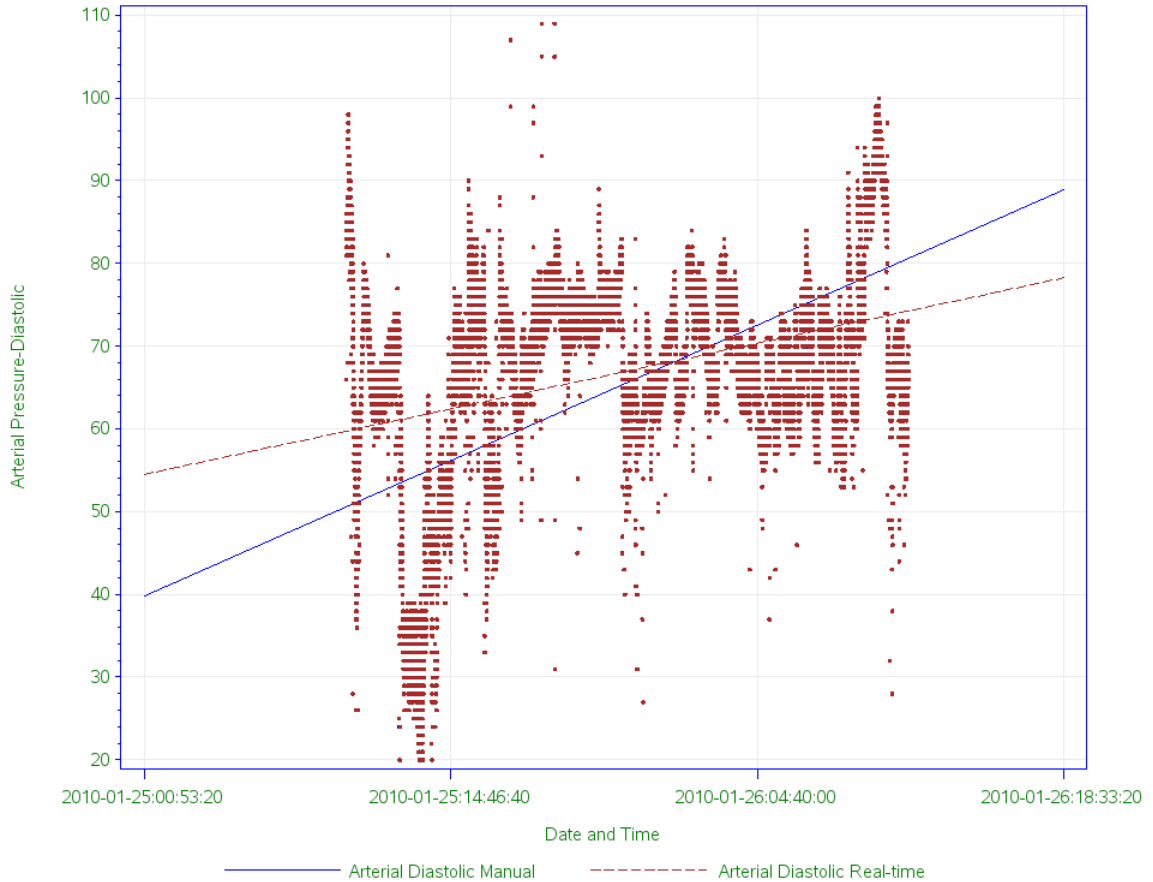
Difference: Art_Dia_manual - Art_Dia_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
47	-0.5957	7.6234	1.1120	-47.0000	8.0000

Mean	95% CL Mean	Std Dev	95% CL Std Dev
-0.5957	-2.8340	1.6426	7.6234

DF	t Value	Pr > t
46	-0.54	0.5947

Subject 012510 COMPARISON OF ARTERIAL DIASTOLIC OF REAL-TIME AND MANUAL DATA



Subject 012510 COMPARISON OF ARTERIAL DIASTOLIC OF REAL-TIME AND MANUAL DATA

The TTEST Procedure

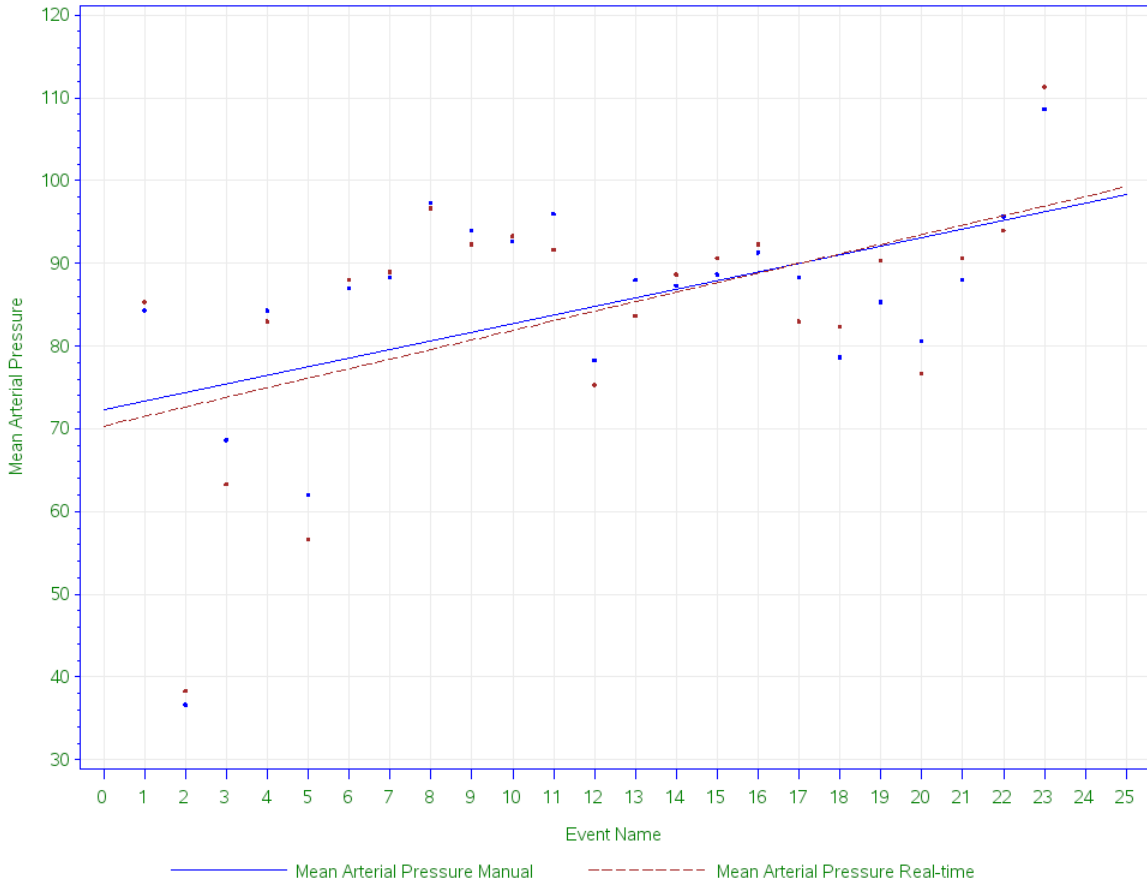
Difference: MeanArtPres_manual - MeanArtPres_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
23	0.5943	3.1081	0.6481	-5.0000	5.3400

Mean	95% CL Mean	Std Dev	95% CL Std Dev
0.5943	-0.7497	1.9384	3.1081

DF	t Value	Pr > t
22	0.92	0.3690

Subject 012510 COMPARISON OF MEAN ARTERIAL PRESSURE OF REAL-TIME AND MANUAL DATA ONLY EVENTS



Subject 012510 COMPARISON OF MEAN ARTERIAL PRESSURE OF REAL-TIME AND MANUAL DATA ONLY EVENTS

The TTEST Procedure

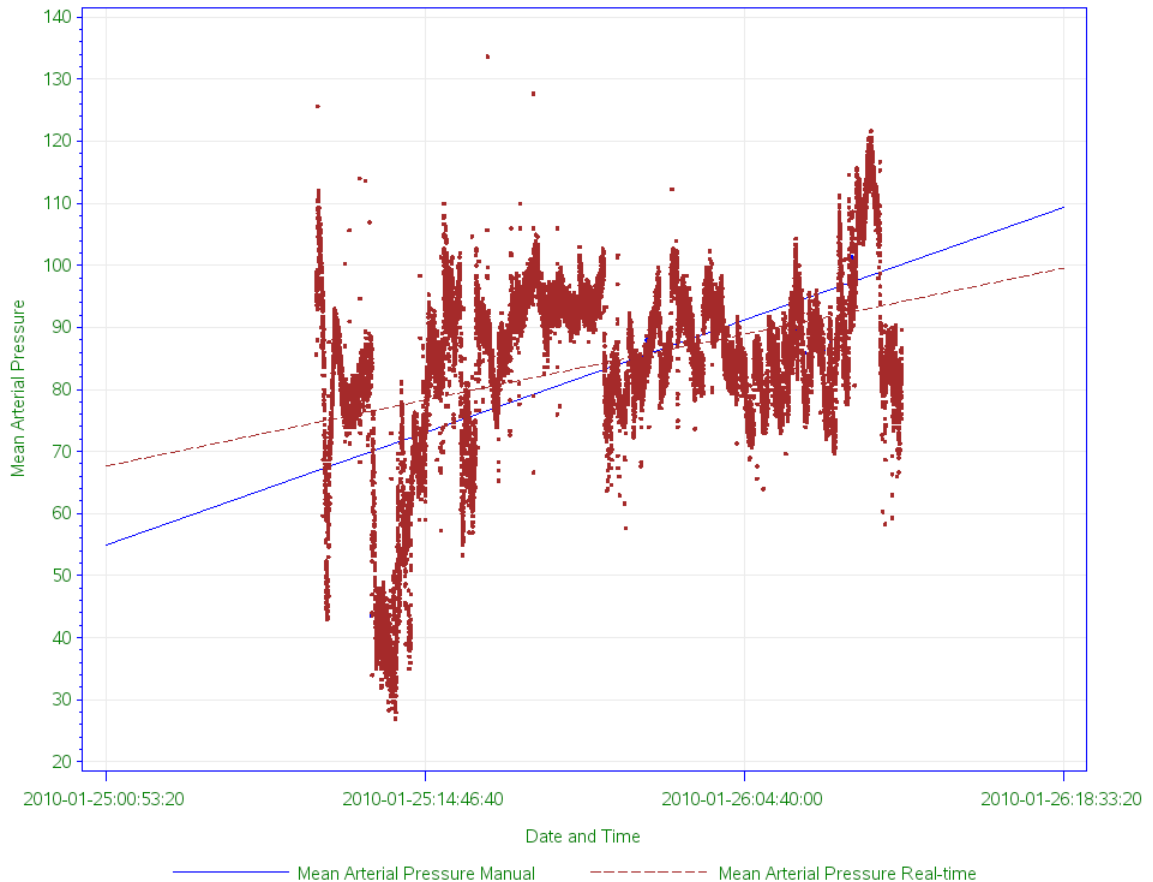
Difference: MeanArtPres_manual - MeanArtPres_realtme

N	Mean	Std Dev	Std Err	Minimum	Maximum
47	-0.4894	7.0845	1.0334	-41.6600	8.0000

Mean	95% CL Mean	Std Dev	95% CL Std Dev
-0.4894	-2.5695	1.5907	7.0845

DF	t Value	Pr > t
46	-0.47	0.6381

Subject 012510 COMPARISON OF MEAN ARTERIAL PRESSURE OF REAL-TIME AND MANUAL DATA



Subject 012510 COMPARISON OF MEAN ARTERIAL PRESSURE OF REAL-TIME AND MANUAL DATA

The TTEST Procedure

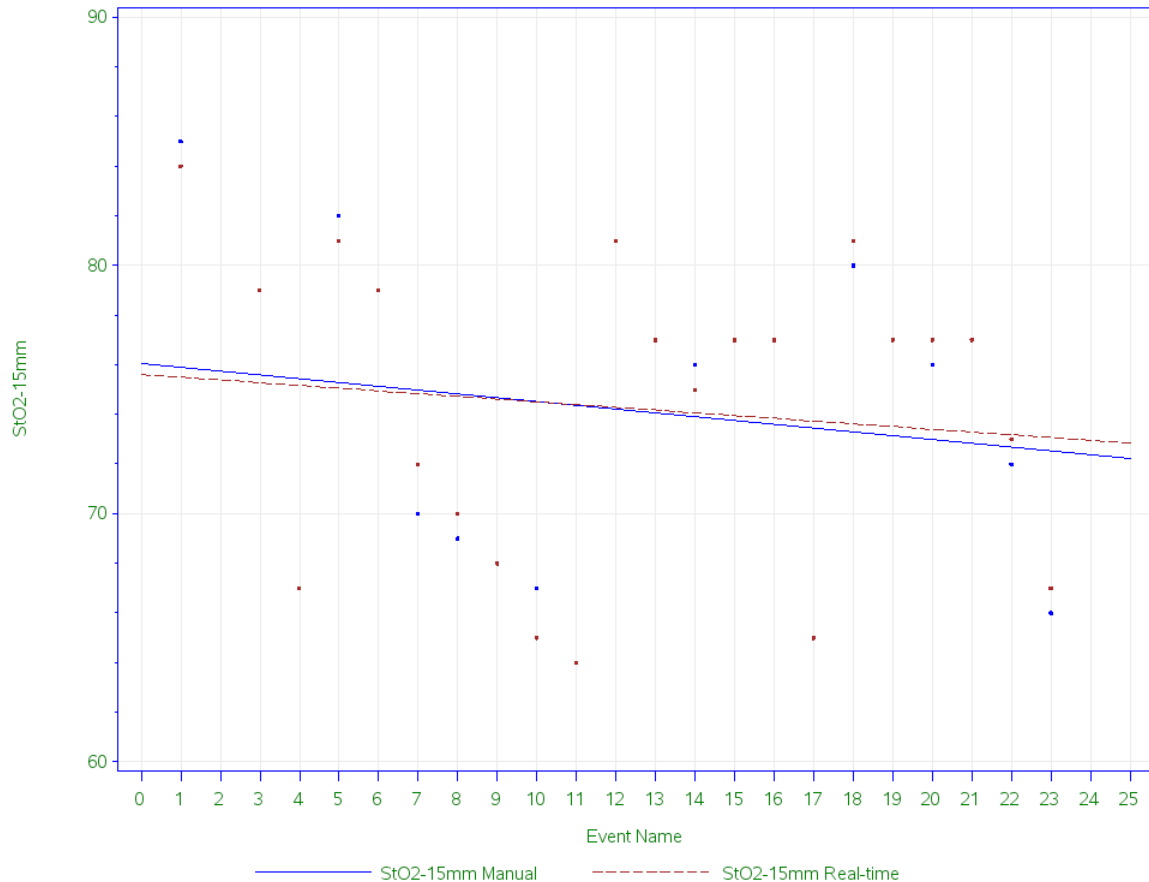
Difference: StO2_15mm_manual - StO2_15mm_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
22	-0.0909	0.8679	0.1850	-2.0000	2.0000

Mean	95% CL Mean	Std Dev	95% CL Std Dev
-0.0909	-0.4757	0.2939	0.6677

DF	t Value	Pr > t
21	-0.49	0.6283

Subject 012510 COMPARISON OF STO2-15MM OF REAL-TIME AND MANUAL DATA ONLY EVENTS



Subject 012510 COMPARISON OF STO2-15MM OF REAL-TIME AND MANUAL DATA ONLY EVENTS

The TTEST Procedure

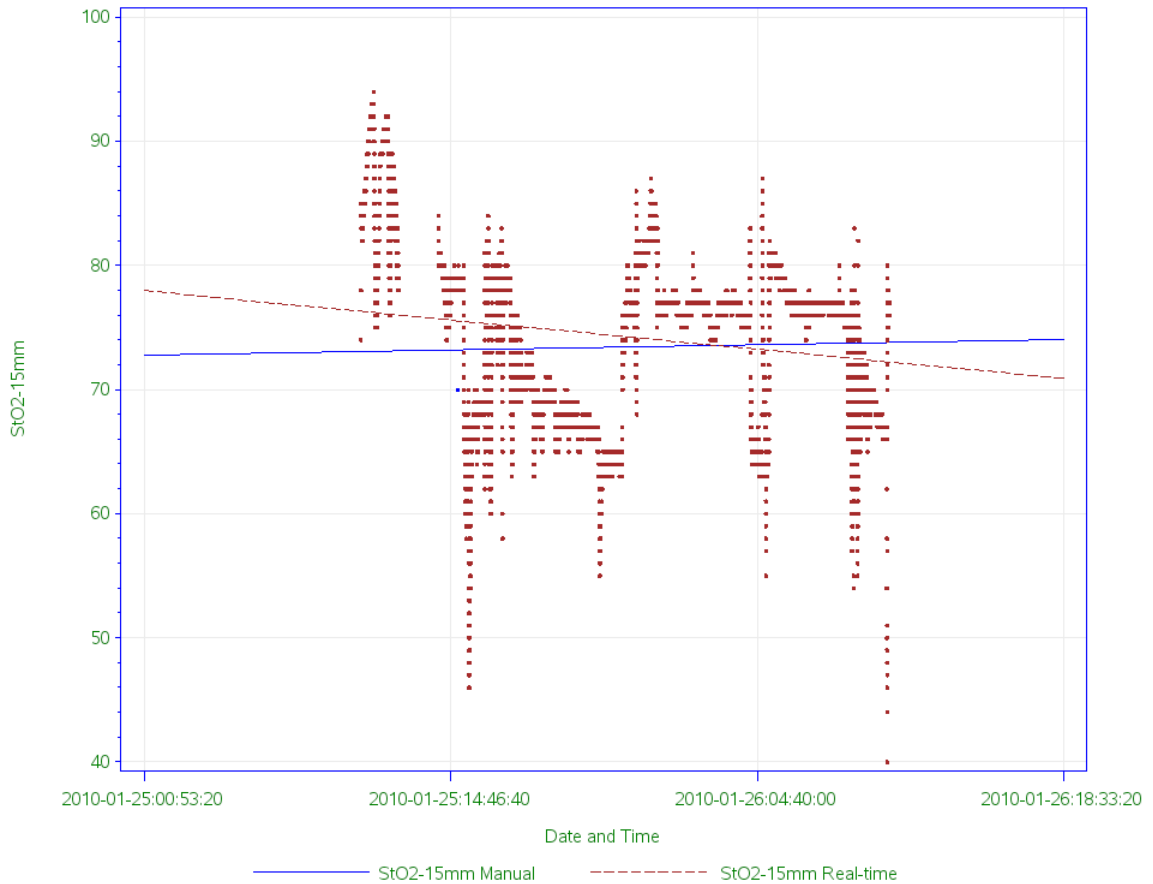
Difference: StO2_15mm_manual - StO2_15mm_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
79	-0.1646	1.3051	0.1468	-9.0000	2.0000

Mean	95% CL Mean	Std Dev	95% CL Std Dev
-0.1646	-0.4569	0.1278	1.3051

DF	t Value	Pr > t
78	-1.12	0.2659

Subject 012510 COMPARISON OF STO2-15MM OF REAL-TIME AND MANUAL DATA



Subject 012510 COMPARISON OF STO2-15MM OF REAL-TIME AND MANUAL DATA

The TTEST Procedure

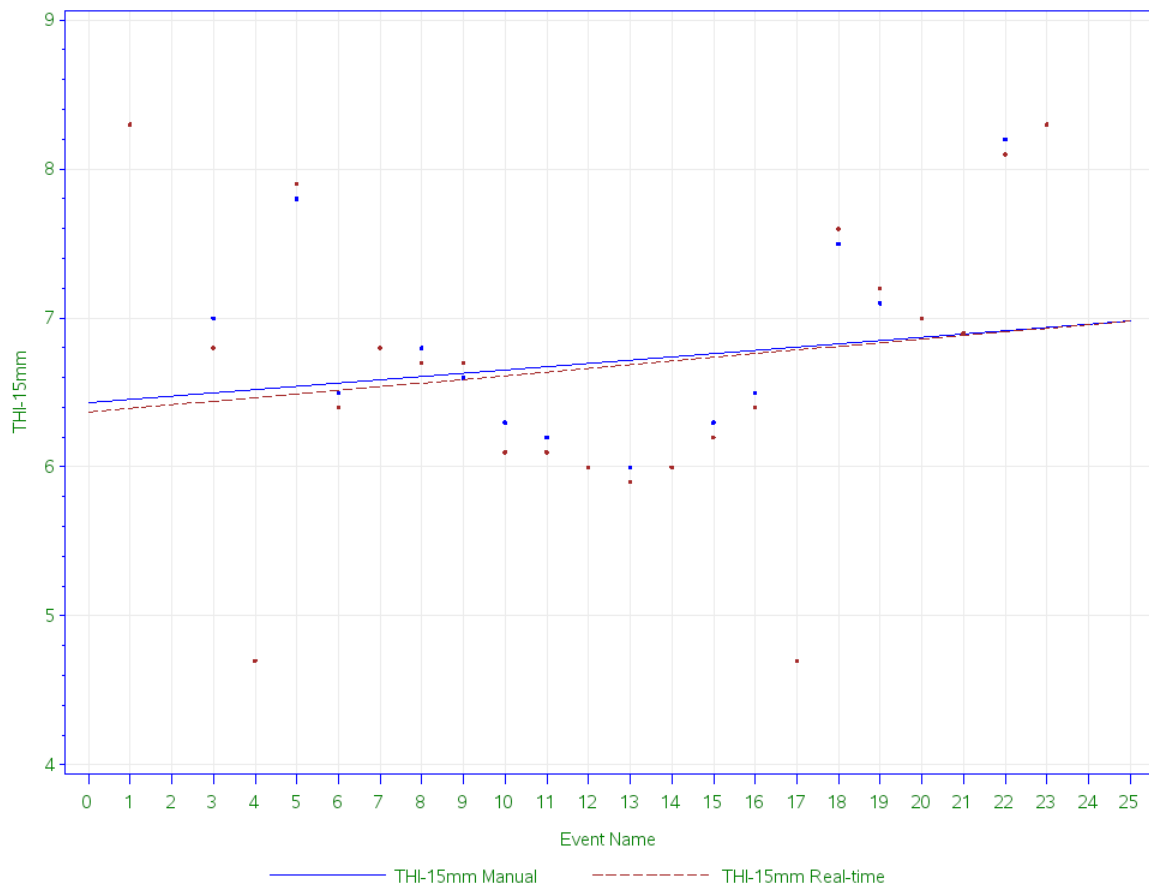
Difference: THI_15mm_manual - THI_15mm_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
22	0.0318	0.0894	0.0191	-0.1000	0.2000

Mean	95% CL Mean	Std Dev	95% CL Std Dev
0.0318	-0.00781	0.0714	0.0894

DF	t Value	Pr > t
21	1.67	0.1098

Subject 012510 COMPARISON OF THI-15mm OF REAL-TIME AND MANUAL DATA ONLY EVENTS



Subject 012510 COMPARISON OF THI-15mm OF REAL-TIME AND MANUAL DATA ONLY EVENTS

The TTEST Procedure

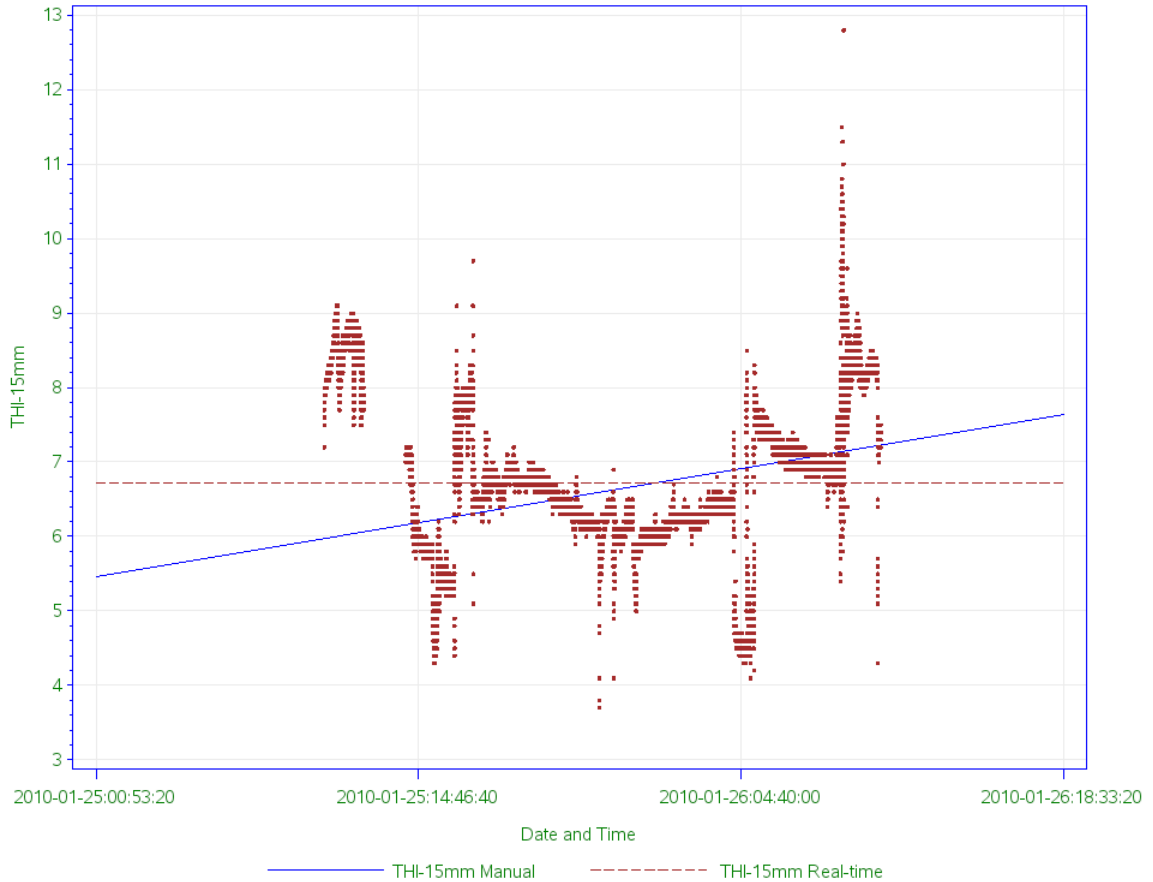
Difference: THI_15mm_manual - THI_15mm_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
79	0.0139	0.0888	0.00999	-0.3000	0.2000

Mean	95% CL Mean	Std Dev	95% CL Std Dev
0.0139	-0.00596	0.0338	0.0888

DF	t Value	Pr > t
78	1.39	0.1672

Subject 012510 COMPARISON OF THI-15mm OF REAL-TIME AND MANUAL DATA



Subject 012510 COMPARISON OF THI-15mm OF REAL-TIME AND MANUAL DATA

The TTEST Procedure

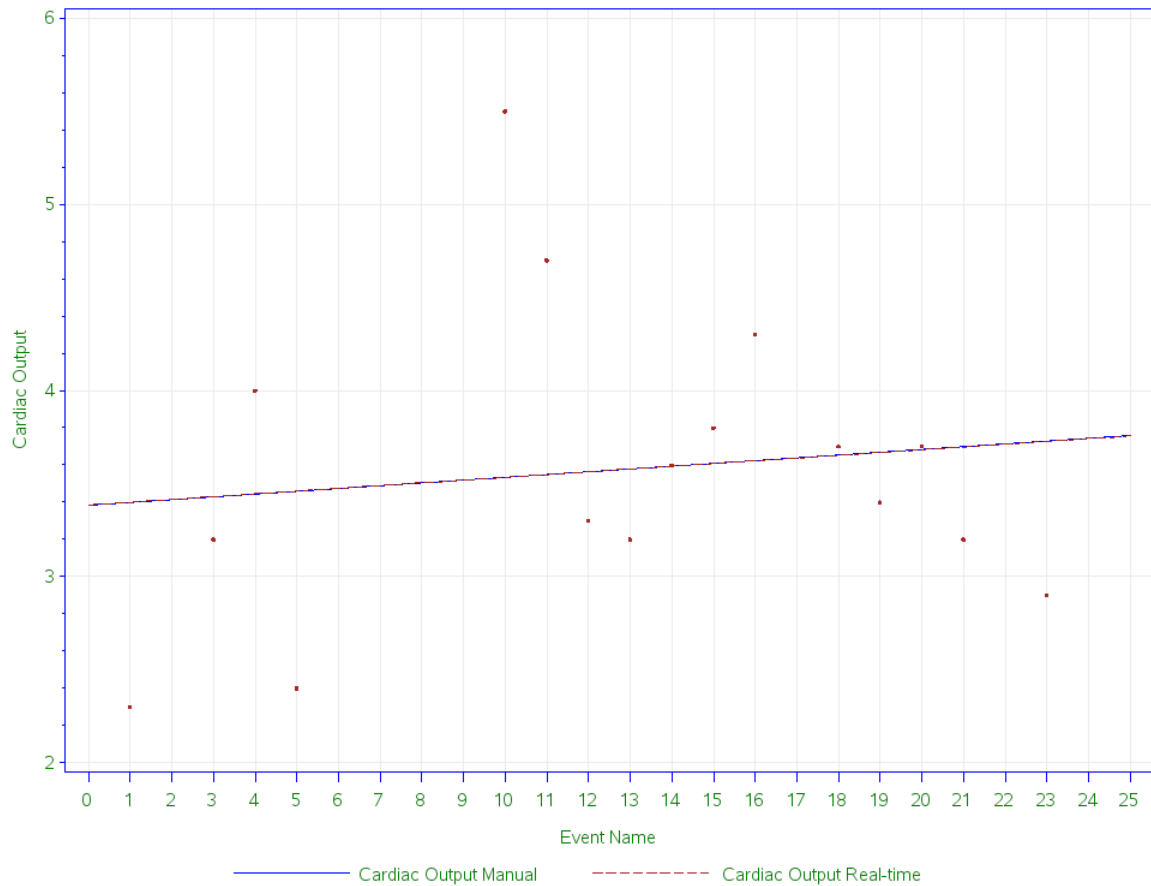
Difference: Cardiac_output_manual - Cardiac_output_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
16	0	0	0	0	0

Mean	95% CL Mean	Std Dev	95% CL Std Dev
0	0	0	0

DF	t Value	Pr > t
15	.	.

Subject 012510 COMPARISON OF CARDIAC OUTPUT OF REAL-TIME AND MANUAL DATA ONLY EVENTS



Subject 012510 COMPARISON OF CARDIAC OUTPUT OF REAL-TIME AND MANUAL DATA ONLY EVENTS

The TTEST Procedure

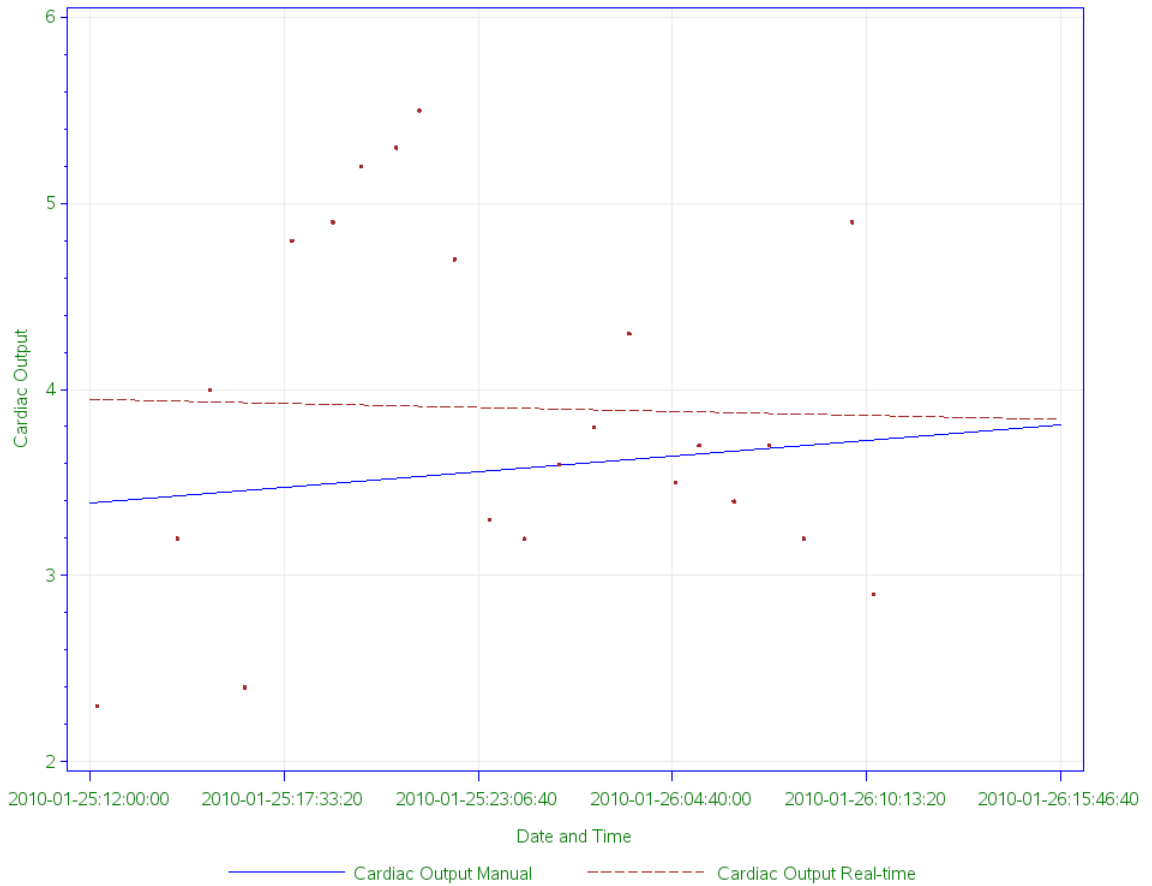
Difference: Cardiac_output_manual - Cardiac_output_realtime

N	Mean	Std Dev	Std Err	Minimum	Maximum
16	0	0	0	0	0

Mean	95% CL Mean	Std Dev	95% CL Std Dev
0	0	0	.

DF	t Value	Pr > t
15	.	.

Subject 012510 COMPARISON OF CARDIAC OUTPUT OF REAL-TIME AND MANUAL DATA



Appendix D. Automated web programming file modification on development machine and synchronization with production machine

Development machine with windows operating system and Apache web server consists of files shown in the Figure 52 useful in programming file modification and synchronization of the code with production machine.

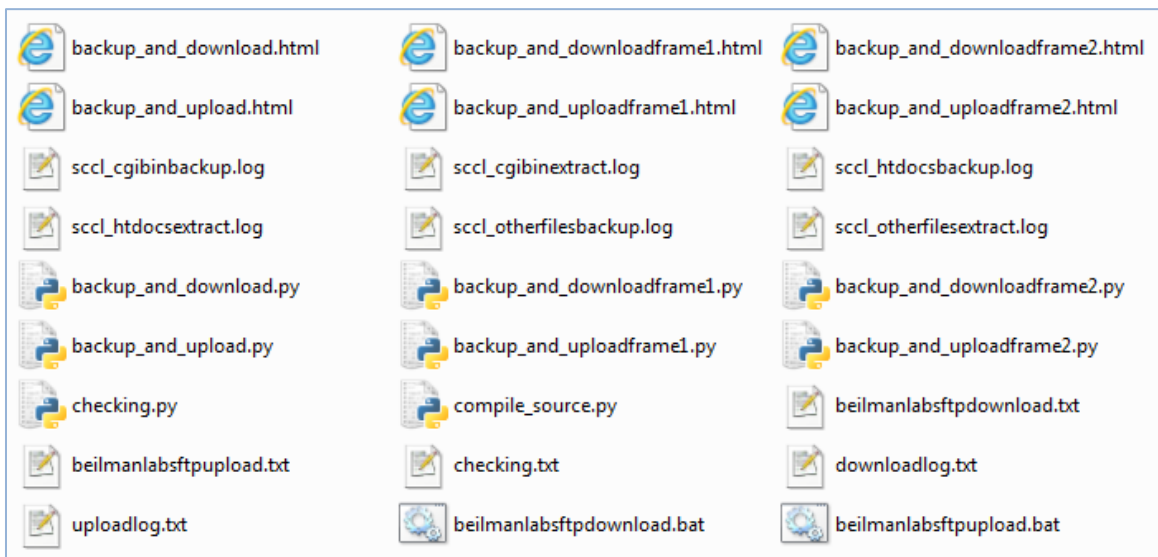


Figure 52 List of files in cgi-bin directory on development machine

URL used for modification of code and file upload to the production server is http://localhost/cgi-bin/backup_and_upload.py. URL used for download of files from the production server is http://localhost/cgi-bin/backup_and_download.py. Contents of log files is not shown below.

Code written for backup_and_download.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Final Code Execution</title>
```

```

    <link rel="stylesheet" type="text/css"
href="/backup_and_download/stylesheets/backup_and_download.css" media="screen" />
    <link rel="icon" type="image/x-icon" href="/SCCL_test/images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon"
href="/SCCL_test/images/favicon.ico"/>
    <script language="javascript" type="text/javascript"
src="/backup_and_download/js/backup_and_download.js"></script>
</head>
<frameset rows="20%,80%" frameborder="0" framespacing="0">
    <frame src="backup_and_downloadframe1.py" frameborder="0" marginwidth="0"
marginheight="0" noresize="noresize" scrolling="no"/>
    <frame src="backup_and_downloadframe2.py" frameborder="0" marginwidth="0"
marginheight="0" noresize="noresize" scrolling="auto" />
</frameset>
</html>

```

Code written for backup_and_downloadframe1.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Data View</title>
    <link rel="stylesheet" type="text/css"
href="/backup_and_download/stylesheets/backup_and_download.css" media="screen" />
</head>
<body>
    <table id="entirepage">
        <tbody>
            <tr>
                <td class="tableholder">
                    <table>
                        <tbody>
                            <tr>
                                <td class="testwebsite">&nbsp;</td>
                                <td class="logout">&nbsp;</td>
                            </tr>
                        </tbody>
                    </table>
                </td>
            </tr>
            <tr>
                <td class="tableholder">
                    <table id="banner" summary="This table defines the banner">
                        <tbody>
                            <tr>
                                <td id="ubanner">
                                    &nbsp;</td>
                                <td class="smallcaption">
                                    SCCL <span style="font-size:16px">Surgical
Critical Care Laboratory</span>
                                </td>
                            </tr>
                        </tbody>
                    </table>
                </td>
            </tr>
        </tbody>
    </table>

```

```

        </tr>
      </tbody>
    </table>
  </td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for backup_and_downloadframe2.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Data View</title>
  <link rel="stylesheet" type="text/css"
href="/backup_and_download/stylesheet/backup_and_download.css" media="screen" />
  <script language="javascript" type="text/javascript"
src="/backup_and_download/js/backup_and_download.js"></script>
</head>
<body onload="document.finalcode.message3.focus()">
  <table id="entirepage">
    <tbody>
      <tr>
        <td class="tableholder">
          <table>
            <tr>
              <td class="tableholder">
                <form id="finalcode" name="finalcode"
action="executethis.py" method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
                  <table>
                    <tbody>
                      <tr>
                        <td style="text-
align:left;padding:10px" id="buttonsholder">
                          <button class="buttons"
type="submit" name="submitidentifiers" id="submitidentifiers"
value="submitidentifiers"
                          tabindex="20" title="Submit
the MONR subject identifier form"
onclick="javascript:executeEditNCopy(0,'cgi',1,0);return false">
                            Backup and Download
                          </button>
                        </td>
                      </tr>
                    </tbody>
                  </table>
                <table>
                  <tbody>
                    <tr>

```

```

                <td id="message1"
class="message">&nbsp;</td>
            </tr>
        </tbody>
    </table>
    <table>
        <tbody>
            <tr>
                <td id="message2" style="text-
align:left; padding-left:10px">&nbsp;</td>
            </tr>
        </tbody>
    </table>
    <table>
        <tbody>
            <tr>
                <td style="text-align:left; padding-
left:10px">
                    <input id="message3"
name="message3" type="text" readonly="readonly"/>
                </td>
            </tr>
        </tbody>
    </table>
</form>
</td>
</tr>
</table>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for backup_and_upload.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Final Code Execution</title>
    <link rel="stylesheet" type="text/css"
href="/backup_and_upload/stylesheet/backup_and_upload.css" media="screen" />
    <link rel="icon" type="image/x-icon" href="/SCCL_test/images/favicon.ico"/>
    <link rel="shortcut icon" type="image/x-icon"
href="/SCCL_test/images/favicon.ico"/>
    <script language="javascript" type="text/javascript"
src="/backup_and_upload/js/backup_and_upload.js"></script>
</head>
<frameset rows="20%,80%" frameborder="0" framespacing="0">
    <frame src="backup_and_uploadframe1.py" frameborder="0" marginwidth="0"
marginheight="0" noresize="noresize" scrolling="no"/>

```

```

        <frame src="backup_and_uploadframe2.py" frameborder="0" marginwidth="0"
marginheight="0" noresize="noresize" scrolling="auto" />
    </frameset>
</html>

```

Code written for backup_and_uploadframe1.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Data View</title>
    <link rel="stylesheet" type="text/css"
href="/backup_and_upload/stylesheets/backup_and_upload.css" media="screen" />
</head>
<body>
    <table id="entirepage">
        <tbody>
            <tr>
                <td class="tableholder">
                    <table>
                        <tbody>
                            <tr>
                                <td class="testwebsite">&nbsp;</td>
                                <td class="logout">&nbsp;</td>
                            </tr>
                        </tbody>
                    </table>
                </td>
            </tr>
            <tr>
                <td class="tableholder">
                    <table id="banner" summary="This table defines the banner">
                        <tbody>
                            <tr>
                                <td id="ubanner">
                                    &nbsp;</td>
                                <td class="smallcaption">
                                    SCCL <span style="font-size:16px">Surgical
Critical Care Laboratory</span>
                                </td>
                            </tr>
                        </tbody>
                    </table>
                </td>
            </tr>
        </tbody>
    </table>
</body>
</html>

```

Code written for backup_and_uploadframe2.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Data View</title>
  <link rel="stylesheet" type="text/css"
href="/backup_and_upload/stylesheets/backup_and_upload.css" media="screen" />
  <script language="javascript" type="text/javascript"
src="/backup_and_upload/js/backup_and_upload.js"></script>
</head>
<body onload="document.finalcode.message3.focus()">
  <table id="entirepage">
    <tbody>
      <tr>
        <td class="tableholder">
          <table>
            <tr>
              <td class="tableholder">
                <form id="finalcode" name="finalcode" action="."
method="post" enctype="application/x-www-form-urlencoded"
onsubmit="javascript:return false">
                  <table>
                    <tbody>
                      <tr>
                        <td style="text-
align:left;padding:10px" id="buttonsholder">
                          <button class="buttons"
type="submit" name="submitidentifiers" id="submitidentifiers"
value="submitidentifiers"
                                tabindex="20" title="Submit
the MONR subject identifier form"
onclick="javascript:executeEditNCopy(0,'cgi',1,0);return false">
                            Backup and Upload
                          </button>
                        </td>
                      </tr>
                    </tbody>
                  </table>
                <table>
                  <tbody>
                    <tr>
                      <td id="message1"
class="message">&nbsp;</td>
                    </tr>
                  </tbody>
                </table>
                <table>
                  <tbody>
                    <tr>
                      <td id="message2" style="text-
align:left; padding-left:10px">&nbsp;</td>
                    </tr>
                  </tbody>
                </table>
              </td>
            </tr>
          </table>
        </td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

```

        <table>
            <tbody>
                <tr>
                    <td style="text-align:left; padding-
left:10px">
                        <input id="message3"
name="message3" type="text" readonly="readonly"/>
                    </td>
                </tr>
            </tbody>
        </table>
    </form>
</td>
</tr>
</table>
</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Code written for backup_and_download.py

```

#!c:/Python/python.exe -u
import MySQLdb as sql
import cgi
import cgitb
cgitb.enable()
import _mysql
import re
import urllib
import string
import shutil
import os
import operator
import datetime
import sys
import subprocess
from subprocess import *
#####
#####
def filecopy(oldfile,newfile):
    shutil.copy2(oldfile,newfile)
#####
#####
form = cgi.FieldStorage()
#####
#####
#try:
cgibindir = "SCCL"
htdocsdir = "../htdocs/SCCL"
jsdir = "../htdocs/SCCL/js"
stylesheetsdir = "../htdocs/SCCL/stylesheets"

```



```

imagesdir = "../htdocs/SCCL/images"
testcgibindir = "SCCL_test"
testhtdocsdir = "../htdocs/SCCL_test"
testjsdir = "../htdocs/SCCL_test/js"
teststylesheetsdir = "../htdocs/SCCL_test/stylesheets"
testimagesdir = "../htdocs/SCCL_test/images"
editedcgibindir = "SCCL_for_upload"
editedhtdocsdir = "../htdocs/SCCL_for_upload"
editedjsdir = "../htdocs/SCCL_for_upload/js"
editedstylesheetsdir = "../htdocs/SCCL_for_upload/stylesheets"
editedimagesdir = "../htdocs/SCCL_for_upload/images"
testeditedcgibindir = "SCCL_for_test_upload"
testeditedhtdocsdir = "../htdocs/SCCL_for_test_upload"
testeditedjsdir = "../htdocs/SCCL_for_test_upload/js"
testeditedstylesheetsdir = "../htdocs/SCCL_for_test_upload/stylesheets"
testeditedimagesdir = "../htdocs/SCCL_for_test_upload/images"
videosdir = "videos"
#####
#####
raj_cgibindir = "raj_sccl"
raj_htdocsdir = "../htdocs/raj_sccl"
raj_jsdir = "../htdocs/raj_sccl/js"
raj_stylesheetsdir = "../htdocs/raj_sccl/stylesheets"
raj_imagesdir = "../htdocs/raj_sccl/images"
raj_testcgibindir = "raj_sccl_test"
raj_testhtdocsdir = "../htdocs/raj_sccl_test"
raj_testjsdir = "../htdocs/raj_sccl_test/js"
raj_teststylesheetsdir = "../htdocs/raj_sccl_test/stylesheets"
raj_testimagesdir = "../htdocs/raj_sccl_test/images"
raj_editedcgibindir = "raj_sccl_for_upload"
raj_editedhtdocsdir = "../htdocs/raj_sccl_for_upload"
raj_editedjsdir = "../htdocs/raj_sccl_for_upload/js"
raj_editedstylesheetsdir = "../htdocs/raj_sccl_for_upload/stylesheets"
raj_editedimagesdir = "../htdocs/raj_sccl_for_upload/images"
raj_testeditedcgibindir = "raj_sccl_for_test_upload"
raj_testeditedhtdocsdir = "../htdocs/raj_sccl_for_test_upload"
raj_testeditedjsdir = "../htdocs/raj_sccl_for_test_upload/js"
raj_testeditedstylesheetsdir = "../htdocs/raj_sccl_for_test_upload/stylesheets"
raj_testeditedimagesdir = "../htdocs/raj_sccl_for_test_upload/images"
raj_videosdir = "raj_videos"
#####
#####
#####modules management
start#####
#####
raj_sccl_test_modules='C:/Python/Lib/site-packages/raj/raj_sccl_test'
raj_sccl_modules='C:/Python/Lib/site-packages/raj/raj_sccl'
sccl_modules='C:/Python/Lib/site-packages/raj/sccl'
sccl_test_modules='C:/Python/Lib/site-packages/raj/sccl_test'
sccl_pth_file='C:/Python/Lib/site-packages/sccl.pth'
#####
#####
web_directories=[cgibindir,htdocsdir,testcgibindir,testhtdocsdir,\
editedcgibindir,editedhtdocsdir,testeditedcgibindir,testeditedhtdocsdir,\
videosdir,\
raj_cgibindir,raj_htdocsdir,raj_testcgibindir,raj_testhtdocsdir,\

```

```

raj_editedcgibindir,raj_editedhtdocsdir,raj_testeditedcgibindir,raj_testeditedhtdo
cdir,\
        raj_videosdir]
module_directories=[raj_sccl_test_modules,raj_sccl_modules,sccl_modules,sccl_test_
modules]
module_setup=['raj/raj_sccl_test','raj/raj_sccl','raj/sccl','raj/sccl_test']
modules_path_set=open(sccl_pth_file,'w')
for thePath in module_setup:
    modules_path_set.write(thePath+'\n')
    if not os.path.exists(thePath):
        os.makedirs(thePath)
    else:
        pass
modules_path_set.close()
#####
#####
if form.has_key("editandcopyfiles"):
    taskToDo = int(form['taskToDo'].value)
    taskNumber = int(form['taskNumber'].value)
    taskNumber = taskNumber+1
    print 'Content-Type: text/javascript\n'
    today = str(datetime.date.today())
    if(taskNumber==1):
        newPaths=["C:/Project_Backup/"+today+"/SCCL_Backup/cgi-bin",\
                "C:/Project_Backup/"+today+"/SCCL_Backup/htdocs",\
                "C:/Project_Backup/"+today+"/SCCL_test_Backup/cgi-bin",\
                "C:/Project_Backup/"+today+"/SCCL_test_Backup/htdocs",\
                "C:/Project_Backup/"+today+"/SCCL_for_upload_Backup/cgi-bin",\
                "C:/Project_Backup/"+today+"/SCCL_for_upload_Backup/htdocs",\
                "C:/Project_Backup/"+today+"/SCCL_for_test_upload_Backup/cgi-
bin",\
                "C:/Project_Backup/"+today+"/SCCL_for_test_upload_Backup/htdocs",\
                "C:/Project_Backup/videos",\
                "C:/Project_Backup/"+today+"/raj_sccl_Backup/cgi-bin",\
                "C:/Project_Backup/"+today+"/raj_sccl_Backup/htdocs",\
                "C:/Project_Backup/"+today+"/raj_sccl_test_Backup/cgi-bin",\
                "C:/Project_Backup/"+today+"/raj_sccl_test_Backup/htdocs",\
                "C:/Project_Backup/"+today+"/raj_sccl_for_upload_Backup/cgi-
bin",\
                "C:/Project_Backup/"+today+"/raj_sccl_for_upload_Backup/htdocs",\
                "C:/Project_Backup/"+today+"/raj_sccl_for_test_upload_Backup/cgi-bin",\
                "C:/Project_Backup/"+today+"/raj_sccl_for_test_upload_Backup/htdocs",\
                "C:/Project_Backup/raj_videos"]
        for x in range(len(web_directories)):
            #for theFile in os.listdir(web_directories[x]):
            try:
                shutil.rmtree(newPaths[x])
            except:
                pass
            try:
                shutil.copytree(web_directories[x],newPaths[x],ignore=None)
            except:
                pass

```

```

        print
"document.getElementById('message2').style.color='black';document.getElementById('
message2').innerHTML='All the files have been backed up<br /><br />';"
    htdocsscldir='../htdocs'
    backup_htdocsdirs=[htdocsscldir]
    backup_htdocsdirs_joined='\"'+('\" \").join(backup_htdocsdirs)+'\"'
    cgibinextractlog='sccl_cgibinextract.log'
    htdocsextractlog='sccl_htdocsextract.log'
    otherfilesextractlog='otherfilesextract.log'
    scclotherfilesextractlog='sccl_otherfilesextract.log'

call('beilmanlabsftpdownload.bat',shell=True,stdout=open('downloadlog.txt','w+'),s
tderr=STDOUT)
    call('7z x ./scclbackup/sccl_cgibin_backup.7z -y -
ppassword',shell=True,stdout=open(cgibinextractlog,'w+'),stderr=STDOUT)
    call('7z x ./scclbackup/sccl_htdocs_backup.7z -o%s -y -
ppassword'%backup_htdocsdirs_joined,shell=True,stdout=open(htdocsextractlog,'w+'),
stderr=STDOUT)
    #call('7z x ./scclbackup/otherfiles_backup.7z -y -
ppassword',shell=True,stdout=open(otherfilesextractlog,'w+'),stderr=STDOUT)
    call('7z x ./scclbackup/sccl_other_backup.7z -y -
ppassword',shell=True,stdout=open(scclotherfilesextractlog,'w+'),stderr=STDOUT)
    #print
"document.getElementById('message2').style.color='black';document.getElementById('
message2').innerHTML='All the files have been downloaded<br /><br />';"
    print "showReport('SFTP download successful','1','green')"
#####
#####
else:
    print 'Content-Type: text/html\n\n'
    filehandle = urllib.urlopen('backup_and_download.html')
    for lines in filehandle.readlines():
        s = lines
        print s
    filehandle.close()
#####
#####
#except:
    #print
"document.getElementById('message2').style.color='black';document.getElementById('
message2').innerHTML='%r';"%sys.exc_info()[1]

```

Code written for backup_and_uploadframe1.py

```

#!c:/Python/python.exe -u
import urllib
#####
#####
print 'Content-Type: text/html\n\n'
filehandle = urllib.urlopen('backup_and_downloadframe1.html')
for lines in filehandle.readlines():
    s = lines
    print s
filehandle.close()
#####
#####

```

Code written for backup_and_downloadframe2.py

```
#!c:/Python/python.exe -u
import urllib
#####
#####
print 'Content-Type: text/html\n\n'
filehandle = urllib.urlopen('backup_and_downloadframe2.html')
for lines in filehandle.readlines():
    s = lines
    print s
filehandle.close()
#####
#####
```

Code written for backup_and_upload.py

```
#!c:/Python/python.exe -u
import MySQLdb as sql
import cgi
import cgi
cgi.enable()
import _mysql
import re
import urllib
import string
import shutil
import os
import operator
import datetime
import subprocess
import py_compile
import datetime
from subprocess import *
#####
#####
def replacemachine(oldfile,newfile,sourceText, replaceText, site):
    if (len(sourceText)==len(replaceText)):
        filex = open(oldfile, "r") #Opens the file in read-mode
        text = filex.readlines() #Reads the file and assigns the value to a
variable
        filex.close() #Closes the file (read session)
        filey = open(newfile, "w") #Opens the new file in write-mode
        textlist = []
        newtext = ''
        #filey.write(text.replace(sourceText[i], replaceText[i])) #replaces all
instances of our keyword
        for line in text:
            if not re.search(r'Ignore_this###',line):
                if not (site=='sclltest'):
                    if re.search(r'Delete_this###',line):
                        pass
                    elif re.search(r'Remove_this###',line):
                        line=line.replace('#Remove_this###','')
                    else:
                        textlist.append(line)
                        newtext += line + '\n'
```

```

        textlist.append(line)
    else:
        for i in range(len(sourceText)):
            line=line.replace(sourceText[i], replaceText[i])
            textlist.append(line)
    else:
        for i in range(len(sourceText)):
            line=line.replace(sourceText[i], replaceText[i])
            textlist.append(line)
    else:
        textlist.append(line)
    newtext = ''.join(textlist)
    file.write(newtext)
    # and writes the whole output when done, wiping over the old contents of
the file
    file.close() #Closes the file (write session)
    filedata=open(newfile,"rb").read()
    newfiledata=filedata.replace("\r\n","\n")
    fileb=open(newfile,"wb")
    fileb.write(newfiledata)
    fileb.close()
else:
    print 'Content-Type: text/javascript\n'
    print "showReport("+str(len(sourceText))+","+str(len(replaceText))+"'Check
the data entered for the text to be replaced')"
    #pass
#####
#####
def convertLineEndings(newfile):
    filedata=open(newfile,"rb").read()
    newfiledata=filedata.replace("\r\n","\n")
    fileb=open(newfile,"wb")
    fileb.write(newfiledata)
    fileb.close()
#####
#####
def filecopy(oldfile,newfile):
    shutil.copy2(oldfile,newfile)
#####
#####
def filemove(oldfile,newfile):
    shutil.move(oldfile,newfile)
#####
#####
def sendThisReport(used_directory,taskNumber):
    global taskToDo,subdirectory,directory,subdirectories,directories_name
    taskToDo=taskToDo+1
    if not (taskToDo==len(os.listdir(used_directory))):
        print
"executeEditNCopy(%d,'%s',%d,%d)"%(taskToDo,directory,subdirectory,taskNumber)
    else:
        taskToDo=0
        if not subdirectory==subdirectories:
            subdirectory=subdirectory+1
        else:
            subdirectory=1
            if
(directories_name.index(directory)+1<len(directories_name)):

```

```

directory=directories_name[(directories_name.index(directory))+1]
    else:
        directory='final'
    print
"executeEditNCopy(%d,'%s',%d,%d)"%(taskToDo,directory,subdirectory,taskNumber)
#####
#####
def showStatus(the_file_name,taskNumber):
    global output_directory
    if re.search(r'.py$',the_file_name):
        color_printed="green"
        report="%s copied to %s folder"%(the_file_name,output_directory)
        print "showReport('%s','%s','%s')"%(report,taskNumber,color_printed)
    elif re.search(r'.html$',the_file_name):
        color_printed="blue"
        report="%s copied to %s folder"%(the_file_name,output_directory)
        print "showReport('%s','%s','%s')"%(report,taskNumber,color_printed)
    elif re.search(r'.js$',the_file_name):
        color_printed="olive"
        report="%s copied to %s folder"%(the_file_name,output_directory)
        print "showReport('%s','%s','%s')"%(report,taskNumber,color_printed)
    elif re.search(r'.css$',the_file_name):
        color_printed="purple"
        report="%s copied to %s folder"%(the_file_name,output_directory)
        print "showReport('%s','%s','%s')"%(report,taskNumber,color_printed)
    elif re.search(r'.jpeg$',the_file_name) or re.search(r'.jpg$',the_file_name)
or re.search(r'.gif$',the_file_name) or re.search(r'.png$',the_file_name) or
re.search(r'.ico$',the_file_name):
        color_printed="maroon"
        report="%s copied to %s folder"%(the_file_name,output_directory)
        print "showReport('%s','%s','%s')"%(report,taskNumber,color_printed)
    #else:
        #color_printed="red"
        #report="%s is ignored"%the_file_name
        #print "showReport('%s','%s','%s')"%(report,taskNumber,color_printed)
#####
#####
form = cgi.FieldStorage()
#####
#####
#os.chdir('C:\Program Files (x86)\Apache Software Foundation\Apache2.2\cgi-bin')
#####
#####
cgibindir = "SCCL"
htdocsdir = "../htdocs/SCCL"
jsdir = "../htdocs/SCCL/js"
stylesheetsdir = "../htdocs/SCCL/stylesheets"
imagesdir = "../htdocs/SCCL/images"
testcgibindir = "SCCL_test"
testhtdocsdir = "../htdocs/SCCL_test"
testjsdir = "../htdocs/SCCL_test/js"
teststylesheetsdir = "../htdocs/SCCL_test/stylesheets"
testimagesdir = "../htdocs/SCCL_test/images"
editedcgibindir = "SCCL_for_upload"
editedhtdocsdir = "../htdocs/SCCL_for_upload"
editedjsdir = "../htdocs/SCCL_for_upload/js"
editedstylesheetsdir = "../htdocs/SCCL_for_upload/stylesheets"

```

```

editedimagesdir = "../htdocs/SCCL_for_upload/images"
testeditedcgibindir = "SCCL_for_test_upload"
testeditedhtdocsdir = "../htdocs/SCCL_for_test_upload"
testeditedjsdir = "../htdocs/SCCL_for_test_upload/js"
testeditedstylesheetsdir = "../htdocs/SCCL_for_test_upload/stylesheets"
testeditedimagesdir = "../htdocs/SCCL_for_test_upload/images"
videodir = "videos"
#####
#####
raj_cgibindir = "raj_sccl"
raj_htdocsdir = "../htdocs/raj_sccl"
raj_jsdir = "../htdocs/raj_sccl/js"
raj_stylesheetsdir = "../htdocs/raj_sccl/stylesheets"
raj_imagesdir = "../htdocs/raj_sccl/images"
raj_testcgibindir = "raj_sccl_test"
raj_testhtdocsdir = "../htdocs/raj_sccl_test"
raj_testjsdir = "../htdocs/raj_sccl_test/js"
raj_teststylesheetsdir = "../htdocs/raj_sccl_test/stylesheets"
raj_testimagesdir = "../htdocs/raj_sccl_test/images"
raj_editedcgibindir = "raj_sccl_for_upload"
raj_editedhtdocsdir = "../htdocs/raj_sccl_for_upload"
raj_editedjsdir = "../htdocs/raj_sccl_for_upload/js"
raj_editedstylesheetsdir = "../htdocs/raj_sccl_for_upload/stylesheets"
raj_editedimagesdir = "../htdocs/raj_sccl_for_upload/images"
raj_testeditedcgibindir = "raj_sccl_for_test_upload"
raj_testeditedhtdocsdir = "../htdocs/raj_sccl_for_test_upload"
raj_testeditedjsdir = "../htdocs/raj_sccl_for_test_upload/js"
raj_testeditedstylesheetsdir = "../htdocs/raj_sccl_for_test_upload/stylesheets"
raj_testeditedimagesdir = "../htdocs/raj_sccl_for_test_upload/images"
raj_videosdir = "raj_videos"
#####
#####
#####modules management
start#####
#####
raj_sccl_test_modules='C:/Python/Lib/site-packages/raj/raj_sccl_test'
raj_sccl_modules='C:/Python/Lib/site-packages/raj/raj_sccl'
sccl_modules='C:/Python/Lib/site-packages/raj/sccl'
sccl_test_modules='C:/Python/Lib/site-packages/raj/sccl_test'
sccl_pth_file='C:/Python/Lib/site-packages/sccl.pth'
#####
#####
web_directories=[cgibindir,htdocsdir,testcgibindir,testhtdocsdir,\

editedcgibindir,editedhtdocsdir,testeditedcgibindir,testeditedhtdocsdir,\
                videodir,\
                raj_cgibindir,raj_htdocsdir,raj_testcgibindir,raj_testhtdocsdir,\

raj_editedcgibindir,raj_editedhtdocsdir,raj_testeditedcgibindir,raj_testeditedhtdo
csdir,\
                raj_videosdir]
module_directories=[raj_sccl_test_modules,raj_sccl_modules,sccl_modules,sccl_test_
modules]
module_setup=['raj/raj_sccl_test','raj/raj_sccl','raj/sccl','raj/sccl_test']
modules_path_set=open(sccl_pth_file,'w')
for thePath in module_setup:
    modules_path_set.write(thePath+'\n')
    if not os.path.exists(thePath):

```

```

        os.makedirs(thePath)
    else:
        pass
modules_path_set.close()
#####modules management
done#####
#####
if form.has_key("editandcopyfiles"):
    taskToDo = int(form['taskToDo'].value)
    taskNumber = int(form['taskNumber'].value)
    #taskToDo=1
    #taskNumber=0
    taskNumber = taskNumber+1
    today = str(datetime.date.today())
    print 'Content-Type: text/javascript\n'
    if(taskNumber==1):
        newPaths=["C:/Project_Backup/"+today+"/SCCL_Backup/cgi-bin",\
                  "C:/Project_Backup/"+today+"/SCCL_Backup/htdocs",\
                  "C:/Project_Backup/"+today+"/SCCL_test_Backup/cgi-bin",\
                  "C:/Project_Backup/"+today+"/SCCL_test_Backup/htdocs",\
                  "C:/Project_Backup/"+today+"/SCCL_for_upload_Backup/cgi-bin",\
                  "C:/Project_Backup/"+today+"/SCCL_for_upload_Backup/htdocs",\
                  "C:/Project_Backup/"+today+"/SCCL_for_test_upload_Backup/cgi-
bin",\
"C:/Project_Backup/"+today+"/SCCL_for_test_upload_Backup/htdocs",\
                  "C:/Project_Backup/videos",\
                  "C:/Project_Backup/"+today+"/raj_sccl_Backup/cgi-bin",\
                  "C:/Project_Backup/"+today+"/raj_sccl_Backup/htdocs",\
                  "C:/Project_Backup/"+today+"/raj_sccl_test_Backup/cgi-bin",\
                  "C:/Project_Backup/"+today+"/raj_sccl_test_Backup/htdocs",\
                  "C:/Project_Backup/"+today+"/raj_sccl_for_upload_Backup/cgi-
bin",\
"C:/Project_Backup/"+today+"/raj_sccl_for_upload_Backup/htdocs",\
"C:/Project_Backup/"+today+"/raj_sccl_for_test_upload_Backup/cgi-bin",\
"C:/Project_Backup/"+today+"/raj_sccl_for_test_upload_Backup/htdocs",\
                  "C:/Project_Backup/raj_videos"]
        for x in range(len(web_directories)):
            #for theFile in os.listdir(web_directories[x]):
            try:
                shutil.rmtree(newPaths[x])
            except:
                pass
            try:
                shutil.copytree(web_directories[x],newPaths[x],ignore=None)
            except:
                pass
        print
"document.getElementById('message2').style.color='black';document.getElementById('
message2').innerHTML='All the files have been backed up<br /><br />';"
    directory = form['directory'].value
    #directory="cgi"#should remove cgi
    directories_name=['cgi', 'js', 'stylesheets', 'images', 'htdocs']
    directory_position=0
    used_directory=''

```



```

used_directories=[testcgibindir, testjsdir, teststylesheetsdir, testimagesdir, testhtdo
ocsdir]
    output_directory=''

output_directories=[cgibindir, editedcgibindir, testeditedcgibindir, raj_testcgibindi
r, raj_cgibindir, raj_editedcgibindir, raj_testeditedcgibindir, \

jsdir, editedjsdir, testeditedjsdir, raj_testjsdir, raj_jsdir, raj_editedjsdir, raj_test
editedjsdir, \

stylesheetsdir, editedstylesheetsdir, testeditedstylesheetsdir, raj_teststylesheetsdi
r, raj_stylesheetsdir, raj_editedstylesheetsdir, raj_testeditedstylesheetsdir, \

imagesdir, editedimagesdir, testeditedimagesdir, raj_testimagesdir, raj_imagesdir, raj_
editedimagesdir, raj_testeditedimagesdir, \

htdocsdir, editedhtdocsdir, testeditedhtdocsdir, raj_testhtdocsdir, raj_htdocsdir, raj_
editedhtdocsdir, raj_testeditedhtdocsdir]
    output_directory_index=0
    directory_index=0
    #subdirectory=1
    subdirectories=7
    color_printed=''
    subdirectory = int(form['subdirectory'].value)#should uncomment this while
testing the code without web browser
    #while True:#should uncomment this while testing the code without web browser

#####
#####
    if (directory=="cgi"):
        #print directory
        directory_position=directories_name.index(directory)
        used_directory=used_directories[directory_position]
        output_directory_index=(directory_position*subdirectories)+(subdirectory-
1)
        output_directory=output_directories[output_directory_index]
        #print output_directory_index, output_directory
        the_file_name=os.listdir(used_directory)[taskToDo]
        if (subdirectory==1):
            if re.match("^.+\.py$", the_file_name):

sourceText=['sccltest_authentication', 'test_', '/project/beilmanlab.umn.edu-
upload/test', 'Location: http://localhost/SCCL_test', 'bloattest']

replaceText=['sccl_authentication', 'monr_', '/project/beilmanlab.umn.edu-
upload', 'Location: http://localhost/SCCL', 'bloat']

replacemachine(used_directory+'/' +the_file_name, output_directory+'/' +the_file_name
, sourceText, replaceText, 'sccl')
    showStatus(the_file_name, taskNumber)
    elif re.match("^.+\.html$", the_file_name):
        sourceText=['<td class="testwebsite">This is a "Test"
website</td>', '/SCCL_test/', 'wfSolChfYxxQWEo0R1NLLw_3d_3d']
        replaceText=['<td
class="testwebsite">&nbsp;</td>', '/SCCL/', 'ht_2ffEr_2bWHbQb0LsdRwrngg_3d_3d']

```

```

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
else:
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)
elif(subdirectory==2):
    if re.match("^.+\.py$",the_file_name):
        sourceText=[ '#!c:/Python/python.exe -
u','sccltest_authentication','test_','/project/beilmanlab.umn.edu-
upload/test','import cgitb','cgitb.enable()','Location:
http://localhost/SCCL_test','bloattest']

replaceText=[ '#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python','sccl_a
uthentication','monr_','/project/beilmanlab.umn.edu-upload','#import
cgitb','#cgitb.enable()','Location: http://sccl.umn.edu','bloat']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
    elif re.match("^.+\.html$",the_file_name):
        sourceText=[ '<td class="testwebsite">This is a "Test"
website</td>','/SCCL_test/','wfSolChfYxxQWEo0R1NLlw_3d_3d']
        replaceText=[ '<td
class="testwebsite">&nbsp;  </td>','/','gRuIzq5s8SRqTbMghIS5dg_3d_3d']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
else:
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)
elif(subdirectory==3):
    if re.match("^.+\.py$",the_file_name):
        sourceText=[ '#!c:/Python/python.exe -u','import
cgitb','cgitb.enable()','Location: http://localhost/SCCL_test']

replaceText=[ '#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python','#impor
t cgitb','#cgitb.enable()','Location: http://sccl.umn.edu/test']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
    showStatus(the_file_name,taskNumber)
    elif re.match("^.+\.html$",the_file_name):
        sourceText=[ '/SCCL_test/','wfSolChfYxxQWEo0R1NLlw_3d_3d']
        replaceText=[ '/test/','3RCEM7PGFMYlmyS3hsZxvA_3d_3d']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
    showStatus(the_file_name,taskNumber)
else:
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)
elif (subdirectory==4):
    if re.match("^.+\.py$",the_file_name):
        sourceText=[ 'Location: http://localhost/SCCL_test']
        replaceText=[ 'Location: http://localhost/raj_sccl_test']

```

```

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
    module_directory=module_directories[0]
    binary_file_name=the_file_name.split('.')[0]
    dfile_for='raj_sccl_test_'+binary_file_name

py_compile.compile(output_directory+'/'+the_file_name,cfile=module_directory+'/'+
raj_sccl_test_'+binary_file_name+'.pyc',dfile=dfile_for)
change_output_file=open(output_directory+'/'+the_file_name,'w')
change_output_file.write('#!c:/Python/python.exe -u\n')
change_output_file.write('from raj_sccl_test_'+binary_file_name+'
import *\n')
    change_output_file.close()
    showStatus(the_file_name,taskNumber)
    elif re.match("^.+\.html$",the_file_name):
        sourceText=['/SCCL_test/']
        replaceText=['/raj_sccl_test/']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
    showStatus(the_file_name,taskNumber)
    else:
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
    elif (subdirectory==5):
        if re.match("^.+\.py$",the_file_name):

sourceText=['sccltest_authentication','test_','/project/beilmanlab.umn.edu-
upload/test','Location: http://localhost/SCCL_test','bloattest']

replaceText=['sccl_authentication','monr_','/project/beilmanlab.umn.edu-
upload','Location: http://localhost/raj_sccl','bloat']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    module_directory=module_directories[1]
    binary_file_name=the_file_name.split('.')[0]
    dfile_for='raj_sccl_'+binary_file_name

py_compile.compile(output_directory+'/'+the_file_name,cfile=module_directory+'/'+
raj_sccl_'+binary_file_name+'.pyc',dfile=dfile_for)
change_output_file=open(output_directory+'/'+the_file_name,'w')
change_output_file.write('#!c:/Python/python.exe -u\n')
change_output_file.write('from raj_sccl_'+binary_file_name+'
import *\n')
    change_output_file.close()
    showStatus(the_file_name,taskNumber)
    elif re.match("^.+\.html$",the_file_name):
        sourceText=['<td class="testwebsite">This is a "Test"
website</td>','/SCCL_test/','wfSolChfYxxQWE0R1NL1w_3d_3d']
        replaceText=['<td
class="testwebsite">&nbsp;</td>','/raj_sccl/','ht_2ffEr_2bWHbQb0LsdRwrngg_3d_3d']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
    else:

```

```

        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
    elif(subdirectory==6):
        if re.match("^.+\.py$",the_file_name):
            sourceText=[ '#!c:/Python/python.exe -
u','sccltest_authentication','test_','/project/beilmanlab.umn.edu-
upload/test','import cgitb','cgitb.enable()','Location:
http://localhost/SCCL_test','bloattest']

replaceText=[ '#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python','sccl_a
uthentication','monr_','/project/beilmanlab.umn.edu-upload','#import
cgitb','#cgitb.enable()','Location: http://sccl.umn.edu','bloat']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    module_directory=module_directories[2]
    binary_file_name=the_file_name.split('.')[0]
    dfile_for='sccl_'+binary_file_name

py_compile.compile(output_directory+'/'+the_file_name,cfile=module_directory+'/'+'
sccl_'+binary_file_name+'.pyc',dfile=dfile_for)
    change_output_file=open(output_directory+'/'+the_file_name,'w')

change_output_file.write('#!/project/beilmanlab.umn.edu/python/version_2.6/bin/pyt
hon\n')
    change_output_file.write('from sccl_'+binary_file_name+' import
*\n')
    change_output_file.close()
    convertLineEndings(output_directory+'/'+the_file_name)
    showStatus(the_file_name,taskNumber)
    elif re.match("^.+\.html$",the_file_name):
        sourceText=[ '<td class="testwebsite">This is a "Test"
website</td>','/SCCL_test/', 'wfSolChfYxxQWEo0R1NlLw_3d_3d' ]
        replaceText=[ '<td
class="testwebsite">&nbsp;</td>', '/', 'gRuIzq5s8SRqTbMghIS5dg_3d_3d' ]

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
    else:
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
    elif(subdirectory==7):
        if re.match("^.+\.py$",the_file_name):
            sourceText=[ '#!c:/Python/python.exe -u','import
cgitb','cgitb.enable()','Location: http://localhost/SCCL_test']

replaceText=[ '#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python','#impor
t cgitb','#cgitb.enable()','Location: http://sccl.umn.edu/test']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
    module_directory=module_directories[3]
    binary_file_name=the_file_name.split('.')[0]
    dfile_for='sccl_test_'+binary_file_name

py_compile.compile(output_directory+'/'+the_file_name,cfile=module_directory+'/'+'
sccl_test_'+binary_file_name+'.pyc',dfile=dfile_for)

```

```

        change_output_file=open(output_directory+'/'+the_file_name,'w')

change_output_file.write('#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python\n')

        change_output_file.write('from sccl_test_'+binary_file_name+'
import *\n')
        change_output_file.close()
        convertLineEndings(output_directory+'/'+the_file_name)
        showStatus(the_file_name,taskNumber)
        elif re.match("^.\.html$", the_file_name):
            sourceText=['/SCCL_test/', 'wfSolChfYxxQWEo0R1Nl1w_3d_3d']
            replaceText=['/test/', '3RCEM7PGFMY1myS3hsZxvA_3d_3d']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
        showStatus(the_file_name,taskNumber)
        else:
            showStatus(the_file_name,taskNumber)
            sendThisReport(used_directory,taskNumber)

#####
#####
        elif(directory=='js'):
            directory_position=directories_name.index(directory)
            used_directory=used_directories[directory_position]
            output_directory_index=(directory_position*subdirectories)+(subdirectory-
1)

            output_directory=output_directories[output_directory_index]
            #print output_directory_index, output_directory
            the_file_name=os.listdir(used_directory)[taskToDo]
            if (subdirectory==1):
                sourceText=['/SCCL_test/', 'test_']
                replaceText=['/SCCL/', 'monr_']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
            showStatus(the_file_name,taskNumber)
            sendThisReport(used_directory,taskNumber)
            elif (subdirectory==2):
                sourceText=['../..../cgi-bin/SCCL_test/', '/cgi-
bin/SCCL_test/', '\'/SCCL_test/', '\'/SCCL_test/', 'domain=localhost;', 'test_']
                replaceText=['../cgi-bin/SCCL/', '/cgi-
bin/SCCL/', '\'/, '\'/, 'domain=sccl.umn.edu;', 'monr_']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
            showStatus(the_file_name,taskNumber)
            sendThisReport(used_directory,taskNumber)
            elif (subdirectory==3):
                sourceText=['\'/SCCL_test/', '\'/SCCL_test/', 'domain=localhost;']
                replaceText=['\'/test/', '\'/test/', 'domain=sccl.umn.edu;']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
            showStatus(the_file_name,taskNumber)
            sendThisReport(used_directory,taskNumber)
            elif (subdirectory==4):
                sourceText=['/SCCL_test/']

```

```

        replaceText=['/raj_sccl_test/']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)
    elif (subdirectory==5):
        sourceText=['/SCCL_test/', 'test_']
        replaceText=['/raj_sccl/', 'monr_']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)
    elif (subdirectory==6):
        sourceText=['../..../cgi-bin/SCCL_test/', '/cgi-
bin/SCCL_test/', '\'/SCCL_test/', '\'/SCCL_test/', 'domain=localhost;', 'test_']
        replaceText=['../cgi-bin/SCCL/', '/cgi-
bin/SCCL/', '\'/, '\'/, 'domain=sccl.umn.edu;', 'monr_']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)
    elif (subdirectory==7):
        sourceText=['\'/SCCL_test/', '\'/SCCL_test/', 'domain=localhost;']
        replaceText=['\'/test/', '\'/test/', 'domain=sccl.umn.edu;']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)

#####
#####
    elif(directory=='stylesheets'):
        directory_position=directories.index(directory)
        used_directory=used_directories[directory_position]
        output_directory_index=(directory_position*subdirectories)+(subdirectory-
1)
        output_directory=output_directories[output_directory_index]
        #print output_directory_index, output_directory
        the_file_name=os.listdir(used_directory)[taskToDo]
        if (subdirectory==1):

filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)
    elif (subdirectory==2):

filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)
    elif (subdirectory==3):

filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)

```

```

        elif (subdirectory==4):
filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
        elif (subdirectory==5):
filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
        elif (subdirectory==6):
filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
        elif (subdirectory==7):
filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)

#####
#####
        elif(directory=='images'):
            directory_position=directories_name.index(directory)
            used_directory=used_directories[directory_position]
            output_directory_index=(directory_position*subdirectories)+(subdirectory-
1)
            output_directory=output_directories[output_directory_index]
            #print output_directory_index, output_directory
            the_file_name=os.listdir(used_directory)[taskToDo]
            if (subdirectory==1):
filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
                showStatus(the_file_name,taskNumber)
                sendThisReport(used_directory,taskNumber)
                elif (subdirectory==2):
filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
                showStatus(the_file_name,taskNumber)
                sendThisReport(used_directory,taskNumber)
                elif (subdirectory==3):
filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
                showStatus(the_file_name,taskNumber)
                sendThisReport(used_directory,taskNumber)
                elif (subdirectory==4):
filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
                showStatus(the_file_name,taskNumber)
                sendThisReport(used_directory,taskNumber)
                elif (subdirectory==5):
filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
                showStatus(the_file_name,taskNumber)
                sendThisReport(used_directory,taskNumber)
                elif (subdirectory==6):

```

```

filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)
elif (subdirectory==7):

filecopy(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name)
    showStatus(the_file_name,taskNumber)
    sendThisReport(used_directory,taskNumber)

#####
#####
elif(directory=='htdocs'):
    directory_position=directories_name.index(directory)
    used_directory=used_directories[directory_position]
    output_directory_index=(directory_position*subdirectories)+(subdirectory-
1)

    output_directory=output_directories[output_directory_index]
    #print output_directory_index, output_directory
    the_file_name=os.listdir(used_directory)[taskToDo]
    if (subdirectory==1):
        if re.match("^.\.py$",the_file_name):
            sourceText=['../../cgi-
bin/SCCL_test','sccltest_authentication','test_','/project/beilmanlab.umn.edu-
upload/test','Location: http://localhost/SCCL_test','bloattest']
            replaceText=['../../cgi-
bin/SCCL','sccl_authentication','monr_','/project/beilmanlab.umn.edu-
upload','Location: http://localhost/SCCL','bloat']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
    else:
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
    elif (subdirectory==2):
        if re.match("^.\.py$",the_file_name):
            sourceText=['#!c:/Python/python.exe -u','import
cgitb','cgitb.enable()','../../cgi-
bin/SCCL_test','sccltest_authentication','test_','/project/beilmanlab.umn.edu-
upload/test','Location: http://localhost/SCCL_test','bloattest']

            replaceText=['#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python','#import
cgitb','cgitb.enable()','../../cgi-
bin/SCCL','sccl_authentication','monr_','/project/beilmanlab.umn.edu-
upload','Location: http://sccl.umn.edu','bloat']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
    else:
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
    elif (subdirectory==3):
        if re.match("^.\.py$",the_file_name):
            sourceText=['#!c:/Python/python.exe -u','import
cgitb','cgitb.enable()','Location: http://localhost/SCCL_test']

```



```

replaceText=['#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python', '#import
cgitb', '#cgitb.enable()', 'Location: http://sccl.umn.edu/test']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
    showStatus(the_file_name,taskNumber)
    else:
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
    elif (subdirectory==4):
        if re.match("^.+\.py$",the_file_name):
            sourceText=['../../cgi-bin/SCCL_test','Location:
http://localhost/SCCL_test']
            replaceText=['../../cgi-bin/raj_sccl_test','Location:
http://localhost/raj_sccl_test']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
    showStatus(the_file_name,taskNumber)
    else:
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
    elif (subdirectory==5):
        if re.match("^.+\.py$",the_file_name):
            sourceText=['../../cgi-
bin/SCCL_test','sccltest_authentication','test_','/project/beilmanlab.umn.edu-
upload/test','Location: http://localhost/SCCL_test','bloattest']
            replaceText=['../../cgi-
bin/raj_sccl','sccl_authentication','monr_','/project/beilmanlab.umn.edu-
upload','Location: http://localhost/raj_sccl','bloat']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
    else:
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
    elif (subdirectory==6):
        if re.match("^.+\.py$",the_file_name):
            sourceText=['#!c:/Python/python.exe -u','import
cgitb','cgitb.enable()','../../cgi-
bin/SCCL_test','sccltest_authentication','test_','/project/beilmanlab.umn.edu-
upload/test','Location: http://localhost/SCCL_test','bloattest']

replaceText=['#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python', '#import
cgitb', '#cgitb.enable()', './cgi-
bin/SCCL','sccl_authentication','monr_','/project/beilmanlab.umn.edu-
upload','Location: http://sccl.umn.edu','bloat']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccl')
    showStatus(the_file_name,taskNumber)
    else:
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)
    elif (subdirectory==7):
        if re.match("^.+\.py$",the_file_name):

```

```

        sourceText=['#!c:/Python/python.exe -u', 'import
cgitb', 'cgitb.enable()', 'Location: http://localhost/SCCL_test']

replaceText=['#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python', '#import
cgitb', '#cgitb.enable()', 'Location: http://sccl.umn.edu/test']

replacemachine(used_directory+'/'+the_file_name,output_directory+'/'+the_file_name
,sourceText,replaceText,'sccltest')
        showStatus(the_file_name,taskNumber)
    else:
        showStatus(the_file_name,taskNumber)
        sendThisReport(used_directory,taskNumber)

#####
#####
    elif(directory=='final'):
        #apachedir='C:/Program Files (x86)/Apache Software Foundation/Apache2.2/'
        cgibinssclmdir='SCCL'
        htdocssclmdir='../htdocs/SCCL'
        cgibinsscltestdir='SCCL_test'
        htdocsscltestdir='../htdocs/SCCL_test'
        cgibinssclluploadir='SCCL_for_upload'
        htdocssclluploadir='../htdocs/SCCL_for_upload'
        cgibinsscltestuploadir='SCCL_for_test_upload'
        htdocsscltestuploadir='../htdocs/SCCL_for_test_upload'
        Other_files='Other_files'
        scclotherfilesdir='sccl_other_files'

backup_cgibindirs=[cgibinssclmdir,cgibinsscltestdir,cgibinssclluploadir,cgibinssclt
estuploadir]

backup_htdocsdirs=[htdocssclmdir,htdocsscltestdir,htdocssclluploadir,htdocssclt
estuploadir]
        backup_otherfilesdirs=[Other_files]
        backup_scclotherfilesdir=[scclotherfilesdir]
        backup_cgibindirs_joined='\"'+('\" \").join(backup_cgibindirs)+'\"'
        backup_htdocsdirs_joined='\"'+('\" \").join(backup_htdocsdirs)+'\"'
        backup_otherfilesdirs_joined='\"'+('\"
\\').join(backup_otherfilesdirs)+'\"'
        backup_scclotherfilesdir_joined='\"'+('\"
\\').join(backup_scclotherfilesdir)+'\"'
        cgibinbackuplog='sccl_cgibinbackup.log'
        htdocsbackuplog='sccl_htdocsbackup.log'
        otherfilesbackuplog='otherfilesbackup.log'
        scclotherfilesbackuplog='sccl_otherfilesbackup.log'
        if os.path.exists('./scclbackup/sccl_cgibin_backup.7z'):
            os.remove('./scclbackup/sccl_cgibin_backup.7z')
        else:
            pass
        if os.path.exists('./scclbackup/sccl_htdocs_backup.7z'):
            os.remove('./scclbackup/sccl_htdocs_backup.7z')
        else:
            pass
        #if os.path.exists('./scclbackup/otherfiles_backup.7z'):
            #os.remove('./scclbackup/otherfiles_backup.7z')
        #else:
            #pass
        if os.path.exists('./scclbackup/sccl_other_backup.7z'):

```

```

        os.remove('./scclbackup/sccl_other_backup.7z')
    else:
        pass
    #print bacup_dirs_joined

#####
#####
    call('7z a -y -ppassword -mhe -t7z ./scclbackup/sccl_cgibin_backup.7z
%s'%backup_cgibindirs_joined,shell=True,stdout=open(cgibinbackuplog,'w+'),stderr=ST
DOUT)
    call('7z a -y -ppassword -mhe -t7z ./scclbackup/sccl_htdocs_backup.7z
%s'%backup_htdocsdirs_joined,shell=True,stdout=open(htdocsbackuplog,'w+'),stderr=ST
DOUT)
    #call('7z a -y -ppassword -mhe -t7z ./scclbackup/otherfiles_backup.7z
%s'%backup_otherfilesdirs_joined,shell=True,stdout=open(otherfilesbackuplog,'w+'),
stderr=STDOUT)
    call('7z a -y -ppassword -mhe -t7z ./scclbackup/sccl_other_backup.7z
%s'%backup_scclotherfilesdir_joined,shell=True,stdout=open(scclotherfilesbackuplog
,'w+'),stderr=STDOUT)

call('beilmanlabsftpupload.bat',shell=True,stdout=open('uploadlog.txt','w+'),stder
r=STDOUT)
    print "showReport('Backup and file transformation successful. 7zip files
created for backup. File upload complete.','%d','black')"%(taskNumber)
    #break
    #print "showReport('Now execute the file beilmanlabsftpupload.exe in cgi-
bin folder','%d','black')"%(taskNumber)
    #subprocess.Popen('beilmanlabsftpupload.exe')

#call(['beilmanlabsftpupload.bat'],stdout=open('uploadlog.txt','w+'),stderr=STDOUT
)
    #os.system('beilmanlabsftpupload.bat')
    #checkoutoutput = open("dobi.txt",'r').read()
    #chekoutoutput = "Successful"
    #print "showReport('SFTP upload successful','%d','black')"%(taskNumber)
    #cmd = os.popen('beilmanlabsftpupload.bat')
    #listOutput= cmd.readlines()
    #result = cmd.close()
    #return_code = os.system ('beilmanlabsftpupload.bat')
    #report = report+str(return_code)
    #print "showReport('%s','%s','green')"%(report,taskNumber)
    #os.system('beilmanlabsftpupload.bat')
    #execfile("try.py")

#subprocess.call('"C:\Users\Administrator\Desktop\runthiswithconsole.exe"',shell=T
rue)
#####
#####
else:
    print 'Content-Type: text/html\n\n'
    filehandle = urllib.urlopen('backup_and_upload.html')
    for lines in filehandle.readlines( ):
        s = lines
        print s
    filehandle.close()
#####
#####

```

Code written for backup_and_uploadframe1.py

```
#!/c:/Python/python.exe -u
import urllib
#####
#####
print 'Content-Type: text/html\n\n'
filehandle = urllib.urlopen('backup_and_uploadframe1.html')
for lines in filehandle.readlines( ):
    s = lines
    print s
filehandle.close()
#####
#####
```

Code written for backup_and_uploadframe2.py

```
#!/c:/Python/python.exe -u
import urllib
#####
#####
print 'Content-Type: text/html\n\n'
filehandle = urllib.urlopen('backup_and_uploadframe2.html')
for lines in filehandle.readlines( ):
    s = lines
    print s
filehandle.close()
#####
#####
```

Code written for checking.py

```
#!/c:/Python/python.exe -u
from subprocess import *
#####
#####
htdocsscldir='./htdocs'
backup_htdocsdirs=[htdocsscldir]
backup_htdocsdirs_joined='\"'+('\" \")'.join(backup_htdocsdirs)+'\"'
cgibinextractlog='sccl_cgibinextract.log'
htdocsextractlog='sccl_htdocsextract.log'
otherfilesextractlog='otherfilesextract.log'
scclotherfilesextractlog='sccl_otherfilesextract.log'
#call(['beilmanlabsftpdownload.bat'],stdout=open('downloadlog.txt','w'),stderr=ST
DOUT)
call('7z x ./scclbackup/sccl_cgibin_backup.7z -y -
ppassword',shell=True,stdout=open(cgibinextractlog,'a'),stderr=STDOUT)
call('7z x ./scclbackup/sccl_htdocs_backup.7z -o%s -y -
ppassword'%backup_htdocsdirs_joined,shell=True,stdout=open(htdocsextractlog,'a'),
stderr=STDOUT)
#call('7z x ./scclbackup/otherfiles_backup.7z -y -
ppassword',shell=True,stdout=open(otherfilesextractlog,'a'),stderr=STDOUT)
call('7z x ./scclbackup/sccl_otherfiles_backup.7z -y -
ppassword',shell=True,stdout=open(scclotherfilesextractlog,'a'),stderr=STDOUT)
```

Code written for compile_source.py

```
#!/c:/Python/python.exe -u
import os
import re
import py_compile
#directory="C:/Program Files/Apache Software Foundation/Apache2.2/cgi-
bin/SCCL_test"
#os.chdir(directory)
file_list=[]
directory_list=['sccl_test','sccl','sccl_for_test_upload','sccl_for_upload']
for the_directory in directory_list:
    for the_file in os.listdir(the_directory):
        if (re.search(r'.py$',the_file)) and not
(re.search(r'compile_source.py',the_file)):
            the_file_name=the_file.split('.')[0]

py_compile.compile(the_directory+'/'+the_file,cfile='raj_'+the_directory+'/'+the_file_name+'.pyc')
            file_list.append(the_file)
        else:
            #print the_file
            pass
```

Code written for beilmanlabsftpdownload.txt

```
option batch on
option confirm off
open sftp://useridhere:passwordhere@hosthere:22 -hostkey="ssh-rsa 1024 rest of the
ssh-rsa key here"
synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl" "/home/sdml/tokachic/Project_Backup/cgi-
bin/SCCL"
synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl_test"
"/home/sdml/tokachic/Project_Backup/cgi-bin/SCCL_test"
synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl_for_upload"
"/home/sdml/tokachic/Project_Backup/cgi-bin/SCCL_for_upload"
synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl_for_test_upload"
"/home/sdml/tokachic/Project_Backup/cgi-bin/SCCL_for_test_upload"
#synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\Other_files" "/home/sdml/tokachic/Project_Backup/cgi-
bin/Other_files"
synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\scclbackup" "/home/sdml/tokachic/Project_Backup/cgi-
bin/scclbackup"
synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl"
"/home/sdml/tokachic/Project_Backup/htdocs/SCCL"
synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl_test"
"/home/sdml/tokachic/Project_Backup/htdocs/SCCL_test"
```

```

synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl_for_upload"
"/home/sdml/tokachic/Project_Backup/htdocs/SCCL_for_upload"
synchronize local -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl_for_test_upload"
"/home/sdml/tokachic/Project_Backup/htdocs/SCCL_for_test_upload"
synchronize local -criteria='both' "C:\Python\Lib\site-packages\raj"
"/home/sdml/tokachic/Project_Backup/raj"
synchronize local -criteria='both' "C:\Python\Lib\site-packages"
"/home/sdml/tokachic/Project_Backup/pathfiles"
exit

```

Code written for beilmanlabsftppupload.txt

```

option batch on
option confirm off
open sftp://userid:password@hostaddress:22 -hostkey="ssh-rsa 1024 rest of the ssh-
rsa key"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl_for_test_upload"
"/project/beilmanlab.umn.edu/www/cgi-bin/SCCL_test"
chmod 644 "/project/beilmanlab.umn.edu/www/cgi-bin/SCCL_test/*.html"
chmod 755 "/project/beilmanlab.umn.edu/www/cgi-bin/SCCL_test/*.py"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl_for_upload"
"/project/beilmanlab.umn.edu/www/cgi-bin/SCCL"
chmod 644 "/project/beilmanlab.umn.edu/www/cgi-bin/SCCL/*.html"
chmod 755 "/project/beilmanlab.umn.edu/www/cgi-bin/SCCL/*.py"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\videos" "/project/beilmanlab.umn.edu/www/cgi-
bin/videos"
chmod 644 "/project/beilmanlab.umn.edu/www/cgi-bin/videos/*.swf"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl_for_test_upload"
"/project/beilmanlab.umn.edu/www/htdocs/test"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl_for_upload"
"/project/beilmanlab.umn.edu/www/htdocs"
synchronize remote -criteria='both' "C:\Python\Lib\site-packages\raj"
"/project/beilmanlab.umn.edu/python/version_2.6/lib/python2.6/site-packages/raj"
put "C:\Python\Lib\site-packages\sccl.pth"
"/project/beilmanlab.umn.edu/python/version_2.6/lib/python2.6/site-packages/"
chmod 644 "/project/beilmanlab.umn.edu/www/htdocs/test/js/*.*"
chmod 644 "/project/beilmanlab.umn.edu/www/htdocs/test/stylesheets/*.*"
chmod 644 "/project/beilmanlab.umn.edu/www/htdocs/test/images/*.*"
chmod 755 "/project/beilmanlab.umn.edu/www/htdocs/test/*.py"
chmod 644 "/project/beilmanlab.umn.edu/www/htdocs/js/*.*"
chmod 644 "/project/beilmanlab.umn.edu/www/htdocs/stylesheets/*.*"
chmod 644 "/project/beilmanlab.umn.edu/www/htdocs/images/*.*"
chmod 755 "/project/beilmanlab.umn.edu/www/htdocs/*.py"
chmod 755 "/project/beilmanlab.umn.edu/python/version_2.6/lib/python2.6/site-
packages/raj/raj_sccl_test/*.pyc"
chmod 755 "/project/beilmanlab.umn.edu/python/version_2.6/lib/python2.6/site-
packages/raj/raj_sccl/*.pyc"

```

```

chmod 755 "/project/beilmanlab.umn.edu/python/version_2.6/lib/python2.6/site-
packages/raj/sccl/*.pyc"
chmod 755 "/project/beilmanlab.umn.edu/python/version_2.6/lib/python2.6/site-
packages/raj/sccl_test/*.pyc"
chmod 644 "/project/beilmanlab.umn.edu/python/version_2.6/lib/python2.6/site-
packages/sccl.pth"
#synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\dbmanager"
"/project/beilmanlab.umn.edu/www/htdocs/dbmanager"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl_for_test_upload"
"/home/sdml/tokachic/Project_Backup/cgi-bin/SCCL_for_test_upload"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl_for_test_upload"
"/home/sdml/tokachic/Project_Backup/htdocs/SCCL_for_test_upload"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl_for_upload"
"/home/sdml/tokachic/Project_Backup/cgi-bin/SCCL_for_upload"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl_for_upload"
"/home/sdml/tokachic/Project_Backup/htdocs/SCCL_for_upload"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl_test"
"/home/sdml/tokachic/Project_Backup/cgi-bin/SCCL_test"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl_test"
"/home/sdml/tokachic/Project_Backup/htdocs/SCCL_test"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\raj_sccl" "/home/sdml/tokachic/Project_Backup/cgi-
bin/SCCL"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\htdocs\raj_sccl"
"/home/sdml/tokachic/Project_Backup/htdocs/SCCL"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\videos" "/home/sdml/tokachic/Project_Backup/cgi-
bin/videos"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\Other_files" "/home/sdml/tokachic/Project_Backup/cgi-
bin/Other_files"
#synchronize remote -criteria='both' "C:\Python\Lib\site-packages\raj"
"/home/sdml/tokachic/Project_Backup/raj"
synchronize remote -criteria='both' "C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\scclbackup" "/home/sdml/tokachic/Project_Backup/cgi-
bin/scclbackup"
put "C:\Python\Lib\site-packages\sccl.pth"
"/home/sdml/tokachic/Project_Backup/pathfiles/"
exit

```

Code written for beilmanlabsftpdownload.bat

```

@echo off
winscp.com /script="C:\Program Files (x86)\Apache Software
Foundation\Apache2.2\cgi-bin\beilmanlabsftpdownload.txt"

```

Code written for beilmanlabsftpupload.bat

```
@echo off  
winscp.com /script="C:\Program Files (x86)\Apache Software  
Foundation\Apache2.2\cgi-bin\beilmanlabsftpupload.txt"
```


Appendix E. Real-time data upload

The list of files used for real-time data upload from the laptop computer collecting experiment data to the production server are shown in Figure 53.

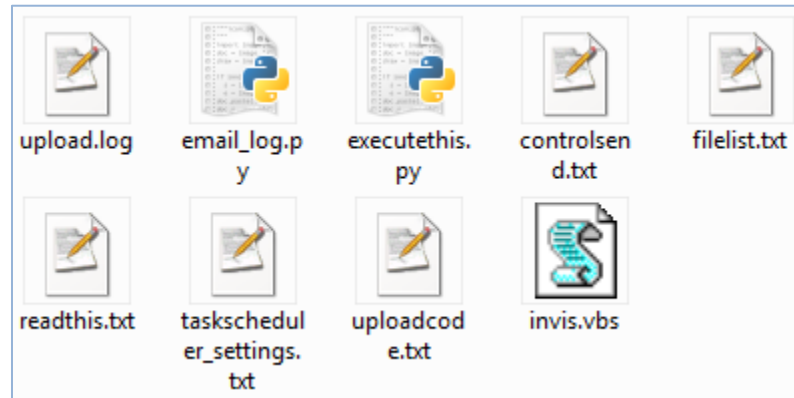


Figure 53 List of files used for silent real-time data upload

Code written for all these files is shown below except for the log files

Code written for email_log.py

```
#!c:/Python/python.exe -u
# Send an HTML email with an embedded image and a plain text message for
# email clients that don't want to display the HTML.
from email.MIMEMultipart import MIMEMultipart
from email.MIMEText import MIMEText
from email.MIMEImage import MIMEImage
from email.MIMEBase import MIMEBase
from email.MIMEAudio import MIMEAudio
from email.Utils import COMMASPACE, formatdate
from email.Message import Message
from email import Encoders
import smtplib
import os
import base64
#####
try:
    controlsend=0
    if not os.path.isfile('controlsend.txt'):
        writetocontrolsend=open('controlsend.txt','w')
        writetocontrolsend.write(str(controlsend))
        writetocontrolsend.close()
    else:
        controlsend=int(open('controlsend.txt','r').read())
        #controlsend=0
```

```

if controlsend==0:
    strFrom = 'sccl@umn.edu'
    strTo = 'tokac006@umn.edu'
    # Create the root message and fill in the from, to, and subject
headers
    msgRoot = MIMEMultipart('related')
    msgRoot['Subject'] = 'Labview upload log'
    msgRoot['From'] = strFrom
    msgRoot['To'] = strTo
    #msg['To'] = COMMASPACE.join(strTo)#If strTo is a list, then use this
line instead of the above
    msgRoot['Date'] = formatdate(localtime=True)
    msgRoot.preamble = 'Labview upload log'
    # Encapsulate the plain and HTML versions of the message body in an
    # 'alternative' part, so message agents can decide which they want to
display.
    msgAlternative = MIMEMultipart('alternative')
    msgRoot.attach(msgAlternative)
    logfileread=open('upload.log','r').read()
    filelistread=open('filelist.txt','r').read()
    msgText = MIMEText(logfileread+'\n'+filelistread)
    msgAlternative.attach(msgText)

    #Important for file attachment
    #Attach first file
    #attach='upload.log'
    #part = MIMEBase('application', 'octet-stream')
    #part.set_payload(open(attach, 'rb').read())
    #Encoders.encode_base64(part)
    #part.add_header('Content-Disposition','attachment; filename="%s"' %
os.path.basename(attach))
    #msgRoot.attach(part)
    #Attach second file
    #attach1='filelist.txt'
    #part1 = MIMEBase('application', 'octet-stream')
    #part1.set_payload(open(attach1, 'rb').read())
    #Encoders.encode_base64(part1)
    #part1.add_header('Content-Disposition','attachment; filename="%s"' %
os.path.basename(attach1))
    #msgRoot.attach(part1)

    # Send the email (this example assumes SMTP authentication is
required)

    session = smtplib.SMTP('smtp.umn.edu',587)
    smtpserver = 'smtp.umn.edu'
    AUTHREQUIRED = 1
    smtpuser = 'sccl'
    smtppass = 'passwordhere'
    session.ehlo(name='umn.edu')
    session.starttls(keyfile=None,certfile=None)
    session.ehlo(name='umn.edu')
    session.login(smtpuser,smtppass)
    session.sendmail(strFrom, strTo, msgRoot.as_string())
    session.quit()
    controlsend=controlsend+1
    writetocontrolsend=open('controlsend.txt','w')
    writetocontrolsend.write(str(controlsend))

```

```

        writetocontrolsend.close()
    else:
        if not controlsend==19:
            controlsend=controlsend+1
            writetocontrolsend=open('controlsend.txt','w')
            writetocontrolsend.write(str(controlsend))
            writetocontrolsend.close()
        else:
            controlsend=0
            writetocontrolsend=open('controlsend.txt','w')
            writetocontrolsend.write(str(controlsend))
            writetocontrolsend.close()
except:
    pass
    raise

```

Code written for executethis.py

```

#!c:/Python/python.exe -u
import re
import string
import shutil
import os
import datetime
from datetime import date,timedelta
import subprocess
import time
from subprocess import *
#####
#####
#subprocess.Popen('beilmanlabsftpupload.exe')
#call(['winscp.com /script="C:/Program Files/Apache Software
Foundation/Apache2.2/cgi-
bin/beilmanlabsftpupload.txt"'],stdout=open('dobi.txt','w'),stderr=STDOUT)
#os.system('winscp.com /script="C:/Program Files/Apache Software
Foundation/Apache2.2/cgi-bin/beilmanlabsftpupload.txt"')
#####
#####
#today = str(date.today())
#yesterday = str(date.today()-timedelta(days=1))
#thedaybefore = str(date.today()-timedelta(days=2))
directory = 'C:\\Program Files\\National Instruments\\LabVIEW
7.0\\user.lib\\beilman.dat'
#directory='C:\\Users\\Administrator\\Desktop\\LabView'
sfile=''
stfile=''
gfile=''
gtfile=''
hfile=''
htfile=''
#####
#####
todaydate = time.strftime('%Y-%m-%d %H:%M:%S',time.localtime())
fmt = '%Y-%m-%d %H:%M:%S'
dt1 = datetime.datetime(*time.strptime(todaydate, fmt)[:6])
#print todaydate

```

```

dt2=''
fmt1='%Y-%m-%d-%H-%M'
#####
#####
S_file_recent=[]
ST_file_recent=[]
G_file_recent=[]
GT_file_recent=[]
H_file_recent=[]
HT_file_recent=[]
for labviewfile in os.listdir(directory):
    if (re.match('^S-',labviewfile)):
        #S_file_recent.append((labviewfile.split('S-')[1]).replace('.txt',''))
        S_file_recent.append(labviewfile)
    elif (re.match('^ST-',labviewfile)):
        ST_file_recent.append(labviewfile)
    elif (re.match('^G-',labviewfile)):
        G_file_recent.append(labviewfile)
    elif (re.match('^GT-',labviewfile)):
        GT_file_recent.append(labviewfile)
    elif (re.match('^H-',labviewfile)):
        H_file_recent.append(labviewfile)
    elif (re.match('^HT-',labviewfile)):
        HT_file_recent.append(labviewfile)
#print
max(S_file_recent),max(ST_file_recent),max(G_file_recent),max(GT_file_recent),max(
H_file_recent),max(HT_file_recent)
S_file_recent=max(S_file_recent)
ST_file_recent=max(ST_file_recent)
G_file_recent=max(G_file_recent)
GT_file_recent=max(GT_file_recent)
H_file_recent=max(H_file_recent)
HT_file_recent=max(HT_file_recent)
#####
#####
for labviewfile in os.listdir(directory):
    if (re.match(S_file_recent,labviewfile)):
        S_file_recent_datestring=(S_file_recent.split('S-')[1]).replace('.txt','')
        dt2 = datetime.datetime(*time.strptime(S_file_recent_datestring,
fmt1)[:6])
        if ((dt1-dt2).days)<3:
            sfile = directory+'\\'+labviewfile
        else:
            pass
    elif (re.match(ST_file_recent,labviewfile)):
        ST_file_recent_datestring=(ST_file_recent.split('ST-
')[1]).replace('.txt','')
        dt2 = datetime.datetime(*time.strptime(ST_file_recent_datestring,
fmt1)[:6])
        if ((dt1-dt2).days)<3:
            stfile = directory+'\\'+labviewfile
        else:
            pass
    elif (re.match(G_file_recent,labviewfile)):
        G_file_recent_datestring=(G_file_recent.split('G-')[1]).replace('.txt','')
        dt2 = datetime.datetime(*time.strptime(G_file_recent_datestring,
fmt1)[:6])
        if ((dt1-dt2).days)<3:

```

```

        gfile = directory+'\\'+labviewfile
    else:
        pass
    elif (re.match(GT_file_recent,labviewfile)):
        GT_file_recent_datestring=(GT_file_recent.split('GT-
')[1]).replace('.txt','')
        dt2 = datetime.datetime(*time.strptime(GT_file_recent_datestring,
fmt1)[:6])
        if ((dt1-dt2).days)<3:
            gtfiler = directory+'\\'+labviewfile
        else:
            pass
    elif (re.match(H_file_recent,labviewfile)):
        H_file_recent_datestring=(H_file_recent.split('H-')[1]).replace('.txt','')
        dt2 = datetime.datetime(*time.strptime(H_file_recent_datestring,
fmt1)[:6])
        if ((dt1-dt2).days)<3:
            hfile = directory+'\\'+labviewfile
        else:
            pass
    elif (re.match(HT_file_recent,labviewfile)):
        HT_file_recent_datestring=(HT_file_recent.split('HT-
')[1]).replace('.txt','')
        dt2 = datetime.datetime(*time.strptime(HT_file_recent_datestring,
fmt1)[:6])
        if ((dt1-dt2).days)<3:
            htfile = directory+'\\'+labviewfile
        else:
            pass
#print sfile,stfile,gfile,gtfile,hfile,htfile
#####
#####
allfiles = '\"'+sfile+'\"'+\" \"+'\"'+stfile+'\"'+\" \"+'\"'+gfile+'\"'+
+\"'+gtfile+'\"'+\" \"+'\"'+hfile+'\"'+\" \"+'\"'+htfile+'\"'
#print allfiles
readthis = open('C:/LabViewDataUploadRaj/readthis.txt','r').read()
readthis = readthis.replace('#files#',allfiles)
uploadcode = open('C:/LabViewDataUploadRaj/uploadcode.txt','w')
uploadcode.write(readthis)
uploadcode.close()
filelist = open('C:/LabViewDataUploadRaj/filelist.txt','w')
filelist.write(sfile+'\n')
filelist.write(stfile+'\n')
filelist.write(gfile+'\n')
filelist.write(gtfile+'\n')
filelist.write(hfile+'\n')
filelist.write(htfile+'\n')
filelist.close()
#print readthis
execfile('email_log.py')
call(['winscp.com','/script="C:/LabViewDataUploadRaj/uploadcode.txt"'],stdout=open
('C:/LabViewDataUploadRaj/upload.log','w'),stderr=STDOUT)

```

Contents of controlsend.txt is a number with initial value zero.

Code written for readthis.txt

```
option batch on
open sftp://username:password@hostname:22 -hostkey="ssh-rsa 1024 rest of the ssh-
rsa key"
option confirm off
put -permissions=644 #files# "/home/sdml/tokachic/labviewfilesforupload/"
call python /home/sdml/tokachic/labviewcode/labviewdataupload.py
exit
```

Code written for invis.vbs (This code is obtained from a random forum on the internet.

Not the original work).

```
CreateObject("Wscript.Shell").Run """" & WScript.Arguments(0) & """" , 0, False
```

Appendix F. Real-time data extraction

List of files needed for real-time data extraction are dbconnection and LabViewFileProcessorFinal.py. File dbconnection consists of information required for logging into the MySQL database. File LabViewFileProcessor.py consists of the code for extracting data from the real-time data files uploaded to the server.

Code written for LabViewFileProcessor.py

```
#!/project/beilmanlab.umn.edu/python/version_2.6/bin/python
import thread
import string
import cgi
import MySQLdb as sql
import re
import shutil
import operator
import os
import time
import datetime
#####
#User defined class declarations
#####
#User defined errors for this file
class Error(Exception):
    #Base class for exceptions in this file
    pass
class InputError(Error):
    def __init__(self,expr,msg):
        self.expr=expr
        self.msg=msg
    def __str__(self):
        return repr(self.msg)
#####
#function declarations
#####
def replacemachine(datalist,newfile,sourceTextlist, replaceTextlist):
    if (len(sourceTextlist)==len(replaceTextlist)): #Check if the lists match in
length
        refineddatalist=[] #Declare an empty list
        writenewfile=open(newfile,'a') #Open the file to write the refined text
for documentation
        for i in range(len(datalist)): #Loop through the list of the data
            refinedstring=datalist[i]
            for j in range(len(sourceTextlist)): #Loop through the list of source
text to replace
                refinedstring=refinedstring.replace(sourceTextlist[j],replaceTextlist[j]) #Replace
the text from each list item (a string)
```

```

        refineddatalist.append(refinedstring) #Append the refined string to
the new list
        writenewfile.write(refinedstring) #Write the refined string to the new
file for documentation
        writenewfile.close() #Close the file which was opened before
        return refineddatalist #Return the new list
    else:
        #If the source text and replace text lengths don't match raise a pre-
defined error
        raise InputError("InputMatch","The sourceText and replaceText lists should
match in length")
#####
def RemoveDuplicates(arr,filename,filestoragepath,pathforduplicatesfile):
    filetowrite = open(filestoragepath+filename,'a') #Creates the file and open's
it for writing data
    #need to change the path below
    duplicates = open(pathforduplicatesfile+'removed_duplicates_'+filename,'a')
    lastValue=''
    finaloutputlist=[]
    for i in range(len(arr)):
        curValue=arr[i][0]+"-"+arr[i][1] #Get the date and time together
        #testcurValue=arr[i][0]+"-"+arr[i][1]
        if (curValue != lastValue): #If the previous line's data and time
(lastValue) is not equal to the present line's data and time(curValue), write the
present line to file
            filetowrite.write(' '.join(arr[i])+'\n') #Write the line by joining
the array to make it a string
            finaloutputlist.append(' '.join(arr[i]))
            lastValue=curValue #Store the current data and time value in the
variable called last value
        else:
            duplicates.write(' '.join(arr[i])+'\n') #If the previous line's data
and time (lastValue) is equal to the present line's data and time(curValue), write
to duplicatesremoved file
            ##print arr[i][0],lastValue,testcurValue
    filetowrite.close() #Close the file once the writing process has completed
    duplicates.close() #Close the file once the writing process has completed
    return finaloutputlist
#####
def
AddDatesSortUnique(arr,filename,filestoragepath,Date1,Date2,pathforduplicatesfile)
:
    #filetowrite = open(filename,'a')
    lastValue='initiate' #Initiate lastValue variable with some string
    const = 0 #Initiate a variable that takes zero and one values
    for i in range(len(arr)): #Supply the array and parse it
        if (lastValue == 'initiate'): #For the first time, lastValue variable will
be 'initiate', so this code will run
            curValue=arr[i][0] #Get the arr[i][0] which is the time when the data
point collected
            lastValue=curValue #Set the lastValue variable to the current time
point
            #arr[i][0]=Date1+' '+arr[i][0]
            arr[i].insert(0,Date1) #Insert the Date1 at the start of the array for
the very first line the array[0]
            elif (lastValue != 'initiate'): #From second time onwards, the lastValue
variable will not be equal to 'initiate'

```



```

        curValue=arr[i][0] #Get the arr[i][0] which is the time when the data
point collected
        curSpl = curValue.split(':') #Split the time of curValue array to get
hours, min, sec needed to differentiate the mis printed values and the next day
values
        lastSpl = lastValue.split(':') #Split the time of lastSpl array to get
hours, min, sec needed to differentiate the mis printed values and the next day
values
        curhrs = int(curSpl[0])
        lasthrs = int(lastSpl[0])
        curmins = int(curSpl[1])
        lastmins = int(lastSpl[1])
        #cursecs = int(curSpl[2])
        #lastsecs = int(lastSpl[2])
        if (curValue < lastValue) and ((lasthrs-curhrs)>15): #This checks if
the last value's hours are 15hours greater than the current value, then it is next
day
            arr[i].insert(0,Date2)
            #arr[i][0]=Date2+' '+arr[i][0]
            const = 1 #Set value of variable const to one inorder to make sure
that the array from now on will get the next date
            elif (curValue > lastValue) and ((curhrs-lasthrs)>15): #This checks if
the current value is greater than the last value but if it is greater by 15 hours,
that means, it is the mis printed value just after the date has changed. So, add
the date1 instead of date 2
                arr[i].insert(0,Date1)
                #arr[i][0]=Date1+' '+arr[i][0]
            elif (curValue > lastValue) and (const == 0): #This checks if the
current value is greater and if it is the first day
                arr[i].insert(0,Date1)
                #arr[i][0]=Date1+' '+arr[i][0]
            elif (curValue < lastValue) and (const == 0): #This checks if the
current value is less than the last value and still it is day1. This happens when
the data point mis prints after some data points
                arr[i].insert(0,Date1)
                #arr[i][0]=Date1+' '+arr[i][0]
            elif (curValue == lastValue) and (const == 0): #This checks if the
time points are equal, and it will continue with the date insertion. This is a
duplicate value and should be dealt with afterwards
                arr[i].insert(0,Date1)
                #arr[i][0]=Date1+' '+arr[i][0]
            elif (curValue > lastValue) and (const == 1): #Same as above but
date2
                arr[i].insert(0,Date2)
                #arr[i][0]=Date2+' '+arr[i][0]
            elif (curValue < lastValue) and (const == 1): #Same as above but date2
                arr[i].insert(0,Date2)
                #arr[i][0]=Date2+' '+arr[i][0]
            elif (curValue == lastValue) and (const == 1): #Same as above but
date2
                arr[i].insert(0,Date2)
                #arr[i][0]=Date2+' '+arr[i][0]
        lastValue = curValue
        reArrange = sorted(arr) #Now sort the array
        #reArrange = sorted(arr,key = operator.itemgetter(1))
        return
RemoveDuplicates(reArrange,filename,filestoragepath,pathforduplicatesfile) #Send
the array to function RemoveDuplicates, to remove duplicates

```

```

#####
def modify_datetime(datalist,backupfile,filestoragepath,time_difference):
    finaloutputlist=[]
    openfile = open(filestoragepath+backupfile,'a') #Open file2 for writing
    for line in datalist:
        machine_datetime = line.split()[0]+' '+line.split()[1]
        fmt = '%Y-%m-%d %H:%M:%S'
        dt1 = datetime.datetime(*time.strptime(machine_datetime, fmt)[:6])
        changed_datetime = dt1-time_difference
        changed_date = (str(changed_datetime)).split()[0]
        changed_time = (str(changed_datetime)).split()[1]
        #The four lines below are added in order to add a zero infront of single
digit hours, for example 01 instead of 1.
        #This will help sort the file properly using sort function in python.
        if len(changed_time)==7:
            changed_time='0'+changed_time
        else:
            pass
        templist = line.split()
        templist.pop(0)
        templist.pop(0)
        templist.insert(0,changed_time)
        templist.insert(0,changed_date)
        openfile.write(' '.join(templist)+'\n')
        finaloutputlist.append(' '.join(templist))
    openfile.close()
    return finaloutputlist
#####
#####
def create_directories(directorypath,directorylist,permissions):
    for sub_directoty in directorylist:
        try:
            #Make sure the path exists
            if not os.path.exists(directorypath+sub_directoty):
                os.umask(0)
                os.makedirs(directorypath+sub_directoty,permissions)
            else:
                pass
        except:
            pass
#####
#####
def updateEventsandTimes(Table_name):
    connection = sql.connect(host=lhost,port=lport,user=luser,
passwd=lpasswd,db=lodb)
    cursor = connection.cursor()
    arterialpressure_events={}
    events_noted={}
    events_missing=[]
    cursor.execute("""SELECT Date_of_experiment,Time_of_experiment,Event_name FROM
%s WHERE Event_name!=99 AND Pig_ID='%s'"""%(Table_name,Pig_ID))
    output=cursor.fetchall()
    for i in range(len(output)):
        arterialpressure_events[output[i][2]]=str(output[i][0])+
'+str(output[i][1])
    cursor.execute("""SELECT Date_of_experiment,Time_of_experiment,Event_name FROM
monr_event_time_info WHERE Pig_ID='%s'"""%(Pig_ID))
    output1=cursor.fetchall()

```

```

if (len(output1)>len(output)):
    events_noted={}
    events_missing=[]
    for k in range(len(output1)):
        events_noted[output1[k][2]]=str(output1[k][0])+' '+str(output1[k][1])
    for l in events_noted:
        if not l in arterialpressure_events or not
arterialpressure_events[l]==events_noted[l]:
            events_missing.append(l)
        else:
            pass
    for n in range(len(events_missing)):
        noteddatetetimeofevent=events_noted[events_missing[n]]
        cursor.execute("""SELECT Date_of_experiment,Time_of_experiment FROM %s
WHERE (TIMESTAMP(Date_of_experiment,Time_of_experiment)>TIMESTAMP('%s')) AND
Pig_ID='%s' LIMIT 1"""%(Table_name,noteddatetetimeofevent,Pig_ID))
        output2=cursor.fetchall()
        if len(output2)>0:
            cursor.execute("""SELECT
Date_of_experiment,Time_of_experiment,Event_name,TIMEDIFF(TIMESTAMP(Date_of_experi
ment,Time_of_experiment),TIMESTAMP('%s')) AS Time_diff FROM %s WHERE (Pig_ID='%s')
AND
((TIMESTAMP(Date_of_experiment,Time_of_experiment)>SUBTIME(TIMESTAMP('%s'),'00:20:
00')) AND
(TIMESTAMP(Date_of_experiment,Time_of_experiment)<ADDTIME(TIMESTAMP('%s'),'00:20:0
0')))) ORDER BY
Date_of_experiment,Time_of_experiment""%(noteddatetetimeofevent,Table_name,Pig_ID,n
oteddatetetimeofevent,noteddatetetimeofevent))
            output3=cursor.fetchall()
            if len(output3)>0:
                timedifflist=[]
                for m in range(len(output3)):
                    timedifflist.append(output3[m][3])

The_date_time_with_min_diff=output3[(timedifflist.index(min(timedifflist)))]
                old_date=str(The_date_time_with_min_diff[0])
                old_time=str(The_date_time_with_min_diff[1])
                old_event=int(The_date_time_with_min_diff[2])
                updated_date=noteddatetetimeofevent.split(' ')[0]
                updated_time=noteddatetetimeofevent.split(' ')[1]
                updated_event=int(events_missing[n])
                Time_diff=str(min(timedifflist))
                cursor.execute("""UPDATE %s SET
Date_of_experiment='%s',Time_of_experiment='%s',Event_name=%d,Time_diff='%s' WHERE
Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Table_name,updated_date,updated_time,updated_event,Tim
e_diff,Pig_ID,old_date,old_time))
                #print "Finished ",updated_event,Table_name
            else:
                pass
        else:
            pass
    else:
        pass
    connection.commit()
    cursor.close()
    connection.close()

```

```

#####
#####
connectparam = []
dbconnection = open('dbconnection','r')
for line in dbconnection:
    connectparam.append(line.strip("\n"))
lhost = connectparam[0]
lport = int(connectparam[1])
luser = connectparam[2]
lpasswd = connectparam[3]
ldb = connectparam[4]
#####
#####
unsortedrawfilestore = os.listdir('rawfilestore')
rawfilestore=sorted(unsortedrawfilestore)
splitfilename=[]
Pig_ID=''
subject_id=''
#for rawfile in rawfilestore:
    #splitfilename = rawfile.split('-')
    #break
#subject_id = splitfilename[2]+splitfilename[3]+splitfilename[1][2:]
#####
#####

#####
#####
for rawfile in rawfilestore:
    splitfilename = rawfile.split('-')
    subject_id = splitfilename[2]+splitfilename[3]+splitfilename[1][2:]
    maindirectorypath='Labviewfiles_processed/'+subject_id+'/'
    directories_for_each_subject = ['Vitals','StO2','BloodGases']
    create_directories(maindirectorypath,directories_for_each_subject,0771)
    directories_for_each_datagroup =
['rawdata','timefiles','removedtext','processflow','extractedoutput']
    tempfilestoredir='rawfilestore/'
    for the_dir in directories_for_each_subject:
        subdirectorypath='Labviewfiles_processed/'+subject_id+'/'+'the_dir+'/'
        create_directories(subdirectorypath,directories_for_each_datagroup,0771)
        #if not (rawfile in column9files) and not (rawfile in column7files) and not
(rawfile in opzerobytefiles):
            if (re.match(r'^S',rawfile)) and not (re.match(r'^ST',rawfile)):
                if not (os.stat(tempfilestoredir+rawfile).st_size==0):
                    #try:
                    #Define some permanent paths
                    processflowpath=maindirectorypath+'Vitals/processflow/'
                    removedtextpath=maindirectorypath+'Vitals/removedtext/'
                    extractedoutputpath=maindirectorypath+'Vitals/extractedoutput/'
                    rawdatapath=maindirectorypath+'Vitals/rawdata/'
                    timefilepath=maindirectorypath+'Vitals/timefiles/'
                    #####Extracting
Date#####
                    splitfilename = rawfile.split("-")
                    startDate = []
                    endDate = []
                    for value in range(1,4):
                        if not (value==3):

```

```

        endDate.append(splitfilename[value]) #File name consists of
the date the file has been created
    else:
        day = int(splitfilename[value])+01
        if (day<10):
            daystring = "0"+str(day)
        else:
            daystring = str(day)
        endDate.append(daystring)
    startDate.append(splitfilename[value])
date1 = "-".join(startDate)
date2 = "-".join(endDate)
#####Date extraction
complete#####
filestage1 = 'firststage.txt'#type the name of the empty file within
quotes

liststage1=[]#Empty list corresponding to the first stage
#####
corresponding_ST_filename=''
extractData = open(tempfilestoredir+rawfile,'r')
storerawdata=open(rawdatapath+rawfile,'a')
writeData1 = open(processflowpath+filestage1,'a')
unwantedlines = open( removedtextpath+'removed_text.txt','a')
justonce=0
machine_date=date1
computer_date=date1
machine_time=''
computer_time=''
for line in extractData:
    storerawdata.write(line)
    if re.match('^\\d\\d:\\d\\d:\\d\\d',line): #Match the lines with time
printed at the start of the line and write it to the firststage.txt
        if (justonce==0):
            machine_time=line.split()[0]
            writeData1.write(line)
            liststage1.append(line)
            justonce=1
            ##print "Machine time = ",machine_time
            for the_file in os.listdir('./rawfilestore'):
                if re.match(r'^ST-'+splitfilename[1]+'-'+splitfilename[2]+'-'+splitfilename[3],the_file):
                    corresponding_ST_filename=the_file

computer_time=open('./rawfilestore/'+the_file,'r').read()
                    storetimefile =
open(timefilepath+corresponding_ST_filename,'w')
                    storetimefile.write(computer_time)
                    storetimefile.close()
                else:
                    pass
                    ##print "Computer time = ",computer_time
            else:
                writeData1.write(line)
                liststage1.append(line)
        else:
            unwantedlines.write(line) #Write removed text to
removed_text.txt
            storerawdata.close()

```

```

extractData.close()
writeData1.close()
unwantedlines.close()
if not machine_time=='':
    machine_datetime=machine_date+' '+machine_time
    computer_datetime=computer_date+' '+computer_time
    ##print machine_datetime
    ##print computer_datetime
    fmt = '%Y-%m-%d %H:%M:%S'
    dt1 = datetime.datetime(*time.strptime(machine_datetime, fmt)[:6])
    dt2 = datetime.datetime(*time.strptime(computer_datetime,
fmt)[:6])

    timedifference=dt1-dt2
    ##print timedifference
    #print '1 out of 6 processes complete'

#####
    filestage2='secondstage.txt'#new file will be created here. So,
just give any name
    liststage2=[]

#####

#shutil.copy2(processflowpath+filestage1,processflowpath+filestage2)
    sourceText1=[' ??' '\2', '/', '??.', '++.', '--.']
    replaceText1=['', ' ', '0', '0', '0']

liststage2=replacemachine(liststage1,processflowpath+filestage2,sourceText1,replaceText1)

    #print '2 out of 6 processes complete'

#####
    filestage3='thirdstage.txt'
    liststage3=[]
    tempCOList=[]

#####
    writeData3 = open(processflowpath+filestage3,'a')
    tempcardiacoutput =
open(processflowpath+'rawcardiacoutput.txt','a')
    for line in liststage2:
        if (len(line.split(' '))>5):
            if(re.match(r'\d\d:\d\d',line.split(' ')[-4]) and
re.match(r'\d.\d',line.split(' ')[-3])): #[-4] and [-3] positions of list is
chosen as last two characters are special characters
                #if (re.search(r'\s\d\d:\d\d\s\d\.\d',line)):
                    tempcardiacoutput.write(line.split(' ')[-4]+'
'+line.split(' ')[-3]+'\\n') #Write the cardiac output data to
rawcardiacoutput.txt. It contains duplicates.
                    tempCOList.append(line.split(' ')[-4]+' '+line.split('
')[-3])

                ##print line.split(' ')[-4]
            else:
                pass
        else:
            pass

```

```

        writeData3.write(re.sub('\s\d\d:\d\d\s\d.\d', '
',line))#Remove cardiac output from secondstage2.txt and write the data to
thirdstage.txt for process flow backup
        liststage3.append(re.sub('\s\d\d:\d\d\s\d.\d', '
',line))#Remove cardiac output from secondstage2.txt and write the data to
thirdstage
        tempcardiacoutput.close()
        writeData3.close()

#####
        cardiacoutput = 'cardiacoutput.txt'
        COlist = []
        temp = []
        for line in tempCOlist:
            temp.append(line.split())

COlist=AddDatesSortUnique(temp,cardiacoutput,extractedoutputpath,date1,date2,remov
edttextpath)

#####
        filestage4='fourthstage.txt'
        liststage4=[]

#####
        writeData4 = open(processflowpath+filestage4,'a')
        unwantedlines = open(removedtextpath+'removed_text.txt','a')
        for line in liststage3:
            matchedstring=re.search('\s\d\d:\d\d\s\s\s\d',line)
            if not matchedstring==None:
                unwantedlines.write(matchedstring.group(0)+'\n')
            else:
                pass
            tempstring=re.sub('\s\d\d:\d\d\s\s\s\d','',line)
            writeData4.write(tempstring)#This is an unwanted line showing
up sometimes and giving error. Remove it from thirdstage.txt and write data to
fourthstage.txt
            liststage4.append(tempstring)
        writeData4.close()
        unwantedlines.close()
        #print '3 out of 6 processes complete'

#####
        filestage5 = 'fifthstage.txt'
        liststage5=[]

#####
        newfile = []
        temp = []
        for line in liststage4:
            temp.append(line.split())

liststage5=AddDatesSortUnique(temp,filestage5,processflowpath,date1,date2,removedt
extpath)
        #print '4 out of 6 processes complete'

#####
        filestage6 = 'sixthstage.txt'
        liststage6=[]

```

```

#####

liststage6=modify_datetime(liststage5,filestage6,processflowpath,timedifference)
#Solve the discrepancy in machine time and computer time
    #print '5 out of 6 processes complete'

#####

    artpressure = 'artpressure.txt'
    papressure = 'papressure.txt'
    spo2 = 'spo2.txt'
    tencolumnfile = 'tencolumnfile.txt'
    eightcolumnfile = 'eightcolumnfile.txt'
    fivecolumnfile = 'fivecolumnfile.txt'
    writeArtP = open(extractedoutputpath+artpressure,'a')
    writePaP = open(extractedoutputpath+papressure,'a')
    writeSpo2 = open(extractedoutputpath+spo2,'a')
    writetencolumns = open(processflowpath+tencolumnfile,'a')
    writeeightcolumns = open(processflowpath+eightcolumnfile,'a')
    writefivecolumns = open(processflowpath+fivecolumnfile,'a')
    discardedArtP = open(removedtextpath+'discarded_ArtP.txt','a')
    discardedPaP = open(removedtextpath+'discarded_PaP.txt','a')
    discardedSpo2 = open(removedtextpath+'discarded_Spo2.txt','a')
    ArtPressure=[]
    PaPressure=[]
    SpO2=[]
    for line in liststage6:
        ArtPtemp = []
        PaPtemp = []
        Spo2temp = []
        if (len(line.split())==10):
            if not (int(line.split()[2])<30\
and not (int(line.split()[2])>200\
and not (int(line.split()[3])<20\
and not (int(line.split()[3])>110:
                ArtPtemp.append(line.split()[0])
                ArtPtemp.append(line.split()[1])
                ArtPtemp.append(line.split()[2])
                ArtPtemp.append(line.split()[3])
                writeArtP.write(' '.join(ArtPtemp)+'\n')
                ArtPressure.append(' '.join(ArtPtemp))
            else:
                ArtPtemp.append(line.split()[0])
                ArtPtemp.append(line.split()[1])
                ArtPtemp.append(line.split()[2])
                ArtPtemp.append(line.split()[3])
                discardedArtP.write(' '.join(ArtPtemp)+'\n')
        if not (int(line.split()[5])<10\
and not (int(line.split()[5])>60\
and not (int(line.split()[6])<3\
and not (int(line.split()[6])>30:
                PaPtemp.append(line.split()[0])
                PaPtemp.append(line.split()[1])
                PaPtemp.append(line.split()[5])
                PaPtemp.append(line.split()[6])
                writePaP.write(' '.join(PaPtemp)+'\n')
                PaPressure.append(' '.join(PaPtemp))
        else:

```



```

        PaPtemp.append(line.split()[0])
        PaPtemp.append(line.split()[1])
        PaPtemp.append(line.split()[5])
        PaPtemp.append(line.split()[6])
        discardedPaP.write(' '.join(PaPtemp)+'\n')
    if not (int(line.split()[8])<60\
    and not (int(line.split()[8])>350\
    and not (int(line.split()[9])<40\
    and not (int(line.split()[9])>100:
        Spo2temp.append(line.split()[0])
        Spo2temp.append(line.split()[1])
        Spo2temp.append(line.split()[8])
        Spo2temp.append(line.split()[9])
        writeSpo2.write(' '.join(Spo2temp)+'\n')
        SpO2.append(' '.join(Spo2temp))
    else:
        Spo2temp.append(line.split()[0])
        Spo2temp.append(line.split()[1])
        Spo2temp.append(line.split()[8])
        Spo2temp.append(line.split()[9])
        discardedSpo2.write(' '.join(Spo2temp)+'\n')
    writetencolumns.write(line+'\n')
elif (len(line.split())==8):
    if not (int(line.split()[2])<30\
    and not (int(line.split()[2])>200\
    and not (int(line.split()[3])<20\
    and not (int(line.split()[3])>110:
        ArtPtemp.append(line.split()[0])
        ArtPtemp.append(line.split()[1])
        ArtPtemp.append(line.split()[2])
        ArtPtemp.append(line.split()[3])
        writeArtP.write(' '.join(ArtPtemp)+'\n')
        ArtPressure.append(' '.join(ArtPtemp))
    else:
        ArtPtemp.append(line.split()[0])
        ArtPtemp.append(line.split()[1])
        ArtPtemp.append(line.split()[2])
        ArtPtemp.append(line.split()[3])
        discardedArtP.write(' '.join(ArtPtemp)+'\n')
    if not (int(line.split()[5])<10\
    and not (int(line.split()[5])>60\
    and not (int(line.split()[6])<3\
    and not (int(line.split()[6])>30:
        PaPtemp.append(line.split()[0])
        PaPtemp.append(line.split()[1])
        PaPtemp.append(line.split()[5])
        PaPtemp.append(line.split()[6])
        writePaP.write(' '.join(PaPtemp)+'\n')
        PaPressure.append(' '.join(PaPtemp))
    else:
        PaPtemp.append(line.split()[0])
        PaPtemp.append(line.split()[1])
        PaPtemp.append(line.split()[5])
        PaPtemp.append(line.split()[6])
        discardedPaP.write(' '.join(PaPtemp)+'\n')
    writeeightcolumns.write(line+'\n')
elif (len(line.split())==5):
    if not (int(line.split()[2])<30\

```

```

and not (int(line.split()[2])>200\
and not (int(line.split()[3])<20\
and not (int(line.split()[3])>110:
    ArtPtemp.append(line.split()[0])
    ArtPtemp.append(line.split()[1])
    ArtPtemp.append(line.split()[2])
    ArtPtemp.append(line.split()[3])
    writeArtP.write(' '.join(ArtPtemp)+'\n')
    ArtPressure.append(' '.join(ArtPtemp))
else:
    ArtPtemp.append(line.split()[0])
    ArtPtemp.append(line.split()[1])
    ArtPtemp.append(line.split()[2])
    ArtPtemp.append(line.split()[3])
    discardedArtP.write(' '.join(ArtPtemp)+'\n')
writefivecolumns.write(line+'\n')
else:
    pass
writeArtP.close()
writePaP.close()
writeSpo2.close()
writetencolumns.close()
writeeightcolumns.close()
writefivecolumns.close()
discardedArtP.close()
discardedPaP.close()
discardedSpo2.close()
#print 'All processes completed'

#####
#####
Pig_ID=subject_id
#print "Uploading Arterial Pressure to database"
reconnect_mysql=1
if len(ArtPressure)>0:
    for line in ArtPressure:
        if(reconnect_mysql==1):
            connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=lhb)
            cursor = connection.cursor()
        else:
            pass
            Date_of_experiment = line.split()[0]
            Time_of_experiment = line.split()[1]
            Event_name=99
            Art_Sys = int(line.split()[2])
            Art_Dia = int(line.split()[3])
            MeanArtPres = float(Art_Dia) + ((float(Art_Sys) -
float(Art_Dia))/3)
            cursor.execute("""INSERT IGNORE INTO
monr_arterialpressure_realtime(Pig_ID,Date_of_experiment,Time_of_experiment,Origin
al_date,Original_time) VALUES
("%s", "%s", "%s", "%s", "%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Date_o
f_experiment,Time_of_experiment))
            cursor.execute("""UPDATE IGNORE
monr_arterialpressure_realtime SET Art_Sys = "%d",Art_Dia = "%d",MeanArtPres =
("%.2f" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND

```

```

Time_of_experiment='%s'""%(Art_Sys,Art_Dia,MeanArtPres,Pig_ID,Date_of_experiment,
Time_of_experiment))

    ##print Art_Sys,' ',Art_Dia,' ',MeanArtPres
    if(reconnect_mysql==1000):
        connection.commit()
        cursor.close()
        connection.close()
        reconnect_mysql=0
    else:
        pass
    reconnect_mysql=reconnect_mysql+1
if(cursor):
    cursor.close()
else:
    pass
if(connection):
    connection.commit()
    connection.close()
else:
    pass
##print "Art Pressure ",reconnect_mysql
else:
    pass
#####
#print "Uploading PA Pressure to database"
reconnect_mysql=1
if len(PaPressure):
    for line in PaPressure:
        if(reconnect_mysql==1):
            connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=ldb)
            cursor = connection.cursor()
        else:
            pass
            Date_of_experiment = line.split()[0]
            Time_of_experiment = line.split()[1]
            Event_name=99
            PA_Sys = int(line.split()[2])
            PA_Dia = int(line.split()[3])
            cursor.execute("""INSERT IGNORE INTO
monr_papressure_realtime(Pig_ID,Date_of_experiment,Time_of_experiment,Original_date,Original_time) VALUES
("%s","%s","%s","%s","%s")""%(Pig_ID,Date_of_experiment,Time_of_experiment,Date_of_experiment,Time_of_experiment))
            cursor.execute("""UPDATE IGNORE monr_papressure_realtime
SET PA_Sys = "%d",PA_Dia = "%d" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(PA_Sys,PA_Dia,Pig_ID,Date_of_experiment,Time_of_experiment))

        if(reconnect_mysql==1000):
            connection.commit()
            cursor.close()
            connection.close()
            reconnect_mysql=0
        else:
            pass
            reconnect_mysql=reconnect_mysql+1
            ##print "PA Pressure ",reconnect_mysql
            #####

```

```

        if(cursor):
            cursor.close()
        else:
            pass
        if(connection):
            connection.commit()
            connection.close()
        else:
            pass
    else:
        pass
    #print "Uploading Cardiac Output to database"
    reconnect_mysql=1
    if len(COlist):
        for line in COlist:
            if(reconnect_mysql==1):
                connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=ldb)
                cursor = connection.cursor()
            else:
                pass
            Date_of_experiment = line.split()[0]
            Time_of_experiment = line.split()[1]
            Event_name=99
            Cardiac_output = float(line.split()[2])
            cursor.execute("""INSERT IGNORE INTO
monr_cardiacoutput_realtime(Pig_ID,Date_of_experiment,Time_of_experiment,Original_
date,Original_time) VALUES
("%s", "%s", "%s", "%s", "%s")"""%(Pig_ID,Date_of_experiment,Time_of_experiment,Date_o
f_experiment,Time_of_experiment))
            cursor.execute("""UPDATE IGNORE
monr_cardiacoutput_realtime SET Cardiac_output = "%.1f" WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'"""%(Cardiac_output,Pig_ID,Date_of_experiment,Time_of_exper
iment))

            if(reconnect_mysql==1000):
                connection.commit()
                cursor.close()
                connection.close()
                reconnect_mysql=0
            else:
                pass
            reconnect_mysql=reconnect_mysql+1
            ##print "Cardiac Output ",reconnect_mysql

#####
        if(cursor):
            cursor.close()
        else:
            pass
        if(connection):
            connection.commit()
            connection.close()
        else:
            pass
    else:
        pass
    #print "Upload to database complete"

```

```

        else:
            pass
    else:
        pass

Table_list=['monr_arterialpressure_realtime','monr_papressure_realtime','monr_cardiacoutput_realtime']
    for table in Table_list:
        updateEventsandTimes(table)
    elif (re.match(r'^H',rawfile)) and not (re.match(r'^HT',rawfile)):
        if not (os.stat('./rawfilestore/'+rawfile).st_size==0):
            #try:
            #Define some permanent paths
            processflowpath=maindirectorypath+'St02/processflow/'
            removedtextpath=maindirectorypath+'St02/removedtext/'
            extractedoutputpath=maindirectorypath+'St02/extractedoutput/'
            rawdatapath=maindirectorypath+'St02/rawdata/'
            timefilepath=maindirectorypath+'St02/timefiles/'
            #####Extracting
Date#####
            splitfilename = rawfile.split("-")
            startDate = []
            endDate = []
            for value in range(1,4):
                if not (value==3):
                    endDate.append(splitfilename[value]) #File name consists of
the date the file has been created
                else:
                    day = int(splitfilename[value])+01
                    if (day<10):
                        daystring = "0"+str(day)
                    else:
                        daystring = str(day)
                    endDate.append(daystring)
            startDate.append(splitfilename[value])
            date1 = "-".join(startDate)
            date2 = "-".join(endDate)
            #####Date extraction
complete#####
            filestage1='firststage.txt'#type the name of the empty file within
quotes
            liststage1=[]#Empty list corresponding to the first stage

            #####
            corresponding_HT_filename=''
            extractData = open(tempfilestoredir+rawfile,'r')
            storerawdata=open(rawdatapath+rawfile,'a')
            writeData1 = open(processflowpath+filestage1,'a')
            unwantedlines = open( removedtextpath+'removed_text.txt','a')
            justonce=0
            machine_date=date1
            computer_date=date1
            machine_time=''
            computer_time=''
            for line in extractData:
                storerawdata.write(line)
                if re.match('^\\d\\d\\d\\d-\\d\\d-\\d\\d',line): #Match the lines with
time printed at the start of the line and write it to the firststage.txt

```

```

        if (justonce==0):
            if len(line.split()[1])>9:
                machine_time=line.split()[1][9:]
                machine_time=machine_time.strip(",")
            else:
                machine_time=line.split()[1]
                machine_time=machine_time.strip(",")
            writeData1.write(line)
            liststage1.append(line)
            justonce=1
            ##print "Machine time = "+machine_time
            for the_file in os.listdir('./rawfilestore'):
                if re.match(r'^HT-'+splitfilename[1]+'-'
'+splitfilename[2]+'-'+splitfilename[3],the_file):
                    corresponding_HT_filename=the_file

computer_time=open('./rawfilestore/'+the_file,'r').read()
                    storetimefile =
open(timefilepath+corresponding_HT_filename,'w')
                    storetimefile.write(computer_time)
                    storetimefile.close()
            else:
                pass
            ##print "Computer time = "+computer_time
        else:
            writeData1.write(line)
            liststage1.append(line)
    else:
        unwantedlines.write(line) #Write removed text to
removed_text.txt
    storerawdata.close()
    extractData.close()
    writeData1.close()
    unwantedlines.close()
    if not machine_time=='':
        machine_datetime=machine_date+' '+machine_time
        computer_datetime=computer_date+' '+computer_time
        ##print machine_datetime
        ##print computer_datetime
        fmt = '%Y-%m-%d %H:%M:%S'
        dt1 = datetime.datetime(*time.strptime(machine_datetime, fmt)[:6])
        dt2 = datetime.datetime(*time.strptime(computer_datetime,
fmt)[:6])

        timedifference=dt1-dt2
        ##print timedifference
        #print '1 out of 6 processes complete'

#####
    filestage2='secondstage.txt'#new file will be created here. So,
just give any name
    liststage2=[]

#####
    #shutil.copy2(filestage1,filestage2)
    sourceText1=[' ','--']
    replaceText1=[' ','0']

```

```

liststage2=replacemachine(liststage1,processflowpath+filestage2,sourceText1,replaceText1)
    #print '2 out of 6 processes complete'

#####
filestage3='thirdstage.txt'
unwantedlines = open( removedtextpath+'removed_text.txt','a')

#####
newfile = []
temp = []
writeData3 = open(processflowpath+filestage3,'a')
for line in liststage2:
    if not (int(line.split()[2]))== 0:
        writeData3.write(line)
        temp.append(line.split())
    else:
        unwantedlines.write(line)
writeData3.close()
unwantedlines.close()
#print '3 out of 6 processes complete'

#####
filestage4='fourthstage.txt'
liststage4=[]

#####
#temp = sorted(temp) #This is sorting the data in some weird
fashion so I am giving up on this
#I sorted in the way above so that I can use the RemoveDuplicates
function as I have defined.
#temp.sort()

liststage4=RemoveDuplicates(temp,filestage4,processflowpath,removedtextpath)
#print '4 out of 6 processes complete'

#####
filestage5='fifthstage.txt'
liststage5=[]

#####
writeData5 = open(processflowpath+filestage5,'a')
for line in liststage4:
    linesplit=line.split()
    temp1 = []
    for j in range(len(linesplit)):
        if(j==1) and (len(linesplit[j])>8):
            temp1.append(linesplit[j][9:])
        elif(j==2):
            pass
        else:
            temp1.append(linesplit[j])
    writeData5.write(' '.join(temp1)+'\n')
    liststage5.append(' '.join(temp1))
writeData5.close()
#print '5 out of 6 processes complete'

```

```

#####
        filestage6='sto2_thi.txt'
        sto2_thi=[]

#####

sto2_thi=modify_datetime(liststage5,filestage6,extractedoutputpath,timedifference)
        #print 'All processes complete'

#####
        Pig_ID=subject_id
        reconnect_mysql=1
        #print "Uploading StO2_15mm and THI_15mm to the database"
        """Below code should be uncommented in order to upload the data to
the database"""
        if len(sto2_thi)>0:
            for line in sto2_thi:
                if(reconnect_mysql==1):
                    connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=lhb)
                    cursor = connection.cursor()
                else:
                    pass
                    Date_of_experiment = line.split()[0]
                    Time_of_experiment = line.split()[1]
                    Event_name=99
                    StO2_15mm = int(line.split()[2])
                    THI_15mm = float(line.split()[3])
                    cursor.execute("""INSERT IGNORE INTO
monr_sto2thi_realtime(Pig_ID,Date_of_experiment,Time_of_experiment,Original_date,O
riginal_time) VALUES
('%s','%s','%s','%s','%s')""%(Pig_ID,Date_of_experiment,Time_of_experiment,Date_o
f_experiment,Time_of_experiment))
                    cursor.execute("""UPDATE IGNORE monr_sto2thi_realtime SET
StO2_15mm = "%d",THI_15mm = "%.1f" WHERE Pig_ID='%s' AND Date_of_experiment='%s'
AND
Time_of_experiment='%s'""%(StO2_15mm,THI_15mm,Pig_ID,Date_of_experiment,Time_of_e
xperiment))

                    if(reconnect_mysql==1000):
                        connection.commit()
                        cursor.close()
                        connection.close()
                        reconnect_mysql=0
                    else:
                        pass
                        reconnect_mysql=reconnect_mysql+1

#####
#####

        if(cursor):
            cursor.close()
        else:
            pass
        if(connection):
            connection.commit()
            connection.close()
        else:

```



```

        pass
    else:
        pass
    #print "Upload to database complete"

#####
else:
    pass
else:
    pass
Table_list=['monr_sto2thi_realtime']
for table in Table_list:
    updateEventsandTimes(table)
elif (re.match(r'^G',rawfile)) and not (re.match(r'^GT',rawfile)):
    if not (os.stat('./rawfilestore/'+rawfile).st_size==0):
        #try:
        #Define some permanent paths
        processflowpath=maindirectorypath+'BloodGases/processflow/'
        removedtextpath=maindirectorypath+'BloodGases/removedtext/'
        extractedoutputpath=maindirectorypath+'BloodGases/extractedoutput/'
        rawdatapath=maindirectorypath+'BloodGases/rawdata/'
        timefilepath=maindirectorypath+'BloodGases/timefiles/'
        #####Extracting
Date#####
        splitfilename = rawfile.split("-")
        startDate = []
        endDate = []
        timedifference=''
        for value in range(1,4):
            if not (value==3):
                endDate.append(splitfilename[value]) #File name consists of
the date the file has been created
            else:
                day = int(splitfilename[value])+01
                if (day<10):
                    daystring = "0"+str(day)
                else:
                    daystring = str(day)
                endDate.append(daystring)
            startDate.append(splitfilename[value])
        date1 = "-".join(startDate)
        date2 = "-".join(endDate)
        #####Date extraction
complete#####
        storerawdata=open(rawdatapath+rawfile,'a')
        filestage1 = 'firststage.txt'#type the name of the empty file within
quotes
        machine_time=''
        extractData = open(tempfilestoredir+rawfile,'r')
        createdirs = ['tempgfiles','arterial_gases','venous_gases']
        for the_dir in createdirs:
            try:
                if not os.path.exists(processflowpath+the_dir):
                    os.umask(0)
                    os.makedirs(processflowpath+the_dir,0771)
            else:
                pass

```

```

        except:
            pass

#####
    try:
        newfile=1
        for line in extractData:
            storerawdata.write(line)
            if newfile==1:

tempgfilename=open(processflowpath+'tempgfiles/'+line.split()[-1]+'.txt','a')
                tempgfilename.write(line)
                newfile=0
            else:
                if re.search(r'_',line):
                    try:
                        tempgfilename.close()
                    except:
                        pass
                    newfile=1

                else:
                    tempgfilename.write(line)
    except:
        pass
    try:
        tempgfilename.close()
    except:
        pass
    extractData.close()
    storerawdata.close()
    temparray=[]
    sortedtemparray=[]
    firstday=date1.replace('-', '')
    nextday=date2.replace('-', '')
    machine_time_find=[]
    for thefile in os.listdir(processflowpath+'tempgfiles'):
        if re.match(r'^'+firstday,thefile) or
re.match(r'^'+nextday,thefile):
            machine_time_find.append(thefile)
            readthefile =
open(processflowpath+'tempgfiles/'+thefile,'r').read()
            if re.search(r'DriftB',readthefile) or
re.search(r'<html>',readthefile):

shutil.copy2(processflowpath+'tempgfiles/'+thefile,removedtextpath+thefile)
                os.remove(processflowpath+'tempgfiles/'+thefile)
            else:
                temparray.append(thefile)
    else:

shutil.copy2(processflowpath+'tempgfiles/'+thefile,removedtextpath+thefile)
        os.remove(processflowpath+'tempgfiles/'+thefile)
        sortedmachinetimearray=sorted(machine_time_find)
        sortedtemparray=sorted(temparray)
        if len(sortedmachinetimearray)>0:
            machinetimestring=sortedmachinetimearray[0].split('.')[0][8:]

```

```

machine_time=machinetimestring[0:2]+'.'+machinetimestring[2:4]+'.'+machinetimestri
ng[4:]
        #print machine_time
    else:
        pass
    if len(sortedtemparray)>0:
        for thefile in os.listdir(processflowpath+'tempgfiles'):
            ##print thefile
            readthelines = open(processflowpath+'tempgfiles/'+thefile,'r')
            i=0
            try:
                for line in readthelines:
                    i=i+1
                    if (i==3)and re.search(r'A$',line):
                        readthelines.close()

shutil.copy2(processflowpath+'tempgfiles/'+thefile,processflowpath+'arterial_gases
/'+thefile)
                    os.remove(processflowpath+'tempgfiles/'+thefile)
                    elif (i==3)and re.search(r'V$',line):
                        readthelines.close()

shutil.copy2(processflowpath+'tempgfiles/'+thefile,processflowpath+'venous_gases/'
+thefile)
                    os.remove(processflowpath+'tempgfiles/'+thefile)
            else:
                pass
        except:
            pass
        #print "1 out of 6 processes completed"

#####
        corresponding_GT_filename=''
        machine_date=date1
        computer_date=date1
        computer_time=''

#####
        for the_file in rawfilestore:
            if re.match(r'^GT-'+splitfilename[1]+'-'+splitfilename[2]+'-
'+splitfilename[3]+'-'+splitfilename[4]+'-'+splitfilename[5],the_file):
                corresponding_GT_filename=the_file
                computer_time=open(tempfilestoredir+the_file,'r').read()
                storetimefile =
open(timefilepath+corresponding_GT_filename,'w')
                storetimefile.write(computer_time)
                storetimefile.close()
            else:
                pass
        machine_datetime=machine_date+' '+machine_time
        computer_datetime=computer_date+' '+computer_time
        #print machine_datetime
        #print computer_datetime
        fmt = '%Y-%m-%d %H:%M:%S'
        dt1 = datetime.datetime(*time.strptime(machine_datetime, fmt)[:6])
        dt2 = datetime.datetime(*time.strptime(computer_datetime,
fmt)[:6])

```



```

        Glucose='NONE'
        Glucose=str(Glucose)
elif re.search(r'K\+',line.split()[2]):
    try:
        Potassium=float(line.split()[3])
    except:
        Potassium='NONE'
        Potassium=str(Potassium)
elif re.search(r'Lac',line.split()[2]):
    try:
        Lactate=float(line.split()[3])
    except:
        Lactate='NONE'
        Lactate=str(Lactate)
elif re.search(r'Na\+',line.split()[2]):
    try:
        Sodium=int(line.split()[3])
    except:
        Sodium='NONE'
        Sodium=str(Sodium)
elif re.search(r'SO2',line.split()[2]):
    try:
        Arterial_O2_sat=int(line.split()[3])
    except:
        Arterial_O2_sat='NONE'
        Arterial_O2_sat=str(Arterial_O2_sat)
elif re.search(r'THb',line.split()[2]):
    try:
        Hb=float(line.split()[3])
    except:
        Hb='NONE'
        Hb=str(Hb)
elif re.search(r'pCO2',line.split()[2]):
    try:
        Arterial_PC02=int(line.split()[3])
    except:
        Arterial_PC02='NONE'
        Arterial_PC02=str(Arterial_PC02)
elif re.search(r'pCO2(T)',line.split()[2]):
    try:
        Arterial_PC02=int(line.split()[3])
    except:
        Arterial_PC02='NONE'
        Arterial_PC02=str(Arterial_PC02)
elif re.search(r'pH',line.split()[2]):
    try:
        Arterial_pH=float(line.split()[3])
    except:
        Arterial_pH='NONE'
        Arterial_pH=str(Arterial_pH)
elif re.search(r'pH(T)',line.split()[2]):
    try:
        Arterial_pH=float(line.split()[3])
    except:
        Arterial_pH=' '
        Arterial_pH=str(Arterial_pH)
elif re.search(r'pO2',line.split()[2]):
    try:

```



```

extractData = open(cleanedvenous,'r').readlines()
for line in extractData:
    i=i+1
    if (len(line.split())>3):
        if re.search(r'SO2',line.split()[2]):
            try:
                Venous_O2_sat=int(line.split()[3])
            except:
                Venous_O2_sat='NONE'
                Venous_O2_sat=str(Venous_O2_sat)
        elif re.search(r'pO2',line.split()[2]):
            try:
                Venous_P02=int(line.split()[3])
            except:
                Venous_P02='NONE'
                Venous_P02=str(Venous_P02)
        elif re.search(r'pO2(T)',line.split()[2]):
            try:
                Venous_P02=int(line.split()[3])
            except:
                Venous_P02='NONE'
                Venous_P02=str(Venous_P02)
        ##print ((line.split()[2]).lower()).replace(' ','')
    else:
        pass
    venous_processed.write(thedate+' '+thetimepoint+'
'+Venous_O2_sat+' '+Venous_P02+' '+str(Event_name)+'\n')
    venous_processed_list.append(thedate+' '+thetimepoint+'
'+Venous_O2_sat+' '+Venous_P02+' '+str(Event_name))
    ##print 'Venous ',Event_name
    venous_processed.close()
    #print '4 out of 6 processes completed'

#####
#####
    arterial_processed_final='arterial_processed_final.txt'
    arterial_processed_final_list=[]
    venous_processed_final='venous_processed_final.txt'
    venous_processed_final_list=[]

arterial_processed_final_list=modify_datetime(arterial_processed_list,arterial_pro
cessed_final,extractedoutputpath,timedifference)

venous_processed_final_list=modify_datetime(venous_processed_list,venous_processed
_final,extractedoutputpath,timedifference)
    #print '5 out of 6 processes completed'

#####
#####
    Pig_ID=subject_id
    #print "Uploading arterial blood gas data to database"
    reconnect_mysql=1
    if len(arterial_processed_final_list)>0:
        for line in arterial_processed_final_list:
            if(reconnect_mysql==1):
                connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=lhb)
                cursor = connection.cursor()

```

```

else:
    pass
Date_of_experiment = line.split()[0]
Time_of_experiment = line.split()[1]
try:
    Event_name=int(line.split()[12])
except:
    Event_name=99
try:
    Calcium=float(line.split()[2])
except:
    Calcium=''
try:
    Glucose=int(line.split()[3])
except:
    Glucose=''
try:
    Potassium=float(line.split()[4])
except:
    Potassium=''
try:
    Lactate=float(line.split()[5])
except:
    Lactate=''
try:
    Sodium=int(line.split()[6])
except:
    Sodium=''
try:
    Arterial_O2_sat=int(line.split()[7])
except:
    Arterial_O2_sat=''
try:
    Hb=float(line.split()[8])
except:
    Hb=''
try:
    Arterial_PC02=int(line.split()[9])
except:
    Arterial_PC02=''
try:
    Arterial_pH=float(line.split()[10])
except:
    Arterial_pH=''
try:
    Arterial_PO2=int(line.split()[11])
except:
    Arterial_PO2=''
cursor.execute("""INSERT IGNORE INTO
monr_arterial_bloodgases_realtime(Pig_ID,Date_of_experiment,Time_of_experiment,Original_date,Original_time) VALUES
("%s","%s","%s","%s","%s")"""%(Pig_ID,Date_of_experiment,Time_of_experiment,Date_of_experiment,Time_of_experiment))
cursor.execute("""UPDATE monr_arterial_bloodgases_realtime
SET Calcium="%s",Glucose="%s",Potassium="%s",Lactate =
"%s",Sodium="%s",Arterial_O2_sat="%s",Hb="%s",Arterial_PC02="%s",Arterial_pH="%s",
Arterial_PO2="%s" WHERE Pig_ID='%s' AND Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Calcium,Glucose,Potassium,Lactate,Sodium,Arterial_O2_s

```



```

at,Hb,Arterial_PC02,Arterial_pH,Arterial_PO2,Pig_ID,Date_of_experiment,Time_of_exp
eriment))

        if(reconnect_mysql==1000):
            connection.commit()
            cursor.close()
            connection.close()
            reconnect_mysql=0
        else:
            pass
        reconnect_mysql=reconnect_mysql+1
    if(cursor):
        cursor.close()
    else:
        pass
    if(connection):
        connection.commit()
        connection.close()
    else:
        pass
else:
    pass

#####
#####
        #print "Uploading venous blood gas data to database"
        reconnect_mysql=1
        if len(venous_processed_final_list)>0:
            for line in venous_processed_final_list:
                if(reconnect_mysql==1):
                    connection =
sql.connect(host=lhost,port=lport,user=luser, passwd=lpasswd,db=ldb)
                    cursor = connection.cursor()
                else:
                    pass
                Date_of_experiment = line.split()[0]
                Time_of_experiment = line.split()[1]
                try:
                    Event_name=int(line.split()[4])
                except:
                    Event_name=99
                try:
                    Venous_O2_sat=int(line.split()[2])
                except:
                    Venous_O2_sat=''
                try:
                    Venous_PO2=int(line.split()[3])
                except:
                    Venous_PO2=''
                cursor.execute("""INSERT IGNORE INTO
monr_venous_bloodgases_realtime(Pig_ID,Date_of_experiment,Time_of_experiment,Original_date,Original_time) VALUES
('%s','%s','%s','%s','%s')"""%(Pig_ID,Date_of_experiment,Time_of_experiment,Date_of_experiment,Time_of_experiment))
                cursor.execute("""UPDATE monr_venous_bloodgases_realtime
SET Venous_O2_sat='%s',Venous_PO2='%s' WHERE Pig_ID='%s' AND
Date_of_experiment='%s' AND
Time_of_experiment='%s'""%(Venous_O2_sat,Venous_PO2,Pig_ID,Date_of_experiment,Time_of_experiment))

```

```

        if(reconnect_mysql==1000):
            connection.commit()
            cursor.close()
            connection.close()
            reconnect_mysql=0
        else:
            pass
        reconnect_mysql=reconnect_mysql+1
    if(cursor):
        cursor.close()
    else:
        pass
    if(connection):
        connection.commit()
        connection.close()
    else:
        pass
else:
    pass
else:
    pass
Table_list=['monr_arterial_bloodgases_realtime','monr_venous_bloodgases_realtime']
for table in Table_list:
    updateEventsandTimes(table)
else:
    pass

```