CISC 498:
Information Technology Project (2016-17)

Final Documentation

Smith Sustainability Studies System

Team 3 members: Emma Wong (10084142), Gabriel Bufardeci (10100032),
Seema Hejazi (10050728) and Dylan Liu (10107580)

Supervisor: Juergen Dingel

Customers: Jane Webster and Sandy Staples

Purpose of this Document

The purpose of this document, as well as the additional resources provided, is to aid in operating and maintaining the Smith Sustainability Studies System. Along with this document, we have provided a test suite which was used to test the web application, in addition to the "Host your Application in The Amazon Cloud with XAMPP and Bitnami" PDF that will be needed when maintaining and migrating the system. It contains a helpful step-by-step guide for certain procedures.

Introduction

Our customers, Jane Webster and Sandy Staples are interested in implementing Green Information Systems which use Information Systems to improve the eco-sustainability of businesses and society. The purpose of the system is for researchers to run ongoing studies related to pro-environmental behaviour to better understand what stimuli could reduce personal energy consumption from electronics in the workplace. The Smith Sustainability Studies System is a user-friendly web-application which will assist them in the creation and conduction of studies by offering administrative control over the design of each study, the creation and management of participants' accounts, the ability to monitor study progress, extract and analyse data, and finally provide interactive material in the form of persuasive information and reports of energy usage. Study participants, referred to as users, can input daily data as well as track their progress and standings in the system depending on their constraints in the study.

User Manual

Definitions

Super Admin: Jane Webster and Sandy Staples, who are in charge of conducting sustainability studies using this system, as well as any tech team that may be maintaining and administering the application server (see more about application server in Maintenance Suggestions). Super Admins have all of the privileges of Admins, but with access to all studies and the ability to create Admin and Super Admin accounts.

Admin: Any researchers or assistants designated by Super Admins who is using this system to conduct their own studies or aid with a study they are assigned. They are able to create a study, monitor progress and actions of users in their study and interact with the participants.

User: As participants of a study, each user is assigned a condition group allowing them access to functionality within the Admin set rules. Their capabilities include inputting data, monitoring their progress and actions and interacting with users and admins in their study through community posts.

Study: The Smith Sustainability Studies System is to be used to determine what methods of interaction are most effective in improving environmental sustainability in the workplace. Super Admins will choose which studies to conduct and administer them.

Condition Group: Analogous with condition group in the scientific sense, it is a method for separating Users into control groups and experimental groups.

Phases: Users in a condition group will go through various phases for which, each will have a different set of permissions so that the effectiveness of those permissions may be tested.

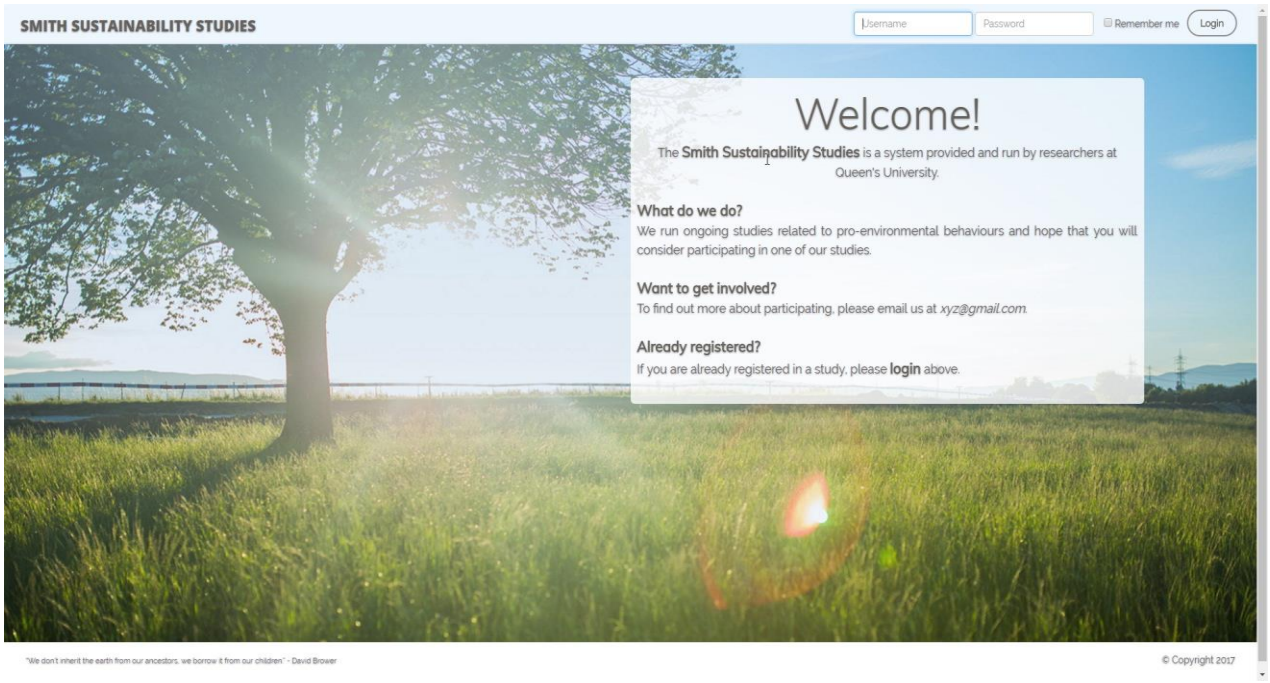
Permissions: These are mechanisms for interacting with the Users of the system such as allowing the viewing of statistics or community posts. These are explained in detail in the Create Study section of this document.

Sub-team: Within a condition group, Users may belong to separate teams. This is meant to encourage competitiveness within the condition groups.

Page Structure

Welcome

The first page anyone will be met with is the **Welcome** page. If somebody is not yet in a study and is interested in participating, they can contact the administrators to learn more about the program and how to get involved. Otherwise, a current user of the system can input their given username and password to login.

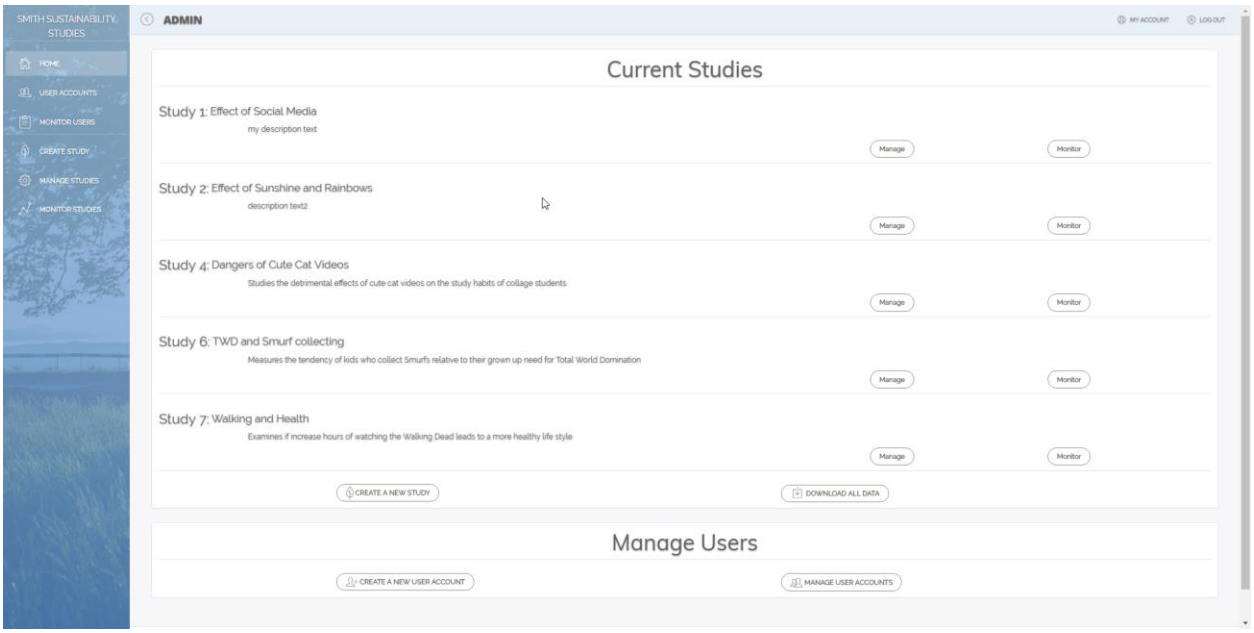


After logging in, if you are a regular Admin, the landing page will be the same as if you were a Super Admin, but with some view and functional limitations. This will be explained further in the **Admin** section of this document. If you are a User however, you will be greeted with an alternate landing page, described in detail in the **User** section of this document.

Super Admin

Home:

The page that a Super Admin is greeted with is shown below. The page is divided into sections as follows: **Current Studies** and **Manage Users**.



In the **Current Studies** section, you will be able to see **all** of the studies that are currently running in the system. For each of them you will be given the option to either **Manage** or **Monitor** the study. There is also the option to **Create New Study** as well as **Extract Raw Data** for all of the studies that are currently running. **Extract Raw Data** will simply fetch all of the available studies' information and export it as a formatted CSV file and **Create New Study** will redirect you to the **Create Study** page.

If **Manage** is selected for a given study, you will be redirected to the **Manage Studies** page, with the details of the selected study expanded by default. If **Monitor** is selected for a given study, similar to the previous case, then you will be redirected to the **Monitor Studies** page with the selected study having its information previously expanded on page load.

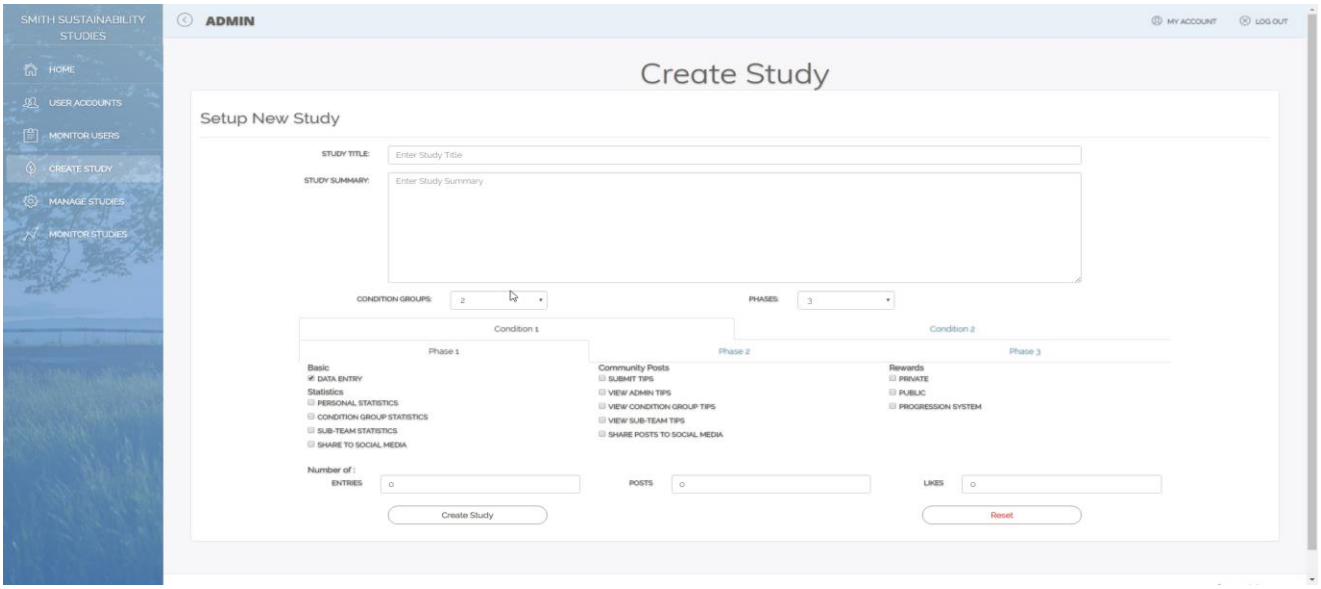
As for the Manage Users section, there are the options to either **Create Account**, or **Manage User Accounts** for the existing users. If **Create Account** is selected, the Admin will be redirected to the **User Accounts** page, with the **Create Account** form expanded by default. If **Manage User Accounts** is selected however, you will simply be redirected to the default **User Accounts** page.

Create Study:

On the **Create Study** page, there are multiple sections to fill out. There are the **Study Title** and **Study Summary** sections which are simple ways to differentiate various studies. There

are also spaces to enter both the number of **Condition Groups** as well as the number of **Phases** for the study being created.

Below those fields is a set of tabs that contain the application's gamification elements. This is the foundation of the system as it allows the various gamification mechanisms to be toggled on or off in order to conduct a given study. For each condition, and each phase in those conditions, the Admin can select the checkboxes to enable certain gamification elements.



Mechanism definitions:

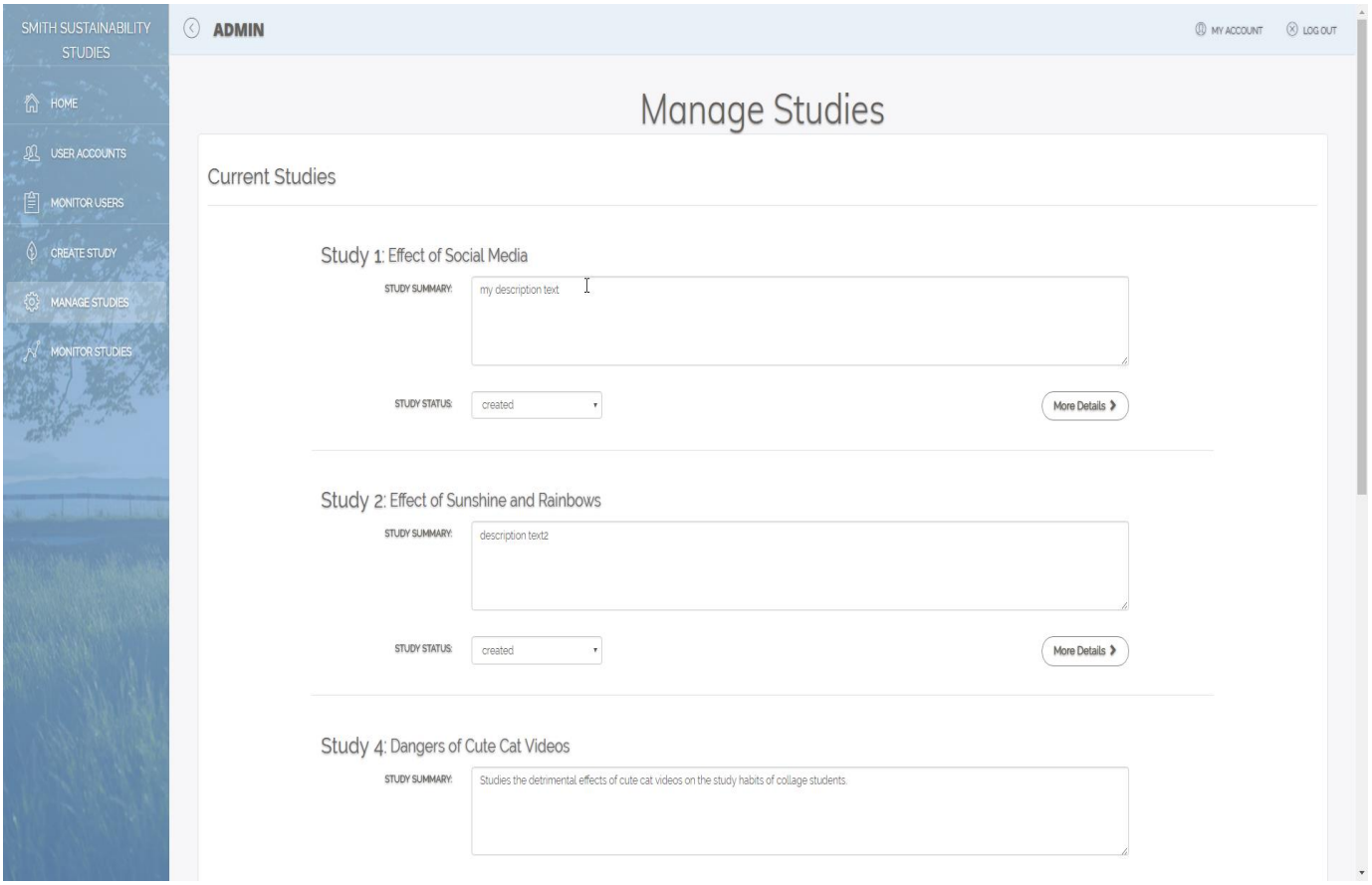
- *Data Entry*
    - *The ability for users in the study to input their daily energy usage amounts. This option is recommended to be enabled for most studies as this is the basis for each study being conducted.*
- *Personal Statistics*
    - *Toggles the ability for a User to observe their own statistics, including their past entries as well as a trend for those entries.*
- *Condition Group Statistics*
    - *Toggles a section on the User's **Statistics** page, which shows their rank against the rest of the users in the current study.*
- *Sub-Team Statistics*

    - *Enables/disables the ability for a User to see the score of their sub-team compared to the other sub-team in the condition group on their **Statistics** page.*
- *Submit Tips*
    - *Toggles the ability for users in the study to send their personal tips on saving energy to the super admins on the **Community Posts** page.*
- *View Admin Tips*
    - *Allows users in the study to view energy saving tips from the admins on their **Community Posts** page.*
- *View Condition-Group Tips*
    - *Toggles the option for a User to view tips given by other users in their given condition group on the **Community Posts** page.*
- *View Sub-Team Tips*
    - *Allows users in the study to view energy saving tips from the users in their sub-team by going to their **Community Posts** page.*
- *Private Rewards*
    - *Toggles the availability of rewards which users in the study have the ability to unlock for completing certain tasks.*
- *Public Rewards*
    - *Toggles the option of rewards that are available to each condition group in the given study.*

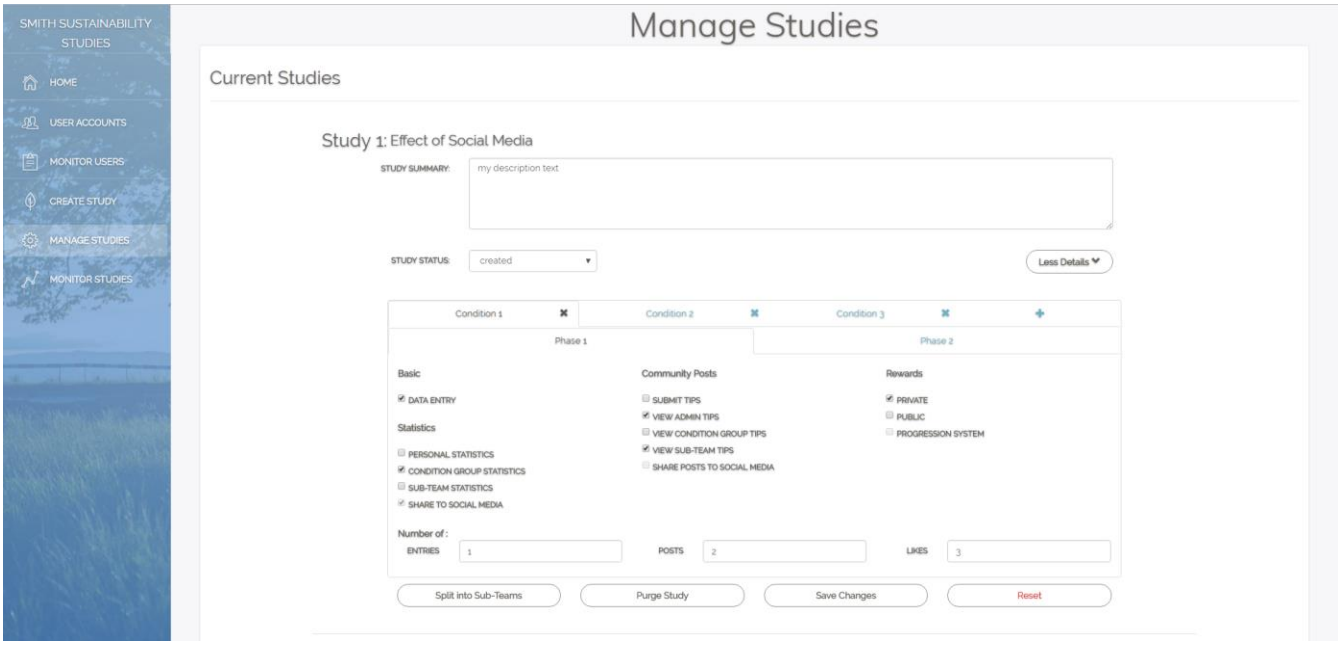
Finally, at the bottom of the page, there are buttons to either **Save** the study that has currently been configured, or to **Reset Fields** that have been filled out so far in order to easily start again. Once the study has been saved, it will be in effect in the system, and an Admin may begin adding users to it.

Manage Studies:

On the **Manage Studies** page, the Admin will be greeted with a list of all studies currently running in the system. For each study, there is the option to expand it, using the **Details** button.



Once a given study is expanded, the study's name and description will be displayed as well as the permission tabs, similar to that of the **Create Study** page, with the checkboxes reflecting the permissions currently enabled in the study.

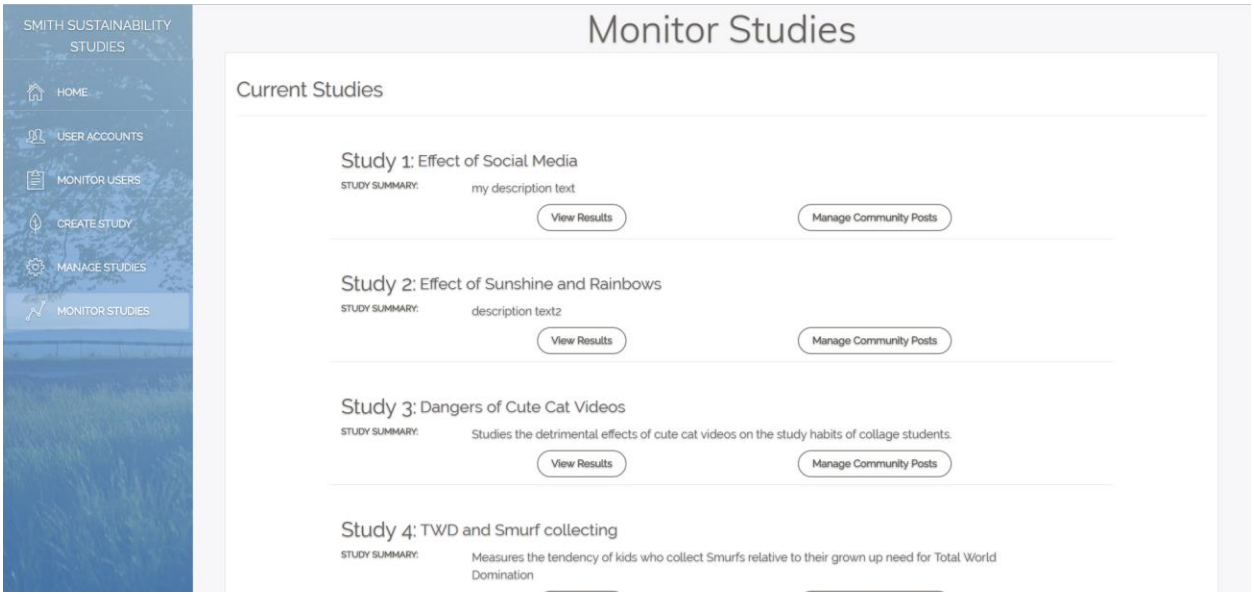


Here, all the fields available in the **Create Study** page will have the option to be changed, as well as a new option called **Study Status** which tells the system whether the study has newly been created, if it is currently active, or if it is an old study that is simply archived.

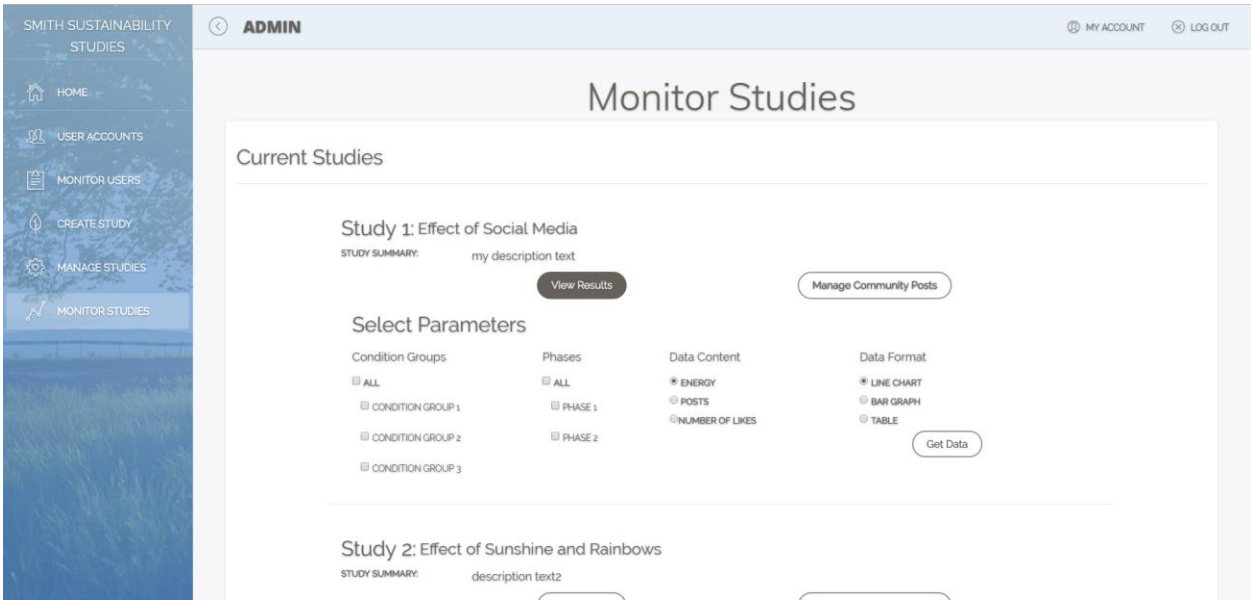
As for the buttons at the bottom of the form, other than the options to **Save Changes**, and to **Reset** the fields, there is also the option for the current study to be **Split Into Sub-Teams** as well as **Purge Study** which will completely delete it from the system. Finally, there is also the option to add condition groups using the **+** button on the condition groups tab bar.

Monitor Studies:

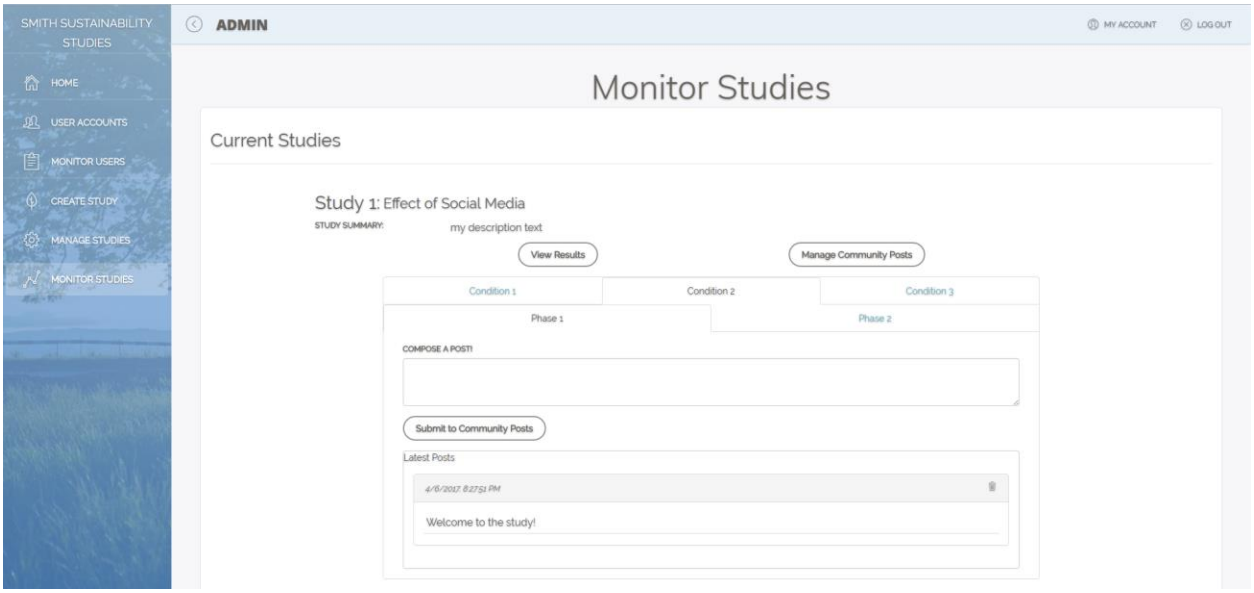
On the **Monitor Studies** page, the Admin will be able to select either **View Results**, or **Manage Community Posts** for each study in effect.



View Results will expand to a set of parameters that the Admin must select, so that the selected data may be presented in a pretty, graphical manner, customized to the selected options.

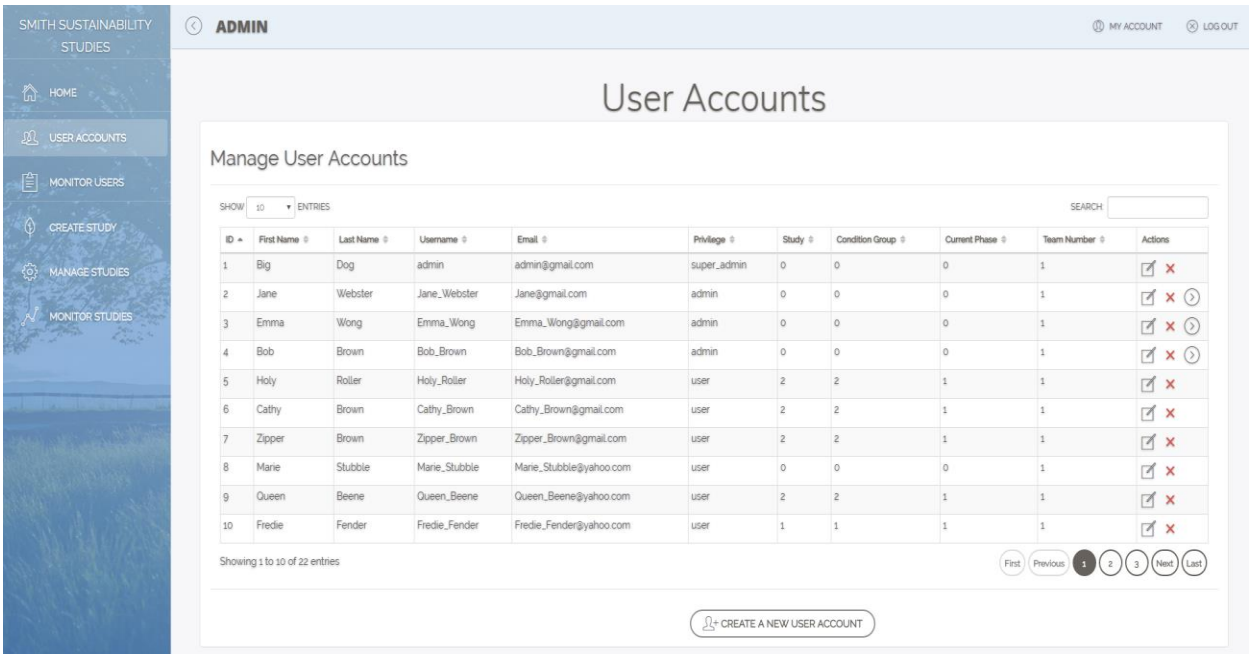


Finally, **Manage Community Posts** will expand to display the latest posts in the study, for the selected condition group and phase, and will also give the option to create a new post for the given condition group and phase, in the selected study. The Admin may also delete posts by clicking on a trash can icon.

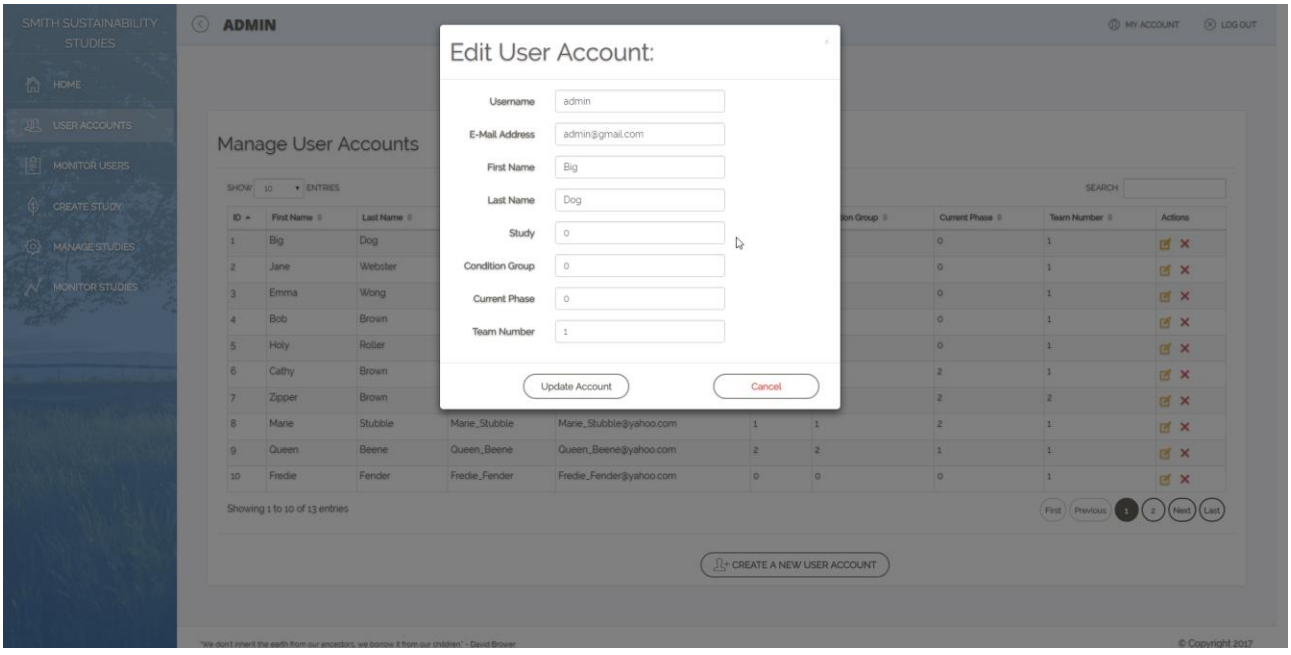


User Accounts

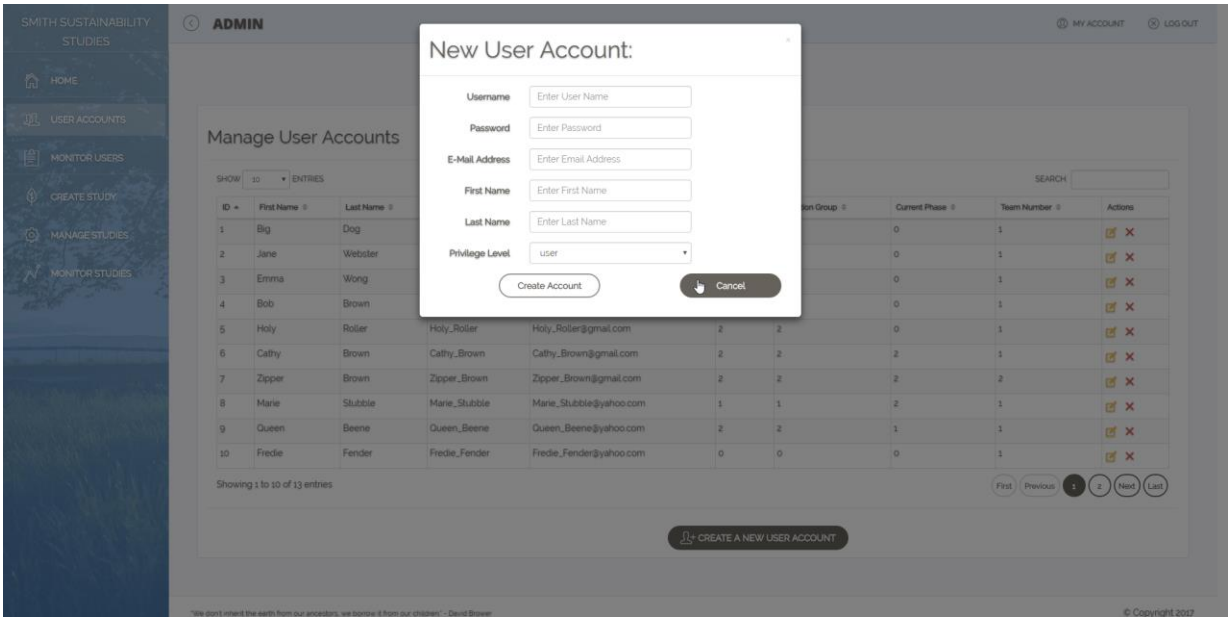
If an Admin wants to check details about the system's users, they can go to the **User Accounts** page. This page contains a table of users, along with the following information presented: user ID, first and last name, username, email, study ID, condition group number and the current phase they are in and finally, their team number. Keep in mind that if two users have the same condition group number, it does not necessarily mean that they are in the same condition group. For this to be true, they would also have to share their study ID. This is the same for their current phase number as well as their team number. If a condition group has not been split into teams, then the default team for each user is Team 1.



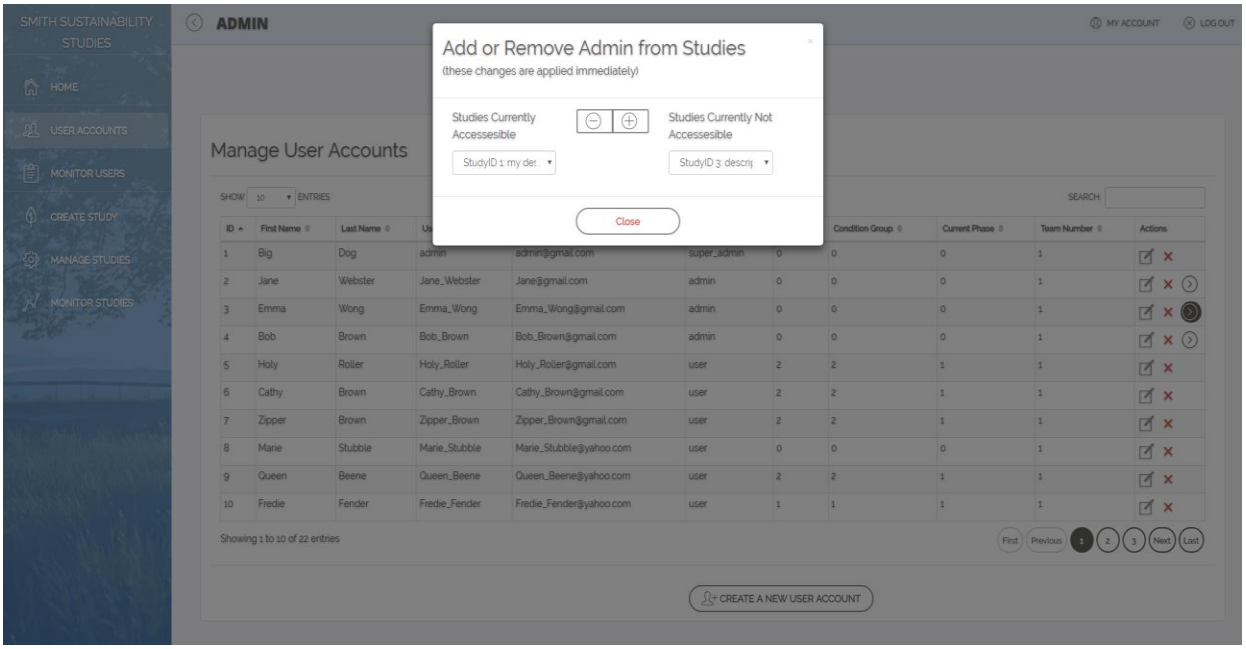
Any User (including Admins) can be removed from the system by simply clicking the ✕ icon and confirming that deleting the user is what you really want to do. There is also the option to **Edit a User** by clicking on the icon. This will open the following pop-up. Here the Admin will be able to edit any of the information that is shown in the table. When a study is complete, the Admin can move the users to other studies and reset their progress.



This page also contains the **Create a New User Account** form to add users to the system. A default username and password is entered, as well as the User's full name and their e-mail address and finally the **privilege level** for that User. The **privilege level** attribute corresponds to the Super Admin, Admin and User discussed in this document.

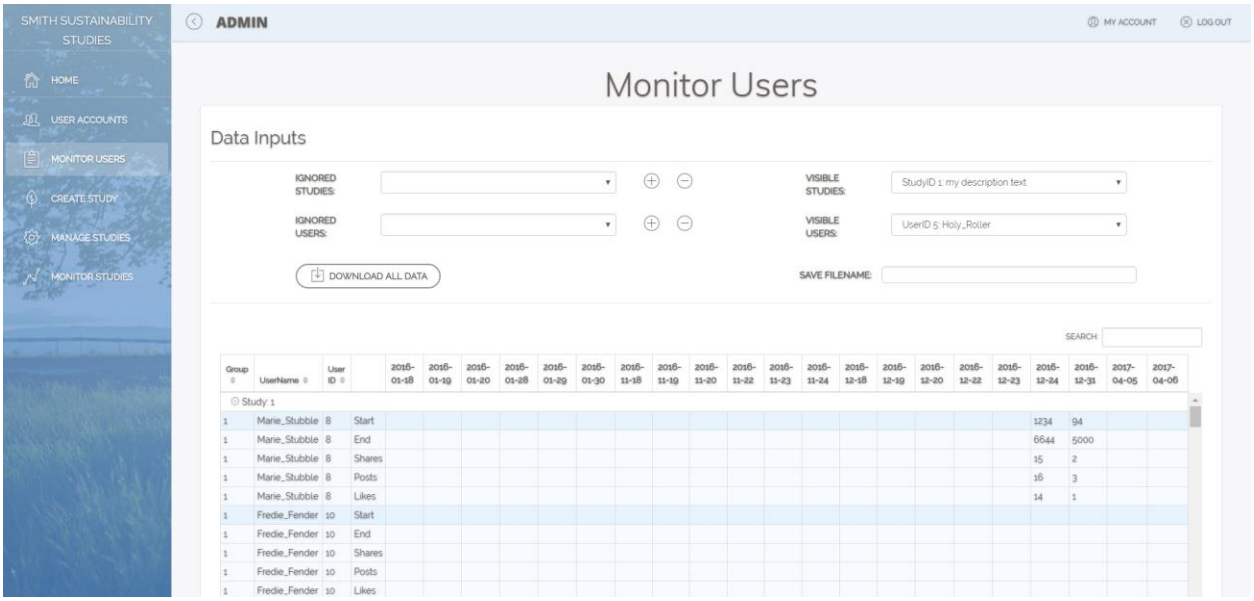


A Super Admin can also add and remove Admins from studies.



Monitor Users

Where the **User Accounts** page was geared towards the user's information in the system, the **Monitor Users** page is where an Admin would go if they were looking for detailed information about users in regards to the studies that they are participating in. In the displayed table, an Admin can view the study ID, group number, user ID, as well as the given User's entries so far. Furthermore, for each User, their starting and ending energy amounts are displayed as well as their shares and posts on the system.



Furthermore, at the top of the page, there are options to customize what is visible in the table. The +/- buttons add and remove studies and users, respectively. The boxes on the left hand side of the buttons show which studies/users have been removed from the table. The boxes on the left of the buttons show which studies/user are currently being shown in the table. An Admin can download the data in the table by specifying the filename and giving it a file type - Note: only CSV and txt file extensions are supported. If no file extension is provided, the website will export a txt file.

Admin

The Admin role in this system is quite similar to the Super Admin role. They can do almost everything that a Super Admin can do, with a few caveats, but the most restriction is present on what the Admin can or cannot see. All of the properties of a Super Admin above apply to a regular Admin except for the details mentioned here.

User Accounts and Monitor Users:

A regular Admin may view user accounts in both the **User Accounts** page as well as the **Monitor Users** page. However, Admin will only be shown the user accounts they have created or those within studies which they are assigned to access.

Admins may also create user accounts, but are not be able to create a Super Admin account or an Admin account. In other words, Admins can only create basic User accounts.

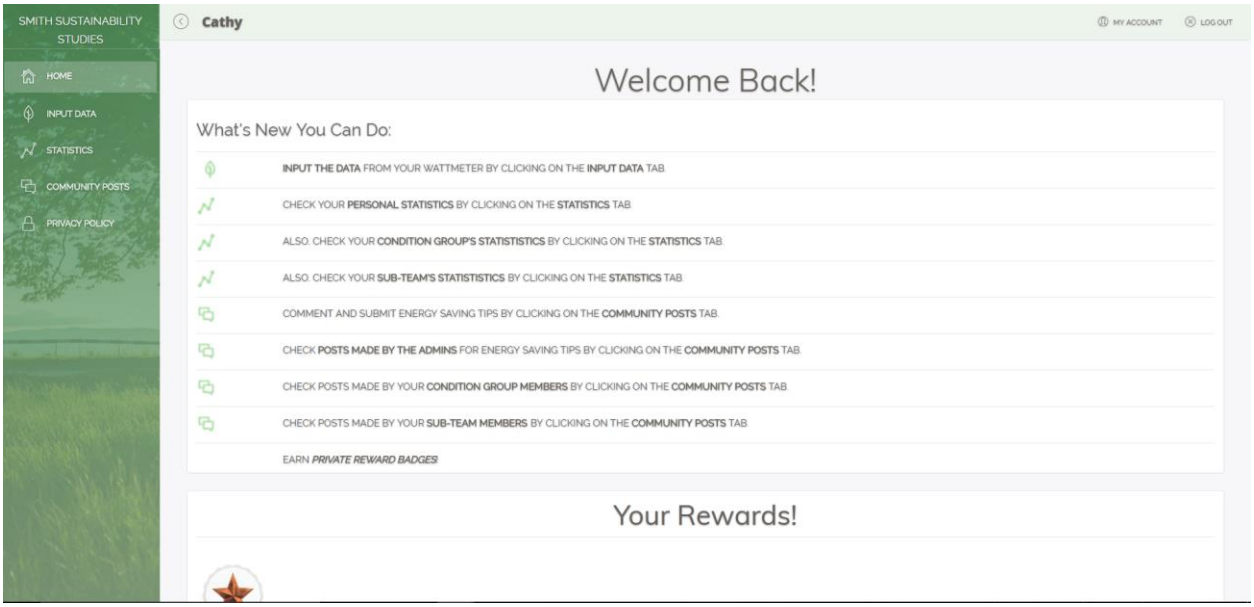
Manage and Monitor Studies:

Admins receive most of the functionality of the **Manage Studies** page and the **Monitor Studies** page, but they will only be able to see the studies of which they are involved in. Furthermore, they will not be given the option to purge a study from the system, this can only be done by a Super Admin.

User

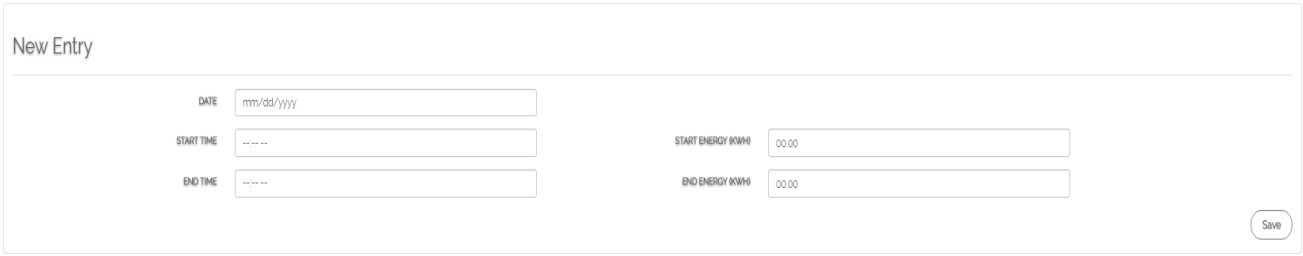
Home

The first page any user will be greeted with is the User **Home** page. It contains a **Welcome Back** section as well as a **Your Rewards** section. The **Welcome Back** section of the page gives tips to the User on what to do next, such as inputting data on the **Input Data** page or checking statistics on their **Statistics** page. Furthermore, the **Your Rewards** section simply has badges corresponding to certain “challenges” that can be completed in the system, such as reaching a particular number of entries or liking a certain number of community posts tips.

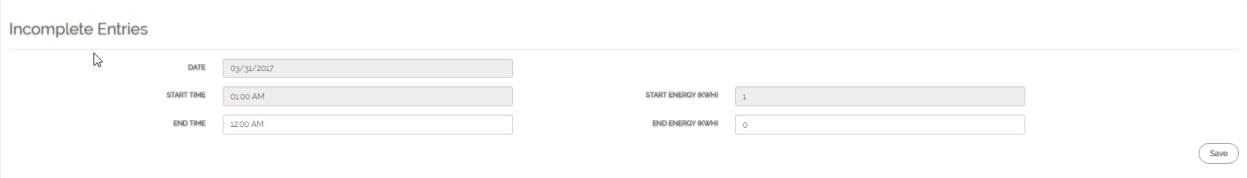


Input Data

The most common activity, most Users will need to do, is to input their wattmeter readings, twice a day. For each day that the User is in a phase, so long as data input is enabled in that phase, they will be required to enter a Date, Start time and Energy as well as an End time, and Energy. Energy will be measured in KWH and read off of the User's wattmeter which should be installed beforehand.

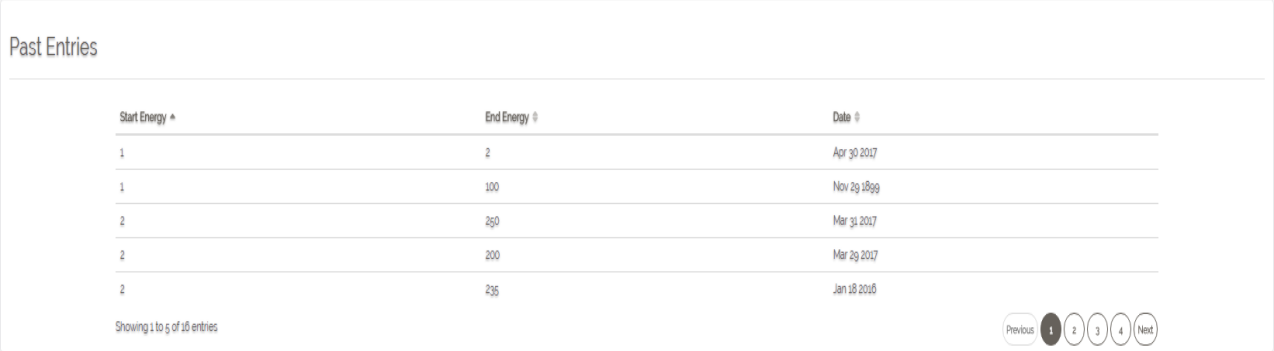


At the start of the day the user enters their starting data and can press save, which will send the unfinished entry to the **Incomplete Entries** section below. This will be stored there until they return at end of day to complete the data entry and save it to the database. For days that the User has forgotten or was not able to input their closing energy levels, the lower **Incomplete Entries** section will have them listed. The User may simply complete the form and submit it to catch up on their entries.

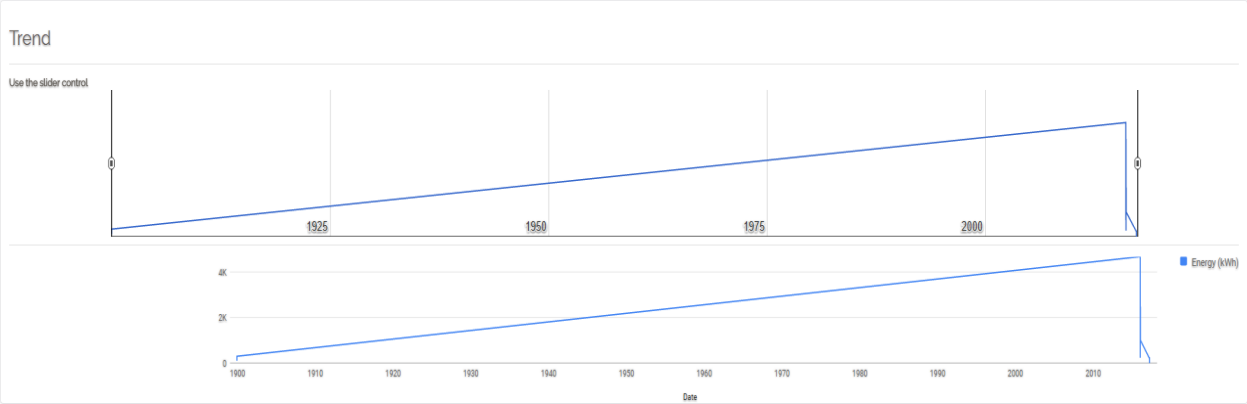


Statistics

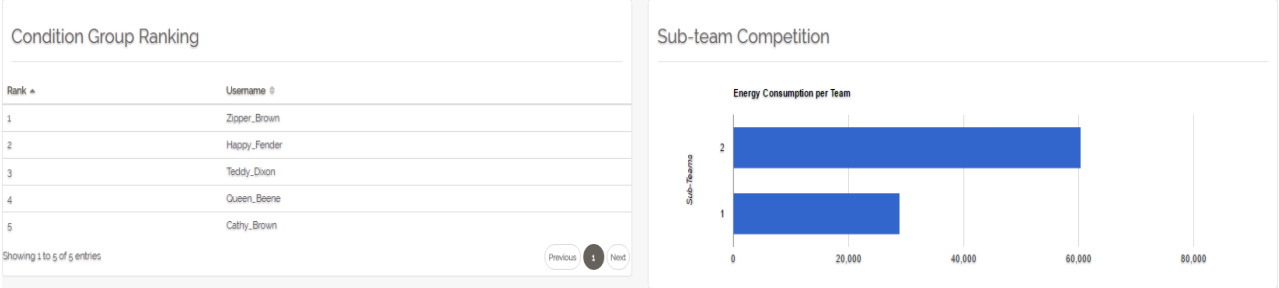
On the **Statistics** page, there are a number of sections that can be toggled through permissions. The first of which is the **Past Entries** section. It simply lists the User's entries into the system so far, showing the **Start Energy** levels**, End Energy** levels as well as their **Date**, in three columns, on multiple pages.



The next section is the **Trend** graph. Here, the User will be able to look at their **Energy** levels, against each **Date**, in the form of a graph instead of a table. The User can also use the slider to examine a specific part of his or her trend, or to focus on a particular time span.

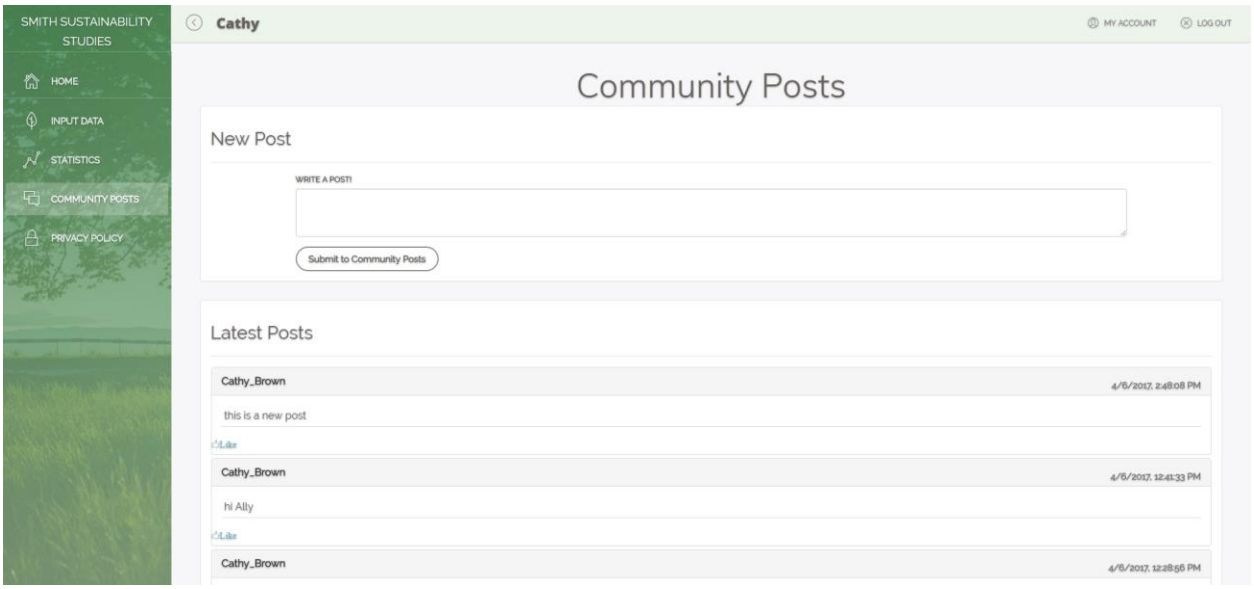


The final section on the **Statistics** page is related to a User's ranking. There is a **Condition Group Ranking** table, displaying the **Usernames** of the other Users in the condition group, as well as their rank in relation to how much energy the other members in the condition group have been using. There is also the **Sub-Team Competition** graph that simply shows the comparison of total energy used, per sub-team.



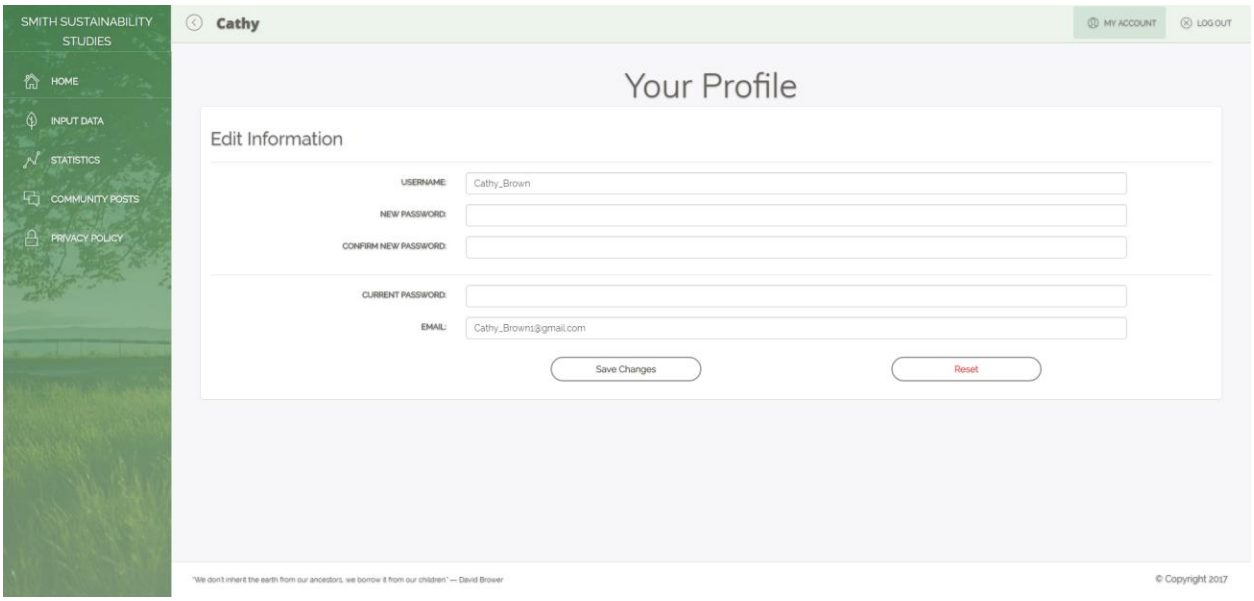
Community Posts

When a User visits the **Community Posts** page, they will be able to do two things. First, they will be able to write a post. Once a post is written, and sent, it will be added to the next section of the **Community Posts** page. This section is where the User may view all posts that have been made so far. It is called the **Latest Posts** section.



My Account

This page is identical to all privilege levels and is the same for anyone in the system. The **My Account** page simply contains a form for the user to change their password or email by inputting their **Username,** and **Current Password** as well as a **New Password,** which they must **Confirm** and finally their **Email Address**. If the user wants his or her username changed, he or she must contact an admin to make the change.



Maintenance Suggestions

Previous qualifications

Anyone who has worked in web development, should have no problem making changes to the Smith Sustainability Studies System codebase. The code is modular enough for anyone to make sense of each part, and things related to each other, more-or-less are located in the same place. However, it is possible for someone new to web development to make changes, so long as they do so carefully and reference the appropriate documents. Although, for anything slightly complex, it is recommended to have an experienced member on the team (i.e. someone who knows their way around Javascript/jQuery, PHP and SQL) to offer their assistance. In either case, if changes are going to be made to the system, the following information will be useful.

The Architecture

This section will give a brief overview of our architecture. Please refer to the Internal Design document for a more detailed breakdown. The Internal Design outlines the architecture followed to develop the application as well as the way it was designed. In addition, it shows how the relationship between the components and what they do.

The system follows a Model-View-Control (MVC) architecture. The Model is comprised of the database; access to the database (i.e. connection and queries) is done through the /server/model.php. The Controller part is made up of individual page controllers. There are page controllers written in Javascript (Front End controllers) which process input from the client side and page controllers written in PHP (Back End controllers) which make requests to the model for the required information. There are both Front End and Back End controllers for each page. Together, the Front End and Back End controllers provide the functionality for the website. Lastly, there is the View. The View is made up of HTML (how the data is displayed) and CSS (the website style). Please note that some of the View is the Front End page controllers (Javascript files), as there is quite a bit of dynamic information that is displayed.

Recommendations

It is highly recommended to copy the system's environment onto a local machine running a full-stack development application such as XAMPP before making any changes. In this case, changes can be made on the local machine and will not affect the deployed system. This is done just in case the system is brought to a state where it can no longer run, so that the original, working application can still be accessed and used. This will prevent studies from being halted due to work being performed on the application's codebase.

When one section of the application is changed, it is quite possible that another component of the application will require modification as well. Therefore, the best course of action would be to plan the changes that are to be made and check the related components to see how they would be affected.

When changes are made without knowing the consequences on the neighboring components in the system and those components are modified on-the-fly, then it is quite

possible to run into the problem of spaghetti code. This is when code cannot be kept track of, due to the fact that it affects numerous areas of the system and it can cause the system to be quite unmanageable. Therefore, it is important to understand the implications of making changes to the code in one area, as well as on the adjacent components in the system.

Where to Start

Typically, the best way to edit a running system, or even creating a system, is to modify the elements which one can see (HTML, CSS). This is important because as the code gets closer to the Back End, it is more difficult to keep track of. Much of what happens in the Back End is behind the scenes and does not give a lot of opportunity for feedback.

If changes are to be made, such as adding a mini-game that users can play, the interface is the best thing to create first. This way, there is a visual cue to the changes made from then-on. Next, would be to connect the interface that has just been created, to the Front End controller. This enables the ability to make the content dynamic since the Front End controllers help interface with Javascript.

The next component that would need to be changed is the Back End controller to receive the requests from the Front End controller. For example, to keep track of high scores on the mini game, the functions would need to be used to pass the information from the Front End controller to the Back End controller. Once the information is verified by the time it gets back to the Back End controller, the Back End can be fully connected.

The Back End controller will handle sending and retrieving information to and from the database. Going back to the example, here is where the saving of users' scores would take place, as well as retrieving that information if it were to be displayed on the scoreboard. This is not enough to have the Back End fully working with these changes; a table, or as many as required, must be added to the database to store all the information. Once that is done, then the new functionality should be mostly done, save for some testing to make sure that is the case.

Worthy Mentions

Like many things, there is no one way to maintain code. The content here simply consists of suggestions. Some programmers like to work front-to-back, and some like to start in the middle (although that is seldom recommended). The best way is the way that the developer is comfortable with, and when in doubt, consult someone who has experience when it comes to working with web applications.

Additional Resources

Besides the documentation given by our team, there are many other sources for information all over the internet, many of which were used to make this application. Therefore, the following may be useful:

- To learn web development techniques:
    - www.w3schools.com
    - www.webdesigndegreecenter.org/learn-to-code/
- To ask the development community, code-related questions:
    - http://stackoverflow.com/

Known Limitations and Workarounds

Rewards: There is currently no functionality to modify the rewards. These are set in the code as 10 (Bronze level), 20 (Silver level), 30 (Gold level), 40 and more (Platinum level) for each of the total number of entries (data inputs), posts, and likes. The work around for this would be go into the code (js/utilities.js) and change the numbers there. In addition, (js/utilities.js) is the location where the reward messages can be set.

Image Posts: This is not supported in this current version of the web application. Since there is no option to upload images, the work around is to post a link in the text section of the post.

Share to Social Media: There is no option to share posts and statistics to social media sites such as Yammer, Facebook and Twitter.

Installation Guide

Provided with this document are the SQL files to create the tables in the database; they are located in /server/support/clean

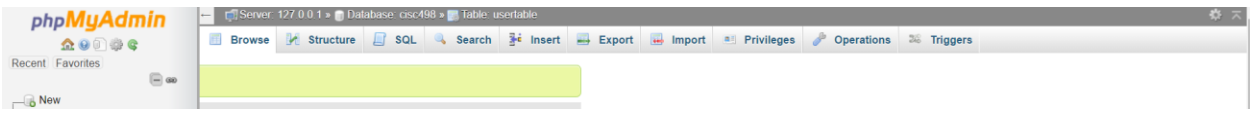
The system is already configured with Jane and Sandy being Super Admins. Their account information is below. The usernames, passwords and emails can be modified after logging in.

Username: Jane_Webster	Password: Jane_Webster	Email: editThis@gmail.com
Username: Sandy_Staples	Password: Sandy_Staples	Email: editThis2@gmail.com

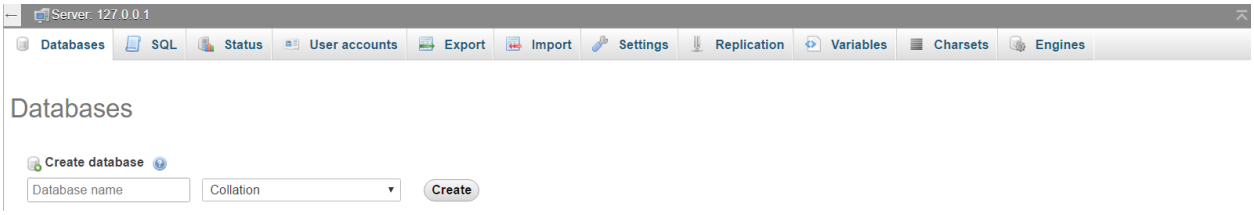
The system does not have any studies or users (besides Jane and Sandy).

Transferring the Database

There are two processes in order to set up the web application on a separate server. Assuming ITS has audited the code and has approved it, access phpMyAdmin or the equivalent interface ITS uses. ITS will provide instructions on how to access this. Once logged into phpMyAdmin (or equivalent), you now have access to the web application's database and will look something like the below.



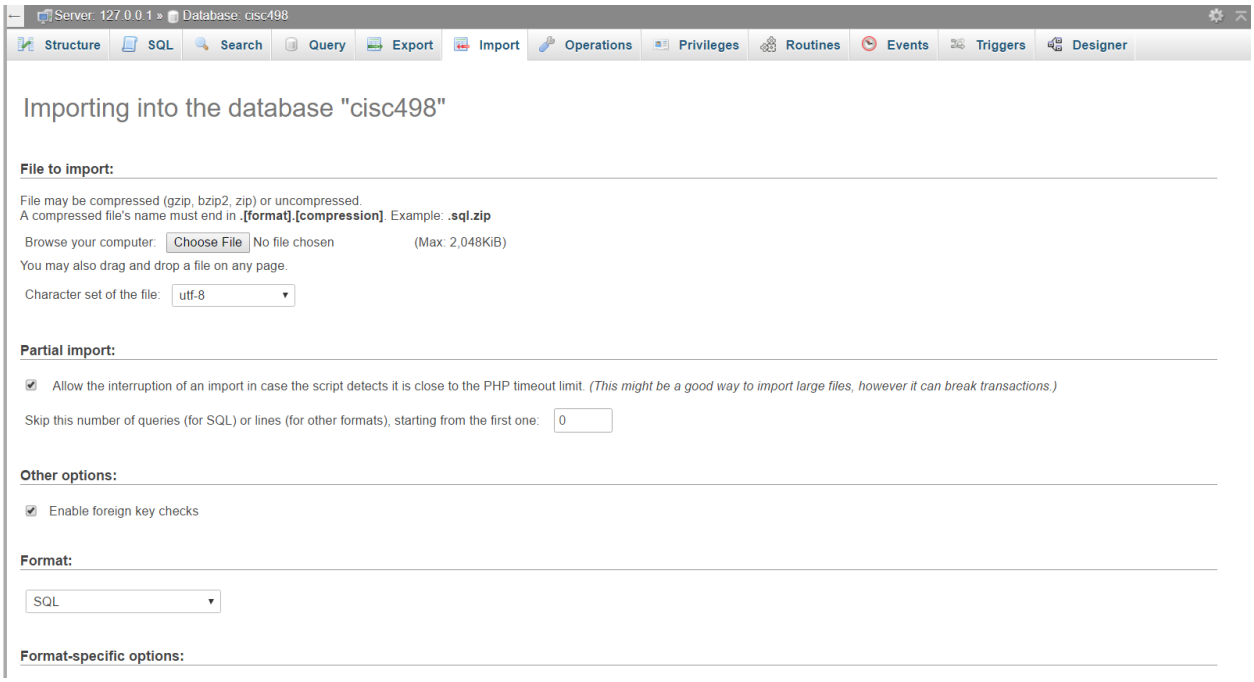
Create your database by clicking "New":



Name your database and in the drop-down, select utf8_general_ci. Press “Create”.

Note: the current database in place is named **cisc498**, and the password is **devcpp**. If you decide to change the name and/or password, these must be updated in the dbConnect function found in /server/model.php - it is the last function in the file. Failure to update the database name and password in the file will result in the website not being able to reach the database.

Now to create the tables: go to “Import” and you will be brought to a screen like the below:



Choose a file from the /server/support/clean directory to import a table into the database. Scroll to the bottom of the page and click “Go”. Repeat this until all tables have been added to the database. You can always come back to this page to import a clean version of the tables.

Transferring Files to the Server

Please refer to the provided PDF, “Host your Application in The Amazon Cloud with XAMPP and Bitnami,” as this provides an excellent step-by-step process of setting up a FTP (File Transfer Protocol) connection to the current server, which is on Bitnami. The PDF itself is a user-guide on how to create and establish a server and database on AWS and Bitnami and is the document we followed to create this web application’s server and database. It is also is helpful in explaining how to set up a local environment (using XAMPP) for further development on the site. A similar process from section 3 below can be used to transfer files from one server to another.

“Host your Application in The Amazon Cloud with XAMPP and Bitnami” pages breakdown:

1. Pages 1- 21: Creating an AWS and Bitnami accounts, connecting the accounts and launching the server
2. Pages 21-30: Establishing an SSH connection using PuTTY, getting phpMyAdmin on Bitnami running to access the database
3. Pages 30-36: Deploying local changes from XAMPP to the cloud server using FileZilla
4. Page 37: Importing into the database
5. Pages 38-42: Overview of Bitnami Cloud Hosting Dashboard and Monitoring Tools
6. Pages 42-44: Improve Application Performance

Sections 2 and 3 will be the most useful and used throughout the continued improvement of the web application.

For further development of the web application, it is recommended that XAMPP be used to support a local development environment. Any changes done locally on a machine must be pushed up to the server through File Transfer Protocol.

Summary of Updates to Previous Documents

Requirements Document

- The reward system was added and implemented; users can earn badges based on their number of data inputs, posts and likes. In the current version of the web application, these rewards are hard coded; however, the infrastructure to customize and store rewards in the database is in place.
- Studies can now be archived and purged. This allows the admins to end a study and keep the data for further analysis until the study is purged (i.e. removed from the database completely)
- An additional level of admin was created, the Super Admin. The additional privilege level adds security to the web application by restricting all functionality to the highest privilege level. An Admin only has access to his/her studies (unless a Super Admin has assigned that Admin to a study) and users.

Please refer to the Known Limitations and Work Arounds section to see the functionalities the current version of the website does not include.

Internal Design Document

From being inspired by an MVC architecture, this web application implements it. The interface between the Front End and Back End has been created and established. As mentioned in the Maintenance Suggestions section, there are two controllers for each page: one for the Front End (in Javascript) and one for the Back End (written in PHP) to connect the Front End and Back End. The Javascript controllers accept the client-side input and pass along the request to the PHP controllers. The Model part of the MVC is all in one file, /server/model.php, which the PHP controllers call specific functions to make queries to the database. The PHP controllers then return the information from the database to the Javascript controllers, and then subsequently to the View (HTML and CSS).

In the database, the userTable stores more information than initially documented. It now additionally keeps track of a user's number of entries (data inputs), posts, and likes total and per phase. It also keeps track of the user's team number (sub-team) and earned rewards.

External Design Document

The styling for the website has been greatly improved upon. To see screen shots for each page, please see the User Manual section of this document.

Recommendations for Further Improvement

The following is split up into sections: Security and Functionality. The aim of this section is to provide a basis for website improvement.

Security

The website cannot be hacked by regular (i.e. non-malicious) users. The web application does do basic input checking; however, more input checking can always be done and is recommended. Like all websites, this web application is susceptible to malicious users who target websites to bring them down, and/or corrupt the system.

HTTPS: The website does not have a security certificate. In order to obtain one, a designated administrator must own the domain of the website. He or she will use the domain account to request and pay for a certificate. As of now, the domain is owned by an external party. It is recommended that a domain is purchased.

SQL Injection: The website does basic input cleaning. For further security, it is recommended that the website's model (as part of the model-view-control design pattern) be configured to support SQL prepared statements. This is to ensure that no other commands than what is expected will be processed.

Cross-Site Scripting (XSS) Attacks & Cross Site Forgery Attacks: Currently, the website offers no protection for this. It is recommended that validation of raw input be done.

Destroying Sessions: To ensure that a user is properly logged out (i.e. should not be able to get back to the previous page after logging out), the cache must be cleared upon logout. This is because the system has some direct calls to Javascript to load the pages; clearing the cache would get rid of the Javascript (and hence the page calls). To fix this, it is recommended that all requests for the page views go through a controller to vet the requests based on authentication, or set up the server in such a way that there is no direct access to the view files on the server; the server will only be able to access those files. Option two seems, at this point, to be the better option.

Login/Logout: These are always areas of importance for any website, and it is recommended that further investigation in securing the login and logout.

Accessing the website without logging in: A user who is not logged can access the user-template.html and admin-template.html pages; however, there will be no data pulled from the database. The fix for this would be to create a controller which processes website requests and redirects to the index.

Functionality

Customization of rewards: The Customers have expressed that they would like to be able to create their own rewards for studies. This can be done in a similar manner as the Create Study is done.

Sharing to Social Media: Users should be able to share posts and statistics to social media. This needs to be done in such a way that maintains a user's privacy and security integrity. In addition, since external system users will not be able to access the page (as they will not have

login credentials), the shared post/statistic will have to rendered to allow external users to see the post/statistic. One option may be to take a partial screenshot and share that.

Super Admin/Admin Upload Media to Community Posts: Because of the challenges faced with handling dynamic uploading and displaying of images and other media, this function is not supported in the current version of the website. It is recommended that someone look into how to handle dynamic BLOBs in the database.

Put all Text in a File: Having all the static text (i.e. headers, welcome messages) in one file would make it easy to edit, as well as making it easy to support multiple languages.

Mobile Devices: Currently, on mobile devices and tablets, the website's home page is the only page that can be accessed. One of the reasons for this is because a lot of the events are triggered by onClick(), which touch screen devices do not register. The improvement here would be to add touch support.