**SQS Quality Assurance:**

**Enhancing the Training Web Application**

Team 6

December 4, 2017

Jake Davis

Dylan Jenkins

Binod Katwal

Manni Mashaee

Customer: SQS

Assignment: Maintenance

**Analysis**

Requirements

SQS has the intentions of this website being the training tool for their employees. With that being the case there are quite a few requirements in order for this to be fully implemented. A typical SQS employee should be able to take all the files provided, set up an Apache server running a MySQL database easily by using one of the various software programs to do so (LAMP, WAMP, XAMPP). From there the employee should be able to follow the README file provided, create the appropriate tables, and run the website from any available web browser.

The user will begin by signing up, the signup process will begin by asking the user for the required information (Name, Email, Password). The user will then be brought to an address information page. Asking for the user's address, email, phone number, etc... This information is not required but highly recommended, the user may choose to fill out the information or continue to the next page. Upon completion of the address information page the user will be brought to a page of questions about the user's software skills. After this there will be a hardware skills page asking about the user's hardware skills.

Again, this information is optional and may be left blank allowing the user to simply register with no information other than the required information on the first page. During the redesigned registration process, the user should be able to navigate backwards and forwards and each page should retain the information filled out. The progress bar should appear during this process and will update according to how far along the user is in the process, also allowing for backwards and forwards navigation and retaining the information on each page, if the user needs to go back a page, the information on the current page is kept unless canceled or cleared by the user.

Then behind the scenes each clickable item will have a unique element id to identify it on the website. This is something that will be beneficial for the automation team that will eventually automate the website. These element IDs will provide a means to differentiate each button, text box, link, and anything else a user can interact with or click on. Upon completion all of these features should integrate seamlessly into the web application as if they have always been there.

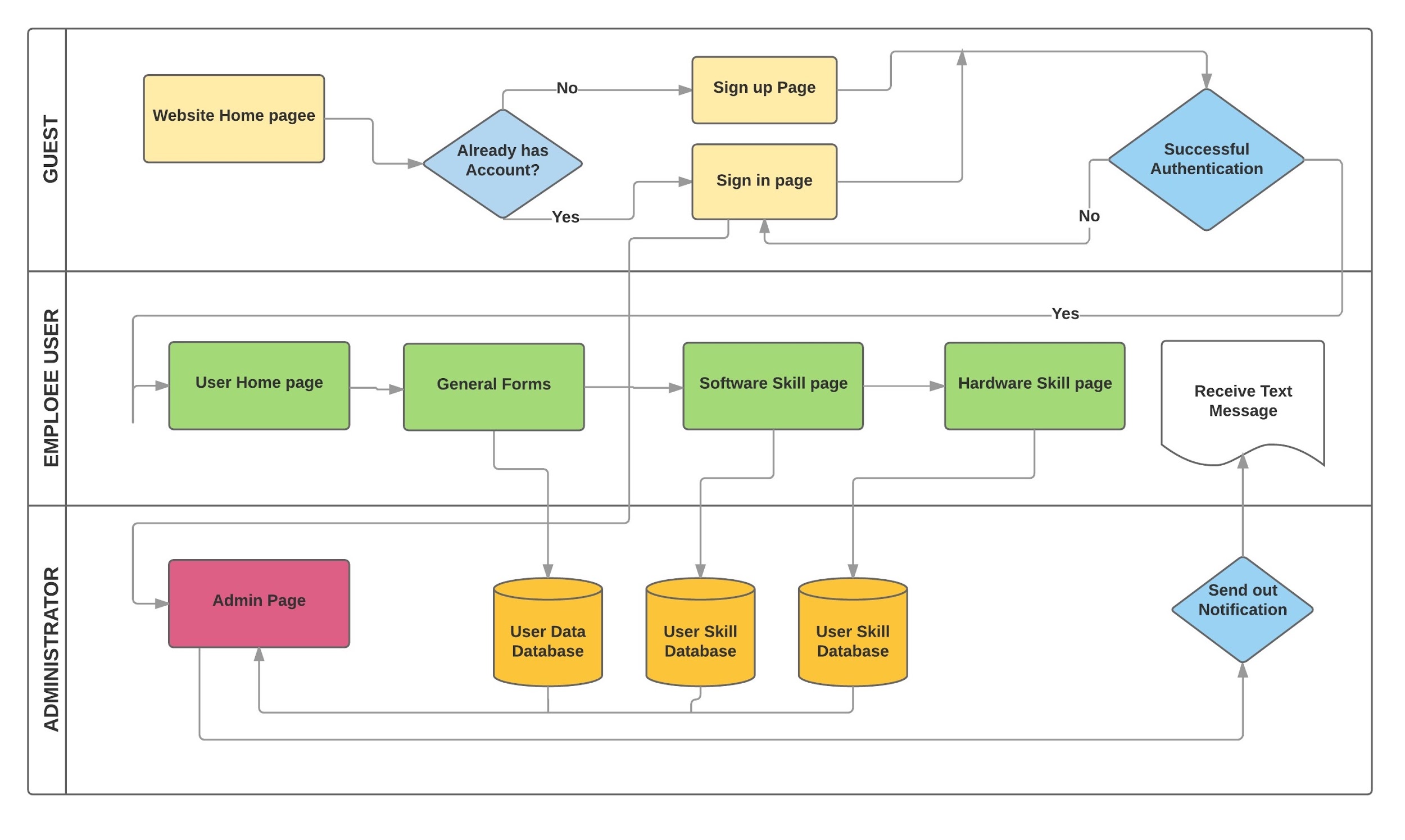
Lastly, for our twist in the project we were tasked with implementing an individual progress bar that keeps track of exactly how much information the user has filled out. This progress bar is to be displayed on each user's profile page and will be set depending on how much information the user has input into the database. 100% being everything the user can fill out, has been filled out. The percentage changes pending on how much information the user has input and may be clicked on to direct the user to the pages to enter the remaining information. Each Admin can see these percentages on the Admin page corresponding to each user.

**High Level Design**

Architecture Diagram

All components of the stack interact with each other to provide the required functionality of our website. Looking at **Figure 2**, The five components are: front-end(HTML/CSS/Javascript), back-end(PHP), Apache Web Server, Web Application, and the MySQL database. When a user initially types the url of our website a request is sent from their browser to the Apache Web Server for the website home page and because PHP is used in the website’s code, the server will turn over the request to the PHP module, which reads the PHP and replaces the code with the output it generated back into the page, and outputs it onto the browser. Three levels of users will be allowed on the website: guest, employee and admin. Each user type will have a different user interface and access to different types of data.

User’s will first be routed to website home page, and be asked the sign up or sign in, if not signed in, the signup process will begin by asking the user for the required information. Otherwise the user will login and send a request to the Apache Web Server in an attempt to be authenticated from the MySQL database. If the credentials provided are valid, a response will be sent back authorizing the user and sending the appropriate page requested from the database back to the server and displayed to the browser. If the credentials are not valid, the user will not be authorized and the requested page will not be sent. After authentication, based on the credentials entered the appropriate page will be served. If valid admin credentials were entered the page served will be the admin page which is the only page where you can edit other user’s information, otherwise, the user home page will be served.

Figure 1: Architecture Diagram

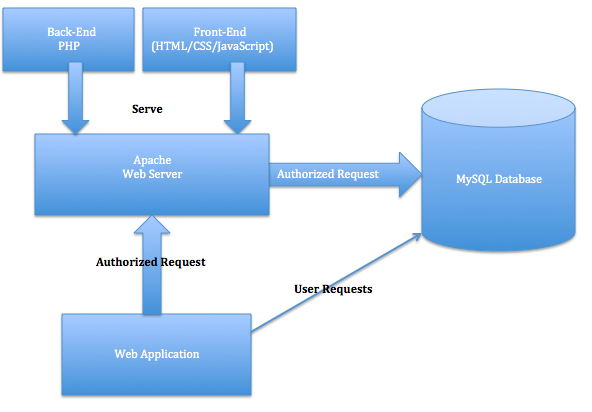
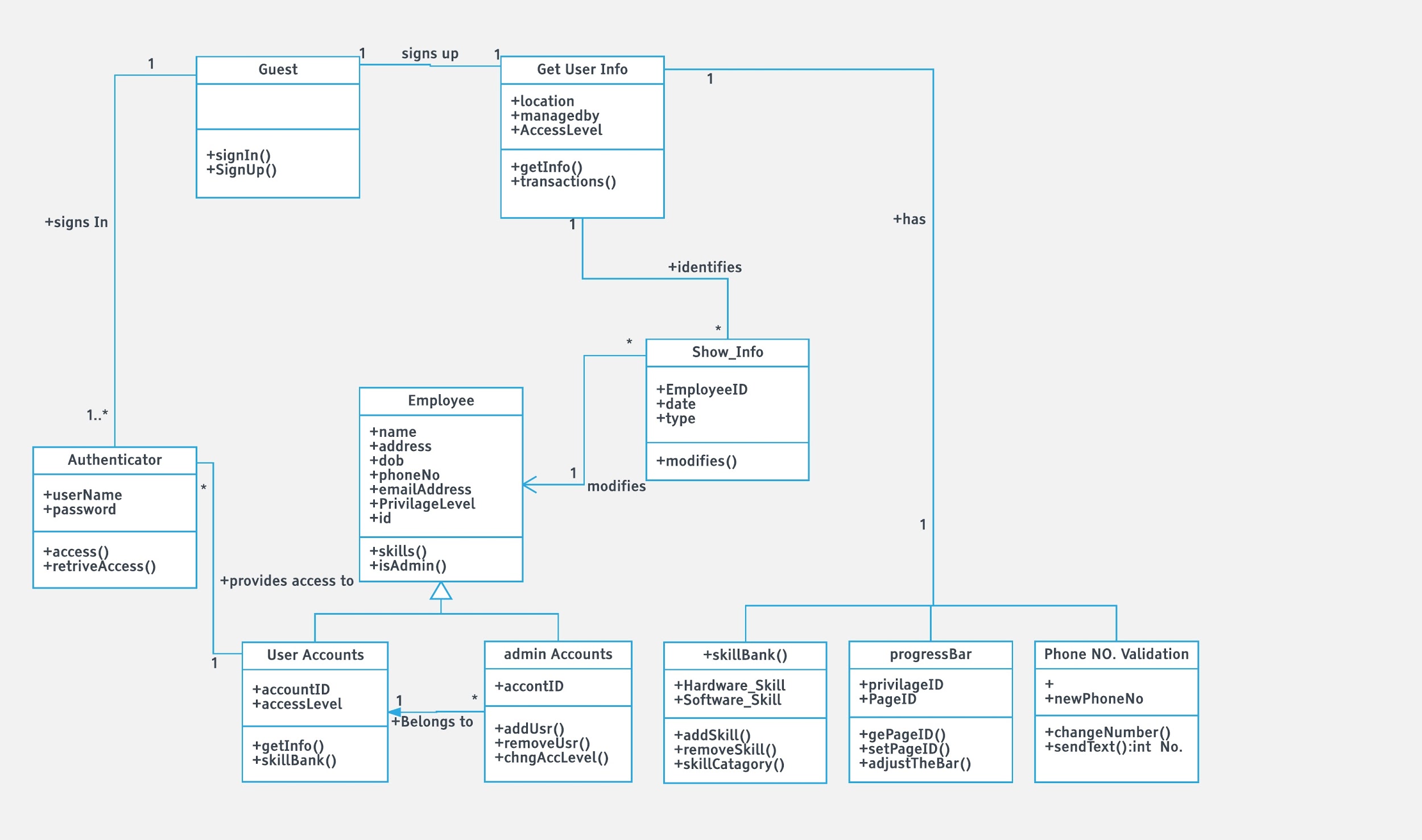
****

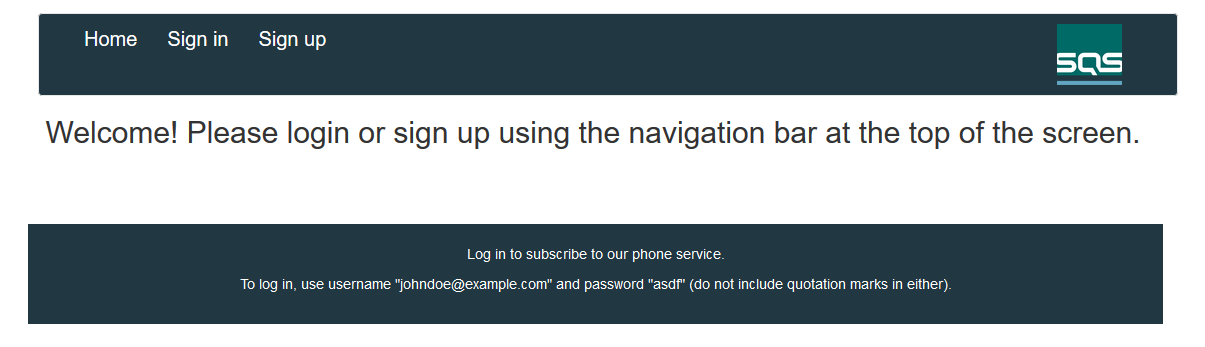
Figure 2: Architecture Diagram

**Detailed Design**

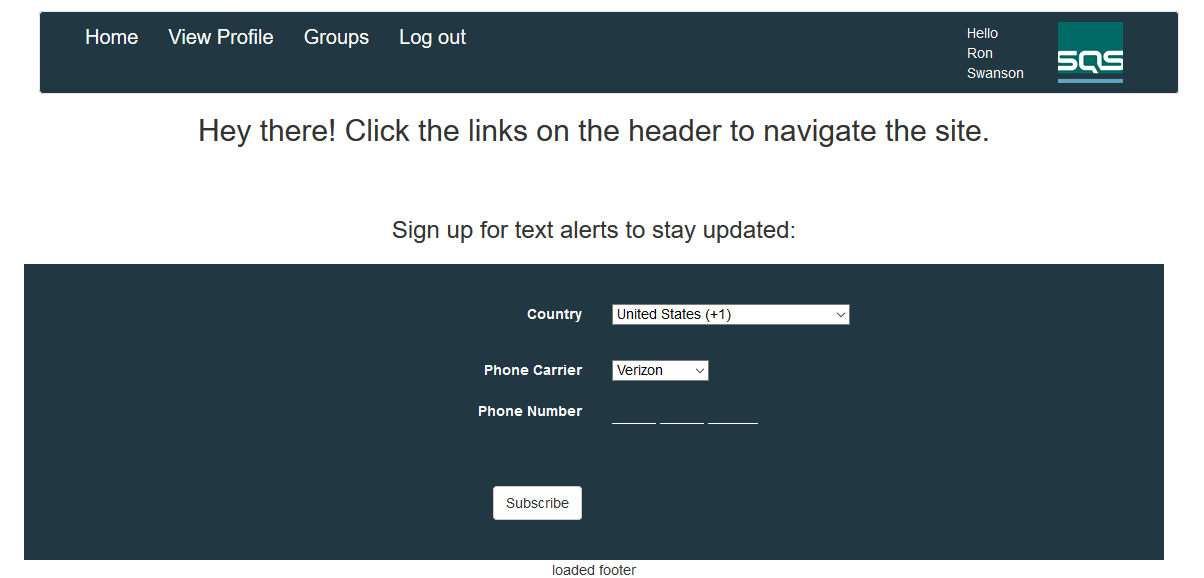
Detailed Design

Figure 3: Class Diagram

User Interface Design

The web application SQS has provided has a lot going on behind the scenes. Thus, they designed the interface of the web application to be as easy as can be. The first image shown is the home page where no user has signed in. 

For this page, the user really only has a couple options. Either to sign in or sign up. If the user decides to sign up they will be directed registration pages shown in the User’s Manual. Otherwise Signing in will direct them to a page asking for their email and password. Then redirecting them to the homepage after signing with added features to the user’s homepage.

****

Once the user has signed in they may subscribe to the phone listings. This simply adds the user to the corresponding table in the database. From here the user may click View Profile to view their profile which displays all the information the user has submitted to the database or the user may click Groups which directs them to a page that displays all active groups and their group leaders. Images for these screens are also in the User’s Manual.

Finally there needs to be a difference between the privileges of users. So there is a regular user, a superuser, and an admin. A regular user has all the privileges shown so far, a superuser and an admin will have an additional link to a page that allows them to edit groups. However, a superuser may only edit the groups. An admin has the ability to edit groups and group leaders. Due to their similarity only the admin page is shown.

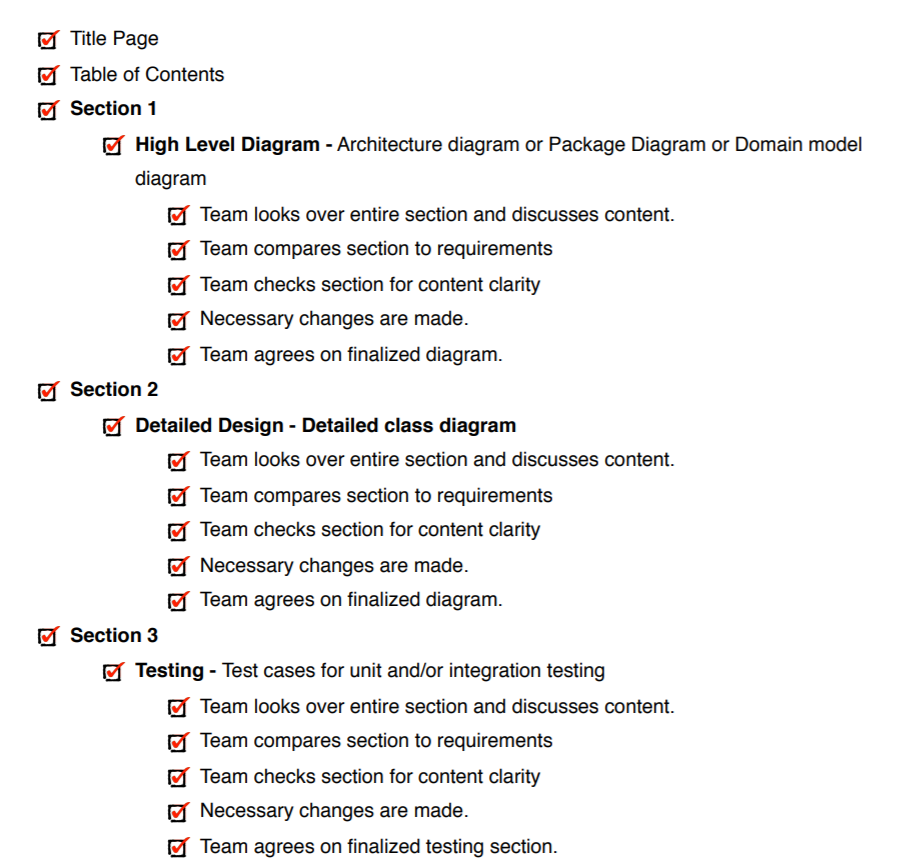
The interface for the database itself will really vary pending on the desired method of setup. Currently are using XAMPP and phpMyAdmin. Alas, within the database there needs to be 11 tables. The sql statements to create the tables were provided. The tables needed are assigned\_features, email\_list, features\_availbale, groups, group\_members, levels, level\_privileges, mail\_address, phone\_list, privilege\_list, subscriber, and user.

Design Pattern Use

For the web application that SQS has provided and tasked us with enhancing, there were no real design patterns in use. Thus, we did not implement any specific design patterns.

Assurance Review

With the quality assurance review in mind, a personalized checklist was devised for the product system. The checklist, found below, was used in confirming that the overall architecture and each section of the document met all the necessary specifications required by the assignment prompt.



**Implementation**

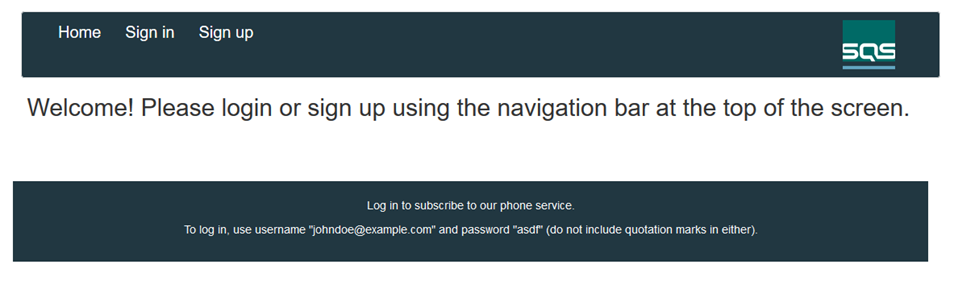
Source Code

The source code for the project has been hidden from public access. The code is saved in a private repository hosted using Git and one of UK’s various servers. Any access to the code must be given from a current project member.

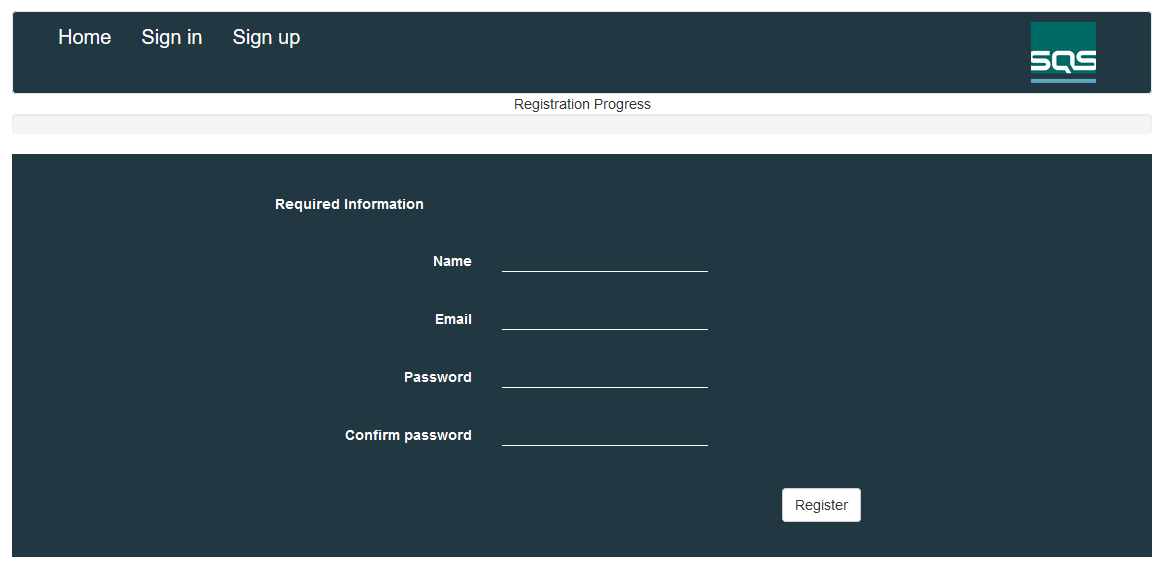
User’s Manual

The web application has four types of users. An Unregistered User, a User, a Superuser, and Admins. The User’s Manual will cover a typical user interaction.

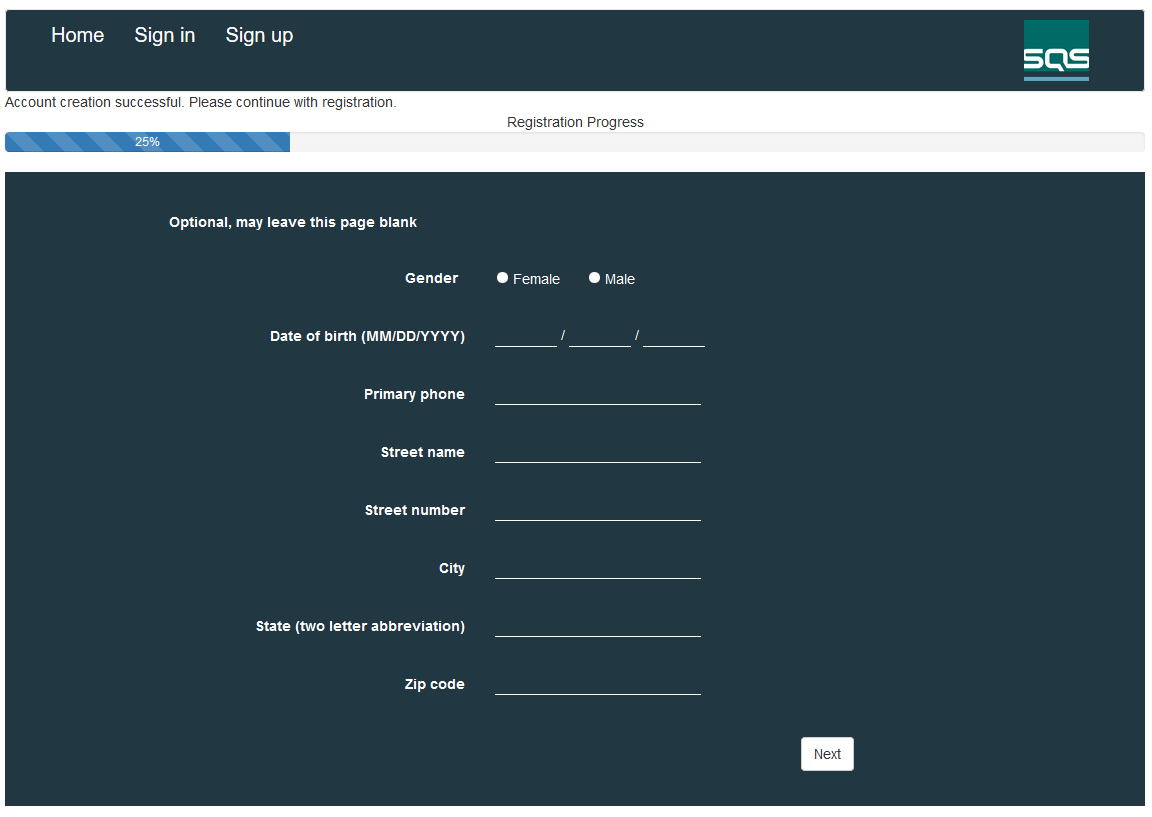
Once the web application has been set up and the correct database tables created. A user will load up the web application and be presented with a page looking like.



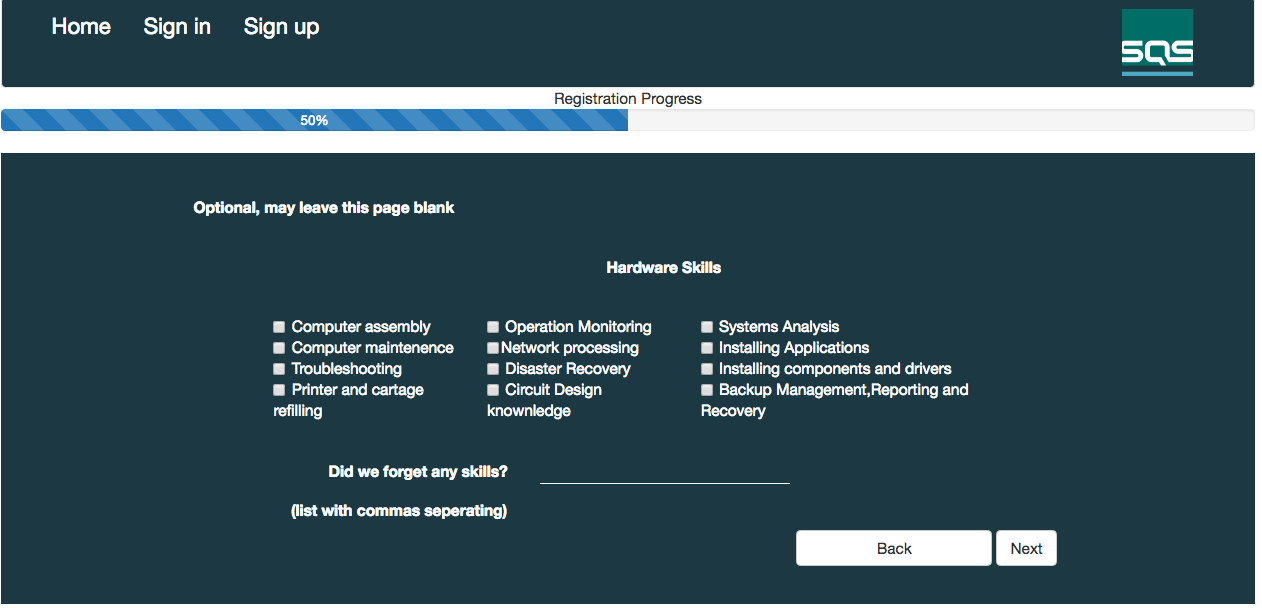
From here a user has two options. To Sign in or Sign up. At this very moment the user is considered an Unregistered User. If the user needs to create/register for an account they may click on Sign up and will be taken to the registration process.

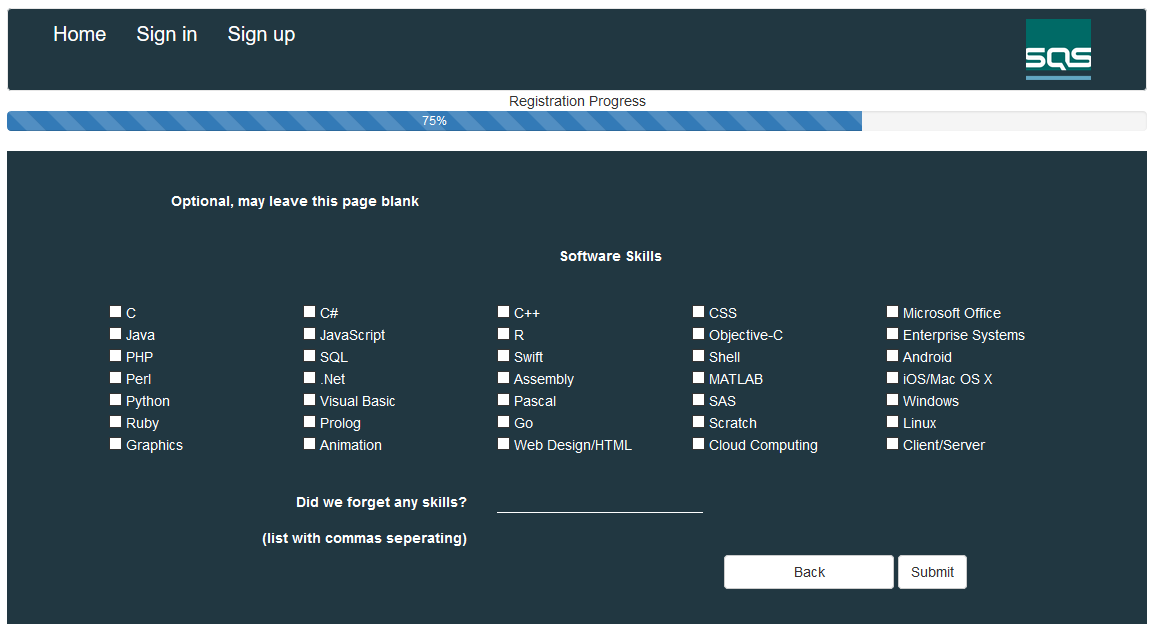


The first page the user is presented with for the registration process is the required information. This information must be filled out in order to create an account. Once this information has been filled out the user will click Register. This will then register the account and verify it not a duplicate.



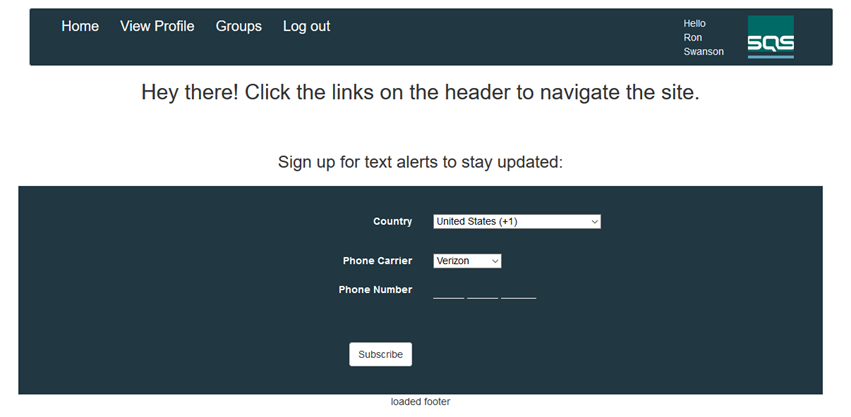
The user will then be brought to the second page in the registration process. This information is entirely optional and may be left blank, if the user wishes to proceed they can simply click Next to proceed to the next page or Sign in to skip the entire registration process. The second page of the registration process is simply an address information page, obtaining basic address information, birthday and gender. When the user clicks Next they will be taken to the third page in the registration process.

  
  
This page is also optional and may be treated the same as the second page. However, the third page is asking for the user's Hardware skills. We have provided check boxes covering the main areas we felt were desirable and left a small textbox at the bottom for the user to add any skills we may have missed. Once again, the user will click Next to proceed to the fourth and final page in the registration process.

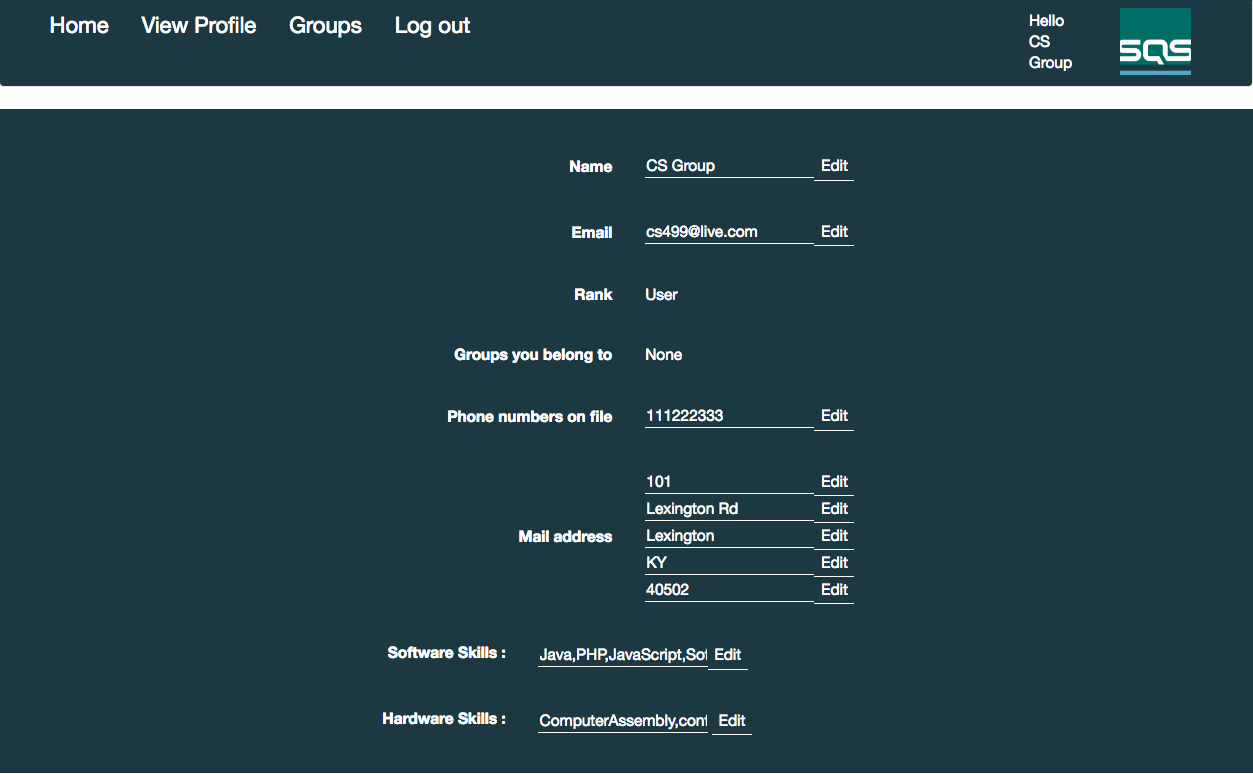


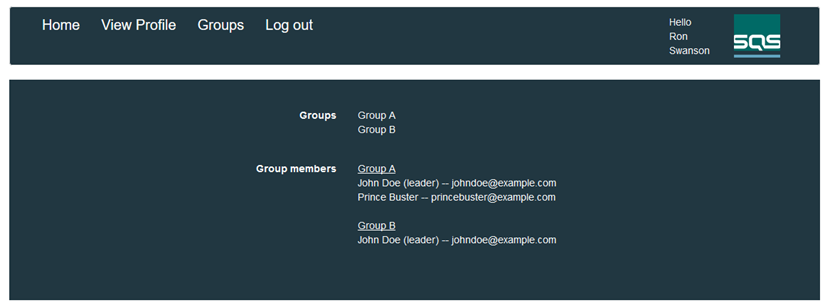
The final page can be treated like the second and third page. The fourth page is asking for the user's Software skills. We have provided check boxes covering the main areas we felt were desirable and left a small textbox at the bottom again for the user to add any skills we may have missed. Finally the user will click Submit to finish the registration process and will be directed back to the home page. Now that the Unregistered User has become a User they may sign in using the credentials entered during registration.

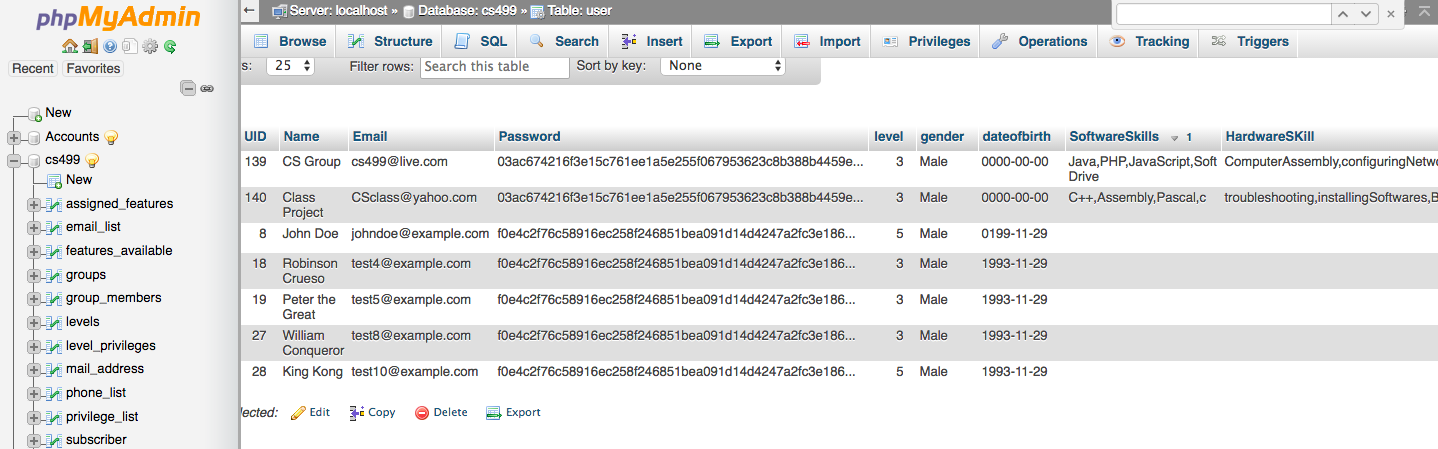




Once the user has signed in they are given a couple more options. They can subscribe to the phone listings placing them in the corresponding database table. From here a User may click to View Profile, simply displaying all the information the user has entered from registration on screen. The twist was to add a progress bar for each user to indicate how complete each user's profile is or how much information the user has input into the database. The other option is to look at Groups, this will direct the user to a page displaying all the users in groups, the groups, and their corresponding leaders in the database on the server. The user may also Log out. Lastly we added an image of the database to give an idea as to what is going on behind the scenes.



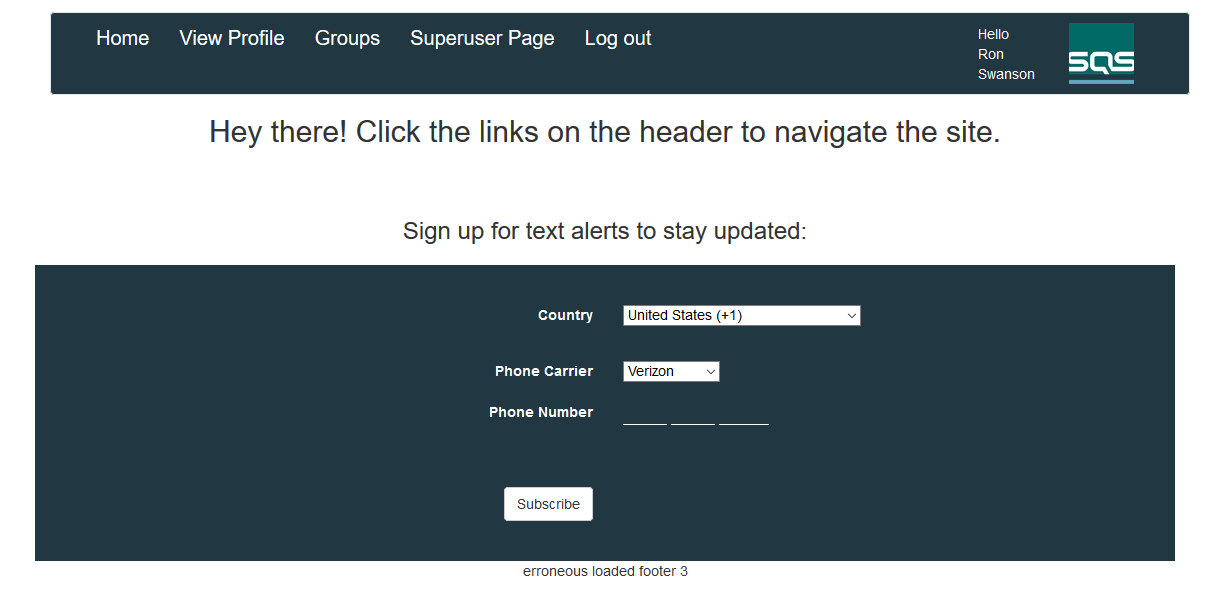




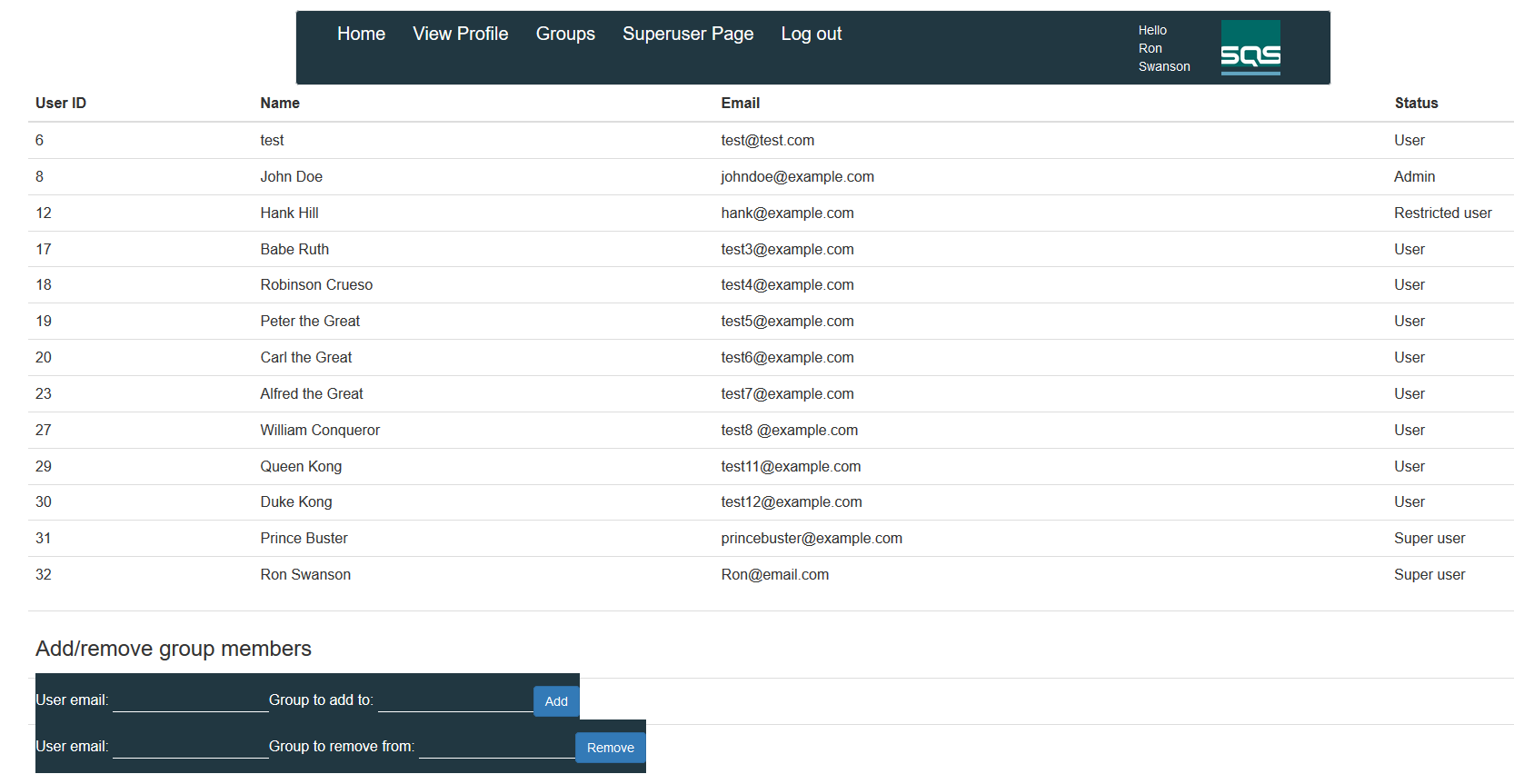
Administrator’s Manual

The Administrator's Manual is really very similar to the User’s. Once the user has logged in, if they are a Superuser or an Admin they will be brought to two different home pages.

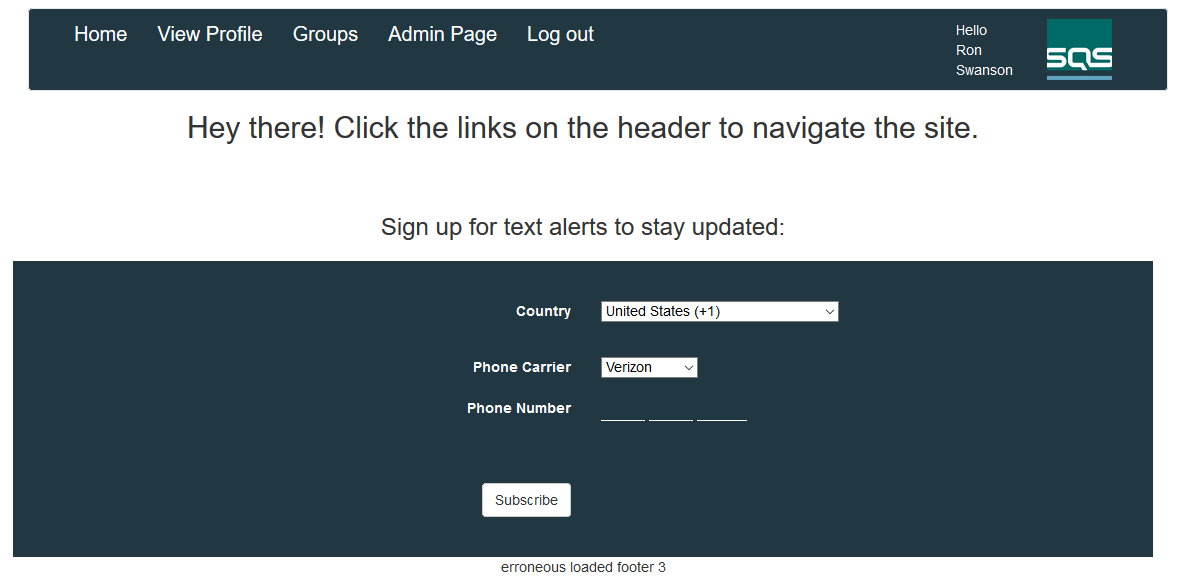
Superuser’s Home Page:



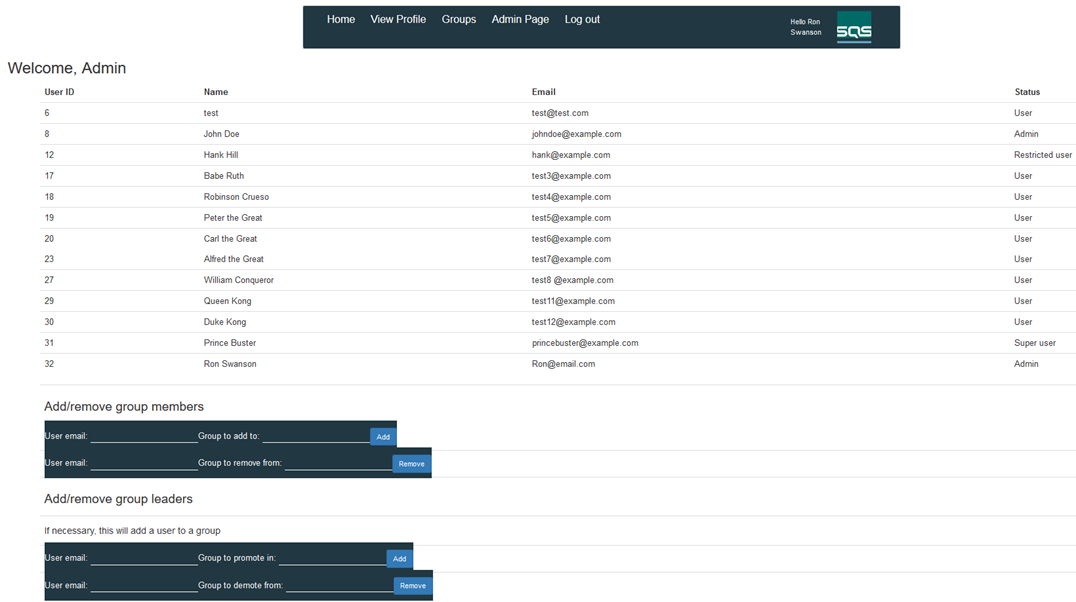
The Superuser's home page looks quite similar to a User’s home page but it has the addition of one option. This is the Superuser Page, from here the Superuser may look at all users registered in the database and has the privilege of being able to add/remove users from groups.



Admin’s Home Page:



The Admin has one additional option on their home page. This is simply titled Admin Page, below is shown an image of what this page looks like. From there the Admin can look at all users in the database and add/remove users from groups as well as add/remove users as a leader of the group. This is the main difference between the Superuser and the Admin.

****

Code Review

The review of code for our twist brought no bugs or errors to our attention. It appears to work how we would like it to. It passes all tests.

**Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Valid new user sign up.  *Required information check* | User account created in the database. | User account creation fail and error is displayed. | Pass |
| 1b | Invalid new user sign up.  *Required Information:*  *Passwords don’t match* | User account creation unsuccessful and error is displayed. | User account creation successful. | Pass |
| 1c | Invalid new user sign up.  *Required Information:*  *Email in use* | User account creation unsuccessful and error is displayed. | User account creation successful. | Pass |
| 2 | Valid user sign in. | User login successful.  Redirected to home page. | User login fail. | Pass |
| 2b | Invalid user sign in. | User login unsuccessful.  Error displayed. | User login successful. | Pass |
| 3 | Address information submission | User address information in the database. Visible on user profile. | Information not found in database or on user profile. | Pass |
| 4 | Hardware skills submission | User hardware skills in the database. Visible on user profile. | Skills not found in database or on user profile. | Pass |
| 5 | Software skills submission | User software skills in the database. Visible on user profile. | Skills not found in database or on user profile. | Pass |
| 6 | User update’s hardware skills. | Updated hardware skills in the database and on user profile. | Skills fail to update in the database and/or on user profile. | Pass |
| 6b | User update’s software skills. | Updated software skills in the database and on user profile. | User skills fail to update in the database and/or on user profile. | Pass |
| 7 | Progress bar update. | The progress bar updates according to where the user is in registration process. Forwards or backwards. | Progress bar does not update or does not appear. | Pass |
| 8 | Registration process cookies | User information is retained on the pages during registration process. | User information is not retained. | Pass |

**Technical Metric Collection**

Lines of Code & Story Points

SQS has given us seven features with 29 story points and a total of 12 unit test cases. At about 450 lines of code added to the existing web application. Some lines were altered in order to provide the required features desired and may not have been counted as a new line added.

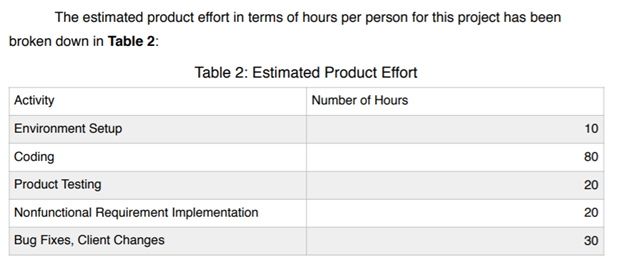
Estimated: Approximately 50 lines of code will be added to the web application.

Actual: Approximately 48 lines of code were added to the web application.

Complexity

This system does not have inheritance due to the nature of Javascript, html, and php. With no classes or objects there is a lack of an object oriented infrastructure. Coupling between classes is also not applicable since none in our system depend on the other.

Product Effort



Defects

No defects were found during the quality assurance review. Majority were hammered out during coding and testing.