

GLAS
Employee Selector & Resume Generator App
Project Architecture
CS499 Spring 2019
March 4th, 2019

Editors:

Authors: Angelina Grosso, Lexi Kahwaji, Grant Sturgill, and Sara Vasquez

SQS Contact: Ben Fox

High-level Architecture

The SQS Employee Selector and Resume Generator is an extension to an SQS's training website to help facilitate the SQS employee and client matching done by the SQS sales personnel. The goal of this extension is to better ensure that SQS clients are matched with the SQS employee who is best suited to meet the client's needs based on mastered skills, previous clients, education, and previous verticals of work. Since this extension is part of a previously developed training website, the employee selector and resume generator will utilize the same technology as previous developments. The site was intended to run on a Linux machine and be accessible through any web browser, so the website was originally developed using a LAMP stack. This is a combination of Linux, Apache, MySQL, and PHP intended for web development. This restricts use of the training site to those with access to a Linux machine.

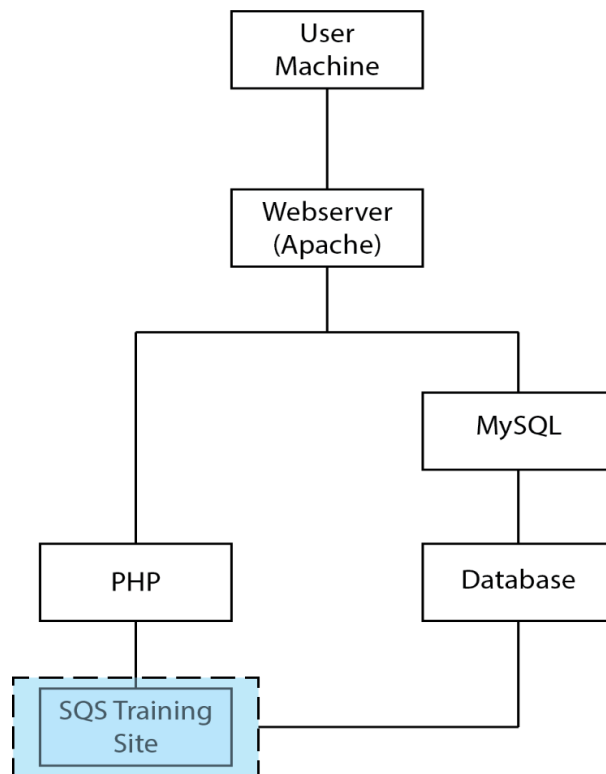


Figure 1: SQS Training Site Architecture

Upon installing the website software, the site will be hosted on the user's machine with Apache acting as a webserver to connect the user's machine and any given web browser. The software installation also includes the installation of MySQL and PHP, which will be utilized to maintain user data and visualize the website. Apache interacts with the user's machine and the chosen web browser to visualize the site contained in PHP scripts. PHP continually interacts with Apache and MySQL to visualize the site and maintain user data as the user interacts with the website.

Detailed Design

This project is programmed in an MVC (Model-View-Controller) structure. In this structure, the user is shown and interacts with the frontend page, the *View*, which triggers functions in the business logic *Controller*, which updates the data structures in the *Model*, which in turn updates the *View*.

Methods

There are mainly four functionalities being added to this project. The first is the addition of the Sales User, and the updated verification system attached to the new user type. When a user is creating their account, they will have the option to select whether they are a Sales User by clicking a checkbox. If they do so, they will be redirected to the awaiting verification page, where an administrative user will go to verify their account. The diagram below gives a detailed representation of how this verification will be implemented.

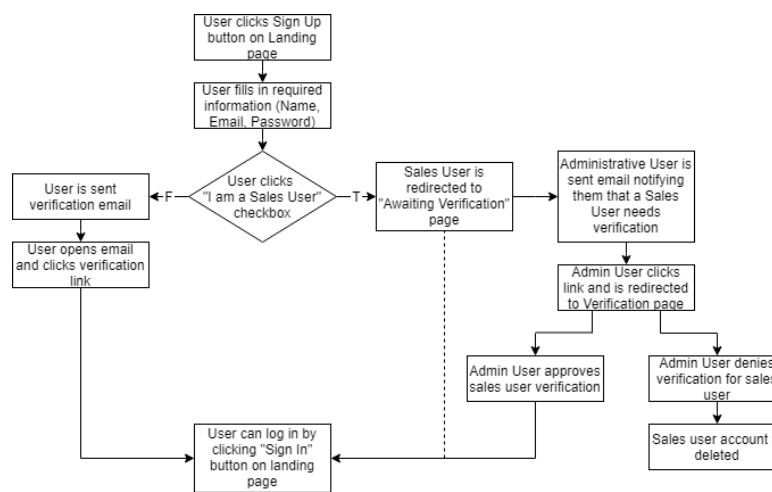


Figure 2: New Verification System

The second and third new functions are the Search and Compare pages. These pages allow a sales user to search for regular users by criteria such as skills and education. In the Search page, the user inputs a keyword (such as C# for a skill) to be run against the site database. The database then retrieves all users that have the C# skill. The matching users are displayed in a list view, wherein the sales user can select users to compare by clicking a checkbox at the end of each row. The sales user can also add filters and more search criteria, either by selecting tags in the sidebar on the left of the Search page, or by editing their original query. When the sales user edits their search, the database runs the new query, and the list is updated. Once the sales user has selected the users they want to compare, they click the "Compare" button, which send the list of user IDs to the Compare page, which then opens in a new tab.

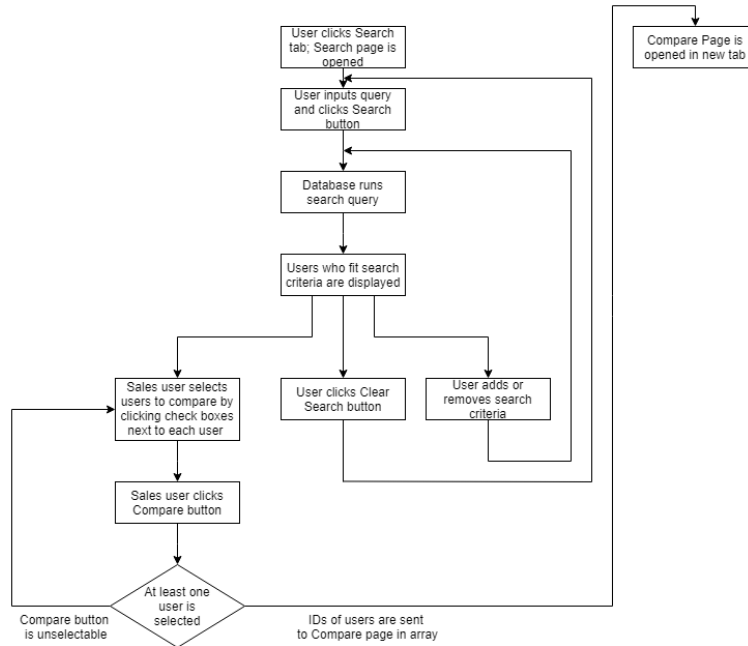


Figure 3: Search Page Methods

In the compare page, the selected users are displayed in a tile format. The sales user can remove users from the comparison by clicking an “X” icon on the user tile, which deletes that tile from the page. They can also open a user’s full profile by clicking the “Expand User” button, which opens the user’s profile page in a modal. Additionally, the sales user can show or hide user information such as certifications or previous clients by clicking buttons at the top of the screen.

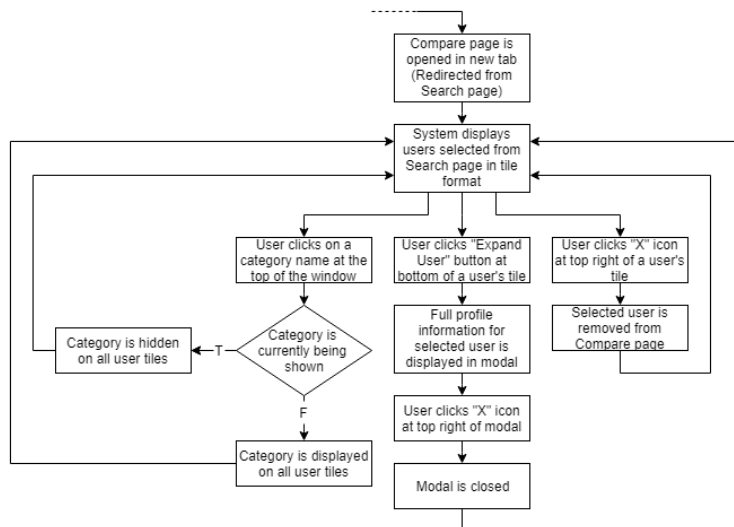


Figure 4: Compare Page Methods

The last functionality that will be added is the ability to select a skill level and years of experience for a skill that has been added to a regular user’s profile. When a regular user adds a skill, either during account creation or in their Profile page, the user will be prompted by a modal to select a

skill level and input the number of years corresponding to their experience with the skill. Once the user has input that information, the user skill and attached information will be added to the database.

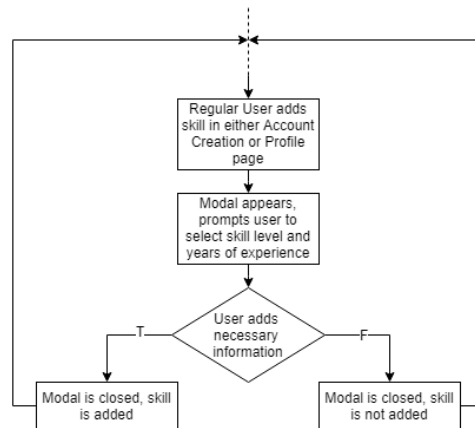


Figure 5: New Skill Information

User Interface

There are only three interface pages being added in the scope of this project: Administrator Verification, Search, and Compare. All three of these pages will be able to be viewed by administrative users, while sales user will only interact with Search and Compare. Regular users will not be able to use any of the three. Wireframe representations of the new pages are shown below.

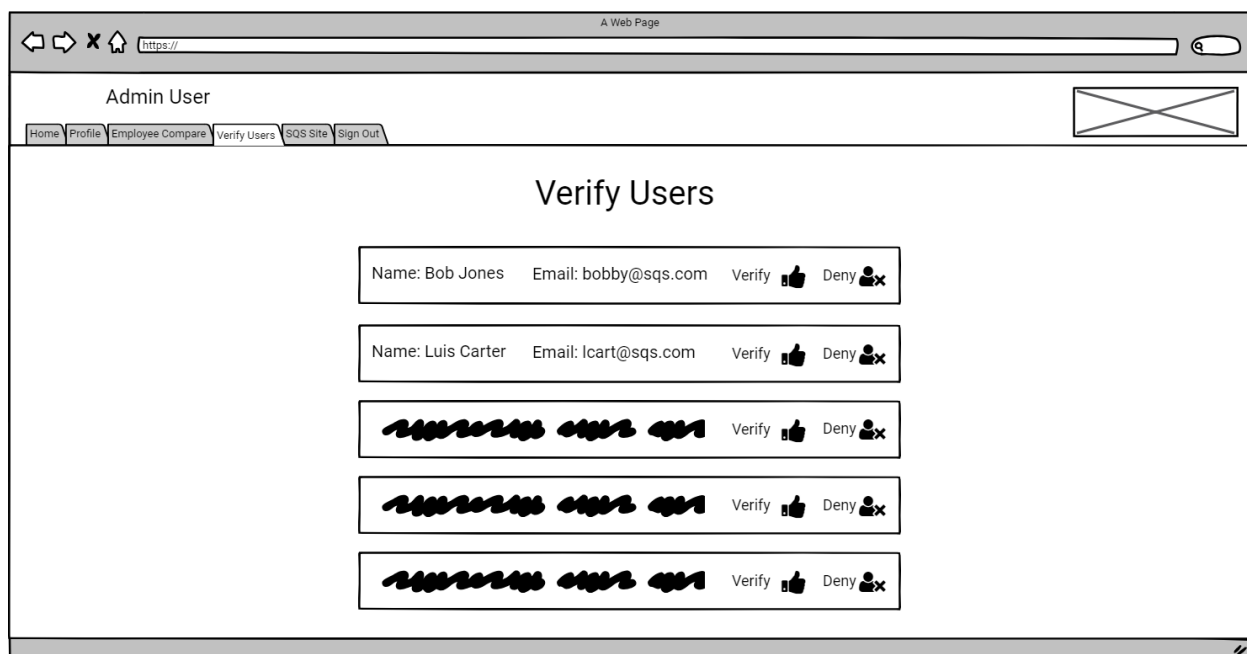


Figure 6: Administrator Verification Page

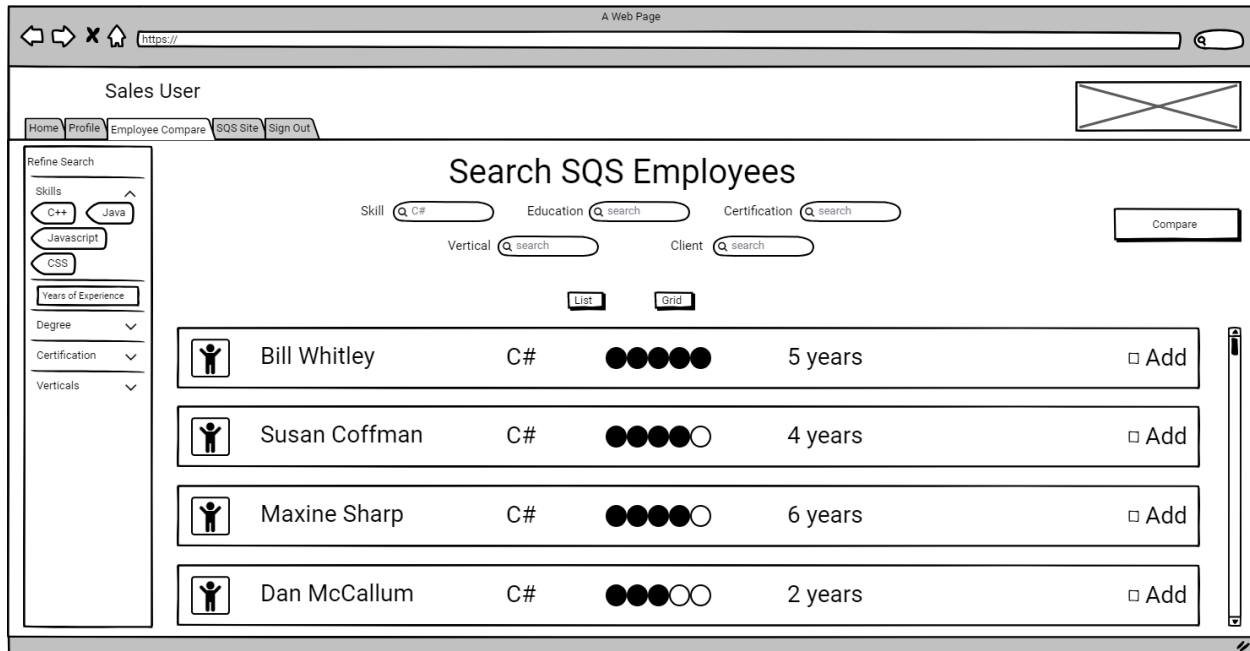


Figure 7: Search Page

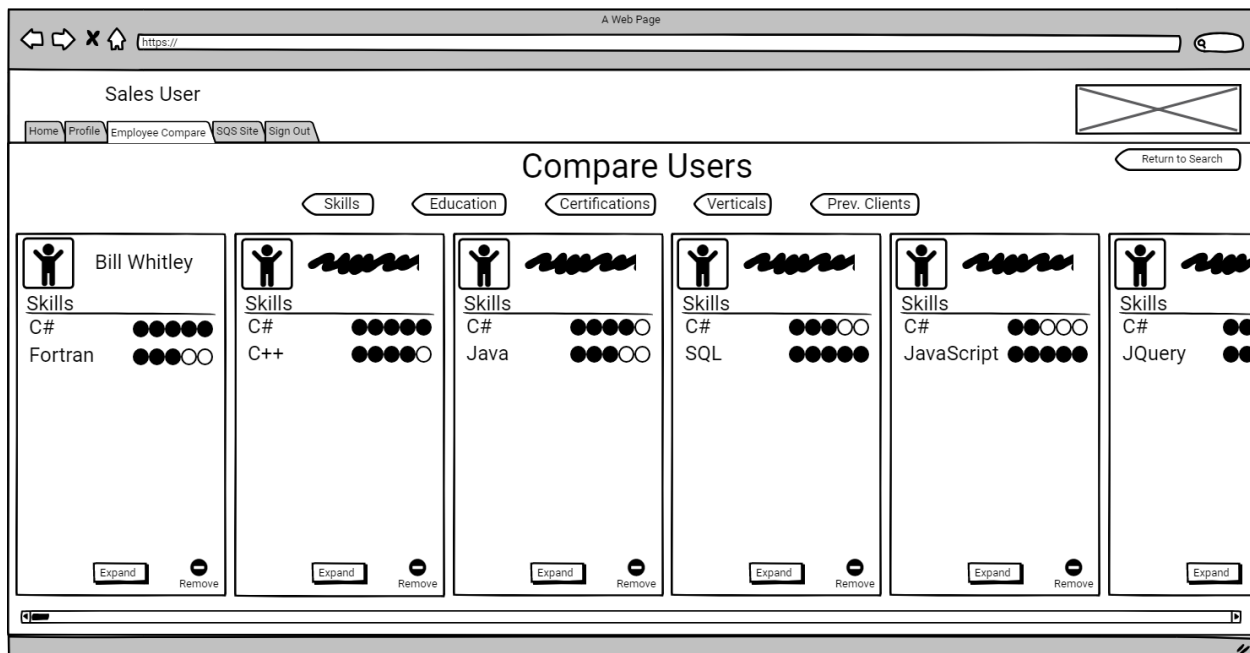


Figure 8: Compare Page

Testing

Most of our unit test cases will be done manually as automating the unit tests will be complicated with our project setup. A summary of the current known unit test cases are as follows:

Unit Test Description	Positive Outcome (Pass)	Negative Outcome (Fail)
User clicks the checkbox "I am a sales user" and presses the "Next" button.	The sales user is redirected to an "Awaiting Verification Page"	The general "error" page displays. OR The user is redirected to fill out the profile pages associated with a regular user.
An administrator user is sent an email notifying them that a Sales User needs verification.	The administrator user receives the email.	The administrator does not receive an email. OR The email is sent to a regular user or a sales user.
The administrator user clicks the link included in the above email.	The link successfully redirects to the Sales User verification page.	The link does not work. OR The link redirects to another page.
A sales or administrator user clicks the "Search Users" tab.	The user is redirected to the search page.	Nothing happens. OR Regular users can access the "Search Users" tab.
A user inputs a query and clicks the "Search" button.	Results that match the intended query are displayed.	No results are displayed. OR Results that do not fit the intended query are displayed. OR Results that should match the intended query are not displayed.
A user clicks the "Clear Search" button.	The query boxes are cleared, and previously displayed results disappear.	The query boxes do not clear. OR Previous results do not clear. OR The user is redirected away from the page.
Sales user clicks "Compare" button.	Those users which were selected to be compared will be added to an array and POSTed to the compare page. If no users were selected, the compare button is unselectable.	The array is not generated. OR The data does not properly post from one page to the other. OR The correct list of users is not properly generated on the compare page.

	The compare page will generate a list of users showing relevant details.	
Sales user clicks to remove a search category from view.	The user tiles reformat and hide the respective information.	Nothing happens. OR Multiple pieces of information are hidden. OR Tile formatting is messed up.
Sales user clicks "Expand User" button at bottom of a user's tile.	Full profile information for the selected user is displayed in a modal.	Nothing happens. OR Some user information is missing. OR The wrong user profile is displayed.
Sales user clicks the "X" button at the top right of a modal.	The modal is closed, and the page remains as it was initially.	The whole page clears. OR The modal does not close.
Sales user clicks the "X" button at the top right of a user's tile on the compare page.	The selected user is removed from the compare page view.	The user is not removed. OR Multiple users are removed.
<p>A regular user attempts to add a skill (in either initial account creation or their profile page).</p> <p>A modal appears, prompting user to select skill level and years of experience.</p>	<p>If user adds necessary information, new skill and corresponding details are added to the database.</p> <p>Modal is closed.</p>	Data is not properly inserted into the database.

Review

On Sunday, March 3, 2019, our group met in person at Lexi's apartment at 3:00 PM. We went through the above sections. The group members that did not originally write the section read through and commented on the section. Different pieces were added and edited. This included rephrasing the introduction for the high-level architecture, the reformatting of figure 3 for clarity, and addition of more test cases.

Metrics

Our application consists of an estimated 258 total story points. We have broken these points up into 109.5 high priority points, 29 medium priority points, and 119.5 low priority points. Based on an arbitrary 1-point story, we estimate that 1 point is approximately equivalent to half an hour. With this estimation and the knowledge that we will encounter unforeseen circumstances, the project is expected to take no more 105 womanhours and 35 manhours—roughly 5 hours per person per week.

Though we do not currently have an accurate estimate of how many lines of code our project will have, we expect our project to involve writing over 4500 lines of code. While we have a lot of functionality to implement, the complexity of our system is low, but our project consists of 58 requirements as of the time this document was written.

We have detected 2 defects thus far: the installer for the program, which was given to us by last semester's senior design team, is faulty and has not behaved for us or SQS at all, and the previously implemented email verification function is error prone.

Project Page

<https://glas1.github.io/>