SW Engineering CSC648/848 Fall 2022
Milestone 2
Section 2 - Team 6

Garage.io

Requirements, Specs, Architecture, UI mock-ups

Thomas (tmichel@sfsu.edu)	**Team lead / Back-end lead / DBA**
Mohammad (malhabli@mail.sfsu.edu)	**Front-end lead**
Joe (jsand@sfsu.edu)	**Back-end Engineer**
Tyler (tfulinara@sfsu.edu)	**Back-end Engineer**
Nathneal (ngebre@sfsu.edu)	**Front-end Engineer / Github Master**
Young (yso@mail.sfsu.edu)	**Front-end Engineer**
Jiaming (jzhao19@sfsu.edu)	**Front-end Engineer**

Revision history

10/03/2022	PM section + High Level Architecture section

1. Product Summary

- Garage.io
- Unique features of Garage.io
  - One click application
  - Direct message between applicant and recruiter
  - Apply to multiple job internships at once
  - connect external links and upload files to profile
  - Favorites list
  - Tech job oriented and includes many tech startups
- http://ec2-54-82-238-201.compute-1.amazonaws.com/

Committed Functions:

1. Student
   a. Students shall use the search bar to search for jobs without being authenticated
   b. Students shall be able to register their own account
   c. Students shall be able to log into their account
   d. Students shall be able to log out of their account
   e. Students shall be able to upload a picture for their profile
   f. Student shall enable/disable alerts for matching job interests
   g. Students shall be able to display details of a job post by clicking
   h. Students shall have access to a “one-click” application for jobs, while recruiters shall be able to set boundaries to accept applicants from said applicants
2. Company
   a. Companies shall be able to register their own account
   b. Companies shall be able to log into their account
   c. Companies shall be able to log out of their account
   d. Companies shall be able to upload their company logo
   e. Companies shall submit multiple job posts to the platform
   f. Companies shall have the list of all their job posts available on the platform
   g. Companies shall access student applications for a job post
3. Administrator
   a. Administrators shall trigger alerts for matching job interests for every student that enables this option
   b. Administrators shall be able to log onto the administration dashboard
   c. Administrators shall be able log out of the administration dashboard
4. Search
   a. Job posts shall be searchable by job area
   b. Job posts list shall be filtered by category and by job type

2. Usability Testing

Search testing

Test Objective: Searching is one of the most basic functions within this web application, since it offers the customers a helpful tool to narrow down the results. Whether the searching function provides the user with convenience and whether it correctly responds to the user's input will make the first as well as the most important impression. Moreover, the validation for input is also necessary and should be tested before the final delivery. On one hand, we should restrict the length and content of user's input to avoid any inappropriate code injection. On the other hand, if the input is valid, we should display the items that approximately match user's search text. All of the test objectives mentioned above are actually trying to collect feedback from users, in order to help our development team bring better user experience for both existing and potential customers.

Test background and setup

System Setup: From the users perspective, the setup process is fast and simple. From the user's perspective we see that there would be no installation requirements. The user needs a computer with access to the internet, they will need to open their preferred browser (we have tested Opera, Chrome, edge, and Mozilla) so we recommend users use those browsers until we are able to try more browsers. The user will the n need to type in our URL and start working

Starting Point: The starting point of usage of the search function is the home page, when the user accesses the provided link they will be led to the search page where they will find the search bar at the top where they will be able to type in what they are looking for. If what the user enters is valid they will receive the search results otherwise they will be prompted to make valid inputs.

Intended User: There are two intended users. The user may be a returning customer with a search history, a new user who just signed up or an anonymous user who is browsing through the available jobs before deciding to sign up. Therefore the intended user is anyone who is interested in applying for the jobs listed weather or not they have an account already with garage.io

Url of system to be tested: http://ec2-54-82-238-201.compute-1.amazonaws.com/

Usability Task description

Task Number	Task for usability testing

1	search for a job as an anonymous user
2	search for a job as a returning user
3	search for a job with several filters used

For searching as a new user we will be looking at how quickly and accurately the users are able to navigate the website. We will look at what we need to highlight to make sure the essential parts are more clearly visible for first-time users

For returning users we will be looking at what information that they stored was likely to make them want to return, we want to make getting an account worth it by saving essential progress made so that the customers will be encouraged to keep visiting the website

We will be testing the search feature along with the filters to make sure we have not put up too many restrictions and that the users are able to come back with what they are looking for.

Lickert subjective test

	Strongly agree	Agree	Neutral	Disagree	Strongly Disagree
Search function is convenient to use					
The user interface is easy to use all functions					
Returning to the page is easy to pick up where you left off					

Feedback to improve product

3. QA test plan

Test objectives: We will be testing the overall user experience to check how they were able to navigate the page and find what they are looking for. We will see if their experience was positive, we will find out what they found to be frustrating.

HW and SW Setup

Server Host: AWS Free tier
Amazon Web Service - t2.micro: 1 vCPU, 1GiB RAM

Apac has been successfully set up, which is required for running PHP.
PostgresSQL has been successfully set up, which is necessary for running the database.
PHP has been successfully set up, which is the back-end programming language.

Web Browser: we will be using latest versions of Chrome, Firefox, and Opera for testing

Features

Validation of input: We will be testing the if input validation works as intended

Approximate matching: For text search, the results do not have to perfectly match user's input. Instead, approximate matching will be executed.

Responsiveness: By expanding or shrinking the browser, the list page of search results is responsive to the change in browser size

Test Plan
To make it easier for our users we are using a table of questions where they will be prompted to agree or disagree to varying degrees regarding their experience.

Questionnaire

I knew exactly what the forms on the Registration page were asking for

Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

I got the result I was looking for				
Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

I did not have to wait too long to get results for my search				
Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

I could easily find what I was looking for				
Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

Every page on the site loaded quickly				
Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

I made typos during the search but still received the desired result				
Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

The filters were adequate for removing unnecessary search results				
Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

All aspects of the web search were clear and distinct				
Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

4. Code Review: We are using pull request (a feature of Github) to review the code that we will be deployed on the production

Leads developers (Frontend & Backend) are reviewing the implementation and test by themselves locally if the code is working
If they approve, the code can be merged and deployed, otherwise some changes will be requested.

Code style:
Offers uniformity to the code created by different engineers.
Enables the creation of reusable code.
Makes it easier to detect errors.
Make code simpler, more readable, and easier to maintain.
Designed to boost programmer efficiency and generates faster results.

Feedback: We are having peers review our code to make sure it is clear to developers who are not involved in the building of the web page. We will use the review to make changes to make the code more understandable.

5. Security
We are protecting password in the DB by hasing it using Blowfish Algorithm
We are using input data validation on the the forms we have to:
Register Form: We make sure every field is filled and the passwords minimum length is at least 8 characters
Login form: Same as Register form
Validation of the edition of the comapny profile & student profile not done yet
Security on both frontend & backend

6. Self-check: Adherence to original Non-functional specs – performed by
team leads

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO): ON TRACK
2. Application shall be optimized for standard desktop/laptop browsers e.g., must render correctly on the two latest versions of two major browsers: DONE
3. Selected application functions must render well on mobile devices: ISSUE
4. Data shall be stored in the team's chosen database technology on the team's deployment server: DONE
5. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users : DONE
6. The language used shall be English: DONE
7. Application shall be very easy to use and intuitive: DONE
8. Google maps and analytics shall be added: ISSUE
9. No e-mail clients shall be allowed. You shall use webmail: DONE
10. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI: DONE
11. Site security: basic best practices shall be applied (as covered in the class): DONE
12. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development: DONE
13. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2022. For Demonstration Only" at the top of the WWW page. (Important so not to confuse this with a real application). : ON TRACK