SW Engineering CSC648/848 Fall 2022
Milestone 1
Section 2 - Team 6

Garage.io
Project description

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)	**Back-end Engineer / Github Master**
Young (yso@mail.sfsu.edu)	**Front-end Engineer**
Jiaming (jzhao19@sfsu.edu)	**Front-end Engineer**

Revision history

09/18/2022	Milestone 1 delivery version

Content and structure for Milestone 1 document for review

Executive summary

Currently many students struggle with recognizing which direction to take their education. Students have skills and passions that drive their everyday lives. However, many students aren't able to connect these skills and passions to their careers, as they are inexperienced and uninformed to all the different career opportunities that are out there in the tech industry. Our goal at **Garage.io** is to offer students a free online service that will inform students of job positions through a simple search of their interests.

Garage.io provides an online service that shall give users different tools to get access to current job positions in the easiest way. Our application will provide users with a way to search for current job positions based on technology trends that they may show interest in. Garage.io will allow users to filter out different jobs based on different skills and passions they have. After searching and filtering, users will be met with a detailed list of all available job positions that are organized in a way to provide accessibility and usability to any user regardless of prior experience or knowledge.

Presently, there are many platforms that allow users to search for job positions based on categories such as salary, location, and many more. We want to focus on students that may use these applications, but aren't able to search for jobs because they don't know exactly what they want to search for. **Garage.io** differs from these other job searching platforms by catering to those students that lack experience and knowledge of technological job positions that are currently available through easy to use filters.

As an online service that caters to students job searching in the technological field, **Garage.io** shall provide different companies an outlet to reach this specific audience of students. Companies shall possess tools that will allow them to create different job listings directly. These tools shall provide companies high accessibility and usability to allow them to dictate exactly how they want to convey their job posting to a variety of users.

Our vision at **Garage.io** is an online service that will lay out a pathway for students heading towards the tech industry by providing a service that informs these students exactly the different job positions they can obtain after their education that coincide with their interests. Garage.io shall also provide the most accessible and reliable tools to companies that would like to provide job postings that target our users that we at **Garage.io** are catering specifically to.

Our start-up team consists of different students that are currently building their knowledge and experience trying to get job positions in the tech industry. As a team, we understand first-hand the hardships that an inexperienced student may have when trying to land that first tech job. Each one of our members is a student in a position very similar to our target users. By understanding first-hand our users, we understand as a team exactly what problems students may have when looking for a job position. Our vision for **Garage.io** is to provide as many people like ourselves the easiest and fastest way to land that job that coincides with our interests.

Personas and main use cases

Use Case 1

Persona: David is a computer science major at SFSU looking for an internship before graduation

Use Case: David signs up with the SSO option from the create an account page, he uploads his resume on Garage.io and starts looking for listings through the search option which consists of bubble options to narrow down job opportunities. David checks the internship button and checks the paid internship button and is taken to the results page by a submit button. He then finds an opportunity that matches his qualifications and hits the submit resume button to the company.

Use Case 2

Persona: Jihane is a front-end engineer looking for junior positions after completing a bootcamp and having a project-centered portfolio.

Use Case: Jihane signs up using the “sign up with google” option, sets up her profile by adding a resume and linking to her portfolio. Jihane then starts to look for opportunities with the search bar option. She enters “Junior Front-End engineer”, and a few opportunities pop up. Jihane finds out she fits all of these roles and clicks the “apply to multiple” option in the corner and checks the listing she wants and then presses the “submit resume” button and it shows a pop up that she has applied to the chosen listings.

Use Case 3

Persona: Erica is a recruiter for a startup in the San Francisco Bay Area and wants to post a listing for a job opportunity on Garage.io.

Use Case: Erica is presented with the sign up page when she opens up the website. She realizes she needs to sign up as an employer instead of signing up as an

employee. She notices a clickable link that is titled “sign up as an employer” and clicks it. The page presents a form that asks for the company name, job opportunity, date and description. She completes the form, chooses what tags fit this listing, and publishes the listing.

Use Case 4

Persona: Ahmed is a hiring manager looking to review candidate applications for a position his company is offering.

Use Case: Ahmed logs in through the employer log in page, Ahmed then clicks on the envelope icon titled “Applications”, which redirects to a page that contains all the job postings Ahmed’s company has posted and each listing card has a bell icon with a number depending on the amount of resumes submitted. Ahmed clicks on the desired job notifications and scrolls through applicants to find the most suitable ones. He selects multiple using checkboxes and sends a message to them individually that includes the name of the applicant handled from the backend.

Use Case 5

Persona: Jonathan is a senior engineer who has applied to a lot of companies for a job opportunity.

Use Case: Jonathan signs in as an employee, and he’s greeted with a small pop up next to the bell icon that has the number of notifications next to it, the pop up states “you have new responses from x company and more!”. Jonathan clicks on the bell to be greeted with a list of previously applied opportunities. The ones with the green check mark indicate that this employer would like to speak more or call in Jonathan for an interview. The ones with the red X symbol are job opportunities that Jonathan has been rejected from. Jonathan chooses a listing with a green check mark and it opens up a chatbox to the employer that contains a message saying they would like to have Jonathan called in for an interview, Jonathan replies and is then sent a link to schedule a time for the interview.

Main data items and entities

Main entities:

- ➔ *ADMINISTRATOR* : **users that have all the privileges on the platform**
- ➔ *STUDENT* **: users that may apply to *JOB POST* on the platform**

➔ *COMPANY* : **users that may submit a *JOB POST* on the platform**

➔ *JOB POST* : **ad for a job that a *COMPANY* may submit on the platform and which a *STUDENT* may apply to**

➔ *SKILL* : **a competence a *STUDENT* may have & a *JOB POST* may require to apply**

Data items:

➔ *JOB AREA* **: the field in which a *JOB POST* is related**

➔ *JOB TYPE* : **the type of job a JOB POST is offering (Full-time, Part-time, Internship,...)**

High-level functional requirements

- Students shall use the search bar to search for jobs.
- Students shall be able to register their own account.
- Students shall be able to login into their account.
- Students shall be able to log out of their account.
- Students shall be able to upload their own image.
- Students shall be able to edit and correct the data after it is submitted.
- Students shall be able to search for jobs without login.
- Registration form: required for students to register. Contains name and e-mail, and optionally address, phone, and affiliation. Stored in the database.
- Customer transactions shall be executed via a secure internet connection.
- Any users shall be able to browse by category and search from text, trends, date added, and location.
- Student profile page shall have incorporated areas to post/store a resume, cover letter, certificates, and so on.
- Companies shall be able to create an account and can access the resume.
- Students shall be able to add links to their profile, such as LinkedIn.
- Students shall have the favorite list, filter by the application deadline, and add the favorite list.
- Users shall have their unique identity to get into their accounts.
- Students shall be able to have all the information on the number of jobs they applied for.
- Students or companies shall report any problem they have to the app team.
- Students shall check the recorded history to show what they have searched for.

- 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.
- Students & Companies shall have an option to reset their password when they can’t log in to their accounts.
- Students shall be able to apply for multiple jobs/internships at once.

High-level non-functional specifications

➔ 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).

➔ Application shall be optimized for standard desktop/laptop browsers e.g., must render correctly on the two latest versions of two major browsers.

➔ Selected application functions must render well on mobile devices.

➔ Data shall be stored in the team’s chosen database technology on the team’s deployment server.

➔ Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.

➔ The language used shall be English.

➔ Application shall be very easy to use and intuitive.
➔ Google maps and analytics shall be added.

➔ No email clients shall be allowed. You shall use webmail.

➔ Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.

➔ Site security: basic best practices shall be applied.

➔ Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.

➔ 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.

Competitive analysis

	Garage.io	Indeed	LinkedIn	Glassdoor
"One-click" application	✔	✔	✔	✔
Direct Messaging between applicants and recruiters.	✔	✔	✔	
Apply for multiple jobs/internships at once.	✔			
Connect external links and upload files to profile.	✔		✔	✔
Favorites list	✔	✔	✔	
Tech Job Oriented with Tech Startups	✔			

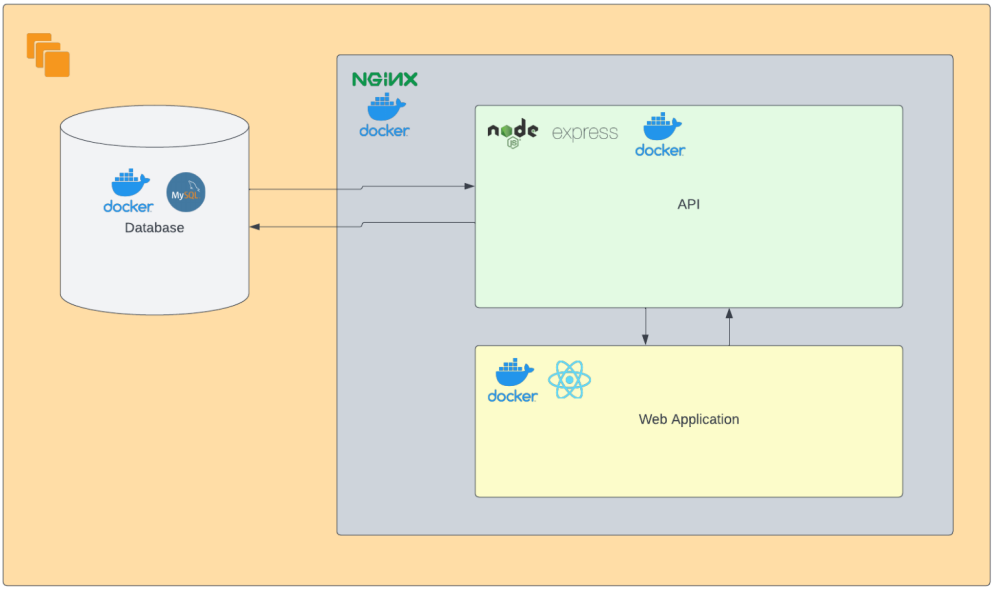
Our product shall compete amongst the most popular job search web applications with even more ease and accessibility for both the student and company. Garage.io will maintain the "One-click" application standard in the industry by allowing people to efficiently apply for jobs with a simple click. We go beyond our competitors by allowing students to select multiple opportunities and apply for those positions at once. student profiles are customizable and keep everything in one place by allowing users to connect external links and upload files to best show their qualifications. Unlike most job sites we value the importance of communication between the recruiter and applicant and our application shall allow direct messaging for important questions and dialogue. Students shall be able to add job/internship opportunities to their favorites list to keep a close eye for updates and deadlines. Our strongest advantage is that our web application shall be tailored for tech job opportunities allowing for tech startups to quickly recruit and applicants to find a job in the tech industry. Ultimately, Garage.io maintains the key features that users love in the most popular job searching applications and adds superior functions that serve both the recruiter and applicant to create a pleasant and effective experience.

High-level system architecture & technologies used

We designed a simple & efficient system architecture to create a reliable service.
We have chosen to deploy our application on a server provided by **AWS EC2** running **Ubuntu 22.04**.
The application is divided into **3 parts**:

- **Database**: **MySQL v8.0.30**
- **API**: **Node.js** API made using **ExpressJS** framework
- **Web Application**: developed using **React** Framework (JavaScript) & **TailwindCSS 3.1.8**

The interactions between API & Database will be done using Prisma ORM.
The API & Web application will be available thanks to a **Nginx 1.23.1** web server.
Docker will be used to containerize every piece of the architecture.



Team and roles

Firstname	Role(s)

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)	**Back-end Engineer / Github Master**
Young (yso@mail.sfsu.edu)	**Front-end Engineer**
Jiaming (jzhao19@sfsu.edu)	**Front-end Engineer**

Checklist

- Team found a time slot to meet outside class: **DONE**
- Github master chosen: **DONE**
- The team decided and agreed together on using the listed SW Tools & deployment server: **DONE**
- Team ready and able to use the chosen back and front-end frameworks and those who need to learn are working on learning and practicing: **ON TRACK**
- Team lead ensured all the team members read the final M1 and agree/understand it before submission: **DONE**
- Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.): **DONE**