SW Engineering CSC648/848 Spring 2021

648-02 | Team 02 | ProSpector

Cameron Cirini (Team Lead) ccirini@mail.sfsu.edu,
Franklin Arevalo (Frontend Lead), Faisa Jama (GitHub Master),
Tony Cao (Backend Lead), Zhuojun He (Backend Support)

“Milestone 1”

02/23/21

Date Submitted	Date Revised
02/23/21	02/28/21

1. Executive Summary:

In our world, companies in every sector are constantly developing and competing with each other with a common goal of producing revolutionary products to serve their customers. As these companies evolve and grow, so does their need to hire more employees to progress the company's mission. For the last decade, the tech industry has become a very popular, exciting, and interesting field of study and work for many students and young adults; these people are either starting their own companies or looking to join a company that interests them. Because of this, one major problem that arises is the disconnect between candidates looking for a job and companies looking for their new potential employee. This is where ProSpector comes into play. We built our service with a single goal: fast and accurate connectivity for both the company looking to hire and the job searching candidate.

ProSpector will serve as the central hub for SFSU student employment in the Tech industry. Our application lets companies create a profile, which gives access to post job advertisements with their certain qualifications. Our algorithm will then match them with a potential candidate. Our service allows companies to search for candidates based on the requirements they are looking for; if the company is looking for a female rising college senior studying computer science with experience in data management, they will be shown a list of many potential candidates' profiles from which they can manually select. Candidates looking for a job in the High-Tech industry also have access to our services. Their profile will have a resume, video of themself, their demographic and/or major. As they wait for a match, candidates can look up job advertisements to see if any interest them. Both companies and candidates are able to sign up to be alerted when a match is made. ProSpector has many features that separates our application from others, one of which is by adding a third party aside from just the company and the candidate and that is college professors. Our service allows professors of these candidates to rate them and enter recommendations. By having these, candidates are able to stand out and companies will no longer ask for recommendation letters. Our algorithm is developed to produce fast and accurate matches; this will help shorten the wait time for both the company and candidate.

Our team is passionate about computer science and applying their knowledge to their daily lives. We all discovered a passion for CS at different times in different ways. Many of us enjoy working out through a variety of sports such as badminton, basketball, and soccer. Our team is composed of different levels and skill sets, and we are all eager to learn more about web development to deliver a great product. We have students from different backgrounds who bring in a lot of new exciting ideas. Within our team, we have members who all have excellent communication, accountability, and determination. In our free time, different team members enjoy traveling, exploring, cooking, gaming, and watching sports. In the future, we all hope to secure a stable job that allows us to grow and bring us joy.

2. Personas and main Use Cases:

Personas:

1. Alex is a SFSU student who is about to graduate this summer. He is planning to look for an internship or entry level job once he graduates. He heard that our web page is one of the good web-based services to help SFSU students to get faster to the job market targeting the High-Tech industry in the Silicon Valley area. So Alex decides to register and upload his profile there. He goes to our site and uses his email address as the account name to register. Our web system sends him a verification email to his email address and asks him to verify his email correctly. Alex checks his email and clicks the confirm link. The web gives him a response page to confirm that his account is successfully registered.

2. Allen is a new registered user on our site and tries to add his personal information and skills on his page, and hoping any employers can search and hire him based on his profiles. Allen first starts to upload his profile image on our site and hopes the employers will actually get a feeling what he looks like. Secondly, Allen start to add his personal information such as name, address, major, education, school, (especially, the periods of starting and ending, the activities or projects he did at school, the classes he took during school etc.), experience (what he did in the job, what is his responsibility, and what he learnt from the job), interest, honors, skills, and soon on at each different text boxes. Last but not least, he also uploads his resume and a small introduction video on the site.

3. George is a SFSU student looking for a new job. After he logs into our site, he first sees his profile page. He wants to search for a new job posting by using the key word, “software engineer”. After he hits the search key, the result of a list of matching jobs is shown on the center of the page. The result page has about 10 to 20 jobs on each page. Each section has information about the job title, corporation of the job, a short description about this job. George finds the one he is interested in and clicks into it. More detailed information about this job such as title, company, address, job description, responsibility, salary, time pop up on the detail job page. George finds this job is not right for him, he goes back to the search page and keeps searching again. After a few hours searching, George finally finds the job he likes and he uses the apply button at the bottom of the job page to show his interest in the company. The apply button may link to the company’s job apply page and give a notification to our registered employer users.

4. Bob is a SFSU faculty. And he accepts the invitation from Allen to write a recommendation in Allen’s profile on our site because Allen did an excellent job in his class last semester. Bob uses his professor account to log in to our website, and then uses Allen's student ID number to look for his profile. When Bob finds Allen's profile, he clicks in it. He saw some of Allen's personal information. He scrolled down the page and found a section for rating the student. Bob gives Allen a good score, and starts to write a letter of recommendation for Allen in the comment section. Finally Bob saves his work and exits our website

5. Billy is a HR of ABC tech company. He is looking for people who have the website building and maintain background to join his company. He knows that our website is a good place to look for new graduate students, because this website provides evaluations from school professors on students. After logging into our website with an employer account, he uses the keyword "full stack software engineer" to find the qualified students. A lot of qualified students are displayed on the page, including a brief description about the student. He begins to click and look into it one by one. In each student's page, he saw the student's profile, such as name, education, work experience, teacher rating and comments (if there is any). After careful screening, he found some very good students, and used the “contact” option under the page to send invitation emails to those qualified students.

6. Samantha is a rising Senior attending SF State who is majoring in computer science. She goes on our website to look for a summer opportunity for some professional experience. She finished setting up her account and configured her profile to reflect her experiences. One week has passed and Jim, who is a recruiter for a startup company based in San Francisco, goes onto our site to look for a potential candidate who is a female in computer science to come work with him and his team at the startup. As Jim searches our website’s database, he checks “female” and “computer science” as a requirement and a list of candidates shows up. Jim decides on Samantha's profile and requests to connect with her. Samantha instantly gets an email notification from Jim saying he wanted to interview her for a potential role at his company.

Use Cases:

1. **All users** shall sign up to use the application.
   a. Once the user is on the home page, they shall click the sign up button. The system shall direct the user to a student, recruiter, or instructor

sign up page. Once the sign up page is loaded, the user shall provide their email, desired password, and any other needed information. Once completed, they have successfully signed up.

2. **Students** shall create a profile with resume, video and tags.
   a. Once the student is on the home page, they shall click the button to sign in. The system shall move the student to their home page which shows their matches. The student shall click the account button to be directed to their profile page. The student shall customize their profile with information, and have the choice to upload resumes and videos.

3. **Recruiters** shall post jobs with certain qualifications.
   a. Once the recruiter is on the home page, they shall click the button to sign in. The system shall move the recruiter to their home page which shows their matches. The recruiter shall click the button to create a new job listing. The recruiter shall add information about the job, and their qualification in demographics, experience and education.

4. **Professors** shall be able to rate students in a scale from 1-5 fashion
   a. Once the professor is on the home page, they shall click the button to sign in. The professor shall be able to click the search button on their home page. The system shall then direct the professor to the advanced search page. Once the professor finds the student he wants to rate, he shall press the recommend button on the student's profile. The system shall move the Professor to the rating page. The professor shall be asked to rate in a scale from 1-5 fashion about the student's knowledge, responsibility, teamwork, leadership, and commitment to success.

5. Once the system makes a match, it shall send an **alert** both the **student** and the **recruiter**.
   a. The system shall match the students with a job listing based on the information given by the student and the qualifications for the job. The system shall then send a notification to both accounts. Once the matched student or recruiter is on the home page, they shall click the button to sign in. The system shall ask for their email and password. Once signed in, The system shall direct the matched student or recruiter to the alerts page. They shall see a list of alerts. They shall be able to accept or decline the offer.

3. List of main data items and entities:

We have 3 main user entities which are the **students**, **recruiters** & **professors**. We then have 3 sub entities as support which are **addresses**, **companies**, & **posts**. The tables below indicate the different attributes corresponding to each entity.

Student Table

userId (int)	User Name (string)	Password (string)	Student Id (int)	first Name (string)	last Name (string)	address Id (int)	Email (string)	Resume (pdf.doc etc.)	Ethnicity (string)	Major (string)

Recruiter Table

userId (int)	Username (string)	Password (string)	firstName (string)	lastName (string)	Email (string)	Company (string)	Post Id (int)

Professor Table

userId (int)	Username (string)	Password (string)	firstName (string)	lastName (string)	Email (string)	University (string)

Address Table

addressId (int)	streetName (string)	APT (int)	City (string)	State (string)	zipCode (int)

Company Table

companyId (int)	companyName (string)	Description (string)	Logo (png, img etc.)	address Id	websiteLink (url. string…)

Post Table

postId (int)	jobTitle (string)	Description (string)	Salary (double)	Time (date)	addressId (int)

4.Initial list of functional requirements:

1. **Students** shall be able to **sign in**.
    a. If users of our application have a registered account with us, they shall be allowed to sign in to their profile and access our services.
2. **Students** shall be able to **sign up**.
    a. If users of our application do not have a registered account with us, they shall be able to register themselves by providing necessary information. After, they shall be about to sign in and access our services.
3. **Students** shall be able to **upload resumes and videos** and enter any demographics, experience and education for their profile.
    a. If users who are looking for a job use our application and they have a registered account with us, they shall be able to upload their resumes, personalized videos, and/or enter any demographics, experiences and education to their profile.
4. **Students** can **update** their profile information.
    a. If users who are looking for a job use our application and they have a registered account with us, they shall be able to update their profile information at any time without restrictions.
5. **Students** can **retrieve their password** if forgotten.
    a. If users who are looking for a job use our application and they have a registered account with us and they have forgotten their account password, they shall be able to request to either change their password or retrieve their old password at any time without restrictions.
6. **Students** shall **receive alerts** to get ready for interviews.
    a. If users who are looking for a job use our application and they have a registered account, they shall be able to receive alerts when they are matched with a potential employer to get ready for their interviews.
7. **Professors** shall have the ability to **rate students** in a scale from 1-5 fashion (being 5 being the highest ratings that implies knowledgeable, responsible, teamwork, leadership, committed to success, etc).
    a. College professors of registered candidates shall have the ability to rate their students for knowledge, responsibility, teamwork, leadership, committed to success, etc).
8. **Professors** shall have the ability to **enter recommendations**.
    a. College professors of registered candidates shall have the ability to write any recommendations for candidates.

9. Tech **Companies** shall have the ability to **register** to the portal.
    a. Companies using our services to search for potential hires shall be able to register themselves onto your application.
10. Tech **Companies** shall have the ability to filter candidates with **advanced search**.
    a. Registered companies of our application shall have the ability to search for candidates with the option for advanced search options based on major and demographics.
11. **Advanced search** can be done on major and demographics.
    a. Registered companies of our application using our advanced searches shall be able to search based on major and demographics.
12. Tech **Companies** shall have the ability to **view applications**.
    a. Registered companies of our application shall have the ability to view candidate applications.
13. Tech **Companies** shall have the ability to **download resumes**.
    a. Registered companies of our application shall have the ability to download candidate resumes.
14. Tech **Companies** shall **receive alerts** for any and all profiles that match their job listings.
    a. Registered companies of our application shall be alerted if and when a match has been made by our algorithm.
15. Tech **Companies** can get more than one alert for the job listed.
    a. Registered companies of our application shall get more than one alert for a job listed if there are multiple matches.
16. **Admin** shall have the ability to approve **company registration**.
    a. Companies who register to use our services will need to be approved by our admins.
17. **Admin** shall have the ability to **view all profiles**.
    a. Admins shall have the ability to view all profiles registered on our application from candidates to professors to companies.
18. **Admin** shall have the ability to **view all job listings**.
    a. Admins shall have the ability to view all job postings made by registered companies.
19. A **Verification email** shall be sent to the user whenever he/she registers for the first time.
    a. Users of our application registering for the first time shall get a verification email sent to them.
20. All **Users** shall have the ability to **delete their profile**.
    a. All registered users of our application shall have the ability to delete their profile and close their account at any time.

5.List of non-functional requirements:

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).
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. Selected application functions must render well on mobile devices
4. Data shall be stored in the team's chosen database technology on the team's deployment server.
5. No more than 100 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
7. The language used shall be English.
8. Application shall be very easy to use and intuitive.
9. Google maps and analytics shall be added
10. No email clients shall be allowed. You shall use webmail.
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
12. Site security: basic best practices shall be applied (as covered in the class)
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
14. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2021. For Demonstration Only" at the top of the WWW page. (Important so not to confuse this with a real application).

6.Competitive analysis:

Feature	LinkedIn	Indeed	Xing	Glassdoor	ProSpector
Job Seeker Languages	++	-	+	-	+
Upload Resume	-	+	-	+	++
Endorsements	+	-	-	-	++

/Reference					
Job Seeker demographics	-	-	-	-	++
Expected Salary	-	++	+	-	+

+feature exists; ++ superior; - does not exist

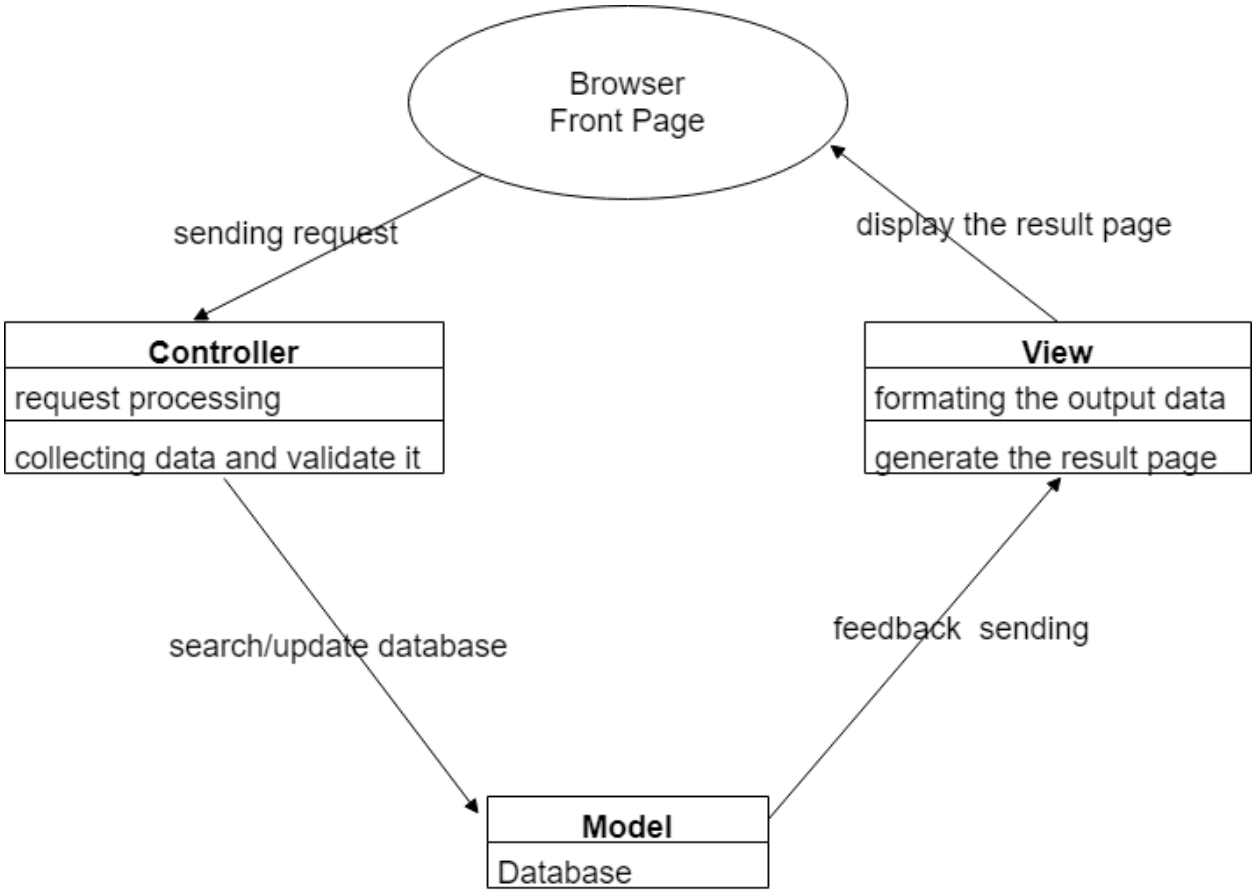
Summary:
After analysing each app individually, a list of relevant features from all these apps were compiled. Giving the option to upload an existing resume makes the process less time-consuming for users. For users to show their accomplishments, ProSpector shall give users the ability to specify the language they prefer to work in. ProSpector shall have the feature that lets certains certified users leave recommendations about what a user could do, to make their profile more attractive to recruiters. Unlike any other competitive products, ProSpector shall let users specify their demographics to stand out between other users. From the analysis, we see that 3/5 apps have the expected salary field as well, which can be a good addition for users with professional experience.

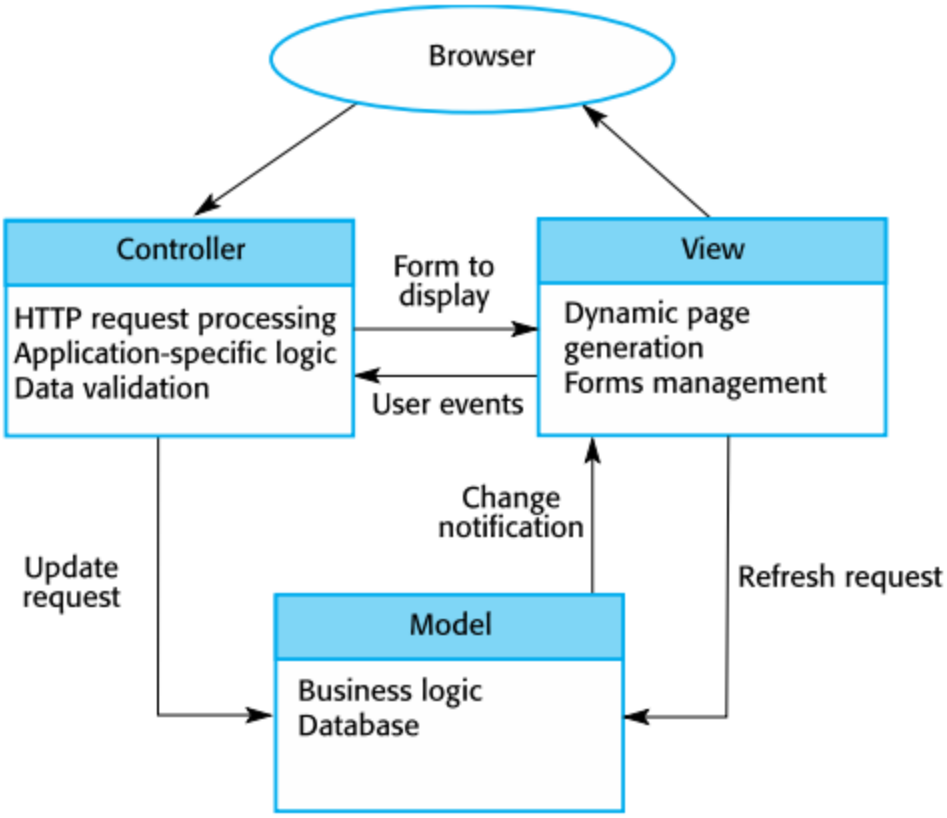
7.High-level system architecture and technologies used:

Below is a list of the technologies used in Team02's software stack:
Server Host: AWS t2.micro instance
Operating System: Ubuntu 20.10
Server Database: MySQL 8.0.23
Web Server: NGINX 1.19.6
Server-Side Language: JavaScript
Additional Technologies:
- Web Framework: React
- IDE: VSCode
- Web Analytics: Google Analytics
- SSL Cert: Lets Encrypt w/Cert Bot
- SASS: 3.6.4
- React Bootstrap: 1.5.0
- NodeJS: 15.6.0
- ExpressJS: 4.17.1

We will use the Model View Controller architecture. The front page will collect inputs from the users and then send them to the controller. There are a bunch of

node.js which will handle different requests processing, data collection and validation etc. And it also gives direction on how to manipulate our database. Database is stored alone in the model component. And it sends out feedback to the view section, based on the result from the request processed. And finally, the view section generates the result page and sends it to the browser again.





8.Team and roles:

Cameron Cirini	Team Lead
Faisa Jama	Github Master
Tony Cao	Backend Lead
Franklin Arevalo	Frontend Lead
Zhuojun He	Backend Support

9. Checklist:

Meeting Time Outside of class	DONE
Github Master Chosen	DONE
SW tools and Deployment Server agreed upon	DONE
Team ready and able to use chosen tools	ON TRACK

All members have read M1 and agree/ understand upon details	DONE
Github organized as discussed in class	DONE