SW Engineering CSC648/848 Spring 2022
LinkedSF - Section 02 Team3

Milestone 1 (22 FEB 2022)

Christian Mcglothen (Team Lead) (cmcglothen@mail.sfsu.edu)
Jeff Lo (Back End Leader)
Briget Soriano (Front End Leader)
Dominique Dutton (Front End)
Richard Li (Github Specialist)
Justin Yee (Back End)

Date Submitted	Date Revised
22 FEB 2022	

Executive Summary:

As the software industry grows rapidly these years, the increase of job opportunities offered by companies and students graduating from engineering backgrounds is obvious and gradually beyond control of current recruiting platforms. We figure out searching/matching jobs just by filters and texts is no longer efficient for tremendous jobs and applicants the market has right now. Also, paragraphs of details about the offer without highlights are causing a huge burden to applicants and mislead applications to the companies. Thus, we propose our product: LinkedSF, a fast and accurate recruiting platform with a concise interface.

More than filters and string matchings, LinkedSF provides both employer and employees a precise list of recommendations based on their characteristics and the character preference learned from past experience. The users will not have to be drowned in large but redundant searching results and they will not miss any possible desired match due to multiple filters that are used to narrow down the amount of matches. LinkedSF believes matching should be easy and efficient. Unlike how the current platforms display a swarm of words in the info page, LinkedSF displays customized info pages of the users' own choice. What the users receive is always what they want to know. Users' experience will from then be changed from onerous to brisk.

As students studying computer science, our team feels not only for the rapid growth of the industry but also the problems it might bring. By presenting LinkedSF, we hope it can bring practical benefits to both employees and employers while we are doing what we love.

Personas and Main Use Cases:

- **Student Persona**
  - About:
    - Finishing up Bachelor's degree in Computer Science at SFSU
    - Knowledgeable on Artificial Intelligence algorithms and concepts
    - Proficient communicator/good soft skills
    - Frequently uses job search sites such as LinkedIn
  - Goals & Scenario:
    - Seeks job opportunities for post-graduation
    - Has difficulty navigating and refining searches with other sites

        - Wants to constantly be up-to-date and informed on job information and statuses
- **Employer Persona**
    - About:
        - Hiring manager at a blockchain tech startup (amongst other things)
        - Very time-constrained due to multiple responsibilities in the workplace
        - Eager to bring new people onboard
    - Goals & Scenario:
        - Wants to present job opportunities to large group of potential hirees
        - Has had trouble finding people, due to limited reach and credibility of being a new company
- **Administrator Persona**
    - About:
        - Faculty member at SFSU
        - Aware of job offer scams occurring in the school email system
    - Goals & Scenario:
        - Wants to help students find lucrative employment opportunities
        - Seeks to ensure low-quality and/or deceptive “offers” are not present within the system
- **Student Use Case 1**
    - With graduation quickly approaching, Jordan is interested in finding some tech employment opportunities. He navigates to our site, LinkedSF, and searches for a job within his field. He then proceeds to filter those results further to only those he has the prerequisite skills for. After finding a job he thinks he is both qualified for and interested in, he proceeds to look at the information in the listing, perhaps to apply in the near future.
- **Student Use Case 2**
    - Kate wants to remain updated on the state of jobs she has been interested in. She registers an account using the pertinent

information, such as email, and then logs in. Then, she navigates to the job page of interest, and enables alerts for it.

- **Employer Use Case 1**
    - Eager to bring new talent onto the team, Donovan is looking to post a listing. He logs into his account, then navigates to create a new listing. After entering the relevant information needed, he publishes the listing for public view.
- **Administrator Use Case 1**
    - George must ensure that the listings have appropriate information. He logs into his account with admin privileges, and proceeds to observe the newly created listings. Upon investigating one page, he notices some links which could not be verified to be trustworthy. He notifies the creator and prevents the listing from being publicly viewed until the links are removed.

List of main data items and entities: (Richard)

- **Student:** Newly graduate student who is looking for a job
- **Employer/Tech Companies:** Companies looking to hire new talent
    - **Fields/Areas:** 9 possible areas of expertise
        - **Job Titles:** Specific job position that is open
        - **Description:** Summary of tasks that the position entails
        - **Skills:** Minimum requirement for the position
    - **Hiring Manager:** Employee of the company looking through applications
- **Administrator:** Faculty member filtering untrustworthy job listings
- **User:** Either a student, employer, or administrator
    - **Basic Account Information:** Includes name, username, email, and password

Initial list of functional requirements:

- **Student**
  - GUI:
    - Create Registration Page for STUDENTS
    - Should Ask for Basic Info Such As (Username, Fname, Lname, Email, Password)
    - Once STUDENT is Logged In, GUI should Display a few job Posts.
    - Should have a search engine (Filter/Sort Job Positions)
  - Functionality:
    - Once a student has completed registration, they should be redirected to the student login page.
    - The completion of the registration form will populate the database with the following info (Username, Fname, Lname, Email, Password, Etc)
    - Students will be able to search/filter jobs based on tech area, job position, and skills (Parametric Search)

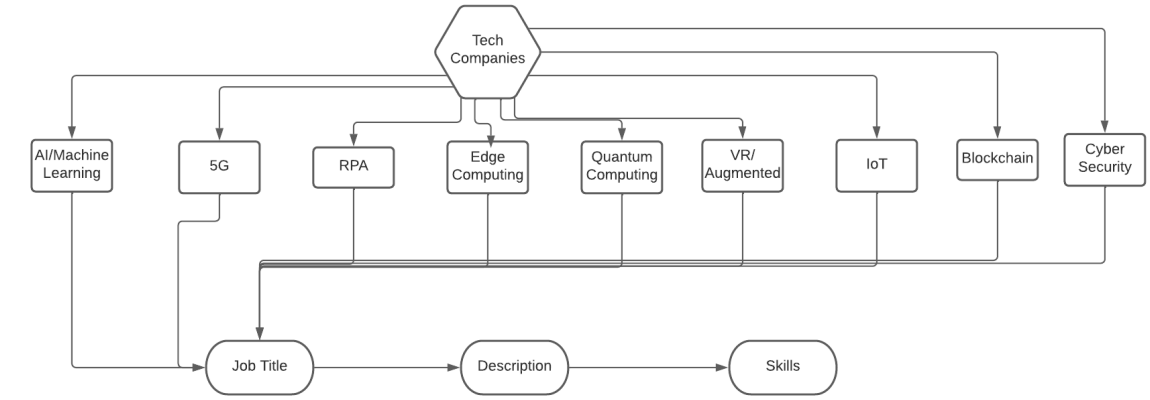
- **Employer/Company**
  - GUI
    - Will allow the Company to Sign-up/Register
    - Will contain fields such as (company name, email, username, etc)
    - Once the company has signed up, they will be redirected to the login page
    - Once they have completed login, they will be redirected to their homepage in which they will have the capability to create job posts and possibly see the amount of traffic their post recieved

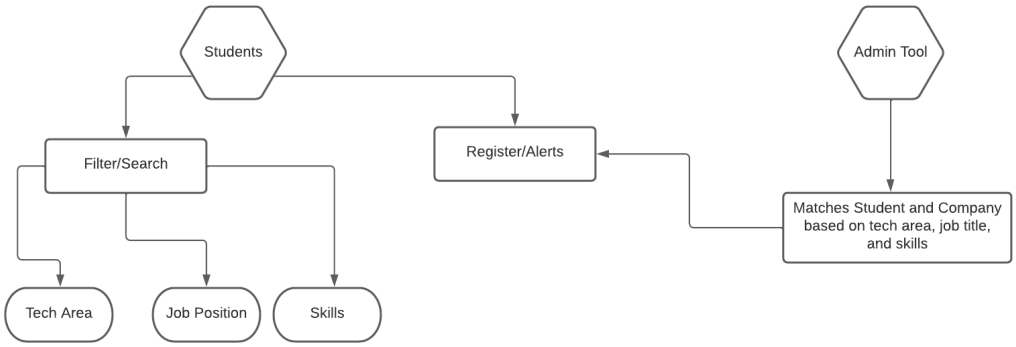
  - Functionality:
    - Each company will be allowed to post jobs in 9 areas (AI/Machine Learning, 5G, RPA, Edge Computing,

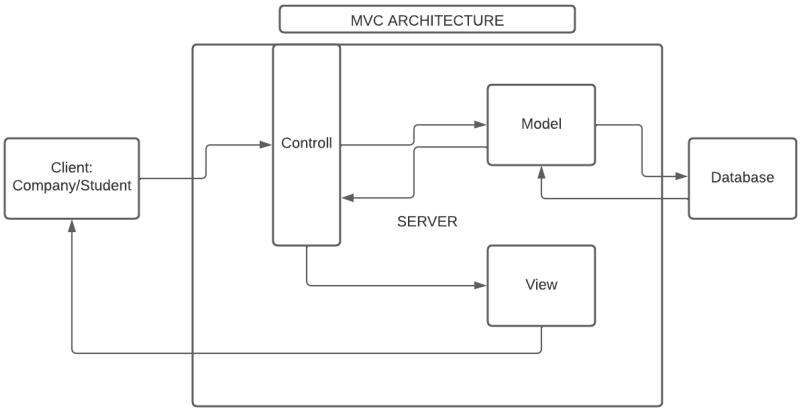
Quantum Computing, VR/Augmented, IoT, Blockchain,Cyber Security)
        - Each field area will consist of 3 specifications (Job Title, Description, and Skills)
        - Companies may also have the ability to filter their own posts

- **Admin Tool**
    - Functionality:
        - When a student and company have similar interests, both the company and the student will be altered/sent a notification
        - Manage student searches and companies ability to post jobs

Diagram of Functional Requirements:







List of non-functional requirements:

1. Applications 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. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
6. The language used shall be English.
7. Application shall be very easy to use and intuitive.
8. Google maps and analytics shall be added
9. No email clients shall be allowed. You shall use webmail.
10. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in the UI.
11. Site security: basic best practices shall be applied (as covered in the class) 12. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
13. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only" at the top of the WWW page. (Important so not to confuse this with a real application).

Competitive analysis:

Feature	**Nexxt**	**Indeed**	**ZipRecruiter**	**LinkedIn**	**Our Product**
Filter by position	-	+	+	+	+
Filter by skills	-	++	-	-	+
Job Alerts	+	+	+	-	+

Admin Alerts	-	+	++	-	+
Display Jobs in 9 Areas	-	-	-	-	++

Clearly shown in the table above, our exclusive job search in the tech industry is what makes us stand out from the rest. No other job seeking website has narrowed down job searching to these 9 areas: Artificial Intelligence and Machine Learning, Robotic Process Automation (RPA), Edge Computing, Quantum Computing, Virtual Reality and Augmented Reality, Blockchain, Internet of Things (IoT), 5G, and Cyber Security. Narrowing job areas to a specified number of areas may seem counterproductive, but this capability helps those interested in the tech industry to find their desired career paths. Other job websites come along with a chaotic atmosphere showing numerous job areas in senior level positions. Our product has shortened the job search to allow users to save time.

High-level system architecture and technologies used:

- **Server host:** AWS 1vCPU 1 GB RAM
- **Operating System:** Ubuntu 18.04
- **Database:** MySQL 8.0.23
- **Web Server:** Linux/Unix (EC2 instance) (Anaconda3)
- **Language:** Python 3.9, CSS, HTML
- **Flask IDE:** Visual Studio Code
- **Front End:** Bootstrap 5.0.1
- **Supported Browser:** Google Chrome/ Mozilla Firefox (current versions)

Team and roles:

Christian Mcglothen (Team Lead/Back End/Front End)
Jeff Lo (Back End Leader)
Briget Soriano (Front End Leader)
Dominique Dutton (Front End)
Richard Li (Github Specialist)
Justin Yee (Back End)

Checklist:

• Team found a time slot to meet outside of the class (DONE)

• Github master chosen (DONE)

• Team decided and agreed together on using the listed SW tools and 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 (DONE)

• Team lead ensured that all team members read the final M1 and agreed/understood it before submission (DONE)

• Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) (DONE)