Milestone 02 – 03/08/2022

SW Engineering CSC648 Spring 2022 Section 02

Team 06

Application Name: "GitJob"

Team: Allison (Team Lead), TJ (Github Master), Joshua (Front End Lead), Sedric (Front End Team Member), Thien (Back End Lead)

Email: aadad@mail.sfsu.edu

History Table:

Milestone	Date Submitted	Revision Date
M1	**02/22/22**	**3/06/22**
M2	**03/08/22**	

1. Functional Requirements Prioritized:

Priority 1:

1. Employers (companies, recruiters) shall be able to post job listings within these 9 fields: 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. Students shall be able to search for jobs within these fields.

2. Users shall be able to search and filter through job posts, through filters such as location, field (tech area), job position/title, skills, education level. Users shall also be able to find posts, workshops, forums, related to their search.

3. Employers shall be able to post (and delete) job listings that must contain a job title, description of position, qualifications, and skills necessary.

4. Registration form: required for users (including employers) to register. Contains name and e-mail, and optionally address, phone, affiliation, optionally location. Stored in the database.

8. Users will have a dashboard that comprises saved job listings, forum topics followed by the user, new job listings, and posts from companies/employers followed.

9. Users shall be able to explore the site without creating an account, but must create an account if they want to apply to jobs, post a job listing, reply to forums, attend workshops, and use other specific features.

10. Users who are logged in shall be able to save job listings, company profile pages, forum posts/topics, and posts from other users.

12. The application shall have a sticky navigation bar that holds dashboard, forums, profile page, so these features are easily accessible to user and do not overlap.

16. Content Restriction: the developers shall ensure that no inappropriate material (text or images) is presented on the site, forums, posts, etc..

17. User shall be able to edit and correct the data after it is submitted

18. Application shall not allows messaging between users.

20. System shall store job listings, user forum posts/replies, resume upload, into database.

Priority 2:

5. Users shall have a profile page that is viewable by other users or just employers. Users shall be able to upload the following to their profile page: previous job experience, contact information (optional), resume (optional), description of the user themselves, skills, and interests.

i. The profile may/may not be viewable by people on/off the app like on LinkedIn

ii. it may just serve the purpose for the user to interact and find job postings, but they may not have to customize their profile.

7. Forums: exist for users who wish to discuss anything tech-career-related, post questions about tech-careers, reply to questions, and have discussions in the forums. Will allow anyone to answer questions, post, and communicate through the forum. The forum is meant to be informal, sort of

like Reddit. Allows users to follow forum topics, which they can view on their dashboard as the topic updates with posts/replies.

21. Users shall be able to investigate Salaries for different type of job position

Priority 3:

5. Users who are registered shall be able to opt in for Alert Notifications (for job postings, forum replies/responses, etc.)

11. Users shall be able to 'like' forum posts, posts from employers, forum replies.

19. User shall be able to see the job listings they applied to under a section called 'Applications'.

22. Employers/Companies shall be able to have posts aside from Job Postings

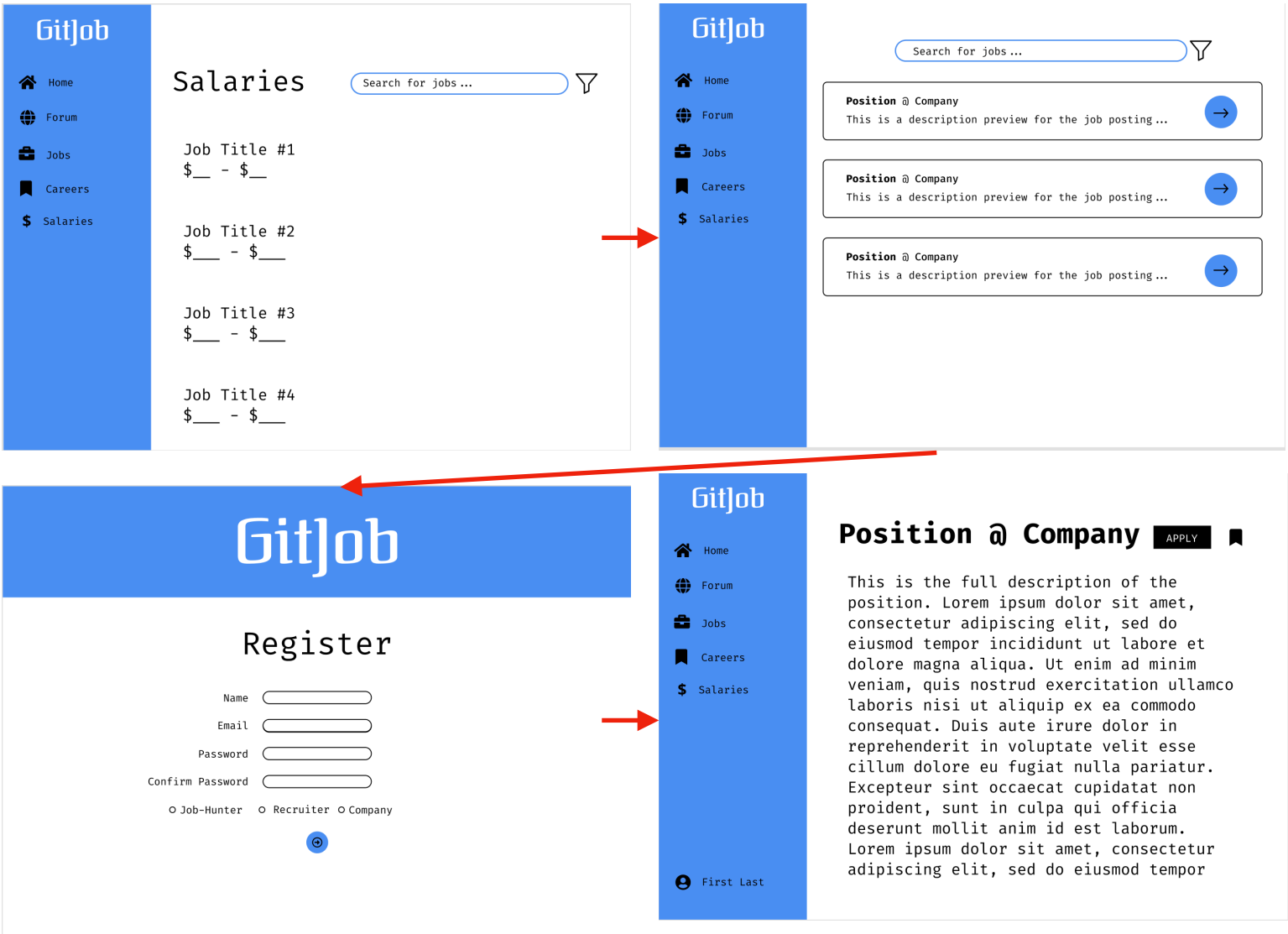
23. Users shall be able to comment/leave replies to forum posts, posted by other users

24. Users shall be able to follow companies/employers

2. UI Mockups and Storyboards:

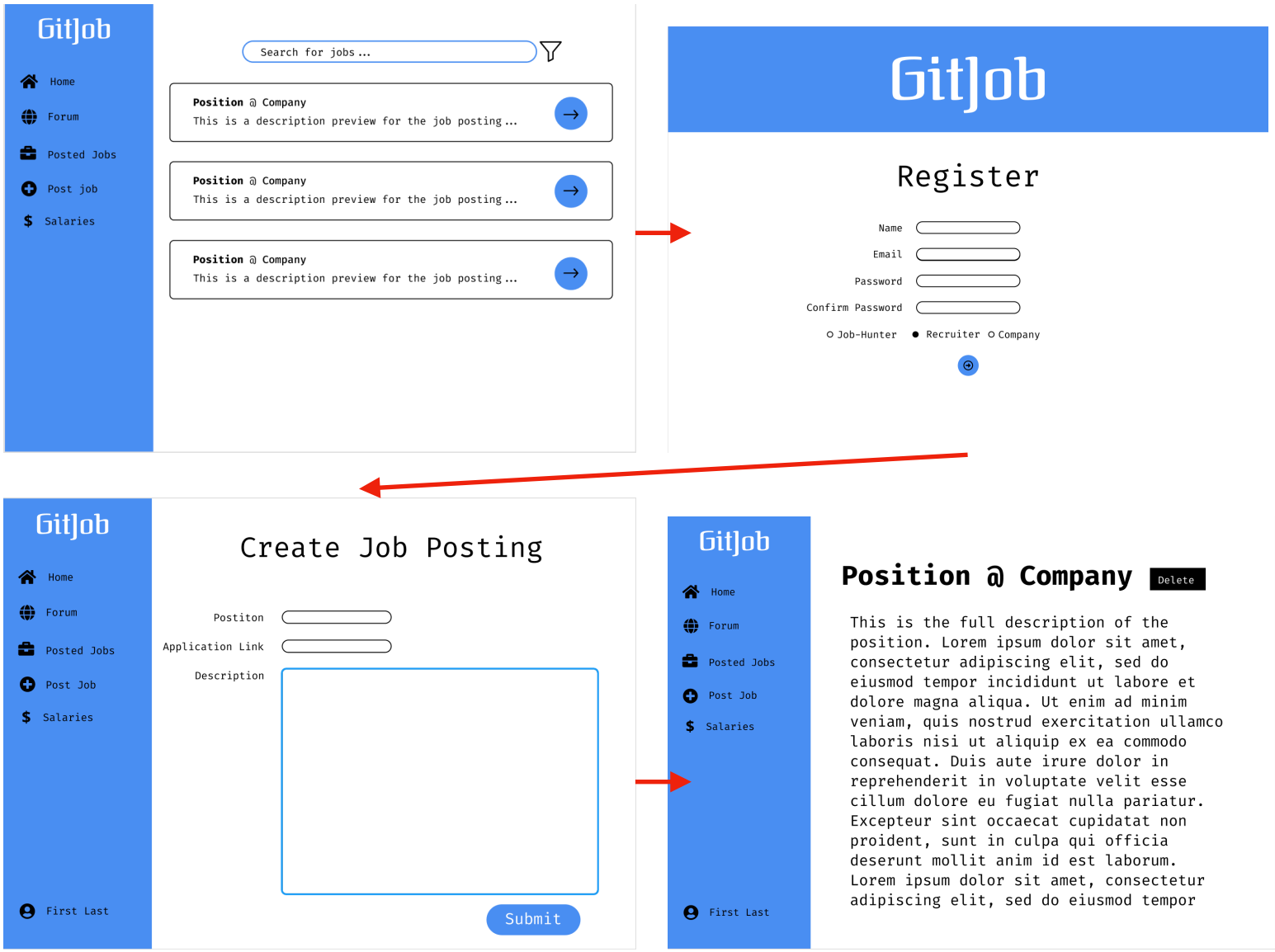
Main Use Case 1 :

‘X’ wants to search for an entry tech job involving autonomous integration. They use the career section on the site to explore the different jobs that one can attain with a CS degree in tech. Once ‘X’ clicks a job they are interested in. ‘X’ can use GitJob’s search feature to filter for jobs that contain keywords in their description related to their interests in social justice. They are prompted to register or log in, to access the job’s information/content. ‘X’ saves job listings they are interested in and follows companies they are interested in applying in. ’X’ can now view the saved job postings and easily apply.



Main Use Case 2 :

‘Y’ wants to search for candidates that meet the company’s requirements involving a robotics startup. ‘Y’ goes to our site and before being able to find where to create a post, she is prompted to register as an employer to be able to create a job post. Once logged in/ registered, ‘Y’ can fill out a form with the start-up’s job description and what is required for the position. ‘Y’ will get applications for the start-up. ‘Y’ will email back to the candidate that they want to interview. ‘Y’ can then delete the post if they have found a candidate.

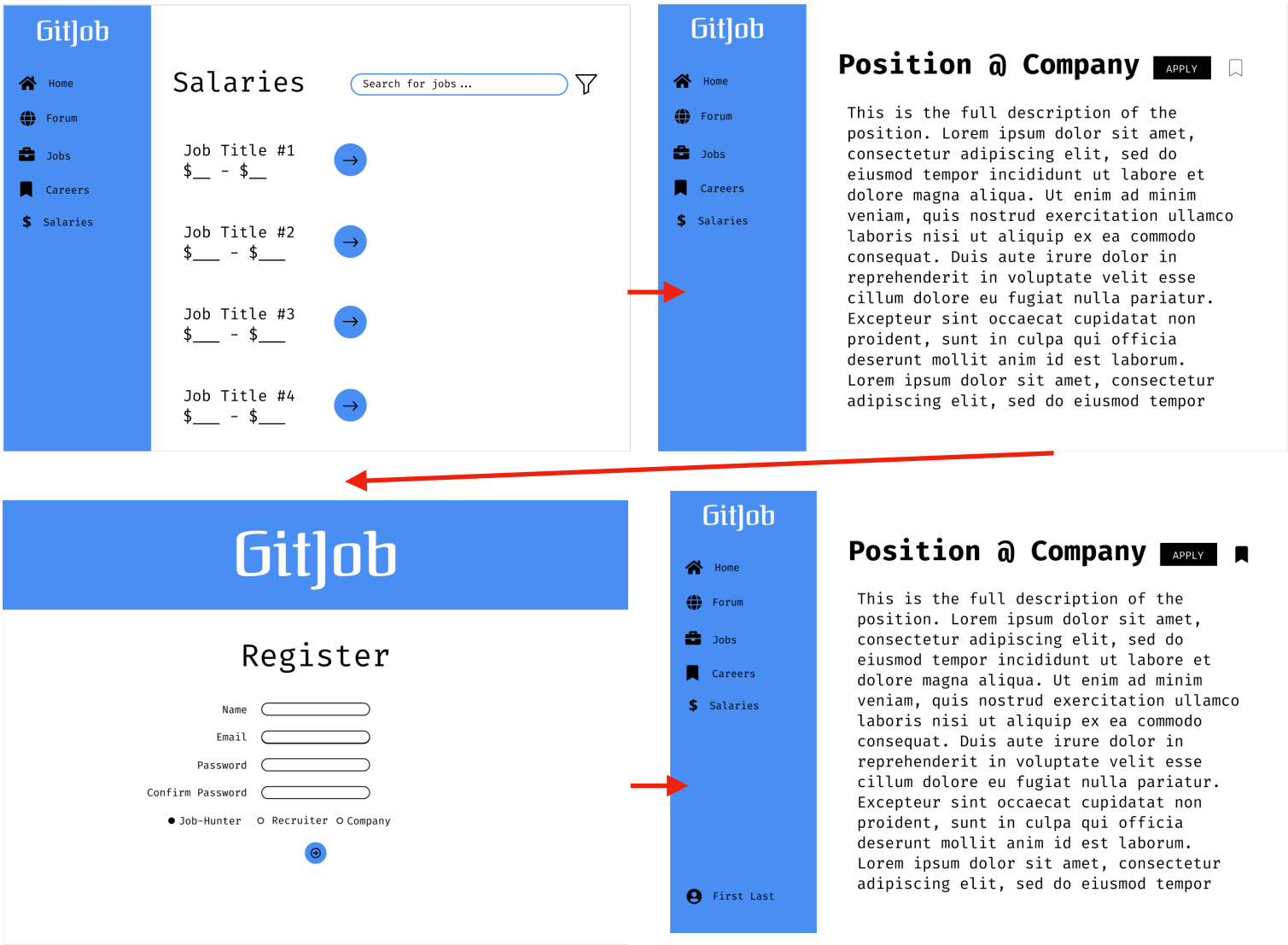


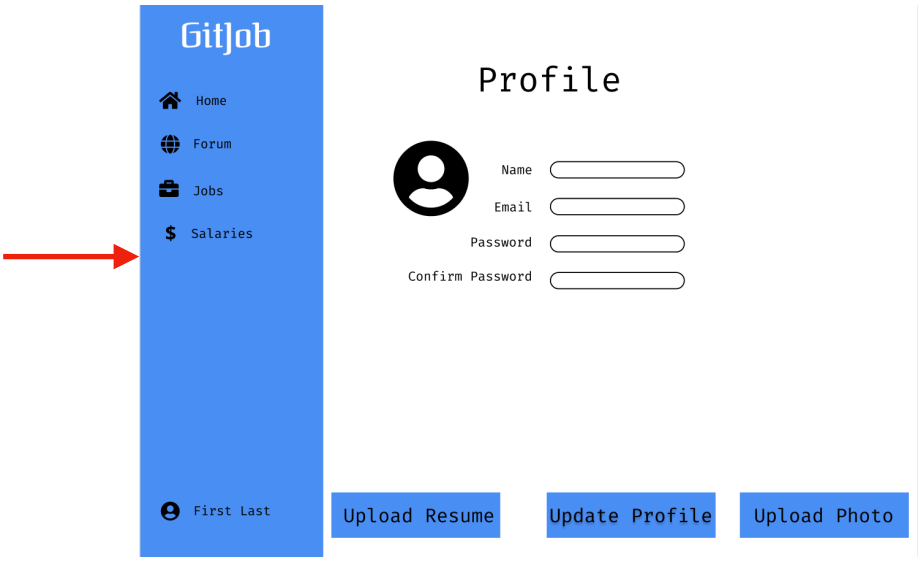
Main Use Case 3 :

‘O’ wants to search for a tech job that accepts interns or part-time employees. ‘O’ goes to our site and filters out the searches by “internship” and “part-time”. ‘O’ has found a job and clicks on the job posting and has read the description and realizes that he needs to learn some new technologies for the position. ‘O’ can now view the job posting even though he is not registered, they want to save the post, so they are prompted to register. ‘O’ takes advantage of the forum to find resources on how to brush up on his coding skills and finds workshops related to improving his resume. ’O’ can see the forum and can leave a comment. ‘O’ has improved his skills and has come back to apply to the job. ‘O’ eventually gets an internship.

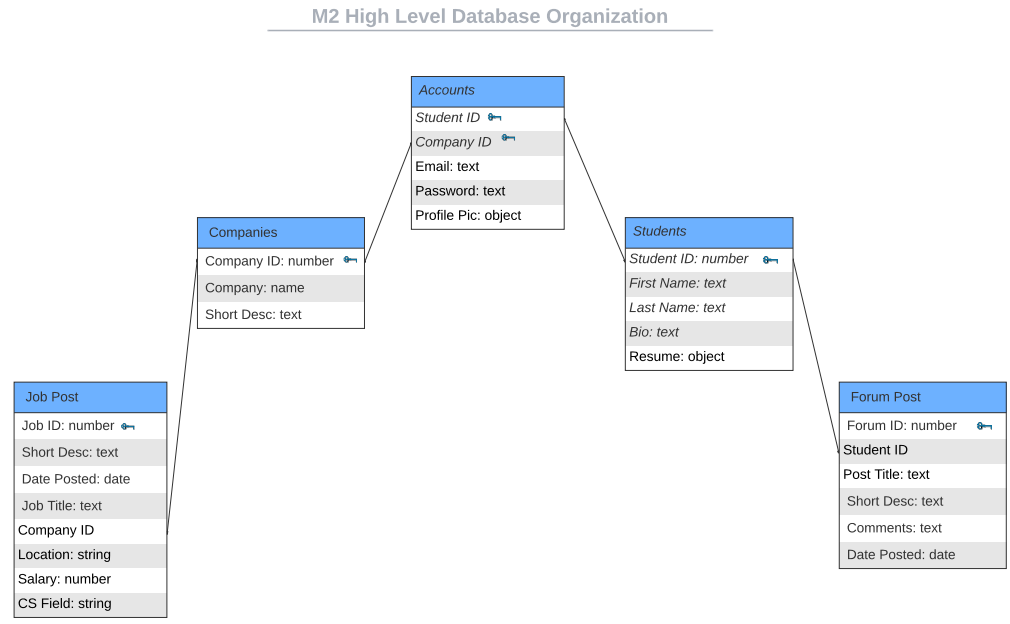
Main Use Case 4 :

‘Z’ wants to leave his former job to get a pay raise.’ Z’ is a part-time student at SFSU. ‘Z’ wants to look for a senior role in a tech company. ‘Z’ goes to our GitJobs Salary section and can compare salaries for “Senior Robotics Engineer” in different locations and different companies. ‘Z’ can view the pay rate for positions at specific companies and can see several job postings for “Senior Robotics Engineer” from different companies. He saves these job postings to compare them later. Before being able to save or click apply to the job, ‘Z’ is prompted to register or log in. ‘Z’ can now view and apply to the job posting once registered. Now that ‘Z’ has created an account, he is eager to get hired so he customized his profile page by uploading his resume, adding a description about himself, adding his skills and interests. ‘Z’ applies to 5 jobs that pay $10k+ more than his current salary





3. High Level Architecture, Database Organization:



https://bus206.pressbooks.com/chapter/chapter-4-data-and-databases/

In order to properly create this relationship, a primary key must be selected for each table. This key is a unique identifier for each record in the table. For example, in the Students table,

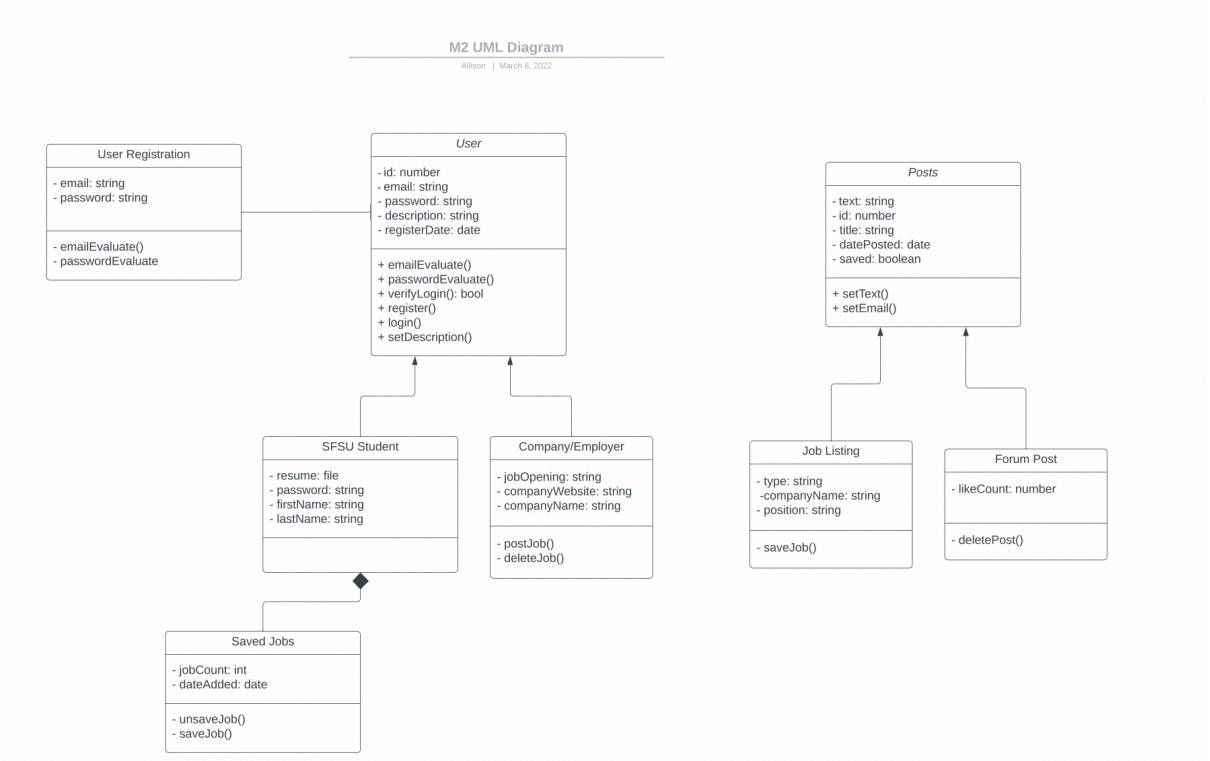
Media Storage:

There won't be able audio or video as of now but images (in pdf, jpeg, jpg format) will be kept in DB BLOBs.

Search/filter architecture and implementation:

Database terms that the user will be able to search by include: salary, job name, location, company, CS field (9 fields total that exist). These are all database terms that exist as fields for the Job Post Table. There will exist a Free Text Entry Field with a Pull Down that as a few search paramters (salary, location, company, CS field, Job Title) that is made with an SQL filter. It will be coded using SQL, and we will try to implement KW Search (keyword search) since our database contains keys.

4. High Level UML Diagrams:



5. Risks:

- *Skills risk:* There is some unfamiliarity regaaurding React. Also, the team feels comfortable with Javascript but is not at an expert level.
  - Solution: Team has agreed to brush up on the software stack so we are prepared to implement features when the time comes.
- *Technical Risk:* The forum feature may be too complicated to implement. We have not done something like this before in other classes.
  - Solution: We will research what it takes to implement a feature like this. Then we will gather and decide if we can allocate enough time for it, and prioritize other features.

6. Project Management:

M2 was divded amongst the front-end and back-end team. The front-end did the Story Boards, the back-end did the High Level Architecture portion. We are using Trello, now, to communicate dates, and we use discord channels for each team. Both teams came together at the end to review each others work and give any feedback. At the end of our meetings, which we usually have twice or more a week, we discuss what we are going to do next and priorities to focus on, or due dates coming up. The team-lead (myself) checks in with team members on progress periodically and assigned tasks to specific team members within each team.