CSC 648/848 Software Engineering

Fall 2022 Section 02

Milestone 2

September 20th, 2022

<u>JobHunter</u>

Submitted by:

Team 04 -
Muzaffar Sharapov - Team Lead
Jiayi Gu - Front End Developer
Julian Manaois - Database & Github check
Chengkai Yang - Front End Developer
Yagiz Saatci - Back End Developer

Date submitted for review:
Date revised after feedback:

Contents

1. Functional Requirements (Prioritized)

Priority List

P1	Critical (must have)
P2	Important (should/could have)
P3	Opportunistic (nice to have)

P1

1. Have a working application

   To develop a web-based application to help SFSU students to find their dream jobs and career paths in Tech. This application will help students with the search for their ideal job (career path) according to their skills and passion.

2. The web application must allow Tech companies to post jobs in 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. Each of these areas would have Job titles, descriptions, and skills as a minimum.

3. Allow to search & filter jobs based on tech areas, job positions, and skills.

   Pipe and filter architecture: Filter has independent entities called filters (components) which perform transformation on data and process the input from *the users. Pipes serve as connectors for the stream of data being transformed, each connected to the next component in the pipeline. Data Source -> Filter -> Filter -> Data Target.*

4. Users can get alerts for matching job interests, and Users can turn off alerts if needed.

   users can opt in/out to receive alerts from the application on job listings that they are interested in or have applied to. Users feel free to close it if they don't want to get the alerts from the application.

5. The site will have administrator capabilities to trigger the matching alerts to the corresponding users.
6. Have login and sign-up pages for users.

   For the sign-up page, we will check whether the email address user input is valid or not(to verify a proper email address from users, the application may require email confirmation as they received an email from the system) and check the username is taken or not, and also we will check the passwords' security level.

7. Each user will have a profile page.

   The page will show their skills, resume, job experiences, education, and links for other social media.

8. The application will *focus on desktop/laptop browsers, and not on native app development for mobile browsers*.

Customers will use the applications primarily via the web and using PC/laptops and occasionally via their mobile devices. The application will however have to be developed using responsive UI implementation so that it renders well on mobile devices.

9. The site will utilize a user database set (management system) for storing all user information and postings.

   Database management systems can connect well, and also be able to update the information while changing some data.

10. Recruiters will be able to view applicant profiles, post job listing, edit their job posting, contact applications about their job postings,

    Recruiters can also sign up their account

11. Users can change or edit their search on the same page.

    Application will have a navigation bar to make sure each page has the bar to let the user edit and search other jobs/users at the same page.

12. User credentials will be encrypted.

    User credentials are typically a username and password combination used for logging in to online accounts. However, they can be combined with more secure authentication tools and biometric elements to confirm user identities with a greater degree of certainty.

13. Users will be able to take Guidance Quiz when logged in.

Decide to do a Job Pathway advise pop-up window (Career Advice Robot). Users will be able to answer some questions to find the correct job direction.

14. Users can navigate the Helpful Links page without registering or logging into their account.

    Post job webpage will have the company's link and some details.

<u>P2</u>

15. Job postings will have direct links to company sites for interested applicants.

    Post detail (location, address, company size, link for the company's website, etc).

16. Search box to search for other users and follow them.

    Adding a searching box especially for searching the users, searching by name/account id/email/or others.

17. Users can search job listings without registering or logging into their account.

    All job posting web pages should be open for everyone. Edit personal profile, apply for a job, take a look at other user's profile and follow other users need to sign up an account.

18. Users will be able to see what listings they have applied for.

    Each job user applied will store into the database, and show as a list in the user's profile page. (optional to let other see this list)

19. Users can apply directly for job listings without filling out a third party application.

    Create a table to let users fill in their information(resume, contact information, etc), and store all information details to the database and also send a confirmation email to the user's email to confirm that job has been applied for successfully.

20. Users can contact recruiters from the job listing page.

    Show recruiters contact information, such as email.

21. Job application process will be streamlined by utilizing users' uploaded resume and follow-up questions if applicable.

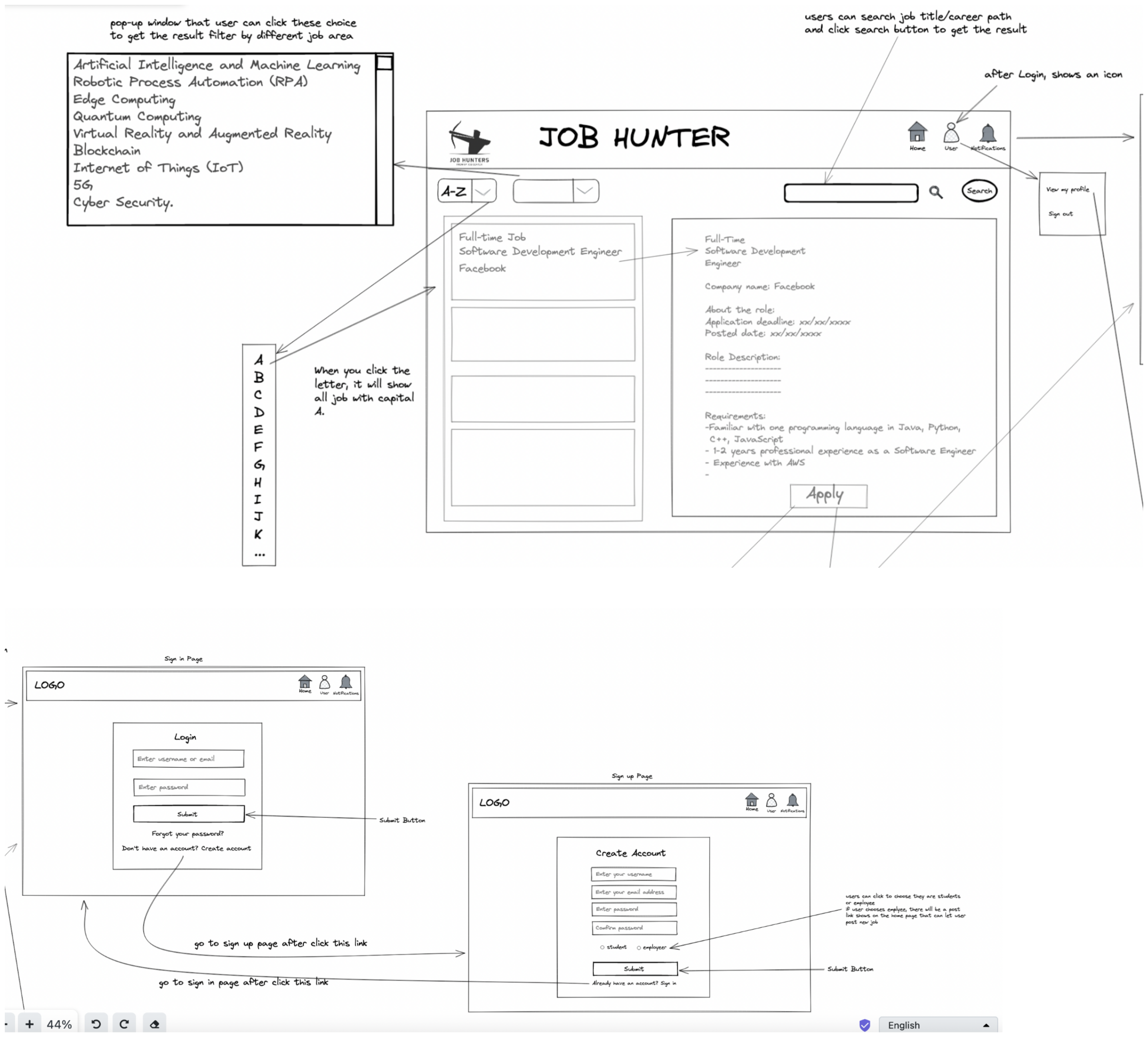
    Storing resume and personal information from the profile into the database to let users upload their resume easily when applying for a job position.

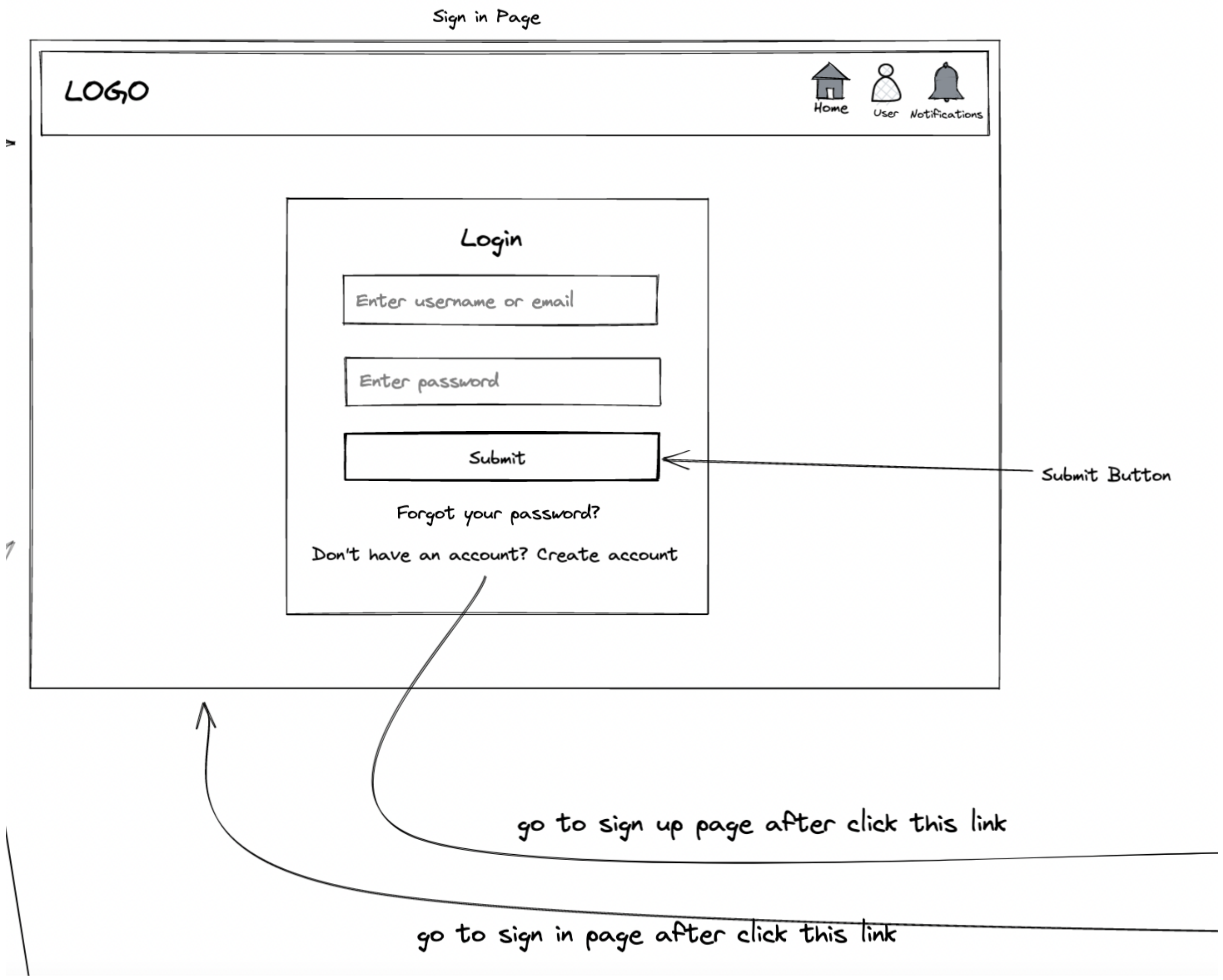
<u>P3</u>

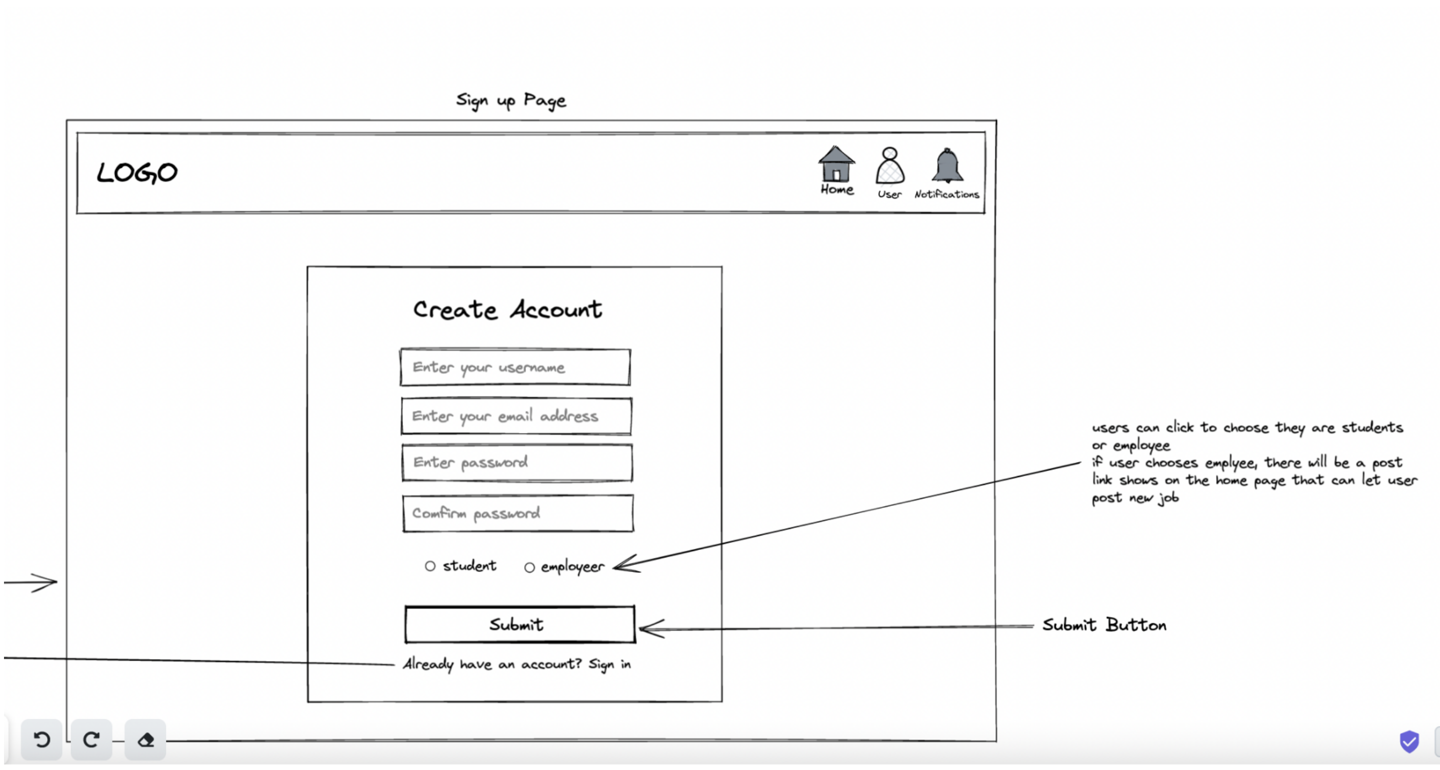
22. Users can delete their account. *(P3)*

    Delete Account means the action of deleting an Account. This removes the Account from the database entirely. A deleted Account may be recovered through a Restore Account action for a limited time after the date of deletion, until it is fully purged from all databases.

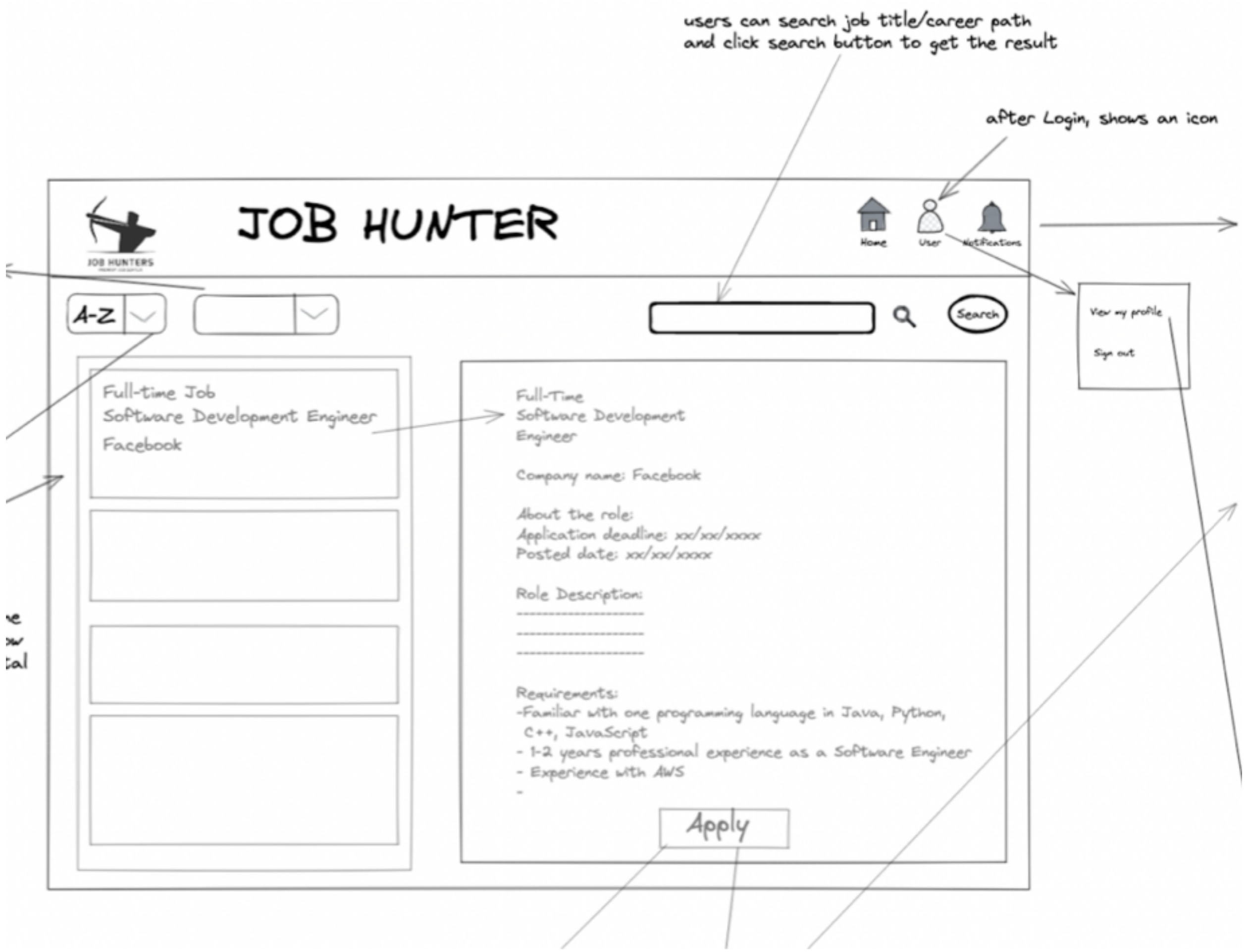
2. UI Mockups and Storyboard


pop-up window that user can click these choice to get the result filter by different job area
Artificial Intelligence and Machine Learning
Robotic Process Automation (RPA)
Edge Computing
Quantum Computing
Virtual Reality and Augmented Reality
Blockchain
Internet of Things (IoT)
5G
Cyber Security.
users can search job title/career path and click search button to get the result
after Login, shows an icon
JOB HUNTERS
JOB HUNTER
Home
User
Notifications
A-Z
Search
View my profile
Sign out
Full-time Job
Software Development Engineer
Facebook
Full-Time
Software Development Engineer
Company name: Facebook
About the role:
Application deadline: xx/xx/xxxx
Posted date: xx/xx/xxxx
Role Description:
Requirements:
-Familiar with one programming language in Java, Python, C++, JavaScript
- 1-2 years professional experience as a Software Engineer
- Experience with AWS
Apply
A B C D E F G H I J K ...
When you click the letter, it will show all job with capital A.
Sign in Page
LOGO
Login
Enter username or email
Enter password
Submit
Forgot your password?
Don't have an account? Create account
Submit Button
go to sign up page after click this link
go to sign in page after click this link
Sign up Page
Create Account
Enter your username
Enter your email address
Enter password
Confirm password
student
employee
Already have an account? Sign in
users can click to choose they are students or employee
if user chooses emplyee, there will be a post link shows on the home page that can let user post new job
Submit Button
44%
English


Sign in Page
LOGO
Home
User
Notifications
Login
Enter username or email
Enter password
Submit
Forgot your password?
Don't have an account? Create account
Submit Button
go to sign up page after click this link
go to sign in page after click this link



users can click to choose they are students or employee
if user chooses emplyee, there will be a post link shows on the home page that can let user post new job



users can search job title/career path and click search button to get the result

after Login, shows an icon

Full-time Job
Software Development Engineer
Facebook

Full-Time
Software Development
Engineer

Company name: Facebook

About the role:
Application deadline: xx/xx/xxxx
Posted date: xx/xx/xxxx

Role Description:

Requirements:
-Familiar with one programming language in Java, Python, C++, JavaScript
- 1-2 years professional experience as a Software Engineer
- Experience with AWS
-

3. High-level Architecture and Database Organization

Database Organization

Tables

- userData - stores data of basic user accounts (i.e students)
  - id - integer (unique)
  - Name - string (char)
  - Email - string (varchar) (unique)
  - Password - string (varchar)
  - userType - integer (default 0 for student, 1 for recruiters)
  - imageData - linked as foreign key for user's profile picture
  - postData - can access job postings but can only edit/post if user is a recruiter
  - interestData - string (char) stores interest information from sign up
- postData - stores data of job posting data
  - id - integer (unique)
  - Author - string (char) - linked to recruiterData
  - Title - string (char)
  - Description - string (char)
  - Category - linked as foreign key for search purposes
- categoryData - stores category data of posts to allow for precise search
  - id - integer (unique)
  - jobArea - string (varchar)

  - jobLevel - string (varchar)
- quizData - stores quiz questions and accesses postData & categoryData to recommend job postings to users
  - id - integer (unique)
  - Question - string (varchar)
- linkData - stores links to be used on Helpful Links page, accesses categoryData for search capabilities
  - id - integer (unique)
  - Name - string (varchar)
  - categoryData - linked as foreign key to be used for search capability
- imageData - stores images
  - id - integer (unique)
  - Name - string (varchar)
  - Image - integer

Images will be stored in BLOBs so as to not go over the allocated storage space for our server. Our site will not be utilizing videos or other types of media except images and pdfs, so jpeg, png, other image file types, and pdfs will be the only ones stored. These images/pdfs will then be linked to the user accounts.
For our search implementation, we are planning on using the Sequential Search algorithm, more specifically the Linear Search, since it is easier to implement with unsorted data. Search terms will be organized by Most Relevant, with the terms

most relevant to the search being the first link the user will see. Through filters, we will also be filtering the postings by categories, with job postings falling under one of nine categories and will show up when the user wants to filter their results. When accessing the DB, the search function will look at both title and desc columns to search for relevant terms to display for the user's search.

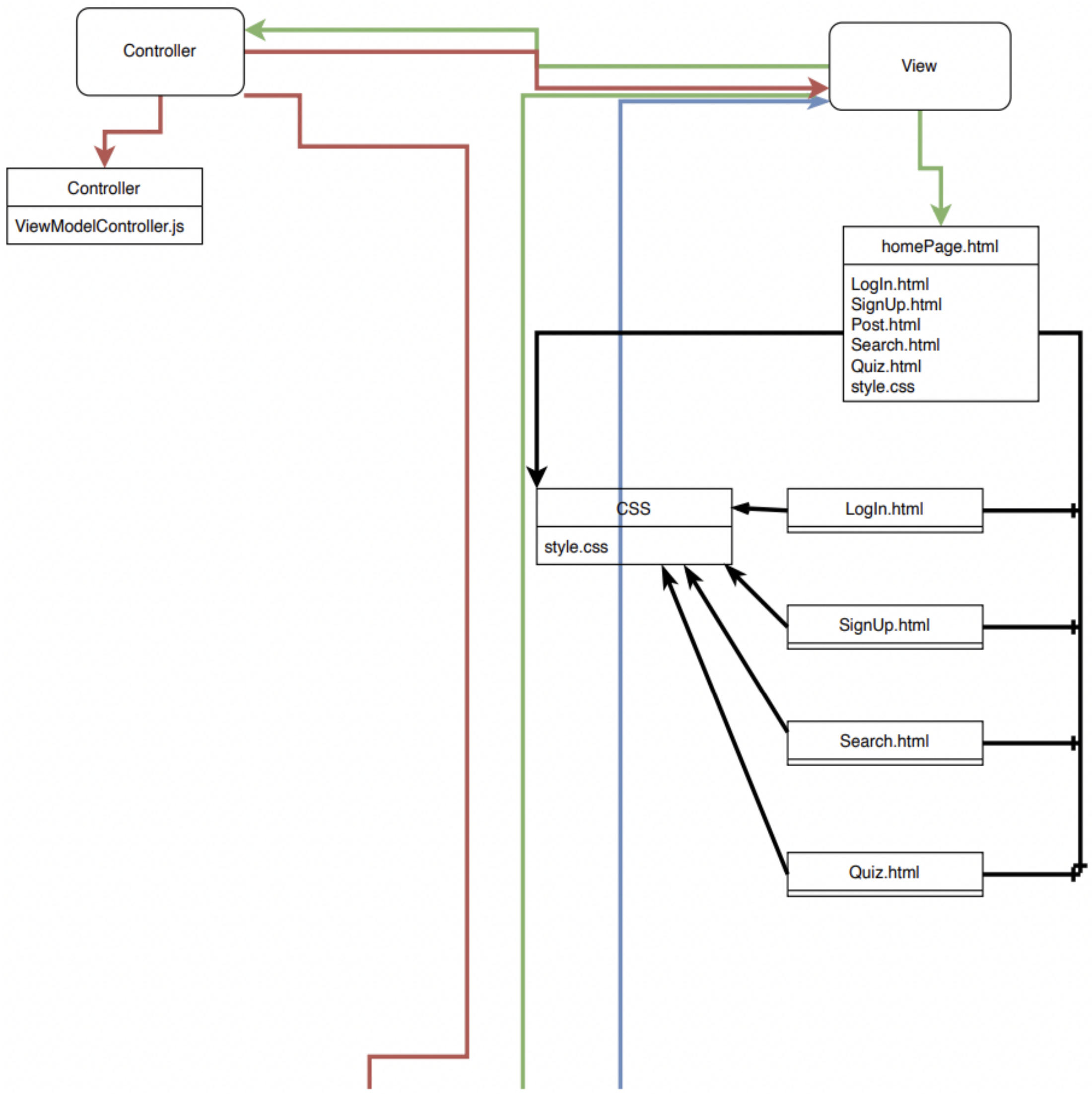
We will implement the search by utilizing the %like query in SQL. For example, if searching for AI job postings, the query will be:

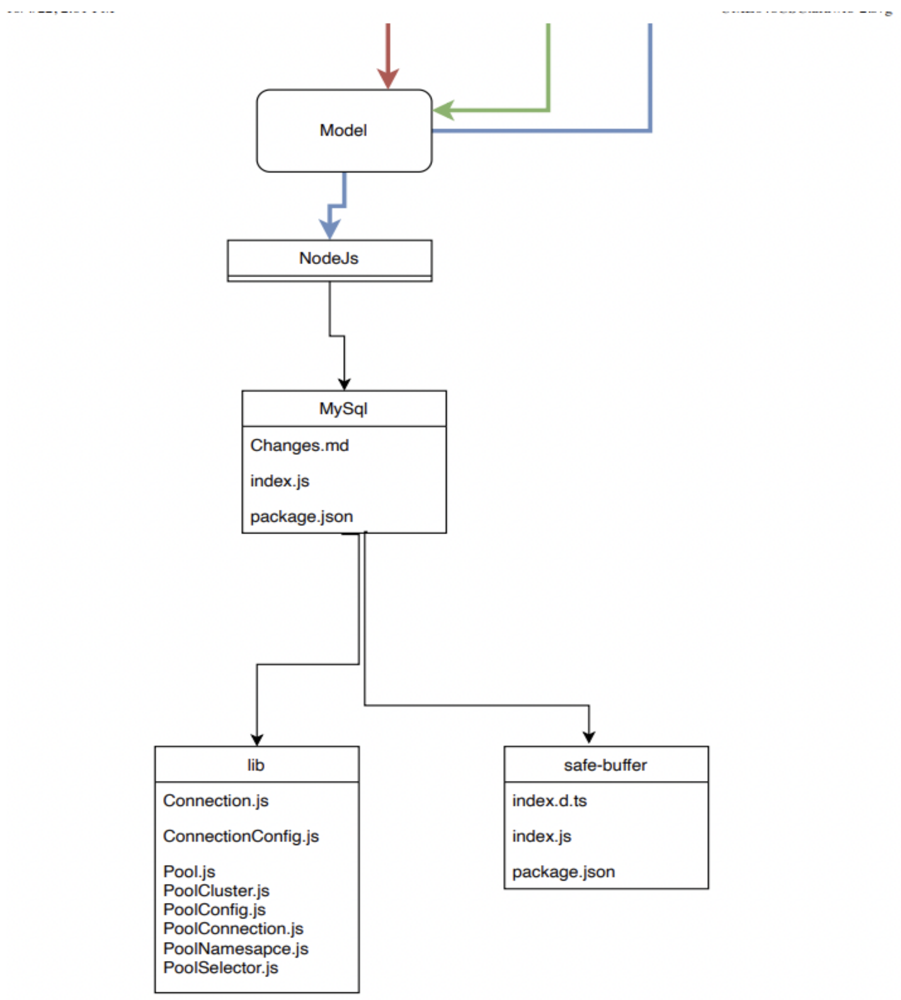
```
SELECT
        *
FROM
        jobData
WHERE
        Category LIKE '%AI%'
        OR title LIKE '%AI%'
        OR desc LIKE '%AI%';
```

The database will store all job posting information in the jobData table, and within that table will be the title, category, and description fields for the search function to look for the searched keyword. Then, the page will show these found results to the user.

For prioritization, we plan to implement the filter to sort by alphabet (ascending/descending), most relevant (default), category, and skills required (for jobs). The default display will be by Relevance to the searched keyword, but the user can change the filter to whatever suits their needs.

4. High-level UML Diagrams




Model
NodeJs
MySql
Changes.md
index.js
package.json
lib
Connection.js
ConnectionConfig.js
Pool.js
PoolCluster.js
PoolConfig.js
PoolConnection.js
PoolNamesapce.js
PoolSelector.js
safe-buffer
index.d.ts
index.js
package.json

5. Key Risks

Skill Risks

- Learning SQL database, and linking to front-end of application for data to be fetched when search function is evoked

To address this issue, the back-end will review sql usage with full stack as soon as possible in order to not hinder the team when testing the Vertical SW Prototype and for future testing.

- Coding Career Advice Robot for usage with Guidance Quiz

To address this issue, the team will research and/or review how other sites use similar features and will implement those uses into our project.

- Reviewing HTML and CSS principles to create visually-appealing web pages

To address this issue, the team will review important/key principles for web page design and implement for our application.

Schedule Risks

- Checking Discord server as much as possible in order to stay up-to-date with task management

To address this issue, team lead can have all members acknowledge delegated tasks when they are assigned so that no one is out of the loop.

- Allocating sufficient time to code web pages, update database, and test application's functionality

To address this issue, the team can set a designated period of time or day to work on the project in order to get it done and not interfere with other classwork.

- Allocating sufficient time for all team members to review completed tasks such as documentation before submitting for feedback

To address this issue, the team lead will make sure all members review the completed tasks before submitting to Class CEO/CTO.

Technical Risks

- Not having the server running when testing application

To address this issue, the team will communicate to Back-end developers to turn on the server when undergoing testing of the application.

- Database is offline when testing application

To address this issue, the team will communicate to Back-end developers to turn on the database when undergoing testing of the application.

Teamwork Risks

- Communicating when tasks are completed on-time

To address this issue, the team will confirm deadlines when tasks are assigned. If someone is finished early, they can assist those that are running behind.

- Ensuring team is aware of all features of application so there is no confusion

To address this issue, all team members will review both M1 and M2 documents to ensure they are aware of all application features and what is expected of them.

Legal/Content Risks

- Ensuring images used for job listings are not copyrighted or owned by different owners/without permission

To address this issue, we will look for copyright-free images to use so that there are no issues with lawsuits.

- Making sure server is ran when needed but not as to incur costs/payment for service

To address this issue, we will make sure that the server is turned off when testing is not in use, and only turn it on when we are testing.

6. Project Management

To manage the workload for Milestone 2 and onward, we utilize many features and principles in order to stay on task. The Team Lead confirms with the entire team the project description and what tasks are assigned, then the tasks are divided to whoever is most suitable for said task. We all also go over the deadline and set an *earlier* deadline for ourselves so that we have time to review the work and get feedback from the Class CEO. Once a team member completes their task, they inform the team and assist the other departments if needed. We plan on following

this principle moving forward, since it allows us to get the work done early, and gives us time for feedback and improvement. We also have the entire team review all code or documentation prior to submission, so that everyone is aware of what is being turned in and allows for changes when needed. In the future we'll use the Discord Task Manager Bot.