

# Job Board Management System



**Supervisor:** Trần Phước Sinh

**Class:** T1.2109.M1

**Group:** 1

<b>Serial No.</b>	<b>Student Name</b>	<b>Student ID</b>
-------------------	---------------------	-------------------

1	Nguyễn Anh Vũ	Student1339037	Leader
2	Phạm Nhật Minh	Student1328429	
3	Nguyễn Gia Thiện	Student1335976	

## TABLE OF CONTENT

ACKNOWLEDGEMENT.....	4
INTRODUCTION.....	4
<b>REVIEW 1.....</b>	<b>5</b>
1.1 PROBLEM DEFINITION.....	5
FROM THE VIEW POINT OF THE CANDIDATES.....	5
FROM THE VIEW POINT OF THE EMPLOYERS.....	5
FROM THE VIEW POINT OF THE WEBSITES'SADMINISTRATOR.....	6
1.1.1 THE REQUIREMENTS OF THE PROJECT.....	6
1.1.2 CURRENT STATE OF THE PROBLEM, AND SOLUTION.....	7
Current State of the Problem:.....	7
Proposed Solution:.....	7
1.2. CUSTOMER'S REQUIREMENTS SPECIFICATIONS (CRS).....	9
1.2.1 REQUIRE FUNCTION.....	9
1.2.2 NON-FUNCTIONAL REQUIREMENTS.....	10
1.2.3 DATA PROCESSING PROCESSES.....	11
1.2.4 ALGORITHMS.....	12
• Search.....	12
• Sort.....	13
• Machine learning.....	14
1.2.5 HARDWARE/ SOFTWARE REQUIREMENTS.....	14
HARDWARE.....	14
SOFTWARE.....	15
TECHNOLOGY.....	15
1.2.6 LIMITS OF THE PROJECT.....	16
1.3 FUNCTION DIAGRAM.....	18
1.4 TASK SHEET REVIEW 1.....	19
<b>REVIEW 2.....</b>	<b>20</b>
2.1 ARCHITECTURE AND DESIGN OF THE EPROJECT.....	20
2.2 USECASE AND SEQUENCE DIAGRAM.....	22
2.2.1 USECASE GENERAL.....	22
2.2.2 USECASE DETAILS.....	24
Use Case List.....	25
Employer.....	27
Controller.....	36
UnRegister Candidate.....	46
2.2.3 SEQUENCE DIAGRAM.....	56
2.3 ENTITY RELATIONSHIP DIAGRAM (ERD).....	75
2.3.1 ERD General.....	75

2.3.2 ERD Details.....	76
2.4 CLASS DIAGRAM.....	80
2.5 TASK SHEET REVIEW 2.....	82
<b>REVIEW 3.....</b>	<b>84</b>
3.1 INSTALL DATABASE PHYSICAL.....	84
Create database and table.....	84
job.....	84
reasonstojoin.....	84
jobsdescriptions.....	85
skillsexperiences.....	85
whyyouloveworkinghere.....	85
employer.....	85
skill.....	86
jobskill.....	86
jobmonitor.....	86
review.....	89
candidate.....	89
answer.....	90
question.....	90
test.....	90
resumetemplate.....	90
resumepdf.....	91
Create relationship between tables.....	91
DATABASE TABLE & RELATIONSHIP DESIGN.....	95
3.2 SITE MAP.....	96
3.3 CONSTRUCTOR PROJECT DIRECTORY.....	97
- Back end.....	97
- Frond end.....	100
3.4 GUI DESIGN.....	102
3.4.1 DESIGN INTERFACE (FIGMA).....	102
WEB.....	102
For Candidate.....	102
Search job.....	102
View job details.....	104
Comments.....	104
Create CV.....	105
predict salary.....	106
Sign in.....	107
sign up.....	108
Test.....	109

For Employer.....	110
Home.....	110
Post new job.....	110
update posted job.....	111
List posted job.....	111
create account.....	112
For Controller.....	113
Home.....	113
Manage controller manage employer, manage candidate.....	113
Monitor behavior.....	114
review & approve job.....	116
MOBILE.....	117
menubar.....	117
Home.....	118
All job lists.....	119
login.....	120
sign up.....	121
profile.....	122
employer.....	123
login.....	124
sign up new employer.....	125
job details.....	126
job apply.....	127
All company lists.....	128
Company details.....	129
company review.....	130
upload/create CV.....	131
create new CV.....	132
favorite list.....	133
take the test.....	134
List Job Posted.....	135
Review CV.....	136
CV details.....	137
Download CV.....	139
Update Employer Account.....	141
3.4.2 DETAILED DESCRIPTION OF THE FUNCTION.....	145
AND INPUT/OUTPUT DATA ANALYSIS.....	145
Web.....	145
For candidate.....	145
Show jobs.....	145

View job details.....	147
Comments.....	148
Create CV.....	150
predict salary.....	151
Sign in.....	152
sign up.....	153
Test.....	155
For Employer.....	156
Post new job.....	156
update posted job.....	157
For Controller.....	158
Manage controller manage employer, manage candidate.....	158
Monitor behavior.....	160
review & approve job.....	160
MOBILE.....	161
1. Home Page.....	161
2. Login.....	162
Login Candidate.....	163
Login Employer.....	165
3. Sign Up.....	166
Sign Up Candidate.....	167
Sign Up Employer.....	169
4. Profile.....	171
6. Jobs Favorite List.....	174
7. Take The Test.....	176
9. Job Details.....	178
10. Job Apply.....	179
11. Company List.....	180
12. Company Details.....	181
3.4.4 DESCRIPTION OF DATA PROCESS, APPLIED ALGORITHM (IF ANY).....	184
3.5 TASK SHEET REVIEW 3.....	187

## ACKNOWLEDGEMENT

Our group would like to devote this chapter to express our sincere appreciation to all those who offered us valuable assistance during in carrying out this Project.

Firstly, we own a debt of gratitude to our dedicated instructor, Mr. Trần Phuốc Sinh<sup>1</sup>, whose suggestions and encouragement helped us to overcome not only technological challenges but also psychological tension. Without him, our work could not complete successfully.

Secondly, we highly appreciate the Project Team at the Head Office, which has provided us an excellent opportunity to apply what we learned into practice. We are certain that the experiences we gained from this project will facilitate our future career greatly.

Thirdly, we would also like to express my gratitude to all the staff members of my center for not only providing me with the opportunity to work with them on this project, but also for their support and encouragement throughout the process.

Finally yet importantly, we wish to say a personal thank you to all our dear classmates and team members who are always willing to support us openly and enthusiastically.

## INTRODUCTION

In today's digital era, information technology has revolutionized the way we conduct business, including the process of recruitment. With the latest technology, managing a job board system has become more efficient than ever. As a result, we have developed a comprehensive application called "Job Board Management System."

The Job Board Management System caters to organizations and companies aiming to establish a robust online job board platform, specifically tailored for candidates seeking careers in the IT field. This platform serves as a comprehensive resource where IT job seekers can efficiently search for suitable employment opportunities. Simultaneously, it offers employers a centralized space to post job listings and identify potential IT candidates who align with their specific requirements.

By functioning as a bridge between candidates and employers, the Job Board Management System facilitates effective communication and collaboration. It empowers both parties to identify shared interests, fostering mutually beneficial connections within the IT job market.

---

<sup>1</sup>

# REVIEW 1

## 1.1 PROBLEM DEFINITION

The current state of the job market, particularly in the IT field, presents several challenges for both job seekers and employers. These challenges include inefficient job search processes, difficulty in finding suitable candidates, and a lack of streamlined communication between the two parties. As a result, there is a need for a comprehensive Job Board Management System to address these issues and enhance the overall recruitment experience.

the Job Board Management System aims to provide a user-friendly and comprehensive platform where job seekers can easily search and apply for IT job opportunities. Simultaneously, employers can post job listings and efficiently identify and connect with qualified candidates. By streamlining the recruitment process and fostering effective communication, the system aims to enhance the overall job search experience and bridge the gap between job seekers and employers in the IT field.

After analyzing the project specifications of car show room, to fulfill customer's requirements properly, we came to conclusion that in order to achieve the best result, we should deal with following queries:

### FROM THE VIEW POINT OF THE CANDIDATES

Job seekers in the IT field face challenges when searching for suitable career opportunities. The existing job search methods, such as browsing multiple platforms and websites, often lead to a fragmented and time-consuming experience. Candidates may struggle to find relevant job listings and encounter difficulty in effectively showcasing their skills and qualifications to potential employers. A comprehensive Job Board Management System would address these issues by providing a user-friendly platform that allows candidates to easily search for IT jobs, upload their resumes, and highlight their expertise. This system would streamline the job search process, enhance visibility for candidates, and improve their overall experience in finding their desired IT careers.

### FROM THE VIEW POINT OF THE EMPLOYERS

Employers seeking IT talent often face challenges in attracting qualified candidates for their job openings. Traditional recruitment methods may not effectively reach a wide pool of candidates, resulting in limited choices and extended hiring cycles. Employers may struggle to find candidates with the specific skills and qualifications required for their IT positions. A Job Board Management System would be beneficial from the employer's viewpoint as it would provide a centralized platform to post job listings, effectively target the desired talent pool, and streamline the hiring process. The system would allow employers to easily access a diverse range of qualified candidates, review their profiles and resumes, and efficiently communicate with potential hires, thereby simplifying the process of identifying and recruiting top IT talent.

## FROM THE VIEW POINT OF THE WEBSITES'ADMINISTRATOR

As the administrator of the Job Board Management System, there are several challenges and considerations to address. The system must be designed to handle a high volume of job listings, candidate profiles, and employer accounts efficiently. The website should provide a user-friendly interface that allows easy navigation for both candidates and employers. It is crucial to ensure the security and privacy of user data, implementing measures to protect personal information and prevent unauthorized access. The system should also incorporate robust communication features to facilitate seamless interactions between candidates and employers. Additionally, the administrator needs to monitor and maintain the website's functionality, perform regular updates, and address any technical issues or user feedback promptly. By managing these aspects effectively, the administrator can ensure a smooth and reliable operation of the Job Board Management System, providing a valuable resource for both candidates and employers in the IT industry.

### 1.1.1 THE REQUIREMENTS OF THE PROJECT

- **User Registration and Profiles:**
  - Allow job seekers and employers to register and create user profiles.
  - Collect and store relevant information, such as contact details, job preferences, and qualifications.
  - Provide options to update and manage user profiles.
- **Job Posting and Management:**
  - Enable employers to create and manage job postings with detailed information, including job title, description, requirements, and application instructions.
  - Support features like job categories, location, salary range, and duration.
  - Allow employers to edit, deactivate, or delete job postings as needed.
- **Job Search and Filtering:**
  - Provide a robust search functionality for job seekers to find relevant job opportunities.
  - Implement filters based on criteria such as job title, location, skills, experience level, and industry.
  - Offer advanced search options like Boolean search, saved searches, and sorting capabilities.
- **Application Submission and Tracking:**
  - Allow job seekers to submit applications through the platform.
  - Enable employers to review, manage, and track received applications.
  - Provide notifications and status updates to both candidates and employers.
- **Candidate Profile Management:**
  - Allow job seekers to create and manage their profiles, including personal information, work experience, education, skills, and certifications.
  - Provide options to upload resumes, cover letters, and other relevant documents.

- **Analytics and Reporting:**

- Generate analytics and reports on user activity, job postings, application rates, and other relevant metrics.
- Provide insights into job market trends, skill demands, and candidate demographics.
- Enable administrators to monitor and analyze the platform's performance.

- **Security and Data Privacy:**

- Implement robust security measures to protect user data, including encryption, secure authentication, and data backup protocols.
- Ensure compliance with data protection regulations, such as GDPR or CCPA.
- Allow users to manage their data privacy settings and provide consent for data processing.

- **Responsive Design and User Experience:**

- Develop a responsive and user-friendly interface accessible across different devices (desktop, mobile).
- Implement intuitive navigation, clear layouts, and a visually appealing design.
- Conduct usability testing and gather user feedback to continuously improve the user experience.

### 1.1.2 CURRENT STATE OF THE PROBLEM, AND SOLUTION

#### Current State of the Problem:

In the current job market, particularly in the IT field, job seekers and employers face several challenges. Job seekers struggle with inefficient job search processes, difficulty in finding suitable positions, and a lack of streamlined communication with employers. On the other hand, employers face challenges in attracting qualified candidates and efficiently identifying the right fit for their job openings. These issues lead to a mismatch between job seekers and employers, resulting in extended hiring cycles, missed opportunities, and frustration for both parties.

#### Proposed Solution:

The Job Board Management System aims to address these challenges and provide a comprehensive solution for job seekers and employers in the IT field. By developing an online platform, the system connects candidates and employers, facilitating a streamlined and efficient recruitment process.

- Enhanced Job Search: The system offers an advanced job search functionality, allowing candidates to search for IT job opportunities based on their preferences, such as location, job title, skills, and experience. This improves the efficiency of job searching, ensuring that candidates can find relevant positions that match their qualifications and interests.
- Comprehensive Job Postings: Employers can post detailed job listings, including job descriptions, required qualifications, and application instructions. This ensures that job postings provide

comprehensive information, helping candidates understand the job requirements and make informed decisions when applying.

- Candidate Profile Management: Job seekers can create detailed profiles showcasing their skills, experience, education, and certifications. This allows employers to assess candidate suitability based on their profiles and easily access relevant information during the hiring process.
- Communication and Application Tracking: The system facilitates seamless communication between candidates and employers. It allows candidates to submit applications directly through the platform, and employers can track and manage applications efficiently. This ensures timely communication and feedback, reducing delays in the hiring process.
- Customized Recommendations: The system incorporates a recommendation engine that suggests relevant job opportunities to candidates based on their profiles, preferences, and job search history. This personalization improves the job matching process, enhancing the chances of finding suitable positions.
- Analytics and Insights: The system provides analytics and insights to both candidates and employers. Candidates can gain insights into job market trends, skill requirements, and salary ranges, helping them make informed decisions about their career paths. Employers can access data on candidate demographics, application rates, and other metrics to refine their hiring strategies.
- Monetization Options: The system offers monetization options for employers, such as subscription plans, pay-per-post models, and premium features. This allows the platform to generate revenue while providing value-added services to employers.

By implementing the Job Board Management System, job seekers benefit from an improved job search experience, increased visibility, and enhanced communication with employers. Employers gain access to a larger pool of qualified candidates, streamlined recruitment processes, and valuable insights into the job market. Overall, the solution aims to bridge the gap between job seekers and employers, making the recruitment process more efficient and effective for both parties.

## 1.2. CUSTOMER'S REQUIREMENTS SPECIFICATIONS (CRS)

The project is to design a car Job board Management System.

### 1.2.1 REQUIRE FUNCTION

- There are 3 types of actors in the application as follow:
  - Candidate** who are looking for a job:
    - If he or she login, they will be “candidate”.
    - If he or she just visit the website and not login, they will be “visitor”
  - Employer** who are posting jobs on the website and are seeking for potential candidates
  - Controller** who are controlling the website

No.	Index	Required function
1	REQ 1:	<b>Candidate</b>
2	REQ 1.1:	<b>For Unregistered candidate</b>
3	REQ 1.1.1	View all list job
4	REQ 1.1.2	View job detail, job information like company's name, location, skill required, job description, ...
5	REQ 1.1.3	Search jobs, find jobs by location, skill, salary range.
6	REQ 1.1.4	Sign up.
7	REQ 1.2	<b>For Registered candidate</b>
8	REQ 1.2.1	Sign in.
9	REQ 1.2.2	View job detail, job information like company's name, location, skill required, job description, ...
10	REQ 1.2.3	Search jobs, find jobs by location, skill, salary range.
11	REQ 1.2.4	Take some tests to analyze your personal characteristics and abilities to determine whether you are suitable for the job's working environment.
12	REQ 1.2.5	Comment/view review on company's working environment
13	REQ 1.2.6	Predict salary based on Machine Learning big data
14	REQ 1.2.7	Create new CV by filling the CV templates of website
15	REQ 1.2.8	Upload CV with format PDF on website
16	REQ 1.2.9	Submit CV.
17	REQ 1.2.10	Add job to favorite, add considerable job to the favorite list so you don't have to search for it again .
18	REQ 2	<b>Employer</b>
19	REQ 2.1	Sign up (create new account)
20	REQ 2.2	Sign in
21	REQ 2.3	Update employer account
22	REQ 2.4	Post Job profiles (Allow registered employers in the company to post jobs on the website)

23	REQ 2.5	Update job profiles (Allow employer can update their posting job)
24	REQ 2.6	Review CV (Allow registered employers in the company read the CV that candidates apply)
25	REQ 2.7	Download CV of candidate applied
26	REQ 2.8	Create Account inside the system (Allow registered employers in the company to create an account for people in the company to manage)
27	<b>REQ 3</b>	<b>Controller</b>
28	REQ 3.1	Review & Approve post job (Allow admin/staff in the system to read all jobs posted by Employer)
29	REQ 3.2	Review CV (Allow admin/staff in the system to read all CV that applied by candidates)
30	REQ 3.3	Create account (Allow admin to create an account for another staff in the system)
31	REQ 3.4	Admin has CRUD function for admin/staff (Allow admin to create-read-update-disable an admin/staff in the system)
32	REQ 3.5	Monitor candidates/employers behavior (Allow admin/staff can observe the behavior of candidates/employers behavior when they use the website )
33	REQ 3.6	Review report job trending, and sales (Allow admin/staff to see how popular each job and the sale)
34	REQ 3.7	Manage (view, search, enable/disable) candidate or employer
35	REQ 3.8	Manage job posting (view, search, enable/disable)
36	REQ 3.9	Manage comments (view, add, disable/enable, edit)

## 1.2.2 NON-FUNCTIONAL REQUIREMENTS

- There are several non-functional requirements that should be fulfilled by the system. The system should be:
  - Safe to use:** The system should not result in any malicious downloads or unnecessary file downloads.
  - Accessible:** The system should have clear and legible fonts, user-interface elements, and navigation elements.
  - User-friendly:** The system should be easy to navigate with clear menus and other elements and easy to understand.
  - Operability:** The system should operate in a reliably efficient manner.
  - Performance:** The system should demonstrate high value of performance through speed and throughput. In simple terms, the system should be fast to load and page redirection should be smooth.
  - Security:** The system should implement adequate security measures such as authentication. For example, only registered users can access certain features.
  - Capacity:** The system should support large number of users.
  - Availability:** The system should be available 24/7 with minimum downtime.

- **Compatibility:** The system should be compatible with latest browsers.

### 1.2.3 DATA PROCESSING PROCESSES

- The data in this project will be stored in the SQL server database.
- **User Registration and Profile Creation:**
  - When candidates and employers register on the job board platform, their personal information such as name, contact details, and professional background needs to be collected. This data is processed to create user profiles that can be used for job applications, job postings, and communication purposes.
- **Job Posting:**
  - Employers can post job listings on the platform, providing details such as job title, description, required qualifications, and application instructions. The system processes this information to display the job postings correctly and categorize them based on relevant criteria, such as job type, location, and industry.
- **Job Search and Filtering:**
  - Candidates utilize the job board system to search for suitable job opportunities based on their preferences and qualifications. The system processes search queries and applies filters such as location, job title, and skills to retrieve relevant job listings from the database. This involves matching the criteria specified by the candidate with the information associated with each job posting.
- **Application Submission:**
  - Candidates can apply for jobs through the platform by submitting their resumes and cover letters. The system processes these application materials and stores them securely, associating them with the respective job posting and candidate profile. This allows employers to review and evaluate applications efficiently.
- **Communication and Notifications:**
  - The system facilitates communication between candidates and employers. This includes sending notifications to candidates regarding application status updates, interview invitations, and other relevant information. Employers can also use the platform to communicate with candidates, schedule interviews, and provide feedback. The system processes these communication activities and ensures the timely and accurate delivery of messages.
- **Analytics and Reporting:**
  - The Job Board Management System may generate analytics and reports to provide insights into the platform's performance, job market trends, and user activity. This involves aggregating and analyzing data related to job postings, candidate profiles, application rates, and other relevant metrics. These insights can assist in making data-driven decisions and optimizing the platform's functionality.

- **Data Privacy and Security:**

- Data processing in the job board project must adhere to strict privacy and security measures. User data, including personal information, resumes, and communication records, needs to be protected from unauthorized access and handled in compliance with data protection regulations. Encryption, access controls, and regular backups are among the security measures implemented to safeguard the data.

- **Resume Parsing and Data Extraction:**

- When candidates upload their resumes, the system can employ resume parsing techniques to extract relevant information automatically. This includes extracting details like work experience, education, skills, and contact information from the resume documents. The extracted data is processed and stored in a structured format, facilitating efficient search and matching algorithms.

- **Data Analytics and Insights:**

- The system may leverage data analytics techniques to extract meaningful insights from the accumulated data. This can include analyzing trends in job market demand, salary ranges, skill requirements, and candidate preferences. These insights can assist employers in making informed decisions regarding their hiring strategies and enable continuous improvement of the job board platform.

- **Predict salary by using machine learning on big data:**

- In the context of job board systems, the utilization of machine learning on big data to predict salary offers the potential for enhanced salary insights and decision-making. By leveraging large-scale data sets comprising job postings, candidate profiles, and historical salary information, machine learning algorithms can analyze patterns and factors influencing salary levels. This approach allows for the development of predictive models that estimate salaries based on job attributes, candidate qualifications, and market trends. Employers can benefit from these predictions to optimize their salary offerings and attract top talent, while candidates gain valuable insights into salary expectations for specific roles. Overall, applying machine learning to big data enables more informed salary negotiations and fosters a more efficient job market.

## 1.2.4 ALGORITHMS

- **Search**

- **Linear Search:**

- This is a basic search algorithm that sequentially checks each element in a collection until a match is found or the end of the collection is reached. Linear search has a time complexity of  $O(n)$ , where  $n$  is the number of elements in the collection.

- **Binary Search:**

- Binary search is a more efficient search algorithm used on sorted collections. It works by repeatedly dividing the search space in half until the desired element is found. Binary search has a time complexity of  $O(\log n)$  and is particularly useful for large sorted datasets.

- **Hashing:**

- Hashing is a search technique that uses a hash function to map the search key to an index in an array (hash table). By accessing the element at the computed index, the desired item can be quickly retrieved. Hashing provides constant-time search complexity on average, but it may degrade to  $O(n)$  in the worst case due to collisions.

- **Tree-based Search:**

- Tree-based search algorithms, such as Binary Search Trees (BST) or Balanced Binary Search Trees (e.g., AVL, Red-Black Trees), organize data in a tree-like structure. These algorithms allow for efficient searching by recursively traversing the tree based on comparisons between the search key and the nodes.

- **Graph-based Search:**

- Graph-based search algorithms, such as Depth-First Search (DFS) and Breadth-First Search (BFS), are used to traverse and search elements in graph structures. These algorithms are often employed for path-finding, graph traversal, or graph-based searching problems.

- **Sort**

- **Insertion sort**

- Insertion Sort is another elementary sorting algorithm. It works by iteratively building the final sorted array one element at a time. Initially, the first element is considered sorted, and each subsequent element is compared with the already sorted portion, finding its correct position and inserting it there. This process is repeated until the entire array is sorted. Insertion Sort has a time complexity of  $O(n^2)$ , but it performs better than Bubble Sort on partially sorted or small datasets.

- **Buble sort**

- Bubble Sort is a simple and commonly used sorting algorithm. It works by repeatedly swapping adjacent elements if they are in the wrong order until the entire list is sorted.

The algorithm iterates through the list multiple times, comparing adjacent pairs and swapping them if necessary. The largest (or smallest) element gradually "bubbles" up to its correct position with each iteration. Bubble Sort has a time complexity of  $O(n^2)$ , making it less efficient for large datasets.

- Machine learning

- Linear regression

- Linear regression is a statistical modeling technique used to establish a relationship between a dependent variable and one or more independent variables. It aims to fit a linear equation to the data, enabling prediction and understanding of the relationship between variables.

- Logistic regression

- Logistic regression is a predictive modeling algorithm used for binary classification tasks. It estimates the probability of an event occurring based on input variables. It is widely used in fields such as healthcare, finance, and social sciences to predict outcomes and make informed decisions.

- Decision tree and random forest

- Decision trees are hierarchical structures used for classification and regression tasks. They split data based on features to create a tree-like model for decision-making. Random Forest is an ensemble learning method that combines multiple decision trees to improve accuracy and mitigate overfitting, making it suitable for complex problems.

- K Nearest Neighbour (KNN) & Support Vector Machine (SVM)

- KNN is a non-parametric classification algorithm that assigns a data point to the majority class of its nearest neighbors. SVM is a supervised learning algorithm that separates data points using a hyperplane, aiming to maximize the margin between classes. Both methods are widely used for classification tasks in various domains.

## 1.2.5 HARDWARE/ SOFTWARE REQUIREMENTS

### HARDWARE

#### Web Server

<b>Processor</b>	Intel Core I7 or higher.
<b>Memory</b>	32 GB RAM or greater.

<b>Modem/ADSL</b>	Internet access is required.
-------------------	------------------------------

**Client**

<b>Processor</b>	Intel Core I5 or higher.
<b>Memory</b>	8 GB RAM or greater.
<b>Monitor</b>	Super VGA (1024x768) or higher resolution.
<b>Modem/ADSL</b>	Internet access is required.

**SOFTWARE**

## Web Server

<b>Operation System</b>	Window 7 or later.
<b>Browser</b>	Google Chrome version 35.
<b>Database</b>	SQL SERVER
<b>Software</b>	<p>Internet Information Service</p> <ul style="list-style-type: none"> <li>• NETBEANS</li> <li>• VISUAL STUDIO 2022</li> <li>• VISUAL STUDIO CODE</li> <li>• POWER BI DESKTOP &amp; SERVICES</li> <li>• MICROSOFT SQL SERVER MANAGEMENT STUDIO</li> <li>• GOOGLE CHROME</li> <li>• FIGMA</li> <li>• ANDROID STUDIO</li> <li>• ChatGPT</li> </ul>

**TECHNOLOGY**

<b>Language</b>	<b>Framework</b>
<ul style="list-style-type: none"> <li>• Javascript.</li> <li>• Typescript</li> </ul>	Reactjs

• Jsx	
• HTML	
• CSS/SASS	
• Dax	Power BI Desktop & Services
• Dart	Flutter
• C#	ASP.NET Core Web API / DevExpress
• Java	Java Spring Boot Rest API
• Python	Flask
• javascript in Node JS (NodeJs is a runtime environment only)	Express JS

### 1.2.6 LIMITS OF THE PROJECT

- **Limited Scope:** The job board project focuses primarily on facilitating the job search and recruitment process within the IT field. It may not cater to other industries or professions, which can limit its applicability to a broader range of job seekers and employers.
- **Dependency on User Engagement:** The success of a job board platform relies heavily on user engagement and adoption. If there is insufficient participation from candidates and employers, the effectiveness and value of the platform may be diminished.
- **Quality of Job Postings:** The job board system relies on employers to accurately and comprehensively provide job descriptions and requirements. However, there is a possibility of incomplete or vague postings that may result in misalignment between candidate expectations and actual job responsibilities.
- **Competition and Market Saturation:** The job board industry is highly competitive, with several established platforms already in existence. Launching a new job board project may face challenges in gaining traction and attracting a significant user base, especially in saturated markets.
- **Passive Job Seekers:** Not all job seekers actively utilize job board platforms, as some prefer alternative methods of job searching or rely on personal networks for employment opportunities. This can limit the reach and effectiveness of the platform in connecting with potential candidates.
- **Lack of Guarantee for Job Placement:** While the job board platform facilitates the job search process and connects candidates with employers, it does not guarantee job placement. The ultimate

hiring decision rests with the employers, and the platform's role is to facilitate the introduction and initial screening process.

- **Evolving Technological Landscape:** The field of information technology is constantly evolving, with new technologies, programming languages, and skills emerging over time. The job board project may face challenges in keeping pace with these changes and ensuring that the platform remains relevant and up to date.

## 1.3 FUNCTION DIAGRAM



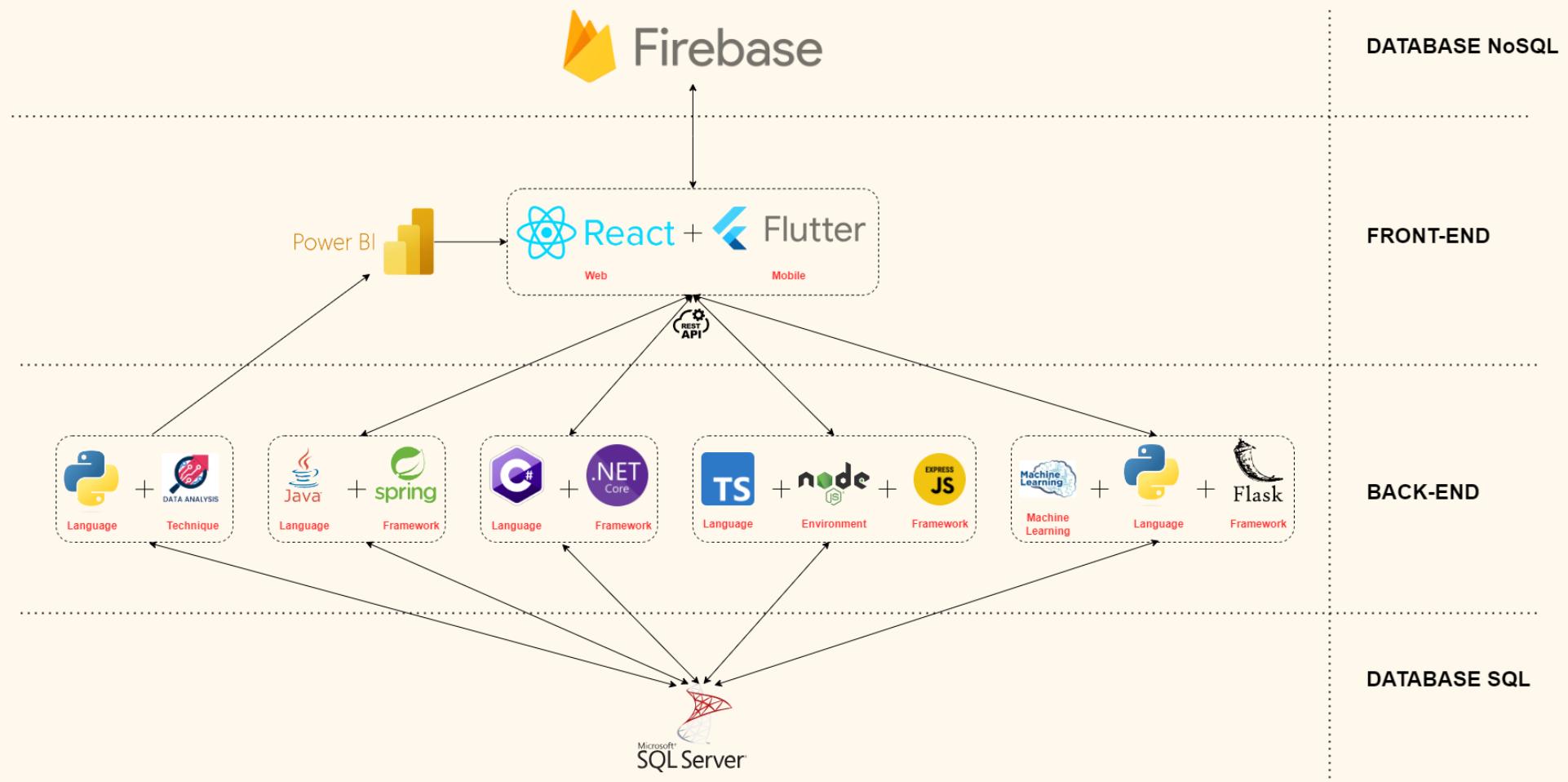
## 1.4 TASK SHEET REVIEW 1

Project Ref.No		Project Title	Activity plan prepared by	Date of preparation of Activity plan				
#	Task			Start date	End date	Actual days	Member's name	Status
1	Acknowledgement	Job Board Management System	Nguyễn Anh Vũ	2023-06-28	2023-06-30	2	Nguyễn Anh Vũ	Done
2	Introduction			2023-06-28	2023-06-30	2	Nguyễn Anh Vũ	Done
3	Problem definition			2023-06-28	2023-06-30	2	Nguyễn Anh Vũ	Done
4	The requirements of the project			2023-06-28	2023-06-30	2	Nguyễn Anh Vũ	Done
5	Current state of the problem, and solution			2023-06-28	2023-06-30	2	Nguyễn Gia Thiện	Done
6	Customer requirement specification (CSR)			2023-06-28	2023-06-30	2	Phạm Nhật Minh	Done
7	Required function for Candidate			2023-06-28	2023-06-30	2	Phạm Nhật Minh	Done
8	Required function for Controller			2023-06-28	2023-06-30	2	Nguyễn Gia Thiện	Done
9	Required function for Employer			2023-06-28	2023-06-30	2	Nguyễn Gia Thiện	Done
10	Non-functional requirement			2023-06-28	2023-06-30	2	Nguyễn Anh Vũ	Done
11	Data processing processes			2023-06-28	2023-06-30	2	Phạm Nhật Minh	Done
12	Algorithms			2023-06-28	2023-06-30	2	Nguyễn Anh Vũ	Done
13	Hardware/software requirement			2023-06-28	2023-06-30	2	Nguyễn Gia Thiện	Done
14	Limits of the project			2023-06-28	2023-06-30	2	Phạm Nhật Minh	Done
15	Function diagram			2023-06-28	2023-06-30	2	Nguyễn Anh Vũ	Done
16	Task Sheet review 1			2023-06-28	2023-06-30	2	Nguyễn Anh Vũ	Done

Signature of instructor	Signature of Team Leader
Mr. Trần Phước Sinh	Mr. Nguyễn Anh Vũ

## REVIEW 2

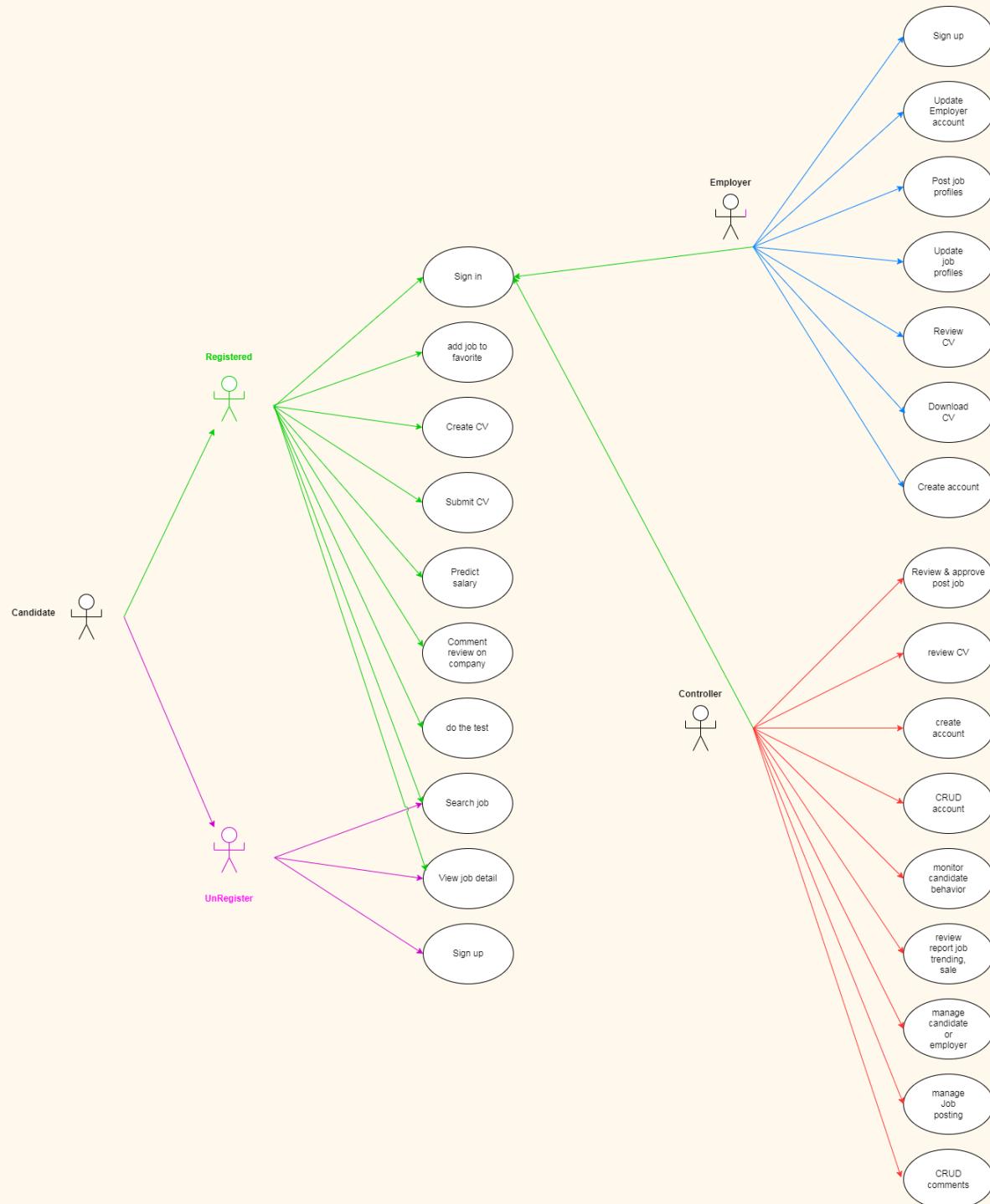
### 2.1 ARCHITECTURE AND DESIGN OF THE EPROJECT



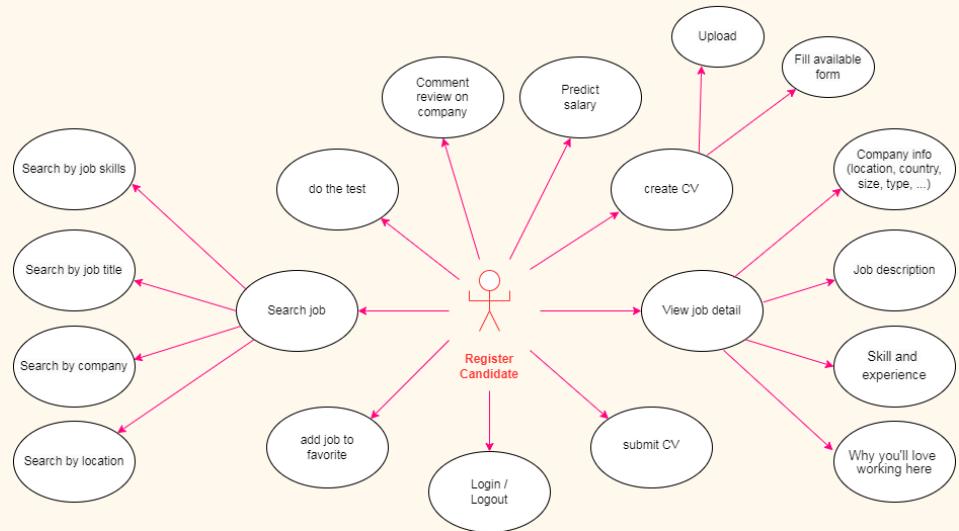
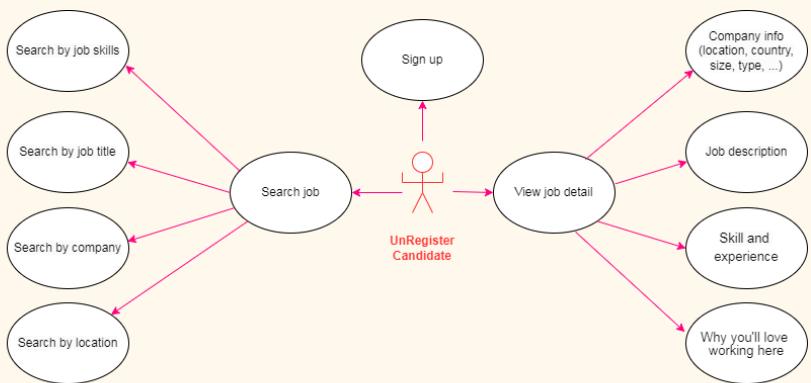


## 2.2 USECASE AND SEQUENCE DIAGRAM

### 2.2.1 USECASE GENERAL



## 2.2.2 USECASE DETAILS



## Use Case List

#	Code	Name	Description
<b>Employer</b>			
1	UC101	Sign Up	Sign up a new employer account
2	UC102	Update Employer Account	Allow Employer in the company to update Employer-Account information
3	UC103	Post Job Profile	Allow Employers to post job profiles on the page
4	UC104	Update Job Profile	Allow Employer to self-update information
5	UC105	Review CV	Allow the Employer to read all the CV that the candidate uploaded
6	UC106	Download CV	Allow the Employer to download the CV that the candidate uploaded
7	UC107	Create Account	Allow Employer to create Employer account for another employee in the company
<b>Controller</b>			
8	UC201	Review & Approve Post Job	Allow Admin/Staff to review and approve jobs that posted by Employer
9	UC202	Review CV	Allow Admin/Staff to read all the CV of all candidates uploaded

10	UC203	Create Account	Allow Admin/Staff to create an account type "admin/staff"
11	UC204	CRUD Account	Allow Admin/Staff to have permission to Create-Read-Update-Delete Admin/Staff account
12	UC205	Monitor Candidate Behavior	Allow Admin/Staff to view how many views that candidate clicks/hovers in a specific job
13	UC206	Review Report Job trending, sale	Allow Admin/Staff to know which job was trending and the salary
14	UC207	CRUD Candidate & Employer	Allow Admin/Staff to have permission to Create-Read-Update-Delete Candidate/Employer account
15	UC208	CRUD Job posting	Allow Admin/Staff to have permission to Create-Read-Update-Delete Employer posted job
16	UC209	CRUD Comments	Allow Admin/Staff to have permission to Create-Read-Update-Delete Employer posted comments on the web

UnRegister Candidate			
17	UC210	View all jobs list	
18	UC211	View Job details	View job information like company's name, location, skill Required, job description, ... (except salary)
19	UC212	Search jobs	find jobs by location, skill, company
20	UC213	Sign up	

Register Candidate			
21	UC214	Sign In	
22	UC215	View Job details	View job information like company's name, location, skill Required, job description, ... (include salary)
23	UC216	Search jobs	find jobs by location, skill, company, salary range
24	UC217	Take some Tests	To analyse your personal characteristics and abilities to determine whether you are suitable for the job's working environment.
25	UC218	Comment/view review on company	
26	UC219	Predict salary	Predict salary based on Machine Learning big data
27	UC220	Create new CV	Fill CV templates of website
28	UC221	Upload new CV	Upload CV with format PDF on website
29	UC222	Submit CV	
30	UC223	Add job to favourite	Add considerable job to the favourite list so you don't have to search for it again

## Employer

### Sign Up

ID and Name:	UC101 – Sign Up
--------------	-----------------

Primary Actor:	Unauthorized user	Secondary Actor:	n/a
Description:	Anonymous users sign up for the system		
Pre-condition:	Anonymous user		
Post-condition:	<b>Success:</b> claim access token and will be redirected to the login page  <b>Fail:</b> the system shows an error message		
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>1. Anonymous users try to access a restricted resource</li><li>2. He/she will be redirected to sign up Page</li><li>3. User fills in the Sign-Up form</li><li>4. User clicks on the “Create Account” button</li><li>5. Auth0 authorization server validates given credentials and response access token</li><li>6. The browser immediately redirects the user to Login Page</li><li>7. Home Page shows up specific features based on the user’s role</li></ol> <b>Mobile application:</b> <ol style="list-style-type: none"><li>1. Anonymous user opens mobile application</li><li>2. Sign Up Screen will show up</li><li>3. User fills in the sign-up form</li><li>4. User clicks on the “Create Account” button</li></ol>		

	<p>5. Auth0 authorization server validates given credentials and response access token</p> <p>6. Mobile app redirects users to Login Screen</p>
Alternative Flow:	n/a
Exception Flow:	<p><b>Anonymous user fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>Frontend shows messages MSG08</li> </ul> <p><b>The anonymous user provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>Frontend shows message MSG04.</li> </ul> <p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Update Employer Account

ID and Name:	UC102 – Update Employer Account		
Primary Actor:	Employer	Secondary Actor: Admin/Staff	n/a

Description:	
Pre-condition:	Employer
Post-condition:	<p><b>Success:</b> redirected to the employer profile page</p> <p><b>Fail:</b> the system shows an error message</p>
Normal Flow:	<p><b>Web application:</b></p> <ol style="list-style-type: none"><li>1. Employer click the “Change Account” button</li><li>2. He/she will be redirected to Update-Account Page</li><li>3. User fills in the Update-Account form</li><li>4. User clicks on the “Update Account” button</li><li>5. Auth0 authorization server validates given credentials and response token</li><li>6. The browser immediately redirects the user to Employer Profile Page</li></ol> <p><b>Mobile application:</b></p> <ol style="list-style-type: none"><li>1. Employer click the “Change Account” button</li><li>2. Update Screen will show up</li><li>3. User fills in the Update form</li><li>4. User clicks on the “Update Account” button</li><li>5. Auth0 authorization server validates given credentials and response token</li><li>6. Mobile app redirects users to Profile Screen</li></ol>

Alternative Flow:	n/a
Exception Flow:	<p><b>Anonymous user fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>Frontend shows messages MSG08</li> </ul> <p><b>The anonymous user provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>Frontend shows message MSG04.</li> </ul> <p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Post Job Profile

ID and Name:	UC103 – Post Job Profile		
Primary Actor:	Employer	Secondary Actor: Admin/Staff	n/a
Description:			

Pre-condition:	Employer
Post-condition:	<b>Success:</b> redirected to the Home page  <b>Fail:</b> the system shows an error message
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>1. Employer click the “Creating Post” button</li><li>2. He/she will be redirected to Creating-Post Page</li><li>3. User fills in the post form</li><li>4. User clicks on the “Create” button</li><li>5. Auth0 authorization server validates given credentials and response token</li><li>6. The browser immediately redirects the user to Home Page</li></ol> <b>Mobile application:</b> <ol style="list-style-type: none"><li>1. Employer click the “Creating Post” button</li><li>2. Post Screen will show up</li><li>3. User fills in the Post form</li><li>4. User clicks on the “Create” button</li><li>5. Auth0 authorization server validates given credentials and response token</li><li>6. Mobile app redirects users to Home Screen</li></ol>
Alternative Flow:	n/a

Exception Flow:	<p><b>Anonymous user fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>Frontend shows messages MSG08</li> </ul> <p><b>The anonymous user provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>Frontend shows message MSG04.</li> </ul> <p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Update Job Profile

ID and Name:	UC104 – Update Job Profile		
Primary Actor:	Employer	Secondary Actor: Admin/Staff	n/a
Description:			
Pre-condition:	Employer		

Post-condition:	<p><b>Success:</b> redirected to the Home page</p> <p><b>Fail:</b> the system shows an error message</p>
Normal Flow:	<p><b>Web application:</b></p> <ol style="list-style-type: none"><li>1. Employer click the “Update Post” button</li><li>2. He/she will be redirected to Update-Post Page</li><li>3. User fills in the post form</li><li>4. User clicks on the “Update” button</li><li>5. Auth0 authorization server validates given credentials and response token</li><li>6. The browser immediately redirects the user to Home Page</li></ol> <p><b>Mobile application:</b></p> <ol style="list-style-type: none"><li>1. Employer click the “Update Post” button</li><li>2. Post Screen will show up</li><li>3. User fills in the Post form</li><li>4. User clicks on the “Update” button</li><li>5. Auth0 authorization server validates given credentials and response token</li><li>6. Mobile app redirects users to Home Screen</li></ol>
Alternative Flow:	<i>n/a</i>

Exception Flow:	<p><b>Anonymous user fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>Frontend shows messages MSG08</li> </ul> <p><b>The anonymous user provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>Frontend shows message MSG04.</li> </ul> <p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Review CV

ID and Name:	UC105 – Review CV		
Primary Actor:	Employer	Secondary Actor: Admin/Staff	n/a
Description:			
Pre-condition:	Employer		

Post-condition:	<b>Success:</b> redirected to the Manage Posted CV page  <b>Fail:</b> the system shows an error message
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>Employer click the “Manage Posted CV” button</li><li>He/she will be redirected to Posted CV Page</li><li>User fills in the post form</li></ol> <b>Mobile application:</b> <ol style="list-style-type: none"><li>Employer click the “Manage Posted CV” button</li><li>Posted CV Screen will show up</li></ol>
Alternative Flow:	<i>n/a</i>
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"><li>Frontend forwards to Lost Internet page.</li></ul>
Priority:	High
Frequency of Usage:	Sometimes

## Download CV

ID and Name:	UC106 – Download CV		
Primary Actor:	Employer	Secondary Actor: Admin/Staff	n/a
Description:			
Pre-condition:	Employer		
Post-condition:	<b>Success:</b> redirected to the Manage Posted CV page  <b>Fail:</b> the system shows an error message		
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>1. Employer click the “Download CV” button</li><li>2. Auth0 authorization server validates given credentials and response token</li><li>3. The browser immediately send data and started to download</li></ol> <b>Mobile application:</b> <ol style="list-style-type: none"><li>1. Employer click the “Download CV” button</li><li>2. Auth0 authorization server validates given credentials and response token</li></ol>		

	3. The browser immediately send data and started to download
Alternative Flow:	n/a
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"> <li>Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Create Account

ID and Name:	UC107 – Create Account		
Primary Actor:	Employer	Secondary Actor: Admin/Staff	n/a
Description:	Employer create account for other Employer in the company		
Pre-condition:	Employer		

Post-condition:	<p><b>Success:</b> claim access token and will be redirected to the home page</p> <p><b>Fail:</b> the system shows an error message</p>
Normal Flow:	<p><b>Web application:</b></p> <ol style="list-style-type: none"><li>1. Employer click the “Create Account” button</li><li>2. System will redirect to Create new Account Page</li><li>3. Provided information to create</li><li>4. Click the “submit” button</li><li>5. Redirect to home page</li></ol> <p><b>Mobile application:</b></p> <ol style="list-style-type: none"><li>1. Employer user opens mobile application</li><li>2. Click the “Create new Account”</li><li>3. Create Screen will show up</li><li>4. Employer fills in the Create form</li><li>5. Employer clicks on the “Create Account” button</li><li>6. Auth0 authorization server validates given credentials and response access token</li><li>7. Mobile app redirects users to Home Screen</li></ol>
Alternative Flow:	<i>n/a</i>

Exception Flow:	<b>Employer user fails to fill in all the required fields</b> <ul style="list-style-type: none"> <li>Frontend shows messages MSG08</li> </ul> <b>The Employer user provides incorrect credentials</b> <ul style="list-style-type: none"> <li>Frontend shows message MSG04.</li> </ul> <b>Internet connection lost</b> <ul style="list-style-type: none"> <li>Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Controller

### Review & Approve Post Job

ID and Name:	UC201 – Review & Approve Post Job		
Primary Actor:	Admin	Secondary Actor: Staff	n/a
Description:	Admin/Staff review & approve jobs that posted by Employer		
Pre-condition:			

Post-condition:	<b>Success:</b> redirected to the Manage Posted Job page  <b>Fail:</b> the system shows an error message
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>Employer click the “Manage Posted Job” button</li><li>He/she will be redirected to Manage Posted Job Page</li><li>Admin/Staff can read &amp; approve by following button: “View Detail”, “Aprrove”</li></ol> <b>Mobile application:</b> <ol style="list-style-type: none"><li>Employer click the “Manage Posted Job” button</li><li>Manage Screen will show up</li><li>Admin/Staff can read &amp; approve by following button: “View Detail”, “Approve”</li></ol>
Alternative Flow:	n/a
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"><li>Frontend forwards to Lost Internet page.</li></ul>
Priority:	High
Frequency of Usage:	Sometimes

ID and Name:	UC202 – Review CV		
Primary Actor:	Admin	Secondary Actor: Staff	n/a
Description:	Admin/Staff review CV posted by Candidates		
Pre-condition:			
Post-condition:	<p><b>Success:</b> redirected to the Home page</p> <p><b>Fail:</b> the system shows an error message</p>		
Normal Flow:	<p><b>Web application:</b></p> <ol style="list-style-type: none"> <li>1. Employer click the “Read detail” button</li> <li>2. He/she will be redirected to CV Detail Page</li> </ol> <p><b>Mobile application:</b></p> <ol style="list-style-type: none"> <li>1. Employer click the “Read Detail” button</li> <li>2. CV Detail Screen will show up</li> </ol>		
Alternative Flow:	n/a		
Exception Flow:	<p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>• Frontend forwards to Lost Internet page.</li> </ul>		

Priority:	High
Frequency of Usage:	Sometimes

## Create Account

ID and Name:	UC203 – Create Account		
Primary Actor:	Admin	Secondary Actor: Staff	n/a
Description:	Admin/Staff create account for Admin/Staff		
Pre-condition:			
Post-condition:	<b>Success:</b> redirected to the Home page  <b>Fail:</b> the system shows an error message		

Normal Flow:	<p><b>Web application:</b></p> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Create Account for Admin/Staff” button</li><li>2. He/she will be redirected to Create Account Page</li><li>3. Fills all the requirement to create account</li><li>4. Click the “Submit” button to create</li><li>5. Auth0 authorization server validates given credentials and response token</li><li>6. The browser immediately redirects the user to Home Page</li></ol> <p><b>Mobile application:</b></p> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Create Account for Admin/Staff” button</li><li>2. Create Account Screen will show up</li><li>3. Fills all the requirement to create account</li><li>4. Click the submit button</li><li>5. Auth0 authorization server validates given credentials and response token</li><li>6. The browser immediately redirects the user to Home Page</li></ol>
Alternative Flow:	<i>n/a</i>

Exception Flow:	<p><b>Admin/Staff fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>Frontend shows messages MSG08</li> </ul> <p><b>Admin/Staff provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>Frontend shows message MSG04.</li> </ul> <p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## CRUD Account

ID and Name:	UC204 – CRUD Account		
Primary Actor:	Admin	Secondary Actor: Staff	n/a
Description:	Admin/Staff have permission to CRUD account		

Pre-condition:	
Post-condition:	<b>Success:</b> redirected to the Home page  <b>Fail:</b> the system shows an error message
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Manage Account” button</li><li>2. He/she will be redirected to Manage Account Page</li><li>3. Admin/Staff can manage all account at here</li></ol> <b>Mobile application:</b> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Manage Account” button</li><li>2. Manage Account Screen will show up</li></ol>
Alternative Flow:	<i>n/a</i>
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"><li>• Frontend forwards to Lost Internet page.</li></ul>
Priority:	High

Frequency of Usage:	Sometimes
---------------------	-----------

## Monitor Candidate Behavior

ID and Name:	UC205 – Monitor Candidate Behavior		
Primary Actor:	Admin	Secondary Actor: Staff	n/a
Description:	Admin/Staff access to Candidate's Behavior manage		
Pre-condition:			
Post-condition:	<b>Success:</b> redirected to the Home page  <b>Fail:</b> the system shows an error message		
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Manage Behavior” button</li><li>2. He/she will be redirected to Manage Behavior Page</li><li>3. Admin/Staff see the statistics here</li></ol> <b>Mobile application:</b>		

	<ol style="list-style-type: none"> <li>1. Admin/Staff click the “Manage Behavior” button</li> <li>2. Statistics Screen will show up</li> </ol>
Alternative Flow:	n/a
Exception Flow:	<p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>• Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Report Job Trending & Sale

ID and Name:	UC206 – Report Job Trending & Sale		
Primary Actor:	Admin	Secondary Actor: Staff	n/a
Description:	Admin/Staff can manage the statistic of Job trending & sale		
Pre-condition:			

Post-condition:	<b>Success:</b> redirected to the Home page  <b>Fail:</b> the system shows an error message
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Statistics of Job Trending &amp; Sale” button</li><li>2. He/she will be redirected to Statistics Page</li></ol> <b>Mobile application:</b> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Statistics of Job Trending &amp; Sale” button</li><li>2. Statistics Screen will show up</li><li>3. Admin/Staff see the statistics of Job Trending &amp; Sale here</li></ol>
Alternative Flow:	<i>n/a</i>
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"><li>• Frontend forwards to Lost Internet page.</li></ul>
Priority:	High
Frequency of Usage:	Sometimes

## CRUD Candidate & Employer

ID and Name:	UC207 – CRUD Candidate & Employer		
Primary Actor:	Admin	Secondary Actor: Staff	n/a
Description:	Admin/Staff have permission to CRUD Candidate & Employer		
Pre-condition:			
Post-condition:	<b>Success:</b> redirected to the Home page  <b>Fail:</b> the system shows an error message		
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Manage Account” button</li><li>2. He/she will be redirected to Manage Account Page</li><li>3. Admin/Staff can manage candidate &amp; employer here</li></ol> <b>Mobile application:</b> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Manage Account” button</li><li>2. Manage Account Screen will show up</li></ol>		

	3. Admin/Staff can manage candidate & employer here
Alternative Flow:	n/a
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"> <li>Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## 2.15 CRUD Jobs Posting

ID and Name:	UC208 – CRUD Job Posting		
Primary Actor:	Admin	Secondary Actor: Staff	n/a
Description:	Admin/Staff have permission to CRUD Job Posting		
Pre-condition:			

Post-condition:	<b>Success:</b> redirected to the Home page  <b>Fail:</b> the system shows an error message
Normal Flow:	<b>Web application:</b>  2. Admin/Staff click the “Manage Job Posting” button 2. He/she will be redirected to Manage Job Posting Page 3. Admin/Staff can manage Job Posting here  <b>Mobile application:</b>  4. Admin/Staff click the “Manage Job Posting” button 5. Manage Job Posting Screen will show up 6. Admin/Staff can manage Job Posting here
Alternative Flow:	n/a
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"><li>Frontend forwards to Lost Internet page.</li></ul>
Priority:	High
Frequency of Usage:	Sometimes

## 2.16 CRUD Comments

ID and Name:	UC209 – CRUD Comments		
Primary Actor:	Admin	Secondary Actor: Staff	n/a
Description:	Admin/Staff have permission to CRUD Comments		
Pre-condition:			
Post-condition:	<b>Success:</b> redirected to the Home page  <b>Fail:</b> the system shows an error message		
Normal Flow:	<b>Web application:</b> <ol style="list-style-type: none"><li>1. Admin/Staff click the “CRUD Comments” button</li><li>2. He/she will be redirected to Manage Posted Comments Page</li><li>3. Admin/Staff can manage Posted Comments here</li></ol> <b>Mobile application:</b> <ol style="list-style-type: none"><li>1. Admin/Staff click the “Manage Posted Comments” button</li><li>2. Manage Posted Comments Screen will show up</li><li>3. Admin/Staff can manage Posted Comments here</li></ol>		

Alternative Flow:	n/a
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"> <li>Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## UnRegister Candidate

### View all jobs list

ID and Name:	UC101 – View all jobs list		
Primary Actor:	Any user	Secondary Actor:	n/a
Description:	The jobs list available on the home page		
Pre-condition:			
Post-condition:			

Normal Flow:	Jobs list is available on the homepage of the website or app mobile
Alternative Flow:	n/a
Exception Flow:	
Priority:	High
Frequency of Usage:	Always

## Sign In

ID and Name:	UC201 - Sign In		
Primary Actor:	Unregister user	Secondary Actor:	n/a
Description:	Unregister user login into the system, and give access to specific features according to his/her role		
Pre-condition:	Unregister user		
Post-condition:	<b>Success:</b> claim access token and will be redirected to home page <b>Fail:</b> the system show error message		
Normal Flow:	1. Unregister user tap on Sign In button 2. He/she will be redirected to Sign In Page		

	<p>3. User fill in the sign In form:</p> <ul style="list-style-type: none"> <li>○ Username/Email: text input</li> <li>○ Password: password input</li> </ul> <p>4. User clicks on “Sign In” button</p> <p>5. Auth0 authorization server validates given credentials and response access token</p> <p>6. The browser immediately redirects the user to Home Page</p> <p>7. Home Page shows up specific features based on the user’s role</p>
Alternative Flow:	n/a
Exception Flow:	<p><b>Anonymous user fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>● Frontend shows messages MSG08</li> </ul> <p><b>Anonymous user provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>● Frontend shows message MSG04.</li> </ul> <p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>● Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## View Job details

ID and Name:	UC202 – View Job details		
Primary Actor:	Register user	Secondary Actor:	n/a

Description:	Details of the jobs
Pre-condition:	Unregister user
Post-condition:	
Normal Flow:	<ol style="list-style-type: none"> <li>1. Sign In by the available account</li> <li>2. User tap on the job</li> <li>3. He/she will be redirected to Job details page</li> </ol>
Alternative Flow:	<i>n/a</i>
Exception Flow:	<p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>• Frontend forwards to Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Always

## Search jobs

ID and Name:	UC203 – Search jobs
--------------	---------------------

Primary Actor:	Register user	Secondary Actor:	n/a
Description:	Find jobs by location, skill, company, salary range		
Pre-condition:	Register user		
Post-condition:	<b>Success:</b> Show job that related to the search keyword		
Normal Flow:	<ol style="list-style-type: none"> <li>1. Sign In by the available account</li> <li>2. User enter keyword into the Search box on homepage and submit</li> <li>3. Website of app mobile will show the list of jobs that related to the keyword that he/she provided</li> </ol>		
Alternative Flow:	n/a		
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"> <li>• Frontend forwards to the Lost Internet page.</li> </ul>		
Priority:	High		
Frequency of Usage:	Always		

## Take some Tests

ID and Name:	UC204 – Take some Tests		
Primary Actor:	Register user	Secondary Actor:	n/a
Description:	To analyse your personal characteristics and abilities to determine whether you are suitable for the job's working environment.		
Pre-condition:	Register user		
Post-condition:	<b>Success:</b> Show The mark of the test and some Diagnostic for the test		
Normal Flow:	<ol style="list-style-type: none"><li>1. Sign In by the available account</li><li>2. Tap on the “take test” button</li><li>3. Redirect to the Test page where the user will provide answers for the question on the form and submit.</li><li>4. The mark of the test and some Diagnostic will be shown.</li></ol>		
Alternative Flow:	n/a		
Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"><li>• Frontend forwards to the Lost Internet page.</li></ul>		

Priority:	High
Frequency of Usage:	Sometimes

## Comment/view review on company

ID and Name:	UC205 – Comment/view review on company		
Primary Actor:	Register user	Secondary Actor:	n/a
Description:	See review on company and make some review for the company		
Pre-condition:	Register user		
Post-condition:	<b>Success:</b> pop-up the success message after submit the comment		
Normal Flow:	<ol style="list-style-type: none"> <li>1. Sign In by the available account</li> <li>2. Go the the Detail page of a company</li> <li>3. Tap on the “Comment/Review” button</li> <li>4. Redirect to the Review page where the user see the review of that company.</li> <li>5. Tap on the Comment button to make comment</li> </ol>		
Alternative Flow:	n/a		

Exception Flow:	<b>Internet connection lost</b> <ul style="list-style-type: none"> <li>Frontend forwards to the Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Predict salary

ID and Name:	UC206 – Predict salary		
Primary Actor:	Register user	Secondary Actor:	n/a
Description:	Predict salary based on Machine Learning big data		
Pre-condition:	Register user		
Post-condition:	<b>Success:</b> Show predicted salary for the job		

Normal Flow:	<ol style="list-style-type: none"> <li>1. Sign In by the available account</li> <li>2. Tap on “Predict salary” button</li> <li>3.</li> </ol>
Alternative Flow:	n/a
Exception Flow:	<p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>• Frontend forwards to the Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Create new CV

ID and Name:	UC207 – Create new CV		
Primary Actor:	Register user	Secondary Actor:	n/a
Description:	Fill CV templates of website		

Pre-condition:	Register user
Post-condition:	<b>Success:</b> pop-up the success message after submit the new CV
Normal Flow:	<ol style="list-style-type: none"> <li>1. Sign In by the available account</li> <li>2. Go to personal page</li> <li>3. Tap on Create CV and the CV templates will appear</li> <li>4. Fill the CV template and submit</li> </ol>
Alternative Flow:	<i>n/a</i>
Exception Flow:	<p><b>Anonymous user fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>• Frontend shows messages MSG08</li> </ul> <p><b>The anonymous user provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>• Frontend shows message MSG04.</li> </ul> <p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>• Frontend forwards to the Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

**Upload new CV**

ID and Name:	UC208 – Upload new CV		
Primary Actor:	Register user	Secondary Actor:	n/a
Description:	Upload CV with format PDF on website		
Pre-condition:	Register user		
Post-condition:	<b>Success:</b> pop-up the success message after upload		
Normal Flow:	<ol style="list-style-type: none"> <li>1. Sign In by the available account</li> <li>2. Go to personal page</li> <li>3. Tap on Upload CV and choose the pdf CV file</li> <li>4. Submit to upload the CV to your personal info</li> </ol>		
Alternative Flow:	n/a		
Exception Flow:	<p><b>Anonymous user fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>• Frontend shows messages MSG08</li> </ul> <p><b>The anonymous user provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>• Frontend shows message MSG04.</li> </ul>		

	<b>Internet connection lost</b> <ul style="list-style-type: none"><li>Frontend forwards to the Lost Internet page.</li></ul>
Priority:	High
Frequency of Usage:	Sometimes

## Submit CV

ID and Name:	UC209 – Submit CV		
Primary Actor:	Register user	Secondary Actor:	n/a
Description:	Submit CV to the job		
Pre-condition:	Register user		
Post-condition:	<b>Success:</b> pop-up the success message after submit		

Normal Flow:	<ol style="list-style-type: none"> <li>1. Sign In by the available account</li> <li>2. Go to Details job page</li> <li>3. Tap on Submit CV and choose the CV from your personal info</li> </ol>
Alternative Flow:	n/a
Exception Flow:	<p><b>Anonymous user fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>• Frontend shows messages MSG08</li> </ul> <p><b>The anonymous user provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>• Frontend shows message MSG04.</li> </ul> <p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>• Frontend forwards to the Lost Internet page.</li> </ul>
Priority:	High
Frequency of Usage:	Sometimes

## Add job to favourite

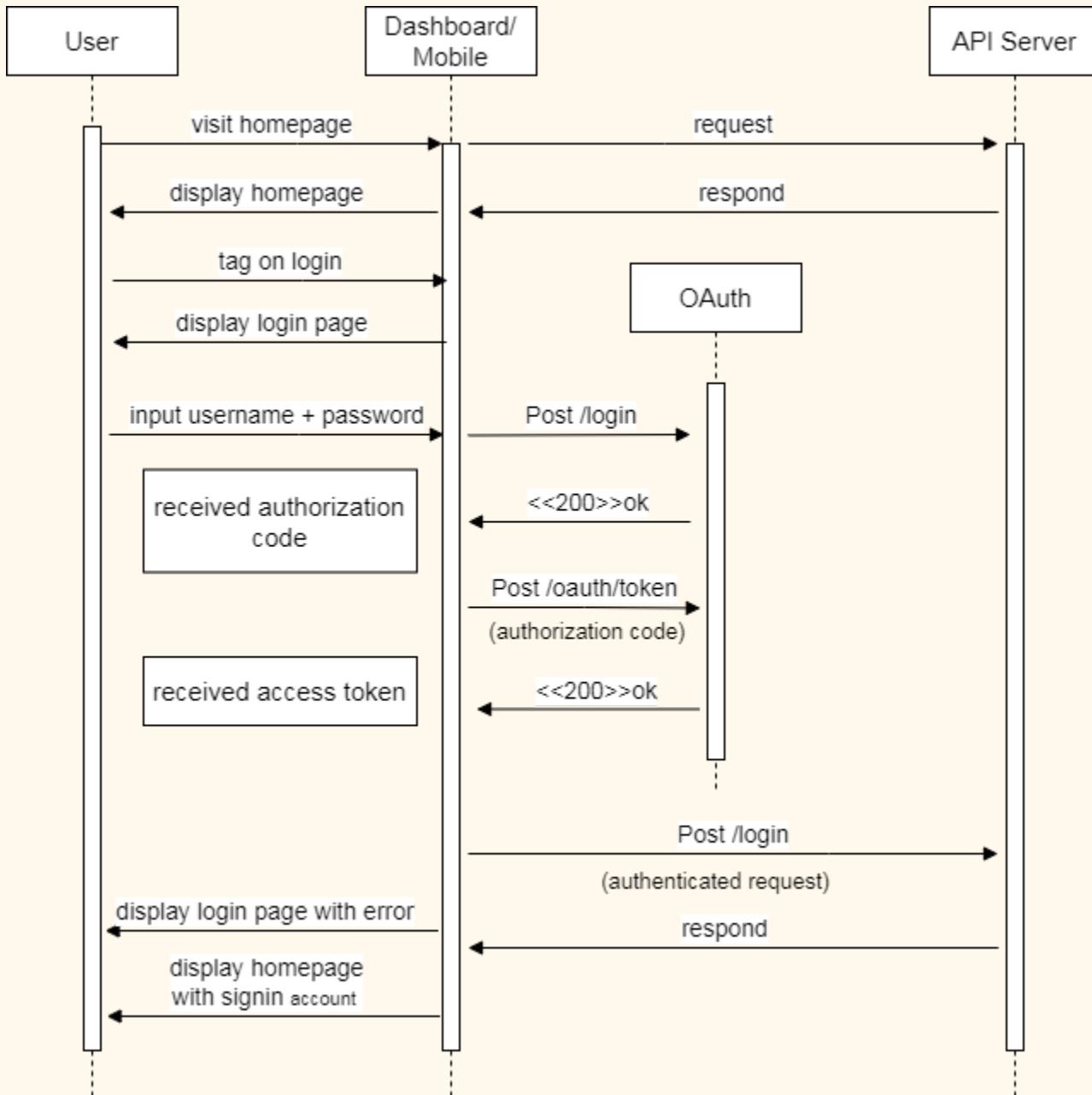
ID and Name:	UC210 – Add job to favourite
--------------	------------------------------

Primary Actor:	Register user	Secondary Actor:	n/a
Description:	Add considerable job to the favourite list so you don't have to search for it again		
Pre-condition:	Register user		
Post-condition:	<b>Success:</b> pop-up the success message after click on the heart icon		
Normal Flow:	<ol style="list-style-type: none"> <li>1. Sign In by the available account</li> <li>2. Go to Details job page</li> <li>3. Tap on the Heart Icon</li> </ol>		
Alternative Flow:	n/a		
Exception Flow:	<p><b>Anonymous user fails to fill in all the required fields</b></p> <ul style="list-style-type: none"> <li>• Frontend shows messages MSG08</li> </ul> <p><b>The anonymous user provides incorrect credentials</b></p> <ul style="list-style-type: none"> <li>• Frontend shows message MSG04.</li> </ul> <p><b>Internet connection lost</b></p> <ul style="list-style-type: none"> <li>• Frontend forwards to the Lost Internet page.</li> </ul>		
Priority:	High		

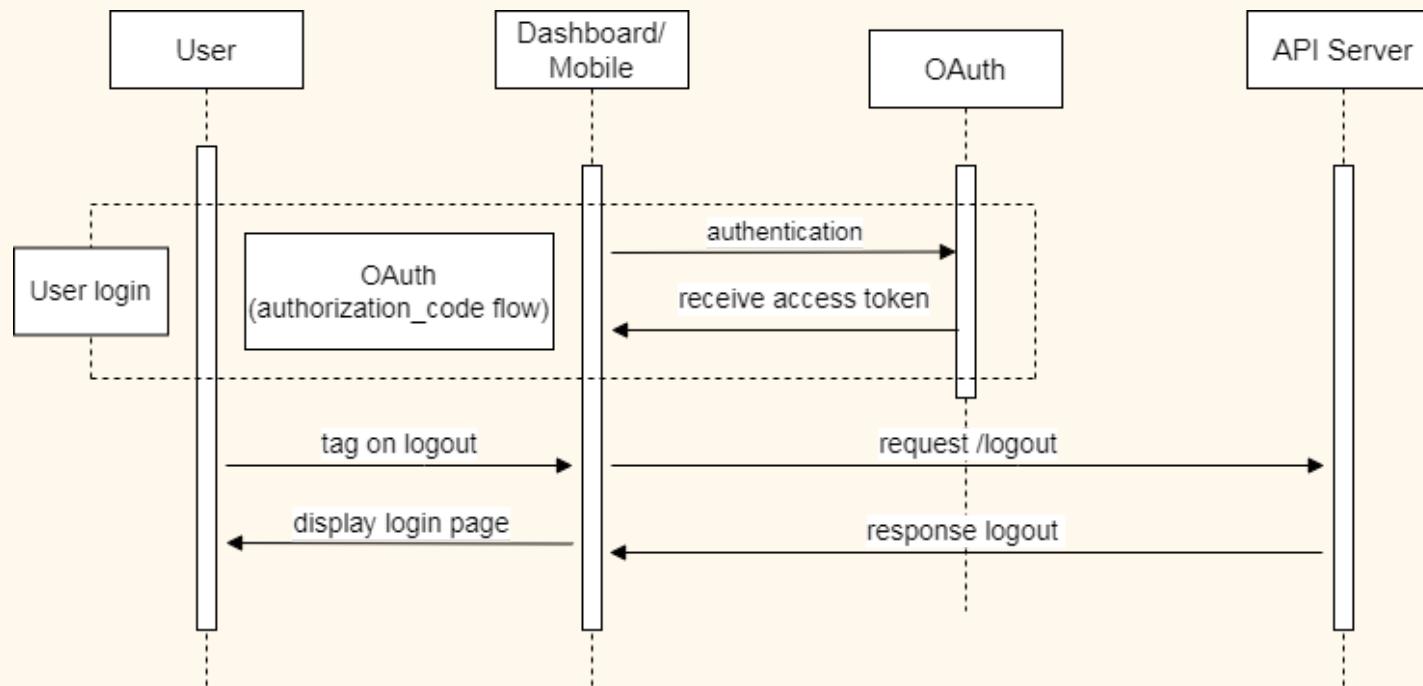
Frequency of Usage:	Sometimes
---------------------	-----------

### 2.2.3 SEQUENCE DIAGRAM

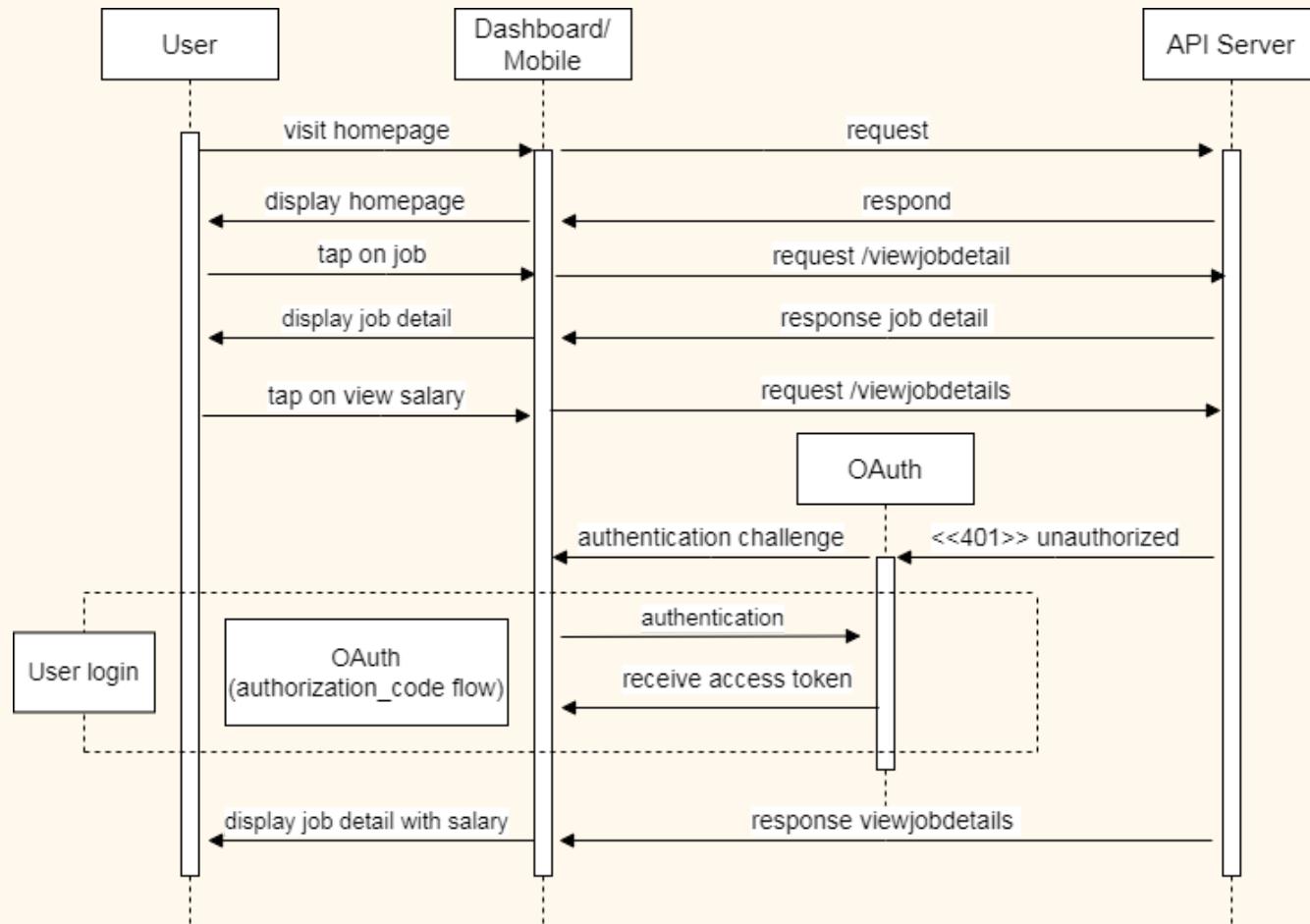
#### Login in



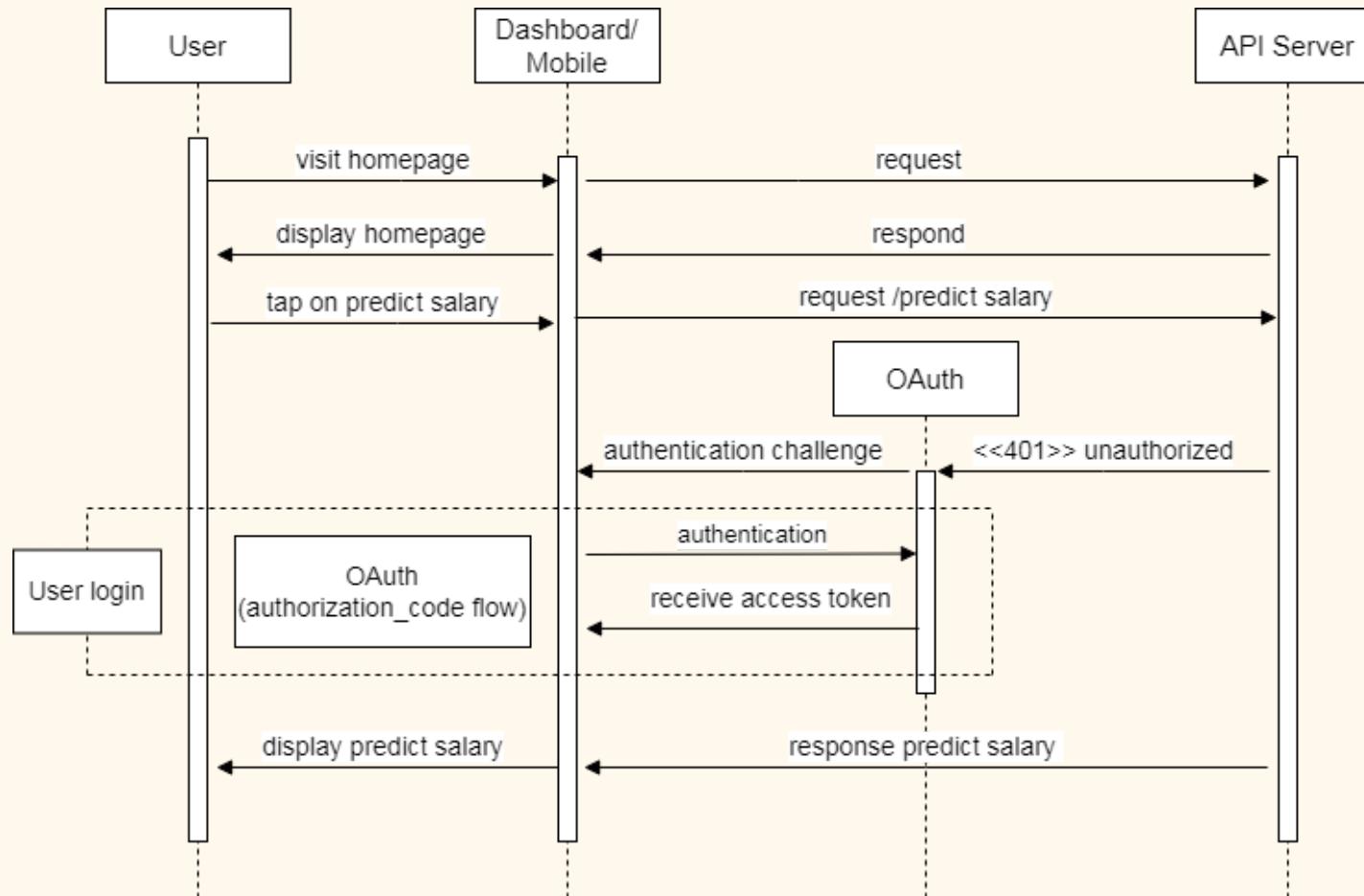
## Log out



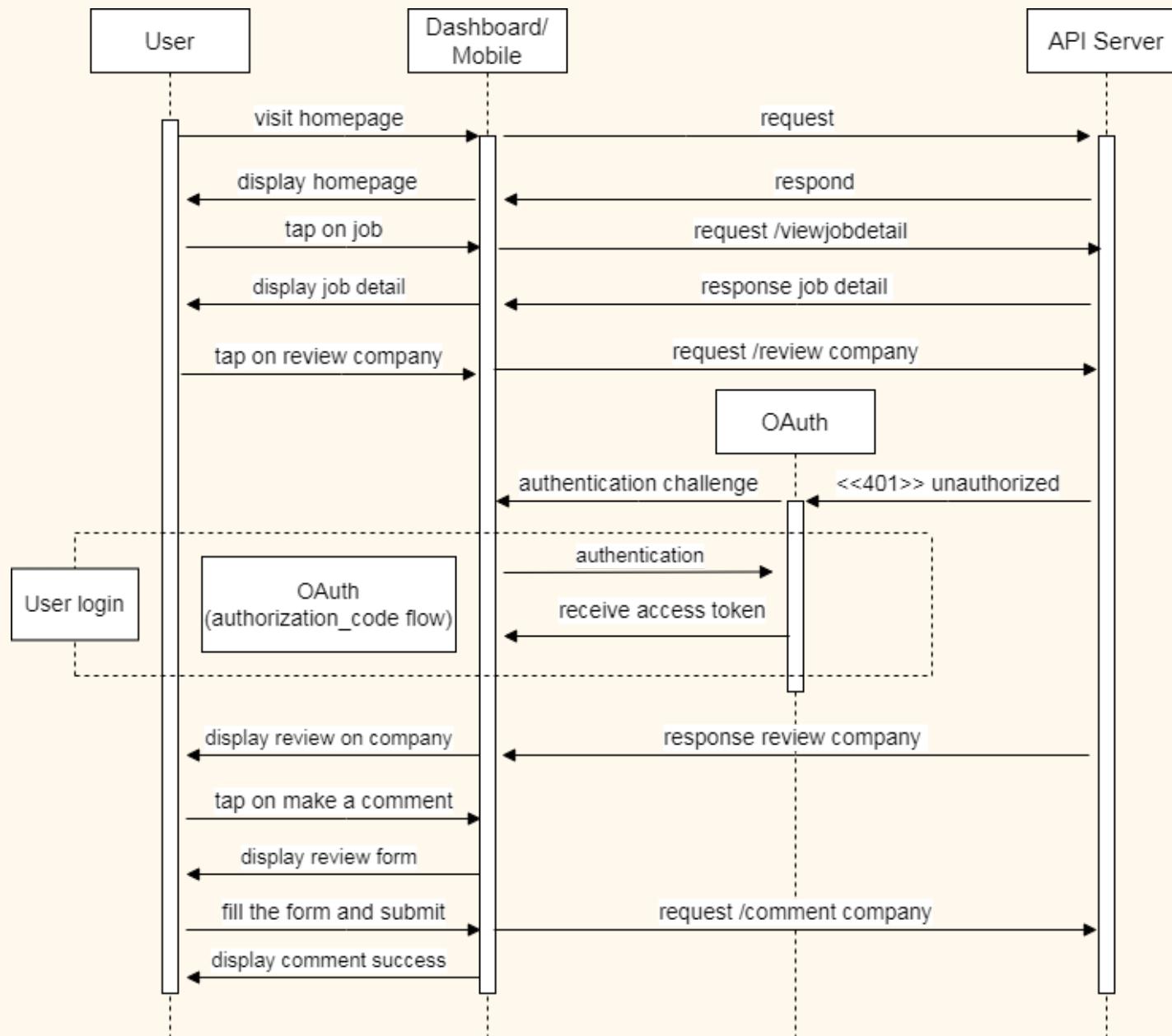
## View job details



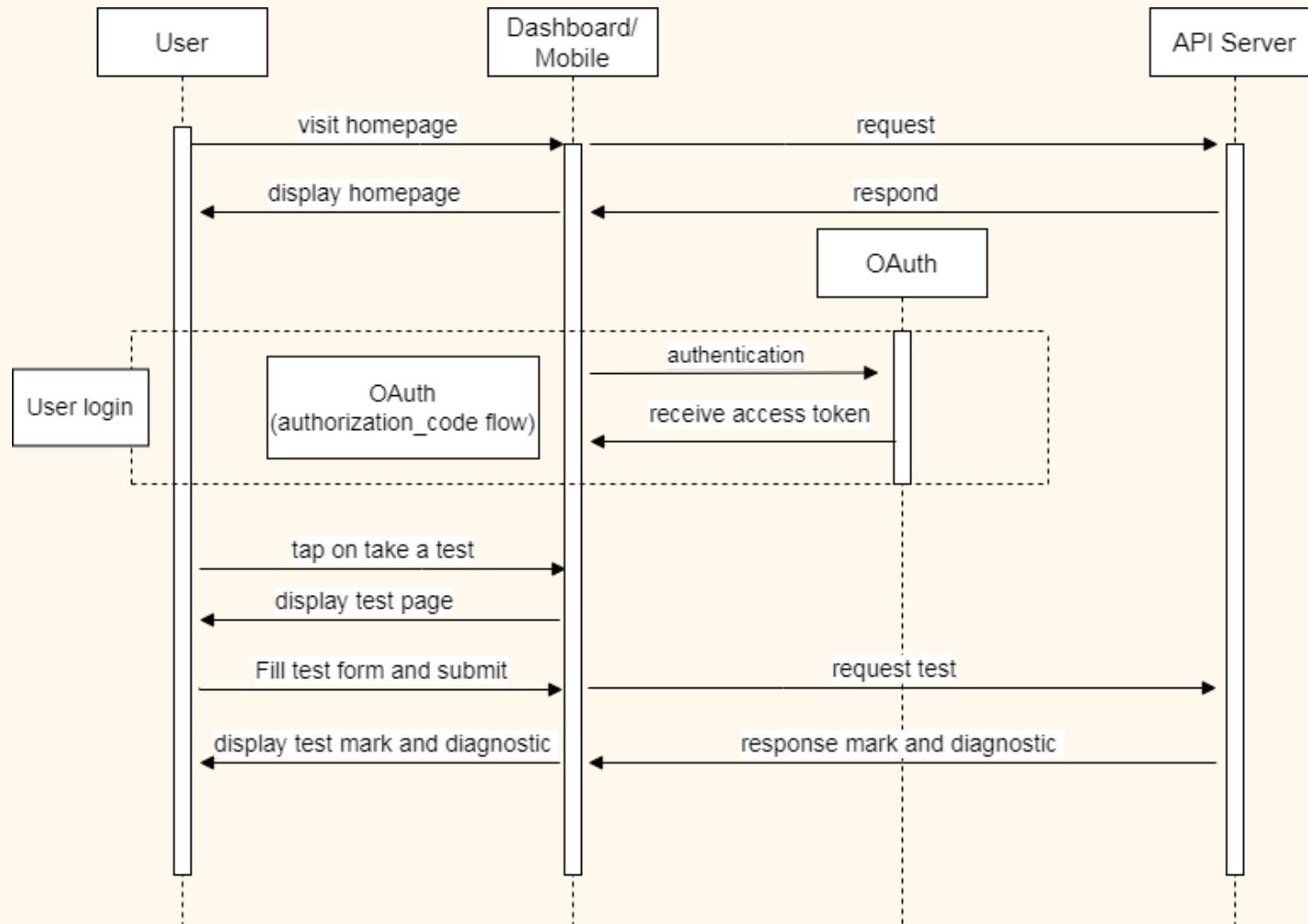
**predict salary**

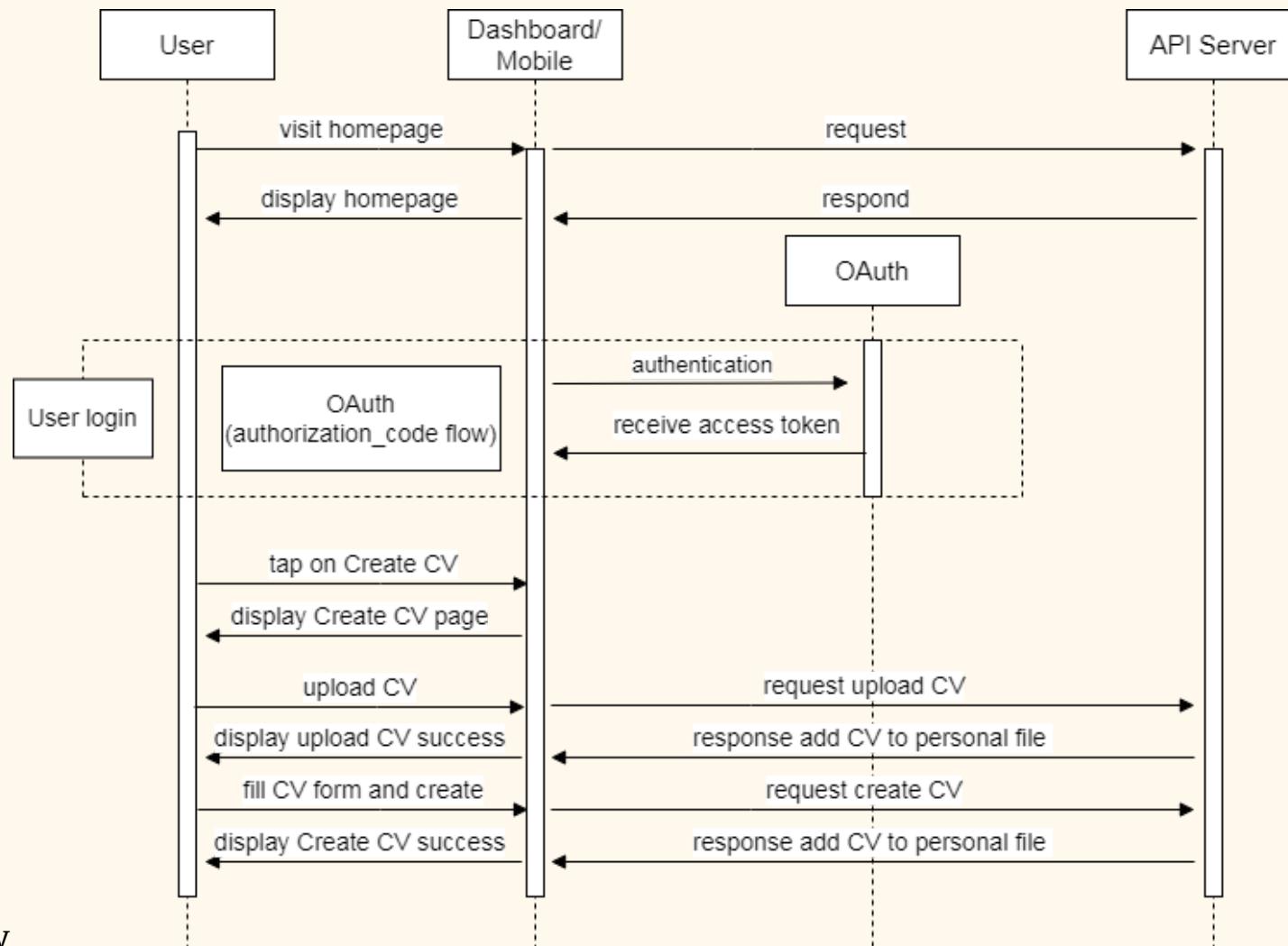


**Comment review**

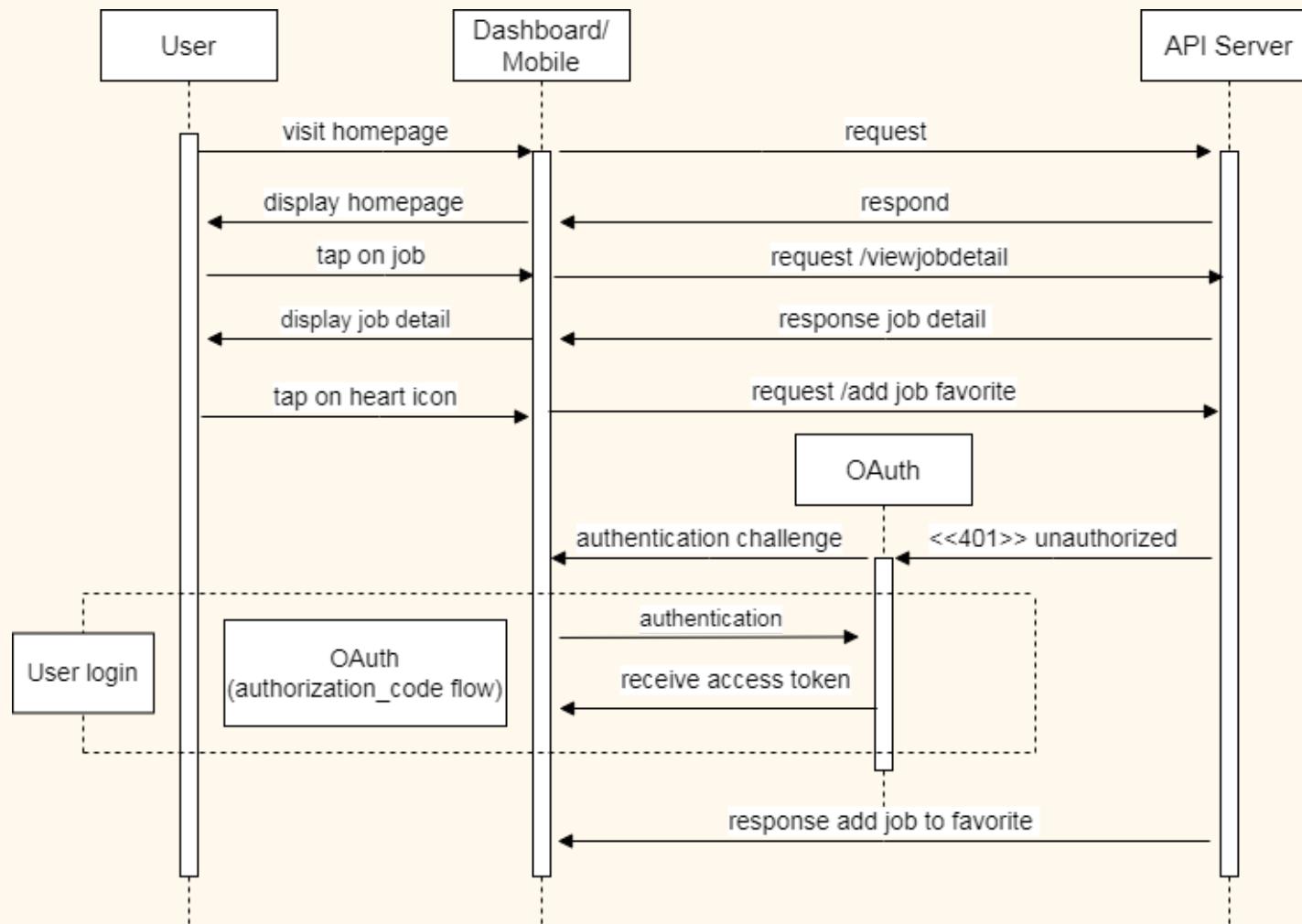


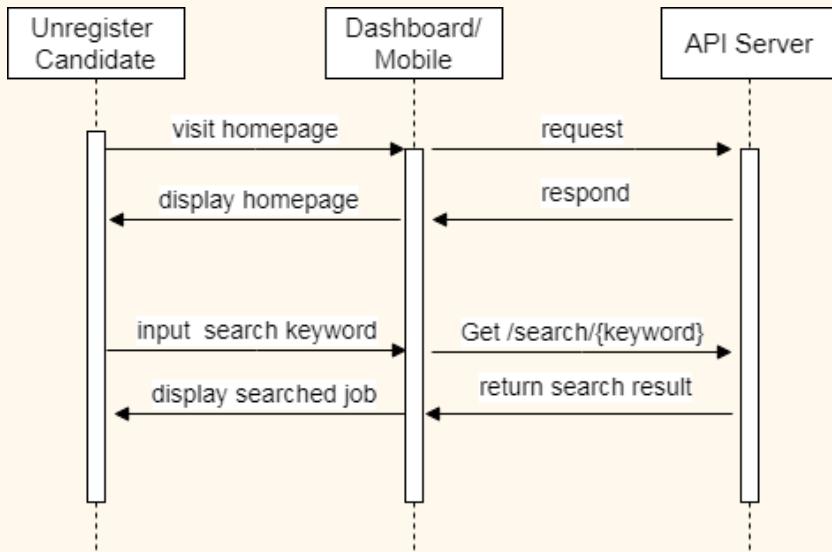
**do test/examination**



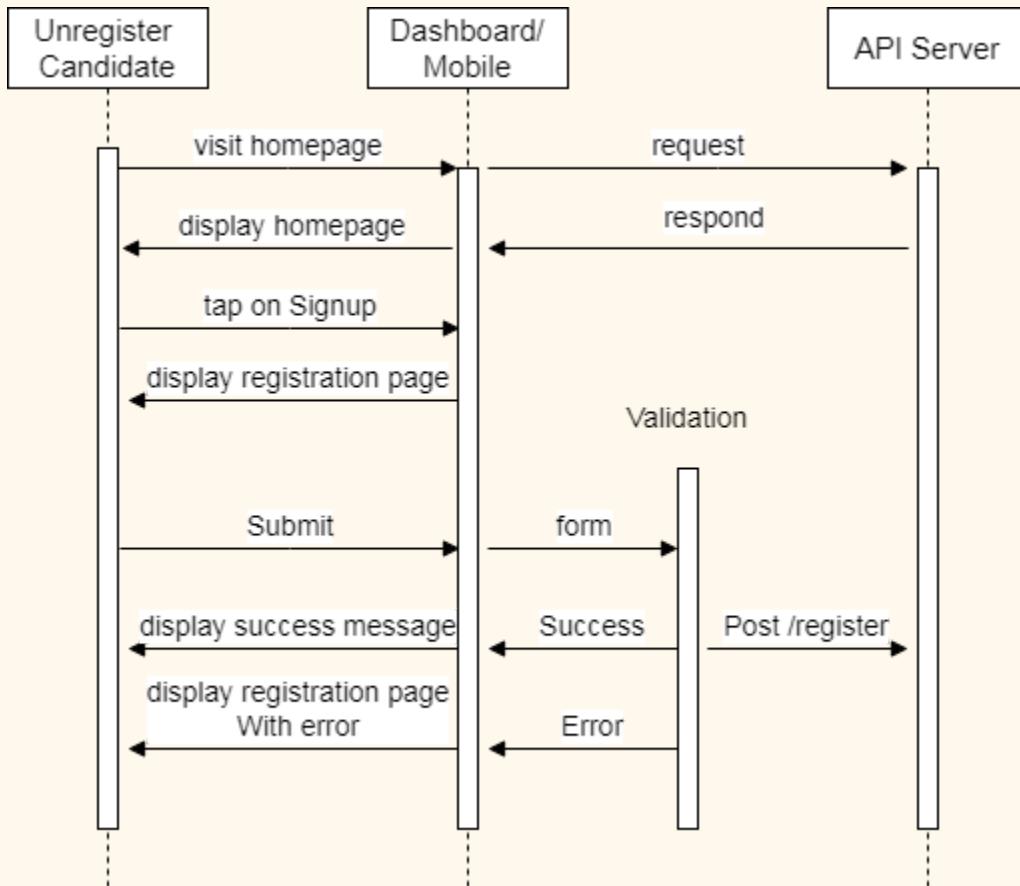
**Create CV**

**add job to favourite**

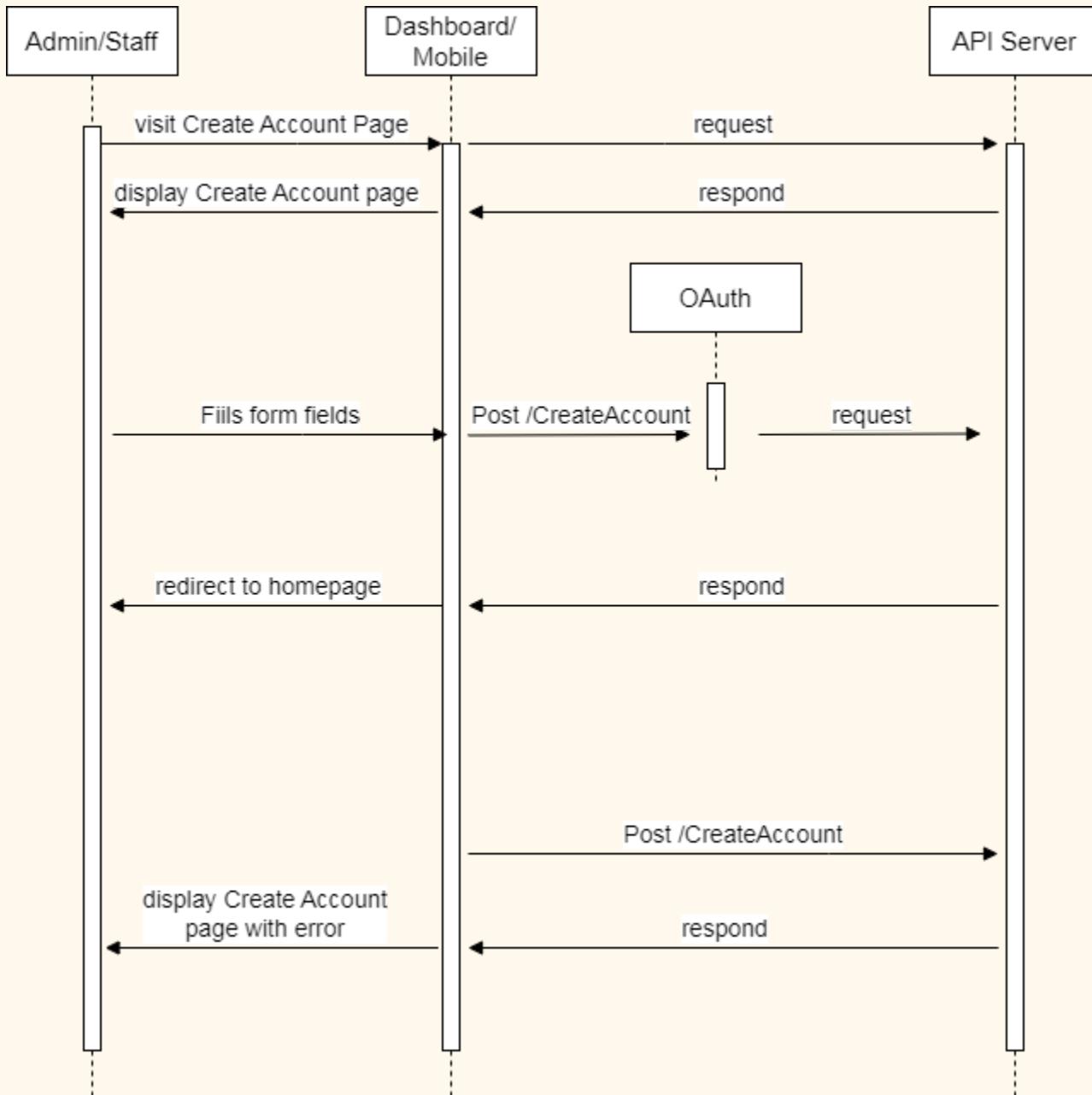
**search job**

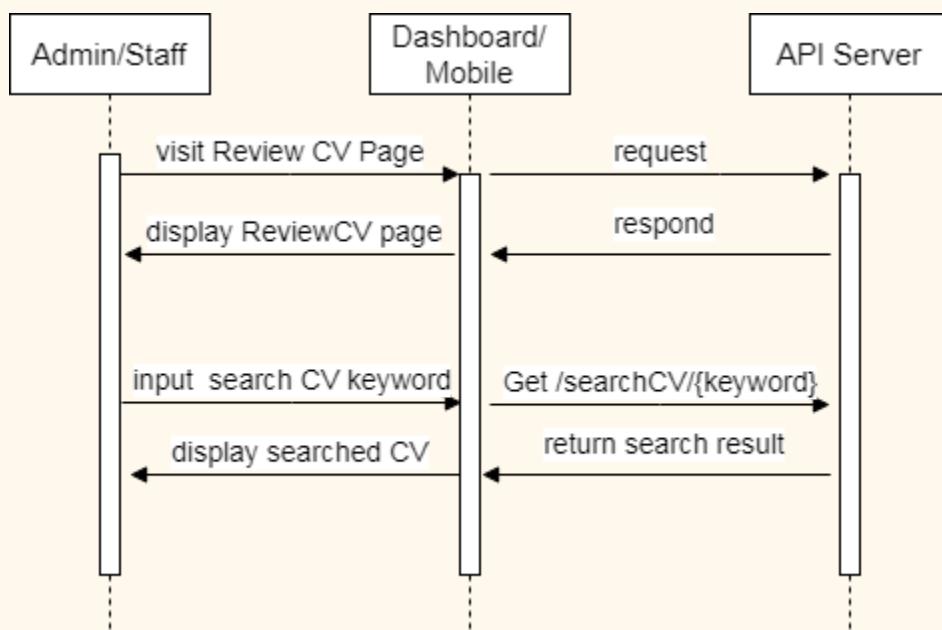


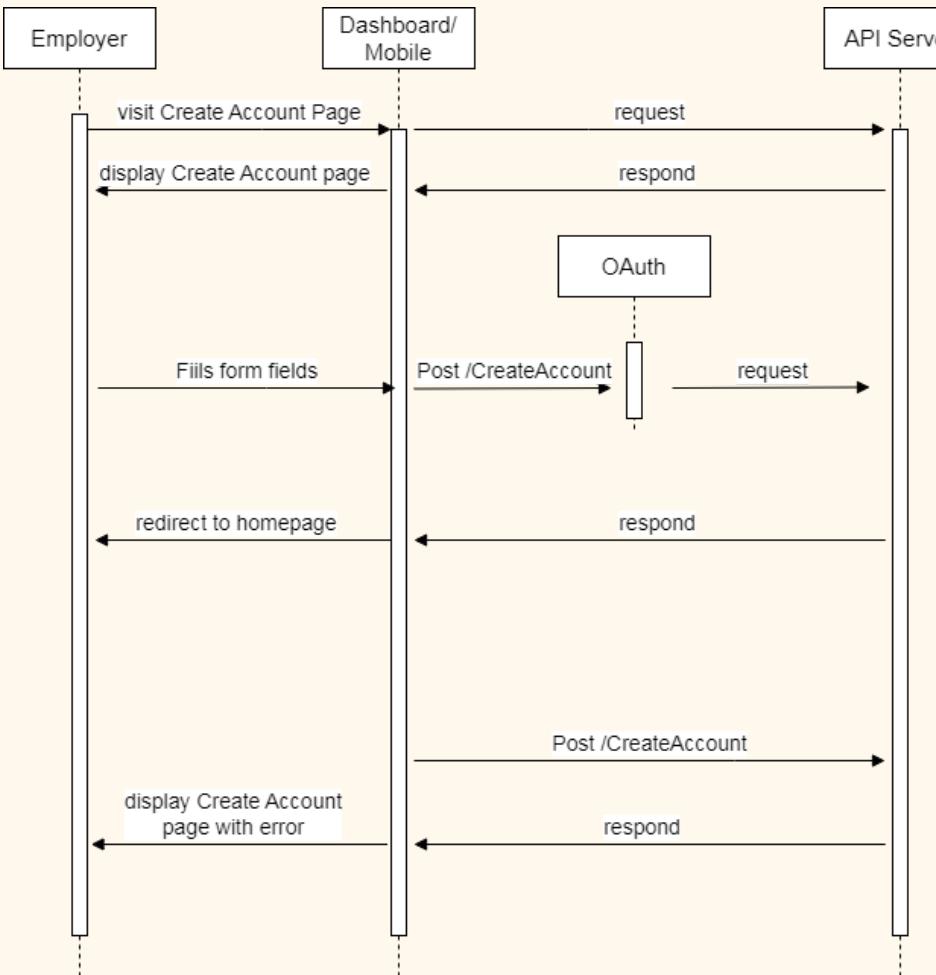
**sign up**

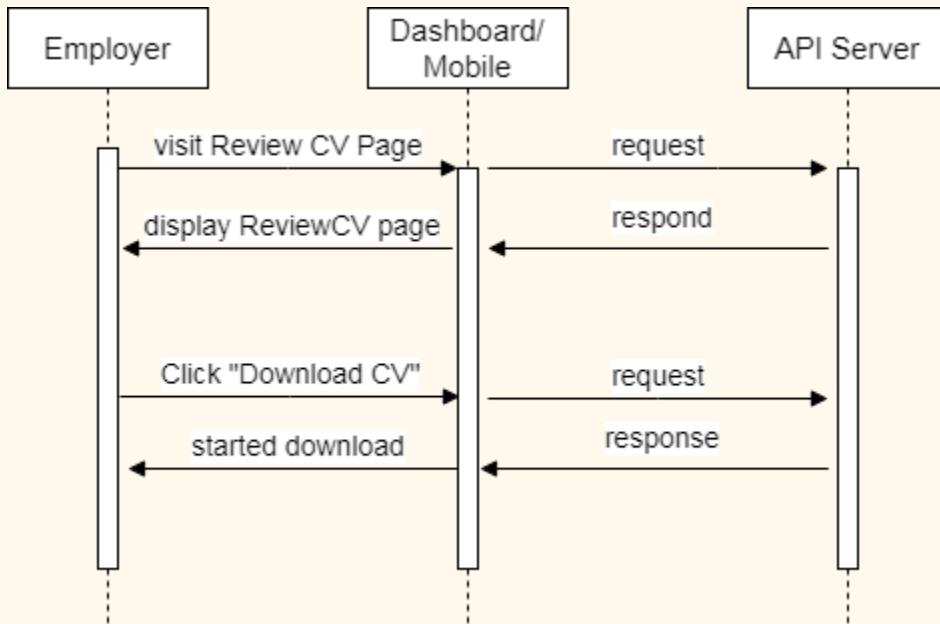


**Controller create account**

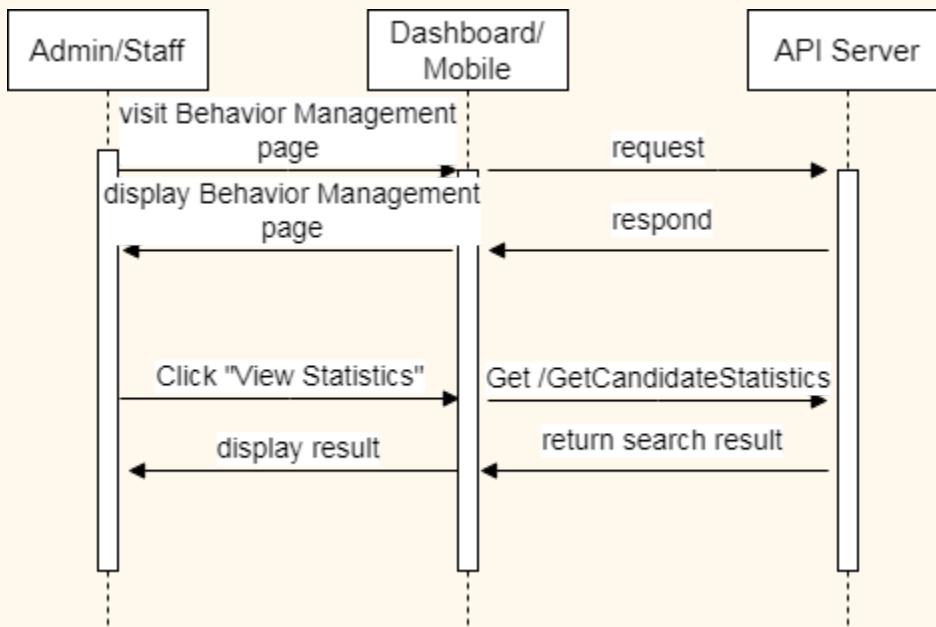


**controller review CV****controller create account**

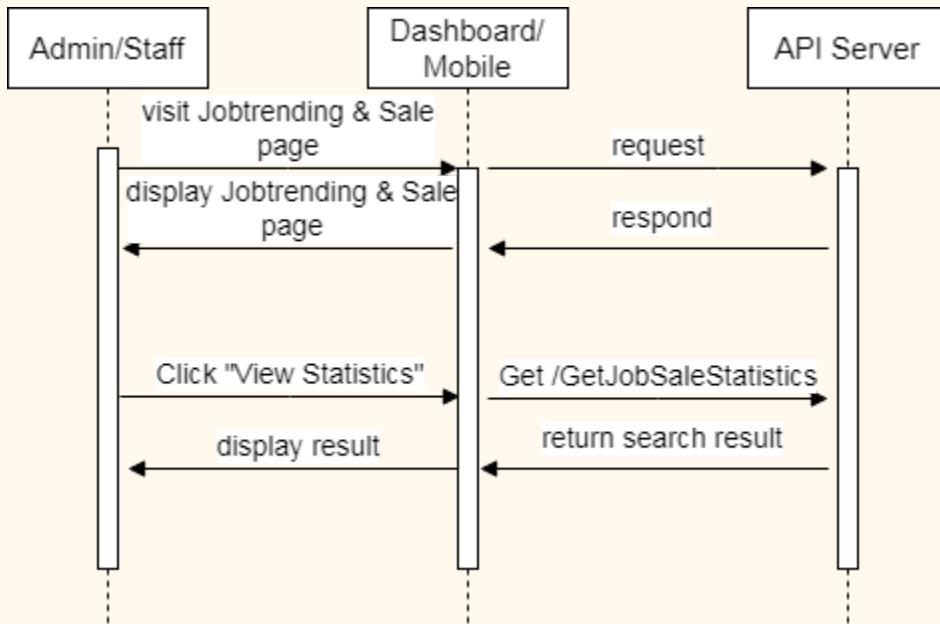
**Controller Download CSV**



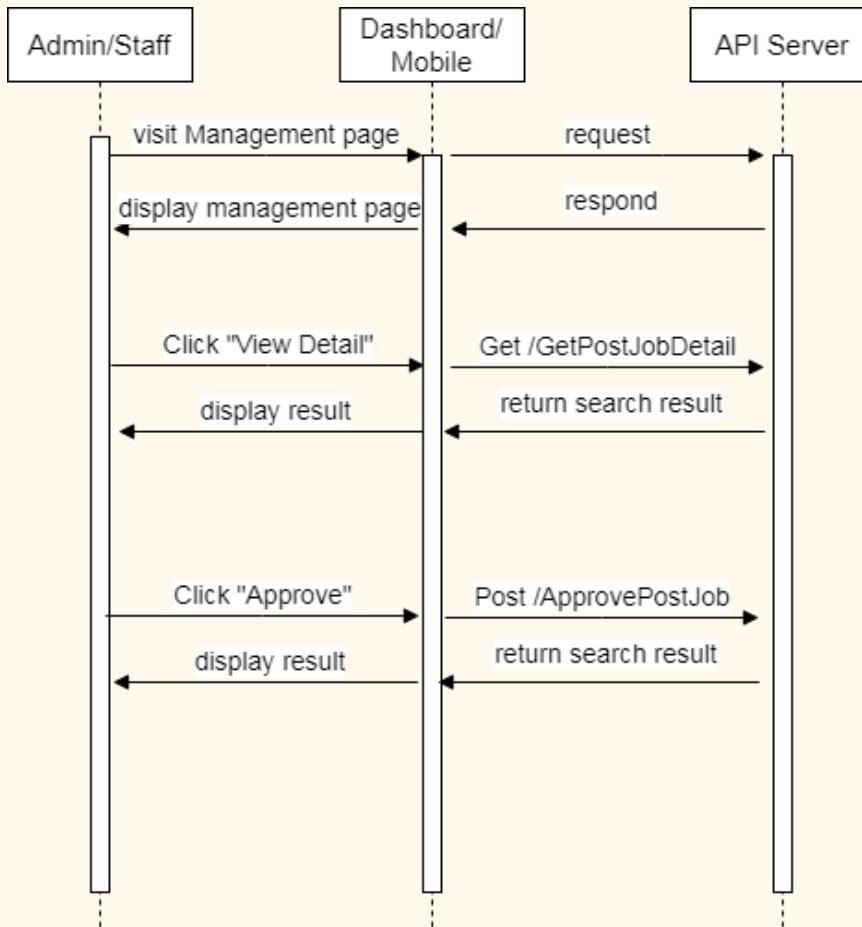
### Monitor candidate behavior

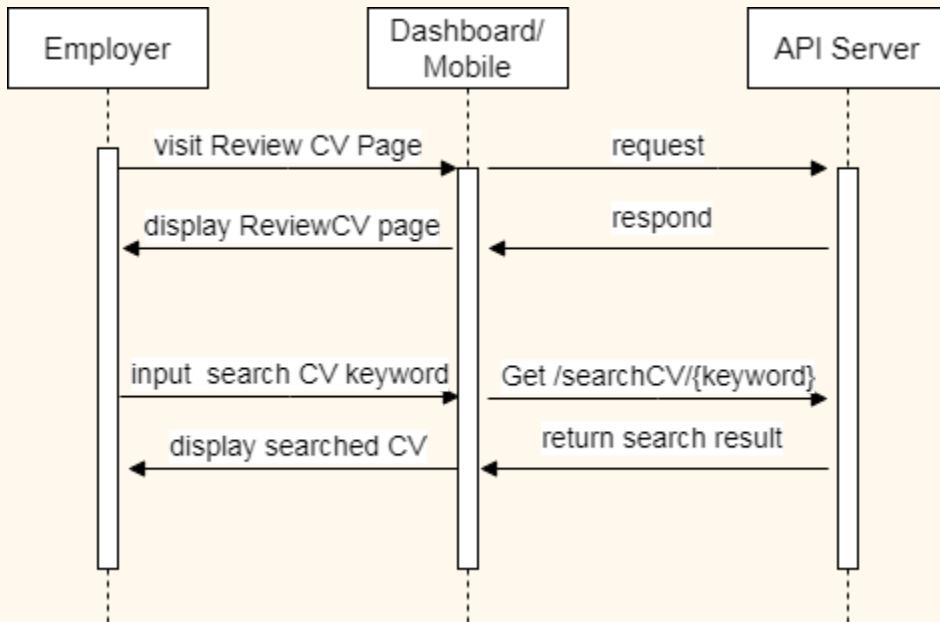


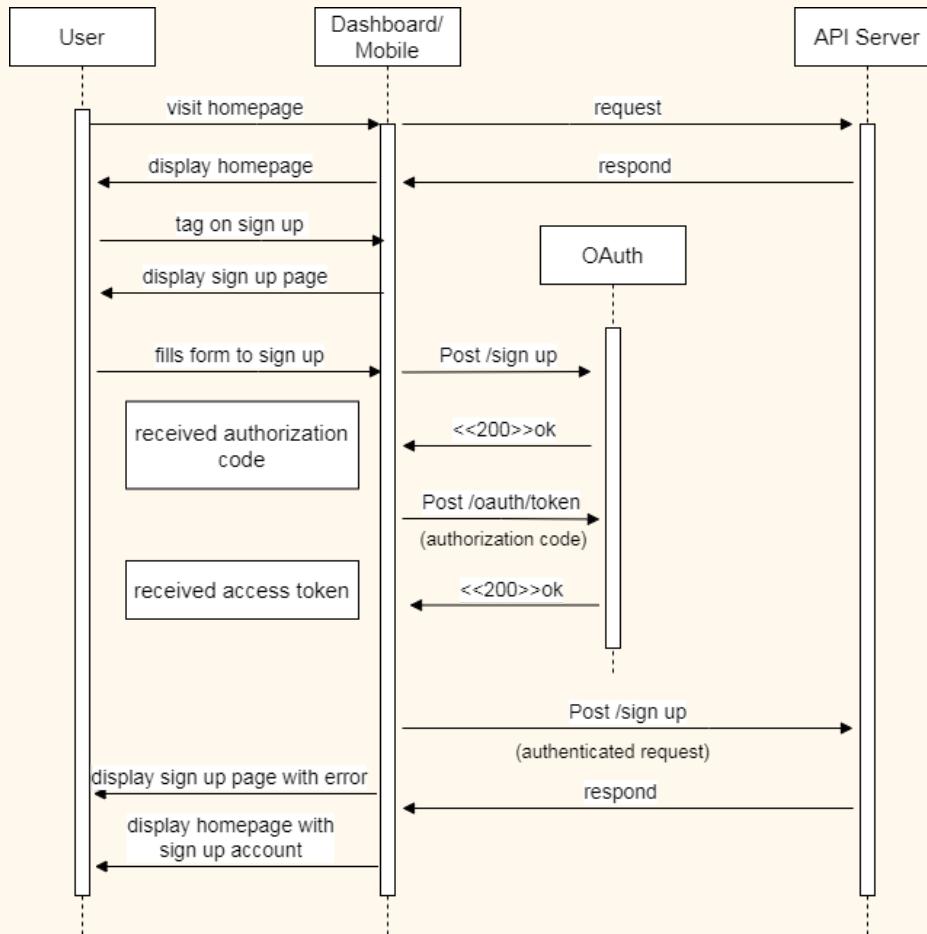
### Report job trending



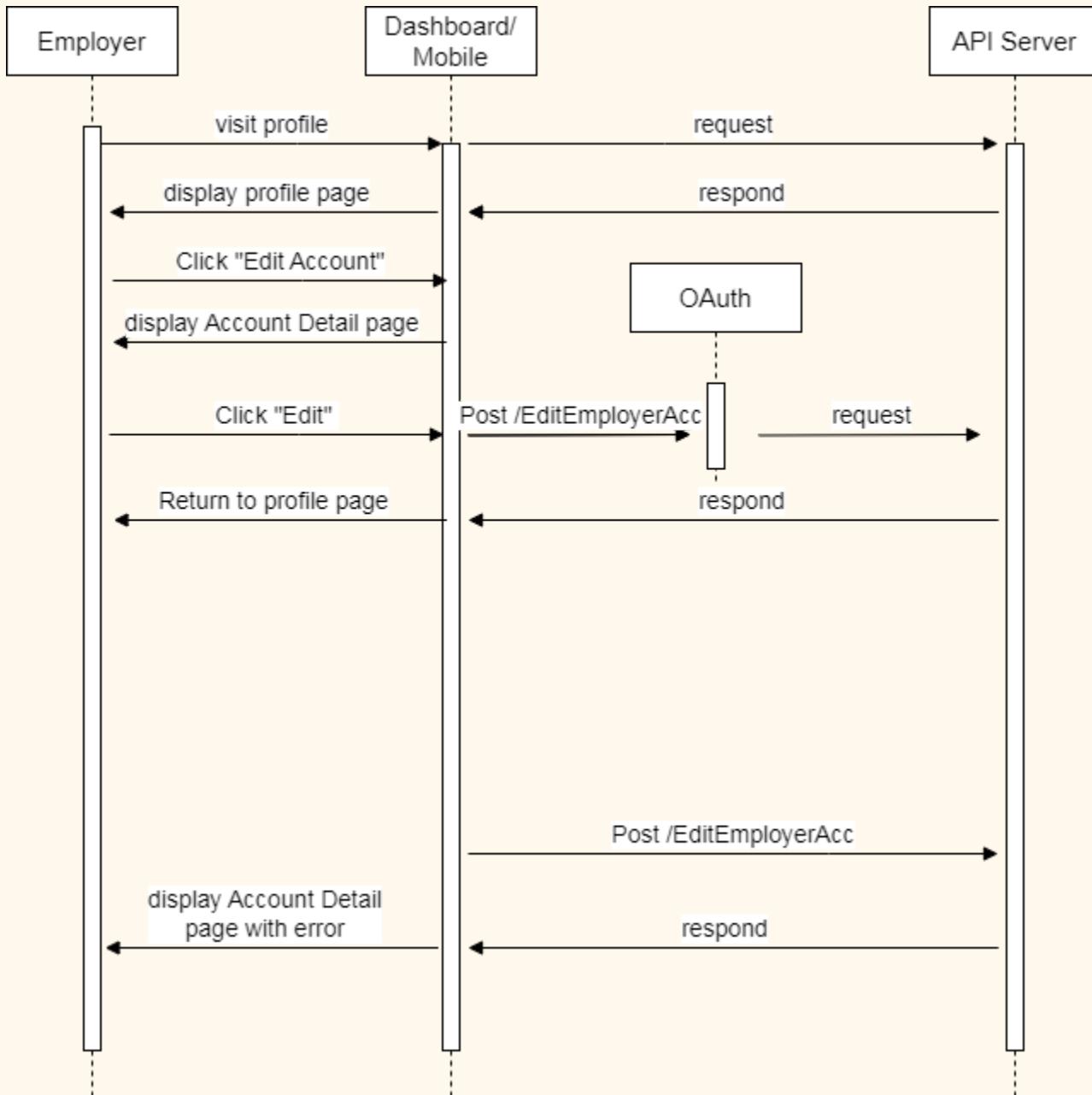
### Review and approve job post

**Review CV**

**Sign up**



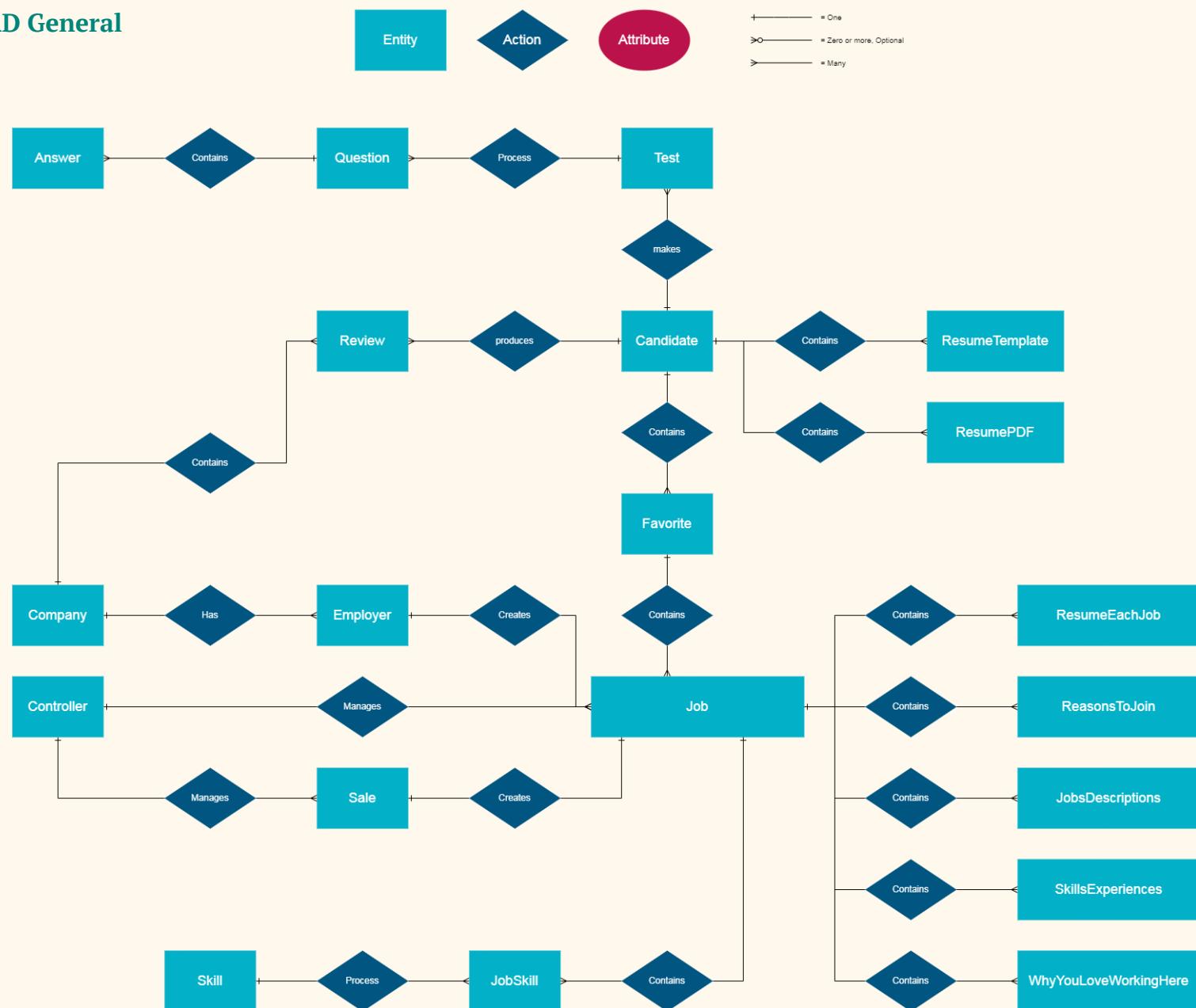
**update employer account**



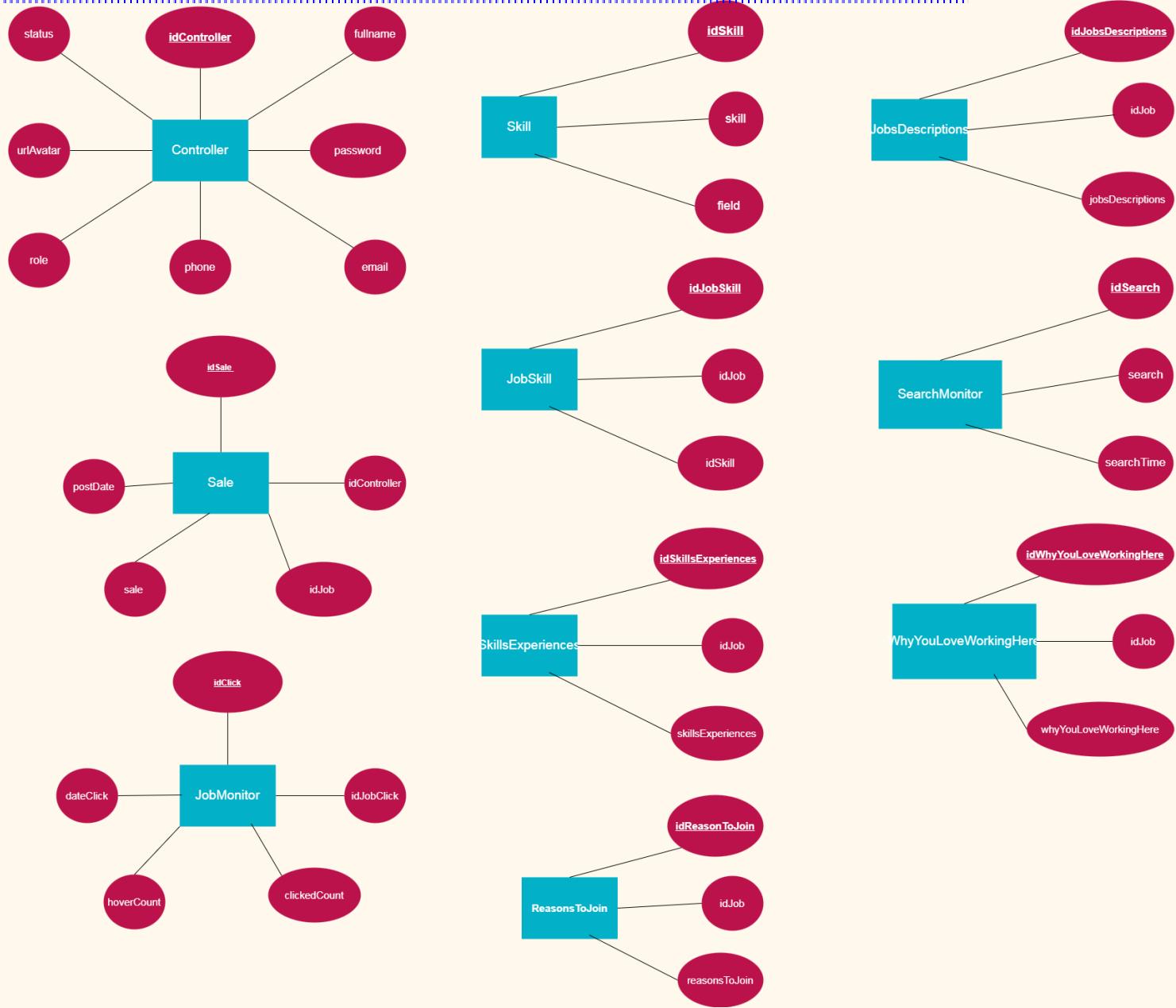


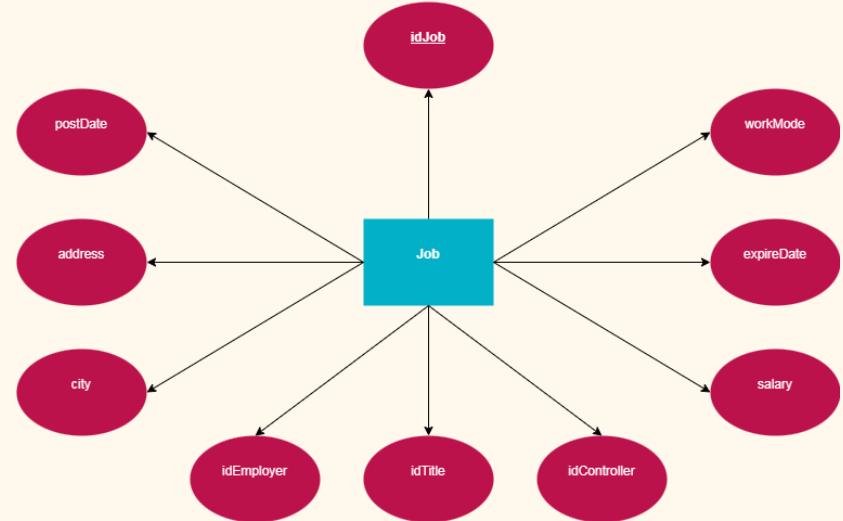
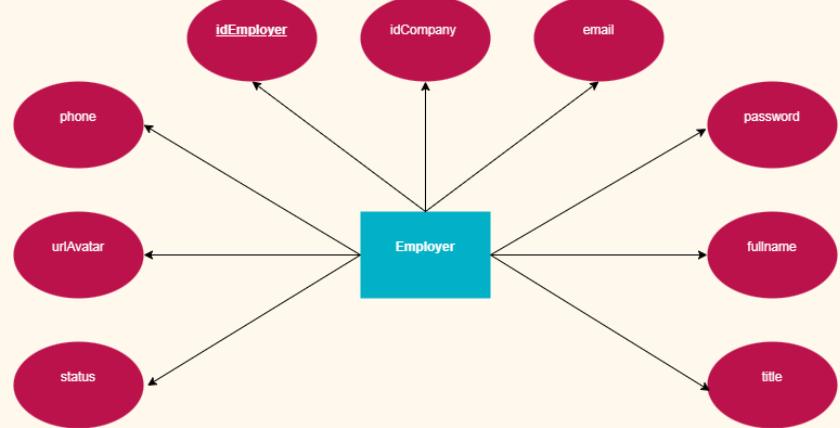
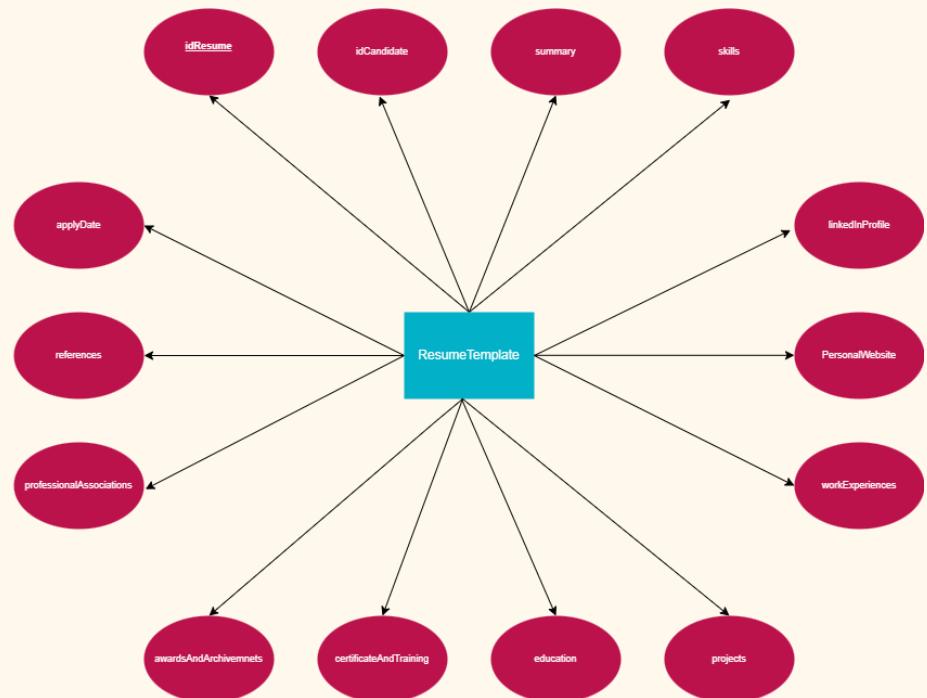
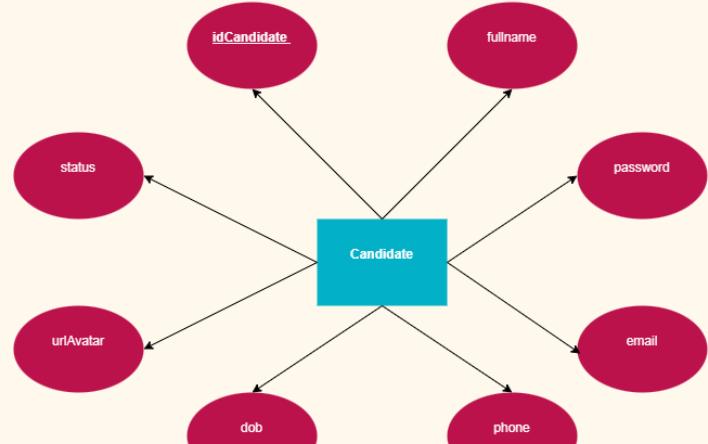
## 2.3 ENTITY RELATIONSHIP DIAGRAM (ERD)

### 2.3.1 ERD General

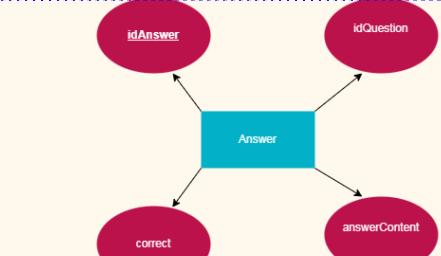


### **2.3.2 ERD Details**

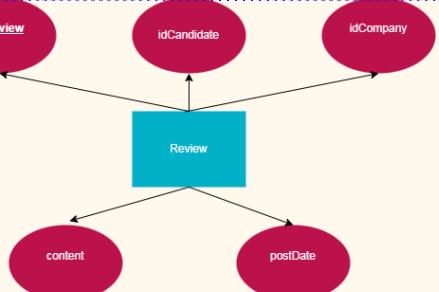




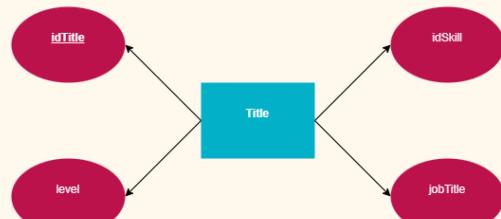
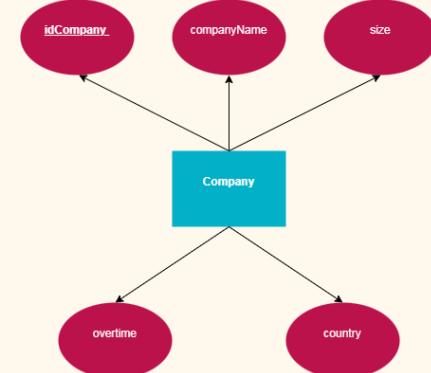
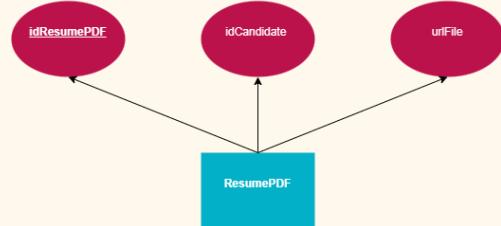
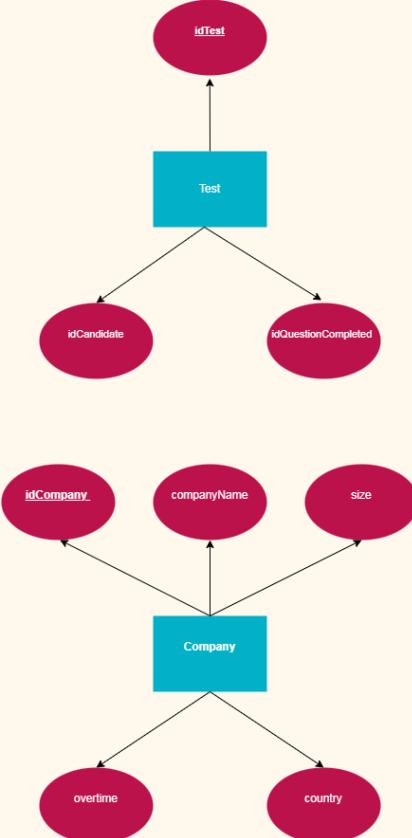
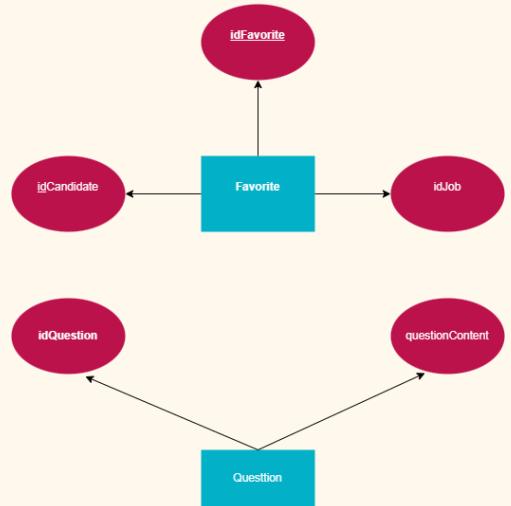
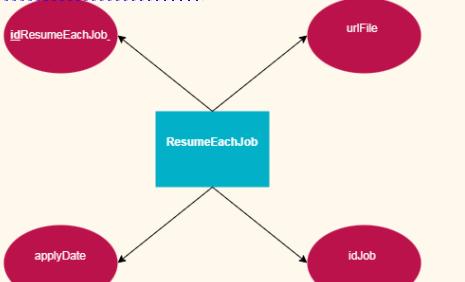
## Design plan: Job board Management System



## Document Name: Review 1+2+3

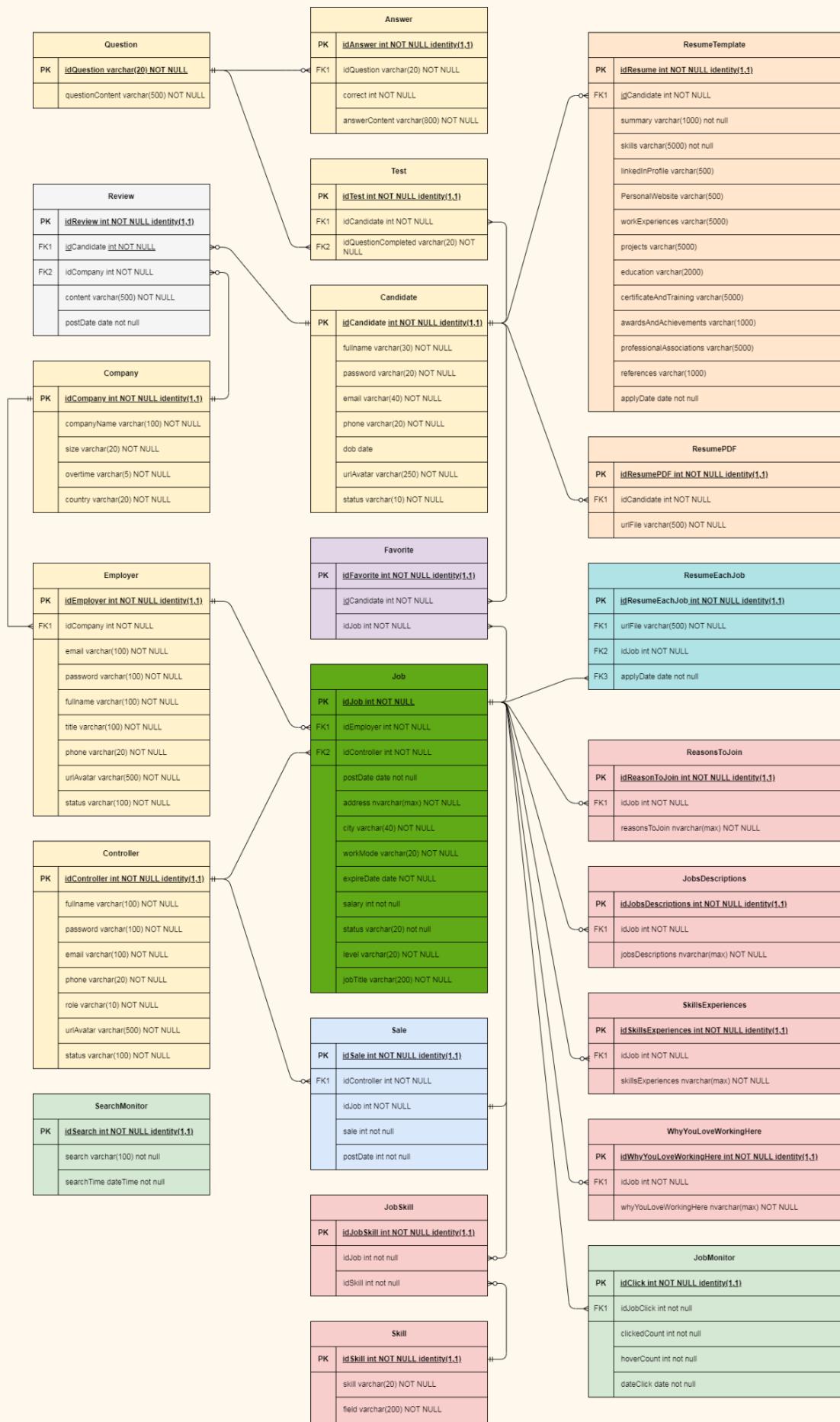


## Project group: 01



## 2.4 CLASS DIAGRAM

- There are **22 tables** in total and they are splitted into two categories below
- **Dimensional table:**
  - Question
  - Review
  - Company
  - Employer
  - Controller
  - SearchMonitor
  - Answer
  - Candidate
  - Skill
  - ResumeTemplate
  - ResumePDF
  - ReasonsToJoin
  - JobsDescription
  - SkillExperiences
  - WhyYouLoveWorkingHere
  - JobMonitor
- **Fact table:**
  - Test
  - Favourite
  - Job
  - Sale
  - JobSkill
  - ResumeEachJob



## 2.5 TASK SHEET REVIEW 2

Project Ref.No		Project Title	Activity plan prepared by	Date of preparation of Activity plan				
#	Task			Start date	End date	Actual days	Member's name	Status
1	Architecture and design	Job Board Management System	Nguyễn Anh Vũ	2023-07-01	2023-07-05	5	Nguyễn Anh Vũ	Done
2	Use case general			2023-07-01	2023-07-05	5	Nguyễn Anh Vũ	Done
3	Use case details 50%			2023-07-01	2023-07-06	6	Nguyễn Gia Thiện	Done
4	Use case details 50%			2023-07-01	2023-07-06	6	Phạm Nhật Minh	Done
5	Sequence diagram 50%			2023-07-01	2023-07-07	6	Nguyễn Gia Thiện	Done
6	Sequence diagram 50%			2023-07-01	2023-07-07	6	Phạm Nhật Minh	Done
7	Entity Relationship Diagram (ERD) general			2023-07-01	2023-07-05	5	Nguyễn Anh Vũ	Done
8	ERD details 50%			2023-07-01	2023-07-06	6	Nguyễn Gia Thiện	Done
9	ERD details 50%			2023-07-01	2023-07-06	6	Phạm Nhật Minh	Done
10	Class Diagram			2023-07-01	2023-07-05	5	Nguyễn Anh Vũ	Done
11	Task Sheet review 2			2023-07-07	2023-07-07	6	Nguyễn Anh Vũ	Done

Signature of instructor	Signature of Team Leader
Mr. Trần Phước Sinh	Mr. Nguyễn Anh Vũ

## REVIEW 3

### 3.1 INSTALL DATABASE PHYSICAL

Download SQL Server:

- Visit the Microsoft website or the SQL Server download center to get the installation package for the SQL Server version you want to install. Make sure to download the appropriate edition (Express, Standard, or Enterprise) based on your needs.

Run the Installer:

- Locate the downloaded installer and run it as an administrator.

#### Create database and table

```
create database DB4
```

```
go
```

```
use DB4
```

```
go
```

```
job
```

---

```
create table job
```

```
(
```

```
    idjob int not null,  
    idemployer int not null,  
    idcontroller int not null,  
    postdate date not null,  
    [address] nvarchar(300) not null,  
    city nvarchar(40) not null,  
    workmode nvarchar(20) not null,  
    [expiredate] date not null,  
    salary int not null,  
    [status] nvarchar(20) not null,  
    [level] nvarchar(20) not null,
```

```
    jobtitle nvarchar(50) not null,  
    primary key (idjob)  
)  
go
```

### reasonstojoin

```
create table reasonstojoin  
(  
    idreasontojoin int not null identity(1,1),  
    idjob int not null,  
    reasonstojoin nvarchar(300) not null,  
    primary key (idreasontojoin)  
)  
go
```

### jobsdescriptions

```
create table jobsdescriptions  
(  
    idjobsdescriptions int not null identity(1,1),  
    idjob int not null,  
    jobsdescriptions nvarchar(300) not null,  
    primary key (idjobsdescriptions)  
)  
go
```

### skillsexperiences

```
create table skillsexperiences  
(  
    idskillsexperiences int not null identity(1,1),  
    idjob int not null,  
    skillsexperiences nvarchar(300) not null,  
    primary key (idskillsexperiences)  
)
```

go

### whyyouloveworkinghere

```
create table whyyouloveworkinghere
(
    idwhyyouloveworkinghere int not null identity(1,1),
    idjob int not null,
    whyyouloveworkinghere nvarchar(300) not null,
    primary key (idwhyyouloveworkinghere)
)
go
```

### employer

```
create table employer
(
    idemployer int not null identity(1,1),
    idcompany int not null,
    email nvarchar(40) not null,
    [password] nvarchar(20) not null,
    fullname nvarchar(30) not null,
    title nvarchar(15) not null,
    phone nvarchar(20) not null,
    urlavatar nvarchar(250) not null,
    [status] nvarchar(10) not null,
    primary key (idemployer)
)
go
```

### skill

```
create table skill
(
```

```
    idskill int not null identity(1,1),  
    skill nvarchar(30) not null,  
    field nvarchar(30) not null,  
    primary key (idskill)  
)  
go
```

### jobskill

```
create table jobskill  
(  
    idjobskill int not null identity(1,1),  
    idjob int not null,  
    idskill int not null,  
    primary key (idjobskill)  
)  
go
```

### jobmonitor

```
create table jobmonitor  
(  
    idclick int not null identity(1,1),  
    idjobclick int not null,  
    clickedcount int not null,  
    hovercount int not null,  
    dateclick date not null,  
    primary key (idclick)  
)  
go
```

---

```
truncate table dbo.jobmonitor;  
go
```

```
-- import the file  
bulk insert dbo.jobmonitor  
from 'D:\CODE_LOCAL\PYTHON\db\jobmonitor.csv'
```

with

```
(  
    format='csv',  
    firstrow=2  
)
```

go

---

create table searchmonitor

```
(  
    idsearch int not null identity(1,1),  
    search nvarchar(100) not null,  
    searchtime datetime not null,  
    primary key (idsearch)  
)
```

go

create table sale

```
(  
    idsale int not null identity(1,1),  
    idcontroller int not null,  
    idjob int not null,  
    sale int not null,  
    postdate date not null,  
    primary key (idsale)  
)
```

go

---

truncate table dbo.sale;

go

-- import the file

```
bulk insert dbo.sale  
from 'D:\CODE_LOCAL\ PYTHON\ db\ sale.csv'  
with  
(  
    format='csv',  
    firstrow=2
```

```
)  
go
```

```
create table controller
```

```
(  
    idcontroller int not null identity(1,1),  
    fullname nvarchar(50) not null,  
    [password] nvarchar(200) not null,  
    email nvarchar(50) not null,  
    phone nvarchar(15) not null,  
    [role] nvarchar(10) not null,  
    urlavatar nvarchar(250) not null,  
    [status] nvarchar(15) not null,  
    primary key (idcontroller)
```

```
)  
go
```

```
truncate table dbo.controller;
```

```
go
```

```
-- import the file
```

```
bulk insert dbo.controller  
from 'D:\CODE_LOCAL\ PYTHON\ db\ controller.csv'  
with  
(  
    format='csv',  
    firstrow=2  
)  
go
```

```
create table favorite
```

```
(  
    idfavorite int not null identity(1,1),  
    idcandidate int not null,  
    idjob int not null,  
    primary key (idfavorite)  
)
```

go

---

```
truncate table dbo.favorite;
```

```
go
```

```
-- import the file
```

```
bulk insert dbo.favorite  
from 'D:\CODE_LOCAL\ PYTHON\ db\ favorite.csv'
```

```
with
```

```
(  
    format='csv',  
    firstrow=2
```

```
)
```

```
go
```

---

```
create table resumeeachjob
```

```
(  
    idresumeeachjob int not null identity(1,1),  
    idcandidate int not null,  
    urlfile nvarchar(500) not null, -- nôp link file pdf đã post lên firebase  
    idjob int not null,  
    applydate date not null,  
    primary key (idresumeeachjob)
```

```
)
```

```
go
```

```
create table company
```

```
(  
    idcompany int not null identity(1,1),  
    companyname nvarchar(100) not null,  
    size nvarchar(20) not null,  
    overtime nvarchar(5) not null,  
    country nvarchar(20) not null,  
    primary key (idcompany)
```

```
)
```

```
go
```

---

```
truncate table dbo.company;
```

```
go
```

```
-- import the file
bulk insert dbo.company
from 'D:\CODE_LOCAL\PYTHON\db\company.csv'
with
(
    format='csv',
    firstrow=2
)
go
```

---

### review

```
create table review
(
    id int not null identity(1,1),
    idgroup int not null,
    idreview int not null,
    idcandidate bool not null,
    idcontroller int not null,
    idemployer int not null,
    idcompany int not null ,
    content nvarchar(500) not null,
    postdate datetime not null,
    primary key (id)
)
go
```

### candidate

```
create table candidate
(
    idcandidate int not null identity(1,1),
    fullname nvarchar(30) not null,
```

```
email nvarchar(40) not null,  
[password] nvarchar(20) not null,  
phone nvarchar(20) not null,  
dob date not null,  
urlavatar nvarchar(250) not null,  
[status] nvarchar(10) not null,  
primary key (idcandidate)  
)  
go
```

### answer

```
create table answer  
(  
    idanswer int not null identity(1,1),  
    idquestion nvarchar(20) not null,  
    correct int not null,  
    answercontent nvarchar(800) not null,  
    primary key (idanswer)  
)  
go
```

### question

```
create table question  
(  
    idquestion nvarchar(20) not null,  
    questioncontent nvarchar(500) not null,  
    primary key (idquestion)  
)  
go
```

### test

```
create table test  
(
```

```
    idtest int not null identity(1,1),
    idcandidate int not null,
    idquestioncompleted nvarchar(20) not null,
    primary key (idtest)
)
go
```

### resumetemplate

```
create table resumetemplate
(
    idresume int not null identity(1,1),
    idcandidate int not null,
    fullname nvarchar(40) not null,
    [address] nvarchar(500) not null,
    dob date,
    email nvarchar(70) not null,
    phone nvarchar(20) not null,
    summary nvarchar(3500) not null,
    skills1 nvarchar(50) not null,
    skills2 nvarchar(50),
    skills3 nvarchar(50),
    skills4 nvarchar(50),
    skills5 nvarchar(50),
    workexperiences1 nvarchar(3500) not null,
    workexperiences2 nvarchar(3500),
    workexperiences3 nvarchar(3500),
    workexperiences4 nvarchar(3500),
    workexperiences5 nvarchar(3500),
    workexperiencestitle1 nvarchar(200) not null,
    workexperiencestitle2 nvarchar(200),
    workexperiencestitle3 nvarchar(200),
    workexperiencestitle4 nvarchar(200),
    workexperiencestitle5 nvarchar(200),
    education1 nvarchar(3500) not null,
    education2 nvarchar(3500),
    education3 nvarchar(3500),
```

```
education4 nvarchar(3500),  
education5 nvarchar(3500),  
certificateandtraining nvarchar(3500),  
urlavatar nvarchar(350) not null,  
applydate date not null,  
primary key (idresume)  
)  
go
```

### resumepdf

```
create table resumepdf  
(  
    idresumepdf int not null identity(1,1),  
    idcandidate int not null,  
    urlfile nvarchar(500) not null,  
    primary key (idresumepdf)  
)  
go
```

### Create relationship between tables

```
alter table job  
add  
constraint fk1 foreign key(idemployer) references employer(idemployer),  
constraint fk3 foreign key(idcontroller) references controller(idcontroller)  
go
```

```
alter table reasonstojoin
```

add

```
constraint fk4 foreign key(idjob) references job(idjob)
```

go

alter table jobsdescriptions

add

```
constraint fk5 foreign key(idjob) references job(idjob)
```

go

alter table skillsexperiences

add

```
constraint fk6 foreign key(idjob) references job(idjob)
```

go

alter table whyyouloveworkinghere

add

```
constraint fk7 foreign key(idjob) references job(idjob)
```

go

```
alter table jobskill
```

```
add constraint fk8 foreign key(idskill) references skill(idskill)
```

```
go
```

```
alter table jobskill
```

```
add constraint fk9 foreign key(idjob) references job(idjob)
```

```
go
```

```
alter table jobmonitor
```

```
add constraint fk10 foreign key(idjobclick) references job(idjob)
```

```
go
```

```
alter table sale
```

```
add constraint fk11 foreign key(idcontroller) references controller(idcontroller)
```

```
go
```

```
alter table favorite
```

```
add constraint fk12 foreign key(idjob) references job(idjob)
```

go

alter table resumeeachjob

add constraint fk13 foreign key(idjob) references job(idjob)

go

alter table employer

add constraint fk14 foreign key(idcompany) references company(idcompany)

go

alter table review

add constraint fk15 foreign key(idcompany) references company(idcompany)

go

alter table review

add constraint fk15\_1 foreign key(idcandidate) references candidate(idcandidate)

go

alter table favorite

add constraint fk16 foreign key(idcandidate) references candidate(idcandidate)

go

alter table test

add constraint fk17 foreign key(idquestioncompleted) references question(idquestion)

go

alter table answer

add constraint fk18 foreign key(idquestion) references question(idquestion)

go

alter table resumetemplate

add constraint fk19 foreign key(idcandidate) references candidate(idcandidate)

go

alter table resumepdf

add constraint fk20 foreign key(idcandidate) references candidate(idcandidate)

go

select \* from job

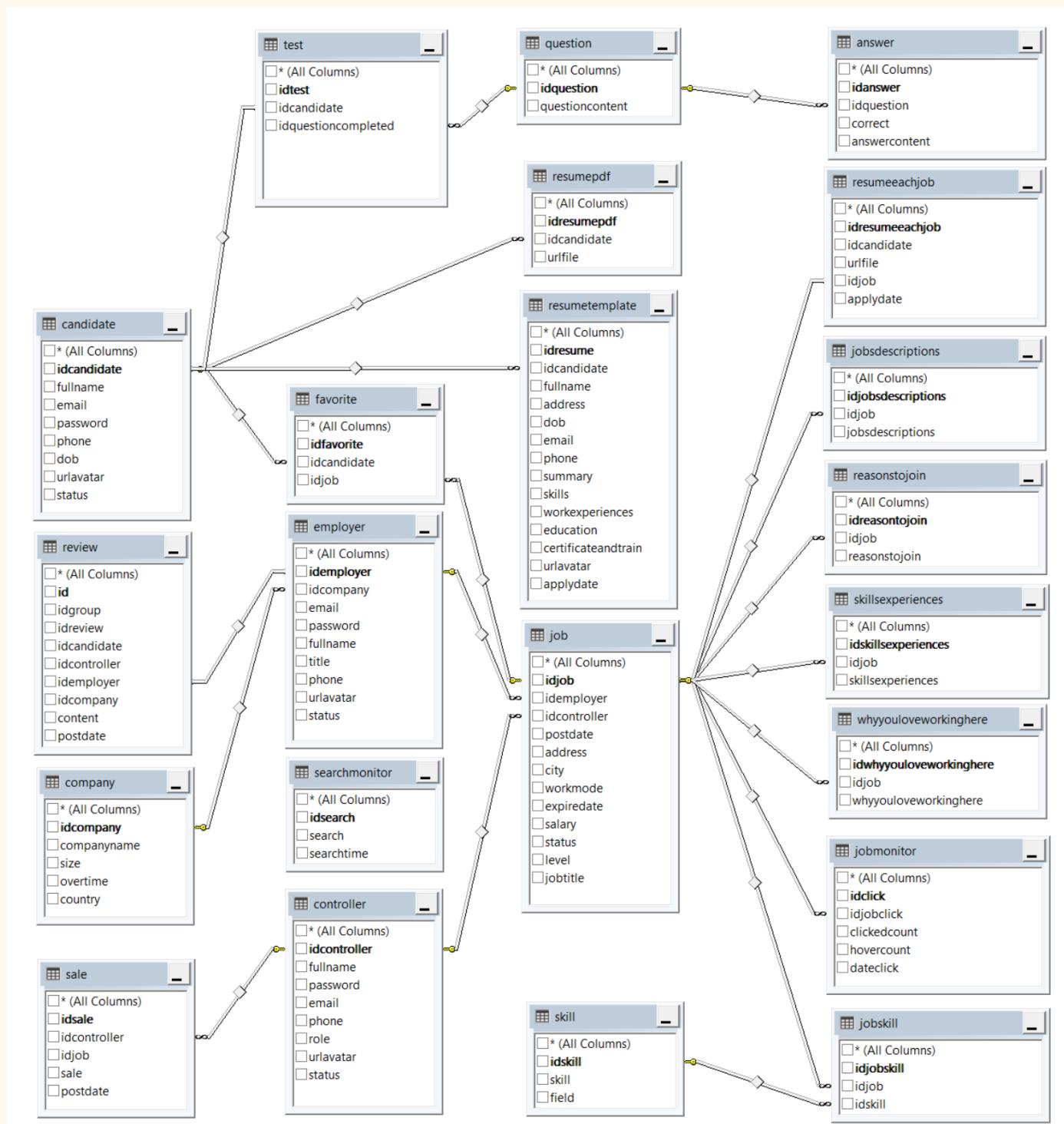
go

select \* from employer

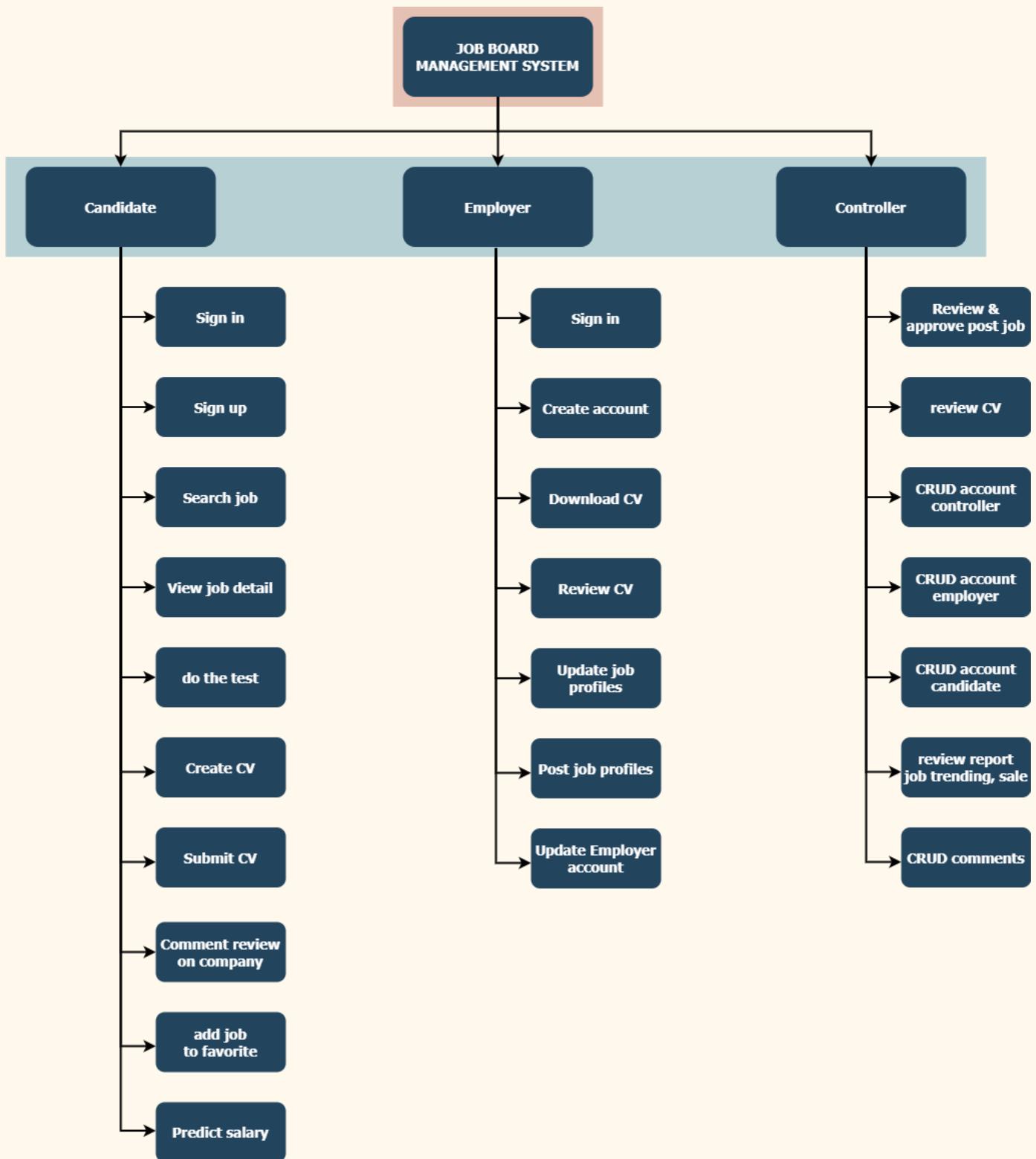
go



## DATABASE TABLE &amp; RELATIONSHIP DESIGN

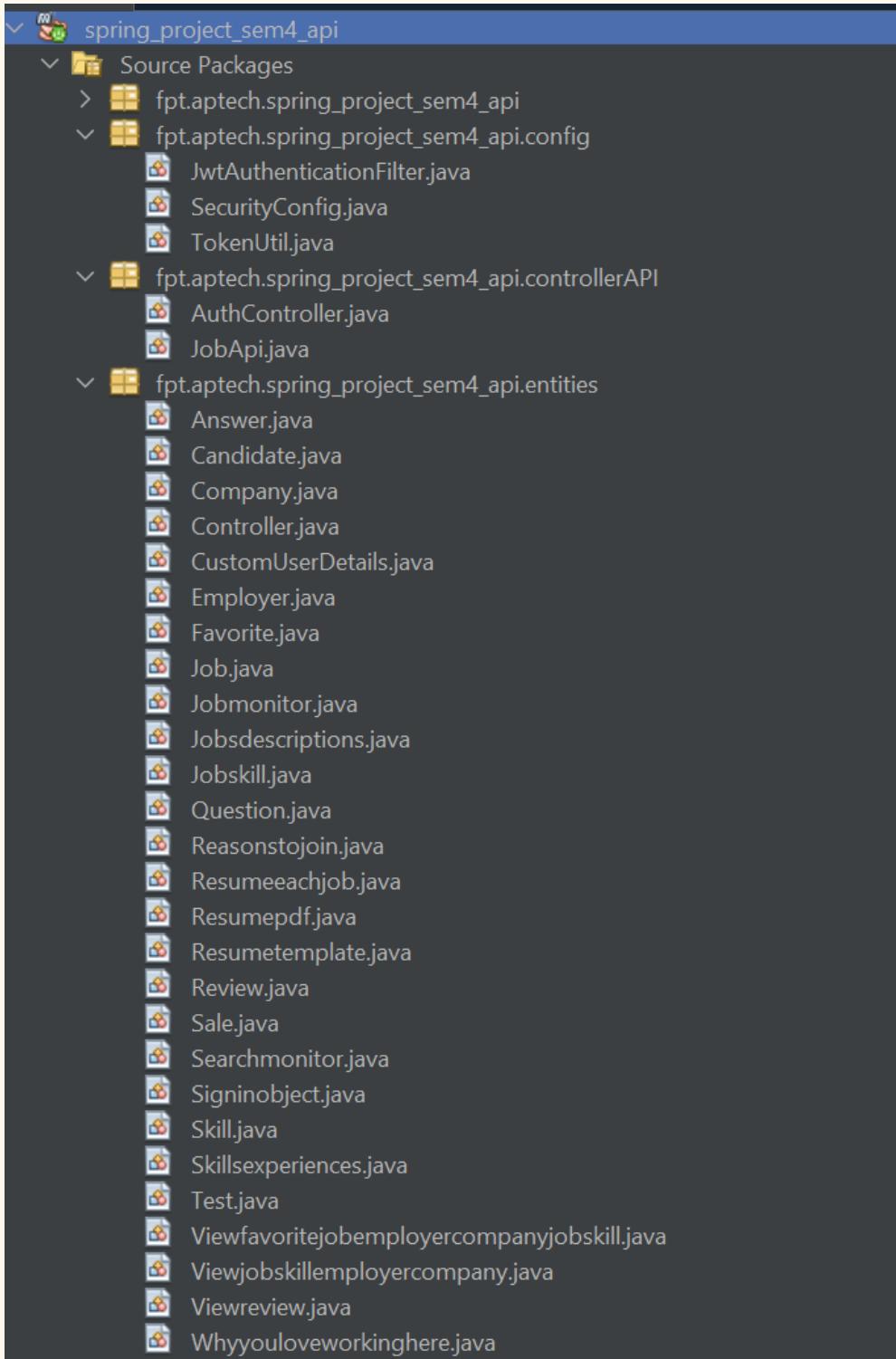


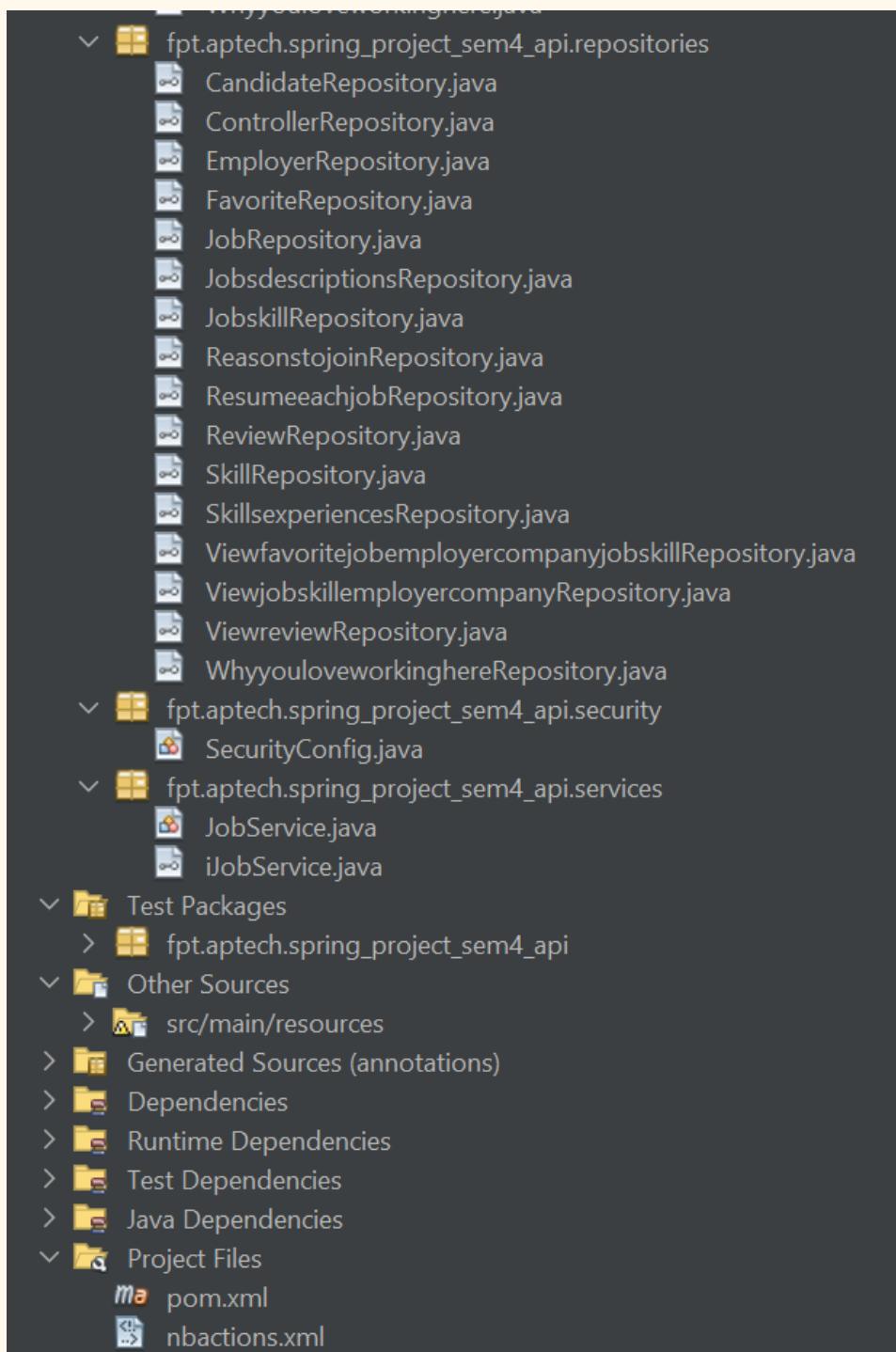
## 3.2 SITE MAP



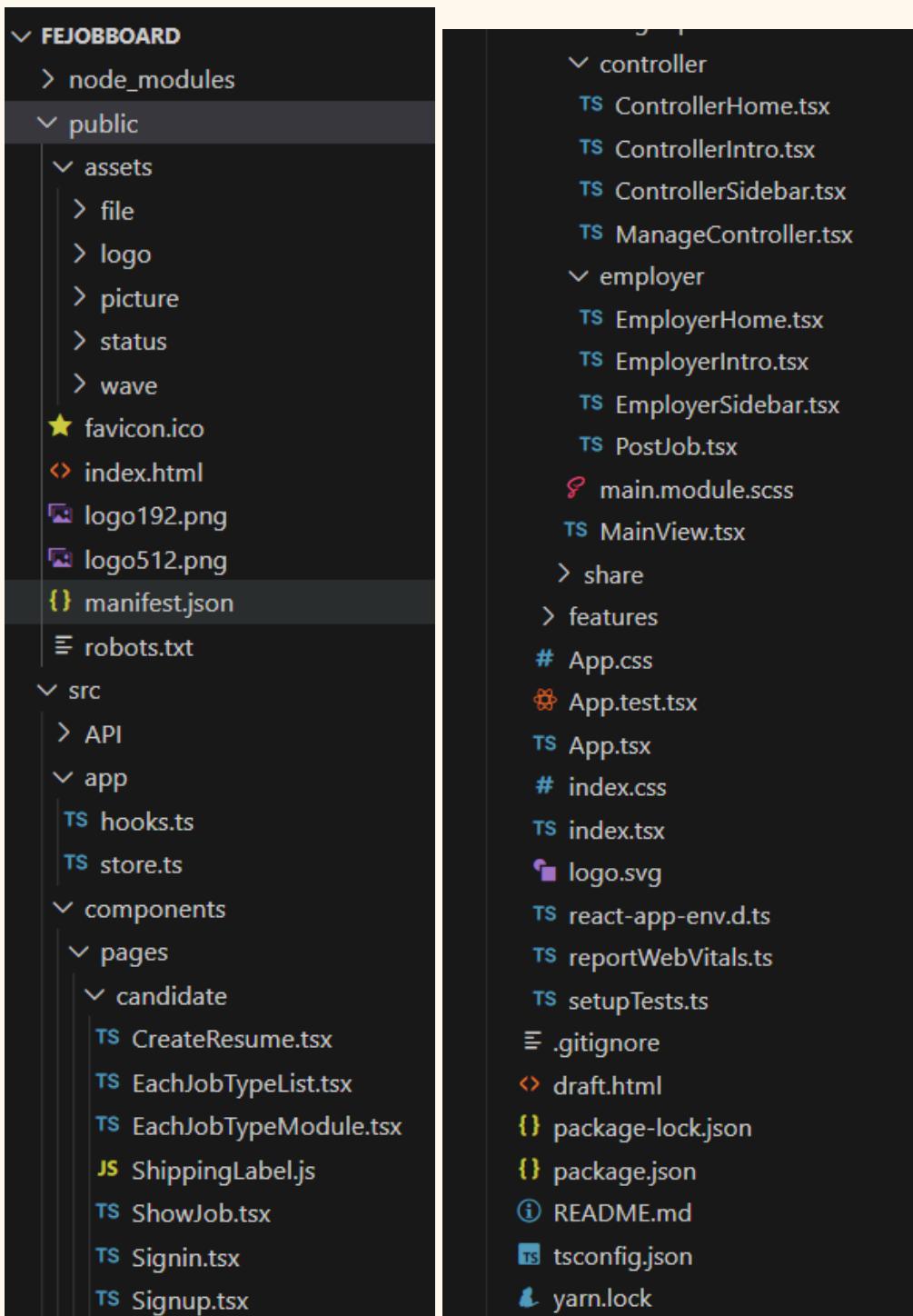
### 3.3 CONSTRUCTOR PROJECT DIRECTORY

- Back end

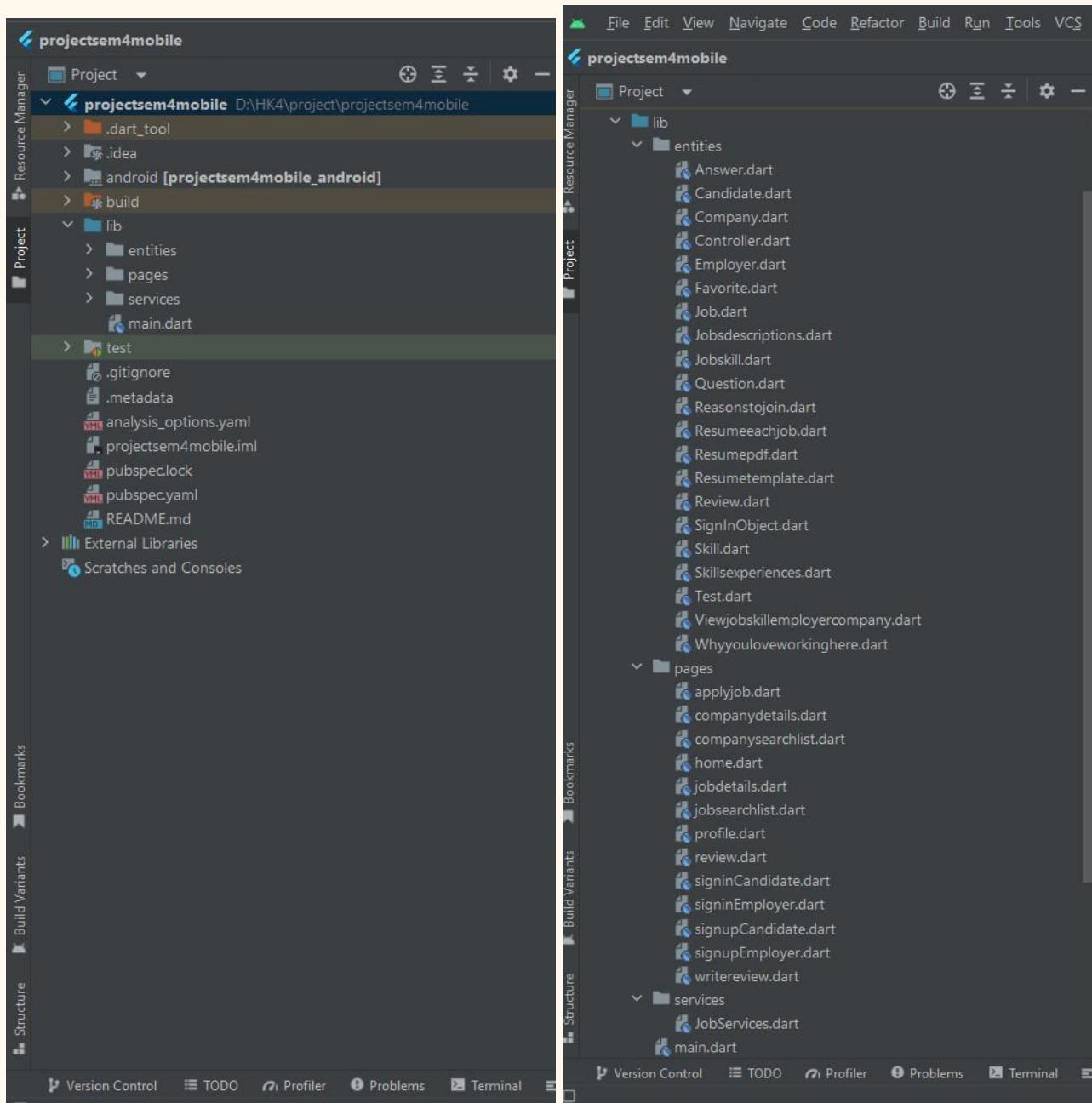




## - Front end React JS



## Front end Fluter



## 3.4 GUI DESIGN

### 3.4.1 DESIGN INTERFACE (FIGMA)

#### WEB

For Candidate

#### Search job

LOGO

All Jobs

Test

CV Templates

Salary Prediction

Help

SIGN IN

City

Level

Country

Salary range

Filter

Search

Total Jobs: 1000

FAVORITE

Job Title: Frontend Developer  
Company Name: FPT Software  
Salary: 500  
Work Mode: hybrid  
City: Ho Chi Minh  
Skill: javascript ReactJS

View Add to Favorite

Job Title: Frontend Developer  
Company Name: FPT Software  
Salary: 500  
Work Mode: hybrid  
City: Ho Chi Minh  
Skill: javascript ReactJS

View Add to Favorite

Job Title: Frontend Developer  
Company Name: FPT Software  
Salary: 500  
Work Mode: hybrid  
City: Ho Chi Minh  
Skill: javascript ReactJS

View Add to Favorite

Job Title: Frontend Developer  
Company Name: FPT Software  
Salary: 500  
Work Mode: hybrid  
City: Ho Chi Minh  
Skill: javascript ReactJS

View Add to Favorite

Job Title: Frontend Developer  
Company Name: FPT Software  
Salary: 500  
Work Mode: hybrid  
City: Ho Chi Minh  
Skill: javascript ReactJS

View Add to Favorite

Job Title: Frontend Developer  
Company Name: FPT Software  
Salary: 500  
Work Mode: hybrid  
City: Ho Chi Minh  
Skill: javascript ReactJS

View Add to Favorite

Job Title: Frontend Developer  
Company Name: FPT Software  
Salary: 500  
Work Mode: hybrid  
City: Ho Chi Minh  
Skill: javascript ReactJS

View Add to Favorite

Job Title: Frontend Developer  
Company Name: FPT Software  
Salary: 500  
Work Mode: hybrid  
City: Ho Chi Minh  
Skill: javascript ReactJS

View Add to Favorite

Job Title: Frontend Developer  
Company Name: FPT Software  
Salary: 500  
Work Mode: hybrid  
City: Ho Chi Minh  
Skill: javascript ReactJS

View Add to Favorite

Pagination

footer

## View job details

# JOB DETAILS

**address:** 56 Yen The- Ward 2- Tan Binh  
**city:** Ho Chi Minh  
**companyname:** Smartee Vina  
**country:** South Korea  
**expiredate:** 2023-10-30  
**field:** Front-end  
**jobtitle:** Frontend Developer (ReactJS) Trainee  
**level:** Fresher  
**overtime:** yes  
**postdate:** 2023-08-01  
**salary:** \$ 460  
**size:** 1000+ persons  
**skill:** ReactJSjavascript

[Submit CV](#) [Cancel](#)

## Comments

# COMMENTS

Aiden Nelson  
lack of training for employees  
2023-07-13T17:00:00.000+00:00 

Please enter your comment here... [Send](#)

## Create CV

**Step 1:** Please click on file to download sample excel file [Download sample](#)

**Step 2:** Please fill in your info in the column B of the excel file you just downloaded

**Step 3:** After completion of step 2, please upload your excel file here  
 Upload your picture avatar here  No file chosen  No file chosen

**Step 4:** Check below preview CV in different template, if you want to change any info, kindly redo from step 2. Otherwise you can download it now. [Download your CV](#)

Your CV would look the same below:

[TEMPLATE 1](#) [TEMPLATE 2](#) [TEMPLATE 3](#)



**General info**

**Name**  
Jack Sparrow

**Address**  
86/38 Ong Ich Khiem st,  
District 11, HCM city

**Phone**  
69544645

**Email**  
asdjfjhbjhv@gmail.com

**Birth**

**Apply**

**Position**  
Fullstack Developer

**Company**  
ABC Software

**Summary**  
akshjfg alieuhf oeu owe oiywoeigwoeig oweigwei gowe gwoe woeigwo eowigwo; eig gwo;eigw;oi eo;wgwo;eig wo eg

**Skill**

Reactjs	Java	.NET
Good	Good	Good

**Education**

## predict salary

**PREDICT SALARY**

Title:	<input type="text" value="data engineer..."/>	City:	<input type="text" value="Ho Chi Minh"/>
Level:	<input type="text" value="Fresher"/>	Work mode:	<input type="text" value="office"/>
Skill:	<input type="text" value="python..."/>	Size:	<input type="text" value="50"/>
Field:	<input type="text" value="Data"/>	Country:	<input type="text" value="Viet Nam"/>

**Your input information are:**

Title:	City:
Level:	Work Mode:
Skill:	Size:
Field:	Country:

**Your salary prediction based on your input information is:**

**\$ 0.00**

## Sign in

**SIGN IN**

Email:	<input type="text" value="ethan.adams@google.com"/>
Password:	<input type="password" value="..."/>

Does not have an account ? Click here to [sign Up](#)

**sign up**

## SIGN UP

**Full name:** Nguyen Anh Vu

**Email:** vuna141@google.com

**Password:** **...**

Password must have minimum 8 characters, maximum 20, including 1 uppercase, 1 lowercase, 1 number, 1 special character and no space between.  
Example: sdfjhLKhl349(\*&)

**Password confirm:**

**Phone:** 0982547769

**Date of Birth:** mm / dd / yyyy

**Avatar:**  Choose File No file chosen

**Submit** **Reset**

Already have an account ? Click here to [sign In](#)

## Test

LOGO	All Jobs	Test	CV Templates	Salary Prediction	Help	SIGN IN
<b>TEST</b>						
<b>Questions Collection</b>	<b>Topic: React JS</b> <b>Questions complete: 1/17</b> <a href="#">Previous</a> <a href="#">Next</a>					
React JS	<b>Questions:</b> What are the main features of Java ?					
Java						
C#	<b>Choose correct Answers:</b>					
algorithm & structure	<ul style="list-style-type: none"> <li><input checked="" type="radio"/> The main features of Java include platform independence object-oriented programming automatic memory management (garbage collection) multithreading and exception handling.</li> <li><input type="radio"/> The main features of Java include platform dependence object-oriented programming automatic memory management (garbage collection) multithreading and exception handling.</li> <li><input type="radio"/> The main features of Java include platform free object-oriented programming automatic memory management (garbage collection) multithreading and exception handling.</li> <li><input type="radio"/> The main features of Java include platform greate object-oriented programming automatic memory management (garbage collection) multithreading and exception handling.</li> </ul>					
<b>Summary</b>						
React JS	5/17					
Java	0/20					
C#	5/20					
algorithm & structure	8/20					

## For Employer

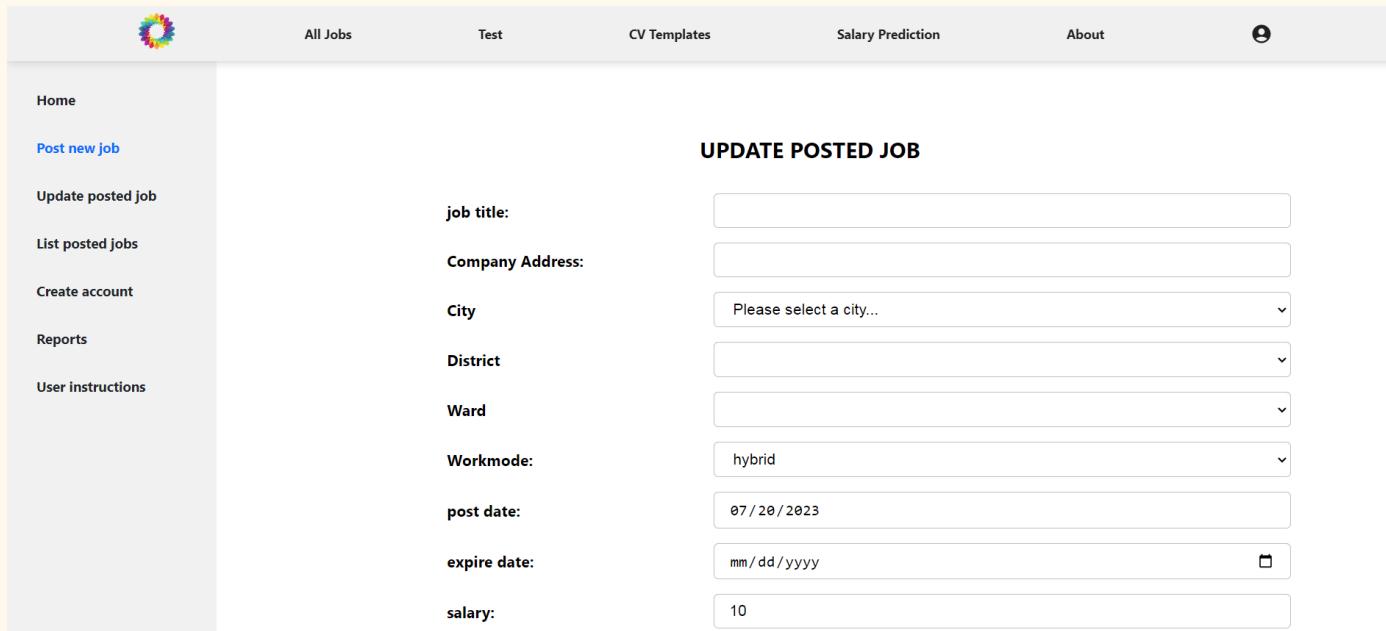
## Home

The dashboard features a central title "EMPLOYER DASHBOARD" above a 2x3 grid of rounded rectangular buttons. The buttons are labeled: "Home", "Post new job", "Update posted job" in the top row; and "List posted jobs", "Create account", "Reports" in the bottom row.

## Post new job

The form is titled "Post new Job". It includes fields for "job title", "Company Address", "City" (with a dropdown placeholder "Please select a city..."), "District" (dropdown), "Ward" (dropdown), "Workmode" (dropdown with "hybrid" selected), "post date" (text input with "07/20/2023"), "expire date" (text input with "mm/dd/yyyy" and a calendar icon), and "salary" (text input with "10").

## update posted job



The screenshot shows a web-based job posting system. On the left is a sidebar with links: Home, Post new job, Update posted job (which is selected and highlighted in blue), List posted jobs, Create account, Reports, and User instructions. The main content area has a title "UPDATE POSTED JOB". It contains several input fields: "job title:" with a text input box; "Company Address:" with a text input box; "City" with a dropdown menu showing "Please select a city..."; "District" with a dropdown menu; "Ward" with a dropdown menu; "Workmode:" with a dropdown menu showing "hybrid"; "post date:" with a text input box containing "07/20/2023"; "expire date:" with a text input box containing "mm/dd/yyyy" and a calendar icon; and "salary:" with a text input box containing "10".

## List posted job

#	jobtitle	companyname	workmode	city	postdate	expiredate	salary	action
1	Frontend Developer	FPT Software	hybrid	Ho Chi Minh	2023-06-21	2023-09-19	500	<button>View</button>
2	Frontend Developer	AhaMove	office	Ho Chi Minh	2023-06-09	2023-09-07	420	<button>View</button>
3	JavaScript Developer	CoderSchool	office	Ho Chi Minh	2023-07-06	2023-10-04	1330	<button>View</button>
4	JavaScript Developer Trainee	CMC Corporation	office	Ha Noi	2023-06-23	2023-09-21	500	<button>View</button>
5	Frontend Developer (ReactJS) Trainee	Sparxo	office	Ha Noi	2023-06-16	2023-09-14	420	<button>View</button>
6	ReactJS Software Engineering Intern	Daoukiwoom Innovation	office	Ho Chi Minh	2023-06-22	2023-09-20	100	<button>View</button>

## create account

## SIGN UP

**Full name:** Nguyen Anh Vu

**Email:** vuna141@google.com

**Password:**  ...

Password must have minimum 8 characters, maximum 20, including 1 uppercase, 1 lowercase, 1 number, 1 special character and no space between.  
Example: sdfjhLKhl349(\*&)

**Password confirm:**

**Phone:** 0982547769

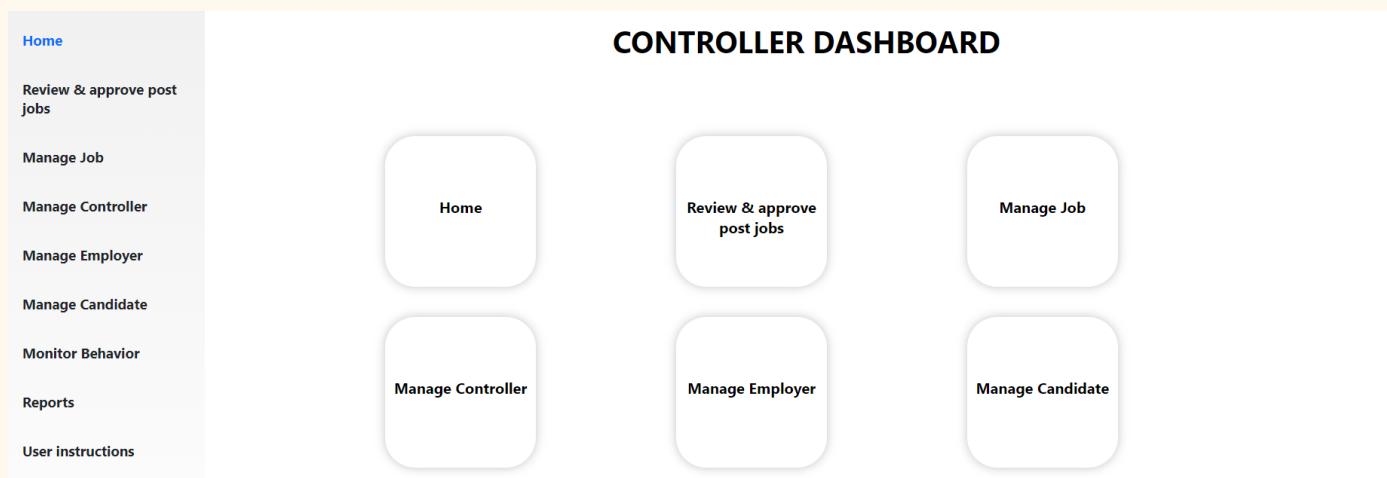
**Date of Birth:**  mm / dd / yyyy

**Avatar:**  Choose File No file chosen

Already have an account ? Click here to [sign In](#)

## For Controller

## Home



## Manage controller manage employer, manage candidate

The diagram shows a 'MANAGE CONTROLLER' table interface. On the left is a sidebar with the following menu items:

- Home
- Review & approve post jobs
- Manage Job
- Manage Controller**
- Manage Employer
- Manage Candidate
- Monitor Behavior
- Reports
- User instructions

The top navigation bar includes links for All Jobs, Test, CV Templates, Salary Prediction, About, and a user icon.

The main table has the following columns:

Name	Email	Role	Picture	Status	Action
Nguyen Anh Vu	nguyenganhv@gmail.com	admin		active	Disable Edit View
Nguyen Gia Thien	nguyengiaithien@gmail.com	admin		active	Disable Edit View
Pham Nhat Minh	phamnhatminh@gmail.com	admin		active	Disable Edit View
Steve Jobs	stevejobs@gmail.com	admin		active	Disable Edit View
Tim Cook	timcook@gmail.com	admin		active	Disable Edit View
Jose Mourinho	mourinho@gmail.com	admin		active	Disable Edit View
Pep Guardiola	guardiola@gmail.com	admin		active	Disable Edit View
klopp jurgen	klop@gmail.com	staff		active	Disable Edit View
Leonel Messi	messi@gmail.com	staff		active	Disable Edit View

## Monitor behavior

The screenshot shows a user interface for a job board management system. The top navigation bar includes links for LOGO, All Jobs, Test, CV Templates, Salary Prediction, Help, and SIGN IN. The main content area is titled "Monitor behavior". On the left, there's a sidebar with various menu items. The main content area features a table with four columns: No., Button, Location, and Count. The table has 15 rows, with only the first row containing data.

No.	Button	Location	Count
1	Search	page All Jobs	100
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...
...	...	.....	...

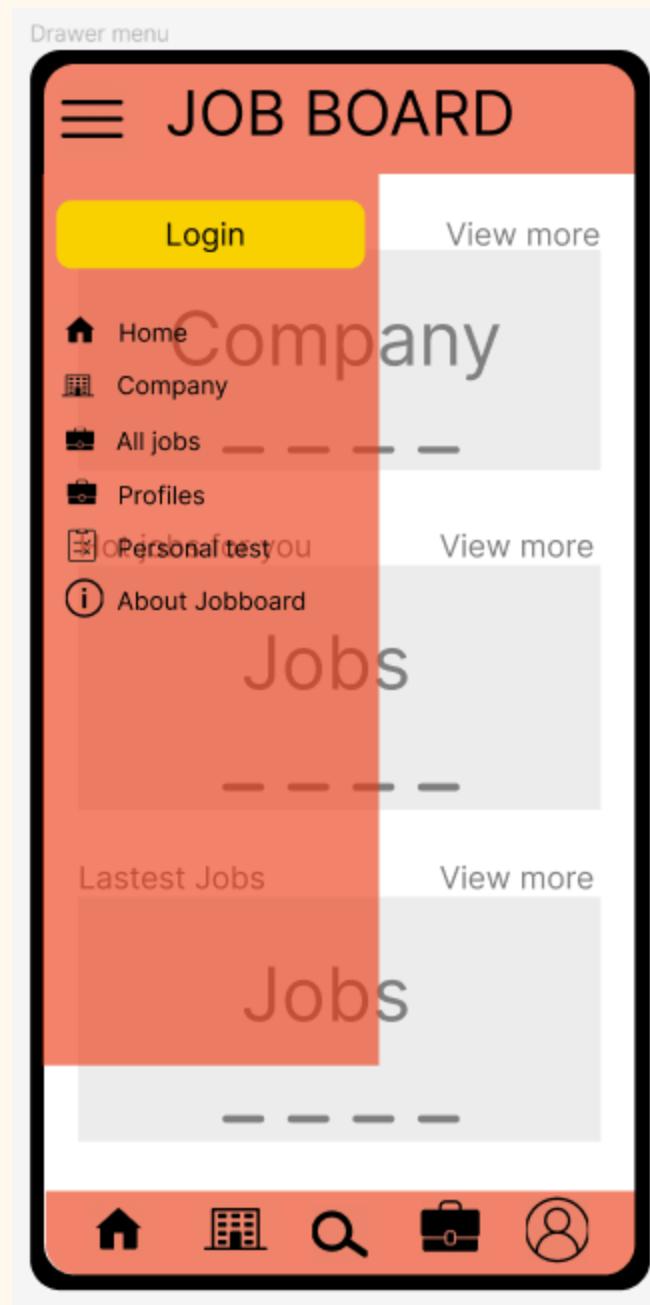
## review & approve job

The interface is a web-based application for managing job posts. It features a top navigation bar with links for LOGO, All Jobs, Test, CV Templates, Salary Prediction, Help, and SIGN IN. On the left, a sidebar lists various management functions: Home, Review & approve post jobs (selected), Manage Job, Manage Controller, Manage Employer, Manage Candidate, Monitor Behavior, Reports, and User instructions. The main content area is titled "Review & approve post job". It contains a heading and a table with 12 rows. Each row has a "Job info" column and a "Status" column with three buttons: approved, pending, and expired.

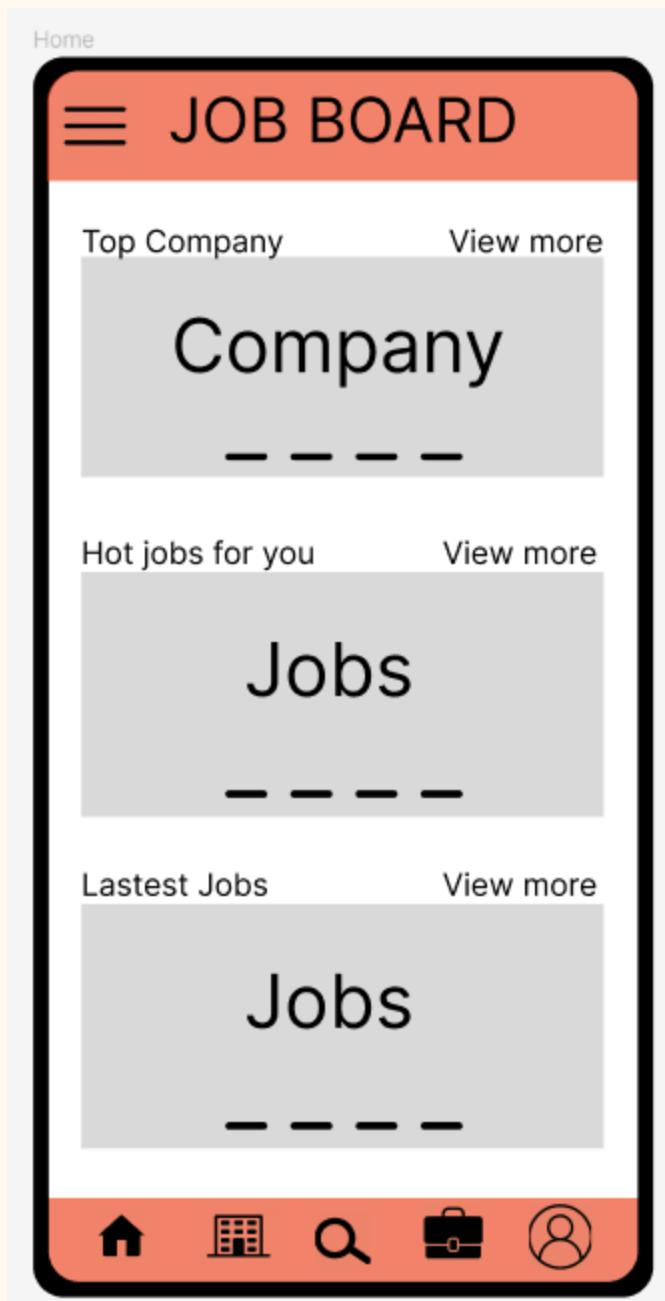
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired
Job info	approved	pending	expired

## MOBILE

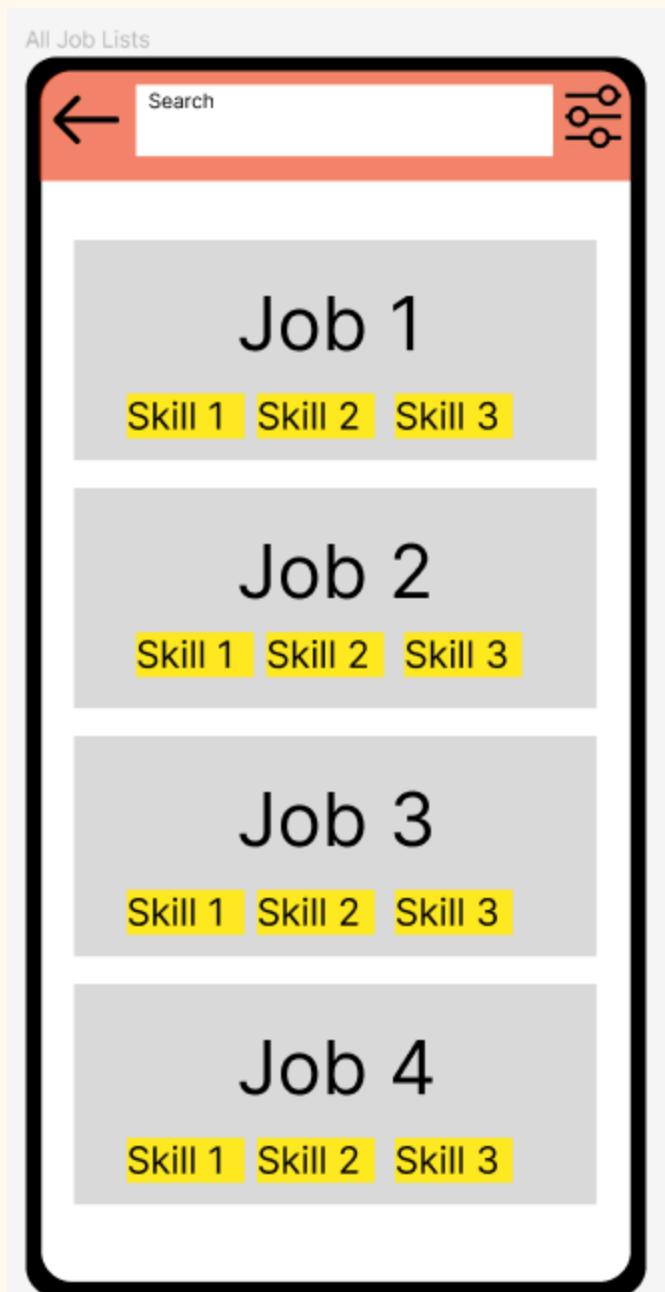
### menubar



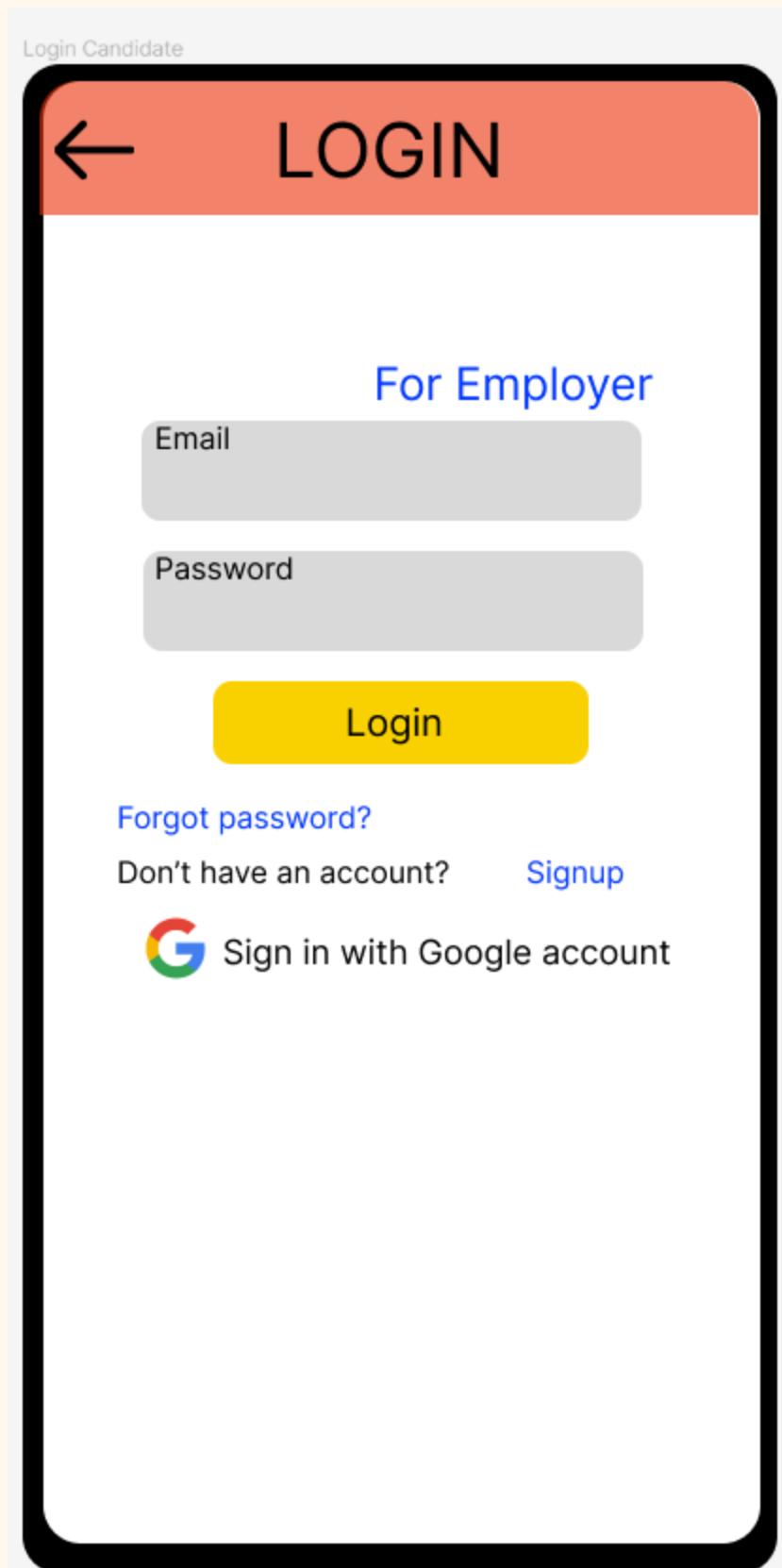
## Home



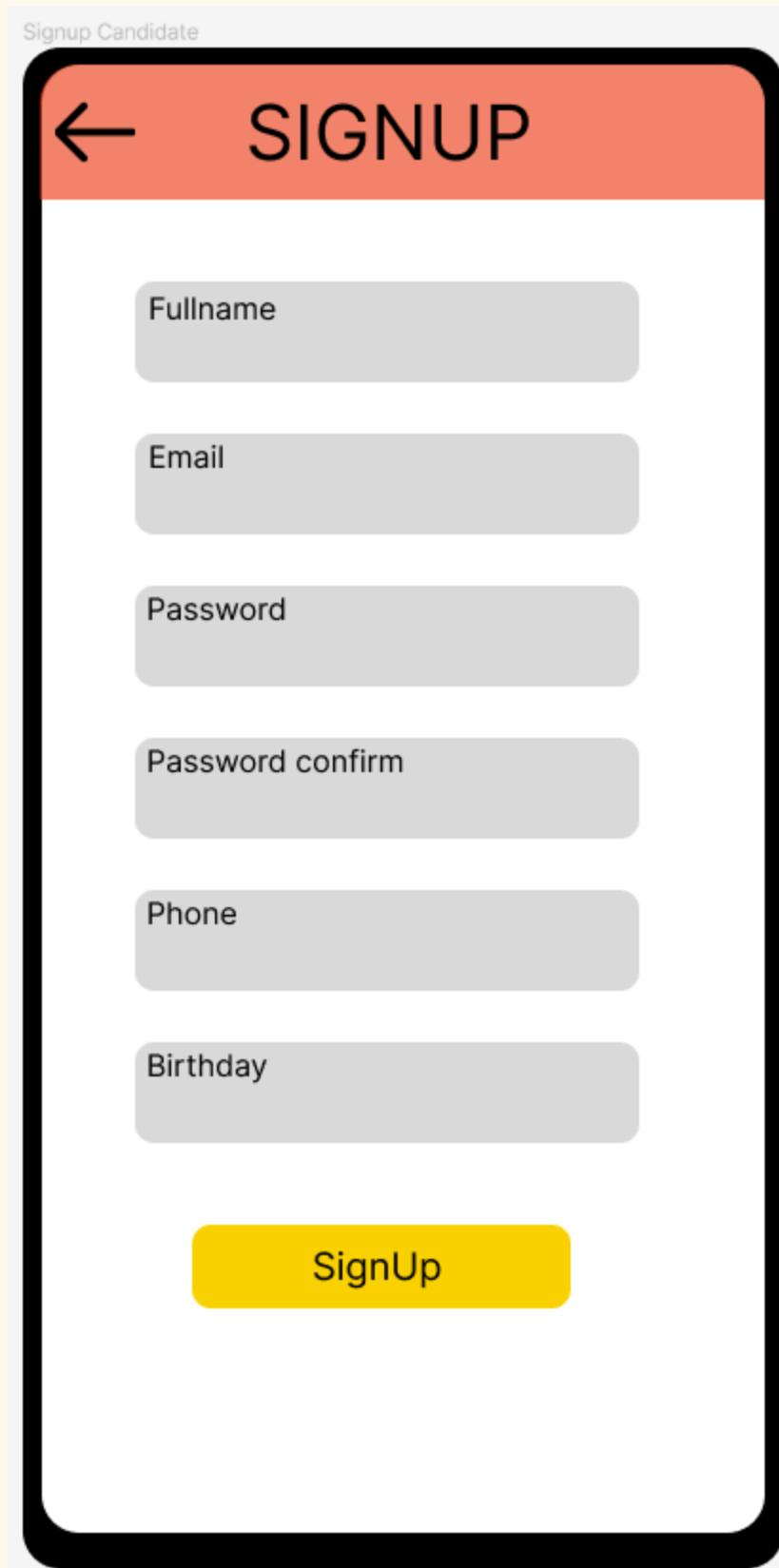
## All job lists



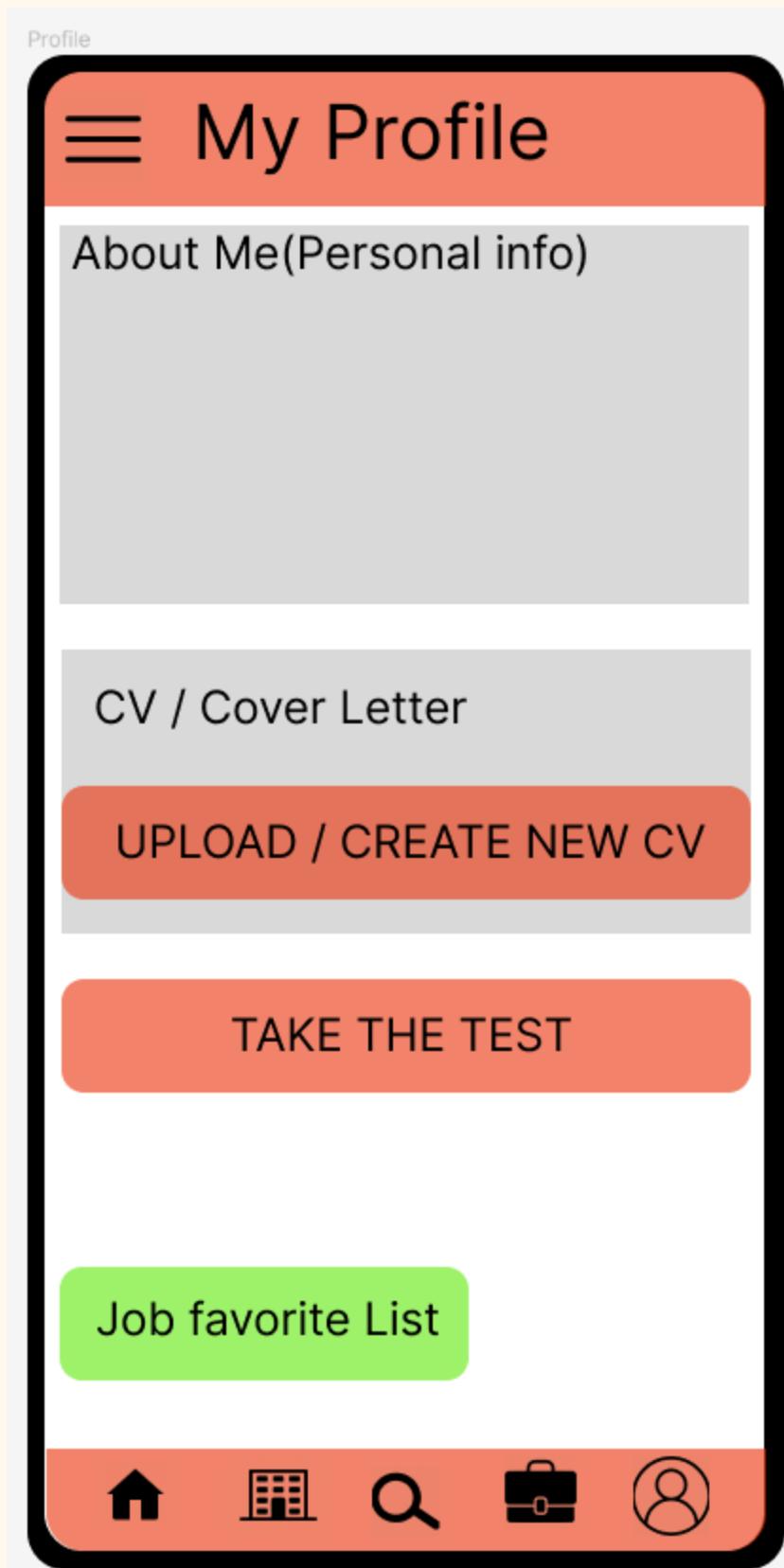
## login



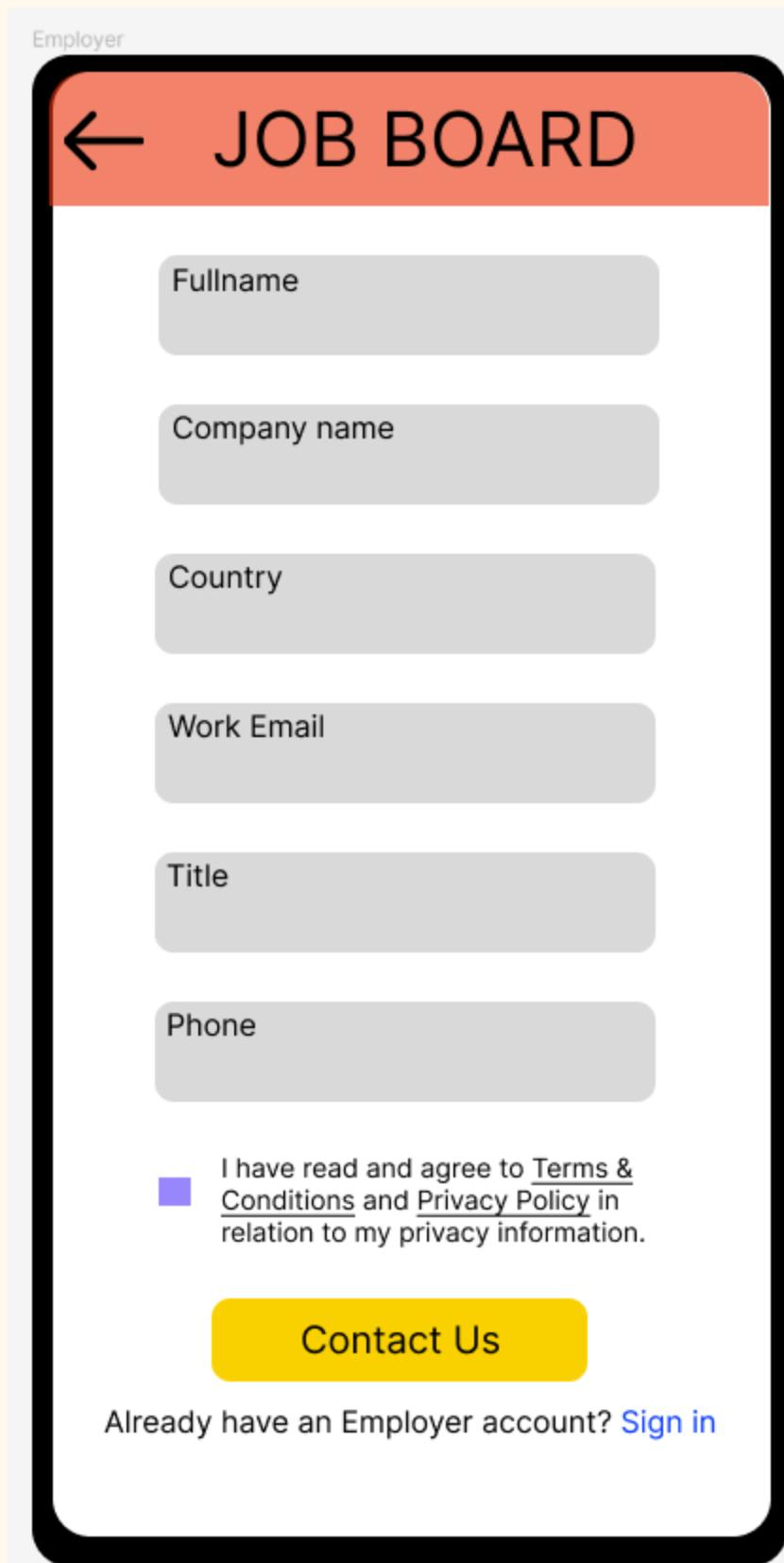
## sign up



## profile



## employer



The image shows a mobile application interface for employer registration. At the top left is a back arrow icon and the text "JOB BOARD". On the far left, above the input fields, is the word "Employer". The form consists of six input fields: "Fullname", "Company name", "Country", "Work Email", "Title", and "Phone". Below these fields is a checkbox followed by a terms and conditions statement. A large yellow button at the bottom right says "Contact Us". At the very bottom, there is a link for existing users.

Employer

# JOB BOARD

Fullname

Company name

Country

Work Email

Title

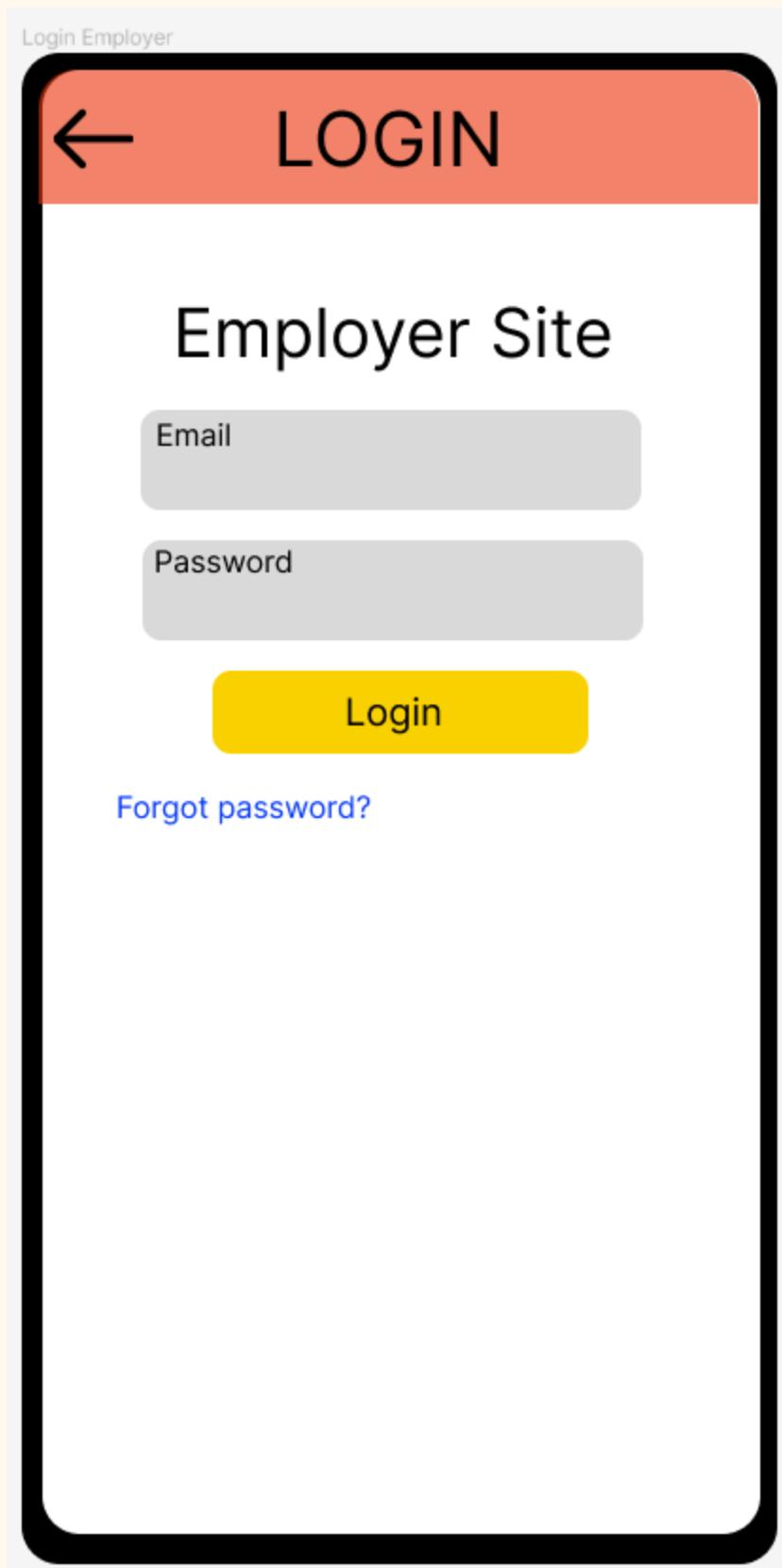
Phone

I have read and agree to [Terms & Conditions](#) and [Privacy Policy](#) in relation to my privacy information.

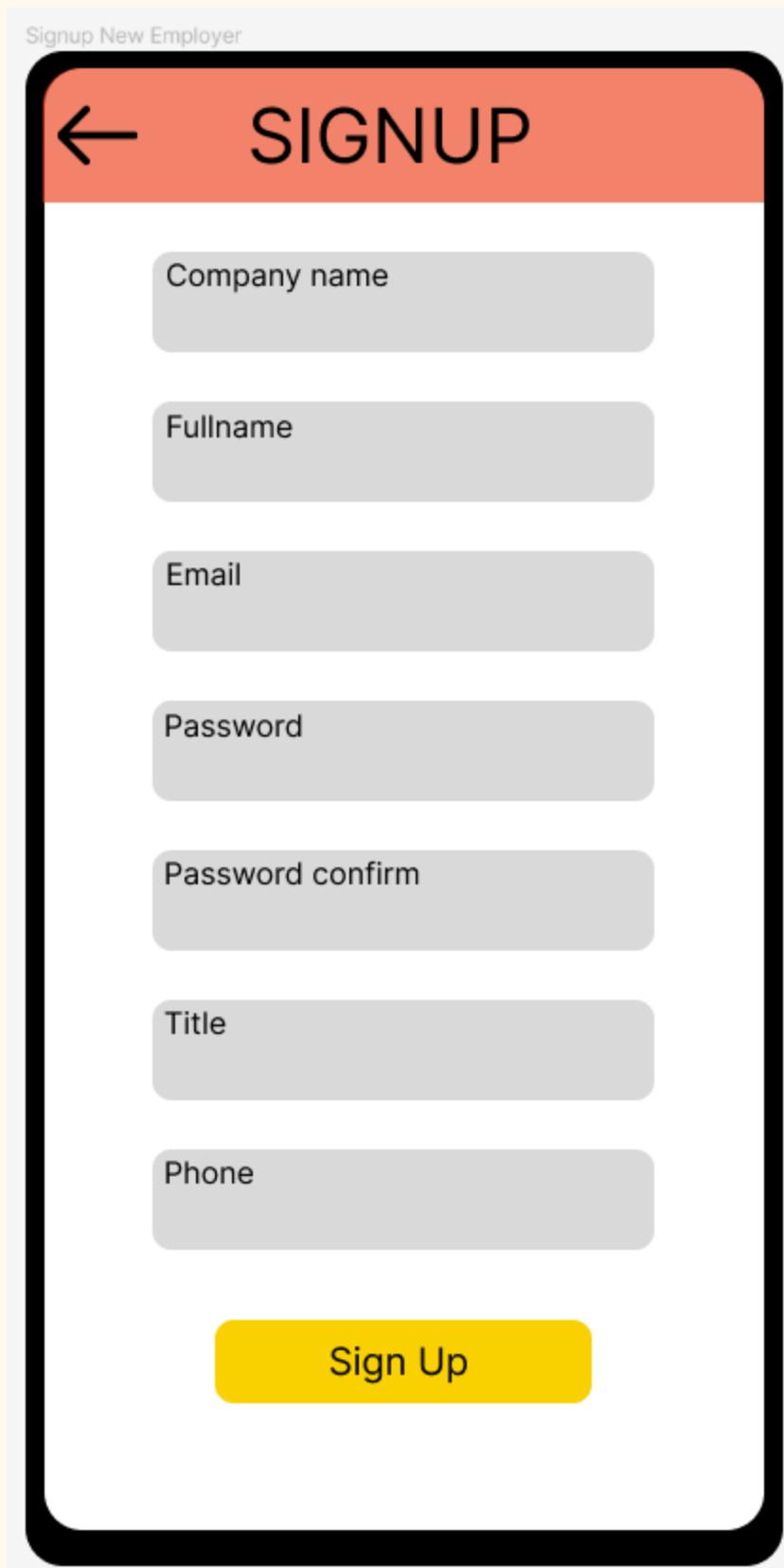
Contact Us

Already have an Employer account? [Sign in](#)

## login



## sign up new employer



## job details

Job Details

### ← Job Details

Job Title  
Company name  
Skill 1 Skill 2 Skill 3

job Information

Save Apply

The form consists of several sections: a header with a back arrow and title, a main content area with job title, company name, and skill tags, a large empty box for job information, and a footer with save and apply buttons.

## job apply

Job Apply

## Apply To

Job Title  
Company name  
Skill 1 Skill 2 Skill 3

Upload your CV  
.doc, .docx, .pdf, ... <5mb)

Select your CV

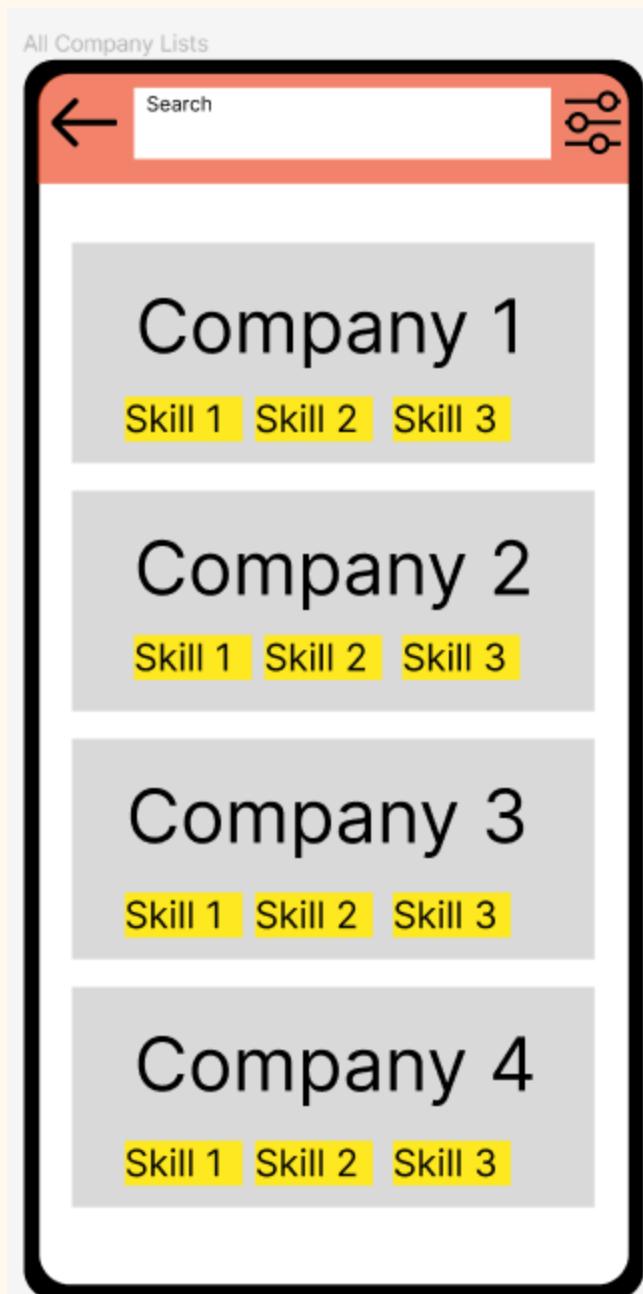
CV 1: .....

CV 2: .....

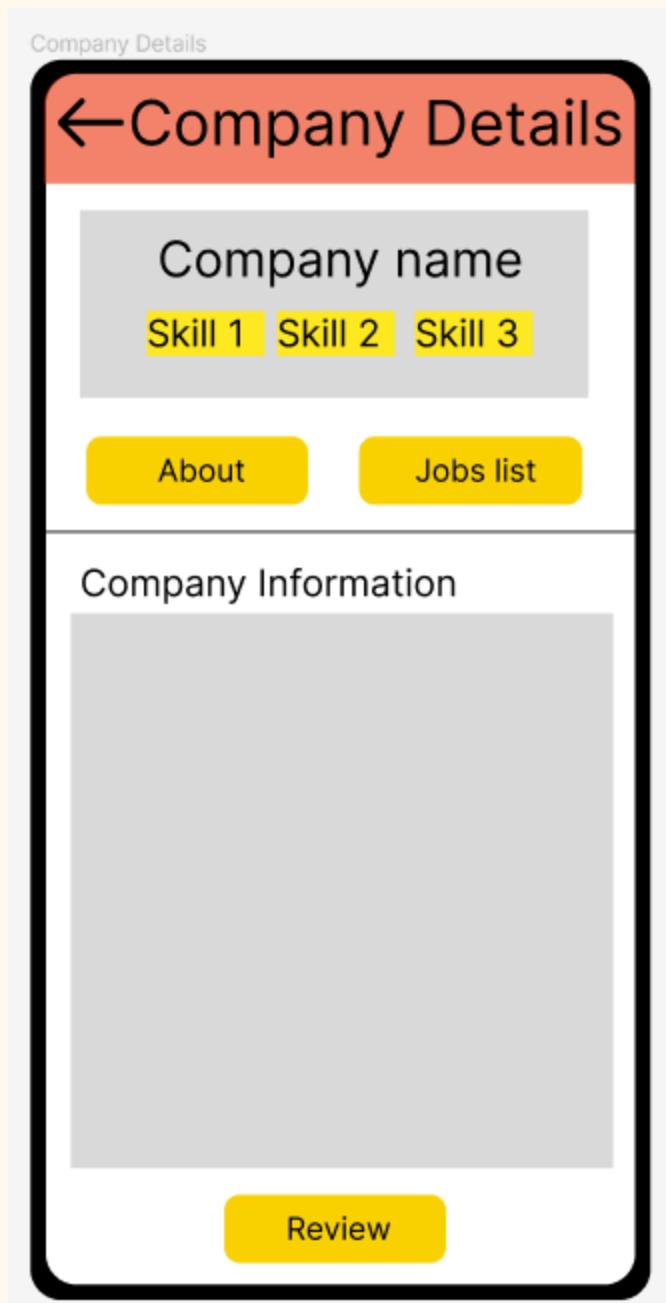
Fullname  
Email  
Phone

Confirm Apply

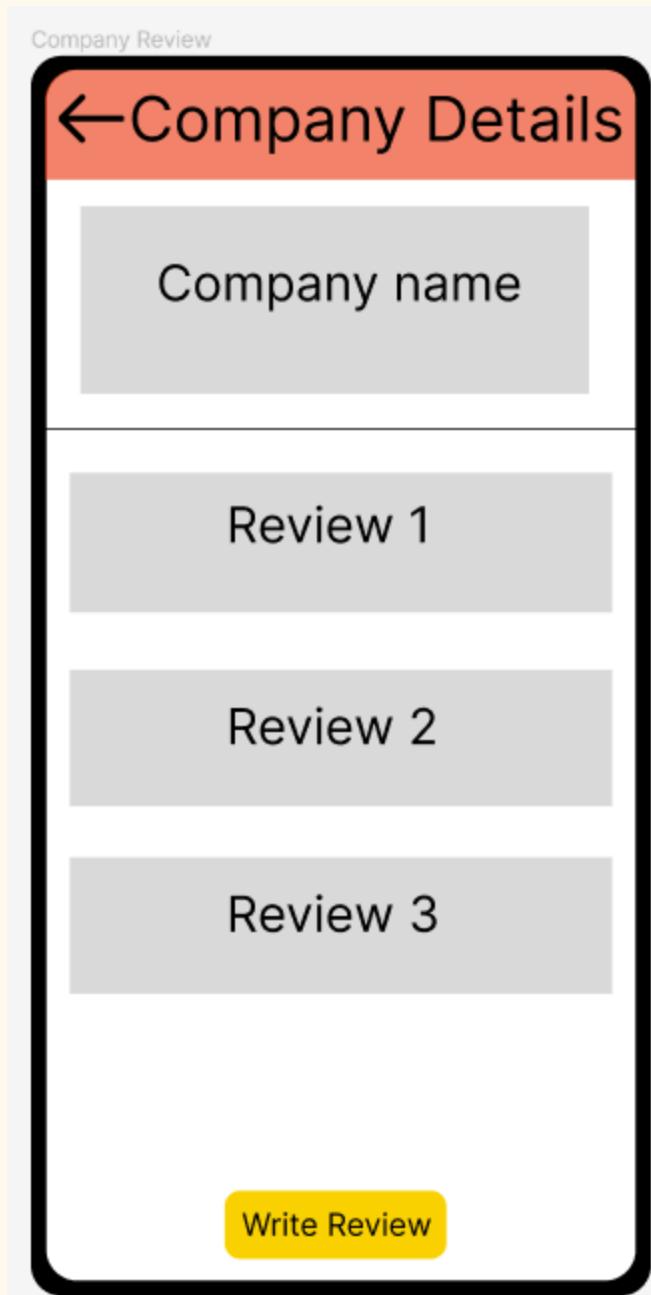
## All company lists



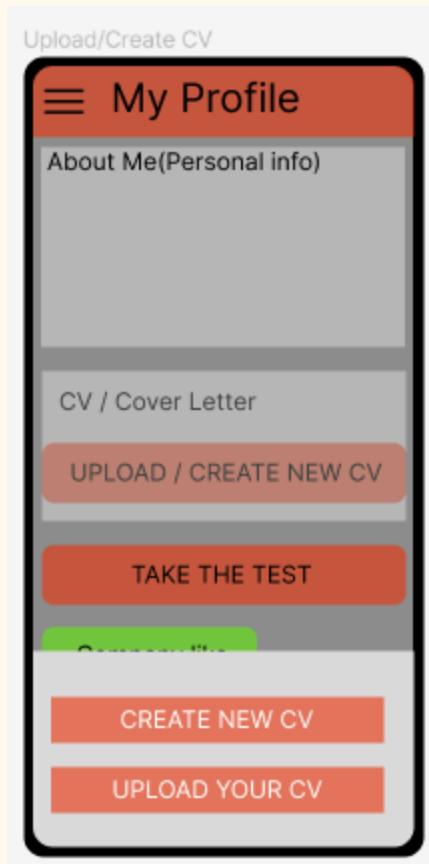
## Company details



## company review



## upload/create CV



## create new CV

The image displays two side-by-side mobile screen prototypes for a "Create New CV" feature. Both screens have a light blue header bar with the title "Create new CV" and a back arrow icon.

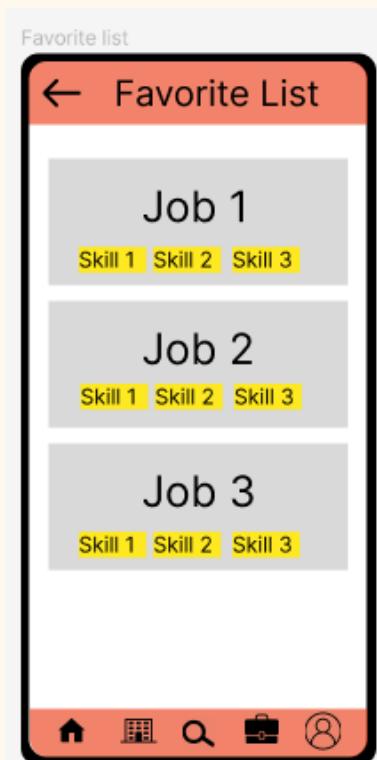
**Screen 1 (Left):**

- Fields: "CV Name" and "Language".
- Buttons at the bottom: "START MAKING CV" (orange) and "CANCEL" (orange).

**Screen 2 (Right):**

- Fields: "Fullname", "Email", "Phone", "Address", "Birthday", "Skill", "Certification", "Education", "Work experiences", and "Summary".
- Buttons at the bottom: "SAVE" (orange) and "PREVIEW" (orange).

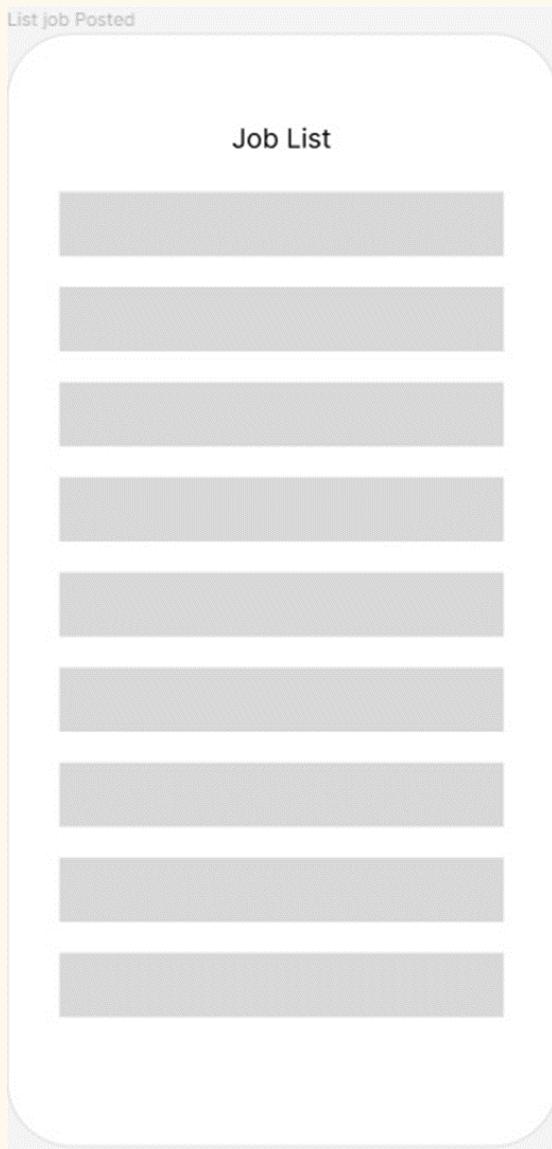
## favorite list



## take the test



## List Job Posted



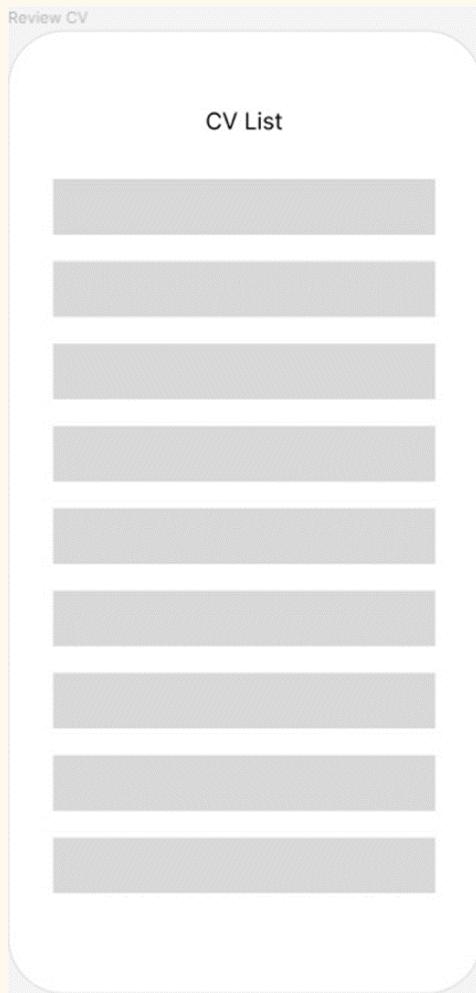
1.1 Description: Description: The "List Job Posted" function is a core feature of your job board project that allows employers or recruiters to view and manage the jobs they have posted on the platform. It provides a comprehensive and organized list of the job postings made by the user, enabling them to review, edit, and potentially remove job listings as needed.

Input: The "List Job Posted" function does not take any input parameters since it retrieves the job postings directly associated with the logged-in employer's account.

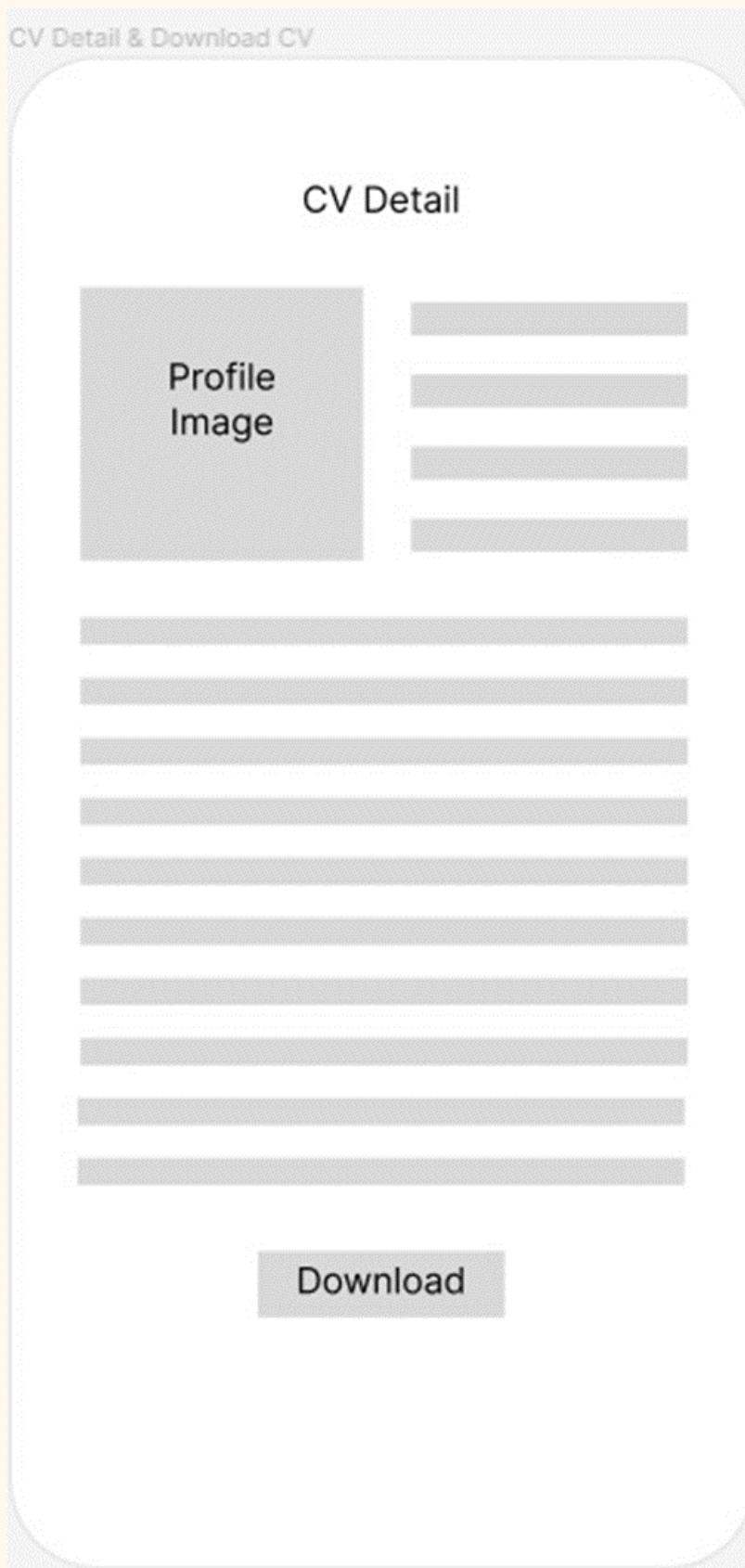
Output: The output of the "List Job Posted" function is a well-structured list of job postings made by the employer. Each job posting entry contains essential details about the job, such as job title, company name, location, posting date, and job status.

Algorithm: Ensure the user is authenticated and authorized as an employer or recruiter before proceeding with the function.

## Review CV



## **CV details**



Description: As a job finder, reviewing CVs (resumes) is a critical step in your job search process. Evaluating candidates' CVs effectively helps you identify potential matches for job opportunities and make informed decisions on who to connect with or apply to.

Input: CV (Resume): The CV (resume) of the job seeker that needs to be reviewed. The CV contains information about the job seeker's qualifications, work experience, skills, education, and other relevant details.

Output: Evaluation of the CV: The output of the CV review is an assessment of the job seeker's qualifications and suitability for the specific job or job search criteria. It includes feedback on the candidate's overall profile, strengths, areas of improvement, and alignment with the job requirements. Additionally, the output may indicate whether the candidate is a potential match for the job or if further consideration is needed.

## Download CV

Description: The "Download CV" function is an essential feature of your job board project that allows employers or recruiters to download the CV (resume) of a candidate who has applied for a job position. This function empowers employers to access and save the candidate's CV locally, enabling them to review it offline or share it with other decision-makers in the hiring process.

Input:

Candidate ID or Application ID: A unique identifier associated with the candidate's job application. This ID allows the system to identify the specific candidate whose CV needs to be downloaded.

Output:

CV File: The output of the "Download CV" function is the candidate's CV file, typically in a portable document format (PDF) or other commonly used formats. The CV file is downloaded to the employer's device for offline access and review.

Algorithm: Ensure the user (employer or recruiter) is authenticated and authorized to access and download candidate CVs. Only authorized users should have access to this function.

the candidate ID or application ID from the employer as input. This ID is usually provided through a user interface, such as a button or link associated with each candidate's application.

Validate the candidate ID or application ID received from the employer to ensure it corresponds to a legitimate candidate's application in the database.

Using the verified candidate ID, query the database to retrieve the relevant candidate's information, including their CV file path or document data.

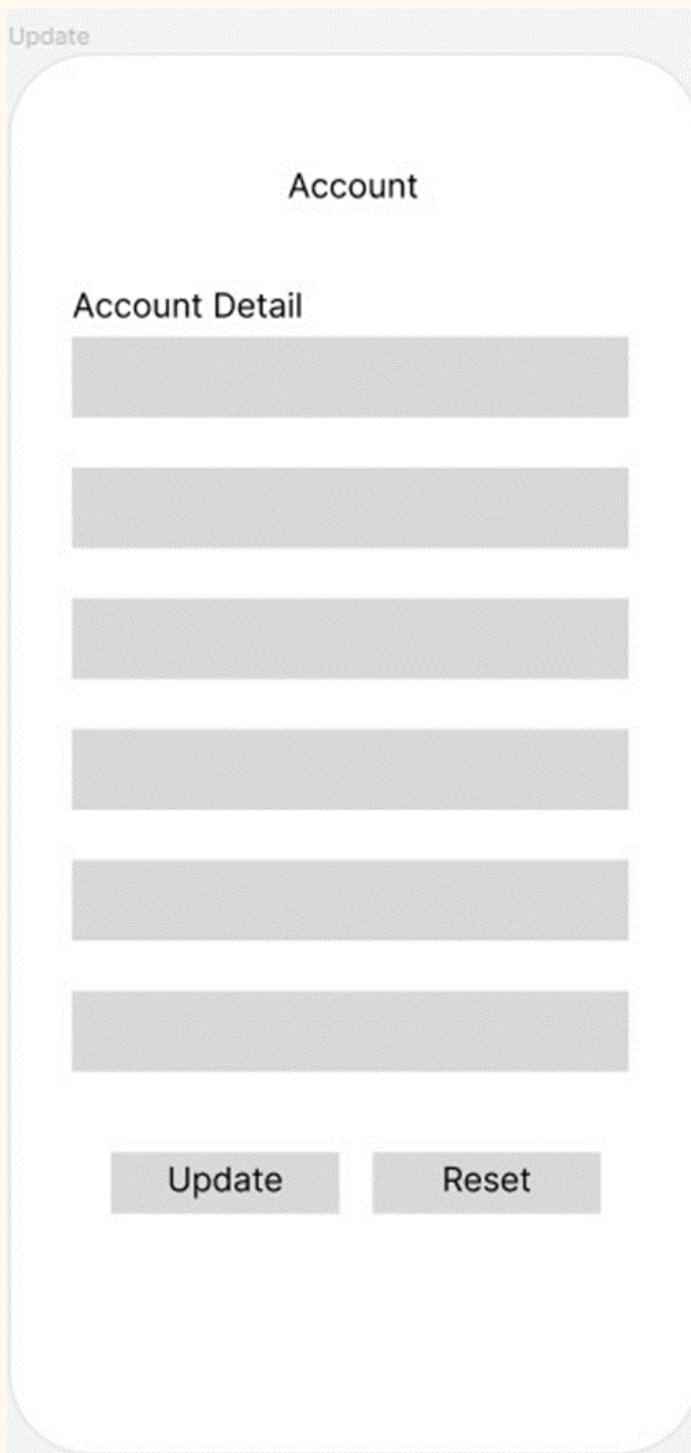
Once the candidate's information is retrieved, generate the CV file in the desired format (e.g., PDF) using the CV data obtained from the database.

Provide the generated CV file to the employer for download. This can be done by triggering the download process in the web application, prompting the user to save the CV file to their local device.

Optionally, log the download activity for auditing purposes. Keep a record of which employers have downloaded candidates' CVs and when.

Implement appropriate error handling to handle scenarios such as invalid candidate IDs, database connectivity issues, or any other errors that may occur during the download process. Display meaningful error messages to the employer in case of any issues.

## Update Employer Account



4.1 Description: The "Update Employer Account" function is a vital feature of your job board project that allows employers to modify and update their account information. Employers may need to update their company details, contact information, profile, or other relevant data to ensure accuracy and keep their accounts up-to-date.

**Input:** Employer ID: A unique identifier associated with the employer's account. This ID helps the system identify the specific employer account that needs to be updated.

**Updated Information:** The new or modified information that the employer wishes to update in their account. This may include company details, contact information, password change, or any other relevant data.

**Output:**

**Updated Account Information:** The output of the "Update Employer Account" function is confirmation that the employer's account information has been successfully updated. This feedback assures the employer that their changes have been saved.

**Algorithm:** Ensure the user is authenticated and authorized as an employer before proceeding with the update. Only authorized employers should have access to modify their account information.

Receive the employer's unique ID (or username/email) as input to identify the specific account that needs updating.

Obtain the updated information from the employer, which may include changes to company details, contact information, or other relevant data.

Validate the employer's identity by confirming that the provided employer ID matches the currently logged-in employer's account or the owner of the account making the update.

Using the verified employer ID, access the database or data storage where the employer's account information is stored.

Update the account information with the provided changes. This may involve updating fields such as company name, contact email, phone number, website, or any other editable data.

If the employer requested a password change, ensure that the new password meets the platform's password security requirements. Hash and securely store the updated password in the database.

Provide the employer with a confirmation message indicating that their account information has been successfully updated. This feedback assures the employer that their changes were saved and applied.

Implement proper error handling to manage any potential issues that may arise during the update process. Display meaningful error messages if there are problems with database connectivity, data validation, or any other errors.

#### 4.2 Input/Output data analysis:

4.2.1.input: click the “Edit Account” button

4.2.1.output: Display edit form

4.2.2.input: Fill out the form and click the “Update” button

4.2.2.output: return to the “Employer Profile” page if successful and display an error if fail

## 5. Create Account

Create Account

Account Detail

Create      Reset

This is a wireframe representation of a 'Create Account' form. At the top, it says 'Create Account'. Below that is a section titled 'Account Detail' which contains six horizontal grey bars, each representing a field for input. At the bottom of the form are two buttons: 'Create' on the left and 'Reset' on the right.

5.1 Description: The "Create Account" function is a fundamental feature of your job board project that allows job seekers and employers to register and create new accounts on the platform. This function enables users to provide their essential information, such as personal details, contact information, and login credentials, to gain access to the job board's full range of services and features.

**Input:**

User Type: The type of user account being created. This could be "Job Seeker" or "Employer," depending on whether the user is a candidate seeking job opportunities or an employer looking to post jobs and hire candidates.

Personal Information: The user's personal details, including full name, email address, and other relevant information required for account creation.

Company Information (For Employers): If the user is registering as an employer, collect additional details such as company name, company website, industry, and a brief description of the company.

Password: The user's chosen password used for account authentication. The password should be securely hashed before being stored in the database.

**Output:**

Account Creation Confirmation: The output of the "Create Account" function is a confirmation message indicating that the user's account has been successfully created. The confirmation message may also include instructions on how to proceed, such as logging in to the newly created account.

**Algorithm:**

Provide a user type selection option during the account creation process. Allow users to choose between "Job Seeker" and "Employer" to determine the type of account they wish to create.

Request the user's personal information, such as full name, email address, and any other necessary details based on the chosen user type (job seeker or employer).

Perform validation on the provided user information to ensure it meets the required format and validity criteria. Check for unique email addresses to prevent duplicate accounts.

Ask the user to create a password for their account. Enforce strong password requirements, such as minimum length, a combination of uppercase and lowercase letters, numbers, and special characters, to enhance security.

Hash the user's chosen password using a secure hashing algorithm (e.g., bcrypt or SHA-256) before storing it in the database. Storing hashed passwords enhances account security by protecting user passwords from unauthorized access.

If the user selected "Employer" as their user type, collect additional company-related information, such as company name, website, industry, and a brief company description.

Create a new user account in the database with all the collected user information. Assign a unique user ID and store the account details securely.

Optionally, implement an email verification process to validate the user's email address. Send a verification link to the provided email, and the user must click on the link to confirm the registration.

Provide the user with a confirmation message indicating that their account has been successfully created. Optionally, include instructions on how to proceed, such as logging in to their new account.

Implement proper error handling throughout the account creation process to handle any issues that may arise, such as database errors, validation errors, or other exceptions. Display informative error messages to guide users in case of errors.

## 5.2 Input/Output data analysis

5.2.input: Fill out the form and click the “Create” button

5.2.output: return to the “Account Management” page if successful and display an error if fail

### 3.4.2 DETAILED DESCRIPTION OF THE FUNCTION

#### AND INPUT/OUTPUT DATA ANALYSIS

Web

For candidate

Show jobs

Description:

The searchJob() function is a core component of the job board project, enabling users to search for job listings based on their specific criteria. It empowers job seekers to find relevant job opportunities and helps employers reach potential candidates by providing a robust and efficient search mechanism. This function takes various search parameters as input and returns a list of job listings that match the provided criteria.

**Input:**

- keywords (optional): A string containing keywords related to the job title, required skills, or company name that the user wants to search for. If provided, the function will filter job listings that include these keywords in their descriptions or titles.
- location (optional): A string specifying the desired location for the job, such as a city, state, or country. If provided, the function will narrow down job listings based on this location.
- jobType (optional): A string indicating the type of job the user is interested in (e.g., full-time, part-time, contract, internship). If provided, the function will filter job listings that match the specified job type.
- experienceLevel (optional): A string representing the desired experience level for the job (e.g., entry-level, mid-level, senior). If provided, the function will include only job listings that match the specified experience level.
- category (optional): A string specifying the job category or industry (e.g., IT, healthcare, finance). If provided, the function will filter job listings that fall under the specified category.
- salaryRange (optional): A numerical range indicating the desired salary for the job. If provided, the function will include only job listings that offer salaries within the specified range.
- educationLevel (optional): A string representing the minimum education level required for the job (e.g., high school, bachelor's degree, master's degree). If provided, the function will filter job listings based on the specified education level.

**Output:**

function returns a list of job listings that match the user's search criteria. Each job listing includes essential details about the job, such as:

- jobTitle: The title or position of the job.
- companyName: The name of the hiring company or organization.
- location: The location of the job.
- jobType: The type of job (e.g., full-time, part-time, contract).
- experienceLevel: The required experience level for the job.
- salary: The salary offered for the job (if available).
- jobDescription: A brief description of the job responsibilities and requirements.

**Algorithm:**

- Receive user inputs for the search criteria, including keywords, location, jobType, experienceLevel, category, salaryRange, and educationLevel.
- Query the job database or API to retrieve all available job listings.
- Apply filtering to the retrieved job listings based on the provided search criteria. If a specific criterion is not provided, skip it during the filtering process.
- Rank the filtered job listings based on relevance or any other predefined criteria.
- Create a list of job objects containing the essential details of each job listing that matches the user's search criteria.
- Return the list of job objects as the output of the searchJob() function.

**View job details****Description:**

- function allows users to access comprehensive information about a specific job listing. When users search for jobs on the job board, they can click on individual job listings to view more in-depth details about the job, including the job description, required qualifications, company information, application instructions, and any other relevant details.

**Input:**

- Job ID or Unique Identifier: The unique identifier of the job listing that the user wants to view details for. This ID is usually obtained when the user clicks on a specific job listing from the search results.

**Output:**

- Job Details: A detailed view of the selected job listing, containing the following information:
  - Job Title: The title or position name of the job.
  - Company Name: The name of the company offering the job.
  - Job Location: The location of the job (e.g., city, state, country).
  - Job Type: The type of job (e.g., full-time, part-time, contract, internship).
  - Experience Level: The required experience level for the job (e.g., entry-level, mid-level, senior).

- Job Category: The category or industry the job belongs to (e.g., IT, healthcare, finance).
- Salary Range: The salary range for the job (if available).
- Education Level: The minimum education level required for the job (e.g., high school, bachelor's degree, master's degree).
- Job Description: A detailed description of the job, including responsibilities, tasks, and any specific requirements.
- Required Qualifications: A list of skills, experience, and qualifications that candidates must possess.
- Company Profile: Information about the hiring company, such as its mission, culture, and values.
- Application Instructions: Guidance on how to apply for the job, including any specific application requirements or documents needed.
- Application Deadline: If applicable, the deadline for submitting job applications.

### Algorithm:

Receive the unique Job ID as input from the user.

Query the job database or API using the Job ID to retrieve the detailed information for the selected job listing.

Format the retrieved information into a structured view, presenting the job details in a user-friendly manner.

Display the job details on the job board interface for the user to view.

### Comments

#### Descriptions

This function allows users of your job board platform to leave feedback, reviews, and comments about companies they have worked for or interacted with during their job search process. This function is valuable as it provides job seekers with insights into the company's work culture, reputation, and overall employee experiences. Additionally, it helps employers understand how their company is perceived in the job market and gives them an opportunity to respond to feedback and improve their employer brand.

**Input:**

- Company Name: The name of the company that the user wants to leave a comment about.
- User ID (optional): The unique identifier of the user leaving the comment. If user authentication is enabled, this can be automatically retrieved from the logged-in user's account.
- Comment: The text of the comment or review left by the user. This can include both positive and negative feedback, work environment assessment, growth opportunities, management, and any other relevant aspects of the company.

**Output:**

- Success/Failure Message: A confirmation message indicating whether the comment was successfully submitted or if there was an error during the process.

**Algorithm:**

Receive the company name, user ID (if available), and the comment text from the user.

Validate the inputs to ensure that the company name exists in the database and that the comment text is within character limits and free of any harmful content.

If user authentication is enabled, check if the user ID is valid and has the necessary permissions to leave a comment about the company.

Save the comment and associated details in the company's record in the database. If the company does not have a comments section, create one to store user comments.

Optionally, implement a moderation process, where comments are reviewed by a moderator before being publicly displayed to ensure compliance with community guidelines and prevent inappropriate content from being published.

Provide users with the ability to edit or delete their comments in case they want to update their feedback at a later time.

Display the company's average rating or sentiment score based on the cumulative feedback received from multiple users.

Allow other users to view and read comments about the company on the company's profile page.

Implement a reporting mechanism that allows users to flag inappropriate or fraudulent comments for review and removal if necessary.

## Create CV

### Description:

The createCV() function is a crucial feature of the job board platform that allows candidates (job seekers) to create and build their professional Curriculum Vitae (CV) directly on the platform. A CV is a comprehensive document that showcases a candidate's educational background, work experience, skills, achievements, and other relevant information to potential employers. By offering this function, the job board platform provides candidates with a streamlined and user-friendly way to create a well-structured and visually appealing CV that can enhance their chances of getting noticed by employers.

### Input:

- Personal Information: The candidate provides personal details, such as their full name, contact information (phone number, email address), address, and optionally, a professional profile picture.
- Education Details: The candidate inputs information about their educational qualifications, including school/university names, degrees earned, majors, and graduation dates.
- Work Experience: The candidate lists their work history, including job titles, company names, employment dates, job responsibilities, and any notable achievements.
- Skills: The candidate mentions their relevant skills, both technical and soft skills, that are valuable in their desired job field.
- Certifications: If applicable, the candidate includes any certifications or licenses they have obtained that can boost their qualifications.
- Projects or Portfolio (Optional): Candidates can showcase their projects, portfolio, or links to their professional website or GitHub repositories to demonstrate their expertise.
- Career Objective or Summary (Optional): Candidates may add a brief summary or career objective statement to highlight their career goals and aspirations.

### Output:

- CV Preview: After the candidate inputs their information, the createCV() function generates a live preview of the CV, allowing the candidate to view how their CV will appear to potential employers. This preview enables candidates to make real-time adjustments and ensure the CV looks professional and well-organized.

- Save or Download Options: Once the candidate is satisfied with their CV, they can choose to save it directly on the platform for future editing or download it as a PDF or other standard document format. The download option allows candidates to share their CV with prospective employers or print it for offline use.

## **predict salary**

### **Description:**

function is designed to estimate or predict a salary range for a job based on input information provided by the user. This function can be a valuable tool for both job seekers and employers, helping them understand salary expectations for specific job roles and making informed decisions during the hiring process.

### **Input:**

- Job Title: The title or name of the job role for which the user wants to predict the salary.
- Location: The location where the job is based, as salaries can vary significantly between regions.
- Experience Level: The user's or candidate's level of experience in the field (e.g., entry-level, mid-level, senior).
- Education Level: The highest level of education attained by the user (e.g., high school, bachelor's degree, master's degree).
- Industry: The industry or sector in which the job is located (e.g., IT, healthcare, finance).
- Skills and Certifications: Any specific skills or certifications required for the job that may impact the salary estimation.

### **Output:**

- Predicted Salary Range: The function will provide an estimated salary range for the specified job based on the input information provided by the user. The range may be in the form of a minimum and maximum salary value.

### **Algorithm:**

- Receive user inputs for the job title, location, experience level, education level, industry, and any relevant skills or certifications.
- Query the salary database or API to retrieve historical salary data for similar job roles in the specified location and industry. The database should ideally contain salary ranges based on experience and education levels.
- Apply statistical analysis or machine learning algorithms to predict the salary range based on the provided input information and historical salary data.

- If the specific job title or role is not found in the database, consider using similar job titles or roles to make an estimate.
- Adjust the predicted salary range based on factors like the candidate's experience level and education level, as these can influence the final salary offer.
- Display the estimated salary range to the user in a clear and understandable format.

## Sign in

### Description:

function is a fundamental component of any web application or platform that requires user accounts and personalized experiences. It allows users to access their accounts by providing their credentials (username/email and password) and authenticate their identity. Here's a detailed description of the "Sign In" function:

### Input:

- Username/Email: The unique identifier associated with the user's account. This could be their username, email address, or a combination of both.
- Password: The user's secret passphrase used for authentication. Passwords should be securely hashed before storing in the database.

### Output:

- Authentication Status: A response indicating whether the sign-in attempt was successful or not.
- User Information (Optional): If the sign-in attempt is successful, the function may return basic user information, such as their username, email, user ID, or any other relevant data associated with the user's account.

### Algorithm:

Receive the username/email and password entered by the user on the sign-in form.

Check the database or user repository to find an account that matches the provided username/email.

If no account is found, return an authentication error message (e.g., "Invalid username/email or password").

If an account is found, retrieve the associated password hash from the database.

Hash the entered password using the same hashing algorithm used during registration and compare it with the stored password hash.

If the hashes match, the user is authenticated, and the sign-in attempt is successful. Create a session for the user, granting them access to protected areas of the website or application.

If the hashes do not match, return an authentication error message (e.g., "Invalid username/email or password").

Optionally, log successful and unsuccessful sign-in attempts for security monitoring and analysis.

## sign up

### Descriptions:

function is a fundamental component of many web applications, including job boards, social media platforms, e-commerce websites, and more. It allows new users to create an account and register on the platform, enabling them to access personalized features, save preferences, and interact with the site's content or services. Below is a detailed description of the "Sign Up" function:

### Input:

- Username: The desired username chosen by the user. It should be unique and not already in use by another user.
- Email: The user's email address, which will serve as a primary identifier for the account and a means of communication with the user.
- Password: The user's chosen password for the account. It should meet security requirements, such as minimum length and a combination of letters, numbers, and special characters.
- Confirm Password: A field to re-enter the chosen password to ensure the user enters the correct password twice.
- Optional Fields (e.g., Full Name, Date of Birth, Profile Picture, etc.): Additional information that the user can provide during the sign-up process to complete their profile or enhance their experience on the platform.

### Output:

- Success Message: A confirmation message indicating that the sign-up process was successful, and the user's account has been created.
- User Account: The newly created user account with all the provided information. The account will be stored in the platform's user database for future logins and interactions.

**Algorithm:**

Receive the user's input, including username, email, password, and any optional fields, via a sign-up form on the platform's web page or mobile app.

Validate the input data to ensure that all required fields are filled correctly, the email is in a valid format, and the password meets the security requirements.

Check if the chosen username and email are unique and not already associated with an existing account in the user database. If either the username or email is already in use, prompt the user to choose a different one.

Hash the user's password to securely store it in the database, ensuring that the actual password remains encrypted for security purposes.

Create a new user account in the platform's user database, including all the provided information (username, email, optional fields).

Generate a unique user ID for the new account to serve as a primary key in the database.

Send a verification email to the user's provided email address to confirm the sign-up and activate the account. Include a verification link that the user can click to verify their email address and activate the account.

Display a success message on the sign-up page, indicating that the registration was successful and providing any additional instructions for the user.

Optionally, automatically log the user in after successful registration, redirecting them to a personalized welcome or onboarding page.

## Test

### Description:

This function allows users to practice their interview skills by answering a set of predefined questions directly on the platform. It aims to simulate a real interview experience, helping users become more confident and prepared for actual job interviews.

### Input:

- Predefined Interview Questions: A list of interview questions stored in the platform's database. These questions should cover a variety of topics, such as strengths and weaknesses, past experiences, problem-solving skills, and industry-specific knowledge.

### Output:

- User's Answers and Feedback: After the user completes the trial test interview by answering all the questions, the platform provides feedback on their responses. The feedback can be in the form of general tips, suggestions for improvement, or specific feedback on each answer.

### Algorithm:

Retrieve the predefined interview questions from the platform's database or storage.

Display the instructions to the user, explaining the purpose of the trial test interview and the number of questions they need to answer.

Randomly select a subset of questions from the predefined list (e.g., 5-10 questions) to present to the user during the trial test interview. This randomization helps keep the practice sessions fresh and varied.

Present each question to the user one by one, allowing them a moment to read and understand the question before proceeding.

Provide a text box or input field for the user to type or dictate their answer directly on the platform.

Set a time limit for each question (e.g., 1-3 minutes) to simulate the time constraints of a real interview and encourage concise, focused responses.

Allow users to skip a question if they are unsure or not comfortable answering it. This can help reduce stress and make the experience more user-friendly.

Once the user answers all the questions or completes the time limit for the trial test interview, process their responses.

Provide feedback on the user's performance. The feedback may include general advice on interview techniques, communication skills, and body language. Additionally, it may offer specific feedback on each answer, highlighting strengths and areas for improvement.

Offer the option for users to review their recorded responses if the platform supports audio or video recording. This enables users to self-assess and learn from their performance.

Encourage users to practice the trial test interview multiple times with different sets of questions to gain more confidence and improve their interview skills.

## For Employer

### Post new job

#### Descriptions:

function is a critical feature of your job board platform that allows employers or recruiters to create and publish new job listings on the platform. This function facilitates the process of adding job opportunities to the platform, making them visible to potential candidates who are actively searching for employment.

#### Input:

- Job Title: A string representing the title or designation of the job position.
- Company Name: The name of the company or organization offering the job opportunity.
- Job Location: The physical location or remote work status of the job.

- Job Type: The type of job (e.g., full-time, part-time, contract, internship).
- Job Description: A detailed description of the job, including responsibilities, qualifications, and other relevant information.
- Experience Level: The desired experience level for the job (e.g., entry-level, mid-level, senior).
- Required Skills: A list of skills or qualifications necessary for the job.
- Education Level: The minimum education level required for the job (e.g., high school, bachelor's degree, master's degree).
- Salary Range: The salary range offered for the position (optional).
- Application Deadline: The date by which candidates must apply for the job (optional).

**Output:**

- Confirmation Message: A message indicating that the job listing has been successfully posted on the platform.

## update posted job

**Description:**

This function is a critical feature of your job board platform that allows employers to make changes or updates to the job listings they have posted. Employers often need to update job details, such as the job description, requirements, application deadline, or other relevant information, and this function facilitates those modifications efficiently.

**Input:**

- Job ID: A unique identifier for the job listing that the employer wants to update. This ID helps the system identify the specific job to be edited.
- Updated Job Details: The revised information and changes that the employer wants to make to the job listing. This could include changes to the job title, description, location, salary, application process, etc.

**Output:**

- Confirmation Message: A message confirming that the job listing has been successfully updated and reflecting the changes made.

## For Controller

### Manage controller manage employer, manage candidate

#### Descriptions:

functions for managing users are fundamental operations in most applications that involve user accounts or user data. These operations allow administrators or authorized users to manage user information effectively. Here's a detailed description of each CRUD function for user management:

#### 1. Create User

##### Input:

- User details: This includes information such as username, email address, password (encrypted), full name, contact information, role/permissions, and any other relevant user-specific attributes.

##### Output:

- User ID: Upon successful creation, the function returns a unique identifier (ID) for the newly created user.

Description: The "createUser" function is responsible for adding a new user to the system. It takes user inputs containing the necessary details for creating a user account, validates the inputs, and then saves the user information in a database or user storage system. Before saving, the function may perform additional checks, such as ensuring the username and email address are unique and meet specific validation criteria. After successful creation, the function returns the unique user ID, which can be used for future references.

#### 2. Read User

##### Input:

- User ID: The unique identifier of the user whose information needs to be retrieved.

##### Output:

- User Details: The function returns all the available information associated with the user, such as username, email address, full name, contact information, role/permissions, and other relevant attributes.

Description: The "readUser" function retrieves user information based on the provided user ID. It searches the database or user storage system for the user with the matching ID and returns all the stored details associated with that user. This function is essential for displaying user profiles, accessing user-specific settings, or performing user-related actions that require access to user data.

### 3. Update User

Input:

- User ID: The unique identifier of the user whose information needs to be updated.
- Updated User Details: The new information that should replace the existing user details, such as a new email address, updated contact information, or changes in user roles/permissions.

Output:

- Success/Failure: The function returns a status indicating whether the update was successful or encountered an error.

Description: The "updateUser" function allows administrators or authorized users to modify the information associated with a specific user account. It takes the user ID and the updated user details as input, then searches for the corresponding user in the database or user storage system. If the user is found, the function updates the relevant attributes with the new information. After the update is complete, the function returns a success status. If the user ID is not found or if any other error occurs during the update process, the function returns a failure status.

### 4. Delete User

Input:

- User ID: The unique identifier of the user account that needs to be deleted.

Output:

- Success/Failure: The function returns a status indicating whether the deletion was successful or encountered an error.

Description: The "deleteUser" function is responsible for removing a user account from the system. It takes the user ID as input, searches for the corresponding user in the database or user storage system, and

then deletes the user record. After successful deletion, the function returns a success status. If the user ID is not found or if any error occurs during the deletion process, the function returns a failure status. It's important to note that some applications may implement soft deletes, where user accounts are flagged as deleted instead of being permanently removed from the database.

## Monitor behavior

### Descriptions:

function is a critical component of a system that tracks and analyzes the behavior of users interacting with a website, application, or digital platform. Its primary purpose is to gather valuable data on how users navigate, interact, and engage with the system's features and content. By monitoring user behavior, businesses and developers can gain insights into user preferences, pain points, and usage patterns, which can inform decision-making and lead to improvements in the user experience.

### Input:

- User click on any button

### Output:

- Count the number of any click to the database

## review & approve job

### Descriptions:

function is to approve the job post by employer, before show on the web for candidate to view and submit CV

### Input:

- User click on approved button

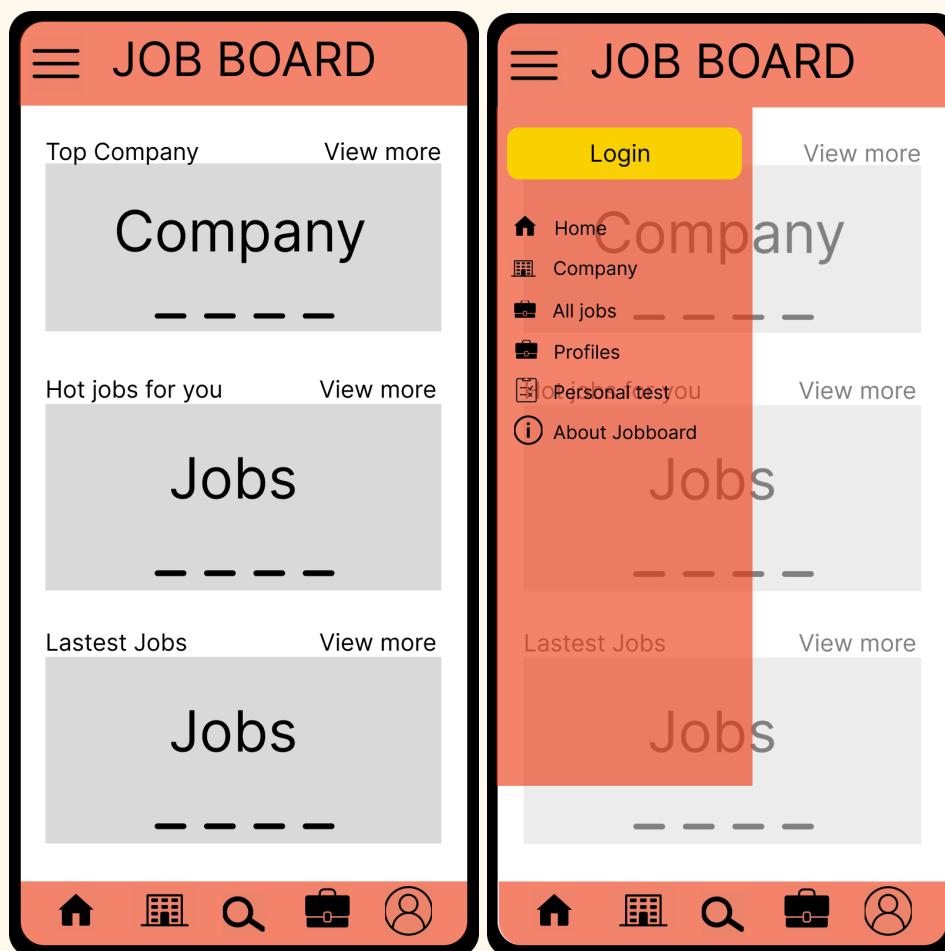
### Output:

- Approved job show up to the web for candidate to apply

## MOBILE

### 1. Home Page

**Description:** Home page will show some information about company and jobs for the database.



## 2. Login

**Description:** The login function of [Your Job Board Name] allows registered users to access their accounts and unlock a world of opportunities. Whether you're a job seeker looking to apply for jobs or an employer managing your company's job listings, logging in is the first step towards a seamless and personalized experience.

### Why Login Matters:

**Security and Privacy:** The login function ensures that only authorized individuals can access specific account information, maintaining the confidentiality and security of personal data.

**Personalization:** Logging in allows [Your Job Board Name] to tailor the user experience based on individual preferences and past interactions, making the job search or hiring process more efficient and relevant.

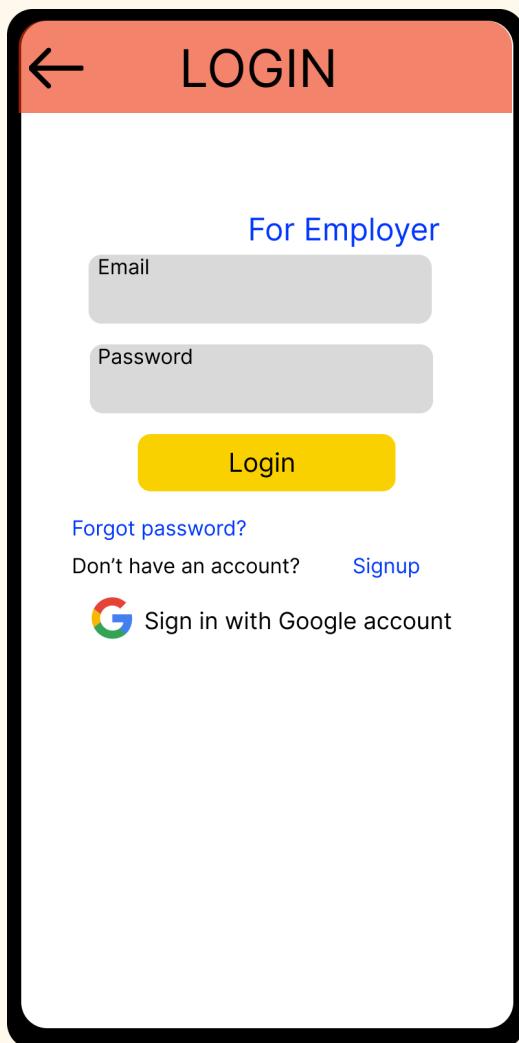
**Seamless Experience:** By logging in, users can seamlessly switch between various features and functionalities, streamlining the overall user journey.

## Login Candidate

**Access Your Dashboard:** Once logged in, candidates gain access to their personalized dashboard. From here, you can view and manage your profile, track job applications, and set up job alerts.

**Apply to Jobs Effortlessly:** Logging in enables you to apply to job openings with just a few clicks. Say goodbye to filling out lengthy forms repeatedly – your application details will be saved, making the process faster and more convenient.

**Stay Updated:** Keep yourself informed with the latest job listings and industry news by opting to receive regular updates via email or notifications.



### Input:

**email:** The email provided by the user during login.

**password:** The user's password associated with the provided email.

### Output:

Return to Home page with Candidate authorized



## Login Employer

**Manage Your Job Listings:** Employers can log in to their accounts to create, edit, and remove job listings as needed. Your dashboard provides a centralized hub for managing all your company's recruitment activities.

**Review Applications:** View and evaluate the applications received for your job postings in an organized manner. Shortlist candidates, schedule interviews, and communicate with potential hires, all within the platform.

**Monitor Your Company's Presence:** Gain insights into your company's performance on [Your Job Board Name]. See how many candidates have viewed your listings, the number of applications received, and other essential metrics.

**JOB BOARD**

Fullname

Company name

Country

Work Email

Title

Phone

I have read and agree to [Terms & Conditions](#) and [Privacy Policy](#) in relation to my privacy information.

**Contact Us**

Already have an Employer account? [Sign in](#)

**LOGIN**

**Employer Site**

Email

Password

**Login**

[Forgot password?](#)

## Input:

Employer will provide information about the company.

**Output:**

Employer will receive the Employer administrator account from Job's Board.

### 3. Sign Up

**Description:** The signup function empowers new users to join our vibrant community and access a world of opportunities. Whether you're a job seeker eager to explore exciting career prospects or an employer looking to discover top-tier talent, creating an account is the first step towards unlocking a range of benefits.

**Why Sign Up Matters:**

**Personalized Experience:** Creating an account allows us to tailor your job search or hiring journey based on your preferences and interactions, ensuring a more relevant and efficient experience.

**Job Seeker-Recruiter Connection:** The signup function facilitates seamless connections between job seekers and recruiters, fostering meaningful and productive relationships.

**Data Security:** Your personal information is safe and secure with us. We follow industry best practices to protect your data and prioritize your privacy.

## Sign Up Candidate

**Build Your Profile:** Creating an account allows you to build a comprehensive profile that showcases your skills, experiences, and qualifications. Your profile acts as a digital resume, helping you stand out to potential employers.

**Browse Jobs:** As a registered user, you can explore a wide array of job listings across various industries and locations. Our intuitive search and filtering options make it easy to find the most relevant opportunities.

**Apply with Confidence:** With an account, you can apply to jobs confidently and track your applications in one centralized dashboard. We streamline the process to save you time and effort.

**Receive Job Alerts:** Stay ahead of the competition by setting up personalized job alerts. We'll notify you when new positions matching your preferences become available.

The image shows a mobile application interface for sign-up. At the top, there is a red header bar with a white back arrow icon on the left and the word "SIGNUP" in white capital letters. Below the header are six input fields, each with a light gray background and a thin black border. The fields are labeled "Fullname", "Email", "Password", "Password confirm", "Phone", and "Birthday". At the bottom of the screen is a large, solid yellow rectangular button with the text "SignUp" in white. The entire form is set against a white background with a thick black border around the main content area.

### Input:

**email:** The email address provided by the user during the signup process. This will be used as a unique identifier for the user's account.

**password:** The user's chosen password for the account. It should be securely hashed before storage.

**Additional User Information:** Depending on your registration requirements, you may also collect other details such as name, contact information, and preferences.

### **Output:**

**Account Creation Result:** A boolean value indicating whether the account creation was successful (true) or unsuccessful (false). If the provided email is unique, and all required information is valid, the function returns true, indicating successful account creation.

**Error Message (Optional):** If the account creation fails, the function may return an error message detailing the reason for the failure. Common reasons could include an existing email address or validation errors in the input data.

**User Data (Optional):** Upon successful account creation, the function may return relevant user data, such as the user's profile information, account type, and any other preferences or settings associated with the account.

**Email Verification Status (Optional):** If your application requires email verification, the function may return a status indicating whether the user's email has been verified or is pending verification.

## Sign Up Employer

**Post Job Openings:** Creating an account enables employers to post job listings and reach a diverse pool of talented candidates. Showcase your company's culture and unique offerings to attract the right fit.

**Find Top Talent:** Access a rich talent pool of registered job seekers. Our platform allows you to search for candidates based on specific criteria, making it easier to find the perfect match for your company.

**Manage Your Listings:** Keep track of all your job listings in one convenient dashboard. Edit, update, or remove listings with ease.

**Enhanced Company Profile:** With an account, you can create a comprehensive company profile, providing job seekers with valuable insights into your organization.

The image shows a mobile application's sign-up interface. At the top left is a back arrow icon, and the word "SIGNUP" is centered above a series of input fields. There are seven input fields, each with a placeholder label: "Company name", "Fullname", "Email", "Password", "Password confirm", "Title", and "Phone". A prominent yellow "Sign Up" button is positioned at the bottom of the screen.

**Input:**

**email:** The email address provided by the user during the signup process. This will be used as a unique identifier for the user's account.

**password:** The user's chosen password for the account. It should be securely hashed before storage.

**Additional User Information:** Depending on your registration requirements, you may also collect other details such as name, contact information, and preferences.

### Output:

**Account Creation Result:** A boolean value indicating whether the account creation was successful (true) or unsuccessful (false). If the provided email is unique, and all required information is valid, the function returns true, indicating successful account creation.

**Error Message (Optional):** If the account creation fails, the function may return an error message detailing the reason for the failure. Common reasons could include an existing email address or validation errors in the input data.

**User Data (Optional):** Upon successful account creation, the function may return relevant user data, such as the user's profile information, account type, and any other preferences or settings associated with the account.

**Email Verification Status (Optional):** If your application requires email verification, the function may return a status indicating whether the user's email has been verified or is pending verification.

#### 4. Profile

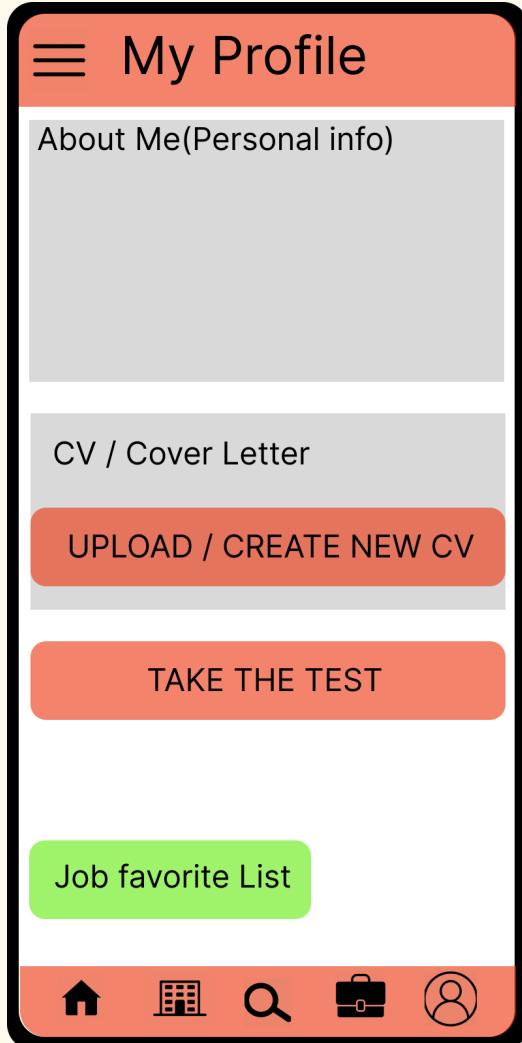
**Highlight Your Expertise:** Create a comprehensive profile that highlights your professional expertise, education, work experience, certifications, and any other relevant achievements. Make a strong first impression that leaves a lasting impact on recruiters and hiring managers.

**Showcase Your Accomplishments:** Use your profile to showcase your career accomplishments and successful projects. Demonstrate your value and potential as a valuable addition to any organization.

**Upload Your Resume:** Upload your updated resume to provide additional insights into your skills and experiences. Employers can easily view and download your resume directly from your profile.

**Personalize Your Pitch:** Craft a compelling summary or objective that introduces yourself and explains your career goals. This personalized pitch helps employers understand your motivations and aspirations.

**Manage Your Applications:** Keep track of all your job applications in one place. View the status of each application, and receive notifications when employers respond to your submissions.



## 5. Create New CV

### Description:

**Personal Information:** Provide your essential details, such as your full name, contact information, and professional photo (optional). This information helps employers get to know you better.

**Work Experience:** List your previous work experiences, including job titles, company names, employment periods, and key responsibilities. Showcase your contributions and achievements in each role.

**Education and Qualifications:** Highlight your educational background, including degrees, certifications, and any relevant training programs. Display your academic achievements and specializations.

**Skills and Competencies:** Present a comprehensive list of your skills and competencies. Include both hard skills (e.g., programming languages, software proficiency) and soft skills (e.g., communication, leadership).

**Projects and Accomplishments:** If applicable, describe notable projects you've worked on or significant accomplishments throughout your career. Provide details about your role and the impact of your contributions.

**Additional Sections (Optional):** Tailor your CV to your specific strengths by including optional sections such as publications, awards, languages spoken, and professional affiliations.

The image shows three mobile application screens for creating a new CV, arranged horizontally. The first screen is a navigation bar labeled "My Profile" with a red header and a black border. It contains a section titled "About Me(Personal info)" with a grey background, a button "UPLOAD / CREATE NEW CV" in a red box, a button "TAKE THE TEST" in a red box, a green progress bar labeled "Creating profile", a button "CREATE NEW CV" in a red box, and a button "UPLOAD YOUR CV" in a red box. The second screen is titled "← Create New CV" with a red header and a black border. It has two input fields: "CV Name" and "Language", both with grey backgrounds. At the bottom are buttons "START MAKING CV" and "CANCEL". The third screen is also titled "← Create New CV" with a red header and a black border. It lists several input fields with grey backgrounds: "Fullname", "Email", "Phone", "Address", "Birthday", "Skill", "Certification", "Education", "Work experiences", and "Summary". At the bottom are buttons "SAVE" and "PREVIEW".

Input:

**Personal Information:** The candidate provides personal details, such as their full name, contact information (phone number, email address), address, and optionally, a professional profile picture.

**Education Details:** The candidate inputs information about their educational qualifications, including school/university names, degrees earned, majors, and graduation dates.

**Work Experience:** The candidate lists their work history, including job titles, company names, employment dates, job responsibilities, and any notable achievements.

**Skills:** The candidate mentions their relevant skills, both technical and soft skills, that are valuable in their desired job field.

**Certifications:** If applicable, the candidate includes any certifications or licenses they have obtained that can boost their qualifications.

**Projects or Portfolio (Optional):** Candidates can showcase their projects, portfolio, or links to their professional website or GitHub repositories to demonstrate their expertise.

**Career Objective or Summary (Optional):** Candidates may add a brief summary or career objective statement to highlight their career goals and aspirations.

#### **Output:**

**CV Preview:** After the candidate inputs their information, the createCV() function generates a live preview of the CV, allowing the candidate to view how their CV will appear to potential employers. This preview enables candidates to make real-time adjustments and ensure the CV looks professional and well-organized.

**Save or Download Options:** Once the candidate is satisfied with their CV, they can choose to save it directly on the platform for future editing or download it as a PDF or other standard document format. The download option allows candidates to share their CV with prospective employers or print it for offline use.

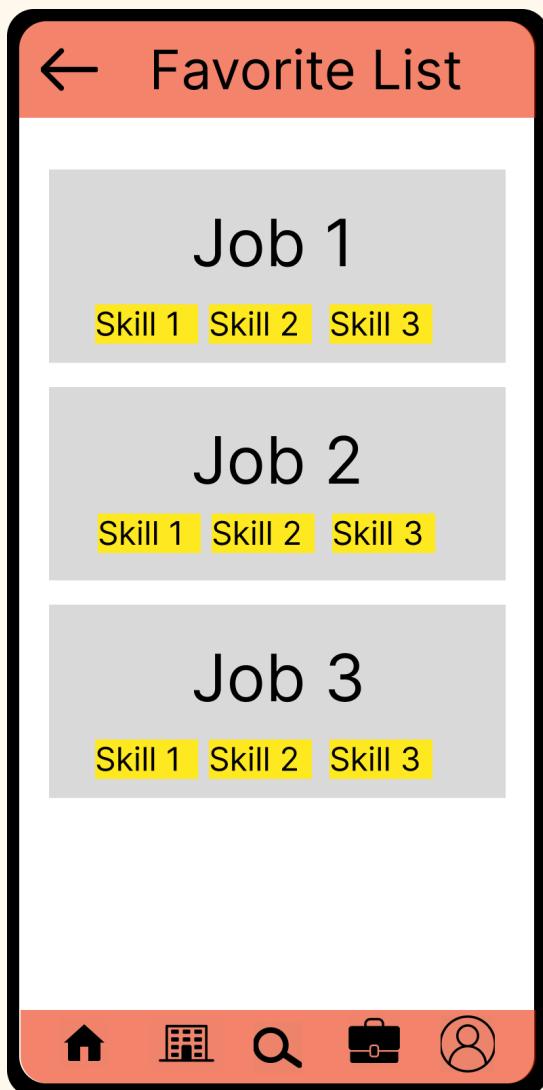
## 6. Jobs Favorite List

**Save Preferred Jobs:** As you browse through job listings, you can easily mark positions that catch your interest as favorites. Click the "Add to Favourites" button on any job listing, and it will be added to your job favorites list.

**Track and Compare Opportunities:** Access your job favorites list to review the job details of each position. Compare different opportunities side by side to make informed decisions about which roles align best with your career goals.

**Application Prioritization:** Prioritize your job applications by focusing on the positions saved in your favorites list. Tailor your application materials to these opportunities to maximize your chances of success.

**Organize Your Job Search:** With a centralized favorites list, you can keep your job search organized and efficient. No more searching through multiple listings to find the ones you were interested in!





## 7. Take The Test

**Skill Validation:** By taking tests related to your field of expertise, you can validate your skills and knowledge to potential employers. This helps you stand out as a qualified candidate.

**Expand Your Profile:** Upon completing a test, the results and certifications can be added to your profile. This showcases your dedication to continuous learning and your commitment to professional growth.

**Increased Visibility:** A strong performance on tests can improve your ranking in search results, making you more visible to employers looking for candidates with specific skill sets.

**Employer Attraction:** Showcasing your test results can attract employers who are actively seeking candidates with your skill set, increasing your chances of getting noticed.



## 8. All Jobs List.

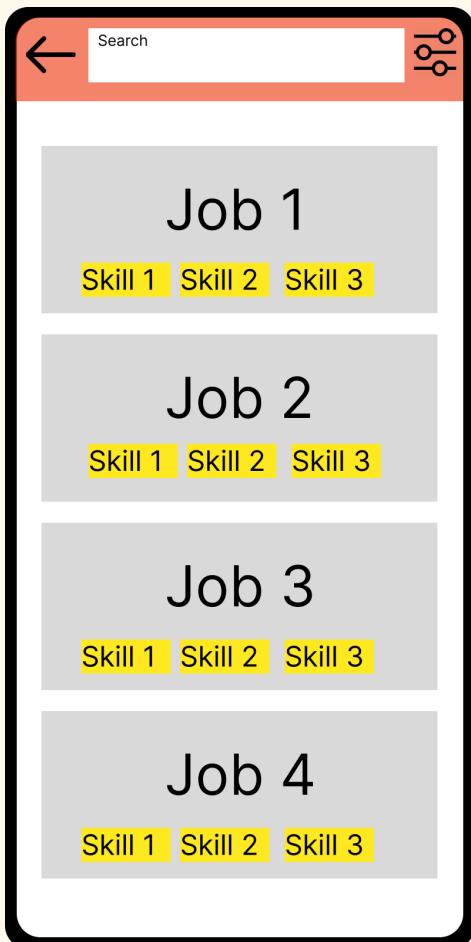
**Description:** Candidates can use this feature to browse through a diverse range of opportunities, while employers can gain exposure for their job openings and connect with potential candidates.

**Extensive Job Selection:** Access an extensive list of job opportunities across various industries, companies, and locations. Whether you're a fresh graduate or an experienced professional, you'll find a wide range of roles to explore.

**Filter and Sort:** Utilize our advanced search filters to narrow down the job listings based on your preferences. Sort by location, job type, experience level, and more to find the most relevant positions.

**View Detailed Job Descriptions:** Click on any job listing to view its detailed description, including job responsibilities, qualifications, company information, and application instructions.

**One-Stop Destination:** The "All Jobs List" serves as a one-stop destination for your job search needs, providing a holistic view of available opportunities on [Your Job Board Name].



## 9. Job Details.

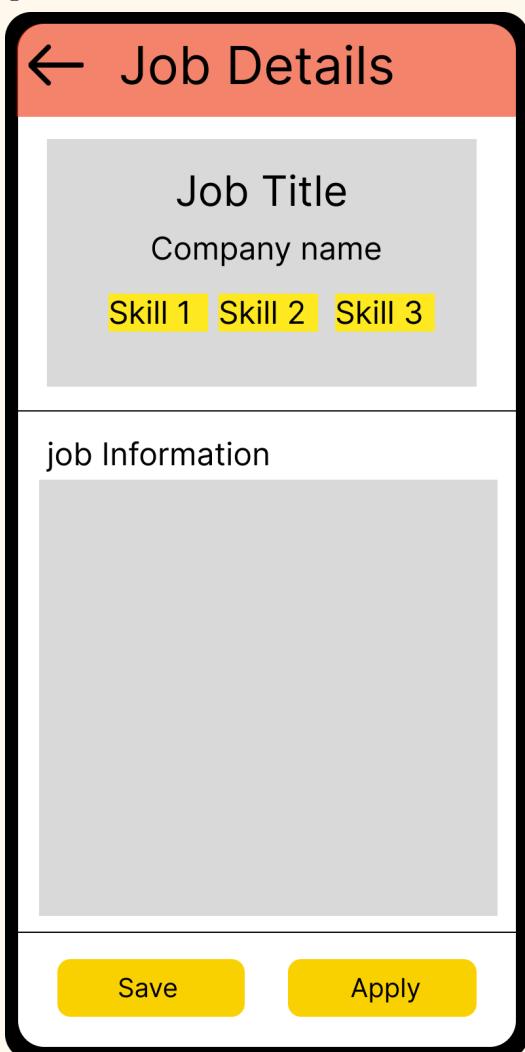
### Description:

**In-Depth Job Descriptions:** Click on any job listing to access a detailed job description. Learn about the role's responsibilities, required qualifications, and any specific skills or experience desired by the employer.

**Company Insights:** Gain valuable insights into the hiring company, including its mission, culture, values, and any perks or benefits they offer to employees.

**Application Instructions:** Find clear and straightforward application instructions, such as the preferred method of application, submission deadlines, and contact details for inquiries.

**Similar Job Recommendations:** Discover related job listings based on the position you're viewing. This feature helps you explore additional opportunities that match your interests and qualifications.



## 10. Job Apply

**Description:** Applying to job listings is made simple, efficient, and convenient through this feature, ensuring that your applications reach potential employers with ease.

**Seamless Application Process:** The "Job Apply" function streamlines the application process. Apply to job listings directly through [Your Job Board Name] without the need for external emails or application portals.

**Tailor Your Application:** Craft personalized cover letters and messages to showcase your genuine interest in the position and demonstrate how your skills align with the job requirements.

**Track Your Applications:** Keep track of all the job listings you've applied to in one organized dashboard. Monitor the status of each application and stay updated on potential responses.

**Upload Supporting Documents:** Easily upload your CV, resume, portfolio, or other supporting documents to provide employers with a comprehensive view of your qualifications.

← Apply To

Job Title  
Company name

Skill 1 Skill 2 Skill 3

Upload your CV (.doc, .docx, .pdf, ... <5mb)

Select your CV

CV 1: .....

CV 2: .....

Fullname

Email

Phone

Confirm Apply

## 11. Company List

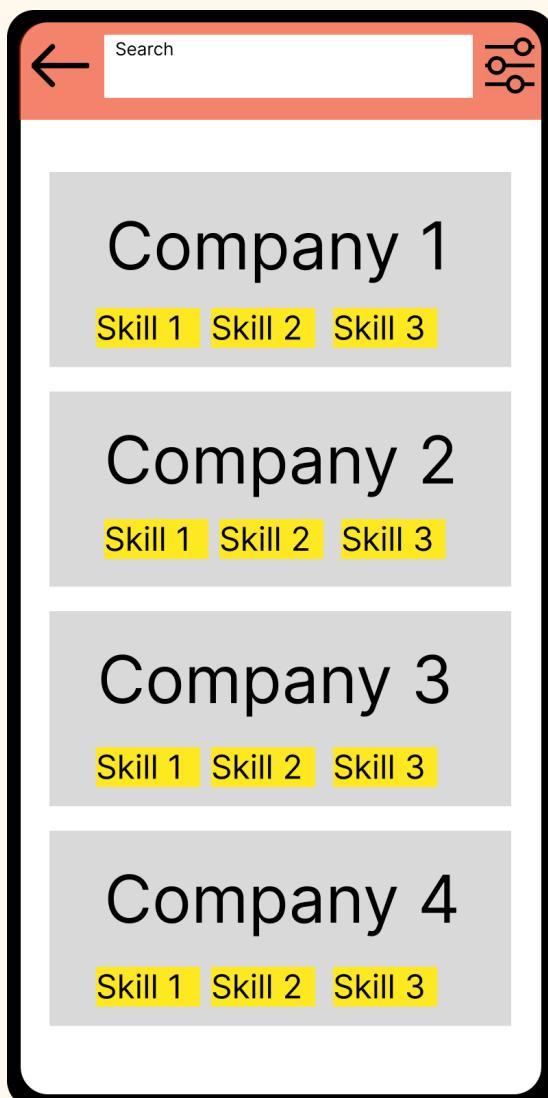
**Description:** Get insights into the company's profile, culture, values, and job opportunities they offer. This feature empowers job seekers to make informed decisions about their potential employers.

**Company Profiles:** Access detailed profiles of various companies hiring through [Your Job Board Name]. Learn about their history, mission, size, and industry presence.

**Company Culture:** Gain valuable insights into a company's culture, work environment, and employee benefits. Find companies that align with your values and professional aspirations.

**Job Opportunities:** Explore the job openings posted by each company on the platform. Easily navigate through positions that match your skills and career goals.

**Follow Companies:** Stay updated on companies you're interested in by following them. Receive notifications about new job listings, company news, and events.



## 12. Company Details

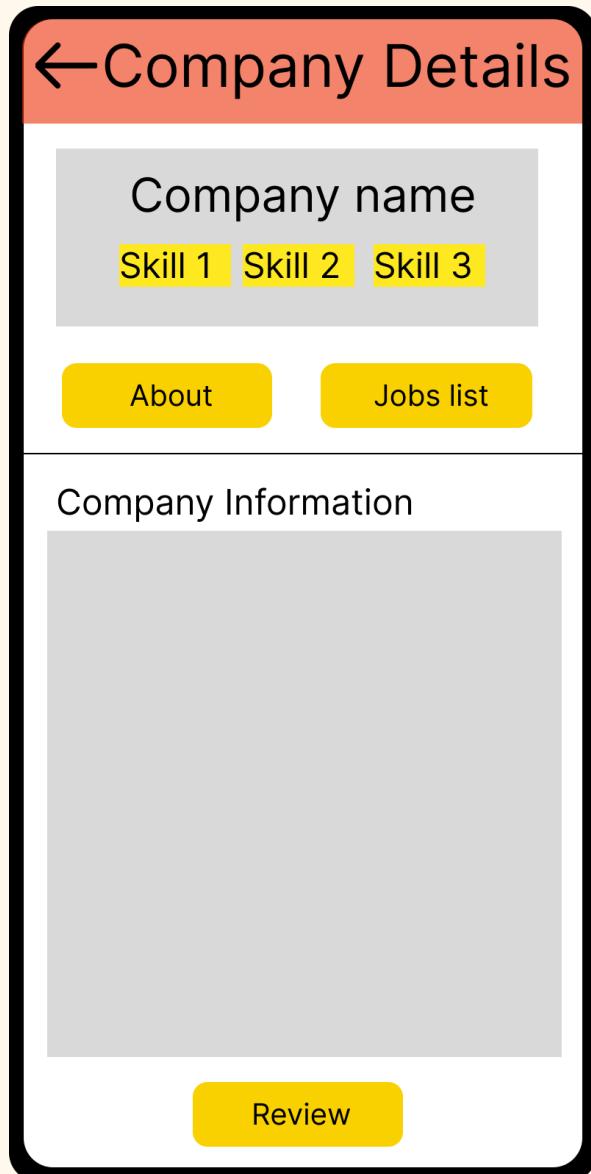
**Description:** gain valuable insights into a company's history, culture, work environment, and available job opportunities.

**Company Overview:** Access comprehensive profiles of companies hiring through [Your Job Board Name]. Learn about their mission, vision, values, and the industries they operate in.

**Company Culture:** Get a glimpse into a company's work culture, employee benefits, and the opportunities they offer for career growth and development.

**Job Openings:** Find out about the job opportunities posted by the company on our platform. Easily navigate through the available positions that match your skills and interests.

**Follow Companies:** Stay updated on companies you're interested in by following them. Receive notifications about new job listings, company news, and events.



### 13. Review Company/Job

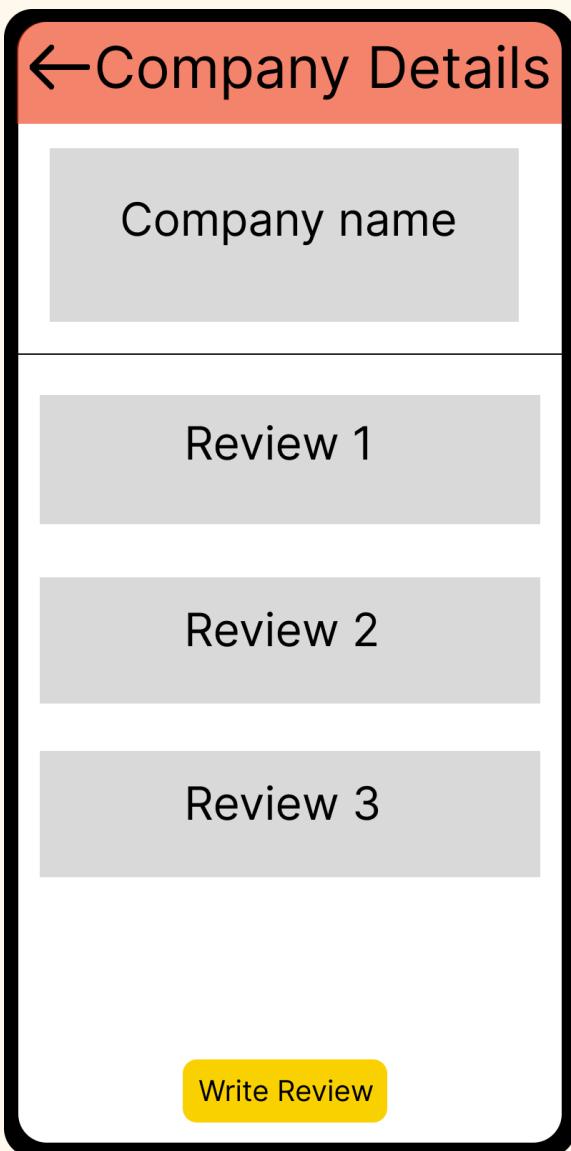
**Description:** help others make informed decisions about potential employers and job opportunities.

**Honest Feedback:** Leave honest and constructive reviews about your experiences with companies you have worked for or job positions you have applied to. Your insights can guide other job seekers in their career decisions.

**Company Culture:** Share your thoughts on the company's work culture, work-life balance, management, and employee benefits. This helps others gauge if the company is the right fit for them.

**Job Position Insights:** Provide valuable information about the job position you applied for, such as the job responsibilities, growth opportunities, and the application process.

**Anonymous Reviews:** Choose to leave anonymous reviews to express your thoughts freely while maintaining privacy.



**Input:**

**user\_id:** A unique identifier for the user who is leaving the review. This is used to associate the review with the correct user profile.

**company\_id:** A unique identifier for the company being reviewed. This helps to link the review with the specific company profile.

**job\_id (optional):** An optional unique identifier for the specific job position being reviewed. This associates the review with the relevant job listing, if applicable.

**rating:** The numeric rating (e.g., on a scale of 1 to 5) given by the user to the company or job position. This represents the user's overall satisfaction or experience.

**review\_text:** The textual content of the review provided by the user. This contains the user's detailed feedback, insights, and comments.

**anonymous (optional):** A boolean flag indicating whether the user wants to submit the review anonymously. If true, the user's identity is kept confidential.

**Output:**

**Success/Failure Indication:** The function should provide feedback on whether the review submission was successful or if there was an error in the process.

**Review Moderation:** If the platform requires review moderation, the function may return a message indicating that the review is pending approval before being displayed to other users.

**Updated Company/Job Data:** If the review submission is successful, the function might trigger updates to the overall rating or review count for the company or job position, helping other users see the latest feedback.

**Employer Response:** If employers can respond to reviews, the function may display any responses submitted by the company in reaction to the review.

### 3.4.4 DESCRIPTION OF DATA PROCESS, APPLIED ALGORITHM (IF ANY)

- **Data Collection:** The first step is to collect data from various sources. In the context of a job board website, this may involve gathering job postings from employers, user profiles, resumes, and any other relevant information.
- **Data Cleaning:** Once the data is collected, it needs to be cleaned and preprocessed. This involves removing any irrelevant or duplicate entries, handling missing data, and converting data into a consistent format.
- **Data Storage:** The cleaned data is then stored in a database or data warehouse. For your project, you can use a relational database like MySQL or PostgreSQL to store information related to job listings, user profiles, and other relevant data.

- Data Retrieval: When users interact with the web site, the back end will retrieve the required data from the database based on their queries and requests. For example, when a user searches for job openings, the back end will fetch the relevant job postings based on the search criteria.
- Data Processing and Analysis: Depending on the complexity of your project, you might need to perform some data processing and analysis tasks. For instance, you could use natural language processing (NLP) techniques to extract keywords and skills from job descriptions or resumes. This analysis can help improve search relevance and provide personalized job recommendations.
- Data Presentation: Once the relevant data is retrieved and processed, it's sent back to the front end in a structured format, typically as JSON. The front end then presents this data to the user in a user-friendly manner, such as displaying job listings, user profiles, and other relevant information.

### 3.5 TASK SHEET REVIEW 3

Project Ref.No		Project Title	Activity plan prepared by	Date of preparation of Activity plan				
#	Task			Start date	End date	Actual days	Member's name	Status
1	Install database physical	Job Board Management System	Nguyễn Anh Vũ	2023-07-09	2023-07-12	3	Nguyễn Anh Vũ	Done
2	Site map			2023-07-09	2023-07-11	2	Nguyễn Anh Vũ	Done
3	Constructor project directory			2023-07-09	2023-07-15	6	Nguyễn Gia Thiện	Done
4	Design GUI Figma			2023-07-09	2023-07-16	7	Phạm Nhật Minh	Done
5	Function descriptions			2023-07-09	2023-07-16	7	Nguyễn Gia Thiện	Done
6	Input / output data analyst			2023-07-09	2023-07-16	7	Phạm Nhật Minh	Done
7	Description of data process			2023-07-09	2023-07-15	6	Nguyễn Anh Vũ	Done

Signature of instructor	Signature of Team Leader
Mr. Trần Phước Sinh	Mr. Nguyễn Anh Vũ