

## **DEPARTMENT OF SOFTWARE ENGINEERING**

### **SE-202, OBJECT ORIENTED SOFTWARE ENGINEERING LABORATORY MANUAL**

#### **CASE STUDY: Job Search Portal**

SUBMITTED BY:

ABHINAV CHAITANYA (2K21/SE/07)

AMRITI GUPTA (2K21/SE/29)

UNDER THE SUPERVISION OF:

PROF. RUCHIKA MALHOTRA



**DELHI TECHNOLOGICAL UNIVERSITY**

**BAWANA ROAD, DELHI- 110042**

## **INDEX**

<b>S.No.</b>	<b>Title</b>	<b>Date</b>	<b>Page No.</b>	<b>Signature</b>
1.	Problem Statement	19 <sup>th</sup> January 2023	3	
2.	Initial Requirement Document	2 <sup>nd</sup> February 2023	4	
3.	Software Requirement Specifications	9 <sup>th</sup> February 2023	5	
4.	Test Case Matrix	20 <sup>th</sup> April 2023	40	

## **PROBLEM STATEMENT**

Job Portals are like the meeting points for the recruiters as well as the job seekers where both aim at meeting their individual requirements. Job seekers try to find a job opportunity where they can apply their knowledge, acquire new skills, and grow as a professional. On the other hand, recruiters try to fill their job openings with the right candidate who has the perfect aptitude and qualification to handle the responsibilities efficiently. In recent times, job search has shifted from offline to online mode. Websites have made the process less hefty for both the recruiters and the job-seekers. Several job search portals are available these days which provide a variety of facilities. However, there are certain limitations that exist in these websites which are as follows:

- No proper feedback through the portal after the employer viewed the profile.
- Not all the portals have a cover letter in the job application.
- Lengthy registration for Job applications
- No tips available for job seekers
- Duplication of jobs - Sometimes job seekers could not reach the employers who originally posted the job.
- No proper contact page for seeking customer support

So, we are proposing a job search website. Our website aims to resolve the above issues. The main features of our website are as follows:

- Recruiter-
  1. Create a vacancy
  2. Remove vacancy
  3. View applications
  4. Set the application status of the candidate
- Candidate
  1. Add skills, resume, and interests
  2. View/ search jobs with filters
  3. View the recruiter profile
  4. Apply for a job
  5. View applications and status
- Tips and suggestion blogs for candidates
- FAQs and Contact

## **INITIAL REQUIREMENT DOCUMENT**

<b>Title of the Project</b>	Job search portal
<b>Stakeholders involved in capturing requirements</b>	Job Seekers, Recruiters, Developers, Administrator
<b>Techniques used for requirement capturing</b>	Interviewing and Brainstorming
<b>Name of persons along with designation</b>	Amriti Gupta , Abhinav Chaitanya
<b>Date</b>	February 2023
<b>Version</b>	1.0

### **Consolidated list of requirements-**

1. A system is to be implemented which can be deployed on the web which can be used on personal computer or mobile with an internet connection.
2. There are two types of members:- Job Seeker and Job Recruiter.
3. The portal shall generate a unique job-ID for every job posted by a recruiter.
4. The portal shall generate a unique application-ID for each vacancy that the job-seeker applies for.
5. The recruiter can create a vacancy with proper details for the role.
6. The recruiter can set the status of an application to waitlisted, rejected, or selected.
7. The job seeker can filter the jobs based on location, stipend/salary, experience, sectors etc.
8. The job seeker can upload his/her resume and CV and add his skills.
9. The job seeker is given a provision to check his account information in the dashboard and change the account information.
10. The job seekers are provided with the jobs available roster and allowed to apply for the job they want.
11. The job seeker can check his previous applications and his application status from the dashboard.
12. The job seeker can also withdraw any application whenever required.
13. Provisions for the job seekers to apply for the job they want, if all the other required rules hold good.
14. Admin can add the tips and suggestions blog which can be accessed by the job seekers.
15. After applying for a job, a confirmation email is sent to the job applicant on his/her registered email ID.
16. The job-seekers and recruiters can view the code of conduct and the administrator contact.
17. The portal should have a landing page that redirects the users for login and register.

## **Software Requirement Specification**

### **Structure of SRS-**

1.	Introduction 1.1 Purpose 1.2 Scope 1.3 Definitions, acronyms, and Abbreviations 1.4 References 1.5 Overview
2.	Overall description 2.1 Product perspective 2.1.1 System interfaces 2.1.2 User interfaces 2.1.3 Hardware interfaces 2.1.4 Software interfaces 2.1.5 Communications interfaces 2.2 Product functions 2.3 User characteristics 2.4 Constraints
3.	Specific requirements 3.1 External interfaces 3.2 Functions 3.3 Performance requirements 3.4 Design constraints 3.5 Software system attributes 3.5.1 Reliability 3.5.2 Availability 3.5.3 Security 3.5.4 Maintainability 3.5.5 Portability 3.6 Logical database requirements 3.7 Other Requirements
4	Appendix 4.1 Use Case Diagram 4.2 Class Diagram 4.3 Sequence Diagrams 4.4 Activity Diagrams 4.5 Statechart Diagrams

## **1. Introduction**

### **1.1 Purpose**

The job search portal is a web application to facilitate the process of job search for the job seeker and hiring for the recruiter. It maintains the details of recruiters and seekers who are registered on the portal and the details of their applications.

### **1.2 Scope**

The name of the software is getToWork. The proposed software must be able to perform the following functions:-

#### **Dos**

1. Issue of Login ID and password to system operators.
2. Maintain details of the users(job seekers and job recruiters) registered in the system.
3. Maintain details of the jobs posted by recruiters in the system.
4. Maintain details of the job applications in the system.
5. Post a job
6. Apply for a job
7. Search for a job, using the required filters and keywords

#### **Don'ts**

1. Blogs section is not covered.
2. Interview or interaction facility through the portal is not covered.
3. Premium facility is not available and the software is free to use.

#### **Benefits**

The portal provides the following benefits:

1. Ability to search for a job using filters and keywords hence easy searching of jobs
2. Efficient way of posting and applying for a job with email notifications.
3. Ability to maintain the application details in a concise manner.
4. Provision to upload CV and resume.

### **1.3 Definition, Acronyms, and Abbreviations**

- SRS - Software Requirement Specification
- System operator - System administrator, software developers, and staff

- RAM - Random Access Memory
- System administrator / Admin – The user who has all the privileges to operate the portal
- Job Recruiter- Users who are posting the job
- Job Seeker - Users who are applying for the job
- FAQ(s)- Frequently Asked Question(s)

#### **1.4 References**

- A. IEEE Standard for Software Test Documentation - IEEE Std. 829-1998.
- B. IEEE Recommended Practice for Software Requirements Specifications - IEEE Std. 830-1998
- C. Software Engineering by K.K. Aggarwal & Yogesh Singh, New Age Publishing House, 3rd Edition, 2008.

## **2. Overall Description**

The online job search portal (getToWork) is an application to facilitate the process of job search, application, and recruitment by maintaining the details of job-seekers and recruiters who are registered on the website. The job-seekers and recruiters can create an account on the portal with a valid email-id, username, and password. Thereafter the user must log in to the portal for accessing the features and accessibilities provided to them by the portal.

Recruiters can create job openings on the portals together with the details of the job for information to the applicants. The job-seeker can view or search for a particular job with the set of filters provided on the search page or otherwise. Thereafter the job-seeker shall apply for the post by attaching a resume, cover letter, and the personal details that are asked for while applying, through the portal. The recruiters shall thereafter, review the applications and contact the applicants (job-seekers) by using the contact details provided by the applicants or setting the status of the application with the company. The users can also manage their personal details on their profile page on the portal.

The landing page of the portal is open for all to view and shall contain FAQs and

The system administrator can maintain the following information-

- Vacancy details
- Application details

The job-seeker will perform the following functions-

- Create an account on the portal
- Login and logout
- View / update his profile.

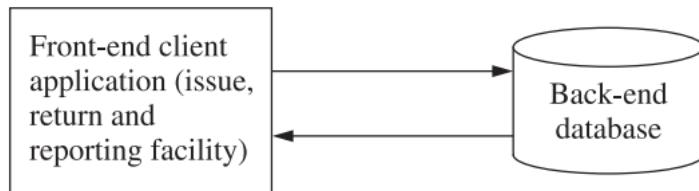
- Search for jobs
- Apply for a job
- View past applications
- View FAQs and Admin contact

The recruiter will perform the following functions-

- View/update company profile.
- Create, update, or delete a job opening/vacancy
- View applications and set application status
- View FAQs and Admin contact
- View job openings created in the past

## 2.1 Product Perspective

The website shall be developed using a client-server architecture and will be compatible with all operating servers. The front-end of the system shall be developed with HTML, CSS, JavaScript (both vanilla and es6), Bootstrap, and React.js. The back-end will be developed using Node.js, Express.js, and MongoDB (NoSQL) database. The website shall be deployed on the web on Cyclic or Renderz servers and the back-end shall be supported with MongoDB Atlas. The website can be used on the latest versions of Edge, Google Chrome, Mozilla Firefox, and Brave web browsers. The use of the website requires an active internet connection.



### 2.1.1 System Interfaces

The system shall require MongoDB Atlas as a cloud-based database which will be constantly involved in querying via the front-end. The use of NPM (Node package manager) packages is expected as the Node dependencies of the project.

### 2.1.2 User Interfaces

The getToWork will have the following user-friendly and menu-driven interfaces-

1. **Registration** – To allow new users to join the portal by creating an account for themselves by filling in the required details.
2. **Login** - to allow the entry of only authorized users through a valid email ID and password.
3. **Vacancy details** – to maintain vacancies created by a recruiter
4. **User details** – to maintain user details

5. **Application details** – to maintain the job applications submitted by the job seekers.
6. **Job search** – to allow job seekers to search for jobs.
7. **Apply for a Job** – to allow job seekers to apply for jobs.

#### **2.1.3 Hardware Interfaces**

- a) Screen resolution of 280 x 640 pixels or more.
- b) Stable internet connection on the device of use.

#### **2.1.4 Software Interfaces**

- a) Latest version of internet browser (Chrome, Mozilla Firefox, Edge, Brave).
- b) HTML, CSS, JS, React.js, and Bootstrap to develop front-end.
- c) MongoDB atlas, Node.js and Express.js for backend.

**2.1.5 Communication Interfaces** – the device must have an internet connection.

**2.1.6 Memory constraints-** The online course website should be designed to handle many users simultaneously. The website should be optimized for speed and should not have any memory constraints that would limit its performance.

#### **2.1.7 Operations-**

The website should be easy to maintain, with regular updates and patches to fix any security vulnerabilities or bugs. The website should also be scalable, allowing for the addition of new features and functionality as the website grows.

**2.2 Product Functions** – The summary of major functions that getToWork performs is-

- Allow users to create an account on the portal.
- A log in facility for enabling only authorized access to the system.
- Allows users to edit their profile.
- Allows recruiters to create vacancies as and when required.
- Allows job seekers to search and apply for a job or withdraw their application.
- Allows all users to view the FAQs and Admin contact.

#### **2.3 User Characteristics –**

- User must have basic knowledge of using the device with the internet connection and the latest version of the browser.

#### **2.4 Constraints –**

- The user will not be allowed to update the primary key.
- There shall be only one admin in the system.
- Admin has the right to delete accounts and related information created by the user on the system.

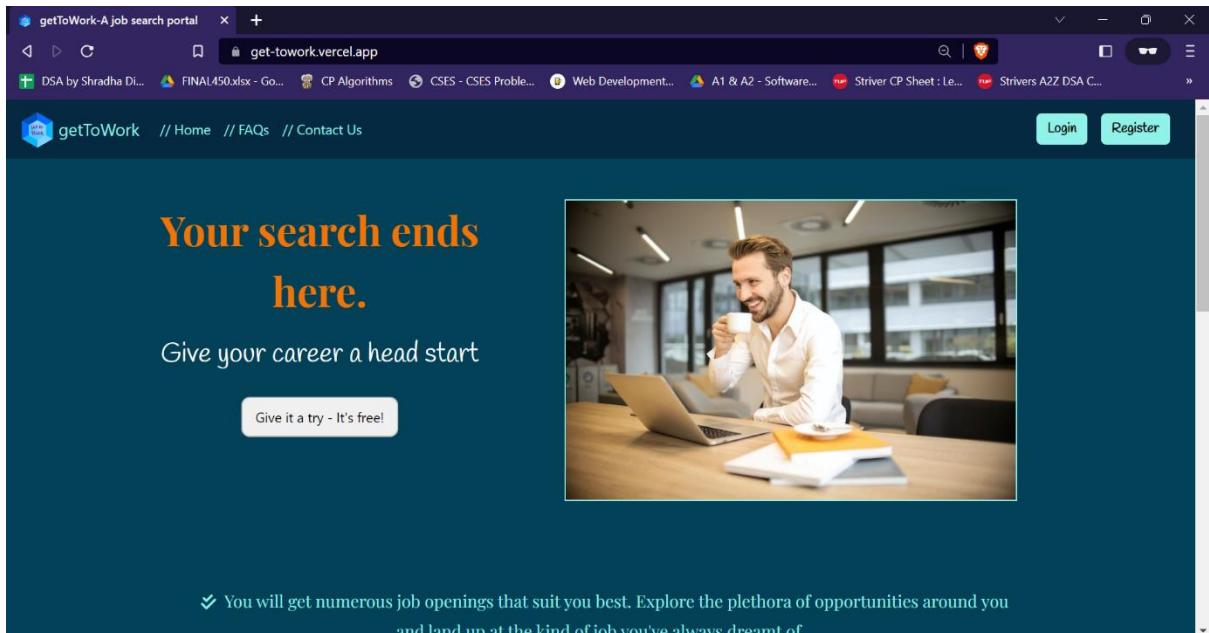
### **3. Specific Requirements-**

#### **3.1 External Interfaces-**

### 3.1.1 User Interfaces

The following user interfaces shall be provided by the system.

- a) **Landing Page-** This is the first page that a user encounters on arriving at the website. It provides the option for a user to register (which means creating a fresh account for a new user) and the option of login in for already registered users. It contains the basic information about the website and the FAQs and Admin Contact, which are accessible on this page itself.



- b) **Register-** This page is meant for creating a new account and contains a form that gives the option to register as a recruiter or as a job seeker. The user selects the option and is asked for credentials. For Recruiter the fields to be filled are-

- Company Name – Alphanumeric (more than 3 characters and no acronyms recommended).
- Email address – alphanumeric with an @ sign compulsory as a means of validation.
- Password- alphanumeric of minimum 6 characters. Special characters are allowed.
- Confirm Password – a field meant for validation. The password same as above is expected to be entered here.

For a Job seeker, the fields to be filled are-

- Username – Alphabetical ( more than 3 characters required).
- Email address – alphanumeric with an @ sign compulsory as a means of validation.
- Password- alphanumeric of minimum 6 characters. Special characters are allowed.

- Confirm Password – a field meant for validation. The password same as above is expected to be entered here.

The screenshot shows the 'SignUp' page of the 'getToWork' job search portal. The page features a dark teal header with the 'getToWork' logo and navigation links. Below the header is a large white 'SignUp' button. A form is centered on the page with fields for 'Select User Type \*', 'Your Name \*', 'Username \*', 'Email-ID \*', and 'Password \*'. Each field has its validation message below it. At the bottom is a teal 'SignUp' button.

c) **Login** – This page is meant for users' login to the system via authentication. Existing users shall select the option to log in as a recruiter or job seeker depending on the type of account they have created and are asked for the email ID and password they have entered at the time of registration.

- Email ID – alphanumeric with an @ sign compulsory as a means of validation.
- Password- alphanumeric of minimum 6 characters. Special characters are allowed.

Then the user is provided the button to log in and enter the system.

The screenshot shows the 'Login' page of the 'getToWork' job search portal. The page features a dark teal header with the 'getToWork' logo and navigation links. Below the header is a large white 'Login' button. A form is centered on the page with fields for 'Select User Type \*', 'Registered Email-ID \*', and 'Password \*'. At the bottom is a teal 'Login' button.

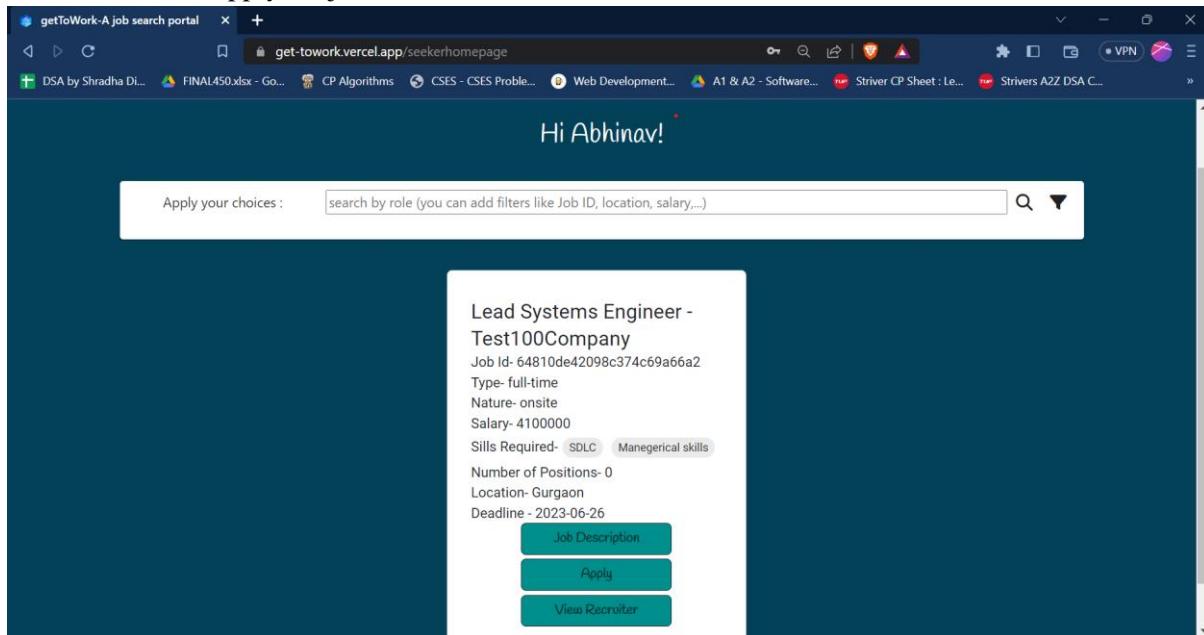
d) **Home Page** – Once the user is successfully logged into the system the user is given an option according to the type of account.

For a recruiter

- an option is provided to view/Edit the vacancies created.
- Edit company profile.

For a job seeker-

- Search Jobs
- View/ Edit profile
- Apply for jobs



e) **Maintain Vacancy details-** Upon selection of option vacancies created option on the recruiter home page, the new page to maintain vacancy details appears. On this page, the recruiter is given the option to-

Create a vacancy –

- Name of Role – alphabets and parentheses allowed.
- Job description – alphanumeric, maximum ( 150 words allowed).
- Salary – numeric and currency special characters.
- Skills required – list
- Location – alphabetical, list
- Duration – Time (alphanumeric)
- Number of vacancies – numerical

- Deadline – Date(DD/MM/YY)

Create a vacancy (We request you to avoid confusing terms and phrases).

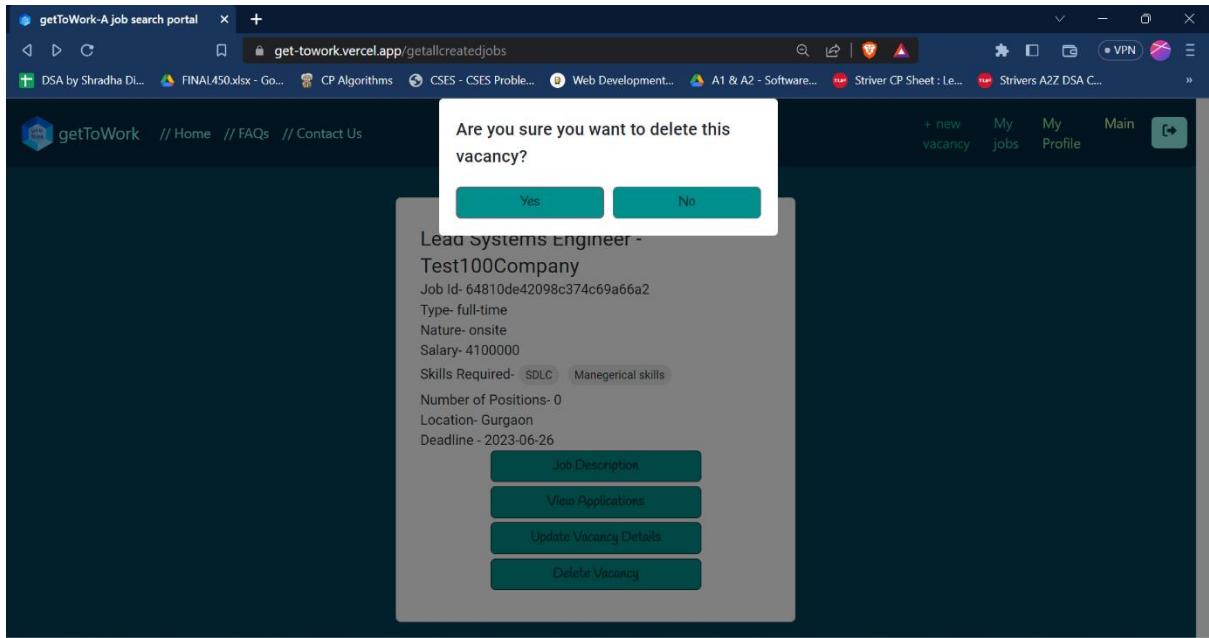
Role*:	<input type="text"/>
Job Type*:	<input type="text"/> Select one
Nature*:	<input type="text"/> onsite,remote or hybrid ...
Annual salary (in INR)*:	<input type="text"/>
Duration:	<input type="text"/> if applicable
Skills-	
Type and press enter <input type="text"/>	
Deadline for applications:	<input type="text"/> dd-mm-yyyy
Number of positions: <input type="text"/>	

Update vacancy – the above fields with their previously filled values appear. The changes can be made and the recruiter is given the option to either confirm the changes made or exit the update window.

Lead Systems Engineer -  
Test100Company  
Job Id- 64810de42098c374c69a66a2  
Type- full-time  
Nature- onsite  
Salary- 4100000  
Skills Required-  SDLC  Managerical skills  
Number of Positions- 0  
Location- Gurgaon  
Deadline - 2023-06-26

[Job Description](#)  
[View Applications](#)  
[Update Vacancy Details](#)  
[Delete Vacancy](#)

Delete vacancy- The user is given the option to delete a particular vacancy from the list of vacancies created by the recruiter. The recruiter is given the option to confirm or reject deletion.



f) **Maintain user details-**

Upon selection of profile option on the homepage.

For recruiter, the fields to fill or edit are-

- Company name - alphabetical
- Recruiter name - alphabetical
- Contact details- email ID (alphanumeric and special characters) and phone number (numeric)
- Company Size- numeric
- Description- alphanumeric and punctuation marks (maximum 150 words)
- Recruiter ID- (proof in form of ID provided by the company)

Your Profile

Name Of Company	Test100Company
Recruiter's Name	Test100
Email ID	test100@gmail.com
Company Size	10-100
Employer Description	A great place to work at. Meet the best minds from the industry and work on real-world problems to make lives easier & interesting.
Website	www.test100companies.com

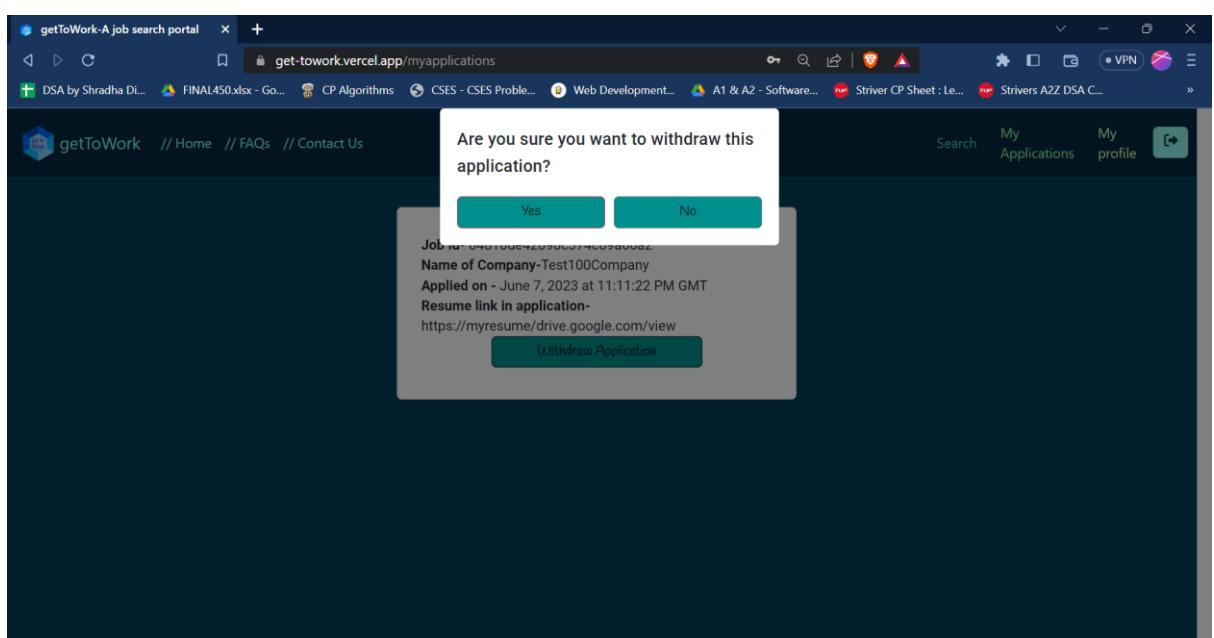
**Edit Profile**

For job seekers the fields to fill or edit are-

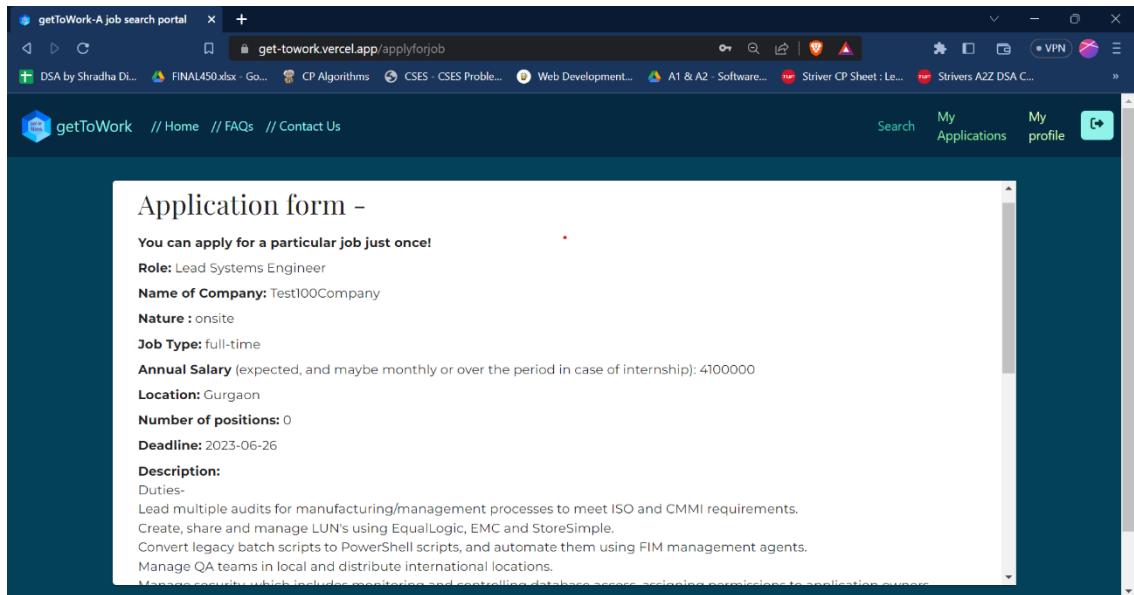
- Name – Alphabetical
- Date of birth – date (DD/MM/YY)
- Highest level of education-alphanumeric
- Add resume – file (pdf format)
- Add photo – image (jpg, png)
- Skills- alphanumeric
- Marital Status- Boolean
- Gender – option

#### g) Maintain Application details-

Inside the profile of the job seekers, he/she gets the list of jobs applied for. The job seeker has the option to withdraw the application. The job seeker gets the option to confirm the withdrawal.



**h) Apply for a job-** on the homepage of the portal for the job seeker searching for a job we get the list of available jobs for that search. Each of them has the option to apply if the job-seeker has not already applied for it. On clicking the apply button a confirmation prompt is generated.



## 3.2 Functions

### 1. Registration

#### A. Use Case Description

##### **Introduction:**

This use case documents the steps that the job-seeker or recruiter must follow to create an account on the job-search portal.

##### **Actors:**

Recruiter

Job-seeker

##### **Precondition:**

The job seeker must be present on the landing page of the portal.

##### **Postcondition:**

If the use case is successful the job-seeker/ recruiter is redirected to the corresponding homepage.

##### **Flow of events:**

###### **Basic flow**

- The use case starts when actor chooses the option to register on the landing page of the portal.
- The actor must choose whether he wants to register as a job seeker or recruiter.
- After the actor chooses the aforesaid option he is provided the option to enter the username, password, and a unique email-id. After the actor submits the form, the portal validates the entered details and creates an account for the actor.

###### **Alternate flow**

- **Alternative Flow 1: Incorrect/Incomplete details:**

If in the basic flow, the actor enters already existing email-id or username or leaves a field empty the actor is redirected to the start of the basic flow and an error message is displayed.

- **Alternative Flow 2: User Exists:**

exit the system.

**Special requirements:** none

**Related use cases:** none

#### B. Validity Checks-

- The email and username entered must be unique
- No field should be left blank
- Email entered must be valid
- Password must be at least 5 characters long

## 2. Login

#### A. Use Case Description

##### **Introduction:**

This use case documents the steps that the job-seeker or recruiter must follow to access their account on the job-search portal.

##### **Actors:**

Recruiter

Job seeker

##### **Preconditions:**

Actor must be registered in the system and be present on the landing page of the portal.

##### **Postconditions:**

The actor is logged into his account and corresponding home page for the job-seeker or recruiter is displayed.

##### **Flow of events:**

###### **Basic flow**

- The use case starts when actor chooses the option to login on the landing page of the portal.
- The actor must choose whether he wants to login as a job seeker or recruiter.
- After the actor chooses the aforesaid option he is provided the option to enter the password and his unique email-id. After the actor submits the form, the portal validates the entered details against the registration details.
- If the entered details are correct the actor is logged into his account and corresponding home page for the job-seeker or recruiter is displayed.

###### **Alternate flow**

- **Alternative Flow 1: Incorrect/Incomplete details:**

If in the basic flow, the actor enters already existing or invalid email-id or leaves a field empty or enters wrong password the actor is redirected to the start of the basic flow and corresponding error message is displayed.

- **Alternative Flow 2: Forgot password:**

If the user forgets their password, they can click on the "Forgot password" link, which will initiate the password reset flow.

- **Alternative Flow 3: User Exists:**

exit the system.

**Special requirements:** none

**Related use cases:** none

#### B. Validity Checks-

- The email and username entered must be unique

- No field should be left blank
- Email entered must be valid
- Password must be at least 5 characters long

### 3. FAQs and Contact

#### A. Use Case Description

**Introduction:** This use case documents the steps that an actor must follow in order to view the FAQs and contact information.

**Actors:**

Job Recruiter

Job Seeker

Any internet User

**Precondition:**

The actor must be at the landing page.

**Postcondition:**

FAQs and contact details are presented on the website.

**Flow of Events:**

**Basic flow-**

The use case starts when the actor clicks on the FAQs and contact button on the landing page. System directs to the page which lists the contact information and frequently asked questions(FAQs).

**Alternate Flow-**

**Alternative Flow 1: User exits**

The user closes the job portal website.

**Special Requirements:** None

**Associated Use Cases:** None

### 4. Logout

#### A. Use Case Description

**Introduction:**

This use case documents the steps to be followed by an actor to end their current session and secure their personal information.

**Actors:**

Recruiter

Job seeker

**Preconditions:**

The student must be logged into the portal.

**Postconditions:**

The actor's session is terminated, and their personal information, such as their progress in the course, is secured from unauthorized access.

**Flow of Events:**

**Basic Flow:**

- The actor selects the logout option present on the webpage.
- The website displays a confirmation prompt, asking the actor if they are sure they want to log out.
- The actor confirms the logout request by clicking "Yes" or "Logout".
- The actor's session is ended, and they are redirected to the login page.

**Alternative Flow:**

**Alternative Flow 1: Session continues:**

If the actor does not confirm the logout request, their session remains active, and the flow resumes to the start of the basic flow of this use case.

**Alternative Flow 2: Actor closes the browser window:**

If the actor closes the browser window without clicking the "Logout" button, their session will end automatically, but their personal information will not be secured. In this case, the actor will need to log in again to continue using the portal.

**Special Requirements:**

None

**Associated Use case:**

Login

## 5. Maintain Vacancy details

### A. Use Case Description

<b>Introduction:</b> This use case documents the steps to be followed for managing the vacancy details by a recruiter for a vacancy created by him. It includes viewing, creating, updating, or deleting a vacancy.
<b>Actors:</b> Recruiter Admin
<b>Preconditions:</b> The recruiter must be logged in into the portal and must have chosen the option to maintain vacancy details.
<b>Postconditions:</b> Once the use case is successful, the vacancy is viewed, created, updated, or deleted in accordance to the options selected in the basic flow of the use case.
<b>Flow of events</b> <b>Basic Flow:</b> The basic flow starts when the recruiter chooses to view, create, update, or delete a vacancy. Portal provides the recruiter with these four options to choose from and one of the sub flows will be executed according to it. <ol style="list-style-type: none"><li>1. <b>Create a vacancy</b><ul style="list-style-type: none"><li>• The portal asks the recruiter to provide the vacancy details such as the name of the role, experience and qualifications required (eligibility criteria), job description and responsibilities, number of openings, nature of work, pay scale, duration of work and deadline for the application.</li><li>• Once the recruiter submits the form the vacancy is created with a unique job-id which shall be visible to both the job seeker and the recruiter for that vacancy.</li></ul></li><li>2. <b>View a vacancy</b><ul style="list-style-type: none"><li>• The portal asks the recruiter to choose the vacancy that he wants to view from the vacancies created by the recruiter.</li><li>• Once the recruiter selects a vacancy to view, the option is provided to the recruiter to view the applications of the candidates and set the status of their candidature as viewed, waitlist or rejected, or to view the information about the vacancy that he had posted.</li></ul></li><li>3. <b>Update a vacancy</b><ul style="list-style-type: none"><li>• The portal asks the recruiter to choose the vacancy that he wants to update from the vacancies created by the recruiter.</li><li>• Once the recruiter makes a choice, he is provided with the fields with the preexisting values to which the recruiter makes changes and clicks on the submit. A confirmation message is prompted to finalize the change. Once the recruiter confirms the change, the vacancy details are updated.</li></ul></li><li>4. <b>Delete a vacancy</b><ul style="list-style-type: none"><li>• The portal asks the recruiter to choose the vacancy that he wants to delete from the vacancies created by the recruiter.</li><li>• Once the recruiter makes a choice, the recruiter is asked to confirm the deletion. Upon confirmation the vacancy is deleted.</li></ul></li></ol>
<b>Alternate Flow:</b> <b>Alternate Flow 1: Update Cancelled / Delete cancelled-</b> If the recruiter chooses not to delete or update the vacancy or denies the confirmation at respective steps the basic flow is restarted from the beginning.

<b>Alternative Flow 2: User Exits:</b> Actor logs out portal or close the browser window.
<b>Special Requirements:</b> None
<b>Associated Use case:</b> Login, Maintain user details

## B. Validity Checks-

- Only administrator and users logged as recruiters have the access to create a vacancy.
- Update, delete vacancy sub-flow shall be successful only upon confirmation via the confirmation prompt on the screen.
- Deadline must be of Date datatype Date rest all the inputs shall be of string datatype.
- Job role, duration, job description , salary and deadline are compulsory fields and can't be left empty.

### 6. Maintain User Details-

## A. Use Case Description-

<b>Introduction:</b> This use case documents the steps to be followed for maintaining the user details by the user himself and involves viewing or updating the profile.
<b>Actors:</b> Recruiter Job seeker
<b>Preconditions:</b> The actor must be logged into the system.
<b>Postcondition:</b> Upon successful execution of the use case the details are viewed or updated.
<b>Flow of Events:</b> <b>Basic Flow:</b> The basic flow starts when the actor chooses to view, update his profile. Portal provides the actor with the options of view or update to choose from and one of the sub flows will be executed according to it. <ol style="list-style-type: none"> <li><b>View Profile:</b> <ul style="list-style-type: none"> <li>When the user selects the option to view profile the user is taken to his profile section which has bio and vacancies created in the case of recruiter and resume and jobs applied for in the case of job-seeker along with their username and registered email address and contact information.</li> </ul> </li> <li><b>Update Profile:</b> <ul style="list-style-type: none"> <li>When user chooses the option to update profile while being present on the view profile section of the portal a form containing the fields with pre-existing values is prompted. The user makes the necessary changes and confirms his submission by clicking on the submit option. The profile is updated thereon.</li> </ul> </li> </ol>
<b>Alternate Flow:</b> <b>Alternate Flow 1: Update cancelled</b> - If the user chooses not to update the or denies the confirmation at respective steps the basic flow is restarted from the beginning. <b>Alternative Flow 2: User Exits:</b> Actor logs out portal or close the browser window.
<b>Special Requirements:</b> None

**Associated Use case:**

Login, Maintain vacancy details, Maintain application details

**B. Validity Checks-**

- The job seeker and recruiter can view their respective profiles and update them.
- All the fields that are marked required (full name, email, and username) cannot be left empty.
- All data is of string datatype.
- Update is not successful unless the actor confirms the update through the confirmation prompt.

**7. Maintain Application Details-****A. Use Case Description****Introduction:**

This use case documents the steps to be followed for managing the application details by a job seeker for a vacancy he has applied for. It includes viewing and withdrawing the application.

**Actors:**

Job seeker

Admin

**Preconditions:**

The job seeker must be logged in into the portal and must have chosen the option to maintain applications details.

**Postconditions:**

Once the use case is successful, the application is viewed or withdrawn in accordance to the options selected in the basic flow of the use case.

**Flow of events****Basic Flow:**

The basic flow starts when the job seeker chooses to view or withdraw his application.

Portal provides the job-seeker with these two options to choose from and one of the sub flows will be executed according to it.

**1. View an application**

- The portal asks the job-seeker to choose the application that he wants to view from the applications applied for.
- Once the job seeker selects an application to view, the application details such as time of application, resume and cover letter uploaded at time of applying is made available.

**2. Withdraw application**

- The portal asks the job seeker to choose the application he wants to withdraw from.
- Once the job-seeker makes a choice, he is asked to confirm the withdrawal by asking the user to sign the responsibility of withdrawal. Upon confirmation application is withdrawn a notification mail is sent to the job seeker.

**Alternate Flow:**

**Alternate Flow 1: Withdraw Cancelled** - If the job-seeker chooses not to withdraw or denies the confirmation at respective steps the basic flow is restarted from the beginning.

**Alternative Flow 2: User Exits:**

Actor logs out portal or close the browser window.

**Special Requirements:**

None

**Associated Use case:**

Login, Maintain user details

## B. Validity checks-

- Job seekers can only access the applications that have been applied with the status of their application.
- The default status for an application shall be set to ‘applied.’
- Job seeker shall not be able to withdraw successfully unless confirming on the application prompt on the screen.

## 8. Search for job

### A. Use Case Description

#### Introduction-

This use case documents the steps followed by the job-seeker to search for jobs.

#### Actors:

Job seeker

#### Preconditions:

The job seeker must be logged into the portal and present on the search page/ homepage of the portal.

#### Postconditions:

After successful execution of the use case the job-seeker is able to view the details of a job selected to view out of the search results.

#### Flow of events:

##### Basic Flow:

- The job-seeker can search by the name of roles from the dropdown menu and apply filter in terms of PayScale, part-time or full-time or internship, or location and duration or the job id and the relevant open vacancies shall be presented.
- The job seeker can choose from the search results and see the details posted by the recruiter for that vacancy and shall be provided the option to apply for the vacancy.

##### Alternate Flow:

**Alternate Flow 1: Search Cancelled** - If the job-seeker chooses not to continue with the search or resets the search filters the use case restarts from the start of the basic flow.

##### Alternative Flow 2: User Exits:

Actor logs out portal or close the browser window.

##### Special Requirements:

Login Credentials

##### Associated Use case:

Login, Maintain vacancy details

## B. Validity Checks-

- Only those users logged in as job seekers shall have the access to search for a job

## 9. Apply for a vacancy

### A. Use Case Description

#### Introduction-

This use case documents the steps followed by the job-seeker to apply for the jobs present in the search results.

<b>Actors:</b> Job seeker
<b>Preconditions:</b> The job seeker must be logged into the portal and present on the search page/ homepage of the portal.
<b>Postconditions:</b> After successful execution of the use case the application for the job is successful and receive a confirmation mail in case of successful application.
<b>Flow of events:</b> <b>Basic Flow:</b> <ul style="list-style-type: none"> <li>After the job seeker has searched for the interested jobs and selected one of them. The option is provided to him to apply for the job.</li> <li>After the job seeker chose the option to apply, a form is presented to the job seeker asking for personal details, resume and cover letter. After the actor fills all these fields. He is prompted the submit.</li> <li>Upon clicking the submit option a confirmation request with checkbox to sign the terms and conditions is prompted. If the user gives the confirmation, the application is finally submitted on the portal and a confirmation mail is sent on the registered email address of the job seeker.</li> </ul>
<b>Alternate Flow:</b> <b>Alternate Flow 1: Application process interrupted</b> - If the job-seeker chooses not to continue with the application process, the use case restarts from the start of the basic flow. <b>Alternate Flow 2: Already applied for the vacancy-</b> If the job seeker has already applied for the vacancy. He cannot apply again and is redirected back to the search page. <b>Alternative Flow 3: User Exits:</b> Actor logs out portal or close the browser window.
<b>Special Requirements:</b> None
<b>Associated Use case:</b> Login, maintain vacancy details, maintain application details, Maintain user details

## B. Validity Checks-

- Only those users logged in as job seekers shall have the access to apply for a job
- Resume field shall be filled by a suitable link by the jobseeker to access his resume.
- Resume field shall not be empty
- It is necessary to confirm the application prompt for a successful application.

## 3.3 Performance Requirement-

- Latest version of browser
- Responses within 3 seconds
- Should support atleast 500 users at a time.

## 3.4 Design Constraints – None

## 3.5 Software System Attributes-

- Usability- The application will be user-friendly and easy to operate and the functions will be easily understandable.

- Reliability- The application shall be available to both job-seekers and the recruiters for the lifetime once they are registered and shall have a high degree of fault tolerance.
- Security- The application shall be password protected and requires all its users to login with a valid email ID and password provided by them at the time of registration.
- Maintainability- The application shall be designed in a maintainable manner to incorporate the upcoming changes/ requirements in the individual modules.
- Portability – The application shall be deployed on the world wide web (WWW) and shall be accessible on any device with latest version of any internet browser.

### **3.6 Logical database instructions-**

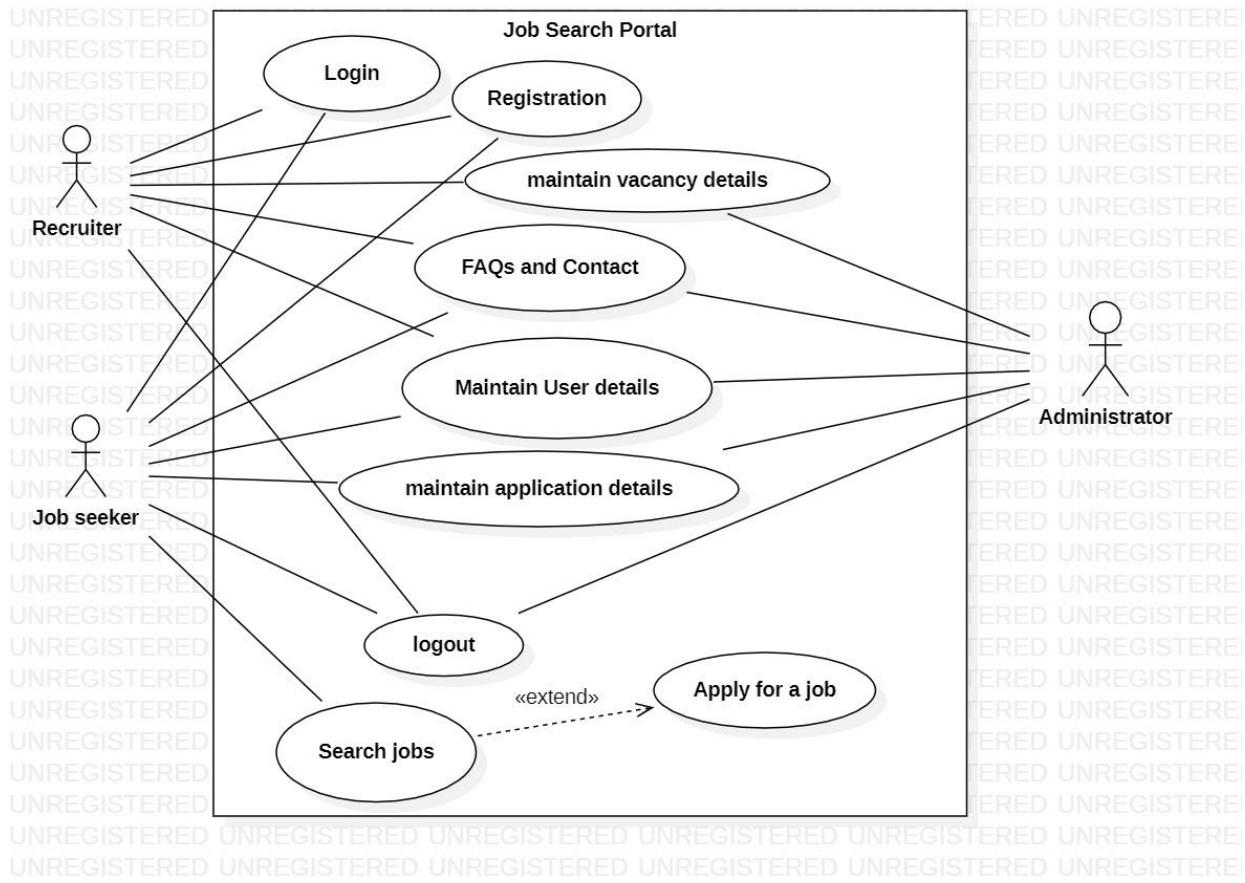
The following information will be place in the database-

Collection Name	Description
Jobseeker	Stores the login details and other profile details of the job seeker
Recruiter	Stores the login details and other profile details of the recruiter.
Created Jobs	Stores the jobs that are created by the recruiters.
Applied Jobs	Stores the job applications filled by the job-seekers for various jobs created by the recruiters on the website.

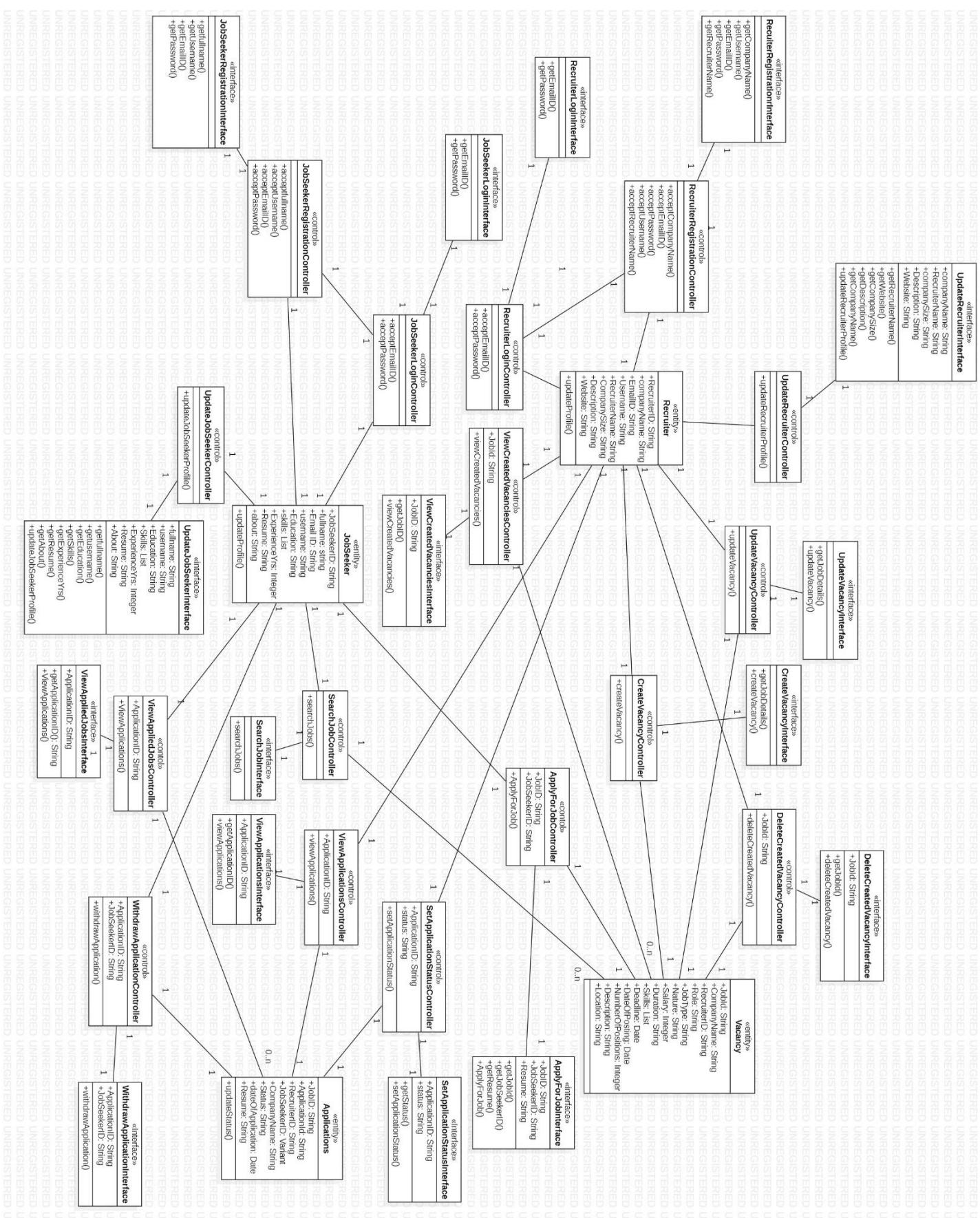
### **3.7 Other Requirements- None**

## **4. Appendix-**

### **4.1 Use Case Diagram-**



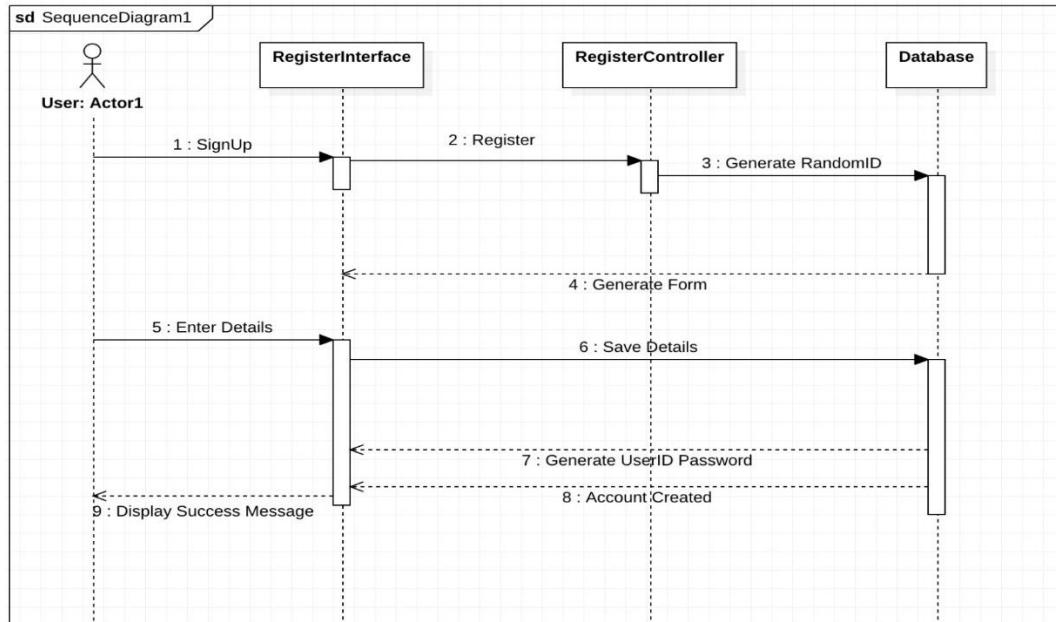
## 4.2 Class Diagram



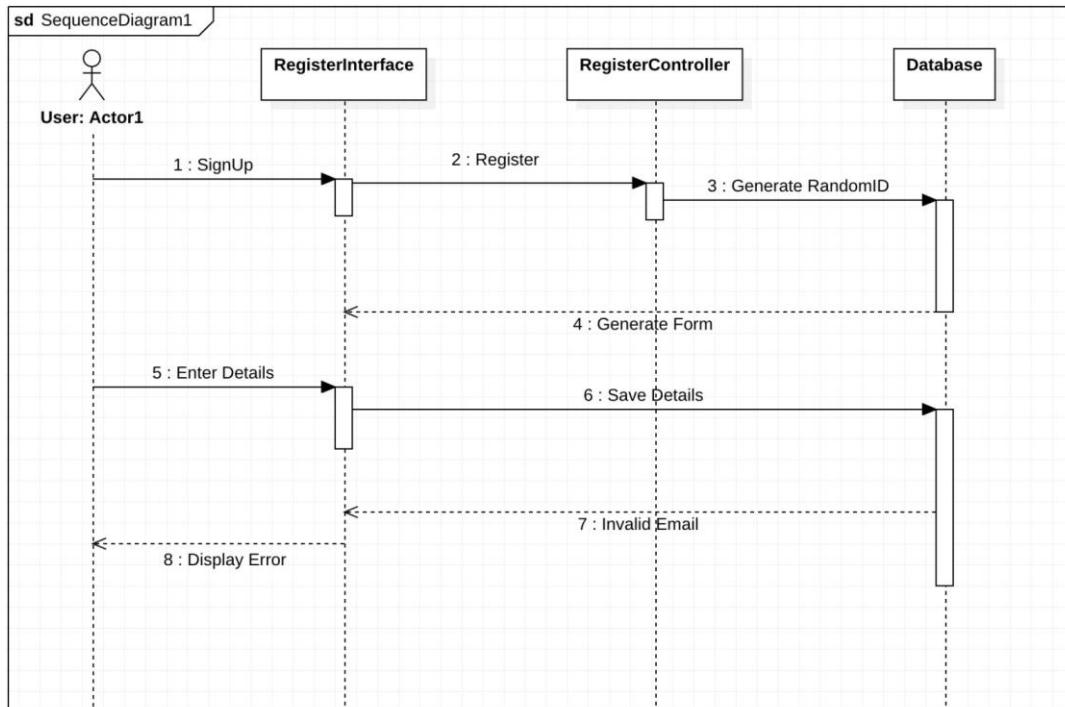
## 4.3 Sequence Diagram

### 4.3.1 Register use case-

Basic Flow-

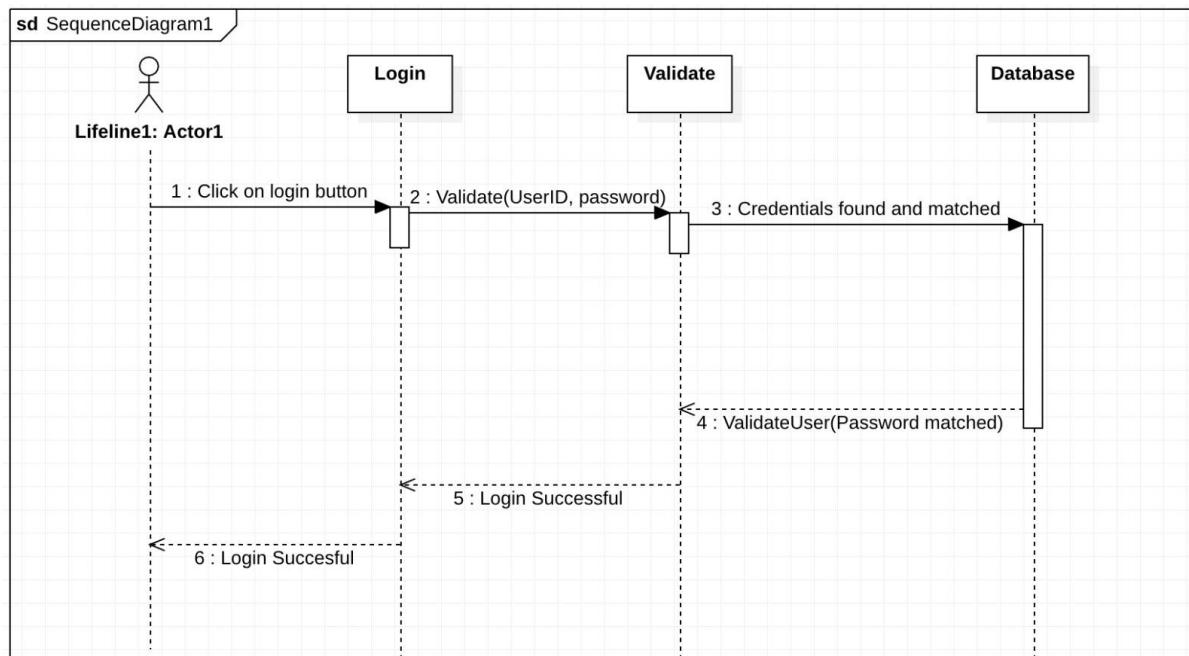


Alternate Flow-1: Invalid email

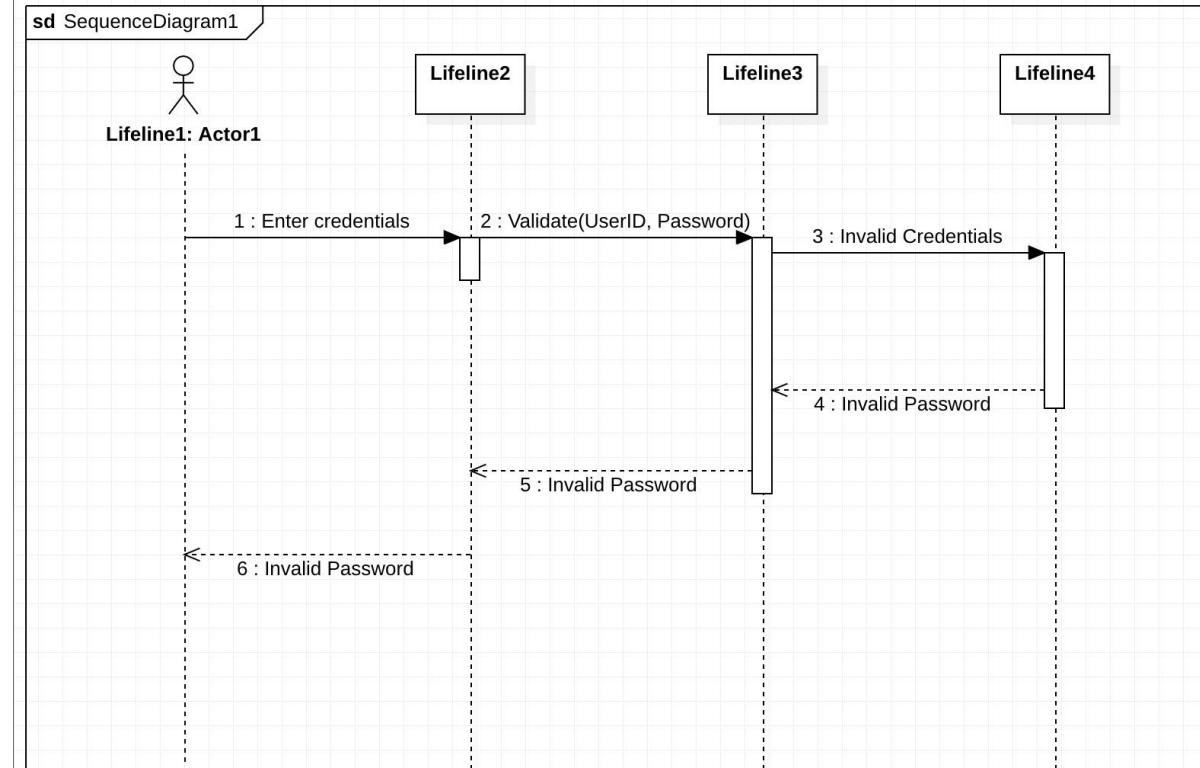


#### 4.3.2 Login use case-

Basic flow-

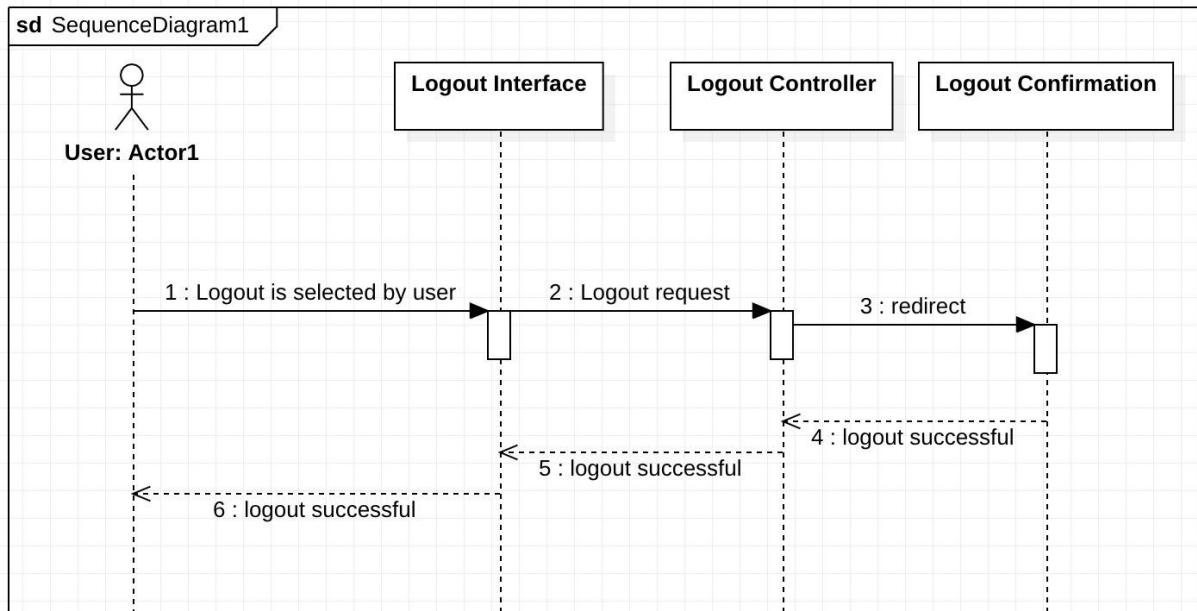


Alternate flow-1: Invalid credentials-



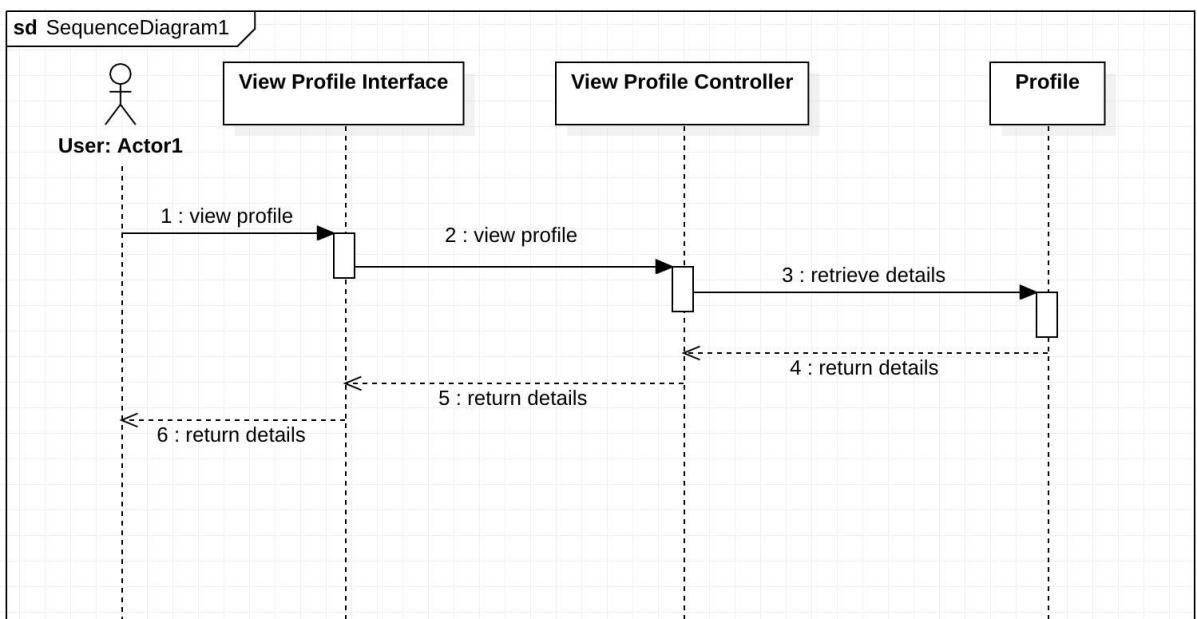
### **4.3.3 Logout use case-**

Basic flow-

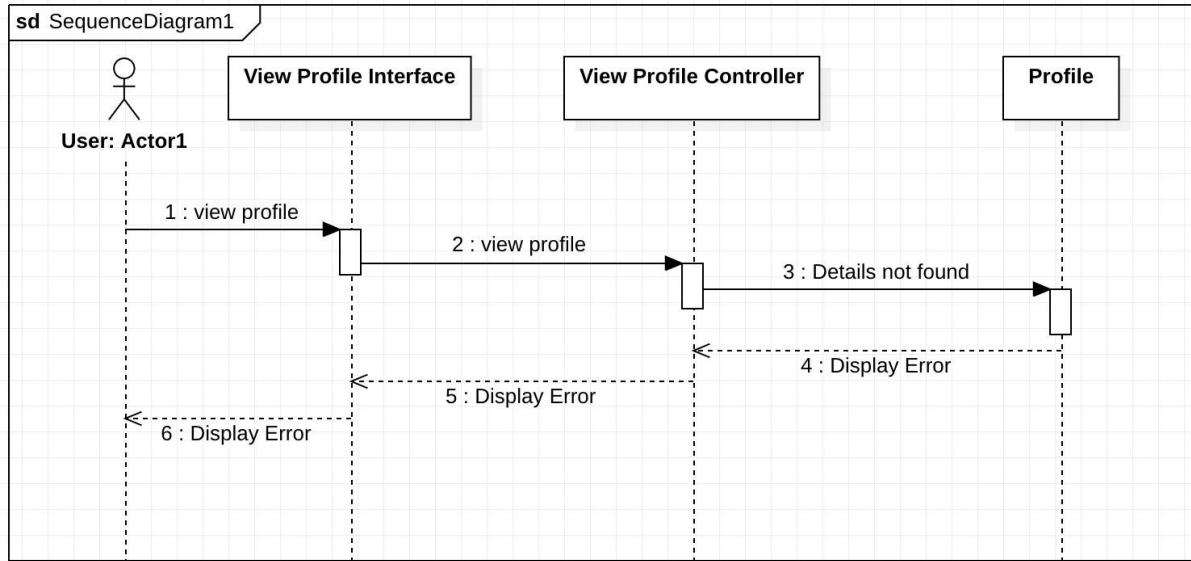


### **4.3.4 Manage User details use case-**

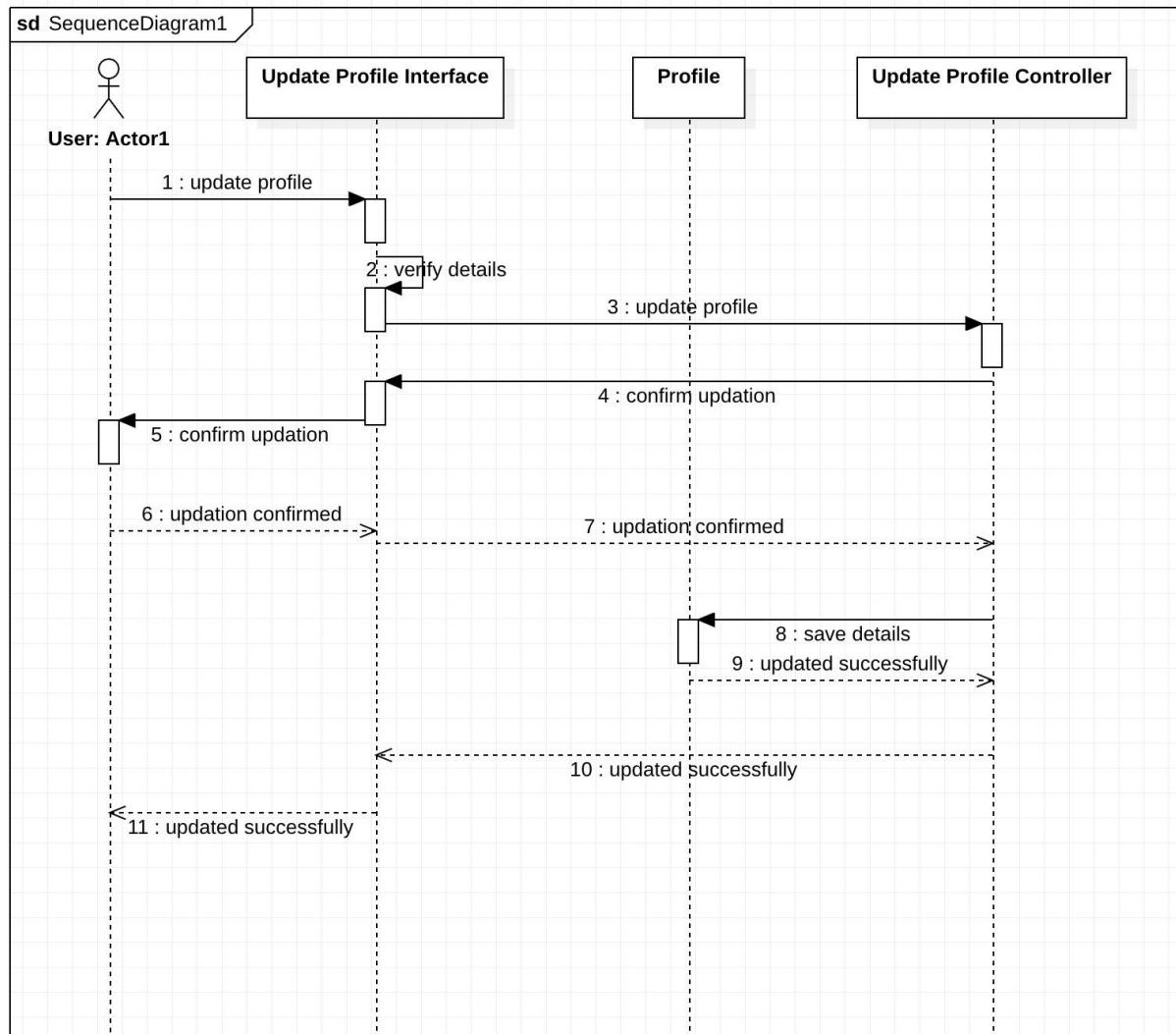
Basic flow-1: View User



### Alternate flow-1: The user does not exist

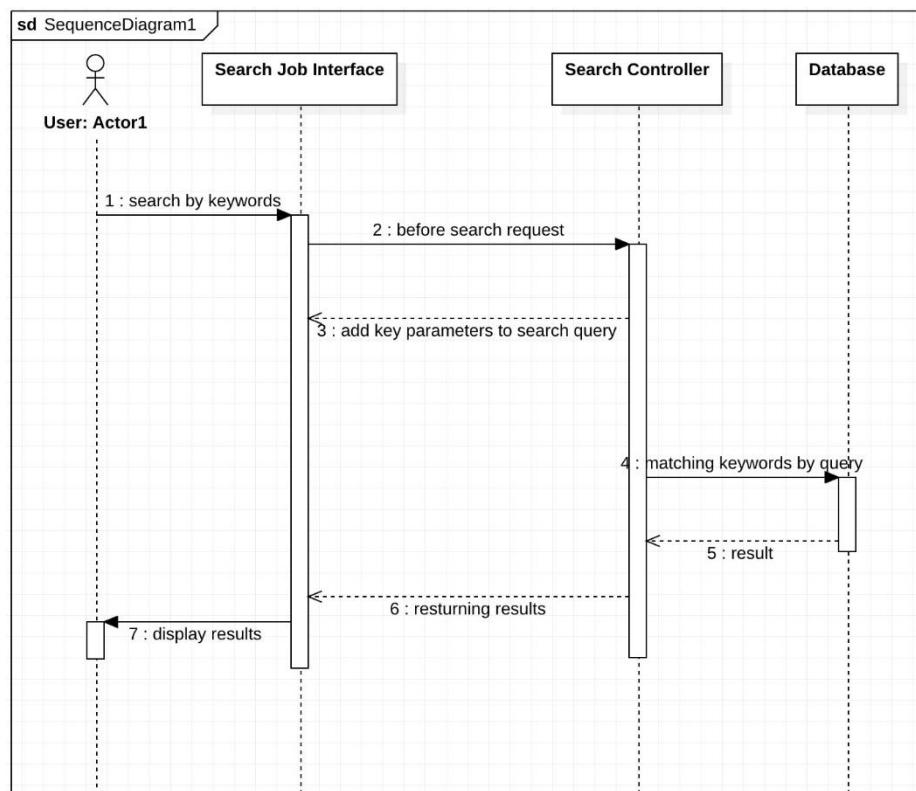


### Basic Flow-2 : Update Profile-

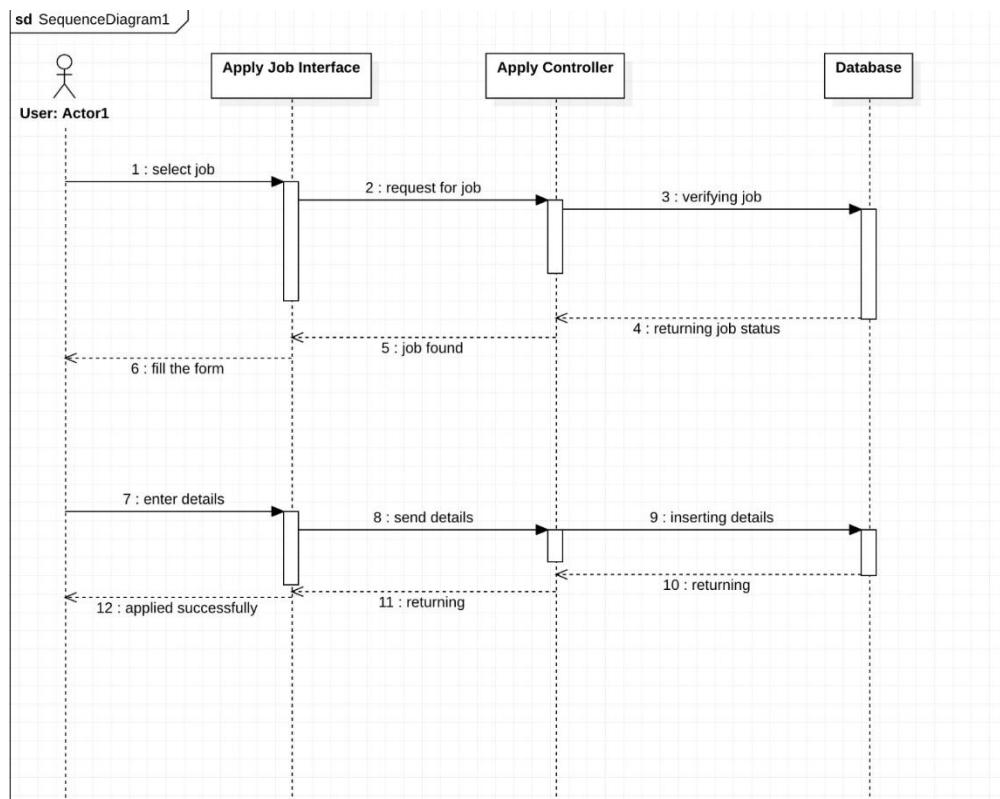


#### 4.3.5 Search Job use case-

Basic flow-

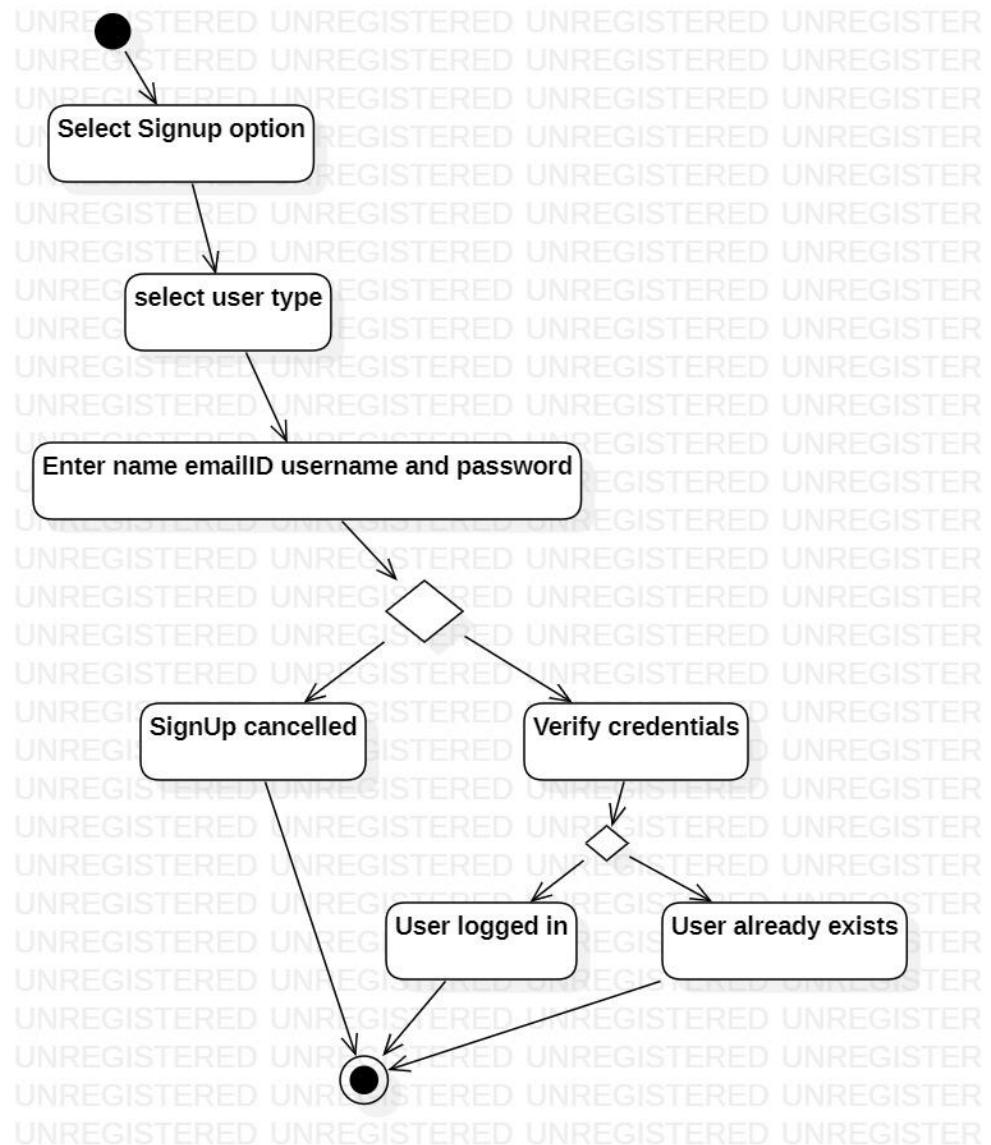


#### 4.3.6 Apply for a Job use case-

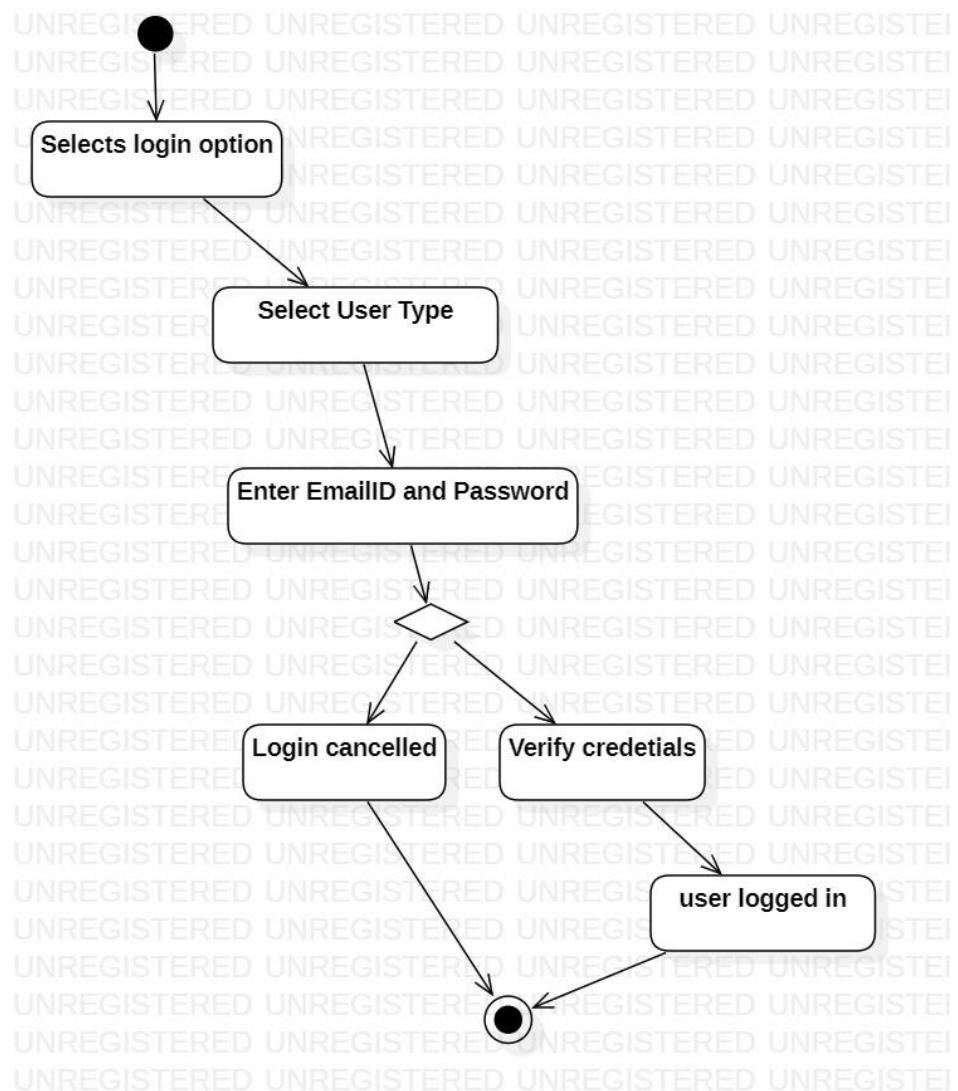


## 4.4 Activity Diagrams

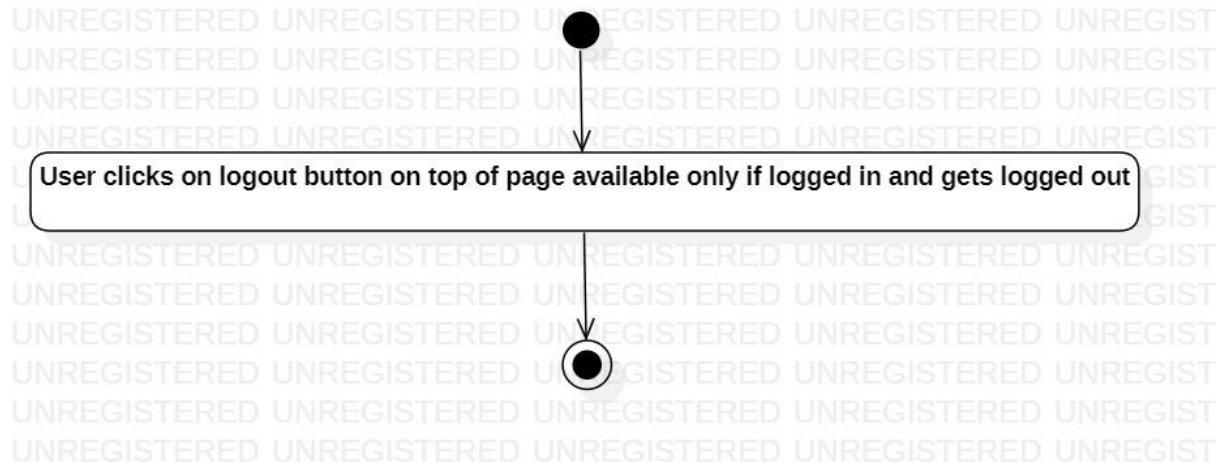
### 4.4.1 Activity diagram for ‘Registration’ use case-



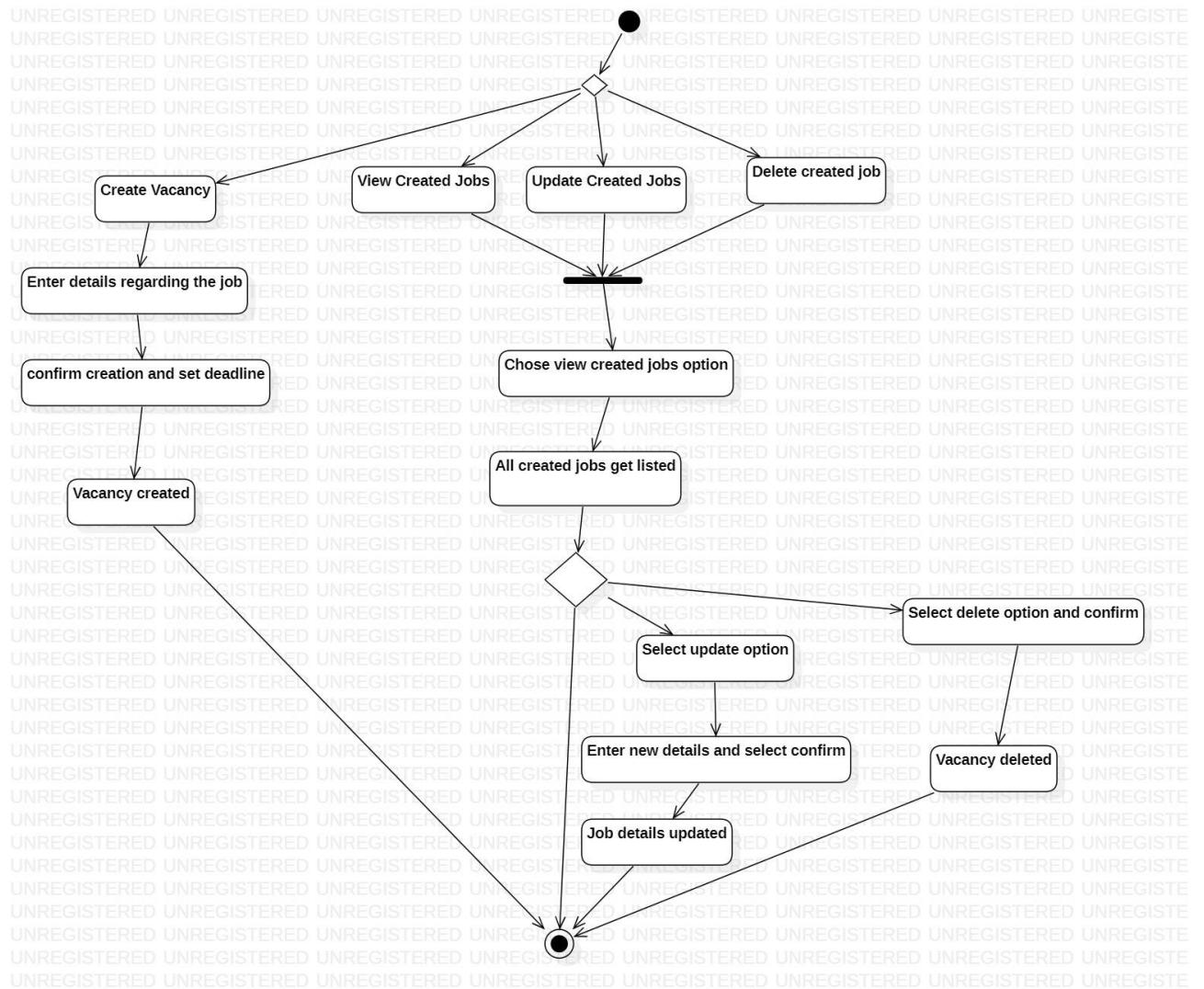
#### 4.4.2 Activity diagram for ‘Login’ use case-



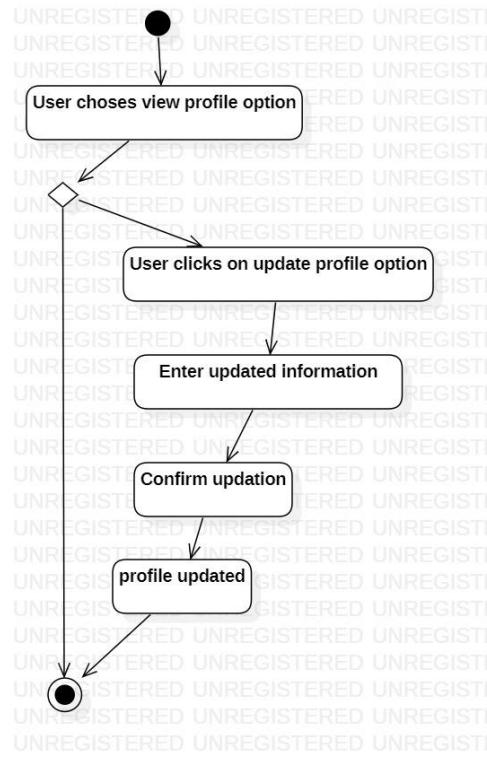
#### 4.4.3 Activity Diagram for ‘Logout’ use case-



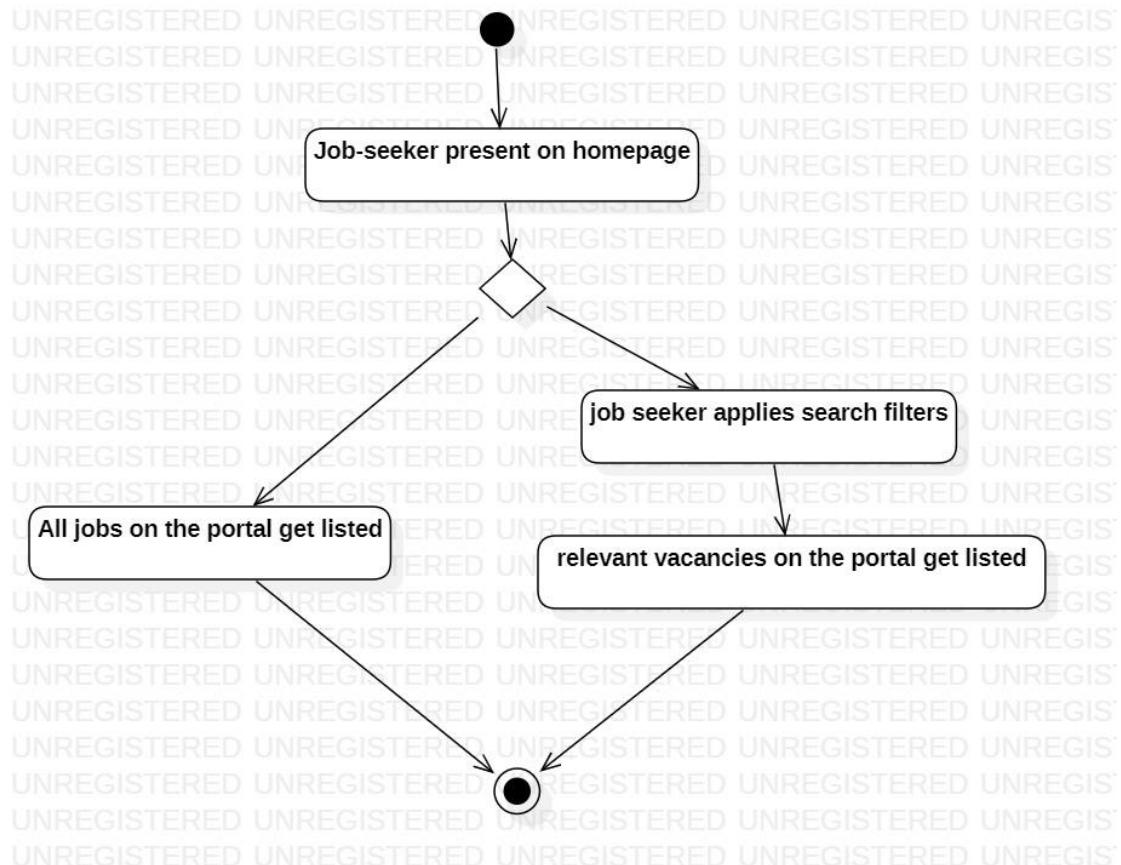
#### 4.4.4 Activity diagram for ‘Maintain Vacancy details’ use case-



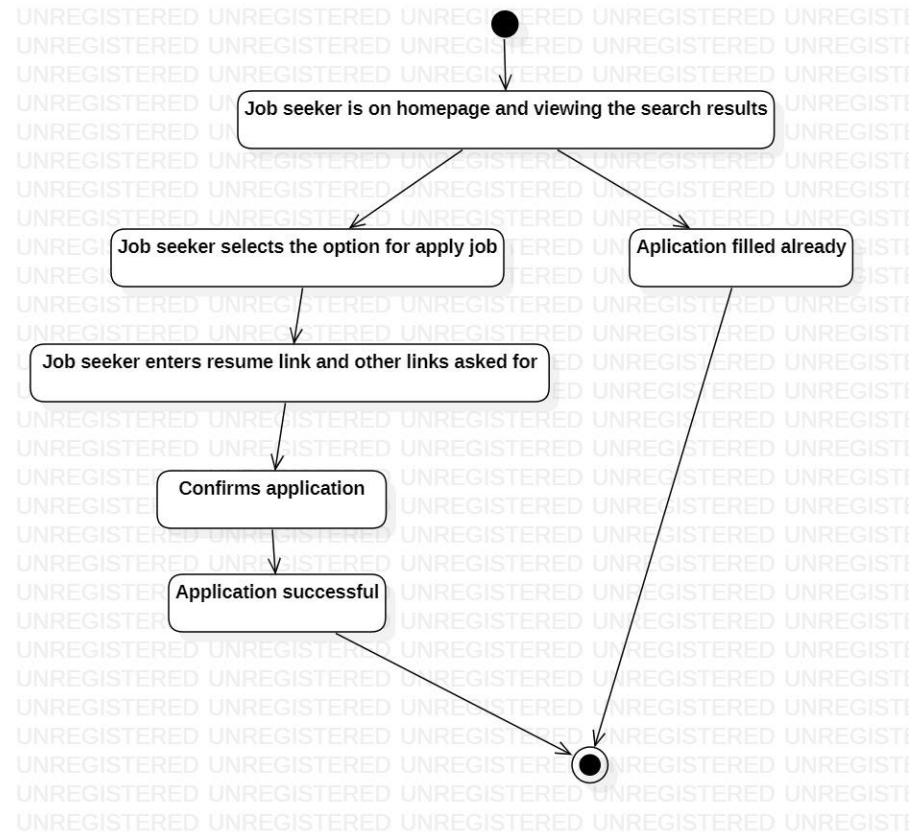
#### 4.4.5 Activity Diagram for ‘Maintain User Profile’ use case-



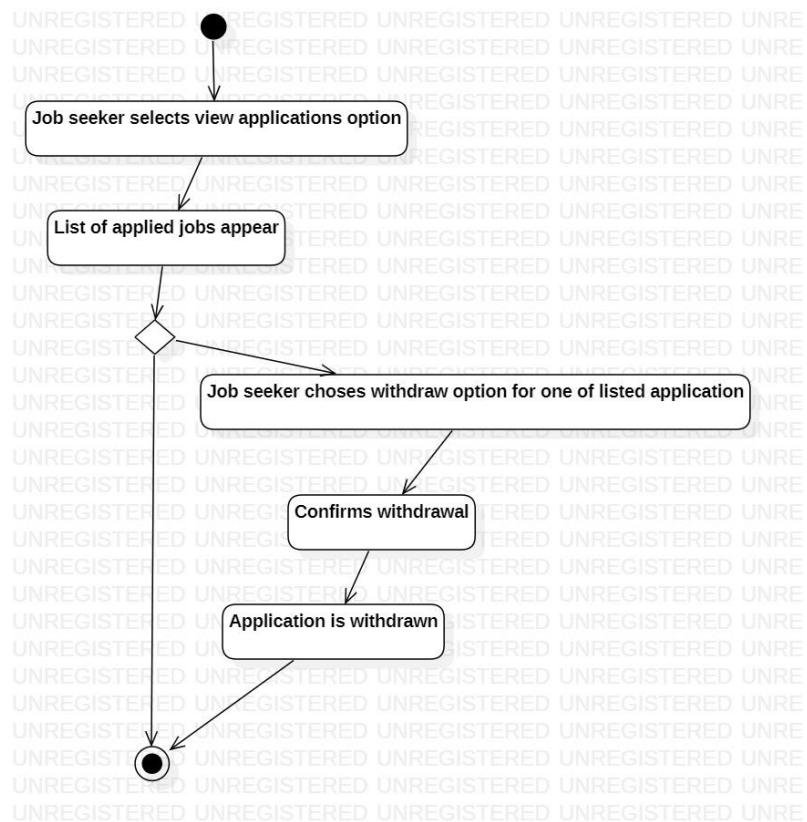
#### 4.4.6 Activity diagram for ‘Search Job’ use case-



#### 4.4.7 Activity diagram for ‘Apply for a vacancy’ use case-

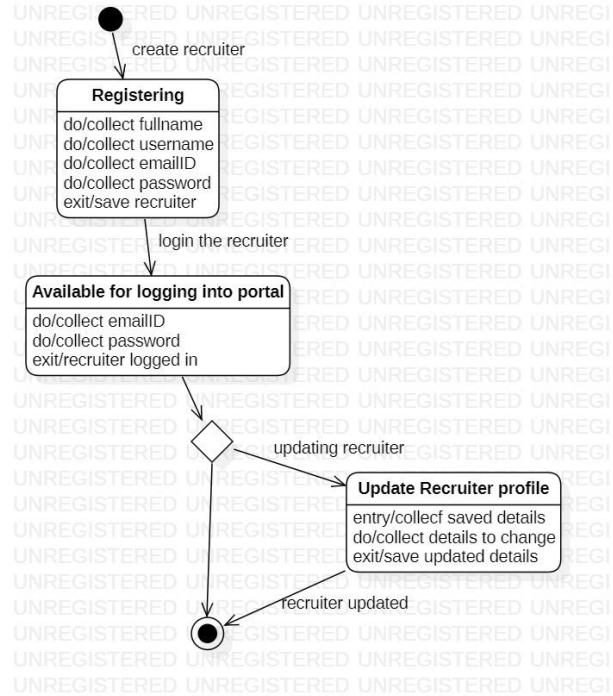


#### 4.4.8 Activation diagram for ‘Maintain Application Details’ use case-

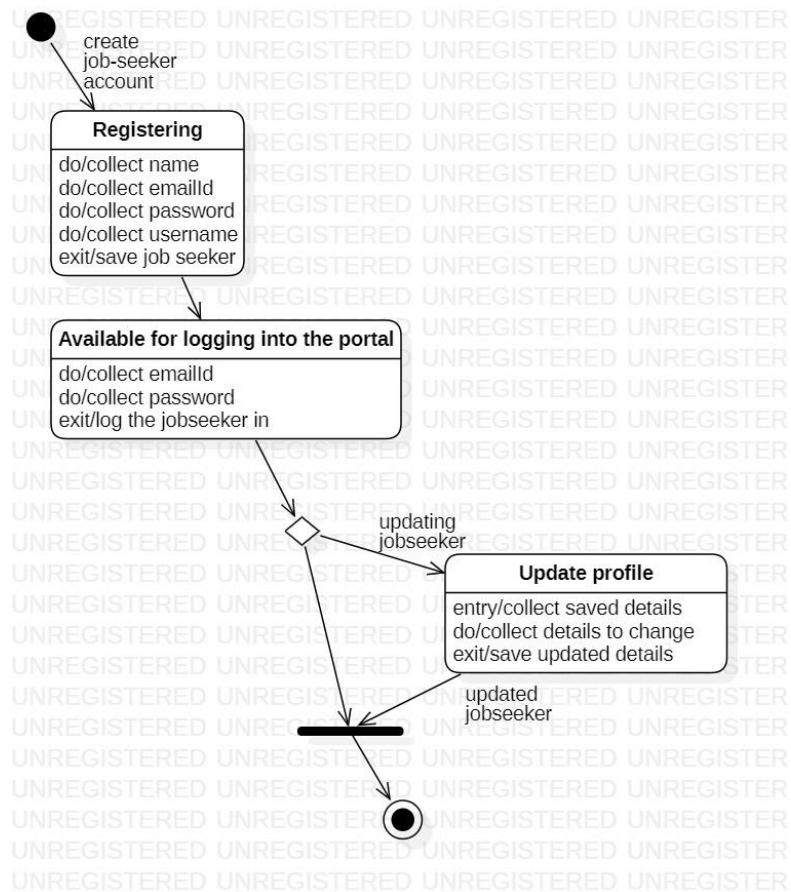


## 4.5 Statechart diagrams-

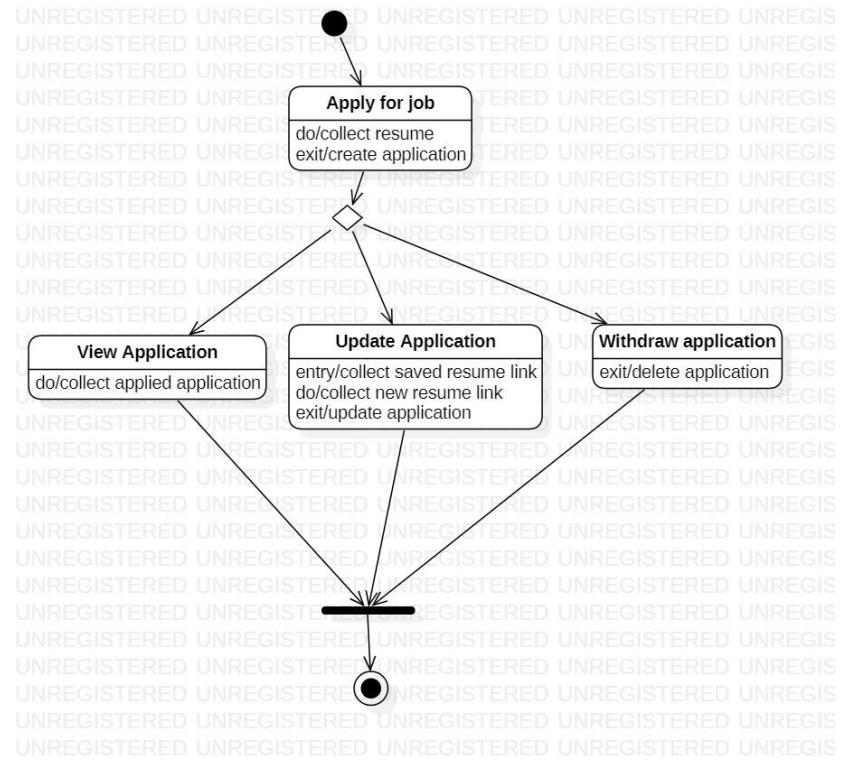
### 4.5.1 Statechart diagram for ‘Recruiter’ entity class-



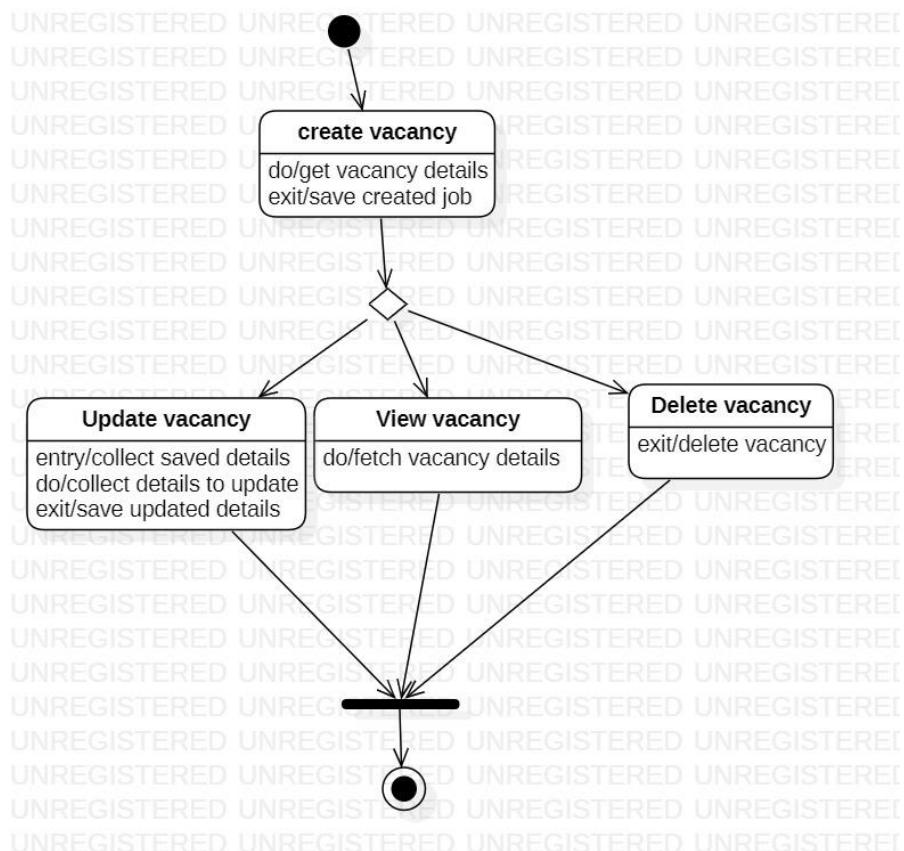
### 4.5.2 Statechart diagram for ‘Job-seeker’ entity class-



#### 4.5.3 Statechart diagram for ‘Application’ entity class-



#### 4.5.4 Statechart diagram for ‘Vacancy’ entity class-



## Test Case Matrices for Job Search Portal

### **Test Case Matrix for Register Use Case-**

Test Case ID	Scenario and description	Input-1 User type	Input-2 Full Name	Input -3 Username	Input-4 Email ID	Input-5 Password	Expected Result	Remarks
TC 1	Scenario 1- Register user	Valid	Valid	Valid	Valid	Valid	The user's account is created as per the user type.	-
TC 2	Scenario 2- Register User Alternate flow: Invalid entry	Invalid	Valid	Valid	Valid	Valid	An account is not created	User type not chosen.
TC 3	Scenario 3- Register User Alternate flow: Invalid entry	Valid	Invalid	Valid	Valid	Valid	An account is not created	Invalid full name ( less than 2 characters)
TC 4	Scenario 4- Register User Alternate flow: Invalid entry	Valid	Valid	Invalid	Valid	Valid	An account is not created	Invalid username
TC 5	Scenario 5- Register User Alternate flow: Invalid entry	Valid	Valid	Valid	Invalid	Valid	An account is not created	Invalid email id.
TC 6	Scenario 6- Register User Alternate flow: Invalid entry	Valid	Valid	Valid	Valid	Invalid	An account is not created	Invalid password
TC 7	Scenario 7- Register User Alternate flow: The user already exists	Valid	Valid	Invalid	Valid	Valid	An account is not created.	The username is already registered for an account.
TC 8	Scenario 8- Register User Alternate flow: The user already exists	Valid	Valid	Valid	Invalid	Valid	An account is not created.	Email already registered with an account.
TC9	Scenario 9- Register user Alternate flow: User exits	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	An account is not created, and the user reaches another page of choice or outside the browser window.	User exits.

### Test Case Matrix for Register Use Case (with actual inputs) -

Test Case ID	Scenario and description	Input-1 User type	Input-2 Full Name	Input -3 Username	Input-4 Email ID	Input-5 Password	Expected Result	Remarks
TC 1	Scenario1- Register user	Job seeker	Abhinav	Abhinav01	<a href="mailto:abhinav@gmail.com">abhinav@gmail.com</a>	*****	The user's account is created as per the user type.	-
TC 2	Scenario 2- Register User Alternate flow: Invalid entry	-	Abc	Abc01	<a href="mailto:abc@gmail.com">abc@gmail.com</a>	*****	An account is not created	User type not chosen.
TC 3	Scenario 3- Register User Alternate flow: Invalid entry	Job seeker	Q	Quanta1	<a href="mailto:quanta@gmail.com">quanta@gmail.com</a>	*****	An account is not created	Invalid full name ( less than 2 characters)
TC 4	Scenario 4- Register User Alternate flow: Invalid entry	Job seeker	Abhinav	A	<a href="mailto:abcd@gmail.com">abcd@gmail.com</a>	*****	An account is not created	Invalid username
TC 5	Scenario 5- Register User Alternate flow: Invalid entry	Job seeker	Amriti Gupta	Amriti	Amriti2003	*****	An account is not created	Invalid email id.
TC 6	Scenario 6- Register User Alternate flow: Invalid entry	Job seeker	Amriti Gupta	Amriti Gupta	<a href="mailto:amriti20@gmail.com">amriti20@gmail.com</a>	***	An account is not created	Invalid password (less than 5 characters)
TC 7	Scenario 7- Register User Alternate flow: The user already exists	Job seeker	Chips Yu	Abhinav01	<a href="mailto:abcd189@gmail.com">abcd189@gmail.com</a>	*****	An account is not created.	The username is already registered for an account.
TC 8	Scenario 8- Register User Alternate flow: The user already exists	Job seeker	Tywin Lannister	Aegon Targaryen	<a href="mailto:abhinav01@gmail.com">abhinav01@gmail.com</a>	*****	An account is not created.	Email already registered with an account.
TC9	Scenario 9- Register user Alternate flow: The user exits	Recruiter	Amriti Gupta	Amriti Gupta	<a href="mailto:amriti20@gmail.com">amriti20@gmail.com</a>	*****	An account is not created, and the user reaches another page of choice or outside the browser window.	User exits.

### **Test Case Matrix for Login Use Case-**

Test Case ID	Scenario name and description	Input 1 Login ID	Input 2 Password	Expected output	remarks
TC1	Scenario 1-Login	Valid	Valid	User allowed to login	-
TC2	Scenario 2-Login alternative flow: invalid entry	invalid	valid	Login ID invalid	Login ID not in specified format
TC3	Scenario 2-Login alternative flow: invalid entry	valid	valid	Login ID invalid	Login ID does not exist in database
TC4	Scenario 2-Login alternative flow: invalid entry	valid	invalid	Password invalid	Password not in specified format
TC5	Scenario 2-Login alternative flow: invalid entry	valid	valid	Password invalid	Password does not exist in the database
TC6	Scenario 2-Login alternative flow: invalid entry	invalid	invalid	Login ID and password	Login ID and password not in the specified format.
TC7	Scenario 3-User exits	valid/invalid	valid/invalid	User comes out of the system	-

### **Test Case Matrix for Login Use Case (with actual inputs) –**

Test Case ID	Scenario name and description	Input 1 Login ID	Input 2 Password	Expected output	remarks
TC1	Scenario 1-Login	<a href="mailto:abc@xyz.com">abc@xyz.com</a>	12345678	User allowed to login	-
TC2	Scenario 2-Login alternative flow: invalid entry	abc*	12345678	Login ID invalid	Login ID not in specified format
TC3	Scenario 2-Login alternative flow: invalid entry	<a href="mailto:xyz@abc.com">xyz@abc.com</a>	12345678	Login ID invalid	Login ID does not exist in database
TC4	Scenario 2-Login alternative flow: invalid entry	<a href="mailto:abc@xyz.com">abc@xyz.com</a>	123	Password invalid	Password not in specified format
TC5	Scenario 2-Login alternative flow: invalid entry	<a href="mailto:abc@xyz.com">abc@xyz.com</a>	87654321	Password invalid	Password does not exist in the database
TC6	Scenario 2-Login alternative flow: invalid entry	abc*	123	Login ID and password	Login ID and password not in the specified format.
TC7	Scenario 3-User exits	<a href="mailto:abc@xyz.com">abc@xyz.com</a>	12345678	User comes out of the system	-

### Test case matrix for Maintain Vacancy Details use case-

Test Case ID	Scenario and description	Input-1 Company Name	Input-2 Role	Input - 3 Job Type	Input-4 Salary	Input -5 Deadline	Update Confirmed	Delete Confirmed	Expected Result	Remarks
TC 1	Scenario1- Add a job	Valid	Valid	Valid	Valid	Valid	n/a	n/a	The recruiter has created a vacancy.	-
TC 2	Scenario 2- Add a job Alternate flow: Invalid entry	Invalid	Valid	Valid	Valid	Valid	n/a	n/a	A vacancy is not created	Invalid Company Name
TC 3	Scenario 3- Add a job Alternate flow: Invalid entry	Valid	Invalid	Valid	Valid	Valid	n/a	n/a	A vacancy is not created	Invalid role
TC 4	Scenario 4- Add a job Alternate flow: Invalid entry	Valid	Valid	Invalid	Valid	Valid	n/a	n/a	A vacancy is not created	Invalid job type
TC 5	Scenario 5- Add a job Alternate flow: Invalid entry	Valid	Valid	Valid	Invalid	Valid	n/a	n/a	A vacancy is not created	Invalid salary
TC 6	Scenario 6- Add a job Alternate flow: Invalid entry	Valid	Valid	Valid	Valid	Invalid	n/a	n/a	A vacancy is not created	Invalid date in deadline.
TC 7	Scenario7- Add a job Alternate flow: User Exits	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	n/a	n/a	A vacancy is not created	-
TC8	Scenario-8 View vacancies	n/a	n/a	n/a	n/a	n/a	n/a	n/a	The list of vacancies is found.	-
TC9	Scenario-9 View created vacancies Alternate flow: The	n/a	n/a	n/a	n/a	n/a	n/a	n/a	The recruiter is directed back to normal flow.	-

	recruiter exits									
TC10	Scenario-10 Update created job	n/a	n/a	n/a	Valid	Valid	Yes	n/a	The vacancy has been updated.	Vacancy updated with new salary and deadline.
TC11	Scenario-11 Update a created job Alternative flow: Invalid entry	n/a	n/a	n/a	Invalid	Valid	No	n/a	The vacancy has not been updated.	Invalid salary.
TC 12	Scenario-12 Update created job Alternative flow: Invalid entry	n/a	n/a	n/a	Valid	Invalid	No	n/a	The vacancy has not been updated.	Invalid date in deadline.
TC13	Scenario-13 Update created job Alternative flow: Recruiter exits	n/a	n/a	n/a	Valid/ Invalid	Valid/ Invalid	No	n/a	The recruiter is directed back to normal flow and the vacancy is not updated.	-
TC14	Scenario-14 Delete Job	n/a	n/a	n/a	n/a	n/a	n/a	Yes	The created vacancy has been deleted	-
TC15	Scenario-15 Delete job Alternative flow: Recruiter exits	n/a	n/a	n/a	n/a	n/a	n/a	No	The recruiter is directed back to normal flow.	-

#### Test case matrix for Maintain Vacancy Details use case (with actual inputs)-

Test Case ID	Scenario and description	Input-1 Company Name	Input-2 Role	Input -3 Job Type	Input-4 Salary	Input -5 Deadline	Update Confirmed	Delete Confirmed	Expected Result	Remarks
TC 1	Scenario1- Add a job	Google	Software Development Intern	Full-time, onsite	800000	26-07-2023, 23:59	n/a	n/a	The recruiter has created a vacancy.	-
TC 2	Scenario 2- Add a job Alternate flow: Invalid entry	G	Software Development Intern	Full-time, onsite	800000	26-07-2023, 23:59	n/a	n/a	A vacancy is not created	Invalid Company Name

TC 3	Scenario 3- Add a job Alternate flow: Invalid entry	Google	S	Full-time, onsite	800000	26-07-2023, 23:59	n/a	n/a	A vacancy is not created	Invalid role
TC 4	Scenario 4- Add a job Alternate flow: Invalid entry	Google	Software Development Intern	on	800000	26-07-2023, 23:59	n/a	n/a	A vacancy is not created	Invalid job type
TC 5	Scenario 5- Add a job Alternate flow: Invalid entry	Google	Software Development Intern	Fulltime, onsite	ABC	26-07-2023, 23:59	n/a	n/a	A vacancy is not created	Invalid salary
TC 6	Scenario 6- Add a job Alternate flow: Invalid entry	Google	Software Development Intern	Fulltime, onsite	800000	26-02-2005, 23:59	n/a	n/a	A vacancy is not created	Invalid date in deadline.
TC 7	Scenario7-Add a job Alternate flow: User Exits	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	n/a	n/a	A vacancy is not created	-
TC8	Scenario-8 View vacancies	n/a	n/a	n/a	n/a	n/a	n/a	n/a	The list of vacancies is found.	-
TC9	Scenario-9 View created vacancies Alternate flow: The recruiter exits	n/a	n/a	n/a	n/a	n/a	n/a	n/a	The recruiter is directed back to normal flow.	-
TC10	Scenario-10 Update created job	n/a	n/a	n/a	800000	26-07-2023, 23:59	Yes	n/a	The vacancy has been updated.	Vacancy updated with new salary and deadline.
TC11	Scenario-11 Update a created job Alternative flow: Invalid entry	n/a	n/a	n/a	ABC	26-07-2023, 23:59	No	n/a	The vacancy has not been updated.	Invalid salary.
TC 12	Scenario-12 Update created job	n/a	n/a	n/a	800000	26-02-2005, 23:59	No	n/a	The vacancy has not	Invalid date in deadline.

	Alternative flow: Invalid entry								been updated.	
TC13	Scenario-13 Update created job Alternative flow: Recruiter exits	n/a	n/a	n/a	Valid/ Invalid	Valid/ Invalid	No	n/a	The recruiter is directed back to normal flow and the vacancy is not updated.	-
TC14	Scenario-14 Delete Job	n/a	n/a	n/a	n/a	n/a	n/a	Yes	The created vacancy has been deleted	-
TC15	Scenario-15 Delete job Alternative flow: Recruiter exits	n/a	n/a	n/a	n/a	n/a	n/a	No	The recruiter is directed back to normal flow.	-