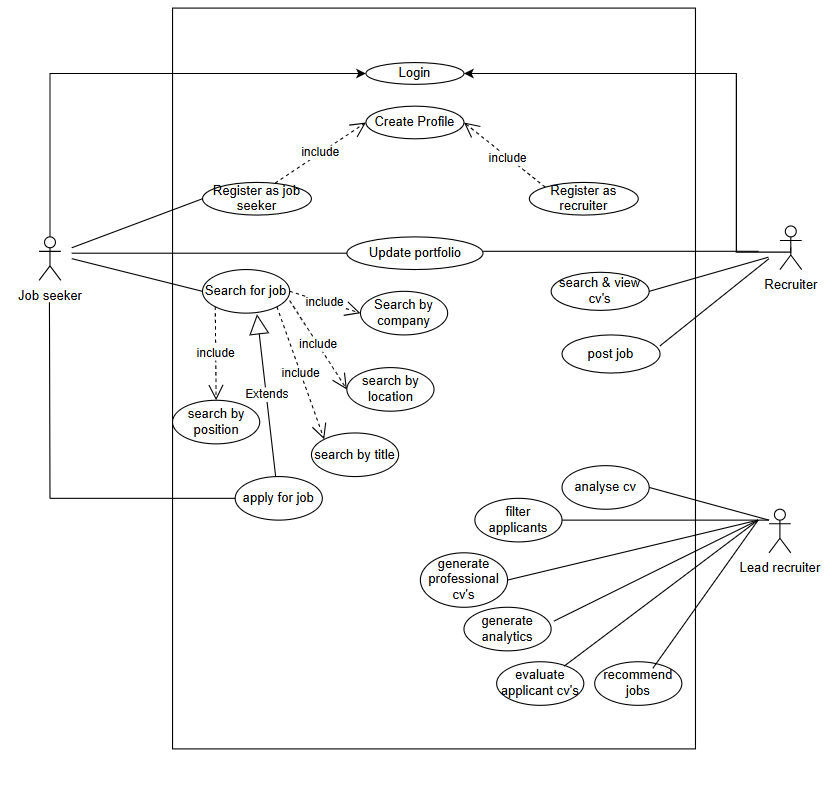
**Document 6- Prepare a use case diagram, activity diagram & a use case specification document**

**Use Case Diagram**

****

**Activity Diagram**

**Use case documents**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-001 | | |
| Use Case Name | Login | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Primary Actor: Job Seeker / Recruiter  Secondary Actor: System | | |
| Description | This case describes how a registered user logs into the NextGreek platform. | | |
| Precondition: | User must be registered. | | |
| Post condition | User is logged in and redirected to the dashboard. | | |
| Normal Flow of Events / Basic Flow: | 1. The user enters their username and password. 2. The system verifies the credentials. 3. If valid, the user is granted access to the dashboard. | | |
| Alternative Flow: | - If incorrect credentials are entered, the system prompts for a retry. - If the user forgets the password, they can reset it via email verification. | | |
| Exceptions: | - If the system is down, the login process fails. - If multiple failed attempts occur, the account is locked for security reasons. | | |
| Frequency of Use: | High (Users log in frequently to access features). | | |
| Constraints | The system must handle multiple concurrent logins efficiently. | | |
| Dependencies | Authentication service and database must be operational. | | |
| Inputs and Outputs | Inputs: Username, password  Outputs: Access granted, dashboard displayed, error messages (if applicable). | | |
| Business Rules | - The password must be encrypted and stored securely.  - The system must implement two-factor authentication for security. | | |
| Miscellaneous Information | - Login sessions expire after a certain period of inactivity. | | |
| Assumptions: | Users have valid credentials and internet access. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-002 | | |
| Use Case Name | Register as job seeker | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Primary Actor: Job seeker | | |
| Description | A job seeker registers in the system to apply for jobs. | | |
| Precondition: | The user must provide a valid email ID and mobile number. | | |
| Post condition | The job seeker account is successfully created. | | |
| Normal Flow of Events / Basic Flow: | 1. The job seeker clicks on "Register".  2. They enter personal details (name, email, phone, etc.).  3. The system verifies email.  4. Registration is completed. | | |
| Alternative Flow: | - If email verification fails, the system prompts for a retry.  - If the user already exists, they are asked to log in. | | |
| Exceptions: | - The email ID is invalid or already in use.  - The internet connection is lost during registration. | | |
| Frequency of Use: | Low (Users register only once). | | |
| Assumption | The job seeker has a valid email ID and internet access. | | |
| Constraints | - The system must validate email and phone number formats. - The registration form must not be left incomplete. | | |
| Dependencies | - Email verification service. - Database for storing user information. | | |
| Inputs and Outputs | Inputs: Name, Email ID, Phone number, Password. Outputs: Confirmation email, Registration success message. | | |
| Business Rules | - A unique email ID must be used. - Password must meet security criteria (e.g., minimum 8 characters, special character, etc.). | | |
| Miscellaneous Information | - The user should receive an email confirmation with an activation link. - The registration form should be user-friendly and accessible. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-003 | | |
| Use Case Name | Register as recruiter | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Primary Actor: Recruiter | | |
| Description | A recruiter registers in the system to post job openings. | | |
| Precondition: | The recruiter must provide company details and valid contact information. | | |
| Post condition | The recruiter account is successfully created. | | |
| Normal Flow of Events / Basic Flow: | 1. The recruiter clicks on "Register". 2. They enter the company and contact details. 3. The system verifies the recruiter’s credentials. 4. Registration is completed. | | |
| Alternative Flow: | - If verification fails, the system prompts a retry. - If the recruiter already exists, they are asked to log in. | | |
| Exceptions: | - The company provided details are incorrect or not verified. - The system detects duplicate registrations. | | |
| Frequency of Use: | Low (Users register only once). | | |
| Assumptions: | The recruiter provides genuine company details. | | |
| Constraints | - The system must validate company authenticity. - The recruiter’s email and phone number must be unique. | | |
| Dependencies | - Business verification service. - Database for storing recruiter information. | | |
| Inputs and Outputs | Inputs: Company name, Contact person, Email, Phone number. Outputs: Confirmation email, Registration success message. | | |
| Business Rules | - Recruiters must be affiliated with a registered business. - A recruiter cannot use the same email as an existing user. | | |
| Miscellaneous Information | |  | | --- | |  |  |  | | --- | | - The recruiter should receive an email confirmation with an activation link. - The registration form should be user-friendly and easy to navigate. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-004 | | |
| Use Case Name | Create Profile | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Job seeker | | |
| Description | A job seeker creates and completes their profile. | | |
| Precondition: | The job seeker must be registered in the system. | | |
| Post condition | The job seeker profile is successfully created. | | |
| Normal Flow of Events / Basic Flow: | 1. The user logs in. 2. They enter educational and professional details. 3. The system saves the profile. | | |
| Alternative Flow: | - If profile completion is incomplete, the system prompts a reminder. | | |
| Exceptions: | - The system fails to save the profile due to a technical issue. | | |
| Frequency of Use: | Medium (Users update profiles occasionally). | | |
| Assumptions: | The job seeker provides accurate information. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-005 | | |
| Use Case Name | Update portfolio | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Job seeker | | |
| Description | A job seeker updates their portfolio, including skills, certifications, and experience. | | |
| Precondition: | The user must be logged in. | | |
| Post condition | The portfolio is updated successfully. | | |
| Normal Flow of Events / Basic Flow: | 1. The user selects the "Update Portfolio" option. 2. They edit details such as skills, work experience, and certifications. 3. The system saves the updates. | | |
| Alternative Flow: | - If the user forgets to save changes, the system prompts a reminder. | | |
| Exceptions: | - If there’s a system failure, updates may not be saved. | | |
| Frequency of Use: | Medium (Users update portfolios periodically). | | |
| Assumptions: | The job seeker provides truthful updates. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-006 | | |
| Use Case Name | Search for Job | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Job seeker | | |
| Description | A job seeker searches for job openings using filters. | | |
| Precondition: | The job seeker must be logged in. | | |
| Post condition | The system displays relevant job listings. | | |
| Normal Flow of Events / Basic Flow: | 1. The user selects "Search for Job". 2. They apply filters (company, location, job title, etc.). 3. The system retrieves matching job listings. 4. The user views job details. | | |
| Alternative Flow: | - If no filters are applied, all available jobs are displayed. - If no jobs match the filters, the system suggests similar jobs. | | |
| Exceptions: | - If the system is down, search results cannot be retrieved. | | |
| Frequency of Use: | High (Users frequently search for jobs). | | |
| Assumptions: | The job database is up-to-date and accessible. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-007 | | |
| Use Case Name | Apply for Job | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Job seeker | | |
| Description | A job seeker applies for a job through the system. | | |
| Precondition: | The job seeker must have a complete profile and be logged in. | | |
| Post condition | The job application is successfully submitted. | | |
| Normal Flow of Events / Basic Flow: | 1. The user selects a job listing. 2. They click "Apply". 3. The system verifies the application. 4. The recruiter is notified of the application. | | |
| Alternative Flow: | - If a CV is missing, the system prompts the user to upload it. | | |
| Exceptions: | - If the job is closed, the application cannot be submitted. | | |
| Frequency of Use: | High (Users frequently apply for jobs). | | |
| Assumptions: | The recruiter actively reviews applications. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-008 | | |
| Use Case Name | Post Job | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Recruiter | | |
| Description | A recruiter posts a job opening on the platform. | | |
| Precondition: | The recruiter must be logged in and verified. | | |
| Post condition | The job is listed and visible to job seekers. | | |
| Normal Flow of Events / Basic Flow: | 1. The recruiter selects "Post Job". 2. They enter job details (title, description, requirements, etc.). 3. The system verifies and saves the job listing. 4. The job is made available to seekers. | | |
| Alternative Flow: | If the recruiter saves a draft, the job is not published immediately. | | |
| Exceptions: | - If mandatory fields are missing, the job cannot be posted. | | |
| Frequency of Use: | Medium (Recruiters post jobs periodically). | | |
| Assumptions: | The job details are accurate and complete. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-009 | | |
| Use Case Name | Search & View CVs | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Recruiter | | |
| Description | A recruiter searches for and views job seekers' CVs. | | |
| Precondition: | The recruiter must be logged in. | | |
| Post condition | The recruiter can view candidate details. | | |
| Normal Flow of Events / Basic Flow: | 1. The recruiter selects "Search CVs". 2. They apply filters (skills, location, experience). 3. The system retrieves matching candidates. 4. The recruiter views CVs. | | |
| Alternative Flow: | - If no filters are applied, all available CVs are displayed. | | |
| Exceptions: | - If no candidates match, the recruiter sees no results. | | |
| Frequency of Use: | High (Recruiters search for candidates frequently). | | |
| Assumptions: | The job seeker profiles are complete and up-to-date. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-0010 | | |
| Use Case Name | Analyze CVs | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Lead Recruiter | | |
| Description | The system analyzes CVs based on job requirements. | | |
| Precondition: | The recruiter must have shortlisted candidates. | | |
| Post condition | The system provides a CV analysis report. | | |
| Normal Flow of Events / Basic Flow: | 1. The recruiter selects "Analyze CV". 2. The system compares CVs with job criteria. 3. A report is generated. 4. The recruiter reviews the analysis. | | |
| Alternative Flow: | - If CVs lack required details, a warning is displayed. | | |
| Exceptions: | - If the system fails, CVs are not analyzed. | | |
| Frequency of Use: | Medium (Recruiters analyze CVs when shortlisting candidates). | | |
| Assumptions: | The system has a robust CV analysis algorithm. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-0011 | | |
| Use Case Name | Filter Applicants | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Lead Recruiter | | |
| Description | The system filters applicants based on selected criteria. | | |
| Precondition: | Applicants must have applied for the job. | | |
| Post condition | The recruiter sees a list of filtered candidates. | | |
| Normal Flow of Events / Basic Flow: | 1. The recruiter selects "Filter Applicants". 2. They apply filters (experience, skills, etc.). 3. The system retrieves matching applicants. | | |
| Alternative Flow: | - If no filters are applied, all applicants are shown. | | |
| Exceptions: | - If no applicants meet the criteria, an empty list is displayed. | | |
| Frequency of Use: | Medium (Recruiters filter candidates as needed). | | |
| Assumptions: | The filters work accurately. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-0012 | | |
| Use Case Name | Evaluate Applicant CVs | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Lead Recruiter | | |
| Description | A lead recruiter evaluates applicant CVs. | | |
| Precondition: | The recruiter must have a list of filtered applicants. | | |
| Post condition | The recruiter selects candidates for the next stage. | | |
| Normal Flow of Events / Basic Flow: | 1. The recruiter selects an applicant. 2. They review the CV and qualifications. 3. The recruiter marks the candidate as shortlisted/rejected. | | |
| Alternative Flow: | - The recruiter may leave notes for other team members. | | |
| Exceptions: | - If a CV is missing, it cannot be evaluated. | | |
| Frequency of Use: | Medium (Recruiters evaluate CVs before interviews). | | |
| Assumptions: | The CVs are genuine and contain relevant details. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-0013 | | |
| Use Case Name | Recommend Jobs | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Lead Recruiter | | |
| Description | The system recommends jobs to job seekers based on their profiles. | | |
| Precondition: | Job seekers must have an updated profile. | | |
| Post condition | Job seekers receive job recommendations. | | |
| Normal Flow of Events / Basic Flow: | 1. The system scans job seeker profiles. 2. It matches them with job openings. 3. Job recommendations are displayed. | | |
| Alternative Flow: | - If no exact match is found, similar jobs are suggested. | | |
| Exceptions: | - If job postings are outdated, recommendations may be inaccurate. | | |
| Frequency of Use: | High (Job recommendations are frequently updated). | | |
| Assumptions: | The matching algorithm is efficient. | | |

**Document 7- Screens and pages**

**Document 8- Tools-Visio and Axure**

**Visio Usage**

In my NextGreek project, I extensively used Visio to create detailed activity diagrams and UML diagrams that mapped out the core recruitment workflows. These included candidate sourcing, interview scheduling, offer management, and pipeline tracking. The visual representations helped break down complex hiring processes into structured workflows, making it easier for developers and stakeholders to understand system functionalities. I ensured that the diagrams accounted for real-world scenarios, such as handling duplicate candidate profiles, rescheduling interviews, and updating hiring statuses across multiple recruiters and clients. By incorporating alternate flows and decision points, I helped the team identify gaps and improve system logic, particularly in areas like automated candidate matching and performance tracking. Regular updates to these diagrams ensured that they remained aligned with evolving project requirements and business goals.

**Axure Usage**

Axure played a crucial role in prototyping interactive UI designs for key recruitment processes. I leveraged Axure to create interactive prototypes for key recruitment processes, ensuring that UI/UX elements were intuitive and aligned with business needs. I developed high-fidelity prototypes for modules like job posting, applicant tracking, team performance analytics, and client-specific hiring dashboards. These prototypes allowed stakeholders to interact with the interface, simulate user actions, and validate functionalities before development. This hands-on testing was particularly valuable for refining features such as resume parsing, automated candidate ranking, and real-time communication tools. Feedback from hiring managers and recruiters helped improve dashboard layouts, filtering mechanisms, and candidate engagement workflows.

**Overall Experience**

Overall, using Visio and Axure in the NextGreek project played a vital role in bridging the gap between business requirements and technical execution. Visio helped streamline requirement documentation by providing structured process flows, while Axure ensured that stakeholder expectations were met through interactive and iterative design improvements. By integrating these tools, I enhanced collaboration between teams, minimized system ambiguities, and contributed to a more efficient and user-friendly recruitment platform.

**Document 9- BA experience**

1. **Requirement gathering:**

* In this project, I leveraged my expertise as a Business Analyst to gather comprehensive requirements using the MoSCoW technique.
* I conducted stakeholder interviews, workshops, and surveys to document the needs and expectations of recruiters, hiring managers, and IT staff.
* I validated the requirements using the FURPS technique (Functionality, Usability, Reliability, Performance, and Supportability).
* Various elicitation techniques such as use case analysis, process mapping, and prototyping were utilized to ensure a thorough understanding of functional and non-functional requirements.

1. **Requirement Analysis:**

* After gathering requirements, I categorized and prioritized them based on business needs, feasibility, and technical constraints.
* Created process flow diagrams and use case diagrams to map system behavior and workflow.
* Conducted gap analysis to identify missing functionalities and potential improvements.
* Worked closely with stakeholders to clarify and refine ambiguous requirements.
* Documented business rules, data flow diagrams (DFD), and system context diagrams for better understanding.

1. **Design:**

* Collaborated with UI/UX designers to create wireframes and prototypes using Balsamiq & Visio.
* Ensured user experience (UX) best practices were implemented for better usability.
* Defined data models and prepared ER diagrams to ensure seamless database integration.
* Designed the system workflow and functional specifications document (FSD).
* Provided inputs for API integrations based on business needs.

1. **Development:**

* Acted as a liaison between business and technical teams to ensure the requirements were implemented correctly.
* Conducted sprint planning and backlog refinement meetings in Agile methodology.
* Reviewed development progress and ensured adherence to requirement specifications.
* Addressed developer queries and provided clarifications when needed.
* Updated documentation in case of change requests or scope modifications.

1. **Testing:**

* Collaborated with the QA team to define test cases, test scripts, and acceptance criteria.
* Conducted User Acceptance Testing (UAT) and gathered feedback from stakeholders.
* Ensured that functional, performance, and security aspects were tested thoroughly.
* Identified and logged defects, followed up for resolution, and ensured smooth bug tracking.
* Verified that all business requirements were met before deployment.

1. **Deployment:**

* Assisted in Go-Live planning and post-deployment activities.
* Ensured all training materials, user guides, and SOPs were ready for end-users.
* Conducted training sessions and knowledge transfer for recruiters and hiring managers.
* Monitored system performance after deployment and gathered feedback for further improvements.
* Ensured a smooth transition and continued support for any post-deployment issues.