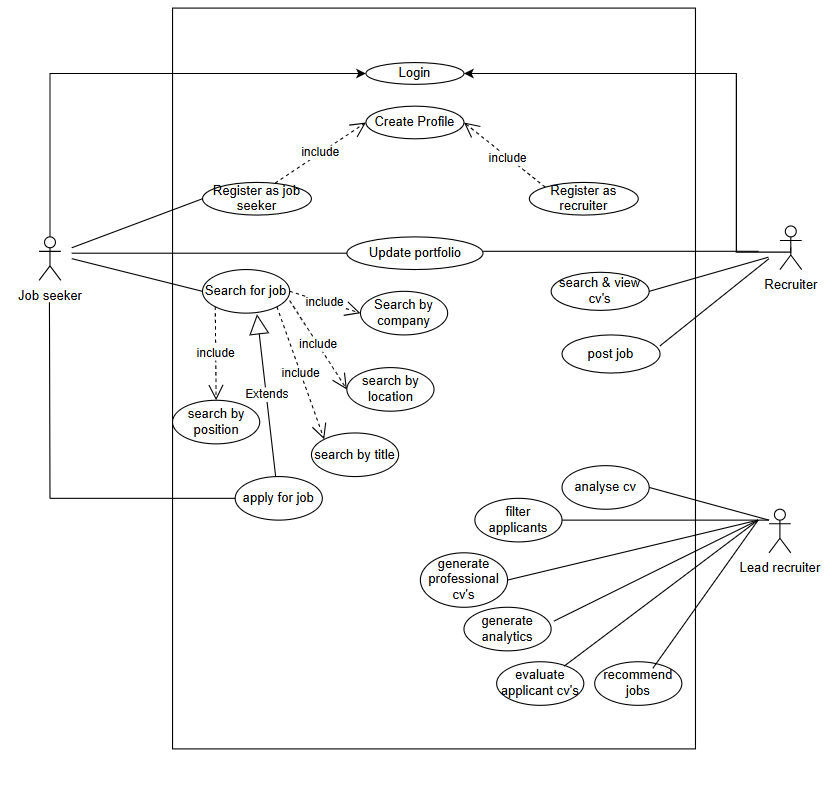
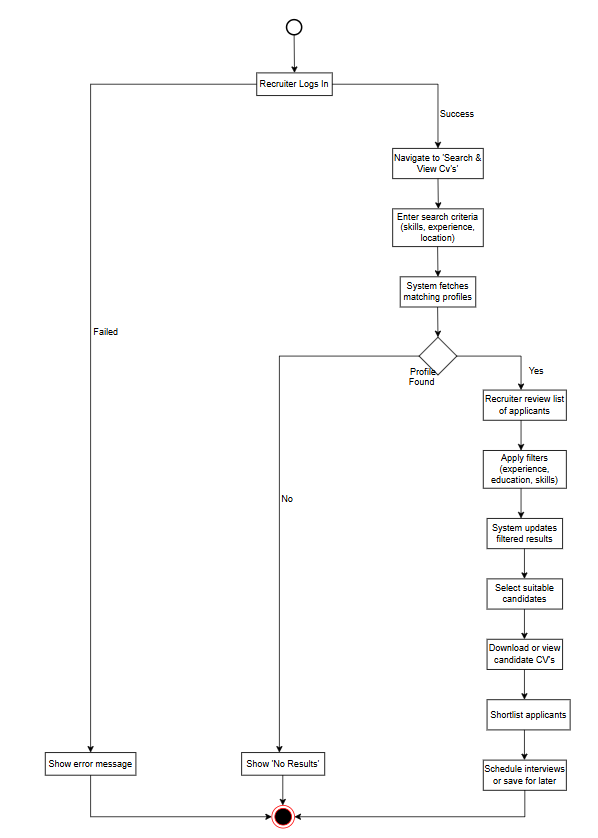
**Document 6- Prepare a use case diagram, activity diagram & a use case specification document**

**Use Case Diagram**

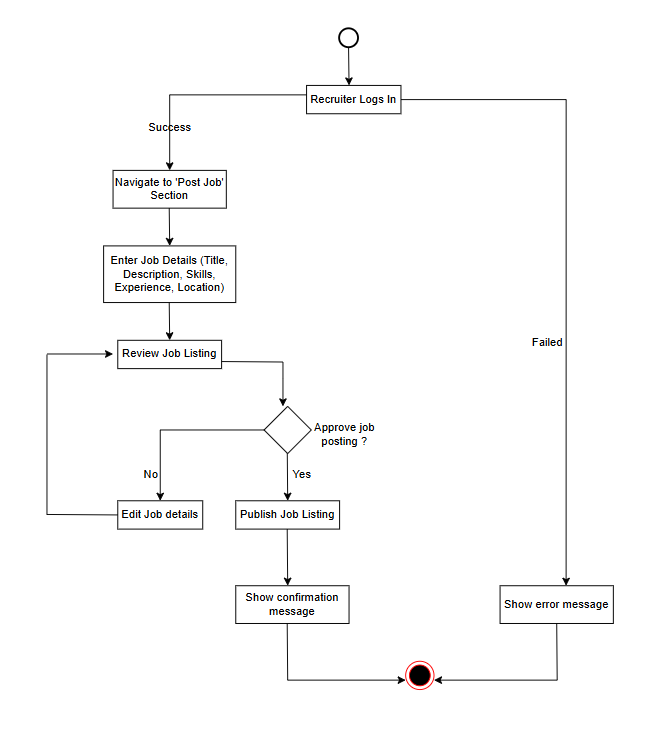
****

**Activity Diagram**

Search & Shortlisting Candidates

****

Job Posting:



**Use case documents**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-001 | | |
| Use Case Name | Search and Shortlist Applicants | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Primary Actor: Recruiter  Secondary Actor: System (NextGreek Platform) | | |
| Description | This use case describes how a recruiter search for candidates based on specific criteria and shortlists them for further review or interview scheduling. | | |
| Precondition: | - The recruiter must be logged into the system.  - The recruiter must have valid access permissions to search for and shortlist candidates. | | |
| Post condition | - The recruiter successfully views, filters, and shortlists candidates.  - Shortlisted candidates may be contacted for interviews or saved for future reference. | | |
| Normal Flow of Events / Basic Flow: | 1. Recruiter logs into the system.  2. Recruiter navigates to the "Search & Shortlist Candidates" section.  3. Recruiter enters search criteria such as skills, experience, and location.  4. The system fetches matching candidate profiles.  5. If profiles are found, the recruiter reviews the list of applicants.  6. Recruiter applies additional filters (experience, education, skills).  7. System updates the filtered results.  8. Recruiter selects suitable candidates.  9. Recruiter downloads or views candidate CVs.  10. Recruiter shortlists candidates.  11. Recruiter schedules interviews or saves candidates for later. | | |
| Alternative Flow: | - If no profiles match the search criteria, the system displays a "No Results" message. | | |
| Exceptions: | - If the system encounters an error, it displays an error message.  - If the recruiter enters invalid search criteria, the system prompts for corrections. | | |
| Frequency of Use: | High (Recruiters frequently search for and shortlist candidates). | | |
| Assumption | - Candidates have uploaded their CVs and provided relevant details.  - The system database contains enough candidate profiles for effective searches. | | |
| Constraints | - The search must adhere to platform policies and filtering rules.  - Large database queries may take additional processing time. | | |
| Dependencies | - The recruiter’s account must have appropriate permissions.  - The system database must be up to date with candidate profiles. | | |
| Inputs and Outputs | Inputs: Search criteria (skills, experience, location, education, etc.).  Outputs: List of matching candidate profiles, filtered results, shortlisted candidates, or error messages. | | |
| Business Rules | - Recruiters can only access candidate profiles as per platform privacy policies.  - Filters must be applied based on predefined criteria.  - Candidates must have a complete profile for visibility in searches. | | |
| Miscellaneous Information | - Recruiters should be able to save or bookmark candidates for later review.  - The system should allow bulk downloading of candidate CVs. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | UC-002 | | |
| Use Case Name | Post a Job | | |
| Created by | Vishnu UT | Last updated by | 10-Mar-25 |
| Date created | Feb-25 | Last revision date | 16-Mar-25 |
| Actors | Primary Actor: Recruiter  Secondary Actor: System (NextGreek Platform) | | |
| Description | This case describes how a recruiter posts a job listing on the platform. | | |
| Precondition: | - The recruiter must be logged in.  - The recruiter must have valid permissions to post jobs. | | |
| Post condition | - The job listing is successfully published and visible to job seekers.  - A confirmation message is displayed to the recruiter. | | |
| Normal Flow of Events / Basic Flow: | 1. Recruiter logs into the system.  2. Recruiter navigates to the "Post Job" section.  3. Recruiter enters job details (title, description, skills required, experience level, location, etc.).  4. The system allows the recruiter to review the job details.  5. Recruiter approves and submits the job posting.  6. System publishes the job and displays a confirmation message. | | |
| Alternative Flow: | - If the recruiter wants to edit job details before publishing, they can return to the job form and make changes. | | |
| Exceptions: | - If required fields are missing, the system prompts an error message.  - If the system fails to save the job due to technical issues, an error message is displayed.  - If the recruiter does not have the required permissions, they are restricted from posting the job. | | |
| Frequency of Use: | High (Recruiters post jobs frequently). | | |
| Assumption | - Recruiters have the necessary permissions to post jobs.  - The system is available and functional. | | |
| Constraints | - The job listing must meet the platform's guidelines and policies.  - Certain job categories may require additional approval. | | |
| Dependencies | - The recruiter’s account must be verified.  - Database service must be available to store job details. | | |
| Inputs and Outputs | Inputs: Job details (Title, Description, Skills, Experience, Location, Salary, etc.).  Outputs: Published job listing, confirmation message, or error message. | | |
| Business Rules | - Only verified recruiters can post jobs.  - Job listings must meet platform standards and policies.  - Duplicate job postings should be flagged. | | |
| Miscellaneous Information | - The system should provide an option to save job postings as drafts.  - The recruiter should be able to edit or delete a job post after publishing. | | |

**Document 7- Screens and pages**

Login Page

**A screenshot of a login screen

AI-generated content may be incorrect.**

Search Page

**A screenshot of a search box

AI-generated content may be incorrect.**

Job Posting Page

**A screenshot of a computer

AI-generated content may be incorrect.**

**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.