**Application Tracking System**

**Waterfall Model Documents**

**Waterfall Project2 – Part -2/2**

**Document 6 - Please prepare a use case diagram, activity diagram and a use case specification document.**

**About Application Tracking System**

The Application Tracking System (ATS) is designed to streamline and automate the recruitment process for organizations. It helps HR teams and recruiters manage job postings, track candidate applications, schedule interviews, and generate reports, all in one centralized platform**.**

**Use Case Diagram**





**Use Case Specification:-**

1. **Login**

|  |  |
| --- | --- |
| Use case Name | User Login to ATS |
| Use case Description | This use case describes the process where users (Recruiters, Hiring Managers) log into the ATS using their credentials. It ensures secure authentication and access control based on roles. |
| Primary Actor | **Recruiter** – Logs in to manage job postings and candidates.**Hiring Manager** – Logs in to review applications and provide feedback. |
| Secondary Actor | **System Administrator** – Manages user roles and authentication settings.**Third-party Authentication Provider** – Validates login credentials (e.g., OAuth, SSO). |
| Basic Flow | The actor navigates to the ATS login page. The actor enters a valid username/email and password. The system validates the credentials against the database. If authentication is successful, the system identifies the user's role. The system redirects the actor to the respective dashboard (Recruiter, Hiring Manager, or Candidate). The actor gains access to ATS features based on their role. |
| ALTERNATE FLOW | A: Login via Single Sign-On (SSO):  The actor selects "Login with Google/Microsoft." The system redirects the user to the third-party authentication provider. The third-party provider validates the credentials. If successful, the user is redirected to their ATS dashboard.5B: Remember Me Option:  The actor selects the "Remember Me" checkbox before logging in. The system securely stores the session for a pre-defined period. On the next visit, the actor is automatically logged in. |
| Exceptional flows | A: Invalid Credentials:  The actor enters an incorrect username/email or password. The system displays an error message: "Invalid username or password." The actor is prompted to re-enter the correct details. B: Account Locked Due to Multiple Failed Attempts:  The actor enters incorrect credentials 5 times consecutively. The system locks the account for 15 minutes and sends an email notification. The actor must either wait or reset their password. C: Password Reset Flow:  The actor clicks on "Forgot Password?" The system prompts the actor to enter their registered email. The system sends a password reset link via email. The actor clicks the link, enters a new password, and confirms the change. |
| Pre- Conditions | The user must be registered in the ATS system. The system must be connected to the authentication database. If using SSO, third-party authentication services must be available. |
| post-conditions | If successful, the user is logged into the system and redirected to the correct dashboard. If unsuccessful, an appropriate error message is displayed. The system logs the login attempt for security and audit purposes |
| Assumptions | The user has an active internet connection. The user remembers their login credentials. The authentication database is accessible and operational. |
| Constraints | The login process should be completed within 5 seconds. The system should support multi-factor authentication (MFA) if enabled. Password complexity rules (e.g., minimum 8 characters, at least one special character) must be enforced. |
| Dependencies | The login system depends on: The user database for authentication. External SSO providers if applicable. Email service for password reset. |
| Inputs and Outputs | Inputs:  Username/Email Password Optional: SSO credentials (Google, Microsoft) Outputs:  Access to the ATS dashboard (if successful) Error messages (if login fails) Security alerts (if multiple failed attempts) |
| Business Rules | Users must use valid credentials to access the system. Password complexity rules must be enforced. If MFA is enabled, the user must enter an OTP. Idle users should be logged out after 15 minutes of inactivity. A user cannot have multiple active sessions (if session management is enforced). |
| Miscellaneous Information | Future enhancements may include biometric authentication (fingerprint/face ID). Login security measures should comply with GDPR and industry best practices. The system should support mobile-friendly login for better accessibility. |

1. **Submit Resume in ATS**

|  |  |
| --- | --- |
| Use case Name | **Submit Resume in ATS** |
| Use case Description | This use case describes the process where a candidate submits their resume to apply for a job in the ATS. The system validates and stores the resume, extracts relevant details, and updates the candidate’s profile. |
| Primary Actor | Recruiter – Uploads resume in ATS |
| Secondary Actor | System – Parses and extracts resume details. Third-party Resume Parser – Processes the resume to extract key information (optional). |
| Basic Flow | The Recruiter logs into the ATS and navigates to the job application page. TheRecruiter clicks "Upload Resume" and selects a file from their device. The System validates the file format and size. If valid, the resume is uploaded and stored in the ATS. The System extracts key details (name, contact info, skills, experience) using the Resume Parser. The extracted details are displayed for candidate review and edits if necessary. The Candidate confirms the submission. The System associates the resume with the job application and updates the candidate profile. A confirmation message is displayed: "Resume submitted successfully!" The System notifies the Recruiter that a new resume has been submitted |
| ALTERNATE FLOW | Instead of uploading a new file, the Recruiter selects a previously uploaded resume. The ATS attaches the existing resume to the job application. The Recruiter reviews and confirms the submission. The Recruiter emails the resume to a designated ATS email address. The System automatically associates the resume with the candidate profile |
| Exceptional flows | Resume Parsing Fails  The System fails to extract information from the resume due to an unreadable format. The recruiter is prompted to manually enter their details. 6D: Session Timeout  The recruiter takes too long to upload the resume, and their session expires. The System logs the Candidate out, requiring them to log in again. |
| Pre- Conditions | The recruiter must have an active ATS account. The recruiter must have a valid resume file in the correct format. The ATS must be connected to the resume parsing service (if applicable) |
| post-conditions | If successful, the resume is uploaded and linked to the job application. If unsuccessful, the candidate receives an error message and must retry. The Recruiter is notified of the resume submission. |
| Assumptions | The Recruiter has a resume ready for submission. The resume parsing service is functioning correctly. The Recruiter will review the resume manually if parsing fails. |
| Constraints | Maximum resume file size: 5MB. Supported file formats: PDF, DOCX. A recruiter can upload only one resume per application. Resume submission should not take more than 10 seconds. |
| Dependencies | The resume parser service (if used). The candidate database where resumes are stored. The notification system that alerts recruiters. |
| Inputs and Outputs | Inputs:  Resume file (PDF, DOCX). Candidate’s LinkedIn profile (if using "Apply with LinkedIn"). Outputs:  Resume stored in the ATS. Extracted candidate details displayed for review. Confirmation message and recruiter notification. |
| Business Rules | A resume must be attached to complete the job application. The most recent resume submitted is used for review. If parsing fails, manual input is required. recruiter cannot submit multiple resumes for the same job. If a resume is submitted via email, it must be linked to the correct job posting. |
| Miscellaneous Information | Future enhancements may include AI-powered resume analysis for better job matching. Resume storage should comply with GDPR and data privacy policies. Mobile ATS users should be able to upload resumes from cloud storage (Google Drive, OneDrive, etc.). |

1. **Review Application :-**

|  |  |
| --- | --- |
| Use case Name | **Reviews Application** |
| Use case Description | This use case describes the process where a recruiter reviews a job application in the ATS, evaluates the candidate’s resume, and updates the application status accordingly. |
| Primary Actor | Primary Actor:  Recruiter – Reviews job applications, provides feedback, and moves candidates to the next stage. |
| Secondary Actor | Candidate – Receives status updates on the application. Hiring Manager – May review the recruiter’s feedback before scheduling interviews. ATS System – Stores and processes job applications, resumes, and recruiter feedback. Resume Parser (if applicable) – Extracts key candidate details for quick review. |
| Basic Flow | Recruiter logs into the ATS and navigates to the "Job Applications" section. The recruiter selects a job posting and views the list of applicants. The recruiter clicks on an application to view the candidate’s resume, cover letter, and extracted details. The recruiter evaluates the candidate’s profile based on qualifications, experience, and job requirements. The recruiter performs one of the following actions: Shortlists the candidate for the next stage. Rejects the candidate with a reason. Moves the candidate to a talent pool for future opportunities. The ATS updates the application status and sends a notification to the candidate. The recruiter adds notes or comments for the hiring manager if necessary. If the candidate is shortlisted, the recruiter schedules the next steps (e.g., interview or assessment). |
| ALTERNATE FLOW | Recruiter Requests More Information from Candidate The recruiter finds the resume insufficient for evaluation. The recruiter requests additional details (e.g., work samples, certifications). The candidate receives an email and uploads the required documents. The recruiter re-evaluates the application after receiving the documents. B: Recruiter Assigns the Application to Another Reviewer The recruiter decides to assign the application to another recruiter or hiring manager. The ATS allows reassignment with comments. The new reviewer receives a notification and proceeds with the review. C: Automatic Candidate Scoring Based on AI Screening The ATS automatically ranks candidates based on predefined criteria. The recruiter filters applications based on scores and prioritizes high-ranking candidates. |
| Exceptional flows | Missing or Incomplete Resume The recruiter finds the resume missing or unreadable. The recruiter requests a new resume from the candidate. The candidate uploads the resume, and the recruiter resumes the review. B: ATS System Downtime or Slow Response The recruiter is unable to access applications due to system downtime. The recruiter waits for system recovery or contacts IT support. The ATS logs the issue and notifies the recruiter when the system is restored. C: Candidate Withdraws Application Midway The recruiter opens an application but finds it marked as withdrawn. The recruiter cannot proceed and receives a system notification. |
| Pre- Conditions | The recruiter must have an active ATS account with appropriate access. Candidates must have submitted their applications successfully. The ATS system must be functional and accessible. |
| post-conditions | The candidate’s application status is updated in the ATS. The recruiter’s feedback is recorded in the system. The candidate is notified of the application decision. The hiring manager (if involved) has access to recruiter comments. |
| Assumptions | The recruiter is familiar with the ATS interface. The candidate’s resume and application details are complete and readable. The ATS sends notifications automatically after status updates. |
| Constraints | The recruiter must complete reviews within a defined SLA (e.g., within 7 days). The ATS must support bulk review and filtering for efficiency. Application data must comply with data protection regulations (e.g., GDPR, CCPA). |
| Dependencies | The candidate database storing application details. The resume parser (if used) to extract resume information. The notification system to inform candidates of updates. The hiring manager’s review process (if applicable). |
| Inputs and Outputs | Inputs:  Job application (resume, cover letter, profile details). Recruiter feedback/comments. Outputs:  Updated application status (Shortlisted, Rejected, Under Review, etc.). Notifications sent to candidates and hiring managers. Recruiter notes for hiring managers. |
| Business Rules | Recruiters must review applications within the defined SLA. Candidates cannot edit their application after submission. Recruiters must provide a rejection reason if rejecting a candidate. Recruiters must follow company-defined hiring criteria for shortlisting. If AI-based ranking is enabled, it must be used as a guideline, not the final decision. |
| Miscellaneous Information | Future enhancements may include automated candidate matching with job descriptions. Integration with LinkedIn or other job platforms for enhanced candidate profiles. Option to add video screening for candidate evaluations. |

1. **Shortlist Candidates :-**

|  |  |
| --- | --- |
| Use case Name | **Shortlists Candidate** |
| Use case Description | This use case describes how a recruiter shortlists a candidate for the next stage of the hiring process in the ATS. The recruiter reviews the candidate’s application, selects the "Shortlist" option, and updates the application status, triggering further actions like interview scheduling. |
| Primary Actor | Recruiter – Responsible for shortlisting candidates for the next round. |
| Secondary Actor | Hiring Manager – Reviews shortlisted candidates and schedules interviews.  ATS System – Updates the candidate’s status, notifies stakeholders, and integrates with interview scheduling tools. |
| Basic Flow | The Recruiter logs into the ATS and navigates to the "Job Applications" section. The Recruiter selects a specific job posting to view its applicants. The Recruiter opens a candidate’s application and reviews their resume, cover letter, and profile details. The Recruiter selects the "Shortlist" option. The ATS updates the candidate’s status to "Shortlisted" and saves the recruiter’s decision. The System notifies the Candidate via email or ATS dashboard about the update. The ATS sends the shortlisted candidate’s profile to the Hiring Manager. The Recruiter can add comments or feedback for further review. If required, the Recruiter schedules an interview or assigns the candidate to a talent pipeline for future roles. |
| ALTERNATE FLOW | 5A: Recruiter Adds Candidate to a Talent Pool Instead of Shortlisting The Recruiter decides that the candidate is suitable but not for the current role. Instead of shortlisting, the Recruiter adds the candidate to a talent pool for future opportunities. The ATS stores the profile under a relevant category (e.g., "Future Consideration"). The candidate is notified that their profile is kept for future roles. 5B: Automated Shortlisting Based on AI Scoring The ATS uses AI to rank candidates based on predefined criteria. The recruiter filters and selects high-scoring candidates for manual review. The recruiter shortlists candidates based on AI recommendations or adjusts the selection manually. 5C: Shortlisting Requires Hiring Manager Approval The Recruiter forwards the candidate profile to the Hiring Manager for final approval. The Hiring Manager reviews and approves or rejects the shortlisting decision. If approved, the ATS updates the candidate’s status to "Shortlisted." If rejected, the recruiter receives feedback and takes further action. |
| Exceptional flows | 6A: Candidate Withdraws Application Before Shortlisting The recruiter attempts to shortlist a candidate. The ATS shows an error: "Candidate has withdrawn their application." The recruiter cannot proceed and updates records accordingly. 6B: System Error or Downtime During Shortlisting The recruiter tries to shortlist a candidate, but the ATS experiences a system error. The ATS logs the error, and the recruiter is notified to retry later. The recruiter either waits for system recovery or contacts IT support. 6C: Duplicate Application Detected The recruiter attempts to shortlist a candidate, but the ATS flags duplicate applications for the same job. The recruiter reviews the flagged applications and merges or deletes duplicates before proceeding. |
| Pre- Conditions | The candidate’s application must be complete and accessible in the ATS. The recruiter must have appropriate access to shortlist candidates. The ATS must be functioning correctly without system errors. |
| post-conditions | The candidate’s status is updated to "Shortlisted." The candidate receives a notification regarding their updated status. The hiring manager (if applicable) is notified of the shortlisted candidates. The candidate is either moved to the next round or placed in a talent pool. |
| Assumptions | The recruiter has clear criteria for shortlisting candidates. The ATS automatically tracks and logs shortlisting decisions. Candidates regularly check their notifications for updates. |
| Constraints | The recruiter can only shortlist a limited number of candidates per role based on hiring policies. The ATS must support bulk shortlisting for large applicant volumes. Candidate information must be stored securely to comply with data privacy regulations. |
| Dependencies | Candidate database where applications are stored. Resume parser or AI-based ranking system (if applicable). Hiring manager feedback process (if required for approval). Notification system for informing candidates of status updates. |
| Inputs and Outputs | Inputs:  Candidate application (resume, cover letter, profile details). Recruiter’s shortlisting decision and comments. Outputs:  Updated application status ("Shortlisted"). Notification to the candidate about their shortlisting. Hiring manager receives candidate details for further processing |
| Business Rules | Recruiters must only shortlist candidates who meet job criteria. Shortlisting should be completed within the specified SLA (e.g., within 7 days of application submission). The ATS must log all shortlisting actions for audit purposes. Candidates who are shortlisted but not selected can be moved to a talent pool for future roles. If multiple recruiters review the same role, they must ensure no duplicate shortlisting occurs. |
| Miscellaneous Information | Future enhancements may include automated shortlisting using AI-driven candidate ranking. ATS integration with calendars to allow recruiters to schedule interviews immediately after shortlisting. Option to bulk shortlist candidates for high-volume hiring scenarios. Compliance checks to ensure shortlisting follows diversity and inclusion policies. |

1. **Schedule Interviews**

|  |  |
| --- | --- |
| Use case Name | Schedule Interview |
| Use case Description | This use case describes the process where a recruiter schedules an interview for a shortlisted candidate through the ATS. The recruiter selects an available time slot, invites interviewers, and sends a notification to the candidate. |
| Primary Actor | Recruiter – Schedules the interview, selects time slots, and sends invitations. |
| Secondary Actor | Candidate – Receives an interview invitation and confirms attendance. Hiring Manager/Interviewer – Participates in the interview and provides feedback. ATS System – Manages interview scheduling, sends notifications, and tracks interview statuses. Calendar System (e.g., Outlook, Google Calendar) – Syncs interview schedules with stakeholders’ calendars. Video Conferencing Tool (e.g., Zoom, Microsoft Teams, Google Meet) – Facilitates virtual interviews. |
| Basic Flow | Recruiter logs into the ATS and navigates to the shortlisted candidates' section. Recruiter selects a candidate and clicks the "Schedule Interview" option. The Recruiter selects an interview type (Phone, Virtual, In-Person). The Recruiter chooses an available time slot based on the interviewers' and candidate’s availability. The ATS integrates with the interviewers' calendars and suggests time slots. The Recruiter finalizes the interview details (date, time, duration, mode, location, panel members). The ATS generates an interview invite and sends it to the candidate and interviewers. The Candidate receives a notification with interview details and confirms attendance. The ATS updates the candidate’s application status to "Interview Scheduled." |
| ALTERNATE FLOW | 5A: Candidate Reschedules Interview Candidate requests a reschedule due to unavailability. The recruiter reviews the request and selects a new time slot. The ATS sends updated invitations to all participants. 5B: Recruiter Allows Candidate to Self-Schedule Instead of manually scheduling, the recruiter enables self-scheduling for the candidate. The ATS sends a link to the candidate to choose an available slot. The candidate selects a time, and the ATS automatically confirms the interview. 5C: Panel Interview with Multiple Interviewers The recruiter selects multiple interviewers for a panel interview. The ATS finds common available slots and suggests them. The recruiter selects a final time, and all interviewers receive invitations. 5D: Auto-Scheduling Based on System Availability Check The ATS automatically checks calendar availability for all stakeholders. The system suggests a slot and automatically books the interview if everyone is available. |
| Exceptional flows | 6A: Candidate Fails to Attend the Interview The candidate does not show up for the scheduled interview. The recruiter marks the interview as "No Show", and the ATS notifies the candidate. The recruiter decides whether to reschedule or reject the application. 6B: Interviewer is Unavailable at the Last Minute The interviewer cannot attend the scheduled interview. The recruiter assigns a backup interviewer or reschedules. The ATS notifies all stakeholders of the update. 6C: Technical Issues in Virtual Interview The candidate or interviewer faces technical issues (e.g., video call failure, connectivity issues). The recruiter reschedules the interview or switches to a phone interview. 6D: ATS System Failure During Scheduling The ATS fails to send invitations due to a system error. The recruiter manually contacts the candidate and interviewers to schedule. |
| Pre- Conditions | The candidate must be shortlisted for an interview. The recruiter must have access to scheduling features in the ATS. The interviewers' availability must be visible in the ATS calendar. The ATS must be integrated with email and calendar systems. |
| post-conditions | The candidate’s interview status is updated to "Scheduled" in the ATS. The candidate and interviewers receive confirmation emails with details. The interview is added to interviewers' and candidates' calendars. The ATS tracks interview completion status after the meeting. |
| Assumptions | The candidate and interviewers will be available at the scheduled time. The ATS syncs properly with external calendar systems. Candidates respond to interview invitations in a timely manner. |
| Constraints | The recruiter can only schedule interviews within business hours. Rescheduling is allowed only a limited number of times per candidate. Some companies may have specific approval workflows before scheduling. |
| Dependencies | Calendar System (Google, Outlook, etc.) – For scheduling and availability checks. Email/Notification System – For sending interview invites and reminders. Video Conferencing Tool (if virtual) – For online interviews. Candidate Database – Stores shortlisted candidates and interview details. |
| Inputs and Outputs | Inputs:  Candidate details (Name, Email, Resume). Recruiter-selected interview slot and interview type. Interviewer availability data. Outputs:  Interview invitation sent to candidate and interviewers. Updated interview schedule in ATS and calendar system. Notification to candidate and interviewers. |
| Business Rules | Recruiters must schedule interviews within a set timeframe after shortlisting. Interview invitations must include all relevant details (date, time, location, link). Candidates can reschedule only once/twice (as per company policy). Hiring Managers or Panel members must confirm availability before scheduling. Candidates who miss interviews without notice may be disqualified. |
| Miscellaneous Information | Future enhancements may include AI-powered scheduling based on candidate and interviewer behavior. ATS may introduce automated reminders to reduce no-shows. Integration with assessment tools to include pre-interview tests. Support for different interview formats (e.g., one-on-one, panel, case study). |

**6.Manage Job Posting**

|  |  |
| --- | --- |
| Use case Name | Manage Job Posting |
| Use case Description | This use case describes how a recruiter or HR manager creates, updates, publishes, and removes job postings within the Applicant Tracking System (ATS). It includes defining job details, selecting job boards, setting expiration dates, and tracking applications. |
| Primary Actor | Recruiter / HR Manager – Responsible for creating, updating, publishing, and removing job postings. |
| Secondary Actor | Hiring Manager – Reviews and approves job postings before publishing. ATS System – Stores, processes, and distributes job postings to internal and external job boards. Job Boards (LinkedIn, Indeed, Company Website, etc.) – External platforms where job postings are published. |
| Basic Flow | The Recruiter logs into the ATS and navigates to the "Manage Job Postings" section. The Recruiter clicks "Create New Job Posting" and enters details: Job Title, Job Description, Location, Salary Range, Employment Type Required Skills, Experience Level, Responsibilities, Benefits The Recruiter selects where to post (company website, job boards, internal portal). The Recruiter submits the job for approval (if required by company policy). The Hiring Manager reviews and approves the job posting. The ATS publishes the job posting on selected platforms. The System tracks applications submitted for the job. The Recruiter can edit or update the job posting if needed.B.Update an Existing Job Posting The recruiter selects an existing job posting from the ATS. The recruiter modifies job details (e.g., changing requirements, salary range, deadline). If major changes occur, re-approval may be required. The ATS updates the job posting across all platforms. C: Remove or Close a Job Posting The recruiter selects a job posting that needs to be closed. The recruiter marks the job as "Closed" in the ATS. The ATS removes the posting from job boards. The system retains application data for reporting and analytics. |
| ALTERNATE FLOW | Auto-Publish Jobs Based on Predefined Templates The recruiter selects a pre-approved job template. The job posting is automatically generated with pre-filled details. The recruiter reviews and publishes the job without manual entry. 5B: Multi-Location Job Posting The recruiter selects multiple locations for a single job posting. The ATS creates localized versions and posts them on relevant job boards. 5C: Internal-Only Job Posting The recruiter restricts visibility to internal employees for internal hiring. The ATS ensures external candidates cannot view/apply. |
| Exceptional flows | Missing Required Information The recruiter submits a job posting with missing fields. The ATS flags missing details and prevents submission. The recruiter must fill in the required fields before proceeding. 6B: Job Posting Rejected by Hiring Manager The hiring manager reviews the posting and rejects it with feedback. The recruiter revises the job details and resubmits for approval. 6C: System Failure or Job Board Integration Error The ATS fails to post the job due to API integration issues with external job boards. The recruiter is notified and can manually post the job on external platforms. 6D: Job Posting Expired Automatically The job reaches its expiration date without manual closure. The ATS removes the job from external job boards automatically. |
| Pre- Conditions | The recruiter must have the necessary access rights to create or modify job postings. The ATS must be integrated with job boards for external postings. The hiring manager must be available to approve job postings if required. |
| post-conditions | The job is successfully posted and visible to candidates. The recruiter receives applications and candidate data. The hiring process can proceed based on the posted job requirements. |
| Assumptions | Recruiters follow company hiring policies when posting jobs. External job boards accept the job posting format from the ATS. Candidates will apply through the ATS or linked job board platforms. |
| Constraints | Job postings must meet regulatory and compliance requirements. Some job boards may require additional fees or verification before posting. Job postings expire after a predefined duration (e.g., 30 days). |
| Dependencies | Company’s career portal for internal job postings. External job boards (LinkedIn, Indeed, Glassdoor, etc.). HR policies and approval workflows. ATS system settings for job posting automation. |
| Inputs and Outputs | Inputs:  Job title, description, salary range, employment type. Posting location and visibility settings. Recruiter’s selection of job boards and posting duration. Outputs:  Job posted successfully on selected platforms. Job posting ID generated in the ATS. Notification sent to hiring managers for approvals (if applicable). |
| Business Rules | Only authorized recruiters can create, edit, or close job postings. Job postings must be reviewed and approved before publication (if applicable). Jobs must have clear and complete descriptions to be published externally. Expired job postings must be archived after removal from job boards. Companies may restrict job postings to internal employees before external publishing. |
| Miscellaneous Information | Future enhancements may include AI-based job description recommendations. ATS could integrate with social media for job promotion. Job postings might include pre-screening questionnaires or assessments. Multi-language support for global job postings could be added. |

**7.Reject Application**

|  |  |
| --- | --- |
| Use case Name | Reject Application |
| Use case Description | This use case describes the process of rejecting a job application within an Applicant Tracking System (ATS). It ensures that applicants are notified about the rejection and that the system updates the application status accordingly. |
| Primary Actor | Recruiter |
| Secondary Actor | HR Manager, Applicant, ATS System |
| Basic Flow | The recruiter logs into the ATS.  The recruiter navigates to the list of job applications.  The recruiter selects an application to review.  The recruiter decides to reject the application.  The recruiter selects the rejection reason and submits it.  The system updates the application status to "Rejected."  The system sends an automated rejection notification to the applicant.  The recruiter’s dashboard is updated accordingly. |
| ALTERNATE FLOW | AF1:Rejection with Custom Message  Instead of selecting a predefined reason, the recruiter chooses to enter a custom rejection message.  The system allows manual input of a rejection message.  The system sends a rejection email with the custom message to the applicant.  AF2: HR Manager Overrides Rejection  The HR Manager reviews rejected applications.  If necessary, the HR Manager can override the rejection and reinstate the application.  The system updates the status and notifies the recruiter and the applicant. |
| Exceptional flows | EF1: System Error While Updating Status  The recruiter submits the rejection decision.  Due to a system failure, the status is not updated.  The system logs the error and notifies the recruiter.  The recruiter retries or reports the issue to technical support.  EF2: Applicant Email Not Found  The system attempts to send a rejection email.  If the applicant's email is missing or invalid, the system logs the error.  The recruiter is notified to take corrective action. |
| Pre- Conditions | The recruiter must be logged into the ATS.  The job application must exist in the system.  The recruiter must have the necessary permissions to reject an application. |
| post-conditions | The application status is updated to "Rejected."  The applicant receives a rejection notification.  The recruiter’s dashboard reflects the updated status. |
| Assumptions | Recruiters have the authority to reject applications.  The ATS has predefined rejection reasons.  The system supports automated email notifications. |
| Constraints | The rejection process must comply with company hiring policies.  The system should not allow multiple rejections for the same application. |
| Dependencies | Email notification service must be operational.  The ATS must have real-time status updates |
| Inputs and Outputs | Inputs: Application ID, rejection reason (predefined or custom), recruiter’s credentials.  Outputs: Status change confirmation, rejection email notification, recruiter’s dashboard update. |
| Business Rules | Only authorized personnel (recruiters and HR managers) can reject applications.  Rejected applications cannot be reinstated unless overridden by an HR manager.  Custom rejection messages must adhere to company communication guidelines. |
| Miscellaneous Information | The system logs all rejection actions for audit purposes.  Recruiters can filter applications based on their status (e.g., Pending, Rejected, Approved). |

**8.Finalize Hiring Decision**

|  |  |
| --- | --- |
| Use case Name | Approval Offer Letter |
| Use case Description | This use case describes the process of approving a job offer letter by the Hiring Manager within an Applicant Tracking System (ATS). It ensures that the offer letter is reviewed, approved, and forwarded for further processing. |
| Primary Actor | Hiring Manager |
| Secondary Actor | Recruiter, HR Manager, ATS System, Applicant |
| Basic Flow | The Hiring Manager logs into the ATS.  The Hiring Manager navigates to the pending offer letters.  The Hiring Manager reviews the details of the offer letter.  The Hiring Manager approves the offer letter.  The system updates the offer status to "Approved."  The system notifies the recruiter and HR Manager about the approval.  The system forwards the approved offer letter for further processing. |
| ALTERNATE FLOW | AF1: Request for Modifications  The Hiring Manager requests changes to the offer letter instead of approving it.  The system sends the modification request to the recruiter.  The recruiter updates the offer letter and resubmits it for approval.  AF2: HR Manager Overrides Approval  The HR Manager reviews an approved offer letter.  If necessary, the HR Manager can override the approval and request further changes.  The system notifies the Hiring Manager and recruiter of the override. |
| Exceptional flows | EF1: System Error While Updating Status  The Hiring Manager submits the approval decision.  Due to a system failure, the status is not updated.  The system logs the error and notifies the Hiring Manager.  The Hiring Manager retries or reports the issue to technical support.  EF2: Notification Failure  The system attempts to send an approval notification.  If the notification fails, the system logs the error.  The recruiter is notified to take corrective action. |
| Pre- Conditions | The Hiring Manager must be logged into the ATS.  The offer letter must exist in the system.  The Hiring Manager must have the necessary permissions to approve an offer letter. |
| post-conditions | The offer letter status is updated to "Approved."  The recruiter and HR Manager are notified of the approval.  The system forwards the approved offer letter for further processing. |
| Assumptions | Hiring Managers have the authority to approve offer letters.  The ATS provides an option to request modifications.  The system supports automated notifications. |
| Constraints | The approval process must comply with company hiring policies.  The system should not allow multiple approvals for the same offer letter without changes |
| Dependencies | Email notification service must be operational.  The ATS must have real-time status updates. |
| Inputs and Outputs | Inputs: Offer letter ID, Hiring Manager’s credentials, approval decision.  Outputs: Status change confirmation, approval notification, recruiter and HR Manager updates. |
| Business Rules | Only authorized personnel (Hiring Managers and HR Managers) can approve offer letters.  Approved offer letters cannot be modified unless requested by the Hiring Manager or HR Manager.  Approval notifications must be logged for audit purposes. |
| Miscellaneous Information | The system logs all approval actions for audit purposes.  Hiring Managers can track the status of offer letters in the ATS dashboard. |

**9.Manage User Roles**

|  |  |
| --- | --- |
| Use case Name | Manage User Roles |
| Use case Description | This use case describes the process of managing user roles within an Applicant Tracking System (ATS). It ensures that users have appropriate access and permissions based on their roles. |
| Primary Actor | System Administrator |
| Secondary Actor | Recruiter, Hiring Manager, HR Manager, ATS System |
| Basic Flow | The System Administrator logs into the ATS.  The System Administrator navigates to the user management section.  The System Administrator selects a user to manage roles.  The System Administrator assigns, modifies, or removes a role.  The system updates the user's role and permissions.  The system notifies the user about role changes.  The System Administrator verifies the changes in the system. |
| ALTERNATE FLOW | AF1: Role Change Approval Required  The System Administrator assigns a new role that requires HR approval.  The system sends a request to the HR Manager for review.  The HR Manager approves or rejects the role change.  If approved, the system updates the role; otherwise, the System Administrator is notified.  AF2: Bulk Role Assignment  The System Administrator selects multiple users for role changes.  The system processes bulk assignments and updates roles.  The system notifies affected users of their updated roles. |
| Exceptional flows | EF1: System Error While Updating Roles  The System Administrator submits a role change request.  Due to a system failure, the role is not updated.  The system logs the error and notifies the System Administrator.  The System Administrator retries or reports the issue to technical support.  EF2: Unauthorized Role Change Attempt  A user without sufficient permissions attempts to modify roles.  The system prevents the change and logs the unauthorized attempt.  The System Administrator is notified for review. |
| Pre- Conditions | The System Administrator must be logged into the ATS.  The user account to be modified must exist in the system.  The System Administrator must have the necessary permissions to manage roles |
| post-conditions | The user role is updated in the system.  The affected user is notified of role changes.  The system maintains an audit log of role modifications. |
| Assumptions | System Administrators have the authority to manage roles.  The ATS supports role-based access control.  Users are informed about role changes via notifications. |
| Constraints | Role modifications must comply with company policies.  The system should prevent conflicting roles for a single user |
| Dependencies | Email or system notification service must be operational.  The ATS must support real-time role updates. |
| Inputs and Outputs | Inputs: User ID, new role assignment, System Administrator’s credentials.  Outputs: Status change confirmation, user notification, system audit log update. |
| Business Rules | Only authorized personnel (System Administrators, HR Managers) can manage roles.  Role modifications should be logged for audit and compliance purposes.  Certain role changes may require approval from HR Managers. |
| Miscellaneous Information | The system logs all role management actions for security and compliance.  Administrators can filter and view user roles through the ATS dashboard. |

**10.Configure System Settings**

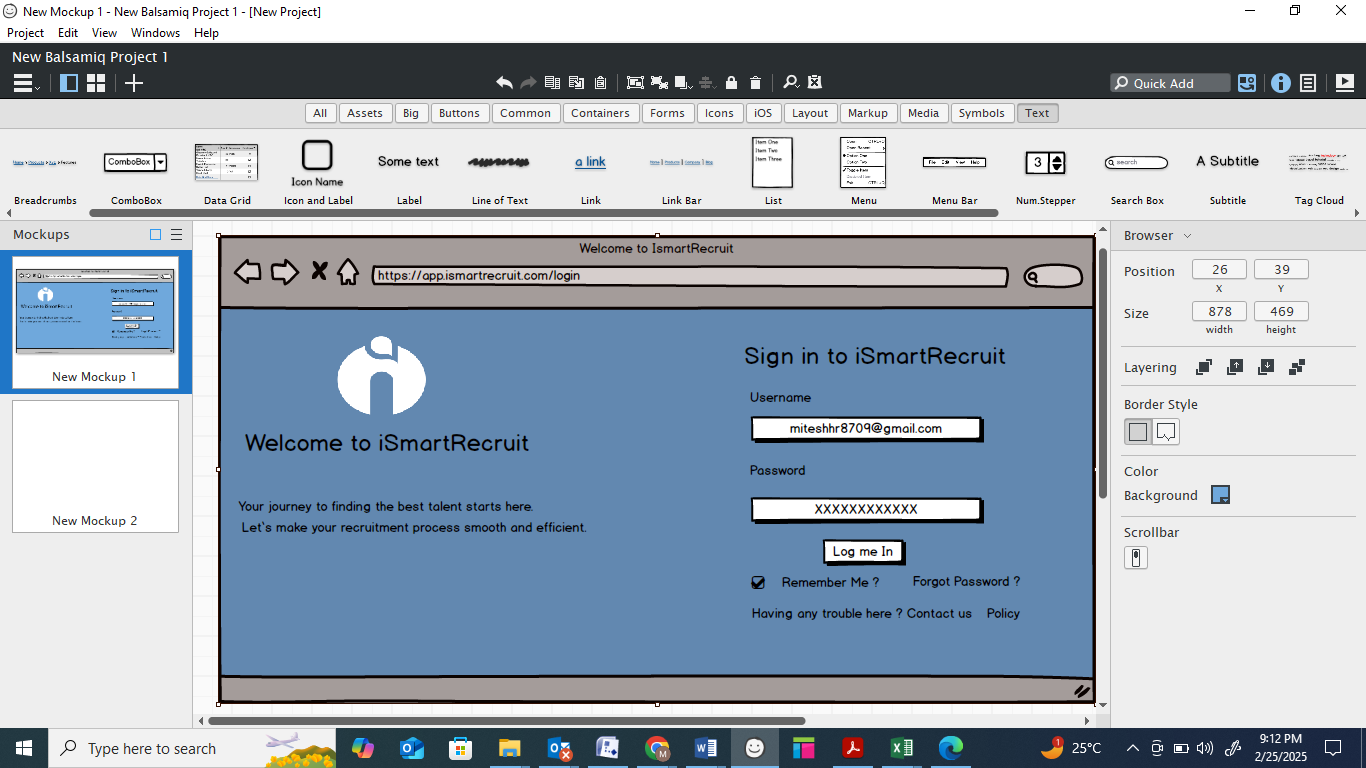
|  |  |
| --- | --- |
| Use case Name | Configure System Settings |
| Use case Description | This use case describes the process of configuring system settings within an Applicant Tracking System (ATS). It ensures that system administrators can modify global settings to align with organizational requirements. |
| Primary Actor | System Administrator |
| Secondary Actor | IT Support, HR Manager, ATS System |
| Basic Flow | The System Administrator logs into the ATS.  The System Administrator navigates to the system settings section.  The System Administrator selects a configuration category (e.g., user access, notification settings, workflow automation).  The System Administrator updates the necessary settings.  The system validates the changes.  The system applies the updates and saves the new configuration.  The system notifies relevant stakeholders about the configuration changes.  The System Administrator verifies the updates. |
| ALTERNATE FLOW | AF1: Configuration Requires Additional Approval  The System Administrator submits a configuration change that requires HR or IT approval.  The system sends a request to the HR Manager or IT Support for review.  The reviewer approves or rejects the changes.  If approved, the system applies the new settings; otherwise, the System Administrator is notified.AF2: Reverting to Previous Configuration  The System Administrator selects an option to revert settings to a previous state.  The system retrieves and applies the last saved configuration.  The system notifies relevant stakeholders of the reversion. |
| Exceptional flows | EF1: System Error While Applying Settings  The System Administrator submits a configuration update.  Due to a system failure, the settings are not updated.  The system logs the error and notifies the System Administrator.  The System Administrator retries or reports the issue to technical support.  EF2: Unauthorized Configuration Attempt  A user without sufficient permissions attempts to modify system settings.  The system prevents the change and logs the unauthorized attempt.  The System Administrator is notified for review. |
| Pre- Conditions | The System Administrator must be logged into the ATS.  The system settings must be accessible and modifiable.  The System Administrator must have the necessary permissions to update system settings. |
| post-conditions | The system settings are updated as per the configuration.  Relevant stakeholders are notified of configuration changes.  The system maintains an audit log of configuration modifications |
| Assumptions | System Administrators have the authority to configure system settings.  The ATS allows granular configuration options.  Users are informed about significant system configuration changes. |
| Constraints | Configuration changes must comply with company policies and security guidelines.  The system should prevent conflicting settings that may impact usability. |
| Dependencies | The ATS must support real-time configuration updates.  The notification system must be operational to inform users of changes. |
| Inputs and Outputs | Inputs: Configuration parameters, System Administrator’s credentials, optional approval inputs.  Outputs: Status change confirmation, configuration update logs, stakeholder notifications. |
| Business Rules | Only authorized personnel (System Administrators, IT Support) can configure system settings.  Configuration changes should be logged for security and audit purposes.  Certain system settings may require additional approval before implementation. |
| Miscellaneous Information | The system logs all configuration actions for compliance and troubleshooting.  Administrators can filter and view system settings history through the ATS dashboard. |

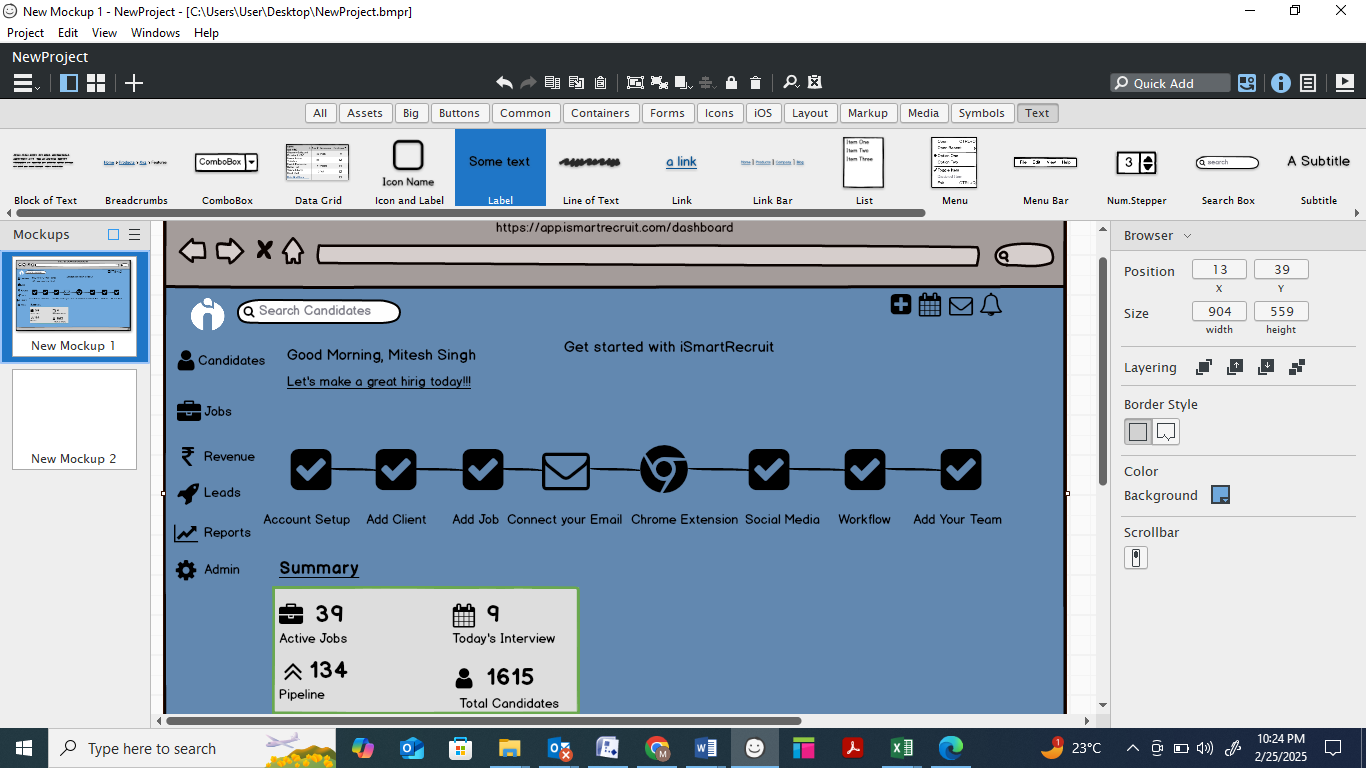
**11.Monitor System Usage**

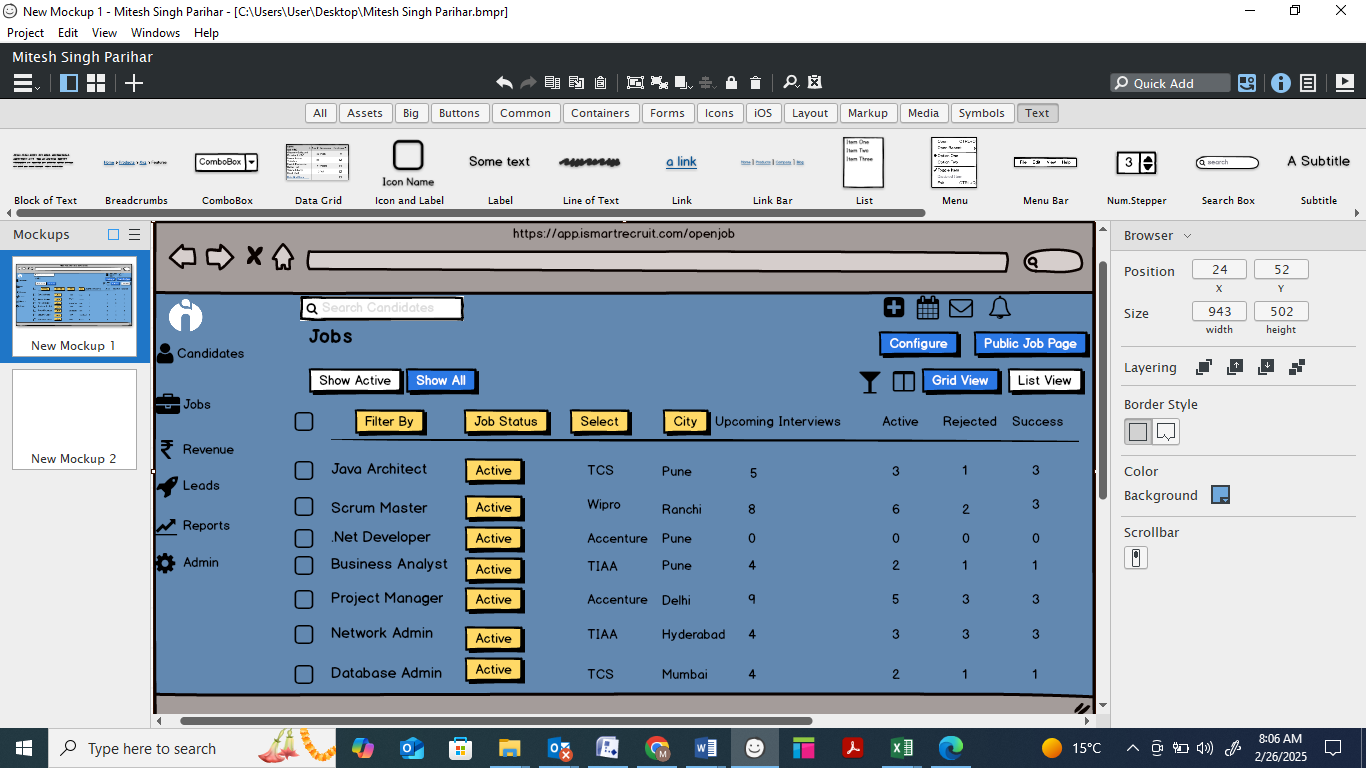
|  |  |
| --- | --- |
| Use case Name | Monitor System Usage |
| Use case Description | This use case describes the process of monitoring system usage within an Applicant Tracking System (ATS). It ensures that administrators can track system activity, user behavior, and resource utilization for security, compliance, and performance optimization |
| Primary Actor | System Administrator |
| Secondary Actor | IT Support, HR Manager, ATS System |
| Basic Flow | The System Administrator logs into the ATS.  The System Administrator navigates to the system usage dashboard.  The System Administrator selects the desired monitoring parameters (e.g., active users, login attempts, job postings, application submissions).  The system retrieves and displays real-time usage data.  The System Administrator analyzes the data for anomalies or trends.  The System Administrator generates reports if needed.  The system logs all monitoring activities for audit purposes. |
| ALTERNATE FLOW | AF1: Customizing System Usage Reports  The System Administrator customizes the monitoring report parameters (e.g., date range, user roles, specific activities).  The system filters and presents the relevant data.  The System Administrator exports or shares the report as needed.  AF2: Setting Usage Alerts  The System Administrator configures automated alerts for specific usage thresholds (e.g., excessive failed login attempts, unusual user activity).  The system monitors usage in real-time and triggers alerts when conditions are met.  The system notifies the relevant stakeholders of any alerts. |
| Exceptional flows | EF1: System Error While Retrieving Usage Data  The System Administrator requests system usage data.  Due to a system failure, the data is not retrieved.  The system logs the error and notifies the System Administrator.  The System Administrator retries or reports the issue to technical support.  EF2: Unauthorized Access to Usage Reports  A user without sufficient permissions attempts to access system usage reports.  The system prevents access and logs the unauthorized attempt.  The System Administrator is notified for review |
| Pre- Conditions | The System Administrator must be logged into the ATS.  The system usage dashboard must be accessible.  The System Administrator must have the necessary permissions to monitor system usage |
| post-conditions | The system usage data is retrieved and displayed.  Reports or alerts are generated as needed.  The system logs all monitoring activities for compliance and auditing. |
| Assumptions | System Administrators have the authority to monitor system usage.  The ATS provides real-time and historical system usage data.  Alerts can be configured for specific usage thresholds. |
| Constraints | Monitoring must comply with company policies and data privacy regulations.  The system should not allow unauthorized access to sensitive usage reports |
| Dependencies | The ATS must store and process usage data in real-time.  The notification system must be operational to send alerts. |
| Inputs and Outputs | Inputs: Monitoring parameters, System Administrator’s credentials.  Outputs: System usage dashboard data, reports, alerts, audit logs. |
| Business Rules | Only authorized personnel (System Administrators, IT Support) can monitor system usage.  System activity logs must be maintained for security and compliance.  Alerts should be configurable based on predefined thresholds. |
| Miscellaneous Information | The system logs all monitoring actions for audit and security purposes.  Administrators can filter and analyze system usage trends through the ATS dashboard. |

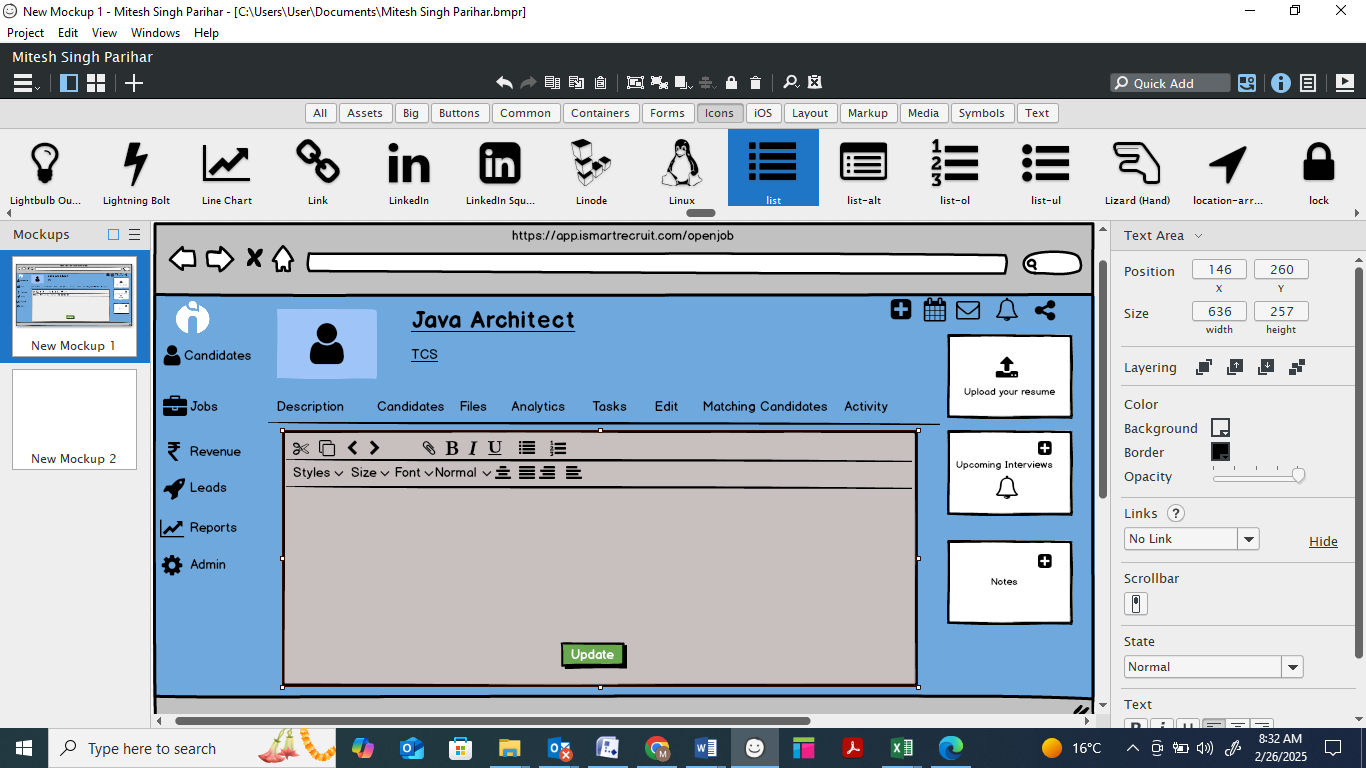
**Document 7- Screens and pages**

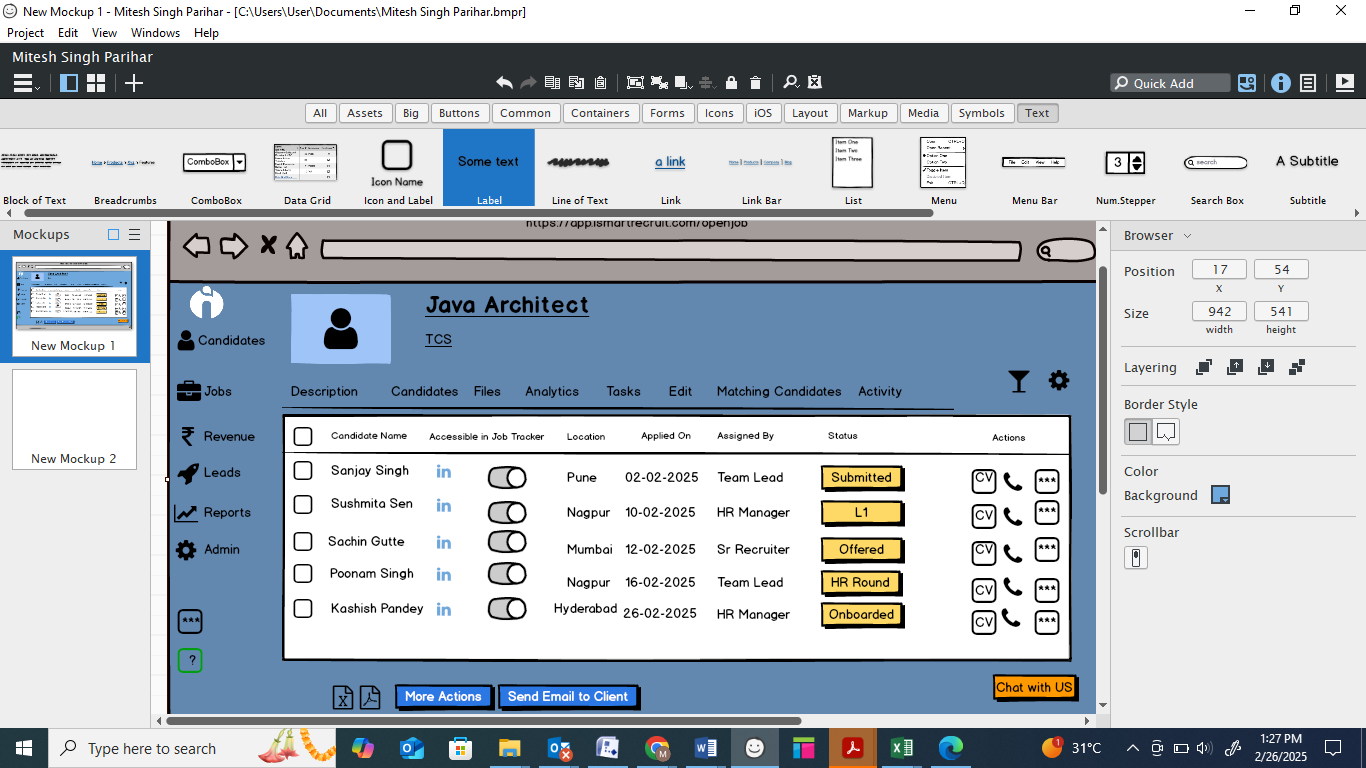
**Submission of Resume End to End ( Login to Logout )**

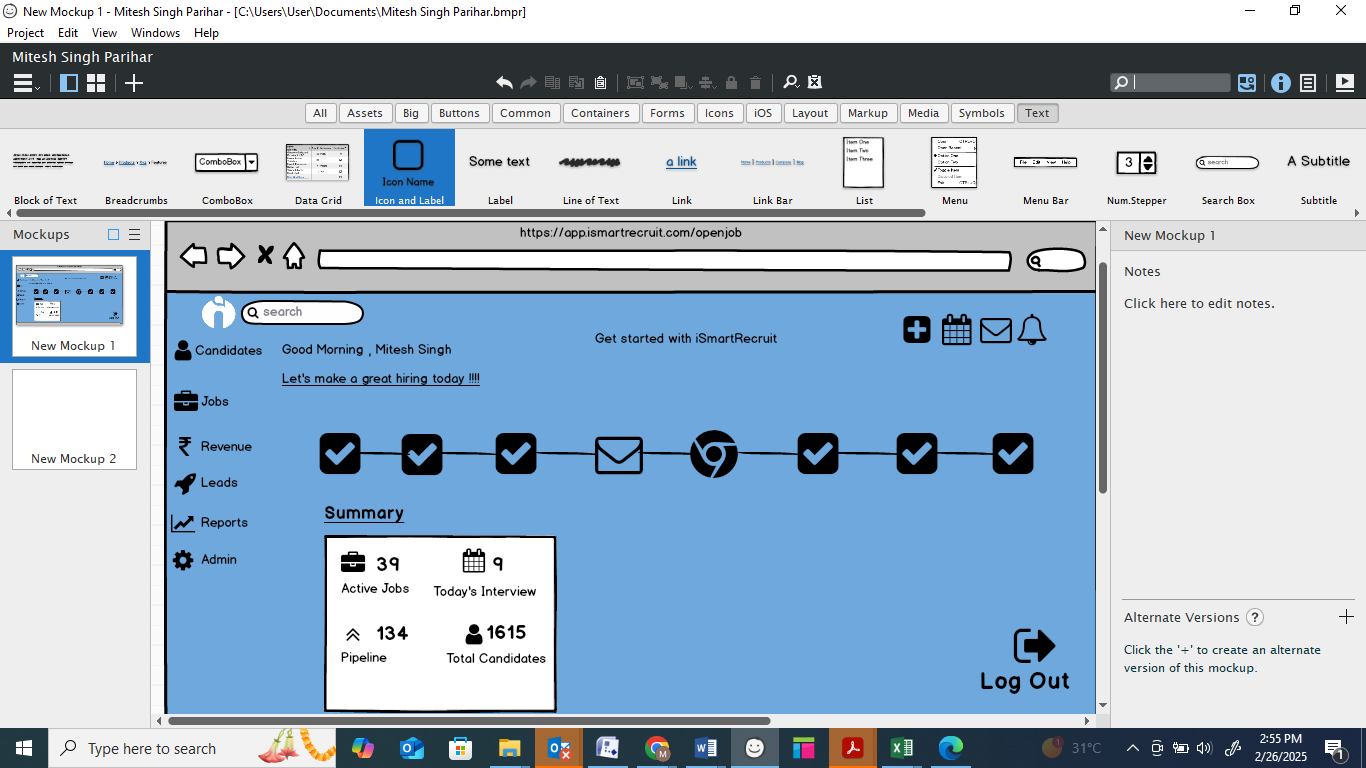












**Document 8- Tools-Visio and Axure**

**Write a paragraph on your experience using Visio and Axure for the project.**

In my experience working on the **Applicant Tracking System (ATS) project**, I extensively used **Microsoft Visio** and **Axure RP** for creating process flows, wireframes, and prototypes to enhance requirement clarity and stakeholder communication. **Visio** was instrumental in designing workflow diagrams, such as the **candidate application process, job posting flow, and interview scheduling lifecycle**. I created **swimlane diagrams** to define responsibilities across recruiters, candidates, and hiring managers, ensuring a clear visualization of role-based interactions. Additionally, **Axure RP** played a crucial role in developing **interactive prototypes** to simulate user journeys, including **job application submission, recruiter dashboard navigation, and interview scheduling interfaces**. By using **Axure’s dynamic panels and conditional logic**, I provided stakeholders with a realistic, clickable prototype that helped gather early feedback and refine UI/UX elements before actual development. This approach significantly

And also , During the **Applicant Tracking System (ATS) project**, I leveraged **Microsoft Visio** and **Axure RP** to streamline requirement visualization and enhance stakeholder collaboration. **Visio** was crucial for mapping out the **end-to-end recruitment workflow**, including **job posting, candidate application, screening, interview scheduling, and hiring decisions**. I designed **process flow diagrams, swimlane diagrams, and decision trees** to clearly define role-based responsibilities for **recruiters, hiring managers, and candidates**, ensuring a well-structured representation of business processes. Additionally, I used **Axure RP** to develop **interactive, high-fidelity wireframes** for key ATS features such as the **recruiter dashboard, job listing interface, and candidate profile management**. By incorporating **dynamic panels, conditional logic, and prototype interactions**, I enabled stakeholders to experience a **realistic simulation of the ATS user journey**, allowing for early feedback and design refinements. This approach significantly improved **requirement clarity, reduced ambiguities, and expedited approvals**, ultimately leading to a more user-friendly and efficient ATS implementation.

**Document 9- BA experience**

* **Question: Can you describe your experience as a Business Analyst in the Requirement Gathering phase?**

**Answer:**

**1. Understanding Business Needs & Defining Scope**

* My role as a **Business Analyst** starts with understanding the **business goals, objectives, and problems** that the project aims to solve.
* I worked closely with **business stakeholders, product owners, and end-users** to define the **project scope** and ensure alignment with business strategies.
* To avoid **scope creep**, I documented **in-scope and out-of-scope** items early in the project.

**2. Stakeholder Identification & Management**

* Identifying the **right stakeholders** is critical for gathering accurate and complete requirements.
* In a previous project, I faced challenges with **stakeholder unavailability**, so I proactively identified **alternate points of contact** to ensure smooth progress.
* I also managed **conflicting stakeholder expectations** by facilitating **requirement review meetings** to align all parties.

**3. Requirement Elicitation Using Various Techniques**

* I applied different **elicitation techniques** depending on the project and stakeholders, such as:
  + **Workshops & Brainstorming** – Engaged multiple stakeholders to define and refine requirements collaboratively.
  + **Surveys & Questionnaires** – Used structured surveys when dealing with a large group of users.
  + **Observation & Shadowing** – Observed current business processes to identify pain points and inefficiencies.
  + **Prototyping** – Worked with **UI/UX teams** to create mockups that helped stakeholders visualize the system.

**4. Requirement Prioritization (MoSCoW Technique)**

* Once requirements were gathered, I used the **MoSCoW method** to classify them as:
  + **Must-Have** – Critical for system functionality (e.g., resume upload in an ATS).
  + **Should-Have** – Important but can be deferred if needed (e.g., integration with LinkedIn).
  + **Could-Have** – Nice-to-have features (e.g., AI-based resume screening).
  + **Won’t-Have** – Out of scope for this phase but may be considered in future releases.
* This helped ensure that **business-critical features were delivered on time** while managing scope effectively.

**5. Requirement Validation Using FURPS**

* After gathering requirements, I validated them using the **FURPS framework**:
  + **Functionality:** Does the requirement align with business needs?
  + **Usability:** Is it user-friendly and intuitive?
  + **Reliability:** Can the system handle failures and exceptions?
  + **Performance:** Does it meet response time and scalability expectations?
  + **Supportability:** Is it maintainable and extendable in the future?
* Example: If a requirement stated, *“The ATS should support resume uploads,”* I validated:
  + **File format constraints (Functionality)**
  + **User-friendly upload process (Usability)**
  + **Handling large file sizes (Performance)**

**6. Handling Duplicate & Conflicting Requirements**

* Often, different stakeholders provide **similar or contradictory requirements**.
* I ensured that **redundant requirements were eliminated** by:
  + Maintaining a **Requirement Traceability Matrix (RTM)**.
  + Consolidating similar requirements into a **single, well-defined feature**.
  + Organizing **review meetings** to resolve conflicts among stakeholders.

**7. Business Process Analysis & Workflow Mapping**

* As part of requirement gathering, I conducted a **gap analysis** to compare **existing vs. proposed processes**.
* I used **process flow diagrams (BPMN), swimlane diagrams, and data flow diagrams (DFD)** to document business processes.
* This helped stakeholders **visualize process changes** and understand the impact of new requirements.

**8. Documentation of Requirements & Approval Process**

* I documented requirements in structured formats, including:
  + **Business Requirement Document (BRD)** – Captured high-level business needs.
  + **Functional Specification Document (FSD)** – Defined system functionalities in detail.
  + **Use Case Documents** – Explained system interactions.
  + **User Stories & Acceptance Criteria** – For Agile projects.
* Before finalizing, I conducted **requirement walkthroughs with stakeholders** to get feedback and approval.

**9. Managing Change Requests**

* Requirement changes are inevitable, so I followed a **structured change management process**:
  + Assessed the **impact of the change** on the scope, timeline, and budget.
  + Discussed the change with stakeholders and obtained **formal approvals** before implementation.
  + Updated the **requirement documentation** and communicated changes to the development team.

**10. Collaboration with Cross-Functional Teams**

* Throughout the requirement gathering phase, I acted as a **bridge between business and technical teams**.
* I collaborated with:
  + **Developers** – To ensure technical feasibility.
  + **QA Team** – To define test cases based on requirements.
  + **Project Managers** – To ensure alignment with project timelines.
* **Can you describe your experience as a Business Analyst in the Requirement Anaysis phase?**

**Requirement Analysis Phase – My Role as a Business Analyst**

**Visualizing Requirements Using UML Diagrams**

* To ensure a **clear understanding of system behavior**, I created **UML diagrams** such as:
  + **Use Case Diagrams** to represent system interactions.
  + **Class Diagrams** to define system entities and relationships.
  + **Sequence Diagrams** to illustrate system workflows.

**Defining Process Flow with Activity Diagrams**

* I used **Activity Diagrams** to visually represent the **step-by-step process flow** of different system functionalities.
* These diagrams helped in clarifying the **logical flow of operations**, ensuring alignment among stakeholders.

**Stakeholder Collaboration & Managing Feedback**

* I presented the diagrams to the development, QA, and business teams to validate accuracy.
* In some cases, **team members provided feedback or suggested modifications**, which I carefully analyzed before updating the diagrams.
* As a BA, I ensured that all modifications aligned with business objectives and did not impact other requirements.

**Documenting Requirements**

* I prepared key requirement documents:
  + **Business Requirement Specification (BRS)** – Capturing high-level business needs and objectives.
  + **Software Requirement Specification (SRS)** – Detailing functional and non-functional requirements, system workflows, and dependencies.
* These documents were reviewed and signed off before moving to the design phase.

**Additional Roles of a Business Analyst in the Requirement Analysis Phase**

**Requirement Feasibility Analysis**

* Assess whether the gathered requirements are **technically and functionally feasible** within project constraints such as **budget, time, and available technology**.
* Work closely with developers and architects to ensure that proposed solutions are **achievable**.

**Identifying Dependencies and Risks**

* Analyze **interdependencies** between requirements and ensure they do not conflict with other system functionalities.
* Identify potential **risks** (such as scope creep, conflicting stakeholder expectations, or technical limitations) and propose mitigation strategies.

**Gap Analysis**

* Compare **current system capabilities vs. desired system capabilities** to identify gaps.
* Suggest enhancements or modifications to bridge the gap between **business needs and technical capabilities**.

**Creating Data Models**

* If the project involves data-heavy functionalities, create **ER (Entity-Relationship) diagrams** and **data flow diagrams** to define how data moves across the system.
* Work with the database team to validate **data storage, retrieval, and integration** requirements.

**Impact Analysis for Change Requests**

* If any changes arise during the analysis phase, conduct an **impact analysis** to assess their effect on other requirements, timelines, and costs.
* Ensure that changes are documented and approved through a structured **Change Control Process**. **Stakeholder Alignment and Sign-off**
* Conduct review meetings with stakeholders to confirm that analyzed requirements align with business needs.
* Get formal **sign-off** on the **finalized requirements, UML diagrams, and requirement documents (BRS, SRS)** before transitioning to the design phase.
* **Can you describe your experience as a Business Analyst in the Design phase?**

**Role of a Business Analyst in the Design Phase**

**Translating Requirements into Design Specifications**

* After finalizing the **Software Requirement Specification (SRS)**, I worked closely with the **UI/UX designers, architects, and development team** to ensure the system design aligned with business requirements.
* I ensured that the **wireframes, prototypes, and system architecture** correctly reflected the functional and non-functional requirements.

**Deriving Test Cases from Use Case Diagrams**

* Based on **Use Case Diagrams**, I helped in defining **test scenarios and test cases** to ensure that all business flows were covered.
* These test cases acted as a bridge between business requirements and the testing phase.

**Communicating Design & Solution Documents with Clients**

* I coordinated with the client to **review and validate design documents** such as **solution architecture, UI wireframes, and system workflows**.
* Gathered **feedback** and ensured necessary modifications were incorporated before finalization.

**Writing Positive & Negative Test Cases**

* I contributed to writing **both positive and negative test cases** to ensure robust system validation.
  + **Positive Test Cases:** Verify the system behaves as expected under normal conditions.
  + **Negative Test Cases:** Check how the system handles unexpected inputs, invalid data, and edge cases.
* This approach helped in **identifying potential defects early**, reducing rework in later stages.

**Ensuring No Test Case is Missed**

* I ensured that **all functional, non-functional, and boundary test cases** were covered to prevent missing any critical scenarios.
* Missing a test case could lead to **unexpected issues during development or production**, so I closely reviewed test scenarios with the QA team.

**Preparing Test Data for Testing**

* I worked with stakeholders to define **realistic test data** based on business scenarios.
* This helped testers execute their cases with **accurate inputs** and validate system behavior effectively.

**Updating the Requirement Traceability Matrix (RTM)**

* Maintained and updated the **Requirement Traceability Matrix (RTM)** to ensure that every requirement was **mapped to a design component and test case**.
* This helped in tracking coverage and ensuring **no requirement was left untested**.
* **Can you describe your experience as a Business Analyst in the Design phase?**

**Organizing Joint Application Development (JAD) Sessions**

* **Purpose:** JAD sessions help align stakeholders, developers, and testers by ensuring a common understanding of the system requirements.
* **My Role:**
  + **Scheduled and facilitated JAD sessions** involving business users, developers, and testers.
  + Created an **agenda** to focus on critical requirements, system workflows, and potential technical challenges.
  + Ensured that each team member provided input and that all concerns were addressed before development proceeded.

**Impact:** These sessions **reduced misunderstandings**, saved time, and ensured all stakeholders were aligned before coding started.

**Clarifying Queries of the Tech Team During Coding**

* **Challenge:** Developers often have doubts while implementing business logic, system integration, or data handling.
* **My Role:**
  + Acted as the **primary point of contact** for the technical team to clarify requirements.
  + Ensured that **business rules, process flows, and functional logic** were implemented correctly.
  + Provided additional **documentation, examples, and real-world scenarios** to help developers understand complex logic.

**Impact:** This prevented **misinterpretation of requirements**, reducing rework and ensuring that development stayed on track.

**Handling Team Conflicts & Ensuring Cooperation**

* **Challenge:** Some developers or stakeholders may disagree with the approach or be uncooperative during discussions.
* **My Role:**
  + Identified individuals who were **not aligned** with the project objectives.
  + Conducted **one-on-one discussions** to understand their concerns and resolve them.
  + Explained the **business impact** of their resistance, showing how delays or misalignment could affect the overall project.
  + Created a **collaborative and open environment** where concerns were addressed proactively.

**Impact:** This **ensured a productive and positive team environment**, leading to smoother development.

**Referring Diagrams to Code the Unit**

* **Challenge:** Developers may require **visual references** to understand how system components interact.
* **My Role:**
  + Provided **Use Case Diagrams** to show system interactions.
  + Shared **Activity Diagrams and Process Flows** to clarify the logical flow of the system.
  + Ensured that developers understood **ER Diagrams and Data Flow Diagrams (DFD)** for database implementation.

**Impact:** Improved **development accuracy**, ensuring that implementation matched business requirements.

**Conducting Regular Meetings with the Technical Team & Client**

* **Challenge:** Tracking progress and ensuring all teams stay informed can be difficult, especially when team members are unavailable for meetings.
* **My Role:**
  + **Scheduled weekly meetings** to track progress, discuss roadblocks, and align expectations.
  + **Challenge:** Some team members were unavailable due to different priorities.
    - **Solution:**
      * Recorded the meeting and shared the **recording and summary**.
      * Conducted **one-on-one discussions** to update missing team members.
  + Facilitated **review sessions** with the client to ensure development progress matched expectations.

**Impact:** Ensured that **all team members stayed informed** and that any roadblocks were addressed promptly.

**Updating the Requirement Traceability Matrix (RTM)**

* **Challenge:** Ensuring that all business requirements were being implemented correctly.
* **My Role:**
  + Updated the **RTM** to **map each requirement to a development unit**.
  + Verified that all functionalities were covered **without any missing requirements**.

**Impact:** Ensured **100% requirement coverage**, preventing functionality gaps.

* **Can you describe your experience as a Business Analyst in the Testing phase?**

The **Testing Phase** is crucial because it validates whether the system meets business requirements and functions correctly. As a **Business Analyst**, I played a key role in ensuring smooth testing by **preparing test cases, verifying requirement coverage, resolving defects, and preparing stakeholders for UAT**.

**Preparing Test Cases from Use Cases**

* **Challenge:** Ensuring that all business scenarios, edge cases, and workflows were tested.
* **My Role:**
  + Developed **test cases based on Use Cases, Business Rules, and Process Flows**.
  + Created both **positive and negative test cases** to validate different scenarios.
  + Focused on **boundary value analysis and edge cases** to test system limitations.
  + Ensured that test cases covered both **functional and non-functional requirements** (e.g., performance, security).
  + Worked closely with **QA testers** to ensure they understood the business logic behind test cases.

**Impact:** Ensured that all functionalities were tested before moving to the next phase, reducing defects in production.

**Test Case: Login Functionality in Application Tracking System**

**Test Case ID: TC\_LOGIN\_001**

**Module: Login**

**Description: Verify that the login functionality works as expected for valid and invalid inputs.**

**Preconditions:**

1. The user must have a valid account registered in the ATS.
2. The ATS login page must be accessible.
3. Browser/device should have an active internet connection.

**Test** Data:

* Valid Username: testuser@example.com
* Valid Password: Password@123
* Invalid Username: invaliduser@example.com
* Invalid Password: WrongPassword

**Test Steps:**

**Scenario 1: Successful Login with Valid Credentials**

1. Navigate to the ATS login page.
2. Enter the valid username (testuser@example.com) in the username field.
3. Enter the valid password (Password@123) in the password field.
4. Click the "Login" button.
5. Verify that the user is redirected to the dashboard or home page.
6. Verify that the user's session is active and the username is displayed correctly.

**Scenario 2: Login with Invalid Username**

1. Navigate to the ATS login page.
2. Enter an invalid username (invaliduser@example.com) in the username field.
3. Enter a valid password (Password@123) in the password field.
4. Click the "Login" button.
5. Verify that an appropriate error message is displayed (e.g., "Invalid username or password").
6. Verify that the user is not redirected to the dashboard.

**Scenario 3: Login with Invalid Password**

1. Navigate to the ATS login page.
2. Enter a valid username (testuser@example.com) in the username field.
3. Enter an invalid password (WrongPassword) in the password field.
4. Click the "Login" button.
5. Verify that an appropriate error message is displayed (e.g., "Invalid username or password").
6. Verify that the user is not redirected to the dashboard.

**Scenario 4: Login with Empty Username**

1. Navigate to the ATS login page.
2. Leave the username field empty.
3. Enter a valid password (Password@123) in the password field.
4. Click the "Login" button.
5. Verify that an appropriate error message is displayed (e.g., "Username is required").
6. Verify that the user is not redirected to the dashboard.

**Scenario 5: Login with Empty Password**

1. Navigate to the ATS login page.
2. Enter a valid username (testuser@example.com) in the username field.
3. Leave the password field empty.
4. Click the "Login" button.
5. Verify that an appropriate error message is displayed (e.g., "Password is required").
6. Verify that the user is not redirected to the dashboard.

**Scenario 6: Login with Empty Username and Password**

1. Navigate to the ATS login page.
2. Leave both the username and password fields empty.
3. Click the "Login" button.
4. Verify that an appropriate error message is displayed (e.g., "Username and password are required").
5. Verify that the user is not redirected to the dashboard.

**Scenario 7: Password Masking**

1. Navigate to the ATS login page.
2. Enter a valid password (Password@123) in the password field.
3. Verify that the password is masked (e.g., displayed as ••••••••).

**Scenario 8: Forgot Password Functionality**

1. Navigate to the ATS login page.
2. Click the "Forgot Password" link.
3. Verify that the user is redirected to the password recovery page.
4. Enter a valid email address (testuser@example.com).
5. Click the "Submit" button.
6. Verify that a confirmation message is displayed (e.g., "Password reset instructions have been sent to your email").

**Scenario 9: Session Timeout**

1. Log in with valid credentials.
2. Wait for the session timeout period (e.g., 15 minutes) without any activity.
3. Verify that the user is automatically logged out and redirected to the login page.
4. Verify that an appropriate message is displayed (e.g., "Your session has expired. Please log in again").

**Test Cases for Submit Resume (Recruiter Perspective)**

**Test Case ID: TC\_RESUME\_REC\_001**

1. **Description**: Verify that a recruiter can successfully upload a resume for a candidate in a supported format (e.g., PDF, DOCX).  
   **Steps**:
2. Log in as a recruiter.
3. Navigate to the "Add Candidate" or "Upload Resume" section.
4. Fill in the candidate's details (e.g., name, email, job applied for).
5. Upload a resume in a supported format (e.g., PDF).
6. Save the candidate's profile.  
   **Expected Result**: The resume is successfully uploaded, and the candidate's profile is created. A confirmation message is displayed.

**Test Case ID: TC\_RESUME\_REC\_002**

1. **Description**: Verify that a recruiter cannot upload a resume in an unsupported format (e.g., JPG, TXT).  
   **Steps**:
2. Log in as a recruiter.
3. Navigate to the "Add Candidate" or "Upload Resume" section.
4. Fill in the candidate's details (e.g., name, email, job applied for).
5. Upload a resume in an unsupported format (e.g., JPG).
6. Save the candidate's profile.  
   **Expected Result**: An error message is displayed (e.g., "Unsupported file format. Please upload a PDF or DOCX file."), and the candidate's profile is not created.

**Test Case ID: TC\_RESUME\_REC\_003**

**Description**: Verify that a recruiter cannot create a candidate profile without uploading a resume.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Add Candidate" or "Upload Resume" section.
3. Fill in the candidate's details (e.g., name, email, job applied for).
4. Leave the resume upload field empty.
5. Save the candidate's profile.  
   **Expected Result**: An error message is displayed (e.g., "Resume is required."), and the candidate's profile is not created.

**Test Case ID: TC\_RESUME\_REC\_004**

**Description**: Verify that the system handles large resume files gracefully.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Add Candidate" or "Upload Resume" section.
3. Fill in the candidate's details (e.g., name, email, job applied for).
4. Upload a resume file larger than the allowed size limit (e.g., 5 MB).
5. Save the candidate's profile.  
   **Expected Result**: An error message is displayed (e.g., "File size exceeds the limit. Maximum allowed size is 5 MB."), and the candidate's profile is not created.

**Test Case ID: TC\_RESUME\_REC\_005**

1. **Description**: Verify that a recruiter can upload multiple resumes for different candidates.  
   **Steps**:
2. Log in as a recruiter.
3. Navigate to the "Add Candidate" or "Upload Resume" section.
4. Fill in the details for Candidate 1 and upload their resume.
5. Save the profile.
6. Repeat the process for Candidate 2.  
   **Expected Result**: Resumes for both candidates are successfully uploaded, and their profiles are created.

**Test Case ID: TC\_RESUME\_REC\_006**

**Description**: Verify that the system parses candidate information (e.g., name, email, skills) from the uploaded resume.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Add Candidate" or "Upload Resume" section.
3. Upload a resume containing candidate information (e.g., name, email, skills).
4. Check if the system auto-populates the candidate's details.  
   **Expected Result**: The system successfully parses the resume and auto-populates the candidate's details.

**Test Case ID: TC\_RESUME\_REC\_007**

1. **Description**: Verify that the system handles corrupted or unreadable resume files.  
   **Steps**:
2. Log in as a recruiter.
3. Navigate to the "Add Candidate" or "Upload Resume" section.
4. Upload a corrupted or unreadable resume file.
5. Save the candidate's profile.  
   **Expected Result**: An error message is displayed (e.g., "The file is corrupted or unreadable. Please upload a valid resume."), and the candidate's profile is not created.

**Test Cases for Review Application (Recruiter Perspective)**

**Test Case ID: TC\_REVIEW\_APP\_001**

1. **Description**: Verify that a recruiter can view all submitted applications for a specific job posting.  
   **Steps**:
2. Log in as a recruiter.
3. Navigate to the "Job Postings" or "Applications" section.
4. Select a specific job posting.
5. Click "View Applications."  
   **Expected Result**: All submitted applications for the selected job posting are displayed.

**Test Case ID: TC\_REVIEW\_APP\_002**

**Description**: Verify that a recruiter can filter applications by status (e.g., New, In Review, Rejected, Hired).  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Applications" section.
3. Use the filter dropdown to select a status (e.g., "New").
4. Click "Apply Filter."  
   **Expected Result**: Only applications with the selected status are displayed.

**Test Case ID: TC\_REVIEW\_APP\_003**

**Description**: Verify that a recruiter can sort applications by submission date, name, or relevance.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Applications" section.
3. Use the sort dropdown to select a sorting option (e.g., "Submission Date - Newest First").
4. Verify the order of applications.  
   **Expected Result**: Applications are displayed in the selected order.

**Test Case ID: TC\_REVIEW\_APP\_004**

**Description**: Verify that a recruiter can view detailed information for a specific application.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Applications" section.
3. Click on a specific application.  
   **Expected Result**: The candidate's detailed information (e.g., resume, cover letter, contact details) is displayed.

**Test Case ID: TC\_REVIEW\_APP\_005**

1. **Description**: Verify that a recruiter can update the status of an application (e.g., New → In Review → Rejected/Hired).  
   **Steps**:
2. Log in as a recruiter.
3. Navigate to the "Applications" section.
4. Select a specific application.
5. Change the status (e.g., from "New" to "In Review").
6. Save the changes.  
   **Expected Result**: The application status is updated, and the change is reflected in the system.

**Test Cases for Shortlist Candidates (Recruiter Perspective)**

**Test Case ID: TC\_SHORTLIST\_001**

**Description**: Verify that a recruiter can shortlist a candidate from the list of applications.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Applications" section.
3. Select a specific job posting.
4. Click "Shortlist" for a candidate.  
   **Expected Result**: The candidate is marked as "Shortlisted," and the status is updated.

**Test Case ID: TC\_SHORTLIST\_002**

1. **Description**: Verify that a recruiter can view all shortlisted candidates for a specific job posting.  
   **Steps**:
2. Log in as a recruiter.
3. Navigate to the "Applications" section.
4. Select a specific job posting.
5. Click "View Shortlisted Candidates."  
   **Expected Result**: A list of all shortlisted candidates for the job posting is displayed.

**Test Case ID: TC\_SHORTLIST\_003**

**Description**: Verify that a recruiter can filter shortlisted candidates by status (e.g., New, Interview Scheduled, Hired).  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Shortlisted Candidates" section.
3. Use the filter dropdown to select a status (e.g., "Interview Scheduled").
4. Click "Apply Filter."  
   **Expected Result**: Only shortlisted candidates with the selected status are displayed.

**Test Case ID: TC\_SHORTLIST\_004**

**Description**: Verify that a recruiter can remove a candidate from the shortlist.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Shortlisted Candidates" section.
3. Select a shortlisted candidate.
4. Click "Remove from Shortlist."  
   **Expected Result**: The candidate is removed from the shortlist, and the status is updated.

**Test Case ID: TC\_SHORTLIST\_005**

**Description**: Verify that a recruiter can add notes or comments to a shortlisted candidate's profile.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Shortlisted Candidates" section.
3. Select a shortlisted candidate.
4. Click "Add Notes" and enter a comment (e.g., "Strong technical skills, good cultural fit").
5. Save the notes.  
   **Expected Result**: The notes are saved and displayed in the candidate's profile.

**Test Case ID: TC\_SHORTLIST\_006**

1. **Description**: Verify that a recruiter can schedule an interview for a shortlisted candidate.  
   **Steps**:
2. Log in as a recruiter.
3. Navigate to the "Shortlisted Candidates" section.
4. Select a shortlisted candidate.
5. Click "Schedule Interview."
6. Fill in the interview details (e.g., date, time, interviewer).
7. Save the schedule.  
   **Expected Result**: The interview is scheduled, and the candidate receives a notification.

**Test Cases for Schedule Interviews (Recruiter Perspective)**

**Test Case ID: TC\_SCHEDULE\_INTERVIEW\_001**

**Description**: Verify that a recruiter can schedule an interview for a candidate.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Applications" or "Shortlisted Candidates" section.
3. Select a candidate.
4. Click "Schedule Interview."
5. Fill in the interview details (e.g., date, time, location, interviewer).
6. Save the schedule.  
   **Expected Result**: The interview is successfully scheduled, and the candidate receives a notification.

**Test Case ID: TC\_SCHEDULE\_INTERVIEW\_002**

**Description**: Verify that a recruiter cannot schedule an interview without filling in mandatory fields (e.g., date, time, interviewer).  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Applications" or "Shortlisted Candidates" section.
3. Select a candidate.
4. Click "Schedule Interview."
5. Leave mandatory fields (e.g., date) blank.
6. Save the schedule.  
   **Expected Result**: An error message is displayed (e.g., "Date and time are required."), and the interview is not scheduled.

**Test Case ID: TC\_SCHEDULE\_INTERVIEW\_003**

**Description**: Verify that a recruiter can reschedule an interview.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Scheduled Interviews" section.
3. Select a scheduled interview.
4. Click "Reschedule."
5. Update the interview details (e.g., change the date or time).
6. Save the changes.  
   **Expected Result**: The interview is rescheduled, and the candidate receives a notification about the updated details.

**Test Case ID: TC\_SCHEDULE\_INTERVIEW\_004**

**Description**: Verify that a recruiter can cancel a scheduled interview.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Scheduled Interviews" section.
3. Select a scheduled interview.
4. Click "Cancel Interview."
5. Confirm the cancellation.  
   **Expected Result**: The interview is canceled, and the candidate receives a notification.

**Test Case ID: TC\_SCHEDULE\_INTERVIEW\_005**

**Description**: Verify that a recruiter can add notes or comments to a scheduled interview.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Scheduled Interviews" section.
3. Select a scheduled interview.
4. Click "Add Notes" and enter a comment (e.g., "Focus on technical skills during the interview.").
5. Save the notes.  
   **Expected Result**: The notes are saved and displayed in the interview details.

**Test Case ID: TC\_SCHEDULE\_INTERVIEW\_006**

**Description**: Verify that a recruiter can view all scheduled interviews for a specific job posting.  
**Steps**:

1. Log in as a recruiter.
2. Navigate to the "Job Postings" section.
3. Select a specific job posting.
4. Click "View Scheduled Interviews."  
   **Expected Result**: A list of all scheduled interviews for the job posting is displayed.

### Performing High-Level Testing (Business Validation****)****

* **Challenge:** Identifying critical business defects before formal testing by the QA team.
* **My Role:**
  + Conducted **high-level business validation testing** (a preliminary check before detailed testing).
  + Performed **sanity and smoke testing** to ensure major functionalities worked before deeper testing.
  + Focused on testing **critical business processes** rather than just system functionalities.
  + Verified that workflows matched **business expectations and real-world operations**.

**Impact:** Identified critical issues early, ensuring that the system behaved as expected before deeper testing.

### Assisting in Test Data Preparation & Requesting Data from Clients

* **Challenge:** Ensuring that test cases were executed with **realistic business data**.
* **My Role:**
  + Collaborated with the client to collect **real-world test data** for meaningful validation.
  + Worked with the **QA team** to identify missing or incomplete data.
  + Ensured that **confidential data was anonymized** when required for compliance.
  + Assisted in **data migration validation** (if applicable) to check the accuracy of imported data.

**Impact:** Helped in **realistic system validation** using business-relevant data, improving test accuracy.

### Updating the Requirement Traceability Matrix (RTM)

* **Challenge:** Ensuring that every business requirement was **tested and verified**.
* **My Role:**
  + Mapped each test case to a specific **business requirement** in the RTM.
  + Verified that all test scenarios were **traceable to requirements and design artifacts**.
  + Ensured that no requirement was left untested before moving to UAT.

**Impact:** Ensured **100% requirement coverage**, avoiding gaps in system functionality.

### Defect Triage & Handling Testing Defects

* **Challenge:** Managing and resolving testing defects efficiently.
* **My Role:**
  + Participated in **defect triage meetings** with testers and developers to prioritize issues.
  + Helped categorize defects as **critical, major, or minor** based on business impact.
  + Assisted in **clarifying defect root causes** (requirement issue vs. development issue).
  + Worked with stakeholders to **decide on fixes and retesting priorities**.

**Impact:** Helped in **faster defect resolution**, reducing back-and-forth discussions.

### Taking Sign-Off from the Client Before UAT

* **Challenge:** Ensuring that the client was satisfied with the system before User Acceptance Testing (UAT).
* **My Role:**
  + Conducted a **review session with stakeholders** to present test results.
  + Addressed any last-minute **concerns, issues, or change requests**.
  + Ensured that all critical business requirements were **validated** before proceeding to UAT.
  + Obtained a **formal sign-off** from stakeholders confirming that the system was ready for UAT.

**Impact:** Ensured that the client was **confident in the system’s functionality** before moving to UAT.

### Preparing the Client for User Acceptance Testing (UAT)

* **Challenge:** Ensuring that the client could independently validate the system in real-world conditions.
* **My Role:**
  + **Created UAT scenarios and test scripts** that aligned with business workflows.
  + Provided **UAT training** to business users on how to execute test cases.
  + Assisted in setting up **UAT test data** and environment for real-world testing.
  + Facilitated **feedback collection from UAT users** and reported issues to the development team.
  + Ensured that any defects found in UAT were **tracked, resolved, and retested** before production release.

**Impact:** Ensured a **smooth UAT process**, enabling business users to validate the system effectively.

* **Can you describe your experience as a Business Analyst in the Deployment phase?**

The **Deployment Phase** is a critical step where the system goes live and is handed over to the end-users. As a **Business Analyst**, my role was to **ensure a smooth transition** by coordinating with stakeholders, validating system readiness, organizing training sessions, and ensuring proper **documentation.**

### Forwarding RTM to the Client & Project Closure Documentation

* **Challenge:** Ensuring that the system was **fully tested and validated** before deployment.
* **My Role:**
  + Updated and **forwarded the Requirement Traceability Matrix (RTM)** to the client.
  + Ensured the RTM was **attached to the Project Closure document** to confirm all requirements were met.
  + Verified that all **business-critical functionalities** were accounted for.
  + Assisted in preparing **compliance reports** if required.

**Impact:** Provided a clear audit trail proving that **all business needs were implemented and tested**.

### 2Coordinating the Completion & Distribution of End-User Manuals

* **Challenge:** Ensuring end-users had the necessary documentation for smooth system adoption.
* **My Role:**
  + Coordinated with **technical writers and SMEs** to finalize the **End-User Manuals**.
  + Reviewed the manuals to ensure they were **clear, concise, and business-oriented**.
  + Distributed manuals to **stakeholders and training teams** before go-live.
  + Created **FAQs or quick reference guides** to address common user concerns.

**Impact:** Helped end-users **quickly adapt to the system**, reducing post-deployment support issues.

### Planning & Organizing Training Sessions

* **Challenge:** Ensuring all users were trained before system rollout.
* **My Role:**
  + Identified **key user groups** who needed training (business users, support teams, etc.).
  + Designed a **training plan and schedule** that accommodated all stakeholders.
  + Organized **live training sessions, webinars, and hands-on workshops**.
  + Ensured training materials, including **videos, presentations, and guides**, were prepared in advance.
  + Set up **training environments** where users could practice before the system went live.

**Impact:** Enabled **smooth system adoption**, minimizing confusion and resistance from users.

### Ensuring All Candidates Attend the Training Sessions

* **Challenge:** Managing stakeholder availability and ensuring full participation.
* **My Role:**
  + **Sent invitations** well in advance to ensure maximum attendance.
  + Tracked attendance and followed up with those who missed training.
  + **Recorded sessions** and shared them with absentees.
  + Arranged **one-on-one training** for key users who required extra support.

**Impact:** Ensured that **all stakeholders were prepared**, leading to minimal disruptions post-deployment.

### Coordinating the Go-Live Process

* **Challenge:** Ensuring a seamless transition from the old system to the new one.
* **My Role:**
  + Ensured all **pre-go-live checks** were completed.
  + Coordinated with IT and support teams for a **smooth system launch**.
  + Monitored the system post-go-live to address any **immediate issues**.
  + Conducted a **hypercare phase** where additional support was available for users.

**Impact:** Ensured a **successful go-live** with minimal disruptions to business operations.

### Gathering Feedback & Finalizing Post-Deployment Support

* **Challenge:** Identifying any post-deployment issues early.
* **My Role:**
  + Collected **feedback from end-users** regarding system performance and usability.
  + Documented any issues and worked with IT to **resolve post-go-live defects**.
  + Provided **support and knowledge transfer** to the IT helpdesk team.
  + Ensured all **pending change requests** were properly documented for future releases.

**Impact:** Improved **user satisfaction** and ensured a structured **post-deployment support process**.