**Waterfall Project2 – Part -2/2**

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

**Use Case diagram**

**Essential Usecase**

**Supporting Use Cases:**

* Enter Email Id
* Create Password
* Enter Phone Number
* Enter Email Id
* Enter Password
* Choose Net Banking
* Choose UPI
* Choose Card
* Choose Card Payment (Make Payment)

**Essential Use Cases:**

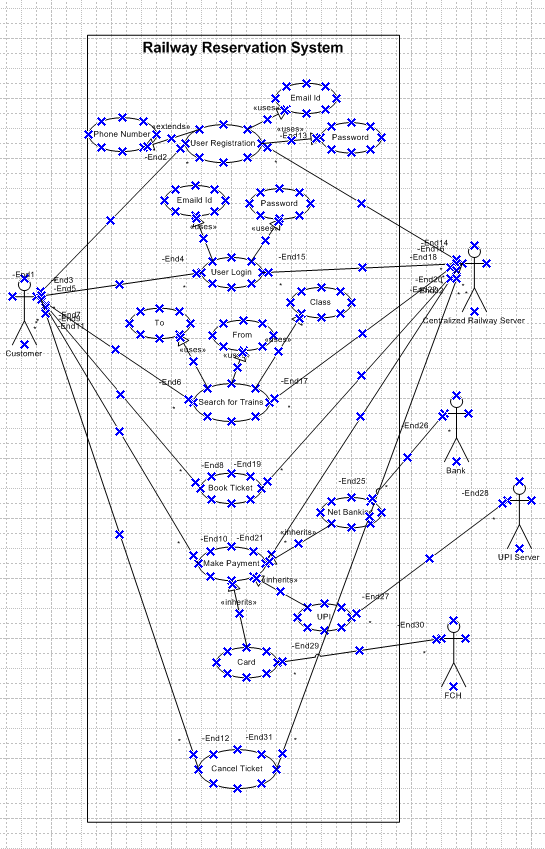
* User Registration
* Login to Application
* Search Trains
* Book Ticket
* Make Payment
* Cancel Ticket

**Primary Actors:**

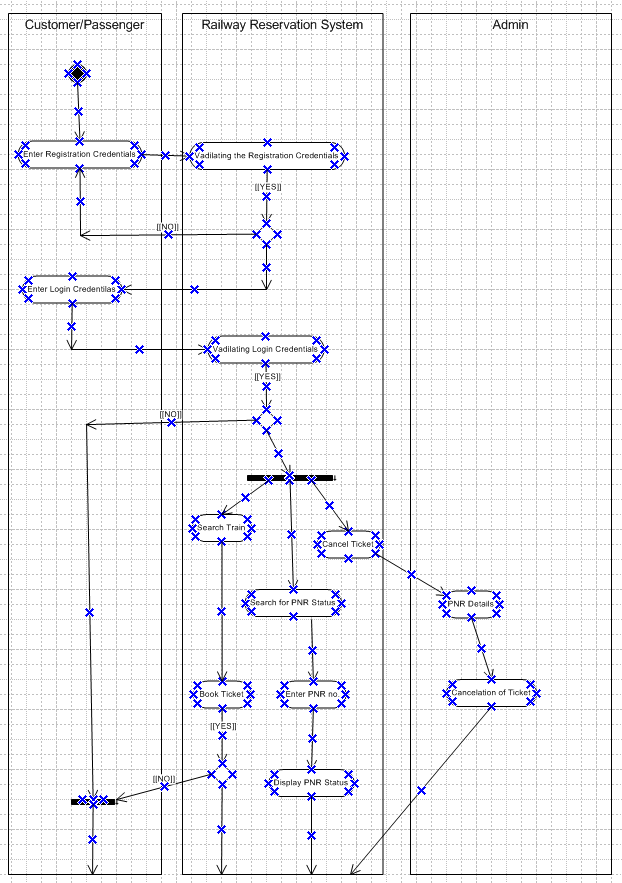
* Customer

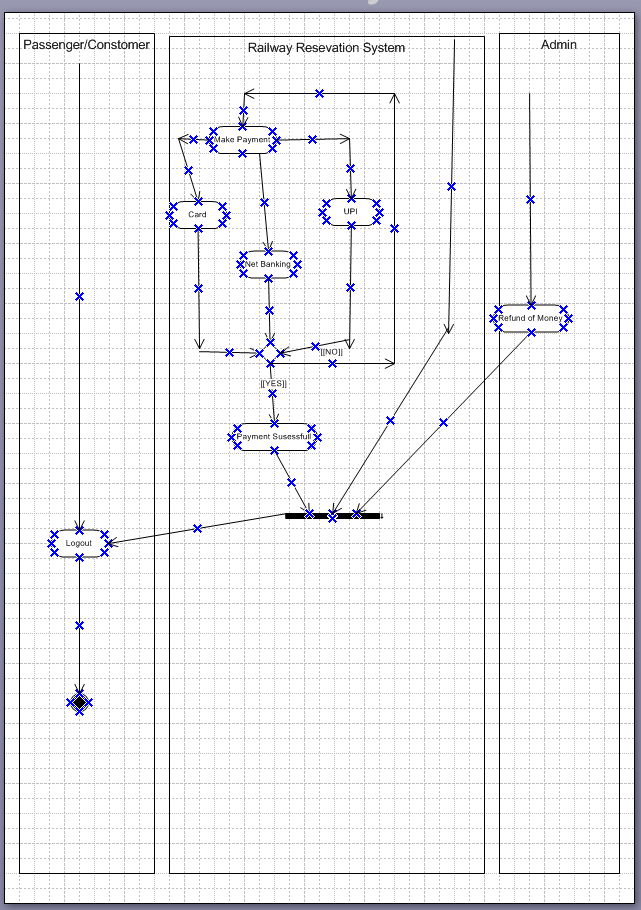
**Secondary Actors:**

* UPI Payment Gateway
* FCH
* Bank
* Centralized Railway System



**Activity Diagram**

****



**Use case specs**

A Use case Specification Document which provides a detailed description of a use case, outlining how users (actors) will interact with the system to achieve a specific goal.

**User Registration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | DC001 | | |
| **Use case Name** | User Registration | | |
| Created By | Mr. ABC | Last Update By | April 5th 2024 |
| Date Created | March 25th 2023 | Last Revision Date | April 1st 024 |
| Actors | Customer/Passenger | | |
| Description | This use case describes how users can register for an account in the new railway reservation system. | | |
| Pre- condition | User eligibility, required information, internet access, valid email, agreement to terms. | | |
| Post condition | Account creation, confirmation email, user logged in, profile setup, access to features. | | |
| Normal flow of event/ Basic flow / Happy path | 1. The user navigates to the registration page.  2. The user enters their details (name, email, phone number, password).  3. The system validates the information.  4. The system creates the user account.  5. The user is logged in and navigated to the dashboard.  6. The basic flow ends here. | | |
| Alternate Flow | If the information is invalid, the system prompts the user to re-enter their details. | | |
| Exceptions | If the system encounters an error, it displays an error message and prompts the user to try again later. | | |
| Frequency of Use | High | | |
| Assumptions | It is assumed that users have internet access.  Users meet eligibility criteria.  Users have necessary registration information.  Email and phone number are unique.  The registration page is accessible and functioning correctly.  The system's database is available and can store user information. | | |
| **Constraints** | The system must handle high registration volumes. | | |
| **Dependencies** | Database for storing user information. | | |
| **Inputs and Outputs** | Inputs: User details (name, email, phone number, password).  Outputs: Account creation confirmation. | | |
| **Business Rules** | Follow data privacy laws. | | |
| **Miscellaneous Information** | Email and phone number must be unique. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | DC002 | | |
| **Use case Name** | Login to Application | | |
| Created By | Mr. ABC | Last Update By | April 5th 2024 |
| Date Created | March 25th 2023 | Last Revision Date | April 1st 024 |
| Actors | Customer | | |
| Description | This use case describes how users can log into the application. | | |
| Pre- condition | The user must have a registered account in the application. | | |
| Post condition | The user is successfully logged into the system and navigated to their dashboard. | | |
| Normal flow of event/ Basic flow / Happy path | Step 1: The user navigates to the login page.  Step 2: The user enters their email and password.  Step 3: The user clicks on the "Login" button.  Step 4: The system validates the credentials.  Step 5: The system logs the user in and navigates to the dashboard.  Step 6: The basic flow ends here. | | |
| Alternate Flow | At step 4: The system displays an error message if user enters wrong password or email  At step 5: The user is prompted to re-enter their credentials. | | |
| Exceptions | If internet connectivity is lost while doing this Use case, system displays “check with your internet connectivity”  The user clicks on the "Forgot Password" link.  The system prompts the user to enter their registered email.  The system sends a password reset link to the user's email.  The user follows the link to reset their password. | | |
| Frequency of Use | High | | |
| Assumptions | It is assumed that users have registered accounts and remember their login credentials.  It is assumed the login page is accessible and functioning correctly.  It is assumed the email and password validation process is secure and reliable. | | |
| **Constraints** | The system must prevent unauthorized access. | | |
| **Dependencies** | Database for storing user credentials. | | |
| **Inputs and Outputs** | Inputs: User credentials (email, password).  Outputs: Access to user account. | | |
| **Business Rules** | Use secure authentication methods. | | |
| **Miscellaneous Information** | Three failed attempts lock the account for security. | | |

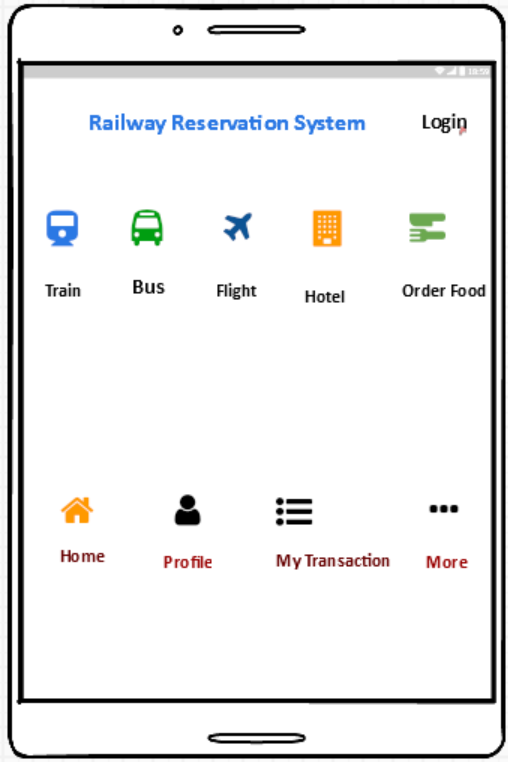
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | DC003 | | |
| **Use case Name** | Search Trains | | |
| Created By | Mr. ABC | Last Update By | April 5th 2024 |
| Date Created | March 25th 2023 | Last Revision Date | April 1st 024 |
| Actors | Customer, Train Information System | | |
| Description | This use case describes how users can search for trains by date, source, and destination in the new railway reservation system. | | |
| Pre- condition | The user must be logged in and have access to the search page. | | |
| Post condition | The user is presented with a list of matching trains. | | |
| Normal flow of event/ Basic flow / Happy path | 1. The user navigates to the search page.  2. The user enters the search criteria (date, source, destination).  3. The system retrieves matching trains from the Train Information System.  4. The results are displayed to the user.  5. The basic flow ends here. | | |
| Alternate Flow | If no matching trains are found, the system notifies the user and suggests alternative dates. | | |
| Exceptions | If the system encounters an error, it displays an error message and prompts the user to try again later. | | |
| Frequency of Use | High | | |
| Assumptions | It is assumed that users have internet access.  It is assumed the search page is accessible and functioning correctly.  It is assumed the train schedule data is accurate and up-to-date. | | |
| **Constraints** | The system must process searches quickly. | | |
| **Dependencies** | Train Information System for real-time updates. | | |
| **Inputs and Outputs** | Inputs: Search criteria (date, source, destination).  Outputs: List of matching trains. | | |
| **Business Rules** | Display only accurate and current train schedules. | | |
| **Miscellaneous Information** | Real-time updates on train availability must be accurate | | |

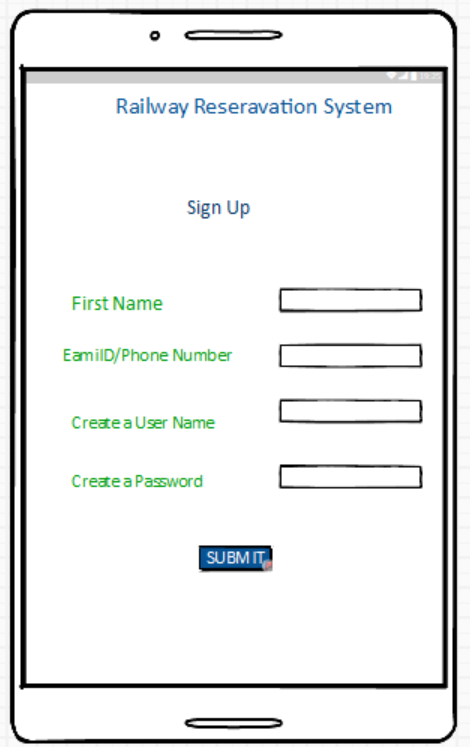
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | DC004 | | |
| **Use case Name** | Book Tickets | | |
| Created By | Mr. ABC | Last Update By | April 5th 2024 |
| Date Created | March 25th 2023 | Last Revision Date | April 1st 024 |
| Actors | Customer | | |
| Description | This use case describes how users can book train tickets in the new railway reservation system. | | |
| Pre- condition | User must be registered and logged in.  Selected train must have available seats. | | |
| Post condition | The ticket is successfully booked, and the user receives a confirmation. | | |
| Normal flow of event/ Basic flow / Happy path | 1. The user searches for and selects a train. 2. The user selects the seats, class, and berth preferences. 3. The system displays the booking summary. 4. The user confirms the booking details. 5. The system generates the booking confirmation. 6. The basic flow ends here. | | |
| Alternate Flow | If the selected seats are unavailable, the system prompts the user to select different seats.. | | |
| Exceptions | If the system encounters an error, it displays an error message and prompts the user to try again later. | | |
| Frequency of Use | High | | |
| Assumptions | Users have valid payment information.  The booking pages are accessible and functioning correctly.  Train schedule data is accurate and up-to-date. | | |
| **Constraints** | The system must handle high volumes of booking requests. | | |
| **Dependencies** | Database for storing user and booking information. | | |
| **Inputs and Outputs** | Inputs: Selected train, seats, class, berth preferences.  Outputs: Booking confirmation details. | | |
| **Business Rules** | Ensure accurate seat availability information. | | |
| **Miscellaneous Information** | The ticket must be generated and sent to the user. | | |

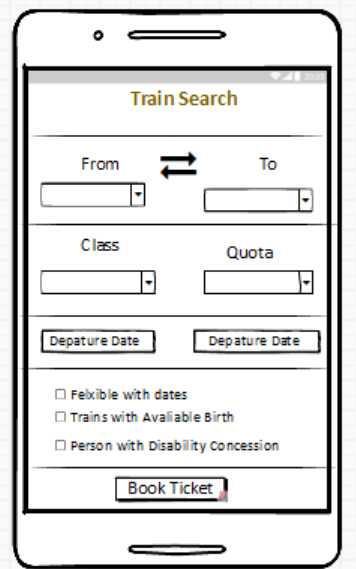
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | DC005 | | |
| **Use case Name** | Make Payment | | |
| Created By | Mr. ABC | Last Update By | April 5th 2024 |
| Date Created | March 25th 2023 | Last Revision Date | April 1st 024 |
| Actors | Customer,Payment Gateway | | |
| Description | This use case describes how users can make payments for booking train tickets in the new railway reservation system. | | |
| Pre- condition | The user must have selected a train and seats and be ready to proceed with payment. | | |
| Post condition | The payment is successfully processed, and the user receives a payment confirmation. | | |
| Normal flow of event/ Basic flow / Happy path | 1. The user reviews the booking summary and selects the payment option.  2. The user enters payment details.  3. The system processes the payment via the Payment Gateway.  4. The system confirms the payment.  5. The system generates and sends the ticket to the user.  6. The basic flow ends here. | | |
| Alternate Flow | At step 2: If the payment details are invalid, the system prompts the user to re-enter the details. | | |
| Exceptions | If the system encounters an error, it displays an error message and prompts the user to try again later. | | |
| Frequency of Use | High | | |
| Assumptions | It is assumed that users have valid payment information.  It is assumed the payment page is accessible and functioning correctly. | | |
| **Constraints** | The system must handle high volumes of payment processing requests. | | |
| **Dependencies** | Payment Gateway for processing payments. | | |
| **Inputs and Outputs** | Inputs: Payment details (card number, expiry date, CVV, etc.).  Outputs: Payment confirmation and ticket details. | | |
| **Business Rules** | Ensure secure payment processing. | | |
| **Miscellaneous Information** | The payment gateway must be reliable and provide timely responses to the payment requests. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | DC006 | | |
| **Use case Name** | Cancel Ticket | | |
| Created By | Mr. ABC | Last Update By | April 5th 2024 |
| Date Created | March 25th 2023 | Last Revision Date | April 1st 024 |
| Actors | Customer | | |
| Description | This use case describes how users can cancel their booked train tickets in the new railway reservation system. | | |
| Pre- condition | The user must be logged in and have a booked ticket to cancel. | | |
| Post condition | The ticket is successfully canceled, and the user receives a cancellation confirmation. | | |
| Normal flow of event/ Basic flow / Happy path | 1. The user navigates to the "My Bookings" page.  2. The user selects the ticket to cancel.  3. The user clicks on the "Cancel Ticket" button.  4. The system prompts the user to confirm the cancellation.  5. The user confirms the cancellation.  6. The system processes the cancellation and updates the booking status.  7. The system generates a cancellation confirmation and sends it to the user.  8. The basic flow ends here. | | |
| Alternate Flow | At step 2: If the user selects a non-cancellable ticket, the system notifies the user that the ticket cannot be canceled. | | |
| Exceptions | If the system encounters an error, it displays an error message and prompts the user to try again later. | | |
| Frequency of Use | High | | |
| Assumptions | It is assumed that users have valid bookings.  It is assumed the cancellation page is accessible and functioning correctly. | | |
| **Constraints** | The system must handle high volumes of cancellation requests. | | |
| **Dependencies** | Database for storing booking and cancellation information. | | |
| **Inputs and Outputs** | Inputs: Selected ticket for cancellation.  Outputs: Cancellation confirmation. | | |
| **Business Rules** | Ensure accurate and timely updates of booking status. | | |
| **Miscellaneous Information** | The system should provide a clear message about any applicable cancellation fees. | | |

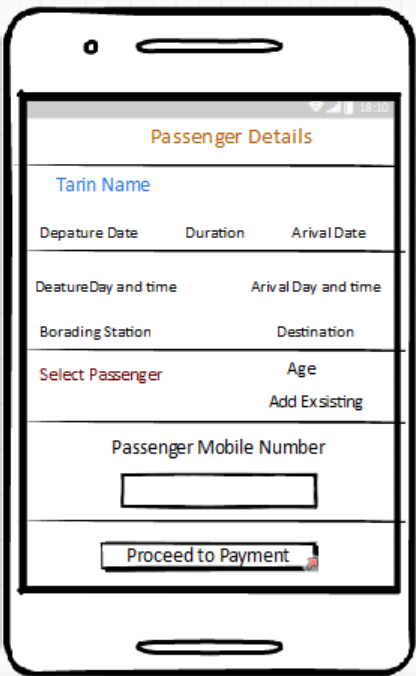
**Document 7- Screens and pages**

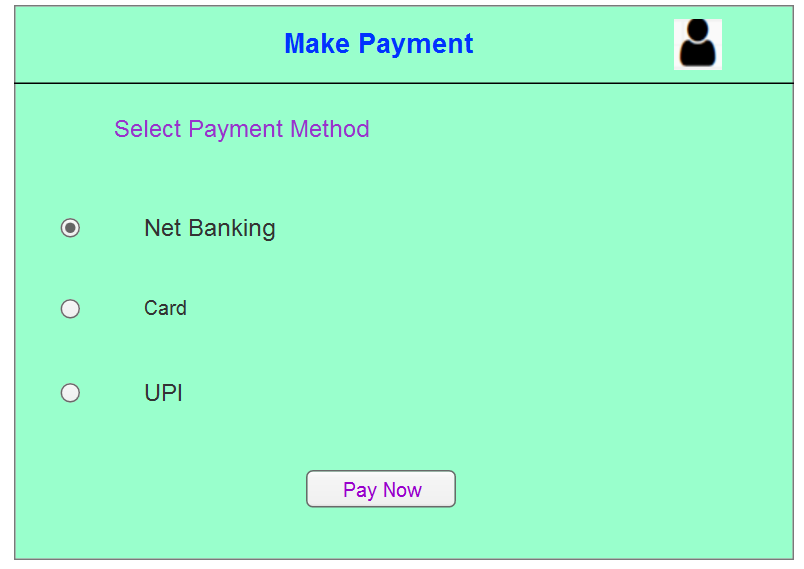
**1) Home Page 2) Sign Up**



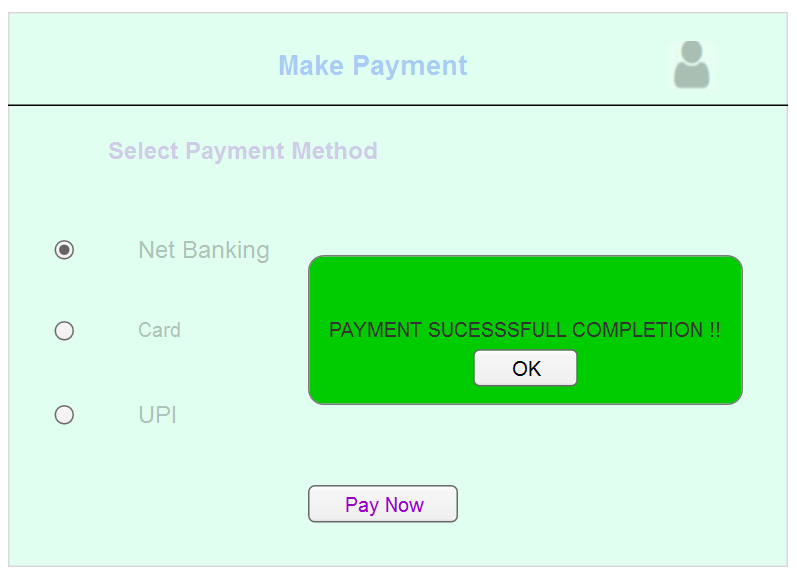
3) **Login Page**  4) **Search Page** 

5) Book Ticket 6) Make Payment

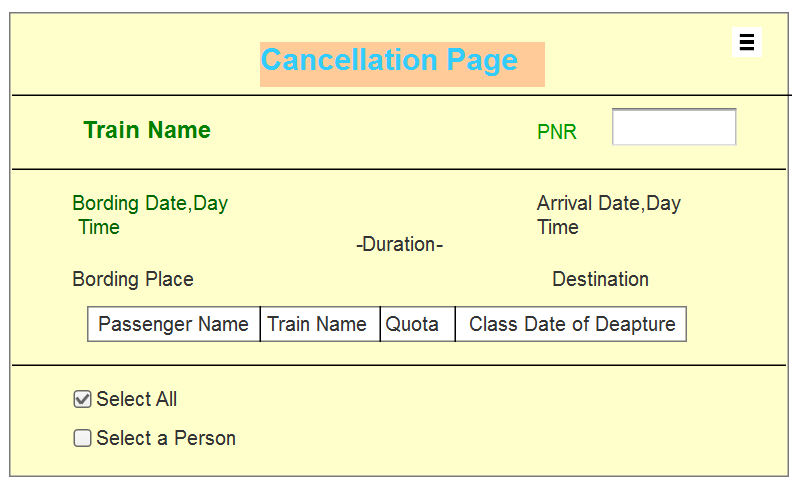




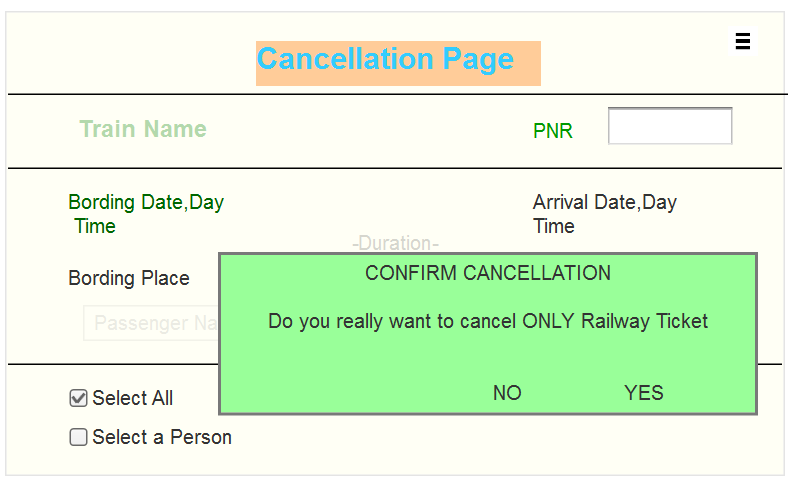
6) Payment Successful Page



7) Cancellation Page



8) Cancelation Confirmation Page



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

During the project, I used Microsoft Visio and Axure RP to make our design and development process smoother.

**Visio** was incredibly helpful for creating detailed **use case and activity diagrams.** Its user-friendly interface made it easy to show how users interact with the system and map out step-by-step workflows. This visual representation helped us identify user needs and spot potential issues early on. With its drag-and-drop tools, Visio was a **breeze to use.**

For wireframing and prototyping, I turned to **Axure RP**. This tool allowed me to design various page layouts and build interactive prototypes that closely mimicked the final user interface. Axure's intuitive interface made creating these prototypes quick and easy. Stakeholders could interact with these prototypes, giving them a real feel of the design and allowing for effective user testing and feedback.

**Document 9- BA experience My experience as BA in following phases:**

**Pre-Project Stage**

I conducted feasibility studies to evaluate the project's feasibility and align it with business objectives. I also identify key stakeholders and their needs, and document the business problem and expected outcomes.

**Artifacts Created:** Feasibility Report, Stakeholder Analysis, Business Case

**Planning Stage**

A detailed project plan outlining activities, timelines, and resources was developed. I define the project scope and objectives, and create a Work Breakdown Structure (WBS) to break down the project into manageable tasks.

**Artifacts Created:** Project Plan, Scope Statement, Work Breakdown Structure (WBS)

**Project Initiation Stage**

Clear and measurable project objectives defined, assemble the project team, and define roles and responsibilities. I also create the project charter and get stakeholder approval.

**Artifacts Created:** Project Charter, Stakeholder Register, Communication Plan

**Requirements Gathering Stage**

* Conduct stakeholder interviews and workshops.
* Use MOSCOW technique to prioritize requirements.
* Perform SWOT analysis to assess strengths, weaknesses, opportunities, and threats.
* Source alternative contacts if the client is unavailable.
* Validate requirements using FURPS technique.
* Remove duplicate requirements.
* Use prototyping for more specific requirements.
* Document everything in a Business Requirement Document (BRD).
* Identify initial risks.

**Requirements Analysis Stage**

* Ensure requirements are complete, clear, and feasible.
* Perform GAP analysis to identify gaps between the current state and the desired state.
* Create detailed functional and non-functional specifications.
* Develop a Requirements Traceability Matrix (RTM).
* Update risk assessments based on detailed analysis.
* Prepare BRS and SRS documents, Functional Specification (FS), Functional Requirement Specification (FRS), Requirements Traceability Matrix (RTM)

**Design Stage**

* Draw UML diagrams to visually describe the requirements.
* Use activity diagrams to describe the process flow.
* Communicate with the client on design and solution documents.
* Communicate diagrams to the team and consider their feedback for modifications.
* Ensure the design meets business requirements by working with the technical team.
* Review and validate design documents like the High-Level Design Document (HDD) and Architectural Design Document (ADD).
* Update risk assessments based on design reviews.
* Prepare test cases from use case diagrams.
* Write both negative and positive test cases.
* Ensure no test case is missed to avoid impacting development later.
* Prepare test data for testing.
* Update the RTM to make sure all requirements are met.

**Development Stage**

* Organized JAD sessions.
* Clarified queries from the tech team during coding.
* Handled disagreements gently with one-on-one discussions, explaining how their actions could affect the project.
* Referred to diagrams to code the unit.
* Conducted regular meetings with the technical team and the client, recording sessions for those who missed them and having one-on-one discussions as needed.
* Provided clarifications to developers about requirements.
* Ensured alignment of development with documented requirements.
* Updated the RTM to reflect the current status of requirements.
* Monitored and managed risks during development.

**Artifacts Created:** Low-Level Design Document (LLD), Component Design Document (CDD), Application

**Testing Stage**

* Assisted the QA team in creating test cases based on requirements.
* Facilitated User Acceptance Testing (UAT) to ensure the application met business requirements.
* Performed high-level testing.
* Validated that all requirements were tested and met.
* Updated the RTM to reflect testing status.
* Documented any remaining risks or issues.

**Artifacts Created:** Test Plan, Test Cases, Defect Logs, Updated RTM, Risk Register

**UAT (User Acceptance Testing)**

* Plan UAT:Develop a detailed plan for UAT.
* Conduct UAT:Work with users to test the system in a real-world environment.
* Gather Feedback:Collect feedback from users to identify any issues or improvements**.**
* Update Documentation:Reflect any changes based on UAT feedback.

**Deployment Stage**

* Assisted in planning deployment activities, including go-live plans.
* Created training materials and conducted training sessions for end-users.
* Communicated deployment plans to ensure stakeholders were aware of schedules.
* Ensured all candidates attended the meetings.
* Provided post-deployment support to address any issues.
* Forwarded RTM to the client to be attached to the project closure document.

**Artifacts Created:** Deployment Plan, Training Materials, Communication Plan, Project Closure Document