

**HIGHWAY BUS SEAT BOOKING**

**SYSTEM**

**Project Name :** SEAMLESS BUS | Highway Bus Seat Booking System

**Group Number :** CS 23

# Details of Project Supervisor, Co-supervisor, and Team members

**Group Members**

| **Name** | **Index Number** | **Registration Number** |
| --- | --- | --- |
| Mr. Sandaruwan K.L. | 21001715 | 2021/CS/171 |
| Ms. Rajapaksha T.H | 21001545 | 2021/CS/154 |
| Mr. De Hoedt D.W. | 21000379 | 2021/CS/037 |
| Ms. Thushani U.T. | 21001962 | 2021/CS/196 |

**Project Supervisor**

**Name :** Mr.Pubudu Liyanage

**Signature :**

**Date :** 29/04/2024

**Project Co-supervisor**

**Name** : MR.THARAKA ISURULAL

**Signature :**

**Date : 29/04/2024**

# Table Of Contents

1. Project Introduction
   1. Introduction
   2. Domain Description
   3. Current System & It’s limitations
   4. Proposed Solution
   5. Objectives & Goals
   6. Assumptions, Constraints & limitations
   7. Scope (In-Scope & Out-Scope)

1. Feasibility Study

1. Requirements
   1. Stakeholders (Actors)
   2. Use cases & Use case diagrams
   3. Functional Requirements
   4. Quality Attributes Requirements and how to achieve those

1. System Architecture
   1. Component Diagrams
   2. Components & their responsibilities
   3. Component Interactions

1. System Design
   1. Class Diagrams
   2. ER Diagrams
   3. Sequence Diagrams
   4. Activity Diagrams
   5. State Transition Diagrams

1. User Interface Flow
   1. Designed user interfaces

1. Usability & User Experience
2. Testing
3. Overall System Completeness
4. Special Attributes Of Project
5. Individual Contribution
6. Declaration

## 1 Introduction

### 1.1 purpose

The purpose of this document is to define the detailed requirements for the "Seamless Bus" project. This document serves as a foundation for all project stakeholders, including developers, testers, project managers, and users, to understand the goals, constraints, and functional specifications of the system.

### 1.2 Domain description

The existing system relies heavily on manual ticketing processes, requiring passengers to purchase tickets in person at bus stations or ticket counters. This manual approach can be timeconsuming and inefficient, leading to long queues and delays, especially during peak travel hours. During peak hours, buses often experience overcrowding, resulting in long queues of passengers waiting to board. Overcrowding not only inconveniences passengers but also poses challenges for bus conductors and service providers in managing passenger loads. Passengers face a significant challenge due to the lack of access to real-time updates on bus schedules, delays, and seat availability. The absence of this critical information makes it difficult for passengers to plan their journeys effectively, resulting in uncertainty and inconvenience.

While there are some online platforms available for ticket booking, these platforms can be complex and user-unfriendly. The booking process often involves multiple steps, which can be frustrating and time-consuming for frequent travelers. Conductors are required to manually check and validate passengers' tickets, which can be time-consuming, especially during peak hours when the number of passengers is high. This manual process can lead to inefficiencies and delays. The current system lacks transparency regarding bus schedules and delays. Passengers often do not have access to real-time updates on bus arrivals, leading to uncertainty about when their bus will arrive and resulting in long waiting times.

Existing online booking systems do not offer advanced reservation features, such as guaranteed seat reservations or efficient management of passenger loads during peak hours. This results in overcrowding and inconvenience for passengers. The current system does not provide passengers with flexibility in the event of schedule changes or delays. Passengers may not be promptly informed of changes, leaving them with limited options in case of unexpected disruptions. Conductors often lack tools to efficiently manage passenger reservations, making it challenging to ensure a smooth journey for passengers.

### 1.3 Current System & It’s limitations

Current system

Passengers typically purchase bus tickets through in-person transactions at bus stations or ticket counters. This manual process involves queuing and can be time-consuming. During peak travel hours, buses often experience overcrowding. This results in long queues of passengers waiting to board the buses, leading to inconveniences for travelers. Passengers lack access to real-time updates on bus schedules, delays, or seat availability. They may not know when their specific bus will arrive, causing uncertainty and inconvenience. While there are some online platforms available for ticket booking, these platforms can be complex and user-unfriendly. The booking process often involves multiple steps, making it frustrating and time-consuming for frequent travelers. Conductors manually check and validate passengers' tickets, which can be timeconsuming, especially during peak hours. This manual process can lead to inefficiencies and delays. The current system lacks transparency regarding bus schedules and delays.

Limitations

The manual processes of ticketing and validation are inefficient, leading to long waiting times for passengers. Overcrowding and inefficiencies pose operational challenges for bus service providers, impacting their ability to manage passenger loads effectively. Passengers often experience inconvenience due to the lack of real-time updates, complex online booking procedures, and waiting in long queues. The absence of real-time information leads to passenger uncertainty about bus schedules and delays. The current system does not offer advanced reservation features, such as guaranteed seat reservations or efficient management of passenger loads during peak hours. This results in overcrowding and inconvenience. The complex online booking process can be frustrating and time-consuming for passengers, especially those who travel frequently. Manual ticket validation and checking by conductors are time-consuming and prone to errors, affecting the overall efficiency of the system.

### 1.4 Proposed Solution

The "Seamless Bus" project proposes an innovative digital solution to enhance the current highway bus seat booking system in Sri Lanka It provides a user-friendly interface, supports conductor effectiveness, prioritizes security and privacy, and ensures accessibility for all passengers. Additionally, it incorporates novel features and allows for continuous improvement and scalability. The project takes a collaborative approach, involving stakeholders to refine and enhance the system based on real-world usage and feedback. Overall, "Seamless Bus" aims to transform the highway bus travel experience, making it more convenient and efficient for passengers while benefiting bus service providers.

### 1.5 Goals & Objectives

Project Goals:

The "Seamless Bus" project aims to enhance the highway bus travel experience by providing realtime updates, improving scheduling, and streamlining ticket validation. It prioritizes passenger convenience, data security, inclusivity, and collaboration with stakeholders to offer an efficient and innovative solution.

Objectives:

The objectives include creating a user-friendly platform, optimizing bus scheduling, ensuring realtime updates, streamlining ticket validation, prioritizing data security, promoting accessibility, introducing innovative features, enabling continuous improvement, and fostering collaboration. These objectives collectively aim to redefine the way passengers and bus service providers interact with highway bus services.

### 1.6 Assumptions, Constraints & limitations

Assumptions

The successful use of the "Seamless Bus" platform assumes that passengers have access to reliable internet connectivity to make bookings and access real-time updates. Passengers are assumed to have access to smartphones or computers for making reservations and accessing the system. The assumption is that conductors will adopt and effectively use the mobile app for ticket validation. The project assumes a cooperative approach from bus service providers, passengers, and other stakeholders to contribute to system improvements and adoption.

Constraints

The project will operate within the allocated budget, which may limit the extent of features and technologies that can be implemented. The project's success is constrained by the available technology stack and any limitations it may have. The project must adhere to the specified timeline, which could impact the depth of development and testing.

Limitations

The success of the system is subject to the adoption and acceptance of users, including passengers and conductors. Real-time updates are dependent on data accuracy and timely input from bus service providers, which may have limitations. The inclusivity of the system may have limitations in catering to all possible passenger needs and disabilities. Compliance with data privacy regulations is a limitation that may require ongoing monitoring and adjustments. External factors such as weather, traffic conditions, and infrastructure may still impact bus schedules and passenger experiences.

### 1.7 Scope

The scope of the "Seamless Bus" project encompasses the development and implementation of a comprehensive web-based platform and mobile application designed to revolutionize the 6 Highway Bus Seat Booking system highway bus travel experience. The project will focus on delivering the following key functionalities

* User Registration and Authentication: Allow passengers to create user accounts securely. Implement authentication mechanisms to ensure user data privacy.
* Real-time Bus Schedule and Updates: Provide passengers with access to real-time bus schedules, including available routes and time slots. Display live updates on any schedule changes, delays, or cancellations.
* Advanced Seat Booking System: Enable passengers to book bus seats in advance based on their preferred routes and time slots. Implement a seat reservation mechanism that optimizes available seats to accommodate passenger demand.
* Mobile Application for Conductors: Develop a mobile application for conductors to validate passenger tickets efficiently using QR code scanning and manual entry.
* User Profiles and Preferences: Allow registered users to create profiles and save their travel preferences, making future bookings faster and more personalized.
* Security and Data Protection: Implement robust security measures to protect user data and payment information. Ensure compliance with data protection regulations, including GDPR standards.

Out of scope

The project does not involve the physical operation of buses, driver management, or maintenance aspects. While the project handles seat reservations, it does not handle financial transactions or payments directly. External payment gateways will be used. Ensuring the legal compliance of bus drivers and conductors with regulations falls outside the scope of the project. The project assumes that users have access to their own devices (smartphones, computers) and internet connectivity.

## 2 Feasibility Study

Technical Feasibility

The technical feasibility of the "Seamless Bus" project is a critical consideration to ensure that the proposed system can be effectively developed and implemented. Here are the key aspects of technical feasibility:

* Choice of Technology: The project had selected PHP as the primary programming language for web development. PHP is a widely-used language for web applications and is technically feasible for building the proposed platform.

* Server Requirements: The server requirements, including web hosting, database management, and server resources, are well-defined and can be met using available technology and hosting providers.

* Database Integration: The project intends to integrate with databases to manage user information, bus schedules, and bookings. Popular databases like MySQL is technically feasible for this purpose.

* API Integration: To provide real-time updates and notifications, the project may need to integrate with external services and APIs. These integrations are technically feasible based on the availability and compatibility of APIs.

* Mobile App Development: Creating a mobile app for conductors is feasible using technologies Swift

* Data Security: Ensuring data security and privacy is technically feasible through the implementation of encryption, secure authentication, and compliance with data protection regulations.

* Scalability: The project's scalability is technically feasible by designing the system architecture to accommodate increasing user demand and potential expansion to new routes.

* User Interface: The development of a user-friendly interface is technically feasible using web design and development best practices to ensure accessibility and ease of use.
* System Testing: The project's technical feasibility includes the ability to conduct rigorous testing to identify and address issues, ensuring the system's reliability and functionality.

Economic Feasibility

The economic feasibility of the "Seamless Bus" project is promising. The use of cost-effective technologies, open-source platforms, and careful budgeting ensures that initial development costs are manageable. The project anticipates a positive return on investment through improved operational efficiency for bus service providers and increased passenger satisfaction. The revenue generation model for bus service providers adds to the project's sustainability. Economic feasibility is further enhanced by a thorough cost-benefit analysis and ongoing cost management. Overall, the project is well-positioned to achieve economic viability.

Operational Feasibility

Operational feasibility is a pivotal aspect of our project's success, and our analysis confirms that the proposed system will seamlessly integrate into existing operations:

* Bus conductors' transition to the mobile app for ticket validation is anticipated to be smooth due to the app's user-friendly interface and QR code validation.
* Passengers' engagement with the booking system is designed to be intuitive and convenient, aligning with their current booking behaviors.
* The system's adaptability to dynamic schedule changes ensures its effectiveness in realworld operational scenarios.

Legal & Ethical Feasibility

The legal and ethical feasibility of the "Seamless Bus" project is a key consideration. It prioritizes data security, privacy, and compliance with data protection regulations. The project is committed to transparent data usage and consent mechanisms to ensure ethical data handling. This commitment, along with robust encryption and secure authentication, enhances the legal and ethical feasibility of the project. The goal is to build trust among users and maintain adherence to legal and ethical standards in data management.

Schedule Feasibility

The project timeline is designed to ensure the completion of the "Seamless Bus" system within the set timeframe:

* A phased development approach, following the Waterfall method, ensures that each component is thoroughly developed, tested, and integrated.

* The timeline considers potential risks and contingencies, allowing room for adjustments while adhering to the specified delivery date.

* Estimated working hours per a week = 5 hours

Estimated working hours per a weekend = 8 hours

Estimated working hours per a week = 13 hours (By 1 team member)

Estimated working hours for 36 weeks = 13 hours x 36 = 468 hours (By 1 team member)

In conclusion, our feasibility analysis demonstrates that the "Seamless Bus" project is not only technically viable but also economically, operationally, and ethically sound. The comprehensive considerations undertaken across these dimensions assure stakeholders and investors of the project's success in revolutionizing the highway bus travel experience.

## 3 Requirements

### 3.1 Stakeholders (Actors)

3.1.1 Registered passenger

* Identified as passengers who create user accounts on the platform.
* They use the system for booking bus tickets and accessing real-time bus schedules.
* They are key users of the platform and have personal accounts for booking convenience.

3.1.2 Guest Passenger

* Identified as passengers who use the platform to book bus tickets without creating user accounts.
* They may choose to create accounts for a more personalized experience.
* They access bus schedules without the need for account creation.

3.1.3 Conductor

* Identified as bus conductors responsible for validating passenger tickets.
* They are provided with a mobile app for ticket validation.
* They access booking lists and use the app for efficient ticket checking

3.1.4 Scheduler

* Identified as individuals responsible for managing and updating bus schedules.
* They log in to the system to make real-time schedule changes.
* They allocate buses to time slots and oversee schedule operations.

3.1.5 Owner

* Identified as owners of bus services or companies that register in the system.
* They provide bank details for revenue collection.
* They have access to system reports, select conductors for buses, and manage their services.

3.1.6 Administrator

* Identified as individuals responsible for system administration and user management.
* They verify and add schedulers to the system, ensuring smooth operations.
* They play a crucial role in system governance and user access control.

3.2 Use cases & Use case diagrams

#### 3.2.1 Registered user use case

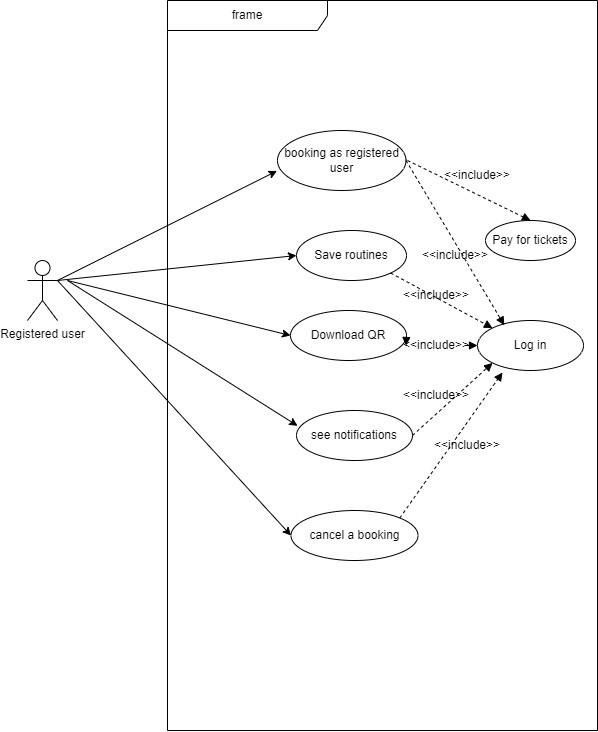


Figure 1

#### 3.2.2 Guest user use case

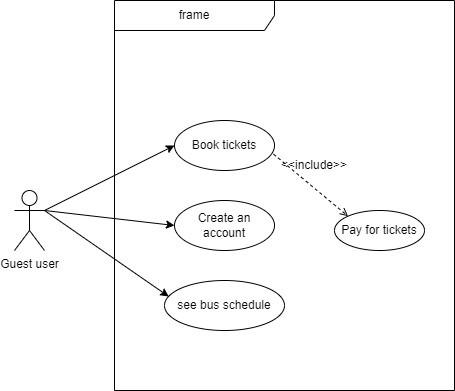
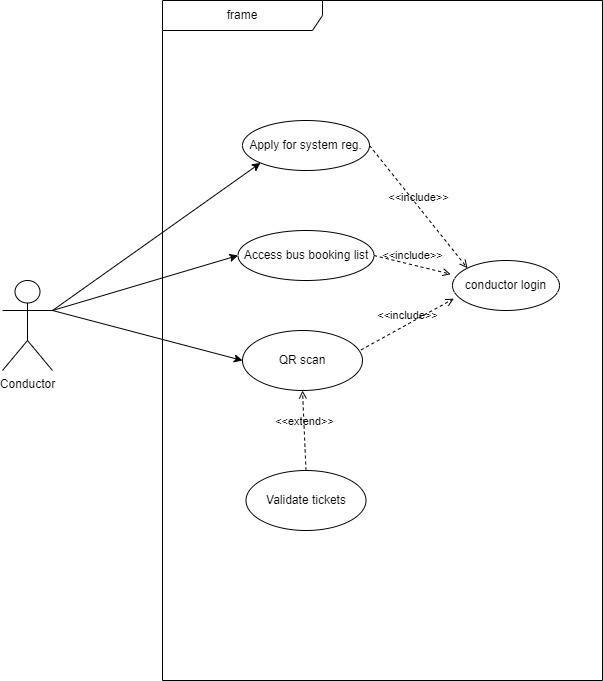


Figure 2

#### 3.2.3 Conductor use case

Figure 3 

#### 3.2.4 Scheduler use case

###### Figure 4

#### 3.2.5 Owner use case

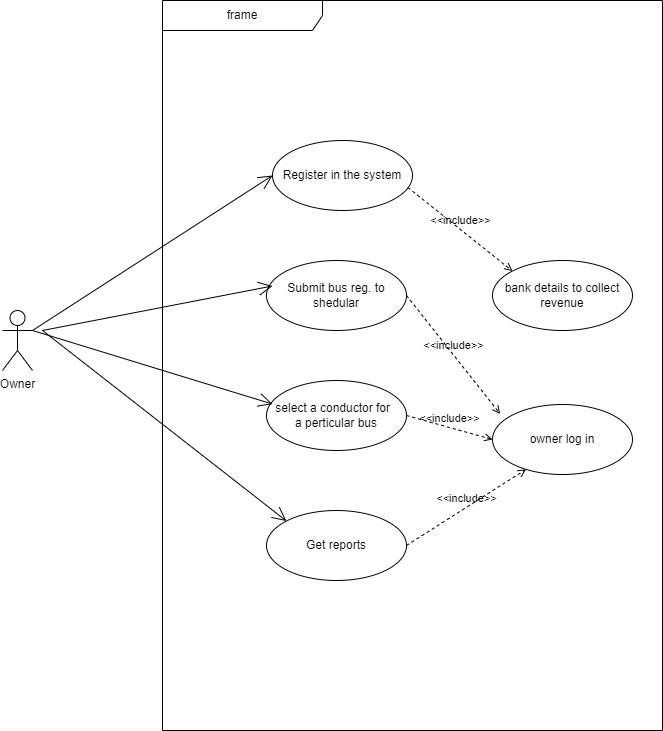


Figure 5

#### 3.2.6 Administrator use case

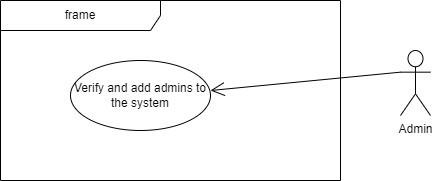


Figure 6

Use Case Narratives

Passenger Registration

Figure 7

|  | Use Case 01 : User Registration |  |
| --- | --- | --- |
| Actor | | Passenger |
| Precondition | | Passenger has access to the system |
| Main Success Scenario | | * Passenger accesses the system and selects the "Register" option. * Passenger provides required information, including name, contact details, and password. * System validates the provided information and creates a user account. * System confirms successful registration and provides login credentials. |
| Alternative Scenarios | | * Scenario 1: If the provided information is incomplete or invalid, the system prompts the passenger to correct it. * Scenario 2: If the chosen username is already taken, the system requests an alternative username. |
| Postcondition | | Passenger's account is created, and they can log in using the provided credentials. |

Passenger Book Tickets

Figure 8

|  | Use Case 02 : Passenger Book a Ticket |  |
| --- | --- | --- |
| Actor | | Passenger |
| Precondition | | Passenger has an account on Seamless Bus and is logged in. |
| Main Success Scenario | | * Passenger selects the bus route and departure time. * Passenger selects the number of seats they want to book. * Passenger enters their contact information. * Passenger pays for the tickets. * Passenger receives a confirmation email with their ticket details. |
| Alternative Scenarios | | * Scenario 1: The bus is full. The passenger is notified and can either choose to wait for the next bus or cancel their ticket. * Scenario 2: The passenger enters incorrect contact information. The passenger is asked to correct their information before proceeding. * Scenario 3: The passenger's payment is declined. The passenger is asked to try again with a different payment method. |
| Postcondition | | The passenger has a confirmed ticket for the selected bus route and departure time. |

Passenger cancel a Ticket

Figure 9

|  | Use Case 02 : Passenger cancel a Ticket |  |
| --- | --- | --- |
| Actor | | Passenger |
| Precondition | | Passenger has a confirmed ticket for a bus journey. |
| Main Success Scenario | | * Passenger logs into Seamless Bus.      * Passenger selects the ticket they want to cancel. * Passenger confirms that they want to cancel the ticket. |
|  | | ❖ Passenger's ticket is canceled and the passenger is refunded the full amount |
| Alternative Scenarios | | * Scenario 1: The passenger cannot 21 find their ticket. The passenger can contact customer support for assistance. * Scenario 2: The passenger's ticket is not eligible for cancellation. The passenger is not refunded the full amount. |
| Postcondition | | The passenger's ticket is canceled and the passenger is refunded the full amount, if applicable. |

Conductor System Registration

Figure 10

|  | Use Case 02 : conductor system registration | |
| --- | --- | --- |
| Actor | | Conductor |
| Precondition | | The conductor is not registered in the system |
| Main Success Scenario | | * The conductor accesses the system registration page * The conductor enter the their name, phone number, NIC number and other details * The conductor create the passwords. * The conductor clicks the “register” button. * The admin validates the conductor details and verifies the details. * The system creates the conductors account. * The system sends an SMS confirmation to the conductors. * System gives access to use the mobile   application. |
| Alternative Scenarios | | * Scenario 1 :conductor enter the invalid information the system display error message. * Scenario 2 :the conductor details is already in use the system display an error message and already use this details. |
| Postcondition | | The conductor has an account in the system |

QR Scan

Figure 11

|  | Use Case 02 : QR Scan |  |
| --- | --- | --- |
| Actor | | Conductor |
| Precondition | | The conductor is logging into the system and is on a bus with booked passengers.The separate mobile app for QR scanning is installed on the conductors’ mobile. |
| Main Success Scenario | | * The conductor accesses the bus booking list for the ongoing trip through the mobile application . * The system displays the booked seats. * Passengers present their booking QR codes. * Conductor open the mobile app and scanning the QR codes . * The mobile app verifies the QRs and marks the passengers as boarded. * The mobile app communicate with the system and update the onboard recorder. * 4-7 steps repeats |
| Alternative Scenarios | | ❖ If the QR code cannot be scanned due to technical issues with the mobile app, the conductor manually verifies the passenger's booking details and updates the onboard record through the main system. |
| Postcondition | | Passenger’s seat is booked |

Conductor Logging

Figure 12

|  | Use Case 03 : Conductor logging |  |
| --- | --- | --- |
| Actor | | Conductor |
|  | |  |
| Precondition | | The conductor has an account in the system and logging in |
| Main Success Scenario | | * The conductor clicks the “ logout” button. * The system logs the conductors out of   the system. |
| Alternative Scenarios | | ❖ If the conductor enters incorrect login credentials, the system prompts them to retry |
| Postcondition | | The conductor has successfully logged into the system and gained access to their assigned tasks. |

Accessing The Booking List

Figure 13

|  | Use Case 04 : Accessing The Booking List |  |
| --- | --- | --- |
| Actor | | Conductor |
| Precondition | | The conductor has an account in the system using the mobile application |
| Main Success Scenario | | * The conductor logs into their registered account using the mobile Application. * The mobile app presents a Dashboard With the various option , including accessing the “bus booking list”. * Conductor selects the desired bus route,bus number, date and time. * Conductor can view how many seats fill in the bus. |
| Alternative Scenarios | | ❖ If there are no booked seats for the selected trip, the mobile app notifies the conductor accordingly. |
| Postcondition | | The conductor has successfully accessed the bus booking list for the specified trip using the mobile application. |

Verify Conductors

Figure 14

|  | Use Case 01: Verify Conductors |  |
| --- | --- | --- |
| Actor | | scheduler |
| Precondition | | Scheduler is at the bus stand and The scheduler has an account in the system. |
| Main Success Scenario | | * The scheduler logs in to the system. * The scheduler clicks the "Verify   Conductors" tab. The system displays a list of all pending conductor verifications  .   * The scheduler can view the details of each conductor, including their name, contact number, and assigned route,and legal documentary. * The scheduler can verify the conductor's identity by checking their ID card, driver's license,and legal documents. * The scheduler can mark the conductor as verified. |
| Alternative Scenarios | | ❖ The conductor's ID card or driver's license is invalid.The scheduler does not verify the conductor's identity. |
| Postcondition | | The conductor is verified. |

Verify The Buses

Figure 15

|  | Use Case 01: Verify the buses |  |
| --- | --- | --- |
| Actor | | scheduler |
| Precondition | | The scheduler has an account in the system. |
| Main Success Scenario | | * The scheduler logs in to the system. The scheduler clicks the "Verify Buses" tab. * The system display a list of pending bus verifications . * The scheduler can view the details of each bus, including its license plate number, make, model, and capacity. |
|  | | * The scheduler can verify the bus's identity by checking its license plate number. * The scheduler can mark the bus as   verified |
| Alternative Scenarios | | ❖ The bus's license plate number is invalid. The scheduler does not verify the bus's identity |
| Postcondition | | The bus is verified. |

Update Schedules

Figure 16

|  | Use Case 03: Update Schedules |  |
| --- | --- | --- |
| Actor | | scheduler |
| Precondition | | The scheduler has an account in the system |
| Main Success Scenario | | * The scheduler logs in to the system. * The scheduler clicks the "Update Schedules" tab. * The system displays a list of all scheduled bus routes. * The scheduler can view the details of each bus route, including the departure time, arrival time, and stops . * The scheduler can update the schedule for any bus route. * The scheduler clicks the "Save" button to   save the changes. |
| Alternative Scenarios | | * The scheduler tries to update a schedule for a bus route that does not exist. * The system displays an error message. * The scheduler tries to update a schedule for a bus route that is already in use. * The system displays an error message. |
| Postcondition | | The buses are added to the time slots. |

Accessing All Booking List

Figure 17

|  | Use Case 04: Accessing all booking list | |
| --- | --- | --- |
| Actor | | scheduler |
| Precondition | | The scheduler has an account in the system |
| Main Success Scenario | | * The scheduler logs in to the system. * The scheduler clicks the "All Booking Lists" tab. * The system displays a list of all booking lists for all bus routes. The scheduler can view the details of each booking list, including the number of bookings. * Give the updates to the conductors with number of booking seats. * View the total revenue, and the start and   end dates. |
| Alternative Scenarios | | * There are no booking lists for any bus routes. * The system displays a message indicating   that there are no booking lists. |
| Postcondition | | The scheduler has access to all booking list |

Adding Buses To Time

Figure 18

|  | Use Case 05: Adding Buses To Time |  |
| --- | --- | --- |
| Actor | | scheduler |
| Precondition | | The scheduler has an account in the system |
| Main Success Scenario | | * The scheduler logs in to the system. * The scheduler clicks the "Add Buses to Time Slots" tab. * The system displays a list of all available time slots. * The scheduler can choose a time slot and add one or more buses to the time slots. * The scheduler clicks the "Save" button to   save the changes |
| Alternative Scenarios | | ❖ The scheduler tries to add buses to a time slot that does not exist. |
|  | | ❖ The system displays an error message |
| Postcondition | | The buses are added to the time slots. |

Register in the system

Figure 19

|  | Use Case 01: Register in the system |  |
| --- | --- | --- |
| Actor | | owner |
| Precondition | | The bus Owner does not have an account in the system |
| Main Success Scenario | | * The bus owner visits the system registration page. * The bus owner enters their name, contact number, and email address. ❖ The bus owner creates a password. * The bus owner clicks the "Register" button. * The system validates the bus owner's information. * The system creates the bus owner's account. * The system sends as SMS |
| Alternative Scenarios | | * The bus owner enters invalid information. * The system displays an error message * The bus owner register the system   already in use display an error message |
| Postcondition | | The bus owner has an account in the system |

Submitted bus registration to the Scheduler

Figure 20

|  | Use Case 02: submitted bus registration to the scheduler | |
| --- | --- | --- |
| Actor | | owner |
| Precondition | | The bus owner has an account in the system |
| Main Success Scenario | | * The bus owner logs in to the system. * The bus owner clicks the "Bus Registration" tab. |
|  | | * The bus owner enters the details of their bus, including the license plate number, make, model, and capacity. * The bus owner clicks the "Submit" button. * The system validates the bus registration information. * The system submits the bus registration information to the scheduler. * The scheduler receives an SMS notification of the bus registration. |
| Alternative Scenarios | | * The bus owner enters invalid information. * The system displays an error message. * The bus owner's bus is already registered in the system. * The system displays an error message. |
| Postcondition | | The bus registration information is submitted to the scheduler. |

Selecting the conductor for a particular bus

Figure 21

|  | Use Case 03: submitted bus registration to the scheduler | |
| --- | --- | --- |
| Actor | | owner |
| Precondition | | The bus owner has an account in the system |
| Main Success Scenario | | * he bus owner clicks the "Select Conductor" tab. * The system displays a list of all conductors who are currently available. * The bus owner can choose a conductor for their bus. * The bus owner clicks the "Select" button. * The system assigns the conductor to the   bus |
| Alternative Scenarios | | ❖ There are no conductors who are currently available. |
|  | | * The system displays a message indicating that there are no conductors who are currently available. * The bus owner does not select a   conductor for their bus. |
| Postcondition | | The conductor is assigned to the bus. |

Getting Reports

Figure 22

|  | Use Case 04: Getting reports |  |
| --- | --- | --- |
| Actor | | owner |
| Precondition | | The bus owner has an account in the system |
| Main Success Scenario | | * The bus owner clicks the "Reports" tab. * The system displays a list of all available reports. * The bus owner can choose a report to view. * The system generates the report and   displays it to the bus owner. |
| Alternative Scenarios | | * There are no reports available. * The system displays a message indicating that there are no reports available. * The bus owner does not view any   reports |
| Postcondition | | The bus owner does not view any reports. |

Adding the bank details

Figure 23

|  | Use Case 04: Adding the bank details |  |
| --- | --- | --- |
| Actor | | owner |
| Precondition | | The bus owner has an account in the system |
| Main Success Scenario | | * The bus owner log in to the system. * The bus owner click to the bank details. |
|  | | * The bus owner enter the their bank account number routing number and name of the bank. * The bus owner click the “save “ button. * The system validate the bank details. ❖ The system save the bank details |
| Alternative Scenarios | | ❖ The bus owner enters invalid bank details The system display an error message. |
| Postcondition | | The bus owner’ bank details are saved |

Logging The System

Figure 24

|  | Use Case 04: Logging The System |  |
| --- | --- | --- |
| Actor | | owner |
| Precondition | | The bus owner has an account in the system |
| Main Success Scenario | | * The bus owner visits the system login page. * The bus owner enters their username and password . * The bus owner click the login button . * The system validates the owner's credentials . * Owner system logging |
| Alternative Scenarios | | * The bus owner enters invalid credentials. * The system displays an error message. * The bus owner enters their correct   credentials and tries to log in again. |
| Postcondition | | The bus owner logged into to the system |

### 3.3 Functional Requirements

3.3.1 Registered Passenger

* Log into user account
* Book tickets
* Save routines
* See bus schedules
* See notifications
* Cancel a booking
* Download QR

3.3.2 Guest Passenger

* Book Tickets
* Create an account and become a registered user
* See bus schedule

3.3.3 Conductor

* Apply for system registration Conductors will get an mobile app to,
* Access booking list
* Scan QR and validate tickets

3.3.4 Scheduler

* Login
* Verify Conductors
* Verify Buses to system
* Update/edit schedule
* Add busses to timeslots
* Access all booking lists

3.3.5 Owner

* Register in the system
* Input bank details to get revenue
* Login
* Verify buses to system
* Select conductors for buses
* See reports

3.3.6 Administrator

* Verify and add schedulers to the system

### 3.3 Quality Attributes Requirements and how to achieve those

Usability

**Requirement**: The system should be user-friendly for both passengers and conductors.

**Achievement**: Achieve usability through a user-centered design approach. Conduct user testing and gather feedback for iterative improvements. Ensure a clean and intuitive user interface with clear navigation.

Reliability

**Requirement**: Passengers rely on accurate bus schedules, and conductors need a dependable ticket validation system.

**Achievement**: Implement robust data management and ensure real-time updates are accurate. Regularly monitor and maintain system performance to minimize downtime.

Performance

**Requirement**: The system should handle a large number of concurrent users without slowdowns or crashes.

**Achievement**: Employ efficient coding practices, optimize database queries, and utilize caching mechanisms. Perform load testing to identify and address performance bottlenecks.

Security

**Requirement**: Passenger data and payment information must be secure.

**Achievement**: Use encryption, secure authentication, and access control mechanisms to protect sensitive data. Regularly update and patch system components to address security vulnerabilities.

Scalability

**Requirement**: The system should be able to accommodate increased user demand and potential expansion to new routes.

**Achievement**: Design a scalable architecture that can handle growing user numbers. Use cloudbased solutions to dynamically allocate resources as needed.

Maintainability

**Requirement**: The system should be easy to maintain and update as needed.

**Achievement**: Adhere to coding standards and use version control systems. Document the codebase and establish a process for regular updates and enhancements.

Accessibility

**Requirement**: The system should be accessible to passengers with disabilities.

**Achievement**: Follow accessibility guidelines and best practices to ensure that the system is usable by individuals with different needs.

Data Privacy

**Requirement**: Passenger data must be handled with the utmost privacy and comply with data protection regulations.

**Achievement**: Implement strong data anonymization practices, ensure clear consent mechanisms, and regularly audit data handling processes to maintain privacy and compliance.

Flexibility

**Requirement**: Passengers should have flexibility in managing their bookings and schedules.

**Achievement**: Offer features like seat cancellation and provide real-time updates for schedule changes to enhance passenger flexibility.

Collaboration

**Requirement**: The system should support collaboration with bus service providers and other stakeholders.

**Achievement**: Establish a robust feedback loop with stakeholders, conduct regular meetings, and use collaborative tools for effective communication and cooperation.

4 System Architecture

##### 4.1 Component Diagram

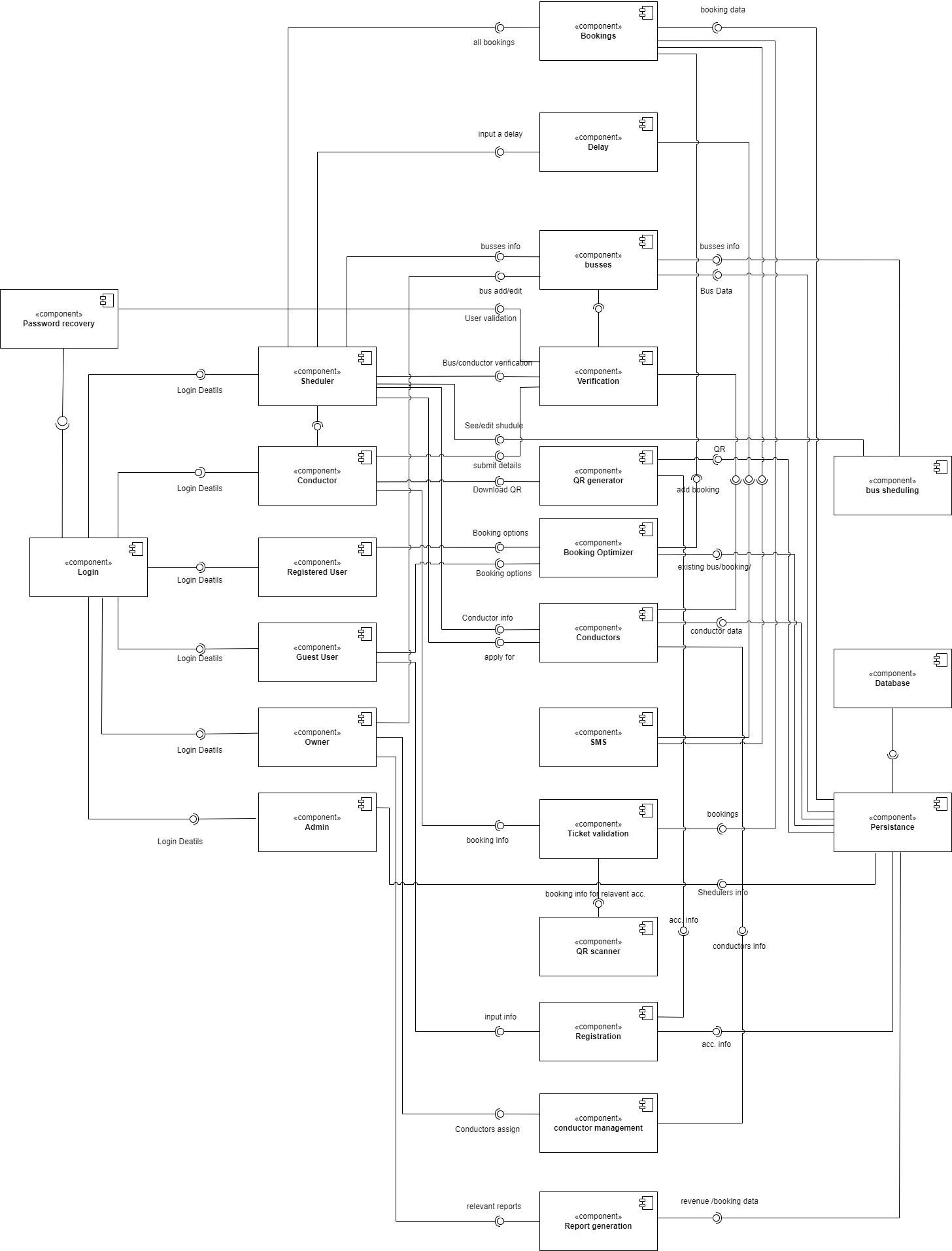


Figure 25

##### 4.2 Components and their Responsibilities

Figure 26

| **Component** | **Responsibility** |
| --- | --- |
| Login | * Authenticates users to access the system. * Allows users to create and manage their accounts. * Provides users with access to their account information, such as booking history and upcoming trips. |
| Booking | * Allows users to search for and view available bus tickets. * Displays detailed information about bus tickets, such as the departure and arrival times, bus type, and fare. * Allows users to select and book bus tickets. * Processes payments for bus tickets. * Generates and sends confirmation emails to users for their bookings. |
| Conductor Management | * Allows conductors to view their schedules and passenger lists. * Allows conductors to check in passengers and validate tickets. * Provides conductors with tools to manage passenger boarding and disembarkation. * Generates reports for conductors on their passenger loads and revenue. |
| Registration | * Allows users to register for events that are associated with bus tickets. * Collects user information, such as name, email address, and contact information. * Provides users with access to event information, such as the date, time, and location. * May also allow users to purchase additional items, such as event merchandise or food and beverage vouchers. |
|  |  |
| Ticket Validation | * Validates bus tickets to ensure that they are valid and have not been used previously. * May use a variety of methods to validate tickets, such as barcode scanning, QR code scanning, or ticket numbers. * May also be used to track passenger boarding and disembarkation. |
| Delay | * Responsible for monitoring for delays in bus arrivals and departures. * It may also be responsible for updating the system accordingly and notifying users of any delays. |
| Password Recovery | * This component allows users to recover their passwords if they forget them. * It may do this by sending a password reset link to the user's email address or by asking the user security questions. |
| Component Scheduler | * This component may be responsible for scheduling buses and assigning conductors to them. * It may also be responsible for generating reports on bus schedules and conductor assignments. |
| Buses | * Represents the buses that are available for booking in the system. * It may contain information about the bus type, capacity, and amenities. |

### 4.3 Component Interactions

Login

**User Database**: The Login component queries the User Database to authenticate users.

**Session Manager**: The Login component generates a session token and stores it in the user's browser when the user is successfully authenticated. The Session Manager is responsible for managing the user's session and ensuring that they are logged in to the system.

Booking

**Bus Schedule**: The Booking component queries the Bus Schedule to find available buses and tickets.

**Payment Processor**: The Booking component processes payments for bus tickets.

**Email Service**: The Booking component sends confirmation emails to users for their bookings.

**Passenger Database**: The Booking component stores information about passengers in the Passenger Database. This information is used to generate passenger lists for conductors and to track passenger bookings.

Conductor Management

**Bus Schedule**: The Conductor Management component queries the Bus Schedule to view conductor schedules and passenger lists.

**Ticket Validation**: The Conductor Management component uses the Ticket Validation component to validate passenger tickets.

**Passenger Database**: The Conductor Management component uses the Passenger Database to check in passengers and manage passenger boarding and disembarkation.

Registration

**Event Database**: The Registration component queries the Event Database to get information about events.

**Passenger Database**: The Registration component stores information about passengers in the Passenger Database. This information is used to generate attendee lists for events and to track passenger registrations.

Ticket Validation

**Passenger Database**: The Ticket Validation component queries the Passenger Database to check the validity of tickets.

Delay

**Bus Schedule**: The Delay component may query the Bus Schedule to get the latest bus locations and arrival times.

**Notification Service**: The Delay component may use the Notification Service to notify users of bus delays.

Password Recovery

**User Database**: The Password recovery component may query the User Database to get user information.

**Email Service:** The Password recovery component may use the Email Service to send password reset links to users.

Component Scheduler

**Bus Schedule**: The Component Scheduler component uses the Bus Schedule to schedule buses and assign conductors to them.

**Conductor Database**: The Component Scheduler component stores information about conductors in the Conductor Database.

Buses

**Bus Schedule**: The Busses component is used by the Bus Schedule to generate bus schedules.

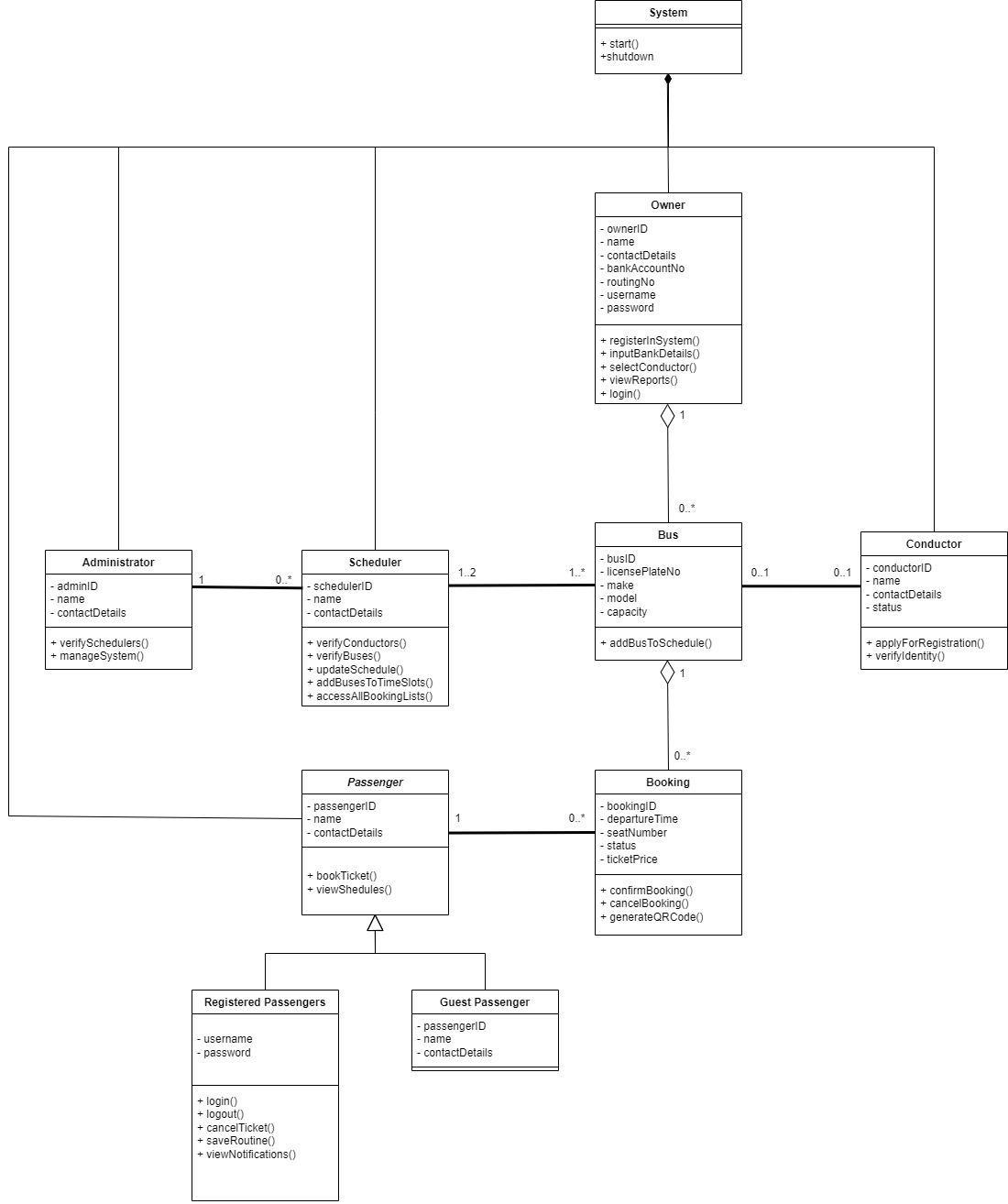
**Booking**: The Busses component is used by the Booking component to display available buses to users.

**Conductor Management**: The Busses component is used by the Conductor Management component to provide conductors with information about the buses they are assigned to.

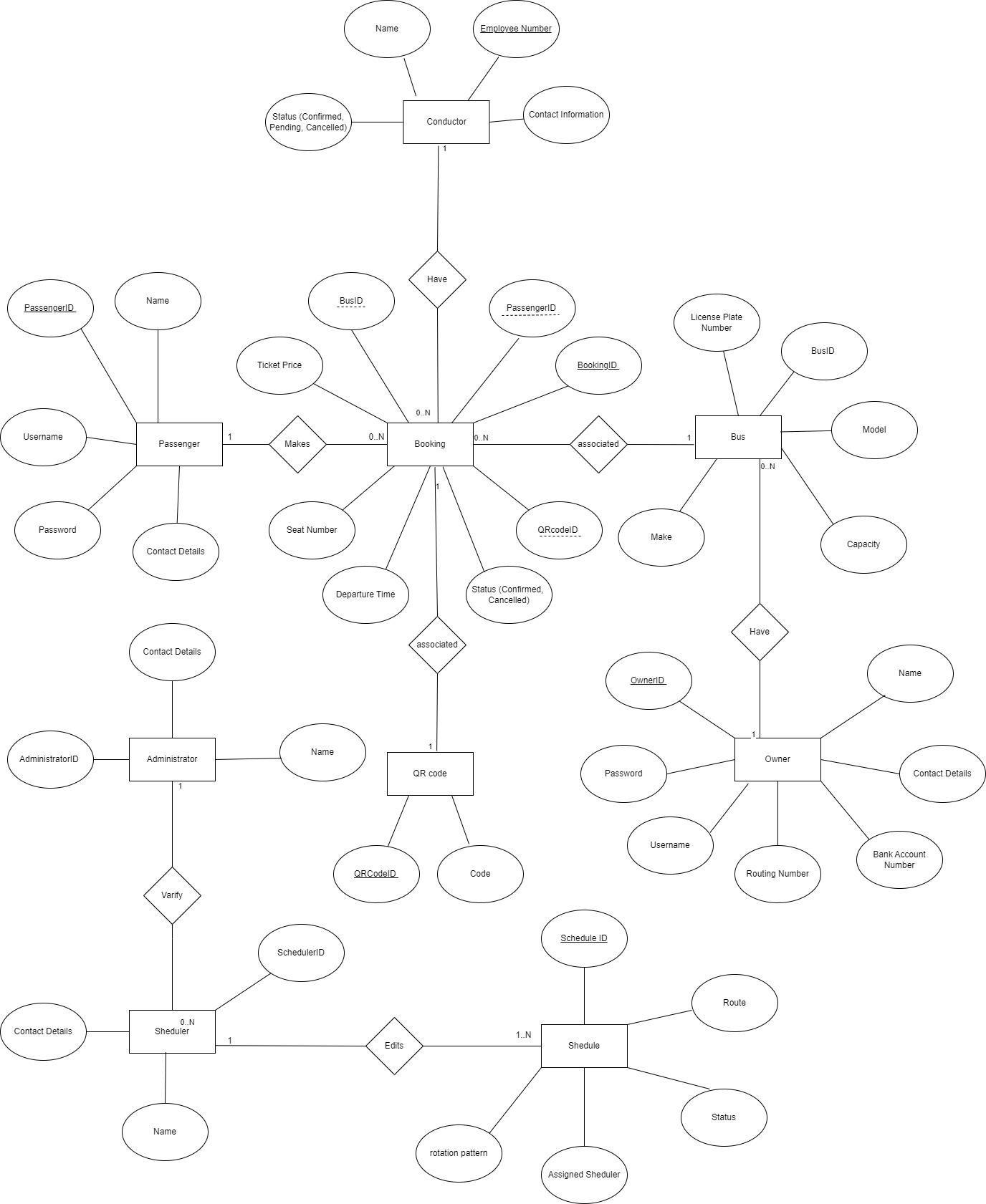
5 System’s Design

### 5.1 Class Diagram

Figure 27

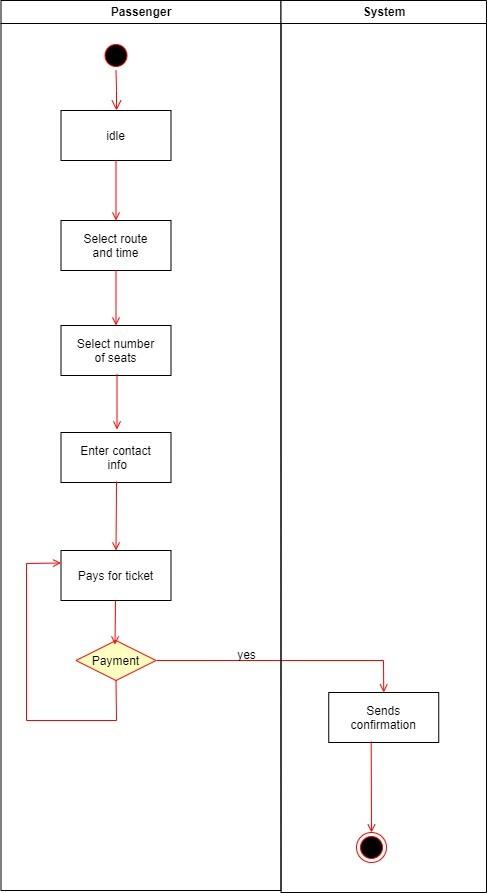


### 5.2 ER Diagram

Figure 28 

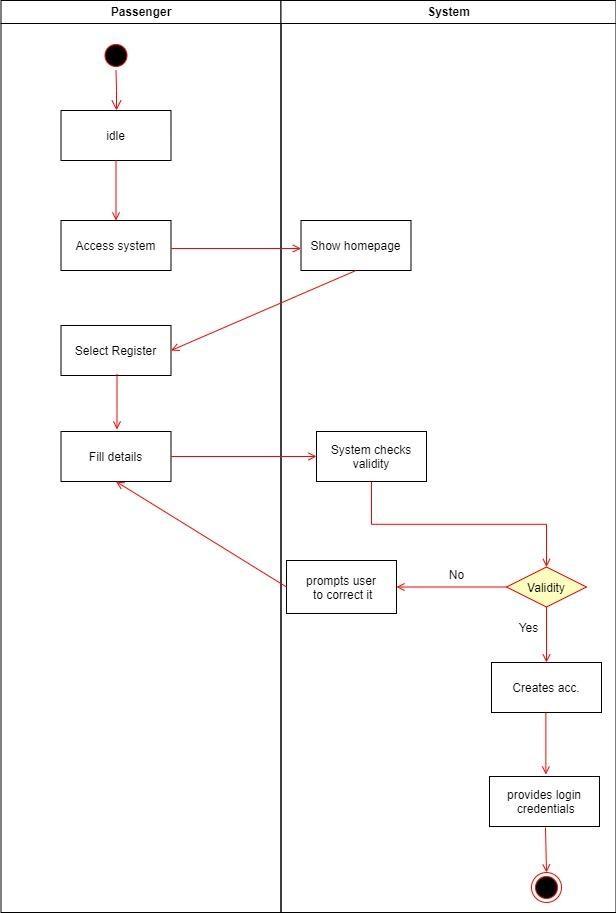
### 5.3 Activity Diagram

5.3.1 Guest Passenger Seat Booking

Figure 29 

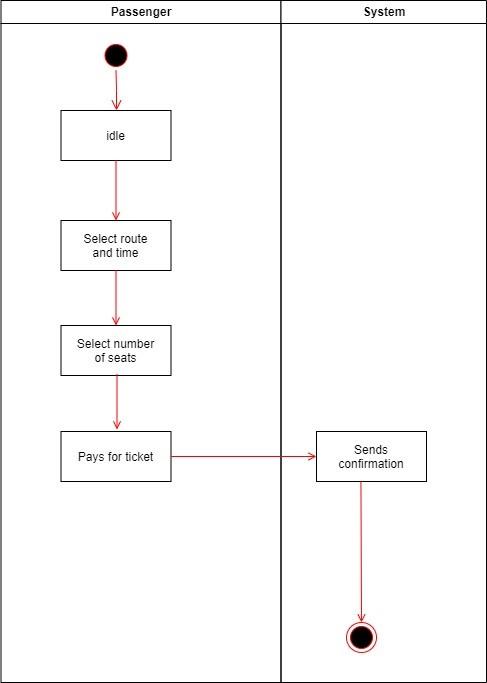
5.3.2 Passenger Seat Booking

Figure 30



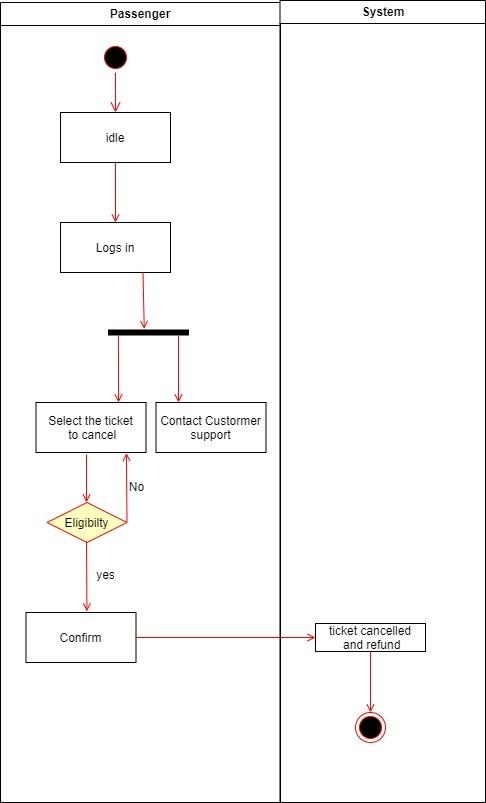
#### 5.3.3 Registered Passenger Seat Booking

Figure 31



5.3.4 Passenger Booking Cancellation

Figure 32



### 5.4 State Diagram

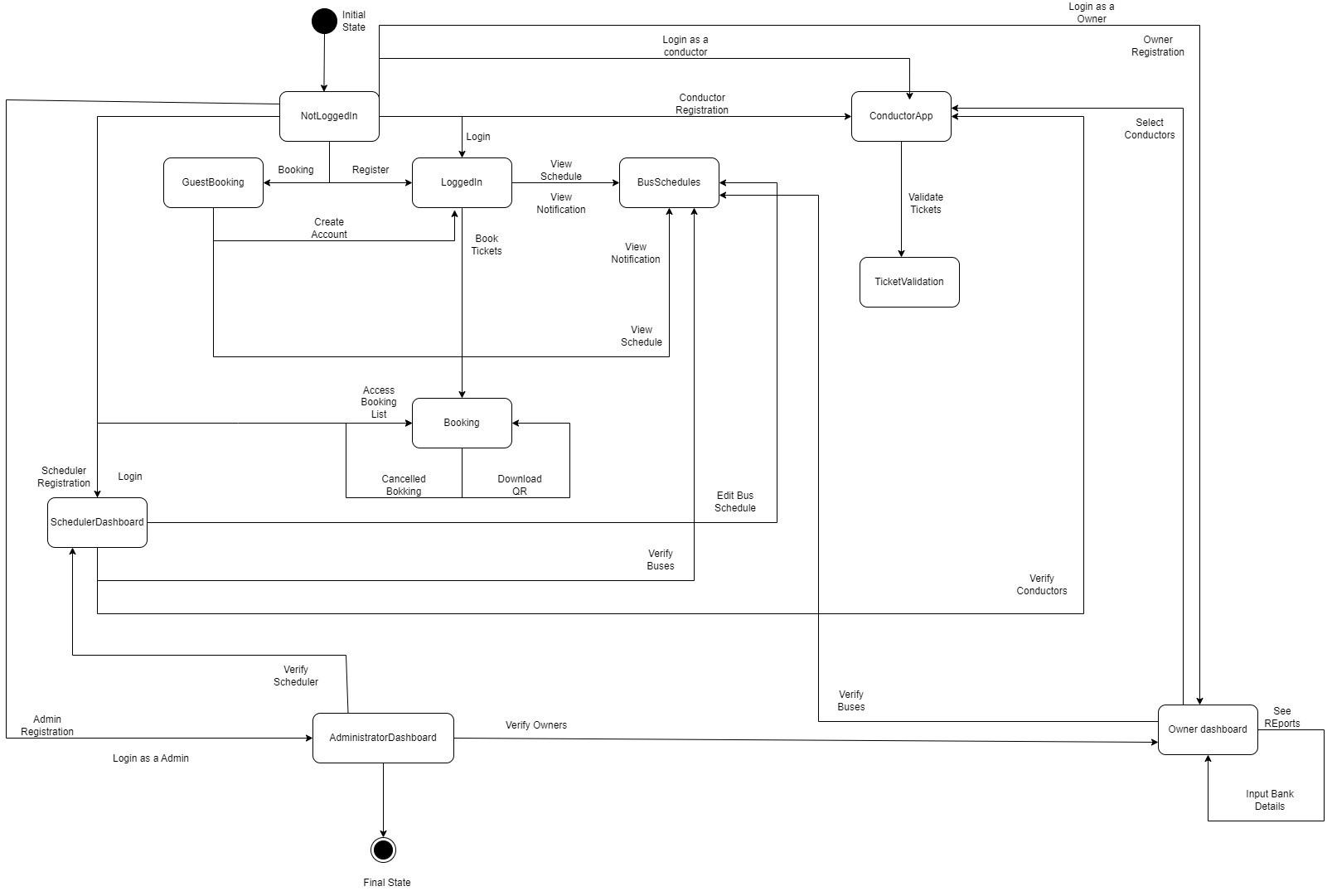
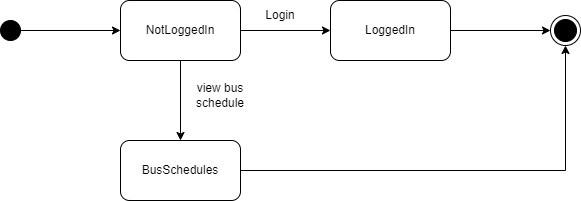


Figure 32

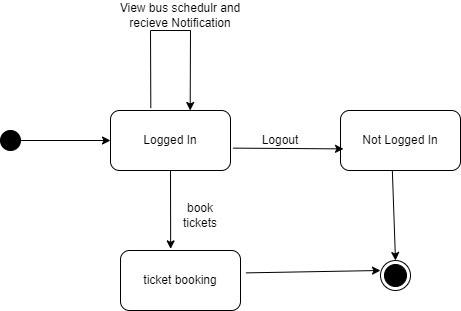
#### 5.4.1 Not Logged In State Diagram

Figure 33



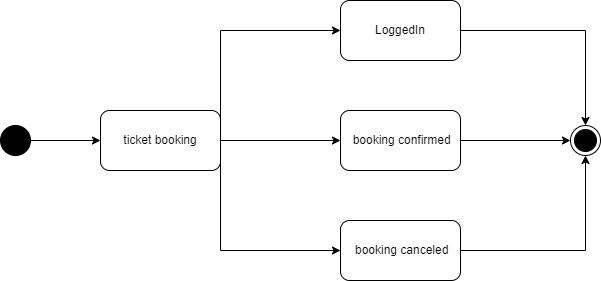
5.4.2 Logged In State Diagram

#### Figure 34



#### 5.4.3 Ticket Booking State Diagram

Figure 35



#### 5.4.4 Booking Confirmed State Diagram

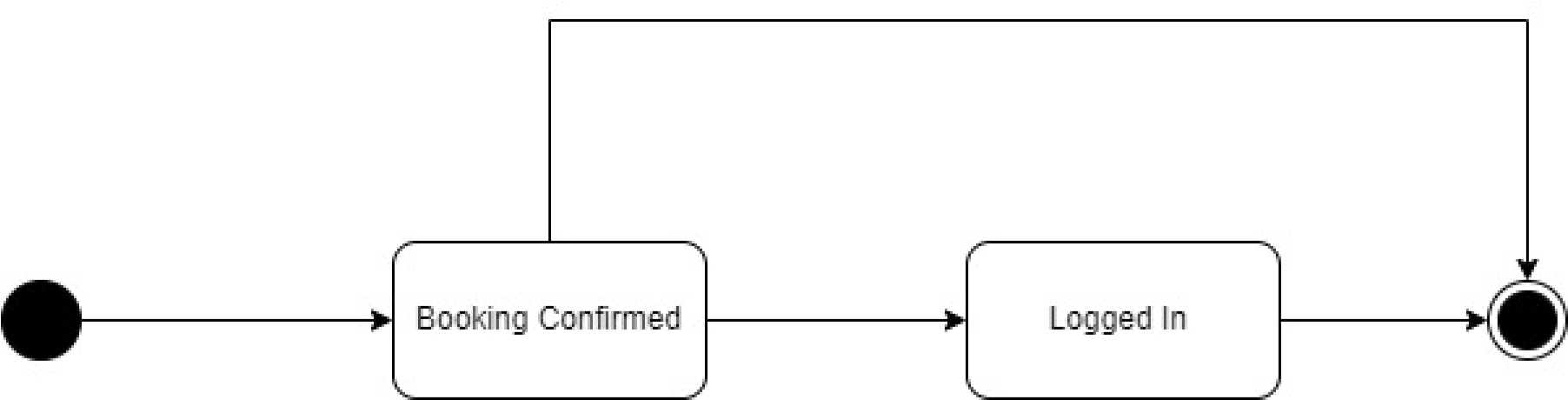


Figure 36

#### 5.4.5 Booking Canceled State Diagram

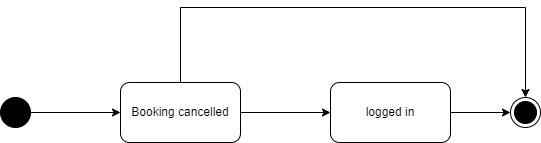


Figure 37

### 5.5 Sequence Diagrams

# m

###### Figure 38

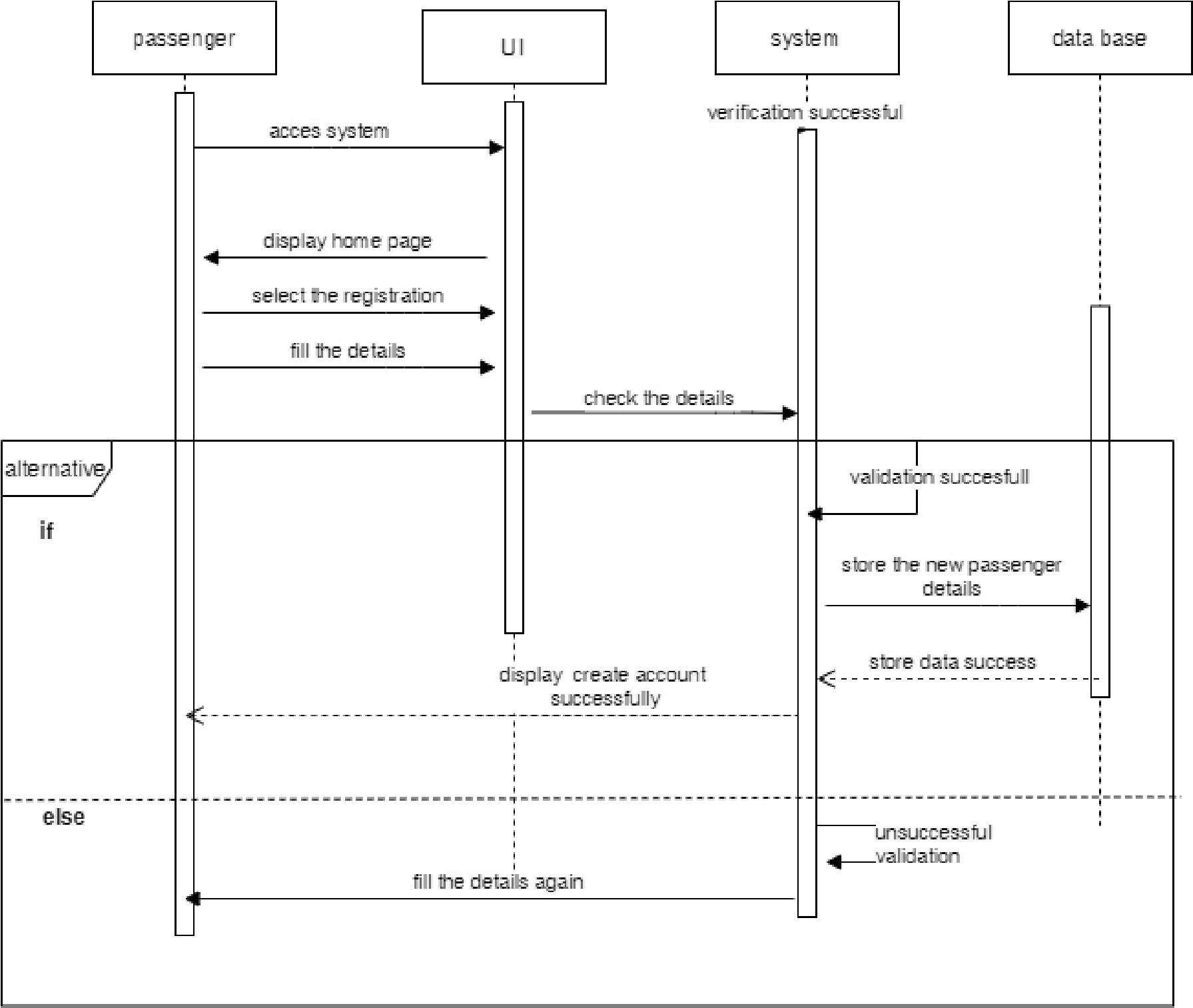


Figure 39

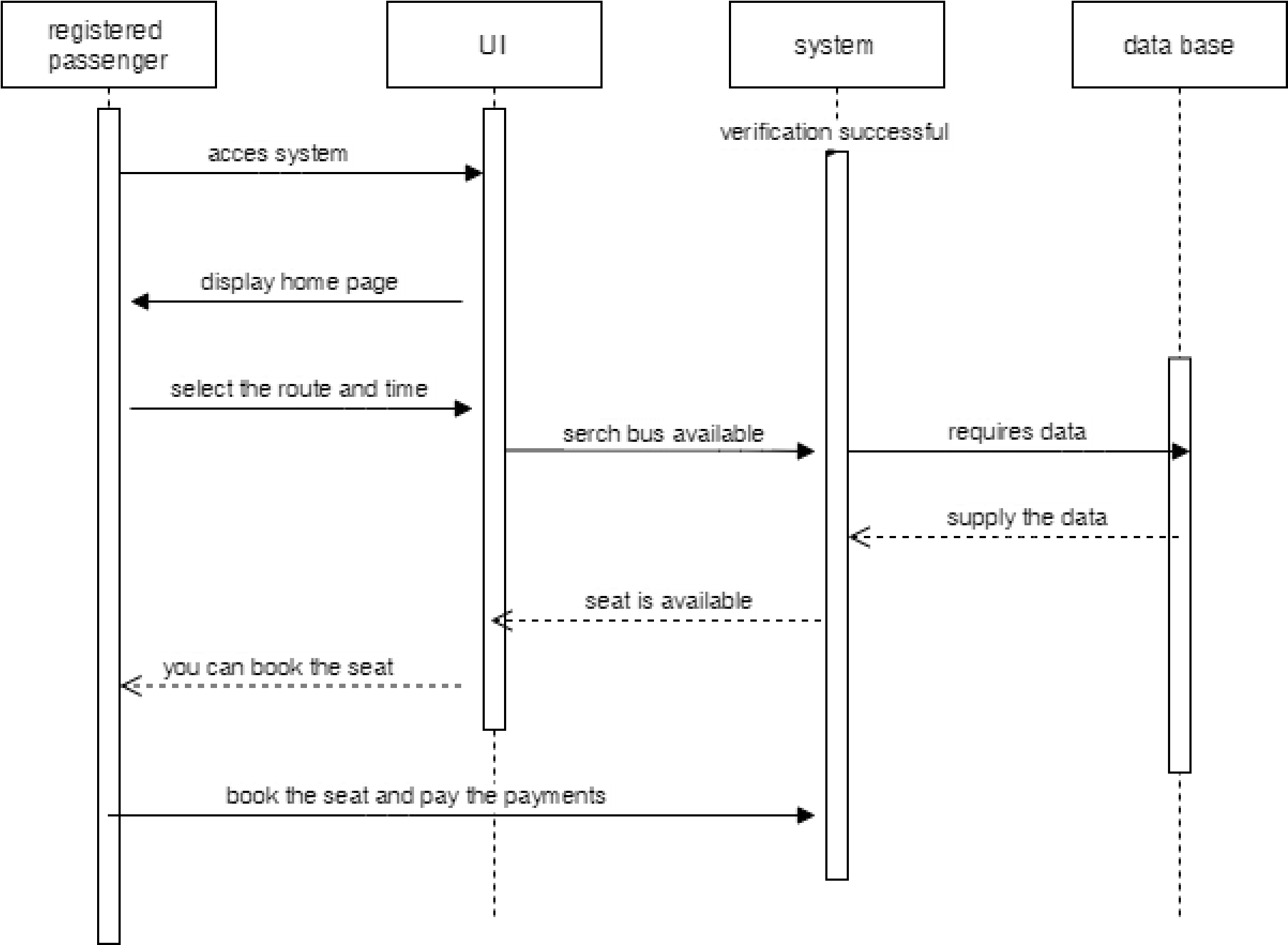


Figure 40

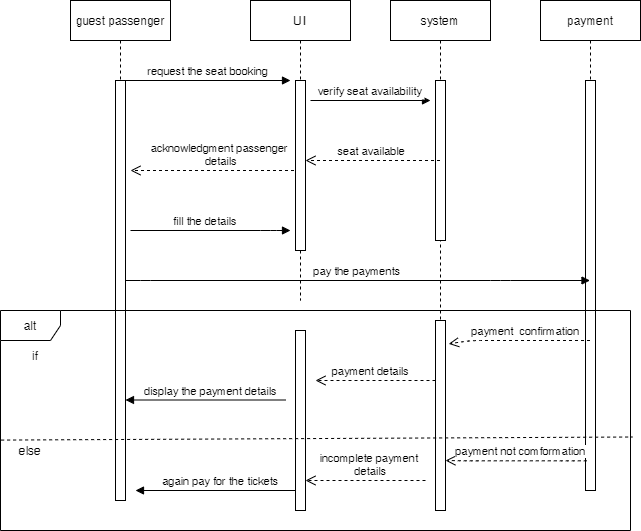


Figure 41

## 6 DESIGNED UI FLOW

##### 6.1 Wireframes

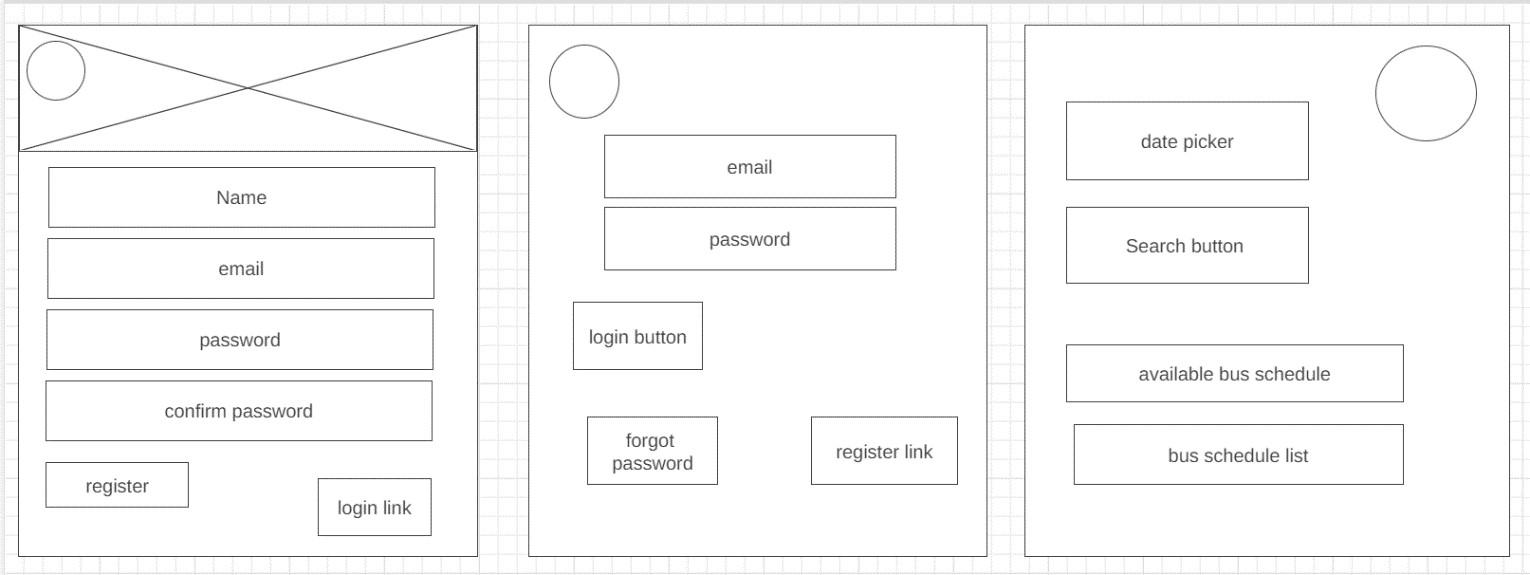
Figure 42 

Figure 43

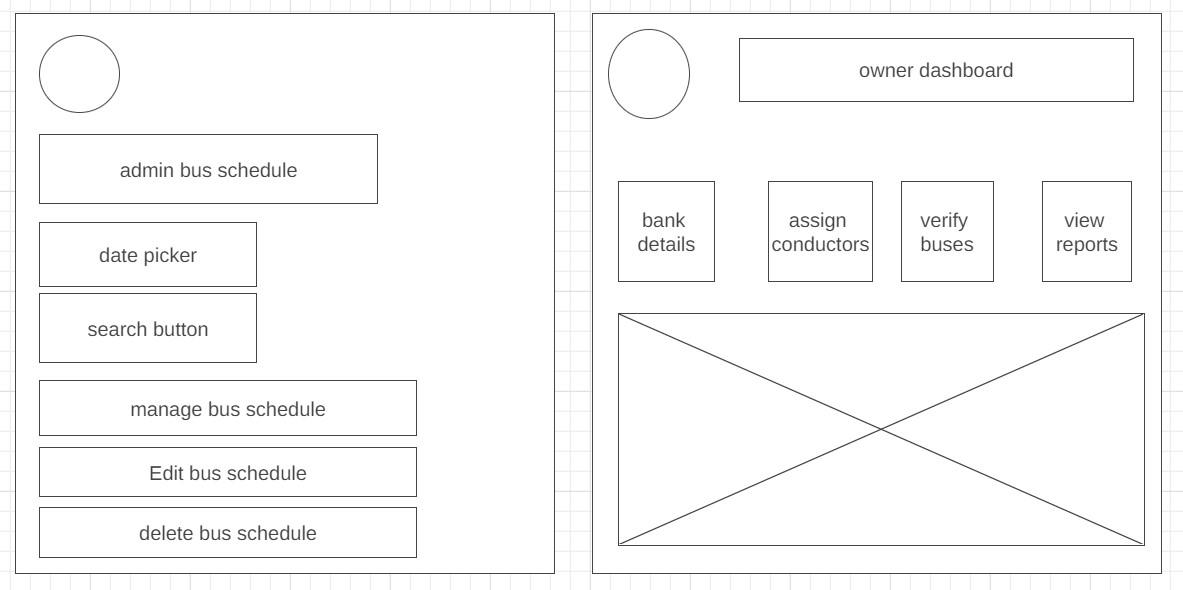


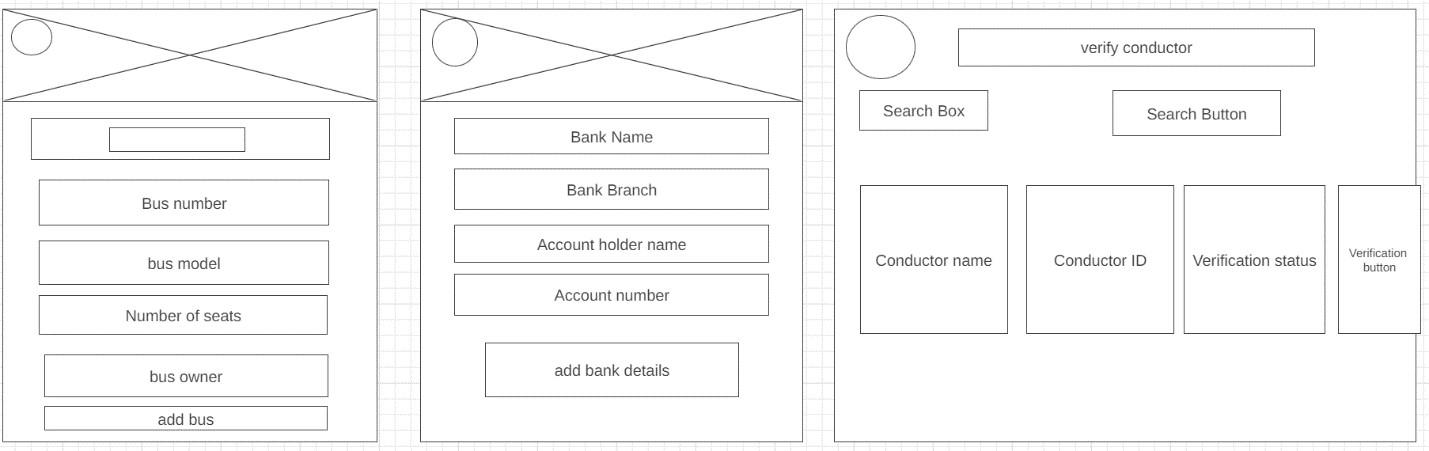
Figure 44 

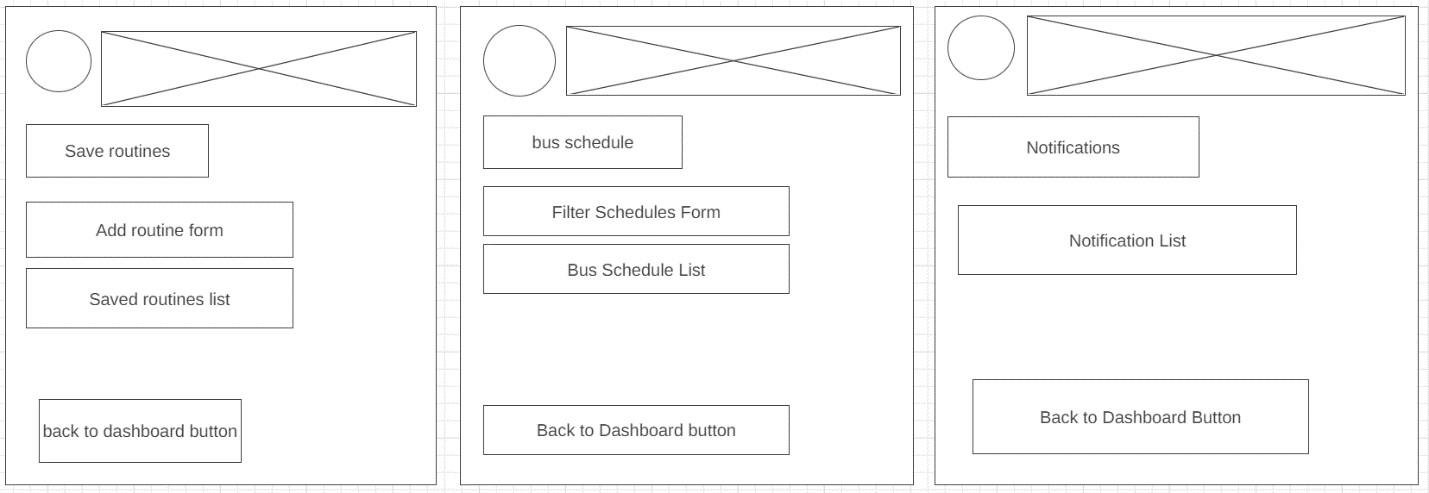
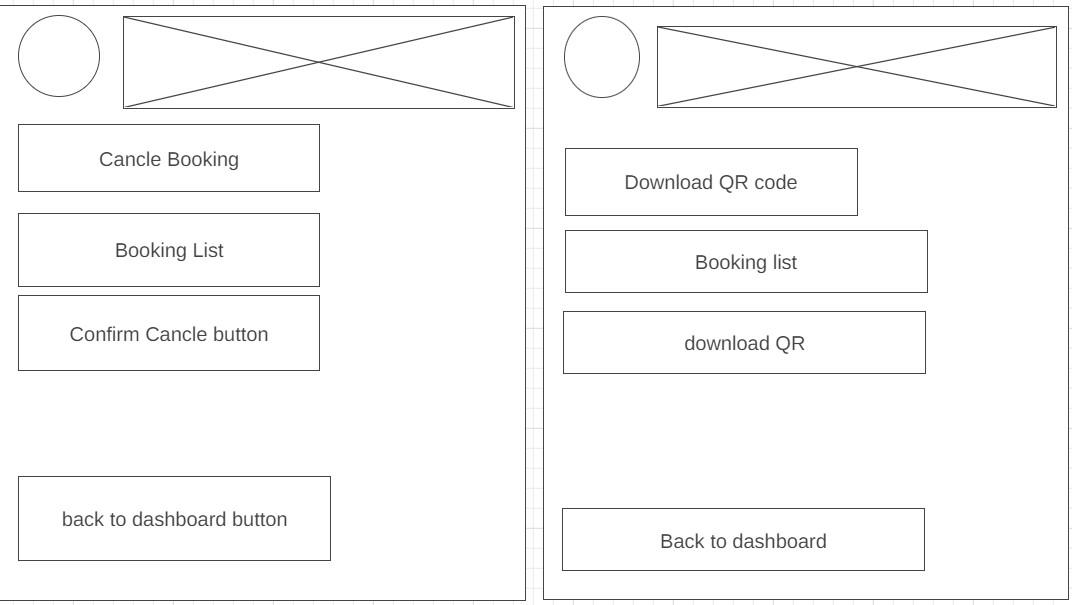
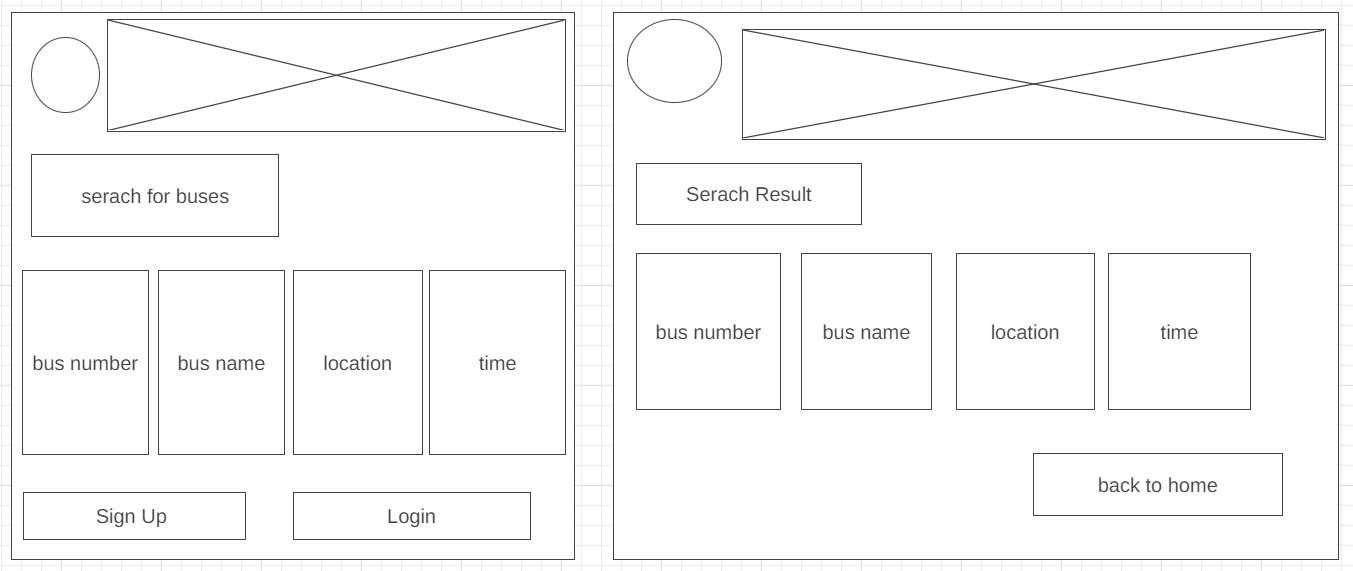
Figure 45 Figure 46 

Figure 47 

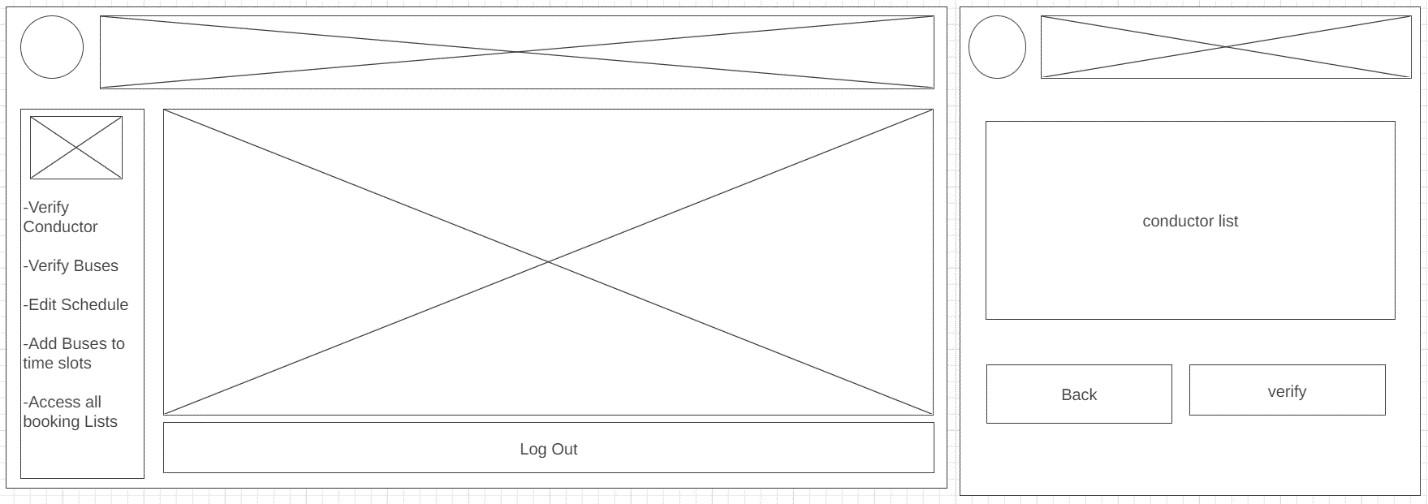
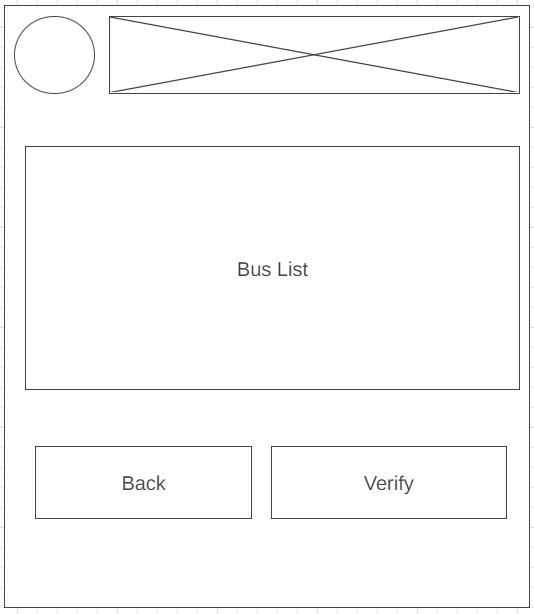
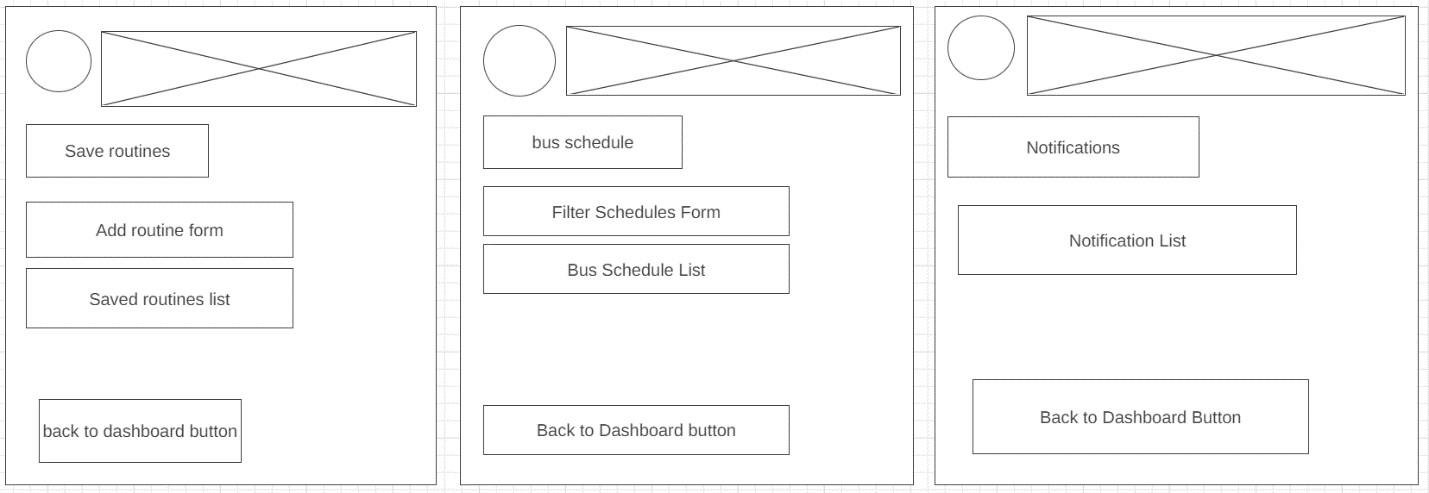


Figure 48

Figure 49 Figure 50 

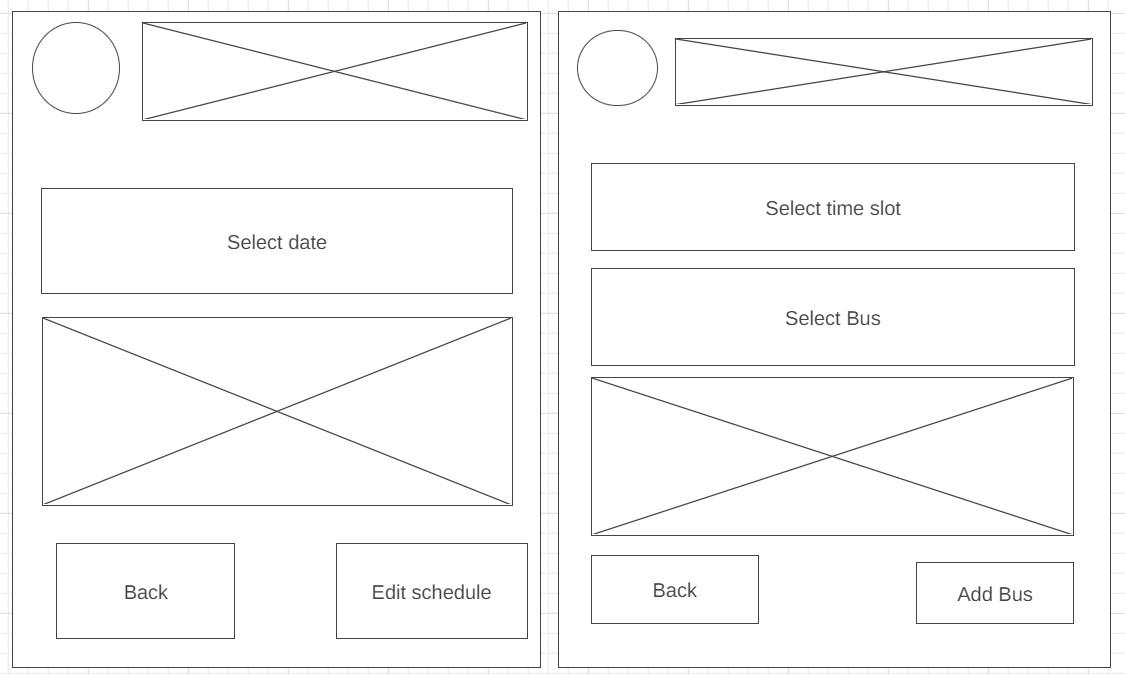


Figure 51

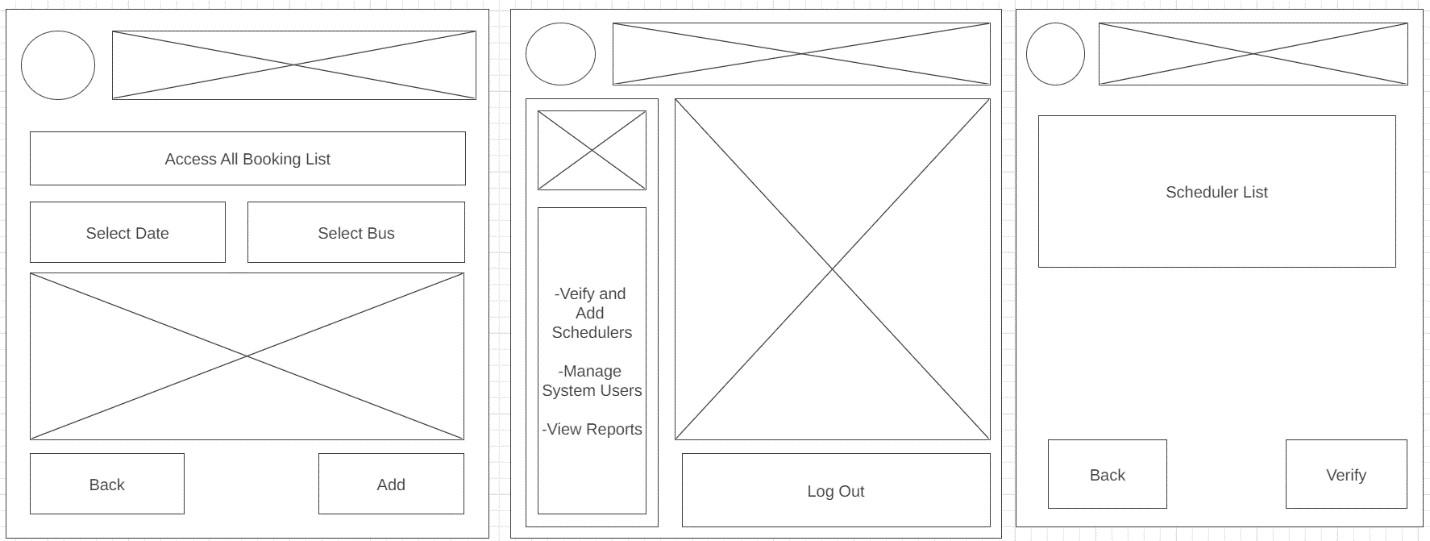


Figure 52

###### Figure 53

# 7. Improvements

1.Schedule Automation using mySQL events.

2.Bus Rotation Automation using mySQL events.

3.Reduced data entering work flows of scheduler and admin as much as possible by automations and access handling

# 8.Changes

1.Removed the Seat Selecting ability for the booking that are not within today now the seat selecting is only available for bookings that are within today itself.

2.Removed ability to edit ability to the generated schedule for schedulers.

3.Removed ability to edit bus rotation manually for schedulers.

1. Test Cases

**Login Page**

| Scenario  TID | Scenario Description | Test  Case ID | Pre-Condition | Steps to Execute | Excepted  Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Login Scenario with valid Credentials | 1 | Should be in the login page with valid credentials | 1.Enter user email  “[passenger@gmail.com](mailto:passenger@gmail.com)”  & password  “Passenger123”  2.click on the the login button | Login should work with valid credentials | Login is working with valid credentials | Pass  [login.png](https://drive.google.com/file/d/1JI61t9hzgCodLLUXavq9DABdHmIK4_pb/view?usp=drive_link) |
|  | Login Scenario with invalid Credentials | 2 | Seamless Login page for Registered passenger | 1.navigate to the seamless website  2.Enter user email  “[passenger123@gmail.com](mailto:passenger123@gmail.com)”  & password  “Passenger123”  3.click on the the login button | Login should be not work with invalid credentials | Login is not working with invalid credentials | Pass |

**Sign up**

| Scenario  TID | Scenario Description | Test  Case ID | Pre-Condition | Steps to Execute | Excepted  Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | Signing up with valid Credentials | 1 | signup->select user type  Signup page is loaded | 1.Fill the input fields with valid values  2.click on submit | Redirected to the login page with success message | Redirected to the login page with success message | Pass  [sign up.png](https://drive.google.com/file/d/1z4rzZ3rxXUH68EHDpmrKY4z4MIAEZNml/view?usp=drive_link) |
|  | Signing up with invalid Credentials or empty fields | 2 | Signup page is loaded | 1.Fill the input fields with invalid values or keep few of them empty  2.click on submit | validation failures will be displayed | validation failures will be displayed | Pass |

**Bus Schedule & Booking For Guest Passenger**

| **ScenarioTID** | **Scenario Description** | **TestCase ID** | **Pre-Condition** | **Steps to Execute** | **Excepted Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3.1 | Guest Passenger can view and filter bus schedules | 1 | Schedules page is visited by clicking on Bus Schedules | 1. Enter From, to, date and route 2. Click on Search button | Updated details of bus schedules, filtered accordingly appears | Updated details of bus schedules, filtered accordingly appears | Pass  [Guest Passenger 1](https://drive.google.com/drive/folders/1n4WGPsrXor-98Yv5ARzaGeG5Raox09-m?usp=drive_link) |
| 3.2 | Guest passenger booking a seat for a bus departing today | 2 | date selected as the current day of the week | 1. Enter NIC, Name, Phone. 2. Select a route and click on Book | Takes user to a page that allows to select seats for booking | Takes user to a page that allows to select seats for booking | Pass  [Guest Passenger 2](https://drive.google.com/drive/folders/1Itw_w2Qsq-rbZKkMk0u9fwY_YgpD_fbc?usp=drive_link) |
| 3.3 | Guest passenger booking a seat for a bus departing on a day that is not today | 3 | date selected is not the current day of the week | 1. Enter NIC, Name, Phone. 2. Select a route and click on Book | Takes user to a page that allows to input number of seats for booking | Takes user to a page that allows to input number of seats for booking | Pass  [Guest Passenger 3](https://drive.google.com/drive/folders/1YkFAlwt_nYhS4csCEhtA6x-7VGvm6mcZ?usp=drive_link) |
| 3.4 | Guest passenger selecting going into checkout | 4 | Passenger is able to view selected schedule details | 1. select the seats or the number of seats and click on checkout | Takes the passenger to the payment gateway | Takes the passenger to the payment gateway | Pass  [Guest Passenger 4](https://drive.google.com/drive/folders/1VxeAj561eTqYpJb_tfz39uKezCttpWU_?usp=drive_link) |

**Bus Schedule & Booking For Registered Passenger**

| **ScenarioTID** | **Scenario Description** | **TestCase ID** | **Pre-Condition** | **Steps to Execute** | **Excepted Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4.1 | Registered Passenger can view and filter bus schedules | 1 | Schedules page is visited by clicking on Bus Schedules | 1. Enter From, to, date and route 2. Click on Search button | Updated details of bus schedules, filtered accordingly appears | Updated details of bus schedules, filtered accordingly appears | Pass [Registered Passenger 1](https://drive.google.com/drive/folders/1LjXikoO1X9dl61C19AHe6yLMf077kL5W?usp=drive_link) |
| 4.2 | Registered passenger booking a seat for a bus departing today | 2 | date selected as the current day of the week | 1. Select a route and click on Book | Takes user to a page that allows to select seats for booking | Takes user to a page that allows to select seats for booking | Pass  [Registered Passenger 2](https://drive.google.com/drive/folders/1FhnLgR6rlXZ4Wy2E94MquFqQVDQQVf_a?usp=drive_link) |
| 4.3 | Registered passenger booking a seat for a bus departing tomorrow | 3 | date selected is not the current day of the week | 1. Select a route and click on Book | Takes user to a page that allows to input number of seats for booking | Takes user to a page that allows to input number of seats for booking | Pass  [Registered Passenger 3](https://drive.google.com/drive/folders/1bpS6z4KUB1nISw9ZcE1fWeTVCraqv7bk?usp=drive_link) |
| 4.4 | Registered passenger going into checkout | 4 | Passenger is able to view selected schedule details | 1. select the seats or the number of seats and click on checkout | Takes the passenger to the payment gateway | Takes the passenger to the payment gateway | Pass  [Registered Passenger 4](https://drive.google.com/drive/folders/1sbbCU2sJiBU7fSJAGPfqKXtxsmDxxM6U?usp=drive_link) |

| Scenario  TID | Scenario Description | Test  Case ID | Pre-Condition | Steps to Execute | Excepted  Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | (View bus schedules) get update the information about bus schedule | 1 | Bus schedule page is loaded | 1.we can put from, to, date and route what we want information about schedule  2.click on search button | Then you can see the in details about schedule what you want date as a table | Then you can see the in details about schedule what you want date as a table | Pass |
|  | (View bus schedules) get update the information about bus schedule | 2 | Bus schedule page is loaded | 1.we can put from, to, date and route what we want information about schedule  2.click on reset button | We can see the what previously we added data remove and you can repeat steps to execute | We can see the what previously we added data remove and you can repeat steps to execute | Pass |

**Admin’s Functionalities**

| Scenario  TID | Scenario Description | Test  Case ID | Pre-Condition | Steps to Execute | Excepted  Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | manage schedulers | 1 | Logged into the admin account | 1.click on manage schedulers | Then you can assign schedulers to the station or stations to the schedulers | Then you can assign schedulers to the station or stations to the schedulers | Pass  [Manage Schedulers.png](https://drive.google.com/file/d/1Lvr8ei3MjkklN2rSXaIK5goTn-gsWLNQ/view?usp=drive_link) |
|  | Add a new station to the system | 2 | Logged into the admin account | 1.click on dashboard’s add stations button and move to the add stations | list of currently available stations and option to delete also add new stations | list of currently available stations and option to delete also add new stations | Pass  [Add Stations.png](https://drive.google.com/file/d/1W4akXHpdCxZYDooWtgaiwN715JDpa_pj/view?usp=drive_link) |

**Registered Passenger other functionalities**

| Scenario  TID | Scenario Description | Test  Case ID | Pre-Condition | Steps to Execute | Excepted  Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | See My QR | 1 | Logged into a Registered Passenger account | 1.Click on dashboard’s My QR button | Your QR is displayed | Your QR is displayed | Pass [view QR.png](https://drive.google.com/file/d/1yFvQcqN66K4kbHEwlr5l9kza-jvS7kAL/view?usp=drive_link) |
|  | profile and bookings | 2 | Logged into a Registered Passenger account | 1.click on dashboard’s profile & Bookings button | Then you can move to the passenger’s profile details and you can edit these details  also can see the active and finished bookings there | Then you can move to the passenger’s profile details and you can edit these details  also can see the active and finished bookings there | Pass  [Passenger Profile.png](https://drive.google.com/file/d/11qTrbrAjqabhQWhpmnSKz-1UD6W3TA0u/view?usp=drive_link) |

**Owner’s Functionalities**

| Scenario  TID | Scenario Description | Test  Case ID | Pre-Condition | Steps to Execute | Excepted  Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | Select conductors for their buses | 1 | Select conductor page is loaded | Click on the select conductor and move to the their respective bus list  click on the select button in bus list  then click on the select button in conductor list | Then you can see the assign the conductor for that selected bus and the conductor status is already assigned | Then you can see the assign the conductor for that selected bus and the conductor status is already assigned | Pass |
|  | add buses | 2 | Add bus page is loaded | fill the form and click the register button | Then the request is sent to the scheduler | Then the request is sent to the scheduler | Pass |
|  | Add feedbacks | 3 | Add feedback page is loaded | fill the form and click on the save button | Add the feedback for the read feedback page (we can see the feedback in see report page) | Add the feedback for the read feedback page (we can see the feedback in see report page) | Pass |
|  | See reports | 4 | See Reports page is loaded | Click on dashboard’s see reports button | Then you can see the owners bus report  include performance matrix , bookings & ongoing buses and read the feedbacks | Then you can see the owners bus report  include performance matrix , bookings & ongoing buses and read the feedbacks | Pass |

**Scheduler’s Functionalities**

| Scenario  TID | Scenario Description | Test  Case ID | Pre-Condition | Steps to Execute | Excepted  Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | No action is possible until assigned to a station by an Admin | 1 | Logged into the system | Try to use any functionality on the scheduler dashboard | cannot do any action | cannot do any action | Pass  [scheduler-no access.png](https://drive.google.com/file/d/1n3riKQqYrHcqYciD4aiEkcvpD9MJhMZI/view?usp=drive_link) |
| 11 | Verify bus details | 2 | logged into the system and has been assigned in to a station | 1.click on Busses in the dashboard  2.go to Verify bus Details  3.Accept or Decline requests | When accepted bus is added to the relevant route  otherwise rejected | When accepted bus is added to the relevant route  otherwise rejected | Pass  [Verify busses.png](https://drive.google.com/file/d/1m5JaC4-XK2yGWSup9q1DnhT55ZF_6VOz/view?usp=drive_link) |
| 12 | access currently accepted buses | 3 | logged into the system and has been assigned in to a station | 1.click on Busses in the dashboard  2.go to See bus Details  3.Do any action as u need | You can see the accepted buses list in table and could take any action(remove of pause) | You can see the accepted buses list in table and could take any action(remove of pause) | Pass  [accepted buses.png](https://drive.google.com/file/d/1TAsI8z8NQznXhgYoRkWaIOG2NXDih4Pr/view?usp=drive_link) |
| 13 | Manage Schedule | 4 | logged into the system and has been assigned in to a station | 1.click on the manage schedule in the dashboard | an interface to define the schedule of the relevant routes to the station that you are assigned | an interface to define the schedule of the relevant routes to the station that you are assigned | Pass  [Manage Schedule](https://drive.google.com/drive/folders/1O01L2WUGVrjhrBwmN27ETNCHVGltmzCZ?usp=drive_link) |
| 14 | Add a new route | 5 | logged into the system and has been assigned in to a station | 1.click on the routes in the dashboard | you will get a table of currently available routes in your station with a ability to edit ticket price or delete the route  and a form below to add new routes | you will get a table of currently available routes in your station with a ability to edit ticket price or delete the route  and a form below to add new routes | pass  [add new route.png](https://drive.google.com/file/d/1JMJ7stChfSVk-wuZ75C9j-xSAVJB3iUu/view?usp=drive_link) |

10 Individual Contribution

**Student Index Number: 21000379**

**Student Name: D. W. DE HOEDT**

**Group Number: CS23**

**Group Name: CS group 23**

**● Components you have implemented.**

**1. Administrator**

**2. Bus Schedule**

**● Description of the component(s):**

**1. CRUD operations to manage Schedulers in the system, assign Schedulers to stations, Add and Delete Stations, View schedulers and stations in the system and a Dashboard to ease access to functionalities.**

**2. Two Read operations to filter and view bus schedules**

**Student Index Number: 21001962**

**Student Name: U. T. Thushani**

**Group Number: CS23**

**Group Name: CS group 23**

**● Components you have implemented.**

**1. Add routes**

**2. Verify Bus details**

**3. See bus details**

**● Description of the component(s):**

**1. Adding routes relevant to the station that the scheduler is assigned to**

**2. Accept or Decline requests from owners to add there buses to a rotation that relevant to the scheduler**

**3. see accepted buses and pause or remove buses from a station**

**Student Index Number: 21001715**

**Student Name: K. L. Sandaruwan**

**Group Number: CS23**

**Group Name: CS group 23**

**● Components you have implemented.**

**1. Add Bookings**

**2. Registered Passenger Profile**

**3. Registered Passenger QR**

**● Description of the component(s):**

**1. add a booking to the system**

**2. Profile Page of the registered Passenger**

**3. QR code of the registered passenger**

**Student Index Number: 21001545**

**Student Name: T. H. Rajapaksha**

**Group Number: CS23**

**Group Name: CS group 23**

**● Components you have implemented.**

**1. Sign up and logins**

**2. add buses**

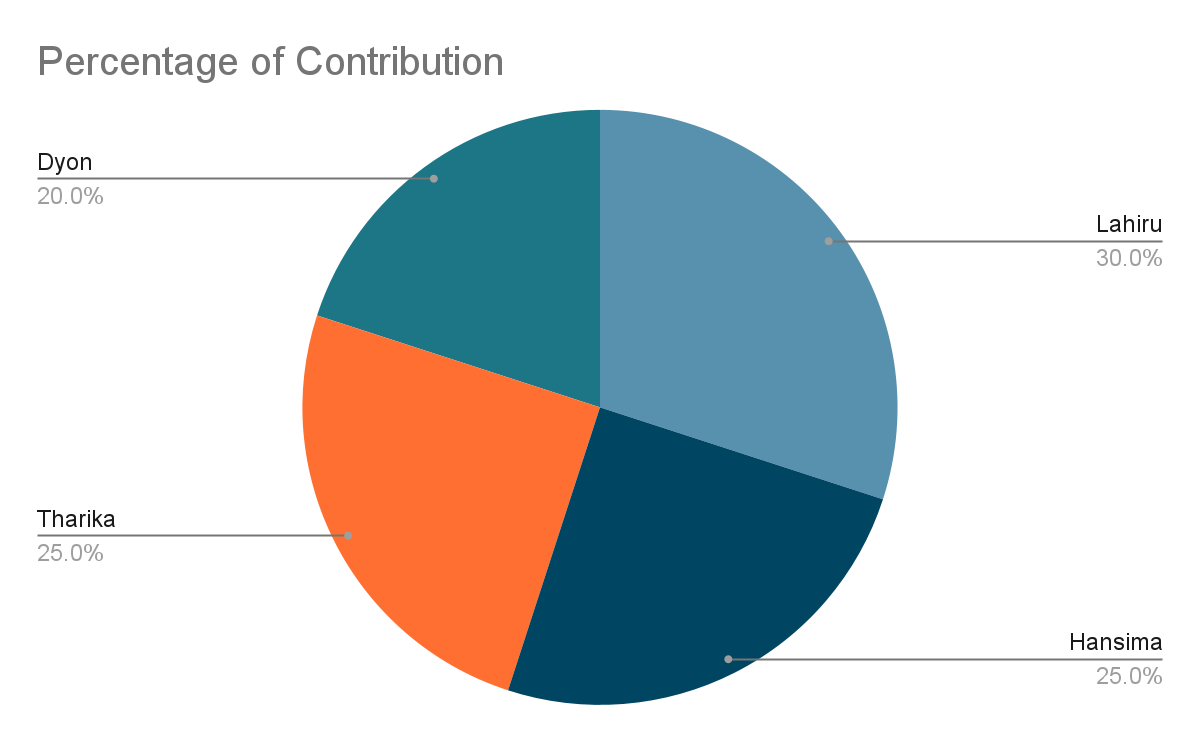
**3. Dashboard of the owner**

**● Description of the component(s):**

**1. Signup for the bus owner and the login**

**2. Owners can add Buses to the system.with the ability to request a route**

**3. In the dashboard all the options of the owner is shown and in the table below there are options to manage buses**



13 Declarations

We, the undersigned, acknowledge and approve the contents of this Software Requirements Specification (SRS) document for the "Seamless Bus" system. We have reviewed the outlined goals, objectives, scope, feasibility, and requirements, and hereby provide our endorsement of the SRS.

| NAME | DESCRIPTION | SIGNATURE | DATE |
| --- | --- | --- | --- |
| K.L.SANDARUWAN |  |  | 29/04/2024 |
| DYON DE HOEDT |  |  | 29/04/2024 |
| THARIKA THUSHANI |  |  | 29/04/2024 |
| HANSIMA  RAJAPAKSHA |  |  | 29/04/2024 |