Software Requirements and Design Document

for

Efficio

{Bus Management System}

Prepared By

Muhammad Iqrash Qureshi (22i-1174)

Hammad Shabbir (22i-1140)

Daniyal Khawar (22i-1066)

Department of Computer Science

FAST, National University of Computer and Emerging Sciences

Date: 19th November, 2024

Table of Contents

1. Introduction **1**
1.1 Purpose 1
1.2 Product Scope 1
1.3 Title 1
1.4 Objectives 1
1.5 Problem Statement 1
2. Overall Description **2**
2.1 Product Perspective 2
2.2 Product Functions 2
2.3 List of Use Cases 2
2.4 Extended Use Cases 3
2.5 Use Case Diagram 14
3. Other Nonfunctional Requirements **16**
3.1 Performance Requirements 16
3.2 Safety Requirements 16
3.3 Security Requirements 16
3.4 Software Quality Attributes 16
3.5 Business Rules 17
3.6 Operating Environment 17
3.7 User Interface 16
4. Domain Model **17**
5. System Sequence Diagram **18**
6. Sequence Diagram **29**
7. Class Diagram **41**
8. Package Diagram **42**
9. Component Diagram **43**
10. Deployment Diagram **44**

1 Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the software requirements for the Efficio Bus Management System App. Efficio is designed to streamline and enhance the management of bus operations, including ticket bookings, schedule management, and customer feedback. This document covers the complete scope of Efficio's functionality, focusing on passenger-facing features, administrative controls, and operator-specific tools.

1.2 Product Scope

Efficio is a desktop-based bus management system that focuses on providing services for passengers who want to travel from city to city. This application will manage bus scheduling, tickets bookings, cancellations and refunds. It provides a unified and automated solution, addressing the pain points of manual systems. Daewoo express (mobile based application) works in a similar fashion. People travelling from one city to another can benefit from Efficio.

1.3 Title

Efficio: Bus Scheduling, Ticketing and Management solution

1.4 Objectives

- Automating the ticket booking (pre-booking) process for passengers.
- Providing a platform for bus scheduling, cancellations, and refunds.
- Enabling users to post and track complaints, ensuring customer feedback is addressed efficiently.

1.5 Problem Statement

This problem is	that current intercity travelling through bus often relies on outdated and manual booking systems.
This affects	passengers who require a reliable and easy to use system for booking and managing bus travels.
This leads to	inconvenience such as long waiting times and inability to get tickets at the last moment. Moreover, there are troublesome processes for cancellations and refunds. All of this ultimately leads to sufferings for both the passengers and bus operators.
A successful solution would be	offering a bus management system that provides real time access to bus scheduling, allowing tickets booking and cancellations. In addition to this, Efficio offers various ticket categories, including Premium, First Class and Business Class, allowing passengers to choose from according to one’s convenience. The app is committed to simplifying process for passengers and improving efficiency by automating the system.

2 Overall Description

2.1 Product Perspective

Efficio is a new, self-contained product designed to address the gaps in the current market for bus management systems. Unlike the limited options available, such as Daewoo's app, which primarily caters to a single bus service provider with basic functionality, Efficio offers a more versatile and comprehensive solution. Our application supports multiple operators, provides enhanced features such as detailed schedule management, advanced booking options, and an integrated review system to improve customer feedback handling. While Daewoo's app focuses on a specific user base and predefined routes, Efficio is designed to cater to diverse operators and broader route networks, making it a scalable solution for varied user needs. By integrating robust administrative tools and an intuitive passenger interface, Efficio establishes itself as an innovative and much-needed improvement in the bus management system domain.

2.2 Product Functions

Following is some of the key features:

- Bus Schedule Management:
  Update bus schedules (for administrators).
  Search for buses based on departure and arrival locations.
- Ticket Booking:
  Search available seats for selected routes.
  Book tickets (for passengers).
  Cancel or modify bookings (within specified time frames).
  Process payments for bookings and refund in case of cancellation.
- Feedback Management:
  Allow passengers to submit feedback on bus services.
  View and manage feedback (for station manager and admin).
- Administrative Functions:
  Manage bus schedules, routes, and stations.

2.3 List of Use Cases

- Create an Account
- Book Tickets
- Manage Bus
- Refund Ticket
- Check Schedule
- Cancel Booking
- Give Feedback
- Manage Route
- Lodge Complaint
- Add Station
- Review Complaint

2.4 Extended Use Cases

- **1**

<table>
<tr><td>Use Case Name</td><td>Create an Account</td></tr>
<tr><td>Scope of the System</td><td>Efficio</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Passenger</td></tr>
<tr><td>Stakeholders and Interests</td><td>Passenger: Wants to use the bus services by creating an account.
Efficio Administration: Wants to make sure that the passenger registers with correct information.</td></tr>
<tr><td>Preconditions</td><td>User/Passenger has access to the Efficio app.
User/Passenger does not have an account already.</td></tr>
<tr><td>Postconditions</td><td>An account for a new passenger is successfully created.
User’s credentials are verified and stored in the database.
User is redirected to the home page.
Efficio Admin now allows the user to book a ticket.</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. User opens the application and selects “Create an Account” option.</td><td>2. System prompts the user to fill in requires fields.</td></tr>
<tr><td>3. User submits the form.</td><td>4. System validates the input.</td></tr>
<tr><td></td><td>5. System creates the user account.</td></tr>
<tr><td>6. User’s account is created
and is redirected to login page.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>3a. If any field is missing, the systems indicate an error and says the user to fill it.
4a. Because of some invalid input, the account is not created.
5a. Due to internal error, the account is not created and a failure message is displayed.
6a. Failed to be redirected to login/homepage.</td></tr>
</table>

- **2**

Use Case Name	Book Tickets

<table>
<tr><td>Scope of the System</td><td>Efficio</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Passenger</td></tr>
<tr><td>Stakeholders and Interests</td><td>Passenger: Wants a reserved seat for their journey.
Efficio Admin: Needs to ensure smooth operations, seat availability, and customer satisfaction at the station.</td></tr>
<tr><td>Preconditions</td><td>Passenger must be logged into Efficio.
Bus schedules and seat availability are up to date.</td></tr>
<tr><td>Postconditions</td><td>The seat is reserved, and a booking confirmation is generated.
Seat availability is updated.</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Passenger selects 'Book Tickets' from the menu.</td><td>2. System displays available booking options (routes, buses, schedules).</td></tr>
<tr><td>3. Passenger selects the desired bus route, date, time, and seat.</td><td>4. System checks the availability of scheduled seats.</td></tr>
<tr><td>5. Passenger reviews and confirms the booking.</td><td>6. System displays booking details (fare, seat number, bus info), processes the booking, and requests payment.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>4a. No seats available. The system notifies the passenger that the selected seat is unavailable and suggests alternative buses or routes.
5a. Passenger is not logged in. System prompts the user to first login to confirm the booking.</td></tr>
</table>

- **3**

Use Case Name	Manage Bus
Scope of the System	Efficio
Level	Admin Goal
Primary Actor	Admin
Stakeholders and Interests	**Admin:** Wants to efficiently manage buses by adding or deleting them based on requirements. **Station Manager:** Needs up-to-date information for management for buses.

<table>
<tr><td></td><td>Passengers: Rely on accurate bus information to book tickets and plan their travel.
Government/Regulators: Ensures that the system compiles with local transportation regulations and safety.</td></tr>
<tr><td>Preconditions</td><td>The admin is logged into the system with appropriate permissions.
The system/admin has access to bus data.</td></tr>
<tr><td>Postconditions</td><td>The system updates the information about buses according to update/change made.
New buses added or existing buses are deleted.
A ticket can be booked by the passengers according to revised schedule.</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Admin navigates to the “Manage Bus” option.</td><td>2. System presents admin with option to add new bus or delete already existing bus.</td></tr>
<tr><td>3.Admin chooses “Add Bus” or “Delete Bus”.</td><td></td></tr>
<tr><td>4. If adding, the admin inputs required bus details.
If deleting, the admin selects the bus from the list of existing buses.</td><td>5. System shows success message to the admin and makes changes in database.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>4a. The details provided are invalid in case of adding bus.
4b. The bus identity told by admin is not correct in case of deleting bus.
5a. If bus is currently travelling, prevent bus deletion.
5b. If tickets for future travelling of the bus are booked, refund the passengers who already booked the tickets.</td></tr>
</table>

- **4**

Use Case Name	Refund Ticket
Scope of the System	Efficio
Level	Admin Goal (Station Manager)
Primary Actor	Station Manager
Stakeholders and Interests	**Passenger:** Wants to receive a refund for a cancelled or unused ticket.

<table>
<tr><td></td><td>Station Manager: Responsible for processing refunds and ensuring that it is performed accurately.
Admin: Ensures the refund process is smooth and all financial records are up to date.
Payment Gateway: Ensures that the refunded amount is sent correctly to desired passenger.</td></tr>
<tr><td>Preconditions</td><td>Passenger has a valid ticket eligible for refund.
Station manager is logged into the system with appropriate permissions.
The system has access to payment and booking details.</td></tr>
<tr><td>Postconditions</td><td>The ticket refund is successfully processed.
The refund is recorded in the system.
The system updates ticket and seat availability status.</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Station Manager navigates to the “Refund Ticket” option.</td><td>2.System displays all the passengers who wants refund of their ticket.</td></tr>
<tr><td>3. Station manager checks validity of the refund and checks payment method.</td><td></td></tr>
<tr><td>4. Station manager approves the refund.</td><td>5. System processes the refund by crediting the payment method.</td></tr>
<tr><td></td><td>6. The system updates the ticket/seat availability.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>2a. There are no passengers asking for refund.
3a. If the ticket is invalid for refund, the passenger is informed.
5a. The amount is not credited to the destined payment method.
6a. Due to some error, the status of the ticket is not updated.</td></tr>
</table>

- **5**

Use Case Name	Check Schedule
Scope of the System	Efficio
Level	User Goal
Primary Actor	Passenger
Stakeholders and Interests	**Passenger:** Wants to view available schedule options.

<table>
<tr><td></td><td>System Administrator: Ensures the system runs smoothly and maintains the schedule.
Support Staff: Aims to assist customers with their inquiries.</td></tr>
<tr><td>Preconditions</td><td>The Passenger must be authenticated and logged into the system. The schedule data must be up-to-date and accessible.</td></tr>
<tr><td>Postconditions</td><td>The Passenger successfully views the schedule.
The system may log the access for tracking purposes.</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. The Passenger logs into the system.</td><td>2. The system verifies credentials and grants access.</td></tr>
<tr><td>3. Passenger requests to view the available schedules.</td><td>4. The system retrieves the schedule data.</td></tr>
<tr><td>5. Passenger selects a schedule to view details.</td><td>6. The system displays the list of available schedule options.</td></tr>
<tr><td>7. The Passenger reviews the schedule options.</td><td>8. The system may provide options to book or inquire further.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1a. The Passenger attempts to log in with incorrect credentials.
2a. The system displays an error message indicating failed login.
3a. Passenger requests to view the schedule, but it’s not available.
4a. The system indicating schedule unavailability.
6a. System show incorrect information about scheduling options.
7a. Passenger doesn’t review the schedule options due to issues in customer system.</td></tr>
</table>

- **6**

Use Case Name	Cancel Booking
Scope of the System	Efficio
Level	User Goal

<table>
<tr><td>Primary Actor</td><td>Passenger</td></tr>
<tr><td>Stakeholders and Interests</td><td>Passenger: Wants to cancel purchased tickets and receive a refund if applicable.
Ticketing Staff: Ensures tickets are cancelled properly and updates are reflected in the system. System Administrator: Responsible for maintaining the integrity of ticketing data.</td></tr>
<tr><td>Preconditions</td><td>The Passenger must have purchased tickets.
The Passenger must be logged into the Efficio app.
The tickets must still be within the valid cancellation period (before the journey date or based on the refund policy).</td></tr>
<tr><td>Postconditions</td><td>The Passenger successfully cancels the tickets.
The ticket status is updated to "Canceled" in the system.
A refund is processed if applicable (based on the ticket type and cancellation policy).</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. The Passenger navigates to the "My Bookings" section.</td><td>2. The system displays the list of purchased tickets.</td></tr>
<tr><td>3. The Passenger selects the bookings to be cancelled.</td><td>4. The system confirms the selection of bookings for cancellation.</td></tr>
<tr><td>5. The Passenger submits the cancellation request.</td><td>6. The system updates the status of the booking to "Cancelled."</td></tr>
<tr><td>7. The Passenger reviews the refund policy (if applicable).</td><td>8. The system displays a confirmation message of successful cancellation and initiates the refund if necessary.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>3a. The system displays an error message indicating that the tickets are not eligible for cancellation or already cancelled.
4a. The system displays an error indicating technical issues or incorrect request format.
5a. The Passenger submits an incorrect cancellation request.
6a. The system fails to update the ticket status to "Cancelled."
8a. The system displays an error and suggests retrying the cancellation request.</td></tr>
</table>

- **7**

<table>
<tr><td>Use Case Name</td><td>Give Feedback</td></tr>
<tr><td>Scope of the System</td><td>Efficio</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Passenger</td></tr>
<tr><td>Stakeholders and Interests</td><td>Passenger: Wants to provide feedback on their experience.
Business Owner/Management: Seeks insights for service improvement.
Support Staff: Aims to address customer concerns based on feedback received.</td></tr>
<tr><td>Preconditions</td><td>The passenger must be logged into the app.
The feedback form must be accessible.</td></tr>
<tr><td>Postconditions</td><td>The passenger’s feedback is successfully submitted and stored in the system.
A confirmation message is displayed to the customer.</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1.The Passenger navigates to the feedback section.</td><td>2. The system displays the feedback form.</td></tr>
<tr><td>3. The Passenger fills out the feedback form and submits it.</td><td>4. The system validates the entered information and stores it.</td></tr>
<tr><td></td><td>5. The system displays a confirmation message and thanks for the submission.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>3a. The customer submits an incomplete or invalid form.
3b. The customer encounters an error during submission.
4a. The system displays error messages indicating the issues.
5a. The system provides options to retry or contact support.</td></tr>
</table>

- **8**

Use Case Name	Manage Route
Scope of the System	Efficio
Level	User Goal
Primary Actor	Admin
Stakeholders and Interests	**Admin**: Wants to efficiently add or delete routes. **Customers**: Expect reliable and updated route information.

<table>
<tr><td></td><td>System Administrator: Ensures that the system operates smoothly and accurately reflects the current routes.</td></tr>
<tr><td>Preconditions</td><td>The admin must be authenticated and logged into the system.
The route management interface must be accessible.</td></tr>
<tr><td>Postconditions</td><td>The system updates the route list based on the admin’s actions (either addition or deletion). Confirmation of the update is provided to the admin.</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. The admin navigates to the "Manage Routes" section.</td><td>2. The system displays the current list of routes.</td></tr>
<tr><td>3. The admin selects to add a new route or delete one.</td><td></td></tr>
<tr><td>4. If adding, the admin fills in the route details and submits.</td><td></td></tr>
<tr><td>5. If deleting, the admin selects a route to delete and confirms the action.</td><td>6. The system processes and confirms the route addition or deletion.</td></tr>
<tr><td></td><td>7. The system updates the route list and reflects the changes.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>2a. System doesn’t fetch correct list of routes to admin.
3a. The admin attempts to add a route or delete a route that does not exist.
4a,5a. Admin fills incorrect details that doesn’t support the system.
6a. The system displays an error message indicating the route is not found.
7a. Due to technical issues system doesn’t confirm a add or delete a route or updates it.</td></tr>
</table>

- **9**

Use Case Name	Lodge Complaint
Scope of the System	Efficio
Level	User Goal

<table>
<tr><td>Primary Actor</td><td>Passenger</td></tr>
<tr><td>Stakeholders and Interests</td><td>Passenger: Wants to lodge a complaint regarding an issue and expects a resolution.
Station Manager: Needs to address service or operational issues efficiently.
Admin: Needs to make sure complaints are solved timely by the managers and customer is satisfied.</td></tr>
<tr><td>Preconditions</td><td>The customer must be logged into Efficio.
A complaint form or system must be available.</td></tr>
<tr><td>Postconditions</td><td>The complaint is lodged, and the customer receives a confirmation with a reference number for tracking.</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Passenger opens the 'Lodge Complaint' section.</td><td>2. System displays a complaint form.</td></tr>
<tr><td>3. Passenger submits the complaint after entering details.</td><td>4. System notifies user after successful submission.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1a. Customer not logged in. System prompts the user to first login to lodge the complaint.
3a. Incomplete or invalid complaint details. System prompts the user to correct the form.
4a. System error during submission. System notifies the passenger of the error and suggests trying again or contacting support.</td></tr>
</table>

- **10**

Use Case Name	Add Station
Scope of the System	Efficio
Level	Admin Goal
Primary Actor	Admin
Stakeholders and Interests	**Owner**: Wants to expand the bus network by adding new stations, ensuring proper data is added to the system for future routes. **Customer**: Expects new stations to be reflected in the booking system for a smoother travel experience.
Preconditions	The admin must be logged in with admin privileges.
Postconditions	A new station is successfully added to the system and is available for route creation.
Main Success Scenario	

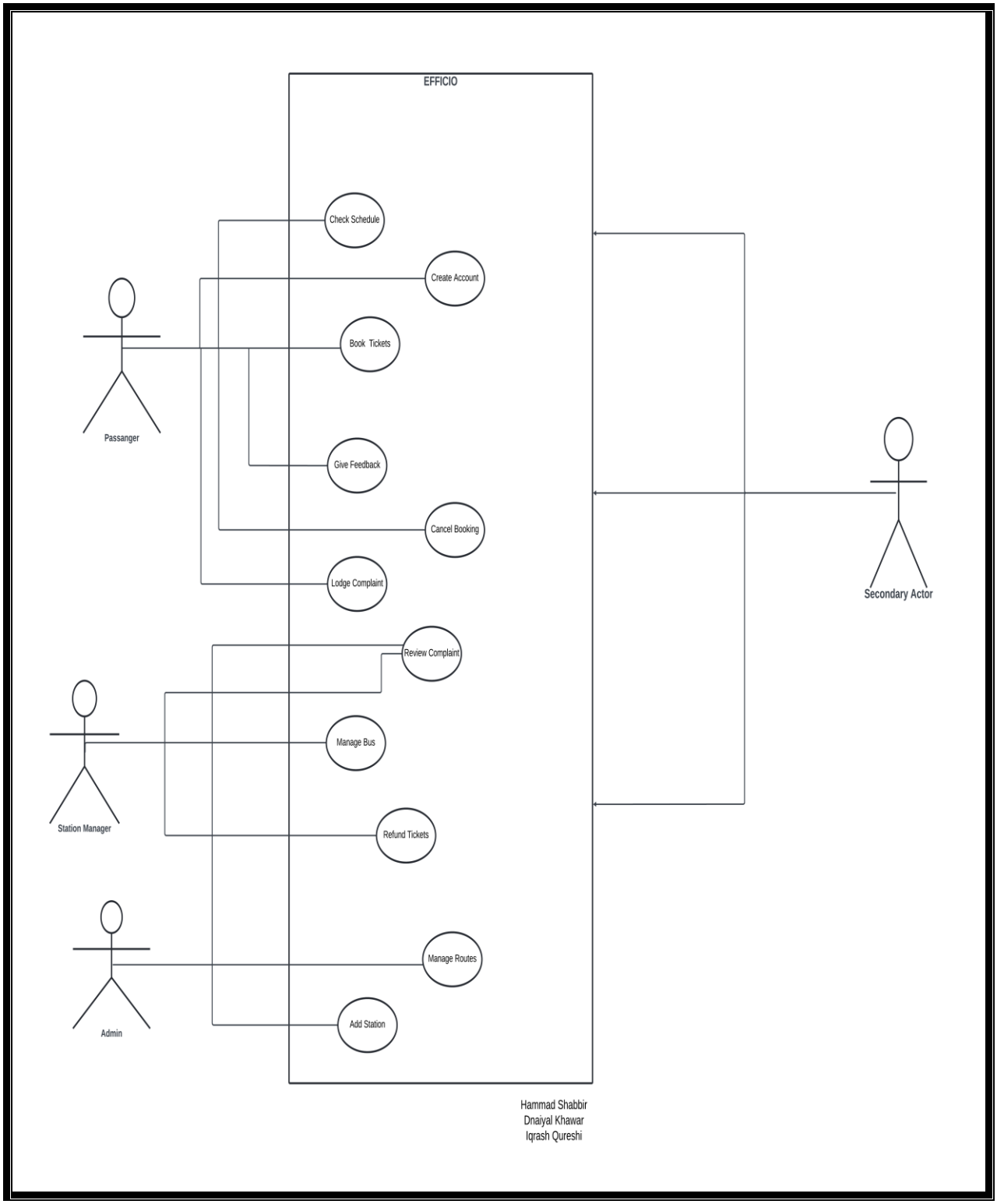
	<table><tr><th>Actor</th><th>System</th></tr><tr><td>1. Admin selects 'Add Station' from the admin dashboard.</td><td>2. System displays a form where the owner can input new station details</td></tr><tr><td>3. Admin enters station details.</td><td>4. The system validates the entered details to ensure all required fields are completed correctly.</td></tr></table>
Extensions	1a. Owner not logged in. System prompts the owner to first login to add station. 3a**.** Incomplete or invalid station details. System prompts the owner to correct the form. 4a. System error during submission. System notifies the passenger of the error and suggests trying again or contacting support.

- **11**

Use Case Name	Review Complaint
Scope of the System	Efficio
Level	Manager Goal
Primary Actor	Station Manager
Stakeholders and Interests	**Station Manager**: Needs to review and resolve customer complaints in a timely manner to maintain service quality. **Customer**: Expects that their complaint is addressed and resolved effectively. **Owner**: Wants to ensure that complaints are handled efficiently to maintain customer satisfaction and system reliability.
Preconditions	The station manager must be logged into Efficio with the necessary privileges. There must be at least one complaint lodged in the system.
Postconditions	The complaint is reviewed, and actions are taken or documented for resolution. The customer is notified about the review.

<table>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Station Manager navigates to the 'Review Complaints' section.</td><td>2. The system displays a list of lodged complaints.</td></tr>
<tr><td>3. Station Manager selects a complaint from a list to review.</td><td>4. The system retrieves and displays the full details of the selected complaint.</td></tr>
<tr><td>5. Station Manager marks the complaint as resolved or pending further action.</td><td>6. The system updates the complaint status based on the manager's input, and sends a confirmation notification to the customer regarding the resolution or the need for further follow-up.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1a. Station Manager not logged in. System redirects to the login page.
4a. Complaint details not available. System alerts the manager of an issue retrieving complaint data and suggests trying again later.
5a. No action taken. System prompts the station manager to log an action or response before proceeding.</td></tr>
</table>

2.5 Use Case Diagram



3 Other Non-Functional Requirements

3.1 Performance Requirements

The Efficio Bus Management System (BMS) is designed to ensure high performance and responsiveness across diverse operational scenarios, providing a seamless user experience. The system should process key actions such as managing buses/routes, ticket bookings, and schedule views within 2 seconds for the majority of user interactions under standard conditions. To handle peak usage periods efficiently, the application must support up to 1,000 concurrent users without any noticeable decline in performance. Critical processes, including ticket booking and payment transactions, should complete within 3 seconds to maintain efficiency and user trust. These performance benchmarks are essential to ensure smooth operation and uphold user satisfaction across all major functionalities of the system.

3.2 Safety Requirements

The Efficio Bus Management System (BMS) must incorporate strong safety measures to ensure the security of data and the reliability of operations. The system must enforce data validation and error-checking mechanisms to prevent inaccuracies or corruption during key operations, such as ticket bookings and payment processing. Automated daily backups must be conducted, with a recovery process capable of restoring all data within 30 minutes in the event of a system failure to minimize disruption. To safeguard against unauthorized access, sensitive functionalities like financial transactions, user management, and system configurations must be accessible only to authorized personnel with appropriate credentials. The BMS must also comply with relevant safety regulations and standards to ensure secure and ethical management of user data.

3.3 Security Requirements

Security and privacy are paramount for the Efficio Bus Management System (BMS). The system must enforce secure authentication mechanisms, such as strong passwords and OTP's, to prevent unauthorized access. Sensitive data, both at rest and in transit, must be encrypted using industry-standard encryption protocols, such as AES-256, to protect against data breaches and unauthorized access. Role-based access control (RBAC) must be implemented to ensure users can only access data and functions relevant to their roles, such as passengers, operators, or administrators. The system must comply with applicable security regulations and standards, ensuring the protection of user data and financial transactions.

3.4 Software Quality Attribute

The Efficio Bus Management System (BMS) must meet several key quality attributes to ensure high customer satisfaction and effective system management. It must be high reliable with an uptime of 99.9%, providing continuous availability even during peak usage. The user interface must be intuitive and user-friendly, making it easy for passengers, operators, and administrators to navigate and perform tasks efficiently. The application should support maintainability through a modular code structure and comprehensive documentation, allowing for seamless updates, bug fixes, and enhancements. It should also

be highly flexible, capable of adapting to new requirements or expanded user bases without extensive modifications, and must ensure interoperability with third-party systems like payment gateways. It should support testability, enabling efficient testing of individual components to ensure feature correctness, and demonstrate robustness by handling unexpected errors and maintaining system stability. Additionally, its features must be designed for reusability, facilitating future system extensions or feature additions. These specific and measurable attributes will ensure Efficio delivers a reliable, efficient, and adaptable solution for bus management.

3.5 Business Rules

The Efficio Bus Management System (BMS) must adhere to specific business rules to ensure proper operation and maintain consistency across all functions. These rules define which roles can perform certain actions under specific circumstances. For example, only authorized administrators should be able to manage bus schedules, update routes. Station managers are responsible for managing bookings, reviewing complaints, while passengers should only have access to booking tickets, viewing schedules, and providing feedback. Additionally, only authorized personnel should be able to modify system configurations. These business rules are essential for enforcing role-based access, ensuring operational integrity, and preventing unauthorized access or actions within the system.

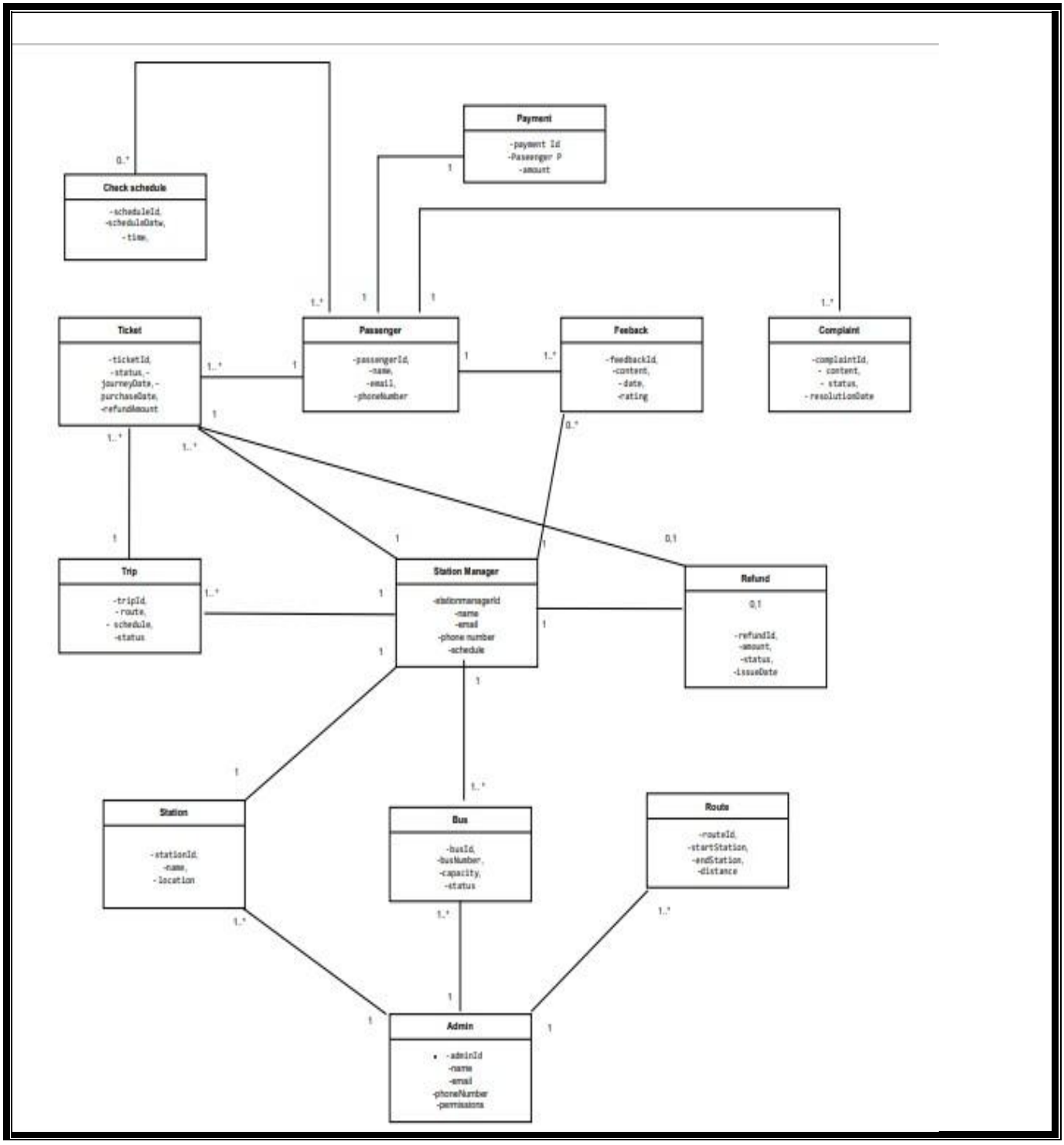
3.6 Operating Environment

The Efficio Bus Management System (BMS) must be designed to operate in a typical transportation environment, accommodating various hardware and software components. It should be compatible with standard devices such as desktop computers, and mobile devices used by both operators and passengers. The system must support major operating systems, including Windows, macOS, iOS, and Android, ensuring compatibility across a wide range of devices. Furthermore, the BMS must integrate seamlessly with other relevant software applications commonly used in transportation management, such as payment gateways and scheduling systems. This compatibility will ensure smooth integration and operation within the broader technological ecosystem of transit services.

3.7 User Interfaces

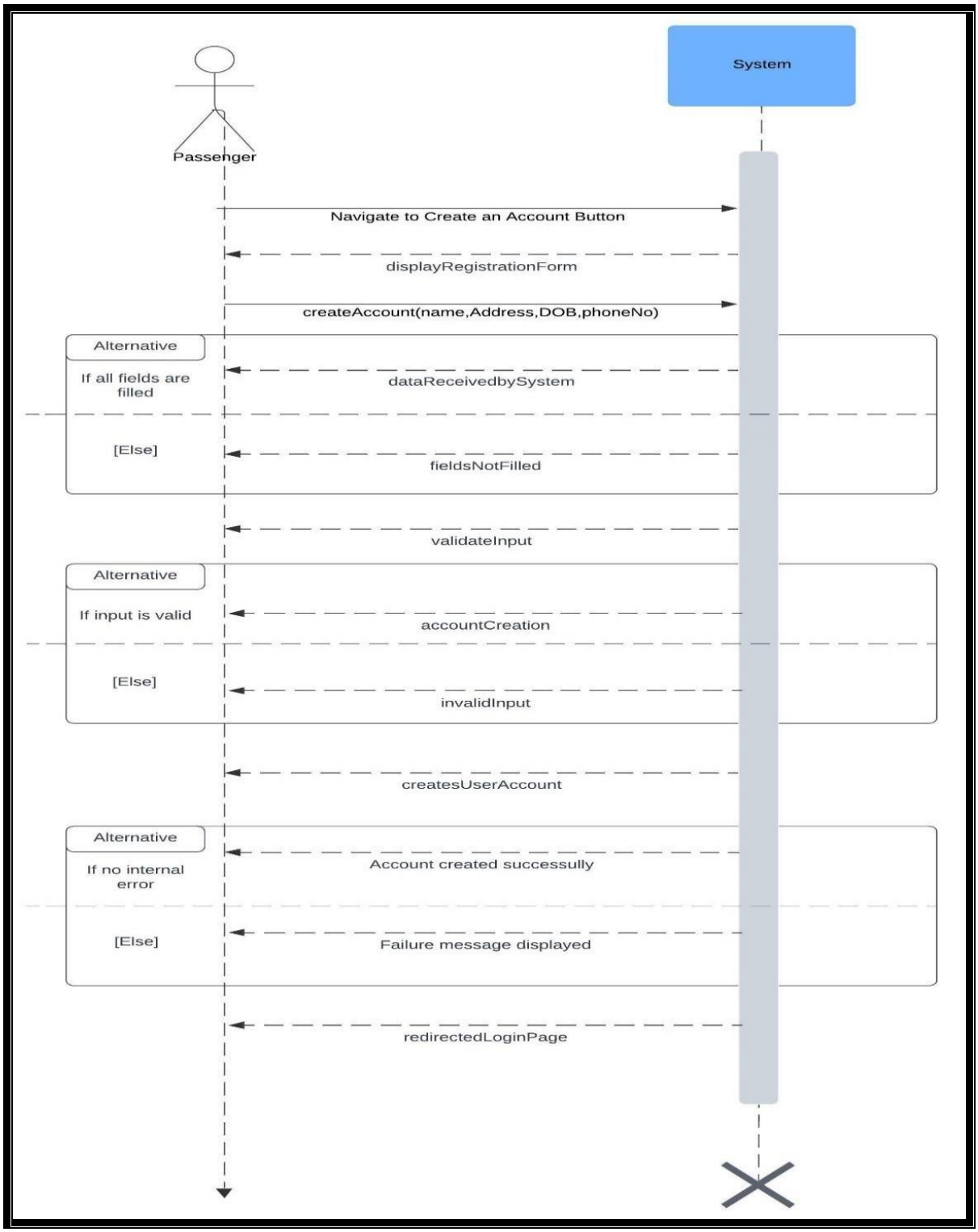
The user interface of the Efficio Bus Management System (BMS) must be intuitive, user-friendly, and consistent with established GUI standards to ensure a smooth user experience. Each interface should include standard elements such as navigation buttons, help options, and easy-to-understand icons, maintaining a uniform look and feel across the system. Screens must follow consistent layout patterns. Error messages must be clear, concise, and consistent throughout the application. The system will require user interfaces for various components, including bus schedule viewing, ticket booking, passenger feedback, and administrative functions.

4 Domain Model

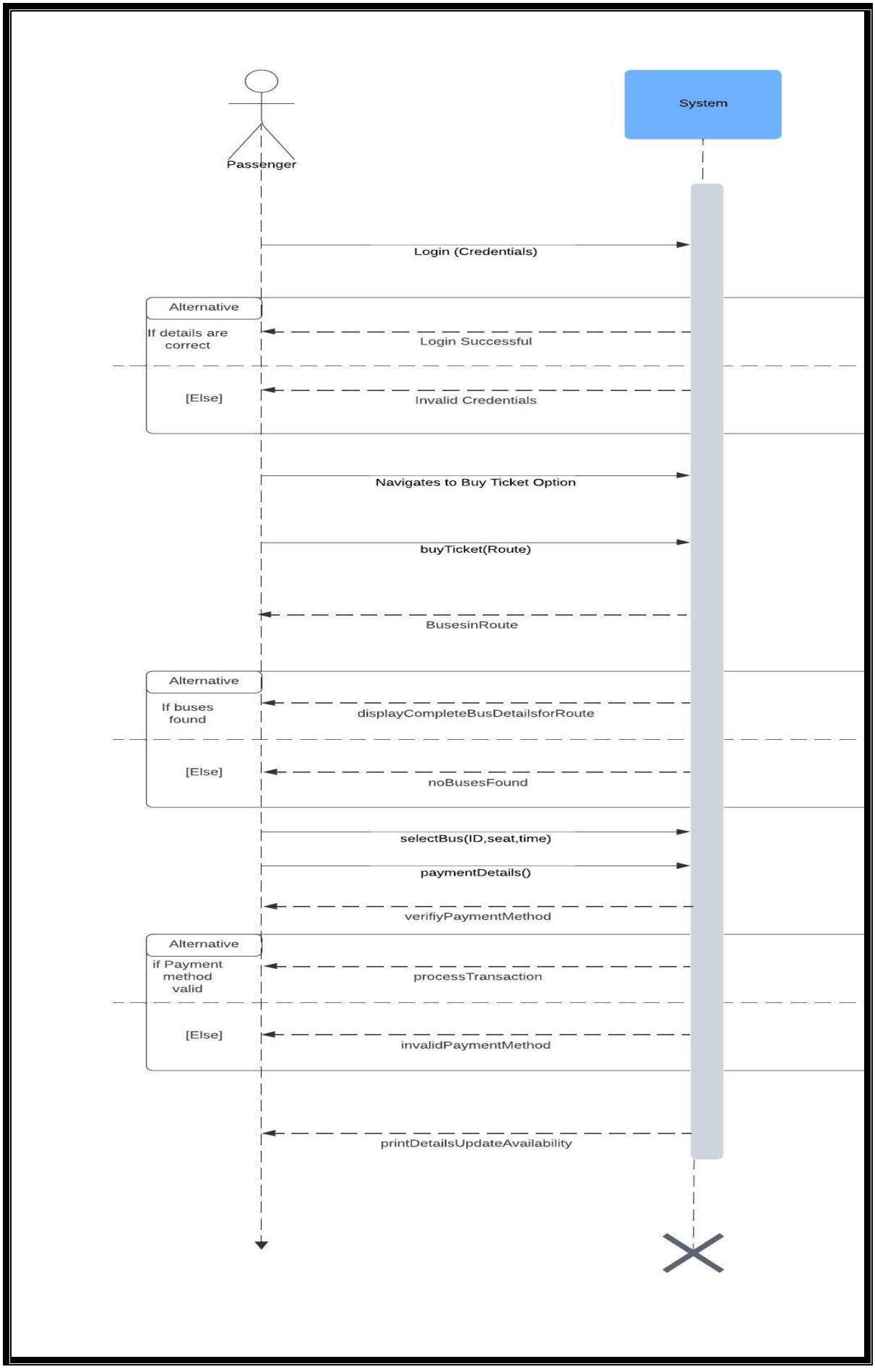


5 System Sequence Diagram

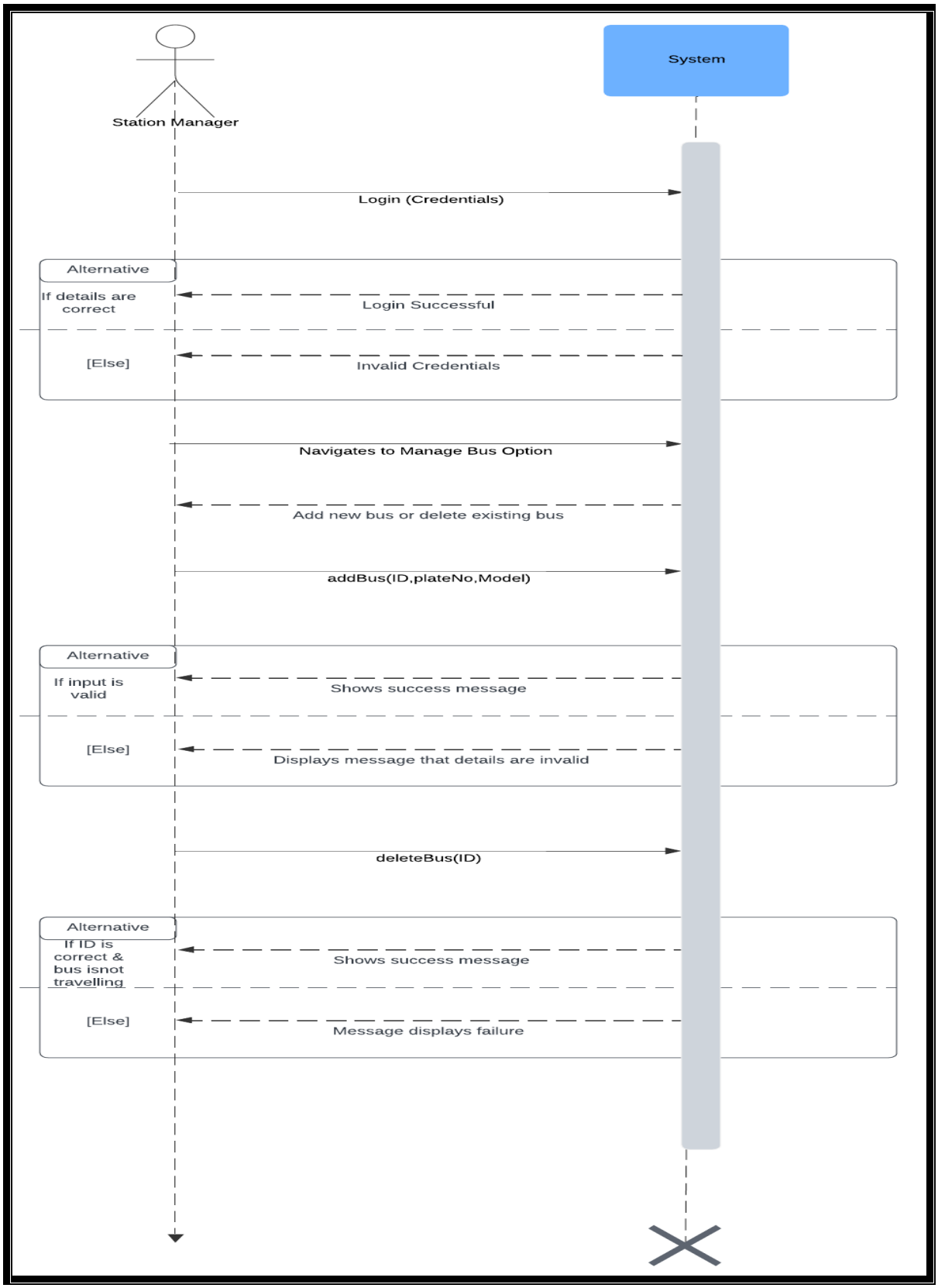
Create An Account



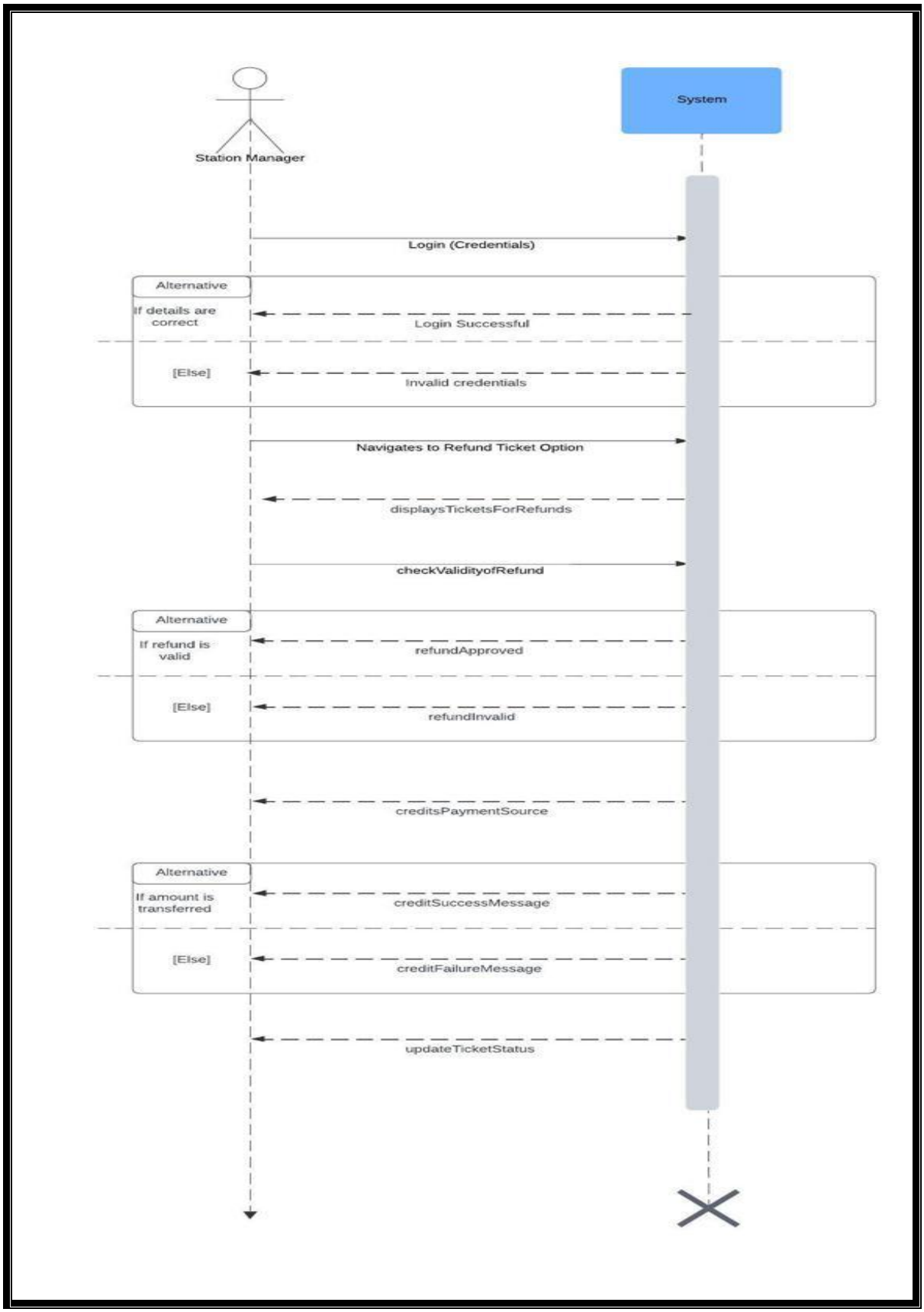
Book Tickets


Passenger
System
Login (Credentials)
Alternative
If details are correct
Login Successful
[Else]
Invalid Credentials
Navigates to Buy Ticket Option
buyTicket(Route)
BusesinRoute
Alternative
If buses found
displayCompleteBusDetailsforRoute
[Else]
noBusesFound
selectBus(ID,seat,time)
paymentDetails()
verifiyPaymentMethod
Alternative
if Payment method valid
processTransaction
[Else]
invalidPaymentMethod
printDetailsUpdateAvailability

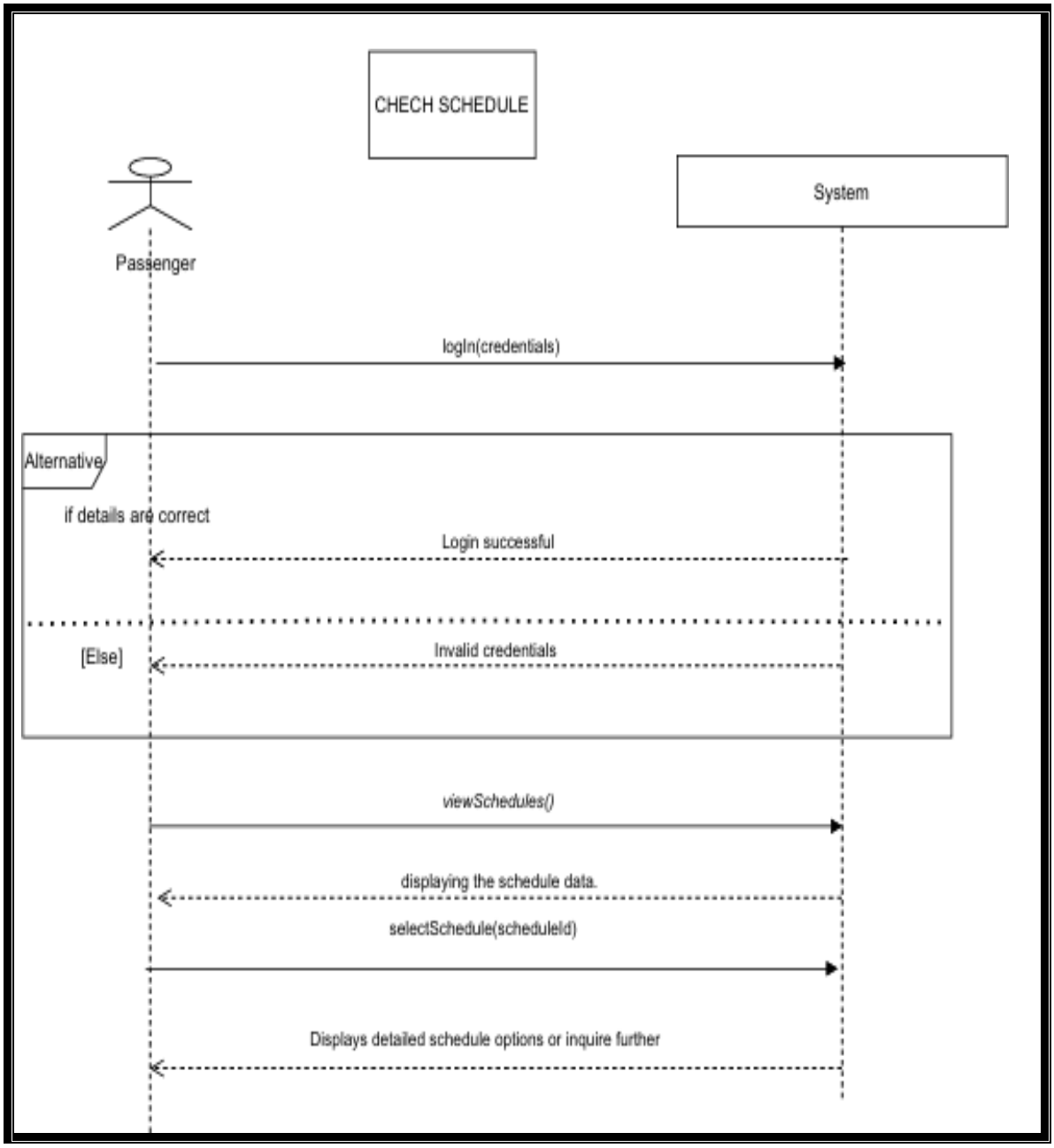
Manage Bus


Station Manager
System
Login (Credentials)
Alternative
If details are correct
Login Successful
[Else]
Invalid Credentials
Navigates to Manage Bus Option
Add new bus or delete existing bus
addBus(ID,plateNo,Model)
Alternative
If input is valid
Shows success message
[Else]
Displays message that details are invalid
deleteBus(ID)
Alternative
If ID is correct & bus isnot travelling
Shows success message
[Else]
Message displays failure

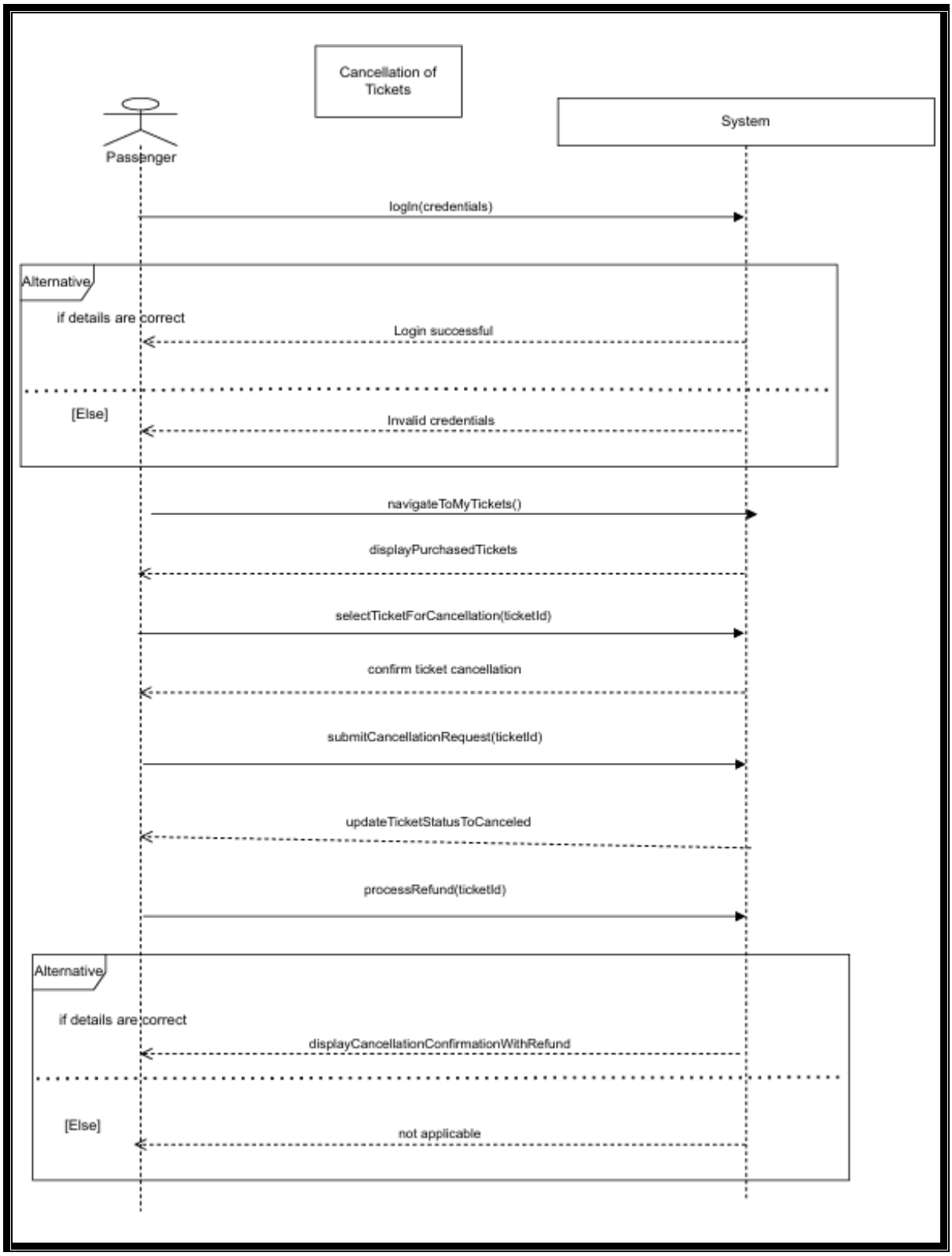
Refund Ticket



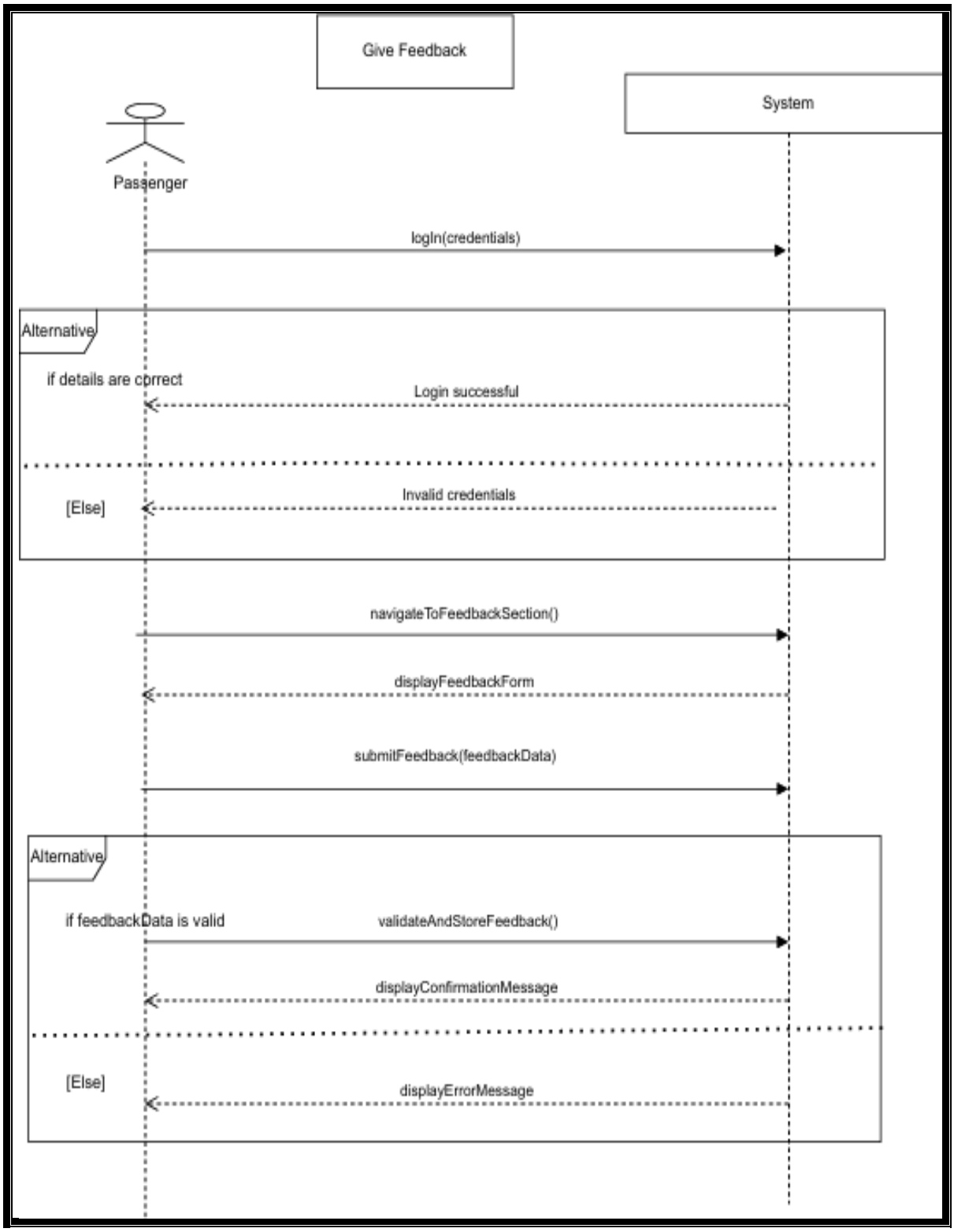
Check Schedule



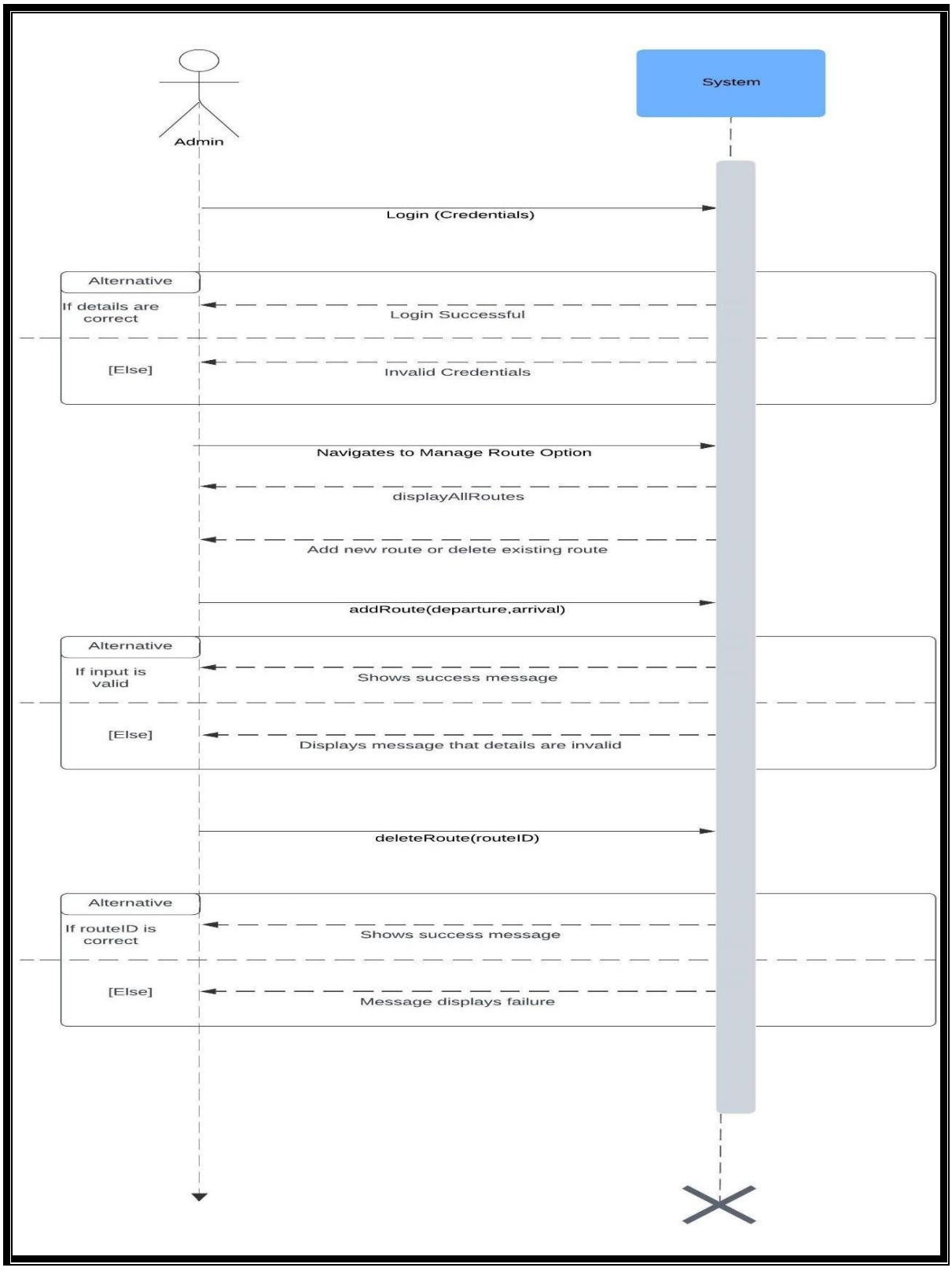
Cancel Booking



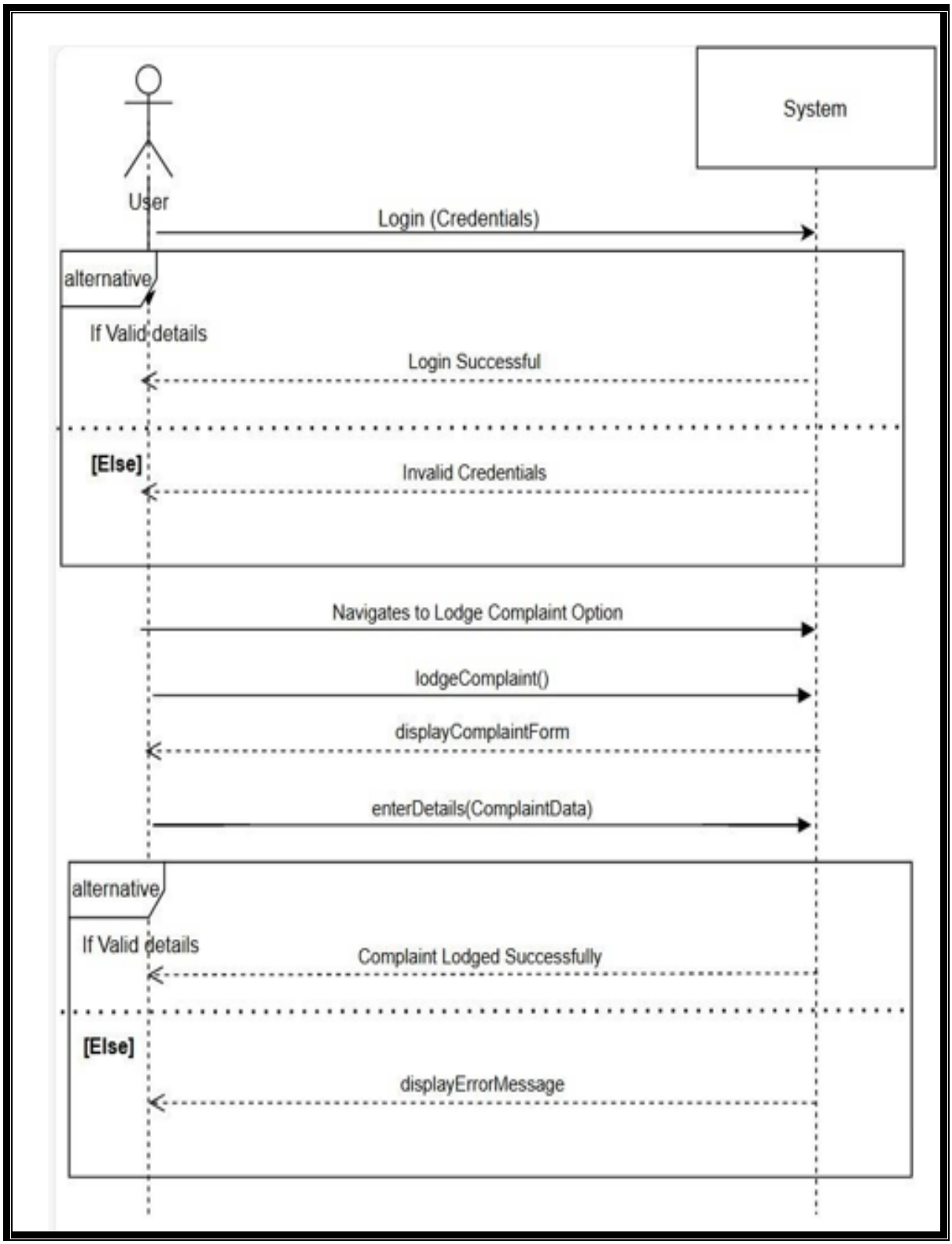
Give Feedback



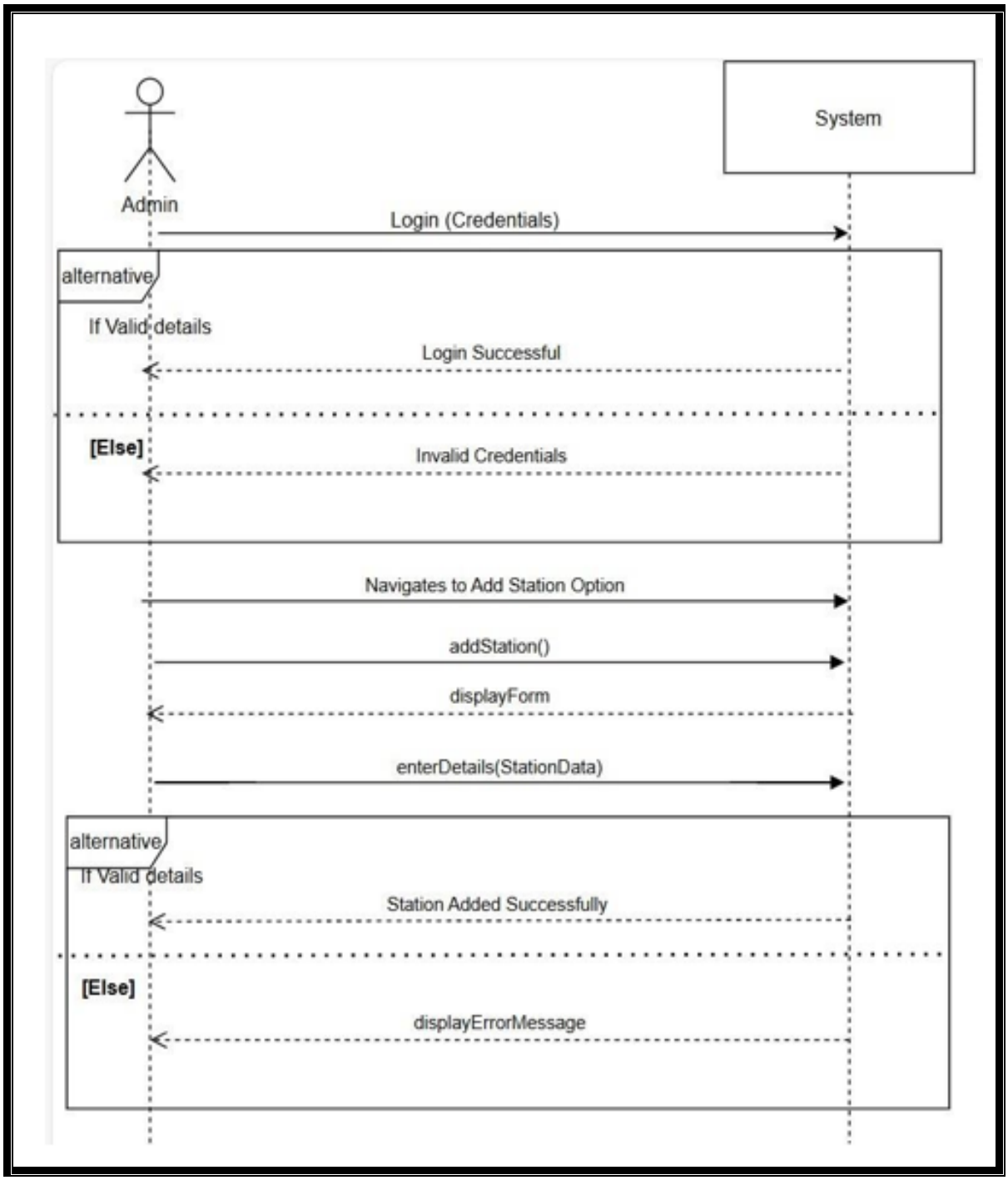
Manage Route



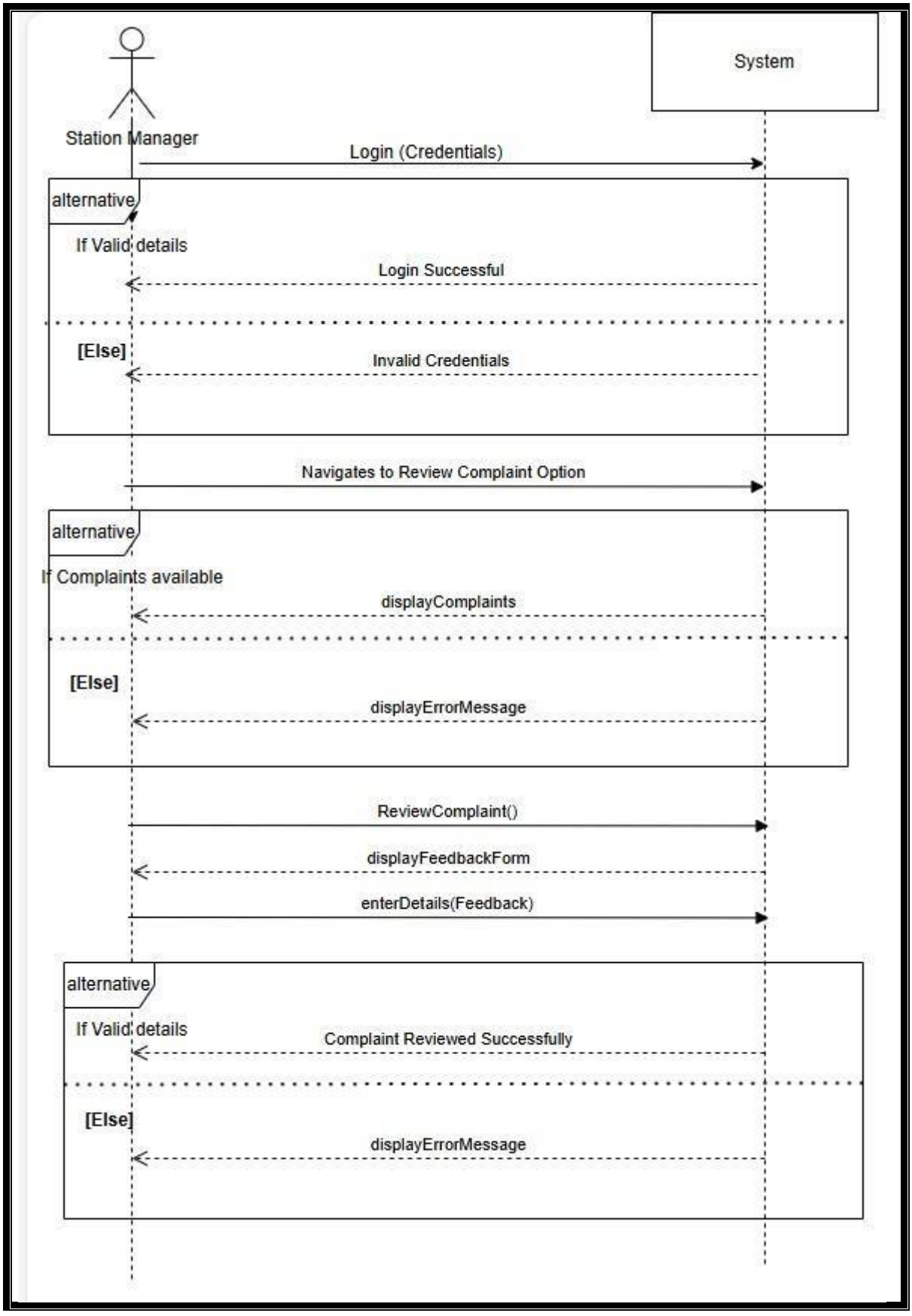
Lodge Complaint



Add Station


System
Admin
Login (Credentials)
alternative
If Valid details
Login Successful
[Else]
Invalid Credentials
Navigates to Add Station Option
addStation()
displayForm
enterDetails(StationData)
alternative
If Valid details
Station Added Successfully
[Else]
displayErrorMessage

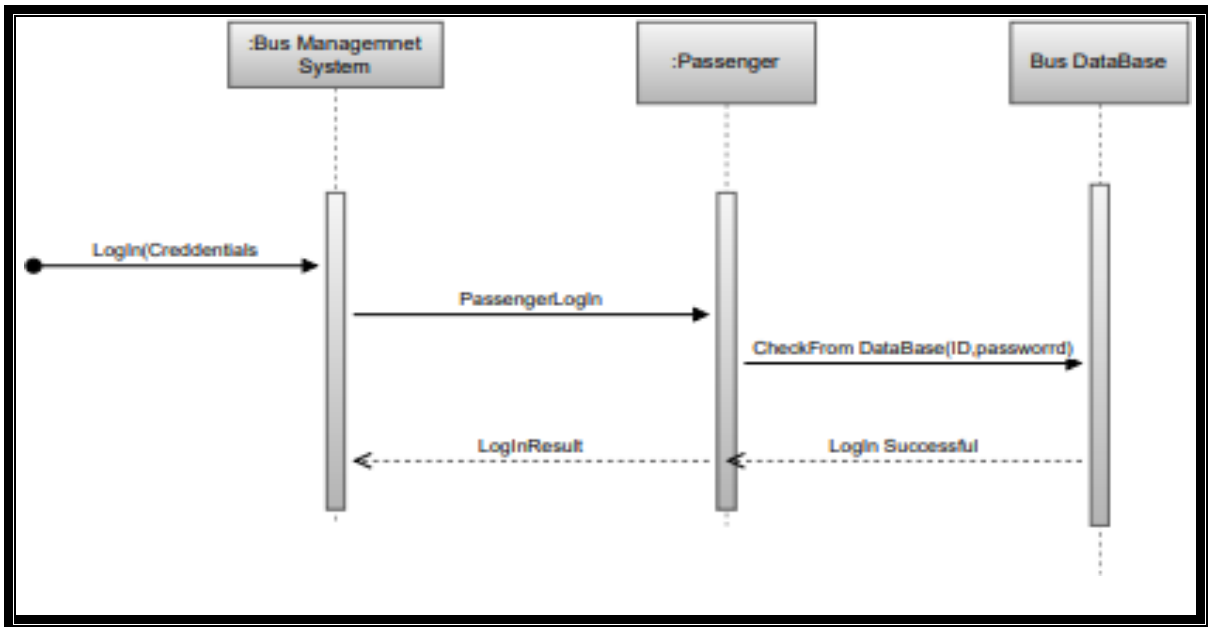
Review Complaint



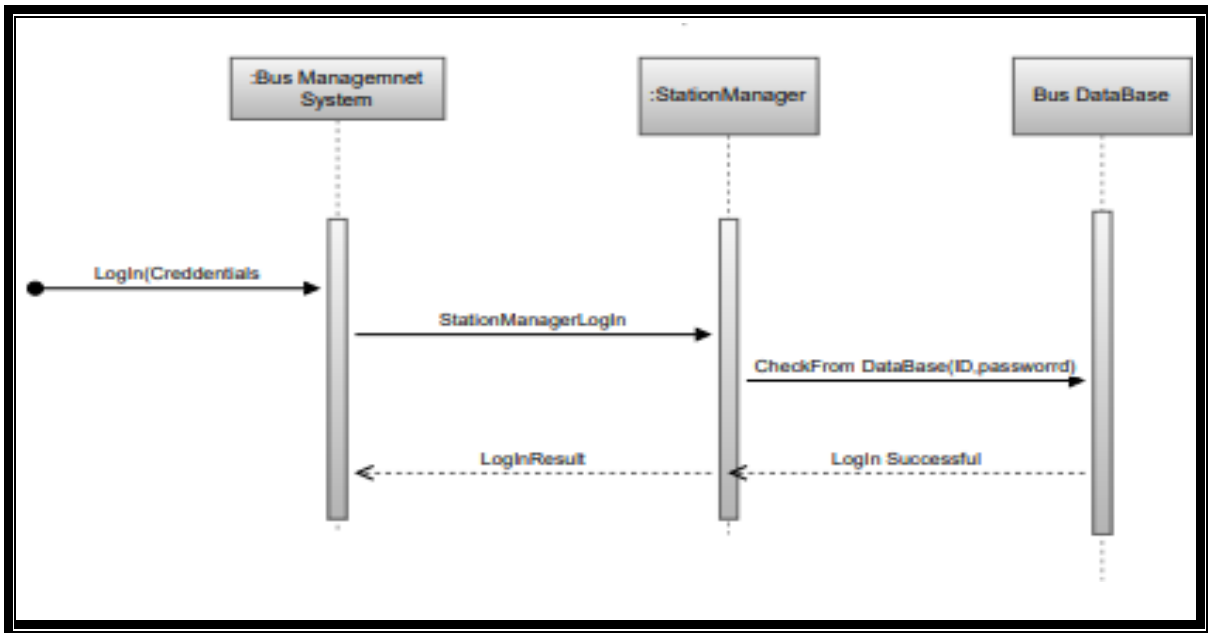
6 Sequence Diagram

Login

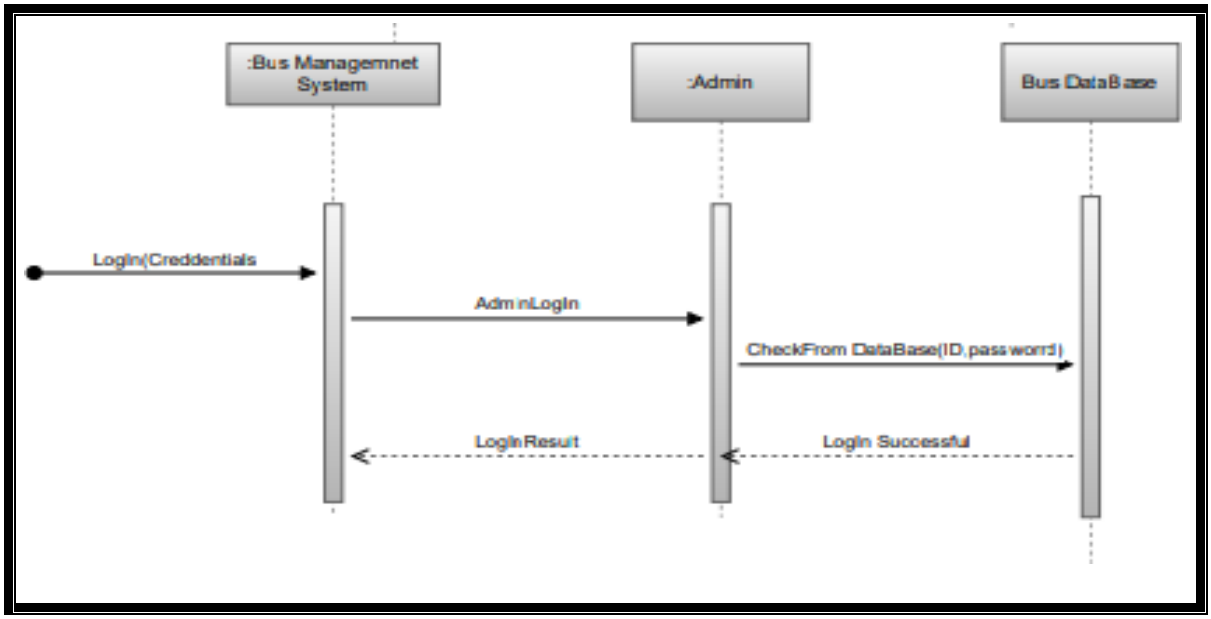
1



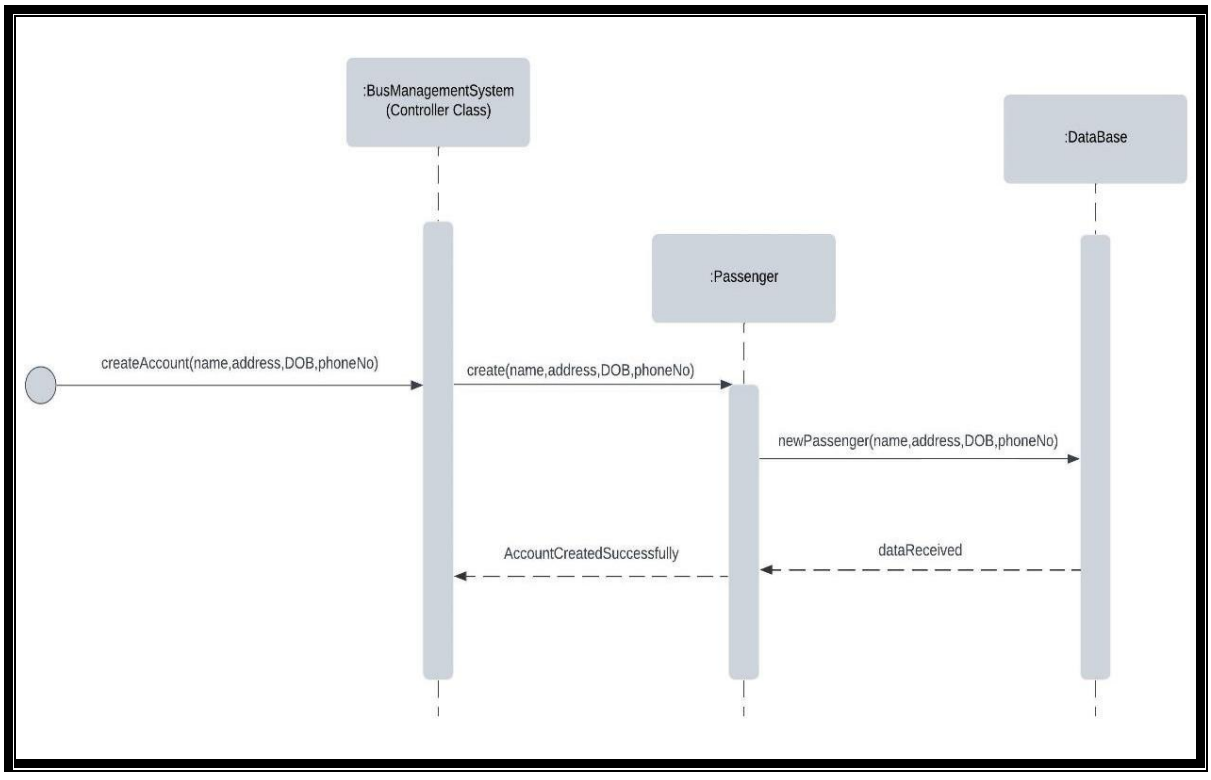
2



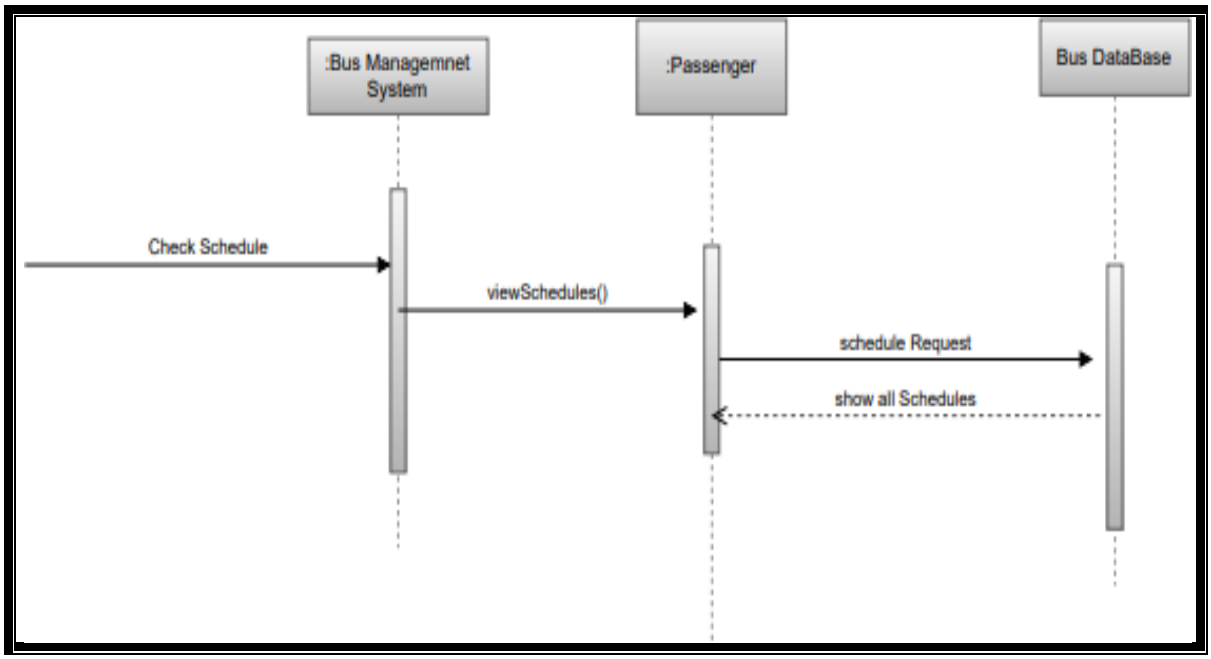
3

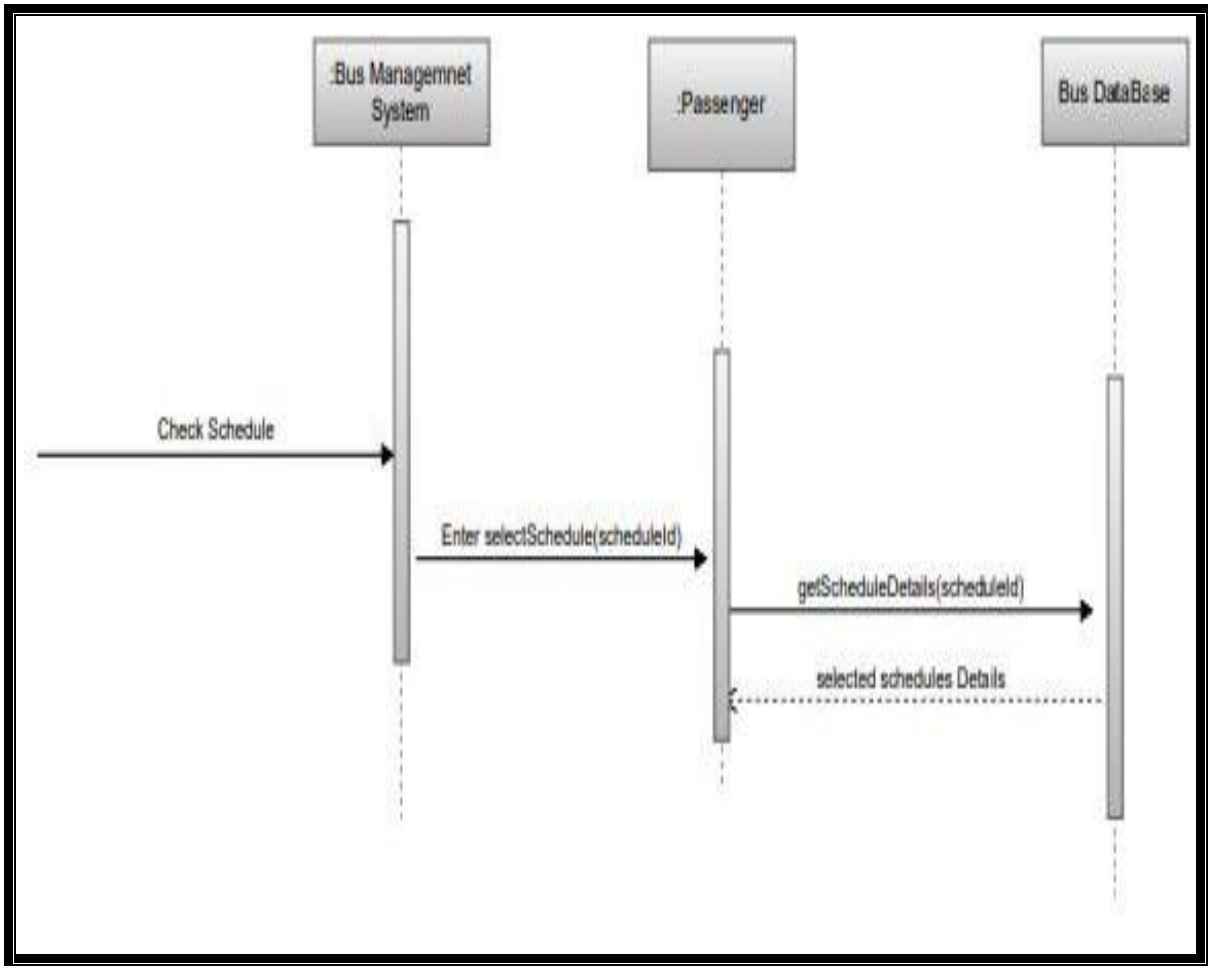


Create an Account

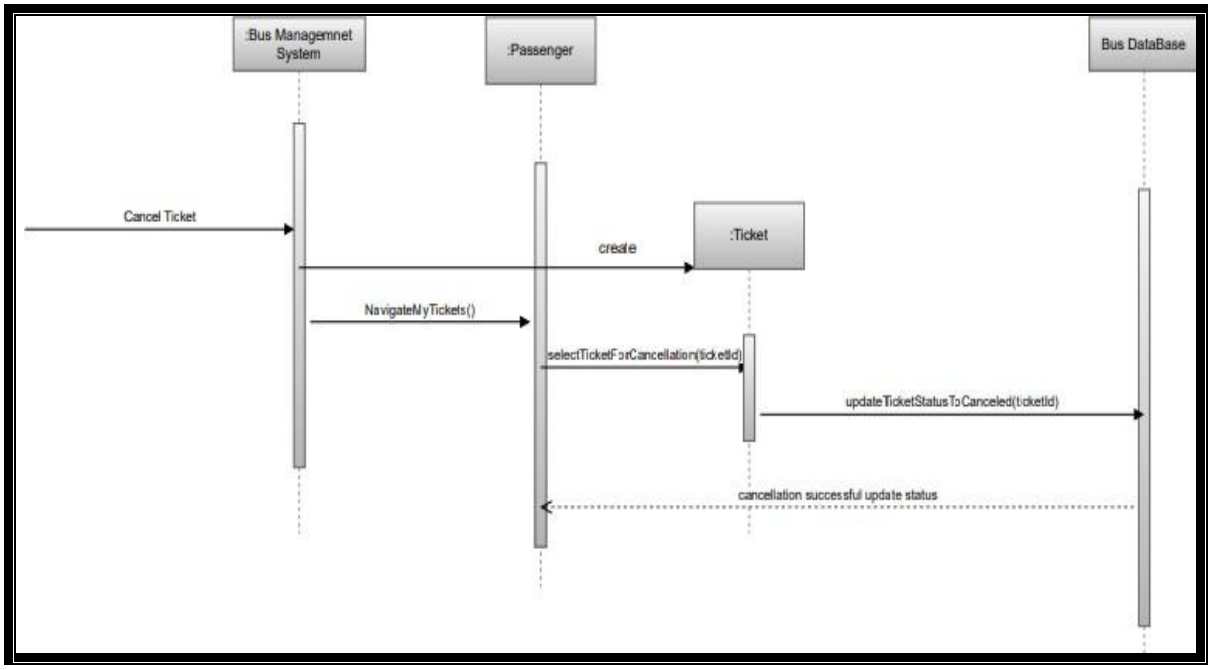


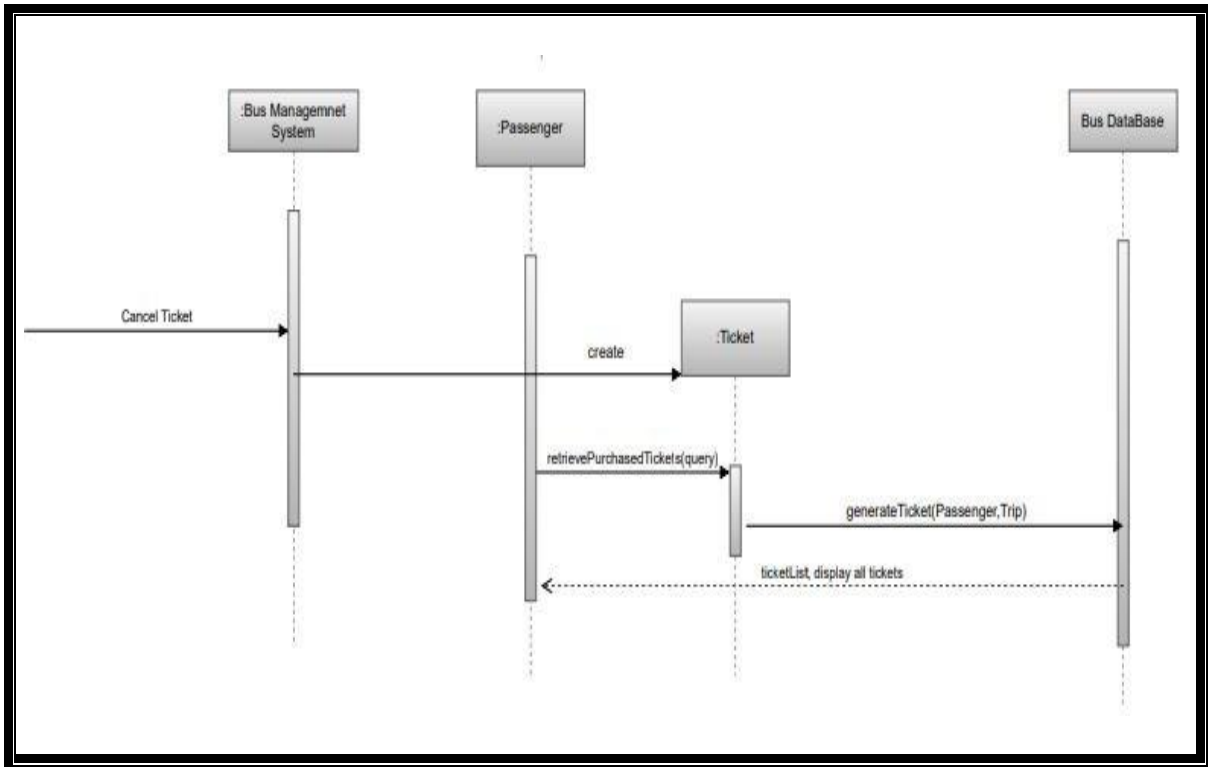
Check Schedule



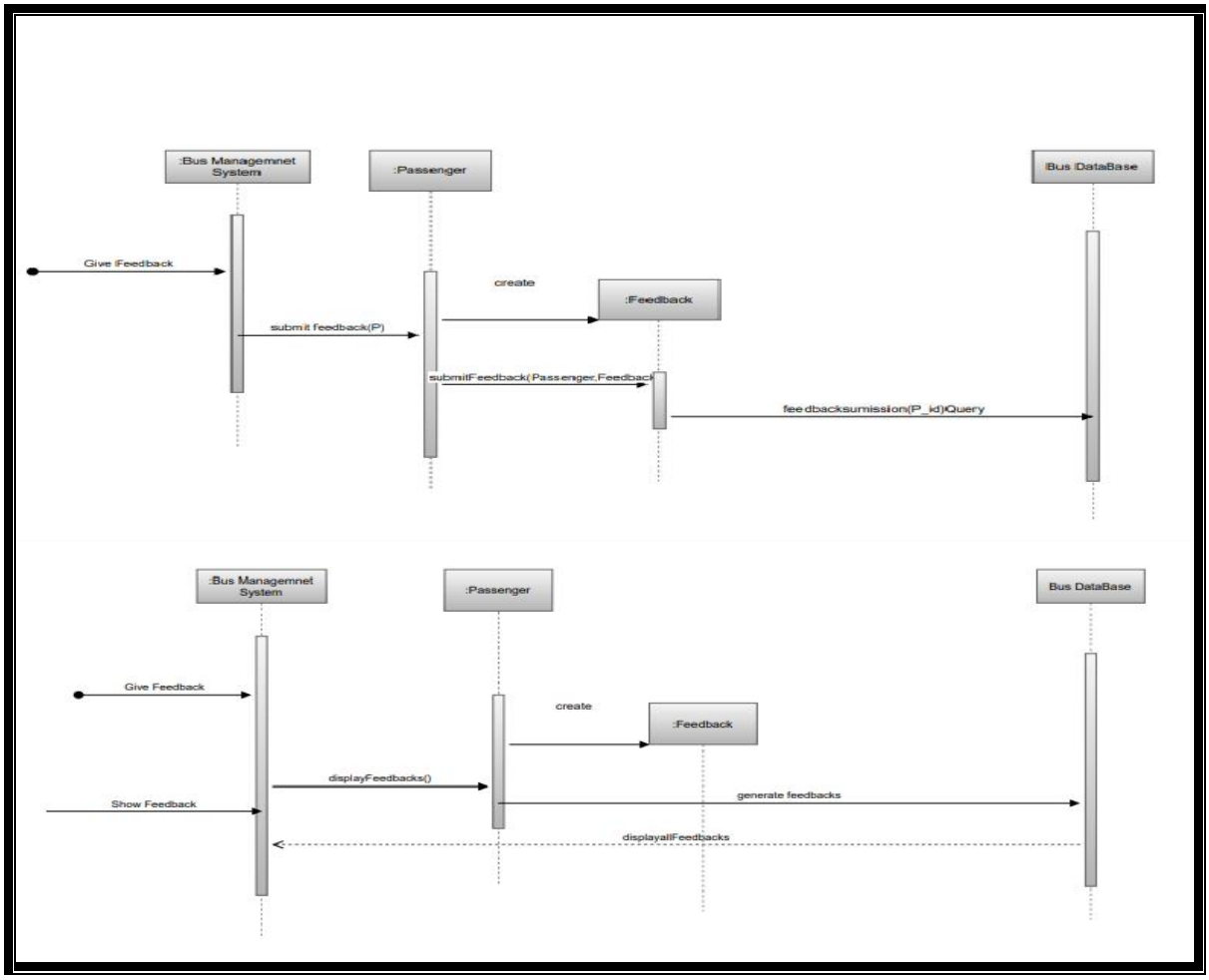


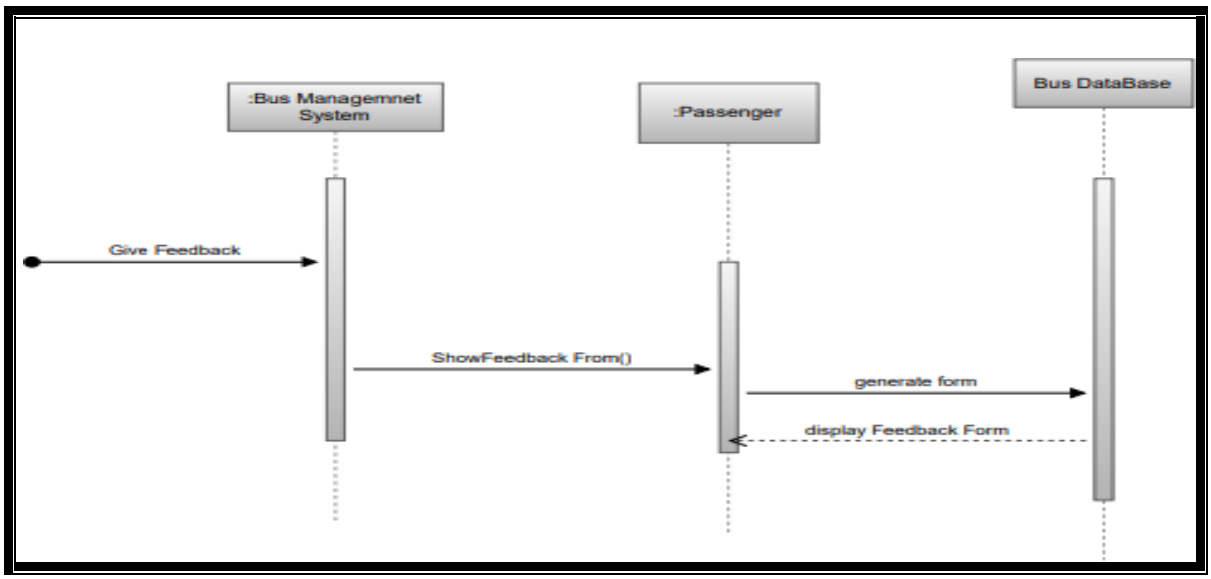
Cancel Booking



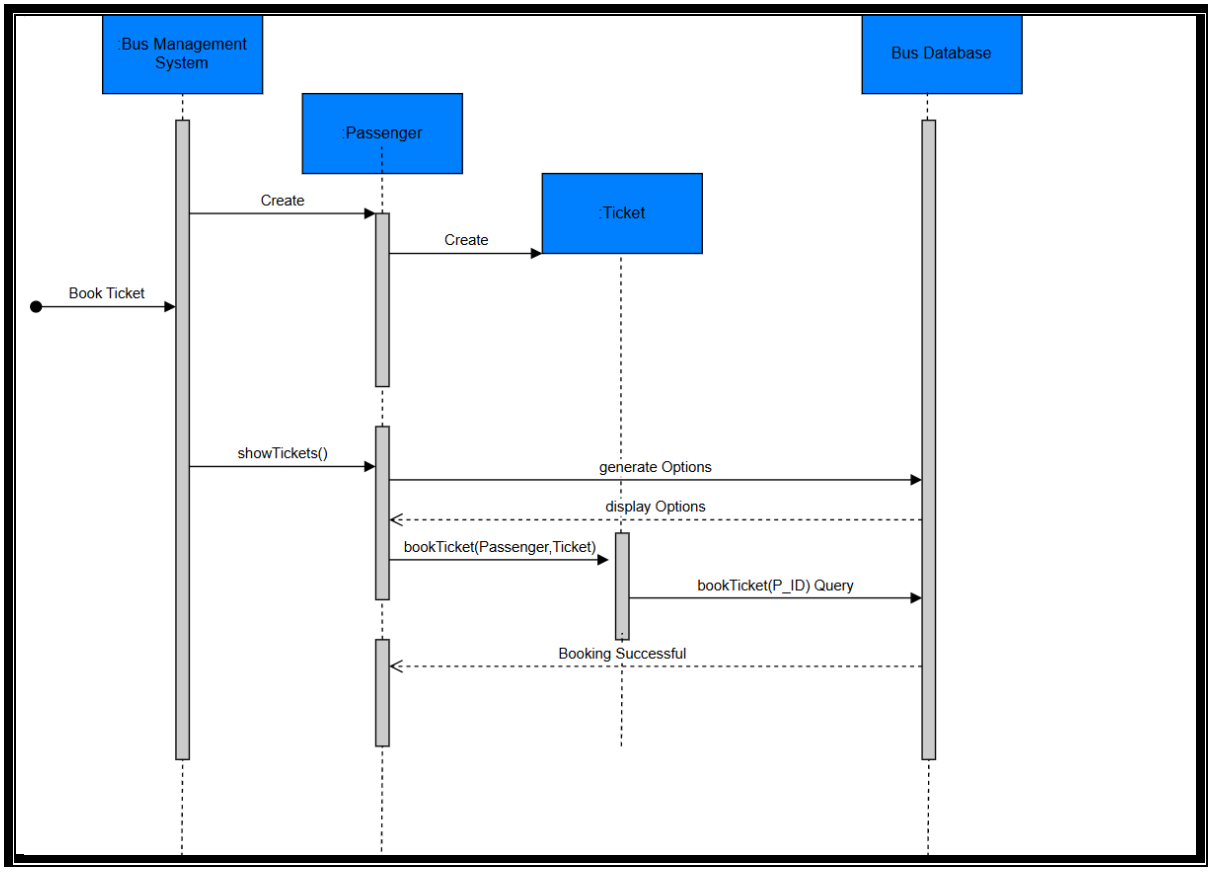


Give Feedback

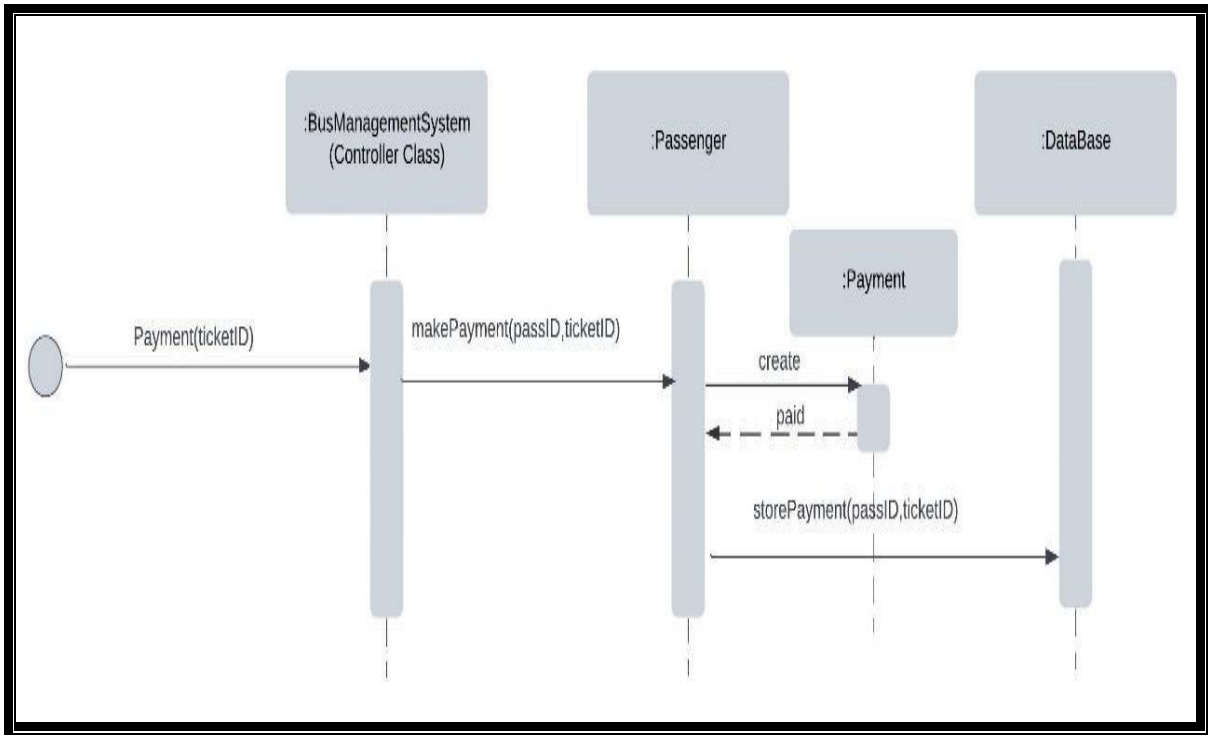




Book Ticket

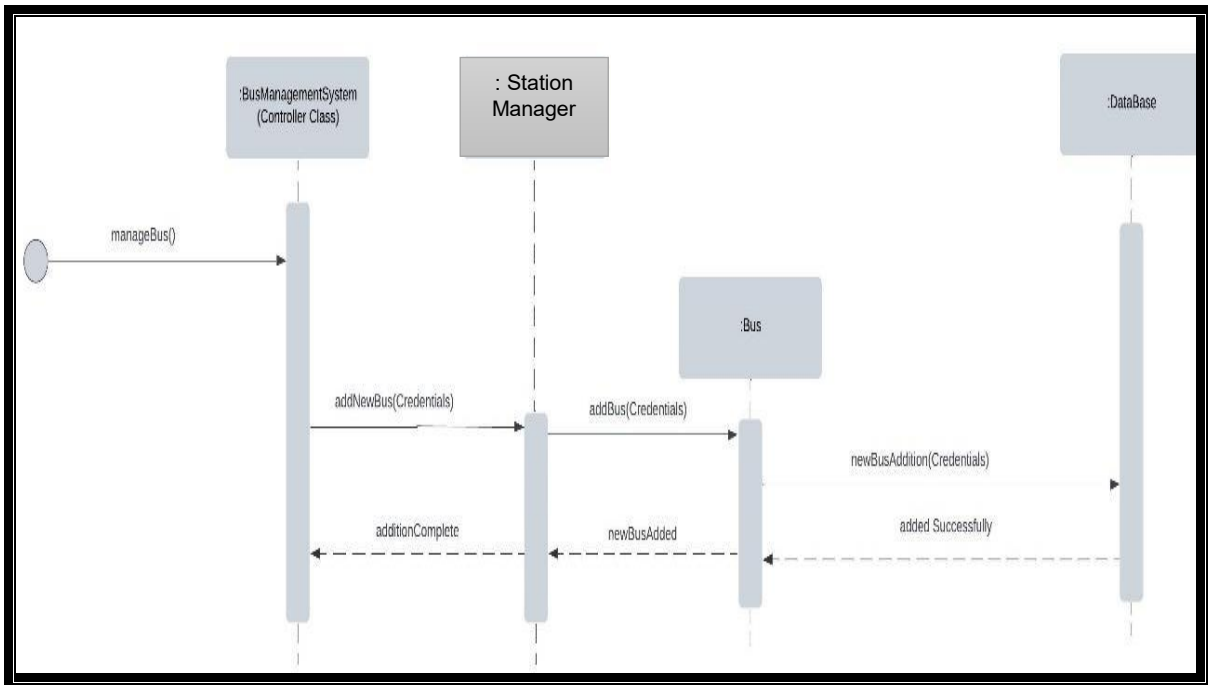


Proceed Payment

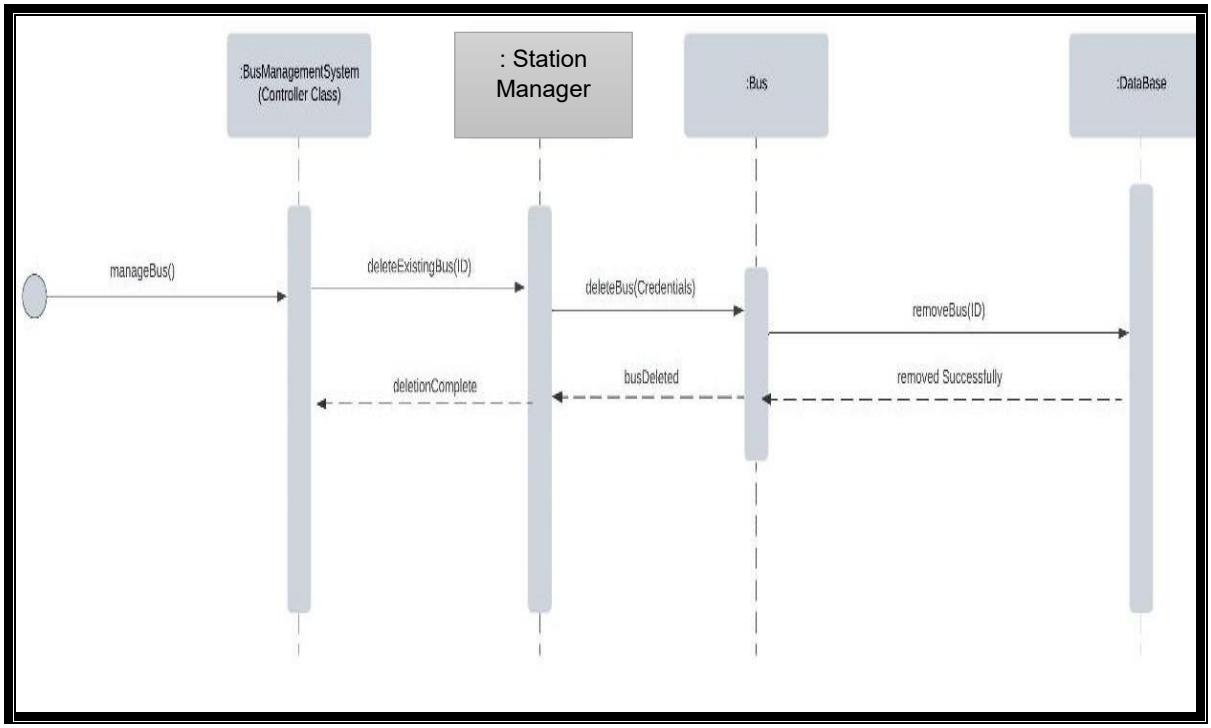


Manage Bus

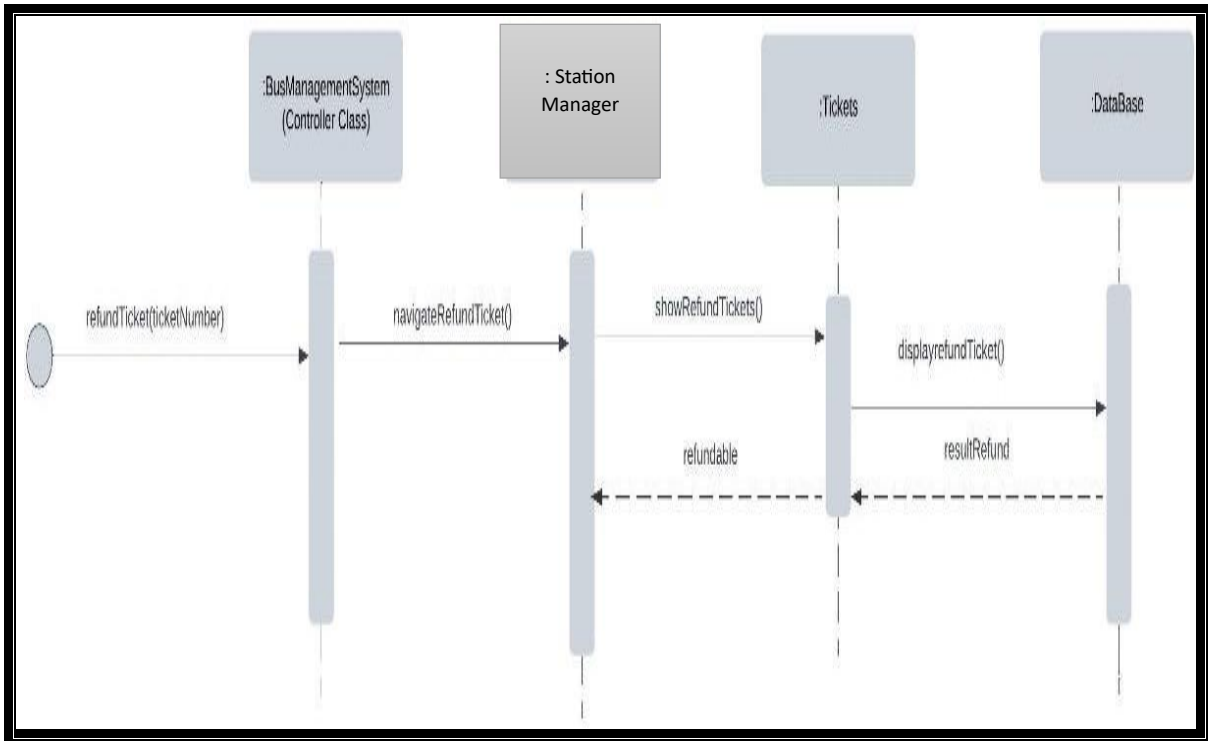
Add Bus

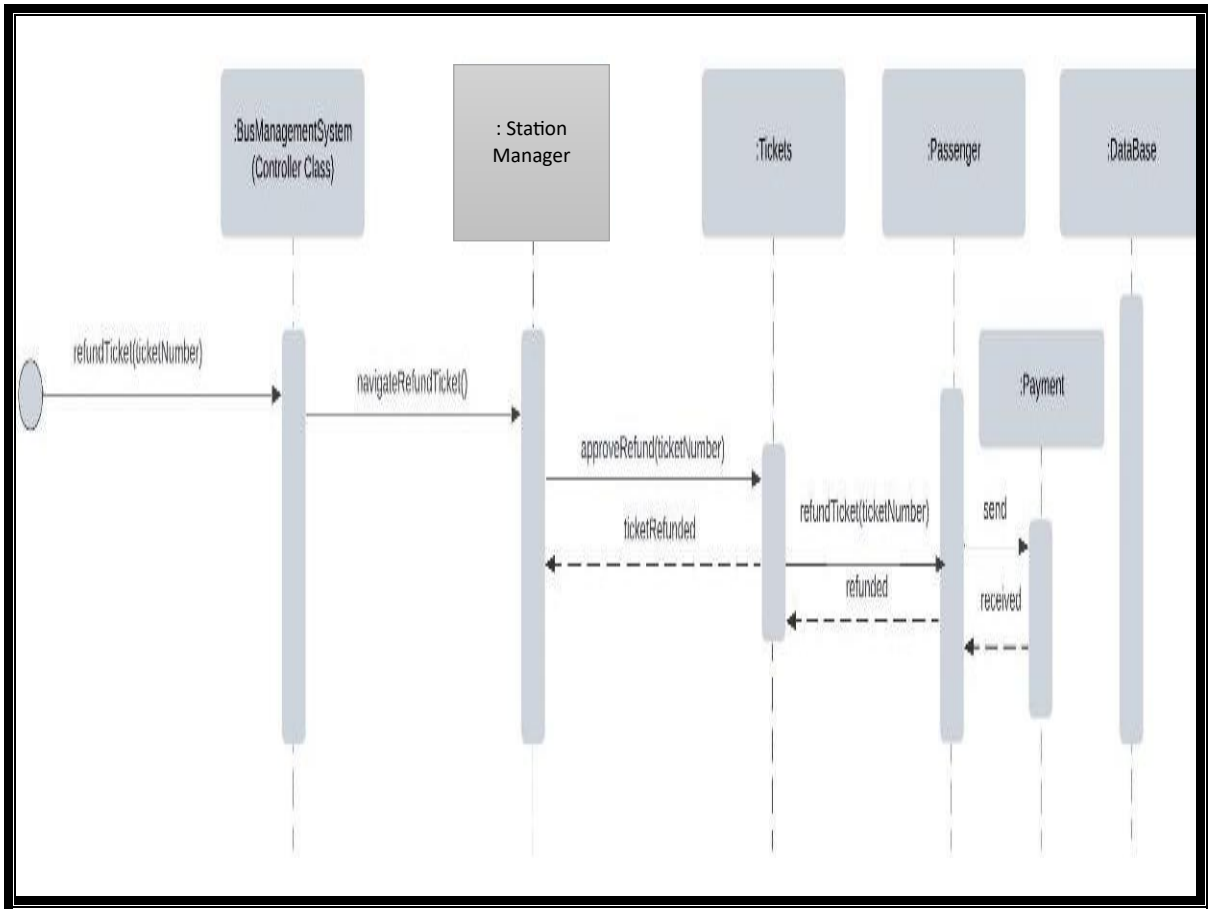


Delete Bus



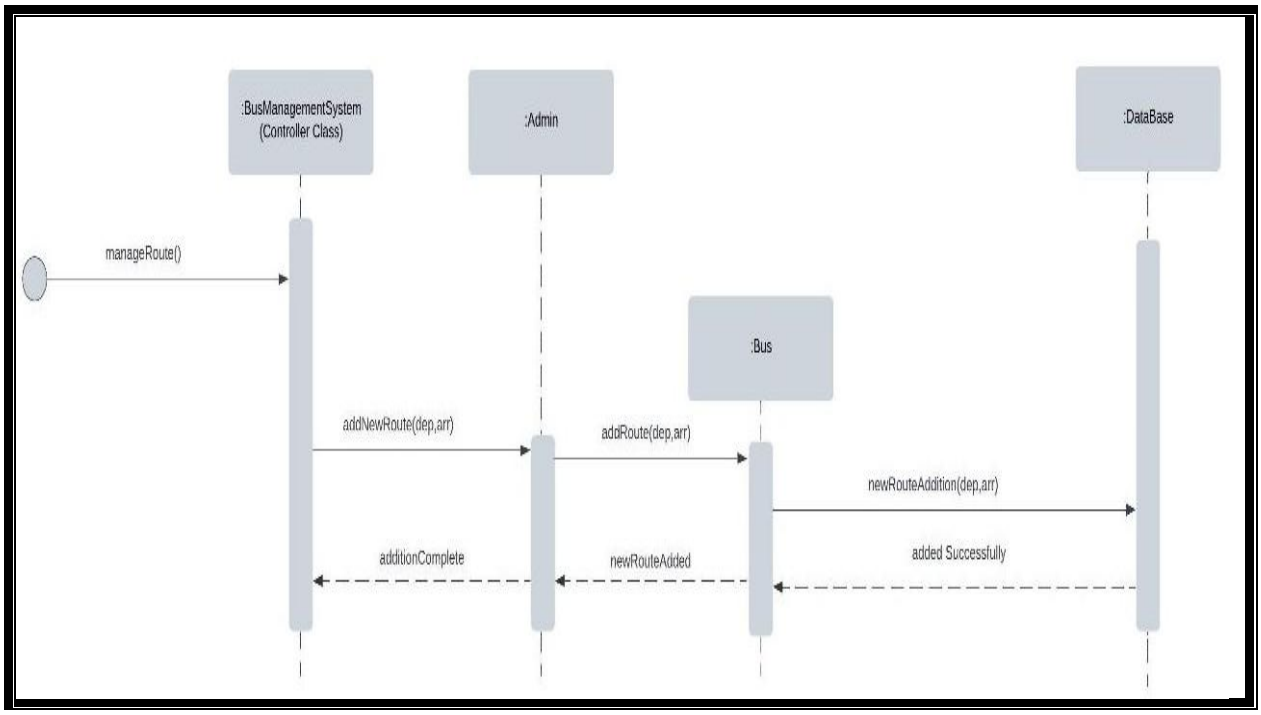
Refund Ticket



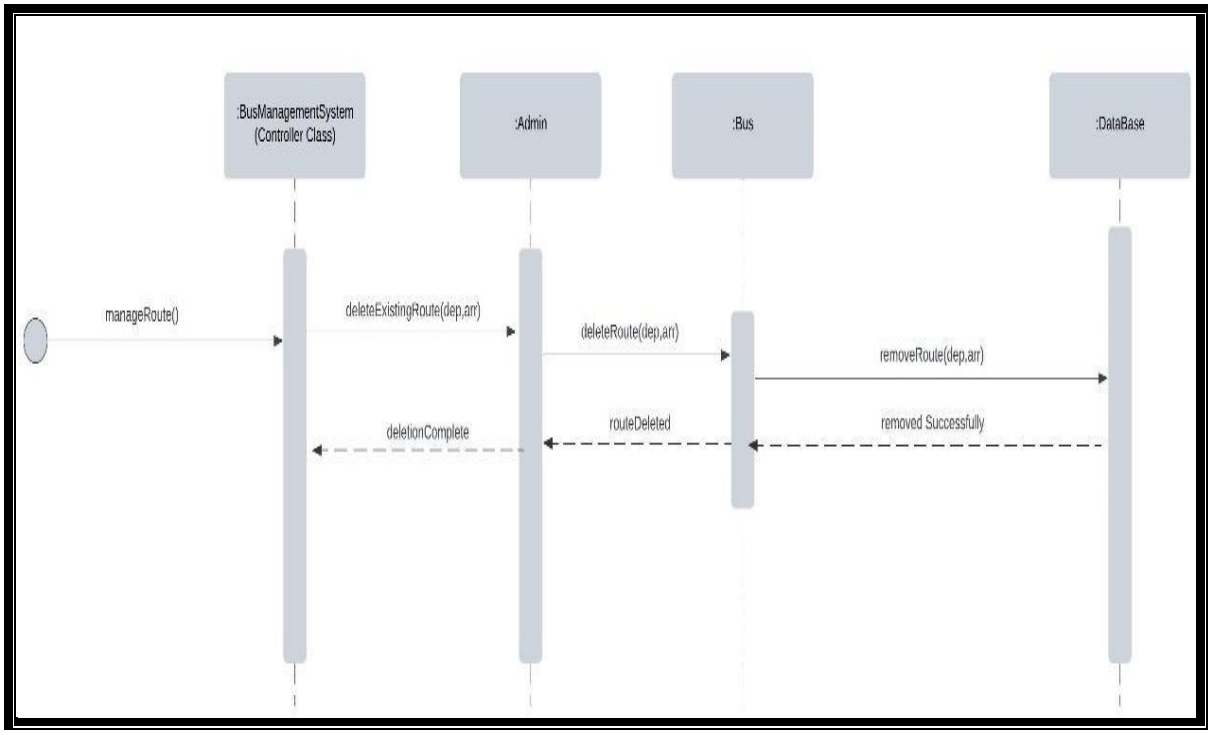


Manage Route

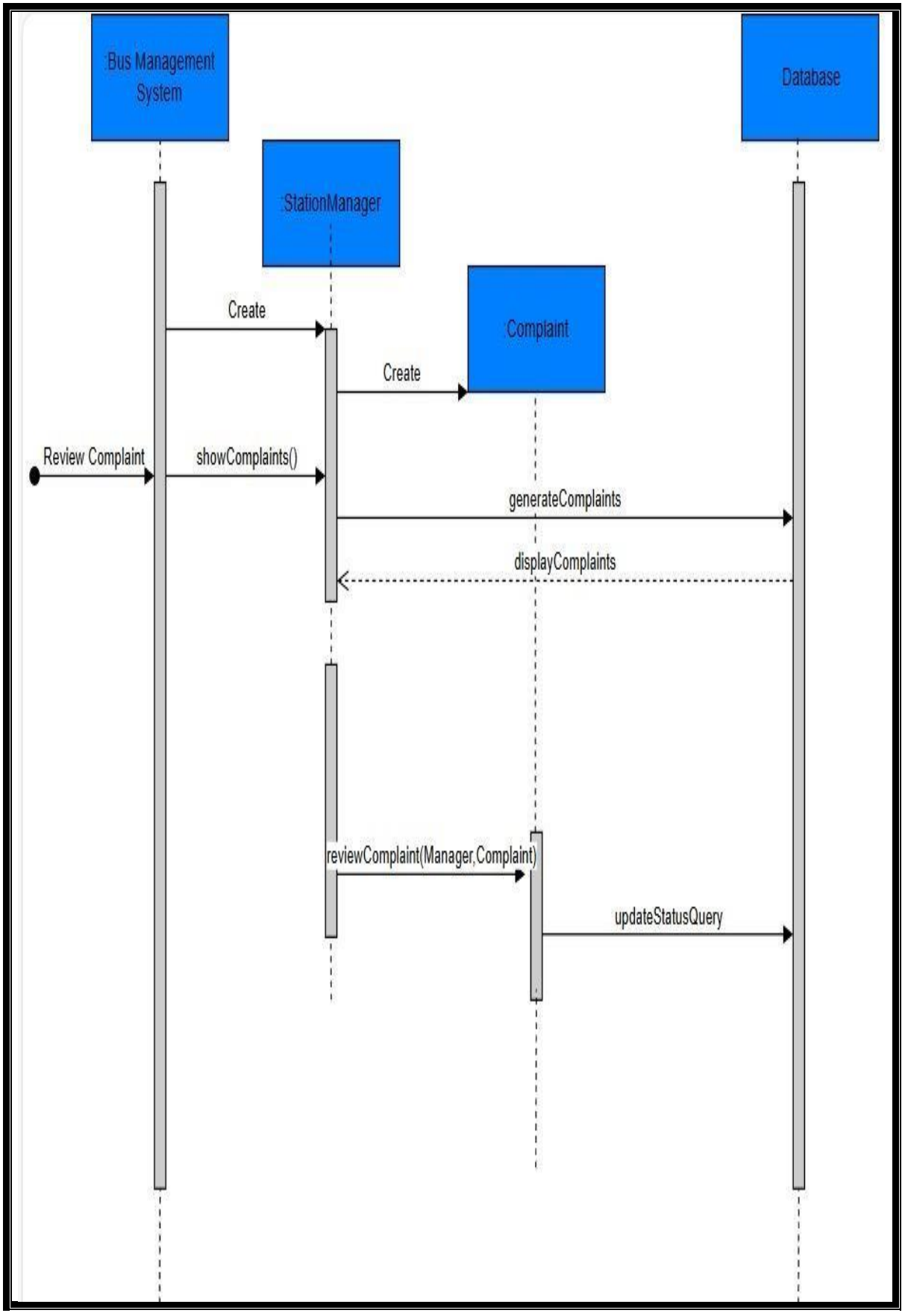
Add Route



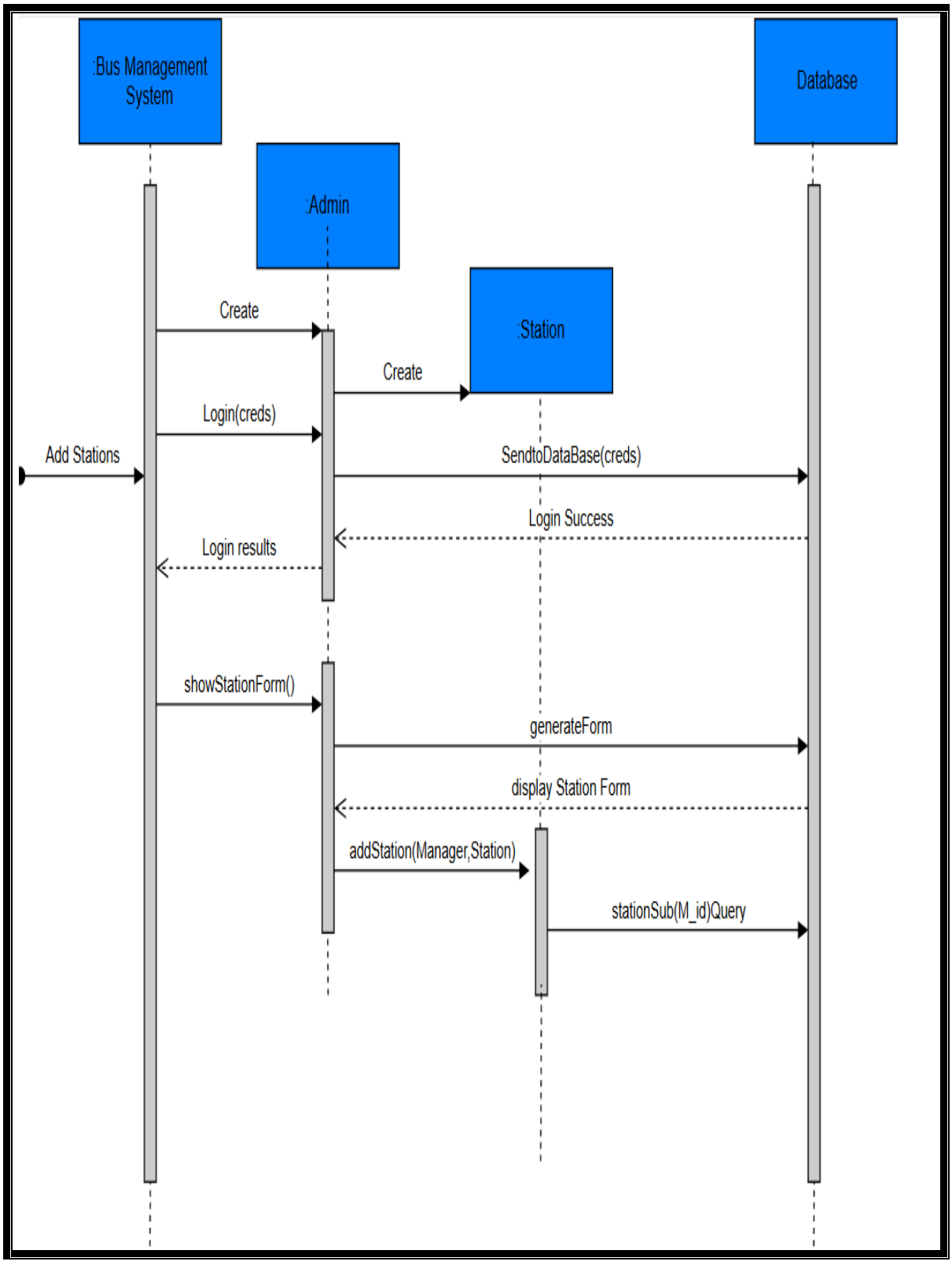
Delete Route



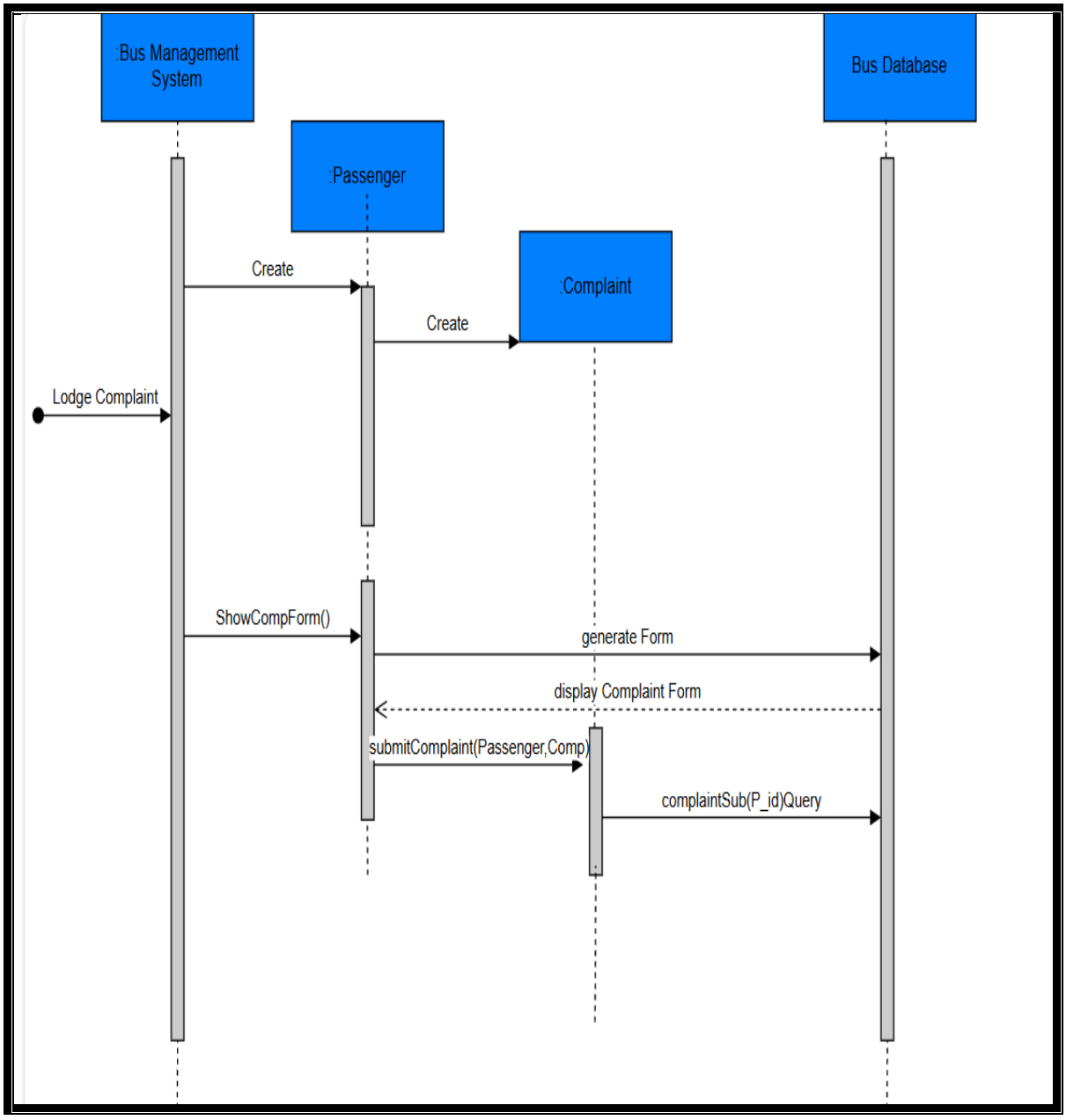
Review Complaint



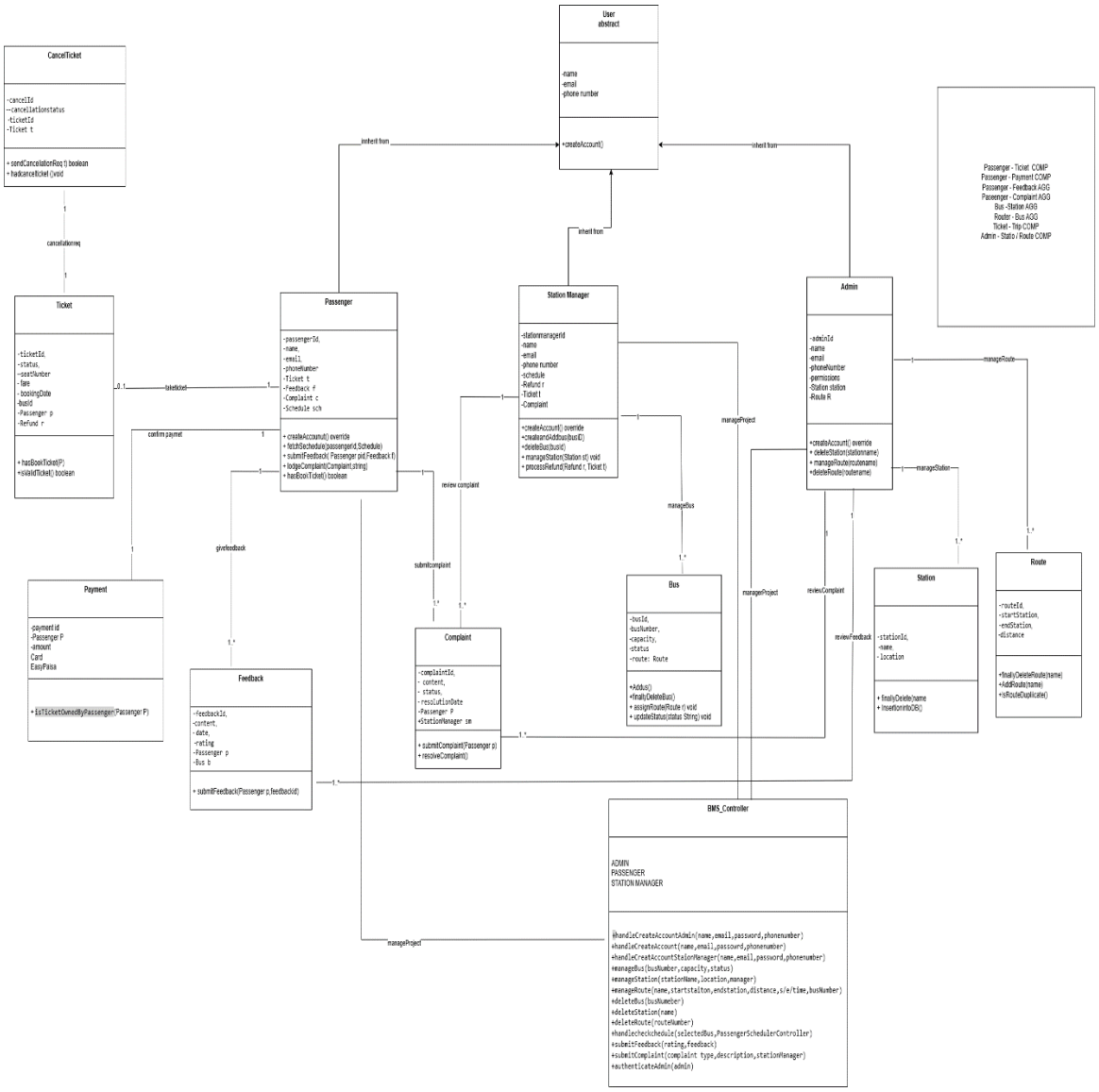
Add Station


:Bus Management System
:Admin
:Station
Database
Create
Create
Login(creds)
Add Stations
SendtoDataBase(creds)
Login Success
Login results
showStationForm()
generateForm
display Station Form
addStation(Manager,Station)
stationSub(M_id)Query

Lodge Complaint

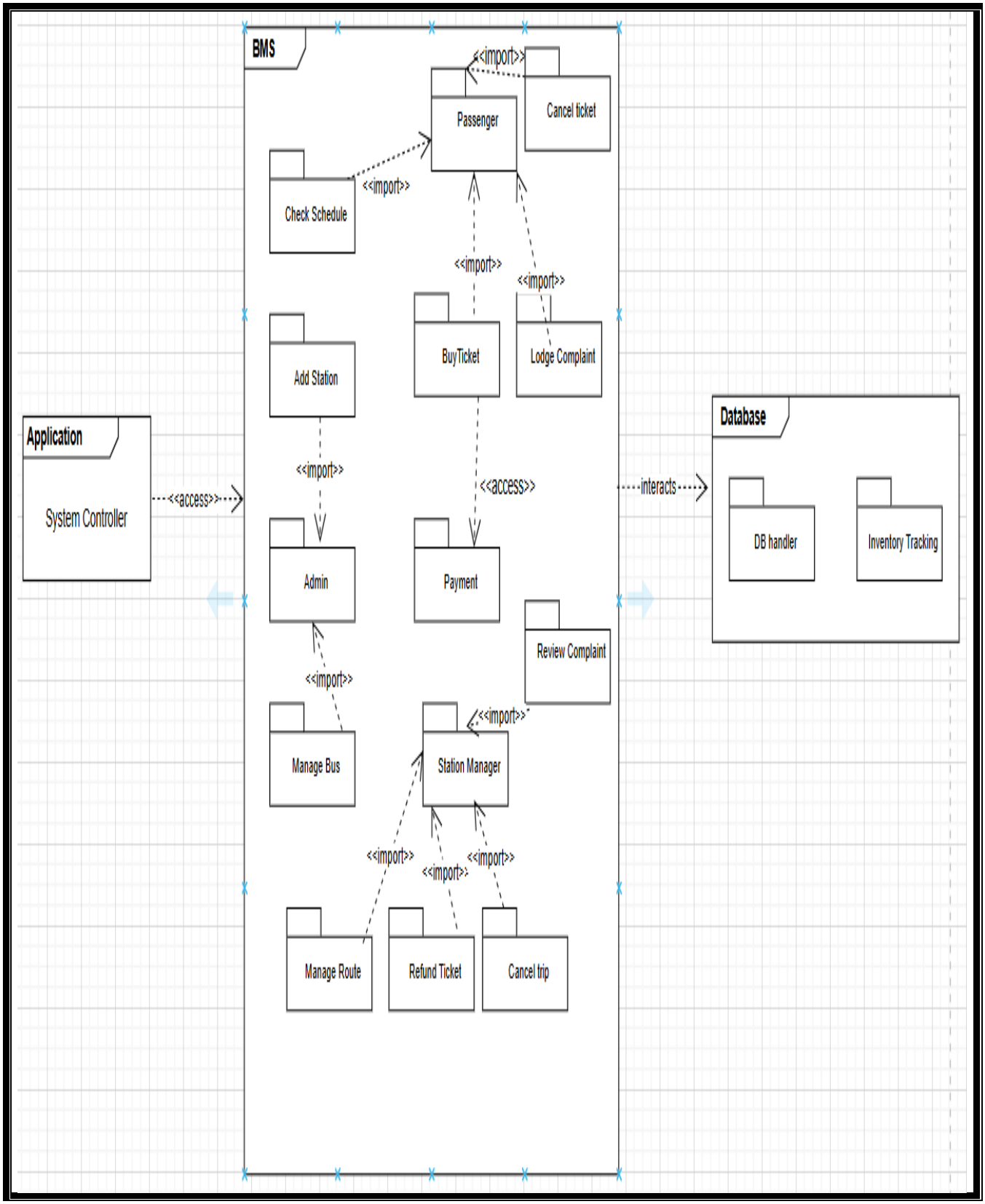


7 Class Diagram

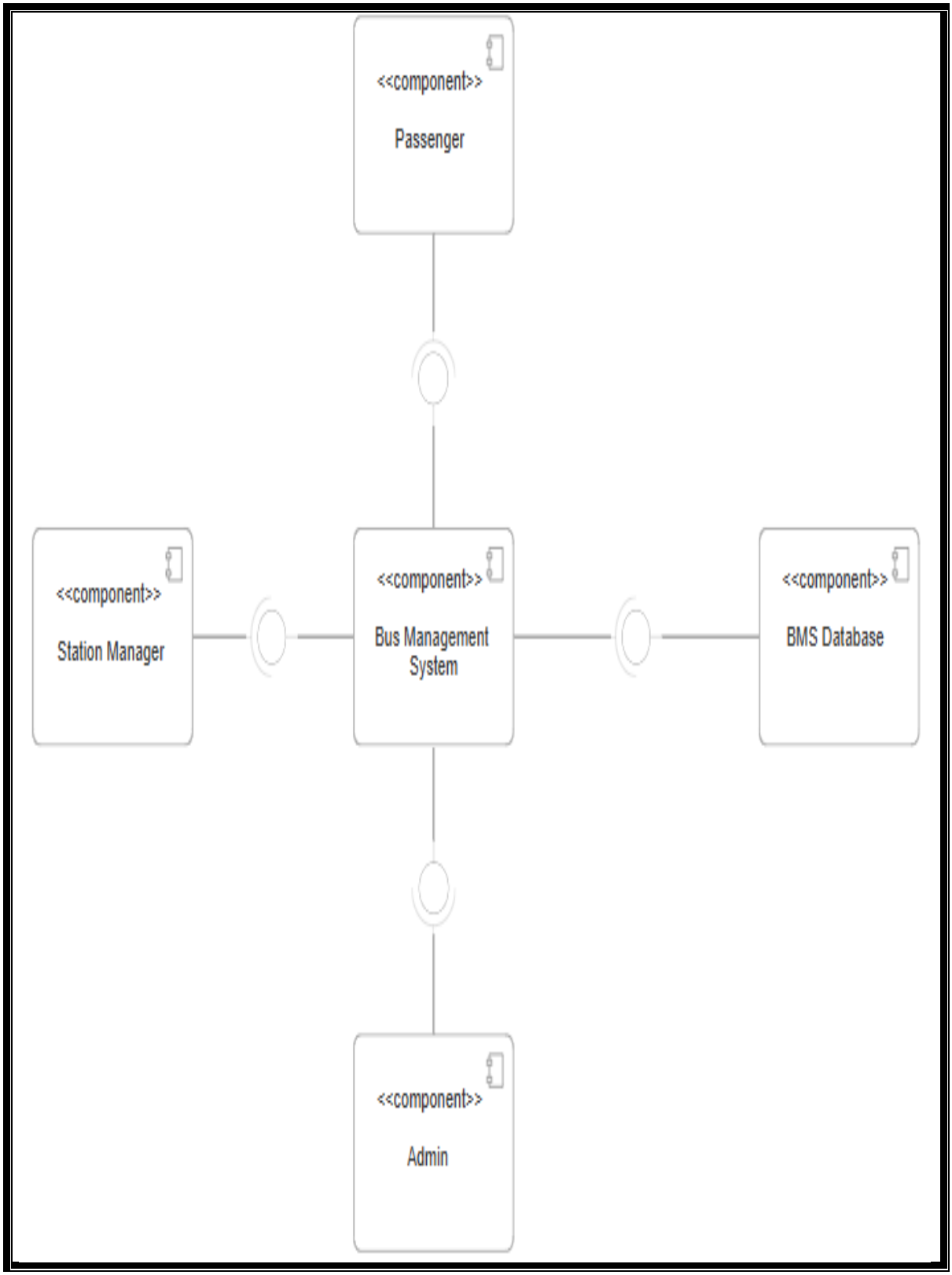


Passenger - Ticket COMP
Passenger - Payment COMP
Passenger - Feedback AGG
Passenger - Complaint AGG
Bus -Station AGG
Router - Bus AGG
Ticket - Trip COMP
Admin - Statio / Router COMP

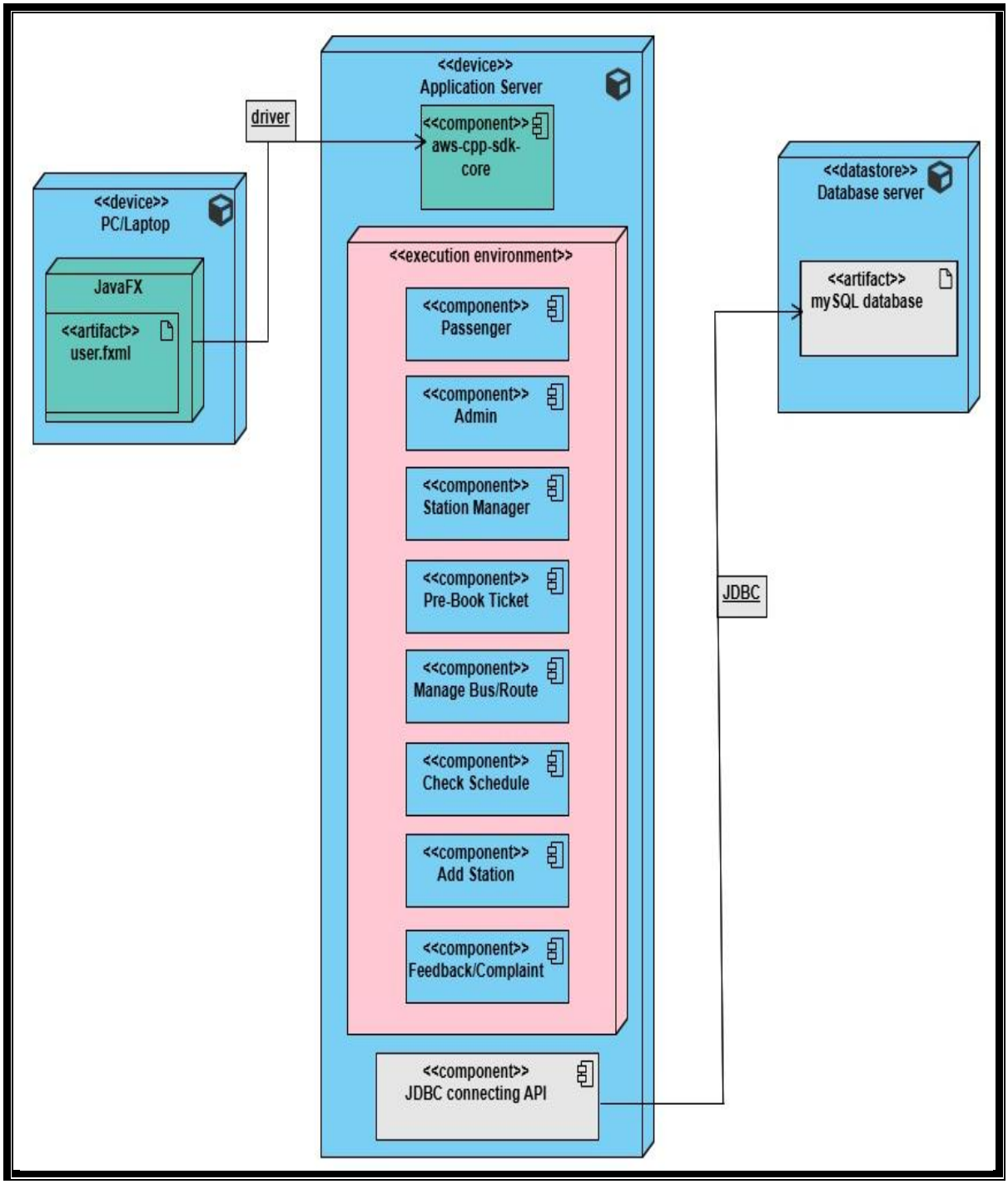
8 Package Diagram



9 Component Diagram



10 Deployment Diagram


<<device>>
Application Server
<<component>>
aws-cpp-sdk-
core
driver
<<device>>
PC/Laptop
JavaFX
<<artifact>>
user.fxml
<<execution environment>>
<<component>>
Passenger
<<component>>
Admin
<<component>>
Station Manager
<<component>>
Pre-Book Ticket
<<component>>
Manage Bus/Route
<<component>>
Check Schedule
<<component>>
Add Station
<<component>>
Feedback/Complaint
<<component>>
JDBC connecting API
JDBC
<<datastore>>
Database server
<<artifact>>
mySQL database

THANK YOU