Software Requirements and Design Document

for

Railway Management System

Fariz Ahmad & Awais Naeem

21i-0552 && 21i-0453

Rail Verge

Table of Contents

1. Introduction

1.1 Purpose

This is the single subsystem in which the customer is allowed to book the tickets online from our portal and the admin has the access to all the details of the train and bookings of all the customers with access to their luggage records.

1.2 Product Scope

- Aims to develop a comprehensive railway management system that streamlines operations, enhances safety, and improves passenger experience for rail networks of varying scales.
- This system will encompass all aspects of railway management, from scheduling and maintenance to ticketing and passenger information.

1.3 Title

The title is "**RailVerge**."

1.4 Objectives

We had to make a desktop application in which data regarding the bookings and information on train schedules is saved and can be accessed easily.
Real-Time Availability: Provide real-time information on train availability, routes, and schedules to help users make informed decisions.
Passenger Experience Improvement: To provide passengers with real-time information about train schedules, delays, and seat availability, making their journey more convenient and enjoyable.
Data Analytics: To gather and analyze data on railway operations, passenger preferences and maintenance needs, enabling data-driven decision-making.
User-Friendly Interface: Create an intuitive and user-friendly interface that allows users of all technical backgrounds to navigate the app effortlessly.
Secure Payments: Implement robust security measures to ensure safe and secure online payments for ticket bookings.
Cancellation and Refunds: Simplify the ticket cancellation process and ensure a smooth refund procedure when necessary. Personalization: Implement user profiles to store preferences, making the booking process more personalized and efficient

1.5 Problem Statement

In Pakistan, the railway ticket booking process is often plagued by inefficiencies, leading to inconvenience for passengers. There is a pressing need for a modern and efficient online railway reservation system to address the following issues:

2. Overall Description

2.1 Product Perspective

Pakistan's railway system plays a crucial role in the country's transportation network .It connects major cities and towns. However, several challenges and inconveniences have persisted within the railway ticket booking process, necessitating the development of an innovative solution.

Manual Booking Hassles:

Traditional ticket booking methods in Pakistan often involve long queues at railway stations , leading to time wastage and discomfort for passengers.

Lack of User-Friendly Interfaces:

Many existing systems like Pakistan Railway, NCS rail, RABTA app lack user-friendly interfaces, making it challenging for passengers to navigate and book tickets online. These existing online booking platforms lead to booking delays, especially during peak travel seasons (specially on Eid holidays).

2.2 Product Functions

- Book ticket
- Select seats
- Print tickets
- Payment
- Cancel ticket
- Tax Payment
- Enter Details
- Check Space
- Book luggage
- Set arrival time
- Set departure time
- Check Train
- Check Station

2.3 List of Use Cases

- **Book Ticket**
- **Cancel Ticket**
- **Register User**
- **Manage Trains**
- **Manage Luggage**
- **Change Seat**
- **Check Fare**
- **Manage Profile**
- **Generate Reports**
- **Handle Complaints**

❖ Book Ticket:

- **Primary Actor:** Passenger
- **Level:** Primary
- **Scope:**
  - Allows passengers to book train tickets online, covering journey selection, payment processing, and ticket issuance.
- **Stakeholders and Interests:**
  - **Passenger:** wants to book tickets for a desired train journey.
  - **Train Operator:** wants to sell tickets and collect revenue from passengers.
  - **Payment Service Provider:** wants to process payments securely and reliably.
- **Preconditions:**
  - The passenger has access to the internet and a web browser.
  - The train operator has a website that allows passengers to search and book tickets online.
- **Post-Conditions:**
  - A booking record is created in the train operator's system with journey details, passenger information, and payment confirmation.
  - The payment service provider sends payment confirmations to the train operator and the passenger.
  - Seats are reserved for the passenger on the selected train, and their availability is updated.
  - E-tickets are generated and sent to the passenger via email or SMS, which can be printed or saved.
  - The passenger receives confirmation of the successful booking and payment, along with the e-tickets.
  - Reserved seats are held temporarily, allowing the passenger to complete the booking. If the passenger cancels or abandons the process, these seats are released.
  - In case of unavailability, payment failure, or passenger cancellation, appropriate error messages are displayed, and the booking process terminates, releasing reserved seats for others to book.
- **Main Success Scenario:**
  - Passenger inputs CNIC, train name, arrival time, destination, and number of seats.
  - The website displays available trains, fares, and seat classes for the specified journey.
  - Passenger selects a train, fare, and seat class, and clicks "Book Now."
  - Redirected to the payment service provider's site, passenger enters payment details and confirms.
  - Payment verified; confirmations sent to train operator and passenger.
  - Train operator issues e-tickets sent via email or SMS.
  - Passenger receives confirmation and can print or save e-tickets.

- **Extension:**
  - No trains are available for the requested journey.
    1. The website displays a message informing the passenger that no trains are available and suggests alternative dates or routes.
    2. The passenger can either modify their search criteria or cancel the booking process.
  - The passenger cancels the payment process or closes the browser window before completing the payment.
    1. The payment service provider notifies the train operator that the payment was not completed.
    2. The train operator releases the reserved seats and cancels the booking.
    3. The passenger can either restart the booking process or abandon it.
  - The payment is declined or fails due to technical or other reasons.
    1. The payment service provider displays an error message to the passenger and asks them to try again or use a different payment method.
    2. The passenger can either retry the payment or cancel the booking process.

❖ Cancel Ticket

- **Primary Actor:** Passenger
- **Level:** Primary
- **Scope:**
  - Allows passengers to cancel their train reservations through the train operator's website or app, considering cancellation policies and refund processing.
- **Stakeholders and Interests:**
  - **Passenger:** wants to cancel their reservation and get a refund if applicable.
  - **Train Operator:** wants to handle cancellations efficiently and retain passenger satisfaction.
  - **Payment Service Provider:** wants to process refunds securely and reliably.
- **Preconditions:**
  - The passenger has booked tickets for a train journey through the train operator's website or app.
  - The train operator has a cancellation policy that specifies the conditions and fees for canceling reservations.
  - The payment service provider has an API that integrates with the train operator's website or app.
- **Post-Conditions:**
  - The passenger's reservation is successfully canceled in the train operator's system.
  - Reserved seats are released and become available for other passengers to book.
  - The payment service provider processes any applicable refunds in accordance with the cancellation policy.
  - Confirmation of the cancellation and, if applicable, refund details are sent to the passenger via email or SMS.
  - The passenger receives the cancellation confirmation and checks their refund status on their bank account or credit card statement.
  - The passenger's booking status is updated to reflect the cancellation, ensuring accurate records are in the train operator's database.
  - Any associated fees or charges are applied as per the cancellation policy and reflected in the passenger's account.
- **Main Success Scenario:**

- Passenger logs into the train operator's website or app.
- Website/app displays bookings; passenger selects one or more to cancel.
- Passenger chooses a booking and clicks "Cancel Booking."
- Website/app shows cancellation policy; passenger confirms, agrees to pay fees if applicable.
- Cancellation request sent to train operator and payment service provider.
- Train operator cancels reservation, releases seats for others.
- Payment service provider processes refund per cancellation policy, sends confirmation.
- Train operator sends cancellation confirmation with refund details via email or SMS.

- Passenger receives cancellation confirmation, checks refund status.

Extension:

- The passenger requests a refund for a non-refundable booking.
- The website or app informs the passenger that the booking is non-refundable and provides details of the cancellation policy.
- The passenger can choose to proceed with the cancellation without a refund or cancel the cancellation request.

❖ Register User

Primary Actor: Passenger

Level: Primary

Scope:

- Allows passengers to book train tickets online, covering journey selection, payment processing, and ticket issuance.

Stakeholders and Interests:

- **Passenger:** wants to book tickets for a desired train journey.
- **Train Operator:** wants to sell tickets and collect revenue from passengers.
- **Payment Service Provider:** wants to process payments securely and reliably.

Preconditions:

- The passenger has access to the internet and a web browser.
- The train operator has a website that allows passengers to search and book tickets online.

Post-Conditions:

- A booking record is created in the train operator's system with journey details, passenger information, and payment confirmation.
- The payment service provider sends payment confirmations to the train operator and the passenger.
- Seats are reserved for the passenger on the selected train, and their availability is updated.
- E-tickets are generated and sent to the passenger via email or SMS, which can be printed or saved.
- The passenger receives confirmation of the successful booking and payment, along with the e-tickets.
- Reserved seats are held temporarily, allowing the passenger to complete the booking. If the passenger cancels or abandons the process, these seats are released.
- In case of unavailability, payment failure, or passenger cancellation, appropriate error messages are displayed, and the booking process terminates, releasing reserved seats for others to book.

Main Success Scenario:

- Passenger visits the train operator's website and selects the "Register" option.
- Passenger inputs their CNIC and creates a password.
- The system validates the uniqueness of the CNIC and the strength of the password.
- Upon successful validation, the passenger's registration is confirmed, and an account is created.
- The registered passenger can now log in with their CNIC and password for future bookings.

Extension:

- CNIC already registered:
  - System prompts the passenger that the CNIC is already associated with an account.
  - Passenger may recover the password or use a different CNIC.
- Weak password:
  - System notifies the passenger that the chosen password does not meet security requirements.
  - Passenger selects a stronger password.

- Unsuccessful registration:
  - System displays an error message if registration fails for any reason.
  - Passenger may try again or contact support for assistance.

❖ Change Seat

- **Primary Actor:** Passenger
- **Level:** Primary
- **Scope:**
  - Allows passengers to assess seat availability for a desired train journey.
- **Stakeholders and Interests:**
  - **Passenger:** wants to check seat availability for a desired train journey before booking tickets.
  - **Train Scheduler:** wants to provide accurate and real-time seat availability information to passengers and encourage them to book tickets online.
- **Preconditions:**
  - The passenger has access to the internet and a web browser.
  - The train scheduler has a website that allows passengers to search and check seat availability online.
  - The train scheduler has a database that stores seat availability information for each train schedule.
- **Post-Conditions:**
  - The passenger receives real-time seat availability information for the selected journey, including details of available seats, waitlisted status, or sold-out status.
  - The passenger's seat selection, if made during the booking process, may need to be adjusted if seat availability changes due to simultaneous bookings.
  - Passengers can proceed to book tickets, modify their search criteria, choose different seats, select alternative trains, or cancel the booking process based on the seat availability information provided.
- **Main Success Scenario:**

  - Passenger, having initiated the booking process, decides to modify their seat selection.
  - Passenger inputs their CNIC, indicating the desire to change seats for the chosen journey.
  - The system validates the request and confirms the current seat selection.
  - Passenger specifies the new seat preference.
  - The system checks the real-time availability of the new seat and updates the information accordingly.
  - Passenger receives confirmation of the successful seat change.
  - Passenger continues with the booking process, selecting other options or confirming the changes.

- **Extension:**

  - Unavailable new seat:
    - If the new seat is not available due to simultaneous bookings, the system informs the passenger.
    - Passenger can choose an alternative seat or retain the current selection.
  - Cancellation of seat change:
    - At any point during the process, the passenger may opt to cancel the seat change.
    - The system reverts to the previous seat selection, and the passenger continues with the booking process.

❖ Check Fare

- **Primary Actor:** Passenger
- **Level:** Primary
- **Scope:**

- Allows passengers to determine the cost of a desired train journey before booking tickets through the website.

- **Stakeholders and Interests:**
  - **Passenger:** wants to calculate fares for a desired train journey before booking tickets.
  - **Train Scheduler:** wants to provide accurate and transparent fare information to passengers and encourage them to book tickets online.
- **Preconditions:**
  - The passenger has access to the internet and a web browser.
  - The train scheduler has a website that allows passengers to search and calculate fares online.
  - The train scheduler has a fare calculation algorithm that considers various factors such as distance, seat class, demand, taxes, discounts, etc.
- **Main Success Scenario:**

  - Passenger navigates to the train scheduler's website.
  - Website displays a "Check Fare" button.
  - Passenger clicks the button.
  - The website presents a concise list of available trains showing Train Name, Arrival, Destination, and Fare.
  - Passengers can review the information and proceed to book tickets or explore other options.

- **Extension:**
  - None, as this is a straightforward view-only feature without further user interaction.

❖ Manage Profile

- **Primary Actor:** Passenger
- **Level:** Primary
- **Scope:**
  - This use case allows passengers to manage their profile information.
- **Stakeholders and Interests:**
  - **Passenger:** Interested in updating personal information, password, or contact details
  - **Train Scheduler:** Ensures passenger profiles are accurate and up to date.
- **Preconditions:**
  - The passenger is logged into the system(profile).
- **Postconditions:**
  - The passenger's profile information is updated.
- **Main Success Scenario:**

  - Passenger visits the train scheduler's website.
  - Website presents a "Register" button.
  - Passenger clicks the button to initiate the registration process.
  - Passenger inputs CNIC, Name, Password, and Phone Number.
  - System validates the entered information.
  - If valid, the system creates a new account for the passenger.
  - Passenger receives a confirmation message that registration is successful.
  - The registered passenger can now log in using their CNIC and password for future interactions.

- **Extension:**
  - If the CNIC is already associated with an existing account
  - System prompts the passenger that the CNIC is already registered.

❖ Manage Trains

- **Primary Actor:** Train Scheduler
- **Level:** Primary
- **Scope:** Allows train schedulers to manage train-related information.
- **Stakeholders and Interests:**
  - **Train Scheduler:** Interested in updating train schedules, adding new trains, or modifying existing train details.
  - **Railway Company:** Ensures accurate and up-to-date train information.
- **Preconditions:**
  - The train scheduler has access to the system for managing train information.
  - The train scheduler is logged into the system(profile).
  - The train scheduler maintains a database containing train details such as schedules, routes, stations, fares, seat classes, etc.
- **Postconditions:**
  - If the train scheduler successfully creates or updates a train record, the database contains the new or modified train record with accurate details, including route, stations, timings, fares, seat classes, etc.
  - If the train scheduler successfully deletes a train record and confirms the deletion, the database no longer contains the deleted train record.
  - After viewing an existing train record, the train scheduler has access to detailed information about the selected train, including its route, stations, timings, fares, seat classes, etc.
  - The train scheduler receives confirmation messages after each action (create, update, delete, view), ensuring that their requested actions have been processed.
  - If the train scheduler chooses to delete a train record associated with active bookings, the system cancels the linked bookings as well, ensuring that passengers are informed of the cancellations.
  - Train information is updated in the system.
- **Main Success Scenario:**
  - The train scheduler logs into the system and accesses the train information management module.
  - The train scheduler selects the option to manage train information.
  - The system displays existing train data and allows the train scheduler to execute various actions, including creating, updating, deleting, or viewing train information.
  - The train scheduler can perform the following actions:
    1. Create a new train record by inputting required information and clicking "Create Train." The system validates the data and adds it to the database.
    2. Update an existing train record by selecting it, modifying details, and clicking "Update Train." The system validates changes and updates the record.
    3. Delete an existing train record by selecting it, clicking "Delete Train," and confirming. The system removes the train record.
    4. View an existing train record by selecting it and clicking "View Train." The system displays the train's details.
    5. Confirmation messages are provided after each action, allowing the train scheduler to continue or log out.
  - The train scheduler makes desired updates.
  - System saves the changes.

- **Extension:**
  - Train Scheduler attempts to update a record with conflicting information.
    1. The train scheduler attempts to modify a train record with data that conflicts with existing records.
    2. The system displays an error message indicating the conflict and asks the train scheduler to adjust the data.
    3. The train scheduler revises the data and proceeds with the update.
  - Train Scheduler tries to delete a record associated with active bookings.

1. The train scheduler attempts to delete a train record linked to active bookings.
2. The system issues a warning message, informing the train scheduler of the active bookings associated with the record.
3. The train scheduler can choose to continue with the deletion, which also cancels the linked bookings, or cancel the deletion request.

❖ Generate Reports

- **Primary Actor:** Train Operator
- **Level:** Primary
- **Scope:**
  - This use case allows train operators to generate various reports related to reservations and revenue.
- **Stakeholders and Interests:**
  Train Operator: Interested in generating reports for business analysis.
  Railway Company: Uses reports for decision-making and financial planning.
- **Preconditions:**
  - The Train Operator is logged into the system.
- **Postconditions:**
  - Reports are generated and available for download.
- **Main Success Scenario:**
  - Train Operator selects the type of report to generate booking details..
  - System generates the selected report.
  - Train Operator downloads the report.
- **Extension:**
  - None

❖ Handle Complaints

- **Primary Actor:** Train Operator
- **Level:** Primary
- **Scope:**
  - This use case involves handling complaints from passengers.
- **Stakeholders and Interests:**
  Passenger: Interested in specifying special requests for their journey.
  Train Operator: Ensures passenger requests are accommodated.
- **Preconditions:**
  - The train operator has a system in place for receiving and managing passenger complaints.
  - Passengers have the means to submit complaints through various channels, such as email, phone, or an online portal.
- **Postconditions:**
  - The special request is noted and forwarded to the relevant departments for implementation.
- **Main Success Scenario:**
  - A passenger submits a complaint through one of the available channels, providing details about their issue and contact information.
  - The complaint is logged into the system, assigning it a unique reference number.
  - A passenger support representative logs into the complaint management system and views the list of pending complaints.
  - The representative selects a complaint from the list to handle.
  - The representative reviews the complaint details, including the passenger's description of the issue.
  - The system generates and sends a confirmation message to the passenger, acknowledging the receipt and resolution of their complaint.
- **Extension:**
  - If the request cannot be accommodated due to operational constraints, the system informs the passenger and offers alternatives if possible.

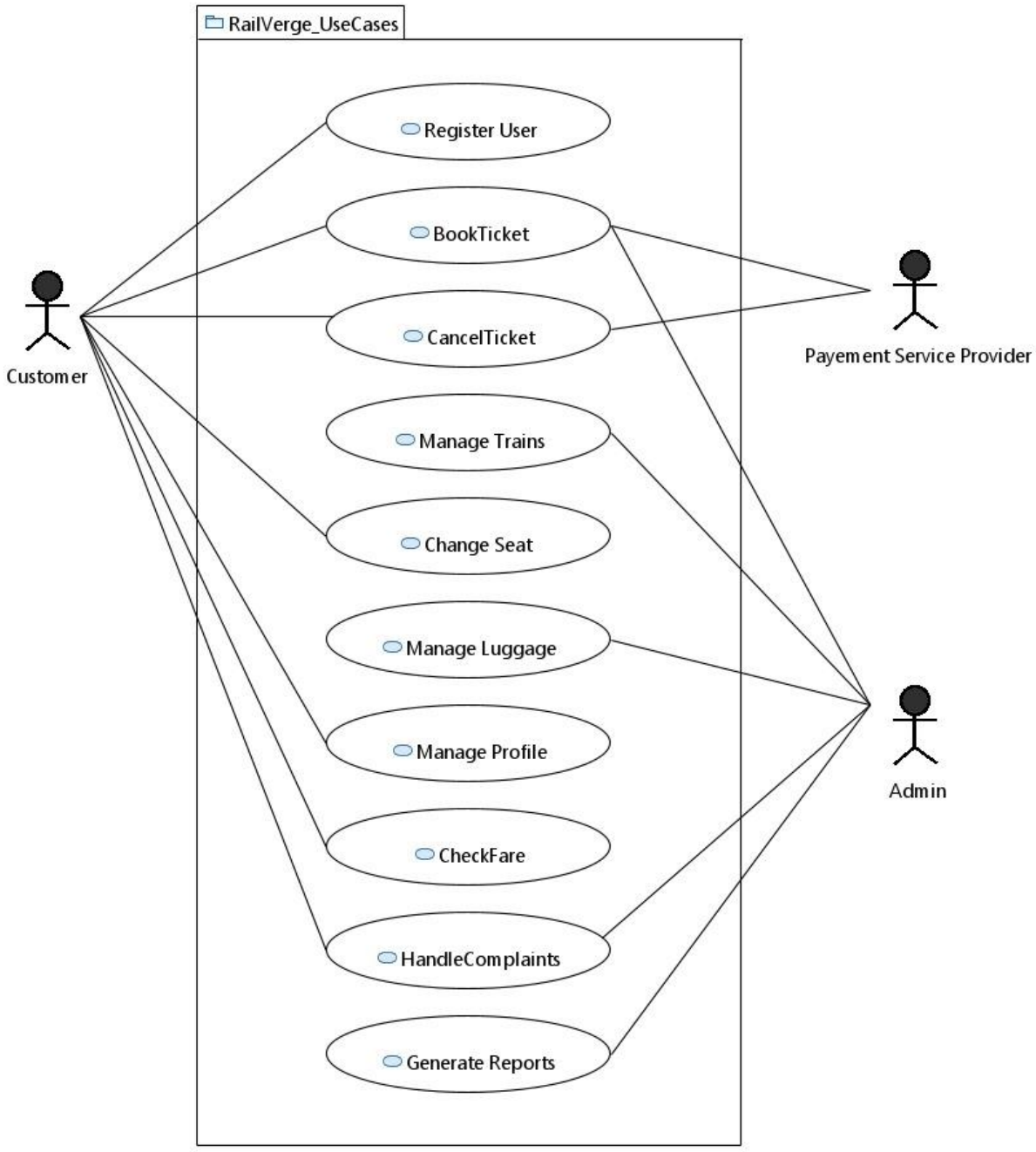
❖ Manage Luggage

- **Primary Actor:** Passenger
- **Level:** Primary
- **Scope:**
  - Allows train scheduler to handle passenger’s luggage.
- **Stakeholders and Interests:**
  - **Passenger:** To take care of the luggage. And the luggage is received by each passenger.
  - **Train Scheduler:** Aims to efficiently manage and transport passengers' luggage while maintaining security and timeliness.
- **Preconditions:**
  - The passenger is traveling that involves luggage.
  - The train scheduler has procedures and systems for managing luggage.
  - Check if the train storage is full or not to load the luggage.
- **Postconditions:**
  - Luggage is tagged and registered with a unique identifier in the transportation provider's system.
  - Every passenger takes their luggage according to their name.
- **Main Success Scenario:**

  - Passenger approaches the luggage check-in counter.
  - Passenger provides necessary identification and travel information to the train scheduler.
  - Train scheduler assigns appropriate names to each luggage based on the passenger's information.
  - Luggage is tagged and registered with a unique identifier in the transportation provider's system.
  - Train scheduler loads the luggage onto the train, ensuring it aligns with the passenger's travel details.
  - Passenger receives the luggage upon arrival, matching the assigned names.

- **Extension:**
  - If the passenger’s luggage is mishandled or delayed (e.g., lost, damaged, or not available upon arrival), the transportation service provider initiates a luggage search and recovery process to locate and deliver the luggage to the passenger.
  - There is not enough space to carry the luggage of some passengers.

2.4 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

3.2 Safety Requirements

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.

3.3 Security Requirements

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

3.4 Software Quality Attributes

This application not only allows customers to book their tickets but also maintains record of them which is accessible at any time. Further, the software is not bound which allows more boundary space thus resulting in more quality. And as it has always been, quality over quantity. That's the thing we have focused on in this project and we are sure we have got where a software can be assured without any hindrance a quality product.

3.5 Business Rules

Only the admin can manage the bookings and have access to all the details whereas the customer can only look up his own bookings and their own details.

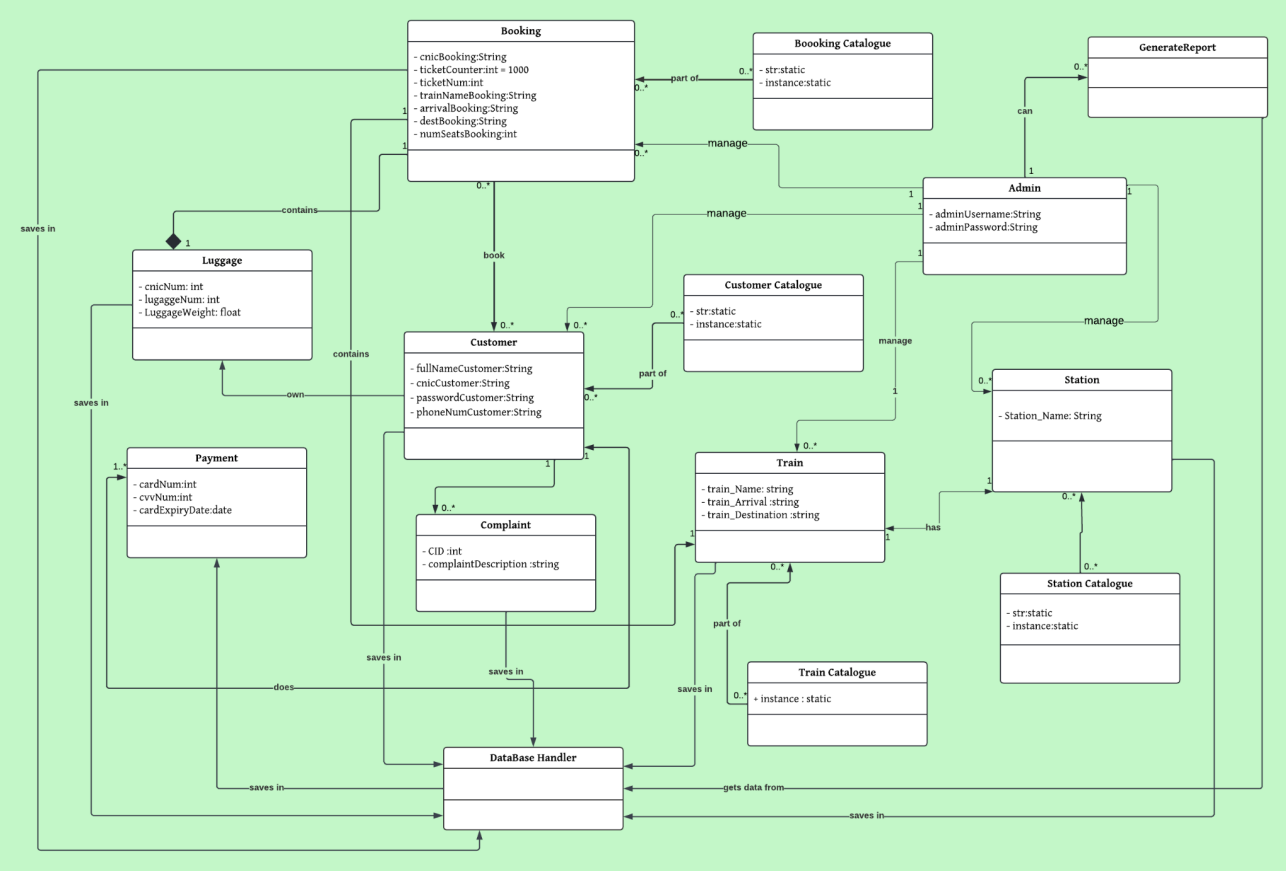
3.6 Operating Environment

This desktop application can work on any operating system if that operating system contains the needed IDE to run the program. Further, the database (more useful if it is MySQL) has to be installed and connected and the paths provided have to be changed according to the hardware used.

3.7 User Interfaces

There is a homepage that allows the user to choose between an admin or a customer depending upon who you are. Further, the customer has to register themselves in order to book a ticket and they also have the option to choose their own train and seat according to their preference. On the other hand, the admin has access to all the bookings of all the customers and they can also change trains and their schedules depending upon the needs of customers.

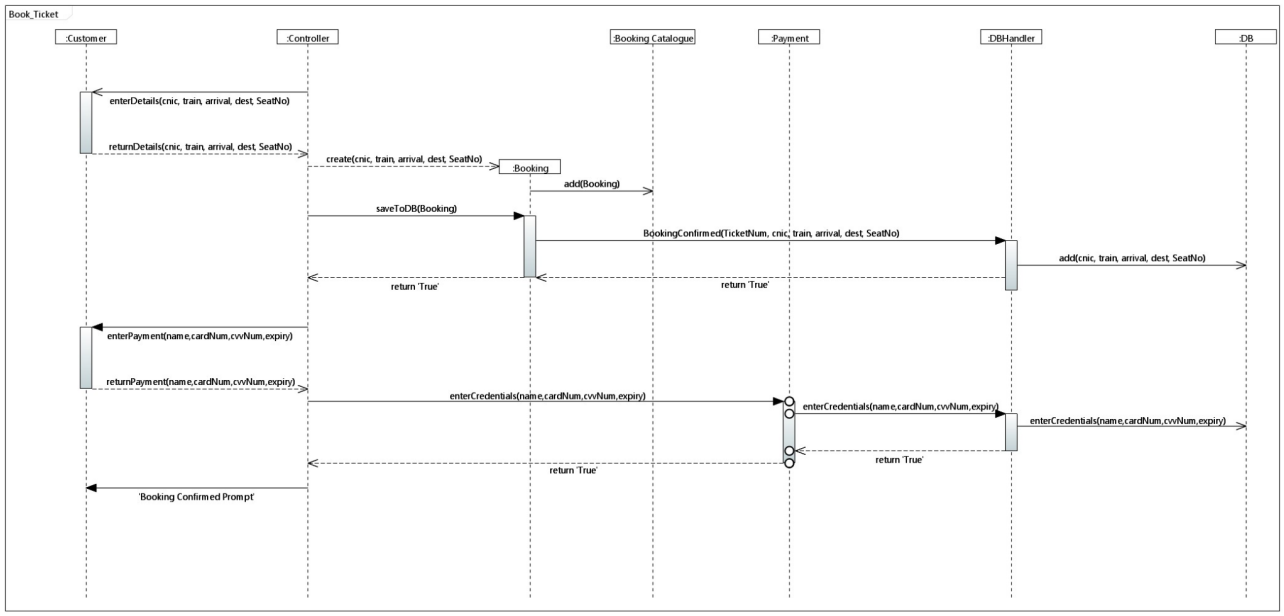
4. Domain Model

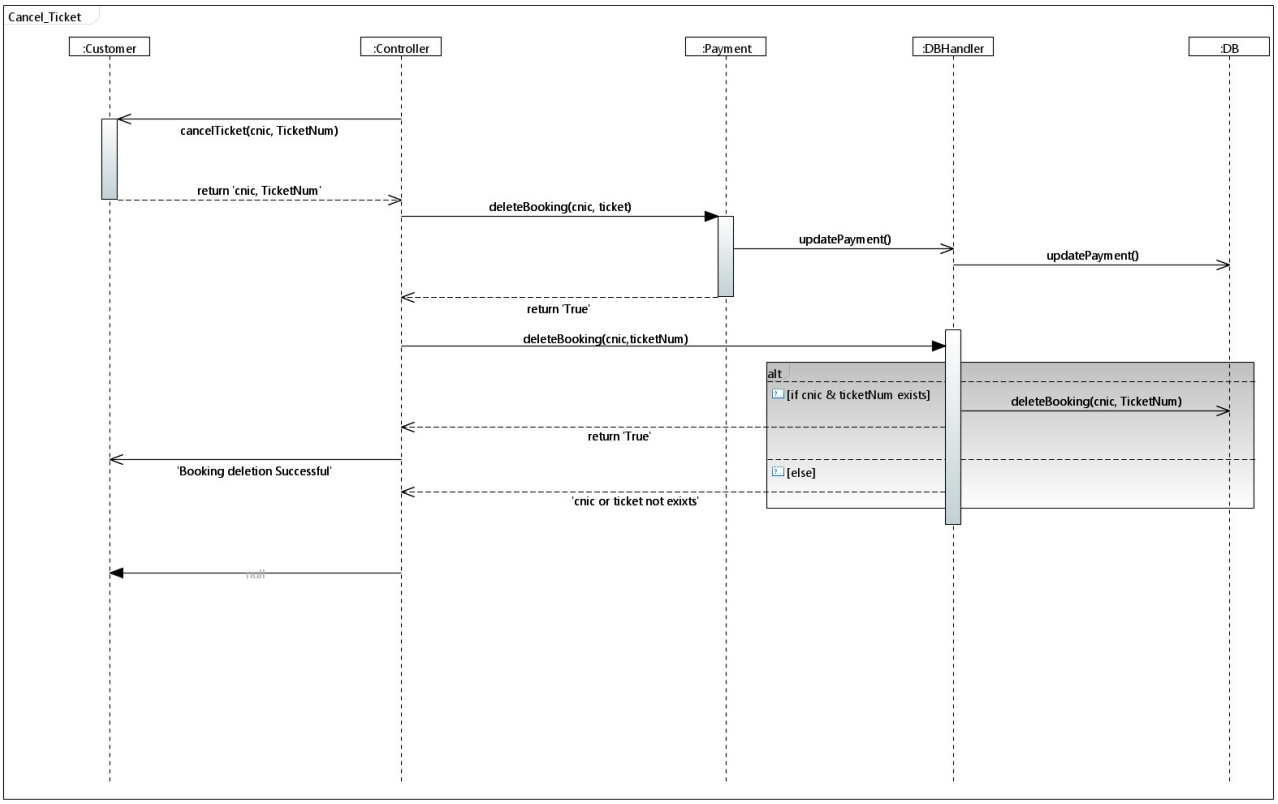


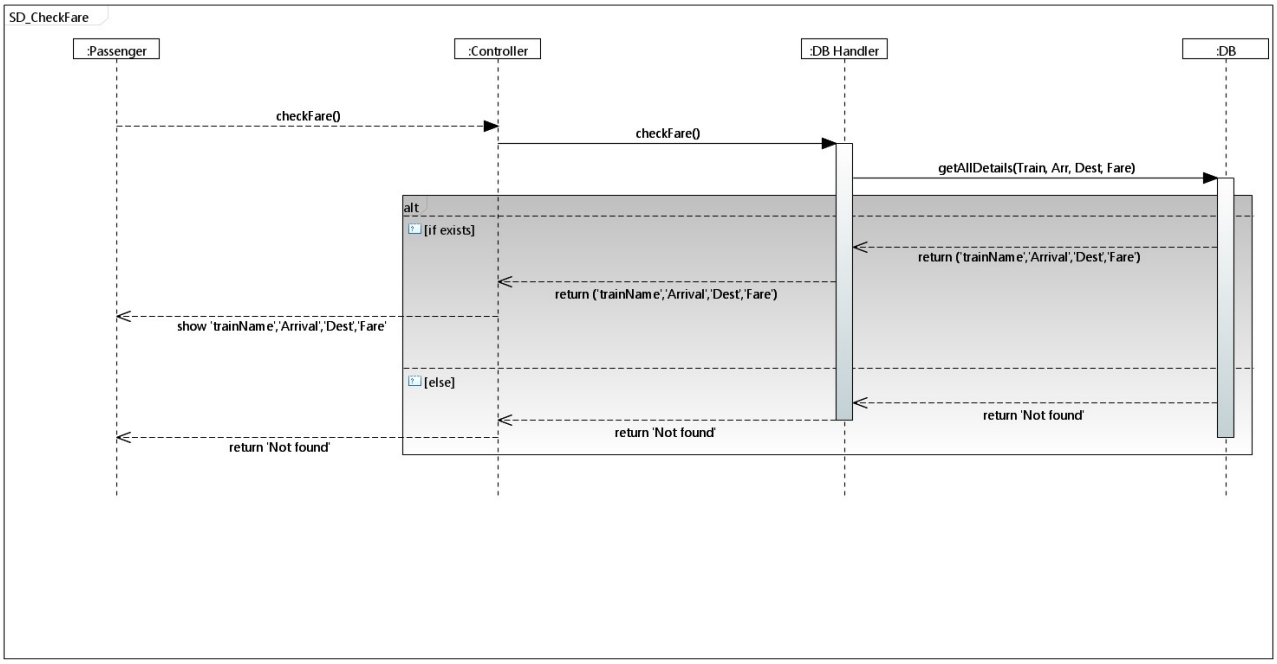
5. System Sequence Diagram

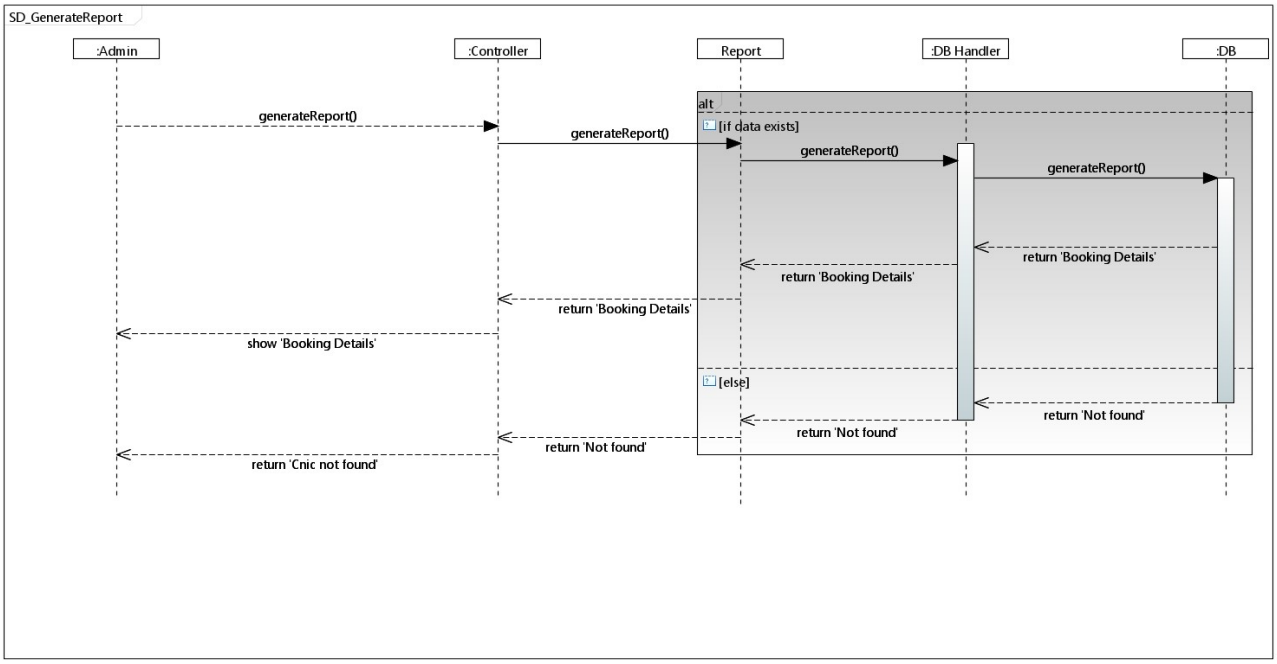
5.1 DIAGRAMS

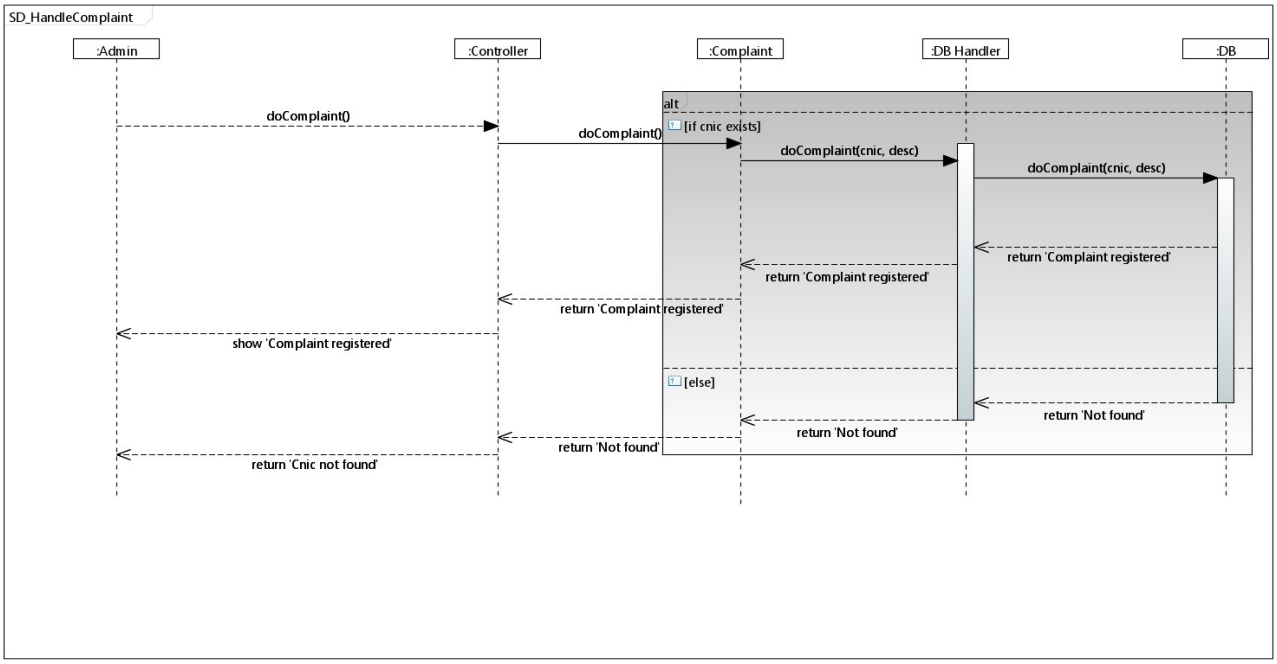
6. Sequence Diagram

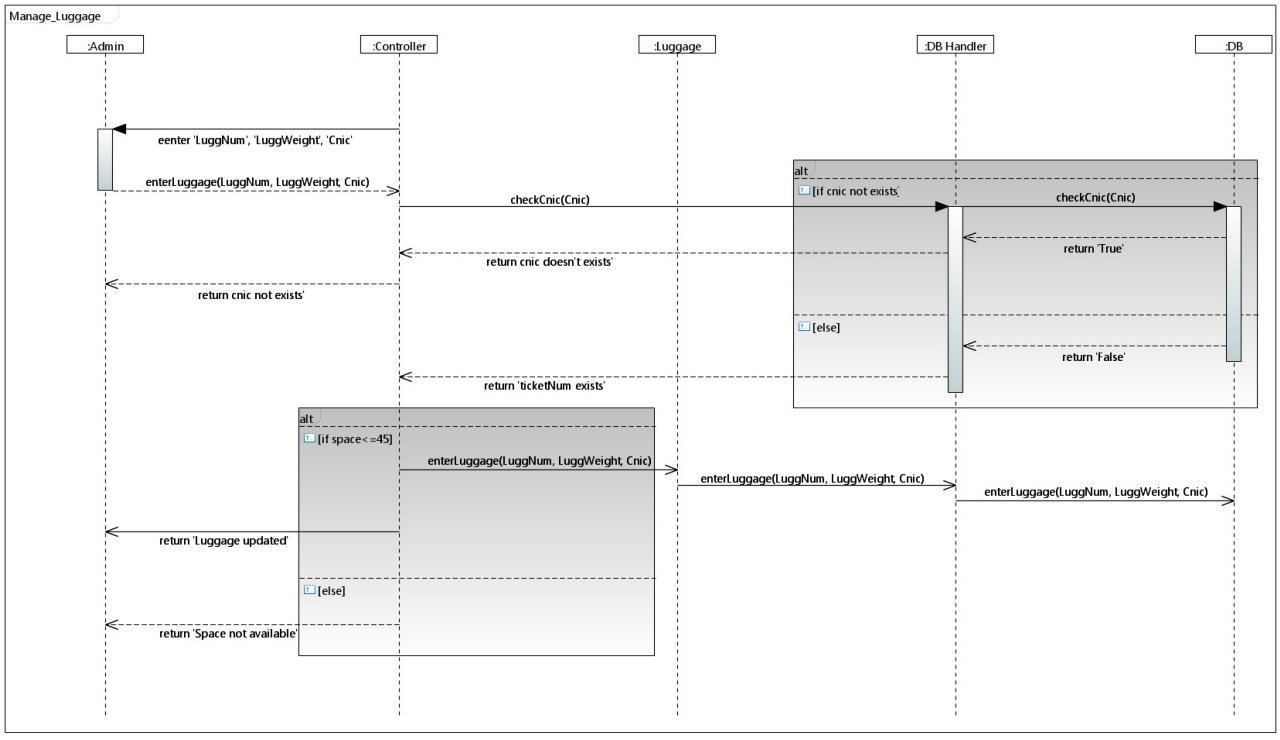


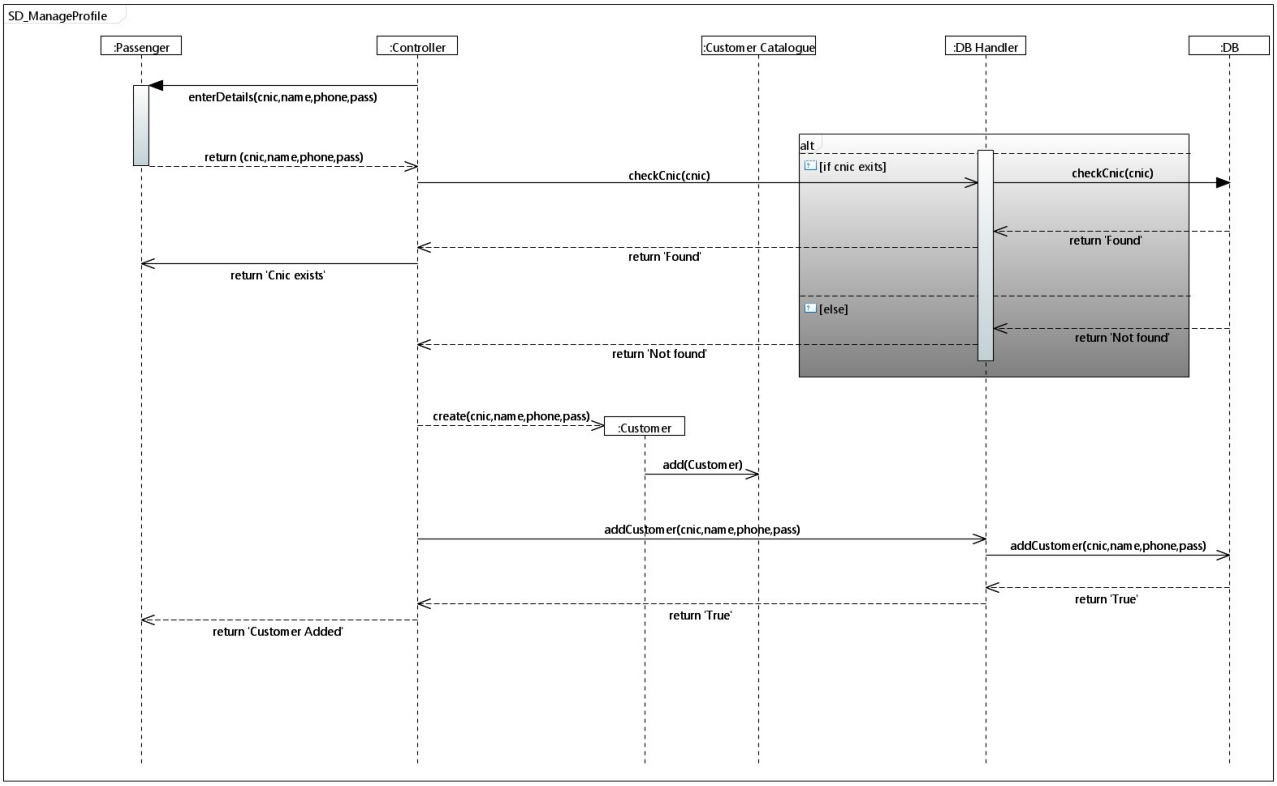


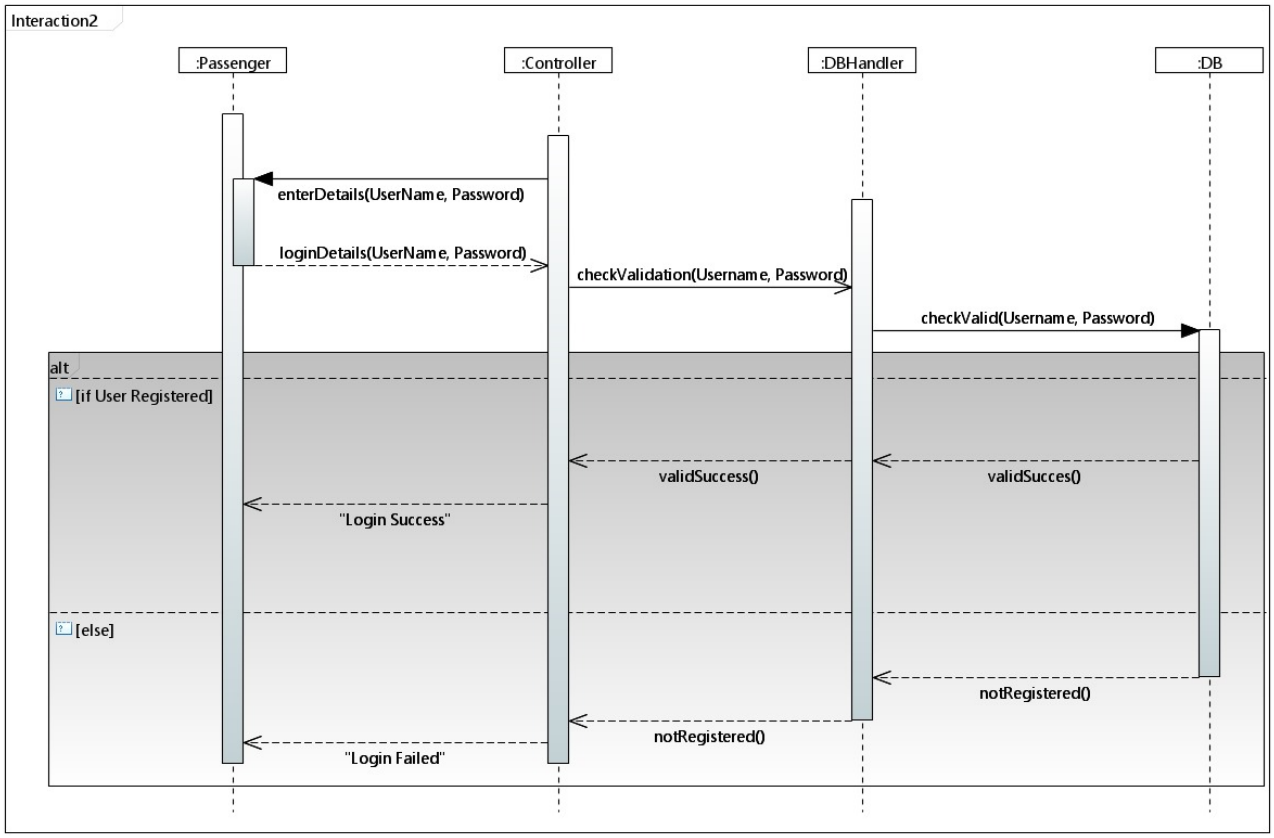

SD_CheckFare
:Passenger
:Controller
:DB Handler
:DB
checkFare()
checkFare()
getAllDetails(Train, Arr, Dest, Fare)
alt
[if exists]
return ('trainName','Arrival','Dest','Fare')
return ('trainName','Arrival','Dest','Fare')
show 'trainName','Arrival','Dest','Fare'
[else]
return 'Not found'
return 'Not found'
return 'Not found'

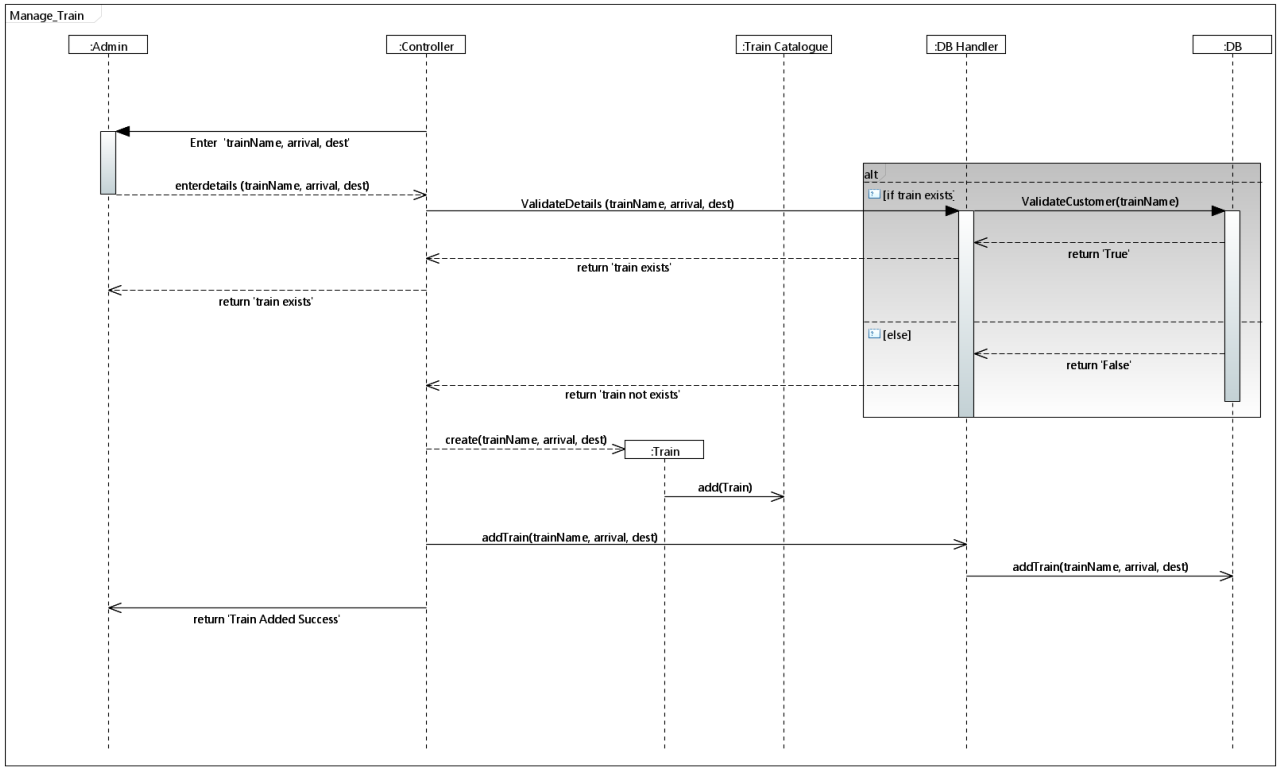

SD_GenerateReport
:Admin
:Controller
Report
:DB Handler
:DB
generateReport()
generateReport()
generateReport()
generateReport()
alt
[if data exists]
return 'Booking Details'
return 'Booking Details'
return 'Booking Details'
show 'Booking Details'
[else]
return 'Not found'
return 'Not found'
return 'Not found'
return 'Cnic not found'

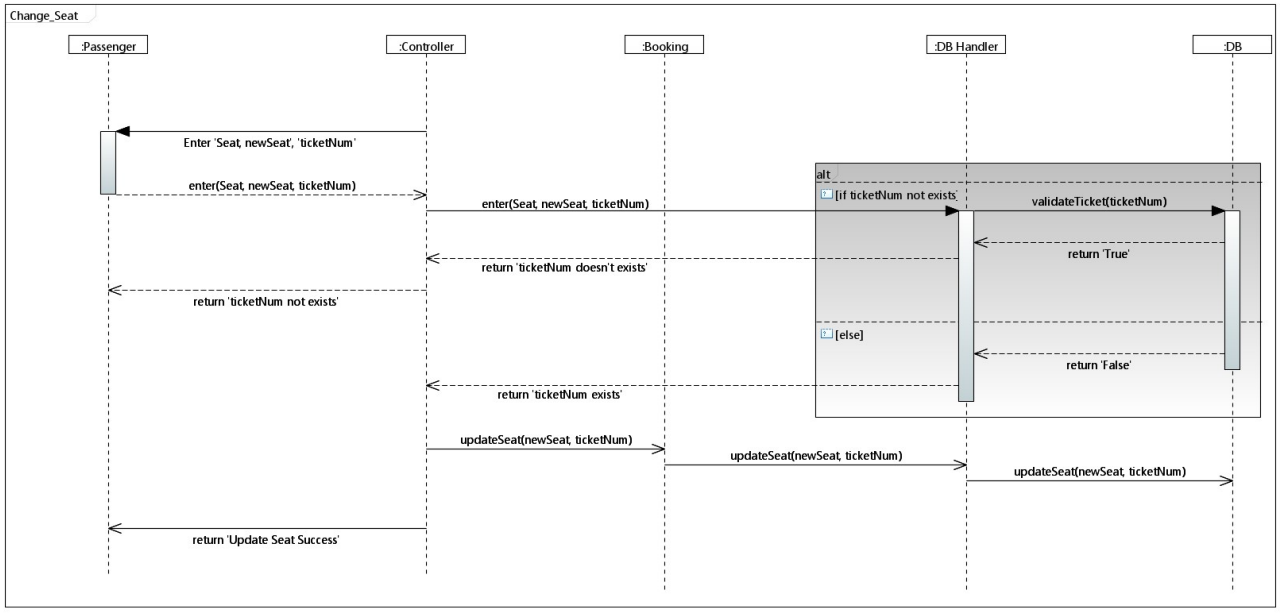

SD_HandleComplaint
:Admin
:Controller
:Complaint
:DB Handler
:DB
alt
[if cnic exists]
doComplaint()
doComplaint()
doComplaint(cnic, desc)
doComplaint(cnic, desc)
return 'Complaint registered'
return 'Complaint registered'
return 'Complaint registered'
show 'Complaint registered'
[else]
return 'Not found'
return 'Not found'
return 'Not found'
return 'Cnic not found'


Manage_Luggage
:Admin
:Controller
:Luggage
:DB Handler
:DB
eenter 'LuggNum', 'LuggWeight', 'Cnic'
enterLuggage(LuggNum, LuggWeight, Cnic)
alt
[if cnic not exists]
checkCnic(Cnic)
checkCnic(Cnic)
return 'True'
return cnic doesn't exists'
return cnic not exists'
[else]
return 'False'
return 'ticketNum exists'
alt
[if space<=45]
enterLuggage(LuggNum, LuggWeight, Cnic)
enterLuggage(LuggNum, LuggWeight, Cnic)
enterLuggage(LuggNum, LuggWeight, Cnic)
return 'Luggage updated'
[else]
return 'Space not available'


SD_ManageProfile
:Passenger
:Controller
:Customer Catalogue
:DB Handler
:DB
enterDetails(cnic,name,phone,pass)
return (cnic,name,phone,pass)
checkCnic(cnic)
alt
[if cnic exits]
checkCnic(cnic)
return 'Found'
return 'Found'
return 'Cnic exists'
[else]
return 'Not found'
return 'Not found'
create(cnic,name,phone,pass)
:Customer
add(Customer)
addCustomer(cnic,name,phone,pass)
addCustomer(cnic,name,phone,pass)
return 'True'
return 'True'
return 'Customer Added'


Interaction2
:Passenger
:Controller
:DBHandler
:DB
enterDetails(UserName, Password)
loginDetails(UserName, Password)
checkValidation(Username, Password)
checkValid(Username, Password)
alt
[if User Registered]
validSuccess()
validSucces()
"Login Success"
[else]
notRegistered()
notRegistered()
"Login Failed"


Manage_Train
:Admin
:Controller
:Train Catalogue
:DB Handler
:DB
Enter 'trainName, arrival, dest'
enterdetails (trainName, arrival, dest)
alt
[if train exists]
ValidateDetails (trainName, arrival, dest)
ValidateCustomer(trainName)
return 'True'
return 'train exists'
return 'train exists'
[else]
return 'False'
return 'train not exists'
create(trainName, arrival, dest)
:Train
add(Train)
addTrain(trainName, arrival, dest)
addTrain(trainName, arrival, dest)
return 'Train Added Success'


Change_Seat
:Passenger
:Controller
:Booking
:DB Handler
:DB
Enter 'Seat, newSeat', 'ticketNum'
enter(Seat, newSeat, ticketNum)
alt
[if ticketNum not exists]
enter(Seat, newSeat, ticketNum)
validateTicket(ticketNum)
return 'True'
return 'ticketNum doesn't exists'
return 'ticketNum not exists'
[else]
return 'False'
return 'ticketNum exists'
updateSeat(newSeat, ticketNum)
updateSeat(newSeat, ticketNum)
updateSeat(newSeat, ticketNum)
return 'Update Seat Success'

7. Class Diagram