Test Case ID	Scenario Name and Description	INPUT 1 Train no	INPUT 2 User Details	INPUT 3 Seat category	Expected Output	Remarks
TC_1	Scenario 1- Book ticket	Valid	Valid	Valid	Redirected for payment	Total amount
TC_2	Scenario 2- Book ticket Alternate Flow: Invalid user details (age,phone no)	valid	Invalid	valid	Invalid passen ger details	Age is no not in the specified format
TC_3	Invalid Train no	Invalid	Valid/i nvalid	-	No such train available	Enter train no present in the database
TC_4		Valid	Valid	Invalid	This seat already booked	Select vacant seats only
TC_5	Scenario 3- Book ticket Alternate Flow: User Exits	*	*	*	User comes out of the system	-

EXPERIMENT 9: TEST CASE GENERATION DIAGRAM

AIM: To design the Test Cases for Railway Reservation System

Theory:

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

Software testing plays an important role in software development because it can minimize the development cost. Software testing approaches are divided into three parts i.e. code-based testing, specification-based testing, and model-based testing. In model-based testing, the testing begins at the design phase. So, early detection of faults can be achieved by using this approach further reducing the time, cost, and efforts of the developer to a large extent. In this paper, we propose a technique for generating test cases using both a UML sequence diagram and a state chart diagram.

EXPERIMENT 8: ACTIVITY DIAGRAM

AIM: To design the Activity Diagram for Railway Reservation System

Theory:

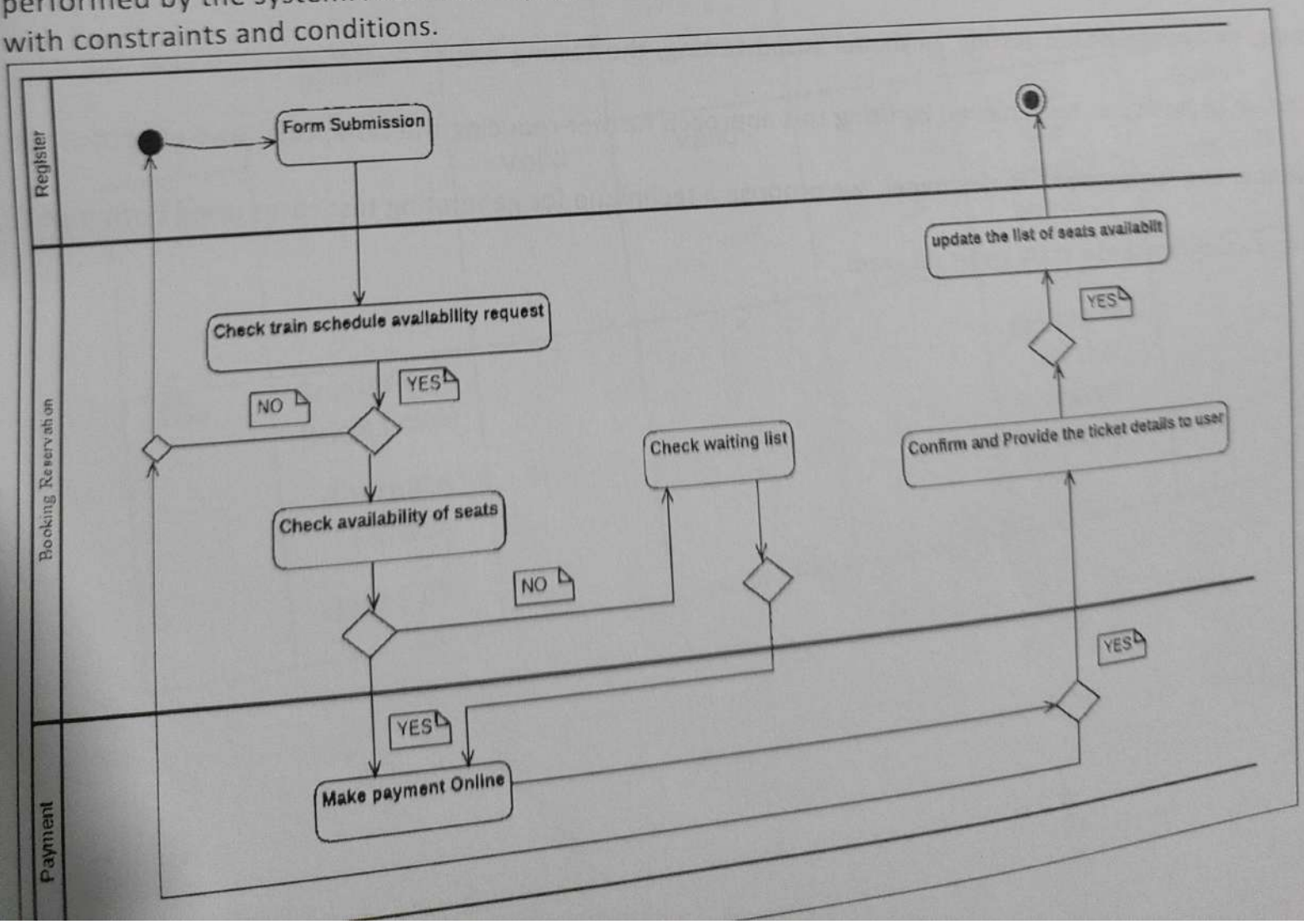
Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

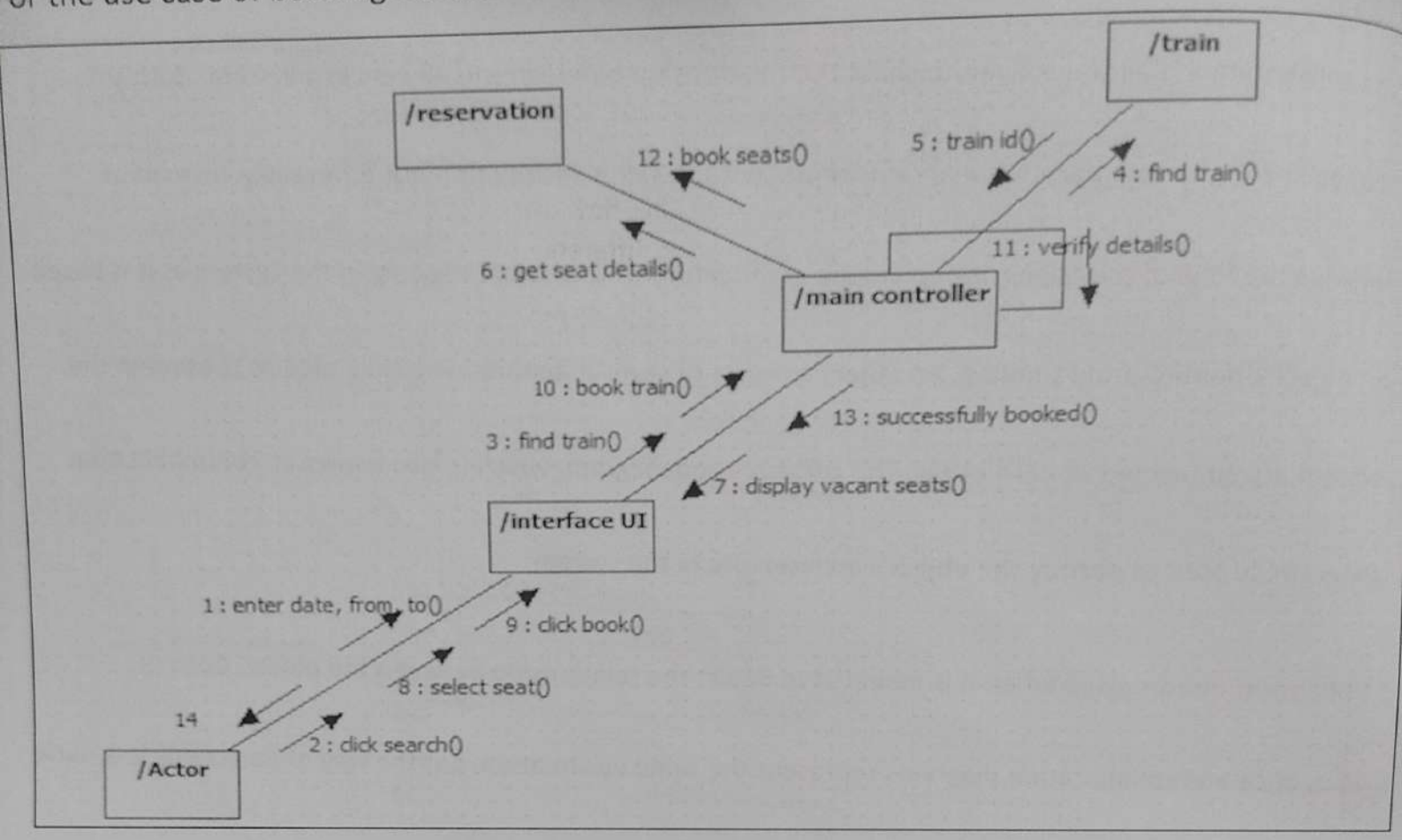
The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc.

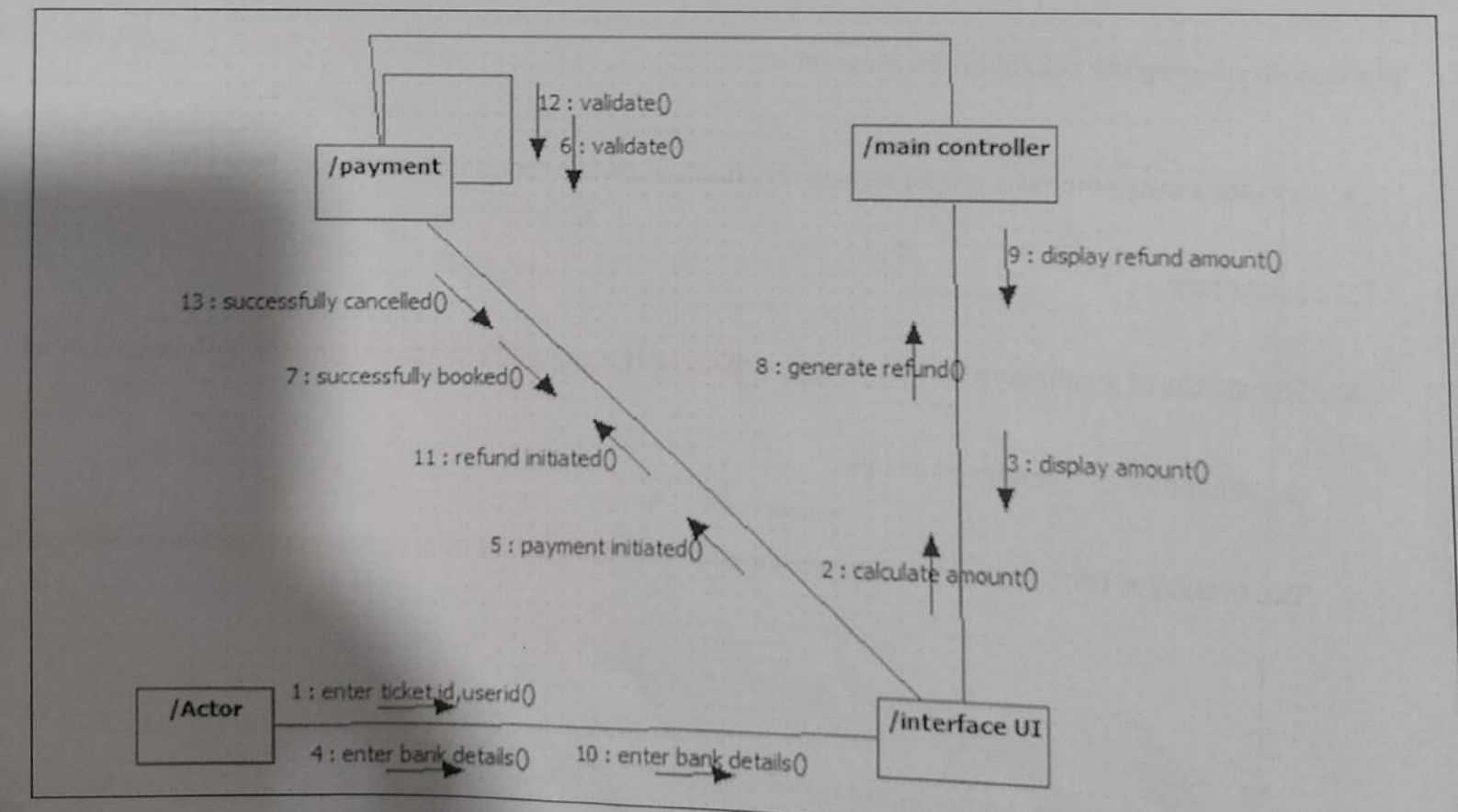
Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.



For the use case of booking a seat:



For the use case of payment:



EXPERIMENT 7: COLLABORATION DIAGRAM

AIM: To design the Collaboration Diagram for Railway Reservation System

Theory:

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as communication diagram, is used to portray the object's architecture in the system.

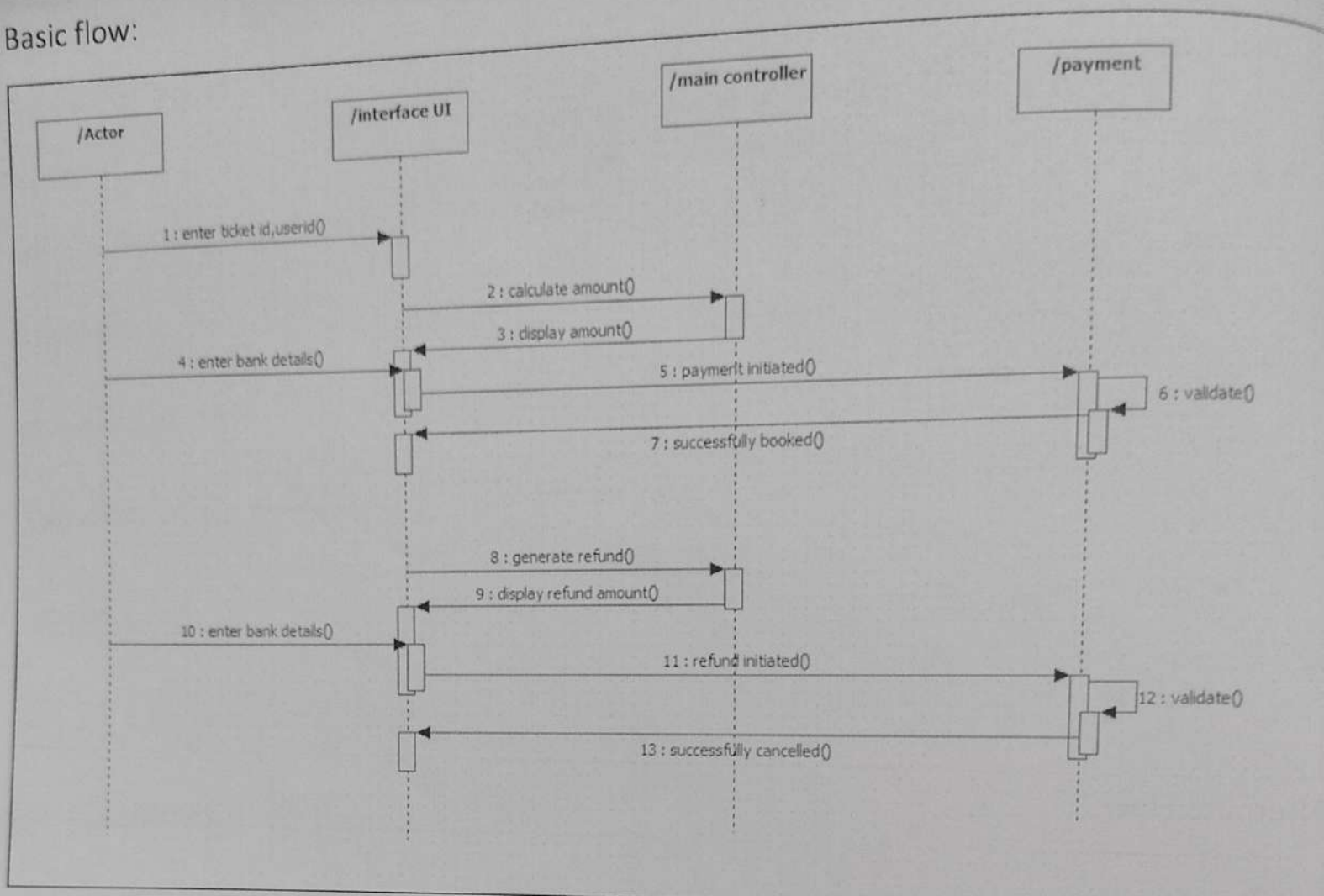
Collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it is quite different. The collaboration diagrams are best suited for analyzing use cases.

The benefit of using the collaboration diagram is:

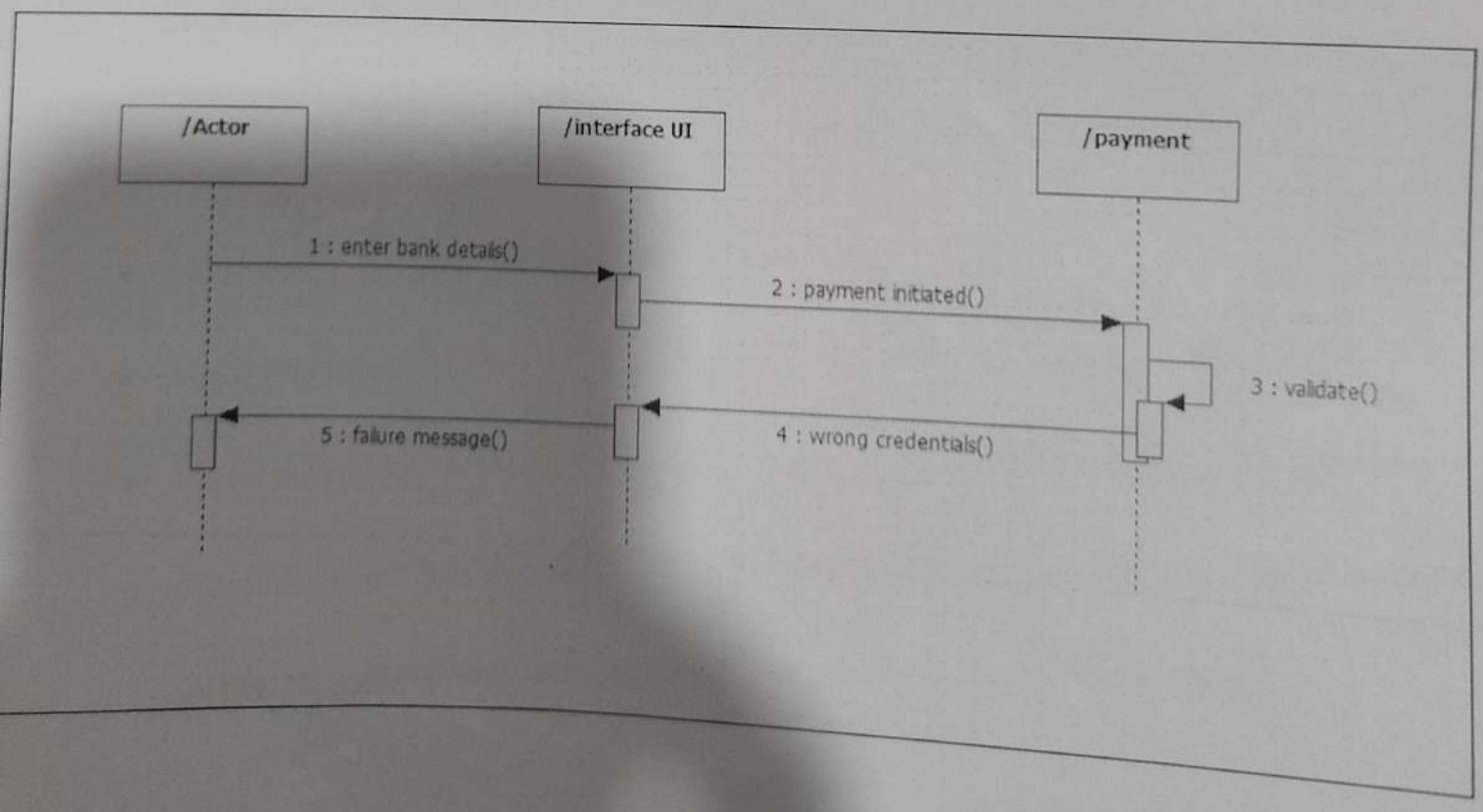
- It mainly puts emphasis on the structural aspect of an interaction diagram, i.e., how lifelines are connected.
- The syntax of a collaboration diagram is similar to the sequence diagram; just the difference is that the lifeline does not consist of tails.
- The messages transmitted over sequencing are represented by numbering each individual message.

For the use case of payment:

Basic flow:

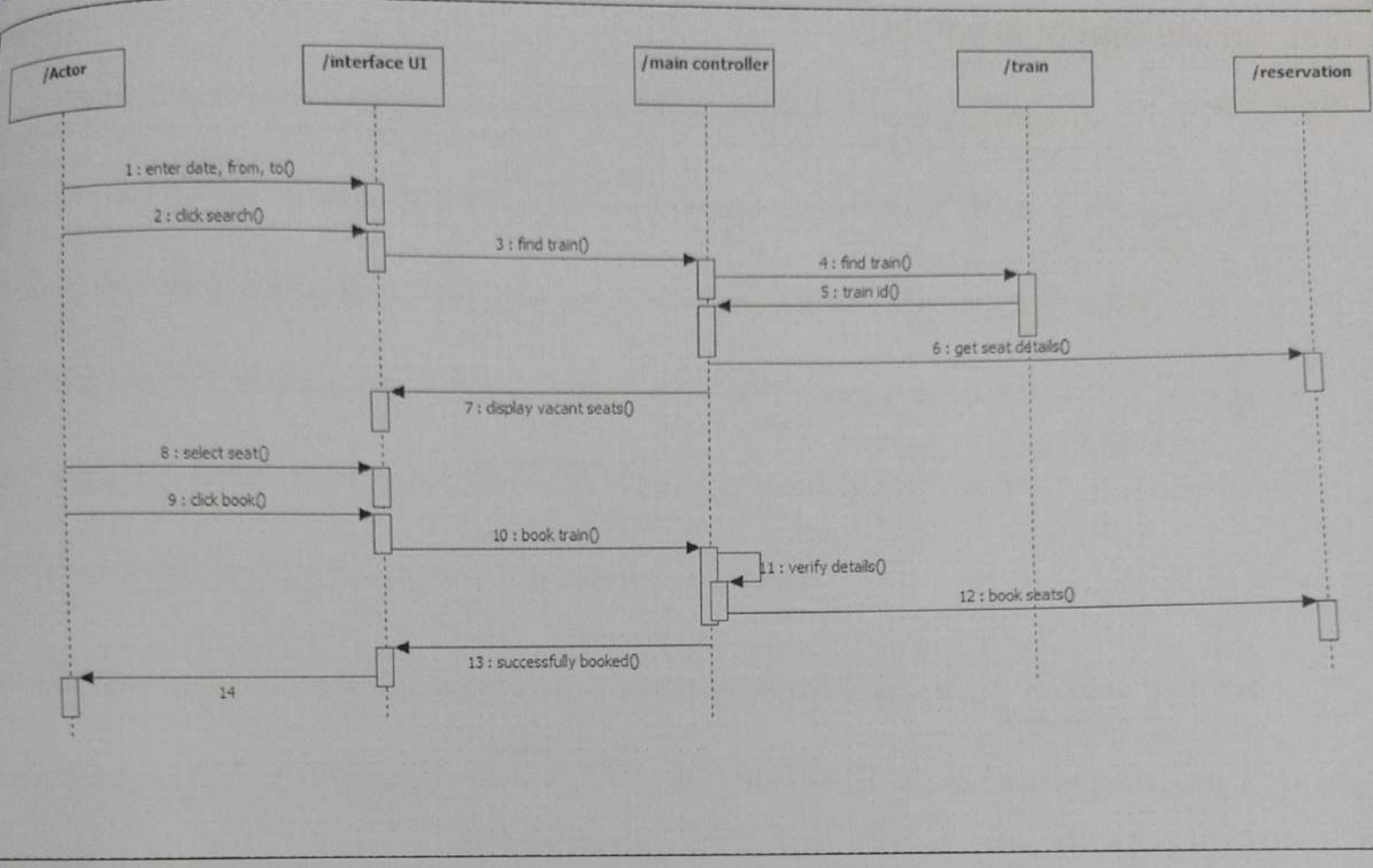


Alternate Flow:

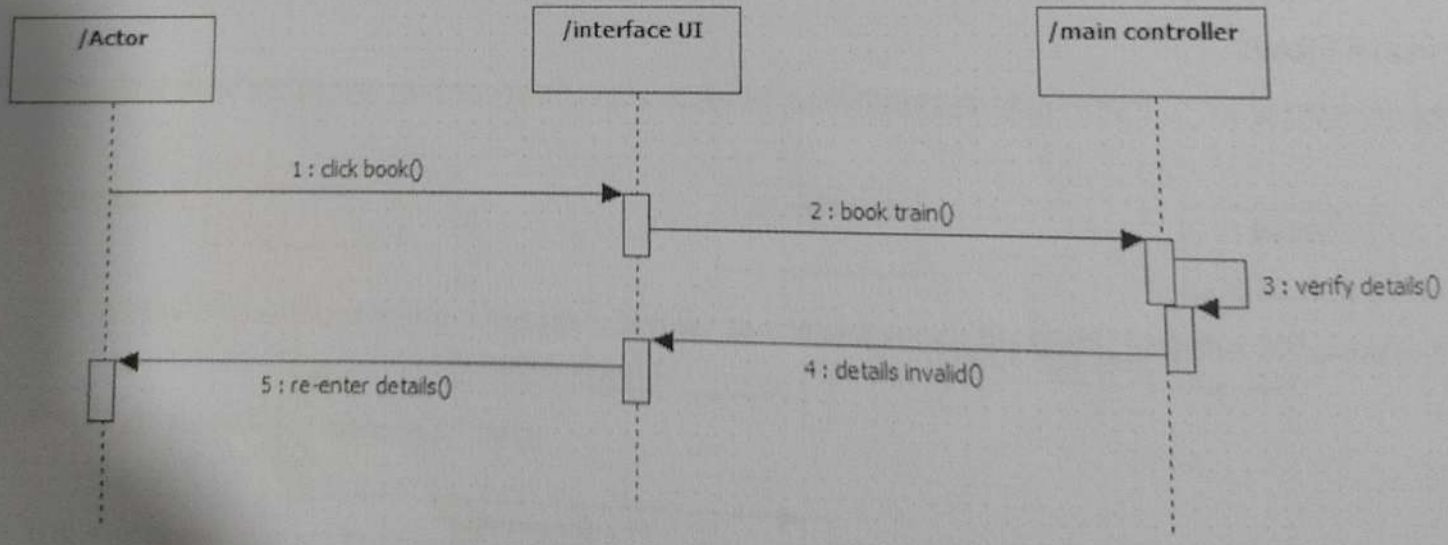


For the use case of booking a seat:

Basic flow:



Alternate Flow:



EXPERIMENT 6: SEQUENCE DIAGRAM

AIM: To design the Sequence Diagram for Railway Reservation System

Theory:

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time-focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent the time what messages are sent and when.

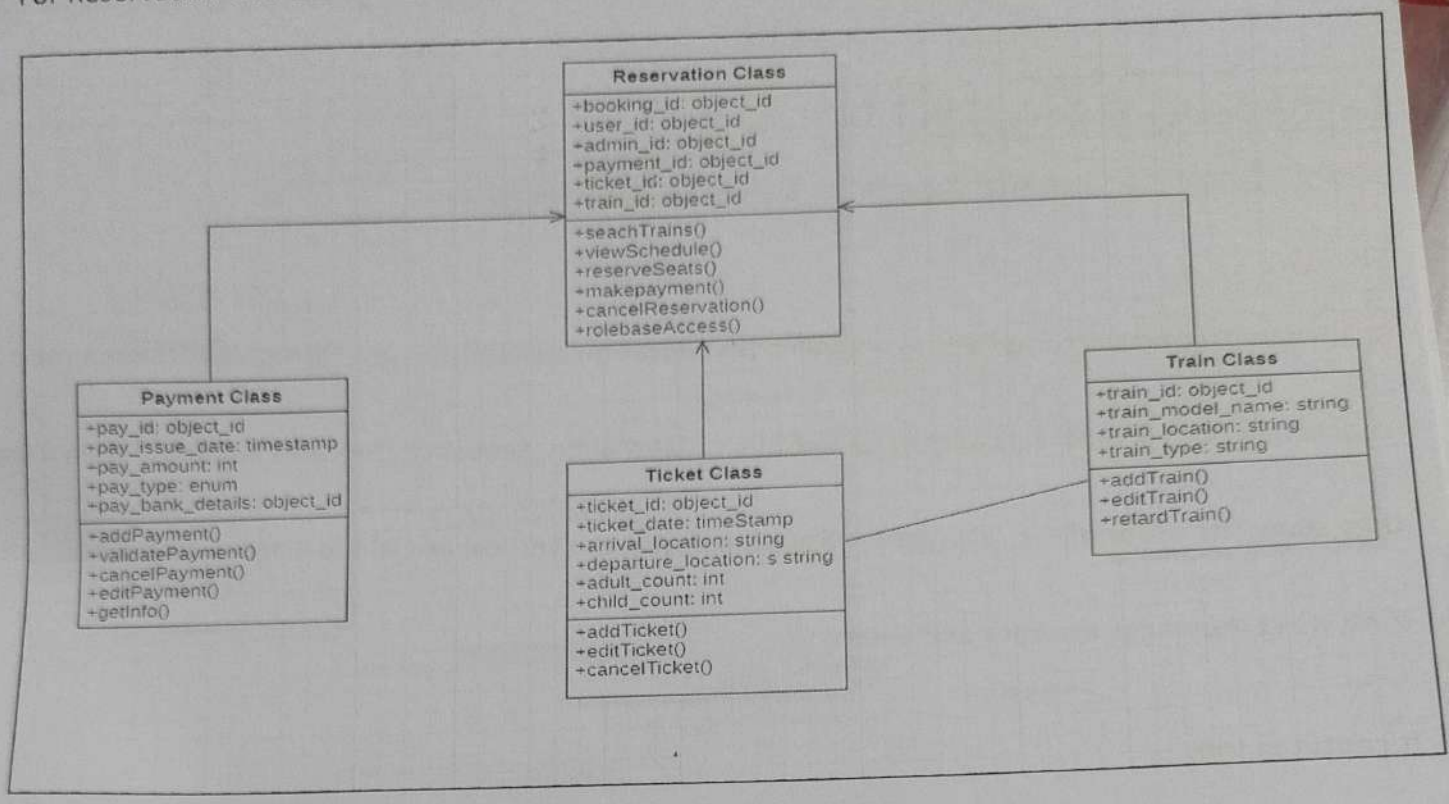
It captures the:

- interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- high-level interactions between the user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)

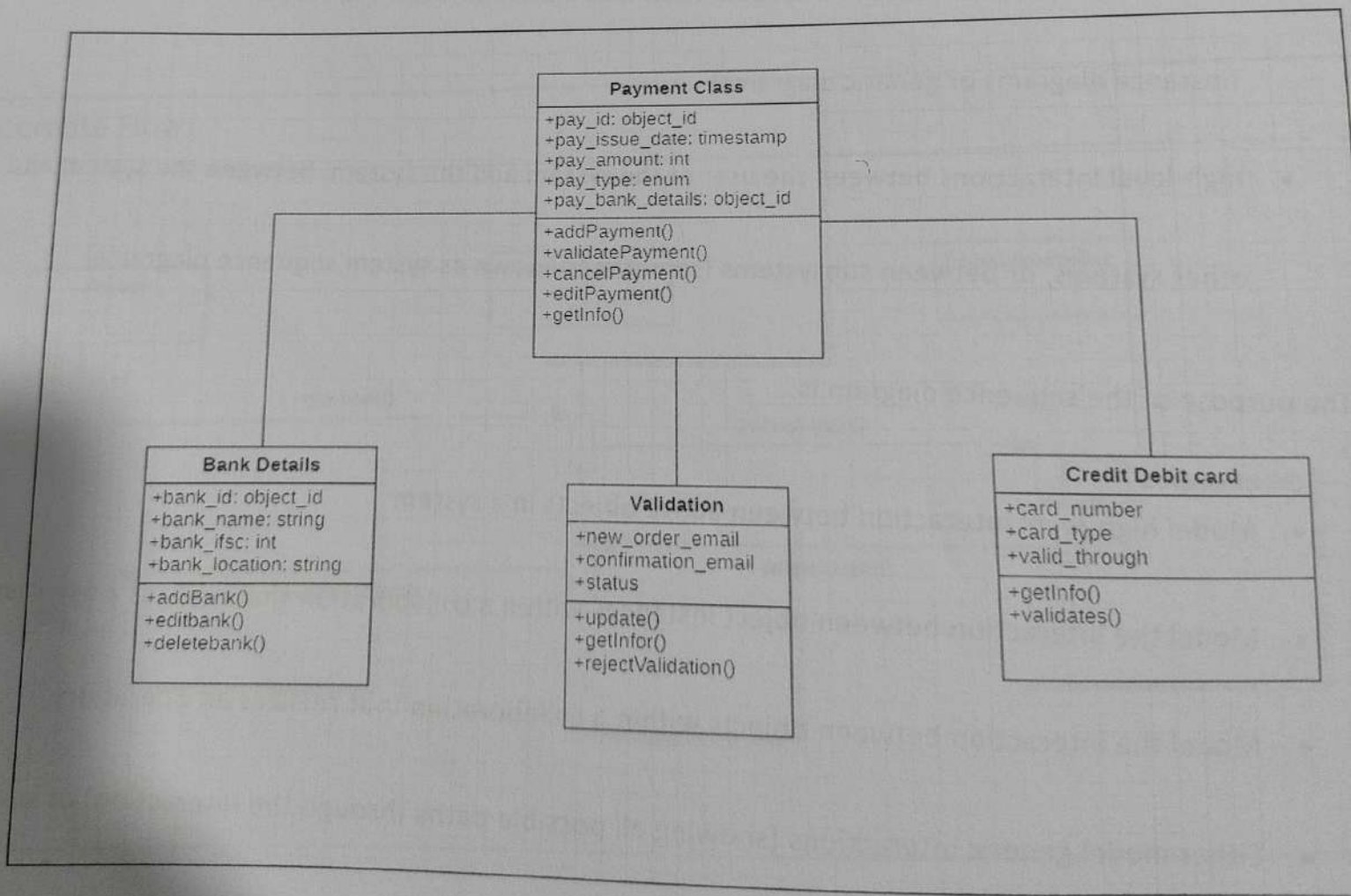
The purpose of the sequence diagram is:

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case.
- Model the interaction between objects within a collaboration that realizes an operation.
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of an interaction (showing just one path through the interaction).

For Reservation Use Case:



For Payment Use Case:



EXPERIMENT 5: CLASS DIAGRAM

AIM: To design the Class Diagram for Railway Reservation System

Theory:

- The class diagram is a static diagram and it is used to model the static view of a complete system. The static view describes the overall vocabulary of the system.
- The class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.
- Class diagrams are the only diagrams that can be directly mapped with object-oriented languages and are thus widely used at the time of construction.
- The class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of the class diagram represents the whole system.
- The class diagram is also considered the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.
- The class diagram clearly shows the mapping with object-oriented languages such as Java, C++, etc. From practical experience, a class diagram is generally used for construction purposes.
- In brief, the purposes of class diagrams can be said to be as follows –
    - Analysis and design of the static view of an application.
    - Describe the responsibilities of a system.
    - The base for component and deployment diagrams.
    - Forward and reverse engineering.

6. Managing and updating data (Ticket Status)

Introduction: This use case is for the management of user details and updating them when needed.

Actors:

1. Admin

Precondition: The actor must be logged onto the system.

Post-condition: If the use case is successful, then the actor can view the data according to query or successfully update the records.

Event Flow

Basic Flow

1. The validation is checked first.
2. Actor can sort the data according to his query i.e., arrival time, departure time, platform etc.
3. Actor updates the data if there are any changes in schedule or trains.
4. The database is managed accordingly.
5. Database gets updated whenever any changes are made like booking, cancelling etc.

Alternative Flows

Actor exits the use case.

Special requirement:

None

Associated use case:

Login

5. Payment

Introduction: This Use case deals with the payment portal of the system which further confirms the reservation.

Actors:

1. Admin
2. User

Precondition: The actor must be logged in.

Post-condition: If the use case is successful, then the seat is permanently booked or cancelled as per the request.

Event Flow

Basic Flow 1

1. Amount for booking the seats is generated according to the destination and number of passengers.
2. User is sent to the payment gateway.
3. User enters the bank details and confirms payment.
4. Seat is permanently booked for the user.

Basic Flow 2

1. Refund is generated according to the norms of the authority for the cancelled ticket.
2. User is asked to enter the bank details for refund transfer.
3. Details are verified and seat is permanently cancelled for the user.

Alternative Flows

1. **AlternativeFlow1:** The bank details are not verified.
   If the details are wrong the payment fails and seat is not booked or cancelled.
2. **AlternativeFlow2:** User exits

This allows the user to exit at any time during the use case. The use case ends.

Special requirement:

None

Associated use case:

1. Booking reservation, Cancellation of ticket

4. Cancellation of ticket

Introduction: This use case is for the cancelling the already booked seats if required.

Actors:

1. Admin
2. User

Precondition: The actor must be logged on to the system and must know the ticket ID.

Post-condition: If the use case is successful, then the seat is un-booked and refund is generated.

Event Flow

Basic Flow

1. Actor enters the ticket ID and user ID of the ticket he wants to cancel.
2. Details are authenticated by system.
3. Actor is asked to confirm the cancellation.
4. If okay, the ticket is cancelled and corresponding seats are put to vacant.
5. Refund is requested and payment use case is called.

Alternative Flows

1. **AlternativeFlow1:** When details are not correct.

   The use case falls back and actor is again asked for details.

2. **AlternativeFlow2:** User exits or doesn't confirm to cancel.

This allows the user to exit at any time during the use case. The use case ends.

Special requirement:

None

Associated use case:

1. Payment

3. Booking Enquiry and Reservation

Introduction: This use case is for reserving seats for the users according to their needs.

Actors:

1. User

Precondition: The actor must be logged onto the system

Post-condition: If the use case is successful, the seat is temporarily booked for the user which will be confirmed after payment.

Event Flow

Basic Flow

1. User enters the details i.e., destination from, destination to and date of departure.
2. According to the info, details of the train available will be shown to user.
3. User can check availability of seats.
4. User books the required number of vacant seats.
5. Payment use case is called.

Alternative Flows

1. **Alternative Flow 1:** User exits

This allows the user to exit at any time during the use case. The use case ends.

Special requirement:

None

Associated use case:

1. Payment

2. Recovery of Account And Profile Management

Introduction: This use case is for the profile recovery of the user.
Actors: 1. Admin 2. User
Precondition: The actor must have a registered email or mobile number.
Post-condition: If the use case is successful, then the actor is provided with security code and authenticated.
Event Flow **Basic Flow** 1. The actor enters the registered email id or phone number. 2. Entered info is authenticated. 3. If successful, actor is provided with the security key. 4. Actor gets to reset his password using that key.
Alternative Flows ID or mobile number entered by the actor is not valid. The Actor is not authenticated and is sent back.
Special requirement: None
Associated use case: None

1. Authenticated Login

Introduction: This use case describes the registration/login of the user into the system.

Actors:

1. Admin
2. User

Precondition: The actor must have a valid id password. If he's a new user, he should have valid id and phone number to sign up to the system.

Post-condition: If the use case is successful, then the actor is logged in.

Event Flow

Basic Flow

1. Actor enters the ID and password or the details to sign-up.
2. The details are authenticated.
3. Actor is logged-in/Signed-up.
4. Database is updated if new user enters.

Alternative Flows

1. **Alternative Flow 1 :** The actor is not authenticated. The actor is sent back to the login page.
2. **Alternative Flow 2 :** User/Admin exits the system. The use case ends.

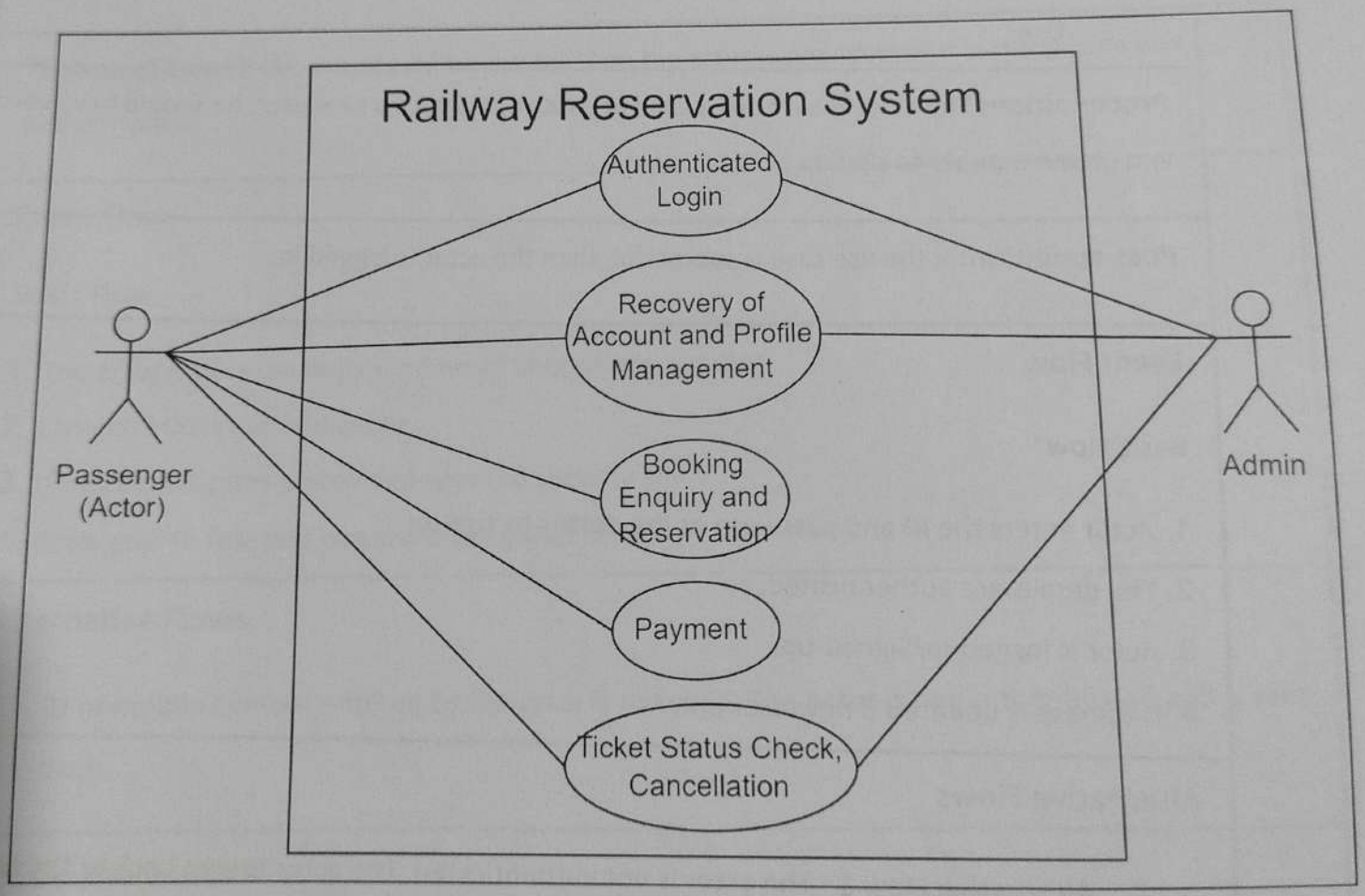
Special requirement:

None

Associated use case:

Login

- In brief, the purposes of class diagrams can be said to be as follows –
    - Analysis and design of the static view of an application.
    - Describe the responsibilities of a system.
    - The base for component and deployment diagrams.
    - Forward and reverse engineering.



Use Case Diagram for RRS

EXPERIMENT 4: USE CASE DIAGRAM

AIM: To design the Use Case diagram for Railway Reservation System

Theory:

- The class diagram is a static diagram and it is used to model the static view of a complete system. The static view describes the overall vocabulary of the system.
- The class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.
- Class diagrams are the only diagrams that can be directly mapped with object-oriented languages and are thus widely used at the time of construction.
- The class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of the class diagram represents the whole system.
- The class diagram is also considered the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.
- The class diagram clearly shows the mapping with object-oriented languages such as Java, C++, etc. From practical experience, a class diagram is generally used for construction purposes.

3.7 Organizing the specific requirements

3.7.1 System mode

3.7.2 User class

3.7.3 Objects

3.7.4 Feature

3.7.5 Stimulus

3.7.6 Response

3.7.7 Functional hierarchy

3.8 Additional comments: The software is such that as the time goes by the need of the RRS may keep on changing thus it is made to change from time to time.

4. Supporting Information

Supporting information makes the SRS easier to use. It may include table of contents, index, appendixes, etc.

3.3 Performance requirements

- Should support at least 8 terminals.
- Should support at least 7 users simultaneously.
- Should run on a 500 MHz, 512 MB RAM machine.
- Responses should be within 2 seconds.

3.4 Logical Database

- **Data Records**

1. **User Record:** - This record helps for the authorization for allowing users to book their reservations. It provides the username and password for the user.
2. **Train Records:** - This record will help to store the record of all the static information like number of trains along with their number, stations numbers, available seats, arrival and departure time, etc.

3.5 Design Constraints

The constraints at the designing time are that the needs of the employees and the administrators may keep on changing so the designers must.

3.6 Software System Attributes

3.6.1 Usability The application will be user-friendly and easy to operate, and the functions will be easily understandable.

3.6.2 Reliability The applications will be available to the students throughout the registration period and have a high degree of fault tolerance.

3.6.3 Security The application will be password protected. Users will have to enter the correct login ID and password to access the application.

3.6.4 Maintainability The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules.

3.6.5 Portability The application will be easily portable on any windows-based system that has MongoDB Server installed

VI. Managing and updating data

Introduction: This use case is for the management of user details and updating them when needed.
Actors: 1. Admin
Precondition: The actor must be logged onto the system.
Post-condition: If the use case is successful, then the actor can view the data according to query or successfully update the records.
Event Flow **Basic Flow** 1. The validation is checked first. 2. Actor can sort the data according to his query i.e., arrival time, departure time, platform etc. 3. Actor updates the data if there are any changes in schedule or trains. 4. The database is managed accordingly. 5. Database gets updated whenever any changes are made like booking, cancelling etc.
Alternative Flows Actor exits the use case.
Special requirement: None
Associated use case: Login

V. Payment

Introduction: This Use case deals with the payment portal of the system which further confirms the reservation.

Actors:

1. Admin
2. User

Precondition: The actor must be logged in.

Post-condition: If the use case is successful, then the seat is permanently booked or cancelled as per the request.

Event Flow

Basic Flow 1

1. Amount for booking the seats is generated according to the destination and number of passengers.
2. User is sent to the payment gateway.
3. User enters the bank details and confirms payment.
4. Seat is permanently booked for the user.

Basic Flow 2

1. Refund is generated according to the norms of the authority for the cancelled ticket.
2. User is asked to enter the bank details for refund transfer.
3. Details are verified and seat is permanently cancelled for the user.

Alternative Flows

1. **AlternativeFlow1:** The bank details are not verified.
   If the details are wrong the payment fails and seat is not booked or cancelled.

2. **AlternativeFlow2:** User exits
This allows the user to exit at any time during the use case. The use case ends.

Special requirement:

None

Associated use case:

1. Booking reservation, Cancellation of ticket

IV. Cancellation of ticket

Introduction: This use case is for the cancelling the already booked seats if required.

Actors:

1. Admin
2. User

Precondition: The actor must be logged on to the system and must know the ticket ID.

Post-condition: If the use case is successful, then the seat is un-booked and refund is generated.

Event Flow

Basic Flow

1. Actor enters the ticket ID and user ID of the ticket he wants to cancel.
2. Details are authenticated by system.
3. Actor is asked to confirm the cancellation.
4. If okay, the ticket is cancelled and corresponding seats are put to vacant.
5. Refund is requested and payment use case is called.

Alternative Flows

1. **AlternativeFlow1:** When details are not correct.

   The use case falls back and actor is again asked for details.

2. **AlternativeFlow2:** User exits or doesn't confirm to cancel.

This allows the user to exit at any time during the use case. The use case ends.

Special requirement:

None

Associated use case:

1. Payment

III. Booking Reservation

Introduction: This use case is for reserving seats for the users according to their needs.

Actors:

1. User

Precondition: The actor must be logged onto the system

Post-condition: If the use case is successful, the seat is temporarily booked for the user which will be confirmed after payment.

Event Flow

Basic Flow

1. User enters the details i.e., destination from, destination to and date of departure.
2. According to the info, details of the train available will be shown to user.
3. User can check availability of seats.
4. User books the required number of vacant seats.
5. Payment use case is called.

Alternative Flows

1. **Alternative Flow 1:** User exits

This allows the user to exit at any time during the use case. The use case ends.

Special requirement:

None

Associated use case:

1. Payment

II. Recovery of Account

Introduction: This use case is for the profile recovery of the user.

Actors:

1. Admin
2. User

Precondition: The actor must have a registered email or mobile number.

Post-condition: If the use case is successful, then the actor is provided with security code and authenticated.

Event Flow

Basic Flow

1. The actor enters the registered email id or phone number.
2. Entered info is authenticated.
3. If successful, actor is provided with the security key.
4. Actor gets to reset his password using that key.

Alternative Flows

ID or mobile number entered by the actor is not valid. The Actor is not authenticated and is sent back.

Special requirement:

None

Associated use case:

None

3.3 Functions

I. Authenticated Login

Introduction: This use case describes the registration/login of the user into the system.
Actors: 1. Admin 2. User
Precondition: The actor must have a valid id password. If he's a new user, he should have valid id and phone number to sign up to the system.
Post-condition: If the use case is successful, then the actor is logged in.
Event Flow **Basic Flow** 1. Actor enters the ID and password or the details to sign-up. 2. The details are authenticated. 3. Actor is logged-in/Signed-up. 4. Database is updated if new user enters.
Alternative Flows 1. **Alternative Flow 1 :** The actor is not authenticated. The actor is sent back to the login page. 2. **Alternative Flow 2 :** User/Admin exits the system. The use case ends.
Special requirement: None
Associated use case: Login

2.6 Apportioning of Requirements

This subsection list those requirements that can be delayed and may be incorporated in the future version of the software system. For the LMS, there is no such requirement.

---

3. Specific requirements

This section contains the software requirements in detail along with the various forms to be developed.

3.1 External interfaces

External Interfaces user who will use the system to book the tickets online.

3.2 Functions requirements

Registration: If the customer wants to book some seats, then he/she must be registered, the unregistered user can't book the seats.

Login: Customer logins to the system by entering a valid email id and password for the seats to be booked

Search: The user can search their required shows and check for available seats at different theatres.

Select: User is allowed to select a required number of seats at liked places.

Payment: Payment is done through net banking or debit/credit cards or through UPI. The seats will be booked and blocked only after successful payment.

Receipt Generation: A unique transaction id is generated for future references

Ticket Generation: A ticket is produced with seat numbers, time and place of the show on it, and also a QR code.

Error handling: If any of the above validation/sequencing flow does not hold true, appropriate error messages will be prompted to the user for doing the needful.

2.1.3 Hardware Interfaces

(a) Screen resolution of at least 640 × 480 or above.

(b) Support for printer (dot matrix, deskjet, laserjet).

(c) Computer systems will be in the networked environment as it is a multi-user system.

2.2 Product Functions

The RRS will allow access only to the authorized users with specific roles (system administrator, and users). Depending upon the user's role, he/she will be able to access only specific modules of the system. A summary of major functions that the RRS shall perform includes:

- A log-in facility for enabling only authorized access to the system.
- The system administrator will be able to add, modify, delete or view details of users and trains.
- The system administrator/library staff will be able to book and cancel the reservation of the user in special cases.
- Most users will book the tickets and complete the transaction.
- User and admin both can recover their profile.

2.3 User Characteristics Qualification:

At least 16+ and comfortable with English.

Experience: Needs not to be well informed with the procedure of booking tickets.
Technical experience: Elementary knowledge of computers.

2.4 Constraints

- The user won't be able to make changes directly into the database.
- Passwords for all the user and admin profiles will be first encrypted or hashed then stored in the database.
- There will not be multiple roles of admin, all the admin will perform similar operations.

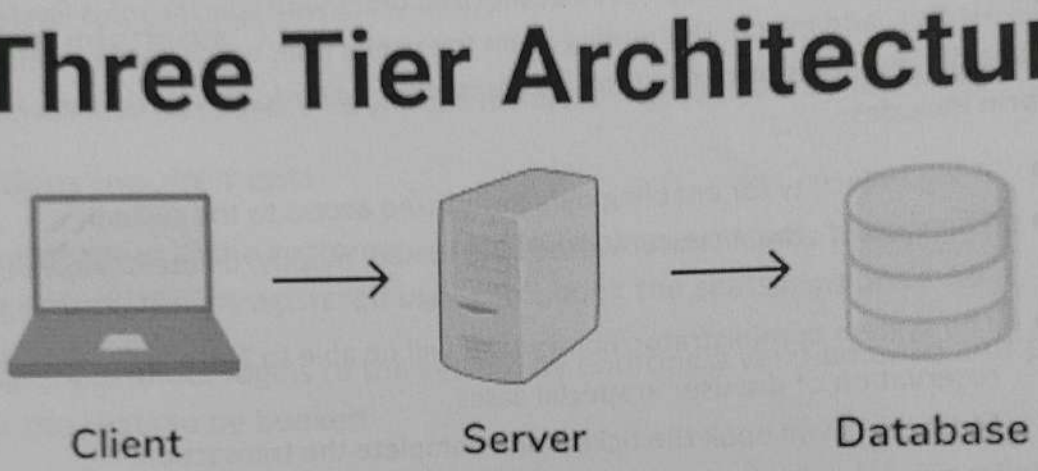
2.5 Assumptions and Dependencies

- Login Id for all the admin will be created at the development stage.
- All the information related to booking of details would be pre-stored.
- A hosting platform will be required.

- User details.
- Reservation details.
- Admin details.
- Payment details.

2.1 Product Perspective

The RRS shall be developed using client/server architecture and will be compatible with desktop and mobile users. The front-end of the system will be developed using React or angular-like frameworks and the back-end will be developed using Node.js and express along with MongoDB for the database.



2.1.1 System Interfaces

System interfaces are addressed which are required by the product during its usage.

2.1.2 User Interfaces

The RRS will have the following user-friendly and menu-driven interfaces:

(a) **Login**: to allow the entry of only authorized users through valid log-in ID and password.

(b) **Trains details**: to book suitable tickets for trains.

(c) **User profile**: In the profile, the user will have its previous as well as current booking status.

(d) **Change/forget Password:** to recover the account or profile.

(e) **Booking reservations**: After providing the required information user can book the tickets.

(f) **Payment**: to complete the process of booking tickets.

1.4 References

- IEEE Recommended Practice for Software Requirements Specifications— IEEE Std 830-1998.
- http://mca.ignougroup.com/2016/11/srs-of-railway-reservation-system.html
- https://www.slideshare.net/khushikalaria/srs-for-railway-reservation-system.

1.5 Overview

This subsection of the SRS gives an overview of the software system. Explain what the rest of the SRS contains and also give details about the organization of the SRS document.

---

2. Overall Description

The RRS is a platform that enables users to book their online train tickets.

It allows only those user which has authenticated email address and has been verified. Users can search/query for multiple slots in order to book their reservations. The software interface will be user-friendly and try to navigate the user completely. Users can be reserved or book the available seats/tickets according to their preferences and requirements. Reservation of tickets will require a few information in order to prevent misguidance in the future like Date, station number from and to, number of seats required, number of adults and kids, train preference, etc. After satisfying the need user can proceed with payment using different card options or net banking.

Data management and updation will be required in real-time changes made by one user will reflect the other users and in the database as well. All the static data will be managed priorly like the number of trains, their availability, scheduling dates, station numbers, etc. User data will be managed and updated by the admin or system operator. User can also recover their account in case they forget their password.

Cancellation of reservation is applicable and the fund will be refunded only if the cancellation had been done one day prior. If the user canceled the reservation then we need to roll back the entries that have been executed in the payment schema corresponding to the user and the fund needs to go back to the user's bank account.

A system admin will have to maintain the following information

3. Maintain the details of users and admin who are connected to the RRS.
4. Querying/searching for available slots for reservation.
    - Date
    - A number of seats are required.
    - Train if specific.
    - From To, station number.

5. Complete Reservation.
6. Payment
7. Cancellation for payment.
8. Updating the real-time data.

Don'ts

1. Booking will not be done ahead of 3 months.
2. The software will not have its own payment gateway.
3. Payment cancellation will not take place after the allotted reserved day.

The RRS provides the following benefits:

1. Easy searching for slots.
2. Can book train reservations prior from any location.

1.3 Definitions, Acronyms, and Abbreviations

SRS: Software requirement specification

RRS: Online Train Ticketing System

System operator: System administrator/Admin, the database operator

RAM: Random access memory.

User: Any user who had authenticated email address will be able to access the platform for booking purposes.

System Admin/Administrator: The user having all the privileges to operate the RRS.

Database operator: All the changes in the database, or requested by the admin will be done by the database operator.

1 Introduction

This section gives an overview of the SRS document.

1.1 Purpose

The purpose of the Online Train Ticketing System is to create reservations, cancel reservations, viewing train information, viewing reservation details, updating train information, payment confirmation, and cancel.

1.2 Scope

The name of the software is Online Train Ticketing System. The system will be referred to as RRS in the rest of SRS. The purposed RRS must be able to perform the following functions:

Dos

1. Issue of login credentials to Admin as well as users.
2. Maintain the details of Train number, available seats, station code, arrival and departure time.

EXPERIMENT 3: SOFTWARE REQUIREMENT SPECIFICATION (SRS) DOC

AIM: To generate the SRS (Software Requirement Specifications) document for RRS

1.	**Introduction** 1.1 Purpose 1.2 Scope 1.3 Definitions, acronyms, and abbreviations 1.4 References 1.5 Overview
2.	**Overall description** 2.1 Product perspective 2.1.1 System interfaces 2.1.2 User interfaces 2.1.3 Hardware interfaces 2.1.4 Software interfaces 2.2 Product functions 2.3 User characteristics 2.4 Constraints 2.5 Assumptions and dependencies 2.6 Apportioning of requirements
3.	**Specific requirements** 3.1 External Functions 3.2 Functions requirements 3.3 Performance requirements 3.4 Logical database requirements 3.5 Design Constrain 3.6 Software system attributes 3.6.1 Reliability

Consolidated List of Initial Requirement

1. A system is to be implemented which can run efficiently on station LAN.

2. A booking id should be generated to see the status of the ticket.

3. The system shall be able to generate login ID and password to the system operator.

4. There are two types of members in the system: passengers and station staff.

5. The administrator shall be able to maintain details of all the tickets.

6. The administrator shall be able to maintain details of all the members of the station.

7. The passengers shall be able to book the available tickets after paying the price.

8. The passengers shall be able to cancel the tickets.

9. The maximum number of tickets that can be issued are equal to the seats in the train.

10. The system shall calculate the ticket prices according to distance and passengers.

12. The system shall keep a record of passenger's previous data too.

13. The system shall be able to provide the availability of tickets of particular trains.

14. The system shall be able to provide the interface to passengers to check the availability of tickets and the route of their train.

15. The system should also be able to generate reports like:

(i) Details of all the tickets

Destination from-wise

Destination to-wise

Arrival time-wise

Departure time-wise

(ii) Details of all passengers of a train.

(iii) Details of trains on a specific station.

EXPERIMENT 2: INITIAL REQUIREMENT DOCUMENT (IRD)

AIM: To write the Initial Requirement Document for for Railway Reservation System

Title of the Project	Railway Reservation System UML Diagrams
Stakeholders	Staff, Conductors, Administration, System Engineers, Passengers
Techniques used for requirement capturing	Interviewing and Brainstorming
Date	April 2022
Version	1.0

1. **Basic Requirements:**
    a. Source Station
    b. Destination Station
    c. Date of the journey
2. **Functional Requirements:**
    a. Details of passengers (Name, Age, Total Passengers, Gender, etc.)
    b. Payment Method (Net Banking, Debit Card, Credit Card, etc.)
    c. Details of the booking particulars (Class, Quota, Upgrade, Vikalp, Concession)
    d. Balance and Availability Updation
    e. Final Booking Confirmation and Ticket Details
3. **Non-Functional Requirements:**
    a. Trading system failure.
    b. Unavailability of the berth.
    c. Unavailability of date.
    d. Coach Unavailability
    e. The insufficient amount for making payment.

details about the ticket. The software ensures the safety and security of details about the passengers, their payment details and ensures the integrity of the data being processed: The cancellation of the tickets is also very easy in an online mode such that the passenger can cancel the tickets that he has booked.

Authentication

- Users have to login into the application to avail the facilities.
- Admin also has to be authenticated to perform operations.
- Password change or recovery of the account is also facilitated.

Booking of reservations

- Provide a user-friendly interface to admin as well as to customers.
- Users will easily get the information related to all the trains, pricing, stations, etc.
- Can book the available slots and related data will be stored.
- The payment process would be the last step before completion of the Booking of reservation.
- Cancellation of the reservation after the payment completion should also be handled.

Storing and updating the information

- Keep updating the information about the routes, cancellation of tickets, departure time, arrival time, number of trains available for specific routes, and other such information.
- Keep track of our passengers' details, their previous and current traveling status in the different trains on different dates.
- Store and retrieve information about the various transactions related to customer traveling.
- Storing the records of all the train numbers, routes, capacity, reservations, stations number, costs, etc.

Object Oriented Software Engineering
CO326 OOSE Lab File

EXPERIMENT 1: PROBLEM STATEMENT

AIM: To define the Problem Statement for Railway Reservation System

An online railway reservation is an efficient method to reserve passenger travel tickets from the comfort of their own devices and skip the long queues by not standing in the railway station queue. Today, all major railway systems across the world have their own website or a mobile application for making a reservation to provide better and quick service to the customer. Also, this bypass the difficulties that the users used to face with the manual process of filling out the reservation form wherein they could no longer change the details of the passengers once filled

The main purpose of online railway reservations is to ease out the tedious task of railway activity. Initially, the customer has to create an ID on the appropriate website so that the user can log into the system for doing further activities. An online manager will maintain a database of the accounts records.

To do the login process the customer has to fill a registration form that contains the username, password, first name, last name, and other details relevant to maintain the record. After submitting the form to the server, a unique customer ID is created with username and password, thereby the customers, with only the appropriate ID, can reserve the tickets successfully.

The system also facilitates various PNR and reservation inquiries. Every time a ticket is booked there must be a proper record to be maintained and also the ticket status, availability status must be updated from time to time. There may be any new addition of trains and trains being deleted or stations added or deleted which must be updated prior to the database.

The availability status is changed according to the number of tickets booked by users across their devices. The inquiry option allows the user to enter the PNR number and get the current

Object Oriented Software Engineering
CO326 OOSE Lab File

EXPERIMENT 1: PROBLEM STATEMENT

AIM: To define the Problem Statement for Railway Reservation System

An online railway reservation is an efficient method to reserve passenger travel tickets from the comfort of their own devices and skip the long queues by not standing in the railway station queue. Today, all major railway systems across the world have their own website or a mobile application for making a reservation to provide better and quick service to the customer. Also, this bypass the difficulties that the users used to face with the manual process of filling out the reservation form wherein they could no longer change the details of the passengers once filled

The main purpose of online railway reservations is to ease out the tedious task of railway activity. Initially, the customer has to create an ID on the appropriate website so that the user can log into the system for doing further activities. An online manager will maintain a database of the accounts records.

To do the login process the customer has to fill a registration form that contains the username, password, first name, last name, and other details relevant to maintain the record. After submitting the form to the server, a unique customer ID is created with username and password, thereby the customers, with only the appropriate ID, can reserve the tickets successfully.

The system also facilitates various PNR and reservation inquiries. Every time a ticket is booked there must be a proper record to be maintained and also the ticket status, availability status must be updated from time to time. There may be any new addition of trains and trains being deleted or stations added or deleted which must be updated prior to the database.

The availability status is changed according to the number of tickets booked by users across their devices. The inquiry option allows the user to enter the PNR number and get the current