ABSTRACT

Proposed **"Online Ticket Reservation System"** is developed for to automate the railway reservation system. It includes modules required to successfully operate railway reversion process smoothly. It has railway administrator to add modify train information, Train schedule to enter train journey details include all the station name, arrival time and departure time. It includes automatic fare calculation as per the number of passengers.

Reservation module consist of automatic seat no and coach no allocation system. All master like train master, train schedule, reservation fees, cancellation fees charges can be modified individually from front end and changes reflect in all modules immediately. Therefore proposed **"Online Ticket Reservation System"** has been designed to automate the process of railway for ticket reservation and back office activities. System can make the daily activities efficient and providing the fast response.

FEATURES AND ADVANTAGES:

- Complete end to end system for all the activities related to railway reservation system.
- Complete interface for all the operations including reservation, PNR query, cancellation, system administration.
- Supports the full life cycle of ticket reservation, seat allocation, and ticket cancellation. Reschedule the cancel seat to unconfirmed passenger.
- It is a scalable system.
- Has a clearly arranged and user-friendly interface.
- Easy to use and minimum data entry, all important details is mater driven.
- Option to create a new account.
- Integrated security features.

To implement this sample case study, some assumptions have been made, which are as follow:

1) The number of trains has been restricted to 3.
2) The booking option is open only for next seven days from the current date.
3) Only 3 categories can be booked, namely, 1AC, 2AC & SL.
4) The in-between stoppage & their bookings are not considered.

SOFTWARE REQUIREMENT SPECIFICATION

MODULES

Mange Users, Train Schedule, Fare Calculation Master, Ticket Reservation, Ticket Cancellation, Reports. Seat Allotment, Schedule for Seat allocation for cancel ticket, create ne account.

The functional & non-functional requirements are listed below:

1) FUNCTIONAL REQUIREMENTS:

Req. No.	Requirement	Description
1	User Login	The actor must login into the website
2	Search Train	Search train as per journey plan
3	Check Availability	Check availability for the particular date & class
4	Book Ticket	Book tickets by entering passenger information
5	Payment Gateway	Pay the amount through any preferred methods
6	Cancel Ticket	Cancel ticket either partially or fully
7	Manage Bookings	Check train status using PNR

2) NON-FUNCTIONAL REQUIREMENTS

Req. No.	Requirement	Description
1	Performance	Accessible 24*7
2	Security	128-bit end to end encryption
3	Database	Must be updated periodically
4	Domain	Protected from cyber attacks

3) SOFTWARE REQUIREMENTS:

- x86 based operating system
- Windows 8/Equivalent or higher

4) HARDWARE REQUIREMENTS:

- **Processor:** 3.0 GHz or Greater
- **RAM:** 1 GB or Greater

5) CONDITIONS

Condition No.	Description
1	The entered credentials are valid
2	Ticket availability as per requirements
3	Successful transactions status from payment gateway
4	Ticket details are stored in the account

<h1 style="text-align:center">SOFTWARE DESIGNS</h1>

<h2 style="text-align:center">UML DIAGRAMS</h2>

1) USE CASE DIAGRAM

- A use case diagram is a diagram that shows a set of use cases and actors and relationships.
- Use case commonly contain
  - Use cases
  - Actors
  - Dependency, generalization and association relationships
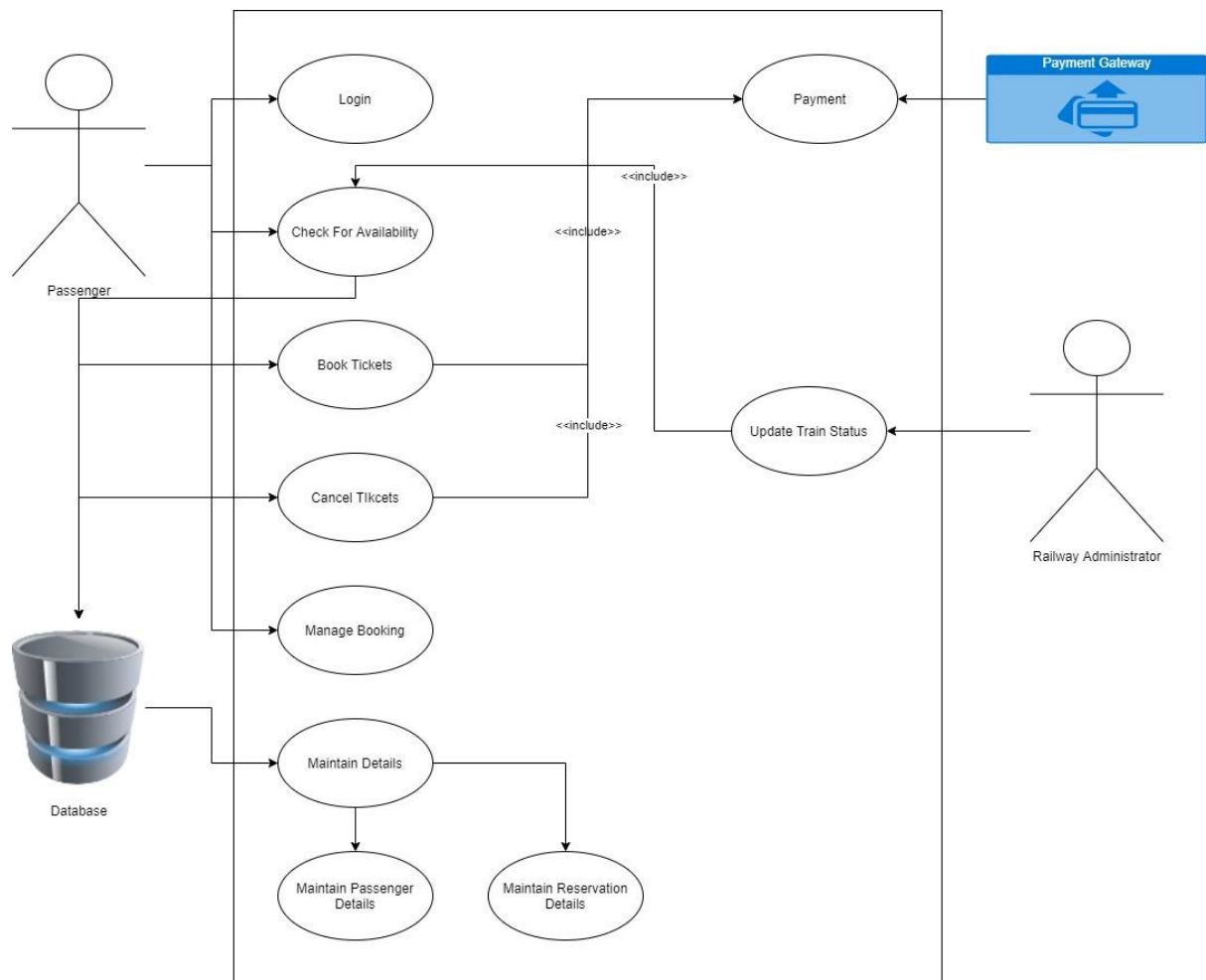
2) CLASS DIAGRAM

- Class diagrams are widely used to describe the types of objects in a system and their relationships.
- Class diagrams model class structure and contents using design elements such as classes, packages and objects.
- Classes are composed of three things: a name, attributes, and operations.
- Class diagrams also display relationships such as containment, inheritance, associations and others.

```
                          ┌──────────────────────┐
                          │       Payment        │
                          ├──────────────────────┤
                          │ +Amount              │
                          ├──────────────────────┤
                          │                      │
                          └──────────────────────┘
                                   1..*
                                    │
                                   0..1
┌──────────────────────┐  ┌──────────────────────┐   ┌──────────────────────┐
│        Ticket        │  │      Passenger       │   │      Ticket Clerk    │
├──────────────────────┤  ├──────────────────────┤   ├──────────────────────┤
│ +TicketNo            │  │ +Name                │   │ +Name                │
│ +StartingLocation    │  │ +Age                 │   │ +Location            │
│ +DestinationLocation │  │ +Address             │   ├──────────────────────┤
│ +PassengerNo         │  ├──────────────────────┤   │ +ReserveSeat()       │
├──────────────────────┤1 │ +SearchTrains()      │1..*│ +ReserveTicket()     │
│ +PaymentAmount()  1..*│  │ +ViewSchedule()    1 │   │ +CancelTicket()      │
│ +CancelTicket()      │  │ +ReserveSeat()       │   │ +ReseivePayment()    │
└──────────────────────┘  │ +PurchaseTicket()    │   └──────────────────────┘
                          │ +CancelTicket()      │            0..1
                          │ +MakePayment()       │             │
                          └──────────────────────┘            1..*
                                   0..1
                                    │
                                   1..*
                          ┌──────────────────────┐   ┌──────────────────────┐
                          │        Train         │   │   Railway Database   │
                          ├──────────────────────┤   ├──────────────────────┤
                          │ +Model               │   │ +ID                  │
                          │ +TrainId             │   ├──────────────────────┤
                          ├──────────────────────┤   │ +Response            │
                          └──────────────────────┘   └──────────────────────┘
```
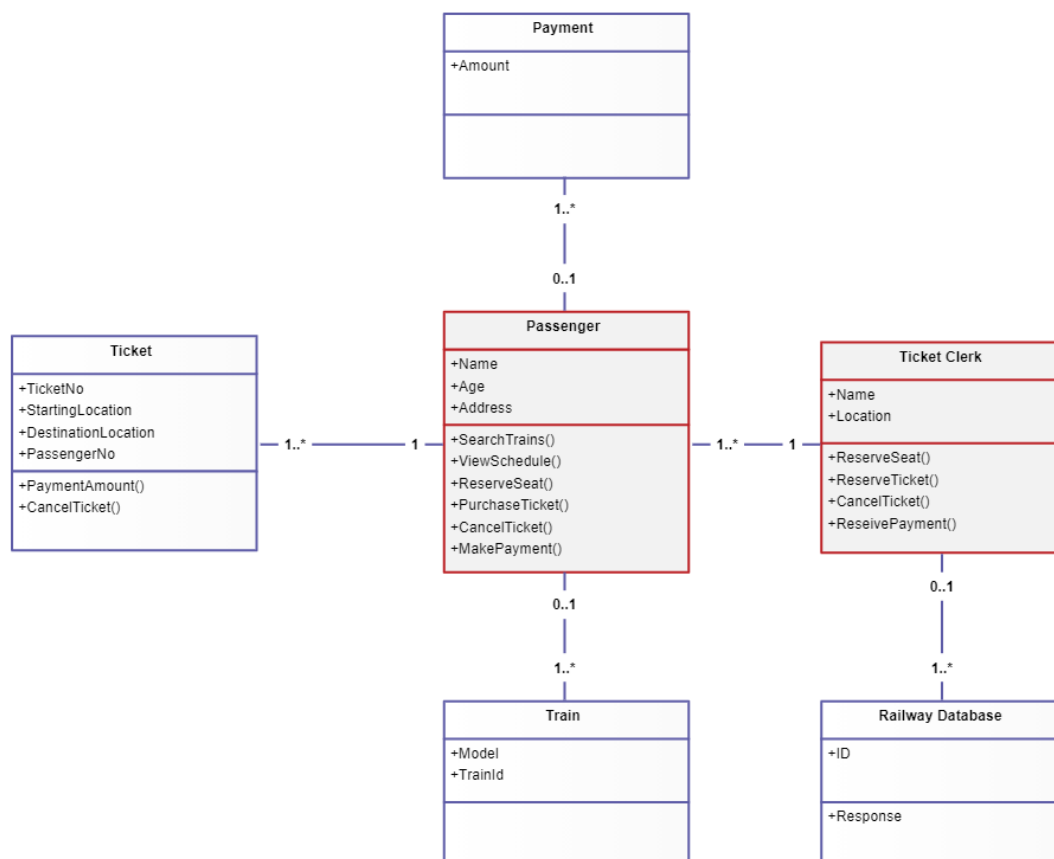
a) CLASS RESPONSIBILITY COLLABORATOR

Passenger
Responsibilities: Search Trains View Schedule Reserve Seats Purchase Ticket Cancel Ticket
Collaborations: Name Age Trainid TicketNo

| Ticket Clerk |
| --- |
| Responsibilities:
Reserve Seats
Cancel Ticket
Revieve Payment |
| Collaborations:
Name
TicketNo |

3) ACTIVITY DIAGRAM

- An activity diagram shows the flow from activity to activity. An activity is an ongoing non-atomic execution within a state machine.
- Activities ultimately result in some action, which is made up of executable atomic computations that result in a change in state of the system or the return of a value.
- Activity diagrams commonly contain
  - Activity states and action states
  - Transitions
  - Objects
- Like all other diagrams, activity diagrams may contain notes and constrains.
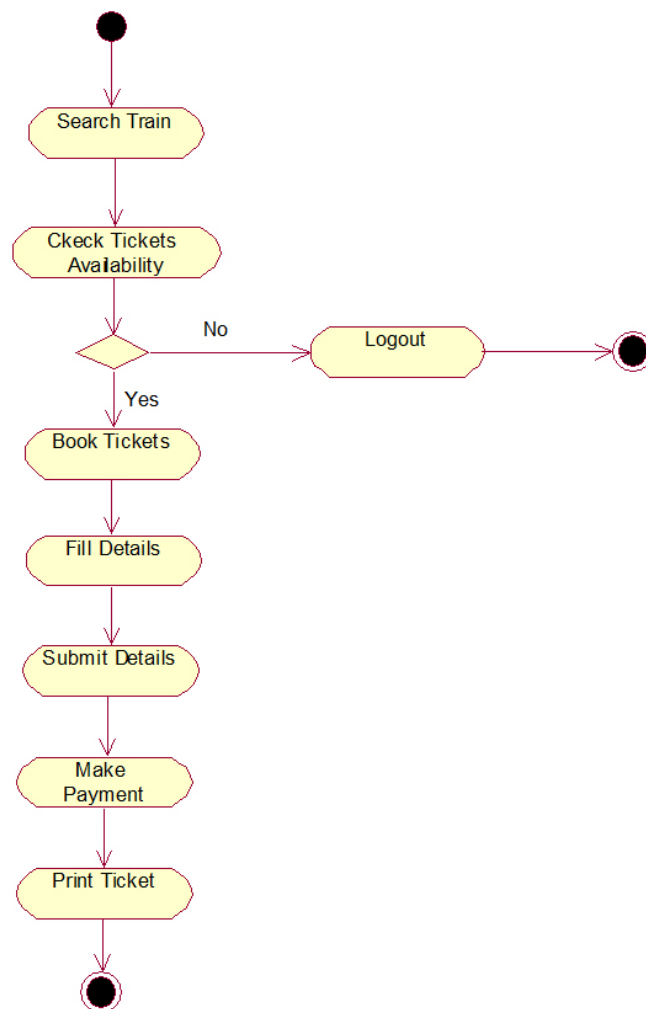
4) SEQUENCE DIAGRAM

- An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.
- A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.
- Graphically, a sequence diagram is a table that shows objects arranged along x-axis and messages, ordered in increasing time, along the y-axis.
- Sequence diagrams commonly contain the following:
  - Objects
  - Links
  - Messages
- Like all other diagrams, sequence diagrams may contain notes and constrains.
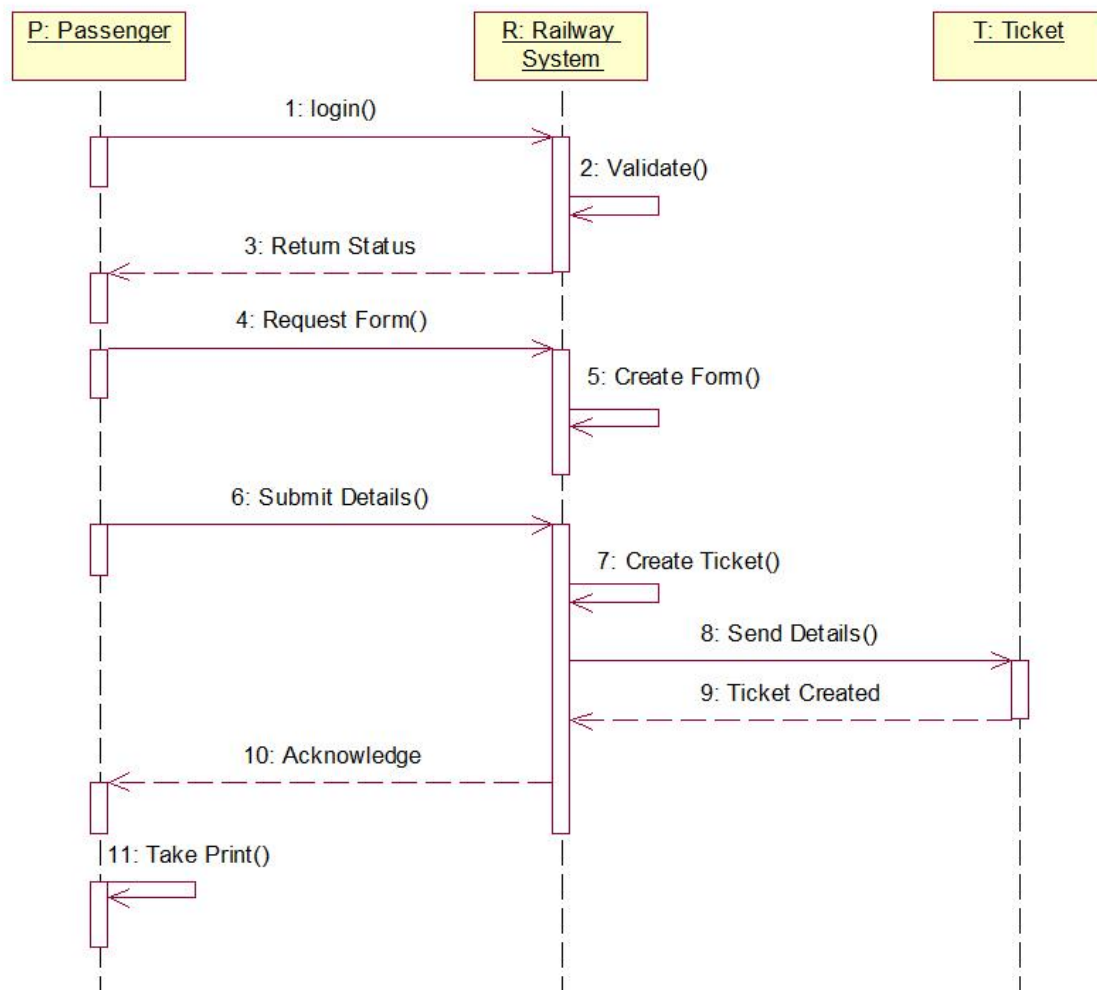
5) STATE DIAGRAM

- State diagrams are used to describe the behavior of a system.
- State diagrams describe all of the possible states of an object as events occur.
- Each diagram usually represents objects of a single class and track the different states of its objects through the system.

```
                    ●
                    │
   Enter login details
                    ▼
              ┌──────────┐
              │Validation│
              └──────────┘
                    │
   Enter train details
                    ▼
              ┌──────────┐
              │Avalability│
              │  Check   │
              └──────────┘
                    │
   Enter self details
                    ▼
              ┌──────────┐
              │ Booking  │
              │  Ticket  │
              └──────────┘
                    │
   Booking successful
                    ▼
              ┌──────────┐
              │ Printing │
              └──────────┘
                    │
   Logout
                    ▼
                    ◉
```