**EXNO : 1**

**USE CASE DIAGRAM FOR ONLINE RAILWAY RESERVATION SYSTEM**

**AIM**

        Design a Use Case diagram for the Online Railway Ticket Booking application by including the passenger operation (User registration, Search Train, View Train, Ticket Booking and Ticket Cancellation) and admin operations (Train, User, Booking management and Report generation)

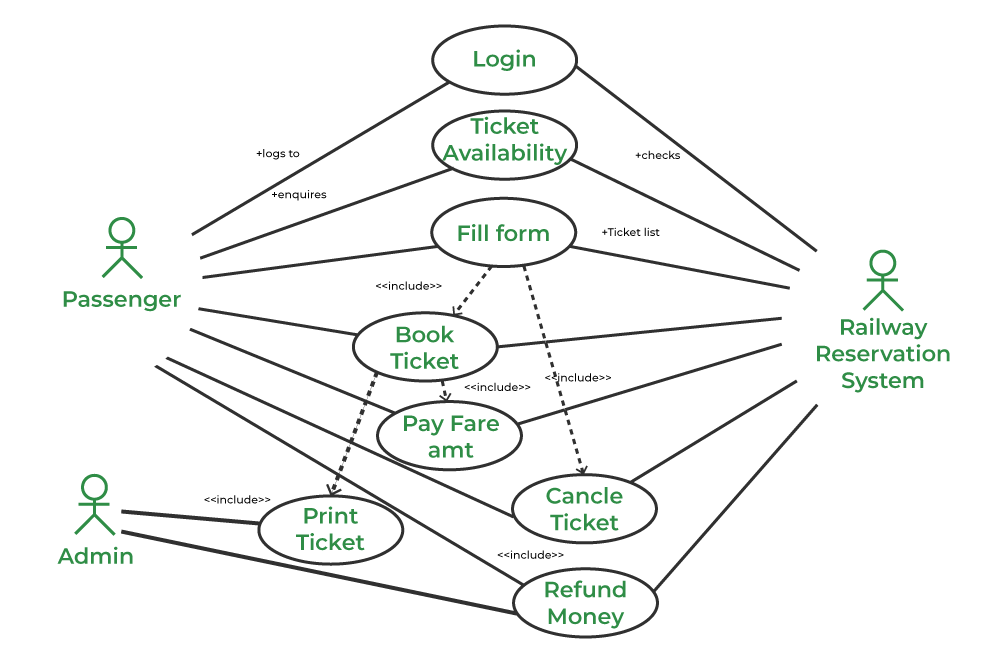
**PROBLEM STATEMENT**

          The online railway reservation system is a digital platform that allows passengers to book train tickets, check seat availability, view train schedules, and manage bookings from the comfort of their homes or mobile devices.

**PROCEDURE**

1. Open the Staruml / Edrawmax tool
2. Use rectangular box to draw the system boundary, representing the boundary between the system and its environment
3. Identify the actors and Use cases according to the online railway ticket booking application.
4. Use the "Association" tool to connect actors to use cases. This represents a relationship between an actor and the use cases they can participate in.
5. Use the "Include" and "Extend" relationships to show how use cases can be related to each other.
6. Double-click on each actor, use case, and relationship to edit its label and provide a clear and concise description.
7. Select the "Save" or "Save As" option from the menu to save the diagram.

**USE CASE DIAGRAM:**



**DOCUMENTATION**

 Actors:

1. Customer/Passenger: A person who uses the application to search for trains, book tickets, and manage bookings.
2. Admin: Manages the overall application, including train schedules, user accounts, and reports.
3. Payment Gateway: An external system that processes payments.
4. Train Operator: Manages train schedules, availability, and other operational details.

Use Cases:

1. Search Trains: Allows customers to search for available trains based on their travel preferences (e.g., date, source, destination).
2. View Train Details: Provides detailed information about a selected train, including timings, stops, and availability.
3. Book Ticket: Enables customers to select a train and book a ticket for their journey.
4. Make Payment: Facilitates payment for the ticket through various methods (credit/debit card, net banking, etc.) via the payment gateway.
5. Cancel Ticket: Allows customers to cancel a booked ticket and receive a refund as per the cancellation policy.
6. View Booking History: Allows customers to view their past and current bookings.

**RESULT**

The above use case diagram was designed successfully and documentation was made

**EXNO : 2**

**CLASS DIAGRAM FOR ONLINE RAILWAY RESERVATION SYSTEM**

**AIM**

        Design a Class diagram for the Online Railway Ticket Booking application by including the passenger and admin operations.

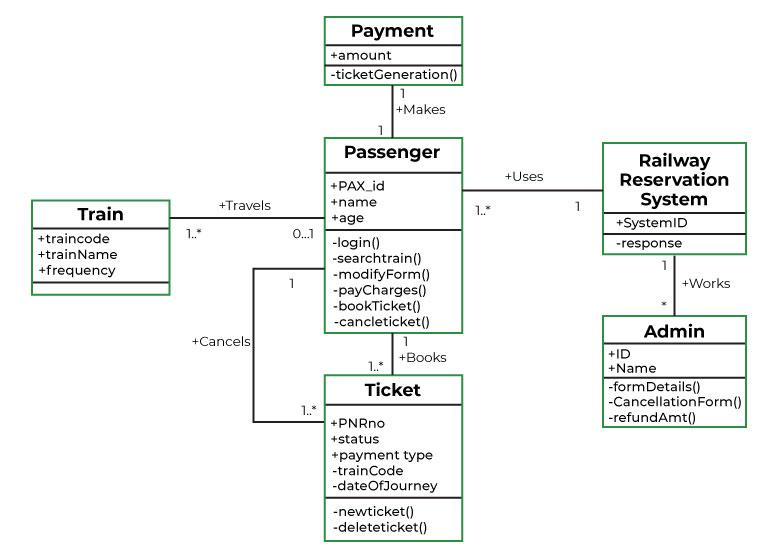
**PROBLEM STATEMENT**

          The online railway reservation system is a digital platform that allows passengers to book train tickets, check seat availability, view train schedules, and manage bookings from the comfort of their homes or mobile devices.

**PROCEDURE**

1. Open the Star UML / Edrawmax tool
2. Identify the required number of classes for the online railway ticket booking application namely User, Ticket, Train, Station, Booking
3. Use the "Class" tool to create the classes: User, Ticket, Train, Station, and Booking.
4. Double-click on each class to add attributes (e.g., id, name) and operations (if necessary).
5. Identify the relationship among the classes such as one to one, one to many, many to many
6. Use the "Association" tool to create relationships between classes.
7. Label the relationships and their multiplicities.
8. Select the "Save" or "Save As" option from the menu to save the diagram.

**CLASS DIAGRAM:**



**DOCUMENTATION**

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

In Online Railway Ticket Booking System, the following are the specifications:

**Core Classes:**

1. Passenger:

Attributes: passengerID, name, address, contact, email, password

Operations: register, login, bookTicket, viewBookings, cancelTicket, updateProfile

1. Ticket:

Attributes: ticketID, passengerID, trainID, journeyDate, sourceStation, destinationStation, seatClass, fare

Operations: generateTicket, cancelTicket

1. Train:

Attributes: trainID, trainName, sourceStation, destinationStation, departureTime, arrivalTime, seatAvailability

Operations: viewTrains, updateTrainInfo

1. Booking:

Attributes: bookingID, passengerID, ticketID, paymentStatus

Operations: confirmBooking, cancelBooking, viewBookingStatus

1. Admin-Specific Classes:

Admin:

Attributes: adminID, username, password

Operations: login, addTrain, updateTrain, deleteTrain, viewBookings, managePassengers

1. PaymentGateway:

Operations: processPayment, refundPayment

**Relationships:**

1. Passenger has many Bookings.
2. Booking belongs to one Passenger and one Ticket.
3. Ticket belongs to one Train.
4. Admin can manage multiple Trains and Passengers.

**RESULT**

The above class diagram was designed successfully and documentation was made.