

EX 3:

Make an Online Airline Reservation System. The activities of the Online Airline Reservation system are listed below user, admin, LOGIN, MANANGE CLASSES, MANANGE WAITING LIST, MANAGE HOLDS, MANAGE DEADLINES, LOGOUT, using this has a step-by-step process draw a CLASS diagram.

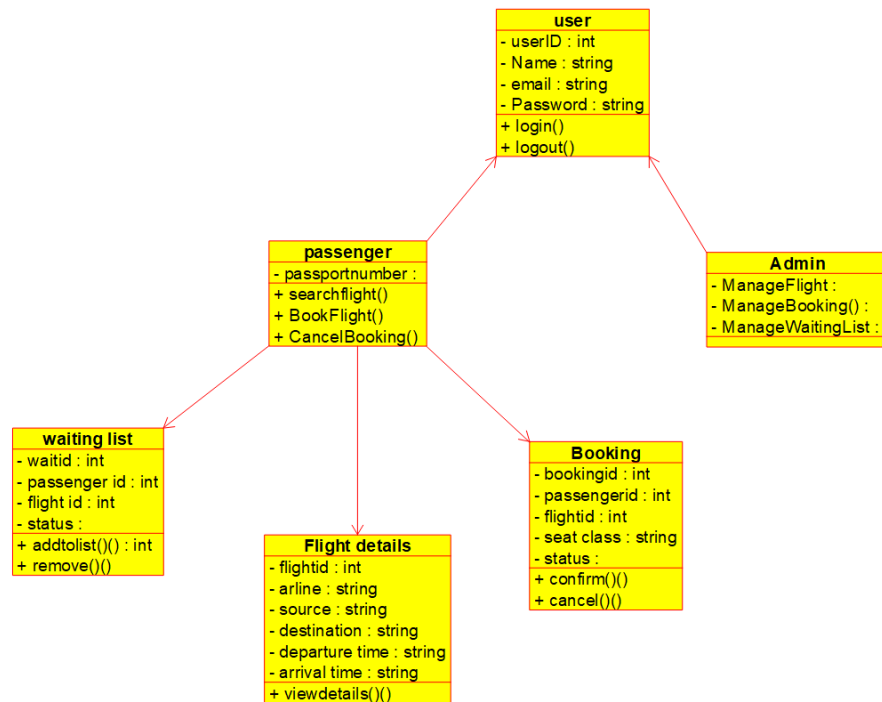
Aim:

The aim is to design a class diagram for an Online Airline Reservation System. The system allows users to search flights, book tickets, and manage bookings, while admins can manage flight classes, waiting lists, holds, and deadlines.

Procedure:

1. **Identify the actors:**
 - The system involves two primary actors: **User** and **Admin**.
2. **Identify the main classes:**
 - **User Class:** Represents customers who can log in, search for flights, and book tickets.
 - **Admin Class:** Represents administrative users who manage flight details, waiting lists, holds, and deadlines.
 - **Flight Class:** Represents the details of individual flights.
 - **Booking Class:** Represents flight bookings made by users.
3. **Define relationships between classes:**
 - **User** and **Booking** have a one-to-many relationship since a user can have multiple bookings.
 - **Admin** inherits from **User**, as an admin is a special type of user with extra privileges.
 - **Flight** and **Booking** have a many-to-one relationship because multiple bookings can be made for the same flight.
4. **Define methods and attributes for each class:**
 - Add essential attributes (e.g., userID, flightID, etc.) and methods (e.g., login(), bookTicket()).
5. **Draw the class diagram:**
 - Represent each class with its attributes and methods.
 - Use arrows to define relationships (inheritance, associations).

Output: Class Diagram



Result:

By following the procedure, a class diagram for the **Online Airline Reservation System** is generated. The diagram will clearly show how **users** and **admins** interact with **flights**, **bookings**, and **system functionalities**. The **admin** has extended control over flight details, while **users** can interact with the system to book flights and manage their reservations. This clear structure enables better system management and functionality implementation for both users and administrators.