Airline reservation system

INTRODUCTION:

Creating an airline reservation system in Java can be a great project to understand object-oriented programming, file handling, and user interfaces. Below is a basic outline to help you get started with such a project. I'll break it down into key components:

Before diving into code, outline what features your system will have. A basic airline reservation system might include:

- User registration and login
- Flight search and booking
- View booked flights
- Cancel bookings
- Admin functionality to add, remove, or update flights

CODE:

```
import java.util.*;


class Flight {     private String
flightNumber;     private String
origin;     private String
destination;     private Date
departureTime;     private Date
arrivalTime;     private int
availableSeats;


   // Constructor
   public Flight(String flightNumber, String origin, String destination, Date departureTime, Date
arrivalTime, int availableSeats) {         this.flightNumber = flightNumber;
```

```
        this.origin = origin;
this.destination = destination;
this.departureTime = departureTime;
this.arrivalTime = arrivalTime;
this.availableSeats = availableSeats;
   }

   // Getters and setters
   // ... (omitted for brevity)

   @Override     public
String toString() {
return "Flight{" +
            "flightNumber='" + flightNumber + '\'' +
            ", origin='" + origin + '\'' +
            ", destination='" + destination + '\'' +
            ", departureTime=" + departureTime +
            ", arrivalTime=" + arrivalTime +
            ", availableSeats=" + availableSeats +
            '}';
   }
}

class Passenger {     private String
passengerName;     private String
passportNumber;     private String
phoneNumber;
   private String email;

   // Constructor
```

```
    public Passenger(String passengerName, String passportNumber, String phoneNumber, String
email) {        this.passengerName = passengerName;        this.passportNumber =
passportNumber;        this.phoneNumber = phoneNumber;
        this.email = email;
    }

    // Getters and setters
    // ... (omitted for brevity)

    @Override     public
String toString() {
return "Passenger{" +
                "passengerName='" + passengerName + '\'' +
                ", passportNumber='" + passportNumber + '\'' +
                ", phoneNumber='" + phoneNumber + '\'' +
                ", email='" + email + '\'' +
                '}';
    }
}

class Reservation {
    private Flight flight;
    private List<Passenger> passengers;

    // Constructor     public Reservation(Flight flight,
List<Passenger> passengers) {
        this.flight = flight;
        this.passengers = passengers;
    }

    // Getters and setters
    // ... (omitted for brevity)

    @Override
    public String toString() {
return "Reservation{" +
                "flight=" + flight +
                ", passengers=" + passengers +
                '}';
    }
}

class AirlineReservationSystem {
    private List<Flight> flights;     private
List<Reservation> reservations;     //
Constructor     public
AirlineReservationSystem() {        flights
= new ArrayList<>();        reservations =
new ArrayList<>();
    }

    // Methods for managing flights, passengers, and reservations
    // ... (implementations omitted for brevity)

    public static void main(String[] args) {
        AirlineReservationSystem system = new AirlineReservationSystem();

        // Add flights, passengers, and create reservations
        // ... (example usage)

        // Display available flights, search for flights, make reservations, etc.
        // ... (implement user interface and logic)
    }
}
```