**Flight.java**

public class Flight {

private int flightId;

private String origin;

private String destination;

private int totalSeats;

private int availableSeats;

public Flight(int flightId, String origin, String destination, int totalSeats) {

this.flightId = flightId;

this.origin = origin;

this.destination = destination;

this.totalSeats = totalSeats;

this.availableSeats = totalSeats;

}

public int getFlightId() {

return flightId;

}

public String getOrigin() {

return origin;

}

public String getDestination() {

return destination;

}

public int getAvailableSeats() {

return availableSeats;

}

public void bookSeat() {

if (availableSeats > 0) {

availableSeats--;

} else {

System.out.println("No available seats.");

}

}

public void cancelSeat() {

if (availableSeats < totalSeats) {

availableSeats++;

}

}

@Override

public String toString() {

return "Flight [ID=" + flightId + ", Origin=" + origin + ", Destination=" + destination + ", Available Seats=" + availableSeats + "]";

}

}

**Booking.java**

public class Booking {

private int bookingId;

private int flightId;

private String passengerName;

public Booking(int bookingId, int flightId, String passengerName) {

this.bookingId = bookingId;

this.flightId = flightId;

this.passengerName = passengerName;

}

public int getBookingId() {

return bookingId;

}

public int getFlightId() {

return flightId;

}

public String getPassengerName() {

return passengerName;

}

@Override

public String toString() {

return "Booking [ID=" + bookingId + ", Flight ID=" + flightId + ", Passenger Name=" + passengerName + "]";

}

}

**Admin.java**

import java.util.ArrayList;

public class Admin {

private ArrayList<Flight> flights;

public Admin() {

flights = new ArrayList<>();

}

public void addFlight(Flight flight) {

flights.add(flight);

}

public void removeFlight(int flightId) {

flights.removeIf(flight -> flight.getFlightId() == flightId);

}

public ArrayList<Flight> getFlights() {

return flights;

}

}

**AirlineReservationSystem.java**

import java.util.ArrayList;

import java.util.Scanner;

public class AirlineReservationSystem {

private static Admin admin = new Admin();

private static ArrayList<Booking> bookings = new ArrayList<>();

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

while (true) {

System.out.println("\nAirline Reservation System");

System.out.println("1. Admin Module");

System.out.println("2. User Module");

System.out.println("3. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

adminModule();

break;

case 2:

userModule();

break;

case 3:

System.out.println("Exiting the application.");

System.exit(0);

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

private static void adminModule() {

System.out.println("\nAdmin Module");

System.out.println("1. Add Flight");

System.out.println("2. Remove Flight");

System.out.println("3. View Flights");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

addFlight();

break;

case 2:

removeFlight();

break;

case 3:

viewFlights();

break;

default:

System.out.println("Invalid choice.");

}

}

private static void addFlight() {

System.out.print("Enter Flight ID: ");

int id = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter Origin: ");

String origin = scanner.nextLine();

System.out.print("Enter Destination: ");

String destination = scanner.nextLine();

System.out.print("Enter Total Seats: ");

int totalSeats = scanner.nextInt();

Flight flight = new Flight(id, origin, destination, totalSeats);

admin.addFlight(flight);

System.out.println("Flight added successfully.");

}

private static void removeFlight() {

System.out.print("Enter Flight ID to remove: ");

int id = scanner.nextInt();

admin.removeFlight(id);

System.out.println("Flight removed successfully.");

}

private static void viewFlights() {

System.out.println("\nAvailable Flights:");

for (Flight flight : admin.getFlights()) {

System.out.println(flight);

}

}

private static void userModule() {

System.out.println("\nUser Module");

System.out.println("1. Search Flights");

System.out.println("2. Book Flight");

System.out.println("3. Cancel Booking");

System.out.println("4. View Bookings");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

searchFlights();

break;

case 2:

bookFlight();

break;

case 3:

cancelBooking();

break;

case 4:

viewBookings();

break;

default:

System.out.println("Invalid choice.");

}

}

private static void searchFlights() {

System.out.print("Enter Origin: ");

scanner.nextLine(); // Consume newline

String origin = scanner.nextLine();

System.out.print("Enter Destination: ");

String destination = scanner.nextLine();

System.out.println("\nMatching Flights:");

for (Flight flight : admin.getFlights()) {

if (flight.getOrigin().equalsIgnoreCase(origin) && flight.getDestination().equalsIgnoreCase(destination)) {

System.out.println(flight);

}

}

}

private static void bookFlight() {

System.out.print("Enter Flight ID: ");

int flightId = scanner.nextInt();

Flight flight = findFlightById(flightId);

if (flight != null && flight.getAvailableSeats() > 0) {

System.out.print("Enter Booking ID: ");

int bookingId = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter Passenger Name: ");

String passengerName = scanner.nextLine();

Booking booking = new Booking(bookingId, flightId, passengerName);

bookings.add(booking);

flight.bookSeat();

System.out.println("Flight booked successfully.");

} else {

System.out.println("Flight not found or no available seats.");

}

}

private static void cancelBooking() {

System.out.print("Enter Booking ID to cancel: ");

int bookingId = scanner.nextInt();

Booking booking = findBookingById(bookingId);

if (booking != null) {

bookings.remove(booking);

Flight flight = findFlightById(booking.getFlightId());

if (flight != null) {

flight.cancelSeat();

}

System.out.println("Booking cancelled successfully.");

} else {

System.out.println("Booking not found.");

}

}

private static void viewBookings() {

System.out.println("\nCurrent Bookings:");

for (Booking booking : bookings) {

System.out.println(booking);

}

}

private static Flight findFlightById(int flightId) {

for (Flight flight : admin.getFlights()) {

if (flight.getFlightId() == flightId) {

return flight;

}

}

return null;

}

private static Booking findBookingById(int bookingId) {

for (Booking booking : bookings) {

if (booking.getBookingId() == bookingId) {

return booking;

}

}

return null;

}

}