```
import javax.swing.*;

import javax.swing.border.EmptyBorder;

import javax.swing.table.DefaultTableModel;

import java.awt.*;

import java.util.ArrayList;

class Flight {

   String flightNumber;

   String destination;

   String departure;

   String time;

   int seatsAvailable;

   public Flight(String flightNumber, String destination, String departure, String time, int
seatsAvailable) {

      this.flightNumber = flightNumber;

      this.destination = destination;

      this.departure = departure;

      this.time = time;

      this.seatsAvailable = seatsAvailable;

   }

}

class Reservation {

   String flightNumber;

   String customerName;

   int seatsBooked;


   public Reservation(String flightNumber, String customerName, int seatsBooked) {
        this.flightNumber = flightNumber;
        this.customerName = customerName;
        this.seatsBooked = seatsBooked;
    }
}
public class AirlineReservationSystem extends JFrame {
    private final ArrayList<Flight> flights = new ArrayList<>();
    private final ArrayList<Reservation> reservations = new ArrayList<>();
    private DefaultTableModel flightTableModel;
    private DefaultTableModel reservationTableModel;
    public AirlineReservationSystem() {
        setTitle("Airline Reservation System");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(900, 600);
        setLocationRelativeTo(null); // Center the window

        // Sample flights
        flights.add(new Flight("AI101", "New York", "Delhi", "10:00 AM", 100));
        flights.add(new Flight("BA202", "London", "Mumbai", "2:00 PM", 80));
        flights.add(new Flight("SQ303", "Singapore", "Chennai", "6:00 AM", 50));
        // Main layout
        JTabbedPane tabbedPane = new JTabbedPane();

        // Search Flights Panel
        JPanel searchPanel = new JPanel(new BorderLayout(10, 10));
        searchPanel.setBorder(new EmptyBorder(10, 10, 10, 10));

        flightTableModel = new DefaultTableModel(new String[]{"Flight No.", "Destination",
"Departure", "Time", "Seats"}, 0);

        JTable flightTable = new JTable(flightTableModel);

        flightTable.setRowHeight(25);

        populateFlightTable();


        JButton bookFlightButton = new JButton("Book Flight");

        bookFlightButton.setFont(new Font("Arial", Font.BOLD, 14));

        searchPanel.add(new JLabel("Available Flights:", SwingConstants.LEFT),
BorderLayout.NORTH);

        searchPanel.add(new JScrollPane(flightTable), BorderLayout.CENTER);

        searchPanel.add(bookFlightButton, BorderLayout.SOUTH);


        // Reservation Panel

        JPanel reservationPanel = new JPanel(new BorderLayout(10, 10));

        reservationPanel.setBorder(new EmptyBorder(10, 10, 10, 10));

        reservationTableModel = new DefaultTableModel(new String[]{"Flight No.", "Customer
Name", "Seats Booked"}, 0);

        JTable reservationTable = new JTable(reservationTableModel);

        reservationTable.setRowHeight(25);


        reservationPanel.add(new JLabel("Reservations:", SwingConstants.LEFT),
BorderLayout.NORTH);

        reservationPanel.add(new JScrollPane(reservationTable), BorderLayout.CENTER);


        // Add panels to tabbedPane
        tabbedPane.addTab("Search Flights", searchPanel);
        tabbedPane.addTab("View Reservations", reservationPanel);

        // Add tabbedPane to JFrame
        add(tabbedPane);

        // Book Flight Button Action Listener
        bookFlightButton.addActionListener(e -> {
            int selectedRow = flightTable.getSelectedRow();
            if (selectedRow == -1) {
                JOptionPane.showMessageDialog(this, "Please select a flight to book.", "Error",
JOptionPane.ERROR_MESSAGE);
                return;
            }

            String flightNumber = (String) flightTableModel.getValueAt(selectedRow, 0);
            String customerName = JOptionPane.showInputDialog(this, "Enter your name:");
            if (customerName == null || customerName.trim().isEmpty()) {
                JOptionPane.showMessageDialog(this, "Name cannot be empty.", "Error",
JOptionPane.ERROR_MESSAGE);
                return;
            }

            int seatsToBook;
            try {
                seatsToBook = Integer.parseInt(JOptionPane.showInputDialog(this, "Enter number of
seats:"));
```

```
                if (seatsToBook <= 0) {
                    throw new NumberFormatException();
                }
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(this, "Invalid number of seats.", "Error",
JOptionPane.ERROR_MESSAGE);
                return;
            }

            Flight flight = flights.stream().filter(f ->
f.flightNumber.equals(flightNumber)).findFirst().orElse(null);
            if (flight != null) {
                if (seatsToBook <= flight.seatsAvailable) {
                    flight.seatsAvailable -= seatsToBook;
                    reservations.add(new Reservation(flightNumber, customerName, seatsToBook));
                    populateFlightTable();
                    populateReservationTable();
                    JOptionPane.showMessageDialog(this, "Booking successful!", "Success",
JOptionPane.INFORMATION_MESSAGE);
                } else {
                    JOptionPane.showMessageDialog(this, "Insufficient seats available.", "Error",
JOptionPane.ERROR_MESSAGE);
                }
            }
        });

        setVisible(true);
```

```
    }

    private void populateFlightTable() {

        flightTableModel.setRowCount(0);

        for (Flight flight : flights) {

            flightTableModel.addRow(new Object[]{flight.flightNumber, flight.destination,
flight.departure, flight.time, flight.seatsAvailable});

        }

    }


    private void populateReservationTable() {

        reservationTableModel.setRowCount(0);

        for (Reservation reservation : reservations) {

            reservationTableModel.addRow(new Object[]{reservation.flightNumber,
reservation.customerName, reservation.seatsBooked});

        }

    }


    public static void main(String[] args) {

        SwingUtilities.invokeLater(AirlineReservationSystem::new);

    }

}
```