A
Project Report
On

"AIRLINE RESERVATION SYSTEM"

Submitted To

BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY), PUNE
In the partial fulfillment for requirement of the degree

BACHELOR OF COMPUTER APPLICATION
(BCA-III SEM-IV 2023-24)

Submitted By

Ms. Jiya Dastgir Shaikh

Ms.Pooja Prabhans Sharma

Under the guidance of

Dr.Suvarna Patil

Through The

INSTITUTE OF MANAGEMENT AND RURAL DEVELOPMENT
ADMINISTRATION, SANGLI

Bharati Vidyapeeth Deemed University
Institute of Management & Rural Development
Administration, Sangli-416416

CERTIFICATE

This is certified that a Project Report titled **“Airline Reservation System”** Submitted By Jiya Shaikh And Pooja Sharma **BCA Sem-IV** in partial fulfillment for requirement of the degree of ‘Bachelor of Computer Application’ submitted to Bharati Vidyapeeth Deemed University, Pune has been completed under my guidance.

To the best of my knowledge and belief matter presented by them is original in nature and has not been copied down from any sources.

Place: Sangli

Date:

Dr.Suvarna Patil	Dr. R. S. Pujari	Dr. P. P. Jamsandekar
(Guide)	(HoD – Computer Applications)	(Director)

Examiner :

Declaration

TO,

THE DIRECTOR;
BVDU IMRDA, Sangli.

Respected sir,

We the undersigned hereby declare that the Project report entitled,

“Airline Reservation System” written and submitted under the guidance of **Dr.Suvarna Patil** is our Original Work. The Empirical findings in this report are based on information collected by us. We have not copied from any report submitted to BVDU IMRDA, Sangli. We understand any such copying is such liable in a punishment in a way that the university authority demits this.

PLACE: Sangli

DATE*:*

Ms. Jiya Dastgir Shaikh

Ms.Pooja Prabhans Sharma

Acknowledgement

We take this opportunity to express our deep sense of obligation to the Bharati Vidyapeeth Deemed University, Pune. The Institute of Management and Rural Development Administration Sangli.

We express our sincere thanks to the **Director Dr. P. P. Jamsandekar** for giving us an opportunity to undertaking this Project work. We owe our deepest gratitude towards respective guide Dr. **Suvarna Patil** and all of our faculty members for their valuable guidance and motivation during the completion of the Project Report.

Finally We very much thankful to everyone who helped us a lot in the completion of this Report to the great extent.

Thanking you all.

Place: Sangli.

Date:-

Ms. Jiya Dastgir Shaikh

Ms.Pooja Prabhans Sharma

INDEX

Sr. No.	Title	Page No.
1	Introduction to Project	6
2	Existing System	7
3	Proposed System	8
4	Objectives	9
5	System Requirement	10
6	System Flow	11
7	Input Design	12
8	Code	16
9	Advantages	38
10	Limitations	39
11	Conclusion	40
12	Bibliography	41

Introduction

We chose to create an airline reservation system project in Java because we wanted to solve a real-world problem that many people face when booking flights. Our main motive behind this project is to make it easier for students, travelers, and anyone else who needs to book flights to do so efficiently and conveniently.

Imagine you want to plan a trip to visit your family or go on a vacation. Booking flights can be a complicated process, especially if you have to visit multiple websites or call different airlines to find the best deals and seats. That's where our airline reservation system comes in!

With our project, you'll be able to easily book your tickets all in one place. No more jumping between websites or waiting on hold with airlines – our system streamlines the entire process for you.

We believe that by creating this airline reservation system, we can help make travel planning simpler and more accessible for everyone. Whether you're a student planning a study abroad trip or a family looking to visit relatives, our project aims to make booking flights a stress-free experience.

Existing System

In the existing system, customers face several challenges when trying to book airline tickets:

1. Manual Process: Customers have to visit various travel agencies or airline offices to inquire about flight details and book tickets. This manual process consumes a lot of time and effort.

2. Limited Accessibility: Not all customers can easily access travel agencies or airline offices, especially if they are located far away or in remote areas.

3. Difficulty in Information Retrieval:Customers may not always get the desired information from these offices, leading to confusion or misguidance regarding their travel plans.

4. Inefficient Record-keeping: Booking details and payment receipts are maintained manually in registers, which can be prone to errors and difficult to manage.

5. Limited Marketing Reach: Advertising through local newspapers or markets has limited reach and may not effectively reach potential customers.

6. Geographical Constraints: The availability of travel services is limited to specific geographical areas, restricting access for customers outside those regions.

The existing system relies on manual processes, lacks accessibility and efficiency, and has limited marketing reach, making it challenging for customers to plan and execute their travel journeys effectively.

Proposed System

1. Simplified Booking Process:The desktop application aims to simplify the booking process for users by providing a user-friendly interface.

2. Automation of Tasks:The application automates various tasks to reduce manual efforts and streamline operations. This includes automated seat allocation,making the booking process faster and more efficient.

3.Enhanced User Experience:User experience is a priority in the desktop application, offering an intuitive interface with easy navigation.

4. Offline Functionality: Users have the option to access certain functionalities of the application offline, allowing them to view their booked flights and make changes to their reservations even when not connected to the internet.

5.Scalability and Flexibility: Designed with scalability and flexibility in mind, the application can adapt to evolving business needs and technological advancements. It is modular in design, allowing for easy expansion and customization to accommodate future growth and changes in market demands.

Objectives

1.Facilitate Booking and Management: The primary objective of the airline reservation system desktop application is to facilitate the seamless booking and management of airline tickets.

2.Enhance User Experience: The desktop application is designed to enhance the user experience by providing a user-friendly interface, intuitive navigation.

3.Empower Users with Information: Similar to a travel management website, theairline reservation system desktop application aims to empower users with comprehensive information about flights.

4.Mobile Responsiveness: While primarily a desktop application, the interface should be responsive and adaptable to different screen sizes, allowing users to access and book flights conveniently from their mobile devices.

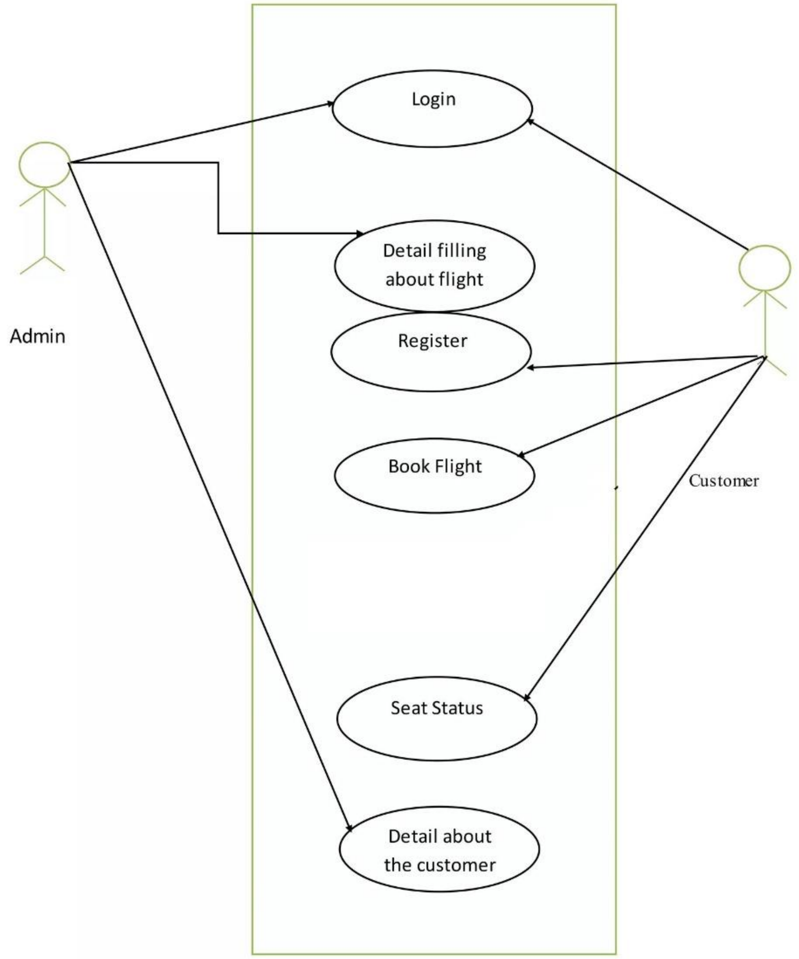
5.Offline Access: Users may not always have access to the internet, so the application should provide offline access to previously viewed bookings and travel information for convenience.

System Requirement

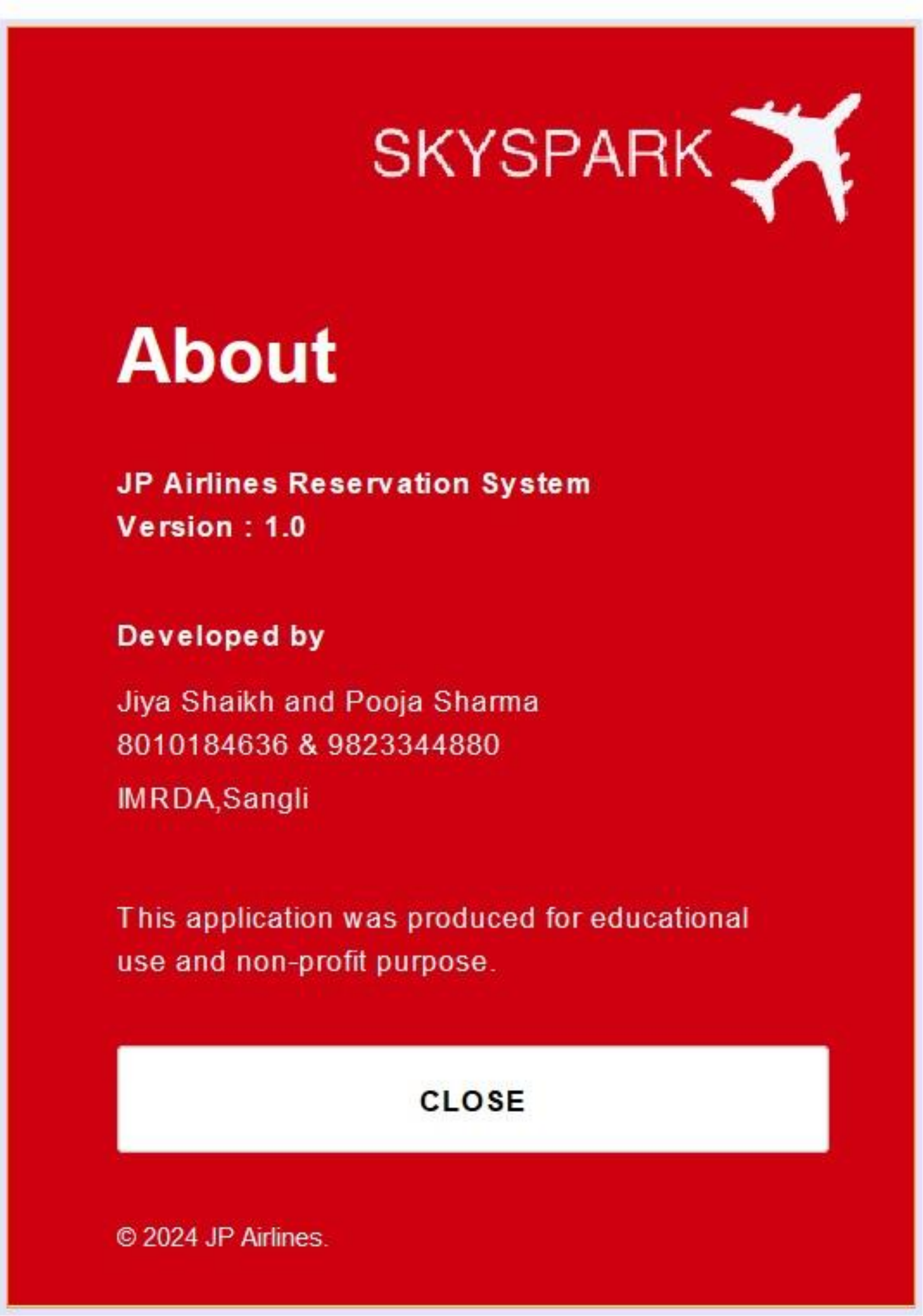
Sr. No.	Name of Resource / Material	Specifications	Quantity	Remark
1	Computer System	Intel(R) Core(TM) i5-3437U CPU, RAM: 8GB, Storage: 256GB SSD, Operating System: Windows 11	1	Required for development and deployment of the application.
2	Integrated Development Environment (IDE)	IntelliJ IDEA, Eclipse, or NetBeans	1	Required for coding, testing, and debugging the application.
3	Java Development Kit (JDK)	Version 8 or higher	1	Required for compiling and running Java code.
4	Documentation Tools	Microsoft Word	1	Required for writing project documentation, user manuals, and technical specifications.
5	Printer and Printing Material	Printer, printing papers, ink, toner	As Required	Optional but useful for printing out reports, tickets, and other documents related to the application

System Flow

Use Case Diagram



Input Design



About

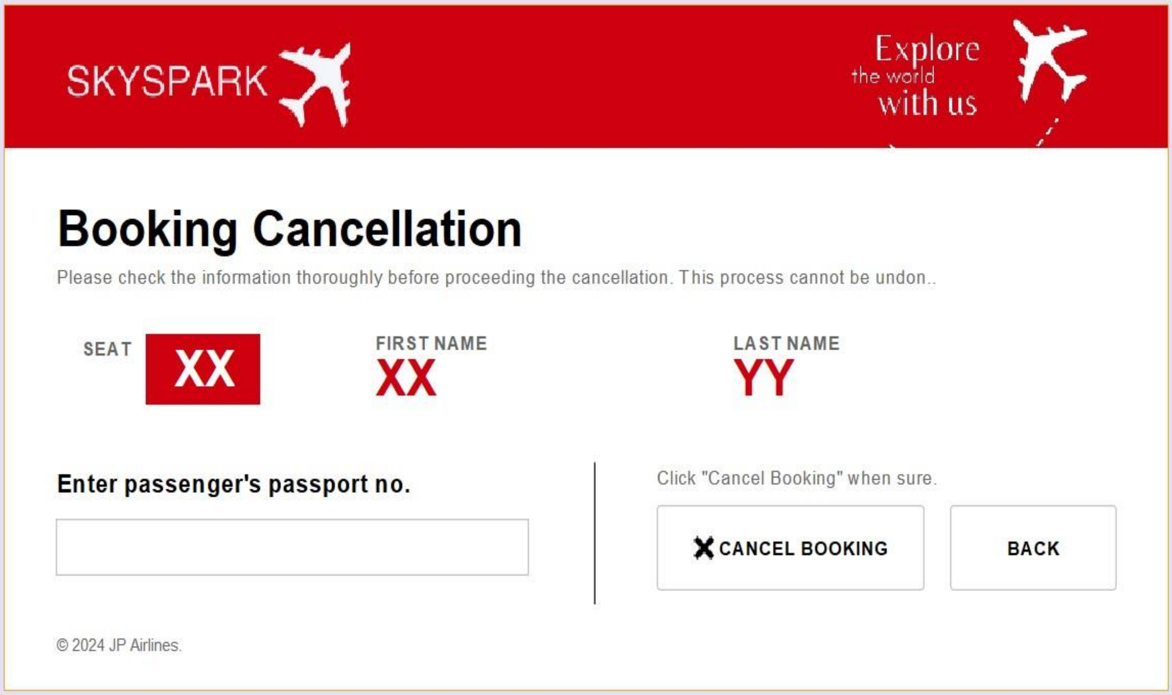
JP Airlines Reservation System
Version : 1.0

Developed by

Jiya Shaikh and Pooja Sharma
8010184636 & 9823344880
IMRDA,Sangli

This application was produced for educational use and non-profit purpose.

© 2024 JP Airlines.



Booking Cancellation

Please check the information thoroughly before proceeding the cancellation. This process cannot be undon..

SEAT XX

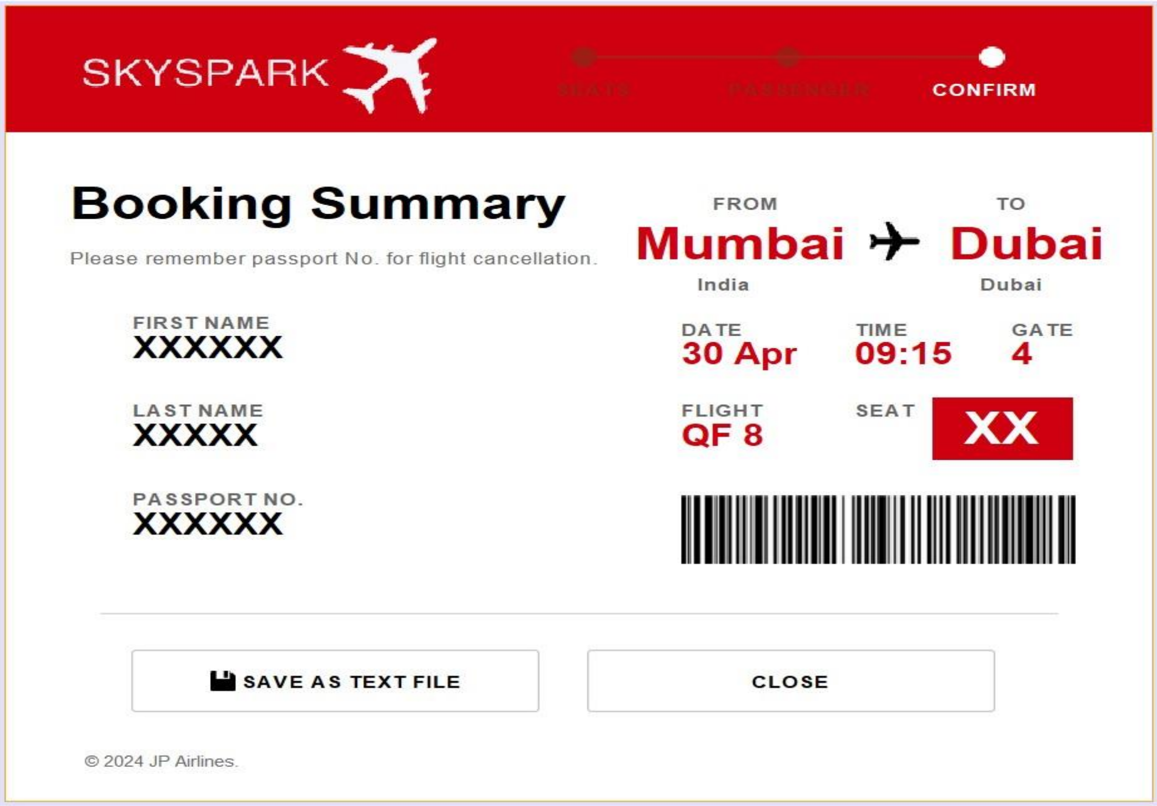
FIRST NAME XX

LAST NAME YY

Enter passenger's passport no.

Click "Cancel Booking" when sure.

© 2024 JP Airlines.



Booking Summary

Please remember passport No. for flight cancellation.

FIRST NAME XXXXXX

LAST NAME XXXXX

PASSPORT NO. XXXXXX

FROM Mumbai India ✈ TO Dubai Dubai

DATE 30 Apr

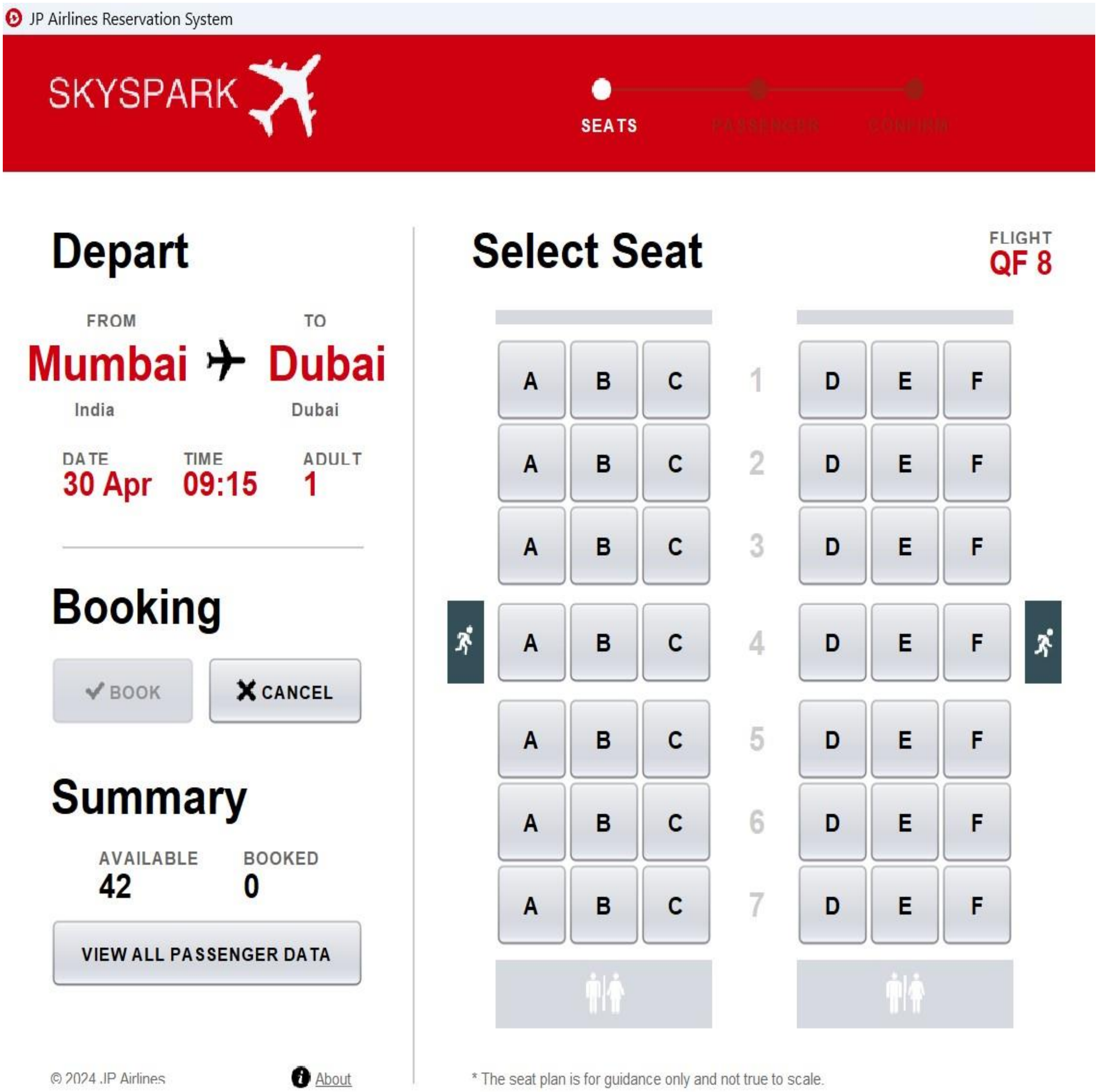
TIME 09:15

GATE 4

FLIGHT QF 8

SEAT XX

© 2024 JP Airlines.



JP Airlines Reservation System

Depart

FROM Mumbai India → TO Dubai Dubai

DATE 30 Apr

TIME 09:15

ADULT 1

Booking

Summary

AVAILABLE 42

BOOKED 0

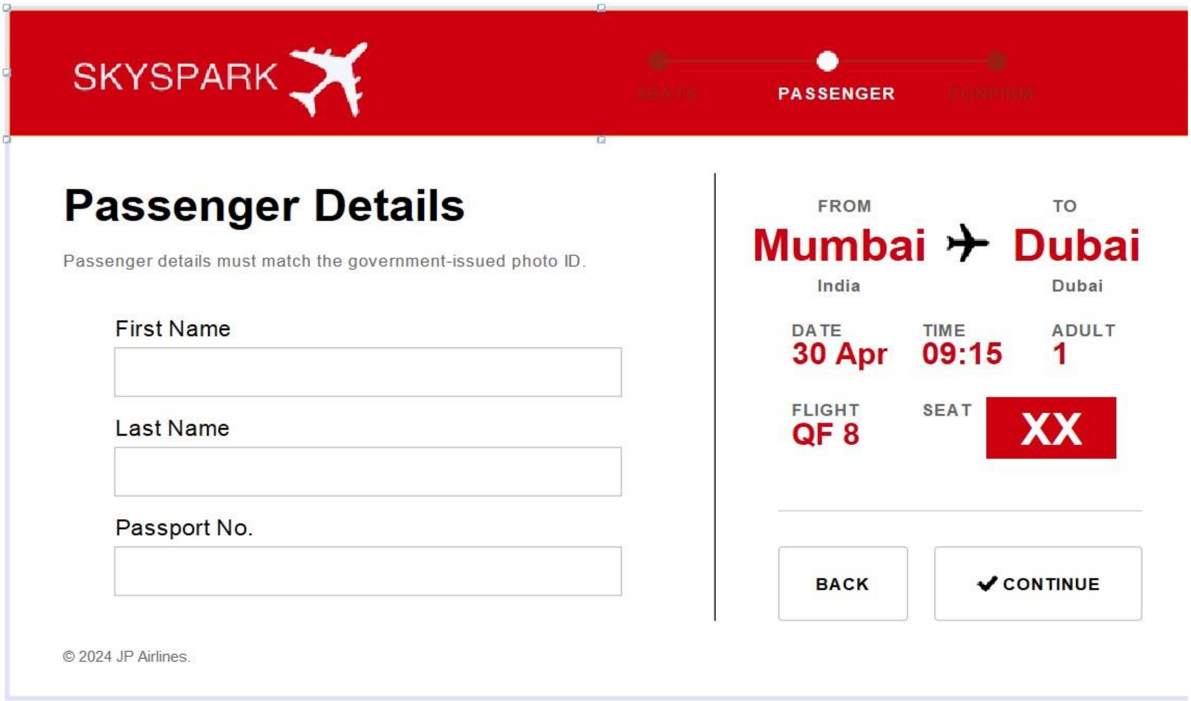
Select Seat

FLIGHT QF 8

A	B	C	1	D	E	F
A	B	C	2	D	E	F
A	B	C	3	D	E	F
A	B	C	4	D	E	F
A	B	C	5	D	E	F
A	B	C	6	D	E	F
A	B	C	7	D	E	F

© 2024 JP Airlines

* The seat plan is for guidance only and not true to scale.



Passenger Details

Passenger details must match the government-issued photo ID.

First Name

Last Name

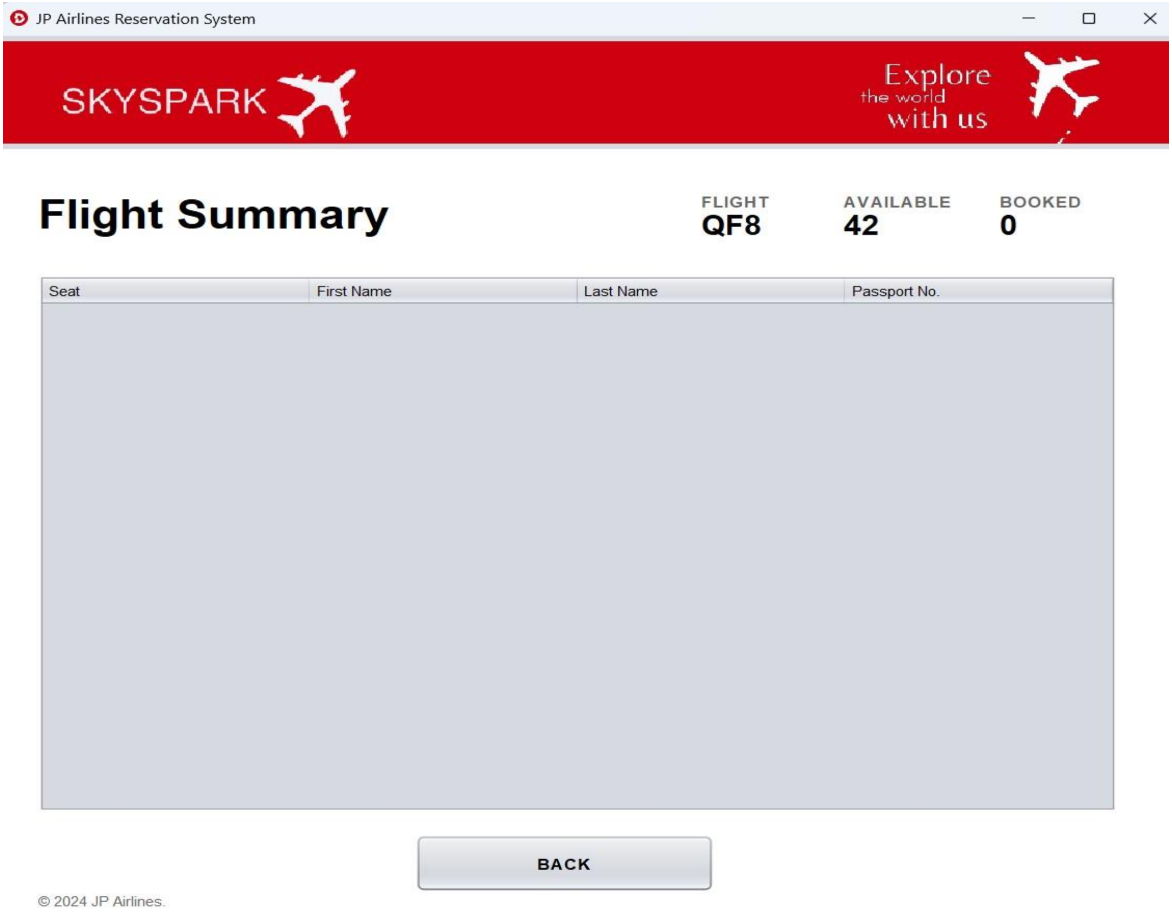
Passport No.

FROM		TO
Mumbai	✈	Dubai
India		Dubai

DATE	TIME	ADULT
30 Apr	09:15	1

FLIGHT	SEAT
QF 8	XX

© 2024 JP Airlines.



Flight Summary

FLIGHT	AVAILABLE	BOOKED
QF8	42	0

Seat	First Name	Last Name	Passport No.

© 2024 JP Airlines.

Code

About.java

```
package seat;

import java.awt.Toolkit;

public class About extends javax.swing.JFrame {
    public About() {
        setIcon();
        initComponents();
    }

    private void CloseButtonActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
    }
    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

        }

       /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new About().setVisible(true);
            }
        });
    }

  private void setIcon() {

setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("appIcon.png"
)));
    }
}
```

Cancellation.java

```java
package seat;

import java.awt.Toolkit;
import javax.swing.JOptionPane;
import static seat.main.flight;

public class Cancellation extends javax.swing.JFrame {

    /**
     * Creates new form Cancellation
     */
    public Cancellation() {
        setIcon();
        initComponents();
    }

    String seat;

    public String getSeat() {
        return seat;
    }

    public void setSeat(String seat) {
        this.seat = seat;
    }

    public Cancellation(String seat) {
        setIcon();
        initComponents();
        SeatTextShow.setText(seat);
        setSeat(seat);
        PassengerFirstNameTextShow.setText(flight.getSeatFirstName(seat));
        PassengerLastNameTextShow.setText(flight.getSeatLastName(seat));
    }

    private void PassportNoFieldActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void ContinueButtonActionPerformed(java.awt.event.ActionEvent evt) {
        String passportNo = PassportNoField.getText();
        if (PassportNoField.getText().isEmpty()) {
            WarningText1.setText("* Passport No. is required");
        } else if (passportNo.equals(flight.getSeatPassportNo(getSeat()))) {
            WarningText1.setText("");
            flight.cancelSeat(getSeat());
            System.out.println("A Seat : " + getSeat() + " has been successfully cancelled.");
            JOptionPane.showMessageDialog(this, "A seat " + getSeat() + " has been
successfully cancelled.", "", JOptionPane.PLAIN_MESSAGE);
            this.setVisible(false);
            new main().setVisible(true);
        } else {
            WarningText1.setText("");
            JOptionPane.showMessageDialog(this, "Incorrect Passport No.", "",
JOptionPane.ERROR_MESSAGE);
        }
    }

    private void CloseButtonActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new main().setVisible(true);
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

        }

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Cancellation().setVisible(true);
            }
        });
    }
    private void setIcon() {

setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("appIcon.png"
)));
    }
```

```
}
```

Confirmation.java

```
package seat;

import java.awt.Toolkit;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.JOptionPane;

public class Confirmation extends javax.swing.JFrame {

public Confirmation() {
        setIcon();
        initComponents();
    }

    public Confirmation(String seat, String firstName, String lastName, String passportNo)
{
        setIcon();
        initComponents();

        SeatTextShow.setText(seat);
        FirstNameShow.setText(firstName);
        LastNameShow.setText(lastName);
        PassportNoShow.setText(passportNo);
        boardingPassPrinter(seat, firstName, lastName, passportNo); //Save boarding pass as
Text File
        main.flight.bookSeat(seat, firstName, lastName, passportNo); //Passing passenger's
data to store in Flight class
    }

    public static void boardingPassPrinter(String seat, String firstName, String lastName,
String passportNo) {
        try {
            FileWriter w = new FileWriter(seat + "_broadingPass.txt");
            w.write("DECIBEL AIRLINES BOARDING PASS\r\n");
            w.write("\r\n");
            w.write("Seat :  " + seat + "\r\n");
            w.write("Name :  " + firstName + " " + lastName + "\r\n");
            w.write("Passport No. :  " + passportNo + "\r\n");
            w.write("\r\nPlease remember your data for flight cancellation. \r\n© 2018 Decibel
Airlines.");
            w.close();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

    private void CloseButtonActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new main().setVisible(true);
    }

    private void PrintButtonActionPerformed(java.awt.event.ActionEvent evt) {
        PrintButton.setText("SAVED");
        PrintButton.setEnabled(false);
        //JOptionPane.showMessageDialog(this, "Boarding pass has been saved.", "",
JOptionPane.PLAIN_MESSAGE);
        System.out.println("#### BOARDING PASS SAVED ####");
    }

    public static void main(String args[]) {
        }

   /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Confirmation().setVisible(true);
            }
        });
    }

    private void setIcon() {

setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("appIcon.png"
)));
    }
}
```

Flight.java

```
package seat;

public class Flight extends Seat {

    private static Seat[] seat = new Seat[42];
    public Flight() { //create an array eg. seat[0] = "A1" ... seat[41] = "F6" to store passenger's data
        String[] label = new String[]{"A", "B", "C", "D", "E", "F"}; //label[0] = "A" ... label[5] = "F"
        for (int i = 1; i <= 6; i++) { //loop for label A-F
            for (int j = 1; j <= 7; j++) { //loop for number 1-7
                int x = (7 * (i - 1)) + (j - 1); //eg. seat[41] = (7 * (6 - 1)) + (7 - 1) means i = 6, j = 7
                this.seat[x] = new Seat(label[i - 1] + "" + j); // Seat("F6") << String seatNo = "F6"
            }
        }
    }

    public boolean[] getAvailableStatus() { //return avaiable status of ALL seats (true = available, false = booked)
        boolean[] availableStatus = new boolean[42];
        for (int i = 0; i < 42; i++) {
            availableStatus[i] = seat[i].isAvailable();
        }
        return availableStatus;
    }

    public int countAvailable() { //return amount of available seats
        int availableCount = 0;
        for (int i = 0; i < 42; i++) {
            if (seat[i].isAvailable()) {
                availableCount++;
            }
        }
        return availableCount;
    }

    public boolean bookSeat(String seatNo, String firstName, String lastName, String passportNo) { //store passenger's data to passenger's seat arrays
        for (int i = 0; i < 42; i++) {
            if (seat[i].getSeatNo().equals(seatNo)) {
                return seat[i].book(firstName, lastName, passportNo);
            }
        }
        return false;
    }

    public void cancelSeat(String seatNo) { //reset THAT seat data to default
        for (int i = 0; i < 42; i++) {
            if (seat[i].getSeatNo().equals(seatNo)) {
                seat[i].reset();
            }
        }
    }

    public boolean getSeatStatus(String seatNo) {  //return avaiable status of THAT seat
(true = available, false = booked)
        for (int i = 0; i < 42; i++) {
            if (seat[i].getSeatNo().equals(seatNo)) {
                return seat[i].isAvailable();
            }
        }
        return false;
    }

    public String getSeatInfo(String seatNo) { //return all data of THAT seat
        for (int i = 0; i < 42; i++) {
            if (seat[i].getSeatNo().equals(seatNo)) {
                return ("#### PASSENGER INFO #### " + " \n Seat: " + seat[i].getSeatNo() + "
\n Full Name: "
                        + seat[i].getFullName() + " \n Passport No: " + seat[i].getPassportNo()
                        + "\n#### successfully booked #### ");
            }
        }
        return "#### PASSENGER INFO : error returning passenger info";
    }

    public String getSeatSeatNo(String seatNo) {
        for (int i = 0; i < 42; i++) {
            if (seat[i].getSeatNo().equals(seatNo)) {
                return (seat[i].getSeatNo());
            }
        }
        return "Error returning getSeatSeatNo";
    }

    public String getSeatFirstName(String seatNo) {
        for (int i = 0; i < 42; i++) {
            if (seat[i].getSeatNo().equals(seatNo)) {
                return (seat[i].getFirstName());
            }
        }
        return "Error returning getSeatFirstName";
    }
    public String getSeatLastName(String seatNo) {
        for (int i = 0; i < 42; i++) {
            if (seat[i].getSeatNo().equals(seatNo)) {
                return (seat[i].getLastName());
            }
        }
        return "Error returning getSeatLastName";
    }

    public String getSeatPassportNo(String seatNo) {
        for (int i = 0; i < 42; i++) {
            if (seat[i].getSeatNo().equals(seatNo)) {
                return (seat[i].getPassportNo());
            }
        }
        return "Error returning getSeatPassportNo";
    }
}
```

RegisterForm.java

```
package seat;

import java.awt.Toolkit;

public class RegisterForm extends javax.swing.JFrame {
public RegisterForm() {
        setIcon();
        initComponents();
    }

    public RegisterForm(String seat) {
        setIcon();
        initComponents();
        SeatTextShow.setText(seat);
    }


    private void PassportNoFieldActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void FirstNameFieldActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void ContinueButtonActionPerformed(java.awt.event.ActionEvent evt) {
        if (FirstNameField.getText().isEmpty() && LastNameField.getText().isEmpty() &&
PassportNoField.getText().isEmpty()) {
            WarningText1.setText("* First Name is Required");
            WarningText2.setText("* Last Name is Required");
            WarningText3.setText("* Passport No. is Required");
        } else if (FirstNameField.getText().isEmpty() &&
LastNameField.getText().isEmpty()) {
            WarningText1.setText("* First Name is Required");
            WarningText2.setText("* Last Name is Required");
            WarningText3.setText("");
        } else if (FirstNameField.getText().isEmpty() &&
PassportNoField.getText().isEmpty()) {
            WarningText1.setText("* First Name is Required");
            WarningText2.setText("");
            WarningText3.setText("* Passport No. is Required");
        } else if (LastNameField.getText().isEmpty() &&
PassportNoField.getText().isEmpty()) {
            WarningText1.setText("");
            WarningText2.setText("* Last Name is Required");
            WarningText3.setText("* Passport No. is Required");
        } else if (FirstNameField.getText().isEmpty()) {
            WarningText1.setText("* First Name is Required");
            WarningText2.setText("");
            WarningText3.setText("");
        } else if (LastNameField.getText().isEmpty()) {
            WarningText1.setText("");
            WarningText2.setText("* Last Name is Required");
            WarningText3.setText("");
        } else if (PassportNoField.getText().isEmpty()) {
            WarningText1.setText("");
            WarningText2.setText("");
            WarningText3.setText("* Passport No. is Required");
        } else {
            String firstName = FirstNameField.getText();
            String lastName = LastNameField.getText();
            String passportNo = PassportNoField.getText();
            String seat = SeatTextShow.getText();

            new Confirmation(seat, firstName, lastName, passportNo).setVisible(true);
//Passing seat, firstName, lastName, passportNo to ConfirmationForm
            System.out.println(main.flight.getSeatInfo(seat));
            this.setVisible(false);
        }
    }

    private void BackButtonActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new main().setVisible(true);
    }

    public static void main(String args[]) {
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new RegisterForm().setVisible(true);
            }
        });
    }



    private void setIcon() {

setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("appIcon.png"
)));
    }
}
```

Seat.java

```
package seat;

import java.awt.List;
import java.util.ArrayList;
import java.util.Arrays;

public class Seat { //This class managed EACH passenger's seat and controlled by mother
"Flight" Class

    private String seatNo;
    private boolean isAvailable;
    private String firstName;
    private String lastName;
    private String passportNo;

    public Seat() {
    }

    public Seat(String seatNo) {
        this.seatNo = seatNo;
        this.isAvailable = true;
        this.firstName = "";
        this.lastName = "";
        this.passportNo = "";
    }

    public String getSeatNo() {
        return this.seatNo;
    }

    public boolean isAvailable() {
        return this.isAvailable;
    }

    public String getFirstName() {
        return this.firstName;
    }

    public String getLastName() {
        return this.lastName;
    }

    public String getFullName() {
        return this.firstName + " " + this.lastName;
    }

    public String getPassportNo() {
        return this.passportNo;
    }

    public void setSeatNo(String seatNo) {
        this.seatNo = seatNo;
    }

    public boolean book(String firstName, String lastName, String passportNo) {
        if (this.isAvailable) {
            this.isAvailable = false;
            this.firstName = firstName;
            this.lastName = lastName;
            this.passportNo = passportNo;
            return true;
        }
        return false;
    }

    public void reset() {
        this.isAvailable = true;
        this.firstName = "";
        this.lastName = "";
        this.passportNo = "";
    }
}
```

ShowData.java

```
package seat;

import java.awt.Toolkit;
import java.util.ArrayList;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import static seat.main.flight;


public class ShowData extends javax.swing.JFrame {

    DefaultTableModel model;

    public ShowData() {
        setIcon();
        initComponents();
        seatMapShowAvailable();
    }

     private void initComponents() {
    private void CloseButtonActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new main().setVisible(true);
    }

    public static void main(String args[]) {
}
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ShowData().setVisible(true);
            }
        });
    }

   private void setIcon() {

setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("appIcon.png"
)));
    }

    private void seatMapShowAvailable() {
        String availableSeat = Integer.toString(flight.countAvailable());
        String bookedSeat = Integer.toString(42 - flight.countAvailable());
        AvailableTextShow.setText(availableSeat);
        ReservedTextShow.setText(bookedSeat);

        model = (DefaultTableModel) Table.getModel();

        ArrayList<String> seatMap = new ArrayList<String>();
        for (char alph = 'A'; alph <= 'F'; alph++) {
            for (int j = 1; j <= 7; j++) {
                String position = alph + "" + j;
                seatMap.add(position);
            } // create String A1, A2, A3 for passing value
        }

        for (int j = 0; j <= 41; j++) { //Add only booked seat data to JTable
            Vector v = new Vector();
            if (flight.getSeatStatus(seatMap.get(j)) == false) {
                v.add(flight.getSeatSeatNo(seatMap.get(j)));
                v.add(flight.getSeatFirstName(seatMap.get(j)));
                v.add(flight.getSeatLastName(seatMap.get(j)));
                v.add(flight.getSeatPassportNo(seatMap.get(j)));
                model.addRow(v);
            } else;
        }
    }
}
```

Main.java

```
package seat;

import java.awt.Toolkit;
import java.util.ArrayList;
import javax.swing.JButton;

public class main extends javax.swing.JFrame {

    public static Flight flight = new Flight();

    public main() {
        initComponents();
        setIcon();
        summaryText();
        seatMapShowAvailable();

        String aboutUnderlined = "<html><u>About</u></html>";
        AboutText.setText(aboutUnderlined);

    private void initComponents() {

   private void C1ActionPerformed(java.awt.event.ActionEvent evt) {
        seatButtonSelected("C1");
    }

    private void CancelButtonActionPerformed(java.awt.event.ActionEvent evt) {
        seatMapShowBooked();
        BookingButton.setEnabled(true);
        CancelButton.setEnabled(!BookingButton.isEnabled());
        WorkingModeHeader.setText("Canceling");
        System.out.println("CANCELING MODE");
    }

    private void BookingButtonActionPerformed(java.awt.event.ActionEvent evt) {
        seatMapShowAvailable();
        BookingButton.setEnabled(!BookingButton.isEnabled());
        CancelButton.setEnabled(true);
        WorkingModeHeader.setText("Booking");
        System.out.println("BOOKING MODE");
    }

    private void A1ActionPerformed(java.awt.event.ActionEvent evt) {
        seatButtonSelected("A1");
    }

    private void B1ActionPerformed(java.awt.event.ActionEvent evt) {
        seatButtonSelected("B1");
    }
```

```
private void D1ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("D1");
}

private void E1ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("E1");
}

private void F1ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("F1");
}

private void A2ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("A2");
}

private void B2ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("B2");
}

private void C2ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("C2");
}

private void D2ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("D2");
}

private void E2ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("E2");
}

private void F2ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("F2");
}

private void A3ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("A3");
}

private void B3ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("B3");
}

private void C3ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("C3");
}

private void D3ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("D3");
}

private void E3ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("E3");
}

private void F3ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("F3");
}

private void A4ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("A4");
}

private void B4ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("B4");
}

private void C4ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("C4");
}

private void D4ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("D4");
}

private void E4ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("E4");
}

private void F4ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("F4");
}

private void A5ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("A5");
}

private void B5ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("B5");
```

```
}

private void C5ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("C5");
}

private void D5ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("D5");
}

private void E5ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("E5");
}

private void F5ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("F5");
}

private void A6ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("A6");
}

private void B6ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("B6");
}

private void C6ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("C6");
}

private void D6ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("D6");
}

private void E6ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("E6");
}

private void F6ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("F6");
}

private void A7ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("A7");
}
```

```
private void B7ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("B7");
}

private void C7ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("C7");
}

private void D7ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("D7");
}

private void E7ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("E7");
}

private void F7ActionPerformed(java.awt.event.ActionEvent evt) {
    seatButtonSelected("F7");
}

private void ViewAllDataButtonActionPerformed(java.awt.event.ActionEvent evt) {
    new ShowData().setVisible(true);
    System.out.println("PASSENGER SUMMARY LAUNCHED");
    this.setVisible(false);
}

private void MainLogo2MouseClicked(java.awt.event.MouseEvent evt) {
    new About().setVisible(true);
}

private void AboutTextMouseClicked(java.awt.event.MouseEvent evt) {
    new About().setVisible(true);
}
public static void main(String args[]) {

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new main().setVisible(true);
        }
    });
}
```

```
    private void setIcon() {

setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("appIcon.png"
)));
    }

    private void summaryText() {
        String availableSeat = Integer.toString(flight.countAvailable());
        String bookedSeat = Integer.toString(42 - flight.countAvailable());
        AvailableTextShow.setText(availableSeat);
        ReservedTextShow.setText(bookedSeat);
    }

    private void seatMapShowAvailable() {
        ArrayList<JButton> seatButton = new ArrayList<JButton>();
        seatButton.add(A1);
        seatButton.add(A2);
        seatButton.add(A3);
        seatButton.add(A4);
        seatButton.add(A5);
        seatButton.add(A6);
        seatButton.add(A7);
        seatButton.add(B1);
        seatButton.add(B2);
        seatButton.add(B3);
        seatButton.add(B4);
        seatButton.add(B5);
        seatButton.add(B6);
        seatButton.add(B7);
        seatButton.add(C1);
        seatButton.add(C2);
        seatButton.add(C3);
        seatButton.add(C4);
        seatButton.add(C5);
        seatButton.add(C6);
        seatButton.add(C7);
        seatButton.add(D1);
        seatButton.add(D2);
        seatButton.add(D3);
        seatButton.add(D4);
        seatButton.add(D5);
        seatButton.add(D6);
        seatButton.add(D7);
        seatButton.add(E1);
        seatButton.add(E2);
        seatButton.add(E3);
        seatButton.add(E4);
        seatButton.add(E5);
        seatButton.add(E6);
        seatButton.add(E7);
        seatButton.add(F1);
        seatButton.add(F2);
        seatButton.add(F3);
        seatButton.add(F4);
        seatButton.add(F5);
        seatButton.add(F6);
        seatButton.add(F7);

        ArrayList<String> seatMap = new ArrayList<String>();
        for (char alph = 'A'; alph <= 'F'; alph++) {
            for (int j = 1; j <= 7; j++) {
                String position = alph + "" + j;
                seatMap.add(position);
            } // create String A1, A2, A3 for passing value
        }

        for (int j = 0; j <= 41; j++) {
            if (flight.getSeatStatus(seatMap.get(j)) == true) {
                (seatButton.get(j)).setEnabled(true);
            } else {
                (seatButton.get(j)).setEnabled(false);
            }
        }
    }

    private void seatMapShowBooked() {
        ArrayList<JButton> seatButton = new ArrayList<JButton>();
        seatButton.add(A1);
        seatButton.add(A2);
        seatButton.add(A3);
        seatButton.add(A4);
        seatButton.add(A5);
        seatButton.add(A6);
        seatButton.add(A7);
        seatButton.add(B1);
        seatButton.add(B2);
        seatButton.add(B3);
        seatButton.add(B4);
        seatButton.add(B5);
```

```
seatButton.add(B6);
seatButton.add(B7);
seatButton.add(C1);
seatButton.add(C2);
seatButton.add(C3);
seatButton.add(C4);
seatButton.add(C5);
seatButton.add(C6);
seatButton.add(C7);
seatButton.add(D1);
seatButton.add(D2);
seatButton.add(D3);
seatButton.add(D4);
seatButton.add(D5);
seatButton.add(D6);
seatButton.add(D7);
seatButton.add(E1);
seatButton.add(E2);
seatButton.add(E3);
seatButton.add(E4);
seatButton.add(E5);
seatButton.add(E6);
seatButton.add(E7);
seatButton.add(F1);
seatButton.add(F2);
seatButton.add(F3);
seatButton.add(F4);
seatButton.add(F5);
seatButton.add(F6);
seatButton.add(F7);

ArrayList<String> seatMap = new ArrayList<String>();
for (char alph = 'A'; alph <= 'F'; alph++) {
    for (int j = 1; j <= 7; j++) {
        String position = alph + "" + j;
        seatMap.add(position);
    } // create String A1, A2, A3 for passing value
}

for (int j = 0; j <= 41; j++) {
    if (flight.getSeatStatus(seatMap.get(j)) == true) {
        (seatButton.get(j)).setEnabled(false);
    } else {
        (seatButton.get(j)).setEnabled(true);
    }
```

```
        }
    }

    public void seatButtonSelected(String buttonSelected) {
        String position = buttonSelected;
        if (this.flight.getSeatStatus(position) == true) { //Seat is available
            System.out.println("BOOKING process of " + position + " starred.");
            new RegisterForm(position).setVisible(true); //Passing seat to RegisterForm
            this.setVisible(false);
        } else {
            System.out.println("CANCELING process of " + position + " started.");
            new Cancellation(position).setVisible(true);
            this.setVisible(false);
        }
    }
}
```

Advantages

1. Convenience:Users can conveniently research, plan, and book their airline tickets and travel arrangements all within the desktop application, eliminating the need to visit multiple websites or agencies.

2. Time-saving:Users can check seat availability quickly and efficiently using the desktop application, saving valuable time in the booking process.

3. Flexibility: The desktop application allows users to book flights and manage their reservations at any time, without being restricted by the operating hours of travel agencies or call centers.

4.Offline Access: Users can access their booking information and travel details even without an internet connection, providing convenience and peace of mind, especially during travel.

5. No Dependencies: Users are not dependent on external factors such as website uptime or internet connectivity, reducing the risk of disruptions or service outages during the booking process.

6. Efficient Resource Utilization:The desktop application optimizes system resources, leading to faster performance and responsiveness compared to web-based applications, especially in areas with limited internet bandwidth.

7. Consistent User Experience: The desktop application offers a consistent user experience across different devices and operating systems, ensuring usability and familiarity for users regardless of their setup.

Limitations

1. Limited Accessibility: As a desktop application, accessibility may be limited to devices where the application is installed, such as desktop computers or laptops. Users may face challenges accessing the application from mobile devices or when away from their primary workstations, reducing convenience and flexibility.

2. Not Web-Based:Unlike web-based applications, the desktop application may not offer the same level of accessibility and convenience for users who require remote access or travel frequently. Users may need to be physically present at their desktops to access the application, limiting its usability in certain scenarios.

3.Limited Features:Compared to comprehensive travel management platforms or web-based booking systems, the desktop application may offer limited features and functionalities. Users may miss out on advanced features such as dynamic pricing, real-time updates, or integration with external services.

4.No Integrated Payment Options: The desktop application does not provide built-in payment processing capabilities, requiring users to complete payment transactions through alternative methods outside the application. This limitation may inconvenience users who prefer to make secure online payments directly within the booking system, leading to additional steps and potential delays in completing reservations.

5.User Inconvenience: Without integrated payment options, users may experience inconvenience and frustration when navigating between the desktop application and external payment platforms. This disjointed process can lead to user dissatisfaction and potentially deter users from completing bookings.

Conclusion

In conclusion, the development of an airline reservation system desktop application offers numerous benefits and opportunities for enhancing efficiency, improving user experience, and streamlining airline operations.

creating a desktop application for airline reservation offers many advantages. It helps airlines work more efficiently and makes booking flights easier for passengers. Desktop apps are reliable and can work without an internet connection, which is helpful.

But there are also things to consider. Desktop apps may not work on all devices, and they can be harder to update. Despite these challenges, desktop apps offer benefits like extra security and the ability to customize.

Overall, a well-made desktop app can make booking flights smoother for everyone involved. It's important to keep an eye on new technology and what users need to make sure the app stays useful and up-to-date.

Bibliography

Links:

- https://stackoverflow.com/
- https://en.wikipedia.org/wiki/Airline_reservations_system
- https://github.com/SampleDeveloper/SampleDesktopApp
- https://www.geeksforgeeks.org/
- https://airlinegeeks.com/2016/08/16/airline-reservations-systems-a-brief-history/