A

PROJECT REPORT

On

AIRLINE RESERVATION SYSTEM

Submitted to:

GURUGRAM UNIVERSITY

Under the Supervision of:

Ms. Namita
Asst. Professor
Govt. College for Girls,Sector 14, Gurugram

Submitted by:

Name: Smarti

Enroll. No: 23126410008

Class: MCA 2nd semester

Govt. College for Girls, Sector-14, Gurugram

DECLARATION

I hereby declare that the project work entitled “**Airline Reservation System**” is an authenticated work carried out by me at Govt. College for Girls, Sector-14, Gurugram and this work has not been submitted for similar purpose anywhere else.

Date: 03-04-2024	**Student Name:** Smarti
Place: Gurugram	**Enroll. No.:** 23126410008
	Class: MCA 2nd Semester

CERTIFICATE

This is to certify that **Smarti** roll no. **23126410008** has worked under my supervision to prepare her project report. The work embodied in this report is original and was conducted at Govt. College for Girls, Gurugram. She has completed all requirements of MCA ordinance.

Ms. Namita
Asst. Professor
(Project Guide)

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher and project guide **Ms. Namita** for his valuable guidance. I am also thankful to all the faculty members of the Comp. Sc. Dept. for their encouragement and help while collecting information about the project.

Student Name: Smarti

Enroll No.: 23126410008

CONTENTS

PROBLEM STATEMENT

The current airline reservation process is plagued by inefficiencies, inconvenience, and lack of flexibility for both passengers and airlines. Passengers often face challenges in finding available flights, making bookings, managing reservations, and receiving timely updates on flight status. Additionally, airlines struggle to efficiently manage flight schedules, seat availability, and passenger data, leading to operational inefficiencies and revenue loss. The lack of a unified and user-friendly reservation system hampers the overall travel experience and negatively impacts the competitiveness and profitability of airlines in the industry. Therefore, there is a pressing need for a modern and comprehensive airline reservation system that addresses these issues by providing a seamless booking experience for passengers, efficient management tools for airlines, and real-time communication between all stakeholders involved in the travel process.

This project on Flight Management System is the automation of registration process of airline system. The system is able to provide much information like passenger's details, flight details and the booking details.

OBJECTIVE

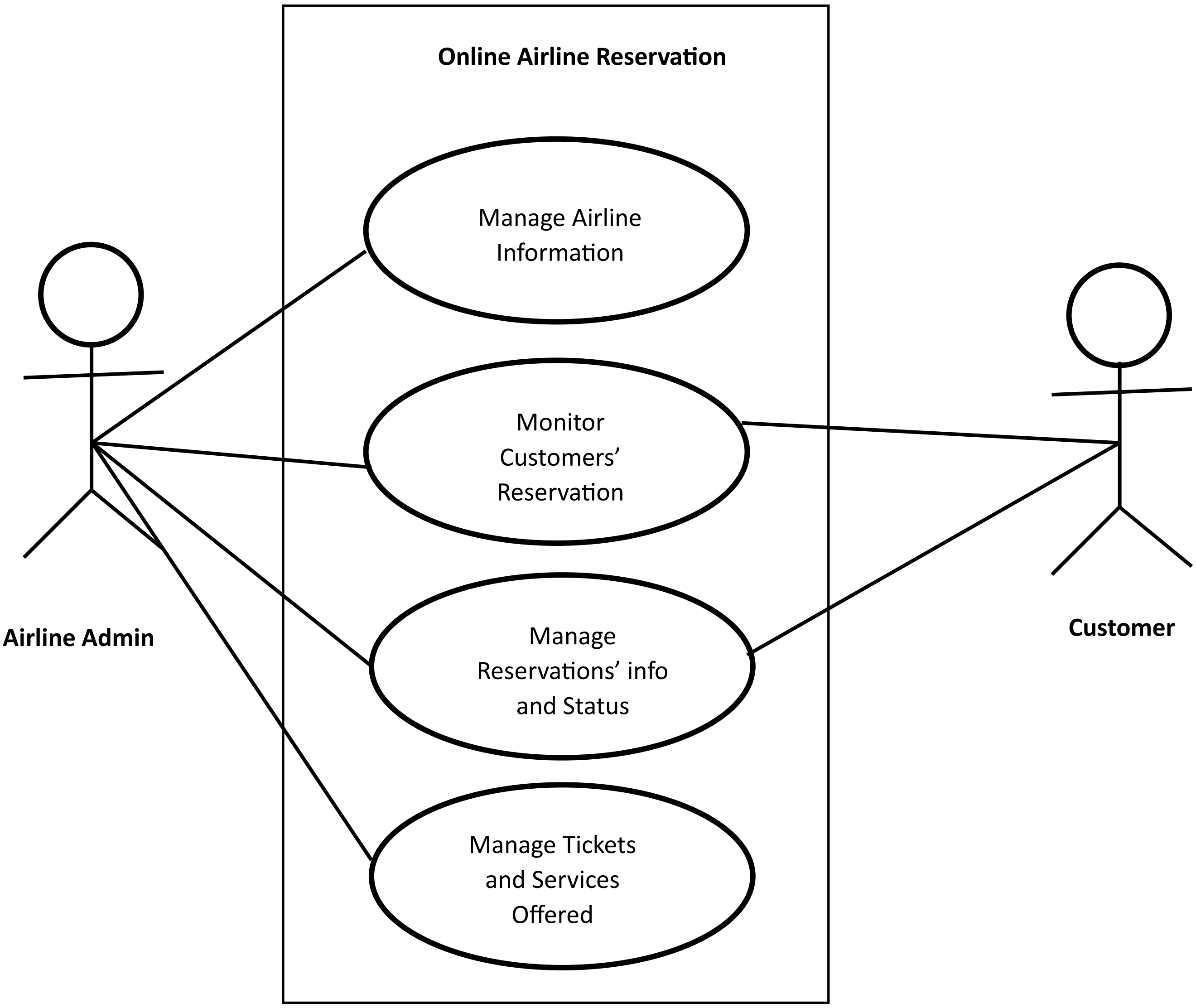
The goal of this project on Airline Management System is to automate the airline registration procedure.

The system includes information such as passenger information, flight information, and a list of all passengers, as well as the ability to store and retrieve data linked to the airline business and conduct air travel transactions.

The objective of the airline reservation system is to provide a user-friendly and efficient platform for passengers to book, manage, and track their flights seamlessly. The system aims to streamline the airline reservation process, allowing passengers to search for available flights, select preferred travel dates and times, choose seating options, make secure payments, and receive confirmation of their bookings in real-time. Additionally, the system facilitates the management of flight schedules, seat availability, and passenger information for airlines, ensuring smooth operations and optimal utilization of resources. Overall, the goal is to enhance the customer experience, improve operational efficiency, and increase revenue for the airline industry.

USE CASE DIAGRAM

1. User Registration/Login:
   - Actors: User
   - Description: Allows users to create an account or log in to the system to access booking features.
2. View Flight Details:
   - Actors: User
   - Description: Allows users to view detailed information about a specific flight, including departure/arrival times, duration, and available seats.
3. Book Flight:
   - Actors: User
   - Description: Allows users to select a flight and book tickets for themselves or others, choosing seat preferences and providing passenger details.
4. Manage Booking:
   - Actors: User
   - Description: Allows users to view and manage their existing bookings, including modifying flight details, adding or removing passengers, or cancel reservations.
5. Flight Comparison:
   - Actors: User
   - Description: Allows users to compare available flights based on criteria such as price, duration, airline, and layovers, helping them make informed decisions when booking flights.
6. Add Extra Services:
   - Actors: User
   - Description: Allows users to add additional services to their bookings, such as in-flight meals, extra baggage, or seat upgrades.
7. Payment Processing:
   - Actors: User, Payment Gateway
   - Description: Handles secure payment processing for flight bookings, integrating with payment gateways to facilitate transactions using various payment methods.
8. Flight Notification Alerts:
   - Actors: User
   - Description: Allows users to subscribe to flight notification alerts, receiving updates via email or SMS regarding changes in flight status, gate changes, or delays.


Online Airline Reservation
Manage Airline Information
Monitor Customers' Reservation
Manage Reservations' info and Status
Manage Tickets and Services Offered
Airline Admin
Customer

PROPOSED SYSTEM

Our proposed system has the following advantages:

- ➢ User friendly interface
- ➢ Fast access to database
- ➢ Less error
- ➢ More Storage Capacity
- ➢ Search facility
- ➢ Look and Feel Environment
- ➢ Quick transaction

All the manual difficulties in managing a Booking flight have been rectified by implementing computerization.

FEASIBILITY ANALYSIS

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

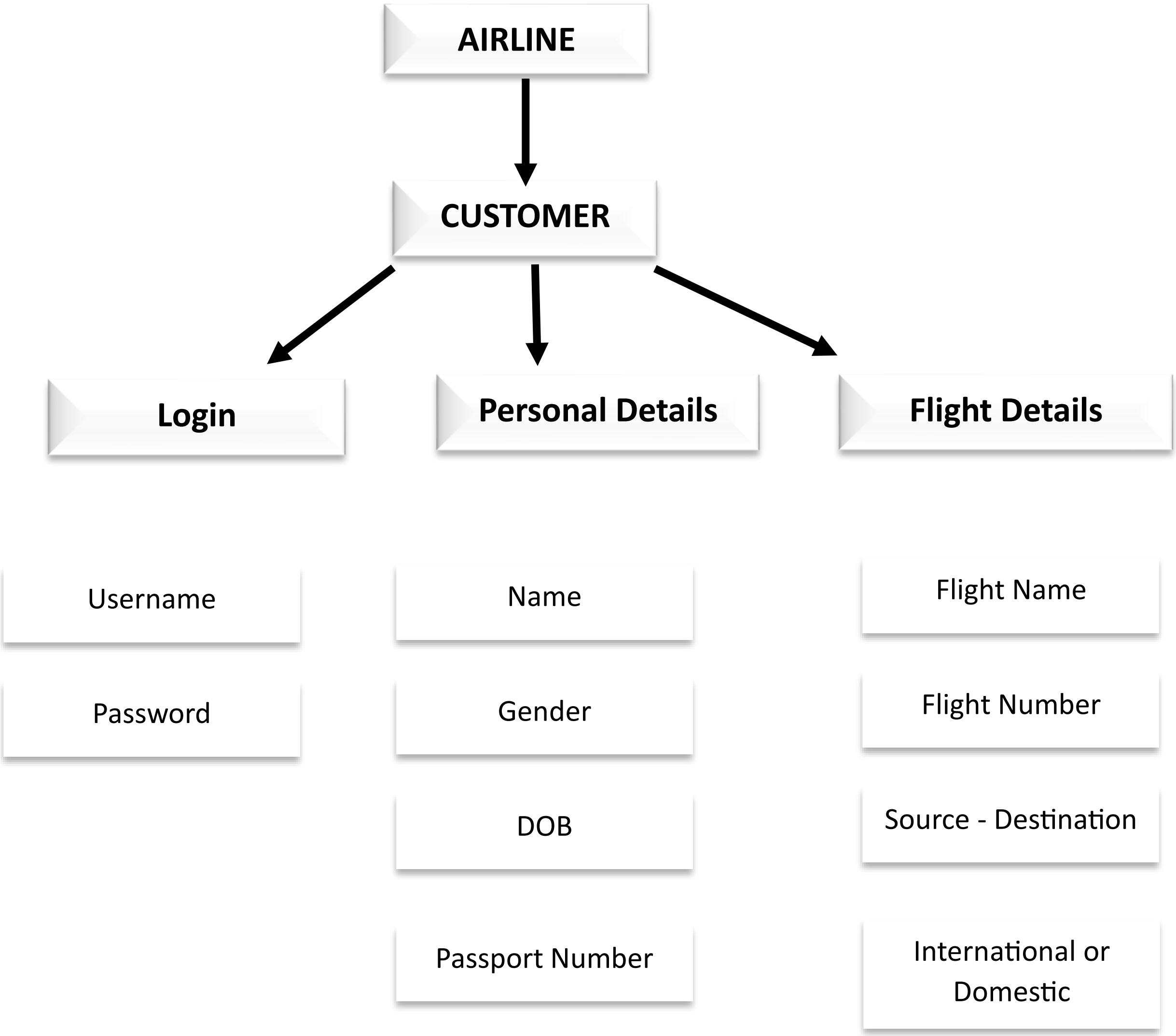
Technical Feasibility:

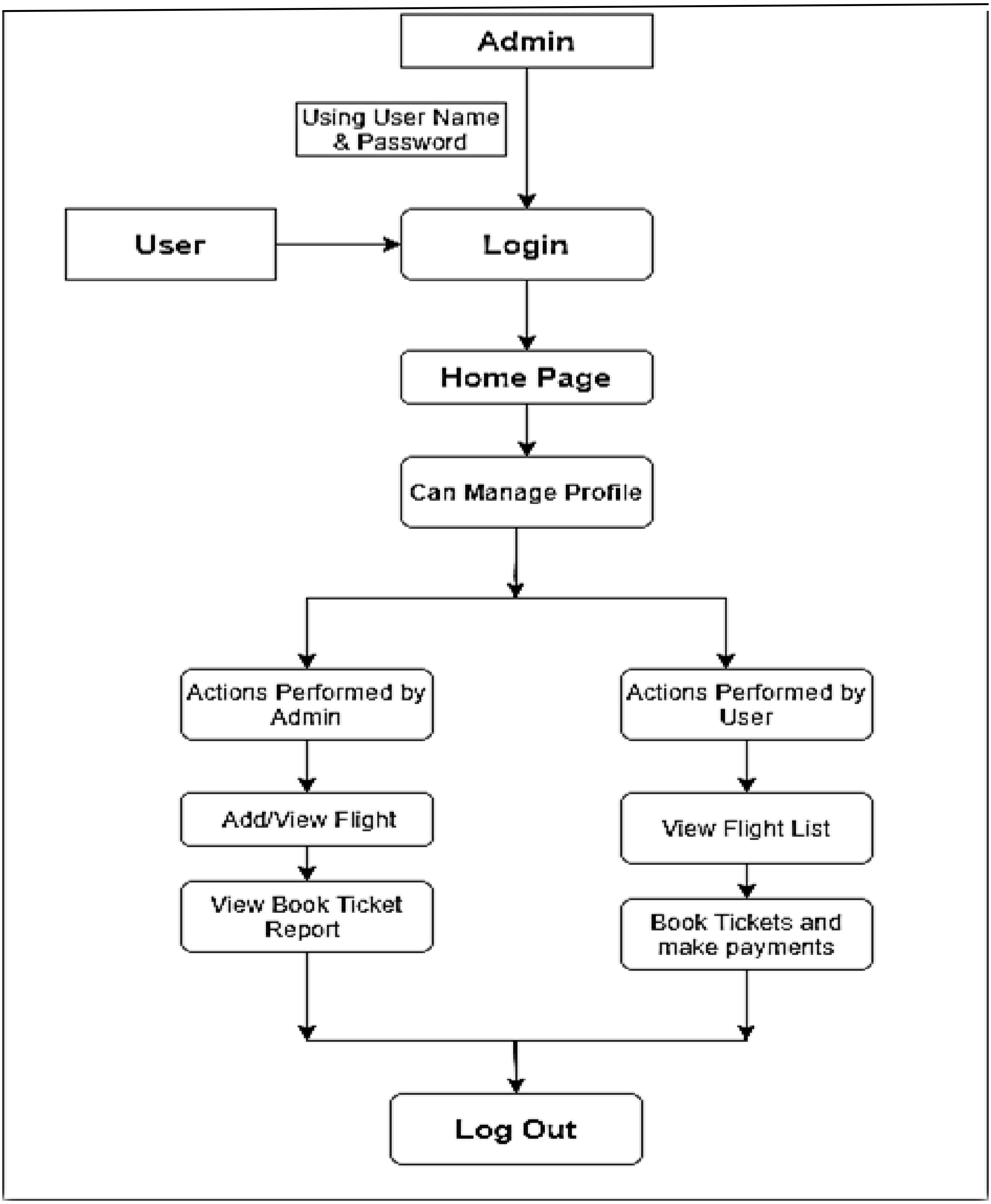
We can strongly says that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

Economical Feasibility:

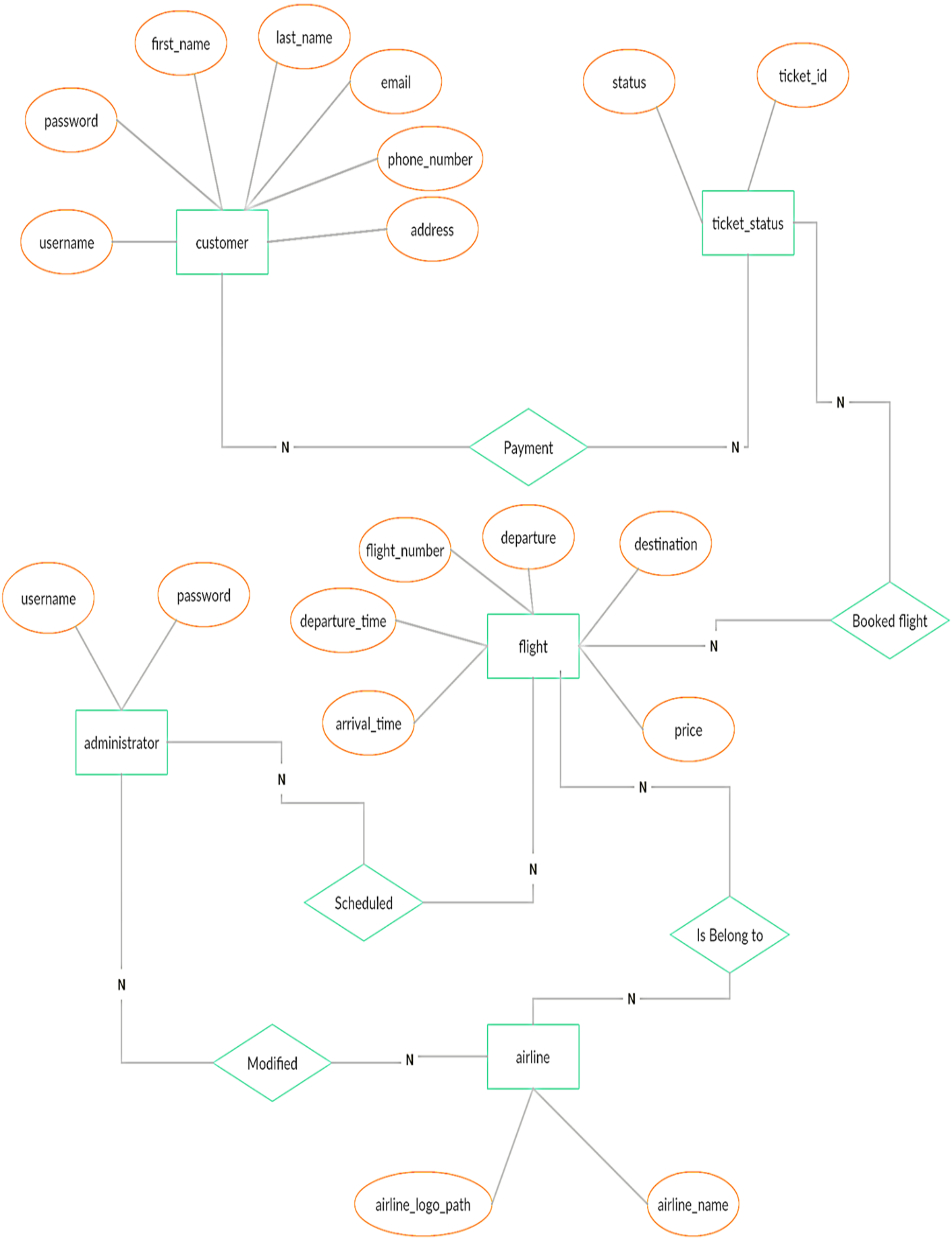
Development of this application is highly economically feasible. The organization needed not spend much m one for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resources. Even after the development, the organization will not be in a condition to invest more in the organization. Therefore, the system is economically feasible.

DFD (DATA FLOW DIAGRAM)




Admin
Using User Name & Password
User
Login
Home Page
Can Manage Profile
Actions Performed by Admin
Add/View Flight
View Book Ticket Report
Actions Performed by User
View Flight List
Book Tickets and make payments
Log Out

ER DIAGRAM



first_name
last_name
email
password
phone_number
username
customer
address
status
ticket_id
ticket_status
N
N
Payment
N
flight_number
departure
destination
username
password
departure_time
flight
N
Booked flight
arrival_time
price
administrator
N
N
N
Scheduled
Is Belong to
N
N
Modified
N
airline
airline_logo_path
airline_name

JAVA SWING

Swing is a Java GUI (Graphical User Interface) toolkit and widget library that allows developers to create sophisticated, platform-independent graphical user interfaces for Java applications. It is part of the Java Foundation Classes (JFC) and provides a rich set of components for building interactive and visually appealing desktop applications.

1. **Platform-Independence**: Swing components are implemented entirely in Java and do not rely on platform-specific native code. This ensures that Swing-based applications can run on any platform that supports Java, such as Windows, macOS, and Linux.

2. **Lightweight Components**: Unlike the earlier Abstract Window Toolkit (AWT) components, Swing components are lightweight and are not tied to the underlying operating system's native GUI controls. This makes Swing components more flexible, customizable, and efficient in terms of resource usage.

3. **Rich Set of Components**: Swing provides a wide range of components to build user interfaces, including buttons, labels, text fields, checkboxes, radio buttons, lists, tables, trees, dialogs, and more. These components are highly customizable and can be combined to create complex UI layouts.

4. **MVC Architecture:** Swing follows the Model-View-Controller (MVC) design pattern, where components (views) are separate from the underlying data (models) and user interactions (controllers). This separation of concerns promotes modularity, reusability, and maintainability of Swing-based applications.

5. **Event-Driven Programming**: Swing applications are event-driven, meaning they respond to user actions (such as button clicks or mouse movements) by triggering event handlers. Developers can register event listeners to handle these events and perform appropriate actions in response.

6. **Layout Managers:** Swing provides layout managers that help organize and position components within containers, such as frames, panels, and dialogs. Layout managers automatically handle resizing and alignment of components, allowing developers to create dynamic and responsive user interfaces.

7. **Customization:** Swing components can be customized extensively using various properties, methods, and listeners. Developers can change the appearance (e.g., colors, fonts) and behavior (e.g., tooltips, drag-and-drop) of components to suit the requirements of their applications.

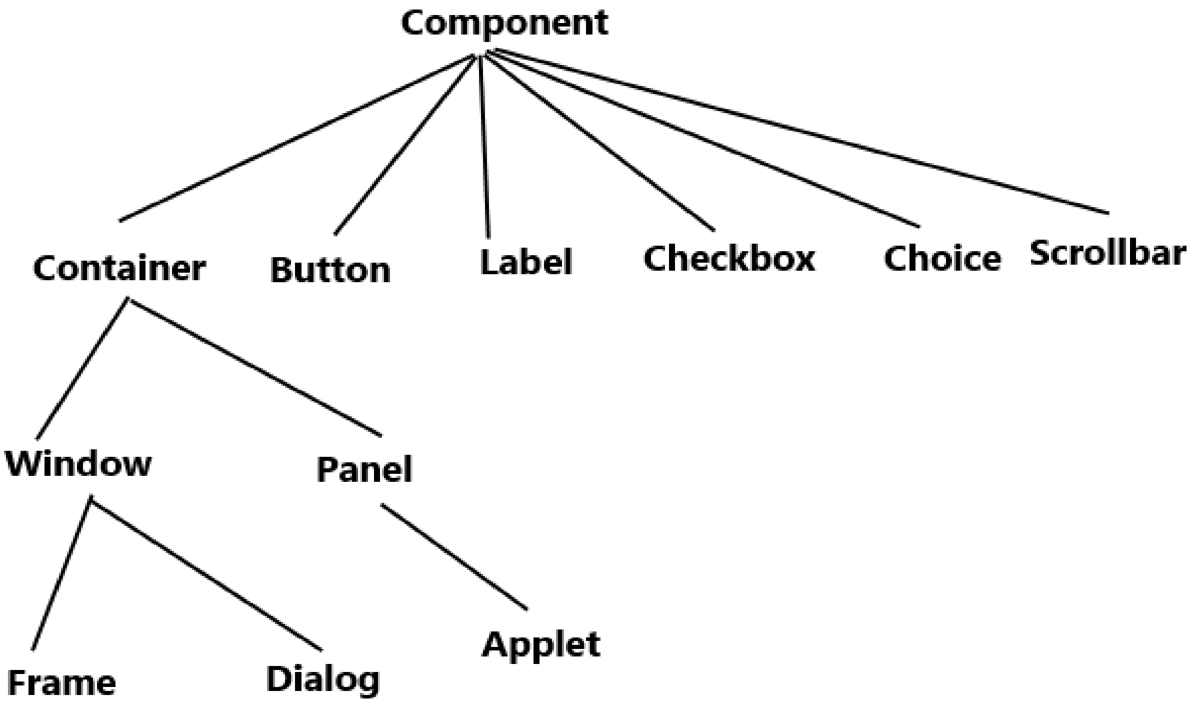
8. **Internationalization and Accessibility:** Swing supports internationalization (i18n) and localization (l10n) features, making it easy to create applications that support multiple languages and locales. Additionally, Swing provides accessibility features to ensure that applications are accessible to users with disabilities.

JAVA AWT

The Abstract Window Toolkit (AWT) supports Graphical User Interface (GUI) programming. AWT features include:

- A set of native user interface components
- A robust event-handling model
- Graphics and imaging tools, including shape, color, and font classes
- Layout managers, for flexible window layouts that do not depend on a particular window size or screen resolution
- Data transfer classes, for cut-and-paste through the native platform clipboard

The AWT was designed to provide a common set of tools for GUI design that could work on a variety of platforms. The tools provided by the AWT are implemented using each platform's native GUI toolkit, hence preserving the look and feel of each platform. This is an advantage of using AWT.



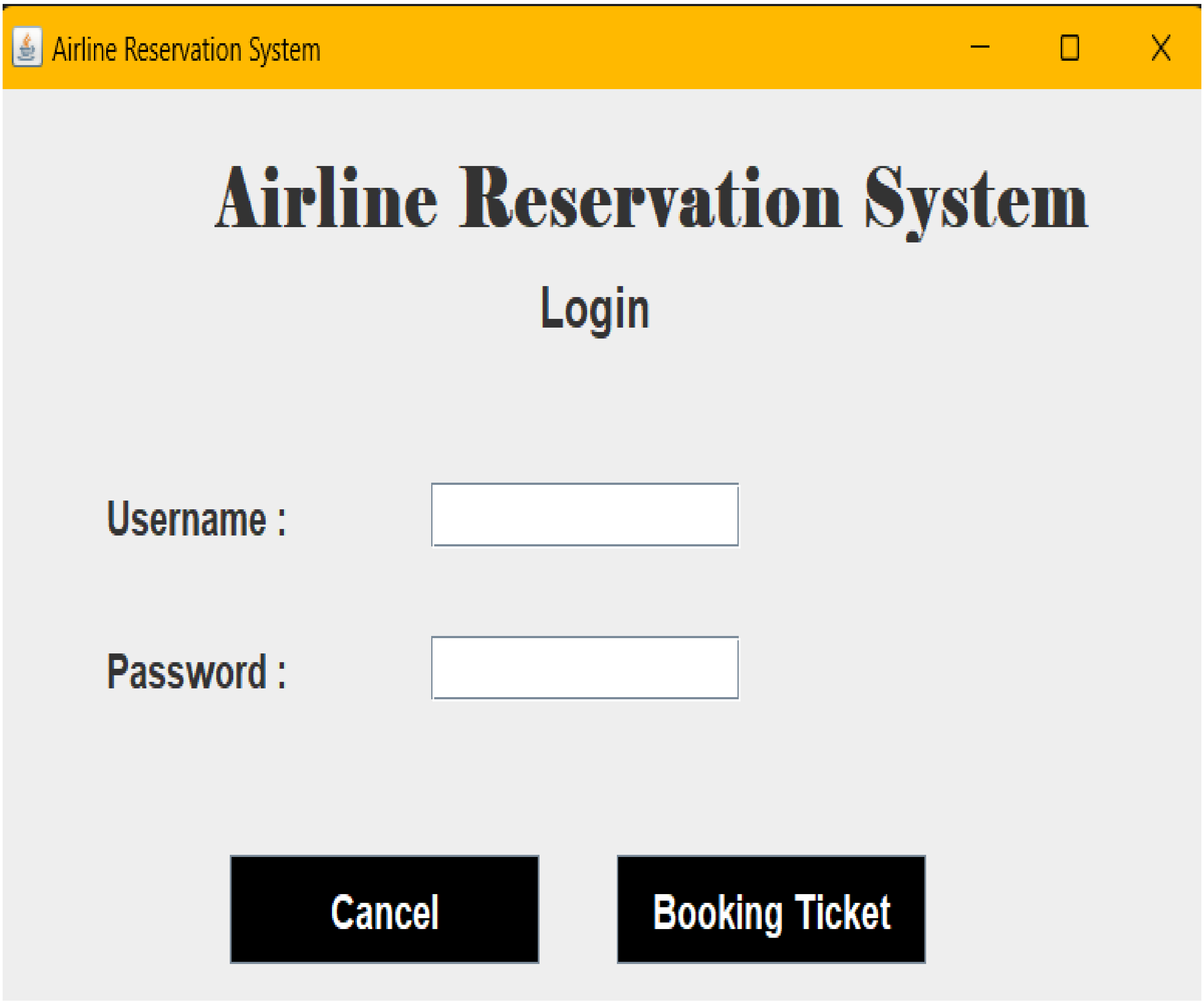
CODE

PAGE 1: LOGIN

```
import javax.swing.*;

import java.awt.event.*;

import java.awt.*;


class Jswindow implements ActionListener

{

        JFrame f;

        JButton next;

        Jswindow()

        {

                f=new JFrame("Airline Reservation System");

        //**TITLE ***

                JLabel Title=new JLabel("Airline Reservation System");

                Title.setFont(new Font("Elephant",Font.BOLD,38));

                Title.setBounds(140,30,600,40);

                f.add(Title);

        //**SUB- TITLE ***

                JLabel SubTitle=new JLabel("Login");

                SubTitle.setFont(new Font("Arial",Font.BOLD,26));

                SubTitle.setBounds(350,80,650,40);

                f.add(SubTitle);

        //**Name ***

                JLabel Name=new JLabel("Username : ");

                Name.setFont(new Font("Raleway",Font.BOLD,22));

                Name.setBounds(70,180,200,30);

                f.add(Name);

                JTextField NameTextField=new JTextField();

                NameTextField.setFont(new Font("Raleway",Font.PLAIN,14));

                NameTextField.setBounds(280,180,200,30);

                f.add(NameTextField);
//**Password ***
        JLabel Password=new JLabel("Password : ");
        Password.setFont(new Font("Raleway",Font.BOLD,22));
        Password.setBounds(70,250,200,30);
        f.add(Password);
        JTextField PasswordTextField=new JTextField();
        PasswordTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        PasswordTextField.setBounds(280,250,200,30);
        f.add(PasswordTextField);

//**Cancel Button ***
        JButton Cancel=new JButton("Cancel");
        Cancel.setBackground(Color.BLACK);
        Cancel.setForeground(Color.WHITE);
        Cancel.setFont(new Font("Raleway",Font.BOLD,22));
        Cancel.setBounds(150,350,200,50);
        Cancel.addActionListener(this);
        f.add(Cancel);

//**Submit Button ***
        JButton Submit =new JButton("Booking Ticket");
        Submit.setBackground(Color.BLACK);
        Submit.setForeground(Color.WHITE);
        Submit.setFont(new Font("Raleway",Font.BOLD,22));
        Submit.setBounds(400,350,200,50);
        Submit.addActionListener(this);
        f.add(Submit);
        f.setBounds(100,140,80,20);
        f.setLayout(null);
        f.setSize(800,600);
        f.setVisible(true);
        f.setLocation(350, 10);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

```
            public void actionPerformed(ActionEvent e)
             {        Page secondFrame=new  Page();
                      f.dispose();
             }
}
class Airline
{
          public static void main(String[] args)
          {
                    new Jswindow();
          }
}
```

OUTPUT



PAGE 2: PERSONAL DETAIL

```
class Page implements ActionListener {

        JFrame f;

        private String dates [] = { "1", "2", "3", "4", "5","6", "7", "8", "9", "10", "11", "12", "13", "14", "15","16",
                        "17", "18", "19", "20","21", "22", "23", "24", "25","26", "27", "28", "29", "30","31" };

        private String months[]= { "Jan", "feb", "Mar", "Apr","May", "Jun", "July", "Aug","Sep", "Oct", "Nov", "Dec" };

        private String years[]= { "1995", "1996", "1997", "1998","1999", "2000", "2001", "2002", "2003", "2004", "2005",
                                "2006","2007", "2008", "2009", "2010","2011", "2012", "2013", "2014","2015", "2016",
                                "2017", "2018","2019" ,"2020", "2021", "2022","2023","2024"};

        private JComboBox date,months,year,lt;

        private String country[]={"Afghanistan","Bangladesh","Canada","China","India","Japan","Malaysia",

                                "Maldives","Pakistan","United Kingdom","United Arab Emirates","United States"};


                Page()

                {

                f=new JFrame("Airline Reservation System");

        //**TITLE ***

                JLabel Title=new JLabel("Airline Reservation System");

                Title.setFont(new Font("Elephant",Font.BOLD,38));

                Title.setBounds(140,30,600,40);

                f.add(Title);

        //**SUB- TITLE ***

                JLabel SubTitle=new JLabel("Page 1: Personal Detail");

                SubTitle.setFont(new Font("Arial",Font.BOLD,26));

                SubTitle.setBounds(280,80,600,40);

                f.add(SubTitle);

        //**Name ***

                JLabel Name=new JLabel("Username : ");

                Name.setFont(new Font("Raleway",Font.BOLD,22));

                Name.setBounds(70,150,200,30);

                f.add(Name);

                JTextField NameTextField=new JTextField();

                NameTextField.setFont(new Font("Raleway",Font.PLAIN,14));

                NameTextField.setBounds(280,150,400,30);

                f.add(NameTextField);
```

```
//**Phone Number ***
            JLabel Phone=new JLabel("Phone Number : ");
            Phone.setFont(new Font("Raleway",Font.BOLD,22));
            Phone.setBounds(70,200,200,30);
            f.add(Phone);
            JTextField PhoneTextField=new JTextField();
            PhoneTextField.setFont(new Font("Raleway",Font.PLAIN,14));
            PhoneTextField.setBounds(280,200,400,30);
            f.add(PhoneTextField);


//**Date Of Birth ***
            JLabel Dob=new JLabel("Date Of Birth : ");
            Dob.setFont(new Font("Raleway",Font.BOLD,22));
            Dob.setBounds(70,250,200,30);
            f.add(Dob);
            date = new JComboBox(dates);
            date.setFont(new Font("Arial", Font.PLAIN, 15));
            date.setSize(50, 20);
             date.setBounds(280,250,200,30);
             f.add(date);
            month = new JComboBox(months);
             month.setFont(new Font("Arial", Font.PLAIN, 15));
             month.setSize(60, 20);
            month.setBounds(480,250,200,30);
            f.add(month);
             year = new JComboBox(years);
            year.setFont(new Font("Arial", Font.PLAIN, 15));
            year.setSize(60, 20);
            year.setBounds(680,250,200,30);
            f.add(year);
//**Gender ***
            JLabel Gender=new JLabel("Gender : ");
            Gender.setFont(new Font("Raleway",Font.BOLD,22));
            Gender.setBounds(70,300,200,30);
```

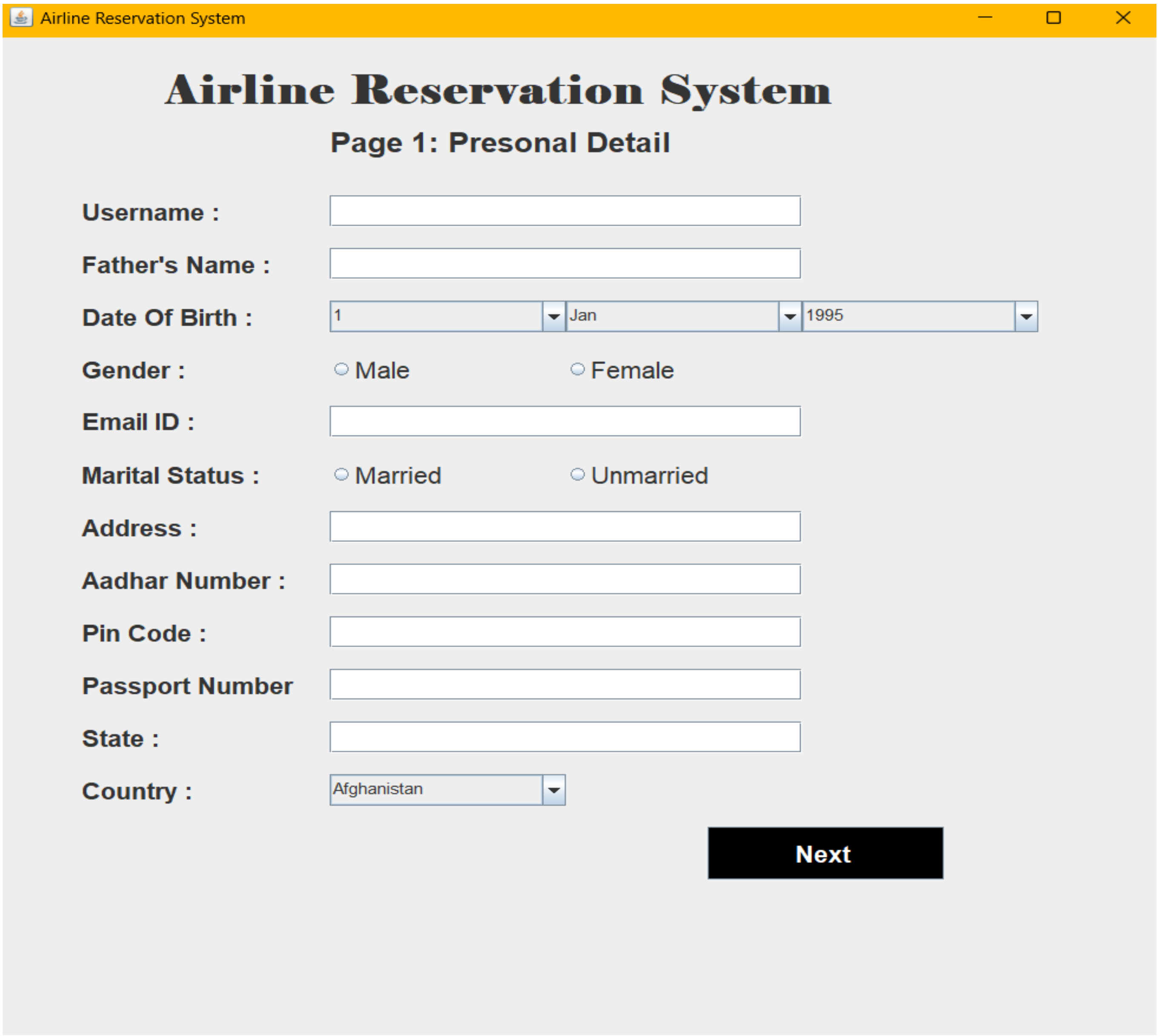
```
        f.add(Gender);
        JRadioButton male=new JRadioButton("Male");
        male.setBounds(280,300,100,30);
        male.setFont(new Font("Raleway",Font.PLAIN,22));
        f.add(male);
        JRadioButton female=new JRadioButton("Female");
        female.setBounds(480,300,150,30);
        female.setFont(new Font("Raleway",Font.PLAIN,22));
        f.add(female);
        ButtonGroup genderGroup= new ButtonGroup();
        genderGroup.add(male);
        genderGroup.add(female);
    //**Email ID ***
        JLabel Email=new JLabel("Email ID : ");
        Email.setFont(new Font("Arial",Font.BOLD,22));
        Email.setBounds(70,350,200,30);
        f.add(Email);
        JTextField EmailTextField=new JTextField();
        EmailTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        EmailTextField.setBounds(280,350,400,30);
        f.add(EmailTextField);
    //**Marital Status ***
        JLabel Marital=new JLabel("Marital Status : ");
        Marital.setFont(new Font("Raleway",Font.BOLD,22));
        Marital.setBounds(70,400,200,30);
        f.add(Marital);
        JRadioButton married=new JRadioButton("Married");
        married.setBounds(280,400,100,30);
        married.setFont(new Font("Raleway",Font.PLAIN,22));
        f.add(married);
        JRadioButton unmarried=new JRadioButton("Unmarried");
        unmarried.setBounds(480,400,150,30);
        unmarried.setFont(new Font("Raleway",Font.PLAIN,22));
        f.add(unmarried);
        ButtonGroup MaritalGroup= new ButtonGroup();
        MaritalGroup.add(married);
        MaritalGroup.add(unmarried);
```

```
//**Address ***
        JLabel Address=new JLabel("Address : ");
        Address.setFont(new Font("Raleway",Font.BOLD,22));
        Address.setBounds(70,450,200,30);
        f.add(Address);
        JTextField AddressTextField=new JTextField();
        AddressTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        AddressTextField.setBounds(280,450,400,30);
        f.add(AddressTextField);
//**Aadhar ***
        JLabel Aadhar=new JLabel("Aadhar Number : ");
        Aadhar.setFont(new Font("Raleway",Font.BOLD,22));
        Aadhar.setBounds(70,500,200,30);
        f.add(Aadhar);
        JTextField AadharTextField=new JTextField();
        AadharTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        AadharTextField.setBounds(280,500,400,30);
        f.add(AadharTextField);
//**Pin Code ***
        JLabel Pin=new JLabel("Pin Code : ");
        Pin.setFont(new Font("Raleway",Font.BOLD,22));
        Pin.setBounds(70,550,200,30);
        f.add(Pin);
        JTextField PinTextField=new JTextField();
        PinTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        PinTextField.setBounds(280,550,400,30);
        f.add(PinTextField);
//**Passport No ***
        JLabel Pass=new JLabel("Passport Number ");
        Pass.setFont(new Font("Raleway",Font.BOLD,22));
        Pass.setBounds(70,600,200,30);
        f.add(Pass);
        JTextField PassTextField=new JTextField();
        PassTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        PassTextField.setBounds(280,600,400,30);
        f.add(PassTextField);
```

```
//**State ***
          JLabel State=new JLabel("State : ");
          State.setFont(new Font("Raleway",Font.BOLD,22));
          State.setBounds(70,650,200,30);
          f.add(State);
          JTextField StateTextField=new JTextField();
          StateTextField.setFont(new Font("Raleway",Font.PLAIN,14));
          StateTextField.setBounds(280,650,400,30);
          f.add(StateTextField);
//**Country ***
          JLabel Country=new JLabel("Country : ");
          Country.setFont(new Font("Raleway",Font.BOLD,22));
          Country.setBounds(70,700,200,30);
          f.add(Country);
          lt = new JComboBox(country);
          lt.setFont(new Font("Arial", Font.PLAIN, 15));
          lt.setSize(60, 20);
          lt.setBounds(280,700,200,30);
          f.add(lt);
//**Next Button ***
          JButton next=new JButton("Next ");
          next.setBackground(Color.BLACK);
          next.setForeground(Color.WHITE);
          next.setFont(new Font("Raleway",Font.BOLD,22));
          next.setBounds(600,750,200,50);
          next.addActionListener(this);
          f.add(next);
          f.setBounds(100,140,80,20);
          f.setLayout(null);
          f.setSize(1000,1000);
          f.setVisible(true);
          f.setLocation(350, 10);
          f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

```
        public void actionPerformed(ActionEvent e)
         {
               Prog secondFrame=new  Prog();
               f.dispose();
         }
}
class page2
{
  public static void main(String[] args) {
        new Prog();
  }
}
```

OUTPUT



Airline Reservation System

Airline Reservation System

Page 1: Presonal Detail

Username :

Father's Name :

Date Of Birth : 1 Jan 1995

Gender : ○ Male ○ Female

Email ID :

Marital Status : ○ Married ○ Unmarried

Address :

Aadhar Number :

Pin Code :

Passport Number

State :

Country : Afghanistan

PAGE 3: ADDITIONAL DETAIL

```
class Prog implements ActionListener  {
        JFrame f;
        private String dates[] = { "1", "2", "3", "4", "5","6", "7", "8", "9", "10", "11", "12", "13", "14", "15","16",
                        "17", "18", "19", "20","21", "22", "23", "24", "25","26", "27", "28", "29", "30","31" };
        private String months[]= { "Jan", "feb", "Mar", "Apr","May", "Jun", "July", "Aug","Sep", "Oct", "Nov", "Dec" };
        private String years[]= { "1995", "1996", "1997", "1998","1999", "2000", "2001", "2002", "2003", "2004", "2005",
                                    "2006","2007", "2008", "2009", "2010","2011", "2012", "2013", "2014","2015", "2016",
                                    "2017", "2018","2019" ,"2020", "2021", "2022","2023","2024"};
        private JComboBox date,months,year,lt;
        private String country[]={"Afghanistan","Bangladesh","Canada","China","India","Japan","Malaysia",

                                    "Maldives","Pakistan","United Kingdom","United Arab Emirates","United States"};


                Prog()
                {
                  f=new JFrame("Airline Reservation System");
        //**TITLE ***
                JLabel Title=new JLabel("Airline Reservation System");
                Title.setFont(new Font("Elephant",Font.BOLD,38));
                Title.setBounds(140,30,600,40);
                f.add(Title);
        //**SUB- TITLE ***
                JLabel SubTitle=new JLabel("Page 2: Additional Details");
                SubTitle.setFont(new Font("Arial",Font.BOLD,26));
                SubTitle.setBounds(280,80,600,40);
                f.add(SubTitle);
        //** FlightName ***
                JLabel FlightName=new JLabel("Flight Name : ");
                FlightName.setFont(new Font("Raleway",Font.BOLD,22));
                FlightName.setBounds(70,150,200,30);
                f.add(FlightName);


                JTextField FlightNameTextField=new JTextField();
                FlightNameTextField.setFont(new Font("Raleway",Font.PLAIN,14));
                FlightNameTextField.setBounds(280,150,400,30);
                f.add(FlightNameTextField);
```

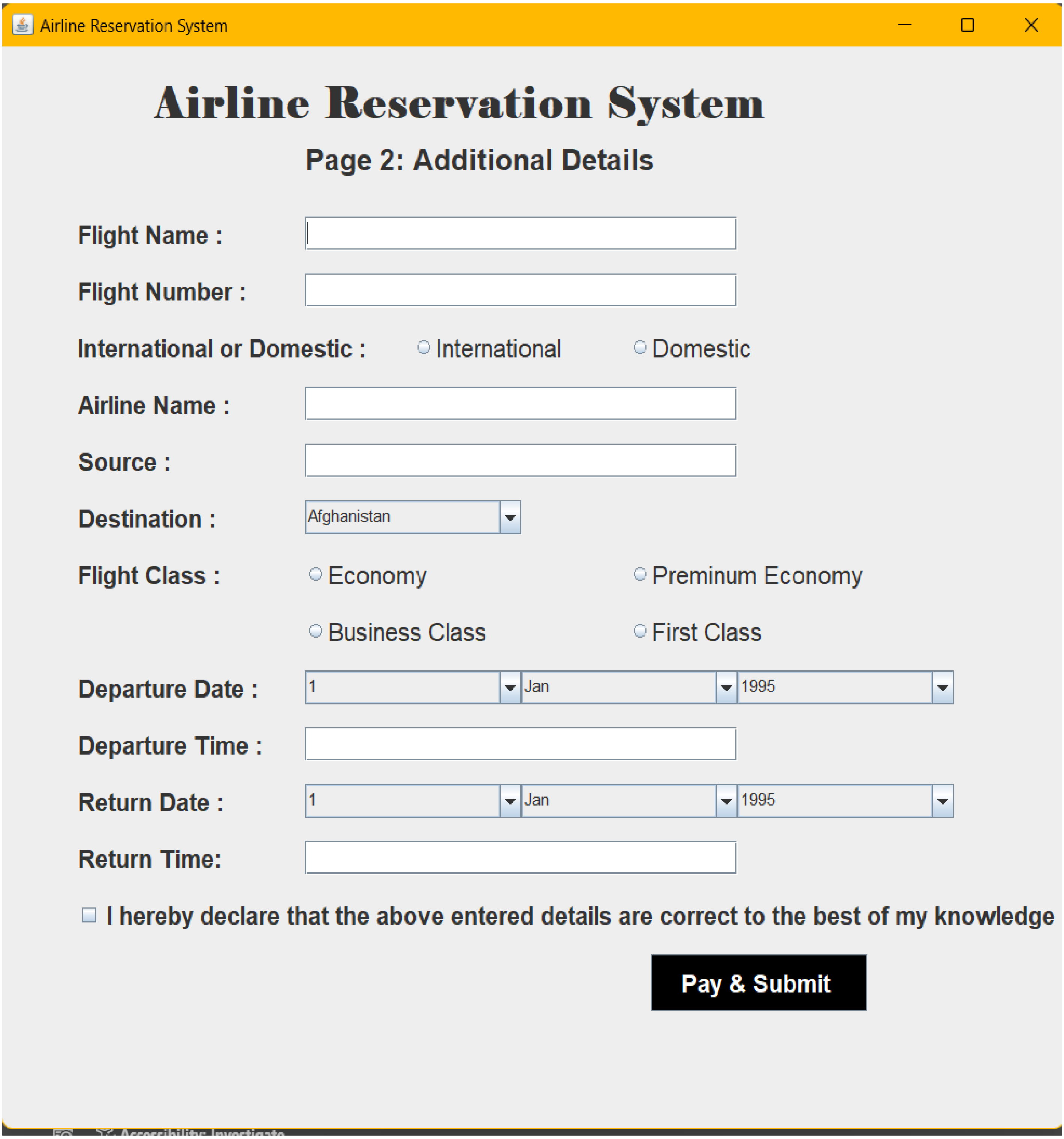
```
//** Flight Number ***
        JLabel FlightNo=new JLabel("Flight Number : ");
        FlightNo.setFont(new Font("Raleway",Font.BOLD,22));
        FlightNo.setBounds(70,200,200,30);
        f.add(FlightNo);
        JTextField FlightNoTextField=new JTextField();
        FlightNoTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        FlightNoTextField.setBounds(280,200,400,30);
        f.add(FlightNoTextField);
//**International or Domestic ***
        JLabel IntDom=new JLabel("International or Domestic : ");
        IntDom.setFont(new Font("Raleway",Font.BOLD,22));
        IntDom.setBounds(70,250,350,30);
        f.add(IntDom);
        JRadioButton International=new JRadioButton("International");
        International.setBounds(380,250,200,30);
        International.setFont(new Font("Raleway",Font.PLAIN,22));
        f.add(International);
        JRadioButton Domestic=new JRadioButton("Domestic");
        Domestic.setBounds(580,250,150,30);
        Domestic.setFont(new Font("Raleway",Font.PLAIN,22));
        f.add(Domestic);
        ButtonGroup IntDomGroup= new ButtonGroup();
        IntDomGroup.add(International);
        IntDomGroup.add(Domestic);
//**Airline ***
        JLabel Airline=new JLabel("Airline Name : ");
        Airline.setFont(new Font("Raleway",Font.BOLD,22));
        Airline.setBounds(70,300,200,30);
        f.add(Airline);
        JTextField AirlineTextField=new JTextField();
        AirlineTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        AirlineTextField.setBounds(280,300,400,30);
        f.add(AirlineTextField);
```

```
    //**Source ***
            JLabel Source=new JLabel("Source : ");
            Source.setFont(new Font("Raleway",Font.BOLD,22));
            Source.setBounds(70,350,200,30);
            f.add(Source);
            JTextField SourceTextField=new JTextField();
            SourceTextField.setFont(new Font("Raleway",Font.PLAIN,14));
            SourceTextField.setBounds(280,350,400,30);
            f.add(SourceTextField);
            lt = new JComboBox(country);
            lt.setFont(new Font("Arial", Font.PLAIN, 15));
            lt.setSize(60, 20);
            lt.setBounds(280,350,200,30);
            f.add(lt);
    //**Destination ***
            JLabel des=new JLabel("Destination : ");
            des.setFont(new Font("Raleway",Font.BOLD,22));
            des.setBounds(70,400,200,30);
            f.add(des);
            lt = new JComboBox(country);
            lt.setFont(new Font("Arial", Font.PLAIN, 15));
            lt.setSize(60, 20);
            lt.setBounds(280,400,200,30);
            f.add(lt);
    //**Class ***
            JLabel Class=new JLabel("Flight Class : ");
            Class.setFont(new Font("Raleway",Font.BOLD,22));
            Class.setBounds(70,450,200,30);
            f.add(Class);
            JRadioButton Economy=new JRadioButton("Economy");
            Economy.setBounds(280,450,250,30);
            Economy.setFont(new Font("Raleway",Font.PLAIN,22));
            f.add(Economy);
            JRadioButton PEconomy=new JRadioButton("Preminum Economy");
            PEconomy.setBounds(580,450,250,30);
            PEconomy.setFont(new Font("Raleway",Font.PLAIN,22));
            f.add(PEconomy);
```

```
        JRadioButton Business=new JRadioButton("Business Class");
        Business.setBounds(280,500,250,30);
        Business.setFont(new Font("Raleway",Font.PLAIN,22));
        f.add(Business);
        JRadioButton First=new JRadioButton("First Class");
        First.setBounds(580,500,250,30);
        First.setFont(new Font("Raleway",Font.PLAIN,22));
        f.add(First);
        ButtonGroup ClassGroup= new ButtonGroup();
        ClassGroup.add(Economy);
        ClassGroup.add(PEconomy);
        ClassGroup.add(Business);
        ClassGroup.add(First);
//**Departure Date ***
        JLabel DepartureD=new JLabel("Departure Date : ");
        DepartureD.setFont(new Font("Raleway",Font.BOLD,22));
        DepartureD.setBounds(70,550,200,30);
        f.add(DepartureD);

         date = new JComboBox(dates);
        date.setFont(new Font("Arial", Font.PLAIN, 15));
        date.setSize(50, 20);
        date.setBounds(280,550,200,30);
         f.add(date);
        month = new JComboBox(months);
        month.setFont(new Font("Arial", Font.PLAIN, 15));
        month.setSize(60, 20);
        month.setBounds(480,550,200,30);
         f.add(month);
        year = new JComboBox(years);
        year.setFont(new Font("Arial", Font.PLAIN, 15));
        year.setSize(60, 20);
        year.setBounds(680,550,200,30);
        f.add(year);
```

```
//**Departure Time ***
        JLabel DepartureT=new JLabel("Departure Time : ");
        DepartureT.setFont(new Font("Raleway",Font.BOLD,22));
        DepartureT.setBounds(70,600,200,30);
        f.add(DepartureT);
        JTextField DepartureTTextField=new JTextField();
        DepartureTTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        DepartureTTextField.setBounds(280,600,400,30);
        f.add(DepartureTTextField);
//**Return Date ***
        JLabel ReturnD=new JLabel("Return Date : ");
        ReturnD.setFont(new Font("Raleway",Font.BOLD,22));
        ReturnD.setBounds(70,650,200,30);
        f.add(ReturnD);
         date = new JComboBox(dates);
        date.setFont(new Font("Arial", Font.PLAIN, 15));
        date.setSize(50, 20);
        date.setBounds(280,550,200,30);
         f.add(date);
        month = new JComboBox(months);
        month.setFont(new Font("Arial", Font.PLAIN, 15));
        month.setSize(60, 20);
        month.setBounds(480,550,200,30);
         f.add(month);
        year = new JComboBox(years);
        year.setFont(new Font("Arial", Font.PLAIN, 15));
        year.setBounds(680,550,200,30);
        f.add(year);
//**Return Time ***
        JLabel ReturnT=new JLabel("Return Time: ");
        ReturnT.setFont(new Font("Raleway",Font.BOLD,22));
        ReturnT.setBounds(70,700,200,30);
        f.add(ReturnT);
        JTextField ReturnTimTextField=new JTextField();
        ReturnTimTextField.setFont(new Font("Raleway",Font.PLAIN,14));
        ReturnTimTextField.setBounds(280,700,400,30);
        f.add(ReturnTimTextField);
        //**Submit ***
                JButton Submit =new JButton("Pay & Submit ");
                Submit.setBackground(Color.BLACK);
                Submit.setForeground(Color.WHITE);
                Submit.setFont(new Font("Raleway",Font.BOLD,22));
                Submit.setBounds(600,800,200,50);
                Submit.addActionListener(this);
                f.add(Submit);
                f.setBounds(100,140,80,20);
                f.setLayout(null);
                f.setSize(1000,1000);
                f.setVisible(true);
                f.setLocation(350, 10);
                f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
        public void actionPerformed(ActionEvent e)
                {
                        System.out.println("Boking Successfully…….");
                }
}
class Program
{

   public static void main(String[] args) {
           new Prog();
   }
}
```

OUTPUT



Airline Reservation System

Page 2: Additional Details

Flight Name :

Flight Number :

International or Domestic : ○ International ○ Domestic

Airline Name :

Source :

Destination : Afghanistan

Flight Class : ○ Economy ○ Preminum Economy

○ Business Class ○ First Class

Departure Date : 1 Jan 1995

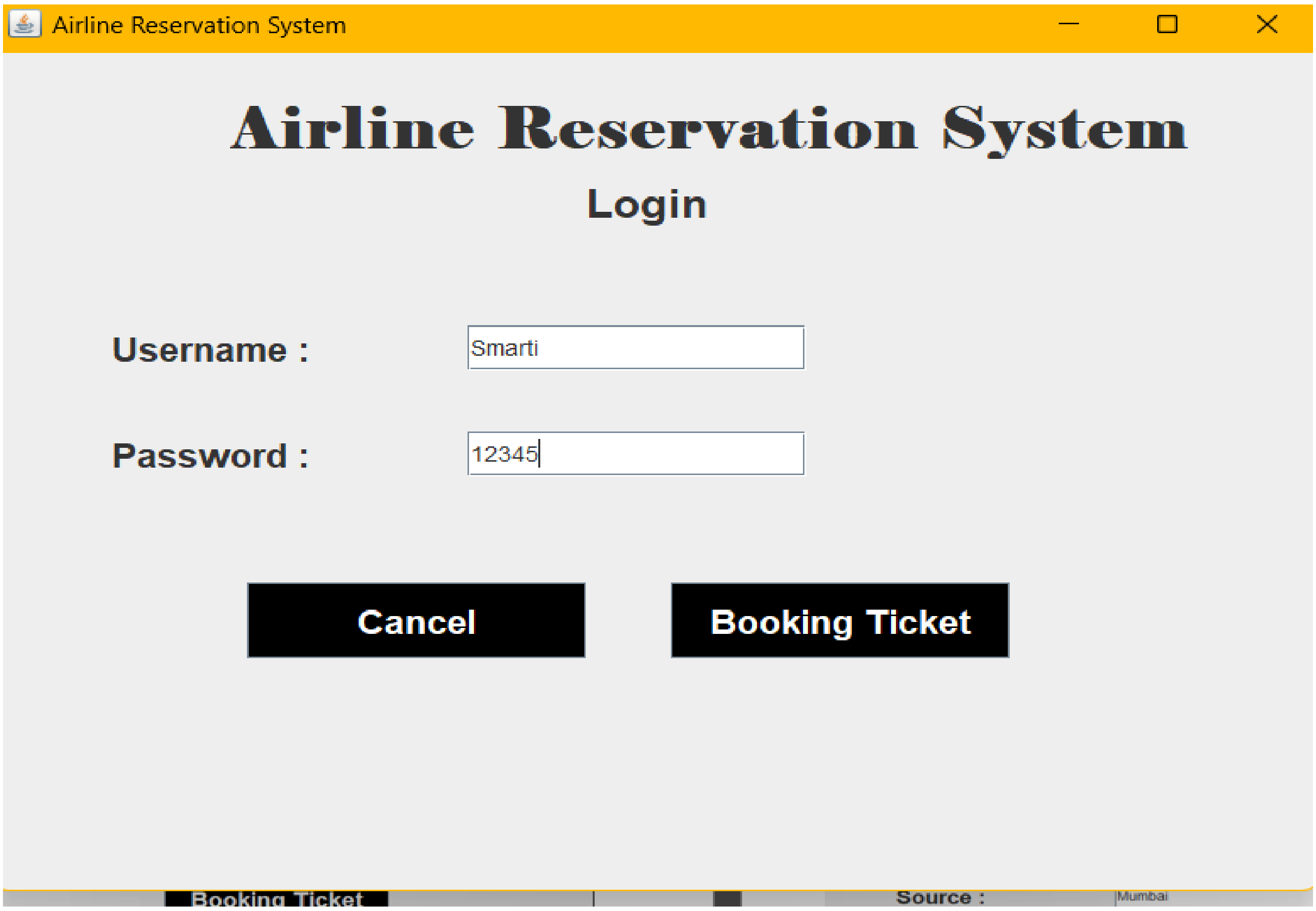
Departure Time :

Return Date : 1 Jan 1995

Return Time:

□ I hereby declare that the above entered details are correct to the best of my knowledge

INPUT SCREEN



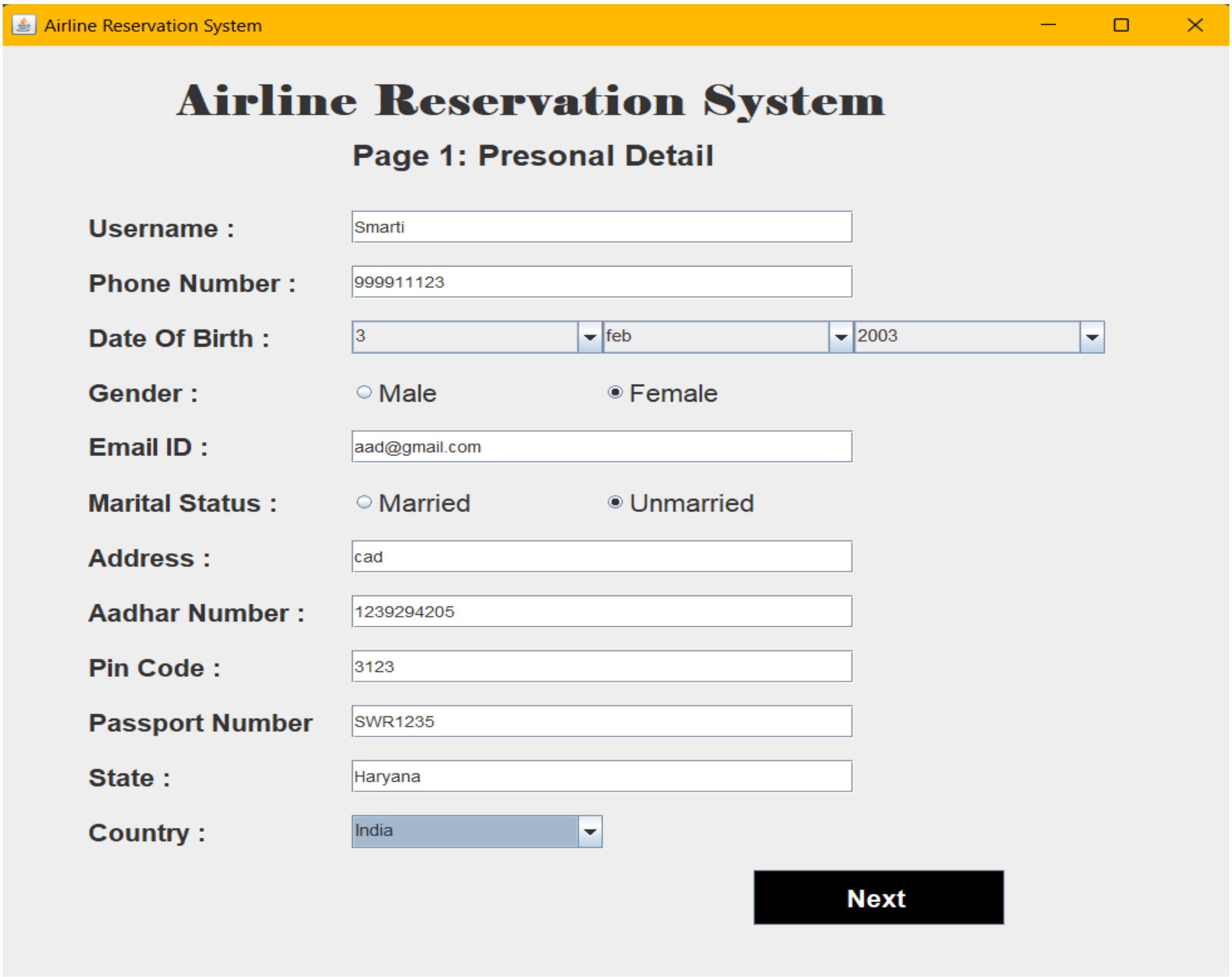
Airline Reservation System

Airline Reservation System

Login

Username : Smarti

Password : 12345



Airline Reservation System

Airline Reservation System

Page 1: Presonal Detail

Username : Smarti

Phone Number : 999911123

Date Of Birth : 3 feb 2003

Gender : ○ Male ● Female

Email ID : aad@gmail.com

Marital Status : ○ Married ● Unmarried

Address : cad

Aadhar Number : 1239294205

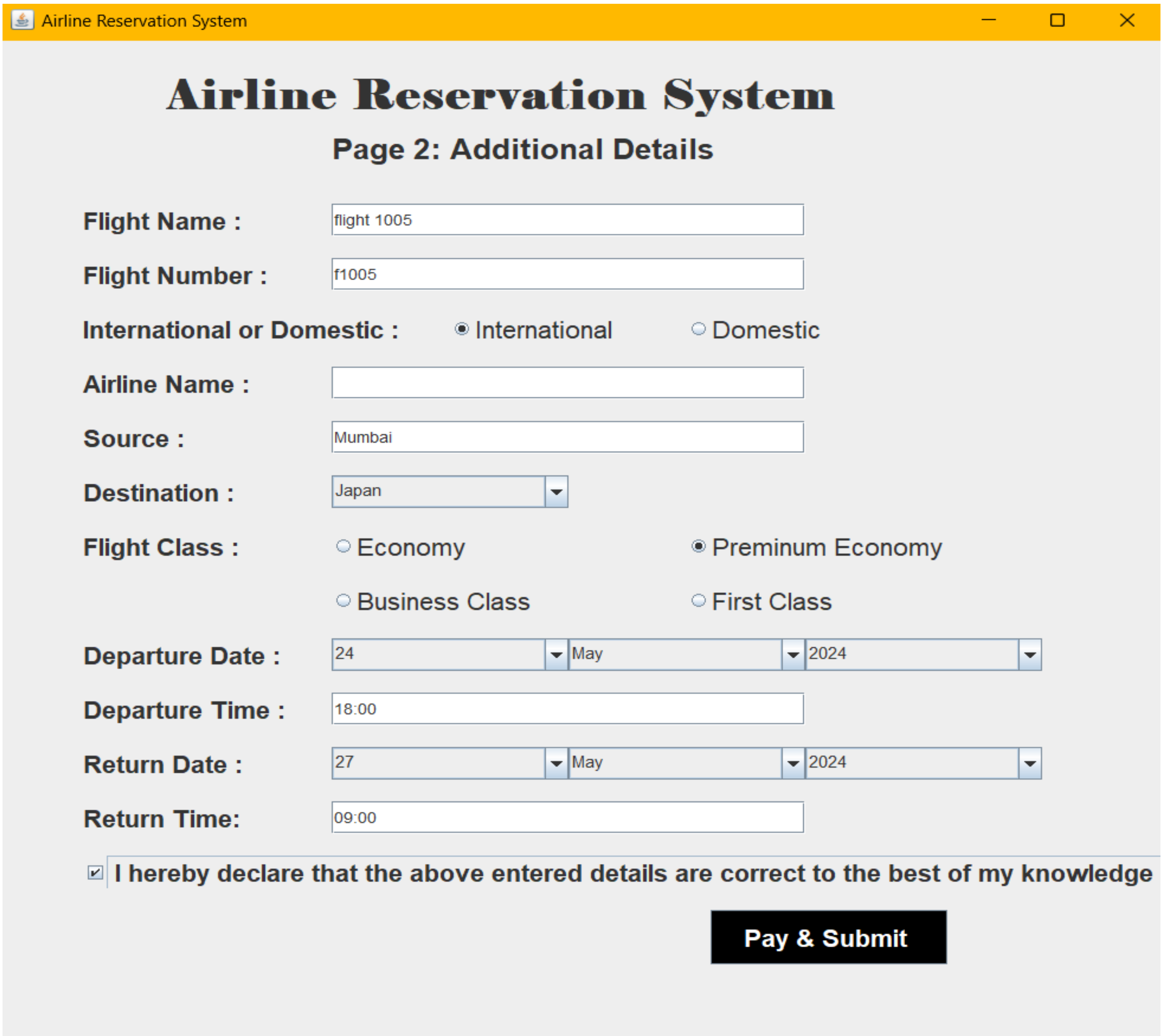
Pin Code : 3123

Passport Number SWR1235

State : Haryana

Country : India

INPUT SCREEN



Airline Reservation System

Page 2: Additional Details

Flight Name : flight 1005

Flight Number : f1005

International or Domestic : ● International ○ Domestic

Airline Name :

Source : Mumbai

Destination : Japan

Flight Class : ○ Economy ● Preminum Economy ○ Business Class ○ First Class

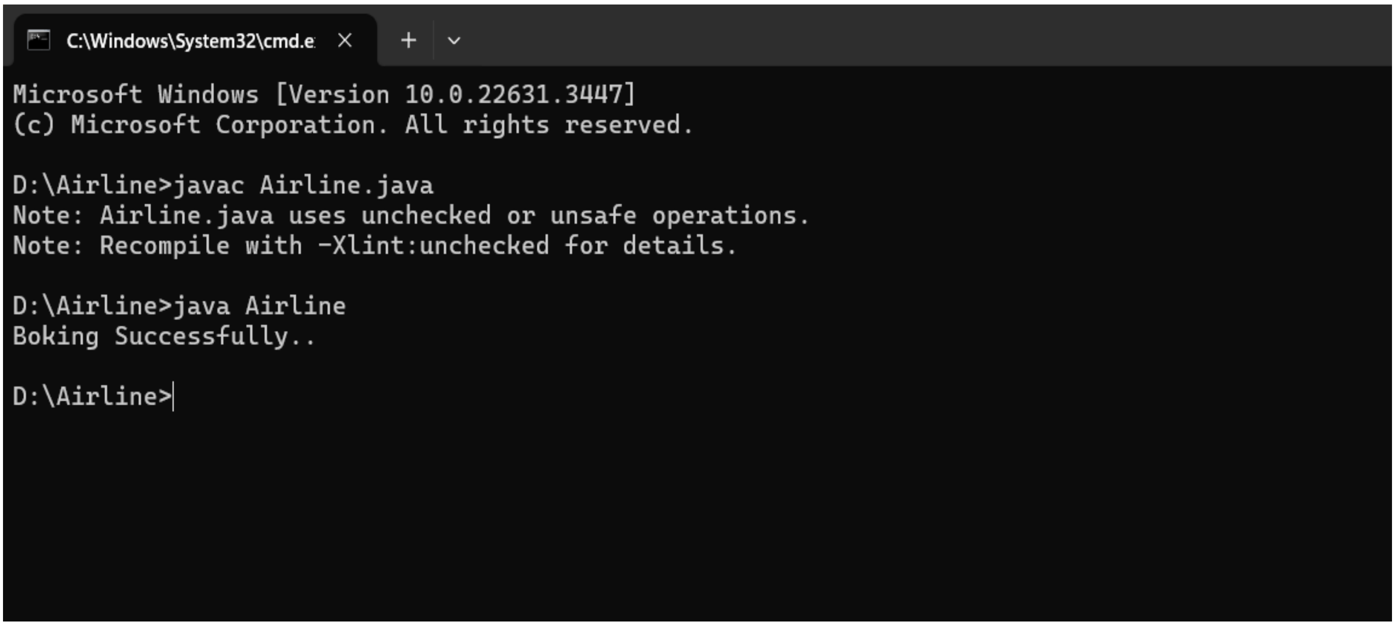
Departure Date : 24 May 2024

Departure Time : 18:00

Return Date : 27 May 2024

Return Time: 09:00

☑ I hereby declare that the above entered details are correct to the best of my knowledge



```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

D:\Airline>javac Airline.java
Note: Airline.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\Airline>java Airline
Boking Successfully..

D:\Airline>
```

CONCLUSION

In conclusion, the development of the Airline Reservation System has been a valuable learning experience in software development. These specifications are developed as part of a Java mini-project and the functionalities described will be developed in depending on the progress of the project. Throughout the project, we have successfully implemented various features to create a functional and user-friendly application for booking flights.

Throughout the development process, we encountered and overcame various challenges, such as handling user input validation, managing state transitions between different screens, and implementing error handling mechanisms. These challenges provided valuable learning opportunities and improved our problem-solving skills.

In conclusion, this mini-project has provided us with a practical understanding of Java Swing development and software engineering principles. It has equipped us with the knowledge and skills necessary to develop functional desktop applications and lays the foundation for further exploration into more complex software projects in the future.