



# **RAJALAKSHMI ENGINEERING COLLEGE**

**An AUTONOMOUS Institution  
Affiliated to ANNA UNIVERSITY, Chennai**

## **AIRLINE MANAGEMENT SYSTEM**

**CS23333 – Introduction to OOPS and JAVA**

**Mini Project Report**

Submitted by

LAVANYA P (231001099)

MADHUMITHA M (231001104)

Of

**BACHELOR OF TECHNOLOGY**

In

**INFORMATION TECHNOLOGY**

**RAJALAKSHMI ENGINEERING COLLEGE, THANDALAM**

**(An Autonomous Institution)**



**RAJALAKSHMI ENGINEERING COLLEGE**  
**BONAFIDE CERTIFICATE**

Certified that this project titled “Airline Management System” is the bonafied work of “LAVANYA P (231001099), MADHUMITHA M (231001104)” who carried out the project work under my supervision.

**SIGNATURE**

**Dr.Valarmathie**

**HEAD OF THE DEPARTMENT**

Information Technology

Rajalakshmi Engineering College,

Rajalakshmi Nagar,Thandalam.

Chennai – 602105

**SIGNATURE**

**Mrs. Sangeetha Tupili**

**COURSE INCHARGE**

Information Technology

Rajalakshmi Engineering College,

Rajalakshmi Nagar, Thandalam.

Chennai – 602105

22/11/2024

This project is submitted for CS23333 – Introduction to Oops and Java held on 22.11.2024.

**INTERNAL EXAMINAR**

**EXTERNAL EXAMINAR**

# ABSTRACT

## ABSTRACT

This project on Airline Management System is the automation of registration process of airlines system. The system provides information like passenger's information, flight information, list of all passengers, it allows storing and retrieving data related to the airline industry and make transactions related to air travel etc. The system also allows us to add records when a passenger reserves a ticket. For data storage and retrieval we use MySQL Database. It enables us to add any number of records in our database. The project "Airline Management System" comprises of a large number of flights which belong to a particular airline. The system we have implemented manages different objects viz.

- . Airline
- . Airline Employees
- . Customers/Traveller

**Each of these accesses a database schema which has corresponding tables.**

Language Used-Java Core

Concept Used-Swing

IDE Used-NetBeans

Database Used-MySQL

# TABLE OF CONTENTS

<b>Abstract-----</b>	<b>3</b>
<b>1. AIRLINE MANAGEMENT SYSTEM</b>	
1.2. Introduction-----	4
1.3. Purpose-----	5
1.4. Scope of Project-----	5
1.5. Software Requirement Specification-----	6
<b>2. System Flow Diagrams-----</b>	<b>13</b>
2.1. Use Case Diagram-----	13
2.2. Entity-relationship Diagrams-----	13
2.3. Data Flow Diagram -----	13
<b>3. Module Description-----</b>	<b>14</b>
3.1. Flight details-----	14
3.2. Add Customer-----	14
3.3. Book flight-----	14
3.4. Journey Details -----	14
3.5. Boarding pass-----	14
3.6. Cancel ticket-----	15
<b>4. Implementation-----</b>	<b>15</b>
4.1. Code-----	15
4.2. Output Screens-----	53
<b>5. Conclusion-----</b>	<b>57</b>
<b>6. References-----</b>	<b>57</b>

## 1.1 INTRODUCTION:

**Airline Management System** is a Software which is helpful for the Authorities. In the current system all the activities are done manually. It is very time consuming and costly. Our Employee Management System deals with the various activities related to the Employees.

By using this software, we can get Name and Salary details of the Employees in a single window page

## 1.3 PURPOSE:

The objective of the **Airline Management System** is to enable the administration of an airline company to efficiently manage and edit information related to flights, passengers, and staff. It provides functionality to track schedules, bookings, and maintenance records, while also allowing employees to update their profiles and passengers to manage their bookings. Overall, it simplifies airline operations, enhances customer service, and ensures effective management of airline data, making it easier for the administration, employees, and passengers to interact with the system seamlessly.

## 1.4 SCOPE OF THE PROJECT:

The proposed system will help the airline administrator manage operations easily and efficiently. It will provide quick responses to queries from users like administrators, staff, and passengers, and fulfill all required tasks as planned.

This project is built using Java (JDBC) and MySQL to manage data such as flight schedules, passenger information, employee details, and maintenance records. It simplifies the management of airline operations, making it faster and more organized.

## **1.5 Software requirement Specification:**

### **Introduction:**

The Airline Management System is designed to handle all the details related to airline operations. These details include flight schedules, passenger information, ticket bookings, staff profiles, and maintenance records. The Airline Management System is an automated version of the manual system, simplifying and streamlining the management of airline operations to ensure accuracy and efficiency.

### **Document Purpose:**

This SRS document contains the complete software requirements for the Airline Management System and describes the design decisions, architectural design, and detailed design needed to implement the system. It provides clear visibility into the system's design and includes all the necessary information required for software development, implementation, and ongoing support.

### **Product Scope:**

The Airline Management System is developed for general use and aims to replace outdated manual or paper-based systems. The system is designed to efficiently manage and provide flight, passenger, and staff information to the airline administration. It includes a flexible mechanism to update and edit records, such as flight schedules, passenger bookings, and staff details, ensuring streamlined and efficient airline operations.

### **Definitions, Acronyms and Abbreviations:**

- **AMS** - Airline Management System
- **SRS** - Software Requirements Specification

## References and Acknowledgement:

[1]

<https://developers.openshift.com/database/mysql.html>

[2]Web <https://youtu.be/UbIIFLsEeiM>

## Overall Description:

The Airline Management System allows authorized users to manage flight schedules, passenger bookings, and staff details. It is designed to simplify airline operations, making it easier for airlines to handle day-to-day tasks like booking, scheduling, and customer service. The system helps improve efficiency and organization for airline companies.

## Product Perspective:

The proposed Airline Management System will be developed using a client/server architecture and will be compatible with the Microsoft Windows operating system. The front end of the system will be developed using Java AWT and SWING, while the backend will be powered by MySQL for efficient data management and storage. This architecture ensures a reliable, user-friendly interface and a robust database to handle the airline's operations seamlessly.

## Product Functionality:

### a) Flight Details:

Manage flight information like flight number, destination, and schedule.

### b) Add Customer Details:

Store new customer information such as name and contact details.

### c) Book Flight:

Allow customers to book a flight based on available options.

#### **d) Journey Details:**

Let customers view their flight details and status.

#### **e) Cancel Ticket:**

Allow customers to cancel their booked tickets.

## **User and Characteristics:**

**Qualification:** At least a high school diploma and comfortable with English.

**Experience:** Should be familiar with airline operations, flight schedules, and booking processes.

**Technical Experience:** Basic knowledge of computer systems and software applications.

## **Operating Environment:**

### **Hardware Requirements:**

- Processor: Any processor over i3
- Operating System: Windows 8, 10, or 11
- Processor Speed: 2.0 GHz or higher
- RAM: 4GB or more
- Hard Disk: 500GB

### **Software Requirements:**

- Database: MySQL
- Frontend: Java (SWING, AWT)
- Technology: Java (JDBC)

## **Constraints:**

- The system will only be accessible by the administrator.
- The delete operation is restricted to the administrator only.



- To simplify the system, there will be no confirmation check before performing the delete operation.
- The administrator must exercise caution before deleting any records, as they will be responsible for maintaining data consistency.

## Specific Requirements:

### User Interface:

The Airline Management System will have a simple interface with the following features:

#### a) **Flight Details:**

Admin can view and manage flight information like flight number, destination, and schedule.

#### b) **Add Customer Details:**

Admin can add new customer details such as name and contact info.

#### c) **Book Flight:**

Customers can book flights based on availability.

#### d) **Journey Details:**

Customers can view their flight details and status.

#### e) **Cancel Ticket:**

Customers can cancel their tickets easily.

## Hardware Interface:

- Screen resolution of at least 640 x 480 or above.
- **Any version of Windows 8,10,11.**

### Hardware Interface:

- **Screen Resolution:** Minimum of 640 x 480 or higher.
- **Operating System:** Any version of Windows 8, 10, or 11.

- **Processor:** Any processor with a speed of 2.0 GHz or higher.
- **RAM:** At least 4GB of RAM.
- **Hard Disk:** Minimum of 500GB of available storage.

## Functional Requirements:

### Log in Module (LM):

- The **admin** can access the Login Module to log into the system.
- The module will have **username** and **password** fields.
- The **password** will be hidden as symbols when typed.
- There will be a **Login button** to submit the details.
- When the **Login button** is clicked, the system will check the credentials against the database.
- If the credentials are correct, the admin can use the system; otherwise, an error message will be shown.

### Registered Users Module (RUM):

- After a successful login, the **admin** can navigate through the application to view detailed **Employee information**.
- The **admin** can update and maintain **Employee profiles**, including personal details and salary information.
- The system will allow the admin to make changes to **salary** and **personal details** of employees as needed.

### Administrator Module (AM):

After logging in, the administrator will have access to functions for managing employee data. The administrator can add new employee details, update existing records, or delete unnecessary entries. By clicking the **Add** button, the administrator can enter new employee information, and with the **Update** button, they can modify existing employee details. When any changes are made, the **Administrator Module** will send the request to the **Server Module**, which will update the database accordingly.

### Server Module (SM):

The **Server Module (SM)** acts as an intermediary between the various modules and the database. It receives requests from other modules, processes them, and formats the pages to be displayed to the user. The Server Module is responsible for validating all incoming requests and ensuring they are executed properly. It handles data interactions, ensuring that updates, deletions, and other changes are made to the database correctly.

## **Non-functional Requirements:**

The system should process 95% of tasks, like booking or updating flights, in under 2 seconds. It must be available 99.9% of the time, with no more than 1 hour of downtime per month. The system should support at least 100 users at once. All sensitive data must be encrypted, and users must have strong passwords.

The interface should be easy to use, with tasks completed in under 5 minutes. The system should allow easy updates and have logs for errors. Daily backups should be made, and recovery should take less than 30 minutes if needed. It should work on Windows 8, 10, and 11, and support browsers like Chrome, Firefox, and Edge. These requirements ensure the system is fast, secure, and reliable.

## **Performance:**

The system and server must be capable of handling real-time errors triggered by user actions. Any failures or issues reported by the server must be addressed immediately to ensure the system remains functional and safe for both users and the system. The system must be robust, capable of managing errors efficiently to maintain smooth operation and avoid downtime or disruptions in service. Safety and reliability should be prioritized to protect user data and system integrity.

## **Reliability:**

The system is safety-critical, meaning it must always operate within normal parameters. If the system deviates from normal operation, the server must be shut down immediately for troubleshooting and fixed as quickly as possible. This emergency shutdown should only occur when absolutely necessary and should not happen without a valid reason. The goal is to ensure that the system is stable and reliable, with minimal disruption to operations.

## **Availability:**

The system should respond to user requests within 1 second under normal conditions. It must also provide immediate feedback on what the system is doing, so the user knows their action was received and is being processed.

## **Security:**

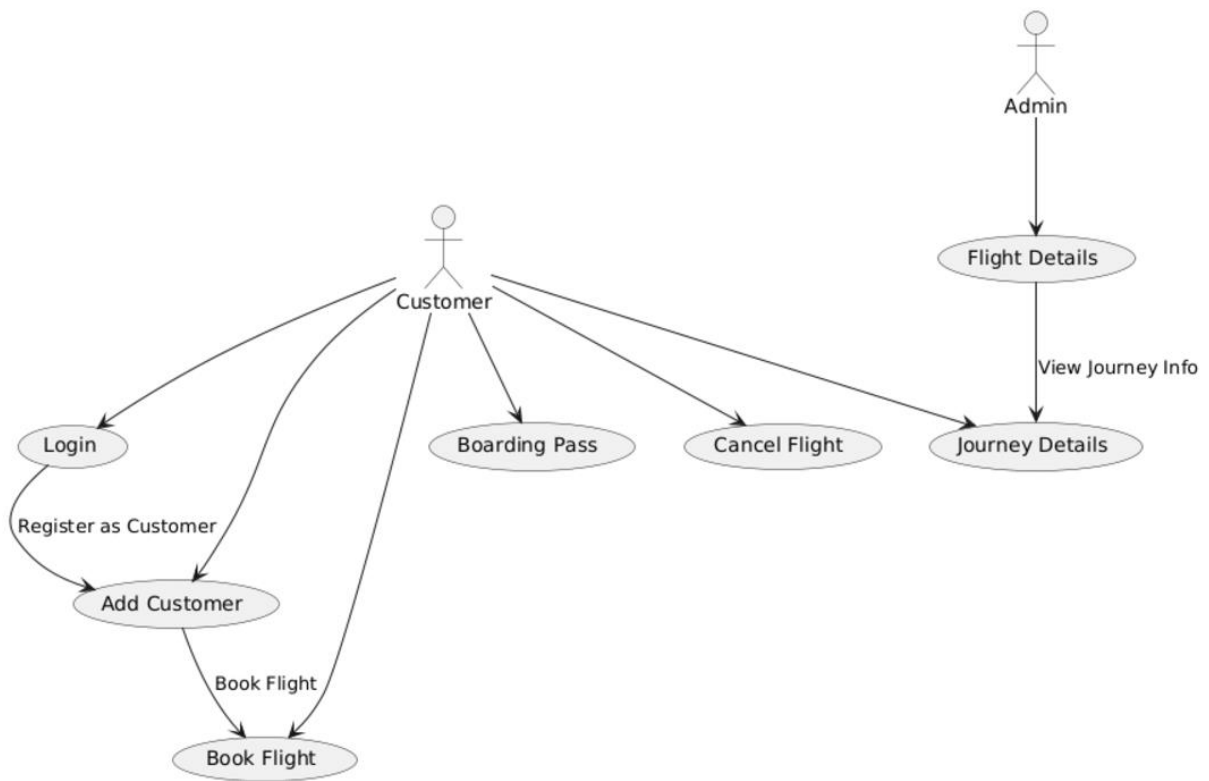
The system must have strong security measures on the server side to prevent unauthorized access or potential damage. All user data, including personal and private information, must be securely stored and protected. This ensures that sensitive details are safe from hackers and unauthorized users at all times.

## **Maintainability:**

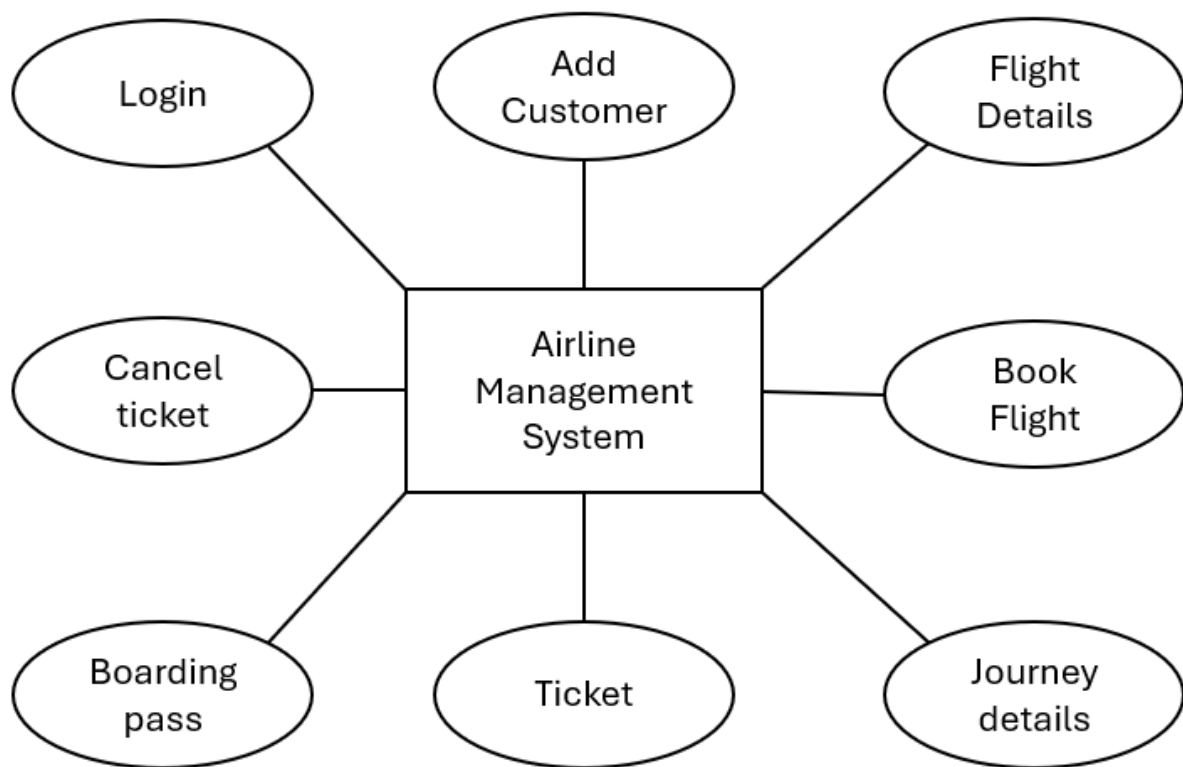
The system will have design documents that describe how to maintain both the software and the database, which stores user details and tracks daily updates or modifications. The administrator will have access to control and maintain the system, ensuring proper functioning both at the front end (user interface) and the back end (server and database). This will help ensure the system remains efficient, updated, and free of issues.

## **2 SYSTEM FLOW DIAGRAMS:**

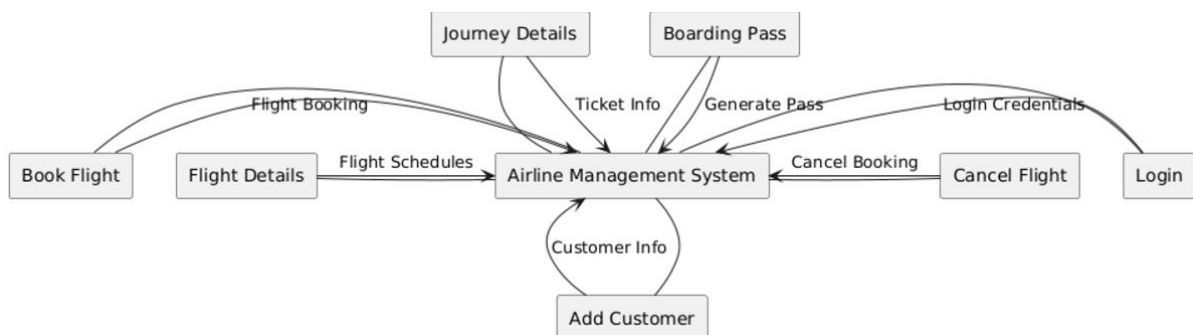
### **2.1 Use case Diagram:**



## 2.2 ER Model Diagram:



## 2.3 Data Flow Diagram:



## 3. MODULE DESCRIPTION:

The modules in the Airline Management System are:

### Flight Details:

Admin manages flight information like flight ID, destination, and schedule.

### **Add Customer Details:**

Admin adds customer information such as name and contact details.

### **Book Flight:**

Customers can book flights by choosing the desired flight.

### **Journey Details:**

Customers can view details of their booked flights.

### **Boarding Pass:**

Customers can view and download their boarding pass after booking a flight.

### **Cancel Ticket:**

Customers can cancel their flight tickets.

## **4. IMPLEMENTATION:**

### **4.1 CODE:**

#### **JDBC Connectivity:**

```
package airlinemanagementsystem;

import java.sql.*;

public class Conn {

    Connection c;

    Statement s;

    public Conn() {

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            C=DriverManager.getConnection("jdbc:mysql:///airlinemanagementsystem", "root", "WJ28@krhps");
```

```

        s = c.createStatement();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

### **Add customer:**

```
package airlinemanagementsystem;
```

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class AddCustomer extends JFrame implements ActionListener{
```

```
    JTextField tfname, tfphone, tfaadhar, tfnationality, tfaddress;
```

```
    JRadioButton rbmale, rbfemale;
```

```
    public AddCustomer() {
```

```
        getContentPane().setBackground(Color.WHITE);
```

```
        setLayout(null);
```

```
        JLabel heading = new JLabel("ADD CUSTOMER DETAILS");
```

```
        heading.setBounds(220, 20, 500, 35);
```

```
        heading.setFont(new Font("Tahoma", Font.PLAIN, 32));
```

```
        heading.setForeground(Color.BLUE);
```



```
add(heading);
```

```
JLabel lblname = new JLabel("Name");
```

```
lblname.setBounds(60, 80, 150, 25);
```

```
lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lblname);
```

```
tfname = new JTextField();
```

```
tfname.setBounds(220, 80, 150, 25);
```

```
add(tfname);
```

```
JLabel lblnationality = new JLabel("Nationality");
```

```
lblnationality.setBounds(60, 130, 150, 25);
```

```
lblnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lblnationality);
```

```
tfnationality = new JTextField();
```

```
tfnationality.setBounds(220, 130, 150, 25);
```

```
add(tfnationality);
```

```
JLabel lblaadhar = new JLabel("Aadhar Number");
```

```
lblaadhar.setBounds(60, 180, 150, 25);
```

```
lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lblaadhar);
```

```
tfaadhar = new JTextField();
```

```
tfaadhar.setBounds(220, 180, 150, 25);  
add(tfaadhar);
```

```
JLabel lbladdress = new JLabel("Address");  
lbladdress.setBounds(60, 230, 150, 25);  
lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lbladdress);
```

```
tfaddress = new JTextField();  
tfaddress.setBounds(220, 230, 150, 25);  
add(tfaddress);
```

```
JLabel lblgender = new JLabel("Gender");  
lblgender.setBounds(60, 280, 150, 25);  
lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblgender);
```

```
ButtonGroup gendergroup = new ButtonGroup();
```

```
rbmale = new JRadioButton("Male");  
rbmale.setBounds(220, 280, 70, 25);  
rbmale.setBackground(Color.WHITE);  
add(rbmale);
```

```
rbfemale = new JRadioButton("Female");  
rbfemale.setBounds(300, 280, 70, 25);
```

```
rbfemale.setBackground(Color.WHITE);  
add(rbfemale);
```

```
gendergroup.add(rbmale);  
gendergroup.add(rbfemale);
```

```
JLabel lblphone = new JLabel("Phone");  
lblphone.setBounds(60, 330, 150, 25);  
lblphone.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblphone);
```

```
tfphone = new JTextField();  
tfphone.setBounds(220, 330, 150, 25);  
add(tfphone);
```

```
JButton save = new JButton("SAVE");  
save.setBackground(Color.BLACK);  
save.setForeground(Color.WHITE);  
save.setBounds(220, 380, 150, 30);  
save.addActionListener(this);  
add(save);
```

```
ImageIcon image = new  
ImageIcon(ClassLoader.getResource("airlinemanagementsystem/icons/em  
p.png"));
```

```
JLabel lblimage = new JLabel(image);  
lblimage.setBounds(450, 80, 280, 400);
```

```

add(lblimage);

setSize(900, 600);
setLocation(300, 150);
setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    String name = tfname.getText();
    String nationality = tfnationality.getText();
    String phone = tfphone.getText();
    String address = tfaddress.getText();
    String aadhar = tfaadhar.getText();
    String gender = null;
    if (rbmale.isSelected()) {
        gender = "Male";
    } else {
        gender = "Female";
    }

    try {
        Conn conn = new Conn();

        String query = "insert into passenger values('"+name+"',
        '"+nationality+"', '"+phone+"', '"+address+"', '"+aadhar+"', '"+gender+"')";

        conn.s.executeUpdate(query);
    }
}

```

```
JOptionPane.showMessageDialog(null, "Customer Details Added  
Successfully");
```

```
setVisible(false);  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}
```

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

### **Boarding Pass:**

```
package airlinemanagementsystem;
```

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
import java.sql.*;  
import java.util.*;
```

```
public class BoardingPass extends JFrame implements ActionListener{
```

```
    JTextField tfpnr;
```

JLabel tfname, tfnationality, lblsrc, lbldest, labelfname, labelfcode, labeldate;

JButton fetchButton;

```
public BoardingPass() {
```

```
    getContentPane().setBackground(Color.WHITE);
```

```
    setLayout(null);
```

```
    JLabel heading = new JLabel("AIR INDIA");
```

```
    heading.setBounds(380, 10, 450, 35);
```

```
    heading.setFont(new Font("Tahoma", Font.PLAIN, 32));
```

```
    add(heading);
```

```
    JLabel subheading = new JLabel("Boarding Pass");
```

```
    subheading.setBounds(360, 50, 300, 30);
```

```
    subheading.setFont(new Font("Tahoma", Font.PLAIN, 24));
```

```
    subheading.setForeground(Color.BLUE);
```

```
    add(subheading);
```

```
    JLabel lblaadhar = new JLabel("PNR DETAILS");
```

```
    lblaadhar.setBounds(60, 100, 150, 25);
```

```
    lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
    add(lblaadhar);
```

```
    tfpnr = new JTextField();
```

```
    tfpnr.setBounds(220, 100, 150, 25);
```

```
    add(tfpnr);
```

```
fetchButton = new JButton("Enter");  
fetchButton.setBackground(Color.BLACK);  
fetchButton.setForeground(Color.WHITE);  
fetchButton.setBounds(380, 100, 120, 25);  
fetchButton.addActionListener(this);  
add(fetchButton);
```

```
JLabel lblname = new JLabel("NAME");  
lblname.setBounds(60, 140, 150, 25);  
lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblname);
```

```
tfname = new JLabel();  
tfname.setBounds(220, 140, 150, 25);  
add(tfname);
```

```
JLabel lblnationality = new JLabel("NATIONALITY");  
lblnationality.setBounds(60, 180, 150, 25);  
lblnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblnationality);
```

```
tfnationality = new JLabel();  
tfnationality.setBounds(220, 180, 150, 25);  
add(tfnationality);
```

```
JLabel lbladdress = new JLabel("SRC");  
lbladdress.setBounds(60, 220, 150, 25);  
lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lbladdress);
```

```
lblsrc = new JLabel();  
lblsrc.setBounds(220, 220, 150, 25);  
add(lblsrc);
```

```
JLabel lblgender = new JLabel("DEST");  
lblgender.setBounds(380, 220, 150, 25);  
lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblgender);
```

```
lbldest = new JLabel();  
lbldest.setBounds(540, 220, 150, 25);  
add(lbldest);
```

```
JLabel lblfname = new JLabel("Flight Name");  
lblfname.setBounds(60, 260, 150, 25);  
lblfname.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblfname);
```

```
labelfname = new JLabel();  
labelfname.setBounds(220, 260, 150, 25);  
add(labelfname);
```



```
JLabel lblfcode = new JLabel("Flight Code");  
lblfcode.setBounds(380, 260, 150, 25);  
lblfcode.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblfcode);
```

```
labelfcode = new JLabel();  
labelfcode.setBounds(540, 260, 150, 25);  
add(labelfcode);
```

```
JLabel lbldate = new JLabel("Date");  
lbldate.setBounds(60, 300, 150, 25);  
lbldate.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lbldate);
```

```
labeldate = new JLabel();  
labeldate.setBounds(220, 300, 150, 25);  
add(labeldate);
```

```
ImageIcon i1 = new  
ImageIcon(ClassLoader.getResource("airlinemanagementsystem/icons/a  
irindia.png"));
```

```
Image i2 = i1.getImage().getScaledInstance(300, 230,  
Image.SCALE_DEFAULT);
```

```
ImageIcon image = new ImageIcon(i2);  
JLabel lblimage = new JLabel(image);  
lblimage.setBounds(600, 0, 300, 300);
```

```

add(lblimage);

setSize(1000, 450);
setLocation(300, 150);
setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    String pnr = tfpnr.getText();

    try {
        Conn conn = new Conn();

        String query = "select * from reservation where PNR = '"+pnr+"'";

        ResultSet rs = conn.s.executeQuery(query);

        if (rs.next()) {
            tfname.setText(rs.getString("name"));
            tfnationality.setText(rs.getString("nationality"));
            lblsrc.setText(rs.getString("src"));
            lbldest.setText(rs.getString("des"));
            labelfname.setText(rs.getString("flightname"));
            labelfcode.setText(rs.getString("flightcode"));
            labeldate.setText(rs.getString("ddate"));
        } else {

```

```

        JOptionPane.showMessageDialog(null, "Please enter correct
PNR");
    }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

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

```

### **Book Flight:**

```

package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import com.toedter.calendar.JDateChooser;
import java.util.*;

public class BookFlight extends JFrame implements ActionListener{

    JTextField tfaadhar;

    JLabel tfname, tfnationality, tfaddress, labelgender, labelfname, labelfcode;

```

JButton bookflight, fetchButton, flight;

Choice source, destination;

JDateChooser dcdate;

public BookFlight() {

    getContentPane().setBackground(Color.WHITE);

    setLayout(null);

    JLabel heading = new JLabel("Book Flight");

    heading.setBounds(420, 20, 500, 35);

    heading.setFont(new Font("Tahoma", Font.PLAIN, 32));

    heading.setForeground(Color.BLUE);

    add(heading);

    JLabel lblaadhar = new JLabel("Aadhar");

    lblaadhar.setBounds(60, 80, 150, 25);

    lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));

    add(lblaadhar);

    JTextField tfaadhar = new JTextField();

    tfaadhar.setBounds(220, 80, 150, 25);

    add(tfaadhar);

    JButton fetchButton = new JButton("Fetch User");

    fetchButton.setBackground(Color.BLACK);

    fetchButton.setForeground(Color.WHITE);

```
fetchButton.setBounds(380, 80, 120, 25);
```

```
fetchButton.addActionListener(this);
```

```
add(fetchButton);
```

```
JLabel lblname = new JLabel("Name");
```

```
lblname.setBounds(60, 130, 150, 25);
```

```
lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lblname);
```

```
tfname = new JLabel();
```

```
tfname.setBounds(220, 130, 150, 25);
```

```
add(tfname);
```

```
JLabel lblnationality = new JLabel("Nationality");
```

```
lblnationality.setBounds(60, 180, 150, 25);
```

```
lblnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lblnationality);
```

```
tfnationality = new JLabel();
```

```
tfnationality.setBounds(220, 180, 150, 25);
```

```
add(tfnationality);
```

```
JLabel lbladdress = new JLabel("Address");
```

```
lbladdress.setBounds(60, 230, 150, 25);
```

```
lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lbladdress);
```

```
tfaddress = new JLabel();  
tfaddress.setBounds(220, 230, 150, 25);  
add(tfaddress);
```

```
JLabel lblgender = new JLabel("Gender");  
lblgender.setBounds(60, 280, 150, 25);  
lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblgender);
```

```
labelgender = new JLabel("Gender");  
labelgender.setBounds(220, 280, 150, 25);  
add(labelgender);
```

```
JLabel lblsource = new JLabel("Source");  
lblsource.setBounds(60, 330, 150, 25);  
lblsource.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblsource);
```

```
source = new Choice();  
source.setBounds(220, 330, 150, 25);  
add(source);
```

```
JLabel lbldest = new JLabel("Destination");  
lbldest.setBounds(60, 380, 150, 25);  
lbldest.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```

add(lbldest);

destination = new Choice();
destination.setBounds(220, 380, 150, 25);
add(destination);

try {
    Conn c = new Conn();
    String query = "select * from flight";
    ResultSet rs = c.s.executeQuery(query);

    while(rs.next()) {
        source.add(rs.getString("source"));
        destination.add(rs.getString("destination"));
    }

} catch (Exception e) {
    e.printStackTrace();
}

flight = new JButton("Fetch Flights");
flight.setBackground(Color.BLACK);
flight.setForeground(Color.WHITE);
flight.setBounds(380, 380, 120, 25);
flight.addActionListener(this);
add(flight);

```

```
JLabel lblfname = new JLabel("Flight Name");  
lblfname.setBounds(60, 430, 150, 25);  
lblfname.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblfname);
```

```
labelfname = new JLabel();  
labelfname.setBounds(220, 430, 150, 25);  
add(labelfname);
```

```
JLabel lblfcode = new JLabel("Flight Code");  
lblfcode.setBounds(60, 480, 150, 25);  
lblfcode.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblfcode);
```

```
labelfcode = new JLabel();  
labelfcode.setBounds(220, 480, 150, 25);  
add(labelfcode);
```

```
JLabel lbldate = new JLabel("Date of Travel");  
lbldate.setBounds(60, 530, 150, 25);  
lbldate.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lbldate);
```

```
dcdate = new JDateChooser();  
dcdate.setBounds(220, 530, 150, 25);
```



```
add(dcddate);
```

```
    ImageIcon i1 = new  
ImageIcon(ClassLoader.getResource("airlinemanagementsystem/icons/d  
etails.jpg"));
```

```
    Image i2 = i1.getImage().getScaledInstance(450, 320,  
Image.SCALE_DEFAULT);
```

```
    ImageIcon image = new ImageIcon(i2);
```

```
    JLabel lblimage = new JLabel(image);
```

```
    lblimage.setBounds(550, 80, 500, 410);
```

```
    add(lblimage);
```

```
    bookflight = new JButton("Book Flight");
```

```
    bookflight.setBackground(Color.BLACK);
```

```
    bookflight.setForeground(Color.WHITE);
```

```
    bookflight.setBounds(220, 580, 150, 25);
```

```
    bookflight.addActionListener(this);
```

```
    add(bookflight);
```

```
    setSize(1100, 700);
```

```
    setLocation(200, 50);
```

```
    setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent ae) {
```

```
    if (ae.getSource() == fetchButton) {
```

```
        String aadhar = tfaadhar.getText();
```

```

try {
    Conn conn = new Conn();

    String query = "select * from passenger where aadhar =
"+aadhar+"";

    ResultSet rs = conn.s.executeQuery(query);

    if (rs.next()) {
        tfname.setText(rs.getString("name"));
        tfnationality.setText(rs.getString("nationality"));
        tfaddress.setText(rs.getString("address"));
        labelgender.setText(rs.getString("gender"));
    } else {
        JOptionPane.showMessageDialog(null, "Please enter correct
aadhar");
    }
} catch (Exception e) {
    e.printStackTrace();
}

} else if (ae.getSource() == flight) {
    String src = source.getSelectedItemAt();
    String dest = destination.getSelectedItemAt();
    try {
        Conn conn = new Conn();

```

```
String query = "select * from flight where source = '"+src+"' and  
destination = '"+dest+"'";
```

```
ResultSet rs = conn.s.executeQuery(query);
```

```
if (rs.next()) {  
    labelfname.setText(rs.getString("f_name"));  
    labelfcode.setText(rs.getString("f_code"));  
} else {  
    JOptionPane.showMessageDialog(null, "No Flights  
Found");  
}  
} catch (Exception e) {  
    e.printStackTrace();  
}  
} else {
```

```
Random random = new Random();
```

```
String aadhar = tfaadhar.getText();
```

```
String name = tfname.getText();
```

```
String nationality = tfnationality.getText();
```

```
String flightname = labelfname.getText();
```

```
String flightcode = labelfcode.getText();
```

```
String src = source.getSelectedItem();
```

```
String des = destination.getSelectedItem();
```

```
String ddate = ((JTextField)  
dcdde.getDateEditor().getUiComponent()).getText();
```

```

try {
    Conn conn = new Conn();

    String query = "insert into reservation values('PNR-
"+random.nextInt(1000000)+"', 'TIC-"+random.nextInt(10000)+"',
"+aadhar+"', '"+name+"', '"+nationality+"', '"+flightname+"', '"+flightcode+"',
"+src+"', '"+des+"', '"+ddate+"')";

    conn.s.executeUpdate(query);

    JOptionPane.showMessageDialog(null, "Ticket Booked
Successfully");

    setVisible(false);
} catch (Exception e) {
    e.printStackTrace();
}
}
}

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

```

### **Cancel Flight:**

```
package airlinemanagementsystem;
```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;

public class Cancel extends JFrame implements ActionListener{

    JTextField tfpnr;
    JLabel tfname, cancellationno, lblfcode, lbldateoftravel;
    JButton fetchButton, flight;

    public Cancel() {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        Random random = new Random();

        JLabel heading = new JLabel("CANCELLATION");
        heading.setBounds(180, 20, 250, 35);
        heading.setFont(new Font("Tahoma", Font.PLAIN, 32));
        add(heading);

        ImageIcon i1 = new
        ImageIcon(ClassLoader.getResource("airlinemanagementsystem/icons/c
        ancel.jpg"));

```

```
Image i2 = i1.getImage().getScaledInstance(250, 250,  
Image.SCALE_DEFAULT);
```

```
ImageIcon i3 = new ImageIcon(i2);
```

```
JLabel image = new JLabel(i3);
```

```
image.setBounds(470, 120, 250, 250);
```

```
add(image);
```

```
JLabel lblaadhar = new JLabel("PNR Number");
```

```
lblaadhar.setBounds(60, 80, 150, 25);
```

```
lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lblaadhar);
```

```
tfpnr = new JTextField();
```

```
tfpnr.setBounds(220, 80, 150, 25);
```

```
add(tfpnr);
```

```
fetchButton = new JButton("Show Details");
```

```
fetchButton.setBackground(Color.BLACK);
```

```
fetchButton.setForeground(Color.WHITE);
```

```
fetchButton.setBounds(380, 80, 120, 25);
```

```
fetchButton.addActionListener(this);
```

```
add(fetchButton);
```

```
JLabel lblname = new JLabel("Name");
```

```
lblname.setBounds(60, 130, 150, 25);
```

```
lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lblname);
```

```
tfname = new JLabel();
```

```
tfname.setBounds(220, 130, 150, 25);
```

```
add(tfname);
```

```
JLabel lblnationality = new JLabel("Cancellation No");
```

```
lblnationality.setBounds(60, 180, 150, 25);
```

```
lblnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lblnationality);
```

```
cancellationno = new JLabel("" + random.nextInt(1000000));
```

```
cancellationno.setBounds(220, 180, 150, 25);
```

```
add(cancellationno);
```

```
JLabel lbladdress = new JLabel("Flight Code");
```

```
lbladdress.setBounds(60, 230, 150, 25);
```

```
lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lbladdress);
```

```
lblfcode = new JLabel();
```

```
lblfcode.setBounds(220, 230, 150, 25);
```

```
add(lblfcode);
```

```
JLabel lblgender = new JLabel("Date");
```

```
lblgender.setBounds(60, 280, 150, 25);
```

```
lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lblgender);
```

```
lbldateoftravel = new JLabel();
```

```
lbldateoftravel.setBounds(220, 280, 150, 25);
```

```
add(lbldateoftravel);
```

```
flight = new JButton("Cancel");
```

```
flight.setBackground(Color.BLACK);
```

```
flight.setForeground(Color.WHITE);
```

```
flight.setBounds(220, 330, 120, 25);
```

```
flight.addActionListener(this);
```

```
add(flight);
```

```
setSize(800, 450);
```

```
setLocation(350, 150);
```

```
setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent ae) {
```

```
    if (ae.getSource() == fetchButton) {
```

```
        String pnr = tfpnr.getText();
```

```
        try {
```

```
            Conn conn = new Conn();
```

```
            String query = "select * from reservation where PNR = '"+pnr+"'";
```



```

ResultSet rs = conn.s.executeQuery(query);

if (rs.next()) {
    tfname.setText(rs.getString("name"));
    lblfcode.setText(rs.getString("flightcode"));
    lbldateoftravel.setText(rs.getString("ddate"));
} else {
    JOptionPane.showMessageDialog(null, "Please enter correct
PNR");
}
} catch (Exception e) {
    e.printStackTrace();
}
} else if (ae.getSource() == flight) {
    String name = tfname.getText();
    String pnr = tfpnr.getText();
    String cancelno = cancellationno.getText();
    String fcode = lblfcode.getText();
    String date = lbldateoftravel.getText();

    try {
        Conn conn = new Conn();

        String query = "insert into cancel values('"+pnr+"', '"+name+"',
        '"+cancelno+"', '"+fcode+"', '"+date+"')";

```

```
        conn.s.executeUpdate(query);
        conn.s.executeUpdate("delete from reservation where PNR =
        '"+pnr+"'");
```

```
        JOptionPane.showMessageDialog(null, "Ticket Cancelled");
        setVisible(false);
```

```
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

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

### **Flight Information:**

```
package airlinemanagementsystem;
```

```
import javax.swing.*;
import java.awt.*;
import java.sql.*;
import net.proteanit.sql.DbUtils;
```

```
public class FlightInfo extends JFrame{
```

```
public FlightInfo() {

    getContentPane().setBackground(Color.WHITE);
    setLayout(null);

    JTable table = new JTable();

    try {
        Conn conn = new Conn();

        ResultSet rs = conn.s.executeQuery("select * from flight");
        table.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        e.printStackTrace();
    }

    JScrollPane jsp = new JScrollPane(table);
    jsp.setBounds(0, 0, 800, 500);
    add(jsp);

    setSize(800, 500);
    setLocation(400, 200);
    setVisible(true);
}
```

```

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

```

## **Home:**

```

package airlinemanagementsystem;

```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

```

```

public class Home extends JFrame implements ActionListener{

```

```

    public Home() {
        setLayout(null);

```

```

        ImageIcon i1 = new
        ImageIcon(ClassLoader.getResource("airlinemanagementsystem/icons/f
ront.jpg"));

```

```

        JLabel image = new JLabel(i1);
        image.setBounds(0, 0, 1600, 800);
        add(image);

```

```

        JLabel heading = new JLabel("AIR INDIA WELCOMES YOU");
        heading.setBounds(500, 40, 1000, 40);
        heading.setForeground(Color.BLUE);
        heading.setFont(new Font("Tahoma", Font.PLAIN, 36));

```

```
image.add(heading);
```

```
JMenuBar menubar = new JMenuBar();
```

```
setJMenuBar(menubar);
```

```
JMenu details = new JMenu("Details");
```

```
menubar.add(details);
```

```
JMenuItem flightDetails = new JMenuItem("Flight Details");
```

```
flightDetails.addActionListener(this);
```

```
details.add(flightDetails);
```

```
JMenuItem customerDetails = new JMenuItem("Add Customer Details");
```

```
customerDetails.addActionListener(this);
```

```
details.add(customerDetails);
```

```
JMenuItem bookFlight = new JMenuItem("Book Flight");
```

```
bookFlight.addActionListener(this);
```

```
details.add(bookFlight);
```

```
JMenuItem journeyDetails = new JMenuItem("Journey Details");
```

```
journeyDetails.addActionListener(this);
```

```
details.add(journeyDetails);
```

```
JMenuItem ticketCancellation = new JMenuItem("Cancel Ticket");
```

```
ticketCancellation.addActionListener(this);
```

```
details.add(ticketCancellation);
```

```
JMenu ticket = new JMenu("Ticket");
```

```
menubar.add(ticket);
```

```
JMenuItem boardingPass = new JMenuItem("Boarding Pass");
```

```
ticket.add(boardingPass);
```

```
setExtendedState(JFrame.MAXIMIZED_BOTH);
```

```
setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent ae) {
```

```
    String text = ae.getActionCommand();
```

```
    if (text.equals("Add Customer Details")) {
```

```
        new AddCustomer();
```

```
    } else if (text.equals("Flight Details")) {
```

```
        new FlightInfo();
```

```
    } else if (text.equals("Book Flight")) {
```

```
        new BookFlight();
```

```
    } else if (text.equals("Journey Details")) {
```

```
        new JourneyDetails();
```

```
    } else if (text.equals("Cancel Ticket")) {
```

```
        new Cancel();
```

```

    }
}

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

```

### **Journey Details:**

```

package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.sql.*;
import java.awt.event.*;
import net.proteanit.sql.DbUtils;

public class JourneyDetails extends JFrame implements ActionListener{
    JTable table;
    JTextField pnr;
    JButton show;

    public JourneyDetails() {

        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
    }
}

```

```
JLabel lblpnr = new JLabel("PNR Details");  
lblpnr.setFont(new Font("Tahoma", Font.PLAIN, 16));  
lblpnr.setBounds(50, 50, 100, 25);  
add(lblpnr);
```

```
pnr = new JTextField();  
pnr.setBounds(160, 50, 120, 25);  
add(pnr);
```

```
show = new JButton("Show Details");  
show.setBackground(Color.BLACK);  
show.setForeground(Color.WHITE);  
show.setBounds(290, 50, 120, 25);  
show.addActionListener(this);  
add(show);
```

```
table = new JTable();
```

```
JScrollPane jsp = new JScrollPane(table);  
jsp.setBounds(0, 100, 800, 150);  
jsp.setBackground(Color.WHITE);  
add(jsp);
```

```
setSize(800, 600);  
setLocation(400, 150);  
setVisible(true);
```



```

    }

    public void actionPerformed(ActionEvent ae) {
        try {
            Conn conn = new Conn();

            ResultSet rs = conn.s.executeQuery("select * from reservation where
PNR = '"+pnr.getText()+"");

            if (!rs.isBeforeFirst()) {
                JOptionPane.showMessageDialog(null, "No Information Found");
                return;
            }

            table.setModel(DbUtils.resultSetToTableModel(rs));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

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

```

### **Login:**

```

package airlinemanagementsystem;

import javax.swing.*;

```

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class Login extends JFrame implements ActionListener{

    JButton submit, reset, close;
    JTextField tfusername;
    JPasswordField tfpassword;

    public Login() {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel lblusername = new JLabel("Username");
        lblusername.setBounds(20, 20, 100, 20);
        add(lblusername);

        tfusername = new JTextField();
        tfusername.setBounds(130, 20, 200, 20);
        add(tfusername);

        JLabel lblpassword = new JLabel("Password");
        lblpassword.setBounds(20, 60, 100, 20);
        add(lblpassword);

        tfpassword = new JPasswordField();
```

```
tfpassword.setBounds(130, 60, 200, 20);  
add(tfpassword);
```

```
reset = new JButton("Reset");  
reset.setBounds(40, 120, 120, 20);  
reset.addActionListener(this);  
add(reset);
```

```
submit = new JButton("Submit");  
submit.setBounds(190, 120, 120, 20);  
submit.addActionListener(this);  
add(submit);
```

```
close = new JButton("Close");  
close.setBounds(120, 160, 120, 20);  
close.addActionListener(this);  
add(close);
```

```
setSize(400, 250);  
setLocation(600, 250);  
setVisible(true);  
}
```

```
public void actionPerformed(ActionEvent ae) {  
    if (ae.getSource() == submit) {  
        String username = tfusername.getText();
```

```

String password = tfpassword.getText();

try {
    Conn c = new Conn();

    String query = "select * from login where username = '"+username+"'
and password = '"+password+"'";

    ResultSet rs = c.s.executeQuery(query);

    if (rs.next()) {
        new Home();
        setVisible(false);
    } else {
        JOptionPane.showMessageDialog(null, "Invalid Username or
Password");
        setVisible(false);
    }
} catch (Exception e) {
    e.printStackTrace();
}

} else if (ae.getSource() == close) {
    setVisible(false);
} else if (ae.getSource() == reset) {
    tfusername.setText("");
    tfpassword.setText("");
}

```

```
}

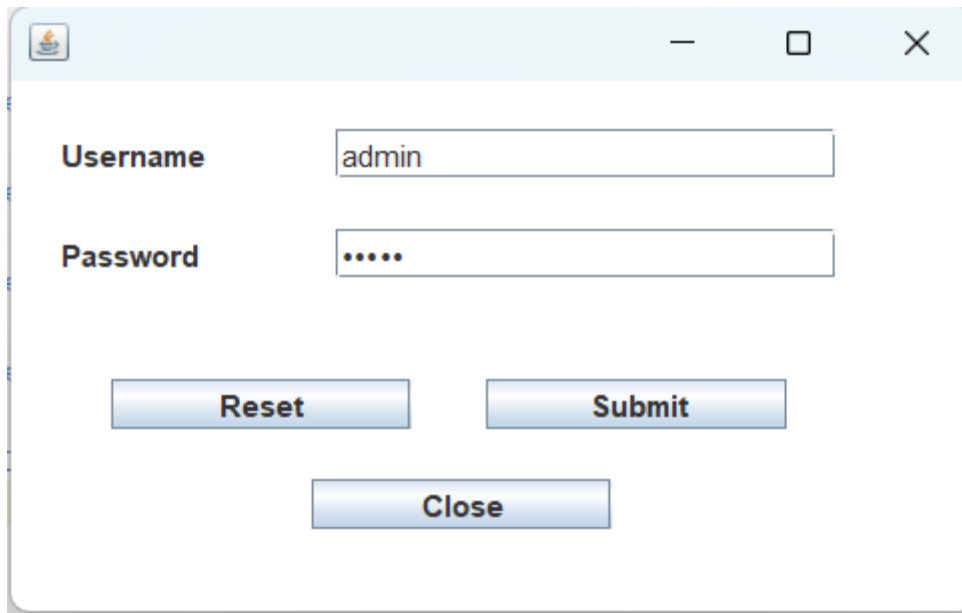
public static void main(String[] args) {
    new Login();
}
}
```

## 4.2 OUTPUT SCREENS:

### Homepage




### Login



A screenshot of a login window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main content area contains two text input fields: the first is labeled "Username" and contains the text "admin"; the second is labeled "Password" and contains five dots. Below these fields are three buttons: "Reset" and "Submit" are positioned side-by-side, and "Close" is centered below them.

## Add Customer:



A screenshot of a form titled "ADD CUSTOMER DETAILS" in blue text. The form is contained within a window with a title bar. The form fields are as follows:

- Name: Vasumathi
- Nationality: Indian
- Aadhar Number: 2134567890
- Address: Nellorepet
- Gender: ☐ Male ☒ Female
- Phone: 4343434343

Below the form fields is a black button labeled "SAVE". To the right of the form is a large black silhouette of a person standing with hands on hips.

## Book Flight

Book Flight

Aadhar

1234567890

Fetch User

Name

Lavanya

Nationality

Indian

Address

69,Big

Gender

Female

Source

Delhi

Destination

Mumbai

Fetch Flights

Flight Name

Flight Code

Date of Travel

22 Nov 2024


Book Flight

## Flight Details

f_code	f_name	source	destination
1001	AI-1212	Delhi	Mumbai
1002	AI-1453	Delhi	Goa
1003	AI-1112	Mumbai	Chennai
1004	AI-3222	Delhi	Amritsar
1005	AI-1212	Delhi	Ayodhya

55 | Page

## Journey Details

 — □ ×

PNR Details

Show Details

PNR	TICKET	aadhar	name	nationality	flightname	flightcode	src	des	ddate
PNR-902429	TIC-5398	1234567890	Lavanya	Indian	AI-1212	1001	Delhi	Mumbai	21 Nov 2024

## Boarding Pass

 — □ ×

AIR INDIA

Boarding Pass

PNR DETAILS

Enter

NAME

Lavanya

NATIONALITY

Indian

SRC

Delhi

DEST

Mumbai

Flight Name

AI-1212

Flight Code

1001

Date

21 Nov 2024



## Cancel Flight





## **CONCLUSION:**

The project titled as Airline Management System was deeply studied and analyzed to design the code and implement. It was done under the guidance of the experienced project guide. All the current requirements and possibilities have been taken care during the project time.

## **REFERENCE LINKS:**

[1]

<https://developers.openshift.com/database/mysqlml>

[2]Web References- <https://youtu.be/UbIIFLsEeiM>