

A Project Report
On
AIRLINE RESERVATION SYSTEM

By
Ankit Bhawsar (307015)
Anup Bandawar (307011)
Omkar Aurangabadkar (307008)

Under the guidance of

B.P. Vasgi



Department of Information Technology
Sinhgad College of Engineering

SAVITRIBAI PHULE PUNE UNIVERSITY

2017-2018

Sinhgad Technical Education Society,
Department of Information Technology
Sinhgad College of Engineering , Pune-41



Sinhgad Institutes

Date:

CERTIFICATE

This is to certify that,

Ankit Bhawsar (307015)
Anup Bandawar (307011)
Omkar Aurangabadkar (307008)

of class T.E IT; have successfully completed their project work on “Airline Reservation System” at SINHGAD COLLEGE OF ENGINEERING in the partial fulfillment of the Graduate Degree course in T.E at the Department of Information Technology, in the academic Year 2017-2018 Semester – I as prescribed by the Savitribai Phule Pune University.

B.P. Vasgi
Guide

Prof.G. R. Pathak
Head of the Department
(Department of Information Technology)

Acknowledgement

I feel great pleasure in expressing my deepest sense of gratitude and sincere thanks to my guide **B.P. Vasgi** for her valuable guidance during the Project work, without which it would have been very difficult task. I have no words to express my sincere thanks for valuable guidance, extreme assistance and cooperation extended to all the **Staff Members** of my Department.

This acknowledgement would be incomplete without expressing my special thanks to **Prof.G. R. Pathak** Head of the Department (Information Technology) for their support during the work.

I would also like to extend my heartfelt gratitude to my **Principal, Dr. S. D. Lokhande** who provided a lot of valuable support, mostly being behind the veils of college bureaucracy.

Last but not least I would like to thanks all the Teaching, Non- Teaching staff members of my Department, my parent and my colleagues those who helped me directly or indirectly for completing of this Project successfully.

Ankit Bhawsar

Contents

1. TITLE OF THE PROJECT

2. ABSTRACT

3. INTRODUCTION

- Problem definition

4. SCOPE

5. SPECIFIC REQUIRMENTS

- Hardware Interface
- Software Interface

6. THEORY OF SOFTWARE USED

- Java (JDK)
- MYSQL

7. DATABASE SCHEMA

8. EER DIAGRAM

9. OUTPUT SCREEN (GUI)

10.SAMPLE CODE

11.CONCLUSION

12.REFERENCES

2. ABSTRACT

Airline reservation System is a computerized system used to store and retrieve information and conduct transactions related to air travel. The project is aimed at exposing the relevance and importance of Airline Reservation Systems. It is projected towards enhancing the relationship between customers and airline agencies through the use of Database Management Systems (DBMS) and an interactive, user-friendly interface, and thereby making it convenient for the customers to book the flights as and when they require such that they can utilize this software to make reservations. This software has two parts: A back-end comprised of My-Sql Database, where all relevant data is stored, and an innovative front-end made up of jForms using Java.

The Software is made with two types of clientele in mind: customer and administrator. Administrator accounts are used by airline authorities. They can use this software in order to avail a large number of functionalities, such as creating a chartered flight, setting ticket costs, blocking or banning disruptive customers, adding locations for flights, and also cancelling said flights. Customers on the other hand can use this software to easily book tickets for flights and print said ticket at the touch of a button. Customers can use various methods to pay for tickets such as credit cards, net banking and also an inbuilt credit system managed by the software, allowing them to book airline tickets with ease and comfort.

Customers and Administrators each have their own login pages by which they can access the functionalities of the website. Any new customers will have to register for an account before they can go ahead to their home page. Admin Accounts on the other hand, can only be created using existing admin accounts so as to prevent unapproved creation of flights. Once logged in, user can seamlessly navigate through the website to book tickets, cancel tickets, print tickets, and update account details.

3. INTRODUCTION

3.1 Existing System

Currently there are many different methods available of booking airline tickets which include buying tickets online at airline websites, airports, brokers, travel agencies, etc. All these systems seem to be outdated. Most of these methods prey on the customers, making them waste time in lines or even pay surcharges and broker fees.

What we have attempted to do is close the gap between airline and end-user, and thereby eliminating the use of middle-men. Customers can buy tickets directly from the Airline Companies without having to pay any extra rates or charges. They can also directly compare flight prices offered by various airlines at the same time and choose the ticket at the best possible price.

3.2 Problem Definition

To develop a computerized reservation system meeting the rising customer interest in booking airline reservations. The system should be convenient, user-friendly and available to users as well as Airline companies.

The system should allow Airline Companies or 'Admins' to create new flights by specifying various parameters such as date, time, source, destination, number of seats, airplane used as well as stating amount of seats being offered to customers and the price of each of them.

The System should allow the user or customer to view entire flight information of the flight, book tickets, view or print tickets and if required, cancel tickets, upon which they should be eligible for a refund.

4. SCOPE

Airline Reservation System make the life of passengers very easy as they don't need to stand in queues for getting their seats reserved and they can easily make reservations on any airline just from a single system. The primary aim of the system is to speed up transactions. Messages are displayed in message boxes to make the system user friendly. The main advantage of the system is the reduction in labor as it will possible so search the details of various places. This Airline Reservation System helps both Passenger as well as Admin by giving different functionalities.

5. SPECIFIC REQUIRMENTS

The proposed system analysis contains a planning and design phases where a logical design of system is developed and to work accordingly a plan is established. Also the requirements of system are identified and the operating environment is identified.

- **Hardware**
 - A computer acting as a Host and Server

- **Software**
 - MySQL Server
 - Java to MySql Driver
 - NetBeans IDE 8.2

6. THEORY OF SOFTWARE USED

JAVA

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". The java is independent to platform so it's important. Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications.

James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time. The language was initially called Oak after an oak tree that stood outside Gosling's office; it went by the name Green later, and was later renamed Java, from a list of random words. Gosling aimed to implement a virtual machine and a language that had a familiar C/C++ style of notation.

Java is an object-oriented programming language developed by Sun Microsystems in 1990s. Since then, Java has gained enormous popularity as a computer language. Java was chosen as the programming language for network computers. It is a universal front end for enterprise database. Sun Microsystems states that, "Java is a simple, object-oriented, distributed, secure, architecture, robust, multi-threaded and dynamic language. The program can be written once, and run anywhere". One of the most significant advantages of Java is that, it has the ability to move easily from one computer to another. It also has the ability to run the same program on many different operating systems. With such exemplary benefits, Java is a hot favourite among techies and software professionals it allows you to create modular programs and reusable codes .

Java Features

1] Simple, Small and familiar

Java is a simple and small language. The Syntax of java is just like c++, so it is very easy to learn. It is simple because it

- i) does not use header files
- ii) eliminated the use of pointer
- iii) operator overloading and virtual base classes are eliminated.

2] Object oriented

Java is a pure Object oriented . everything in java is object. all programs and data reside inside objects and classes

3] Distributed

Java has networking facilities. so java can create application on network.

4] Robust

java gives importance to memory management by using the technique called Garbage Collection and Exception handling.

5] Secure

since java is used on internet, security is an important issue. A security code is asked before a java code is interpreted on internet.

6] Platform independent

Java compiler generates an platform independent code called bytecode.

7] Portable

The Bytecode generated by java can be used on any machine. So it can be portable.

8] Compiled and Interpreted

Generally computer languages are either compiled or interpreted. but java combines both

compiler and interpreted.

9] High performance

The use of bytecode makes the performance high. the speed is also high with comparing c, c++.

10] Multithreading and interactive

Multithreading means handling more than one job at a time. Java supports Multithreading.

11] Dynamic and extensible

Java is a dynamic language. So it is capable of linking dynamic new classes, methods and objects. Java supports functions written in c and c++ also. These functions are called native methods. During Run-Time Native methods can be linked dynamically.

MySQL Database

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of “My”, the name of co-founder Michael Widenius’ daughter, and “SQL”, the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other “AMP” stacks). LAMP is an acronym for “Linux, Apache, MySQL, Perl/PHP/Python”. Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. MySQL is an essential part of almost every open source PHP application.

5.2.1 MySQL Features

1] Easy to use:

MySQL is easy to use. You have to get only the basic knowledge of SQL. You can build and interact with MySQL with only a few simple SQL statements.

2] It is secure:

MySQL consist of a solid data security layer that protects sensitive data from intruders. Passwords are encrypted in MySQL.

3] Client/ Server Architecture:

MySQL follows a client /server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc.

4] Free to download:

MySQL is free to use and you can download it from MySQL official website.

5] It is scalable:

MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.

6] Compatible on many operating systems:

MySQL is compatible to run on many operating systems, like Novell NetWare, Windows*, Linux*, many varieties of UNIX* (such as Sun* Solaris*, AIX, and DEC* UNIX), OS/2, FreeBSD*, and others. MySQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

7] Allows roll-back:

MySQL allows transactions to be rolled back, commit and crash recovery.

8] High Performance:

MySQL is faster, more reliable and cheaper because of its unique storage engine architecture.

9] High Flexibility:

MySQL supports a large number of embedded applications which makes MySQL very flexible.

10] High Productivity:

MySQL uses Triggers, Stored procedures and views which allows the developer to give a higher productivity.

7. DATABASE SCHEMA

1) admin

Column	Type	Null	Key	Default
admin_name	varchar(20)	No	PRI	
admin_password	varchar(20)	No		
active	tinyint(1)	No		0

2) flight

Column	Type	Null	Key	Default
flight_no	varchar(6)	No	PRI	
flight_name	varchar(50)	No		
airplane_name	varchar(50)	Yes		NULL
sid	int(20)	No	MUL	
did	int(20)	No	MUL	
time1	datetime	No		
time2	datetime	No		
no_first	int(5)	Yes		NULL
no_business	int(5)	No		
no_economy	int(5)	No		
price_first	int(5)	Yes		NULL
price_business	int(5)	No		
price_economy	int(5)	No		
admin_name	varchar(20)	Yes	MUL	NULL

3) user

Column	Type	Null	Key	Default
username	varchar(50)	No	PRI	
user_password	varchar(50)	No		
fname	varchar(50)	No		
lname	varchar(50)	No		
email	varchar(50)	No		
address	varchar(1000)	No		
phno	bigint(20)	No		
passno	varchar(50)	No		
credits	int(50)	Yes		0
block_state	int(1)	Yes		0
active_user	tinyint(1)	No		0

4) location

Column	Type	Null	Key	Default
loc_id	int(20)	No	PRI	0
airport	varchar(20)	No		
city	varchar(20)	Yes		
no_of_runways	int(20)	Yes		

5) booking

Column	Type	Null	Key	Default
booking_id	int(11)	No	PRI	
flight_no	varchar(50)	Yes	MUL	NULL
no_first_booked	int(5)	Yes		NULL
no_business_booked	int(5)	Yes		NULL
no_economy_booked	int(5)	Yes		NULL
total_price	int(10)	Yes		NULL
username	varchar(50)	Yes	MUL	NULL
paid	tinyint(1)	No		0

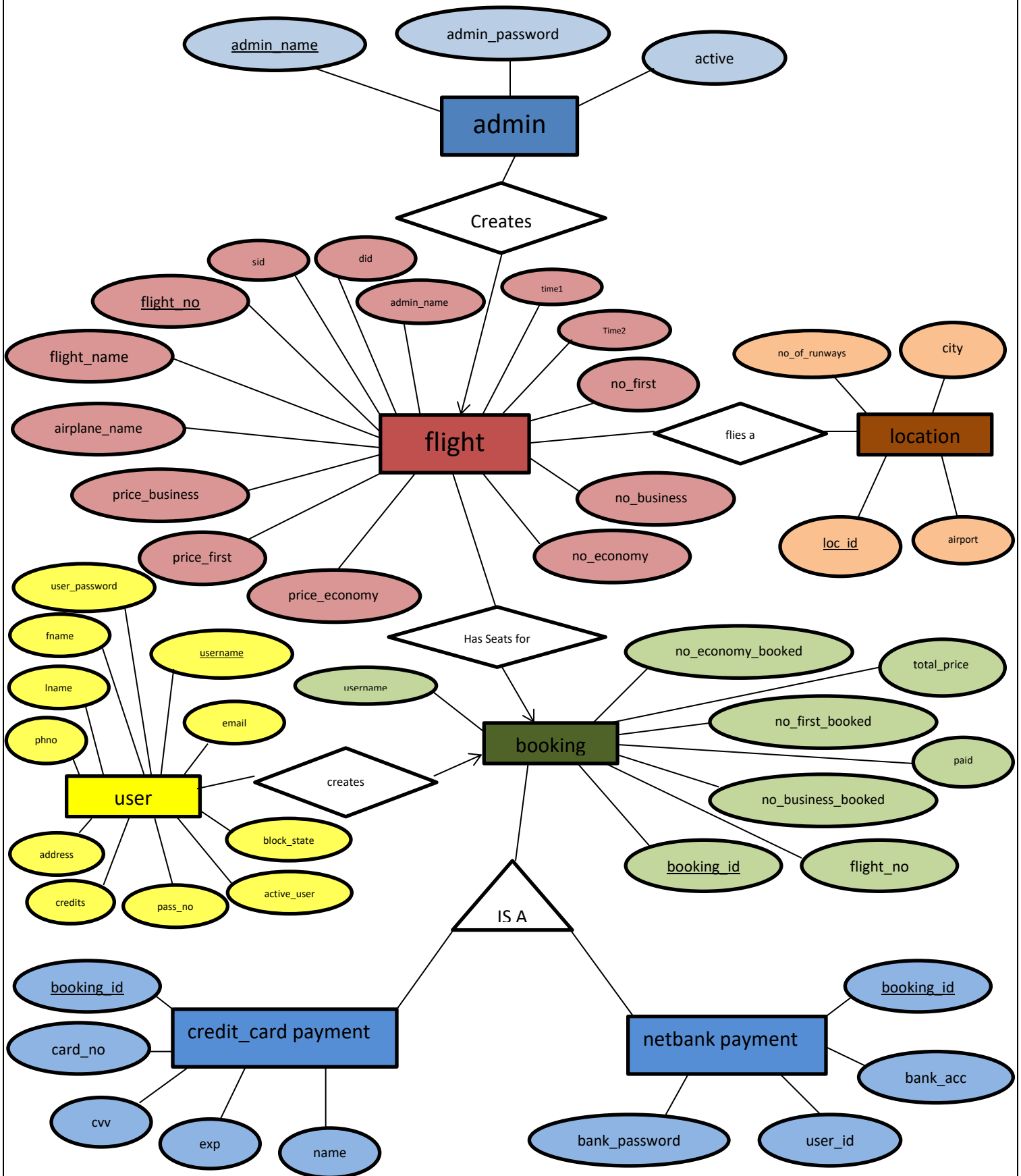
6) credit_card

Column	Type	Null	Key	Default
booking_id	int(11)	No	PRI	
card_no	bigint(20)	Yes		NULL
cvv	int(10)	Yes		NULL
exp	varchar(3)	Yes		NULL
name	varchar(50)	Yes		NULL

7) netbank

Column	Type	Null	Key	Default
booking_id	int(11)	No	PRI	
bank_acc	bigint(20)	Yes		NULL
user_id	varchar(20)	Yes		NULL
bank_password	varchar(20)	Yes		NULL

8. EER DIAGRAM

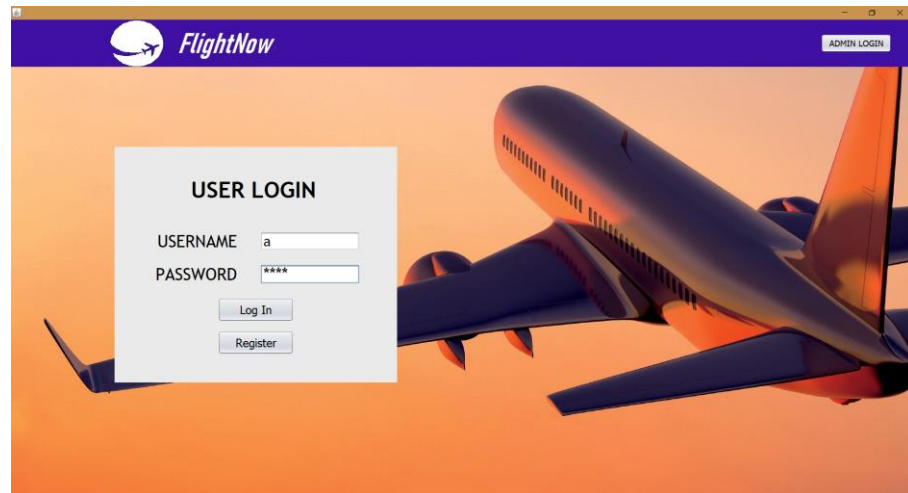


9. OUTPUT SCREENS (GUI)

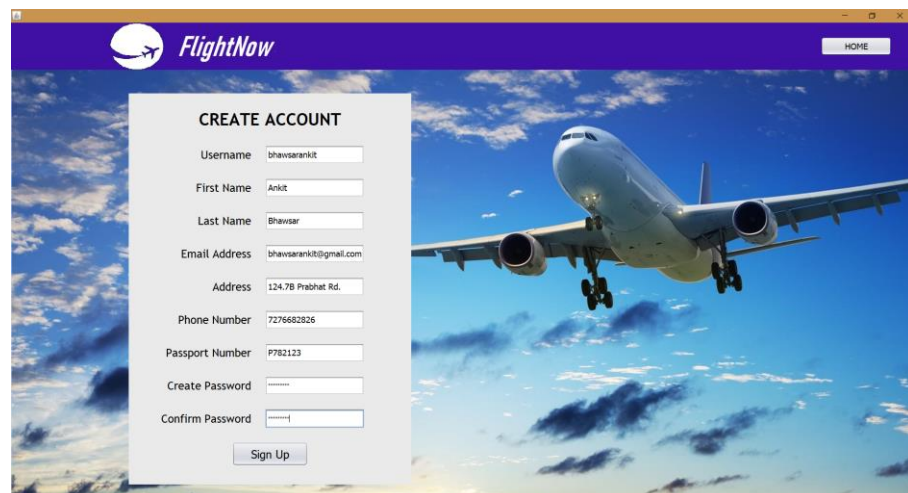
7.1 Customer Side GUI

User Home Page:

Provides Login for User

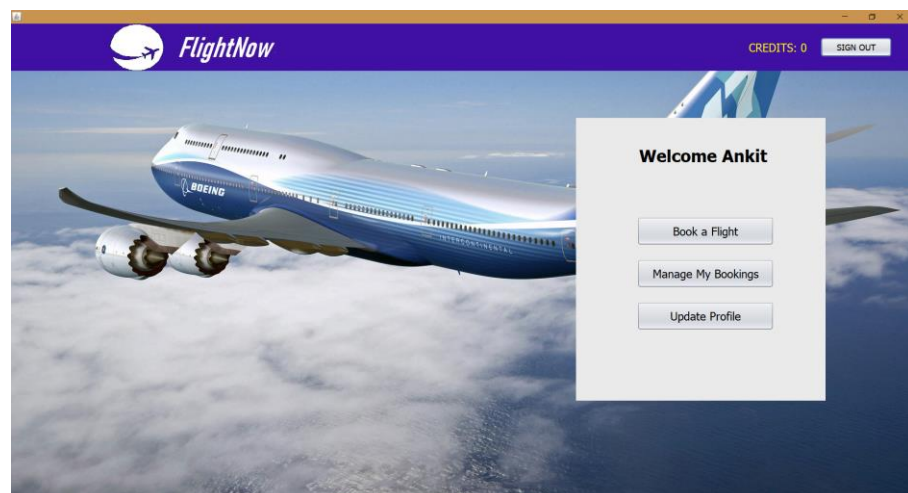


User Register Page



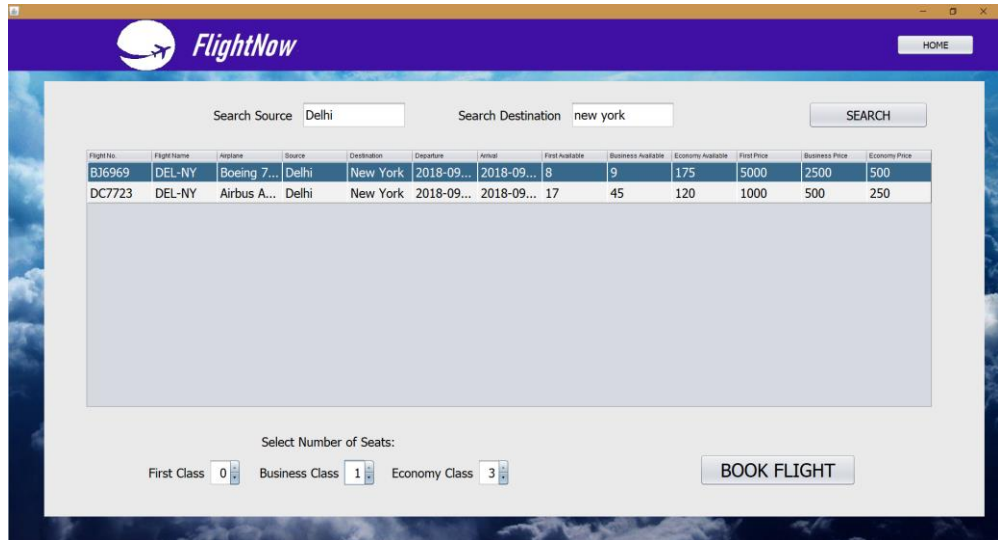
User Home Page:

Displays all functionalities to user



Ticket Booking Page:

Allows user to select and search for flights

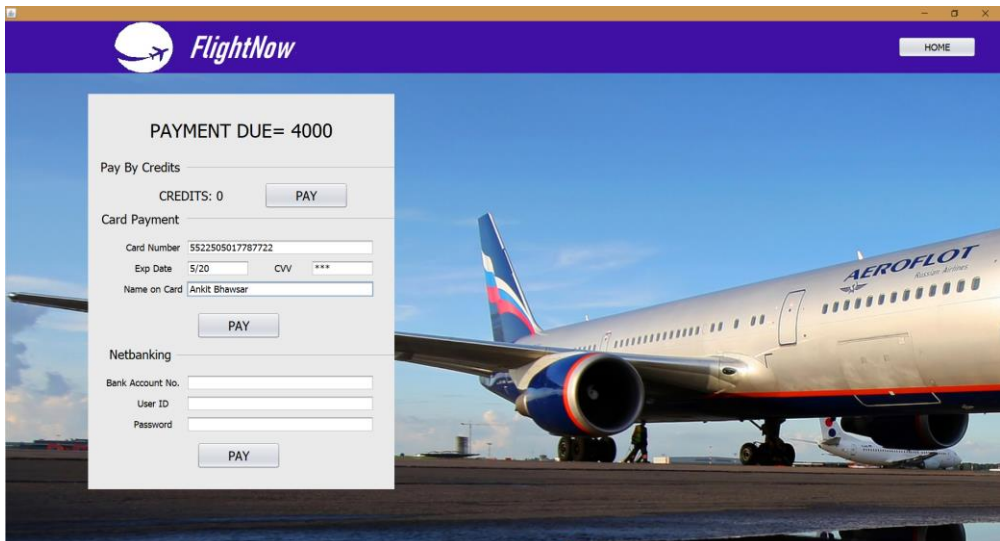


The screenshot shows the FlightNow website's ticket booking interface. At the top, there's a purple header with the FlightNow logo and a 'HOME' button. Below the header, there's a search section with 'Search Source' set to 'Delhi' and 'Search Destination' set to 'new york'. A 'SEARCH' button is to the right. Below the search section is a table with flight details. The table has columns for Flight No., Flight Name, Airplane, Source, Destination, Departure, Arrival, First Available, Business Available, Economy Available, First Price, Business Price, and Economy Price. Two flights are listed: B36969 (DEL-NY, Boeing 7..., Delhi, New York, 2018-09..., 2018-09..., 8, 9, 175, 5000, 2500, 500) and DC7723 (DEL-NY, Airbus A..., Delhi, New York, 2018-09..., 2018-09..., 17, 45, 120, 1000, 500, 250). Below the table is a 'Select Number of Seats:' section with three input fields: 'First Class' (0), 'Business Class' (1), and 'Economy Class' (3). A 'BOOK FLIGHT' button is at the bottom right.

Flight No.	Flight Name	Airplane	Source	Destination	Departure	Arrival	First Available	Business Available	Economy Available	First Price	Business Price	Economy Price
B36969	DEL-NY	Boeing 7...	Delhi	New York	2018-09...	2018-09...	8	9	175	5000	2500	500
DC7723	DEL-NY	Airbus A...	Delhi	New York	2018-09...	2018-09...	17	45	120	1000	500	250

Payment Page:

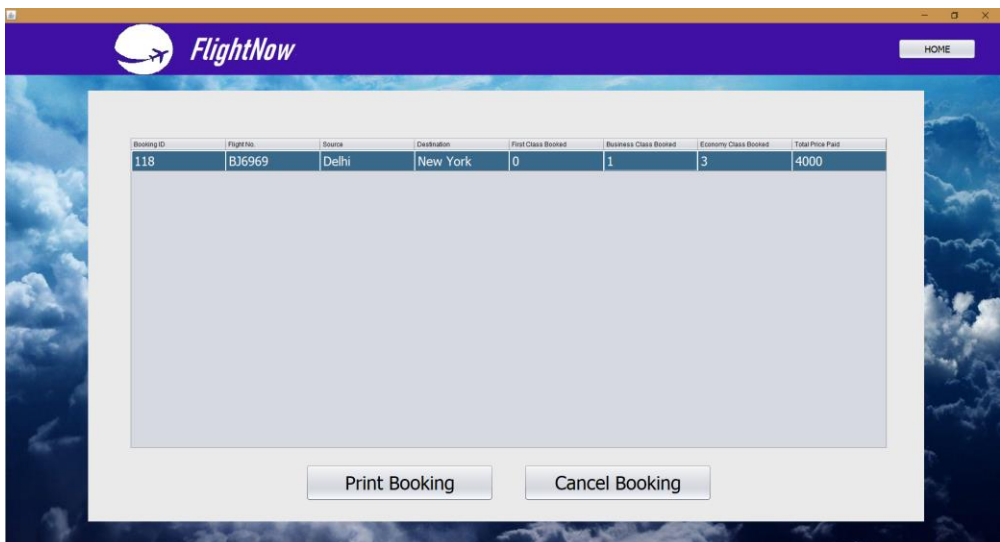
Final payment with various available techniques



The screenshot shows the FlightNow website's payment page. The background features a large image of an Aeroflot airplane on a runway. Overlaid on the left is a payment form. At the top of the form, it says 'PAYMENT DUE= 4000'. Below this, there are three payment methods: 'Pay By Credits' (with 'CREDITS: 0' and a 'PAY' button), 'Card Payment' (with fields for Card Number, Exp Date, CVV, and Name on Card, and a 'PAY' button), and 'Netbanking' (with fields for Bank Account No., User ID, and Password, and a 'PAY' button).

Manage Bookings:

Allows user to view, cancel or print tickets



The screenshot shows the FlightNow website's manage bookings page. The background features a large image of a cloudy sky. Overlaid on the left is a table with booking details. The table has columns for Booking ID, Flight No., Source, Destination, First Class Booked, Business Class Booked, Economy Class Booked, and Total Price Paid. One booking is listed: 118, B36969, Delhi, New York, 0, 1, 3, 4000. Below the table are two buttons: 'Print Booking' and 'Cancel Booking'.

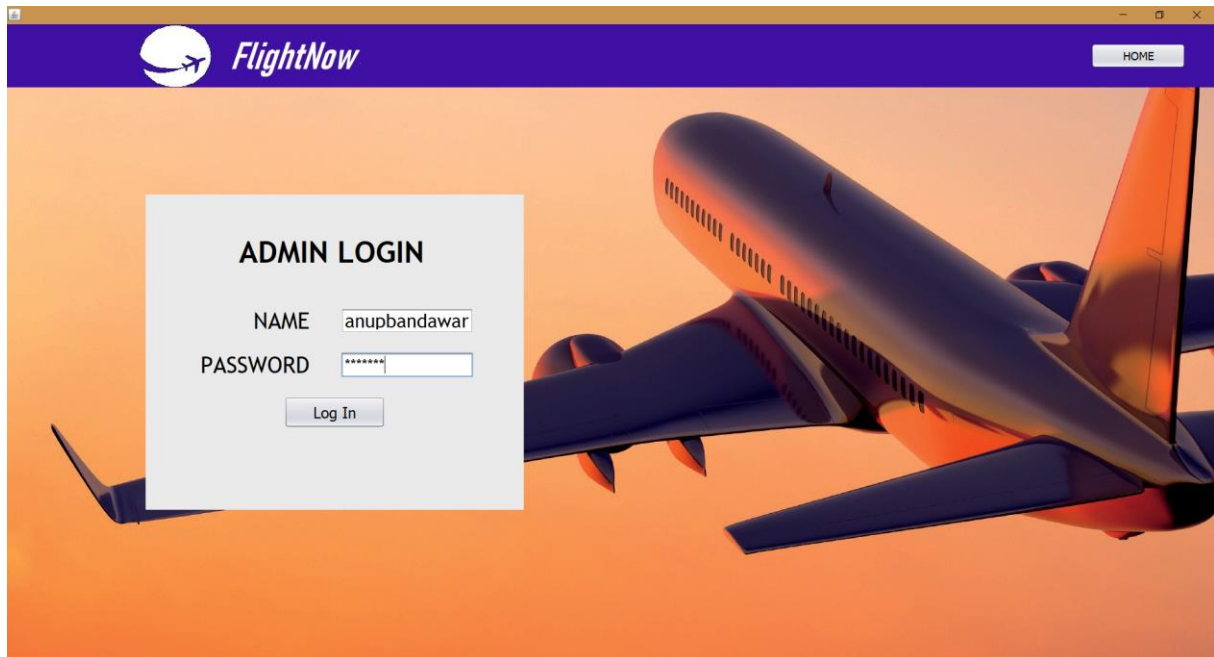
Booking ID	Flight No.	Source	Destination	First Class Booked	Business Class Booked	Economy Class Booked	Total Price Paid
118	B36969	Delhi	New York	0	1	3	4000

Ticket Generated:

PDF E-Ticket Generated
after Payment Process

BOOKING ID	118
FLIGHT NO.	BJ6969
SOURCE	Delhi
DESTINATION	New York
DEPARTURE	2018-09-26 00:00:00.0
ARRIVAL	2018-09-29 00:00:00.0
FIRST TICKETS BOOKED	0
BUSINESS TICKETS BOOKED	1
ECONOMY TICKETS BOOKED	3
TOTAL PRICE	4000
USERNAME	bhawsarankit
FIRST NAME	Ankit
LAST NAME	Bhawsar
EMAIL	bhawsarankit@gmail.com
ADDRESS	124/7B Prabhat Rd
PHONE NUMBER	7276682826
PASSPORT NUMBER	P7723662

7.2 Administrator Side GUI

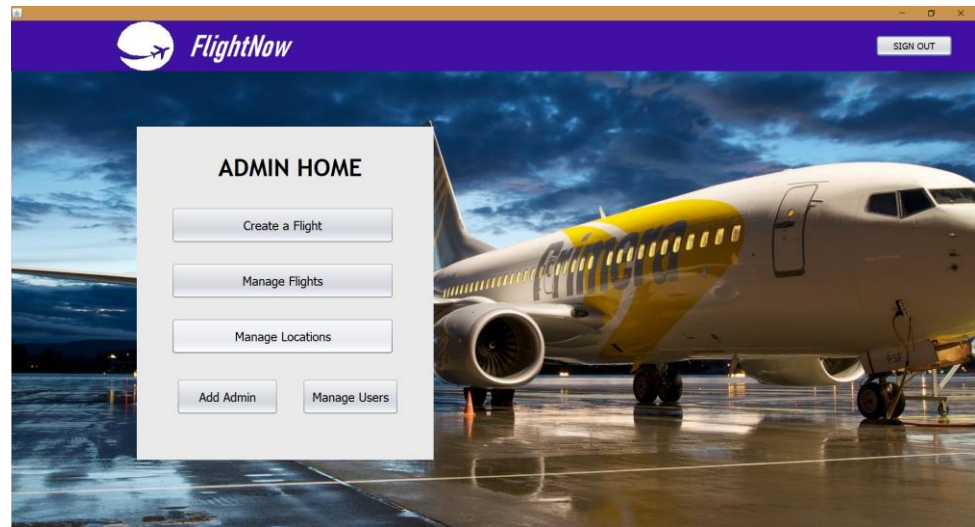


Admin Login Page:

Provides Login for
Administrators

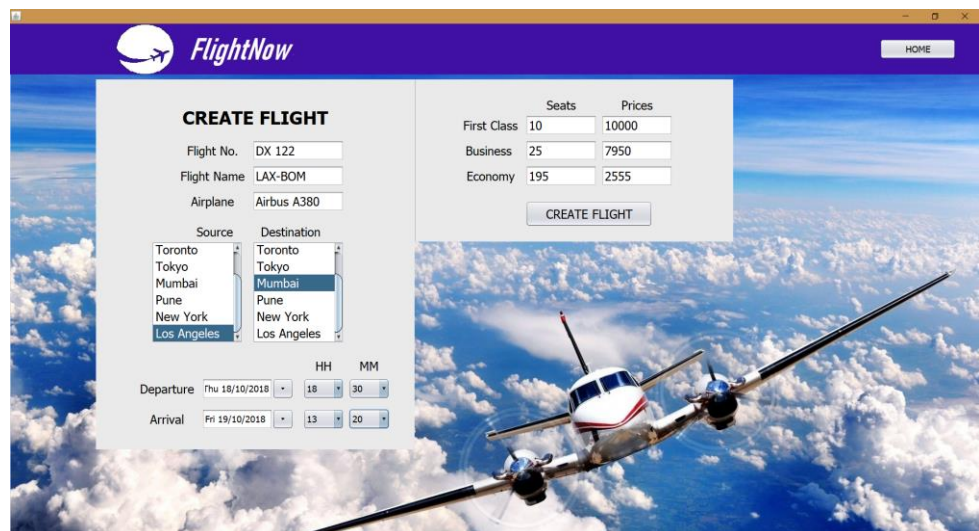
Admin Home Page:

Allows Admin to view all functionalities



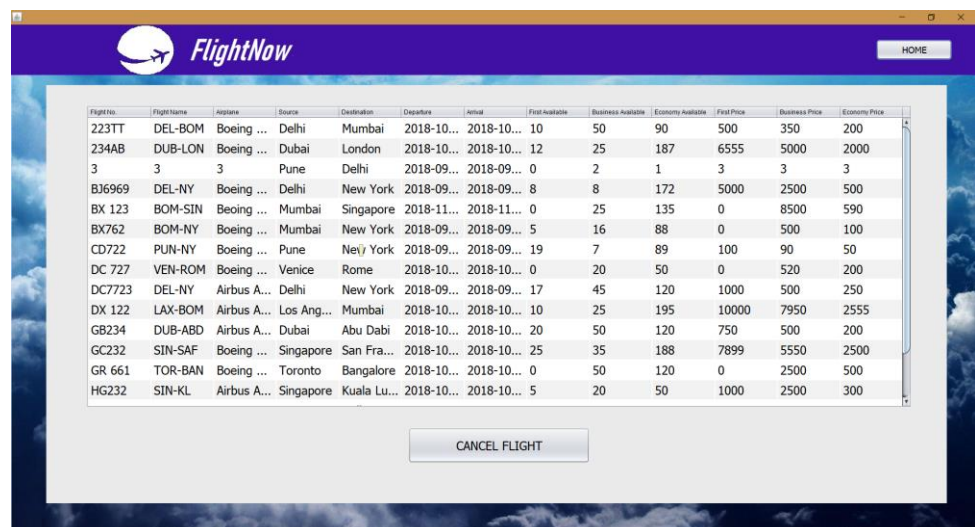
Create Flight Page:

Allows admins to create charter flights



Manage Flights Page:

Allows Admin to view, cancel any flights



Manage Locations:

Allows admin to add or remove locations serviced by flight service

FlightNow

HOME

Location ID: 223, Airport Name: HEA, City Name: London, No of Runways: 2

Add Location

Remove Location

Delhi, Bangalore, Toronto, Tokyo, Mumbai, Pune

Create Admin Page:

Allows an admin to add another admin

FlightNow

HOME

NEW ADMIN

NAME: omkar, PASSWORD: *****

ADD

Manage Users Page:

Allows admin to block, unblock or remove users.

FlightNow

HOME

ALL USERS: a, b, bhawsarankit, d, e, f, h, i, x

BLOCKED USERS: c, g

BLOCK USER, UNBLOCK USER, REMOVE USER

10. SAMPLE CODE

HomePage.java

Source Code for HomePage.java that provides login for user and options for admin login and user registration for new users:

```
package airlineproject;
import java.awt.Component;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
public class HomePage extends javax.swing.JFrame {
    public HomePage() {
        initComponents();
    }
    setExtendedState(java.awt.Frame.MAXIMIZED_BOTH);
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        username1 = new javax.swing.JTextField();
        user_password = new javax.swing.JPasswordField();
        userregisterbutton = new javax.swing.JButton();
        userloginbutton = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setMaximumSize(new java.awt.Dimension(1920, 1080));
        setMinimumSize(new java.awt.Dimension(1920, 1080));
        setPreferredSize(new java.awt.Dimension(1920, 1080));
        getContentPane().setLayout(null);

        jLabel4.setFont(new java.awt.Font("Trebuchet MS", 1, 48)); // NOI18N
        jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel4.setText("USER LOGIN");
```

```

getContentPane().add(jLabel4);
jLabel4.setBounds(330, 340, 370, 40);

jLabel5.setFont(new java.awt.Font("Trebuchet MS", 0, 36)); // NOI18N
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
jLabel5.setText("PASSWORD");
getContentPane().add(jLabel5);
jLabel5.setBounds(270, 520, 210, 40);

jLabel6.setFont(new java.awt.Font("Trebuchet MS", 0, 36)); // NOI18N
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
jLabel6.setText("USERNAME");
getContentPane().add(jLabel6);
jLabel6.setBounds(270, 450, 210, 40);

username1.setFont(new java.awt.Font("Trebuchet MS", 0, 30)); // NOI18N
getContentPane().add(username1);
username1.setBounds(530, 450, 210, 40);

user_password.setFont(new java.awt.Font("Trebuchet MS", 0, 36)); // NOI18N
getContentPane().add(user_password);
user_password.setBounds(530, 520, 210, 40);

userregisterbutton.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
userregisterbutton.setText("Register");
userregisterbutton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        userregisterbuttonActionPerformed(evt);
    }
});
getContentPane().add(userregisterbutton);
userregisterbutton.setBounds(440, 660, 160, 50);

userloginbutton.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
userloginbutton.setText("Log In");
userloginbutton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        userloginbuttonActionPerformed(evt);
    }
});
getContentPane().add(userloginbutton);
userloginbutton.setBounds(440, 590, 160, 50);

jLabel3.setBackground(new java.awt.Color(0, 51, 102));
jLabel3.setFont(new java.awt.Font("SimSun", 0, 13)); // NOI18N

```

```

        jLabel3.setIcon(new
javax.swing.ImageIcon("C:\\Users\\admin\\Documents\\NetBeansProjects\\AirlineProject\\Pictur
es\\Silver.jpg")); // NOI18N
        jLabel3.setText("jLabel3");
        getContentPane().add(jLabel3);
        jLabel3.setBounds(220, 270, 600, 500);

        jButton1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
        jButton1.setText("ADMIN LOGIN");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton1);
        jButton1.setBounds(1720, 30, 150, 40);

        jLabel2.setBackground(new java.awt.Color(0, 0, 102));
        jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel2.setIcon(new
javax.swing.ImageIcon("C:\\Users\\admin\\Documents\\NetBeansProjects\\AirlineProject\\Pictur
es\\Header.jpg")); // NOI18N
        jLabel2.setToolTipText("");
        jLabel2.setRequestFocusEnabled(false);
        getContentPane().add(jLabel2);
        jLabel2.setBounds(0, 0, 1920, 100);

        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setIcon(new
javax.swing.ImageIcon("C:\\Users\\admin\\Documents\\NetBeansProjects\\AirlineProject\\Pictur
es\\Wall1.jpg")); // NOI18N
        jLabel1.setMaximumSize(new java.awt.Dimension(1920, 1080));
        jLabel1.setMinimumSize(new java.awt.Dimension(1920, 1080));
        jLabel1.setPreferredSize(new java.awt.Dimension(1920, 1080));
        getContentPane().add(jLabel1);
        jLabel1.setBounds(0, 0, 1920, 1080);

        pack();
    } // </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        this.setVisible(false);
        new AdminLogin().setVisible(true);
    }

```



```

private void userregisterbuttonActionPerformed(java.awt.event.ActionEvent evt) {

    this.setVisible(false);
    new UserRegister().setVisible(true);
}

private void userloginbuttonActionPerformed(java.awt.event.ActionEvent evt) {

    String name=username1.getText();
    String pass=user_password.getText();

    try{

        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3307/airline_db","root","");

        String sql="Select * from user where username=? and user_password= ?;";

        PreparedStatement pst;
        pst =con.prepareStatement(sql);
        pst.setString(1,name);
        pst.setString(2,pass);
        ResultSet rs=pst.executeQuery();

        boolean found=false,found2=false;
        while(rs.next())
        {
            if(rs.getString("username").equalsIgnoreCase(name))
            {
                found=true;
                if(rs.getString("user_password").equalsIgnoreCase(pass))
                    found2=true;
            }
        }

        String query="select block_state from user where username=?";

        PreparedStatement pst1;
        pst1 =con.prepareStatement(query);
        pst1.setString(1,name);
        ResultSet rs1=pst1.executeQuery();
        rs1.beforeFirst();
    }
}

```

```

        if(found == true && found2 == true)
        {

            while(rs1.next())
            {
                if((rs1.getInt("block_state")==0))
                {
                    JOptionPane.showMessageDialog(this,"Successfully Logged In");
                    this.setVisible(false);

                    String makedeactive="update user set active_user=0";
                    PreparedStatement pst3;
                    pst3 =con.prepareStatement(makedeactive);
                    pst3.executeUpdate();

                    String makeactive="update user set active_user=1 where username=?";
                    PreparedStatement pst2;
                    pst2 =con.prepareStatement(makeactive);
                    pst2.setString(1,name);
                    pst2.executeUpdate();

                    new UserHome().setVisible(true);
                }
                else{
                    JOptionPane.showMessageDialog(this,"Sorry, you have been Blocked!\nContact
your Admin");
                }
            }

        }

        else if(found == false || found2 == false)
        {
            JOptionPane.showMessageDialog(this,"Incorrect Details");
        }

    }

    catch(Exception r){
        System.out.println(r);
    }
}

/**
 * @param args the command line arguments
 */

```

```

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    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) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

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

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;

```

```
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JPasswordField user_password;  
private javax.swing.JButton userloginbutton;  
private javax.swing.JTextField username1;  
private javax.swing.JButton userregisterbutton;  
// End of variables declaration  
  
void setvisible(boolean b) {  
    throw new UnsupportedOperationException("Not supported yet."); //To change body of  
generated methods, choose Tools | Templates.  
}  
  
private Component UserHome() {  
    throw new UnsupportedOperationException("Not supported yet."); //To change body of  
generated methods, choose Tools | Templates.  
}  
}
```

11. CONCLUSION

This project has been developed and deployed to try and provide a better alternative to traditional methods of airline reservation systems. Keeping in mind the drawbacks of the old system, we have developed an Airline Reservation System which will provide user-friendly service to customers and help the user have a seamless process of booking airline tickets.

12. REFERENCES

1. www.mysql.com
2. www.stackoverflow.com
3. www.extralectures.com
4. www.oracle.com