LOVELY PROFESSIONAL UNIVERSITY
PHAGWARA, PUNJAB

MINI PROJECT
On
AIRLINE RESERVATION SYSTEM

Student Name: Harsh Vidit	**Reg**: 12104689	**Roll no**. – 01
Student Name: Hridhanu Bose	**Reg**: 12104798	**Roll no**. – 03
Student Name: Hrutik Bachuwar	**Reg**: 12104720	**Roll no**. – 49

Section- K21GP
Course- B.Tech (Computer Science and Engineering)

Submitted to
Arvind Kumar
Asst. Professor

School of Computer Science and Engineering

Role and Responsibilities of each Student

Student Name: Harsh Vidit
Reg: 12104689

- Coding of the project
- Information gathering
- Github hosting and uploading

Student Name:Hridhanu Bose
Reg: 12104798

- Information gathering
- GUI blueprint
- Logo design

Student Name: Hrutik Bachuwar

- Information gathering
- GUI improvising
- Report creation
- Coding

INDEX

ABSTRACT

Airline reservation systems (**ARS**) are systems that allow an airline to sell their inventory (seats). It contains information on schedules and fares and contains a database of reservations (or passenger name records) and of tickets issued (if applicable). ARSs are part of passenger service systems (PSS), which are applications supporting the direct contact with the passenger.

ARS eventually evolved into the computer reservations system (CRS). A computer reservation system is used for the reservations of a particular airline and interfaces with a global distribution system (GDS) which supports travel agencies and other distribution channels in making reservations for most major airlines in a single system.

INTRODUCTION

Java Swing is a set of graphical user interface (GUI) widgets and tools included as part of the Java Standard Edition platform. It provides a way to create a variety of interactive applications that run on any Java Virtual Machine (JVM). Swing is designed to be platform-independent, which means that you can write once and run anywhere (WORA) code that will work on any platform that supports Java. Swing provides a rich set of GUI components, including buttons, labels, text boxes, text areas, menus, scrollbars, tables, trees, and more. It also provides a set of advanced components, such as JSpinner, JTreeTable, and JEditorPane.

The java.awt (Abstract Window Toolkit) package is a core part of the Java SE (Standard Edition) platform and provides a set of classes for building graphical user interface (GUI) applications in Java. It includes classes for handling windows, graphics, fonts, images, and user input events. The main purpose of the AWT library is to provide a set of GUI components, such as buttons, menus, text fields, and panels, that can be combined together to create complex GUI applications.

The key classes in java.awt package are:

1. Component - It is the base class for all the graphical components in AWT. It provides the basic functionality for rendering and handling user input events for the graphical components.

2. Container - It is a subclass of the Component class and is used to group together multiple graphical components.
3. LayoutManager - It is an interface that provides a way to arrange the graphical components within a container.

4. Graphics - It is a class that provides the basic drawing operations for displaying the graphical components.
5. Font - It is a class that represents the font used for rendering the text on the graphical components.
6. Color - It is a class that represents the color used for rendering the graphical components.

The key event classes in AWT are:
1. ActionEvent - It is generated when a user performs an action, such as clicking a button or selecting a menu item.
2. MouseEvent - It is generated when a user performs a mouse action, such as clicking a mouse button or moving the mouse.
3. KeyEvent - It is generated when a user presses or releases a key on the keyboard.

Overall, the java.awt package provides a rich set of classes and interfaces for building GUI applications in Java. However, it has been largely superseded by the more advanced Swing library, which is built on top of AWT and provides a more powerful set of GUI components and features.
//////
The source code using these packages leads to the following working of the game --
A player can choose between two symbols with his opponent, usual games use “X”and “O”. If first player choose “X” then the second player have to play with “O” and vice versa. A player marks any of

the 3x3 squares with his symbol (may be “X” or “O”) and his aim is to create a straight line horizontally or vertically or diagonally with two intensions: a) Create a straight line before his opponent to win the game. b) Restrict his opponent from creating a straight line first. In case logically no one can create a straight line with his own symbol, the game results a tie. Hence there are only three possible results – a player wins, his opponent (human or computer) wins or it’s a tie. /////

PREVIOUS SYSTEM

Long ago, before internet came into scene, to buy a ticket or to reserve a ticket we had to go to ticket counter in airport or any other reservation counter. In this type of situation we had to go there personally in advance to get the work done.
Reserving tickets in such a fashion consumed a lot of time and energy. Long queues made it very difficult for people to reserve their seats.

PROPOSED SYSTEM

The existing system makes use of internet to reserve seats. Server-Client applications are designed and implemented to ease the process of booking reducing the overhead of customers. Many websites such as MakeMyTrip and GoIbibo use attractive GUI and flawless transactions to make the user experience seamless. Bookings are done in the following manner.The user is required to login to the website or application , followed by selection of mode

of transport(in our case flights), date, time , one-way or roundtrip etc.

After taking all the inputs into consideration the available vehicles(in our case flights) are shown to the user to select from. Fare may vary from company to company depending upon number of factors such as availability and luxury. The user has complete freedom to choose whichever mode suits him/her. Upon selection of the mode of transport, the user is directed to the payment window , where he/she is required to pay the outstanding fee. Finally after payment the booking is successfully done and a receipt in the form of text message is sent to the customer and an online ticket is made available in the 'user account' of the application.

System Requirements for Airline Reservation Java project are minimal and can be easily met by most modern computers.

Hardware Requirements:

- Processor: Intel Pentium 4 or equivalent or higher
- RAM: At least 1 GB (recommended 2 GB or higher)
- Hard Disk Space: At least 100 MB of free disk space
- Display: Monitor capable of displaying a minimum resolution of 1024 x 768 pixels

Software Requirements:

- Java Development Kit (JDK) version 8 or higher

- Integrated Development Environment (IDE) such as Eclipse
- Operating System: Windows, Linux, or macOS

It is essential to have a clear understanding of the project requirements and specifications to avoid any issues during development.

SOURCE CODE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLOutput;

public class Airline extends JDBC
{
    JButton Submit,Cancel,Confirm;
     String fname,lname,date;
    String fname_innerframe,lname_innerframe,mobNo_innerframe;
    String given_age,mob_number;
    Object
source,destination,gender,flying_class,flights_avail,time_in_hours,time_in_minutes
;

    public static void main(String args[])
    {

        String[] sources =
{"Amritsar","Bangalore","Chennai","Delhi","Goa","Hyderabad","Indore","Jaipur","Kol
kata","Lucknow","Mumbai","Nagpur","Patna","Srinagar","Trivandrum","Udaipur","Vizag
```

```
","Aizawal","Bhopal","Ahmedabad","Gaya"};
        String[]
destination={"Amritsar","Bangalore","Chennai","Delhi","Goa","Hyderabad","Indore","
Jaipur","Kolkata","Lucknow","Mumbai","Nagpur","Patna","Srinagar","Trivandrum","Uda
ipur","Vizag","Aizawal","Bhopal","Ahmedabad","Gaya"};

        ImageIcon image = new ImageIcon("D:\\Java
programming\\Programs\\Mcq_question\\src\\logo.jpeg");

        JLabel firstname_label = new JLabel();
        firstname_label.setText("First name:");
        firstname_label.setBounds(0,0,80,20);
        JLabel lastname_label = new JLabel("Last name:");
       lastname_label.setBounds(0,20,80,40);
       JLabel age_label = new JLabel("Age:");
       age_label.setBounds(0,60,80,40);
       JLabel gender_label = new JLabel("Gender:");
       gender_label.setBounds(0,100,80,40);
       JLabel mobilenumber_label= new JLabel("Mobile Number:");
       mobilenumber_label.setBounds(0,140,90,40);
       JLabel from_label = new JLabel("From:");
       from_label.setBounds(300,60,80,40);
       JLabel to_label =new JLabel("TO:");
       to_label.setBounds(500,60,80,40);
       JLabel class_s = new JLabel("Class:");
       class_s.setBounds(665,60,80,40);
       JLabel cancellation_label = new JLabel("Press on 'Cancel Booking' for
cancellation of ticket");
       cancellation_label.setBounds(0,250,290,40);
       JLabel nameof_flights = new JLabel("Flights available:");
       nameof_flights.setBounds(850,60,100,40);
       JLabel date = new JLabel("Date:");
       date.setBounds(500,148,40,30);
       JLabel Time_hours = new JLabel("Time(in hours):");
       Time_hours.setBounds(763,141,100,40);
       JLabel Time_minutes = new JLabel("Time(in minutes):");
       Time_minutes.setBounds(980,141,100,40);

        JTextField first_name = new JTextField();
        first_name.setPreferredSize(new Dimension(200,22));
        first_name.setFont(new Font("Consolas",Font.PLAIN,16));
        first_name.setBounds(80,5,200,22);
        JTextField last_name= new JTextField();
        last_name.setPreferredSize(new Dimension(200,22));
        last_name.setFont(new Font("Consolas", Font.PLAIN,16));
        last_name.setBounds(80,32,200,22);
        JTextField age = new JTextField();
        age.setPreferredSize(new Dimension(200,22));
        age.setFont(new Font("Consolas",Font.PLAIN,16));
        age.setBounds(80,70,200,22);
        JTextField mobile_number = new JTextField();
       mobile_number.setPreferredSize(new Dimension(200,22));
        mobile_number.setFont(new Font("Consolas",Font.PLAIN,16));
        mobile_number.setBounds(92,152,200,22);
        JTextField date_field = new JTextField();
        date_field.setFont(new Font("Consolas",Font.PLAIN,16));
        date_field.setPreferredSize(new Dimension(200,22));
        date_field.setBounds(540,152,200,22);

        JComboBox gender_select = new JComboBox();
```

```
        gender_select.addItem("Male");
        gender_select.addItem("Female");
        gender_select.addItem("Other");
        gender_select.setBounds(80,110,80,20);
        JComboBox from_box = new JComboBox();
        for(int i=0;i< sources.length;i++)
        {
            from_box.addItem(sources[i]);
        }
        from_box.setBounds(340,71,120,20);
        JComboBox to_box= new JComboBox();
        for(int i=0;i<destination.length;i++)
        {
            to_box.addItem(destination[i]);
        }
        to_box.setBounds(530,71,120,20);
        JComboBox class_s_box= new JComboBox();
        class_s_box.addItem("Economy");
        class_s_box.addItem("Business");
        class_s_box.addItem("First");
        class_s_box.setBounds(710,71,120,20);
        JComboBox name_offlights_box = new JComboBox();
        name_offlights_box.addItem("Air India");
       name_offlights_box.addItem("Spice Jet");
       name_offlights_box.addItem("Vistara");
       name_offlights_box.addItem("Indigo");
       name_offlights_box.addItem("Air Aisa");
       name_offlights_box.setBounds(955,70,100,20);
       JComboBox hours_box = new JComboBox();
       for(int i=0;i<24;i++)
       {
           hours_box.addItem(i);
       }
       hours_box.setBounds(860,152,100,20);
       JComboBox minutes_box = new JComboBox();
       for(int i=0;i<60;i++)
       {
           minutes_box.addItem(i);
       }
       minutes_box.setBounds(1090,152,100,20);



        JFrame frame= new JFrame("Airline Reservation System");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setIconImage(image.getImage());
        frame.setSize(1500,600);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.add(firstname_label);
        frame.add(lastname_label);
        frame.add(Time_hours);
        frame.add(Time_minutes);
        frame.add(age_label);
        frame.add(gender_label);
        frame.add(mobilenumber_label);
        frame.add(from_label);
        frame.add(to_label);
        frame.add(cancellation_label);
        frame.add(class_s);
```

```
        frame.add(nameof_flights);
        frame.add(date);
        frame.add(first_name);
        frame.add(last_name);
        frame.add(age);
        frame.add(gender_select);
        frame.add(mobile_number);
        frame.add(date_field);
        frame.add(from_box);
        frame.add(to_box);
        frame.add(class_s_box);
        frame.add(name_offlights_box);
        frame.add(hours_box);
        frame.add(minutes_box);


        Airline obj=new Airline();
        obj.Submit = new JButton("Confirm Booking");
        obj.Submit.setFocusable(false);
        obj.Submit.setBounds(320,200,200,30);
        obj.Submit.setFont(new Font("Consolas",Font.BOLD,16));
        obj.Submit.addActionListener(e -> {obj.fname=first_name.getText();
            obj.lname=last_name.getText();
            obj.given_age = age.getText();
            obj.mob_number= mobile_number.getText();
           obj.gender=gender_select.getItemAt(gender_select.getSelectedIndex());
           obj.source=from_box.getItemAt(from_box.getSelectedIndex());
           obj.destination=to_box.getItemAt(to_box.getSelectedIndex());
           obj.flying_class=class_s_box.getItemAt(class_s_box.getSelectedIndex());

obj.flights_avail=name_offlights_box.getItemAt(name_offlights_box.getSelectedIndex
());
           obj.date=date_field.getText();
           obj.time_in_hours=hours_box.getItemAt(hours_box.getSelectedIndex());

obj.time_in_minutes=minutes_box.getItemAt(minutes_box.getSelectedIndex());
            JDBC insert1 = new JDBC();
            insert1.insertrow(obj.fname, obj.lname, obj.given_age,
obj.mob_number,obj.gender,obj.source,obj.destination,obj.flying_class,obj.flights_
avail,obj.date,obj.time_in_hours,obj.time_in_minutes);
            System.out.println("A ticket for "+obj.source+" to "+obj.destination+"
has successfully been booked by you.Kindly enjoy your journey");
            JLabel thank_label = new JLabel("Booking done successfully");
            thank_label.setBounds(0,0,480,100);
            thank_label.setFont(new Font("Consolas",Font.BOLD,30));
            JFrame thank_frame= new JFrame();
            thank_frame.setSize(480,100);
            thank_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            thank_frame.add(thank_label);
            thank_frame.setVisible(true);
            thank_frame.setLayout(null);
        });

        obj.Cancel = new JButton("Cancel Booking");
        obj.Cancel.setFocusable(false);
        obj.Cancel.setBounds(320,290,200,30);
        obj.Cancel.setFont(new Font("Consolas",Font.BOLD,16));
        obj.Cancel.addActionListener(e->{
```

```
            JLabel firstname_label2 = new JLabel();
            firstname_label2.setText("First name:");
            firstname_label2.setBounds(0,0,80,20);
            JLabel lastname_label2 = new JLabel("Last name:");
            lastname_label2.setBounds(0,20,80,40);
            JLabel mobilenumber_label2= new JLabel("Mobile Number:");
            mobilenumber_label2.setBounds(0,60,90,40);

            JTextField first_name2 = new JTextField();
            first_name2.setPreferredSize(new Dimension(200,22));
            first_name2.setFont(new Font("Consolas",Font.PLAIN,16));
            first_name2.setBounds(80,5,200,22);
            JTextField last_name2= new JTextField();
            last_name2.setPreferredSize(new Dimension(200,22));
            last_name2.setFont(new Font("Consolas", Font.PLAIN,16));
            last_name2.setBounds(80,32,200,22);
            JTextField mobile_number2 = new JTextField();
            mobile_number2.setPreferredSize(new Dimension(200,22));
            mobile_number2.setFont(new Font("Consolas",Font.PLAIN,16));
            mobile_number2.setBounds(100,72,200,22);


            ImageIcon image2 = new ImageIcon("D:\\Java
programming\\Programs\\Mcq_question\\src\\logo.jpeg");
           JFrame frame2 = new JFrame();
           frame2.setIconImage(image2.getImage());
           frame2.setSize(500,500);
           frame2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
           frame2.setVisible(true);
           frame2.setLayout(null);

            obj.Confirm = new JButton("Confirm");
            obj.Confirm.setFocusable(false);
            obj.Confirm.setFont(new Font("Consolas",Font.PLAIN,16));
            obj.Confirm.setBounds(170,150,100,30);
            obj.Confirm.addActionListener(e2->{

                obj.fname_innerframe=first_name2.getText();
                obj.lname_innerframe=last_name2.getText();
                obj.mobNo_innerframe=mobile_number2.getText();
                JDBC delete1 = new JDBC();

delete1.deleterow(obj.fname_innerframe,obj.lname_innerframe,obj.mobNo_innerframe);
            });

           frame2.add(firstname_label2);
           frame2.add(lastname_label2);
           frame2.add(mobilenumber_label2);
           frame2.add(first_name2);
           frame2.add(last_name2);
           frame2.add(mobile_number2);
           frame2.add(obj.Confirm);
        });

        frame.add(obj.Submit);
        frame.add(obj.Cancel);
    }

}
```

Logic of the Code

We have defined ‘Airline’ as the main class which extend the JDBC class defined in the same package(created by us). Inside the class non-static members are declared named as Submit,Cancel,Confirm, fname, lname, given_age, date etc. Inside the main function all the Jframe,Jlabels, Jtextfield and Jcombobox are declared and initialised. All the properties of the Jframe are set according to the need.For eg, frame.setVisible(true) – to make the frame visible ; frame.setLayout(null) to disable the default layout manager; frame.setSize() to describe the size of the frame.

Bounds of all the components of the Jframe are set manually, in order to maintain a specific design(designed by us).The values in some comboboxes are set manually, whereas other combo boxes use loop to iterate and add the elements in them.To add the elements in boxes addItem() method is used.

All the components i.e JLable,JTextField and Jcombobox are added to the frame using add() method.

The button Submit(confirm booking) is declared outside the main function and is accessed using the Object(obj) of Airline class. The button is initialised and the ActionListener is set for the the button using addActionListener() method and by further overriding the void actionPerformed() method, defined in the ActionListener interface. This addActionListener() method will make our button functional. In the overridden method we are extracting all the information provided by the user in the text field and after getting all the string values we create insert1

object of JDBC class(defined in the same package by us). And using this object we call the insertrow method of JDBC class.

Similarly the Cancel Booking button works. We create another frame whenever the Cancel Booking button is hit by the user.And the user is prompted to provide the First name, Last name and Mobile number.

When the Confirm button is hit. The input from the user is fed to the deleterow() method of the JDBC class. The method is called using delete1 object, defined in the actionlistener of the Confirm button.

SOURCE CODE

```
import com.sun.source.tree.TryTree;

import java.util.*;
import java.sql.*;
class JDBC
{
    void insertrow(String a,String b,String c,String d,Object e,Object f,Object
g,Object h,Object i,String j,Object k,Object l)
    {
        try {

            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/airline", "root",
"Biotech@5678");
            String q = "insert into Passenger values(?,?,?,?,?,?,?,?,?,?,?,?)";
            PreparedStatement pstmt = con.prepareStatement(q);
            pstmt.setString(1,a);
            pstmt.setString(2,b);
            pstmt.setString(3,c);
            pstmt.setString(4,d);
            pstmt.setObject(5,e);
            pstmt.setObject(6,f);
            pstmt.setObject(7,g);
            pstmt.setObject(8,h);
            pstmt.setObject(9,i);
            pstmt.setString(10,j);
            pstmt.setObject(11,k);
            pstmt.setObject(12,l);
            pstmt.executeUpdate();
            con.close();

        } catch (Throwable m) {
```

```
            System.out.println(m);
        }
    }
    void deleterow(String a,String b,String c)
    {
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/airline", "root",
"Biotech@5678");
            Statement stmt =con.createStatement();
            stmt.executeUpdate("delete from Passenger where
Passenger_fname='"+a+"' and Passenger_lname='"+b+"' and Passenger_mobNo='"+c+"'");
            System.out.println("Booking successfully cancelled.");
            con.close();
        }catch (Throwable d)
        {
            System.out.println(d);
        }
    }
    public static void main(String arv[])
    {

    }
}
```

Logic of the code

This JDBC implements two methods named as insertrow() and deleterow() to provide the application with database functionalties. One function inserts the rows in the database, whereas the other deletes the rows from the database.

Firstly we need to load the jdbc driver for that we use the forName(path) method.After loading the driver we need to establish the connection , the same is done using the getConnection() method of DriverManger.This methods return Connection type object and hence we use con, a connection referenced variable to the store the object.

After connection establishment we are ready to fire the queries.Different datatypes are used for different queries i.e Statement for queries that return Resultset , PreparedStatement

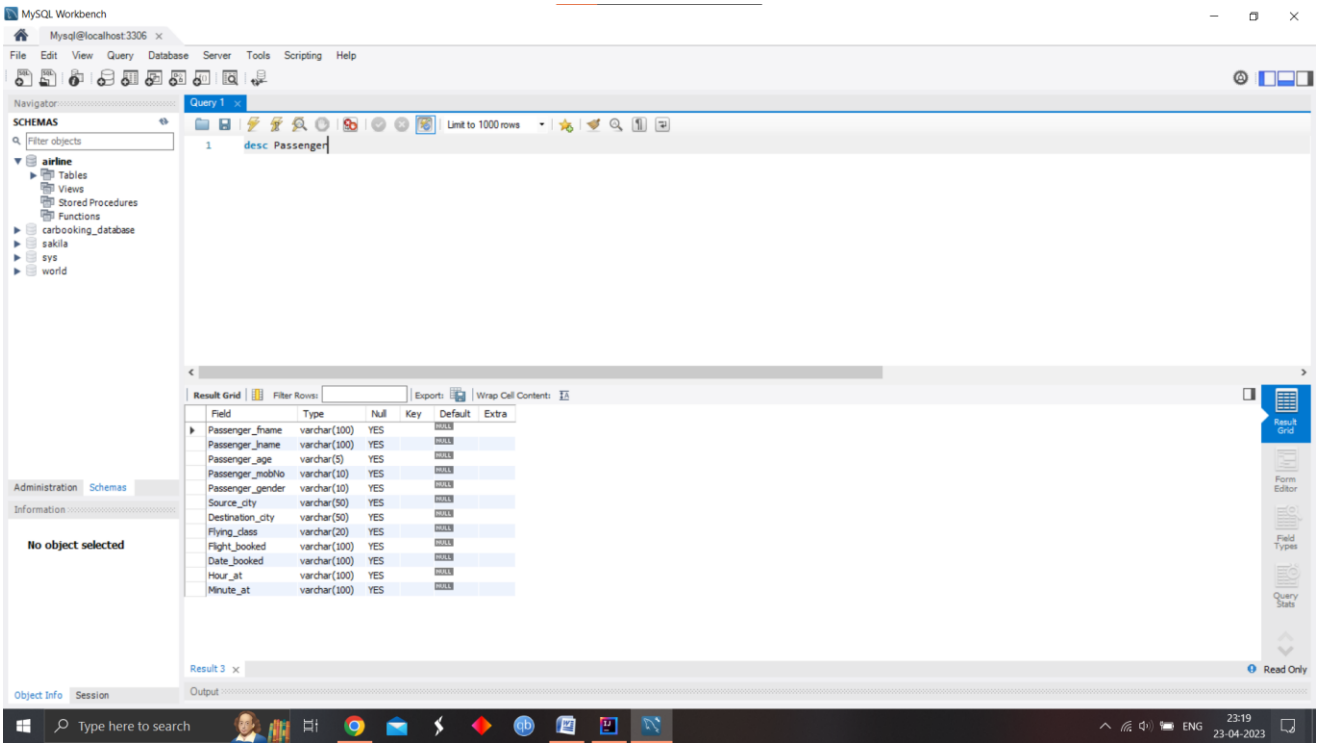
for queries like insert which return int and CallableStatement for methods and procedures.

Respective logic is implemented in the insertrow() and deleterow() methods depending upon the task assigned to them.

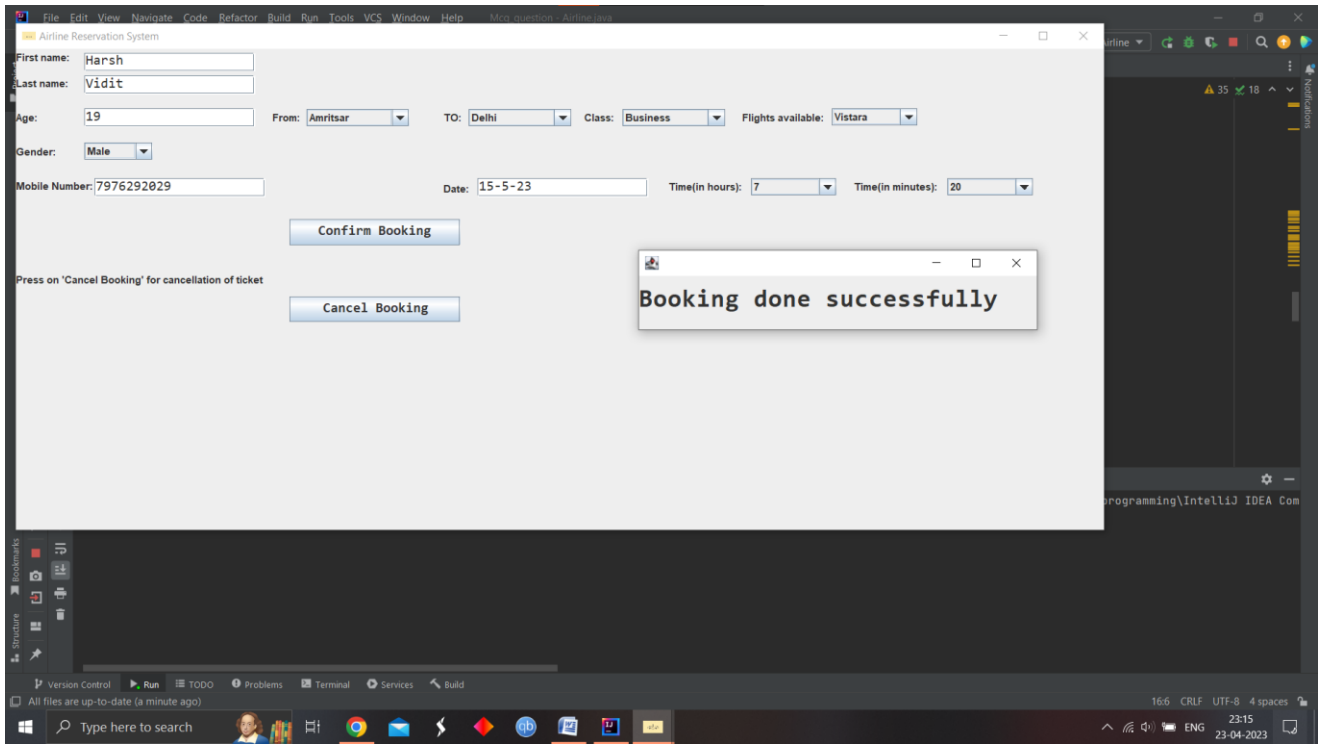
Database Used

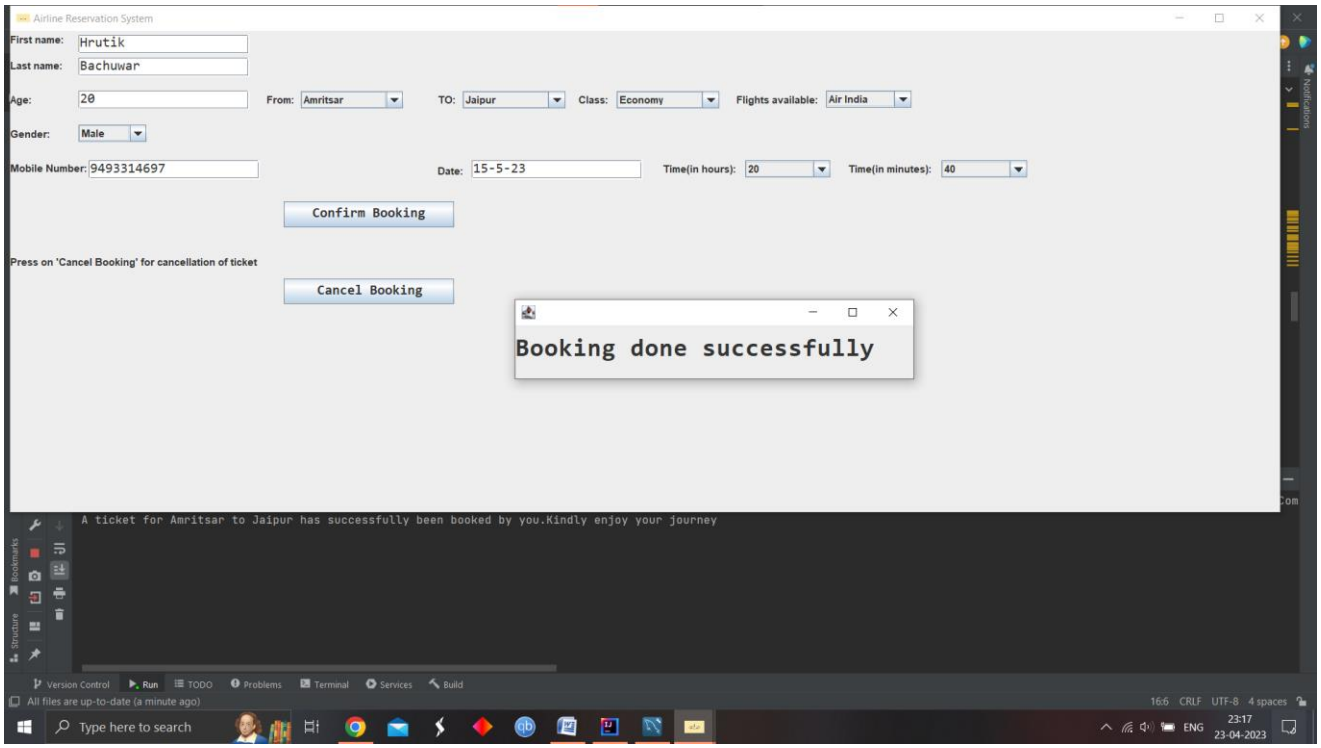
The DBMS that we have used in this project is Mysql and the name of the database is airline. Inside the database the table Passenger is used to store and retrieve the information.

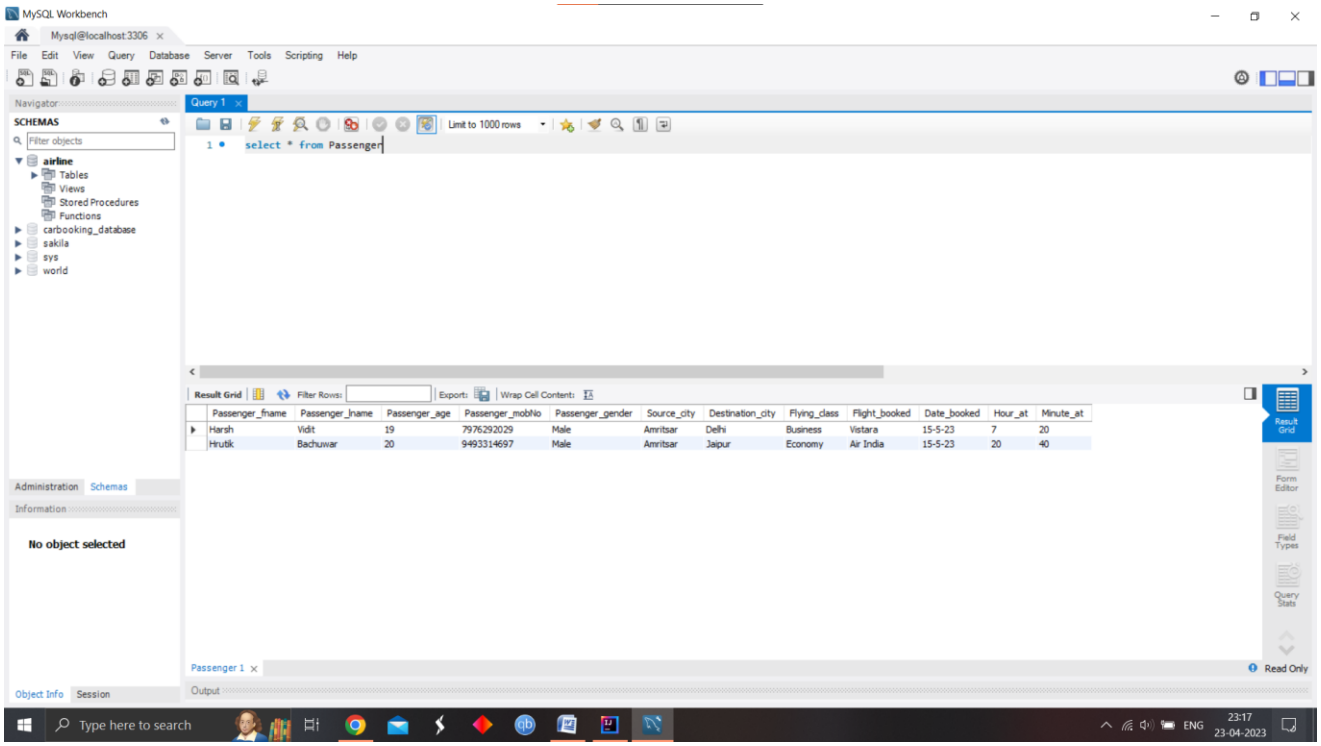
Following depicts the schema of the table—

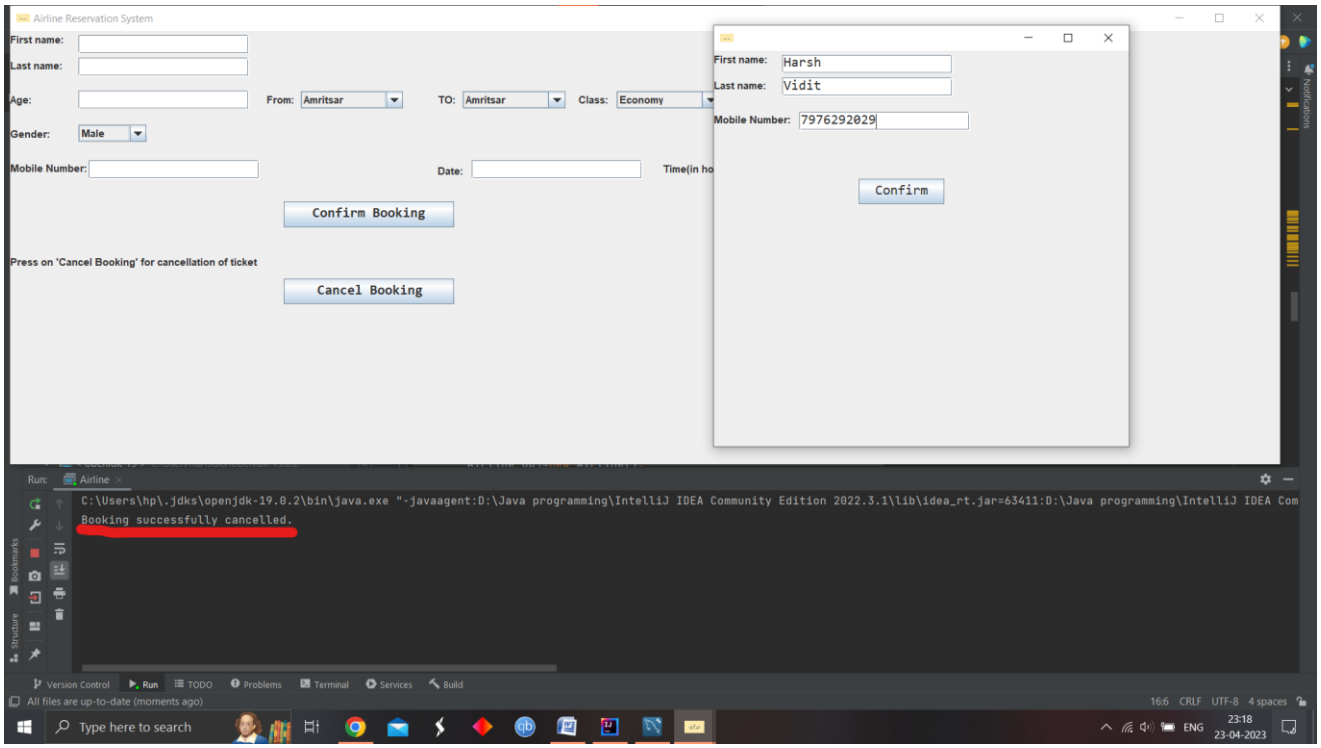


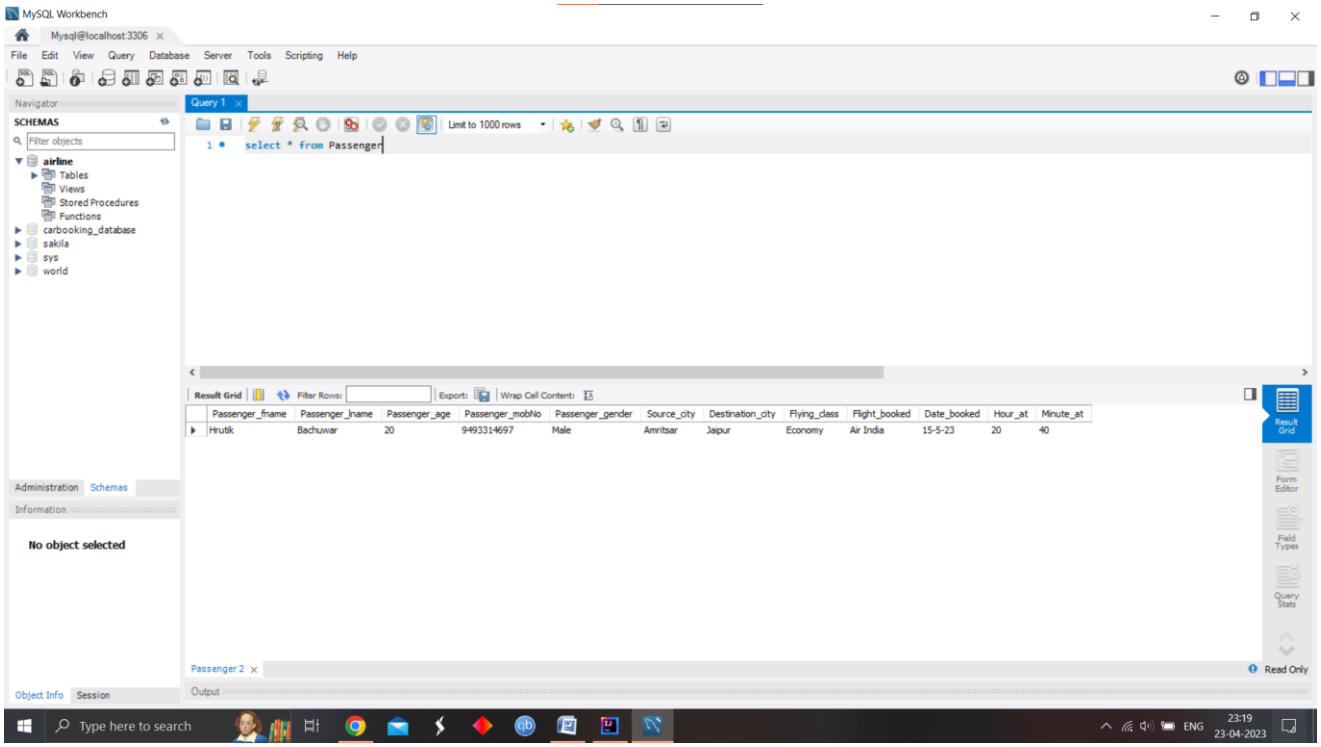
SNAPSHOTS


Airline Reservation System
First name: Harsh
Last name: Vidit
Age: 19
From: Amritsar
TO: Delhi
Class: Business
Flights available: Vistara
Gender: Male
Mobile Number: 7976292029
Date: 15-5-23
Time(in hours): 7
Time(in minutes): 20
Confirm Booking
Press on 'Cancel Booking' for cancellation of ticket
Cancel Booking
Booking done successfully


Airline Reservation System
First name: Hrutik
Last name: Bachuwar
Age: 20
From: Amritsar
TO: Jaipur
Class: Economy
Flights available: Air India
Gender: Male
Mobile Number: 9493314697
Date: 15-5-23
Time(in hours): 20
Time(in minutes): 40
Confirm Booking
Press on 'Cancel Booking' for cancellation of ticket
Cancel Booking
Booking done successfully
A ticket for Amritsar to Jaipur has successfully been booked by you.Kindly enjoy your journey


MySQL Workbench
Mysql@localhost:3306
Query 1
select * from Passenger
SCHEMAS
airline
carbooking_database
sakila
sys
world
No object selected
Passenger_fname Passenger_lname Passenger_age Passenger_mobNo Passenger_gender Source_city Destination_city Flying_class Flight_booked Date_booked Hour_at Minute_at
Harsh Vidit 19 7976292029 Male Amritsar Delhi Business Vistara 15-5-23 7 20
Hrutik Bachuwar 20 9493314697 Male Amritsar Jaipur Economy Air India 15-5-23 20 40


Airline Reservation System
First name:
Last name:
Age:
From: Amritsar
TO: Amritsar
Class: Economy
Gender: Male
Mobile Number:
Date:
Confirm Booking
Press on 'Cancel Booking' for cancellation of ticket
Cancel Booking
First name: Harsh
Last name: Vidit
Mobile Number: 7976292029
Confirm
Run: Airline
C:\Users\hp\.jdks\openjdk-19.0.2\bin\java.exe "-javaagent:D:\Java programming\IntelliJ IDEA Community Edition 2022.3.1\lib\idea_rt.jar=63411:D:\Java programming\IntelliJ IDEA Com
Booking successfully cancelled.
Version Control
Run
TODO
Problems
Terminal
Services
Build
All files are up-to-date (moments ago)
Type here to search
ENG
23:18
23-04-2023


MySQL Workbench
Mysql@localhost:3306
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
airline
Tables
Views
Stored Procedures
Functions
carbooking_database
sakila
sys
world
Query 1
Limit to 1000 rows
1 select * from Passenger
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Passenger_fname Passenger_lname Passenger_age Passenger_mobNo Passenger_gender Source_city Destination_city Flying_class Flight_booked Date_booked Hour_at Minute_at
Hrutik Bachuwar 20 9493314697 Male Amritsar Jaipur Economy Air India 15-5-23 20 40
Administration Schemas
Information
No object selected
Object Info Session
Passenger 2
Read Only
Output
Type here to search
ENG
23:19
23-04-2023

Github link

- https://github.com/HarshVidit/Airline-Management-System.git

REFERENCES

- Youtube channel – Brocode(Java swing playlist)
https://www.youtube.com/watch?v=Kmgo00avvEw&list=WL&index=2&ab_channel=BroCode

- Youtube channel – Code with Durgesh(JDBC playlist)
https://www.youtube.com/watch?v=SEPSc-SOV0o&list=WL&index=1&ab_channel=LearnCodeWithDurgesh

- Website -- Javatpoint