REPORT TYPE : MINI PROJECT REPORT

TITLE OF THE PROJECT : TRAVELLER'S STOP

NAME : NIKHIL SUMESH BABU NAMBIAR

USN : 1NH17CS150

SEMESTER : SIXTH (VI)

SECTION : B

CHAPTER 1

INTRODUCTION

Java is a general-purpose language that is object oriented and is designed to have few implementation dependencies. It is a Write-Once-Run-Anywhere language that is platform independent meaning Java programs can be written on one OS and run on a different one. This is possible because of Java bytecode that can run on any JVM (Java Virtual Machine).

MySQL is an open-source relational database management system developed by MySQL AB, MySQL can be connected to Java using MySQL connector and can be used in Java using the JDBC.

This project has been done using JAVA and MySQL. This project focuses on implementing a Graphical User Interface application on Java and connecting it to a backend database that allows for dynamic change of data on the Graphical User Interface.

1.1 PROBLEM DEFINITION:

➢ The following mini project helps the user to plan his/her entire journey. The user is allowed to book his/her tickets i.e., via plane or train.

➢ They have been given facilities to select among list of hotels with the price mentioned with it. The user can select the hotel he/she wants to live in.

➢ This mini project makes the life easier of those people whose main agenda of travelling is for tourism or vacationing. These people have been given the freedom to plan their whole tour.

➢ They have been given options to choose the mode of transport they want to prefer for touring. In this way, the user will be allowed to plan his itinerary accordingly.

1.2 <u>COURSE OBJECTIVES:</u>

- ➢ This mini project has been basically programmed using JAVA. From this course, we will have a short research on the concept that is used for creation of a Graphical User Interface (GUI). i.e., Swing. Swing is a library of JAVA that provides many tools which is used for creation of a GUI.
- ➢ We can also have a glance about MySQL, which is the database that is being used in this project. MySQL plays a very important role in storage management in this project.
- ➢ We have also learnt how to connect JAVA to MySQL using a tool called JDBC. In order to perform operations on the MySQL table via JAVA, JDBC is required.

1.3 <u>EXPECTED OUTCOME:</u>

- ➢ The outcome of the mini project is that the user will have the freedom to book his whole tour on a single application instead of using various applications to plan his whole tour.
- ➢ Instead of booking tickets on one app, planning his tour on the other, he will be provided the same facility in a single application which provides time efficiency, and mainly, make the user comfortable and help hem store their data (plan) of the whole tour in a single place rather than storing it in various applications.

1.4 HARDWARE AND SOFTWARE REQUIREMENTS:

The requirements of the project are as follows:

Hardware Requirements:

- ➢ **Operating System** – Windows 7, 8, 10
- ➢ **RAM** – 1 GB or more
- ➢ **Processor Chip** – Intel core i3, i5, i7

Software Requirements:

- ➢ **Language** – JAVA Swing
- ➢ **Database** – MySQL
- ➢ **Application** – NetBeans IDE, MySQL Workbench, MySQL Command Client

CHAPTER 2

OBJECT ORIENTED CONCEPTS

Object-oriented programming System (OOPs) is a programming paradigm based on the concept of "objects" that contain data and methods. The primary purpose of object-oriented programming is to increase the flexibility and maintainability of programs. Object oriented programming brings together data and its behaviour(methods) in a single location makes it easier to understand how a program works.

2.1 CLASS:

The class is a group of similar entities. It is only a logical component and not the physical entity. For example, if you had a class called "Expensive Cars" it could have objects like Mercedes, BMW, Toyota, etc. Its properties(data) can be price or speed of these cars. While the methods may be performed with these cars are driving, reverse, braking etc.

➢ **Example:**

```
Public class MyClass
{
        int x=5;
}
```

2.2 OBJECT:

An object can be defined as an instance of a class, and there can be multiple instances of a class in a program. An Object contains both the data and the function, which operates on the data. For example - chair, bike, marker, pen, table, car, etc.

➢ **Example:**

```
public class MyClass
{
        int x=5;
```

```
        public static void main (String[] args)
        {
                MyClass Obj = new MyClass();
                System.out.println(obj.x);
        }
}
```
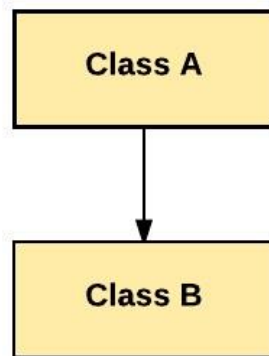
2.3 INHERITANCE:

Inheritance is an OOPS concept in which one object acquires the properties and behaviour of the parent object. It's creating a parent-child relationship between two classes. It offers robust and natural mechanism for organizing and structure of any software.

➢ **Example:**

```
class Employee
{
        float salary=40000;
}
class Programmer extends Employee
{
        int bonus=10000;
        public static void main(String args[]){
                Programmer p=new Programmer();
                System.out.println("Programmer salary is:"+p.salary);
                System.out.println("Bonus of Programmer is:"+p.bonus);
                }
}
```
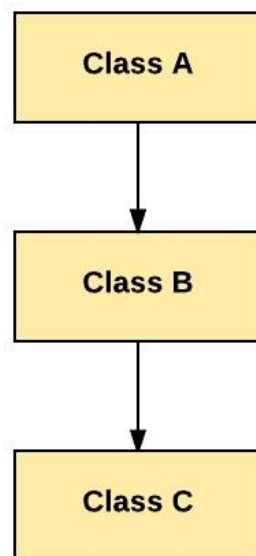
The different types of inheritance are:

➢ **Single Inheritance**: This type of inheritance allows a single subclass to inherit the properties of a parent class.

2.3.0 Single Inheritance

➢ **Multilevel Inheritance**: This type of inheritance allows a chain of inheritance where each subclass inherits the properties of the parent class and the grandparent class.

2.3.1 Multilevel Inheritance

➢ **Hierarchical Inheritance**: In this type of inheritance several subclasses inherit the properties of a single parent class.
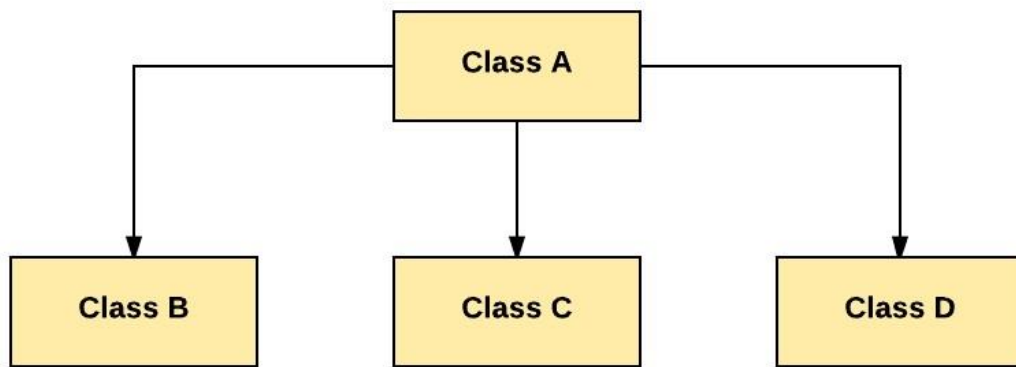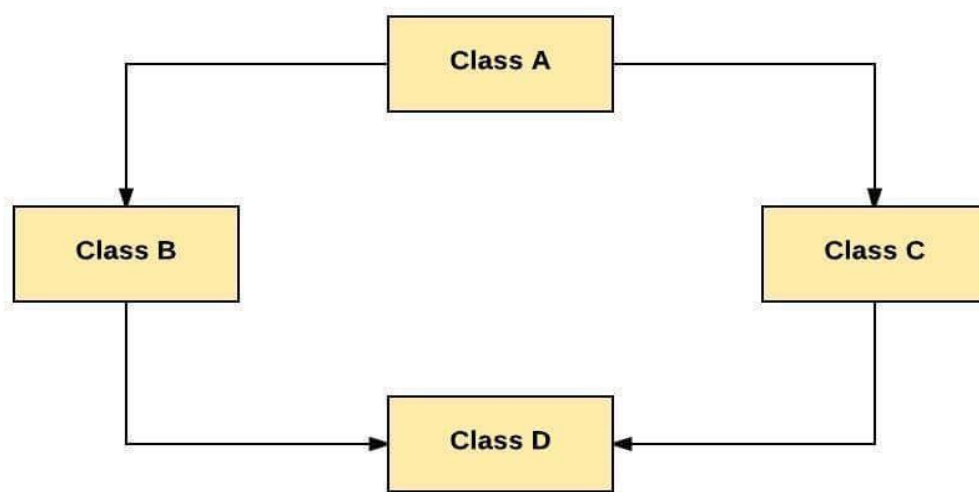
2.3.2 Hierarchical Inheritance

➢ **Hybrid inheritance**: This type of inheritance consists of different types of inheritances put together.

2.3.3 Hybrid Inheritance

2.4 POLYMORPHISM:

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. There are two types of polymorphism:

➤ **Compile-time polymorphism:** The type of polymorphism which his achieved during operations such as method overloading is called compile-time polymorphism.

Method Overloading:

When there are multiple functions with same name but different parameters then these functions are said to be overloaded. Normally, functions can be overloaded either by change in number of arguments or by change in type of arguments**.**

Example:

```
class Multiply {

        static int Mul(int a, int b)
        {
        return a * b;
        }

        static double Mul(double a, double b)
        {
        return a * b;
        }
}

class Main {
        public static void main(String[] args)
        {
        System.out.println(Multiply.Mul(6, 8));

        System.out.println(Multiply.Mul(6.4, 7.8));
        }
}
```

> **Run-time polymorphism:**

It is also known as Dynamic Method Dispatch. It is a process in which a function call to the overridden method is resolved at Runtime. This type of polymorphism is achieved during method overriding.

Example:

```
class Parent {
        void Print()
        {
                System.out.println("Parent Class");
        }
}


class childclass1 extends Parent {
        void Print()
        {
                System.out.println("childclass1");
        }
}
class chldclass2 extends Parent {
        void Print()
        {
                System.out.println("childclass2");
        }
}


Class Main {
        public static void main(String[] args)
        {
                Parent p;

                p = new childclass1();
                p.Print();

                p = new childclass2();
                p.Print();
        }
}
```

2.5 ABSTRACT CLASS:

Abstraction is a process of hiding the implementation details and showing only functionality to the user. it shows only essential things to the user and hides the internal details. It can be achieved using two components: Abstract class and Interfaces.

Abstract class is a restricted class that cannot be used to create objects. In order to access an abstract class, it must be inherited from another class. It can have abstract and non-abstract methods. It needs to be extended. It cannot be instantiated.

> **Example:**
```
abstract class MyClass{
        abstract void print();
}

class Main extends Class{
        void print(){
                System.out.println("Main class");
                }
        public static void main(String args[]){
                 MyClass obj = new Main();
                obj.run();  }
        }
```

2.6 MULTITHREADING:

A thread is a light-weight smallest part of a process that can run concurrently with the other parts(other threads) of the same process.

Java has a feature known as multithreading that allows concurrent execution of different parts of a program, this allows for maximum CPU utilization.

Two methods to implement multithreading are: -

> By extending the Thread class

> By implementing the Runnable interface

The working of a thread is as follows:
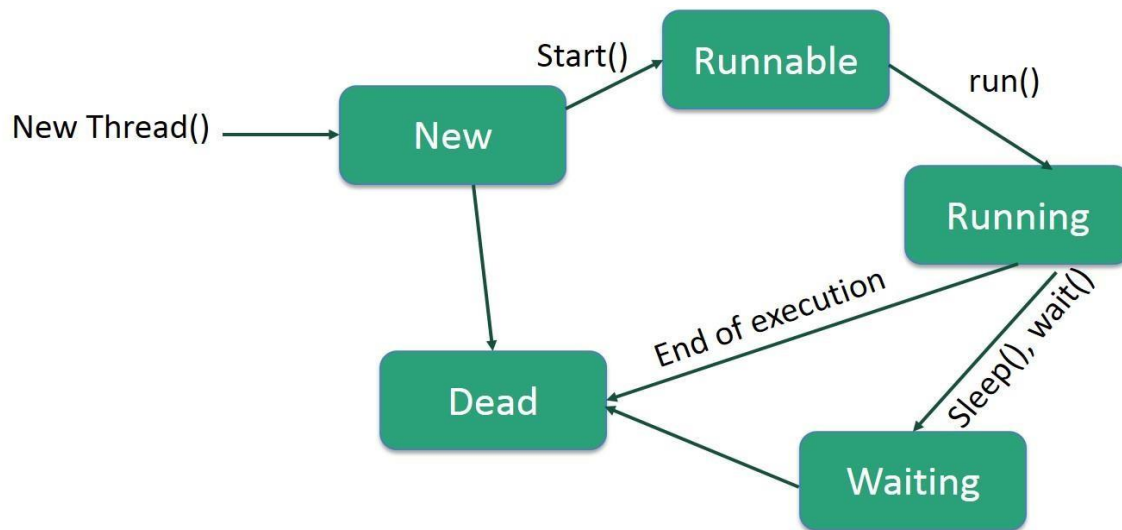
2.6.0 Thread working

2.7 I/O FUNCTIONS:

Java has various I/O streams included with its I/O package the three main streams are as follows: -

➢ **System.in**: System.in is used to read in data from any standard input device. Generally done using a java scanner.

➢ **System.out**: System.out is used to show the result of a particular executed program on a standard output device. There are three System.out functions: -

- **print()**

    Prints data passed as an argument onto the console screen.

- **println()** Does the same function as print but moves cursor to the next line.

➤ **System.err**: The standard error stream used to output the data that is thrown as an error by the program on a standard output device.

2.8 JAVA PACKAGES:

A java package is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two forms, built-in package and user-defined package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc. User-defned package is a package which is created by the user. It contains various methods, variables, etc. which have been added by the user for his purpose.

➤ **Example:**

Package mypack;

public class Main{

   public static void main(String[] args) {

      System.out.println("Welcome!"); }}

To compile and run a JAVA package, the steps are as follows:

   Compile: javac filename.java;

   Run: java Mainclassname;

There are two types of package:

➤ **Built-in Packages**

➤ **User-Defined Packages**

2.9 EXCPETION HANDLING:

An exception is an even that disrupts the flow of a program and is thrown at runtime.

The key words used to handle exceptions in java are as follows: -

➢ **try**

   This keyword is used to specify the code where an exception block is typed

   out.

➢ catch

   This keyword is used to handle the exceptions and precedes the try block.

➢ **throw**

   This keyword is used to throw an exception.

➢ **throws**

   It specifies that an exception may occur in a method.

➢ **finally**

   This keyword is used to execute a block irrespective of whether an exception

   is handled or not.

CHAPTER 3

DESIGN

3.1 DESIGN GOALS:

The design goals of this project are as follows:

➢ By demonstrating OOP concepts like polymorphism, inheritance with proper programming structure.

➢ Implementation of JAVA features like Java Swing, JDBC, I/O manipulation and Exception Handling.

➢ Creation of a GUI application which performs all the operations and provides valid results.

3.2 ALGORITHM/PSEUDOCODE:

➢ **STEP 1**: START

➢ **STEP 2:** CREATE A CLASS LOGIN AND CLASS SIGNUP

- ENTER LABEL AND TEXT FIELDS FOR BOTH THE WINDOWS

- ADD A LOGIN BUTTON AND SIGNUP BUTTON IN SEPARATE WINDOWS RESPECTIVELY.

- CHECK WHETHER THE ENTERED DETAILS IS EQUAL TO THAT IN THE DATABASE, IN CASE IF LOGIN ACTION IS TAKING PLACE.

- ADD THE DETAILS TO THE DATABASE, IF SIGNUP ACTION IS TAKING PLACE.

➢ **STEP 3:** CREATE A CLASS WAYOFTRAVEL.

- ADD LABEL FIELDS FOR TEXT AND RADIOBUTTONS FOR CHOOSING THE WAY OF TRAVEL.

➢ **STEP 4:** CREATE CLASSES FOR AIRLINE BOOKING.

- ADD ENTRY FIELDS FOR INPUT, LABEL FIELDS FOR TEXT AND OTHER ESSENTIALS FOR THE USER TO GIVE HIS/HER DETAILS.

- TAKE THE INPUT FROM THE USER AND PROVIDE THE LIST OF FLIGHTS ACCORIDNG TO THEIR NEEDS.

- CHECK WHETHER THERE IS ANY FLIGHT AVAILABLE IN THE DATABASE. ADD A TABLE TO DISPLAY THE LIST OF FLIGHTS.
- ADD LABEL AND ENTRY FIELDS TO DISPLAY THE FINAL DETAILS OF THE USER AND THE FLIGHT PREFERRED BY HIM/HER.

➢ **STEP 5**: CREATE CLASSES FOR TRAIN BOOKING:

- ADD ENTRY FIELDS FOR INPUT, LABEL FIELDS FOR TEXT AND OTHER ESSENTIALS FOR THE USER TO GIVE HIS/HER DETAILS.
- ACCESS TEHDATABASE TO RETRIEVE TE PRICE OF THE TRAIN TRAVEL.
- ADD LABEL AND ENTRY FIELDS TO DISPLAY THE FINAL DETAILS OF THE USER AND THE TRAIN PREFERRED BY HIM/HER.

➢ **STEP 6:** CREATE CLASSES FOR HOTEL BOOKING.

- ADD ENTRY FIELDS FOR INPUT, LABEL FIELDS FOR TEXT AND OTHER ESSENTIALS FOR THE USER TO GIVE HIS/HER DETAILS.
- TAKE THE INPUT FROM THE USER AND PROVIDE THE LIST OF HOTELS ACCORIDNG TO THEIR NEEDS.
- CHECK WHETHER THERE IS ANY HOTEL AVAILABLE IN THE DATABASE. ADD A TABLE TO DISPLAY THE LIST OF HOTELS.
- ADD LABEL AND ENTRY FIELDS TO DISPLAY THE FINAL DETAILS OF THE USER AND THE FLIGHT PREFERRED BY HIM/HER.

➢ **STEP 7:** CREATE CLASSES FOR CAB BOOKING.

- ADD ENTRY FIELDS FOR INPUT, LABEL FIELDS FOR TEXT AND ADD RADIOBUTTONS FOR TH USER TO SELECT THE MODE OF TRANSPORT FOR TOURING FOR THE USER TO GIVE HIS/HER DETAILS.
- TAKE THE INPUT FROM THE USER AND CHECK WHETHER THERE IS ANY VEHICLE AND DRIVER AVAILABLE FRO THEIR TIMINGS.
- ADD LABEL AND ENTRY FIELDS TO DISPLAY THE FINAL DETAILS OF THE USER AND THE FLIGHT PREFERRED BY HIM/HER.

➢ **STEP 8:** CREATE A CLASS FOR THE FINAL WINDOW.

- ADD LABEL FIELDS AND A SLIDER TO TAKE THE RATING FROM THE USER FOR THE APPLICATION CREATED.

CHAPTER 4

IMPLEMENTATION

4.1 LOGIN AND SIGNUP:

- ➢ **LOGIN:** The login screen consists of two text fields to allow entry of username and password. The Username and Password are authenticated by using a query to check if the entered credentials are same as that in the database. If credentials are correct user is logged in otherwise an error message is displayed.
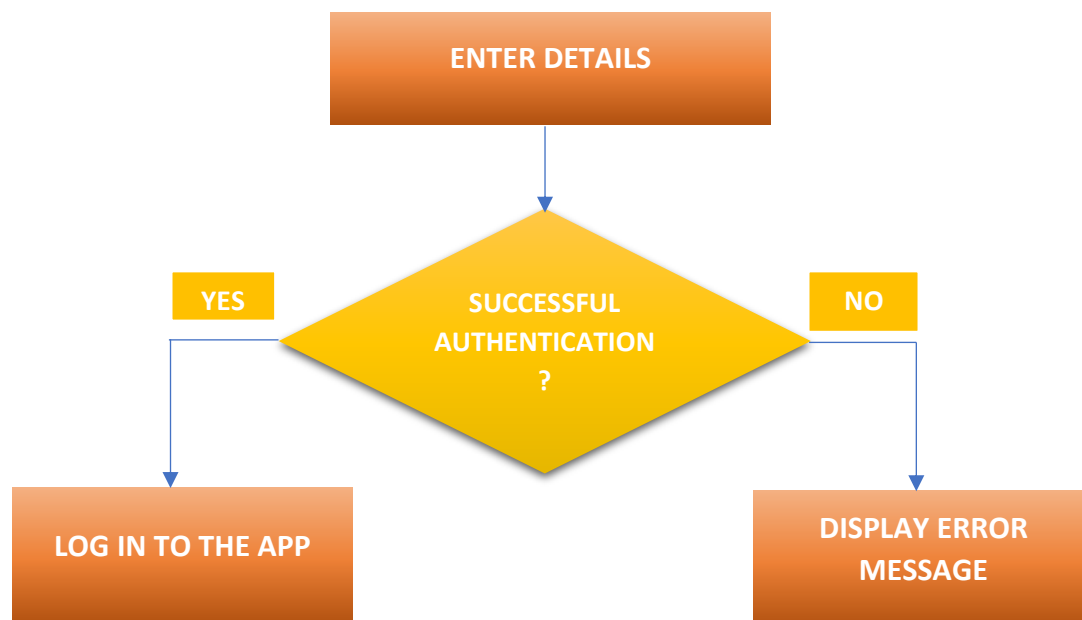
4.1.0 LOGIN

- ➢ **SIGN UP:** The user is provided with various fields to enter his details. Once entered, the details are added to the database and the log in window opens once the insertion of values into the database is successful. In the sign up window if there are fields left empty, a message pops up which asks the user to fill in the details, else, the details will not be inserted inside the database.
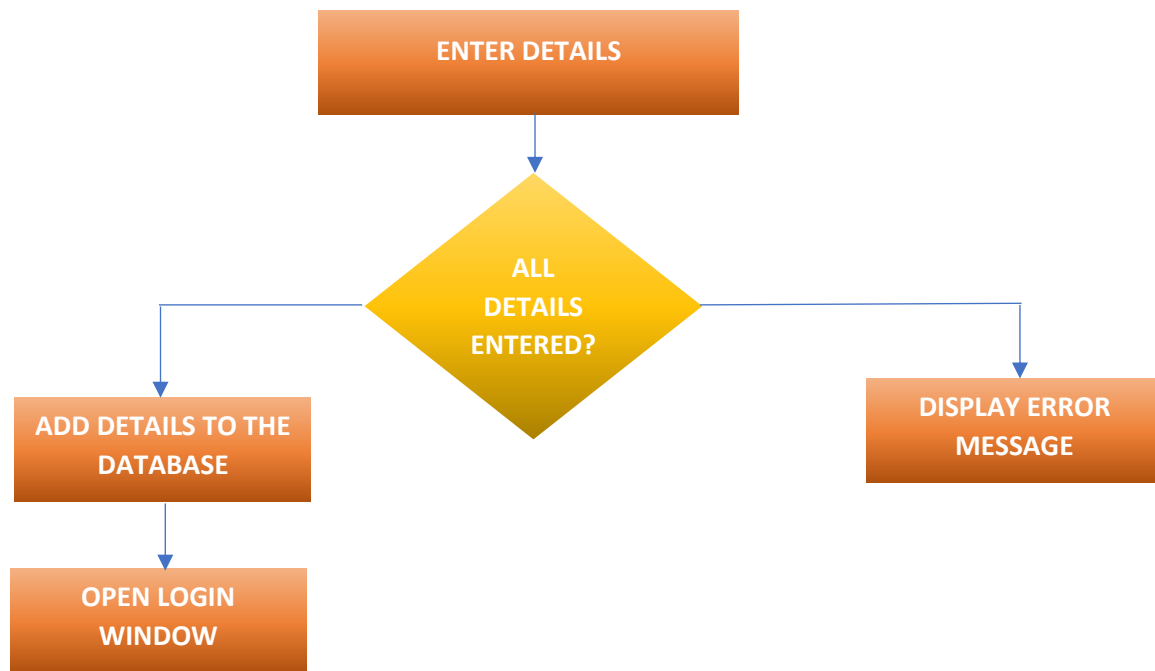
4.1.1 SIGNUP

4.2 WAY OF TRAVEL:

After logging in successfully into the application, a window appears which would ask the user to choose the type of booking he/she wants to proceed with. The user is provided with four options i.e., Flight booking, Train booking, Hotel booking, Cab booking (for touring). The user can select one option and proceed. The cab booking option can be chosen by the user if and only if the user has booked a hotel, else, the user will not be able to access that window.

4.3 FLIGHT BOOKING:

➢ If the user selects Flight Booking, the Flight booking window will appear. The first window asks the user whether their travel is domestic or international. The user must select one of these options and proceed.

➢ Once the option is selected, a new window opens which asks the user to fill the details, regarding the flight, into it. Once the details are filled, the user must click the "Search Flights" button which will enter the database and search for the flights that are best suitable according to the requirements.

➢ Once the button is clicked, a new window appears which contains a table which displays the list of flights that are available. The user can select the flight and enter his/her details

➢ . Once details are added, the final window appears which displays the information of the user and the flight and the total price. Once proceed is clicked, the payment will be successfully done and the details will be stored into the database.

```
SELECT DOMESTIC OR INTERNATIONAL
            │
            ▼
   ENTER THE FLIGHT DETAILS
            │
            ▼
     CHECK THE DATABASE
            │
            ▼
        ┌ ARE THERE ┐
 NO ────┤  FLIGHTS? ├──── YES
 │      └───────────┘         │
 ▼                            ▼
MESSAGE                ENTER USER DETAILS
DISPLAYING," SORRY            │
THERE ARE NO                  ▼
FLIGHTS!"              FINAL DISPLAY OF
                         INFORMATION
                              │
                              ▼
                      ENTRY OF INFORMATION
                        INTO DATABASE
```
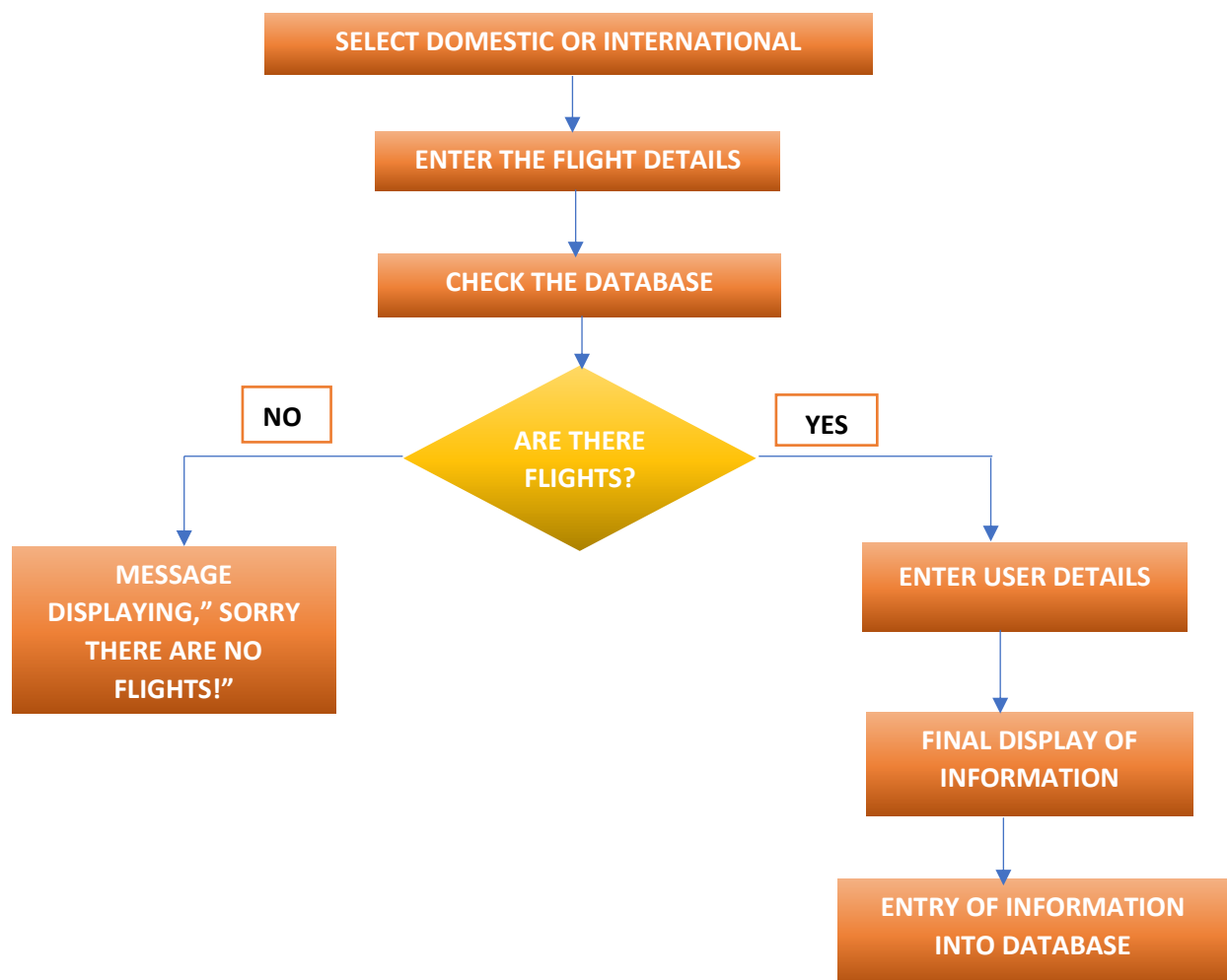
4.3.0 FLIGHT BOOKING

4.4 TRAIN BOOKING:

➢ If the user selects Train Booking as the option, the Train booking windows appears. In this window, the user is asked to enter the train details along with details of the user him/herself.
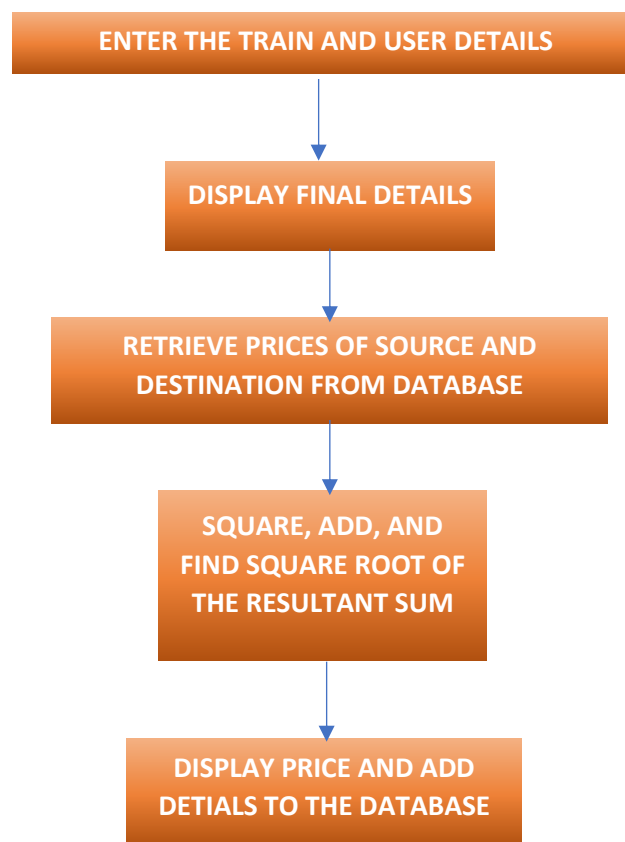
- Once the user enters his details, a window appears which shows the final details of the user and the train information as well.

- Here, there is no database for trains. So, a way to find the price for the source and destination, the database contains a table called "Price" which contains the prices of every city from a chosen centre, i.e. Bhopal.

- So, once the source and destination have been selected by the user, the price of the source to the epicentre and destination to the epicentre are taken, squared, added, and then the resulting sum is squared to give the final price.

```
┌─────────────────────────────────────┐
│   ENTER THE TRAIN AND USER DETAILS   │
└─────────────────────────────────────┘
                   │
                   ▼
        ┌──────────────────────┐
        │ DISPLAY FINAL DETAILS │
        └──────────────────────┘
                   │
                   ▼
    ┌──────────────────────────────┐
    │ RETRIEVE PRICES OF SOURCE AND │
    │  DESTINATION FROM DATABASE    │
    └──────────────────────────────┘
                   │
                   ▼
        ┌──────────────────────┐
        │   SQUARE, ADD, AND    │
        │ FIND SQUARE ROOT OF   │
        │  THE RESULTANT SUM    │
        └──────────────────────┘
                   │
                   ▼
        ┌──────────────────────┐
        │ DISPLAY PRICE AND ADD │
        │ DETIALS TO THE DATABASE│
        └──────────────────────┘
```
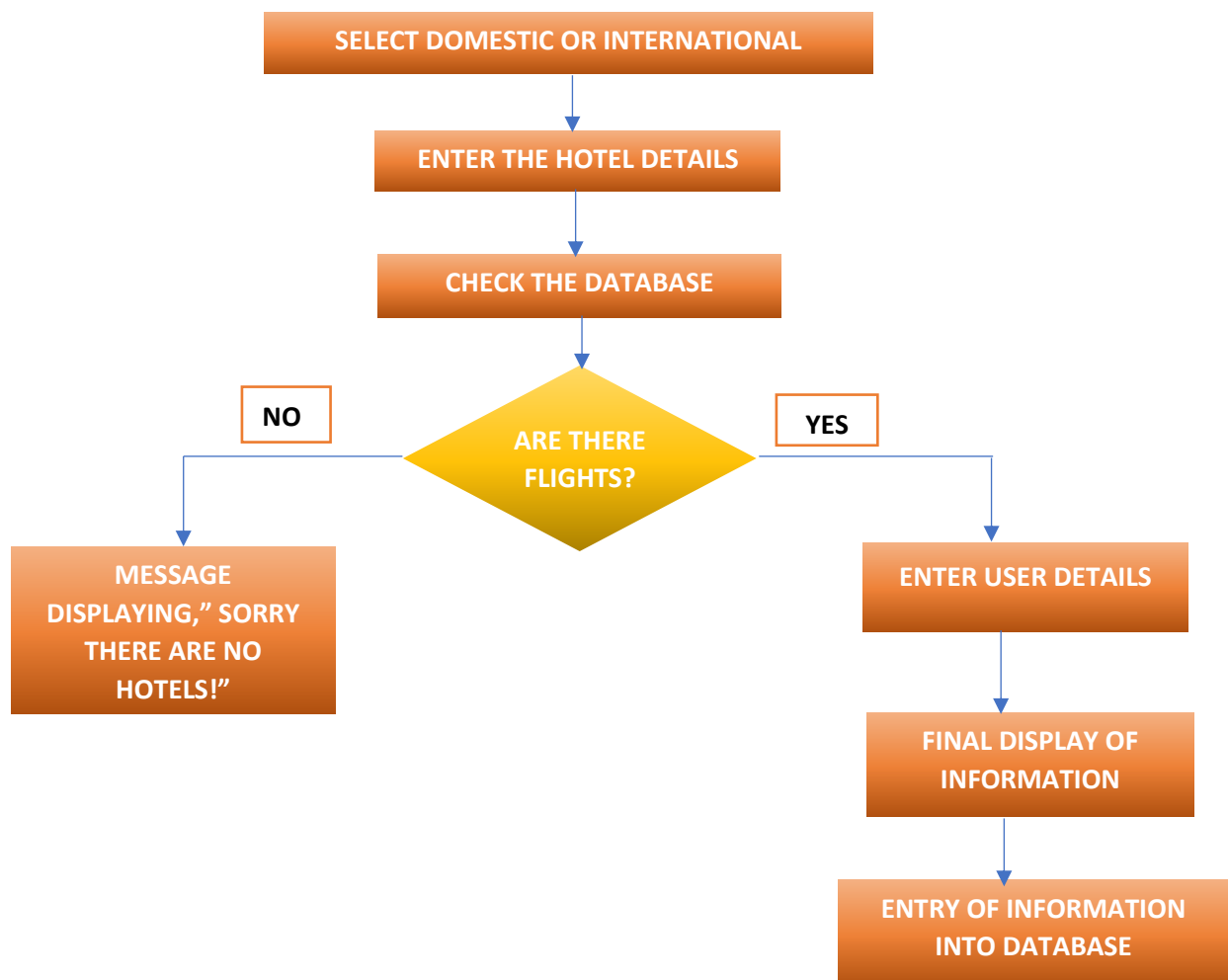
4.4.0 TRAIN BOOKING

4.5 HOTEL BOOKING:

- If the user selects Hotel Booking, the Hotel booking window will appear. The first window asks the user whether they want to book domestic hotels or international hotels. The user must select one of these options and proceed.
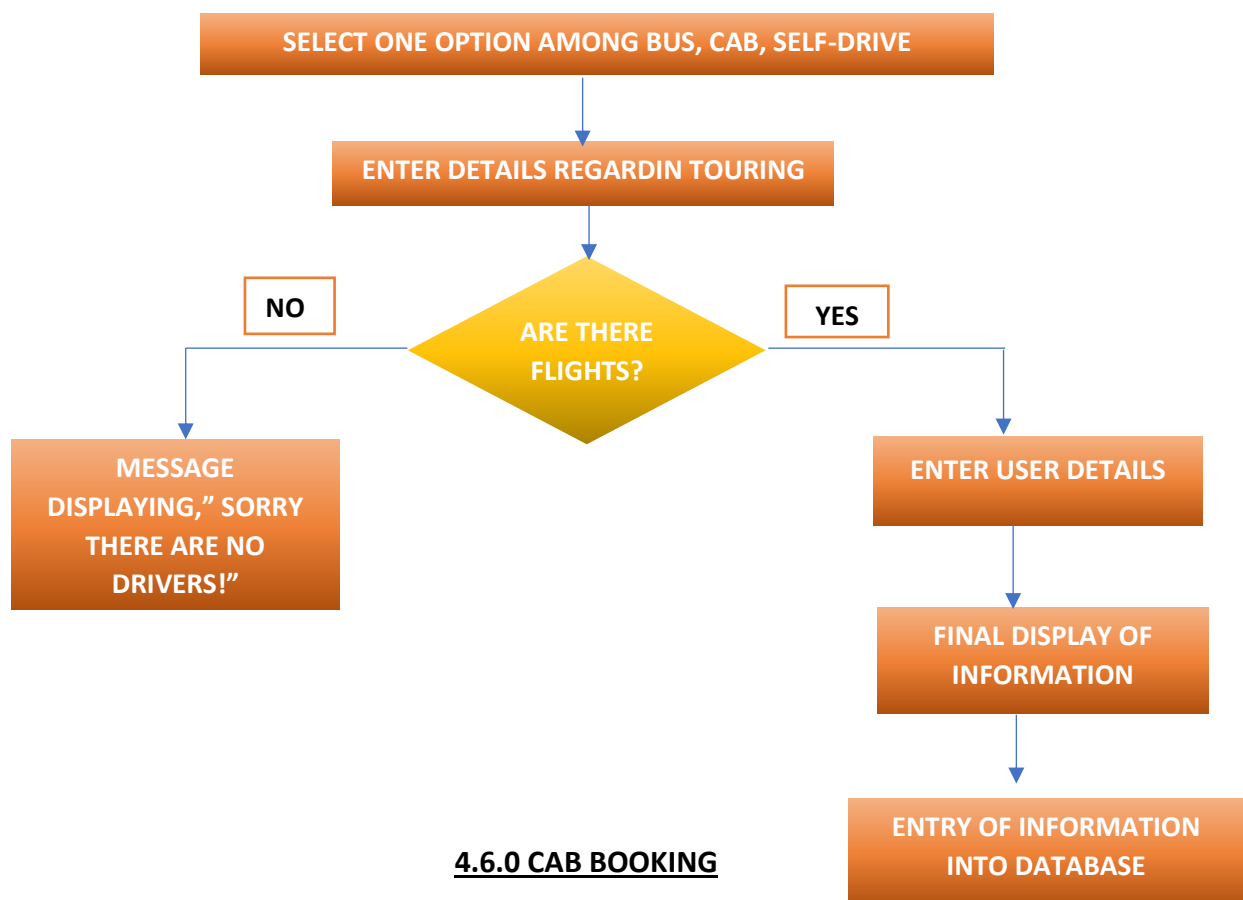
- ➢ Once the option is selected, the user has to fill the details, regarding the hotel and the user, into it. Once the details are filled, the user must click the "Search Hotels" button which will enter the database and search for the hotels that are best suitable according to the requirements.

- ➢ Once the button is clicked, a new window appears which contains a table which displays the list of hotels that are available and the number of rooms. The user can select the hotel.

- ➢ . Once details are added, the final window appears which displays the information of the user and the hotel and the total price. Once proceed is clicked, the payment will be successfully done and the details will be stored into the database.

```
        SELECT DOMESTIC OR INTERNATIONAL
                        │
                        ▼
            ENTER THE HOTEL DETAILS
                        │
                        ▼
               CHECK THE DATABASE
                        │
                        ▼
    NO  ◄─────────  ARE THERE  ─────────►  YES
    │               FLIGHTS?                 │
    ▼                                        ▼
 MESSAGE                            ENTER USER DETAILS
 DISPLAYING," SORRY                         │
 THERE ARE NO                               ▼
 HOTELS!"                          FINAL DISPLAY OF
                                   INFORMATION
                                            │
                                            ▼
                                   ENTRY OF INFORMATION
                                   INTO DATABASE
```

4.5.0 HOTEL BOOKING

4.6 CAB BOOKING:

- ➢ If the user selects Cab booking as an option, the cab booking option would not open until the user has booked a hotel because cab booking is for touring purpose.

- ➢ Once the user books the hotel and then tries cab booking, a new window appears which gives the user three options, i.e., Bus, Cab, Self-Drive.

- ➢ If the user selects Bus, he will be asked to fill the details of the bus and the same is for Cab and Self-Drive.

- ➢ Once the details are entered, the database is checked for the time slot and the driver name, information, and the price is retrieved. The user needs to enter his/her information and then, proceed.

- ➢ Once proceed is clicked, a new window opens which shows the final details of the user and the driver's information as well. Once proceed is clicked, payment is done and the details are entered into the database.

4.6.0 CAB BOOKING

CHAPTER 5

RESULTS

5.1 SIGNUP AND LOGIN:

5.2 BOOKING TYPE:

5.3 FLIGHT BOOKING:

5.3 TRAIN BOOKING:

5.4 HOTEL BOOKING:

5.5 CAB BOOKING:

SELECT MODE OF TRANSPORT FOR TOURING:

○ BUS

◉ CAB

○ SELF-DRIVE

PLEASE ENTER THE DETAILS(FOR BUS):

SELECT STARTING TIME:

SELECT NUMBER OF HOURS:

NAME:

PHONE NUMBER:

PROCEED   CANCEL

ENTER DETAILS(FOR CAB):

STARTING TIME: 1 PM

ENTER NUMBER OF HOURS: 6

ENTER NUMBER OF PEOPLE: 4

ENTER DETAILS(FOR SELF-DRIVE):

ENTER STARTING TIME:

CHOOSE ONE CAR: ----SELECT ONE-----

ENTER NUMBER OF HOURS:

ENTER YOUR DETAILS:

NAME: NIKHIL

PHONE NUMBER: 9880563376

PROCEED   CANCEL

CLICK HERE TO VIEW FINAL DETAILS:

NAME: NIKHIL

PHONE NUMBER: 9880563376

STARTING TIME: 1 PM

NUMBER OF HOURS: 6

DRIVER NAME: KARAN S

MODE OF TRANSPORT: MAHINDRA LOGAN

DRIVER PHONE NUMBER: 9731105769

PRICE: 2700

PROCEED   CANCEL

Message

Your booking was successful!

OK
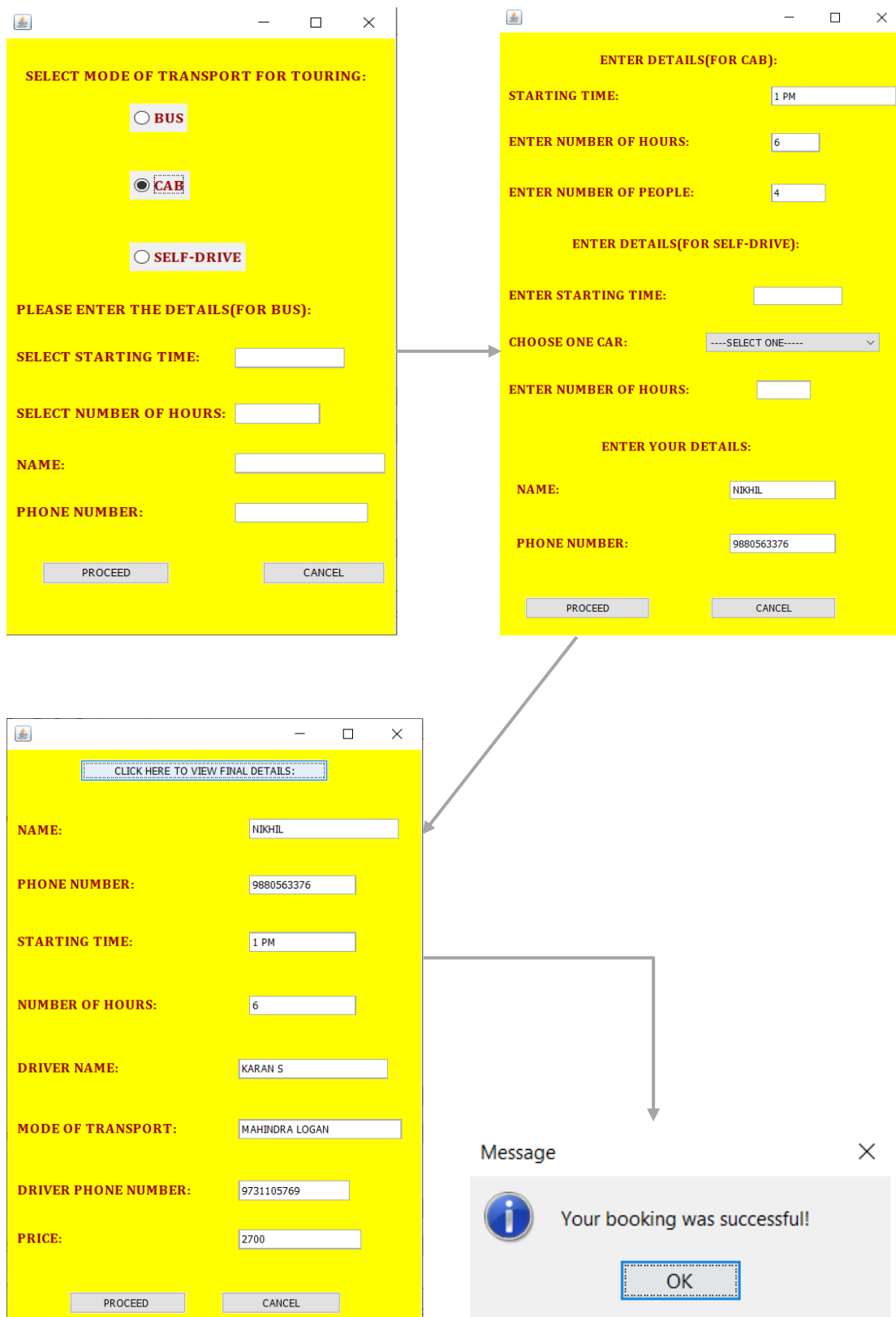
5.6 RATING:
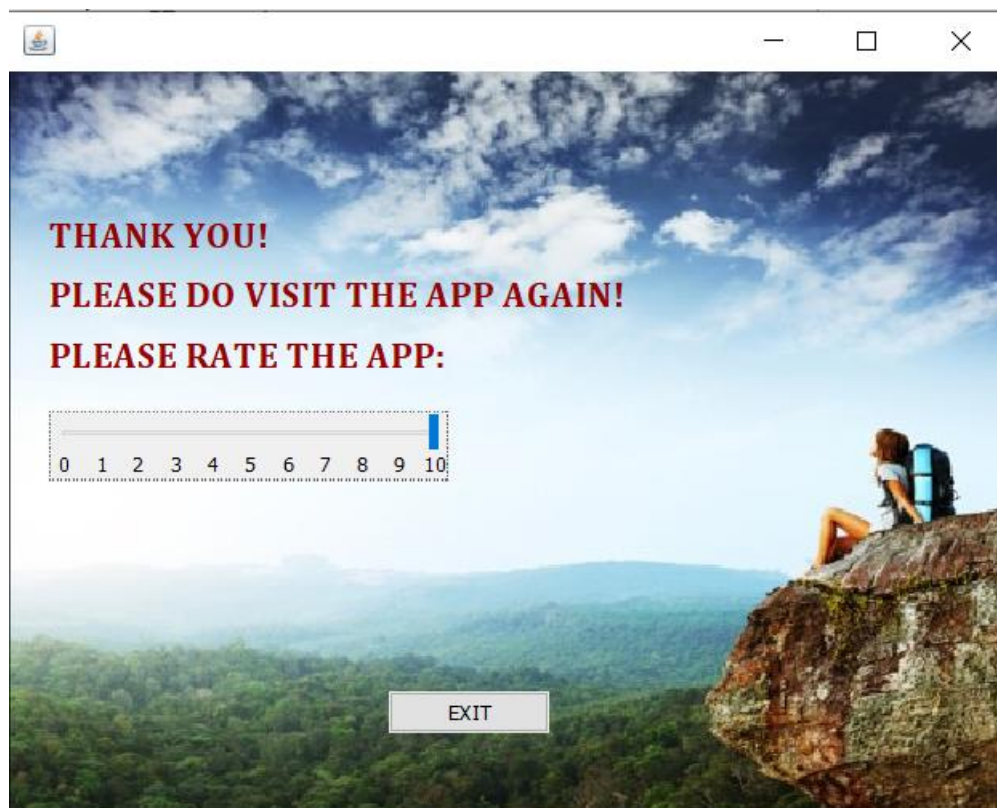
MYSQL TABLES:

➤ USER AND RATING:

```
+----------+------------------------------+-----------+--------+
| Name     | Email                        | Password  | Rating |
+----------+------------------------------+-----------+--------+
| NIKHIL S | nambiarnikhil.nn@gmail.com   | Nikhil123 |      7 |
+----------+------------------------------+-----------+--------+
```

➤ FLIGHT BOOKING DETAILS:

```
+--------+-----+--------+--------------+------------+--------+-------------+-------+
| Name   | Age | Gender | No_of_tickets | FlightName | Source | Destination | Price |
+--------+-----+--------+--------------+------------+--------+-------------+-------+
| NIKHIL |  21 | MALE   |            5 | GO AIR     | MUMBAI | KOLKATA     | 40000 |
+--------+-----+--------+--------------+------------+--------+-------------+-------+
```

> ➤ **TRAIN BOOKING DETAILS:**

Name	Age	Gender	Number_of_tickets	TrainName	TrainNo	Source	Destination	Price
NIKHIL	21	MALE	5	TRIVANDRUM - KOLKATA EXPRESS	232323	TRIVANDRUM	KOLKATA	4920

> ➤ **HOTEL BOOKING DETAILS:**

Name	Age	Gender	Number_of_people	Hotel_Name	Place	Number_of_rooms	Total_days	Grand_total
NIKHIL	21	MALE	6	HOLIDAY INN	KOLKATA	6	4	108000

> ➤ **CAB BOOKING DETAILS:**

Name	Phone	Starting_Time	Number_Of_Hours	Driver_Name	Mode_of_Transport	Driver_Phone	Price
NIKHIL	9880563376	1 PM	6	KARAN S	MAHINDRA LOGAN	9731105769	2700

> ➤ **FLIGHTS DETAILS (FROM WHERE THE LIST OF FLIGHTS ARE RETRIEVED BASED ON USER'S REQUIREMENTS):**

Name	Source	Destination	Price	WAY
Indigo Airlines	BANGALORE	CHENNAI	3000	ROUND TRIP
AIR INDIA	BANGALORE	CHENNAI	3500	ROUND TRIP
AIR INDIA	BANGALORE	CHENNAI	3000	ONE WAY
Indigo Airlines	BANGALORE	DELHI	6000	ROUND TRIP
GO AIR	BANGALORE	DELHI	4500	ONE WAY
AIR INDIA	BANGALORE	DELHI	5000	ONE WAY
AIR INDIA	BANGALORE	DELHI	6500	ROUND TRIP
AIRASIA AIRWAYS	BANGALORE	GOA	2300	ROUND TRIP
INDIGO AIRLINES	BANGALORE	GOA	1500	ONE WAY
SPICEJET	BANGALORE	GOA	2000	ROUND TRIP
GO AIR	BANGALORE	GOA	2500	ROUND TRIP
GO AIR	BANGALORE	GOA	2000	ONE WAY
AIRASIA AIRWAYS	BANGALORE	GOA	1800	ONE WAY
GO AIR	BANGALORE	HYDERABAD	2000	ONE WAY
AIR INDIA	BANGALORE	HYDERABAD	2700	ROUND TRIP
GO AIR	BANGALORE	HYDERABAD	3000	ROUND TRIP
GO AIR	BANGALORE	HYDERABAD	2500	ONE WAY
INDIGO AIRLINES	BANGALORE	KOLKATA	6500	ONE WAY

And it goes on...

➢ **FLIGHTS (FOR INTERNATIONAL - FROM WHERE THE LIST OF FLIGHTS ARE RETRIEVED BASED ON USER'S REQUIREMENTS):**

```
+----------------------+------------+------------+-------+------------+
| Name                 | Source     | Destination | Price | Way        |
+----------------------+------------+------------+-------+------------+
| AIRASIA AIRWAYS      | BANGALORE  | BALI       | 12000 | ONE WAY    |
| AIRASIA AIRWAYS      | BANGALORE  | BALI       | 23000 | ROUND TRIP |
| SINGAPORE AIRLINES   | BANGALORE  | BALI       | 36000 | ROUND TRIP |
| MULTI-AIRLINE        | BANGALORE  | LONDON     | 21000 | ONE WAY    |
| SILK AIR             | BANGALORE  | SINGAPORE  | 33000 | ROUND TRIP |
| SILK AIR             | BANGALORE  | BALI       | 33000 | ROUND TRIP |
| SINGAPORE AIRLINES   | BANGALORE  | SINGAPORE  | 31000 | ROUND TRIP |
| SINGPORE AIRLINES    | BANGALORE  | SINGAPORE  | 15000 | ONE WAY    |
| MULTI-AIRLINE        | BANGALORE  | LONDON     | 57000 | ROUND TRIP |
| MULTI-AIRLINE        | BANGALORE  | HONG KONG  | 19000 | ONE WAY    |
| GULF AIR             | BANGALORE  | LONDON     | 18000 | ONE WAY    |
| EMIRATES AIRLINES    | BANGALORE  | LONDON     | 26000 | ONE WAY    |
| SINGAPORE AIRLINES   | BANGALORE  | BALI       | 20000 | ONE WAY    |
| ETIHAD AIRWAYS       | BANGALORE  | SPAIN      | 39000 | ONE WAY    |
| TURKISH AIRLINES     | BANGALORE  | SPAIN      | 44000 | ONE WAY    |
| TURKISH AIRLINES     | BANGALORE  | LONDON     | 60000 | ROUND TRIP |
| OMAN AIR             | BANGALORE  | LONDON     | 53000 | ROUND TRIP |
| ETIHAD AIRWAYS       | BANGALORE  | SPAIN      | 71000 | ROUND TRIP |
| EMIRATES AIRLINES    | BANGALORE  | SPAIN      | 80000 | ROUND TRIP |
| EMIRATES AIRLINES    | BANGALORE  | SPAIN      | 37000 | ONE WAY    |
| SINGAPORE AIRLINES   | BANGALORE  | HONG KONG  | 25000 | ONE WAY    |
| MULTI-AIRLINE        | BANGALORE  | HONG KONG  | 37000 | ROUND TRIP |
| SINGAPORE AIRLINES   | BANGALORE  | HONG KONG  | 44000 | ROUND TRIP |
| AIR INDIA            | BANGALORE  | SINGAPORE  | 10000 | ONE WAY    |
| INDIGO AIRLINES      | BANGALORE  | SINGAPORE  | 26000 | ROUND TRIP |
| GULF AIR             | CHENNAI    | SPAIN      | 25000 | ONE WAY    |
```
and it goes on....

➢ **HOTELS (FOR DOMESTIC - FROM WHERE THE LIST OF HOTELS ARE RETRIEVED BASED ON USER'S REQUIREMENTS):**

```
+----------------------+------------+-------+----------------+
| Name                 | Place      | Price | RoomsAvailable |
+----------------------+------------+-------+----------------+
| TREEBO HOTELS        | BANGALORE  | 1550  |              5 |
| CHANCERY PAVILION    | BANGALORE  | 3950  |              4 |
| LALIT ASHOK          | BANGALORE  | 5300  |              5 |
| FAB HOTEL LOTUS PARK | BANGALORE  | 1800  |              3 |
| JW MARIOTT           | BANGALORE  | 9800  |              7 |
| COMFORT INN INSYS    | BANGALORE  | 2800  |              3 |
| FAB HOTEL TRANSIT    | DELHI      | 1900  |              2 |
| RADISSON BLU         | DELHI      | 6500  |              4 |
| HOLIDAY INN          | DELHI      | 5600  |              3 |
| JW MARIOTT           | DELHI      | 9700  |              6 |
| TAJ PALACE           | DELHI      | 10000 |              3 |
| ROYALA PLAZA         | DELHI      | 6100  |              2 |
| HILTON               | MUMBAI     | 8600  |              5 |
| TREEBO INSIDE        | MUMBAI     | 3200  |              4 |
| LALIT ASHOK          | MUMBAI     | 7600  |              2 |
| RAMADA PLAZA         | MUMBAI     | 5900  |              5 |
| TAJ                  | MUMBAI     | 12000 |              7 |
| TREEBO TREND VEDANTO | KOLKATA    | 2100  |              4 |
| PEERLESS INN         | KOLKATA    | 4200  |              5 |
| PAI VICEROY          | KOLKATA    | 6500  |              3 |
```
and it goes on...

> ## HOTELS (FOR INTERNATIONAL - FROM WHERE THE LIST OF HOTELS ARE RETRIEVED BASED ON USER'S REQUIREMENTS):

```
+------------------------+-----------+--------+----------------+
| Name                   | Place     | Price  | RoomsAvailable |
+------------------------+-----------+--------+----------------+
| HILTON GRANDEN INN     | SINGAPORE |  6500  |              5 |
| HOLIDAY INN            | SINGAPORE | 10000  |              4 |
| CAPITOL KEMPENSKI      | SINGAPORE | 17800  |              6 |
| MANDARIN ORCHARD       | SINGAPORE | 13500  |              4 |
| JW MARIOTT             | SINGAPORE | 12700  |              9 |
| NOVOTEL                | HONG KONG |  5900  |              4 |
| REGAL REIVERRIDE       | HONG KONG |  3900  |              7 |
| HOTEL ICON             | HONG KONG | 11500  |              5 |
| IMPERIAL               | HONG KONG |  3900  |              2 |
| REGAL ORENTAL          | HONG KONG |  4900  |              4 |
| NH CIUDAD DE ALMERIA   | SPAIN     |  5200  |              9 |
| NH COMPO CARTEGENA     | SPAIN     |  6700  |              2 |
| EXPO HOTEL VALENCIA    | SPAIN     |  3900  |              6 |
| ILUNION BARCELONA      | SPAIN     |  4900  |              3 |
| NOVOTEL MADRID CITY    | SPAIN     |  8000  |              6 |
| TRAVELODGE LONDON ENFIELD | LONDON |  3900  |              3 |
| INHABIT                | LONDON    |  9700  |              7 |
| CAPTHORNE TARA HOTEL   | LONDON    |  7800  |              5 |
```
and it goes on..

> ## CAB (FROM WHERE THE DRIVER DETAILS ARE RETRIEVED IF THE USER SELECTS CAB):

```
+----------------+-------+----------------------+------------------+-------+------------+
| NAME           | TIME  | CAR                  | NUMBER_OF_PEOPLE | PRICE | Phone      |
+----------------+-------+----------------------+------------------+-------+------------+
| SANJAY MISHRA  | 9 AM  | MARUIT SUZUKI SWIFT  |                4 |  400  | 9731105750 |
| ROHAN MISHRA   | 9 AM  | MARUIT SUZUKI ERITGA |                6 |  500  | 9731105751 |
| ROHAN PANDEY   | 9 AM  | TATA INDICA          |                3 |  400  | 9731105752 |
| MANISH PANDEY  | 9 AM  | TOYOTA INNOVA        |                7 |  600  | 9731105753 |
| DHRUV BHAT     | 10 AM | TOYOTA INNOVA        |                7 |  600  | 9731105754 |
| MAHESH BHAT    | 10 AM | TOYOTA ETIOS         |                4 |  400  | 9731105755 |
| SUNIL PATIL    | 10 AM | MARUTI SUZUKI ERTIGA |                6 |  600  | 9731105756 |
| HIMANSHU GUPTA | 10 AM | TATA INDICA          |                3 |  400  | 9731105757 |
| RISHI GUPTA    | 11 AM | TATA INDICA          |                3 |  450  | 9731105758 |
| MURALI NAMBIAR | 11 AM | TOYOTA INNOVA        |                7 |  650  | 9731105759 |
| HIMESH PATEL   | 11 AM | TOYOTA ETIOS         |                4 |  450  | 9731105760 |
| ROHAN TYAGI    | 11 AM | MAHINDRA LOGAN       |                5 |  500  | 9731105761 |
| SUNIL TYAGI    | 11 AM | MARUTI SUZUKI ERITGA |                6 |  600  | 9731105762 |
| NISHANT G      | 12 PM | MARUTI SUZUKI ERITGA |                6 |  650  | 9731105763 |
| ALOK S         | 12 PM | TATA INDICA          |                3 |  450  | 9731105764 |
| ISMAIL AHME    | 12 PM | MAHINDRA LOGAN       |                5 |  500  | 9731105765 |
| HARSH PATEL    | 12 PM | TOYOTA INNOVA        |                7 |  700  | 9731105767 |
| KARAN T        | 12 PM | TOYOTA ETIOS         |                4 |  500  | 9731105768 |
| KARAN S        | 1 PM  | TOYOTA ETIOS         |                4 |  450  | 9731105769 |
| URJIT S        | 1 PM  | TOYOTA INNOVA        |                7 |  650  | 9731105770 |
| MANISH R       | 1 PM  | MARUTI SUZUKI ERTIGA |                6 |  600  | 9731105771 |
| DHRUV S        | 1 PM  | TATA INDICA          |                3 |  400  | 9731105772 |
```

And it goes on...

➢ **BUS (FROM WHERE THE BUS DETAILS ARE RETRIEVED IF THE USER SELECTS BUS):**

```
| NAME      | TIME  | PRICE | Phone      | mode |
+-----------+-------+-------+------------+------+
| ROHAN     | 9 AM  |   850 | 9880563371 | BUS  |
| MURUGAN   | 10 AM |   850 | 9880563372 | BUS  |
| BHARAN    | 11 AM |   900 | 9880563373 | BUS  |
| BINOY     | 12 PM |   950 | 9880563374 | BUS  |
| RISHIKESH | 1 PM  |  1000 | 9880563375 | BUS  |
| MURALI    | 2 PM  |  1000 | 9880563376 | BUS  |
| HIMESH    | 3 PM  |   900 | 9880563377 | BUS  |
| JOHN      | 4 PM  |  1000 | 9880563378 | BUS  |
+-----------+-------+-------+------------+------+
```

➢ **SELF-DRIVE (FROM WHERE THE CAR DETAILS ARE RETRIEVED IF THE USER SELECTS SELF - DRIVE):**

```
+-----------------------------+-------+
| CarName                     | Price |
+-----------------------------+-------+
| MARUTI SUZUKI SWIFT DEZIRE  |   400 |
| MARUTI SUZUKI ERTIGA        |   550 |
| MARUTI SUZUKI VITARA BREZZA |   450 |
| TATA INDICA                 |   350 |
| TOYOTA INNOVA               |   650 |
| MAHINDRA LOGAN              |   500 |
| MAHINDRA SCORPIO            |   600 |
| TOYOTA ETIOS                |   450 |
+-----------------------------+-------+
```

CHAPTER 6

CONCLUSION

Therefore, this mini project, i.e., Project on creating an application called "Traveller's Stop", done using Java's library, Swing and the back-end database, MySQL, is performing every task assigned, successfully. From this project, I have been able to gain knowledge about a lot of things. Firstly, I have learnt about the JAVA's library which is used for the creation of a GUI, i.e., JAVA Swing. It has been very helpful in creating the GUI and the widgets inside it are very helpful is gaining access of the user's data. I have also updated my knowledge on MySQL, which is performing the role of storage of data. I have also learnt about the flow of data i.e., how the data of the user flows from the GUI to the database using JDBC. JDBC has played a very important role in performing operations in MySQL through JAVA. Therefore, the project has been very useful in learning new things about various features of JAVA and MySQL.

CHAPTER 7

REFERENCES

The references used for this mini project are:

- ➢ www.geeksforgeeks.com
- ➢ JAVA: A Complete Reference - Herbert Schildt
- ➢ Programming Knowledge – Youtube channel