JAVA AWT BASED -TRANSPORT MANAGEMENT SYSTEM FOR INTRA CITY TRAVELS -SQL CONNECTIVITY USING JDBC

A

Report

submitted in partial fulfilment of

BE IV SEMESTER DATABASE MANAGEMENT SYSTEM LAB

INFORMATION TECHNOLOGY

By

Kodipyaka Vinay Babu <1602-18-737-117>

Under the Guidance of

B.Leelavathy

Department of Information Technology

Vasavi College of Engineering(Autonomous)

(Affliated to Osmania University)

Ibrahimbagh, Hyderabad-500031

2019-2020

BONAFIDE CERTIFICATE

This is to certify that this project report titled **"TRANSPORT MANAGEMENT SYSTEM FOR INTRA CITY TRAVELS"** is the bonafied mini project work of **Mr. Kodipyaka Vinay babu** bearing hall ticket number **1602-18-737-117** under the guidance of **B. Leelavathy** during 4th sem B.E for the academic year **2019-2020.**

External Examiner

Internal Examiner

B.LEELAVATHY

Assistant professor

Department of Information Technology

AIM AND PRIORITY OF THE PROJECT:

To create a GUI based form for the project of **TRANSPORT MANAGEMENT SYSTEM FOR INTRA CITY TRAVELS** where in a passenger checks a set of vehicles and their cost and reserves a vehicle according to his budget and finally makes payment.

The values entered (insertion, updation, deletion)by the user for respective table in **GUI** should be updated in the database using **JDBC.**

ABSTRACT: The project is about transport management system for intra city travels. Starting with intracity refers to outside the city. There will be different ways of transportation to move from one place to another. The project is about creating a application where in user can book different modes of transport in a single place. Starting with user creates a account giving the details such as name, id, age mobile number, email id etc. cost for each vehicle varies based on its specification. User after creating the account selects the vehicle based on his requirement that is he will select's the vehicle based on the cost and finally makes the payment and user gets the receipt of his transaction. As the details of the availability of each vehicle is stored in the database. After booking the vehicle availability will be decremented.

INTRODUCTION :

REQUIREMENTS:

Tables Required: (5) Passenger, checks, typeofvehicle, reserves, payment

TABLE NAME	DESCRIPTION	ATTRTIBUTE	DATATYPE
PASSENGER	Passenger id	pid	Number (10)
	Age of passenger	age	Number(5)
	Mobile number	mobnum	Number(10)
	Email id of passenger	emailid	Varchar2(20)
	Name of the passenger	pname	Varchar2(20)
CHECKS	Passenger id	pid	Number(10)
	Vehicle id	vid	Number(10)
	Date of travel	day	Date
RESERVES	Passenger id	pid	Number(10)
	Vehicle id	vid	Number(10)
	Payment id	payid	Number(10)
	Date of Reservation	day	date

TYPE OF VEHICLE	Vehicle id	vid	Number(10)
	Availability Number	availability	Number(5)
	Cost of the vehicle	cost	Number(5)
	Specification	type	Varchar2(10)
	Type of vehicle chosen	Modeoftransport	Varchar2(20)
PAYMENT	Payment id	pid	Number(10)
	Payment Status	status	Varchar2(10)
	Bill Reciept	bill	Varchar2(10)

ARCHITECTURE AND TECHNOLOGY USED:

Java Eclipse, Oracle 11g Database, java SE version 8, SQL *plus ,java AWT

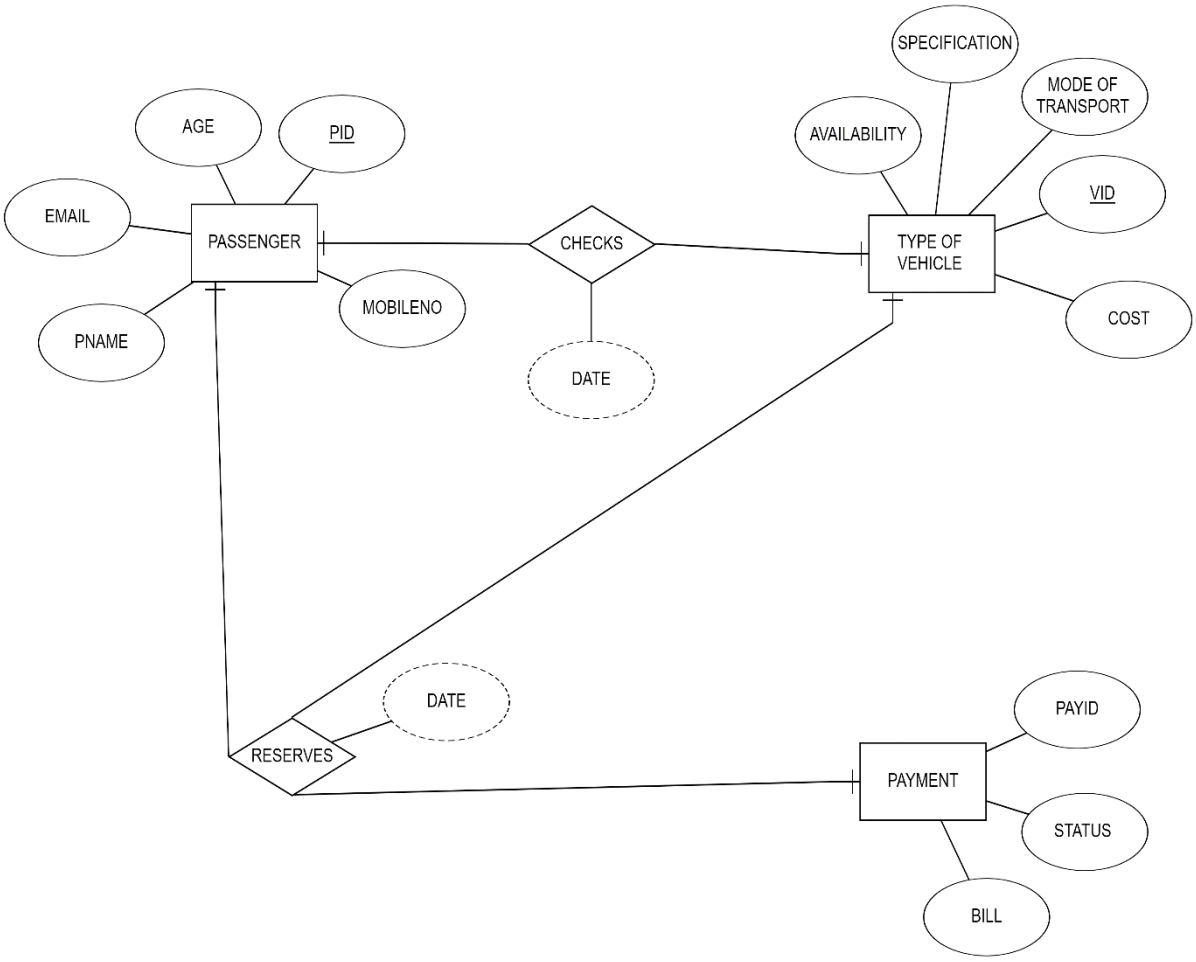
Eclipse: It is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug in system for customizing the environment. The Eclipse software development kit (SDK), which include java development tools is meant for java developers.

SQL *plus: SQL *plus is a command line tool proprietary to oracle. You can send SQL Queries to the server using the tool. It can also help you format the result of query. SQL is the query language that is used to communicate with the oracle server to access and modify data.

JAVA AWT: Abstract window tool kit is an API to develop GUI or Window based applications in java. Java AWT components are platform dependent i.e components are displayed according to the view of operating system. AWT is heavy weight that is components are using the resources of O.S.

JDBC: Java Database Connectivity is an application programming interface (API) for the programming language java **,** which defines how a client may access database. It is a java based data access technology used for java database connectivity. It is the part of java Standard Edition Platform, from oracle corporation.

ER DIAGRAM OF THIS PROJECT:



CREATION OF TABLES:

```
SQL> create table passenger(pid number(5) primary key ,
  2          age number(5) ,
  3          emailId varchar2(20) ,
  4          pname  varchar2(20),
  5          mobnu number(10) );
  SQL> create table typeofvehicle(vid number(5) primary key ,
  2           modeoftransport varchar2(10) ,
  3           cost number(5) ,
  4           availability number(10) ,
  5           type varchar2(10) );
  SQL> create table payment(payid number(10) primary key ,
  2             status varchar2(20) ,
  3             bill varchar2(10) );
SQL> create table checks(pid number(5) ,
  2         vid number(5),
  3         day date,
  4         primary key(pid,vid) ,
  5         foreign key(pid) references passenger ,
  6         foreign key(vid) references typeofvehicle );
SQL> create table reserves(pid number(5) ,
  2         vid number(5) ,
  3          payid number(10) ,
  4          day date ,
```

```
5       primary key(pid,vid,payid) ,
6       foreign key(pid) references passenger ,
7       foreign key(vid) references typeofvehicle ,
8       foreign key(payid) references payment );
```

TABLES DESCRIPTION:

SQL> desc passenger;

Name	Null?	Type
PID	NOT NULL	NUMBER(5)
AGE		NUMBER(5)
EMAILID		VARCHAR2(20)
PNAME		VARCHAR2(20)
MOBNU		NUMBER(10)

SQL> desc checks;

Name	Null?	Type
PID	NOT NULL	NUMBER(5)
VID	NOT NULL	NUMBER(5)
DAY		DATE

SQL> desc typeofvehicle;

Name	Null?	Type
VID	NOT NULL	NUMBER(5)
MODEOFTRANSPORT		VARCHAR2(10)

COST NUMBER(5)

AVAILABILITY NUMBER(10)

TYPE VARCHAR2(10)

SQL> desc reserves;

Name	Null?	Type
PID	NOT NULL	NUMBER(5)
VID	NOT NULL	NUMBER(5)
PAYID	NOT NULL	NUMBER(10)
DAY		DATE

SQL> desc payment;

Name	Null?	Type
PAYID	NOT NULL	NUMBER(10)
STATUS		VARCHAR2(20)
BILL		VARCHAR2(10)

IMPLEMENTATION

I) FRONT END PROGRAMS AND CONNECTIVITY

JAVA-SQL CONNECTIVITY USING JDBC:

The connection to the database can be performed using java programming **(JDBC API)** as**:**

```
public void connectToDB() {
try {
        connection=DriverManager.getConnection("jdbc:oracle:thin:@localhost:
1521:ORCL","dbmsass","dbms");
                statement = connection.createStatement();
}
catch (SQLException connectException) {
    System.out.println(connectException.getMessage());
    System.out.println(connectException.getSQLState());
    System.out.println(connectException.getErrorCode());
    System.exit(1);
}
catch(Exception e){
   System.out.println("Unable to find and load driver"); System.exit(1);
} }
```

AS THIS PROJECT CONTAINS 5 TABLES

i.**e PASSENGER ,CHECKS,TYPEOFVEHICLE, RESERVES, PAYMENT**

BELOW IS THE CODE FOR ALL **DML OPERATIONS** ON THE TABLE **PASSENGER**

MAIN FRAME GUI :

```
package transportmanagement;

import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

class abc extends JFrame{

        private JMenuBar mBar;

private JMenu

mnuPassenger,mnuChecks,mnuReserves,mnuTypeofVehicle,mnuPayment;

private JMenuItem


mnuItemInsert,mnuItemInsert1,mnuItemInsert2,mnuItemInsert3,mnuItemInsert
4;

        private JMenuItem
mnuItemUpdate,mnuItemUpdate1,mnuItemUpdate2,mnuItemUpdate3,mnuItem
Update4;

        private JMenuItem
mnuItemDelete,mnuItemDelete1,mnuItemDelete2,mnuItemDelete3,mnuItemDe
lete4;

        abc(){

        mBar=new JMenuBar();

        mnuPassenger=new JMenu("passenger");

        mnuPassenger.setOpaque(true);

        mnuPassenger.setBackground(Color.blue);

        mnuChecks=new JMenu("checks");

       mnuChecks.setOpaque(true);

        mnuChecks.setBackground(Color.green);
```

```
 mnuReserves=new JMenu("reserves");
 mnuReserves.setOpaque(true);
 mnuReserves.setBackground(Color.MAGENTA);
 mnuTypeofVehicle=new JMenu("typeOfvehicle");
mnuTypeofVehicle.setOpaque(true);
 mnuTypeofVehicle.setBackground(Color.ORANGE);
  mnuPayment=new JMenu("payment");
 mnuPayment.setOpaque(true);
 mnuPayment.setBackground(Color.PINK);
  mnuItemInsert=new JMenuItem("insert passenger");
 mnuItemInsert.setBackground(Color.yellow);
 mnuItemUpdate=new JMenuItem("update passenger");
 mnuItemUpdate.setBackground(Color.yellow);
 mnuItemDelete=new JMenuItem("delete passenger");
 mnuItemDelete.setBackground(Color.yellow);

 mnuItemInsert1=new JMenuItem("insert checks");
 mnuItemInsert1.setBackground(Color.orange);
 mnuItemUpdate1=new JMenuItem("update checks");
 mnuItemUpdate1.setBackground(Color.orange);
 mnuItemDelete1=new JMenuItem("delete checks");
 mnuItemDelete1.setBackground(Color.orange);

 mnuItemInsert2=new JMenuItem("insert reserves");
 mnuItemInsert2.setBackground(Color.pink);
 mnuItemUpdate2=new JMenuItem("update reserves");
 mnuItemUpdate2.setBackground(Color.pink);
```

```
mnuItemDelete2=new JMenuItem("delete reserves");
mnuItemDelete2.setBackground(Color.pink);

 mnuItemInsert3=new JMenuItem("insert typeofvehicle");
 mnuItemInsert3.setBackground(Color.green);
 mnuItemUpdate3=new JMenuItem("update typeofvehicle");
 mnuItemUpdate3.setBackground(Color.green);
 mnuItemDelete3=new JMenuItem("delete typeofvehicle");
 mnuItemDelete3.setBackground(Color.green);

 mnuItemInsert4=new JMenuItem("insert payment");
 mnuItemInsert4.setBackground(Color.MAGENTA);
 mnuItemUpdate4=new JMenuItem("update payment");
 mnuItemUpdate4.setBackground(Color.MAGENTA);
 mnuItemDelete4=new JMenuItem("delete payment");
 mnuItemDelete4.setBackground(Color.MAGENTA);

 mnuPassenger.add(mnuItemInsert);
 mnuPassenger.add(mnuItemUpdate);
 mnuPassenger.add(mnuItemDelete);

 mnuChecks.add(mnuItemInsert1);
 mnuChecks.add(mnuItemUpdate1);
 mnuChecks.add(mnuItemDelete1);

 mnuReserves.add(mnuItemInsert2);
 mnuReserves.add(mnuItemUpdate2);
```

```
        mnuReserves.add(mnuItemDelete2);

        mnuTypeofVehicle.add(mnuItemInsert3);
        mnuTypeofVehicle.add(mnuItemUpdate3);
        mnuTypeofVehicle.add(mnuItemDelete3);

        mnuPayment.add(mnuItemInsert4);
        mnuPayment.add(mnuItemUpdate4);
        mnuPayment.add(mnuItemDelete4);

        mBar.add(mnuPassenger);
        mBar.add(mnuChecks);
        mBar.add(mnuReserves);
        mBar.add(mnuTypeofVehicle);
        mBar.add(mnuPayment);
        setJMenuBar(mBar);
        setSize(2000,1000);
        setLayout(new GridBagLayout());
        JLabel lbl=new JLabel("TRANSPORT MANAGEMENT SYSTEM FOR
INTRA CITY TRAVELS");
        lbl.setFont(lbl.getFont().deriveFont(40.0f));
        add(lbl);
        getContentPane().setBackground(Color.cyan);
        setTitle("TRANSPORT MANAGEMENT SYSTEM FOR INTRA CITY
TRAVELS");
        setVisible(true);
        mnuItemInsert.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent ae) {
```

```
                new Passenger();
                }
});
mnuItemUpdate.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
                new updatePassenger();
        }
});
mnuItemDelete.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                new DeletePassenger();
        }
});
mnuItemInsert1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
                new insertchecks();
        }
});
mnuItemUpdate1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
                new UpdateChecks();
        }
});
mnuItemDelete1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
                new DeleteChecks();
        }
```

```
});
mnuItemInsert2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
                new insertReserves();
        }
});
mnuItemUpdate2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
                new updateReserves();
        }
});
mnuItemDelete2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
                new DeleteReserves();

        }
});
mnuItemInsert3.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
                new insertTypeOfVehicle();
        }
});
mnuItemUpdate3.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                new updateTypeOfVehicle();

        }
```

```
        });
        mnuItemDelete3.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        new DeleteTypeOfVehicle();
                }
        });
        mnuItemInsert4.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent ae) {
                        new insertPayment();
                }
        });
        mnuItemUpdate4.addActionListener(new ActionListener() {

                public void actionPerformed(ActionEvent ae) {
                        new updatePayment();
                }

        });
        mnuItemDelete4.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        new DeletePayment();

                }
        });
        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        }
}
```

```
public class FirstFrameGUI {
	public static void main(String[] args){
		new abc();
	}
}
```

INSERT PASSENGER:

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
public class Passenger extends JFrame{
	private JPanel pnl1, pnl2;
	private JTextField txtPid,txtAge,txtEmail,txtPname,txtMobnu;
	private JLabel lblPid, lblAge, lblEmail, lblPname, lblMobnu;
	private JTextArea txtAre;
	private JButton btnInsert;
	Connection connection;
	Statement statement;
	Passenger(){
		  txtPid=new JTextField(15);
		txtAge=new JTextField(15);
		  txtEmail=new JTextField(15);
		txtPname=new JTextField(15);
		 txtMobnu=new JTextField(15);
		 lblPid=new JLabel("PID");
		lblAge=new JLabel("ÄGE");
		  lblEmail=new JLabel("EMAIL");
```

```
            lblPname=new JLabel("PASSENGER NAME");
              lblMobnu=new JLabel("MOBILE NUMBER");
              pnl1=new JPanel();
            pnl2=new JPanel();
              btnInsert=new JButton("SUBMIT");
              txtAre=new JTextArea(20,100);
              pnl1.setLayout(new GridLayout(5,2));
              pnl1.add(lblPid);            pnl1.add(txtPid);
              pnl1.add(lblAge);            pnl1.add(txtAge);
              pnl1.add(lblEmail);          pnl1.add(txtEmail);
              pnl1.add(lblPname);          pnl1.add(txtPname);
              pnl1.add(lblMobnu);          pnl1.add(txtMobnu);
              pnl2.add(btnInsert); pnl2.add(txtAre);
              pnl1.setSize(600,500);pnl2.setSize(600,500);
              setTitle("INSERT PASSENGER");
              pnl1.setBackground(Color.MAGENTA);
              pnl2.setBackground(Color.CYAN);
              add(pnl1);add(pnl2);
              setSize(2000,1000);setLayout(new GridLayout(2, 1));
              setVisible(true);
              btnInsert.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e) {
        System.err.println("Unable to find and load driver");
```

```
        System.exit(1);
}
try {
        connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL","dbm
sass","dbms");
        statement = connection.createStatement();
}
catch (SQLException connectException) {
        System.exit(1);
}
try {
        String query= "INSERT INTO passenger VALUES(" + txtPid.getText()
+ ", "  + txtAge.getText() + "," +"'" + txtEmail.getText() + "'," + "'" +
txtPname.getText()  + "'," + txtMobnu.getText() + " )";
        int i = statement.executeUpdate(query);
        txtAre.append("\nInserted " + i + " rows successfully");
}
catch (SQLException insertException) {
        System.out.println(insertException);
        }
    }
 });
  this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
}
}
```

UPDATE PASSENGER:

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

```
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
public class updatePassenger extends JFrame {
        private JPanel pnl1,pnl0,pnl2;
        private JTextField txtPid, txtAge, txtEmail,txtPname,txtMobnu;
        private JLabel lblPid,lblAge, lblEmail, lblPname,lblMobnu;
        private List list;
        private JTextArea txtAre,txtAre2;
        private JButton btnUpdate;
        Connection connection;
        Statement statement;
        public void connectToDB() {
try {
connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL","dbm
sass","dbms");
statement = connection.createStatement();
}
catch (SQLException connectException) {
        System.exit(1);
}
updatePassenger(){
        txtPid=new JTextField(15);          txtAge=new JTextField(15);
        txtEmail=new JTextField(15);      txtPname=newJTextField(15);
        txtMobnu=new JTextField(15);    lblPid=new JLabel("PID");
       lblAge=new JLabel("ÄGE");   lblEmail=new JLabel("EMAIL");
       lblPname=new JLabel("PASSENGER NAME");
```

```
        lblMobnu=new JLabel("MOBILE NUMBER");
        pnl0=new JPanel();   pnl1=new JPanel(); pnl2=new JPanel();
        list=new List(10);
        txtAre2=new JTextArea(20, 100);
       btnUpdate=new JButton("MODIFY");
        setTitle("UPDATE PASSENGER");
        pnl0.setBackground(Color.YELLOW);
        pnl1.setBackground(Color.MAGENTA);
        pnl2.setBackground(Color.BLUE);
        pnl1.setLayout(new GridLayout(5,2));
try {
        class.forName("oracle.jdbc.driver.OracleDriver");
    }
   catch (Exception e) {
        System.err.println("Unable to find and load driver");
        System.exit(1);
  }
  connectToDB();
  try {
        ResultSet  rs = statement.executeQuery("SELECT PID FROM
passenger");
      while (rs.next()){
                    list.add(rs.getString("PID"));
      }
}
catch (SQLException e) {
        System.out.println(e);
}
```

```
list.addItemListener(new ItemListener(){
        public void itemStateChanged(ItemEvent e) {
        try {
                ResultSet rs = statement.executeQuery("SELECT * FROM
passenger where pID ="+list.getSelectedItem());
                rs.next();
                txtPid.setText(rs.getString("PID"));
                txtAge.setText(rs.getString("AGE"));
            txtEmail.setText(rs.getString("EMAILID"));
            txtPname.setText(rs.getString("PNAME"));
             txtMobnu.setText(rs.getString("MOBNU"));
    }
    catch (SQLException selectException) {
            System.out.println(e);
    }
}
});
btnUpdate.addActionListener(new ActionListener(){
       public void actionPerformed(ActionEvent e) {
        try {
                Statement statement = connection.createStatement();
                int i = statement.executeUpdate("UPDATE passenger "
                + "SET age=" + txtAge.getText() + ", "+ "emailid='" +
txtEmail.getText() + "', "+ "pname ='"+ txtPname.getText()
"+"mobnu="+txtMobnu.getText()+"WHERE pid = + list.getSelectedItem());
                txtAre2.append("\nUpdated " + i + " rows successfully");
        }
        catch (SQLException insertException) {
```

```
                System.out.println(e);
        }
        }
});
 pnl0.add(list);
                pnl1.add(lblPid);            pnl1.add(txtPid);
                pnl1.add(lblAge);            pnl1.add(txtAge);
                pnl1.add(lblEmail);          pnl1.add(txtEmail);
                pnl1.add(lblPname);          pnl1.add(txtPname);
                pnl1.add(lblMobnu);          pnl1.add(txtMobnu);
                pnl2.add(btnUpdate);         pnl2.add(txtAre2);
                pnl1.setSize(600,500);       pnl2.setSize(600,500);
                add(pnl0);add(pnl1);add(pnl2);
                setSize(2000,1000);
                setLayout(new GridLayout(3, 1));
                setVisible(true);
                this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        }
}
```

DELETE PASSENGER:

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
public class DeletePassenger extends JFrame {
```

```
private JPanel pnl1,pnl0,pnl2;
private JTextField txtPid,txtAge,txtEmail,txtPname,txtMobnu;
private JLabel lblPid,lblAge,lblEmail, lblPname,lblMobnu;
private List list;
private JTextArea txtAre,txtAre2;
private JButton btnDelete;
Connection connection;
Statement statement;
public void connectToDB() {
  try {

connection=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:
ORCL","dbmsass","dbms");
statement = connection.createStatement();
}
catch (SQLException connectException) {
        System.exit(1);
}
}
DeletePassenger(){
        txtPid=new JTextField(15);       txtAge=new JTextField(15);
        txtEmail=new JTextField(15);    txtPname=new JTextField(15);
        txtMobnu=new JTextField(15);
        lblPid=new JLabel("PID");       lblAge=new JLabel("ÄGE");
        lblEmail=new JLabel("EMAIL");
       lblPname=new JLabel("PASSENGER NAME");
lblMobnu=new JLabel("MOBILE NUMBER");
```

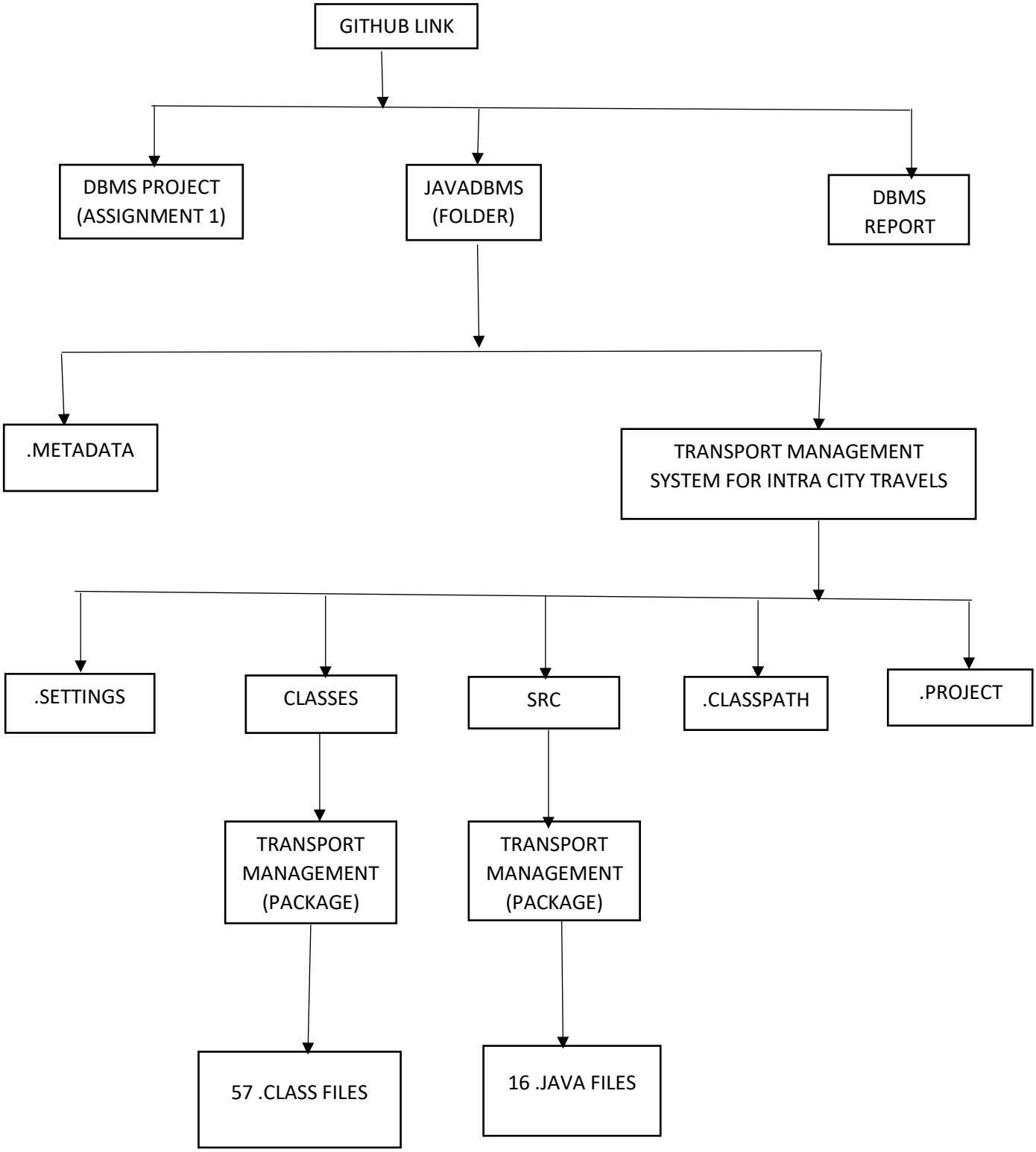
```
pnl0=new JPanel();  pnl1=new JPanel(); pnl2=new JPanel();
list=new List(10);
txtAre=new JTextArea(20,20);   txtAre2=new JTextArea(20, 100);
btnDelete=new JButton("DELETE VALUES");
setTitle("DELETE PASSENGER");
 pnl0.setBackground(Color.CYAN);
pnl1.setBackground(Color.ORANGE);
pnl2.setBackground(Color.PINK);
pnl1.setLayout(new GridLayout(5,2));
try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e) {
        System.err.println("Unable to find and load driver");
        System.exit(1);
}
connectToDB();
try {
        ResultSet  rs = statement.executeQuery("SELECT PID FROM
passenger");
while (rs.next()) {
                        list.add(rs.getString("PID"));
}
}
catch (SQLException e) {
       System.out.println(e);
}
list.addItemListener(new ItemListener(){
```

```
        public void itemStateChanged(ItemEvent e) {
        try {
                ResultSet rs = statement.executeQuery("SELECT * FROM
passenger where pID ="+list.getSelectedItem());
                while (rs.next()) {
                        if (rs.getString("PID").equals(list.getSelectedItem()))
                                        break;
                }
                if (!rs.isAfterLast()) {
                        txtPid.setText(rs.getString("PID"));
                        txtAge.setText(rs.getString("AGE"));
                        txtEmail.setText(rs.getString("EMAILID"));
                        txtPname.setText(rs.getString("PNAME"));
                        txtMobnu.setText(rs.getString("MOBNU"));
                }
}
catch (SQLException selectException) {
                System.out.println(e);
}
});
btnDelete.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent e) {
    try {
         Statement statement = connection.createStatement();
        int i = statement.executeUpdate("DELETE from passenger where pid="
+list.getSelectedItem());
        txtAre2.append("\nDeleted " + i + " rows successfully");
  }
```

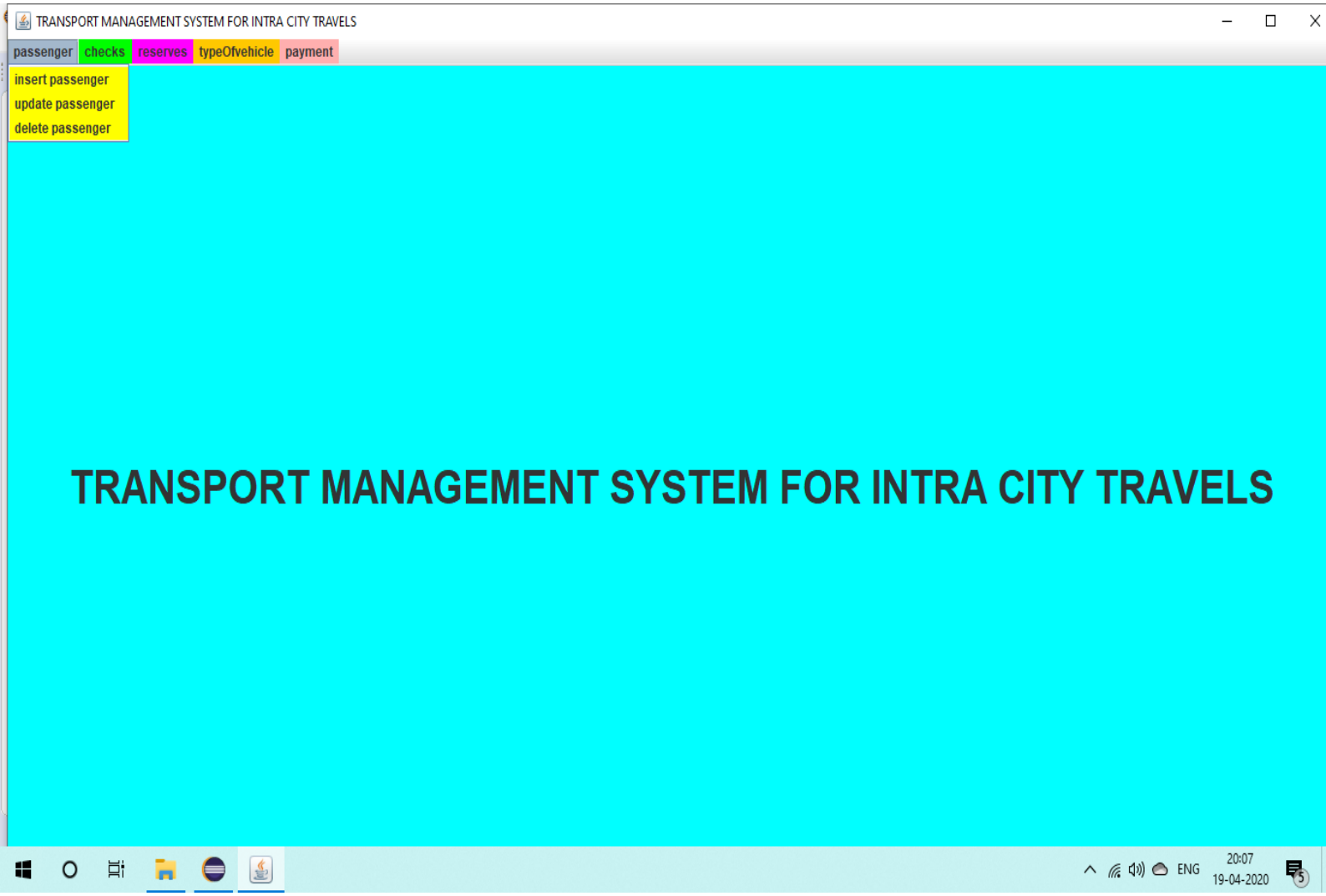
```
catch (SQLException insertException) {
        txtAre2.append(insertException.getMessage());
}
}
});
pnl0.add(list);
pnl1.add(lblPid);              pnl1.add(txtPid);
pnl1.add(lblAge);              pnl1.add(txtAge);
pnl1.add(lblEmail);            pnl1.add(txtEmail);
pnl1.add(lblPname);            pnl1.add(txtPname);
pnl1.add(lblMobnu);            pnl1.add(txtMobnu);
pnl2.add(btnDelete); pnl2.add(txtAre2);
pnl1.setSize(600,500);pnl2.setSize(600,500);
add(pnl0);add(pnl1);add(pnl2);
setSize(2000,1000);
setLayout(new GridLayout(3, 1)); setVisible(true); } }
```

ii .GITHUB LINK AND FOLDER STRUCTURE:

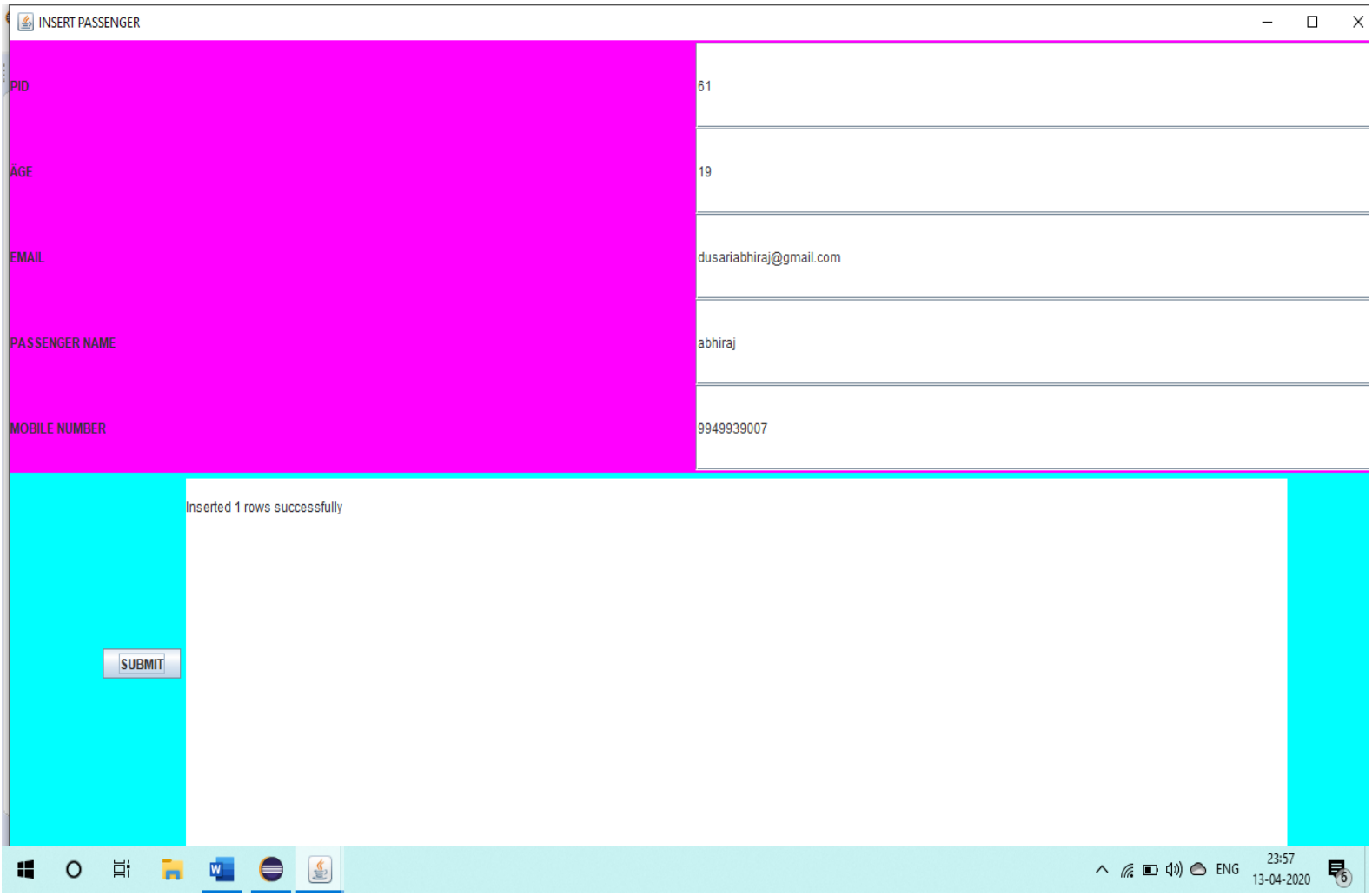
https://github.com/kodipyaka-vinaybabu/DBMS-PROJECT.git



THE FIRST FRAME THAT WILL BE DISPLAYED WHEN WE EXECUTE THE PROGRAM IS



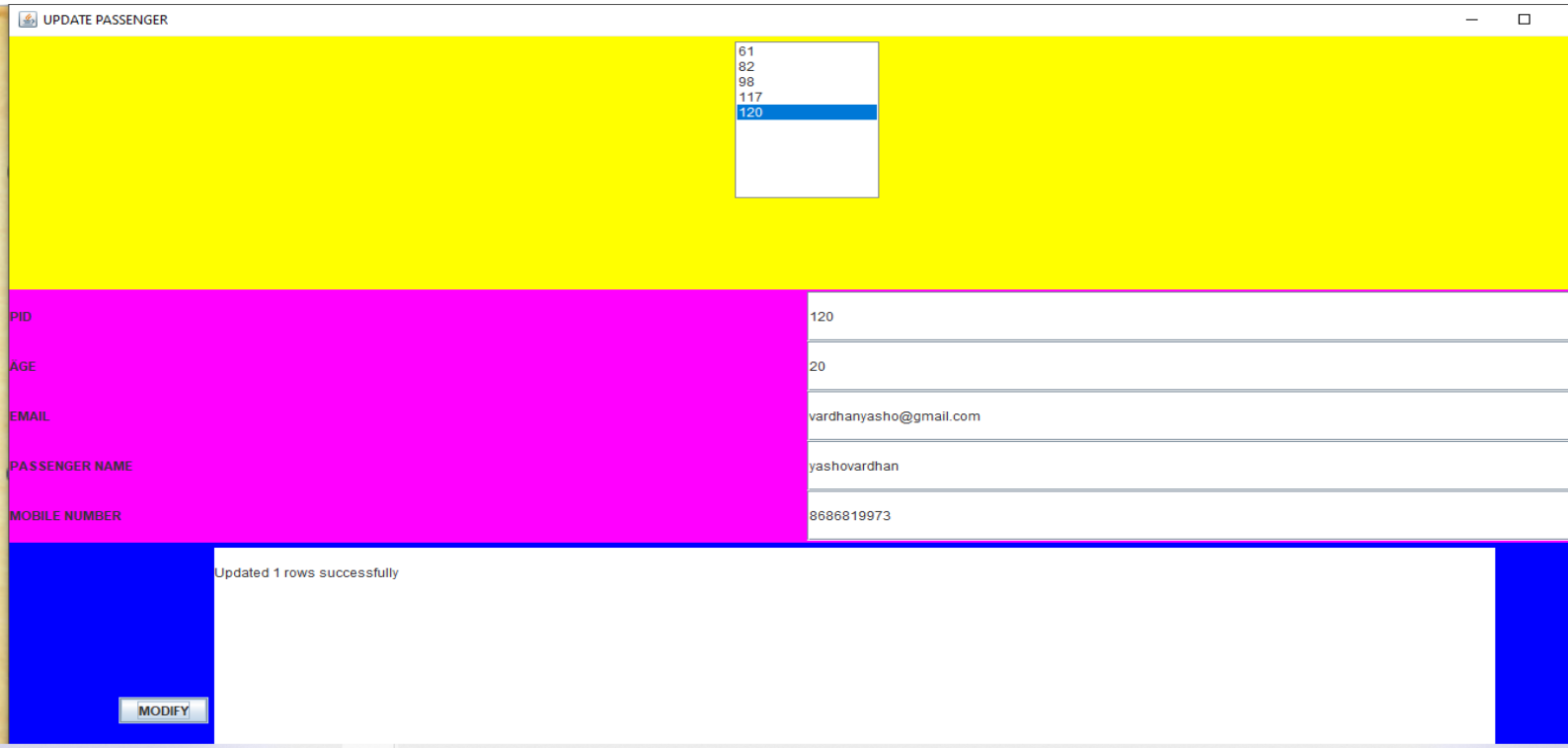
SCREENSHOT OF INSERT PASSENGER



SQL Plus

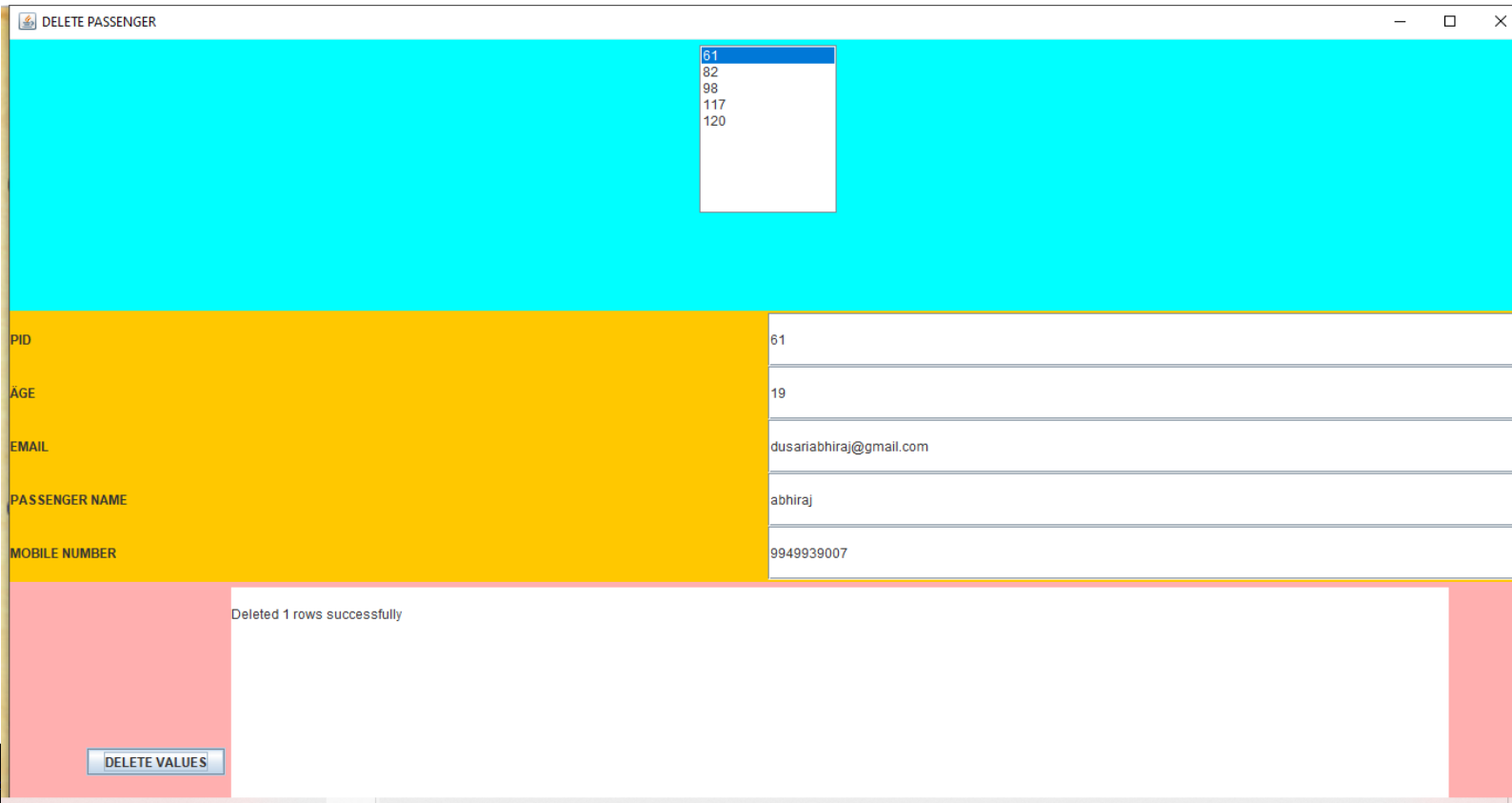
```
SQL> select * from passenger;

       PID        AGE EMAILID
---------- ---------- ----------------------------------------
PNAME                    MOBNU
-------------------- ----------
       117         18 vinaybabu@gmail.com
vinay babu             6303845805

       120         20 vardhanyasho@gmail.com
yasho                  8686819973

        98         19 sheelasai@gmail.com
sairohith              7095716819


       PID        AGE EMAILID
---------- ---------- ----------------------------------------
PNAME                    MOBNU
-------------------- ----------
        82         19 maneethsai1509@gmail.com
maneethsai             9553201509

        61         19 dusariabhiraj@gmail.com
abhiraj                9949939007


SQL>
```

SCREENSHOT OF UPDATE PASSENGER:




```
---------- ---------- ----------------------------------------
PNAME                    MOBNU
-------------------- ----------
       117         18 vinaybabu@gmail.com
vinay babu             6303845805

       120         20 vardhanyasho@gmail.com
yashovardhan           8686819973

        98         19 sheelasai@gmail.com
sairohith              7095716819


       PID        AGE EMAILID
---------- ---------- ----------------------------------------
PNAME                    MOBNU
-------------------- ----------
        82         19 maneethsai1509@gmail.com
maneethsai             9553201509

        61         19 dusariabhiraj@gmail.com
abhiraj                9949939007


SQL>
```

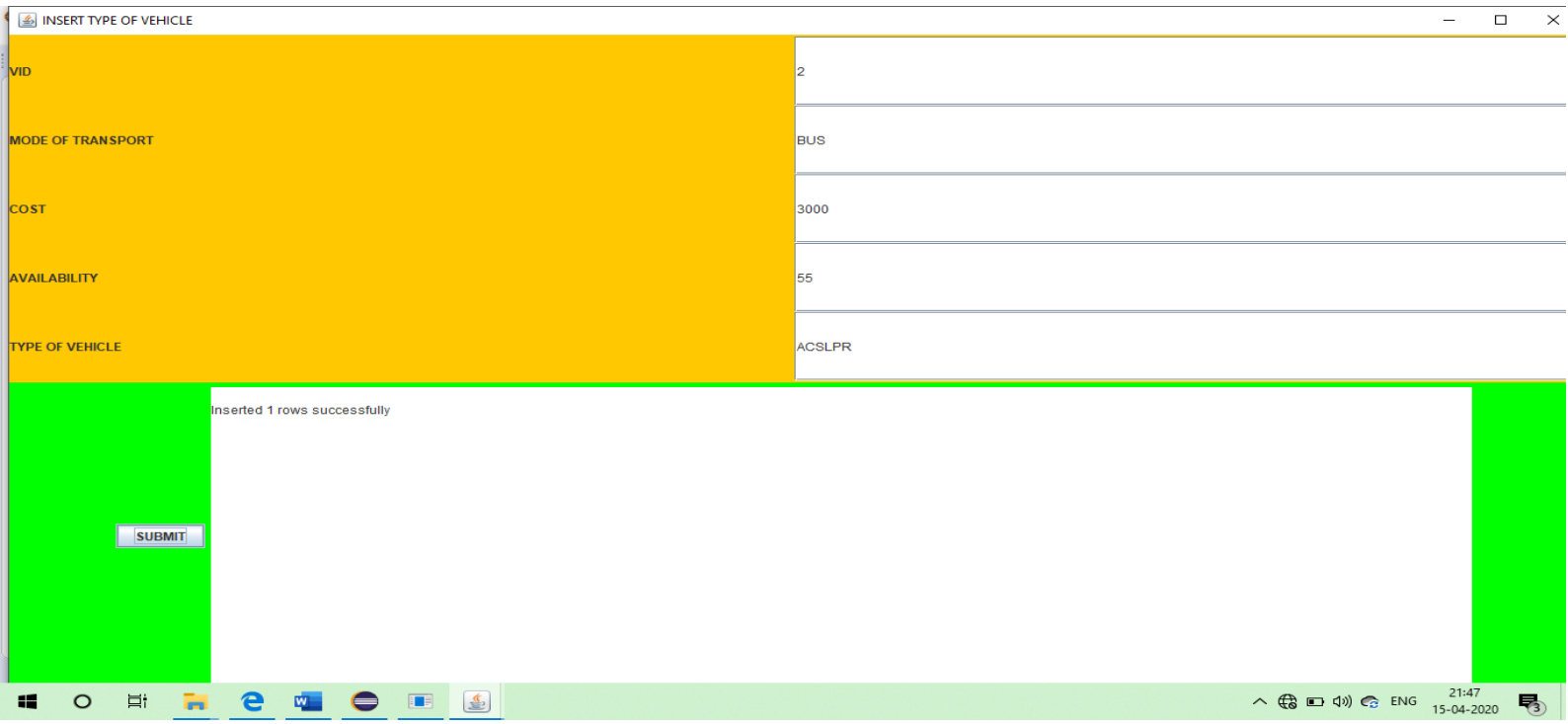
SCREENSHOT OF DELETE PASSENGER

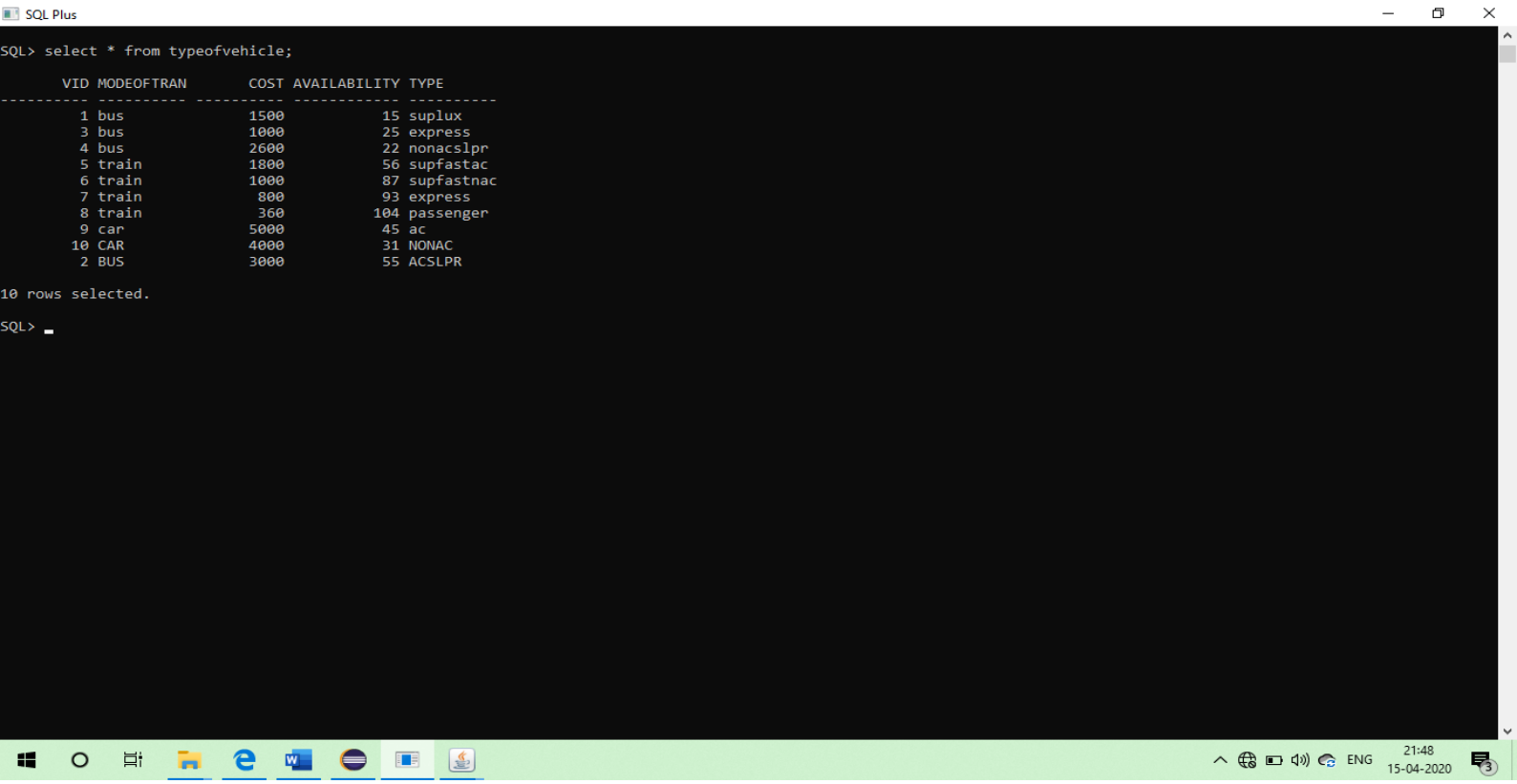



```
PNAME                     MOBNU
-------------------- ----------
       117          18 vinaybabu@gmail.com
vinay babu             6303845805

       120          20 vardhanyasho@gmail.com
yashovardhan           8686819973

        98          19 sheelasai@gmail.com
sairohith              7095716819


       PID        AGE EMAILID
---------- ---------- ----------------------------------------
PNAME                     MOBNU
-------------------- ----------
        82          19 maneethsai1509@gmail.com
maneethsai             9553201509


SQL> _
```

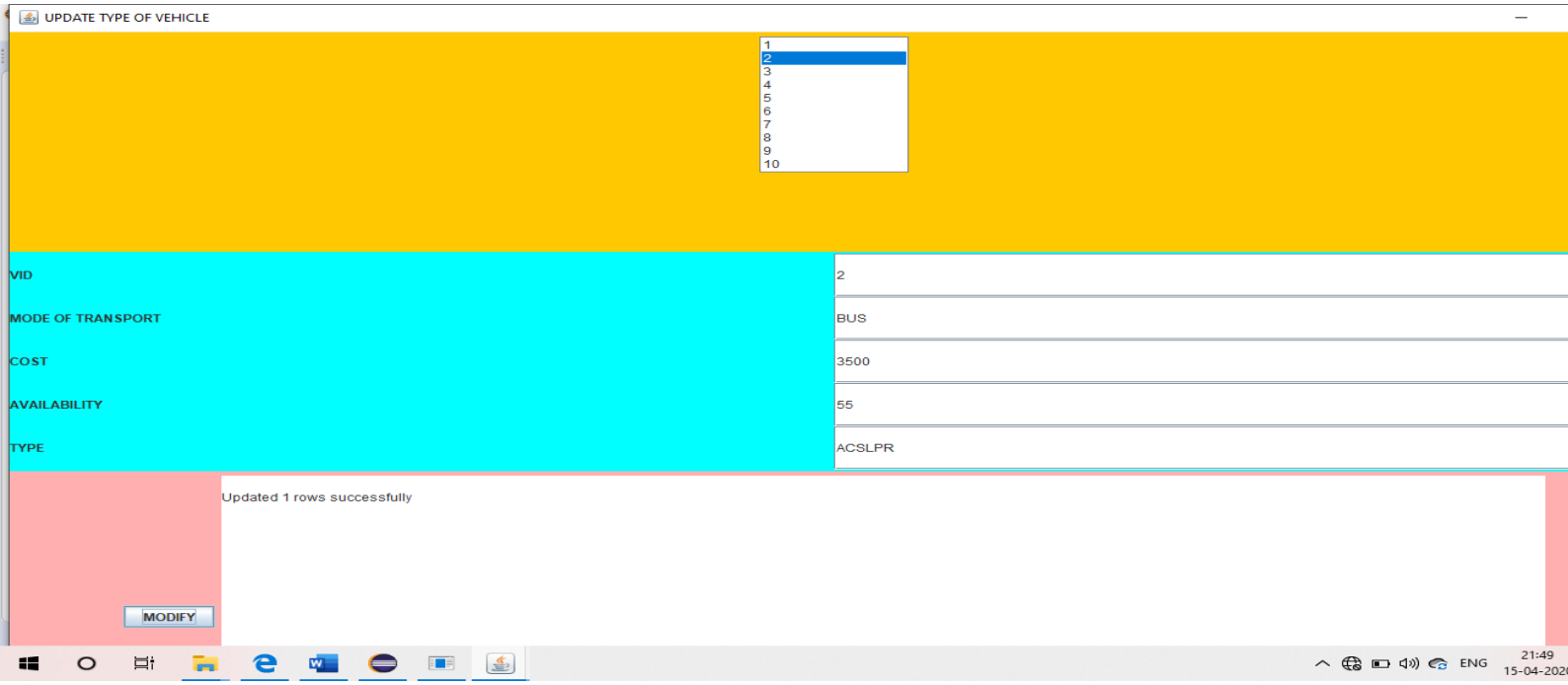
ENG
00:02
14-04-2020

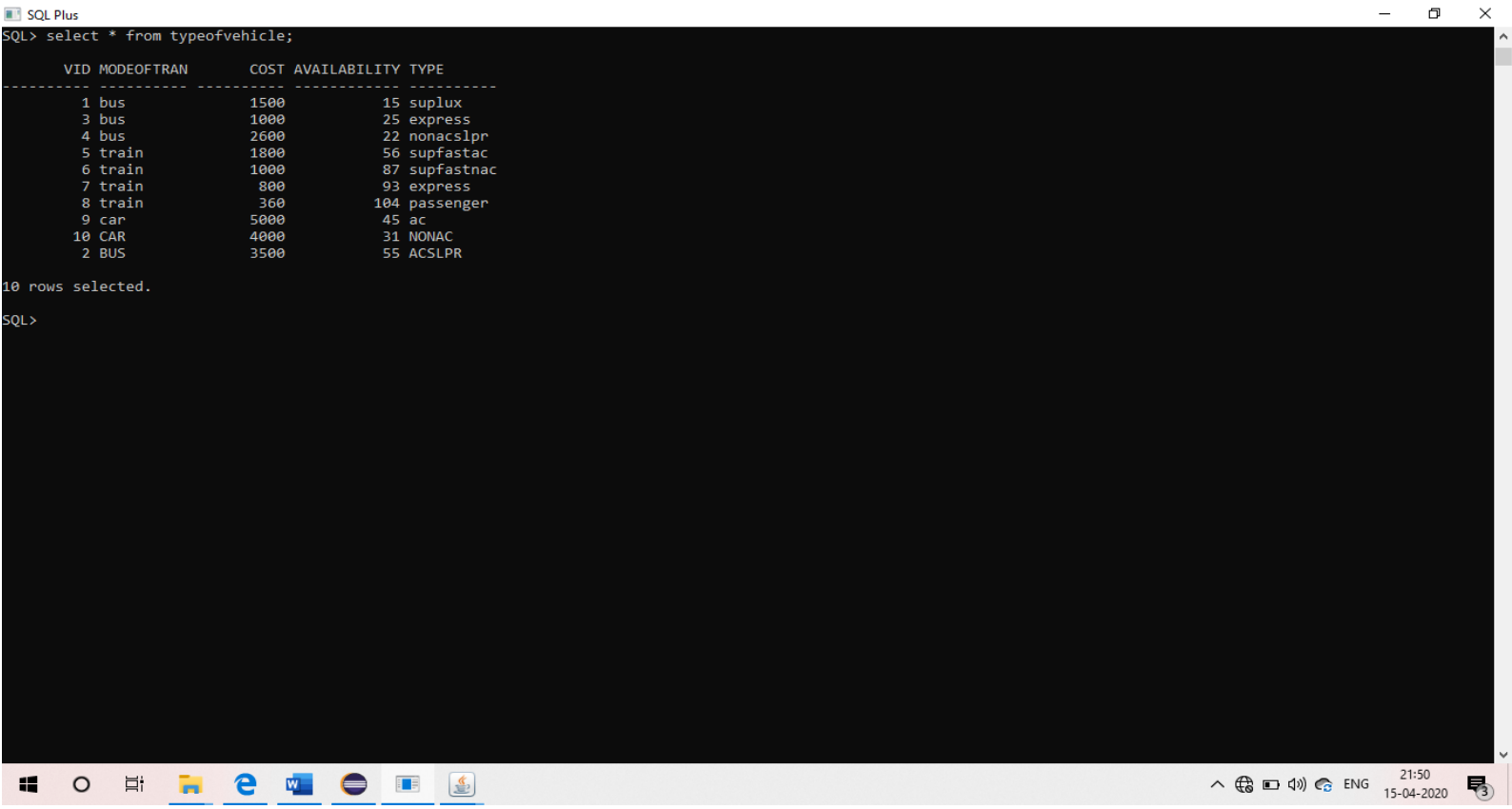
SCREENSHOT OF INSERT TYPEOFVEHICLE



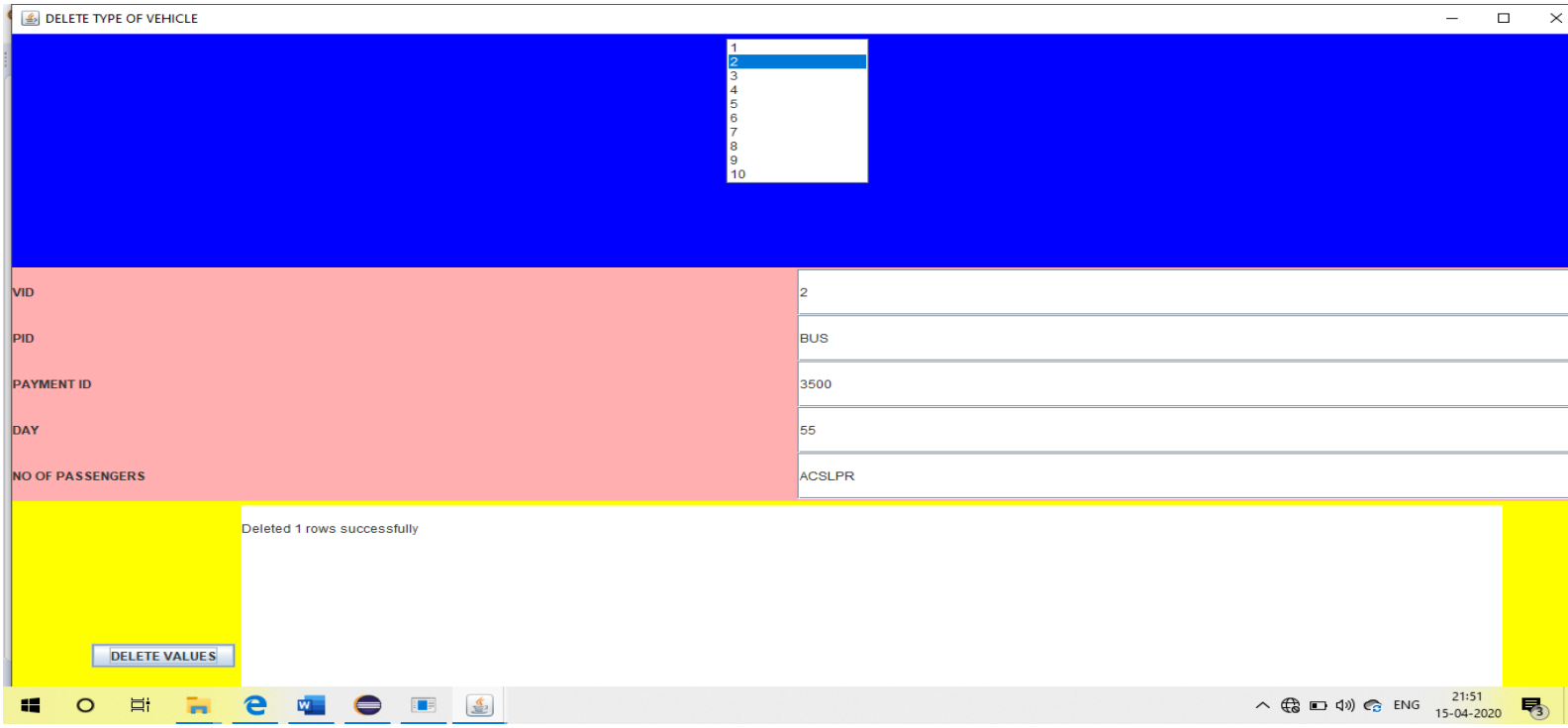


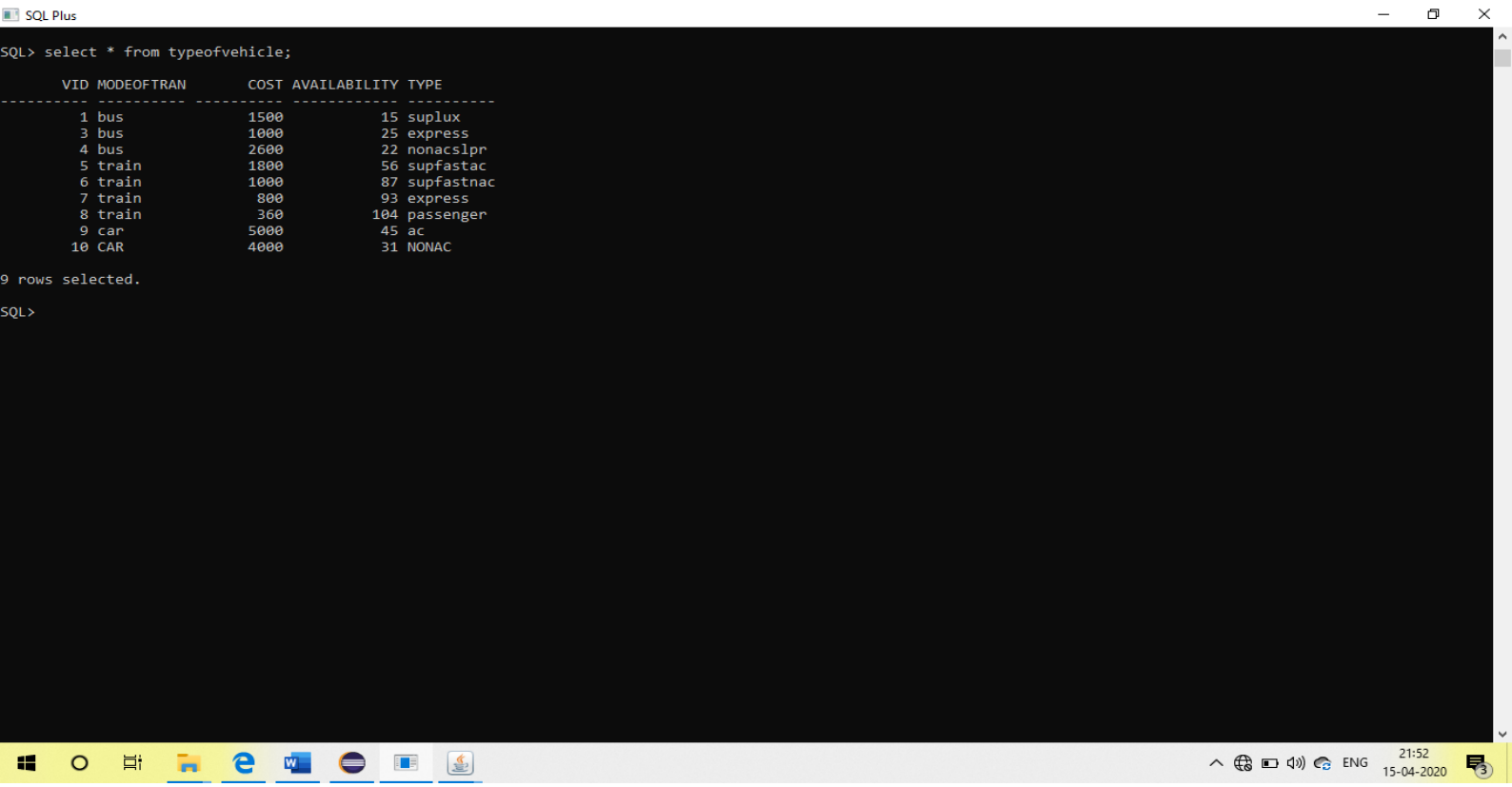
SCREENSHOT OF UPDATE TYPE OF VEHICLE





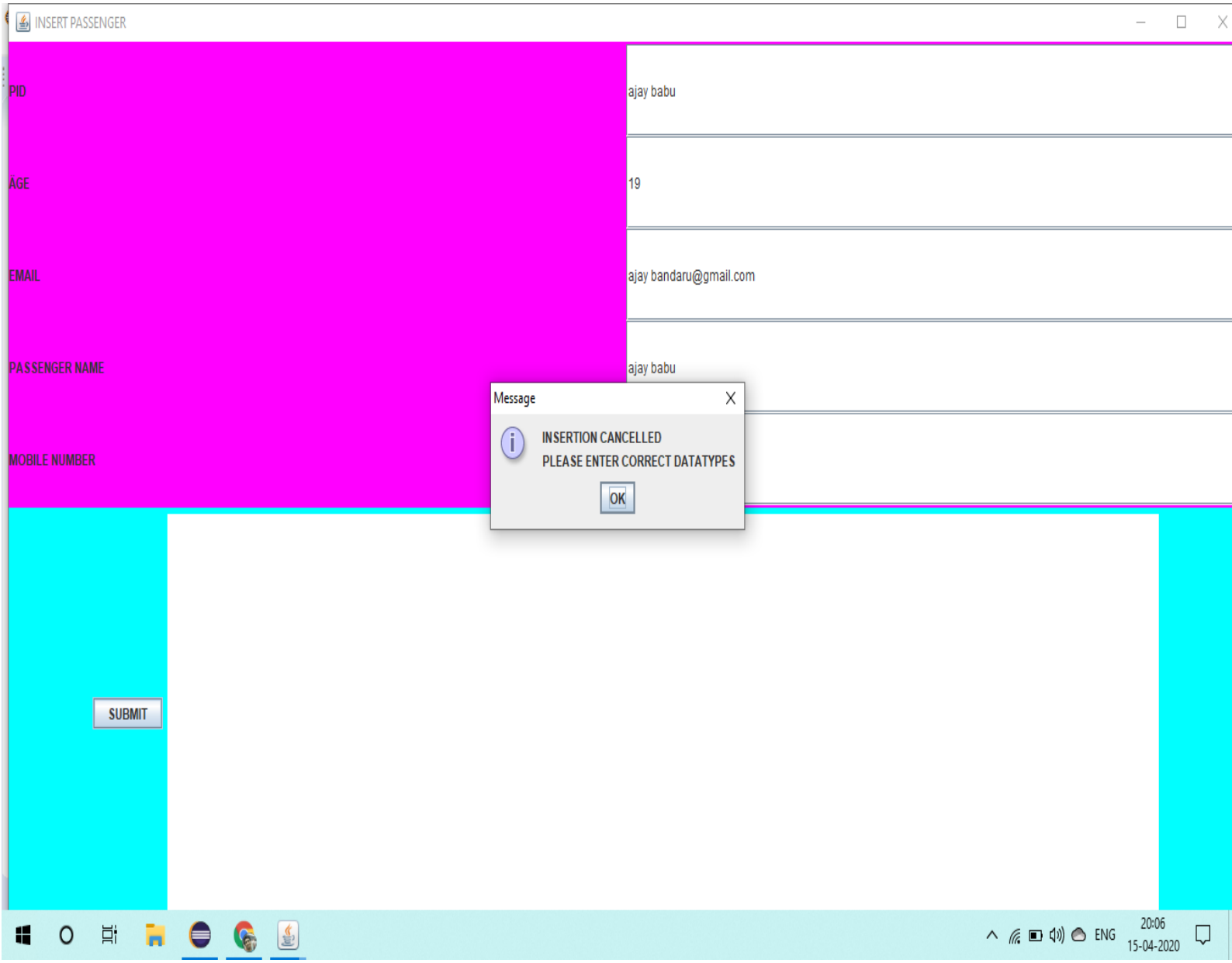
SCREEN SHOT OF DELETE TYPEOFVEHICLE





TESTING:

If a user enters a incorrect datatype the following pop up box will be diplayed!



RESULT: (all screenshots of all dml operations are already kept)

The process of entering information into the frame created by java code so that the data is reflected in the database using **JDBC connectivity** is done successfully.

6.DISCUSSION AND FUTURE WORK!

The application till now done is a basic interface in which a user can enter his details and check the required vehicle based on his budget. **So this project is priorly done for fixed place** i.e fixed start place and fixed destination place. So **in future the project will be edited in such a manner that the fare is calculated based the number of kilometers between two places.**

7.REFERENCES:

https://docs.oracle.com/javase/8/docs/api/

https://www.geeksforgeeks.org/establishing-jdbc-connection-in-java/

https://www.javatpoint.com/java-awt