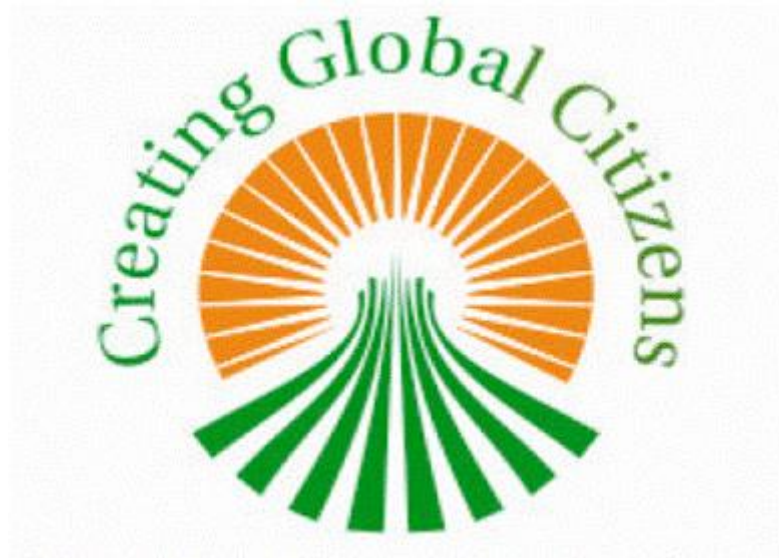# City International School

## Wanowrie, Pune
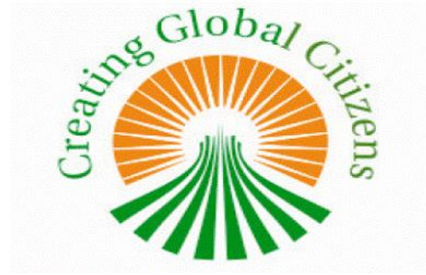


# Information Practices Project

## 2014-2015

### Kshiteej Gilda

Class XII B

Roll No:

# City International School
## Wanowrie, Pune

# **<u>CERTIFICATE</u>**

*This is to certify that Master <u>Kshiteej Gilda</u> of Class <u>XII B</u>, has satisfactorily completed his project work and submitted the project report in given time.*

Internal Examiner:                                        External Examiner:

Principal:                                                        School Stamp:

# Acknowledgement

I would like to like to express my sincere gratitude to my teacher Mrs Rashmi Saxena and my Principal Mrs Vinita Khaladkar for giving me the opportunity to do this project. It allowed me to do a lot of research and gather a lot of knowledge.

I would also like to thank my parents and friends who helped me to successfully make this project in the limited time.

# Introduction

The bus ticketing software offers an interesting solution to modern day bus operator's problems. With their size of bus fleets close to hundred today, it has become very difficult to only rely on manual register entries method. Today customers do most of their tasks online and also expect punctual services and genuine tickets. The bus industry is plagued by issues wherein users are sold tickets of ill maintained aging buses in the name of high quality buses. Although bus operators have now started providing high quality buses like multi - axle Volvo B9R and Scania Metro link, the customers still shy away from preferring bus transport and take up other alternatives due to lack of a transparent process while issuing bus tickets.

In this scenario the bus ticketing software comes to our great aid by providing the following advantages –

- A bus operator can manage all of the ticketing process occurring throughout the country by just using his computer and thus saving unnecessary expenses.
- It ensures that the bus tickets are sold only at the prices fixed by the bus operators and thus prevents the rampant practice of taking commissions by the travel agents.
- The customer can book the tickets from the comfort of his home and pay the bus ticket price using online banking. Nowadays some online sites like redbus.in even offer the facility of tickets being directly sent to your mobile phones in the form of sms and thus saving you the burden of printing the ticket.
- The bus operators can expand their business by venturing into new cities where previously they were unable to find a footing due to lack of manpower. Now the bus operators can just advertise about their service and thus attract new customers.
- By using computers less manpower is now required, thus resulting in more profits.
- Although the initial investment cost for the software is high in future it turns out to be beneficial and in the end profitable for the operator.

To provide an even more enriching experience to the customers the following upgrades can be made to the software:

- Development of apps for mobile devices like Android and IOS.
- Fixing trackers in the buses and having a tool in the software which provides the customers the ability to real time track the progress of buses.
- The system can also have the facility of sending sms to the customer about any route change in the journey of the bus due to any road blockages or due to any other reason.
- Speed limiters can be fixed to the buses fixing the top speed of the buses in the range of 100 – 120 km/hr and the instantaneous speed can be communicated to the software thus ensuring that the driver does not over speed.
- The system can also help in ensuring that the quality of the bus fleet is maintained by keeping a log of scheduled services and daily costs.
  Eg : Volvo provides the above mentioned facility  and this can be integrated with the software.
- While the customer is waiting at the bus stop the software can provide him with the real time status of the bus.

# System Requirements

- Windows XP Professional
- Intel Pentium 4 (or equivalent) CPU running at 1.4GHz or higher
- 32MB DirectX 9.0c compatible video card (Geforce2 MX or higher) using a supported chipset (see below)
- 256 MB RAM
- DirectX compatible sound card
- 100 MB of hard disk space
- 8X CD-ROM

# javaconnect.java (class)

**This class contains the connectivity code for the whole application and is called on wherever the code is required.**

**Code:**

```java
import java.sql.*;

import javax.swing.*;/*

public class javaconnect {

   Connection conn = null;

    public static Connection ConnecrDb(){

       try{

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

     Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/busticketing","root","123");

     return conn;

     } catch(Exception e)

{JOptionPane.showMessageDialog(null,e);

 return null;

}

}}
```
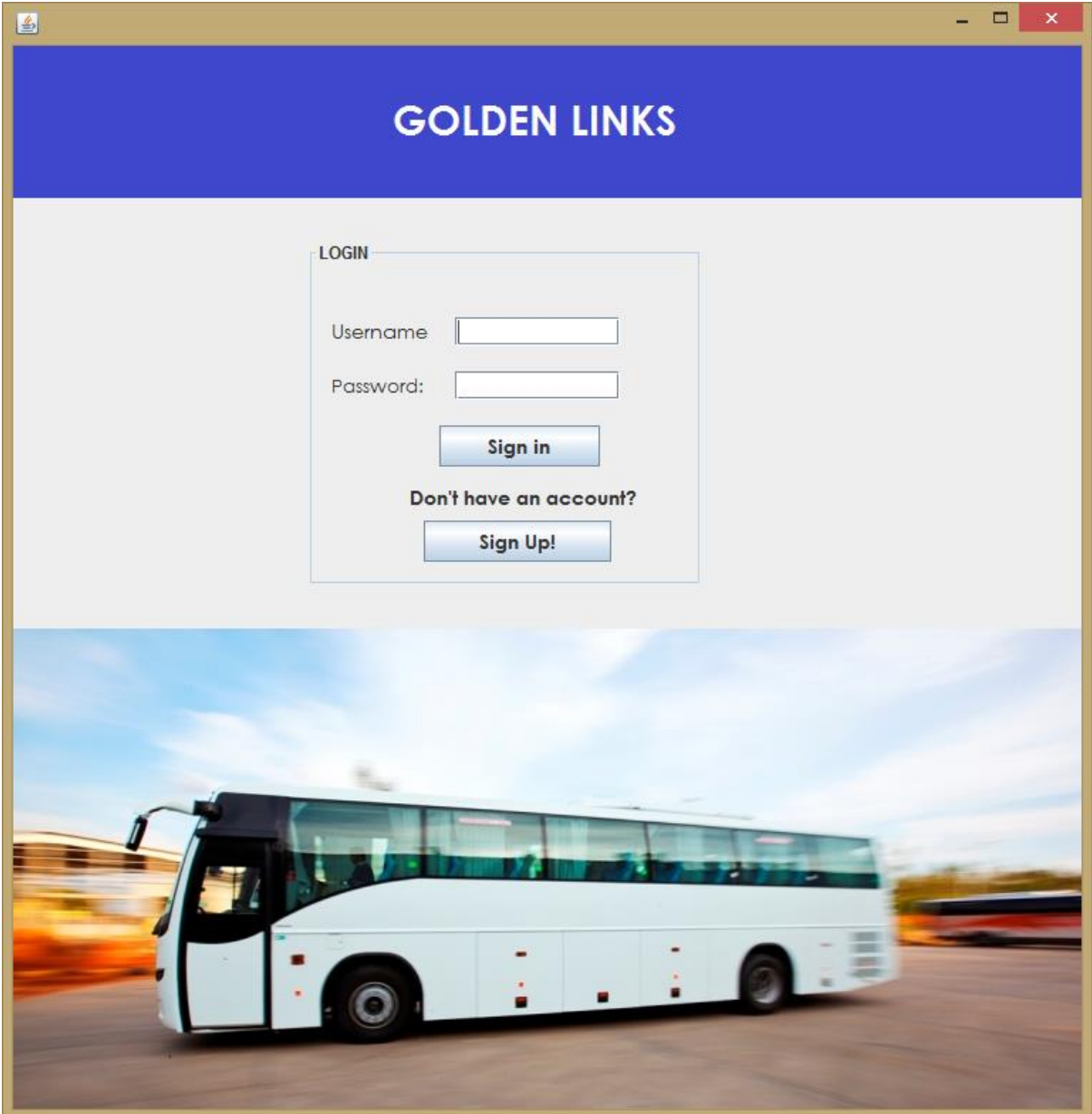
# GOLDEN LINKS

### LOGIN

Username

Password:

**Sign in**

**Don't have an account?**

**Sign Up!**

# Page1_login

**Source Code:**

```java
import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.sql.*;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.*;

import net.proteanit.sql.DbUtils;

import javax.swing.table.DefaultTableModel;

import javax.swing.JOptionPane;

public class Page1_login extends javax.swing.JFrame {

    Connection conn=null;

    Statement pst=null;

    String user, password;

    /** Creates new form Page1_login */

    public Page1_login() {

        conn = javaconnect.ConnecrDb();

        initComponents();

    }

public void close()

    {

        WindowEvent winClosingEvent = new WindowEvent(this,WindowEvent.WINDOW_CLOSING);

        Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);

    }
```

**Code for sign in button :**

```java
private void signinBTNActionPerformed(java.awt.event.ActionEvent evt) {

user = username.getText();

    password = passwrd.getText();
```
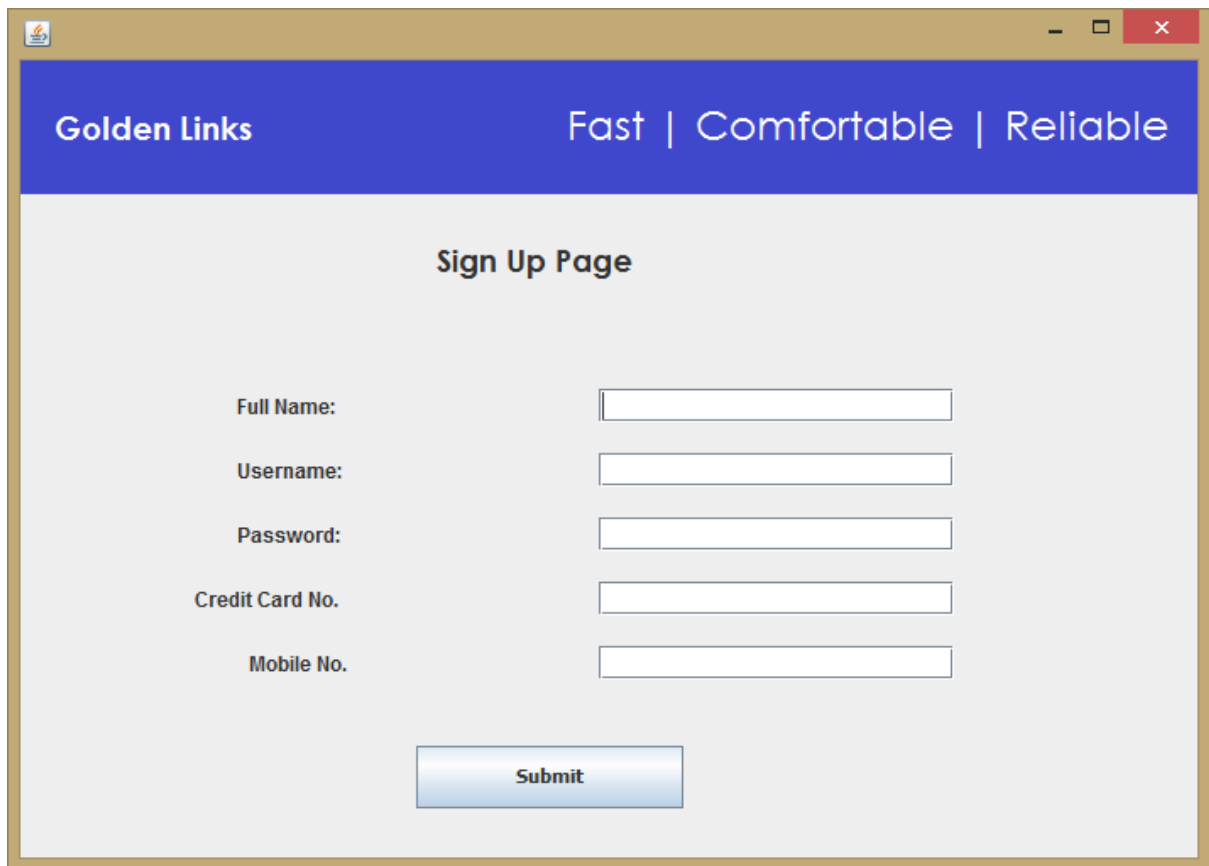
```java
 String sql = "select * from users where username = '"+user+"';";//if u want only password code is:
select password...not select 'password'...

    try{
 pst = conn.createStatement();

ResultSet rs=pst.executeQuery(sql);

if(rs.next()){//this outer if is in case username is incorrect

String p = rs.getString("Password");

if(password.equals(p))

{

Pg2_Welcome s = new Pg2_Welcome();

s.username = user;

close();

s.setVisible(true);

}

else

JOptionPane.showMessageDialog(null, "Please enter valid Username and Password");

    }//this closes inner else

else

{JOptionPane.showMessageDialog(null, "Please enter valid Username and Password");

}

}

    catch(Exception e){

    JOptionPane.showMessageDialog(null, e);

    }
```

**Code for Sign up button:**

```java
private void signupBTNActionPerformed(java.awt.event.ActionEvent evt) {

close();

Page1_SignUp s = new Page1_SignUp();

s.setVisible(true);

  }
```

**Golden Links**                Fast | Comfortable | Reliable

## Sign Up Page

Full Name:          _____

Username:           _____

Password:           _____

Credit Card No.     _____

Mobile No.          _____

[ **Submit** ]

Page1_Signup

**Source Code:**

```java
import com.mysql.jdbc.Connection;

import com.mysql.jdbc.Statement;

import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.sql.SQLException;

import javax.swing.JOptionPane;

public class Page1_SignUp extends javax.swing.JFrame {

    Connection conn=null;

   Statement pst=null;

   public Page1_SignUp() {

      conn = (Connection) javaconnect.ConnecrDb();

      initComponents();

   }

public void close()

   {

      WindowEvent winClosingEvent = new WindowEvent(this,WindowEvent.WINDOW_CLOSING);

      Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);

   }
```

**Code for Submit Button:**

```java
private void submitBTNActionPerformed(java.awt.event.ActionEvent evt) {

if(nameTF.getText().equals("")||usernameTF.getText().equals("")||passwordTF.getText().equals(""))

   {JOptionPane.showMessageDialog(null, "Please enter the required information");}

   else

   {

   name = nameTF.getText();

   username = usernameTF.getText();

   password = passwordTF.getText();

   ccno = Integer.parseInt(ccnoTF.getText());

   /*Now to save this info in users table*/
```

```java
    try {

    String sql = "insert into users values('"+name+"','"+ccno+"','"+username+"','"+password+"')";

    pst = (Statement) conn.createStatement();

        pst.execute(sql);

    } catch (SQLException e) {

        JOptionPane.showMessageDialog(null, e);//Done. Now to create new table to store the
users bbooking detils

    }

    try{

    String sql = "create table "+username+" (Name char(50), Route char(50), BusNo char(5), Date
char(15));";

    pst = (Statement) conn.createStatement();

        pst.execute(sql);

    } catch (SQLException e) {

        JOptionPane.showMessageDialog(null, e);

    }

    obj.user=username;
/*Now to call the next page*/
        close();
Pg2_Welcome s = new Pg2_Welcome();
s.username=username;
s.setVisible(true);
    }
  }
```

# Pg2_Welcome

**Source Code:**

```java
import java.awt.Toolkit;

import java.awt.event.WindowEvent;

public class Pg2_Welcome extends javax.swing.JFrame {

  String username, h;

   /** Creates new form Pg2_Welcome */

public void close()

  {

     WindowEvent winClosingEvent = new WindowEvent(this,WindowEvent.WINDOW_CLOSING);

     Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);

  }
```
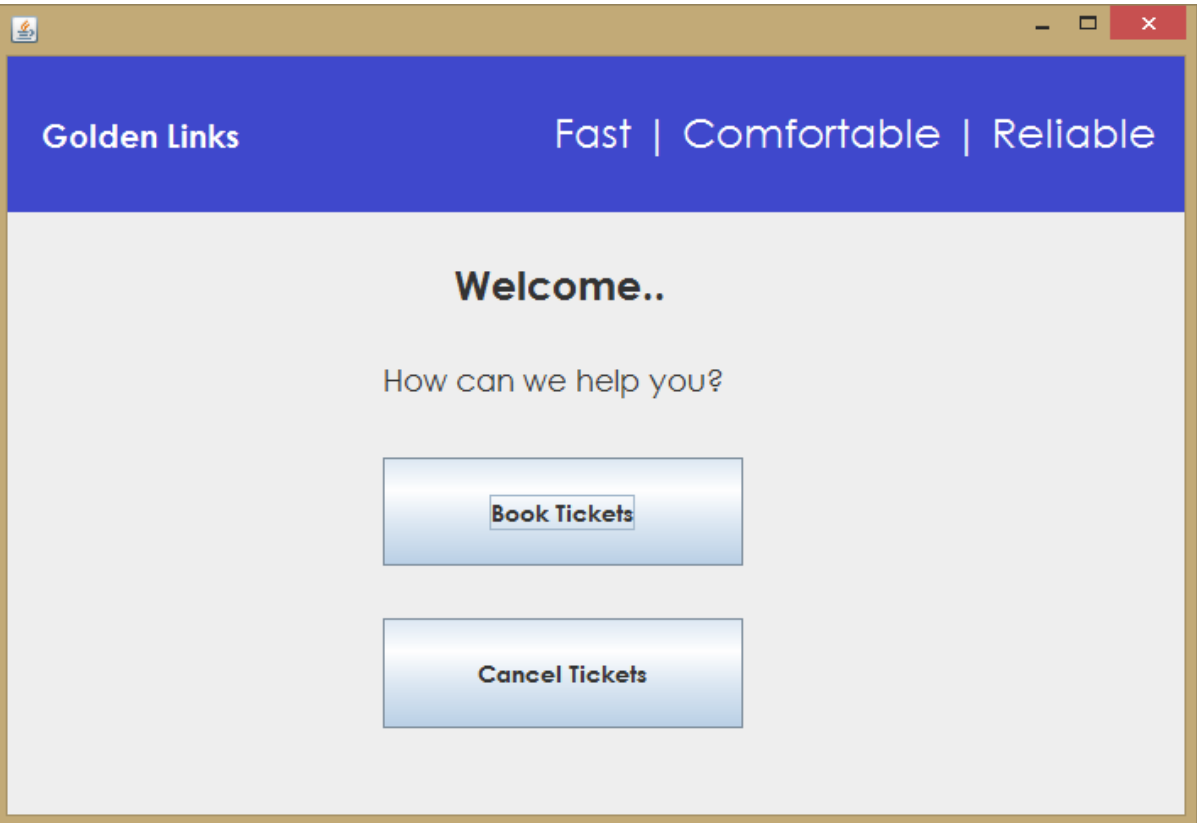
**Code for Book Tickets button:**

```java
private void BookingRedirectBTNActionPerformed(java.awt.event.ActionEvent evt) {

  close();

Booking s = new Booking();

s.username=username;

s.setVisible(true);

}
```

**Code for Cancel Tickets button:**

```java
private void CancelingRedirectBTNActionPerformed(java.awt.event.ActionEvent evt) {

close();

CancelTickets s = new CancelTickets();

s.username=username;

s.setVisible(true);

}
```

Fast | Comfortable | Reliable

# Go ahead! Book your tickets!

Route: **Mumbai to Delhi** ▼

Date: 2014-09-10 🗓

| # | Bus Name | AC | Price (in Rs) |
|---|----------|-----|---------------|
| M2D1 | Volvo B9R | AC | 999 |
| M2D2 | Tata 450V | NONAC | 575 |

**Check Seat Availability**

*Enter Passenger Details Here:*

**Add Passenger**

**Note: Seat Number will be entered autom...**

| Name | Seat No. |
|------|----------|
| g | 31 |

*Select your favourite seats:*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 40 | 36 | 32 | 28 | 24 | 20 | 16 | 12 | 8 | 4 |
| 39 | 35 | 31 | 27 | 23 | 19 | 15 | 11 | 7 | 3 |
| 38 | 34 | 30 | 26 | 22 | 18 | 14 | 10 | 6 | 2 |
| 37 | 33 | 29 | 25 | 21 | 17 | 13 | 9 | 5 | 1 |

**Pay Rs999 and Book Now!**　　　　**BACK TO MENU**

# Booking

**Source Code:**

```java
import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.sql.*;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.*;

import net.proteanit.sql.DbUtils;

import javax.swing.table.DefaultTableModel;

public class Booking extends javax.swing.JFrame {

    int S1roundno=1;

    int S2roundno=1;

    int S3roundno=1;

    int S4roundno=1;

    int S5roundno=1;

    int S6roundno=1;

    int S7roundno=1;

    int S8roundno=1;

    int S9roundno=1;

    int S10roundno=1;

    int S11roundno=1;

    int S12roundno=1;

    int S13roundno=1;

    int S14roundno=1;

    int S15roundno=1;

    int S16roundno=1;

    int S17roundno=1;
```

```java
        int S18roundno=1;

        int S19roundno=1;

        int S20roundno=1;

        int S21roundno=1;

        int S22roundno=1;

        int S23roundno=1;

        int S24roundno=1;

        int S25roundno=1;

        int S26roundno=1;

        int S27roundno=1;

        int S28roundno=1;

        int S29roundno=1;

        int S30roundno=1;

        int S31roundno=1;

        int S32roundno=1;

        int S33roundno=1;

        int S34roundno=1;

        int S35roundno=1;

        int S36roundno=1;

        int S37roundno=1;

        int S38roundno=1;

        int S39roundno=1;

        int S40roundno=1;

        String date, username,route;

        Object[][] busestabledata =null;

        String selbus;

        int price;

         Connection conn=null;
//    ResultSet rs=null;
      Statement pst=null;
      int i = 0, selno=0;
```

```java
   Object[][] data =null;

      public Booking() {

      initComponents();

     setSize(720,700);

      conn=javaconnect.ConnecrDb();

 }

public void close()

  {

     WindowEvent winClosingEvent = new WindowEvent(this,WindowEvent.WINDOW_CLOSING);

     Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);

  }
```

**Code for CBroute combobox:**

```java
private void CBrouteActionPerformed(java.awt.event.ActionEvent evt) {

  route = (String) CBroute.getSelectedItem();

  try{

    // System.out.println("error step");

   String sql = "select * from buses where Route='"+route+"';";

    // System.out.println("");

  pst = conn.createStatement();

ResultSet rs=pst.executeQuery(sql);

  int j=0;

  int i=0;

  while(rs.next()==true)

    {

       //System.out.println("execute1");

       j=0;

       String un=rs.getString("BusNo");

       String ps=rs.getString("BusName");

        String ps2=rs.getString("AC");

      BusesTable.setValueAt(un,i,j);

       j++;
```

```java
        BusesTable.setValueAt(ps,i,j);

        j++;

        BusesTable.setValueAt(ps2,i,j);

        i++;

    //   System.out.println(un+" "+ps);

    }

  BusesTable.setValueAt("999", 0, 3);

  BusesTable.setValueAt("575", 1, 3);

  // .setModel(DbUtils.resultSetToTableModel(rs));

}

  catch(Exception e){

  JOptionPane.showMessageDialog(null, e);

}
```

**Code for check seat availability button:**

```java
private void seatavBTActionPerformed(java.awt.event.ActionEvent evt) {

 date = ((JTextField)dateDC.getDateEditor().getUiComponent()).getText();

  if(date.equals("")||BusesTable.getSelectedRow()==-1)

 {JOptionPane.showMessageDialog(null, "Date or Bus hasn't been selected.");}

  else{

 selbus = (String) BusesTable.getValueAt(BusesTable.getSelectedRow(), 0);

 String sql = "select * from "+selbus+" where date = '"+date+"';";

 try{

 pst = conn.createStatement();

 ResultSet rs=pst.executeQuery(sql);

 TBSeat1.setSelected(false); TBSeat1.setEnabled(true);

 TBSeat2.setSelected(false); TBSeat2.setEnabled(true);

 TBSeat3.setSelected(false); TBSeat3.setEnabled(true);

 TBSeat4.setSelected(false); TBSeat4.setEnabled(true);

 TBSeat5.setSelected(false); TBSeat5.setEnabled(true);

 TBSeat6.setSelected(false); TBSeat6.setEnabled(true);

 TBSeat7.setSelected(false); TBSeat7.setEnabled(true);
```

```
TBSeat8.setSelected(false); TBSeat8.setEnabled(true);

TBSeat9.setSelected(false); TBSeat9.setEnabled(true);

TBSeat10.setSelected(false); TBSeat10.setEnabled(true);

TBSeat11.setSelected(false); TBSeat11.setEnabled(true);

TBSeat12.setSelected(false); TBSeat12.setEnabled(true);

TBSeat13.setSelected(false); TBSeat13.setEnabled(true);

TBSeat14.setSelected(false); TBSeat14.setEnabled(true);

TBSeat15.setSelected(false); TBSeat15.setEnabled(true);

TBSeat16.setSelected(false); TBSeat16.setEnabled(true);

TBSeat17.setSelected(false); TBSeat17.setEnabled(true);

TBSeat18.setSelected(false); TBSeat18.setEnabled(true);

TBSeat19.setSelected(false); TBSeat19.setEnabled(true);

TBSeat20.setSelected(false); TBSeat20.setEnabled(true);

TBSeat21.setSelected(false); TBSeat21.setEnabled(true);

TBSeat22.setSelected(false); TBSeat22.setEnabled(true);

TBSeat23.setSelected(false); TBSeat23.setEnabled(true);

TBSeat24.setSelected(false); TBSeat24.setEnabled(true);

TBSeat25.setSelected(false); TBSeat25.setEnabled(true);

TBSeat26.setSelected(false); TBSeat26.setEnabled(true);

TBSeat27.setSelected(false); TBSeat27.setEnabled(true);

TBSeat28.setSelected(false); TBSeat28.setEnabled(true);

TBSeat29.setSelected(false); TBSeat29.setEnabled(true);

TBSeat30.setSelected(false); TBSeat30.setEnabled(true);

TBSeat31.setSelected(false); TBSeat31.setEnabled(true);

TBSeat32.setSelected(false); TBSeat32.setEnabled(true);

TBSeat33.setSelected(false); TBSeat33.setEnabled(true);

TBSeat34.setSelected(false); TBSeat34.setEnabled(true);

TBSeat35.setSelected(false); TBSeat35.setEnabled(true);

TBSeat36.setSelected(false); TBSeat36.setEnabled(true);

TBSeat37.setSelected(false); TBSeat37.setEnabled(true);

TBSeat38.setSelected(false); TBSeat38.setEnabled(true);
```

```java
TBSeat39.setSelected(false); TBSeat39.setEnabled(true);
TBSeat40.setSelected(false); TBSeat40.setEnabled(true);
    while(rs.next()){
  int sn = rs.getInt("SeatNo");
   if(sn==1)
  {TBSeat1.setSelected(true); TBSeat1.setEnabled(false);}
   if(sn==2)
  {TBSeat2.setSelected(true); TBSeat2.setEnabled(false);}
  if(sn==3)
  {TBSeat3.setFocusable(false); TBSeat3.setEnabled(false);}
  if(sn==4)
  {TBSeat4.setFocusable(false); TBSeat4.setEnabled(false);}
   if(sn==5)
  {TBSeat5.setFocusable(false); TBSeat5.setEnabled(false);}
  if(sn==6)
  {TBSeat6.setFocusable(false); TBSeat6.setEnabled(false);}
   if(sn==7)
  {TBSeat7.setFocusable(false); TBSeat7.setEnabled(false);}
  if(sn==8)
  {TBSeat8.setFocusable(false); TBSeat8.setEnabled(false);}
   if(sn==9)
  {TBSeat9.setFocusable(false); TBSeat9.setEnabled(false);}
   if(sn==10)
  {TBSeat10.setFocusable(false); TBSeat10.setEnabled(false);}
   if(sn==11)
  {TBSeat11.setFocusable(false); TBSeat11.setEnabled(false);}
   if(sn==12)
  {TBSeat12.setFocusable(false); TBSeat12.setEnabled(false);}
  if(sn==13)
  {TBSeat13.setFocusable(false); TBSeat13.setEnabled(false);}
  if(sn==14)
```

```java
{TBSeat14.setFocusable(false); TBSeat14.setEnabled(false);}
 if(sn==15)
{TBSeat15.setFocusable(false); TBSeat15.setEnabled(false);}
if(sn==16)
{TBSeat16.setFocusable(false); TBSeat16.setEnabled(false);}
 if(sn==17)
{TBSeat17.setFocusable(false); TBSeat17.setEnabled(false);}
if(sn==18)
{TBSeat18.setFocusable(false); TBSeat18.setEnabled(false);}
 if(sn==19)
{TBSeat19.setFocusable(false); TBSeat19.setEnabled(false);}
 if(sn==20)
{TBSeat20.setFocusable(false); TBSeat20.setEnabled(false);}
 if(sn==21)
{TBSeat21.setFocusable(false); TBSeat21.setEnabled(false);}
 if(sn==22)
{TBSeat22.setFocusable(false); TBSeat22.setEnabled(false);}
if(sn==23)
{TBSeat23.setFocusable(false); TBSeat23.setEnabled(false);}
 if(sn==23)
{TBSeat23.setFocusable(false); TBSeat23.setEnabled(false);}
if(sn==24)
{TBSeat24.setFocusable(false); TBSeat24.setEnabled(false);}
 if(sn==25)
{TBSeat25.setFocusable(false); TBSeat25.setEnabled(false);}
if(sn==26)
{TBSeat26.setFocusable(false); TBSeat26.setEnabled(false);}
 if(sn==27)
{TBSeat27.setFocusable(false); TBSeat27.setEnabled(false);}
if(sn==28)
{TBSeat28.setFocusable(false); TBSeat28.setEnabled(false);}
```

```java
 if(sn==29)
{TBSeat29.setFocusable(false); TBSeat29.setEnabled(false);}
 if(sn==30)
{TBSeat30.setFocusable(false); TBSeat30.setEnabled(false);}
 if(sn==31)
{TBSeat31.setFocusable(false); TBSeat31.setEnabled(false);}
 if(sn==32)
{TBSeat32.setFocusable(false); TBSeat32.setEnabled(false);}
if(sn==33)
{TBSeat33.setFocusable(false); TBSeat33.setEnabled(false);}
if(sn==34)
{TBSeat34.setFocusable(false); TBSeat34.setEnabled(false);}
 if(sn==35)
{TBSeat35.setFocusable(false); TBSeat35.setEnabled(false);}
if(sn==36)
{TBSeat36.setFocusable(false); TBSeat36.setEnabled(false);}
 if(sn==37)
{TBSeat37.setFocusable(false); TBSeat37.setEnabled(false);}
if(sn==38)
{TBSeat38.setFocusable(false); TBSeat38.setEnabled(false);}
 if(sn==39)
{TBSeat39.setFocusable(false); TBSeat39.setEnabled(false);}
 if(sn==40)
{TBSeat40.setFocusable(false); TBSeat40.setEnabled(false);}
 }
 }
 catch(Exception e){
 JOptionPane.showMessageDialog(null, e);
}
 }
```

```java
    // java.util.Date d = dateDC.getDate();<-- storing date in Date format

  }
```

**Code for Add Passenger Button:**

```java
private void AddPassengerButtonActionPerformed(java.awt.event.ActionEvent evt) {
if(date.equals("")||BusesTable.getSelectedRow()==-1)
 {JOptionPane.showMessageDialog(null, "Date or Bus hasn't been selected.");}
  else{
 String[] columnNames = new String[]{"Name","Seat No."};
     if(i==0){
data = new Object[1][2];
data[0][0] = JOptionPane.showInputDialog("Enter Passenger Name");
data[0][1] = null;
PassengerDetailsTable.setModel(new DefaultTableModel(data,columnNames));
i++;}
  else{
   Object[][] temp = new Object[data.length+1][2];
   for(int j=0;j<data.length;j++){
     for(int k=0;k<2;k++){
       temp[j][k]=data[j][k];
     }
   }
   temp[data.length][0]= JOptionPane.showInputDialog("Enter Passenger Name");
   temp[data.length][1] = null;
   data = temp;
   PassengerDetailsTable.setModel(new DefaultTableModel(data,columnNames));
   i++;
    price = Integer.parseInt((String) BusesTable.getValueAt(BusesTable.getSelectedRow(), 3));
   bookBT.setText("Pay Rs"+price*(i)+" and Book Now!");
price = price*i;
  }
```

```
  }


// TODO add your handling code here:

  }
```

**General code for the seat chooser buttons:**

```java
   if(PassengerDetailsTable.getRowCount()==0)
{JOptionPane.showMessageDialog(null, "Please add passenger before selecting seats");
TBSeat40.setSelected(false);} //inplace of TBseat40 put the desired seat button name
else{
  if(S40roundno%2!=0){
  String[] columnNames = new String[]{"Name","Seat No."};
    Object[][] temp = new Object[data.length][2];
  for(int j=0;j<data.length;j++){
    for(int k=0;k<2;k++){
      if(j==selno&&k==1){temp[j][k]=40;}
      else {temp[j][k]=data[j][k];}
    }}
   //temp[selno][1]=1;
    data = temp;
  PassengerDetailsTable.setModel(new DefaultTableModel(data,columnNames));
   ++selno; S40roundno++;}
  else
  { for(int j=0;j<data.length;j++){
    for(int k=0;k<2;k++){
      if(data[j][k]==(Object)40)//Not being executed! Because data[j][k] is never equal to "1" its
equal to 1
      {data[j][k]="";
    }
    }}//data[j][k].toString()
      String[] columnNames = new String[]{"Name","Seat No."};
      PassengerDetailsTable.setModel(new DefaultTableModel(data,columnNames));selno--
;S40roundno++;
```

```
        }}
// TODO add your handling code here:

    }
```

**Note: The above code is the general code for all the 40 seat choosing buttons. The generic code for the individual buttons can be formed by customising the seat numbers in the above code.**


**Code for Book now button:**


```java
private void bookBTActionPerformed(java.awt.event.ActionEvent evt) {

date = ((JTextField)dateDC.getDateEditor().getUiComponent()).getText();

selbus = (String) BusesTable.getValueAt(BusesTable.getSelectedRow(), 0);

System.out.print(PassengerDetailsTable.getRowCount());

//I want to put in the table corresponding to 'selbus' the elements in passengerdetailstable + the date

for(int r=0; r<PassengerDetailsTable.getRowCount();r++)

 {

   try {   // System.out.println("");

     String sql = "insert into "+selbus+" values('"+data[r][0]+"','"+data[r][1]+"','"+date+"')";

     pst = conn.createStatement();

         pst.execute(sql);

       } catch (SQLException e) {

         JOptionPane.showMessageDialog(null, e);

       }

  }/*done. now to save this into the users own table*/

for(int r=0; r<PassengerDetailsTable.getRowCount();r++)

 {

   try {   // System.out.println("");

     String sql = "insert into "+username+" values('"+data[r][0]+"','"+route+"','"+selbus+"','"+date+"')";

     pst = conn.createStatement();

         pst.execute(sql);

       } catch (SQLException e) {

         JOptionPane.showMessageDialog(null, e);
```

```
        }


   }


//JOptionPane.showMessageDialog(null, "Success");
 close();
success s = new success();
s.route=route;
s.setVisible(true);
}
```

**Code for Back to Menu button:**

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
Pg2_Welcome s = new Pg2_Welcome();
s.username=username;
s.setVisible(true);
   }
```
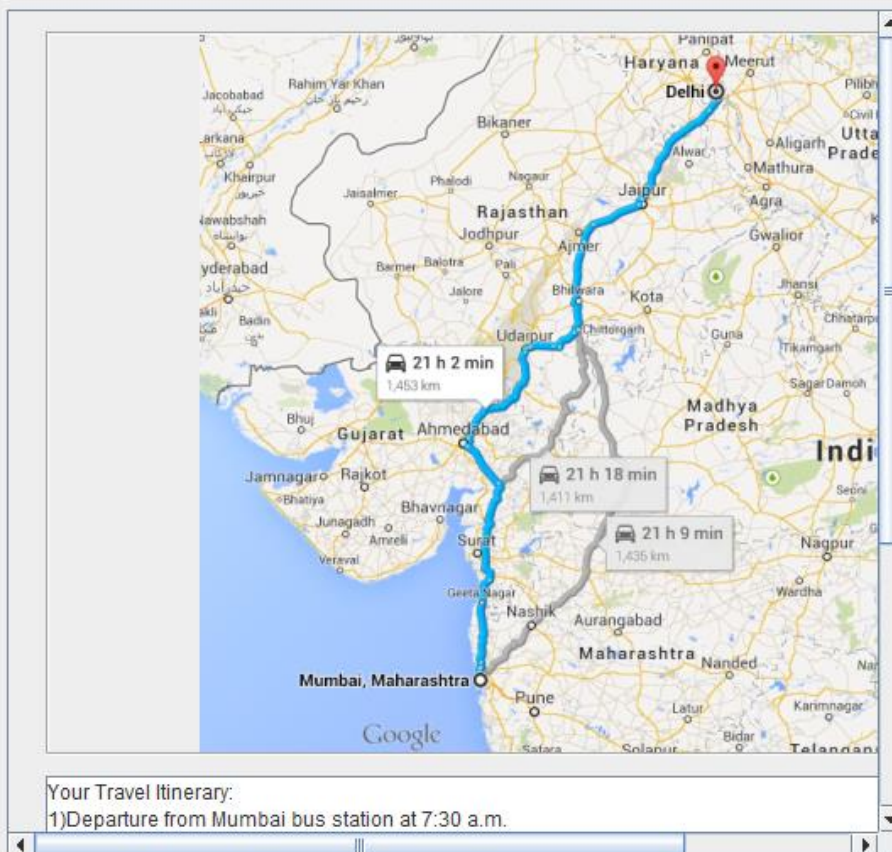
# Golden Links

Fast | Comfortable | Reliable

## Success!

Your tickets have been booked successfully

You may now relax and enjoy your forthcoming trip.

Click Here to View your Itinenary



Your Travel Itinerary:
1)Departure from Mumbai bus station at 7:30 a.m.

**Thank you for using our application! We hope you enjoyed it.**

**Do come again soon. Happy journey!**

EXIT

# Success

**Source Code:**

```java
import javax.swing.ImageIcon;

public class success extends javax.swing.JFrame {

    String route ;

    public success() {

        initComponents();

    }

public void close()

  {

    WindowEvent winClosingEvent = new WindowEvent(this,WindowEvent.WINDOW_CLOSING);

    Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);

  }
```

**Code for Click Here to view your itinerary button:**

```java
private void itinActionPerformed(java.awt.event.ActionEvent evt) {

    if(route.equals("Mumbai to Delhi"))

  {

    map.setIcon(new ImageIcon("E:\\Route Images\\m2d.png"));validate();

    itinTA.setText("Your Travel Itinerary:\n1)Departure from Mumbai bus station at 7:30
a.m.\n2)Stop for lunch and refreshments at 12:00p.m.\n3)Stop for dinner at 8:00p.m.\n4)Arrival at
Delhi Bus Station at 9:00a.m.");

}

    else if(route.equals("Delhi to Kolkata"))

  {

    map.setIcon(new ImageIcon("E:\\Route Images\\d2k.png"));

    itinTA.setText("Your Travel Itinerary:\n1)Departure from Delhi bus station at 7:30 a.m.\n2)Stop
for lunch and refreshments at 12:00p.m.\n3)Stop for dinner at 8:00p.m.\n4)Arrival at Kolkata Bus
Station at 9:00a.m.");

}

    else if(route.equals("Kolkata to Bangalore"))

  {
```

```
        map.setIcon(new ImageIcon("E:\\Route Images\\k2b.png"));

        itinTA.setText("Your Travel Itinerary:\n1)Departure from Kolkata bus station at 7:30
a.m.\n2)Stop for lunch and refreshments at 12:00p.m.\n3)Stop for dinner at 8:00p.m.\n4)Arrival at
Bangalore Bus Station at 9:00a.m.");

}

    else if(route.equals("Bangalore to Mumbai"))

    {

        map.setIcon(new ImageIcon("E:\\Route Images\\b2m.png"));

        itinTA.setText("Your Travel Itinerary:\n1)Departure from Bangalore bus station at 7:30
a.m.\n2)Stop for lunch and refreshments at 12:00p.m.\n3)Stop for dinner at 8:00p.m.\n4)Arrival at
Mumbai Bus Station at 9:00a.m.");

}

    }
```

**Code for Exit Button:**

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0); }
```

**Golden Links**

Fast | Comfortable | Reliable

# Cancel Tickets.. Fast and Simple.

**Simply click on the booking you want to cancel and hit Cancel!**

| Passenger Name | Route | Bus No. | Date |
|---|---|---|---|
| | | | |

[ Click Here to See All Bookings ]   [ Cancel! ]

[ Back to Menu ]

# CancelTickets

**Source Code:**

import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.sql.*;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.*;

import net.proteanit.sql.DbUtils;

import javax.swing.table.DefaultTableModel;

public class CancelTickets extends javax.swing.JFrame {

   Connection conn = null;

   Statement pst=null;

   String username;

   /** Creates new form CancelTickets */

   public CancelTickets() {

     initComponents();

   conn=javaconnect.ConnecrDb();}

private void close() {

   WindowEvent winClosingEvent = new WindowEvent(this,WindowEvent.WINDOW_CLOSING);

   Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);

  }


**Code for click here to sell all bookings button:**

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

//System.out.println("done");

    try{

   // System.out.println("error step");

  String sql = "select * from "+username+";";

```java
        //System.out.println("done");
    pst = conn.createStatement();
ResultSet rs=pst.executeQuery(sql);
 int j=0;
   int i=0;
   Object[][] data =null;
String[] columnNames = new String[]{"Name","Route","BusNo","Date"};
while(rs.next()==true)
{
String name=rs.getString("Name");
  String route=rs.getString("Route");
 String busno=rs.getString("BusNo");
   String date = rs.getString("Date");
//System.out.println("Completed initialization");
if(i==0){
data = new Object[1][4];
data[0][0] = name;
data[0][1] = route;
data[0][2] = busno;
data[0][3] = date;
AllBookingsTable.setModel(new DefaultTableModel(data,columnNames));
i++;}
 else{
  Object[][] temp = new Object[data.length+1][4];
   //System.out.println("enterred else"+i);
  for( j=0;j<data.length;j++){
    for(int k=0;k<4;k++){
      temp[j][k]=data[j][k];
        //System.out.println("inner for loop execution number "+k);
    }//System.out.println("outer for loop execution number "+j);
    } //System.out.println("Completed for loop in else"+i);
```

```
   temp[data.length][0]= name;

   temp[data.length][1] = route;

   temp[data.length][2]= busno;

   temp[data.length][3] = date;

   data = temp;

   AllBookingsTable.setModel(new DefaultTableModel(data,columnNames));

  i++;}
//System.out.println("Completed if else statment"+i);
 }

     }

     catch(Exception e){

  JOptionPane.showMessageDialog(null, e);}

}
```

**Code for cancel button:**

```
private void cancelBTNActionPerformed(java.awt.event.ActionEvent evt) {

     String name = (String) AllBookingsTable.getValueAt(AllBookingsTable.getSelectedRow(), 0);

     String route = (String) AllBookingsTable.getValueAt(AllBookingsTable.getSelectedRow(), 1);

     String busno = (String) AllBookingsTable.getValueAt(AllBookingsTable.getSelectedRow(), 2);

     String date = (String) AllBookingsTable.getValueAt(AllBookingsTable.getSelectedRow(), 3);

     /*Now to delete the row from users own table and table of bus*/

     String sql1 = "delete from "+username+" where name = '"+name+"' && route = '"+route+"' &&
BusNo = '"+busno+"' && Date = '"+date+"';";

     String sql2 = "delete from "+busno+" where Pname = '"+name+"' && Date = '"+date+"';";

 try{   pst = conn.createStatement();

     pst.execute(sql1);

     pst.execute(sql2); }

 catch (SQLException e)

      {

        JOptionPane.showMessageDialog(null, e);

      }
```

```java
    seeAllBTN.doClick();

    JOptionPane.showMessageDialog(null, "Your ticket has been successfully cancelled.\nThank you
for using our application.\nDo come again! :)");

  }

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

close();

Pg2_Welcome s = new Pg2_Welcome();

s.username=username;

s.setVisible(true);      // TODO add your handling code here:

}
```

**Code for Back to Menu button:**

```java
private void backBTNActionPerformed(java.awt.event.ActionEvent evt) {

close();

Pg2_Welcome s = new Pg2_Welcome();

s.username=username;

s.setVisible(true);      // TODO add your handling code here:

}
```

# Table Structures

**b2m1:**

| Field  | Type        | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| PName  | varchar(20) | YES  |     | NULL    |       |
| SeatNo | int(11)     | YES  |     | NULL    |       |
| Date   | char(11)    | YES  |     | NULL    |       |

**b2m2:**

| Field  | Type        | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| PName  | varchar(20) | YES  |     | NULL    |       |
| SeatNo | int(11)     | YES  |     | NULL    |       |
| Date   | char(11)    | YES  |     | NULL    |       |

**d2k1:**

| Field  | Type        | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| PName  | varchar(20) | YES  |     | NULL    |       |
| SeatNo | int(11)     | YES  |     | NULL    |       |
| Date   | char(11)    | YES  |     | NULL    |       |

**d2k2:**

| Field  | Type        | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| PName  | varchar(20) | YES  |     | NULL    |       |
| SeatNo | int(11)     | YES  |     | NULL    |       |
| Date   | char(11)    | YES  |     | NULL    |       |

**k2b1:**

| Field  | Type        | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| PName  | varchar(20) | YES  |     | NULL    |       |
| SeatNo | int(11)     | YES  |     | NULL    |       |
| Date   | char(11)    | YES  |     | NULL    |       |

**k2b2:**

| Field  | Type        | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| PName  | varchar(20) | YES  |     | NULL    |       |
| SeatNo | int(11)     | YES  |     | NULL    |       |
| Date   | char(11)    | YES  |     | NULL    |       |

**m2d1:**

| Field  | Type        | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| PName  | varchar(20) | YES  |     | NULL    |       |
| SeatNo | int(11)     | YES  |     | NULL    |       |
| Date   | char(11)    | YES  |     | NULL    |       |

**m2d2:**

| Field  | Type        | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| PName  | varchar(20) | YES  |     | NULL    |       |
| SeatNo | int(11)     | YES  |     | NULL    |       |
| Date   | char(11)    | YES  |     | NULL    |       |

**buses:**

| Field   | Type        | Null | Key | Default | Extra |
|---------|-------------|------|-----|---------|-------|
| BusNo   | char(5)     | YES  |     | NULL    |       |
| BusName | char(20)    | YES  |     | NULL    |       |
| AC      | char(5)     | YES  |     | NULL    |       |
| Route   | varchar(50) | YES  |     | NULL    |       |

**users:**

| Field           | Type     | Null | Key | Default | Extra |
|-----------------|----------|------|-----|---------|-------|
| Name            | char(50) | YES  |     | NULL    |       |
| Credit Card No. | int(11)  | YES  |     | NULL    |       |
| Username        | char(50) | YES  |     | NULL    |       |
| Password        | char(50) | YES  |     | NULL    |       |