

# **BUS TICKET RESERVATION SYSTEM**

## **SOURCE CODE:**

Admin.java

```
import java.awt.*;
import java.awt.event.*;
import java.awt.event.ActionListener;
import java.util.LinkedList;
import javax.swing.*;
import java.awt.GridBagConstraints;

class admin {

    JFrame frame = new JFrame("ADMIN");
    JPanel windowContent = new JPanel();
    int X, Y;
    JLabel name = new JLabel("Username");
    JLabel password = new JLabel("Password");
    JTextField username = new JTextField(20);
    JPasswordField passwd = new JPasswordField(20);
    JButton login = new JButton("Login");
    ButtonHandler listener = new ButtonHandler();
    JButton backbut = new JButton("Back");
    ButtonHandler1 butlistener = new ButtonHandler1();
```

```
admin() {  
    Toolkit toolKit = Toolkit.getDefaultToolkit();  
    java.awt.Dimension screenSize =  
        toolKit.getScreenSize();  
    X = screenSize.height;  
    Y = screenSize.width;  
    frame.setVisible(true);  
    frame.setBounds(0,0,2*X,Y);  
    frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
    frame.setIconImage(new ImageIcon("E:\\Projects for code-  
projects\\netbeans\\BusMod-master\\bus.jpg").getImage());  
  
    login.addActionListener(listener);  
    backbut.addActionListener(butlistener);  
  
    windowContent.setLayout(new FlowLayout());  
  
    windowContent.add(name);  
    windowContent.add(username);  
    windowContent.add(password);  
    windowContent.add(passwd);  
    windowContent.add(login);
```

```
windowContent.add(backbut);
```

```
frame.setContentPane(windowContent);
```

```
frame.setVisible(true);
```

```
frame.setBounds(0,0,2*X,Y);
```

```
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
}
```

```
private class ButtonHandler implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        if (username.getText().equalsIgnoreCase("admin")) {
```

```
            if (passwd.getText().equals("admin")) {
```

```
                adminMenu ob = new adminMenu("admin");
```

```
                frame.dispose();
```

```
            } else {
```

```
                JOptionPane.showMessageDialog(windowContent, "Incorrect  
Password", "PASSWORD ERROR", JOptionPane.PLAIN_MESSAGE);
```

```
            }
```

```
        } else if (username.getText().equalsIgnoreCase("admin")) {
```

```

        if (passwd.getText().equals("admin")) {
            adminMenu ob = new adminMenu("admin");
            frame.dispose();
        } else {
            JOptionPane.showMessageDialog(windowContent, "Incorrect
Password", "PASSWORD ERROR", JOptionPane.PLAIN_MESSAGE);
        }
    } else if (username.getText().equalsIgnoreCase("tony")) {
        if (passwd.getText().equals("tony")) {
            adminMenu ob = new adminMenu("tony");
            frame.dispose();
        } else {
            JOptionPane.showMessageDialog(windowContent, "Incorrect
Password", "PASSWORD ERROR", JOptionPane.PLAIN_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(windowContent, "Incorrect Username
or Password", "ERROR", JOptionPane.PLAIN_MESSAGE);
    }
}

private class ButtonHandler1 implements ActionListener
{
    // String newSelection;

```

```

@Override

public void actionPerformed(ActionEvent e)
{
    Main ob=new Main();
    frame.dispose();

}

}

public static void main(String[] args){
    admin ob=new admin("VRL");
}
}

```

### **Addbus.java**

```

import java.io.*;
import java.util.ArrayList;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.event.ActionListener;
import java.util.LinkedList;
import javax.swing.*;

```

```
import java.awt.GridBagConstraints;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.SwingConstants;


class addBus {

    JFrame frame;

    JPanel windowContent=new JPanel();

    char[][] seatStruct=new char[10][10];

    int X,Y;


    JButton type=new JButton("Enter Bus Type");


    TypeHandler typeListener=new TypeHandler();

    String inputType;

    JButton src=new JButton("ADD THE SOURCE");

    srcHandler srcListener=new srcHandler();

    String src;


    JButton des=new JButton("ADD THE DESTINATION");

    destHandler destListener=new destHandler();

    String dest;
```

```
JButton pric=new JButton("Enter the price");  
priceHandler priceListener=new priceHandler();  
String price;
```

```
JButton strt=new JButton("Enter the Start Time");  
strtHandler strtListener=new strtHandler();  
String start;
```

```
JButton end=new JButton("Enter the End Time");  
endHandler endListener=new endHandler();  
String etime;  
String rtNo;
```

```
JLabel route=new JLabel("Enter Route No.");  
JTextField inputRt=new JTextField(10);  
JLabel uniq=new JLabel("Enter Unique Key");  
JTextField inputUniq=new JTextField(10);  
JButton rows=new JButton("ENTER THE NO. OF ROW");  
rowHandler rowListener=new rowHandler();  
String row;  
JButton columns=new JButton("ENTER THE NO. OF COLUMNS ");
```



```
colHandler colListener=new colHandler();
```

```
String column;
```

```
JButton confirm=new JButton("CONFIRM");
```

```
addBus.ButtonHandler listener=new addBus.ButtonHandler();
```

```
JButton backbut = new JButton("Back");
```

```
BackHandler1 butlistener = new BackHandler1();
```

```
addBus(String company){
```

```
    frame=new JFrame("ADD A "+company+" BUS");
```

```
    Toolkit toolKit = Toolkit.getDefaultToolkit();
```

```
    java.awt.Dimension screenSize =
```

```
    toolKit.getScreenSize();
```

```
    X = screenSize.height;
```

```
    Y = screenSize.width;
```

```
    frame.setVisible(true);
```

```
    frame.setSize(X,Y);
```

```
    frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
    frame.setIconImage(new ImageIcon("C:\\Rohit\\Java  
Stuff\\NetBeans\\busmodfinal\\bus.jpg").getImage());
```

```
des.addActionListener(destListener);
sorc.addActionListener(srcListener);
pric.addActionListener(priceListener);
strt.addActionListener(strtListener);
end.addActionListener(endListener);
rows.addActionListener(rowListener);
columns.addActionListener(colListener);
backbut.addActionListener(butlistener);
```

```
confirm.addActionListener(listener);
```

```
for(int i = 0;i<10;i++)
{
    for(int j = 0;j<10;j++)
    {
        seatStruct[i][j] = 'X';//adding seats
    }
}
```

```
windowContent.setLayout(new GridLayout(0,1));
inputRt.setMaximumSize(new Dimension(250,20));
inputUniq.setMaximumSize(new Dimension(250,20));
```

```
windowContent.add(type);  
type.addActionListener(typeListener);
```

```
windowContent.add(sorc);  
windowContent.add(des);  
windowContent.add(pric);
```

```
windowContent.add(strt);  
// windowContent.add(inputStrt);  
windowContent.add(end);  
// windowContent.add(inputEnd);  
windowContent.add(route);  
windowContent.add(inputRt);  
windowContent.add(rows);  
// windowContent.add(inputrow);  
//windowContent.add(Box.createRigidArea(new Dimension(250,50)));  
windowContent.add(columns);  
  
//windowContent.add(inputcolumn);  
windowContent.add(confirm);  
windowContent.add(backbut);
```

```
frame.setContentPane(windowContent);

frame.setVisible(true);

frame.setBounds(0,0,2*X,Y);

frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

}
```

```
private class TypeHandler implements ActionListener

{

    @Override

    public void actionPerformed(ActionEvent e)

    {

        String[] options={"AC SLEEPER","AC SEMI SLEEPER","NON AC

SLEEPER","AC"};

        inputType=(String)JOptionPane.showInputDialog(windowContent,

"Choose One", "Input",

JOptionPane.INFORMATION_MESSAGE, null,

options, options[0]);

    }

}
```

```
private class rowHandler implements ActionListener
```

```

{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        String[] options={"1","2","3"};
        row=(String)JOptionPane.showInputDialog(windowContent,
"Choose One", "Input",
JOptionPane.INFORMATION_MESSAGE, null,
options, options[0]);
    }
}

private class colHandler implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        String[] options={"1","2","3","4","5","6","7","8","9"};
        column=(String)JOptionPane.showInputDialog(windowContent,
"Choose One", "Input",
JOptionPane.INFORMATION_MESSAGE, null,
options, options[0]);
    }
}

```

```
    }  
}
```

private class strHandler implements ActionListener

```
{  
    @Override  
    public void actionPerformed(ActionEvent e)  
    {  
        String[]  
options={"00:00","00:30","01:00","01:30","02:00","02:30","03:00","03:30","04:00","04:30","05:00","05:30","06:00","06:30","07:00","07:30","08:00","08:30","09:00","09:30","18:00","18:30","19:00","19:30","20:00","20:30","21:00","21:30","22:00","22:30","23:00","23:30"};  
  
        start=(String)JOptionPane.showInputDialog(windowContent,  
"Choose One", "Input",  
JOptionPane.INFORMATION_MESSAGE, null,  
options, options[0]);  
    }  
}
```

private class endHandler implements ActionListener

```
{
```

```

@Override

public void actionPerformed(ActionEvent e)

{

    String[]
options={"00:00","00:30","01:00","01:30","02:00","02:30","03:00","03:30","04:00","04:30","05:00","05:30","06:00","06:30","07:00","07:30","08:00","08:30","09:00","09:30","18:00","18:30","19:00","19:30","20:00","20:30","21:00","21:30","22:00","22:30","23:00","23:30"};

    etime=(String)JOptionPane.showInputDialog(windowContent,
"Choose One", "Input",
JOptionPane.INFORMATION_MESSAGE, null,
options, options[0]);

    }

}

private class srcHandler implements ActionListener

{

    @Override

    public void actionPerformed(ActionEvent e)

    {

        String[]
options={"Bangalore","Chennai","Kochi","Hyderabad","Manipal"};

        src=(String)JOptionPane.showInputDialog(windowContent,
"Choose One", "Input",
JOptionPane.INFORMATION_MESSAGE, null,

```

```
options, options[0]);
```

```
}
```

```
}
```

```
private class priceHandler implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        String[]
```

```
options={"250","300","350","400","450","500","550","600","650","700","750","800","850","900"};
```

```
        price=(String)JOptionPane.showInputDialog(windowContent,
```

```
"Choose One", "Input",
```

```
JOptionPane.INFORMATION_MESSAGE, null,
```

```
options, options[0]);
```

```
}
```

```
}
```

```
private class destHandler implements ActionListener
```

```
{
```

```
    @Override
```



```

    public void actionPerformed(ActionEvent e)
    {
        String[]
options={"Chennai","Bangalore","Kochi","Hyderabad","Manipal"};
        dest=(String)JOptionPane.showInputDialog(windowContent,
"Choose One", "Input",
JOptionPane.INFORMATION_MESSAGE, null,
options, options[0]);
    }
}

private class BackHandler1 implements ActionListener
{
    // String newSelection;

    @Override
    public void actionPerformed(ActionEvent e)
    {
        Main ob=new Main();
        frame.dispose();

    }

}

```

```

private class ButtonHandler implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        String type=inputType;
        int p=0;
        try{

            rtNo=inputRt.getText();

        }
        catch(NullPointerException w){

            JOptionPane.showMessageDialog(windowContent,"PLEASE ENTER
ROUTE NO. STARTING WITH THE LETTERS 'V' or 'S' or 'D'");

        }
        if(rtNo.charAt(0)=='V')
        {
            Vrl ob=new Vrl();
            ob.destinationchange(dest);
            ob.pricechange(Integer.parseInt(price));
            ob.sourcechange(src);

```

```
ob.typechange(type);
ob.starttimechange(start);
ob.endtimechange(etime);
ob.routechange(rtNo);
ob.key=rtNo;
ob.rows=Integer.parseInt(row);
ob.columns=Integer.parseInt(column);
//ob.seatmatrix=seatStruct;
ob.createseatmatrix(ob.rows,ob.columns);
p=1;

try {
    listener.addtryVrl(ob);
} catch (IOException ex) {
    Logger.getLogger(addBus.class.getName()).log(Level.SEVERE, null,
ex);
} catch (ClassNotFoundException ex) {
    Logger.getLogger(addBus.class.getName()).log(Level.SEVERE, null,
ex);
}
}

if(rtNo.charAt(0)=='D'){
```

```

Durgamba ob=new Durgamba();
ob.destinationchange(dest);
ob.pricechange(Integer.parseInt(price));
ob.sourcechange(src);
ob.typechange(type);
ob.starttimechange(start);
ob.endtimechange(etime);
ob.routechange(rtNo);
ob.key=rtNo;
ob.rows=Integer.parseInt(row);
ob.columns=Integer.parseInt(column);
ob.createseatmatrix(ob.rows,ob.columns);
p=1;
// ob.seatmatrix=seatStruct;
//      ob.createseatmatrix(rows,columns);
      try {
          listener.addtryDur(ob);
      } catch (IOException ex) {
          Logger.getLogger(addBus.class.getName()).log(Level.SEVERE, null,
ex);
      } catch (ClassNotFoundException ex) {
          Logger.getLogger(addBus.class.getName()).log(Level.SEVERE, null,
ex);

```

```
}  
}
```

```
if(rtNo.charAt(0)=='S'){  
    Sugama ob=new Sugama();  
    ob.destinationchange(dest);  
    ob.pricechange(Integer.parseInt(price));  
    ob.sourcechange(src);  
    ob.typechange(type);  
    //    ob.starttimechange(strt);  
    //    ob.endtimechange(end);  
    ob.routechange(rtNo);  
    ob.key=rtNo;  
  
    ob.rows=Integer.parseInt(row);  
    ob.columns=Integer.parseInt(column);  
    //    ob.createseatmatrix(rows,columns);  
    p=1;  
    try {  
        listener.addtrySug(ob);  
    } catch (IOException ex) {
```

```

        Logger.getLogger(addBus.class.getName()).log(Level.SEVERE, null,
ex);

    } catch (ClassNotFoundException ex) {

        Logger.getLogger(addBus.class.getName()).log(Level.SEVERE, null,
ex);

    }

    }

    if(p==1)
    {

        JOptionPane.showMessageDialog(windowContent,"THE BUS HAS
BEEN ADDED");

        Main ob=new Main();

        frame.dispose();

    }

    else

    {

        JOptionPane.showMessageDialog(windowContent,"ENTER ROUTE
STARTING WITH V/D/S");

    }

}

}

public void addtryVrl(Vrl ob) throws IOException, ClassNotFoundException
{

```

```
ArrayList<Vrl> vrltemp = new ArrayList();
```

```
Vrl obj1 = new Vrl();
```

```
Scanner scan = new Scanner(System.in);
```

```
int i;
```

```
int choice;
```

```
boolean eof =false;
```

```
Vrl obj = new Vrl();
```

```
FileInputStream f = new FileInputStream("VrlTesting.dat");
```

```
ObjectInputStream objectinput = new ObjectInputStream(f);
```

```
ArrayList<Vrl> read = (ArrayList<Vrl>) objectinput.readObject();
```

```
objectinput.close();
```

```
// obj.adddetails();
```

```
read.add(obj);
```

```
OutputStream file = new FileOutputStream("VrlTesting.dat");
```

```
OutputStream buffer = new BufferedOutputStream(file);
```

```
ObjectOutput output = new ObjectOutputStream(buffer);
```

```
output.writeObject(read);
```

```
output.close();
```

```
}
```

```
    public void addtryDur(Durgamba ob) throws IOException,  
ClassNotFoundException
```

```
{
```

```
    ArrayList<Durgamba> vrltemp = new ArrayList();
```

```
    Durgamba obj1 = new Durgamba();
```

```
        Durgamba obj = new Durgamba();
```

```
        FileInputStream f = new FileInputStream("DurgambaTesting.dat");
```

```
        ObjectInputStream objectinput = new ObjectInputStream(f);
```

```
        ArrayList<Durgamba> read = (ArrayList<Durgamba>)  
objectinput.readObject();
```

```
        objectinput.close();
```

```
        //    obj.adddetails();
```

```
        read.add(obj);
```

```
        OutputStream file = new FileOutputStream("DurgambaTesting.dat");
```

```
        OutputStream buffer = new BufferedOutputStream(file);
```

```
        ObjectOutput output = new ObjectOutputStream(buffer);
```



```
output.writeObject(read);
```

```
output.close();
```

```
}
```

```
public void addtrySug(Sugama ob) throws IOException,EOFException,  
ClassNotFoundException
```

```
{
```

```
    Sugama obj1 = new Sugama();
```

```
    Scanner scan = new Scanner(System.in);
```

```
    int i;
```

```
    int choice;
```

```
    int count = 0;
```

```
    Sugama obj = new Sugama();
```

```
    FileInputStream f = new FileInputStream("SugamaTesting.dat");
```

```
    ObjectInputStream objectinput = new ObjectInputStream(f);
```

```
    ArrayList<Sugama> read = (ArrayList<Sugama>)  
objectinput.readObject();
```

```
    objectinput.close();
```

```
//obj.adddetails();  
read.add(ob);  
  
OutputStream file = new FileOutputStream("SugamaTesting.dat");  
OutputStream buffer = new BufferedOutputStream(file);  
ObjectOutput output = new ObjectOutputStream(buffer);  
  
output.writeObject(read);  
  
output.close();  
  
}  
}
```

```
public static void main(String[] args){  
    addBus ob=new addBus("VRL");  
}  
  
}
```

## **adminmenu.java**

```
import java.awt.*;

import java.awt.event.*;

import java.awt.event.ActionListener;

import java.util.LinkedList;

import javax.swing.*;

import java.awt.GridBagConstraints;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;
```

```
class adminMenu{

    JFrame frame;

    JPanel windowContent=new JPanel();

    //JButton but1,but2,but3,but4;

    JButton but1=new JButton("ADD A BUS");
```

```
JButton but2=new JButton("MODIFY A BUS");  
JButton but3=new JButton("REMOVE A BUS");  
JButton but4=new JButton("BOOK A TICKET");  
JButton backbut = new JButton("Back");  
ButtonHandler5 butlistener = new ButtonHandler5();  
int flag=0;  
JLabel welcome;
```

```
String company;  
ButtonHandler1 listener1=new ButtonHandler1();  
ButtonHandler2 listener2=new ButtonHandler2();  
ButtonHandler3 listener3=new ButtonHandler3();  
ButtonHandler4 listener4=new ButtonHandler4();
```

```
int X,Y;
```

```
adminMenu(String company)  
{  
    Toolkit toolKit = Toolkit.getDefaultToolkit();  
    java.awt.Dimension screenSize =  
    toolKit.getScreenSize();  
    X = screenSize.height;  
    Y = screenSize.width;
```

```

frame=new JFrame(company+" ADMIN");

frame.setVisible(true);

frame.setSize(X,Y);

frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

frame.setIconImage(new ImageIcon("C:\\Rohit\\Java
Stuff\\NetBeans\\busmodfinal\\bus.jpg").getImage());

this.company=company;

welcome=new
JLabel("*****
*****WELCOME TO "+company.toUpperCase()+" ADMIN
MENU*****
*****");

but1.addActionListener(listener1);

but2.addActionListener(listener2);

but3.addActionListener(listener3);

but4.addActionListener(listener4);

backbut.addActionListener(butlistener);

welcome.setForeground(Color.red);

welcome.setFont(new Font("Helvetica",Font.BOLD,18));

windowContent.setLayout(new GridLayout(0,1));

//but1.setPreferredSize(new Dimension(80,20));

windowContent.add(welcome);

```

```
windowContent.add(but1);  
windowContent.add(but2);  
windowContent.add(but3);  
windowContent.add(but4);  
windowContent.add(backbut);  
frame.setContentPane(windowContent);  
frame.setBounds(0,0,2*X,Y);  
frame.setVisible(true);  
// frame.setSize(X,Y);  
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
  
}
```

```
private class ButtonHandler1 implements ActionListener  
{  
    @Override  
    public void actionPerformed(ActionEvent e)  
    {  
        addBus obj=new addBus(company);  
        frame.dispose();  
    }  
}
```

```
}  
}
```

```
private class ButtonHandler2 implements ActionListener
```

```
{  
    @Override  
    public void actionPerformed(ActionEvent e)  
    {  
        modify obj=new modify();  
        frame.dispose();  
    }  
}
```

```
private class ButtonHandler3 implements ActionListener
```

```
{  
    @Override  
    public void actionPerformed(ActionEvent e)  
    {  
        NullPointerException q=new NullPointerException();  
  
        String rem=(String)JOptionPane.showInputDialog(windowContent,"Enter  
the unique key of the bus you wish to remove","INPUT UNIQUE  
KEY",JOptionPane.PLAIN_MESSAGE, null,null, "");  
  
        //insert code to remove bus
```

```

try{
    if(rem.length()==0){
        flag=1;throw q;}
    }catch(NullPointerException ex){
        JOptionPane.showMessageDialog(windowContent,"ENTER A NON
        NULL VALUE FOR KEY");
    }

```

```

BusMod ob=new BusMod();

```

```

if(rem.charAt(0)=='V')
{
    try {
        if(flag==0){
            Vrl obj = new Vrl();
            FileInputStream f = new FileInputStream("VrlTesting.dat");
            ObjectInputStream objectinput = new ObjectInputStream(f);
            ArrayList<Vrl> read = (ArrayList<Vrl>) objectinput.readObject();
            objectinput.close();
            boolean exists=false;

            for(int i=0;i<read.size();i++){
                if(rem.equals(read.get(i).key))

```



```

        exists=true;
    }

    if(exists){
        JOptionPane.showMessageDialog(windowContent,"THE BUS
"+rem+" WAS REMOVED","SUCCESS",JOptionPane.PLAIN_MESSAGE);

        BusMod.vrlremove(rem);}

    else{

        JOptionPane.showMessageDialog(windowContent,"THE BUS
WITH KEY "+rem+" DOESNT EXIST IN THE LIST");
    }

    }} catch (FileNotFoundException ex) {

        Logger.getLogger(adminMenu.class.getName()).log(Level.SEVERE,
null, ex);

    } catch (IOException ex) {

        Logger.getLogger(adminMenu.class.getName()).log(Level.SEVERE,
null, ex);

    } catch (ClassNotFoundException ex) {

        Logger.getLogger(adminMenu.class.getName()).log(Level.SEVERE,
null, ex);

    }

}

else if(rem.charAt(0)=='D')
{

```

```
try {    if(flag==0){

        Durgamba obj = new Durgamba();

        FileInputStream f = new FileInputStream("DurgambaTesting.dat");

        ObjectInputStream objectinput = new ObjectInputStream(f);

        ArrayList<Durgamba> read = (ArrayList<Durgamba>)
objectinput.readObject();

        objectinput.close();

        boolean exists=false;

        for(int i=0;i<read.size();i++){

            if(read.get(i).key.equals(read.get(i).value))

                exists=true;

        }

        if(exists){

            BusMod.durremove(read.get(i).key);

            JOptionPane.showMessageDialog(windowContent,"THE BUS
"+read.get(i).key+" WAS REMOVED","SUCCESS",JOptionPane.PLAIN_MESSAGE);

        }

    }
```

```

        else

            JOptionPane.showMessageDialog(windowContent, "THE BUS
WITH KEY "+rem+" DOESNT EXIST IN THE LIST");

            }} catch (FileNotFoundException ex) {

                Logger.getLogger(adminMenu.class.getName()).log(Level.SEVERE,
null, ex);

            } catch (IOException ex) {

                Logger.getLogger(adminMenu.class.getName()).log(Level.SEVERE,
null, ex);

            } catch (ClassNotFoundException ex) {

                Logger.getLogger(adminMenu.class.getName()).log(Level.SEVERE,
null, ex);

            }

        }

    }

else if(rem.charAt(0)=='S')

{

    try {

        if(flag==0)

        {

            Sugama obj = new Sugama();

            FileInputStream f = new FileInputStream("SugamaTesting.dat");

            ObjectInputStream objectinput = new ObjectInputStream(f);

```

```

        ArrayList<Sugama> read = (ArrayList<Sugama>)
objectinput.readObject();

        objectinput.close();

        boolean exists=false;

        for(int i=0;i<read.size();i++){

            if(read.get(i).key.equals(rem))

                exists=true;

        }

        if(exists){

            JOptionPane.showMessageDialog(windowContent,"THE BUS
"+rem+" WAS REMOVED","SUCCESS",JOptionPane.PLAIN_MESSAGE);

            BusMod.sugaremove(rem);

        }

        else

            JOptionPane.showMessageDialog(windowContent,"THE BUS
WITH KEY "+rem+" DOESNT EXIST IN THE LIST");

    } }catch (FileNotFoundException ex) {

        Logger.getLogger(adminMenu.class.getName()).log(Level.SEVERE,
null, ex);

    } catch (IOException ex) {

```

```

        Logger.getLogger(adminMenu.class.getName()).log(Level.SEVERE,
null, ex);

    } catch (ClassNotFoundException ex) {

        Logger.getLogger(adminMenu.class.getName()).log(Level.SEVERE,
null, ex);

    }

}

}

}

```

```

private class ButtonHandler4 implements ActionListener

{

    @Override

    public void actionPerformed(ActionEvent e)

    {

        Main ob=new Main();

        frame.dispose();

    }

}

```

```
private class ButtonHandler5 implements ActionListener
{
    // String newSelection;

    @Override
    public void actionPerformed(ActionEvent e)
    {
        Main ob=new Main();
        frame.dispose();

    }

}

public static void main(String[] args)
{
    adminMenu ob=new adminMenu("VRL");
}
}
```

## **Main.java**

```
import java.awt.Container;

import java.awt.*;

import java.awt.event.*;

import java.io.*;

import java.util.*;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.*;

import javax.swing.border.Border;

import javax.swing.plaf.basic.BasicComboBoxUI;


public class Main extends JFrame
{

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

    String svar; // Variable to record the source data

    String dvar; //Variable to record the destination data


    JTextField inputDate=new JTextField("ENTER DATE IN DD-MM-
YYYY",20); //TO INPUT DATE
```

```

JButton confirm=new JButton("Book Ticket");

JButton cancel=new JButton("Cancel Ticket");

JButton admin=new JButton("Login as Admin");

JLabel lab=new
JLabel("*****
*****
*****WELCOME*****
*****
*****");

//JLabel lab=new JLabel("WELCOME");

Choice ch=new Choice();

JLabel l1=new JLabel("Source");

JLabel l2=new JLabel("Destination");

Choice ch2=new Choice();

JPanel windowContent;

ButtonHandler listener=new ButtonHandler();

ButtonHandler1 listener4=new ButtonHandler1();

ButtonHandler2 listener5=new ButtonHandler2();

//Combo listener1=new Combo();


String date;

JFrame frame=new JFrame("Welcome Page");

```



JComboBox patternList;//SOURCE

Combo listener1=new Combo(); // For the Jcombox1,i.e Source

JComboBox list2; //DESTINATION

Combo1 listener2=new Combo1(); // For the Jcombox1,i.e Destination

JPanel j1 = new JPanel();

JPanel j2 = new JPanel();

JPanel j3 = new JPanel();

JPanel j4 = new JPanel();

JComboBox dateList;//DATE

Combo2 listener3=new Combo2();//For the JComboBox2,i.e.date

Main()

{

int X,Y;

Point p;

Toolkit toolKit = Toolkit.getDefaultToolkit();

java.awt.Dimension screenSize = toolKit.getScreenSize();

toolKit.getScreenSize();

X = screenSize.height;//set size of window

Y = screenSize.width;

```
frame.getMaximizedBounds();

confirm.addActionListener(listener);
cancel.addActionListener(listener4);
admin.addActionListener(listener5);
lab.setBounds(0,0,X,Y);
windowContent=new JPanel();
windowContent.setLayout(new FlowLayout());
j1.setLayout(new FlowLayout());
j2.setLayout(new FlowLayout());
    lab.setForeground(Color.red);
    lab.setFont(new Font("Serif", Font.BOLD, 18));

windowContent.add(lab);
//windowContent.add(l1);
j2.add(l1);
lab.setBounds(X,100,20,20);
lab.setBounds(600,10,80,80);

windowContent.add(l2);
// l2.setBounds(0,10,20,20);
//FOR THE JCOMBO BOX(source)
String[] patternExamples = {
```

```

        "Bangalore",
        "Chennai",
        "Hyderabad",
        "Kochi",
        "Manipal"
    };

    svar=patternExamples[0];

    patternList = new JComboBox(patternExamples);
    patternList.setEditable(true);
    patternList.addActionListener(listener1);
    // For the second JCOMBO BOX,destination
    String[] patternExamples1 = {
        "Bangalore",
        "Chennai",
        "Hyderabad",
        "Kochi",
        "Manipal"
    };

    String[] dateExamples=new String[61];
    int k;
    int i=0;
    for(i=0;i<30;i++){

```

```
k=i+1;

dateExamples[i]=k+" NOV"+" 2013";

}

for(int j=0;j<31;j++)

{

    k=j+1;

    dateExamples[i]=k+" DEC"+" 2013";

    ++i;

}

//for date
```

```
date=dateExamples[0];

dateList=new JComboBox(dateExamples);

dateList.setEditable(false);

dateList.addActionListener(listener3);
```

```
dvar=patternExamples1[0];
```

```
list2 = new JComboBox(patternExamples1);

list2.setEditable(true);

list2.addActionListener(listener2);
```

```
//windowContent.add(patternList);

//windowContent.add(l2);

//windowContent.add(list2);

//windowContent.add(dateList);

j2.add(patternList);

j2.add(l2);

j2.add(list2);

j2.add(dateList);


// windowContent.add(confirm);

// windowContent.add(cancel);

// windowContent.add(admin);

j1.add(confirm);

j1.add(cancel);

j1.add(admin);


frame.setLayout(new BorderLayout());

frame.add(windowContent,BorderLayout.NORTH);

frame.add(j2,BorderLayout.CENTER);

frame.add(j1,BorderLayout.SOUTH);

frame.setSize(X,Y);

frame.setVisible(true);//to display the frame

frame.setBounds(0,0,2*X,Y-630);
```

```
frame.setIconImage(new ImageIcon("C:\\Rohit\\Java  
Stuff\\NetBeans\\busmodfinal\\bus.jpg").getImage());
```

```
}
```

```
private class ButtonHandler implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        try {
```

```
            if(svar.equals(dvar))
```

```
                JOptionPane.showMessageDialog(windowContent,"SOURCE AND  
DESTINATION CANNOT BE  
SAME","ERROR",JOptionPane.ERROR_MESSAGE);
```

```
            else {
```

```
                // patternList.addActionListener(listener1);
```

```
                Page2 ob=new Page2(svar,dvar,date);
```

```
                frame.dispose();
```

```
            } }catch (IOException ex) {
```

```
                Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null,  
ex);
```

```
            } catch (ClassNotFoundException ex) {
```

```
        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null,
ex);
```

```
    }
    // Page3 ob=new Page3();
    // frame.dispose();
}
}
```

```
private class Combo implements ActionListener
```

```
{
    //String newSelection;

    @Override
    public void actionPerformed(ActionEvent e)
    {
        JComboBox cb = (JComboBox)e.getSource();
        svar = (String)cb.getSelectedItem();

    }
}
```

```
}
```

```
private class Combo1 implements ActionListener
```

```
{
    // String newSelection;
```

```
@Override  
  
public void actionPerformed(ActionEvent e)  
{  
    JComboBox cb = (JComboBox)e.getSource();  
    dvar = (String)cb.getSelectedItem();  
  
}  
  
}
```

```
private class Combo2 implements ActionListener  
{  
    // String newSelection;  
  
    @Override  
  
    public void actionPerformed(ActionEvent e)  
    {  
        JComboBox cb = (JComboBox)e.getSource();  
        date = (String)cb.getSelectedItem();  
  
    }  
}
```



```
}
```

```
private class ButtonHandler1 implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        cancelTicket ob=new cancelTicket();
```

```
        frame.dispose();
```

```
    }
```

```
}
```

```
private class ButtonHandler2 implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        admin ob=new admin();
```

```
        frame.dispose();
```

```
    }
```

```
}
```

```
public static void main(String[] args){  
    //Main ob=new Main();  
  
    }  
}
```

### **Cancelticket.java**

```
import java.awt.*;  
import java.awt.event.*;  
import java.awt.event.ActionListener;  
import java.util.LinkedList;  
import javax.swing.*;  
import java.awt.GridBagConstraints;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.SwingConstants;  
  
public class cancelTicket
```

```
{

JFrame frame=new JFrame("CANCEL TICKET");

JPanel windowContent=new JPanel();

JButton ret=new JButton("BACK TO BOOKING PAGE");


// JLabel cancel=new JLabel("CANCEL TICKET");

// ButtonHandler cancelListener=new ButtonHandler();


JLabel rtNo=new JLabel("ENTER ROUTE NO.");

ButtonHandler1 rtListener=new ButtonHandler1();


JLabel rowNo=new JLabel("ENTER ROW NO. OF YOUR SEAT");

// ButtonHandler2 rwListener=new ButtonHandler2();


JLabel colNo=new JLabel("ENTER COLUMN NO. OF YOUR SEAT");

// ButtonHandler3 colListener=new ButtonHandler3();


JTextField inputArea=new JTextField(20); // FOR ROUTE

JTextField inputArea2=new JTextField(20); // FOR ROW NUMBER

JTextField inputArea3=new JTextField(20); // FOR COLUMN NUMBER

JButton confirm=new JButton("CONFIRM");

String date,rowno,colno;
```

```

JComboBox dateList;

Combo1 dateListener=new Combo1();//For the JComboBox2,i.e.date

ButtonHandler1 listener=new ButtonHandler1();

ButtonHandler2 listener1=new ButtonHandler2();

int X,Y;

String route;

int row,col;


cancelTicket()
{
    Toolkit toolKit = Toolkit.getDefaultToolkit();

    java.awt.Dimension screenSize =
    toolKit.getScreenSize();

    X = screenSize.height;

    Y = screenSize.width;

    frame.setVisible(true);

    frame.setSize(X,Y);

    frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    String[] dateExamples=new String[61];

    int k;

    int i=0;

    for(i=0;i<30;i++){

        k=i+1;

```

```

        dateExamples[i]=k+" NOV"+" 2013";
    }
    for(int j=0;j<31;j++)
    {
        k=j+1;

        dateExamples[i]=k+" DEC"+" 2013";

        ++i;
    }
    date=dateExamples[0];

    dateList=new JComboBox(dateExamples);

    dateList.setEditable(false);

    dateList.addActionListener(dateListener);


    //windowContent.setLayout(new GridLayout(0,1));


    //cancel.setMaximumSize(new Dimension(50,20));
    rtNo.setMaximumSize(new Dimension(50,20));
    rowNo.setMaximumSize(new Dimension(50,20));
    colNo.setMaximumSize(new Dimension(50,20));
    confirm.setMaximumSize(new Dimension(50,20));
    confirm.addActionListener(listener);
    ret.addActionListener(listener1);
    //windowContent.add(cancel)

```

```

;

windowContent.add(dateList);
windowContent.add(rtNo);
windowContent.add(inputArea);
windowContent.add(rowNo);
windowContent.add(inputArea2);
windowContent.add(colNo);
windowContent.add(inputArea3);
windowContent.add(confirm);
windowContent.add(ret);


frame.setContentPane(windowContent);
frame.setVisible(true);
frame.setBounds(0,0,2*X,Y);
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

frame.setIconImage(new ImageIcon("C:\\Rohit\\Java
Stuff\\NetBeans\\busmodfinal\\bus.jpg").getImage());

}


private class Combo1 implements ActionListener
{

    // String newSelection;

```

```

@Override

public void actionPerformed(ActionEvent e)
{
    JComboBox cb = (JComboBox)e.getSource();
    date = (String)cb.getSelectedItem();

}

}

/* private class ButtonHandler implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        //Insert code to cancel tickethere
    }
}*/

private class ButtonHandler1 implements ActionListener
{
    @Override

```

```

public void actionPerformed(ActionEvent e)
{

//rowno=(String)JOptionPane.showInputDialog(windowContent,"ENTER ROW
NO.", "INPUT", JOptionPane.PLAIN_MESSAGE);

    int ck;

    route=inputArea.getText();

    try
    {
        row=Integer.parseInt(inputArea2.getText());
        col = Integer.parseInt(inputArea3.getText());
        if(route.charAt(0)=='V' || route.charAt(0)=='D' || route.charAt(0)=='S' )
        {
            ck=BusMod.cancelTicket(route,row,col);

            switch (ck)
            {

                case 1:
JOptionPane.showMessageDialog(windowContent,"Invalid Route
Number", "ERROR",JOptionPane.ERROR_MESSAGE);

                    break;

                case 2: JOptionPane.showMessageDialog(windowContent,"Seat
is not occupied", "ERROR",JOptionPane.ERROR_MESSAGE);

                    break;

```



```
        case 3: JOptionPane.showMessageDialog(windowContent, "Ticket
Cancelled", "CONFIRMATION", JOptionPane.PLAIN_MESSAGE);

            break;

        }

    }

    else

        JOptionPane.showMessageDialog(windowContent, "Invalid Route
Number", "ERROR", JOptionPane.PLAIN_MESSAGE);

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

    }

    catch (FileNotFoundException ex)

    {

        Logger.getLogger(cancelTicket.class.getName()).log(Level.SEVERE,
null, ex);

    }

    catch (IOException ex)

    {

        Logger.getLogger(cancelTicket.class.getName()).log(Level.SEVERE,
null, ex);

    }

    catch (ClassNotFoundException ex)

    {
```

```

        Logger.getLogger(cancelTicket.class.getName()).log(Level.SEVERE,
null, ex);
    }

    catch (NumberFormatException e1)
    {
        JOptionPane.showMessageDialog(windowContent,"Invalid
Row/Column Number","ERROR",JOptionPane.ERROR_MESSAGE);
    }

}

}

```

```

private class ButtonHandler2 implements ActionListener

```

```

{
    @Override
    public void actionPerformed(ActionEvent e)
    {

```

```

//colno=(String)JOptionPane.showInputDialog(windowContent,"ENTER
COLUMN NO.", "INPUT", JOptionPane.PLAIN_MESSAGE);

```

```

// System.out.println("ENtered this part");

```

```

Main ob=new Main();

```

```
        frame.dispose();  
    }  
}
```

```
    public static void main(String[] args){  
        cancelTicket ob=new cancelTicket();  
    }  
}
```

### **Modify.java**

```
import java.awt.*;  
import java.awt.event.*;  
import java.awt.event.ActionListener;  
import java.util.LinkedList;  
import javax.swing.*;  
import java.awt.GridBagConstraints;  
import java.io.BufferedOutputStream;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.IOException;
```

```
import java.io.ObjectInputStream;
import java.io.ObjectOutput;
import java.io.ObjectOutputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;

class modify {

    int X,Y;
    int flag=0;
    JFrame frame=new JFrame("MODIFY BUS DETAILS");
    JPanel windowContent=new JPanel();

    JButton uniq=new JButton("ENTER UNIQUE KEY");
    ButtonHandler1 listener1=new ButtonHandler1();
    String key;

    JButton modType=new JButton("MODIFY BUS TYPE");
    ButtonHandler2 listener2=new ButtonHandler2();

    String finalType;//the type of the bus after modification
```

```
JButton modsTime=new JButton("MODIFY BUS START TIME");
```

```
ButtonHandler3 listener3=new ButtonHandler3();
```

```
String finalsTime;//the start time of the bus after modification
```

```
JButton modeTime=new JButton("MODIFY BUS END TIME");
```

```
ButtonHandler4 listener4=new ButtonHandler4();
```

```
String finaleTime;
```

```
JButton modPrice=new JButton("MODIFY TICKET PRICE");
```

```
ButtonHandler5 listener5=new ButtonHandler5();
```

```
String finalPrice;
```

```
int price;
```

```
JButton viewFinal=new JButton("DISPLAY MODIFIED DETAILS");
```

```
ButtonHandler6 listener6=new ButtonHandler6();
```

```
JButton backbut = new JButton("Back");
```

```
ButtonHandler7 butlistener = new ButtonHandler7();
```

```
modify(){
```

```
    Toolkit toolKit = Toolkit.getDefaultToolkit();
```

```
    java.awt.Dimension screenSize =
```

```
    toolKit.getScreenSize();
```

```
    X = screenSize.height;
```

```
    Y = screenSize.width;
```

```
    frame.setVisible(true);
```

```
    frame.setSize(X,Y);
```

```
    frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
    uniq.addActionListener(listener1);
```

```
    modType.addActionListener(listener2);
```

```
    modsTime.addActionListener(listener3);
```

```
    modeTime.addActionListener(listener4);
```

```
    modPrice.addActionListener(listener5);
```

```
    viewFinal.addActionListener(listener6);
```

```
    backbut.addActionListener((ActionListener) butlistener);
```

```
    windowContent.setLayout(new BoxLayout(windowContent,  
    BoxLayout.PAGE_AXIS));
```

```
    windowContent.add(uniq);
```

```
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(modType);
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(modsTime);
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(modeTime);
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(modPrice);
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(Box.createRigidArea(new Dimension(Y,40)));
windowContent.add(viewFinal);
windowContent.add(backbut);
```

```
frame.setIconImage(new ImageIcon("C:\\Rohit\\Java
Stuff\\NetBeans\\busmodfinal\\bus.jpg").getImage());
```

```
frame.setContentPane(windowContent);
```

```
frame.setVisible(true);
```

```
frame.setSize(X,Y);
```

```
frame.setBounds(0,0,2*X,Y);
```

```
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
}
```

```
private class ButtonHandler1 implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        try{
```

```
            key=(String)JOptionPane.showInputDialog(windowContent,"ENTER  
THE UNIQUE KEY","INPUT UNIQUE KEY", JOptionPane.PLAIN_MESSAGE,  
null, null, "");
```

```
        }
```

```
        catch(NullPointerException t)
```

```
        {
```

```
            JOptionPane.showMessageDialog(windowContent,"PLEASE INPUT  
THE KEY");
```

```
        }
```

```
    }
```

```
}
```



```
private class ButtonHandler2 implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        Object[] possibilities = {"AC", "SLEEPER", "VOLVO"};
```

```
        finalType=(String)JOptionPane.showInputDialog(windowContent,"MODIFY BUS  
TYPE","ENTER", JOptionPane.PLAIN_MESSAGE, null, possibilities, "");
```

```
    }
```

```
}
```

```
private class ButtonHandler3 implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        try{
```

```
            finalsTime=(String)JOptionPane.showInputDialog(windowContent,"ENTER THE  
NEW START TIME","ENTER", JOptionPane.PLAIN_MESSAGE, null, null, "");
```

```
        }
```

```

        catch(NullPointerException w)
        {
            JOptionPane.showMessageDialog(windowContent,"PLEASE INPUT
THE START TIME");
        }
    }
}

```

```

private class ButtonHandler4 implements ActionListener

```

```

{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        try{

```

```

        finaleTime=(String)JOptionPane.showInputDialog(windowContent,"ENTER THE
NEW END TIME","ENTER", JOptionPane.PLAIN_MESSAGE, null, null, "");

```

```

        }
        catch(NullPointerException x)
        {
            JOptionPane.showMessageDialog(windowContent,"PLEASE INPUT
THE END TIME");
        }
    }
}

```

```
}
```

```
private class ButtonHandler5 implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        try{
```

```
            finalPrice=(String)JOptionPane.showInputDialog(windowContent,"ENTER THE  
NEW PRICE","ENTER", JOptionPane.PLAIN_MESSAGE, null, null, "");
```

```
            price=Integer.parseInt(finalPrice);
```

```
        }
```

```
        catch(NumberFormatException e1)
```

```
        {
```

```
            JOptionPane.showMessageDialog(windowContent,"Invalid  
Entry,Please enter again");
```

```
        }
```

```
    }
```

```
}
```

```
private class ButtonHandler6 implements ActionListener
```

```
{
```

```

@Override

public void actionPerformed(ActionEvent e)
{
    NullPointerException q=new NullPointerException();

    boolean success=false;

    if(key.charAt(0)=='V'){
        try {
            if(finalsTime.length()==0)
                throw q;
            if(finaleTime.length()==0)
                throw q;
            success=modtryVrl(key);
        } catch (IOException ex) {
            Logger.getLogger(modify.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(modify.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (StringIndexOutOfBoundsException ex){
            JOptionPane.showMessageDialog(windowContent,"PLEASE
ENTER A NON-NULL VALUE FOR KEY AND PRESS CONFIRM");
        }
    }
}

```

```

        }catch(NullPointerException ex){

            JOptionPane.showMessageDialog(windowContent, "PLEASE
ENTER NON NULL VALUES FOR START TIME AND END TIME AND
PRESS CONFIRM");

        }}

else if(key.charAt(0)=='D'){

    try {

        if(finaleTime.length()==0)

            throw q;

        if(finalsTime.length()==0)

            throw q;

        success=modtryDur(key);

    } catch (IOException ex) {

        Logger.getLogger(modify.class.getName()).log(Level.SEVERE,
null, ex);

    } catch (ClassNotFoundException ex) {

        Logger.getLogger(modify.class.getName()).log(Level.SEVERE,
null, ex);

    }catch(StringIndexOutOfBoundsException ex){

        JOptionPane.showMessageDialog(windowContent,"PLEASE
ENTER A NON-NULL VALUE FOR KEY AND PRESS CONFIRM");

    }catch(NullPointerException ex){

```

```
JOptionPane.showMessageDialog(windowContent, "PLEASE  
ENTER NON NULL VALUES FOR START TIME AND END TIME AND  
PRESS CONFIRM");
```

```
}}
```

```
else if(key.charAt(0)=='S'){
```

```
try {
```

```
    if(finalsTime.length()==0)
```

```
        throw q;
```

```
    if(finaleTime.length()==0)
```

```
        throw q;
```

```
    success=modtrySug(key);
```

```
    } catch (IOException ex) {
```

```
        Logger.getLogger(modify.class.getName()).log(Level.SEVERE,  
null, ex);
```

```
    } catch (ClassNotFoundException ex) {
```

```
        Logger.getLogger(modify.class.getName()).log(Level.SEVERE,  
null, ex);
```

```
    }catch(StringIndexOutOfBoundsException ex){
```

```
        JOptionPane.showMessageDialog(windowContent,"PLEASE  
ENTER A NON-NULL VALUE FOR KEY AND PRESS CONFIRM");
```

```
    }catch(NullPointerException ex){
```

```
JOptionPane.showMessageDialog(windowContent, "PLEASE  
ENTER NON NULL VALUES FOR START TIME AND END TIME AND  
PRESS CONFIRM");
```

```
}}
```

```
else
```

```
{
```

```
JOptionPane.showMessageDialog(windowContent,"PLEASE ENTER  
A STRING STARTING FROM 'V' or 'D' or 'S' ");
```

```
}
```

```
String disp="BUS KEY: "+key+"BUS TYPE: "+finalType+"\n"+ "BUS  
START TIME: "+finalsTime+"\n"+"BUS END TIME: "+finaleTime+"\n"+
```

```
"TICKET PRICE: "+finalPrice;
```

```
if(success){
```

```
JOptionPane.showMessageDialog(windowContent,disp,"MODIFIED BUS  
DETAILS",JOptionPane.PLAIN_MESSAGE);
```

```
}
```

```
}
```

```
}
```

```

private class ButtonHandler7 implements ActionListener
{
    // String newSelection;

    @Override
    public void actionPerformed(ActionEvent e)
    {
        Main ob=new Main();

        frame.dispose();

    }
}

public boolean modtrySug(String uniq) throws IOException,
ClassNotFoundException
{

    // System.out.println("Entered the Durgamba function");

    FileInputStream f = new FileInputStream("SugamaTesting.dat");
    ObjectInputStream objectinput = new ObjectInputStream(f);

    ArrayList<Sugama> read2 = (ArrayList<Sugama>)
objectinput.readObject();

    objectinput.close();

    //int flag=0;

```



```

boolean exists=false;
for(int i=0;i<read2.size();i++)
{
    if(uniq.equalsIgnoreCase(read2.get(i).key))
    {exists=true;
        flag=1;
        read2.get(i).typechange(finalType);
        read2.get(i).starttimechange(finalsTime);
        read2.get(i).endtimechange(finaleTime);
        read2.get(i).pricechange(price);
        // System.out.println("Found the bus");

        break;}
    }

    System.out.println(read2.get(0).starttime);
//    obj.adddetails();
// read.add(ob);

    ArrayList<String> keyList=new ArrayList<>();

    for(int i=0;i<read2.size();i++)
    {

```

```
        keyList.add(read2.get(i).key);  
    }
```

```
String disp="";
```

```
for(int i=0;i<read2.size();i++)  
{  
    disp=disp+keyList.get(i);  
    disp=disp+" ";  
}
```

```
if(!exists)  
{
```

```
    JOptionPane.showMessageDialog(windowContent,"PLEASE ENTER  
A KEY OF A BUS WHICH EXISTS AND THEN PRESS CONFIRM");
```

```
    JOptionPane.showMessageDialog(windowContent,disp,"KEY  
LIST",JOptionPane.INFORMATION_MESSAGE);
```

```
}
```

```
OutputStream file2 = new FileOutputStream("SugamaTesting.dat");
```

```
OutputStream buffer2 = new BufferedOutputStream(file2);
```

```
ObjectOutput output2 = new ObjectOutputStream(buffer2);
```

```
output2.writeObject(read2);
```

```
output2.close();
```

```
if(!exists)
```

```
    return false;
```

```
else
```

```
    return true;
```

```
}
```

```
public boolean modtryDur(String uniq) throws IOException,  
ClassNotFoundException,NullPointerException
```

```
{
```

```
    // System.out.println("Entered the Durgamba function");
```

```
    FileInputStream f = new FileInputStream("DurgambaTesting.dat");
```

```
    ObjectInputStream objectinput = new ObjectInputStream(f);
```

```
        ArrayList<Durgamba> read1 = (ArrayList<Durgamba>)
objectinput.readObject();

        objectinput.close();

        boolean exists=false;

        //int flag=0;

        for(int i=0;i<read1.size();i++)
        {
            if(uniq.equalsIgnoreCase(read1.get(i).key))
            {exists=true;

                flag=1;

                read1.get(i).typechange(finalType);
                read1.get(i).starttimechange(finaleTime);
                read1.get(i).endtimechange(finaleTime);
                read1.get(i).pricechange(price);

                // System.out.println("Found the bus");

                break;}
        }

        System.out.println(read1.get(0).starttime);

        //    obj.adddetails();

        // read.add(ob);
```

```
ArrayList<String> keyList=new ArrayList<>();
```

```
for(int i=0;i<read1.size();i++)  
{  
    keyList.add(read1.get(i).key);  
}
```

```
String disp="";
```

```
for(int i=0;i<read1.size();i++)  
{  
    disp=disp+keyList.get(i);  
    disp=disp+" ";  
}
```

```
if(!exists)  
{
```

```
    JOptionPane.showMessageDialog(windowContent,"PLEASE ENTER  
A KEY OF A BUS WHICH EXISTS AND THEN PRESS CONFIRM");
```

```
    JOptionPane.showMessageDialog(windowContent,disp,"KEY  
LIST",JOptionPane.INFORMATION_MESSAGE);  
}
```

```
OutputStream file1 = new FileOutputStream("DurgambaTesting.dat");
```

```
OutputStream buffer1 = new BufferedOutputStream(file1);
```

```
ObjectOutput output1 = new ObjectOutputStream(buffer1);
```

```
output1.writeObject(read1);
```

```
output1.close();
```

```
if(!exists)
```

```
    return false;
```

```
else
```

```
    return true;
```

```
}
```

```
public boolean modtryVrl(String uniq) throws IOException,  
ClassNotFoundException
```

```
{
```

```
FileInputStream f = new FileInputStream("VrlTesting.dat");
```

```
ObjectInputStream objectinput = new ObjectInputStream(f);
```

```

ArrayList<Vrl> read = (ArrayList<Vrl>) objectinput.readObject();
objectinput.close();

//int flag=0;

boolean exists=false;
for(int i=0;i<read.size();i++)
{

    if(uniq.equalsIgnoreCase(read.get(i).key))
    {
        flag=1;exists=true;

        read.get(i).typechange(finalType);
        read.get(i).starttimechange(finaleTime);
        read.get(i).endtimechange(finaleTime);
        read.get(i).pricechange(price);

        // System.out.println("Found the bus");

        break;}
}

System.out.println(read.get(0).starttime);

```

```

ArrayList<String> keyList=new ArrayList<>();

```

```
for(int i=0;i<read.size();i++)  
{  
    keyList.add(read.get(i).key);  
}
```

```
String disp="";
```

```
for(int i=0;i<read.size();i++)  
{  
    disp=disp+keyList.get(i);  
    disp=disp+" ";  
}
```

```
if(!exists)  
{  
    JOptionPane.showMessageDialog(windowContent,"PLEASE ENTER  
A KEY OF A BUS WHICH EXISTS AND THEN PRESS CONFIRM");  
    JOptionPane.showMessageDialog(windowContent, disp,"KEY  
LIST",JOptionPane.INFORMATION_MESSAGE);  
}
```

```
OutputStream file = new FileOutputStream("VrlTesting.dat");
```



```
OutputStream buffer = new BufferedOutputStream(file);  
ObjectOutput output = new ObjectOutputStream(buffer);
```

```
output.writeObject(read);
```

```
output.close();
```

```
if(!exists)  
    return false;  
else  
    return true;
```

```
}
```

```
public static void main(String[] args){  
    modify ob=new modify();  
}  
}
```

### **Page2.java**

```
import java.awt.*;  
import java.util.*;  
import java.awt.event.*;  
import java.awt.event.ActionListener;
```

```
import java.util.LinkedList;

import javax.swing.*;

import java.awt.GridBagConstraints;

import java.io.*;

import java.lang.String;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Page2

{

    JFrame frame2;//=new JFrame("BUS DETAILS");

    JPanel windowContent=new JPanel();


    JButton rowNo=new JButton("ENTER ROW NO.");

    JButton colNo=new JButton("ENTER COLUMN NO.");


    JButton confirm=new JButton("Confirm");

    JButton show=new JButton("Seat Matrix");


    ButtonHandler listener=new ButtonHandler();//actionListener for confirm

    ButtonHandler2 listener2=new ButtonHandler2();//actionListener for
showSeatMatrix


    ButtonHandler3 rowListener=new ButtonHandler3();
```

```
ButtonHandler4 colListener=new ButtonHandler4();
```

```
JLabel temp=new JLabel("ENTER ROUTE.No");
```

```
JLabel temp2=new JLabel("ENTER SEAT No.");
```

```
JLabel temp3=new JLabel("      ");
```

```
JTextField inputArea=new JTextField(); // FOR ROUTE NUMBER
```

```
// JTextField inputArea2=new JTextField(); // FOR SEAT NUMBER
```

```
String rowno,colno;
```

```
int X,Y;
```

```
String svar1,dvar1;
```

```
JButton back=new JButton("BACK");
```

```
ButtonHandler5 listener3=new ButtonHandler5();
```

```
//String svar1;
```

```
//String dvar1;
```

```
String date1;
```

```
Page2(String svar,String dvar,String date) throws IOException,  
ClassNotFoundException
```

```
{
```

```
    svar1=svar;
```

```
    dvar1=dvar;
```

```
date1=date;

Toolkit toolKit = Toolkit.getDefaultToolkit();

// System.out.println("SVAR IS- " +svar);

// System.out.println("DVAR IS- " +dvar);

java.awt.Dimension screenSize =

toolKit.getScreenSize();

X = screenSize.height;

Y = screenSize.width;

String check="Buses On "+svar+"-"+dvar;

frame2=new JFrame(check);

// frame2.setVisible(true);

//frame2.setBounds(0,0,2*X,Y);

//frame2.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

confirm.addActionListener(listener);

show.addActionListener(listener2);

back.addActionListener(listener3);

windowContent.setLayout(new GridLayout(0,8));

FileInputStream f = new FileInputStream("VrlTesting.dat");

ArrayList<Vrl> read;

ObjectInputStream objectinput = new ObjectInputStream(f);

read = (ArrayList<Vrl>) objectinput.readObject();

objectinput.close();

FileInputStream f1 = new FileInputStream("DurgambaTesting.dat");
```

```
ArrayList<Durgamba> read1;  
ObjectInputStream objectinput1 = new ObjectInputStream(f1);  
read1 = (ArrayList<Durgamba>) objectinput1.readObject();  
objectinput1.close();  
FileInputStream f2 = new FileInputStream("SugamaTesting.dat");  
ArrayList<Sugama> read2;  
ObjectInputStream objectinput2 = new ObjectInputStream(f2);  
read2 = (ArrayList<Sugama>) objectinput2.readObject();  
objectinput2.close();  
//System.out.println(read.size());  
//System.out.println(read1.size());  
//System.out.println(read2.size());
```

```
rowNo.addActionListener(rowListener);  
colNo.addActionListener(colListener);
```

```
JLabel tempCompany=new JLabel("COMPANY");  
tempCompany.setFont(new Font("Serif", Font.BOLD, 18));  
tempCompany.setForeground(Color.red);  
tempCompany.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempCompany);  
JLabel tempAmmenities=new JLabel("AMMENITIES");  
tempAmmenities.setFont(new Font("Serif", Font.BOLD, 18));
```

```
tempAmmenities.setForeground(Color.red);

tempAmmenities.setBorder(BorderFactory.createLineBorder(Color.BLACK));

windowContent.add(tempAmmenities);


JLabel tempRoute=new JLabel("ROUTE NUMBER");
tempRoute.setFont(new Font("Serif", Font.BOLD, 18));
tempRoute.setForeground(Color.red);
tempRoute.setBorder(BorderFactory.createLineBorder(Color.BLACK));
windowContent.add(tempRoute);

JLabel tempType=new JLabel("TYPE");
tempType.setFont(new Font("Serif", Font.BOLD, 18));
tempType.setForeground(Color.red);
tempType.setBorder(BorderFactory.createLineBorder(Color.BLACK));
windowContent.add(tempType);

JLabel tempPrice=new JLabel("PRICE"); // Change this
tempPrice.setFont(new Font("Serif", Font.BOLD, 18));
tempPrice.setForeground(Color.red);
tempPrice.setBorder(BorderFactory.createLineBorder(Color.BLACK));
windowContent.add(tempPrice);


JLabel tempDate=new JLabel("DATE");
tempDate.setFont(new Font("Serif", Font.BOLD, 18));
```

```
tempDate.setForeground(Color.red);  
tempDate.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempDate);
```

```
JLabel tempStrt=new JLabel("START TIME");  
tempStrt.setFont(new Font("Serif", Font.BOLD, 18));  
tempStrt.setForeground(Color.red);  
tempStrt.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempStrt);
```

```
JLabel tempEnd=new JLabel("END TIME");  
tempEnd.setFont(new Font("Serif", Font.BOLD, 18));  
tempEnd.setForeground(Color.red);  
tempEnd.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempEnd);
```

```
frame2.setIconImage(new ImageIcon("C:\\Rohit\\Java  
Stuff\\NetBeans\\busmodfinal\\bus.jpg").getImage());
```

```
for(int i=0;i<read.size();i++)
```

```
{
```

```
if(svar.equals (read.get(i).source) && dvar.equals (read.get(i).destination))
{
    tempCompany=new JLabel("VRL");
    tempCompany.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    windowContent.add(tempCompany);

    String ammenities;
    ammenities="CHARGING POINT";

    if(read.get(i).type.equals("AC SLEEPER"))
        ammenities="chargePoint&readLight";
    else if(read.get(i).type.equals("AC SEMI SLEEPER"))
        ammenities="chargePoint&readLight&TV";
    else if(read.get(i).type.equals("NON AC SLEEPER"))
        ammenities="chargePoint&TV";

    tempAmmenities=new JLabel(ammenities);

    tempAmmenities.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    windowContent.add(tempAmmenities);
```



```
tempRoute=new JLabel(read.get(i).key);  
tempRoute.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempRoute);  
tempType=new JLabel(read.get(i).type);  
tempType.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempType);  
Integer k=read.get(i).price;
```

```
tempPrice=new JLabel(k.toString());  
tempPrice.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempPrice);
```

```
tempDate=new JLabel(date);  
tempDate.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempDate);  
windowContent.add(rowNo);  
windowContent.add(colNo);
```

```
tempStrt=new JLabel(read.get(i).starttime);  
tempStrt.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempStrt);
```

```
tempEnd=new JLabel(read.get(i).endtime);
```

```

tempEnd.setBorder(BorderFactory.createLineBorder(Color.BLACK));
windowContent.add(tempEnd);

}
}
for(int i=0;i<read1.size();i++)
{

if(svar.equals (read1.get(i).source) &&dvar.equals (read1.get(i).destination))
{
tempCompany=new JLabel("Durgamba");
tempCompany.setBorder(BorderFactory.createLineBorder(Color.BLACK));
windowContent.add(tempCompany);

String ammenities;
ammenities="CHARGING POINT";

if(read1.get(i).type.equals("AC SLEEPER"))
    ammenities="chargePoint&readLight";
else if(read1.get(i).type.equals("AC SEMI SLEEPER"))
    ammenities="chargePoint&readLight&TV";
else if(read1.get(i).type.equals("NON AC SLEEPER"))

```

```
ammenities="chargePoint&TV";
```

```
tempAmmenities=new JLabel(ammenities);
```

```
tempAmmenities.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempAmmenities);
```

```
tempRoute=new JLabel(read1.get(i).key);
```

```
tempRoute.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempRoute);
```

```
tempType=new JLabel(read1.get(i).type);
```

```
tempType.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempType);
```

```
Integer k=read1.get(i).price;
```

```
tempPrice=new JLabel(k.toString());
```

```
tempPrice.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempPrice);
```

```
tempDate=new JLabel(date);  
tempDate.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempDate);
```

```
tempStrt=new JLabel(read1.get(i).starttime);  
tempStrt.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempStrt);
```

```
tempEnd=new JLabel(read1.get(i).endtime);  
tempEnd.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
windowContent.add(tempEnd);
```

```
}
```

```
}
```

```
for(int i=0;i<read2.size();i++)
```

```
{
```

```
if(svar.equals (read2.get(i).source) && dvar.equals (read2.get(i).destination))
```

```
{
```

```
tempCompany=new JLabel("Sugama");
```

```
tempCompany.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempCompany);

String ammenities;
ammenities="CHARGING POINT";

if(read2.get(i).type.equals("AC SLEEPER"))
    ammenities="chargePoint&readLight";
else if(read2.get(i).type.equals("AC SEMI SLEEPER"))
    ammenities="chargePoint&readLight";
else if(read2.get(i).type.equals("NON AC SLEEPER"))
    ammenities="chargePoint";

tempAmmenities=new JLabel(ammenities);

tempAmmenities.setBorder(BorderFactory.createLineBorder(Color.BLACK));
windowContent.add(tempAmmenities);

tempRoute=new JLabel(read2.get(i).key);
tempRoute.setBorder(BorderFactory.createLineBorder(Color.BLACK));
windowContent.add(tempRoute);
tempType=new JLabel(read2.get(i).type);
tempType.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempType);
```

```
Integer k=read2.get(i).price;
```

```
tempPrice=new JLabel(k.toString());
```

```
tempPrice.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempPrice);
```

```
tempDate=new JLabel(date);
```

```
tempDate.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempDate);
```

```
tempStrt=new JLabel(read2.get(i).starttime);
```

```
tempStrt.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempStrt);
```

```
tempEnd=new JLabel(read2.get(i).endtime);
```

```
tempEnd.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```
windowContent.add(tempEnd);
```

```
}  
}
```

```
//date,
```

```
    windowContent.add(back);  
    windowContent.add(temp);  
    windowContent.add(inputArea);  
    windowContent.add(temp3);  
    //windowContent.add(temp3);  
    windowContent.add(show);  
    // windowContent.add(temp3);  
//    windowContent.add(temp2);  
    // windowContent.add(temp3);  
//    windowContent.add(inputArea2);  
    //windowContent.add(temp3);  
    windowContent.add(rowNo);  
    windowContent.add(colNo);  
    windowContent.add(confirm);  
  
    frame2.setContentPane(windowContent);  
    frame2.setVisible(true);  
    frame2.setBounds(0,0,X,Y);
```

```
frame2.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
}
```

```
private class ButtonHandler2 implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        //int i=0;
```

```
        //JOptionPane.showMessageDialog(windowContent,"Hey","SEAT  
MATRIX",JOptionPane.PLAIN_MESSAGE);
```

```
        FileInputStream f = null;
```

```
        try {
```

```
            f = new FileInputStream("VrlTesting.dat");
```

```
        } catch (FileNotFoundException ex) {
```

```
            Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,  
ex);
```

```
        }
```



```

ArrayList<Vrl> read = null;
ObjectInputStream objectinput = null;
    try {
        objectinput = new ObjectInputStream(f);
    } catch (IOException ex) {
        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
    }
    try {
        read = (ArrayList<Vrl>) objectinput.readObject();
    } catch (IOException ex) {
        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
    }
    try {
        objectinput.close();
    } catch (IOException ex) {
        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
    }
    FileInputStream f1 = null;

```

```
try {
    f1 = new FileInputStream("DurgambaTesting.dat");
} catch (FileNotFoundException ex) {
    Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
}

ArrayList<Durgamba> read1 = null;
ObjectInputStream objectinput1 = null;

try {
    objectinput1 = new ObjectInputStream(f1);
} catch (IOException ex) {
    Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
}

try {
    read1 = (ArrayList<Durgamba>) objectinput1.readObject();
} catch (IOException ex) {
    Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
} catch (ClassNotFoundException ex) {
    Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
}

try {
```

```
        objectinput1.close();
    } catch (IOException ex) {
        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
    }
    FileInputStream f2 = null;
    try {
        f2 = new FileInputStream("SugamaTesting.dat");
    } catch (FileNotFoundException ex) {
        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
    }
    ArrayList<Sugama> read2 = null;
    ObjectInputStream objectinput2 = null;
    try {
        objectinput2 = new ObjectInputStream(f2);
    } catch (IOException ex) {
        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
    }
    try {
        read2 = (ArrayList<Sugama>) objectinput2.readObject();
    } catch (IOException ex) {
```

```

        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);

    } catch (ClassNotFoundException ex) {

        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);

    }

    try {

        objectinput2.close();

    } catch (IOException ex) {

        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);

    }

```

```

String key=inputArea.getText();

```

```

char[][] seat;

```

```

String work="";

```

```

for(int i=0;i<read.size();i++)

```

```

{

    if(read.get(i).key.equalsIgnoreCase(key))

    {

```

```

        seat=read.get(i).seatmatrix;

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

        for(int k=0;k<read.get(i).rows;k++)
        {
            for(int j=0;j<read.get(i).columns;j++)
            {
                work=work+seat[k][j]+" ";

            }

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

            work=work+"\n";

        }

    }
}

```

```

for(int i=0;i<read1.size();i++)
{
    if(read1.get(i).key.equalsIgnoreCase(key))
    {
        seat=read1.get(i).seatmatrix;

        for(int k=0;k<read1.get(i).rows;k++)
        {

```

```
        for(int j=0;j<read1.get(i).columns;j++)  
            work=work+seat[k][j]+" ";  
        work=work+"\n";  
    }  
}  
}
```

```
for(int i=0;i<read2.size();i++)  
{  
    if(read2.get(i).key.equalsIgnoreCase(key))  
    {  
        seat=read2.get(i).seatmatrix;  
  
        for(int k=0;k<read2.get(i).rows;k++)  
        {  
            for(int j=0;j<read2.get(i).columns;j++)  
                work=work+seat[k][j]+" ";  
            work=work+"\n";  
        }  
    }  
}
```

```
JOptionPane.showMessageDialog(windowContent,work+"\n"+"X=>FREE,O=>O  
CCUPIED","SEAT MATRIX", JOptionPane.PLAIN_MESSAGE);
```

```
        //seatMatrix ob=new seatMatrix(seat);  
  
        // g=ob.getGraphics();  
  
        // ob.paint(g);  
  
    }  
}
```

```
private class ButtonHandler3 implements ActionListener  
{  
    @Override  
    public void actionPerformed(ActionEvent e)  
    {  
        try  
        {  
            while(true)
```

```

    {
        rowno=(String)JOptionPane.showInputDialog(windowContent,"ENTER
THE ROW NO.(0-3) ","INPUT ROW", JOptionPane.QUESTION_MESSAGE);

        int i=Integer.parseInt(rowno);

        if(i>3||i<0)
        {
            JOptionPane.showMessageDialog(windowContent,"PLEASE INPUT
NO.S IN THE RANGE 0-3");
            continue;
        }
        else
            break;
    }

}catch(NumberFormatException e1)
{
    JOptionPane.showMessageDialog(windowContent,"Invalid Entry");
}
}
}

```



```

private class ButtonHandler4 implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            while(true)
            {
                colno=(String)JOptionPane.showInputDialog(windowContent,"ENTER
THE COLUMN NO.(0-9) ","INPUT ROW",
JOptionPane.QUESTION_MESSAGE);

                int i=Integer.parseInt(colno);

                if(i>9||i<0)
                {
                    JOptionPane.showMessageDialog(windowContent,"PLEASE INPUT
NO.S IN THE RANGE 0-9");
                    continue;
                }
                else
                break;
            }
        }
    }
}

```

```

    }catch(NumberFormatException e1)
    {
        JOptionPane.showMessageDialog(windowContent,"Invalid Entry");
    }
}
}

```

```

private class ButtonHandler implements ActionListener

```

```

{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        String s1=inputArea.getText(); // for route number
        // String s2=inputArea2.getText();
        // System.out.println("The values taken are " + s1);
        try
        {
            Scanner scan=new Scanner(System.in);
            // Page3 ob=new Page3();
            //System.out.println("Transfer passed");//customer details

```

```
String name;

String contactNo;

String gender;

String email;

//int x,y;

int i=0;

int k=0;

int j = 0;

int l = 0;

int flag = 0;

//files of buses

FileInputStream f1 = new FileInputStream("VrlTesting.dat");

ObjectInputStream objectinput1 = new ObjectInputStream(f1);

ArrayList<Vrl> read1 = (ArrayList<Vrl>) objectinput1.readObject();

objectinput1.close();

FileInputStream f2 = new FileInputStream("SugamaTesting.dat");

ObjectInputStream objectinput2 = new ObjectInputStream(f2);

ArrayList<Sugama> read2 = (ArrayList<Sugama>)
objectinput2.readObject();

objectinput2.close();

FileInputStream f3 = new FileInputStream("DurgambaTesting.dat");

ObjectInputStream objectinput3 = new ObjectInputStream(f3);
```

```

        ArrayList<Durgamba> read3 = (ArrayList<Durgamba>)
objectinput3.readObject();

        objectinput3.close();

        Boolean ischanged=false;

        int flag1=0;

        int flag2=0;

        for(i=0;i<read1.size();i++)
        {

            Vrl obj1 = new Vrl();

            obj1 = (Vrl)read1.get(i);

            if(obj1.key.equals(s1))
            {

                if(!((obj1.source.equals(svar1))&&(obj1.destination.equals(dvar1))))
                {

                    JOptionPane.showMessageDialog(windowContent,"BUS NOT
PRESENT ON THIS ROUTE","ERROR",JOptionPane.ERROR_MESSAGE);

                    return;

                }

                //System.out.println("Entered the seatmatrix loop");

                // while(ischanged==false)

                //{

```

```

if(obj1.seatmatrix[Integer.parseInt(rowno)][Integer.parseInt(colno)]=='X')
    {

        ischanged=true;

        //f1(s1,s2);

        //System.out.println("Your seat is "+obj.seat);

    }
else
    {

        //System.out.println("Occupied book again");

        JOptionPane.showMessageDialog(windowContent,"Occupied Book
Again","ERROR",JOptionPane.ERROR_MESSAGE);

    }

    //if(ischanged==true)

    // f1(s1,s2);

    //}

    // break;

}

}

if(flag1==0)

```

```

{
    for(i=0;i<read3.size();i++)
    {

        Durgamba obj1 = new Durgamba();
        obj1 = (Durgamba)read3.get(i);
        if(obj1.key.equals(s1))
        {
            // while(ischanged==false)
            //{
                if(!((obj1.source.equals(svar1))&&(obj1.destination.equals(dvar1))))
                {
                    JOptionPane.showMessageDialog(windowContent,"BUS NOT
PRESENT ON THIS ROUTE","ERROR",JOptionPane.ERROR_MESSAGE);
                    return;
                }

            if(obj1.seatmatrix[Integer.parseInt(rowno)][Integer.parseInt(colno)]=='X')
                {

                    ischanged=true;
                    // f1(s1,s2);

```

```

        //System.out.println("Your seat is "+obj.seat);

    }

    else

    {

        System.out.println("Occupied book again");

        JOptionPane.showMessageDialog(windowContent,"Occupied Book
Again","ERROR",JOptionPane.ERROR_MESSAGE);

    }

    // }

    // if(ischanged==true)

    //     fl(s1,s2);

    //break;

    }

}

if(flag2==0)

{

    for(i=0;i<read2.size();i++)

    {

        Sugama obj1 = new Sugama();

```

```

obj1 = (Sugama)read2.get(i);
if(obj1.key.equals(s1))
{
    if(!((obj1.source.equals(svar1))&&(obj1.destination.equals(dvar1))))
    {
        JOptionPane.showMessageDialog(windowContent,"BUS NOT
PRESENT ON THIS ROUTE","ERROR",JOptionPane.ERROR_MESSAGE);
        return;
    }

    //System.out.println("Entered the seatmatrix loop");
    // while(ischanged==false)
    //{

if(obj1.seatmatrix[Integer.parseInt(rowno)][Integer.parseInt(colno)]=='X')
    {

        ischanged=true;
        //  f1(s1,s2);

        //System.out.println("Your seat is "+obj.seat);

    }

```



```

        else

        {

            // System.out.println("Occupied book again");

            JOptionPane.showMessageDialog(windowContent,"Occupied Book
Again","ERROR",JOptionPane.ERROR_MESSAGE);

        }

    //}

    //if(ischanged==true)

        //f1(s1,s2);

        //break;

    }          //END OF IF

    }

} //END OF IF

if(ischanged==true)

{

    f1(s1);

}

}

```

```

    } catch (IOException ex) {
        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Page2.class.getName()).log(Level.SEVERE, null,
ex);
    }
    catch(NullPointerException ex)
    {
        JOptionPane.showMessageDialog(windowContent, "Invalid Entry");
    }
    catch(NumberFormatException e2)
    {
        JOptionPane.showMessageDialog(windowContent, "Invalid Entry");
    }

    //Page3 ob=new Page3();
    //  System.out.println("Sup");
    //  frame2.dispose();
}

void f1(String rno) throws IOException, ClassNotFoundException

```

```

{
    //rno=uniqkey

    // System.out.println("Printing to the screen");

    // System.out.println("Route is" +rno);

    FileInputStream f = new FileInputStream("Vr1Testing.dat");

    ArrayList<Vrl> read;

    ObjectInputStream objectinput = new ObjectInputStream(f);

    read = (ArrayList<Vrl>) objectinput.readObject();

    objectinput.close();

    FileInputStream f1 = new FileInputStream("DurgambaTesting.dat");

    ArrayList<Durgamba> read1;

    ObjectInputStream objectinput1 = new ObjectInputStream(f1);

    read1 = (ArrayList<Durgamba>) objectinput1.readObject();

    objectinput1.close();

    FileInputStream f2 = new FileInputStream("SugamaTesting.dat");

    ArrayList<Sugama> read2;

    ObjectInputStream objectinput2 = new ObjectInputStream(f2);

    read2 = (ArrayList<Sugama>) objectinput2.readObject();

    objectinput2.close();

    // BusMod.bookTicket("hey");

    int i=0;

    int flag=0,flag1=0;

    for(i=0;i<read.size();i++)

```

```

{
    // System.out.println("Entered the loop");

    if(rno.equals(read.get(i).key))
    {

//BusMod.bookTicket(rno,Integer.parseInt(rowno),Integer.parseInt(colno));

        Page3 ob=new
Page3(rno,Integer.parseInt(rowno),Integer.parseInt(colno),svar1,dvar1,date1);

        frame2.dispose();

        flag=1;

        break;

    }

}

if(flag==0)
{
    for(i=0;i<read1.size();i++)
    {

        // System.out.println("Entered the Durgamba loop");

        if(rno.equals(read1.get(i).key))
        {

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

            flag1=1;

            //

BusMod.bookTicket(rno,Integer.parseInt(rowno),Integer.parseInt(colno));

```

```

        Page3 ob=new
Page3(rno,Integer.parseInt(rowno),Integer.parseInt(colno),svar1,dvar1,date1);

        frame2.dispose();

        break;

    }

}

if(flag1==0)

{

    for(i=0;i<read2.size();i++)

    {

        //System.out.println("Entered the Sugama loop");

        if(rno.equals(read2.get(i).key))

        {

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

//BusMod.bookTicket(rno,Integer.parseInt(rowno),Integer.parseInt(colno));

            Page3 ob=new
Page3(rno,Integer.parseInt(rowno),Integer.parseInt(colno),svar1,dvar1,date1);

            frame2.dispose();

            break;

        }

    }

}

}

}

```

```
}
```

```
}
```

```
private class ButtonHandler5 implements ActionListener
```

```
{
```

```
    // String newSelection;
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        Main ob=new Main();
```

```
        frame2.dispose();
```

```
    }
```

```
}
```

```
/* public static void main(String[] args)
```

```
{
```

```
    //Page2 obj=new Page2();
```

```
    System.out.println("Hello");//Page2 ob=new Page2(); // TODO code  
application logic here
```

```
    }*/  
}
```

### **Page3.java**

```
import java.awt.*;  
import java.awt.event.*;  
import java.awt.event.ActionListener;  
import java.util.LinkedList;  
import javax.swing.*;  
import java.io.*;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import java.util.*;  
  
class Page3 {  
    JFrame frame3=new JFrame("ENTER PASSENGER DETAILS");  
    JPanel windowContent=new JPanel();  
    JButton back=new JButton("BACK");  
    JButton bookagain=new JButton("BOOK AGAIN");  
    JLabel name=new JLabel("NAME :");  
    JLabel email=new JLabel("E-MAIL :");  
    JLabel contactNo=new JLabel("CONTACT No. :");  
    JLabel gender=new JLabel("GENDER :");  
    JLabel age=new JLabel("AGE :");
```

```
JPanel panel2=new JPanel();
```

```
int X,Y;
```

```
TextField inputName=new TextField(10);
```

```
TextField inputEmail=new TextField(10);
```

```
TextField inputContact=new TextField(10);
```

```
TextField inputGender=new TextField(10);
```

```
TextField inputAge=new TextField(10);
```

```
Button confirm=new Button("CONFIRM");
```

```
ButtonHandler listener=new ButtonHandler();
```

```
ButtonHandler2 listener1=new ButtonHandler2();
```

```
ButtonHandler3 listener2=new ButtonHandler3();
```

```
int r,c;//r -row,c-column
```

```
//int x1;
```

```
int seat1;
```

```
String rt;//for route
```

```
int x;
```

```
String Bname;//for Bus name
```

```
String Starttime;//For Start time
```

```
String Endtime;//for End Time
```

```
String src;//for Source
```

```
String dest;//for Destination
```



```
String Btype;//for Bus Type

String mail;

String svar1,dvar1;

String date1;

public Page3(String rte,int row,int col,String svar,String dvar,String date)
{
    Toolkit toolKit = Toolkit.getDefaultToolkit();

    java.awt.Dimension screenSize =
    toolKit.getScreenSize();

    X = screenSize.height;

    Y = screenSize.width;

    frame3.setVisible(true);

    frame3.setSize(X,Y);

    frame3.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    svar1=svar;

    dvar1=dvar;

    date1=date;

    confirm.addActionListener(listener);

    back.addActionListener(listener1);

    bookagain.addActionListener(listener2);

    windowContent.setLayout(new FlowLayout());

    windowContent.add(name);

    windowContent.add(inputName);
```

```
windowContent.add(email);

windowContent.add(inputEmail);

windowContent.add(gender);

windowContent.add(inputGender);

windowContent.add(age);

windowContent.add(inputAge);

windowContent.add(contactNo);

windowContent.add(inputContact);

panel2.add(confirm);

panel2.add(back);

panel2.add(bookagain);

frame3.setLayout(new BorderLayout());

frame3.add(windowContent,BorderLayout.NORTH);

frame3.add(panel2,BorderLayout.SOUTH);

frame3.setVisible(true);

frame3.setBounds(0,0,2*X,Y-630);

frame3.setIconImage(new ImageIcon("C:\\\\Rohit\\\\Java
Stuff\\\\NetBeans\\\\busmodfinal\\\\bus.jpg").getImage());

//seat1=obj.seat;

rt=rte;

r=row;

c=col;

frame3.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
x=10;

//System.out.println("Route is "+rt);

// System.out.println("Seat is "+seat1);

//r=obj.x

//c=obj.y

}
```

private class ButtonHandler implements ActionListener

```
{

    String name,mail,gender,phno,age;

    @Override

    public void actionPerformed(ActionEvent e)

    {

        //String name,mail,gender,phno,age;

        name=inputName.getText();

        mail=inputEmail.getText();

        gender=inputGender.getText();

        phno=inputContact.getText();

        age=inputAge.getText();

        //BusMod.fl(name,mail,gender,phno,age,r,c); put when p's code comes

        //obj.name=inputName.getText();

        System.out.println(x);

    }

}
```

```
//JOptionPane.showMessageDialog(windowContent,display,"TICKET  
DETAILS",JOptionPane.PLAIN_MESSAGE);
```

```
try {  
    f1();  
} catch (IOException ex) {  
    Logger.getLogger(Page3.class.getName()).log(Level.SEVERE, null,  
ex);  
} catch (ClassNotFoundException ex) {  
    Logger.getLogger(Page3.class.getName()).log(Level.SEVERE, null,  
ex);  
}
```

```
String display="DATE: "+ date1+ "\n "+"NAME:  
"+inputName.getText()+ "\t GENDER: "+inputGender.getText()+ "\nSOURCE:  
"+src+ "\t DESTINATION: "+dest+"\n"+"ROUTE NO: "+ rt +"\t SEAT NO: "+  
seat1 + "\n" + "BUS NAME:" +Bname + "\t BUS TYPE: " + Btype +  
"\n\t\t\tThank you for Using our Software,Wish you a Happy Journey!";
```

```
JOptionPane.showMessageDialog(windowContent,display,"TICKET  
DETAILS",JOptionPane.PLAIN_MESSAGE);
```

```
}  
  
void f1()    throws IOException, ClassNotFoundException  
{  
    PassDtls ob=new PassDtls();  
    ob.name=name;  
    ob.contactno=phno;  
    ob.gender=gender;
```

```
ob.mail=mail;

ob.age=age;

ob.rte=rt;

//ob.seat=seat1;

ob.seatrow=r;

ob.seatcol=c;

// System.out.println(ob.seatrow + ob.seatcol);

//x=299;

if(ob.rte.charAt(0)=='v' || ob.rte.charAt(0)=='V')
{
    ob.seat=BusMod.bookTicket(rt, r, c);

    seat1=ob.seat;

    File f1=new File("VrlPassenger.dat");

    if(!f1.exists())
    {
        ArrayList<PassDtls>list=new ArrayList<PassDtls>();

        list.add(ob);

        OutputStream file1 = new FileOutputStream(f1);

        OutputStream buffer1 = new BufferedOutputStream(file1);

        ObjectOutput output1 = new ObjectOutputStream(buffer1);

        output1.writeObject(list);

        output1.close();

        buffer1.close();
```

```

        file1.close();

    }
else
{
    FileInputStream f = new FileInputStream(f1);

    ObjectInputStream objectinput1 = new ObjectInputStream(f);

    ArrayList<PassDtls> read1 = (ArrayList<PassDtls>)
objectinput1.readObject();

    objectinput1.close();

    f.close();

    read1.add(ob);

    OutputStream file1 = new FileOutputStream(f1);

    OutputStream buffer1 = new BufferedOutputStream(file1);

    ObjectOutput output1 = new ObjectOutputStream(buffer1);

    output1.writeObject(read1);

    output1.close();

    buffer1.close();

    file1.close();

}

File f2=new File("VRLTesting.dat") ;

{

    FileInputStream f = new FileInputStream(f2);

```

```

ObjectInputStream objectinput1 = new ObjectInputStream(f);
ArrayList<Vrl> read1 = (ArrayList<Vrl>) objectinput1.readObject();
objectinput1.close();
f.close();
int i=0;
for(i=0;i<read1.size();i++)
{
    if(rt.equals(read1.get(i).key))
    {
        Bname="VRL";
        Starttime=read1.get(i).starttime;
        Endtime=read1.get(i).endtime;
        src=read1.get(i).source;
        dest=read1.get(i).destination;
        Btype=read1.get(i).type;
        break;
    }
}

}

} // END OF IF WHICH TELLS YOU ITS VRL
else if(ob.rte.charAt(0)=='D' || ob.rte.charAt(0)=='d' )
{

```

```

ob.seat=BusMod.bookTicket(rt, r, c);

seat1=ob.seat;

File f1=new File("DurgambaPassenger.dat");

if(!f1.exists())

{

    ArrayList<PassDtls>list=new ArrayList<PassDtls>();

    list.add(ob);

    OutputStream file1 = new FileOutputStream(f1);

    OutputStream buffer1 = new BufferedOutputStream(file1);

    ObjectOutput output1 = new ObjectOutputStream(buffer1);

    output1.writeObject(list);

    output1.close();

    buffer1.close();

    file1.close();

}

else

{

    FileInputStream f = new FileInputStream(f1);

    ObjectInputStream objectinput1 = new ObjectInputStream(f);

    ArrayList<PassDtls> read1 = (ArrayList<PassDtls>)

objectinput1.readObject();

    objectinput1.close();

```



```

        f.close();

        read1.add(ob);

        OutputStream file1 = new FileOutputStream(f1);

        OutputStream buffer1 = new BufferedOutputStream(file1);

        ObjectOutput output1 = new ObjectOutputStream(buffer1);

        output1.writeObject(read1);

        output1.close();

        buffer1.close();

        file1.close();

    }

    File f2=new File("DurgambaTesting.dat") ;

    {

        FileInputStream f = new FileInputStream(f2);

        ObjectInputStream objectinput1 = new ObjectInputStream(f);

        ArrayList<Durgamba> read1 = (ArrayList<Durgamba>)
objectinput1.readObject();

        objectinput1.close();

        f.close();

        int i=0;

        for(i=0;i<read1.size();i++)

        {

            if(rt.equals(read1.get(i).key))

            {

```

```

        Bname="DURGAMBA";

        Starttime=read1.get(i).starttime;

        Endtime=read1.get(i).endtime;

        src=read1.get(i).source;

        dest=read1.get(i).destination;

        Btype=read1.get(i).type;

        break;

    }

}

}

}

```

```

else if(ob.rte.charAt(0)=='S' || ob.rte.charAt(0)=='s' )

```

```

{

    ob.seat=BusMod.bookTicket(rt, r, c);

    seat1=ob.seat;

    File f1=new File("SugamaPassenger.dat");

    if(!f1.exists())

    {

        ArrayList<PassDtls>list=new ArrayList<PassDtls>();

        list.add(ob);

        OutputStream file1 = new FileOutputStream(f1);
    }
}

```

```

        OutputStream buffer1 = new BufferedOutputStream(file1);
        ObjectOutputStream output1 = new ObjectOutputStream(buffer1);
        output1.writeObject(list);
        output1.close();
        buffer1.close();
        file1.close();

    }
    else
    {
        FileInputStream f = new FileInputStream(f1);
        ObjectInputStream objectinput1 = new ObjectInputStream(f);
        ArrayList<PassDtls> read1 = (ArrayList<PassDtls>)
objectinput1.readObject();
        objectinput1.close();
        f.close();
        read1.add(ob);
        OutputStream file1 = new FileOutputStream(f1);
        OutputStream buffer1 = new BufferedOutputStream(file1);
        ObjectOutputStream output1 = new ObjectOutputStream(buffer1);
        output1.writeObject(read1);
        output1.close();
        buffer1.close();
    }
}

```

```

        file1.close();
    }
File f2=new File("SugamaTesting.dat") ;
{
    FileInputStream f = new FileInputStream(f2);
    ObjectInputStream objectinput1 = new ObjectInputStream(f);
    ArrayList<Sugama> read1 = (ArrayList<Sugama>)
objectinput1.readObject();
    objectinput1.close();
    f.close();
    int i=0;
    for(i=0;i<read1.size();i++)
    {
        if(rt.equals(read1.get(i).key))
        {
            Bname="Sugama";
            Starttime=read1.get(i).starttime;
            Endtime=read1.get(i).endtime;
            src=read1.get(i).source;
            dest=read1.get(i).destination;
            Btype=read1.get(i).type;
            break;
        }
    }
}

```

```

        }
    }

    }    //END OF IF WHICH TELLS YOU ITS SUGAMA

}

}

}

/* public static void main(String[] args)
{
    //PassDtls ob=new PassDtls();
    //Page3 obj=new Page3(ob);
    //System.out.println("Hello");

    //System.out.println(obj.a);//Page2 ob=new Page2(); // TODO code
application logic here
}*/

private class ButtonHandler2 implements ActionListener
{
    // String newSelection;

    @Override
    public void actionPerformed(ActionEvent e)
    {

```

```

try {
    Page2 ob=new Page2(svar1,dvar1,date1);
} catch (IOException ex) {
    Logger.getLogger(Page3.class.getName()).log(Level.SEVERE, null, ex);
} catch (ClassNotFoundException ex) {
    Logger.getLogger(Page3.class.getName()).log(Level.SEVERE, null, ex);
}
frame3.dispose();

}

}

```

```

private class ButtonHandler3 implements ActionListener
{
    // String newSelection;

    @Override
    public void actionPerformed(ActionEvent e)
    {
        Main ob=new Main();
        frame3.dispose();
    }
}

```

}

}

}

## Output Screenshots:

Welcome Page

\*\*\*\*\*WELCOME\*\*\*\*\*

Source  Destination

ENTER PASSENGER DETAILS

NAME :

kanagaraj

E-MAIL :

anagraj@gmail.com

GENDER :

male

AGE :

20

CONTACT No. :

99230570495

CONFIRM

BACK

BOOK AGAIN

Activate Windows

Go to PC settings







