

# DOCUMENTATION



## BUS TICKET RESERVATION SYSTEM

### USING JAVA 7

## Group Details :

### ● Members :-

#### ○ 1. SOMNATH GHOSE ROY (Group Leader)

NETAJI SUBHASH ENGINEERING COLLEGE, GARIA.

Registration No. : 111090110097

#### ○ 2. KUNDAN KUMAR

NETAJI SUBHASH ENGINEERING COLLEGE, GARIA.

Registration No. : 111090110354

#### ○ 3. MANISH SINHA

NETAJI SUBHASH ENGINEERING COLLEGE, GARIA.

Registration No. : 111090110049

#### ○ 4. SABYASACHI PAHARI

NETAJI SUBHASH ENGINEERING COLLEGE, GARIA.

Registration No. : 111090110077

#### ○ 5. SAYANTAN MAJUMDER

NETAJI SUBHASH ENGINEERING COLLEGE, GARIA.

Registration No. : 111090110091

*J Roy Chowdhury*



# INDEX

## TOPICSPAGE NO.

1. ACKNOWLEDGEMENT	4-5
2. INTRODUCTION	6
4. REQUIREMENT SPECIFICATION	7-8
5. FUNCTIONAL REQUIREMENT	9
6. HARDWARE/SOFTWARE REQUIREMENTS	10
7. PAGE FLOW DIAGRAMS	11-16
8. SCREEN SHOTS	17-18
9. FUTURE SCOPE OF IMPROVEMENTS	19
10. CODES	20-95
11. CONCLUSION	96
12. CERTIFICATES	97-101

## ACKNOWLEDGEMENT

We are heartily thankful to Globsyn Skills for providing us such an amazing platform to enhance our knowledge in technical as well as professional fields. We thank the Institution for showing us the right path in developing our confidence and helping us discover new areas of professionalism. Globsyn Skills taught us the basics of coding and led us to its advance level so smoothly that it was fun programming.

We are thankful to the Globsyn Skills unit for arrangement of such a vibrant course. The faculties here are much inspiring and their guidance is trust worthy. They reciprocated what they had promised during Induction. Special thanks to our faculty Mrs. Indrani Kundu who led us to the right track from the very inception to the completion of this project. Her worthy teachings and fruitful guidance has led to the outcome of this project. She was very friendly as well as focussed on what to teach us.

We would like to thank each and all who guided us in making this project happen in time. We would thank our friends, teachers and guides for helping us accomplish this project. We would thank our

*J Roy Chowdhury*



Parents for giving us freedom to choose for the best and their moral support and encouragement at every given second we needed. At last but not the least, we thank the Almighty for His showers of blessings on us every moment.

- THANK YOU.

*J Roy Chowdhury*



# INTRODUCTION

This project is based on a scenario. In this scenario, customer wants to know bus services with their time and date when they decide to travel. They can do this by going to agency where they will get information about services, its departure time, departure date, destination time, suitable seats etc. If they decide to reserve a ticket, then Admin will take some personal information from customer and will reserve a seat that customer selects.

This project's aim is to satisfy a facility to reserve seats, and check cancellation of seats, different types of enquiries like seat status, bus enquiry and bus time.

*J Roy Chowdhury*



# REQUIREMENT SPECIFICATION

## **#Domain Description:**

- This project is basically a proposal for issue and management of bus tickets based on the patent admin. The purpose of this report is to explain detailed description of bus reservation system. It will explain the aim of the project, interfaces of the system and what the project will do.
- The main player/actor of this system are customers and admin. Admin has access to all the fields and customer needs to get them verified by the admin.

## **Manual Problems solved by this system are:**

- 1) Decentralized issue of tickets at a large number of Counters managed by authorized agents at their expense.
- 2) Better capacity utilization of data and records.
- 3) Saves time.

- 4) Customer can buy ticket through the system and no need to queue up to buy bus ticket in the counter.
- 5) To provide anytime service for the customer.  
Customer can buy ticket 24\*7.
- 6) Customer can check the time departure and arrival for every bus through the system.

*J Roy Chowdhury*





## # FUNCTIONAL REQUIREMENTS

### Admin

- \* Add new bus
- \*Delete a bus
- \* View all database
- \* Confirm ticket
- \* Cancel reservation
- \*Modify service
- \*Cancel service

### CUSTOMER

- \* Make reservation
- \*Cancel reservation
- \* View tickets
- \* View available seats

*J Roy Chowdhury*



## **HARDWARE/SOFTWARE REQUIREMENTS**

### **MINIMUM HARDWARE REQUIRED:**

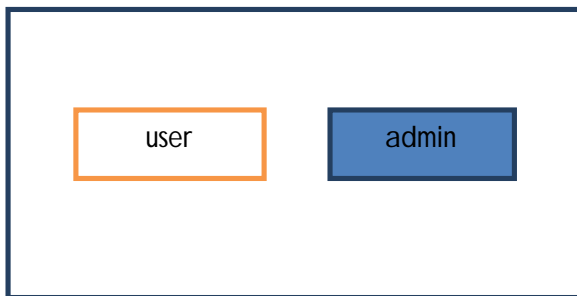
[FOR WINDOWS]

- 1 GB (RECOMMENDED RAM)
- 500 MB free (RECOMMENDED DISK SPACE)
- 1024x768 resolution (higher resolution recommended), with at least 256 colours.

### **SOFTWARE REQUIREMENTS:**

<u>OPERATING SYSTEM</u>	JVM (JDK INSTALLED)
Windows Server 2000 SP4+ Windows 2000 Advanced Server SP4+ Windows Server 2003 Windows XP Pro SP1+	J2SE 1.4.2_10, J2SE 5_06

## PAGE FLOW DIAGRAMS



A form diagram enclosed in an orange border. It contains the following elements:  
- Text: 'Enter user name:' followed by an orange-bordered input field.  
- Text: 'Enter password:' followed by an orange-bordered input field.  
- A blue button labeled 'HELP' and an orange-bordered button labeled 'SUBMIT' positioned below the input fields.

CHECK YOUR SOURCES PROPERLY

Enter username:

Enter password:

BUS RELATED	CONFIRMATION
ADD/REMOVE A BUS	
VIEW ALL BUSES	

BUS RELATED	CONFIRMATION
ADD/REMOVE A BUS	
VIEW ALL BUSES	

**ADD/REMOVE BUS:**

BUS ID	
BUS NAME	
SOURCE	
DESTINATION	
TYPE	
DEPT_TIME	
AARRIVAL_TIME	

**ADD**

**REMOVE**

**REFRESH**

**SAVE**

BUS RELATED  
ADD/REMOVE A  
VIEW ALL BUSES

CONFIRMATION

ID	BUS NAME	SOURCE	DESTINATION	TYPE	DEPT_TIME	ARRIVAL TIME	PRICE
1	Volvo	Patna	Kolkata	Ac	4pm	12am	700
2	Deluxe	Ranchi	Tata	Non_ac	5pm	11pm	300

ADMIN

CUSTOMER

BOOKING

CANCEL

## **BOOKING:**

TICKET NO.

FROM

TO

SEARCH

BUS TYPE

SHOW FARE

BUS FARE

DATE OF

SHOW SEAT

SEAT

SEAT REQUIRED

CONFIRM

*J Roy Chowdhury*



**BOOK**

**CANCEL**

CUSTOMER\_ID

CANCEL

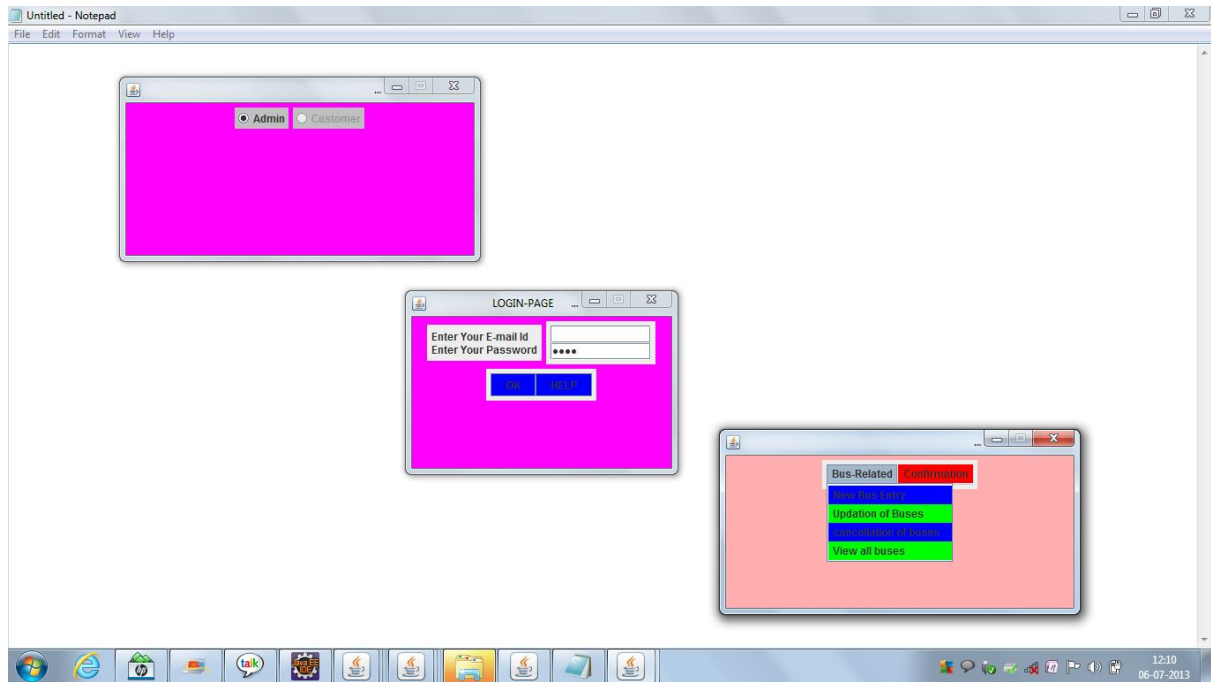
*J Roy Chowdhury*



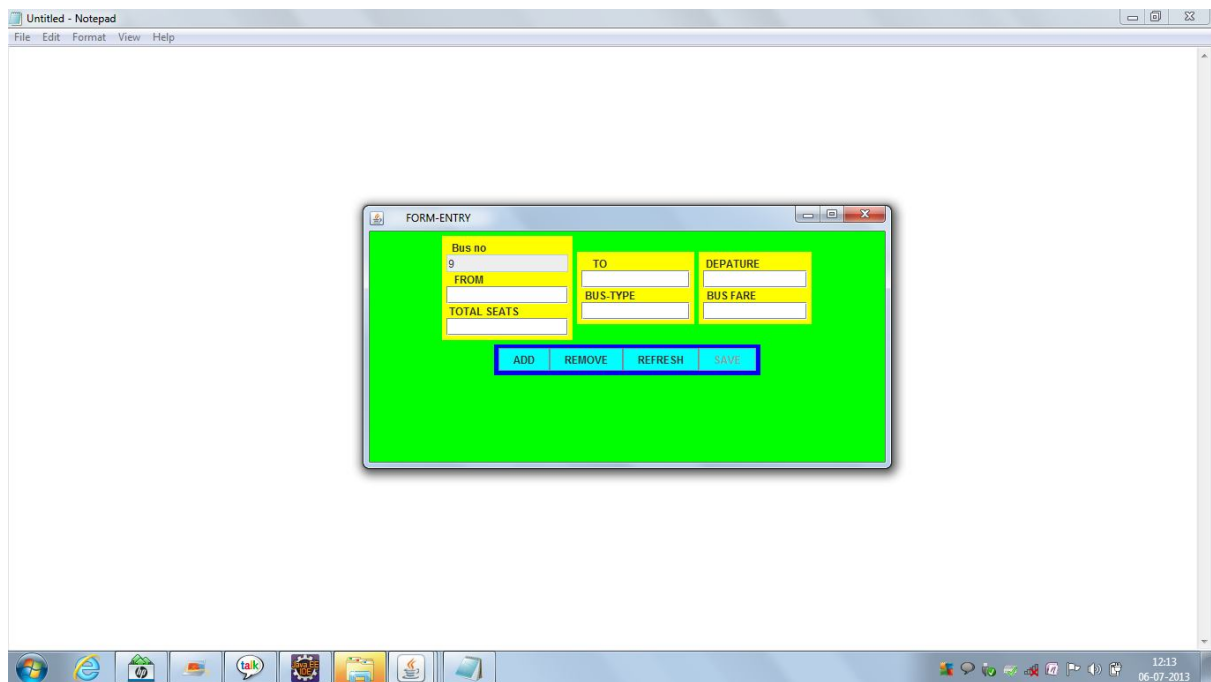


# SCREEN SHOTS

1.



2.



3.

view all buses

	Bus-No	from	To	Bus-Type	DATE	Total seats	Fare
1		delhi	kolkata	delux	1.1.13	23	124
2		Digha	Esplanade	Volvo 1	4.07.2013	34	180
3		Mandarmoni	Kharagpur	Delux 3	5.07.2013	45	120
4		Ranchi	Patna	Volvo 4	6.07.2013	56	100
5		Bankura	Burdwan	Delux 2	7.07.2013	57	60
6		Durgapur	Kolkata	Volvo 1	8.07.2013	60	120
7		Kharagpur	Kolaghat	Volvo 2	8.07.2013	50	100
8		Garia	Newtown	Delux 2	9.07.2013	55	30

*J Roy Chowdhury*



## **FUTURE SCOPE OF IMPROVEMENTS**

In future, there will be new works on this project.

\* Large number of Customers can access this software at the same time

\* Printing system will integrate this process.

*J Roy Chowdhury*



# CODES

```
package Traning;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ButtonGroup;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

import javax.swing.JRadioButton;

public class Welcome implements ActionListener {

    private JFrame f;
    private JPanel panel;
    private JRadioButton rdnadmin, rdncustomer;
    private ButtonGroup bg;
    public Welcome()
    {
        f=new JFrame();
        f.setLayout(new FlowLayout());
        ButtonGroup bg=new ButtonGroup();
        rdnadmin=new JRadioButton("Admin");
        rdncustomer=new JRadioButton("Customer");
        bg.add(rdnadmin);
        bg.add(rdncustomer);
        f.getContentPane().setBackground(Color.MAGENTA);
        rdnadmin.setBackground(Color.lightGray);
```

```

rdncustomer.setBackground(Color.lightGray);
f.setTitle("
WELCOME TO PHOTON TRAVELS");
f.add(rdnadmin);
f.add(rdncustomer);
f.setSize(400,200);
f.setResizable(false);
f.setVisible(true);
rdnadmin.addActionListener(this);
rdncustomer.addActionListener(this);
//panel.add(rdnadmin);
//panel.add(rdncustomer);
//f.add(panel, BorderLayout.CENTER);
//f.add(panelcenter, BorderLayout.EAST);

}

```

```

public void actionPerformed(ActionEvent event)
{

```

```

    if(event.getSource()==rdnadmin)
    {

```

```

        new LoginPage();

```

```

    }

```

```

    if(event.getSource()==rdncustomer)
    {

```

```

        new CustomerPanel();

```

```

    }
}

}

```

```

package Traning;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Box;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextArea;
import javax.swing.JTextField;
public class LoginPage extends ValidationChecking implements
ActionListener{
    JFrame f=new JFrame();
    private JLabel lblemail, lblpassword;
    protected JTextField txtemail;
    protected JPasswordField password;
    private JPanel panellabel, paneltext, panelbutton;
    private Box verticalboxlabel, verticalboxtext, horizontalbox;
    private JButton buttonok, buttonhelp;
    public LoginPage()
    {
        lblemail=new JLabel("Enter your e-mail Id");
        lblpassword=new JLabel("Enter your password");
        txtemail=new JTextField(10);
        password=new JPasswordField(0);
    }
}

```

```

panellabel=new JPanel();
paneltext=new JPanel();
panelbutton=new JPanel();
buttonok=new JButton("OK");
buttonhelp=new JButton("HELP");
verticalboxlabel=Box.createVerticalBox();
verticalboxtext=Box.createVerticalBox();
horizontalbox=Box.createHorizontalBox();
verticalboxlabel.add(lblemail);
verticalboxlabel.add(lblpassword);
verticalboxtext.add( txtemail);
verticalboxtext.add(password);
horizontalbox.add(buttonok);
horizontalbox.add(buttonhelp);
panellabel.add(verticalboxlabel);
paneltext.add(verticalboxtext);
panelbutton.add(horizontalbox);
f.add(panellabel, BorderLayout.WEST);
f.add(paneltext, BorderLayout.EAST);
f.add(panelbutton, BorderLayout.SOUTH);
f.setTitle("          LOGIN-PAGE          ");
f.setLayout(new FlowLayout());

f.getContentPane().setBackground(Color.MAGENTA);
lblemail.setBackground(Color.CYAN);
lblpassword.setBackground(Color.CYAN);
buttonok.setBackground(Color.BLUE);
buttonhelp.setBackground(Color.BLUE);

f.setSize(300,200);
f.setVisible(true);
f.setResizable(false);
f.setBackground(Color.BLUE);
buttonok.addActionListener(this);
buttonhelp.addActionListener(this);

```

```
}
```

```
public void actionPerformed(ActionEvent event)
```

```
{
```

```
    if(event.getSource()==buttonok)
```

```
    {
```

```
        String email=new String(txtemail.getText());
```

```
        String pwd=new String(password.getPassword());
```

```
        //isEmpty(pwd);
```

```
        //isEmpty(email);
```

```
        validationchecking(pwd,email, f);
```

```
    }
```

```
    if(event.getSource()==buttonhelp)
```

```
    {
```

```
        JOptionPane.showMessageDialog(f, " Check your sources  
properly");
```

```
    }
```

```
}
```

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

```
{
```

```
    new LoginPage();
```



```
}  
}
```

```
package Traning;
```

```
import java.awt.BorderLayout;  
import java.awt.Color;  
import java.awt.FlowLayout;  
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;
```

```
import javax.swing.JFrame;  
import javax.swing.JMenu;  
import javax.swing.JMenuBar;  
import javax.swing.JMenuItem;  
import javax.swing.JPanel;
```

```
import DataHandling.ViewAllBuses;
```

```
public class AdministerLogin implements ActionListener{
```

```
    JFrame f;  
    private JMenuBar menubar;  
    private JMenu menubusrelated,menubusconfirmation;  
    private JMenuItem  
    busentry,busupdate,buscancellation,busview,busviewroutes;  
    private JPanel panel ;
```

```

public AdministerLogin()
{

    f=new JFrame();
    menubar=new JMenuBar();

    menubusrelated=new JMenu("Bus-Related");

    menubusconfirmation=new JMenu("Confirmation");

    busentry=new JMenuItem("New Bus Entry");

    busupdate=new JMenuItem("Updation of Buses ");

    buscancellation=new JMenuItem("cancellation of buses");

    busview=new JMenuItem("View all buses");

    // busviewroutes=new JMenuItem("View all routes");
    panel=new JPanel();

    menubusrelated.add(busentry);

    menubusrelated.add(busupdate);

    menubusrelated.add(buscancellation);
    menubusrelated.add(busview);

    // menubusrelated.add(busviewroutes);

    menubar.add(menubusrelated);

```

```

menubar.add(menubusconfirmation);
panel.add(menubar);
f.add(panel, BorderLayout.CENTER);
f.setLayout(new FlowLayout());
f.getContentPane().setBackground(Color.PINK);
menubar.setBackground(Color.RED);
menubusrelated.setBackground(Color.BLUE);
menubusconfirmation.setBackground(Color.BLUE);
busentry.setBackground(Color.BLUE);
busupdate.setBackground(Color.GREEN);

buscancellation.setBackground(Color.BLUE);
busview.setBackground(Color.GREEN);
f.setTitle("
MENU---FORM");

```

```

f.setSize(400,200);
f.setResizable(false);

```

```

f.setVisible(true);
//register

```

```

busentry.addActionListener(this);

```

```

busupdate.addActionListener(this);

```

```

buscancellation.addActionListener(this);

```

```

busview.addActionListener(this);

```

```

// busviewroutes.addActionListener(this);

```

```

}

```

```

public void actionPerformed(ActionEvent event)
{

```

```
if(event.getSource()==busentry)
{
```

```
    new BusEntry();
}
```

```
if(event.getSource()==busupdate)
{
```

```
    new UpdationOfBuses();
}
```

```
if(event.getSource()==buscancellation)
{
```

```
    new CancellationOfBuses();
}
```

```
if(event.getSource()==busview)
{
```

```
    new ViewAllBuses();
}
```

```
}
```

}

**package** Traning;

**import** java.awt.BorderLayout;  
**import** java.awt.Color;  
**import** java.awt.Component;  
**import** java.awt.Container;  
**import** java.awt.FlowLayout;  
**import** java.awt.GridLayout;  
**import** java.awt.event.ActionEvent;  
**import** java.awt.event.ItemEvent;  
**import** java.awt.event.ActionListener;  
**//import** java.awt.event.ItemListener;

**import** javax.swing.Box;  
**import** javax.swing.JButton;  
**import** javax.swing.JComboBox;  
**import** javax.swing.JFrame;  
**import** javax.swing.JLabel;  
**import** javax.swing.JPanel;  
**import** javax.swing.JTextField;

**import** DataHandling.FileReadWrite;  
**import** DataHandling.ViewAllBuses;

**import** java.util.ArrayList;

**public class** BusEntry **extends** ValidationChecking **implements**  
ActionListener **//extends** ViewAllBuses  
{

```

private JFrame f;
private JLabel
lblfrom, lblto, lbldepart, lbltotalseat, lblbustype, lblbusfare, lblbusno;

public static int counter;
public static int n;

//public static int counter1=1;

public static int m;
protected JButton btnadd, btnremove, btnrefresh, btnsave;
protected JTextField
txtfrom, txtto, txtfare, txttotalseat, txtdepart, txtbustype, txtbusno;
    JPanel panelwest, panelcenter, paneleast, panelsouth;
    Box
verticalboxfrom, verticalboxto, verticalboxdepart, verticalboxtotalseat, v
erticalboxbustype, verticalboxbusno, verticalboxinputfrom, verticalboxi
nputto, verticalboxinputdepart, boxinput, verticalboxbusfare;
    private ArrayList<BusType> list;

```

```

public BusEntry ()

```

```

{

    list = new ArrayList<BusType>();
        //reading from file

try{

```

```

        list=FileReadWrite.readingFromBusType();

    }

    catch(Exception e)
    {
        System.out.println("File not found");

    }


    f=new JFrame();

    panelwest=new JPanel();
    paneleast=new JPanel();
    panelcenter=new JPanel();
    panelsouth=new JPanel();

    //f.setLayout(new FlowLayout());
    // f. setLayout(new GridLayout(19,10,12,12));

    // cmbday=new JComboBox();
    // cmdtime=new JComboBox();
    // busdepart=new JButton();
    btnadd=new JButton("ADD");
    btnremove=new JButton("REMOVE");
    btnrefresh=new JButton("REFRESH");
    btnsave=new JButton("SAVE");

    lblbusno=new JLabel("Bus no");
    txtbusno=new JTextField(10);

```

```
txtbustype=new JTextField(10);
lblfrom=new JLabel(" FROM ");
lblto=new JLabel("TO");
lbldepart=new JLabel("DEPARTURE");
lbltotalseat=new JLabel("TOTAL SEATS");
lblbusfare=new JLabel("BUS FARE");
txtfare=new JTextField(10);
lblbustype=new JLabel("BUS-TYPE");
txtfrom=new JTextField(10);
txtto=new JTextField(10);
txtfare=new JTextField(10);
txttotalseat=new JTextField(10);
txtdepart=new JTextField(10);
```

```
verticalboxfrom=Box.createVerticalBox();
verticalboxto=Box.createVerticalBox();
verticalboxdepart=Box.createVerticalBox();
verticalboxtotalseat=Box.createVerticalBox();
verticalboxbustype=Box.createVerticalBox();
verticalboxbusno=Box.createVerticalBox();
// verticalboxbustype=Box.createVerticalBox();
verticalboxinputfrom=Box.createVerticalBox();
verticalboxbusfare=Box.createVerticalBox();
verticalboxinputto=Box.createVerticalBox();

verticalboxinputdepart=Box.createVerticalBox();
boxinput=Box.createHorizontalBox();
```

```
boxinput.add(btnadd);
```

```
boxinput.add(btnremove);
```



boxinput.add(btnrefresh);

boxinput.add(btnsave);

verticalboxbusno.add(lblbusno);

verticalboxbusno.add(txtbusno);

verticalboxfrom.add(lblfrom);

verticalboxfrom.add(txtfrom);

verticalboxto.add(lblto);

verticalboxto.add(txtto);

verticalboxdepart.add(lbldepart);

verticalboxdepart.add(txtdepart);

verticalboxtotalseat.add(lbltotalseat);

verticalboxtotalseat.add(txttotalseat);

verticalboxbustype.add(lblbustype);

verticalboxbustype.add(txtbustype);

verticalboxbusfare.add(lblbusfare);

verticalboxbusfare.add(txtfare);

verticalboxinputfrom.add(verticalboxbusno);

verticalboxinputfrom.add(verticalboxfrom);

verticalboxinputfrom.add(verticalboxtotalseat);

verticalboxinputto.add(verticalboxto);

verticalboxinputto.add( verticalboxbustype);

```
verticalboxinputdepart.add(verticalboxdepart);  
verticalboxinputdepart.add(verticalboxbusfare);
```

```
panelwest.add(verticalboxinputfrom);  
panelcenter.add(verticalboxinputto);  
paneleast.add(verticalboxinputdepart);  
panelsouth.add(boxinput);  
f.setLayout(new FlowLayout());  
f.getContentPane().setBackground(Color.GREEN);
```

```
panelwest.setBackground(Color.YELLOW);  
panelcenter.setBackground(Color.YELLOW);  
paneleast.setBackground(Color.YELLOW);  
panelsouth.setBackground(Color.BLUE);  
btnadd.setBackground(Color.CYAN);  
btnremove.setBackground(Color.CYAN);  
btnrefresh.setBackground(Color.CYAN);  
btnsave.setBackground(Color.CYAN);
```

```
f.setTitle("    FORM-ENTRY    ");
```

```
f.add(panelwest, BorderLayout.WEST);  
f.add(panelcenter, BorderLayout.CENTER);  
f.add(paneleast, BorderLayout.EAST);  
f.add(panelsouth, BorderLayout.SOUTH);  
//f. setLayout(new GridLayout(6,6,12,12));
```

```
//boxfrom.setBounds(20,35,260,20);  
// boxto.setBounds(20,50,300,40);
```

```
//f.setBackground(Color.cyan);
```

```
f.setSize(600,300);  
f.setVisible(true);
```

```
// txtbusno.setText(name);
```

```
btnadd.addActionListener(this);//registering event source
```

```
btnremove.addActionListener(this);
```

```
btnrefresh.addActionListener(this);
```

```
btnsave.addActionListener(this);
```

```
btnsave.setEnabled(false);  
txtbusno.setText(String.valueOf(list.size()+1));
```

```
txtbusno.setEditable(false);
```

```
}
```

```
@Override
```

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

```

        if(event.getSource()==btnadd)

        {

            validationchecking1(txtfrom.getText(),txtto.getText(),txtfare.getText(),txttotalseat.getText(),txtdepart.getText(),txtbustype.getText(),txtbusno.getText());

            btnadd.setEnabled(false);
            btnremove.setEnabled(true);
            btnsave.setEnabled(true);
            list.add(getBustype());
            //m=Integer.parseInt(txtbusno.getText());
            //txtbusno.setText(String.valueOf(++m));
            //counter=Integer.parseInt(txtbusno.getText());

            refreshAll();

        }

        if(event.getSource()==btnremove)
        {

            btnremove.setEnabled(false);

            list.remove(list.size()-1);

            btnadd.setEnabled(true);

```

```

    }

    if(event.getSource()==btnrefresh)
    {

        refreshAll();

    }

    if(event.getSource()==btnsave)
    {
        btnsave.setEnabled(false);
        btnremove.setEnabled(false);
        btnadd.setEnabled(true);

    try
        {
            FileReadWrite.writingToBusType(list);

        }

        catch(Exception e)
        {

            System.out.println("Cannot write to
database");

        }

        txtbusno.setText(String.valueOf(list.size()+1));

```

```

    }
}

    public BusType getBustype()
    {

        int
busid=Integer.parseInt(txtbusno.getText().trim());

        String bustype=txtbustype.getText();    // Bus-
No"," Bus-Type","from","To","Time","Total seats","Fare"

        String from =txtfrom.getText();//String bustype,
String busfrom,
        //String busto, String busdepart, intbustotalseat,
intbusfare,
        //intbusid)

        String to=txtto.getText();
        String depart=txtdepart.getText();
        int
totalseat=Integer.parseInt(txttotalseat.getText().trim());

        int fare=Integer.parseInt(txtfare.getText().trim());

        BusType bus=new BusType(
bustype,from,to,depart,totalseat,fare,busid );

        return bus;
    }

    publicvoid refreshAll()
    {
        txtbusno.setText("");

```

```

        txtbustype.setText(" ");

        txtfrom.setText(" ");

        txtto.setText(" ");

        txtdepart.setText(" ");

        txttotalseat.setText(" ");


        txtfare.setText(" ");

    }
}

```

```

package Traning;

```

```

import java.io.Serializable;

```

```

public class BusType implements Serializable{

```

```

    private String bustype,busfrom,busto,busdepart;
    private int bustotalseat,busfare;
    private int busid;

```

*J Roy Chowdhury*



```

    public BusType( String bustype, String busfrom,
                    String busto, String busdepart, int bustotalseat, int
busfare,
                    int busid) {
        super();
        this.busdepart=busdepart;
        this.bustype = bustype;
        this.busfrom = busfrom;
        this.busto = busto;
        this.bustotalseat = bustotalseat;
        this.busfare = busfare;
        this.busid = busid;
    }
    public String getBusdepart() {
        return busdepart;
    }
    public String getBustype() {
        return bustype;
    }
    public String getBusfrom() {
        return busfrom;
    }
    public String getBusto() {
        return busto;
    }

    public int getBustotalseat() {
        return bustotalseat;
    }
    public int getBusfare() {
        return busfare;
    }
    public int getBusid() {
        return busid;
    }
    public void setBustype(String bustype) {
        this.bustype = bustype;
    }

```



```

    }
    publicvoid setBusfrom(String busfrom) {
        this.busfrom = busfrom;
    }
    publicvoid setBusto(String busto) {
        this.busto = busto;
    }
    publicvoid setBusdepart(String busdepart) {
        this.busdepart = busdepart;
    }
    publicvoid setBustotalseat(int bustotalseat) {
        this.bustotalseat = bustotalseat;
    }
    publicvoid setBusfare(int busfare) {
        this.busfare = busfare;
    }
    publicvoid setBusid(int busid) {
        this.busid = busid;
    }
}

```

**package** Traning;

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

```
package Traning;
```

```
import java.awt.BorderLayout;  
import java.awt.Color;  
import java.awt.FlowLayout;  
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.ItemListener;  
import java.util.ArrayList;
```

```
import javax.swing.Box;  
import javax.swing.JButton;  
import javax.swing.JComboBox;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JPanel;  
import javax.swing.JTextField;
```

```
import DataHandling.FileReadWrite;
```

```

public class CancellationOfBuses extends JFrame implements
ActionListener
{

    //JFrame f;
    private JLabel lblfrom, lblto, lblbusno, lblblank;
    private JComboBox cbbusno;
    private JComboBox combosource, combodest;
    private JButton btnok, btnproceed;
    private JPanel
panelwest, panelcenter, panelsouth, paneleast, panel1;
    private String[][] data;
    private ArrayList<BusType> buslist;
    private String[] source, dest;
    private int counter=1; private String id;
    private String[] busid;
    //ActionListener cmbs = null, cmbto = null;
    Box verticalboxfrom, verticalboxto, verticalproceed,
verticalbox1, verticalboxid, verticalbox2, horizontalbox, verticalbox3;

    public CancellationOfBuses ()
    {

        buslist=new ArrayList<BusType>();
        try
        {
            buslist=FileReadWrite.readingFromBusType();
        }
        catch (Exception exp)
        {
            System.out.println("No Records");
        }
    }
}

```

```

source=new String[buslist.size()];
dest=new String[buslist.size()];
    /*list=new ArrayList<BusType>();
    try
    {
        list=FileReadWrite.readingFromBusType();

    }
    catch(Exception exp)
    {
        System.out.println("No Records");
    }

    data = new String[list.size()][6];*/
    int index=0,row=0,col=0,row1=0,i=0;
    for(index=0;index<=buslist.size()-1;index++)
    {
        BusType busdetails=buslist.get(index);

        source[index]=busdetails.getBusfrom();
        dest[index]=busdetails.getBusto();

    }

    combosource=new JComboBox(source);
    combodest=new JComboBox(dest);

    setLayout(new FlowLayout());
    lblfrom=new JLabel(" FROM ");
        //txtfrom=new JComboBox();
    lblto=new JLabel("TO");

```

```

//txtto=new JComboBox();
lblblank=new JLabel(" ");
lblbusno=new JLabel(" Bus No. ");
btnok=new JButton("ok");
btnproceed=new JButton("Proceed");
cbbusno=new JComboBox();
verticalboxfrom=Box.createVerticalBox();
verticalproceed = Box.createVerticalBox();
verticalbox1= Box.createVerticalBox();
verticalboxto=Box.createVerticalBox();
verticalboxid=Box.createVerticalBox();
verticalbox2= Box.createVerticalBox();
horizontalbox=Box.createHorizontalBox();
verticalboxfrom.add(lblfrom);
verticalboxfrom.add(combosource);
verticalproceed.add(lblblank);
verticalproceed.add(btnproceed);
verticalbox1.add(verticalboxfrom);
verticalbox1.add(verticalproceed);
verticalboxto.add(lblto);
verticalboxto.add(combodest);
verticalboxid.add(lblbusno);
verticalboxid.add(cbbusno);
verticalbox2.add(verticalboxto);
verticalbox2.add(verticalboxid);
horizontalbox.add(btnok);
panelwest=new JPanel();
paneleast=new JPanel();
panelsouth=new JPanel();
panelwest.add(verticalbox1);
paneleast.add(verticalbox2);
panelsouth.add(horizontalbox);
add( panelwest, BorderLayout.WEST);
add(paneleast, BorderLayout.EAST);
add(panelsouth, BorderLayout.PAGE_END);

getContentPane().setBackground(Color.CYAN);

```

```

panelwest.setBackground(Color.YELLOW);

paneleast.setBackground(Color.YELLOW);
panelsouth.setBackground(Color.BLUE);
btnproceed.setBackground(Color.LIGHT_GRAY);
btnok.setBackground(Color.ORANGE);
setSize(300,300);
setVisible(true);

setTitle("Cancel a Bus");
btnok.setEnabled(false);
btnproceed.addActionListener(this);
btnok.addActionListener(this);

}
//@Override
public void actionPerformed(ActionEvent event)
{
    int index=0,row=0,col=0,row1=0,i=1;

    if(event.getSource()==btnproceed)
    {
        btnok.setEnabled(true);
        String
source1=(String)(combosource.getSelectedItem());
        String dest1=(String)combodest.getSelectedItem();

        for(index=0;index<=buslist.size()-1;index++)
        {
            BusType busdetails=buslist.get(index);

            if(source1.equalsIgnoreCase(busdetails.getBusfrom())&&
dest1.equalsIgnoreCase(busdetails.getBusto()))
            {
                cbbusno.addItem(""+busdetails.getBusid());

```

```

    }

}
}

if(event.getSource()==btnok)
{
    String id=(String)(cbbusno.getSelectedItem());
    for(index=0;index<=buslist.size()-1;index++)
    {
        BusType busdetails=buslist.get(index);
        if(id.equals(""+busdetails.getBusid()))
        {
            buslist.remove(index);

        }

    }

try
    {

FileReadWrite.writingToBusType(buslist);

    }

    catch(Exception e)
    {

        System.out.println("Cannot write to
database");

    }

}

}
}
}

```

```
}  
}
```

**package** Traning;

**import** java.awt.BorderLayout;  
**import** java.awt.Color;  
**import** java.awt.FlowLayout;  
**import** java.awt.GridLayout;  
**import** java.awt.event.ActionEvent;  
**import** java.awt.event.ActionListener;  
**import** java.awt.event.ItemListener;  
**import** java.util.ArrayList;

**import** javax.swing.Box;  
**import** javax.swing.JButton;  
**import** javax.swing.JComboBox;  
**import** javax.swing.JFrame;  
**import** javax.swing.JLabel;  
**import** javax.swing.JPanel;  
**import** javax.swing.JTextField;

**import** DataHandling.SeatReadWrite;  
**import** DataHandling.TicketReadWrite;  
**import** DataHandling.ViewTicket;

**public class** CancelTicket **extends** JFrame **implements**  
ActionListener  
{



```

//JFrame f;
private JLabel lblid;
private JTextField txtid;
//private JComboBox combosource,combodest;
private JButton btncancel;
private JPanel panelnorth,panelsouth;
private String[][]data;
private ArrayList<Ticket>ticketlist;
private ArrayList<Seat1>seatlist;
private String[] source,dest;

private int counter=1;private String id;
private String[] busid;

Box vboxfrom,vboxto,vboxproceed,
vbox1,vboxid, vbox2, hbox, vbox3;

public CancelTicket()
{

    ticketlist=new ArrayList<Ticket>();
    try
    {
        ticketlist=TicketReadWrite.readingFromTicket();
    }
catch(Exception exp)
{
    System.out.println("No Records");
}
seatlist=new ArrayList<Seat1>();
try
{
    seatlist=SeatReadWrite.readingFromSeat1();
}
catch(Exception exp)

```

```
{  
System.out.println("No Records");  
}
```

```
setLayout(new FlowLayout());  
lblid=new JLabel(" Enter Your Ticket ID ");
```

```
btncancel=new JButton("Cancel Ticket");
```

```
txtid=new JTextField(20);
```

```
add(lblid);  
add(txtid);  
add(btncancel);  
getContentPane().setBackground(Color.CYAN);  
lblid.setBackground(Color.YELLOW);  
txtid.setBackground(Color.WHITE);  
btncancel.setBackground(Color.orange);
```

```
setSize(300,300);  
setVisible(true);
```

```
setTitle("Cancel Your Ticket");
```

```
btncancel.addActionListener(this);
```

```
}
```

```
//@Override
```

```
public void actionPerformed(ActionEvent event)
```

```

{
    int index=0,row=0,col=0,row1=0,i=1;
    if(event.getSource()==btncancel)
    {

        for(index=0;index<=ticketlist.size()-1;index++)
        {
            Ticket ticketdetails=ticketlist.get(index);

            int id=Integer.parseInt(txtid.getText());
            if(id==ticketdetails.getTicketno())
            {

                for(int index1=0;index1<=seatlist.size()-
1;index1++)
                {
                    Seat1 seat=seatlist.get(index1);

                    if(seat.getDate().equalsIgnoreCase(ticketdetails.getDate())&&se
at.getBusid()==ticketdetails.getBusid())
                    {
                        seat.setBkdseat(seat.getBkdseat()-
ticketdetails.getSeatreq());
                    }
                }

                ticketlist.remove(index);

            try
            {
                TicketReadWrite.writingToTicket(ticketlist);

```

```

    }

    catch(Exception e)
    {

        System.out.println("Cannot write to database");

    }

    break;

    }

}

    new ViewTicket();
}
}

}

```

```

package Traning;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;

```

```

import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

import javax.swing.*;

import DataHandling.FileReadWrite;

public class UpdateFare extends JFrame implements
ActionListener
{
    //JFrame f;
    private JTextField txtfare;
    private JLabel lblfare;
    private JButton btnupdate;
    private JPanel panelnorth, panelsouth;
    private Box horizontalbox1, horizontalbox2;
    private ArrayList<BusType> buslist;
    private String id1;

    public UpdateFare(String id)

    {
        id1=id;
        buslist=new ArrayList<BusType>();
        try
        {
            buslist=FileReadWrite.readingFromBusType();

        }
        catch (Exception exp)
        {
            System.out.println("No Records");
        }
        setLayout(new FlowLayout());
        lblfare=new JLabel("Enter the new fare : ");
    }

```

```

txtfare=new JTextField(10);
btnupdate=new JButton("Update");
horizontalbox1=Box.createHorizontalBox();
horizontalbox2=Box.createHorizontalBox();
horizontalbox1.add(lblfare);
horizontalbox1.add(txtfare);
horizontalbox2.add(btnupdate);
panelnorth=new JPanel();
panelsouth=new JPanel();
panelnorth.add(horizontalbox1);
panelsouth.add(horizontalbox2);
add( panelnorth, BorderLayout.NORTH);
add(panelsouth, BorderLayout.SOUTH);
getContentPane().setBackground(Color.MAGENTA);
btnupdate.setBackground(Color.CYAN);
lblfare.setBackground(Color.PINK);
txtfare.setBackground(Color.WHITE);
panelnorth.setBackground(Color.BLUE);

panelsouth.setBackground(Color.BLUE);
setSize(300,300);
setVisible(true);

setTitle("Update a Bus");
btnupdate.addActionListener(this);

}

```

```

public void actionPerformed(ActionEvent event)
{
    if(event.getSource()==btnupdate)
    {

```

```

int index=0,row=0,col=0,row1=0,i=0;
for(index=0;index<=buslist.size()-1;index++)
    {
        BusType busdetails=buslist.get(index);

        if(id1.equals(""+busdetails.getBusid()))
        {
            buslist.set(index, setBusType(index));

        }

    }

try
    {
        FileReadWrite.writingToBusType(buslist);

    }

    catch(Exception e)
    {

        System.out.println("Cannot write to
database");

    }

}

}

}

private BusType setBusType(int index)
{
    BusType busdetails=buslist.get(index);

```

```

int busid=busdetails.getBusid();

String bustype=busdetails.getBustype();
String source=busdetails.getBusfrom();
String destination=busdetails.getBusto();
String departuretime=busdetails.getBusdepart();
int ts=busdetails.getBustotalseat();
int fare=Integer.parseInt(txtfare.getText());
int busfare=fare;

    BusType busdetails1=new
BusType(bustype,source,destination,departuretime,ts,busfare,busid);

    return busdetails1;
}
}

```

**package** Traning;

```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemListener;
import java.util.ArrayList;

import javax.swing.Box;
import javax.swing.JButton;

```



```

import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

import DataHandling.FileReadWrite;

public class UpdationOfBuses extends JFrame implements
ActionListener
{

    private JLabel lblfrom, lblto, lblbusno, lblblank;
    protected JComboBox cbbusno;
    private JComboBox combosource, combodest;
    private JButton btnok, btnproceed;
    private JPanel
panelwest, panelcenter, panelsouth, paneleast, panel1;
    private String[][] data;
    private ArrayList<BusType> buslist;
    private String[] source, dest;
    private String[] check;
    private String[] check1;
    private String id;
    private String[] busid;
    Box verticalboxfrom, verticalboxto, verticalproceed,
verticalbox1, verticalboxid, verticalbox2, horizontalbox, verticalbox3;

    public UpdationOfBuses ()
    {

        buslist = new ArrayList<BusType>();
        try
        {
            buslist = FileReadWrite.readingFromBusType();

```

```

    }
    catch(Exception exp)
    {
        System.out.println("No Records");
    }

    source=new String[buslist.size()];
    dest=new String[buslist.size()];
    //check=new String[buslist.size()];
    //check1=new String[buslist.size()];
    int index=0,row=0,col=0,row1=0,i=0;
    for(index=0;index<=buslist.size()-1;index++)
    {
        BusType busdetails=buslist.get(index);

        source[index]=busdetails.getBusfrom();
        dest[index]=busdetails.getBusto();

    }

    combosource=new JComboBox(source);
    combodest=new JComboBox(dest);
    setLayout(new FlowLayout());
    lblfrom=new JLabel(" FROM ");
    lblto=new JLabel("TO");
    lblblank=new JLabel(" ");
    lblbusno=new JLabel(" Bus No. ");
    btnok=new JButton("ok");
    btnproceed=new JButton("Proceed");
    cbbusno=new JComboBox();
    verticalboxfrom=Box.createVerticalBox();
    verticalproceed = Box.createVerticalBox();

```

```

verticalbox1= Box.createVerticalBox();
verticalboxto=Box.createVerticalBox();
verticalboxid=Box.createVerticalBox();
verticalbox2= Box.createVerticalBox();
horizontalbox=Box.createHorizontalBox();
verticalboxfrom.add(lblfrom);
verticalboxfrom.add(combosource);
verticalproceed.add(lblblank);
verticalproceed.add(btnproceed);
verticalbox1.add(verticalboxfrom);
verticalbox1.add(verticalproceed);
verticalboxto.add(lblto);
verticalboxto.add(combodest);
verticalboxid.add(lblbusno);
verticalboxid.add(cbbusno);
verticalbox2.add(verticalboxto);
verticalbox2.add(verticalboxid);
horizontalbox.add(btnok);
panelwest=new JPanel();
paneleast=new JPanel();
panelsouth=new JPanel();
panelwest.add(verticalbox1);
paneleast.add(verticalbox2);
panelsouth.add(horizontalbox);
add( panelwest,BorderLayout.WEST);
add(paneleast,BorderLayout.EAST);
add(panelsouth,BorderLayout.PAGE_END);

getContentPane().setBackground(Color.CYAN);
panelwest.setBackground(Color.YELLOW);

paneleast.setBackground(Color.YELLOW);
panelsouth.setBackground(Color.BLUE);
btnproceed.setBackground(Color.LIGHT_GRAY);
btnok.setBackground(Color.ORANGE);
setSize(300,300);
setVisible(true);

```

```

setTitle("Update Selection");
btnok.setEnabled(false);
btnproceed.addActionListener(this);
btnok.addActionListener(this);

}

```

```

public void actionPerformed(ActionEvent event)
{
    int index=0,row=0,col=0,row1=0,i=1;
    if(event.getSource()==btnproceed)
    {
        btnok.setEnabled(true);
        //busid=new String[70];

        for(index=0;index<=buslist.size()-1;index++)
        {
            BusType busdetails=buslist.get(index);
            String
source=(String)(combosource.getSelectedItem());
            String
dest=(String)combodest.getSelectedItem();
            if(source.equals(busdetails.getBusfrom())&&
dest.equals(busdetails.getBusto()))
            {

                cbbusno.addItem(""+busdetails.getBusid());

            }

        }
    }
}

```

```

        if(event.getSource()==btnok)
        {
            String id=(String)(cbbusno.getSelectedItem());
            new UpdateFare(id);
        }
    }
}

```

**package** Traning;

```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;

```

```

import DataHandling.ViewSeat;
import DataHandling.ViewTicket;

```

```

public class CustomerPanel extends JFrame implements
ActionListener {

```

```

JFrame f;
private JMenuBar menubar;
private JMenu menuticketinfo;
private JMenuItem
bookticket,cancelticket,viewticket,showseat;
private JPanel panel ;

public CustomerPanel()
{
    f=new JFrame();
    setLayout(new FlowLayout());
    menubar=new JMenuBar();

    menuticketinfo=new JMenu(" Ticket Informaion ");

    bookticket=new JMenuItem(" BOOK TICKET");
    cancelticket=new JMenuItem(" CANCEL TICKET ");
    viewticket=new JMenuItem(" VIEW TICKET ");

    showseat=new JMenuItem(" SHOW SEAT ");


    menuticketinfo.add(bookticket);
    menuticketinfo.add(cancelticket);
    menuticketinfo.add(viewticket);
    menuticketinfo.add(showseat);
    menubar.add(menuticketinfo);
    panel=new JPanel();
    panel.add(menubar);
    add(panel,BorderLayout.CENTER);
}

```

```
getContentPane().setBackground(Color.PINK);  
bookticket.setBackground(Color.YELLOW);  
cancelticket.setBackground(Color.CYAN);  
viewticket.setBackground(Color.YELLOW);  
showseat.setBackground(Color.CYAN);
```

```
setTitle(" Customer Panel ");  
setSize(700,300);  
setVisible(true);
```

```
bookticket.addActionListener(this);  
cancelticket.addActionListener(this);  
viewticket.addActionListener(this);  
showseat.addActionListener(this);  
}
```

@Override

```
publicvoid actionPerformed(ActionEvent event) {
```

```
    if(event.getSource()==bookticket)  
    {  
        new TicketField();  
    }  
    if(event.getSource()==cancelticket)  
    {  
        new CancelTicket();  
    }  
    if(event.getSource()==viewticket)  
    {  
        new ViewTicket();  
    }  
    if(event.getSource()==showseat)
```

```

        {
            new ViewSeat();
        }
    }
}

```

**package** Traning;

```

import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

```

```

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

```

**publicclass** CustomerField **implements** ActionListener{

```

    private JFrame f;
    protected JLabel
lblname, lbladdress, lblphoneno, lblbustype, lblgender, lblbusfare;
    protected JTextField txtname, txtaddress, txtphoneno;
    protected JButton btnok, btnrefresh;
    protected JComboBox combbustype, combgender, combbusfare;
    private ArrayList<BusType> buslist;
    String[] name= { "Male", "Female" };
    String[] bustype1;
    public CustomerField(){

```

```

        buslist=new ArrayList<BusType>();

```



```

f=new JFrame();
f.setLayout(new GridLayout(8,10,12,12));

bustype1=new String[buslist.size()];
int index=0,row=0,col=0;
for(index=0;index<=buslist.size()-1;index++)

{
    BusType busdetails=buslist.get(index);

    bustype1[index]=busdetails.getBustype();
}

combbustype=new JComboBox(bustype1);
btnok=new JButton("OK");
btnrefresh=new JButton("Refresh");
lblname=new JLabel(" NAME ");
lbladdress=new JLabel("Address");
lblphoneno=new JLabel("Phone No.");
lblbustype=new JLabel(" Bus Type ");
lblgender=new JLabel(" Gender ");
lblbusfare=new JLabel(" Bus Fare ");
txtname=new JTextField(10);
txtaddress=new JTextField(10);
txtphoneno= new JTextField(10);

combgender=new JComboBox(name);
combbusfare=new JComboBox();
f.add(lblname);
f.add(txtname);
f.add(lblgender);
f.add(combgender);

f.add(lbladdress);
f.add(txtaddress);

```

```

f.add(lblphoneno);
f.add(txtphoneno);
f.add(lblbustype);
f.add(combbustype);
f.add(lblbusfare);
f.add(combbusfare);
f.add(btnok);
f.add(btnrefresh);
btnrefresh.addActionListener(this);
f.setTitle("    Customer Field    ");
btnok .addActionListener(this);

f.setSize(300,300);
f.setVisible(true);

}
publicvoid actionPerformed(ActionEvent event) {

```

```

    if(event.getSource()==btnrefresh)
    {

        refreshAll();

    }
}
publicvoid refreshAll()
{
    txtname.setText(" ");

    txtaddress.setText(" ");

    txtphoneno.setText(" ");

```

```
        combbustype.setSelectedItem(" ");

        //txtdepart.setText(" ");

        //txttotalseat.setText(" ");


        // txtfare.setText(" ");

    }

}
```

```
package Traning;
import java.io.Serializable;

public class Seat1 implements Serializable{
    private int busid, bkdseat;
```

*J Roy Chowdhury*



```

    private String date;
    public Seat1(int busid, int bkdseat, String date) {
        super();
        this.busid = busid;
        this.bkdseat = bkdseat;
        this.date = date;
    }

    public int getBusid() {
        return busid;
    }
    public int getBkdseat() {
        return bkdseat;
    }
    public String getDate() {
        return date;
    }
    public void setBusid(int busid) {
        this.busid = busid;
    }
    public void setBkdseat(int bkdseat) {
        this.bkdseat = bkdseat;
    }
    public void setDate(String date) {
        this.date = date;
    }
}

package Traning;

import java.io.Serializable;

import javax.swing.JButton;

```

```
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class Ticket implements Serializable{

    private int ticketno; String from, to, bustype, date ;
    private int fare, seatreq, busid;

    public int getTicketno() {
        return ticketno;
    }

    public String getFrom() {
        return from;
    }

    public String getTo() {
        return to;
    }

    public String getBustype() {
        return bustype;
    }

    public String getDate() {
        return date;
    }

    public void setTicketno(int ticketno) {
        this.ticketno = ticketno;
    }
}
```

```
public void setFrom(String from) {  
    this.from = from;  
}
```

```
public void setTo(String to) {  
    this.to = to;  
}
```

```
public void setBustype(String bustype) {  
    this.bustype = bustype;  
}
```

```
public int getBusid() {  
    return busid;  
}
```

```
public void setBusid(int busid) {  
    this.busid = busid;  
}
```

```
public Ticket(int ticketno, String from, String to, String  
bustype,  
             String date, int fare, int seatreq, int busid) {  
    super();  
    this.ticketno = ticketno;  
    this.from = from;  
    this.to = to;  
    this.bustype = bustype;  
    this.date = date;  
    this.fare = fare;  
    this.seatreq = seatreq;  
    this.busid = busid;  
}
```

```

    public int getFare() {
        return fare;
    }

    public void setFare(int fare) {
        this.fare = fare;
    }

    public void setSeatreq(int seatreq) {
        this.seatreq = seatreq;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public int getSeatreq() {
        return seatreq;
    }

}

```

```

package Traning;

```

```

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

```

```
import javax.swing.JButton;  
import javax.swing.JComboBox;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JTextField;
```

```
import DataHandling.FileReadWrite;  
import DataHandling.SeatReadWrite;  
import DataHandling.TicketReadWrite;  
import DataHandling.ViewSeat;  
import DataHandling.ViewTicket;
```

```
public class TicketField extends ValidationChecking implements  
ActionListener{
```

```
    JFrame f=new JFrame();  
    protected JLabel  
lblfrom, lblto, lblticketno, lblbustype, lbldate, lblseatreq, lblbusfare, lblshow,  
lbltotalfare;  
    protected JTextField  
txtticketno, txtfare, txtdate, textseatreq, txtshow, txttotalfare;  
    protected JButton btnsearch, btnshowfare, btnshowseat, btnconfirm;  
    protected JComboBox combfrom, combto, combbustype;  
    private ArrayList<Seat1> seatlist;  
    private ArrayList<Ticket> ticketlist;  
    private ArrayList<BusType> buslist;  
    private String[] source, dest;  
    private int busid, ts, s;  
    private Object Seat1;  
  
    public TicketField()  
    {  
  
        buslist=new ArrayList<BusType>();
```



```

try{

    buslist=FileReadWrite.readingFromBusType();

    /* btnadd.setEnabled(true);
        btnsave.setEnabled(true);
        btnrefresh.setEnabled(false);*/

}

catch(Exception e)
{
    System.out.println("File not found");

}
System.out.println(buslist.size());
ticketlist=new ArrayList<Ticket>();

```

```

try{

    ticketlist=TicketReadWrite.readingFromTicket();

    /* btnadd.setEnabled(true);
        btnsave.setEnabled(true);
        btnrefresh.setEnabled(false);*/

}

catch(Exception e)
{
    System.out.println("File not found");
}

```

```

}
seatlist=new ArrayList<Seat1>();

try{

    seatlist=SeatReadWrite.readingFromSeat1();

    /* btnadd.setEnabled(true);
        btnsave.setEnabled(true);
        btnrefresh.setEnabled(false);*/

}

catch(Exception e)
{
    System.out.println("File not found");

}

source=new String[buslist.size()];
dest=new String[buslist.size()];
int index=0,row=0,col=0,row1=0,i=0;
for(index=0;index<=buslist.size()-1;index++)

{

    BusType busdetails=buslist.get(index);

    source[index]=busdetails.getBusfrom();
    dest[index]=busdetails.getBusto();
}

```

```

    }
    combfrom=new JComboBox(source);
    combto=new JComboBox(dest);

    f.setLayout(new FlowLayout());

    btnsearch=new JButton("Search");
    btnshowfare=new JButton("Show Fare");
    btnshowseat=new JButton("Show Seat
Availability");
    btnconfirm=new JButton("Confirm");

    lblfrom=new JLabel("    FROM    ");
    lblto=new JLabel("    To    ");
    lblticketno=new JLabel(" Ticket
No ");
    lblbustype=new JLabel(" Bus Type
");
    lbldate=new JLabel(" Date Of
Journey ");
    lblseatreq=new JLabel(" Seat
Required ");
    lblbusfare=new JLabel("    Bus
Fare ");
    lblshow=new JLabel("    Seat
Available ");
    lbltotalfare=new JLabel("    Total
Fare ");

    txtticketno=new JTextField(10);
    txtfare=new JTextField(10);
    txtdate=new JTextField(10);

```

```

textseatreq=new JTextField(10);
txtshow=new JTextField(10);
txttotalfare=new JTextField(10);

txtticketno.setText(String.valueOf(ticketlist.size()+1));

// combfrom=new JComboBox();
// combto=new JComboBox();
combbustype=new JComboBox();

f.add(lblticketno);
f.add(txtticketno);
f.add(lblfrom);
f.add(combfrom);
f.add(lblto);
f.add(combto);
f.add(btnsearch);

f.add(lblbustype);
f.add(combbustype);
f.add(btnshowfare);
f.add(lblbusfare);
f.add(txtfare);
f.add(lbldate);
f.add(txtdate);
f.add(btnshowseat);
f.add(lblshow);
f.add(txtshow);
f.add(lblseatreq);
f.add(textseatreq);
f.add(btnconfirm);
f.add(lbltotalfare);
f.add(txttotalfare);
btnsearch.addActionListener(this);

btnconfirm .addActionListener(this);
btnshowfare.addActionListener(this);
btnshowseat.addActionListener(this);

```

```

        getContentPane().setBackground(Color.ORANGE);
        lblfrom.setBackground(Color.CYAN);
        lblto.setBackground(Color.CYAN);
        lblticketno.setBackground(Color.CYAN);
        lblbustype.setBackground(Color.CYAN);
        lbldate.setBackground(Color.CYAN);
        lblseatreq.setBackground(Color.CYAN);
        lblbusfare.setBackground(Color.CYAN);
        lblshow.setBackground(Color.CYAN);
        lbltotalfare.setBackground(Color.CYAN);
        txtticketno.setBackground(Color.WHITE);
        txtfare.setBackground(Color.WHITE);
        txtdate.setBackground(Color.WHITE);
        textseatreq.setBackground(Color.WHITE);
        txtshow.setBackground(Color.WHITE);
        txttotalfare.setBackground(Color.WHITE);
        btnsearch.setBackground(Color.BLUE);
        btnshowfare.setBackground(Color.BLUE);
        btnshowseat.setBackground(Color.BLUE);
        btnconfirm.setBackground(Color.BLUE);
        combfrom.setBackground(Color.magenta);
        combto.setBackground(Color.magenta);
        combbustype.setBackground(Color.magenta);

```

```

        f.setTitle("    Ticket Field    ");
        f.setSize(690,300);
        f.setResizable(false);
        f.setVisible(true);

```

```

    }
    public void actionPerformed(ActionEvent event) {

```

```

        if(event.getSource()==btnsearch)
        {
            for(int index=0;index<=buslist.size()-1;index++)
            {
                BusType busdetails=buslist.get(index);

```

```

String source=(String)(combfrom.getSelectedItemAt());
String dest=(String)combto.getSelectedItemAt();
if(source.equals(busdetails.getBusfrom())&&
dest.equals(busdetails.getBusto()))
    {
        combbustype.addItem(""+busdetails.getBustype());
    }
}

}

if(event.getSource()==btnshowfare)
{
    txtticketno.setText(String.valueOf(ticketlist.size()+1));
    for(int index=0;index<=buslist.size()-1;index++)
    {
        BusType busdetails=buslist.get(index);
        String source=(String)(combfrom.getSelectedItemAt());
        String dest=(String)combto.getSelectedItemAt();
        String
bustype=(String)combbustype.getSelectedItemAt();

        if(source.equals(busdetails.getBusfrom())&&bustype.equals(bus
details.getBustype())&& dest.equals(busdetails.getBusto()))
            {
                busid=busdetails.getBusid();
                ts=busdetails.getBustotalseat();

                txtfare.setText(""+busdetails.getBusfare());
            }
    }
}

if(event.getSource()== btnshowseat)

```

```

{
    //new ViewSeat();
    //System.out.println(buslist.size());
    int flag=0;
    for(int index=0;index<=buslist.size()-1;index++)
    {
        BusType busdetails=buslist.get(index);
        //Seat1 seatdetails=seatlist.get(index);
        String source=(String)(combfrom.getSelectedItemAt());
        String dest=(String)combto.getSelectedItemAt();
        try
        {
            Seat1 seatdetails=seatlist.get(index);

            if(busid==seatdetails.getBusid()&&txtdate.getText().trim().equals(seatdetails.getDate()))
            {
                int availseat=ts-seatdetails.getBkdseat();
                if(availseat<0)
                    txtshow.setText(0+"");

                elsetxtshow.setText((ts-seatdetails.getBkdseat())+"");
                flag=1;

            }

            //if(txtdate.getText().equals(busdetails.getBusdepart()))

            //combbustype.addItem(""+busdetails.getBustype());

        }
    }
    catch(Exception e)
    {
        //System.out.println("File not found");
    }
}

```

```

    }

    }
    if(flag==0)
        txtshow.setText(""+ts);

}
if(event.getSource()== btnconfirm)
{

validationchecking2(f,txtdate.getText(),textseatreq.getText());
    //String from=(String) combfrom.getSelectedItem();

    //String to=(String)combto.getSelectedItem();

    // String bustype=(String)combbustype.getSelectedItem();

    //int fare=Integer.parseInt(txtfare.getText().trim());

    String date =txtdate.getText();

    int seatreq=Integer.parseInt(textseatreq.getText().trim());
    int flag=0;
    for(int index=0;index<=buslist.size()-1;index++)
    {
        try
        {
            Seat1 seatdetails=seatlist.get(index);

            if(busid==seatdetails.getBusid())&&txtdate.getText().trim().equals(seatdetails.getDate()))

```



```

        {
            seatdetails.setBkdseat(seatdetails.getBkdseat()+seatreq);
            flag=1;
            break;
        }
    }
    catch(Exception e)
    {
        // System.out .println("File Not Found");
    }

    //String time=txtdepart.getText();
    // String totalseat=txttotalseat.getText();

    //String fare=txtfare.getText();
    }
    if (flag==0)
    {
        Seat1 seat=new Seat1(busid,seatreq,date);

        seatlist.add(getSeat());
    }
    try
    {
        SeatReadWrite.writingToSeat(seatlist);
    }

    catch(Exception e)
    {

        System.out.println("Cannot write to database");
    }

```

```

    }
    ticketlist.add(getTicket());
    try
    {
        TicketReadWrite.writingToTicket(ticketlist);
    }

    catch(Exception e)
    {

        System.out.println("Cannot write to database");

    }
    txttotalfare.setText(String.valueOf(Integer.parseInt(txtfare.getText())*Integer.parseInt(textseatreq.getText())));

}
}

private Seat1 getSeat()
{
    s=(Integer.parseInt(textseatreq.getText()));
    String date=txtdate.getText();
    //intbusid, intbkdseat, String date
    Seat1 seat1=new Seat1(busid,s,date);
    return seat1;
}
public Ticket getTicket()
{

    //ticketno,from,to,bustype,fare,date;
    int ticketno=Integer.parseInt(txtticketno.getText().trim());

    String from=(String) combfrom.getSelectedItem();

```

```

        String to=(String)combto.getSelectedItemAt();

        String bustype=(String)combbustype.getSelectedItemAt();

        int fare=Integer.parseInt(txtfare.getText().trim());

        String date =txtdate.getText();

        int seatreq=Integer.parseInt(textseatreq.getText().trim());


        //String time=txtdepart.getText();
        // String totalseat=txttotalseat.getText();


        //String fare=txtfare.getText();
        Ticket ticket=new
Ticket(ticketno,from,to,bustype,date,fare,seatreq,busid);


        return ticket;
    }

}

```

```

package Traning;

```

```

import javax.swing.JFrame;
import javax.swing.JOptionPane;

```

```

import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class ValidationChecking extends JFrame {
    public void validationchecking(String pwd,String
email,JFrame f)
    {
        if( email.matches("[a-z].+@.+\.[a-z]+"))
        {
            if(pwd.equals("1234"))

                new AdministerLogin();
                //JOptionPane.showMessageDialog(f, "You are
logged in");

            else

                JOptionPane.showMessageDialog(null, " Enter the
correct password ");
        }
        else
            JOptionPane.showMessageDialog(null, "Enter the
correct format of email-id ");

    }
    public void validationchecking1(String from,String to,String
fare,String totalseat,String depart,String bustype,String busno)
    {

```

```

        if
        (from.equals("")||to.equals("")||fare.equals("")||totalseat.equals("")||dep
        art.equals("")||bustype.equals("")||busno.equals("") )
        {

            JOptionPane.showMessageDialog(null, "Please enter
all the fields");

        }

    }

```

```

public void validationchecking2(JFrame f,String date,String
seat)
    {

        if(date.matches("[a-z]+")||seat.equals(""))
        {
            JOptionPane.showMessageDialog(f, "Please enter all
the fields correctly");
            new TicketField();

        }
    }
}

```

```

package DataHandling;
import java.util.ArrayList;

import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;

```

```

import Traning.BusType;

public class ViewAllBuses extends JFrame {
    private JTable table;
    private ArrayList<BusType>list;
    private String[] heading={"Bus-No","from","To"," Bus-
Type","DATE","Total seats","Fare"};
    private String[][]data;
    private JScrollPane pane;
    private int counter=1;
    public ViewAllBuses()
    {
        list=new ArrayList<BusType>();
        try
        {
            list=FileReadWrite.readingFromBusType();

        }
        catch(Exception exp)
        {
            System.out.println("No Records");
        }
        data=new String[list.size()][7];

//get object from list and convert to string

        int index=0,row=0,col=0;

        for(index=0;index<=list.size()-1;index++)
        {
            BusType bus=list.get(index);
            data[row][col]=getBusno();

            data[row][++col]=bus.getBusfrom();
            data[row][++col]=bus.getBusto();

```

```

        data[row][++col]=bus.getBustype();
        data[row][++col]=bus.getBusdepart();
        data[row][++col]=String.valueOf(bus.getBustotalseat());
        data[row][++col]=String.valueOf(bus.getBusfare());

        col=0;
        ++row;

    }

    table=new JTable(data,heading);
    pane=new JScrollPane(table);
    add (pane);
    setSize(5000,2000);

    setVisible(true);
    setTitle("view all buses");

}
protected String getBusno() {

    String    Busno=String.valueOf(counter);
    counter++;
    return Busno;

}
}
package DataHandling;

import java.util.ArrayList;

import javax.swing.JFrame;
import javax.swing.JScrollPane;

```

```
import javax.swing.JTable;
```

```
import Traning.BusType;
```

```
import Traning.Seat1;
```

```
public class ViewSeat extends JFrame {  
    private JTable table;  
    private ArrayList<Seat1> seatlist;  
    private String[] heading = { "Busid", "Bkdseat", "date" };  
    private String[][] data;  
    private JScrollPane pane;  
    private int counter = 1;  
    public ViewSeat()  
    {  
        seatlist = new ArrayList<Traning.Seat1>();  
        try  
        {  
            seatlist = SeatReadWrite.readingFromSeat1();  
        }  
        catch (Exception exp)  
        {  
            System.out.println("No Records");  
        }  
        data = new String[seatlist.size()][3];  
  
        //get object from list and convert to string  
  
        int index = 0, row = 0, col = 0;  
  
        for (index = 0; index <= seatlist.size() - 1; index++)  
        {  
            Traning.Seat1 seat = seatlist.get(index);  
            data[row][col] = String.valueOf(seat.getBusid());  
  
            data[row][++col] = String.valueOf(seat.getBkdseat());  
        }  
    }  
}
```



```

        data[row][++col]=(seat.getDate());

        col=0;
        ++row;

    }

    table=new JTable(data,heading);
    pane=new JScrollPane(table);
    add (pane);
    setSize(5000,2000);

    setVisible(true);
    setTitle("view all seats");

}
}

```

```

package DataHandling;
    import java.util.ArrayList;

    import javax.swing.JFrame;
    import javax.swing.JScrollPane;
    import javax.swing.JTable;

```

```

import Traning.BusType;
import Traning.Seat1;
import Traning.Ticket;

public class ViewTicket extends JFrame {
    private JTable table;
    private ArrayList<Ticket> ticketlist;
    // ticketno, String from, String to, String bustype,
    // String date, int fare, int seatreq) {
    private String[] heading = {"Ticket
No.", "From", "To", "Bustype", "Bus ID", "Date", "Fare", "Seats
Booked"};
    private String[][] data;
    private JScrollPane pane;
    private int counter = 1;
    public ViewTicket()
    {
        ticketlist = new ArrayList<Ticket>();
        try
        {

ticketlist = TicketReadWrite.readingFromTicket();

        }
        catch (Exception exp)
        {
            System.out.println("No Records");
        }
        data = new String[ticketlist.size()][8];

        // get object from list and convert to string

        int index = 0, row = 0, col = 0;

        for (index = 0; index <= ticketlist.size() - 1; index++)

```

```

        {
            Ticket ticket=ticketlist.get(index);
            data[row][col]=String.valueOf(ticket.getTicketno());
            data[row][++col]=ticket.getFrom();
            data[row][++col]=ticket.getTo();
            data[row][++col]=ticket.getBustype();
            data[row][++col]=""+ticket.getBusid();
            data[row][++col]=ticket.getDate();
            data[row][++col]=String.valueOf(ticket.getFare());

            data[row][++col]=String.valueOf(ticket.getSeatreq());

            col=0;
            ++row;
        }

        table=new JTable(data,heading);
        pane=new JScrollPane(table);
        add (pane);
        setSize(5000,2000);

        setVisible(true);
        setTitle("Your Ticket");

    }

}

```

```

package DataHandling;
import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

```

```

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.ArrayList;

import Traning.BusType;

public class FileReadWrite

{

    public static void writingToBusType(ArrayList<BusType> list)
throws Exception
    {
        ObjectOutputStream write;

        FileOutputStream file=new FileOutputStream("BusType");

        write=new ObjectOutputStream(file);

        write.writeObject(list);
    }

    public static ArrayList<BusType> readingFromBusType() throws
Exception
    {

        ObjectInputStream read;
        FileInputStream file=new FileInputStream("BusType");
        read=new ObjectInputStream(file);
    }
}

```

```
        ArrayList<Traning.BusType>list=(ArrayList<Traning.BusType  
>)read.readObject();
```

```
        return list;
```

```
    }
```

```
}
```

```
package DataHandling;
```

```
import java.io.FileInputStream;
```

```
import java.io.FileNotFoundException;
```

```
import java.io.FileOutputStream;
```

```
import java.io.IOException;
```

```
import java.io.ObjectInputStream;
```

```
import java.io.ObjectOutputStream;
```

```
import java.util.ArrayList;
```

```
import Traning.Seat1;
```

```
public class SeatReadWrite
```

```
{
```

```
publicstaticvoid writingToSeat(ArrayList<Seat1>seatlist)  
throws Exception
```

```
{  
    ObjectOutputStream write;  
  
    FileOutputStream file=new FileOutputStream("SeatDB");  
  
    write=new ObjectOutputStream(file);  
  
    write.writeObject(seatlist);  
}
```

```
publicstatic ArrayList<Seat1> readingFromSeat1() throws  
Exception
```

```
{  
  
    ObjectInputStream read;  
    FileInputStream file=new FileInputStream("SeatDB");  
    read=new ObjectInputStream(file);  
  
    ArrayList<Seat1>seatlist=(ArrayList<Seat1>)read.readObject();  
    return seatlist;  
  
}
```

```
}  
package DataHandling;
```

```
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.ObjectInputStream;  
import java.io.ObjectOutputStream;  
import java.util.ArrayList;
```

```
import Traning.Ticket;
```

```

//import Training.BusType;
import Traning.TicketField;

publicclass TicketReadWrite

{

    publicstaticvoid writingToTicket(ArrayList<Ticket>list)
throws Exception
    {
        ObjectOutputStream write;

        FileOutputStream file=new FileOutputStream("Ticket");

        write=new ObjectOutputStream(file);

        write.writeObject(list);
    }

    publicstatic ArrayList<Ticket> readingFromTicket() throws
Exception
    {

        ObjectInputStream read;
        FileInputStream file=newFileInputStream("Ticket");
        read=new ObjectInputStream(file);

        ArrayList<Ticket>list=(ArrayList<Ticket>)read.readObject();
        return list;
    }

}

```

## **CONCLUSION**

Nowadays, bus agencies are taking important role in transportation, and to make reservation reliable they need a strong system that they will make reservation easier, faster and safer. This project designed to meet requirements of a bus reservation System. We faced some difficulties in programming but we overcome these difficulties by discussing with our instructor.

Our project, make well connection between pages, roles and lead roles to enter correct information to forms. With this bus reservation system people can travel without any misunderstanding that occurs by system of agencies.

Our aim is to make passengers happy and do not make agencies embarrassed against customer. For that reasons, we tried to create strong system. We believe that we achieved our goals and developed our goals by coding.

*J. Roy Chowdhury*





# CERTIFICATE

This is to certify that Mr Manish Sinha of Netaji Subhash Engineering College, Registration number: 111090110049, has successfully completed a project on Bus Ticket Reservation System using Java J under the guidance of Mrs INDRANI KUNDU.

-----  
Mrs. Indrani Kundu

Globsyn Finishing School

*J Roy Chowdhury*



# CERTIFICATE

This is to certify that Mr Kundan Kumar of Netaji Subhash Engineering College, Registration number: 111090110354, has successfully completed a project on Bus Ticket Reservation System using Java J under the guidance of Mrs INDRANI KUNDU.

-----  
Mrs. Indrani Kundu

Globsyn Finishing School

*J Roy Chowdhury*



# CERTIFICATE

This is to certify that Mr Somnath Ghose Roy of Netaji Subhash Engineering College, Registration number: 111090110097, has successfully completed a project on Bus Ticket Reservation System using Java 7 under the guidance of Mrs INDRANI KUNDU.

-----  
Mrs. Indrani Kundu

Globsyn Finishing School

*J Roy Chowdhury*



# CERTIFICATE

This is to certify that Mr Sayantan Majumder of Netaji Subhash Engineering College, Registration number: 111090110091, has successfully completed a project on Bus Ticket Reservation System using Java 7 under the guidance of Mrs INDRANI KUNDU.

-----  
Mrs. Indrani Kundu

Globsyn Finishing School

*J Roy Chowdhury*



# CERTIFICATE

This is to certify that Mr Sabyasachi Pahari of  
Netaji Subhash Engineering College, Registration  
number: 111090110077, has successfully  
completed a project on Bus Ticket Reservation  
System using Java 7 under the guidance of Mrs  
INDRANI KUNDU.

-----  
Mrs. Indrani Kundu

Globsyn Finishing School

*J. Roy Chowdhury*

