**ACKNOWLEDGMENT**

In the accomplishment of this project successfully many people have best owned upon us their blessings and the heart pledged support, this time we are utilizing to thank people who have been concerned with our project.

Primarily we would thank god for being able to complete this project with success. Then we would like to thank our principal Mrs Meeta Jhakanwal and all the teachers, whose valuable guidance have been the once that helped us patch this project and make it full proof success their suggestions and instructions has served as the major contributor towards the completion of the project.

Then we would like to thank our parents and friends who have helped us with their valuable suggestion and guidance has been helpful in various phases of the completion of the project

Last but not the least we would like to thanks our team leader & members who have helped us a lot.

**CONTENTS**

**Sl.No. Titles Pg.No.**

1. **Introduction to project**
2. **Objective of project**
3. **System Analysis**
   1. Hardware & Software Requirements
   2. Feasibility Study
      1. Technical Feasibility
      2. Operational Feasibility
      3. Economic Feasibility
4. **System Design**
   1. Data Flow Diagram
      1. Zero Level or Context Level D.F.D.
      2. 1st Level D.F.D.
   2. E-R Diagram
5. **Table Description**
6. **Result and Discussion**
   1. Future Scope of the project
   2. Limitations of the proposed system
7. **Annexure**
   1. Module’s screenshots and coding
8. **Future Scope**
9. **Limitations**
10. **Bibliography**
11. **Conclusion**

**INTRODUCTION**

Spa management systems are the computerized systems that can handle the business management and increase our sales, enhance our customer service.

Spa management system enables us to organize the staff's schedule, customer's account, appointment book and many other functions. These spa management systems are the solution for the large business. By using these spa management systems we can easily know about the clients and their details about their style and preferences. Spa management system with a window based interface allows us to work more conveniently even if we are not known to computers. Due to window based interface it also allows us to open many windows at a time. In the spa management system there are different forms for different users with various privileges.

Most of the spa management system allows us to schedule appointments by status such as confirmed, arrived, and booked and others. We can also search for specific appointment to schedule by using the features in spa management system. A fine spa management system allows us to easily book multiple service packages from one screen.

With a spa management system we can know our clients better by listing their relationships to other clients. We can also distribute clients in the membership types. It also allows us to generate sales, and analysis reports as well as client management reports.

It gives us a way to easily process voids, returns, and split a payment between two methods that are cash & cheque.

In existing system, the efficiency is not high. The proposed system is so efficient so that we can quickly and easily add all types of information one time.

**OBJECTIVES**

* This system belongs to a service sector and hence has to be automated.
* To give better service. The main objective of this project is to cover all services being provided in a spa.
* It can service the organisation providing massage & therapies, Beauty treatments, weight loss treatments etc.

* Even though this system is developed as apart of course but would be Beneficial for other similar projects.

The main objective of the entire project is to automate the process of day to day activities of Spa, such type of activity is mentioned below:-

* Customer’s Information Entry activity.
* Member’s Information Entry activity.
* Massager Information Entry activity.
* Creating packages activity.
* Appointments Information Entry activity, Bookings Information Entry activity, and Cancellation Information Entry.
* Search activity and Login activity.
* Prepare various Reports like Daily Bookings reports, Daily Appointments Reports.

The other objectives of the project are following:-

* No data duplication.
* No Paper Work Required.
* Time Efficient.
* Cost Efficient.
* Automatic data validation.
* User friendly environment.
* Data security and reliability.
* Fast data insertion & retrieval.
* Easily check the record etc.

The limited time and resources have restricted us to incorporate in this project, so only main activities that are performed in a Spa Management System, but at most care has been taken to make the system efficient and user friendly. The Spa Management System has been designed to computerize the specific functions.

**ADVANTAGES OF THE PROPOSED SYSTEM**

* Fast Transaction Handling & User-friendly GUI interface.
* Data security and authorization.
* Quick and effective Inventory system.
* Capabilities for data backup and restoration
* Improve user interface, query and report feature etc.

**SOFTWARE AND HARDWARE REQUIREMENTS**

**Software Requirements**

The software is a set of procedure of coded information or a program which when fed into the computer hardware enables the computer to perform the various tasks. Software is like a current inside the wire, which cannot be seen but its effect can felt.

The minimum software requirements to use this package successfully without hampering the ideas with which user have planned before operating on it is as follows:-

**Operating System: -**Windows 7.

**Database System: -**Oracle 10g

**Front End: -Java**

**Application Software: -**Microsoft Word (for documentation purpose).

**Hardware Requirements**

The hardware requirements for implementing this project are as follows:

* Minimum Processor i3 or above
* Processor with a frequency of 2GHz or Higher
* **RAM :** A minimum of 2 GB
* **Hard Drive :** 500 GB for data backup
* **Printers:** Laser Printers

**FEASIBILITY STUDY**

**TECHNICAL STUDY:** The system which is required for running the software must be of high quality i.e. higher processors, colour laser printers for printing, proper back up system, etc. Since a Spa management is of luxurious importance and the income generated is also high. So affording a better and slightly more expensive system is feasible for the organisation.

**OPERATIONAL STUDY:** The front end of the system is Visual Basic

And is developed by using all the graphical components, hence it has

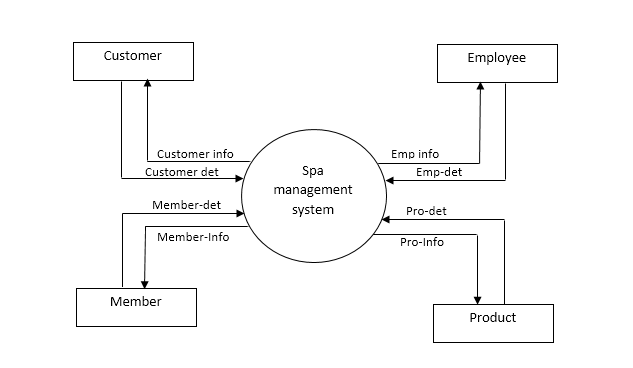
Ease of operations. The operating staff can easily manage the software

As everything is provided in menu, buttons, shortcut keys etc.

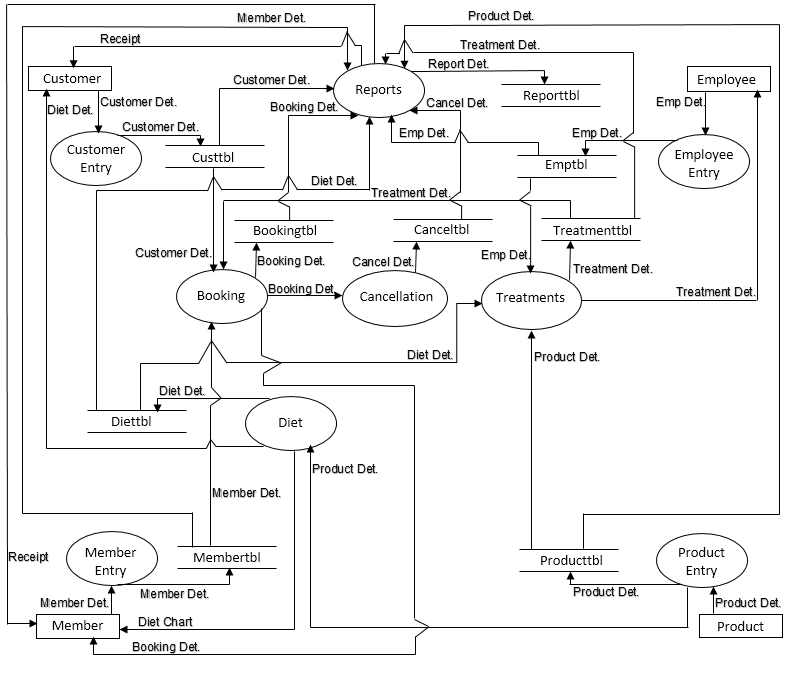
The everyday used operations are separately made available to the user.

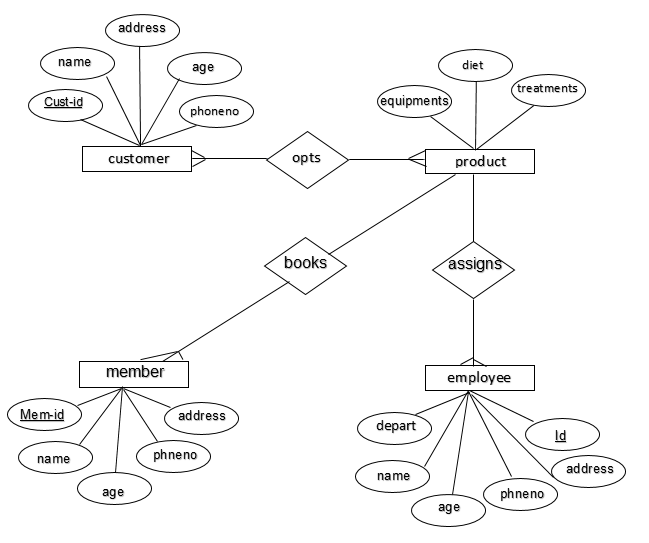
**ECONOMIC STUDY:** This system is implemented at organisations of luxurious importance. The affordability factor is of less importance, the costing is around 20,000 to 30,000 & is affordable by the organisation. The areas and the extent to which the automation is done surpasses the cost factor.

**DATA FLOW DIAGRAM**

CONTEXT LEVEL D.F.D

1st Level D.F.D

**E.R. DIAGRAM**



**DATABASE DESCRIPTON**

**Normalized Tables are as follows:**

TABLE NAME

* logintbl
* membertbl
* treatmenttbl
* customertbl
* bookingtbl
* canceltbl
* dietstbl
* dietttbl
* enquirytbl
* followtbl
* keytbl
* packagetbl
* paymenttbl
* scheduletbl
* stafftbl
* booktbl
* **logintbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| username | Text | 25 | PRIMARY KEY |
| password | Text | 25 | NOT NULL |
| usertype | Text | 20 | NOT NULL |

* **membertbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| mem\_date | Date/Time |  | NOT NULL |
| mem\_id | Text | 10 | PRIMARY KEY |
| mem\_name | Text | 50 | NOT NULL |
| mem\_category | Text | 15 | NOT NULL |
| mem\_gender | Text | 10 | NOT NULL |
| mem\_address | Text | 70 | NOT NULL |
| mem\_adhar | Number | 12 | NOT NULL |
| mem\_mob | Number | 10 | NOT NULL |
| menm\_email | Text | 30 | NOT NULL |
| mem\_joindate | Date/Time |  | NOT NULL |
| mem\_enddate | Date/Time |  | NOT NULL |
| mem\_weight | Number | 10 | NOT NULL |
| mem\_sugar | Number | 10 | NOT NULL |
| mem\_allergy | Text |  | NOT NULL |
| mem\_dis | Number | 10 | NOT NULL |

* **treatmenttbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| treat\_id | Text | 10 | PRIMARY KEY |
| treat\_type | Text | 20 | NOT NULL |
| treat\_name | Text | 50 | NOT NULL |
| t\_cost | Currency |  | NOT NULL |
| t\_duration | Number | 10 | NOT NULL |
| diettype | Text | 30 | NOT NULL |

* **customertbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| custdate | Date/Time |  | NOT NULL |
| custid | Text | 10 | PRIMARY KEY |
| custname | Text | 50 | NOT NULL |
| custaddress | Text | 70 | NOT NULL |
| custmob | Number | 10 | NOT NULL |
| custemail | Text | 50 | NOT NULL |
| custgender | Text | 10 | NOT NULL |

* **bookingtbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| bookdate | Date/Time |  | NOT NULL |
| bookid | Text | 10 | PRIMARY KEY |
| usertype | Text | 15 | NOT NULL |
| cust\_mem\_id | Text | 10 | NOT NULL |
| appointdate | Date/Time |  | NOT NULL |
| disptype | Text | 15 | NOT NULL |
| packid | Text | 10 | NOT NULL |
| discount | Number | 10 | NOT NULL |
| realcost | Currency |  | NOT NULL |
| totalcost | Currency |  | NOT NULL |
| massagerid | Text | 10 | NOT NULL |
| massagername | Text | 50 | NOT NULL |
| starttime | Date/Time |  | NOT NULL |
| endtime | Date/Time |  | NOT NULL |
| pystate | Text | 10 | NOT NULL |
| dietid | Text | 10 | NOT NULL |

* **canceltbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| canceldate | Date/Time |  | NOT NULL |
| bookdate | Date/Time |  | NOT NULL |
| canid | Text | 10 | PRIMARY KEY |
| bookid | Text | 10 | NOT NULL |
| cust\_mem\_id | Text | 10 | NOT NULL |
| appointdate | Date/Time |  | NOT NULL |
| treat\_pack\_id | Text | 10 | NOT NULL |
| totalcost | Currency |  | NOT NULL |
| paystat | Text | 15 | NOT NULL |
| ndays | Number | 10 | NOT NULL |
| deductfund | Currency |  | NOT NULL |
| refund | Currency |  | NOT NULL |
| pyamt | Currency |  | NOT NULL |
| staffid | Text | 10 | NOT NULL |
| stime | Date/Time |  | NOT NULL |
| etime | Date/Time |  | NOT NULL |

* **dietstbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| cust\_mem\_id | Text | 10 | NOT NULL |
| dietid | Text | 10 | NOT NULL |
| diettype | Text | 25 | NOT NULL |
| dietdate | Date/Time |  | NOT NULL |
| remarks | Text | 50 | NOT NULL |

* **diettbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| dietid | Text | 10 | PRIMARY KEY |
| diettype | Text | 25 | NOT NULL |
| foodallowed | Text | 100 | NOT NULL |
| foodnotallowed | Text | 100 | NOT NULL |
| foodlist | Text | 100 | NOT NULL |

* **enquirytbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| enqid | Text | 10 | PRIMARY KEY |
| enqname | Text | 50 | NOT NULL |
| enqdate | Date/Time |  | NOT NULL |
| enqaddress | Text | 50 | NOT NULL |
| enqgender | Text | 10 | NOT NULL |
| enqmobno | Number | 10 | NOT NULL |
| enqmail | Text | 50 | NOT NULL |
| enqpurpose | Text | 50 | NOT NULL |
| enqstatus | Text | 30 | NOT NULL |

* **followtbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| enquiryid | Text | 10 | NOT NULL |
| followid | Text | 10 | PRIMARY KEY |
| status | Text | 25 | NOT NULL |
| fdate | Date/Time |  | NOT NULL |
| fphone | Number | 10 | NOT NULL |
| fmail | Text | 50 | NOT NULL |

* **keytbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| custid | Number | 10 | NOT NULL |
| mem\_id | Number | 10 | NOT NULL |
| treat\_id | Number | 10 | NOT NULL |
| pack\_id | Number | 10 | NOT NULL |
| dietid | Number | 10 | NOT NULL |
| enqid | Number | 10 | NOT NULL |
| followid | Number | 10 | NOT NULL |
| bookingid | Number | 10 | NOT NULL |
| typeid | Number | 10 | NOT NULL |
| staffid | Number | 10 | NOT NULL |
| cancelid | Number | 10 | NOT NULL |
| paymentid | Number | 10 | NOT NULL |
| recno | Number | 10 | NOT NULL |

* **packagetbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| pack\_id | Text | 10 | PRIMARY KEY |
| pack\_name | Text | 50 | NOT NULL |
| treat1\_id | Text | 10 | NOT NULL |
| treat2\_id | Text | 10 | NOT NULL |
| treat3\_id | Text | 10 | NOT NULL |
| treat4\_id | Text | 10 | NOT NULL |
| treat5\_id | Text | 10 | NOT NULL |
| realcost | Currency |  | NOT NULL |
| discount | Number | 15 | NOT NULL |
| tcost | Currency |  | NOT NULL |
| ttax | Number | 15 | NOT NULL |

* **paymenttbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| payid | Text | 10 | PRIMARY KEY |
| bookid | Text | 10 | NOT NULL |
| cust\_mem\_id | Text | 10 | NOT NULL |
| treatid | Text | 10 | NOT NULL |
| pamount | Currency |  | NOT NULL |
| tamount | Currency |  | NOT NULL |
| dop | Date/Time |  | NOT NULL |
| chequeno | Number | 15 | NOT NULL |
| chequedt | Date/Time |  | NOT NULL |
| bankname | Text | 50 | NOT NULL |
| branch | Text | 25 | NOT NULL |

* **scheduletbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| staffid | Text | 10 | NOT NULL |
| bookid | Text | 10 | NOT NULL |
| custmemid | Text | 10 | NOT NULL |
| apdate | Date/Time |  | NOT NULL |
| starttime | Date/Time |  | NOT NULL |
| endtime | Date/Time |  | NOT NULL |

* **stafftbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| staffid | Text | 10 | PRIMARY KEY |
| sname | Text | 50 | NOT NULL |
| sgender | Text | 10 | NOT NULL |
| address | Text | 70 | NOT NULL |
| mobile | Number | 10 | NOT NULL |
| email | Text | 50 | NOT NULL |
| joindate | Date/Time |  | NOT NULL |
| speciality | Text | 50 | NOT NULL |
| previouswork | Text | 50 | NOT NULL |
| salary | Currency |  | NOT NULL |

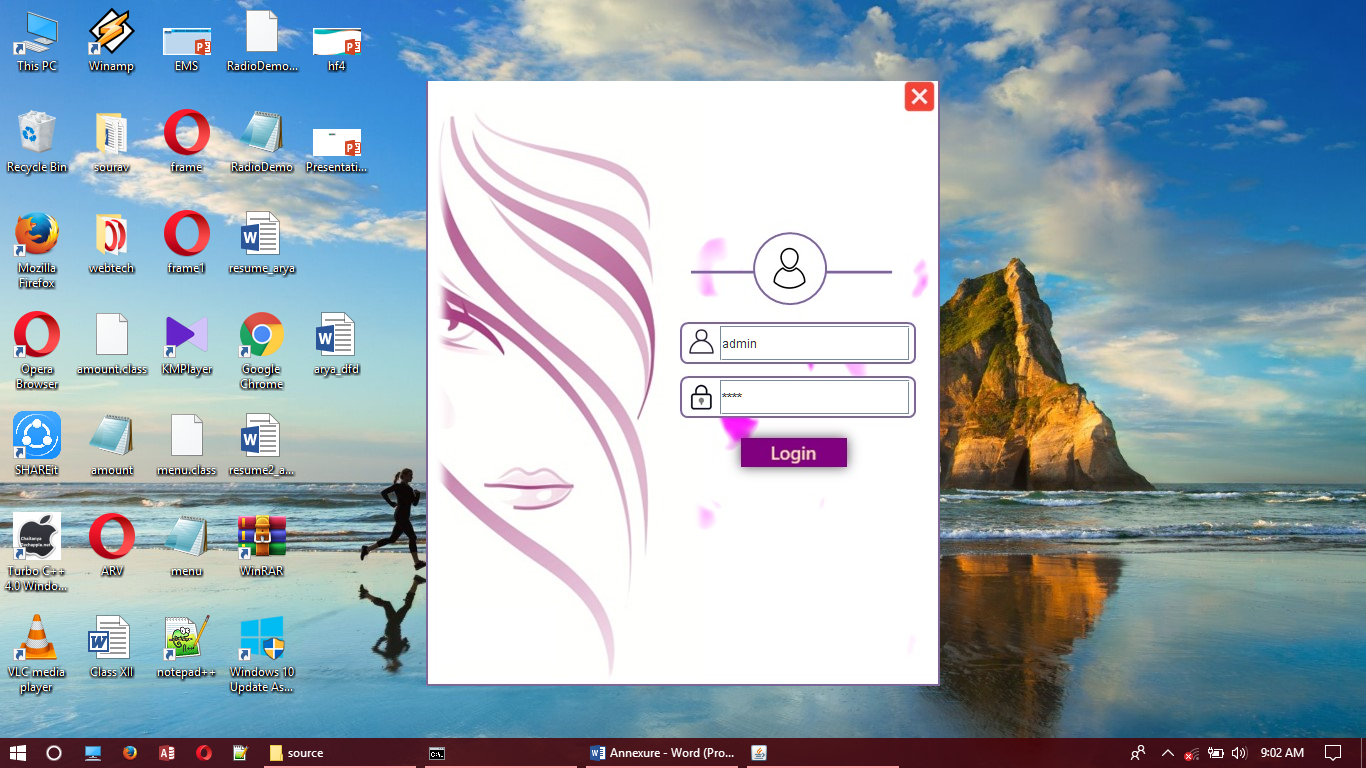
* **booktbl:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields Name** | **Type** | **Field Size** | **Constraint** |
| bookid | Text | 10 | NOT NULL |
| treatid | Text | 10 | NOT NULL |

**Annexure**

**Screenshots and coding**

* **Login:** Here an authorized person can only access the application by their username and password.



import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.io.\*;

import java.sql.\*;

public class login extends JFrame implements ActionListener

{

int c;

JTextField tuser=new JTextField();

JPasswordField tpswrd=new JPasswordField();

JButton sub=new JButton(new ImageIcon("../Images/loginbt.png"));

JButton cancel=new JButton(new ImageIcon("../Images/cancelbt.png"));

JPanel pnl=new JPanel();

public login()

{

c=0;

setLayout(null);

setVisible(true);

setSize(514,605);

pnl.add(lbl);

add(tpswrd);

add(tuser);

add(sub);

add(cancel);

lbl.setBounds(0,0,540,620);

tuser.setBounds(294,245,190,35);

tpswrd.setBounds(294,299,190,35);

sub.setBounds(317,358,104,27);

cancel.setBounds(480,2,27,27);

sub.addActionListener(this);

cancel.addActionListener(this);

}

public void splash(int flg)

{

this.dispose();

if(flg==1)

menu1 obj=new menu1();

else

menu obj=new menu();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==cancel)

{

System.exit(0);

}

if(e.getSource()==sub)

{

int ctr=0;

int flg;

try

{

String type="";

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

Statement st=con.createStatement();

String str="select \* from logintbl where username='"+tuser.getText()+"' and password='"+tpswrd.getText()+"'";

res=st.executeQuery(str);

while(res.next())

{

ctr++;

type=res.getString(3);

}

if(ctr==1)

{

if(type.equals("Admin"))

flg=1;

else

flg=0;

splash(flg);

break p;

}

else

{

JOptionPane.showMessageDialog(null,"Wrong UserName or Password");

c++;

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

}

public static void main(String argv[])

{

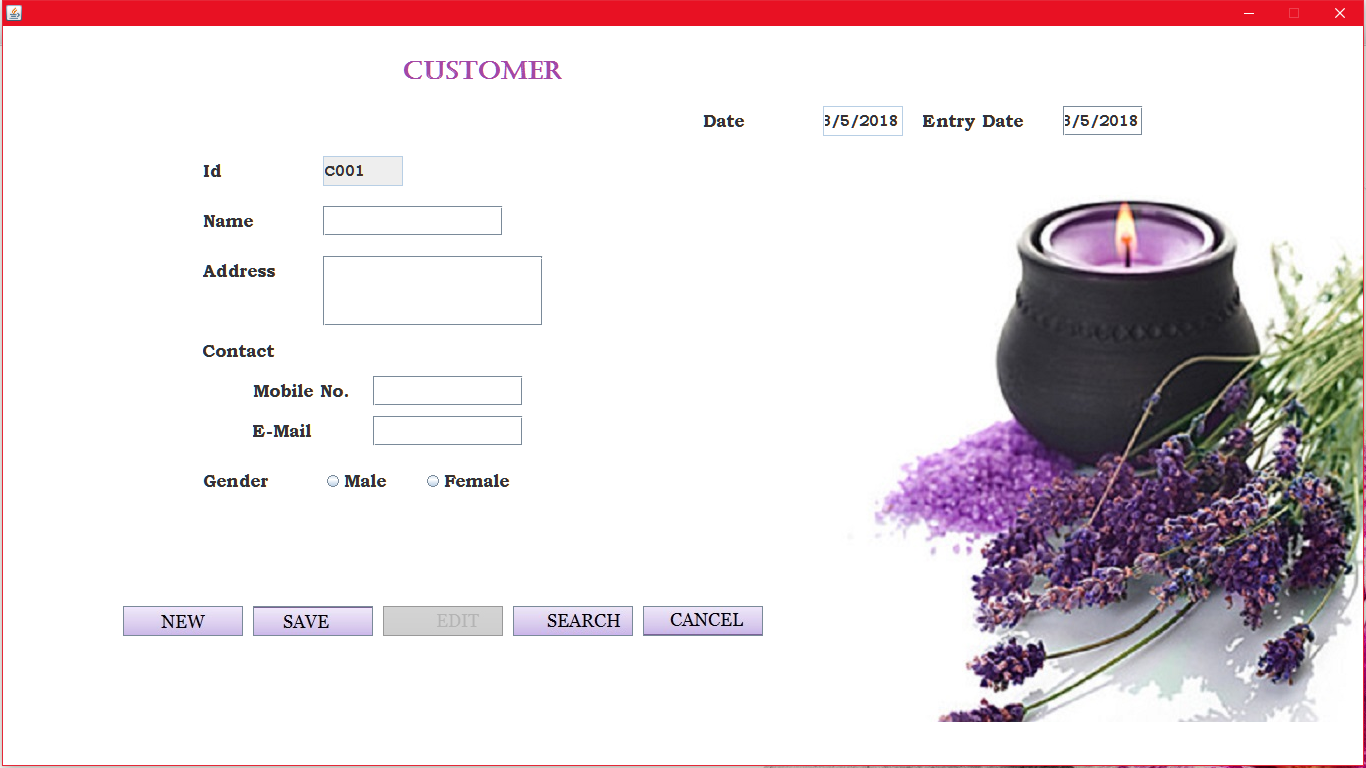
login obj=new login();

}

}

**Master Forms:-**

* **Customer:** Here a customer is a person who approaches to a Spa. He/she is the one who first enquires about different packages or treatments and finally chooses a package or treatment which he wants. In case if he/she did not find any package or treatment which he/she wants he can offer his own package. He/she is the person who pays all the bills.



import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.io.\*;

import java.sql.\*;

import java.util.\*;

public class customer extends JFrame implements ActionListener,FocusListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/mas.jpg"));

JLabel lcustid=new JLabel("Id");

JLabel lcustname=new JLabel("Name");

JLabel lcustaddress=new JLabel("Address");

JLabel lcustcontact=new JLabel("Contact");

JLabel lcustmob=new JLabel("Mobile No.");

JLabel lcustemail=new JLabel("E-Mail");

JLabel lcustgender=new JLabel("Gender");

JLabel ldate=new JLabel("Date");

JLabel lendate=new JLabel("Entry Date");

JLabel lhead=new JLabel("Customer");

JButton custnew=new JButton(new ImageIcon("../Images/mnbt.jpg"));

JButton custsave=new JButton(new ImageIcon("../Images/msbt.jpg"));

JButton custsearch=new JButton(new ImageIcon("../Images/mssbt.jpg"));

JButton custedit=new JButton(new ImageIcon("../Images/mebt.jpg"));

JButton custcancel=new JButton(new ImageIcon("../Images/mcbt.jpg"));

JTextField txdate=new JTextField();

JTextField txendate=new JTextField();

JTextField txid=new JTextField();

JTextField txname=new JTextField();

JTextArea txaddress=new JTextArea();

JTextField txemail=new JTextField();

JTextField txmob=new JTextField();

JRadioButton rmale =new JRadioButton("Male");

JRadioButton rfemale=new JRadioButton("Female");

ButtonGroup bg=new ButtonGroup();

String strgen="";

Calendar cal=Calendar.getInstance();

public customer()

{

setLayout(null);

setVisible(true);

getContentPane().setBackground(Color.WHITE);

lbl.setBounds(0,0,getWidth(),getHeight())

bg.add(rmale);

bg.add(rfemale);

add(txemail);

add(custcancel);

add(custsearch);

add(custsave);

add(custedit);

add(custnew);

add(txmob);

add(rmale);

add(rfemale);

add(lcustgender);

add(lcustcontact);

add(lcustmob);

add(lcustemail);

add(sradd);

add(lcustaddress);

add(txname);

add(txid);

txid.setEditable(false);

txdate.setEditable(false);

add(lcustid);

add(lcustname);

add(ldate);

add(txdate);

add(lendate);

add(txendate);

add(lhead);

lhead.setBounds(400,30,250,30);

lcustid.setBounds(200,130,100,30);

txid.setBounds(320,130,80,30);

ldate.setBounds(700,80,100,30);

txdate.setBounds(820,80,80,30);

lendate.setBounds(920,80,120,30);

txendate.setBounds(1060,80,80,30);

lcustname.setBounds(200,180,100,30);

txname.setBounds(320,180,180,30);

lcustaddress.setBounds(200,230,100,30);

sradd.setBounds(320,230,220,70);

lcustcontact.setBounds(200,310,120,30);

lcustmob.setBounds(250,350,120,30);

txmob.setBounds(370,350,150,30);

lcustemail.setBounds(250,390,120,30);

txemail.setBounds(370,390,150,30);

lcustgender.setBounds(200,440,100,30);

rmale.setBounds(320,440,80,30);

rfemale.setBounds(420,440,100,30);

custnew.setBounds(120,580,120,30);

custsave.setBounds(250,580,120,30);

custedit.setBounds(380,580,120,30);

custsearch.setBounds(510,580,120,30);

custcancel.setBounds(640,580,120,30);

txdate.setText(""+cal.get(Calendar.DATE)+"/"+(cal.get(Calendar.MONTH)+1)+"/"+cal.get(Calendar.YEAR));

custsave.setEnabled(false);

custedit.setEnabled(false);

rmale.addActionListener(this);

rfemale.addActionListener(this);

custsave.addActionListener(this);

custedit.addActionListener(this);

custnew.addActionListener(this);

custcancel.addActionListener(this);

custsearch.addActionListener(this);

txname.addFocusListener(this);

txmob.addFocusListener(this);

txemail.addFocusListener(this);

txendate.addFocusListener(this);

}

public void focusLost(FocusEvent fe)

{

int i,tflg=0;

if(fe.getComponent()==txname)

{

strchk=txname.getText();

if(txname.getText().length()==0)

else

tflg=1;

if(tflg==1)

{

for(i=0;i<strchk.length();i++)

{

if(strchk.charAt(i)>='a'&&strchk.charAt(i)<='z'||strchk.charAt(i)>='A'&&strchk.charAt(i)<='Z'||strchk.charAt(i)==' ')

{

}

else

{

JOptionPane.showMessageDialog(null,"Invalid Name!!!");

txname.setText("");

break;

}

}

}

}

if(fe.getComponent()==txmob)

{

int tmflg=0;

strchk=txmob.getText();

if(txmob.getText().length()==0)

else

tmflg=1;

if(tmflg==1)

{

a:{for(i=0;i<strchk.length();i++)

{ if(Character.getNumericValue(strchk.charAt(i))>=0&&Character.getNumericValue(strchk.charAt(i))<=9)

{

}

else

{

JOptionPane.showMessageDialog(null,"Enter Only Digits!!");

txmob.setText("");

break a;

}

if(strchk.length()!=10)

{

JOptionPane.showMessageDialog(null,"Enter 10-digits Mobile Number!!");

txmob.setText("");

break a;

}}

}

}

}

if(fe.getComponent()==txendate)

{

strchk=txendate.getText();

if(txendate.getText().length()!=0)

{

try

{

SimpleDateFormat cdf = new SimpleDateFormat("dd/MM/yyyy");

Date dat = cdf.parse(strchk);

if (!strchk.equals(cdf.format(dat)))

{

JOptionPane.showMessageDialog(null,"Invalid Date Format!!");

}

}

catch(Exception ee)

{

JOptionPane.showMessageDialog(null,"Invalid Date Format!!");

}

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==custnew)

{

int ctr=1;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(1));

}

if(ctr<10)

txid.setText("C00"+ctr);

if(ctr>=10&&ctr<100)

txid.setText("C0"+ctr);

txname.setText("");

txaddress.setText("");

txemail.setText("");

txmob.setText("");

txendate.setText(txdate.getText());

custsave.setEnabled(true);

custedit.setEnabled(false);

}

catch(Exception ee)

{

JOptionPane.showMessageDialog(null,ee);

}

}

if(e.getSource()==custsave)

{

int flg=0;

custedit.setEnabled(false);

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="insert into customertbl(custdate,custid,custname,custaddress,custmob,custemail,custgender)values(?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

if(txmob.getText().length()==0||txemail.getText().equals("")||txname.getText().length()==0||txaddress.getText().length()==0||bg.getSelection()==null)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

if(flg==0)

{

ps.setString(1,txendate.getText());

ps.setString(2,txid.getText());

ps.setString(3,txname.getText());

ps.setString(4,txaddress.getText());

if(rmale.isSelected())

strgen="Male";

if(rfemale.isSelected()) strgen="Female";

ps.setString(7,strgen);

ps.setString(5,txmob.getText());

ps.setString(6,txemail.getText());

ps.executeUpdate();

String str1="update keytbl set custid=custid+1";

Statement st=con.createStatement();

st.executeUpdate(str1);

JOptionPane.showMessageDialog(null,"SAVED...");

custsave.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==custedit)

{

custsave.setEnabled(false);

custedit.setEnabled(false);

int flg=0,tflg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

if(txid.getText().length()!=0&&sflg==1)

{

if(rmale.isSelected())

strgen="Male";

if(rfemale.isSelected())

strgen="Female";

if(txendate.getText().length()==0||txmob.getText().length()==0||txemail.getText().equals("")||txname.getText().length()==0||txaddress.getText().length()==0||bg.getSelection()==null)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

if(flg==0)

{

String str="Update customertbl set custdate='"+txendate.getText()+"',custname='"+txname.getText()+"',custaddress='"+txaddress.getText()+"',custgender='"+strgen+"',custmob='"+txmob.getText()+"',custemail='"+txemail.getText()+"' where custid='"+txid.getText()+"'";

st.executeUpdate(str);

JOptionPane.showMessageDialog(null,"Updated");

}

}

else

{

JOptionPane.showMessageDialog(null,"Search Record for update!!");

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==custsearch)

{

custsave.setEnabled(false);

sflg=1;

int flg=0;

String t=JOptionPane.showInputDialog("Enter Customer id to search");

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

Statement st=con.createStatement();

res=st.executeQuery("select \* from customertbl where custid='"+t+"'");

while(res.next())

{

custedit.setEnabled(true);

flg=1;

txendate.setText(res.getString(1));

txname.setText(res.getString(3));

txid.setText(res.getString(2));

txaddress.setText(res.getString(4));

txmob.setText(res.getString(5));

txemail.setText(res.getString(6));

if(res.getString(7).equals("Male"))

{

rmale.setSelected(true);

rfemale.setSelected(false);

}

else

{

rmale.setSelected(false);

rfemale.setSelected(true);

}

}

if(flg==0&&t.length()!=0)

{

JOptionPane.showMessageDialog(null,"No Record Found!!");

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==custcancel)

{

this.dispose();

}

}

public static void main(String argv[])

throws IOException

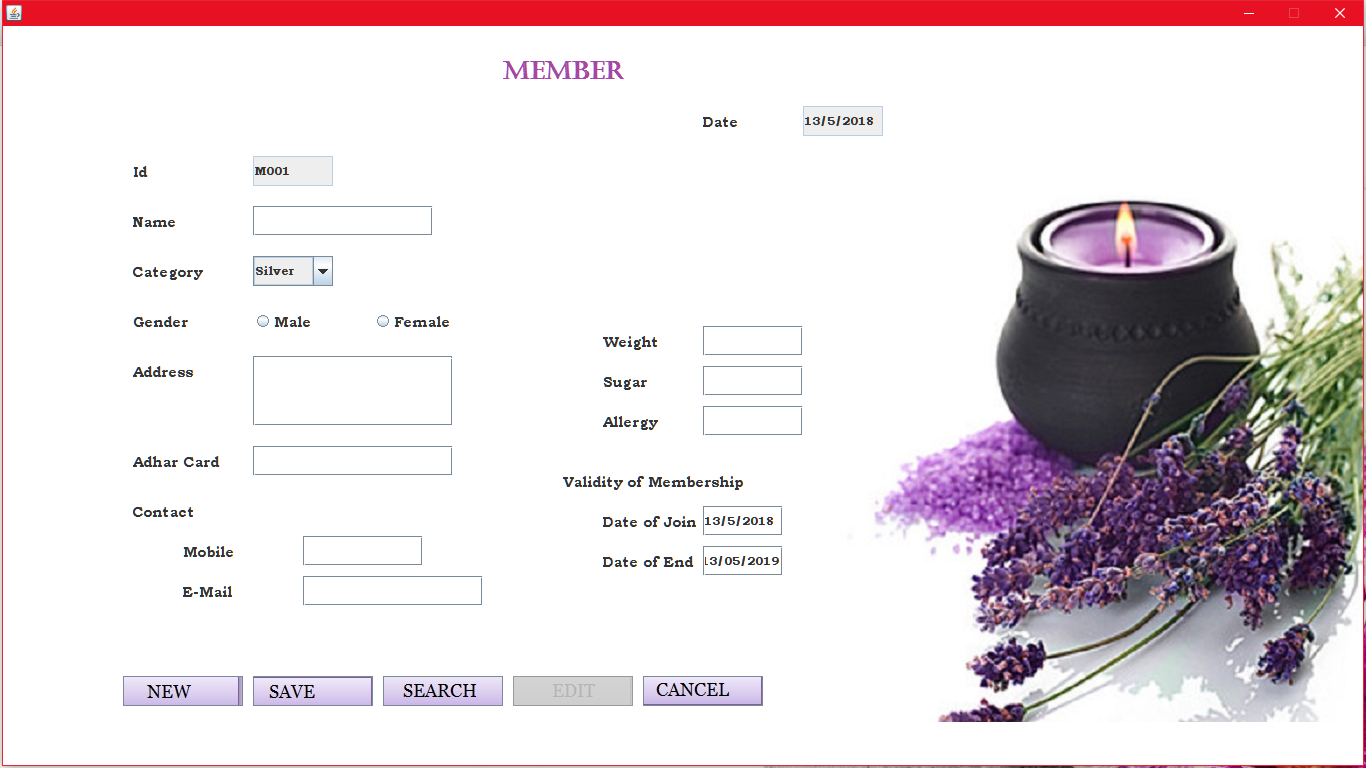
{

customer obj=new customer();

}

}

* **Member:**Member is a person who approaches to a Spa. He/she is the one who first enquires about different membership packages and finally chooses a package which is good for his needs. He/she will be provided with certain offers, discounts and priorities according to the membership package he chooses.



import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.io.\*;

import java.sql.\*;

import java.util.\*;

public class member extends JFrame implements ActionListener,FocusListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/mas.jpg"));

JLabel lhead=new JLabel("Member");

JLabel lmember\_id=new JLabel("Id");

JLabel ldate=new JLabel("Date");

JLabel lmember\_name=new JLabel("Name");

JLabel lmember\_catagory=new JLabel("Category");

JLabel lmember\_add=new JLabel("Address");

JLabel lmember\_gender=new JLabel("Gender");

JLabel lmember\_contact=new JLabel("Contact");

JLabel lmember\_mob=new JLabel("Mobile");

JLabel lmember\_email=new JLabel("E-Mail");

JLabel lmember\_val=new JLabel("Validity of Membership");

JLabel lmember\_join=new JLabel("Date of Join");

JLabel lmember\_end=new JLabel("Date of End");

JLabel lmember\_adhar=new JLabel("Adhar Card");

JLabel lmember\_weight=new JLabel("Weight");

JLabel lmember\_sugar=new JLabel("Sugar");

JLabel lmember\_allergy=new JLabel("Allergy");

JButton member\_new=new JButton("NEW",new ImageIcon("../Images/mnbt.jpg"));

JButton member\_save=new JButton("SAVE",new ImageIcon("../Images/msbt.jpg"));

JButton member\_search=new JButton("SEARCH",new ImageIcon("../Images/mssbt.jpg"));

JButton member\_cancel=new JButton("CANCEL",new ImageIcon("../Images/mcbt.jpg"));

JButton member\_edit=new JButton("EDIT",new ImageIcon("../Images/mebt.jpg"));

JComboBox cate=new JComboBox();

JTextField txmember\_id=new JTextField();

JTextField txdate=new JTextField();

JTextField txmember\_name=new JTextField();

JTextArea txmember\_add=new JTextArea();

JTextField txmember\_mob=new JTextField();

JTextField txmember\_email=new JTextField();

JTextField txmember\_join=new JTextField();

JTextField txmember\_end=new JTextField();

JTextField txmember\_adhar=new JTextField();

JTextField txmember\_weight=new JTextField();

JTextField txmember\_sugar=new JTextField();

JTextField txmember\_allergy=new JTextField();

JRadioButton rmale =new JRadioButton("Male");

JRadioButton rfemale=new JRadioButton("Female");

String strgen="";

Calendar cal=Calendar.getInstance();

public member()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

add(lmember\_id);

add(ldate);

add(lmember\_name);

add(lmember\_catagory);

add(lmember\_gender);

add(lmember\_add);

add(lmember\_contact);

add(lmember\_mob);

add(lmember\_email);

add(lmember\_val);

add(lmember\_join);

add(lmember\_end);

add(lmember\_adhar);

add(lmember\_weight);

add(lmember\_sugar);

add(lmember\_allergy);

add(member\_new);

add(member\_save);

add(member\_search);

add(member\_edit);

add(member\_cancel);

add(cate);

bg.add(rmale);

bg.add(rfemale);

add(rmale);

add(rfemale);

add(txmember\_id);

add(txdate);

add(txmember\_name);

add(sradd);

add(txmember\_mob);

add(txmember\_email);

add(txmember\_join);

add(txmember\_end);

add(txmember\_adhar);

add(txmember\_weight);

add(txmember\_sugar);

add(txmember\_allergy);

lmember\_id.setBounds(130,130,100,30);

txmember\_id.setBounds(250,130,80,30);

lmember\_name.setBounds(130,180,100,30);

txmember\_name.setBounds(250,180,180,30);

lmember\_catagory.setBounds(130,230,100,30);

cate.setBounds(250,230,80,30);

cate.addItem("Silver");

cate.addItem("Gold");

cate.addItem("Platinum");

lmember\_gender.setBounds(130,280,100,30);

rmale.setBounds(250,280,100,30);

rfemale.setBounds(370,280,100,30);

lmember\_add.setBounds(130,330,100,30);

sradd.setBounds(250,330,200,70);

lmember\_adhar.setBounds(130,420,100,30);

txmember\_adhar.setBounds(250,420,200,30);

lmember\_contact.setBounds(130,470,100,30);

lmember\_mob.setBounds(180,510,100,30);

txmember\_mob.setBounds(300,510,120,30);

lmember\_email.setBounds(180,550,100,30);

txmember\_email.setBounds(300,550,180,30);

lmember\_weight.setBounds(600,300,100,30);

txmember\_weight.setBounds(700,300,100,30);

lmember\_sugar.setBounds(600,340,100,30);

txmember\_sugar.setBounds(700,340,100,30);

lmember\_allergy.setBounds(600,380,100,30);

txmember\_allergy.setBounds(700,380,100,30);

lmember\_val.setBounds(560,440,200,30);

lmember\_join.setBounds(600,480,100,30);

txmember\_join.setBounds(700,480,80,30);

lmember\_end.setBounds(600,520,100,30);

txmember\_end.setBounds(700,520,80,30);

ldate.setBounds(700,80,80,30);

txdate.setBounds(800,80,80,30);

member\_new.setBounds(120,650,120,30);

member\_save.setBounds(250,650,120,30);

member\_search.setBounds(380,650,120,30);

member\_edit.setBounds(510,650,120,30);

member\_cancel.setBounds(640,650,120,30);

txdate.setText(""+cal.get(Calendar.DATE)+"/"+(cal.get(Calendar.MONTH)+1)+"/"+cal.get(Calendar.YEAR));

member\_new.addActionListener(this);

member\_save.addActionListener(this);

member\_search.addActionListener(this);

member\_edit.addActionListener(this);

member\_cancel.addActionListener(this);

rmale.addActionListener(this);

rfemale.addActionListener(this);

txmember\_id.setEditable(false);

txdate.setEditable(false);

member\_save.setEnabled(false);

member\_edit.setEnabled(false);

txmember\_name.addFocusListener(this);

txmember\_mob.addFocusListener(this);

txmember\_email.addFocusListener(this);

txmember\_adhar.addFocusListener(this);

txmember\_join.addFocusListener(this);

txmember\_end.addFocusListener(this);

}

public void focusLost(FocusEvent fe)

{

int i,tflg=0;

if(fe.getComponent()==txmember\_name)

{

strchk=txmember\_name.getText();

if(txmember\_name.getText().length()==0)

else

tflg=1;

if(tflg==1)

{

for(i=0;i<strchk.length();i++)

{

if(strchk.charAt(i)>='a'&&strchk.charAt(i)<='z'||strchk.charAt(i)>='A'&&strchk.charAt(i)<='Z'||strchk.charAt(i)==' ')

{

}

else

{

JOptionPane.showMessageDialog(null,"Invalid Name!!!");

break;

}

}

}

}

if(fe.getComponent()==txmember\_mob)

{

int tmflg=0;

strchk=txmember\_mob.getText();

if(txmember\_mob.getText().length()==0)

else

tmflg=1;

if(tmflg==1)

{

a:{for(i=0;i<strchk.length();i++)

{

if(Character.getNumericValue(strchk.charAt(i))>=0&&Character.getNumericValue(strchk.charAt(i))<=9)

{

}

else

{

JOptionPane.showMessageDialog(null,"Enter Only Digits!!");

txmember\_mob.setText("");

break a;

}

if(strchk.length()!=10)

{

JOptionPane.showMessageDialog(null,"Enter 10-digits Mobile Number!!");

txmember\_mob.setText("");

break a;

}}

}

}

}

if(fe.getComponent()==txmember\_adhar)

{

int tmflg=0;

strchk=txmember\_adhar.getText();

if(txmember\_adhar.getText().length()==0)

else

tmflg=1;

if(tmflg==1)

{

p:{for(i=0;i<strchk.length();i++)

{

if(Character.getNumericValue(strchk.charAt(i))>=0&&Character.getNumericValue(strchk.charAt(i))<=9)

{

}

else

{

JOptionPane.showMessageDialog(null,"Enter Only Digits!!");

txmember\_adhar.setText("");

break p;

}

if(strchk.length()!=12)

{

JOptionPane.showMessageDialog(null,"Enter 12-digits Aadhar Number!!");

txmember\_adhar.setText("");

break p;

}}

}

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==member\_new)

{

try

{

int ctr=1;

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(2));

}

if(ctr<10)

txmember\_id.setText("M00"+ctr);

if(ctr>=10&&ctr<100)

txmember\_id.setText("M0"+ctr);

txmember\_name.setText("");

bg.clearSelection();

cate.setSelectedIndex(0);

txmember\_add.setText("");

txmember\_mob.setText("");

txmember\_email.setText("");

txmember\_adhar.setText("");

txmember\_weight.setText("");

txmember\_sugar.setText("");

txmember\_allergy.setText("");

txmember\_join.setText(""+txdate.getText());

SimpleDateFormat formatter=new SimpleDateFormat("dd/MM/yyyy");

Calendar c=Calendar.getInstance();

c.setTime(formatter.parse(txmember\_join.getText()));

c.add(Calendar.YEAR,1);

txmember\_end.setText(""+formatter.format(c.getTime()));

member\_save.setEnabled(true);

member\_edit.setEnabled(false);

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==member\_save)

{

int flg=0;

String tdis="";

member\_edit.setEnabled(false);

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="insert into membertbl(mem\_date,mem\_id,mem\_name,mem\_category,mem\_gender,mem\_address,mem\_adhar,mem\_mob,mem\_email,mem\_joindate,mem\_enddate,mem\_weight,mem\_sugar,mem\_allergy,mem\_dis) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?);";

PreparedStatement ps=con.prepareStatement(str);

if(txmember\_name.getText().length()==0||bg.getSelection()==null||txmember\_add.getText().length()==0||txmember\_adhar.getText().length()==0||txmember\_mob.getText().length()==0||txmember\_email.getText().length()==0||txmember\_join.getText().length()==0||txmember\_end.getText().length()==0||txmember\_weight.getText().length()==0||txmember\_sugar.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

if(flg==0)

{

ps.setString(1,txdate.getText());

ps.setString(2,txmember\_id.getText());

ps.setString(3,txmember\_name.getText());

ps.setString(4,cate.getSelectedItem().toString());

if(rmale.isSelected())

strgen="Male";

if(rfemale.isSelected())

strgen="Female";

ps.setString(5,strgen);

ps.setString(6,txmember\_add.getText());

ps.setString(7,txmember\_adhar.getText());

ps.setString(8,txmember\_mob.getText());

ps.setString(9,txmember\_email.getText());

ps.setString(10,txmember\_join.getText());

ps.setString(11,txmember\_end.getText());

ps.setString(12,txmember\_weight.getText());

ps.setString(13,txmember\_sugar.getText());

if(txmember\_allergy.getText().length()==0)

ps.setString(14,"No");

else

ps.setString(14,txmember\_allergy.getText());

if(cate.getSelectedItem().equals("Silver"))

{

tdis=""+5;

}

if(cate.getSelectedItem().equals("Gold"))

{

tdis=""+10;

}

if(cate.getSelectedItem().equals("Platinum"))

{

tdis=""+15;

}

ps.setString(15,tdis);

ps.executeUpdate();

Statement st=con.createStatement();

st.executeUpdate("update keytbl set mem\_id=mem\_id+1");

JOptionPane.showMessageDialog(null,"Record Saved!!");

member\_save.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==member\_search)

{

sflg=1;

int flg=0;

String t=JOptionPane.showInputDialog("Enter Member id to search");

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

Statement st=con.createStatement();

res=st.executeQuery("select \* from membertbl where mem\_id='"+t+"'");

while(res.next())

{

member\_edit.setEnabled(true);

flg=1;

txmember\_id.setText(res.getString(2));

txmember\_name.setText(res.getString(3));

txdate.setText(res.getString(1));

txmember\_add.setText(res.getString(6));

txmember\_adhar.setText(res.getString(7));

txmember\_mob.setText(res.getString(8));

txmember\_email.setText(res.getString(9));

txmember\_join.setText(res.getString(10));

txmember\_end.setText(res.getString(11));

txmember\_weight.setText(res.getString(12));

txmember\_sugar.setText(res.getString(13));

txmember\_allergy.setText(res.getString(14));

cate.setSelectedItem(res.getString(4));

if(res.getString(5).equals("Male"))

{

rmale.setSelected(true);

rfemale.setSelected(false);

}

else

{

rmale.setSelected(false);

rfemale.setSelected(true);

}

}

member\_save.setEnabled(false);

if(flg==0&&t.length()!=0)

JOptionPane.showMessageDialog(null,"No Record Found!!!");

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==member\_edit)

{

member\_edit.setEnabled(false);

String tdis="";

String tmp=txmember\_allergy.getText();

member\_save.setEnabled(false);

int flg=0,tflg=0;

try

{

if(txmember\_id.getText().length()!=0&&sflg==1)

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

if(rmale.isSelected())

strgen="Male";

if(rfemale.isSelected())

strgen="Female";

if(txmember\_name.getText().length()==0||bg.getSelection()==null||txmember\_add.getText().length()==0||txmember\_adhar.getText().length()==0||txmember\_mob.getText().length()==0||txmember\_email.getText().length()==0||txmember\_join.getText().length()==0||txmember\_end.getText().length()==0||txmember\_weight.getText().length()==0||txmember\_sugar.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

if(flg==0)

{

if(cate.getSelectedItem().equals("Silver"))

{

tdis=""+5;

}

if(cate.getSelectedItem().equals("Gold"))

{

tdis=""+10;

}

if(cate.getSelectedItem().equals("Platinum"))

{

tdis=""+15;

}

String str="update membertbl set mem\_date='"+txdate.getText()+"',mem\_name='"+txmember\_name.getText()+"',mem\_category='"+cate.getSelectedItem()+"',mem\_gender='"+strgen+"',mem\_address='"+txmember\_add.getText()+"',mem\_adhar='"+txmember\_adhar.getText()+"',mem\_mob='"+txmember\_mob.getText()+"',mem\_email='"+txmember\_email.getText()+"',mem\_joindate='"+txmember\_join.getText()+"',mem\_enddate='"+txmember\_end.getText()+"',mem\_weight='"+txmember\_weight.getText()+"',mem\_sugar='"+txmember\_sugar.getText()+"',mem\_allergy='"+tmp+"',mem\_dis='"+tdis+"' where mem\_id='"+txmember\_id.getText()+"'";

Statement st=con.createStatement();

st.executeUpdate(str);

JOptionPane.showMessageDialog(null,"Updated");

}

}

else

{

JOptionPane.showMessageDialog(null,"Search Record for update!!");

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==member\_cancel)

{

this.dispose();

}

}

public static void main(String argv[])

throws IOException

{

member obj=new member();

}

}

* **Staff:**Employee is a person who treats customer/member according to their packages selected. He/she is qualified in a certain treatment. He/she keeps relevant details of treatment and diet to give to the customer/member.



import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.\*;

public class staff extends JFrame implements ActionListener,FocusListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/mas.jpg"));

int sflg=0;

String strgen="";

JLabel head=new JLabel("STAFF INFORMATION");

JLabel sdate=new JLabel("Date");

JLabel sid=new JLabel("Staff ID");

JLabel sname=new JLabel("Name");

JLabel sgen=new JLabel("Gender");

JLabel stype=new JLabel("Speciality In");

JLabel sadd=new JLabel("Address");

JLabel scont=new JLabel("Contact");

JLabel smob=new JLabel("Mobile");

JLabel smail=new JLabel("E-mail");

JLabel sstat=new JLabel("Status");

JLabel ssal=new JLabel("Salary");

JLabel sjoin=new JLabel("Date of Join");

JLabel sprework=new JLabel("Previous Work");

JTextField tdate=new JTextField();

JTextField tsid=new JTextField();

JTextField tsname=new JTextField();

JTextArea tsadd=new JTextArea();

JTextField tsmob=new JTextField();

JTextField tsmail=new JTextField();

JTextField tssal=new JTextField();

JTextField tsjoin=new JTextField();

JTextArea tsprework=new JTextArea();

JButton snew=new JButton("New",new ImageIcon("../Images/mnbt.jpg"));

JButton ssave=new JButton("Save",new ImageIcon("../Images/msbt.jpg"));

JButton ssearch=new JButton("Search",new ImageIcon("../Images/mssbt.jpg"));

JButton sedit=new JButton("Edit",new ImageIcon("../Images/mebt.jpg"));

JButton scancel=new JButton("Cancel",new ImageIcon("../Images/mcbt.jpg"));

JRadioButton rmale=new JRadioButton("Male");

JRadioButton rfemale=new JRadioButton("Female");

JComboBox ctype=new JComboBox();

Calendar cal=Calendar.getInstance();

public staff()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

lbl.setBounds(0,0,getWidth(),getHeight());

head.setBounds(500,30,450,30);

sdate.setBounds(750,80,100,30);

tdate.setBounds(870,80,100,30);

sid.setBounds(130,130,100,30);

tsid.setBounds(250,130,100,30);

sname.setBounds(130,180,100,30);

tsname.setBounds(250,180,140,30);

sgen.setBounds(130,230,100,30);

rmale.setBounds(250,230,80,30);

rfemale.setBounds(350,230,100,30);

sadd.setBounds(130,280,100,30);

sradd.setBounds(250,280,170,70);

scont.setBounds(130,350,100,30);

smob.setBounds(180,390,100,30);

tsmob.setBounds(300,390,100,30);

smail.setBounds(180,430,100,30);

tsmail.setBounds(300,430,150,30);

sjoin.setBounds(570,200,100,30);

tsjoin.setBounds(680,200,100,30);

stype.setBounds(570,250,100,30);

ctype.setBounds(680,250,120,30);

sprework.setBounds(570,300,130,30);

srpwork.setBounds(690,300,160,60);

ssal.setBounds(570,370,100,30);

tssal.setBounds(680,370,100,30);

snew.setBounds(120,600,120,30);

ssave.setBounds(250,600,120,30);

ssearch.setBounds(380,600,120,30);

sedit.setBounds(510,600,120,30);

scancel.setBounds(640,600,120,30);

add(ssal);

add(tssal);

add(scancel);

add(ssearch);

add(sedit);

add(snew);

add(ssave);

add(tsjoin);

add(sjoin);

add(stype);

add(ctype);

add(scont);

add(tsmob);

add(smob);

add(smail);

add(tsmail);

add(sadd);

add(sradd);

add(sgen);

add(rmale);

add(rfemale);

bg.add(rfemale);

bg.add(rmale);

add(sname);

add(tsname);

add(tsid);

add(sid);

add(tdate);

add(sdate);

add(head);

add(sprework);

add(srpwork);

ctype.addItem("Select Type");

snew.addActionListener(this);

ssave.addActionListener(this);

sedit.addActionListener(this);

scancel.addActionListener(this);

ssearch.addActionListener(this);

tdate.setText(""+cal.get(Calendar.DATE)+"/"+(cal.get(Calendar.MONTH)+1)+"/"+cal.get(Calendar.YEAR));

tsid.setEditable(false);

tdate.setEditable(false);

ssave.setEnabled(false);

sedit.setEnabled(false);

tsname.addFocusListener(this);

tsmob.addFocusListener(this);

tsmail.addFocusListener(this);

tsjoin.addFocusListener(this);

tssal.addFocusListener(this);

}

public void focusLost(FocusEvent fe)

{

int i,tflg=0;

if(fe.getComponent()==tsname)

{

strchk=tsname.getText();

if(tsname.getText().length()==0)

else

tflg=1;

if(tflg==1)

{

for(i=0;i<strchk.length();i++)

{

if(strchk.charAt(i)>='a'&&strchk.charAt(i)<='z'||strchk.charAt(i)>='A'&&strchk.charAt(i)<='Z'||strchk.charAt(i)==' ')

{

}

else

{

JOptionPane.showMessageDialog(null,"Invalid Name!!!");

}

}

}

}

if(fe.getComponent()==tsmob)

{

int tmflg=0;

strchk=tsmob.getText();

if(tsmob.getText().length()==0)

else

tmflg=1;

if(tmflg==1)

{

a:{for(i=0;i<strchk.length();i++)

{

if(Character.getNumericValue(strchk.charAt(i))>=0&&Character.getNumericValue(strchk.charAt(i))<=9)

{

}

else

{

JOptionPane.showMessageDialog(null,"Enter Only Digits!!");

break a;

}

if(strchk.length()!=10)

{

JOptionPane.showMessageDialog(null,"Enter 10-digits Mobile Number!!");

break a;

}}

}

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==snew)

{

sedit.setEnabled(false);

int ctr=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from Keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(10));

}

if(ctr<10)

tsid.setText("S00"+ctr);

if(ctr>=10&&ctr<100)

tsid.setText("S0"+ctr);

tsname.setText("");

tsadd.setText("");

tsmob.setText("");

tsmail.setText("");

tsjoin.setText("");

tssal.setText("");

bg.clearSelection();

tsprework.setText("");

tsjoin.setText(tdate.getText());

ssave.setEnabled(true);

res=st.executeQuery("select \* from typetbl");

ctype.addItem("Select Type");

while(res.next())

{

ctype.addItem(res.getString(2));

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==ssave)

{

sedit.setEnabled(false);

int flg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="insert into stafftbl(staffid,sname,sgender,address,mobile,email,joindate,speciality,previouswork,salary)values(?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

if(tsname.getText().length()==0||bg.getSelection()==null||tsadd.getText().length()==0||tsmob.getText().length()==0||tsmail.getText().length()==0||tsjoin.getText().length()==0||tsprework.getText().length()==0||tssal.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

if(flg==0)

{

ps.setString(1,tsid.getText());

ps.setString(2,tsname.getText());

if(rmale.isSelected())

strgen="Male";

else

strgen="Female";

ps.setString(3,strgen);

ps.setString(4,tsadd.getText());

ps.setString(5,tsmob.getText());

ps.setString(6,tsmail.getText());

ps.setString(7,tsjoin.getText());

String tid="";

Statement st1=con.createStatement();

ResultSet res=st1.executeQuery("select \* from typetbl where typename='"+ctype.getSelectedItem().toString()+"'");

while(res.next())

{

tid=res.getString(1);

}

ps.setString(8,tid);

ps.setString(9,tsprework.getText());

ps.setString(10,tssal.getText());

ps.executeUpdate();

Statement st=con.createStatement();

String str1="update keytbl set staffid=staffid+1";

st.executeUpdate(str1);

JOptionPane.showMessageDialog(null,"Saved...");

ssave.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==ssearch)

{

sflg=1;

ssave.setEnabled(false);

String tpid="";

String tname="";

String tmp="";

String t=JOptionPane.showInputDialog("Enter Id");

try

{

int flg=0;

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from stafftbl where staffid='"+t+"'");

while(res.next())

{

sedit.setEnabled(true);

flg=1;

tsid.setText(res.getString(1));

tsname.setText(res.getString(2));

if(res.getString(3).equals("Male"))

rmale.setSelected(true);

else

rfemale.setSelected(true);

tsadd.setText(res.getString(4));

tsmob.setText(res.getString(5));

tsmail.setText(res.getString(6));

tsjoin.setText(res.getString(7));

tpid=res.getString(8);

Statement st1=con.createStatement();

ResultSet res1=st1.executeQuery("select \* from typetbl");

ctype.addItem("Select Type"); while(res1.next())

{

tname=res1.getString(2);

if(tpid.equals(res1.getString(1)))

tmp=tname;

ctype.addItem(tname);

}

ctype.setSelectedItem(tmp);

tsprework.setText(res.getString(9));

tssal.setText(res.getString(10));

}

if(flg==0&&t.length()!=0)

{

JOptionPane.showMessageDialog(null,"No Record Found!!");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==scancel)

{

this.dispose();

}

if(e.getSource()==sedit)

{

ssave.setEnabled(false);

sedit.setEnabled(false);

int flg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

if(tsid.getText().length()!=0&&sflg==1)

{

String tpid="";

String tname="";

String tmp="";

if(rmale.isSelected())

strgen="Male";

if(rfemale.isSelected())

strgen="Female";

Statement st1=con.createStatement();

ResultSet res1=st1.executeQuery("select \* from typetbl where typename='"+ctype.getSelectedItem().toString()+"'");

while(res1.next())

{

tmp=res1.getString(1);

}

if(tmp.length()==0)

tmp="N/A";

if(tsname.getText().length()==0||bg.getSelection()==null||tsadd.getText().length()==0||tsmob.getText().length()==0||tsmail.getText().length()==0||tsjoin.getText().length()==0||tsprework.getText().length()==0||tssal.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

if(flg==0)

{

String str="update stafftbl set sname='"+tsname.getText()+"',sgender='"+strgen+"',address='"+tsadd.getText()+"',mobile='"+tsmob.getText()+"',email='"+tsmail.getText()+"',joindate='"+tsjoin.getText()+"',speciality='"+tmp+"',previouswork='"+tsprework.getText()+"',salary='"+tssal.getText()+"' where staffid='"+tsid.getText()+"'";

st.executeUpdate(str);

JOptionPane.showMessageDialog(null,"updated...");

}

}

else

{

JOptionPane.showMessageDialog(null,"Search Record for update!!");

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

}

public static void main(String argv[])

throws IOException

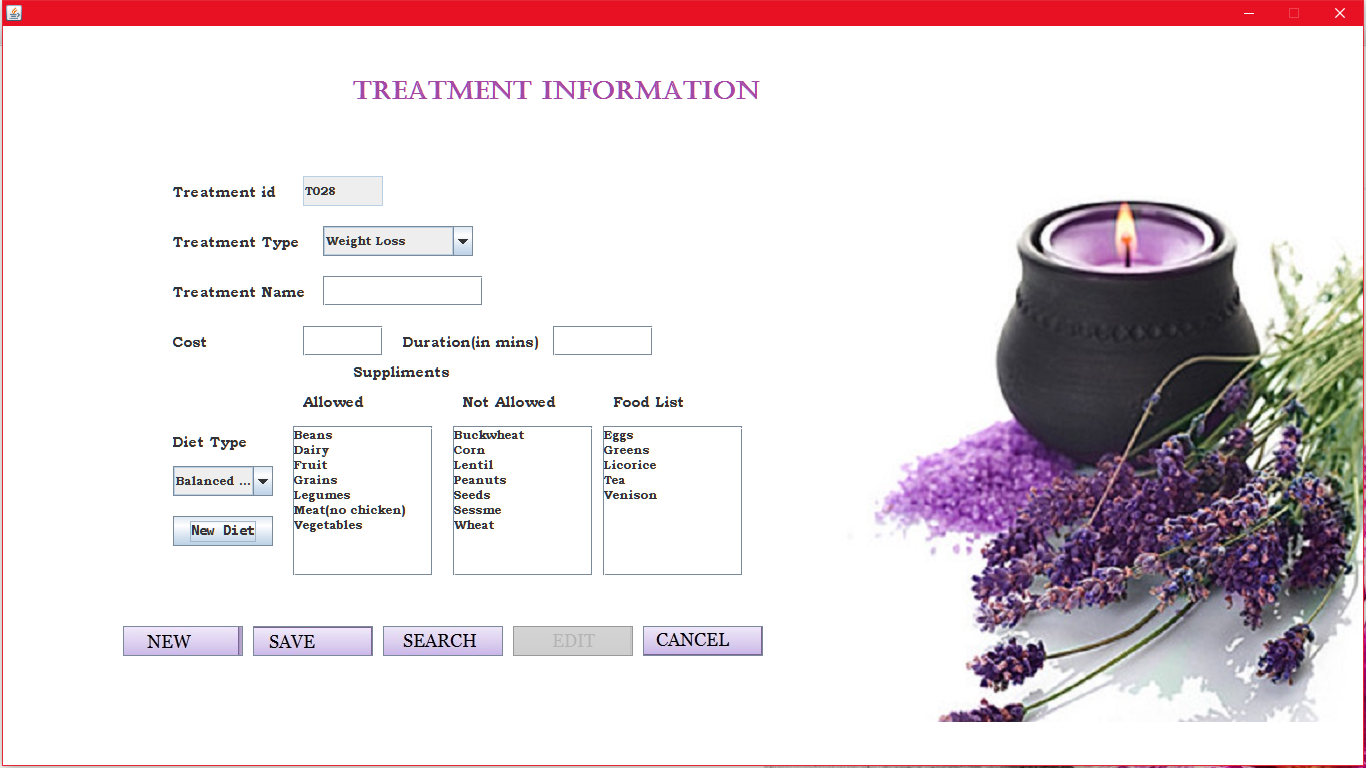
{

staff obj=new staff();

}

}

* **Treatment:**Here Product refers to all available treatments, diets and all other facilities given by the spa.**:**



import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

public class treatment extends JFrame implements ActionListener,FocusListener,ItemListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/mas.jpg"));

String strgen,strpen,dt;

JLabel lid=new JLabel("Treatment id");

JLabel ltype=new JLabel("Treatment Type");

JLabel ltname=new JLabel("Treatment Name");

JLabel lcost=new JLabel("Cost");

JLabel ldur=new JLabel("Duration(in mins)");

JLabel lh=new JLabel("Treatment Information");

JLabel lsup=new JLabel("Suppliments");

JLabel ldiettype=new JLabel("Diet Type");

JLabel lnalw=new JLabel("Not Allowed");

JLabel lalw=new JLabel("Allowed");

JLabel llist=new JLabel("Food List");

JButton tnew=new JButton("New",new ImageIcon("../Images/mnbt.jpg"));

JButton tsave=new JButton("Save",new ImageIcon("../Images/msbt.jpg"));

JButton tsearch=new JButton("Search",new ImageIcon("../Images/mssbt.jpg"));

JButton tedit=new JButton("Edit",new ImageIcon("../Images/mebt.jpg"));

JButton tcancel=new JButton("Cancel",new ImageIcon("../Images/mcbt.jpg"));

JButton tadd=new JButton("New Diet");

JTextField tid=new JTextField();

JTextField ttname=new JTextField();

JTextField tcost=new JTextField();

JTextField tdur=new JTextField();

JTextArea tlist=new JTextArea();

JTextArea tnalw=new JTextArea();

JTextArea talw=new JTextArea();

JComboBox cdtype=new JComboBox();

JComboBox ctype=new JComboBox();

public treatment()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

lbl.setBounds(0,0,getWidth(),getHeight());

lid.setBounds(170,150,120,30);

tid.setBounds(300,150,80,30);

ltype.setBounds(170,200,140,30);

ctype.setBounds(320,200,150,30);

ltname.setBounds(170,250,140,30);

ttname.setBounds(320,250,160,30);

lcost.setBounds(170,300,100,30);

tcost.setBounds(300,300,80,30);

ldur.setBounds(400,300,140,30);

tdur.setBounds(550,300,100,30);

lsup.setBounds(350,330,100,30);

ldiettype.setBounds(170,400,100,30);

cdtype.setBounds(170,440,100,30);

tadd.setBounds(170,490,100,30);

lalw.setBounds(300,360,100,30);

sralw.setBounds(290,400,140,150);

lnalw.setBounds(460,360,100,30);

srnalw.setBounds(450,400,140,150);

llist.setBounds(610,360,100,30);

srlist.setBounds(600,400,140,150);

tnew.setBounds(120,600,120,30);

tsave.setBounds(250,600,120,30);

tsearch.setBounds(380,600,120,30);

tedit.setBounds(510,600,120,30);

tcancel.setBounds(640,600,120,30);

add(lsup);

add(ldiettype);

add(lalw);

add(sralw);

add(lnalw);

add(srnalw);

add(llist);

add(srlist);

add(ctype);

add(cdtype);

add(tcancel);

add(tnew);

add(tsave);

add(tsearch);

add(tedit);

add(tadd);

add(tdur);

add(ldur);

add(lh);

add(tcost);

add(lcost);

add(ltname);

add(ttname);

add(ltype);

add(lid);

add(tid);

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from diettbl");

while(res.next())

{

cdtype.addItem(res.getString(2));

}

ResultSet res1=st.executeQuery("select \* from typetbl");

ctype.addItem("Select Treatment Type");

while(res1.next())

{

ctype.addItem(res1.getString(2));

}

}

catch(Exception ee)

{

System.out.println(ee);

}

cdtype.addActionListener(this);

tnew.addActionListener(this);

tsave.addActionListener(this);

tsearch.addActionListener(this);

tcancel.addActionListener(this);

tedit.addActionListener(this);

ctype.addItemListener(this);

tadd.addActionListener(this);

tid.setEditable(false);

tsave.setEnabled(false);

tcost.addFocusListener(this);

tedit.setEnabled(false);

}

public void display()

{

if(ctype.getSelectedItem().equals("Weight Loss"))

{

lsup.setVisible(true);

ldiettype.setVisible(true);

cdtype.setVisible(true);

lalw.setVisible(true);

sralw.setVisible(true);

lnalw.setVisible(true);

srnalw.setVisible(true);

llist.setVisible(true);

srlist.setVisible(true);

tadd.setVisible(true);

}

else

{

lsup.setVisible(false);

ldiettype.setVisible(false);

cdtype.setVisible(false);

lalw.setVisible(false);

sralw.setVisible(false);

lnalw.setVisible(false);

srnalw.setVisible(false);

llist.setVisible(false);

srlist.setVisible(false);

tadd.setVisible(false);

}

}

public void itemStateChanged(ItemEvent ie)

{

if(ie.getSource()==ctype)

{

if(ie.getStateChange()==ItemEvent.SELECTED)

display();

}

}

public void focusLost(FocusEvent fe)

{

if(fe.getComponent()==tcost)

{

int i,tmflg=0,dflg=0;

strchk=tcost.getText();

if(tcost.getText().length()==0)

else

tmflg=1;

if(tmflg==1)

{

p:{for(i=0;i<strchk.length();i++)

{

if(strchk.charAt(i)>='a'&&strchk.charAt(i)<='z'||strchk.charAt(i)>='A'&&strchk.charAt(i)<='Z')

{

JOptionPane.showMessageDialog(null,"Enter Only Digits!!");

break p;

}

}

}

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==tsave)

{

tedit.setEnabled(false);

int flg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="insert into treatmenttbl(treat\_id,treat\_type,treat\_name,t\_cost,t\_duration,dietid)values(?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

Statement st=con.createStatement();

if(ctype.getSelectedIndex()==0||ttname.getText().length()==0||tcost.getText().length()==0||tdur.getText().length()==0||cdtype.getSelectedIndex()==0||talw.getText().length()==0||tnalw.getText().length()==0||tlist.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

if(flg==0)

{

ps.setString(1,tid.getText());

ps.setString(2,ctype.getSelectedItem().toString());

ps.setString(3,ttname.getText());

ps.setString(4,tcost.getText());

ps.setString(5,tdur.getText());

if(ctype.getSelectedItem().equals("Weight Loss"))

{

ResultSet res=st.executeQuery("select \* from diettbl where diettype='"+cdtype.getSelectedItem()+"'");

while(res.next())

{

dt=res.getString(1);

}

ps.setString(6,dt);

}

else

{

dt="";

}

if(ctype.getSelectedIndex()!=0)

{

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Saved");

}

else

{

JOptionPane.showMessageDialog(null,"Select the Course Type!!");

}

String str1="update keytbl set treat\_id=treat\_id+1";

st.executeUpdate(str1);

tsave.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==tnew)

{

tedit.setEnabled(false);

int ctr=1;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(3));

}

if(ctr<10)

tid.setText("T00"+ctr);

if(ctr>=10&&ctr<100)

tid.setText("T0"+ctr);

ctype.setSelectedIndex(0);

tcost.setText("");

tdur.setText("");

tdur.setText("");

cdtype.setSelectedIndex(0);

tsave.setEnabled(true);

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==tsearch)

{

int flg=0;

sflg=1;

tsave.setEnabled(false);

try

{

String dtyp="";

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

Statement st=con.createStatement();

String t=JOptionPane.showInputDialog("Enter Tid to search");

res=st.executeQuery("select \* from treatmenttbl where treat\_id='"+t+"'");

while(res.next())

{

tedit.setEnabled(true);

flg=1;

ctype.setSelectedItem(res.getString(2));

ttname.setText(res.getString(3));

tcost.setText(res.getString(4));

tid.setText(res.getString(1));

tdur.setText(res.getString(5));

dtyp=res.getString(6);

}

ResultSet res2=st.executeQuery("select \* from diettbl where dietid='"+dtyp+"'");

while(res2.next())

{

cdtype.setSelectedItem(res2.getString(2));

display();

talw.setText(res2.getString(3));

tnalw.setText(res2.getString(4));

tlist.setText(res2.getString(5));

}

if(flg==0&&t.length()!=0)

{

JOptionPane.showMessageDialog(null,"No Record Found!!");

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==tedit)

{

String tmp="";

int flg=0,tflg=0;

tsave.setEnabled(false);

tedit.setEnabled(false);

try

{

if(tid.getText().length()!=0&&sflg==1)

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

if(ctype.getSelectedIndex()==0||ttname.getText().length()==0||tcost.getText().length()==0||tdur.getText().length()==0||cdtype.getSelectedIndex()==0||talw.getText().length()==0||tnalw.getText().length()==0||tlist.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

tflg=1;

}

if(flg==0)

{

if(cdtype.getSelectedItem().equals("Weight Loss"))

{

ResultSet res=st.executeQuery("select \* from diettbl where diettype='"+cdtype.getSelectedItem().toString()+"'");

while(res.next())

{

tmp=res.getString(1);

}

}

else

{

tmp="N/A";

}

String str="update treatmenttbl set treat\_type='"+ctype.getSelectedItem()+"',treat\_name='"+ttname.getText()+"',t\_cost='"+tcost.getText()+"',t\_duration='"+tdur.getText()+"',dietid='"+tmp+"' where treat\_id='"+tid.getText()+"'";

st.executeUpdate(str);

JOptionPane.showMessageDialog(null,"Edit Successfully");

}}}

catch(Exception ee)

{

if(tflg==0)

JOptionPane.showMessageDialog(null,""+ee);

}

}

if(e.getSource()==cdtype)

{

String t1="";

String t2="";

String t3="";

int i;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from diettbl where diettype='"+cdtype.getSelectedItem()+"'");

while(res.next())

{

t1=res.getString(3);

t2=res.getString(4);

t3=res.getString(5);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==tadd)

diet odiet=new diet();

if(e.getSource()==tcancel)

{

this.dispose();

}

}

public static void main(String argv[])

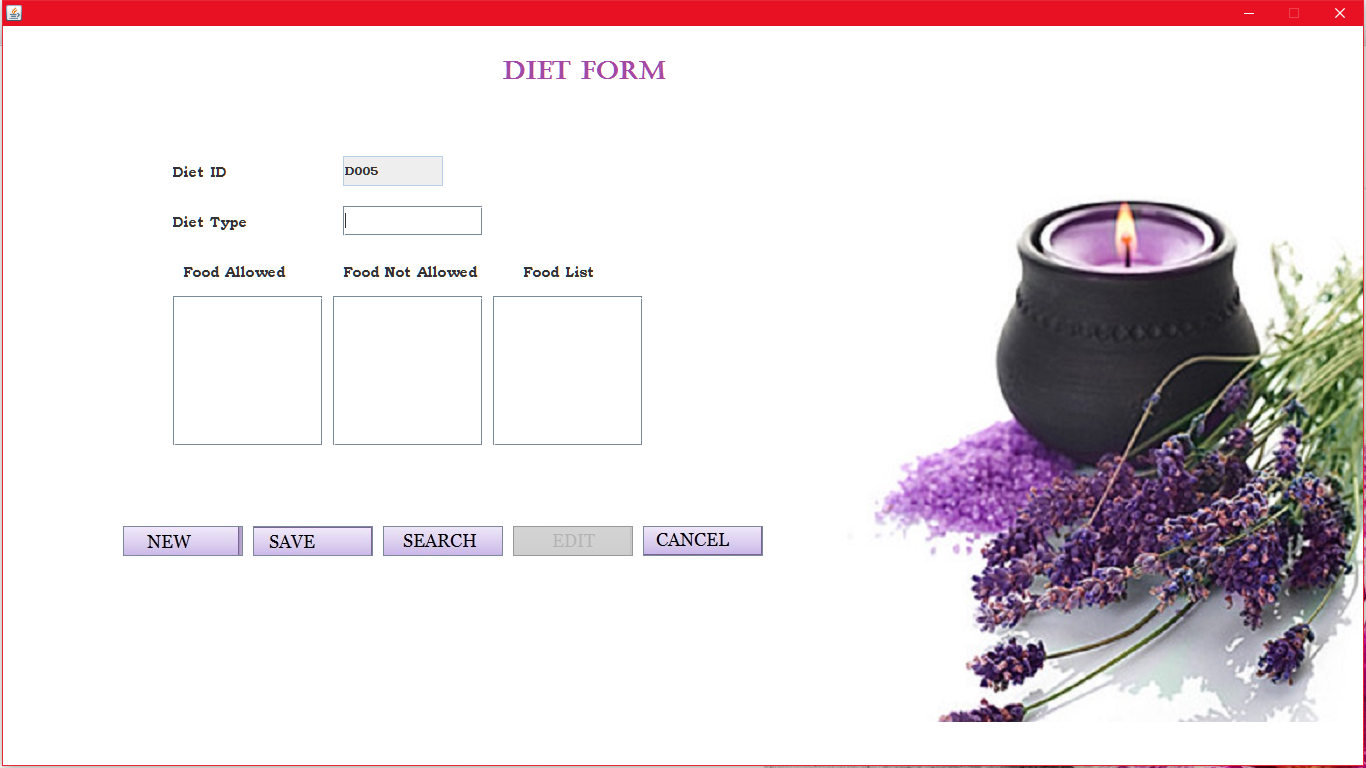
{

treatment obj=new treatment();

}

}

* **Diet:** Here the details of all diets with the weightloss are available.And authorized person can manage the diet records.



import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.io.\*;

import java.sql.\*;

public class diet extends JFrame implements ActionListener,FocusListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/mas.jpg"));

JLabel did=new JLabel("Diet ID");

JLabel dtype=new JLabel("Diet Type");

JLabel dalw=new JLabel("Food Allowed");

JLabel dnalw=new JLabel("Food Not Allowed");

JLabel dhead=new JLabel("Diet Form");

JLabel dlist=new JLabel("Food List");

JTextField dtid=new JTextField();

JTextField dttype=new JTextField();

JTextArea dtalw=new JTextArea();

JTextArea dtnalw=new JTextArea();

JTextArea dtlist=new JTextArea();

JButton dnew=new JButton("New",new ImageIcon("../Images/mnbt.jpg"));

JButton dsave=new JButton("Save",new ImageIcon("../Images/msbt.jpg"));

JButton dedit=new JButton("Edit",new ImageIcon("../Images/mebt.jpg"));

JButton dsearch=new JButton("Search",new ImageIcon("../Images/mssbt.jpg"));

JButton dcan=new JButton("Cancel",new ImageIcon("../Images/mcbt.jpg"));

public diet()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

dhead.setBounds(500,30,200,30);

did.setBounds(170,130,150,30);

dtid.setBounds(340,130,100,30);

dtype.setBounds(170,180,150,30);

dttype.setBounds(340,180,140,30);

dalw.setBounds(180,230,120,30);

sralw.setBounds(170,270,150,150);

dnalw.setBounds(340,230,140,30);

srnalw.setBounds(330,270,150,150);

dlist.setBounds(520,230,140,30);

srlist.setBounds(490,270,150,150);

dnew.setBounds(120,500,120,30);

dsave.setBounds(250,500,120,30);

dsearch.setBounds(380,500,120,30);

dedit.setBounds(510,500,120,30);

dcan.setBounds(640,500,120,30);

add(dcan);

add(dedit);

add(dsave);

add(dsearch);

add(dnew);

add(dnalw);

add(srnalw);

add(dalw);

add(sralw);

add(dtype);

add(dttype);

add(did);

add(dtid);

add(dhead);

add(dlist);

add(srlist);

add(lbl);

dnew.addActionListener(this);

dsave.addActionListener(this);

dsearch.addActionListener(this);

dedit.addActionListener(this);

dcan.addActionListener(this);

dtid.setEditable(false);

dsave.setEnabled(false);

dedit.setEnabled(false);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==dnew)

{

dsave.setEnabled(true);

dedit.setEnabled(false);

int ctr=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(5));

}

if(ctr<10)

dtid.setText("D00"+ctr);

if(ctr>9&&ctr<100)

dtid.setText("D0"+ctr);

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==dsave)

{

dedit.setEnabled(false);

int flg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

if(dtid.getText().length()==0||dttype.getText().length()==0||dtalw.getText().length()==0||dtnalw.getText().length()==0||dtlist.getText().length()==0)

{

flg=1;

JOptionPane.showMessageDialog(null,"Empty Fields");

}

if(flg==0)

{

String str1="update keytbl set dietid=dietid+1";

Statement st=con.createStatement();

st.executeUpdate(str1);

String str="insert into diettbl(dietid,diettype,foodallowed,foodnotallowed,foodlist)values(?,?,?,?,?)"; PreparedStatement ps=con.prepareStatement(str);

ps.setString(1,dtid.getText());

ps.setString(2,dttype.getText());

ps.setString(3,dtalw.getText());

ps.setString(4,dtnalw.getText());

ps.setString(5,dtlist.getText());

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Saved...");

dsave.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==dsearch)

{

dsave.setEnabled(false);

sflg=1;

String t1="";

String t2="";

String t3="";

int flg=0,i;

String t=JOptionPane.showInputDialog("Enter diet ID:");

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

Statement st=con.createStatement();

res=st.executeQuery("select \* from diettbl where dietid='"+t+"'");

while(res.next())

{

dedit.setEnabled(true);

flg=1;

dtid.setText(res.getString(1));

dttype.setText(res.getString(2));

t1=res.getString(3);

t2=res.getString(4);

t3=res.getString(5);

}

if(flg==0)

{

JOptionPane.showMessageDialog(null,"No Record Found!!");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==dedit)

{

dsave.setEnabled(false);

int flg=0;

try

{

if(dtid.getText().length()!=0&&sflg==1)

{

if(dtid.getText().length()==0||dttype.getText().length()==0||dtalw.getText().length()==0||dtnalw.getText().length()==0||dtlist.getText().length()==0)

{

flg=1;

JOptionPane.showMessageDialog(null,"Empty Fields");

}

if(flg==0)

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="update diettbl set diettype='"+dttype.getText()+"',foodallowed='"+dtalw.getText()+"',foodnotallowed='"+dtnalw.getText()+"',foodlist='"+dtlist.getText()+"' where dietid='"+dtid.getText()+"'";

Statement st=con.createStatement();

st.executeUpdate(str);

JOptionPane.showMessageDialog(null,"Updated");

}

}

else

{

JOptionPane.showMessageDialog(null,"Search Record For update!!");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==dcan)

{

this.dispose();

}

}

public static void main(String argv[])

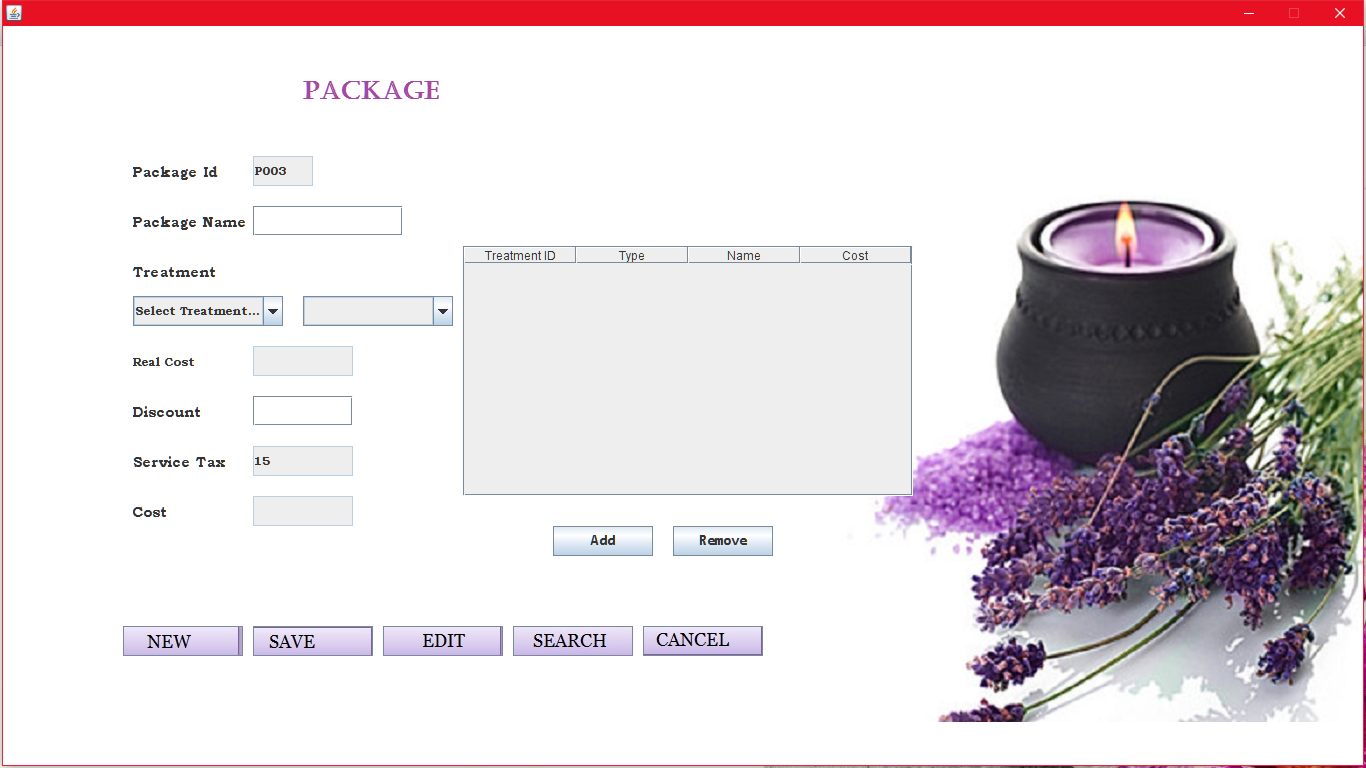
{

diet obj=new diet();

}

}

* **Package:** Here the creation of packages with total of 5 treatments are available.Packages can be created according to the Seasons and Festivals.



import java.awt.event.\*;

import java.applet.\*;

import java.awt.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.\*;

public class packagef extends JFrame implements ActionListener,FocusListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/mas.jpg"));

String tmp,strgen,strpen;

JLabel lpack\_id=new JLabel("Package Id");

JLabel lpack\_name=new JLabel("Package Name");

JLabel lctype=new JLabel("Treatment Type");

JLabel lpack\_rcost=new JLabel("Real Cost");

JLabel lpack\_dis=new JLabel("Discount");

JLabel lpack\_cost=new JLabel("Cost");

JLabel lpack\_head=new JLabel("PACKAGE");

JLabel lpack\_type1=new JLabel("Treatment");

JLabel lpack\_tax=new JLabel("Service Tax");

JTextField txpack\_id=new JTextField();

JTextField txpack\_name=new JTextField();

JTextField txpack\_rcost=new JTextField();

JTextField txpack\_cost=new JTextField();

JTextField txpack\_dis=new JTextField();

JTextField txpack\_tax=new JTextField();

JButton padd=new JButton("Add");

JButton premove=new JButton("Remove");

JButton psave=new JButton("Save",new ImageIcon("../Images/msbt.jpg"));

JButton psearch=new JButton("Search",new ImageIcon("../Images/mssbt.jpg"));

JButton pnew=new JButton("New",new ImageIcon("../Images/mnbt.jpg"));

JButton pedit=new JButton("Edit",new ImageIcon("../Images/mebt.jpg"));

JButton pcancel=new JButton("Cancel",new ImageIcon("../Images/mcbt.jpg"));

JComboBox ctype1=new JComboBox();

JComboBox ctreat1=new JComboBox();

double fcost=0.0;

String []col={"Treatment ID","Type","Name","Cost"};

String tid,ttyp,tnm,tcost;

DefaultTableModel model=new DefaultTableModel(col,0);

int pctr=1;

int ctr=1;

String trt="";

double cost=0.0;

int flg=0;

double c=0.0;

public packagef()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

lbl.setBounds(0,0,getWidth(),getHeight());

lpack\_head.setBounds(300,50,150,30);

lpack\_id.setBounds(130,130,100,30);

txpack\_id.setBounds(250,130,60,30);

lpack\_name.setBounds(130,180,120,30);

txpack\_name.setBounds(250,180,150,30);

lpack\_type1.setBounds(130,230,100,30);

ctype1.setBounds(130,270,150,30);

padd.setBounds(550,500,100,30);

premove.setBounds(670,500,100,30);

ctreat1.setBounds(300,270,150,30);

lpack\_rcost.setBounds(130,320,100,30);

txpack\_rcost.setBounds(250,320,100,30);

lpack\_dis.setBounds(130,370,100,30);

txpack\_dis.setBounds(250,370,100,30);

lpack\_tax.setBounds(130,420,100,30);

txpack\_tax.setBounds(250,420,100,30);

lpack\_cost.setBounds(130,470,100,30);

txpack\_cost.setBounds(250,470,100,30);

pnew.setBounds(120,600,120,30);

psave.setBounds(250,600,120,30);

pedit.setBounds(380,600,120,30);

psearch.setBounds(510,600,120,30);

pcancel.setBounds(640,600,120,30);

add(lpack\_head);

add(lpack\_id);

add(txpack\_id);

add(lpack\_tax);

add(lpack\_name);

add(txpack\_name);

add(txpack\_tax);

add(ctreat1);

add(premove);

add(padd);

JTable tbl=new JTable(data,col);

Panel pnl=new JScrollPane(tbl);

add(pnl);

add(lpack\_rcost);

add(txpack\_cost);

add(lpack\_cost);

add(txpack\_rcost);

add(lpack\_dis);

add(txpack\_dis);

add(psave);

add(pedit);

add(pcancel);

add(psearch);

add(pnew);

pnl.setBounds(460,220,450,250);

tbl.setBounds(470,220,450,100);

add(ctype1);

add(lpack\_type1);

padd.addActionListener(this);

premove.addActionListener(this);

pnew.addActionListener(this);

psave.addActionListener(this);

psearch.addActionListener(this);

pedit.addActionListener(this);

pcancel.addActionListener(this);

ctreat1.addActionListener(this);

ctype1.addActionListener(this);

txpack\_tax.setEditable(false);

txpack\_id.setEditable(false);

psave.setEnabled(false);

txpack\_rcost.setEditable(false);

txpack\_cost.setEditable(false);

txpack\_dis.addFocusListener(this);

}

public void treat(String ttmp,int i)

{

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="select \* from treatmenttbl where treat\_type='"+ttmp+"'";

Statement st=con.createStatement();

ResultSet res=st.executeQuery(str);

if(i==1)

{

ctreat1.removeAllItems();

ctreat1.addItem("");

}

while(res.next())

{

if(i==1)

ctreat1.addItem(res.getString(3).toString());

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==pnew)

{

try

{

c=0.0;

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(4));

}

if(ctr<10)

txpack\_id.setText("P00"+ctr);

if(ctr>=10&&ctr<100)

txpack\_id.setText("P0"+ctr);

txpack\_name.setText("");

txpack\_rcost.setText("");

txpack\_cost.setText("");

txpack\_dis.setText("");

txpack\_tax.setText("");

if(ctype1.getItemCount()==0)

{

ResultSet res1=st.executeQuery("select \* from typetbl");

ctype1.addItem("Select Treatment Type");

while(res1.next())

{

ctype1.addItem(res1.getString(2));

}

}

else

{

ctype1.setSelectedIndex(0);

}

psave.setEnabled(true);

txpack\_tax.setText("15");

}

catch(Exception ee)

{

JOptionPane.showMessageDialog(null,ee.getMessage());

}

}

if(e.getSource()==psave)

{

int flg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="insert into packagetbl(pack\_id,pack\_name,treat1\_id,treat2\_id,treat3\_id,treat4\_id,treat5\_id,realcost,discount,tcost,ttax) values(?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

Statement st=con.createStatement();

if(txpack\_name.getText().length()==0||tbl.getRowCount()==0||txpack\_dis.getText().length()==0||txpack\_tax.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

else

{

if(tbl.getRowCount()==1)

{

JOptionPane.showMessageDialog(null,"Select Treatments more than one!!");

flg=1;

}

}

if(flg==0)

{

ps.setString(1,txpack\_id.getText());

ps.setString(2,txpack\_name.getText());

ps.setString(3,tbl.getValueAt(0,0).toString());

ps.setString(4,""+tbl.getValueAt(1,0).toString());

ps.setString(8,txpack\_rcost.getText());

ps.setString(9,txpack\_dis.getText());

ps.setString(10,txpack\_cost.getText());

ps.setString(11,txpack\_tax.getText());

ps.executeUpdate();

st.executeUpdate("update keytbl set pack\_id=pack\_id+1");

JOptionPane.showMessageDialog(null,"Saved");

psave.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==ctype1)

{

tmp=ctype1.getSelectedItem().toString();

treat(tmp,1);

}

if(e.getSource()==padd)

{

int flg=0;

txpack\_dis.setText("0");

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

Statement st=con.createStatement();

String str="select \* from treatmenttbl where treat\_name='"+ctreat1.getSelectedItem()+"'";

res=st.executeQuery(str);

while(res.next())

{

Vector vec=new Vector();

trt=res.getString(1).toString();

for(int i=0;i<tbl.getRowCount();i++)

{

if(trt.equals(tbl.getValueAt(i,0)))

flg=1;

}

if(flg==1)

{

JOptionPane.showMessageDialog(null,"Already Entered!!!"); }

else

{

vec.add(trt);

vec.add(res.getString(2));

vec.add(res.getString(3));

String tm=res.getString(4);

c=c+Double.parseDouble(tm);

txpack\_rcost.setText(""+c);

txpack\_cost.setText(""+c);

vec.add(tm); model.addRow(vec);

}

}

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==psearch)

{

sflg=1;

psave.setEnabled(false);

String t=JOptionPane.showInputDialog("Enter Package id to search");

int flg=0,i;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

String str="select \* from packagetbl where pack\_id='"+t+"'";

Statement st=con.createStatement();

res=st.executeQuery(str);

String s[]=new String[5];

while(res.next())

{

flg=1;

txpack\_id.setText(res.getString(1));

txpack\_name.setText(res.getString(2));

txpack\_rcost.setText(res.getString(8));

c=Double.parseDouble(txpack\_rcost.getText());

txpack\_dis.setText(res.getString(9));

txpack\_cost.setText(res.getString(10));

txpack\_tax.setText(res.getString(11));

s[0]=res.getString(3);

s[1]=res.getString(4);

s[2]=res.getString(5);

s[3]=res.getString(6);

s[4]=res.getString(7);

}

if(flg==1)

{

int j=0;

while(j<5)

{

str="select \* from treatmenttbl where treat\_id='"+s[j]+"'";

Statement st1=con.createStatement();

ResultSet res1=st.executeQuery(str);

Vector v=new Vector();

while(res1.next())

{

v.add(res1.getString(1));

v.add(res1.getString(2));

v.add(res1.getString(3));

v.add(res1.getString(4));

model.addRow(v);

}

j++;

} }

if(flg==0)

{

JOptionPane.showMessageDialog(null,"No Record Found!!");

}

else

{

if(ctype1.getItemCount()==0)

{

ResultSet res2=st.executeQuery("select \* from typetbl");

ctype1.addItem("Select Treatment Type");

while(res2.next())

{

ctype1.addItem(res2.getString(2));

}

}

}

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==pedit)

{

psave.setEnabled(false);

int tflg=0,flg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

String strm[]=new String[5];

int n=tbl.getRowCount();

for(int i=0;i<5;i++)

{

if(i<n)

strm[i]=tbl.getValueAt(i,0).toString();

else

strm[i]="";

}

if(txpack\_id.getText().length()!=0&&sflg==1)

{

if(txpack\_name.getText().length()==0||tbl.getRowCount()==0||txpack\_dis.getText().length()==0||txpack\_tax.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

if(flg==0)

{

String str="update packagetbl set pack\_name='"+txpack\_name.getText()+"',treat1\_id='"+strm[0]+"',treat2\_id='"+strm[1]+"',treat3\_id='"+strm[2]+"',treat4\_id='"+strm[3]+"',treat5\_id='"+strm[4]+"',realcost='"+txpack\_rcost.getText()+"',discount='"+txpack\_dis.getText()+"',tcost='"+txpack\_cost.getText()+"' where pack\_id='"+txpack\_id.getText()+"'";

st.executeUpdate(str);

JOptionPane.showMessageDialog(null,"Updated");

}

}

else

{

JOptionPane.showMessageDialog(null,"Search Record for update!!");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==premove)

{

int i;

try

{

if(tbl.getRowCount()>0)

{

String t=JOptionPane.showInputDialog("Enter Row Number to Remove");

for(i=0;i<=tbl.getRowCount();i++)

{

if(i==Integer.parseInt(t))

{

String tm=""+tbl.getValueAt(Integer.parseInt(t)-1,3);

model.removeRow(Integer.parseInt(t)-1);

c=Double.parseDouble(txpack\_rcost.getText());

c=c-Double.parseDouble(tm);

txpack\_rcost.setText(""+c);

}

}

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==pcancel)

{

this.dispose();

}

}

public static void main(String argv[])

{

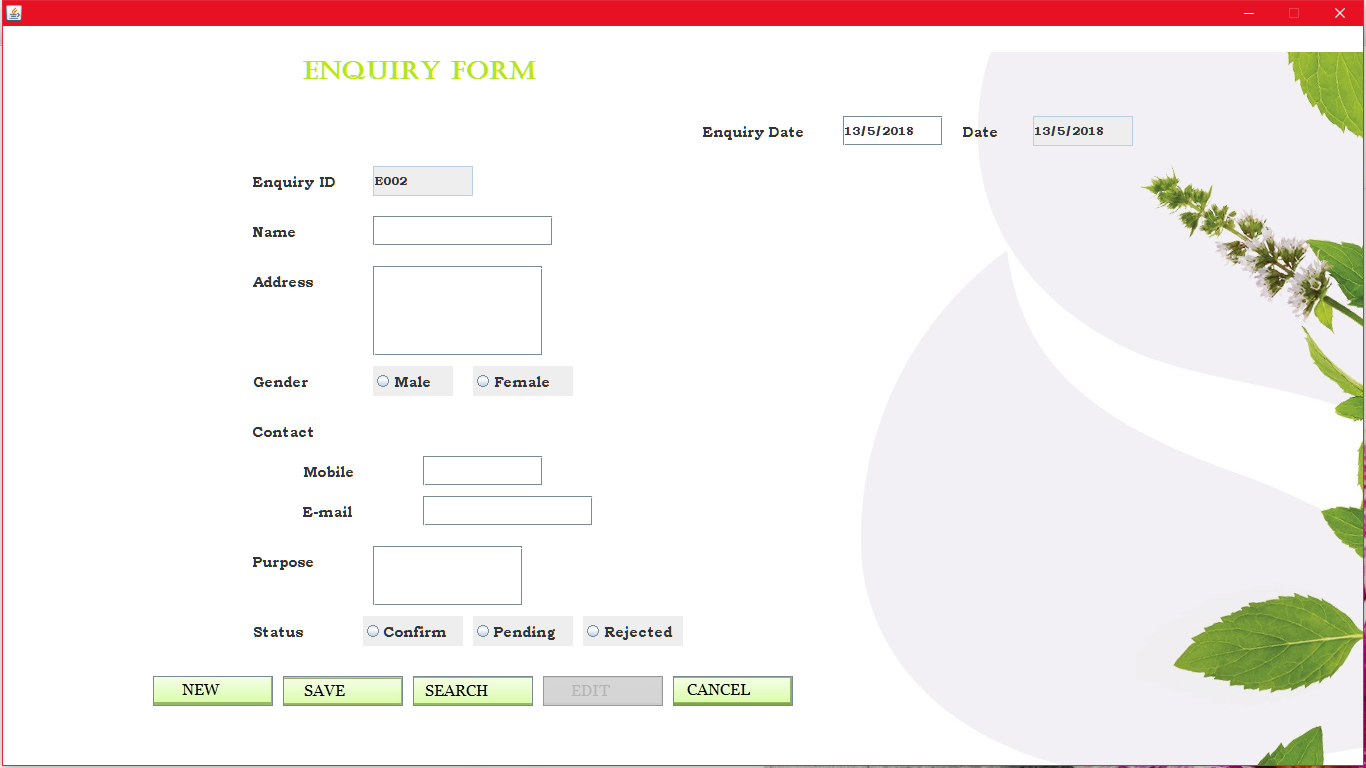
packagef obj=new packagef();

}

}

**Transaction Forms:-**

* **Enquiring:** In this process a customer first visits to a Spa to collect all the information regarding treatments or packages. His / her details are taken and kept in a database for further follow ups. He/she may not be a member but follow-ups are always being done**.**



import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.\*;

public class enquiry extends JFrame implements ActionListener,FocusListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/trac.jpg"));

JLabel eid=new JLabel("Enquiry ID");

JLabel ename=new JLabel("Name");

JLabel eadd=new JLabel("Address");

JLabel egender=new JLabel("Gender");

JLabel emob=new JLabel("Mobile");

JLabel econtact=new JLabel("Contact");

JLabel email=new JLabel("E-mail");

JLabel epurpose=new JLabel("Purpose");

JLabel estatus=new JLabel("Status");

JLabel ehead=new JLabel("Enquiry Form");

JLabel edate=new JLabel("Date");

JLabel endate=new JLabel("Enquiry Date");

JTextField tid=new JTextField();

JTextField tname=new JTextField();

JTextArea tadd=new JTextArea();

JTextField tmob=new JTextField();

JTextField temail=new JTextField();

JTextField tndate=new JTextField();

JTextField tdate=new JTextField();

JTextArea tpurpose=new JTextArea();

JRadioButton rmale=new JRadioButton("Male");

JRadioButton rfemale=new JRadioButton("Female");

Calendar cal=Calendar.getInstance();

JRadioButton rcon=new JRadioButton("Confirm");

JRadioButton rpend=new JRadioButton("Pending");

JRadioButton rrej=new JRadioButton("Rejected");

JButton enew=new JButton("New",new ImageIcon("../Images/nbt.jpg"));

JButton esave=new JButton("Save",new ImageIcon("../Images/sbt.jpg"));

JButton esearch=new JButton("Search",new ImageIcon("../Images/sbtn.jpg"));

JButton eedit=new JButton("Edit",new ImageIcon("../Images/ebt.jpg"));

JButton ecancel=new JButton("Cancel",new ImageIcon("../Images/cbt.jpg"));

String strgen,strpen;

public enquiry()

{

setLayout(null);

setVisible(true);

getContentPane().setBackground(Color.WHITE);

lbl.setBounds(0,0,getWidth(),getHeight());

ehead.setBounds(300,30,300,30);

edate.setBounds(960,90,50,30);

tdate.setBounds(1030,90,100,30);

eid.setBounds(250,140,100,30);

tid.setBounds(370,140,100,30);

ename.setBounds(250,190,100,30);

tname.setBounds(370,190,180,30);

eadd.setBounds(250,240,100,30);

sradd.setBounds(370,240,170,90);

egender.setBounds(250,340,100,30);

rmale.setBounds(370,340,80,30);

rfemale.setBounds(470,340,100,30);

econtact.setBounds(250,390,100,30);

emob.setBounds(300,430,100,30);

tmob.setBounds(420,430,120,30);

email.setBounds(300,470,100,30);

temail.setBounds(420,470,170,30);

epurpose.setBounds(250,520,100,30);

srpurpose.setBounds(370,520,150,60);

endate.setBounds(700,90,120,30);

tndate.setBounds(840,90,100,30);

estatus.setBounds(250,590,100,30);

rcon.setBounds(360,590,100,30);

rpend.setBounds(470,590,100,30);

rrej.setBounds(580,590,100,30);

enew.setBounds(150,650,120,30);

esave.setBounds(280,650,120,30);

esearch.setBounds(410,650,120,30);

eedit.setBounds(540,650,120,30);

ecancel.setBounds(670,650,120,30);

add(epurpose);

add(srpurpose);

add(endate);

add(tndate);

add(email);

add(temail);

add(econtact);

add(emob);

add(tmob);

add(egender);

add(rmale);

add(rfemale);

bg.add(rmale);

bg.add(rfemale);

add(eid);

add(tid);

add(eadd);

add(sradd);

add(edate);

add(tdate);

add(ename);

add(tname);

add(ehead);

tdate.setText(""+cal.get(Calendar.DATE)+"/"+(cal.get(Calendar.MONTH)+1)+"/"+cal.get(Calendar.YEAR));

add(estatus);

bs.add(rpend);

bs.add(rrej);

bs.add(rcon);

add(rcon);

add(rrej);

add(rpend);

add(enew);

add(esave);

add(esearch);

add(eedit);

add(ecancel);

add(lbl);

enew.addActionListener(this);

esave.addActionListener(this);

esearch.addActionListener(this);

eedit.addActionListener(this);

ecancel.addActionListener(this);

esave.setEnabled(false);

tid.setEditable(false);

tdate.setEditable(false);

eedit.setEnabled(false);

tname.addFocusListener(this);

tmob.addFocusListener(this);

temail.addFocusListener(this);

tndate.addFocusListener(this);

}

public void focusLost(FocusEvent fe)

{

int i,tflg=0;

if(fe.getComponent()==tname)

{

strchk=tname.getText();

if(tname.getText().length()==0)

else

tflg=1;

if(tflg==1)

{

for(i=0;i<strchk.length();i++)

{

if(strchk.charAt(i)>='a'&&strchk.charAt(i)<='z'||strchk.charAt(i)>='A'&&strchk.charAt(i)<='Z'||strchk.charAt(i)==' ')

{

}

else

{

JOptionPane.showMessageDialog(null,"Invalid Name!!!");

break;

}

}

}

}

if(fe.getComponent()==tmob)

{

int tmflg=0;

strchk=tmob.getText();

if(tmob.getText().length()==0)

else

tmflg=1;

if(tmflg==1)

{

a:{for(i=0;i<strchk.length();i++)

{

if(Character.getNumericValue(strchk.charAt(i))>=0&&Character.getNumericValue(strchk.charAt(i))<=9)

{

}

else

{

JOptionPane.showMessageDialog(null,"Enter Only Digits!!");

break a;

}

if(strchk.length()!=10)

{

JOptionPane.showMessageDialog(null,"Enter 10-digits Mobile Number!!");

break a;

}}

}

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==enew)

{

eedit.setEnabled(false);

int ctr=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(6));

}

if(ctr<10)

tid.setText("E00"+ctr);

if(ctr>=10&&ctr<100)

tid.setText("E0"+ctr);

tname.setText("");

tadd.setText("");

temail.setText("");

tmob.setText("");

tndate.setText(tdate.getText());

bg.clearSelection();

bs.clearSelection();

tpurpose.setText("");

esave.setEnabled(true);

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==esave)

{

eedit.setEnabled(false);

int flg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn"); String str1="insert into enquirytbl(enqid,enqname,enqdate,enqaddress,enqgender,enqmobno,enqmail,enqpurpose,enqstatus)values(?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str1);

Statement st=con.createStatement();

if(tndate.getText().length()==0||tname.getText().length()==0||tadd.getText().length()==0||bg.getSelection()==null||tmob.getText().length()==0||temail.getText().length()==0||tpurpose.getText().length()==0||bs.getSelection()==null)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

if(flg==0)

{

ps.setString(1,tid.getText());

ps.setString(2,tname.getText());

ps.setString(3,tndate.getText());

ps.setString(4,tadd.getText());

if(rmale.isSelected())

strgen="Male";

if(rfemale.isSelected())

strgen="Female";

ps.setString(5,strgen);

ps.setString(6,tmob.getText());

ps.setString(7,temail.getText());

ps.setString(8,tpurpose.getText());

if(rcon.isSelected())

strpen="Confirm";

else if(rpend.isSelected())

strpen="Pending";

else

strpen="Rejected";

ps.setString(9,strpen);

ps.executeUpdate();

Statement st1=con.createStatement();

ResultSet res=st1.executeQuery("select \* from keytbl");

String tmp="";

while(res.next())

{

tmp="F00"+res.getString(6);

}

String str2="insert into followtbl(enquiryid,followid,status,fdate,fphone,fmail) values(?,?,?,?,?,?)";

PreparedStatement ps1=con.prepareStatement(str2);

ps1.setString(1,tid.getText());

ps1.setString(2,tmp);

ps1.setString(3,strpen);

ps1.setString(4,tndate.getText());

ps1.setString(5,tmob.getText());

ps1.setString(6,temail.getText());

ps1.executeUpdate();

String str3="update keytbl set followid=followid+1";

Statement st2=con.createStatement();

st2.executeUpdate(str3);

String str="update keytbl set enqid=enqid+1";

st.executeUpdate(str);

JOptionPane.showMessageDialog(null,"Saved");

esave.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==esearch)

{

int flg=0;

String tmp="";

String t=JOptionPane.showInputDialog("Enter Enquiry id to search");

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

Statement st=con.createStatement();

res=st.executeQuery("select \* from enquirytbl where enqid='"+t+"'");

while(res.next())

{

flg=1;

tid.setText(res.getString(1));

tname.setText(res.getString(2));

tndate.setText(res.getString(3));

tadd.setText(res.getString(4));

if(res.getString(5).equals("Male"))

{

rmale.setSelected(true);

rfemale.setSelected(false);

}

else

{

rmale.setSelected(false);

rfemale.setSelected(true);

}

tmob.setText(res.getString(6));

temail.setText(res.getString(7));

tpurpose.setText(res.getString(8));

tmp=res.getString(9);

if(tmp.equals("Confirm"))

rcon.setSelected(true);

else if(tmp.equals("Pending"))

rpend.setSelected(true);

else

rrej.setSelected(true);

}

if(flg==0&&t.length()!=0)

{

JOptionPane.showMessageDialog(null,"No Record Found!!");

esave.setEnabled(false);

}

eedit.setEnabled(true);

}

catch(Exception ee)

{

System.out.print(ee);

}

}

if(e.getSource()==eedit)

{

int flg=0,tflg=0;

int ctr1=0;

try

{

esave.setEnabled(false);

if(tid.getText().length()!=0)

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

if(rmale.isSelected())

strgen="Male";

if(rfemale.isSelected())

strgen="Female";

if(rcon.isSelected())

strpen="Confirm";

else if(rpend.isSelected())

strpen="Pending";

else

strpen="Rejected";

if(tndate.getText().length()==0||tname.getText().length()==0||tadd.getText().length()==0||bg.getSelection()==null||tmob.getText().length()==0||temail.getText().length()==0||tpurpose.getText().length()==0||bs.getSelection()==null)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

tflg=1;

}

if(tflg==0)

{

ResultSet res=st.executeQuery("select \* from followtbl where enquiryid='"+tid.getText()+"'");

while(res.next())

{

ctr1++;

}

res.close();

if(ctr1==1)

{

String str="update enquirytbl set enqname='"+tname.getText()+"',enqdate='"+tndate.getText()+"',enqaddress='"+tadd.getText()+"',enqgender='"+strgen+"',enqmobno='"+tmob.getText()+"',enqmail='"+temail.getText()+"',enqpurpose='"+tpurpose.getText()+"',enqstatus='"+strpen+"' where enqid='"+tid.getText()+"'";

st.executeUpdate(str);

st.executeUpdate("update followtbl set status='"+strpen+"' where enquiryid='"+tid.getText()+"'");

JOptionPane.showMessageDialog(null,"Record Updated!!");

eedit.setEnabled(false);

}

else

{

JOptionPane.showMessageDialog(null,"Data can't be updated!!");

}

}

}

else

{

if(tflg==0)

JOptionPane.showMessageDialog(null,"Search Request For Update");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==ecancel)

{

this.dispose();

}

}

public static void main(String argv[])

throws IOException

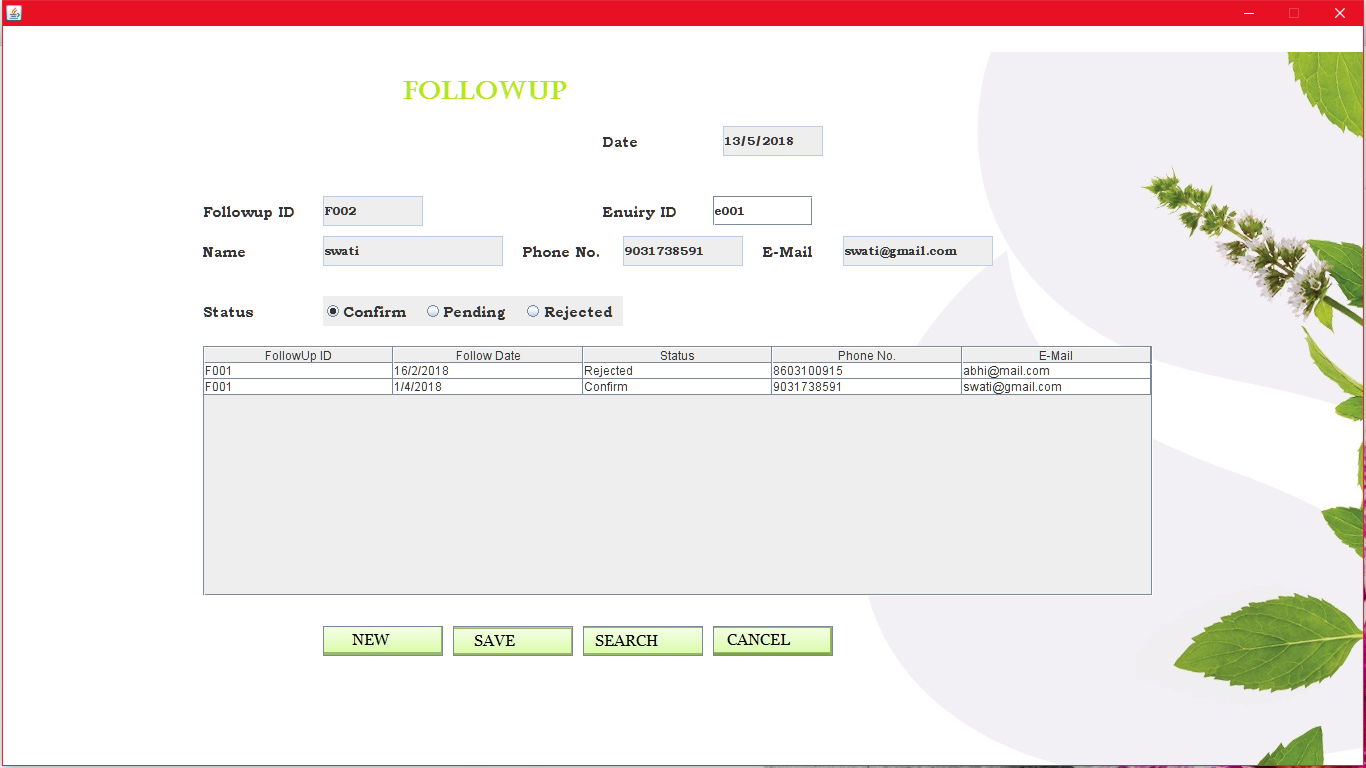
{

enquiry obj=new enquiry();

}

}

* **Follow-Up:** Here the spa stores the all records of past coming persons for enquiry. Using this the Upcoming Offers and Packages can be delivered to them for advertisement purpose.



import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.\*;

public class followup extends JFrame implements ActionListener,FocusListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/trac.jpg"));

JLabel fdate=new JLabel("Date");

JLabel feid=new JLabel("Enuiry ID");

JLabel fid=new JLabel("Followup ID");

JLabel fstatus=new JLabel("Status");

JLabel fhead=new JLabel("Followup");

JLabel fname=new JLabel("Name");

JLabel fph=new JLabel("Phone No.");

JLabel fmail=new JLabel("E-Mail");

JTextField tdate=new JTextField();

JTextField teid=new JTextField();

JTextField tfid=new JTextField();

JTextField tname=new JTextField();

JTextField tph=new JTextField();

JTextField tmail=new JTextField();

JRadioButton rcon=new JRadioButton("Confirm");

JRadioButton rpend=new JRadioButton("Pending");

JRadioButton rrej=new JRadioButton("Rejected");

JButton fnew=new JButton("New",new ImageIcon("../Images/nbt.jpg"));

JButton fsave=new JButton("Save",new ImageIcon("../Images/sbt.jpg"));

JButton fsearch=new JButton("Search",new ImageIcon("../Images/sbtn.jpg"));

JButton fcancel=new JButton("Cancel",new ImageIcon("../Images/cbt.jpg"));

Calendar cal=Calendar.getInstance();

JScrollPane pnl=new JScrollPane();

JTable tbl;

String []col={"FollowUp ID","Follow Date","Status","Phone No.","E-Mail"};

String foid,fdat,sts,phno,email;

Object [][]data={{foid,fdat,sts,phno,email}};

DefaultTableModel model=new DefaultTableModel(col,0);

int flg=0;

public followup()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

lbl.setBounds(0,0,getWidth(),getHeight());

fid.setBounds(200,170,100,30);

tfid.setBounds(320,170,100,30);

feid.setBounds(600,170,100,30);

teid.setBounds(710,170,100,30);

fname.setBounds(200,210,100,30);

tname.setBounds(320,210,180,30);

fph.setBounds(520,210,100,30);

tph.setBounds(620,210,120,30);

fmail.setBounds(760,210,100,30);

tmail.setBounds(840,210,150,30);

fhead.setBounds(400,50,200,30);

fstatus.setBounds(200,270,100,30);

rcon.setBounds(320,270,100,30);

rpend.setBounds(420,270,100,30);

rrej.setBounds(520,270,100,30);

fdate.setBounds(600,100,100,30);

tdate.setBounds(720,100,100,30);

fnew.setBounds(320,600,120,30);

fsave.setBounds(450,600,120,30);

fsearch.setBounds(580,600,120,30);

fcancel.setBounds(710,600,120,30);

tdate.setText(""+cal.get(Calendar.DATE)+"/"+(cal.get(Calendar.MONTH)+1)+"/"+cal.get(Calendar.YEAR));

tbl=new JTable(data,col);

pnl=new JScrollPane(tbl);

pnl.setBounds(200,320,950,250);

add(pnl);

add(fname);

add(fph);

add(fmail);

add(tname);

add(tph);

add(tmail);

add(fdate);

add(tdate);

bs.add(rcon);

bs.add(rrej);

bs.add(rpend);

add(feid);

add(teid);

add(rcon);

add(rrej);

add(rpend);

add(fstatus);

add(fid);

add(tfid);

add(fhead);

add(fcancel);

add(fsearch);

add(fsave);

add(fnew);

add(lbl);

fnew.addActionListener(this);

fsearch.addActionListener(this);

fsave.addActionListener(this);

fcancel.addActionListener(this);

teid.addFocusListener(this);

tdate.setEditable(false);

tfid.setEditable(false);

fsave.setEnabled(false);

tname.setEditable(false);

tph.setEditable(false);

tmail.setEditable(false);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==fnew)

{

int ctr=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(7));

}

if(ctr<10)

tfid.setText("F00"+ctr);

if(ctr>=10&&ctr<100)

tfid.setText("F0"+ctr);

teid.setText("");

tname.setText("");

tph.setText("");

tmail.setText("");

bs.clearSelection();

fsave.setEnabled(true);

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==fsave)

{

String strpen="";

int qflg=0;

if(flg==0&&tbl.getRowCount()>0)

{

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

Statement st=con.createStatement();

if(feid.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

qflg=1;

}

if(qflg==0)

{

res=st.executeQuery("select \* from enquirytbl where enqid='"+teid.getText()+"'");

String r="";

String r1="";

String r2="";

while(res.next())

{

r=res.getString(2);

r1=res.getString(7);

r2=res.getString(8);

}

String str="insert into followtbl(enquiryid,followid,status,fdate) values(?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

ps.setString(1,teid.getText());

ps.setString(2,tfid.getText());

if(rcon.isSelected())

strpen="Confirm";

else

{

if(rpend.isSelected())

strpen="Pending";

else

strpen="Rejected";

}

ps.setString(3,strpen);

ps.setString(4,tdate.getText());

ps.executeUpdate();

if(strpen.equals("Confirm")||strpen.equals("Rejected"))

{

String str2="update enquirytbl set enqstatus='"+strpen+"' where enqid='"+teid.getText()+"'";

Statement st2=con.createStatement();

st2.executeUpdate(str2);

}

String str1="update keytbl set followid=followid+1";

Statement st1=con.createStatement();

st1.executeUpdate(str1);

flg=0;

JOptionPane.showMessageDialog(null,"Data Saved!!!!");

Vector v=new Vector();

v.add(tfid.getText());

v.add(teid.getText());

v.add(r);

v.add(tdate.getText());

v.add(strpen);

v.add(r1);

v.add(r2);

model.addRow(v);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

else

{

JOptionPane.showMessageDialog(null,"Can't Save!!");

}

}

if(e.getSource()==fsearch)

{

int sflg=0;

try

{

String m=JOptionPane.showInputDialog("Enter Enquiry ID to be search:");

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

String t="";

ResultSet res1=st.executeQuery("select \* from enquirytbl where enqid='"+m+"'");

while(res1.next())

{

sflg=1;

teid.setText(res1.getString(1));

tname.setText(res1.getString(2));

tph.setText(res1.getString(6));

tmail.setText(res1.getString(7));

t=res1.getString(9);

if(t.equals("Confirm"))

rcon.setSelected(true);

else if(t.equals("Pending"))

rpend.setSelected(true);

else

rrej.setSelected(true);

}

ResultSet res2=st.executeQuery("select \* from followtbl where enquiryid='"+m+"'");

Vector v;

while(res2.next())

{

v=new Vector();

v.add(res2.getString(2));

v.add(res2.getString(4));

v.add(res2.getString(3));

v.add(res2.getString(5));

v.add(res2.getString(6));

model.addRow(v);

}

if(sflg==0)

{

JOptionPane.showMessageDialog(null,"No Such Record Found!!");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==fcancel)

{

this.dispose();

}

}

public void focusLost(FocusEvent fe)

{

if(fe.getComponent()==teid)

{

flg=0;

if(teid.getText().length()!=0)

{

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

String t="";

tname.setText("");

tph.setText("");

tmail.setText("");

bs.clearSelection();

ResultSet res1=st.executeQuery("select \* from enquirytbl where enqid='"+teid.getText()+"'");

while(res1.next())

{

tname.setText(res1.getString(2)); tmail.setText(res1.getString(7));

tph.setText(res1.getString(6));

t=res1.getString(9);

if(t.equals("Confirm"))

rcon.setSelected(true);

else if(t.equals("Pending"))

rpend.setSelected(true);

else

rrej.setSelected(true);

}

ResultSet res2=st.executeQuery("select \* from followtbl where enquiryid='"+teid.getText()+"'");

Vector v;

while(res2.next())

{

v=new Vector();

v.add(res2.getString(2));

v.add(res2.getString(4));

v.add(res2.getString(3));

v.add(res2.getString(5));

v.add(res2.getString(6));

model.addRow(v);

}

if(model.getRowCount()>0)

{

String s=""+tbl.getValueAt(tbl.getRowCount()-1,2);

if(s.equals("Confirm")||s.equals("Rejected"))

flg=1;

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

}

}

public static void main(String argv[])throws IOException

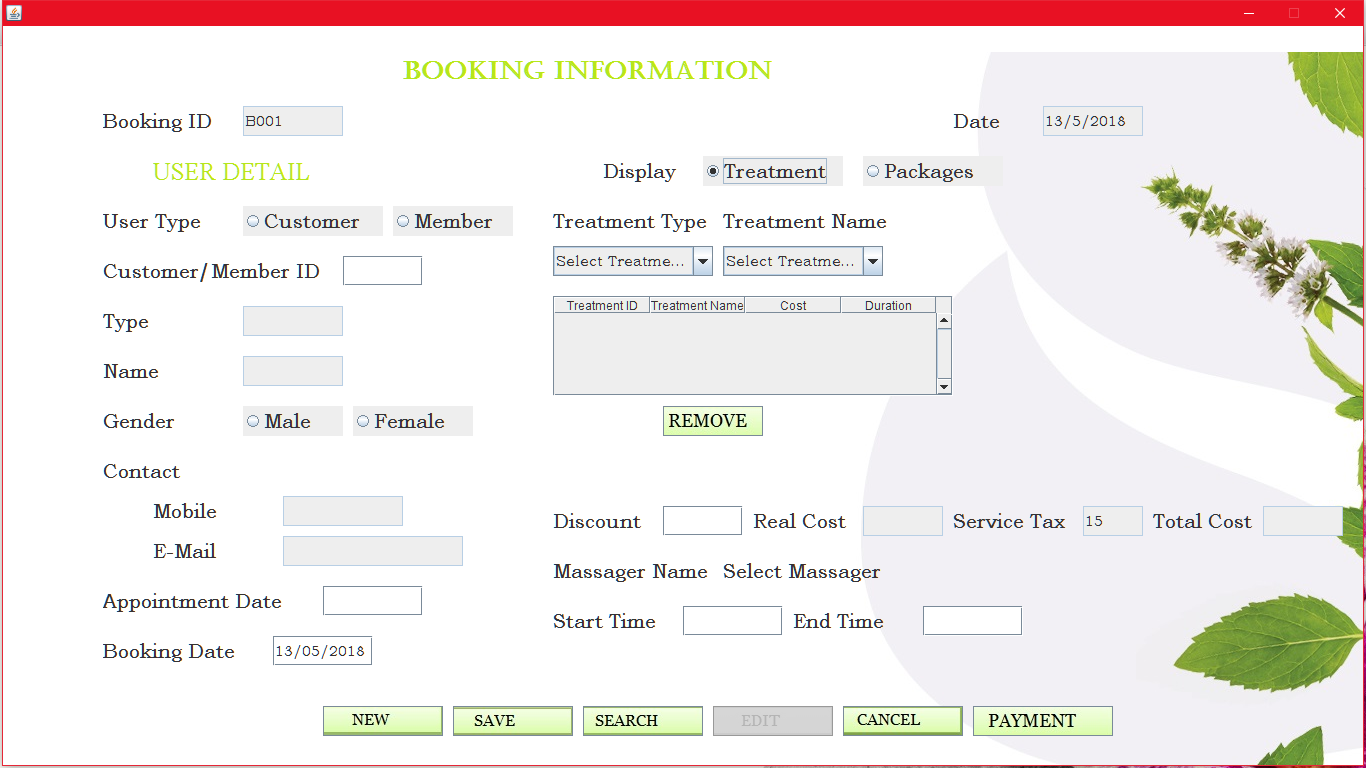
{

followup obj=new followup();

}

}

* **Booking:**Here booking refers to the information regarding the details of customer and the treatment or package he/she adopted for or if he/she is subscribed member then his/her membership package details and time and date for the treatment.



import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.\*;

public class booking extends JFrame implements ActionListener,FocusListener,ItemListener,MouseListener,MouseMotionListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/trac.jpg"));

String strtype="";

String strdis="";

String strgen="";

String strpay="";

Double rt=0.0;

JLabel bid=new JLabel("Booking ID");

JLabel bctype=new JLabel("User Type");

JLabel bcid=new JLabel("Customer/Member ID");

JLabel bname=new JLabel("Name");

JLabel bgen=new JLabel("Gender");

JLabel bcont=new JLabel("Contact");

JLabel bmob=new JLabel("Mobile");

JLabel bemail=new JLabel("E-Mail");

JLabel bhead=new JLabel("BOOKING INFORMATION");

JLabel bdate=new JLabel("Date");

JLabel bapdate=new JLabel("Appointment Date");

JLabel btreat=new JLabel("Treatment Type");

JLabel btname=new JLabel("Treatment Name");

JLabel bdisp=new JLabel("Display");

JLabel luser=new JLabel("USER DETAIL");

JLabel lfaci=new JLabel("FACILITY DETAILS");

JLabel lpack=new JLabel("Package Name");

JLabel lpackid=new JLabel("Package ID");

JLabel loff=new JLabel("Discount");

JLabel lrtotal=new JLabel("Real Cost");

JLabel ltotal=new JLabel("Total Cost");

JLabel lmass=new JLabel("Massager Name");

JLabel lmassf=new JLabel("Select Massager");

JLabel lmas=new JLabel("ID...");

JLabel ldur=new JLabel("Duration");

JLabel lendt=new JLabel("End Time");

JLabel lstart=new JLabel("Start Time");

JLabel lbdt=new JLabel("Booking Date");

JLabel ltype=new JLabel("Type");

JLabel ldis=new JLabel("");

JLabel ltax=new JLabel("Service Tax");

JLabel ldiet=new JLabel("Diet Name");

JTextField tdate=new JTextField();

JTextField tid=new JTextField();

JTextField tcid=new JTextField();

JTextField tname=new JTextField();

JTextField tcont=new JTextField();

JTextField tmob=new JTextField();

JTextField temail=new JTextField();

JTextField tapdate=new JTextField();

JTextField tpackid=new JTextField();

JTextField toff=new JTextField();

JTextField ttotal=new JTextField();

JTextField trtotal=new JTextField();

JTextField tend=new JTextField();

JTextField tstart=new JTextField();

JTextField tbdt=new JTextField();

JTextField ttype=new JTextField();

JTextField ttax=new JTextField();

JComboBox ctype=new JComboBox();

JComboBox cname=new JComboBox();

JComboBox cpack=new JComboBox();

JComboBox cmas=new JComboBox();

JComboBox cdiet=new JComboBox();

JRadioButton rtreat=new JRadioButton("Treatment");

JRadioButton rpack=new JRadioButton("Packages");

JRadioButton rcust=new JRadioButton("Customer");

JRadioButton rmem=new JRadioButton("Member");

JRadioButton rmale=new JRadioButton("Male");

JRadioButton rfemale=new JRadioButton("Female");

JScrollPane pnl=new JScrollPane();

JScrollPane pnl1=new JScrollPane();

JScrollPane pnl2=new JScrollPane();

JButton bnew=new JButton("New",new ImageIcon("../Images/nbt.jpg"));

JButton bsave=new JButton("Save",new ImageIcon("../Images/sbt.jpg"));

JButton bsearch=new JButton("Search",new ImageIcon("../Images/sbtn.jpg"));

JButton bedit=new JButton("Edit",new ImageIcon("../Images/ebt.jpg"));

JButton bcancel=new JButton("Cancel",new ImageIcon("../Images/cbt.jpg"));

JButton bpay=new JButton("Payment",new ImageIcon("../Images/pbt.jpg"));

JButton bremove=new JButton("Removen",new ImageIcon("../Images/rbt.jpg"));

JButton bchk=new JButton("Check");

JCheckBox bcfrm=new JCheckBox("Confirm");

JTable tbl;

String []col={"Treatment ID","Type","Name","Cost"};

String pid,ptyp,pnm,pcost;

Object[][]data={{pid,ptyp,pnm,pcost}};

DefaultTableModel model=new DefaultTableModel(col,0);

JScrollPane pnl3=new JScrollPane();

JTable tbl1;

String []col1={"Treatment ID","Treatment Name","Cost","Duration"};

String ttid,ttyp,tnm,tcost;

Object [][]data1={{ttid,ttyp,tnm,tcost}};

DefaultTableModel model1=new DefaultTableModel(col1,0);

Calendar cal=Calendar.getInstance();

String tmdis="";

String strdiet="";

String strdname="";

public booking()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

bhead.setBounds(400,30,380,30);

bdate.setBounds(950,80,80,30);

tdate.setBounds(1040,80,100,30);

bid.setBounds(100,80,120,30);

tid.setBounds(240,80,100,30);

luser.setBounds(150,130,170,30);

bdisp.setBounds(600,130,80,30);

rtreat.setBounds(700,130,140,30);

rpack.setBounds(860,130,140,30);

bctype.setBounds(100,180,120,30);

rcust.setBounds(240,180,140,30);

rmem.setBounds(390,180,120,30);

bcid.setBounds(100,230,220,30);

tcid.setBounds(340,230,80,30);

ltype.setBounds(100,280,120,30);

ttype.setBounds(240,280,100,30);

bname.setBounds(100,330,120,30);

tname.setBounds(240,330,100,30);

bgen.setBounds(100,380,120,30);

rmale.setBounds(240,380,100,30);

rfemale.setBounds(350,380,120,30);

bcont.setBounds(100,430,120,30);

bmob.setBounds(150,470,120,30);

tmob.setBounds(280,470,120,30);

bemail.setBounds(150,510,120,30);

temail.setBounds(280,510,180,30);

bapdate.setBounds(100,560,200,30);

tapdate.setBounds(320,560,100,30);

lbdt.setBounds(100,610,150,30);

tbdt.setBounds(270,610,100,30);

ldis.setBounds(550,430,150,30);

loff.setBounds(550,480,100,30);

toff.setBounds(660,480,80,30);

lrtotal.setBounds(750,480,100,30);

trtotal.setBounds(860,480,80,30);

ltax.setBounds(950,480,120,30);

ttax.setBounds(1080,480,60,30);

ltotal.setBounds(1150,480,100,30);

ttotal.setBounds(1260,480,80,30);

lmass.setBounds(550,530,160,30);

lmassf.setBounds(720,530,180,30);

lmas.setBounds(910,530,100,30);

cmas.setBounds(1020,530,160,30);

btreat.setBounds(550,180,160,30);

ctype.setBounds(550,220,160,30);

btname.setBounds(720,180,170,30);

cname.setBounds(720,220,160,30);

lpack.setBounds(870,180,160,30);

cpack.setBounds(870,220,150,30);

lpackid.setBounds(1030,180,120,30);

tpackid.setBounds(1140,180,100,30);

lstart.setBounds(550,580,120,30);

tstart.setBounds(680,580,100,30);

lendt.setBounds(790,580,120,30);

tend.setBounds(920,580,100,30);

bnew.setBounds(320,680,120,30);

bsave.setBounds(450,680,120,30);

bsearch.setBounds(580,680,120,30);

bedit.setBounds(710,680,120,30);

bcancel.setBounds(840,680,120,30);

bpay.setBounds(970,680,140,30);

tbl=new JTable(data,col);

pnl=new JScrollPane(tbl);

add(pnl);

pnl.setBounds(870,270,400,100);

tbl1=new JTable(data1,col1);

pnl3=new JScrollPane(tbl1);

add(pnl3);

pnl3.setBounds(550,270,400,100);

ldiet.setBounds(550,376,100,30);

cdiet.setBounds(620,380,100,20);

bcfrm.setBounds(730,380,100,20);

bchk.setBounds(840,380,100,20);

bremove.setBounds(660,380,100,30);

ldiet.setVisible(false);

cdiet.setVisible(false);

bcfrm.setVisible(false);

bchk.setVisible(false);

add(ltax);

add(ttax);

add(ltype);

add(ttype);

add(ldis);

add(lbdt);

add(tbdt);

add(ldiet);

add(cdiet);

add(bcfrm);

add(bchk);

add(bremove);

add(lendt);

add(tend);

add(tstart);

add(lstart);

add(lmas);

add(cmas);

add(lmass);

add(lmassf);

add(loff);

add(ltotal);

add(toff);

add(ttotal);

add(lpackid);

add(tpackid);

add(lpack);

add(cpack);

add(btname);

add(cname);

add(ctype);

add(btreat);

add(bcid);

add(tcid);

add(bapdate);

add(tapdate);

add(bcont);

add(bmob);

add(tmob);

add(bemail);

add(temail);

add(bgen);

add(rmale);

add(rfemale);

bg.add(rfemale);

bg.add(rmale);

add(bdisp);

add(rtreat);

add(rpack);

bd.add(rtreat);

bd.add(rpack);

add(bname);

add(tname);

add(bid);

add(tid);

add(luser);

tid.setEditable(false);

tdate.setEditable(false);

tname.setEditable(false);

tmob.setEditable(false);

temail.setEditable(false);

add(bdate);

add(tdate);

bt.add(rcust);

bt.add(rmem);

add(rcust);

add(rmem);

add(bhead);

add(bctype);

add(cpack);

add(bnew);

add(lrtotal);

add(trtotal);

add(bnew);

add(bsave);

add(bsearch);

add(bedit);

add(bcancel);

add(bpay);

cpack.setVisible(false);

lpack.setVisible(false);

lpackid.setVisible(false);

tpackid.setVisible(false);

pnl.setVisible(false);

pnl3.setVisible(false);

btreat.setVisible(false);

btname.setVisible(false);

ctype.setVisible(false);

cname.setVisible(false);

bremove.setVisible(false);

tdate.setText(""+cal.get(Calendar.DATE)+"/"+(cal.get(Calendar.MONTH)+1)+"/"+cal.get(Calendar.YEAR));

cpack.addItem("Select Package");

bnew.addActionListener(this);

bsave.addActionListener(this);

bsearch.addActionListener(this);

bedit.addActionListener(this);

bcancel.addActionListener(this);

bpay.addActionListener(this);

ctype.addActionListener(this);

cname.addActionListener(this);

cpack.addActionListener(this);

toff.addFocusListener(this);

tcid.addFocusListener(this);

rmem.addActionListener(this);

rcust.addActionListener(this);

rtreat.addItemListener(this);

lmassf.addMouseListener(this);

rpack.addItemListener(this);

cmas.addActionListener(this);

bremove.addActionListener(this);

bcfrm.addMouseListener(this);

bchk.addMouseListener(this);

cmas.setVisible(false);

lmas.setVisible(false);

bsave.setEnabled(false);

bedit.setEnabled(false);

tapdate.addFocusListener(this);

tbdt.addFocusListener(this);

trtotal.setEditable(false);

ttotal.setEditable(false);

ttype.setEditable(false);

ttax.setEditable(false);

try

{

cdiet.removeAllItems();

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("Select \* from diettbl");

cdiet.addItem("Select Diet");

while(res.next())

{

cdiet.addItem(res.getString(2));

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

public void mousePressed(MouseEvent me)

{}

public void mouseReleased(MouseEvent me)

{}

public void mouseExited(MouseEvent me)

{}

public void mouseDragged(MouseEvent me)

{}

public void mouseClicked(MouseEvent me)

{

if(me.getSource()==lmassf)

{

if(rtreat.isSelected()==true)

{

if(ctype.getSelectedIndex()==0||ctype.getItemCount()==0)

{

JOptionPane.showMessageDialog(null,"First Select Treatment Type!!!!");

}

else

{

cmas.setVisible(true);

lmas.setVisible(true);

String ts="";

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

if(cmas.getItemCount()!=0)

cmas.removeAllItems();

String str="select \* from stafftbl";

ResultSet res=st.executeQuery(str);

cmas.addItem("Select Massager");

while(res.next())

{

cmas.addItem(res.getString(2));

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

}

}

if(me.getComponent()==bchk)

{

if(me.getClickCount()==2)

{

if(cdiet.getSelectedItem().toString().equals("Select Diet"))

else

displaydiet disobj=new displaydiet(cdiet.getSelectedItem().toString());

}

}

if(me.getComponent()==bcfrm)

{

if(bcfrm.isSelected())

{

if(cdiet.getSelectedItem().toString().equals("Select Diet"))

{

JOptionPane.showMessageDialog(null,"First Select Diet!!");

bcfrm.setSelected(false);

}

else

{

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from diettbl where diettype='"+cdiet.getSelectedItem().toString()+"'");

while(res.next())

{

strdiet=res.getString(1);

strdname=res.getString(2);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

}

}

}

public void mouseMoved(MouseEvent me)

{}

public void mouseEntered(MouseEvent me)

{}

public void find(String t)

{

try

{

ttax.setText("15");

String tmp="";

String tm="";

String tp="";

String strp="";

int flg=0;

String tn="";

tmdis="15";

String tnd="";

Connection con= DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

ResultSet res;

Statement st=con.createStatement();

res=st.executeQuery("select \* from bookingtbl where bookid='"+t+"'");

while(res.next())

{

bedit.setEnabled(true);

flg=1;

tid.setText(res.getString(2));

if(res.getString(3).equals("Customer"))

rcust.setSelected(true);

else

rmem.setSelected(true);

tmp=res.getString(4);

tcid.setText(tmp);

tapdate.setText(res.getString(5));

tbdt.setText(res.getString(1));

if(res.getString(6).equals("Treatment"))

rtreat.setSelected(true);

else

rpack.setSelected(true);

tm=res.getString(7);

tpackid.setText(tm);

toff.setText(res.getString(8));

trtotal.setText(res.getString(9));

ttotal.setText(res.getString(10));

lmassf.setText(res.getString(12));

tstart.setText(res.getString(13));

tend.setText(res.getString(14));

tnd=res.getString(16);

if(tnd.length()!=0)

{

ldiet.setVisible(true);

cdiet.setVisible(true);

bcfrm.setVisible(true);

bchk.setVisible(true);

Statement st5=con.createStatement();

ResultSet res5;

res5=st5.executeQuery("select \* from diettbl where dietid='"+tnd+"'");

while(res5.next())

{

cdiet.setSelectedItem(""+res5.getString(2));

bcfrm.setSelected(true);

}

}

}

if(flg==0&&t.length()!=0)

{

JOptionPane.showMessageDialog(null,"No Record Found!!");

}

if(rcust.isSelected()==true)

{

ResultSet res1=st.executeQuery("select \* from customertbl where custid='"+tmp+"'");

while(res1.next())

{

tname.setText(res1.getString(3));

if(res1.getString(7).equals("Male"))

rmale.setSelected(true);

else

rfemale.setSelected(true);

tmob.setText(res1.getString(5));

temail.setText(res1.getString(6));

}

}

if(rmem.isSelected()==true)

{

ResultSet res1=st.executeQuery("select \* from membertbl where mem\_id='"+tmp+"'");

while(res1.next())

{

tname.setText(res1.getString(3));

if(res1.getString(5).equals("Male"))

rmale.setSelected(true);

else

rfemale.setSelected(true);

ldis.setText("Elegible for "+res1.getString(15)+" discount");

tmob.setText(res1.getString(8));

temail.setText(res1.getString(9));

ttype.setText(res1.getString(4));

}

}

if(rtreat.isSelected()==true)

{

ResultSet res2=st.executeQuery("select \* from booktbl where bookid='"+t+"'");

Vector v;

model1.setRowCount(0);

while(res2.next())

{

v=new Vector();

v.add(res2.getString(2));

model1.addRow(v);

}

for(int i=0;i<tbl1.getRowCount();i++)

{

res2=st.executeQuery("select \* from treatmenttbl where treat\_id='"+tbl1.getValueAt(i,0).toString()+"'");

while(res2.next())

{

tbl1.setValueAt(res2.getString(3),i,1);

tbl1.setValueAt(res2.getString(4),i,2);

tbl1.setValueAt(res2.getString(5),i,3);

}

}

}

if(rpack.isSelected()==true)

{

System.out.println(tm);

String str="select \* from packagetbl where pack\_id='"+tm+"'";

ResultSet res2=st.executeQuery(str);

String stp="";

while(res2.next())

{

int j=0;

stp=res2.getString(2);

String s[]={res2.getString(3),res2.getString(4),res2.getString(5),res2.getString(6),res2.getString(7)};

while(j<5)

{

str="select \* from treatmenttbl where treat\_id='"+s[j]+"'";

Statement st1=con.createStatement();

ResultSet res3=st1.executeQuery(str);

Vector v=new Vector();

while(res3.next())

{

v.add(res3.getString(1));

v.add(res3.getString(2));

v.add(res3.getString(3));

v.add(res3.getString(4));

model.addRow(v);

}

j++;

}

}

if(cpack.getItemCount()==1)

{

res2=st.executeQuery("Select \* from packagetbl");

while (res2.next())

{

cpack.addItem(res2.getString(2));

}

}

cpack.setSelectedItem(stp);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

public void pack()

{

if(rpack.isSelected()==true)

{

cpack.setVisible(true);

lpack.setVisible(true);

lpackid.setVisible(true);

tpackid.setVisible(true);

pnl.setVisible(true);

lmassf.setText("N/A");

lmas.setVisible(false);

cmas.setVisible(false);

}

else

{

cpack.setVisible(false);

lpack.setVisible(false);

lpackid.setVisible(false);

tpackid.setVisible(false);

pnl.setVisible(false);

}

}

public void treat()

{

if(rtreat.isSelected()==true)

{

toff.setText("");

trtotal.setText("");

ttotal.setText("");

btreat.setVisible(true);

btname.setVisible(true);

ctype.setVisible(true);

cname.setVisible(true);

pnl3.setVisible(true);

model1.setRowCount(0);

bremove.setVisible(true);

lmassf.setText("N/A");

lmas.setText("ID...");

ttax.setText("15");

}

else

{

btreat.setVisible(false);

btname.setVisible(false);

ctype.setVisible(false);

cname.setVisible(false);

pnl3.setVisible(false);

bremove.setVisible(false);

}

}

public void itemStateChanged(ItemEvent ie)

{

if(ie.getSource()==rtreat||ie.getSource()==rpack)

{

if(ie.getStateChange()==ItemEvent.SELECTED)

{

treat();

pack();

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bremove)

{

String rn=JOptionPane.showInputDialog("Enter Row No.");

if(tbl1.getRowCount()!=0)

{

if(Integer.parseInt(rn)>=1&&Integer.parseInt(rn)<=tbl1.getRowCount())

model1.removeRow(Integer.parseInt(rn)-1);

else

JOptionPane.showMessageDialog(null,"Invalid Row Number!!");

}

}

if(e.getSource()==cmas)

{

try

{

if(cmas.getSelectedIndex()==0||cmas.getItemCount()==0)

{

}

else

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

String str="select \* from stafftbl where sname='"+cmas.getSelectedItem().toString()+"'";

ResultSet res=st.executeQuery(str);

while(res.next())

{

lmassf.setText(res.getString(2));

lmas.setText(res.getString(1));

}

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==bcancel)

{

this.dispose();

}

if(e.getSource()==bnew)

{

SimpleDateFormat pr=new SimpleDateFormat("dd/MM/yyyy");

bedit.setEnabled(false);

bd.clearSelection();

pack();

treat();

try

{

int ctr=0;

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(8));

}

if(ctr<10)

tid.setText("B00"+ctr);

if(ctr>=10&&ctr<100)

tid.setText("B0"+ctr);

tcid.setText("");

tname.setText("");

bg.clearSelection();

tmob.setText("");

temail.setText("");

tapdate.setText("");

Date t=pr.parse(tdate.getText());

tbdt.setText(""+pr.format(t));

if(ctype.getItemCount()==0)

{

ResultSet res1=st.executeQuery("select \* from typetbl");

ctype.addItem("Select Treatment Type");

while(res1.next())

{

ctype.addItem(res1.getString(2));

}

}

ttype.setText("");

ldis.setText("");

tpackid.setText("");

toff.setText("");

trtotal.setText("");

ttotal.setText("");

if(cpack.getItemCount()==1)

{

ResultSet res1=st.executeQuery("select \* from packagetbl");

while(res1.next())

{

cpack.addItem(res1.getString(2));

}

}

else

{

cpack.setSelectedIndex(0);

}

bt.clearSelection();

tend.setText("");

tstart.setText("");

bsave.setEnabled(true);

}

catch(Exception ee)

{

System.out.println(ee);

}

rt=0.0;

}

if(e.getSource()==ctype)

{

ttax.setText("15");

String t=ctype.getSelectedItem().toString();

if(t.equals("Weight Loss"))

{

ldiet.setVisible(true);

cdiet.setVisible(true);

bcfrm.setSelected(false);

bcfrm.setVisible(true);

bchk.setVisible(true);

}

else

{

ldiet.setVisible(false);

cdiet.setVisible(false);

bcfrm.setVisible(false);

bchk.setVisible(false);

}

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

cname.removeAllItems();

cname.addItem("Select Treatment Name");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from treatmenttbl where treat\_type='"+t+"'");

while(res.next())

{

cname.addItem(res.getString(3));

}

}

catch(Exception ee)

{

System.out.println(ee);

}

if(lmassf.getText().equals("N/A"))

{

lmas.setText("ID...");

lmas.setVisible(false);

cmas.setVisible(false);

}

}

if(e.getSource()==cname)

{

ttax.setText("15");

String t="";

int flg=0;

String tmp="";

if(cname.getItemCount()!=0)

t=cname.getSelectedItem().toString();

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from treatmenttbl where treat\_name='"+t+"'");

while(res.next())

{

Vector vec=new Vector();

String trt="";

trt=res.getString(1);

for(int i=0;i<tbl1.getRowCount();i++)

{

if(trt.equals(tbl1.getValueAt(i,0)))

flg=1;

}

if(flg==1)

JOptionPane.showMessageDialog(null,"Already Entered!!!");

else

{

vec.add(trt);

vec.add(res.getString(3));

tmp=res.getString(4);

vec.add(tmp);

rt=rt+Double.parseDouble(tmp);

vec.add(res.getString(5));

model1.addRow(vec);

toff.setText("0");

}

}

trtotal.setText(""+rt);

double m=Double.parseDouble(trtotal.getText());

double ttx=Double.parseDouble(ttax.getText());

ttotal.setText(trtotal.getText());

if(rmem.isSelected()==true)

{

double tv=m-(m\*(Double.parseDouble(tmdis)/100));

double tm=tv+(tv\*(ttx/100));

ttotal.setText(""+tm);

}

if(rcust.isSelected()==true)

{

double tm=m+(m\*(ttx/100));

ttotal.setText(""+tm);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==bpay)

{

payment obj=new payment();

}

if(e.getSource()==bsearch)

{

sflg=1;

String t=JOptionPane.showInputDialog("Enter Booking id:");

find(t);

}

if(e.getSource()==cpack&&sflg!=1)

{

ttax.setText("15");

String t=cpack.getSelectedItem().toString();

String stn[]=new String[5];

double tc=0,ts=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from packagetbl where pack\_name='"+t+"'");

while(res.next())

{

tpackid.setText(res.getString(1));

stn[0]=res.getString(3);

stn[1]=res.getString(4);

stn[2]=res.getString(5);

stn[3]=res.getString(6);

stn[4]=res.getString(7);

toff.setText(res.getString(9));

trtotal.setText(res.getString(8));

tc=Double.parseDouble(res.getString(10));

}

ts=Double.parseDouble(ttax.getText());

if(rmem.isSelected()==true)

{

double m=tc-(tc\*(Double.parseDouble(tmdis)/100));

double tv=m+(m\*(ts/100));

ttotal.setText(""+tv);

}

if(rcust.isSelected()==true)

{

double tv=tc+(tc\*(ts/100));

ttotal.setText(""+tv);

}

int p=0;

while(p<5)

{

ResultSet res1=st.executeQuery("select \* from treatmenttbl where treat\_id='"+stn[p]+"'");

Vector v=new Vector();

while(res1.next())

{

v.add(res1.getString(1));

v.add(res1.getString(2));

v.add(res1.getString(3));

v.add(res1.getString(4));

model.addRow(v);

}

p++;

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==bsave)

{

bedit.setEnabled(false);

int flg=0;

try

{

String tmid="",tmname="";

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="insert into bookingtbl(bookdate,bookid,usertype,cust\_mem\_id,appointdate,disptype,packid,discount,realcost,totalcost,massagerid,massagername,starttime,endtime,pystate,dietid)values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

if(tid.getText().length()==0||tcid.getText().length()==0||bt.getSelection()==null||tname.getText().length()==0||bg.getSelection()==null||tmob.getText().length()==0||temail.getText().length()==0||tapdate.getText().length()==0||tbdt.getText().length()==0||bd.getSelection()==null||trtotal.getText().length()==0||ttotal.getText().length()==0||toff.getText().length()==0)

{

flg=1;

JOptionPane.showMessageDialog(null,"Empty Fields!!");

}

else

{

if(tstart.getText().length()==0||tend.getText().length()==0)

{

flg=1;

JOptionPane.showMessageDialog(null,"Empty Timings!!");

}

else

{

if(rtreat.isSelected())

{

if(lmas.getText().equals("ID...")||lmassf.equals("N/A"))

{

Statement st1=con.createStatement();

ResultSet res1;

res1=st1.executeQuery("select \* from stafftbl");

n:{while(res1.next())

{

tmid=res1.getString(1);

tmname=res1.getString(2);

if(checkmass(tmid)==1)

{

lmas.setText(tmid);

lmassf.setText(tmname);

break n; }

}

}

if(lmas.getText().length()==0)

{

flg=1;

JOptionPane.showMessageDialog(null,"Date is Not Available for Booking!!");

}

}

else

{

if(checkmass(lmas.getText())==1)

{

tmid=lmas.getText();

tmname=lmassf.getText();

}

else

{

flg=1;

JOptionPane.showMessageDialog(null,"Massager is not available for booking!!"); }

}

}

}

}

if(flg==0)

{

int i;

int diflg=0;

Statement st5=con.createStatement();

ResultSet res5;

String tstr="";

if(rtreat.isSelected())

{

for(i=0;i<tbl1.getRowCount();i++)

{

tstr="select \* from treatmenttbl where treat\_type='Weight Loss' and treat\_name='"+tbl1.getValueAt(i,1).toString()+"'";

res5=st5.executeQuery(tstr);

while(res5.next())

{

diflg=1;

}

}

}

if(diflg==0)

{

strdiet="";

strdname="";

}

ps.setString(1,tbdt.getText());

ps.setString(2,tid.getText());

if(rcust.isSelected())

strtype="Customer";

else

strtype="Member";

ps.setString(3,strtype);

ps.setString(4,tcid.getText());

ps.setString(5,tapdate.getText());

if(rtreat.isSelected())

strdis="Treatment";

else

strdis="Package";

ps.setString(6,strdis);

ps.setString(7,tpackid.getText());

ps.setString(8,toff.getText());

ps.setString(9,trtotal.getText());

ps.setString(10,ttotal.getText());

ps.setString(11,tmid);

ps.setString(12,tmname);

ps.setString(13,tstart.getText());

ps.setString(14,tend.getText());

ps.setString(15,"Not Paid");

ps.setString(16,strdiet);

ps.executeUpdate();

String str2="";

Statement st=con.createStatement();

for(i=0;i<tbl1.getRowCount();i++)

{

str2="insert into booktbl(bookid,treatid) values('"+tid.getText()+"','"+tbl1.getValueAt(i,0).toString()+"')";

st.executeUpdate(str2);

}

String str1="update keytbl set bookingid=bookingid+1";

Statement s1=con.createStatement();

s1.executeUpdate(str1);

str1="insert into scheduletbl(staffid,bookid,custmemid,apdate,starttime,endtime) values('"+lmas.getText()+"','"+tid.getText()+"','"+tcid.getText()+"','"+tapdate.getText()+"','"+tstart.getText()+"','"+tend.getText()+"')";

st.executeUpdate(str1);

JOptionPane.showMessageDialog(null,"Saved...");

bsave.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==bedit)

{

int flg=0,tflg=0;

bsave.setEnabled(false);

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

if(tid.getText().length()!=0&&sflg==1)

{

if(rcust.isSelected())

strtype="Customer";

else

strtype="Member";

if(rtreat.isSelected())

strdis="Treatment";

else

strdis="Package";

if(tid.getText().length()==0||tcid.getText().length()==0||bt.getSelection()==null||tapdate.getText().length()==0||bd.getSelection()==null||trtotal.getText().length()==0||ttotal.getText().length()==0)

{

flg=1;

JOptionPane.showMessageDialog(null,"Empty Fields");

}

if(flg==0)

{

int i,diflg=0;

Statement st5=con.createStatement();

ResultSet res5;

String tstr="";

if(rtreat.isSelected())

{

for(i=0;i<tbl1.getRowCount();i++)

{

tstr="select \* from treatmenttbl where treat\_type='Weight Loss' and treat\_name='"+tbl1.getValueAt(i,1).toString()+"'";

res5=st5.executeQuery(tstr);

while(res5.next())

{

diflg=1;

}

}

}

if(diflg==0)

{

strdiet="";

strdname="";

}

String str="update bookingtbl set usertype='"+strtype+"',cust\_mem\_id='"+tcid.getText()+"',appointdate='"+tapdate.getText()+"',disptype='"+strdis+"',packid='"+tpackid.getText()+"',discount='"+toff.getText()+"',realcost='"+trtotal.getText()+"',totalcost='"+ttotal.getText()+"',massagerid='"+lmas.getText()+"',massagername='"+lmass.getText()+"',starttime='"+tstart.getText()+"',endtime='"+tend.getText()+"',dietid='"+strdiet+"' where bookid='"+tid.getText()+"'";

st.executeUpdate(str);

st.executeUpdate("delete \* from booktbl where bookid='"+tid.getText()+"'");

String str2="";

for(i=0;i<tbl1.getRowCount();i++)

{

str2="insert into booktbl(bookid,treatid) values('"+tid.getText()+"','"+tbl1.getValueAt(i,0).toString()+"')";

st.executeUpdate(str2);

}

JOptionPane.showMessageDialog(null,"Updated");

bedit.setEnabled(false);

}

}

else

{

JOptionPane.showMessageDialog(null,"Search Record for update!!");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

}

public void focusLost(FocusEvent fe)

{

if(fe.getComponent()==toff)

{

if(toff.getText().length()!=0)

{

int s=Integer.parseInt(toff.getText());

double ttx=Double.parseDouble(ttax.getText());

if(s<=100)

{

Double tc;

tc=Double.parseDouble(trtotal.getText())-(Double.parseDouble(trtotal.getText())\*(Double.parseDouble(toff.getText())/100));

if(rmem.isSelected()==true)

{

double tv=tc-(tc\*(Double.parseDouble(tmdis)/100));

double m=tv+(tv\*(ttx/100));

ttotal.setText(""+m);

}

if(rcust.isSelected()==true)

{

double m=tc+(tc\*(ttx/100));

ttotal.setText(""+m);

}

}

else

{

JOptionPane.showMessageDialog(null,"Invalid Discount");

}

}

}

if(fe.getComponent()==tcid)

{

String t=tcid.getText();

if(tcid.getText().length()!=0)

{

try

{

if(rcust.isSelected()==true)

{

ttype.setText("");

ldis.setText("");

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from customertbl where custid='"+t+"'");

while(res.next())

{

tcid.setText(res.getString(2));

tname.setText(res.getString(3));

if(res.getString(7).equals("Male"))

rmale.setSelected(true);

else

rfemale.setSelected(true);

tmob.setText(res.getString(5));

temail.setText(res.getString(6));

}

ttype.setText("N/A");

}

if(rmem.isSelected()==true)

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from membertbl where mem\_id='"+tcid.getText()+"'");

while(res.next())

{

tcid.setText(res.getString(2));

tname.setText(res.getString(3));

if(res.getString(5).equals("Male"))

rmale.setSelected(true);

else

rfemale.setSelected(true);

tmob.setText(res.getString(8));

temail.setText(res.getString(9));

ttype.setText(res.getString(4));

tmdis=res.getString(15);

ldis.setText("Elegible for "+tmdis+" discount!!!");

}

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

}

if(fe.getComponent()==tapdate)

{

strchk=tapdate.getText();

if(tapdate.getText().length()!=0)

{

try

{

SimpleDateFormat cdf = new SimpleDateFormat("dd/MM/yyyy");

Date dat = cdf.parse(strchk);

if (!strchk.equals(cdf.format(dat)))

{

JOptionPane.showMessageDialog(null,"Invalid Date Format!!");

}

}

catch(Exception ee)

{

JOptionPane.showMessageDialog(null,"Invalid Date Format!!");

}

}

}

}

public static void main(String argv[])

throws IOException

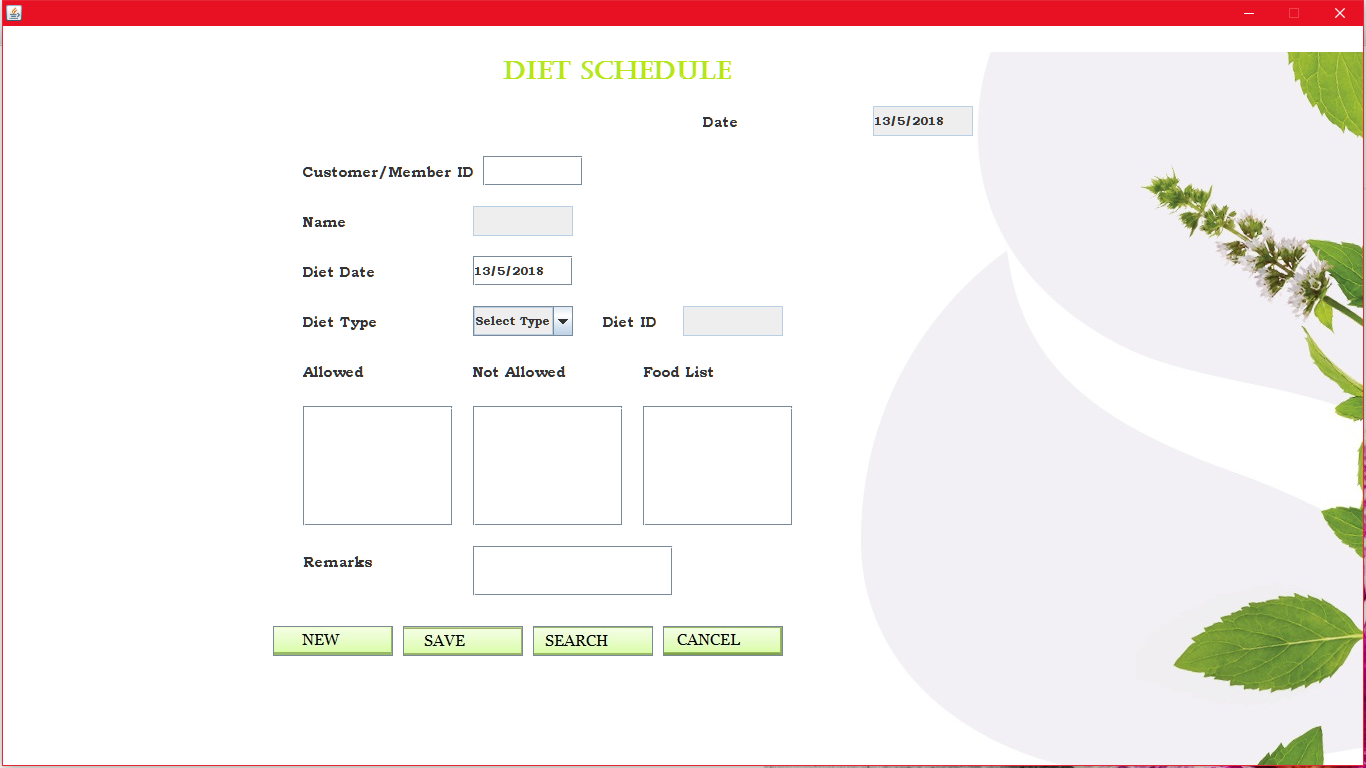
{

booking obj=new booking();

}

}

* **Diet Schedule reporting**: Diet scheduling report consists of all information about the scheduling of a particular diet so that it can be given to an employee or subscribed member.



import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.io.\*;

import java.sql.\*;

import java.util.\*;

public class diett extends JFrame implements ActionListener,FocusListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/trac.jpg"));

JLabel ddate=new JLabel("Date");

JLabel dcmid=new JLabel("Customer/Member ID");

JLabel dname=new JLabel("Name");

JLabel dddate=new JLabel("Diet Date");

JLabel dtype=new JLabel("Diet Type");

JLabel did=new JLabel("Diet ID");

JLabel dalw=new JLabel("Allowed");

JLabel dnalw=new JLabel("Not Allowed");

JLabel dlist=new JLabel("Food List");

JLabel drmk=new JLabel("Remarks");

JLabel dhead=new JLabel("Diet Schedule");

JTextField dtdate=new JTextField();

JTextField dtcmid=new JTextField();

JTextField dtname=new JTextField();

JTextField ddtdate=new JTextField();

JTextField dtid=new JTextField();

JTextArea dtalw=new JTextArea();

JTextArea dtnalw=new JTextArea();

JTextArea dtlist=new JTextArea();

JTextField dtrmk=new JTextField();

JComboBox dctype=new JComboBox();

JButton dnew=new JButton("New",new ImageIcon("../Images/nbt.jpg"));

JButton dsave=new JButton("Save",new ImageIcon("../Images/sbt.jpg"));

JButton dsearch=new JButton("Search",new ImageIcon("../Images/sbtn.jpg"));

JButton dcan=new JButton("Cancel",new ImageIcon("../Images/cbt.jpg"));

Calendar cal=Calendar.getInstance();

JScrollPane pnl=new JScrollPane();

JTable tbl;

String []col={"Si. no.","Diet Name","Diet Date","Remarks"};

String tsi,tname,tddt,trmk;

Object [][]data={{tsi,tname,tddt,trmk}};

DefaultTableModel model=new DefaultTableModel(col,0);

public diett()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

lbl.setBounds(0,0,getWidth(),getHeight());

dhead.setBounds(500,30,250,30);

ddate.setBounds(700,80,150,30);

dtdate.setBounds(870,80,100,30);

dcmid.setBounds(300,130,180,30);

dtcmid.setBounds(480,130,100,30);

dname.setBounds(300,180,150,30);

dtname.setBounds(470,180,100,30);

dddate.setBounds(300,230,150,30);

ddtdate.setBounds(470,230,100,30);

dtype.setBounds(300,280,150,30);

dctype.setBounds(470,280,100,30);

did.setBounds(600,280,100,30);

dtid.setBounds(680,280,100,30);

dalw.setBounds(300,330,150,30);

sralw.setBounds(300,380,150,120);

dnalw.setBounds(470,330,150,30);

srnalw.setBounds(470,380,150,120);

dlist.setBounds(640,330,150,30);

srlist.setBounds(640,380,150,120);

drmk.setBounds(300,520,150,30);

dtrmk.setBounds(470,520,200,50);

tbl=new JTable(data,col);

pnl=new JScrollPane(tbl);

add(pnl);

pnl.setBounds(800,150,500,250);

dnew.setBounds(270,600,120,30);

dsave.setBounds(400,600,120,30);

dsearch.setBounds(530,600,120,30);

dcan.setBounds(660,600,120,30);

dctype.addItem("Select Type");

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemDsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from diettbl");

while(res.next())

{

dctype.addItem(res.getString(2));

}

}

catch(Exception ee)

{

System.out.println(ee);

}

add(dnew);

add(dsave);

add(dsearch);

add(dcan);

add(drmk);

add(dtrmk);

add(dlist);

add(srlist);

add(dnalw);

add(srnalw);

add(dalw);

add(sralw);

add(dtype);

add(dctype);

add(dhead);

add(ddate);

add(dtdate);

add(dcmid);

add(dtcmid);

add(dname);

add(dtname);

add(dddate);

add(ddtdate);

add(did);

add(dtid);

dtname.setEditable(false);

dtdate.setEditable(false);

dtid.setEditable(false);

dtalw.setEditable(false);

dtnalw.setEditable(false);

dtlist.setEditable(false);

dtdate.setText(""+cal.get(Calendar.DATE)+"/"+(cal.get(Calendar.MONTH)+1)+"/"+cal.get(Calendar.YEAR));

ddtdate.setText(dtdate.getText());

pnl.setVisible(false);

dnew.addActionListener(this);

dsave.addActionListener(this);

dsearch.addActionListener(this);

dcan.addActionListener(this);

dtcmid.addFocusListener(this);

dctype.addActionListener(this);

dsave.setEnabled(false);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==dnew)

{

dtcmid.setText("");

ddtdate.setText(dtdate.getText());

dtname.setText("");

dctype.setSelectedIndex(0);

dtid.setText("");

dtalw.setText("");

dtnalw.setText("");

dtlist.setText("");

dtrmk.setText("");

pnl.setVisible(false);

dsave.setEnabled(true);

}

if(e.getSource()==dctype)

{

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemDsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("Select \* from diettbl where diettype='"+dctype.getSelectedItem()+"'");

while(res.next())

{

dtid.setText(res.getString(1));

dtalw.setText(res.getString(3));

dtnalw.setText(res.getString(4));

dtlist.setText(res.getString(5));

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==dsave)

{

int flg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemDsn");

String str="insert into dietstbl(cust\_mem\_id,dietid,diettype,dietdate,remarks) values(?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

if(dtcmid.getText().length()==0||ddtdate.getText().length()==0||did.getText().length()==0||dtalw.getText().length()==0||dtnalw.getText().length()==0||dtlist.getText().length()==0||dtrmk.getText().length()==0)

{

flg=1;

JOptionPane.showMessageDialog(null,"Empty Fields");

}

if(flg==0)

{

ps.setString(1,dtcmid.getText());

ps.setString(2,dtid.getText());

ps.setString(3,dctype.getSelectedItem().toString());

ps.setString(4,ddtdate.getText());

ps.setString(5,dtrmk.getText());

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Saved...");

dsave.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==dsearch)

{

int flg=0;

String tmp=JOptionPane.showInputDialog(null,"Enter Customer/Member ID: ");

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:Spasystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from dietstbl where cust\_mem\_id='"+tmp+"'");

Vector v;

pnl.setVisible(true);

while(res.next())

{

flg=1;

v=new Vector();

v.add(tbl.getRowCount()+1);

dtcmid.setText(tmp);

v.add(res.getString(3));

v.add(res.getString(4));

v.add(res.getString(5));

model.addRow(v);

}

if(flg==0)

{

JOptionPane.showMessageDialog(null,"No Record Found!!!");

dsave.setEnabled(false);

}

else

{

ResultSet res1=null;

if(tmp.charAt(0)=='M'||tmp.charAt(0)=='m')

{

res1=st.executeQuery("select \* from membertbl where mem\_id='"+tmp+"'");

}

if(tmp.charAt(0)=='C'||tmp.charAt(0)=='c')

{

res1=st.executeQuery("select \* from customertbl where custid='"+tmp+"'");

}

while(res1.next())

{

dtname.setText(res1.getString(3));

}

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==dcan)

{

this.dispose();

}

}

public void focusLost(FocusEvent fe)

{

if(fe.getComponent()==dtcmid)

{

if(dtcmid.getText().length()!=0)

{

String tmp=dtcmid.getText();

String ch=""+tmp.substring(0,1);

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemDsn");

ResultSet res=null;

Statement st=con.createStatement();

if(ch.equals("m")||ch.equals("M"))

res=st.executeQuery("select \* from membertbl where mem\_id='"+tmp+"'");

if(ch.equals("c")||ch.equals("C"))

res=st.executeQuery("select \* from customertbl where custid='"+tmp+"'");

while(res.next())

{

dtcmid.setText(res.getString(2));

dtname.setText(res.getString(3));

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

}

}

public static void main(String argv[])

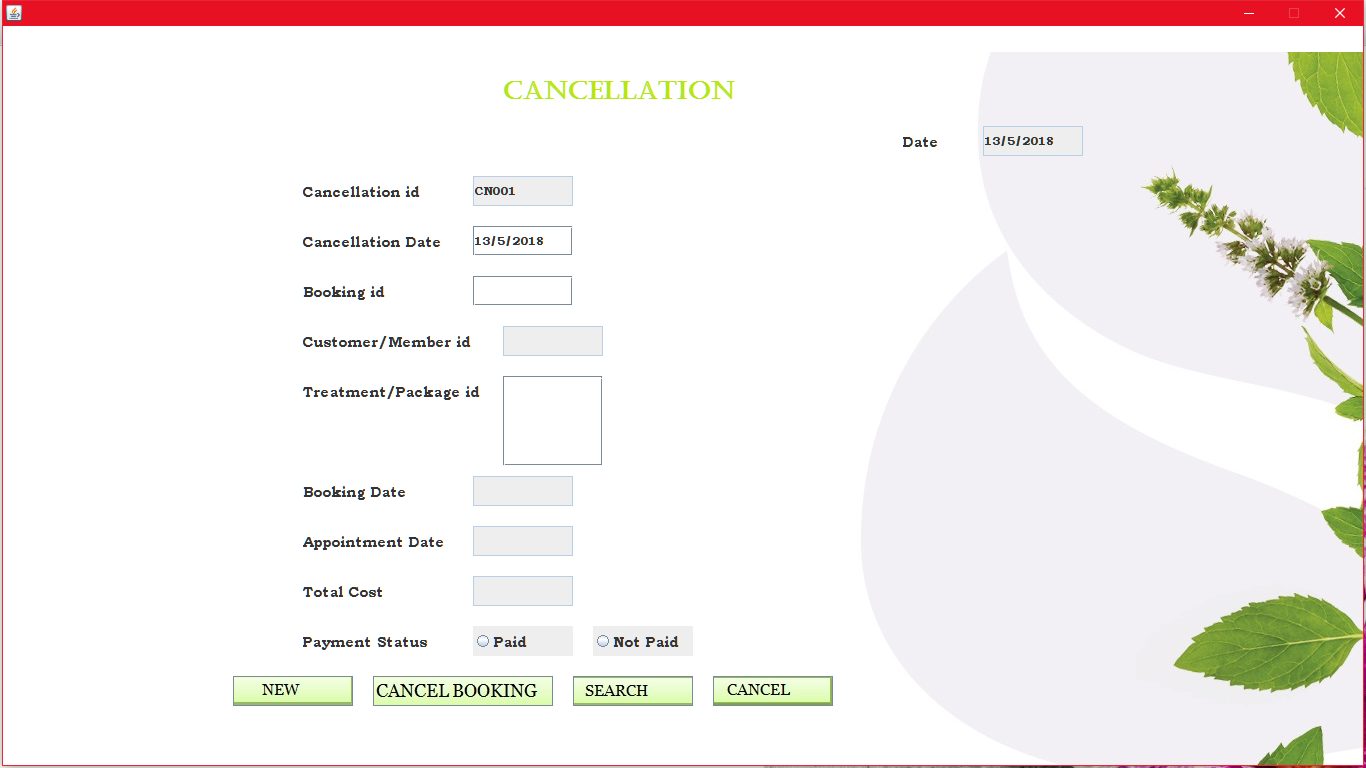
{

diett obj=new diett();

}

}

* **Cancel Booking:**Here the ordered or appointed treatments or packages can be cancelled.



import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.\*;

public class cancellation extends JFrame implements ActionListener,FocusListener,ItemListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/trac.jpg"));

long nday;

JLabel canid=new JLabel("Cancellation id");

JLabel cbookid=new JLabel("Booking id");

JLabel ccmid=new JLabel("Customer/Member id");

JLabel cptid=new JLabel("Treatment/Package id");

JLabel cbookdt=new JLabel("Booking Date");

JLabel capdate=new JLabel("Appointment Date");

JLabel ctcost=new JLabel("Total Cost");

JLabel cpayst=new JLabel("Payment Status");

JLabel cdate=new JLabel("Date");

JLabel lhead=new JLabel("CANCELLATION");

JLabel cdeduct=new JLabel("Deduction Percentage");

JLabel cdeductf=new JLabel("Deduction Fund");

JLabel crefund=new JLabel("Refund");

JLabel cnday=new JLabel("Cancel Days");

JLabel ccandt=new JLabel("Cancellation Date");

JLabel cpaid=new JLabel("Paid Money");

JTextField tcanid=new JTextField();

JTextField tbookid=new JTextField();

JTextField tcmid=new JTextField();

JTextArea tptid=new JTextArea();

JTextField tbookdt=new JTextField();

JTextField tapdate=new JTextField();

JTextField ttcost=new JTextField();

JTextField tdate=new JTextField();

JTextField tdeduct=new JTextField();

JTextField tdeductf=new JTextField();

JTextField trefund=new JTextField();

JTextField tnday=new JTextField();

JTextField tcandt=new JTextField();

JTextField tpaid=new JTextField();

JRadioButton rpay=new JRadioButton("Paid");

JRadioButton rnpay=new JRadioButton("Not Paid");

JButton cnew=new JButton("New",new ImageIcon("../Images/nbt.jpg"));

JButton ccanb=new JButton("Cancel Booking",new ImageIcon("../Images/cbbt.jpg"));

JButton csearch=new JButton("Search",new ImageIcon("../Images/sbtn.jpg"));

JButton ccancel=new JButton("Cancel",new ImageIcon("../Images/cbt.jpg"));

Calendar cal=Calendar.getInstance();

int cflg=0;

public cancellation()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

lhead.setBounds(500,50,300,30);

cdate.setBounds(900,100,100,30);

tdate.setBounds(980,100,100,30);

canid.setBounds(300,150,150,30);

tcanid.setBounds(470,150,100,30);

ccandt.setBounds(300,200,150,30);

tcandt.setBounds(470,200,100,30);

cbookid.setBounds(300,250,150,30);

tbookid.setBounds(470,250,100,30);

ccmid.setBounds(300,300,180,30);

tcmid.setBounds(500,300,100,30);

cptid.setBounds(300,350,180,30);

srptid.setBounds(500,350,100,90);

cbookdt.setBounds(300,450,150,30);

tbookdt.setBounds(470,450,100,30);

capdate.setBounds(300,500,150,30);

tapdate.setBounds(470,500,100,30);

ctcost.setBounds(300,550,150,30);

ttcost.setBounds(470,550,100,30);

cpayst.setBounds(300,600,150,30);

rpay.setBounds(470,600,100,30);

rnpay.setBounds(590,600,100,30);

cpaid.setBounds(700,150,150,30);

tpaid.setBounds(870,150,100,30);

cnday.setBounds(700,200,150,30);

tnday.setBounds(870,200,100,30);

cdeduct.setBounds(700,250,180,30);

tdeduct.setBounds(880,250,100,30);

cdeductf.setBounds(700,300,150,30);

tdeductf.setBounds(870,300,100,30);

crefund.setBounds(700,350,150,30);

trefund.setBounds(870,350,100,30);

add(cpaid);

add(tpaid);

add(ccandt);

add(tcandt);

add(crefund);

add(tdate);

add(canid);

add(tcanid);

add(cbookid);

add(tbookid);

add(ccmid);

add(tcmid);

add(cptid);

add(srptid);

add(cbookdt);

add(tbookdt);

add(capdate);

add(tapdate);

add(ctcost);

add(ttcost);

add(cpayst);

add(rpay);

add(rnpay);

bp.add(rpay);

bp.add(rnpay);

add(cnew);

add(ccanb);

add(csearch);

add(ccancel);

add(cnday);

add(tnday);

cnew.setBounds(230,650,120,30);

ccanb.setBounds(370,650,180,30);

csearch.setBounds(570,650,120,30);

ccancel.setBounds(710,650,120,30);

tdate.setText(""+cal.get(Calendar.DATE)+"/"+(cal.get(Calendar.MONTH)+1)+"/"+cal.get(Calendar.YEAR));

cnew.addActionListener(this);

ccanb.addActionListener(this);

csearch.addActionListener(this);

ccancel.addActionListener(this);

rpay.addActionListener(this);

rnpay.addActionListener(this);

tbookid.addFocusListener(this);

rpay.addItemListener(this);

rnpay.addItemListener(this);

cnday.setVisible(false);

tnday.setVisible(false);

cdeduct.setVisible(false);

tdeduct.setVisible(false);

tdeductf.setVisible(false);

cdeductf.setVisible(false);

crefund.setVisible(false);

trefund.setVisible(false);

cpaid.setVisible(false);

tpaid.setVisible(false);

ccanb.setEnabled(false);

tcanid.setEditable(false);

tcmid.setEditable(false);

tptid.setEditable(false);

tbookdt.setEditable(false);

tapdate.setEditable(false);

tdate.setEditable(false);

ttcost.setEditable(false);

tpaid.setEditable(false);

tnday.setEditable(false);

tdeduct.setEditable(false);

tdeductf.setEditable(false);

trefund.setEditable(false);

tcandt.addFocusListener(this);

}

public void display()

{

if(rpay.isSelected()==true)

{

cnday.setVisible(true);

tnday.setVisible(true);

cdeduct.setVisible(true);

tdeduct.setVisible(true);

tdeductf.setVisible(true);

cdeductf.setVisible(true);

crefund.setVisible(true);

trefund.setVisible(true);

cpaid.setVisible(true);

tpaid.setVisible(true);

}

else

{

cnday.setVisible(false);

tnday.setVisible(false);

cdeduct.setVisible(false);

tdeduct.setVisible(false);

tdeductf.setVisible(false);

cdeductf.setVisible(false);

crefund.setVisible(false);

trefund.setVisible(false);

cpaid.setVisible(false);

tpaid.setVisible(false);

}

}

public void itemStateChanged(ItemEvent ie)

{

if(ie.getSource()==rpay||ie.getSource()==rnpay)

{

if(ie.getStateChange()==ItemEvent.SELECTED)

{

display();

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==cnew)

{

cflg=0;

int ctr=0;

tcanid.setText("");

tbookid.setText("");

tcmid.setText("");

tptid.setText("");

tbookdt.setText("");

tapdate.setText("");

ttcost.setText("");

bp.clearSelection();

tdeduct.setText("");

tdeductf.setText("");

trefund.setText("");

tpaid.setText("");

display();

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

String str="select \* from keytbl";

ResultSet res=st.executeQuery(str);

while(res.next())

{

ctr=Integer.parseInt(res.getString(11));

}

if(ctr<10)

tcanid.setText("CN00"+ctr);

if(ctr<100&&ctr>9)

tcanid.setText("CN0"+ctr);

ccanb.setEnabled(true);

tcandt.setText(tdate.getText());

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==ccanb)

{

int flg=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="insert into canceltbl(canceldate,bookdate,canid,bookid,cust\_mem\_id,appointdate,treat\_pack\_id,totalcost,paystat,ndays,deductfund,refund,pyamt,staffid,stime,etime) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

if(tcanid.getText().length()==0||tcandt.getText().length()==0||tbookid.getText().length()==0||tcmid.getText().length()==0||tptid.getText().length()==0||tbookdt.getText().length()==0||tapdate.getText().length()==0||ttcost.getText().length()==0||bp.getSelection()==null)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!");

flg=1;

}

else

{

if(rpay.isSelected())

{

if(tpaid.getText().length()==0||tnday.getText().length()==0||tdeduct.getText().length()==0||tdeductf.getText().length()==0||trefund.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!");

flg=1;

}

else

{

if(cflg==1)

{

JOptionPane.showMessageDialog(null,"Date for Cancellation Expired!!");

flg=1;

} }

}

else

{

if(cflg==1)

{

JOptionPane.showMessageDialog(null,"Date for Cancellation Expired!!");

flg=1;

} }

}

if(flg==0)

{

ps.setString(1,tdate.getText());

ps.setString(2,tbookdt.getText());

ps.setString(3,tcanid.getText());

ps.setString(4,tbookid.getText());

ps.setString(5,tcmid.getText());

ps.setString(6,tapdate.getText());

ps.setString(7,tptid.getText());

ps.setString(8,ttcost.getText());

String tmp="";

if(rpay.isSelected()==true)

{

tmp="Paid";

ps.setString(9,tmp);

ps.setString(10,tnday.getText());

ps.setString(11,tdeductf.getText());

ps.setString(12,trefund.getText());

ps.setString(13,tpaid.getText());

}

else

{

tmp="Not Paid";

ps.setString(9,tmp);

}

String start="",end="",strs="";

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from bookingtbl where bookid='"+tbookid.getText()+"'");

while(res.next())

{

strs=res.getString(11);

start=res.getString(13);

end=res.getString(14);

}

res.close();

ps.setString(14,strs);

ps.setString(15,start);

ps.setString(16,end);

ps.executeUpdate();

st=con.createStatement();

str="update keytbl set cancelid=cancelid+1";

st.executeUpdate(str);

st=con.createStatement();

st.executeUpdate("delete \* from scheduletbl where staffid='"+strs+"' and bookid='"+tbookid.getText()+"'");

st=con.createStatement();

str="delete \* from bookingtbl where bookid='"+tbookid.getText()+"'";

st.executeUpdate(str);

if(tptid.getText().charAt(0)=='p'||tptid.getText().charAt(0)=='P')

else

{

str="delete \* from booktbl where bookid='"+tbookid.getText()+"'";

st.executeUpdate(str);

}

JOptionPane.showMessageDialog(null,"Booking Cancelled Successfully!!");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==csearch)

{

String tm="";

int flg=0;

ccanb.setEnabled(false);

try

{

tm=JOptionPane.showInputDialog("Enter Cancellation ID: ");

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

String str="select \* from canceltbl where canid='"+tm+"'";

ResultSet res=st.executeQuery(str);

String s="";

while(res.next())

{

flg=1;

tcandt.setText(res.getString(1));

tbookdt.setText(res.getString(2));

tbookid.setText(res.getString(4));

tcanid.setText(res.getString(3));

tcmid.setText(res.getString(5));

tapdate.setText(res.getString(6));

tptid.setText(res.getString(7));

ttcost.setText(res.getString(8));

s=res.getString(9);

if(s.equals("Paid"))

{

rpay.setSelected(true);

display();

}

else

{

rnpay.setSelected(true);

display();

}

tnday.setText(res.getString(10));

tdeductf.setText(res.getString(11));

trefund.setText(res.getString(12));

tpaid.setText(res.getString(13));

nday=Integer.parseInt(tnday.getText());

int n=0;

if(nday>=1 && nday<=10)

{

n=30;

tdeduct.setText("30%");

}

else if(nday>=11 && nday<=15)

{

n=20;

tdeduct.setText("20%");

}

else if(nday<=20)

{

n=15;

tdeduct.setText("15%");

}

else

{

n=5;

tdeduct.setText("5%");

}

}

if(flg==0)

{

JOptionPane.showMessageDialog(null,"Data Not Found");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==ccancel)

{

this.dispose();

}

}

public void focusLost(FocusEvent fe)

{

if(fe.getComponent()==tbookid)

{

int flg=0;

cflg=0;

String tmp="";

String tmpid="",tm="",ttr="",ttp;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

String str="select \* from bookingtbl where bookid='"+tbookid.getText()+"'";

ResultSet res=st.executeQuery(str);

while(res.next())

{

flg=1;

tbookid.setText(res.getString(2));

tcmid.setText(res.getString(4));

tmp=res.getString(6);

tmpid=res.getString(7);

tbookdt.setText(res.getString(1));

tapdate.setText(res.getString(5));

ttcost.setText(res.getString(10));

tm=res.getString(15);

if(tm.equals("Paid"))

rpay.setSelected(true);

else

rnpay.setSelected(true);

}

if(flg==1)

{

Double pamt=0.0;

ResultSet res1=st.executeQuery("select \* from paymenttbl where bookid='"+tbookid.getText()+"'");

while(res1.next())

{

pamt=pamt+Double.parseDouble(res1.getString(5));

}

tpaid.setText(""+pamt);

String stt="";

if(tmp.equals("Treatment"))

{

res1=st.executeQuery("select \* from booktbl where bookid='"+tbookid.getText()+"'" );

while(res1.next())

{

stt=stt+res1.getString(2)+"\n";

}

tptid.setText(stt);

}

if(tmp.equals("Package"))

tptid.setText(tmpid);

SimpleDateFormat formatter=new SimpleDateFormat("dd/MM/yyyy");

Date ndate=formatter.parse(tdate.getText());

Date napdate=formatter.parse(tapdate.getText());

long x=napdate.getTime()-ndate.getTime();

nday=x/86400000;

if(nday>0)

{

tnday.setText(""+nday);

int n=0;

if(nday>=1 && nday<=10)

{

n=30;

tdeduct.setText("30%");

}

else if(nday>=11 && nday<=15)

{

n=20;

tdeduct.setText("20%");

}

else if(nday<=20)

{

n=15;

tdeduct.setText("15%");

}

else

{

n=5;

tdeduct.setText("5%");

}

Double db=Double.parseDouble(tpaid.getText())\*((double)n/100);

tdeductf.setText(""+db);

trefund.setText(""+(Double.parseDouble(tpaid.getText())-db));

}

else

{

cflg=1;

}}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(fe.getComponent()==tcandt)

{

strchk=tcandt.getText();

if(tcandt.getText().length()!=0)

{

try

{

SimpleDateFormat cdf = new SimpleDateFormat("dd/MM/yyyy");

Date dat = cdf.parse(strchk);

if (!strchk.equals(cdf.format(dat)))

{

JOptionPane.showMessageDialog(null,"Invalid Date Format!!");

tcandt.setText("");

}}

catch(Exception ee)

{

JOptionPane.showMessageDialog(null,"Invalid Date Format!!"); }

}} }

public static void main(String argv[])

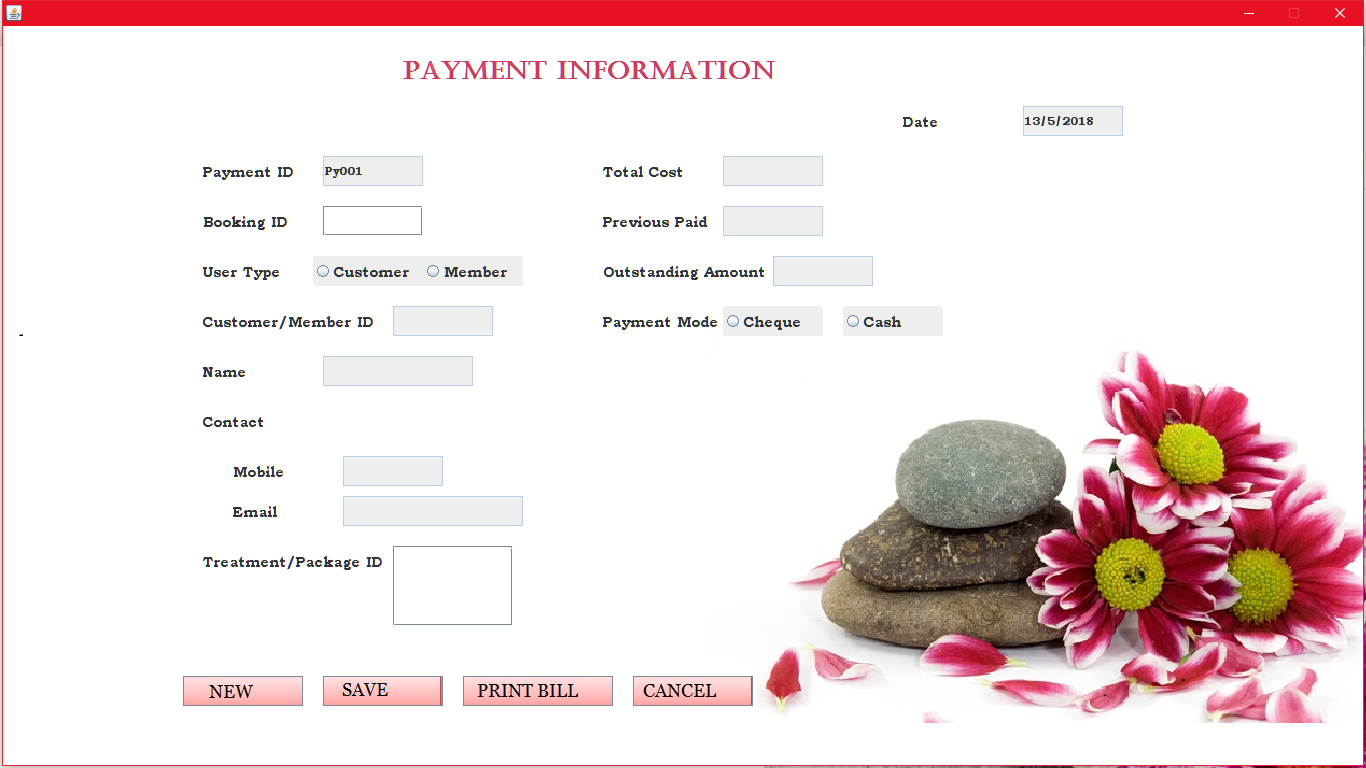
throws IOException{

cancellation obj=new cancellation();

}

}

* **Payment:**Here the payment of treatments and packages are paid and recorded. Payment can be done by Cash or Cheque.



import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.\*;

public class payment extends JFrame implements ActionListener,FocusListener,ItemListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/pt.jpg"));

JLabel pid=new JLabel("Payment ID");

JLabel pbid=new JLabel("Booking ID");

JLabel puser=new JLabel("User Type");

JLabel pcmid=new JLabel("Customer/Member ID");

JLabel pname=new JLabel("Name");

JLabel pcont=new JLabel("Contact");

JLabel pmob=new JLabel("Mobile");

JLabel pemail=new JLabel("Email");

JLabel ptcost=new JLabel("Total Cost");

JLabel pad=new JLabel("Previous Paid");

JLabel pout=new JLabel("Outstanding Amount");

JLabel paymode=new JLabel("Payment Mode");

JLabel pbr=new JLabel("Branch Name");

JLabel pch=new JLabel("Cheque No");

JLabel pchdt=new JLabel("Cheque Date");

JLabel pamt=new JLabel("Amount");

JLabel phead=new JLabel("Payment Information");

JLabel pdate=new JLabel("Date");

JLabel ptreat=new JLabel("Treatment/Package ID");

JLabel pbank=new JLabel("Bank Name");

JLabel pdt=new JLabel("Payment Date");

JTextField ptdate=new JTextField();

JTextArea pttreat=new JTextArea();

JTextField ptid=new JTextField();

JTextField ptbid=new JTextField();

JTextField ptcmid=new JTextField();

JTextField ptname=new JTextField();

JTextField ptcont=new JTextField();

JTextField ptmob=new JTextField();

JTextField ptemail=new JTextField();

JTextField ptc=new JTextField();

JTextField ptad=new JTextField();

JTextField ptmode=new JTextField();

JTextField ptbr=new JTextField();

JTextField ptch=new JTextField();

JTextField ptchdt=new JTextField();

JTextField ptamt=new JTextField();

JTextField ptout=new JTextField();

JTextField ptbank=new JTextField();

JTextField ptdt=new JTextField();

JRadioButton rmem=new JRadioButton("Member");

JRadioButton rcust=new JRadioButton("Customer");

JRadioButton rch=new JRadioButton("Cheque");

JRadioButton rcash=new JRadioButton("Cash");

JButton pnew=new JButton("New",new ImageIcon("../Images/pnbt.jpg"));

JButton psave=new JButton("Save",new ImageIcon("../Images/psbt.png"));

JButton pbill=new JButton("Print Bill",new ImageIcon("../Images/ppbt.jpg"));

JButton pcancel=new JButton("Cancel",new ImageIcon("../Images/pcbt.jpg"));

Calendar cal=Calendar.getInstance();

int sflg=0;

public payment()

{

setLayout(null);

setVisible(true);

this.getContentPane().setBackground(Color.WHITE);

phead.setBounds(400,30,380,30);

pdate.setBounds(900,80,100,30);

ptdate.setBounds(1020,80,100,30);

pid.setBounds(200,130,100,30);

ptid.setBounds(320,130,100,30);

pbid.setBounds(200,180,100,30);

ptbid.setBounds(320,180,100,30);

puser.setBounds(200,230,100,30);

rcust.setBounds(310,230,110,30);

rmem.setBounds(420,230,100,30);

pcmid.setBounds(200,280,190,30);

ptcmid.setBounds(390,280,100,30);

pname.setBounds(200,330,100,30);

ptname.setBounds(320,330,150,30);

pcont.setBounds(200,380,100,30);

pmob.setBounds(230,430,100,30);

ptmob.setBounds(340,430,100,30);

pemail.setBounds(230,470,100,30);

ptemail.setBounds(340,470,180,30);

ptreat.setBounds(200,520,180,30);

srptid.setBounds(390,520,120,80);

ptcost.setBounds(600,130,100,30);

ptc.setBounds(720,130,100,30);

pad.setBounds(600,180,120,30);

ptad.setBounds(720,180,100,30);

pout.setBounds(600,230,170,30);

ptout.setBounds(770,230,100,30);

paymode.setBounds(600,280,120,30);

rch.setBounds(720,280,100,30);

rcash.setBounds(840,280,100,30);

pamt.setBounds(600,320,100,30);

ptamt.setBounds(720,320,100,30);

pbank.setBounds(600,370,100,30);

ptbank.setBounds(720,370,120,30);

pbr.setBounds(600,420,120,30);

ptbr.setBounds(720,420,150,30);

pch.setBounds(600,470,100,30);

ptch.setBounds(720,470,120,30);

pchdt.setBounds(600,520,120,30);

ptchdt.setBounds(720,520,100,30);

pdt.setBounds(600,570,120,30);

ptdt.setBounds(720,570,100,30);

pnew.setBounds(180,650,120,30);

psave.setBounds(320,650,120,30);

pbill.setBounds(460,650,150,30);

pcancel.setBounds(630,650,120,30);

add(pdt);

add(ptdt);

bt.add(rcust);

bt.add(rmem);

bm.add(rch);

bm.add(rcash);

add(ptbid);

add(pbank);

add(ptbank);

add(pbill);

add(pbid);

add(pamt);

add(ptreat);

add(srptid);

add(ptamt);

add(pchdt);

add(ptchdt);

add(pch);

add(ptch);

add(pbr);

add(ptbr);

add(paymode);

add(rch);

add(rcash);

add(pout);

add(ptout);

add(ptcost);

add(pad);

add(ptad);

add(pnew);

add(psave);

add(pcancel);

add(ptc);

add(pcont);

add(pmob);

add(ptmob);

add(pemail);

add(ptemail);

add(pcmid);

add(ptcmid);

add(pname);

add(ptname);

add(puser);

add(rcust);

add(rmem);

add(pid);

add(ptid);

add(ptdate);

add(pdate);

add(phead);

pnew.addActionListener(this);

psave.addActionListener(this);

pcancel.addActionListener(this);

pbill.addActionListener(this);

rch.addItemListener(this);

rcash.addItemListener(this);

ptbid.addFocusListener(this);

ptad.addFocusListener(this);

ptamt.addFocusListener(this);

pbr.setVisible(false);

ptbr.setVisible(false);

pamt.setVisible(false);

ptamt.setVisible(false);

pch.setVisible(false);

ptch.setVisible(false);

ptchdt.setVisible(false);

pchdt.setVisible(false);

pbank.setVisible(false);

ptbank.setVisible(false);

ptdt.setVisible(false);

pdt.setVisible(false);

ptdate.setText(""+cal.get(Calendar.DATE)+"/"+(cal.get(Calendar.MONTH)+1)+"/"+cal.get(Calendar.YEAR));

ptid.setEditable(false);

ptcmid.setEditable(false);

ptname.setEditable(false);

ptmob.setEditable(false);

ptemail.setEditable(false);

pttreat.setEditable(false);

ptc.setEditable(false);

ptad.setEditable(false);

ptout.setEditable(false);

ptdate.setEditable(false);

psave.setEnabled(false);

ptdt.addFocusListener(this);

ptchdt.addFocusListener(this);

}

public void cheque()

{

if(rch.isSelected()==true)

{

pbr.setVisible(true);

ptbr.setVisible(true);

pamt.setVisible(true);

ptamt.setVisible(true);

pch.setVisible(true);

ptch.setVisible(true);

ptchdt.setVisible(true);

pchdt.setVisible(true);

pbank.setVisible(true);

ptbank.setVisible(true);

ptdt.setVisible(true);

pdt.setVisible(true);

}

else

{

pbr.setVisible(false);

ptbr.setVisible(false);

pamt.setVisible(true);

ptamt.setVisible(true);

pch.setVisible(false);

ptch.setVisible(false);

ptchdt.setVisible(false);

pchdt.setVisible(false);

pbank.setVisible(false);

ptbank.setVisible(false);

ptdt.setVisible(false);

pdt.setVisible(false);

}

}

public void cash()

{

if(rcash.isSelected()==true)

{

pamt.setVisible(true);

ptamt.setVisible(true);

pbr.setVisible(false);

ptbr.setVisible(false);

pch.setVisible(false);

ptch.setVisible(false);

ptchdt.setVisible(false);

pchdt.setVisible(false);

pbank.setVisible(false);

ptbank.setVisible(false);

ptdt.setVisible(false);

pdt.setVisible(false);

}

else

{

pbr.setVisible(true);

ptbr.setVisible(true);

pamt.setVisible(true);

ptamt.setVisible(true);

pch.setVisible(true);

ptch.setVisible(true);

ptchdt.setVisible(true);

pchdt.setVisible(true);

ptbank.setVisible(true);

pbank.setVisible(true);

ptdt.setVisible(true);

pdt.setVisible(true);

}

}

public void itemStateChanged(ItemEvent ie)

{

if(ie.getSource()==rch||ie.getSource()==rcash)

{

if(ie.getStateChange()==ItemEvent.SELECTED)

{

cheque();

cash();

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==pnew)

{

int ctr=0;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("Select \* from keytbl");

while(res.next())

{

ctr=Integer.parseInt(res.getString(12));

}

if(ctr<10)

ptid.setText("Py00"+ctr);

if(ctr>9&&ctr<100)

ptid.setText("Py0"+ctr);

bt.clearSelection();

ptcmid.setText("");

ptname.setText("");

ptmob.setText("");

ptemail.setText("");

pttreat.setText("");

ptc.setText("");

ptad.setText("");

ptout.setText("");

bm.clearSelection();

ptbid.setText("");

ptamt.setText("");

ptbank.setText("");

ptbr.setText("");

ptch.setText("");

ptchdt.setText("");

ptdt.setText(ptdate.getText());

psave.setEnabled(true);

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==psave)

{

int flg=0;

String tam="";

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="insert into paymenttbl(payid,bookid,cust\_mem\_id,treatid,pamount,tamount,dop,chequeno,chequedt,bankname,branch)values(?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

if(ptid.getText().length()==0||ptbid.getText().length()==0||bt.getSelection()==null||ptcmid.getText().length()==0||ptname.getText().length()==0||ptmob.getText().length()==0||ptemail.getText().length()==0||pttreat.getText().length()==0||ptdt.getText().length()==0||ptc.getText().length()==0||bm.getSelection()==null||ptamt.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

else

{

if(rmem.isSelected())

{

if(ptad.getText().length()==0||ptout.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1;

}

}

if(rch.isSelected())

{

if(ptbank.getText().length()==0||ptbr.getText().length()==0||ptch.getText().length()==0||ptchdt.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Fields!!!");

flg=1; }

}

try

{

double ad=Double.parseDouble(ptc.getText());

double m=Double.parseDouble(ptamt.getText());

if(rmem.isSelected()==true)

{

double t=Double.parseDouble(ptout.getText());

if(m<=0)

{

throw new Exception("InvalidAmt"); }

if(m>t)

{

flg=1;

JOptionPane.showMessageDialog(null,"Amount Exceeds!!!");

ptamt.setText("");

}

}

if(rcust.isSelected()==true)

{

if(m<=0)

throw new Exception("InvalidAmt"); if(m>ad)

{

flg=1; JOptionPane.showMessageDialog(null,"Amount Exceeds!!");

ptamt.setText("");

}

else

{

if(Double.parseDouble(ptc.getText())==Double.parseDouble(ptamt.getText()))

else

throw new Exception("NotValidAmt");

}

}

}

catch(Exception ee)

{

flg=1; if(ee.getMessage().equals("InvalidAmt"))

JOptionPane.showMessageDialog(null,"Invalid Amount Entered!!");

if(ee.getMessage().equals("NotValidAmt"))

JOptionPane.showMessageDialog(null,"Not Eligible for Half Payment!!");

}

}

if(flg==0)

{

ps.setString(1,ptid.getText());

ps.setString(2,ptbid.getText());

ps.setString(3,ptcmid.getText());

ps.setString(4,pttreat.getText());

ps.setString(5,ptamt.getText());

if(rcust.isSelected())

tam=""+Double.parseDouble(ptamt.getText());

else

tam=""+(Double.parseDouble(ptad.getText())+Double.parseDouble(ptamt.getText()));

ps.setString(6,tam);

ps.setString(7,ptdt.getText());

ps.setString(8,ptch.getText());

ps.setString(9,ptchdt.getText());

ps.setString(10,ptbank.getText());

ps.setString(11,ptbr.getText());

ps.executeUpdate();

Statement st=con.createStatement();

String str1="update keytbl set paymentid=paymentid+1";

st.executeUpdate(str1);

st.executeUpdate("update bookingtbl set pystate='Paid' where bookid='"+ptbid.getText()+"'");

JOptionPane.showMessageDialog(null,"Saved..");

psave.setEnabled(false);

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==pcancel)

{

this.dispose();

}

if(e.getSource()==pbill)

{

int t=0;

String tt="",t1="";

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from keytbl");

while(res.next())

{

t=Integer.parseInt(res.getString(12))-1;

}

res.close();

p:{

while(t>1)

{

if(t<10)

tt="Py00"+t;

if(t>=10&&t<100)

tt="Py0"+t;

res=st.executeQuery("select \* from paymenttbl"); while(res.next())

{

if(tt.equals(res.getString(1)))

{

t1=res.getString(2);

break p;

}

}

t--;

}

}

prbill probj=new prbill(t1);

}

catch(Exception ee)

{

System.out.println(ee);

}

}

}

public void focusLost(FocusEvent fe)

{

if(fe.getComponent()==ptbid)

{

int flg=0;

String str1="";

String str2="";

try

{

if(ptbid.getText().length()!=0)

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st= con.createStatement();

ResultSet res=st.executeQuery("select \* from bookingtbl where bookid='"+ptbid.getText()+"'");

while(res.next())

{

flg=1;

if(res.getString(3).equals("Customer"))

rcust.setSelected(true);

else

rmem.setSelected(true);

ptcmid.setText(res.getString(4));

str1=res.getString(6);

str2=res.getString(7);

ptc.setText(res.getString(10));

}

if(rcust.isSelected()==true)

{

pad.setVisible(false);

ptad.setVisible(false);

ptout.setVisible(false);

pout.setVisible(false);

ResultSet res1=st.executeQuery("select \* from customertbl where custid='"+ptcmid.getText()+"'");

while(res1.next())

{

ptname.setText(res1.getString(3));

ptmob.setText(res1.getString(5));

ptemail.setText(res1.getString(6));

}

}

if(rmem.isSelected()==true)

{

ptad.setVisible(true);

ptout.setVisible(true);

pad.setVisible(true);

pout.setVisible(true);

ResultSet res2=st.executeQuery("select \* from membertbl where mem\_id='"+ptcmid.getText()+"'");

while(res2.next())

{

ptname.setText(res2.getString(3));

ptmob.setText(res2.getString(8));

ptemail.setText(res2.getString(9));

}

String tp="";

res=st.executeQuery("select \* from paymenttbl where bookid='"+ptbid.getText()+"'");

while(res.next())

{

tp=res.getString(6);

}

ptad.setText(tp);

if(ptad.getText().length()==0)

ptad.setText("0.0");

double ad=Double.parseDouble(ptad.getText());

double ts=Double.parseDouble(ptc.getText());

double t=ts-ad;

ptout.setText(""+t);

}

if(flg==0)

{

JOptionPane.showMessageDialog(null,"Invalid Booking ID..");

}

else

{

if(str1.charAt(0)=='p'||str1.charAt(0)=='P')

{

pttreat.setText(str2);

}

if(str1.charAt(0)=='T'||str1.charAt(0)=='t')

{

String stt="";

ResultSet res1=st.executeQuery("select \* from booktbl where bookid='"+ptbid.getText()+"'" );

while(res1.next())

{

stt=stt+res1.getString(2)+"\n";

}

pttreat.setText(stt);

}

}

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(fe.getComponent()==ptad)

{

if(ptad.getText().length()==0)

else

{

double ad=Double.parseDouble(ptad.getText());

double ts=Double.parseDouble(ptc.getText());

double t=ts-ad;

ptout.setText(""+t);

}

}

if(fe.getComponent()==ptamt)

{

if(ptamt.getText().length()==0)

else

{

try

{

double ad=Double.parseDouble(ptc.getText());

double m=Double.parseDouble(ptamt.getText());

if(rmem.isSelected()==true)

{

double t=Double.parseDouble(ptout.getText());

if(m<=0)

throw new Exception("InvalidAmt");

if(m>t)

JOptionPane.showMessageDialog(null,"Amount Exceeds!!!");

}

if(rcust.isSelected()==true)

{

if(m<=0)

throw new Exception("InvalidAmt");

if(m>ad)

JOptionPane.showMessageDialog(null,"Amount Exceeds!!");

else

{

if(Double.parseDouble(ptc.getText())==Double.parseDouble(ptamt.getText()))

else

throw new Exception("NotValidAmt");

}

}

}

catch(Exception ee)

{

if(ee.getMessage().equals("InvalidAmt"))

JOptionPane.showMessageDialog(null,"Invalid Amount Entered!!");

if(ee.getMessage().equals("NotValidAmt"))

JOptionPane.showMessageDialog(null,"Not Eligible for Half Payment!!");

}

}

}

}

}

public void focusGained(FocusEvent fe)

{

}

public static void main(String argv[])

throws IOException

{

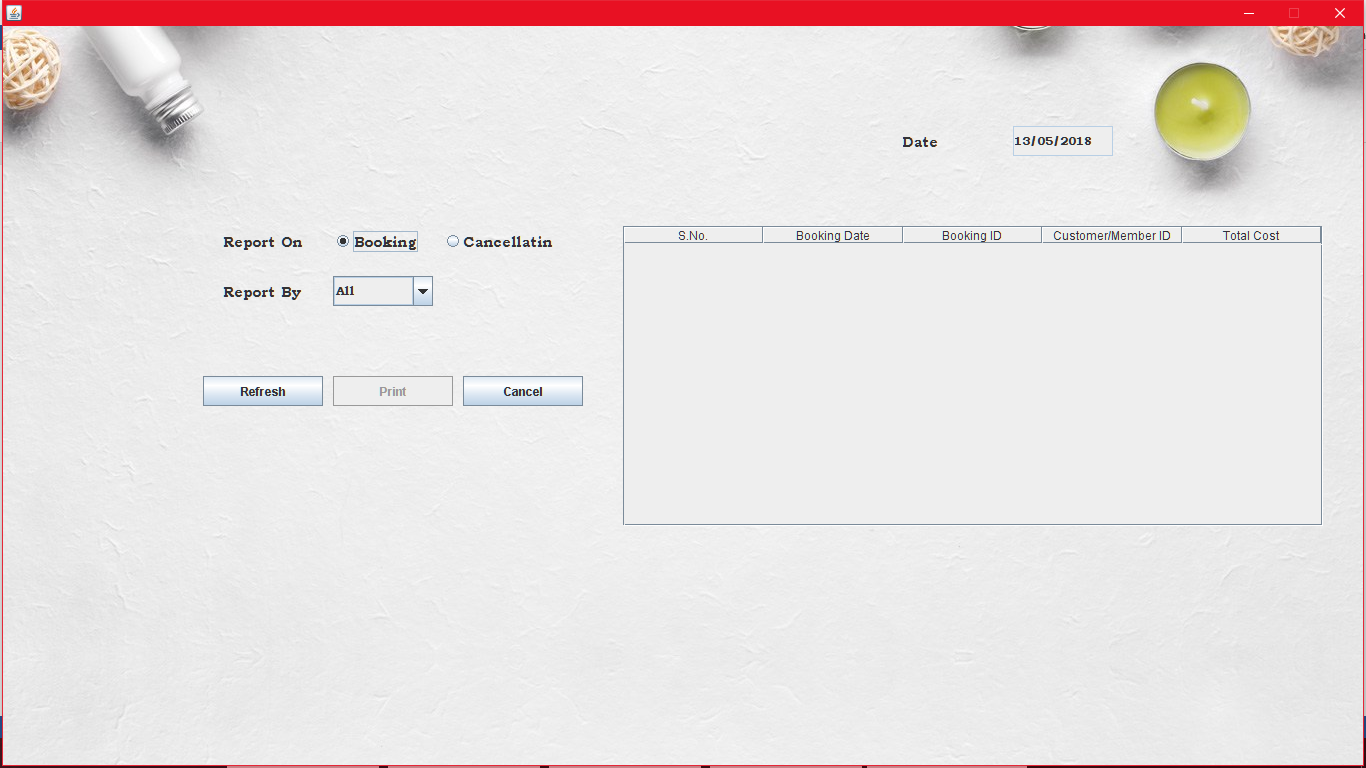
payment obj=new payment();

}

}

**Reporting**

* B**ooking report**:Daily booking report consists of information about how many booking have been done on that particular day.
* **Cancellation report:**Daily cancellation report consists information about how many cancellation have been done on that particular day.



import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.io.\*;

import java.sql.\*;

import java.util.\*;

public class report extends JFrame implements ActionListener,ItemListener,FocusListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/bg1.jpeg"));

JLabel ltype=new JLabel("Report On");

JLabel lsearch=new JLabel("Report");

JLabel lsby=new JLabel("Report By");

JLabel ldate=new JLabel("Date");

JTextField tsearch=new JTextField();

JTextField tdate=new JTextField();

JRadioButton rpbook=new JRadioButton("Booking");

JRadioButton rpcan=new JRadioButton("Cancellatin");

JComboBox mtype=new JComboBox();

JComboBox month=new JComboBox();

JButton brefresh=new JButton("Refresh");

JButton bprint=new JButton("Print");

JButton bcan=new JButton("Cancel");

JTable tbl;

String []col={"S.No.","Booking Date","Booking ID","Customer/Member ID","Total Cost"};

String bno,bdt,bid,bcmid,bcost;

Object[][]data={{bno,bdt,bid,bcmid,bcost}};

DefaultTableModel model=new DefaultTableModel(col,0);

JTable tbl1;

String []col1={"S.No.","Cancellation Date","Cancellation ID","Booking ID","Customer/Member ID","Total Cost"};

String cno,cdt,cid,cbid,ccmid,ccost;

Object[][]data1={{cno,cdt,cid,cbid,ccmid,ccost}};

DefaultTableModel model1=new DefaultTableModel(col1,0);

JScrollPane bpnl=new JScrollPane();

JScrollPane cpnl=new JScrollPane();

Calendar cal=Calendar.getInstance();

int flg,ctr;

public report()

{

flg=0;

setLayout(null);

setVisible(true);

getContentPane().setBackground(Color.WHITE);

ldate.setBounds(900,100,100,30);

tdate.setBounds(1010,100,100,30);

ltype.setBounds(220,200,100,30);

rpbook.setBounds(330,200,100,30);

rpcan.setBounds(440,200,120,30);

lsby.setBounds(220,250,100,30);

mtype.setBounds(330,250,100,30);

lsearch.setBounds(220,300,150,30);

tsearch.setBounds(380,300,180,30);

month.setBounds(380,300,180,30);

brefresh.setBounds(200,350,120,30);

bprint.setBounds(330,350,120,30);

bcan.setBounds(460,350,120,30);

add(ldate);

add(tdate);

add(ltype);

add(rpbook);

add(rpcan);

bg.add(rpbook);

bg.add(rpcan);

add(lsby);

add(mtype);

mtype.addItem("All");

mtype.addItem("Today");

mtype.addItem("Monthly");

mtype.addItem("Date");

add(month);

month.addItem("January");

month.addItem("Feburary");

month.addItem("March");

month.addItem("April");

month.addItem("May");

month.addItem("June");

month.addItem("July");

month.addItem("August");

month.addItem("September");

month.addItem("October");

month.addItem("November");

month.addItem("December");

add(lsearch);

add(tsearch);

add(brefresh);

add(bprint);

add(bcan);

tbl=new JTable(data,col);

bpnl=new JScrollPane(tbl);

bpnl.setBounds(620,200,700,300);

add(bpnl);

tbl1=new JTable(data1,col1);

cpnl=new JScrollPane(tbl1);

cpnl.setBounds(620,200,700,300);

add(cpnl);

String sm="",sd="";

if((cal.get(Calendar.DATE))<10)

sd="0"+cal.get(Calendar.DATE);

else

sd=""+cal.get(Calendar.DATE);

if((cal.get(Calendar.MONTH)+1)<10)

sm="0"+(cal.get(Calendar.MONTH)+1);

else

sm=""+(cal.get(Calendar.MONTH)+1);

tdate.setText(""+sd+"/"+sm+"/"+cal.get(Calendar.YEAR));

tdate.setEditable(false);

bpnl.setVisible(false);

cpnl.setVisible(false);

lsby.setVisible(false);

mtype.setVisible(false);

lsearch.setVisible(false);

tsearch.setVisible(false);

month.setVisible(false);

bprint.setEnabled(false);

add(lbl);

rpbook.addItemListener(this);

rpcan.addItemListener(this);

brefresh.addActionListener(this);

bprint.addActionListener(this);

bcan.addActionListener(this);

mtype.addActionListener(this);

tsearch.addFocusListener(this);

}

public void display()

{

if(rpbook.isSelected()||rpcan.isSelected())

{

mtype.setSelectedIndex(0);

lsby.setVisible(true);

mtype.setVisible(true);

if(rpbook.isSelected())

{

flg=1;

bpnl.setVisible(true);

model.setRowCount(0);

cpnl.setVisible(false);

}

else

{

flg=2;

bpnl.setVisible(false);

cpnl.setVisible(true);

model1.setRowCount(0);

}

}

if(rpbook.isSelected()==false&&rpcan.isSelected()==false)

{

flg=0;

lsby.setVisible(false);

mtype.setVisible(false);

cpnl.setVisible(false);

bpnl.setVisible(false);

}

}

public void bsearch()

{

String str="";

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

Vector v=null;

ResultSet res=null;

if(mtype.getSelectedItem().toString().equals("All"))

str="select \* from bookingtbl";

if(mtype.getSelectedItem().toString().equals("Today"))

str="select \* from bookingtbl where bookdate='"+tdate.getText()+"'";

if(str.length()!=0)

{

res=st.executeQuery(str);

while(res.next())

{

v=new Vector();

v.add(tbl.getRowCount()+1);

v.add(res.getString(1));

v.add(res.getString(2));

v.add(res.getString(4));

v.add(res.getString(10));

model.addRow(v);

}

}

if(mtype.getSelectedItem().toString().equals("Monthly"))

{

String t="";

SimpleDateFormat formatter=new SimpleDateFormat("dd/MM/yyyy");

Date sdate=formatter.parse("1/"+(month.getSelectedIndex()+1)+"/"+tdate.getText().substring(6,tdate.getText().length()));

Date edate=formatter.parse("31/"+(month.getSelectedIndex()+1)+"/"+tdate.getText().substring(6,tdate.getText().length()));

Date tdt;

res=st.executeQuery("select \* from bookingtbl");

while(res.next())

{

t=res.getString(1);

tdt=formatter.parse(t);

if(tdt.after(sdate)&&tdt.before(edate)||tdt.equals(sdate)||tdt.equals(edate))

{

v=new Vector();

v.add(tbl.getRowCount()+1);

v.add(t);

v.add(res.getString(2));

v.add(res.getString(4));

v.add(res.getString(10));

model.addRow(v);

}

}

}

if(mtype.getSelectedItem().toString().equals("Date"))

{

if(tsearch.getText().length()!=0)

{

String t="";

SimpleDateFormat formatter=new SimpleDateFormat("dd/MM/yyyy");

Date cdt=formatter.parse(tsearch.getText());

Date tdt;

res=st.executeQuery("select \* from bookingtbl");

while(res.next())

{

t=res.getString(1);

tdt=formatter.parse(t);

if(tdt.equals(cdt))

{

v=new Vector();

v.add(tbl.getRowCount()+1);

v.add(t);

v.add(res.getString(2));

v.add(res.getString(4));

v.add(res.getString(10));

model.addRow(v);

}

}

}

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

public void csearch()

{

String str="";

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

Vector v=null;

ResultSet res=null;

if(mtype.getSelectedItem().toString().equals("All"))

str="select \* from canceltbl";

if(mtype.getSelectedItem().toString().equals("Today"))

str="select \* from canceltbl where canceldate='"+tdate.getText()+"'";

if(str.length()!=0)

{

res=st.executeQuery(str);

while(res.next())

{

v=new Vector();

v.add(tbl1.getRowCount()+1);

v.add(res.getString(1));

v.add(res.getString(3));

v.add(res.getString(4));

v.add(res.getString(5));

v.add(res.getString(8));

model1.addRow(v);

}

}

if(mtype.getSelectedItem().toString().equals("Monthly"))

{

String t="";

SimpleDateFormat formatter=new SimpleDateFormat("dd/MM/yyyy");

Date sdate=formatter.parse("1/"+(month.getSelectedIndex()+1)+"/"+tdate.getText().substring(6,tdate.getText().length()));

Date edate=formatter.parse("31/"+(month.getSelectedIndex()+1)+"/"+tdate.getText().substring(6,tdate.getText().length()));

Date tdt;

res=st.executeQuery("select \* from canceltbl");

while(res.next())

{

t=res.getString(1);

tdt=formatter.parse(t);

if(tdt.after(sdate)&&tdt.before(edate)||tdt.equals(sdate)||tdt.equals(edate))

{

v=new Vector();

v.add(tbl1.getRowCount()+1);

v.add(t);

v.add(res.getString(3));

v.add(res.getString(4));

v.add(res.getString(5));

v.add(res.getString(8));

model1.addRow(v);

}

}

}

if(mtype.getSelectedItem().toString().equals("Date"))

{

if(tsearch.getText().length()!=0)

{

String t="";

SimpleDateFormat formatter=new SimpleDateFormat("dd/MM/yyyy");

Date cdt=formatter.parse(tsearch.getText());

Date tdt;

res=st.executeQuery("select \* from canceltbl");

while(res.next())

{

t=res.getString(1);

tdt=formatter.parse(t);

if(tdt.equals(cdt))

{

v=new Vector();

v.add(tbl1.getRowCount()+1);

v.add(t);

v.add(res.getString(3));

v.add(res.getString(4));

v.add(res.getString(5));

v.add(res.getString(8));

model1.addRow(v);

}

}

}

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

public void bcopy()

{

int i,j=10;

try

{

PrintWriter pw=new PrintWriter("bookingReport.txt");

pw.println("\t\t\t Aamya Beauty Care Center ");

pw.println("\t\t\t Kalimati Road-8331001");

pw.println("\t\t\t Ph.No.:-8603100915");

pw.println("\t\t\t\t\t\t Date:"+tdate.getText());

pw.println("\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

pw.println("\tS.No. Booking Date\tBooking ID\tCustomer/Member ID\tTotal\t");

pw.println("\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(i=0;i<tbl.getRowCount();i++)

{

pw.println("\t"+tbl.getValueAt(i,0)+"\t"+tbl.getValueAt(i,1)+"\t "+tbl.getValueAt(i,2)+"\t "+tbl.getValueAt(i,3)+"\t\t"+tbl.getValueAt(i,4));

j--;

}

for(;j>0;j--)

pw.println();

pw.println("\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

pw.close();

Runtime rt=Runtime.getRuntime();

Process pr=rt.exec("notepad bookingReport.txt");

}

catch(Exception ee)

{

System.out.println(ee);

}

}

public void ccopy()

{

int i,j=10;

try

{

PrintWriter pw=new PrintWriter("bookingReport.txt");

pw.println("\t\t\t Aamya Beauty Care Center ");

pw.println("\t\t\t Kalimati Road-8331001");

pw.println("\t\t\t Ph.No.:-8603100915");

pw.println("\t\t\t\t\t\t Date:"+tdate.getText());

pw.println("\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

pw.println("\tS.No. Cancellatin Date\tCancellation ID\t Booking ID\tCustomer/Member ID\tTotal\t");

pw.println("\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(i=0;i<tbl1.getRowCount();i++)

{

pw.println("\t"+tbl1.getValueAt(i,0)+"\t"+tbl1.getValueAt(i,1)+"\t "+tbl1.getValueAt(i,2)+"\t "+tbl1.getValueAt(i,3)+"\t\t"+tbl1.getValueAt(i,4)+"\t\t"+tbl1.getValueAt(i,5));

}

for(;j>0;j--)

pw.println();

pw.println("\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

pw.close();

}

catch(Exception ee)

{

System.out.println(ee);

}

}

public void itemStateChanged(ItemEvent ie)

{

if(ie.getSource()==rpbook||ie.getSource()==rpcan)

{

if(ie.getStateChange()==ItemEvent.SELECTED)

display();

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==mtype)

{

lsearch.setVisible(false);

tsearch.setVisible(false);

month.setVisible(false);

if(mtype.getSelectedItem().toString().equals("Monthly"))

{

lsearch.setText("Select Month");

lsearch.setVisible(true);

month.setSelectedIndex(0);

month.setVisible(true);

}

if(mtype.getSelectedItem().toString().equals("Date"))

{

lsearch.setText("Enter Date");

lsearch.setVisible(true);

tsearch.setVisible(true);

}

}

if(e.getSource()==brefresh)

{

if(flg==1)

{

bsearch();

bprint.setEnabled(true);

}

if(flg==2)

{

csearch();

bprint.setEnabled(true);

}

}

if(e.getSource()==bprint)

{

bprint.setEnabled(false);

if(tbl.getRowCount()!=0||tbl1.getRowCount()!=0)

{

if(flg==1)

bcopy();

if(flg==2)

ccopy();

}

else

JOptionPane.showMessageDialog(null,"No Record Available!!");

}

if(e.getSource()==bcan)

this.dispose();

}

public static void main(String argv[])

throws IOException

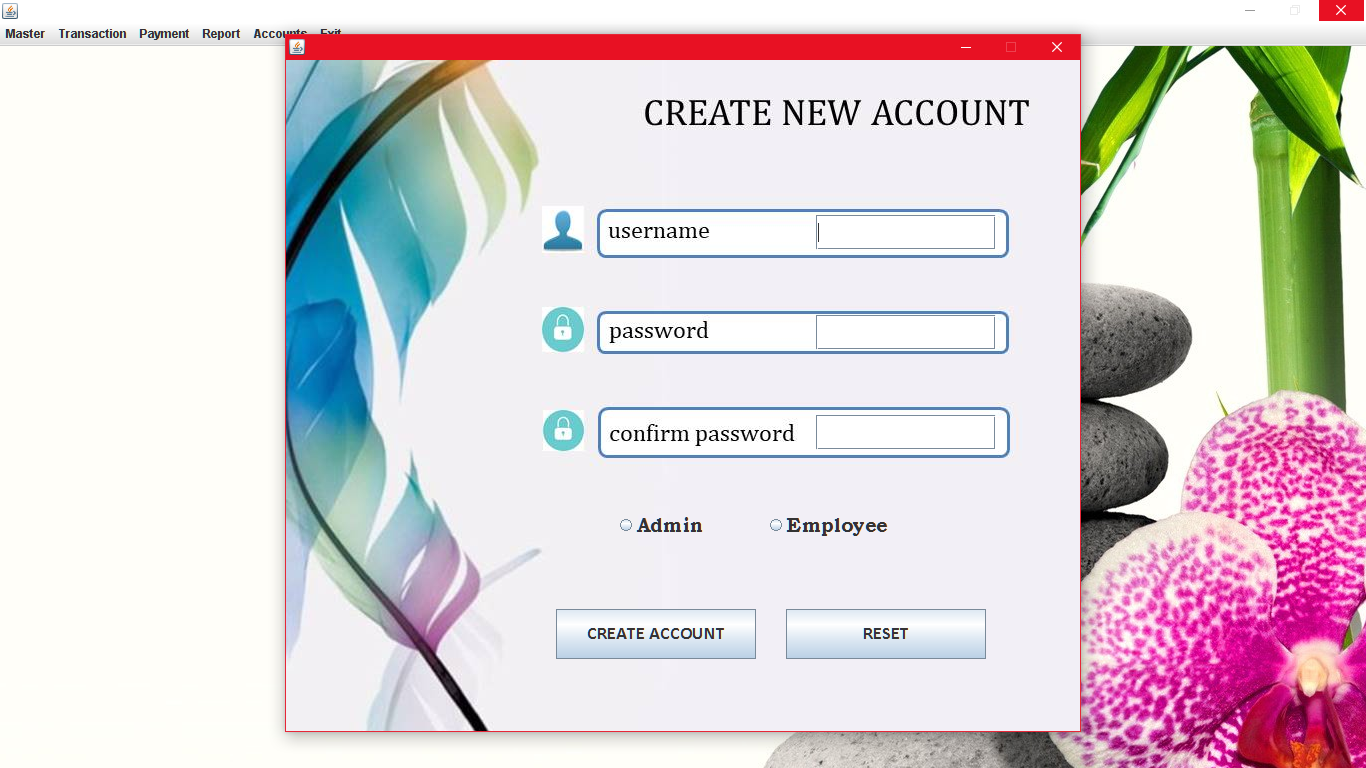
{

report obj=new report();

}

}

* **Create Account:**Here the authorization is given to the person for accessing this application according to their status. This form is only accessed be Admin user.



import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.sql.\*;

public class createuser extends JFrame implements ActionListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/crea.jpg"));

String str="";

JLabel ctype= new JLabel("User Type");

JLabel cname= new JLabel("User Name");

JLabel cpswd= new JLabel("Password");

JLabel ccpswd= new JLabel("Confirm Password");

JTextField ttype=new JTextField();

JTextField tname=new JTextField();

JPasswordField tpswd=new JPasswordField();

JPasswordField tcpswd=new JPasswordField();

JButton bsubmit=new JButton("CREATE ACCOUNT");

JButton breset=new JButton("RESET");

JRadioButton radmin=new JRadioButton("Admin");

JRadioButton ruser=new JRadioButton("Employee");

public createuser()

{

int x,y;

setLayout(null);

setVisible(true);

setSize(800,700);

getContentPane().setBackground(Color.WHITE);

lbl.setBounds(0,0,getWidth(),getHeight());

radmin.setBounds(330,450,100,30);

ruser.setBounds(480,450,130,30);

tname.setBounds(530,155,180,35);

tpswd.setBounds(530,255,180,35);

tcpswd.setBounds(530,355,180,35);

add(bsubmit);

add(breset);

add(tname);

add(tpswd);

add(tcpswd);

bd.add(radmin);

bd.add(ruser);

add(radmin);

add(ruser);

bsubmit.addActionListener(this);

breset.addActionListener(this);

}

public int check(String usr)

{

int flg=1;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from logintbl where username='"+usr+"'");

while(res.next())

{

flg=0;

}

if(flg==0)

JOptionPane.showMessageDialog(null,"Username already Exists!!");

}

catch(Exception ee)

System.out.println(ee);

return flg;

}

public void actionPerformed(ActionEvent e)

{

int flg=0,tflg=0;

String pswrd="";

String cpswrd="";

if(e.getSource()==bsubmit)

{

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

str="insert into logintbl(username,password,usertype) values(?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

if(tname.getText().length()==0||tcpswd.getText().length()==0||tpswd.getText().length()==0||bd.getSelection()==null)

{

flg=1;

JOptionPane.showMessageDialog(null,"Empty Fields");

}

else

{

pswrd=tpswd.getText();

cpswrd=tcpswd.getText();

if(pswrd.equals(cpswrd))

tflg=1;

}

if(tflg==0)

JOptionPane.showMessageDialog(null,"Password is not match");

if(flg==0&&tflg==1&&check(tname.getText())==1)

{

if(radmin.isSelected()==true)

str="Admin";

else

str="Employee";

ps.setString(1,tname.getText());

ps.setString(2,tcpswd.getText());

ps.setString(3,str);

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Saved..");

}

}

catch(Exception ee)

{

System.out.println(ee);

}}

if(e.getSource()==breset)

{

tname.setText("");

tcpswd.setText("");

}}

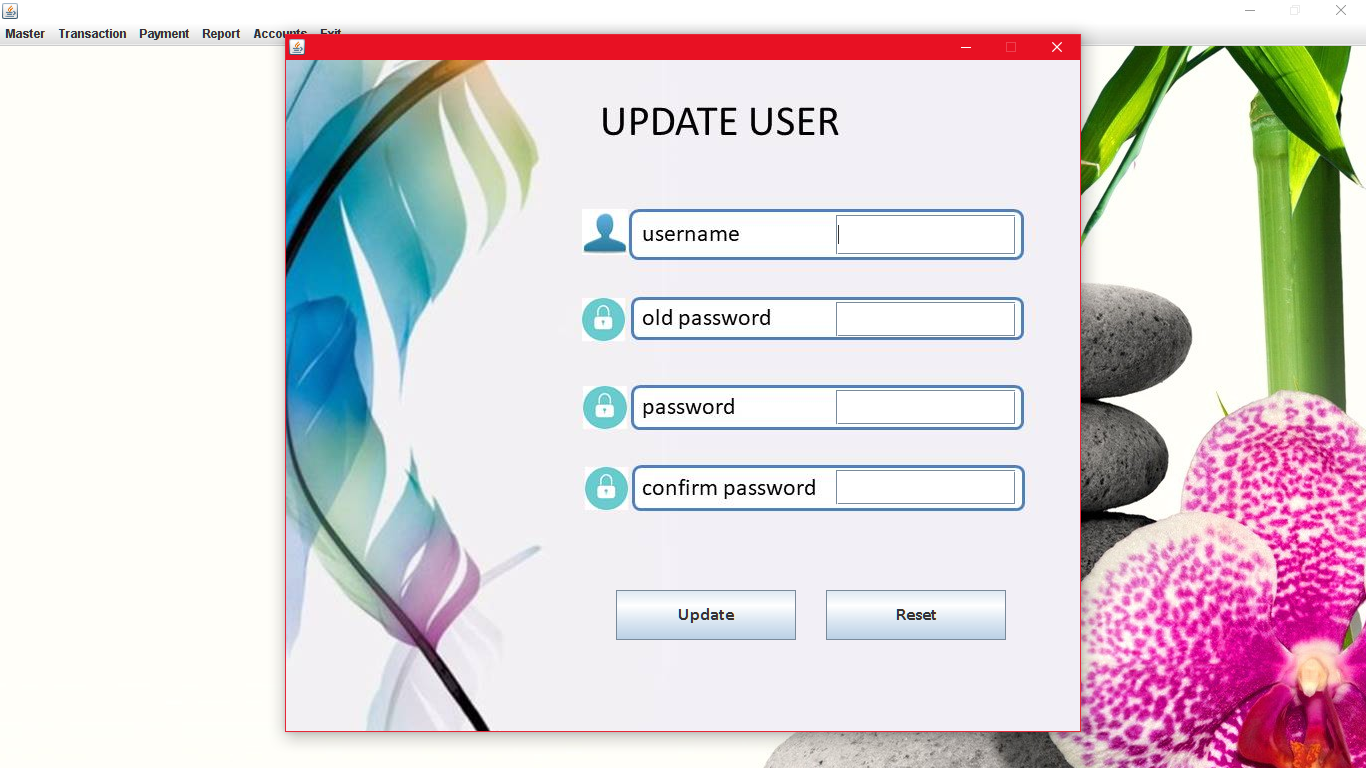
public static void main(String argv[])throws IOException

{

createuser obj=new createuser();

}}

* **Update Account:**Here the existing user can change their account password. This form can only be accessed be the User Admin.



import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.sql.\*;

public class upuser extends JFrame implements ActionListener

{

JLabel lbl=new JLabel(new ImageIcon("../Images/upuser.jpg"));

String str="";

JLabel cname= new JLabel("User Name");

JLabel coldpswd= new JLabel("Old Password");

JLabel cpswd= new JLabel("Password");

JLabel ccpswd= new JLabel("Confirm Password");

JTextField tname=new JTextField();

JPasswordField toldpswd=new JPasswordField();

JPasswordField tpswd=new JPasswordField();

JPasswordField tcpswd=new JPasswordField();

JButton bupdate=new JButton("Update");

JButton breset=new JButton("Reset");

public upuser()

{

int x,y;

setLayout(null);

setVisible(true);

setSize(800,700);

getContentPane().setBackground(Color.WHITE);

lbl.setBounds(0,0,getWidth(),getHeight());

tname.setBounds(550,155,180,40);

toldpswd.setBounds(550,242,180,35);

breset.setBounds(540,530,180,50);

add(tname);

add(tpswd);

add(tcpswd);

add(toldpswd);

add(bupdate);

add(breset);

add(lbl);

bupdate.addActionListener(this);

breset.addActionListener(this);

}

public void upd(String usr,String psd)

{

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

Statement st=con.createStatement();

String str="update logintbl set password='"+psd+"' where username='"+usr+"'";

st.executeUpdate(str);

JOptionPane.showMessageDialog(null,"Account Updated!!");

}

catch(Exception ee)

{

System.out.println(ee);

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bupdate)

{

int flg=0;

String stname=tname.getText();

String stpswd=toldpswd.getText();

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:SpaSystemdsn");

String str="select \* from logintbl";

Statement st=con.createStatement();

ResultSet res=st.executeQuery(str);

p:{while(res.next())

{

if(stname.equals(res.getString(1))&&stpswd.equals(res.getString(2)))

{

if(tpswd.getText().length()==0||tcpswd.getText().length()==0)

{

JOptionPane.showMessageDialog(null,"Empty Field!!!");

break p;

}

if(tpswd.getText().equals(tcpswd.getText()))

{

flg=1;

break p;

}

else

{

JOptionPane.showMessageDialog(null,"Password didn't Matched!!");

break p;

}

}

else

{

JOptionPane.showMessageDialog(null,"Invalid Username & Password!!");

break p;

}

}

}

if(flg==1)

upd(stname,tpswd.getText());

}

catch(Exception ee)

{

System.out.println(ee);

}

}

if(e.getSource()==breset)

{

tname.setText("");

toldpswd.setText("");

tpswd.setText("");

tcpswd.setText("");

}

}

public static void main(String argv[])throws IOException

{

upuser obj=new upuser();

}

}

**Future Scope of the project**

Now days it can be clearly seen that Spa’s are growing very fast. They are expanding their business in all cities across the nation. Every organizer wants to serve their customer in best way. For this they are doing a lot of advertisements on television, newspapers and also on internet etc. Every organizer wants to give their business a professional approach which leads to greater confidence of customer in the organization. But while initial investigation and facts finding, it was seen that traditional way of keeping record of customers, treatments and other relevant entity is difficult for the store keepers. Also they face difficulties when they have to search for a particular record in case of reference. This project will help them to overcome such problems. It is designed in such a way that it fulfils requirements of a SPA.

This software can serve as a prototype for the new requests made by the different agencies. It can serve as a base and can be given as sample for their use. This will increase the confidence of the user as it covers all the requirements of the system.

**Limitations of the project**

* The current Software is stand-alone application. In Future It can be extended to Client-Server.
* The Diet Products are not selled here.Thus the diet products are brought from other places and mentained.
* There is no method or process for backup the data. Thus one has to do this manually.

**Bibliography**

**Books:**

* The Complete Reference Java By [Herbert Schildt](https://www.google.co.in/search?sa=X&q=Herbert+Schildt&stick=H4sIAAAAAAAAAOPgE-LSz9U3SC7MzkvPUAKzjcwyDMyztGSyk630k_Lzs_XLizJLSlLz4svzi7KtEktLMvKLAEOr43A4AAAA&ved=0ahUKEwii36yckoHbAhXH6Y8KHZKMDWAQmxMIsQMoATAU)
* ANSI Java

**Web Sites:**

* www.wikipedia .com
* www.google.co.in

**Conclusion**

* The demand of Java for application is increasing day by day in Software industry, due to high expectations of client companies.
* Hence an attempt of automating an office application had added to our learning experience.
* It has also helped in adopting an analytical approach to solving and made us realize that system development is a step by step process.
* Thereby appreciating the role of SDLC in organizing the complex process of system development into manageable chunks. Indeed it was a great learning experience.

From a proper analysis of positive points and constraints on the components, it can be concluded that the application is working properly and meeting to maximum all user requirements. This component or application can be easily plugged in others systems with different OS.