**Dharmsinh Desai University, Nadiad**

**Department of Information Technology**

**Advanced Java Technology, IT619**

**B.Tech. IT, Sem: VI**

**Experiment – 01**

**Submitted By**

**Roll No.:IT076**

**Name:DISHANT MODH**

**Aim:** Create a GUI based application which can be used as a telephone directory application. The telephone directory is stored as a database and has one table named telephoneDir. The telephoneDir database table stores three different information: telephone no., owner name, and owner address. The owner name is made of three parts: First name, middle name, and last name. The owner address is made of five parts: house no., address 1, address 2, area name, and city name. The application allows search facility. The search is possible using three different ways:

1. Search by telephone no.

2. Search by name (one of first name, middle name, and last name) with exactly match

and part of the name.

3. Search by address (one of address 1, address 2, area name, and city) with exactly

match and part of address.

**Code:**

package dmx;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

class directory extends Frame implements ActionListener,ItemListener{

TextField tf1 = new TextField();

Choice c1 = new Choice();

Choice c2 = new Choice();

Button srcbtn = new Button("SEARCH");

Panel ptop = new Panel();

TextArea ta1 = new TextArea();

Label status = new Label("Records Found =0");

Label error = new Label("Enter the Valid Text");

Connection cn = null;

PreparedStatement pstat = null;

ResultSet rs =null;

public directory()

{

super("Telephone Directory");

Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();

this.setLocation(dim.width/2-this.getPreferredSize().width/2, dim.height/2-this.getPreferredSize().height/2);

setVisible(true);

setSize(800,400);

c1.add("Telephone No.");

c1.add("Name");

c1.add("Address");

c1.addItemListener(this);

srcbtn.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

dispose();

}

});

setLayout(new BorderLayout());

ptop.setLayout(new GridLayout(4,3));

ptop.add(new Label("Search Option 1"));

ptop.add(new Label(""));

ptop.add(c1);

ptop.add(new Label("Search Option 2"));

ptop.add(new Label(""));

ptop.add(c2);

c2.setVisible(false);

ptop.add(new Label("Enter Text:"));

ptop.add(error);

error.setVisible(false);

ptop.add(tf1);

ptop.add(new Label(""));

ptop.add(new Label(""));

ptop.add(srcbtn);

add("North",ptop);

add("Center",ta1);

add("South",status);

ta1.setEditable(false);

String driver = "org.apache.derby.jdbc.ClientDriver";

try {

Class.forName(driver);

cn = DriverManager.getConnection("jdbc:derby://localhost:1527/DMX","dmx","dmx");

}

catch(ClassNotFoundException e)

{

System.out.println(""+e.toString());

}

catch(SQLException se)

{

while(se!=null)

{

System.out.println(""+se.toString());

se = se.getNextException();

}

}

}

public void itemStateChanged(ItemEvent e)

{

String arg = e.getItem().toString();

if(arg.equalsIgnoreCase("Telephone No."))

c2.setVisible(false);

else if(arg.equalsIgnoreCase("Name"))

{

c2.removeAll();

c2.add("First Name");

c2.add("Middle Name");

c2.add("Last Name");

c2.setVisible(true);

}

else if(arg.equalsIgnoreCase("Address"))

{

c2.removeAll();

c2.add("Area");

c2.add("City");

c2.setVisible(true);

}

}

public void actionPerformed(ActionEvent ae)

{

ta1.setText("Refreshed");

String query = new String("select \* from DMX");

int len = 0;

len = tf1.getText().toString().trim().length();

if(len == 0)

{

error.setVisible(true);

}

else

{

error.setVisible(false);

if(c1.getSelectedItem().equals("Telephone No."))

{

query += " where Phone\_Number = ?";

try{

pstat = cn.prepareStatement(query);

pstat.setString(1, tf1.getText().toString().trim());

}

catch(SQLException e)

{

System.out.println(""+e.toString());

}

}

else if(c1.getSelectedItem().equals("Name"))

{

if(c2.getSelectedItem().equals("First Name"))

query +=" where First\_Name = ?";

if(c2.getSelectedItem().equals("Middle Name"))

query +=" where Middle\_Name = ?";

if(c2.getSelectedItem().equals("Last Name"))

query +=" where Last\_Name = ?";

try{

pstat = cn.prepareStatement(query);

pstat.setString(1, tf1.getText().toString().trim());

}

catch(SQLException e)

{

System.out.println(""+e.toString());

}

}

else if(c1.getSelectedItem().equals("Address"))

{

if(c2.getSelectedItem().equals("Area"))

query +=" where Area = ?";

if(c2.getSelectedItem().equals("City"))

query +=" where City = ?";

try{

pstat = cn.prepareStatement(query);

pstat.setString(1, tf1.getText().toString().trim());

}

catch(SQLException e)

{

System.out.println(""+e.toString());

}

}

else

{

try{

pstat = cn.prepareStatement(query);

pstat.setString(1, tf1.getText().toString().trim());

}

catch(SQLException e)

{

System.out.println(""+e.toString());

}

}

try

{

System.out.println(query);

rs=pstat.executeQuery();

}

catch(NullPointerException ne)

{

System.out.println("Text Null3");

ta1.setText("No Records Found.");

status.setText("Records Found = 0");

} catch (SQLException ex) {

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

}

if(rs != null)

{

ta1.setText("Number\t\tFName\t\tMName\t\tLNAME\t\tHouse\_Number\t\ttAddress1\t\tAddress2\t\tArea\t\tCity\n");

int count=0;

try{

while(rs.next())

{

ta1.append("" + rs.getString(1) + "\t");

ta1.append("" + rs.getString(2) + "\t\t");

ta1.append("" + rs.getString(3) + "\t\t");

ta1.append("" + rs.getString(4) + "\t\t");

ta1.append("" + rs.getString(5) + "\t\t");

ta1.append("" + rs.getString(6) + "\t\t");

ta1.append("" + rs.getString(7) + "\t\t");

ta1.append("" + rs.getString(8) + "\t\t");

ta1.append("" + rs.getString(9) + "\t\t\n");

count++;

}

}

catch(SQLException e)

{

System.out.println(""+e.toString());

}

status.setText("Records Found = " + count);

}

}

}

}

public class DMX {

public static void main(String[] args) {

// TODO code application logic here

Frame Dir = new directory();

}

}

**Input/Output:**

