**Dharmsinh Desai University, Nadiad**

**Department of Information Technology**

**Advanced Java Technology, IT619**

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

**Experiment – 02**

**Submitted By**

**Roll No.:IT076**

**Name:DISHANT MODH**

**Aim:**  Create a GUI based application which can be used for telephone directory

modification (administrator part for the above problem statement). The application allows

two modification operations: create new telephone connection, and delete a telephone

connection. The insert operation takes telephone no., name, and address as input

parameters. The delete operation has verification step in which the user must perform the

verification of the telephone connection which is about to be deleted. Once the

verification is done, the application allows deleting the telephone connection. Design

appropriate GUI to accommodate all stated features.

**Code:**

package dmx2;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

class InsertRecord {

private static Dialog d;

String driver;

Label status1 = new Label("Number Is Not Valid");

Label status = new Label("Data Inserted Successfully");

Label status2= new Label("Number Is Already Exist");

TextField itf1 = new TextField();

TextField itf2 = new TextField();

TextField itf3 = new TextField();

TextField itf4 = new TextField();

TextField itf5 = new TextField();

TextField itf6 = new TextField();

TextField itf7 = new TextField();

TextField itf8 = new TextField();

TextField itf9 = new TextField();

PreparedStatement pst = null;

Connection cn = null;

InsertRecord(Frame f, String driver) {

this.driver = driver;

Label il1 = new Label("Telephone No");

Label il2 = new Label("First Name");

Label il3 = new Label("Middle Name");

Label il4 = new Label("Last Name");

Label il5 = new Label("House No");

Label il6 = new Label("Address1");

Label il7 = new Label("Address2");

Label il8 = new Label("Area");

Label il9 = new Label("City");

Button ibtn1 = new Button ("Done");

Button ibtn = new Button ("Add New");

Panel dp = new Panel();

Panel dp1 = new Panel();

dp.setLayout(new GridLayout(10,3));

dp1.setLayout(new FlowLayout());

dp.add(il1);

dp.add(itf1);

dp.add(status1);

dp.add(il2);

dp.add(itf2);

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

dp.add(il3);

dp.add(itf3);

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

dp.add(il4);

dp.add(itf4);

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

dp.add(il5);

dp.add(itf5);

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

dp.add(il6);

dp.add(itf6);

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

dp.add(il7);

dp.add(itf7);

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

dp.add(il8);

dp.add(itf8);

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

dp.add(il9);

dp.add(itf9);

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

dp.add(status);

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

dp.add(status2);

status.setVisible(false);

status1.setVisible(false);

status2.setVisible(false);

dp1.add(ibtn);

dp1.add(ibtn1);

d = new Dialog(f , "Insert Contact", true);

d.setLayout( new BorderLayout() );

d.add("North",dp);

d.add("South",dp1);

ibtn.addActionListener ( new ActionListener()

{

public void actionPerformed( ActionEvent e )

{

getdata();

}

});

ibtn1.addActionListener ( new ActionListener()

{

public void actionPerformed( ActionEvent e )

{

d.dispose();

}

});

d.addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

d.dispose();

}

});

d.setSize(500,300);

d.setVisible(true);

}

void getdata()

{

try{

Class.forName(this.driver);

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

pst=cn.prepareStatement("insert into DMX values(?,?,?,?,?,?,?,?,?)");

int i = itf1.getText().toString().trim().length();

PreparedStatement pst1=cn.prepareStatement("select \* from DMX where Phone\_Number = ?");

pst1.setString(1,itf1.getText().toString().trim());

ResultSet rs1=pst1.executeQuery();

if(rs1.next() == true)

{

status.setVisible(false);

status2.setVisible(true);

}

else if(i != 10)

{

status.setVisible(false);

status1.setVisible(true);

status2.setVisible(false);

}

else{

status1.setVisible(false);

status2.setVisible(false);

pst.setString(1,itf1.getText().toString().trim());

pst.setString(2,itf2.getText().toString().trim());

pst.setString(3,itf3.getText().toString().trim());

pst.setString(4,itf4.getText().toString().trim());

pst.setString(5,itf5.getText().toString().trim());

pst.setString(6,itf6.getText().toString().trim());

pst.setString(7,itf7.getText().toString().trim());

pst.setString(8,itf8.getText().toString().trim());

pst.setString(9,itf9.getText().toString().trim());

int ix=pst.executeUpdate();

System.out.println(ix+" records inserted");

cn.close();

status.setVisible(true);

}

}

catch(ClassNotFoundException e)

{

System.out.println("Class GoTO Bhai Pela");

}

catch(SQLException e){

System.out.println("SQL No Locho");

}

}

}

class DeleteRecord {

private static Dialog d;

private static Dialog d2;

PreparedStatement pst = null;

Connection cn = null;

String driver;

TextField dtf1 = new TextField();

DeleteRecord(Frame f, String driver)

{

this.driver = driver;

Label dl1 = new Label("Telephone No:");

Label d2l2 = new Label("Are You Sure That You want to Delete the Contact.");

Label status = new Label("Data Deleted Successfully");

Button dbtn1 = new Button ("Delete");

Button dbtn = new Button ("Cancel");

Button db2l1 = new Button ("YES");

Button db2l2 = new Button ("NO");

Panel dp = new Panel();

Panel dp1 = new Panel();

Panel dp2 = new Panel();

dp.setLayout(new GridLayout(2,2));

dp1.setLayout(new FlowLayout(FlowLayout.CENTER));

dp2.setLayout(new FlowLayout(FlowLayout.CENTER));

dp.add(dl1);

dp.add(dtf1);

dp.add(status);

dp1.add(dbtn);

dp1.add(dbtn1);

dp2.add(db2l1);

dp2.add(db2l2);

d = new Dialog(f , "Delete Contact", true);

d2 = new Dialog(d , "Verification", true);

d.setLayout( new BorderLayout() );

d2.setLayout( new BorderLayout());

d.add("North",dp);

d.add("South",dp1);

d2.add("North",d2l2);

d2.add("South",dp2);

d2.pack();

dbtn.addActionListener ( new ActionListener()

{

public void actionPerformed( ActionEvent e )

{

d.dispose();

}

});

dbtn1.addActionListener ( new ActionListener()

{

public void actionPerformed( ActionEvent e )

{

status.setVisible(false);

d2.setVisible(true);

}

});

db2l1.addActionListener ( new ActionListener()

{

public void actionPerformed( ActionEvent e )

{

deleteData();

d2.dispose();

status.setVisible(true);

}

});

db2l2.addActionListener ( new ActionListener()

{

public void actionPerformed( ActionEvent e )

{

d2.dispose();

}

});

d2.addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

d2.dispose();

}

});

d.addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

d.dispose();

}

});

d.pack();

status.setVisible(false);

d.setVisible(true);

}

void deleteData()

{

try{

Class.forName(this.driver);

final String DELETE\_USERS\_SQL = "delete from DMX where Phone\_NUMBER = ?";

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

pst = null;

pst=cn.prepareStatement(DELETE\_USERS\_SQL);

pst.setString(1,dtf1.getText().toString().trim());

int i=pst.executeUpdate();

System.out.println(i+" records deleted");

cn.close();

}

catch(ClassNotFoundException e)

{

System.out.println("Class GoTO Bhai Pela");

}

catch(SQLException e){

System.out.println("SQL No Locho");

}

}

}

class dir extends Frame implements ActionListener,ItemListener

{

TextField tf1 = new TextField();

Choice c1 = new Choice();

Choice c2 = new Choice();

Button srcbtn = new Button("SEARCH");

Button Insert = new Button("INSERT");

Button Delete = new Button("DELETE");

Panel ptop = new Panel();

TextArea ta1 = new TextArea();

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

Label error = new Label("Enter The Valid Data:");

Connection cn = null;

PreparedStatement pstat = null;

ResultSet rs =null;

public dir()

{

Frame f = new Frame("Telephone Directory");

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

this.setLocation(dim.width/4-this.getPreferredSize().width/2, dim.height/4-this.getPreferredSize().height/3);

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)

{

System.exit(0);

}

});

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(Insert);

Insert.setVisible(true);

ptop.add(Delete);

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();

}

}

Insert.addActionListener ( new ActionListener()

{

public void actionPerformed( ActionEvent e )

{

new InsertRecord(f,driver);

}

});

Delete.addActionListener ( new ActionListener()

{

public void actionPerformed( ActionEvent e )

{

new DeleteRecord(f,driver);

}

});

}

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 upper(First\_Name) = ?";

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

query +=" where upper(Middle\_Name) = ?";

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

query +=" where upper(Last\_Name) = ?";

try{

pstat = cn.prepareStatement(query);

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

}

catch(SQLException e)

{

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

}

}

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

{

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

query +=" where upper(Area) = ?";

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

query +=" where upper(City) = ?";

try{

pstat = cn.prepareStatement(query);

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

}

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(dir.class.getName()).log(Level.SEVERE, null, ex);

}

if(rs != null)

{

ta1.setText("Number\t\tFName\t\tMName\t\tLNAME\t\tHouse No.\t\tAddress1\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");

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

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

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

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

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

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

count++;

}

}

catch(SQLException e)

{

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

}

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

}

}

}

}

public class DMX2 {

public static void main(String[] args) {

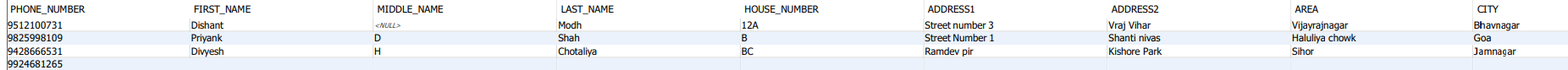
new dir();

}

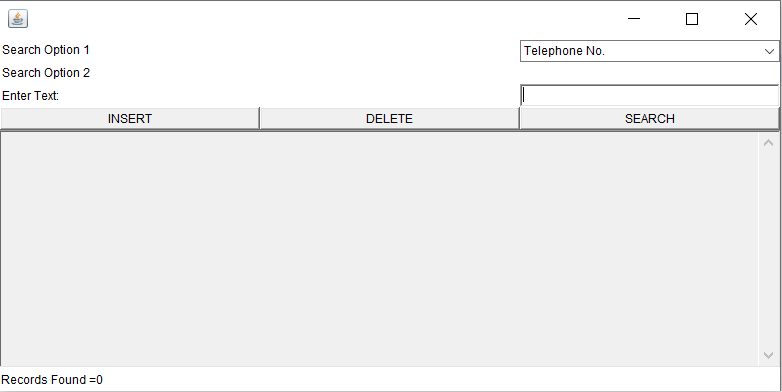
}

**Input/Output:**

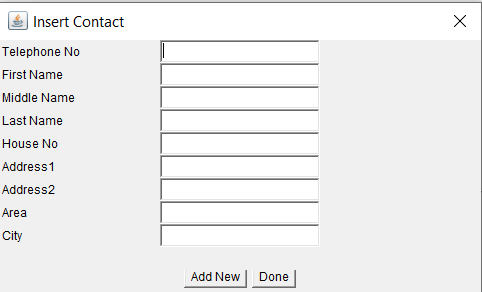
DATABASE



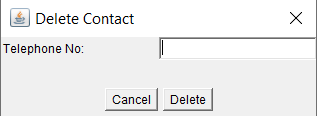
LAYOUT



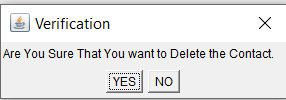
INSERT



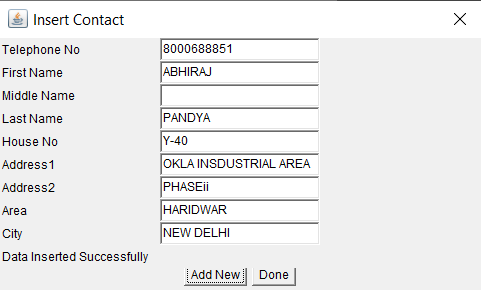
DELETE



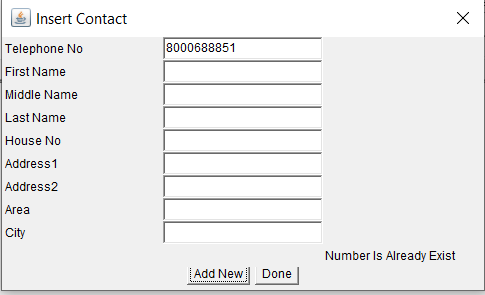
VERIFICATION DELETE



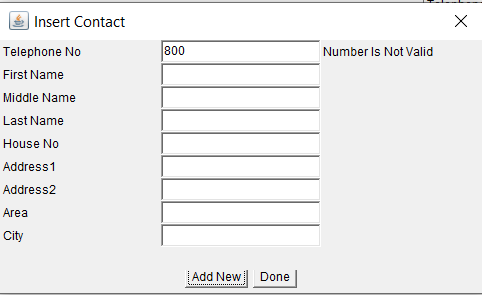
INSERTING DATA



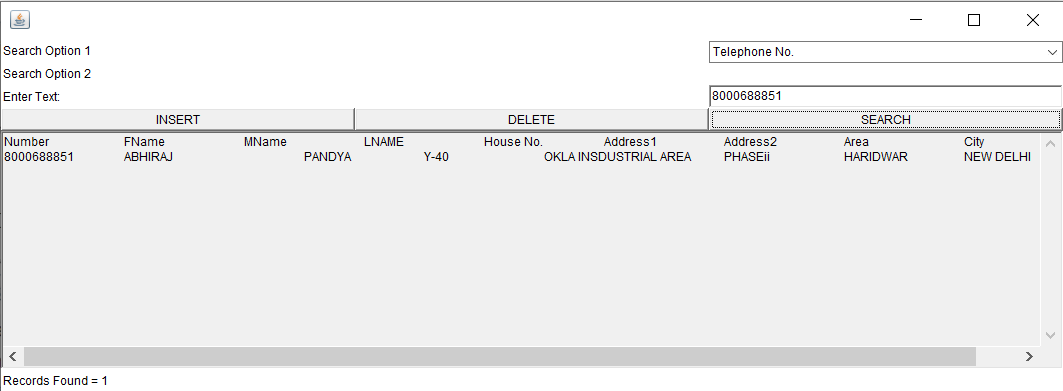
WHEN NUMBER/DATA IS ALREADY EXIST



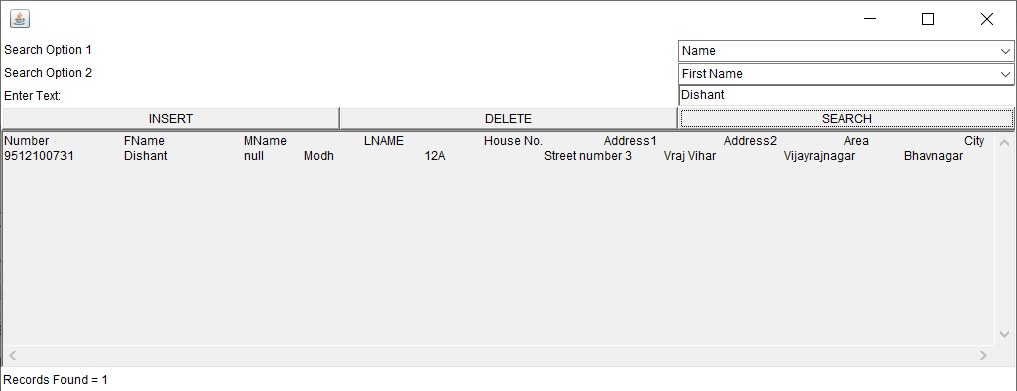
NOT VALID NUMBER



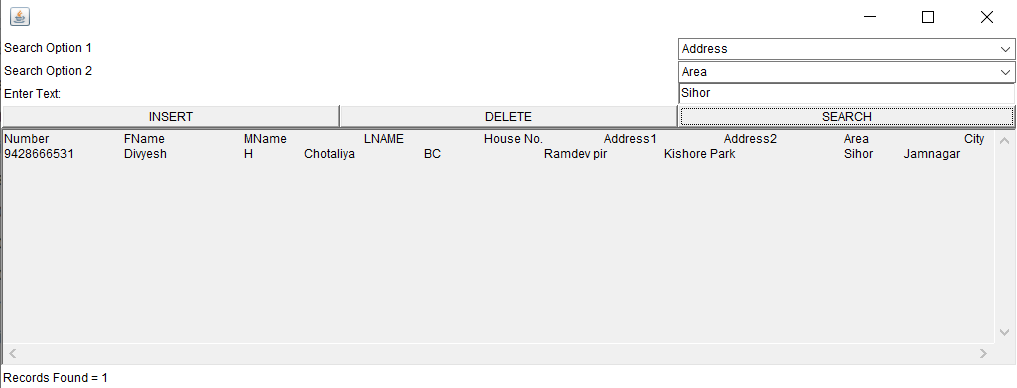
SEARCHING DATA BY NUMBER



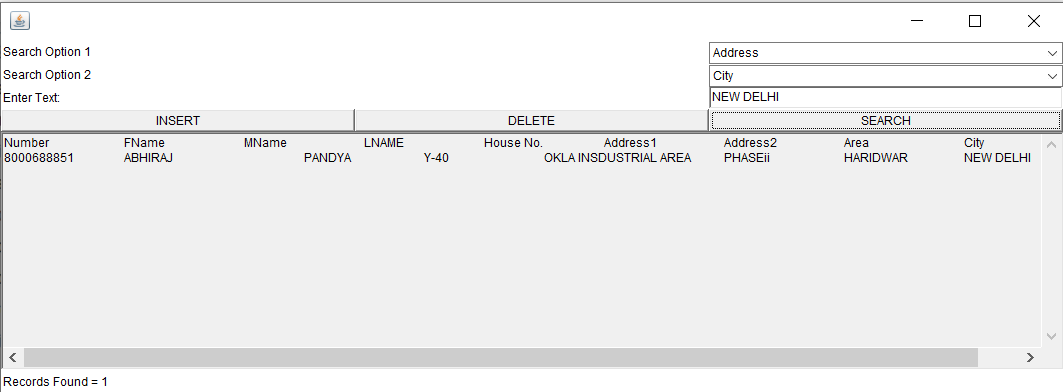
SEARCHING DATA BY NAME



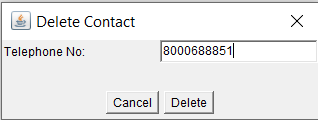
SEARCHING DATA BY AREA



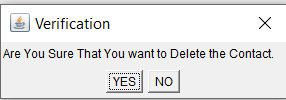
SEARCHING DATA BY CITY



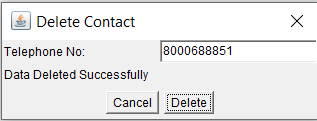
DELETING THE NUMBER



THEN IT WILL ASK YES/ NO FOR VERIFICATION



AFTER CLICKING ON YES THEN IT WILL SHOW STATUS AS DELETED SUCCESSFULLY



SO, THE DATA IS DELETED

NOW SEARCHING THE DELETED DATA

