JAVA SWING BASED – DBASE PROJECT-SPAM-SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of
BACHELOR OF TECHNOLOGY
IN

INFORMATION TECHNOLOGY

By

Shanigaram Suhidhar
<1602-20-737-048>

Under the Guidance of

B. Leelavathy

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2021-2022

BONAFIDE CERTIFICATE

This to Certify that the project report titled

"DBASE PROJECT-SPAM" project work of Mr.Shanigaram Suhidhar bearing Roll.no:1602-20-737-048 who carried out this project under my supervision in the IV semester forthe academic year 2021-2022.

Signature
external examiner

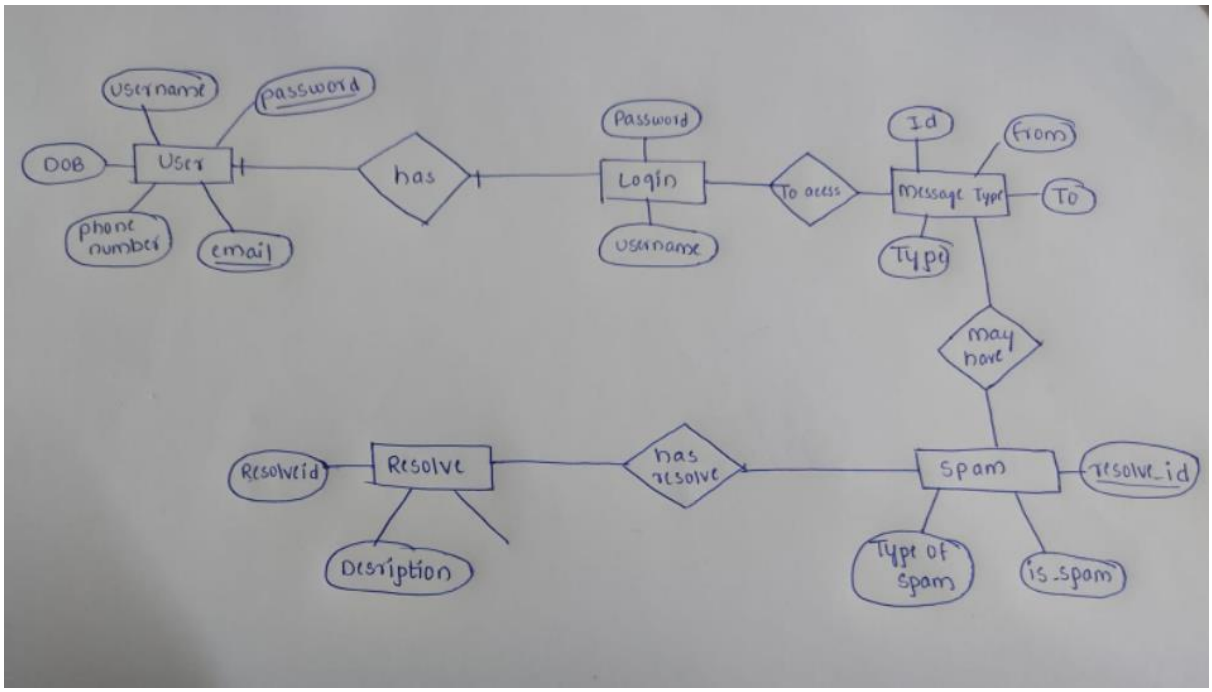
Signature
internal examiner

DBASE PROJECT-SPAM

ABSTRACT:

Dbase project spam helps us in identifying spams. If the user has similar id and password then we can know whether that is a spam or not. If spam then it resolves how to solve it.

ER DIAGRAM



DDL OPERATIONS:

create table user_details(username varchar2(20),email varchar2(20),password varchar2(20),ph_num number(10), d_o_b date);

```
SQL> create table user_details(username varchar2(20),email varchar2(20),password varchar2(20),ph_num number(10)
, d_o_b date);

Table created.

SQL> desc user_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERNAME                                           VARCHAR2(20)
 EMAIL                                              VARCHAR2(20)
 PASSWORD                                           VARCHAR2(20)
 PH_NUM                                             NUMBER(10)
 D_O_B                                              DATE
```

alter table user_details add primary key(email);

```
SQL> alter table user_details add primary key(email);

Table altered.

SQL> desc user_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 USERNAME                                           VARCHAR2(20)
 EMAIL                                     NOT NULL VARCHAR2(20)
 PASSWORD                                           VARCHAR2(20)
 PH_NUM                                             NUMBER(10)
 D_O_B                                              DATE
```

create table login(email varchar2(30), password varchar2(20));

```
SQL> create table login(
  2  email varchar2(30),
  3  password varchar2(20));

Table created.

SQL> desc login
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 EMAIL                                              VARCHAR2(30)
 PASSWORD                                           VARCHAR2(20)
```

alter table login add foreign key(email) references user_details;

```
SQL> alter table login add foreign key(email) references user_details;

Table altered.

SQL> desc login;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 EMAIL                                              VARCHAR2(30)
 PASSWORD                                           VARCHAR2(20)

SQL>
```

```
create table message_type(
  2  id number(5),
  3  from_ varchar2(20),
  4  to_ varchar2(20),
  5  type varchar2(20));
```

```
SQL> create table message_type(
  2  id number(5),
  3  from_ varchar2(20),
  4  to_ varchar2(20),
  5  type varchar2(20));

Table created.

SQL> desc message_type;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 ID                                                 NUMBER(5)
 FROM_                                              VARCHAR2(20)
 TO_                                                VARCHAR2(20)
 TYPE                                               VARCHAR2(20)

SQL>
```

```
create table spam(
  2  type_of_spam varchar2(20),
  3  is_spam varchar2(5),
  4  resolve_id number(5));
```

```
SQL> create table spam(
  2  type_of_spam varchar2(20),
  3  is_spam varchar2(5),
  4  resolve_id number(5));

Table created.

SQL> desc spam;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------
 TYPE_OF_SPAM                                       VARCHAR2(20)
 IS_SPAM                                            VARCHAR2(5)
 RESOLVE_ID                                         NUMBER(5)
```

alter table spam add primary key(resolve_id);

```
SQL> alter table spam add primary key(resolve_id);

Table altered.

SQL> desc spam;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 TYPE_OF_SPAM                                       VARCHAR2(20)
 IS_SPAM                                            VARCHAR2(5)
 RESOLVE_ID                                NOT NULL NUMBER(5)
```

create table resolve(
  2 resolve_id number(5),
  3 description varchar2(100));

```
SQL> create table resolve(
  2  resolve_id number(5),
  3  description varchar2(100));

Table created.

SQL> desc resolve;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RESOLVE_ID                                         NUMBER(5)
 DESCRIPTION                                        VARCHAR2(100)
```

DML OPERATIONS:

```
SQL> /
Enter value for username: suhidhar_prabhu
Enter value for email: suhidhar@gmail.com
Enter value for password: suhi@123
Enter value for ph_num: 8629812345
Enter value for d_o_b: 2-may-2002
old   1: insert into user_details values('&username','&email','&password',&ph_num,'&d_o_b')
new   1: insert into user_details values('suhidhar_prabhu','suhidhar@gmail.com','suhi@123',8629812345,'2-may-2
002')

1 row created.
```

```
SQL> /
Enter value for username: ramu
Enter value for email: ramu@gmail.com
Enter value for password: ramu@456
Enter value for ph_num: 9845092769
Enter value for d_o_b: 20-march-2003
old   1: insert into user_details values('&username','&email','&password',&ph_num,'&d_o_b')
new   1: insert into user_details values('ramu','ramu@gmail.com','ramu@456',9845092769,'20-march-2003')

1 row created.

SQL> select * from user_details;

USERNAME             EMAIL                PASSWORD                 PH_NUM
-------------------- -------------------- -------------------- ----------
D_O_B
---------
suhidhar_prabhu      suhidhar@gmail.com   suhi@123             8629812345
02-MAY-02

ramu                 ramu@gmail.com       ramu@456             9845092769
20-MAR-03
```

```
SQL> insert into login values('&email','&password');
Enter value for email: suhidhar@gmail.com
Enter value for password: suhi@123
old   1: insert into login values('&email','&password')
new   1: insert into login values('suhidhar@gmail.com','suhi@123')

1 row created.

SQL> /
Enter value for email: ramu@gmail.com
Enter value for password: ramu_432
old   1: insert into login values('&email','&password')
new   1: insert into login values('ramu@gmail.com','ramu_432')

1 row created.
```

```
SQL> select * from login;

EMAIL                          PASSWORD
------------------------------ --------------------
suhidhar@gmail.com             suhi@123
ramu@gmail.com                 ramu_432
```

```
SQL> insert into message_type values(&id,'&from_','&to_','&type');
Enter value for id: 101
Enter value for from_: pranay
Enter value for to_: suhidhar
Enter value for type: email
old   1: insert into message_type values(&id,'&from_','&to_','&type')
new   1: insert into message_type values(101,'pranay','suhidhar','email')

1 row created.

SQL> /
Enter value for id: 102
Enter value for from_: swetha
Enter value for to_: anil
Enter value for type: sms message
old   1: insert into message_type values(&id,'&from_','&to_','&type')
new   1: insert into message_type values(102,'swetha','anil','sms message')
```

```
SQL> select * from message_type;

        ID FROM_                TO_                  TYPE
---------- -------------------- -------------------- ------------------
       101 pranay               suhidhar             email
       102 swetha               anil                 sms message
```

```
SQL> insert into spam values('&type_of_spam','&is_spam',&resolve_id);
Enter value for type_of_spam: new_device_login
Enter value for is_spam: yes
Enter value for resolve_id: 201
old   1: insert into spam values('&type_of_spam','&is_spam',&resolve_id)
new   1: insert into spam values('new_device_login','yes',201)

1 row created.

SQL> /
Enter value for type_of_spam: multiple_login
Enter value for is_spam: no
Enter value for resolve_id: 0
old   1: insert into spam values('&type_of_spam','&is_spam',&resolve_id)
new   1: insert into spam values('multiple_login','no',0)
```

```
SQL> select * from spam;

TYPE_OF_SPAM         IS_SP RESOLVE_ID
-------------------- ----- ----------
new_device_login     yes          201
multiple_login       no             0
```

ARCHITECTURE AND TECHNOLOGY USED:

SOFTWARE USED:

Java Eclipse, Oracle 11g Database, Java SE version 8, SQL Plus.

Java SWING:

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

SQL:

Structure Query Language (SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

Java-SQL Connectivity using JDBC:

Java Database Connectivity (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
   try
   {
            Class.forName("oracle.jdbc.driver.OracleDriver");
   }
  catch (Exception e)
{
            System.err.println("Unable to find and load driver");
            System.exit(1);
}
public void connectToDB()
   {
                        try
                        {
                         connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:ORCL","mydbms","mydbms");
                         statement = connection.createStatement();

                        }
                        catch (SQLException connectException)
                        {
                         System.out.println(connectException.getMessage());
                         System.out.println(connectException.getSQLState());
                         System.out.println(connectException.getErrorCode());
                         System.exit(1);
                        }
   }
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

IMPLEMENTATION:

MainUI.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class MainUI extends JFrame implements ActionListener
{
	UserUI ob1;
	LoginUI ob2;
	MessageUI ob3;
	SpamUI ob4;
	ResolveUI ob5;

	JButton submit,modify,delete,m1,m2,m3,m4,m5;
	JPanel p1,p2,p3,p4,p5,pb;
	JMenuBar mb;

	public MainUI()
	{
		setSize(600,550);
		setLayout(null);
		setVisible(true);
		setTitle("DBASE PROJECT SPAM");

		ob1 = new UserUI();
		ob2 = new LoginUI();
		ob3 = new MessageUI();
		ob4 = new SpamUI();
		ob5 = new ResolveUI();

		createPanels();
		createMenu();
		createButtons();
		addComponents();
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
	}

	void createPanels()
	{
		p1 = ob1.p;
		p2 = ob2.p;
		p3 = ob3.p;
		p4 = ob4.p;
    p5 = ob5.p;

		pb = new JPanel(new FlowLayout(FlowLayout.CENTER,50,0));
		pb.setBounds(0,400,600,150);
	}

	void createMenu()
	{
		mb = new JMenuBar();

		m1 = new JButton("User");
		m1.setFocusable(false);

		m2 = new JButton("Login");
		m2.setFocusable(false);


		m3 = new JButton("Message");
```

```
		m3.setFocusable(false);

		m4 = new JButton("Spam");
		m4.setFocusable(false);


		m5 = new JButton("Resolve");
		m5.setFocusable(false);


		m1.addActionListener(this);
		m2.addActionListener(this);
		m3.addActionListener(this);
		m4.addActionListener(this);
		m5.addActionListener(this);

		mb.add(m1);
		mb.add(m2);
		mb.add(m3);
		mb.add(m4);
		mb.add(m5);

	}

	public void actionPerformed(ActionEvent e)
	{
		remove(p1);
		remove(p2);
		remove(p3);
		remove(p4);
		remove(p5);

		if(e.getSource()==m1)
			add(p1);

		else if(e.getSource()==m2)
			add(p2);
		else if(e.getSource()==m3)
			add(p3);


		else if(e.getSource()==m4)
			add(p4);

		else
			add(p5);

	}

	void createButtons()
	{
		submit = new JButton("Submit");
		submit.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent e)
			{
				JOptionPane.showMessageDialog(new JFrame(),"Successfully
Inserted!","NOTICE",JOptionPane.INFORMATION_MESSAGE);
			}
		});


		modify = new JButton("Modify");
```

```
		modify.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent e)
			{
				JOptionPane.showMessageDialog(new JFrame(),"Successfully
Modified!","NOTICE",JOptionPane.INFORMATION_MESSAGE);
			}
		});


		delete = new JButton("Delete");
		delete.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent e)
			{
				JOptionPane.showMessageDialog(new JFrame(),"Successfully
Deleted!","NOTICE",JOptionPane.INFORMATION_MESSAGE);
			}
		});

		pb.add(submit);
		pb.add(modify);
		pb.add(delete);
	}

	void addComponents()
	{
		add(p1);
		add(pb);
		setJMenuBar(mb);
	}

	public static void main(String a[])
	{
		new MainUI();
	}
}
```

UserUI.java

```
import javax.swing.*;

class UserUI
{
	JTextField tf1,tf2,tf3,tf4,tf5;
	JLabel l1,l2,l3,l4,l5;
	JPanel p;

	public UserUI()
	{
		createComponents();
		addComponents();
	}

	void createComponents()
	{
		tf1 = new JTextField();
		tf1.setBounds(250,20,200,30);
```

```
		tf2 = new JTextField();
		tf2.setBounds(250,80,200,30);

		tf3 = new JTextField();
		tf3.setBounds(250,140,200,30);

		tf4 = new JTextField();
		tf4.setBounds(250,200,200,30);

		tf5 = new JTextField();
		tf5.setBounds(250,260,200,30);

		l1 = new JLabel("Username :  ");
		l1.setBounds(100,20,100,30);

		l2 = new JLabel("Password :  ");
		l2.setBounds(100,80,100,30);

		l3 = new JLabel("DOB :  ");
		l3.setBounds(100,140,100,30);

		l4 = new JLabel("Phone Number :  ");
		l4.setBounds(100,200,100,30);

		l5 = new JLabel("Email:  ");
		l5.setBounds(100,260,100,30);



		p = new JPanel(null);
		p.setBounds(0,0,600,400);
	}

	void addComponents()
	{
		p.add(l1);
		p.add(tf1);
		p.add(l2);
		p.add(tf2);
		p.add(l3);
		p.add(tf3);
		p.add(l4);
		p.add(tf4);
		p.add(tf5);
		p.add(l5);


	}
}
```

LOGINUI.JAVA

```
import javax.swing.*;
class LoginUI
{
	JTextField tf1,tf2;
	JLabel l1,l2;
	JPanel p;

	public LoginUI()
	{
		createComponents();
		addComponents();
	}

	void createComponents()
	{
		tf1 = new JTextField();
		tf1.setBounds(250,20,200,30);

		tf2 = new JTextField();
		tf2.setBounds(250,80,200,30);


		l1 = new JLabel("Username:  ");
		l1.setBounds(100,20,100,30);

		l2 = new JLabel("Password :  ");
		l2.setBounds(100,80,100,30);


		p = new JPanel(null);
		p.setBounds(0,0,600,400);
	}

	void addComponents()
	{
		p.add(l1);
		p.add(tf1);
		p.add(l2);
		p.add(tf2);

	}
}
```

MessageUI.java

```
import javax.swing.*;

class MessageUI
{
```

```
JTextField tf1,tf2,tf3,tf4;
JLabel l1,l2,l3,l4;
JPanel p;

public MessageUI()
{
	createComponents();
	addComponents();
}

void createComponents()
{
	tf1 = new JTextField();
	tf1.setBounds(250,20,200,30);

	tf2 = new JTextField();
	tf2.setBounds(250,80,200,30);

	tf3 = new JTextField();
	tf3.setBounds(250,140,200,30);

	tf4 = new JTextField();
	tf4.setBounds(250,200,200,30);

	l1 = new JLabel("ID :  ");
	l1.setBounds(100,20,100,30);

	l2 = new JLabel("From :  ");
	l2.setBounds(100,80,100,30);

	l3 = new JLabel("To :  ");
	l3.setBounds(100,140,100,30);

	l4 = new JLabel("Type :  ");
	l4.setBounds(100,200,100,30);

	p = new JPanel(null);
	p.setBounds(0,0,600,400);
}

void addComponents()
{
	p.add(l1);
	p.add(tf1);
	p.add(l2);
	p.add(tf2);
	p.add(l3);
```

```
            p.add(tf3);
            p.add(l4);
            p.add(tf4);

            //add(p1);
      }
}
```

SpamUI.java

```
import javax.swing.*;

class SpamUI
{
      JTextField tf1,tf2,tf3;
      JLabel l1,l2,l3;
      JPanel p;

      public SpamUI()
      {
            createComponents();
            addComponents();
      }

      void createComponents()
      {
            tf1 = new JTextField();
            tf1.setBounds(250,20,200,30);

            tf2 = new JTextField();
            tf2.setBounds(250,80,200,30);

            tf3 = new JTextField();
            tf3.setBounds(250,140,200,30);


            l1 = new JLabel("Type of spam :  ");
            l1.setBounds(100,20,100,30);

            l2 = new JLabel("IS SPAM(Y/N):  ");
            l2.setBounds(100,80,100,30);

            l3 = new JLabel("ResolveID:  ");
            l3.setBounds(100,140,100,30);


            p = new JPanel(null);
            p.setBounds(0,0,600,400);
```

```
	}

	void addComponents()
	{
		p.add(l1);
		p.add(tf1);
		p.add(l2);
		p.add(tf2);
		p.add(l3);
		p.add(tf3);

		//add(p1);
	}
}
```

ResolveUI.java

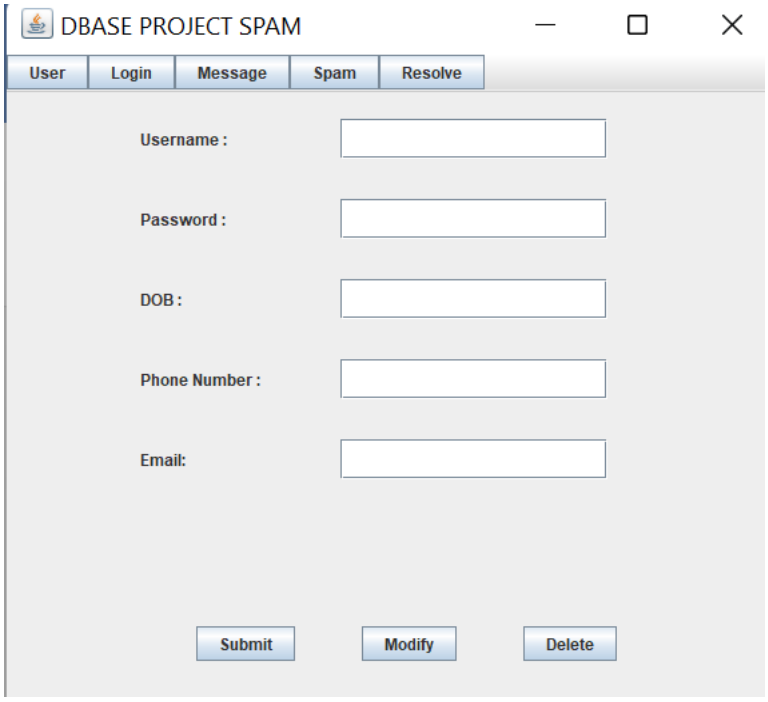
```
import javax.swing.*;
class ResolveUI
{
	JTextField tf1,tf2,tf3,tf4;
	JLabel l1,l2,l3,l4;
	JPanel p;

	public ResolveUI()
	{
		createComponents();
		addComponents();
	}

	void createComponents()
	{
		tf1 = new JTextField();
		tf1.setBounds(250,20,200,30);

		tf2 = new JTextField();
		tf2.setBounds(250,80,200,30);

		l1 = new JLabel("Resolve id:  ");
		l1.setBounds(100,20,100,30);

		l2 = new JLabel("Description:  ");
		l2.setBounds(100,80,100,30);



		p = new JPanel(null);
		p.setBounds(0,0,600,400);
```

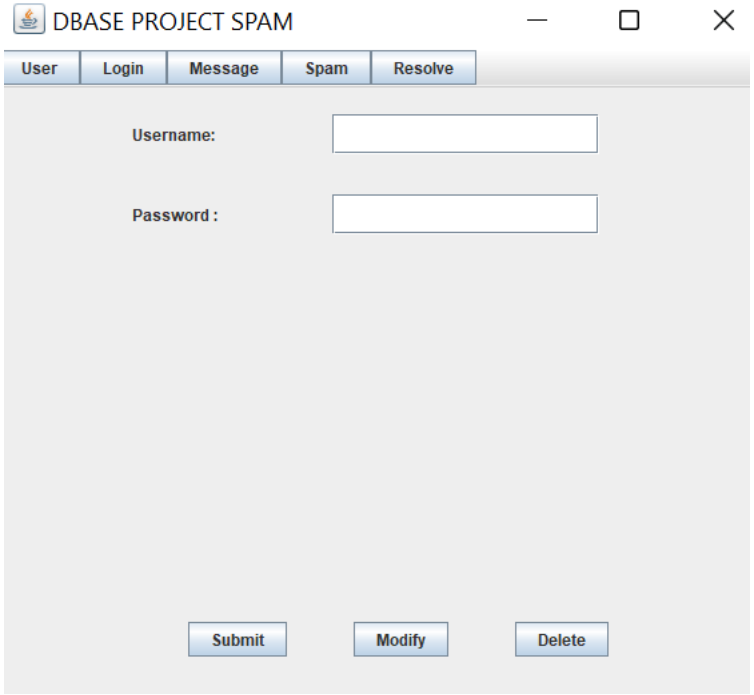
```
	}

	void addComponents()
	{
		p.add(l1);
		p.add(tf1);
		p.add(l2);
		p.add(tf2);

	}
 }
```

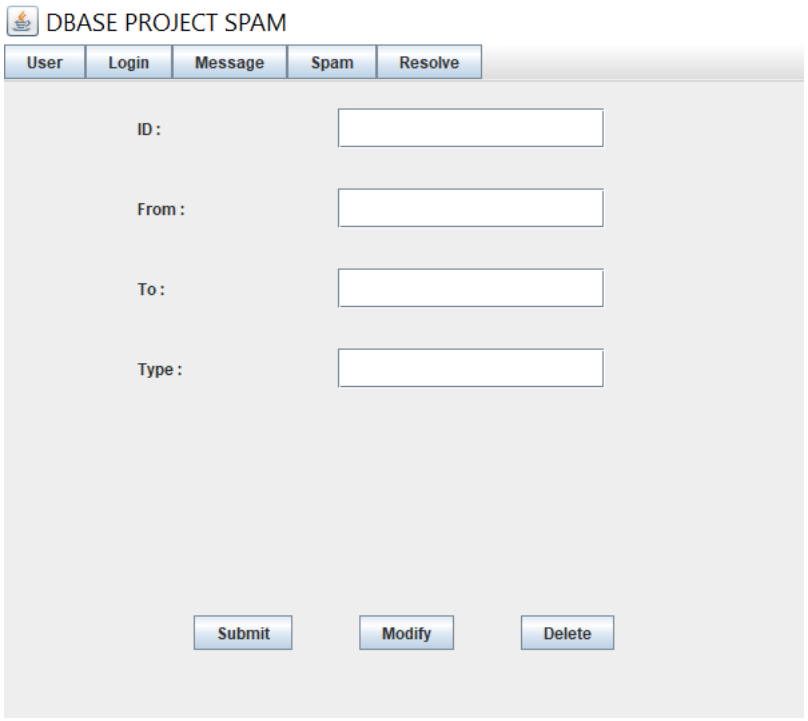
Connection Code:

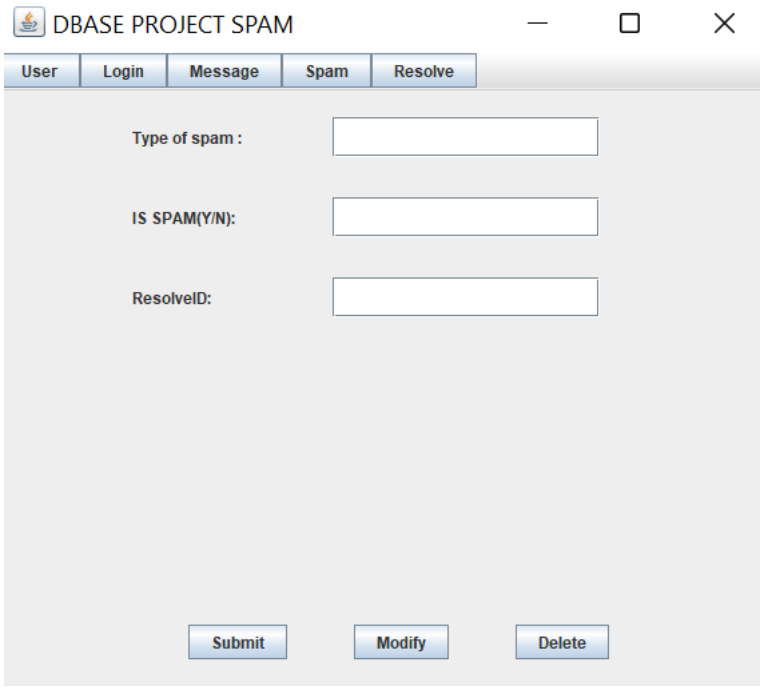
```
import java.sql.*;
public class trial{
	public static void main(String[] args){
		try{
			Class.forName("oracle.jdbc.OracleDriver");
			Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","suh
idhar","vasavi");
			Statement stmt=con.createStatement();
			ResultSet rs=stmt.executeQuery("select * from login");
			while(rs.next())
				System.out.println(rs.getInt(1)+"
"+rs.getString(2));
			con.close();
		}
		catch(Exception e){
			System.out.println(e);
		}
	}
}
```

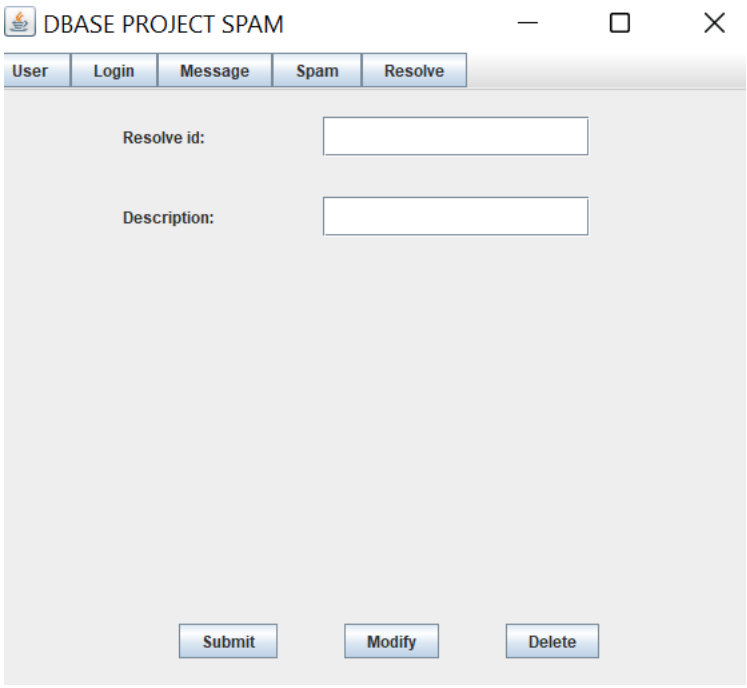
TESTING:

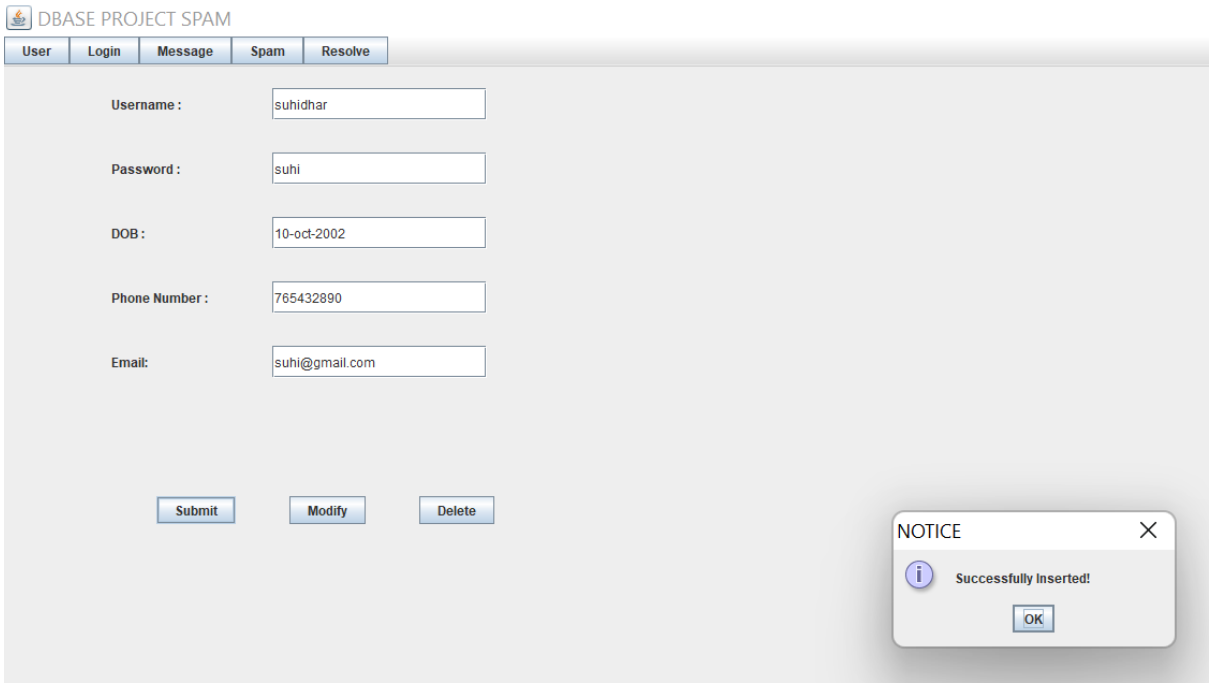

DBASE PROJECT SPAM
User
Login
Message
Spam
Resolve
Username :
Password :
DOB :
Phone Number :
Email:
Submit
Modify
Delete

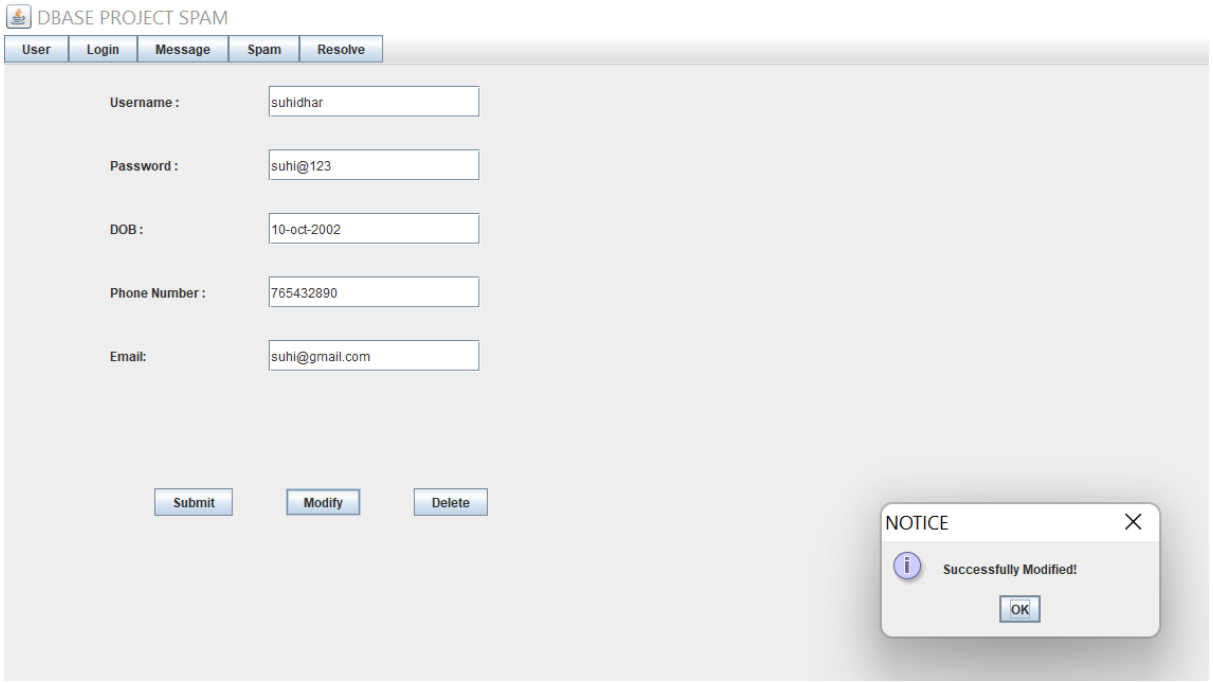

DBASE PROJECT SPAM
User
Login
Message
Spam
Resolve
Username:
Password :
Submit
Modify
Delete

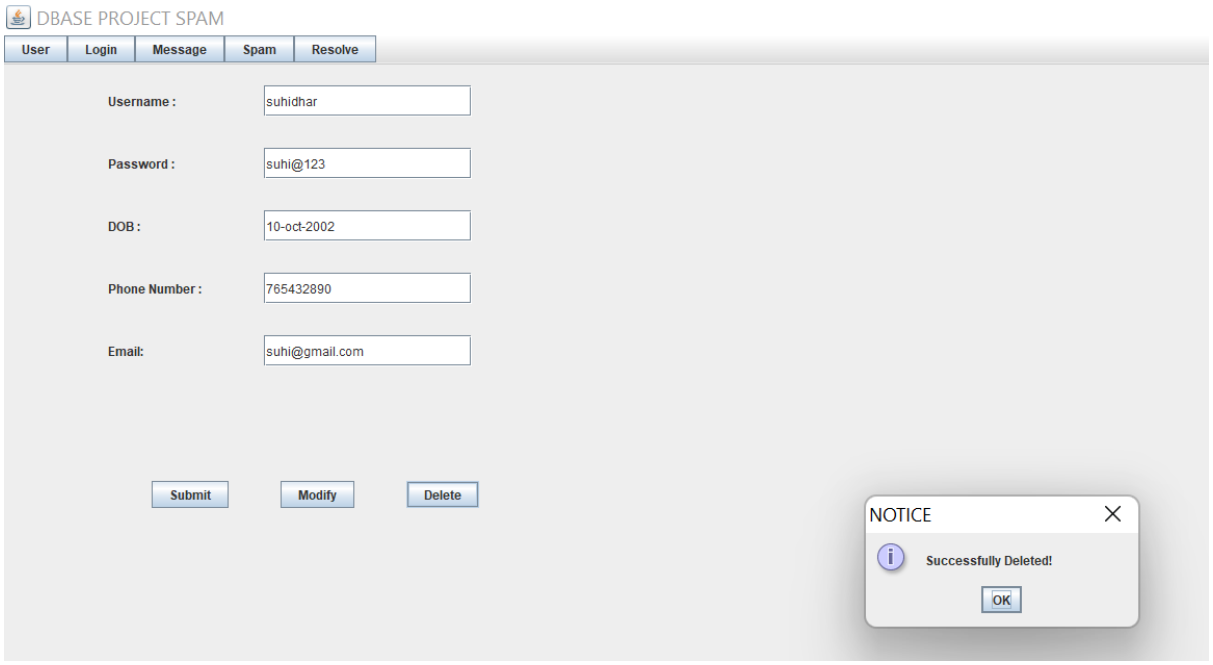

DBASE PROJECT SPAM
User
Login
Message
Spam
Resolve
ID :
From :
To :
Type :
Submit
Modify
Delete


DBASE PROJECT SPAM
User
Login
Message
Spam
Resolve
Type of spam :
IS SPAM(Y/N):
ResolveID:
Submit
Modify
Delete


DBASE PROJECT SPAM
User
Login
Message
Spam
Resolve
Resolve id:
Description:
Submit
Modify
Delete


DBASE PROJECT SPAM
User
Login
Message
Spam
Resolve
Username :
suhidhar
Password :
suhi
DOB :
10-oct-2002
Phone Number :
765432890
Email:
suhi@gmail.com
Submit
Modify
Delete
NOTICE
Successfully Inserted!
OK





Results:

I successfully completed this MINI PROJECT “DBASE PROJECT-SPAM”.

Discussion and Future work

While doing this project I got new ideas I understood how to work on projects. This project has been a huge learning curve for me as I have learnt many new technologies. I also learnt how to write queries efficiently and I have a thorough understanding of DBMS. A huge credit goes to my mentor **Ms.B.Leelavathy** whose guidance and

motivation has lead me to keep improving the project.

Now to further extend this project I want to create Android app so that this will be very handy for both students and tros.

References:

- https://www.academia.edu/36893248/Ramakrishnan_-_Database_Management_Systems_3rd_Edition
- https://docs.oracle.com/javase/7/docs/index.html
- https://www.javatpoint.com/dbms-tutorial
- http://www.sqlines.com/articles/java/sql_server_jdbc_connection