INDEX

Java Front-End Overview

The Front End Application controls between end-users, the Translator and possibly other applications (for e.g. ,a form repository , workflow application, and so on) you can also use the Front End Application to populate a form with data before presenting the form to a user, and to extract data from a form after the user submits it.

1) For software applications, front end is the same as user interface.
2) In client/server applications, the client part of the program is often called the front end and the server part is called the back end.

User interface design is the design of software applications, and websites with the focus on the user's experience and interaction as simple and efficient as possible, in terms of accomplishing user goals- what is often called user-centered design (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs

A good user interface needs to have following features:

1>**CLEAR:** Clarity is the most important element of user interface design. Indeed, the whole purpose of user interface design is to enable people to interact with your system by communicating meaning and function. If people can't figure out how your application works or where to go on your website they will get confused and frustrated.

2>**CONCISE:** Clarity in a user interface is to keep things clear but also keep things concise. When you can explain a feature in a sentence instead of three, do it. When you can label an item with one word instead of two, do it. Save the valuable time of users by keeping things concise.

3>**FAMILIAR:** Familiar is just that: Something which appears like something else you have encountered before. When you are familiar with something, you know how it behaves_ you know what to expect. Identify things that are familiar to your users and integrate that into your user interface.

4>**RESPONSIVE:** Responsive means a couple of things. First of all, responsive means fast. The interface, if not the software behind it, should work fast. Waiting for things to load and using laggy and slow interface is frustrating. Seeing things load quickly, or at the very least an interface that loads quickly (even if the content is yet to catch up) improves the user experience. Responsive also means the interface provides some form of feedback.

5>CONSISTENT: Consistent interfaces allow users to develop usage patterns-they will learn what the different buttons, tabs, icons and other interface elements look like and will recognize them and realize what they do min different contexts. They will also learn how certain things work, will be able to work out how to operate new features quicker, extrapolating from those extra experiences.

6>ATTRACTIVE: A good interface should be attractive. Attractive in a sense that it makes the use of that interface enjoyable when your software is pleasant to use, your costumer or staff will not simply be using it they will look forward to using it. There are of course many different types of software and websites, all produced for different markets and audiences. What looks 'good' for anyone particular audience will vary. This means that you should fashion the look and feel if your interface for your audience.

7>EFFICIENT: A user interface is vehicle that takes you places. Those places are the different functions of the software application or websites. A good interface should allow you to perform those functions faster and less effort. Now, 'efficient' sounds like a fairly vague attribute-if you combine all of the other things on this list, surely the interface will end up being efficient. What you really need to do make an interface efficient is to figure out what exactly the user is trying to achieve, and then let them do exactly that without any fuss.

BACK END Overview

A back-end is a database that is accessed by users indirectly through an external application rather than by application programming stored within the database itself or by low level manipulation of the data (e.g. through MySQL command).

A back-end is a database stores data but does not include end-user application elements such as stored queries, forms, macros or reports.

A database which is fronted by a web server and can be accessed by browser that connected by browsers that connects into the server. Such database are usually employed in e-commerce applications:
For examples, online bookstores details of books on a back end database which can be browsed by users looking for a specific book. Increasingly mainstream database management system provide facilities for database to be easily connected to a Web server.

Introduction to Java net beans

Java is a popular 3rd-Generation programming language, which can be used to perform any of the thousands of things that a computer can do. With the features it offers, Java has become the language of choice for Internet and Intranet applications. Java plays an important role for the proper functioning of many software-based devices attached to a network. The kind of functionality the Java offers, has contributed a lot towards the popularity of Java.

A computer is a set of instructions given to computer. These instructions initiate some action and hence sometimes called executable instructions. In java programs, the executable instructions are specified through methods or a function is a sequence of some declaration statements and executable statements. In other programming languages, methods are known as functions, sometimes procedures, subprograms or subroutines.

In Java, which is strictly object-oriented any action can take place through methods and methods have to exist as a part of a class.

Introduction to MySQL

A database system is basically a computer based record keeping system. The collection of data, usually referred to as the database, contains ion about one particular enterprise. In a typical file-processing system, permanent records stored in various files. A number of different application programs are written to extract records from files and add records to the appropriate files. But this scheme has a number of major limitations and disadvantages, such as data redundancy (duplication of data), data inconsistency, unsharable data unstandardized data insecure data, incorrect data etc. A database management system is answer to all these problems

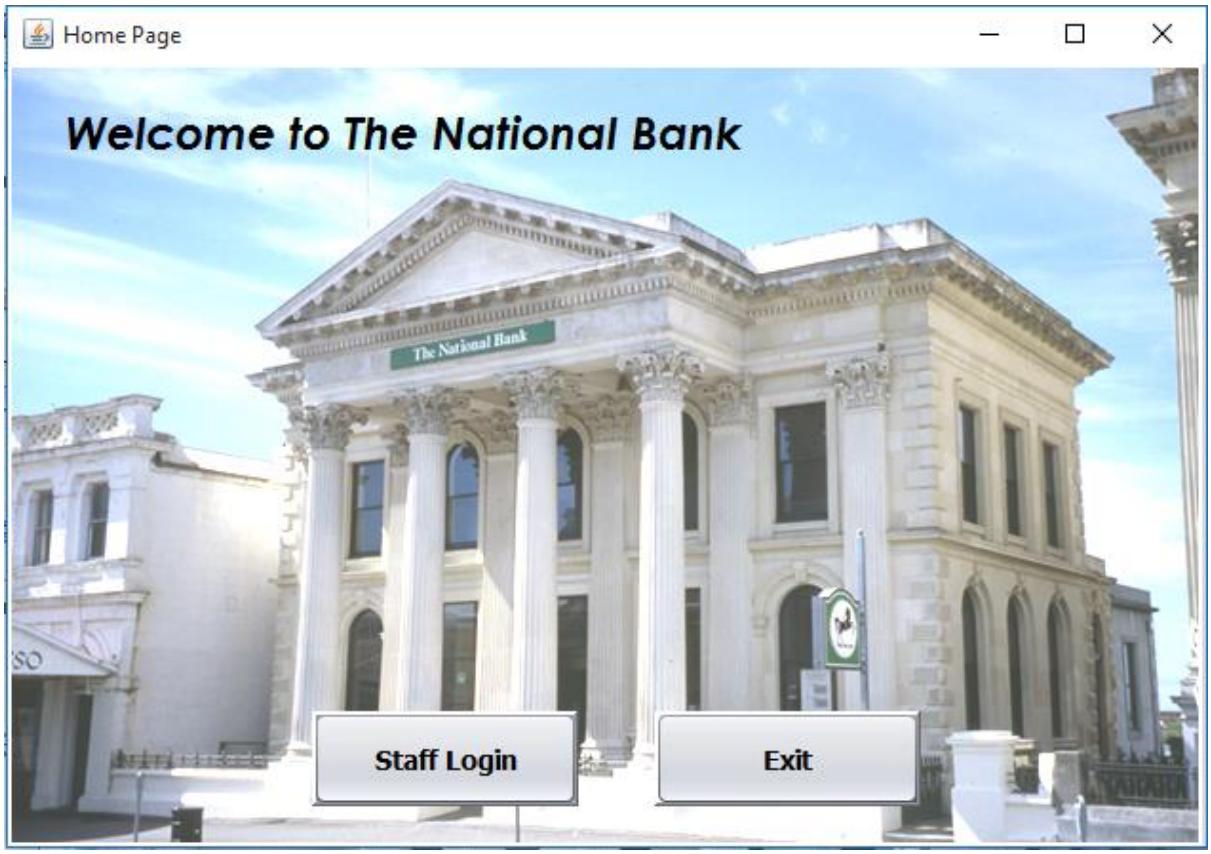
As it provides a centralized control of the data.

Various advantages of database systems are:

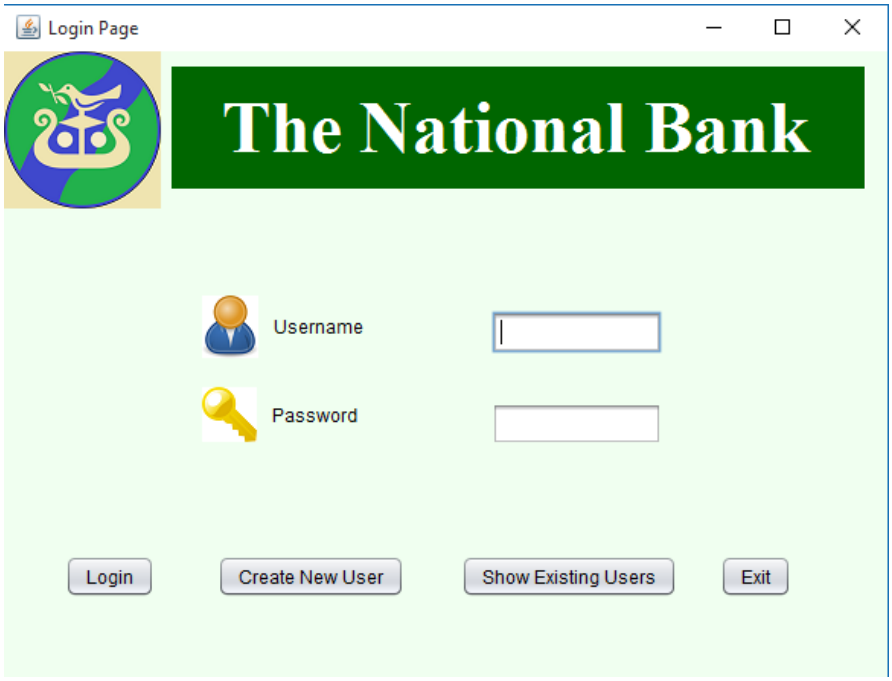
- Database systems reduce data redundancy (data duplication) to a large extent.
- Database systems control data inconsistency to a large extent.
- Database facilitate sharing of data.
- Database enforce standards.
- Centralized database can ensure data security.
- Integrity can be maintained through database.

Snapshots of Java form

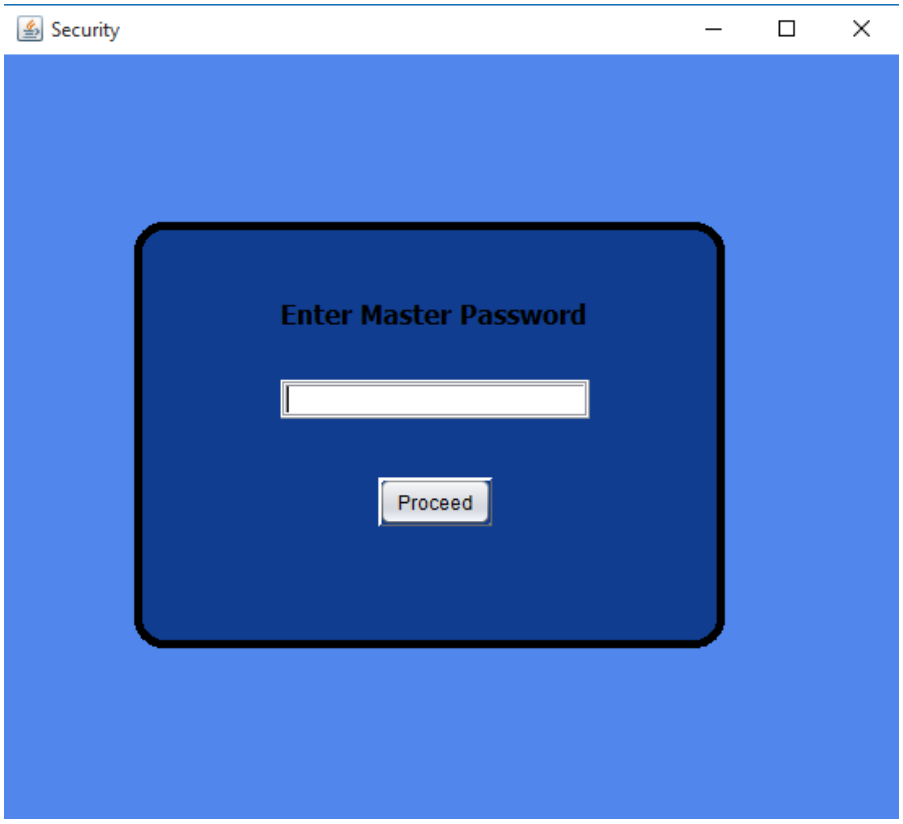
Home Page:



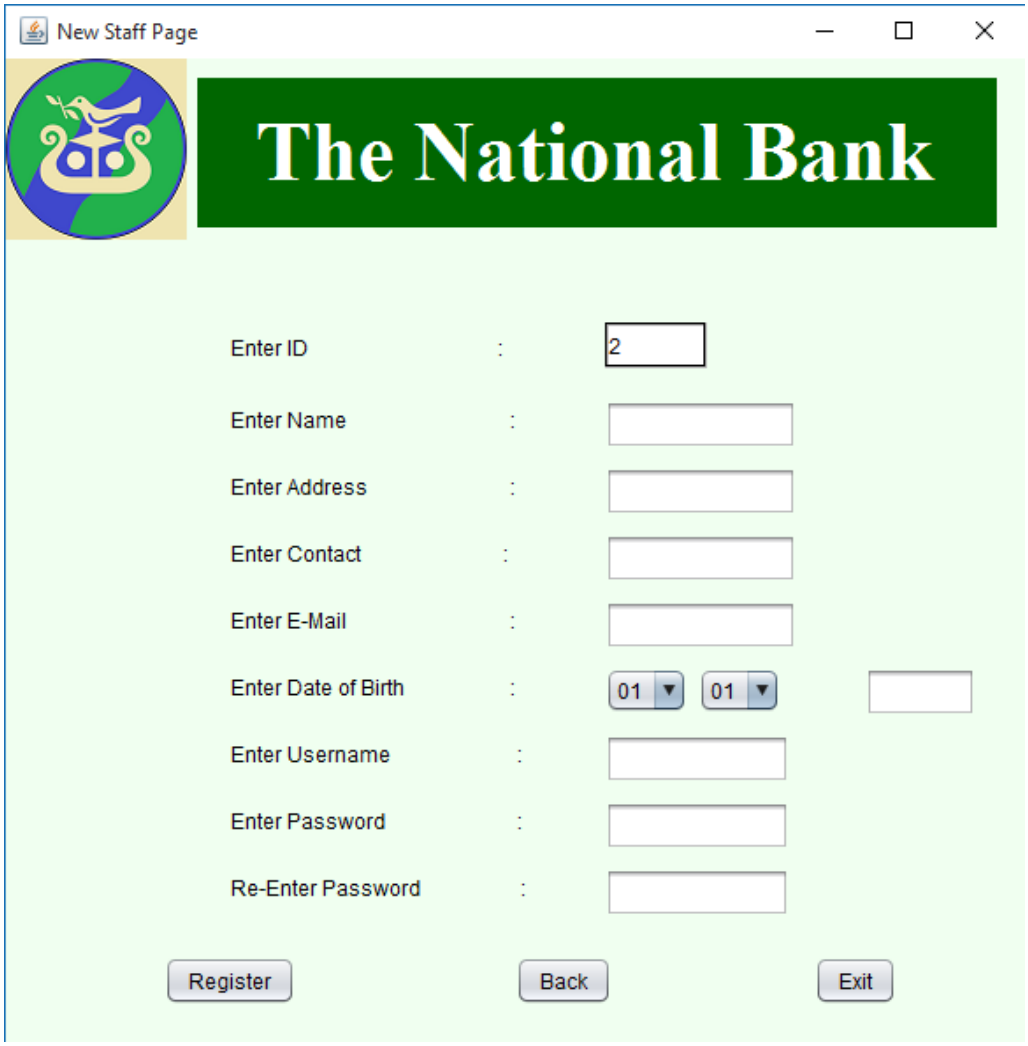
Staff Login Page:



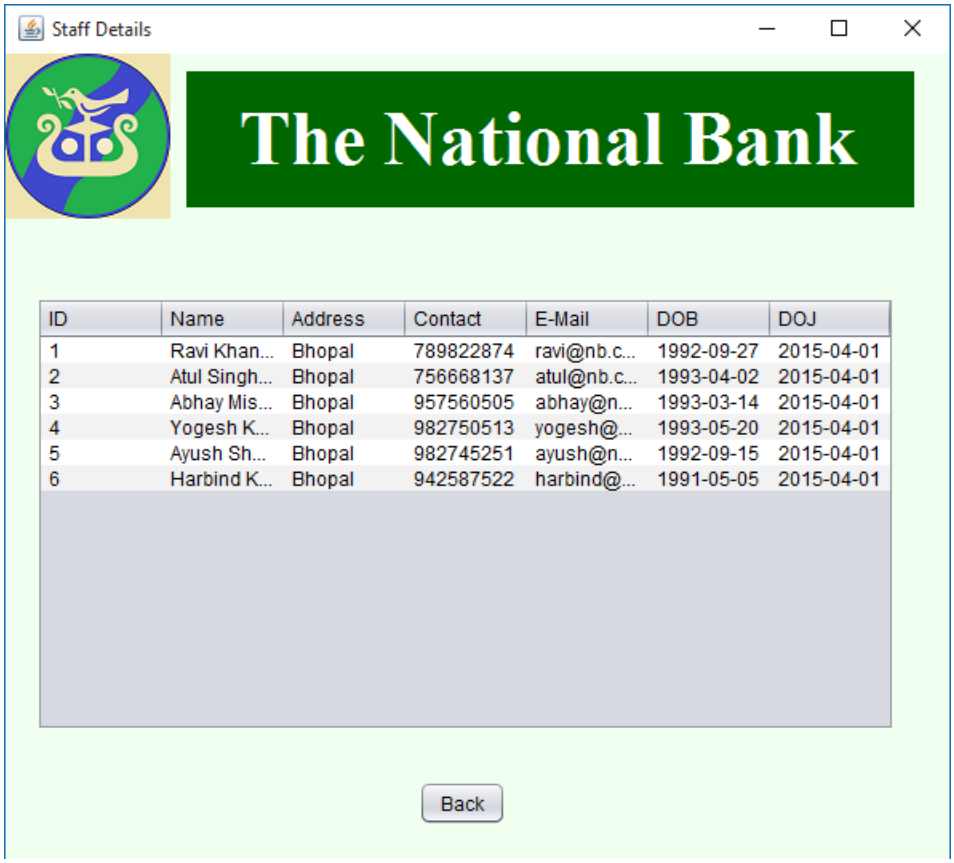
Security Page:



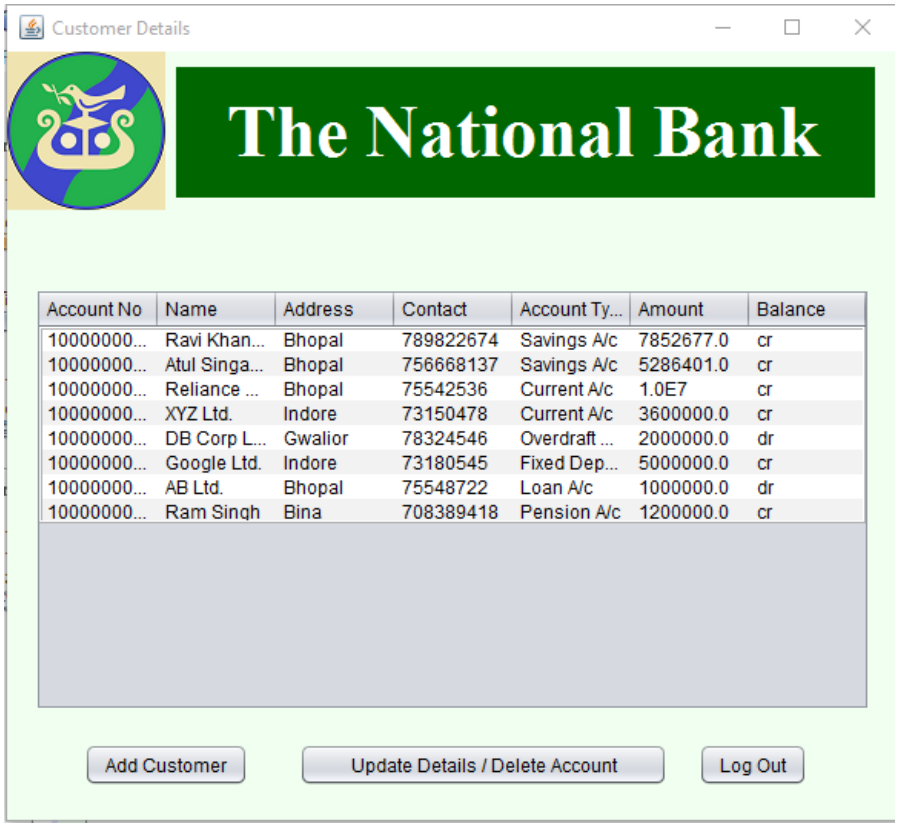
New Staff Page:



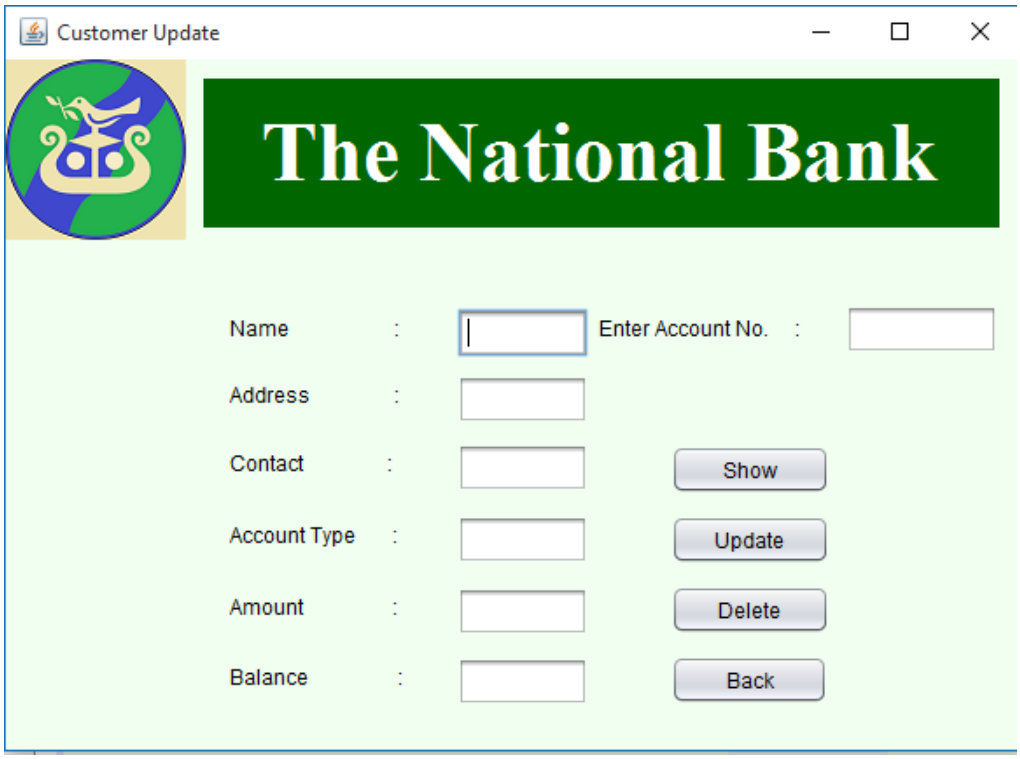
Staff Details Page:



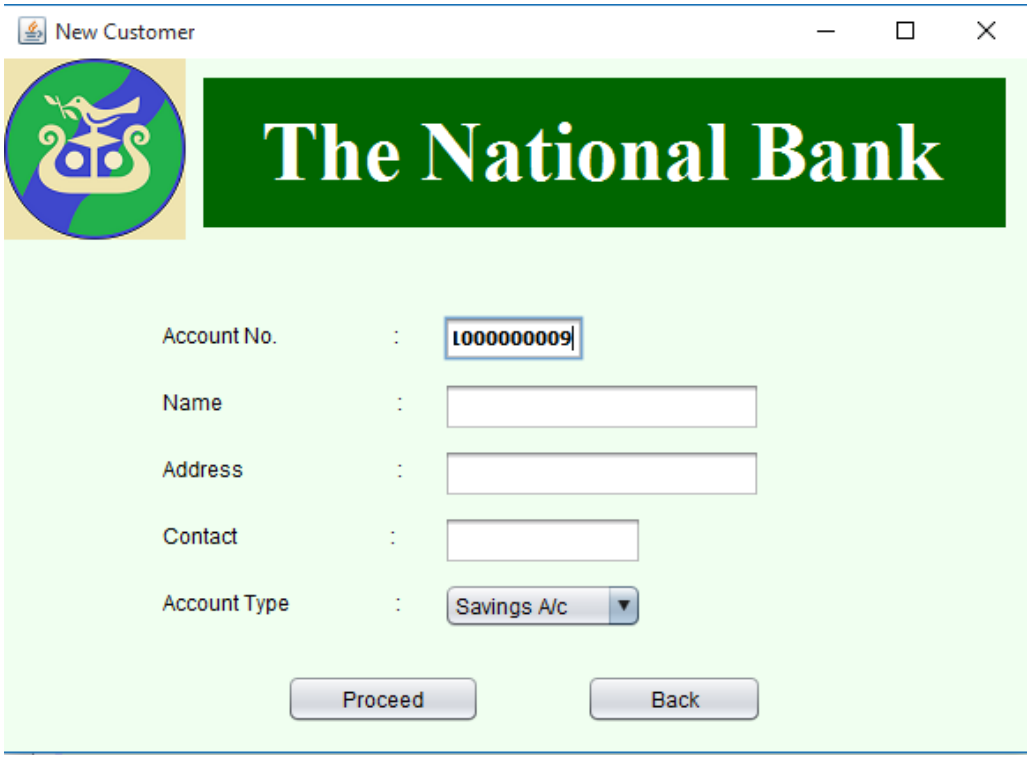
Customer Details Page:



Customer Update Page:



New Customer Page:



Coding of Java Net-Beans

Classes Imported:

```
import java.sql.*;

import javax.swing.JOptionPane;
```

Home Page:

Staff Login Button-

```
this.dispose();

new Staff_Login().setVisible(true);
```

Exit Button-

```
System.exit(0);
```

Staff Login Page:

Login Button-

```
String user,pass,user1 = null,pass1 = null;

    int x=0;

    user=jTextField1.getText();
```

```
        pass=jPasswordField1.getText();
        try{
            Class.forName("java.sql.Driver");
            Connection
conn=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/b
ank","root"," ");
            Statement s=conn.createStatement();
            String q="Select * from staff_login";
            ResultSet r=s.executeQuery(q);
            while (r.next())
            {
                user1=r.getString(1);
                pass1=r.getString(2);

                if((user.equals(user1))&&(pass.equals(pass1)))
                {
                    x=1;
                }
            }
            if(x==1)
            {
                this.dispose();
                new Staff_Security().setVisible(true);
            }
```

```
            else  JOptionPane.showMessageDialog(null,"Wrong  Username/Password.
Please try again!!!");
            r.close();
            s.close();
            conn.close();
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(null,e.getMessage());
        }
```

Create User Button-

```
this.dispose();
new Staff_New().setVisible(true);
```

Show Existing User Button-

```
String pass=JOptionPane.showInputDialog(null, "Enter Manager's Password");
        if(pass.equals("$$$$"))
        {
            this.dispose();
            new Staff_Details().setVisible(true);
        }
```

Exit Button-

```
System.exit(0);
```

Security Page:

Proceed Button-

```
String a=jPasswordField1.getText();
        if((a.equals("thenationalbank")) || (a.equals("  ")))
        {
            this.dispose();
            new Cust_Details().setVisible(true);
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Wrong Password Entered and Program will Terminate Now!!!");
            this.dispose();
        }
```

New Staff Page:

Register Button-

```
int id;
        String name,address,email,dd,mm,yy,user,pass,re_pass,contact;
        id=Integer.parseInt(jTextField1.getText());
        name=jTextField2.getText();
        address=jTextField3.getText();
        contact=jTextField4.getText();
```

```
        email=jTextField5.getText();
        user=jTextField7.getText();
        pass=jPasswordField1.getText();
        re_pass=jPasswordField2.getText();
        dd=jComboBox1.getSelectedItem().toString();
        mm=jComboBox2.getSelectedItem().toString();
        yy=jTextField6.getText();
        String date=yy+"-"+mm+"-"+dd;
        try{
            Class.forName("java.sql.Driver");
            Connection
conn=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/b
ank","root"," ");
            Statement s=conn.createStatement();
            if(pass.equals(re_pass))
            {
                String q="insert into staff_details
values("+id+",'"+name+"','"+address+"',"+contact+",'"+email+"','"+date+"','2015-
11-12')";
                String q1="insert into staff_login values('"+user+"','"+pass+"')";
                int x=s.executeUpdate(q);
                int x1=s.executeUpdate(q1);
                if((x>0)&&(x1>0))
                {
                    JOptionPane.showMessageDialog(null,"Record Added");
```

```
            }
        }
        else JOptionPane.showMessageDialog(null,"Please Check Password");
        s.close();
        conn.close();
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
```

Back Button-

```
this.dispose();
new Staff_Login().setVisible(true);
```

Exit Button-

```
System.exit(0);
```

formWindowOpened-

```
try{
        Class.forName("java.sql.Driver");
        Connection
conn=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/b
ank","root"," ");
```

```
        Statement s=conn.createStatement();
        String q="select max(id) from staff_details";
        ResultSet r=s.executeQuery(q);
        if(r.next())
        {
            int i=r.getInt(1);
            i++;
            jTextField1.setText(""+i);
        }
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
```

Staff Details Page:

Back Button-

```
this.dispose();
new Staff_Login().setVisible(true);
```

formWindowOpened-

```
int contact,id;
    String name,address, dob, doj, email;
```

```
        Double amt;
        DefaultTableModel t=(DefaultTableModel)jTable1.getModel();
        try{
            Class.forName("java.sql.Driver");
            Connection
conn=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/b
ank","root"," ");
            Statement s=conn.createStatement();
            String q="select * from staff_details";
            ResultSet r=s.executeQuery(q);
            while(r.next())
            {
                id=r.getInt(1);
                name=r.getString(2);
                address=r.getString(3);
                contact=r.getInt(4);
                email=r.getString(5);
                dob=r.getString(6);
                doj=r.getString(7);
                t.addRow(new Object[]{id,name,address,contact,email,dob,doj});
            }
            jTable1.setModel(t);
            r.close();
            s.close();
```

```
            conn.close();
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
```

Customer Details Page:

Add Customer Button-

```
this.dispose();
new Cust_Add().setVisible(true);
```

Update Details/Delete Account Button-

```
this.dispose();
new Cust_Update().setVisible(true);
```

Logout Button-

```
JOptionPane.showMessageDialog(null, "You had Successfully Logged Out!!!");
this.dispose();
new Staff_Login().setVisible(true);
```

Customer Update Page:

Show Button-

```
int acc,contact = 0;
        String name = null,address = null,acc_type = null,balance = null;
        Double amt = null;
        acc=Integer.parseInt(jTextField7.getText());
        try{
            Class.forName("java.sql.Driver");
            Connection
conn=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/b
ank","root"," ");
            Statement s=conn.createStatement();
            String q="select * from cust_details where acc_no="+acc;
            ResultSet r=s.executeQuery(q);
            while (r.next())
            {
                name=r.getString(2);
                address=r.getString(3);
                contact=r.getInt(4);
                acc_type=r.getString(5);
                amt=r.getDouble(6);
                balance=r.getString(7);
            }
            jTextField1.setText(""+name);
```

```
            jTextField2.setText(""+address);
            jTextField3.setText(""+contact);
            jTextField4.setText(""+acc_type);
            jTextField5.setText(""+amt);
            jTextField6.setText(""+balance);
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
```

Update Button-

```
int acc,contact = 0;
        String name = null,address = null,acc_type = null,balance = null;
        Double amt = null;
        acc=Integer.parseInt(jTextField7.getText());
        name=jTextField1.getText();
        address=jTextField2.getText();
        contact=Integer.parseInt(jTextField3.getText());
        acc_type=jTextField4.getText();
        amt=Double.parseDouble(jTextField5.getText());
        balance=jTextField6.getText();
        try{
            Class.forName("java.sql.Driver");
```

```
        Connection
conn=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/b
ank","root"," ");

        Statement s=conn.createStatement();

        String q="update cust_details set
name='"+name+"',address='"+address+"',contact_no="+contact+",acc_type='"+ac
c_type+"',amount="+amt+",balance='"+balance+"' where acc_no="+acc;

        int x=s.executeUpdate(q);

        if(x==1)

        {

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

        }

        else JOptionPane.showMessageDialog(null, "ERROR!!!");

    }

    catch (Exception e)

    {

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

    }
```

Delete Button-

```
int acc,contact = 0;

    String name = null,address = null,acc_type = null,balance = null;

    Double amt = null;

    acc=Integer.parseInt(jTextField7.getText());

    name=jTextField1.getText();
```

```
        address=jTextField2.getText();
        contact=Integer.parseInt(jTextField3.getText());
        acc_type=jTextField4.getText();
        amt=Double.parseDouble(jTextField5.getText());
        balance=jTextField6.getText();
        try{
            Class.forName("java.sql.Driver");
            Connection
conn=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/b
ank","root"," ");
            Statement s=conn.createStatement();
            String q="update cust_details set
name='"+name+"',address='"+address+"',contact_no="+contact+",acc_type='"+ac
c_type+"',amount="+amt+",balance='"+balance+"' where acc_no="+acc;
            int x=s.executeUpdate(q);
            if(x==1)
            {
                JOptionPane.showMessageDialog(null,"Record Updated!!!");
            }
            else JOptionPane.showMessageDialog(null, "ERROR!!!");
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
```

Back Button-

```
this.dispose();

new Cust_Details().setVisible(true);
```

Customer Add Page:

Proceed Button-

```
int acc=Integer.parseInt(jTextField1.getText());

        String name=jTextField2.getText();

        String address=jTextField3.getText();

        int contact=Integer.parseInt(jTextField4.getText());

        String acc_type=(String)jComboBox1.getSelectedItem();

        try{

            Class.forName("java.sql.Driver");

            Connection
conn=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/b
ank","root"," ");

            Statement s=conn.createStatement();

            String q="insert into cust_details
values("+acc+",'"+name+"','"+address+"',"+contact+",'"+acc_type+"',null,null)";

            int x=s.executeUpdate(q);

            if(x>0)

               {

                   JOptionPane.showMessageDialog(null,"Record Added");
```

```
            }
        s.close();
        conn.close();
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e.getMessage());
    }
```

Back Button-

```
this.dispose();
new Cust_Details().setVisible(true);
```

formWindowOpened-

```
try{
        Class.forName("java.sql.Driver");
        Connection
conn=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/b
ank","root"," ");
        Statement s=conn.createStatement();
        String q="select max(acc_no) from cust_details";
        ResultSet r=s.executeQuery(q);
        if(r.next())
        {
```

```
        int i=r.getInt(1);
        i++;
        jTextField1.setText(""+i);
    }
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
}
```

Coding of MY SQL

Creating Tables:

```
create table staff_details
 (id int,
 name varchar(20),
 address varchar(30),
 contact int(11),
 email varchar(25),
 dob date,
 doj date);


create table staff_login
(username varchar(25),
Password varchar(25));


create table cust_details
(acc_no int(10),
Name varchar(25),
Address varchar(35),
contact_no int(11),
acc_type varchar(20),
amount double(15,2),
balance char(2));
```

Structure of Tables:

Desc staff_details;

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI		
name	varchar(40)	YES		NULL	
address	varchar(40)	YES		NULL	
contact	int(11)	YES		NULL	
email	varchar(35)	YES		NULL	
dob	date	YES		NULL	
doj	date	YES		NULL	

Desc staff_login;

Field	Type	Null	Key	Default	Extra
username	varchar(25)	YES		NULL	
Password	varchar(25)	YES		NULL	

Desc cust_details;

Field	Type	Null	Key	Default	Extra
acc_no	int(10)	YES		NULL	
Name	varchar(25)	YES		NULL	
Address	varchar(35)	YES		NULL	
contact_no	int(11)	YES		NULL	
acc_type	varchar(20)	YES		NULL	
amount	double(15,2)	YES		NULL	
balance	char(2)	YES		NULL	

Inserting Values in Staff_Details:

```
insert into staff_details

values(1, "Ravi Khandelwal", "Bhopal","789822874","ravi@nb.com", "1992-09-27", "2015-04-01");
```

```
insert into staff_details

values(2, "Atul Singhasiya", "Bhopal","756668137","atul@nb.com", "1993-04-
02", "2015-04-01");


insert into staff_details

values(3, "Abhay Mishra", "Bhopal","957560505","abhay@nb.com", "1993-03-
14", "2015-04-01");


insert into staff_details

values(4, "Yogesh Kene", "Bhopal","982750513","yogesh@nb.com", "1993-05-
20", "2015-04-01");


insert into staff_details

values(5, "Ayush Sharma", "Bhopal","982745251","ayush@nb.com", "1992-09-
15", "2015-04-01");


insert into staff_details

values(6, "Harbind Kaur", "Bhopal","942587522","harbind@nb.com", "1991-05-
05", "2015-04-01");
```

Inserting Values in Staff_Login:

```
insert into staff_login

values("ravi","ravi");


insert into staff_login

values("atul","atul");
```

```
insert into staff_login
values("abhay","abhay");

insert into staff_login
values("yogesh","yogesh");

insert into staff_login
values("ayush","ayush");

insert into staff_login
values("harbind","harbind");
```

Inserting Values in Cust_Details:

```
insert into cust_details
values(1000000001,"Ravi Khandelwal","Bhopal", 789822674,"Saving A/c",
7852677,"cr");

insert into cust_details
values(1000000002," Atul Singahsiya","Bhopal", 756668137,"Saving A/c",
5286401,"cr");

insert into cust_details
values(1000000003," Reliance Retail Ltd.","Bhopal", 075542536,"Current
A/c", 10000000,"cr");
```

```
insert into cust_details

values(1000000004," XYZ Ltd.","Indore", 073150478,"Current A/c",
3600000,"cr");


insert into cust_details

values(1000000005," DB Corp Ltd.," Gwalior", 078324546," Overdraft A/c",
2000000,"dr");


insert into cust_details

values(1000000006," Google Ltd.," Indore", 073180545," Fixed Deposit",
5000000,"cr");


insert into cust_details

values(1000000007," AB Ltd.," Bhopal", 0 75548722," Loan A/c", 1000000,"dr");


insert into cust_details

values(1000000008,"Ram Singh","Bina",708389418,"Pension
A/c",1200000,"cr");


select *

from staff_details;
```

id	name	address	contact	email	dob	doj
1	Ravi Khandelwal	Bhopal	789822874	ravi@nb.com	1992-09-27	2015-04-01
2	Atul Singhasiya	Bhopal	756668137	atul@nb.com	1993-04-02	2015-04-01
3	Abhay Mishra	Bhopal	957560505	abhay@nb.com	1993-03-14	2015-04-01
4	Yogesh Kene	Bhopal	982750513	yogesh@nb.com	1993-05-20	2015-04-01
5	Ayush Sharma	Bhopal	982745251	ayush@nb.com	1992-09-15	2015-04-01
6	Harbind Kaur	Bhopal	942587522	harbind@nb.com	1991-05-05	2015-04-01

```
Select *

From staff_login;
```

username	Password
a	a
ravi	1
ravi	ravi
atul	atul
ravi	ravi
atul	atul
abhay	abhay
yogesh	yogesh
ayush	ayush
harbind	harbind

```
Select *

From cust_details;
```

acc_no	Name	Address	contact_no	acc_type	amount	balance
1000000001	Ravi Khandelwal	Bhopal	789822674	Savings A/c	7852677.00	cr
1000000002	Atul Singahsiya	Bhopal	756668137	Savings A/c	5286401.00	cr
1000000003	Reliance Retail Ltd.	Bhopal	75542536	Current A/c	10000000.00	cr
1000000004	XYZ Ltd.	Indore	73150478	Current A/c	3600000.00	cr
1000000005	DB Corp Ltd.	Gwalior	78324546	Overdraft A/c	2000000.00	dr
1000000006	Google Ltd.	Indore	73180545	Fixed Deposit	5000000.00	cr
1000000007	AB Ltd.	Bhopal	75548722	Loan A/c	1000000.00	dr
1000000008	Ram Singh	Bina	708389418	Pension A/c	1200000.00	cr

Conclusion

The making of this project is a quite interesting and acknowledgeable approach to Java Database Connectivity for us. This project can be instrumental in Bank Management.

The project **Bank Management System (BMS)** is for computerizing the working in a bank. The software takes care of all the requirements of a bank and is capable to provide easy and effective storage of information related to customers that have account in the bank.

This software deals with opening of an account, managing them by updating or deleting them and it shows record of the customers having an account in the bank. Details of staff can also be viewed but it requires a special password that is only known to the manager of bank. There is no frame for removing the staff members as it is only with the manager to remove them by use of database.

Thus this SOFTWARE is extremely useful in saving the time and efforts of users.