

JAVA PROJECT

ONLINE BANKING SYSTEM



# **SUBMITTED BY :**

### K.SAI KRISHNA-319126511031

### N.SHANMUKH TEJA-319126511040

### P.SAI VIVEK-319126511046

### S.PRASANTH-319126511049

### S.REHAMAN-319126511053

ACKNOWLEDGEMENTS

We express the sincere gratitude to our principal for his administration towards our academic growth. We express sincere gratitude to our Coordinator and HOD-IT for her leadership and constant motivation, provided in successful completion of our academic semester.

We record it as our privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality. We express our sincere thanks to our JAVA madam BHANU SRI for her novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully. We felt very happy for working under her guidance.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

Project title:

Online Banking System

Purpose:

The typical way to retain the data of the customer in a bank was to enter and Register the details. Any time a consumer has to make any transactions, he must go to the bank and take the required steps, which might not be possible all the time. It can also be a hard-hitting job for users and bankers. The project offers a real-life understanding of the Online Banking Environment and the tasks undertaken by the different positions in the supply chain. Here, we are offering automation for the banking system over the Internet. The Online Banking Framework project gathers tasks undertaken by various positions in real-life banking and offers improved strategies to keep the details needed up-to-date, resulting in productivity. The project offers a real-life understanding of the Online Banking Environment and the tasks undertaken by the different positions in the supply chain.

Scope of the Project:

This project explores the entry threshold for the provision of a new transaction processing channel using a real options strategy, where the entry threshold is determined using an Internet banking framework built for the use of regular users (individuals), Industrialists, Manufacturers, Educational Institutions (Financial Sections), Organizations and Academicians under transaction rate uncertainty.

• The customer must have a correct user Id and password to log in to your device.

• If the incorrect password is given three times in succession, the account will be locked and the user will not be able to access it. If an incorrect password is used, the user will be alerted that his account will be closed.

• When a legitimate user logs in, the number of accounts he has with the bank will be revealed.

• When choosing the desired account, it is taken to a page that displays the existing balance of that particular account number.

• Users may request details of the last 'n' number of transactions that they have carried out.

• The study can also be taken from this.

• Users may move funds to another account in the same bank. The use giyen a transaction password that is separate from the login password.

• The user can move funds from his account to any other bank account. If the transaction is successful, a note should show to the customer, if the transaction is unsuccessful, a valid message should be given to the customer as to why the transaction failed.

• Users can order the checkbook/address change/stop payment of the check.

• Users will display both their monthly and annual statements. He may even take the printout of the same item.

• Generate reports for each segment

• The administrator should take thebackup of the database periodically for any instance that occurs.

• Both customers are authenticated to access the services

• The FAQ portion is also included for the support of end-users. Definitions, Acronyms, and Abbreviations

Administrator: The super user who can add new customers to the banking system and assigns the required username, password, type of account, and other information. If any customer cancels their bank account, their account can be removed and transactions can be stopped immediately.

Customers: After logging in customer can request for balance inquiry in his account, Funds Transfer to another account in the same bank, Requests for checkbook/change of address/stop payment(Viewing Monthly and annual statements)

Project module:

The following modules are included in the Online Banking System :

1) Login Process: This module helps legitimate customers to access the functions offered by the bank.

2) Balance Enquiry: This module preserves the knowledge of the balance of a given account.

3) Upgrade Profile: This module helps the user to update their account profile.

4) Movement of Money: This module enables consumers to transfer funds from one account to another within the same bank.

5) Password update: this module helps consumers to change their passwords.

6) Mini Statements: This module helps consumers to access descriptions of their purchases

Future Scope of the Project:

The new framework is being developed as a software application. In the future, we would like to improve it for handheld devices such as mobile phones, wap, or GPRS links.

### JCOMPONENTS :

### 

### Java JLabel : The object of JLabel class is a component for placing text in a container. It is usedto display a single line of read only text. The text can be changed by anapplication but a user cannot edit it directly. It inherits JComponent class.

### Java JTextField : The object of a JTextField class is a text component that allows the editing of asingle line text. It inherits JTextComponent class

### Java JFrame : The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.

### Java JButton : The JButton class is used to create a labeled button that has platformindependent implementation. The application result in some action when thebutton is pushed. It inherits Abstract Button class.

JAVA LISTENER INTERFACES:

### Java ActionListener Interface : The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. The ActionListener interface is found in java.awt.event package. It has only one method: actionPerformed()

### ActionPerformed() method : The actionPerformed() method is invoked automatically whenever you click on registered.

ALGORITHM

Step 1 : Create a JFrame.

Step 2 : Create a JTextfield.

Step 3 : Create a JLabel for providing input of subjects.

Step 4 : Create JLabels for asking the feedback questions.

Step 5 : Create Jbuttons for choosing and submitting the feedback.

Step 6 : Create JComboboxes for the subject to be selected and the teacher to be given feedback.

Step 7 : Create JRadiobuttons for selecting the options of the feedback questions.

Step 8 : Give setbounds for all the created Jlabels , JTextfield , JButtons and JComboboxes.

Step 9 : After the completion of giving feedback a message should be displayed.

CODE:

Here we divide our code into 6 parts :

1.O\_bank : this is the main interface of the bank which contains the login form.

2.looking for : this is the interface which redirects us to the required operation.

3.deposit : here entered will be deposited into the account.

4.withdraw : here entered amount will be withdrawn from account.

5.balance : this gives a pop up message showing the account balance.

6.mini statement : this gives the list of transactions done by a user.

**O\_Bank code:**

package banking;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

public class O\_bank {

JFrame f= new JFrame("BANK LOGIN");

public O\_bank()

{

JLabel l1=new JLabel("USERNAME:-");

l1.setBounds(50,100,100,30);

JTextField t1=new JTextField("");

t1.setBounds(150,100,200,30);

JLabel l2=new JLabel("PASSWORD:-");

l2.setBounds(50,150,100,30);

JPasswordField p1=new JPasswordField("");

p1.setBounds(150,150,200,30);

JButton b1=new JButton("LOGIN");

b1.setBounds(175,250,100,30);

b1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

if(t1.getText().equals(""))

{

JOptionPane.showMessageDialog(null,"username cannot be empty","message",JOptionPane.INFORMATION\_MESSAGE);

}

else if((p1.getText().length())==0)

{

JOptionPane.showMessageDialog(null,"password cannot be empty","message",JOptionPane.INFORMATION\_MESSAGE);

}

else

{

f.setVisible(false);

new LookingFor();

}

}

});

f.add(b1);

f.add(l1);

f.add(t1);

f.add(l2);

f.add(p1);

f.setBounds(500,100,500,500);

f.setLayout(null);

f.getContentPane().setBackground(Color.RED);

f.setVisible(true);

}

public static void main(String[] args)

{

new O\_bank();

}

}

**LookingFor code:**

package banking;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

public class LookingFor

{

public JFrame f1 = new JFrame("Looking for");

public static int balance=0,i=0;

public LookingFor(){

JButton b1=new JButton("DEPOSIT");

JButton b2=new JButton("WITHDRAW");

JButton b3=new JButton("BALANCE");

JButton b4=new JButton("MINI\_STATEMENT");

JButton b=new JButton("END");

b1.setBounds(20,80,150,40);

b2.setBounds(20,300,150,40);

b3.setBounds(300,80,150,40);

b4.setBounds(300,300,150,40);

b.setBounds(175,400,100,40);

f1.add(b1);

f1.add(b2);

f1.add(b3);

f1.add(b4);

f1.add(b);

b1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

f1.setVisible(false);

new Deposit();

}

});

b2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

f1.setVisible(false);

new Withdraw();

}

});

b3.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

new Balance();

}

});

b4.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

new Mini\_statement();

}

});

b.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

f1.dispose();

}

});

f1.setLayout(null);

f1.getContentPane().setBackground(Color.RED);

f1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f1.setBounds(500,100,500,500);

f1.setVisible(true);

}

}

**Deposit Code:**

package banking;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

public class Deposit extends LookingFor

{

int amount;

public JFrame f2 = new JFrame("Deposit");

public Deposit()

{

JLabel l = new JLabel("Enter amount");

l.setSize(60,30);

JTextField tf = new JTextField();

tf.setSize(100,30);

JButton b = new JButton("deposit");

JButton b1 =new JButton("back");

b.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

try {

amount=Integer.parseInt(tf.getText());

JOptionPane.showMessageDialog(null,amount+" rupees has deposited");

balance+=amount;

new Mini\_statement(amount,1);

}

catch(NumberFormatException e)

{

if(tf.getText().equals(""))

{

JOptionPane.showMessageDialog(null,"enter amount to deposit");

}

else

{

JOptionPane.showMessageDialog(null,"enter amount in numbers only");

}

}

}

});

b1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

f1.setVisible(true);

f2.setVisible(false);

}

});

b1.setBounds(20,400,100,40);

l.setBounds(40,40,150,40);

tf.setBounds(200,40,200,40);

b.setBounds(175,150,100,40);

f2.add(l);f2.add(tf);f2.add(b);

f2.add(b1);

f2.setLayout(null);

f2.getContentPane().setBackground(Color.RED);

f2.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f2.setBounds(500,100,500,500);

f2.setVisible(true);

}

}

**Withdraw Code:**

package banking;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

public class Withdraw extends LookingFor

{

int amount;

public JFrame f3 = new JFrame("WITHDRAW");

Withdraw()

{

JLabel l = new JLabel("amount");

l.setSize(30,30);

JTextField tf = new JTextField();

tf.setSize(100,30);

JButton b =new JButton("withdraw");

JButton b1 =new JButton("back");

b.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

try {

amount=Integer.parseInt(tf.getText());

if(amount<=balance)

{

JOptionPane.showMessageDialog(null,amount+" rupees had withdrawed");

balance-=amount;

new Mini\_statement(amount,2);

}

else

{

JOptionPane.showMessageDialog(null,"insufficient balance");

}

}

catch(NumberFormatException e)

{

if(tf.getText().equals(""))

{

JOptionPane.showMessageDialog(null,"enter amount to withdraw");

}

else

{

JOptionPane.showMessageDialog(null,"enter amount in numbers only");

}

}

}

});

b1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

f1.setVisible(true);

f3.dispose();

}

});

b1.setBounds(20,400,100,40);

l.setBounds(40,40,150,40);

tf.setBounds(200,40,200,40);

b.setBounds(175,150,100,40);

f3.add(l);

f3.add(tf);

f3.add(b);

f3.add(b1);

f3.setLayout(null);

f3.getContentPane().setBackground(Color.RED);

f3.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f3.setBounds(500,100,500,500);

f3.setVisible(true);

}

}

**Balance code:**

package banking;

import javax.swing.JOptionPane;

public class Balance extends LookingFor

{

Balance(){

JOptionPane.showMessageDialog(null,"current balance= "+balance+"rupees","message",JOptionPane.INFORMATION\_MESSAGE);

f1.dispose();

}

}

**MiniStatement Code**:

package banking;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class Mini\_statement

{

public static int i=0;

public static String str[]=new String[100];

public Mini\_statement()

{

JFrame f=new JFrame("MINI STATEMENT");

JLabel la=new JLabel("previouos transactions");

la.setBounds(20,20,150,40);

f.add(la);

for(int j=i-1,x=20,y=60,l=80,w=40;j>=0;j--)

{

JLabel l1=new JLabel(str[j]);

l1.setBounds(x,y,l,w);

f.add(l1);

y+=40;

if(y>500)

{

x+=100;

y=20;

}

}

JButton b=new JButton("Back");

b.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

f.dispose();

}

});

b.setBounds(400,400,80,40);

f.add(b);

f.setBounds(500,100,500,500);

f.setLayout(null);

f.getContentPane().setBackground(Color.RED);

f.setVisible(true);

}

public Mini\_statement(int s,int n)

{

if(n==1)

{

str[i++]="+"+s;

}

if(n==2)

{

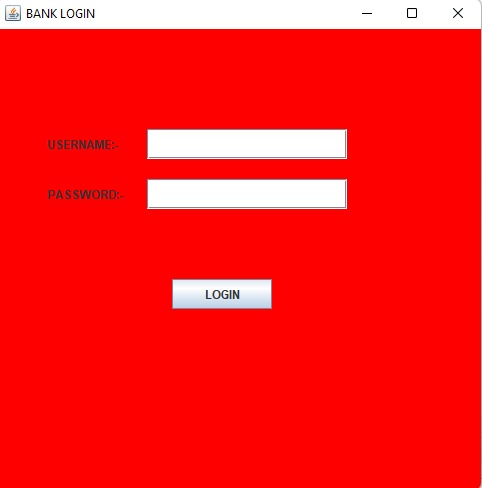
str[i++]="-"+s;

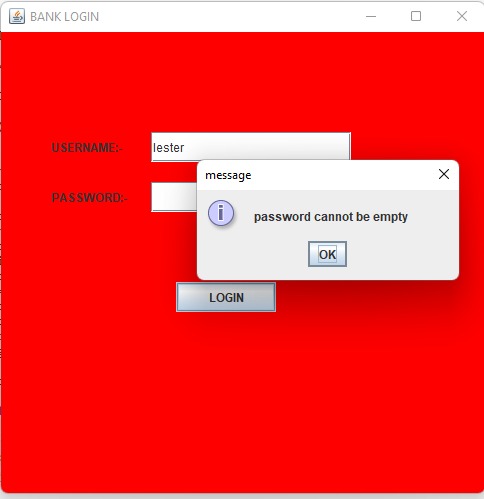
}

}

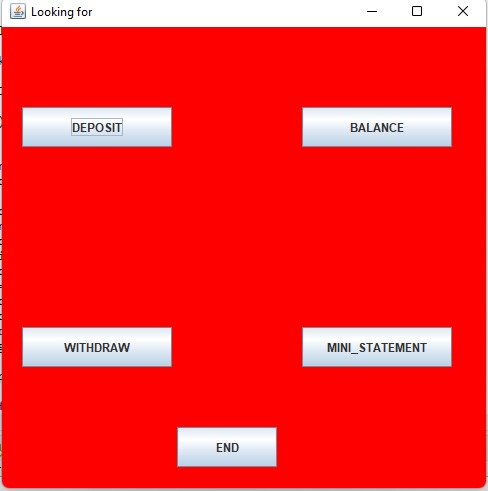
}

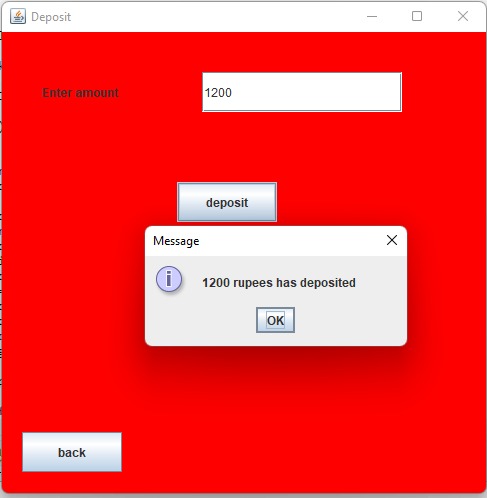
**Output:**

****

****

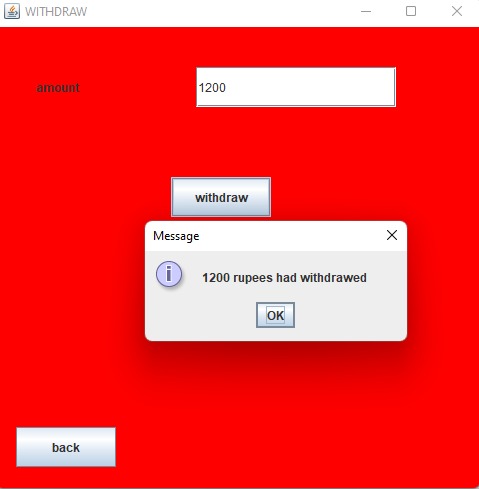
**This is the login page and it does not accept if any of the text field is empty . after entering proper details when we press login button we will be redirected to Looking for page.**

**These are the operations we can perform .**

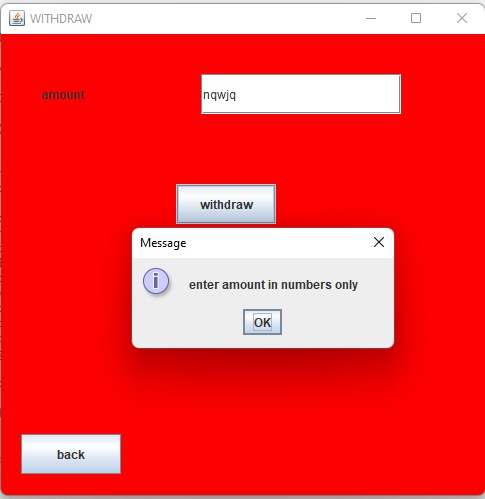
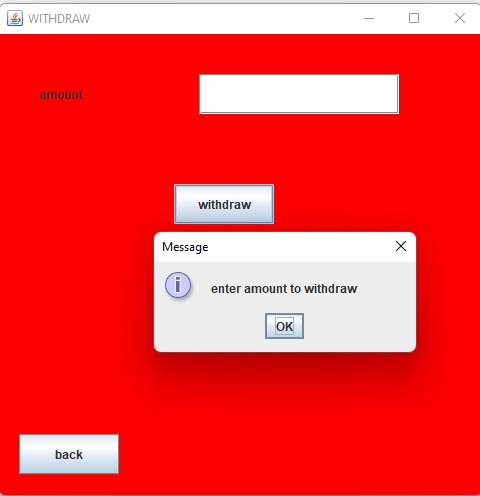
****

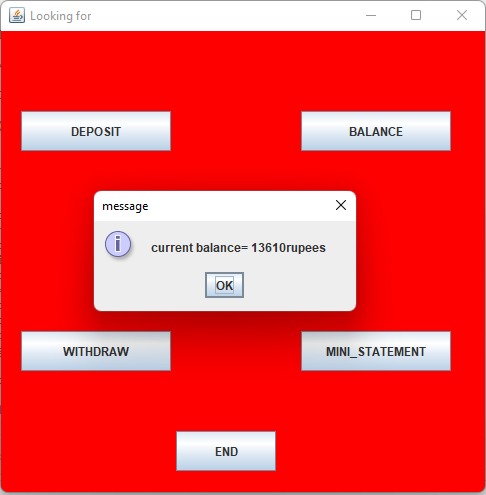
**If we click deposit we can find a text field to enter amount and by clicking deposit button we can deposit that money to our account.**

**And by clicking back button we can go back to lookingfor page.**

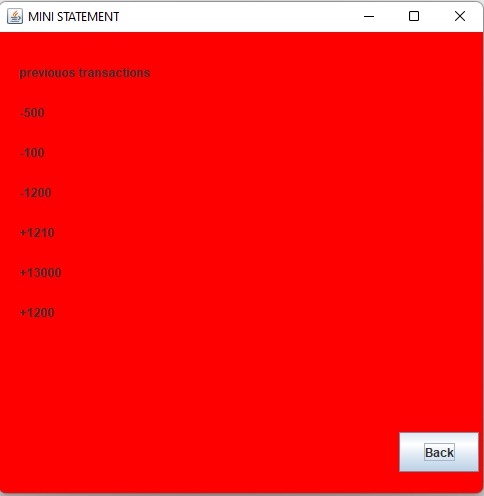
****

**By clicking withdraw button we can find a text field to enter amount to be withdrawn . If we click withdraw button without entering amount or if we enter any characters in the text field ,it will raise an error message.**

****

****

**By clicking balance button , we will get a dialog box showing our account balance.**

****

**By clicking mini statement button we will get the transaction history of the user all deposits and withdraws .**