**BANK APPLICATION**

Mini Project for the subject

Java Programming Lab

**S. E. Information Technology**

By

**Rhea Sera Rodrigues 06**

**Mukunda Gujjar 23**

**Reuben Coutinho 19**

**Joylyn D’souza 11**

Supervisor:

**Ms. Monika Pal**

Assistant Professor

****

Department of Information Technology

St. Francis Institute of Technology

(Engineering College)

University of Mumbai

2019-2020

**BANK APPLICATION**

**AIM: To develop a Banking Application**

**SOFTWARE REQUIREMENTS: JDK 1.8.0,JRE 1.8.0, NetBeans 8.2, wamp server, word document.**

**THEORY:**

1. **INTRODUCTION**

Bank management system can be considered as the most important thing in economic world.in the present scenario the banking sector is the common need in everyday life. In day to day life we face the problems and then we realize something is not done in this sector like we want to change the location (branch) of our account then we need to fill the application and then some day waiting to complete bank process. In this process amount of time is more as well as here occurs manual work which is increases man power.

So to make things simpler we have designed an application which will  facilitate the addition of new Customer A/c, deletion of A/c& modification of existing customer A/c,View the customer account details and also transfer money from one account to another account just by the click of a button.

1. **PROGRAM CODE**

**FRAME 1:-SPLASH SCREEN**

**Code:-**

public SplashScreen() throws InterruptedException

{

initComponents();

setVisible(true);

for (int i=1;i<4;i++)

{

Thread.sleep(1000);

}

dispose();

MAINAPP M = new MAINAPP();

}

**FRAME 2:- MAIN APP FRAME**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Registration\_page1 rp= new Registration\_page1();

rp.setVisible(true);

this.dispose();

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Customer\_login C=new Customer\_login();

C.setVisible(true);

this.dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Adminlogin A1=new Adminlogin();

A1.setVisible(true);

this.dispose();

}

**FRAME 3:ADMIN LOGIN FRAME**

private void logi\_for\_adminActionPerformed(java.awt.event.ActionEvent evt) {

MAINAPP M=new MAINAPP();

M.setVisible(true);

this.dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

String id=getadmin\_username.getText();

String password=getadmin\_password.getText();

if(evt.getSource()==jButton1)

{

if(id.compareTo("ADMIN")==0&&password.compareTo("ADMIN")==0)

{

Admin\_Page C=new Admin\_Page();

C.setVisible(true);

dispose();

}

else

JOptionPane.showMessageDialog(null,"INCORRECT CREDENTIALS");

}

}

**FRAME 4:- ADMIN PAGE**

private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {

adding\_balance a =new adding\_balance();

a.setVisible(true);

this.dispose();

}

private void withdraw\_money1MouseClicked(java.awt.event.MouseEvent evt) {

ToWithDrawMoney w=new ToWithDrawMoney();

w.setVisible(true);

this.dispose();

}

private void CUSTOMER\_LISTMouseClicked(java.awt.event.MouseEvent evt) {

Customer\_list c=new Customer\_list();

c.setVisible(true);

this.dispose();

}

private void deposit\_money1MouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

ToDepositMoney viewpage=new ToDepositMoney();

viewpage.setVisible(true);

this.dispose();

}

private void jLabel3MouseClicked(java.awt.event.MouseEvent evt) {

Delete\_account Da=new Delete\_account();

Da.setVisible(true);

this.dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Adminlogin al=new Adminlogin();

al.setVisible(true);

this.dispose();

}

**FRAME 5:- ADDING A BASE BALANCE**

private void search\_accountActionPerformed(java.awt.event.ActionEvent evt) {

try{

Database D=new Database();

D.dbConnect();

String sql="Select \* from register2 where acc\_number=? ";

PreparedStatement pstmt=D.dbcon.prepareStatement(sql);

pstmt.setString(1, account\_number.getText());

RS=pstmt.executeQuery();

if(RS.next())

{

JOptionPane.showMessageDialog(null,"Correct account number has been Entered \n Enter The balance");

pstmt.close();

}

else{

JOptionPane.showMessageDialog(null,"Enter the Correct account number");

}

}catch(Exception e)

{

JOptionPane.showMessageDialog(null,e);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

try{

Database D=new Database();

D.dbConnect();

String sql="update register2 set balance = ? where acc\_number = ? ";

PreparedStatement pstmt=D.dbcon.prepareStatement(sql);

pstmt.setFloat(1,Float.parseFloat(balance.getText()));

pstmt.setString(2, account\_number.getText());

pstmt.executeUpdate();

JOptionPane.showMessageDialog(null,"Money ENTERED");

pstmt.close();

}catch(Exception e)

{

JOptionPane.showMessageDialog(null,e);

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Admin\_Page A=new Admin\_Page();

A.setVisible(true);

this.dispose();

}

**FRAME 6:- TO WITHDRAW MONEY**

private void search\_buttonActionPerformed(java.awt.event.ActionEvent evt) {

try {

Database D=new Database();

D.dbConnect();

String sql="Select \* from register2 where acc\_number=? and acc\_pin=? ";

PreparedStatement pstmt=D.dbcon.prepareStatement(sql);

pstmt.setString(1,account\_number.getText());

pstmt.setString(2,account\_pin.getText());

RS=pstmt.executeQuery();

if(RS.next())

{

String add2=RS.getString("Balance");

get\_account\_balance.setText(add2);

RS.close();

pstmt.close();

}

else{

JOptionPane.showMessageDialog(null,"Enter the Correct account number\n or password");

}

}catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

try{

String balance=get\_account\_balance.getText();

String withdraw=amount\_to\_withdraw.getText();

float bal=Float.valueOf(balance);

float with=Float.valueOf(withdraw);

if(with<bal&&with>=0)

{

float sum=Float.parseFloat(balance)-Float.parseFloat(withdraw);

String total=String.valueOf(sum);

calculated\_amount.setText(total);

Database D=new Database();

D.dbConnect();

String total\_for\_query=calculated\_amount.getText();

String acc\_no=account\_number.getText();

String withdraw\_query="UPDATE register2 set Balance='"+total\_for\_query+"'where acc\_number='"+acc\_no+"'";

PreparedStatement pstmt=D.dbcon.prepareStatement(withdraw\_query);

pstmt.executeUpdate();

JOptionPane.showMessageDialog(null,"MONEY WITHDRAWN");

}else

{

JOptionPane.showMessageDialog(null,"Amount is Exceeding\n Or It is a Negative Number");

}

}catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Admin\_Page viewpage=new Admin\_Page();

viewpage.setVisible(true);

this.dispose();

}

**FRAME 7:- TO DEPOSIT MONEY**

private void searchbuttonActionPerformed(java.awt.event.ActionEvent evt) {

try {

Database D=new Database();

D.dbConnect();

String sql="select \* from register1,register2 where account\_number=? and acc\_pin=?";

PreparedStatement pstmt=D.dbcon.prepareStatement(sql);

pstmt.setString(1, account\_number.getText());

pstmt.setString(2,account\_pin.getText());

RS=pstmt.executeQuery();

if(RS.next())

{

String add1=RS.getString("acc\_number");

account\_no\_display.setText(add1);

String add2=RS.getString("Balance");

get\_account\_balance2.setText(add2);

RS.close();

pstmt.close();

}

else{

JOptionPane.showMessageDialog(null,"Enter the Correct account number\n or password");

}

}catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

try{

String balance=get\_account\_balance2.getText();

String deposit=amount\_to\_be\_deposited.getText();

float bal=Float.valueOf(balance);

float depo=Float.valueOf(deposit);

float r=(float) 0.0;

if(depo>r)

{

float sum=Float.parseFloat(balance)+Float.parseFloat(deposit);

String total=String.valueOf(sum);

calculated\_amount2.setText(total);

Database D=new Database();

D.dbConnect();

String total\_for\_query=calculated\_amount2.getText();

String acc\_no=account\_number.getText();

String withdraw\_query="UPDATE register2 set Balance='"+total\_for\_query+"'where acc\_number='"+acc\_no+"'";

PreparedStatement pstmt=D.dbcon.prepareStatement(withdraw\_query);

pstmt.executeUpdate();

JOptionPane.showMessageDialog(null,"The Money Has been added to your account");

}

else

{

JOptionPane.showMessageDialog(null,"Amount cannot be negative");

}

}catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

viewingcustomerpage viewpage=new viewingcustomerpage();

viewpage.setVisible(true);

this.dispose();

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

Admin\_Page viewpage=new Admin\_Page();

viewpage.setVisible(true);

this.dispose();

}

**FRAME 8:- TO VIEW CUSTOMER LIST**

public class Customer\_list extends javax.swing.JFrame {

ResultSet rs,rs1;

/\*\*

\* Creates new form Customer\_lIst

\*/

public Customer\_list() {

initComponents();

Table1();

Table2();

}

public void Table1()

{

try {

Database D=new Database();

D.dbConnect();

String take\_in\_infromation="select account\_number,first\_name,last\_name,date\_of\_birth,gender,city from register1";

PreparedStatement pstmt1=D.dbcon.prepareStatement(take\_in\_infromation);

rs1=pstmt1.executeQuery();

Table1.setModel(DbUtils.resultSetToTableModel(rs1));

rs1.close();

pstmt1.close();

rs1=pstmt1.executeQuery();

} catch (ClassNotFoundException ex) {

Logger.getLogger(Customer\_list.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(Customer\_list.class.getName()).log(Level.SEVERE, null, ex);

}

}

public void Table2()

{

try {

Database D=new Database();

D.dbConnect();

String take\_in\_infromation="select acc\_number,acc\_pin,nominee,personal\_id from register2";

PreparedStatement pstmt12=D.dbcon.prepareStatement(take\_in\_infromation);

rs1=pstmt12.executeQuery();

Table2.setModel(DbUtils.resultSetToTableModel(rs1));

rs1.close();

pstmt12.close();

rs1=pstmt12.executeQuery();

} catch (ClassNotFoundException ex) {

Logger.getLogger(Customer\_list.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(Customer\_list.class.getName()).log(Level.SEVERE, null, ex);

}

}

**FRAME 9:- TO DELETE AN ACCOUNT**

private void delete\_accountActionPerformed(java.awt.event.ActionEvent evt) {

try {

Database D=new Database();

D.dbConnect();

String get\_acc=account\_number.getText();

String query\_for\_delete="DELETE from register2 where acc\_number ='"+get\_acc+"'";

PreparedStatement pstmt\_delete=D.dbcon.prepareStatement(query\_for\_delete);

pstmt\_delete.execute();

String query\_for\_delete1="DELETE from register1 where account\_number ='"+get\_acc+"'";

PreparedStatement pstmt\_delete1=D.dbcon.prepareStatement(query\_for\_delete1);

pstmt\_delete1.execute();

JOptionPane.showMessageDialog(null,"ACCOUNT HAS BEEN SUCCESSFULLY DELETED");

} catch (ClassNotFoundException ex) {

Logger.getLogger(Delete\_account.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(Delete\_account.class.getName()).log(Level.SEVERE, null, ex);

}

**FRAME 10:- REGISTER PART 1**

private void REGISTERBUTTONActionPerformed(java.awt.event.ActionEvent evt) {

try{

Database D=new Database();

D.dbConnect();

String sql="insert into register1 values(?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement pstmt=D.dbcon.prepareStatement(sql);

pstmt.setInt(1,Integer.parseInt(account\_number.getText()));

pstmt.setString(2,register\_firstname.getText());

pstmt.setString(3,register\_lastname.getText());

pstmt.setInt(4,Integer.parseInt(register\_mobno.getText()));

pstmt.setString(5,getDateOfBirth.getText());

if(male.isSelected())

{

GENDER="MALE";

}

if(female.isSelected())

{

GENDER="FEMALE";

}

pstmt.setString(6,GENDER);

pstmt.setString(7,register\_address.getText());

pstmt.setString(8,City.getText());

pstmt.setString(9,Register\_state.getText());

pstmt.setString(10,Username.getText());

pstmt.setString(11,register\_password.getText());

pstmt.executeUpdate();

JOptionPane.showMessageDialog(null,"You will Now be Directed To the account section");

Registration\_Page\_Part2 P=new Registration\_Page\_Part2();

P.setVisible(true);

this.dispose();

}catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Customer\_login cu=new Customer\_login();

cu.setVisible(true);

this.dispose();

}

**FRAME 11:- REGISTER PART 2**

private void REGISTERBUTTONActionPerformed(java.awt.event.ActionEvent evt) {

try{

Database D=new Database();

D.dbConnect();

String sql="insert into register2 values(?,?,?,?,?,?)";

PreparedStatement pstmt=D.dbcon.prepareStatement(sql);

pstmt.setInt(1,Integer.parseInt(account\_no.getText()));

pstmt.setString(2,account\_nominee.getText());

pstmt.setInt(3,Integer.parseInt(get\_account\_pin.getText()));

pstmt.setInt(4,Integer.parseInt(get\_addhar.getText()));

String Acctype;

Acctype=Account\_type.getSelectedItem().toString();

pstmt.setString(5,Acctype);

pstmt.setInt(6,Integer.parseInt(balance\_getter.getText()));

pstmt.executeUpdate();

JOptionPane.showMessageDialog(null,"Registration Successful \n You Can Now reutrn to Login page");

this.dispose();

Customer\_login cl=new Customer\_login();

cl.setVisible(true);

}catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Customer\_login cu=new Customer\_login();

cu.setVisible(true);

this.dispose();

}

**FRAME 12:- CUSTOMER LOGIN**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

Registration\_page1 p = new Registration\_page1();

p.setVisible(true);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

Database D=new Database();

D.dbConnect();

String user\_name\_enter=Username\_of\_login.getText();

String pass\_enterd=Password.getText();

String sql="select \* from register1 where Username='"+user\_name\_enter+"'and password='"+pass\_enterd+"'";

PreparedStatement PST=D.dbcon.prepareStatement(sql);

RS=PST.executeQuery();

if(RS.next())

{

JOptionPane.showMessageDialog(null,"Username and Password is correct");

viewingcustomerpage v=new viewingcustomerpage();

v.setVisible(true);

this.dispose();

}

else{

JOptionPane.showMessageDialog(null,"Username and Password is incorrect");

}

} catch (ClassNotFoundException ex) {

Logger.getLogger(Customer\_login.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(Customer\_login.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void logi\_for\_adminActionPerformed(java.awt.event.ActionEvent evt) {

MAINAPP M=new MAINAPP();

M.setVisible(true);

this.dispose();

}

**FRAME 13:- TO VIEW PERSONAL ACCOUNT**

private void View\_ProfileActionPerformed(java.awt.event.ActionEvent evt) {

customer\_details cusd=new customer\_details();

cusd.setVisible(true);

this.dispose();

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

customerbalanceViewer B=new customerbalanceViewer();

B.setVisible(true);

this.dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Customer\_login viewpage=new Customer\_login();

viewpage.setVisible(true);

this.dispose();

}

private void ToTransferActionPerformed(java.awt.event.ActionEvent evt) {

transferpage t=new transferpage();

t.setVisible(true);

this.dispose();

}

**FRAME 14:- TO VIEW PROFILE**

private void view\_detailsActionPerformed(java.awt.event.ActionEvent evt) {

try{

Database D=new Database();

D.dbConnect();

String sql="select \* from register1,register2 where account\_number=? and acc\_pin=?";

PreparedStatement PST=D.dbcon.prepareStatement(sql);

PST.setString(1,account\_no.getText());

PST.setString(2,account\_pin.getText());

RS=PST.executeQuery();

if(RS.next())

{

String add1=RS.getString("first\_name");

firstname.setText(add1);

String add2=RS.getString("last\_name");

Lastname.setText(add2);

String add3=RS.getString("mobile\_numer");

Phonenumber.setText(add3);

String add4=RS.getString("date\_of\_birth");

dateofbirth.setText(add4);

String add5=RS.getString("gender");

gender.setText(add5);

String add6=RS.getString("address");

adress.setText(add6);

String add7=RS.getString("city");

cityname.setText(add7);

String add8=RS.getString("state");

state.setText(add8);

RS.close();

PST.close();

firstname.setEnabled(false);

Lastname.setEnabled(false);

dateofbirth.setEnabled(false);

gender.setEnabled(false);

Phonenumber.setEditable(true);

adress.setEditable(true);

cityname.setEditable(true);

state.setEditable(true);

}

else{

JOptionPane.showMessageDialog(null,"Data was not able to be displayed");

}

}catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

private void UPDATEActionPerformed(java.awt.event.ActionEvent evt) {

try {

Database D=new Database();

D.dbConnect();

String account\_number=account\_no.getText();

String edited\_phone\_number=Phonenumber.getText();

String edited\_address=adress.getText();

String edited\_cityname=cityname.getText();

String edited\_statename=state.getText();

String sql\_query\_to\_update="Update register1 set mobile\_number='"+edited\_phone\_number+"',address='"+edited\_address+"',city='"+edited\_cityname+"',state='"+edited\_statename+"'where account\_number='"+account\_number+"'";

PreparedStatement pst=D.dbcon.prepareStatement(sql\_query\_to\_update);

pst.executeUpdate();

JOptionPane.showMessageDialog(null,"Succesfully Updated");

} catch (ClassNotFoundException ex) {

Logger.getLogger(customer\_details.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(customer\_details.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

viewingcustomerpage viewpage=new viewingcustomerpage();

viewpage.setVisible(true);

this.dispose();

}

**FRAME 15:- TO VIEW BALANCE**

private void select1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

Database D=new Database();

D.dbConnect();

String sql="select \* from register2 where acc\_number=? and acc\_pin=?";

PreparedStatement PST=D.dbcon.prepareStatement(sql);

PST.setString(1,Accountnumber.getText());

PST.setString(2,account\_pin.getText());

RS=PST.executeQuery();

if(RS.next())

{

String add1=RS.getString("nominee");

Accountnominee.setText(add1);

String add2=RS.getString("personal\_ID");

Addarnumber.setText(add2);

String add3=RS.getString("account\_type");

Accounttype.setText(add3);

String add4=RS.getString("Balance");

BALANCE.setText(add4);

Accountnominee.setEnabled(false);

Addarnumber.setEnabled(false);

Accounttype.setEnabled(false);

BALANCE.setEnabled(false);

}

else

{

JOptionPane.showMessageDialog(null,"DATA NOT ABLE TO DISPLAY :ERROR");

}

} catch (ClassNotFoundException ex) {

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

} catch (SQLException ex) {

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

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

viewingcustomerpage viewpage=new viewingcustomerpage();

viewpage.setVisible(true);

this.dispose();

}

**FRAME 16:- TO TRANSFER MONEY**

private void search\_buttonActionPerformed(java.awt.event.ActionEvent evt) {

try {

Database D=new Database();

D.dbConnect();

String sql="Select \* from register2 where acc\_number=? and acc\_pin=? ";

PreparedStatement pstmt=D.dbcon.prepareStatement(sql);

pstmt.setString(1, Sender\_Account\_no.getText());

pstmt.setString(2,account\_pin.getText());

RS=pstmt.executeQuery();

if(RS.next())

{

String add2=RS.getString("Balance");

balance.setText(add2);

RS.close();

pstmt.close();

}

else{

JOptionPane.showMessageDialog(null,"Enter the Correct account number\n or pin");

}

}catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

viewingcustomerpage v=new viewingcustomerpage();

v.setVisible(true);

this.dispose();

}

private void transfer\_buttonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

try {

String acount\_no\_of\_sender=Sender\_Account\_no.getText();

String balance\_1=balance.getText();

String wthdraw\_1=withdraw.getText();

String receiver=revceiver\_account\_number.getText();

float bal=Float.valueOf(balance\_1);

float with=Float.valueOf(wthdraw\_1);

float depo=Float.valueOf(receiver);

Database D=new Database();

D.dbConnect();

if(with<bal&&with>0)

{ float sum=Float.parseFloat(balance\_1)-Float.parseFloat(wthdraw\_1);

String total=String.valueOf(sum);

String withdrawing="Update register2 set Balance='"+total+"'where acc\_number='"+acount\_no\_of\_sender+"'";

PreparedStatement pstmt=D.dbcon.prepareStatement(withdrawing);

pstmt.executeUpdate();

String receiver\_sql="Select \*from register2 where acc\_number='"+receiver+"'";

PreparedStatement pstmt\_of\_receiver=D.dbcon.prepareStatement(receiver\_sql);

rs2=pstmt\_of\_receiver.executeQuery();

if(rs2.next())

{

String balance\_of\_reciever=rs2.getString("Balance");

float sum\_of\_receiver=Float.parseFloat(balance\_of\_reciever)+Float.parseFloat(wthdraw\_1);

String total\_of\_receiver=String.valueOf(sum\_of\_receiver);

String deposit="Update register2 set Balance='"+total\_of\_receiver+"'where acc\_number='"+receiver+"'";

PreparedStatement pstmt\_final=D.dbcon.prepareStatement(deposit);

pstmt\_final.executeUpdate();

JOptionPane.showMessageDialog(null,"The Money Has been Successfully Transferred");

}

else

{

JOptionPane.showMessageDialog(null,"Amount Not Transferred");

}

}

else

{

JOptionPane.showMessageDialog(null,"Amount is Exceeding") }

} catch (ClassNotFoundException ex) {

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

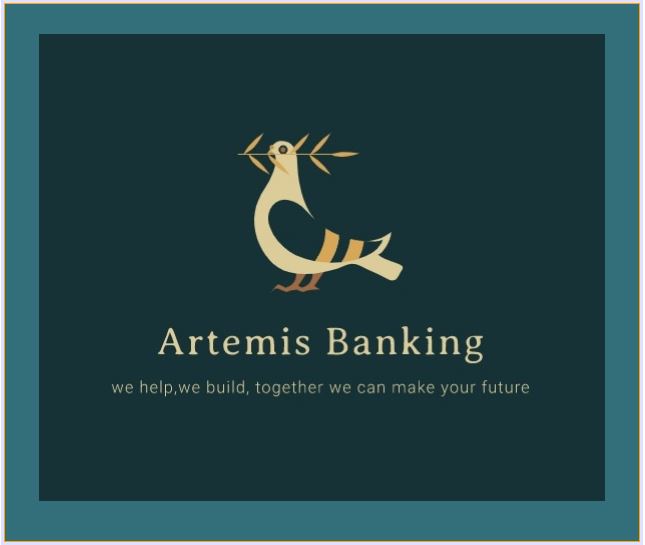
} catch (SQLException ex) {

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

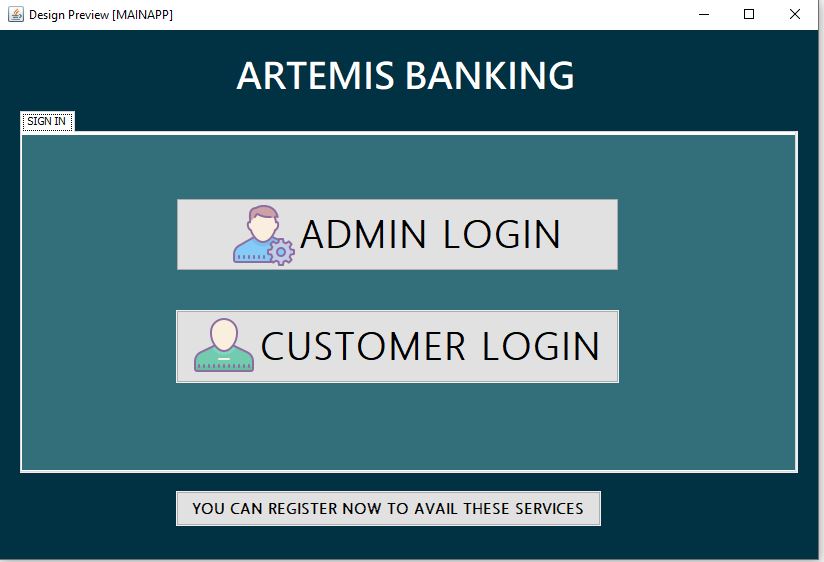
};}

**OUTPUT:-**

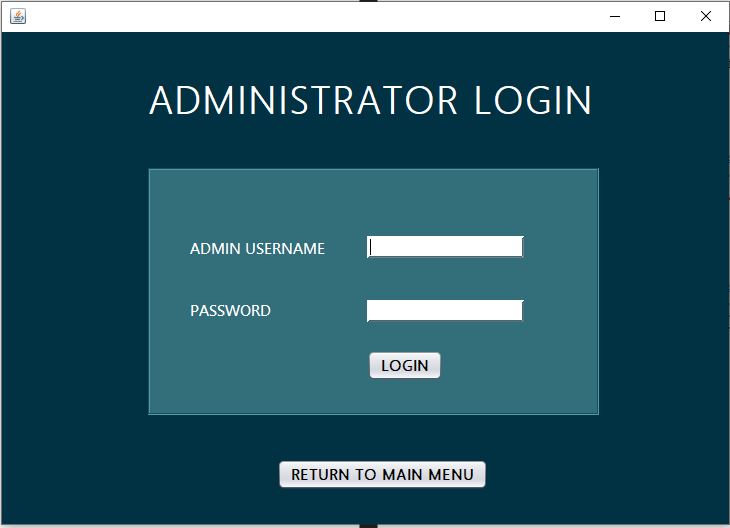
**FRAME 1:-SPLASH SCREEN**

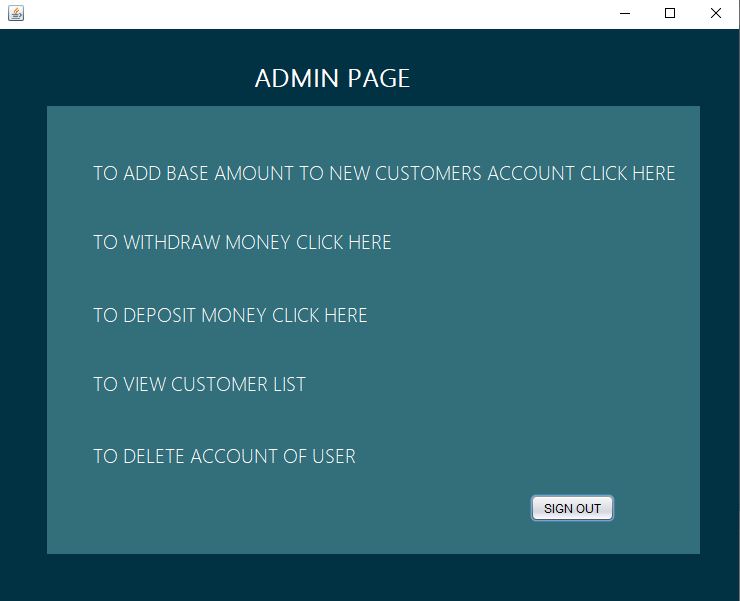
****

**FRAME 2:- MAIN APP FRAME**

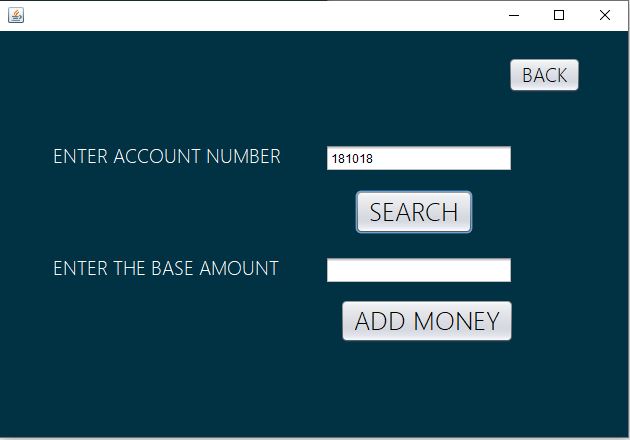
****

**FRAME 3:ADMIN LOGIN FRAME**

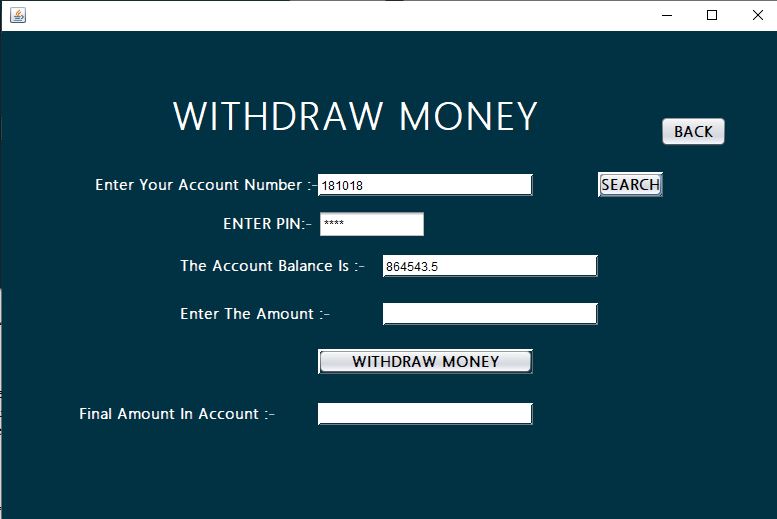
****

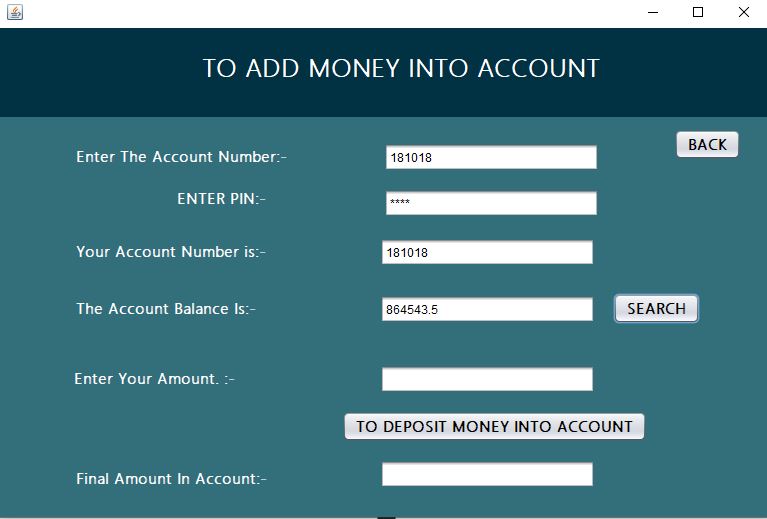
**FRAME 4:- ADMIN PAGE**

**FRAME 5:- ADDING A BASE BALANCE**

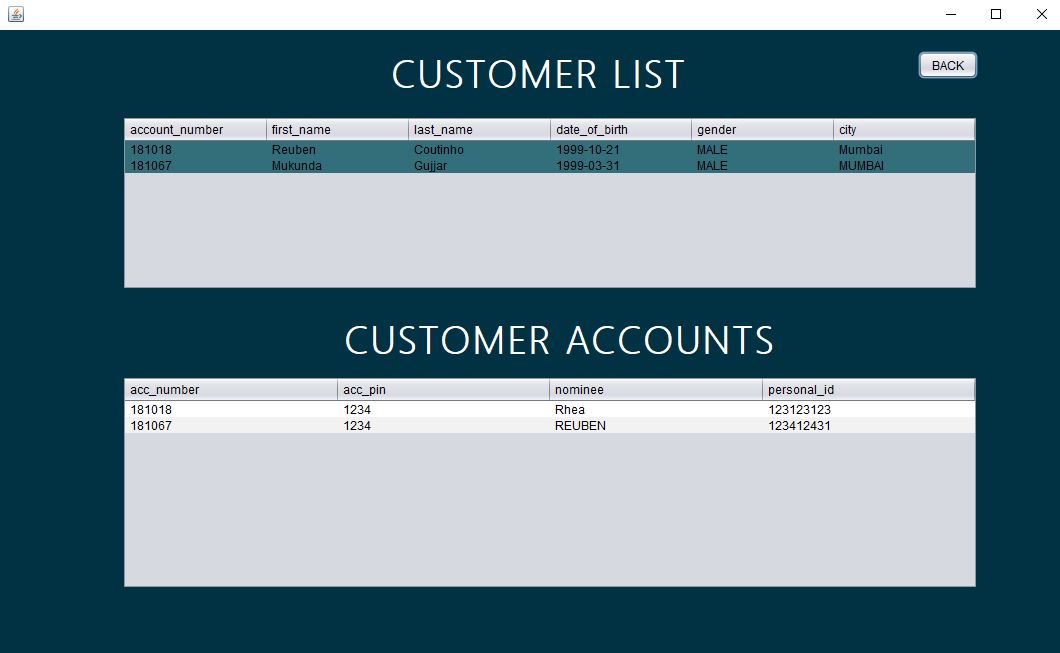


**FRAME 6:- TO WITHDRAW MONEY**

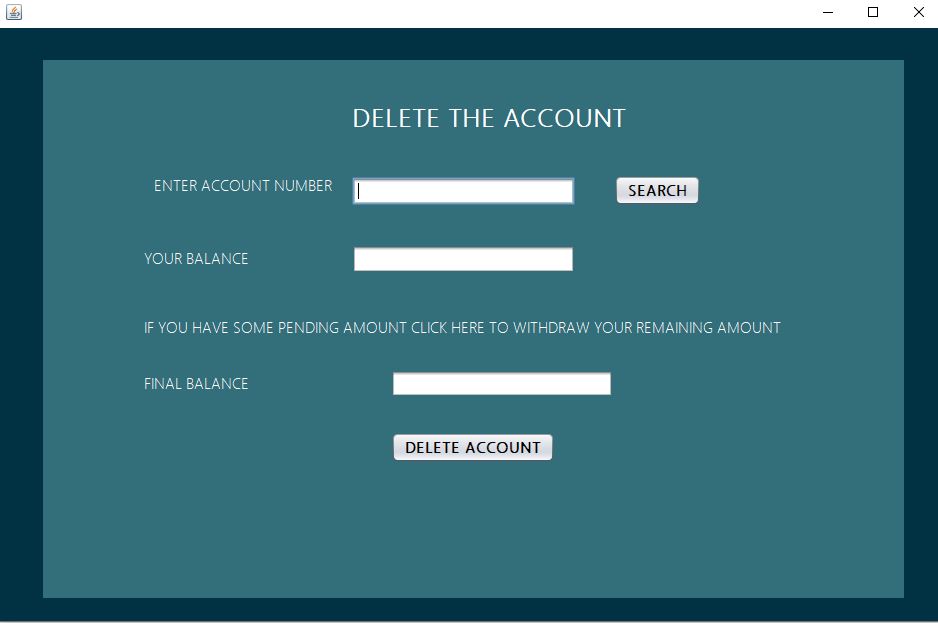
****

**FRAME 7:- TO DEPOSIT MONEY **

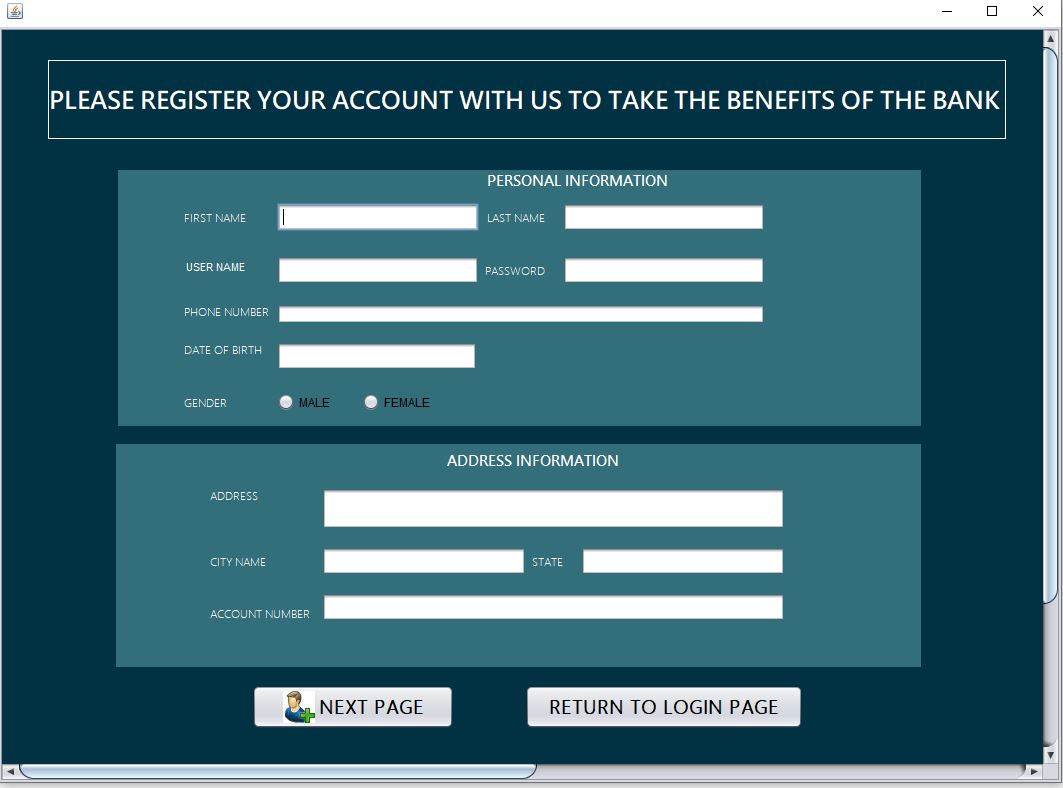
**FRAME 8:- TO VIEW CUSTOMER LIST**

****

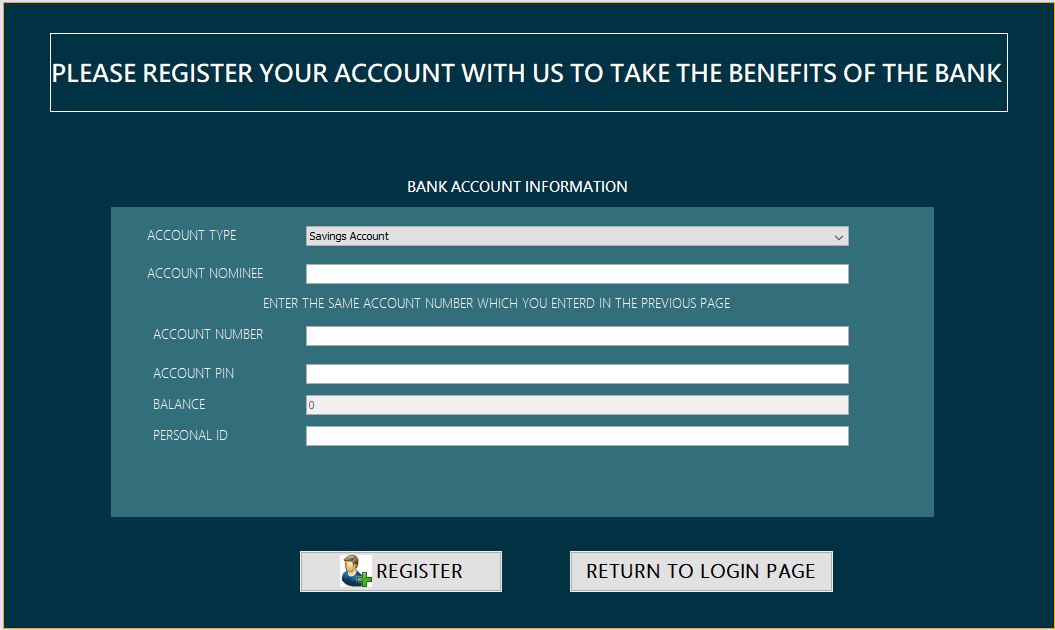
**FRAME 9:TO DELETE ACCOUNT**

****

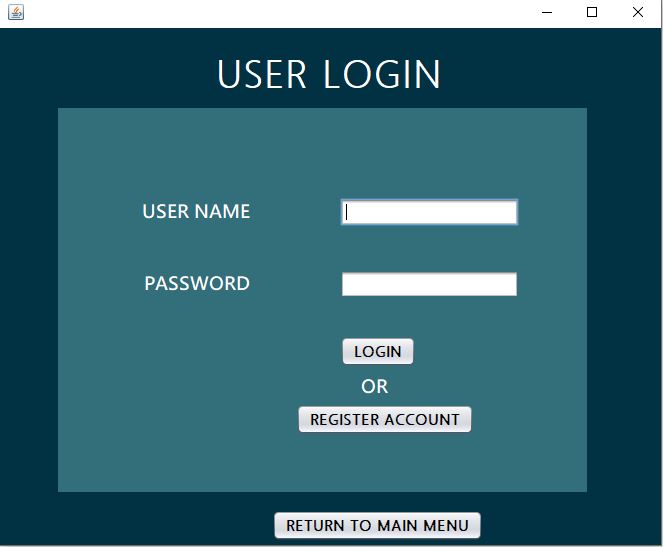
**FRAME 10:- REGISTER PART 1**

****

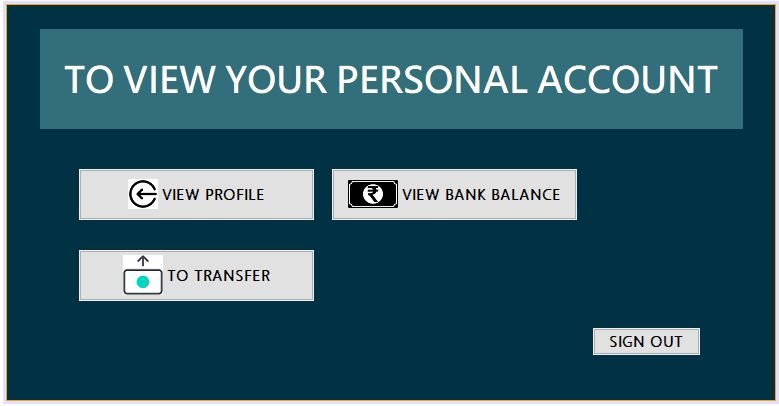
**FRAME 11:- REGISTER PART 2**

****

**FRAME 12:- CUSTOMER LOGIN**

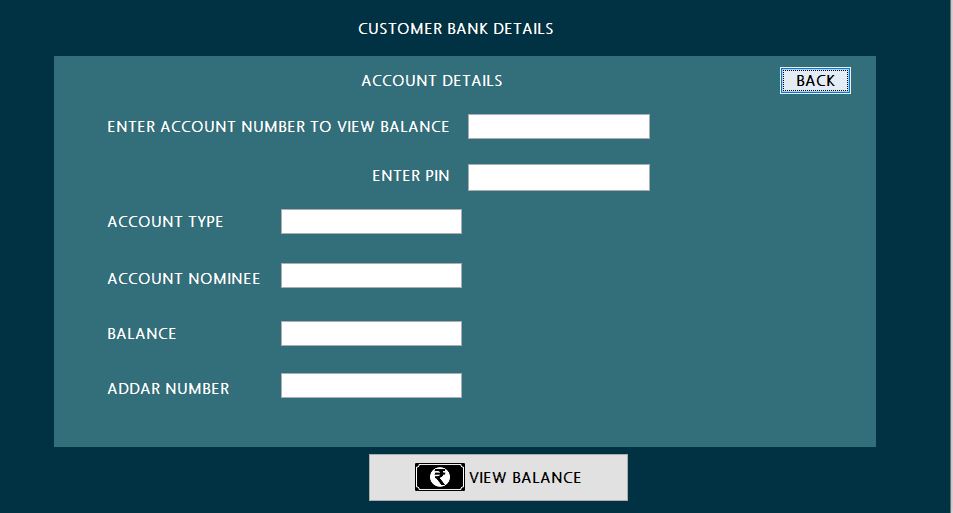
****

**FRAME 13:- TO VIEW PERSONAL ACCOUNT**

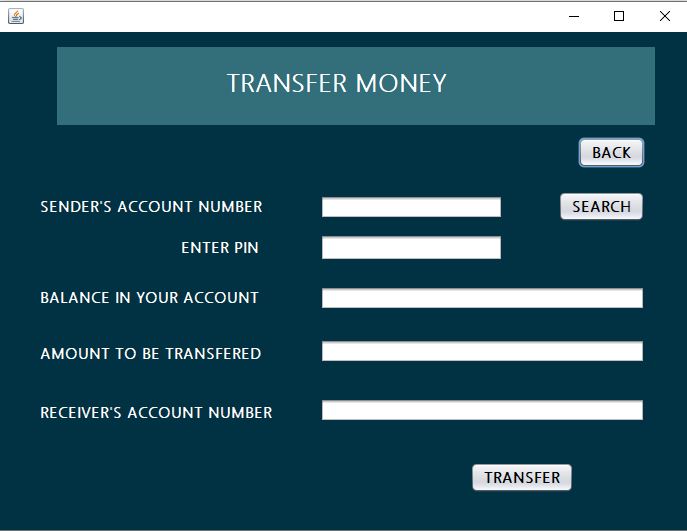
****

**FRAME 14:- TO VIEW PROFILE**

**FRAME 15:- TO VIEW BALANCE**

****

**FRAME 16:- TO TRANSFER MONEY**

****

1. **REFERENCES AND CONCLUSION**

**Conclusion:-**

From the above project we were learned how to develop a bank application and link it with the SQL server to create a database. We learned how the connection of front-end takes place with the back end. We implemented various functions such as the administrator was able to add a base amount, withdraw amount ,deposit amount, to view the customer list ,and delete the account at the end.At the User End the User is able to View his balance and his details and is also allowed to update the details. He is also able to transfer Money From one account to another account.

**Future scope of the project**:

This project was developed to fulfill user requirement; however there are lots of scope to improve the performance of the Banking System in the area of user interface, database performance, and query processing time. Etc. So there are many things for future enhancement of this project. The future enhancements that are possible in the project are as follows.

 ∙ Interest calculation system is not implemented yet. It can be enhanced later.

 ∙ Linking and integration of any legacy system for accounting.

∙ Integration with other bank and government agencies through Web Services

 ∙ Connection to third-party OLAP applications

∙ Electronic Data Interchange (EDI) system for ATM machine

 ∙ Web Interface for net banking.

 ∙ In the area of data security and system security.

**References:-**

1. <https://www.youtube.com/watch?v=wPfAmxMttEs>
2. <https://www.youtube.com/playlist?list=PL_Ke9hJMFeR_2F2JXcF8txUCxJGtbIZ6E>
3. [**https://www.youtube.com/watch?v=uuhEb0k3vVE&t=115s**](https://www.youtube.com/watch?v=uuhEb0k3vVE&t=115s)
4. <https://www.youtube.com/watch?v=QnKHOPXQd6E&t=229s>