***A REPORT ON***



GROUP MEMBERS

CHIKKULAPALLY VINEETH

CHANDAN SATYENDRA



* ACKNOWLEDGEMENT
* ABSTRACT

1. **INTRODUCTION**

ABOUT THE PROJECT

1. **SYSTEM REQUIREMENT**

HARDWARE REQUIREMENT

SOFTWARE REQUIREMENT

1. **PROBLEM DEFINITION AND PROPOSAL**

INTRODUCTION

* EXISTING SYSTEM
* PROPOSED SYSTEM

**4. SYSTEM DESIGN**

FLOW CHART

* MODULE DISCRIPTION AND SCREENSHOTS

1. **TESTING**

TESTING BASED ON ENTRIES

VALIDATION CHECK

* + 1. **CONCLUSION AND FUTURE ENHANCEMENT**
* [CONCLUSION](#page23)
* FUTURE ENHANCEMENT
* REFERENCE
  + 1. **SOURCE CODE**
  1. **CERTIFICATE**

**ACKNOWLEDGEMENT**

We take this opportunity to express our profound gratitude and deep regards to our faculty PROF **Mr. KABANI** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We are highly obliged to our project team members for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our assignment.

PROJECT DEVELOPERS:-

CH.VINEETH

C.SATYENDRA

**ABSTRACT**

The adoption of Electronic Banking by commercial enterprises has been in existence since the mid 90’s, much greater in number due to lower operating costs associated with it. Electronic banking has initially been in the form of automatic teller machines and telephone transactions. More recently a new delivery channel for banking services that benefits both customers and banks is in implementation i.e. using computers.

In this proposed interface, manager and accountant have access to the customer’s account, and manager have special preference to access all details of banking system.

Banking system services can include:

* Client
* Front Office Advisors
* Managers

**INTRODUCTION**

A Bank is a commercial or state institution that provides financial services, including issuing money in form of coins, bank notes or debit cards, receiving deposits of money, lending money and processing transections. A commercial bank accepts deposits from customers and in turn makes loan based on those deposits.

This project is basically providing a full service user friendly interface primarily specializing in Banking Management System;

We are using following details of clients:

* Client ID
* First Name
* Last Name
* Photo
* Date Of Birth
* Account Type
* Gender
* Telephone
* Address
* Occupation
* Nationality
* Martial Status
* Email ID

The bank will provide following functionalities to clients:

* Create his account
* View and modify their personal information
* View the balance and transaction history of their accounts
* Extract statements of account(s) in .csv format
* Make a Transfer(Internal or external)
* Feed and account
* Viewing transections of deposit.

The bank will provide the functionalities to the Front office advisers:

* See the list of clients
* View clients personal information
* View the balance and transaction history of all client accounts

The bank will provide following functionalities to managers:

* + View the sum of customer deposit in defined period
  + Generate reports on customer activities in defined period
* We will provide an email alert to the customer after the account creation providing them with their account numbers.
* We will also provide email alert for each deposit and withdrawal of money from his/her account.

**SYSTEM REQUIREMENTS**

HARDWARE REQUIREMENT

* Laptop Intel i3 3rd Gen clock speed 2.5 GHz and above.
* Hard disk : min. 10 GB
* Ram : 1 GB
* Monitor resolution : 800 X 600

SOFTWARE REQUIREMENT

* ANY Operating System (Windows 10 / 8, Linux, Mac)
* JDK 8 (Java Development Kit)
* JRE 8 (Java Runtime Environment)
* Language Used : JAVA 8
* Eclipse Neon IDE
* File system

**PROBLEM DEFINITION AND PROPOSAL**

INTRODUCTION

In proposed system, the management needs not to keep any type of registers, which they use to keep in old one. They have work only with one computer. All the details are stored in computer files. The dual entries are done very quickly as entry in one file only , affects the other file where it has to record. In the way there is no need to record the computer operator.

The working through registers system is not only very laborious and time consuming , but there are number of errors found. In comparison to old system the proposed system of computer processes many advantages.

Some of the merits of proposed system are as follows:-

* **Accuracy:-**

It is one of the features of computer that it provides accuracy result. It is often said, “Man does mistakes but machines never”. When a man works there seems numbers of errors and mistakes in performing calculation and other things, while it is not in case with computers.

* **Security:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| The | security is maintained vary comfortably in computers. | | Secretes | are |
| not | disclosed off, as there are password settings in | the | computers. | So |
| the firm can get the benefit of non-disclosure of | | its | secrets to | the |
| outsider competitors as well as other non-authorized | | persons. | |  |

* **Time saving:-**

The computerized system is time saving. On an immediate inquiry , the operator or any authorized person has to just click some switches and get the information in few seconds.

* **Perform Repetitive Task Very Well:-**

It is said that computer never tires. I do the repetitive task very well as it never feels laziness. Performance of repetitive task by a man is not well Performed, as main tires very quickly and feels lazy. It is not in the case with computer. So, proposed system is beneficial than Old one.

**EXISTING SYSTEM**

Presently there is no such system available. Some are there with limited access and following limitations:

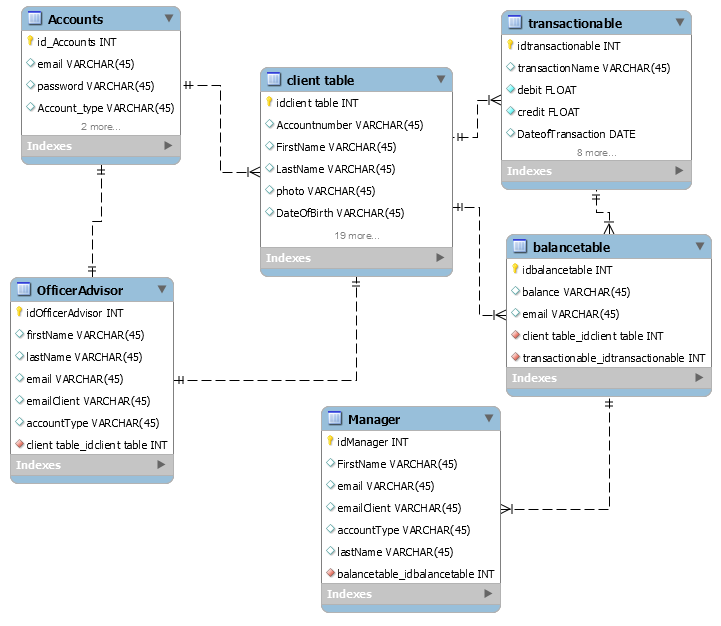
* Time consuming…
* More expensive…
* Searching problem…
* Maintains problem of all registers…
* Less accuracy…
* Problem for marking of not normal value…

**PROPOSED SYSTEM**

In our system we are looking forward to provide following advantages to users:

* Time saving…
* Less expensive…
* Powerful searching…
* Solve the problem of maintains register…
* More accuracy…
* Easy generate report…
* Not display multiple reports…

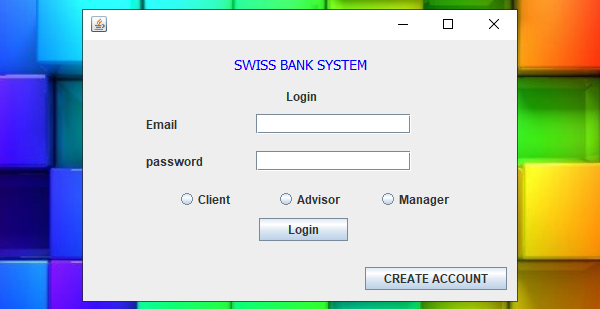
**UML DIAGRAM**



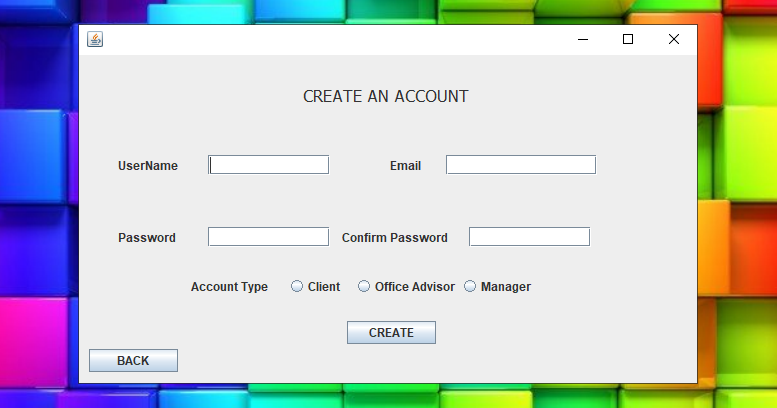
**MODULES DESCRIPTION AND SCREENSHOTS**

* WELCOME FRAME

This page will redirect to login page. This page is common for client, advisor & Manager .



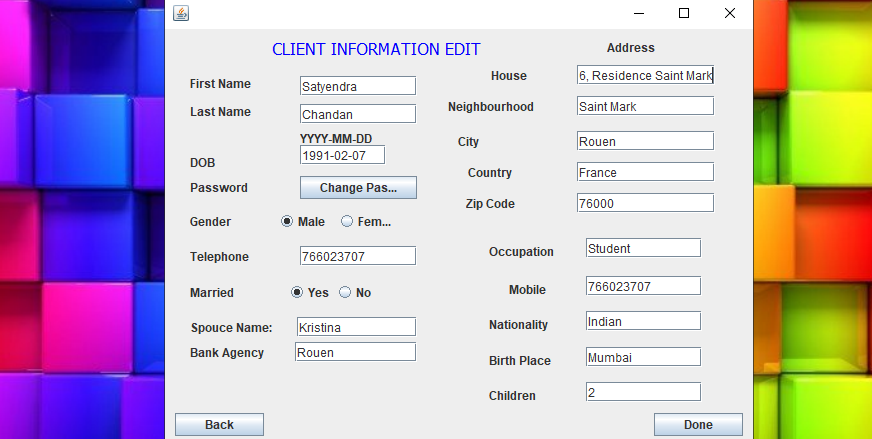
Creating An Account :



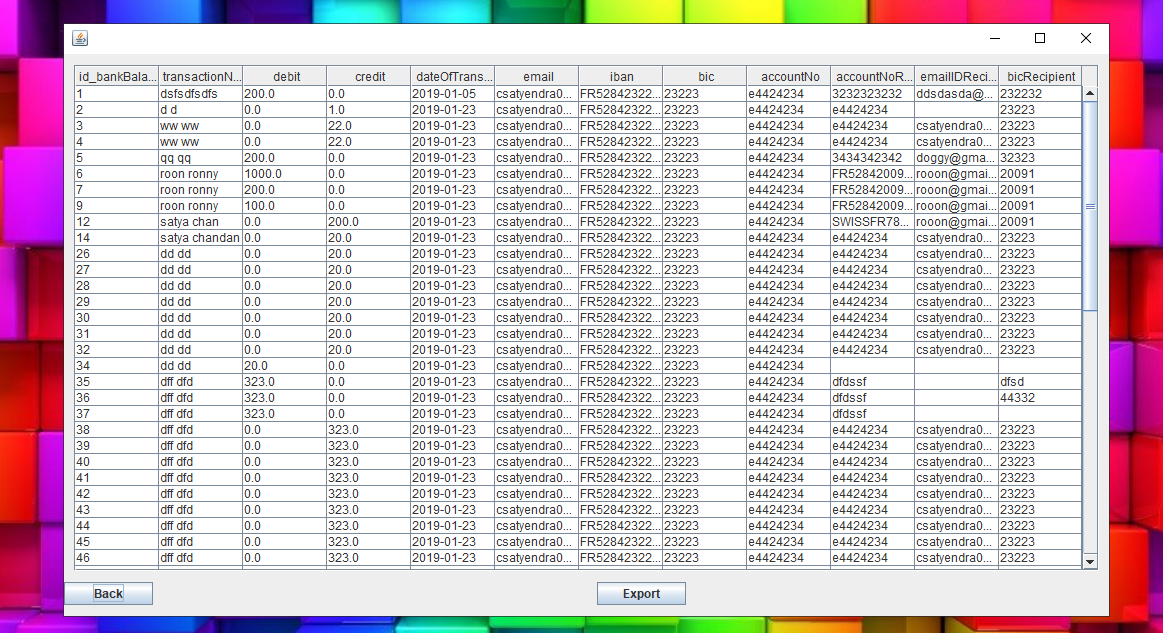
CLIENT INFORMATION:



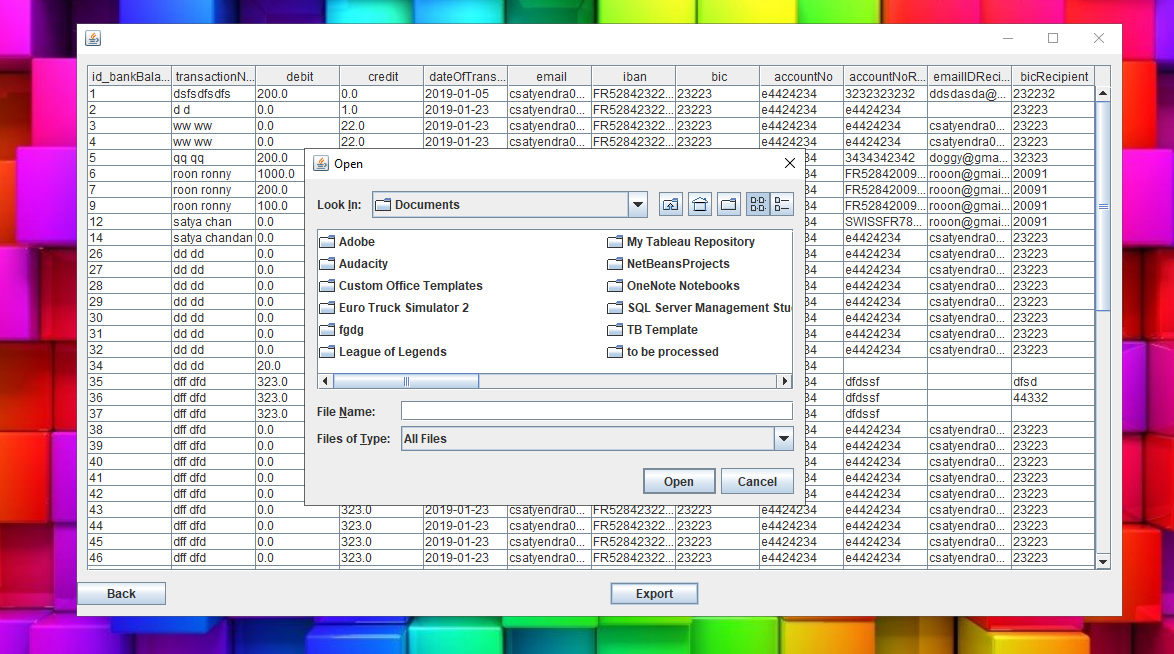
CLIENT PERSONAL DETAILS EDIT:

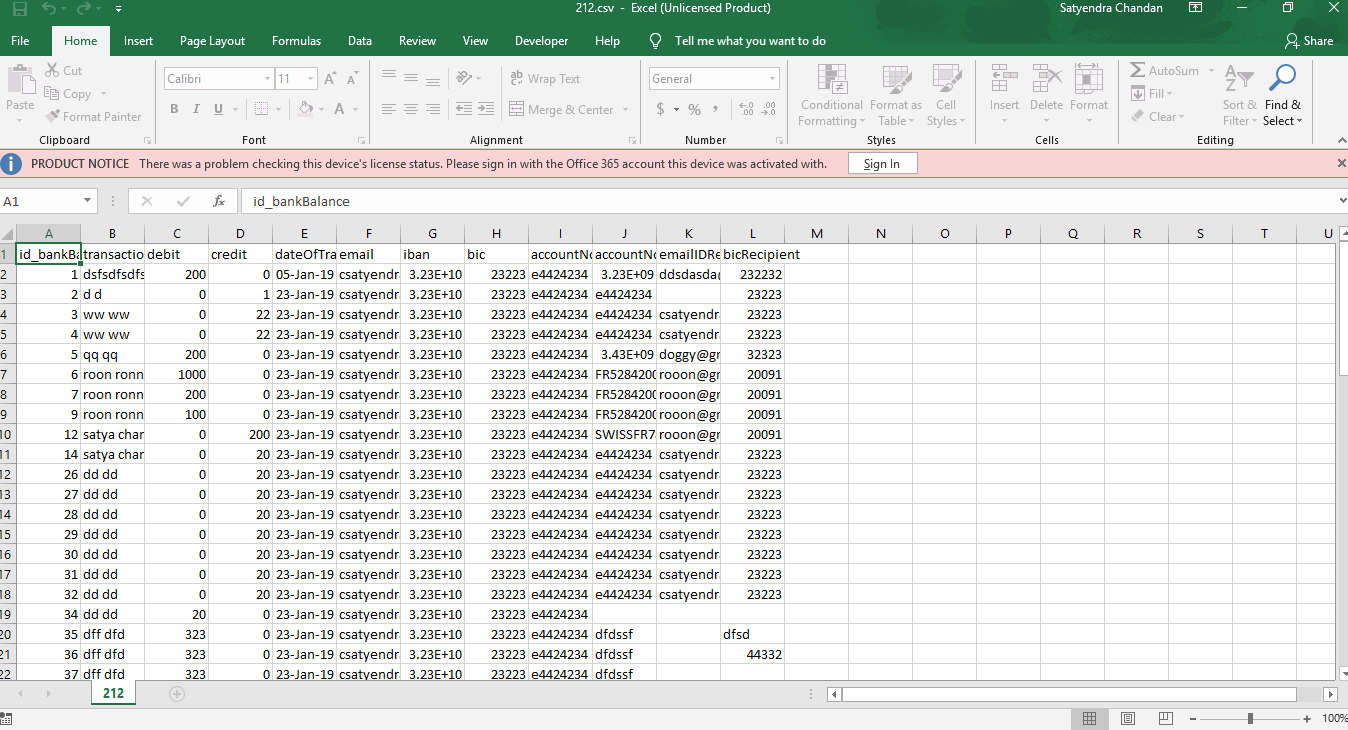


TRANSACTION DETAILS: Here we can see total transaction details including with benefices details.

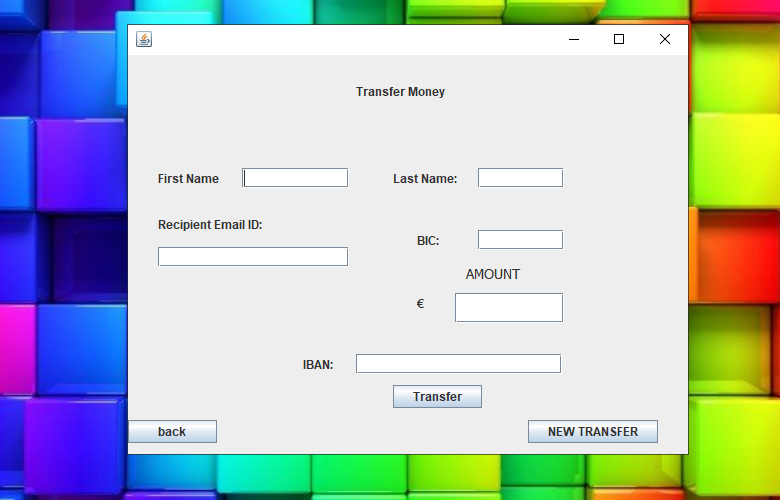


EXPORT TRANSACTION DETAILS:

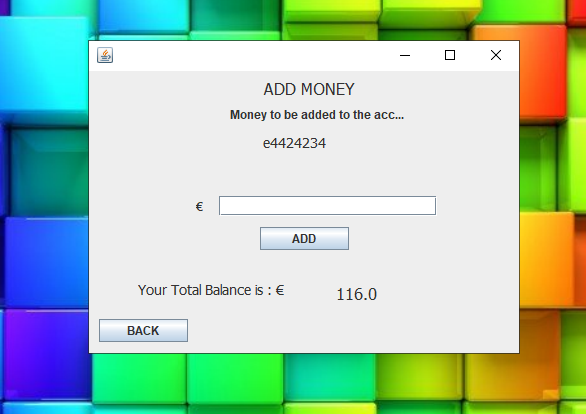




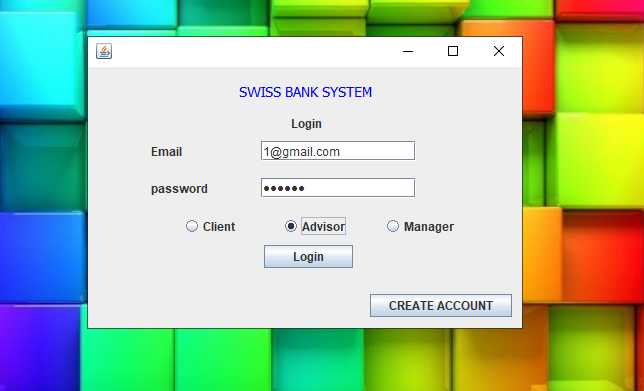
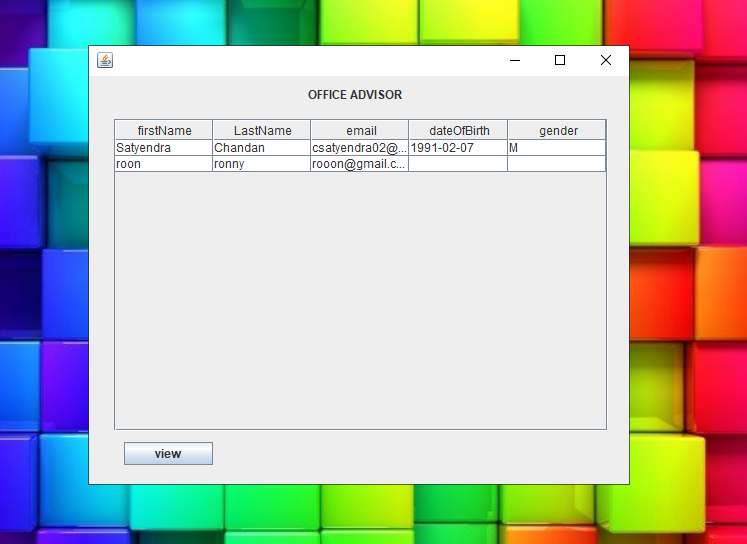
TRANSFER MONEY:



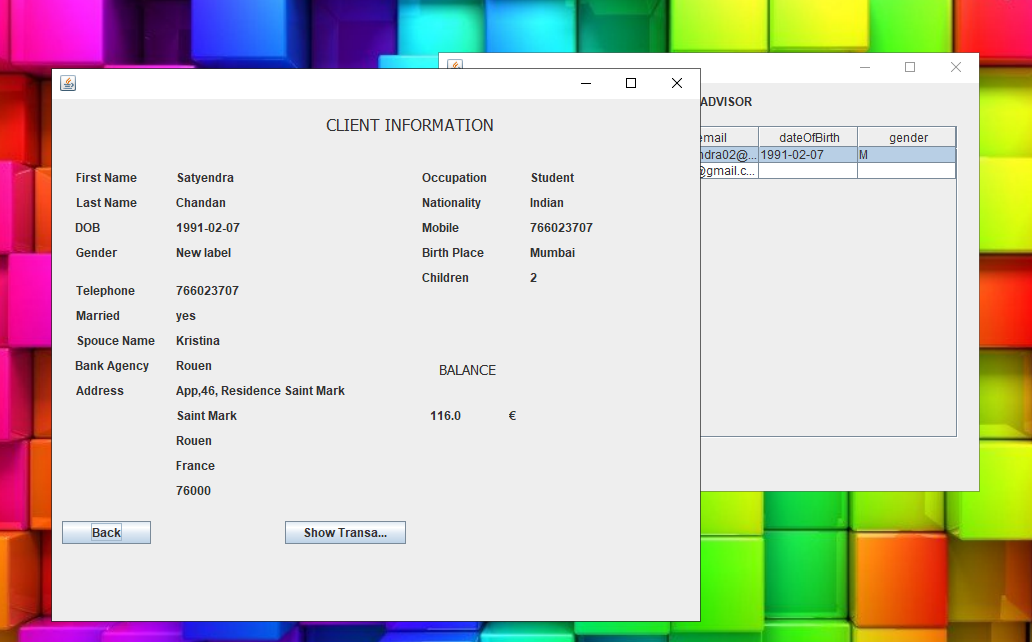
ADD MONEY:



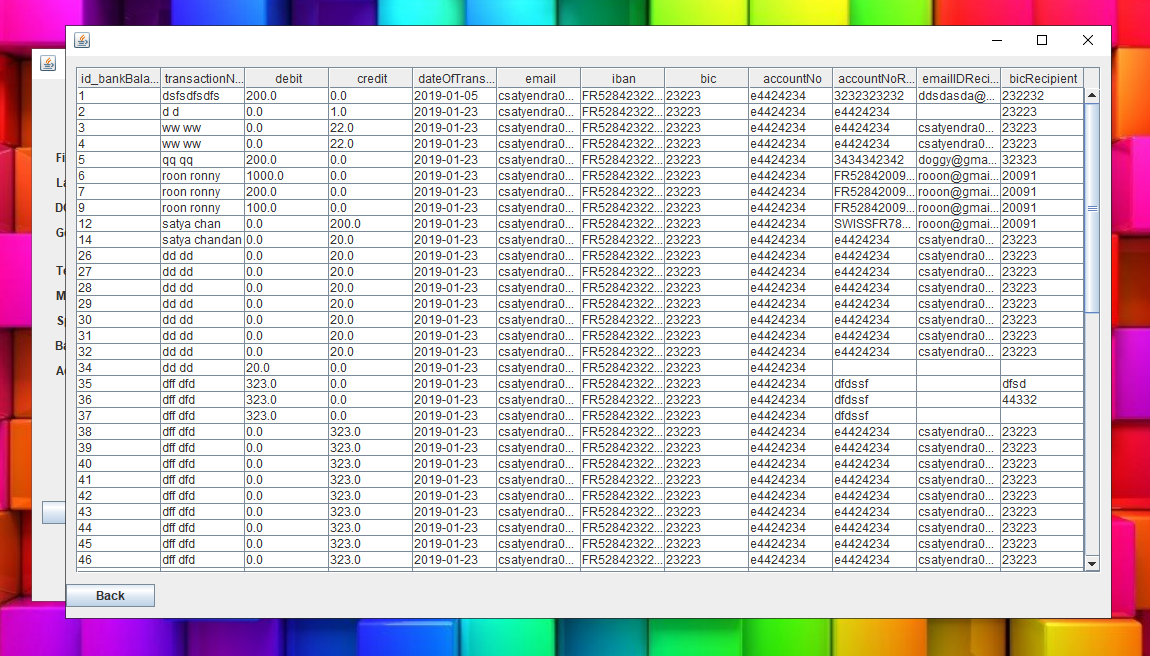
TO VIEW CLIENTS DETAILS BY OFFICE ADVISOR:

* Office advisor can see client information Office advisor should login by his mail ID & Password and choose advisor option as shown in above image.
* By logging he can see all client names.

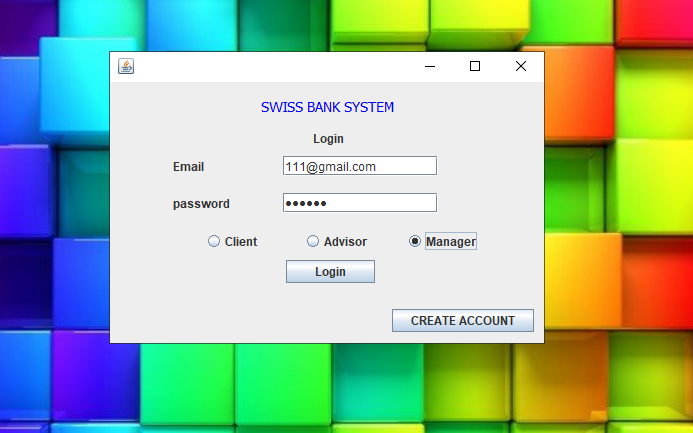
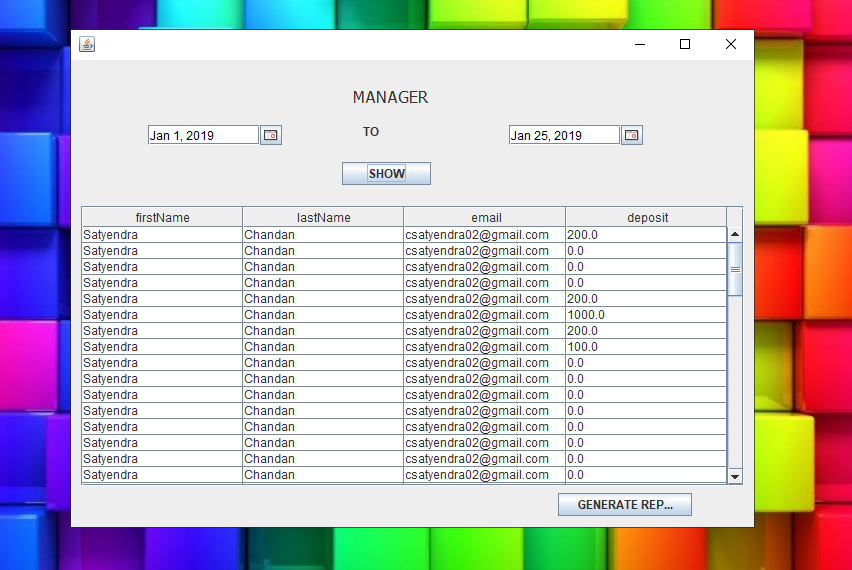


Office advisor can see client information

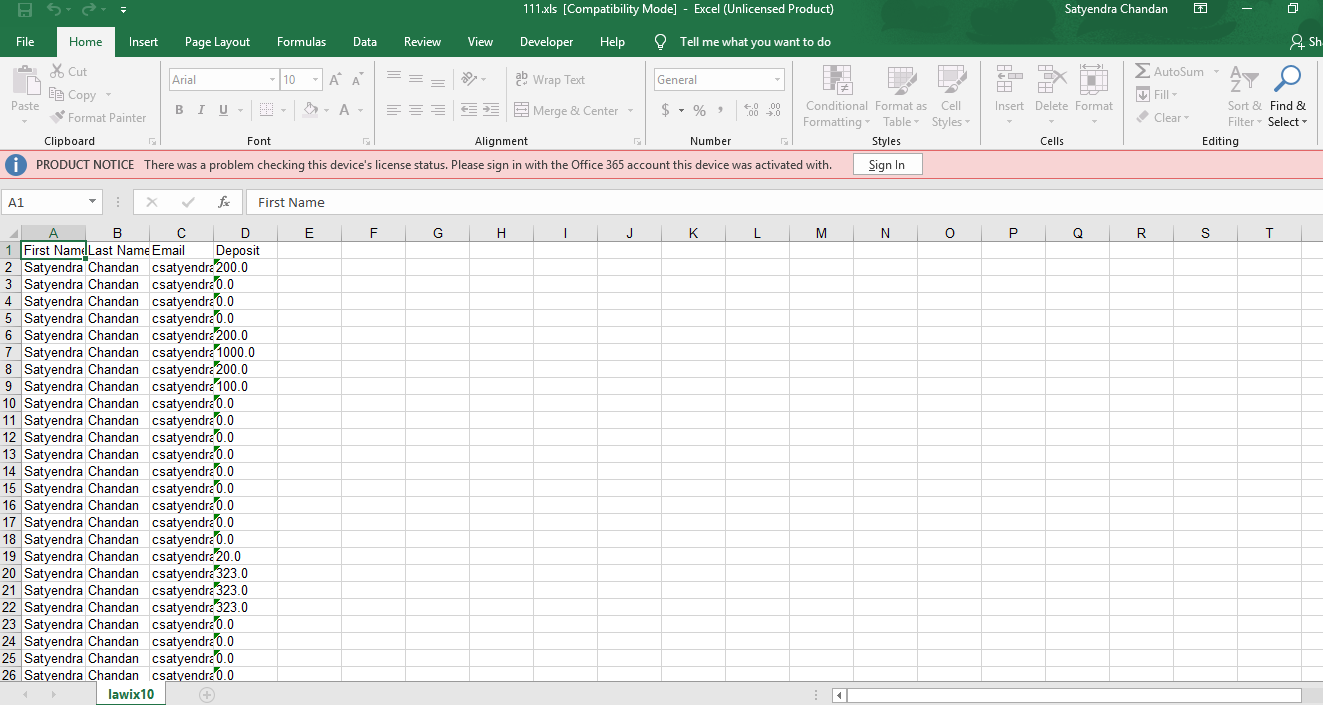


Office advisor can see total transaction details of clients

TO VIEW CLIENTS DETAILS BY MANAGER:

* Manager can see client information by logging with his/her mail ID & Password and choose manager option as shown in above image.
* By logging he can see all client names and their deposited amount.

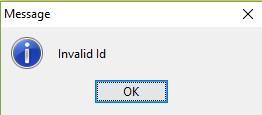


Client deposited amount details in Excel format.

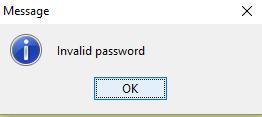
* VALIDATION CHECK

Following message box will display when there is a violation of field validation.

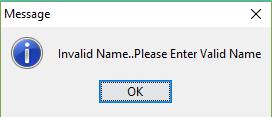
* When the user ID is empty or incorrect;



* When the user Password is empty or incorrect;



* When the customer name field is empty (it is a mandatory field);



**CONCLUSION AND FUTURE ENHANCEMENT**

**CONCLUSION**

We conclude to build a sealable interface that simplifies the banking system handling. It has the following objectives:

Improved access to all transection and other operation. Easy access and handling of customers account.

Detailed access of customer is easy for accountant and all customer and accountant details access is simple for Manager.



**FUTURE ENHANCEMENT**

The file managed containing all the information regarding:

* Customers
* Accountant
* Transactions

It will always be updated with the recently added details.

We are looking forward to add following features to this software:

* Online banking.
* SMS alerts on customer’s registered mobile number.
* Online shopping payment gateway.
* Customer can video chat with accountant or the manager.

**SOURCE CODE**

**Login:**

package bankingsystemswing;

import java.awt.Button;

import java.awt.EventQueue;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Random;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import org.apache.commons.codec.digest.DigestUtils;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import javax.swing.JPasswordField;

import javax.swing.JRadioButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.awt.Font;

import java.awt.Color;

public class Login {

JFrame frame;

private JTextField textField;

private JPasswordField passwordField;

private String emailID;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Login window = new Login();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public Login() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

Connection connection = null;

private String accountType;

private String accountNumber;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e1) {

e1.printStackTrace();

} catch (SQLException e1) {

e1.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 450, 300);

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

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("Email");

lblNewLabel.setBounds(63, 77, 89, 14);

frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("SWISS BANK SYSTEM");

lblNewLabel\_1.setForeground(Color.BLUE);

lblNewLabel\_1.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblNewLabel\_1.setBounds(151, 11, 212, 27);

frame.getContentPane().add(lblNewLabel\_1);

textField = new JTextField();

textField.setBounds(173, 74, 155, 20);

frame.getContentPane().add(textField);

textField.setColumns(10);

// Encrpyted password

passwordField = new JPasswordField();

passwordField.setBounds(173, 111, 155, 20);

frame.getContentPane().add(passwordField);

String encryptedPassword = DigestUtils.md5Hex(passwordField.getText());

JLabel lblNewLabel\_2 = new JLabel("password");

lblNewLabel\_2.setBounds(63, 114, 89, 14);

frame.getContentPane().add(lblNewLabel\_2);

JButton btnNewButton = new JButton("Login");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

emailID = textField.getText();

AddMoneyClient amc = new AddMoneyClient();

amc.setEmailID(emailID);

String query = "select email, password,account\_type from accounts where email=? and password =? and account\_type = ? ";

try {

PreparedStatement pst = connection.prepareStatement(query);

pst.setString(1, textField.getText());

pst.setString(2, encryptedPassword );

pst.setString(3, accountType);

ResultSet rs = pst.executeQuery();

int count = 0;

while (rs.next()) {

count++;

accountType = rs.getString("account\_type");

}

if (count == 1 && accountType.equals("C")) {

GlobalVariable.getInstance().setLogin(textField.getText());

frame.dispose();

ClientInformation window = new ClientInformation();

window.frmClientInformation.setVisible(true);

} else if (count == 1 && accountType.equals("M")) {

frame.dispose();

Manager1 window = new Manager1();

window.frame.setVisible(true);

} else if (count == 1 && accountType.equals("A")) {

frame.dispose();

officeAdvisor window = new officeAdvisor();

window.frame.setVisible(true);

} else if (count > 1 || count == 0) {

JOptionPane.showMessageDialog(frame, "Enter Correct password");

}

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

btnNewButton.setBounds(176, 178, 89, 23);

frame.getContentPane().add(btnNewButton);

JRadioButton rbClient = new JRadioButton("Client");

rbClient.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (rbClient.isSelected())

accountType = "C";

}

});

rbClient.setBounds(94, 148, 89, 23);

frame.getContentPane().add(rbClient);

JRadioButton rbAdvisor = new JRadioButton("Advisor");

rbAdvisor.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (rbAdvisor.isSelected())

accountType = "A";

}

});

rbAdvisor.setBounds(193, 148, 92, 23);

frame.getContentPane().add(rbAdvisor);

JRadioButton rbManager = new JRadioButton("Manager");

rbManager.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (rbManager.isSelected())

accountType = "M";

}

});

rbManager.setBounds(295, 148, 109, 23);

frame.getContentPane().add(rbManager);

ButtonGroup bg = new ButtonGroup();

bg.add(rbClient);

bg.add(rbAdvisor);

bg.add(rbManager);

// CREATE ACCOUNT FRAME

JButton btnCreateAccount = new JButton("CREATE ACCOUNT");

btnCreateAccount.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frame.dispose();

CreateAccount ca = new CreateAccount();

ca.frame.setVisible(true);

}

});

btnCreateAccount.setBounds(282, 227, 142, 23);

frame.getContentPane().add(btnCreateAccount);

JLabel lblNewLabel\_3 = new JLabel("Login");

lblNewLabel\_3.setBounds(203, 49, 46, 14);

frame.getContentPane().add(lblNewLabel\_3);

}

}

**Create Account :**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JTextField;

import org.apache.commons.codec.digest.DigestUtils;

import javax.swing.JPasswordField;

import javax.swing.JRadioButton;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.util.Random;

import java.awt.event.ActionEvent;

public class CreateAccount {

JFrame frame;

private JTextField textField\_2;

private JTextField textField\_3;

private JPasswordField passwordField;

private JPasswordField passwordField\_1;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

CreateAccount window = new CreateAccount();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public CreateAccount() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

Connection connection = null;

private String person;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 634, 367);

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

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("CREATE AN ACCOUNT");

lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 16));

lblNewLabel.setBounds(224, 21, 187, 41);

frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_3 = new JLabel("UserName");

lblNewLabel\_3.setBounds(39, 103, 79, 14);

frame.getContentPane().add(lblNewLabel\_3);

textField\_2 = new JTextField();

textField\_2.setBounds(129, 100, 122, 20);

frame.getContentPane().add(textField\_2);

textField\_2.setColumns(10);

JLabel lblNewLabel\_4 = new JLabel("Email");

lblNewLabel\_4.setBounds(311, 103, 46, 14);

frame.getContentPane().add(lblNewLabel\_4);

textField\_3 = new JTextField();

textField\_3.setBounds(367, 100, 151, 20);

frame.getContentPane().add(textField\_3);

textField\_3.setColumns(10);

JLabel lblNewLabel\_5 = new JLabel("Password");

lblNewLabel\_5.setBounds(39, 175, 65, 14);

frame.getContentPane().add(lblNewLabel\_5);

//Password to md5

passwordField = new JPasswordField();

passwordField.setBounds(129, 172, 122, 20);

frame.getContentPane().add(passwordField);

String encrpytedPassword = DigestUtils.md5Hex(passwordField.getText());

JLabel lblNewLabel\_6 = new JLabel("Confirm Password");

lblNewLabel\_6.setBounds(263, 175, 117, 14);

frame.getContentPane().add(lblNewLabel\_6);

passwordField\_1 = new JPasswordField();

passwordField\_1.setBounds(390, 172, 122, 20);

frame.getContentPane().add(passwordField\_1);

JLabel lblNewLabel\_7 = new JLabel("Account Type");

lblNewLabel\_7.setBounds(112, 224, 79, 14);

frame.getContentPane().add(lblNewLabel\_7);

JRadioButton rdbtnNewRadioButton = new JRadioButton("Client");

rdbtnNewRadioButton.addActionListener(new ActionListener() {

// Selection of the radio button

public void actionPerformed(ActionEvent arg0) {

if (rdbtnNewRadioButton.isSelected())

person = "C";

}

});

rdbtnNewRadioButton.setBounds(208, 220, 65, 23);

frame.getContentPane().add(rdbtnNewRadioButton);

JRadioButton rdbtnNewRadioButton\_1 = new JRadioButton("Office Advisor");

rdbtnNewRadioButton\_1.addActionListener(new ActionListener() {

// Radio button selection Office Advisor

public void actionPerformed(ActionEvent e) {

if (rdbtnNewRadioButton\_1.isSelected())

person = "A";

}

});

rdbtnNewRadioButton\_1.setBounds(275, 220, 105, 23);

frame.getContentPane().add(rdbtnNewRadioButton\_1);

JRadioButton rdbtnNewRadioButton\_2 = new JRadioButton("Manager");

rdbtnNewRadioButton\_2.addActionListener(new ActionListener() {

// Radio button selection Manager

public void actionPerformed(ActionEvent e) {

if (rdbtnNewRadioButton\_2.isSelected())

person = "M";

}

});

rdbtnNewRadioButton\_2.setBounds(381, 220, 109, 23);

frame.getContentPane().add(rdbtnNewRadioButton\_2);

// Button Group

ButtonGroup bg = new ButtonGroup();

bg.add(rdbtnNewRadioButton);

bg.add(rdbtnNewRadioButton\_1);

bg.add(rdbtnNewRadioButton\_2);

// Create Account button

Random rand = new Random();

JButton btnNewButton = new JButton("CREATE");

btnNewButton.addActionListener(new ActionListener() {

// Create an account button action

public void actionPerformed(ActionEvent e) {

// for creating an account number

String accountNumber = null;

String IBAN = null;

String BIC = null;

accountNumber = "SWISSFR" + rand.nextInt(9999) + 1;

BIC = "" + rand.nextInt(9999) + 1;

IBAN = "FR" + rand.nextInt(9999) + BIC + 1 + rand.nextInt(9999) + 1 + accountNumber;

if (person.equals("C")) {

String query = "insert into accounts(email,password,account\_type,username) values(?,?,?,?)";

String query2 = "insert into clienttable(accountNo,iban,BIC,accountType,email) values (?,?,?,?,?)";

try {

PreparedStatement pst = connection.prepareStatement(query);

pst.setString(1, textField\_3.getText());

pst.setString(2, encrpytedPassword);

pst.setString(3, person);

pst.setString(4, textField\_2.getText());

pst.execute();

pst = connection.prepareStatement(query2);

pst.setString(1, accountNumber);

pst.setString(2, IBAN);

pst.setString(3, BIC);

pst.setString(4, person);

pst.setString(5, textField\_3.getText());

pst.execute();

} catch (SQLException e1) {

e1.printStackTrace();

}

}

if(person.equals("A") || person.equals("M"))

{

String query = "insert into accounts(email,password,account\_type,username) values(?,?,?,?)";

try {

PreparedStatement pst = connection.prepareStatement(query);

pst.setString(1, textField\_3.getText());

pst.setString(2, encrpytedPassword);

pst.setString(3, person);

pst.setString(4, textField\_2.getText());

pst.execute();

} catch (SQLException e1) {

e1.printStackTrace();

}

}

JOptionPane.showMessageDialog(frame, "Account Created Successfully");

frame.dispose();

Login log = new Login();

log.frame.setVisible(true);

}

});

btnNewButton.setBounds(268, 266, 89, 23);

frame.getContentPane().add(btnNewButton);

// BACK BUTTON TAKES TO the login screen

JButton btnNewButton\_1 = new JButton("BACK");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frame.dispose();

Login login1 = new Login();

login1.frame.setVisible(true);

}

});

btnNewButton\_1.setBounds(10, 294, 89, 23);

frame.getContentPane().add(btnNewButton\_1);

}

}

**Client Information:**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import java.awt.BorderLayout;

import javax.swing.JLabel;

import javax.swing.JButton;

import java.awt.Panel;

import javax.swing.JTabbedPane;

import java.awt.Font;

import java.awt.Color;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.awt.event.ActionEvent;

public class ClientInformation {

JFrame frmClientInformation;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ClientInformation window = new ClientInformation();

window.frmClientInformation.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public ClientInformation() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

Connection connection = null;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

} catch (SQLException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

frmClientInformation = new JFrame();

frmClientInformation.setTitle("Client Information");

frmClientInformation.setResizable(false);

frmClientInformation.setBounds(100, 100, 819, 499);

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

frmClientInformation.getContentPane().setLayout(null);

JLabel lblSwissBank = new JLabel("SWISS BANK");

lblSwissBank.setForeground(Color.BLUE);

lblSwissBank.setFont(new Font("Tahoma", Font.PLAIN, 18));

lblSwissBank.setBounds(10, 11, 145, 40);

frmClientInformation.getContentPane().add(lblSwissBank);

JButton btnNewButton = new JButton("INFORMATION");

btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 8));

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

}

});

btnNewButton.setBounds(10, 54, 89, 23);

frmClientInformation.getContentPane().add(btnNewButton);

JButton btnNewButton\_1 = new JButton("TRANSACTION");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frmClientInformation.dispose();

Transaction transaction = new Transaction();

transaction.main(null);

}

});

btnNewButton\_1.setFont(new Font("Tahoma", Font.PLAIN, 8));

btnNewButton\_1.setBounds(10, 88, 89, 23);

frmClientInformation.getContentPane().add(btnNewButton\_1);

JButton btnNewButton\_2 = new JButton("TRANSFER");

btnNewButton\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frmClientInformation.dispose();

TransferMoney transfermoney = new TransferMoney();

transfermoney.frame.setVisible(true);

}

});

btnNewButton\_2.setFont(new Font("Tahoma", Font.PLAIN, 8));

btnNewButton\_2.setBounds(10, 122, 89, 23);

frmClientInformation.getContentPane().add(btnNewButton\_2);

JPanel panel = new JPanel();

panel.setBounds(109, 54, 694, 331);

frmClientInformation.getContentPane().add(panel);

panel.setLayout(null);

JLabel lblNewLabel = new JLabel("ACC No:");

lblNewLabel.setBounds(10, 24, 71, 14);

panel.add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("EmailID:");

lblNewLabel\_1.setBounds(10, 49, 71, 14);

panel.add(lblNewLabel\_1);

JLabel lblNewLabel\_2 = new JLabel("Telephone:");

lblNewLabel\_2.setBounds(10, 99, 71, 14);

panel.add(lblNewLabel\_2);

JLabel lbltelephone = new JLabel("New label");

lbltelephone.setFont(new Font("Tahoma", Font.PLAIN, 11));

lbltelephone.setBounds(123, 99, 204, 14);

panel.add(lbltelephone);

JLabel lblNewLabel\_5 = new JLabel("Address:");

lblNewLabel\_5.setBounds(48, 124, 71, 14);

panel.add(lblNewLabel\_5);

JLabel lblAccno = new JLabel("New label");

lblAccno.setFont(new Font("Tahoma", Font.PLAIN, 11));

lblAccno.setBounds(125, 24, 202, 14);

panel.add(lblAccno);

JLabel lblemail = new JLabel("New label");

lblemail.setFont(new Font("Tahoma", Font.PLAIN, 11));

lblemail.setBounds(123, 49, 204, 14);

panel.add(lblemail);

JLabel lblGender = new JLabel("Gender:");

lblGender.setBounds(10, 74, 71, 14);

panel.add(lblGender);

JLabel lblgender = new JLabel("New label");

lblgender.setFont(new Font("Tahoma", Font.PLAIN, 11));

lblgender.setBounds(123, 74, 204, 14);

panel.add(lblgender);

JLabel lblNewLabel\_4 = new JLabel("House No:");

lblNewLabel\_4.setBounds(10, 149, 73, 14);

panel.add(lblNewLabel\_4);

JLabel lblNewLabel\_6 = new JLabel("Neighbourhood:");

lblNewLabel\_6.setBounds(10, 174, 98, 14);

panel.add(lblNewLabel\_6);

JLabel lblNewLabel\_8 = new JLabel("City:");

lblNewLabel\_8.setBounds(10, 199, 71, 14);

panel.add(lblNewLabel\_8);

JLabel lblNewLabel\_9 = new JLabel("Country:");

lblNewLabel\_9.setBounds(10, 226, 71, 14);

panel.add(lblNewLabel\_9);

JLabel lblNewLabel\_10 = new JLabel("Zip Code:");

lblNewLabel\_10.setBounds(10, 251, 71, 14);

panel.add(lblNewLabel\_10);

JLabel lblNewLabel\_11 = new JLabel("Occupation:");

lblNewLabel\_11.setBounds(10, 276, 84, 14);

panel.add(lblNewLabel\_11);

JLabel lblNewLabel\_12 = new JLabel("Mobile:");

lblNewLabel\_12.setBounds(10, 301, 71, 14);

panel.add(lblNewLabel\_12);

JLabel lblhouse = new JLabel("New label");

lblhouse.setFont(new Font("Tahoma", Font.PLAIN, 11));

lblhouse.setBounds(123, 149, 204, 14);

panel.add(lblhouse);

JLabel lblneighbour = new JLabel("New label");

lblneighbour.setFont(new Font("Tahoma", Font.PLAIN, 11));

lblneighbour.setBounds(123, 174, 204, 14);

panel.add(lblneighbour);

JLabel lblcity = new JLabel("New label");

lblcity.setBounds(123, 199, 204, 14);

panel.add(lblcity);

JLabel lblcountry = new JLabel("New label");

lblcountry.setBounds(123, 226, 204, 14);

panel.add(lblcountry);

JLabel lblzip = new JLabel("New label");

lblzip.setBounds(125, 251, 202, 14);

panel.add(lblzip);

JLabel lblOccupation = new JLabel("New label");

lblOccupation.setFont(new Font("Tahoma", Font.PLAIN, 11));

lblOccupation.setBounds(123, 276, 204, 14);

panel.add(lblOccupation);

JLabel lblMobile = new JLabel("New label");

lblMobile.setBounds(123, 301, 204, 14);

panel.add(lblMobile);

JLabel lblNewLabel\_13 = new JLabel("Nationality:");

lblNewLabel\_13.setBounds(386, 24, 78, 14);

panel.add(lblNewLabel\_13);

JLabel lblNewLabel\_14 = new JLabel("Married:");

lblNewLabel\_14.setBounds(386, 49, 78, 14);

panel.add(lblNewLabel\_14);

JLabel lblNewLabel\_15 = new JLabel("Spouce Name:");

lblNewLabel\_15.setBounds(386, 74, 98, 14);

panel.add(lblNewLabel\_15);

JLabel lblNewLabel\_16 = new JLabel("Birth Place:");

lblNewLabel\_16.setBounds(386, 99, 78, 14);

panel.add(lblNewLabel\_16);

JLabel lblNewLabel\_17 = new JLabel("Children:");

lblNewLabel\_17.setBounds(386, 124, 78, 14);

panel.add(lblNewLabel\_17);

JLabel lblNewLabel\_18 = new JLabel("Bank Agency:");

lblNewLabel\_18.setBounds(386, 149, 78, 14);

panel.add(lblNewLabel\_18);

JLabel lblNewLabel\_19 = new JLabel("DOB:");

lblNewLabel\_19.setBounds(386, 174, 78, 14);

panel.add(lblNewLabel\_19);

JLabel lblNewLabel\_20 = new JLabel("IBAN");

lblNewLabel\_20.setBounds(455, 276, 46, 14);

panel.add(lblNewLabel\_20);

JLabel lbliban = new JLabel("New label");

lbliban.setFont(new Font("Tahoma", Font.PLAIN, 14));

lbliban.setBounds(337, 294, 347, 25);

panel.add(lbliban);

JLabel lblNewLabel\_22 = new JLabel("BIC Code:");

lblNewLabel\_22.setBounds(386, 251, 78, 14);

panel.add(lblNewLabel\_22);

JLabel lblnationality = new JLabel("New label");

lblnationality.setBounds(497, 24, 187, 14);

panel.add(lblnationality);

JLabel lblmarried = new JLabel("New label");

lblmarried.setBounds(497, 49, 187, 14);

panel.add(lblmarried);

JLabel lblSpouce = new JLabel("New label");

lblSpouce.setBounds(497, 74, 187, 14);

panel.add(lblSpouce);

JLabel lblBirthPlace = new JLabel("New label");

lblBirthPlace.setBounds(497, 99, 187, 14);

panel.add(lblBirthPlace);

JLabel lblChildren = new JLabel("New label");

lblChildren.setBounds(497, 124, 187, 14);

panel.add(lblChildren);

JLabel lblAgency = new JLabel("New label");

lblAgency.setBounds(497, 149, 187, 14);

panel.add(lblAgency);

JLabel lblDOB = new JLabel("New label");

lblDOB.setBounds(497, 174, 187, 14);

panel.add(lblDOB);

JLabel lblBIC = new JLabel("New label");

lblBIC.setBounds(497, 251, 187, 14);

panel.add(lblBIC);

JLabel lblFirstName = new JLabel("New label");

lblFirstName.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblFirstName.setBounds(266, 20, 238, 14);

frmClientInformation.getContentPane().add(lblFirstName);

JLabel lblNewLabel\_7 = new JLabel("BALANCE");

lblNewLabel\_7.setBounds(227, 412, 83, 14);

frmClientInformation.getContentPane().add(lblNewLabel\_7);

// Edit the information page

JButton btnEdit = new JButton("Edit");

btnEdit.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frmClientInformation.dispose();

ClientInformationEdit cfe = new ClientInformationEdit();

cfe.frame.setVisible(true);

}

});

btnEdit.setBounds(681, 408, 89, 23);

frmClientInformation.getContentPane().add(btnEdit);

JButton btnAddMoney = new JButton("ADD MONEY");

btnAddMoney.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frmClientInformation.dispose();

AddMoneyClient.main(null);

}

});

btnAddMoney.setFont(new Font("Tahoma", Font.PLAIN, 8));

btnAddMoney.setBounds(10, 156, 89, 23);

frmClientInformation.getContentPane().add(btnAddMoney);

JLabel lblLastName = new JLabel("New label");

lblLastName.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblLastName.setBounds(514, 20, 221, 14);

frmClientInformation.getContentPane().add(lblLastName);

JLabel label = new JLabel("New label");

label.setFont(new Font("Tahoma", Font.PLAIN, 16));

label.setBounds(354, 408, 221, 19);

frmClientInformation.getContentPane().add(label);

JLabel lblNewLabel\_3 = new JLabel("\u20AC");

lblNewLabel\_3.setFont(new Font("Tahoma", Font.PLAIN, 16));

lblNewLabel\_3.setBounds(320, 408, 24, 19);

frmClientInformation.getContentPane().add(lblNewLabel\_3);

// To get the data from the database

//String email11 = ;

//String query = "select firstName,lastName,dateOfBirth,gender,telephone,house,neighbourhood,city,country,zipcode,occupation,mobile,nationality,Married,spouceName,birthPlace,children,bankAgency,email,iban,BIC,accountNo from clienttable where email= ?";

String query = "SELECT \* FROM clienttable WHERE email = ?";

String query2 = "select balance from balancetable where email =?";

ResultSet rs = null;

PreparedStatement pst = null;

try {

pst = connection.prepareStatement(query);

System.out.println(GlobalVariable.getInstance().getLogin());

pst.setString(1, GlobalVariable.getInstance().getLogin());

//System.out.println(email11);

pst.setString(1, GlobalVariable.getInstance().getLogin());

rs = pst.executeQuery();

while(rs.next())

{

lblFirstName.setText(rs.getString("firstName"));

lblLastName.setText(rs.getString("lastName"));

lblAccno.setText(rs.getString("accountNo"));

lblemail.setText(rs.getString("email"));

lblgender.setText(rs.getString("gender"));

lbltelephone.setText(rs.getString("telephone"));

lblhouse.setText(rs.getString("house"));

lblneighbour.setText(rs.getString("neighbourhood"));

lblcity.setText(rs.getString("city"));

lblcountry.setText(rs.getString("country"));

lblzip.setText(rs.getString("zipcode"));

lblOccupation.setText(rs.getString("occupation"));

lblMobile.setText(rs.getString("mobile"));

lblnationality.setText(rs.getString("nationality"));

lblmarried.setText(rs.getString("Married"));

lblSpouce.setText(rs.getString("spouceName"));

lblBirthPlace.setText(rs.getString("birthPlace"));

lblChildren.setText(rs.getString("children"));

lblAgency.setText(rs.getString("bankAgency"));

lblDOB.setText(rs.getString("dateOfBirth"));

lblBIC.setText(rs.getString("BIC"));

lbliban.setText(rs.getString("iban"));

}

pst = connection.prepareStatement(query2);

pst.setString(1, GlobalVariable.getInstance().getLogin());

rs = pst.executeQuery();

while(rs.next())

{

label.setText(rs.getString("balance"));

}

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} finally {

// pst.close();

// rs.close();

}

}

}

**Client Information Edit:**

public class ClientInformationEdit {

JFrame frame;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_4;

private JTextField textFieldhouse;

private JTextField textFieldneighbourhood;

private JTextField textFieldcity;

private JTextField textField\_8;

private JTextField textField\_9;

private JTextField textField\_10;

private JTextField textField\_11;

private JTextField textField\_12;

private JTextField textField\_13;

private JTextField textField\_14;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ClientInformationEdit window = new ClientInformationEdit();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public ClientInformationEdit() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private String married;

private String gender;

Connection connection = null;

private String accountType;

private JTextField textFieldCountry;

private JTextField textFieldZipcode;

private JTextField textField\_2;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

} catch (SQLException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 604, 457);

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

frame.getContentPane().setLayout(null);

JLabel lblClientInformationEdit = new JLabel("CLIENT INFORMATION EDIT");

lblClientInformationEdit.setForeground(Color.BLUE);

lblClientInformationEdit.setFont(new Font("Tahoma", Font.PLAIN, 16));

lblClientInformationEdit.setBounds(107, 5, 283, 31);

frame.getContentPane().add(lblClientInformationEdit);

JLabel lblNewLabel = new JLabel("First Name");

lblNewLabel.setBounds(25, 47, 65, 14);

frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("Last Name");

lblNewLabel\_1.setBounds(25, 75, 65, 14);

frame.getContentPane().add(lblNewLabel\_1);

JLabel lblNewLabel\_2 = new JLabel("DOB");

lblNewLabel\_2.setBounds(25, 126, 46, 14);

frame.getContentPane().add(lblNewLabel\_2);

JLabel lblNewLabel\_4 = new JLabel("Gender");

lblNewLabel\_4.setBounds(25, 185, 46, 14);

frame.getContentPane().add(lblNewLabel\_4);

textField = new JTextField();

textField.setBounds(135, 47, 117, 20);

frame.getContentPane().add(textField);

textField.setColumns(10);

textField\_1 = new JTextField();

textField\_1.setBounds(135, 75, 117, 20);

frame.getContentPane().add(textField\_1);

textField\_1.setColumns(10);

JRadioButton rdbtnMale = new JRadioButton("Male");

rdbtnMale.addActionListener(new ActionListener() {

// gender radiobutton selection

public void actionPerformed(ActionEvent e) {

if (rdbtnMale.isSelected())

gender = "M";

}

});

rdbtnMale.setBounds(112, 181, 58, 23);

frame.getContentPane().add(rdbtnMale);

JRadioButton rdbtnfemale = new JRadioButton("Female");

rdbtnfemale.addActionListener(new ActionListener() {

// gender radiobutton selection

public void actionPerformed(ActionEvent e) {

if (rdbtnfemale.isSelected())

gender = "F";

}

});

rdbtnfemale.setBounds(172, 181, 59, 23);

frame.getContentPane().add(rdbtnfemale);

ButtonGroup bg1 = new ButtonGroup();

bg1.add(rdbtnMale);

bg1.add(rdbtnfemale);

JLabel lblNewLabel\_5 = new JLabel("Telephone");

lblNewLabel\_5.setBounds(25, 220, 65, 14);

frame.getContentPane().add(lblNewLabel\_5);

textField\_4 = new JTextField();

textField\_4.setBounds(135, 217, 117, 20);

frame.getContentPane().add(textField\_4);

textField\_4.setColumns(10);

JLabel lblAddress = new JLabel("Address");

lblAddress.setBounds(442, 11, 66, 14);

frame.getContentPane().add(lblAddress);

textFieldhouse = new JTextField();

textFieldhouse.setBounds(412, 36, 138, 20);

frame.getContentPane().add(textFieldhouse);

textFieldhouse.setColumns(10);

textFieldneighbourhood = new JTextField();

textFieldneighbourhood.setBounds(412, 67, 138, 20);

frame.getContentPane().add(textFieldneighbourhood);

textFieldneighbourhood.setColumns(10);

textFieldcity = new JTextField();

textFieldcity.setBounds(412, 102, 138, 20);

frame.getContentPane().add(textFieldcity);

textFieldcity.setColumns(10);

JLabel lblNewLabel\_6 = new JLabel("Occupation");

lblNewLabel\_6.setBounds(324, 215, 78, 14);

frame.getContentPane().add(lblNewLabel\_6);

textField\_8 = new JTextField();

textField\_8.setBounds(421, 209, 116, 20);

frame.getContentPane().add(textField\_8);

textField\_8.setColumns(10);

JLabel lblNewLabel\_7 = new JLabel("Mobile");

lblNewLabel\_7.setBounds(344, 253, 46, 14);

frame.getContentPane().add(lblNewLabel\_7);

textField\_9 = new JTextField();

textField\_9.setBounds(421, 247, 116, 20);

frame.getContentPane().add(textField\_9);

textField\_9.setColumns(10);

JLabel lblNewLabel\_8 = new JLabel("Nationality");

lblNewLabel\_8.setBounds(324, 288, 78, 14);

frame.getContentPane().add(lblNewLabel\_8);

textField\_10 = new JTextField();

textField\_10.setBounds(421, 282, 116, 20);

frame.getContentPane().add(textField\_10);

textField\_10.setColumns(10);

JLabel lblNewLabel\_9 = new JLabel("Married");

lblNewLabel\_9.setBounds(25, 256, 46, 14);

frame.getContentPane().add(lblNewLabel\_9);

JRadioButton rdbtnYes = new JRadioButton("Yes");

rdbtnYes.addActionListener(new ActionListener() {

// Married radiobutton action

public void actionPerformed(ActionEvent e) {

if (rdbtnYes.isSelected())

married = "yes";

}

});

rdbtnYes.setBounds(122, 252, 48, 23);

frame.getContentPane().add(rdbtnYes);

JRadioButton rdbtnNo = new JRadioButton("No");

rdbtnNo.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (rdbtnNo.isSelected())

married = "no";

}

});

rdbtnNo.setBounds(170, 252, 46, 23);

frame.getContentPane().add(rdbtnNo);

ButtonGroup bg2 = new ButtonGroup();

bg2.add(rdbtnYes);

bg2.add(rdbtnNo);

JLabel lblSpouceName = new JLabel("Spouce Name:");

lblSpouceName.setBounds(25, 291, 97, 14);

frame.getContentPane().add(lblSpouceName);

textField\_11 = new JTextField();

textField\_11.setBounds(132, 288, 120, 20);

frame.getContentPane().add(textField\_11);

textField\_11.setColumns(10);

JLabel lblBirthPlace = new JLabel("Birth Place");

lblBirthPlace.setBounds(324, 324, 78, 14);

frame.getContentPane().add(lblBirthPlace);

textField\_12 = new JTextField();

textField\_12.setBounds(421, 318, 116, 20);

frame.getContentPane().add(textField\_12);

textField\_12.setColumns(10);

JLabel lblChildren = new JLabel("Children");

lblChildren.setBounds(324, 359, 66, 14);

frame.getContentPane().add(lblChildren);

textField\_13 = new JTextField();

textField\_13.setBounds(421, 353, 116, 20);

frame.getContentPane().add(textField\_13);

textField\_13.setColumns(10);

JLabel lblBankAgency = new JLabel("Bank Agency");

lblBankAgency.setBounds(25, 316, 89, 14);

frame.getContentPane().add(lblBankAgency);

textField\_14 = new JTextField();

textField\_14.setBounds(130, 313, 122, 20);

frame.getContentPane().add(textField\_14);

textField\_14.setColumns(10);

JButton btnDone = new JButton("Done");

btnDone.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// inserting into the database

String query ="update clienttable set firstName = ?,lastName = ?,dateOfBirth = ?,gender = ?,telephone = ?,house = ?,neighbourhood = ?,city = ?,country = ?, zipcode = ?,occupation = ?,mobile = ?,nationality = ?,Married = ?,spouceName = ?,birthPlace = ?,children = ?,bankAgency = ? where email = ?";

try {

PreparedStatement pst= connection.prepareStatement(query);

pst.setString(1,textField.getText());

pst.setString(2,textField\_1.getText());

pst.setString(3,textField\_2.getText());

pst.setString(4, gender);

pst.setString(5,textField\_4.getText() );

pst.setString(6,textFieldhouse.getText() );

pst.setString(7,textFieldneighbourhood.getText() );

pst.setString(8,textFieldcity.getText() );

pst.setString(9,textFieldCountry.getText() );

pst.setString(10,textFieldZipcode.getText() );

pst.setString(11, textField\_8.getText());

pst.setString(12, textField\_9.getText());

pst.setString(13, textField\_10.getText());

pst.setString(14, married);

pst.setString(15, textField\_11.getText() );

pst.setString(16, textField\_12.getText());

pst.setString(17,textField\_13.getText());

pst.setString(18, textField\_14.getText());

pst.setString(19, GlobalVariable.getInstance().getLogin());

pst.executeUpdate();

JOptionPane.showMessageDialog(null, "Data updated successfully");

} catch (SQLException e1) {

e1.printStackTrace(); }

}

});

btnDone.setBounds(489, 384, 89, 23);

frame.getContentPane().add(btnDone);

// Back button takes to the login page

JButton btnNewButton = new JButton("Back");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frame.dispose();

ClientInformation cif = new ClientInformation();

cif.frmClientInformation.setVisible(true);

}

});

btnNewButton.setBounds(10, 384, 89, 23);

frame.getContentPane().add(btnNewButton);

JLabel lblNewLabel\_3 = new JLabel("Password");

lblNewLabel\_3.setBounds(25, 151, 78, 14);

frame.getContentPane().add(lblNewLabel\_3);

//Change password

JButton btnNewButton\_1 = new JButton("Change Password");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

ChangePassword cp = new ChangePassword();

cp.frame.setVisible(true);

}

});

btnNewButton\_1.setBounds(135, 147, 117, 23);

frame.getContentPane().add(btnNewButton\_1);

textFieldCountry = new JTextField();

textFieldCountry.setBounds(412, 133, 138, 20);

frame.getContentPane().add(textFieldCountry);

textFieldCountry.setColumns(10);

textFieldZipcode = new JTextField();

textFieldZipcode.setBounds(412, 164, 138, 20);

frame.getContentPane().add(textFieldZipcode);

textFieldZipcode.setColumns(10);

JLabel lblNewLabel\_10 = new JLabel("House");

lblNewLabel\_10.setBounds(326, 39, 76, 14);

frame.getContentPane().add(lblNewLabel\_10);

JLabel lblNewLabel\_11 = new JLabel("Neighbourhood");

lblNewLabel\_11.setBounds(283, 70, 119, 14);

frame.getContentPane().add(lblNewLabel\_11);

JLabel lblNewLabel\_12 = new JLabel("City");

lblNewLabel\_12.setBounds(293, 105, 97, 14);

frame.getContentPane().add(lblNewLabel\_12);

JLabel lblNewLabel\_13 = new JLabel("Country");

lblNewLabel\_13.setBounds(303, 136, 99, 14);

frame.getContentPane().add(lblNewLabel\_13);

JLabel lblNewLabel\_14 = new JLabel("Zip Code");

lblNewLabel\_14.setBounds(301, 167, 101, 14);

frame.getContentPane().add(lblNewLabel\_14);

textField\_2 = new JTextField();

textField\_2.setBounds(135, 116, 86, 20);

frame.getContentPane().add(textField\_2);

textField\_2.setColumns(10);

JLabel lblNewLabel\_15 = new JLabel("YYYY-MM-DD");

lblNewLabel\_15.setBounds(135, 102, 117, 14);

frame.getContentPane().add(lblNewLabel\_15);

String query3 = "select firstName,lastName,dateOfBirth,telephone,house,neighbourhood,city,country,zipcode,occupation,mobile,nationality,spouceName,birthPlace,children,bankAgency,gender,Married from clienttable where email = ?";

try {

PreparedStatement pst3 = connection.prepareStatement(query3);

pst3.setString(1, GlobalVariable.getInstance().getLogin());

ResultSet rs2 = pst3.executeQuery();

while (rs2.next()) {

textField.setText(rs2.getString("firstName"));

textField\_1.setText(rs2.getString("lastName"));

textField\_2.setText(rs2.getString("dateOfBirth"));

textField\_4.setText(rs2.getString("telephone"));

textFieldhouse.setText(rs2.getString("house"));

textFieldneighbourhood.setText(rs2.getString("neighbourhood"));

textFieldcity.setText(rs2.getString("city"));

textFieldCountry.setText(rs2.getString("country"));

textFieldZipcode.setText(rs2.getString("zipcode"));

textField\_8.setText(rs2.getString("occupation"));

textField\_9.setText(rs2.getString("mobile"));

textField\_10.setText(rs2.getString("nationality"));

textField\_11.setText(rs2.getString("spouceName"));

textField\_12.setText(rs2.getString("birthPlace"));

textField\_13.setText(rs2.getString("children"));

textField\_14.setText(rs2.getString("bankAgency"));

if(rs2.getString("gender").equals("M"))

{

rdbtnMale.setSelected(true);

rdbtnfemale.setSelected(false);

}else

{

rdbtnMale.setSelected(false);

rdbtnfemale.setSelected(true);

}

if(rs2.getString("Married").equals("yes"))

{

rdbtnYes.setSelected(true);

rdbtnNo.setSelected(false);

}else

{

rdbtnYes.setSelected(false);

rdbtnNo.setSelected(true);

}

}

} catch (Exception e2 ) {

e2.printStackTrace();

}

}

}

**Add Money Client:**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.awt.event.ActionEvent;

public class AddMoneyClient {

private JFrame frame;

private JTextField textField;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

AddMoneyClient window = new AddMoneyClient();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public AddMoneyClient() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private String emailID;

public void setEmailID(String emailID) {

this.emailID = emailID;

}

Connection connection = null;

// Balance textField

JLabel lblNewLabel\_3;

// main

private void initialize() {

System.out.println(emailID);

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

} catch (SQLException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 446, 321);

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

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("ADD MONEY ");

lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 16));

lblNewLabel.setBounds(174, 11, 115, 14);

frame.getContentPane().add(lblNewLabel);

JLabel lblMoneyToBe = new JLabel("Money to be added to the account:");

lblMoneyToBe.setBounds(141, 36, 178, 14);

frame.getContentPane().add(lblMoneyToBe);

textField = new JTextField();

textField.setBounds(130, 125, 218, 20);

frame.getContentPane().add(textField);

textField.setColumns(10);

JLabel lblNewLabel\_1 = new JLabel("Acc Number");

lblNewLabel\_1.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblNewLabel\_1.setBounds(174, 61, 100, 20);

frame.getContentPane().add(lblNewLabel\_1);

// Account Number Text Field

String query3 = "select accountNo from clienttable where email = ?";

try {

PreparedStatement pst1 = connection.prepareStatement(query3);

pst1.setString(1, GlobalVariable.getInstance().getLogin());

ResultSet rs1 = pst1.executeQuery();

while (rs1.next()) {

lblNewLabel\_1.setText(rs1.getString("accountNo"));

}

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

JLabel lblNewLabel\_2 = new JLabel("\u20AC");

lblNewLabel\_2.setBounds(107, 128, 13, 14);

frame.getContentPane().add(lblNewLabel\_2);

// Get the balance

JButton btnNewButton = new JButton("ADD");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

emailID = GlobalVariable.getInstance().getLogin();

if (!textField.getText().isEmpty()) {

String query1 = "select balance from balancetable where email = ?";

String query2 = "update balancetable set balance = ? where email = ?";

try {

PreparedStatement pst = connection.prepareStatement(query1);

pst.setString(1, GlobalVariable.getInstance().getLogin());

ResultSet rs = pst.executeQuery();

float money = 0;

while (rs.next()) {

money = rs.getFloat("balance");

}

pst = connection.prepareStatement(query2);

money += Float.parseFloat(textField.getText());

pst.setFloat(1, money);

pst.setString(2, GlobalVariable.getInstance().getLogin());

pst.execute();

String MoneyString = String.valueOf(money);

lblNewLabel\_3.setText(MoneyString);

} catch (Exception e1) {

e1.getStackTrace();

}

JOptionPane.showMessageDialog(frame, "Amount added Successfully to your account");

} else {

JOptionPane.showMessageDialog(frame, "Please enter the money!!!");

}

}

});

btnNewButton.setBounds(171, 156, 89, 23);

frame.getContentPane().add(btnNewButton);

JLabel lblYourTotalBalance = new JLabel("Your Total Balance is : \u20AC");

lblYourTotalBalance.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblYourTotalBalance.setBounds(49, 207, 188, 23);

frame.getContentPane().add(lblYourTotalBalance);

lblNewLabel\_3 = new JLabel("Acc Balance");

lblNewLabel\_3.setFont(new Font("Tahoma", Font.PLAIN, 16));

lblNewLabel\_3.setBounds(247, 209, 127, 28);

frame.getContentPane().add(lblNewLabel\_3);

// BAlance Text Field

String query4 = "select balance from balancetable where email = ?";

try {

PreparedStatement pst2 = connection.prepareStatement(query4);

pst2.setString(1, GlobalVariable.getInstance().getLogin());

ResultSet rs2 = pst2.executeQuery();

while (rs2.next()) {

lblNewLabel\_3.setText(rs2.getString("balance"));

}

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

// BACK to the main window

JButton btnNewButton\_1 = new JButton("BACK");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frame.dispose();

ClientInformation ci = new ClientInformation();

ci.frmClientInformation.setVisible(true);

}

});

btnNewButton\_1.setBounds(10, 248, 89, 23);

frame.getContentPane().add(btnNewButton\_1);

}

}

**Change Password:**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JTextField;

import javax.swing.WindowConstants;

import org.apache.commons.codec.digest.DigestUtils;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class ChangePassword {

JFrame frame;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_2;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ChangePassword window = new ChangePassword();

window.frame.setVisible(true);

window.frame.setTitle("Password Change");

window.frame.setDefaultCloseOperation(WindowConstants.DO\_NOTHING\_ON\_CLOSE);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public ChangePassword() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

Connection connection = null;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 479, 251);

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

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("CHANGE PASSWORD");

lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblNewLabel.setBounds(158, 11, 209, 34);

frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("Old Password");

lblNewLabel\_1.setBounds(75, 68, 108, 19);

frame.getContentPane().add(lblNewLabel\_1);

//Password old encrypted

textField = new JTextField();

textField.setBounds(212, 67, 155, 20);

frame.getContentPane().add(textField);

textField.setColumns(10);

String encryptedPassword = DigestUtils.md5Hex(textField.getText());

JLabel lblNewLabel\_2 = new JLabel("New Password");

lblNewLabel\_2.setBounds(75, 98, 108, 14);

frame.getContentPane().add(lblNewLabel\_2);

// Password encrypted New

textField\_1 = new JTextField();

textField\_1.setBounds(212, 95, 155, 20);

frame.getContentPane().add(textField\_1);

textField\_1.setColumns(10);

String encryptedPasswordNew = DigestUtils.md5Hex(textField\_1.getText());

JLabel lblNewLabel\_3 = new JLabel("Confirm Password");

lblNewLabel\_3.setBounds(75, 123, 127, 14);

frame.getContentPane().add(lblNewLabel\_3);

textField\_2 = new JTextField();

textField\_2.setBounds(212, 120, 155, 20);

frame.getContentPane().add(textField\_2);

textField\_2.setColumns(10);

// Change password

JButton btnNewButton = new JButton("CHANGE");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

String query1 = "select password from accounts where email = ?";

String query = "update accounts set password = ? where email = ?";

try {

PreparedStatement pst = connection.prepareStatement(query1);

pst.setString(1, GlobalVariable.getInstance().getLogin());

ResultSet rs = pst.executeQuery();

System.out.println("inside change password : " + GlobalVariable.getInstance().getLogin());

while (rs.next()) {

GlobalVariable.getInstance().setPassword(rs.getString("password"));

}

System.out

.println("inside change password password : " + GlobalVariable.getInstance().getPassword());

if (GlobalVariable.getInstance().getPassword().equals(encryptedPassword)) {

pst = connection.prepareStatement(query);

pst.setString(1, encryptedPasswordNew);

pst.setString(2, GlobalVariable.getInstance().getLogin());

pst.executeUpdate();

JOptionPane.showMessageDialog(frame, "Password Changed Sucessfully");

frame.setVisible(false);

} else {

JOptionPane.showMessageDialog(frame, "Please enter correct password");

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

});

btnNewButton.setBounds(189, 166, 127, 23);

frame.getContentPane().add(btnNewButton);

}

}

**Client Information Advisor:**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JRadioButton;

public class ClientInformationAdvisor {

private JFrame frame;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_2;

private JTextField textField\_3;

private JTextField textField\_4;

private JTextField textField\_5;

private JTextField textField\_6;

private JTextField textField\_7;

private JTextField textField\_8;

private JTextField textField\_9;

private JTextField textField\_10;

private JTextField textField\_11;

private JTextField textField\_12;

private JTextField textField\_13;

private JTextField textField\_14;

private JTextField textField\_15;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ClientInformationAdvisor window = new ClientInformationAdvisor();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public ClientInformationAdvisor() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

frame = new JFrame();

frame.setBounds(100, 100, 596, 453);

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

frame.getContentPane().setLayout(null);

JLabel label = new JLabel("First Name");

label.setBounds(20, 51, 65, 14);

frame.getContentPane().add(label);

textField = new JTextField();

textField.setColumns(10);

textField.setBounds(130, 48, 117, 20);

frame.getContentPane().add(textField);

JLabel label\_1 = new JLabel("Last Name");

label\_1.setBounds(20, 87, 65, 14);

frame.getContentPane().add(label\_1);

textField\_1 = new JTextField();

textField\_1.setColumns(10);

textField\_1.setBounds(130, 87, 117, 20);

frame.getContentPane().add(textField\_1);

JLabel label\_2 = new JLabel("DOB");

label\_2.setBounds(20, 122, 46, 14);

frame.getContentPane().add(label\_2);

textField\_2 = new JTextField();

textField\_2.setColumns(10);

textField\_2.setBounds(130, 122, 117, 20);

frame.getContentPane().add(textField\_2);

JLabel label\_3 = new JLabel("Account Type");

label\_3.setBounds(20, 159, 89, 14);

frame.getContentPane().add(label\_3);

textField\_3 = new JTextField();

textField\_3.setColumns(10);

textField\_3.setBounds(130, 159, 117, 20);

frame.getContentPane().add(textField\_3);

JLabel label\_4 = new JLabel("Gender");

label\_4.setBounds(20, 197, 46, 14);

frame.getContentPane().add(label\_4);

JRadioButton radioButton = new JRadioButton("Male");

radioButton.setBounds(107, 193, 58, 23);

frame.getContentPane().add(radioButton);

JRadioButton radioButton\_1 = new JRadioButton("Female");

radioButton\_1.setBounds(167, 193, 59, 23);

frame.getContentPane().add(radioButton\_1);

textField\_4 = new JTextField();

textField\_4.setColumns(10);

textField\_4.setBounds(130, 229, 117, 20);

frame.getContentPane().add(textField\_4);

JLabel label\_5 = new JLabel("Telephone");

label\_5.setBounds(20, 232, 65, 14);

frame.getContentPane().add(label\_5);

JLabel label\_6 = new JLabel("Married");

label\_6.setBounds(20, 268, 46, 14);

frame.getContentPane().add(label\_6);

JRadioButton radioButton\_2 = new JRadioButton("Yes");

radioButton\_2.setBounds(117, 264, 48, 23);

frame.getContentPane().add(radioButton\_2);

JRadioButton radioButton\_3 = new JRadioButton("No");

radioButton\_3.setBounds(165, 264, 46, 23);

frame.getContentPane().add(radioButton\_3);

textField\_5 = new JTextField();

textField\_5.setColumns(10);

textField\_5.setBounds(127, 300, 120, 20);

frame.getContentPane().add(textField\_5);

JLabel label\_7 = new JLabel("Spouce Name:");

label\_7.setBounds(20, 303, 97, 14);

frame.getContentPane().add(label\_7);

JLabel label\_8 = new JLabel("Bank Agency");

label\_8.setBounds(20, 328, 89, 14);

frame.getContentPane().add(label\_8);

textField\_6 = new JTextField();

textField\_6.setColumns(10);

textField\_6.setBounds(125, 325, 122, 20);

frame.getContentPane().add(textField\_6);

textField\_7 = new JTextField();

textField\_7.setColumns(10);

textField\_7.setBounds(407, 323, 116, 20);

frame.getContentPane().add(textField\_7);

JLabel label\_9 = new JLabel("Email");

label\_9.setBounds(351, 320, 46, 14);

frame.getContentPane().add(label\_9);

textField\_8 = new JTextField();

textField\_8.setColumns(10);

textField\_8.setBounds(407, 292, 116, 20);

frame.getContentPane().add(textField\_8);

JLabel label\_10 = new JLabel("Children");

label\_10.setBounds(331, 295, 66, 14);

frame.getContentPane().add(label\_10);

textField\_9 = new JTextField();

textField\_9.setColumns(10);

textField\_9.setBounds(407, 257, 116, 20);

frame.getContentPane().add(textField\_9);

JLabel label\_11 = new JLabel("Birth Place");

label\_11.setBounds(331, 260, 78, 14);

frame.getContentPane().add(label\_11);

textField\_10 = new JTextField();

textField\_10.setColumns(10);

textField\_10.setBounds(407, 221, 116, 20);

frame.getContentPane().add(textField\_10);

JLabel label\_12 = new JLabel("Nationality");

label\_12.setBounds(331, 224, 78, 14);

frame.getContentPane().add(label\_12);

textField\_11 = new JTextField();

textField\_11.setColumns(10);

textField\_11.setBounds(407, 186, 116, 20);

frame.getContentPane().add(textField\_11);

JLabel label\_13 = new JLabel("Mobile");

label\_13.setBounds(351, 189, 46, 14);

frame.getContentPane().add(label\_13);

textField\_12 = new JTextField();

textField\_12.setColumns(10);

textField\_12.setBounds(407, 148, 116, 20);

frame.getContentPane().add(textField\_12);

JLabel label\_14 = new JLabel("Occupation");

label\_14.setBounds(331, 151, 78, 14);

frame.getContentPane().add(label\_14);

textField\_13 = new JTextField();

textField\_13.setColumns(10);

textField\_13.setBounds(407, 114, 138, 20);

frame.getContentPane().add(textField\_13);

textField\_14 = new JTextField();

textField\_14.setColumns(10);

textField\_14.setBounds(407, 79, 138, 20);

frame.getContentPane().add(textField\_14);

textField\_15 = new JTextField();

textField\_15.setColumns(10);

textField\_15.setBounds(407, 48, 138, 20);

frame.getContentPane().add(textField\_15);

JLabel label\_15 = new JLabel("Address");

label\_15.setBounds(331, 51, 66, 14);

frame.getContentPane().add(label\_15);

JLabel label\_16 = new JLabel("CLIENT INFORMATION EDIT");

label\_16.setBounds(219, 11, 156, 14);

frame.getContentPane().add(label\_16);

}

}

**Database Connector:**

package bankingsystemswing;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class DBConnector {

public static Connection getConnection() throws SQLException, ClassNotFoundException {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/banksystem", "root", "");

return connection;

}

}

**Generate Report:**

package bankingsystemswing;

import java.io.FileOutputStream;

import java.io.OutputStream;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.sl.usermodel.Sheet;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Workbook;

public class GenerateReport {

GenerateReport()

{

Workbook wb = new HSSFWorkbook();

try(OutputStream fileOut = new FileOutputStream("Generated Report.xls")) {

Sheet sheet1 = (Sheet) wb.createSheet("Main Sheet");

Row row = ((org.apache.poi.ss.usermodel.Sheet) sheet1).createRow(2);

Cell cell = row.createCell(5);

cell.setCellValue("Satyenda Chandan is there!!! !!!!! !!!");

wb.write(fileOut);

}catch(Exception e) {

System.out.println(e.getMessage());

}

}

}

**Global Variable:**

package bankingsystemswing;

import java.sql.ResultSet;

public class GlobalVariable {

private static GlobalVariable instance = null;

private String login;

private String iban1;

private String accountNumber;

private String bic;

private Float balance;

private String filename;

private String password;

ResultSet rs2 ;

//Email

public String getLogin() {

return login;

}

public void setLogin(String login) {

this.login = login;

}

//IBAN

public String getIban1() {

return iban1;

}

public void setIban1(String iban1) {

this.iban1 = iban1;

}

//Account Number

public String getAccountNumber() {

return accountNumber;

}

public void setAccountNumber(String accountNumber) {

this.accountNumber = accountNumber;

}

//BIC Code

public String getBicNumber() {

return bic;

}

public void setBicNumber(String bic) {

this.bic = bic;

}

//Balance

public Float getBalance() {

return balance;

}

public void setBalance(Float balance) {

this.balance = balance;

}

//File name

public String getFileName() {

return filename;

}

public void setFileName(String filename) {

this.filename = filename;

}

//ResultSet rs2

public void setResultSet(ResultSet prepare)

{

this.rs2 = prepare;

}

public ResultSet getResultSet()

{

return rs2;

}

//Password

public String getPassword() {

return password;

}

public void setPassword(String Password) {

this.password = Password;

}

public static synchronized GlobalVariable getInstance() {

if (instance == null)

instance = new GlobalVariable();

return instance;

}

}

**Global Variable External Recipient:**

package bankingsystemswing;

public class GlobalVariableExternalRecipient {

private static GlobalVariableExternalRecipient instance = null;

private String login;

private String iban1;

private String accountNumber;

private String bic;

private Float balance;

//Email

public String getLogin() {

return login;

}

public void setLogin(String login) {

this.login = login;

}

//IBAN

public String getIban1() {

return iban1;

}

public void setIban1(String iban1) {

this.iban1 = iban1;

}

//Account Number

public String getAccountNumber() {

return accountNumber;

}

public void setAccountNumber(String accountNumber) {

this.accountNumber = accountNumber;

}

//BIC Code

public String getBicNumber() {

return bic;

}

public void setBicNumber(String bic) {

this.bic = bic;

}

//Balance

public Float getBalance() {

return balance;

}

public void setBalance(Float balance) {

this.balance = balance;

}

public static synchronized GlobalVariableExternalRecipient getInstance() {

if (instance == null)

instance = new GlobalVariableExternalRecipient();

return instance;

}

}

**Global Variable Office Advisor Email:**

package bankingsystemswing;

public class GlobalVariableOfficeAdvisorEmail {

private static GlobalVariableOfficeAdvisorEmail instance = null;

private String email;

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public static synchronized GlobalVariableOfficeAdvisorEmail getInstance() {

if (instance == null)

instance = new GlobalVariableOfficeAdvisorEmail();

return instance;

}

}

**Manager:**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Time;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.regex.Pattern;

import javax.swing.JButton;

import javax.swing.JFileChooser;

import javax.swing.JTable;

import org.apache.poi.hssf.usermodel.HSSFRow;

import org.apache.poi.hssf.usermodel.HSSFSheet;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import javax.swing.JScrollPane;

import com.toedter.calendar.JDateChooser;

import net.proteanit.sql.DbUtils;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.awt.event.ActionEvent;

public class Manager1 extends Thread {

JFrame frame;

private JTable table;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Manager1 window = new Manager1();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public Manager1() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

Connection connection = null;

JDateChooser dateChooser;

JDateChooser dateChooser\_1;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 699, 506);

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

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("MANAGER");

lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 16));

lblNewLabel.setBounds(282, 22, 134, 31);

frame.getContentPane().add(lblNewLabel);

// Get the data for a defined period

JButton btnNewButton = new JButton("SHOW");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

String date1 = dateChooser.getDate().toString();

String date[] = date1.split(" ");

if (date[1].equalsIgnoreCase("jan")) {

date[1] = "01";

} else if (date[1].equalsIgnoreCase("feb")) {

date[1] = "02";

} else if (date[1].equalsIgnoreCase("mar")) {

date[1] = "03";

} else if (date[1].equalsIgnoreCase("apr")) {

date[1] = "04";

} else if (date[1].equalsIgnoreCase("may")) {

date[1] = "05";

} else if (date[1].equalsIgnoreCase("jun")) {

date[1] = "06";

} else if (date[1].equalsIgnoreCase("jul")) {

date[1] = "07";

} else if (date[1].equalsIgnoreCase("aug")) {

date[1] = "08";

} else if (date[1].equalsIgnoreCase("sep")) {

date[1] = "09";

} else if (date[1].equalsIgnoreCase("oct")) {

date[1] = "10";

} else if (date[1].equalsIgnoreCase("nov")) {

date[1] = "11";

} else if (date[1].equalsIgnoreCase("dec")) {

date[1] = "12";

}

String date2 = date[5] + "-" + date[1] + "-" + date[2];

String date3 = dateChooser\_1.getDate().toString();

String date4[] = date3.split(" ");

if (date4[1].equalsIgnoreCase("jan")) {

date4[1] = "01";

} else if (date4[1].equalsIgnoreCase("feb")) {

date4[1] = "02";

} else if (date4[1].equalsIgnoreCase("mar")) {

date4[1] = "03";

} else if (date4[1].equalsIgnoreCase("apr")) {

date4[1] = "04";

} else if (date4[1].equalsIgnoreCase("may")) {

date4[1] = "05";

} else if (date4[1].equalsIgnoreCase("jun")) {

date4[1] = "06";

} else if (date4[1].equalsIgnoreCase("jul")) {

date4[1] = "07";

} else if (date4[1].equalsIgnoreCase("aug")) {

date4[1] = "08";

} else if (date4[1].equalsIgnoreCase("sep")) {

date4[1] = "09";

} else if (date4[1].equalsIgnoreCase("oct")) {

date4[1] = "10";

} else if (date4[1].equalsIgnoreCase("nov")) {

date4[1] = "11";

} else if (date4[1].equalsIgnoreCase("dec")) {

date4[1] = "12";

}

String date5 = date4[5] + "-" + date4[1] + "-" + date4[2];

String query = "select clienttable.firstName,clienttable.lastName,clienttable.email, transactiontable.debit as deposit from clienttable join transactiontable on\r\n"

+ "clienttable.email = transactiontable.email where transactiontable.dateOfTransaction between ? and ?";

try {

PreparedStatement pst = connection.prepareStatement(query);

pst.setString(1, date2);

pst.setString(2, date5);

ResultSet rs = pst.executeQuery();

table.setModel(DbUtils.resultSetToTableModel(rs));

GlobalVariable.getInstance().setResultSet(pst.executeQuery());

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

});

btnNewButton.setBounds(271, 102, 89, 23);

frame.getContentPane().add(btnNewButton);

JScrollPane scrollPane = new JScrollPane();

scrollPane.setBounds(10, 146, 663, 280);

frame.getContentPane().add(scrollPane);

table = new JTable();

scrollPane.setViewportView(table);

// Generate Report using Apache POI

JButton btnNewButton\_1 = new JButton("GENERATE REPORT");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

// TO DO

JFileChooser chooser = new JFileChooser();

chooser.showOpenDialog(null);

File f = chooser.getSelectedFile();

GlobalVariable.getInstance().setFileName(f.getAbsolutePath());

System.out.println(GlobalVariable.getInstance().getFileName());

JOptionPane.showMessageDialog(frame, "File Saved Sucessfully");

HSSFWorkbook workbook = new HSSFWorkbook();

HSSFSheet sheet = workbook.createSheet("lawix10");

HSSFRow rowhead = sheet.createRow(0);

rowhead.createCell(0).setCellValue("First Name");

rowhead.createCell(1).setCellValue("Last Name");

rowhead.createCell(2).setCellValue("Email");

rowhead.createCell(3).setCellValue("Deposit");

ResultSet rs3 = GlobalVariable.getInstance().getResultSet();

int i = 1;

try {

while (rs3.next()) {

HSSFRow row = sheet.createRow((short) i);

row.createCell((short) 0).setCellValue(rs3.getString("clienttable.firstName"));

row.createCell((short) 1).setCellValue(rs3.getString("clienttable.lastName"));

row.createCell((short) 2).setCellValue(rs3.getString("clienttable.email"));

row.createCell((short) 3).setCellValue(rs3.getString("deposit"));

i++;

}

String yemi = GlobalVariable.getInstance().getFileName();

System.out.println("yemi " + yemi);

FileOutputStream fileOut = new FileOutputStream(yemi);

workbook.write(fileOut);

fileOut.close();

} catch (SQLException | IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

});

btnNewButton\_1.setBounds(487, 433, 134, 23);

frame.getContentPane().add(btnNewButton\_1);

dateChooser = new JDateChooser();

dateChooser.setBounds(77, 65, 134, 20);

frame.getContentPane().add(dateChooser);

JLabel lblNewLabel\_1 = new JLabel("TO");

lblNewLabel\_1.setBounds(292, 64, 46, 14);

frame.getContentPane().add(lblNewLabel\_1);

dateChooser\_1 = new JDateChooser();

dateChooser\_1.setBounds(438, 65, 134, 20);

frame.getContentPane().add(dateChooser\_1);

}

}

**Office Advisor Client Information:**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import java.awt.Font;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class OfficeAdvisorClientInformation {

JFrame frame;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

OfficeAdvisorClientInformation window = new OfficeAdvisorClientInformation();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public OfficeAdvisorClientInformation() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

Connection connection = null;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 664, 561);

frame.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JLabel lblClientInformation = new JLabel("CLIENT INFORMATION");

lblClientInformation.setFont(new Font("Tahoma", Font.PLAIN, 16));

lblClientInformation.setBounds(274, 11, 187, 31);

frame.getContentPane().add(lblClientInformation);

JLabel lblFirstName = new JLabel("First Name");

lblFirstName.setBounds(24, 71, 82, 14);

frame.getContentPane().add(lblFirstName);

JLabel lblLastName = new JLabel("Last Name");

lblLastName.setBounds(24, 96, 82, 14);

frame.getContentPane().add(lblLastName);

JLabel lblDob = new JLabel("DOB");

lblDob.setBounds(23, 121, 46, 14);

frame.getContentPane().add(lblDob);

JLabel label = new JLabel("");

label.setBounds(24, 146, 62, 14);

frame.getContentPane().add(label);

JLabel lblGender = new JLabel("Gender");

lblGender.setBounds(24, 146, 46, 14);

frame.getContentPane().add(lblGender);

JLabel lblTelephone = new JLabel("Telephone");

lblTelephone.setBounds(24, 184, 82, 14);

frame.getContentPane().add(lblTelephone);

JLabel lblMarried = new JLabel("Married");

lblMarried.setBounds(24, 209, 62, 14);

frame.getContentPane().add(lblMarried);

JLabel lblSpouceName = new JLabel("Spouce Name");

lblSpouceName.setBounds(24, 234, 94, 14);

frame.getContentPane().add(lblSpouceName);

JLabel lblBankAgency = new JLabel("Bank Agency");

lblBankAgency.setBounds(23, 259, 83, 14);

frame.getContentPane().add(lblBankAgency);

JLabel lblAddress = new JLabel("Address");

lblAddress.setBounds(24, 284, 69, 14);

frame.getContentPane().add(lblAddress);

JLabel lblhouse = new JLabel("New label");

lblhouse.setBounds(124, 284, 202, 14);

frame.getContentPane().add(lblhouse);

JLabel lblneighbourhood = new JLabel("New label");

lblneighbourhood.setBounds(124, 309, 202, 14);

frame.getContentPane().add(lblneighbourhood);

JLabel lblcity = new JLabel("New label");

lblcity.setBounds(124, 334, 202, 14);

frame.getContentPane().add(lblcity);

JLabel lblcountry = new JLabel("New label");

lblcountry.setBounds(124, 359, 202, 14);

frame.getContentPane().add(lblcountry);

JLabel lblzip = new JLabel("New label");

lblzip.setBounds(124, 384, 202, 14);

frame.getContentPane().add(lblzip);

JLabel lblBankagency = new JLabel("New label");

lblBankagency.setBounds(124, 259, 202, 14);

frame.getContentPane().add(lblBankagency);

JLabel lblSpouce = new JLabel("New label");

lblSpouce.setBounds(124, 234, 202, 14);

frame.getContentPane().add(lblSpouce);

JLabel lblNewLabel = new JLabel("New label");

lblNewLabel.setBounds(124, 209, 202, 14);

frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("New label");

lblNewLabel\_1.setBounds(124, 184, 202, 14);

frame.getContentPane().add(lblNewLabel\_1);

JLabel lblNewLabel\_2 = new JLabel("New label");

lblNewLabel\_2.setBounds(124, 146, 202, 14);

frame.getContentPane().add(lblNewLabel\_2);

JLabel lblNewLabel\_3 = new JLabel("New label");

lblNewLabel\_3.setBounds(124, 121, 202, 14);

frame.getContentPane().add(lblNewLabel\_3);

JLabel lblNewLabel\_4 = new JLabel("New label");

lblNewLabel\_4.setBounds(124, 96, 202, 14);

frame.getContentPane().add(lblNewLabel\_4);

JLabel lblNewLabel\_5 = new JLabel("New label");

lblNewLabel\_5.setBounds(124, 71, 202, 14);

frame.getContentPane().add(lblNewLabel\_5);

JLabel lblOccupation = new JLabel("Occupation");

lblOccupation.setBounds(370, 71, 82, 14);

frame.getContentPane().add(lblOccupation);

JLabel lblNationality = new JLabel("Nationality");

lblNationality.setBounds(370, 96, 82, 14);

frame.getContentPane().add(lblNationality);

JLabel lblMobile = new JLabel("Mobile");

lblMobile.setBounds(370, 121, 62, 14);

frame.getContentPane().add(lblMobile);

JLabel lblBirthPlace = new JLabel("Birth Place");

lblBirthPlace.setBounds(370, 146, 82, 14);

frame.getContentPane().add(lblBirthPlace);

JLabel lblChildren = new JLabel("Children");

lblChildren.setBounds(370, 171, 72, 14);

frame.getContentPane().add(lblChildren);

//Back button dispose frame

JButton btnBack = new JButton("Back");

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

frame.dispose();

}

});

btnBack.setBounds(10, 422, 89, 23);

frame.getContentPane().add(btnBack);

//Show Transaction of the client when clicked

JButton btnShowTransaction = new JButton("Show Transaction ");

btnShowTransaction.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

OfficeAdvisorTransaction oat = new OfficeAdvisorTransaction();

oat.frame.setVisible(true);

}

});

btnShowTransaction.setBounds(233, 422, 121, 23);

frame.getContentPane().add(btnShowTransaction);

JLabel lblBalance = new JLabel("BALANCE");

lblBalance.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblBalance.setBounds(387, 259, 94, 23);

frame.getContentPane().add(lblBalance);

JLabel lblNewLabel\_6 = new JLabel("New label");

lblNewLabel\_6.setBounds(378, 309, 74, 14);

frame.getContentPane().add(lblNewLabel\_6);

JLabel lblNewLabel\_7 = new JLabel("\u20AC");

lblNewLabel\_7.setBounds(457, 309, 46, 14);

frame.getContentPane().add(lblNewLabel\_7);

JLabel lbloccup = new JLabel("New label");

lbloccup.setBounds(478, 71, 160, 14);

frame.getContentPane().add(lbloccup);

JLabel lblnation = new JLabel("New label");

lblnation.setBounds(478, 96, 160, 14);

frame.getContentPane().add(lblnation);

JLabel lblmobile = new JLabel("New label");

lblmobile.setBounds(478, 121, 160, 14);

frame.getContentPane().add(lblmobile);

JLabel lblbirthplc = new JLabel("New label");

lblbirthplc.setBounds(478, 146, 160, 14);

frame.getContentPane().add(lblbirthplc);

JLabel lblchild = new JLabel("New label");

lblchild.setBounds(478, 171, 160, 14);

frame.getContentPane().add(lblchild);

String query = "Select firstName,LastName,dateOfBirth,telephone,house,neighbourhood,city,country,zipcode,occupation,mobile,nationality,Married,spouceName,birthPlace,children,bankAgency from clienttable where email=?";

try {

PreparedStatement pst = connection.prepareStatement(query);

officeAdvisor oa = new officeAdvisor();

pst.setString(1,GlobalVariableOfficeAdvisorEmail.getInstance().getEmail() );

ResultSet rs = pst.executeQuery();

while(rs.next())

{

lblNewLabel\_5.setText(rs.getString("firstName"));

lblNewLabel\_4.setText(rs.getString("lastName"));

lblNewLabel\_3.setText(rs.getString("dateOfBirth"));

lblNewLabel\_1.setText(rs.getString("telephone"));

lblhouse.setText(rs.getString("house"));

lblneighbourhood.setText(rs.getString("neighbourhood"));

lblcity.setText(rs.getString("city"));

lblcountry.setText(rs.getString("country"));

lblzip.setText(rs.getString("zipcode"));

lblNewLabel.setText(rs.getString("Married"));

lblSpouce.setText(rs.getString("spouceName"));

lblBankagency.setText(rs.getString("bankAgency"));

lbloccup.setText(rs.getString("occupation"));

lblnation.setText(rs.getString("nationality"));

lblmobile.setText(rs.getString("mobile"));

lblbirthplc.setText(rs.getString("birthPlace"));

lblchild.setText(rs.getString("children"));

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

//For Balance

String query2 = "select balance from balancetable where email=?";

try {

PreparedStatement pst1 = connection.prepareStatement(query2);

pst1.setString(1, GlobalVariableOfficeAdvisorEmail.getInstance().getEmail());

ResultSet rs2 = pst1.executeQuery();

while(rs2.next()) {

lblNewLabel\_6.setText(rs2.getString("balance"));

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

**Office Advisor Transaction:**

package bankingsystemswing;

import java.awt.EventQueue;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import net.proteanit.sql.DbUtils;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import javax.swing.ListSelectionModel;

public class OfficeAdvisorTransaction {

JFrame frame;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

OfficeAdvisorTransaction window = new OfficeAdvisorTransaction();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public OfficeAdvisorTransaction() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

Connection connection = null;

private JTable table;

private JScrollPane scrollPane;

private JButton btnBack;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e1) {

e1.printStackTrace();

} catch (SQLException e1) {

e1.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 1061, 601);

frame.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

scrollPane = new JScrollPane();

scrollPane.setBounds(10, 11, 1025, 506);

frame.getContentPane().add(scrollPane);

table = new JTable();

table.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

table.setColumnSelectionAllowed(true);

table.setCellSelectionEnabled(false);

table.setEnabled(true);

table.isEnabled();

scrollPane.setViewportView(table);

try {

String query = "Select \* from transactiontable where email=?";

PreparedStatement pst = connection.prepareStatement(query);

pst.setString(1, GlobalVariableOfficeAdvisorEmail.getInstance().getEmail());

ResultSet rs = pst.executeQuery();

table.setModel(DbUtils.resultSetToTableModel(rs));

btnBack = new JButton("Back");

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frame.dispose();

}

});

btnBack.setBounds(0, 528, 89, 23);

frame.getContentPane().add(btnBack);

} catch (Exception e) {

e.printStackTrace();

} }}

**Transaction:**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JTable;

import com.opencsv.CSVWriter;

import net.proteanit.sql.DbUtils;

import java.awt.Color;

import javax.swing.JScrollPane;

import java.sql.\*;

import javax.swing.\*;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.awt.event.ActionEvent;

public class Transaction {

private JFrame frame;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Transaction window = new Transaction();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public Transaction() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

Connection connection = null;

private JTable table;

private JScrollPane scrollPane;

private JButton btnExport;

private JButton btnBack;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e1) {

e1.printStackTrace();

} catch (SQLException e1) {

e1.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 1061, 601);

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

frame.getContentPane().setLayout(null);

scrollPane = new JScrollPane();

scrollPane.setBounds(10, 11, 1025, 506);

frame.getContentPane().add(scrollPane);

table = new JTable();

table.setEnabled(false);

scrollPane.setViewportView(table);

try {

String query = "Select \* from transactiontable where email = ?";

PreparedStatement pst = connection.prepareStatement(query);

pst.setString(1, GlobalVariable.getInstance().getLogin());

ResultSet rs = pst.executeQuery();

table.setModel(DbUtils.resultSetToTableModel(rs));

//Export transaction to csv

btnExport = new JButton("Export");

btnExport.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

JFileChooser chooser = new JFileChooser();

chooser.showOpenDialog(null);

File f = chooser.getSelectedFile();

String filename = f.getAbsolutePath();

try {

//FileWriter fw = new FileWriter(filename);

ResultSet rs2 = pst.executeQuery();

CSVWriter writer = new CSVWriter(new FileWriter(filename));

writer.writeAll(rs2, true,true);

writer.close();

/\*fw.append("id\_bankBalance");

fw.append(",");

fw.append("transaction");

fw.append(",");

fw.append("debit");

fw.append(",");

while(rs2.next())

{

fw.append(rs2.getString(1));

fw.append(",");

fw.append(rs2.getString(2));

fw.append(",");

fw.append(rs2.getString(3));

fw.append(",");

fw.append(rs2.getString(4));

fw.append(",");

fw.append(rs2.getString(5));

fw.append("\n");

}

fw.flush();

fw.close();\*/

JOptionPane.showMessageDialog(frame, "CSV File is created successfully");

} catch (IOException | SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

});

btnExport.setBounds(533, 528, 89, 23);

frame.getContentPane().add(btnExport);

btnBack = new JButton("Back");

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frame.dispose();

ClientInformation clientinformation = new ClientInformation();

clientinformation.main(null);

}

});

btnBack.setBounds(0, 528, 89, 23);

frame.getContentPane().add(btnBack);

}

catch(Exception e){

e.printStackTrace();

}}}

**Transfer Money:**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import org.apache.poi.util.SystemOutLogger;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.awt.event.ActionEvent;

import java.awt.Font;

public class TransferMoney {

JFrame frame;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_4;

private JTextField textField\_5;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

TransferMoney window = new TransferMoney();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public TransferMoney() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

Connection connection = null;

private JTextField textField\_9;

private JTextField textField\_recipientEmailID;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e2) {

e2.printStackTrace();

} catch (SQLException e2) {

e2.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 576, 438);

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

frame.getContentPane().setLayout(null);

JLabel lblTransferMoney = new JLabel("Transfer Money");

lblTransferMoney.setBounds(228, 21, 107, 30);

frame.getContentPane().add(lblTransferMoney);

textField = new JTextField();

textField.setBounds(114, 113, 107, 20);

frame.getContentPane().add(textField);

textField.setColumns(10);

textField\_1 = new JTextField();

textField\_1.setBounds(350, 113, 86, 20);

frame.getContentPane().add(textField\_1);

textField\_1.setColumns(10);

JLabel lblNewLabel = new JLabel("First Name");

lblNewLabel.setBounds(30, 116, 77, 14);

frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("Last Name:");

lblNewLabel\_1.setBounds(265, 116, 86, 14);

frame.getContentPane().add(lblNewLabel\_1);

JButton btnTransfer = new JButton("Transfer");

btnTransfer.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//Warning

if(textField\_4.getText().isEmpty() || textField.getText().isEmpty() || textField\_1.getText().isEmpty() || textField\_9.getText().isEmpty() || textField\_5.getText().isEmpty())

{

JOptionPane.showMessageDialog(frame, "Please Enter BIC Code & Name & IBAN & Amount ");

}

// Transfer Money

String query2 = "select iban,accountNo,BIC from clienttable where email =? ";

String query6 = "select balance from balancetable where email = ?";

try {

PreparedStatement pst4 = connection.prepareStatement(query2);

pst4.setString(1, GlobalVariable.getInstance().getLogin());

ResultSet rs4 = pst4.executeQuery();

while (rs4.next()) {

GlobalVariable.getInstance().setIban1(rs4.getString("iban"));

GlobalVariable.getInstance().setAccountNumber(rs4.getString("accountNo"));

GlobalVariable.getInstance().setBicNumber(rs4.getString("BIC"));

}

// balance Prepared Statement

pst4 = connection.prepareStatement(query6);

pst4.setString(1, GlobalVariable.getInstance().getLogin());

rs4 = pst4.executeQuery();

while (rs4.next()) {

GlobalVariable.getInstance().setBalance(rs4.getFloat("balance"));

}

} catch (SQLException e2) {

e2.printStackTrace();

}

// If IBAN equal transfer money

if (textField\_5.getText().toString().equals(GlobalVariable.getInstance().getIban1().toString())) {

Date date = new Date();

DateFormat dateformat = new SimpleDateFormat("YYYY-MM-DD");

String dateFormated = dateformat.format(date);

String transactionName = textField.getText() + " " + textField\_1.getText();

String credit = textField\_9.getText();

// Transaction Table insert

String query3 = "insert into transactiontable(transactionName,credit,dateOfTransaction,email,iban,accountNoRecipient,bicRecipient,bic,accountNo,emailIDRecipient)values(?,?,?,?,?,?,?,?,?,?)";

try {

PreparedStatement pst1 = connection.prepareStatement(query3);

pst1.setString(1, transactionName);

pst1.setString(2, credit);

pst1.setString(3, dateFormated);

pst1.setString(4, GlobalVariable.getInstance().getLogin());

pst1.setString(5, GlobalVariable.getInstance().getIban1());

pst1.setString(6, GlobalVariable.getInstance().getAccountNumber());

pst1.setString(7, GlobalVariable.getInstance().getBicNumber());

pst1.setString(8, textField\_4.getText());

pst1.setString(9, GlobalVariable.getInstance().getAccountNumber());

pst1.setString(10, GlobalVariable.getInstance().getLogin());

pst1.execute();

} catch (Exception e1) {

e1.printStackTrace();

}

}

// Transfer Money if IBAN not equal

if (!(textField\_5.getText().toString().equals(GlobalVariable.getInstance().getIban1().toString()))) {

Date date = new Date();

DateFormat dateformat = new SimpleDateFormat("YYYY-MM-DD");

String dateFormated = dateformat.format(date);

String transactionName = textField.getText() + " " + textField\_1.getText();

String debit = textField\_9.getText();

String creditExternal = textField\_9.getText();

// Balance update

Float updatedBalance = GlobalVariable.getInstance().getBalance() - Float.parseFloat(debit);

// Transaction Table insert

String query3 = "insert into transactiontable(transactionName,debit,dateOfTransaction,email,iban,accountNoRecipient,bicRecipient,bic,accountNo,emailIDRecipient)values(?,?,?,?,?,?,?,?,?,?)";

String query5 = "update balancetable set balance = ? where email = ?";

// Update the transaction detail of the external account

String query8 = "select \* from clienttable where email = ?";

String query9 = "insert into transactiontable(transactionName,credit,dateOfTransaction,email,iban,accountNoRecipient,bicRecipient,bic,accountNo,emailIDRecipient)values(?,?,?,?,?,?,?,?,?,?)";

// Update query balance of external account

String query10 = "select balance from balancetable where email = ?";

String query11 = "update balancetable set balance = ? where email =?";

try {

PreparedStatement pst1 = connection.prepareStatement(query3);

pst1.setString(1, transactionName);

pst1.setString(2, debit);

pst1.setString(3, dateFormated);

pst1.setString(4, GlobalVariable.getInstance().getLogin());

pst1.setString(5, GlobalVariable.getInstance().getIban1());

pst1.setString(6, textField\_5.getText());

pst1.setString(7, textField\_4.getText());

pst1.setString(8, GlobalVariable.getInstance().getBicNumber());

pst1.setString(9, GlobalVariable.getInstance().getAccountNumber());

pst1.setString(10, textField\_recipientEmailID.getText());

pst1.execute();

// query4

pst1 = connection.prepareStatement(query5);

pst1.setFloat(1, updatedBalance);

pst1.setString(2, GlobalVariable.getInstance().getLogin());

pst1.executeUpdate();

// query 8

pst1 = connection.prepareStatement(query8);

pst1.setString(1, textField\_recipientEmailID.getText());

ResultSet rs8 = pst1.executeQuery();

while (rs8.next()) {

GlobalVariableExternalRecipient.getInstance().setLogin(rs8.getString("email"));

GlobalVariableExternalRecipient.getInstance().setBicNumber(rs8.getString("BIC"));

GlobalVariableExternalRecipient.getInstance().setIban1(rs8.getString("iban"));

GlobalVariableExternalRecipient.getInstance().setAccountNumber(rs8.getString("accountNo"));

}

// query 9

pst1 = connection.prepareStatement(query9);

pst1.setString(1, transactionName);

pst1.setString(2, creditExternal);

pst1.setString(3, dateFormated);

pst1.setString(4, GlobalVariableExternalRecipient.getInstance().getLogin());

pst1.setString(5, GlobalVariableExternalRecipient.getInstance().getIban1());

pst1.setString(6, GlobalVariable.getInstance().getAccountNumber());

pst1.setString(7, GlobalVariable.getInstance().getBicNumber());

pst1.setString(8, textField\_4.getText());

pst1.setString(9, GlobalVariableExternalRecipient.getInstance().getAccountNumber());

pst1.setString(10, GlobalVariable.getInstance().getLogin());

pst1.execute();

// query10

pst1 = connection.prepareStatement(query10);

pst1.setString(1, GlobalVariableExternalRecipient.getInstance().getLogin());

rs8 = pst1.executeQuery();

while (rs8.next()) {

GlobalVariableExternalRecipient.getInstance().setBalance(rs8.getFloat("balance"));

}

//query 11

Float ExternalRecipientBalance = GlobalVariableExternalRecipient.getInstance().getBalance() + Float.parseFloat(textField\_9.getText());

GlobalVariableExternalRecipient.getInstance().setBalance(ExternalRecipientBalance);

pst1 = connection.prepareStatement(query11);

pst1.setFloat(1, ExternalRecipientBalance);

pst1.setString(2, GlobalVariableExternalRecipient.getInstance().getLogin());

pst1.executeUpdate();

} catch (Exception e1) {

e1.printStackTrace();

}

}

// TransferMoney table

String query = "insert into transfermoney (FirstName,LastName,bic,amount,iban) values(?,?,?,?,?)";

try {

PreparedStatement pst3 = connection.prepareStatement(query);

pst3.setString(1, textField.getText());

pst3.setString(2, textField\_1.getText());

pst3.setString(3, textField\_4.getText());

pst3.setFloat(4, Float.parseFloat(textField\_9.getText()));

pst3.setString(5, textField\_5.getText());

pst3.executeUpdate();

System.out.println("=========================================");

System.out.println(textField.getText());

System.out.println(textField\_1.getText());

System.out.println(textField\_4.getText());

System.out.println(textField\_9.getText());

System.out.println( textField\_5.getText());

} catch (SQLException e1) {

e1.printStackTrace();

}

//Transfer Successful dialog box code

Float amm = Float.parseFloat(textField\_9.getText());

if(textField\_4.getText().isEmpty() || textField.getText().isEmpty() || textField\_1.getText().isEmpty() || textField\_5.getText().isEmpty() )

{

if(amm == 0)

{

//JOptionPane.showMessageDialog(frame, "Transfer Sucessfully amount credited");

JOptionPane.showMessageDialog(frame, "Please Enter BIC Code & Name & IBAN & Amount ");

}

JOptionPane.showMessageDialog(frame, "Please Enter BIC Code & Name & IBAN & Amount ");

}else

{

JOptionPane.showMessageDialog(frame, "Transfer Sucessfully amount credited");

}

}

});

btnTransfer.setBounds(265, 330, 89, 23);

frame.getContentPane().add(btnTransfer);

JButton btnBack = new JButton("back");

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

frame.dispose();

ClientInformation clientInformation = new ClientInformation();

clientInformation.main(null);

}

});

btnBack.setBounds(0, 365, 89, 23);

frame.getContentPane().add(btnBack);

JLabel lblNewLabel\_4 = new JLabel("BIC:");

lblNewLabel\_4.setBounds(289, 178, 46, 14);

frame.getContentPane().add(lblNewLabel\_4);

textField\_4 = new JTextField();

textField\_4.setBounds(350, 175, 86, 20);

frame.getContentPane().add(textField\_4);

textField\_4.setColumns(10);

JLabel lblNewLabel\_5 = new JLabel("IBAN:");

lblNewLabel\_5.setBounds(175, 302, 46, 14);

frame.getContentPane().add(lblNewLabel\_5);

textField\_5 = new JTextField();

textField\_5.setBounds(228, 299, 206, 20);

frame.getContentPane().add(textField\_5);

textField\_5.setColumns(10);

JLabel lblAmount = new JLabel("AMOUNT");

lblAmount.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblAmount.setBounds(338, 203, 98, 30);

frame.getContentPane().add(lblAmount);

textField\_9 = new JTextField();

textField\_9.setBounds(327, 238, 109, 30);

frame.getContentPane().add(textField\_9);

textField\_9.setColumns(10);

JLabel lblNewLabel\_3 = new JLabel("\u20AC");

lblNewLabel\_3.setBounds(289, 241, 28, 14);

frame.getContentPane().add(lblNewLabel\_3);

// New Transfer resets textField

JButton btnNewButton = new JButton("NEW TRANSFER");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

textField.setText("");

textField\_1.setText("");

textField\_4.setText("");

textField\_recipientEmailID.setText("");

textField\_9.setText("");

textField\_5.setText("");

}

});

btnNewButton.setBounds(400, 365, 130, 23);

frame.getContentPane().add(btnNewButton);

JLabel lblRecipientEmailId = new JLabel("Recipient Email ID:");

lblRecipientEmailId.setBounds(30, 162, 130, 14);

frame.getContentPane().add(lblRecipientEmailId);

textField\_recipientEmailID = new JTextField();

textField\_recipientEmailID.setBounds(30, 192, 191, 20);

frame.getContentPane().add(textField\_recipientEmailID);

textField\_recipientEmailID.setColumns(10);

}

}

**Demo Combo:**

package bankingsystemswing;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JComboBox;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class democombo {

private JFrame frame;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

democombo window = new democombo();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public democombo() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

frame = new JFrame();

frame.setBounds(100, 100, 450, 300);

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

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("Account Type");

lblNewLabel.setBounds(45, 77, 77, 14);

frame.getContentPane().add(lblNewLabel);

JComboBox comboBox = new JComboBox();

comboBox.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String accountType = comboBox.getSelectedItem().toString();

System.out.println(accountType);

if (accountType == "Client") {

accountType = "C";

}

else if (accountType == "Officer") {

accountType = "A";

} else if (accountType == "Manager") {

accountType = "M";

}

System.out.println(accountType);

}

});

comboBox.setBounds(150, 74, 77, 20);

frame.getContentPane().add(comboBox);

comboBox.addItem("Client");

comboBox.addItem("Officer");

comboBox.addItem("Manager");

}

}

**Office Advisor:**

package bankingsystemswing;

import java.awt.EventQueue;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTable;

import javax.swing.table.TableModel;

import net.proteanit.sql.DbUtils;

import javax.swing.JScrollPane;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import javax.swing.JButton;

public class officeAdvisor {

JFrame frame;

private JTable table;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

officeAdvisor window = new officeAdvisor();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public officeAdvisor() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

public ClientInformationAdvisor cia = new ClientInformationAdvisor();

Connection connection = null;

private JButton btnView;

private void initialize() {

try {

connection = DBConnector.getConnection();

} catch (ClassNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

frame = new JFrame();

frame.setBounds(100, 100, 556, 447);

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

frame.getContentPane().setLayout(null);

JLabel lblOfficeAdvisor = new JLabel("OFFICE ADVISOR");

lblOfficeAdvisor.setBounds(219, 11, 112, 14);

frame.getContentPane().add(lblOfficeAdvisor);

JScrollPane scrollPane = new JScrollPane();

scrollPane.setBounds(25, 43, 494, 312);

frame.getContentPane().add(scrollPane);

table = new JTable();

/// Mouse clicked event

table.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent arg0) {

boolean a = table.isEditing();

int index = table.getSelectedRow();

GlobalVariableOfficeAdvisorEmail.getInstance().setEmail((String) table.getValueAt(index, 2));

OfficeAdvisorClientInformation oaci = new OfficeAdvisorClientInformation();

oaci.frame.setVisible(true);

}

});

scrollPane.setViewportView(table);

try {

String query = "Select firstName,LastName,email,dateOfBirth,gender from clienttable";

PreparedStatement pst = connection.prepareStatement(query);

ResultSet rs = pst.executeQuery();

table.setModel(DbUtils.resultSetToTableModel(rs));

btnView = new JButton("view");

btnView.setBounds(35, 366, 89, 23);

frame.getContentPane().add(btnView);

} catch (Exception e) {

e.printStackTrace();

}

}

}

**XML :**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>bankingsystemswing</groupId>

<artifactId>bankingsystemswing</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>BankingSystemSwing</name>

<description>Banking System using swing </description>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<!-- https://mvnrepository.com/artifact/com.opencsv/opencsv -->

<dependency>

<groupId>com.opencsv</groupId>

<artifactId>opencsv</artifactId>

<version>4.4</version>

</dependency>

<dependency>

<groupId>commons-codec</groupId>

<artifactId>commons-codec</artifactId>

<version>1.11</version>

</dependency>

</dependencies>

</project>

**SQL:**

-- --------------------------------------------------------

-- Host: 127.0.0.1

-- Server version: 10.1.34-MariaDB - mariadb.org binary distribution

-- Server OS: Win32

-- HeidiSQL Version: 9.5.0.5196

-- --------------------------------------------------------

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!50503 SET NAMES utf8mb4 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

-- Dumping database structure for banksystem

DROP DATABASE IF EXISTS `banksystem`;

CREATE DATABASE IF NOT EXISTS `banksystem` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;

USE `banksystem`;

-- Dumping structure for table banksystem.accounts

DROP TABLE IF EXISTS `accounts`;

CREATE TABLE IF NOT EXISTS `accounts` (

`id\_Account` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`email` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL,

`account\_type` char(50) NOT NULL,

`username` varchar(50) NOT NULL,

PRIMARY KEY (`id\_Account`),

UNIQUE KEY `email` (`email`)

) ENGINE=InnoDB AUTO\_INCREMENT=21 DEFAULT CHARSET=latin1;

-- Dumping data for table banksystem.accounts: ~17 rows (approximately)

DELETE FROM `accounts`;

/\*!40000 ALTER TABLE `accounts` DISABLE KEYS \*/;

INSERT INTO `accounts` (`id\_Account`, `email`, `password`, `account\_type`, `username`) VALUES

(1, 'csatyendra02@gmail.com', 'astro1', 'C', 'satya'),

(4, 'csatyendra02@gmaile.com', 'astro11111', 'C', ''),

(5, 'csatyendra02@gmaieele.com', 'astro11111', 'C', 'satya1'),

(6, '111@gmail.com', 'astro1', 'M', ''),

(7, '1@gmail.com', 'astro1', 'A', ''),

(8, 'tommyhilfigger@gmail.com', 'astro123', 'C', 'tommy'),

(9, 'jonny@gmail.com', 'astro123', 'C', 'jonny'),

(10, 'rooon@gmail.com', 'astro1', 'C', 'rooom'),

(11, 'csatyen@gmail.com', 'astro11', 'C', 'sdfsdf'),

(12, 'dd@gmail.com', '1234', 'C', 'dddd'),

(13, 'ddd@gmail.com', '1111', 'C', 'ddd'),

(14, 'sam@gmail.com', 'astro1', 'C', 'samuel'),

(15, 'dogmail@gmail.com', 'astro1', 'C', 'dogmail'),

(16, 'dom@gmail.com', 'astro1', 'C', 'dom'),

(17, 'fff@gmail.com', 'astro1', 'C', 'fff'),

(18, 'domm@gmail.com', 'astro1', 'C', 'domm'),

(19, 'gatin@gmail.com', 'astro1', 'A', 'gatin'),

(20, 'manager@gmail.com', 'astro1', 'M', 'manager');

/\*!40000 ALTER TABLE `accounts` ENABLE KEYS \*/;

-- Dumping structure for table banksystem.balancetable

DROP TABLE IF EXISTS `balancetable`;

CREATE TABLE IF NOT EXISTS `balancetable` (

`id\_Balance` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`balance` float unsigned NOT NULL DEFAULT '0',

`email` varchar(50) NOT NULL,

PRIMARY KEY (`id\_Balance`),

KEY `FK\_BalanceTable\_accounts\_email` (`email`),

CONSTRAINT `FK\_BalanceTable\_accounts\_email` FOREIGN KEY (`email`) REFERENCES `accounts` (`email`)

) ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=latin1;

-- Dumping data for table banksystem.balancetable: ~2 rows (approximately)

DELETE FROM `balancetable`;

/\*!40000 ALTER TABLE `balancetable` DISABLE KEYS \*/;

INSERT INTO `balancetable` (`id\_Balance`, `balance`, `email`) VALUES

(1, 96, 'csatyendra02@gmail.com'),

(17, 898, 'rooon@gmail.com');

/\*!40000 ALTER TABLE `balancetable` ENABLE KEYS \*/;

-- Dumping structure for table banksystem.clienttable

DROP TABLE IF EXISTS `clienttable`;

CREATE TABLE IF NOT EXISTS `clienttable` (

`id\_client` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`accountNo` varchar(50) NOT NULL,

`firstName` varchar(50) DEFAULT NULL,

`lastName` varchar(50) DEFAULT NULL,

`photo` blob,

`dateOfBirth` varchar(50) DEFAULT NULL,

`accountType` char(50) NOT NULL,

`gender` char(50) DEFAULT NULL,

`telephone` varchar(50) DEFAULT NULL,

`house` varchar(100) DEFAULT NULL,

`neighbourhood` varchar(100) DEFAULT NULL,

`city` varchar(100) DEFAULT NULL,

`country` varchar(100) DEFAULT NULL,

`zipcode` varchar(100) DEFAULT NULL,

`occupation` varchar(50) DEFAULT NULL,

`mobile` varchar(50) DEFAULT NULL,

`accountDepositType` char(50) DEFAULT NULL,

`nationality` varchar(50) DEFAULT NULL,

`Married` char(50) DEFAULT NULL,

`spouceName` varchar(50) DEFAULT NULL,

`birthPlace` varchar(50) DEFAULT NULL,

`children` int(11) DEFAULT NULL,

`bankAgency` varchar(50) DEFAULT NULL,

`email` varchar(50) DEFAULT NULL,

`iban` varchar(50) NOT NULL,

`BIC` varchar(50) NOT NULL,

PRIMARY KEY (`id\_client`),

UNIQUE KEY `accountNo` (`accountNo`),

UNIQUE KEY `iban` (`iban`),

UNIQUE KEY `BIC` (`BIC`),

KEY `FK\_clienttable\_accounts\_email` (`email`),

CONSTRAINT `FK\_clienttable\_accounts\_email` FOREIGN KEY (`email`) REFERENCES `accounts` (`email`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

-- Dumping data for table banksystem.clienttable: ~3 rows (approximately)

DELETE FROM `clienttable`;

/\*!40000 ALTER TABLE `clienttable` DISABLE KEYS \*/;

INSERT INTO `clienttable` (`id\_client`, `accountNo`, `firstName`, `lastName`, `photo`, `dateOfBirth`, `accountType`, `gender`, `telephone`, `house`, `neighbourhood`, `city`, `country`, `zipcode`, `occupation`, `mobile`, `accountDepositType`, `nationality`, `Married`, `spouceName`, `birthPlace`, `children`, `bankAgency`, `email`, `iban`, `BIC`) VALUES

(1, 'e4424234', 'rwerwerw', 'erwer', NULL, '2019-01-03', 'saving', 'M', '343432434', '342342342dsfsdfsfsfs', 'dfffdfdsf', 'rouen', 'france', '76000', 'dfddf', '34343434343', 'saving', 'indian', 'no', 'dfsdf', 'dfdfs', 2, 'rouen', 'csatyendra02@gmail.com', '32323323232', '23223'),

(2, 'SWISSFR60001', NULL, NULL, NULL, NULL, 'C', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, 'FR866462891194831SWISSFR60001', '62891'),

(3, 'SWISSFR78731', 'roon', 'ronny', NULL, NULL, 'C', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, 'rooon@gmail.com', 'FR528420091154041SWISSFR78731', '20091');

/\*!40000 ALTER TABLE `clienttable` ENABLE KEYS \*/;

-- Dumping structure for table banksystem.manager

DROP TABLE IF EXISTS `manager`;

CREATE TABLE IF NOT EXISTS `manager` (

`id\_Manager` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`firstName` varchar(50) DEFAULT NULL,

`lastName` varchar(50) DEFAULT NULL,

`email` varchar(50) NOT NULL,

`emailClient` varchar(50) NOT NULL,

`accountType` char(50) NOT NULL,

PRIMARY KEY (`id\_Manager`),

KEY `FK\_Manager\_accounts\_email` (`email`),

KEY `FK\_Manager\_clienttable\_email` (`emailClient`),

CONSTRAINT `FK\_Manager\_accounts\_email` FOREIGN KEY (`email`) REFERENCES `accounts` (`email`),

CONSTRAINT `FK\_Manager\_clienttable\_email` FOREIGN KEY (`emailClient`) REFERENCES `clienttable` (`email`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table banksystem.manager: ~0 rows (approximately)

DELETE FROM `manager`;

/\*!40000 ALTER TABLE `manager` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `manager` ENABLE KEYS \*/;

-- Dumping structure for table banksystem.officeadvisor

DROP TABLE IF EXISTS `officeadvisor`;

CREATE TABLE IF NOT EXISTS `officeadvisor` (

`id\_OfficeAdvisor` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`firstName` varchar(50) DEFAULT NULL,

`lastName` varchar(50) DEFAULT NULL,

`email` varchar(50) NOT NULL,

`emailClient` varchar(50) NOT NULL,

`accountType` char(50) NOT NULL,

PRIMARY KEY (`id\_OfficeAdvisor`),

KEY `FK\_OfficeAdvisor\_accounts\_email` (`email`),

KEY `FK\_officeadvisor\_clienttable\_email` (`emailClient`),

CONSTRAINT `FK\_OfficeAdvisor\_accounts\_email` FOREIGN KEY (`email`) REFERENCES `accounts` (`email`),

CONSTRAINT `FK\_officeadvisor\_clienttable\_email` FOREIGN KEY (`emailClient`) REFERENCES `clienttable` (`email`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table banksystem.officeadvisor: ~0 rows (approximately)

DELETE FROM `officeadvisor`;

/\*!40000 ALTER TABLE `officeadvisor` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `officeadvisor` ENABLE KEYS \*/;

-- Dumping structure for table banksystem.transactiontable

DROP TABLE IF EXISTS `transactiontable`;

CREATE TABLE IF NOT EXISTS `transactiontable` (

`id\_bankBalance` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`transactionName` varchar(100) NOT NULL,

`debit` float unsigned DEFAULT '0',

`credit` float unsigned DEFAULT '0',

`dateOfTransaction` date NOT NULL,

`email` varchar(50) NOT NULL,

`iban` varchar(50) NOT NULL,

`bic` varchar(50) NOT NULL,

`accountNo` varchar(100) NOT NULL,

`accountNoRecipient` varchar(50) NOT NULL,

`emailIDRecipient` varchar(50) DEFAULT NULL,

`bicRecipient` varchar(50) NOT NULL,

PRIMARY KEY (`id\_bankBalance`),

KEY `FK\_transactiontable\_accounts\_email` (`email`),

KEY `FK\_transactiontable\_clienttable\_iban` (`iban`),

KEY `FK\_transactiontable\_clienttable\_bic` (`bic`),

KEY `FK\_transactiontable\_clienttable\_accountNo` (`accountNo`),

CONSTRAINT `FK\_transactiontable\_accounts\_email` FOREIGN KEY (`email`) REFERENCES `accounts` (`email`),

CONSTRAINT `FK\_transactiontable\_clienttable\_accountNo` FOREIGN KEY (`accountNo`) REFERENCES `clienttable` (`accountNo`),

CONSTRAINT `FK\_transactiontable\_clienttable\_bic` FOREIGN KEY (`bic`) REFERENCES `clienttable` (`BIC`),

CONSTRAINT `FK\_transactiontable\_clienttable\_iban` FOREIGN KEY (`iban`) REFERENCES `clienttable` (`iban`)

) ENGINE=InnoDB AUTO\_INCREMENT=80 DEFAULT CHARSET=latin1;

-- Dumping data for table banksystem.transactiontable: ~67 rows (approximately)

DELETE FROM `transactiontable`;

/\*!40000 ALTER TABLE `transactiontable` DISABLE KEYS \*/;

INSERT INTO `transactiontable` (`id\_bankBalance`, `transactionName`, `debit`, `credit`, `dateOfTransaction`, `email`, `iban`, `bic`, `accountNo`, `accountNoRecipient`, `emailIDRecipient`, `bicRecipient`) VALUES

(1, 'dsfsdfsdfs', 200, 0, '2019-01-05', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '3232323232', 'ddsdasda@gmail.com', '232232'),

(2, 'd d', 0, 1, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', NULL, '23223'),

(3, 'ww ww', 0, 22, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(4, 'ww ww', 0, 22, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(5, 'qq qq', 200, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '3434342342', 'doggy@gmail.com', '32323'),

(6, 'roon ronny', 1000, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'FR528420091154041SWISSFR78731', 'rooon@gmail.com', '20091'),

(7, 'roon ronny', 200, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'FR528420091154041SWISSFR78731', 'rooon@gmail.com', '20091'),

(8, 'roon ronny', 200, 0, '2019-01-23', 'rooon@gmail.com', 'FR528420091154041SWISSFR78731', '20091', 'SWISSFR78731', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(9, 'roon ronny', 100, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'FR528420091154041SWISSFR78731', 'rooon@gmail.com', '20091'),

(10, 'roon ronny', 0, 100, '2019-01-23', 'rooon@gmail.com', 'FR528420091154041SWISSFR78731', '20091', 'SWISSFR78731', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(11, 'satya chan', 200, 0, '2019-01-23', 'rooon@gmail.com', 'FR528420091154041SWISSFR78731', '20091', 'SWISSFR78731', '32323323232', 'csatyendra02@gmail.com', '23223'),

(12, 'satya chan', 0, 200, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'SWISSFR78731', 'rooon@gmail.com', '20091'),

(13, 'a a', 2, 0, '2019-01-23', 'rooon@gmail.com', 'FR528420091154041SWISSFR78731', '20091', 'SWISSFR78731', '2323', '', '323'),

(14, 'satya chandan', 0, 20, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(26, 'dd dd', 0, 20, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(27, 'dd dd', 0, 20, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(28, 'dd dd', 0, 20, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(29, 'dd dd', 0, 20, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(30, 'dd dd', 0, 20, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(31, 'dd dd', 0, 20, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(32, 'dd dd', 0, 20, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(34, 'dd dd', 20, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(35, 'dff dfd', 323, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'dfdssf', '', 'dfsd'),

(36, 'dff dfd', 323, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'dfdssf', '', '44332'),

(37, 'dff dfd', 323, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'dfdssf', '', ''),

(38, 'dff dfd', 0, 323, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(39, 'dff dfd', 0, 323, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(40, 'dff dfd', 0, 323, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(41, 'dff dfd', 0, 323, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(42, 'dff dfd', 0, 323, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(43, 'dff dfd', 0, 323, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(44, 'dff dfd', 0, 323, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(45, 'dff dfd', 0, 323, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(46, 'dff dfd', 0, 323, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'e4424234', 'csatyendra02@gmail.com', '23223'),

(47, 'dd dd', 33, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'dfsd', '', 'dfd'),

(48, 'dd dd', 33, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'dfsd', '', ''),

(49, 'dd dd', 33, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(50, 'dd ', 20, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'dfd', '', 'fdfdf'),

(51, 'dd ', 20, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'dfd', '', 'fdfdf'),

(52, 'dd ', 20, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(53, ' ', 1, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', '1'),

(54, ' ', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(55, ' ', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(56, ' ', 3, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(57, ' ', 3, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(58, ' ', 3, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(59, 'fdfsdfsdf dfsdfs', 3, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'dsfdfs', '', ''),

(60, 'fdfsdfsdf dfsdfs', 3, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', 'dsfdfs', '', ''),

(61, 'fdfsdfsdf dfsdfs', 3, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(62, ' ', 4, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(63, 'dfdfd fdfs', 4, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(64, 'dfdfd fdfs', 4, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(65, 'dfdf ', 3, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(66, ' ', 3, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', '3'),

(67, ' ', 3, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(68, ' ', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(69, 'ewew rer', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(70, ' ', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(71, ' ', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(72, '33 ', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(73, '33 44', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', ''),

(74, '33 44', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '555', '', ''),

(75, '33 44', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '555', '', '444'),

(76, '33 44', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '555', '', '444'),

(77, '33 44', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '555', '', '444'),

(78, '33 44', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', '444'),

(79, '33 44', 2, 0, '2019-01-23', 'csatyendra02@gmail.com', '32323323232', '23223', 'e4424234', '', '', '444');

/\*!40000 ALTER TABLE `transactiontable` ENABLE KEYS \*/;

-- Dumping structure for table banksystem.transfermoney

DROP TABLE IF EXISTS `transfermoney`;

CREATE TABLE IF NOT EXISTS `transfermoney` (

`idTransfer` int(11) NOT NULL AUTO\_INCREMENT,

`FirstName` varchar(50) NOT NULL,

`LastName` varchar(50) NOT NULL,

`bic` varchar(50) NOT NULL,

`amount` float NOT NULL,

`iban` varchar(50) NOT NULL,

PRIMARY KEY (`idTransfer`)

) ENGINE=InnoDB AUTO\_INCREMENT=88 DEFAULT CHARSET=latin1;

-- Dumping data for table banksystem.transfermoney: ~82 rows (approximately)

DELETE FROM `transfermoney`;

/\*!40000 ALTER TABLE `transfermoney` DISABLE KEYS \*/;

INSERT INTO `transfermoney` (`idTransfer`, `FirstName`, `LastName`, `bic`, `amount`, `iban`) VALUES

(1, 'dffdfd', 'dfdfdf', '42323', 20000, '313142133131212'),

(4, 'satyendra', 'chandan', 'rouen', 232331, '2322242232323232'),

(5, 'dango', 'chand', '34342', 499, '34343425245245'),

(6, 'dhc', '', '', 0, ''),

(7, 'jonny', 'walker', '23223', 2000, '32323323232'),

(8, 'dom', 'dom', '23223', 323, '32323323232'),

(9, 'dom', 'dom', '23223', 323, '32323323232'),

(10, 'dom', 'dom', '23223', 323, '32323323232'),

(11, 'dom', 'dom', '23223', 323, '32323323232'),

(12, 'dom', 'dom', '23223', 32, '32323323232'),

(13, 'doom', 'doom', '23223', 33, '32323323232'),

(14, 'doom', 'doom1', '23223', 44, '32323323232'),

(15, 'ddoo', 'dd', '23223', 33, '32323323232'),

(16, 'dd', 'dd', '23223', 22, '32323323232'),

(17, 'dfdf', 'dfdf', '23223', 323, '32323323232'),

(18, 'dfdf', 'dfdf', '23223', 323, '32323323232'),

(19, 'dsd', 'dds', '23223', 33, '32323323232'),

(20, 'eee', 'eee', '23223', 232, '32323323232'),

(21, 'ss', 'ss', '23223', 1, '32323323232'),

(22, 'ss', 'ss', '23223', 1, '32323323232'),

(23, 'dd', 'd', '23223', 1, '32323323232'),

(24, 'd', 'd', '23223', 1, '32323323232'),

(25, 'd', 'd', '23223', 1, '32323323232'),

(26, 'qq', 'qq', '32323', 3, '323232323'),

(27, 'qq', 'qq', '2133131', 22, '1211311'),

(28, 'ww', 'ww', '23223', 22, '32323323232'),

(29, 'qq', 'qq', '32323', 200, '3434342342'),

(30, 'roon', 'ronny', '20091', 1000, 'FR528420091154041SWISSFR78731'),

(31, 'roon', 'ronny', '20091', 200, 'FR528420091154041SWISSFR78731'),

(32, 'roon', 'ronny', '20091', 100, 'FR528420091154041SWISSFR78731'),

(33, 'satya', 'chan', '23223', 200, '32323323232'),

(34, 'a', 'a', '323', 2, '2323'),

(35, 'satya', 'chandan', '23223', 20, '32323323232'),

(36, 'satya', 'chandan', '', 20, '32323323232'),

(37, 'satya', 'chandan', '', 20, '32323323232'),

(38, 'satya', 'chandan', '', 20, '32323323232'),

(39, 'satya', 'chandan', '', 20, '32323323232'),

(40, 'dd', 'dd', '', 20, '32323323232'),

(41, 'dd', 'dd', '', 20, ''),

(42, 'dd', 'dd', '', 0, '32323323232'),

(43, 'dff', 'dfd', 'dfsd', 323, 'dfdssf'),

(44, 'dff', 'dfd', '44332', 323, 'dfdssf'),

(45, 'dff', 'dfd', '', 323, 'dfdssf'),

(46, 'dff', 'dfd', '23223', 323, '32323323232'),

(47, 'dff', 'dfd', '23223', 323, '32323323232'),

(48, 'dff', 'dfd', '23223', 323, '32323323232'),

(49, 'dff', 'dfd', '23223', 323, '32323323232'),

(50, 'dff', 'dfd', '23223', 323, '32323323232'),

(51, 'dff', 'dfd', '23223', 323, '32323323232'),

(52, 'dff', 'dfd', '23223', 323, '32323323232'),

(53, 'dff', 'dfd', '23223', 323, '32323323232'),

(54, 'dff', 'dfd', '23223', 323, '32323323232'),

(55, 'dd', 'dd', 'dfd', 33, 'dfsd'),

(56, 'dd', 'dd', '', 33, 'dfsd'),

(57, 'dd', 'dd', '', 33, ''),

(58, 'dd', '', 'fdfdf', 20, 'dfd'),

(59, 'dd', '', 'fdfdf', 20, 'dfd'),

(60, 'dd', '', '', 20, ''),

(61, '', '', '1', 1, ''),

(62, '', '', '', 2, ''),

(63, '', '', '', 2, ''),

(64, '', '', '', 3, ''),

(65, '', '', '', 3, ''),

(66, '', '', '', 3, ''),

(67, 'fdfsdfsdf', 'dfsdfs', '', 3, 'dsfdfs'),

(68, 'fdfsdfsdf', 'dfsdfs', '', 3, 'dsfdfs'),

(69, 'fdfsdfsdf', 'dfsdfs', '', 3, ''),

(70, '', '', '', 4, ''),

(71, 'dfdfd', 'fdfs', '', 4, ''),

(72, 'dfdfd', 'fdfs', '', 4, ''),

(73, 'dfdf', '', '', 3, ''),

(74, '', '', '3', 3, ''),

(75, '', '', '', 3, ''),

(76, '', '', '', 2, ''),

(77, 'ewew', 'rer', '', 2, ''),

(78, '', '', '', 2, ''),

(79, '', '', '', 2, ''),

(80, '33', '', '', 2, ''),

(81, '33', '44', '', 2, ''),

(82, '33', '44', '', 2, '555'),

(83, '33', '44', '444', 2, '555'),

(84, '33', '44', '444', 2, '555'),

(85, '33', '44', '444', 2, '555'),

(86, '33', '44', '444', 2, ''),

(87, '33', '44', '444', 2, '');

/\*!40000 ALTER TABLE `transfermoney` ENABLE KEYS \*/;

/\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=IF(@OLD\_FOREIGN\_KEY\_CHECKS IS NULL, 1, @OLD\_FOREIGN\_KEY\_CHECKS) \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;