**BANKING SYSTEM**

**Description**:

In this project, we will see how banking system provides Account holders details like Username , Account Number and performs the amount transaction between the account holders like debit ,withdraw using Java program .Further ,it connects with the database to store the details

**Development Steps:**

1. **Design the User Interface**: Create mockups or wireframes of the user interface screens.
2. **Implement Front End**: Develop the JavaFX application with the designed screens and user interactions.
3. **Set Up the Back End**: Create the server-side logic using Java .
4. **Database Setup**: Design and set up the database schema and tables.
5. **Integrate Front End with Back End**: Connect the front end to the back end using HTTP requests or other communication protocols.
6. **Test and Debug**: Test the application thoroughly, identify and fix any issues.
7. **Deployment**: Deploy the application to a server or cloud platform(Github).

**GUI Appilcation:**

package Bank;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.HashMap;

import java.util.Map;

public class BankingSystemGUI extends JFrame {

private Map<String, Double> accounts;

private JTextField UserNameField;

private JTextField accountNumberField;

private JTextField amountField;

private JTextArea outputArea;

public BankingSystemGUI() {

accounts = new HashMap<>();

setTitle("Simple Banking System");

setSize(550, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

createComponents();

}

private void createComponents() {

add(new JLabel("UserName:"));

UserNameField = new JTextField(15);

add(UserNameField);

add(new JLabel("Account Number:"));

accountNumberField = new JTextField(15);

add(accountNumberField);

add(new JLabel("Amount:"));

amountField = new JTextField(15);

add(amountField);

JButton createAccountButton = new JButton("Create Account");

createAccountButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

createAccount();

}

});

add(createAccountButton);

JButton depositButton = new JButton("Deposit");

depositButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

deposit();

}

});

add(depositButton);

JButton withdrawButton = new JButton("Withdraw");

withdrawButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

withdraw();

}

});

add(withdrawButton);

JButton balanceButton = new JButton("Check Balance");

balanceButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

checkBalance();

}

});

add(balanceButton);

outputArea = new JTextArea(10, 30);

outputArea.setEditable(false);

add(new JScrollPane(outputArea));

}

private void createAccount() {

String accountNumber = accountNumberField.getText();

String UserName=UserNameField.getText();

double amount=0;

String output=outputArea.getText();

if (!accounts.containsKey(accountNumber)) {

accounts.put(accountNumber, 0.0);

outputArea.append("Account created: " + accountNumber + "\n");

} else {

outputArea.append("Account already exists: " + accountNumber + "\n");

}

try {

new Db(UserName,accountNumber,amount,output);

} catch (Exception ex) {

throw new RuntimeException(ex);

}

}

private void deposit() {

String accountNumber = accountNumberField.getText();

double amount = Double.parseDouble(amountField.getText());

String UserName=UserNameField.getText();

String output=outputArea.getText();

if (accounts.containsKey(accountNumber)) {

double balance = accounts.get(accountNumber);

balance += amount;

accounts.put(accountNumber, balance);

outputArea.append("Deposited " + amount + "\n");

} else {

outputArea.append("Account not found: " + accountNumber + "\n");

}

try {

new Db(UserName,accountNumber,amount,output);

} catch (Exception ex) {

throw new RuntimeException(ex);

}

}

private void withdraw() {

String accountNumber = accountNumberField.getText();

double amount = Double.parseDouble(amountField.getText());

String UserName=UserNameField.getText();

String output=outputArea.getText();

if (accounts.containsKey(accountNumber)) {

double balance = accounts.get(accountNumber);

if (balance >= amount) {

balance -= amount;

accounts.put(accountNumber, balance);

outputArea.append("Withdrawn " + amount +"\n");

} else {

outputArea.append("Insufficient funds in account " + accountNumber + "\n");

}

} else {

outputArea.append("Account not found: " + accountNumber + "\n");

} try {

new Db(UserName,accountNumber,amount,output);

} catch (Exception ex) {

throw new RuntimeException(ex);

}

}

private void checkBalance() {

String accountNumber = accountNumberField.getText();

Double amount= (double) 0;

String UserName=UserNameField.getText();

String output=outputArea.getText();

if (accounts.containsKey(accountNumber)) {

double balance = accounts.get(accountNumber);

outputArea.append("Balance in account " + balance + "\n");

} else {

outputArea.append("Account not found: " + accountNumber + "\n");

}

try {

new Db(UserName,accountNumber,amount,output);

} catch (Exception ex) {

throw new RuntimeException(ex);

}}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

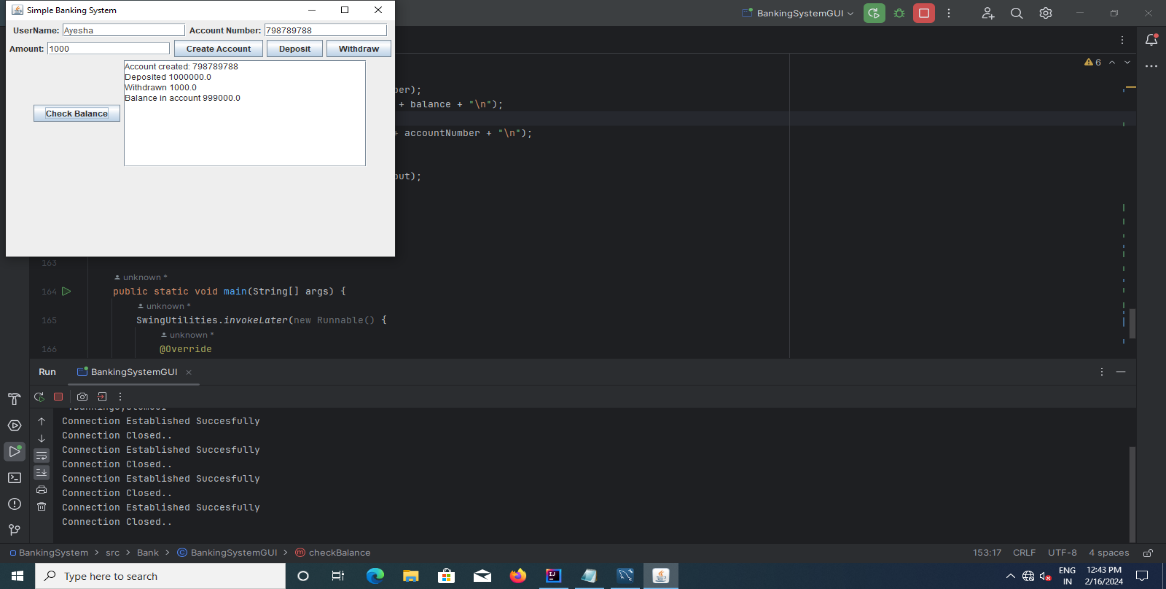
new BankingSystemGUI().setVisible(true);

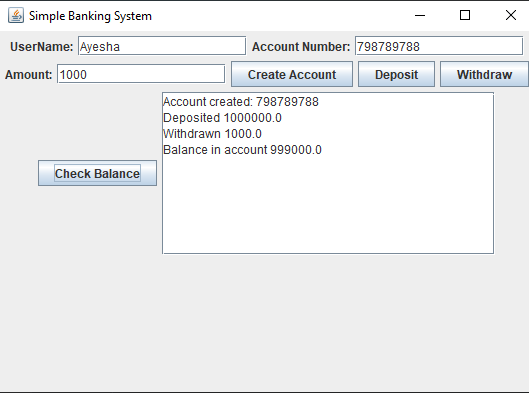
}

});

}}

**Output:**





**DataBaseConnectivity:**

package Bank;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

public class Db {

public Db(String UserName,String accountNumber,Double amount,String output) throws ClassNotFoundException, SQLException {

String url="jdbc:mysql://localhost:3307/Account";

String username="root";

String password="root";

String query="select \*from Account";

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

Connection con = DriverManager.getConnection(url,username,password);

System.out.println("Connection Established Succesfully");

String qu="insert into det(Name,Acct\_no,Amount,Balance)values('"+UserName+"','"+accountNumber+"','"+amount+"','"+output+"')";

Statement smt=con.createStatement();

smt.execute(qu);

con.close();

System.out.println("Connection Closed..");

}

}

**Output:**

