Department of Computer Science

Comsats University Islamabad

Semester project

Group members:

Name:	**Reg.no:**
FAIZA RIAZ	**SP23-BCT-015**
MAHER MUHAMMAD SHAMOON ABBAS	**FA22-BSE-018**

Submitted to: Mam Saneha Amir

Date: 4th June,2024.

Bank Management System:

Classes:

Customer

Account

Manager

Registration

Login

Login time

Account transactions

Bank Management System(Runner)

GUI:

User GUI

User operation GUI

Bank GUI

Simple GUI

Code:

Customer class:

```
package bankmanagementsystem;

import java.util.*;
import java.io.*;
import java.awt.GridLayout;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

public class Customer implements Serializable{
    private Registration register;
```

```java
    private String accountNumber;

    Customer(){
        accountNumber="";
    }
    Customer(Registration r1){
        Scanner input=new Scanner(System.in);//As the Scanner class is not
serializable
        //so to avoid its serialization we're not gonna declare it as instance
variable

        register=r1;
        accountNumber="";

        String otp=generateOTP();
        JOptionPane.showMessageDialog(null,"Your generated OTP is "+otp);
//        System.out.println("Your generated otp: "+otp);

        //making new frame for otp validation
        JFrame otpVerificarion=new JFrame("OTP Verification");

        JButton b1=new JButton("Enter");

        JLabel l1=new JLabel("OTP");
        JTextField t1=new JTextField(20);

        otpVerificarion.setSize(400,400);
        otpVerificarion.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);//only
closes the current frame
        otpVerificarion.setVisible(true);
        otpVerificarion.setLayout(new GridLayout(3,0));

        otpVerificarion.add(l1);
        otpVerificarion.add(t1);
        otpVerificarion.add(b1);

        b1.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent ae){
               String pas=t1.getText();

               if(pas.equals(otp)){
                   accountNumber=generateAccoutNumber();
                   JOptionPane.showMessageDialog(null,"Your Account is
successfully registered.\nYour account number is "+accountNumber);
```

```
//                    UserGUI u=new UserGUI();
//                     System.out.println("Account created successfully");
//                     System.out.println("Your account number is
"+accountNumber);
                }
                else{
                    JOptionPane.showMessageDialog(null,"Sorry.\nIncorrect
OTP...Account is not registered");
                }
            }
        });


//        System.out.print("Please enter the otp: ");
//        String pas=input.next();
    }
    Customer(Customer c){
        register=c.register;
        accountNumber=c.accountNumber;
    }

    public void setRegister(Registration r){
        register=r;
    }
    public void setAccountNumber(String a){
        accountNumber=a;
    }

    public Registration getRegister(){
        return register;
    }
    public String getAccountNumber(){
        return accountNumber;
    }


    private String generateOTP(){//Generating otp for more security
        String otp="";
        for(int i=0;i<4;i++){
            int pass=(int)(Math.random()*10);
            otp+=pass;
        }
        return otp;
    }
```

```java
private String generateAccoutNumber(){//Generating the random Account Number
    String account="";
    char alphabets;
    int numbers;
    for(int i=0;i<7;i++){
        if(i==0 || i==1){
            alphabets=(char)('A'+ Math.random()*26);
            account+=alphabets;
        }
        else{
            numbers=(int)(Math.random()*10);
            account+=numbers;
        }
    }
    accountNumber=account;
    Registration.writeToRegister(Customer.this);
    Customer.writeToCustomer(Customer.this);
    return accountNumber;
}

public void display(){
    register.display();
    System.out.println("Account Number: "+accountNumber);
}

public String toString(){
    return register.toString()+"    Account Number: "+accountNumber;
}



public static void writeToCustomer(Customer c){

try{
    File f=new File("customers.ser");
    ObjectOutputStream oos;

    if(f.exists()){
            oos=new MyObjectOutputStream(new FileOutputStream(f,true));
    }
    else{
        oos=new ObjectOutputStream(new FileOutputStream(f));
    }
        oos.writeObject(c);
```

```
            oos.close();
            System.out.println(c.register.getName()+" is written successfully in
the customers file");
        }
        catch(IOException e){
            System.out.println("Error in file handling for writing in customers
file");
        }
    }


    public static ArrayList<Customer> readAllCustomers(){
        ArrayList<Customer> list=new ArrayList<Customer>();

        ObjectInputStream ois;
        try{
            ois=new ObjectInputStream(new FileInputStream("customers.ser"));
            while(true){
                Customer c=(Customer)ois.readObject();
                list.add(c);
                System.out.println(c.register.getName());
            }
        }
        catch(ClassNotFoundException e1){
            System.out.println("Class Not found for reading from customers
file");
        }
        catch(EOFException e2){
            return list;
        }
        catch(IOException e3){
            System.out.println("Error in file reading of customers file");
            e3.printStackTrace();
        }
        return list;
    }

    public static void deleteACustomer(String accountNum,String pass){
        ArrayList<Customer> list=readAllCustomers();

        boolean flag=true;
        for(int i=0;i<list.size();i++){
            if(list.get(i).accountNumber.equals(accountNum) &&
list.get(i).register.getPassword().equals(pass)){
```

```
                System.out.println(list.get(i).register.getName()+" is deleted
suucessfully in customers file");
                list.remove(i);
                flag=false;
            }
        }
        if(flag){
            System.out.println("Customer not found to be deleted from customers
file");
        }

        try{
            ObjectOutputStream obj=new ObjectOutputStream(new
FileOutputStream("customers.ser"));
            //also deleting the particular customer from registerations file
            ObjectOutputStream oos=new ObjectOutputStream(new
FileOutputStream("registerations.ser"));
            for(int i=0;i<list.size();i++){
                oos.writeObject(list.get(i));
                obj.writeObject(list.get(i));
            }
        }
        catch(IOException e){
            System.out.println("Error in file writing in delete method of
customers file");
        }
    }

    public static void updateACustomer(String accountNum,String pass,String
email,long phone,String address){
        ArrayList<Customer> list=readAllCustomers();

        boolean flag=true;
        for(int i=0;i<list.size();i++){
            if(list.get(i).accountNumber.equals(accountNum) &&
list.get(i).register.getPassword().equals(pass)){
                list.get(i).register.setEmail(email);
                list.get(i).register.setPhone(phone);
                list.get(i).register.setAddress(address);
                System.out.println(list.get(i).register.getName()+"'s email,
phone and home address updated successfully");
                flag=false;
            }
        }
        if(flag){
```

```
            System.out.println("Customer not found to be updated in customers
file");
        }

        try{
            ObjectOutputStream oos=new ObjectOutputStream(new
FileOutputStream("customers.ser"));
            //also writing the updated customer in registrations file
            ObjectOutputStream obj=new ObjectOutputStream(new
FileOutputStream("registerations.ser"));
            for(int i=0;i<list.size();i++){
                obj.writeObject(list.get(i));
                oos.writeObject(list.get(i));
            }
        }
        catch(IOException e){
            System.out.println("Error in writing file in update function in
customers file");
        }

    }

}
```

Account class:

```
package bankmanagementsystem;

import java.io.*;
import java.util.*;
import javax.swing.JOptionPane;

public class Account implements Serializable {
    private Customer customer;
    private String accountType;
    private double balance;
    private String accountID;

    Account(){

    }
    Account(Customer cust,String type,double cash){
        Scanner input=new Scanner(System.in);
```

```
        customer=cust;
        accountType=type;
        balance=cash;

//        String otp=generateOTP();
//        System.out.println("Your generated otp: "+otp);
//        System.out.print("Please enter the otp: ");
//        String pas=input.next();
//
//        if(pas.equals(otp)){
            accountID=generateAccoutID();
            System.out.println("Account created successfully");
            writeToAccount(this);
            System.out.println("Your account ID is "+accountID);
//        }
//        else{
//            System.out.println("Sorry...Incorrect password...Account not
created");
//        }
    }
    Account(Account a){
        customer=a.customer;
        accountType=a.accountType;
        balance=a.balance;
        accountID=a.accountID;
    }

    public void setAccountType(String type){
        accountType=type;
    }
    public void setBalance(double bal){
        balance=bal;
    }
    public void setAccountID(String id){
        accountID=id;
    }
    public void setCustomer(Customer c){
        customer=c;
    }

    public String getAccountType(){
        return accountType;
    }
    public double getBalance(){
```

```
        return balance;
    }
    public String getAccountID(){
        return accountID;
    }
    public Customer getCustomer(){
        return customer;
    }
//    private String generateOTP(){//Generating otp for more security
//        String otp="";
//        for(int i=0;i<4;i++){
//            int pass=(int)(Math.random()*10);
//            otp+=pass;
//        }
//        return otp;
//    }

    private String generateAccoutID(){//Generating the random Account ID
        String account="";
        char alphabets;
        int numbers;
        for(int i=0;i<11;i++){
            if(i==0 || i==1){
                alphabets=(char)('A'+ Math.random()*26);
                account+=alphabets;
            }
            else{
                numbers=(int)(Math.random()*10);
                account+=numbers;
            }
        }
        return account;
    }


    public void display(){
        customer.display();
        System.out.println("Account ID: "+accountID);
        System.out.println("Account Type: "+accountType);
        System.out.println("Balance: "+balance);
    }

    public String toString(){
        return customer.toString()+"    Account ID: "+accountID+"    Account
Type: "+accountType
```

```
            +"    Balance: "+balance;
    }




    public static void writeToAccount(Account a){

    try{
        File f=new File("accounts.ser");
        ObjectOutputStream oos;

        if(f.exists()){
                oos=new MyObjectOutputStream(new FileOutputStream(f,true));
        }
        else{
            oos=new ObjectOutputStream(new FileOutputStream(f));
        }
            oos.writeObject(a);
            oos.close();
            System.out.println(a.getAccountID()+" is written successfully in the
accounts file");
        }
        catch(IOException e){
            System.out.println("Error in file handling for writing in accounts
file");
        }
    }

    public static void updateAnAccount(String accountNum,String pass,String
email,long phone,String address,double amount){
        ArrayList<Account> list=readAllAccounts();

        boolean flag=true;
        for(int i=0;i<list.size();i++){
            Customer cust = list.get(i).getCustomer();
            if(list.get(i).getCustomer()!=null){
                if(list.get(i).getCustomer().getAccountNumber().equals(accountNum
) && list.get(i).getCustomer().getRegister().getPassword().equals(pass)){
                    list.get(i).getCustomer().updateACustomer(accountNum,pass,ema
il,phone,address);
//                  list.get(i).setBalance(amount);
//                    JOptionPane.showMessageDialog(null,"Account updated
Successfully");
                    System.out.println("Account updated successfully");
```

```
                    flag=false;
                }
            }
            else{
                System.out.println("customer is null in updateAnAccount method");
            }
        }
        if(flag){
            JOptionPane.showMessageDialog(null,"Invalid credentials\nAccount not
found");
            System.out.println("Account not found to be updated in accounts
file");
        }

        //writing the updated list into the accounts file
        try{
            ObjectOutputStream obj=new ObjectOutputStream(new
FileOutputStream("accounts.ser"));
            for(int i=0;i<list.size();i++){
                obj.writeObject(list.get(i));
            }
        }
        catch(IOException e){
            System.out.println("Error in file writing in delete method of
accounts file");
        }

    }


    public static ArrayList<Account> readAllAccounts(){
        ArrayList<Account> list=new ArrayList<Account>();

        ObjectInputStream ois;
        try{
            ois=new ObjectInputStream(new FileInputStream("accounts.ser"));
            while(true){
                Account a=(Account)ois.readObject();
                list.add(a);
                System.out.println(a.getAccountID());
            }
        }
        catch(ClassNotFoundException e1){
            System.out.println("Class Not found for reading from accounts file");
        }
```

```
        catch(EOFException e2){
            return list;
        }
        catch(IOException e3){
            System.out.println("Error in file reading of accounts file");
            e3.printStackTrace();
        }
        return list;
    }

    public static void deleteAnAccount(String ID,String type){
        ArrayList<Account> list=readAllAccounts();

        boolean flag=true;
        for(int i=0;i<list.size();i++){
            if(list.get(i).getAccountID().equals(ID) &&
list.get(i).getAccountType().equals(type)){
                System.out.println(list.get(i).getAccountID()+" is deleted
suucessfully in accounts file");
                list.remove(i);
                flag=false;
                JOptionPane.showMessageDialog(null,"Account deleted
successfully");
            }
        }
        if(flag){
            JOptionPane.showMessageDialog(null,"Account not found");
            System.out.println("Customer not found to be deleted from accounts
file");
        }

        try{
            ObjectOutputStream obj=new ObjectOutputStream(new
FileOutputStream("accounts.ser"));
            for(int i=0;i<list.size();i++){
                obj.writeObject(list.get(i));
            }
        }
        catch(IOException e){
            System.out.println("Error in file writing in delete method of
accounts file");
        }
    }

    public void withdraw(double amount){
```

```
        if(amount<=balance){
            balance-=amount;
            updateAnAccount(this.getCustomer().getAccountNumber(),this.getCustome
r().getRegister().getPassword(),
                this.getCustomer().getRegister().getEmail(),this.getCustomer().ge
tRegister().getPhone()
                ,this.getCustomer().getRegister().getAddress(),amount);
            Date d=new Date();
            String str=" Withdrawn "+amount+" on "+d;
            AccountTransaction a=new AccountTransaction(this,str);

            JOptionPane.showMessageDialog(null,amount+" withdrawn successfully
from your account.\nYour current balance is "+this.getBalance());
//            System.out.println(amount+" withdrawn successfully from the
account");
        }
        else{
            JOptionPane.showMessageDialog(null,"Sorry for the
Transaction\n"+amount+" is greater than your Balance.\nYour current balance is
"+this.getBalance());
//            System.out.println("Sorry.Your balance is less for this
transaction");
        }
    }

    public void deposit(double amount){
            balance+=amount;
            updateAnAccount(this.getCustomer().getAccountNumber(),this.getCustome
r().getRegister().getPassword(),
                this.getCustomer().getRegister().getEmail(),this.getCustomer().ge
tRegister().getPhone()
                ,this.getCustomer().getRegister().getAddress(),amount);
            Date d=new Date();
            String str=" Deposited "+amount+" on "+d;
            AccountTransaction a=new AccountTransaction(this,str);
            JOptionPane.showMessageDialog(null,amount+" is deposited successfully
into your account.\nYour current balance is "+this.getBalance());
//            System.out.println(amount+" deposited successfully in the
account");

    }

    public void transfer(double amount,String id){
        ArrayList<Account> list=readAllAccounts();
        boolean notFound=true;
```

```
        boolean small=this.balance>=amount;

        for(int i=0;i<list.size();i++){
            if(id.equals(list.get(i).getAccountID())){
                notFound=false;
                if(small){
                    this.balance-=amount;

                    updateAnAccount(this.getCustomer().getAccountNumber(),this.ge
tCustomer().getRegister().getPassword(),
                        this.getCustomer().getRegister().getEmail(),this.getCusto
mer().getRegister().getPhone()
                        ,this.getCustomer().getRegister().getAddress(),this.balan
ce);
                    Date d=new Date();
                    String str=" Transfered "+amount+" on "+d+" to account ID
"+id;
                    AccountTransaction a1=new AccountTransaction(this,str);
//                    System.out.println(amount+" is transfered to other account
"+id);

                    double newAmount=list.get(i).getBalance()+amount;
                    list.get(i).setBalance(newAmount);
                    updateAnAccount(list.get(i).getCustomer().getAccountNumber(),
list.get(i).getCustomer().getRegister().getPassword(),
                        list.get(i).getCustomer().getRegister().getEmail(),list.g
et(i).getCustomer().getRegister().getPhone()
                        ,list.get(i).getCustomer().getRegister().getAddress(),new
Amount);

                    Date d1=new Date();
                    String str1=" Received "+amount+" on "+d+" from account ID
"+this.accountID;
                    AccountTransaction a2=new AccountTransaction(this,str1);
//                    System.out.println(amount+" is received from other account
"+this.accountID);

                    JOptionPane.showMessageDialog(null,amount+" is successfully
transfered to other account with Account ID: "+id+"\nYour current balance is
"+this.getBalance());
                }
            }
        }
        if(notFound){
```

```
            JOptionPane.showMessageDialog(null,"Sorry for the
Transaction\nAccount with ID "+id+" not found");
//            System.out.println("Account not found to transfer money");
        }
        if(!small){
            JOptionPane.showMessageDialog(null,"Sorry for the transaction\nYou
don't have sufficient balance to transfer to other account\nYour current balance
is "+this.getBalance());
//            System.out.println("Sorry...You don't have sufficient balance to
tranfer money");
        }
}


}
```

Account transaction:

```
package bankmanagementsystem;

import java.io.*;
import java.util.*;

public class AccountTransaction implements Serializable {
    private Account account;
    private String info;

    AccountTransaction(){

    }
    AccountTransaction(Account a,String i){
        account=a;
        info=i;

        writeToAccountTransaction(this);
    }
//    AccountTransaction(AccountTransaction a){
//        account=a.account;
//        info=a.info;
//    }

    public void setAccount(Account a){
        account=a;
```

```java
    }
    public void setInfo(String i){
        info=i;
    }

    public Account getAccount(){
        return account;
    }
    public String getInfo(){
        return info;
    }

    public String toString(){
        return account.toString()+"    Transaction Info:"+info;
    }

    public void display(){
        account.display();
        System.out.println("Transaction Info: "+info);
    }

    public static void writeToAccountTransaction(AccountTransaction a){

        try{
            File f=new File("transactions.ser");
            ObjectOutputStream oos;

            if(f.exists()){
                oos=new MyObjectOutputStream(new FileOutputStream(f,true));
            }
            else{
                oos=new ObjectOutputStream(new FileOutputStream(f));
            }

            oos.writeObject(a);
            oos.close();
            System.out.println("Transaction written successfully in transactions
file");

        }
        catch(IOException e1){
            System.out.println("Error in file writing in transactions file");
        }
    }
```

```
    public static ArrayList<AccountTransaction> readAllAccountTransactions(){
        ArrayList<AccountTransaction> list=new ArrayList<AccountTransaction>();

        ObjectInputStream ois;
        try{
            ois=new ObjectInputStream(new FileInputStream("transactions.ser"));
            while(true){
                AccountTransaction a=(AccountTransaction)ois.readObject();
                list.add(a);
                System.out.println(a.info);
            }
        }
        catch(ClassNotFoundException e1){
            System.out.println("Class Not found for reading from transactions
file");
        }
        catch(EOFException e2){
            return list;
        }
        catch(IOException e3){
            System.out.println("Error in file reading of transactions file");
        }
        return list;
    }

    public static AccountTransaction searchTransaction(Account a){
        ArrayList<AccountTransaction> list=readAllAccountTransactions();

        for(AccountTransaction t:list){
            if(a.getAccountType().equals(t.getAccount().getAccountType()) &&
                a.getAccountID().equals(t.getAccount().getAccountID())){
                return t;
            }
        }
        return new AccountTransaction();
    }

}
```

Login class:

```
package bankmanagementsystem;
```

```java
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.util.*;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;


public class Login implements Serializable {
    private String accountNumber;
    private String password;

    Login(){

    }
    Login(String account,String pass,Customer c){
        accountNumber=account;
        password=pass;
        //to get user logged in
        login(c);
        c.display();
    }
    Login(Login l){
        accountNumber=l.accountNumber;
        password=l.password;
    }

    public String getAccountNumber(){
        return accountNumber;
    }
    public String getPassword(){
        return password;
    }



    private boolean searchInRegisterations(Login l){
        boolean flag=false;

        ArrayList<Customer> list=Customer.readAllCustomers();
        for(Customer c: list){
```

```
            System.out.println("number dekh");
            System.out.println("Checking account: " + c.getAccountNumber() + "
with login account: " + l.getAccountNumber());
            System.out.println("Checking password: " +
c.getRegister().getPassword() + " with login password: " + l.getPassword());

            if(c.getAccountNumber().equals(l.getAccountNumber()) &&
                c.getRegister().getPassword().equals(l.getPassword())){
                flag=true;
                System.out.println("search ho gya");
            }
        }
        System.out.println("search not");
        return flag;
    }

    private String generateOTP(){//Generating otp for more security
        String otp="";
        for(int i=0;i<4;i++){
            int pass=(int)(Math.random()*10);
            otp+=pass;
        }
        return otp;
    }

    private void login(Customer c){
        Scanner input=new Scanner(System.in);

        if(searchInRegisterations(Login.this)){
            String otp=generateOTP();
            JOptionPane.showMessageDialog(null,"Your generated OTP is "+otp);
//            System.out.println("Your generated otp: "+otp);

            //making new frame for otp validation
            JFrame otpVerificarion=new JFrame("OTP Verification");

            JButton b1=new JButton("Enter");

            JLabel l1=new JLabel("OTP");
            JTextField t1=new JTextField(20);

            otpVerificarion.setSize(400,400);
            otpVerificarion.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);//o
nly closes the current frame
            otpVerificarion.setVisible(true);
```

```
            otpVerificarion.setLayout(new GridLayout(3,0));

            otpVerificarion.add(l1);
            otpVerificarion.add(t1);
            otpVerificarion.add(b1);

            b1.addActionListener(new ActionListener(){
                public void actionPerformed(ActionEvent ae){
                    String pas=t1.getText();

                    if(pas.equals(otp)){
//                      System.out.println("Successfully logged in customer
account");
                        JOptionPane.showMessageDialog(null,"Successfully log into
the account");
                        //also writing to the logins file to keep user login
track
                        writeToLogin(Login.this); // Explicitly reference the
outer class's instance;

                        //to let UserOperationsGUI have the accessof Customer and
Login
                        new UserOperationsGUI(c,Login.this);


                    }
                    else{
//                      System.out.println("Wrong Account number or
password.\nUnable to login.");
                        JOptionPane.showMessageDialog(null,"Unable to
login.\nWrong OTP");
                    }
                }
            });
//            System.out.print("Please enter the otp: ");
//            String pas=input.next();
            }
        else{
            JOptionPane.showMessageDialog(null,"Invalid Account number or
password");
//            System.out.println("Customer Account not registered");
        }
    }

    public static void writeToLogin(Login l){
```

```
        try{
            File f=new File("logins.ser");
            ObjectOutputStream oos;
            LoginTime t=new LoginTime(l);

            //creating the object of LoginTime class
            if(f.exists()){
                oos=new MyObjectOutputStream(new FileOutputStream(f,true));
            }
            else{
                oos=new ObjectOutputStream(new FileOutputStream(f));
            }

            oos.writeObject(t);
            oos.close();
            System.out.println("Login time written successfully in file");
        }
        catch(IOException e){
            System.out.println("Error writing in logins file");
            e.printStackTrace();
        }
    }

    public static ArrayList<LoginTime> readAllLogins(){
        ArrayList<LoginTime> list=new ArrayList<LoginTime>();

        try{
            ObjectInputStream ois=new ObjectInputStream(new
FileInputStream("logins.ser"));
            while(true){
                LoginTime t=(LoginTime)ois.readObject();
                System.out.println(t.currentTime);
                list.add(t);
            }
        }
        catch(ClassNotFoundException e1){
            System.out.println("Class not found exception in logins file
reading");
        }
        catch(EOFException e2){
            return list;
        }
        catch(IOException e3){
            System.out.println("Error in file reading in logins file");
            e3.printStackTrace();
```

```
        }

        return list;
    }

    public static boolean searchLogin(Login L){
        ArrayList<LoginTime> list=readAllLogins();

        for(LoginTime t:list){
            if(t.login.getAccountNumber().equals(t.login.getAccountNumber())
                && t.login.getPassword().equals(t.login.getPassword())){
                return true;
            }
        }
        return false;
    }

    public String toString(){
        return "Account number: "+accountNumber+"    Password: "+password;
    }
    public void display(){
        System.out.println("Account Number: "+accountNumber+"\nPassword:
"+password);
    }
}
```

Login time:

```
package bankmanagementsystem;

import java.io.*;
import java.util.*;


public class LoginTime implements Serializable{
    Login login;
    String currentTime;

    public LoginTime(){

    }
    LoginTime(Login l1){
        login=l1;
```

```
        currentTime=findTime();
    }

    private String findTime(){
        Date d = new Date();
        return d.toString();
    }

    public Login getLogin(){
        return login;
    }
    public String getCurrentTime(){
        return currentTime;
    }

    public String toString(){
        return login.toString()+"    Current Time: "+currentTime;
    }
    public void display(){
        login.display();
        System.out.println("Current Time: "+currentTime);
    }
}
```

Manager class:

```
package bankmanagementsystem;

import static bankmanagementsystem.Login.writeToLogin;

import java.io.*;
import java.util.*;

public class Manager implements Serializable{
    private String userName;
    private String email;
    private String password;

    Manager(){

    }
    Manager(String user,String mail,String pass){
        userName=user;
```

```
        email=mail;
        password=pass;
        writeToManager(this);
    }
    Manager(Manager m){
        userName=m.userName;
        email=m.email;
        password=m.password;
    }

    public String getUserName(){
        return userName;
    }
    public String getEmail(){
        return email;
    }
    public String getPassword(){
        return password;
    }

    public String toString(){
        return "User Name: "+userName+"    Email: "+email+"    Password:
"+password;
    }
    public void display(){
        System.out.println("User Name: "+userName+"\nEmail: "+email+"\nPassword:
"+password);
    }

    public static ArrayList<Account> viewAllAccounts(){
        return Account.readAllAccounts();
    }

    public AccountTransaction viewTransactionHistory(String custName,String
type){
        ArrayList<AccountTransaction>
list=AccountTransaction.readAllAccountTransactions();

        for(AccountTransaction i:list){
            if(i.getAccount().getCustomer().getRegister().getName().equals(custNa
me)
                && i.getAccount().getAccountType().equals(type)){
                return i;
            }
        }
```

```
        return null;
    }

    public void closeCustomerAccount(String id,String name,String type){
        ArrayList<Account> list=Account.readAllAccounts();

        boolean flag=true;
        for(Account i:list){
            if(i.getAccountID().equals(id) && i.getAccountType().equals(type)
                && i.getCustomer().getRegister().getName().equals(name)){
                flag=false;
                Account.deleteAnAccount(id,type);
                System.out.println(name+"'s Account removed successfully by
manager");
            }
        }
        if(flag){
            System.out.println("Account not found to be deleted by manager.");
        }
    }

    public static void writeToManager(Manager m){
        try{
        File f=new File("manager.ser");
        ObjectOutputStream oos;

        if(f.exists()){
                oos=new MyObjectOutputStream(new FileOutputStream(f,true));
        }
        else{
            oos=new ObjectOutputStream(new FileOutputStream(f));
        }
            oos.writeObject(m);
            oos.close();
            System.out.println(m.getUserName()+" is written successfully in the
manager file");
        }
        catch(IOException e){
            System.out.println("Error in file handling for writing in manager
file");
        }
    }

    public static ArrayList<Manager> readAllManagers(){
        ArrayList<Manager> list=new ArrayList<Manager>();
```

```
        ObjectInputStream ois;
        try{
            ois=new ObjectInputStream(new FileInputStream("manager.ser"));
            while(true){
                Manager m=(Manager)ois.readObject();
                list.add(m);
                System.out.println(m.getUserName());
            }
        }
        catch(ClassNotFoundException e1){
            System.out.println("Class Not found for reading from manager file");
        }
        catch(EOFException e2){
            return list;
        }
        catch(IOException e3){
            System.out.println("Error in file reading of manager file");
        }
        return list;
    }

    public static boolean searchAManager(Manager m){
        ArrayList<Manager> list=readAllManagers();

        for(Manager i:list){
            if(i.getUserName().equals(m.getUserName())
                && i.getEmail().equals(m.getEmail()) &&
i.getPassword().equals(m.getPassword())){
                return true;
            }
        }
        return false;
    }

    public void managerLogin(){
        Scanner input=new Scanner(System.in);

        if(searchAManager(this)){
            String otp=generateOTP();
            System.out.println("Your generated otp: "+otp);
            System.out.print("Please enter the otp: ");
            String pas=input.next();

            if(pas.equals(otp)){
```

```
                System.out.println("Successfully logged in");

            }
            else{
                System.out.println("Can't login to Manager Account. Wrong
Credentials");
            }
        }
        else{
            System.out.println("Invalid Manager's Account");
        }
    }

    private String generateOTP(){//Generating otp for more security
        String otp="";
        for(int i=0;i<4;i++){
            int pass=(int)(Math.random()*10);
            otp+=pass;
        }
        return otp;
    }
}
```

Registration:

```
package bankmanagementsystem;

import java.io.*;
import java.util.*;


public class Registration implements Serializable {
    private String name;
    private String email;
    private String CNIC;
    private String address;
    private long phone;
    private String password;


    Registration(){

    }
```

```java
    Registration(String n,String e,String c,String a,long p,String pas){
        name=n;
        email=e;
        CNIC=c;
        address=a;
        phone=p;
        password=pas;
    }
    Registration(Registration r){
        name=r.name;
        email=r.email;
        CNIC=r.CNIC;
        address=r.address;
        phone=r.phone;
        password=r.password;
    }

    public void setName(String n){
        name=n;
    }
    public void setEmail(String e){
        email=e;
    }
    public void setCNIC(String c){
        CNIC=c;
    }
    public void setAddress(String a){
        address=a;
    }
    public void setPhone(long p){
        phone=p;
    }
    public void setPassword(String pass){
        password=pass;
    }

    public String getName(){
        return name;
    }
    public String getEmail(){
        return email;
    }
    public String getCNIC(){
        return CNIC;
    }
```

```
    public String getAddress(){
        return address;
    }
    public long getPhone(){
        return phone;
    }
    public String getPassword(){
        return password;
    }

//    public static void writeAdminToRegister(Customer c){
//
//    try{
//        File f=new File("registrations.ser");
//        ObjectOutputStream oos;
//
//        if(f.exists()){
//                oos=new MyObjectOutputStream(new FileOutputStream(f,true));
//        }
//        else{
//            oos=new ObjectOutputStream(new FileOutputStream(f));
//        }
//            oos.writeObject(c);
//            oos.close();
//            System.out.println(c.getRegister().getName()+" Admin is written
successfully in the customers file");
//        }
//        catch(IOException e){
//            System.out.println("Error in file handling for writing in customers
file");
//        }
//    }

    public String toString(){
        return "Name: "+name+"    Email: "+email+"    CNIC: "+CNIC+"    Address:
"+address+"    Phone: "+phone
                +"    Password: "+password;
    }

    public void display(){
        System.out.println("Name: "+name+"\nEmail: "+email+"\nCNIC:
"+CNIC+"\nAddress: "+address+"\nPhone: "+phone
            +"\nPassword: "+password);
    }
```

```
    public static void writeToRegister(Customer c){

        File f=new File("registrations.ser");
        ObjectOutputStream oos;
        try{
            if(f.exists()){
                oos=new MyObjectOutputStream(new FileOutputStream(f,true));
            }
            else{
                oos=new ObjectOutputStream(new FileOutputStream(f));
            }
            if(!searchARegistration(c)){
                oos.writeObject(c);
                oos.close();
                System.out.println(c.getRegister().getName()+" written
successfully in registrations file");
            }
            else{
                System.out.println("Already Registered.Can't write in
registrations file");
            }

        }
        catch(IOException e){
            System.out.println("Error in file writing of registrations file");
        }
    }

    public static ArrayList<Customer> readAllRegistrations(){
        ArrayList<Customer> list=new ArrayList<Customer>();

        try{
            ObjectInputStream ois=new ObjectInputStream(new
FileInputStream("registrations.ser"));

            while(true){
                Customer c=(Customer)ois.readObject();
                System.out.println(c.getRegister().getName());
                list.add(c);
            }
        }
        catch(ClassNotFoundException e1){
            System.out.println("Class not found for file reading in registrations
file");
        }
```

```
        catch(EOFException e2){
            return list;
        }
        catch(IOException e3){
            System.out.println("Error in file reading in registrations file");
        }
        return list;
    }

    public static boolean searchARegistration(Customer c){
        ArrayList<Customer> list=readAllRegistrations();

        boolean flag=false;
        for(Customer c1:list){
            if(c.getRegister().getEmail().equals(c1.getRegister().getEmail())
                && c.getRegister().getCNIC().equals(c1.getRegister().getCNIC())
                &&
c.getRegister().getAddress().equalsIgnoreCase(c1.getRegister().getAddress())
                && c.getRegister().getPhone()==(c1.getRegister().getPhone())){
                flag=true;
            }
        }
        return flag;
    }


}
```

Bank GUI:

```
package bankmanagementsystem;

import java.awt.GridLayout;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;


public class BankGUI {
    JFrame bank=new JFrame("Bank");
    JButton button1;
```

```
    BankGUI(){
        bank.setSize(400,400);
        bank.setDefaultCloseOperation(EXIT_ON_CLOSE);
        bank.setVisible(true);
        bank.setLayout(new GridLayout());

        button1=new JButton("WELCOME TO APNA SECURE BANK");

        bank.add(button1);

        button1.addActionListener(new ActionListener(){
           public void actionPerformed(ActionEvent ae){
               SimpleGUI g=new SimpleGUI();
           }
        });
    }
}
```

User GUI:

```
package bankmanagementsystem;

import java.awt.GridLayout;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

public class UserGUI {

    UserGUI(){
        JFrame userHome=new JFrame("User Home");
        JButton b1;
        JButton b2;
        JButton b3;

        userHome.setSize(400,400);
        userHome.setDefaultCloseOperation(EXIT_ON_CLOSE);
        userHome.setVisible(true);
        userHome.setLayout(new GridLayout(3,0));
        b1=new JButton("Register");
        b2=new JButton("Login");
```

```
        b3=new JButton("Home");

        userHome.add(b1);
        userHome.add(b2);
        userHome.add(b3);

        b1.addActionListener(new ActionListener(){

                    //Anonymous class to avoid making many classes
            @Override
            public void actionPerformed(ActionEvent ae){
                JFrame userRegister=new JFrame("User Registration");

                userRegister.setSize(400,400);
                userRegister.setDefaultCloseOperation(EXIT_ON_CLOSE);
                userRegister.setVisible(true);
                userRegister.setLayout(new GridLayout(7,1));

                JLabel l1=new JLabel("Name");
                JTextField t1=new JTextField(20);
                JLabel l2=new JLabel("Email");
                JTextField t2=new JTextField(20);
                JLabel l3=new JLabel("CNIC");
                JTextField t3=new JTextField(20);
                JLabel l4=new JLabel("Address");
                JTextField t4=new JTextField(20);
                JLabel l5=new JLabel("Phone");
                JTextField t5=new JTextField(20);
                JLabel l6=new JLabel("Password");
                JTextField t6=new JTextField(20);

                JButton button1=new JButton("Register");
                JButton button2=new JButton("User Home");

                userRegister.add(l1);
                userRegister.add(t1);
                userRegister.add(l2);
                userRegister.add(t2);
                userRegister.add(l3);
                userRegister.add(t3);
                userRegister.add(l4);
                userRegister.add(t4);
                userRegister.add(l5);
                userRegister.add(t5);
                userRegister.add(l6);
```

```
                userRegister.add(t6);

                userRegister.add(button1);
                userRegister.add(button2);

                button1.addActionListener(new ActionListener(){

                    public void actionPerformed(ActionEvent ae){

                        String name=t1.getText();
                        String mail=t2.getText();
                        String cnic=t3.getText();
                        String adrs=t4.getText();
                        long ph=Integer.parseInt(t5.getText());
                        String pswd=t6.getText();

                        Customer c=new Customer(new
Registration(name,mail,cnic,adrs,ph,pswd));

                    }
                });

                button2.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent ae){
                        UserGUI s=new UserGUI();
                    }
                });
            }
        });

        b2.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent ae){
                JFrame userLogin=new JFrame("User Login");

                userLogin.setSize(400,400);
                userLogin.setDefaultCloseOperation(EXIT_ON_CLOSE);
                userLogin.setLayout(new GridLayout(3,1));
                userLogin.setVisible(true);


                JLabel l1=new JLabel("Account Number");
                JTextField t1= new JTextField(20);
                JLabel l2=new JLabel("Password");
                JTextField t2= new JTextField(20);
```

```
                JButton button1=new JButton("Login");
                JButton button2=new JButton("User Home");

                userLogin.add(l1);
                userLogin.add(t1);
                userLogin.add(l2);
                userLogin.add(t2);
                userLogin.add(button1);
                userLogin.add(button2);

                button1.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent ae){
                        String accountNum=t1.getText();
                        String pass=t2.getText();


                        ArrayList<Customer> list=Customer.readAllCustomers();
                        for(Customer c1:list){
                            if(c1.getAccountNumber().equals(accountNum)){
                                if (c1 != null) {
                                    Login lo = new Login(accountNum, pass, c1);
                                }
                                else {
                                    JOptionPane.showMessageDialog(null, "No
registered customer found. Please register first.");
                                }
                            }
                        }


                    }
                });

                button2.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent ae){
                        UserGUI u=new UserGUI();
                    }
                });
            }
        });

        b3.addActionListener(new ActionListener(){
           public void actionPerformed(ActionEvent ae){
               SimpleGUI a=new SimpleGUI();
           }
```

```
        });
    }

}
```

Manager GUI:

```
package bankmanagementsystem;

import java.awt.GridLayout;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

public class ManagerGUI {
    private String userName;
    private String email;
    private String password;
    private String otp;

    ManagerGUI(){
        Manager m=new Manager("admin","admin@gmail.com","12345");
        Manager.writeToManager(m);
        JFrame man=new JFrame("Manager Login");
        man.setSize(400,400);
                man.setDefaultCloseOperation(EXIT_ON_CLOSE);
                man.setVisible(true);
                man.setLayout(new GridLayout(8,0));

                JLabel l1=new JLabel("Name");
                JTextField t1=new JTextField(20);
                JLabel l2=new JLabel("Email");
                JTextField t2=new JTextField(20);
                JLabel l3=new JLabel("Password");
                JTextField t3=new JTextField(20);

                JButton button1=new JButton("Login");
                JButton button2=new JButton("Home");

                man.add(l1);
                man.add(t1);
                man.add(l2);
```

```
                man.add(t2);
                man.add(l3);
                man.add(t3);

                man.add(button1);
                man.add(button2);


                button1.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent ae){
                        String n=t1.getText();
                        String m=t2.getText();
                        String p=t3.getText();


                        boolean flag=false;
                        ArrayList<Manager> list=Manager.readAllManagers();
                        for(Manager i:list){
                            if(i.getUserName().equals(n) &&
i.getEmail().equals(m) && i.getPassword().equals(p)){
                                flag=true;
                            }
                        }

                        if(flag){
                            otp="";
                            for(int i=0;i<4;i++){
                                int pass1=(int)(Math.random()*10);
                                otp+=pass1;
                            }

                            JOptionPane.showMessageDialog(null,"Your generated
OTP is "+otp);
                            JFrame otpVerificarion=new JFrame("Account Login");

                            JButton b1=new JButton("Enter");

                            JLabel l1=new JLabel("OTP");
                            JTextField t1=new JTextField(20);

                            otpVerificarion.setSize(400,400);
                            otpVerificarion.setDefaultCloseOperation(JFrame.DISPO
SE_ON_CLOSE);//only closes the current frame
                            otpVerificarion.setVisible(true);
                            otpVerificarion.setLayout(new GridLayout(3,0));
```

```
                            otpVerificarion.add(l1);
                            otpVerificarion.add(t1);
                            otpVerificarion.add(b1);

                            b1.addActionListener(new ActionListener(){
                                public void actionPerformed(ActionEvent ae){
                                    String pas=t1.getText();

                                    if(pas.equals(otp)){

                                        JFrame created=new JFrame("Manager
Operations");
                                        created.setSize(400,400);
                                        created.setLayout(new GridLayout(2,1));
                                        created.setVisible(true);
                                        created.setDefaultCloseOperation(EXIT_ON_
CLOSE);

                                        JButton btn1=new JButton("View
Accounts");
                                        JButton btn2=new JButton("Manager Home");

                                        created.add(btn1);
                                        created.add(btn2);

                                        btn1.addActionListener(new
ActionListener(){
                                            public void
actionPerformed(ActionEvent ae){

                                                ArrayList<Account>
list=Manager.viewAllAccounts();

                                                StringBuilder accounts=new
StringBuilder();
                                                if(list.isEmpty())
                                                    accounts.append("No Accounts
yet");

                                                for(Account i:list){
                                                    accounts.append(i).append("\n
");
                                                }
```

```
                                                JFrame show=new JFrame("View All
Accounts");
                                                show.setSize(400, 300);
                                                JTextArea textArea = new
JTextArea(accounts.toString());
                                                textArea.setEditable(false);
                                                show.add(new
JScrollPane(textArea));
                                                show.setVisible(true);
                                            }
                                        });

                                        btn2.addActionListener(new
ActionListener(){
                                            public void
actionPerformed(ActionEvent ae){
                                                ManagerGUI g=new ManagerGUI();
                                            }
                                        });

                                    }
                                    else{
                                        JOptionPane.showMessageDialog(null,"Wrong
OTP\nUnable to login`");
                                    }
                                }

                });
                        }else{
                            JOptionPane.showMessageDialog(null,"Invalid
credentials\nUnable to login`");
                        }




    }
                });

                button2.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent ae){
                        SimpleGUI s=new SimpleGUI();
                    }
                });
    }
```

```
}
```

User operation GUI:

```
package bankmanagementsystem;

//import static bankmanagementsystem.Login.writeToLogin;
import java.awt.GridLayout;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

public class UserOperationsGUI {

    Account a;
    Customer customer;
    Login login;
    private String otp;

    UserOperationsGUI(Customer c,Login l){
//        c.display();
        customer=c;
        login=l;

        JFrame newly=new JFrame("User Operations");
        newly.setSize(400,400);
        newly.setLayout(new GridLayout(3,0));
        newly.setVisible(true);
        newly.setDefaultCloseOperation(EXIT_ON_CLOSE);

        JButton b1=new JButton("Account Creation");
        JButton b2=new JButton("Account Login");
        JButton b3=new JButton("User Home");

        newly.add(b1);
        newly.add(b2);
        newly.add(b3);

        b1.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent ae){
                JFrame info=new JFrame("Account Creation");
```

```
                info.setSize(400,400);
                info.setLayout(new GridLayout(4,2));
                info.setVisible(true);
                info.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

                JLabel l1=new JLabel("Account Type");
                JTextField t1=new JTextField(20);
                JLabel l2=new JLabel("Cash");
                JTextField t2=new JTextField(20);

                JButton button1=new JButton("Create Account");
                JButton button2=new JButton("Back to User Operation page");

                info.add(l1);
                info.add(t1);
                info.add(l2);
                info.add(t2);
                info.add(button1);
                info.add(button2);

                button1.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent ae){
                        String type=t1.getText();
                        double cash=Integer.parseInt(t2.getText());

                        a=new Account(customer,type,cash);
                        JOptionPane.showMessageDialog(null,"Account created
successfully\nYour Account ID is "+a.getAccountID());


                        JFrame created=new JFrame("User Operations");
                        created.setSize(400,400);
                        created.setLayout(new GridLayout(9,0));
                        created.setVisible(true);
                        created.setDefaultCloseOperation(EXIT_ON_CLOSE);

                        JButton bro1=new JButton("Check Balance");
                        JButton bro2=new JButton("Withdraw Money");
                        JButton bro3=new JButton("Deposit Money");
                        JButton bro4=new JButton("Transfer Money");
                        JButton bro5=new JButton("Edit Account Details");
                        JButton bro6=new JButton("Transaction History");
                        JButton bro7=new JButton("Login History");
                        JButton bro8=new JButton("Delete an Account");
```

```
                        JButton bro9=new JButton("Log out");

                        created.add(bro1);
                        created.add(bro2);
                        created.add(bro3);
                        created.add(bro4);
                        created.add(bro5);
                        created.add(bro6);
                        created.add(bro7);
                        created.add(bro8);
                        created.add(bro9);



                        bro1.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){
                                JOptionPane.showMessageDialog(null,"Your current
balance is "+a.getBalance());
                            }
                        });

                        bro2.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Withdraw Money");
                                with.setSize(400,400);
                                with.setLayout(new GridLayout(3,0));
                                with.setVisible(true);
                                with.setDefaultCloseOperation(JFrame.DISPOSE_ON_C
LOSE);

                                JLabel lab1=new JLabel("Amount to be Withdrawn");
                                JTextField m=new JTextField(20);
                                JButton btn1=new JButton("Withdraw");

                                with.add(lab1);
                                with.add(m);
                                with.add(btn1);


                                btn1.addActionListener(new ActionListener(){
                                    public void actionPerformed(ActionEvent ae){
                                        double
cash=Integer.parseInt(m.getText());
                                        a.withdraw(cash);
```

```
                                    }
                                });
                            }
                        });

                        bro3.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Deposit Money");
                                with.setSize(400,400);
                                with.setLayout(new GridLayout(3,0));
                                with.setVisible(true);
                                with.setDefaultCloseOperation(JFrame.DISPOSE_ON_C
LOSE);

                                JLabel lab1=new JLabel("Amount to be Deposited");
                                JTextField m=new JTextField(20);
                                JButton btn1=new JButton("Deposit");

                                with.add(lab1);
                                with.add(m);
                                with.add(btn1);


                                btn1.addActionListener(new ActionListener(){
                                    public void actionPerformed(ActionEvent ae){
                                        double
cash=Integer.parseInt(m.getText());
                                        a.deposit(cash);

                                    }
                                });
                            }
                        });


                        bro4.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Tranfer Money");
                                with.setSize(400,400);
                                with.setLayout(new GridLayout(5,0));
                                with.setVisible(true);
```

```
                                with.setDefaultCloseOperation(JFrame.DISPOSE_ON_C
LOSE);

                                JLabel lab1=new JLabel("Amount to be
Transfered");
                                JTextField m=new JTextField(20);
                                JLabel lab2=new JLabel("Account ID of receiver
Account");
                                JTextField m1=new JTextField(20);
                                JButton btn1=new JButton("Transfer");

                                with.add(lab1);
                                with.add(m);
                                with.add(lab2);
                                with.add(m1);
                                with.add(btn1);


                                btn1.addActionListener(new ActionListener(){
                                    public void actionPerformed(ActionEvent ae){
                                        double
cash=Integer.parseInt(m.getText());
                                        String id=m1.getText();

                                        a.transfer(cash, id);
                                    }
                                });
                            }
                        });


                        bro5.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Edit Account Details");
                                with.setSize(400,400);
                                with.setLayout(new GridLayout(11,0));
                                with.setVisible(true);
                                with.setDefaultCloseOperation(JFrame.DISPOSE_ON_C
LOSE);

                                JLabel lab1=new JLabel("Account Number");
                                JTextField m=new JTextField(20);
                                JLabel lab2=new JLabel("Password");
                                JTextField m1=new JTextField(20);
```

```
                                JLabel lab3=new JLabel("New Email");
                                JTextField m2=new JTextField(20);
                                JLabel lab4=new JLabel("New Phone");
                                JTextField m3=new JTextField(20);
                                JLabel lab5=new JLabel("New Address");
                                JTextField m4=new JTextField(20);

                                JButton btn1=new JButton("Update");

                                with.add(lab1);
                                with.add(m);
                                with.add(lab2);
                                with.add(m1);
                                with.add(lab3);
                                with.add(m2);
                                with.add(lab4);
                                with.add(m3);
                                with.add(lab5);
                                with.add(m4);
                                with.add(btn1);


                                btn1.addActionListener(new ActionListener(){
                                    public void actionPerformed(ActionEvent ae){
                                        String accountNum=m.getText();
                                        String pass=m1.getText();
                                        String email=m2.getText();
                                        long
phone=Integer.parseInt(m3.getText());
                                        String address=m4.getText();
                                        double amount=0;
                                        Account.updateAnAccount(accountNum,pass,e
mail,phone,address,amount);
                                        JOptionPane.showMessageDialog(null,"Accou
nt updated Successfully");
                                    }
                                });
                            }
                        });


                        bro6.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Transaction History");
```

```
                                StringBuilder history=new StringBuilder();
                                ArrayList<AccountTransaction>
list=AccountTransaction.readAllAccountTransactions();
                                for(AccountTransaction i:list){
                                    if(i.getAccount().getAccountID().equals(a.get
AccountID())){
                                        history.append(i).append("\n");
                                    }
                                }
                                if(history.isEmpty()){
                                    history.append("No transactions made yet");
                                }

                                with.setSize(400, 300);
                                JTextArea textArea = new
JTextArea(history.toString());
                                textArea.setEditable(false);
                                with.add(new JScrollPane(textArea));
                                with.setVisible(true);

                            }
                        });


                        bro7.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Login History");

                                StringBuilder history=new StringBuilder();
                                ArrayList<LoginTime> list=Login.readAllLogins();
                                for(LoginTime i:list){
                                    System.out.println("hota ha");
                                    if(i.getLogin().getAccountNumber().equals(log
in.getAccountNumber())
                                        &&
i.getLogin().getPassword().equals(login.getPassword())){
                                        history.append(i).append("\n");
                                    }
                                }
                                if(history.isEmpty()){
                                    history.append("No logins yet");
                                }
```

```
                                with.setSize(400, 300);
                                JTextArea textArea = new
JTextArea(history.toString());
                                textArea.setEditable(false);
                                with.add(new JScrollPane(textArea));
                                with.setVisible(true);

                            }
                        });


                        bro8.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){


                                otp="";
                                for(int i=0;i<4;i++){
                                    int pass=(int)(Math.random()*10);
                                    otp+=pass;
                                }

                                JOptionPane.showMessageDialog(null,"Your
generated OTP is "+otp);
//            System.out.println("Your generated otp: "+otp);

            //making new frame for otp validation
                                JFrame otpVerificarion=new JFrame("Account
Deletion");

                                JButton b1=new JButton("Enter");

                                JLabel l1=new JLabel("OTP");
                                JTextField t1=new JTextField(20);

                                otpVerificarion.setSize(400,400);
                                otpVerificarion.setDefaultCloseOperation(JFrame.D
ISPOSE_ON_CLOSE);//only closes the current frame
                                otpVerificarion.setVisible(true);
                                otpVerificarion.setLayout(new GridLayout(3,0));

                                otpVerificarion.add(l1);
                                otpVerificarion.add(t1);
                                otpVerificarion.add(b1);

                                b1.addActionListener(new ActionListener(){
```

```
                                    public void actionPerformed(ActionEvent ae){
                                        String pas=t1.getText();

                                        if(pas.equals(otp)){

                                            Account.deleteAnAccount(a.getAccountI
D(),a.getAccountType());
                                        }
                                        else{
                                            JOptionPane.showMessageDialog(null,"U
nable to login.\nWrong OTP");
                                        }

                                    }
                                });

                            }
                        });


                        bro9.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){
                                JOptionPane.showMessageDialog(null,"Loged out
successfully");
                                UserGUI h=new UserGUI();
                            }
                        });

                    }
                });


                    button2.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent ae){
                            UserOperationsGUI g=new
UserOperationsGUI(customer,login);
                        }
                    });

                }
            });


        b2.addActionListener(new ActionListener(){
            @Override
```

```
            public void actionPerformed(ActionEvent ae){
                JFrame jeo=new JFrame("Account Login");
                JLabel l1=new JLabel("Account ID");
                JTextField t1=new JTextField(20);
                JLabel l2=new JLabel("Password");
                JTextField t2=new JTextField(20);

                JButton btn1=new JButton("Login");
                JButton btn2=new JButton("Back to User Operation page");

                jeo.add(l1);
                jeo.add(t1);
                jeo.add(l2);
                jeo.add(t2);
                jeo.add(btn1);
                jeo.add(btn2);

                jeo.setSize(400,400);
                jeo.setLayout(new GridLayout(6,0));
                jeo.setVisible(true);
                jeo.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);



                btn1.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent ae){
                        //for logging into the user bank account
                        String accountId=t1.getText();
                        String pass=t2.getText();

                        ArrayList<Account> list=Account.readAllAccounts();
                        boolean flag=false;
                        for(Account i:list){
                            if(i.getAccountID().equals(accountId) &&
i.getCustomer().getRegister().getPassword().equals(pass)){
                                a=i;
                                flag=true;
                            }
                        }

                        if(flag){
                            otp="";
                            for(int i=0;i<4;i++){
                                int pass1=(int)(Math.random()*10);
                                otp+=pass1;
```

```
                            }

                            JOptionPane.showMessageDialog(null,"Your generated
OTP is "+otp);
//            System.out.println("Your generated otp: "+otp);

            //making new frame for otp validation
                            JFrame otpVerificarion=new JFrame("Account Login");

                            JButton b1=new JButton("Enter");

                            JLabel l1=new JLabel("OTP");
                            JTextField t1=new JTextField(20);

                            otpVerificarion.setSize(400,400);
                            otpVerificarion.setDefaultCloseOperation(JFrame.DISPO
SE_ON_CLOSE);//only closes the current frame
                            otpVerificarion.setVisible(true);
                            otpVerificarion.setLayout(new GridLayout(3,0));

                            otpVerificarion.add(l1);
                            otpVerificarion.add(t1);
                            otpVerificarion.add(b1);

                            b1.addActionListener(new ActionListener(){
                                public void actionPerformed(ActionEvent ae){
                                    String pas=t1.getText();

                                    if(pas.equals(otp)){

                                        Login.writeToLogin(login);
                                        JFrame created=new JFrame("User
Operations");
                        created.setSize(400,400);
                        created.setLayout(new GridLayout(9,0));
                        created.setVisible(true);
                        created.setDefaultCloseOperation(EXIT_ON_CLOSE);

                        JButton bro1=new JButton("Check Balance");
                        JButton bro2=new JButton("Withdraw Money");
                        JButton bro3=new JButton("Deposit Money");
                        JButton bro4=new JButton("Transfer Money");
                        JButton bro5=new JButton("Edit Account Details");
                        JButton bro6=new JButton("Transaction History");
                        JButton bro7=new JButton("Login History");
```

```
                        JButton bro8=new JButton("Delete an Account");
                        JButton bro9=new JButton("Log out");

                        created.add(bro1);
                        created.add(bro2);
                        created.add(bro3);
                        created.add(bro4);
                        created.add(bro5);
                        created.add(bro6);
                        created.add(bro7);
                        created.add(bro8);
                        created.add(bro9);



                        bro1.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){
                                JOptionPane.showMessageDialog(null,"Your current
balance is "+a.getBalance());
                            }
                        });

                        bro2.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Withdraw Money");
                                with.setSize(400,400);
                                with.setLayout(new GridLayout(3,0));
                                with.setVisible(true);
                                with.setDefaultCloseOperation(JFrame.DISPOSE_ON_C
LOSE);

                                JLabel lab1=new JLabel("Amount to be Withdrawn");
                                JTextField m=new JTextField(20);
                                JButton btn1=new JButton("Withdraw");

                                with.add(lab1);
                                with.add(m);
                                with.add(btn1);


                                btn1.addActionListener(new ActionListener(){
                                    public void actionPerformed(ActionEvent ae){
                                        double
cash=Integer.parseInt(m.getText());
```

```
                                        a.withdraw(cash);

                                    }
                                });
                            }
                        });

                        bro3.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Deposit Money");
                                with.setSize(400,400);
                                with.setLayout(new GridLayout(3,0));
                                with.setVisible(true);
                                with.setDefaultCloseOperation(JFrame.DISPOSE_ON_C
LOSE);

                                JLabel lab1=new JLabel("Amount to be Deposited");
                                JTextField m=new JTextField(20);
                                JButton btn1=new JButton("Deposit");

                                with.add(lab1);
                                with.add(m);
                                with.add(btn1);


                                btn1.addActionListener(new ActionListener(){
                                    public void actionPerformed(ActionEvent ae){
                                        double
cash=Integer.parseInt(m.getText());
                                        a.deposit(cash);

                                    }
                                });
                            }
                        });


                        bro4.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Tranfer Money");
                                with.setSize(400,400);
                                with.setLayout(new GridLayout(5,0));
                                with.setVisible(true);
```

```
                                with.setDefaultCloseOperation(JFrame.DISPOSE_ON_C
LOSE);

                                JLabel lab1=new JLabel("Amount to be
Transfered");
                                JTextField m=new JTextField(20);
                                JLabel lab2=new JLabel("Account ID of receiver
Account");
                                JTextField m1=new JTextField(20);
                                JButton btn1=new JButton("Transfer");

                                with.add(lab1);
                                with.add(m);
                                with.add(lab2);
                                with.add(m1);
                                with.add(btn1);


                                btn1.addActionListener(new ActionListener(){
                                    public void actionPerformed(ActionEvent ae){
                                        double
cash=Integer.parseInt(m.getText());
                                        String id=m1.getText();

                                        a.transfer(cash, id);
                                    }
                                });
                            }
                        });


                        bro5.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Edit Account Details");
                                with.setSize(400,400);
                                with.setLayout(new GridLayout(11,0));
                                with.setVisible(true);
                                with.setDefaultCloseOperation(JFrame.DISPOSE_ON_C
LOSE);

                                JLabel lab1=new JLabel("Account Number");
                                JTextField m=new JTextField(20);
                                JLabel lab2=new JLabel("Password");
                                JTextField m1=new JTextField(20);
```

```
                                JLabel lab3=new JLabel("New Email");
                                JTextField m2=new JTextField(20);
                                JLabel lab4=new JLabel("New Phone");
                                JTextField m3=new JTextField(20);
                                JLabel lab5=new JLabel("New Address");
                                JTextField m4=new JTextField(20);

                                JButton btn1=new JButton("Update");

                                with.add(lab1);
                                with.add(m);
                                with.add(lab2);
                                with.add(m1);
                                with.add(lab3);
                                with.add(m2);
                                with.add(lab4);
                                with.add(m3);
                                with.add(lab5);
                                with.add(m4);
                                with.add(btn1);


                                btn1.addActionListener(new ActionListener(){
                                    public void actionPerformed(ActionEvent ae){
                                        String accountNum=m.getText();
                                        String pass=m1.getText();
                                        String email=m2.getText();
                                        long
phone=Integer.parseInt(m3.getText());
                                        String address=m4.getText();
                                        double amount=0;
                                        Account.updateAnAccount(accountNum,pass,e
mail,phone,address,amount);
                                        JOptionPane.showMessageDialog(null,"Accou
nt updated Successfully");
                                    }
                                });
                            }
                        });


                        bro6.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Transaction History");
```

```
                                StringBuilder history=new StringBuilder();
                                ArrayList<AccountTransaction>
list=AccountTransaction.readAllAccountTransactions();
                                for(AccountTransaction i:list){
                                    if(i.getAccount().getAccountID().equals(a.get
AccountID())){
                                        history.append(i).append("\n");
                                    }
                                }
                                if(history.isEmpty()){
                                    history.append("No transactions made yet");
                                }

                                with.setSize(400, 300);
                                JTextArea textArea = new
JTextArea(history.toString());
                                textArea.setEditable(false);
                                with.add(new JScrollPane(textArea));
                                with.setVisible(true);

                            }
                        });


                        bro7.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){

                                JFrame with=new JFrame("Login History");

                                StringBuilder history=new StringBuilder();
                                ArrayList<LoginTime> list=Login.readAllLogins();
                                for(LoginTime i:list){
                                    if(i.getLogin().getAccountNumber().equals(log
in.getAccountNumber())
                                        &&
i.getLogin().getPassword().equals(login.getPassword())){
                                        history.append(i).append("\n");
                                    }
                                }
                                if(history.isEmpty()){
                                    history.append("No logins yet");
                                }

                                with.setSize(400, 300);
```

```
                                JTextArea textArea = new
JTextArea(history.toString());
                                textArea.setEditable(false);
                                with.add(new JScrollPane(textArea));
                                with.setVisible(true);

                            }
                        });


                        bro8.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){


                                otp="";
                                for(int i=0;i<4;i++){
                                    int pass=(int)(Math.random()*10);
                                    otp+=pass;
                                }

                                JOptionPane.showMessageDialog(null,"Your
generated OTP is "+otp);
//            System.out.println("Your generated otp: "+otp);

            //making new frame for otp validation
                                JFrame otpVerificarion=new JFrame("Account
Deletion");

                                JButton b1=new JButton("Enter");

                                JLabel l1=new JLabel("OTP");
                                JTextField t1=new JTextField(20);

                                otpVerificarion.setSize(400,400);
                                otpVerificarion.setDefaultCloseOperation(JFrame.D
ISPOSE_ON_CLOSE);//only closes the current frame
                                otpVerificarion.setVisible(true);
                                otpVerificarion.setLayout(new GridLayout(3,0));

                                otpVerificarion.add(l1);
                                otpVerificarion.add(t1);
                                otpVerificarion.add(b1);

                                b1.addActionListener(new ActionListener(){
                                    public void actionPerformed(ActionEvent ae){
```

```
                                        String pas=t1.getText();

                                        if(pas.equals(otp)){

                                            Account.deleteAnAccount(a.getAccountI
D(),a.getAccountType());
                                        }
                                        else{
                                            JOptionPane.showMessageDialog(null,"U
nable to login.\nWrong OTP");
                                        }

                                    }
                                });

                            }
                        });


                        bro9.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent ae){
                                JOptionPane.showMessageDialog(null,"Loged out
successfully");
                                UserGUI h=new UserGUI();
                            }
                        });


                    }
                                    else{
                                        JOptionPane.showMessageDialog(null,"Wrong
OTP\nUnable to login`");
                                    }
                                }
                            });

                        }
                        else{
                            JOptionPane.showMessageDialog(null,"Wrong
credentilas\nUnable to login.");
                        }
                    }

                });
```

```
                btn2.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent ae){
                        UserOperationsGUI g=new
UserOperationsGUI(customer,login);
                    }

                });

            }
        });


        b3.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent ae){
                UserGUI g=new UserGUI();
            }
        });


    }
}
```

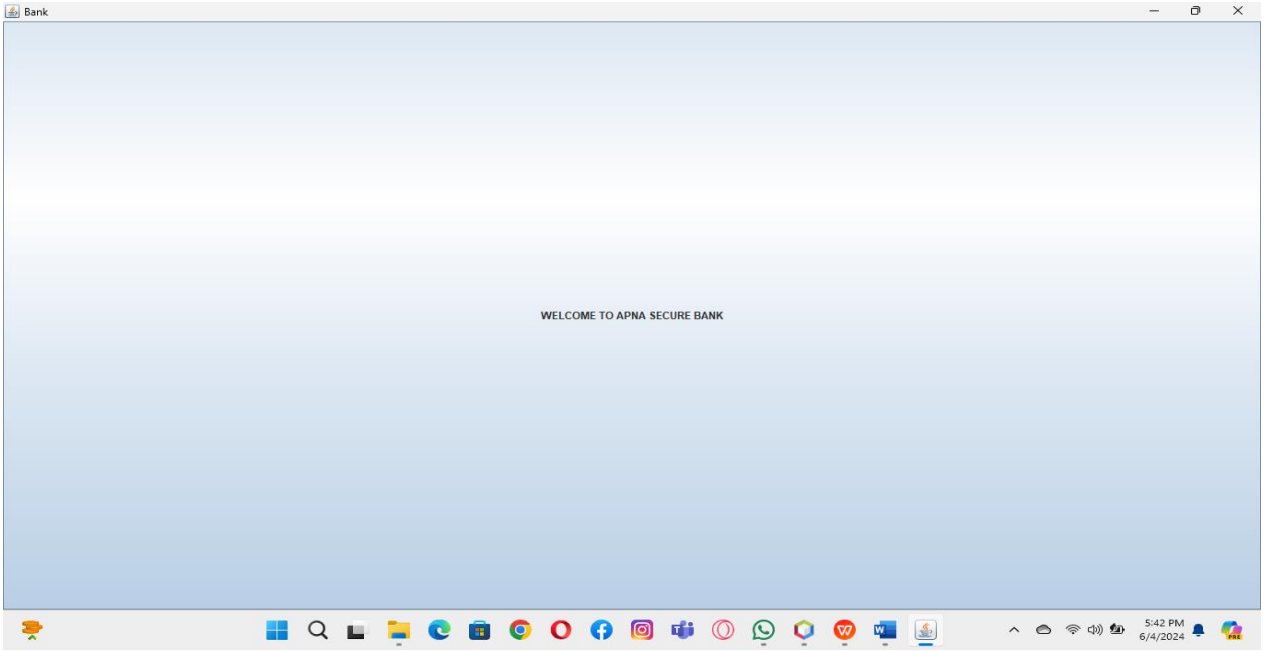
Main GUI:

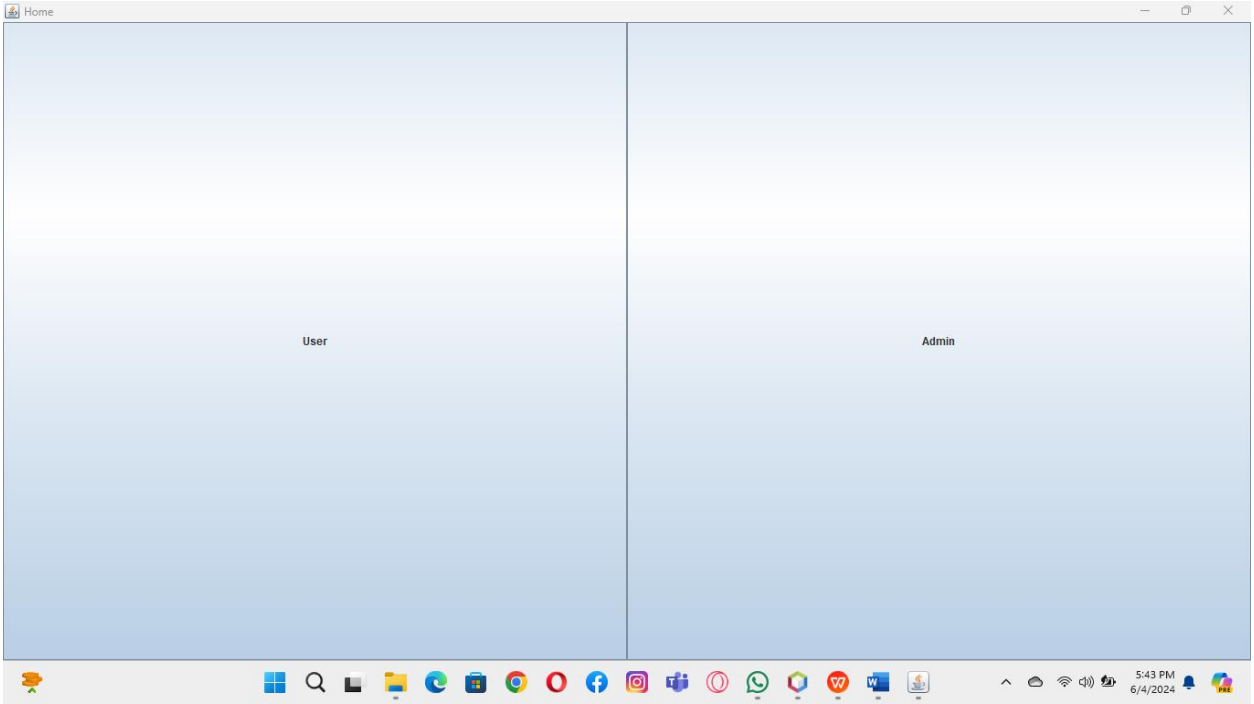
```
package bankmanagementsystem;

import java.util.*;
import java.awt.*;

public class BankManagementSystem {

    public static void main(String[] args) {

            BankGUI g=new BankGUI();

    }
}
```

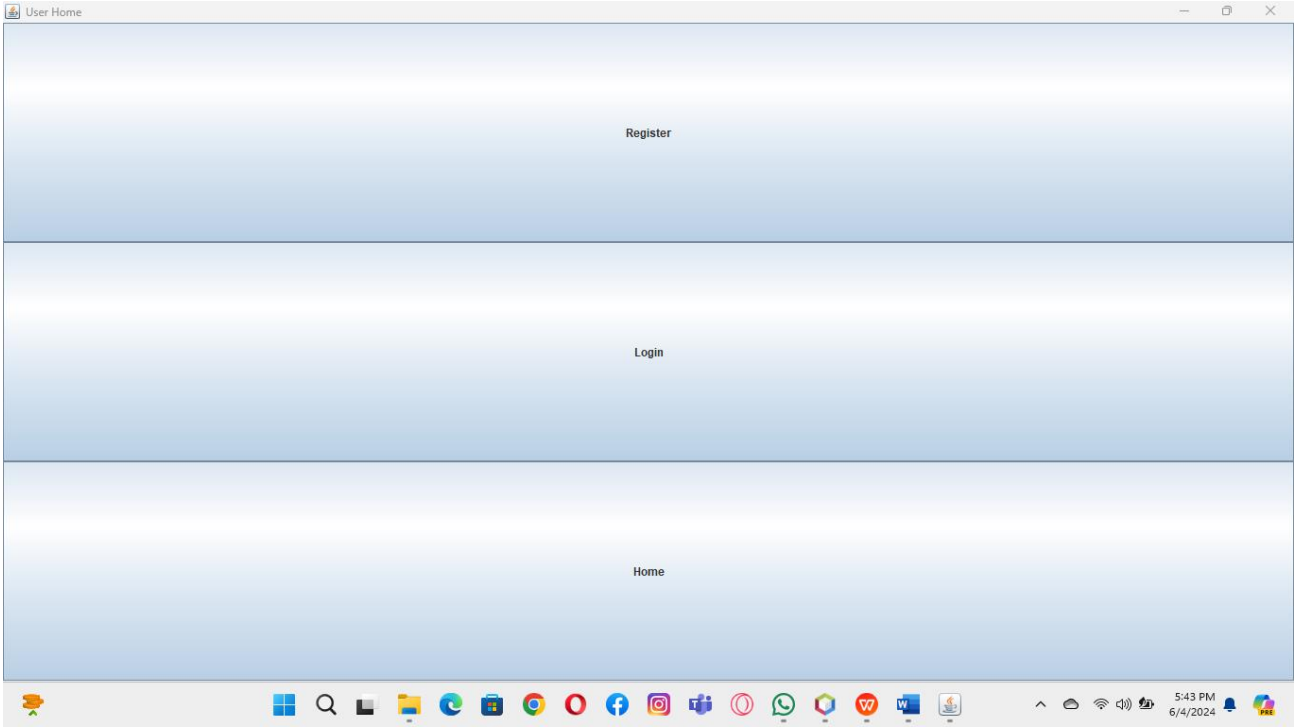
My action listener:

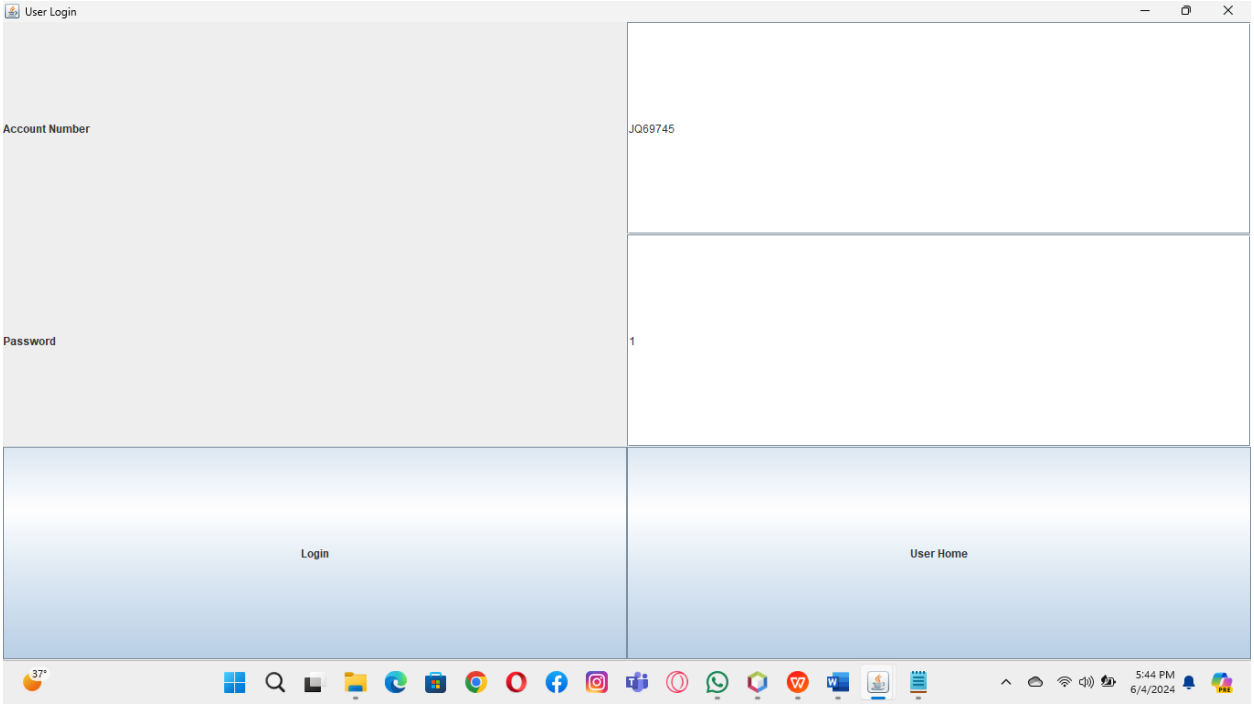
```java
package bankmanagementsystem;

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;


public class MyActionListener implements ActionListener{

        @Override
        public void actionPerformed(ActionEvent ae){

            if(ae.getActionCommand().equals("User")){
                UserGUI q=new UserGUI();

            }

            else if(ae.getActionCommand().equals("Admin")){
                ManagerGUI m=new ManagerGUI();
            }



    }
}
```

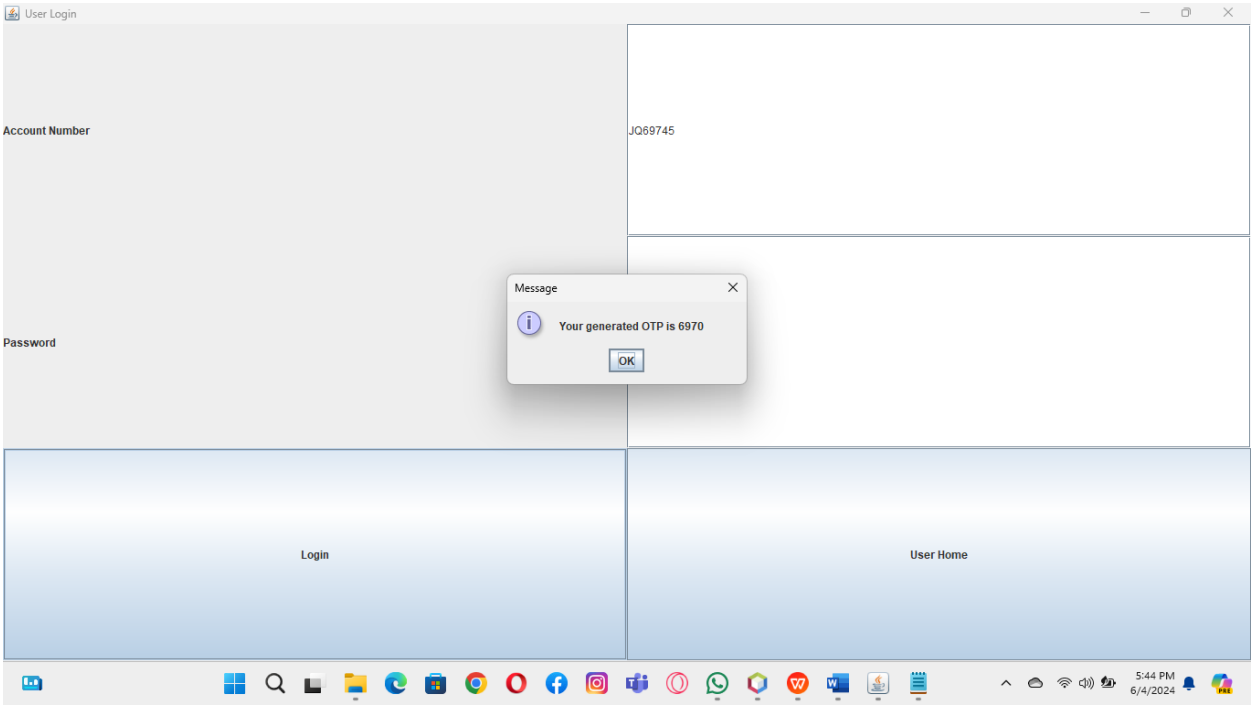
GUI:

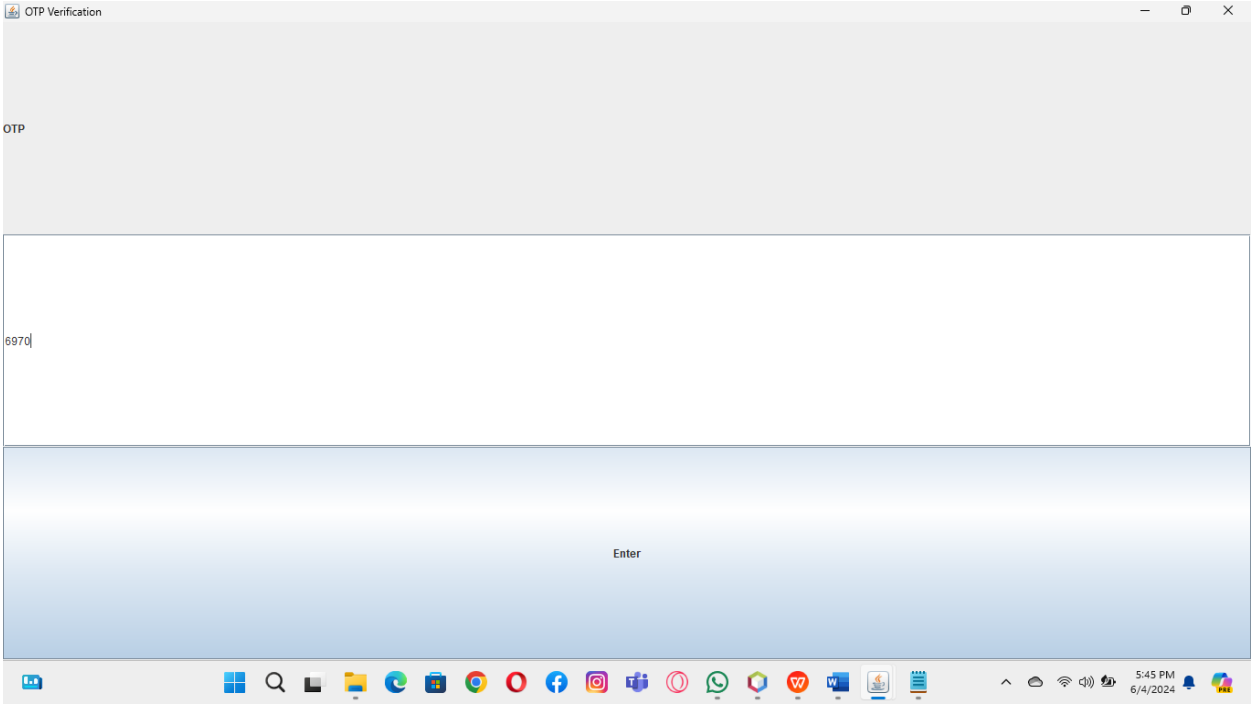


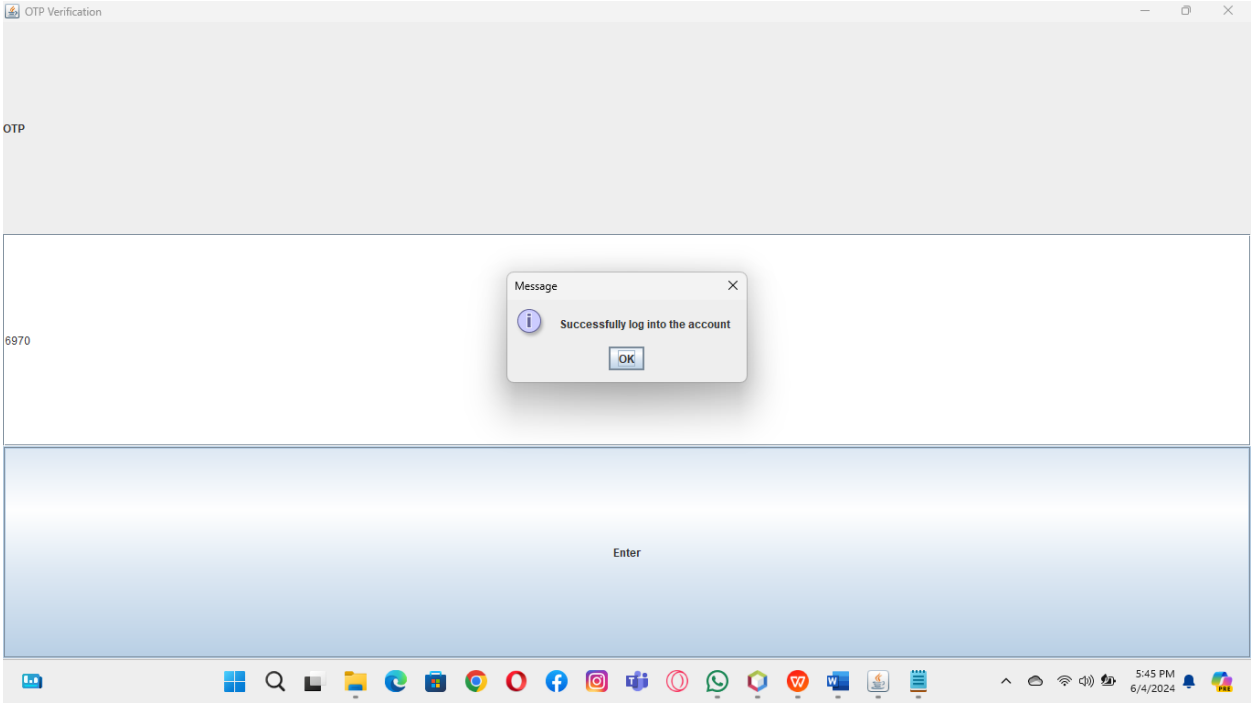

Home
User
Admin
5:43 PM
6/4/2024

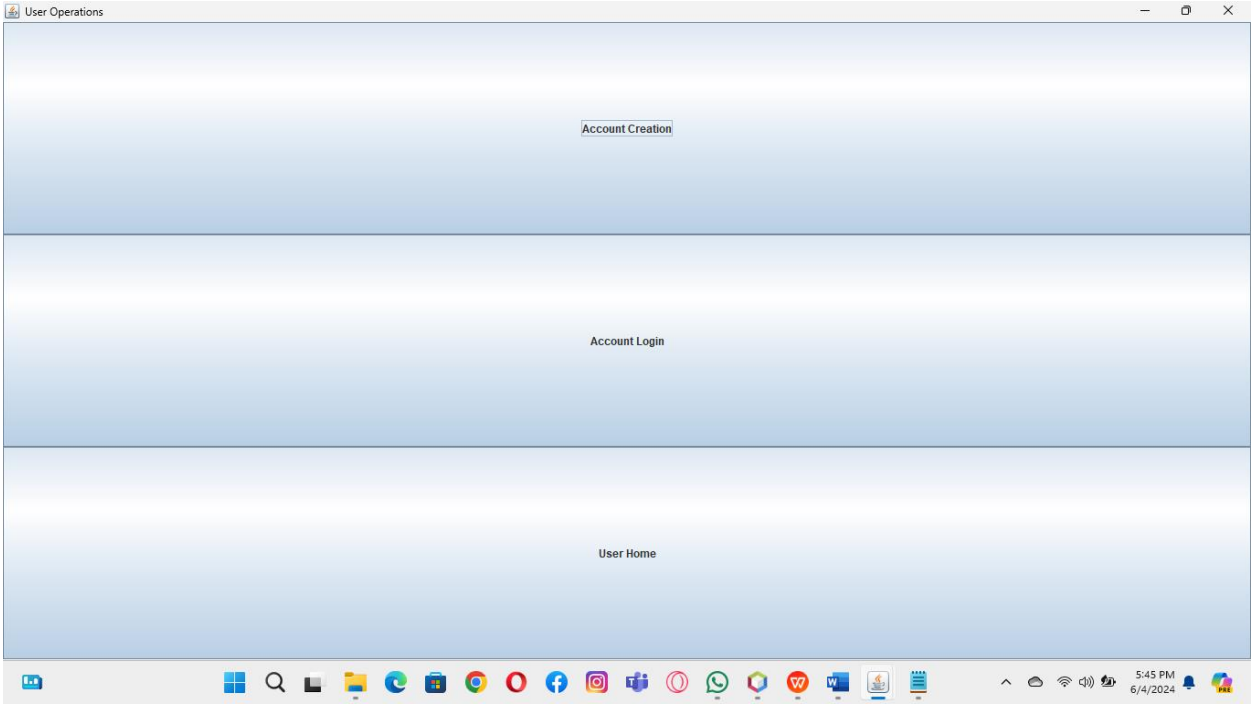

User Home
Register
Login
Home
5:43 PM
6/4/2024

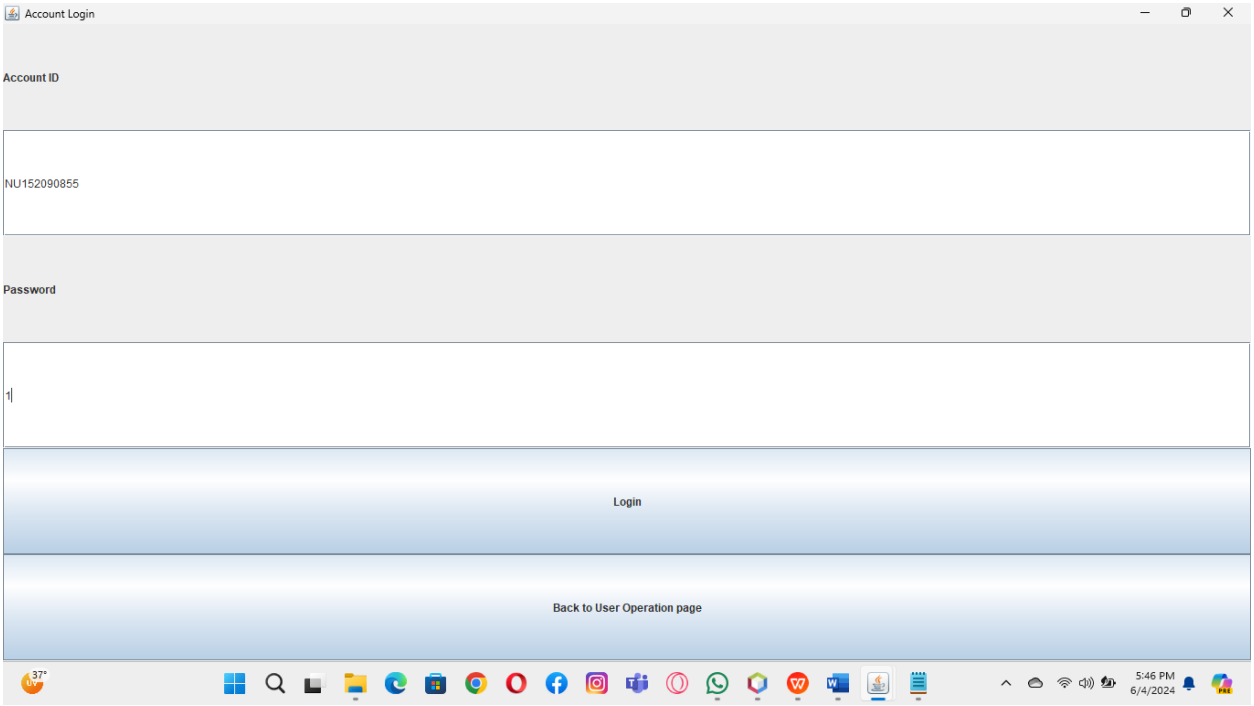

User Login
Account Number
JQ69745
Password
1
Login
User Home
5:44 PM
6/4/2024

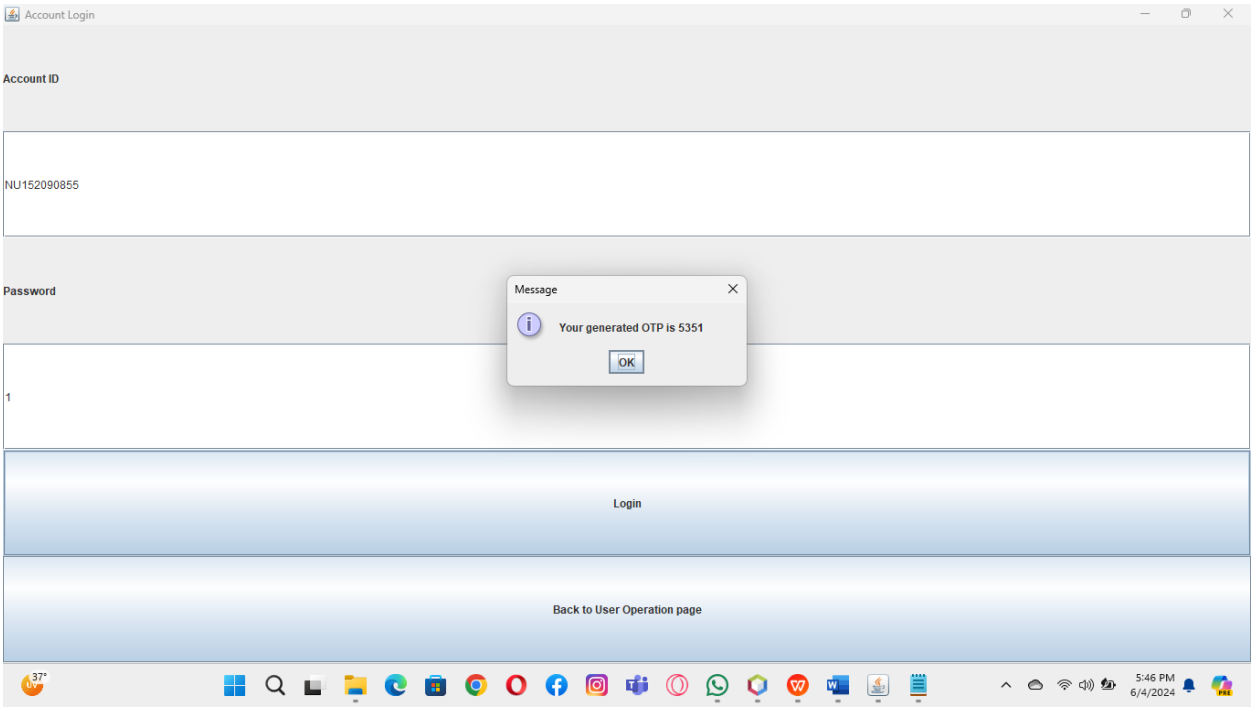

User Login
Account Number
JQ69745
Password
Message
Your generated OTP is 6970
OK
Login
User Home
5:44 PM
6/4/2024

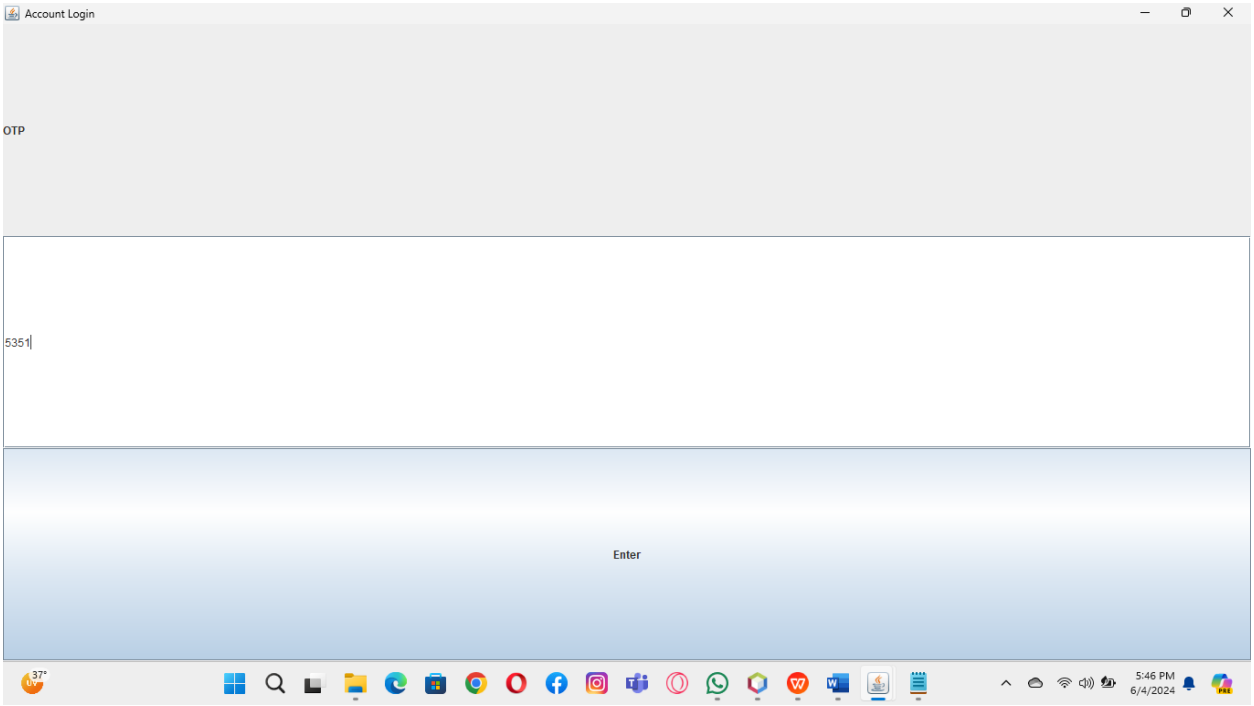

OTP Verification
OTP
6970
Enter
5:45 PM
6/4/2024

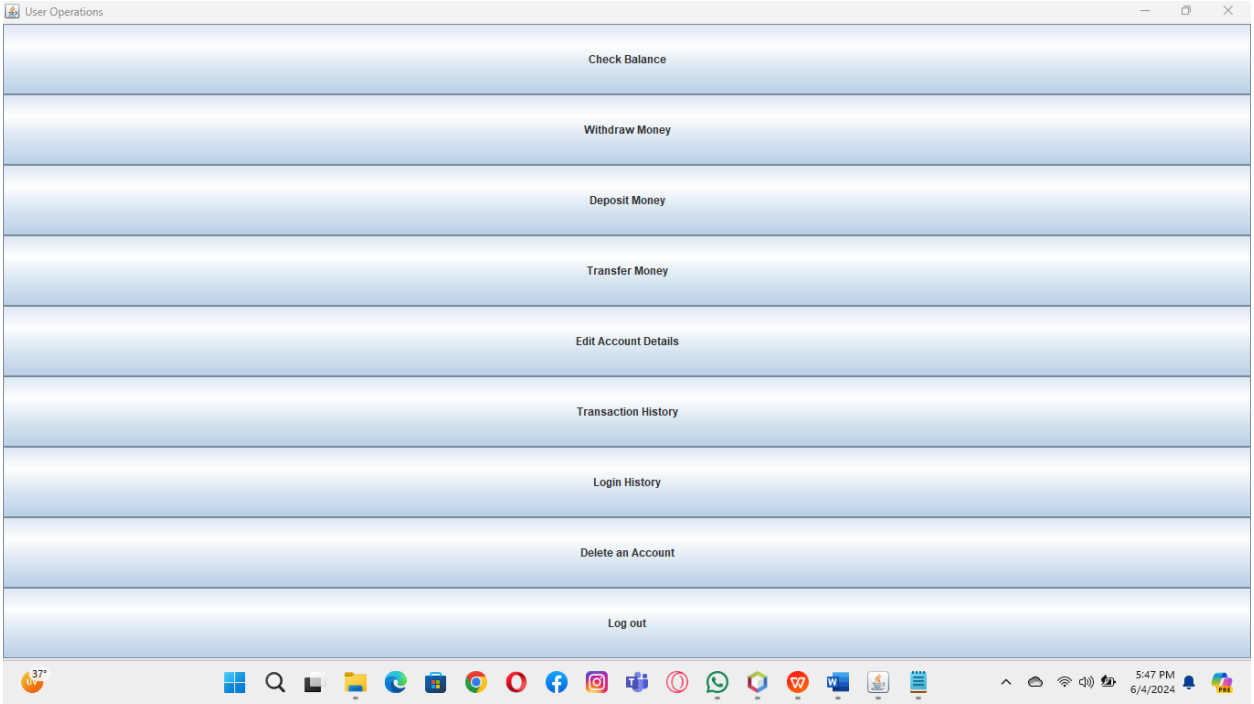

OTP Verification
OTP
6970
Message
Successfully log into the account
OK
Enter
5:45 PM
6/4/2024

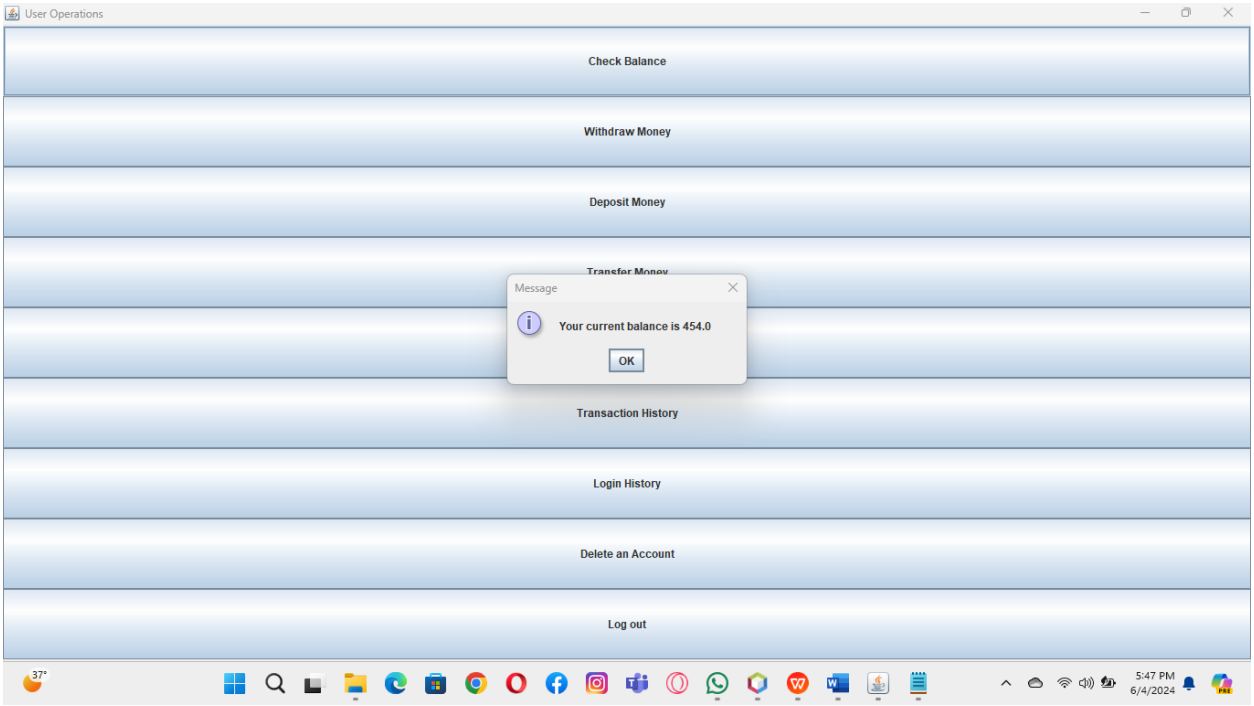

User Operations
Account Creation
Account Login
User Home
5:45 PM
6/4/2024

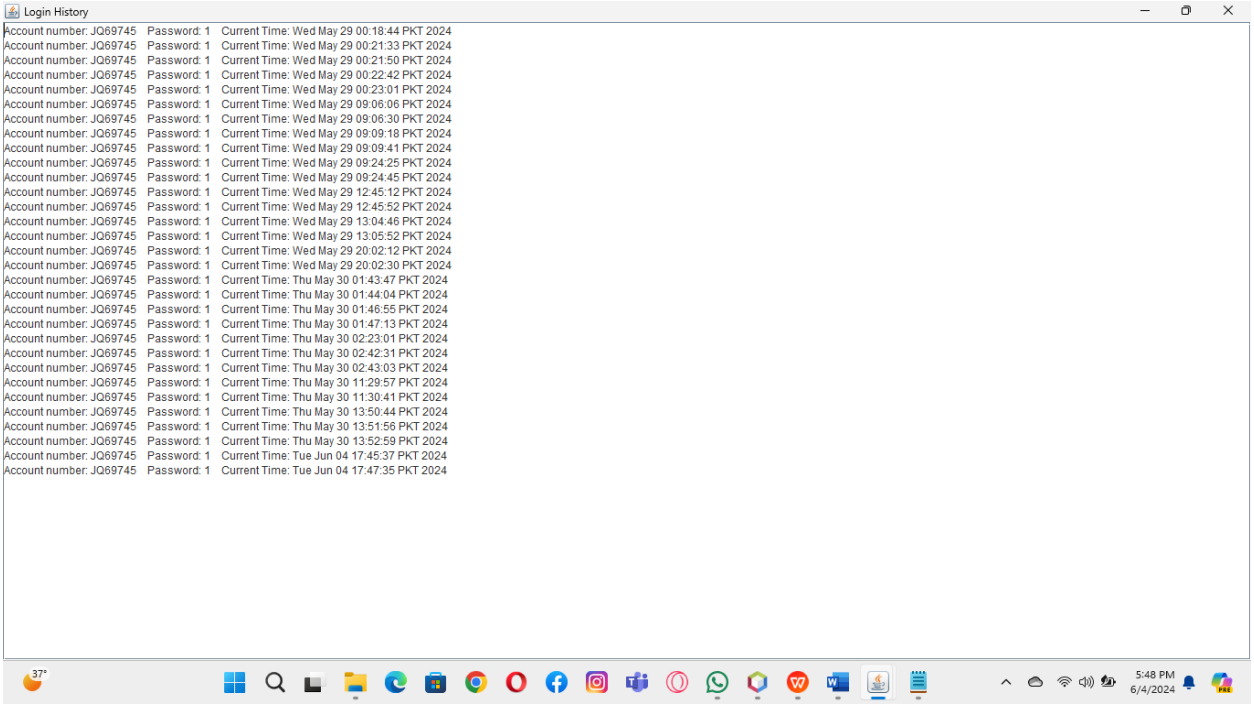

Account Login
Account ID
NU152090855
Password
1
Login
Back to User Operation page
5:46 PM
6/4/2024

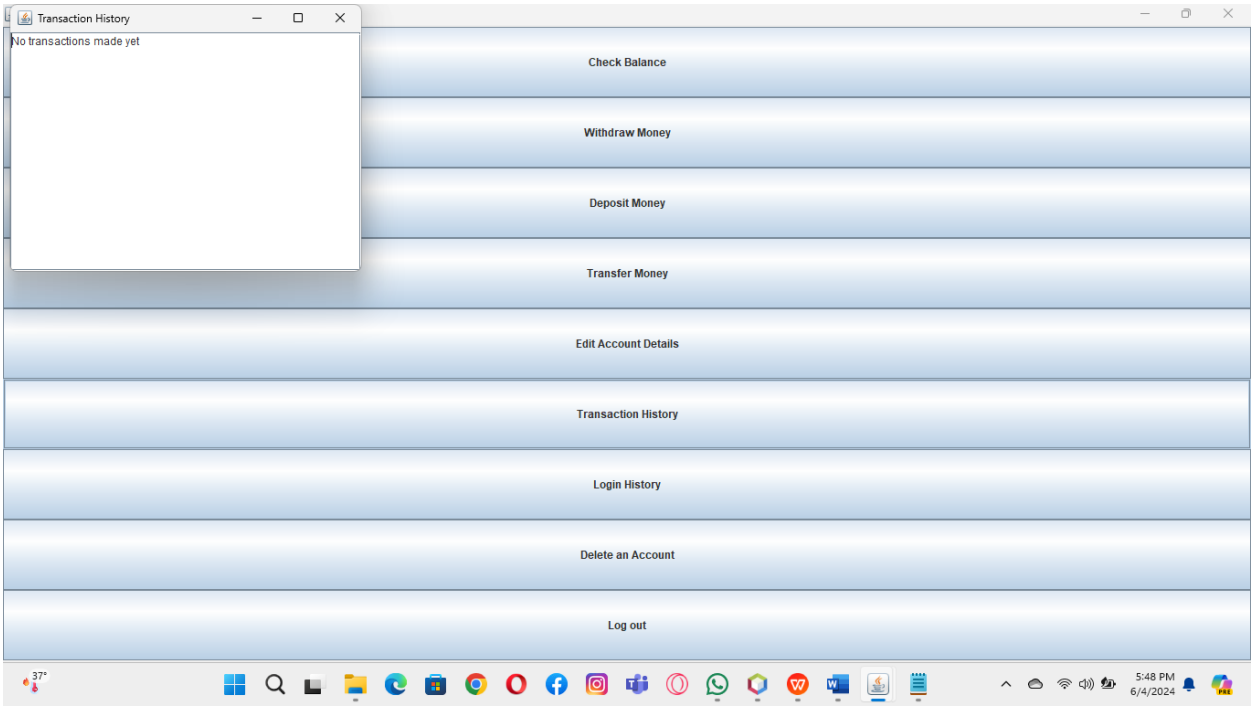

Account Login
Account ID
NU152090855
Password
1
Message
Your generated OTP is 5351
OK
Login
Back to User Operation page
5:46 PM
6/4/2024

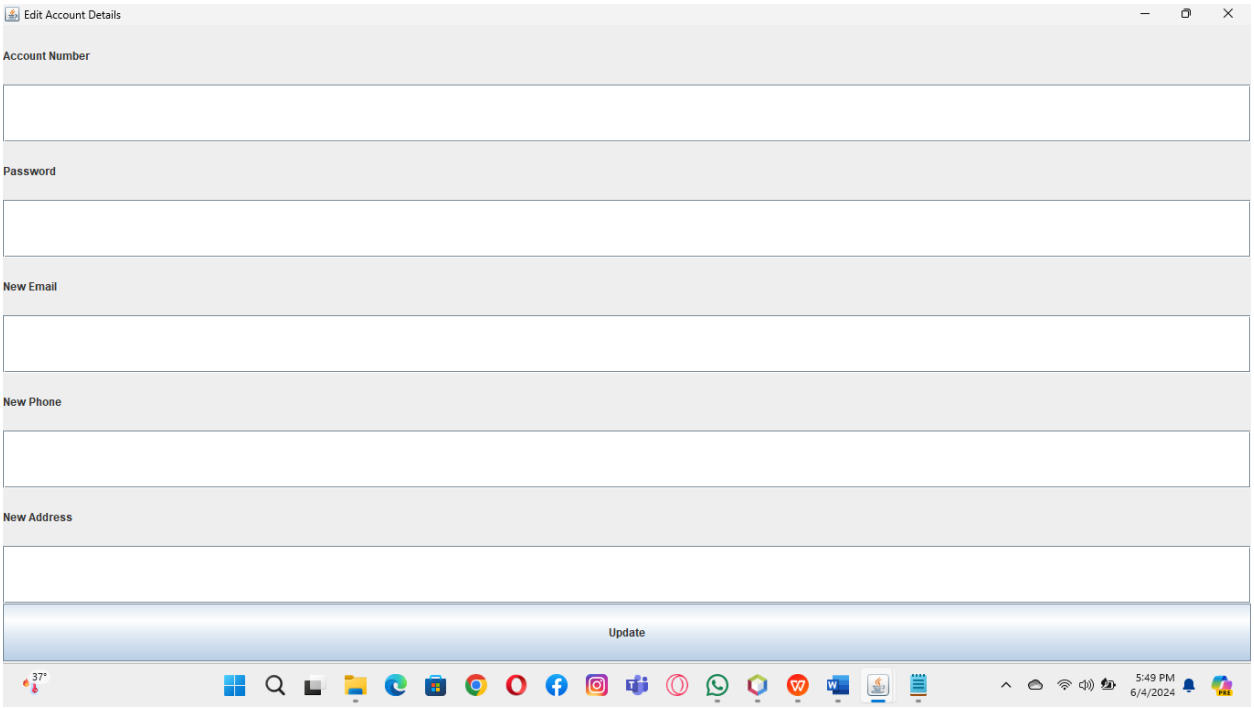

Account Login
OTP
5351
Enter
5:46 PM
6/4/2024

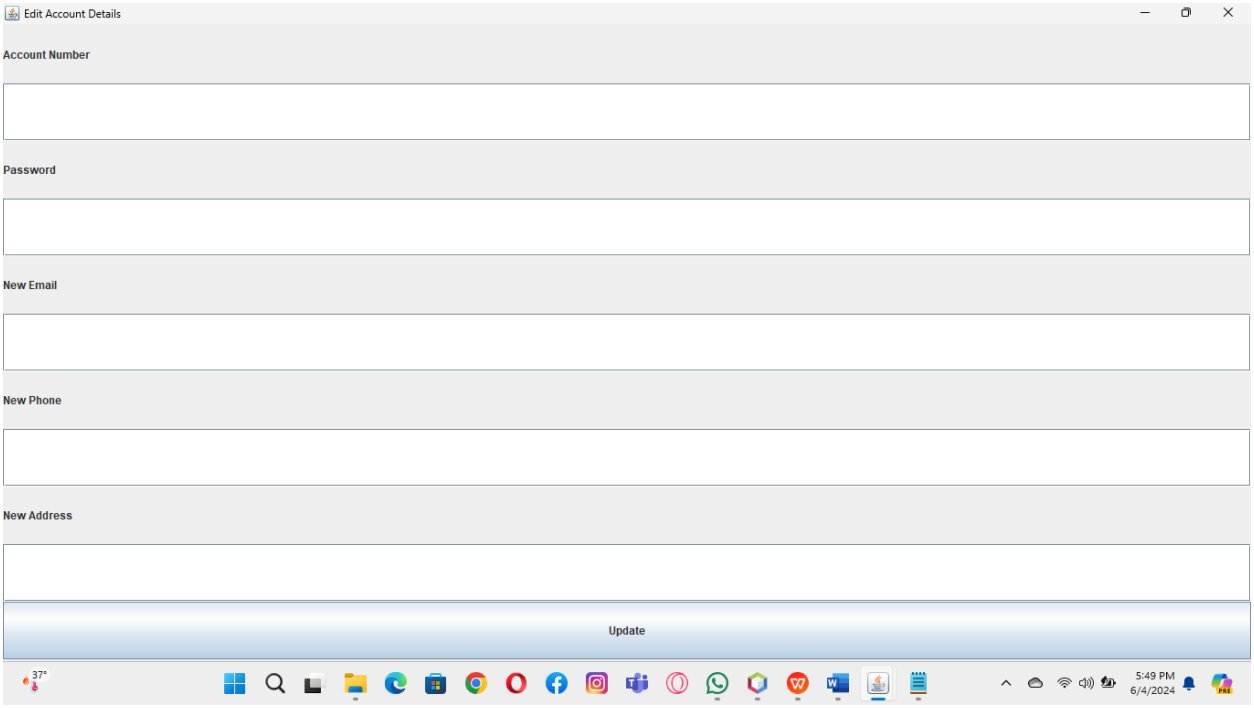

User Operations
Check Balance
Withdraw Money
Deposit Money
Transfer Money
Edit Account Details
Transaction History
Login History
Delete an Account
Log out
5:47 PM
6/4/2024

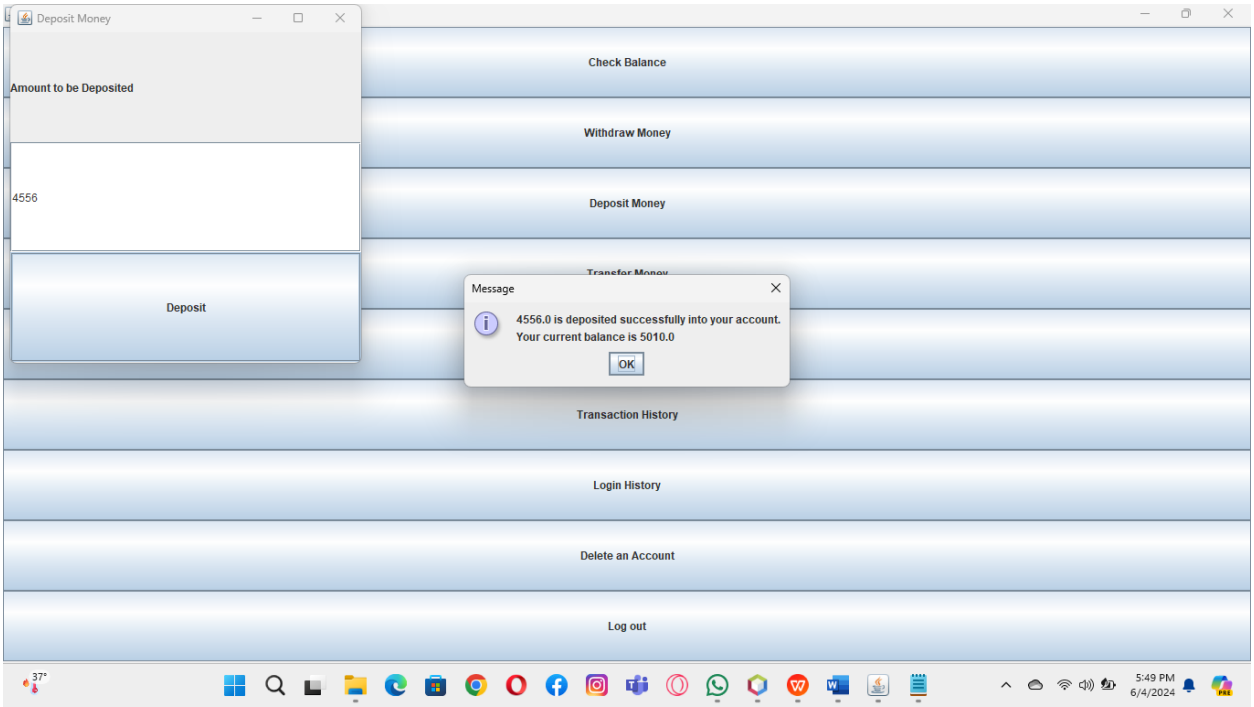

User Operations
Check Balance
Withdraw Money
Deposit Money
Transfer Money
Message
Your current balance is 454.0
OK
Transaction History
Login History
Delete an Account
Log out
5:47 PM
6/4/2024

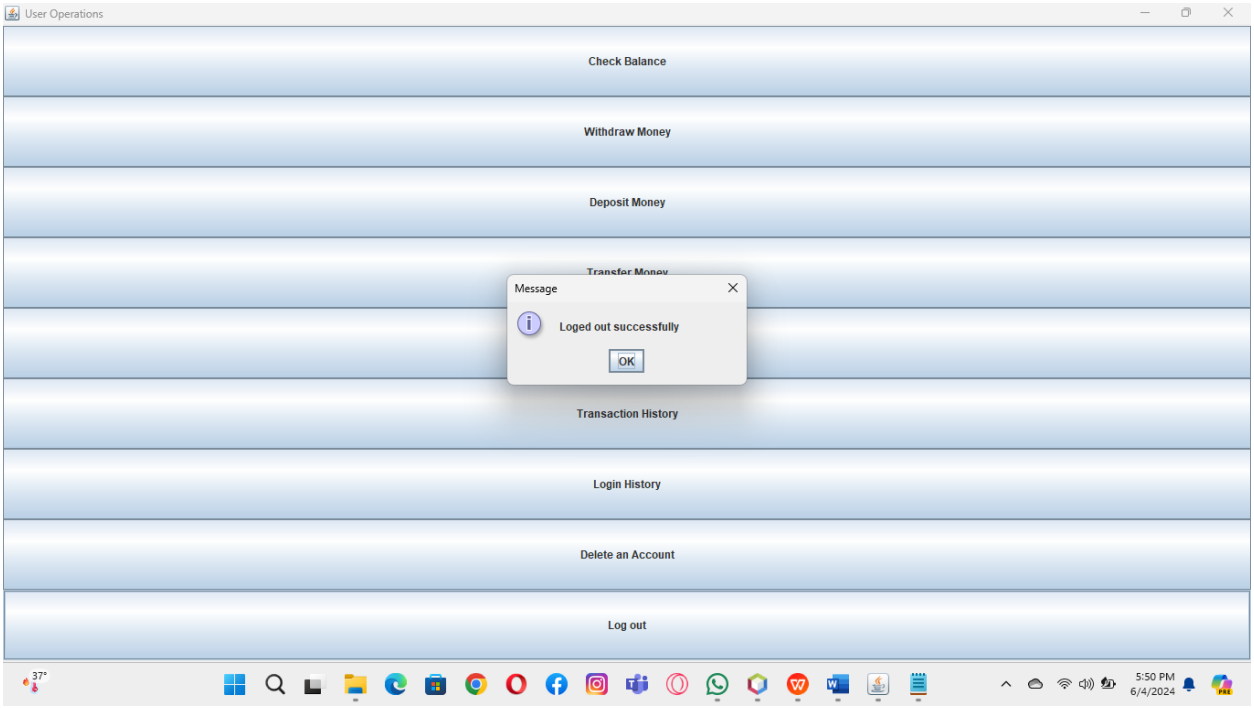

Login History
Account number: JQ69745 Password: 1 Current Time: Wed May 29 00:18:44 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 00:21:33 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 00:21:50 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 00:22:42 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 00:23:01 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 09:06:06 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 09:06:30 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 09:09:18 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 09:09:41 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 09:24:25 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 09:24:45 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 12:45:12 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 12:45:52 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 13:04:46 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 13:05:52 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 20:02:12 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Wed May 29 20:02:30 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 01:43:47 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 01:44:04 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 01:46:55 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 01:47:13 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 02:23:01 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 02:42:31 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 02:43:03 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 11:29:57 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 11:30:41 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 13:50:44 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 13:51:56 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Thu May 30 13:52:59 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Tue Jun 04 17:45:37 PKT 2024
Account number: JQ69745 Password: 1 Current Time: Tue Jun 04 17:47:35 PKT 2024
5:48 PM
6/4/2024

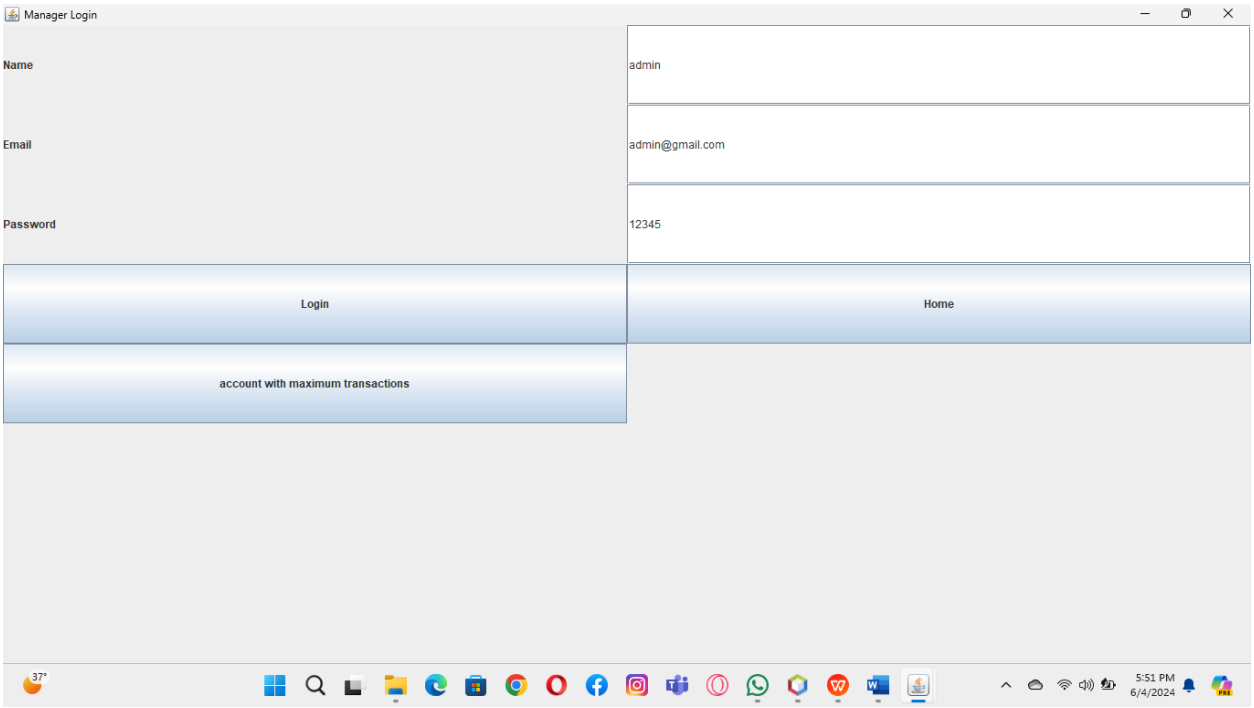

Transaction History
No transactions made yet
Check Balance
Withdraw Money
Deposit Money
Transfer Money
Edit Account Details
Transaction History
Login History
Delete an Account
Log out
5:48 PM
6/4/2024

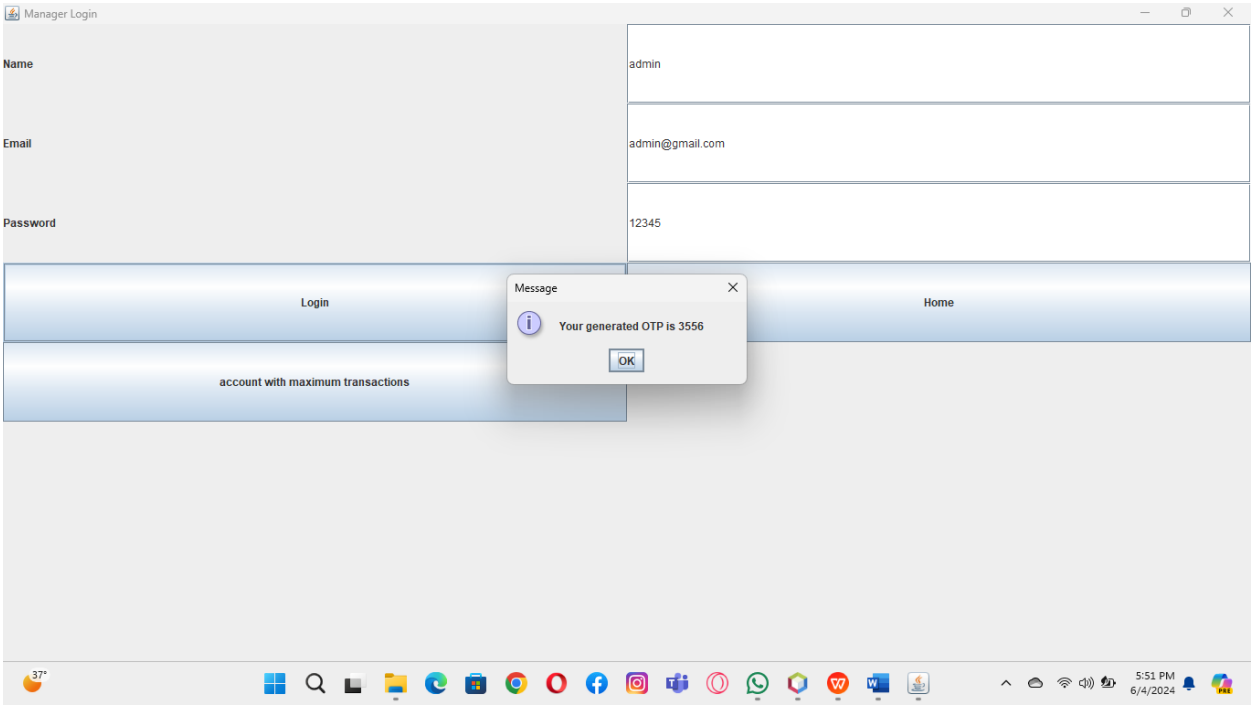

Edit Account Details
Account Number
Password
New Email
New Phone
New Address
Update
37°
5:49 PM
6/4/2024

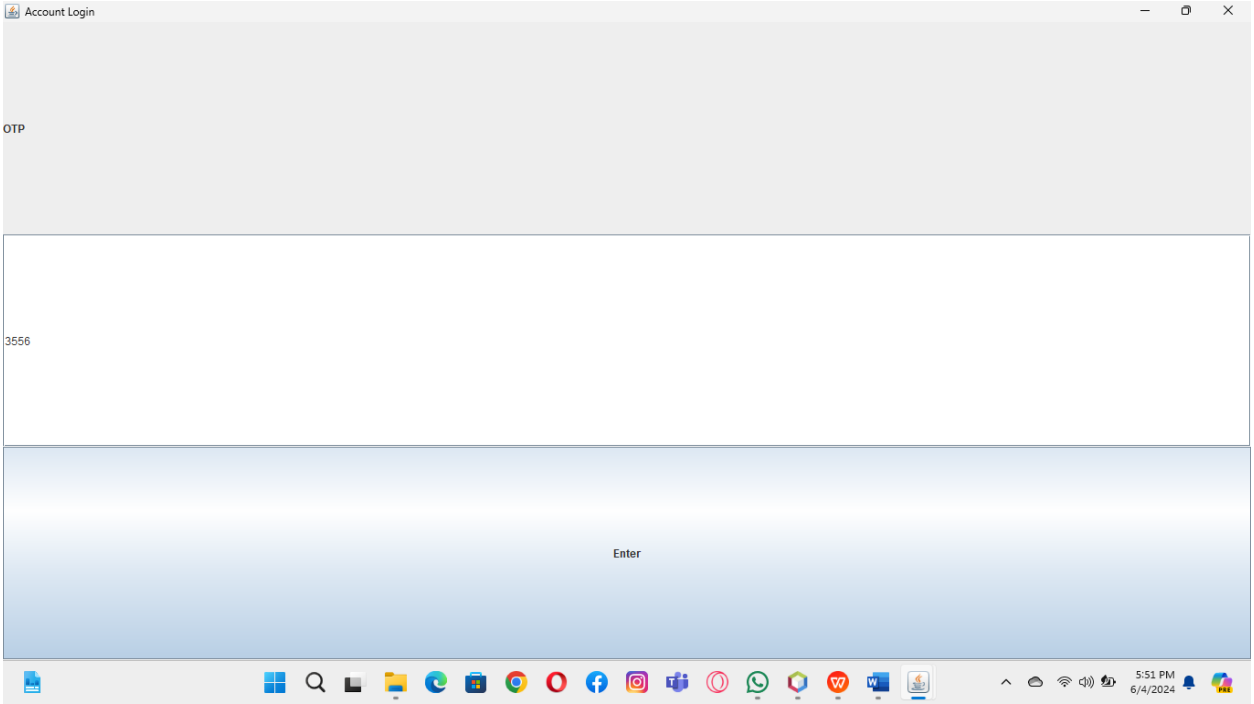

Edit Account Details
Account Number
Password
New Email
New Phone
New Address
Update
37°
5:49 PM
6/4/2024

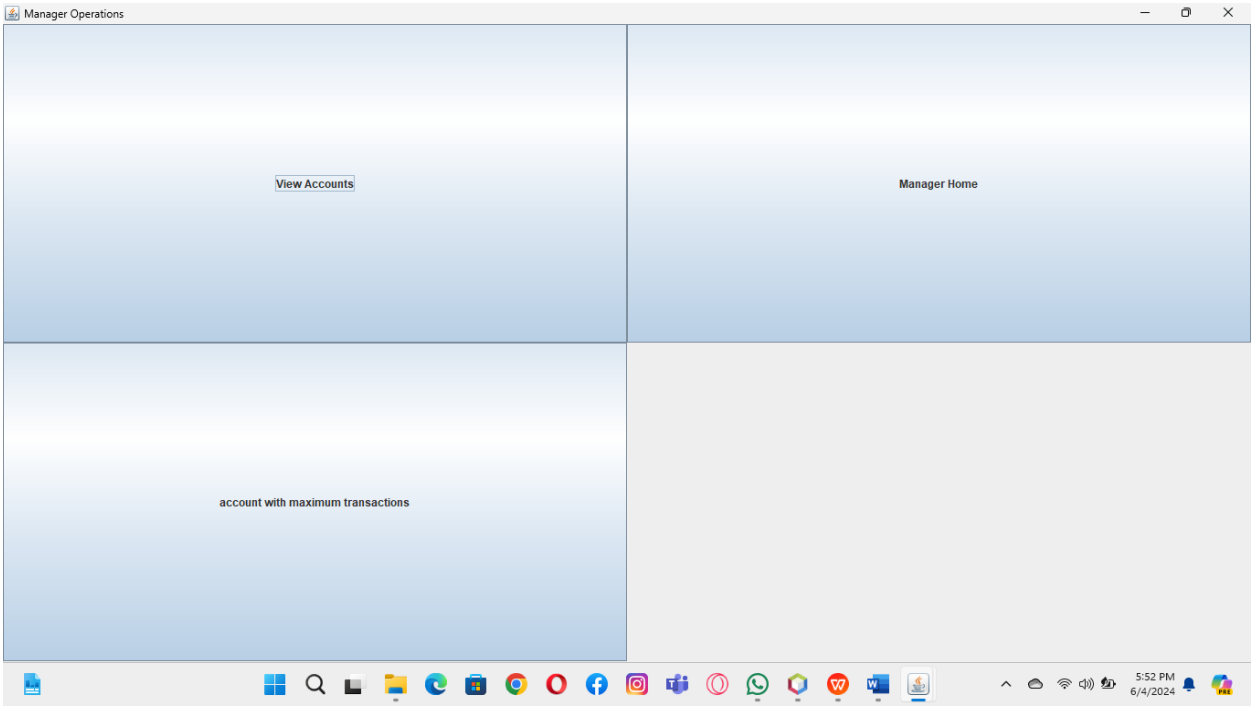

Deposit Money
Amount to be Deposited
4556
Deposit
Check Balance
Withdraw Money
Deposit Money
Transfer Money
Message
4556.0 is deposited successfully into your account.
Your current balance is 5010.0
OK
Transaction History
Login History
Delete an Account
Log out
5:49 PM
6/4/2024

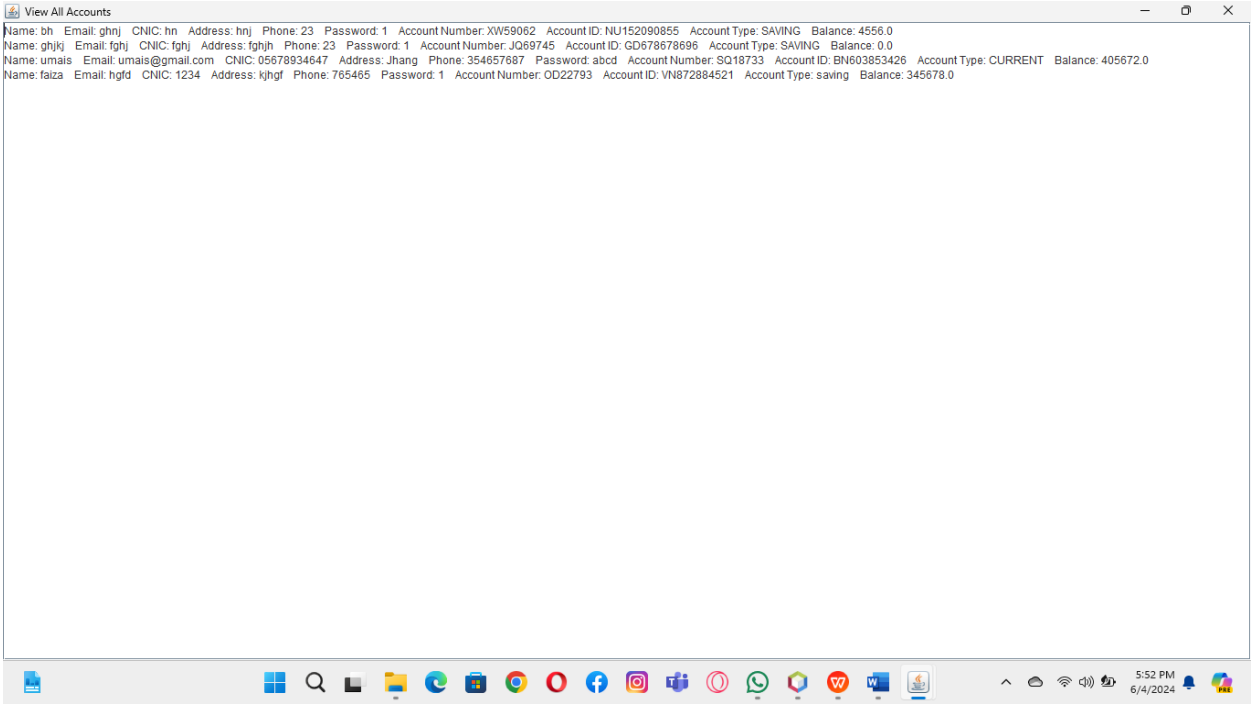

User Operations
Check Balance
Withdraw Money
Deposit Money
Transfer Money
Message
Loged out successfully
OK
Transaction History
Login History
Delete an Account
Log out
5:50 PM
6/4/2024


Manager Login
Name
admin
Email
admin@gmail.com
Password
12345
Login
Home
account with maximum transactions
5:51 PM
6/4/2024


Manager Login
Name
admin
Email
admin@gmail.com
Password
12345
Login
Home
account with maximum transactions
Message
Your generated OTP is 3556
OK
5:51 PM
6/4/2024


Account Login
OTP
3556
Enter
5:51 PM
6/4/2024


Manager Operations
View Accounts
Manager Home
account with maximum transactions
5:52 PM
6/4/2024


View All Accounts
Name: bh Email: ghnj CNIC: hn Address: hnj Phone: 23 Password: 1 Account Number: XW59062 Account ID: NU152090855 Account Type: SAVING Balance: 4556.0
Name: ghjkj Email: fghj CNIC: fghj Address: fghjh Phone: 23 Password: 1 Account Number: JQ69745 Account ID: GD678678696 Account Type: SAVING Balance: 0.0
Name: umais Email: umais@gmail.com CNIC: 05678934647 Address: Jhang Phone: 354657687 Password: abcd Account Number: SQ18733 Account ID: BN603853426 Account Type: CURRENT Balance: 405672.0
Name: faiza Email: hgfd CNIC: 1234 Address: kjhgf Phone: 765465 Password: 1 Account Number: OD22793 Account ID: VN872884521 Account Type: saving Balance: 345678.0
5:52 PM
6/4/2024

XX----------XX