Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)
A CHRISTIAN MINORITY RESIDENTIAL INSTITUTION
AICTE Approved & NAAC Accredited

A SKILL BASED EVALUATION REPORT

SUBMITTED BY

J JENOLIN JEBA (URK22CS5026)

COURSE CODE
20CS2035

COURSE NAME

OBJECT ORIENTED PROGRAMMING

OCTOBER 2023

DIVISION OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

INDUSTRIAL CERTIFICATION

Infosys
Navigate your next

COURSE COMPLETION CERTIFICATE

The certificate is awarded to

J Jenolin Jeba

for successfully completing the course

Java Programming Fundamentals

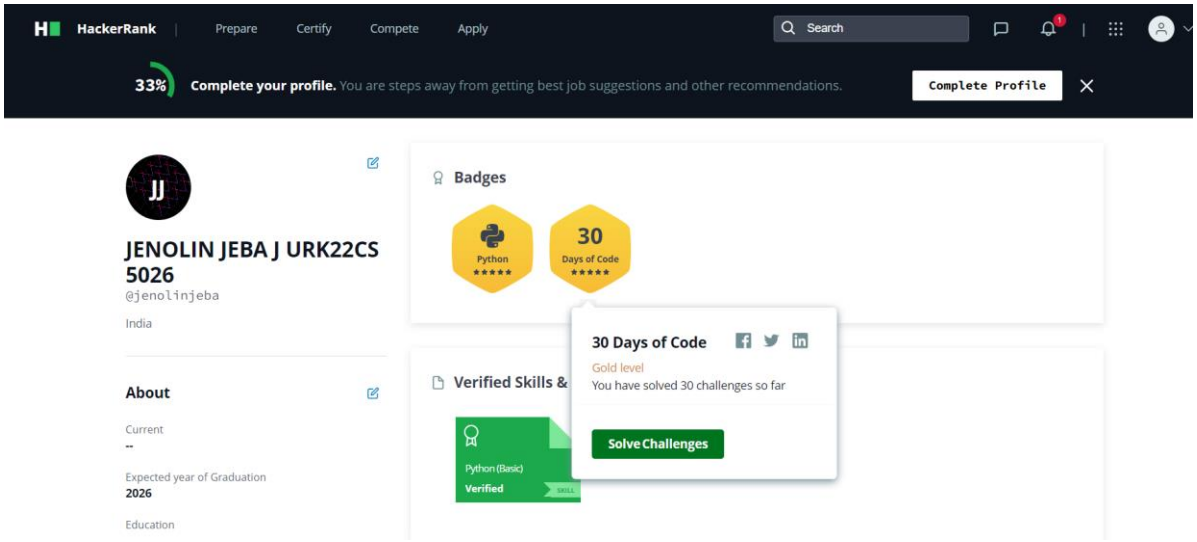
on August 25, 2023

Infosys | Springboard

Congratulations! You make us proud!

Thirumala Arohi
Senior Vice President and Head
Education, Training and Assessment (ETA)
Infosys Limited

Issued on: Friday, August 25, 2023
To verify, scan the QR code at https://verify.onwingspan.com



BANK MANAGEMNET SYSTEM

A REAL TIME APPLICATION REPORT

Submitted by

J JENOLIN JEBA (URK22CS5026)

DIVISION OF COMPUTER SCIENCE AND ENGINEERING

KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed-to-be-under Sec-3 of the UGC Act, 1956) Karunya Nagar, Coimbatore - 641 114. INDIA

OCTOBER 2023

ABSTRACT

The Bank Management System is a comprehensive software application designed to streamline and enhance the operations of a modern banking institution. In today's fast-paced financial environment, efficient and secure management of customer accounts, transactions, and services is paramount. Every time we need to make use of the bank application bank account is used by the people which may be found quite difficult at times.

Our Bank application starts with registering a user who can deposit money, view their balance, make transfer amounts within our bank as well as to other banks, provides the facility to recharge as well as top up to mobile phones with their appropriate phone numbers. Then the user can login through their username and password to deposit, view balance, recharge and make transfers.

The data given by the user while registering is stored in a database in the form of table and the money is incremented or decremented according to the user’s transferal of money. Even if the user forgets the password, after three attempts a forget password appears in which the password can be reset.

In today's rapidly evolving financial landscape, the efficient and secure management of customer accounts, transactions, and services is of paramount importance. This system simplifies the complex processes involved in banking, making it user-friendly and easy to manage

PROBLEM STATEMENT

In this fast-moving world, people find it hard to go to bank or ATMs instead we look forward to completing everything in a simpler way. By the context people find it more difficult to create an account in the bank with a lot of verification process. Traditional methods of creating bank accounts often involve lengthy verification processes and the collection of a significant amount of personal data, which can be cumbersome and time-consuming. To address this challenge, some banks and financial institutions have streamlined the account creation and transaction processes, allowing customers to perform various tasks with just a username and password. Their personal data is not needed so, privacy is not concerned.

So, with just by providing a username and password an account is created the simple process of depositing, transferring money, checking balance and recharging can be done. The users can manage their accounts and perform various transactions without needing to visit a physical branch or ATM.

Moreover, it can significantly reduce the need for physical bank visits, which is not only time-consuming but can also be inconvenient, particularly for those with busy schedules or limited mobility. This approach aligns with the broader trend of digitization in the financial industry, where mobile and online banking are becoming increasingly popular. It allows customers to manage their finances from the comfort of their homes or on the go, improving their overall banking experience.

In this fast-paced world, simplifying the account creation and transaction processes can save time and reduce the need for physical visits to banks or ATMs. However, it's important to strike a balance between convenience and security to ensure that users' financial information remains safe and protected.

METHODOLOGY / ARCHITECTURE

User Registration

The Bank Management System begins with the user registration process. This step is crucial for creating a secure and personalized banking experience for customers. Users are required to provide their username and password which is stored in a database in a structured table format. The data is securely stored, ensuring data integrity and confidentiality.

Account Management

Once registered, users have the ability to manage their accounts efficiently. This system empowers customers to perform a variety of account-related tasks with ease. They can deposit money into their accounts, view their account balance, and initiate transfers within the same bank or to other banks. This streamlines the financial operations and offers customers a seamless experience.

Deposits and Transfers

The Bank Management System simplifies the process of making deposits and transfers. Customers can deposit funds into their accounts quickly, allowing them to keep their finances organized. Additionally, they have the flexibility to transfer money to other accounts within the same bank or to external banks. This feature promotes financial fluidity and convenience, ensuring that customers can manage their assets effectively.

Mobile Phone Recharge

In addition to traditional banking services, this system also offers the added convenience of mobile phone recharge. Users can recharge their mobile phones with their appropriate phone numbers directly through the application. This integrated feature eliminates the need for separate transactions and provides a one-stop solution for banking and communication needs.

Login and Security

To access their accounts and utilize the various services offered, users can log in with their unique username and password. This login process ensures the security and privacy of user data. The system implements security measures to protect against unauthorized access and fraudulent activities. After three unsuccessful login attempts, a "forgot password" option appears, allowing users to reset their password securely, further enhancing account security.

Database Management

The system employs a robust database to store and manage user information and transactional data. This database ensures data consistency, accuracy, and reliability. It plays a pivotal role in keeping track of user accounts, transactions, and balances.

User-Friendly Interface

The application is designed with the end-user in mind, providing a seamless and intuitive experience. Users can navigate through the system effortlessly, even without extensive technical knowledge. The straightforward design makes it accessible to a wide range of customers, from tech-savvy individuals to those less familiar with technology.

IMPLEMENTATION – CODING AND OUTPUT SCREENSHOT

Main class:

```
package Bank;

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

public class DashBoard {
    public static double balance;
    public static double senderBalance;
    public static String username;
    public static String name;
    public static int loginAttempts = 0;
    JFrame Frame;
    public static String usernamelogin;
    public JPanel buttonPanel= new JPanel();
    public JFrame f= new JFrame("JJ BANK");
    public JPanel panel= new JPanel(new BorderLayout()); ;

    void Dash(){
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setExtendedState(JFrame.MAXIMIZED_BOTH);
        f.getContentPane().setBackground(Color.WHITE);

        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        int x = (screenSize.width - f.getWidth()) / 2;
        int y = (screenSize.height - f.getHeight()) / 2;
        f.setLocation(x, y);

        ImageIcon bankIcon = new ImageIcon("C:/Users/USER/Downloads/bank.jpg");
        JLabel bankImageLabel = new JLabel();
        Image originalImage = bankIcon.getImage();
        int width = 300;
        int height = 300;
        Image scaledImage = originalImage.getScaledInstance(width, height,
Image.SCALE_SMOOTH);
     bankImageLabel.setIcon(new ImageIcon(scaledImage));



        panel.add(bankImageLabel, BorderLayout.WEST);

        JLabel l = new JLabel("<html>"
                + "<div style='text-align: justify;'>"
```

```
            + "<div style='font-size: 14px;'>  Banks play a crucial role in the
global economy and offer a wide range of services that benefit individuals, businesses, and
society as a   whole.</div>"
            + "<div style='font-size: 14px;'>  Banks provide a safe and secure
place for individuals to store their money, ensuring it is protected from theft and loss.</div>"
            + "<div style='font-size: 14px;'>  Banks offer a variety of savings and
investment products, such as savings accounts, certificates of deposit, and mutual funds,
which   help individuals grow their wealth over time.</div>"
            + "<div style='font-size: 14px;'>  Now all the money transfer and
checking money balance can be done at home comfortably.</div>"
            + "<div style='font-size: 18px; font-weight: bold; text-align: center;'>Welcome to
JJ BANK</div>"
            + "</div>"
            + "</html>");
        l.setHorizontalAlignment(JLabel.CENTER);
        panel.add(l, BorderLayout.CENTER);
        //JPanel buttonPanel = new JPanel();

        JButton loginButton = new JButton("Login");
        loginButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                new LLogin();
            }
        });
        buttonPanel.add(loginButton);

        JButton registerButton = new JButton("Register");
        registerButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                new Register(); // Open the registration window
            }
        });
        buttonPanel.add(registerButton);
        f.add(panel, BorderLayout.CENTER);
        f.add(buttonPanel, BorderLayout.SOUTH);
        f.setVisible(true);
    }}

class Display{
public static void main(String[] args){
    DashBoard d=new DashBoard();
    d.Dash();

}}
```

LLogin class

```
public class LLogin extends DashBoard {
   public LLogin(){
            JFrame loginFrame = new JFrame("Login Page");
            loginFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            loginFrame.setSize(800, 600);

            Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
            int x = (screenSize.width - loginFrame.getWidth()) / 2;
            int y = (screenSize.height - loginFrame.getHeight()) / 2;
            loginFrame.setLocation(x, y);

            JPanel loginPanel = new JPanel(new GridBagLayout());
            GridBagConstraints gbc = new GridBagConstraints();
            gbc.insets = new Insets(5, 5, 5, 5);

            JLabel usernameLabel = new JLabel("Username:");
            gbc.gridx = 0;
            gbc.gridy = 0;
            loginPanel.add(usernameLabel, gbc);

            JTextField usernameField = new JTextField(20);
            gbc.gridx = 1;
            loginPanel.add(usernameField, gbc);

            JLabel passwordLabel = new JLabel("Password:");
            gbc.gridx = 0;
            gbc.gridy = 1;
            loginPanel.add(passwordLabel, gbc);

            JPasswordField passwordField = new JPasswordField(20);
            gbc.gridx = 1;
            loginPanel.add(passwordField, gbc);

            JButton loginButton = new JButton("Login");
            gbc.gridx = 1;
            gbc.gridy = 2;
            loginPanel.add(loginButton, gbc);

            loginButton.addActionListener(new ActionListener() {
               public void actionPerformed(ActionEvent ae) {
                  String passwordlogin = new String(passwordField.getPassword());
                  String usernamelogin = usernameField.getText();
```

```
                try {
                    Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");
                    PreparedStatement preparedStatement =
connection.prepareStatement("SELECT * from users where username = ? AND password =
?");
                    preparedStatement.setString(1, usernamelogin);
                    preparedStatement.setString(2, passwordlogin);
                    ResultSet resultSet = preparedStatement.executeQuery();

                    if (resultSet.next()) {
                        name = resultSet.getString("username");
                        username = usernamelogin;
                        balance = resultSet.getDouble("balance");
                        OpenWindow o=new OpenWindow();
                    } else {
                        loginAttempts++;

                        if (loginAttempts >= 3) {
                            JOptionPane.showMessageDialog(loginPanel, "Authentication failed.
You've exceeded the maximum login attempts.\nPlease register to continue.");
                            JButton j = new JButton("Forgot Password?");
                            gbc.gridx = 0;
                            gbc.gridy = 3;
                            loginPanel.add(j, gbc);
                            loginFrame.revalidate();
                            loginFrame.repaint();

                            JButton r2 = new JButton("Register");
                            gbc.gridx = 2;
                            gbc.gridy = 3;

                            loginPanel.add(r2, gbc);
                            loginFrame.revalidate();
                            loginFrame.repaint();

                            r2.addActionListener(new ActionListener() {
                            public void actionPerformed(ActionEvent ae) {
                                new Register();
                            }
                        });
                            j.addActionListener(new ActionListener() {
                                public static boolean usernameExists;
                                public static ResultSet resultSet;
                                public void actionPerformed(ActionEvent ae) {
                                String usernamelogin = usernameField.getText();
```

```
                    try {
                        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");
                        PreparedStatement preparedStatement =
connection.prepareStatement("SELECT * FROM users WHERE username = ?");
                        preparedStatement.setString(1, usernamelogin);
                        ResultSet resultSet = preparedStatement.executeQuery();

                        if (resultSet.next()) {
                            JFrame resetPasswordFrame = new JFrame("Reset Password");

resetPasswordFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                            resetPasswordFrame.setSize(400, 200);

                            JPanel resetPasswordPanel = new JPanel(new
GridBagLayout());
                            GridBagConstraints gbc = new GridBagConstraints();
                            gbc.insets = new Insets(5, 5, 5, 5);

                            JLabel newPasswordLabel = new JLabel("New Password:");
                            gbc.gridx = 0;
                            gbc.gridy = 0;
                            resetPasswordPanel.add(newPasswordLabel, gbc);

                            JPasswordField newPasswordField = new JPasswordField(20);
                            gbc.gridx = 1;
                            resetPasswordPanel.add(newPasswordField, gbc);

                            JLabel reenterPasswordLabel = new JLabel("Re-enter New
Password:");
                            gbc.gridx = 0;
                            gbc.gridy = 1;
                            resetPasswordPanel.add(reenterPasswordLabel, gbc);

                            JPasswordField reenterPasswordField = new
JPasswordField(20);
                            gbc.gridx = 1;
                            resetPasswordPanel.add(reenterPasswordField, gbc);

                            JButton resetPasswordButton = new JButton("Reset
Password");
                            gbc.gridx = 1;
                            gbc.gridy = 2;
                            resetPasswordPanel.add(resetPasswordButton, gbc);
```

```
                                resetPasswordButton.addActionListener(new ActionListener()
{
                                    public void actionPerformed(ActionEvent ae) {
                                        String newPassword = new
String(newPasswordField.getPassword());
                                        String reenteredPassword = new
String(reenterPasswordField.getPassword());

                                        if (newPassword.equals(reenteredPassword)) {
                                            // Update the database with the new password for the
current user (username)
                                            if (newPassword.length() < 8) {

JOptionPane.showMessageDialog(resetPasswordPanel, "New password should have a
minimum length of 8.");
                                            } else {
                                                try {

                                                    PreparedStatement preparedStatement2 =
connection.prepareStatement("UPDATE users SET password=? WHERE username=?");
                                                    preparedStatement2.setString(1, newPassword);
                                                    preparedStatement2.setString(2, usernamelogin);
                                                    preparedStatement2.executeUpdate();


JOptionPane.showMessageDialog(resetPasswordPanel, "Password reset successful.");
                                                    resetPasswordFrame.dispose(); // Close the
password reset window
                                                    connection.close();
                                                } catch (SQLException e) {
                                                    e.printStackTrace();
                                                }
                                            }
                                        } else {
                                            JOptionPane.showMessageDialog(resetPasswordPanel,
"Passwords do not match. Please re-enter the same password.");
                                        }
                                    }
                                });

                                resetPasswordFrame.add(resetPasswordPanel);
                                resetPasswordFrame.setVisible(true);
                            } else {
                                JOptionPane.showMessageDialog(loginPanel, "Username not
found in the database. Please check the username.");
                            }
```

```
                        }
                        catch(SQLException e){}
                    }
                });
            } else {
                JOptionPane.showMessageDialog(loginPanel, "Authentication failed.
Invalid username or password.");
            }
        }
        connection.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
  }
});

loginFrame.add(loginPanel);
loginFrame.setVisible(true);
}

}
```

Register class

```
public class Register extends DashBoard {
   public Register(){
        JFrame jFrame = new JFrame("Register Page");
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        int x = (screenSize.width - jFrame.getWidth()) / 2;
        int y = (screenSize.height - jFrame.getHeight()) / 2;
        jFrame.setLocation(x, y);

        JPanel registerForm = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(5, 5, 5, 5);

        JLabel l = new JLabel("Name :");
        gbc.gridx = 0;
        gbc.gridy = 0;
        registerForm.add(l, gbc);

        JTextField t = new JTextField(20);
        gbc.gridx = 1;
        registerForm.add(t, gbc);

        JLabel l1 = new JLabel("Username :");
```

```
            gbc.gridx = 0;
            gbc.gridy = 1;
            registerForm.add(l1, gbc);

            JTextField t1 = new JTextField(20);
            gbc.gridx = 1;
            registerForm.add(t1, gbc);

            JLabel l2 = new JLabel("Password:");
            gbc.gridx = 0;
            gbc.gridy = 2;
            registerForm.add(l2, gbc);

            JPasswordField t2 = new JPasswordField(20);
            gbc.gridx = 1;
            registerForm.add(t2, gbc);

            JButton addButton = new JButton("Submit");
            gbc.gridx = 1;
            gbc.gridy = 3;
            registerForm.add(addButton, gbc);

            addButton.addActionListener(new ActionListener() {
               public void actionPerformed(ActionEvent ae) {
                  String enteredName = t.getText();
                  String enteredUsername = t1.getText();
                  String enteredPassword = new String(t2.getPassword());

                  if (enteredUsername.isEmpty() || enteredPassword.isEmpty() ||
enteredName.isEmpty()) {
                     JOptionPane.showMessageDialog(registerForm, "Username, password, or
name should not be empty");
                  } else if (enteredPassword.length() < 8) {
                     JOptionPane.showMessageDialog(registerForm, "Password should have a
minimum length of 8");
                  } else {
                     try {
                        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");
                        PreparedStatement checkStatement =
connection.prepareStatement("SELECT COUNT(*) FROM users WHERE username = ?");
                  checkStatement.setString(1, enteredUsername);
                  ResultSet resultSet = checkStatement.executeQuery();
                  resultSet.next();
                  int count = resultSet.getInt(1);
```

```
                if (count > 0) {
                    JOptionPane.showMessageDialog(registerForm, "Username already exists.
Please choose a different username.");
                } else {
                    //Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");
                    PreparedStatement preparedStatement =
connection.prepareStatement("INSERT INTO users (username, password, balance) VALUES
(?, ?, ?)");
                    preparedStatement.setString(1, enteredUsername);
                    preparedStatement.setString(2, enteredPassword);
                    preparedStatement.setDouble(3, 0.0);
                    preparedStatement.executeUpdate();

                    //name = enteredName;
                    username = enteredUsername;
                    OpenWindow o=new OpenWindow();
                    connection.close();}
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }
    });

    jFrame.add(registerForm);
    jFrame.setSize(800, 600);
    jFrame.setLocationRelativeTo(null);
    jFrame.setVisible(true);
  }
}
```

OpenWindow class

```
public class OpenWindow extends DashBoard{
   public OpenWindow(){
      JFrame Frame = new JFrame("BANK");
      Frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      JPanel b = new JPanel(new GridLayout(5, 1, 10, 10));
      JButton b1 = new JButton("1. Balance");
      b.add(b1);
      JButton b2 = new JButton("2. Deposit");
      b.add(b2);
      JButton b3 = new JButton("3. Transfer");
      b.add(b3);
```

```
        JButton b4 = new JButton("4. Recharge");
        b.add(b4);
        JButton b5 = new JButton("5. Exit");
        b.add(b5);
        Frame.add(b);

        b2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                Deposit d=new Deposit();
            }
        });

        b1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                try {
            Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");

            PreparedStatement senderBalanceStatement =
connection.prepareStatement("SELECT balance FROM users WHERE username = ?");
            senderBalanceStatement.setString(1, username);
            ResultSet senderBalanceResult = senderBalanceStatement.executeQuery();
            senderBalanceResult.next();
            double senderBalance = senderBalanceResult.getDouble("balance");
                JOptionPane.showMessageDialog(Frame, "Balance Amount: Rs" +
senderBalance);
                }
            catch(SQLException e){}
            }
        });
        b3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                String[] options = { "JJ Bank", "Other Banks" };
                int selectedOption = JOptionPane.showOptionDialog(Frame, "Select a Bank to
Transfer to:", "Bank Selection",
                        JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null,
options, options[0]);
                if (selectedOption == 0) {
                    JJBankTransfer jjBankTransfer = new JJBankTransfer();
                } else if (selectedOption == 1) {
                    OtherBankTransferDialog otherBankTransferDialog = new
OtherBankTransferDialog();
                }
            }
        });
```

```
        b4.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
           Recharge re=new Recharge();
        }});

           b5.addActionListener(new ActionListener() {
              public void actionPerformed(ActionEvent ae) {
                 JOptionPane.showMessageDialog(Frame, "Vist us Again.Thank You");
                 System.exit(0);
              }
           });

           Frame.setSize(800, 600);
           Frame.setLocationRelativeTo(null);

           Frame.setVisible(true);
        }
     }
```

Deposit class

```
class Deposit extends DashBoard{
   public Deposit(){
   boolean valid = false;
            while (!valid) {
               String depositAmountStr = JOptionPane.showInputDialog(Frame, "Enter the
deposit amount:");
               if (depositAmountStr == null) {
                  valid = true;
                  return;
               }
               try {
                  double depositAmount = Double.parseDouble(depositAmountStr);
                  if (depositAmount <= 0) {
                     JOptionPane.showMessageDialog(Frame, "Deposit amount must be greater
than 0.");
                  } else if (balance < 500 && depositAmount < 500) {
                     JOptionPane.showMessageDialog(Frame, "The initial deposit must be at
least 500.");
                  } else {
                     valid = true;
                     balance += depositAmount;
                     JOptionPane.showMessageDialog(Frame, "Deposit of Rs" + depositAmount
+ " was successful.");
```

```
                    try {
                        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");
                        PreparedStatement retrieveStatement =
connection.prepareStatement("SELECT balance FROM users WHERE username=?");
                retrieveStatement.setString(1, username);
                ResultSet resultSet = retrieveStatement.executeQuery();
                if (resultSet.next()) {
                    double currentBalance = resultSet.getDouble("balance");
                    currentBalance += depositAmount;
                    PreparedStatement updateStatement =
connection.prepareStatement("UPDATE users SET balance=? WHERE username=?");
                    updateStatement.setDouble(1, currentBalance);
                    updateStatement.setString(2, username);
                    updateStatement.executeUpdate();
                    connection.close();
                    } }catch (SQLException e) {
                        e.printStackTrace();
                    }
                }
            } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(Frame, "Invalid input. Please enter a valid
amount.");
            }
        }
}
}
```

Recharge class

```
public class Recharge extends DashBoard {
    public Recharge() {
        JFrame rechargeFrame = new JFrame("Recharge");
        rechargeFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        rechargeFrame.setSize(500, 500);

        JPanel rechargePanel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(5, 5, 5, 5);

        JLabel phoneNumberLabel = new JLabel("Phone number:");
        gbc.gridx = 0;
        gbc.gridy = 0;
        rechargePanel.add(phoneNumberLabel, gbc);
```

```
        JTextField phoneNumberField = new JTextField(15); // Allowing for a 10-digit phone number
        gbc.gridx = 2;
        rechargePanel.add(phoneNumberField, gbc);

        JButton rechargeButton = new JButton("Recharge");
        gbc.gridx = 0;
        gbc.gridy = 1;
        gbc.gridwidth = 2;
        rechargePanel.add(rechargeButton, gbc);

        JButton tp = new JButton("Top up");
        gbc.gridx = 2;
        gbc.gridy = 1;
        gbc.gridwidth = 2;
        rechargePanel.add(tp, gbc);

        rechargeButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent event) {
                String phoneNumber = phoneNumberField.getText();

                if (phoneNumber.isEmpty()) {
                    JOptionPane.showMessageDialog(rechargeFrame, "Enter a phone number.");
                } else if (phoneNumber.length() != 10 || !phoneNumber.matches("\\d+")) {
                    JOptionPane.showMessageDialog(rechargeFrame, "Invalid phone number.");
                } else {
                    JFrame plansFrame = new JFrame("Recharge Plans");
                    plansFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                    plansFrame.setSize(400, 200);

                    JPanel plansPanel = new JPanel(new GridBagLayout());
                    GridBagConstraints gbc = new GridBagConstraints();
                    gbc.insets = new Insets(5, 5, 5, 5);

                    JLabel selectPlanLabel = new JLabel("Select Recharge Plan:");
                    gbc.gridx = 0;
                    gbc.gridy = 0;
                    plansPanel.add(selectPlanLabel, gbc);

                    // Create a combo box with recharge plans and their prices
                    String[] plans = {"Rs. 279 - For 28 days (1.5 GB/day)", "Rs. 359 - For 28 days (2 GB/day)", "Rs. 459 - For 40 days (1.5 GB/day)"};
                    JComboBox<String> planComboBox = new JComboBox<>(plans);
                    gbc.gridx = 1;
                    gbc.gridy = 0;
                    plansPanel.add(planComboBox, gbc);
```

```
            JButton confirmButton = new JButton("Confirm Recharge");
            gbc.gridx = 0;
            gbc.gridy = 1;
            gbc.gridwidth = 2;
            plansPanel.add(confirmButton, gbc);

            confirmButton.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent event) {
                    String selectedPlan = (String) planComboBox.getSelectedItem();
                    double rechargeAmount = extractRechargeAmount(selectedPlan);
                    if (balance < rechargeAmount) {
                        JOptionPane.showMessageDialog(plansFrame, "Insufficient balance.
Your balance is Rs " + balance);
                    } else {
                        balance -= rechargeAmount;
                        try {
                            Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");
                            PreparedStatement retrieveStatement =
connection.prepareStatement("SELECT balance FROM users WHERE username=?");
                            retrieveStatement.setString(1, username);
                            ResultSet resultSet = retrieveStatement.executeQuery();
                            if (resultSet.next()) {
                                double currentBalance = resultSet.getDouble("balance");
                                currentBalance -= rechargeAmount;
                                PreparedStatement updateStatement =
connection.prepareStatement("UPDATE users SET balance=? WHERE username=?");
                                updateStatement.setDouble(1, currentBalance);
                                updateStatement.setString(2, username);
                                updateStatement.executeUpdate();
                                connection.close();
                            }
                            JOptionPane.showMessageDialog(plansFrame, "Recharge of Rs " +
rechargeAmount + " was successful.");
                        } catch (SQLException e) {
                            e.printStackTrace();
                        }
                    }
                    plansFrame.dispose();
                }
            });
            plansFrame.add(plansPanel);
            plansFrame.setLocationRelativeTo(null);
            plansFrame.setVisible(true);
```

```
                rechargeFrame.dispose();
            }
        }
    });

    tp.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent event) {
            String phoneNumber = phoneNumberField.getText();

            if (phoneNumber.isEmpty()) {
                JOptionPane.showMessageDialog(rechargeFrame, "Enter phone number.");
            } else if (phoneNumber.length() != 10 || !phoneNumber.matches("\\d+")) {
                JOptionPane.showMessageDialog(rechargeFrame, "Invalid phone number.");
            } else {
                JFrame tplans = new JFrame("TopUp Plans");
                tplans.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                tplans.setSize(400, 200);

                JPanel plansPanel = new JPanel(new GridBagLayout());
                GridBagConstraints gbc = new GridBagConstraints();
                gbc.insets = new Insets(5, 5, 5, 5);

                JLabel selectPlanLabel = new JLabel("Select Recharge TopUp Plan:");
                gbc.gridx = 0;
                gbc.gridy = 0;
                plansPanel.add(selectPlanLabel, gbc);

                String[] plans = {"Rs. 19 (2 GB/day)", "Rs. 25 - (3 GB/day)", "Rs. 55 - (6
GB/day)"};
                JComboBox<String> planComboBox = new JComboBox<>(plans);
                gbc.gridx = 1;
                gbc.gridy = 0;
                plansPanel.add(planComboBox, gbc);

                JButton confirmButton = new JButton("Confirm TopUp");
                gbc.gridx = 0;
                gbc.gridy = 1;
                gbc.gridwidth = 2;
                plansPanel.add(confirmButton, gbc);

                confirmButton.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent event) {
                        String selectedPlan = (String) planComboBox.getSelectedItem();
                        double rechargeAmount = extractRechargeAmount(selectedPlan);
```

```
                    if (balance < rechargeAmount) {
                        JOptionPane.showMessageDialog(tplans, "Insufficient balance. Your
balance is Rs " + balance);
                    } else {
                        balance -= rechargeAmount;
                        try {
                            Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");
                            PreparedStatement retrieveStatement =
connection.prepareStatement("SELECT balance FROM users WHERE username=?");
                            retrieveStatement.setString(1, username);
                            ResultSet resultSet = retrieveStatement.executeQuery();
                            if (resultSet.next()) {
                                double currentBalance = resultSet.getDouble("balance");
                                currentBalance -= rechargeAmount;
                                PreparedStatement updateStatement =
connection.prepareStatement("UPDATE users SET balance=? WHERE username=?");
                                updateStatement.setDouble(1, currentBalance);
                                updateStatement.setString(2, username);
                                updateStatement.executeUpdate();
                                connection.close();
                            }
                            JOptionPane.showMessageDialog(tplans, "Recharge of Rs " +
rechargeAmount + " was successful.");
                        } catch (SQLException e) {
                            e.printStackTrace();
                        }
                    }

                    tplans.dispose();
                }
            });
            tplans.add(plansPanel);
            rechargeFrame.setLocationRelativeTo(null);
            tplans.setLocationRelativeTo(null);
            tplans.setVisible(true);
            rechargeFrame.dispose();
        }
    }
});

rechargeFrame.add(rechargePanel);
rechargeFrame.setLocationRelativeTo(null);
rechargeFrame.setVisible(true);
}
```

```
    private double extractRechargeAmount(String plan) {
        String[] parts = plan.split(" ");
        double rechargeAmount = Double.parseDouble(parts[1]);
        return rechargeAmount;
    }
}
```

JJBankTransfer class

```
public class JJBankTransfer extends DashBoard{
    public JJBankTransfer() {
        JFrame transferFrame = new JFrame("JJ Bank Transfer");
        transferFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JPanel transferPanel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(5, 5, 5, 5);

        JLabel lblRecipientUsername = new JLabel("Recipient Username:");
        gbc.gridx = 0;
        gbc.gridy = 0;
        transferPanel.add(lblRecipientUsername, gbc);

        JTextField txtRecipientUsername = new JTextField(20);
        gbc.gridx = 1;
        transferPanel.add(txtRecipientUsername, gbc);

        JLabel lblAmount = new JLabel("Amount:");
        gbc.gridx = 0;
        gbc.gridy = 1;
        transferPanel.add(lblAmount, gbc);

        JTextField txtAmount = new JTextField(20);
        gbc.gridx = 1;
        transferPanel.add(txtAmount, gbc);

        JLabel lblReason = new JLabel("Reason:");
        gbc.gridx = 0;
        gbc.gridy = 2;
        transferPanel.add(lblReason, gbc);

        JComboBox<String> reasonComboBox = new JComboBox<>();
        reasonComboBox.addItem("Payment");
        reasonComboBox.addItem("Transfer");
        reasonComboBox.addItem("Education purpose");
```

```
        reasonComboBox.addItem("Loan");
        reasonComboBox.addItem("Other");
        gbc.gridx = 1;
        transferPanel.add(reasonComboBox, gbc);

        JButton transferButton = new JButton("Transfer");
        gbc.gridx = 1;
        gbc.gridy = 3;
        transferPanel.add(transferButton, gbc);

        transferButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                String senderUsername = username;
                String recipientUsername = txtRecipientUsername.getText();
                String transferAmountStr = txtAmount.getText();

                if (isRecipientUsernameValid(recipientUsername) &&
isAmountValid(transferAmountStr)) {
                    double transferAmount = Double.parseDouble(transferAmountStr);

                    if (performTransfer(senderUsername, recipientUsername, transferAmount)) {
                        JOptionPane.showMessageDialog(transferFrame, "Transfer successful.");
                        transferFrame.dispose();
                    } else {
                        JOptionPane.showMessageDialog(transferFrame, "Transfer failed. Please
check your balance.");
                    }
                } else {
                    if (!isRecipientUsernameValid(recipientUsername)) {
                        JOptionPane.showMessageDialog(transferFrame, "Recipient username not
found in the database. Please enter a valid username.");
                    } else {
                        JOptionPane.showMessageDialog(transferFrame, "Invalid amount. Please
enter a valid numeric amount.");
                    }
                }
            }
        });

        transferFrame.add(transferPanel);
        transferFrame.setSize(400, 250);
        transferFrame.setLocationRelativeTo(null);
        transferFrame.setVisible(true);
    }
```

```
    private boolean isRecipientUsernameValid(String recipientUsername) {
        boolean isValid = false;

        try {
            Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");
            PreparedStatement checkStatement = connection.prepareStatement("SELECT
COUNT(*) FROM users WHERE username = ?");
            checkStatement.setString(1, recipientUsername);
            ResultSet resultSet = checkStatement.executeQuery();
            resultSet.next();
            int count = resultSet.getInt(1);

            if (count > 0) {
                isValid = true;
            }

            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return isValid;
    }

    private boolean isAmountValid(String amountStr) {
        try {
            double transferAmount = Double.parseDouble(amountStr);
            return transferAmount >= 0; // Ensure that the amount is a non-negative number
        } catch (NumberFormatException e) {
            return false; // Invalid input, not a valid numeric amount
        }
    }

    private boolean performTransfer(String senderUsername, String recipientUsername,
double amount) {
        try {
            Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");
            PreparedStatement senderBalanceStatement =
connection.prepareStatement("SELECT balance FROM users WHERE username = ?");
            senderBalanceStatement.setString(1, senderUsername);
            ResultSet senderBalanceResult = senderBalanceStatement.executeQuery();
            senderBalanceResult.next();
            double senderBalance = senderBalanceResult.getDouble("balance");
```

```
            if (senderBalance >= amount) {
                senderBalance -= amount;
                PreparedStatement updateSenderBalanceStatement =
connection.prepareStatement("UPDATE users SET balance = ? WHERE username = ?");
                updateSenderBalanceStatement.setDouble(1, senderBalance);
                updateSenderBalanceStatement.setString(2, senderUsername);
                updateSenderBalanceStatement.executeUpdate();

                PreparedStatement recipientBalanceStatement =
connection.prepareStatement("SELECT balance FROM users WHERE username = ?");
                recipientBalanceStatement.setString(1, recipientUsername);
                ResultSet recipientBalanceResult = recipientBalanceStatement.executeQuery();
                recipientBalanceResult.next();
                double recipientBalance = recipientBalanceResult.getDouble("balance");
                recipientBalance += amount;

                PreparedStatement updateRecipientBalanceStatement =
connection.prepareStatement("UPDATE users SET balance = ? WHERE username = ?");
                updateRecipientBalanceStatement.setDouble(1, recipientBalance);
                updateRecipientBalanceStatement.setString(2, recipientUsername);
                updateRecipientBalanceStatement.executeUpdate();
                connection.close();
                return true;
            } else {
                connection.close();
                return false;
            }
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

OtherBankTransferdialog class

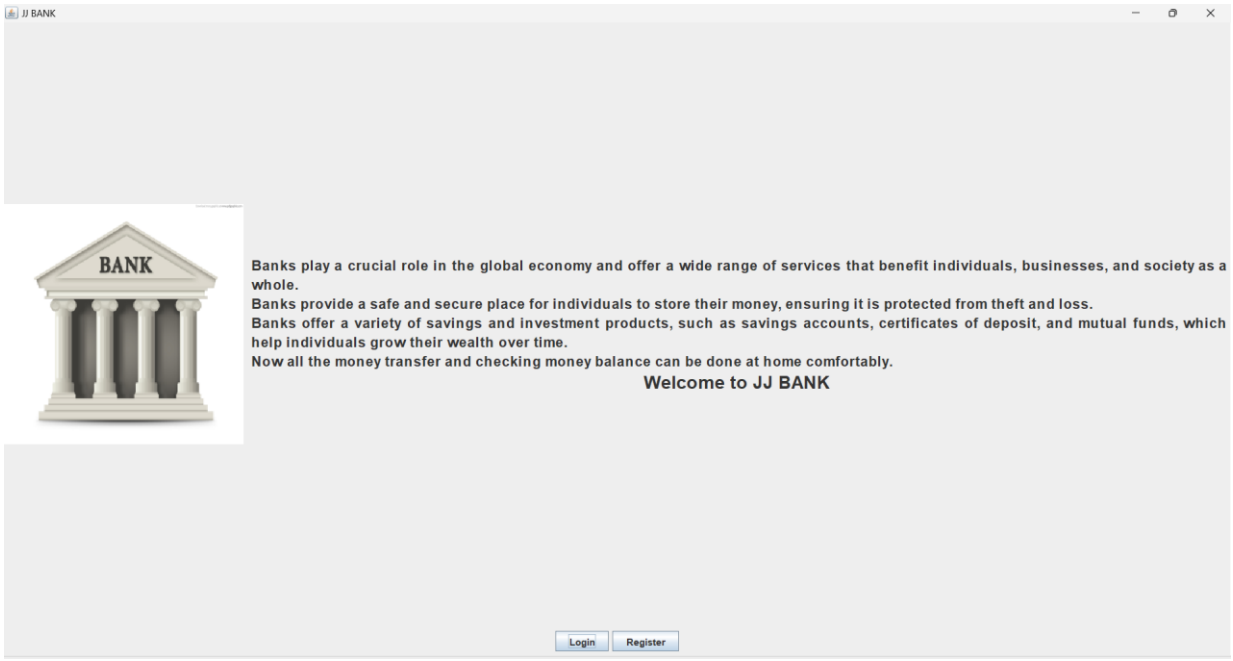
```
public class OtherBankTransferDialog extends DashBoard{
    public OtherBankTransferDialog(){
    JTextField accountNumberField;
    JTextField bankNameField;
    JTextField amountField;
    JComboBox<String> reasonComboBox;
    JFrame jj=new JFrame("Other banks");
    JPanel transferPanel = new JPanel(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.insets = new Insets(5, 5, 5, 5);
```

```
JLabel lblAccountNumber = new JLabel("Account Number:");
gbc.gridx = 0;
gbc.gridy = 0;
transferPanel.add(lblAccountNumber, gbc);

accountNumberField = new JTextField(20);
gbc.gridx = 1;
transferPanel.add(accountNumberField, gbc);

JLabel lblBankName = new JLabel("Bank Name:");
gbc.gridx = 0;
gbc.gridy = 1;
transferPanel.add(lblBankName, gbc);

bankNameField = new JTextField(20);
gbc.gridx = 1;
transferPanel.add(bankNameField, gbc);

JLabel lblAmount = new JLabel("Amount:");
gbc.gridx = 0;
gbc.gridy = 2;
transferPanel.add(lblAmount, gbc);

amountField = new JTextField(20);
gbc.gridx = 1;
transferPanel.add(amountField, gbc);

JLabel lblReason = new JLabel("Reason:");
gbc.gridx = 0;
gbc.gridy = 3;
transferPanel.add(lblReason, gbc);

reasonComboBox = new JComboBox<>();
reasonComboBox.addItem("Payment");
reasonComboBox.addItem("Transfer");
reasonComboBox.addItem("Education purpose");
reasonComboBox.addItem("Loan");
reasonComboBox.addItem("Other");
gbc.gridx = 1;
transferPanel.add(reasonComboBox, gbc);

JButton transferButton = new JButton("Transfer");
gbc.gridx = 1;
gbc.gridy = 4;
transferPanel.add(transferButton, gbc);
```

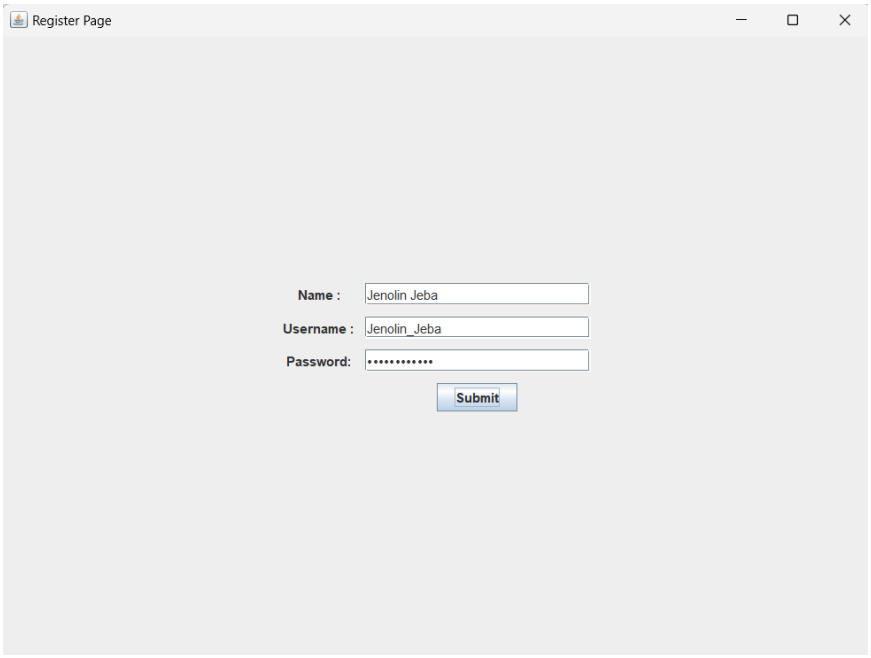
```
        transferButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                String accountNumber = accountNumberField.getText();
                String bankName = bankNameField.getText();
                String transferAmountStr = amountField.getText();
                String transferReason = reasonComboBox.getSelectedItem().toString();

                if (isAmountValid(transferAmountStr)) {
                    double transferAmount = Double.parseDouble(transferAmountStr);
                if (performTransfer(accountNumber, transferAmount)) {
                        JOptionPane.showMessageDialog(jj, "Amount Successfully Transferred");
                        jj.dispose();
                    } else {
                        JOptionPane.showMessageDialog(jj, "Transfer failed. Please check the
account information.");
                    }
                } else {
                    JOptionPane.showMessageDialog(jj, "Invalid amount. Please enter a valid
numeric amount.");
                }
            }
        });

        jj.setVisible(true);
        jj.add(transferPanel);
        jj.setSize(400, 250);
        jj.setLocationRelativeTo(null);

    }

    private boolean isAmountValid(String amountStr) {
        try {
            double transferAmount = Double.parseDouble(amountStr);
            return transferAmount >= 0; // Ensure that the amount is a non-negative number
        } catch (NumberFormatException e) {
            return false; // Invalid input, not a valid numeric amount
        }
    }

    private boolean performTransfer(String accountNumber, double transferAmount) {
        String updateBalanceQuery = "UPDATE users SET balance = balance - ? WHERE
username = ?";
```

```
        try {
            Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bank", "root", "root");

            PreparedStatement senderBalanceStatement =
connection.prepareStatement(updateBalanceQuery);
            senderBalanceStatement.setDouble(1, transferAmount);
            senderBalanceStatement.setString(2, username);
            senderBalanceStatement.executeUpdate();
            connection.close();
            return true;

        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

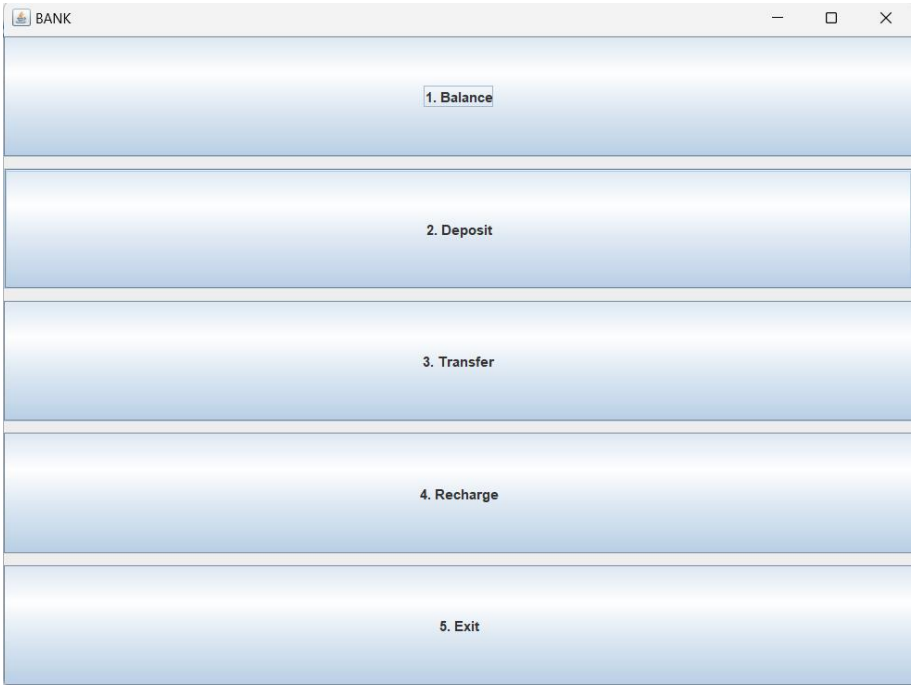
OUTPUT:



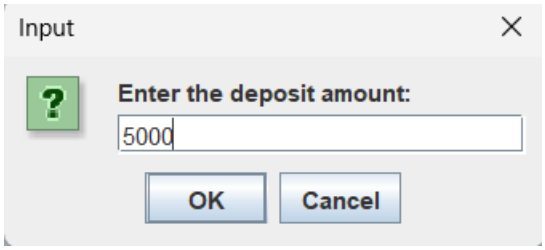
Register Page

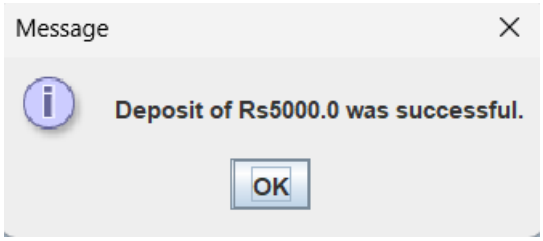


Bank Page

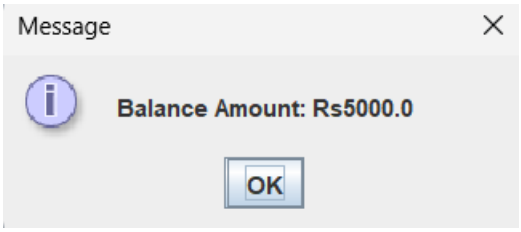


Deposit Dialog

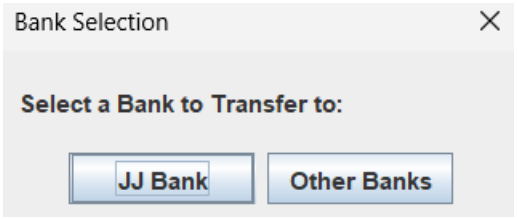




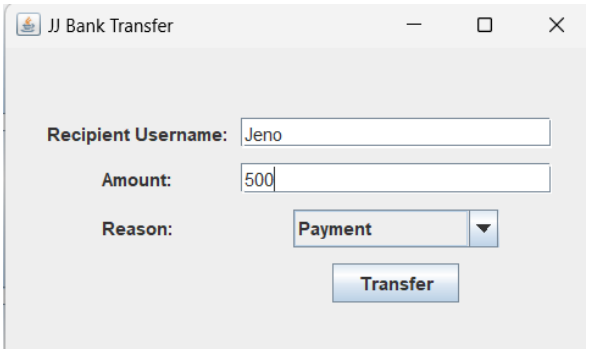
Balance Dialog

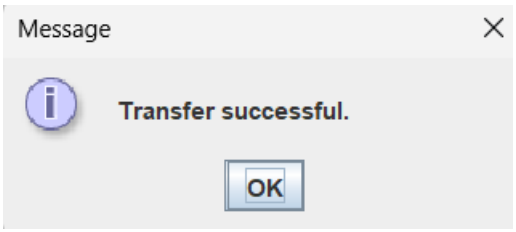


Money Transfer Dialog

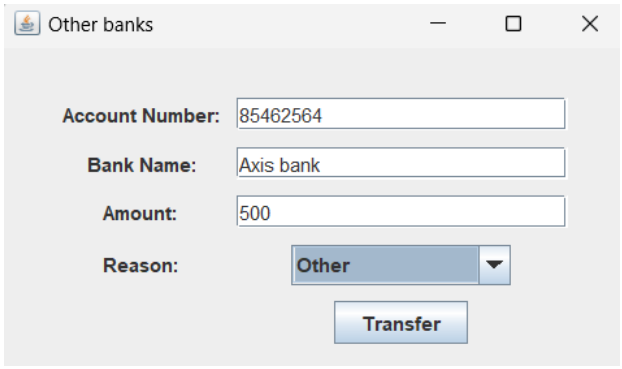


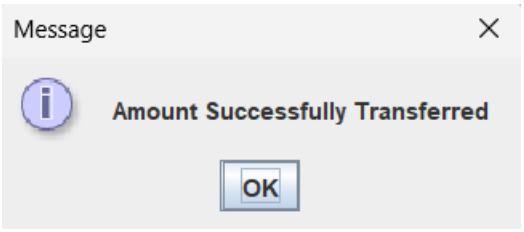
Transfer to another user of JJ Bank



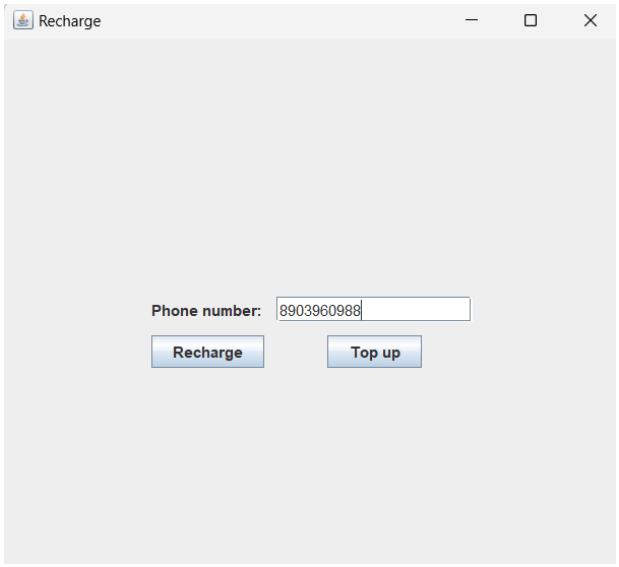


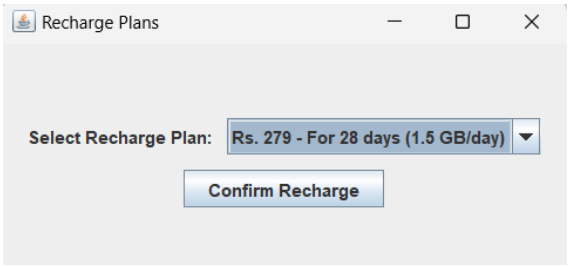
Transfer to another Bank

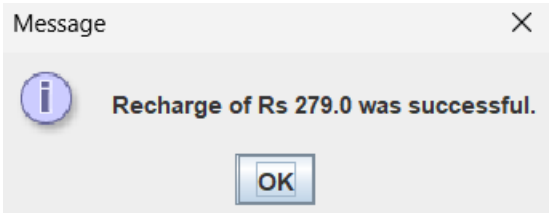




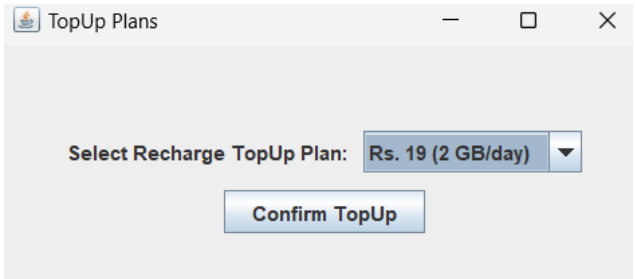
Recharge Page

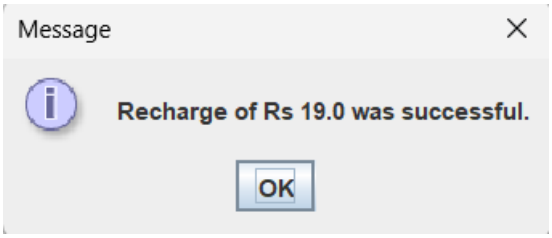




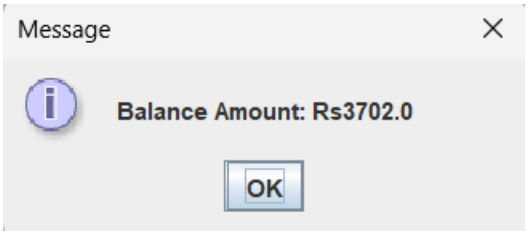


TopUp

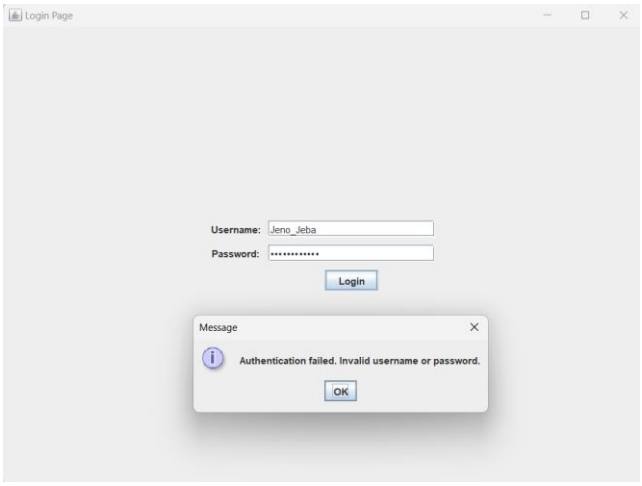




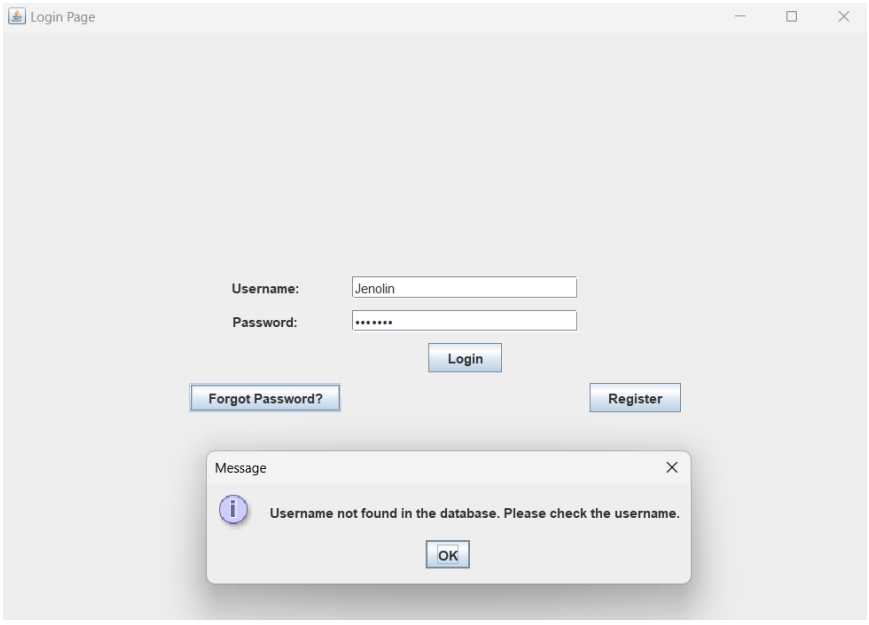
Balance dialog



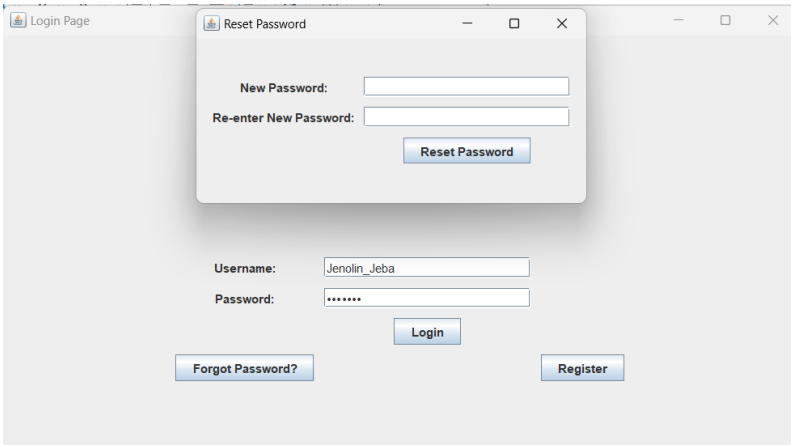
Login Failure Page



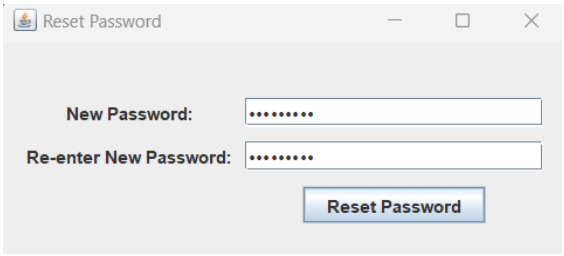
After 3 attempts of failing to Login

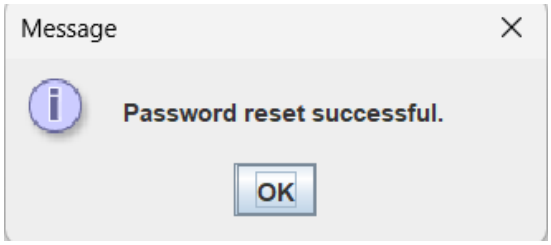


Forgot password Page

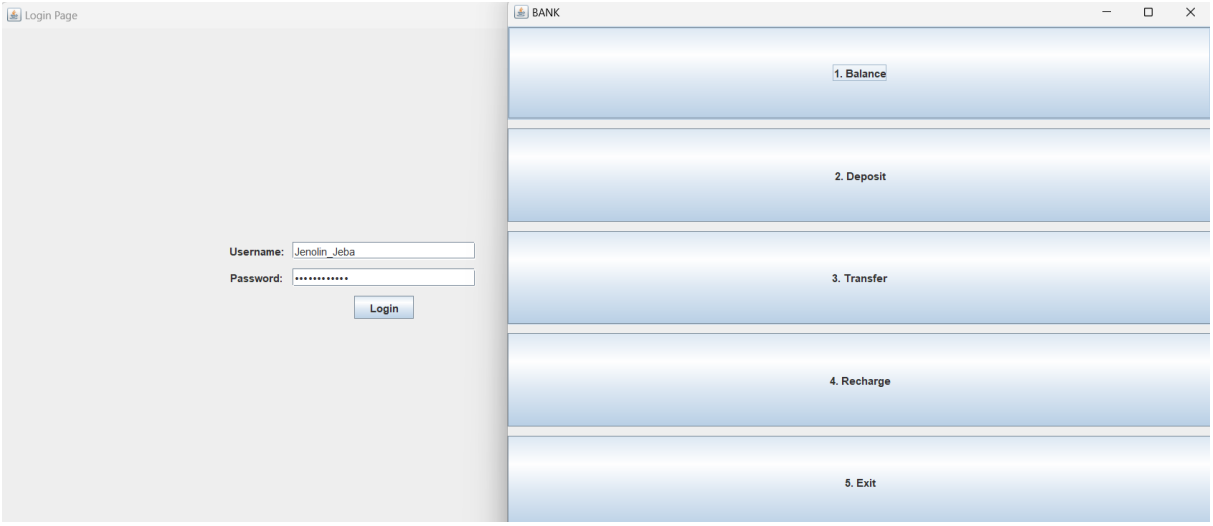


Resetting Password





Successful Login



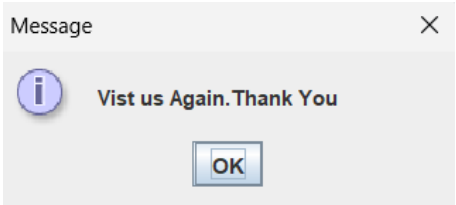
The database after deposit, transferring money and recharge

S_No	username	password	balance
1	Jeno	Jeno	643.00
2	JenoJeba	JenoJeba	3150.00
3	JenoJJJJ	JenoJJJJ	600.00
5	Jeni	JeniJeba	1000.00
6	Jenolin_Jeba	JenoJeba1222	3702.00

The database after resetting password

S_No	username	password	balance
1	Jeno	Jeno	643.00
2	JenoJeba	JenoJeba	3150.00
3	JenoJJJJ	JenoJJJJ	600.00
5	Jeni	JeniJeba	1000.00
6	Jenolin_Jeba	JJenoJeba	3702.00

The Exit Dialog



CONCLUSION

The Bank Management System has emerged as a valuable tool for modern banking institutions. It simplifies and enhances banking operations, making account management, deposits, transfers, and mobile phone recharge more convenient than ever. The system also prioritizes security, offering multiple layers of protection to safeguard customer data and financial transactions. Its user-friendly interface ensures that individuals of varying technological backgrounds can access and utilize the application with ease.

Overall, the Bank Management System is a testament to the continuous evolution of the banking industry, providing a comprehensive solution to meet the demands of today's customers and streamline the operations of modern banking institutions. This system represents a bridge between traditional banking and the digital age, offering a secure, efficient, and user-friendly banking experience for all.

Moreover, the user-friendly interface of the system ensures that individuals, whether tech-savvy or not, can navigate and utilize the application effortlessly. This inclusive approach bridges the gap between traditional banking and the digital age, making banking services accessible to a wider audience.

Bank Management System is a testimony to the ever-evolving banking industry. It addresses the demands of modern customers by offering a secure, efficient, and user-friendly banking experience. It enhances operational efficiency for banking institutions and enhances customer satisfaction by simplifying account management and transactions while maintaining a strong focus on security. As banking continues to adapt to the digital age, this system represents a significant stride towards a more seamless and customer-centric future in the financial sector.

EVALUATION SHEET

Reg.No : URK22CS5026

Name: J Jenolin Jeba

Course code: 20CS2035

Course Name: Object Oriented Programming

S.No	Rubrics	Maximum Marks	Marks Obtained
1	Industrial Certification	10	
2	Real – Time Application Design	30	
Total		40	

Rubrics	Excellent	Good	Average	Below Average
Classes and Inheritance				
Concept Used				
GUI				
Database				
Innovation				
Presentation and Viva				
Report				

Signature of the Faculty-in-charge