AUTOMATED TELLER MACHINE

A MINI-PROJECT REPORT

Submitted by

ABISHEK (231001006)

DINESH S (231001040)

in partial fulfillment of the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

November 2024

BONAFIDE CERTIFICATE

Certified that this Project report **"ATM MACHINE"** is the bonafide work of **ABISHEK (231001006), DINESH S (231001040)** who carried out the project work under my supervision.

SIGNATURE
Dr.P.Valarmathie
HEAD OF THE DEPARTMENT

Information Technology
Rajalakshmi Engineering College

SIGNATURE
Mrs.Usha S
COURSE INCHARGE
Assistant Professor (S.G)

Information Technology
Rajalakshmi Engineering College

This project is submitted for CS23333 – Object Oriented Programming Using JAVA held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Table of Contents

1.1 Abstract:

The primary objective of the **ATM Machine Management System** in Java is to provide a robust, secure, and user-friendly platform that enables seamless banking transactions. This system allows users to perform essential banking operations such as cash withdrawal, deposit, account balance inquiry, and fund transfers efficiently. By utilizing Java's JDBC for database connectivity, the project ensures real-time interaction with the database, maintaining transaction accuracy and data security. Designed to mimic real-world ATM functionality, this project emphasizes reliability, scalability, and ease of use, serving as a foundation for understanding banking systems and financial software development.

1.2 Introduction:

The **ATM Machine Management System** is an advanced software application designed to simulate real-world banking transactions with enhanced efficiency and security. This system introduces automation to streamline essential banking operations, reducing manual intervention and operational complexities. Secured by robust authentication mechanisms, it ensures that only authorized users can access sensitive functions such as cash withdrawal, deposit, balance inquiry, and fund transfers. By leveraging Java and MySQL, the system provides a user-friendly interface for seamless interactions. With a focus on reliability and scalability, this system offers a modern and efficient solution to meet the dynamic demands of digital banking in today's era.

1.3 Purpose:

The purpose of the ATM Machine Management System is to develop a secure and user-friendly platform for performing essential banking operations such as cash withdrawal, deposits, balance inquiries, and fund transfers. The system aims to streamline banking processes while ensuring a seamless and secure user experience. By utilizing Java, MySQL, and JDBC, the project ensures efficient integration between the frontend and backend, enabling real-time transaction processing and data management. With its emphasis on modern technologies, the system provides a reliable solution for the growing demands of digital banking, while also establishing a foundation for future scalability and feature enhancements.

1.4 Scope of the project:

The scope of the **ATM Machine Management System** includes the development of a fully functional simulation of an ATM, enabling users to perform essential banking transactions. The system will provide features such as cash withdrawal, deposits, balance inquiries, and fund transfers, ensuring a secure and intuitive user experience. The backend, powered by MySQL, will manage user authentication, account information, and transaction records. JDBC will facilitate seamless communication between the Java-based frontend and the MySQL database, ensuring real-time transaction processing and data integrity.

1.5 Software Requirement Specification:

Introduction:

The **ATM Machine Management System** is a comprehensive software solution designed to replicate real-world ATM functionalities for efficient banking operations. This system allows users to perform essential tasks such as cash withdrawals, deposits, fund transfers, and balance inquiries in a secure and user-friendly environment. By leveraging Java for the application frontend and MySQL for database management, the system ensures seamless communication and robust data handling through JDBC.

Document Purpose:

This Software Requirements Specification (SRS) document provides a comprehensive outline of the ATM Machine Management System's requirements. It includes design considerations, architectural structure, and technical specifications, ensuring efficient implementation and long-term maintainability.

Product Scope:

The ATM Machine Management System aims to provide users with a platform to perform essential banking operations like cash withdrawals, deposits, balance inquiries, and fund transfers. It features user account management, transaction records, and a secure backend powered by MySQL for real-time data storage and processing. By utilizing Java for the frontend and JDBC for database connectivity, the system ensures an efficient and secure user experience.

References and Acknowledgement:

- Java AWT Tutorial - javatpoint
- Java Swing Guide - javatpoint
- SQL Tutorial - W3Schools
- Introduction to JDBC - GeeksforGeeks

Product Perspective:

The ATM Machine Management System is designed as a standalone application that integrates a Java-based frontend and a MySQL backend. The system simulates real-world ATM functionalities, such as secure transaction processing and user authentication, while leveraging JDBC for seamless database connectivity. This application is intended to serve as a model for modern banking operations, focusing on security, scalability, and ease of use.

Product Functionality:

- **User Login:** Secure login functionality for users with authentication.
- **Cash Withdrawal:** Allows users to withdraw money while validating available account balance.
- **Cash Deposit:** Enables users to deposit cash into their accounts securely.
- **Balance Inquiry:** Provides a quick view of the user's account balance.
- **Fund Transfer:** Facilitates secure transfers between user accounts.
- **Transaction History:** Displays a list of recent transactions for user reference.
- **Admin Module:** Provides administrative functionalities, such as managing user accounts and resetting passwords.

User and Characteristics:

Qualifications: Basic literacy and familiarity with banking operations.

Experience: Users should have prior experience with ATM systems or basic knowledge of bank transactions.

Technical Knowledge: Basic computer skills are sufficient to interact with the system effectively.

Operating Environment:

Hardware Requirements:

- **Processor:** Any Processor above i3.
- **Operating System:** Windows 8, 10, 11.
- **Processor Speed:** Minimum 2.0 GHz.
- **RAM:** 4 GB.
- **Hard Disk:** 500 GB.

Software Requirements:

- **Database:** MySQL.
- **Frontend:** Java (Swing, AWT).
- **Technology:** Java (JDBC).

Constraints:

- Transactions are limited to authorized users with verified credentials.
- Administrative operations, like account deletion, must be performed cautiously to maintain data integrity.

Assumptions and Dependencies:

- Admins create user accounts and securely distribute credentials.
- The system assumes consistent power supply and network connectivity for reliable operation.

Specific Requirements:

User Interface:

The ATM Machine Management System offers a user-friendly interface for:

- **User Login:** Securely log in with username and password.
- **Cash Withdrawal:** Withdraw amounts within available balance limits.
- **Cash Deposit:** Add funds to the user account securely.
- **Fund Transfer:** Transfer money to other accounts.
- **Transaction History:** View detailed logs of account activity.

Hardware Interface:

- Screen resolution: Minimum 640x480.
- Compatible with Windows 8, 10, 11.

Software Interface:

- **Operating System:** MS-Windows.
- **Frontend Development:** Java (AWT, Swing).
- **Backend Database:** MySQL.
- **Platform:** Java Language.
- **IDE:** NetBeans or VS Code.

Functional Requirements:

1. User Authentication Module (UAM):

- Supports secure login with username and masked passwords.
- Verifies credentials with database before granting access.

2. Transaction Module (TM):

- Handles cash withdrawals, deposits, and balance inquiries.
- Validates account balance for withdrawal operations.

3. Admin Module (AM):

- Allows administrators to manage user accounts and reset credentials.
- Facilitates audit of transactions for fraud detection.

4. Server Module (SM):

- Ensures secure communication between frontend and database.
- Processes transaction requests in real time.

Non-Functional Requirements:

Performance:

- All operations should be processed within 2 seconds under normal conditions.

Reliability:

- High uptime is required, with immediate restoration measures in case of failures.

Availability:

- The system must be accessible 24/7 for seamless banking operations.

Security:

- Implements robust encryption for sensitive data like passwords and transaction details.

Maintainability:

- Regular updates and backups must be ensured for continued reliability.

2. System Flow Diagram:

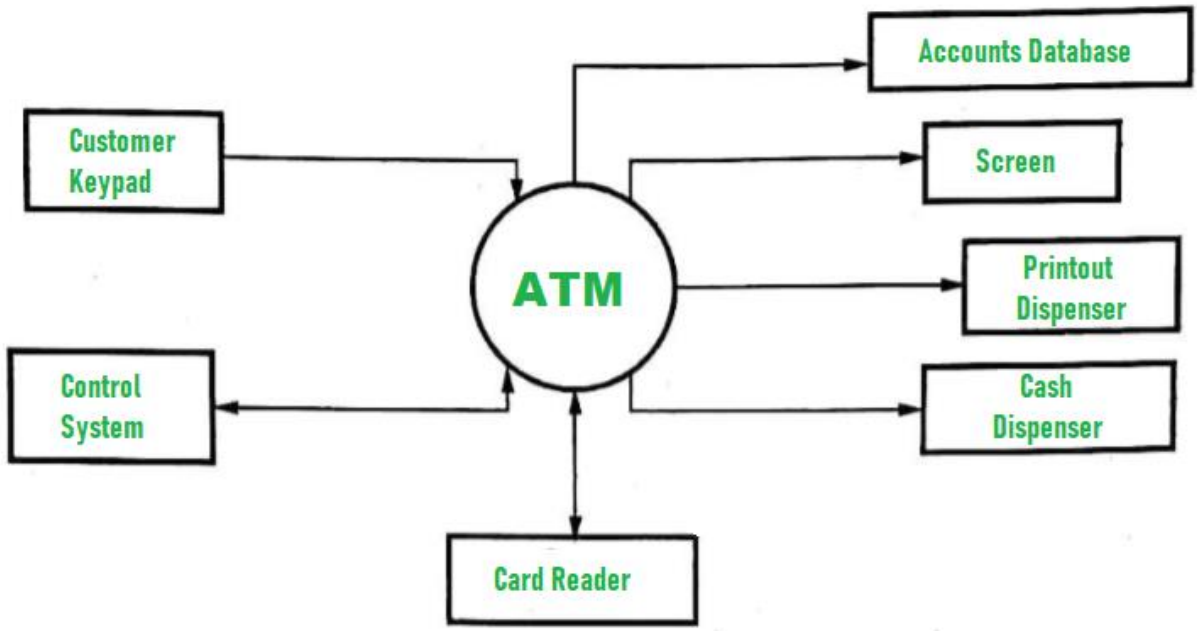

FIGURE 2.1 USE CASE DIAGRAM

2.2 Entity Relationship Diagram:

E-R (Entity-Relationship) Diagram is used to represent the relationships between entities in the table.

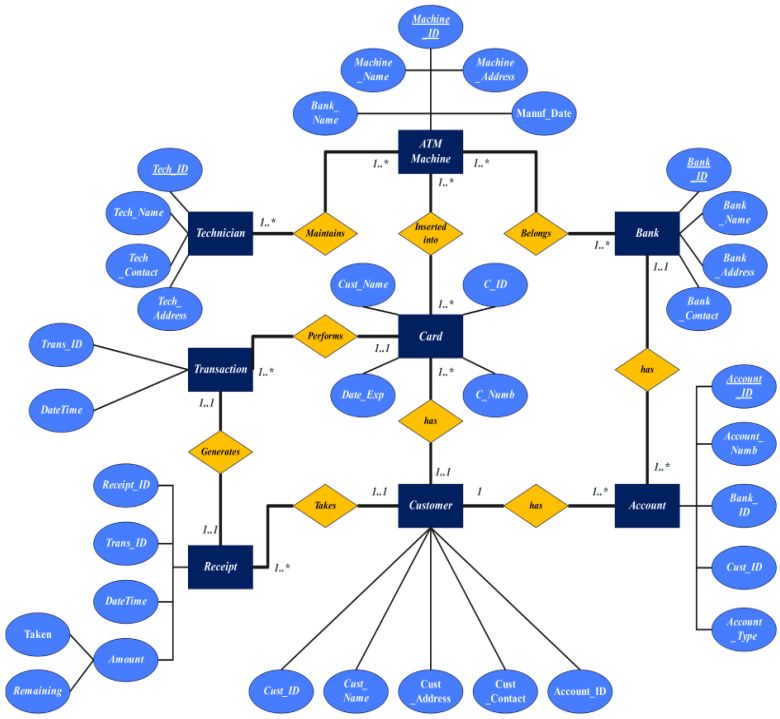

FIGURE 2.2 Entity-relationship diagram

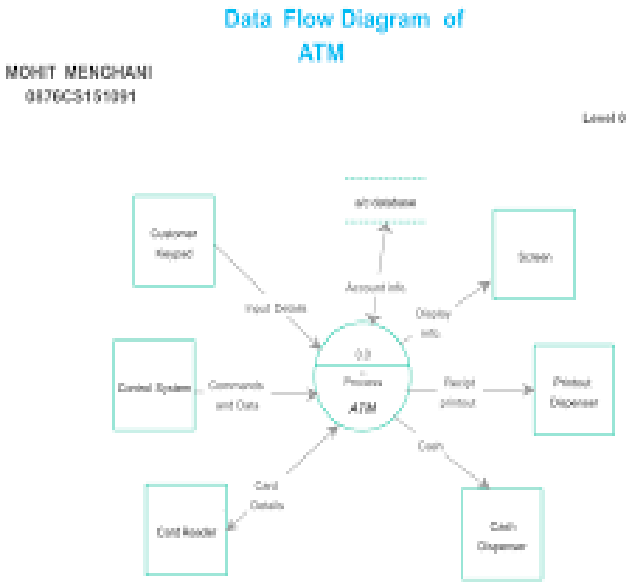

FIGURE 2.3 Data-flow diagram

Module Description:

Register:

The admin can register by providing a username and password for secure account creation.

Login:

The admin can log in using their registered username and password.

After Login:

1.Cash Withdrawal

The admin can simulate user withdrawals by inputting the desired amount. The system checks the account balance and processes the transaction if sufficient funds are available.

2. Cash Deposit

The admin can deposit funds into a user's account by entering the amount to be added. The system updates the account balance accordingly.

3. Balance Inquiry

The admin can view account details, including the current account balance and recent transactions.

4. Fund Transfer

The admin can facilitate fund transfers between user accounts by specifying the source account, destination account, and transfer amount.

5.Transaction History

The admin can view a log of all transactions, including withdrawals, deposits, and fund transfers, for audit and record-keeping purposes.

6. Manage Accounts

The admin can add, update, or delete user account details, ensuring the database is accurate and up-to-date

4.Implementation:

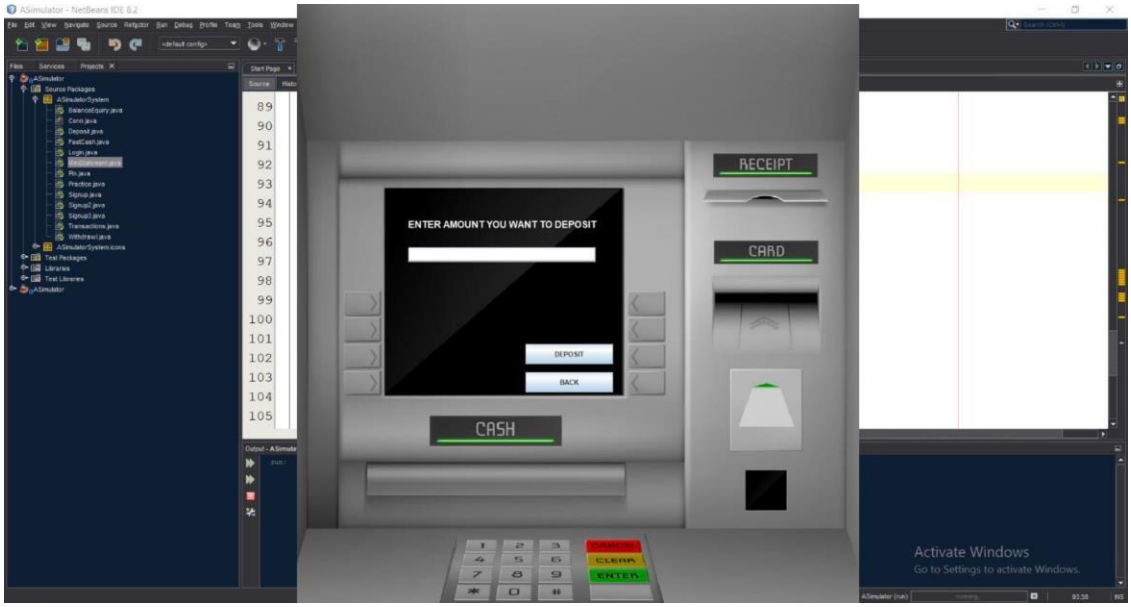

Figure 4.1 Home page

FIGURE 4.2 TRANSACTION TYPE

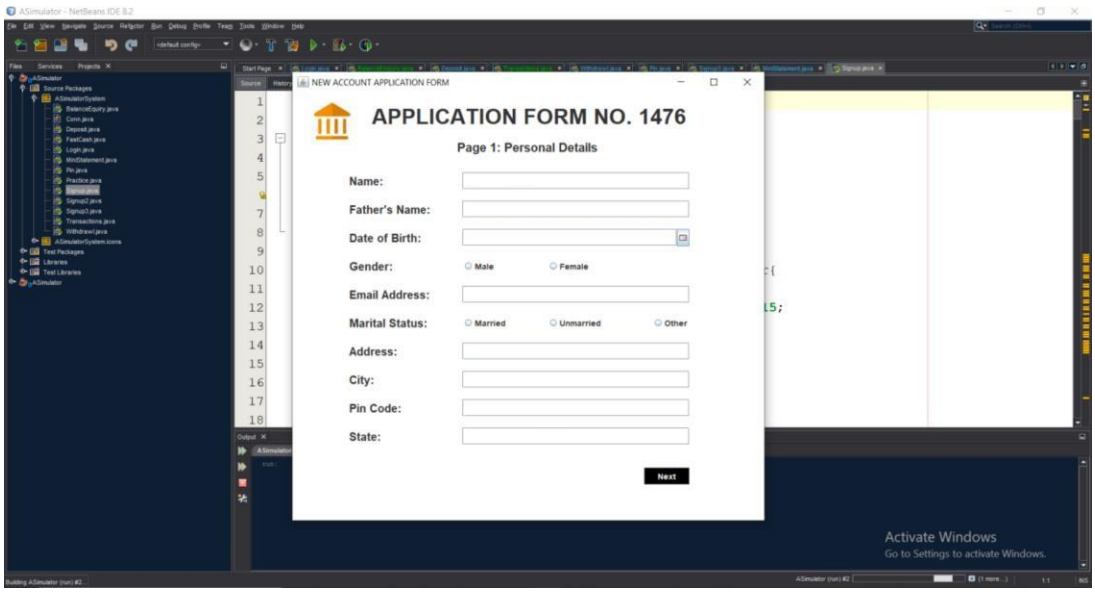

Figure 4.2 Application Form

4.2 Database Design

The database design is a fundamental aspect of the system, focusing on structuring and organizing data efficiently for the ATM Machine Management System. It ensures seamless storage, retrieval, and updating of critical information like user accounts, transaction records, and account balances. A well-structured database minimizes redundancy and inconsistencies through the process of normalization, thereby optimizing performance and maintaining data integrity.

MySQL has been chosen as the database management system for this project due to its reliability, scalability, and ease of integration with Java using JDBC. MySQL provides robust tools for managing relational data while ensuring secure and efficient transactions. Its high performance and strong community support make it an ideal choice for applications like ATM systems, where real-time processing and data security are paramount. By utilizing MySQL, the system achieves a scalable and cost-effective database solution tailored to meet the demands of modern financial operations.

Source Code

Login.java

```
package bank.management.system;

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

public class Login extends JFrame implements ActionListener{
    JLabel l1,l2,l3;
    JTextField tf1;
    JPasswordField pf2;
    JButton b1,b2,b3;

    Login(){
        setTitle("AUTOMATED TELLER MACHINE");

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
logo.jpg"));
        Image i2 = i1.getImage().getScaledInstance(100, 100, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l11 = new JLabel(i3);
        l11.setBounds(70, 10, 100, 100);
        add(l11);

        l1 = new JLabel("WELCOME TO ATM");
        l1.setFont(new Font("Osward", Font.BOLD, 38));
        l1.setBounds(200,40,450,40);
        add(l1);

        l2 = new JLabel("Card No:");
        l2.setFont(new Font("Raleway", Font.BOLD, 28));
        l2.setBounds(125,150,375,30);
        add(l2);

        tf1 = new JTextField(15);
        tf1.setBounds(300,150,230,30);
        tf1.setFont(new Font("Arial", Font.BOLD, 14));
        add(tf1);

        l3 = new JLabel("PIN:");
        l3.setFont(new Font("Raleway", Font.BOLD, 28));
        l3.setBounds(125,220,375,30);
        add(l3);

        pf2 = new JPasswordField(15);
        pf2.setFont(new Font("Arial", Font.BOLD, 14));
        pf2.setBounds(300,220,230,30);
        add(pf2);

        b1 = new JButton("SIGN IN");
        b1.setBackground(Color.BLACK);
        b1.setForeground(Color.WHITE);
```

```
    b2 = new JButton("CLEAR");
    b2.setBackground(Color.BLACK);
    b2.setForeground(Color.WHITE);

    b3 = new JButton("SIGN UP");
    b3.setBackground(Color.BLACK);
    b3.setForeground(Color.WHITE);

    setLayout(null);

    b1.setFont(new Font("Arial", Font.BOLD, 14));
    b1.setBounds(300,300,100,30);
    add(b1);

    b2.setFont(new Font("Arial", Font.BOLD, 14));
    b2.setBounds(430,300,100,30);
    add(b2);

    b3.setFont(new Font("Arial", Font.BOLD, 14));
    b3.setBounds(300,350,230,30);
    add(b3);

    b1.addActionListener(this);
    b2.addActionListener(this);
    b3.addActionListener(this);

    getContentPane().setBackground(Color.WHITE);

    setSize(800,480);
    setLocation(550,200);
    setVisible(true);

}
public void actionPerformed(ActionEvent ae){
    try{
        if(ae.getSource()==b1){
            Conn c1 = new Conn();
            String cardno  = tf1.getText();
            String pin  = pf2.getText();
            String q  = "select * from login where cardno = '"+cardno+"' and pin = '"+pin+"'";

            ResultSet rs = c1.s.executeQuery(q);
            if(rs.next()){
                setVisible(false);
                new Transactions(pin).setVisible(true);
            }else{
                JOptionPane.showMessageDialog(null, "Incorrect Card Number or PIN");
            }
        }else if(ae.getSource()==b2){
            tf1.setText("");
            pf2.setText("");
        }else if(ae.getSource()==b3){
            setVisible(false);
            new Signup().setVisible(true);
        }
    }catch(Exception e){
        e.printStackTrace();
    }
}
public static void main(String[] args){
    new Login().setVisible(true);
}
```

```
}
```

Signup.java

```
package bank.management.system;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import com.toedter.calendar.JDateChooser;
import java.util.*;

public class Signup extends JFrame implements ActionListener{

    JLabel l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15;
    JTextField t1,t2,t3,t4,t5,t6,t7;
    JRadioButton r1,r2,r3,r4,r5;
    JButton b;
    JDateChooser dateChooser;


    Random ran = new Random();
    long first4 = (ran.nextLong() % 9000L) + 1000L;
    String first = "" + Math.abs(first4);

    Signup(){

        setTitle("NEW ACCOUNT APPLICATION FORM");

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
logo.jpg"));
        Image i2 = i1.getImage().getScaledInstance(100, 100, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l11 = new JLabel(i3);
        l11.setBounds(20, 0, 100, 100);
        add(l11);

        l1 = new JLabel("APPLICATION FORM NO. "+first);
        l1.setFont(new Font("Raleway", Font.BOLD, 38));

        l2 = new JLabel("Page 1: Personal Details");
        l2.setFont(new Font("Raleway", Font.BOLD, 22));

        l3 = new JLabel("Name:");
        l3.setFont(new Font("Raleway", Font.BOLD, 20));

        l4 = new JLabel("Father's Name:");
        l4.setFont(new Font("Raleway", Font.BOLD, 20));

        l5 = new JLabel("Date of Birth:");
        l5.setFont(new Font("Raleway", Font.BOLD, 20));
```

```
l6 = new JLabel("Gender:");
l6.setFont(new Font("Raleway", Font.BOLD, 20));

l7 = new JLabel("Email Address:");
l7.setFont(new Font("Raleway", Font.BOLD, 20));

l8 = new JLabel("Marital Status:");
l8.setFont(new Font("Raleway", Font.BOLD, 20));

l9 = new JLabel("Address:");
l9.setFont(new Font("Raleway", Font.BOLD, 20));

l10 = new JLabel("City:");
l10.setFont(new Font("Raleway", Font.BOLD, 20));

l11 = new JLabel("Pin Code:");
l11.setFont(new Font("Raleway", Font.BOLD, 20));

l12 = new JLabel("State:");
l12.setFont(new Font("Raleway", Font.BOLD, 20));

l13 = new JLabel("Date");
l13.setFont(new Font("Raleway", Font.BOLD, 14));

l14 = new JLabel("Month");
l14.setFont(new Font("Raleway", Font.BOLD, 14));

l15 = new JLabel("Year");
l15.setFont(new Font("Raleway", Font.BOLD, 14));


t1 = new JTextField();
t1.setFont(new Font("Raleway", Font.BOLD, 14));

t2 = new JTextField();
t2.setFont(new Font("Raleway", Font.BOLD, 14));

t3 = new JTextField();
t3.setFont(new Font("Raleway", Font.BOLD, 14));

t4 = new JTextField();
t4.setFont(new Font("Raleway", Font.BOLD, 14));

t5 = new JTextField();
t5.setFont(new Font("Raleway", Font.BOLD, 14));

t6 = new JTextField();
t6.setFont(new Font("Raleway", Font.BOLD, 14));

t7 = new JTextField();
t7.setFont(new Font("Raleway", Font.BOLD, 14));



b = new JButton("Next");
b.setFont(new Font("Raleway", Font.BOLD, 14));
b.setBackground(Color.BLACK);
b.setForeground(Color.WHITE);

r1 = new JRadioButton("Male");
r1.setFont(new Font("Raleway", Font.BOLD, 14));
```

```
r1.setBackground(Color.WHITE);

r2 = new JRadioButton("Female");
r2.setFont(new Font("Raleway", Font.BOLD, 14));
r2.setBackground(Color.WHITE);

ButtonGroup groupgender = new ButtonGroup();
groupgender.add(r1);
groupgender.add(r2);

r3 =  new JRadioButton("Married");
r3.setFont(new Font("Raleway", Font.BOLD, 14));
r3.setBackground(Color.WHITE);

r4 = new JRadioButton("Unmarried");
r4.setFont(new Font("Raleway", Font.BOLD, 14));
r4.setBackground(Color.WHITE);

r5 = new JRadioButton("Other");
r5.setFont(new Font("Raleway", Font.BOLD, 14));
r5.setBackground(Color.WHITE);

ButtonGroup groupstatus = new ButtonGroup();
groupstatus.add(r3);
groupstatus.add(r4);
groupstatus.add(r5);

dateChooser = new JDateChooser();
    //dateChooser.setBorder(new LineBorder(new Color(0, 0, 0), 1, true));
    dateChooser.setForeground(new Color(105, 105, 105));
    dateChooser.setBounds(137, 337, 200, 29);
    add(dateChooser);

setLayout(null);
l1.setBounds(140,20,600,40);
add(l1);

l2.setBounds(290,80,600,30);
add(l2);

l3.setBounds(100,140,100,30);
add(l3);

t1.setBounds(300,140,400,30);
add(t1);

l4.setBounds(100,190,200,30);
add(l4);

t2.setBounds(300,190,400,30);
add(t2);

l5.setBounds(100,240,200,30);
add(l5);

dateChooser.setBounds(300, 240, 400, 30);

l6.setBounds(100,290,200,30);
add(l6);

r1.setBounds(300,290,60,30);
```

```
    add(r1);

    r2.setBounds(450,290,90,30);
    add(r2);

    l7.setBounds(100,340,200,30);
    add(l7);

    t3.setBounds(300,340,400,30);
    add(t3);

    l8.setBounds(100,390,200,30);
    add(l8);

    r3.setBounds(300,390,100,30);
    add(r3);

    r4.setBounds(450,390,100,30);
    add(r4);

    r5.setBounds(635,390,100,30);
    add(r5);



    l9.setBounds(100,440,200,30);
    add(l9);

    t4.setBounds(300,440,400,30);
    add(t4);

    l10.setBounds(100,490,200,30);
    add(l10);

    t5.setBounds(300,490,400,30);
    add(t5);

    l11.setBounds(100,540,200,30);
    add(l11);

    t6.setBounds(300,540,400,30);
    add(t6);

    l12.setBounds(100,590,200,30);
    add(l12);

    t7.setBounds(300,590,400,30);
    add(t7);

    b.setBounds(620,660,80,30);
    add(b);

    b.addActionListener(this);

    getContentPane().setBackground(Color.WHITE);

    setSize(850,800);
    setLocation(500,120);
    setVisible(true);
}
```

```
public void actionPerformed(ActionEvent ae){

        String formno =  first;
        String name = t1.getText();
        String fname = t2.getText();
        String dob = ((JTextField) dateChooser.getDateEditor().getUiComponent()).getText();
        String gender = null;
        if(r1.isSelected()){
           gender = "Male";
        }else if(r2.isSelected()){
           gender = "Female";
        }

        String email = t3.getText();
        String marital = null;
        if(r3.isSelected()){
           marital = "Married";
        }else if(r4.isSelected()){
           marital = "Unmarried";
        }else if(r5.isSelected()){
           marital = "Other";
        }

        String address = t4.getText();
        String city = t5.getText();
        String pincode = t6.getText();
        String state = t7.getText();


        try{

           if(t6.getText().equals("")){
              JOptionPane.showMessageDialog(null, "Fill all the required fields");
           }else{
              Conn c1 = new Conn();
              String q1 = "insert into signup
values('"+formno+"','"+name+"','"+fname+"','"+dob+"','"+gender+"','"+email+"','"+marital+"','"+
address+"','"+city+"','"+pincode+"','"+state+"')";
              c1.s.executeUpdate(q1);

              new Signup2(first).setVisible(true);
              setVisible(false);
           }

        }catch(Exception e){
            e.printStackTrace();
        }

    }


    public static void main(String[] args){
        new Signup().setVisible(true);
    }
}
```

Signup2.java

```
package bank.management.system;

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

public class Signup2 extends JFrame implements ActionListener{

    JLabel l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13;
    JButton b;
    JRadioButton r1,r2,r3,r4;
    JTextField t1,t2,t3;
    JComboBox c1,c2,c3,c4,c5;
    String formno;
    Signup2(String formno){

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
logo.jpg"));
        Image i2 = i1.getImage().getScaledInstance(100, 100, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l14 = new JLabel(i3);
        l14.setBounds(150, 0, 100, 100);
        add(l14);


        this.formno = formno;
        setTitle("NEW ACCOUNT APPLICATION FORM - PAGE 2");

        l1 = new JLabel("Page 2: Additonal Details");
        l1.setFont(new Font("Raleway", Font.BOLD, 22));

        l2 = new JLabel("Religion:");
        l2.setFont(new Font("Raleway", Font.BOLD, 18));

        l3 = new JLabel("Category:");
        l3.setFont(new Font("Raleway", Font.BOLD, 18));

        l4 = new JLabel("Income:");
        l4.setFont(new Font("Raleway", Font.BOLD, 18));

        l5 = new JLabel("Educational");
        l5.setFont(new Font("Raleway", Font.BOLD, 18));

        l11 = new JLabel("Qualification:");
        l11.setFont(new Font("Raleway", Font.BOLD, 18));

        l6 = new JLabel("Occupation:");
        l6.setFont(new Font("Raleway", Font.BOLD, 18));

        l7 = new JLabel("PAN Number:");
        l7.setFont(new Font("Raleway", Font.BOLD, 18));

        l8 = new JLabel("Aadhar Number:");
        l8.setFont(new Font("Raleway", Font.BOLD, 18));

        l9 = new JLabel("Senior Citizen:");
        l9.setFont(new Font("Raleway", Font.BOLD, 18));

        l10 = new JLabel("Existing Account:");
```

```
        l10.setFont(new Font("Raleway", Font.BOLD, 18));

        l12 = new JLabel("Form No:");
        l12.setFont(new Font("Raleway", Font.BOLD, 13));

        l13 = new JLabel(formno);
        l13.setFont(new Font("Raleway", Font.BOLD, 13));

        b = new JButton("Next");
        b.setFont(new Font("Raleway", Font.BOLD, 14));
        b.setBackground(Color.BLACK);
        b.setForeground(Color.WHITE);

        t1 = new JTextField();
        t1.setFont(new Font("Raleway", Font.BOLD, 14));

        t2 = new JTextField();
        t2.setFont(new Font("Raleway", Font.BOLD, 14));

        r1 = new JRadioButton("Yes");
        r1.setFont(new Font("Raleway", Font.BOLD, 14));
        r1.setBackground(Color.WHITE);

        r2 = new JRadioButton("No");
        r2.setFont(new Font("Raleway", Font.BOLD, 14));
        r2.setBackground(Color.WHITE);

        r3 = new JRadioButton("Yes");
        r3.setFont(new Font("Raleway", Font.BOLD, 14));
        r3.setBackground(Color.WHITE);

        r4 = new JRadioButton("No");
        r4.setFont(new Font("Raleway", Font.BOLD, 14));
        r4.setBackground(Color.WHITE);


        String religion[] = {"Hindu","Muslim","Sikh","Christian","Other"};
        c1 = new JComboBox(religion);
        c1.setBackground(Color.WHITE);
        c1.setFont(new Font("Raleway", Font.BOLD, 14));

        String category[] = {"General","OBC","SC","ST","Other"};
        c2 = new JComboBox(category);
        c2.setBackground(Color.WHITE);
        c2.setFont(new Font("Raleway", Font.BOLD, 14));

        String income[] = {"Null","<1,50,000","<2,50,000","<5,00,000","Upto 10,00,000","Above
10,00,000"};
        c3 = new JComboBox(income);
        c3.setBackground(Color.WHITE);
        c3.setFont(new Font("Raleway", Font.BOLD, 14));

        String education[] = {"Non-Graduate","Graduate","Post-Graduate","Doctrate","Others"};
        c4 = new JComboBox(education);
        c4.setBackground(Color.WHITE);
        c4.setFont(new Font("Raleway", Font.BOLD, 14));

        String occupation[] = {"Salaried","Self-
Employmed","Business","Student","Retired","Others"};
        c5 = new JComboBox(occupation);
        c5.setBackground(Color.WHITE);
```

```
c5.setFont(new Font("Raleway", Font.BOLD, 14));


setLayout(null);


l12.setBounds(700,10,60,30);
add(l12);

l13.setBounds(760,10,60,30);
add(l13);

l1.setBounds(280,30,600,40);
add(l1);

l2.setBounds(100,120,100,30);
add(l2);

c1.setBounds(350,120,320,30);
add(c1);

l3.setBounds(100,170,100,30);
add(l3);

c2.setBounds(350,170,320,30);
add(c2);

l4.setBounds(100,220,100,30);
add(l4);

c3.setBounds(350,220,320,30);
add(c3);

l5.setBounds(100,270,150,30);
add(l5);

c4.setBounds(350,270,320,30);
add(c4);

l11.setBounds(100,290,150,30);
add(l11);

l6.setBounds(100,340,150,30);
add(l6);

c5.setBounds(350,340,320,30);
add(c5);

l7.setBounds(100,390,150,30);
add(l7);

t1.setBounds(350,390,320,30);
add(t1);

l8.setBounds(100,440,180,30);
add(l8);

t2.setBounds(350,440,320,30);
add(t2);

l9.setBounds(100,490,150,30);
```

```
        add(l9);

        r1.setBounds(350,490,100,30);
        add(r1);

        r2.setBounds(460,490,100,30);
        add(r2);

        l10.setBounds(100,540,180,30);
        add(l10);

        r3.setBounds(350,540,100,30);
        add(r3);

        r4.setBounds(460,540,100,30);
        add(r4);

        b.setBounds(570,640,100,30);
        add(b);

        b.addActionListener(this);

        getContentPane().setBackground(Color.WHITE);

        setSize(850,750);
        setLocation(500,120);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae){
        String religion = (String)c1.getSelectedItem();
        String category = (String)c2.getSelectedItem();
        String income = (String)c3.getSelectedItem();
        String education = (String)c4.getSelectedItem();
        String occupation = (String)c5.getSelectedItem();

        String pan = t1.getText();
        String aadhar = t2.getText();

        String scitizen = "";
        if(r1.isSelected()){
            scitizen = "Yes";
        }
        else if(r2.isSelected()){
            scitizen = "No";
        }

        String eaccount = "";
        if(r3.isSelected()){
            eaccount = "Yes";
        }else if(r4.isSelected()){
            eaccount = "No";
        }

        try{
            if(t2.getText().equals("")){
                JOptionPane.showMessageDialog(null, "Fill all the required fields");
            }else{
                Conn c1 = new Conn();
                String q1 = "insert into signup2
```

```
values('"+formno+"','"+religion+"','"+category+"','"+income+"','"+education+"','"+occupation+"',
'"+pan+"','"+aadhar+"','"+scitizen+"','"+eaccount+"')";
            c1.s.executeUpdate(q1);

            new Signup3(formno).setVisible(true);
            setVisible(false);
          }



       }catch(Exception ex){
           ex.printStackTrace();
       }


   }


   public static void main(String[] args){
       new Signup2("").setVisible(true);
   }
}
```

Signup3.java

```
package bank.management.system;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.util.*;

public class Signup3 extends JFrame implements ActionListener{

   JLabel l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12;
   JRadioButton r1,r2,r3,r4;
   JButton b1,b2;
   JCheckBox c1,c2,c3,c4,c5,c6,c7;
   String formno;
   Signup3(String formno){
       this.formno = formno;
       setTitle("NEW ACCOUNT APPLICATION FORM - PAGE 3");

       ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
logo.jpg"));
       Image i2 = i1.getImage().getScaledInstance(100, 100, Image.SCALE_DEFAULT);
       ImageIcon i3 = new ImageIcon(i2);
       JLabel l14 = new JLabel(i3);
       l14.setBounds(150, 0, 100, 100);
       add(l14);

       l1 = new JLabel("Page 3: Account Details");
       l1.setFont(new Font("Raleway", Font.BOLD, 22));

       l2 = new JLabel("Account Type:");
```

```
l2.setFont(new Font("Raleway", Font.BOLD, 18));

l3 = new JLabel("Card Number:");
l3.setFont(new Font("Raleway", Font.BOLD, 18));

l4 = new JLabel("XXXX-XXXX-XXXX-4184");
l4.setFont(new Font("Raleway", Font.BOLD, 18));

l5 = new JLabel("(Your 16-digit Card number)");
l5.setFont(new Font("Raleway", Font.BOLD, 12));

l6 = new JLabel("It would appear on ATM Card/Cheque Book and Statements");
l6.setFont(new Font("Raleway", Font.BOLD, 12));

l7 = new JLabel("PIN:");
l7.setFont(new Font("Raleway", Font.BOLD, 18));

l8 = new JLabel("XXXX");
l8.setFont(new Font("Raleway", Font.BOLD, 18));

l9 = new JLabel("(4-digit password)");
l9.setFont(new Font("Raleway", Font.BOLD, 12));

l10 = new JLabel("Services Required:");
l10.setFont(new Font("Raleway", Font.BOLD, 18));

l11 = new JLabel("Form No:");
l11.setFont(new Font("Raleway", Font.BOLD, 14));

l12 = new JLabel(formno);
l12.setFont(new Font("Raleway", Font.BOLD, 14));

b1 = new JButton("Submit");
b1.setFont(new Font("Raleway", Font.BOLD, 14));
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);

b2 = new JButton("Cancel");
b2.setFont(new Font("Raleway", Font.BOLD, 14));
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);


c1 = new JCheckBox("ATM CARD");
c1.setBackground(Color.WHITE);
c1.setFont(new Font("Raleway", Font.BOLD, 16));

c2 = new JCheckBox("Internet Banking");
c2.setBackground(Color.WHITE);
c2.setFont(new Font("Raleway", Font.BOLD, 16));

c3 = new JCheckBox("Mobile Banking");
c3.setBackground(Color.WHITE);
c3.setFont(new Font("Raleway", Font.BOLD, 16));

c4 = new JCheckBox("EMAIL Alerts");
c4.setBackground(Color.WHITE);
c4.setFont(new Font("Raleway", Font.BOLD, 16));

c5 = new JCheckBox("Cheque Book");
c5.setBackground(Color.WHITE);
```

```
        c5.setFont(new Font("Raleway", Font.BOLD, 16));

        c6 = new JCheckBox("E-Statement");
        c6.setBackground(Color.WHITE);
        c6.setFont(new Font("Raleway", Font.BOLD, 16));

        c7 = new JCheckBox("I hereby declares that the above entered details correct to th best of
my knowledge.",true);
        c7.setBackground(Color.WHITE);
        c7.setFont(new Font("Raleway", Font.BOLD, 12));


        r1 = new JRadioButton("Saving Account");
        r1.setFont(new Font("Raleway", Font.BOLD, 16));
        r1.setBackground(Color.WHITE);

        r2 = new JRadioButton("Fixed Deposit Account");
        r2.setFont(new Font("Raleway", Font.BOLD, 16));
        r2.setBackground(Color.WHITE);

        r3 = new JRadioButton("Current Account");
        r3.setFont(new Font("Raleway", Font.BOLD, 16));
        r3.setBackground(Color.WHITE);

        r4 = new JRadioButton("Recurring Deposit Account");
        r4.setFont(new Font("Raleway", Font.BOLD, 16));
        r4.setBackground(Color.WHITE);

        ButtonGroup groupgender = new ButtonGroup();
        groupgender.add(r1);
        groupgender.add(r2);
        groupgender.add(r3);
        groupgender.add(r4);

        setLayout(null);

        l11.setBounds(700,10,70,30);
        add(l11);

        l12.setBounds(770,10,40,30);
        add(l12);

        l1.setBounds(280,40,400,40);
        add(l1);

        l2.setBounds(100,140,200,30);
        add(l2);

        r1.setBounds(100,180,150,30);
        add(r1);

        r2.setBounds(350,180,300,30);
        add(r2);

        r3.setBounds(100,220,250,30);
        add(r3);

        r4.setBounds(350,220,250,30);
        add(r4);

        l3.setBounds(100,300,200,30);
```

```
    add(l3);

    l4.setBounds(330,300,250,30);
    add(l4);

    l5.setBounds(100,330,200,20);
    add(l5);

    l6.setBounds(330,330,500,20);
    add(l6);

    l7.setBounds(100,370,200,30);
    add(l7);

    l8.setBounds(330,370,200,30);
    add(l8);

    l9.setBounds(100,400,200,20);
    add(l9);

    l10.setBounds(100,450,200,30);
    add(l10);

    c1.setBounds(100,500,200,30);
    add(c1);

    c2.setBounds(350,500,200,30);
    add(c2);

    c3.setBounds(100,550,200,30);
    add(c3);

    c4.setBounds(350,550,200,30);
    add(c4);

    c5.setBounds(100,600,200,30);
    add(c5);

    c6.setBounds(350,600,200,30);
    add(c6);

    c7.setBounds(100,680,600,20);
    add(c7);

    b1.setBounds(250,720,100,30);
    add(b1);

    b2.setBounds(420,720,100,30);
    add(b2);


    getContentPane().setBackground(Color.WHITE);

    setSize(850,850);
    setLocation(500,120);
    setVisible(true);

    b1.addActionListener(this);
    b2.addActionListener(this);

}
```

```
  public void actionPerformed(ActionEvent ae){
      String atype = null;
      if(r1.isSelected()){
         atype = "Saving Account";
      }
      else if(r2.isSelected()){
         atype = "Fixed Deposit Account";
      }
      else if(r3.isSelected()){
         atype = "Current Account";
      }else if(r4.isSelected()){
         atype = "Recurring Deposit Account";
      }

      Random ran = new Random();
      long first7 = (ran.nextLong() % 90000000L) + 5040936000000000L;
      String cardno = "" + Math.abs(first7);

      long first3 = (ran.nextLong() % 9000L) + 1000L;
      String pin = "" + Math.abs(first3);

      String facility = "";
      if(c1.isSelected()){
         facility = facility + " ATM Card";
      }
      if(c2.isSelected()){
         facility = facility + " Internet Banking";
      }
      if(c3.isSelected()){
         facility = facility + " Mobile Banking";
      }
      if(c4.isSelected()){
         facility = facility + " EMAIL Alerts";
      }
      if(c5.isSelected()){
         facility = facility + " Cheque Book";
      }
      if(c6.isSelected()){
         facility = facility + " E-Statement";
      }

      try{
         if(ae.getSource()==b1){

            if(atype.equals("")){
               JOptionPane.showMessageDialog(null, "Fill all the required fields");
            }else{
               Conn c1 = new Conn();
               String q1 = "insert into signup3
values('"+formno+"','"+atype+"','"+cardno+"','"+pin+"','"+facility+"')";
               String q2 = "insert into login values('"+formno+"','"+cardno+"','"+pin+"')";
               c1.s.executeUpdate(q1);
               c1.s.executeUpdate(q2);
               JOptionPane.showMessageDialog(null, "Card Number: " + cardno + "\n Pin:"+ pin);

               new Deposit(pin).setVisible(true);
               setVisible(false);
            }

         }else if(ae.getSource()==b2){
```

```
            System.exit(0);
        }

      }catch(Exception ex){
         ex.printStackTrace();
      }

   }

   public static void main(String[] args){
      new Signup3("").setVisible(true);
   }

}
```

Deposit.java

```
package bank.management.system;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;

public class Deposit extends JFrame implements ActionListener{

   JTextField t1,t2;
   JButton b1,b2,b3;
   JLabel l1,l2,l3;
   String pin;
   Deposit(String pin){
      this.pin = pin;
      ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
atm.jpg"));
      Image i2 = i1.getImage().getScaledInstance(1000, 1180, Image.SCALE_DEFAULT);
      ImageIcon i3 = new ImageIcon(i2);
      JLabel l3 = new JLabel(i3);
      l3.setBounds(0, 0, 960, 1080);
      add(l3);

      l1 = new JLabel("ENTER AMOUNT YOU WANT TO DEPOSIT");
      l1.setForeground(Color.WHITE);
      l1.setFont(new Font("System", Font.BOLD, 16));

      t1 = new JTextField();
      t1.setFont(new Font("Raleway", Font.BOLD, 22));

      b1 = new JButton("DEPOSIT");
      b2 = new JButton("BACK");

      setLayout(null);

      l1.setBounds(190,350,400,35);
      l3.add(l1);

      t1.setBounds(190,420,320,25);
      l3.add(t1);
```

```
        b1.setBounds(390,588,150,35);
        l3.add(b1);

        b2.setBounds(390,633,150,35);
        l3.add(b2);

        b1.addActionListener(this);
        b2.addActionListener(this);

        setSize(960,1080);
        setUndecorated(true);
        setLocation(500,0);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae){
        try{
            String amount = t1.getText();
            Date date = new Date();
            if(ae.getSource()==b1){
                if(t1.getText().equals("")){
                    JOptionPane.showMessageDialog(null, "Please enter the Amount to you want to
Deposit");
                }else{
                    Conn c1 = new Conn();
                    c1.s.executeUpdate("insert into bank values('"+pin+"', '"+date+"', 'Deposit',
'"+amount+"')");
                    JOptionPane.showMessageDialog(null, "Rs. "+amount+" Deposited Successfully");
                    setVisible(false);
                    new Transactions(pin).setVisible(true);
                }
            }else if(ae.getSource()==b2){
                setVisible(false);
                new Transactions(pin).setVisible(true);
            }
        }catch(Exception e){
            e.printStackTrace();
        }

    }

    public static void main(String[] args){
        new Deposit("").setVisible(true);
    }
}
```

Conn.java

```
package bank.management.system;

import java.sql.*;

public class Conn{
    Connection c;
    Statement s;
    public Conn(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
```

```
            c =DriverManager.getConnection("jdbc:mysql:///
bankmanagementsystem","root","kharthik@123");
            s =c.createStatement();



        }catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Pin.java

```
package bank.management.system;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;

public class Pin extends JFrame implements ActionListener{

    JPasswordField t1,t2;
    JButton b1,b2;
    JLabel l1,l2,l3;
    String pin;
    Pin(String pin){
        this.pin = pin;
        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
atm.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1000, 1180, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l4 = new JLabel(i3);
        l4.setBounds(0, 0, 960, 1080);
        add(l4);

        l1 = new JLabel("CHANGE YOUR PIN");
        l1.setFont(new Font("System", Font.BOLD, 16));
        l1.setForeground(Color.WHITE);

        l2 = new JLabel("New PIN:");
        l2.setFont(new Font("System", Font.BOLD, 16));
        l2.setForeground(Color.WHITE);

        l3 = new JLabel("Re-Enter New PIN:");
        l3.setFont(new Font("System", Font.BOLD, 16));
        l3.setForeground(Color.WHITE);

        t1 = new JPasswordField();
        t1.setFont(new Font("Raleway", Font.BOLD, 25));

        t2 = new JPasswordField();
        t2.setFont(new Font("Raleway", Font.BOLD, 25));

        b1 = new JButton("CHANGE");
        b2 = new JButton("BACK");

        b1.addActionListener(this);
        b2.addActionListener(this);
```

```
    setLayout(null);

    l1.setBounds(280,330,800,35);
    l4.add(l1);

    l2.setBounds(180,390,150,35);
    l4.add(l2);

    l3.setBounds(180,440,200,35);
    l4.add(l3);

    t1.setBounds(350,390,180,25);
    l4.add(t1);

    t2.setBounds(350,440,180,25);
    l4.add(t2);

    b1.setBounds(390,588,150,35);
    l4.add(b1);

    b2.setBounds(390,633,150,35);
    l4.add(b2);

    setSize(960,1080);
    setLocation(500,0);
    setUndecorated(true);
    setVisible(true);

}

public void actionPerformed(ActionEvent ae){
    try{
        String npin = t1.getText();
        String rpin = t2.getText();

        if(!npin.equals(rpin)){
            JOptionPane.showMessageDialog(null, "Entered PIN does not match");
            return;
        }

        if(ae.getSource()==b1){
            if (t1.getText().equals("")){
                JOptionPane.showMessageDialog(null, "Enter New PIN");
            }
            if (t2.getText().equals("")){
                JOptionPane.showMessageDialog(null, "Re-Enter new PIN");
            }

            Conn c1 = new Conn();
            String q1 = "update bank set pin = '"+rpin+"' where pin = '"+pin+"' ";
            String q2 = "update login set pin = '"+rpin+"' where pin = '"+pin+"' ";
            String q3 = "update signup3 set pin = '"+rpin+"' where pin = '"+pin+"' ";

            c1.s.executeUpdate(q1);
            c1.s.executeUpdate(q2);
            c1.s.executeUpdate(q3);

            JOptionPane.showMessageDialog(null, "PIN changed successfully");
            setVisible(false);
            new Transactions(rpin).setVisible(true);
```

```
            }else if(ae.getSource()==b2){
              new Transactions(pin).setVisible(true);
              setVisible(false);
            }
        }catch(Exception e){
            e.printStackTrace();
        }
    }

    public static void main(String[] args){
        new Pin("").setVisible(true);
    }
}
```

FastCash.java

```
package bank.management.system;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.util.Date;

public class FastCash extends JFrame implements ActionListener {

    JLabel l1, l2;
    JButton b1, b2, b3, b4, b5, b6, b7, b8;
    JTextField t1;
    String pin;

    FastCash(String pin) {
        this.pin = pin;
        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
atm.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1000, 1180, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l3 = new JLabel(i3);
        l3.setBounds(0, 0, 960, 1080);
        add(l3);

        l1 = new JLabel("SELECT WITHDRAWL AMOUNT");
        l1.setForeground(Color.WHITE);
        l1.setFont(new Font("System", Font.BOLD, 16));

        b1 = new JButton("Rs 100");
        b2 = new JButton("Rs 500");
        b3 = new JButton("Rs 1000");
        b4 = new JButton("Rs 2000");
        b5 = new JButton("Rs 5000");
        b6 = new JButton("Rs 10000");
        b7 = new JButton("BACK");

        setLayout(null);

        l1.setBounds(235, 400, 700, 35);
        l3.add(l1);
```

```
        b1.setBounds(170, 499, 150, 35);
        l3.add(b1);

        b2.setBounds(390, 499, 150, 35);
        l3.add(b2);

        b3.setBounds(170, 543, 150, 35);
        l3.add(b3);

        b4.setBounds(390, 543, 150, 35);
        l3.add(b4);

        b5.setBounds(170, 588, 150, 35);
        l3.add(b5);

        b6.setBounds(390, 588, 150, 35);
        l3.add(b6);

        b7.setBounds(390, 633, 150, 35);
        l3.add(b7);

        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        b6.addActionListener(this);
        b7.addActionListener(this);

        setSize(960, 1080);
        setLocation(500, 0);
        setUndecorated(true);
        setVisible(true);

    }

    public void actionPerformed(ActionEvent ae) {
        try {
            String amount = ((JButton)ae.getSource()).getText().substring(3); //k
            Conn c = new Conn();
            ResultSet rs = c.s.executeQuery("select * from bank where pin = '"+pin+"'");
            int balance = 0;
            while (rs.next()) {
                if (rs.getString("mode").equals("Deposit")) {
                    balance += Integer.parseInt(rs.getString("amount"));
                } else {
                    balance -= Integer.parseInt(rs.getString("amount"));
                }
            } String num = "17";
            if (ae.getSource() != b7 && balance < Integer.parseInt(amount)) {
                JOptionPane.showMessageDialog(null, "Insuffient Balance");
                return;
            }

            if (ae.getSource() == b7) {
                this.setVisible(false);
                new Transactions(pin).setVisible(true);
            }else{
                Date date = new Date();
                c.s.executeUpdate("insert into bank values('"+pin+"', '"+date+"', 'Withdrawl',
'"+amount+"')");
```

```
            JOptionPane.showMessageDialog(null, "Rs. "+amount+" Debited Successfully");

            setVisible(false);
            new Transactions(pin).setVisible(true);
          }
        } catch (Exception e) {
          e.printStackTrace();
        }

    }

    public static void main(String[] args) {
        new FastCash("").setVisible(true);
    }
}
```

MiniStatement.java

```
package bank.management.system;

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

public class MiniStatement extends JFrame implements ActionListener{

    JButton b1, b2;
    JLabel l1;
    MiniStatement(String pin){
        super("Mini Statement");
        getContentPane().setBackground(Color.WHITE);
        setSize(400,600);
        setLocation(20,20);

        l1 = new JLabel();
        add(l1);

        JLabel l2 = new JLabel("Indian Bank");
        l2.setBounds(150, 20,  100, 20);
        add(l2);

        JLabel l3 = new JLabel();
        l3.setBounds(20, 80, 300, 20);
        add(l3);

        JLabel l4 = new JLabel();
        l4.setBounds(20, 400, 300, 20);
        add(l4);

        try{
            Conn c = new Conn();
            ResultSet rs = c.s.executeQuery("select * from login where pin = '"+pin+"'");
            while(rs.next()){
                l3.setText("Card Number:    " + rs.getString("cardno").substring(0, 4) +
"XXXXXXXX" + rs.getString("cardno").substring(12));
            }
        }catch(Exception e){}
```

```
        try{
            int balance = 0;
            Conn c1  = new Conn();
            ResultSet rs = c1.s.executeQuery("SELECT * FROM bank where pin = '"+pin+"'");
            while(rs.next()){
                l1.setText(l1.getText() + "<html>"+rs.getString("date")+
"     " + rs.getString("mode") +
"          " +
rs.getString("amount") + "<br><br><html>");
                if(rs.getString("mode").equals("Deposit")){
                    balance += Integer.parseInt(rs.getString("amount"));
                }else{
                    balance -= Integer.parseInt(rs.getString("amount"));
                }
            }
            l4.setText("Your total Balance is Rs "+balance);
        }catch(Exception e){
            e.printStackTrace();
        }

        setLayout(null);
        b1 = new JButton("Exit");
        add(b1);

        b1.addActionListener(this);

        l1.setBounds(20, 140, 400, 200);
        b1.setBounds(20, 500, 100, 25);
    }
    public void actionPerformed(ActionEvent ae){
        this.setVisible(false);
    }

    public static void main(String[] args){
        new MiniStatement("").setVisible(true);
    }

}
```

BalanceEnquiry.java

```
package bank.management.system;

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

class BalanceEnquiry extends JFrame implements ActionListener {

    JTextField t1, t2;
    JButton b1, b2, b3;
    JLabel l1, l2, l3;
    String pin;

    BalanceEnquiry(String pin) {
```

```
        this.pin = pin;

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
atm.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1000, 1180, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l3 = new JLabel(i3);
        l3.setBounds(0, 0, 960, 1080);
        add(l3);

        l1 = new JLabel();
        l1.setForeground(Color.WHITE);
        l1.setFont(new Font("System", Font.BOLD, 16));

        b1 = new JButton("BACK");

        setLayout(null);

        l1.setBounds(190, 350, 400, 35);
        l3.add(l1);

        b1.setBounds(390, 633, 150, 35);
        l3.add(b1);
        int balance = 0;
        try{
            Conn c1 = new Conn();
            ResultSet rs = c1.s.executeQuery("select * from bank where pin = '"+pin+"'");
            while (rs.next()) {
                if (rs.getString("mode").equals("Deposit")) {
                    balance += Integer.parseInt(rs.getString("amount"));
                } else {
                    balance -= Integer.parseInt(rs.getString("amount"));
                }
            }
        }catch(Exception e){}

        l1.setText("Your Current Account Balance is Rs "+balance);

        b1.addActionListener(this);

        setSize(960, 1080);
        setUndecorated(true);
        setLocation(500, 0);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        setVisible(false);
        new Transactions(pin).setVisible(true);
    }

    public static void main(String[] args) {
        new BalanceEnquiry("").setVisible(true);
    }
}
```

Transaction.java

```
package bank.management.system;
```

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

public class Transactions extends JFrame implements ActionListener{

    JLabel l1;
    JButton b1,b2,b3,b4,b5,b6,b7;
    String pin;
    Transactions(String pin){
        this.pin = pin;
        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
atm.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1000, 1180, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l2 = new JLabel(i3);
        l2.setBounds(0, 0, 960, 1080);
        add(l2);

        l1 = new JLabel("Please Select Your Transaction");
        l1.setForeground(Color.WHITE);
        l1.setFont(new Font("System", Font.BOLD, 16));


        b1 = new JButton("DEPOSIT");
        b2 = new JButton("CASH WITHDRAWL");
        b3 = new JButton("FAST CASH");
        b4 = new JButton("MINI STATEMENT");
        b5 = new JButton("PIN CHANGE");
        b6 = new JButton("BALANCE ENQUIRY");
        b7 = new JButton("EXIT");

        setLayout(null);

        l1.setBounds(235,400,700,35);
        l2.add(l1);

        b1.setBounds(170,499,150,35);
        l2.add(b1);

        b2.setBounds(390,499,150,35);
        l2.add(b2);

        b3.setBounds(170,543,150,35);
        l2.add(b3);

        b4.setBounds(390,543,150,35);
        l2.add(b4);

        b5.setBounds(170,588,150,35);
        l2.add(b5);

        b6.setBounds(390,588,150,35);
        l2.add(b6);

        b7.setBounds(390,633,150,35);
        l2.add(b7);


        b1.addActionListener(this);
```

```
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        b6.addActionListener(this);
        b7.addActionListener(this);


        setSize(960,1080);
        setLocation(500,0);
        setUndecorated(true);
        setVisible(true);



    }

    public void actionPerformed(ActionEvent ae){
        if(ae.getSource()==b1){
            setVisible(false);
            new Deposit(pin).setVisible(true);
        }else if(ae.getSource()==b2){
            setVisible(false);
            new Withdrawl(pin).setVisible(true);
        }else if(ae.getSource()==b3){
            setVisible(false);
            new FastCash(pin).setVisible(true);
        }else if(ae.getSource()==b4){
            new MiniStatement(pin).setVisible(true);
        }else if(ae.getSource()==b5){
            setVisible(false);
            new Pin(pin).setVisible(true);
        }else if(ae.getSource()==b6){
            this.setVisible(false);
            new BalanceEnquiry(pin).setVisible(true);
        }else if(ae.getSource()==b7){
            System.exit(0);
        }
    }

    public static void main(String[] args){
        new Transactions("").setVisible(true);
    }
}
```

Withdrawl.java

```
package bank.management.system;

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

public class Withdrawl extends JFrame implements ActionListener{
```

```
    JTextField t1,t2;
    JButton b1,b2,b3;
    JLabel l1,l2,l3,l4;
    String pin;
    Withdrawl(String pin){
        this.pin = pin;
        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/
atm.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1000, 1180, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l3 = new JLabel(i3);
        l3.setBounds(0, 0, 960, 1080);
        add(l3);

        l1 = new JLabel("MAXIMUM WITHDRAWAL IS RS.10,000");
        l1.setForeground(Color.WHITE);
        l1.setFont(new Font("System", Font.BOLD, 16));

        l2 = new JLabel("PLEASE ENTER YOUR AMOUNT");
        l2.setForeground(Color.WHITE);
        l2.setFont(new Font("System", Font.BOLD, 16));

        t1 = new JTextField();
        t1.setFont(new Font("Raleway", Font.BOLD, 25));

        b1 = new JButton("WITHDRAW");
        b2 = new JButton("BACK");

        setLayout(null);

        l1.setBounds(190,350,400,20);
        l3.add(l1);

        l2.setBounds(190,400,400,20);
        l3.add(l2);

        t1.setBounds(190,450,330,30);
        l3.add(t1);

        b1.setBounds(390,588,150,35);
        l3.add(b1);

        b2.setBounds(390,633,150,35);
        l3.add(b2);

        b1.addActionListener(this);
        b2.addActionListener(this);

        setSize(960,1080);
        setLocation(500,0);
        setUndecorated(true);
        setVisible(true);
    }


    public void actionPerformed(ActionEvent ae){
        try{
            String amount = t1.getText();
            Date date = new Date();
            if(ae.getSource()==b1){
                if(t1.getText().equals("")){
```

```
JOptionPane.showMessageDialog(null, "Please enter the Amount to you want to Withdraw");
            }else{
                Conn c1 = new Conn();

                ResultSet rs = c1.s.executeQuery("select * from bank where pin = '"+pin+"'");
                int balance = 0;
                while(rs.next()){
                  if(rs.getString("mode").equals("Deposit")){
                     balance += Integer.parseInt(rs.getString("amount"));
                  }else{
                     balance -= Integer.parseInt(rs.getString("amount"));
                  }
                }
                if(balance < Integer.parseInt(amount)){
                   JOptionPane.showMessageDialog(null, "Insuffient Balance");
                   return;
                }

                c1.s.executeUpdate("insert into bank values('"+pin+"', '"+date+"', 'Withdrawl',
'"+amount+"')");
                JOptionPane.showMessageDialog(null, "Rs. "+amount+" Debited Successfully");

                setVisible(false);
                new Transactions(pin).setVisible(true);
            }
          }else if(ae.getSource()==b2){
            setVisible(false);
            new Transactions(pin).setVisible(true);
          }
       }catch(Exception e){
            e.printStackTrace();
            System.out.println("error: "+e);
       }

   }



   public static void main(String[] args){
      new Withdrawl("").setVisible(true);
   }
}
```

5. Conclusion

The **ATM Machine Management System** is a reliable and user-centric solution designed to streamline banking transactions and enhance customer convenience. Utilizing Java for the frontend, MySQL for the backend, and JDBC for seamless database connectivity, the system ensures secure and efficient operations. It facilitates core banking functions such as cash withdrawals, deposits, fund transfers, and balance inquiries, catering to the needs of users in a fast-paced financial environment.

This project successfully showcases the integration of cutting-edge technologies to create a functional and scalable banking application. It provides a robust foundation for future upgrades, such as biometric authentication, cardless transactions, and AI-powered fraud detection. By meeting the evolving demands of modern banking, this system ensures accessibility, security, and adaptability, offering a progressive approach to enhancing customer satisfaction and banking efficiency.

6.Reference links:

- https://www.javatpoint.com/java-awt
- https://www.javatpoint.com/java-swing
- https://www.w3schools.com/sql/
- https://www.geeksforgeeks.org/introduction-to-jdbc/