[BANKING MANAGAMENT SYSTEM]

A MINI PROJECT REPORT

Submitted by

[VIVEK KUMAR]

[92201703084]

BACHELOR OF ENGINEERING

in

Computer Engineering

Marwadi University, Rajkot

[October, 2023]

Marwadi University

Rajkot

CERTIFICATE

This is to certify that the project report submitted along with the project entitled < **banking management system** > has been carried out by < **Vivek kumar , 92201703084>** under my guidance in partial fulfillment for the degree of Bachelor of Technology in Computer Engineering, 3rd Semester of Marwadi University, Rajkot during the academic year 2022-23.

<Sign>	<Sign>
< Charmy Vora>	< Prof Dr. Krunal Vaghela>
Internal Guide	Head of the Department
Computer Engineering	Computer Engineering
Marwadi university	Marwadi university

Marwadi University

Rajkot

DECLARATION

We hereby declare that the Mini Project report submitted along with the Project entitled **<Banking management system>** submitted in partial fulfilment for the degree of Bachelor of Technology in **Computer Engineering** to Marwadi University, Rajkot, is a bonafide record of original project work carried out by me / us at Marwadi University under the supervision of < **Charmy Vora** > and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student	Sign of Student
1. VIVEK KUMAR	

List of Figures

List of Tables

Abbreviations

ALU	Arithmetical & Logical Unit
SDLC	Software Development Life Cycle

Table of Contents

CHAPTER 1

OVERVIEW OF JAVA

1.1 Introduction of Java

(add topics/contents wherever required as per your project requirements)

Java is a widely used, high-level, object-oriented programming language developed by Sun Microsystems (now owned by Oracle Corporation) in the mid-1990s. It has since become one of the most popular programming languages in the world due to its portability ,platform independence, and wide range of applications. Here's a brief introduction to Java:

Platform Independence: One of Java's most significant features is its ability to run on multiple platforms without modification. This is achieved through the use of the Java Virtual Machine (JVM), which allows Java programs to be compiled once and run on any platform that has a compatible JVM.

Object-Oriented: Java is an object-oriented programming (OOP) language, which means it's based on the concept of objects and classes. This promotes the reuse of code and makes it easier to structure and manage complex software projects.

Syntax: Java's syntax is similar to other C-based languages like C++ and C#. This makes it relatively easy for developers who are already familiar with these languages to learn Java.
Automatic Memory Management: Java has a built-in garbage collector that automatically reclaims memory that is no longer in use. This helps prevent memory leaks and makes memory management less error-prone.

Rich Standard Library: Java comes with a vast standard library that provides a wide range of classes and methods for common programming tasks, such as working with data structures, networking, input/output, and more. This extensive library saves developers time and effort when writing code.

Security: Java was designed with security in mind. The JVM includes features like bytecode verification and security manager to ensure that running Java code is safe and doesn't harm the host system.

Multi-threading: Java provides built-in support for multithreading, allowing developers to write concurrent programs that can take full advantage of multi-core processors.
Community and Ecosystem: Java has a large and active community of developers. It also has a rich ecosystem of tools, frameworks, and libraries that make it suitable for a wide variety of applications, from web development to mobile app development.

Write Once, Run Anywhere (WORA): This is a key principle in Java. Code written in Java can be executed on any device or platform that has a compatible JVM, making it a versatile choice for cross-platform development.

Popular Use Cases: Java is used in a wide range of applications, including web development (Java EE), Android app development, enterprise software (Java SE), scientific computing, and more.

Popular Frameworks: Java has a range of popular frameworks and technologies, such as Spring, Hibernate, JavaFX, and JavaServer Faces (JSF), which simplify the development of various types of applications.

CHAPTER 2

OVERVIEW OF PROJECT

2.1 Introduction of Banking Management System

(add topics/contents wherever required as per your project requirements)

A Banking Management System is a specialized software or information system designed to help banks and financial institutions efficiently manage their day-to-day operations and provide a wide range of banking services to customers. It encompasses a suite of tools and features that streamline various banking processes, from customer transactions to internal operations. Here's an introduction to a typical Banking Management System:

Customer Management: Banking Management Systems store and manage customer information, including personal details, account data, and transaction history. This helps banks maintain accurate customer records and ensures data security.

Account Management: The system handles the creation, modification, and closure of customer accounts. It tracks account balances, transactions, and generates account statements.

Transaction Processing: Banking Management Systems facilitate a variety of transactions, such as deposits, withdrawals, fund transfers, and bill payments. These transactions can be carried out both in-branch and through digital channels like online banking or mobile apps.

Loan and Credit Management: Banks use the system to process and manage loan applications, credit assessments, and approvals. It also tracks loan repayments and interest calculations.

Interest and Investment Calculations: The system can calculate and manage interest rates, investments, and other financial products, helping customers make informed decisions about their savings and investments.

Customer Service and Support: It often includes customer service features like chat support, email communication, and help desk management to address customer inquiries and complaints.

ATM and Card Management: Banking Management Systems manage ATM operations, including cash dispensing and deposits. They also monitor debit/credit card operations and provide security features to prevent fraudulent transactions.

Online and Mobile Banking: Many systems offer online and mobile banking functionalities, allowing customers to access their accounts, make transactions, and perform various banking activities through the internet or mobile apps.

Risk Management: These systems help banks identify and manage financial risks, such as credit risk, market risk, and operational risk. They can also help with fraud detection and prevention.

Regulatory Compliance: Banking systems ensure that banks adhere to regulatory requirements and reporting standards, such as Anti-Money Laundering (AML) and Know Your Customer (KYC) regulations.

Reporting and Analytics: The system generates reports and provides data analytics tools to help banks make informed decisions, assess performance, and identify trends in customer behavior.

Security and Access Control: Banking Management Systems are equipped with robust security measures to safeguard customer data and financial transactions. They use encryption, authentication, and access controls to prevent unauthorized access.

Integration: These systems often need to integrate with various external systems and services, such as payment gateways, clearinghouses, and credit bureaus.

Scalability: Banking Management Systems are designed to accommodate the growth and changing needs of a bank, making them scalable to meet the demands of a larger customer base.

Audit Trail: A comprehensive audit trail is maintained to record all activities within the system for accountability and compliance purposes.

2.2 Drawbacks of Existing System

The existing systems in banking management often face several drawbacks and limitations, which can impede efficiency, security, and customer satisfaction. Some of the common drawbacks of the existing banking management systems include:

- Limited Accessibility:
- Manual Processes:
- Security Vulnerabilities:
- Limited Self-Service Options
- Data Redundancy
- Scalability Issues
- Costly Maintenance
- Limited Automation

2.3 Advantages of Proposed System

A proposed Banking Management System aims to overcome the limitations of the existing system and introduce new features and improvements that enhance the efficiency, security, and overall quality of service provided by the bank. Here are some advantages of a well-designed proposed system in a banking management project:

- Enhanced Accessibility:
- Automation:
- Real-time Transactions:
- Data Management:
- Regulatory Compliance:
- Scalability:
- Cost Savings:
- Technology Advancements:
- Improved User Experience:

2.4 Functional Requirements

Functional requirements for a Banking Management System in a project define the specific features and capabilities that the system must have to meet the needs of the bank and its customers. These requirements help ensure that the system functions correctly and efficiently. Here are some key functional requirements for a Banking Management System:

- User Authentication and Authorization:
- Customer Registration:
- Account Management:
- Transaction Processing:
- Loan and Credit Management:
- ATM and Card Management:
- Account Locking and Alerts:
- Electronic Funds Transfer

2.4.1 Tools

Programming Languages:

- **Java**: It's a popular choice for developing banking systems due to its security features.
- **IDE (Integrated Development Environment**): Choose an IDE like IntelliJ IDEA or Eclipse for efficient development.
- **Html:** HTML documents are structured with elements enclosed in tags. Tags are usually in angle brackets like <tagname>. The basic structure of an HTML document includes:

WebSockets :

Implement real-time communication using WebSockets for instant messaging. Libraries like Java-WebSocket can help.

Database Management:

- **SQL Server, or MySQL**: These relational databases are often used to store customer and transaction data.

2.4.2 Front End and Back End

- **Front-End Technologies**: Building a web-based application, using HTML, Java, SQL and Query
- **Back-End Technologies:**

Server-Side Frameworks: Spring Framework (Java) Popular for building robust and secure Java-based applications.
Database Management:

Xmapp Connection, SQL Server, or MySQL: Common relational databases for managing customer data and transactions.

API Development:
RESTful APIs: REST is a common architectural style for designing networked applications. It's often used for building APIs in banking systems.

CHAPTER 3

PROJECT SOURCE CODE

3.1 Banking Management System Source Code

Login.java

```
package bank.management.system;


import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
public class Login extends JFrame implements ActionListener{
     JButton login,singup,clear;
    JTextField cardTextField;
    JPasswordField pinTextField;

    Login(){
        setTitle("AUTOMATED TALLER MACHINE");
         setLayout(null);

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/logo.jpg"));
        Image i2 =i1.getImage().getScaledInstance(100,100,Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel label = new JLabel(i3);
        label.setBounds(70,10,100,100);
        add(label);
        JLabel text = new JLabel("Welcome to ATM");
        text.setFont(new Font("Osward", Font.BOLD,38));
        text.setBounds(200,40,400,40);
        add(text);
        JLabel cardno = new JLabel("Card No:");
        cardno.setFont(new Font("Raleway", Font.BOLD,28));
        cardno.setBounds(120,150,150,30);
        add(cardno);
         cardTextField = new JTextField();
        cardTextField.setBounds(300,150,230,30);
        cardTextField.setFont(new Font("Arial",Font.BOLD,14));
        add(cardTextField);
        JLabel pin = new JLabel("PIN :");
        pin.setFont(new Font("Raleway", Font.BOLD,28));
        pin.setBounds(120,220,250,30);
        add(pin);
        pinTextField = new JPasswordField();
        pinTextField.setBounds(300,220,230,30);
        pinTextField.setFont(new Font("Arial",Font.BOLD,14));
        add(pinTextField);
         login = new JButton("SIGN IN");
        login.setBounds(300,300,100,30);
        login.setBackground(Color.black);
        login.setForeground(Color.white);
        login.addActionListener(this);
        add(login);
          clear = new JButton("CLEAR ");
         clear.setBounds(430,300,100,30);
         clear.setBackground(Color.black);
         clear.setForeground(Color.white);
         clear.addActionListener(this);
         add(clear);
```

```
          singup = new JButton("SING UP");
         singup.setBounds(300,350,230,30);
         singup.setBackground(Color.black);
         singup.setForeground(Color.white);
         singup.addActionListener(this);
         add(singup);

        getContentPane().setBackground(Color.white);
         setSize(800,480);
        setVisible(true);
        setLocation(350,200);
    }
    public void actionPerformed(ActionEvent ae){
      if(ae.getSource()==clear) {
          cardTextField.setText("");
          pinTextField.setText("");
      }
      else if(ae.getSource() == login){
         conn con = new conn();
         String cardnumber = cardTextField.getText();
         String pinnumber = pinTextField.getText();
         String query = "Select * from login where cardnumber ='"+cardnumber+"'and pin ='"+pinnumber+"'";
         try{
           ResultSet rs = con.s.executeQuery(query);
          if(rs.next()){
              setVisible(false);
              new Transactions(pinnumber).setVisible(true);
          }
          else {
              JOptionPane.showMessageDialog(null,"Incorrect card Number or Pin");
          }
          }
          catch (Exception e){
              e.printStackTrace();
          }
      }
      else if(ae.getSource()== singup){
          setVisible(false);
          new SingupOne().setVisible(true);
      }
    }
        public static void main(String[] args) {
          new Login();
    }
}
```

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");
            String url = "jdbc:mysql://localhost:3306/bankmanagementsystem";
            String user = "root";
            String password = "";
            c = DriverManager.getConnection(url, user, password);
            s = c.createStatement();
        } catch (Exception e) {
            System.out.println(e);
        }     }
}
```

SingupOne.java

```
package bank.management.system;
import com.toedter.calendar.JDateChooser;
import java.awt.*;
import javax.swing.*;
import java.util.*;
import com.toedter.calendar.JDateChooser;
import java.awt.event.*;
import java.lang.*;
public class SingupOne extends JFrame implements ActionListener{
         long random;
     JTextField
nameTextField,fnameTextField,dobTextField,gmailTextField,addressTextField,
cityTextField ,stateTextField ,pincodeTextField;
     JButton next;
     JRadioButton male,female,other,married,unmarried;
     JDateChooser dateChooser;
        SingupOne(){
            setLayout(null);
           Random ran = new Random();
            random = Math.abs((ran.nextLong()  % 9000L) + 1000L);
             JLabel formno = new JLabel("APPLICATION FORM NO." + random );
            formno.setFont(new Font("Raleway",Font.BOLD,38));
            formno.setBounds(140,20,600,40);
            add(formno);
```

```
JLabel personDetails = new JLabel("page. 1: Personal Details");
personDetails.setFont(new Font("Raleway",Font.BOLD,22));
personDetails.setBounds(290,80,400,30);
add(personDetails);

JLabel name = new JLabel("Name:");
name.setFont(new Font("Raleway",Font.BOLD,20));
name.setBounds(100,140,100,30);
add(name);
nameTextField = new JTextField();
nameTextField.setFont(new Font("Raleway", Font.BOLD,14));
nameTextField.setBounds(300,140,400,30);
add(nameTextField);
 JLabel fname = new JLabel(" Father's Name:");
fname.setFont(new Font("Raleway",Font.BOLD,20));
fname.setBounds(100,190,200,30);
add(fname);
fnameTextField = new JTextField();
fnameTextField.setFont(new Font("Raleway", Font.BOLD,14));
fnameTextField.setBounds(300,190,400,30);
add(fnameTextField);
JLabel dod = new JLabel(" Date of birth:");
dod.setFont(new Font("Raleway",Font.BOLD,20));
dod.setBounds(100,240,200,30);
add(dod);
dateChooser = new JDateChooser();
dateChooser.setBounds(300,240,400,30);
dateChooser.setForeground(new Color(105,105,105));
add(dateChooser);
JLabel gender = new JLabel(" Gender:");
gender.setFont(new Font("Raleway",Font.BOLD,20));
gender.setBounds(100,290,200,30);
add(gender);
  male = new JRadioButton("Male");
 male.setBounds(300,290,60,30);
 male.setBackground(Color.white);
 add(male);
 female = new JRadioButton("Female");
 female.setBounds(450,290,120,30);
 female.setBackground(Color.white);
 add(female);
  ButtonGroup gendergroup = new ButtonGroup();
 gendergroup.add(male);
 gendergroup.add(female);
JLabel email = new JLabel(" Email Address:");
email.setFont(new Font("Raleway",Font.BOLD,20));
email.setBounds(100,340,200,30);
add(email);
 gmailTextField = new JTextField();
gmailTextField.setFont(new Font("Raleway", Font.BOLD,14));
gmailTextField.setBounds(300,340,400,30);
add(gmailTextField);
JLabel Marital = new JLabel(" Marital Status:");
Marital.setFont(new Font("Raleway",Font.BOLD,20));
Marital.setBounds(100,390,200,30);
add(Marital);
 married = new JRadioButton("Married");
 married.setBounds(300,390,100,30);
 married.setBackground(Color.white);
 add(married);
  unmarried = new JRadioButton("Unmarried");
 unmarried.setBounds(450,390,100,30);
 unmarried.setBackground(Color.white);
 add(unmarried);
 other = new JRadioButton("Other");
 other.setBounds(630,390,100,30);
 other.setBackground(Color.white);
 add(other);
 ButtonGroup marritalgroup = new ButtonGroup();
 marritalgroup.add(married);
 marritalgroup.add(unmarried);
 marritalgroup.add(other);
JLabel address = new JLabel(" Address:");
address.setFont(new Font("Raleway",Font.BOLD,20));
address.setBounds(100,440,200,30);
add(address);
addressTextField = new JTextField();
addressTextField.setFont(new Font("Raleway", Font.BOLD,14));
addressTextField.setBounds(300,440,400,30);
add(addressTextField);
JLabel city = new JLabel(" City:");
city.setFont(new Font("Raleway",Font.BOLD,20));
city.setBounds(100,490,200,30);
add(city);
cityTextField = new JTextField();
cityTextField .setFont(new Font("Raleway", Font.BOLD,14));
cityTextField .setBounds(300,490,400,30);
add(cityTextField );
JLabel state = new JLabel(" State:");
state.setFont(new Font("Raleway",Font.BOLD,20));
state.setBounds(100,540,200,30);
add(state);
```

```
       stateTextField = new JTextField();
       stateTextField .setFont(new Font("Raleway", Font.BOLD,14));
       stateTextField .setBounds(300,540,400,30);
       add(stateTextField );

       JLabel pincode = new JLabel(" Pincode:");
       pincode.setFont(new Font("Raleway",Font.BOLD,20));
       pincode.setBounds(100,590,200,30);
       add(pincode);

        pincodeTextField = new JTextField();
       pincodeTextField .setFont(new Font("Raleway", Font.BOLD,14));
       pincodeTextField .setBounds(300,590,400,30);
       add(pincodeTextField );



       next = new JButton("Next");
       next.setBackground(Color.black);
       next.setForeground(Color.white);
       next.setFont(new Font("Raleway",Font.BOLD,14));
       next.setBounds(620,660,80,30);
       next.addActionListener(this);
       add(next);
       getContentPane().setBackground(Color.white);
       setSize(850,800);
       setLocation(350,10);
       setVisible(true);
   }
@Override
  public void actionPerformed(ActionEvent ae){
    String formno ="" +random;
String name = nameTextField.getText();
String fname = fnameTextField.getText();
String dob = ((JTextField)dateChooser.getDateEditor().getUiComponent()).getText();
String gender = null;
if(male.isSelected()){
    gender ="Male";
}
else if(female.isSelected()){
    gender ="Female";
}
String email = gmailTextField.getText();
String marital = null;
if(married.isSelected()){
    marital = "Marital";
}
else if(unmarried.isSelected()){
    marital = "Unmarital";
}
 else if(other.isSelected()){
    marital = "Other";
}
 String address = addressTextField.getText();
 String city = cityTextField.getText();
 String state = stateTextField.getText();
 String pin = pincodeTextField.getText();
 try{
    if(name.equals("")){
        JOptionPane.showMessageDialog(null, "Name is required");
    }
    else {
        conn cn = new conn();
//                    String query ="insert into signup values(NULLs,'"+name+"','"+fname+"','"+dob+"','"+gender+"','"+email+"','"+marital+"','"+address+"','"+city+"','"+state+"','"+pin+"')";
//          String query="INSERT INTO signup (formno, name, father_name, dob, Gender, Email, marital_status, Address, city, state, pincode) VALUES (NULL,'"+name+"', '"+fname+"', '"+dob+"', '"+gender+"', '"+email+"', '"+marital+"','"+address+"', '"+city+"', '"+state+"', '"+pin+"'); ";
                  String query= "INSERT INTO signup (formno, name, father_name, dob, gender, email, marital_status, Address, city, state, pincode) VALUES ('"+formno+"','"+name+"', '"+fname+"', '"+dob+"', '"+gender+"', '"+email+"', '"+marital+"','"+address+"', '"+city+"', '"+state+"', '"+pin+"'); ";
                 int i=cn.s.executeUpdate(query);
                 System.out.print(i);
               setVisible(false);
                 new SingupTwo(formno).setVisible(true);
                 System.out.print(i);
             }
          }catch (Exception e){
              e.printStackTrace();
          }
    }
    public static void main(String args[]) {
      new SingupOne();
    }
}
```

SingupTwo.java

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

```
public class SingupTwo extends JFrame implements ActionListener{
    JTextField  pan,aadhar;
    JButton next;
    JRadioButton syes,sno,eyes,eno;
    JComboBox religion,category,occupation,education,income;
    String formno;
          SingupTwo(String formno){
         this.formno = formno;
         setLayout(null);
         setTitle("NEW ACCOUNT APPLICATION FORM - PAGE 2");
         JLabel additionalDetails = new JLabel("page. 2: additional Details");
         additionalDetails.setFont(new Font("Raleway",Font.BOLD,22));
         additionalDetails.setBounds(290,80,400,30);
         add(additionalDetails);
         JLabel name = new JLabel("Religion:");
         name.setFont(new Font("Raleway",Font.BOLD,20));
         name.setBounds(100,140,100,30);
         add(name);
         String valreligion[] = {"Hindu" , "Muslim","Sikh","Christian","Other"};
         religion = new JComboBox(valreligion);
         religion.setBounds(300,140,400,30);
         religion.setBackground(Color.white);
         add(religion);
         JLabel fname = new JLabel(" Category:");
         fname.setFont(new Font("Raleway",Font.BOLD,20));
         fname.setBounds(100,190,200,30);
         add(fname);
```

```
        String valcategory[] = {"General","OBC","SC","ST","Other"};
        category = new JComboBox(valcategory);
        category.setBounds(300,190,400,30);
        category.setBackground(Color.white);
        add(category);
        JLabel dod = new JLabel(" Income:");
        dod.setFont(new Font("Raleway",Font.BOLD,20));
        dod.setBounds(100,240,200,30);
        add(dod);

        String incomecategory[] = {"NUll","<1,50,000","<2,50,000","<5,00,000","upto 10,00,000"};
        income = new JComboBox(incomecategory);
        income.setBounds(300,240,400,30);
        income.setBackground(Color.white);
        add(income);
        JLabel gender = new JLabel(" Educational:");
        gender.setFont(new Font("Raleway",Font.BOLD,20));
        gender.setBounds(100,290,200,30);
        add(gender);
        JLabel email = new JLabel("Qualification:");
        email.setFont(new Font("Raleway",Font.BOLD,20));
        email.setBounds(100,320,200,30);
        add(email);

        String educationvalues[] = {"Non Graduation","Graduation","Post Graduation","Doctrate","Others"};
        education = new JComboBox(educationvalues);
        education.setBounds(300,320,400,30);
        education.setBackground(Color.white);
        add(education);
        JLabel Marital = new JLabel(" Occupation:");
        Marital.setFont(new Font("Raleway",Font.BOLD,20));
        Marital.setBounds(100,390,200,30);
        add(Marital);

        String Occupationvalues[] = {"salaried","self-Employed","Bussiness","Student","Retired","Others"};
        occupation = new JComboBox(Occupationvalues);
        occupation.setBounds(300,390,400,30);
        occupation.setBackground(Color.white);
        add(occupation);
        JLabel panno = new JLabel(" PAN number:");
        panno.setFont(new Font("Raleway",Font.BOLD,20));
        panno.setBounds(100,440,200,30);
        add(panno);
        pan = new JTextField();
        pan.setFont(new Font("Raleway", Font.BOLD,14));
        pan.setBounds(300,440,400,30);
        add(pan);
        JLabel adharno = new JLabel(" Aadhar number:");
        adharno.setFont(new Font("Raleway",Font.BOLD,20));
        adharno.setBounds(100,490,200,30);
        add(adharno);
        aadhar = new JTextField();
        aadhar .setFont(new Font("Raleway", Font.BOLD,14));
        aadhar .setBounds(300,490,400,30);
        add(aadhar );
        JLabel state = new JLabel(" Senior Citizen:");
        state.setFont(new Font("Raleway",Font.BOLD,20));
        state.setBounds(100,540,200,30);
        add(state);
```

```
 syes = new JRadioButton("Yes");
 syes.setBounds(300,540,100,30);
 syes.setBackground(Color.white);
 add(syes);
 sno = new JRadioButton("No");
 sno.setBounds(450,540,100,30);
 sno.setBackground(Color.white);
 add(sno);
 ButtonGroup SeniorCitizengroup = new ButtonGroup();
 SeniorCitizengroup.add(syes);
 SeniorCitizengroup.add(sno);
JLabel pincode = new JLabel(" Exisition account:");
pincode.setFont(new Font("Raleway",Font.BOLD,20));
pincode.setBounds(100,590,200,30);
add(pincode);
 eyes = new JRadioButton("Yes");
 eyes.setBounds(300,590,100,30);
 eyes.setBackground(Color.white);
 add(eyes);
 eno = new JRadioButton("No");
 eno.setBounds(450,590,100,30);
 eno.setBackground(Color.white);
 add(eno);
 ButtonGroup Exisitionaccountgroup = new ButtonGroup();
 Exisitionaccountgroup.add(eyes);
 Exisitionaccountgroup.add(eno);
next = new JButton("Next");
next.setBackground(Color.black);
next.setForeground(Color.white);
next.setFont(new Font("Raleway",Font.BOLD,14));
next.setBounds(620,660,80,30);
```

```
            next.addActionListener(this);
            add(next);
            getContentPane().setBackground(Color.white);
            setSize(850,800);
            setLocation(350,10);
            setVisible(true);
        }
     @Override
       public void actionPerformed(ActionEvent ae){
//          String formno ="" +random;
          String sreligion = (String) religion.getSelectedItem();
          String scategory = (String) category.getSelectedItem();
          String sincome = (String) income.getSelectedItem();
          String seducation = (String) education.getSelectedItem();
          String soccupation = (String) occupation.getSelectedItem();
           String seniorcitizen = null;
          if(syes.isSelected()){
              seniorcitizen ="Yes";
          }
          else if(sno.isSelected()){
              seniorcitizen ="No";
          }
          String existingaccount = null;
          if(eyes.isSelected()){
              existingaccount = "Yes";
          }
          else if(eno.isSelected()){
              existingaccount = "No";
          }
            String span = pan.getText();
           String saadhar = aadhar.getText();
        try{
               conn cn = new conn();
               String query = "INSERT INTO signuptwo (formno, religion, category, income,
education, occupation, pan, aadhar, seniorcitizen, existingaccount) VALUES
('"+formno+"','"+sreligion+"', '"+scategory+"', '"+sincome+"', '"+seducation+"',
'"+soccupation+"', '"+span+"','"+saadhar+"', '"+seniorcitizen+"', '"+existingaccount+"'); ";
              int i=cn.s.executeUpdate(query);
              System.out.print(i);
             setVisible(false);
             new SingupThree(formno).setVisible(true);
                }catch (Exception e){
           e.printStackTrace();
       }
   }
      public static void main(String args[]) {
    new SingupTwo("");
   }
}
```

SingupThree.java

```
package bank.management.system;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
public class SingupThree extends JFrame implements ActionListener{
     JRadioButton r1 , r2 , r3 , r4;
    JCheckBox c1 , c2 , c3 , c4 , c5 , c6 , c7;
    JButton submit,cancel;
    String formno;
```

```
SingupThree(String formno){
    this.formno=formno;
    setLayout(null);
     JLabel l1 = new JLabel("Page 3: Account Details");
    l1.setFont(new Font("Raleway",Font.BOLD,22));
    l1.setBounds(280,40,400,40);
    add(l1);
     JLabel type = new JLabel("Account Type:");
    type.setFont(new Font("Raleway",Font.BOLD,22));
    type.setBounds(100,140,200,30);
    add(type);
    r1 = new JRadioButton("Saving Account");
    r1.setFont(new Font("Raleway",Font.BOLD,16));
    r1.setBackground(Color.WHITE);
    r1.setBounds(100,180,150,20);
    add(r1);
    r2 = new JRadioButton("Fixed Deposit Account");
    r2.setFont(new Font("Raleway",Font.BOLD,16));
    r2.setBackground(Color.WHITE);
    r2.setBounds(350,180,250,20);
    add(r2);
    r3 = new JRadioButton("Current Account");
    r3.setFont(new Font("Raleway",Font.BOLD,16));
    r3.setBackground(Color.WHITE);
    r3.setBounds(100,220,250,20);
    add(r3);
     r4 = new JRadioButton("Recurring Deposit Account");
    r4.setFont(new Font("Raleway",Font.BOLD,16));
    r4.setBackground(Color.WHITE);
    r4.setBounds(350,220,250,20);
    add(r4);
```

```
ButtonGroup groupaccount = new ButtonGroup();
groupaccount.add(r1);
groupaccount.add(r2);
groupaccount.add(r3);
groupaccount.add(r4);
JLabel card = new JLabel("Card Number:");
card.setFont(new Font("Raleway",Font.BOLD,22));
card.setBounds(100,300,200,30);
add(card);
JLabel number = new JLabel("XXXX-XXXX-XXXX-4184");
number.setFont(new Font("Raleway",Font.BOLD,22));
number.setBounds(330,300,300,30);
add(number);
  JLabel carddetail = new JLabel("Your 16 Digit Card Number");
carddetail.setFont(new Font("Raleway",Font.BOLD,12));
carddetail.setBounds(100,330,300,20);
add(carddetail);
JLabel pin = new JLabel("Pin :");
pin.setFont(new Font("Raleway",Font.BOLD,22));
pin.setBounds(100,370,200,30);
add(pin);
JLabel pnumber = new JLabel("XXXX");
pnumber.setFont(new Font("Raleway",Font.BOLD,22));
pnumber.setBounds(330,370,300,30);
add(pnumber);
JLabel pindetail = new JLabel("Your 4 Digit password");
pindetail.setFont(new Font("Raleway",Font.BOLD,12));
pindetail.setBounds(100,400,300,20);
add(pindetail);
```

```
JLabel services = new JLabel("Services Required :");
services.setFont(new Font("Raleway",Font.BOLD,22));
services.setBounds(100,450,400,30);
add(services);
c1 = new JCheckBox("ATM CARD");
c1.setBackground(Color.WHITE);
c1.setFont(new Font("Raleway",Font.BOLD,16));
c1.setBounds(100,500,200,30);
 add(c1);
c2 = new JCheckBox("Internet Banking");
c2.setBackground(Color.WHITE);
c2.setFont(new Font("Raleway",Font.BOLD,16));
c2.setBounds(350,500,200,30);
add(c2);
c3 = new JCheckBox("Mobile Banking");
c3.setBackground(Color.WHITE);
c3.setFont(new Font("Raleway",Font.BOLD,16));
c3.setBounds(100,550,200,30);
add(c3);
c4 = new JCheckBox("EMAIL & SMS Alerts");
c4.setBackground(Color.WHITE);
c4.setFont(new Font("Raleway",Font.BOLD,16));
c4.setBounds(350,550,200,30);
 add(c4);
c5 = new JCheckBox("Cheque Book");
c5.setBackground(Color.WHITE);
c5.setFont(new Font("Raleway",Font.BOLD,16));
c5.setBounds(100,600,200,30);
add(c5);
c6 = new JCheckBox("E-Statement");
c6.setBackground(Color.WHITE);
```

```
        c6.setFont(new Font("Raleway",Font.BOLD,16));
        c6.setBounds(350,600,200,30);
        add(c6);
        c7 = new JCheckBox("Hereby Declares that the above entered details are correct to the best of my knowledge");
        c7.setBackground(Color.WHITE);
        c7.setFont(new Font("Raleway",Font.BOLD,12));
        c7.setBounds(100,680,600,30);
        add(c7);
        submit = new JButton("Submit");
        submit.setBackground(Color.black);
        submit.setForeground(Color.WHITE);
        submit.setFont(new Font("Raleway",Font.BOLD,14));
        submit.setBounds(250,720,100,30);
        submit.addActionListener(this);
        add(submit);
        cancel = new JButton("Cancel");
        cancel.setBackground(Color.black);
        cancel.setForeground(Color.WHITE);
        cancel.setFont(new Font("Raleway",Font.BOLD,14));
        cancel.setBounds(420,720,100,30);
        cancel.addActionListener(this);
        add(cancel);
        getContentPane().setBackground(Color.white);
        setSize(850 ,820);
        setLocation(350 ,0);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent ae){
        if(ae.getSource()==submit){
            String accounttype = null;
```

```
if(r1.isSelected()){
    accounttype = "Saving Account";
}
else if(r2.isSelected()){
    accounttype = "Fixed Deposit Account";
}
 else if(r3.isSelected()){
    accounttype = "Current Account";
}
 else if(r4.isSelected()){
    accounttype = " Reccuring Deposit Account";
}
Random random =  new Random();
String cardnumber =" " + Math.abs((random.nextLong() % 90000000L) + 5040936000000000L);
String pinnumber =" " + Math.abs((random.nextLong() % 9000L) + 1000L);
String facility ="";
if(c1.isSelected()){
    facility = facility + " AMT card";
}
else if (c2.isSelected()){
    facility = facility + " Internet Banking";
}
else if (c3.isSelected()){
    facility = facility + " Mobile Banking";
}
else if (c4.isSelected()){
    facility = facility + " EMAIL & SMS Alerts";
}
else if (c5.isSelected()){
    facility = facility + " Cheque Book";
```

```
         }
         else if (c6.isSelected()){
             facility = facility + " E-Statement";
         }
        try{
            if(accounttype.equals("")){
               JOptionPane.showMessageDialog(null,"Account Type is Required");
            }
            else {
              conn con = new conn();
              String query1 = "INSERT INTO signupthree (formno, accountType, cardnumber,
pin, facility) VALUES ('"+formno+"', '"+accounttype+"', '"+cardnumber+"', '"+pinnumber+"',
'"+facility+"')";
              String query2 ="INSERT INTO login (formno,cardnumber,pin) VALUES
('"+formno+"', '"+cardnumber+"','"+pinnumber+"')";
              int q1=con.s.executeUpdate(query1);
              int q2=con.s.executeUpdate(query2);

               System.out.println(q1);
               System.out.println(q2);

               JOptionPane.showMessageDialog(null,"Card Number :" + cardnumber + "\n pin
:"+pinnumber);

               setVisible(false);
               new Deposit(pinnumber).setVisible(false);

            }

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

        }else if( ae.getSource() == cancel){
            setVisible(false);
            new Login().setVisible(true);
         }
    }
    public static void main(String[] args) {
        new SingupThree("");
    }

}
```

Transaction.java

```
package bank.management.system;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Transactions extends JFrame implements ActionListener{
    JButton deposit,Withdrawl,ministatement,pinchange,fastcash,balanceenquiry,exit;
    String pinnumber;
    Transactions(String pinnumber){
        this.pinnumber=pinnumber;
    setLayout(null);
    ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/atm.jpg"));
    Image i2 =i1.getImage().getScaledInstance(900,900,Image.SCALE_DEFAULT);
    ImageIcon i3 = new ImageIcon(i2);
    JLabel image = new JLabel(i3);
    image.setBounds(0,0,900,900);
    add(image);
```

```
JLabel text = new JLabel("Please select your Transaction");
text.setFont(new Font("System", Font.BOLD,16));
text.setForeground(Color.WHITE);
text.setBounds(210,300,700,35);
image.add(text);
 deposit = new JButton("Deposit");
 deposit.setBounds(170,415,150,30);
 deposit.addActionListener(this);
 image.add(deposit);

 Withdrawl = new JButton("Cash Withdrawl");
 Withdrawl.setBounds(355,415,150,30);
 Withdrawl.addActionListener(this);
 image.add(Withdrawl);

 fastcash = new JButton("fast Cash");
 fastcash.setBounds(170,450,150,30);
 fastcash.addActionListener(this);
 image.add(fastcash);

 ministatement = new JButton("Mini Statement");
 ministatement.setBounds(355,450,150,30);
 ministatement.addActionListener(this);
 image.add(ministatement);

 pinchange = new JButton("Pin Change");
 pinchange.setBounds(170,485,150,30);
 pinchange.addActionListener(this);
 image.add(pinchange);
```

```
  balanceenquiry = new JButton("Balance Enquiry");
  balanceenquiry.setBounds(355,485,150,30);
  balanceenquiry.addActionListener(this);
  image.add(balanceenquiry);
  exit = new JButton("exit");
  exit.setBounds(355,520,150,30);
  exit.addActionListener(this);
  image.add(exit);
   setSize(900,900);
   setLocation(300,0);
   setUndecorated(true);
   setVisible(true);
 }
public void actionPerformed(ActionEvent ae){
   if(ae.getSource() == exit){
      System.exit(0);
   }else if(ae.getSource() == deposit){
      setVisible(false);
      new Deposit(pinnumber).setVisible(true);
   }
   else if(ae.getSource() == Withdrawl){
      setVisible(false);
      new Withdrawl(pinnumber).setVisible(true);
   }
   else if(ae.getSource() == fastcash){
      setVisible(false);
      new FastCash(pinnumber).setVisible(true);
   }
   else if(ae.getSource() == pinchange){
      setVisible(false);
      new PinChange(pinnumber).setVisible(true);
      }
      else if(ae.getSource() == balanceenquiry){
          setVisible(false);
          new BalanceEnquiry(pinnumber).setVisible(true);
      }
      else if(ae.getSource() == ministatement){
          new MiniStatement(pinnumber).setVisible(true);
      }
   }
    public static void main(String[] args) {
         new Transactions("");
    }

}
```

Deposit.java

```
package bank.management.system;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
public class Deposit  extends JFrame implements ActionListener{
        JButton deposit,back;
        JTextField amount;
        String pinnumber;
        Deposit(String pinnumber){
        this.pinnumber= pinnumber;
        setLayout(null);

    ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/atm.jpg"));
    Image i2 =i1.getImage().getScaledInstance(900,900,Image.SCALE_DEFAULT);
```

```
ImageIcon i3 = new ImageIcon(i2);
JLabel image = new JLabel(i3);
image.setBounds(0,0,900,900);
add(image);
JLabel text = new JLabel("Enter the amount you want to deposit");
text.setFont(new Font("System", Font.BOLD,16));
text.setForeground(Color.WHITE);
text.setBounds(170,300,400,20);
image.add(text);
amount = new JTextField();
amount.setFont(new Font("Raleway", Font.BOLD,22));
amount.setBounds(170,350,320,25);
image.add(amount);
deposit = new JButton("Deposit");
deposit.setBounds(355,485,150,30);
deposit.addActionListener(this);
image.add(deposit);
back = new JButton("Back");
back.setBounds(355,520,150,30);
back.addActionListener(this);
image.add(back);
  setSize(900,900);
  setLocation(300,0);
  setVisible(true);
            }
public void actionPerformed(ActionEvent ae){
    if(ae.getSource() == deposit){
        String number = amount.getText();
        Date date = new Date();
        if(number.equals("")){
                JOptionPane.showMessageDialog(null,"Please enter the amount you want to
deposit");
            }
            else{
                try{
                conn con1 = new conn();
                String query = "INSERT INTO bank (pin  ,date , type , amount) VALUES
('"+pinnumber+"', '"+date+"','Deposit', '"+number+"');";
                int i=con1.s.executeUpdate(query);
                JOptionPane.showMessageDialog(null,"Rs"+number+"Deposited Successfully");
                setVisible(false);
                new Transactions(pinnumber).setVisible(true);
                System.out.print(i);

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

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

BalanceEnquiry.java

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

public class BalanceEnquiry extends JFrame implements ActionListener{
    String pinnumber;
    JButton back;
    BalanceEnquiry(String pinnumber){
        this.pinnumber=pinnumber;
     setLayout(null);

    ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/atm.jpg"));
    Image i2 =i1.getImage().getScaledInstance(900,900,Image.SCALE_DEFAULT);
    ImageIcon i3 = new ImageIcon(i2);
    JLabel image = new JLabel(i3);
    image.setBounds(0,0,900,900);
    add(image);
    back = new JButton("Back");
    back.setBounds(355,520,150,30);
    back.addActionListener(this);
    image.add(back);
    conn c = new conn();
    int balance = 0;
         try{
            String query2 =" select * from bank where pin = '"+pinnumber+"'";
            ResultSet rs = c.s.executeQuery(query2);
            while(rs.next()){
                if(rs.getString("type").equals("Deposit")){
                    balance +=Integer.parseInt(rs.getString("amount"));
                }else{
                    balance -= Integer.parseInt(rs.getString("amount"));
                }
            }
        }
        catch(Exception e){
            System.out.println(e);
        }
        JLabel text = new JLabel("Your current Account balance is Rs :"+balance);
        text.setForeground(Color.white);
        text.setBounds(170,300,400,30);
        image.add(text);

      setSize(900,900);
      setLocation(300,0);
      setUndecorated(true);
      setVisible(true);
    }
        public void actionPerformed(ActionEvent ae){
        setVisible(false);
        new Transactions(pinnumber).setVisible(true);
    }
    public static void main(String[] args) {
        new BalanceEnquiry("");
    }

}
```

FastCash.java

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

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
public class BalanceEnquiry extends JFrame implements ActionListener{
    String pinnumber;
    JButton back;
    BalanceEnquiry(String pinnumber){
        this.pinnumber=pinnumber;

     setLayout(null);
    ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/atm.jpg"));
    Image i2 =i1.getImage().getScaledInstance(900,900,Image.SCALE_DEFAULT);
    ImageIcon i3 = new ImageIcon(i2);
    JLabel image = new JLabel(i3);
    image.setBounds(0,0,900,900);
    add(image);
    back = new JButton("Back");
    back.setBounds(355,520,150,30);
    back.addActionListener(this);
    image.add(back);
    conn c = new conn();
    int balance = 0;
         try{
             String query2 =" select * from bank where pin = '"+pinnumber+"'";
             ResultSet rs = c.s.executeQuery(query2);
             while(rs.next()){
                 if(rs.getString("type").equals("Deposit")){
                     balance +=Integer.parseInt(rs.getString("amount"));
                 }else{
                     balance -= Integer.parseInt(rs.getString("amount"));
```

```
  }
          }
       }
       catch(Exception e){
          System.out.println(e);
       }
      JLabel text = new JLabel("Your current Account balance is Rs :"+balance);
      text.setForeground(Color.white);
      text.setBounds(170,300,400,30);
      image.add(text);
    setSize(900,900);
    setLocation(300,0);
    setUndecorated(true);
    setVisible(true);
   }

   public void actionPerformed(ActionEvent ae){
      setVisible(false);
      new Transactions(pinnumber).setVisible(true);
   }
   public static void main(String[] args) {
      new BalanceEnquiry("");
   }

}
```

MiniStatement.java

```
package bank.management.system;

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
public class BalanceEnquiry extends JFrame implements ActionListener{
    String pinnumber;
    JButton back;
    BalanceEnquiry(String pinnumber){
        this.pinnumber=pinnumber;
     setLayout(null);

    ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/atm.jpg"));
    Image i2 =i1.getImage().getScaledInstance(900,900,Image.SCALE_DEFAULT);
    ImageIcon i3 = new ImageIcon(i2);
    JLabel image = new JLabel(i3);
    image.setBounds(0,0,900,900);
    add(image);
    back = new JButton("Back");
    back.setBounds(355,520,150,30);
    back.addActionListener(this);
    image.add(back);
    conn c = new conn();
    int balance = 0;
         try{
             String query2 =" select * from bank where pin = '"+pinnumber+"'";
             ResultSet rs = c.s.executeQuery(query2);
             while(rs.next()){
                 if(rs.getString("type").equals("Deposit")){
                     balance +=Integer.parseInt(rs.getString("amount"));
                 }else{
                     balance -= Integer.parseInt(rs.getString("amount"));
                 }
             }
         }
         catch(Exception e){
             System.out.println(e);
         }
        JLabel text = new JLabel("Your current Account balance is Rs :"+balance);
        text.setForeground(Color.white);
        text.setBounds(170,300,400,30);
        image.add(text);
     setSize(900,900);
     setLocation(300,0);
     setUndecorated(true);
     setVisible(true);
    }
    public void actionPerformed(ActionEvent ae){
        setVisible(false);
        new Transactions(pinnumber).setVisible(true);
    }
    public static void main(String[] args) {
        new BalanceEnquiry("");
    }

}
```

PinChange.java

```
package bank.management.system;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
```

```
public class PinChange extends JFrame implements ActionListener{
    JPasswordField pin,repin;
    JLabel pintext,repintext;
     JButton change,back;
     String pinnumber;
    PinChange(String pinnumber){
        this.pinnumber= pinnumber;
        setLayout(null);
    ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/atm.jpg"));
    Image i2 =i1.getImage().getScaledInstance(900,900,Image.SCALE_DEFAULT);
    ImageIcon i3 = new ImageIcon(i2);
    JLabel image = new JLabel(i3);
    image.setBounds(0,0,900,900);
    add(image);
    JLabel text = new JLabel("CHANGE YOUR PIN");
    text.setFont(new Font("System", Font.BOLD,16));
    text.setForeground(Color.WHITE);
    text.setBounds(250,280,500,35);
    image.add(text);
    pintext = new JLabel("NEW PIN:");
    pintext.setFont(new Font("System", Font.BOLD,16));
    pintext.setForeground(Color.WHITE);
    pintext.setBounds(165,320,180,25);
    image.add(pintext);
    pin = new JPasswordField();
    pin.setFont(new Font("Raleway", Font.BOLD,25));
    pin.setBounds(330,320,180,25);
    image.add(pin);
    repintext = new JLabel("Re-Enter New PIN:");
    repintext.setFont(new Font("System", Font.BOLD,16));
    repintext.setForeground(Color.WHITE);
    repintext.setBounds(165,360,180,25);
    image.add(repintext);
    repin = new JPasswordField();
    repin.setFont(new Font("Raleway", Font.BOLD,25));
    repin.setBounds(330,360,180,25);
    image.add(repin);
    change = new JButton("CHANGE");
    change.setBounds(355,485,150,30);
    change.addActionListener(this);
    image.add(change);
    back = new JButton("Back");
    back.setBounds(355,520,150,30);
    back.addActionListener(this);
    image.add(back);
     setSize(900,900);
     setLocation(300,0);
     setUndecorated(true);
     setVisible(true);
    }
    public void actionPerformed(ActionEvent ae){
        if(ae.getSource()== change){
           try{
            String npin = pin.getText();
            String rpin = repin.getText();
            if(!npin.equals(rpin)){
                JOptionPane.showMessageDialog(null,"Entered PIN does not match");
                return;
            }
           if(npin.equals("")){
           JOptionPane.showMessageDialog(null,"Please enter PIN");
```

```
            return;
            }
            if(rpin.equals("")){
            JOptionPane.showMessageDialog(null,"Please re-enter PIN");
            return;
            }
            conn conn = new conn();
            String query1 = "Update bank set pin ='"+rpin+"' where pin = '"+pinnumber+"'";
            String query2 = "Update login set pin ='"+rpin+"' where pin = '"+pinnumber+"'";
            String query3 = "Update sin set pignupthree ='"+rpin+"' where pin = '"+pinnumber+"'";
            conn.s.executeUpdate(query1);
            conn.s.executeUpdate(query2);
            conn.s.executeUpdate(query3);
        JOptionPane.showMessageDialog(null," PIN changed succussfully");
        setVisible(false);
        new Transactions(rpin).setVisible(true);
        }catch(Exception e){
            e.printStackTrace();
        }
        }else{
            setVisible(false);
            new Transactions(pinnumber). setVisible(true);
        }
    }
    public static void main(String[] args) {
        new PinChange("").setVisible(true);
    }
}
```

Withdrawl.java

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

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
public class Withdrawl  extends JFrame implements ActionListener{
  JButton Withdraw,back;
   JTextField amount;
   String pinnumber;
   Withdrawl(String pinnumber){
   this.pinnumber= pinnumber;
   setLayout(null);
   ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/atm.jpg"));
   Image i2 =i1.getImage().getScaledInstance(900,900,Image.SCALE_DEFAULT);
   ImageIcon i3 = new ImageIcon(i2);
   JLabel image = new JLabel(i3);
   image.setBounds(0,0,900,900);
   add(image);
   JLabel text = new JLabel("Enter the amount you want to Withdraw");
   text.setFont(new Font("System", Font.BOLD,16));
   text.setForeground(Color.WHITE);
   text.setBounds(170,300,400,20);
   image.add(text);
   amount = new JTextField();
   amount.setFont(new Font("Raleway", Font.BOLD,22));
   amount.setBounds(170,350,320,25);
   image.add(amount);
   Withdraw = new JButton("Withdraw");
   Withdraw.setBounds(355,485,150,30);
   Withdraw.addActionListener(this);
   image.add(Withdraw);
   back = new JButton("Back");
```

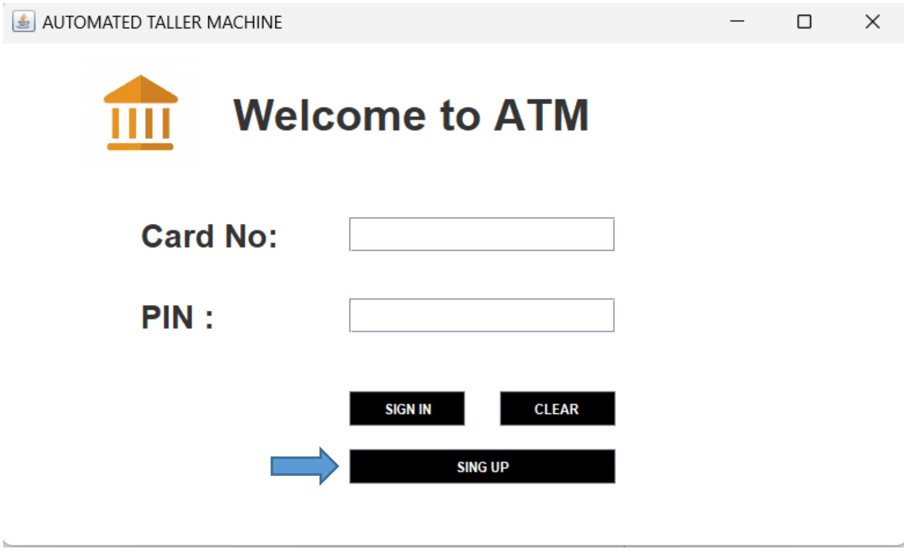
```
    back.setBounds(355,520,150,30);
    back.addActionListener(this);
    image.add(back);
       setSize(900,900);
      setLocation(300,0);
      setVisible(true);
        }
    public void actionPerformed(ActionEvent ae){
        if(ae.getSource() == Withdraw){
            String number = amount.getText();
            Date date = new Date();
            if(number.equals("")){
                JOptionPane.showMessageDialog(null,"Please enter the amount you want to
Withdraw");
            }
            else{
                try{
                conn con1 = new conn();
                String query = "INSERT INTO bank (pin  ,date , type , amount) VALUES
('"+pinnumber+"', '"+date+"','Withdrawl', '"+number+"');";
                int i=con1.s.executeUpdate(query);
                JOptionPane.showMessageDialog(null,"Rs "+number+" Withdraw Successfully");
                setVisible(false);
                new Transactions(pinnumber).setVisible(true);
                System.out.print(i);

                }catch (Exception e){
                    e.printStackTrace();
                }
            }
        }else if(ae.getSource() == back){
            setVisible(false);
         new Transactions(pinnumber).setVisible(true);
        }
    }
        public static void main(String[] args) {
            new Withdrawl("");
    }
}
```

CHAPTER 4

SCREENSHOTS

4.1 Admin Screenshot

(minimum 5 screenshots)

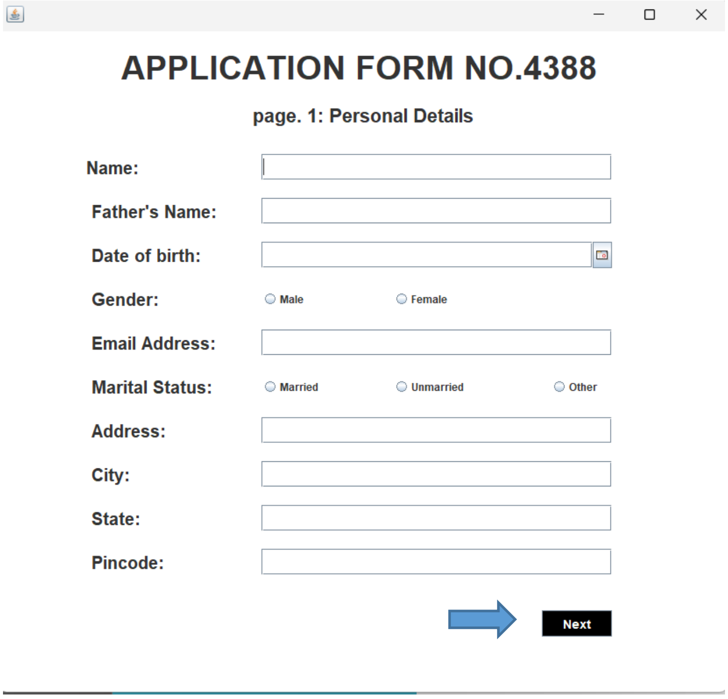


AUTOMATED TALLER MACHINE

Welcome to ATM

Card No:

PIN :



APPLICATION FORM NO.4388

page. 1: Personal Details

Name:

Father's Name:

Date of birth:

Gender: ○ Male ○ Female

Email Address:

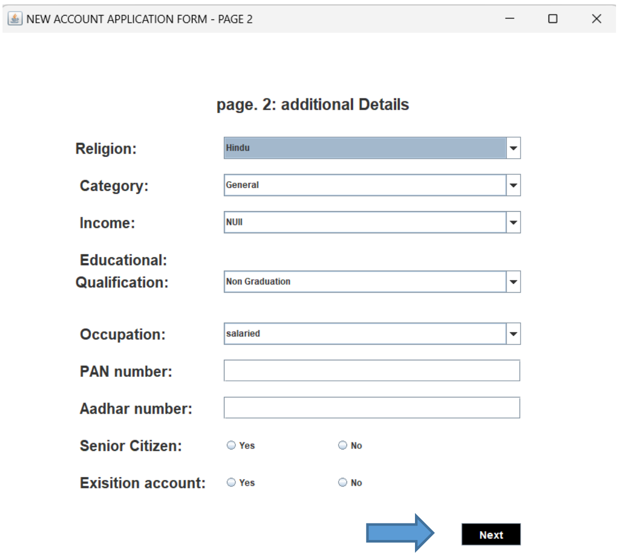
Marital Status: ○ Married ○ Unmarried ○ Other

Address:

City:

State:

Pincode:



NEW ACCOUNT APPLICATION FORM - PAGE 2

page. 2: additional Details

Religion: Hindu

Category: General

Income: NUll

Educational:
Qualification: Non Graduation

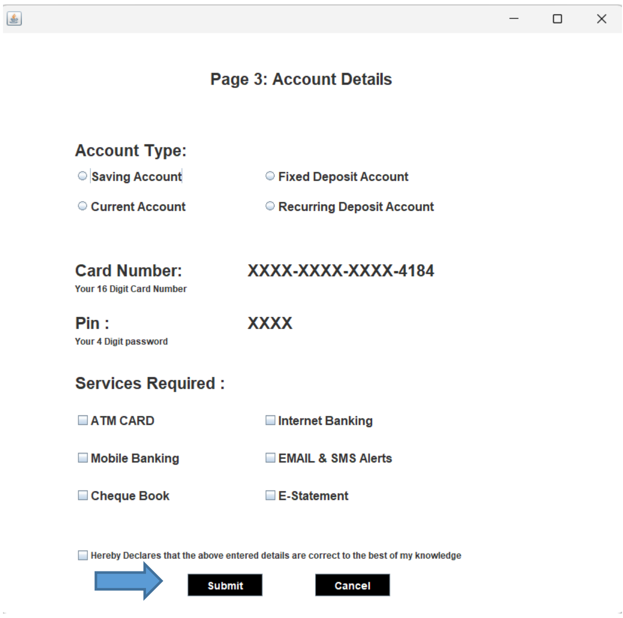
Occupation: salaried

PAN number:

Aadhar number:

Senior Citizen: ○ Yes ○ No

Exisition account: ○ Yes ○ No



Page 3: Account Details

Account Type:

○ Saving Account ○ Fixed Deposit Account

○ Current Account ○ Recurring Deposit Account

Card Number: XXXX-XXXX-XXXX-4184
Your 16 Digit Card Number

Pin : XXXX
Your 4 Digit password

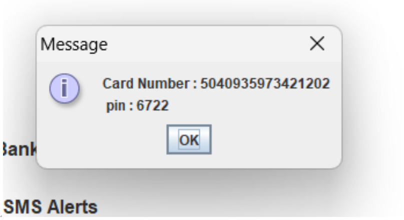
Services Required :

□ ATM CARD □ Internet Banking

□ Mobile Banking □ EMAIL & SMS Alerts

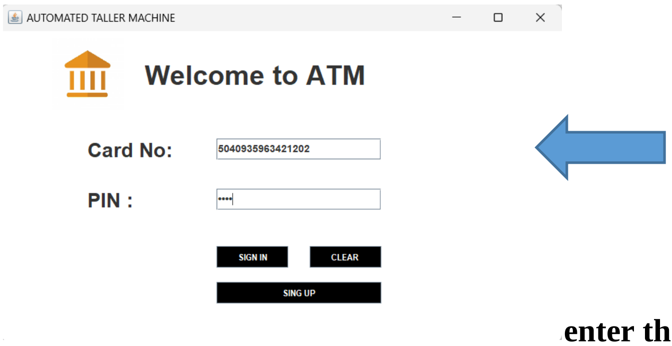
□ Cheque Book □ E-Statement

□ Hereby Declares that the above entered details are correct to the best of my knowledge



Message

Card Number : 5040935973421202
pin : 6722



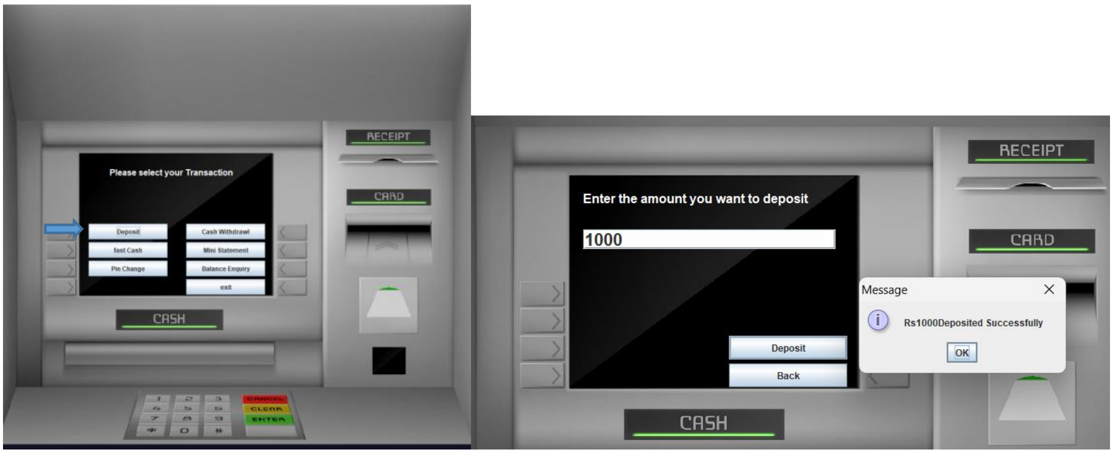
AUTOMATED TALLER MACHINE

Welcome to ATM

Card No: 5040935963421202

PIN : ••••

enter the card number and pin



Enter the amount you want to deposit

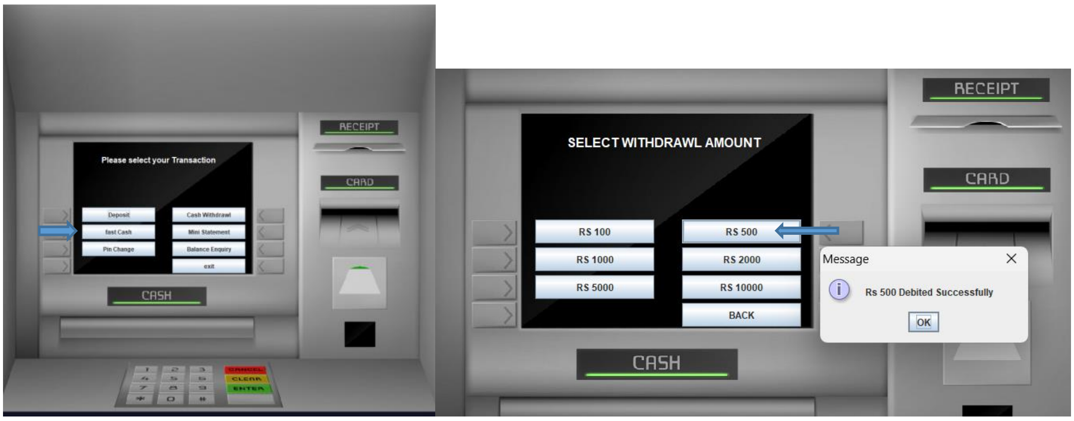
1000

Message

Rs1000Deposited Successfully

OK

Deposit

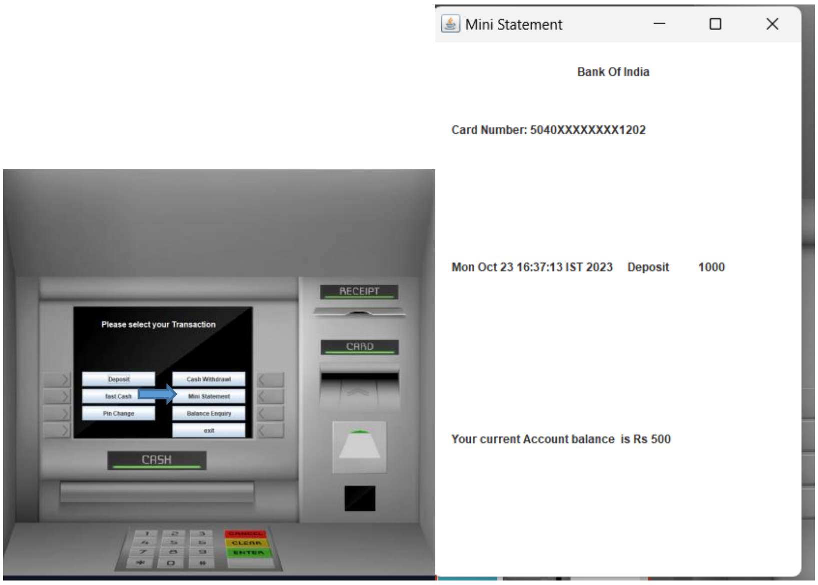


Message

Rs 500 Debited Successfully

OK

Withdrawl



Mini Statement

Bank Of India

Card Number: 5040XXXXXXXX1202

Mon Oct 23 16:37:13 IST 2023   Deposit   1000

Your current Account balance is Rs 500

Mini Statement

CHAPTER 5
CONCLUSION AND FUTURE ENHANCEMENTS

5.1 Conclusion

A Banking Management System (BMS) is a critical software application that helps financial institutions manage their operations efficiently. In the conclusion of a project related to a Banking Management System, you typically summarize the key points and outcomes of the project. Here are some elements you might include in your conclusion:

5.2 Future Enhancements

- **Advanced Data Analytics and Reporting:**
  Predictive Analytics: Implement predictive analytics capabilities to forecast market trends, customer behavior, and potential risks. This can help banks make more informed decisions and proactively manage their portfolios.
- **Real-time Reporting:** Enhance reporting tools to provide real-time insights into financial data, allowing for quicker and more data-driven decision-making.

  Customer Insights: Develop tools that analyze customer data to provide a deeper understanding of individual customer needs, enabling personalized banking services and marketing.
- **Enhanced Security Measures:**
  Biometric Authentication: Integrate advanced biometric authentication methods such as fingerprint recognition, facial recognition, or iris scanning to enhance the security of customer transactions and account access.
- **Blockchain Technology**: Explore the use of blockchain for secure and transparent record-keeping. Implementing blockchain can increase trust, reduce fraud, and streamline transaction processes.
- **AI-Driven Fraud Detection:** Develop machine learning models to detect and prevent fraudulent activities in real-time. These models can learn from historical data to identify patterns indicative of fraud.

REFERENCES

<In IEEE Format – Add minimum 5 references>

[1] Axelrod, CW (2013). Managing the Risks of Cyber-Physical Systems. In 2013 Systems, Applications and Technology Conf. (LISAT), pp. 1–6. IEEE: Long Island.

[2] www.researchgate.com/doc/library_management

Intelligent Networks — Telecommunications Solutions for the 1990s by Dr. Dimitris N. Chorafas and Heinrich Steinmann.
Java 2, The Complete Reference by Patrick Naughton and Herbert Schildt.
[3]
Web site of the Institute for Telecommunication Services, [online] Available: http://glossary.its. bldrdoc.gov/.

[4]. Intelligent Networks by Jan Thorner
.
[5].S. Ghosh, Q. Razouqi, H. J. Schumacher and A. celmins, "A Survey of Recent Advances in Fuzzy Logic in Telecommunication Networks and New Challenges", *IEEE Transactions On Fuzzy Systems*, vol. 6, no. 3, August 1998.