A
Project Report
On

“Banking Management System”

Submitted in partial fulfillment of
the requirements for the 7th Semester Sessional Examination of

BACHELOR OF TECHNOLOGY
IN

Computer Science and Engineering

By

ADITYA KUMAR GOSWAMI (21UG010124)
SAKET KUMAR GUPTA (21UG010560)
AKHILESH KUMAR SHEKHAR(21UG010142)

Under the esteemed guidance of

MR DEBASISH SAHOO

SCHOOL OF ENGINEERING AND TECHNOLOGY
Department of Computer Science and Engineering
GIET University, GUNUPUR – 765022
2024

GIET UNIVERSITY

GUNUPUR - 765022, ODISHA, www.giet.edu

CERTIFICATE

This is to certify that the project work entitled "Banking Management System" is done by Aditya Kumar Goswami- (21UG010124.)- , Saket Kumar Gupta-(21UG010560.)-,Akhilesh Kumar Shekhar (21UG010142)- in partial fulfillment of the requirements for the 7th Semester Sessional Examination of Bachelor of Technology in Computer Science and Engineering during the academic year 2024-25. This work is submitted to the department as a part of evaluation of 7th Semester Major Project-1.

Debashish Sahoo
Project Supervisor

Dr.Sachikanta Dash
HoD, 4th Yr.

Dr. K Murali Gopal
Dy. Dean, SOET

ACKNOWLEDGEMENT

We express our sincere gratitude to our project guide Prof Debasish Sahoo of Computer science and engineering for giving me an opportunity to accomplish the project. Without his active support and guidance, this project report has been successfully completed.
We also thanks **Debasish Sahoo**, our Class Teacher, Dr. **Sachikanta Dash**, Head of the department of computer science and Dr. **K Murali Gopal**, Dy. Dean, SOET for their consistent support, guidance and help.

Aditya Kumar Goswami (21UG010124)
Saket Kumar Gupta (21UG010560)
Akhilesh Kumar Shekhar (21UG010142)

ABSTRACT

The Banking Management System (BMS) is a software solution developed to manage various banking operations with greater efficiency, security, and accuracy. As banks handle millions of transactions and customer details daily, a robust system is crucial to manage data securely and streamline processes, such as account management, customer service, transactions, loans, and reporting. This system facilitates the digital transformation of traditional banking by automating processes and enhancing user experience.

The BMS centralizes customer information and provides a secure platform to manage data, ensuring compliance with regulatory standards and safeguarding against unauthorized access. Key features include a user-friendly interface for bank employees, online access for customers to view account balances, transfer funds, pay bills, and manage investments, as well as tools for loan processing, interest calculations, and financial forecasting. The system also incorporates security protocols, such as encryption, multi-factor authentication, and regular audits, to prevent data breaches and fraudulent activities.

The implementation of this Banking Management System aims to reduce operational costs, increase efficiency, and deliver a higher level of customer satisfaction. By automating routine tasks, the BMS allows banking staff to focus on value-added services, improving decision-making and facilitating better customer relationships. Through modular and scalable architecture, the system is adaptable to the evolving needs of the banking sector, supporting integration with new technologies, compliance requirements, and digital banking innovations.

CONTENT

Chapter-1

INTRODUCTION

1.1 PURPOSE

The purpose of a **Banking Management System (BMS)** is to provide a comprehensive, efficient, and secure platform for managing the daily operations and services of a bank. The system is designed to streamline banking processes, improve customer satisfaction, and ensure data security and compliance with regulatory standards. Here are the primary purposes:

1. **Efficient Data Management:** BMS centralizes customer information, account details, and transaction histories, making it easy to retrieve, update, and manage data without manual intervention.
2. **Transaction Automation:** It automates common banking transactions, such as deposits, withdrawals, and fund transfers, reducing processing times and minimizing errors.
3. **Enhanced Customer Experience:** The system offers an online interface for customers to access their accounts, view balances, transfer funds, apply for loans, and pay bills, enhancing convenience and accessibility.
4. **Security and Compliance:** BMS includes encryption, multi-factor authentication, and audit trails to secure sensitive data and comply with regulatory requirements, protecting against fraud and unauthorized access.
5. **Operational Efficiency:** By automating routine tasks like interest calculations, loan processing, and report generation, BMS reduces the need for manual work, lowering operational costs and improving productivity.
6. **Better Decision-Making:** The system provides bank managers and decision-makers with real-time data and analytics on financial performance, customer trends, and risk factors, enabling more informed and strategic decisions.
7. **Scalability and Adaptability:** BMS is modular and scalable, allowing banks to add new features, integrate with emerging technologies, and adapt to changing regulatory requirements or customer needs over time.

Overall, the Banking Management System aims to enhance the efficiency, security, and quality of banking services while providing customers with a seamless digital banking experience

1.2 SCOPE

scope of a Banking Management System (BMS) encompasses a broad range of functionalities aimed at managing all core aspects of banking operations. A well-designed BMS can support various banking activities and stakeholders, including customers, employees, and administrators. Here is an outline of the scope:

1. **Customer Account Management:**
   - Handling all types of customer accounts (savings, current, fixed deposits).
   - Managing customer profiles, contact information, and KYC (Know Your Customer) documentation.
   - Providing secure, real-time access to account information, transaction history, and statements.
2. **Transaction Management:**
   - Enabling and recording all types of transactions, such as deposits, withdrawals, fund transfers, bill payments, and foreign exchanges.
   - Automating processes for faster, error-free transactions.
   - Ensuring real-time updates and monitoring of account balances.
3. **Loan and Credit Processing:**
   - Facilitating loan applications, processing, approvals, and disbursement.
   - Tracking repayment schedules, interest calculations, and overdue payments.
   - Offering credit scoring and risk assessment tools for accurate decision-making.
4. **Security and Compliance:**
   - Ensuring data security through encryption, access controls, and securecommunication protocols.
   - Complying with regulatory requirements (e.g., anti-money laundering, dataprotection laws) and conducting regular audits.
   - Providing fraud detection, alert mechanisms, and multi-factorauthentication for customer accounts.
5. **Online and Mobile Banking:**
   - Offering digital access to customers via online and mobile platforms forremote banking.
   - Providing functionalities like viewing account information, fund transfers,bill payments, and transaction notifications.
   - Supporting features like card management, mobile deposits.

1.3 PROJECT FEATURES

1. User Authentication and Authorization

- Multi-factor authentication (MFA) and password protection.
- Role-based access control (e.g., admin, employee, customer) to manage permissions and restrict unauthorized access.

2. Customer Account Management

- Account creation and profile management, including savings, checking, and fixed deposit accounts.
- Updating KYC (Know Your Customer) details and managing customer contact information.
- Viewing account balances, transaction history, and generating account statements.

3. Transaction Management

- Processing and tracking deposits, withdrawals, fund transfers, and direct debits.
- Enabling real-time updates of account balances and transaction status.
- Handling recurring transactions, such as automatic payments and standing orders.

4. Loan and Credit Management

- Online loan application and approval workflow, with document management for submitted applications.
- Interest calculation, repayment schedule generation, and overdue notifications.
- Credit scoring and risk assessment tools to evaluate borrower eligibility.

5. Security and Fraud Detection

- Data encryption, secure communication protocols (e.g., HTTPS), and regular security audits.
- Fraud detection tools to flag unusual activities, with automated alerts for suspicious transactions.
- User activity logs and audit trails for compliance and monitoring.

Chapter-2

SYSTEM ANALYSIS

2.1 USER REQUIREMENTS

User requirements for a **Banking Management System (BMS)** outline the core needs and expectations of both customers and banking staff, ensuring the system delivers a smooth, secure, and efficient banking experience. Here’s a breakdown of the primary user requirements:

1. Account Management

- Customers should be able to create, view, and manage their accounts, such assavings, checking, or fixed deposits.
- Staff should be able to create and manage customer accounts, update details, and handle account closures as needed.
- The system should support customer profile management, including contact details,KYC documentation, and account preferences.

2. User Authentication and Security

- The system should implement multi-factor authentication (MFA) for customer loginto enhance security.
- Role-based access control for employees to restrict permissions based on roles (e.g.,teller, manager, admin).
- Security features like encryption, activity logging, and automatic session timeouts for data protection.

3. Transaction Management

- Customers should be able to conduct various transactions, such as deposits,withdrawals, and fund transfers, easily and securely.
- Real-time transaction processing and balance updates are essential for an accurate view of funds.
- Employees should have the ability to view, approve, and monitor transactions, withaudit trails for accountability.

4. Loan and Credit Management

- Customers should be able to apply for loans online, track the application status, and view loan terms and repayment schedules.
- Employees need tools for processing loan applications, performing creditassessments, calculating interest, and managing repayments.

2.2 FUNCTIONAL REQUIREMENTS

Functional requirements for a **Banking Management System (BMS)** specify the core functionalities the system must support to meet user needs and operational goals. Here's a breakdown of key functional requirements:

1. User Authentication and Authorization

- Provide secure login with multi-factor authentication (MFA) for customers andemployees.
- Implement role-based access control (RBAC) to restrict access to specificfunctionalities based on user roles (e.g., admin, teller, customer).
- Enable password recovery, account lockout after failed attempts, and secure logout.

2. Customer Account Management

- Support account creation, modification, and deletion for various types (e.g., savings, checking, fixed deposit).
- Allow customers to view and update their profile, contact information, and accountsettings.
- Generate account statements, transaction history, and summary reports forcustomer access.

3. Transaction Management

- Facilitate deposits, withdrawals, fund transfers (internal and external), and balanceinquiries.
- Provide real-time transaction processing with balance updates.
- Support recurring transactions, such as scheduled transfers and standing orders.
- Generate unique transaction IDs, maintain transaction logs, and support audit trails.

4. Loan and Credit Management

- Allow customers to apply for loans, view loan status, and access repaymentschedules.
- Support loan processing, including verification, approval, disbursement, andinterest calculation.
- Track loan repayments, send reminders for upcoming payments, and calculatepenalties for overdue accounts.
- Integrate credit scoring and risk assessment for evaluating loan eligibility.

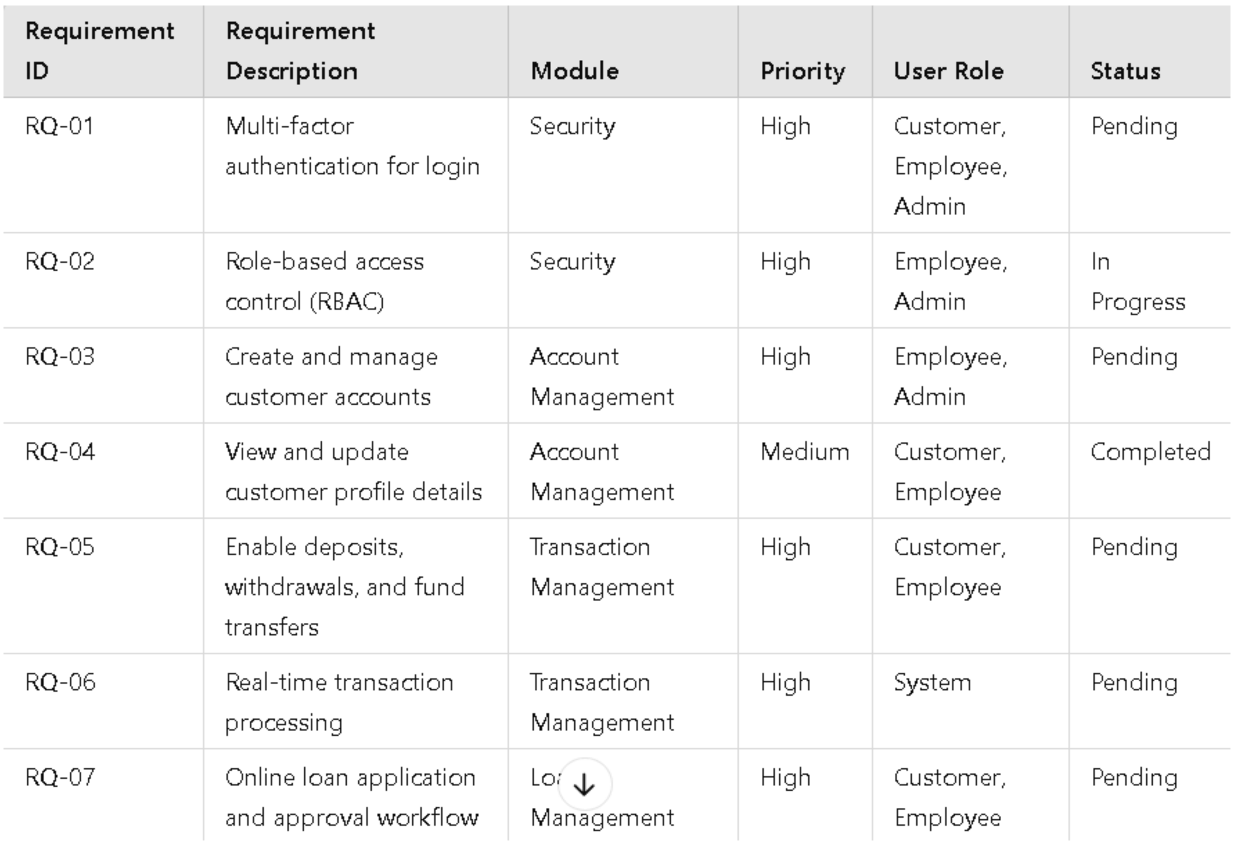
5. Online and Mobile Banking Services

- Provide web and mobile access to customer accounts, balances, and transaction history.

- Support mobile functionalities, such as quick balance checks, mobile payments, and online fund transfers.
- Allow secure session management with automatic logout and idle session timeout.
- Send real-time notifications and alerts for important account activities

2.3 REQUIREMENT MATRIX

A Requirement Matrix (also known as a Requirements Traceability Matrix) for a Banking Management System (BMS) helps organize and track system requirements by associating them with specific functional modules and user roles. It allows project stakeholders to verify that all requirements are fulfilled in the design, development, and testing phases. Below is a sample requirements matrix for a BMS:



Requirement ID	Requirement Description	Module	Priority	User Role	Status
RQ-01	Multi-factor authentication for login	Security	High	Customer, Employee, Admin	Pending
RQ-02	Role-based access control (RBAC)	Security	High	Employee, Admin	In Progress
RQ-03	Create and manage customer accounts	Account Management	High	Employee, Admin	Pending
RQ-04	View and update customer profile details	Account Management	Medium	Customer, Employee	Completed
RQ-05	Enable deposits, withdrawals, and fund transfers	Transaction Management	High	Customer, Employee	Pending
RQ-06	Real-time transaction processing	Transaction Management	High	System	Pending
RQ-07	Online loan application and approval workflow	Lo Management	High	Customer, Employee	Pending

Key:

- **Module**: The area of functionality to which the requirement belongs (e.g., Security, Account Management).

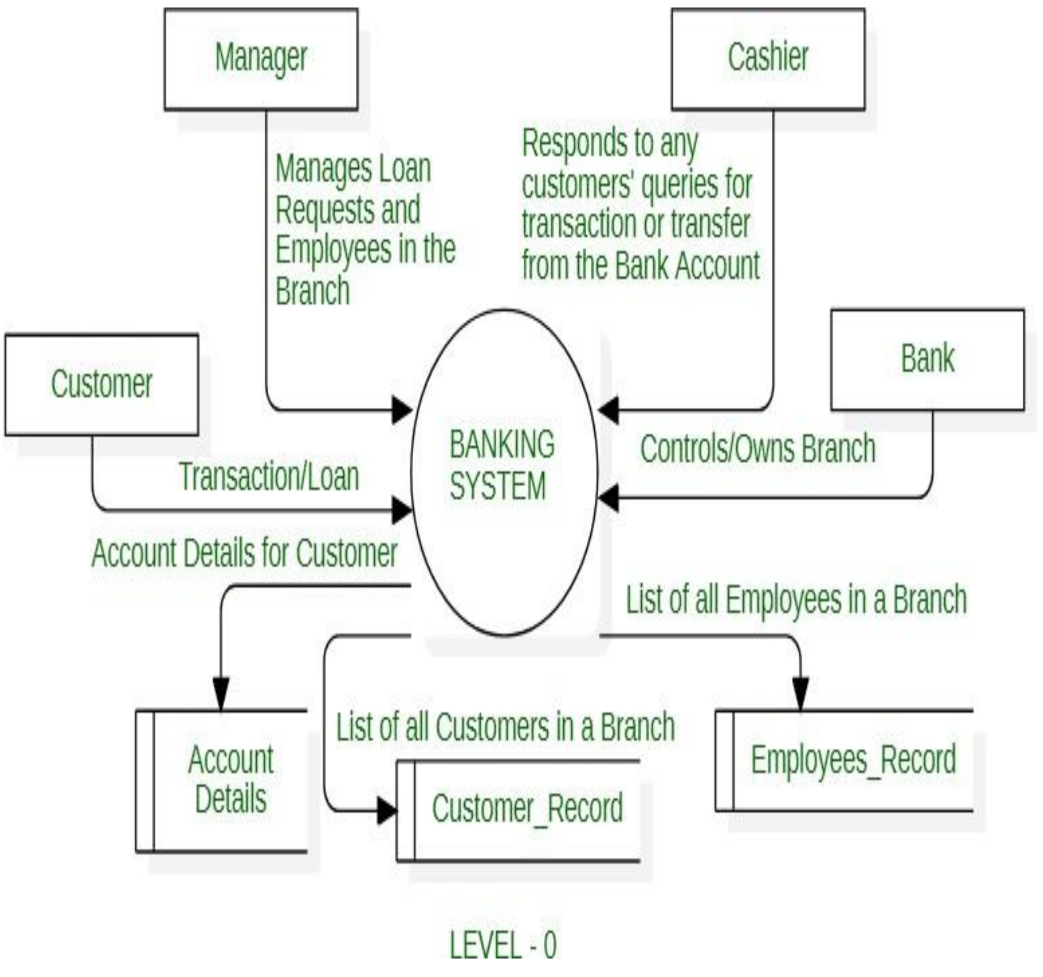
- **Priority**: Importance of the requirement, typically "High," "Medium," or "Low."
- **User Role**: The primary user or system responsible for the requirement (e.g., Customer, Employee, Admin).
- **Status**: Current implementation stage (e.g., Pending, In Progress, Completed).

This matrix can be expanded with additional columns for **Test Cases**, **Acceptance Criteria**, or **Completion Date** to further track requirement fulfillment during project phases.

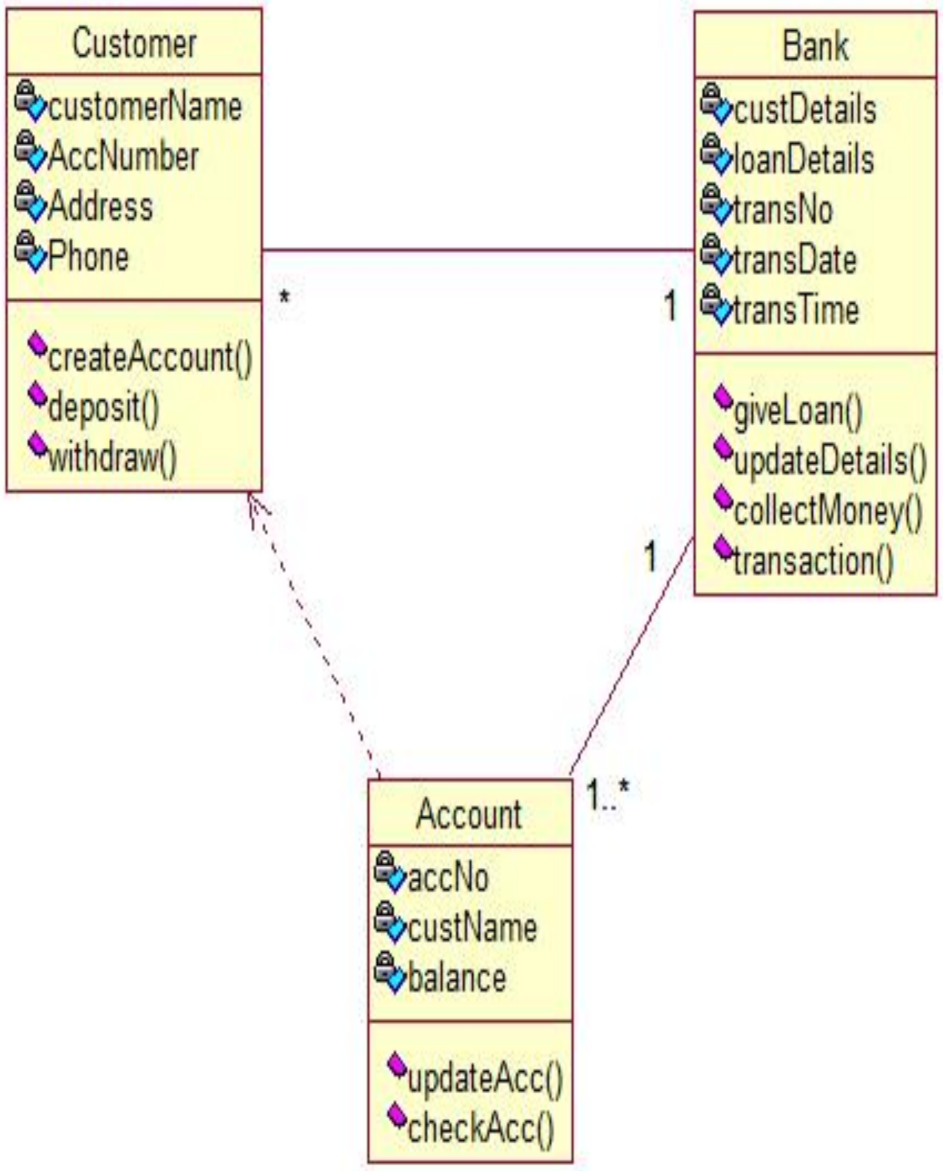
Chapter-3

SYSTEM DESIGN AND SPECIFICATION

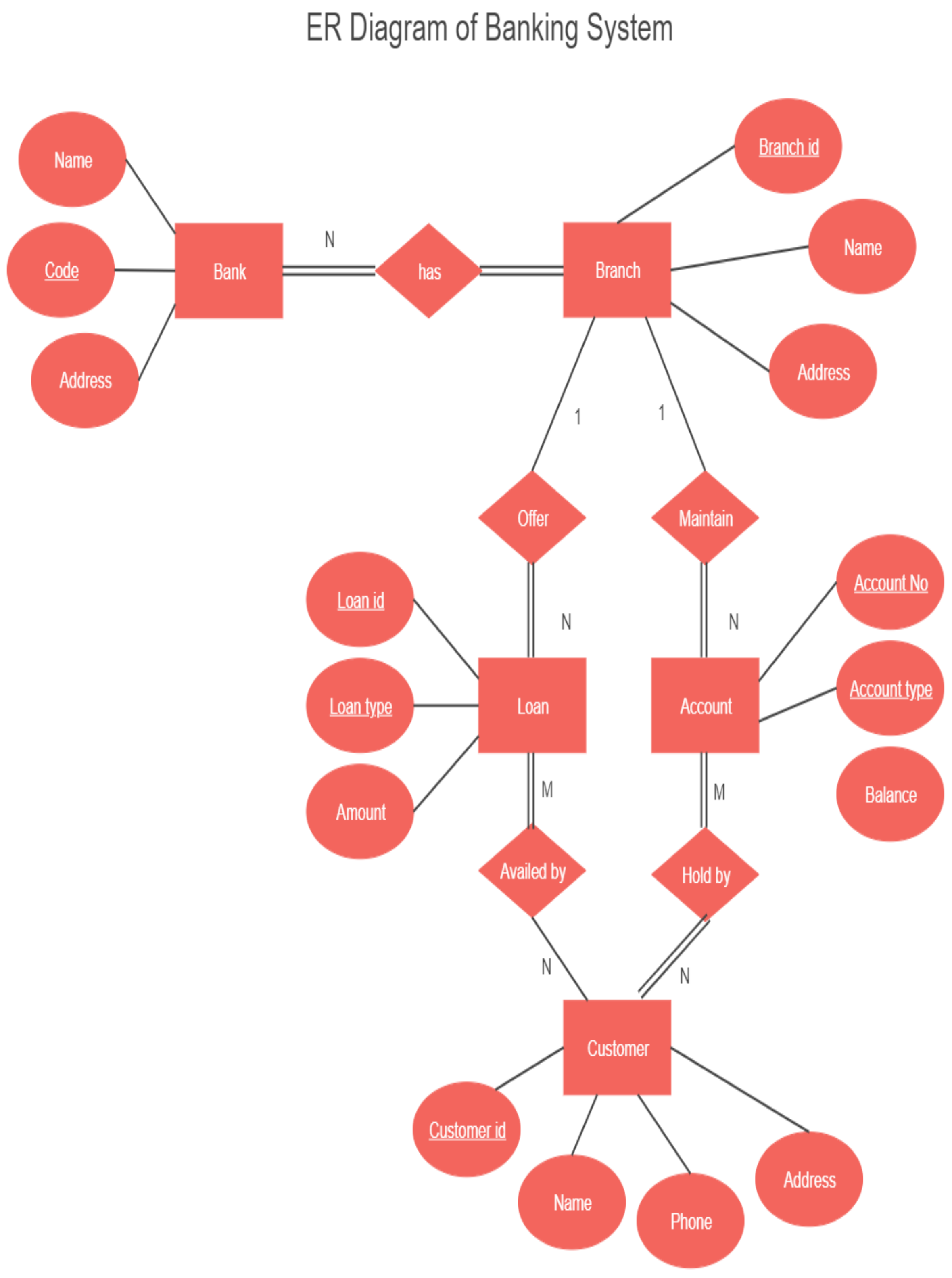
3.1 HIGH LEVEL DESIGN


Manager
Cashier
Manages Loan Requests and Employees in the Branch
Responds to any customers' queries for transaction or transfer from the Bank Account
Customer
Bank
BANKING SYSTEM
Transaction/Loan
Controls/Owns Branch
Account Details for Customer
List of all Employees in a Branch
List of all Customers in a Branch
Account Details
Customer_Record
Employees_Record
LEVEL - 0

3.2 URL DESIGN



3.3 UX DESIGN


ER Diagram of Banking System
Name
Code
Address
Bank
N
has
Branch
Branch id
Name
Address
1
1
Offer
Maintain
Loan id
Loan type
Amount
N
Loan
N
Account
Account No
Account type
Balance
M
M
Availed by
Hold by
N
N
Customer
Customer id
Name
Phone
Address

Chapter-4

CODING

```
************ LOGIN PAGE ************
package bank.management.system;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;


public class Login extends JFrame implements ActionListener {

    JLabel label1, label2, label3;
    JTextField textField2;
    JPasswordField passwordField3;

    JButton button1,button2,button3;
    Login(){
      super("Bank Management System");
      ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icon/bank.png"));
      Image i2 = i1.getImage().getScaledInstance(100,100,Image.SCALE_DEFAULT);
      ImageIcon i3 = new ImageIcon(i2);
      JLabel image = new JLabel(i3);
      image.setBounds(350,10,100,100);
      add(image);

      ImageIcon ii1 = new ImageIcon(ClassLoader.getSystemResource("icon/card.png"));
      Image ii2 = ii1.getImage().getScaledInstance(100,100,Image.SCALE_DEFAULT);
      ImageIcon ii3 = new ImageIcon(ii2);
      JLabel iimage = new JLabel(ii3);
      iimage.setBounds(630,350,100,100);
      add(iimage);

      label1 = new JLabel("WELCOME TO ATM");
      label1.setForeground(Color.WHITE);
      label1.setFont(new Font("AvantGarde", Font.BOLD, 38));
      label1.setBounds(230,125,450,40);
      add(label1);
```

```
label2 = new JLabel("Card No:");
label2.setFont(new Font("Ralway", Font.BOLD, 28));
label2.setForeground(Color.WHITE);
label2.setBounds(150,190,375,30);
add(label2);

textField2 = new JTextField(15);
textField2.setBounds(325,190,230,30);
textField2.setFont(new Font("Arial", Font.BOLD,14));
add(textField2);

label3 = new JLabel("PIN: ");
label3.setFont(new Font("Ralway", Font.BOLD, 28));
label3.setForeground(Color.WHITE);
label3.setBounds(150,250,375,30);
add(label3);

passwordField3 = new JPasswordField(15);
passwordField3.setBounds(325,250,230,30);
passwordField3.setFont(new Font("Arial", Font.BOLD, 14));
add(passwordField3);

button1 = new JButton("SIGN IN");
button1.setFont(new Font("Arial", Font.BOLD, 14));
button1.setForeground(Color.WHITE);
button1.setBackground(Color.BLACK);
button1.setBounds(300,300,100, 30);
button1.addActionListener(this);
add(button1);

button2 = new JButton("CLEAR");
button2.setFont(new Font("Arial", Font.BOLD, 14));
button2.setForeground(Color.WHITE);
button2.setBackground(Color.BLACK);
button2.setBounds(430,300,100, 30);
button2.addActionListener(this);
add(button2);

button3 = new JButton("SIGN UP");
button3.setFont(new Font("Arial", Font.BOLD, 14));
```

```
        button3.setForeground(Color.WHITE);
        button3.setBackground(Color.BLACK);
        button3.setBounds(300,350,230, 30);
        button3.addActionListener(this);
        add(button3);

        ImageIcon iii1 = new
ImageIcon(ClassLoader.getSystemResource("icon/backbg.png"));
        Image iii2 = iii1.getImage().getScaledInstance(850,480,Image.SCALE_DEFAULT);
        ImageIcon iii3 = new ImageIcon(iii2);
        JLabel iiimage = new JLabel(iii3);
        iiimage.setBounds(0,0,850,480);
        add(iiimage);



        setLayout(null);
        setSize(850,480);
        setLocation(450,200);
        setUndecorated(true);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        try{
            if (e.getSource()==button1){
                Connn c = new Connn();
                String cardno = textField2.getText();
                String pin = passwordField3.getText();
                String q = "select * from login where card_no = '"+cardno+"' and pin =
'"+pin+"'";
                ResultSet resultSet = c.statement.executeQuery(q);
                if (resultSet.next()) {
                    setVisible(false);
                    new main_Class(pin);

                }else {
                    JOptionPane.showMessageDialog(null,"Incorrect Card Number or PIN");
                }
            }else if (e.getSource() == button2){
                textField2.setText("");
                passwordField3.setText("");
            }else if (e.getSource() == button3){
                new Signup();
                setVisible(false);
            }
        }catch (Exception E){
            E.printStackTrace();
        }

    }

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

==

---------------------------------: PAGE 2 : -------------------------

************************** SIGN UP 3 ***************************

```
package bank.management.system;
import javax.print.attribute.standard.JobHoldUntil;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.rmi.server.ExportException;
import java.util.Random;

public class Signup3 extends JFrame implements ActionListener  {


    JRadioButton r1,r2,r3,r4;
    JCheckBox c1,c2,c3,c4,c5,c6;
    JButton s,c;
    String formno;
    Signup3(String formno){
```

```
this.formno = formno;

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icon/bank.png"));
Image i2 = i1.getImage().getScaledInstance(100,100,Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel image = new JLabel(i3);
image.setBounds(150,5,100,100);
add(image);

JLabel l1 = new JLabel("Page 3:");
l1.setFont(new Font("Raleway",Font.BOLD,22));
l1.setBounds(280,40,400,40);
add(l1);

JLabel l2 = new JLabel("Account Details");
l2.setFont(new Font("Raleway",Font.BOLD,22));
l2.setBounds(280,70,400,40);
add(l2);

JLabel l3 = new JLabel("Account Type:");
l3.setFont(new Font("Raleway",Font.BOLD,18));
l3.setBounds(100,140,200,30);
add(l3);

r1 = new JRadioButton("Saving Account");
r1.setFont(new Font("Raleway",Font.BOLD,16));
r1.setBackground(new Color(215,252,252));
r1.setBounds(100,180,150,30);
add(r1);

r2 = new JRadioButton("Fixed Deposit Account");
r2.setFont(new Font("Raleway",Font.BOLD,16));
r2.setBackground(new Color(215,252,252));
r2.setBounds(350,180,300,30);
add(r2);

r3 = new JRadioButton("Current Account");
r3.setFont(new Font("Raleway",Font.BOLD,16));
r3.setBackground(new Color(215,252,252));
r3.setBounds(100,220,250,30);
add(r3);
```

```
        r4 = new JRadioButton("Recurring Deposit Account");
        r4.setFont(new Font("Raleway",Font.BOLD,16));
        r4.setBackground(new Color(215,252,252));
        r4.setBounds(350,220,250,30);
        add(r4);

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

        JLabel l4 = new JLabel("Card Number:");
        l4.setFont(new Font("Raleway",Font.BOLD,18));
        l4.setBounds(100,300,200,30);
        add(l4);

        JLabel l5 = new JLabel("(Your 16-digit Card Number)");
        l5.setFont(new Font("Raleway",Font.BOLD,12));
        l5.setBounds(100,330,200,20);
        add(l5);

        JLabel l6 = new JLabel("XXXX-XXXX-XXXX-4841");
        l6.setFont(new Font("Raleway",Font.BOLD,18));
        l6.setBounds(330,300,250,30);
        add(l6);

        JLabel l7 = new JLabel("(It would appear on atm card/cheque Book and
Statements)");
        l7.setFont(new Font("Raleway",Font.BOLD,12));
        l7.setBounds(330,330,500,20);
        add(l7);

        JLabel l8 = new JLabel("PIN:");
        l8.setFont(new Font("Raleway",Font.BOLD,18));
        l8.setBounds(100,370,200,30);
        add(l8);

        JLabel    l9    =    new    JLabel("XXXX");
        l9.setFont(new Font("Raleway",Font.BOLD,18));
```

```
l9.setBounds(330,370,200,30);
add(l9);

JLabel l10 = new JLabel("(4-digit Password)");
l10.setFont(new Font("Raleway",Font.BOLD,12));
l10.setBounds(100,400,200,20);
add(l10);

JLabel l11 = new JLabel("Services Required:");
l11.setFont(new Font("Raleway",Font.BOLD,18));
l11.setBounds(100,450,200,30);
add(l11);

c1 = new JCheckBox("ATM CARD");
c1.setBackground(new Color(215,252,252));
c1.setFont(new Font("Raleway",Font.BOLD,16));
c1.setBounds(100,500,200,30);
add(c1);

c2 = new JCheckBox("Internet Banking");
c2.setBackground(new Color(215,252,252));
c2.setFont(new Font("Raleway",Font.BOLD,16));
c2.setBounds(350,500,200,30);
add(c2);

c3 = new JCheckBox("Mobile Banking");
c3.setBackground(new Color(215,252,252));
c3.setFont(new Font("Raleway",Font.BOLD,16));
c3.setBounds(100,550,200,30);
add(c3);

c4 = new JCheckBox("EMAIL Alerts");
c4.setBackground(new Color(215,252,252));
c4.setFont(new Font("Raleway",Font.BOLD,16));
c4.setBounds(350,550,200,30);
add(c4);

c5 = new JCheckBox("Cheque Book");
c5.setBackground(new Color(215,252,252));
c5.setFont(new Font("Raleway",Font.BOLD,16));
c5.setBounds(100,600,200,30);
```

```
        add(c5);

        c6 = new JCheckBox("E-Statement");
        c6.setBackground(new Color(215,252,252));
        c6.setFont(new Font("Raleway",Font.BOLD,16));
        c6.setBounds(350,600,200,30);
        add(c6);

        JCheckBox c7 = new JCheckBox("I here by decleares that the above entered details
correct to the best of my knlowledge.",true);
        c7.setBackground(new Color(215,252,252));
        c7.setFont(new Font("Raleway",Font.BOLD,12));
        c7.setBounds(100,680,600,20);
        add(c7);

        JLabel l12 = new JLabel("Form No : ");
        l12.setFont(new Font("Raleway", Font.BOLD,14));
        l12.setBounds(700,10,100,30);
        add(l12);

        JLabel l13 = new JLabel(formno);
        l13.setFont(new Font("Raleway", Font.BOLD,14));
        l13.setBounds(760,10,60,30);
        add(l13);



        s = new JButton("Submit");
        s.setFont(new Font("Raleway", Font.BOLD,14));
        s.setBackground(Color.BLACK);
        s.setForeground(Color.WHITE);
        s.setBounds(250,720,100,30);
        s.addActionListener(this);
        add(s);

        c = new JButton("Cancel");
        c.setFont(new Font("Raleway", Font.BOLD,14));
        c.setBackground(Color.BLACK);
        c.setForeground(Color.WHITE);
        c.setBounds(420,720,100,30);
        c.addActionListener(this);
        add(c);
        getContentPane().setBackground(new Color(215,252,252));
        setSize(850,800);
        setLayout(null);
        setLocation(400,20);
        setVisible(true);
    }



@Override
    public void actionPerformed(ActionEvent e) {
        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) + 1409963000000000L;
        String cardno = "" + Math.abs(first7);

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

        String fac = "";
        if(c1.isSelected()){
            fac = fac+"ATM CARD ";
        } else if (c2.isSelected()) {
            fac = fac+"Internet Banking";
        } else if (c3.isSelected()) {
            fac = fac+"Mobile Banking";
        } else if (c4.isSelected()) {
            fac = fac+"EMAIL Alerts";
        } else if (c5.isSelected()) {
            fac=fac+"Cheque Book";
        } else if (c6.isSelected()) {
            fac=fac+"E-Statement";
        }
        try {
            if (e.getSource()==s){
                if (atype.equals("")){
                    JOptionPane.showMessageDialog(null,"Fill all the fields");
                }else {
                    Connn c1 = new Connn();
                    String q1 = "insert into signupthree values('"+formno+"',
'"+atype+"','"+cardno+"','"+pin+"','"+fac+"')";
                    String q2 = "insert into login
values('"+formno+"','"+cardno+"','"+pin+"')";
                    c1.statement.executeUpdate(q1);
                    c1.statement.executeUpdate(q2);
                    JOptionPane.showMessageDialog(null,"Card Number : "+cardno+"\n Pin :
"+pin );
                    new Deposit(pin);
                    setVisible(false);
                }
            } else if (e.getSource()==c) {
                System.exit(0);
            }

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

    }

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

==

--------: PAGE 3 : ----

******************** SIGNUP 2 **********************

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


public class Signup2 extends JFrame implements ActionListener {


    JComboBox comboBox,comboBox2,comboBox3,comboBox4,comboBox5;
    JTextField textPan,textAadhar;
    JRadioButton r1,r2, e1,e2;
    JButton next;
    String formno;
    Signup2(String formno){
      super("APPLICATION FORM");

      ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icon/bank.png"));
      Image i2 = i1.getImage().getScaledInstance(100,100,Image.SCALE_DEFAULT);
      ImageIcon i3 = new ImageIcon(i2);
      JLabel image = new JLabel(i3);
      image.setBounds(150,5,100,100);
      add(image);

      this.formno = formno;

      JLabel l1 = new JLabel("Page 2 :-");
      l1.setFont(new Font("Raleway", Font.BOLD,22));
      l1.setBounds(300,30,600,40);
      add(l1);

      JLabel l2 = new JLabel("Additonal Details");
      l2.setFont(new Font("Raleway", Font.BOLD,22));
      l2.setBounds(300,60,600,40);
      add(l2);

      JLabel l3 = new JLabel("Religion :");
      l3.setFont(new Font("Raleway", Font.BOLD,18));
      l3.setBounds(100,120,100,30);
```

```
        add(l3);

        String religion[] = {"Hindu","Muslim","Sikh", "Christian", "Other"};
        comboBox = new JComboBox(religion);
        comboBox.setBackground(new Color(252,208,76));
        comboBox.setFont(new Font("Raleway",Font.BOLD,14));
        comboBox.setBounds(350,120,320,30);
        add(comboBox);

        JLabel l4 = new JLabel("Category : ");
        l4.setFont(new Font("Raleway", Font.BOLD,18));
        l4.setBounds(100,170,100,30);
        add(l4);

        String Category [] = {"General","OBC","SC", "ST", "Other"};
        comboBox2 = new JComboBox(Category);
        comboBox2.setBackground(new Color(252,208,76));
        comboBox2.setFont(new Font("Raleway",Font.BOLD,14));
        comboBox2.setBounds(350,170,320,30);
        add(comboBox2);

        JLabel l5 = new JLabel("Income : ");
        l5.setFont(new Font("Raleway", Font.BOLD,18));
        l5.setBounds(100,220,100,30);
        add(l5);

        String income [] = {"Null","<1,50,000","<2,50,000", "5,00,000", "Uptp
10,00,000","Above 10,00,000"};
        comboBox3 = new JComboBox(income);
        comboBox3.setBackground(new Color(252,208,76));
        comboBox3.setFont(new Font("Raleway",Font.BOLD,14));
        comboBox3.setBounds(350,220,320,30);
        add(comboBox3);

        JLabel l6 = new JLabel("Educational : ");
        l6.setFont(new Font("Raleway", Font.BOLD,18));
        l6.setBounds(100,270,150,30);
        add(l6);

        String educational [] = {"Non-Graduate","Graduate","Post-Graduate",
"Doctrate", "Others"};
```

```
        comboBox4 = new JComboBox(educational);
        comboBox4.setBackground(new Color(252,208,76));
        comboBox4.setFont(new Font("Raleway",Font.BOLD,14));
        comboBox4.setBounds(350,270,320,30);
        add(comboBox4);



        JLabel l7 = new JLabel("Occupation : ");
        l7.setFont(new Font("Raleway", Font.BOLD,18));
        l7.setBounds(100,340,150,30);
        add(l7);

        String Occupation [] = {"Salaried","Self-Employed","Business", "Student",
"Retired", "Other"};
        comboBox5 = new JComboBox(Occupation);
        comboBox5.setBackground(new Color(252,208,76));
        comboBox5.setFont(new Font("Raleway",Font.BOLD,14));
        comboBox5.setBounds(350,340,320,30);
        add(comboBox5);

        JLabel l8 = new JLabel("PAN Number : ");
        l8.setFont(new Font("Raleway", Font.BOLD,18));
        l8.setBounds(100,390,150,30);
        add(l8);

        textPan = new JTextField();
        textPan.setFont(new Font("Raleway", Font.BOLD,18));
        textPan.setBounds(350,390,320,30);
        add(textPan);

        JLabel l9 = new JLabel("Aadhar Number : ");
        l9.setFont(new Font("Raleway", Font.BOLD,18));
        l9.setBounds(100,440,180,30);
        add(l9);

        textAadhar = new JTextField();
        textAadhar.setFont(new Font("Raleway", Font.BOLD,18));
        textAadhar.setBounds(350,440,320,30);
        add(textAadhar);
```

```
JLabel l10 = new JLabel("Senior Citizen : ");
l10.setFont(new Font("Raleway", Font.BOLD,18));
l10.setBounds(100,490,180,30);
add(l10);

r1 = new JRadioButton("Yes");
r1.setFont(new Font("Raleway", Font.BOLD,14));
r1.setBackground(new Color(252,208,76));
r1.setBounds(350,490,100,30);
add(r1);
r2 = new JRadioButton("No");
r2.setFont(new Font("Raleway", Font.BOLD,14));
r2.setBackground(new Color(252,208,76));
r2.setBounds(460,490,100,30);
add(r2);

JLabel l11 = new JLabel("Existing Account : ");
l11.setFont(new Font("Raleway", Font.BOLD,18));
l11.setBounds(100,540,180,30);
add(l11);

e1 = new JRadioButton("Yes");
e1.setFont(new Font("Raleway", Font.BOLD,14));
e1.setBackground(new Color(252,208,76));
e1.setBounds(350,540,100,30);
add(e1);
e2 = new JRadioButton("No");
e2.setFont(new Font("Raleway", Font.BOLD,14));
e2.setBackground(new Color(252,208,76));
e2.setBounds(460,540,100,30);
add(e2);

JLabel l12 = new JLabel("Form No : ");
l12.setFont(new Font("Raleway", Font.BOLD,14));
l12.setBounds(700,10,100,30);
add(l12);

JLabel l13 = new JLabel(formno);
l13.setFont(new Font("Raleway", Font.BOLD,14));
l13.setBounds(760,10,60,30);
add(l13);
```

```
    next = new JButton("Next");
    next.setFont(new Font("Raleway",Font.BOLD,14));
    next.setBackground(Color.WHITE);
    next.setForeground(Color.BLACK);
    next.setBounds(570,640,100,30);
    next.addActionListener(this);
    add(next);



    setLayout(null);
    setSize(850,750);
    setLocation(450,80);
    getContentPane().setBackground(new Color(252, 208, 76));
    setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    String rel = (String) comboBox.getSelectedItem();
    String cate = (String) comboBox2.getSelectedItem();
    String inc = (String) comboBox3.getSelectedItem();
    String edu = (String) comboBox4.getSelectedItem();
    String occ = (String) comboBox5.getSelectedItem();

    String pan = textPan.getText();
    String addhar = textAadhar.getText();

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

```
            try{
               if (textPan.getText().equals("") || textAadhar.getText().equals("")){
                  JOptionPane.showMessageDialog(null,"Fill all the fields");
               }else {
                  Connn c = new Connn();
                  String q = "insert into Signuptwo values('"+formno+"', '"+rel+"',
'"+cate+"','"+inc+"','"+edu+"','"+occ+"','"+pan+"','"+addhar+"','"+scitizen+"','"+eAcc
ount+"')";
                  c.statement.executeUpdate(q);
                  new Signup3(formno);
                  setVisible(false);
               }


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



        }

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

=====================================================================
```

```
                --------------------------------:  PAGE 4  :--------------------------
       ******************** MAIN _ CLASS *********************
 package bank.management.system;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class main_Class  extends JFrame implements ActionListener{

        JButton b1,b2,b3,b4,b5,b6,b7,b8;
        String pin;
        main_Class(String pin){
```

```
this.pin = pin;

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icon/atm2.png"));
Image i2 = i1.getImage().getScaledInstance(1550,830,Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel l3 = new JLabel(i3);
l3.setBounds(0,0,1550,830);
add(l3);

JLabel label = new JLabel("Please Select Your Transaction");
label.setBounds(430,180,700,35);
label.setForeground(Color.WHITE);
label.setFont(new Font("System",Font.BOLD,28));
l3.add(label)
b1 = new JButton("DEPOSIT");
b1.setForeground(Color.WHITE);
b1.setBackground(new Color(65,125,128));
b1.setBounds(410,274,150,35);
b1.addActionListener(this);
l3.add(b1);

b2 = new JButton("CASH WITHDRAWL");
b2.setForeground(Color.WHITE);
b2.setBackground(new Color(65,125,128));
b2.setBounds(700,274,150,35);
b2.addActionListener(this);
l3.add(b2);

b3 = new JButton("FAST CASH");
b3.setForeground(Color.WHITE);
b3.setBackground(new Color(65,125,128));
b3.setBounds(410,318,150,35);
b3.addActionListener(this);
l3.add(b3);

b4 = new JButton("MINI STATEMENT");
b4.setForeground(Color.WHITE);
b4.setBackground(new Color(65,125,128));
b4.setBounds(700,318,150,35);
b4.addActionListener(this);
l3.add(b4);
```

```
    b5 = new JButton("PIN CHANGE");
    b5.setForeground(Color.WHITE);
    b5.setBackground(new Color(65,125,128));
    b5.setBounds(410,362,150,35);
    b5.addActionListener(this);
    l3.add(b5);

    b6 = new JButton("BALANCE ENQUIRY");
    b6.setForeground(Color.WHITE);
    b6.setBackground(new Color(65,125,128));
    b6.setBounds(700,362,150,35);
    b6.addActionListener(this);
    l3.add(b6);

    b7 = new JButton("EXIT");
    b7.setForeground(Color.WHITE);
    b7.setBackground(new Color(65,125,128));
    b7.setBounds(700,406,150,35);
    b7.addActionListener(this);
    l3.add(b7);

    b8 = new JButton("LOGIN");
    b8.setForeground(Color.WHITE);
    b8.setBackground(new Color(65, 125, 128));
    b8.setBounds(410, 406, 150, 35);
    b8.addActionListener(this);
    l3.add(b8);



    setLayout(null);
    setSize(1550,1080);
    setLocation(0,0);
    setVisible(true);

}

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == b1) {
            new Deposit(pin);
            setVisible(false);
        } else if (e.getSource() == b7) {
            System.exit(0);
        } else if (e.getSource() == b2) {
            new Withdrawl(pin);
            setVisible(false);
        } else if (e.getSource() == b6) {
            new BalanceEnquriy(pin);
            setVisible(false);
        } else if (e.getSource() == b3) {
            new FastCash(pin);
            setVisible(false);
        } else if (e.getSource() == b5) {
            new Pin(pin);
            setVisible(false);
        } else if (e.getSource() == b4) {
            new mini(pin);
        } else if (e.getSource() == b8) {
            new Login(); // Open the login page instead of exiting
            setVisible(false);

        }
    }

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

==

--------------------: PAGE 5 : --------------------------

********************* SIGN UP *********************

```
package bank.management.system;
import com.toedter.calendar.JDateChooser;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

public class Signup extends JFrame implements ActionListener {
    JRadioButton r1,r2,r3,m1,m2,m3;
    JButton next;

    JTextField textName,textFname , textEmail, textMs , textAdd , textcity
,textState,textPin;
    JDateChooser dateChooser;
    Random ran = new Random();
    long first4 =(ran.nextLong() % 9000L) +1000L;
    String first = " " + Math.abs(first4);
    Signup(){
      super ("APPLICATION FORM");

      ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icon/bank.png"));
      Image i2 = i1.getImage().getScaledInstance(100,100,Image.SCALE_DEFAULT);
      ImageIcon i3 = new ImageIcon(i2);
      JLabel image = new JLabel(i3);
      image.setBounds(25,10,100,100);
      add(image);

      JLabel label1 = new JLabel("APPLICATION FORM NO."+ first);
      label1.setBounds(160,20,600,40);
      label1.setFont(new Font("Raleway",Font.BOLD,38));
      add(label1);

      JLabel label2 = new JLabel("Page 1");
      label2.setFont(new Font("Ralway",Font.BOLD, 22));
      label2.setBounds(330,70,600,30);
      add(label2);

      JLabel label3 = new JLabel("Personal Details");
      label3.setFont(new Font("Raleway", Font.BOLD,22));
      label3.setBounds(290,90,600,30);
      add(label3);

      JLabel labelName = new JLabel("Name :");
      labelName.setFont(new Font("Raleway", Font.BOLD, 20));
      labelName.setBounds(100,190,100,30);
      add(labelName);

      textName = new JTextField();
      textName.setFont(new Font("Raleway",Font.BOLD, 14));
      textName.setBounds(300,190,400,30);
      add(textName);

      JLabel labelfName = new JLabel("Father's Name :");
      labelfName.setFont(new Font("Raleway", Font.BOLD, 20));
      labelfName.setBounds(100,240,200,30);
      add(labelfName);

      textFname = new JTextField();
      textFname.setFont(new Font("Raleway",Font.BOLD, 14));
      textFname.setBounds(300,240,400,30);
      add(textFname);

      JLabel DOB = new JLabel("Date of Birth");
      DOB.setFont(new Font("Raleway", Font.BOLD, 20));
      DOB.setBounds(100,340,200,30);
      add(DOB);

      dateChooser = new JDateChooser();
      dateChooser.setForeground(new Color(105,105,105));
      dateChooser.setBounds(300,340,400,30);
      add(dateChooser);

      JLabel labelG = new JLabel("Gender");
      labelG.setFont(new Font("Raleway", Font.BOLD, 20));
      labelG.setBounds(100,290,200,30);
      add(labelG);

      r1 = new JRadioButton("Male");
      r1.setFont(new Font("Raleway", Font.BOLD,14));
      r1.setBackground(new Color(222,255,228));
      r1.setBounds(300,290,60,30);
      add(r1);

      r2 = new JRadioButton("Female");
      r2.setBackground(new Color(222,255,228));
```

```
r2.setFont(new Font("Raleway", Font.BOLD,14));
r2.setBounds(450,290,90,30);
add(r2);

r3 = new JRadioButton("Others");
r3.setBackground(new Color(222,255,228));
r3.setFont(new Font("Raleway", Font.BOLD,14));
r3.setBounds(450,290,90,30);
add(r3);

ButtonGroup buttonGroup = new ButtonGroup();
buttonGroup.add(r1);
buttonGroup.add(r2);
buttonGroup.add(r3);

JLabel labelEmail = new JLabel("Email address :");
labelEmail.setFont(new Font("Raleway", Font.BOLD, 20));
labelEmail.setBounds(100,390,200,30);
add(labelEmail);

textEmail = new JTextField();
textEmail.setFont(new Font("Raleway",Font.BOLD, 14));
textEmail.setBounds(300,390,400,30);
add(textEmail);



JLabel labelMs = new JLabel("Marital Status :");
labelMs.setFont(new Font("Raleway", Font.BOLD, 20));
labelMs.setBounds(100,440,200,30);
add(labelMs);

textMs = new JTextField();
textMs.setFont(new Font("Raleway",Font.BOLD, 14));
textMs.setBounds(300,390,400,30);
add(textMs);

m1 = new JRadioButton("Married");
m1.setBounds(300,440,100,30);
m1.setBackground(new Color(222,255,228));
m1.setFont(new Font("Raleway", Font.BOLD,14));
add(m1);
```

```
m2 = new JRadioButton("Unmarried");
m2.setBackground(new Color(222,255,228));
m2.setBounds(450,440,100,30);
m2.setFont(new Font("Raleway", Font.BOLD,14));
add(m2);

m3 = new JRadioButton("Other");
m3.setBackground(new Color(222,255,228));
m3.setBounds(635,440,100,30);
m3.setFont(new Font("Raleway", Font.BOLD,14));
add(m3);

ButtonGroup buttonGroup1 = new ButtonGroup();
buttonGroup1.add(m1);
buttonGroup1.add(m2);
buttonGroup1.add(m3);

JLabel labelAdd = new JLabel("Address :");
labelAdd.setFont(new Font("Raleway", Font.BOLD, 20));
labelAdd.setBounds(100,490,200,30);
add(labelAdd);

textAdd = new JTextField();
textAdd.setFont(new Font("Raleway",Font.BOLD, 14));
textAdd.setBounds(300,490,400,30);
add(textAdd);

JLabel labelCity = new JLabel("City :");
labelCity.setFont(new Font("Raleway", Font.BOLD, 20));
labelCity.setBounds(100,540,200,30);
add(labelCity);

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

JLabel labelPin = new JLabel("Pin Code :");
labelPin.setFont(new Font("Raleway", Font.BOLD, 20));
labelPin.setBounds(100,590,200,30);
```

```
        add(labelPin);

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

        JLabel labelstate = new JLabel("State :");
        labelstate.setFont(new Font("Raleway", Font.BOLD, 20));
        labelstate.setBounds(100,640,200,30);
        add( labelstate);

        textState = new JTextField();
        textState.setFont(new Font("Raleway",Font.BOLD, 14));
        textState.setBounds(300,640,400,30);
        add(textState);

        next = new JButton("Next");
        next.setFont(new Font("Raleway",Font.BOLD, 14));
        next.setBackground(Color.BLACK);
        next.setForeground(Color.WHITE);
        next.setBounds(620,710,80,30);
        next.addActionListener(this);
        add(next);

        getContentPane().setBackground(new Color(222,255,228));
        setLayout(null);
        setSize(850,800);
        setLocation(360,40);
        setVisible(true);

    }

    @Override
    public void actionPerformed(ActionEvent e) {

        String formno = first;
        String name = textName.getText();
        String fname = textFname.getText();
        String dob = ((JTextField)
dateChooser.getDateEditor().getUiComponent()).getText();
```

```
        String gender = null;
        if(r1.isSelected()){
            gender = "Male";
        }else if (r2.isSelected()){
            gender = "Female";
        }
        String email = textEmail.getText();
        String marital =null;
        if (m1.isSelected()){
            marital = "Married";
        } else if (m2.isSelected()) {
            marital = "Unmarried";
        } else if (m3.isSelected()) {
            marital = "Other";
        }

        String address = textAdd.getText();
        String city = textcity.getText();
        String pincode = textPin.getText();
        String state = textState.getText();

        try{
            if (textName.getText().equals("")){
                JOptionPane.showMessageDialog(null, "Fill all the fields");
            }else {
                Connn c = new Connn();
                String q = "insert into signup values('"+formno+"',
'"+name+"','"+fname+"','"+dob+"','"+gender+"','"+email+"','"+marital+"',
'"+address+"', '"+city+"','"+pincode+"','"+state+"' )";
                c.statement.executeUpdate(q);
                new Signup2(formno);
                setVisible(false);
            }

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

    }

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

===

---------------------------------: PAGE 6 :------------------------

```
******************* BALANCE ENQURIY *****************
package bank.management.system;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
public class BalanceEnquriy extends JFrame implements ActionListener {


    String pin;
    JLabel label2;
    JButton b1;
    BalanceEnquriy(String pin){
        this.pin =pin;

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icon/atm2.png"));
        Image i2 = i1.getImage().getScaledInstance(1550,830,Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l3 = new JLabel(i3);
        l3.setBounds(0,0,1550,830);
        add(l3);

        JLabel label1 = new JLabel("Your Current Balance is Rs ");
        label1.setForeground(Color.WHITE);
        label1.setFont(new Font("System", Font.BOLD, 16));
        label1.setBounds(430,180,700,35);
        l3.add(label1);

        label2 = new JLabel();
        label2.setForeground(Color.WHITE);
        label2.setFont(new Font("System", Font.BOLD, 16));
        label2.setBounds(430,220,400,35);
        l3.add(label2);

        b1 = new JButton("Back");
        b1.setBounds(700,406,150,35);
        b1.setBackground(new Color(65,125,128));
        b1.setForeground(Color.WHITE);
        b1.addActionListener(this);
        l3.add(b1);

        int balance =0;
        try{
            Connn c = new Connn();
            ResultSet resultSet = c.statement.executeQuery("Select * from bank where pin =
'"+pin+"'");
            while (resultSet.next()){
                if (resultSet.getString("type").equals("Deposit")){
                    balance += Integer.parseInt(resultSet.getString("amount"));
                }else {
                    balance -= Integer.parseInt(resultSet.getString("amount"));
                }
            }
        }catch (Exception e){
            e.printStackTrace();
        }

        label2.setText(""+balance);

        setLayout(null);
        setSize(1550,1080);
        setLocation(0,0);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        new main_Class(pin);
    }

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

```
}
```

Chapter-5

TESTING

Creating a robust testing strategy for a banking management system (BMS) requires careful planning, as it involves sensitive financial data, regulatory requirements, and security needs. Here's a breakdown of an effective testing strategy for a BMS:

1. Requirements Analysis

- **Objective:** Ensure that all requirements are understood and covered by test cases.
- **Activities:** Collaborate with stakeholders, business analysts, and development teams to verify that requirements are complete and accurate.

2. Test Planning

- **Scope:** Define what needs to be tested (modules, features, user journeys).
- **Resources:** Assign roles and responsibilities for each team member.
- **Schedule:** Develop a timeline with milestones for testing.
- **Budget:** Allocate resources for testing tools, environments, and personnel.
- **Risk Assessment:** Identify potential risks (e.g., regulatory changes, security vulnerabilities).

3. Test Environment Setup

- **Production-like Environment:** Ensure the test environment resembles the production environment for accurate results.
- **Data Preparation:** Use anonymized or dummy data to mirror real-world scenarios.
- **Security Measures:** Implement access control to prevent unauthorized access to the test environment.

4. Types of Testing

a. Functional Testing

markdown

Copy code

- **Objective:** Verify that each feature functions as expected.
- **Approach:**
    - **Unit Testing:** Test individual components (e.g., login, transaction).
    - **Integration Testing:** Verify data flow between modules (e.g., transferring funds).
    - **System Testing:** Check end-to-end functionality (e.g., opening an account to making a transaction).

b. Non-Functional Testing

markdown

Copy code

- **Performance Testing:** Ensure the system can handle high loads (e.g., peak banking hours).
- **Security Testing:** Validate data protection, encryption, and access control to prevent unauthorized access.
- **Usability Testing:** Assess the user experience for intuitive navigation and ease of use.

c. Compliance Testing

css

Copy code

- Verify adherence to regulatory standards (e.g., KYC, AML) and ensure proper audit logs and data protection measures.

d. Database Testing

markdown

Copy code

- **Data Consistency and Integrity:** Verify that transactions and other data remain consistent.
- **Backup and Recovery Testing:** Ensure data can be recovered in case of a failure.

e. User Acceptance Testing (UAT)

sql

Copy code

- **End-User Scenarios:** Validate that the system meets end-user needs and workflows.

- **Feedback and Sign-off:** Obtain final approval from stakeholders.

5. Regression Testing

- Ensure that new changes haven't introduced bugs in existing functionalities, using automation where possible.

6. Automated Testing

- Automate repetitive tests (e.g., login, fund transfers, report generation).
- Select tools that support banking transactions, security, and compliance requirements.

7. Performance and Load Testing

- **Objective:** Ensure the system can handle peak loads without degradation.
- **Load Testing:** Simulate heavy user loads and analyze performance.
- **Stress Testing:** Push the system beyond its limit to test failure handling and recovery.

8. Security and Penetration Testing

- Identify and address potential vulnerabilities.
- Perform regular penetration testing to simulate hacker attacks and strengthen defenses.

9. Defect Management

- Log and track defects from discovery to resolution.
- Prioritize critical defects for immediate fixes, especially in security and compliance areas.

10. Documentation and Reporting

- Document all test cases, test results, defects, and resolutions.
- Generate regular reports for stakeholders to provide insights into testing progress and areas of concern

REFERENCES AND BIBLOGRAPHY

[1] Hamlin M.R.A and Mayan J.A, (2016), ‘Blood Donation and Life Saver-Blood Donation App’, Int. Conf. on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumara coil, India, 2016, Doi: 10.1109/ICCICCT.2016.798802, pp 625-628. Meiappane A, et al. (2019) ‘DWORLD: Blood Donation App Using Android’,

[2] IEEE Int. Conf. on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, DOI: 10.1109/ICSCAN.2019.8878830, pp 1-5. Pohandulkar S.S and Khandelwal C.S (2018),

Resources and Websites -

https://www.solarwinds.com/database-performance-monitor/integrations/mysql-
https://sqledit.com/dg/
https://www.atlassian.com/software/crucible