Online Banking Management System

“ Developing secure and efficient banking solutions”

DIVESH PANDEY

(23SCSE1180268)

Md. AATIF HASSAN

(23SCSE1180120)

SHIVAM JHA

(23SCSE1180272)

KESHAV JHA

(23SCSE11550)

Project Incharge :

MISS RUCHI SHARMA

INTRODUCTIONS

Overview :
Briefly introduce the Banking Management System and Java.

Banking Management System:
Software to manage banking operations such as customer accounts, transactions, loans, etc.

Why Java:
Java is a widely used, secure, and platform-independent language, making it a strong choice for banking systems.

Objectives of a Banking Management System-

- Automating and managing daily banking operations.
- Enhancing security in handling financial transactions.
- Providing customer-friendly interfaces.
- Ensuring data integrity and consistency.

STRUCTURAL FORMATION OF THE PROJECT

WHY USE JAVA FOR BANKING SYSTEMS

Platform Independence: "Write Once, Run Anywhere" capability.
Robust Security Features: Built-in security APIs, encryption.
Multi-threading: Supports multiple processes at the same time, crucial for transaction handling.-
Rich APIs: For networking, file handling, and database connectivity.
Scalability: Can handle large-scale systems with ease.

SYSTEMARCHITECTURE (JAVA-BASED)

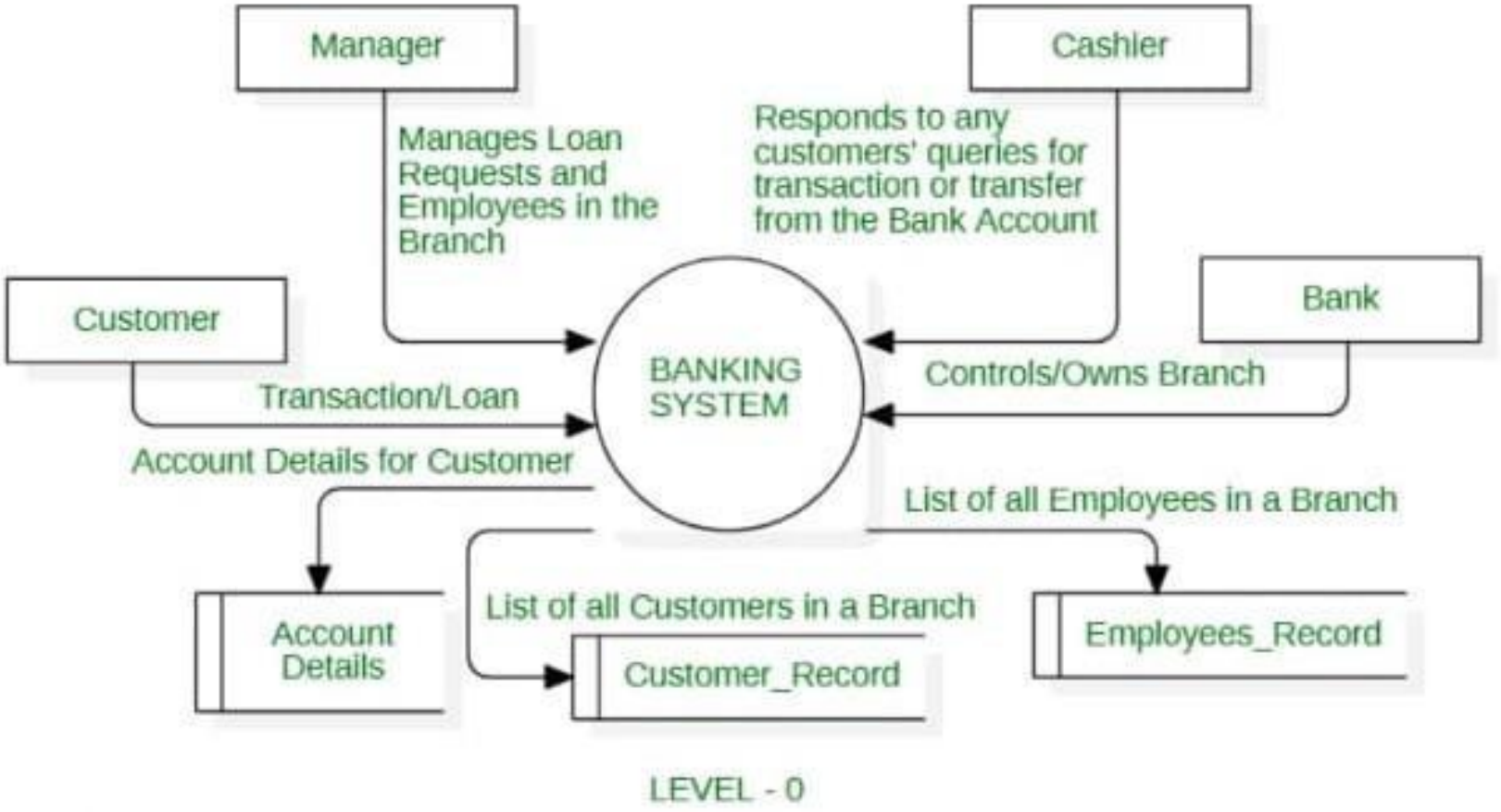
Client Layer : GUI applications (Java Swing, JavaFX) or Web interface (JSP/Servlets).
Business Logic Layer : Core functionalities implemented in Java classes.-
Database Layer : Java Database Connectivity (JDBC) used to connect to a backend database (MySQL, Oracle, etc.).

PROBLEM STATEMENT

Once a time, people have to spend three to four hours to go for bank transaction sometimes cost of transaction was more than that of money deposited or withdrawn. E-banking allows customer to conduct financial transaction on a secure website. Nowadays User Friendly Technology is becoming more popular among customers, most of the banks are providing e-banking facility. Today, most of the customers are increasingly using the technological banking facilities available in banking sector. It reduces cost and saves time. From the customers perceptive towards technological banking provides a convenient and effective way to manage finance that is easily accessible at 24hours a day in 7 days a week. On the other hand, online banking has certain problems such as lack of knowledge to operate the technology, set-up cost, legal issues, lack of relationship among banker and customer, securely and privacy issues. For some people the User Friendly Technology really simplifies their life style, while for others it is very much threatening and complex. Therefore in this context, it is necessary to study the perception of customers' challenges towards User Friendly Technology.

GALGOTIAS UNIVERSITY

FLOWCHART



LEVEL - 0

GALGOTIAS UNIVERSITY

SOURCE CODE {ONLINE BANKING MANAGEMENT SYSTEM}

```java
import java.util.Scanner;

class BankDetails {
    private String accno;
    private String name;
    private String acc_type;
    private long balance;
    Scanner sc = new Scanner(System.in);

    // Method to open a new account
    public void openAccount() {
        System.out.print("Enter Account No: ");
        accno = sc.next();
        System.out.print("Enter Account type: ");
        acc_type = sc.next();
        System.out.print("Enter Name: ");
        name = sc.next();
        System.out.print("Enter Balance: ");
        balance = sc.nextLong();
    }

    // Method to display account details
    public void showAccount() {
```

```java
    // Method to display account details
    public void showAccount() {
        System.out.println("Name of account holder: " + name);
        System.out.println("Account no.: " + accno);
        System.out.println("Account type: " + acc_type);
        System.out.println("Balance: " + balance);
    }

    // Method to deposit money
    public void deposit() {
        System.out.print("Enter the amount you want to deposit: ");
        long amt = sc.nextLong();
        if (amt > 0) {
            balance += amt;
            System.out.println("Amount Deposited: " + amt);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }
```

GALGOTIAS UNIVERSITY

```
    // Method to withdraw money
    public void withdrawal() {
        System.out.print("Enter the amount you want to withdraw: ");
        long amt = sc.nextLong();
        if (balance >= amt) {
            balance -= amt;
            System.out.println("Withdrawal successful. Remaining balance: " + balance);
        } else {
            System.out.println("Insufficient balance. Transaction failed!");
        }
    }

    // Method to search for an account number
    public boolean search(String ac_no) {
        if (accno.equals(ac_no)) {
            showAccount();
            return true;
        }
        return false;
    }
}
```

```
public class BankingApp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Create initial accounts
        System.out.print("How many customers do you want to input? ");
        int n = sc.nextInt();
        BankDetails[] accounts = new BankDetails[n];

        for (int i = 0; i < accounts.length; i++) {
            accounts[i] = new BankDetails();
            accounts[i].openAccount();
        }

        // Menu-driven application
        int choice;
        do {
            System.out.println("\n** Banking System Application **");
            System.out.println("1. Display all account details");
            System.out.println("2. Search by Account number");
            System.out.println("3. Deposit the amount");
            System.out.println("4. Withdraw the amount");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();
```

GALGOTIAS UNIVERSITY

```
switch (choice) {
    case 1:
        for (BankDetails account : accounts) {
            account.showAccount();
        }
        break;
    case 2:
        System.out.print("Enter account no. you want to search: ");
        String ac_no = sc.next();
        boolean found = false;
        for (BankDetails account : accounts) {
            found = account.search(ac_no);
            if (found) break;
        }
        if (!found) {
            System.out.println("Search failed! Account doesn't exist.");
        }
        break;
```

```
            case 3:
                System.out.print("Enter Account no.: ");
                ac_no = sc.next();
                found = false;
                for (BankDetails account : accounts) {
                    found = account.search(ac_no);
                    if (found) {
                        account.deposit();
                        break;
                    }
                }
                if (!found) {
                    System.out.println("Search failed! Account doesn't exist.");
                }
                break;
            case 4:
                System.out.print("Enter Account No: ");
                ac_no = sc.next();
                found = false;
                for (BankDetails account : accounts) {
                    found = account.search(ac_no);
                    if (found) {
                        account.with↓val();
                        break;
```

GALGOTIAS UNIVERSITY

```
                    }
                }
                if (!found) {
                    System.out.println("Search failed! Account doesn't exist.");
                }
                break;
            case 5:
                System.out.println("Thank you for using our banking system. See you so
                break;
            default:
                System.out.println("Invalid choice! Please try again.");
        }
    } while (choice != 5);
    sc.close();
  }
}
```

OUTPUT

Main.java

Share Run

```
126         for (BankDetails account : accounts) {
127             found = account.search(ac_no);
128             if (found) {
129                 account.withdrawal();
130                 break;
131             }
132         }
133         if (!found) {
134             System.out.println("Search failed! Account doesn't
                    exist.");
135         }
136         break;
137     case 5:
138         System.out.println("Thank you for using our banking system
```

Output

```
java -cp /tmp/kCLRUQArhF/BankingApp
How many customers do you want to input?  55
Enter Account No: 6521452133
Enter Account type: Savings
Enter Name: Abhinav Shukla
```

TECHNOLOGY STACK

Frontend:

JavaFX or Java Swing for desktop applications; JSP/Servlets for web-based systems.

Backend:

Core Java for business logic, using Object-Oriented Programming principles.

Database:

JDBC for database interaction (MySQL, Oracle, etc.).

Security:

Java Security APIs for encryption, SSL for secure data transmission.

Server:

Apache Tomcat (if web-based) or standalone Java applications.