BANKING MANEGEMENT SYSTEM

A PROJECT REPORT

Submitted by

HARIOM KUMAR 220101120036

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

DEPARTMENT OF CSE

CENTURION UNIVERSITY OF TECHNOLOGY

&MANAGEMENT: ODISHA

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report "***Banking management system"*** is the Bonafide work of " **HARIOM KUMAR,**" who carried out theproject work under my supervision. This is to further certify to the best of my knowledge, that this project has not been carried out earlier in this institute and the university.

SIGNATURE
(PROF. ARYA LOPPA)
SUPERVISOR
(Professor)

Certified that the above mentioned project has been duly carried out as per the norms of the college and statutes of the university

SIGNATURE
HEAD OF THE DEPARTMENT
(Asst. Professor)

DEPARTMENT SEAL

ACKNOWLEDGEMENTS

I wish to express my profound and sincere gratitude to Prof.Arya loppa, Department of Computer Science Engineering, Paralakhemundi, who guided me into the intricacies of this project non-chalantly with matchless magnanimity.

I thank Prof. DEBENDRA MAHARANA , Head of the Dept. of Computer Science Engineering, and Prof. A.R DASH , DEAN, SOET for extending their support during Course of this investigation.

I would be failing in my duty if I don't acknowledge the co-operation rendered during various stages of image interpretation by PROF. ARYA LOPPA.

I am highly grateful to PROF. ARYA LOPPA who evinced keen interest and invaluable support in the progress and successful completion of my project work.

I am indebted to Prof. A.R DASH for their constant encouragement, co-operation and help. Words of gratitude are not enough to describe the accommodation and fortitude which they have shown throughout my endeavor.

By,

HARIOM KUMAR 220101120036

TABLE OF CONTENTS

ALL THE BASIC WORK RELATED BANKING SYSYTEM

- **CREATING BANK ACCOUNT**
- **DEPOSITE MONEY**
- **WITHDRAWAL MONEY**
- **TRANSFER THE MONEY**
- **CANCEL THE ACOOUNT FROM THE BANK**

INTRODUCTION

The **Simple Banking System in Java** is a console-based application designed to showcase fundamental concepts of object-oriented programming and user interaction in Java. The project simulates basic banking operations, providing users with functionalities such as account creation, deposit, withdrawal, balance check, fund transfer, and account cancellation.

Objectives

The main objectives of this project are as follows:
implement a basic banking system using Java.
Demonstrate key principles of object-oriented programming.
Enable users to perform standard banking operations through a simple console interface.

Methodology

The development of the Simple Banking System in Java followed a structured methodology, encompassing the design, implementation, and testing phases. The project methodology can be outlined as follows:

1. Requirement Analysis

The initial phase involved understanding the project requirements and defining the scope. Key functionalities were identified, including account creation, fund transactions, balance inquiries, and account cancellation. User interaction through a console-based menu was chosen for simplicity and clarity.

2. Object-Oriented Design

The system was designed using object-oriented principles to model real-world entities. Three main classes were conceptualized:

Account Class: Manages individual bank accounts and transactions.
Customer Class: Represents a bank customer and manages associated accounts.
Banking System Class: Orchestrates user interaction and system flow.

CODE

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

class BankAccount {
    private String name;
    private String accountNumber;
    private double balance;

    public BankAccount(String name, String accountNumber, double balance) {
        this.name = name;
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    public String getAccountNumber() {
        return accountNumber;
    }

    public double getBalance() {
        return balance;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited ₹" + amount + ". New balance: ₹" + balance);
    }
     public void withdraw(double amount) {
            if (amount <= balance) {
                balance -= amount;
                System.out.println("Withdrawn ₹" + amount + ". New balance: ₹" +
balance);
            } else {
                System.out.println("Insufficient funds. Withdrawal failed.");
            }
        }
public void displayBalance() {
        System.out.println("Account Balance ₹ " + balance);
    }
```

```
    public void generateStatement() {
        System.out.println("Account Statement for Account Number: " +
accountNumber);
        System.out.println("Customer Name: " + name);
        System.out.println("Current Balance: ₹ " + balance);
    }
}

public class BankingManagementSystem {
    private static Map<String, BankAccount> accounts = new HashMap<>();
    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        int choice;

        do {
            System.out.println("\n1. Create Account");
            System.out.println("2. Withdraw");
            System.out.println("3. Deposit");
            System.out.println("4. Balance Enquiry");
            System.out.println("5. Statement");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character

            switch (choice) {
                case 1:
                    createAccount();
                    break;
case 2:
                    performWithdrawal();
                    break;
                case 3:
                    performDeposit();
                    break;
                case 4:
                    checkBalance();
                    break;
                case 5:
                    generateStatement();
                    break;
                case 6:
                    System.out.println("Exiting the system. Goodbye!");
                    break;
```

```
                default:
                    System.out.println("Invalid choice. Please try again.");
            }

        } while (choice != 6);
    }

    private static void createAccount() {
        System.out.print("Enter customer name: ");
        String name = scanner.nextLine();
        System.out.print("Enter initial balance: ");
        double initialBalance = scanner.nextDouble();
        scanner.nextLine(); // Consume the newline character

        String accountNumber = generateAccountNumber();
        BankAccount account = new BankAccount(name, accountNumber,
initialBalance);
        accounts.put(accountNumber, account);

        System.out.println("Account created successfully. Account Number: " +
accountNumber);
    }

    private static void performWithdrawal() {
        System.out.print("Enter account number: ");
        String accountNumber = scanner.nextLine();

        if (accounts.containsKey(accountNumber)) {
            BankAccount account = accounts.get(accountNumber);
            System.out.print("Enter withdrawal amount: ");
            double amount = scanner.nextDouble();
            scanner.nextLine(); // Consume the newline character

            account.withdraw(amount);
        } else {
            System.out.println("Account not found.");
        }
    }

    private static void performDeposit() {
        System.out.print("Enter account number: ");
        String accountNumber = scanner.nextLine();

        if (accounts.containsKey(accountNumber)) {
            BankAccount account = accounts.get(accountNumber);
            System.out.print("Enter deposit amount: ");
            double amount = scanner.nextDouble();
            scanner.nextLine(); // Consume the newline character

            account.deposit(amount);
        } else {
            System.out.println("Account not found.");
        }
    }

    private static void checkBalance() {
        System.out.print("Enter account number: ");
```

OUTPUT

Of the various activities which we are going to present here

What are the options available here in this system

```
PS C:\Users\AJAY> cd "d:\JAVA  CODE\" ; if ($?) { javac Project.java } ;
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option:
```

1. Enter the user name

```
Select an option: 1
Enter user name: Sundram Kumari
User Sundram Kumari created.
```

```
PS C:\Users\AJAY> cd "d:\JAVA  CODE\" ; if ($?) { javac Project.java } ; if ($?) { java Project }
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 1
Enter user name: Sundram Kumari
User Sundram Kumari created.
```

```
PS C:\Users\AJAY> cd "d:\JAVA  CODE\" ; if ($?) { javac Project.java }
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 1
Enter user name: Ajay Kumar Jaiswal
User Ajay Kumar Jaiswal created.
```

2. Deposite money

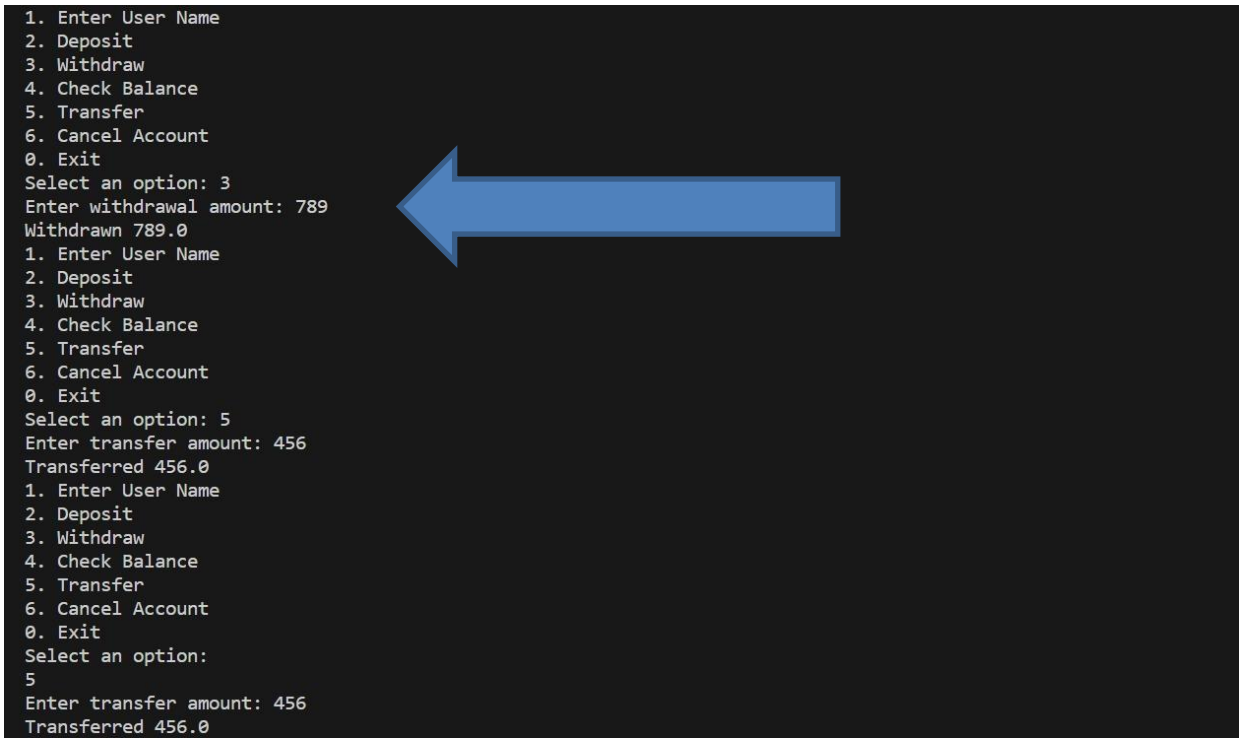
This is the option which comes inside the seen Code to deposite the money in the created account

```
PS C:\Users\AJAY> cd "d:\JAVA  CODE\" ; if ($?) { javac Project.java } ; if ($?) { java Project }
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 1
Enter user name: Ajay Kumar Jaiswal
User Ajay Kumar Jaiswal created.
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 2
Enter deposit amount: 5000
Deposited 5000.0
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 4
```

We have pinned the snap of the deposite of account holder Hariom kumar

```
Enter user name: Sundram Kumari
User Sundram Kumari created.
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 2
Enter deposit amount: 10000
Deposited 10000.0
```

3. Withdrawal output



4. Check balance

5. Transfer the money

```
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 3
Enter withdrawal amount: 789
Withdrawn 789.0
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 5
Enter transfer amount: 456
Transferred 456.0
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option:
5
Enter transfer amount: 456
Transferred 456.0
```

NIKHIL AND ASHWIN REPORT IS HERE

```
PS C:\Users\AJAY> cd "d:\JAVA  CODE\" ; if ($?) { javac Project.java } ; if ($?) { java Project }
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 1
Enter user name: Nikhil Kumar
User Nikhil Kumar created.
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 1
Enter user name: Ashwani Kumar Sharma
User Ashwani Kumar Sharma created.
1. Enter User Name
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer
6. Cancel Account
0. Exit
Select an option: 2
Enter deposit amount: 5689
```

CONCLUSION

In conclusion, the development of the Simple Banking System in Java has proven to be a valuable exercise in applying object- oriented programming principles to real-world scenarios. This project aimed to provide users with a straightforward banking experience through a console interface, and its success lies in achieving fundamental banking operations.

Key Achievements

Object-Oriented Design:The project successfully demonstrated the effective use of object-oriented principles with the creation of the and classes. This design promotes code modularity and scalability.

User Interaction: The console-based user interface offers a clear and intuitive experience for users to perform basic banking operations. The menu-driven approach ensures ease of navigation.

Functionality: Core banking functionalities, including account creation, deposit, withdrawal, balance check, fund transfer, and account cancellation, were implemented and tested successfully