**Introduction**

The Bank Management System is a software solution designed to automate and manage various banking operations such as account creation, deposits, withdrawals, transfers, and balance inquiries. The purpose of this project is to simulate the functionality of a real-world banking system in a simple and user-friendly interface using Java. The system allows users to perform banking transactions securely, manage customer accounts, and maintain transaction records, all while adhering to basic banking principles.

In real-life banking systems, customers need a way to access their accounts, check balances, transfer money, and perform other financial transactions efficiently and securely. The Bank Management System project seeks to replicate these processes in a controlled, software-driven environment, providing hands-on experience with core programming concepts like object-oriented programming (OOP), data management, and transaction handling.

**UML Diagram**

BankSystem

- accounts: Map<Integer,

BankAccount>

- scanner: Scanner

+ createAccount(): void

+ depositMoney(): void |

+ withdrawMoney(): void

+ checkBalance(): void

+ accountDetails(): void

BankAccount

accountHolderName: String

- accountNumber: int

- balance: double

+ BankAccount(name: String,

+ main(String[]): void

accountNumber: int, initialDeposit: double)

+ deposit(amount: double): void + withdraw(amount: double): + getBalance(): double

+ displayAccountInfo(): void

| |

Start Program (Main Menu)

Display Menu of Options

1. Create Account,

2. Deposit Money,

3. Withdraw Money,

4.Check Balance,

5. Account Details,

6. Exit) |

User Input for Option

1.Option Create Account

2 Option Deposit Money

Enter Account Number and

Deposit Amount

Enter User

Name and Initial

Deposit

Deposit Amount to Account

Generate Random

Account Number

Display Deposit Confirmation

Create Account and

Store in Accounts Map

Return to Main Menu

Return to Main Menu

Option 3

Withdraw

Money

Enter Account

Number and

Withdrawal

| Amount |

Withdraw Amount from Account

Display Withdraw Confirmation

End Program

**code**

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class BankSystem {

private static Map<Integer, BankAccount> accounts = new HashMap<>();

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

while (true) {

System.out.println("Welcome to the Bank Management System");

System.out.println("1. Create Account");

System.out.println("2. Deposit Money");

System.out.println("3. Withdraw Money");

System.out.println("4. Check Balance");

System.out.println("5. Account Details");

System.out.println("6. Exit");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

createAccount();

break;

case 2:

depositMoney();

break;

case 3:

withdrawMoney();

break;

case 4:

checkBalance();

break;

case 5:

accountDetails();

break;

case 6:

System.out.println("Thank you for using the Bank Management System!");

System.exit(0);

break;

default:

System.out.println("Invalid option, please try again.");

}

}

}

private static void createAccount() {

System.out.print("Enter account holder name: ");

scanner.nextLine();

String name = scanner.nextLine();

System.out.print("Enter initial deposit: ");

double initialDeposit = scanner.nextDouble();

int accountNumber = (int) (Math.random() \* 1000000);

BankAccount account = new BankAccount(name, accountNumber, initialDeposit);

accounts.put(accountNumber, account);

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

}

private static void depositMoney() {

System.out.print("Enter account number: ");

int accountNumber = scanner.nextInt();

if (accounts.containsKey(accountNumber)) {

System.out.print("Enter deposit amount: ");

double amount = scanner.nextDouble();

BankAccount account = accounts.get(accountNumber);

account.deposit(amount);

} else {

System.out.println("Account not found.");

}

}

private static void withdrawMoney() {

System.out.print("Enter account number: ");

int accountNumber = scanner.nextInt();

if (accounts.containsKey(accountNumber)) {

System.out.print("Enter withdrawal amount: ");

double amount = scanner.nextDouble();

BankAccount account = accounts.get(accountNumber);

account.withdraw(amount);

} else {

System.out.println("Account not found.");

}

}

private static void checkBalance() {

System.out.print("Enter account number: ");

int accountNumber = scanner.nextInt();

if (accounts.containsKey(accountNumber)) {

BankAccount account = accounts.get(accountNumber);

System.out.println("Current balance: " + account.getBalance());

} else {

System.out.println("Account not found.");

}

}

private static void accountDetails() {

System.out.print("Enter account number: ");

int accountNumber = scanner.nextInt();

if (accounts.containsKey(accountNumber)) {

BankAccount account = accounts.get(accountNumber);

account.displayAccountInfo();

} else {

System.out.println("Account not found.");

}

}

}

import java.util.Scanner;

public class BankAccount {

private String accountHolderName;

private int accountNumber;

private double balance;

public BankAccount(String accountHolderName, int accountNumber, double initialDeposit) {

this.accountHolderName = accountHolderName;

this.accountNumber = accountNumber;

this.balance = initialDeposit;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Deposited: " + amount);

} else {

System.out.println("Invalid deposit amount.");

}

}

public void withdraw(double amount) {

if (amount > 0 && amount <= balance) {

balance -= amount;

System.out.println("Withdrawn: " + amount);

} else {

System.out.println("Invalid withdrawal amount or insufficient funds.");

}

}

public double getBalance() {

return balance;

}

public void displayAccountInfo() {

System.out.println("Account Holder: " + accountHolderName);

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

System.out.println("Balance: " + balance);

}

public String getAccountHolderName() {

return accountHolderName;

}

public int getAccountNumber() {

return accountNumber;

}

public void setBalance(double balance) {

this.balance = balance;

}

}

**output**

Welcome to the Bank Management System

1. Create Account

2. Deposit Money

3. Withdraw Money

4. Check Balance

5. Account Details

6. Exit

Choose an option: 1

Enter account holder name: John Doe

Enter initial deposit: 1000

Account created successfully. Account Number: 123456

Welcome to the Bank Management System

1. Create Account

2. Deposit Money

3. Withdraw Money

4. Check Balance

5. Account Details

6. Exit

Choose an option: 2

Enter account number: 123456

Enter deposit amount: 500

Deposited: 500.0

Welcome to the Bank Management System

1. Create Account

2. Deposit Money

3. Withdraw Money

4. Check Balance

5. Account Details

6. Exit

Choose an option: 4

Enter account number: 123456

Current balance: 1500.0

**Conclusion:**

The Bank Management System project is a practical and essential tool for learning Java programming and object-oriented design. It simulates basic banking operations, providing users with secure access to their accounts, performing transactions, and viewing account information. The project can be expanded further by adding more advanced features such as loan management, reporting, and real-time database integration.