Banking Application (Java Project)

Name: Rahul Yadav

Roll No: 2401410021

Course: B.Tech Cybersecurity (2024-2028)

Java Code:

import java.util.Scanner;

//Account Class class Account { private long accountNumber; //long for large numbers private String accountHolderName; private double balance; private String email; private String phoneNumber;

//Constructor

public Account(long accountNumber, String accountHolderName, double balance, String email, String phoneNumber) { this.accountNumber = accountNumber; this.accountHolderName = accountHolderName; this.balance = balance; this.email = email;

this.phoneNumber = phoneNumber;

}

//Deposit method public void deposit(double amount) { if (amount > 0) { balance += amount;

System.out.println("₹" + amount + " deposited successfully.");

} else {

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

}

}

//Withdraw method public void withdraw(double amount) { if (amount > 0 && balance >= amount) { balance -= amount;

System.out.println("₹" + amount + " withdrawn successfully.");

} else if (amount <= 0) {

System.out.println("Invalid withdrawal amount!");

} else {

System.out.println("Insufficient balance!");

}

}

//Display account details public void displayAccountDetails() {

System.out.println("\n--- Account Details ---");

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

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

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

System.out.println("Email: " + email);

System.out.println("Phone: " + phoneNumber);

System.out.println("------------------------");

}

//Update contact details public void updateContactDetails(String email, String phoneNumber) { this.email = email; this.phoneNumber = phoneNumber;

System.out.println("Contact details updated successfully!");

}

//Getter for account number public long getAccountNumber() { return accountNumber;

}

}

//User Interface Class public class BankingApplication { private static Account[] accounts = new Account[100]; //Array of accounts private static int accountCount = 0;

private static long accountNumberGenerator = 7651110987611L; //Start with large

number private static Scanner sc = new Scanner(System.in);

//Preload some accounts (optional) public static void preloadAccounts() {

accounts[accountCount++] = new Account(++accountNumberGenerator, "Mohit",

10000, "mohit@gmail.com", "8877665544");

accounts[accountCount++] = new Account(++accountNumberGenerator, "John

Doe", 5000, "john@example.com", "9988776655");

accounts[accountCount++] = new Account(++accountNumberGenerator, "Alice",

7500, "alice@example.com", "8899776655");

}

//Create new account public static void createAccount() {

System.out.print("Enter account holder name: "); String name = sc.nextLine();

System.out.print("Enter initial deposit amount: "); double balance = sc.nextDouble(); sc.nextLine(); //consume newline

System.out.print("Enter email address: ");

String email = sc.nextLine();

System.out.print("Enter phone number: "); String phone = sc.nextLine();

long accountNumber = ++accountNumberGenerator;

accounts[accountCount++] = new Account(accountNumber, name, balance, email,

phone);

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

}

//Find account by account number public static Account findAccount(long accNum) { for (int i = 0; i < accountCount; i++) { if (accounts[i].getAccountNumber() == accNum) { return accounts[i];

}

} return null;

}

//Deposit public static void performDeposit() { System.out.print("Enter account number: "); long accNum = sc.nextLong(); System.out.print("Enter amount to deposit: "); double amount = sc.nextDouble(); sc.nextLine(); //consume newline Account acc = findAccount(accNum); if (acc != null) { acc.deposit(amount);

} else {

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

}

}

//Withdraw public static void performWithdrawal() { System.out.print("Enter account number: "); long accNum = sc.nextLong(); System.out.print("Enter amount to withdraw: "); double amount = sc.nextDouble(); sc.nextLine();

Account acc = findAccount(accNum); if (acc != null) { acc.withdraw(amount);

} else {

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

}

}

//Show account details public static void showAccountDetails() { System.out.print("Enter account number: "); long accNum = sc.nextLong();

sc.nextLine();

Account acc = findAccount(accNum); if (acc != null) { acc.displayAccountDetails();

} else {

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

}

}

//Update contact details public static void updateContact() { System.out.print("Enter account number: "); long accNum = sc.nextLong();

sc.nextLine();

Account acc = findAccount(accNum); if (acc != null) {

System.out.print("Enter new email: ");

String email = sc.nextLine();

System.out.print("Enter new phone number: "); String phone = sc.nextLine(); acc.updateContactDetails(email, phone); } else {

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

}

}

//Main Menu public static void mainMenu() { while (true) {

System.out.println("\n--- Banking Application ---");

System.out.println("1. Create a new account");

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

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

System.out.println("4. View account details");

System.out.println("5. Update contact details");

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

System.out.print("Enter your choice: ");

int choice = sc.nextInt(); sc.nextLine(); //consume newline

switch (choice) { case 1: createAccount(); break; case 2: performDeposit(); break; case 3: performWithdrawal(); break; case 4: showAccountDetails(); break; case 5: updateContact(); break;

case 6: System.out.println("Thank you for using the Banking Application!"); return; default: System.out.println("Invalid choice! Please try again.");

}

}

}

//Main method public static void main(String[] args) { preloadAccounts(); //Optional: preload some accounts mainMenu();

}