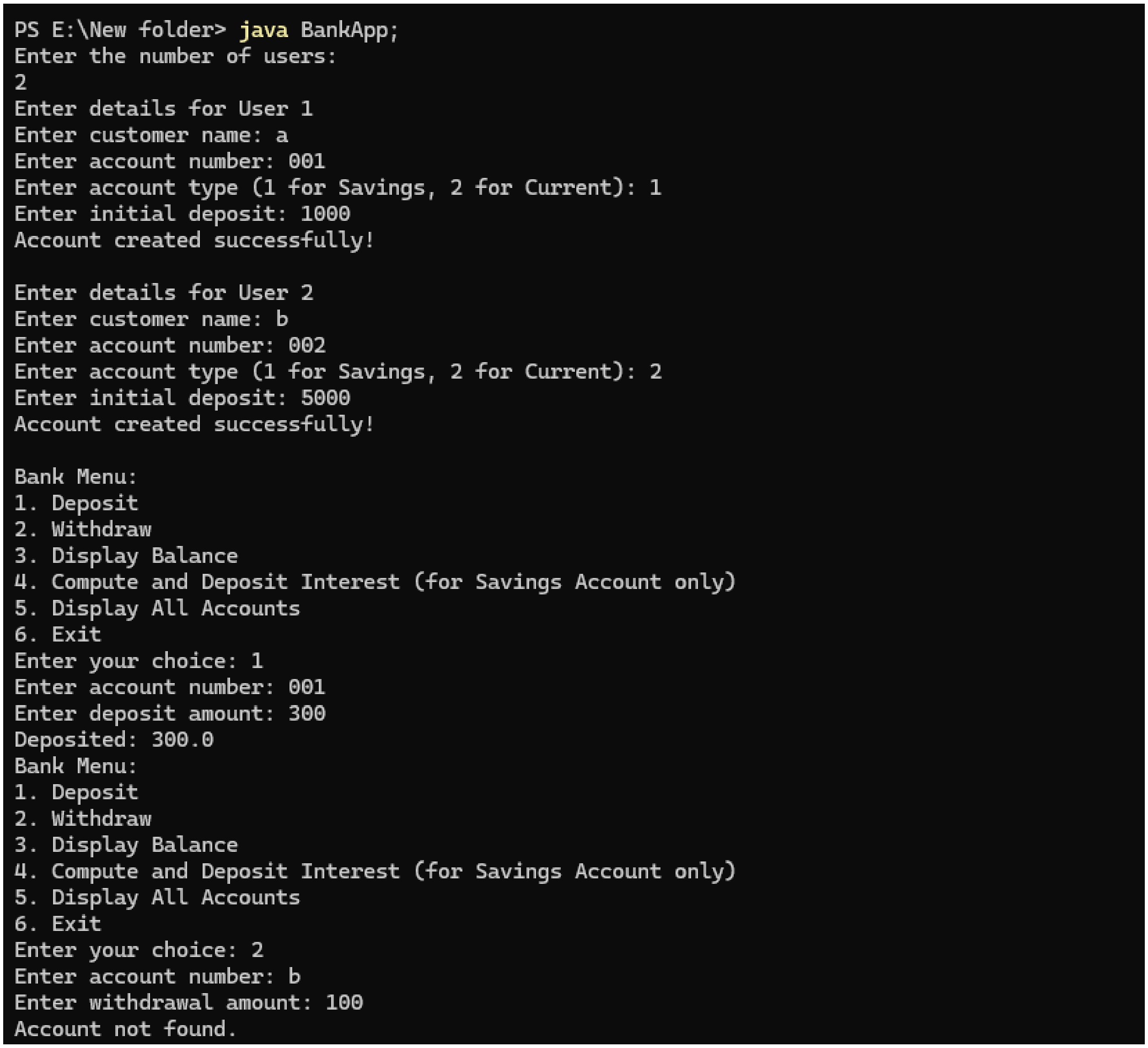
```
import java.util.*;


abstract class Account {
    protected String customerName;
    protected String accountNumber;
    protected double balance;

    public Account(String customerName, String accountNumber, double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }


    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }


    public void displayBalance() {
        System.out.println("Account Balance: " + balance);
    }


    public abstract void withdraw(double amount);
}

class CurrentAccount extends Account {
    private static final double MIN_BALANCE = 1000;
    private static final double SERVICE_CHARGE = 50;

    public CurrentAccount(String customerName, String accountNumber, double
initialBalance) {
        super(customerName, accountNumber, initialBalance);
    }
```

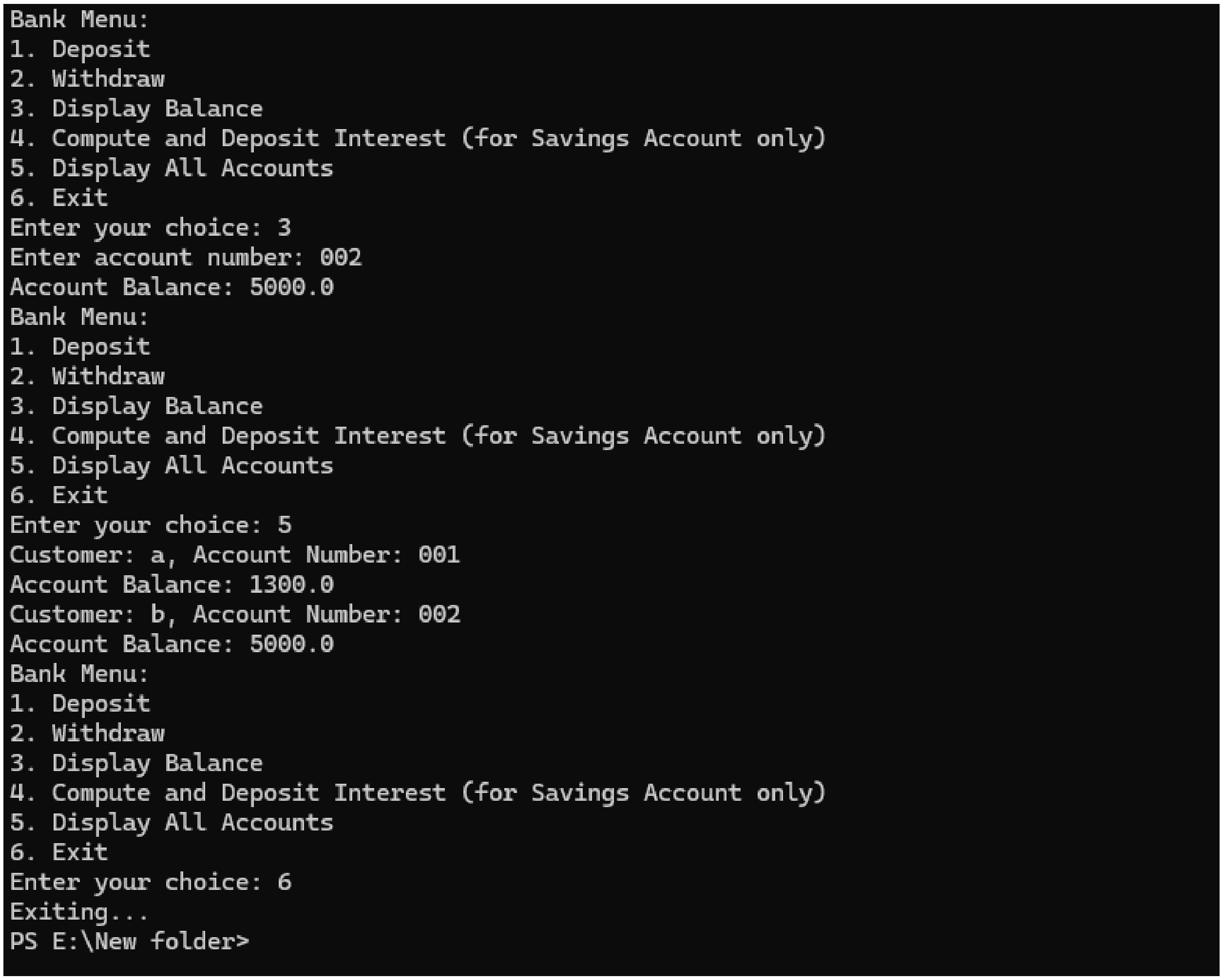
```
      public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
            if (balance < MIN_BALANCE) {
                balance -= SERVICE_CHARGE; // Impose service charge
                System.out.println("Balance below minimum. Service charge of " + SERVICE_CHARGE + " imposed.");
            }
        } else {
            System.out.println("Insufficient funds!");
        }
    }
}

class SavingsAccount extends Account {
    private static final double INTEREST_RATE = 0.05;

    public SavingsAccount(String customerName, String accountNumber, double initialBalance) {
        super(customerName, accountNumber, initialBalance);
    }



    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient funds!");
        }
    }


    public void computeAndDepositInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest of " + interest + " deposited.");
    }
}

class Bank {
    private List<Account> accounts = new ArrayList<>();


    public void addAccount(Account account) {
        accounts.add(account);
    }


    public void displayAllAccounts() {
        for (Account account : accounts) {
            System.out.println("Customer: " + account.customerName + ", Account Number: " + account.accountNumber);
            account.displayBalance();
        }
    }


    public Account getAccount(String accountNumber) {
        for (Account account : accounts) {
            if (account.accountNumber.equals(accountNumber)) {
                return account;
            }
        }
        return null;
    }
}

public class BankApp {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Bank bank = new Bank();
        int n;

          System.out.println("Enter the number of users:");
        n = scanner.nextInt();
        scanner.nextLine();
```

```
           for (int i = 0; i < n; i++) {
            System.out.println("Enter details for User " + (i + 1));

            System.out.print("Enter customer name: ");
            String customerName = scanner.nextLine();

            System.out.print("Enter account number: ");
            String accountNumber = scanner.nextLine();

            System.out.print("Enter account type (1 for Savings, 2 for Current): ");
            int accountType = scanner.nextInt();
            scanner.nextLine(); // Consume newline character

            System.out.print("Enter initial deposit: ");
            double initialDeposit = scanner.nextDouble();
            scanner.nextLine(); // Consume newline character

            Account account;
            if (accountType == 1) {
                account = new SavingsAccount(customerName, accountNumber,
initialDeposit);
            } else {
                account = new CurrentAccount(customerName, accountNumber,
initialDeposit);
            }

            bank.addAccount(account);
            System.out.println("Account created successfully!\n");
        }


        boolean exit = false;
        while (!exit) {
            System.out.println("Bank Menu:");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            System.out.println("4. Compute and Deposit Interest (for Savings Account
only)");
            System.out.println("5. Display All Accounts");
            System.out.println("6. Exit");

            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline character

            switch (choice) {
                case 1:
                    System.out.print("Enter account number: ");
                    String depositAccount = scanner.nextLine();
                    System.out.print("Enter deposit amount: ");
                    double depositAmount = scanner.nextDouble();
                    scanner.nextLine(); // Consume newline character
                    Account accountToDeposit = bank.getAccount(depositAccount);
                    if (accountToDeposit != null) {
                        accountToDeposit.deposit(depositAmount);
                    } else {
                        System.out.println("Account not found.");
                    }
                    break;

                case 2:
                    System.out.print("Enter account number: ");
                    String withdrawAccount = scanner.nextLine();
                    System.out.print("Enter withdrawal amount: ");
                    double withdrawAmount = scanner.nextDouble();
                    scanner.nextLine(); // Consume newline character
                    Account accountToWithdraw = bank.getAccount(withdrawAccount);
                    if (accountToWithdraw != null) {
                        accountToWithdraw.withdraw(withdrawAmount);
                    } else {
                        System.out.println("Account not found.");
                    }
                    break;

                case 3:
                    System.out.print("Enter account number: ");
                    String displayAccount = scanner.nextLine();
                    Account accountToDisplay = bank.getAccount(displayAccount);
                    if (accountToDisplay != null) {
                        accountToDisplay.displayBalance();
```

```
                    } else {
                        System.out.println("Account not found.");
                    }
                    break;

                case 4:
                    System.out.print("Enter account number: ");
                    String interestAccount = scanner.nextLine();
                    Account accountForInterest = bank.getAccount(interestAccount);
                    if (accountForInterest != null && accountForInterest instanceof
SavingsAccount) {
                        ((SavingsAccount) accountForInterest).computeAndDepositInterest();
                    } else {
                        System.out.println("Invalid account or not a Savings Account.");
                    }
                    break;

                case 5:
                    bank.displayAllAccounts();
                    break;

                case 6:
                    exit = true;
                    System.out.println("Exiting...");
                    break;

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



    }
}
```



```
PS E:\New folder> java BankApp;
Enter the number of users:
2
Enter details for User 1
Enter customer name: a
Enter account number: 001
Enter account type (1 for Savings, 2 for Current): 1
Enter initial deposit: 1000
Account created successfully!

Enter details for User 2
Enter customer name: b
Enter account number: 002
Enter account type (1 for Savings, 2 for Current): 2
Enter initial deposit: 5000
Account created successfully!

Bank Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (for Savings Account only)
5. Display All Accounts
6. Exit
Enter your choice: 1
Enter account number: 001
Enter deposit amount: 300
Deposited: 300.0
Bank Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (for Savings Account only)
5. Display All Accounts
6. Exit
Enter your choice: 2
Enter account number: b
Enter withdrawal amount: 100
Account not found.
```



```
Bank Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (for Savings Account only)
5. Display All Accounts
6. Exit
Enter your choice: 3
Enter account number: 002
Account Balance: 5000.0
Bank Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (for Savings Account only)
5. Display All Accounts
6. Exit
Enter your choice: 5
Customer: a, Account Number: 001
Account Balance: 1300.0
Customer: b, Account Number: 002
Account Balance: 5000.0
Bank Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (for Savings Account only)
5. Display All Accounts
6. Exit
Enter your choice: 6
Exiting...
PS E:\New folder>
```