Problem Statement: Simple Banking System

Design a simple banking System that allows customers to have savings and checking accounts. Customers should be able to deposited money, withdraw money and check their account balances.

BankAccount.java:

```
public interface BankAccount {
   void deposit(double amount);
   void withdraw(double amount);
   double getBalance();
}
```

BankCustomer.java:

```
public class BankCustomer {
    private String customerName;
    private SavingsAccount savingsAccount;
    private CheckingAccount checkingAccount;

    public BankCustomer(String customerName) {
        this.customerName = customerName;
        this.savingsAccount = new SavingsAccount();
        this.checkingAccount = new CheckingAccount();
    }

    public void addAccount(String accountType, BankAccount account) {
        if (accountType.equalsIgnoreCase("Savings")) {
            savingsAccount = (SavingsAccount) account;
        } else if (accountType.equalsIgnoreCase("Checking")) {
            checkingAccount = (CheckingAccount) account;
        } else {
            System.out.println("Invalid account type");
        }
    }

    public void deposit(String accountType, double amount) {
        if (accountType.equalsIgnoreCase("Savings")) {
            savingsAccount.deposit(amount);
        } else if (accountType.equalsIgnoreCase("Checking")) {
            checkingAccount.deposit(amount);
        } else {
            System.out.println("Invalid account type");
        }
```

```
    }

    public void withdraw(String accountType, double amount) {
        if (accountType.equalsIgnoreCase("Savings")) {
            savingsAccount.withdraw(amount);
        } else if (accountType.equalsIgnoreCase("Checking")) {
            checkingAccount.withdraw(amount);
        } else {
            System.out.println("Invalid account type");
        }
    }

    public void checkBalance(String accountType) {
        if (accountType.equalsIgnoreCase("Savings")) {
            System.out.println("Balance of Savings Account: " +
savingsAccount.getBalance());
        } else if (accountType.equalsIgnoreCase("Checking")) {
            System.out.println("Balance of Checking Account: " +
checkingAccount.getBalance());
        } else {
            System.out.println("Invalid account type");
        }
    }
}
```

CheckingAccount.java:

```
public class CheckingAccount implements BankAccount {
   private double balance;



   public void deposit(double amount) {
      balance += amount;
   }



   public void withdraw(double amount) {
      if (balance >= amount) {
         balance -= amount;
      } else {
         System.out.println("Insufficient funds");
      }
   }



   public double getBalance() {
      return balance;
   }
}
```

SavingsAccount.java:

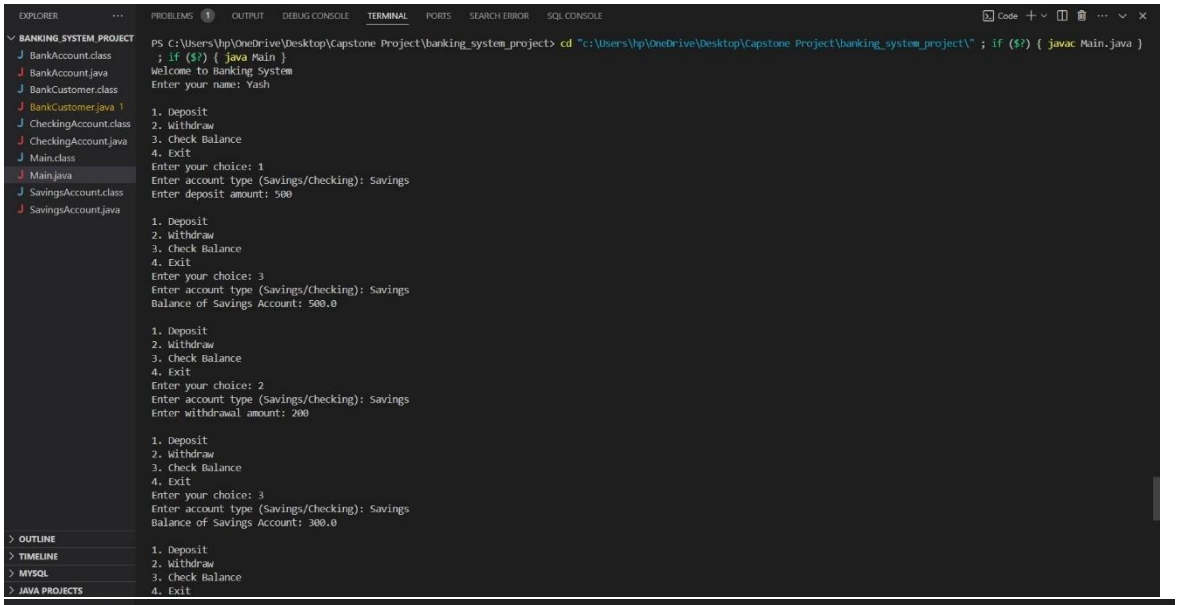
```
public class SavingsAccount implements BankAccount {
    private double balance;



    public void deposit(double amount) {
        balance += amount;
    }

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
        } else {
            System.out.println("Insufficient funds");
        }
    }



    public double getBalance() {
        return balance;
    }
}
```

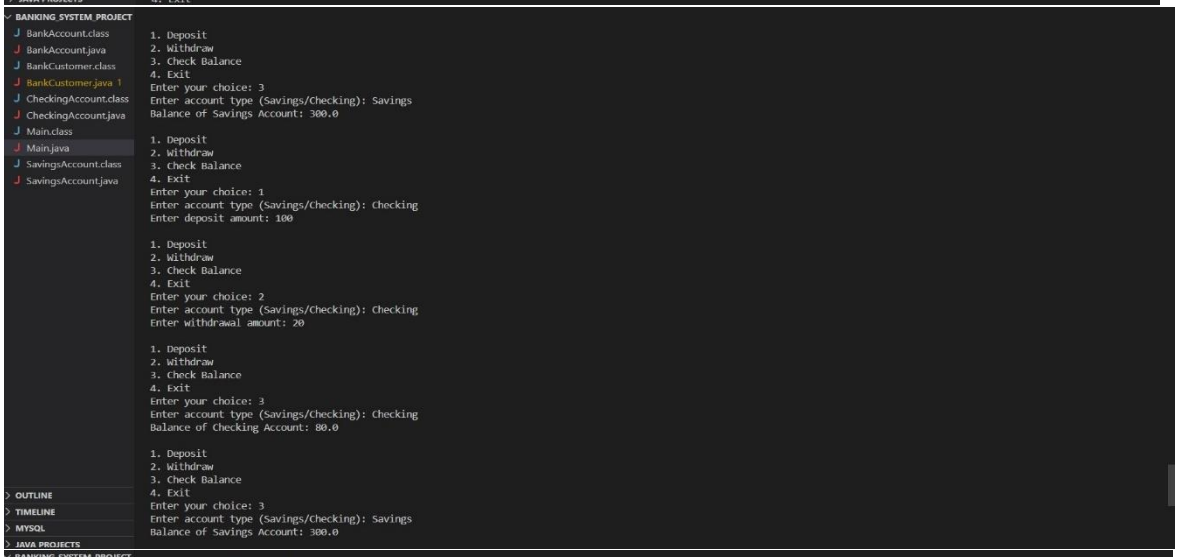
Main.java:

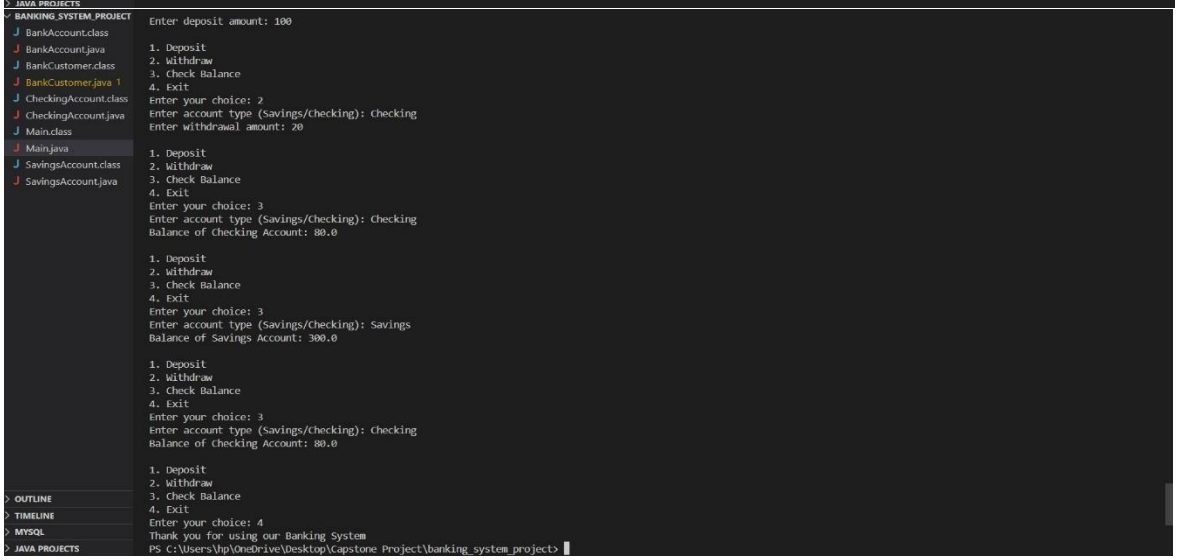
```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Welcome to Banking System");
        System.out.print("Enter your name: ");
        String customerName = scanner.nextLine();

        BankCustomer customer = new BankCustomer(customerName);
        SavingsAccount savingsAccount = new SavingsAccount();
        CheckingAccount checkingAccount = new CheckingAccount();

        customer.addAccount("Savings", savingsAccount);
        customer.addAccount("Checking", checkingAccount);

        boolean on = true;

        while (on) {
            System.out.println("\n1. Deposit\n2. Withdraw\n3. Check Balance\n4. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter account type (Savings/Checking): ");
                    String accountType = scanner.next();
                    System.out.print("Enter deposit amount: ");
```

```
                double depositAmount = scanner.nextDouble();
                customer.deposit(accountType, depositAmount);
                break;
            case 2:
                System.out.print("Enter account type (Savings/Checking): ");
                accountType = scanner.next();
                System.out.print("Enter withdrawal amount: ");
                double withdrawAmount = scanner.nextDouble();
                customer.withdraw(accountType, withdrawAmount);
                break;
            case 3:
                System.out.print("Enter account type (Savings/Checking): ");
                accountType = scanner.next();
                customer.checkBalance(accountType);
                break;
            case 4:
                on = false;
                System.out.println("Thank you for using our Banking System");
                break;
            default:
                System.out.println("Invalid choice");
        }
    }
    scanner.close();
  }
}
```

Outputs:


EXPLORER
BANKING_SYSTEM_PROJECT
BankAccount.class
BankAccount.java
BankCustomer.class
BankCustomer.java 1
CheckingAccount.class
CheckingAccount.java
Main.class
Main.java
SavingsAccount.class
SavingsAccount.java
OUTLINE
TIMELINE
MYSQL
JAVA PROJECTS
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SEARCH ERROR
SQL CONSOLE
Code
PS C:\Users\hp\OneDrive\Desktop\Capstone Project\banking_system_project> cd "c:\Users\hp\OneDrive\Desktop\Capstone Project\banking_system_project\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Welcome to Banking System
Enter your name: Yash
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 1
Enter account type (Savings/Checking): Savings
Enter deposit amount: 500
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 3
Enter account type (Savings/Checking): Savings
Balance of Savings Account: 500.0
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 2
Enter account type (Savings/Checking): Savings
Enter withdrawal amount: 200
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 3
Enter account type (Savings/Checking): Savings
Balance of Savings Account: 300.0
1. Deposit
2. Withdraw
3. Check Balance
4. Exit


BANKING_SYSTEM_PROJECT
BankAccount.class
BankAccount.java
BankCustomer.class
BankCustomer.java 1
CheckingAccount.class
CheckingAccount.java
Main.class
Main.java
SavingsAccount.class
SavingsAccount.java
OUTLINE
TIMELINE
MYSQL
JAVA PROJECTS
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 3
Enter account type (Savings/Checking): Savings
Balance of Savings Account: 300.0
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 1
Enter account type (Savings/Checking): Checking
Enter deposit amount: 100
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 2
Enter account type (Savings/Checking): Checking
Enter withdrawal amount: 20
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 3
Enter account type (Savings/Checking): Checking
Balance of Checking Account: 80.0
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 3
Enter account type (Savings/Checking): Savings
Balance of Savings Account: 300.0


BANKING_SYSTEM_PROJECT
BankAccount.class
BankAccount.java
BankCustomer.class
BankCustomer.java 1
CheckingAccount.class
CheckingAccount.java
Main.class
Main.java
SavingsAccount.class
SavingsAccount.java
OUTLINE
TIMELINE
MYSQL
JAVA PROJECTS
Enter deposit amount: 100
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 2
Enter account type (Savings/Checking): Checking
Enter withdrawal amount: 20
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 3
Enter account type (Savings/Checking): Checking
Balance of Checking Account: 80.0
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 3
Enter account type (Savings/Checking): Savings
Balance of Savings Account: 300.0
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 3
Enter account type (Savings/Checking): Checking
Balance of Checking Account: 80.0
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 4
Thank you for using our Banking System
PS C:\Users\hp\OneDrive\Desktop\Capstone Project\banking_system_project>