

SLTC Research University

Mid Assignment

CCS 313

AA 1711

SOA & Microservices

Mr. Chamin De Silva



Due Date
November 30,2022

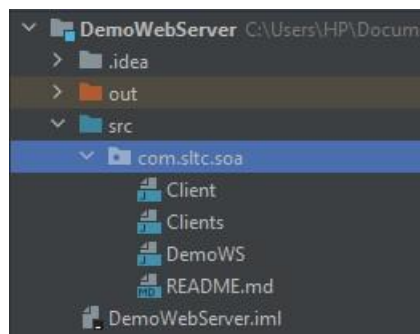
Question 01

I Created the **SOAP** based Web service for the Bank with wsdl using java language. Firstly, we have to code the web server part. Because here we are using soap web-based service. With my application I provided these features to clients,

- **Create account** in the bank server with minimum Rs:1000/= deposit.
- **Login** with user name & password to server.
- **Check balance** of own account.
- **Deposit** Money to Own Account.
- **Withdraw** Money from Own Account.
- **Transfer** Money from Own Account to Another account.
- **Password protection** with each transaction.

So, many functions in the server package than the Web Service Client.

As you can see, apart from the “**DemoWS**” class, I Implement the “**Client**” class for handling the client’s data. Further, I created “**Clients**” class to store the all the client classes and In the Clients class I created 02 HashMap’s for Username Sorted and Account sorted to get the client details. So, there are 03 class in the server side and main class is DemoWS class.



In the main method in DemoWS class, I set the URL to host the wsdl and show users that wsdl file is hosting now like below.

```
public static void main(String[] args){
    Endpoint.publish("http://localhost:8888/DemoWebService", new DemoWS());
    System.out.println("=====");//Display that sever was began.
    System.out.println("===== Bank Server Starts =====");
    System.out.println("=====");
}
```

As I mentioned below, I implemented 08 functions for wsdl that necessary to communicate with client. Basically, these functions do something and return the response to web service client that got. There are variety of responses like, ints, floats, Booleans and Strings.

```

8  @WebService
9  @SOAPBinding(style = SOAPBinding.Style.DOCUMENT)
10 public class DemoWS {
11     Client connectedClient;
12     @WebMethod
13     public int authoriseUser(String userName, int password) {
14         if(Client.getUsernameList().contains(userName)){
15             connectedClient = Clients.clientProfilesByUserName.get(userName);
16             if(connectedClient.getPassword() == password){
17                 connectedClient.showClient();
18                 return 1;
19             }
20             return -1;
21         }
22         else
23             return -2;
24     }
25     @WebMethod
26     public boolean checkUserName(String userName){
27         if(Client.getUsernameList().contains(userName))
28             return true;
29         else
30             return false;
31     }
32     @WebMethod
33     public boolean checkUserAccount(String userAcc){
34         if(Client.getAccountNumberList().contains(userAcc))
35             return true;
36         else
37             return false;
38     }
39     @WebMethod
40     public boolean checkNIC(String nic){
41         if(Client.getNICList().contains(nic))
42             return true;
43         else
44             return false;
45     }
46
47
48     @WebMethod
49     public int addNewClient(String firstName, String lastName, String nic, String username, int password, float amount){
50         Client temporaryClient = new Client(firstName,lastName,nic,username,password,amount);
51         Clients.clientProfilesByUserName.put(username,temporaryClient);
52         Clients.userNameByAccountNumber.put(temporaryClient.getAccountNumber(),username);
53         return 0;
54     }
55
56     @WebMethod
57     public int deposit(float amount, String username){
58         connectedClient.deposit(amount);
59         return 1;
60     }
61     @WebMethod
62     public int withdraw(float amount, String username){
63         return connectedClient.withdraw(amount);
64     }
65     @WebMethod
66     public int transfer(String username, float amount, String userAccount){
67         return connectedClient.transfer(amount,userAccount);
68     }
69     @WebMethod
70     public float getBalance(String username){
71         float balance = Clients.clientProfilesByUserName.get(username).getBalance();
72         System.out.println( " Updated Balance of "+username+" is RS: " + balance+"/=");
73         return balance;
74     }
75     @WebMethod
76     public String getAccountNumber(String username){
77         String accountNumber = Clients.clientProfilesByUserName.get(username).getAccountNumber();
78         return accountNumber;
79     }

```

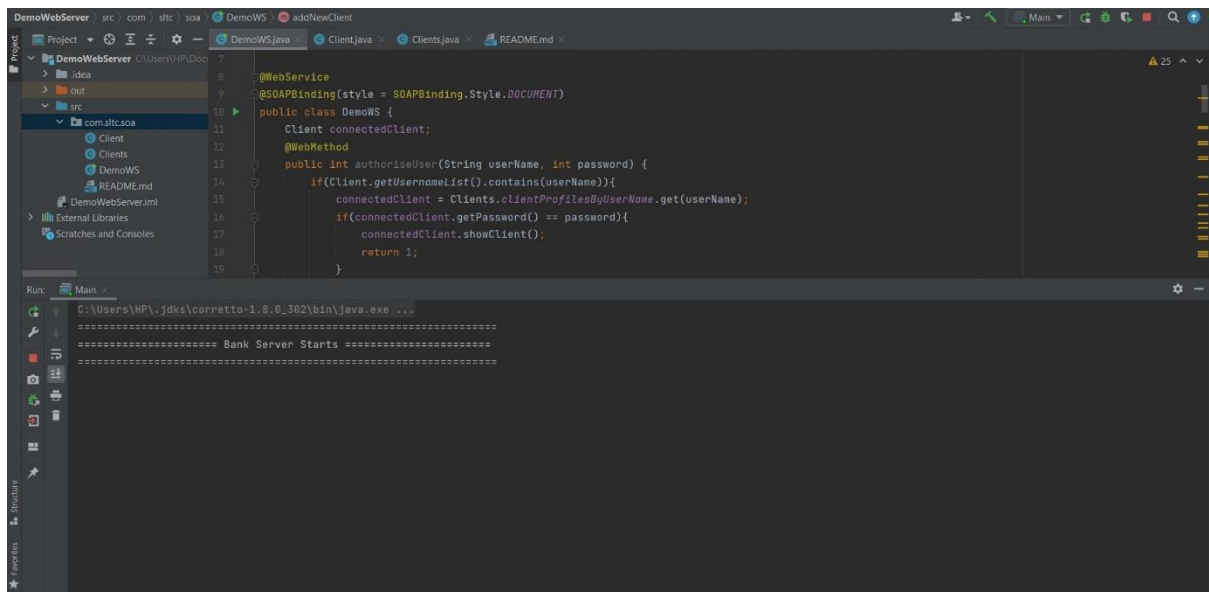
In the **Client Class** I defined variables that helps to verify user and need to be in user bank account. I Implemented basic functions that do with user account like below. Additionally generated getter & setters.

```
5 public class Client {
6     private String FirstName;
7     private String LastName;
8     private String NIC;
9     private String username;
10    private int Password;//store hashcode password
11    private String accountNumber;
12    private float balance = 1000;
13    private static HashSet<String> accountNumberList = new HashSet<>();
14    protected static HashSet<String> usernameList = new HashSet<>();
15    protected static HashSet<String> NICList = new HashSet<>();
16
17    public Client(String firstName, String lastName, String nic, String username, int password, float balance) {
18        FirstName = firstName;
19        LastName = lastName;
20        NIC = nic;
21        this.username = username;
22        Password = password;
23        this.balance += balance;
24        int acc = new Random().nextInt( bound: (9999 - 100) + 1) + 10;
25        while(accountNumberList.contains(Integer.toString(acc))){
26            acc = new Random().nextInt( bound: (9999 - 100) + 1) + 10;
27        }
28        this.accountNumber = Integer.toString(acc);
29        accountNumberList.add(accountNumber);
30        usernameList.add(username);
31        NICList.add(NIC);
32    }
33
34    public void showClient(){
35        String profile= "First Name : "+FirstName+"\nLast Name : "+LastName+"\nNIC no : "+NIC
36            +"\nUser Name : "+username+"\nAccount Number : "+accountNumber+"\nBalance : "+balance+"\n";
37        System.out.println(profile);
38    }
39    public void deposit(float amount) { balance+=amount; }
40
41
42
43    public int transfer(float amount, String accountNumber){
44        if(amount <= balance){ //success
45            balance-=amount;
46            String tempUserName = Clients.userNameByAccountNumber.get(accountNumber);
47            Clients.clientProfilesByUserName.get(tempUserName).deposit(amount);
48            return 1;
49        }else{ //insufficient balance
50            return 0;
51        }
52    }
53    public int withdraw(float amount){
54        if(amount <= balance){ //success
55            balance-=amount;
56            return 1;
57        }else{ //insufficient balance
58            return 0;
59        }
60    }
61
62    public String getFirstName() { return FirstName; }
63
64
65    public void setFirstName(String firstName) { FirstName = firstName; }
```

Here the **Clients** class.

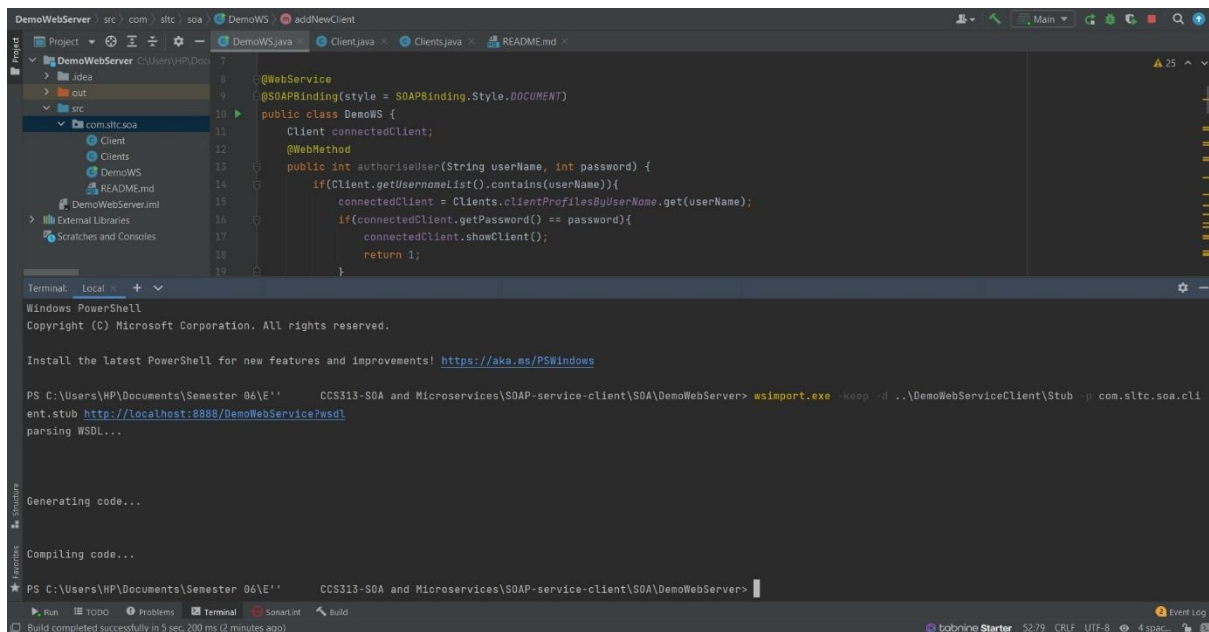
```
5 public class Clients extends HashMap {
6     public static HashMap<String, Client> clientProfilesByUserName = new HashMap<String, Client>();
7     public static HashMap<String, String> userNameByAccountNumber = new HashMap<String, String>();
8
9 }
```

That's all from the DemoWebServer package. Then we have to generate the preferred java class to our **DemoWebServiceClient** package. There is a **wsimport.exe** in the jdk/bin file to do this job for me. So, I run the below commands in the terminal of DemoWebServer package. So firstly, we have to **run the DemoWS java file** Like this,



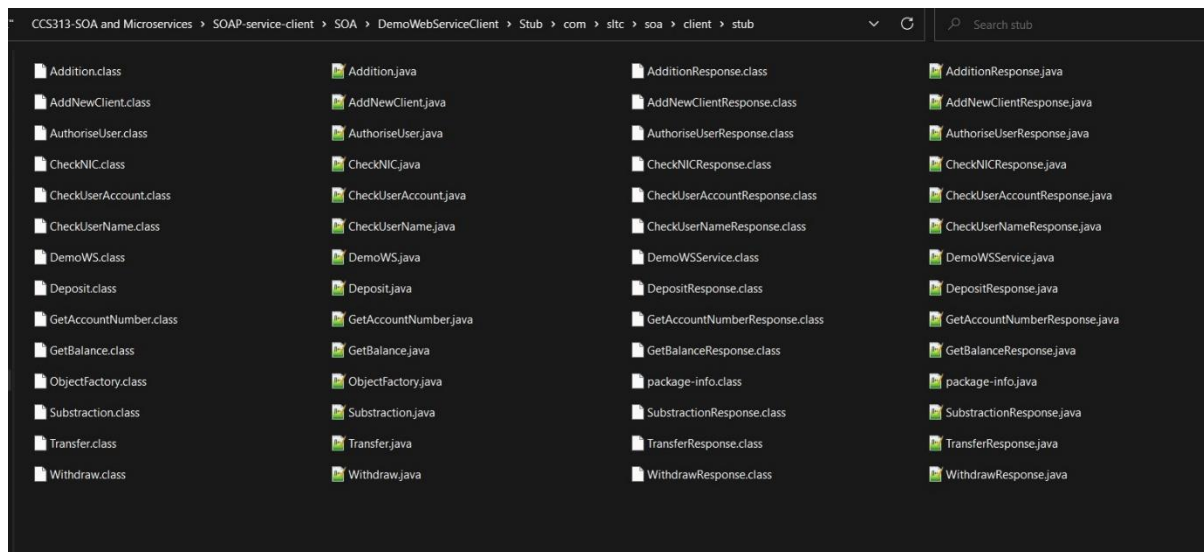
The screenshot shows an IDE with the `DemoWS.java` file open. The file is part of the `com.sltc.soa` package and implements the `@WebService` interface. It contains a `connectedClient` attribute and an `authoriseUser` method. The `authoriseUser` method checks if the username is in the `ClientProfilesByUserName` list and if the password matches. If both conditions are met, it calls `showClient()` and returns 1. The `Run` tab at the bottom shows the output of the `java.exe` command, which displays "Bank Server Starts".

After running the DemoWS java file then we can open terminal and **type the command** like this and press enter.

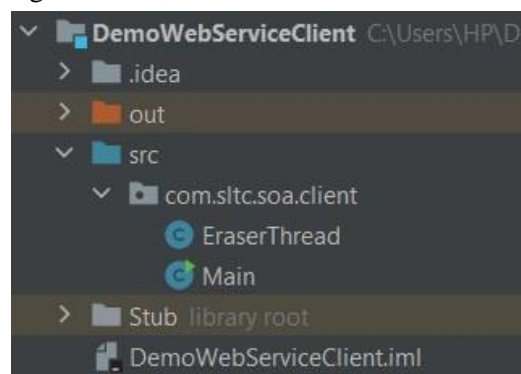


The screenshot shows the same IDE with the `DemoWS.java` file open. The `Terminal` tab at the bottom shows the execution of the `wsimport.exe` command. The command is: `wsimport.exe -keep -d ..\DemoWebServiceClient\Stub -p com.sltc.soa.client.stub http://localhost:8888/DemoWebService?wsdl`. The output shows that the command is parsing the WSDL and generating code. The `Run` tab at the bottom shows the output of the `wsimport.exe` command, which displays "Generating code..." and "Compiling code...".

Then eventually **generate the java class** files according to wsdl file in the Stub folder in the WebServerClient package like this,



Then we have to do further development in the **WebServerClient** package. So, there are two classes in my WebServerClient package like this,



In here I Created the **EraserThread** class for mask the password that user input in the command line because creating these kinds of applications for the banks, should be reliable application. Eraser Thread class like below,

```
1 package com.sltc.soa.client;
2 import static java.lang.Thread.sleep;
3
4 class EraserThread implements Runnable {
5     private boolean stop;
6     public EraserThread() {}
7     public void run() {
8         stop = true;
9         while (stop) {
10             try {
11                 System.out.print("\010*");
12                 System.out.print("\010*");
13                 System.out.print("\010*");
14                 sleep(100);
15             } catch (InterruptedException e) {
16                 e.printStackTrace();
17             }
18         }
19     }
20 }
```


In the Main java class, I implemented some functions to get user input with user-friendly Command line application and using the classes that **generated** in Stub folder I created, I communicate with the **wsdl** file that runs on the **port**. So, I called functions in the **main** class that are created in the DemoWS class. Here is the **Main** class look like,

```

13 public static void main(String[] args) {
14     clientTerminal();
15 }
16
17 private static void clientTerminal() {
18
19     DemoWSService demoWSService = new DemoWSService();
20     DemoWS demoWSPort = demoWSService.getDemoWSPort();
21     Scanner sc = new Scanner(System.in);
22
23     System.out.println("=====");
24     System.out.println("Welcome to MID BANK - WEB Server =====");
25     System.out.println("AA 1711 - NuYuN Pabasara =====");
26     System.out.println("=====");
27     System.out.println("Enter 'quit' and press enter, any time to quit from the console.");
28     System.out.println("If you are a new user please Type create and press enter:");
29     System.out.print("Please Enter Your User ID : ");
30     String name;
31     String fName;
32     String lName;
33     String nic;
34     int password;
35     String userAccount;
36     float deposit;
37     String menuKey = "";
38     int hashCode;
39
40     for (name = sc.nextLine(); !name.toLowerCase().equals("quit"); name = sc.nextLine()) { //Get the unique Name ID from user
41         int returnValue = -1;
42         if (name.toLowerCase().equals("create")) {
43             System.out.print("First Name : ");
44             fName = sc.nextLine();
45             System.out.print("Last Name : ");
46             lName = sc.nextLine();
47             System.out.print("User Name : ");
48             name = sc.nextLine();
49             while (demoWSPort.checkUserName(name) || name.isEmpty()) {
50                 System.out.print("Enter Valid User Name : ");
51                 name = sc.nextLine();
52             }
53             System.out.print("NIC : ");
54             nic = sc.nextLine();
55         }
56     }
57 }

```

After 50 lines later, code look like this,

```

110
111 while (!menuKey.toLowerCase().equals("quit")) {
112
113     System.out.println("=====");
114     System.out.println();
115     System.out.println("1.check Balance");
116     System.out.println("2.Money Deposit");
117     System.out.println("3.Money Withdraw");
118     System.out.println("4.Money Transfer");
119     System.out.println("<<Type \"quit\" to exit>>");
120     System.out.print("Enter the number do you want to proceed :");
121     menuKey = sc.nextLine();
122
123     if (menuKey.toLowerCase().equals("1")) {
124         System.out.println("Account No : " + demoWSPort.getAccountNumber(CurrentClientName));
125         System.out.println("Balance : " + demoWSPort.getBalance(CurrentClientName));
126
127     } else if (menuKey.toLowerCase().equals("2")) {
128
129         System.out.print("Enter amount for deposit : ");
130         Float amount = readInputInt();
131         // System.out.print("Enter your Password : ");
132         if (demoWSPort.authoriseUser(CurrentClientName, consoleFunc().hashCode()) == 1) {
133             demoWSPort.deposit(amount, CurrentClientName);
134             System.out.println("Deposit Successful:");
135             System.out.println("Updated Balance : Rs:- " + demoWSPort.getBalance(CurrentClientName) + " /=");
136         } else {
137             System.out.println("Password Incorrect. Please Try again:");
138         }
139     }
140
141     } else if (menuKey.toLowerCase(Locale.ROOT).equals("3")) {
142
143         int response = 0;
144         System.out.print("Enter amount for withdraw : ");
145         Float amount = readInputInt();
146         // System.out.print("Enter your Password : ");
147         if (demoWSPort.authoriseUser(CurrentClientName, consoleFunc().hashCode()) == 1) {
148             response = demoWSPort.withdraw(amount, CurrentClientName);
149         }
150     }
151 }

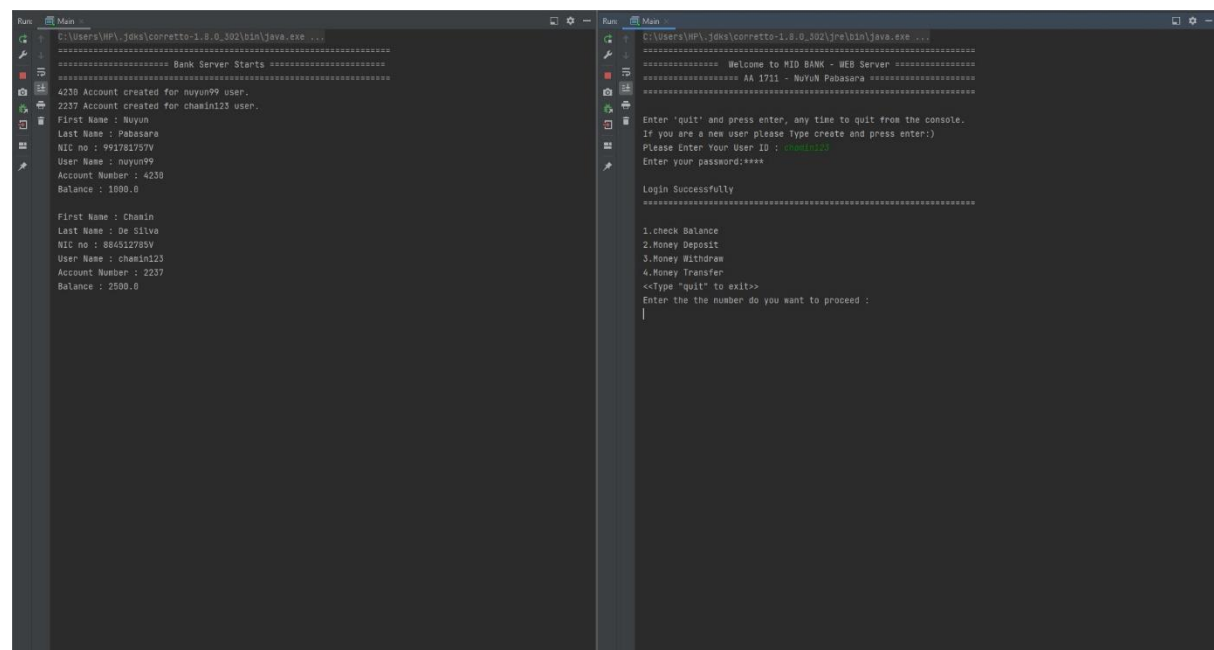
```

In the end of Main class look like this,

```
187 public static String consoleFunc() {
188     Scanner scan = new Scanner(System.in);
189     System.out.print("Enter your password: ");
190     EraserThread et = new EraserThread();
191     Thread mask = new Thread(et);
192     mask.start();
193     String pass = scan.nextLine();
194     mask.stop();
195     return pass;
196 }
197
198 private static float readInputInt() {
199     float inputInt = 0;
200     boolean numberFound = false;
201     Scanner scan = new Scanner( System.in );
202     do {
203         String inputStr = scan.next();
204         try {
205             inputInt = Float.parseFloat(inputStr);
206             numberFound = true;
207         } catch( Exception e ) {
208             System.out.println( "Invalid input " + inputStr + ". Please input a number." );
209             System.out.print( "Please input The amount: Rs: " );
210         }
211     } while( !numberFound );
212     return inputInt;
213 }
214
215 }
```

Further, I add the zip file that including the server & client codes. Here I add some screenshots of my command line application that shows you how this application works.

- **User Creation** - Can't login with create the client in server.



- **User Login** - Have to enter correct user name and password.

```

C:\Users\HP\jds\corretto-1.8.0_302\bin\java.exe ...
===== Bank Server Starts =====
4230 Account created for nuyun99 user.
2237 Account created for chamin23 user.

C:\Users\HP\jds\corretto-1.8.0_302\bin\java.exe ...
===== Welcome to MID BANK - WEB Server =====
===== AA 1711 - Nuyun Pabasara =====
=====

Enter 'quit' and press enter, any time to quit from the console.
If you are a new user please Type create and press enter:)
Please Enter Your User ID : nuyun99
Enter your password:****

Incorrect Username or Password, Try Again.
Please Enter Your User ID : chamin23
First Name : Nuyun
Last Name : Pabasara
User Name : nuyun99
NIC : 991761757V
Enter your password:****
Enter amount for deposit : 0
Do you want to proceed the request (y/n) : y
Account Creation Successful
Please Enter Your User ID to login: nuyun99
First Name : Nuyun
Last Name : De Silva
User Name : chamin23
NIC : 884512785V
Enter your password:****
Enter amount for deposit : 1000
Do you want to proceed the request (y/n) : y
Account Creation Successful
Please Enter Your User ID to login:

```

- **Check Balance** - shows the current balance with account number (onetime Generated account number)

```

C:\Users\HP\jds\corretto-1.8.0_302\bin\java.exe ...
===== Bank Server Starts =====
4230 Account created for nuyun99 user.
2237 Account created for chamin23 user.
First Name : Nuyun
Last Name : Pabasara
NIC no : 991761757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

First Name : Chamin
Last Name : De Silva
NIC no : 884512785V
User Name : chamin23
Account Number : 2237
Balance : 2500.0

First Name : Nuyun
Last Name : Pabasara
NIC no : 991761757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 1000.0/-

C:\Users\HP\jds\corretto-1.8.0_302\bin\java.exe ...
===== Welcome to MID BANK - WEB Server =====
===== AA 1711 - Nuyun Pabasara =====
=====

Enter 'quit' and press enter, any time to quit from the console.
If you are a new user please Type create and press enter:)
Please Enter Your User ID : nuyun99
Enter your password:****

Login Successfully
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
1
Account No : 4230
Balance : 1000.0
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :

```

- **Deposit** - Do deposit function with getting again user password.

```

Run Main
C:\Users\HP\jdk\corretto-1.8.0.302\bin\java.exe ...
===== Bank Server Starts =====
4230 Account created for nuyun99 user.
2237 Account created for chamin123 user.
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781797V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

First Name : Chamin
Last Name : De Silva
NIC no : 884512785V
User Name : chamin123
Account Number : 2237
Balance : 2500.0

First Name : Nuyun
Last Name : Pabasara
NIC no : 991781797V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 1000.0/-
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781797V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 3500.0/-

Run Main
C:\Users\HP\jdk\corretto-1.8.0.302\bin\java.exe ...
===== Welcome to MID BANK - WEB Server =====
===== AA 1711 - Nuyun Pabasara =====

Enter 'quit' and press enter, any time to quit from the console.
If you are a new user please Type create and press enter:)
Please Enter Your User ID : nuyun99
Enter your password:****

Login Successfully
=====
1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
1
Account No : 4230
Balance : 1000.0
=====
1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
2
Enter amount for deposit : 2500
Enter your password:****
Deposit Successful:)
Updated balance : RS:3500.0/-
=====
1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :

```

- **Withdraw** - Do withdraw function with getting again user password.

```

Run Main
C:\Users\HP\jdk\corretto-1.8.0.302\bin\java.exe ...
===== Bank Server Starts =====
4230 Account created for nuyun99 user.
2237 Account created for chamin123 user.
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781797V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

First Name : Chamin
Last Name : De Silva
NIC no : 884512785V
User Name : chamin123
Account Number : 2237
Balance : 2500.0

First Name : Nuyun
Last Name : Pabasara
NIC no : 991781797V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 1000.0/-
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781797V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 3500.0/-
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781797V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 3500.0/-

Run Main
C:\Users\HP\jdk\corretto-1.8.0.302\bin\java.exe ...
1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
1
Account No : 4230
Balance : 1000.0
=====
1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
2
Enter amount for deposit : 2500
Enter your password:****
Deposit Successful:)
Updated balance : RS:3500.0/-
=====
1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
3
Enter amount for withdraw : 500.56
Enter your password:****
Withdraw Successful:)
Updated balance : RS:2914.44/-
=====
1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :

```

- **Transfer** - Transfer money to other authorized account number.

```
Run Main
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

First Name : Chamin
Last Name : De Silva
NIC no : 884512785V
User Name : chamin123
Account Number : 2237
Balance : 2500.0

First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 1000.0/+
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 3500.0/+
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 3500.0

Updated Balance of nuyun99 is RS: 2914.44/+
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 2914.44

Updated Balance of nuyun99 is RS: 2000.0/+

Run Main
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
3
Enter amount for deposit : 2500
Enter your password:****
Deposit Successful:)
Updated balance : Rs:3500.0/+
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
3
Enter amount for withdraw : 500.00
Enter your password:****
Withdraw Successful:)
Updated balance : Rs:2914.44/+
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
3
Enter amount for Transfer : 914.44
Enter the user Account : 2237
Enter your password:****
Transfer Successful:)
Updated balance : Rs:2000.0/+
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
```

- **Exit from server** - Exit from console by typing quit anytime.

```
Run Main
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

First Name : Chamin
Last Name : De Silva
NIC no : 884512785V
User Name : chamin123
Account Number : 2237
Balance : 2500.0

First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 1000.0/+
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 3500.0/+
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 3500.0

Updated Balance of nuyun99 is RS: 2914.44/+
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 2914.44

Updated Balance of nuyun99 is RS: 2000.0/+

Run Main
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
3
Enter amount for deposit : 2500
Enter your password:****
Deposit Successful:)
Updated balance : Rs:3500.0/+
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
3
Enter amount for withdraw : 500.00
Enter your password:****
Withdraw Successful:)
Updated balance : Rs:2914.44/+
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
3
Enter amount for Transfer : 914.44
Enter the user Account : 2237
Enter your password:****
Transfer Successful:)
Updated balance : Rs:2000.0/+
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the the number do you want to proceed :
quit

Process finished with exit code 0
```

- **Check balance** - Check the balance in other authorized user to make sure money was transferred.

```

Run Main
User Name : chamin123
Account Number : 2237
Balance : 2500.0

First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 1000.0/-
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 1000.0

Updated Balance of nuyun99 is RS: 3500.0/-
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 3500.0

Updated Balance of nuyun99 is RS: 2914.44/-
First Name : Nuyun
Last Name : Pabasara
NIC no : 991781757V
User Name : nuyun99
Account Number : 4230
Balance : 2914.44

Updated Balance of nuyun99 is RS: 2000.0/-
First Name : Chamin
Last Name : De Silva
NIC no : 884512785V
User Name : chamin123
Account Number : 2237
Balance : 3414.44

Updated Balance of chamin123 is RS: 3414.44/-

Run Main
C:\Users\HP\jdk\corretto-1.8.0_302\jre\bin\java.exe ...
===== Welcome to MID BANK - WEB Server =====
===== SA 1711 - Nuyun Pabasara =====

Enter 'quit' and press enter, any time to quit from the console.
If you are a new user please Type create and press enter:)
Please Enter Your User ID : chamin123
Enter your password:****

Login Successfully

=====
1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the number do you want to proceed :
1
Account No : 2237
Balance : 3414.44
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the number do you want to proceed :
3
=====

1.check Balance
2.Money Deposit
3.Money Withdraw
4.Money Transfer
<<Type "quit" to exit>>
Enter the number do you want to proceed :
quit
=====

Process finished with exit code 0

```

That's all the basic scenarios I made for the screenshots. Further, I try to handle expectations and get the true value from the user. You can run my code with you command line and check it furthermore if needed. Thank you.

Question 02

If I had to implement the same for a real-world bank, I will consider things like,

- Reliability
- Performance
- Aesthetics
- Usability
- Supportability
- Integrity
- Maintainability

So, Bank system have to make the server very reliable and functional for the clients, Use aesthetics GUIs for better user experience, Supportability of other systems like ATMs, other bank servers, etc., Usable for clients and easy to use main features, Maintain integrity level for manage the system more secure and Security system that breachers can't attack to the system. Further, we can add OTP verification, fingerprint to increase the security. For better supportability and performance, we can moving to Rest API for real-world bank system. But Soap is more secure than rest APIs so we can use both technologies to overcome this.

