Informatics Institute of Technology
Department of Computing

Module: 4COSC0010C – Programming Principles 02

Degree Program: BScCS

Tutorial Group: Group A

Module Leader: Mr. Guganathan Poravi

Coursework 01 – Stage 3

Date of submission: 25.02.2018

Student ID - IIT Student ID: 2017091

UoW ID : 16737363/1

Student First Name: Dinithi

Student Surname : Jayasekara

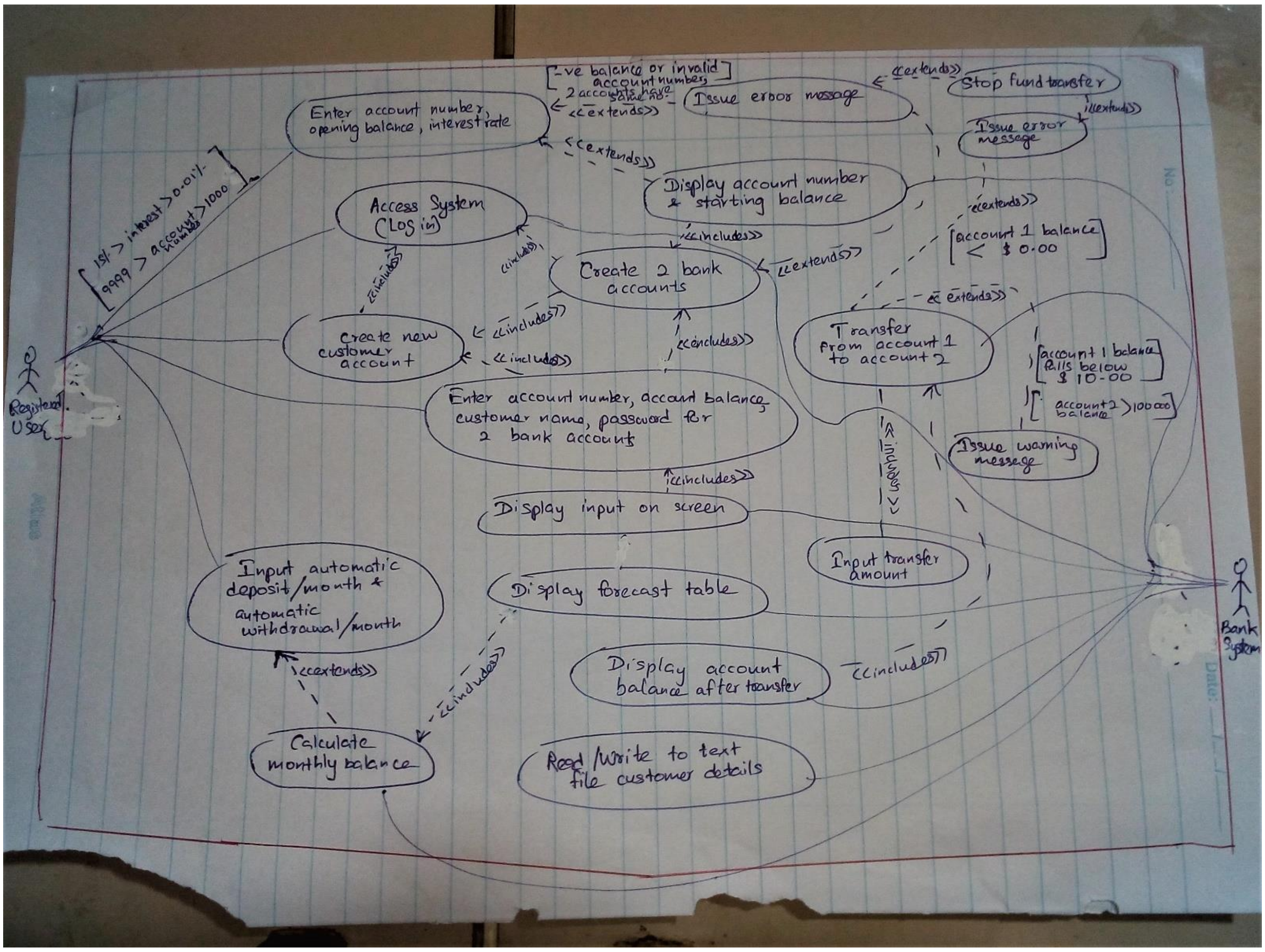
Table of Contents

1. INTRODUCTION

This is a JAVA console-based application developed for InterBanking Pty. The purpose of this system is to produce a next generation customer and account management system. This application has been developed for employees of the bank to add new customer accounts and bank accounts and is very user friendly serving its purpose to the fullest.

2. DESIGN

UML DIAGRAM:



Registered User
Bank System
[1% > interest > 0.01% / 9999 > account number > 1000]
Enter account number, opening balance, interest rate
[-ve balance or invalid account numbers, 2 accounts have same no.]
<<extends>>
Issue error message
<<extends>>
Stop fund transfer
<<extends>>
Issue error message
<<extends>>
Display account number & starting balance
Access System (Log in)
<<includes>>
<<includes>>
<<includes>>
Create 2 bank accounts
<<extends>>
<<extends>>
[account 1 balance < $ 0.00]
<<extends>>
Create new customer account
<<includes>>
<<includes>>
<<includes>>
Transfer from account 1 to account 2
[account 1 balance falls below $ 10-00]
[account2 balance > 100 000]
Enter account number, account balance, customer name, password for 2 bank accounts
<<includes>>
Issue warning message
<<includes>>
Display input on screen
Input transfer amount
Input automatic deposit/month & automatic withdrawal/month
Display forecast table
<<extends>>
<<includes>>
Display account balance after transfer
<<includes>>
Calculate monthly balance
Read/Write to text file customer details

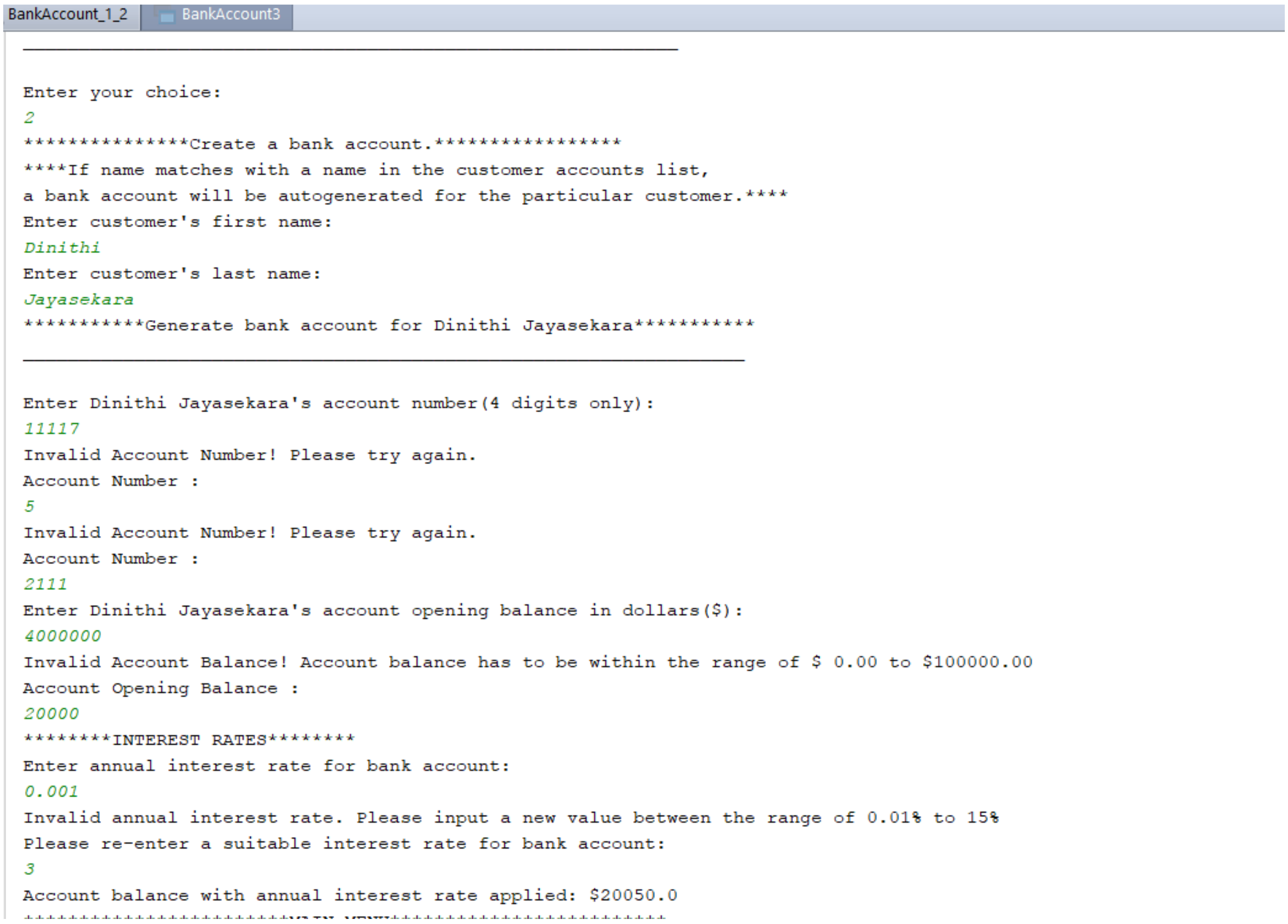
3. IMPLEMENTATION

```
package com.company;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Scanner;


public class BankAccount3 {
    private static ArrayList<customerAccount> customerAccount = new ArrayList ();
    private static ArrayList<BankAccount> accounts = new ArrayList ();
    public static final Scanner sc = new Scanner (System.in);
    public static final DecimalFormat df = new DecimalFormat ("#.##");

    public static void main(String[] args) throws IOException {

        System.out.println ("***************Welcome to Interbanking
Pty.***************");
        System.out.println ("*****************Account Management
Unit*****************");
        System.out.println ("*************_______Login______*************\n");
        System.out.println ("Enter Username: ");
        String empName = sc.next ();
        System.out.println ("Enter Password: ");
        String empPassword = sc.next ();

        if (!Login (empName, empPassword)) {
            System.out.println ("Please enter a valid username or password");
            main (null);
        } else
            Options ();


        char choice = '\0';

        do {
            choice = sc.next ().charAt (0);
            switch (choice) {
                case '1':
                    one ();
                    Options ();
                    break;


                case '2':
                    two ();
                    Options ();
                    break;


                case '3':
                    three ();
                    Options ();
                    break;

                case '4':
                    four ();
                    Options ();
                    break;

                case '5':
                    five ();
                    Options ();
                case '6':
                    six ();
                    Options ();
                case '0':
                    six ();
                    System.out.println ("Thank you for your valuable service.");
                    System.exit (0);
                default:
                    System.out.println ("Enter your choice: ");
            }
        } while (choice != '0');

    }


    public static boolean Login(String userName, String password) {
        if (userName.equals ("admin") && password.equals ("admin")) {
            return true;
        }
        return false;
    }

    public static void Options() {
        System.out.println ("***********************MAIN
MENU************************");
        System.out.println ("--------------------------------------------------------
--\n");
        System.out.println ("          1 - Create a new Customer Account.");
        System.out.println ("          2 - Generate a Bank Account. (Options include
applying annual interest rate and input of opening balance.)");
        System.out.println ("          3 - Display Bank Account Details.");
        System.out.println ("          4 - Transfer cash from bank account 1 to bank
account 2 of a particular customer.");
        System.out.println ("          5 - View forecast of yearly balance for each
month with interest rates,automatic " +
                "withdrawals & automatic deposits applied.");
        System.out.println ("          0 - Exit. Customer account details will be
written into a file. ");
        System.out.println
("_____________________________________________________________\n");
        System.out.println ("Enter your choice: ");
    }


    public static void one() {
        System.out.println ("Your choice is 1");
        System.out.println ("Enter customer's first name: ");
        String fName = sc.next ();
        System.out.println ("Enter customer's last name: ");
        String lName = sc.next ();
        System.out.println ("Enter customer's username: ");
        String username = sc.next ();
        System.out.println ("Enter password for customer: ");
        String password = sc.next ();

        customerAccount newUser = new customerAccount ();
        newUser.fName = fName;
        newUser.lName = lName;
        newUser.username = username;
        newUser.password = password;
        customerAccount.add (newUser);
    }

    public static void two() {
        System.out.println ("***************Create a bank
account.*****************");
        System.out.println ("****If name matches with a name in the customer accounts
list, " +
                "\na bank account will be autogenerated for the particular
customer.****");
        System.out.println ("Enter customer's first name: ");
        String first = sc.next ();
        System.out.println ("Enter customer's last name: ");
        String last = sc.next ();
        BankAccount newAccount = new BankAccount ();
        for (int i = 0; i < customerAccount.size (); i++) {
            if ((customerAccount.get (i).fName.equalsIgnoreCase (first)) &&
                    (customerAccount.get (i).lName.equalsIgnoreCase (last))) {
                System.out.println ("***********Generate bank account for " + first +
" " + last + "***********");
                System.out.println
("_________________________________________________________________\n");
                System.out.println ("Enter " + first + " " + last + "'s" + " account
number(4 digits only): ");
                newAccount.accountNo = sc.nextInt ();
                while (!(newAccount.accountNo >= 1000 && newAccount.accountNo <
10000)) {
                    System.out.println ("Invalid Account Number! Please try again.");
                    System.out.println ("Account Number : ");
                    newAccount.accountNo = sc.nextInt ();
                }
                System.out.println ("Enter " + first + " " + last + "'s" + " account
opening balance in dollars($): ");
                newAccount.accountBal = sc.nextDouble ();
                while (newAccount.accountBal < 0 || newAccount.accountBal > 100000) {
                    System.out.println ("Invalid Account Balance! Account balance has
to be within the range of $ 0.00 to $100000.00");
                    System.out.println ("Account Opening Balance :");
                    newAccount.accountBal = sc.nextDouble ();
                }
                System.out.println ("********INTEREST RATES********");
                System.out.println ("Enter annual interest rate for bank account: ");
                newAccount.interest = sc.nextDouble ();
                do {
                    if (!(newAccount.interest > 0.01 && newAccount.interest < 15)) {
                        System.out.println ("Invalid annual interest rate. Please
input a new value between the range of 0.01% to 15%");
                        System.out.println ("Please re-enter a suitable interest rate
for bank account: ");
                        newAccount.interest = sc.nextDouble ();
                    }
                } while ((!(newAccount.interest > 0.01 && newAccount.interest <
15)));
                newAccount.accountBal = (newAccount.accountBal *
((newAccount.interest / 100) * 1 / 12)) + newAccount.accountBal;
                System.out.println ("Account balance with annual interest rate
applied: " + "$" + Double.parseDouble (df.format (newAccount.accountBal)));
                accounts.add (newAccount);
            } else if (!((customerAccount.get (i).fName.equalsIgnoreCase (first)) &&
                    (customerAccount.get (i).lName.equalsIgnoreCase (last)))) {
                System.out.println ("Customer details mismatch. Please check customer
names for spelling errors or case sensitivity.");
            }
        }
    }

    public static void three() {
        int count = 0;
        System.out.println ("***************Display bank account
details.*****************");
        System.out.println
("_____________________________________________________________\n");
        System.out.println ("***Bank details are displayed only if the bank account
number matches with the stored customer account details.***");
        System.out.println ("Enter customer's first name: ");
        String name1 = sc.next ();
        System.out.println ("Enter customer's last name: ");
        String name2 = sc.next ();
        for (int i = 0; i < accounts.size (); i++) {
            for (i = 0; i < customerAccount.size (); i++) {
                if ((customerAccount.get (i).fName.equalsIgnoreCase (name1)) &&
(customerAccount.get (i).lName.equalsIgnoreCase (name2))) {
                    System.out.println ("Account holder's full name: " +
                            customerAccount.get (i).fName + " "
                            + customerAccount.get (i).lName);
                    System.out.println ("No. of accounts for customer: " +
accounts.size ());
                    for (i = 0; i < accounts.size (); i++) {
                        count = count + 1;
                        System.out.println ("Account " + count + " interest rate: " +
accounts.get (i).interest);
                        System.out.println ("Account " + count + " Balance with
interest rate applied: " + "$ " +
                                Double.parseDouble (df.format (accounts.get
(i).accountBal)));
                    }
                } else {
                    System.out.println ("Customer details mismatch. Try again.");
                }
            }
        }
    }

    public static void four() {
        double transferAmount;
        String name1, name2;
        double account1 = 0;
        double account2 = 0;
        double account1Bal = 0;
        double account2Bal = 0;
        for (int i = 0; i < customerAccount.size (); i++) {
            System.out.println ("Enter customer's first name: ");
            name1 = sc.next ();
            System.out.println ("Enter customer's last name: ");
            name2 = sc.next ();
            if ((customerAccount.get (i).fName.equalsIgnoreCase (name1)) &&
(customerAccount.get (i).lName.equalsIgnoreCase (name2))) {
                System.out.println ("**********You can now transfer money from
account 1 to account 2 belonging to " + customerAccount.get (i).fName +
                        " " + customerAccount.get (i).lName + ".");
            } else {
                System.out.println ("Incorrect customer details. Please enter
customer name accurately to confirm the customer account for money transfer.");
                four ();
            }
        }
        for (int i = 0; i < accounts.size (); i++) {
            if (i == 0) {
                account1 = accounts.get (i).accountNo;
                account1Bal = accounts.get (i).accountBal;

            } else {
                account2 = accounts.get (i).accountNo;
                account2Bal = accounts.get (i).accountBal;
            }
        }

        if (!(account1 == account2)) {
            System.out.println ("Please enter the amount you want to transfer in
dollars from account 1 to account 2: ");
            transferAmount = sc.nextDouble ();
            account1Bal = account1Bal - transferAmount;
            account2Bal = account2Bal + transferAmount;
            if (account1Bal < 0) {
                System.out.println ("Error! First account will have a balance of
$0.00 after transfer. Transfer not possible.");
                System.exit (0);
            }
            if (account1Bal <= 10) {
                System.out.println ("Warning! First account has a balance less than
$10.");
            }
            if (account2Bal >= 100000) {
                System.out.println ("Warning! Second account has reached the highest
amount that is federally insured.");
            }
            System.out.println ("Account 1 balance after transfer: " +
Double.parseDouble (df.format (account1Bal)));
            System.out.println ("Account 2 balance after transfer: " +
Double.parseDouble (df.format (account2Bal)));
        } else {
            System.out.println ("Invalid account! Suggestion: Go to option 4 and view
bank account details of the customer for further clarification.");
            Options ();
        }
    }

    public static void five() {
        double forecastBal;
        for (int i = 0; i < accounts.size (); i++) {
            System.out.println ("Enter monthly automatic deposit amount for bank
account " + (i + 1) + ": ");
            accounts.get (i).autoDeposit = sc.nextDouble ();
            System.out.println ("Enter monthly automatic withdrawal amount for bank
account " + (i + 1) + ": ");
            accounts.get (i).autoWithdraw = sc.nextDouble ();
            forecastBal = (accounts.get (i).accountBal + accounts.get
(i).autoDeposit) - accounts.get (i).autoWithdraw;
            System.out.println ("****************************************ANNUAL
FORECAST OF ACCOUNT" + " " + (i + 1) + " BALANCE**********" +
                    "*********************************");
            System.out.println
("___________________________________________________________________________________
________________________________");
            System.out.println ("         MONTH          " + "         YEAR       " + "
MONTHLY STARTING BALANCE     " + "      MONTHLY ENDING BALANCE");
            System.out.println
("___________________________________________________________________________________
________________________________");
            System.out.println ("          Jan           " + "         2018       " + "
" +
                    Double.parseDouble (df.format (accounts.get (i).accountBal)) + "
" + Double.parseDouble (df.format (forecastBal)));
```

```
            double add = (accounts.get (i).autoDeposit) - (accounts.get
(i).autoWithdraw);
            System.out.print ("          Feb          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          Mar          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          Apr          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          May          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          Jun          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          Jul          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          Aug          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          Sept          " + "       2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          Oct          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          Nov          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.print ("          Dec          " + "        2018       " + "
" + Double.parseDouble (df.format (forecastBal)));
            forecastBal = forecastBal + add;
            System.out.println ("                        " + Double.parseDouble
(df.format (forecastBal)));
            System.out.println
("___________________________________________________________________________________
________________________________");
        }
    }

    public static void six() throws IOException {
        String fileName = "temp.txt";
        try {
            FileWriter fileWriter = new FileWriter ("temp.txt");
            BufferedWriter bufferedWriter = new BufferedWriter (fileWriter);
            for (int i = 0; i < customerAccount.size (); i++) {
                bufferedWriter.write ("******************************CUSTOMER " + (i
+ 1) + " DETAILS*************************************");
                bufferedWriter.newLine ();
                bufferedWriter.write ("Name: ");
                bufferedWriter.write (customerAccount.get (i).fName + " " +
customerAccount.get (i).lName);
                bufferedWriter.newLine ();
                bufferedWriter.write ("Username: " + customerAccount.get
(i).username);
                bufferedWriter.newLine ();
                bufferedWriter.write ("Password: " + customerAccount.get
(i).password);
                bufferedWriter.newLine ();
                for (i = 0; i < accounts.size (); i++) {
                    bufferedWriter.write ("Total no. of bank accounts: " +
accounts.size ());
                    bufferedWriter.newLine ();
                    for (i = 0; i < accounts.size (); i++) {
                        bufferedWriter.write ("Bank account " + (i + 1) + " interest
rate: " + accounts.get (i).interest);
                        bufferedWriter.newLine ();
                        bufferedWriter.write ("Bank account " + (i + 1) + " auto
deposit amount:  " + "$" + accounts.get (i).autoDeposit);
                        bufferedWriter.newLine ();
                        bufferedWriter.write ("Bank account " + (i + 1) + " interest
rate(%): " + accounts.get (i).autoWithdraw);
                        bufferedWriter.newLine ();
                        bufferedWriter.write ("Bank account " + (i + 1) + " balance
with interest rate applied: " + "$" + accounts.get (i).accountBal);
                        bufferedWriter.newLine ();
                    }
                }
                bufferedWriter.close ();
            }
        }
        catch (IOException ex) {
            System.out.println ("Error writing to file.");
        }
    }


    static class customerAccount {

        public Scanner sc = new Scanner (System.in);
        public String fName;
        public String lName;
        public String username;
        public String password;

        public ArrayList<BankAccount3.BankAccount> accounts;

        public customerAccount(Scanner sc, String fName, String lName, String
username, String password, ArrayList<BankAccount3.BankAccount> accounts) {
            this.sc = sc;
            this.fName = fName;
            this.lName = lName;
            this.username = username;
            this.password = password;
            this.accounts = accounts;

        }
        customerAccount() {
        }

    }

    static class BankAccount {
        int accountNo;
        double accountBal;
        double interest;
        double autoWithdraw;
        double autoDeposit;
        int year, month;

        public BankAccount(int accountNo, double accountBal, double interest, double
autoWithdraw, double autoDeposit, int year, int month) {
            this.accountNo = accountNo;
            this.accountBal = accountBal;
            this.interest = interest;
            this.autoWithdraw = autoWithdraw;
            this.autoDeposit = autoDeposit;
            this.month = month;
            this.year = year;
        }


        BankAccount() {
        }
    }
}
```

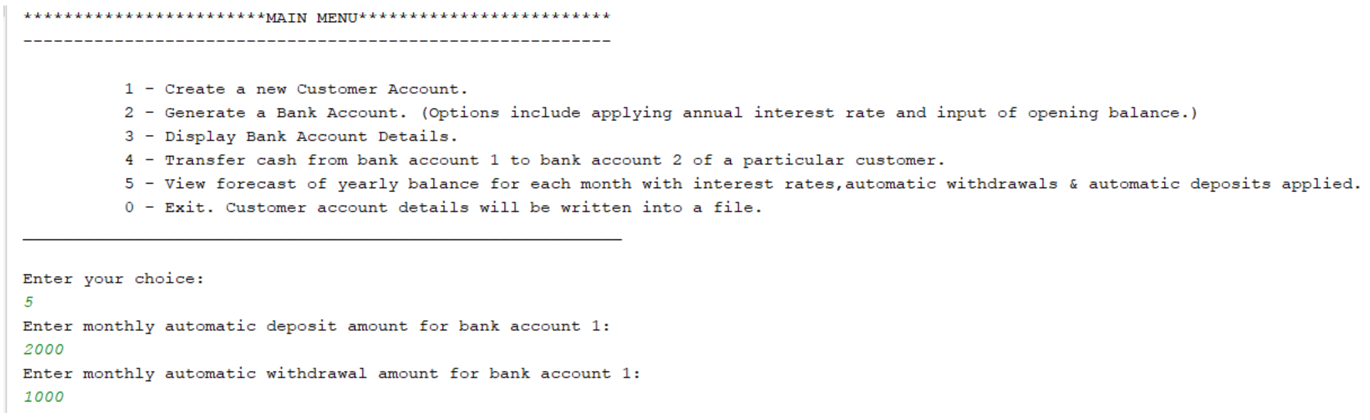
4. SCREENSHOTS

- Interest rates with validation and new balances



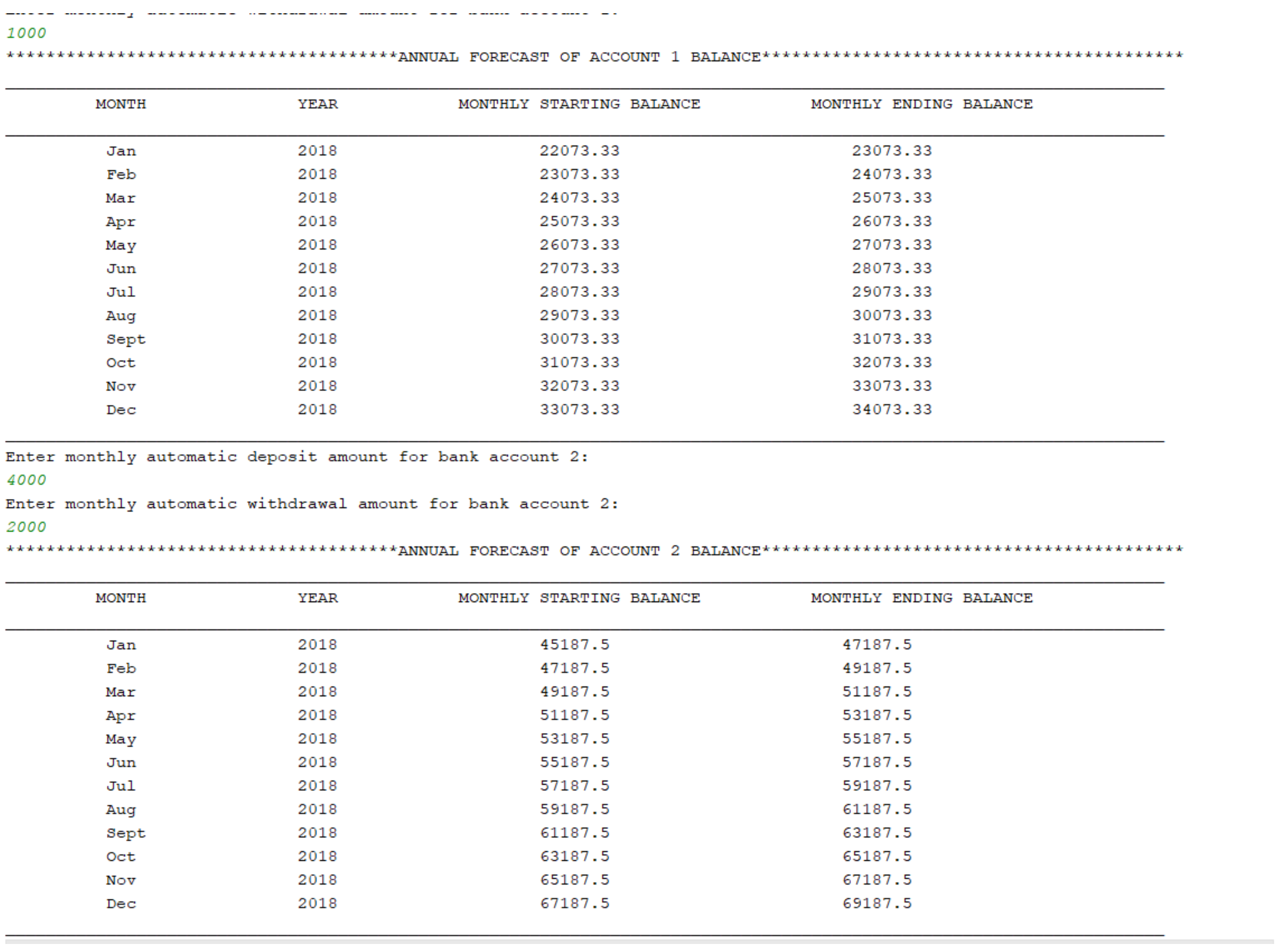
```
________________________________________________________________

Enter your choice:
2
***************Create a bank account.*****************
****If name matches with a name in the customer accounts list,
a bank account will be autogenerated for the particular customer.****
Enter customer's first name:
Dinithi
Enter customer's last name:
Jayasekara
***********Generate bank account for Dinithi Jayasekara***********
_______________________________________________________________________

Enter Dinithi Jayasekara's account number(4 digits only):
11117
Invalid Account Number! Please try again.
Account Number :
5
Invalid Account Number! Please try again.
Account Number :
2111
Enter Dinithi Jayasekara's account opening balance in dollars($):
4000000
Invalid Account Balance! Account balance has to be within the range of $ 0.00 to $100000.00
Account Opening Balance :
20000
********INTEREST RATES********
Enter annual interest rate for bank account:
0.001
Invalid annual interest rate. Please input a new value between the range of 0.01% to 15%
Please re-enter a suitable interest rate for bank account:
3
Account balance with annual interest rate applied: $20050.0
```

- Prompt user for automatic deposit and withdrawal



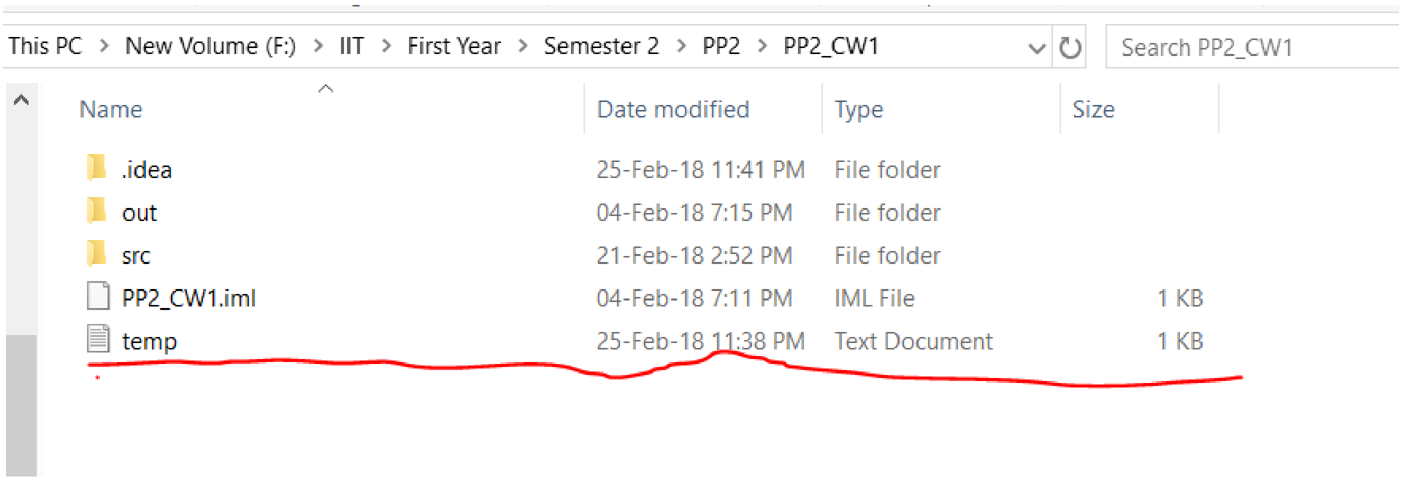
```
**********************MAIN MENU***********************
------------------------------------------------------

         1 - Create a new Customer Account.
         2 - Generate a Bank Account. (Options include applying annual interest rate and input of opening balance.)
         3 - Display Bank Account Details.
         4 - Transfer cash from bank account 1 to bank account 2 of a particular customer.
         5 - View forecast of yearly balance for each month with interest rates,automatic withdrawals & automatic deposits applied.
         0 - Exit. Customer account details will be written into a file.

_______________________________________________________

Enter your choice:
5
Enter monthly automatic deposit amount for bank account 1:
2000
Enter monthly automatic withdrawal amount for bank account 1:
1000
```

- Forecast annual balance for each account

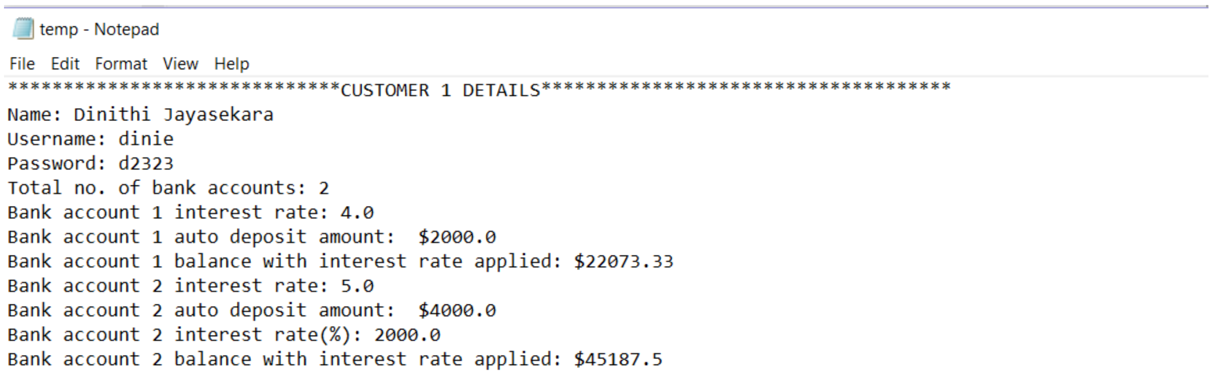


```
1000
**************************************ANNUAL FORECAST OF ACCOUNT 1 BALANCE*****************************************
_________________________________________________________________________________________________________
        MONTH              YEAR          MONTHLY STARTING BALANCE          MONTHLY ENDING BALANCE
_________________________________________________________________________________________________________
         Jan               2018                22073.33                        23073.33
         Feb               2018                23073.33                        24073.33
         Mar               2018                24073.33                        25073.33
         Apr               2018                25073.33                        26073.33
         May               2018                26073.33                        27073.33
         Jun               2018                27073.33                        28073.33
         Jul               2018                28073.33                        29073.33
         Aug               2018                29073.33                        30073.33
         Sept              2018                30073.33                        31073.33
         Oct               2018                31073.33                        32073.33
         Nov               2018                32073.33                        33073.33
         Dec               2018                33073.33                        34073.33
_________________________________________________________________________________________________________
Enter monthly automatic deposit amount for bank account 2:
4000
Enter monthly automatic withdrawal amount for bank account 2:
2000
**************************************ANNUAL FORECAST OF ACCOUNT 2 BALANCE*****************************************
_________________________________________________________________________________________________________
        MONTH              YEAR          MONTHLY STARTING BALANCE          MONTHLY ENDING BALANCE
_________________________________________________________________________________________________________
         Jan               2018                45187.5                        47187.5
         Feb               2018                47187.5                        49187.5
         Mar               2018                49187.5                        51187.5
         Apr               2018                51187.5                        53187.5
         May               2018                53187.5                        55187.5
         Jun               2018                55187.5                        57187.5
         Jul               2018                57187.5                        59187.5
         Aug               2018                59187.5                        61187.5
         Sept              2018                61187.5                        63187.5
         Oct               2018                63187.5                        65187.5
         Nov               2018                65187.5                        67187.5
         Dec               2018                67187.5                        69187.5
_________________________________________________________________________________________________________
```

- Write customer details to text file


This PC > New Volume (F:) > IIT > First Year > Semester 2 > PP2 > PP2_CW1

Name	Date modified	Type	Size
.idea	25-Feb-18 11:41 PM	File folder	
out	04-Feb-18 7:15 PM	File folder	
src	21-Feb-18 2:52 PM	File folder	
PP2_CW1.iml	04-Feb-18 7:11 PM	IML File	1 KB
temp	25-Feb-18 11:38 PM	Text Document	1 KB



```
****************************CUSTOMER 1 DETAILS*************************************
Name: Dinithi Jayasekara
Username: dinie
Password: d2323
Total no. of bank accounts: 2
Bank account 1 interest rate: 4.0
Bank account 1 auto deposit amount:  $2000.0
Bank account 1 balance with interest rate applied: $22073.33
Bank account 2 interest rate: 5.0
Bank account 2 auto deposit amount:  $4000.0
Bank account 2 interest rate(%): 2000.0
Bank account 2 balance with interest rate applied: $45187.5
```

5. CONCLUSION

All the required functions work efficiently as possible. Input data is stored in a text file and arrays.