**INTERNET BANKING**

**A MINI PROJECT**

**OBJECT ORIENTED PROGRAMMING USING JAVA**

**(CS1304)**

**III SEMESTER**



Submitted By:

SAGARIKA CHOUDHARY

Registration No.:

189303026

Roll No.:

58

Submitted To:

Dr. ANKIT SRIVASTAVA

INTRODUCTION

1.1 INTERNET BANKING

Banks have traditionally been in the forefront of harnessing technology to improve their products, services and efficiency. They have, over a long time, been using electronic and telecommunication networks for delivering a wide range of value added products and services. The delivery channels include direct dial – up connections, private networks, public networks etc and the devices include telephone, Personal Computers including the Automated Teller Machines, etc. With the popularity of PCs, easy access to Internet and World Wide Web (WWW), Internet is increasingly used by banks as a channel for receiving instructions and delivering their products and services to their customers. This form of banking is generally referred to as Internet Banking, although the range of products and services offered by different banks vary widely both in their content and sophistication.

**1.2 WHAT IS JAVA**

JAVA is a high-level programming language which evolved from the earlier versions like C++.The Java programming language consists out of a Java compiler, the Java virtual machine, and the Java class libraries. The Java virtual machine (JVM) is a software implementation of a computer that executes programs like a real machine. The Java compiler translates Java coding into so-called byte-code. The Java virtual machine interprets this byte-code and runs the program. The Java virtual machine is written specifically for a specific operating system. The Java runtime environment (JRE) consists of the JVM and the Java class libraries. The target of Java is to write a program once and then run this program on multiple operating systems.

Java has the following properties:

* Platform independent
* Object-orientated programming language
* Strongly-typed programming language
* Interpreted and compiled language
* Automatic memory management.

**CURRENT WORK**

The project has been successfully completed and executes smoothly. Concepts being used are:

IMPORTING PACKAGES

EXCEPTION HANDLING

ABSTRACTION

STATIC FINAL KEYWORD

ABSTRACT METHODS

STRING HANDLING

METHOD OVERRIDING

FILE HANDLING (WRITING INTO AN EXISTING FILE)

WRAPPER CLASS

INHERITANCE

POLYMORPHISM(OVERRIDING)

FILE HANDLING (CREATE A NEW FILE)

EXCEPTION HANDLING

MENU-DRIVING

OBJECY CREATION

**REQUIREMENTS**

**SOFTWARE:**

1) Windows platform.

2) Command Prompt

3) Notepad installed in Windows

4) Microsoft Word installed on Windows

**OVERVIEW OF THE PROJECT**

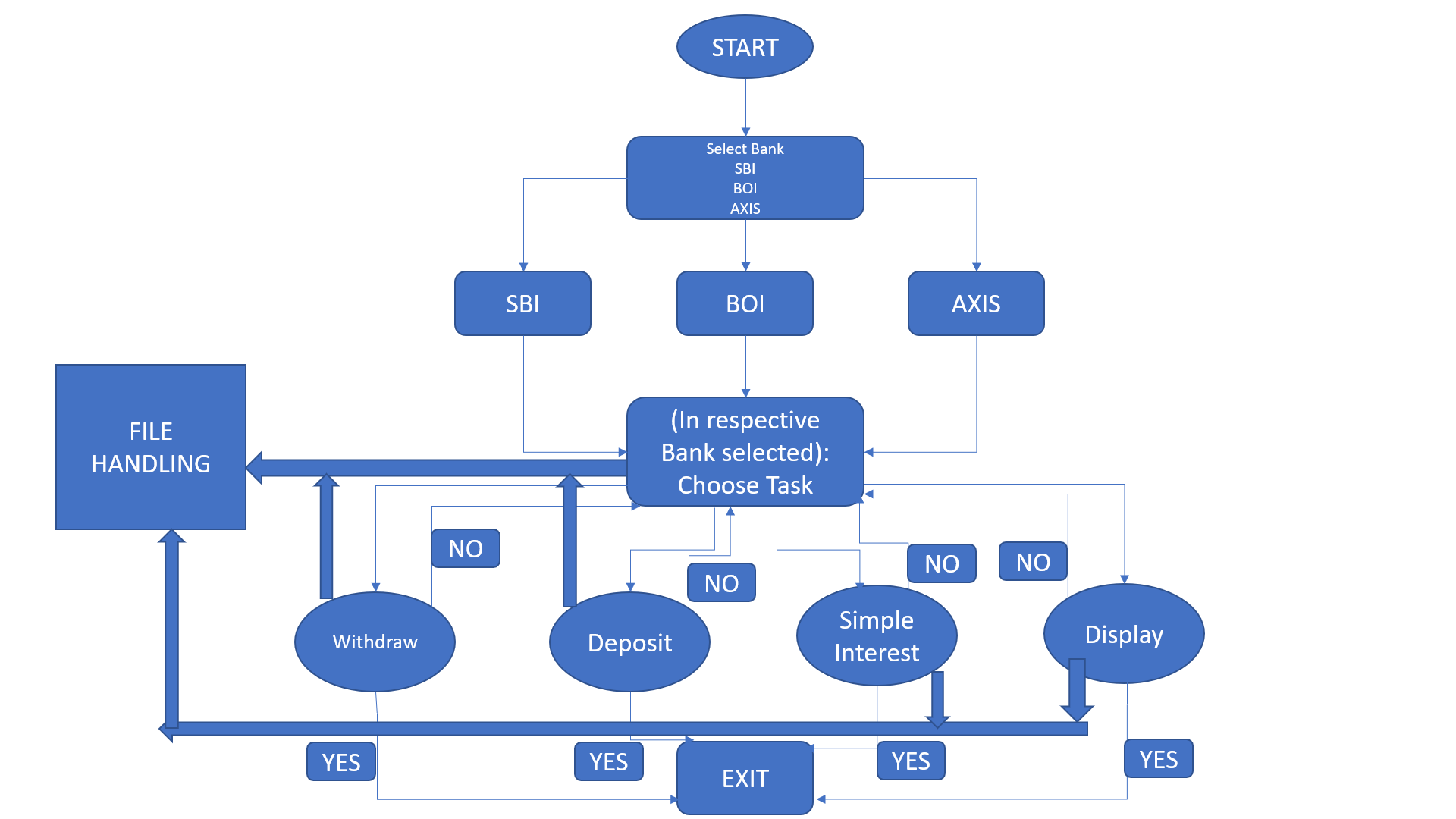
**System Description:**

The proposed system is used to maintain an account record of the customer of a Bank by storing entries for customer, investments and showing their account summary.

The project has the following parts:

1. The user will logged in into his account.
2. He will be asked to select his Bank preference.
3. After this, he will be asked to choose from a range of actions to be performed: withdrawing money, depositing money, displaying simple interest for the amount of money present in the account currently, displaying the details, etc.
4. The transaction details keep getting stored into a .txt file.

**FLOWCHART**



# CODE

===========================================================================

import java.util.Scanner; /\*IMPORTING PACKAGES\*/

import java.io.\*;

class xxx

{

public static void appendStrToFile(String fileName,String str)

{

try /\*EXCEPTION HANDLING\*/

{

BufferedWriter bw = new BufferedWriter(new FileWriter(fileName,true));

PrintWriter out = new PrintWriter(bw);

out.println(str);

out.close();

}catch(IOException e){System.out.println("Exception occured : "+e);}

}

}

abstract class Bank extends xxx /\*ABSTRACTION\*/

{

static final float ROI=7; /\*STATIC FINAL KEYWORD\*/

//String file = D:/mybank.txt;

double cust\_bal=200000;

abstract public void withdraw(int amt);

abstract public void deposit(int amt);

abstract public void SI();

abstract public void display(); /\*ABSTRACT METHODS\*/

}

abstract class Intermediate extends Bank

{

public void deposit(int amt)

{

System.out.println("Processing request");

cust\_bal= cust\_bal+amt;

System.out.println("Request processed"+"\n Remaining balance in account:"+cust\_bal);

String s1 = Double.toString(cust\_bal); /\*STRING HANDLING\*/

s1 = "DEPOSITED. Remaining balance in account:"+ s1;

appendStrToFile("mybank.txt",s1); /\*METHOD OVERRIDING\*/

}

public void SI()

{

double si;

int t=10;

si= (cust\_bal\*ROI\*t)/100;

System.out.println("The interest generated by your current balance is:"+si);

/\*FILE HANDLING(WRITING INTO AN EXISTING FILE)\*/

String s1 = Double.toString(si); /\*WRAPPER CLASS\*/

s1 = "SIMPLE INTEREST is :"+ s1;

appendStrToFile("mybank.txt",s1);

}

abstract public void display();

abstract public void withdraw(int amt);

}

class SBI extends Intermediate /\*INHERITANCE\*/

{

public void withdraw(int amt)

{

if(amt>cust\_bal)

System.out.println("Insufficient funds");

else

{

System.out.println("Processing request");

if(amt<50000)

{cust\_bal=cust\_bal-amt;

String s1 = Double.toString(cust\_bal);

s1 = "WITHDRAWN. Remaining balance in account:"+ s1;

appendStrToFile("mybank.txt",s1);}

else

{

cust\_bal=(cust\_bal-amt-amt\*0.01);

System.out.println("Request processed"+"\n Remaining balance in account:"+cust\_bal);

String s1 = Double.toString(cust\_bal);

s1 = "WITHDRAWN. Remaining balance in account:"+ s1;

appendStrToFile("mybank.txt",s1);

}

}

}

public void display()

{

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("The current customer details are:");

System.out.println("Name:Sagarika \n Bank name:SBI");

System.out.println("Balance in account:"+cust\_bal);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

class BOI extends Intermediate

{

public void withdraw(int amt) /\*POLYMORPHISM(OVERRIDING)\*/

{

if(amt>cust\_bal)

System.out.println("Insufficient funds");

else

{

System.out.println("Processing request");

if(amt<50000)

cust\_bal=cust\_bal-amt;

else

{

cust\_bal=cust\_bal-amt-(amt\*0.02);

System.out.println("Request processed"+"\n Remaining balance in account:"+cust\_bal);

}

}

}

public void display()

{

System.out.println("The current customer details are:");

System.out.println("Name:Sagarika \n Bank name:BOI");

System.out.println("Balance in account:"+cust\_bal);

}

}

class AXIS extends Intermediate

{

public void withdraw(int amt)

{

if(amt>cust\_bal)

System.out.println("Insufficient funds");

else

{

System.out.println("Processing request");

if(amt<50000)

cust\_bal=cust\_bal-amt;

else

{

cust\_bal=cust\_bal-amt-amt\*0.03;

System.out.println("Request processed"+"\n Remaining balance in account:"+cust\_bal);

}

}

}

public void display()

{

System.out.println("The current customer details are:");

System.out.println("Name:Sagarika \n Bank name:AXIS");

System.out.println("Balance in account:"+cust\_bal);

}

}

class DemoBank

{

public static void main(String args[])throws Exception

{

int option,choice,amt;

System.out.println("Which bank do you wish to transact with: 1)SBI 2)BOI 3)AXIS");

System.out.println("Enter your option:");

Scanner sc= new Scanner(System.in);

option=sc.nextInt();

BufferedWriter bw = null;

File file = new File("D:/mybank.txt");

/\*FILE HANDLING(CREATE A NEW FILE)\*/

if (!file.exists()) {

file.createNewFile();

}

try{ /\*EXCEPTION HANDLING\*/

String mycontent = "CUSTOMER NAME : SAGARIKA CHOUDHARY ";

FileWriter fw = new FileWriter(file);

bw = new BufferedWriter(fw);

PrintWriter pw = new PrintWriter(bw);

pw.println(mycontent);

pw.close();

if(option==1)

{

String ss1="\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO SBI BANK\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

pw.println(ss1);

pw.close();

}

else if(option==2)

{

String ss2="\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO BOI BANK\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

pw.println(ss2);

pw.close();

}

else if(option==3)

{

String ss3="\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO AXIS BANK\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

pw.println(ss3);

pw.close();

}

}

catch (IOException ioe) {ioe.printStackTrace();}

switch(option) /\*MENU-DRIVING\*/

{

case 1:

{

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO SBI BANK\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

SBI S=new SBI();

int n=5;

do{

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("1. Withdraw \n2.deposit \n3.display simple interest \n4.Display details \n5.No pending work");

System.out.println("Enter what is your option:");

choice=sc.nextInt();

n--;

switch(choice)

{

case 1:

System.out.println("Enter amount to withdraw");

amt=sc.nextInt();

S.withdraw(amt);

break;

case 2:

System.out.println("Enter amount to deposit");

amt=sc.nextInt();

S.deposit(amt);

break;

case 3:

S.SI();

case 4:

S.display();

break;

case 5:

System.out.println("Thank you");

break;

default:

System.out.println("Invalid choice");

}}while(n>0);

}

while(choice!=5);

}break;

case 2:

{

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO BOI BANK\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

BOI B=new BOI();

int n=5;

do{

System.out.println("1. Withdraw \n2.deposit \n3.display simple interest \n4.Display details \n5.No pending work");

System.out.println("Enter what is your option:");

choice=sc.nextInt();

n--;

switch(choice)

{

case 1:

System.out.println("Enter amount to withdraw");

amt=sc.nextInt();

B.withdraw(amt);

break;

case 2:

System.out.println("Enter amount to deposit");

amt=sc.nextInt();

B.deposit(amt);

break;

case 3:

B.SI();

case 4:

B.display();

break;

case 5:

System.out.println("Thank you");

break;

default:

System.out.println("Invalid choice");

}}while(n>0);

}

while(choice!=5);

}break;

case 3:

{

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO AXIS BANK\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

AXIS A=new AXIS();

int n=5;

do{

System.out.println("1. Withdraw \n2.deposit \n3.display simple interest \n4.Display details \n5.No pending work");

System.out.println("Enter what is your option:");

choice=sc.nextInt();

n--;

switch(choice)

{

case 1:

System.out.println("Enter amount to withdraw");

amt=sc.nextInt();

A.withdraw(amt);

break;

case 2:

System.out.println("Enter amount to deposit");

amt=sc.nextInt();

A.deposit(amt);

break;

case 3:

A.SI();

case 4:

A.display();

break;

case 5:

System.out.println("Thank you");

break;

default:

System.out.println("Invalid choice");

}}while(n>0);

}

while(choice!=5);

}break;

}

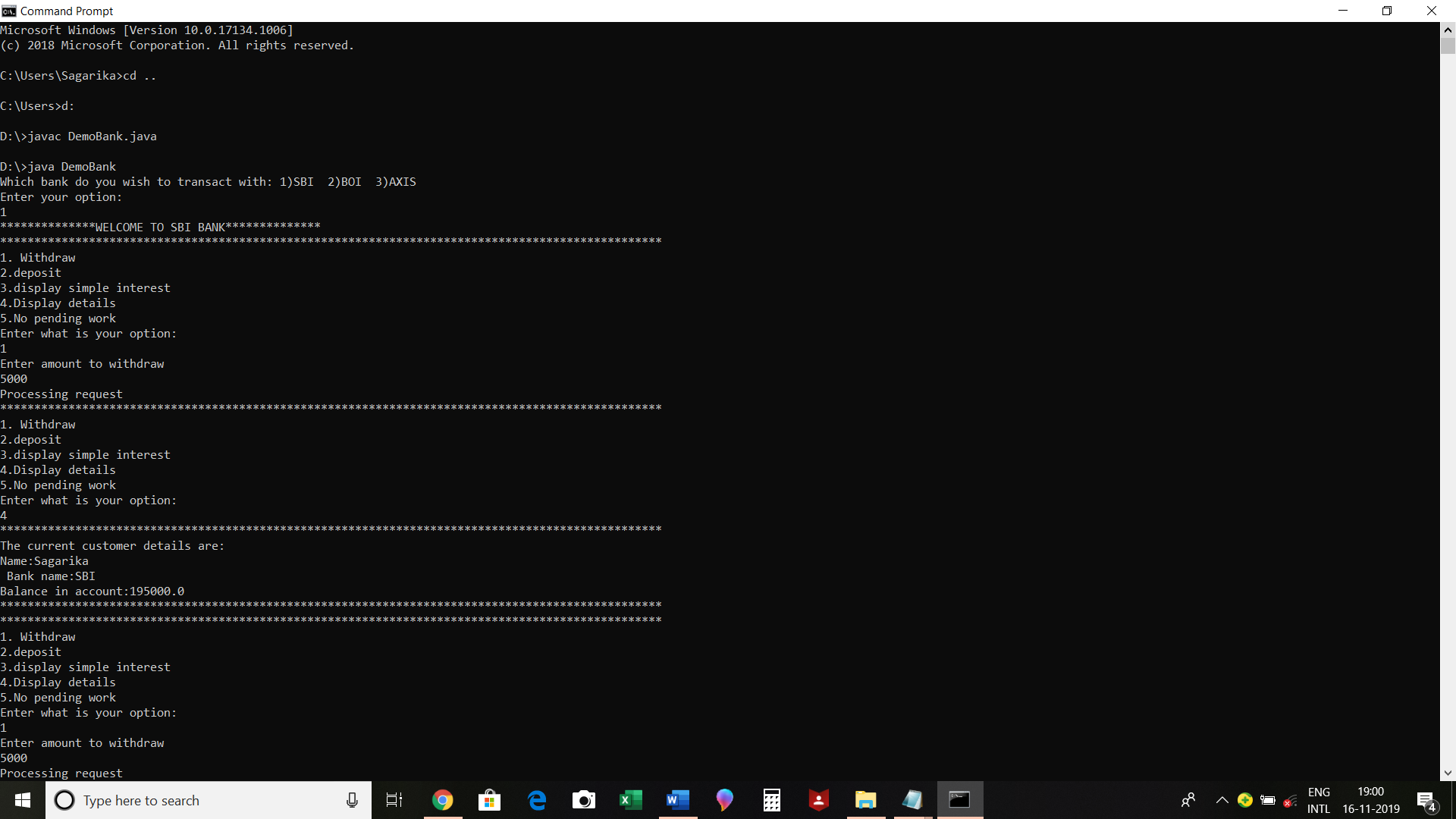
System.out.println("All Transaction over");

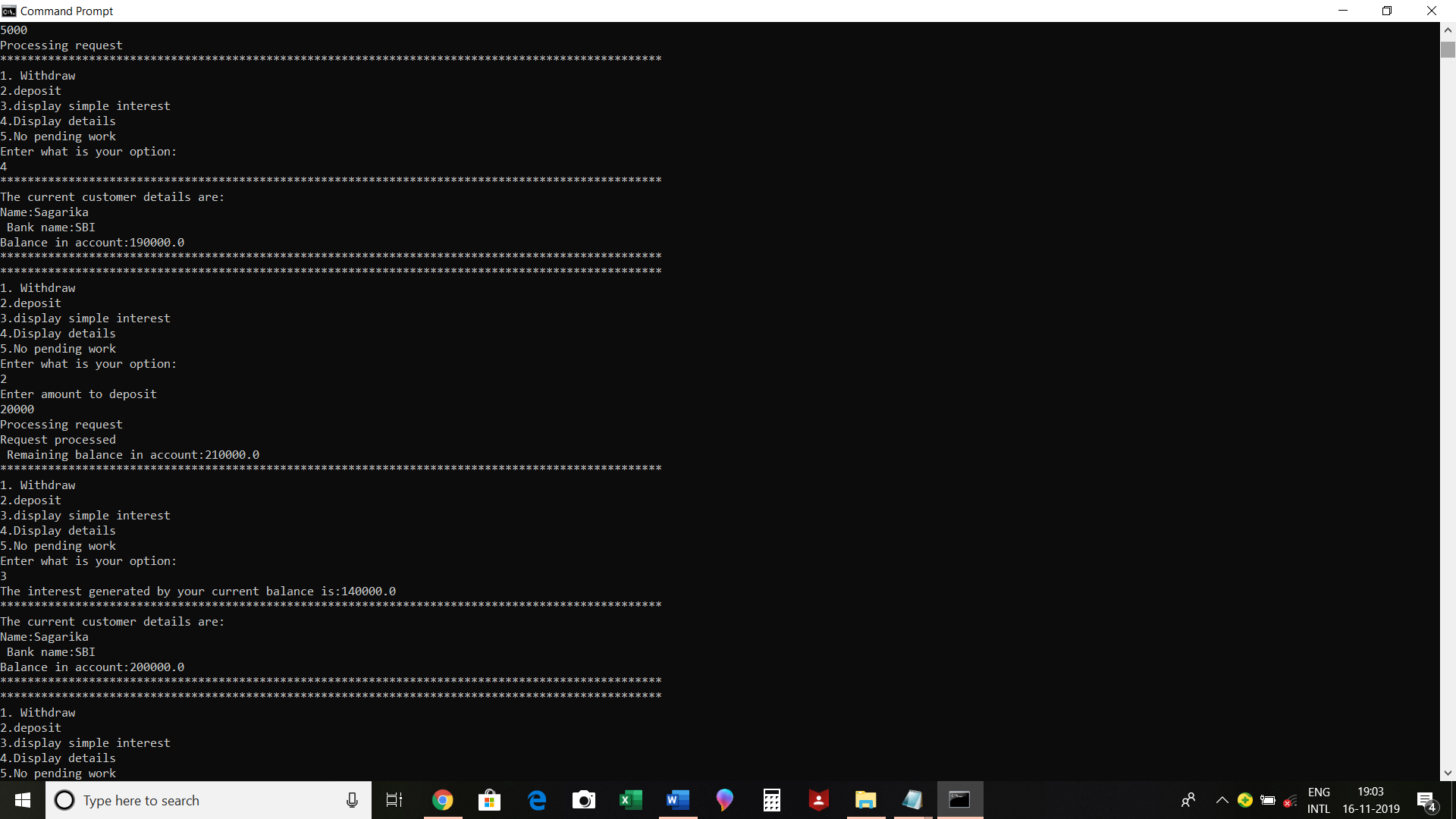
}

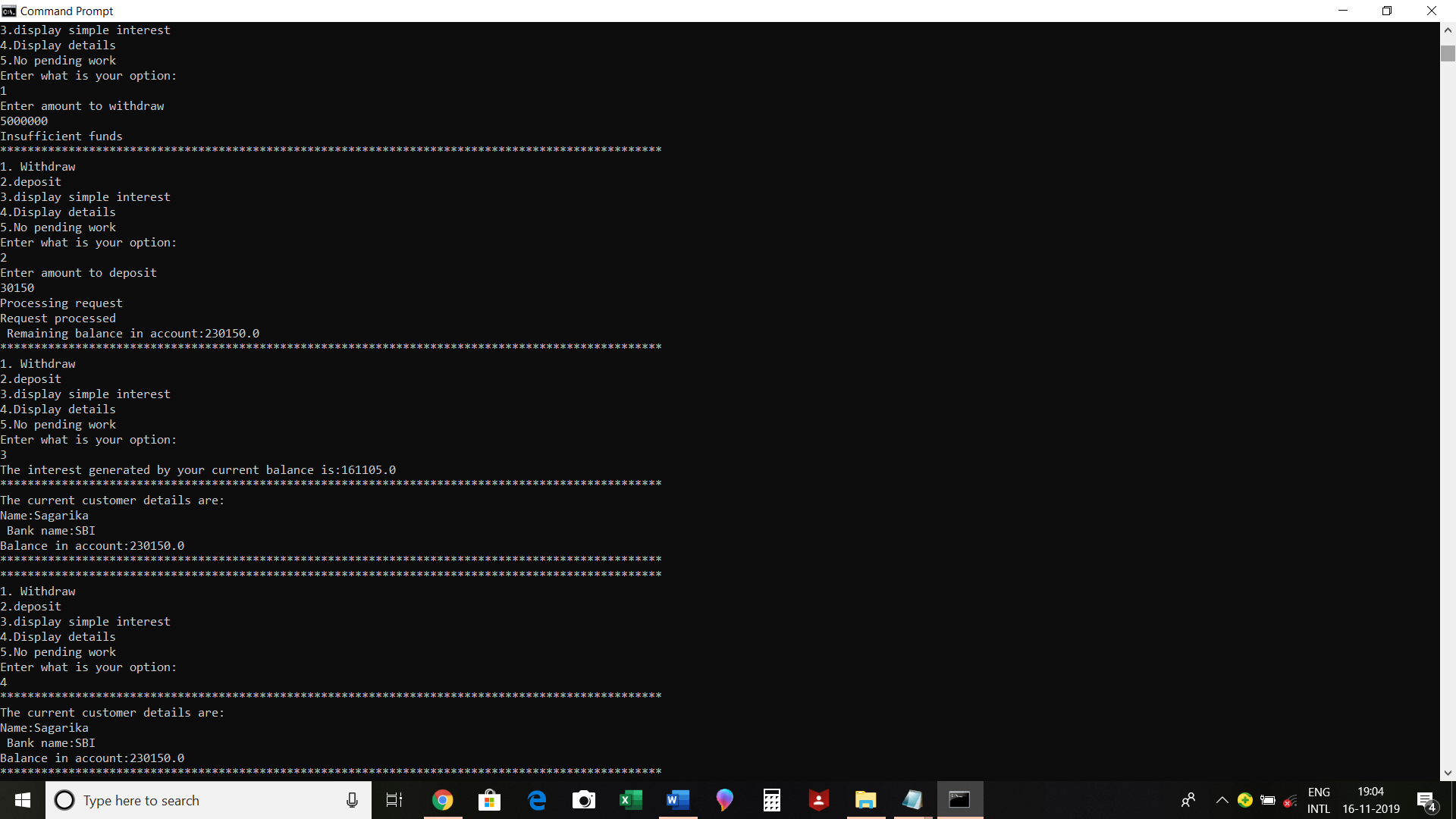
}

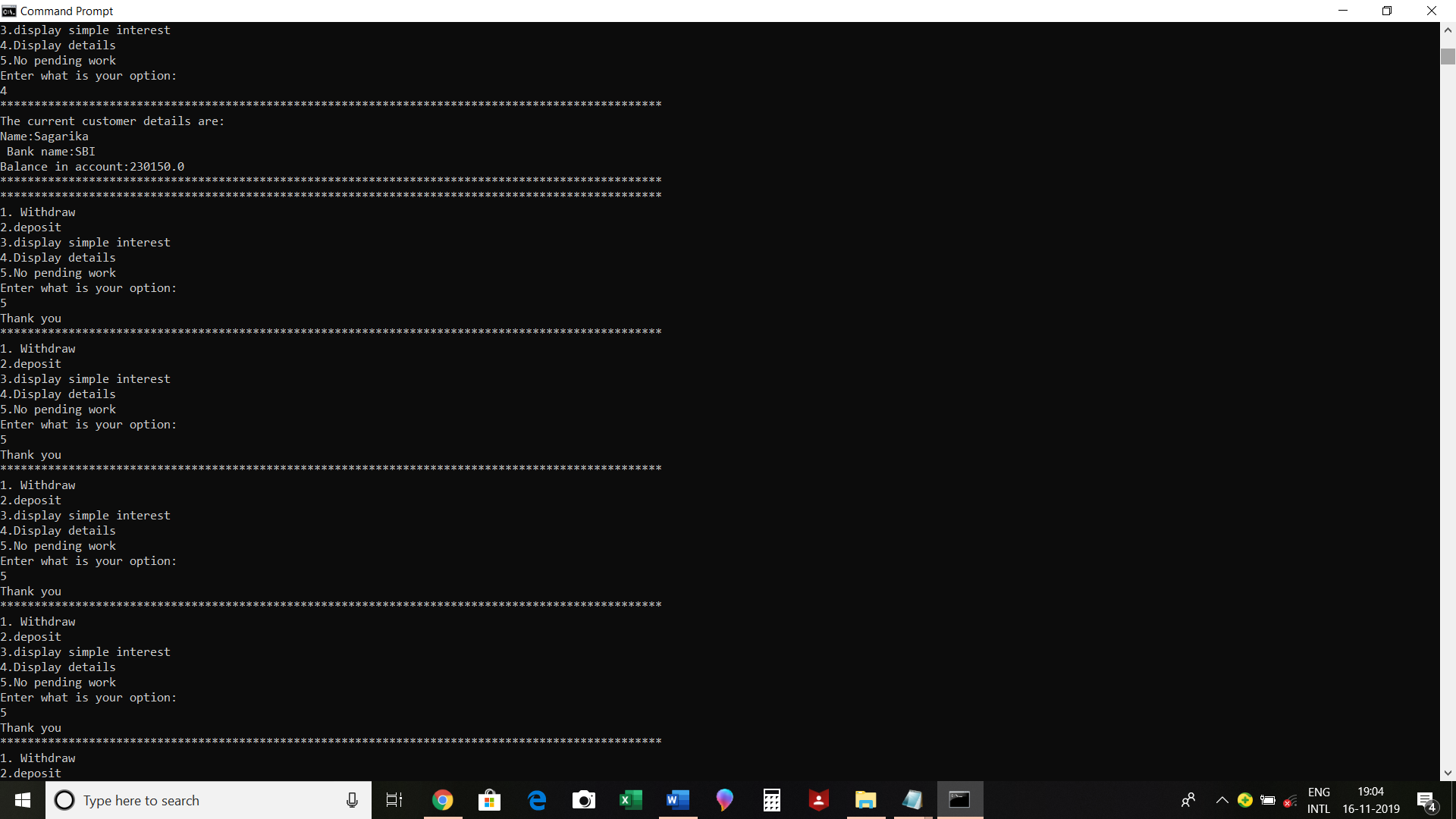
===========================================================================

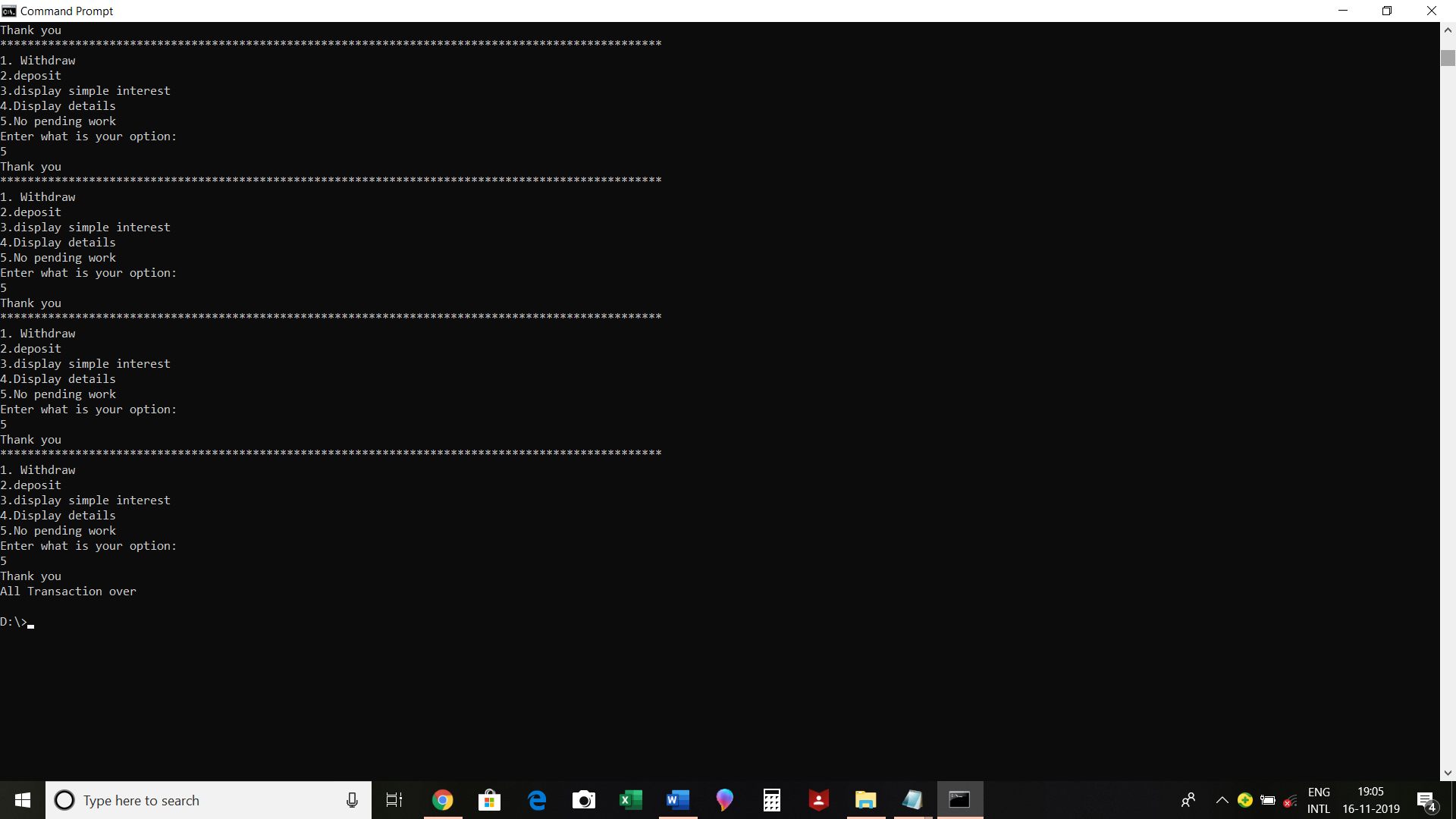
**OUTPUT**

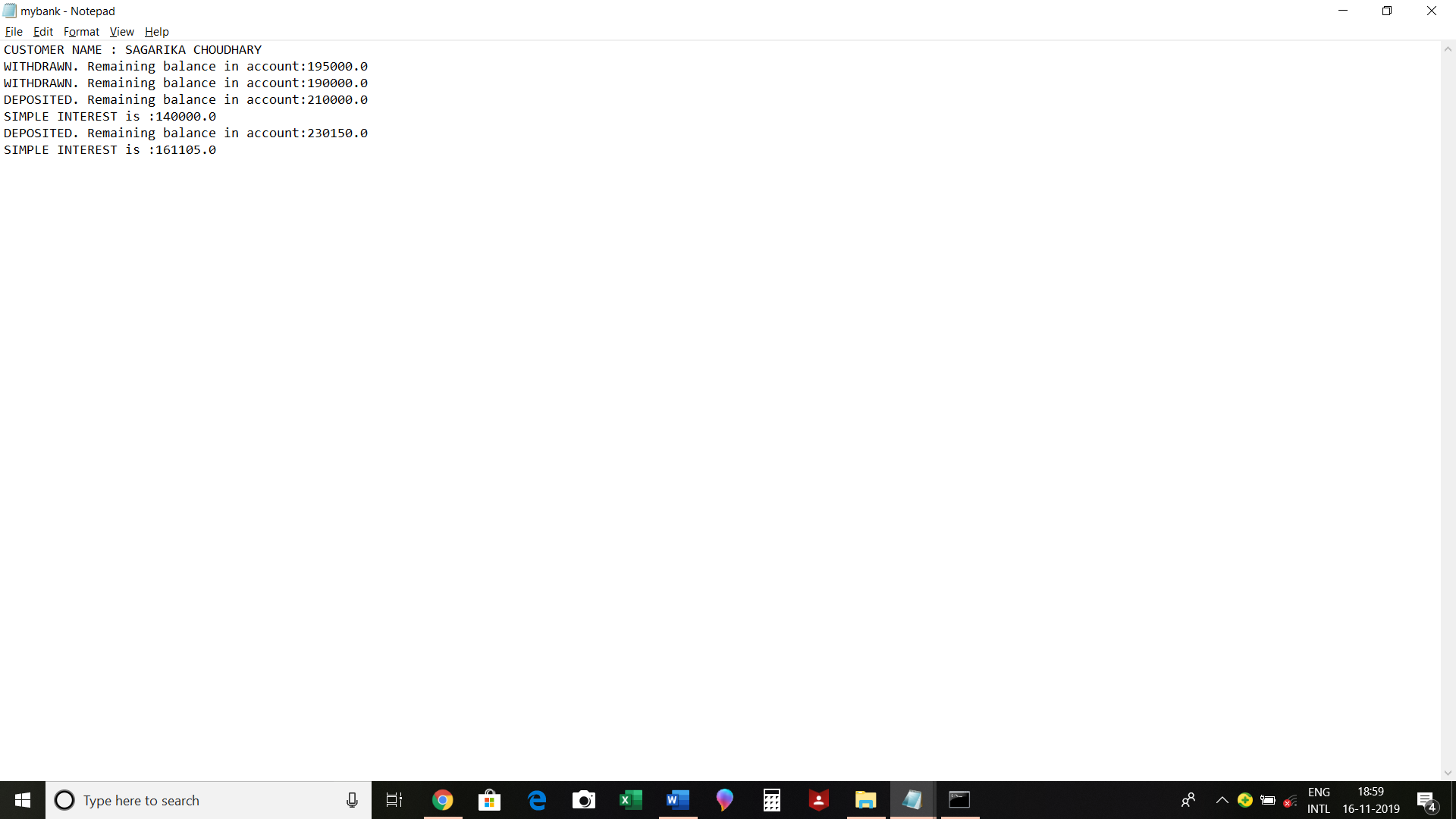












# SCOPE OF ENHANCEMENT

The project can be made for multi-user purpose by the usage of multithreading. With implementation of multithreading and multi-users being coming into enforcement, security of the account holders could be increased as user login credentials could be implemented.

**CONCLUSION**

An INTERNET BANKING SYSYTEM has been developed and the system was tested with sample data.

The system results in regular timely preparations of required outputs. The system provides a user-friendly environment for the customers to do banking without going to the bank itself, allows the facility like withdrawing, depositing, etc.