A PROJECT REPORT ON

SIMPLE BANKING APPLICATION

By

PALAK GUPTA

Batch-EON (2021-5736)

Under the Guidance of,

AMRUTA DEORE

(Technical Trainer)

EduBridge India Pvt. Ltd.

INDEX

ABSTRACT

Now-a-days, time is treated as money. No one is willing to spend their half a day's time at bank for transactions like balance enquiry, money transfer etc. Complete banking system which is implemented on Java provides complete solution for banking related transactions and you can perform your most important banking chores from the comfort of your home.

Existing System:

In the existing system all the transactions are carried out at manually. This is a time consuming process with lots of paper work involved. Moreover, security is also a major concern.

Proposed System:

Complete Banking system enables bank users to access their accounts and general information about bank's products and services from the comfort of your PC. This application aims at secure banking system which will be accessible to any authenticated user

with valid user id and password. Through this system, user can access his account from anywhere.

CHAPTER 1
INTRODUCTION

1.1 .INTRODUCTION AND OBJECTIVE:

The ‘SIMPLE BANKING APPLICATION’ Interface is targeted to the future banking solution for the users who have multiple bank accounts in different banks. This interface integrates all existing banks and provides business solutions for both retail and corporate. System Involves

• This interface integrates all existing banks and provides business solutions for both retailers and corporate.

• This system acts as a standard interface between the clients and the banks

• Users who have accounts in various banks can login here and can make any kind of transactions.

• In the backend, system will take care of the entire obligation required in order to carry on transaction smoothly.

1.2. PURPOSE OF THE PROJECT

Its purpose is to create a common portal for multiple banks. So users can login here andcan. Access any of the available banks and can do required transactions .

1.3. EXISTING SYSTEM & DISADVANTAGES

Currently we are having lot of banks in the market and any person can do transactions of any individual bank either manually or in online. But no one can do all banks transactions in a single portal or in single bank. This is the main disadvantage in existing system to avoid this problem we are introducing “SIMPLE BANKING APPLICATION”.

1.4. PROPOSED SYSTEM & ITS ADVANTAGES

The Multi Banking System Interface is targeted to the future banking solution for the userswho is having multiple bank accounts in multiple banks. This interface integrates allexisting banks and provides business solutions for both retail and corporate.This system acts as a standard interface between the clients and all the banks, By usingthis portal any client who maintain accounts in various banks can directly log on to MultiBanking System Interface and make any kind of transactions. In the backend, system willtake care of the entire obligation required in order to carry on transaction smoothly.

CHAPTER 2

REQUIREMENTS SPECIFICATION

2.1 FUNCTIONAL REQUIREMENTS SPECIFICATION

- **Login Module**
- **Open account**
- **Balance Enquiry**
- **Account History**
- **Withdraw Account**
- **Check Rate Of Intrest**
- **Exit.**

Module Description:

1. **Login Module**

Users can log in with user id and password

2. Open account

This is for new users. They can add an account. For this they need to fill a form at their bank and the bank provides with the user id and password to authenticate themselves.

3. Balance Enquiry

This module facilitates balance enquiry.

4. Account History

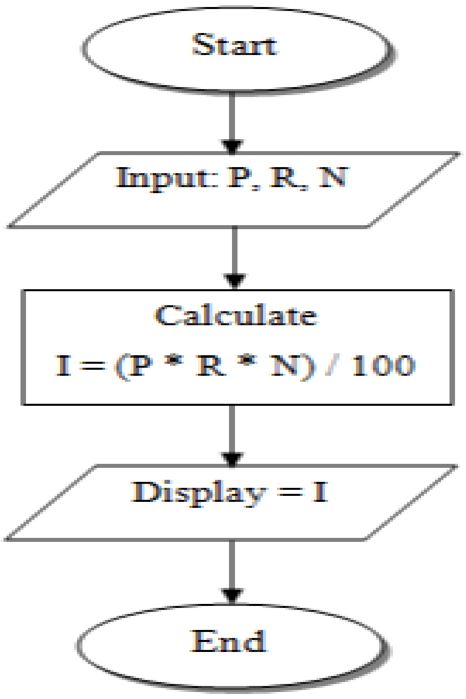
User can check the account history. Account history includes account details, transaction details etc. It simulates mini transaction operation but slightly different as it gives all details.

5. Withdraw Account

Withdraw money from the bank in person: You can also withdraw money by going into a branch and talking to a bank teller. Most of the time, just like an ATM, you'll need the card associated with the account you wish to draw from, as the teller will run the card, and also request that you enter your PIN, to access funds.

6. Calculate Rate of Intrest

The term of the deposit ranges from as less as 7 days to more than 10 years. For Senior Citizens (individuals who are aged 60 years and above), the bank offers 0.50% per annum additional rate of interest over and above the existing interest rates for the general public on a minimum deposit of Rs 5000 for term deposits.



7. Exit

Exit from Application.

Software Requirements

- Operating System : Windows10.
- Programming Language : Java
- IDE's : Eclipse(Oxygen)

Hardware Requirements

- RAM : 4.00 GB
- Processor: Intel CORE i3
- Hard disk: Minimum 10 GB

INTRODUCTION TO JAVA

About Java:

Initially the language was called as “oak” but it was renamed as “java” in 1995.The primary motivation of this language was the need for a platform-independent(i.e. architecture neutral)language that could be used to create software to be embedded in variousconsumer electronic devices.

Java is a programmer’s language Java is cohesive and consistent Except for those constraint imposed by the Internet environment. Java gives the programmer, full control.

Finally Java is to Internet Programming where c was to System Programming

Importance of Java to the Internet:

Java has had a profound effect on the Internet. This is because; java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer.

They are passive information and Dynamic active programs. in the areas of Security and probability. But Java addresses these concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

Applications and applets

An application is a program that runs on our Computer underthe operating system of that computer. It is more or less like one creating using C or C++.Java's ability to create Applets makes it important. An Applet I san application, designed tobe transmitted over the Internet and executed by a Java-compatible web browser. An applet I actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be react to the user input and dynamically change.

Java Architecture:

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

<u>Compilation of code:</u>

When you compile the code, the Java compiler creates machine code (called byte code)for a hypothetical machine called Java Virtual Machine(JVM). The JVM is supposed t executed the byte code. The JVM is created for the overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines This machine is called Java Virtual Machine .

<u>Compiling and interpreting java source code:</u>

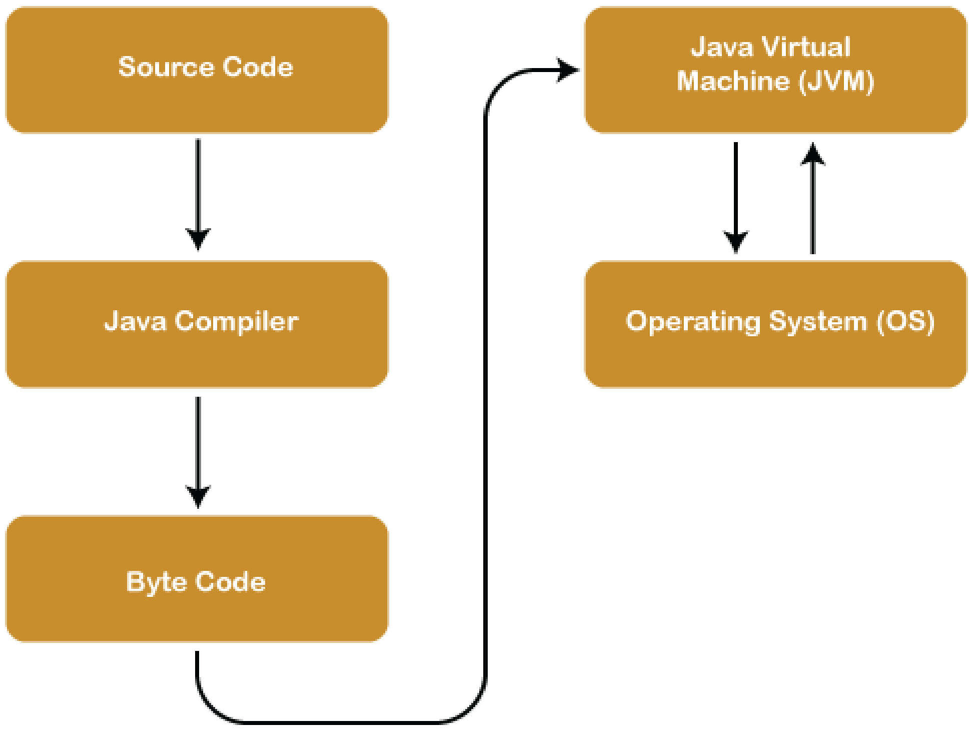
Java Architecture is a collection of components, i.e., **JVM, JRE,** and **JDK**. **It** integrates the process of interpretation and compilation. It defines all the processes involved in creating a

Java program. **Java Architecture** explains each and every step of how a program is compiled and executed.

Java Architecture can be explained by using the following steps:

- There is a process of compilation and interpretation in Java.
- Java compiler converts the Java code into byte code.
- After that, the JVM converts the byte code into machine code.
- The machine code is then executed by the machine.

The following figure represents the **Java Architecture** in which each step is elaborate graphically.



Components of Java Architecture

The Java architecture includes the three main components:

- Java Virtual Machine (JVM)
- Java Runtime Environment (JRE)
- Java Development Kit (JDK)

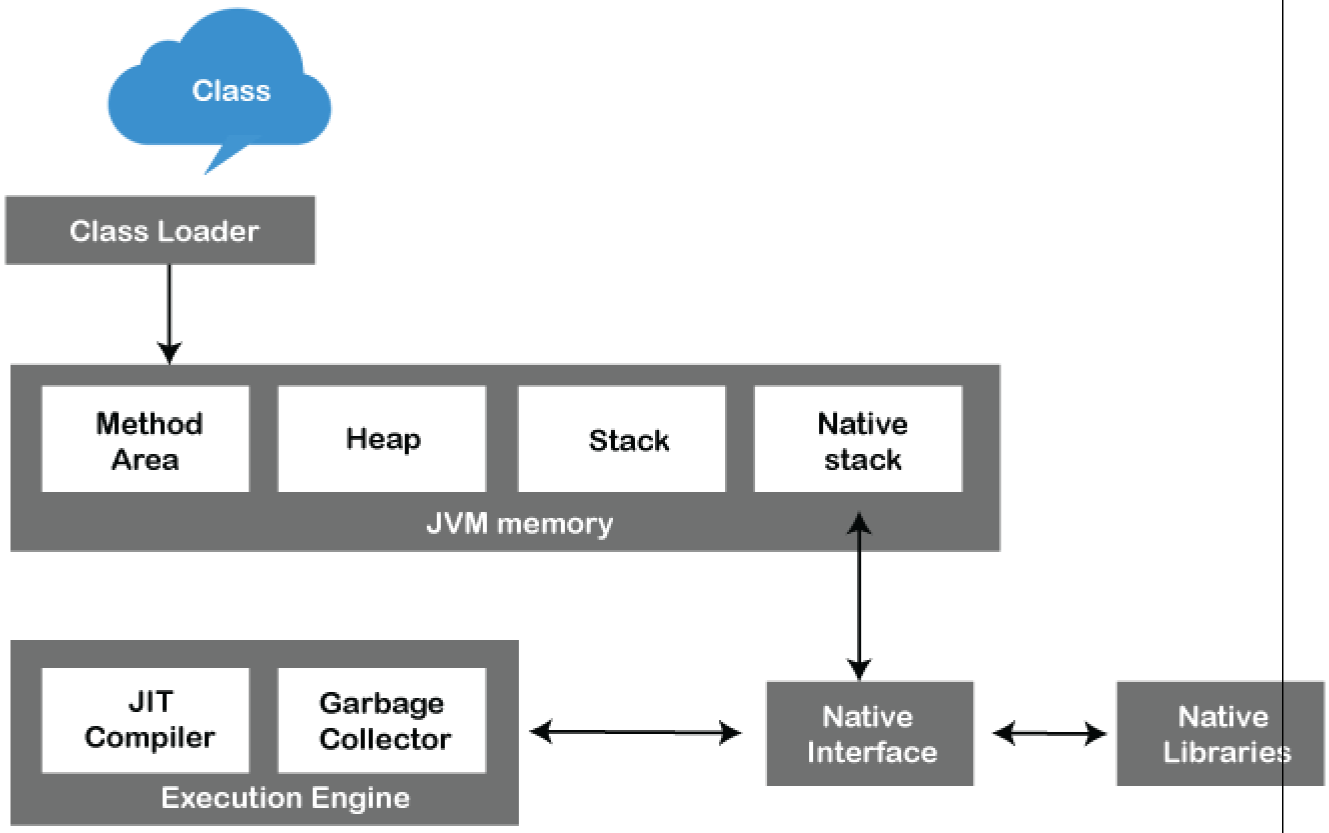
Java Virtual Machine

The main feature of Java is **WORA**. WORA stands for **Write Once Run Anywhere**. The feature states that we can write our code once and use it anywhere or on any operating system. Our Java program can run any of the platforms only because of the Java Virtual Machine. It is a Java platform component that gives us an environment to execute java programs. JVM's main task is to convert byte code into machine code.

JVM, first of all, loads the code into memory and verifies it. After that, it executes the code and provides a runtime environment. Java Virtual Machine (JVM) has its own architecture, which is given below:

JVM Architecture

JVM is an abstract machine that provides the environment in which Java bytecode is executed. The falling figure represents the architecture of the JVM.



ClassLoader: ClassLoader is a subsystem used to load class files. ClassLoader first loads the Java code whenever we run it.

Class Method Area: In the memory, there is an area where the class data is stored during the code's execution. Class method area holds the information of static variables, static methods, static blocks, and instance methods.

Heap: The heap area is a part of the JVM memory and is created when the JVM starts up. Its size cannot be static because it increase or decrease during the application runs.

Stack: It is also referred to as thread stack. It is created for a single execution thread. The thread uses this area to store the elements like the partial result, local variable, data used for calling method and returns etc.

Native Stack: It contains the information of all the native methods used in our application.

Execution Engine: It is the central part of the JVM. Its main task is to execute the byte code and execute the Java classes. The execution engine has three main components used for executing Java classes.

- **Interpreter:** It converts the byte code into native code and executes. It sequentially executes the code. The interpreter interprets continuously and even the same method multiple times. This reduces the performance of the system, and to solve this, the JIT compiler is introduced.
- **JIT Compiler:** JIT compiler is introduced to remove the drawback of the interpreter. It increases the speed of execution and improves performance.
- **Garbage Collector:** The garbage collector is used to manage the memory, and it is a program written in Java. It works in two phases, i.e., **Mark** and **Sweep**. Mark is an area where the garbage collector identifies the used and unused chunks of memory. The Sweep removes the identified object from the **Mark**

Java Native Interface

Java Native Interface works as a mediator between Java method calls and native libraries.

Java Runtime Environment

It provides an environment in which Java programs are executed. JRE takes our Java code, integrates it with the required libraries, and then starts the JVM to execute it. To learn more about the Java Runtime Environment.

Java Development Kit

It is a software development environment used in the development of Java applications and applets. Java Development Kit holds JRE, a compiler, an interpreter or loader, and several development tools in it. To learn more about the Java Development Kit.

Features of Java:

Simple:Java was designed to be easy for the Professional programmer to learn and to useeffectively. If you are an experienced C++ Programmer. Learning Java will orientedfeatures of C++ . Most of the confusing concepts from C++ are either left out of Java orimplemented in a cleaner, more approachable manner. In Java there are a small number ofclearly defined ways to accomplish a given task.

Object oriented : Java was not designed to be source-code compatible with any other language. Thisallowed the Java team the freedom to design with a blank state. One outcome of this was aclean usable, pragmatic approach to objects. The object model in Java is simple and easyto extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust : The multi-platform environment of the web places extraordinary demands on a program,because the program must execute reliably in a variety of systems. The ability to createrobust programs. Was given a high priority in the design of Java. Java is strictly typedlanguage; it checks your code at compile time and runtime.Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

WHAT IS ECLIPSE:

In the context of computing, Eclipse is an integrated development environment (IDE) for developing applications using the Java programming language and other programming languages such as C/C++, Python, PERL, Ruby etc.

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins. Developed using Java, the Eclipse platform can be used to develop rich client applications, integrated development environments and other tools. Eclipse can be used as an IDE for any programming language for which a plug-in is available.

The Java Development Tools (JDT) project provides a plug-in that allows Eclipse to be used as a Java IDE, PyDev is a plugin that allows Eclipse to be used as a Python IDE, C/C++ Development Tools (CDT) is a plug-in that allows Eclipse to be used for developing application using C/C++, the Eclipse Scala plug-in allows Eclipse to be used an IDE to develop Scala applications and PHP eclipse is a plug-in to eclipse that provides complete development tool for PHP.

CHAPTER 2

SYSTEM DESIGN

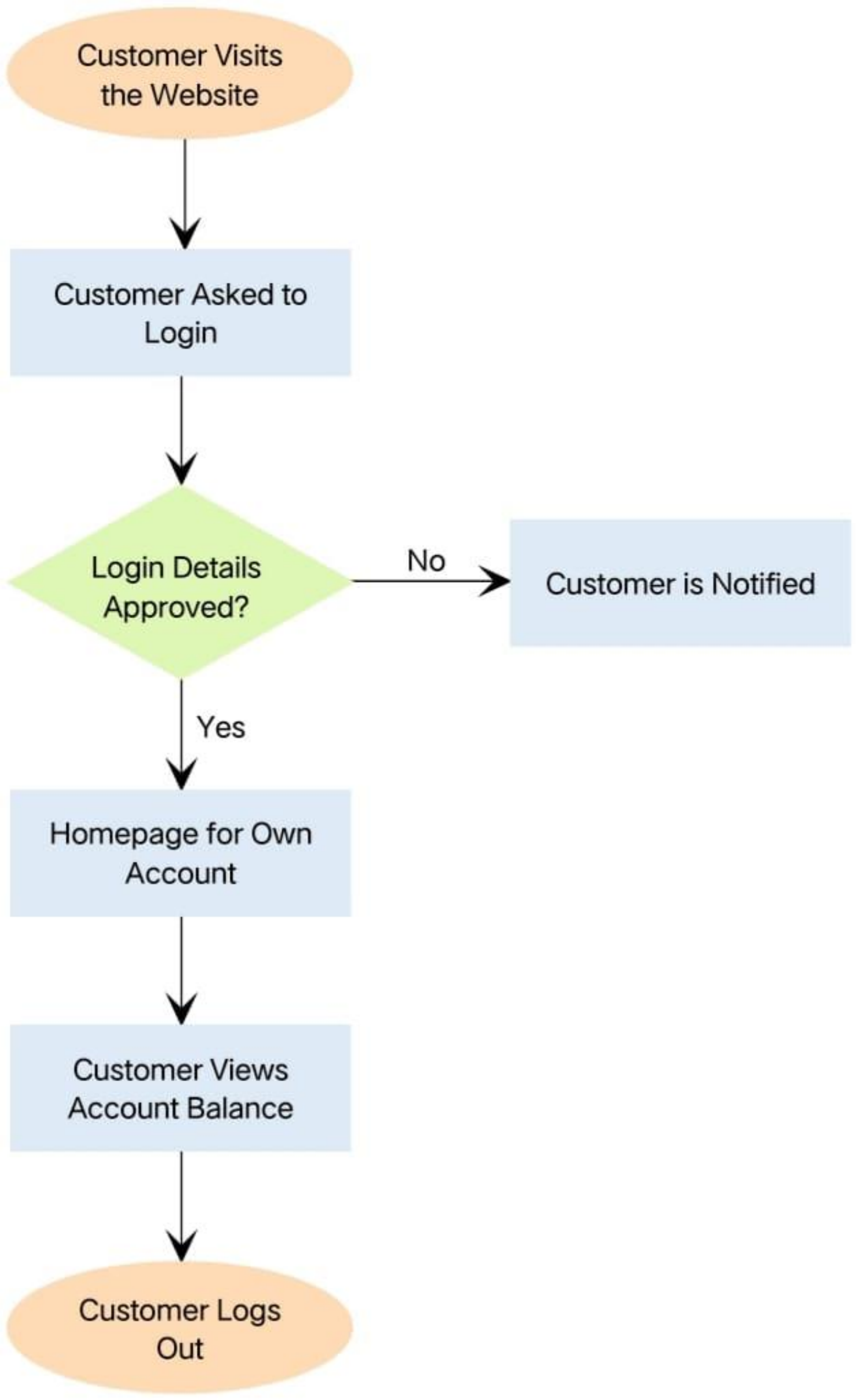
3.1 INTRODUCTION

Systems design

It is the process or art of defining the architecture ,components , modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering…

FLOW CHART:

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence. They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes. If we consider all the various forms of flowcharts, they are one of the most common diagrams on the planet, used by both technical and non-technical people in numerous fields.



DATA FLOW DIAGRAMS:

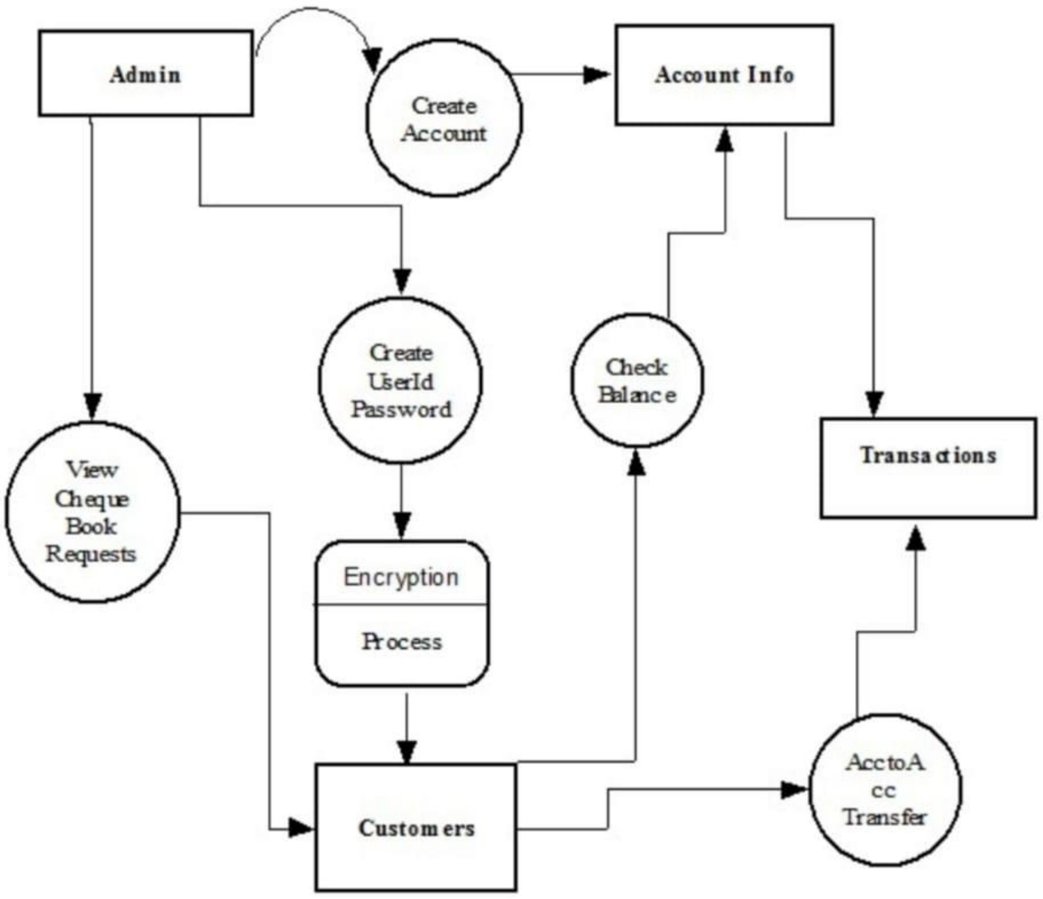
Data flow diagram will act as a graphical representation of the system in terms ofinteraction between the system, external entities, and process and how data stored in certainlocation.

•External entities

•Data stores

• Process

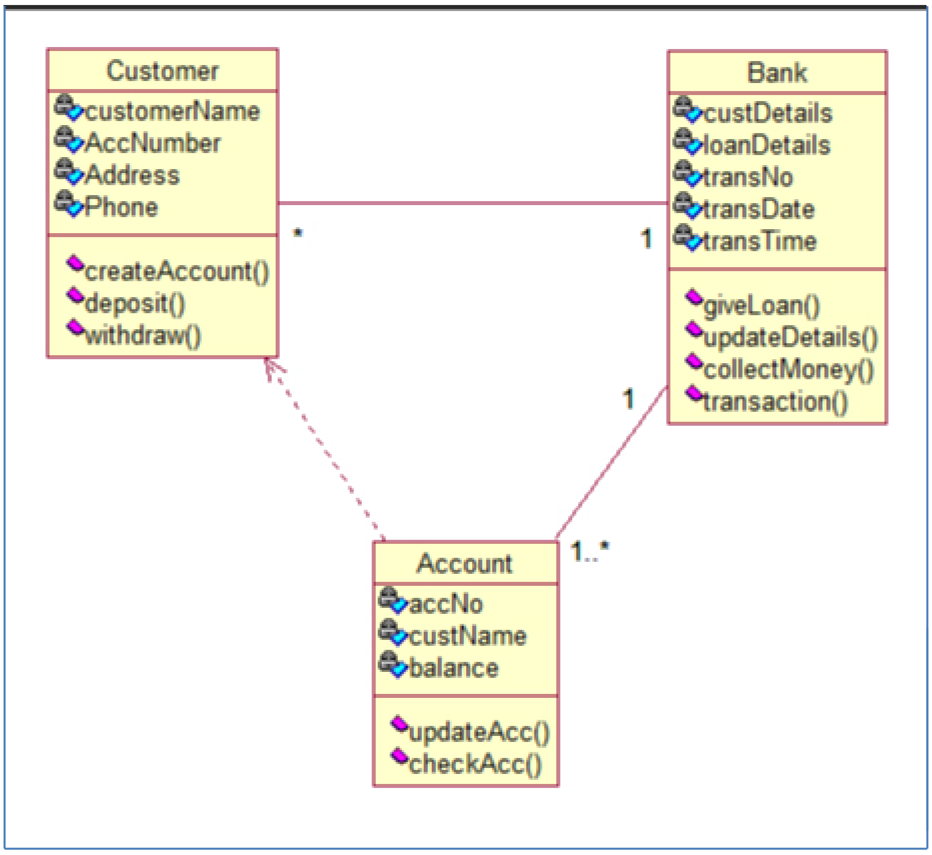
• Data Flow



UML DIAGRAMS

Unified Modeling Language:

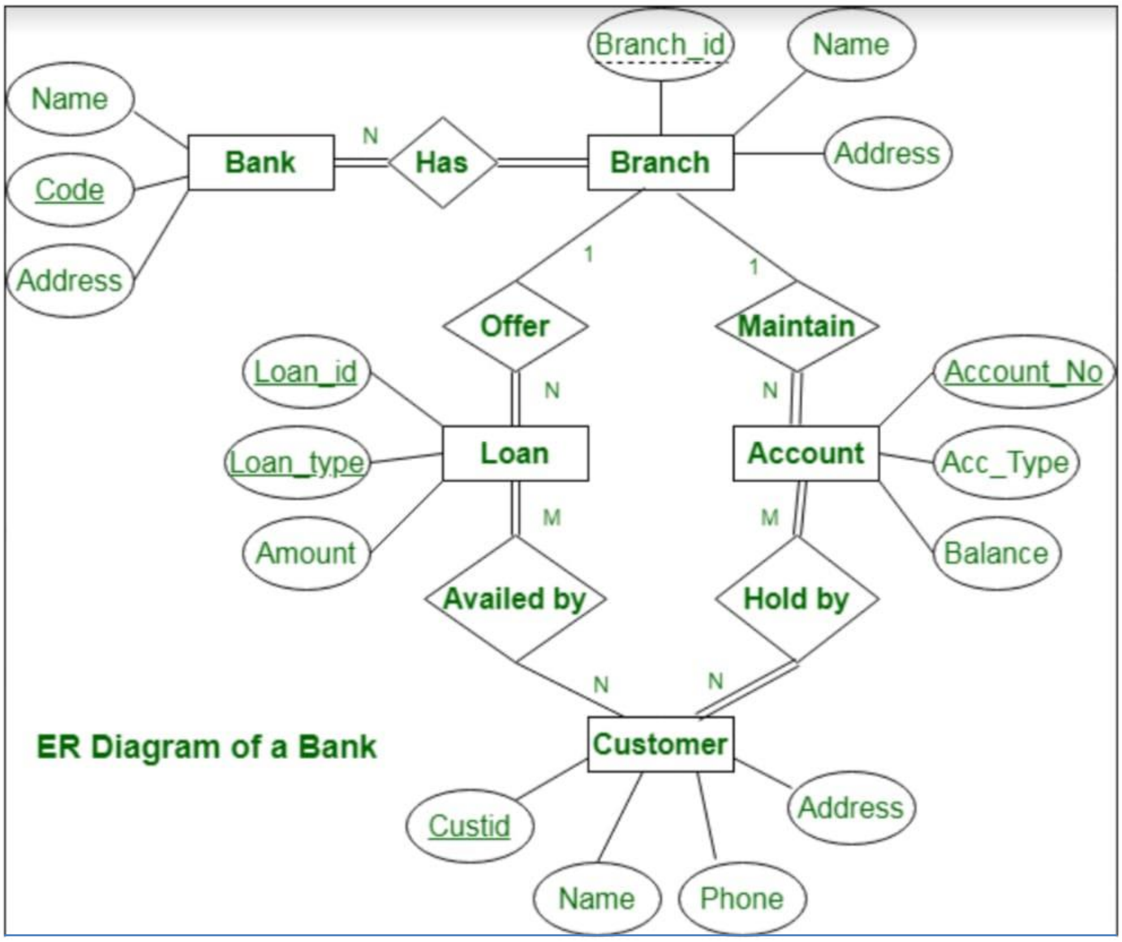
The Unified Modeling Language allows the software engineer to express an analysis modelusing the modeling notation that is governed by a set of syntactic semantic and pragmaticrules



ER Diagram:

- The ER model is a high-level data model diagram

- ER diagrams are a visual tool which is helpful to represent the ER model
- Entity relationship diagram displays the relationships of entity set stored in a database
- ER diagrams help you to define terms related to entity relationship modeling
- ER model is based on three basic concepts: Entities, Attributes & Relationships.



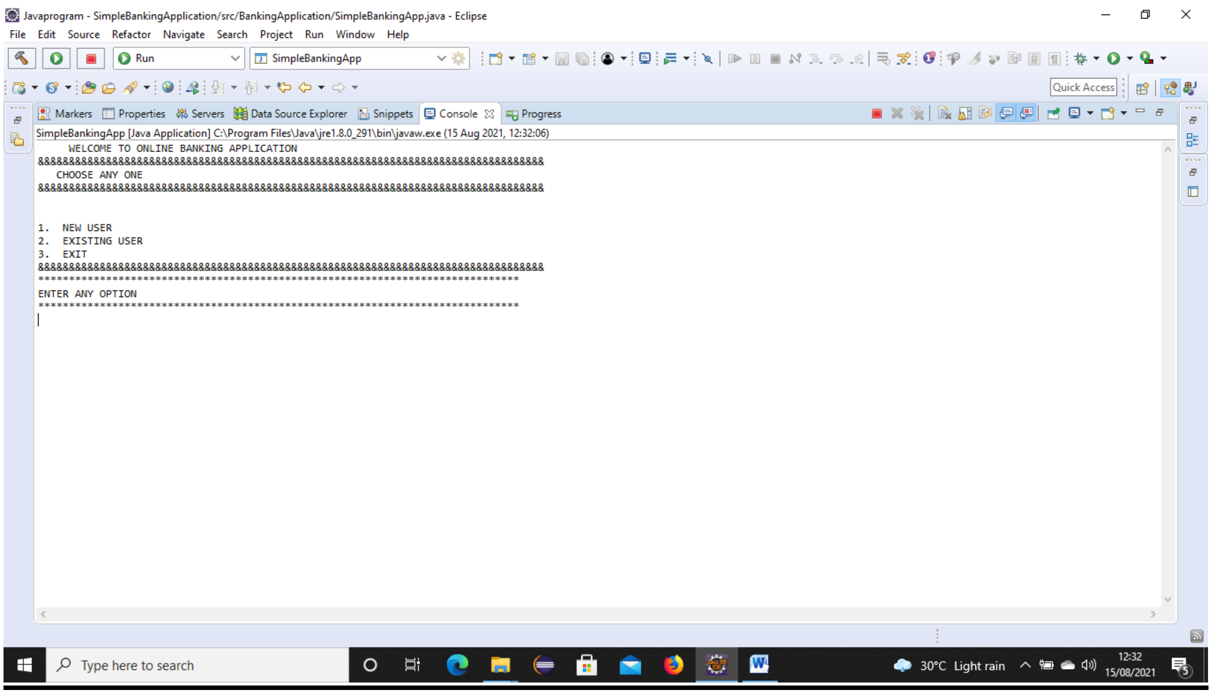
CHAPTER 4

OUTPUT SCREEN AND CODING

4.1 .LOGIN PAGE :

```
void show()
          {
                    int choice=0;
                    Scanner sc = new Scanner(System.in);

          System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&");
                    System.out.println("   CHOOSE ANY ONE   ");

          System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&");
                    System.out.println(" \n ");
                    System.out.println("1.  NEW USER");
                    System.out.println("2.  EXISTING USER");
                    System.out.println("3.  EXIT");

          System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&");
                    do
                    {

          System.out.println("***********************************************************************************");
                              System.out.println("ENTER ANY OPTION ");

          System.out.println("***********************************************************************************");
                              choice = sc.nextInt();
                              System.out.println(" \n ");
                              switch(choice)
                              {
                              case 1:

          System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&");
                                        NewUser();

          System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&");
                              case 2:

          System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&");
                                        existing();

System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&");
                              case 3:

          System.out.println("£££££££££££££££££££££££££££££££££££££££££££££££££");
                              break;
                              default:
```

```java
                        System.out.println("SORRY!!!! ");
                        System.out.println("INVALID OPTION!!!!! ");
                        System.out.println("PLEASE ENTER AGAIN... ");
                        break;
                        }
                }
                        while(choice!= 3);
                System.out.println(" THANK YOU FOR VISITING....");
                Show_Menu();

        }
```

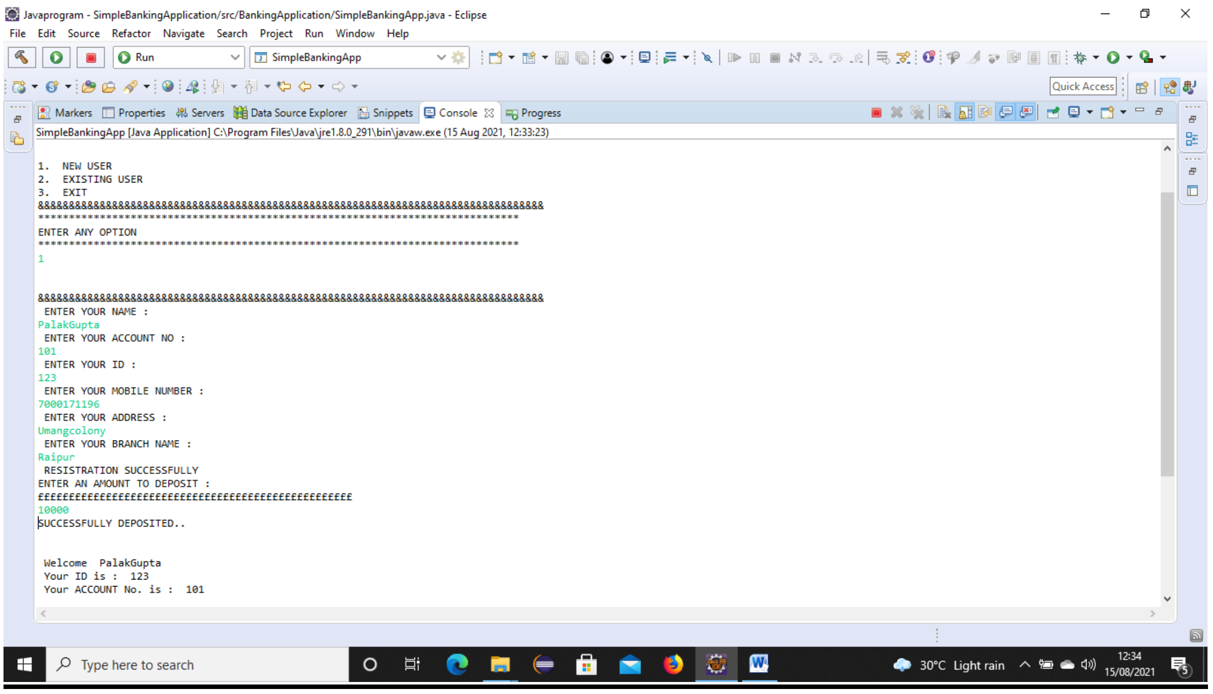


Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Run
SimpleBankingApp
Quick Access
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Progress
Type here to search
30°C Light rain
12:32
15/08/2021

```
SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 12:32:06)
     WELCOME TO ONLINE BANKING APPLICATION
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
   CHOOSE ANY ONE
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&


1.  NEW USER
2.  EXISTING USER
3.  EXIT
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
******************************************************************************
ENTER ANY OPTION
******************************************************************************
|
```

4.2 . NEW USER LOGIN :

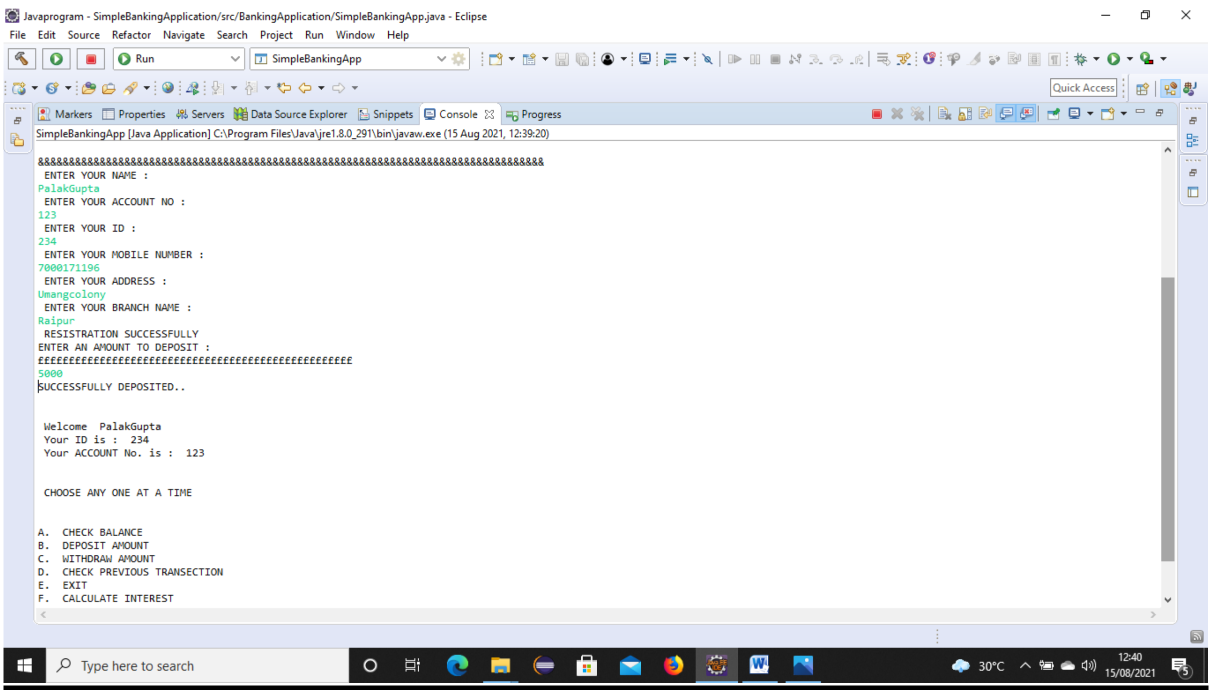
```
void NewUser()
        {
                Scanner sc = new Scanner(System.in);

                long Mobile_No;
                String Address;
                String Branch_Name;
                System.out.println(" ENTER YOUR NAME :");
                Customer_Name=sc.next();
                System.out.println(" ENTER YOUR ACCOUNT NO :");
                Account_No=sc.nextInt();
                System.out.println(" ENTER YOUR ID :");
                Customer_Id=sc.nextInt();
                System.out.println(" ENTER YOUR MOBILE NUMBER :");
                Mobile_No=sc.nextLong();
                System.out.println(" ENTER YOUR ADDRESS :");
                Address=sc.next();
                System.out.println(" ENTER YOUR BRANCH NAME :");
                Branch_Name=sc.next();
                System.out.println(" RESISTRATION SUCCESSFULLY ");
                System.out.println("ENTER AN AMOUNT TO DEPOSIT : ");

        System.out.println("££££££££££££££££££££££££££££££££££££££££££££££££££");
                int amount=sc.nextInt();
                Deposit(amount);
                System.out.println("SUCCESSFULLY DEPOSITED..");
                System.out.println(" \n ");
                Show_Menu();
}
```



Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse

SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 12:33:23)

```
1.  NEW USER
2.  EXISTING USER
3.  EXIT
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
******************************************************************************
ENTER ANY OPTION
******************************************************************************
1


&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
 ENTER YOUR NAME :
PalakGupta
 ENTER YOUR ACCOUNT NO :
101
 ENTER YOUR ID :
123
 ENTER YOUR MOBILE NUMBER :
7000171196
 ENTER YOUR ADDRESS :
Umangcolony
 ENTER YOUR BRANCH NAME :
Raipur
 RESISTRATION SUCCESSFULLY
ENTER AN AMOUNT TO DEPOSIT :
££££££££££££££££££££££££££££££££££££££££££££££££££
10000
SUCCESSFULLY DEPOSITED..


 Welcome  PalakGupta
 Your ID is :  123
 Your ACCOUNT No. is :  101
```


Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Run
SimpleBankingApp
Quick Access
Markers Properties Servers Data Source Explorer Snippets Console Progress
Type here to search
30°C
12:40
15/08/2021

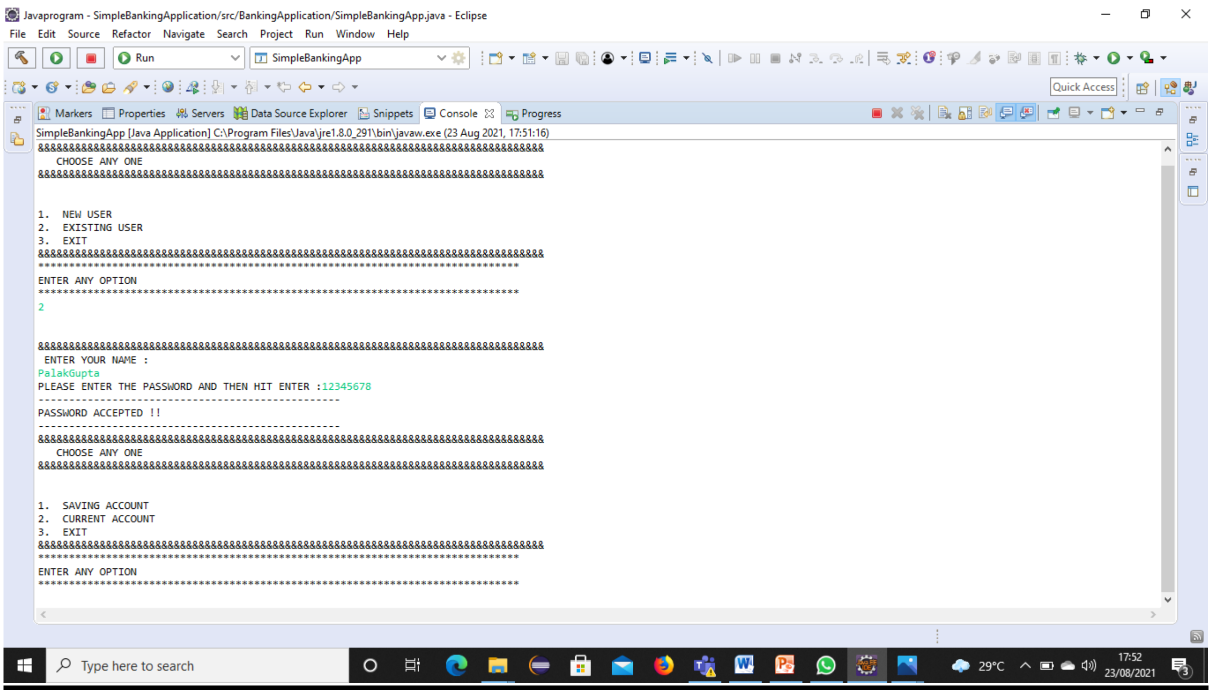
```
SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 12:39:20)

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
 ENTER YOUR NAME :
PalakGupta
 ENTER YOUR ACCOUNT NO :
123
 ENTER YOUR ID :
234
 ENTER YOUR MOBILE NUMBER :
7000171196
 ENTER YOUR ADDRESS :
Umangcolony
 ENTER YOUR BRANCH NAME :
Raipur
 RESISTRATION SUCCESSFULLY
ENTER AN AMOUNT TO DEPOSIT :
££££££££££££££££££££££££££££££££££££££££££££££££££££
5000
SUCCESSFULLY DEPOSITED..


 Welcome  PalakGupta
 Your ID is :  234
 Your ACCOUNT No. is :  123


 CHOOSE ANY ONE AT A TIME


A.  CHECK BALANCE
B.  DEPOSIT AMOUNT
C.  WITHDRAW AMOUNT
D.  CHECK PREVIOUS TRANSECTION
E.  EXIT
F.  CALCULATE INTEREST
```

4.3 . EXISTING USER LOGIN:

4.3.1 . PASSWORD VERIFICATION:

```
void password()
         {
         Scanner sc = new Scanner(System.in);
         boolean valid = false;
         String password;
         System.out.println(" ENTER YOUR NAME :");
         Customer_Name=sc.next();
         while(true)
         {
         System.out.print("PLEASE ENTER THE PASSWORD AND THEN HIT ENTER :");
         password = sc.next();
         if (password.length()<8)
            {
               valid = false;
               System.out.println("PASSWORD MUST HAVE ATLEAST 8 CHARACTER !!!");
            }
         else
         {
   for (int i = 0; i < password.length(); i++)
   {
                  char c = password.charAt(i);

                  if ( ('a' <= c && c <= 'z') || ('A' <= c && c <= 'Z')|| ('0' <= c && c <= '9') )
                   {
                     valid = true;
                  }
                  else
```

```
                {
                    System.out.println("Password denied");
                    System.out.println("Only letter & digits are acceptable.");
                    valid = false;
                    break;
                }
    }

         if (valid == true)
         {
         System.out.println("------------------------------------------------");
         System.out.println("PASSWORD ACCEPTED !!");
         System.out.println("------------------------------------------------");
             break;
                 }
         }
         }
                                                                         }
```



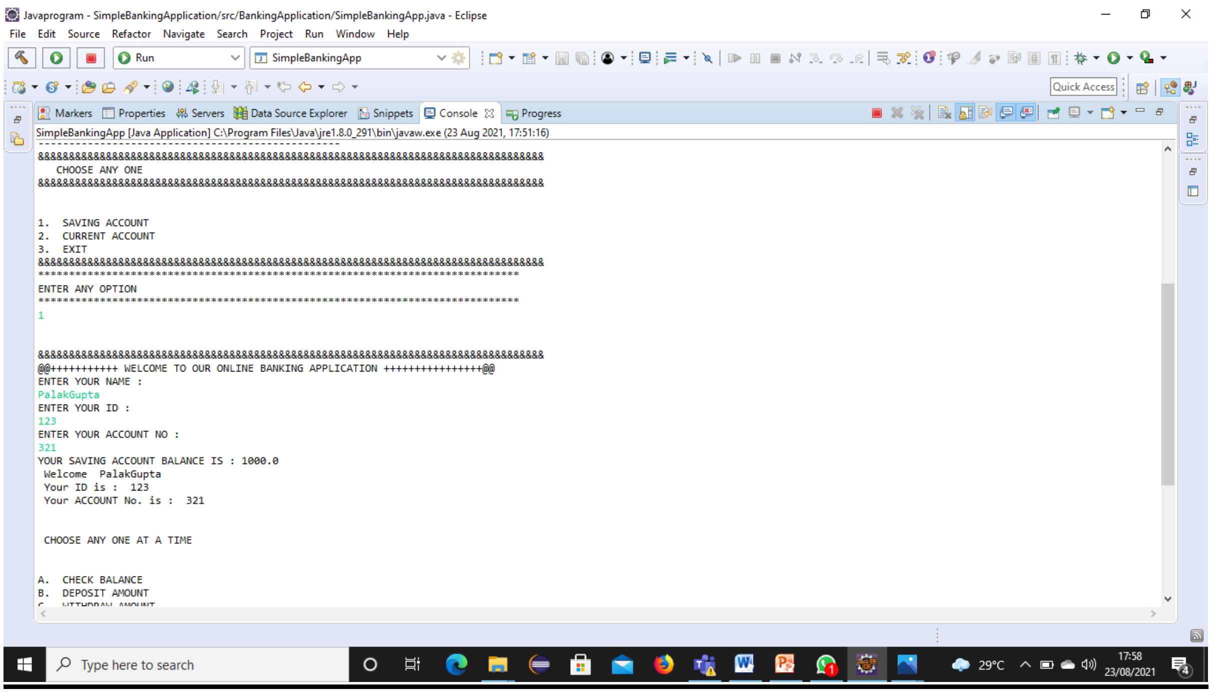
Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse

SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (23 Aug 2021, 17:51:16)

```
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
   CHOOSE ANY ONE
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&


1.  NEW USER
2.  EXISTING USER
3.  EXIT
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
******************************************************************************
ENTER ANY OPTION
******************************************************************************
2


&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
 ENTER YOUR NAME :
PalakGupta
PLEASE ENTER THE PASSWORD AND THEN HIT ENTER :12345678
-------------------------------------------------
PASSWORD ACCEPTED !!
-------------------------------------------------
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
   CHOOSE ANY ONE
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&


1.  SAVING ACCOUNT
2.  CURRENT ACCOUNT
3.  EXIT
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
******************************************************************************
ENTER ANY OPTION
******************************************************************************
```

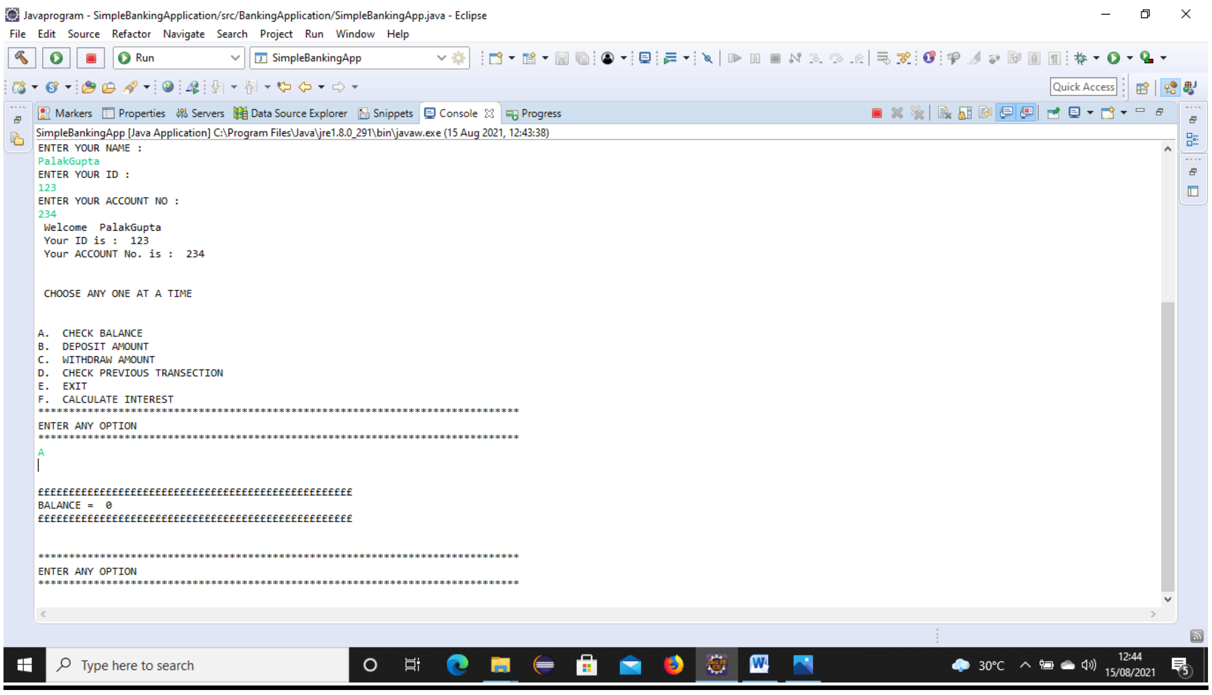
4.3.2 . SAVING ACCOUNT:

```
void SavingAccount()
        {
                Scanner sc = new Scanner(System.in);
                        System.out.println("@@+++++++++++ WELCOME TO OUR ONLINE
BANKING APPLICATION +++++++++++++++@@");
                        System.out.println("ENTER YOUR NAME : ");
                        Customer_Name=sc.next();
                        System.out.println("ENTER YOUR ID : ");
                        Customer_Id=sc.nextInt();
                        System.out.println("ENTER YOUR ACCOUNT NO : ");
                        Account_No=sc.nextInt();
                double SavingAcc= 1000;
                System.out.println("YOUR SAVING ACCOUNT BALANCE IS : " + SavingAcc );
                 char option = '\0';
                        System.out.println(" Welcome  " +Customer_Name);
                        System.out.println(" Your ID is :  "+Customer_Id);
                        System.out.println(" Your ACCOUNT No. is :  "+Account_No);
                        System.out.println(" \n ");
                        System.out.println(" CHOOSE ANY ONE AT A TIME ");
                        System.out.println(" \n ");
                        System.out.println("A.  CHECK BALANCE");
                        System.out.println("B.  DEPOSIT AMOUNT");
                        System.out.println("C.  WITHDRAW AMOUNT");
                        System.out.println("D.  CHECK PREVIOUS TRANSECTION");
                        System.out.println("E.  EXIT");
                        System.out.println("F.  CALCULATE INTEREST ");

                        do
                        {

        System.out.println("****************************************************************************************");
                                System.out.println("ENTER ANY OPTION ");

        System.out.println("****************************************************************************************");
                                option = sc.next().charAt(0);
                                System.out.println(" \n ");

                                switch(option)
                                {
                                case 'A':

        System.out.println("£££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        System.out.println("BALANCE =  " +(Balance+SavingAcc));

        System.out.println("£££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        System.out.println(" \n ");
                                        break;

                                case 'B':

        System.out.println("£££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        System.out.println("ENTER AN AMOUNT TO DEPOSIT : ");

        System.out.println("£££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        int amount=sc.nextInt();
                                        Deposit(amount);
                                        System.out.println("SUCCESSFULLY DEPOSITED..");
                                        System.out.println(" \n ");
                                        break;

                                  case 'C':

          System.out.println("££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        System.out.println("ENTER AN AMOUNT TO WITHDRAW  :
");

          System.out.println("££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        int amount2=sc.nextInt();
                                        Withdraw(amount2);
                                        System.out.println("SUCCESSFULLY WITHDRAWN FROM
YOUR ACCOUNT ..");

                                        System.out.println(" \n ");
                                        break;

                                  case 'D':

          System.out.println("££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        get_Previous_Transection();

          System.out.println("££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        System.out.println(" \n ");
                                        break;

                                  case 'E':

          System.out.println("££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        break;

                                  case 'F':

          System.out.println("££££££££££££££££££££££££££££££££££££££££££££££££££");
          Calculate_Interest();

          System.out.println("££££££££££££££££££££££££££££££££££££££££££££££££££");
                                        System.out.println(" \n ");
                                        break;

                              default:
                                  System.out.println("SORRY!!!! ");
                                  System.out.println("INVALID OPTION!!!!! ");
                                  System.out.println("PLEASE ENTER AGAIN... ");
                                  break;

                                        }
                                  }
                            while(option!= 'E');
                            System.out.println(" THANK YOU FOR USING OUR ONLINE BANKING
APPLICATION SERVICES..");

                                                            }
```



```
SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (23 Aug 2021, 17:51:16)
------------------------------------------------
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
   CHOOSE ANY ONE
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&


1.  SAVING ACCOUNT
2.  CURRENT ACCOUNT
3.  EXIT
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*****************************************************************************
ENTER ANY OPTION
*****************************************************************************
1


&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
@@+++++++++++ WELCOME TO OUR ONLINE BANKING APPLICATION ++++++++++++++++@@
ENTER YOUR NAME :
PalakGupta
ENTER YOUR ID :
123
ENTER YOUR ACCOUNT NO :
321
YOUR SAVING ACCOUNT BALANCE IS : 1000.0
 Welcome  PalakGupta
 Your ID is :  123
 Your ACCOUNT No. is :  321


 CHOOSE ANY ONE AT A TIME


A.  CHECK BALANCE
B.  DEPOSIT AMOUNT
```

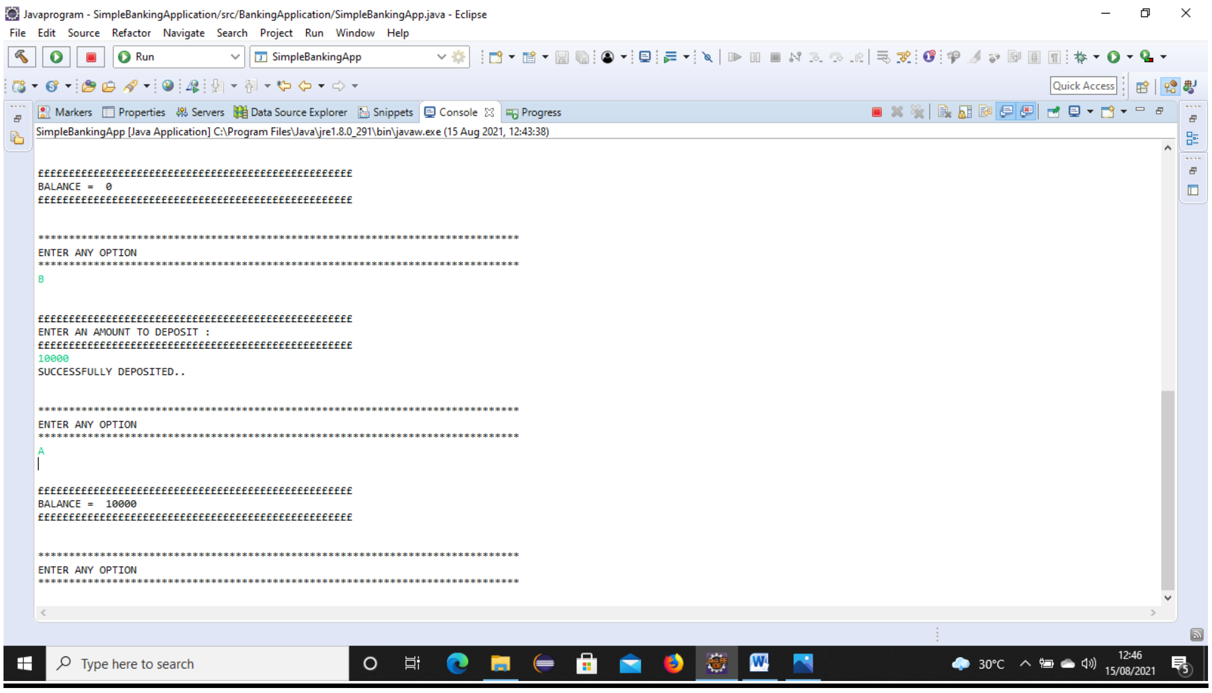
4.4 . CHECK BALANCE:



```
SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 12:43:38)
ENTER YOUR NAME :
PalakGupta
ENTER YOUR ID :
123
ENTER YOUR ACCOUNT NO :
234
 Welcome  PalakGupta
 Your ID is :  123
 Your ACCOUNT No. is :  234


 CHOOSE ANY ONE AT A TIME


A.  CHECK BALANCE
B.  DEPOSIT AMOUNT
C.  WITHDRAW AMOUNT
D.  CHECK PREVIOUS TRANSECTION
E.  EXIT
F.  CALCULATE INTEREST
******************************************************************************
ENTER ANY OPTION
******************************************************************************
A


££££££££££££££££££££££££££££££££££££££££££££££££££££
BALANCE =  0
££££££££££££££££££££££££££££££££££££££££££££££££££££


******************************************************************************
ENTER ANY OPTION
******************************************************************************
```

4.5 . DEPOSIT AMOUNT:

```
void Deposit(int amount)

	{
		if(amount>0)
		{
			Balance = Balance + amount;
			Previous_Transection = amount;

		}
}
```



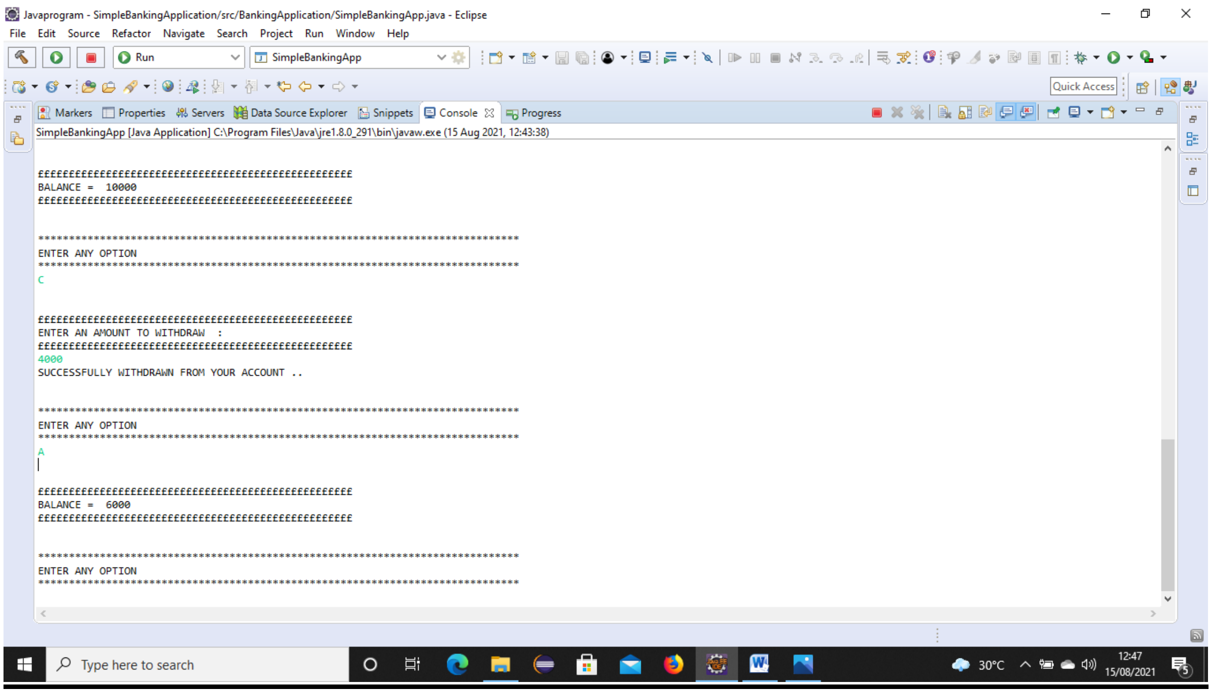
Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Run
SimpleBankingApp
Quick Access
Markers Properties Servers Data Source Explorer Snippets Console Progress
Type here to search
30°C
12:46
15/08/2021

SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 12:43:38)

```
££££££££££££££££££££££££££££££££££££££££££££££££££
BALANCE =  0
££££££££££££££££££££££££££££££££££££££££££££££££££


*****************************************************************************
ENTER ANY OPTION
*****************************************************************************
B


££££££££££££££££££££££££££££££££££££££££££££££££££
ENTER AN AMOUNT TO DEPOSIT :
££££££££££££££££££££££££££££££££££££££££££££££££££
10000
SUCCESSFULLY DEPOSITED..


*****************************************************************************
ENTER ANY OPTION
*****************************************************************************
A


££££££££££££££££££££££££££££££££££££££££££££££££££
BALANCE =  10000
££££££££££££££££££££££££££££££££££££££££££££££££££


*****************************************************************************
ENTER ANY OPTION
*****************************************************************************
```

4.6 . WITHDRAW AMOUNT:

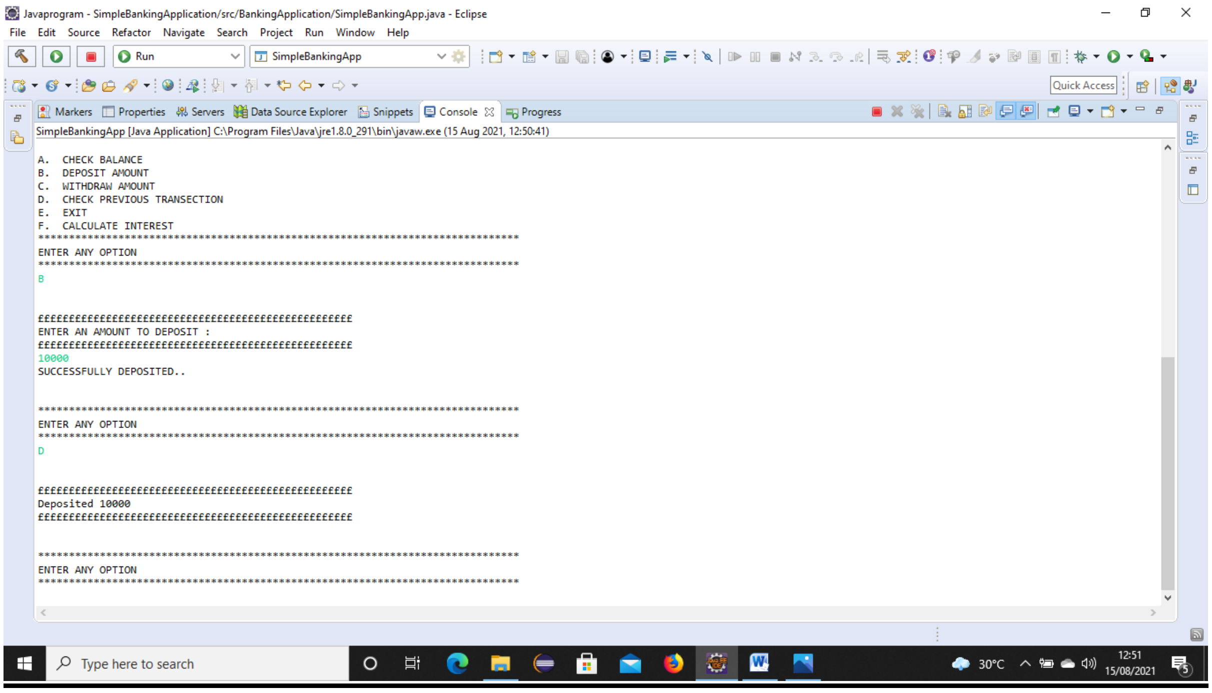
```
void Withdraw(int amount)
        {
                if(amount>0)
                {
                        Balance =Balance - amount;
                        Previous_Transection=Previous_Transection - amount;

                }
        }
```


Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Run
SimpleBankingApp
Quick Access
Markers Properties Servers Data Source Explorer Snippets Console Progress
Type here to search
30°C
12:47
15/08/2021

```
SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 12:43:38)


£££££££££££££££££££££££££££££££££££££££££££££££££££
BALANCE =  10000
£££££££££££££££££££££££££££££££££££££££££££££££££££


******************************************************************************
ENTER ANY OPTION
******************************************************************************
C


£££££££££££££££££££££££££££££££££££££££££££££££££££
ENTER AN AMOUNT TO WITHDRAW  :
£££££££££££££££££££££££££££££££££££££££££££££££££££
4000
SUCCESSFULLY WITHDRAWN FROM YOUR ACCOUNT ..


******************************************************************************
ENTER ANY OPTION
******************************************************************************
A


£££££££££££££££££££££££££££££££££££££££££££££££££££
BALANCE =  6000
£££££££££££££££££££££££££££££££££££££££££££££££££££


******************************************************************************
ENTER ANY OPTION
******************************************************************************
```

4.7 . CHECK PREVIOUS TRANSECTION:

```
void get_Previous_Transection()
        {
                if(Previous_Transection > 0)
                {
                        System.out.println("Deposited "   +Previous_Transection);
                }
                else if(Previous_Transection < 0)
                {
                        System.out.println("WithDrawn  "   +Math.abs(Previous_Transection));
                }
                else
                {
                        System.out.println("No Transection Occured");
                }
                                                }
```

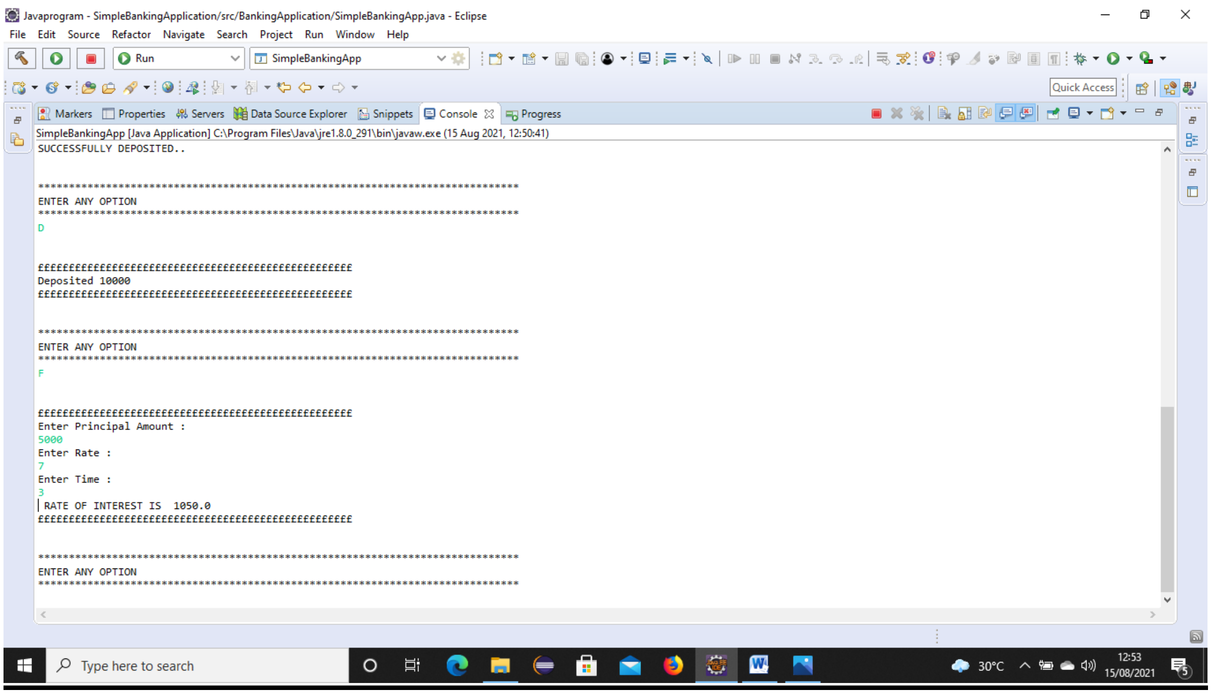

Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Run
SimpleBankingApp
Quick Access
Markers Properties Servers Data Source Explorer Snippets Console Progress
Type here to search
30°C
12:51
15/08/2021

SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 12:50:41)

```
A.  CHECK BALANCE
B.  DEPOSIT AMOUNT
C.  WITHDRAW AMOUNT
D.  CHECK PREVIOUS TRANSECTION
E.  EXIT
F.  CALCULATE INTEREST
******************************************************************************
ENTER ANY OPTION
******************************************************************************
B


£££££££££££££££££££££££££££££££££££££££££££££££££££
ENTER AN AMOUNT TO DEPOSIT :
£££££££££££££££££££££££££££££££££££££££££££££££££££
10000
SUCCESSFULLY DEPOSITED..


******************************************************************************
ENTER ANY OPTION
******************************************************************************
D


£££££££££££££££££££££££££££££££££££££££££££££££££££
Deposited 10000
£££££££££££££££££££££££££££££££££££££££££££££££££££


******************************************************************************
ENTER ANY OPTION
******************************************************************************
```

4.8 . CALCULATE RATE OF INTREST:

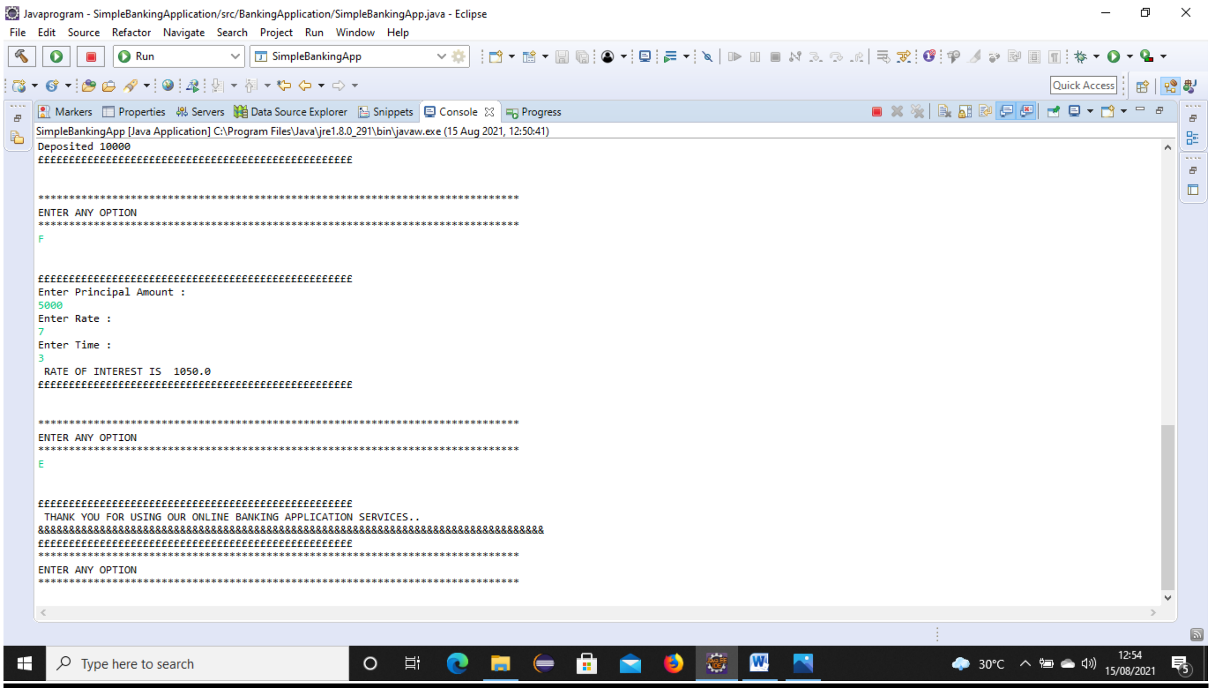
```
void Calculate_Interest()
        {
                int Principal,Rate,Time;
                double RateOfInterest;
                Scanner sc= new Scanner(System.in);
                System.out.println("Enter Principal Amount : ");
                Principal=sc.nextInt();
                System.out.println("Enter Rate : ");
                Rate=sc.nextInt();
                System.out.println("Enter Time : ");
                Time=sc.nextInt();
                RateOfInterest=(Principal*Rate*Time)/100;
                System.out.println(" RATE OF INTEREST IS  " +RateOfInterest);

                                                    }
```


SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 12:50:41)

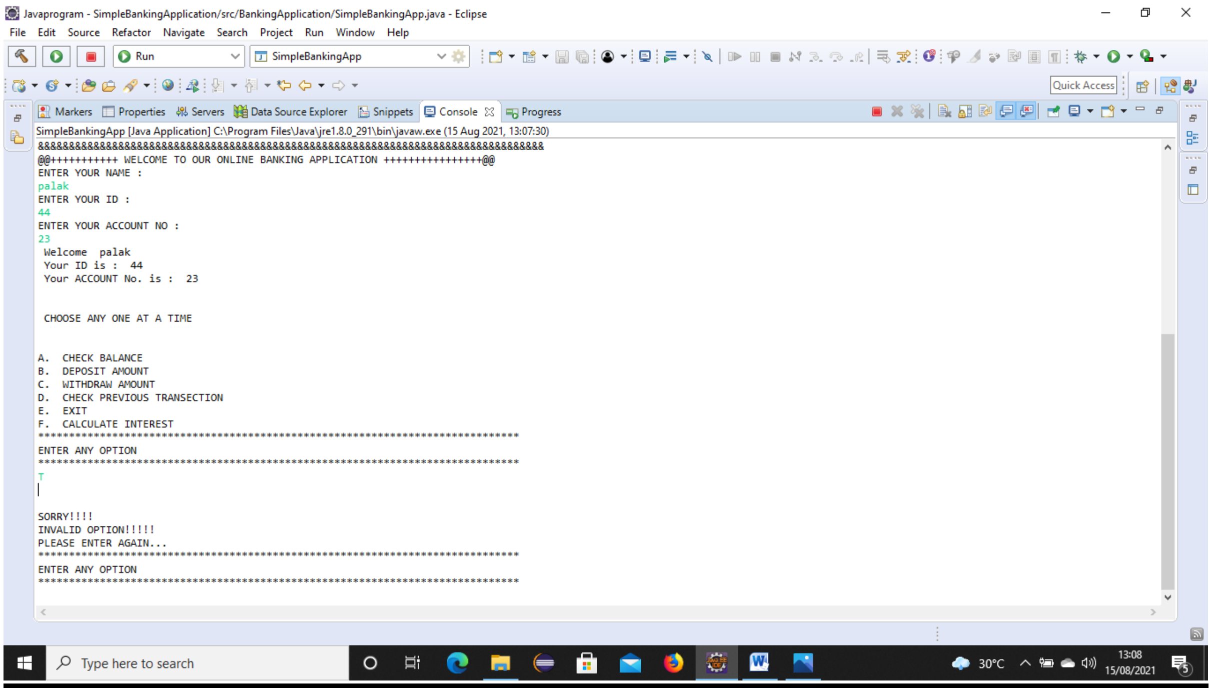
```
SUCCESSFULLY DEPOSITED..


******************************************************************************
ENTER ANY OPTION
******************************************************************************
D


£££££££££££££££££££££££££££££££££££££££££££££££££££
Deposited 10000
£££££££££££££££££££££££££££££££££££££££££££££££££££


******************************************************************************
ENTER ANY OPTION
******************************************************************************
F


£££££££££££££££££££££££££££££££££££££££££££££££££££
Enter Principal Amount :
5000
Enter Rate :
7
Enter Time :
3
 RATE OF INTEREST IS  1050.0
£££££££££££££££££££££££££££££££££££££££££££££££££££


******************************************************************************
ENTER ANY OPTION
******************************************************************************
```

Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Run
SimpleBankingApp
Quick Access
Markers Properties Servers Data Source Explorer Snippets Console Progress
Type here to search
30°C
12:53
15/08/2021

4.9 EXIT AND WRONG INPUT:


Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Run
SimpleBankingApp
Quick Access
Markers Properties Servers Data Source Explorer Snippets Console Progress
Type here to search
30°C
12:54
15/08/2021

```
SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 12:50:41)
Deposited 10000
££££££££££££££££££££££££££££££££££££££££££££££££


*******************************************************************************
ENTER ANY OPTION
*******************************************************************************
F


££££££££££££££££££££££££££££££££££££££££££££££££
Enter Principal Amount :
5000
Enter Rate :
7
Enter Time :
3
 RATE OF INTEREST IS  1050.0
££££££££££££££££££££££££££££££££££££££££££££££££


*******************************************************************************
ENTER ANY OPTION
*******************************************************************************
E


££££££££££££££££££££££££££££££££££££££££££££££££
 THANK YOU FOR USING OUR ONLINE BANKING APPLICATION SERVICES..
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
££££££££££££££££££££££££££££££££££££££££££££££££
*******************************************************************************
ENTER ANY OPTION
*******************************************************************************
```


Javaprogram - SimpleBankingApplication/src/BankingApplication/SimpleBankingApp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Run
SimpleBankingApp
Quick Access
Markers Properties Servers Data Source Explorer Snippets Console Progress
Type here to search
30°C
13:08
15/08/2021

```
SimpleBankingApp [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (15 Aug 2021, 13:07:30)
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
@@+++++++++++ WELCOME TO OUR ONLINE BANKING APPLICATION +++++++++++++++@@
ENTER YOUR NAME :
palak
ENTER YOUR ID :
44
ENTER YOUR ACCOUNT NO :
23
 Welcome  palak
 Your ID is :  44
 Your ACCOUNT No. is :  23


 CHOOSE ANY ONE AT A TIME


A.  CHECK BALANCE
B.  DEPOSIT AMOUNT
C.  WITHDRAW AMOUNT
D.  CHECK PREVIOUS TRANSECTION
E.  EXIT
F.  CALCULATE INTEREST
*******************************************************************************
ENTER ANY OPTION
*******************************************************************************
T


SORRY!!!!
INVALID OPTION!!!!!
PLEASE ENTER AGAIN...
*******************************************************************************
ENTER ANY OPTION
*******************************************************************************
```

CONCLUSION

Banking systems have been with us for as long as people have been using money. Banks and other financial institutions provide security for individuals, businesses and governments, alike. Let's recap what has been learned with this tutorial:

In general, what banks do is pretty easy to figure out. For the average person banks accept deposits, make loans, provide a safe place for money and valuables, and act as payment agents between merchants and banks.

Banks are quite important to the economy and are involved in such economic activities as issuing money, settling payments, credit intermediation, maturity transformation and money creation in the form of fractional reserve banking.

To make money, banks use deposits and whole sale deposits, share equity and fees and interest from debt, loans and consumer lending, such as credit cards and bank fees.

In addition to fees and loans, banks are also involved in various other types of lending and operations including, buy/hold securities, non-interest income, insurance and leasing and payment treasury services.

History has proven banks to be vulnerable to many risks, however, including credit, liquidity, market, operating, interesting rate and legal risks. Many global crises have been the result of such vulnerabilities and this has led to the strict regulation of state and national banks.

However, other financial institutions exist that are not restricted by such regulations. Such institutions include: savings and loans, credit unions, investment and merchant banks, shadow banks, Islamic banks and industrial banks.

REFERENCE

- WWW.EDUBRIDGEINDIA.COM