NAME: S.RAMYASHREE

BRANCH: COMPUTER SCIENCE ENGINEERING

PROJECT NAME: E-BANKING

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

In the modern world of technology, computers associate degreed technologies are playing a very important role whereas the bank related works like opening new bank account, depositing cash or withdrawal of cash are done manually by ink pen and paper which consumes lot of time and far efforts and very slow process. Therefore, the aim of the project is to design a program for Electronic banking which demonstrates the speed of its growth is incredibly quick, fast and efficient and consumes less time and efforts.

1.2 OBJECTIVES

The main objectives of this project are:

- It is user friendly
- It saves the time spent in banks
- It provides banking activities throughout the year 24/7 days from any place and anywhere
- It provides well-organized cash management system for net improvement
- It additionally provides more convenience in terms of capital, labor and time resources needed to make a transaction
- It is easy to deposit or withdrawal the cash from the existing bank account
- It is convenient for the clients to view their bank balance enquiry

1.3 METHODOLOGY

In this project, Object Oriented Concepts of java plays a major role in e-banking system.

- In this project, it offers the choice for the clients that whether they wish to open a new bank account or they have to view their existing bank account which includes deposit or withdrawal of money.
- If a client desires to open a new bank account then the details are accessed and stored.
- If client has existing bank account then they can deposit the money or they can withdraw the money or they can also view their bank details together with bank balance.

This is how the project deals with the e-banking system.

1.4 EXPECTED OUTCOMES

MAIN MENU:

1. Create new account

2. Search by account

3. Deposit

4. Withdrawal

5. Display all the account

6. Exit

IF A CLIENT WISH TO SEARCH THEIR BANK ACCOUNT (CHOICE 2)

CASE 1: If bank account exists, then it displays the details of the particular bank account

Enter the bank account u want to search: 100

Account number: 100

Name: Dhanush

Age: 19

Father name: Manish

City: Bangalore

State: Karnataka

Balance: 1500

CASE 2: If bank account does not exist, then it displays account not exist.

```
Enter the bank account u want to search: 500

SEARCH FAILED.. ACCOUNT NOT EXIST..
```

1.5 HARDWARE REQUIREMENTS

- Processor : any processor
- RAM : 256MB or more
- Output device : any screen

1.6 SOFTWARE REQUIREMENTS

- Operating system : Windows, MAC or linux
- Programming language : java
- User interface : HTML, APPLET

CHAPTER 2

OBJECT ORIENTED CONCEPTS

Java is an object-oriented program, platform independent, multi-purpose programming language which is produced by Sun Micro system which is currently the subsidiary of Oracle. Java is a high-level programming language which is portable and platform independent. Java is fast, secure and reliable. It was first released in the year 1995; it was developed to be a machine independent web technology. It was developed based on C and C++ language syntax to make it easy and simple for programmers. Since then, Java has earned a prominent place in the world of computer programming.

FEATURES OF JAVA / CHARACTERISTICS OF JAVA / BUZZWORDS OF JAVA ARE:

- Simple
- Secure
- High performance
- Object oriented
- Distributed
- Complied and Interpreted
- Portable
- Dynamic
- Architecture neutral
- Robust
- Multithread

SIMPLE

Java was designed to make easy for professional programmers to learn quickly and use effectively. It is simple and easy to learn for the programmer who already know the basic concepts of C / C++ language because It was developed from C / C ++ Language syntax.

SECURE

Java is confined solely to the Java execution environment- JVM (JAVA VIRTUAL MACHINE). When a web browser of java compatible is used, downloading can be done safely and easily without any concern or fear of viral infections.

HIGH PERFORMANCE

Performance of the Java is high because of usage of the byte code. Byte code is the instruction set designed for the efficient execution. Byte code simply interprets the code into native machine code thus it enables us to execute in any operating system. Java is faster when compared with other traditional interpreted programming language.

OBJECT ORIENTED

Java is an object-oriented language. All the program code and data reside within the objects and classes. The object model in Java is simple, easy to extend, easy to maintain and it is also reusability. Java comes with an extensive group of classes that are organized in packages which may be used in programs through inheritance.

DISTRIBUTED

Java is intended to develop a distributed environment. Java is used for creating applications on network. It permits programmers in multiple remote locations to collaborate and work together on a single project.

COMPILED AND INTERPRETED

Computer language is either compiled or interpreted. But Java combines both compiled and interpreted and makes it into two stage system.

- **COMPILED:** Java enables the creation of a cross platform programs by compiling it into an intermediate representation which is called Java Bytecode.
- **INTERPRETED:** After compiled, Byte code is interpreted which generates machine code that can be directly executed by the machine which provides a Java Virtual Machine.

PORTABLE

It helps in generating Portable executable code by providing a way to download programs dynamically to all various kinds of platforms connected to the internet.

DYNAMIC

Java can link in new methods and strategies, new class libraries and objects. It can also link native methods (the function which is written in different programming language such as C and C++). It additionally has a compilation and automatic memory management.

ARCHITECTURE NEUTRAL

Java language and Java Virtual Machine (JVM) helped in achieving the goal i.e, "WRITE ONCE; RUN ANYWHERE, ANY TIME, FOREVER". change in operating systems or update in operating systems, processor and system resources will not force any changes in Java programs.

ROBUST

It provides strong memory management, avoids security issues, automatic garbage-collection and additionally exception handling. It also provides several options that build the program execute reliably in numerous types of environment. Java is strictly typed language/ written Language which checks the code both at compile time and runtime.

MULTITHREADED

It helps in parallel execution i.e., several tasks performing at once. Multithreaded programs handle multiple tasks at the same time that helps in creating interactive, networked programs. Java run time system support the synchronization of multi process which is used to construct smoothly interactive systems.

JAVA ENVIRONMENT

The programming Java environment consists of three components mainly:

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

These three components / elements are platform dependent because the configuration of each Operating System is totally different from one another. Whereas, Java is platform independent.

1) JAVA DEVELOPMENT KIT (JDK)

JDK is the environment for software development which is used to develop Java applications and applets.

SEVEN MAIN TOOLS IN JDK ARE: -

- The Java compiler - javac
- The Java interpreter - Java
- Generates documentation in HTML - javadoc
- The Java interpreter to execute Java applets - appletviewer
- The java debugger to sort out bugs and fix bugs in Java program - jdb
- The Java disassemble to displays the accessable functions, information and data - javap
- To Create interface between Java and C routines – javah

2) JAVA RUNTIME ENVIRONMENT (JRE)

JRE provides minimum needs (requirements) for executing a Java application. It includes the Java Virtual Machine (JVM), core classes and supporting files.

3) JAVA VIRTUAL MACHINE (JVM)

JVM is an abstract machine. It is additionally called as Virtual Machine because it doesn't exit physically. It can run other programs which are written in other languages and compiled to Java bytecode. It can also provide a run-time environment in which Java bytecode can be executed.

The three notions of JVM are:

- **SPECIFICATION** The operating of Java Virtual Machine (JVM) is fixed whereas the implementation provided was independent to pick out the algorithm. Implementation of JVM was provided by Oracle and other companies.

- **IMPLEMENTATION:** It is a computer program which meets the requirements of the Java Virtual Machine's specification. Its implementation is additionally known as Java Runtime Environment (JRE).
- **RUNTIME INSTANCE:** An instance of JVM is formed when the programmer writes Java command on the command prompt to run the Java class

JVM performs few main tasks such as loads the code, verifies/checks the code and eventually executes the code.

- The main method which is present in Java code was called by JVM.
- JVM is also known as Runtime Interpreter
- JVM widely helps in the abstraction of the inner implementation for the programmers who utilities the libraries to develop their programs from JDK.

DATA TYPES IN JAVA

1) primitive data type

2) non-primitive data type

1) PRIMITIVE DATA TYPE

It is the fundamental building blocks of data manipulation in Java. It can be directly manipulated by machine instructions.

It is classified into 2 types:

a) Boolean - It is a special type of data which stores only 2 possible values i.e, true and false.

b) Numerical - It is classified into character and integral.

- CHARACTER: Character includes char data type, which is used to represent the symbols within the set of Character, such as numbers and letters.
- INTEGRAL: Integral is further classified into Integer and floating point.
  - INTEGER:
    - Byte
    - Short
    - Int

        - Long
    - FLOATING POINT
        - Float
        - Double

2) NON-PRIMITIVE DATA TYPE

- String
- Array
- Class
- Instance

2.1 CLASS

Class is defined as blueprint or template which describes the behaviors or states or properties from which objects gets created. Class can also defined as cluster of objects having common properties. Class can contain variables, Methods, Constructors, blocks and interface. Declaration of class is done by using "class" keyword.

SYNTAX:

```
class classname{
datatype instancevar1;
datatype instancevar2;
//...
datatype instancevarN;
type methodname1(parameter_list)
{
//Body of the method
}
type methodname2(parameter_list)
{
//Body of the method
}
//...
type methodnameN(parameter_list)
{
//Body of the method
}
}
```

2.1.1 TYPES OF VARIABLES IN CLASS

- Local variables
- Instance variables
- Static variables

LOCAL VARIABLES

- A variable which are declared inside the methods, constructors, or blocks are called local variables.
- Local variables are created during the entrance of method, constructor, or block and variables are destroyed when the methods, constructors or blocks exits.
- Access modifier such as private, default, public, protected etc. cannot be used for local variables
- The declaration and initialization of the variables should be done before the first use because local variables do not have default values.

INSTANCE VARIABLE

- Instance variable are the variables which are declared inside the class but outside the body of the constructors, blocks or methods.
- Value of the variable is instance specific and it cannot share among instances. Therefore, it is known as instance variables.

- Instance variables can be created when an object is formed with the help of the keyword "new" and instance variable get be destroyed when the objects are destroyed.
- Access modifiers like public, private, protected can be used for instance variables.
- There is a default value for instance variable. The default value for numbers is 0, default values for Boolean is false, and default value for object reference is null.

STATIC VARIABLE

- Variables which are declared inside the class with a "static" keyword but outside the constructors, block and method are called static variable.
- Programmers can create a single copy of static variables that can be shared among all the instances of the class.
- These variables are created when the program runs or starts, and these can be destroyed when the program stops.
- There is a default value for static variable. The default value for numbers is 0, default values for Boolean is false, and default value for object reference is null.
- For static variables, memory allocation takes place only single time when the class is loaded in the memory.

EXAMPLE FOR ALL THE VARIABLES:

```
class expforvariable

{

int data =50;                 // instance variable

static int value = 100;       // static variable

void method ()

{

int a = 25:.                 //Local variable

}

}
```

2.2 OBJECT

Object is an instance of classes. Objects contains States and behaviors. For example, a cat has states such as color of the cat, name of the cat and breed. It additionally has behaviors such as scratching, eating and generally wagging the tail. Object is a two step process. The primary step is to declare the variables of the class type which refers to an object. The Second step is to acquire an actual copy of the object, physical copy of the object and assigning it to the variable by using the "new" operator. The "new" operator allocates the memory dynamically (i.e., allocates memory at run time) for an object and returns the reference to it. This reference is the address in memory of the object which is allocated by "new" operator.

It requires three steps to form an object. They are declaration, instantiation and initialization.

- **DECLARATION -** This is the step where declaring a variable takes place i.e, variable declaring with the variable name with an object.
- **INSTANTIATION -** The "new" operator is used to create the object and to allocate the memory.
- **INITIALIZATION -** The "new" operator which is followed by a call to the constructor. This call to the constructor initializes the new object.

SYNTAX:

```
classname objname;        // reference to the object
Objname = new classname();     // allocates a class object
```

Where

Objname = name of the object

classname() = class name followed by parentheses () represents the constructor for the class.

EXAMPLE:

```
Cube mycube;
mycube = new Cube();
```

2.3 INHERITANCE

Inheritance is one of the mechanisms which permits the creation of hierarchical classification. It creates a general class which includes common sets of features or items. This is can be used to design specific classes which has the features of general class and has unique features and properties. Derived class can have all the features of the base class / general class and it can also have some extra features i.e., child class acquires or inherits all the properties, features or behaviors of the parent class. Inheritance in Java is used to develop the code reusability.

"extends" is the keyword used for the Inheritance. In simple words, inherited class is referred as Super class and inheriting class is referred as sub class.

SYNTAX:

```
class super_class_name

{

}

class sub_class_name extends super_class_name

{

{
```

EXAMPLE:

```
public class A     //superclass

{

//Methods and body of the class

}

public class B extends A    //subclass

{

//Methods and body of the class

}
```

2.3.1 IMPORTANT TERMINOLOGIES OF INHERITANCE IN JAVA

- **CLASS -** Class is the cluster of objects which contains common properties. Class acts as a template / blueprint which helps in creation of object.
- **SUPER CLASS -** Super class is the class whose properties or features are inherited. Super class is additionally known as base class or parent class.
- **SUB CLASS -** Sub class is the class which inherits the properties or features of the other class (i.e., super class). Sub class is additionally known as derived class or extended class or child class. Derived class can have the fields and methods of the super class and additionally it can also have its own fields and methods.
- **REUSABILITY -** Inheritance helps in writing the reusable code. As an example, when a software engineer desires to create a new class and there is an already existing class which contains some of the code (i.e., few fields and methods) that programmers requires. Then instead of writing the code once more they can derive new class from the already existing class. Thus, the programmers are reusing the fields and methods of the existing class. Hence reusability in the code is achieved.

2.3.2 DIFFERENT FORMS OF INHERITANCE

There are 3 different kinds of inheritance.

- Single inheritance
- Hierarchical inheritance
- Multi-level inheritance

SINGLE INHERITANCE

Single inheritance is one of the inheritances in which the features of the super class (parent class) was inherited by the sub class (child class).

GENERAL FORM OF SINGLE INHERITANCE:

```
class A

{

}
```

```
class B extends A
{
}
```

HIERARCHICAL INHERITANCE

Hierarchical inheritance contains more than one sub class (child class). In hierarchical inheritance one class acts as the super class (parent class) for the sub class which is more than one (child class).

GENERAL FORM OF HIERARCHICAL INHERITANCE:

```
class A
{
}
class B extends A
{
}
class C extends A
{
}
```

MULTILEVEL INHERITANCE

In multi-level inheritance, base class was inherited by the derived class and the same derived class serves as the base class to the other classes.

GENERAL FORM OF MULTILEVEL INHERITANCE:

```
class A
{
}
class B extends A
{
}
class C extends B
{
}
```

2.4 POLYMORPHISM

Polymorphism in Java is a process in which single action can be performed in several ways. Polymorphism is the combination of two words: poly and morphs which are derived from Greek. The meaning of poly is many, and the meaning of morphs is forms. Thus, polymorphism means "many forms".

REAL LIFE EXAMPLE FOR POLYMORPHISM: A person can have various characteristics for various situation at the same time. Once at a time woman could be a mother to her child, a wife to her husband and an employee in the office. Thus, there are totally different characteristics in the same person for different situations. Hence this process can be referred as polymorphism.

Polymorphism contains two forms:

- **STATIC BINDING**: Static binding takes place during the compilation time. Static binding can use final, private and static methods as well as variables which are bonded by compiler.
- **DYNAMIC BINDING**: Dynamic binding takes place during the run time. Overridden methods are bonded during runtime which depends upon the type of runtime object.

2.4.1 TYPES OF POLYMORPHISM

Polymorphism in Java has 2 types:

- Runtime polymorphism
- Compile time polymorphism

RUNTIME POLYMORPHISM

Runtime polymorphism is additionally referred as dynamic method dispatch. It determines the mechanism by which it call to a method of overridden that is resolved at runtime, not by compile time. The version of the overridden method is decided/determined by the kind of the object being referred instead of the kind of reference variable. Method overridden is the one of the way to achieve runtime polymorphism.

- **METHOD OVERRIDING:** This property permits the derived class (sub class) to provide specific implementation of the method which is already provided by one amongst its base class (super class). In other words, implementation in derived class (sub class) overrides the implementation in base class (super class) by providing methods of same parameters, same name and same return type.

EXAMPLE FOR METHOD OVERRIDING

```
class X
{
}
class Y extends X
{
public void display()
{
System.out.println(“Method1”);
}
}
class Z extends X
```

```
{
public void display()
{
System.out.println(“Method2”);
}
}
```

COMPILE TIME POLYMORPHISM

Compile time polymorphism is additionally known as static polymorphism. Method overloading or operator overloading is one of the ways to achieve compile time polymorphism. The increment of the dependableness (reliability) of the program can be achieved by the method overloading.

- **METHOD OVERLOADING:** This is the property in which a class contains two methods or more than two methods which having same name but totally different parameters. Method overloading can be achieved by changing number of parameters (arguments) and by changing the data type of the parameters (arguments). Change in return type does not affect the method overloading.

EXAMPLE FOR METHOD OVERLOADING

```
class Adder
{
Static int add(int a, int b)     //int datatype and 2 arguments
{
return a+b;
}
Static float add(float a, float b, float c)     //float datatype and 3 arguments
{
return a+b+c;
}
}
class Overloading
{
public Static void main(String args[ ])
{
System.out.println(Adder.add(10,10));
System.out.println(Adder.add(10.2f, 10.3f, 10.5f));
}
}
```

2.5 ABSTRACT CLASS

In Java, "abstract" is the keyword which is used to create the abstract class. It can either have abstract methods and non-abstract methods. It can also additionally contain final keyword, final methods and final variables. Abstract classes can be sub classed, but they cannot be instantiated. Any derived class of abstract must either implement all the abstract method in the super class or it can be declared as abstract itself.

SYNTAX:

```
abstract class class_name
{
non abstract methods()
{
//Body
}
abstract methods();
}
```

2.6 MULTITHREADING

In Java, Multithreading is a process in which execution of two or more threads takes place simultaneously to maximize the utilization of computer processing unit (CPU). It's conjointly referred as Concurrency in Java because the applications of Multithread execute two or more threads run concurrently. Each thread in java runs parallel to each other. Multithreads will not allot separate memory area thus they occupy less memory. Switching between the threads takes place which reduces the time consumption.

2.6.1 BENEFITS OF MULTITHREADS

- Even though the programmers or users performs multiple tasks simultaneously they were not blocked because threads are independent. Thus, the programmer can perform multiple operations simultaneously.
- Even though one thread meets an exception it does not have effect on the remaining threads.

2.6.2 DIFFERENT STAGES OF LIFE CYCLE OF THREADS

- New
- Runnable
- Running
- Waiting
- Dead

2.7 I/O FUNCTIONS

Java Input/output functions are used to give the input and take the output. To perform the I/O operation fast, we tend to use the concepts of stream in Java. File handling can be performed by using Java I/O API. Stream is a sequence of data which is composed of bytes.

In java, there are three streams created automatically

- **System.in:** it is a standard input stream
- **System.out:** it is a standard output stream
- **System.err:** it is a standard error stream

EXAMPLE:

```
Scanner s=new Scanner(System.in);

System.out.println(“NHCE”);

System.err(0);
```

In Java application, input streams are used to read the data or information from the source. output streams are used to write the data or information to the destination. Source and destination can be a file, peripheral decide, an array or socket.

2.8 JAVA PACKAGES

Java provides a technique for the partitioning (separating) class name space into manageable chunks. Java package is a bunch of similar kinds of interfaces, classes and sub -packages. Built -in packages and user -defined packages are categorized into two forms of packages. Java, Lang, awt, javax, swing is the few built -in packages in Java. Benefits of packages are it provides access protection and eliminates the naming collision. The "package" is the keyword used to create a package in Java.

GENERAL FORM OF PACKAGE STATEMENT:

```
package pkg;
```

Where pkg is the package name

To compile the Java package program: javac -d directoryfilename.java

To run the Java package program: Java package name.filename

2.9 EXCEPTION HANDLING

In Java, one amongst powerful mechanisms is exception handling which handles the runtime error when there is a disrupts in the normal flow of the program. Checked exception and unchecked exception are the two types of exception handling. Checked exception is the class in which they are directly inherits the throwable class. They are checked at compile time. Unchecked exception is the class in which they inherit runtime exception. They are checked at runtime. IOException is one of the examples for checked exception whereas arithmetic exceptions, null pointer exception are the example for unchecked exception.

2.9.1 FIVE KEYWORDS OF EXCEPTION HANDLING

1. TRY

- "try" keyword is used to represent a block where exception code should be placed. Try block cannot be executed alone. try block ought to be followed by either catch block or finally block.

2. CATCH

- Usage of the "catch" block is to handle the exceptions. Catch block is as same as try block, which cannot be executed alone. It should be preceded by try block. Finally block can be succeeded after catch block.

3. FINALLY

- "finally" block is used to execute the important code of the program. It doesn't matter whether exception is handled or not, finally block will be executed.

4. THROW

- Usage of the "throw" keyword is to throw an exception.

5. THROWS

- "throws" keyword is used to declare the exceptions.

CHAPTER 3

ALGORITHM

Algorithm is a procedure which describes step by step to solve the problem or to reach the programming goal in understandable way and intelligible approach. Algorithm includes a begin (start), a middle and last step as finish (end). In fact, programmers has the tendency to label the first step as start and last step as end. Pseudocode is a semi - programming language which is used to describe the steps in associative algorithmic program.

Step 1: start

Step 2: insert the header files which are required for the project.

Step 3: declare all the variables which are required for the project.

Step 4: create the methods and functions to Perform required specifications.

Step 5: to develop the program more effectively as well as efficiently, oops concepts such as class, array objects, inheritance and exceptional handling was implemented.

Step 6: Initially, it displays the bank name "EASYBANKFORYOU" then it displays the address of the bank as "KADUBESANAHALLI BRANCH, MARATHAHALLI"

Step 7: client or admin can input the number of customers initially and has to enter the details to create bank account.

Step 8: it displays the main menu. Client has to enter the choice.

1. Create a new account
2. Search by account number
3. Deposit the money
4. Withdraw the money
5. Display all the accounts
6. Exit

Step 9: by entering the choice 1, client can create the new bank account and has to enter the details such as:

- Name
- Age

- Father name
- City
- State
- Balance

After entering all the details it prints "YOUR ACCOUNT SUCCESSFULLY CREATED" and it displays the account number as "YOUR ACCOUNT NUMBER IS XXX" (XXX is the account number)

Step 10: by inputting the choice 2, client can search their bank account details by entering their bank account number. If the customer's bank account exists, then the details of bank account gets displayed. If account doesn't exist, then it displays "SEARCH FAILED, ACCOUNT DOES NOT EXISTS".

Step 11: by entering the choice 3, client can deposit the cash into their existing bank account. If bank account exists, then the cash gets deposited and then it displays "SUCCESSFULLY DEPOSITED". If account doesn't exits, then it prints "SEARCH FAILED, ACCOUNT DOES NOT EXISTS".

Step 12: by inputting the choice 4, client can withdraw their cash from their existing bank account. If bank account exists along with the sufficient amount (i.e., amount in the account should be greater than or equal to the requested amount) then they can withdraw the money and it displays "SUCCESSFULL WITHDRAWAL". If account doesn't exists, then it prints "TRANSACTION FAILED".

Step 13: by entering the choice 5, client can view all the bank account details by entering the correct password.

Step 14: When client enters the last choice 6, then it displays " THANK YOU!!!" and then they Automatically gets exited from the program.

Step 15: End.

CHAPTER 4

IMPLEMENTATION

4.1 IMPLEMENTATION OF CONSTRUCTOR

Step 1: create a constructor whose name must be same as its class name. In this project, "bank" is the class name as well as constructor name.

Step 2: constructor can be default constructor or parametrized construction. This project deals with the default constructor in which there is no argument passed.

Step 3: the values for the variables are initialized inside the constructor. String variables are initialized to null and integer variables are initialized to zero.

4.2 IMPLEMENTATION OF INHERITANCE

Step 1: create a super class (parent class) which contains features and variables

Step 2: in this project, features of super class contains the methods to perform bank operations such as opening an account, displays all accounts, search by account number, deposit the money and withdraw the money from the existing account.

Step 3: create the sub class (child class) which inherits the features of super class (parent class). "extends" is the keyword to inherit the features of super class.

Step 4: in the sub class, create an object by using the "new" operator which allocates the memory for the objects and then initialize using constructor and access the member variables.

4.3 IMPLEMENTATION OF EXCEPTION HANDLING

Step 1: in Java try, catch, throw, throws and finally are the five keywords for exception handling. In this project try and catch are the two keywords implemented for the exception handling.

Step 2: code is enclosed within the try block. Try block is followed by the catch clauses which specifies the exception type catch.

Step 3: once the exception occurs in try block, then the cursor comes out of the try block and executes the catch statement.

Step 4: after the execution of catch statement, the program control continues with the next line in the program.

4.4 IMPLEMENTATION OF THE ARRAY OF OBJECTS

Step 1: an array of objects created using 'object'class.

Step 2: the array element stores the location of the reference variables of objects.

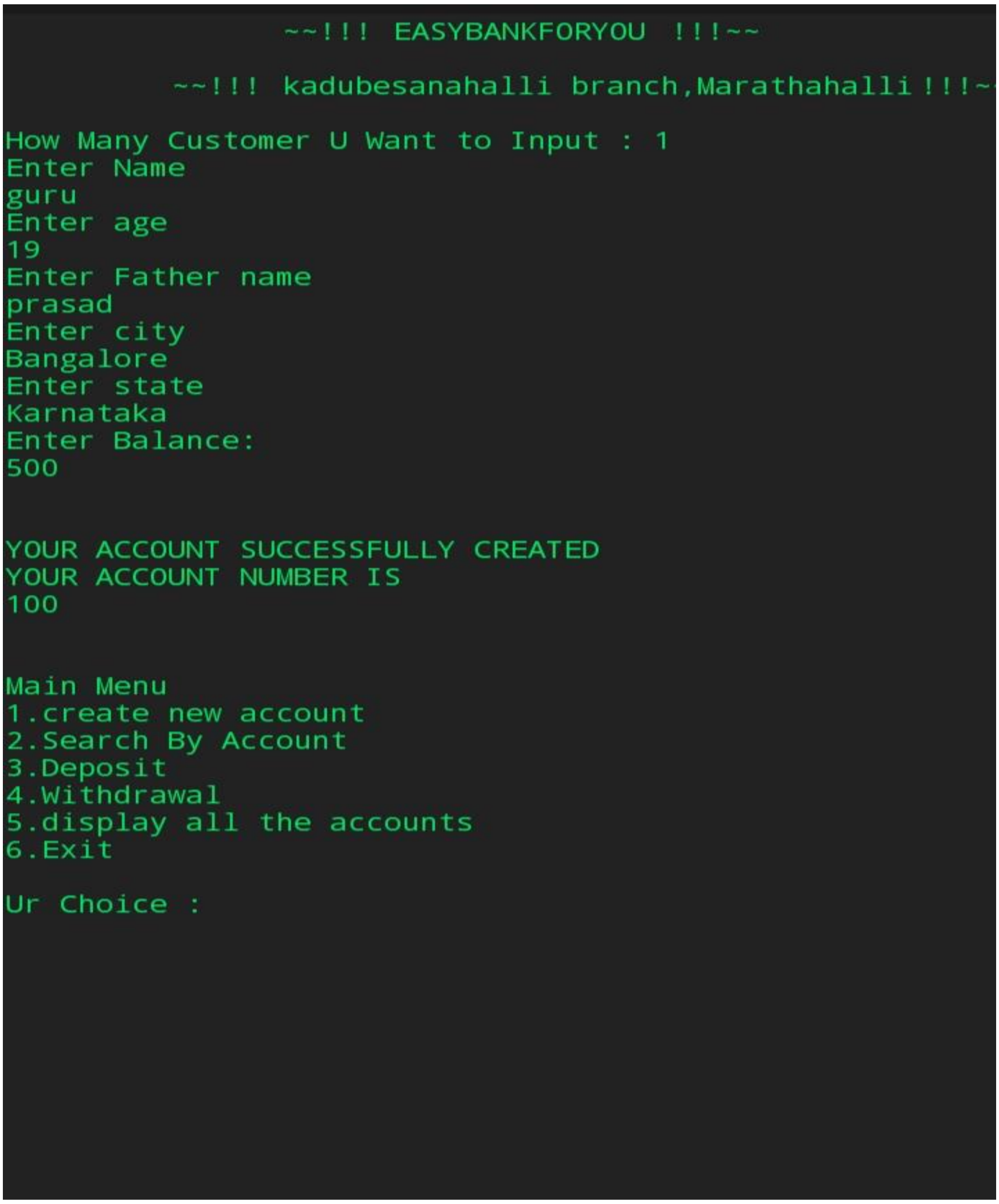
Step 3: the following statement is used to create an array of objects: Class_name objarr[ ];

Step 4: declare and instantiate the array of object : class_name objarr[ ] = new class_name[array_length];

CHAPTER 5

RESULTS

Customer number 1:

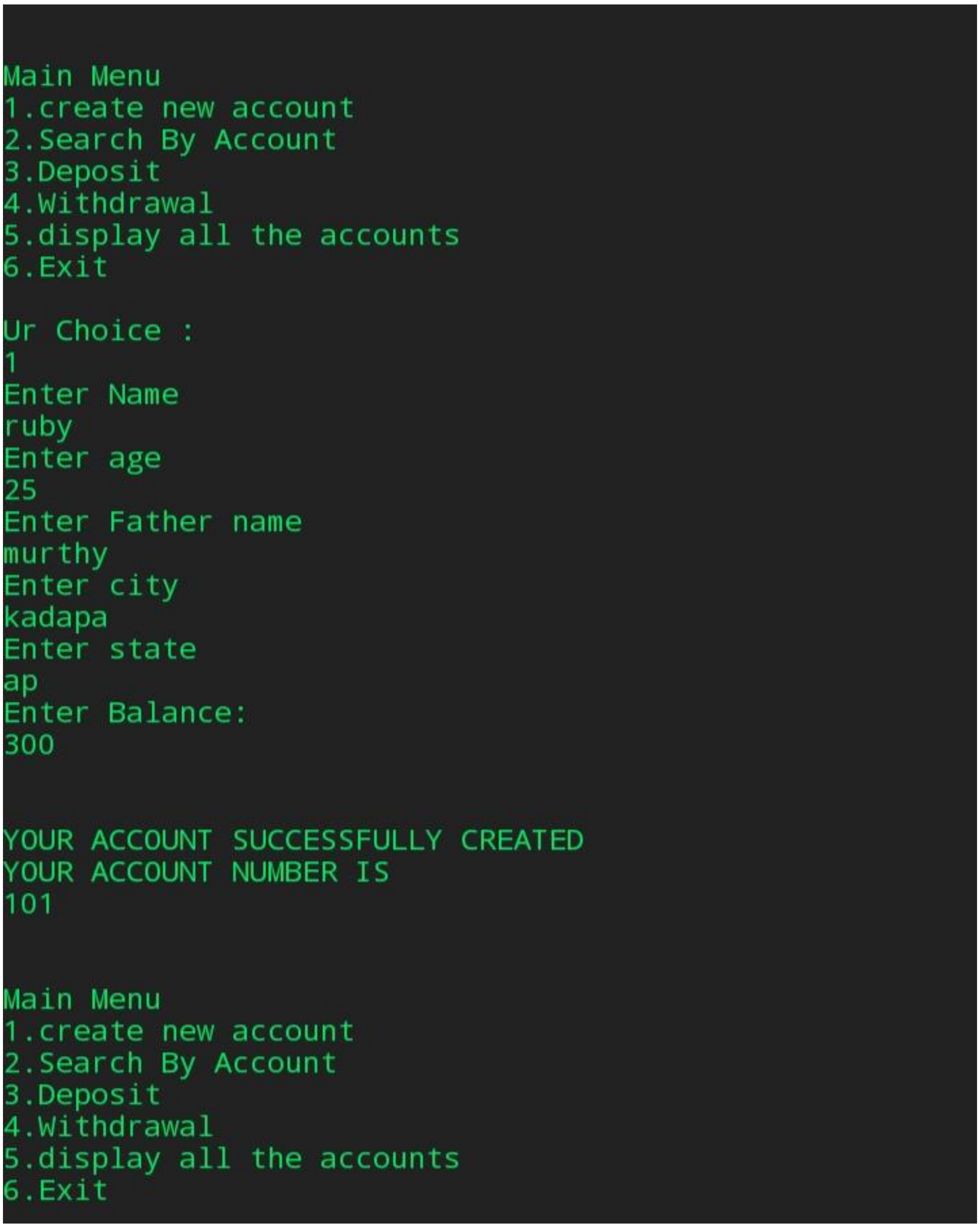


```
                ~~!!! EASYBANKFORYOU !!!~~

          ~~!!! kadubesanahalli branch,Marathahalli!!!~

How Many Customer U Want to Input : 1
Enter Name
guru
Enter age
19
Enter Father name
prasad
Enter city
Bangalore
Enter state
Karnataka
Enter Balance:
500


YOUR ACCOUNT SUCCESSFULLY CREATED
YOUR ACCOUNT NUMBER IS
100


Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
```

Fig 5.1

Initially when customer wants to create the new bank account

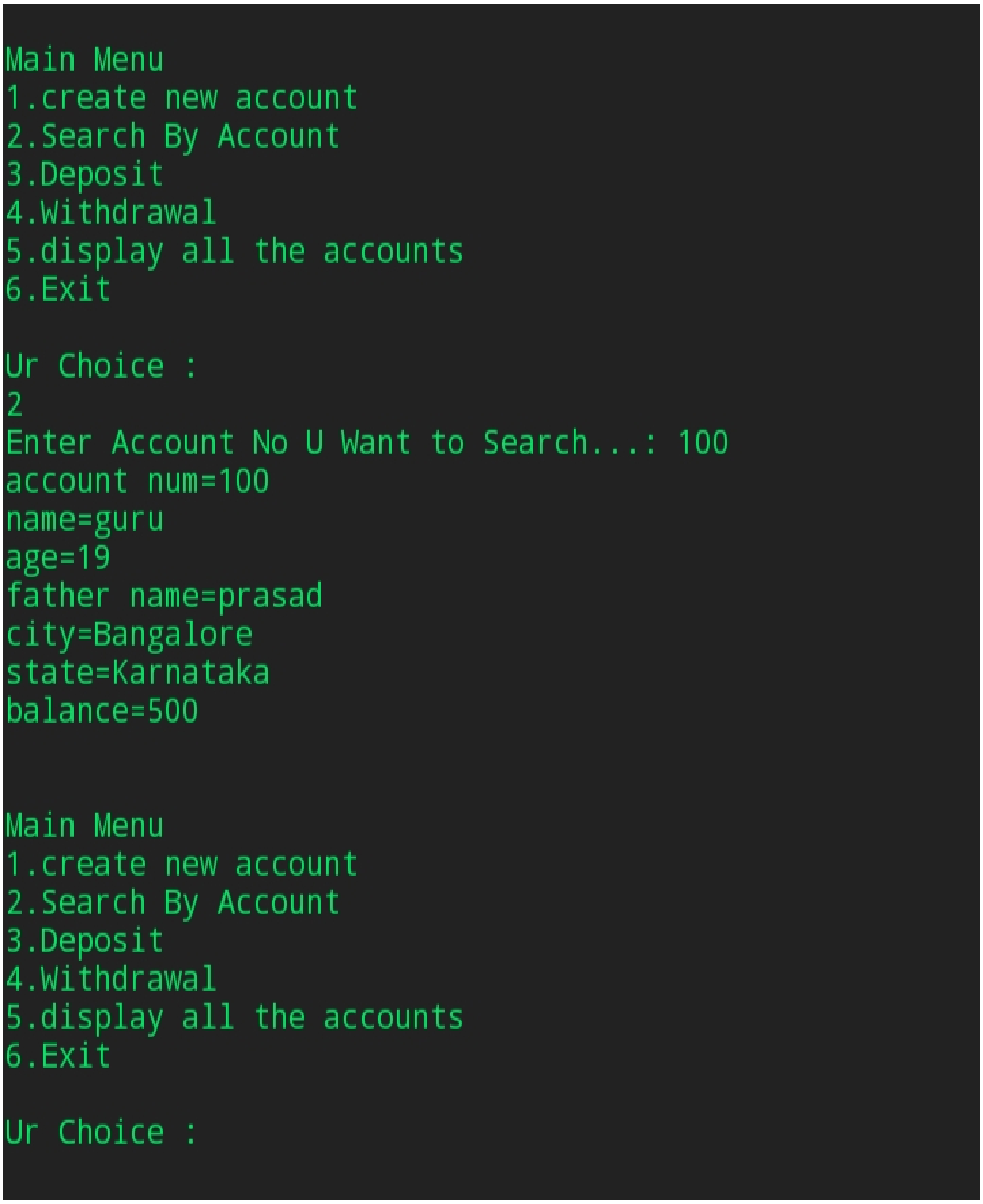
Customer number 2:



```
Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
1
Enter Name
ruby
Enter age
25
Enter Father name
murthy
Enter city
kadapa
Enter state
ap
Enter Balance:
300


YOUR ACCOUNT SUCCESSFULLY CREATED
YOUR ACCOUNT NUMBER IS
101


Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit
```

Fig 5.2

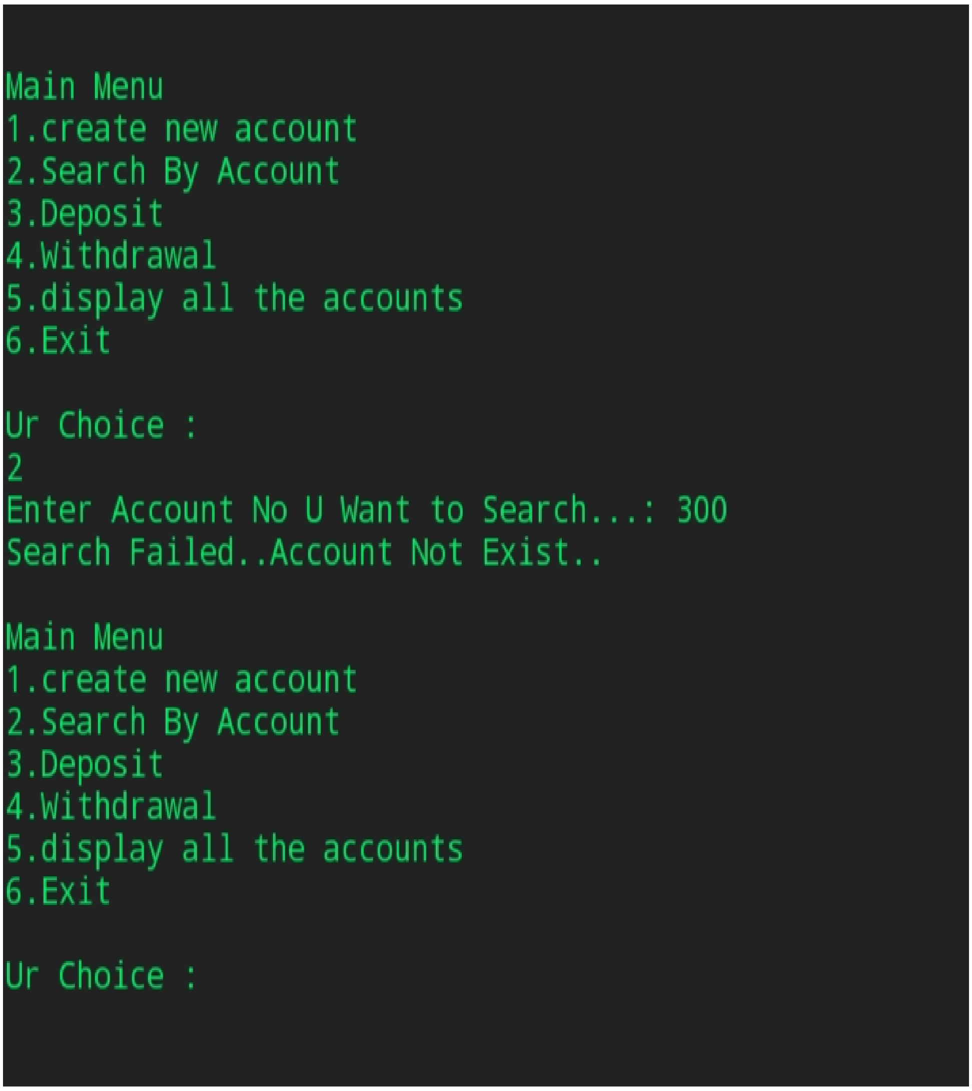
When customer wants to create a new bank account then the customer wants to enter the choice 1



```
Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
2
Enter Account No U Want to Search...: 100
account num=100
name=guru
age=19
father name=prasad
city=Bangalore
state=Karnataka
balance=500


Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
```

Fig 5.3

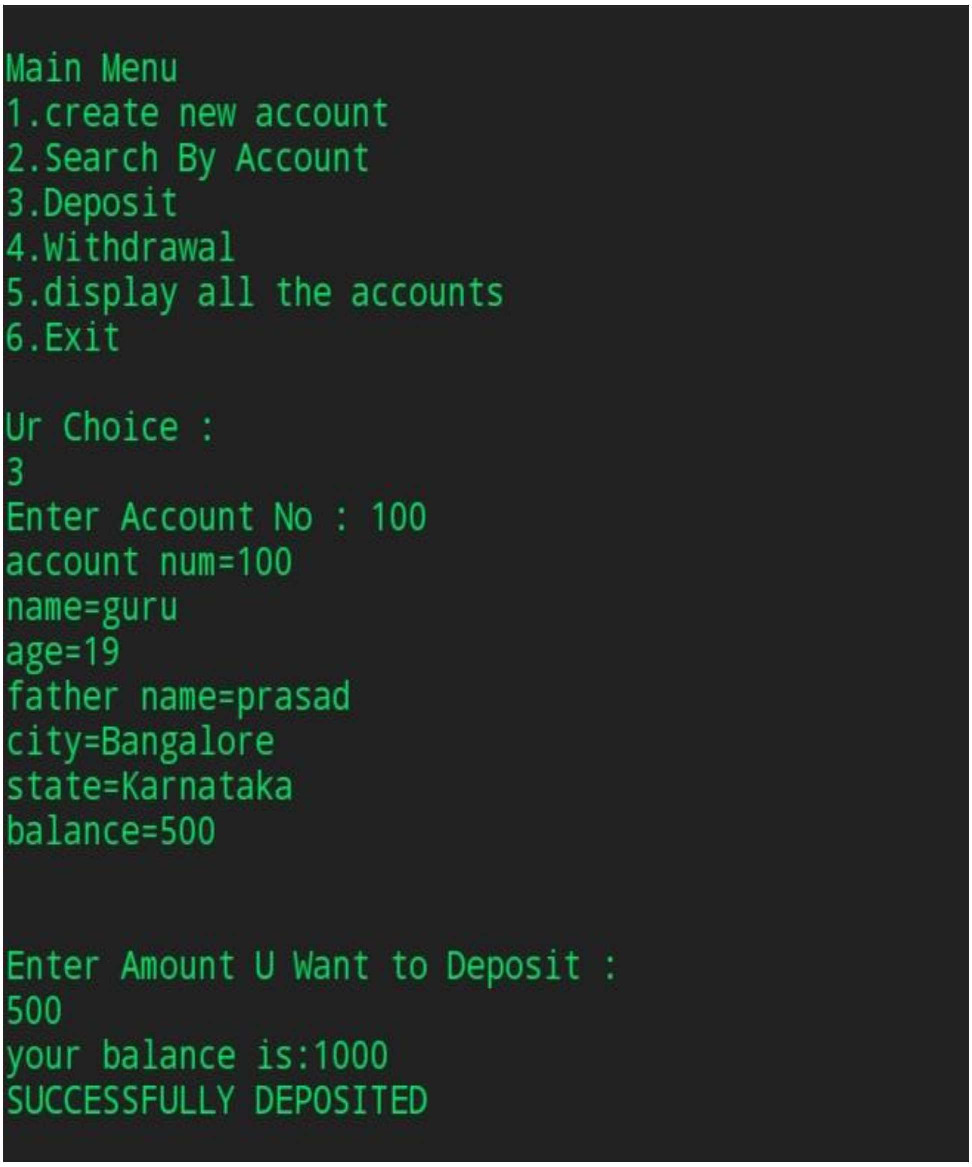
When customer wants to search their bank account then they have to enter the choice 2 and has to enter the account number.If the account number exists then it displays the details of the particular account number.



```
Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
2
Enter Account No U Want to Search...: 300
Search Failed..Account Not Exist..

Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
```

Fig 5.4

When customer enters the Choice 2 to search the account number then they have to enter the account number. If the account number doesn't not exists then it displays search failed. Account does not exists.

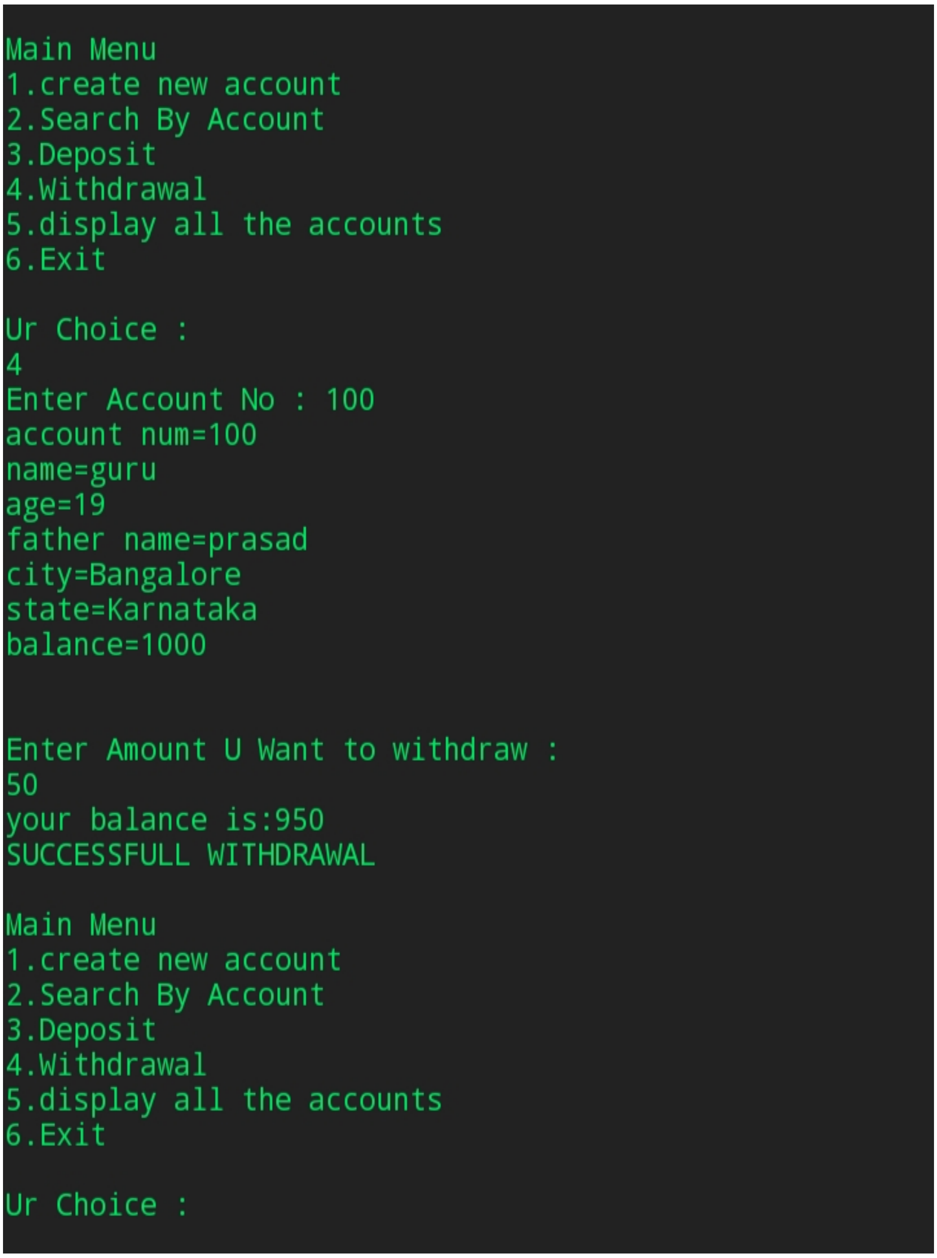


```
Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
3
Enter Account No : 100
account num=100
name=guru
age=19
father name=prasad
city=Bangalore
state=Karnataka
balance=500


Enter Amount U Want to Deposit :
500
your balance is:1000
SUCCESSFULLY DEPOSITED
```

Fig 5.5

When the customer enters the Choice 3 to deposit the money then they have to enter the account number. If the account exists then it asks the customer to enter the amount. After the deposition it displays the current balance and prints

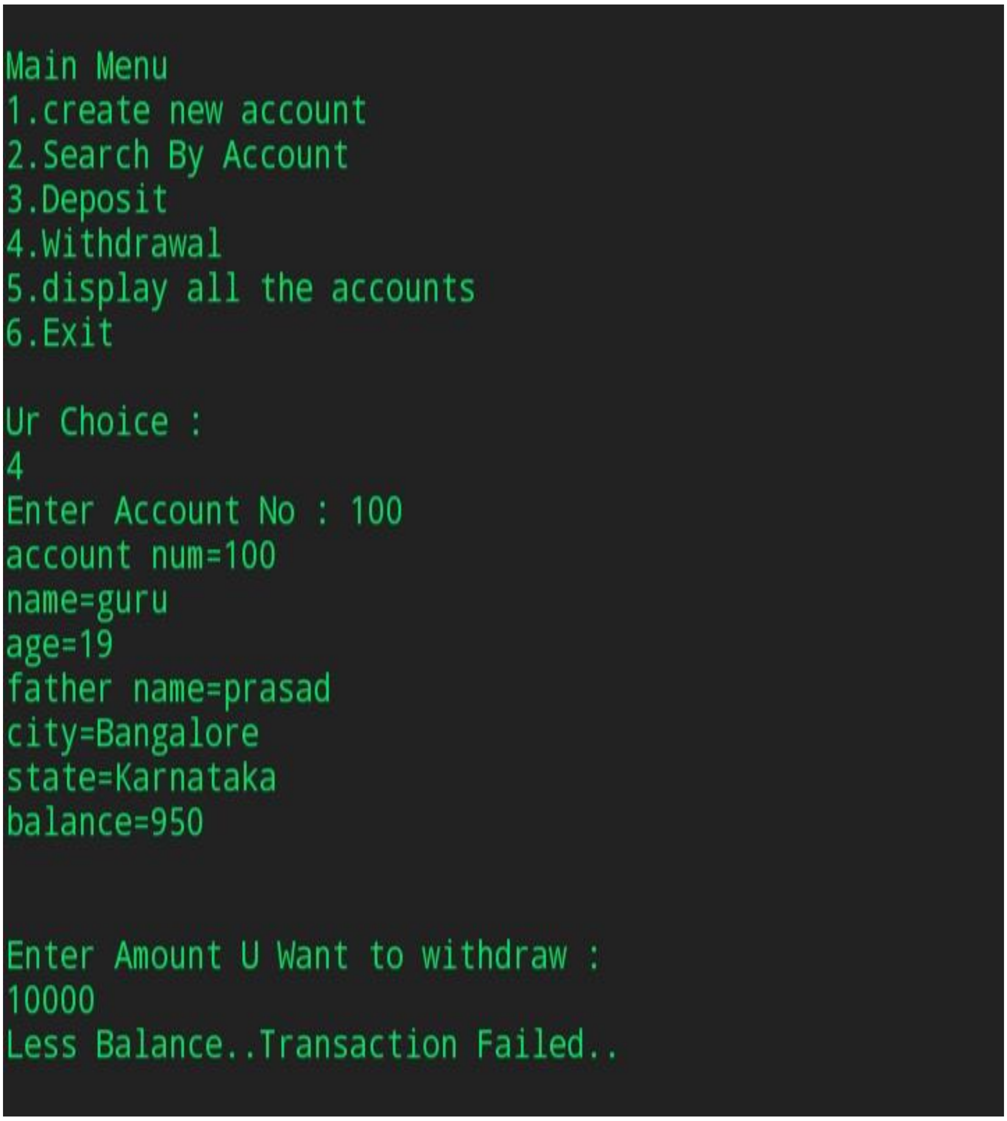
deposited.



```
Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
4
Enter Account No : 100
account num=100
name=guru
age=19
father name=prasad
city=Bangalore
state=Karnataka
balance=1000


Enter Amount U Want to withdraw :
50
your balance is:950
SUCCESSFULL WITHDRAWAL

Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
```

Fig 5.6

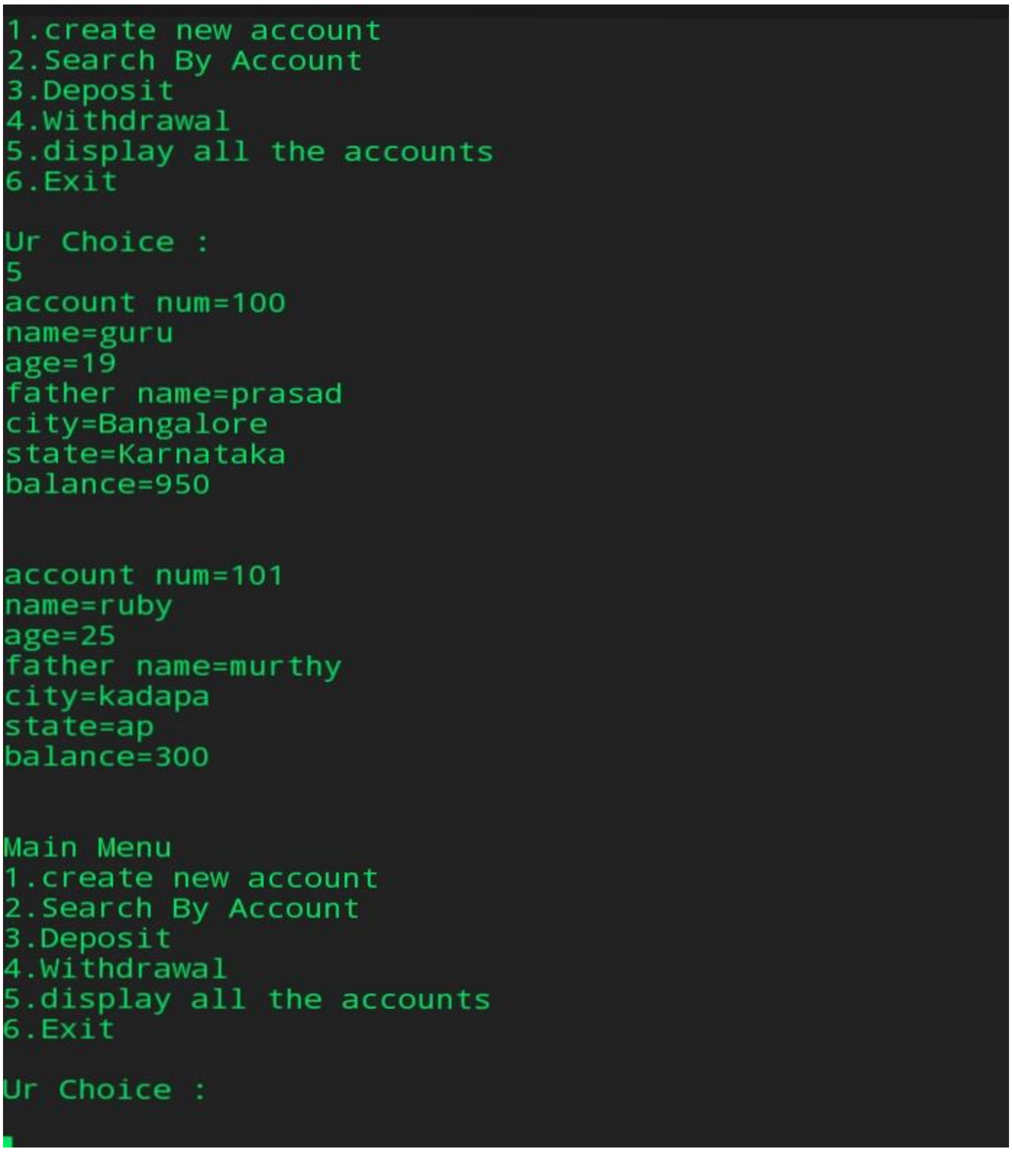
When the customer enters the choice 4 to withdraw the money from their existing bank account. If the withdrawal amount is lesser than their balance then withdrawal takes place successfully and displays the current balance.



```
Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
4
Enter Account No : 100
account num=100
name=guru
age=19
father name=prasad
city=Bangalore
state=Karnataka
balance=950


Enter Amount U Want to withdraw :
10000
Less Balance..Transaction Failed..
```

Fig 5.7

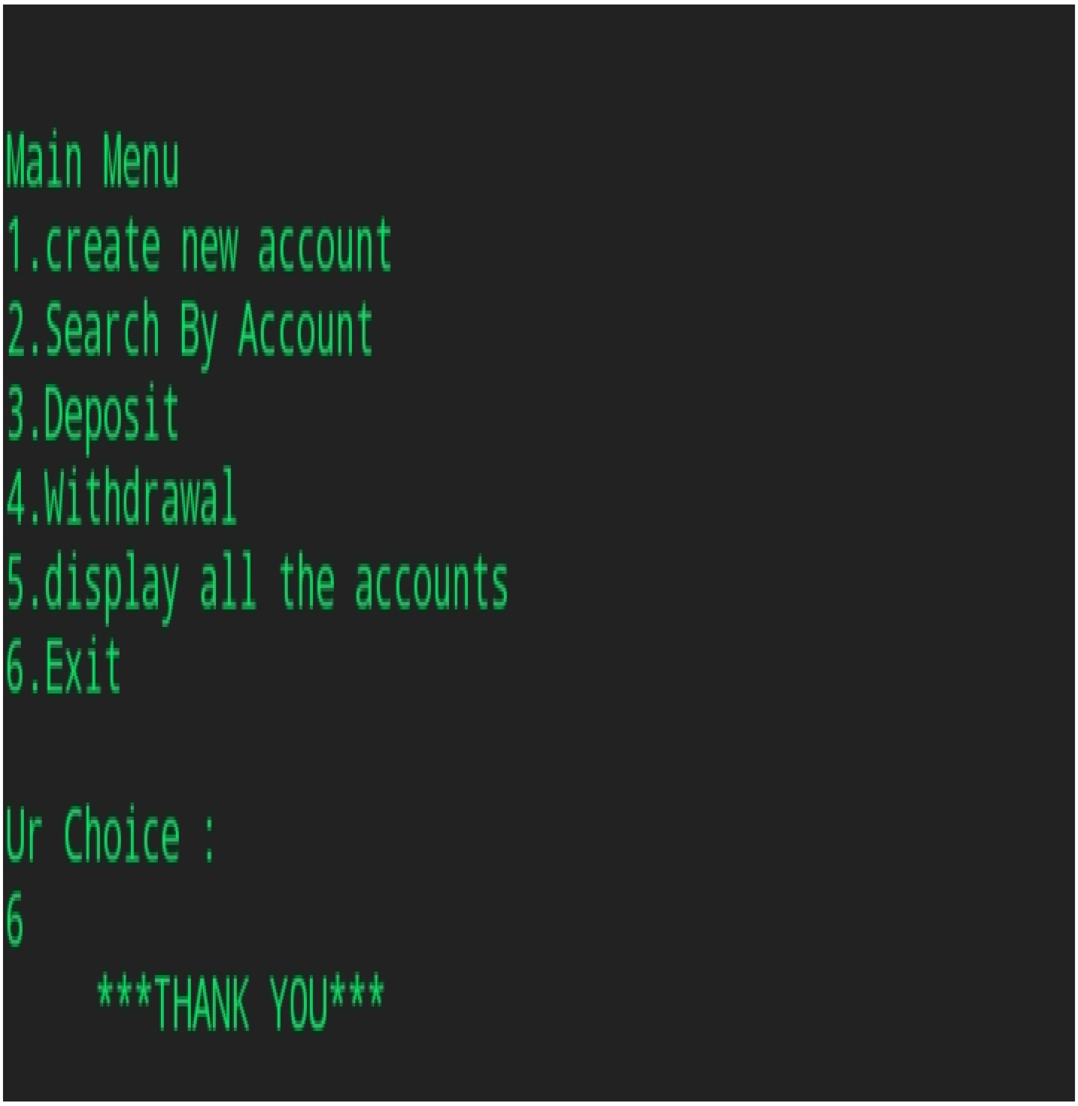
When the customer enters the choice 4 to withdraw the money from their existing bank account. If the withdrawal amount is greater than their balance then it displays less balance, transaction failed.



```
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
5
account num=100
name=guru
age=19
father name=prasad
city=Bangalore
state=Karnataka
balance=950


account num=101
name=ruby
age=25
father name=murthy
city=kadapa
state=ap
balance=300


Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
```

Fig 5.8

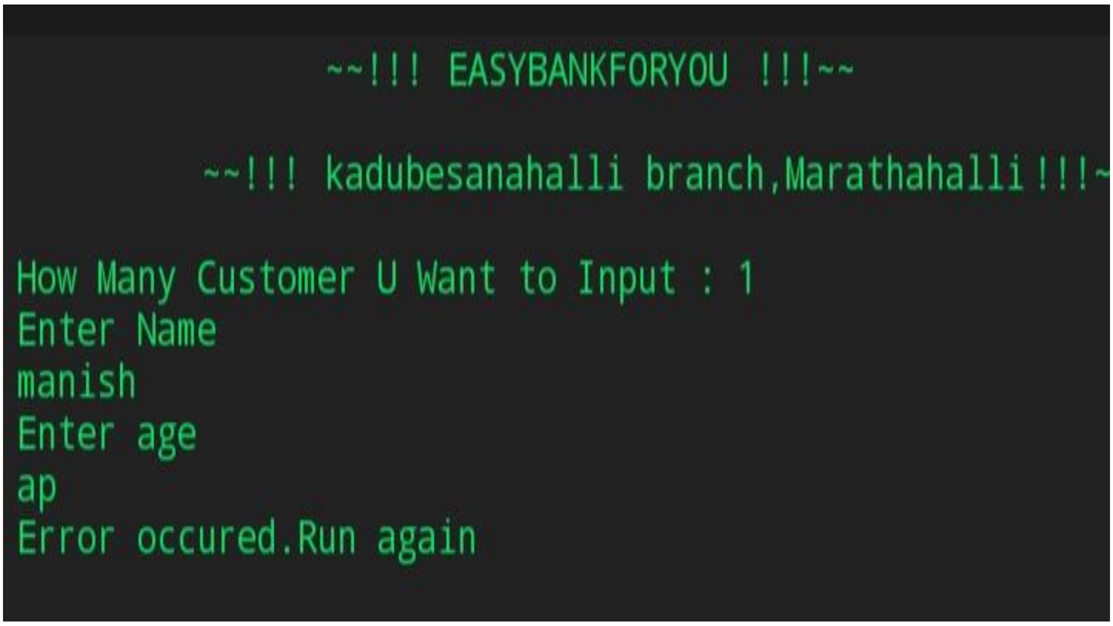
When the customer enters the choice 5 then it displays all the existing account details.



```
Main Menu
1.create new account
2.Search By Account
3.Deposit
4.Withdrawal
5.display all the accounts
6.Exit

Ur Choice :
6
    ***THANK YOU***
```

Fig 5.9

When customer wants to exit then they have to enter the choice 6



```
                 ~~!!! EASYBANKFORYOU !!!~~

         ~~!!! kadubesanahalli branch,Marathahalli!!!~

How Many Customer U Want to Input : 1
Enter Name
manish
Enter age
ap
Error occured.Run again
```

Fig 5.10

When customer enters wrong statement/options then it displays error occurred, run again.

CHAPTER 6

CONCLUSION

This project mainly deals with the e-banking system which is implemented by the object-oriented concepts in Java. It has its own merits and demerits. The main advantage of implementing the e-banking system is to reach the clients (customer) satisfaction. Customers do not have to go the bank in order to access their accounts such as to view the bank balance, deposit the money or to withdraw the money. They can access their Bank account at any time, any place, anywhere.