NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

MORTGAGE UDHAR

A MINI PROJECT

REPORT

Submitted by

JOYDEEP SINGH
1NH18CS220

In partial fulfilment for the award of

the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

Certificate

This is to certify that the mini project work titled

MORTGAGE UDHAR

Submitted in partial fulfilment of the degree of Bachelor of Engineering in Computer Science and Engineering

Submitted by

JOYDEEP SINGH

1NH18CS220

DURING

EVEN SEMESTER 2019-2020

For

19CSE48

Signature of Reviewer   Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner   *Signature with date*

1.____________________________   ___________________________

2.____________________________   ___________________________

ABSTRACT

The concept of Mortgage udhar is to provide a safe, easy, faster and a better to provide loan. The main aim of mortgage is to take the farmers and patients into consideration because they are the ones who need money instantly. It works on the concepts of object oriented programming and the programming language used is java which makes it more secure because the data remains encrypted. The mortgage is just a dummy or demo of how actually loan system works but it also shows how this system can be made easy and user friendly. Lot of websites and mobile applications are already present in market to provide wages to the costumer but they don't have any concern about the background of the client, they just give loan and charge equally. Mortgage udhar provides different loan interests to different categories, so that the true meaning of equality is maintained.

After all the process is done, the client is provided with a loan receipt which has all the details and the amount. No third party is included, its simple like you purchase some items from the mart but the only difference is that you pay at the end of year. Nowadays online loan is getting a lot of attention because no one wants to be in the queues and waste their whole day just to get small loans. Instead of waiting in queues, they prefer online. The major concern is showing fraud details and getting loan in a very easy way but Mortgage udhar has tried to grab all the possible security measures to ensure secure environment for clients and serving them best possible.

Keywords: mortgage udhar, receipt, loan, packages etc

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal NHCE, for his constant support and encouragement.

I am grateful to **Dr. Prashanth C.S.R**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for her constant support.

I express my gratitude to **Ms. Suganya R**, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

JOYDEEP SINGH
1NH18CS220

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 What is Java

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users .Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Java can be used to write applications and applets. A Java application is similar to any other high-level language program: It can only be compiled and then run on the same machine. An applet is compiled on one machine, stored on a server in binary, and can be sent to another machine over the Internet to be interpreted by a Java-aware browser. Java comes with a large library of ready-made classes and objects. The key difference between Java 1.0 and 1.1 was in this library. Similarly, Java 2.0 has a very much larger library for handling user interfaces (Swing by name) but only small changes to the core of the language.

The coding ease defines how simple a programming language is. This holds true for Java as well. Because it has a less ambiguous syntax terminology. Anyone can start right off with Java having the fundamental principles of programming. And the key feature of java is that, it keeps the data encrypted.

1.2 PROBLEM DEFINITION

A recent research by Ipsos group done in different regions of world suggest a need for money increases to sustain the financial needs of different individuals in the area such as the need to support farming, hospital needs and other needs. Many people have chosen to avail loans from underground lending which offer as high as 20% interest rate which makes borrowers sunk from their financial obligations. Hard working society like farmers, who are the backbone of the economy have to also sometimes take loan in order to harvest, grow or transport their crops in order to earn money and make their living. Money needed in hospital for emergency surgery which sometimes is not affordable at that time and all of this, the very irritating thing is that it takes lot of time to get loan from banks or any third party. Waiting in queues for your turn and then providing the documents and sometimes the client forgets the documents at home or some where else, which delays the process further and may result in something unwanted. So

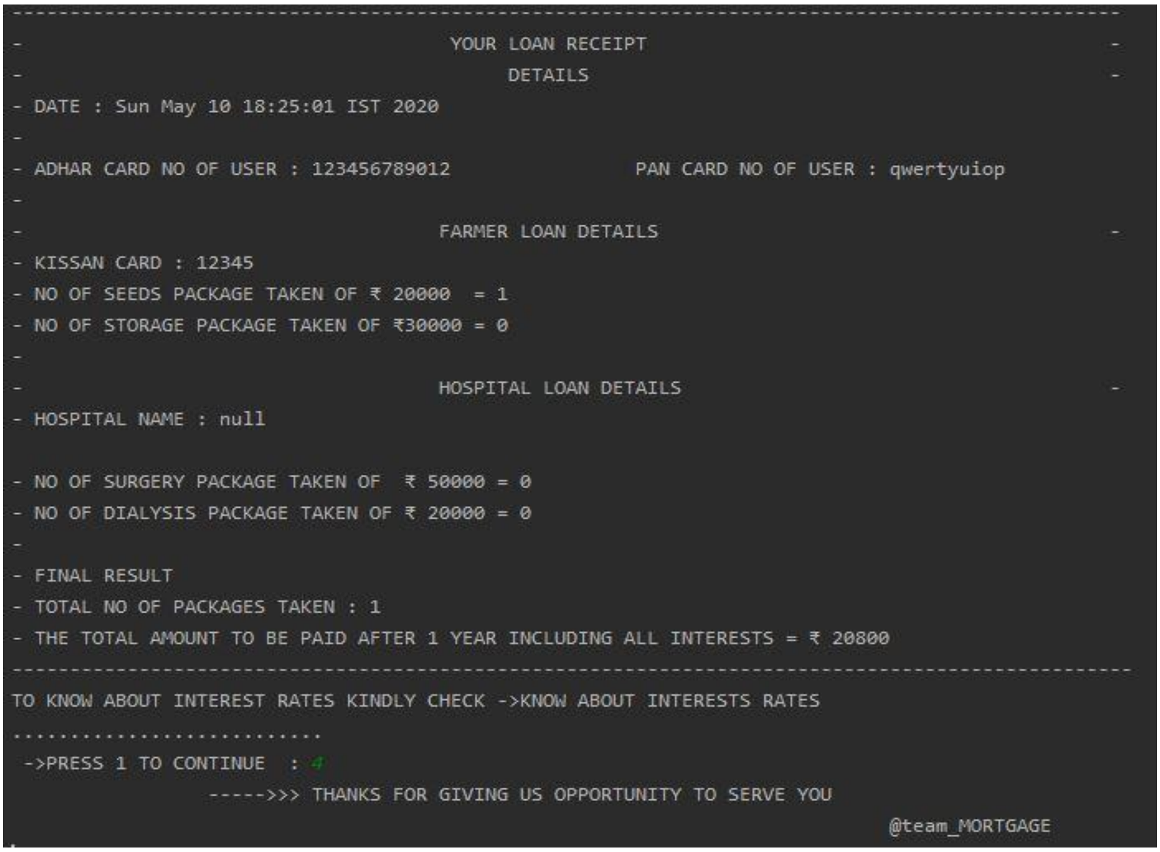
1.3 OBJECTIVES OF THE PROJECT

Processor: X86 compatible processor with 1.7 GHZ clock. The main objectives of the project are:

- Applying object oriented programming concepts to programs.
- To understand and implement classes and objects in real life solving problems.
- Usage of concepts like inheritance, polymorphism, exception handling, i/o functions, multi-threading and packages.
- Comparing and knowing the benefits of object oriented programming over other approach.
- To develop problem solving and debugging skills.
- To improve the technology.
- To increase the knowledge in algorithms.

1.4 EXPECTED OUTCOME

The user enters all the required details for taking loan package or packages and when he selects display option ,user is provided with the receipt that has all the details including the package name, user Aadhar no, user pan card no, reason for the taking loan, total amount of money to be paid including all the interests and the time of purchase.



```
-------------------------------------------------------------------------------------------------
-                                     YOUR LOAN RECEIPT                                         -
-                                          DETAILS                                              -
- DATE : Sun May 10 18:25:01 IST 2020
-
- ADHAR CARD NO OF USER : 123456789012                PAN CARD NO OF USER : qwertyuiop
-
-                                    FARMER LOAN DETAILS                                        -
- KISSAN CARD : 12345
- NO OF SEEDS PACKAGE TAKEN OF ₹ 20000  = 1
- NO OF STORAGE PACKAGE TAKEN OF ₹30000 = 0
-
-                                    HOSPITAL LOAN DETAILS                                      -
- HOSPITAL NAME : null

- NO OF SURGERY PACKAGE TAKEN OF  ₹ 50000 = 0
- NO OF DIALYSIS PACKAGE TAKEN OF ₹ 20000 = 0
-
- FINAL RESULT
- TOTAL NO OF PACKAGES TAKEN : 1
- THE TOTAL AMOUNT TO BE PAID AFTER 1 YEAR INCLUDING ALL INTERESTS = ₹ 20800
--------------------------------------------------------------------------------------------------
TO KNOW ABOUT INTEREST RATES KINDLY CHECK ->KNOW ABOUT INTERESTS RATES
...........................
 ->PRESS 1 TO CONTINUE  : 4
                ----->>> THANKS FOR GIVING US OPPORTUNITY TO SERVE YOU
                                                                              @team_MORTGAGE
```

Fig 1.4 Expected outcome

1.5 REQUIREMENT SPECIFICATION

The system requirements are categories in two parts:

1.4.1 Hardware Requirement

- Processor: Intel core i5 8th Gen
- RAM :512 MB or more
- Hard disk: 500GB
- keyboard

2.2.2 Software Requirements

- IntelliJ idea, eclipse or any IDE.
- Windows 7 or higher.
- JVM.

CHAPTER 2

OBJECT ORIENTED PROGRAMMING

2.1 OBJECT ORIENTED PROGRAMMING

Object-oriented programming (OOP) refers to a type of computer programming (software design) in which programmers define the data type of a data structure, and also .data structure becomes an object that includes both data and functions. In addition, programmers can create relationships between one object and another. For example, objects can inherit characteristics from other objects.

Advantages of Object Oriented Programming

One of the principal advantages of object-oriented programming techniques over procedural programming techniques is that they enable programmers to create modules that do not need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its features from existing objects. This makes object-oriented programs easier to modify.

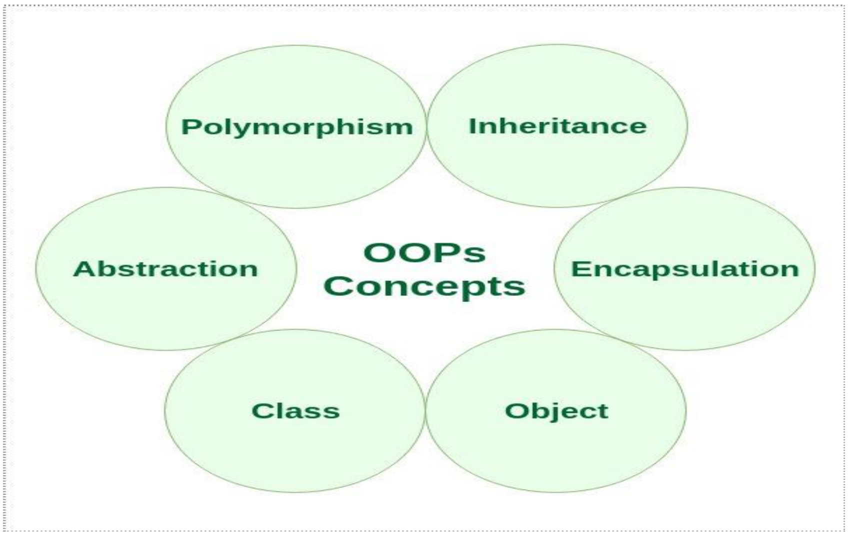


Fig 2.1 object oriented programming

2.2 CLASS

The class is at the core of java. It is the logical construct upon which the entire java language is built because it defines the shape and nature of an object. As such, the class forms the basis for object-oriented programming in java. Any concept you wish to Implement in java program must be encapsulated within a class.

When you define a class, you declare its exact form and nature. You do this by specifying the data that it contains and the code that operates on that data. While very simple classes may contain only code or only data, most real-world classes contain both. As you will see, a class code defines the interface to its data. A class is defined by the use of class keyword. The classes that have been used up to this point are actually very limited examples of its complex form. A simplified general form of a class definition is shown below: -

```
Class classname {
Type instance-variable;
 Type-method name {
//code to be executed
}
}
```

The data, or variables, defined within a class are called instance variables. The code is contained within methods. Collectively, the methods and variables defined within a class are called members of the class. In most cases, the instance variable is acted upon and accessed by the methods defined for that class. Thus, as a general rule, it is the method that determines how a class data can be used.

2.3 OBJECT

As just explained, when we create a class, we are creating a new data type. You can use this type to declare objects of that type. However, obtaining objects of a class is a two- step process. First, you must declare a variable of that class type. This variable does not define an object. Instead, it is simply a variable that can refer to an object. Second, we must acquire an actual, physical copy of the object and assign it to that variable. You can do this using the new operator. The new operator dynamically allocates memory for an object and returns a reference to it. This reference is essentially the address in a memory of the object allocated by new. this reference Is then stored in the variable. Thus, in a java all class object must be dynamically allocated.

Syntax: -

Classname object name = new classname ();

Here classname is the class which object is to be created followed by object name with the help of new operator and classname along with parenthesis is constructor. A constructor defines what occurs when an object of a class is created. Constructor are an Important part of all classes and have many significant attributes. Most real-world cases explicitly define their own constructors within their class definition. However, if no explicit constructor is specified. Then java will automatically supply a default constructor.it is important to understand that new allocates memory for an object during run time. The advantage of this approach is that your program can create as many or as few objects it needs during your execution of these program. However, since memory is infinite it is possible that new will not be able to allocate memory for an object because insufficient memory exists. If this happens a run time error will occur. Let’s once again distinct between a class and a object. A class creates a new data type that can be used to create to objects. That is a class creates a logical framework that defines the relationship between its members. When you declare an object of that class you are creating an instance of that class. Thus, a class is a logical construct. An object has physical reality.

2.4 INHERITANCE

Inheritance is one of the cornerstones of object-oriented programming because it allows the creation of hierarchical classifications. Using inheritance, you can create a general class that defines traits common to a set of related items. This class can then be inherited by other more specific classes each adding those things that are unique to it. In the terminology of java, a class that is inherited is called a superclass. The class that does the inheriting is called a subclass. Therefore, a subclass is a specialized version of a superclass. It inherits all of the members defined by the superclass and adds it own unique elements. To inherits a class, we simply incorporate the definition of one class into another by using the extends keyword. There are various types of inheritance in java. They are described below:

1. **Single inheritance**

   When a class extends another one class only then we call it a single inheritance. The below flow diagram shows that class B extends only one class which is A. Hera A is a parent class of B and B would be a child class of A.

2. **Multiple inheritance**

   Multiple inheritance refers to the concept of one class extending more than one base class. The inheritance we learnt earlier had the concept of one base class or parent. The problem with multiple inheritance is that derived class will have to manage the dependency on two base classes.

3. **Multilevel inheritance**

   Multilevel inheritance refers to a mechanism in OO technology where one can inherit from a derived class, thereby making this derived class the base class for the new class. As you can see in below flow diagram C is a subclass or child class of B and B is a child class of A.

4. **Hierarchical inheritance**

   In such kind of inheritance one class is inherited by many sub classes. In below class B, C and D inherits the same class A.As you can see in the below diagram that when a class has more than one child classes or in other words more than one child classes have the same parent class then this type of inheritance is known as hierarchical inheritance.

5. **Hybrid inheritance**

   In simple terms you can say that Hybrid inheritance is a combination of single and Multiple inheritance. A typical flow diagram would look like below. A hybrid inheritance can be achieved in the java in a same way as multiple inheritance can be. By using interface, we can have multiple as well as hybrid inheritance in java.

Syntax:

```
Class derived-class extends base-class
{
   // methods and fields
}
```

2.5 POLYMORPHISM

Polymorphism means “many forms” and it occurs when we have many classes that are related to each other by inheritance. Like we specified in the previous chapter. Inheritance lets us inherit attributed and methods from another class. Polymorphism uses those methods to perform different tasks. This allows us to perform a single action in different ways. Polymorphism is the capability of a method to do different things based on the object that it is acting upon. In other words, Polymorphism allows you to define one interface and have multiple implementations. As we have seen in the above example that we have the method and have the multiple implementations of it in the different 2 sub classes.

Types of polymorphism:

1. Method overloading in java – This is an example of compile time.
2. Method overriding in java – This is an example of run time.

2.6 ABSTRACT CLASS

There are situations in which you will want to define a superclass that declares the structures of a given abstraction without providing a complete implementation of every method. That is sometimes you will want to create a superclass that only defines a generalized form that will be shared by all of it subclass leaving it to each subclass to fill in the details. Such a class determines the nature of the methods that the subclasses must implement. One way this situation can occur is when a superclass is unable to create a meaningful implementation for a method. This is the case with the class Figure used in the preceding example. The definition of area is simply a placeholder. It will not compute and display the area of any type of object. As you will see as you can create your own libraries it is uncommon for a method to have no meaningful definition in the context of its superclass. You can handle this situation in two ways. One way as shown in the previous example is to simply have it report a warning message. While this approach can be useful in certain situations such as debugging it is not usually appropriate. You may have methods that must be overridden by the subclass in order for the subclass to have any meaning. Consider the class triangle. It has no meaning if area is not defined. In this case, you want some way to ensure that a subclass does, indeed override all necessary methods.

2.7 MULTITHREADING

Java provides built-in support for multithreading programming. A multithread program contains two or more parts that can run concurrently. Each part of such a program is called a thread, and each thread defines a separate path of execution. Thus, multithreading is a specialized form of multitasking. You are almost certainly acquainted with multitasking because it is supported by virtually all modern operating systems. However, there are two distinct types of multitasking process based and thread based. It is important to understand the difference between the two. For many readers process based multitasking is the more familiar form. A process is in essence, a program that is executing. Thus, process-based multitasking is the feature that allows your computer to run two or more programs concurrently. For example, a process-based multitasking enables your t run the java compiler at the same time that you are using a text editor or visiting a website. In process-based multitasking a program is the smallest unit of code that can be dispatched by the scheduler. In a thread based multitasking environment the thread is the smallest unit of dispatchable code. This means that a single program can perform two or more tasks simultaneously. For instance, a text editor can format text at the same time that it is printing as long as these two actions are being performed by two separate threads. Thus, process based multitasking deals with the big picture and thread-based multitasking handles the details. Multitasking threads require less overhead than multitasking processes. Processes are heavyweight tasks that require their own separate address spaces. Inter process communication is expensive and limited. Context switching from one process to another is also costly.

2.8 I/O FUNCTIONS

This chapter introduces one of java most important packages java.io which supports java basic input output system, including file I / O. Support for i/o comes from java core api libraries not from language keywords. In fact, aside from print and println () none of the I/O methods have been used significantly. The reason is simple most real application of java are not text-based console programs. Rather they are graphically oriented programs that rely on one of java graphical user interface frameworks such as swing the awt or javafx, for user interaction or they are web application. Although text-based console programs are excellent as teaching examples, they do not constitute an important use for java in the real world. Also, java support for console I/O is limited and somewhat awkward to use even in simple programs. Text based console I/O is just not that useful in the real-world java programming.java programs perform I/O through streams. A stream is an abstraction that either produces or consume information. A stream is linked to physical devices by the java i/o system. All streams behave in the same manner even if the physical devices to which they are linked differ. Thus, the same i/o classes and methods can be applied to different types of devices. This means that an input stream can abstract many different kinds of input from a disk file, A keyboard or a network socket. Likewise, an output stream may refer to the console a disk file or a network connection. Streams are a clean way to deal with input/output without having every part of your code understand the difference between a keyboard and a network. For example, java implements streams within class hierarchies defined in the java.io package. As you know all java programs automatically import the java. Lang package. This package defines in a class called systems which encapsulates several aspects of the run time environment. For examples using some of its methods you can obtain the current time and setting of the various properties associated with the systems. System also contain three predefined stream variables: in, out and err. These fields are declared as public static and final within System. This means that they can be used by any part of your program and without reference to a specific System object. System. Out refers to the standard output stream. By default, this is console.

2.9 JAVA PACKAGES

The name of each example class was taken from the same name space. This means that a unique name had to be used for each class to avoid name collisions. After a while without some way to manage the name space, you could run out of convenient descriptive names of individual classes. You also need some way to be assured that the name you choose for a class will be a reasonably unique and not collide with class name chosen by other programmers. Imagine a small group of programmers fighting over who gets to use the name foobar as a classname. Or imagine the entire Internet community arguing over who first named a class Espresso. Thankfully java provides a mechanism for partitioning the class name space into more manageable chunks. This mechanism is the package. The package is both a naming and a visibility control mechanism. You can define classes inside a package that are not accessible by code outside that package. You can also define class members that are exposed only to other members of the same package. This allows your classes to have intimate knowledge of each other's but not expose that knowledge to the rest of the world. To create a package is quite easy, simply include a package command as the first statement in a java source file. Any classes declared within that file will belong to the specified package. The package statement defines a name space in which classes are stored. If you omit the package statement the class names are put into default package, which has no name. While the default package is fine for short, sample programs, it is inadequate for real applications. Most of the time, you will define a package for your code. The general form of package is given below:

Package pkg;

Here pkg is the name of the package. For example, the following creates a package called mypackage:

Package mypackage;

Java uses file system directories to store package. For example, the .class files for any classes you declare to be part of mypackage must be stored in a directory called mypackage. Remember that class is significant and the directory name must match the package name exactly. More than one file can include the same package statement. The package statement simply specifies to which package the classes defined in a file belong. It does not exclude other classes in other files from being part of that same package. Most real-world packages are spread across many files. You can create hierarchy of packages. To do so simply separate each package name from the one above it by use of a period. The general form of a multileveled package statement is shown like package pkg1[.pkg2] [.pkg3]. A package hierarchy must be reflected in the file system of your java development system.

2.10 EXCEPTION HANDLING

A java exception is an object that describes an exceptional condition that has occurred in a piece of code. When an exceptional condition arises an object representing that exception is created and thrown in the method that caused the error. That method may choose to handle the exception itself or pass it on Either way at some point the exception is caught and processed. Exception can be generated by the java run time system or they can be manually generated by your code. Exception thrown by java relate to fundamental errors that violate the rules of the java language or the constraints of the java execution environment. Manually generated exception are typically used to report some error condition to the caller of a method. Java exception handling is managed via five keywords: try, catch, throws, throw and finally. Briefly here is how they work. Programs statement that you want to monitor for exception are contained within a try block. If an exception occurs within a try block. If an exception occurs it is thrown. Your code can catch this exception and handle it in some relational manner. System generated exception are automatically thrown by the java run time system. To manually throw an exception use the keyword throw. Any exception that is thrown out of a method must be a specified as such by a thrown clause. Any code that absolutely must be executed after a try block completes is put in a finally block. All exception types are subclasses of the built-in class Throwable. Thus, Throwable is at the top of the exception class hierarchy. Immediately below Throwable are two subclasses that partition exceptions into two distinct branches.

CHAPTER 3

DESIGN

Pseudo code gives, out the partial required outcomes, which helps to insert new ideas and logics which in turn, help to get and trace the efficient logic to implement later on according to usage and requirement.

3.1 ALGORITHM

- First, we have to declare the variables in the class.
- Create one main class in which we are going to call all other methods by using objects.
- With the help of object, it will call a method which will be having the switch condition.
- By the help of switch condition, it will access my other methods.
- If the client goes into farmer the count is increased by default.
- And we have created another class where we will perform inheritance.
- We will use abstract class as well so that we will declare the method and we will define in another classes.
- We have used method overriding to rewrite the cost of farmer method.
- The receipt is displayed when the client selects display.
- Receipt shows all the packages taken with all the client details, amount to be paid and the date and time of purchase.

3.2 FLOWCHART

This flowchart gives us the client-side view clearly.

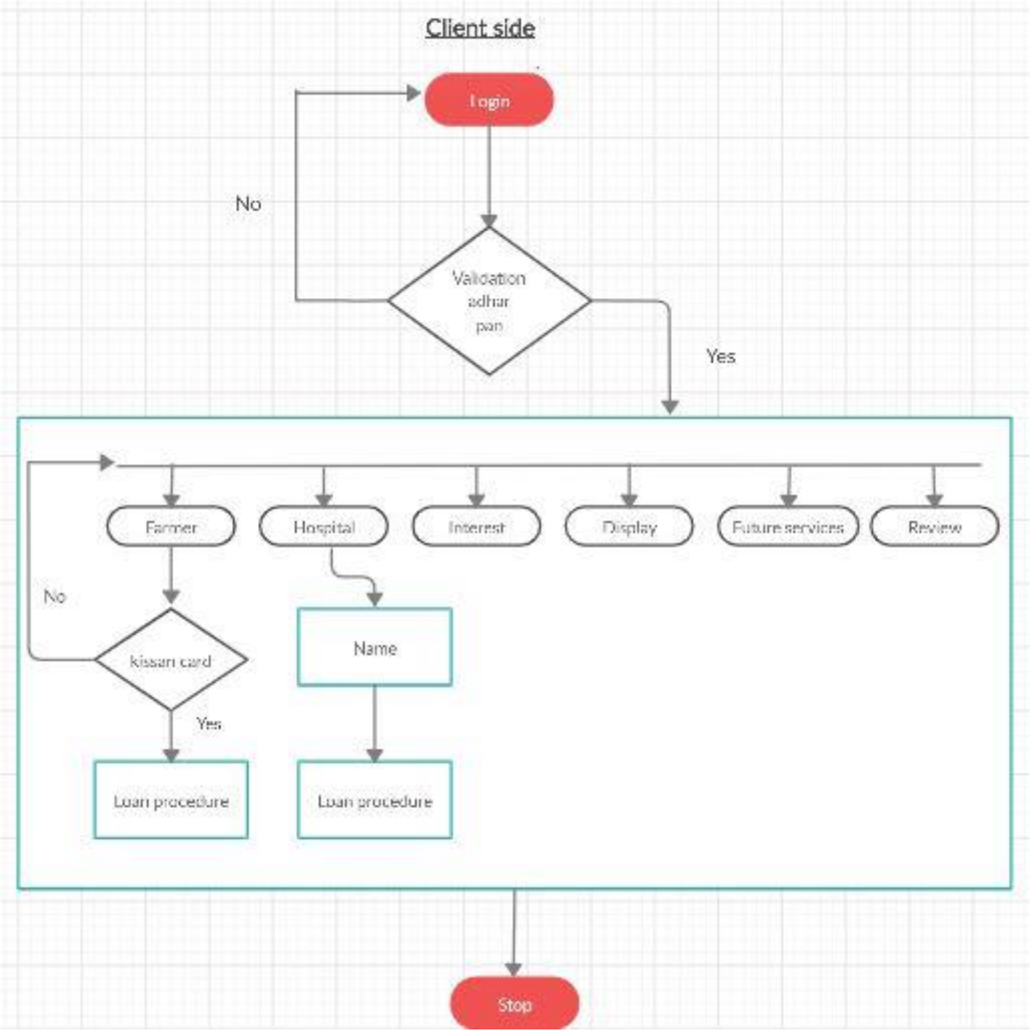


Fig 3.2 Flowchart 1

CHAPTER 4

IMPLEMENTATION

4.1 MODULE 1 FUNCTIONALLITY

This module has been created in a package pro and it has an abstract class q. All the variables are dropped as initialization.

```
package pro;
abstract class q{
String s="WELCOME TO MORTGAGE ";
abstract void display();
abstract void hospital();
String r="REQUEST ACCEPTED";
String adhar="123456789012";
String pan="qwertyuiop";
String kissan="12345";
static String hospi;
public Date paymentDate;
Scanner hos=new Scanner(System.in);
int tol_far_loan =0, tol_sergery =0,count=0,amount=0, trans_store =0, no_of_seed =0, tol_hos_loan =0, tol_dailysis=0;
void Farmer()
{
System.out.println("-------------"+s+"-------------");
System.out.println("PLEASE ENTER YOUR CHOICE ");
System.out.println("----->");

System.out.println("      1. SEEDS AND PESTICIDE LOON ");

System.out.println("      2. TRANSPORTATION AND STORAGE LOON ");

Scanner f=new Scanner(System.in);

int i=f.nextInt();

switch(i)

{

case 1: System.out.println(r); no_of_seed++; amount=amount+1000; count++;

break; case 2:

System.out.println(r); trans_store++; amount=amount+2000; count++;

break;

}

}

}
```

4.2 MODULE 2 FUNCTIONALITY

Here another class to represent the concept of over ridding and the concept of overloading and threading concept.

```
class r extends q{

String x

void Farmer()

{

paymentDate=new Date();

String s="WELCOME TO THE FARMER'S PORTAL ";
String ss=s.concat(" YOU ARE THE BACKBONE OF OUR COUNTRY ");
System.out.println(super.s);
System.out.println(ss);
System.out.println("PLEASE ENTER YOUR CHOICE");
System.out.println("----->");
System.out.println("      1. SEEDS AND PESTICIDE LOON        -> ₹ 20000 ");
System.out.println("      2. TRANSPORTATION AND STORAGE LOON -> ₹ 30000");
Scanner f=new Scanner(System.in);
int i=f.nextInt();
switch(i)
{
case 1: System.out.println(super.r);
if(no_of_seed==2)
{
System.out.println("YOU HAVE ALREADY CROSSED THE LIMIT TO TAKE LOAN ");
System.out.println("     KINDLY PLEASE TRY FOR OTHER LOANS   ");
}
else {
no_of_seed++;
amount = amount + 20000+800;
count++;
}
break;
case 2: System.out.println(super.r);

if(trans_store==2)

{

System.out.println("YOU HAVE ALREADY CROSSED THE LIMIT TO TAKE LOAN ");

System.out.println("      KINDLY PLEASE TRY FOR OTHER LOANS   ");

}

else {

trans_store++;

amount = amount + 30000+1200;

count++;

}

break;

}

}

void hospital()

{

System.out.println(super.s);

paymentDate=new Date();

System.out.println(" ");

System.out.println("PLEASE ENTER YOUR HOSPITAL NAME");

hospi=hos.next();

System.out.println("ENTER YOUR CHOICE");

System.out.println("----->");
System.out.println("1.    SURGERY LOAN  -> ₹ 50000");

System.out.println("2.    DIALYSIS LOAN -> ₹ 20000");

Scanner f=new Scanner(System.in);

int i=f.nextInt();

switch(i)

{

case 1: System.out.println(super.r);

if(tol_sergery==2)

{

System.out.println("YOU HAVE ALREADY CROSSED THE LIMIT TO TAKE LOAN ");

System.out.println("      KINDLY PLEASE TRY FOR OTHER LOANS   ");

}

else {

tol_sergery++;

amount = amount + 50000+3000;

count++;

}

break;

case 2: System.out.println(super.r);

if(tol_dailysis==2)

{

System.out.println("YOU HAVE ALREADY CROSSED THE LIMIT TO TAKE LOAN ");

System.out.println("      KINDLY PLEASE TRY FOR OTHER LOANS   ");
}
else {
tol_dailysis++;
amount = amount + 20000+1200;
count++;
}
break;
}}
void review()
{
Random r=new Random();
int p=(int) (5*r.nextDouble());
if(p==1)
System.out.println("feedback is ::BAD");
else if(p==2)
System.out.println("feedback is ::OK");
else if(p==3)
System.out.println("feedback is ::SATISFACTION");
else if(p==4)
System.out.println("feedback is ::GOOD");
else if(p==5)
System.out.println("feedback is ::EXCELLENT");
else if(p==0)
```

```
System.out.println("NEED MORE IMPROVEMENT");

}

void display()

{

System.out.println("------------------------------------------------------------------------------------------
-----");

System.out.println("-                               YOUR LOAN RECEIPT                              -");

System.out.println("-                                   DETAILS                                  -");

System.out.println("- DATE : "+paymentDate);

System.out.println("-");

System.out.println("- ADHAR CARD NO OF USER : "+adhar+   "  " +"    "    +   "    "   +"   "+"
" +  "   " +"PAN CARD NO OF USER : "+pan );

System.out.println("-");

System.out.println("-                              FARMER LOAN DETAILS                               -
");

if(no_of_seed>0 || trans_store>0)

{

System.out.println("- KISSAN CARD : "+kissan);

}

else

{

System.out.println("- KISSAN CARD : NULL");

}

System.out.println("- NO OF SEEDS PACKAGE TAKEN OF ₹ 20000  = "+ no_of_seed);
System.out.println("- NO OF STORAGE PACKAGE TAKEN OF ₹30000 = "+ trans_store);
System.out.println("-");
System.out.println("-                    HOSPITAL LOAN DETAILS                    -");
System.out.println("- HOSPITAL NAME : "+hospi                                  +"\n" );
System.out.println("- NO OF SURGERY PACKAGE TAKEN OF  ₹ 50000 = "+ tol_sergery);
System.out.println("- NO OF DIALYSIS PACKAGE TAKEN OF ₹ 20000 = "+ tol_dailysis);
System.out.println("-");
System.out.println("- FINAL RESULT  ");
System.out.println("- TOTAL NO OF PACKAGES TAKEN : "+count);
System.out.println("- THE TOTAL AMOUNT TO BE PAID AFTER 1 YEAR INCLUDING ALL INTERESTS = ₹ "+amount);
System.out.println("------------------------------------------------------------------------------------------------");
System.out.println("TO KNOW ABOUT INTEREST RATES KINDLY CHECK ->KNOW ABOUT INTERESTS RATES");
}
void display(String i)
{
x=i;
System.out.println("THE UPCOMING SERVICES "+x);
}
void startpoint()
```

```
{
int y;
int f;
System.out.println();
System.out.println("---------------WELCOME TO MORTGAGE-------------- ");
System.out.println("          WHERE WE UNDERSTAND YOUR NEEDS ");
System.out.println();
System.out.println("PLEASE ENTER YOUR DETAILS BELOW -> ");
Scanner input=new Scanner(System.in);
System.out.println(" ");
do
{
System.out.println(" ENTER YOUR ADHAR CARD NO:");
String ad = input.next();
System.out.println(" ENTER YOUR PAN CARD NO  :");
String pa = input.next();
if(ad.equals(adhar)&&pa.equals(pan))
{
System.out.println("DETAILS VERIFIED");
f=0;
}
else
{
```

```
System.out.println("INVALID DETAILS");

System.out.println("-------ENTER AGAIN---------");

f = 1;

System.out.println(" ");

}

}while(f==1);

do

{

System.out.println("HERE IS A LIST :");

System.out.println("------>");

System.out.println("       1. FARMER'S LOAN");

System.out.println("       2. HOSPITAL LOAN");

System.out.println("       3. KNOW ABOUT INTEREST RATES");

System.out.println("       4. DISPLAY RECEIPT ");

System.out.println("       5. FUTURE COMING SERVICES");

System.out.println("       6. REVIEW");

//System.out.println("");

Scanner ch=new Scanner(System.in);

System.out.print("ENTER YOUR CHOICE HERE: ");

int a=ch.nextInt();

switch(a)

{

case 1:
```

```
System.out.println("ENTER YOUR KISSAN CARD NO");

String kiss=hos.next();

if(kiss.equals(kissan))

{

Farmer();

}

else {

System.out.println("WRONG KISSAN CARD ENTERED");

System.out.println("****  PLEASE START AGAIN FOR SECURITY REASONS  *****");

}

break;

case 2: hospital();

break;

case 3:ok k=new ok();

k.interest();

break;

case 4: display();

break;

case 5:

display(" EDUCATIONAL LOAN");

break;

case 6:

review();
```

```
break;

default: System.out.println("DO YOU WANT TO QUIT"+ " -------> PRESS 2 TO QUIT ");

}

Scanner r=new Scanner(System.in);

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

System.out.print(" ->PRESS 1 TO CONTINUE  : ");

y=r.nextInt();

}

while(y==1);

}}

class th1 extends Thread{ void det()

{

System.out.println("              MISSIONS");

System.out.println("TO PROVIDE BEST POSSIBLE HELP FOR NEEDY");

try{

Thread.sleep(100);

}

catch(InterruptedException e)

{

System.out.println(e);

}

}

public void run()
```

```
{
det();
}
}
class th2 extends Thread{
void show()
{
System.out.println("                 ----->>> THANKS FOR GIVING US OPPORTUNITY TO SERVE
YOU                        ");
System.out.println("                                                                            @team_MORTGAGE
");
try{
Thread.sleep(10000);
}
catch(InterruptedException e)
{
System.out.println(e);
}
}
public void run()
{
show();
}
}
```

4.3 MODULE 3 FUNCTIONALITY

This is the main class

```
class project {

public static void main(String[] arg) {

FileInputStream instream = null;

FileOutputStream outstream = null;

try {

File infile = new File("D:\\pro.project\\filein.txt");

File outfile = new File("D:\\pro.project\\fileout.txt");

instream = new FileInputStream(infile);

outstream = new FileOutputStream(outfile);

byte[] buffer = new byte[1024];

int len;

while ((len = instream.read(buffer)) > 0) {

outstream.write(buffer, 0, len);

}

instream.close();

outstream.close();

System.out.println("        LOOKING FOR SOME MONEY            " + "\n" + "       YOU HAVE COME TO RIGHT PLACE");

} catch (IOException ioe) {

ioe.printStackTrace();

}
th1 tread1 = new th1();

//  th2 tread2 = new th2();

tread1.start();

r obj = new r();

obj.startpoint();

th2 tread2 =new th2();

tread2.start();

}

}
```

4.4 MODULE 4 FUNCTIONALITY

Another package

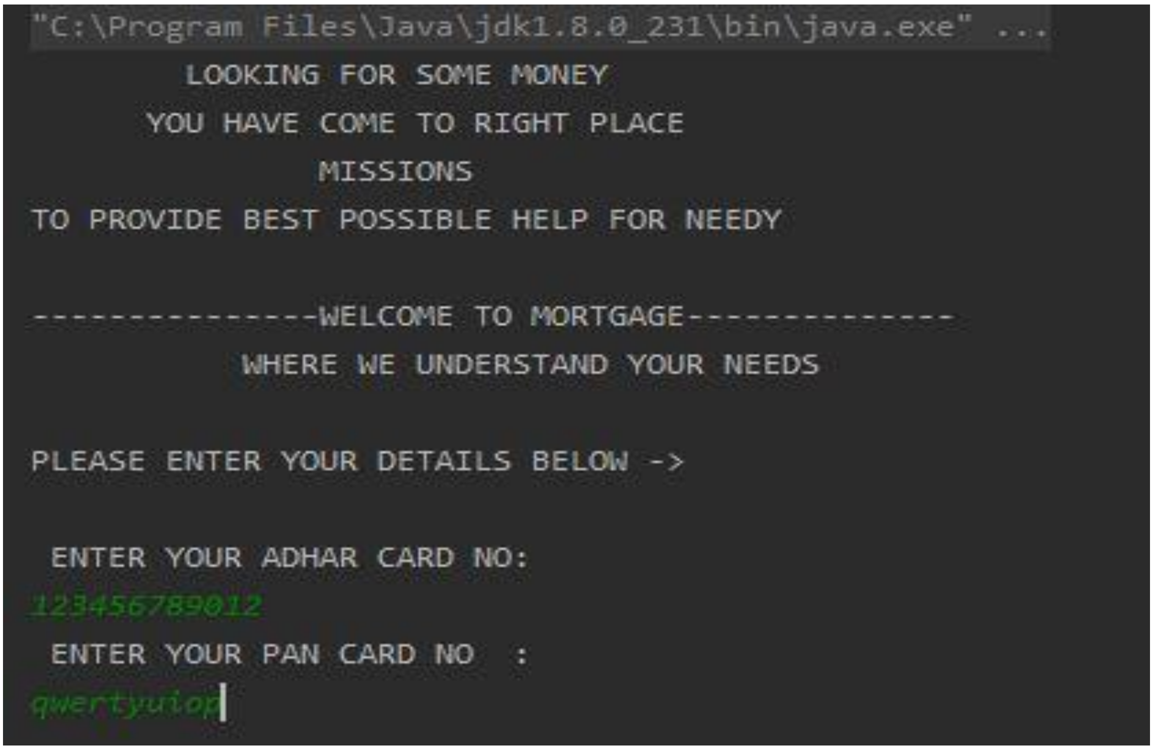
```
package ject;

public class ok

{

public void interest()

{

System.out.print("THE INTEREST RATES ON ALL FARMER PACKAGES PER YEAR IS   : ");

System.out.println("4%");

System.out.println("--------------------");

System.out.print("THE INTEREST RATES ON ALL HOSPITAL PACKAGES PER YEAR IS : ");

System.out.println("6%");

}

}
```

CHAPTER 5

SAMPLE OUTPUT

5.1 LOGIN

It is main log in page where client enter adhar card and pan card no in order to get access inside it. If the inputs don't match, the client has to again enter the details until access granted.

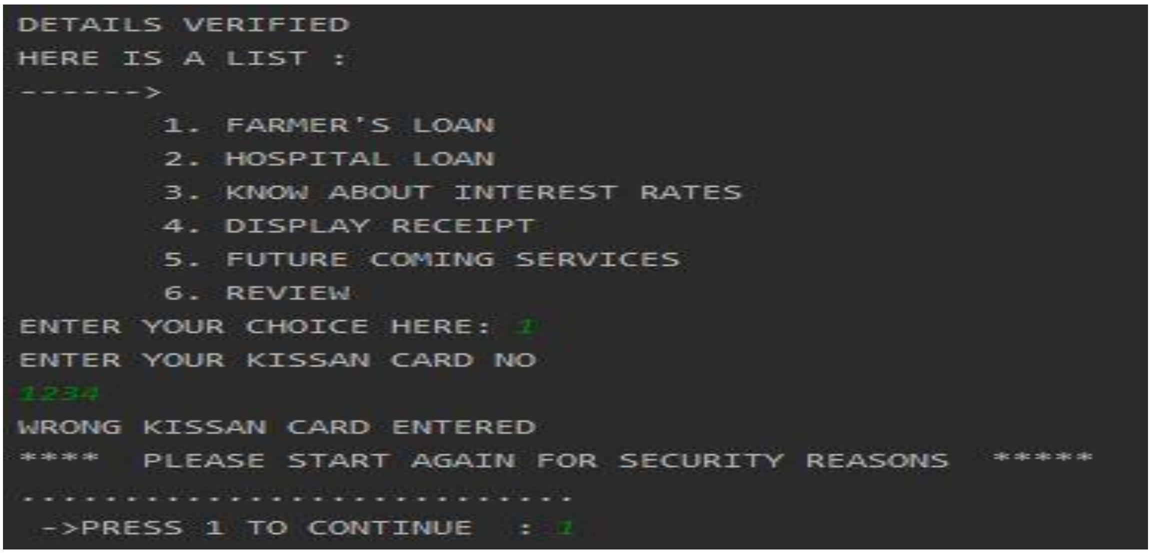


```
"C:\Program Files\Java\jdk1.8.0_231\bin\java.exe" ...
        LOOKING FOR SOME MONEY
      YOU HAVE COME TO RIGHT PLACE
               MISSIONS
TO PROVIDE BEST POSSIBLE HELP FOR NEEDY

---------------WELCOME TO MORTGAGE--------------
           WHERE WE UNDERSTAND YOUR NEEDS

PLEASE ENTER YOUR DETAILS BELOW ->

 ENTER YOUR ADHAR CARD NO:
123456789012
 ENTER YOUR PAN CARD NO  :
qwertyuiop
```

Fig 5.1 Login

5.2 FARMER'S LOCK

Farmer loan provides loan with interest rate of 4%, So the client thinks of taking this package in order to payback with less interest but due to farmer's lock, the client has to enter the kissan card no.

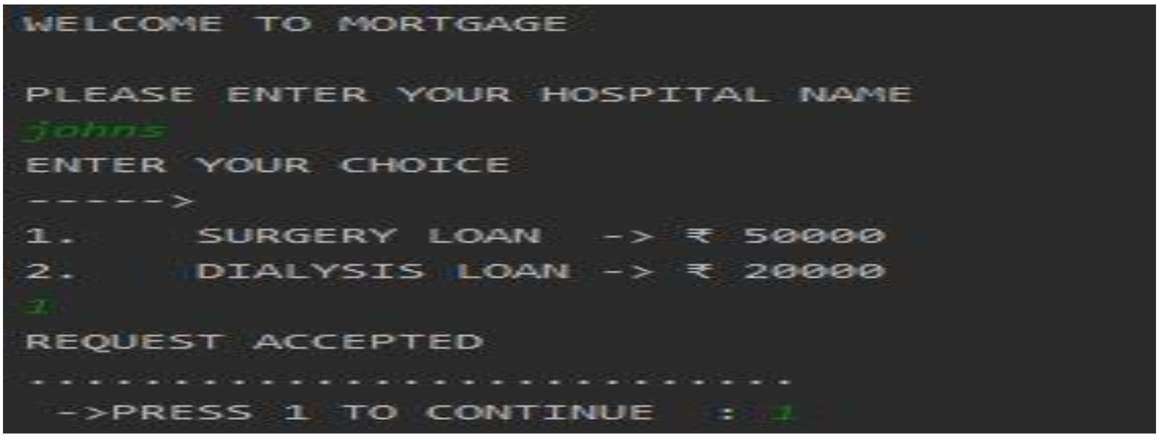


```
DETAILS VERIFIED
HERE IS A LIST :
------>
       1. FARMER'S LOAN
       2. HOSPITAL LOAN
       3. KNOW ABOUT INTEREST RATES
       4. DISPLAY RECEIPT
       5. FUTURE COMING SERVICES
       6. REVIEW
ENTER YOUR CHOICE HERE: 1
ENTER YOUR KISSAN CARD NO
1234
WRONG KISSAN CARD ENTERED
****  PLEASE START AGAIN FOR SECURITY REASONS  *****
...........................
 ->PRESS 1 TO CONTINUE  : 1
```

Fig 5.2 Farmer's lock

5.3 HOSPITAL LOAN PORTAL

This window asks for the client's hospital name and the packages available. The client can purchase these loan packages with the interest rate of 6%.

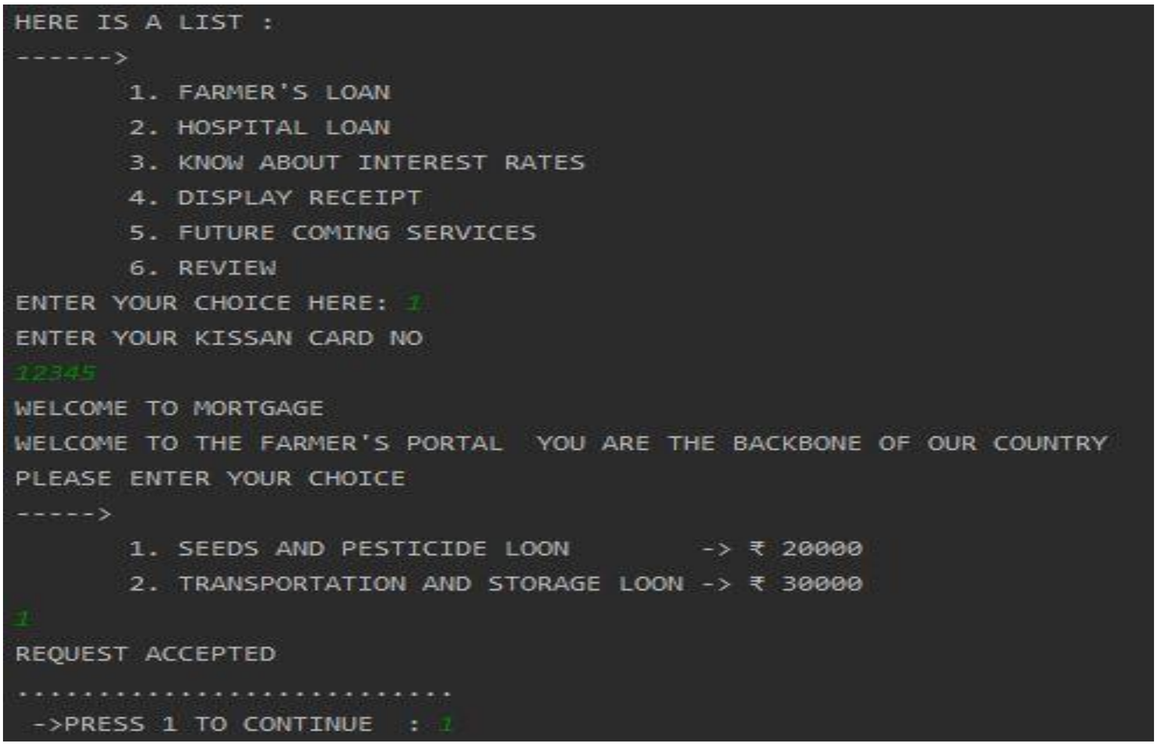


```
WELCOME TO MORTGAGE

PLEASE ENTER YOUR HOSPITAL NAME
johns
ENTER YOUR CHOICE
----->
1.     SURGERY LOAN   -> ₹ 50000
2.     DIALYSIS LOAN  -> ₹ 20000
1
REQUEST ACCEPTED
...........................
 ->PRESS 1 TO CONTINUE  : 1
```

Fig 5.3 Hospital

5.4 FARMER LOAN PORTAL

Here farmer enters the kissan card no the get the access inside it and purchase loan package with interest rate of 4%.

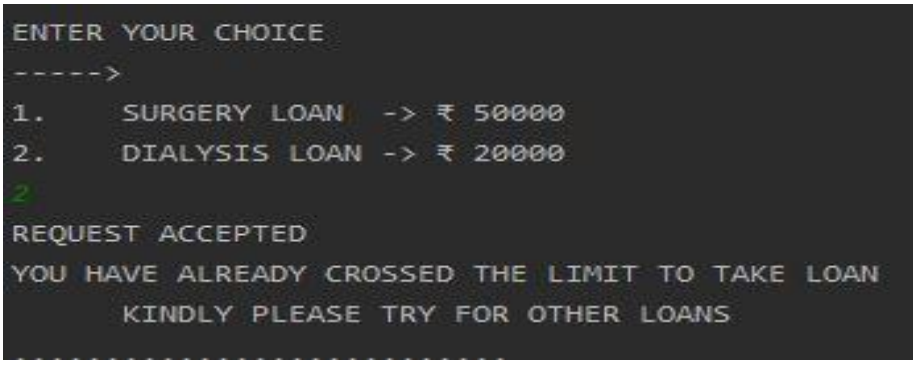


```
HERE IS A LIST :
------>
       1. FARMER'S LOAN
       2. HOSPITAL LOAN
       3. KNOW ABOUT INTEREST RATES
       4. DISPLAY RECEIPT
       5. FUTURE COMING SERVICES
       6. REVIEW
ENTER YOUR CHOICE HERE: 1
ENTER YOUR KISSAN CARD NO
12345
WELCOME TO MORTGAGE
WELCOME TO THE FARMER'S PORTAL  YOU ARE THE BACKBONE OF OUR COUNTRY
PLEASE ENTER YOUR CHOICE
----->
       1. SEEDS AND PESTICIDE LOON        -> ₹ 20000
       2. TRANSPORTATION AND STORAGE LOON -> ₹ 30000
1
REQUEST ACCEPTED
...........................
 ->PRESS 1 TO CONTINUE  : 1
```

Fig 5.4 Farmer

5.5 LOAN LIMIT

The client cannot take more than two same package and if it is tried, it gets blocked by the program.

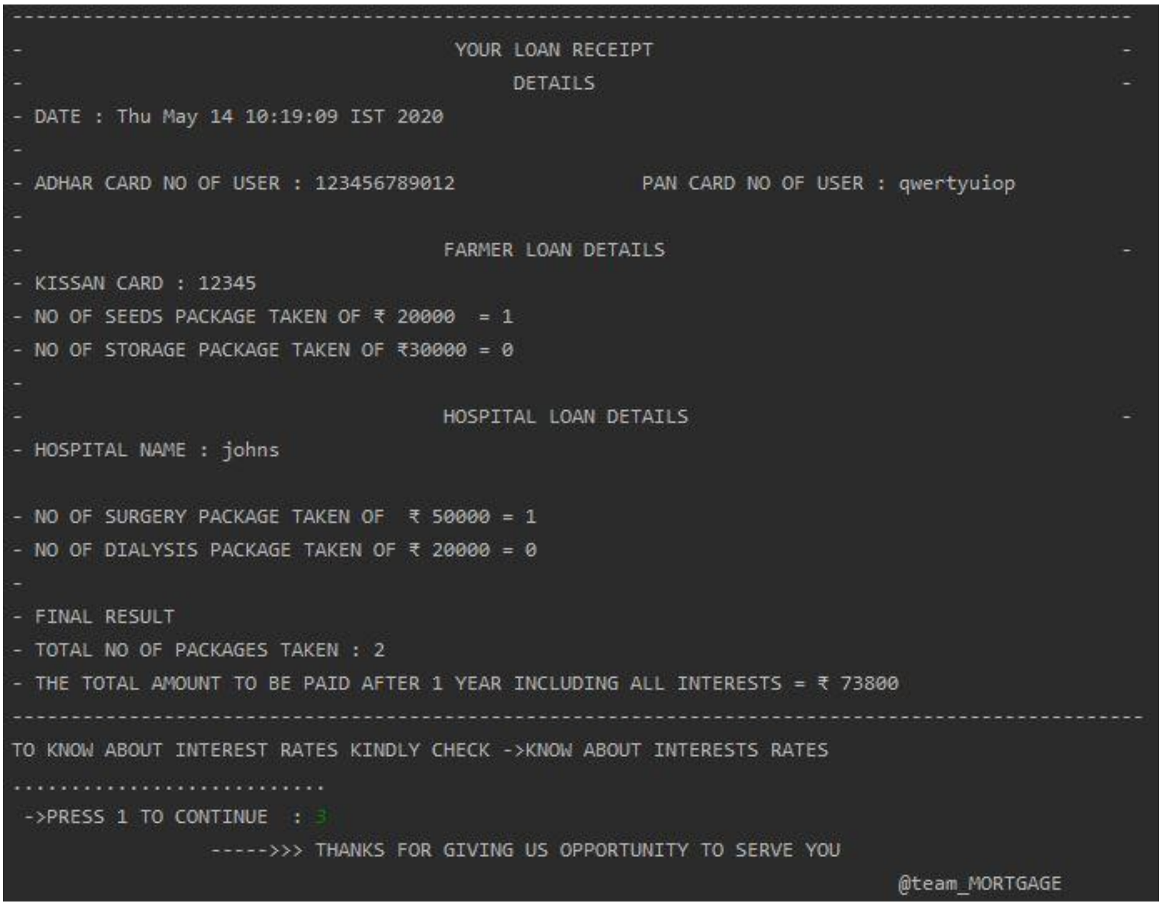


```
ENTER YOUR CHOICE
----->
1.     SURGERY LOAN  -> ₹ 50000
2.     DIALYSIS LOAN -> ₹ 20000
2
REQUEST ACCEPTED
YOU HAVE ALREADY CROSSED THE LIMIT TO TAKE LOAN
       KINDLY PLEASE TRY FOR OTHER LOANS
...........................
```

Fig 5.5 Loan limit

5.6 RECEIPT

After all the purchases done by the client, the receipt is displayed and it contains all the details including the total amount to be paid by inclusive of all the interests.



```
-----------------------------------------------------------------------------------------------
-                                    YOUR LOAN RECEIPT                                        -
-                                         DETAILS                                             -
- DATE : Thu May 14 10:19:09 IST 2020
-
- ADHAR CARD NO OF USER : 123456789012                PAN CARD NO OF USER : qwertyuiop
-
-                                   FARMER LOAN DETAILS                                       -
- KISSAN CARD : 12345
- NO OF SEEDS PACKAGE TAKEN OF ₹ 20000  = 1
- NO OF STORAGE PACKAGE TAKEN OF ₹30000 = 0
-
-                                   HOSPITAL LOAN DETAILS                                     -
- HOSPITAL NAME : johns

- NO OF SURGERY PACKAGE TAKEN OF  ₹ 50000 = 1
- NO OF DIALYSIS PACKAGE TAKEN OF ₹ 20000 = 0
-
- FINAL RESULT
- TOTAL NO OF PACKAGES TAKEN : 2
- THE TOTAL AMOUNT TO BE PAID AFTER 1 YEAR INCLUDING ALL INTERESTS = ₹ 73800
------------------------------------------------------------------------------------------------
TO KNOW ABOUT INTEREST RATES KINDLY CHECK ->KNOW ABOUT INTERESTS RATES
...........................
 ->PRESS 1 TO CONTINUE  : 3
                 ----->>> THANKS FOR GIVING US OPPORTUNITY TO SERVE YOU
                                                                             @team_MORTGAGE
```

Fig 5.6 Loan receipt

CHAPTER 6

CONCLUSION

We can conclude by saying that we have meet the objectives of our mini project by using the object oriented programming concepts. Mortgage udhar is very simple concept to lend money for the farmers and hospital patients, it also makes less human effort and the process is also safer for all the different sections, which makes it safer to purchase a certain package of loan. As it provides a valid receipt of the purchases to the client, So no third party is included. By doing mini project on this topic I have learnt a lot about object oriented programming in java and also Improved my coding knowledge, but object oriented programming plays core role in implementing the logic, as the functionality of the program is object oriented.

As the present technology asks for safe and faster outputs without any intruder to penetrate inside, there is high scope for object oriented programming and also for new efficient logics. From this mini project I am able to understand the need of object oriented programming in the modern day, where we need to find logic and code a program for real time problems. It also helped to demonstrate the part of object oriented programming as well as few concepts of Java programming language. Being an Computer Science Engineer it is not just to know the concepts of object oriented programming, but it is to understand the depth of the program coded and also to understand the background logics and crack down the new advanced logic or technique which perform same functionality with additional advanced outcome step by step logic of the code with necessary result flowing to next level of code has to be understood to find the best of all.

REFERENCES

[1] WIKIPEDIA

[2] GEEKS FOR GEEKS

[3] PROGRAMMING WITH JAVA- A PRIMER

[4] Referred links are

https://www.javatpoint.com/java-oops-concepts

https://www.guru99.com/java-oops-concept.html