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

nirf TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RENDITION SERVICES

A MINI PROJECT REPORT

Submitted by

NALLAVENGANNAGARI JAGADEESWAR REDDY

USN: 1NH18CS124

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. Accredited by NBA

nirf TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the mini project work titled

RENDITION SERVICES

Submitted in partial fulfilment of the degree of Bachelor of Engineering

NALLAVENGANNAGARI JAGADEESWAR REDDY

USN: 1NH18CS124

DURING

EVEN SEMESTER 2019-2020

COURSE CODE: 19CSE39

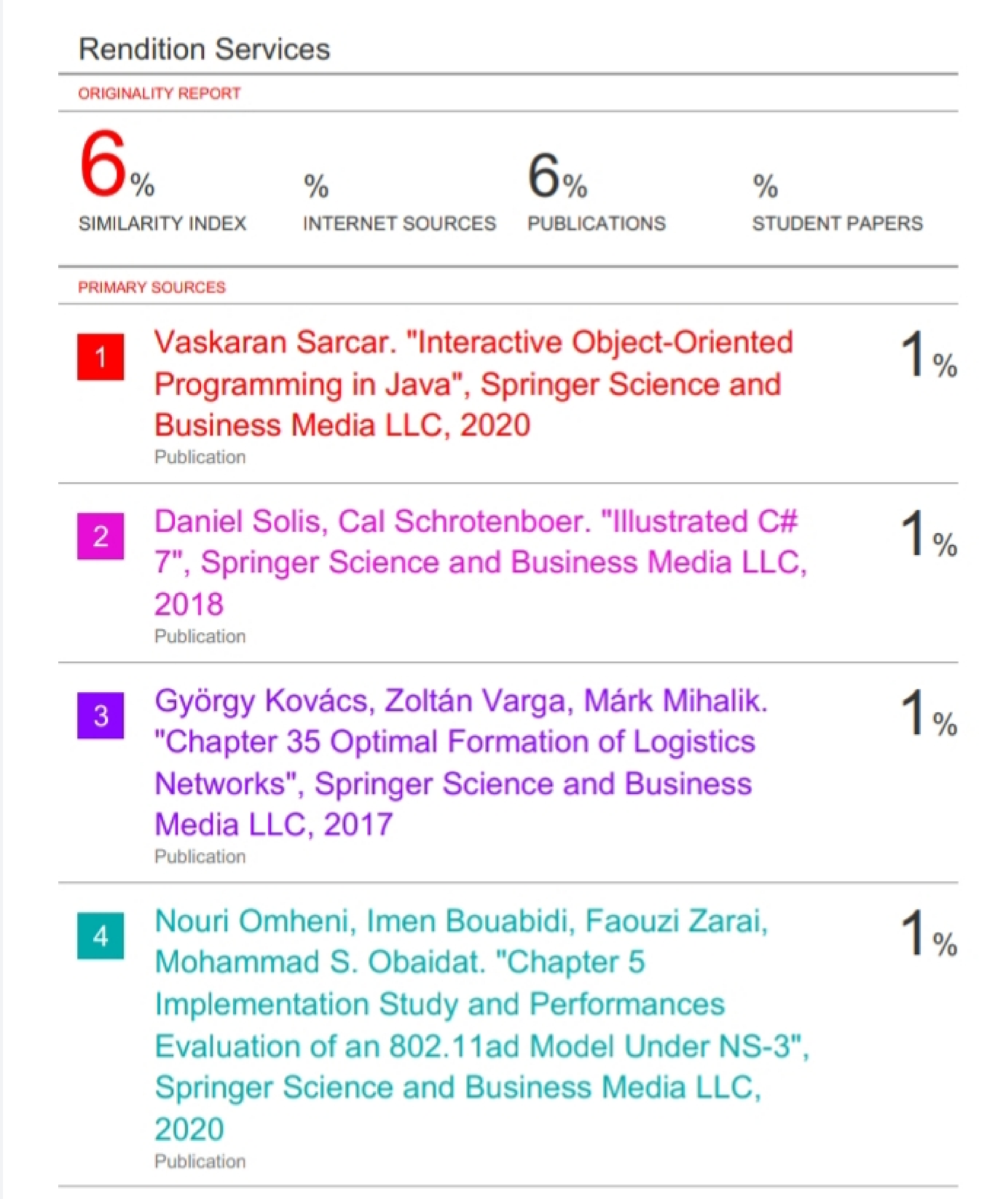
Signature of Reviewer   Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner   *Signature with date*

1.____________________________   ___________________________

2.____________________________   _____________________________



Rendition Services

ORIGINALITY REPORT

6%	%	6%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Vaskaran Sarcar. "Interactive Object-Oriented Programming in Java", Springer Science and Business Media LLC, 2020 Publication	1%
2	Daniel Solis, Cal Schrotenboer. "Illustrated C# 7", Springer Science and Business Media LLC, 2018 Publication	1%
3	György Kovács, Zoltán Varga, Márk Mihalik. "Chapter 35 Optimal Formation of Logistics Networks", Springer Science and Business Media LLC, 2017 Publication	1%
4	Nouri Omheni, Imen Bouabidi, Faouzi Zarai, Mohammad S. Obaidat. "Chapter 5 Implementation Study and Performances Evaluation of an 802.11ad Model Under NS-3", Springer Science and Business Media LLC, 2020 Publication	1%

ABSTRACT

Nowadays it is very important for people to send or receive the courier from one place to another place , from one person to other person. It is not easy to do this process manually because it would become very hectic and it consumes lot of time.

Hence it is suggested to automate the process by developing the relevant software as the world is moving from manual working to information and technology era where automation becomes important in all part of life.

The main purpose of this system is to send courier from one place to another place by knowing the distance of destination and upon the distance the price depends and after the payment the customer will get an otp for boarding the courier and the project is done using object oriented concepts of java. The system will be used for day to day rendition services.This system increases the efficiency and increases the customer satisfaction level.

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.Lakshmi,** assistant professor, 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.

N.JAGADEESWAR REDDY

(1NH18CS124)

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

The project entitled as “RENDITION SERVICES” is developed to transfer the some important and personal messages from one place to another. So this is used for global connections. Immediate processing will be done. The important notification of this project is to maintain a high level of trading.

Today people like to send the letter and other things with the help of this courier. Because it is maintaining the timings and the minimum expenditures. The international and cargo trading can be done in this courier. This project is provided for making this work as computerized one. So the manual workings can be reduced by this valuable processing.

This courier is also used in domestic and it is also provided in remote areas. So the person who is unknown to this process is also learned easily. The branch of this courier is provided all over the world. The courier is much faster than the postal letters, so the people like to have this kind of approaches.

1.2 OBJECTIVES

Primary Objectives are to:

- Consistently provide the best standards of customer care.
- Always deliver your packages on time.
- Work in partnership with your business to deliver the services you require.
- Linking people, delivering business.
- To be a totally committed and valued workforce.
- To care about our customers and our colleagues.
- To excel through development, teamwork and innovation.

1.3 METHODOLOGY TO BE FOLLOWED

In this project we use OOP's with java concepts as a main topic to determine waiting time of customer in a queue using token number.

In this project first we need to create a class or oops concepts for car reservation application. First, we need to enter our details and we should select type ofpayment in which customer it's a customer choice.

Using oops with java concept in this project we can able to get the required bike for booking.

1.4. EXPECTED OUTCOMES:

We run the project as java application.

1.details of destination

2.distance of courier in kms

3.payment option

4.amout to pay

5.otp

Your courier is ready for boarding.

1.5. HARDWARE AND SOFTWARE REQUIREMENTS

1.5.1 HARDWARE REQUIREMENTS:

- Processor: Inte1 core i3/i4/i5/i6/i7
- Speed:3.20 GHz to 3.60 GHz
- RAM:512 MB RAM or more.
- ROM:2 GB DDR3

1.5.2 SOFTWARE REQUIREMENTS:

- Jdk installer by Oracle.
- Eclipse IDE for Enterprise java Developers.
- Windows XP/Windows 7/Windows 8/Windows 10.

1.5.3.1 Eclipse IDE.

Eclipse is an integrated development environment (IDE) used in computer programming.[6] It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia,[7]Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

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.

2.1 CLASS

- class is like a blueprint or a template for creating objects in java. It defines the state or behavior of the object created. class can have any number of variables, and methods of various types to access to different values.,
- Each class has a constructor, it can be of type default or parameterized. These constructors are used to initialize objects, with default values. class can also inherit characteristics from other class.
- While defining a class, we can declare its exact form and nature, by specifies the data that it contains and the code which operates on the data.
- The class is declared by use of the class keyword. The general form of a class definition is as follows:

```
Class className {
type instance-variable 1;
type instance-variable 2;
//…
type instance-variable N;
type method name 1(parameter-list) {
//body of method
}
type method name 2(parameter-list) {
//body of method
}
type method name 3(parameter-list) {
//body of method
}
}
```

- A class declaration can include these in order:

Access Modifier -> Class name -> Superclass -> Interface -> Body

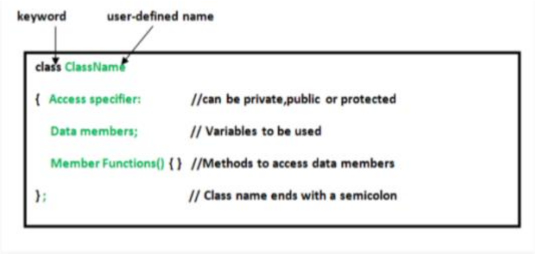
There are various types of classes such as:

- Nested class

- ➢ Anonymous class
- ➢ Lambda expressions

Syntax:

class ClassName



```
class ClassName
{  Access specifier:          //can be private,public or protected
   Data members;              // Variables to be used
   Member Functions() { }     //Methods to access data members
};                            // Class name ends with a semicolon
```

Fig 2.1 (a)

- The data,or variables,defined wothin a class are called instance variables.The code contained within methods.Collectively the methods and variableddefined within a class are called members of the class. In most classes,the instance variables are acted upon and accessed by the methods defined for that class.thus the methods determine how a class data can be used.
- Putting the number elements and methods into together in the definition of a class is called encapsulation.

2.2 OBJECT

- The object is a self-contained entity that has a state, behavior and identity.
- Examples: A dog has different states colors, names, breeds as well as behaviors.
- Object allocates memory for the template class. It defines the behavior of the class. It is a specimen or instance, of class used to invoke or execute any of the methods or features of the class for which object is created.
- An object contains physical as well as logical entity whereas a class does not. Memory or storage allocation takes place for a class when object is created. The methods and the variables of a class are accessed using objects.
- Objects of a class has a two steps process:
    - ➢ The First, must be declare a variable of the class types

- Second. must acquire an actual, physical copy of the object and assign it to the variable, using the new operator.

Syntax:

Classnameobjectname;

Classnamereference_variable=new Classname();

2.2.1 Method Overriding

- It is a method in a subclass has the same name and type signature as a method in its superclassthen the method in the subclass is said to be override method in the superclass.
- Method overriding occurs when the names and also the type signatures of the two methods which are identical. Then the two methods are simply overloaded.
- So, to access the super class version of an overridden method can be called using Super.

2.2.2 Declaring Objects

- When a class is created, we are creating a new data types.
- This type is also used to declare objects of that type.
- However, obtaining the objects of a class is a two-step process,
- In First case, we must declare a variable of the class type. This variable does not define any object. Instead of it is simply a variable that can refer to an object.
- In Second case, we must acquire an actual, physical copy of the object and assign it to that variable by using the new operator.
- The new operator is dynamically allocates memory for an object and returns a to it.

2.2.3 CONSTRUCTOR:

- Constructor in java is a specified type of method which is used to initialize the object. The java constructor is invoked at the time of object creation. So, automatic initialization is performed through the use of a constructor.
- It constructs the valuei.e provides data for the thing that's why it is referred to as Constructor
- There are two rules defined to the constructor:

➢ Constructor name must be same as its class name

➢ Constructor must have no explicit return type

- There are two types of Constructors:

➢ Default constructor (no-arg constructor)

➢ Parameterized constructor: It accepts a certain parameter which is called Parameterized constructor.

➢ Garbage Collection: It occurs only sporadically during the execution of programs. The objects are dynamically allocated by using the new operator, how such objects are destroyed and their memory released for later reallocation.

2.3 INHERITANCE

Inheritance is an oops concept in java that permits us to define a category from an existing class. The keyword 'extends' is employed for inheritance.

• Superclass: The parent/base class from which attributes, methods are inherited.

• Subclass: The child/derived class which inherits attributes, methods.

1. SINGLE INHERITANCE

Single inheritance is when a category inherits properties from one class only. All the attributes except private members are inherited or extended by child class from parent class.

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

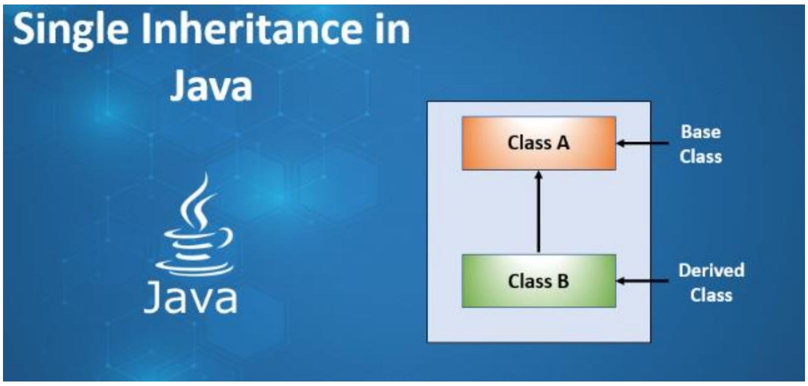


Fig 2.3 (a)

2. MULTILEVEL INHERITANCE

Multilevel inheritance is when a category inherits properties from derived class. This derived class becomes the parent for the new child class. It allows accessing of grandparent class attributes by the kid class also.

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

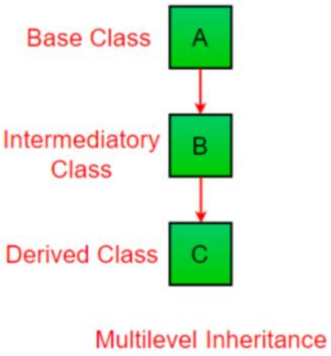


Fig 2.3 (b)

3. HIERARCHICAL INHERITANCE

Hierarchical inheritance is when a category is inherited two or more classes. during this sort of inheritance all of the super class’s sub classes inherit same attributes of the parent class.

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

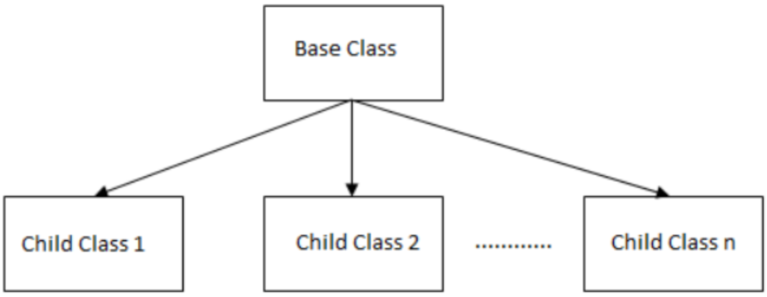


Fig 2.3 (c)

2.4 POLYMORPHISM

Polymorphism may be a vital concept in object-oriented programming. It means an equivalent object, method or operator acts differently in several cases.

Types of polymorphism are:

- Run-time polymorphism
- Compile-time polymorphism

Run-time polymorphism is completed using method overriding.

Method Overriding: It means different methods have same syntax and return type.

```
class A{ }
class B extends A{
public void display(){
System.out.println(“Method1”);
}}
class C extends A{
public void display(){
System.out.println(“Method2”);
}}
```

Compile-time polymorphism is completed through method overloading and operator overloading.

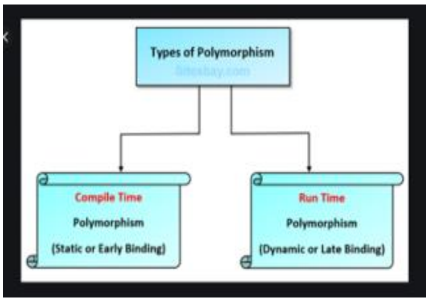
Method Overloading: It means different methods with same name differ in number, type or sequence of arguments passed in them.

```
class A{
public void display(int a){
```

```
System.out.println(a);

}}

class B{

public void display(String s){

System.out.println(s);

}}

class C{

public void display(int x, int y){

System.out.println(x+” “+y);

}}
```

Operator overloading: ‘+’ operator is employed for concatenation also as addition operator.

```
System.out.println(m+”ways”);

System.out.println(a+b);
```



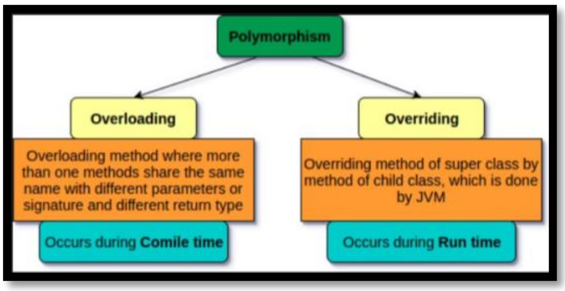

Polymorphism
Overloading
Overriding
Overloading method where more than one methods share the same name with different parameters or signature and different return type
Overriding method of super class by method of child class, which is done by JVM
Occurs during Comile time
Occurs during Run time

Fig 2.4 (a) Fig 2.4 (b)

2.5 ABSTRACT CLASS

An abstract class may be a template definition of methods and variables of a category which may be a category of objects that contains one or more abstracted methods.

Abstract classes are utilized in all OOP languages. Objects or classes maybe abstracted, which suggests that they're summarized into characteristics that are relevant to the present program’s operation.

Individual instances that are resulting from classes are objects. Declaring a category as abstract means it cannot be directly instantiated, which suggests that an object cannot be created from it. That protects the code from getting used incorrectly. An Abstractclass which subclasses are to be further define as attributes are necessary for individual. Abstract classes similar with main classes, which are the default type. A concrete class has no abstracted methods and should be instantiated and utilized in code.

Syntax:

Class abstract classname {…}

Points to Remember:

- An abstract class should be announced with an abstract keyword.
- It have only abstract methods and also non-abstract methods.
- It can't be instantiated.
- It can have constructors as well as static methods also.
- It can have final methods which may force the subclass to not change the body of the tactic.

2.6 MULTITHREADING

Multithreading in java could also be a process of executing multiple threads. A thread may be a lightweight sub-process which is that the smallest unit of processing. Multiprocessing as well as multithreading, both are used to perform multitasking. However, we use multithreading than multitasking than multiprocessing because threads use a shared memory. They won’t allocate separate memory spacesto saves memory, and switching between the threadswhich takes less time to process. Java Multithreading is typically utilized in games, animation, etc.

Multitasking could also be a process of executing multiple tasks simultaneously. We use multitasking to utilize the CPU. Multitasking are often achieved in two ways:

o Process-based Multitasking (Multiprocessing)

o Thread-based Multitasking (Multithreading)

1) Process-based Multitasking (Multiprocessing)

- Each process has an address in memory. Process based multitasking allocates a separate memory areas.
- A process is heavyweight.
- Cost of communication between the tactic is high.
- Switching from one process to a different requires a while for saving and loading register, memory maps, updating lists, etc.

2) Thread-based Multitasking (Multithreading)

- Threads share an equivalent address space.
- A thread is lightweight.
- Cost of communication between the thread Is slow

2.7 I/O FUNCTIONS IN JAVA

I/O functions in java are wont to process the input and provide output. It uses concept of Streams to operations fast. We will also perform file handling in java using these streams.

- System. Out
- System.in
- System. Err
- Output Stream
- Input Stream

Example:

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

System.out.println(“Hello”);

System.err(0);
```

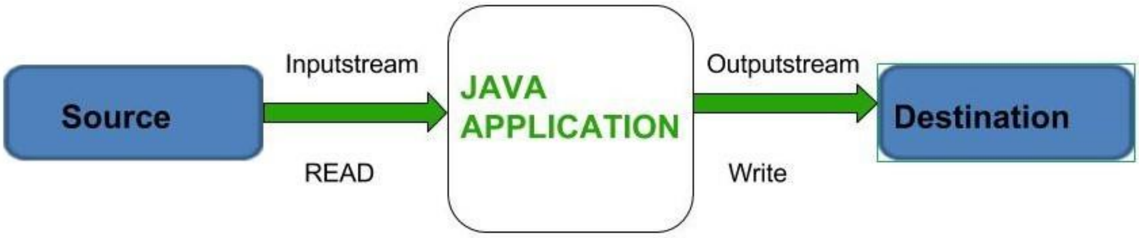


Fig 2.7(a)

2.8 PACKAGES IN JAVA

Packages in java are to encapsulate a gaggle of class’s sub-packages and interfaces. It prevents naming conflicts and also provides controlled access and is referred to as data encapsulation.

- Packages are containers for classes that are used the category name.
  - Through the utilization of the interface keyword; Java allows to completely abstract the interface from its implementation.
  - By using interface, we will specify the set of methods which will be implemented by one or more class.
  - In interface itself, doesn't actually define any implementation.
  - A class can implement quite one interface.,
  - Java provides a partitioning the category name space into more manageable chunks. This mechanism is named a package.
  - In package are both a naming and a visibility control mechanism.

• It is feasible to define classes inside a package that aren't accessible by code outside those packages.

2.8.1 Defining Package

• To create a package simply include a package command is that the first statement during a Java source file.

• Any classes declared within that file is belong to the required package.

• Package statement defines name space during which classes are stored.

• If we skip the package statement the category names are put into the default package, which has no name.

• The general sort of the package statement is as follows package pkg;

• During this pkg is that the name of the package.

• For instance we will consider the subsequent statement creates a package called My Package.

2.9 Exception handling

- It allows us to handle the runtime errors caused by exceptions.
- An exception is a not normal, It occurs during the execution of a program like compile time that includes the flow of instructions.
- Languages that do not support exception handling, errors must be checked and handled manually—typically through the use of error codes [system generated error codes from 0 to 499].
- This approach is as cumbersome as it is troublesome.
- Java's exception handling avoids these problems and, brings semantic error management into the object- oriented world. All exception handling types are subclasses of the built-in class which is called Throw able.

2.9.1 Exception Types:

- Throw able is at the highest of the exception class hierarchy.

- In the below Throw able they are two subclasses which takes partition exceptions into two different branches.
- One branch is headed by Exception.
- This class is employed for exceptional conditions that user programs should catch.
- This is also the category that you simply will subclass to make your own custom exception types.
- There is a crucial subclass of Exception, called Runtime Exception. Exceptions of this sort are automatically defined for the programs that you simply write and include things like division by zero and invalid array indexing.
- The other branch is topped by Error.
- These are not expected to be caught under normal circumstances by your program, are typically created in response to catastrophic failures that cannot usually be handled by your program.
- Exceptions of type Error are employed by the Java run-time system to point errors having to try to with the run-time environment, itself.
- Stack overflow is an example of such a mistake.

2.9.2 Exception-Handling in Java five keywords

I. try

Program statements that you simply want to watch for exceptions are contained within a try block.

If an exception occurs within the try block, it is thrown.[an object representing that exception is created and thrown in the method that caused it]

II. catch

The code which going to can catch this exception (using catch) and handle it in some rational manner.

It generated exceptions which are automatically thrown by the Java run-time application.

III. throw

For manually throw an exception, we use the keyword called as throw.

IV. throws

Any exception that's thrown out of a way must be specified intrinsically by a throws clause.

V. finally

Any code that absolutely must be executed after a try block completes, is put during a finally block.

CHAPTER 3

DESIGN

3.1 ALGORITHM

- First user need to create an class by using objected oriented concepts.
- Classes can consist any topics which were related to object oriented concepts
- And object should be created for calling the class.
- First customer should enter destination and distance in kms for this user is using java in built functions like java.util
- Here user uses if else statements to check the customer age if customer age is equal to 18 or greater than 18 then customer can be accessed to book his courier
- If customer is less than 18 then customer is not able to continue further process
- After checking customer age then customer has to enter their payment option
- After selecting the payment options will displayed by using print statements and if else statements are for payments
- There will be five payments options will be displayed on the screen we select and customer should complete their payment
- After completion of payment OTP will generated customer need to enter the OTP if OTP is verified then courier order will be completed.

CHAPTER 4

IMPLEMENTATION

4.1 IMPLEMENTATION OF OBJECT

Step 1: declare a variable of the class type that can refer to an object.

Step 2: create an object by using the "new" keyword

Step 3: "new" keyword should be followed by the call of the constructor. This call initializes the new object.

Step 4: after the creation of object, we can call a class method by obj_var_name.method_name();

Example:

```
h ob=new h();

ob.m();
```

4.2 IMPLEMENTATION OF INHERITANCE

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

Step 2: in this project, features of super class contains the methods to perform online payment for courier service.

Step 3: create the sub class (child class) which inherits the features of super class (parent class). .

Step 4: ."extends" is the keyword to inherit the features of super class. In this project q acts as super class whereas h acts as sub class.

4.3 IMPLEMENTATION OF STACK

Stack is a linear data structure which follows the LIFO (last-in first-out) principle. LIFO states that the elements can be inserted or removed only at one end of the stack (top side). Stack performs push and pop operation. In this project, push operation was implemented.

PUSH OPERATION

The process of pushing a new data element onto the stack is known as Push operation.

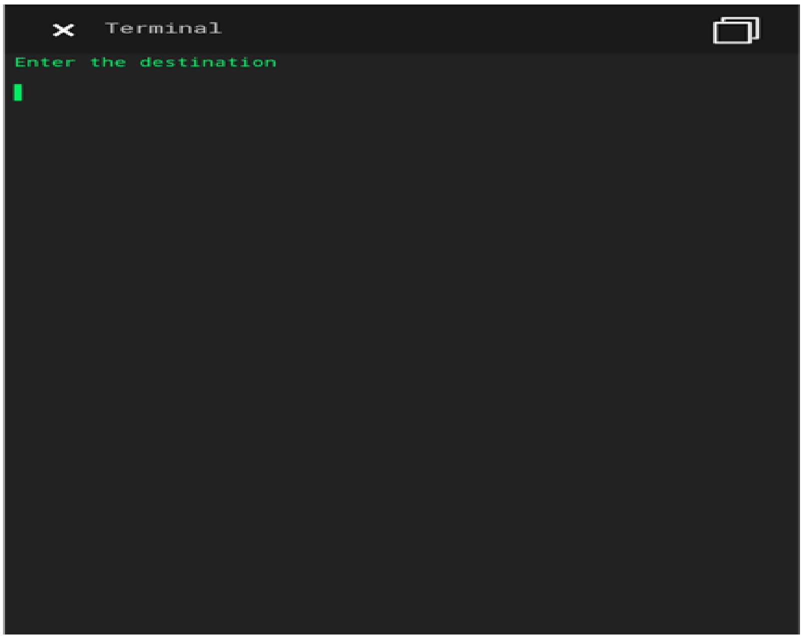
Step 1: initialize the stack size and top value. Basically, Top value will be initialized to -1 value.

Step 2: if the stack is full, displays the statement "MAX NUMBER OF PARCEL REACHED"

Step 3: if stack is not full, then increments the top value and points to the next empty space and also adds the element to the stack.

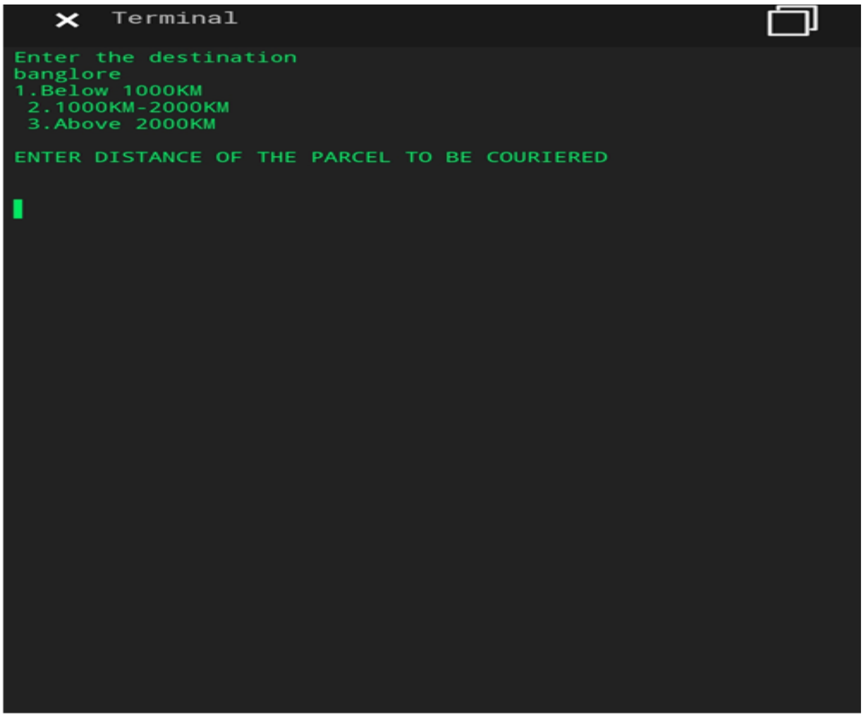
CHAPTER 5

RESULTS



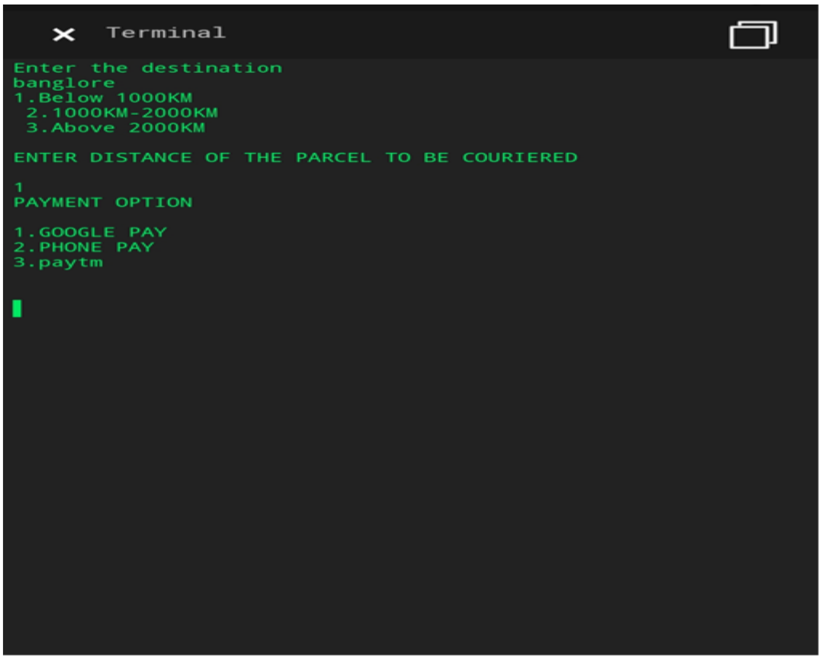
```
Enter the destination
```

FIG 5.1 : Initially the user has to enter the destination to where the parcel has to be delivered .



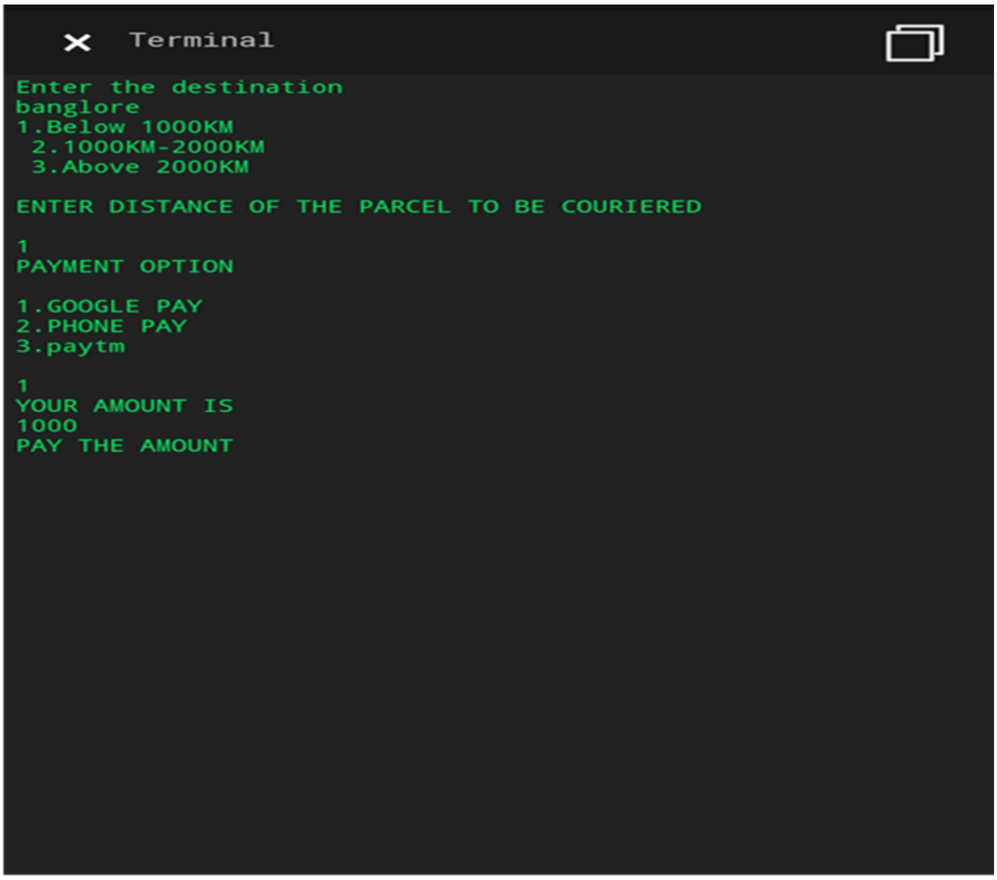
```
Enter the destination
banglore
1.Below 1000KM
 2.1000KM-2000KM
 3.Above 2000KM

ENTER DISTANCE OF THE PARCEL TO BE COURIERED
```

FIG 5.2 : Here we have to select the distance the parcel has to travel in kms.



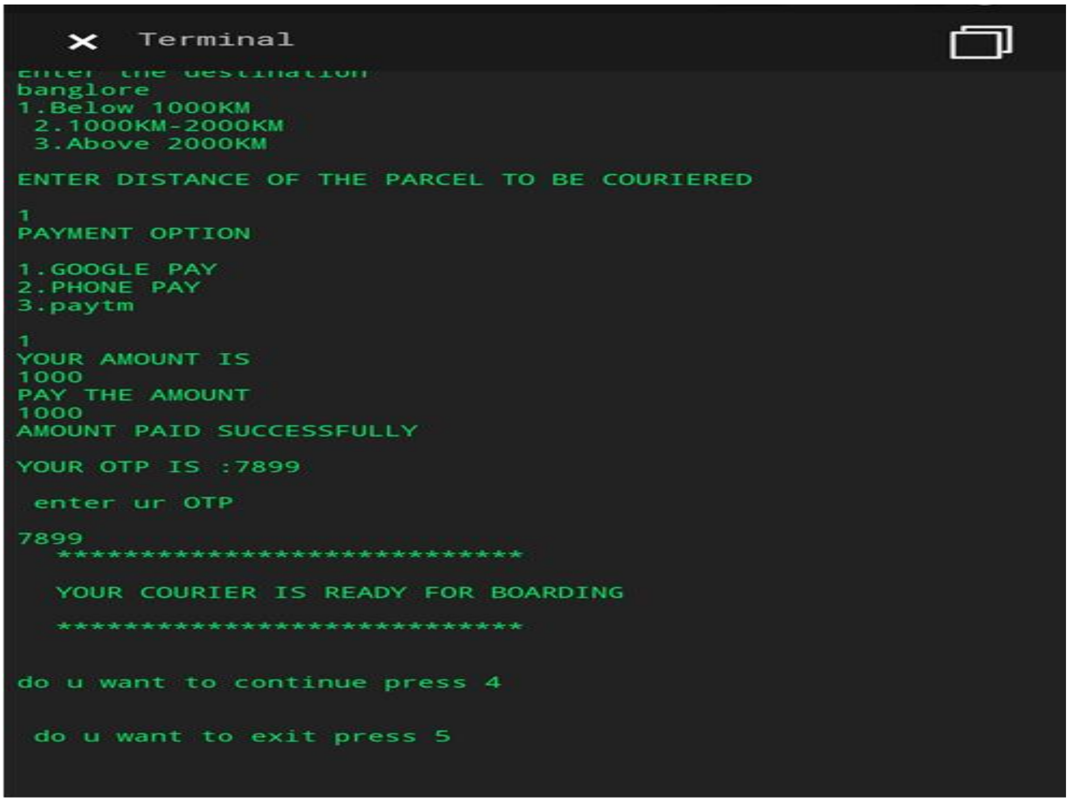
```
Enter the destination
banglore
1.Below 1000KM
 2.1000KM-2000KM
 3.Above 2000KM

ENTER DISTANCE OF THE PARCEL TO BE COURIERED

1
PAYMENT OPTION

1.GOOGLE PAY
2.PHONE PAY
3.paytm
```

FIG 5.3 : After selecting the distance user has to select the payment option and we have provided three modes of online payment.



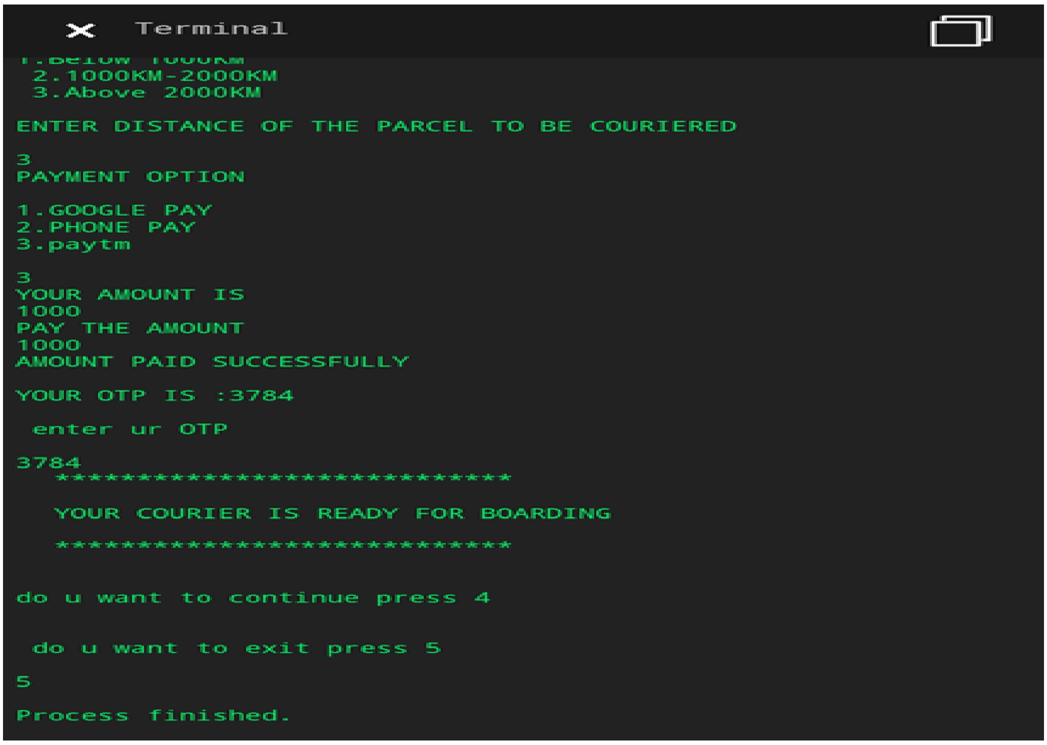
```
Enter the destination
banglore
1.Below 1000KM
 2.1000KM-2000KM
 3.Above 2000KM

ENTER DISTANCE OF THE PARCEL TO BE COURIERED

1
PAYMENT OPTION

1.GOOGLE PAY
2.PHONE PAY
3.paytm

1
YOUR AMOUNT IS
1000
PAY THE AMOUNT
```

FIG 5.4 : Here the user will get the amount to be paid by the customer for the delivery.



```
Enter the destination
banglore
1.Below 1000KM
 2.1000KM-2000KM
 3.Above 2000KM

ENTER DISTANCE OF THE PARCEL TO BE COURIERED

1
PAYMENT OPTION

1.GOOGLE PAY
2.PHONE PAY
3.paytm

1
YOUR AMOUNT IS
1000
PAY THE AMOUNT
1000
AMOUNT PAID SUCCESSFULLY

YOUR OTP IS :7899

 enter ur OTP

7899
   *****************************

   YOUR COURIER IS READY FOR BOARDING

   *****************************


do u want to continue press 4

 do u want to exit press 5
```

FIG 5.5 : After the amount is paid successfully the customer will get the otp and the otp has to be entered here then the courier is ready for delivery.



```
1.Below 1000KM
 2.1000KM-2000KM
 3.Above 2000KM

ENTER DISTANCE OF THE PARCEL TO BE COURIERED

3
PAYMENT OPTION

1.GOOGLE PAY
2.PHONE PAY
3.paytm

3
YOUR AMOUNT IS
1000
PAY THE AMOUNT
1000
AMOUNT PAID SUCCESSFULLY

YOUR OTP IS :3784

 enter ur OTP

3784
   ****************************

   YOUR COURIER IS READY FOR BOARDING

   ****************************


do u want to continue press 4


 do u want to exit press 5

5

Process finished.
```

FIG 5.6 : If the process is completed then the customer can continue by pressing 4 and if he want to exit he has to press 5. Then the process is completed.

CHAPTER 6

CONCLUSION

Rendition service is one of the basic java application in java created using different OOPS concepts available.

Rendition service has the basic options that is needed

1. destination details
2. distance in kms
3. payment method
4. otp

This project is done successfully and also executed successfully without any errors

In this project customer has to enter destination and distance in kms and the payment options will be displayed and customer needs to select the option like googlepay phonepe etc And after completion of payment OTP will be displayed and the customer need to enter the OTP then courier order will complete.

REFERENCES

1. Herbert Schildt-Java_ The Complete Reference-McGraw-Hill Education (2017)

2. Head First Java: A Brain-Friendly Guide, 2nd Edition - Kathy Sierra, Bert Bates(2003)

3. PROGRAMMING WITH JAVA- PRIMER A.