Module 1 (Function)

(1) What is SDLC ?

SDLC (Software development life cycle) is a Structure imposed on the development of a Software product that defines the process for Planning, Analysis , Design , Implementation , Testing , and Maintenance.

(2) What is agile Methodology ?

Agile Methodology is a combination of iterative and incremental process model with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

(3) What is SRS ?

SRS (Software Requirement Specification) is termed as SRS document. This document serves as a detailed of functional and non-functional Requirements needed that Software should fulfil.

(4) What is OOPS ?

OOPS Stands For(Object Oriented Programming) It is about creating objects that contain both data and function. Object-Oriented programming has several advantages over procedural languages. As OOPS is faster and easier to execute it becomes more powerful than procedural languages like C++. OOPS is the most important and flexible paradigm of modern programming. It is specifically useful in modelling real-world problems.

(5) Write Basic Concepts of OOPS ?

- Concepts of OOPS
  1. Object
  2. Class
  3. Encapsulation
  4. Inheritance
  5. Polymorphism
  6. Abstraction

(6) What is Object ?

An object is an identifiable, real-time entity with some characteristics and behaviour. An object is an instance of a class. A class can have multiple objects. An object can also be defined as a bundle of data and its behaviour.

(7) What is Class ?

A class is defined as a group of objects with some properties and relationships. A class can also be defined as a blue print or a template having certain attributes from which we create objects.

(8) What is Encapsulation ?

Encapsulation in object oriented programming encapsulation is defined as the wrapping up of data under a single unit. A formal definition of encapsulation would be encapsulation is binding together the data and related function that can manipulate the data its called encapsulation.

(9) What is Inheritance ?

Inheritance it is the capability of a class to inherit or derive properties or Characteristics other class. It is very important and object oriented Program as it allows reusability using a method defined in another class by using inheritance. The class that derives Properties from other class is known as child class or subclass and the class from which the properties are inherited is base class or parent class.

(10) What is Polymorphism ?

Polymorphism the name defines polymorphism is multiple forms. Which means polymorphism is the ability of object oriented programming to do some work using multiple forms. The behaviour of the method is dependent on the type or the situation in which the method is called. They are operator overloading and function overloading.

1.Operator Overloading;- In operator overloading operator can have multiple behaviour in different instances of usage.

2.Function Overloading:-Function with the same name that can do multiple types based on some condition.

(11) What is RDBMS ?

RDBMS stands for Relational Database Management System in this Database Management System. In this Database Management the data is Organized into the related tables. To access the database it uses Structured Query language (SQL). This model is based on the mathematical theory of relational algebra and calculus.

(12) What is SQL?

SQL is structured Query Language, Which is a computer language for Storing, manifulating and retrieving data stored in relational database.

(13) Write SQL Commands?

SQL Commands

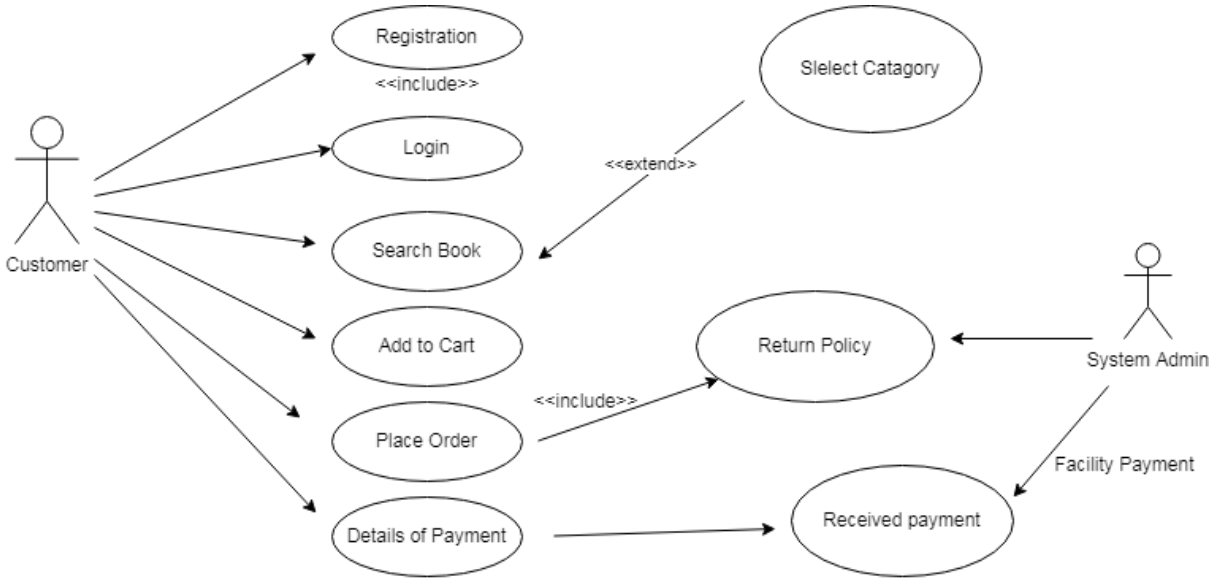
DDL = Data Definition Language

DML = Data Manipulation Language

DCL = Data Control Language

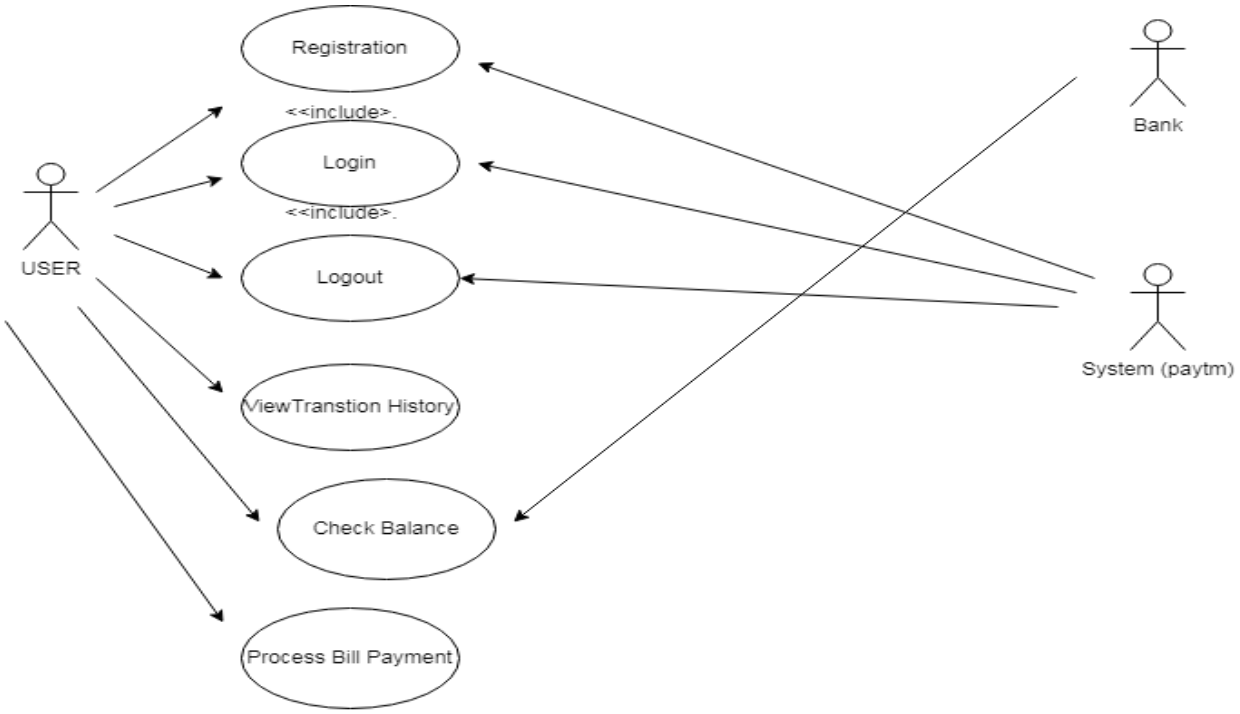
DQL = Data Query Language

(14) Draw use case on online book shopping?



Online Book Shoping

(15) Draw use case on online bill payment system



Online Bill Payment

(16) Write SDLC Phase with basic introduction?

(1) Requirement:-

In this Phase, all requirements are collected from the customer/client. They are provided in document called SRS (System Requirement Specification). All the details are discussed with the Customer/client in detail.

(2) Design:-

- High level design:- It gives the architecture of software products.
- Low level design:- It describes how each and every feature in the products should work and every component.

(3)Implementation:-

- This is the longest phase.
- This phase consists of Frontend + Middleware + Backend.
- Infront-end:-Development of coding is done even SEO settings are done.
- In Middleware:-They connect both the front end and backend.
- In the back-end:-A database is created.

(4) Texting:-

Testing is carried out to verify the entire system. The aim of the tester is find out the gaps and defects within the system and also to check whether the system is running according to the requirement of the customer/client.

(5) Deployment: -

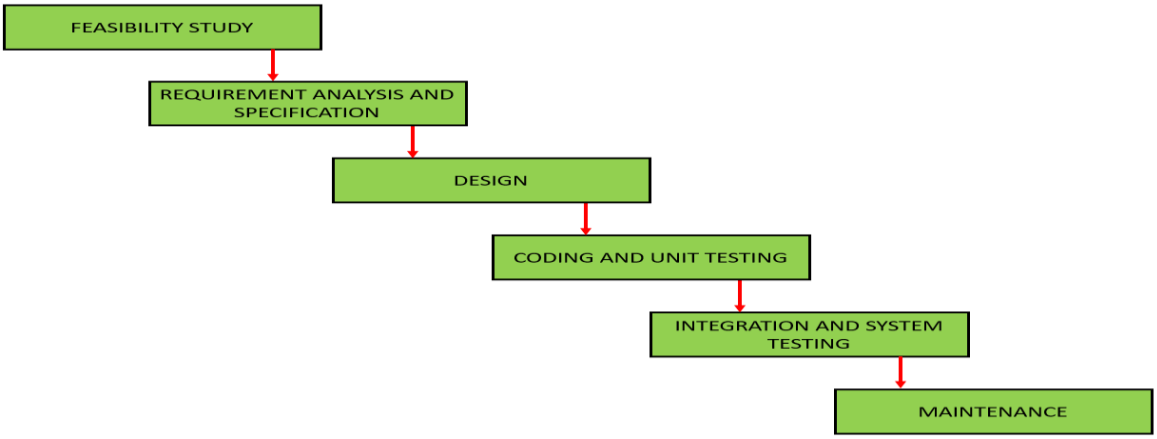
After successful testing, the product is delivered/deployed to the Client, and even clients are trained on how to use the product.

(6) Maintenance:-

Once the product has been delivered to the client a task of Maintenance starts as when the client will come up with an error the issue should be fixed from time to time.

(17) Explain Phases of the Waterfall model?

The different sequential phases of the classical waterfall model are shown in the below figure:



1) Feasibility Study: The main goal of this phase is to determine whether it would be financially and technically feasible to develop the software.

The feasibility study involves understanding the problem and then determining the various possible strategies to solve the problem. These different identified solutions are analyzed based on their benefits and drawbacks, The best solution is chosen and all the other phases are carried out as per this solution strategy.

(2) Requirements analysis and specification: The aim of the requirement analysis and specification phase is to understand the exact requirements of the customer and document them properly. This phase consists of two different activities.

- Requirement gathering and analysis: Firstly all the requirements regarding the software are gathered from the customer and then the gathered requirements are analyzed. The goal of the analysis part is to remove incompleteness (an incomplete requirement is one in which some parts of the actual requirements have been omitted) and inconsistencies (an inconsistent requirement is one in which some part of the requirement contradicts some other part).
- Requirement specification: These analyzed requirements are documented in a software requirement specification (SRS) document. SRS document serves as a contract between the development team and customers. Any future dispute between the customers and the developers can be settled by examining the SRS document.

(3) Design: The goal of this phase is to convert the requirements acquired in the SRS into a format that can be coded in a programming language. It includes high-level

and detailed design as well as the overall software architecture. A Software Design Document is used to document all of this effort (SDD)

(4) Coding and Unit testing: In the coding phase software design is translated into source code using any suitable programming language. Thus each designed module is coded. The aim of the unit testing phase is to check whether each module is working properly or not.

(5) Integration and System testing: Integration of different modules are undertaken soon after they have been coded and unit tested. Integration of various modules is carried out incrementally over a number of steps. During each integration step, previously planned modules are added to the partially integrated system and the resultant system is tested. Finally, after all the modules have been successfully integrated and tested, the full working system is obtained and system testing is carried out on this.

System testing consists of three different kinds of testing activities as described below :

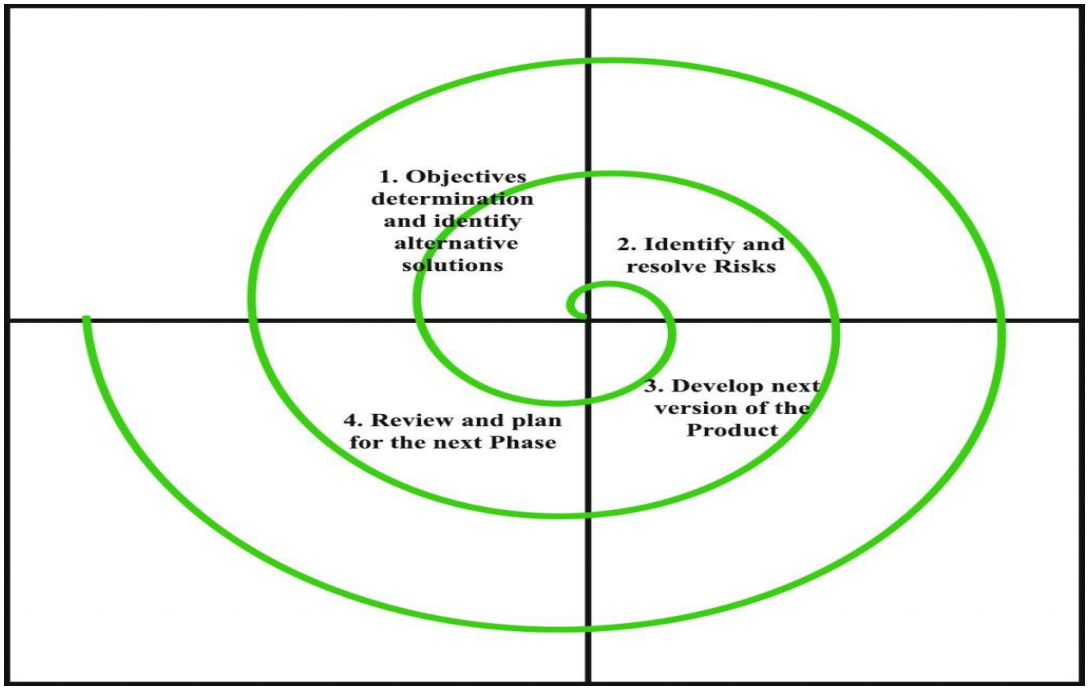
- Alpha testing: Alpha testing is the system testing performed by the development team.
- Beta testing: Beta testing is the system testing performed by a friendly set of customers.
- Acceptance testing: After the software has been delivered, the customer performed acceptance testing to determine whether to accept the delivered software or reject it.

(6) Maintenance: Maintenance is the most important phase of a software life cycle. The effort spent on maintenance is 60% of the total effort spent to develop a full software. There are basically three types of maintenance :

- Corrective Maintenance: This type of maintenance is carried out to correct errors that were not discovered during the product development phase.
- Perfective Maintenance: This type of maintenance is carried out to enhance the functionalities of the system based on the customer's request.
- Adaptive Maintenance: Adaptive maintenance is usually required for porting the software to work in a new environment such as working on a new computer platform or with a new operating system.

(18) Write Phases of spiral model?

The functions of these four quadrants are discussed below-



(1) Objectives determination and identify alternative solutions: Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.

(2) Identify and resolve Risks: During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.

(3) Develop next version of the Product: During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.

(4) Review and plan for the next Phase: In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

(19) Write agile manifesto Principles?

Agile Manifesto Principles

- Individuals and interactions - in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- Working software - Demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.
- Customer collaboration - As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements..
- Responding to change - agile development is focused on quick responses to change and continuous development.

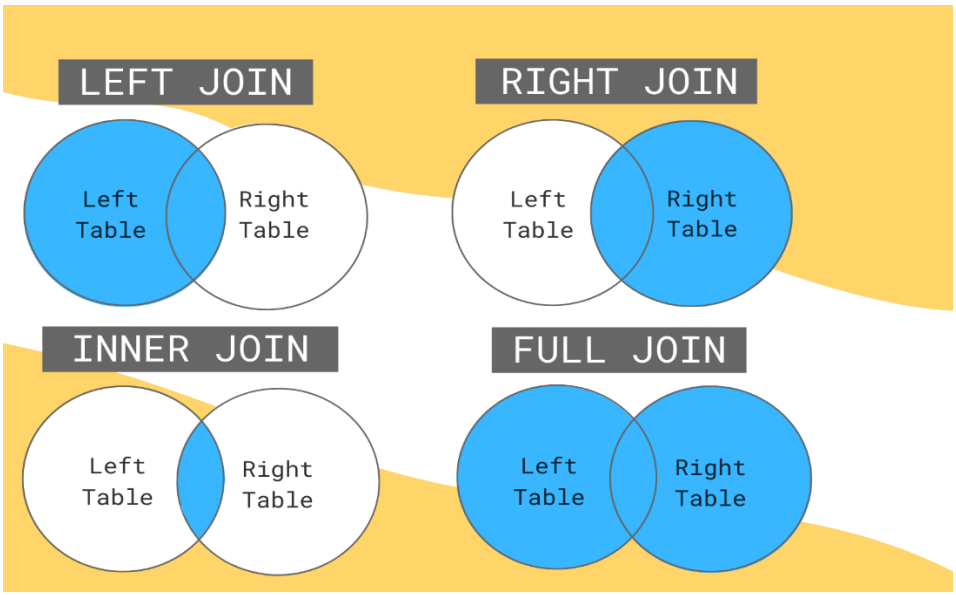
(20) What is join?

Join Statement is used to combined data or rows from two or more Tables based on a common field between them.

(21) Write type of join?

Different types of Joins are as follows:

- Inner Join
- Left Join
- Right Join
- Full Join



(22) Explain working methodology of agile model and also write Pros and Cons?

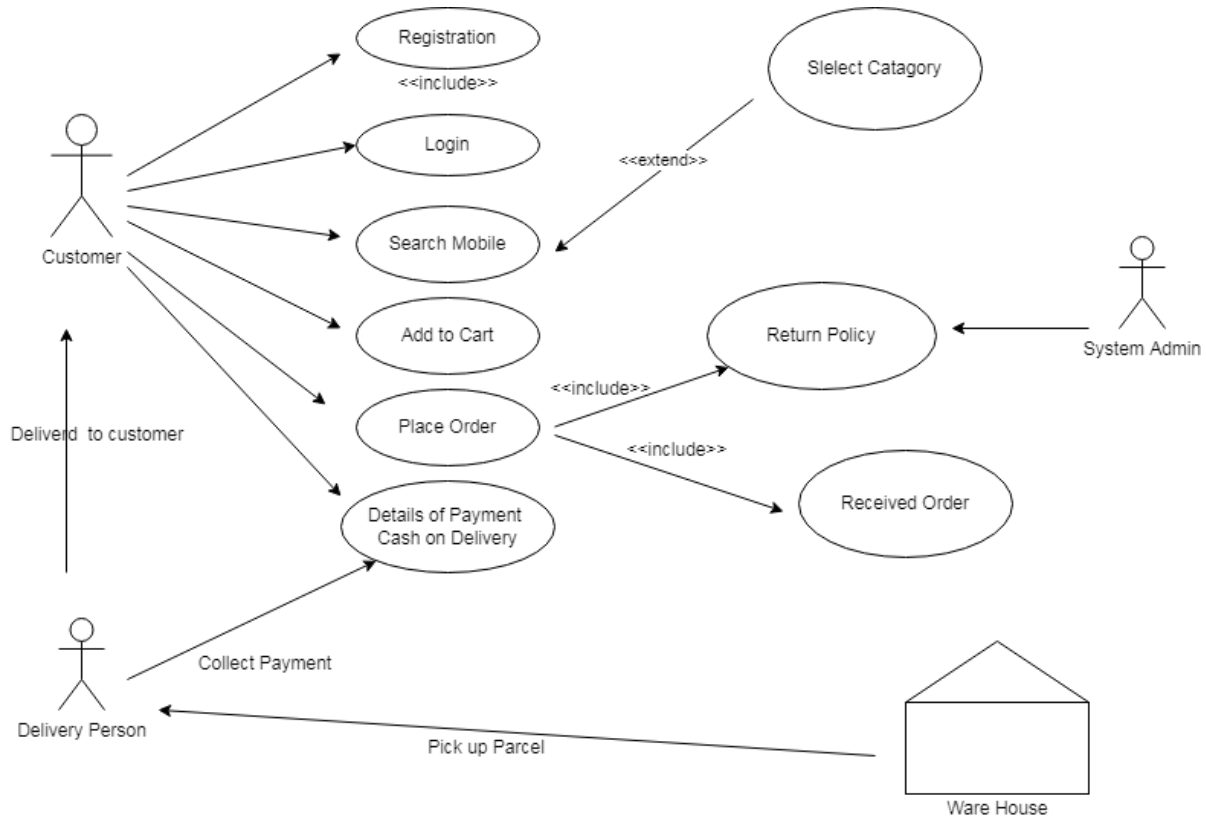
Pros

- Is a very realistic approach to software development
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required
- Easy to manage
- Gives flexibility to developers

Cons

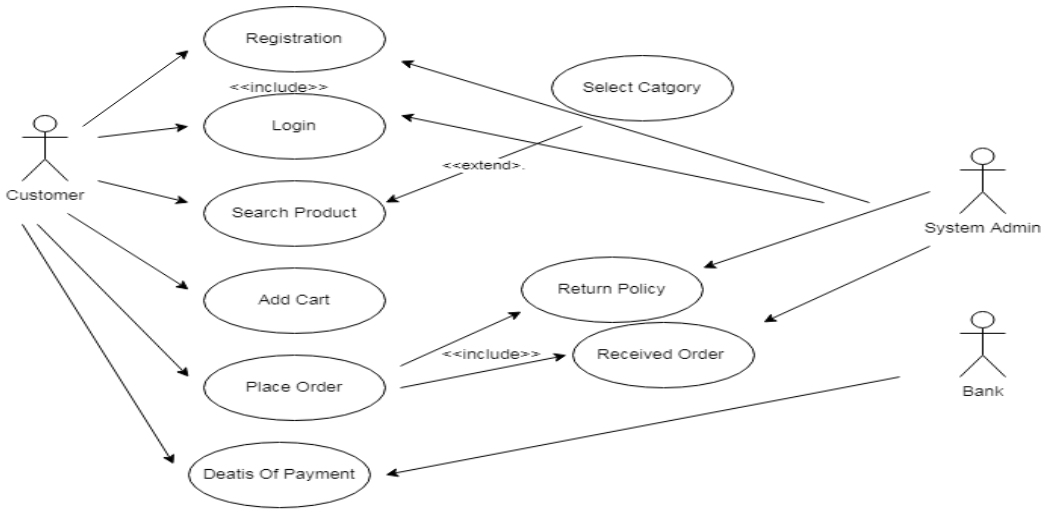
- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not
- work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments
- to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in
- the wrong direction.
- There is very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of.

(23) Draw use case on online shopping product using COD.



Online Shoping Product Using COD

(24) Draw use case on online shopping product using payment gateway.



Online Payment Gateway