Module – 1 (Fundamental)

1)What is SDLC ?

- SDLC is a structure imposed on the development of a software product that defines the process for planning ,implementation,testing ,documentation,deployment , and ongoing maintenance and support.

- There are a number of different development models

- A Software development life cycle is essentially a series of steps, or phases, that provide a model for the development and life cycle management of an application or piece of software.

- The Methodology within the SDLC process can very across industries and organizations, but standards such as ISO/IEC 12207 represent processes that establish a life cycle for software, and provide a mode for the development acquisition, and configuration of software systems.

2) What is Software Testing ?

- Software testing is the process of Evaluating and verifying that a software product or application does what it is supposed to do.The benefits of testing include preventing bugs, reducing development costs and improving performance.

- Software testing is an activity conducted in the software develepment life cycle to verify that the software is accurate and works according to the requirements. testing plays in integral part in any software development project.

- Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

- Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements

- According to ANSI/IEEE 1059 standard, Testing can be defined as A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

- Software testing is a process of executing a program or application with the intent of finding the software bugs.

- When asked, people often think that Testing only consists of running tests, i.e. executing the software Test execution is only a part of testing, but not all of the testing activities Test activities exist before and after test execution

- It can also be stated as the process of validating and verifying that a software program or application or product:

    - Meets the Business and Technical Requirements that guided it's design and development.
    - Works as expected.
    - Can be implemented with the same Characteristic.

- Let's break the definition of Software testing into the following parts:

    - **Process :** Testing is a process rather than a single activity.

    - **All Life Cycle Activities:** Testing is a process that's take place throughout the Software Development Life Cycle (SDLC). The process of designing tests early in the life cycle can help to prevent defects from being introduced in the code. Sometimes it's referred as "verifying the test basis via the test design". The test basis includes documents such as the requirements and design specifications.

- **Static Testing:** : It can test and find defects without executing code. Static Testing is done during verification process. This testing includes reviewing of the documents (including source code) and static analysis. This is useful and cost effective way of testing. For example: reviewing, walkthrough, inspection, etc.
- **Dynamic Testing:** In dynamic testing the software code is executed to demonstrate the result of running tests. It's done during validation process. For example: unit testing, integration testing, system testing, etc.
- **Planning:** We need to plan as what we want to do. We control the test activities, we report on testing progress and the status of the software under test.
- **Preparation**: We need to choose what testing we will do, by selecting test conditions and designing test cases.
- **Evaluation**: During evaluation we must check the results and evaluate the software under test and the completion criteria, which helps us to decide whether we have finished testing and whether the software product has passed the tests.
- **Software products and related work products:** Along with the testing of code the testing of requirement and design specifications and also the related documents like operation, user and training material is equally important.

Testing Activities

- Planning and control
- Choosing test conditions
- Designing test cases
- Checking results
- Evaluating completion criteria
- Reporting on the testing process and system under test
- Finalizing or closure (e.g. after a test phase has been completed)
- Testing also includes reviewing of documents (including source code) and static analysis

3) what is Agile Methodology ?

- Agile SDLC model is a combination of iterative and incremental process model. Agile Methods Break the Product into small incremental builds.

- Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks.

- Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

- At the end of the iteration a working product is displayed to the customer and important stakeholders.

- Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

- Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

4) what is SRS?

- A software requirements specification (SRS) is a complete descriptionof the behavior of the system to be developed.

- It includes a set of use cases that describe all of the interactions that the users will have with the Software.

- Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional (or supplementary) requirements.

- Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).

- Recommended approaches for the specification of software requirements are described by IEEE 830-1998. This standard describes possible structures, desirable contents, and qualities of a software requirements specification.

Types Of Requirements:

- Requirements are categorized in several ways. The following are common categorizations of requirements that relate to technical management:

    - Customer Requirements
    - Functional Requirements
    - Non-Functional Requirements

5) what is OOPS ?

- object oriented Programming is way of writing the programs in organized way.

- object like a Black box where Data are Hidden.

    - Secure
    - less code Redundancy
    - less Memory occupy (Fast)

- Object is derived from abstract data type
- Object-oriented programming has a web of interacting objects, each house-keeping its own state.
- Objects of a program interact by sending messages to each other.

6) Write Basic Concepts Of OOPS?

- Class
- Object
- Inheritance
- Polymorphism
    - Overriding
    - Overloading
- Encapsulation
- Abstraction

7) what is Object ?

- Object Gives the permission to Acess Functionlity Of Class.
- An object represents an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain.
- An "object" is anything to which a concept applies.
- This is the basic unit of object oriented programming(OOP).
- That is both data and function that operate on data are bundled as a unit called as object.

8) What is Class ?

- Class is a Collection of Data Member And Member Function.

E.x, Jio Sim,Calling,Data,Branch,car,Fan

- When you define a class, you define a blueprint for an object.
- A class represents an abstraction of the object and abstracts the properties and behavior of that object.
- An object is a particular instance of a class which has actual existence and there can be many objects (or instances) for a class.

9) what is Encapsulation ?

- The Process Wraping The Data in Singel Unit To Secure the Data From Outside World.

- Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.
- Encapsulation in Java is the process of wrapping up of data (properties) and behavior (methods) of an object into a single unit; and the unit here is a Class (or interface).

E.x, Medicine Capsule.

10) what is Inheritance ?

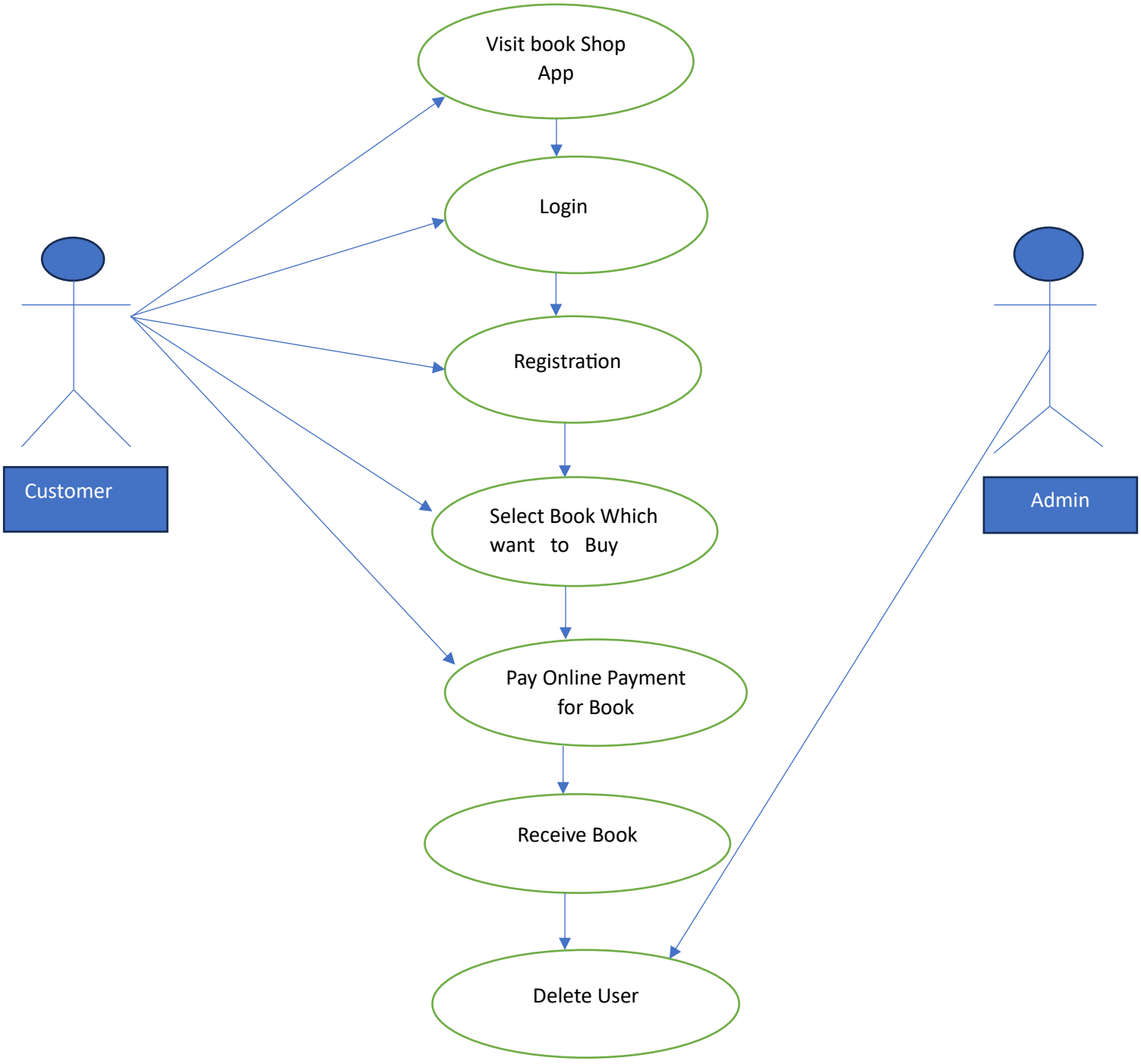
- Inheritance means that one class inherits the characteristics of another class. This is also called a "is a" relationship.
- One Name Multiple From.
- This is a very important concept of object-oriented programming since this feature helps to reduce the code size.
- Inheritance describes the relationship between two classes. A class can get some of its characteristics from a parent class and then add unique features of its own.
- Making A Class From an Existing Class.

E.x, Child Class Property, Code

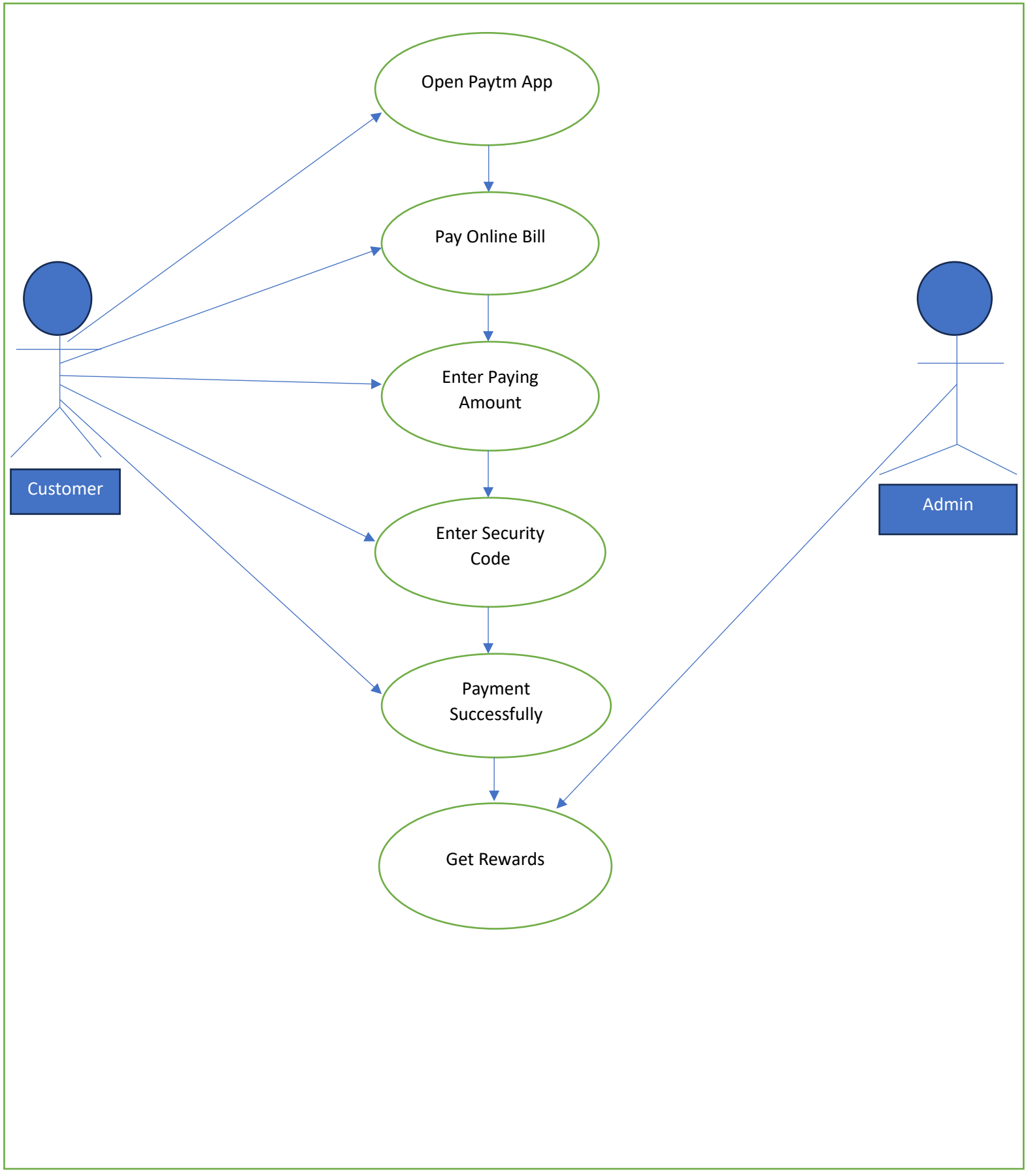
11) what is Polymorphism ?

- One Name Multiple From. Polymorphism Means “having many forms”.
- It allows different objects to respond to the same message in different ways, the response specific to the type of the object.
- The most important aspect of an object is its behaviour (the things it can do). A behaviour is initiated by sending a message to the object (usually by calling a method).
- The ability to change form is known as polymorphism.
- There is two types of polymorphism in Java:-
    - Compile time polymorphism(Overloading)
    - Runtime polymorphism(Overriding)

12) Draw Use Case on Online Book Shopping.


Visit book Shop App
Login
Registration
Select Book Which want to Buy
Pay Online Payment for Book
Receive Book
Delete User
Customer
Admin

13) Draw Use case on Online Bill Payment System (Paytm).



14) write SDLC Phases With Basic Introduction.

- Requirement - Establish customer needs.
- Analysis - Model And Specify the Requirements "What".
- Design - Model And Spcecify a Solution In Software – “Why”
- Implementation - Construct a Solution In software.
- Testing - Validate the solution Against the Requirements.
- Maintenance - Repair Defects And Adapt the solution to the New Requirements.

15) Explain Phases Of the WaterFall Model.

- Requirements Analysis
- System Design
- Implementation
- Testing
- Deployment
- Maintenance

16) Write Phases Of Spiral Model.

- Budget Constaint And Risk evalution is More Important.
- Long Term Project Commitment Priorities As the Requirement Change With Time.

- Customer Is Not Sure of Their Requirements Which Are Usaully the case.

17) Write Agile Manifesto Principles.

- Customer Statisfaction Though early and Continuous Software Delivery.
- Accommodate Changing Requirement Through the Development Process.
- Frequent Delivery of Working Software.
- Regular Reflaction On How to Become More Effective.
- Enable Face To Face Interactions.

18) Explain working Methodology of Agile Model and also Write Pros and Cons.

- Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
- Agile Methods break the product into small incremental builds.
- These builds are provided in iterations.
- Each iteration typically lasts from about one to three weeks.
- Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
- At the end of the iteration a working product is displayed to the customer and important stakeholders.

- **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.

- **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 documentation.

19)Draw Usecase on Online shopping product using COD.

ONLINE SHOPPING PRODUCT

20)Draw Usecase on Online shopping product using payment gateway.

ONLINE PAYMENT GATEWAY