(1) What is SDLC?

~ The Software Development Life Cycle (SDLC) is a process used in software development for planning, implementation, documentation, deployment, and ongoing maintenance and support.
~ It provides a framework for a standard set of activities and deliverables.
~ It is also called the Application Development Life Cycle.

(2) What is Software Testing?

~ Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is **Defect** or **Bug** free.
It can be done with two ways:
a) Manual Testing
b) Automation Testing

(3) What is an agile methodology?

~ 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 products.
~ 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.

(4) What is SRS?

~ Software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfil the needs of all stakeholders (business, users).
~ A software requirements specification (SRS) is a complete description of the behaviour 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.

(5) What is oops?

~ Object Oriented Programming is a way of writing the program in an organised way.
~ Objects are like a black box where data is hidden.
~ Security
~ Less space Occupy.
~ Less code redundancy.

(6) Write Basic Concepts of oops?

~ Class
~ Object
~ Inheritance
~ Polymorphism
~ Encapsulation
~ Encapsulation
~ Abstraction

(7) What is an object?

~ An object is an instance of a Class. It contains properties and functions. They are like real-world objects. For example, your car, house, laptop, etc.

~ Object gives the permission to access functionality of class.

(8) What is class?

~ Class is a collection of data members and data functions.

~ A class is a way of organising information about a type of data so a programmer can reuse elements when making multiple instances of that data type—for example, if a programmer wanted to make three instances of Car , maybe a BMW, a Ferrari, and a Ford instance.

(9) What is encapsulation?

~ Encapsulation can be defined as wrapping up data and functions together into a single unit.

~ Wrapping them into a single unit helps in keeping them secure from outside access.

~ It helps with data hiding.

(10)What is inheritance?

~ Inheritance is a mechanism of acquiring the features and behaviours of a class by another class.

~ Inheritance is a way of representing real-world relationships between the two. Here's an example – car, bus, bike – all of these come under a broader category called Vehicle. That means they've inherited the properties of class vehicles, i.e., all are used for transportation.

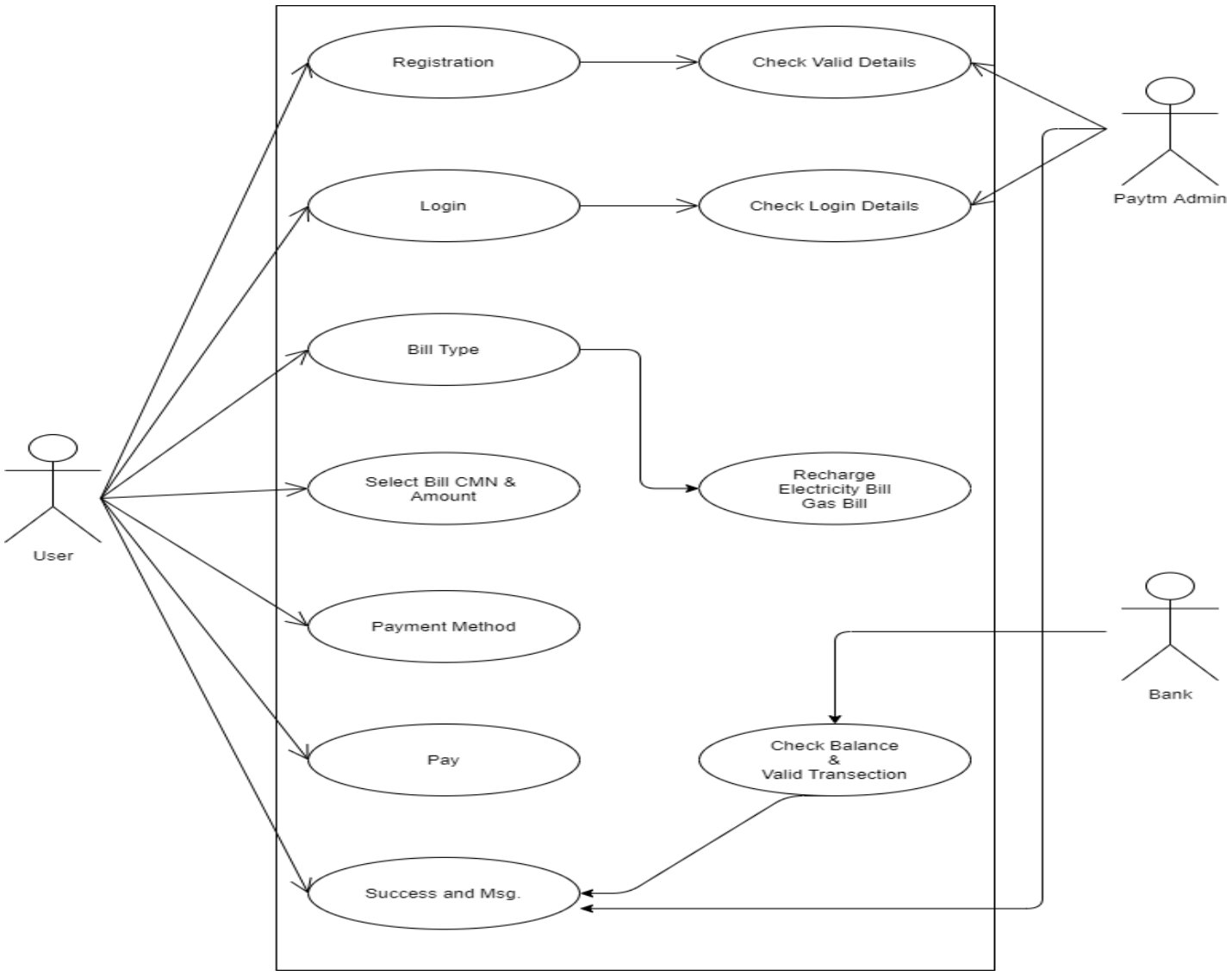
(11)What is polymorphism?

~ Polymorphism is one of the most important concepts in OOP. It describes the ability of something to have or to be displayed in more than one form.

~ Examples - Bird, Blood Group etc.

(12)Draw Usecase on Online book shopping?

(13)Draw Usecase on online bill payment system (paytm)?



(14)Write SDLC phases with basic introduction?

~ SDLC has 6 phases.

(a) Requirement collection / Gathering

In this phase collects all requirements that a customer needs in a software product.

(b) Analysis

In this phase Analyser, Analysing the requirements of customer, according to requirement analyser select the model for product and, required time and budget of project and completes the documentation of the project.

It's defined as the "What" phase.

(c) Design

The technical details of the design are discussed with the stakeholder and various parameters such as risks, technologies to be used, capability of the team, project constraints, time and budgets are reviewed and then the best design approach is selected for the product.

(d) Software Implementation

In this phase developers start the coding to develop software after getting all documents by the above 3 phases.

(e) Testing

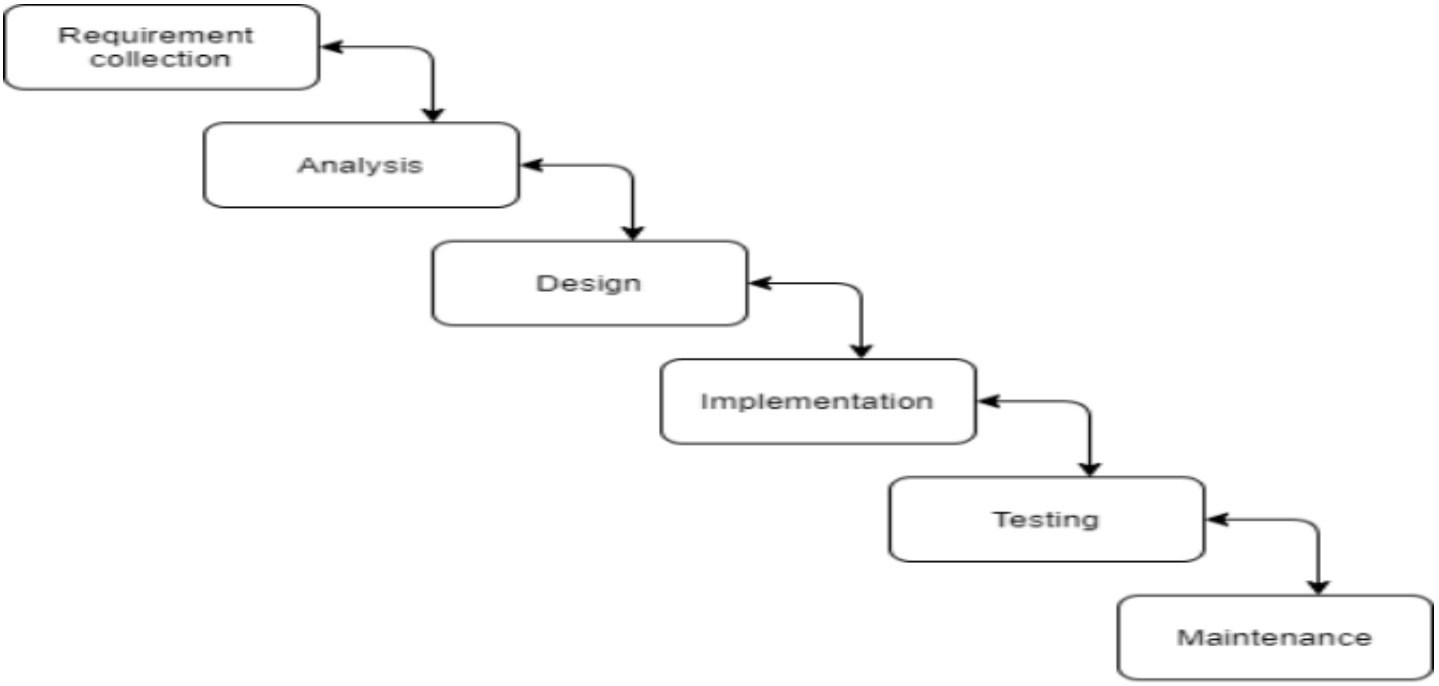
After develop software, it comes in Testing phase. In this phase the tester checks all Requirements that stakeholder wants. And testers find the bugs to deliver accurate Software.

(f) Maintenance

The maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system's performance, correct problems, enhance security, or address user requirements.

(15)Explain Phases of the waterfall model?

~ Waterfall model illustrates the software development process in a linear sequential flow. This Means that any phase in the development begins only if the previous phase is complete. Therefore, phases do not overlap.



Waterfall model

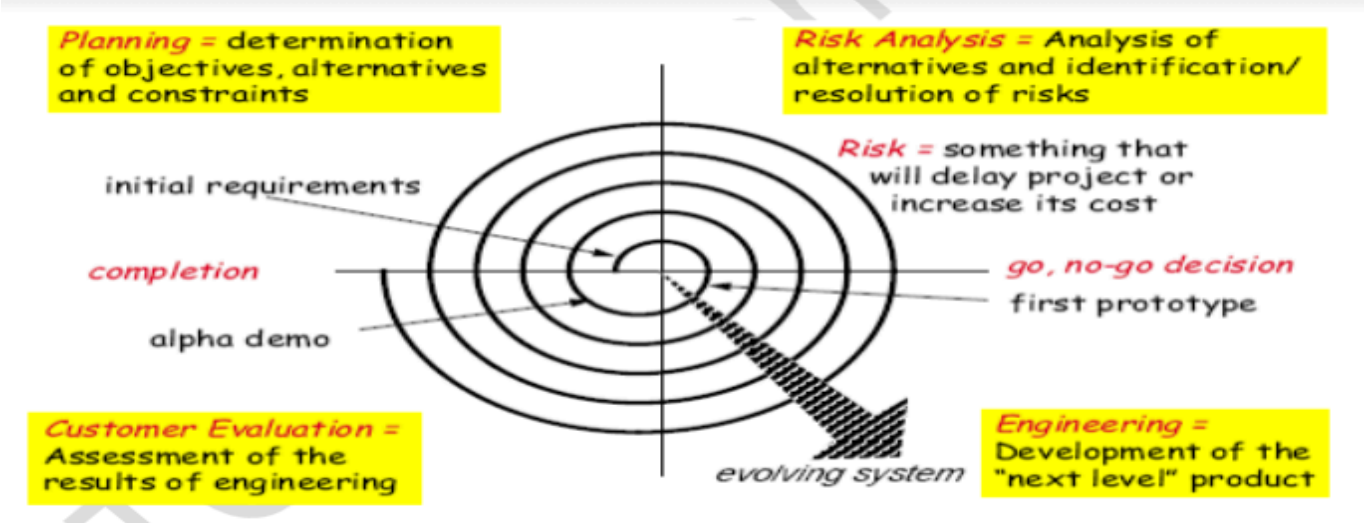
(16)Write phases of the spiral model?

~ Spiral model is iterative in nature and it is also called the version control model.

~ It allows incremental releases of the product or incremental refinement through each iteration Around the spiral.

~ It overcomes the limitation / drawback of the waterfall model.

~ It is best suited for large projects or where there is dependency in the model.



There are 4 phases in spiral model:

a) Planning

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

b) Risk Analysis

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.

c) Engineering

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.

d) Customer Evaluation

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.

(17)Write agile manifesto principles?

~ Agile manifesto principles
~ **Individuals and interactions -** in agile development, self-organisation 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 requirements, 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.

(18)Explain the working methodology of the agile model and also write pros and cons?

~ The Agile software development methodology is one of the simplest and effective processes to turn a vision for a business need into software solutions. Agile is a term used to describe software development approaches that employ continual planning, learning, improvement, team collaboration, evolutionary development, and early delivery. It encourages flexible responses to change.
~ **The agile software development emphasises four core values.**
1. Individual and team interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

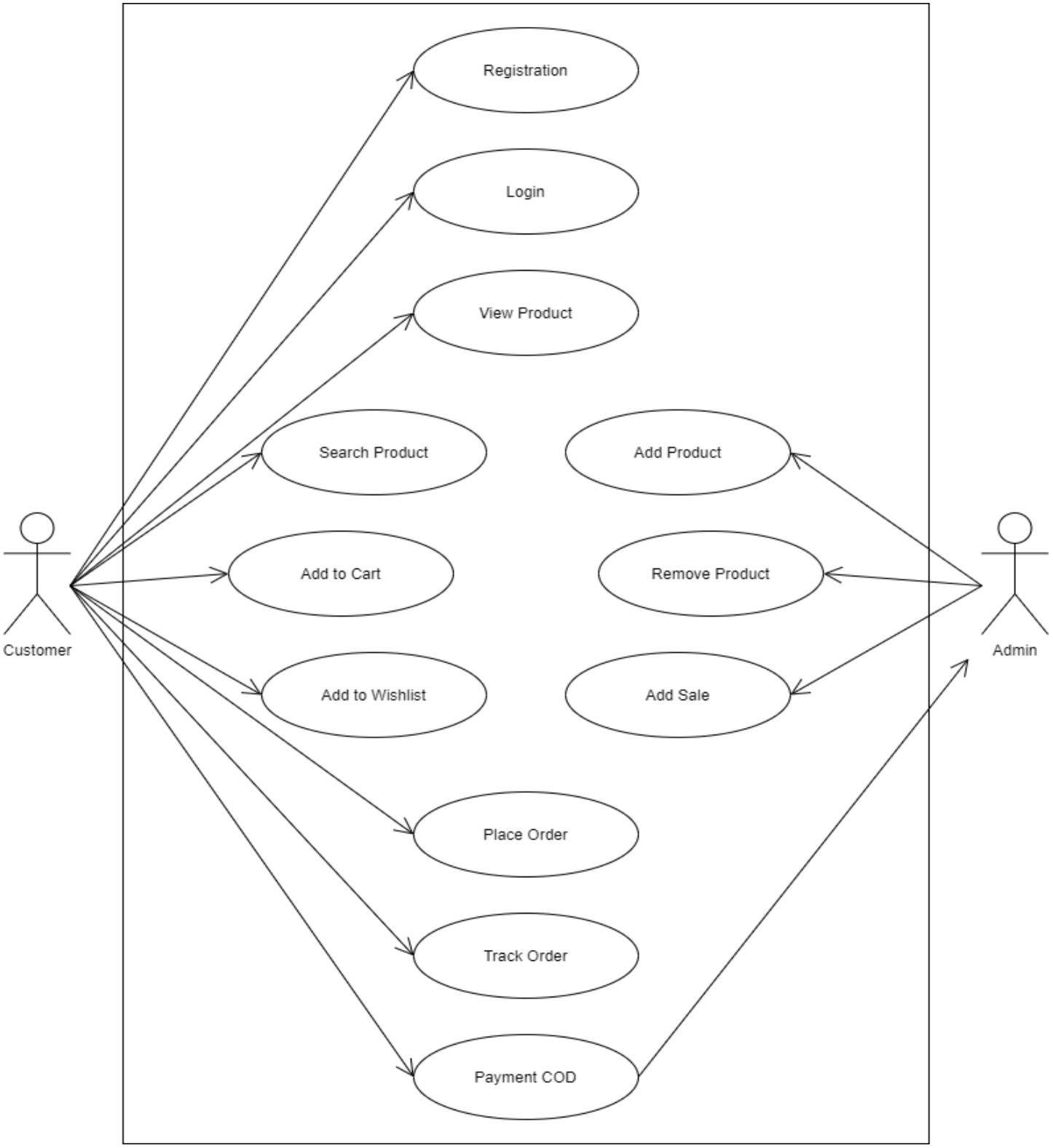
Pros.

~ Is a very realistic approach to software development
~ Promotes teamwork and cross training.
~ Resource requirements are minimum.
~ Suitable for fixed or changing requirements
~ Little or no planning required
~ 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

(19)Draw usecase on Online shopping product using COD?



(20)Draw usecase on Online shopping product using payment gateway?