1.What is SDLC?

Software Development Life Cycle is essentially a series of steps or phases that can provide a module for the development life cycle management of application or piece of software.

2. What is Software Testing?

Software Testing is process of evaluating a system or its components with the intend to find whether it satisfies the specified requirements or not.

3. What is Agile Methodology?

The Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement. Teams follow a cycle of planning, executing, and evaluating.

4. What is SRS?

Software Requirement Specification is a description of an application which is to be developed. It contains use case diagram that describes all the interaction user will have with the software application.

5. What is OOPS?

Object Oriented Programming is way of writing the programs in organized way. Objects are like a black box where data are hidden.

6.Write Basic Concepts of OOPS?

The basic concepts of OOPS are

1. Class
2. Object
3. Inheritance
4. Polymorphism
5. Encapsulation
6. Abstraction

7. **What is Object?**

   Object gives permission to access functionality of class.

8. **What is Class?**

   Class is a collection of data member and member function.

9. **What is Encapsulation?**

   The process wrapping the data in a single unit. To secure the data from outside world.

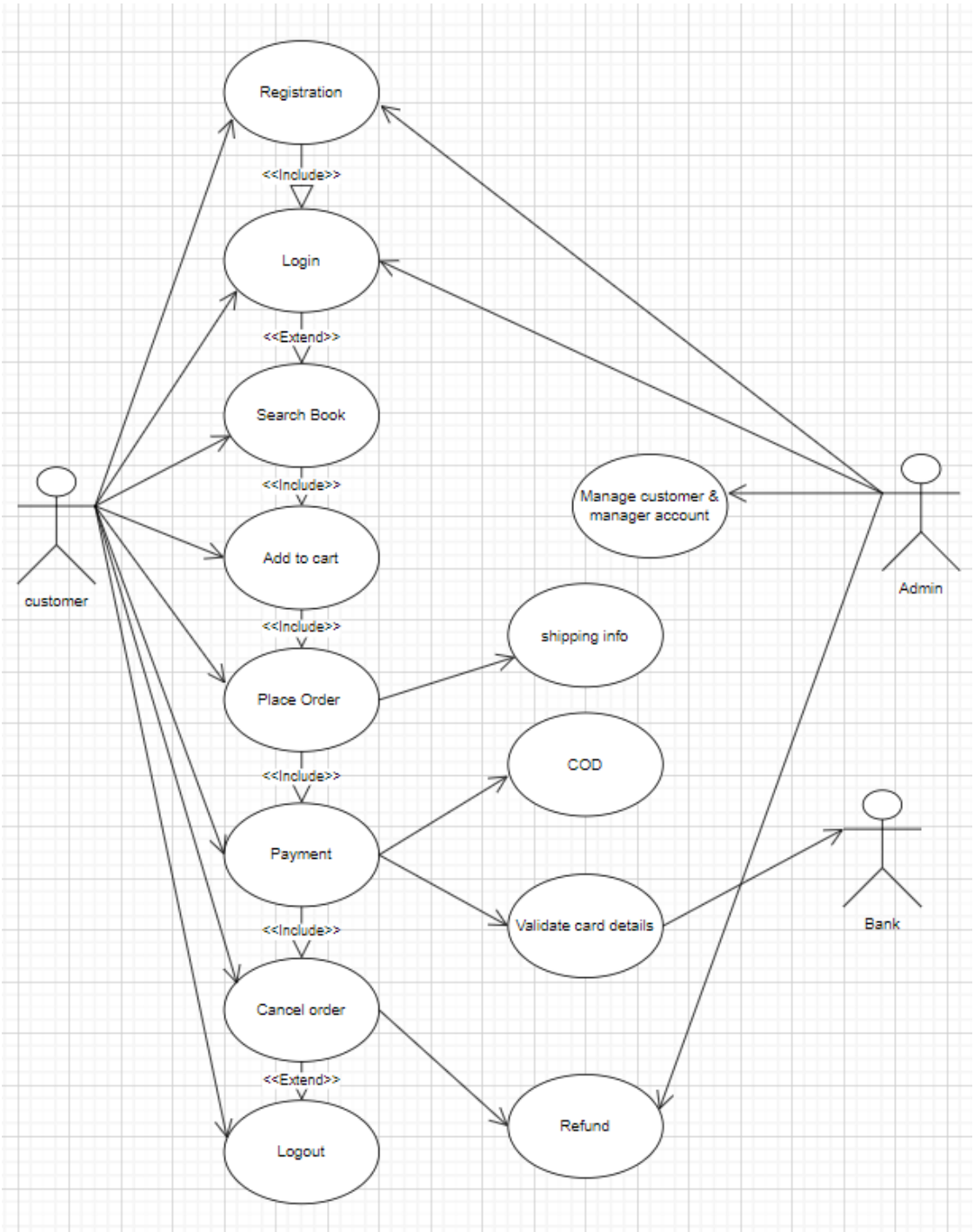
10. What is Inheritance?

Making a class from an existing class. Deriving the attributes of some other class.

11. What is Polymorphism?

One name multiple form.

12. Draw usecase on Online book shoping?



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

14. Write SDLC phases with basic introduction ?

- A software development life cycle is essentially a series of steps or phases that provide a model for the development & life cycle management of an application of software. There are six phases in SDLC.

  (i). Requirements collection gathering -- > Establish customers needs

  (ii). Analysis -- > Model & specify the requirements – “What”

  (iii). Design -- > Model & specify a solution – “Why”

  (iv). Implementation -- > Construct the solution in software

(v). Testing -- > Validate the solution against the requirements

(vi). Repair defects & adopt the solution to the new requirements

(i). Requirements collection gathering :-

- Gathering information about the software requirements from stakeholders, such as customers, end-users, & business analysts.

- Three types of problem can arise :-

(a). Lack of clarity

(b). Requirements confusion

(c). Requirements amalgamation

- Types of requirements :-

(a). Functional requirements

(b). Non-Functional requirements

(ii). Analysis phase :-

- The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.
- This phase defines the problem that the customer is trying to solve.
- Ideally, this document states in a clear & precise fashion what is to be built.
- This analysis represent the “What” phase.
- This phase starts with the requirements document delivered by the requirement phase & maps the requirements into architecture.

(iii). Design phase :-

- Design architecture document
- Implementation plan
- Critical priority analysis
- Performance analysis
- Test plan
- The architecture team also converts the typical scenarios into a test plan.

(iv). Implementation phase :-

- In the implementation phase, the team builds the components either from scratch or by composition.
- Eg :- A component may be narrowly designed for the particular system, or the component may be made more general to satisfy a reusability guideline (a). Implementation code (b). Critical error removal

(v). Testing phase :-

- Simply stated, quality is very important many companies have not learned that quality is important & deliver more claimed functionality But at a lower quality level.
- It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality.
- A customer satisfied with the quality of a product will remain loyal & wait for new functionality in the next version,
- Quality is a distinguishing attribute of a system indicating the degree of excellence.

(vi). Maintenance phase :-

- Maintenance is the process of changing a system after it has been deployed.
- (a). Corrective maintenance :- Identifying & repairing defect
- (b). Adaptive maintenance :- Adapting the existing solution to the new platforms
- (c). Perfective maintenance :- Implementing the new requirements

15. Explain phases of the waterfall model ?

- The classical software life cycle model the software development as a step- by-step "waterfall" between the various development phases

(i). Requirements collection

(ii). Analysis

(iii). Design

(iv). Implementation

(v). Testing

(vi). Maintenance

- The waterfall is unrealistic for many reasons especially :- Requirement must be "frozen" to early in the life cycle. Requirements validated late.

(i). Requirements collection :-

- Gathering information about the software requirements from stakeholders, such as customers, end-users, & business analysts.

- Three types of problem can arise :-

(a). Lack of clarity

(b). Requirements confusion

(c). Requirements amalgamation

- Types of requirements :-

(a). Functional requirements

(b). Non-Functional requirements

(ii). Analysis :-

- The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.
- This phase defines the problem that the customer is trying to solve.
- Ideally, this document states in a clear & precise fashion what is to be built.
- This analysis represent the "What" phase.
- This phase starts with the requirements document delivered by the requirement phase & maps the requirements into architecture.

(iii). Design :-

- Design architecture document
- Implementation plan
- Critical priority analysis
- Performance analysis
- Test plan
- The architecture team also converts the typical scenarios into a test plan.

(iv). Implementation :-

- In the implementation phase, the team builds the components either from scratch or by composition.
- Eg :- A component may be narrowly designed for the particular system, or the component may be made more general to satisfy a reusability guideline (a). Implementation code (b). Critical error removal

(v). Testing phase :-

- Simply stated, quality is very important many companies have not learned that quality is important & deliver more claimed functionality But at a lower quality level.
- It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality.
- A customer satisfied with the quality of a product will remain loyal & wait for new functionality in the next version,
- Quality is a distinguishing attribute of a system indicating the degree of excellence.

(vi). Maintenance phase :-

- Maintenance is the process of changing a system after it has been deployed.

- (a). Corrective maintenance :- Identifying & repairing defect
- (b). Adaptive maintenance :- Adapting the existing solution to the new platforms

(c). Perfective maintenance :- Implementing the new requirements

16. Write a phases of spiral model ?

- In spiral model there are four phases :-

(a). Planning :- Determination of objectives, alternatives & constraints

(b). Risk analysis :- Analysis of alternatives & identification/ resolution of risks

(c). Engineering :- Development of the "next level" product

(d). Customer evaluation :- Assessment of the results of engineering

17. Write agile manifesto principles ?

- Agile manifesto principles :-

(a). Customer satisfaction through early & continuous software delivery

(b). Accommodate changing requirements throughout the development process

(c). Frequent delivery of working software

(d). Collaboration between the business stakeholders & developers throughout the project

(e). Support, trust, & motivate the people involved

(f). Enable face-to-face interactions

(g). Working software is the primary measure of progress

(h). Agile processes to support a consistent development pace

(i). Attention to technical detail & design enhances agility

(j). Simplicity

(k). Self-organizing teams encourage great architectures, requirements, & design

(l). Regular reflections on how to become more effective

18. Explain working methodology of agile model & also write pros & cons ?

- It is a combination of iterative & incremental model & it's working methodology is given below :-

(a). It divides the software into small incremental builds are provided in iterations, that means the big projects are divided into small chunks.

(b). Each iteration last about two to four weeks.

(c). Each iteration involves all the team members working on simultaneously on areas like planning, requirement analysis, design, coding, unit-testing & acceptance testing.

(d). At the end of the iteration the working product is displayed to the customer or the important stakeholders & it is released in the market.

(e). After the release we check for the feedback of the deployed software.

(f). If any enhancement is needed in the project then it's done & it's re-released.

Pros :-

(a). Frequent delivery

(b). Face to face communication with the customer

(c). Less time

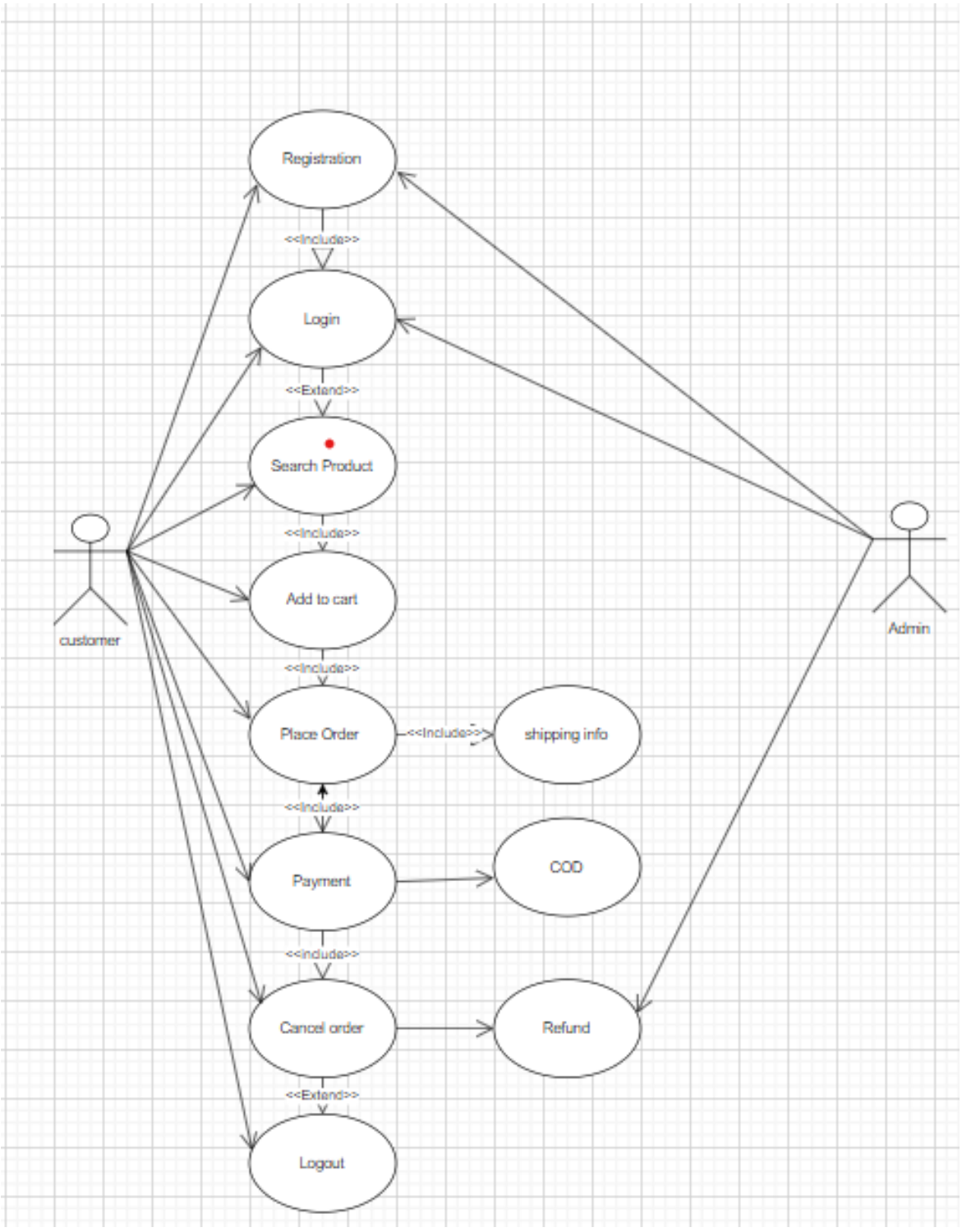
(d). Adaptability

Cons :-

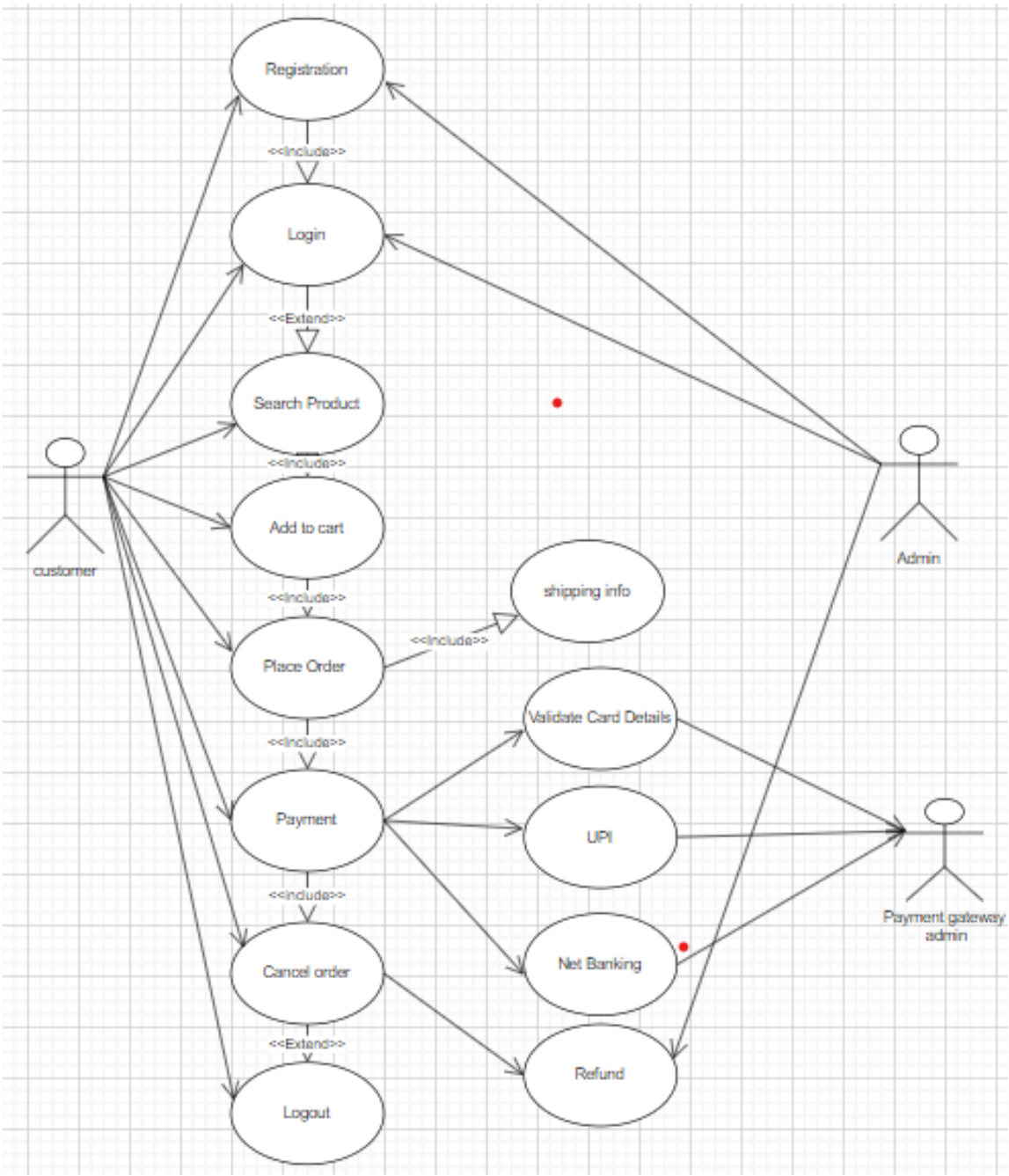
(a). Less documentation

(b). Maintenance problem

19. Draw use-case on online shopping product using COD ?



20. Draw use-case on online shopping product using payment gateway ?


Registration
<<Include>>
Login
<<Extend>>
Search Product
<<Include>>
Add to cart
<<Include>>
Place Order
<<Include>>
Payment
<<include>>
Cancel order
<<Extend>>
Logout
customer
shipping info
Validate Card Details
UPI
Net Banking
Refund
Admin
Payment gateway admin