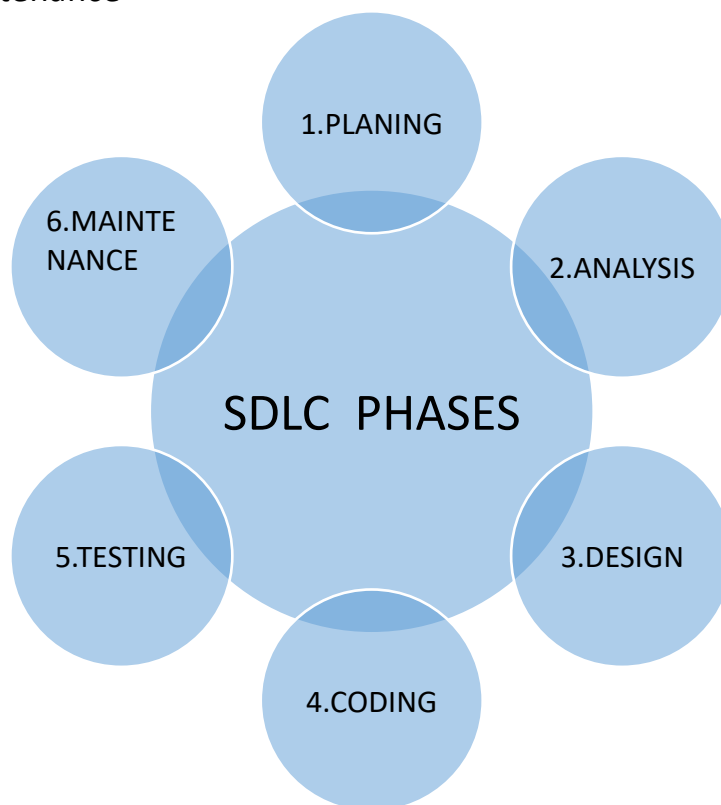


Assignment -1 [Fundamentals]

1..what is SDLC?

SDLC is a methodology or step by step approach to produce software with high quality, lowest cost in the shortest possible time by defining the phases like.....

- planing
- Analysis
- Design
- Coding & implementation
- Testing
- Maintenance



2.What is software testing ?

- software testing is process which is use to identify the correctness, completeness,quality of the developed software.
- In simpal word, testing is executing a system in order to identify any gaps,errors or missing requirement in contrary to actual desire or requirement.
- Test activities exist before and after test execution

- Software testing is a method to check whether the actual software product matches expected requirement and to ensure that software product is defect
- Testing types :
-
- 1) manual testing
- 2) automation testing

3. What is 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 product.
- Agile model is a very realistic approach.
- This model is a functionality can be developed rapidly

4. what is SRS ?

- SRS full form : Software requirement specification .
- ❖ SRS is a complete description of the behaviour of the system to be developed.
- Types of requirements :
- 1. customer requirement
- 2. functional requirement
- 3. Non- functional requirement

5. What is OOPS ?

- Object oriented programming is viewed as a collection of objects.
- It is used to structure the software program into simple reusable code .
- Here it is referred as functional testing or black box testing .
- OOPS Basic concepts :
- ❖ Class
- ❖ Object
- ❖ Encapsulation
- ❖ Polymorphism
- ❖ Inheritance

6. write basic concept of OOPS ?

- ❖ Class
- ❖ Object
- ❖ Encapsulation

- ❖ Polymorphism
- ❖ Inheritance

7. what is object ?

➤ Object :

An object is the basic unit of OOP which is accessed by its properties called data member & member function.

➤ It creates the memory for the class

8. what is class ?

➤ Class :

Class is collection of a data member (variables) and member function with its behavior .

➤ Class is blueprint a template to describe the properties and behavior of the objects.

9. What is Encapsulations ?

➤ Encapsulations :

A wrapping up of data and functions into a single unit is called encapsulation .

➤ It hide/ included private access of data member & member function .

10. What is inheritance ?

➤ Inheritance :

one class (super , base) inherits the properties of another class (sub, derived)

❖ Type of Inheritance :

1. Single Inheritance
2. Hybrid inheritance
3. Multilevel inheritance
4. Multiple inheritance
5. Hierarchical inheritance

11. What is polymorphism ?

➤ Polymorphism :

An ability to take one name having many different forms.

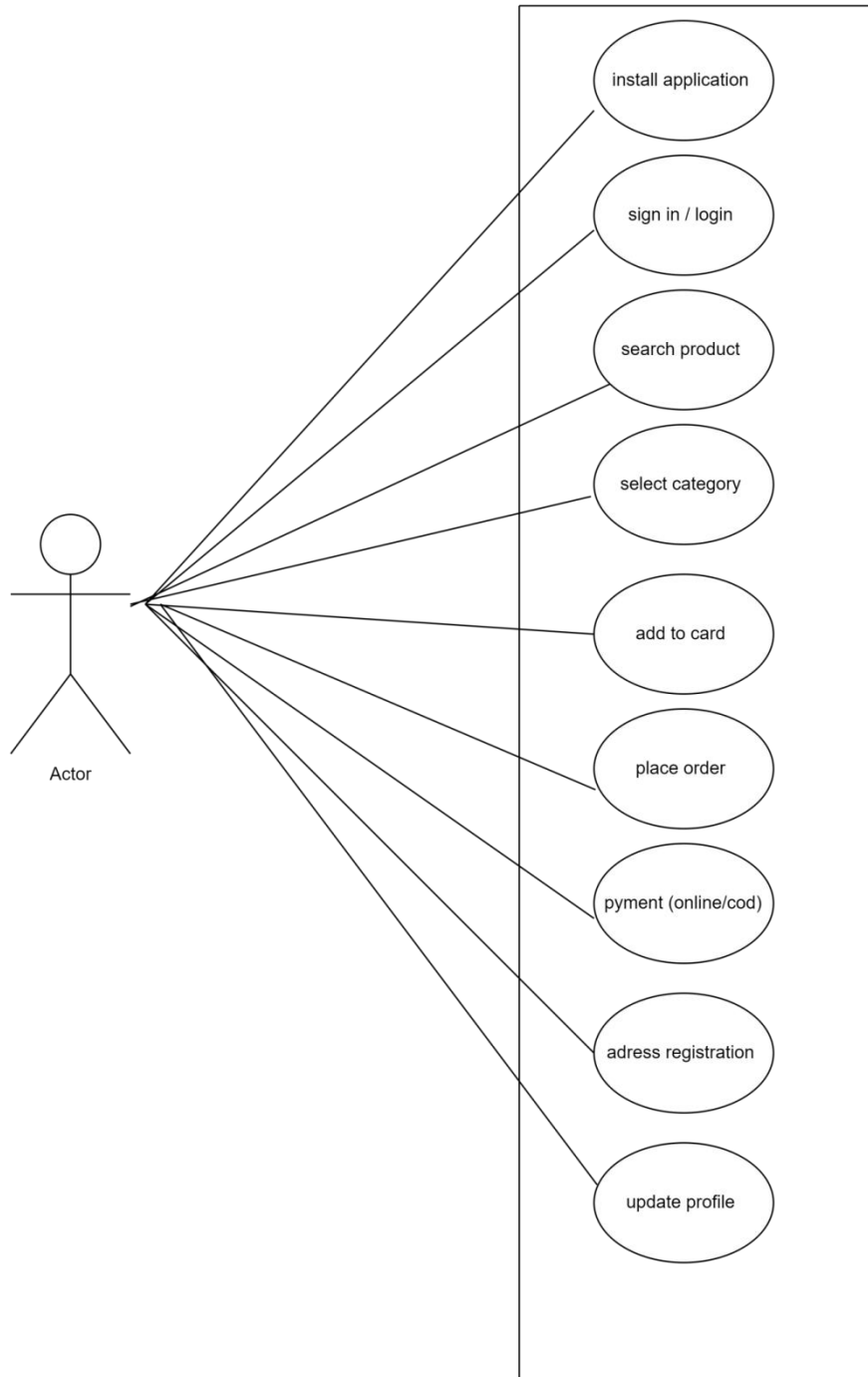
❖ Run time Polymorphism :

Method should be same in super class and sub class but its behavior is different.

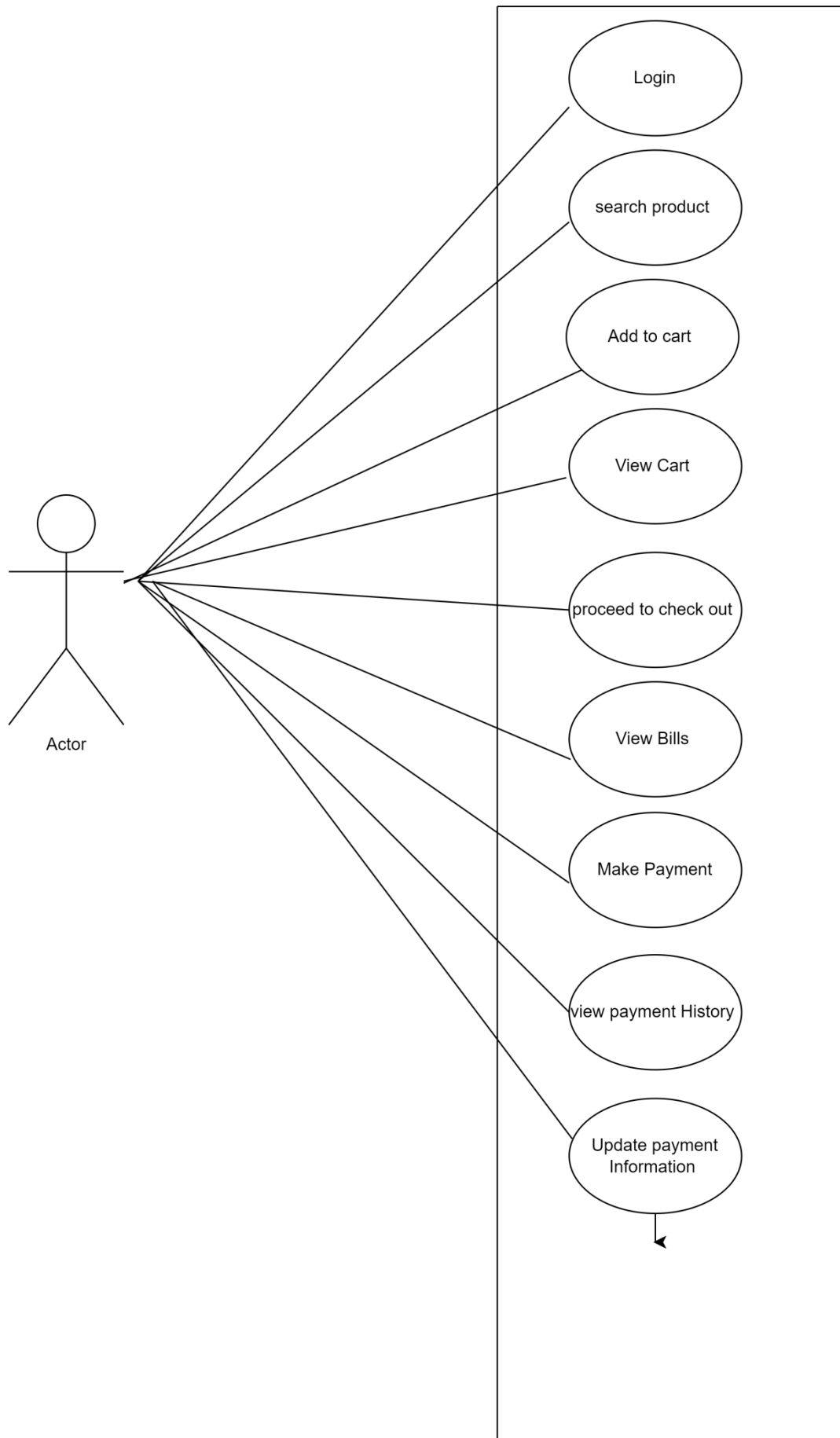
❖ compile time Polymorphism :

Method name should be same in single class but its behavior (arguments & data type) is different

12. Draw usecase on online book shopping on



13.draw usecase on online bill payment system (pytm)



14. write SDLC phases with basic introduction ?

- SDLC is a methodology or step by step approach to produce software with high quality, lowest cost in the shortest possible time by defining the phases like.....
- Requirement collection/ gathering
- Analysis
- Design
- Coding / implementation
- Testing
- Maintenance

❖ Basic introduction for phases.....

A.requirement collection / gathering :

- Establish customer needs
- Build continuous feedback into the project plan
- Early prototyping can help clarify the requirements
- types of problem Three can arise
 - 1.lack of clarity
 - 2.requirement confusion
 - 3.requirement amalgamation

B.Analysis phase :

- This analysis represents the **what** phase.
- The analysis phase defines the requirements of the system independent of how these requirements will be accomplished
- The deliverable design document is the architecture

C.Design phase :

- requirement documents must guide this decision process
- Low level design & high level design
- Design architecture document
- Test plan

D. Coding & implementation phase :

- The end deliverable is the product itself . there are already many established techniques with implementation
- Construct a solution in software

E.Testing phases :

- Testing phases is a separate phase which is performed by a different team after the implementation is complete.

E. Maintenance phases :

- The developing organization or team will have some mechanism to document and track defects and deficiencies.
- Configuration and version management

- Updating all analysis, design, and user documentation

15. Explain phases of the waterfall model ?

- Waterfall model :

The waterfall model is a classical software lifecycle that models the software development as a step by step “waterfall” between the various development phases .

- The waterfall is unrealistic for many reasons, especially:
 - > Requirements are validated too late
 - > Requirements must be “frozen” too early in the life cycle

- ❖ Waterfall phases :

1. Requirements collection
2. Analysis
3. Design
4. Implementation
5. Testing
6. Maintenance

- ❖ Requirements collection :

- Early prototyping can help clarify the requirements . types of problem that arise
 1. lack of clarity
 2. requirement confusion
 3. requirement amalgamation

- ❖ Analysis :

- This analysis represents the **what** phase.
- The analysis phase defines the requirements of the system independent of how these requirements will be accomplished .

- ❖ Design :

- > requirement documents must guide this decision process.
- > Low level design & high level design

- ❖ Implementation :

The end deliverable is the product itself . there are already many established techniques with implementation .

- ❖ Testing :

Testing phases is a separate phase which is performed by a different team after the implementation is complete.

❖ Maintenance :

The developing organization or team will have some mechanism to document and track defects and deficiencies.

16. Write phases of spiral model ?

➤ Spiral model :

Spiral model is very widely used in the software industry as it is in synch with the natural development process of any product. Learning with maturity also involves minimum risk for the customer as well as the development firms.

➤ Planning :

➤ Risk analysis

➤ Engineering

➤ Customer evaluation

❖ Planning :

Determination of objectives , alternatives and constraints .

❖ Risk analysis :

Analysis of alternatives and identification / resolution of risk.

❖ Engineering :

Development of the “next level” product .

❖ Customer evaluation :

Assessment of the results of engineering .

17. Write agile manifesto principles ?

➤ Agile model :

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 manifesto principles :

- i. Individuals and interactions over processes and tools .
- I. Working software over comprehensive documentation
- II. Customer collaboration over contract negotiation.
- III. Responding to the change over following a plan.

18. Explain working methodology of agile model and also write pros and Cons .

➤ Agile model :

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.

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

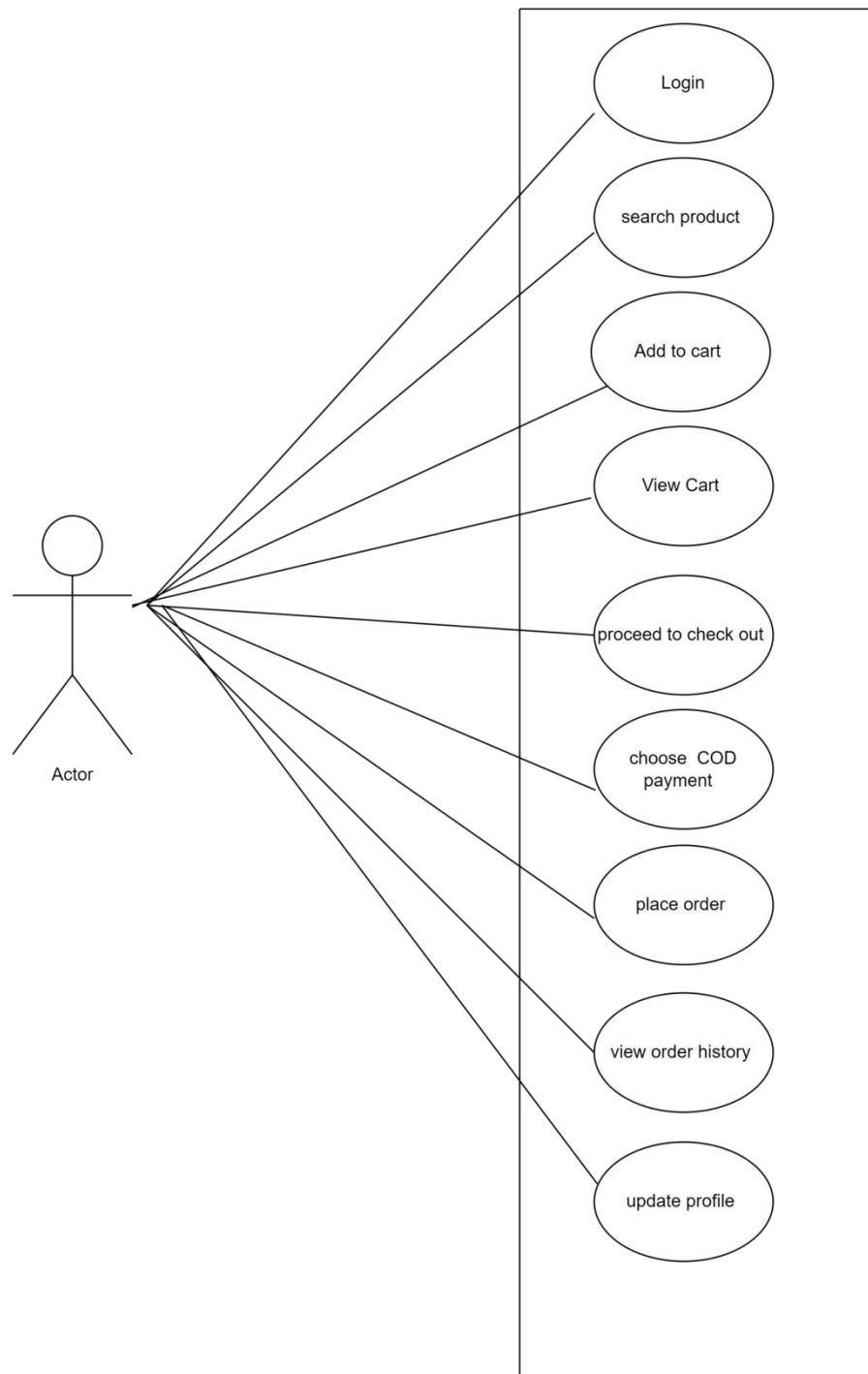
◆ Pros :

1. Functionality can be developed rapidly.
2. Very realistic approach.
3. Promotes teamwork and cross training.
4. Gives flexibility to developers.
5. Rapid delivery.
6. Little or no planning required.

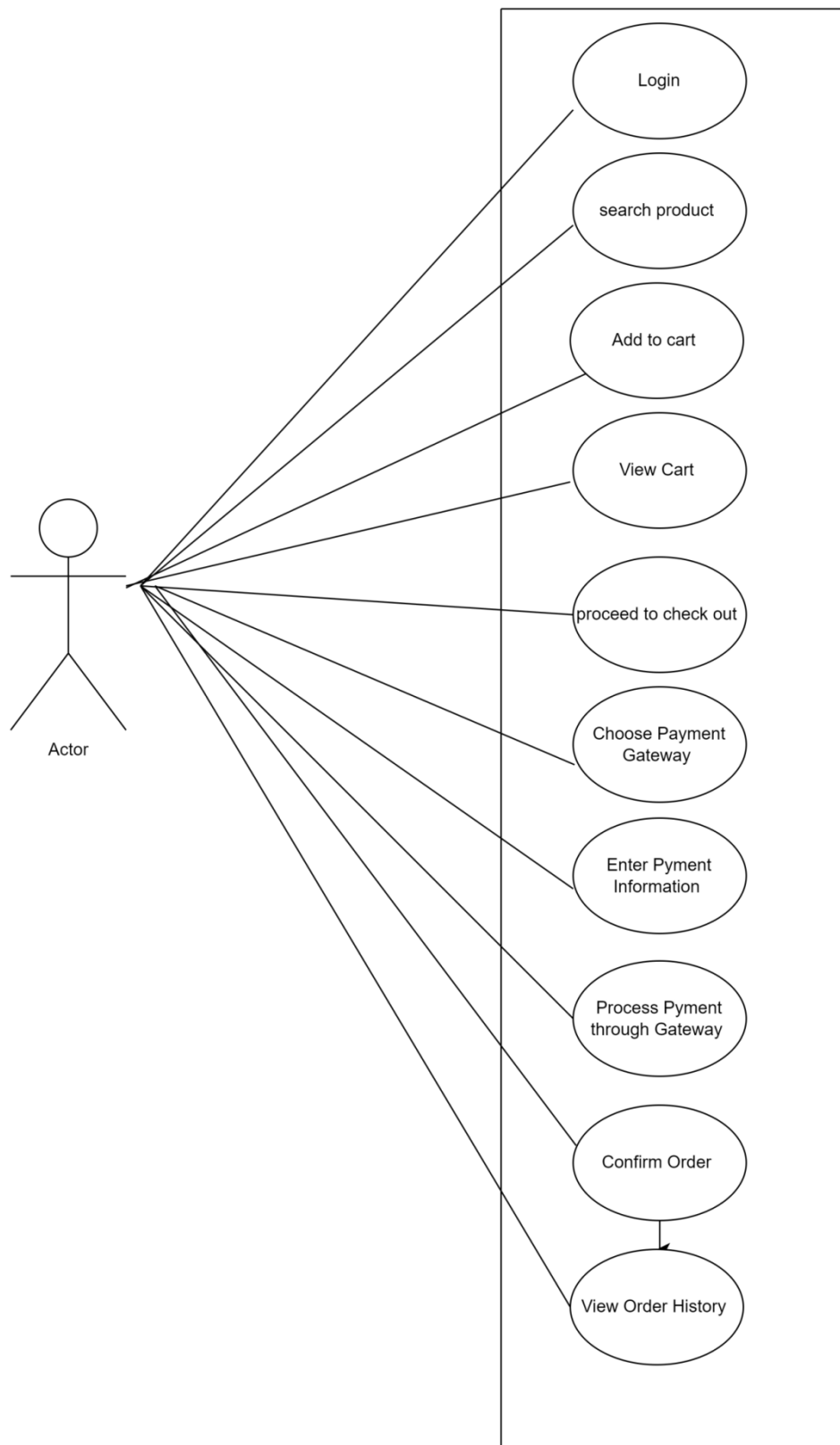
◆ Cons :

1. Not useful for small projects.
2. Depends heavily on customer interactions.
3. More risk of sustainability, maintainability and extensibility.
4. Not suitable for handling complex dependencies

19. Draw use case on online shopping product using COD C



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



21. What are 7 key principles? Explain in details?

1. Testing shows presence of Defects

2. Defect Clustering

3. Absence of Errors Fallacy

4. Exhaustive Testing is Impossible!

5. Early Testing

6. The Pesticide Paradox

7. Testing is Context Dependent

● Testing shows presence of Defects :

- ❖ > Testing can show that defects are present, but cannot prove that there are no defects.
- ❖ > As we find more defects, the probability of undiscovered defects remaining in a system reduces.
- ❖ > Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.

● Defect Clustering :

- ❖ A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.
- ❖ > Defects are not evenly spread in a system
- ❖ > They are 'clustered'
- ❖ > Similarly, most operational failures of a system are usually confined to a small number of modules.

● Absence of Errors Fallacy :

- ❖ > If the system built is unusable and does not fulfill the user's needs and expectations then finding and fixing defects does not help.
- ❖ > Even after defects have been resolved it may still be unusable and/or does not fulfil the users' needs and expectations.

● Exhaustive Testing is Impossible! :

- ❖ > Testing everything including all combinations of inputs and preconditions is not possible.
- ❖ > That is we must Prioritise our testing effort using a Risk Based Approach.

- ❖ > So, accessing and managing risk is one of the most important activities and reason for testing in any project.
- Early Testing :
 - ❖ > Testing activities should start as early as possible in the development life cycle.
 - ❖ Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.
- The Pesticide Paradox :
 - ❖ If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects
 - ❖ Testing identifies bugs, and programmers respond to fix them.
- Testing is Context Dependent :
 - > Testing is done differently in different contexts
 - > Also different industries impose different testing standards.
 - > Testing is basically context dependent.

22. Difference between Verification and Validation ?

Verification	Validation
1.verification is a process which is performed at development.	1.validation is a process which is performed at testing level.
2.it is a process of evaluating product of development to check whether the specified requirements meet or not.	2.it is a process of evaluating product of development to check whether it satisfied business requirements or not.

3.verification can be achieved by asking “are you building a product right ?”	3. validation can be achieved by asking “are you building a right product ?”
4. verification phases are : Business requiremet analysis System design / System requiremet Architectural design Module design	5. validation phases are : Unit testing Intergration testing System testing Acceptance testing