Software Testing Assignment

Module:-1

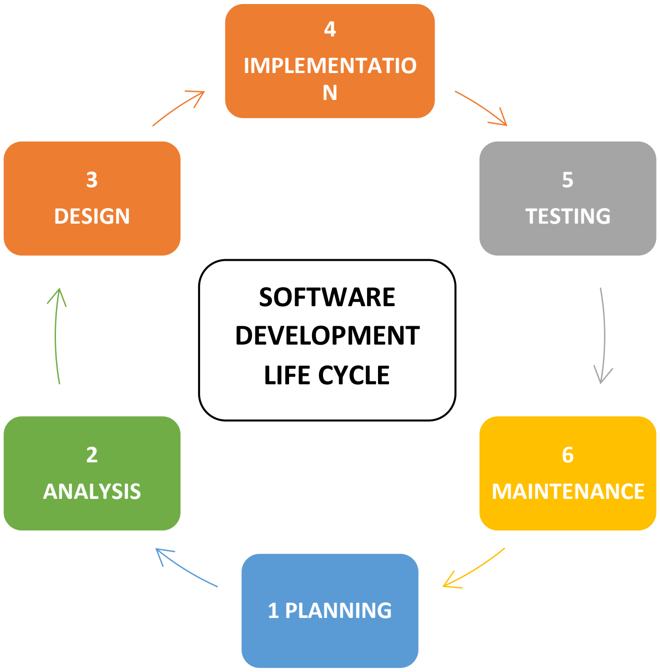
(1) What is SDLC?

- SDLC is Software Development Life Cycle.
- SDLC is a methodology or step-by-step approach to produce software with high quality, lowest coast in the shortest possible time by difining the phases like Planning, Analysis & Design, Coding & Implementation, and Testing & Maintenance.

  - **SDLC Phases:**

    1) Planning
    2) Analysis
    3) Design
    4) Coding & Implementation
    5) Testing & integration
    6) Maintenance

☺ <u>SDLC 6 Phases Diagram</u> ☺



(2) What is software testing?

- Testing is process which is used to identify the correctness, complitness, quality of the developed software.
- 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.
- Test activities exist before and after test execution

- **Testing Types:-**

  (1) Manual testing:

  - Execute test case manually by the tester without using any automation tool.

  (2) Automation:

  - Execute test case with using any automation tool.
    Ex. Selenium
  - Java, C#, Python etc. languages are used for automation testing script.

(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 Models Pros:

  1) Process adaptability & Customer Satisfation
  2) Rapid delivery
  3) Working software
  4) Frame work

- Agile Models Cons:

  1) More risk of sustainability.
  2) Not useful for small projects.
  3) Depends heavily on customer interactions.
  4) Very high individual dependency.

(4) What is SRS?

- SRS means Software Requirement Specification.

- SRS is a document that describes what the software will do and how it will be expected to perform. It also describes the functionaliyu the product needs to fulfill the needs of all users.

- Types of Requirements:

- ♣ Customer Requirement
- ♣ Functional Requirement
- ♣ Non-Functional Requirement

(5) What is oops?

Object-orianted programming is a core of java programming which is used for designing a program using classes and objects. Oops can also be characterized as data controlling for accessing the code.

- ➢ Here it is referred as functional testing or black box testing .
- ➢ OOPS basic concepts:
  - - Class
  - - Object
  - - Interface
  - - Abstact class
  - - Polymorphism
  - -Incpulation

(6) Write basic concepts of oops?

- ❖ Class
- ❖ Object
- ❖ Interface
- ❖ Abstact class
- ❖ Polymorphism
- ❖ Incpulation

(7) What is object?

Any entity that has state and behavior is known as an object. For example, a chair, pen, table, keyboard, bike, etc. It can be physical or logical.

- ➢ It creates the memory for the class.
- ➢

(8) What is class?

- Class is collection of a data member (veriables) and member function with its behavior.
- Collection of objects is called class. It is a logical entity.
- A class can also be defined as a blueprint from which you can create an individual object. Class doesn't consume any space.

(9) What is encapsulation?

Encapsulation is a protection mechanism for the data members and methods present inside the class. In the encapsulation technique, we are restricting the data members from access to outside world end-user.

- It hide/ include privet access of data member & member function.

(10) What is inheritance?

When one object acquires all the properties and behaviors of a parent object, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

-> In Java, we use method overloading and method overriding to achieve polymorphism

- Type of Inheritance :

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

(11) What is polymorphism?

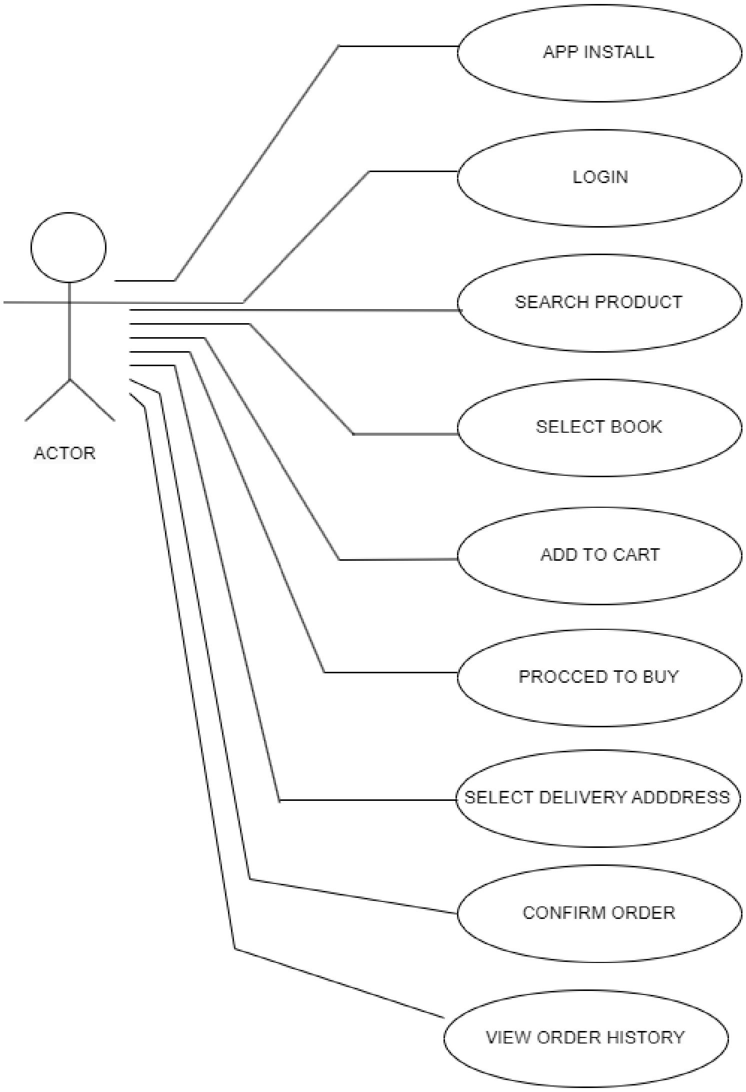
Polymorphism in Java is a concept by which we can perform a *single action in different ways*. Polymorphism is derived from 2 Greek words: poly and morphs. The word "poly" means many and "morphs" means forms. So polymorphism means many forms.

- **Two types of polymorphism**

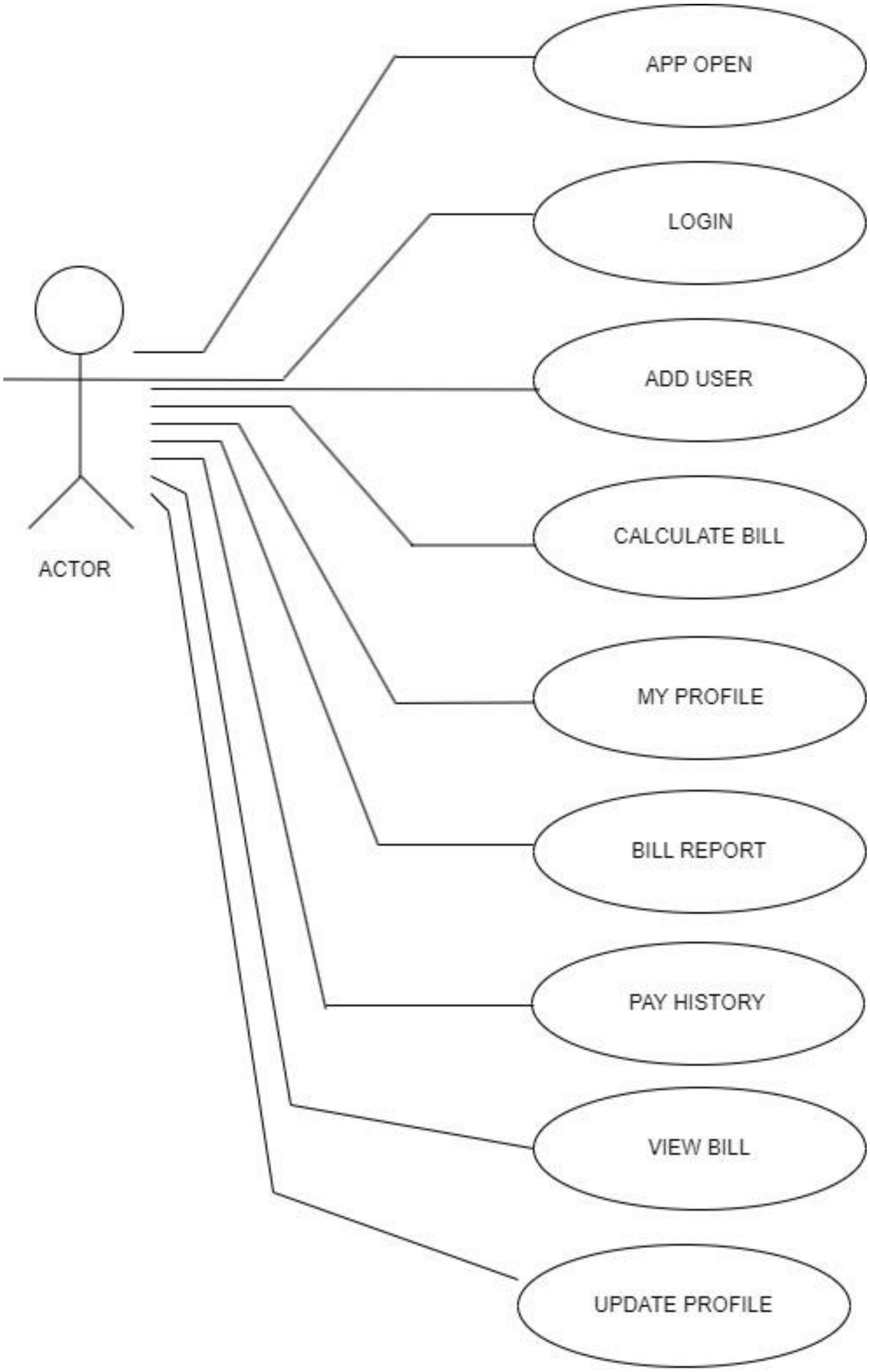
I) Runtime Polymorphism (operator overriding)

II) Compile time Polymorphism (operator overloading)

(12) Draw Usecase on Online book shopping.



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



(14) Write SDLC phases with basic introduction.

- SDLC is a methodology or step-by-step approach to produce software with high quality, lowest coast in the shortest possible time by difining the phases like Planning, Analysis & Design, Coding & Implementation, and Testing & Maintenance.

  - **SDLC Phases:**

    1) Planning
    2) Analysis
    3) Design
    4) Coding & Implementation
    5) Testing & integration
    6) Maintenance

- **Basic introduction of phases:**

  1) **Requirements Collection/Gathering:** The most important phase of the SDLC is the requirement gathering and analysis phase because this is when the project team begins to understand what the customer wants from the project

     ❖ Types of Requirements:

     1. Customer Requirements
     2. Functional Requirements
     3. Non-Functional Requirements

2) Analysis phase: The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.

- This analysis represents the **"what"** phase.

3) Design: Design phase is a stage where software developers define the technical details of the product.

* Low level design & high level design
* Design architecture document
* Test plan

4) Coding & implementation: In this phase, the tasks are divided into modules or units and assigned to various developers.

- Construct a solution in software.

5) Testing: A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version.

6) Maintenance: Corrective Maintenance, Adaptive Maintenance, Perfective Maintenance.

(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.

1) Requirements collection: There are three types of problems can arise

I) Lack of clarity

II) Requirement confusion

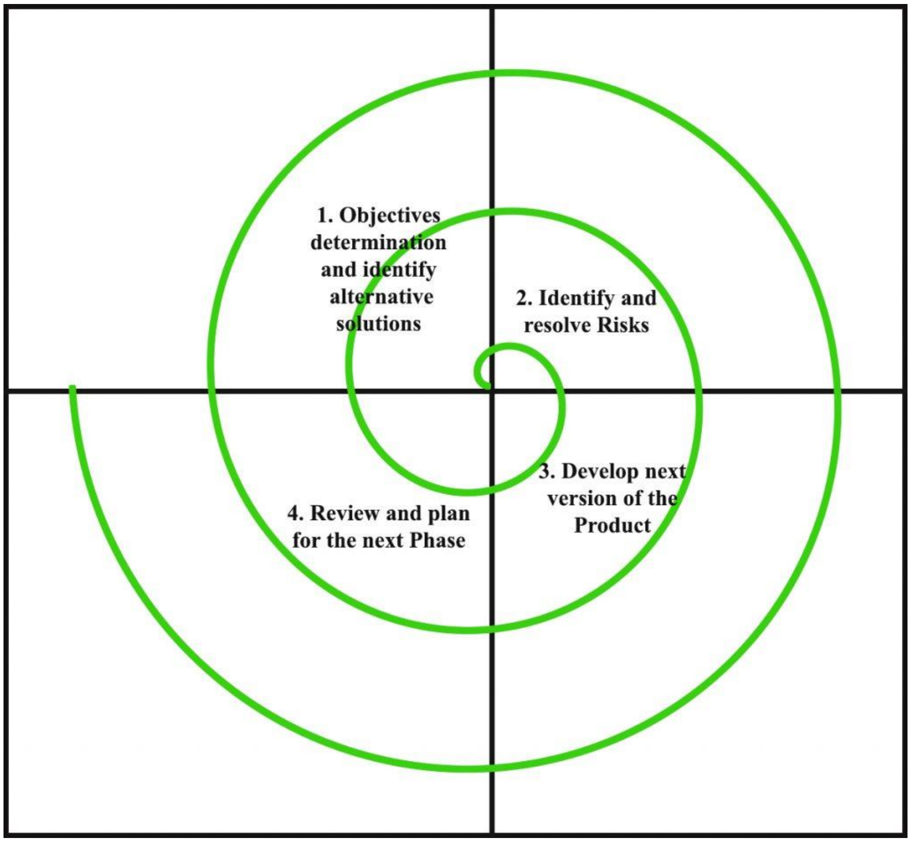
IV) Requirement Amalgamation

2) **Analysis**: The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.
3) **Design**: Design phase is a stage where software developers define the technical details of the product.
4) **Implementation**: In this phase, the tasks are divided into modules or units and assigned to various developers.
5) **Testing**: A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version.
6) **Maintenance**: Corrective Maintenance, Adaptive Maintenance, Perfective Maintenance.

(16) Write Phases of spiral model?

*Spiral model phases like…

- **Planning:** The first phase of the Spiral Model is the planning phase, where the scope of the project is determined and a plan is created for the next iteration of the spiral.
- **Risk Analysis:** In the risk analysis phase, the risks associated with the project are identified and evaluated.
- **Engineering:** In the engineering phase, the software is developed based on the requirements gathered in the previous iteration.
- **Customer Evaluation:** Customer satisfaction - The spiral model facilitates customer feedback. If the software is being designed for a customer, then the customer will be able to see and evaluate their product in every phase.


1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

(17) Write agile manifesto principles?

- The Twelve Principle of Agile Manifesto:

I. **Customer Satisfaction**

II. **Welcome Change**

III. **Deliver the Working Software**

IV. **Collaboration**

V. **Motivation**

VI. **Face-to-face Conversation**

VII. **Measure the Progress as per the Working Software**

VIII. **Maintain Constant Pace**

IX. **Monitoring**

X. **Simplicity**

XI. **Self-organized Teams**

XII. **Review the Work Regularly**

(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.

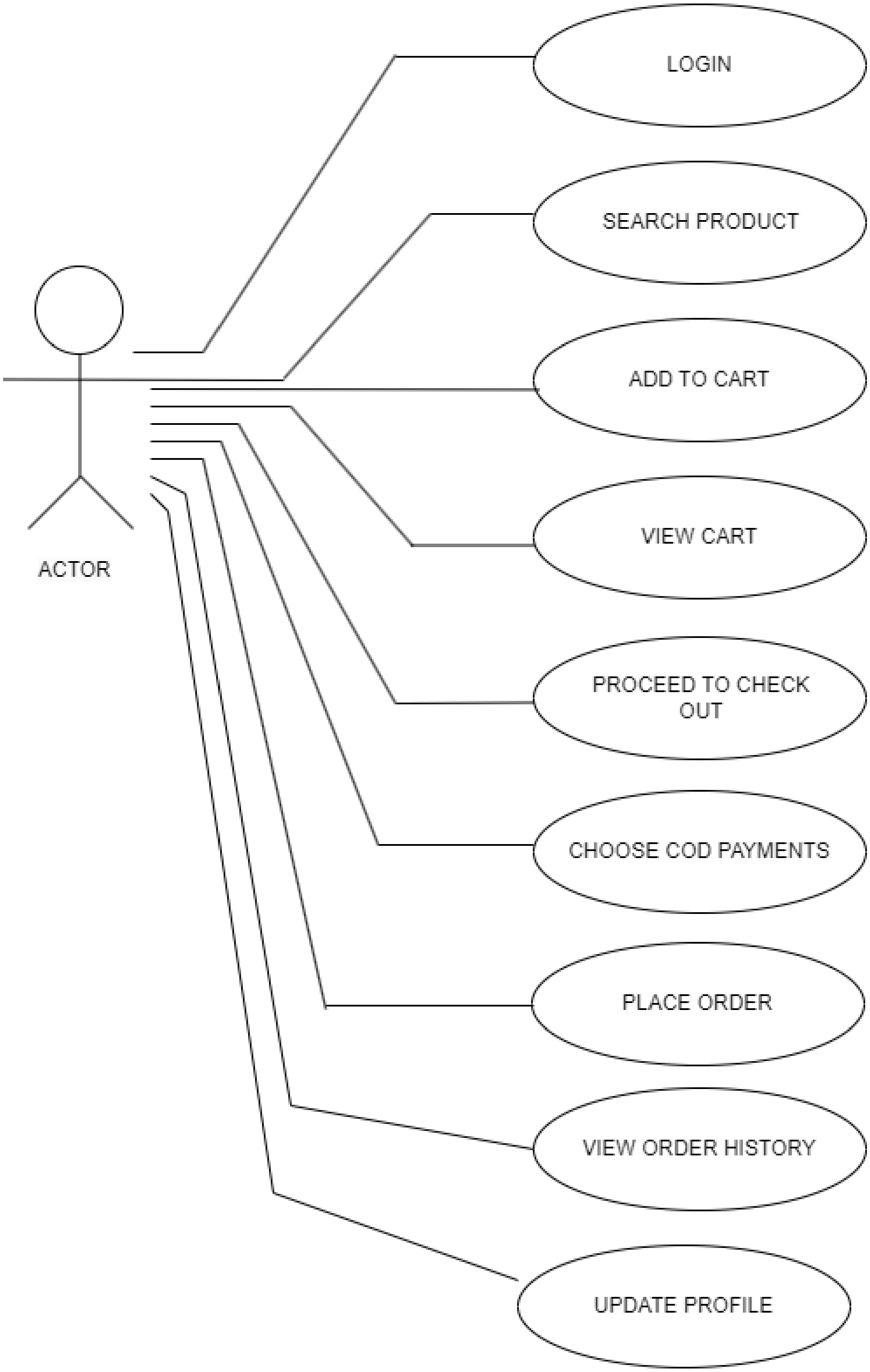
- **Pros of Agile Methodology**

1. Faster Delivery
2. Collaborative
3. Adaptability
4. Detecting Problems Faster
5. Transparency
6. Moving Constantly

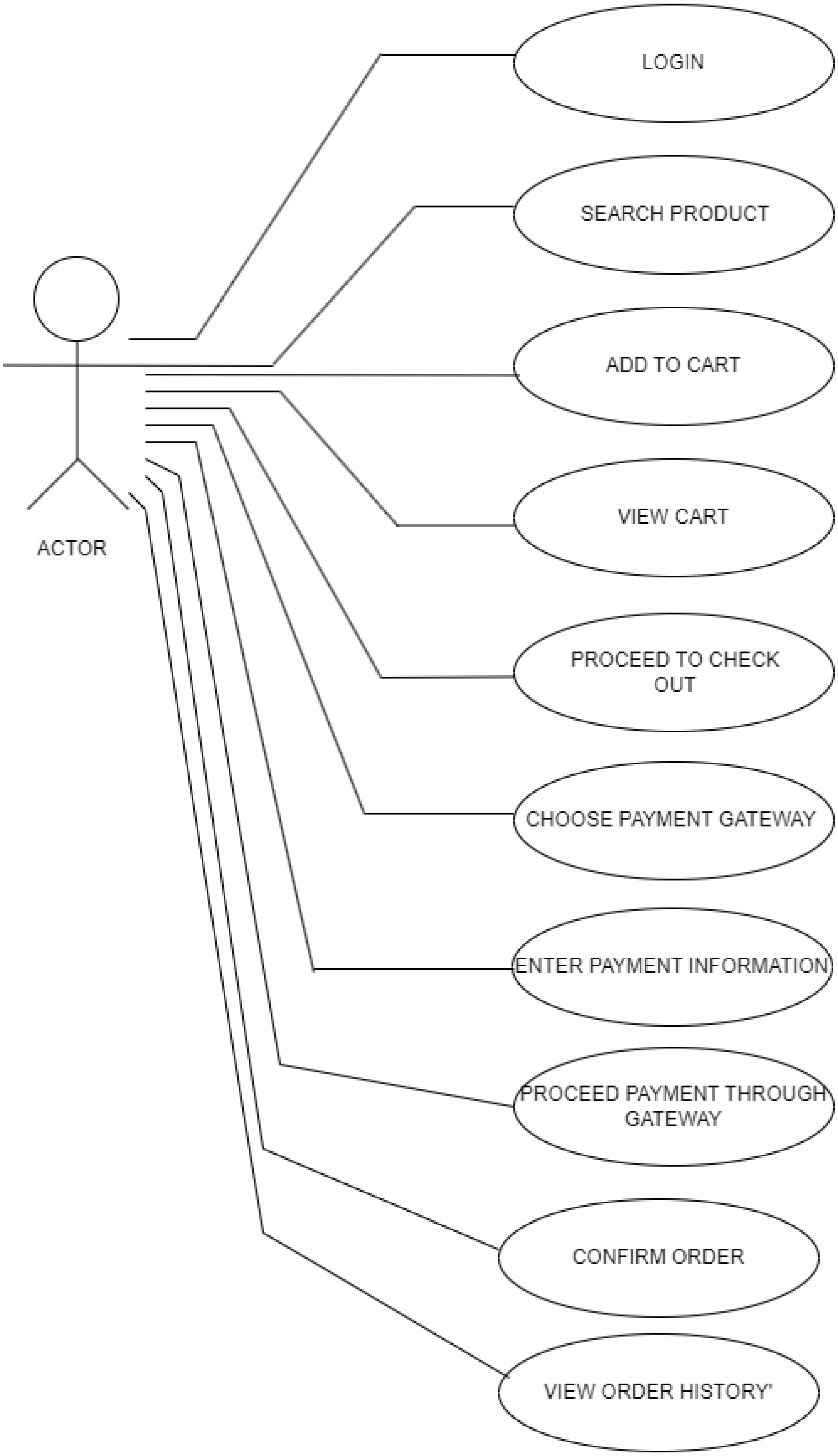
- **Cons of Agile Methodology**

1. Short Term Planning
2. Tricky paradigm Shift
3. Much AD-Hocs
4. No Clarity On Timeframe & Budget
5. Lack OF Proper Documentation

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



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



(21)What is 7 key principels? 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

1) Testing shows presence of Defects:

- Testing can show that defects are present, but cannot prove that there

  are no defects.

- 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.
- As we find more defects, the probability of undiscovered defects remaining in a system reduces.

2) 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.
- Similarly, most operational failures of a system are usually confined to a small number of modules.

3) 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.

4) Exhaustive Testing is Impossible!

- Testing everything including **all combinations of inputs and preconditions is not possible**.
- That is, we must Priorities 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.

5) Early Testing

- Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.
- Testing activities should start as early as possible in the development life cycle.

6) The Pesticide Paradox

- The test cases need to be regularly **reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.**
- 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.

7) Testing is Context Dependent

- Testing is done differently in different contexts.
- Testing is basically context dependent.
- Different kinds of sites are tested differently.

(22) Difference between Verification and Validation?

Verification	**Validation**
I. Verification is a process which is performed at development.	I. Validation is a process which is performed at testing level.
II. Verification can be achieved by asking “are you building a product right?”	II. Validation can be achieved by asking “are you building a right product?”
III. Done in control quality process.	III. Done in validate scope process.
IV. Internal process; Done by Quality control dept.	IV. External process; Done by the customer.