[MODULE-1- INTRODUCTION &FUNDAMENTAL]

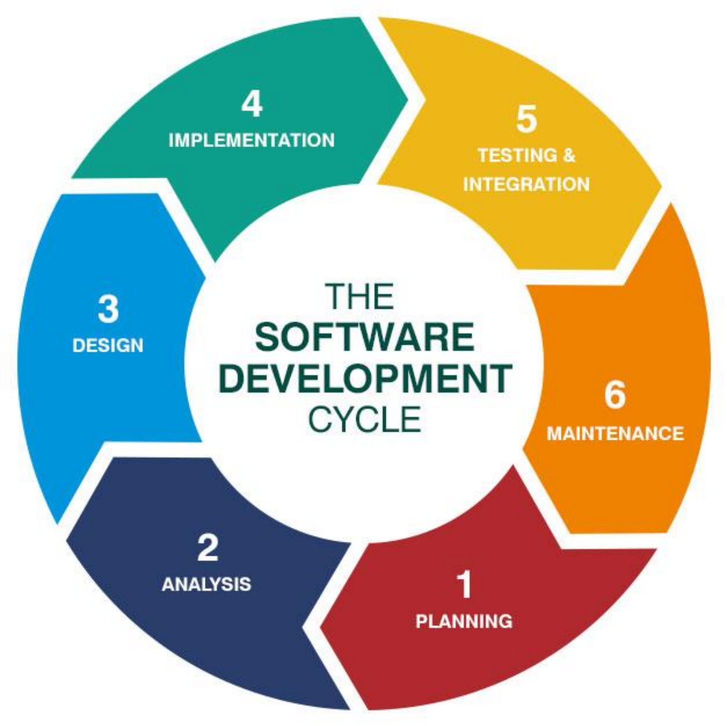
[ASSIGNMENT-1]

NAME: - PANCHAL KUSH MAHENDRABHAI

SUBJECT: - SOFTWARE TESTING

INSTITUTE NAME: - TOPS TECHNOLOGIES

1.WHAT IS SDLC (SOFTWARE DEVELOPMENT LIFE CYCLE)?



- SDLC is a step by step approach to develop any software/product with high quality, within the time and within the cost.
- The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time.

2. WHAT IS SOFTWARE TESTING?

- Software Testing is a Process to verify that the requirement are fulfilled or not.
- Software testing is a process which is used to identify the correctness, completeness, quality of the developed software.

3.WHAT IS SRS (SOFTWARE REQUIREMENT SPECIFICATION)?

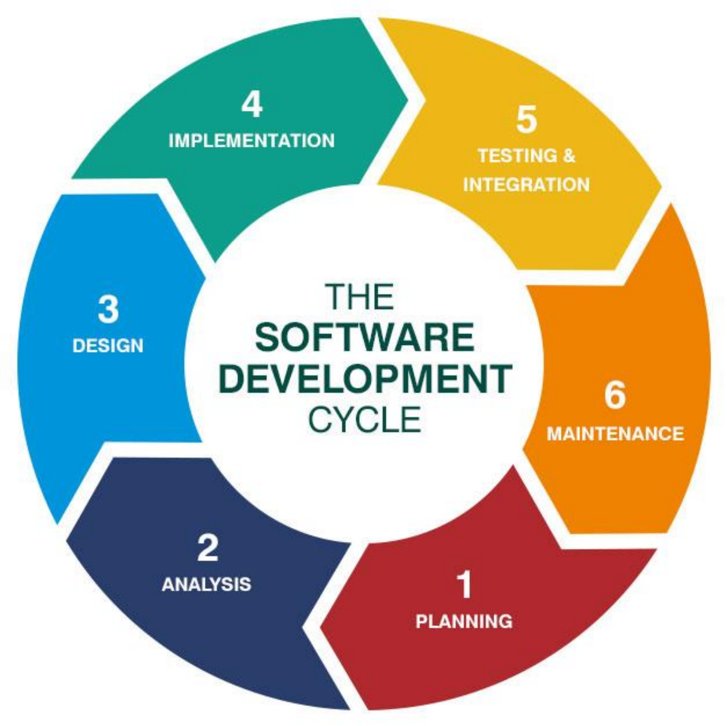
- A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform.
- Complete behaviour of the system.
- SRS is the official statement of what the system developers should implement.
- REQUIREMENTS: -

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

4.WHAT IS TESTING OBJECTIVE?

- Finding defects & Preventing defects.
- Gaining confidence in and providing information about the level of quality.
- Both dynamic testing and static testing can be used for testing objectives.
- To prevent defect to be entered into Project Life Cycle.
- Reviews of documents throughout the lifecycle.
- They provide information in order to improve: -
    1. The system to be tested.
    2. The development and testing processes.
    3. Live operation.

5.WRITE SDLC PHASES WITH BASIC INTRODUCTION?



1) PLANNING / REQUIREMENT GATHERING: -

- Problems can be arisen while gathering the requirement.
- Lack of Clarity.
- Requirement confusion (Functional / Non-Functional).
- Requirement Amalgamation (Group).

2) ANALYSIS: -

  - How the requirement can be executed.

- E.g. SRS (Software Requirement Specification)
- Complete behaviour of the system.

3) DESIGNING: -

- Visualize the software / system by designing.
- E.g.: - 1. DFD: Data Flow Diagram
  2. ER: Entity Relationship

4) IMPLEMENTATION/ BUILDING/ CODING: -

- Software can be implemented by the technology (Java, Python, PHP).
- Fulfil hardware & software resources.

5) TESTING (QA): -

- Verify & Validate the software.

6) MAINTENANCE (DEPLOYMENT): -

- Sales after service.

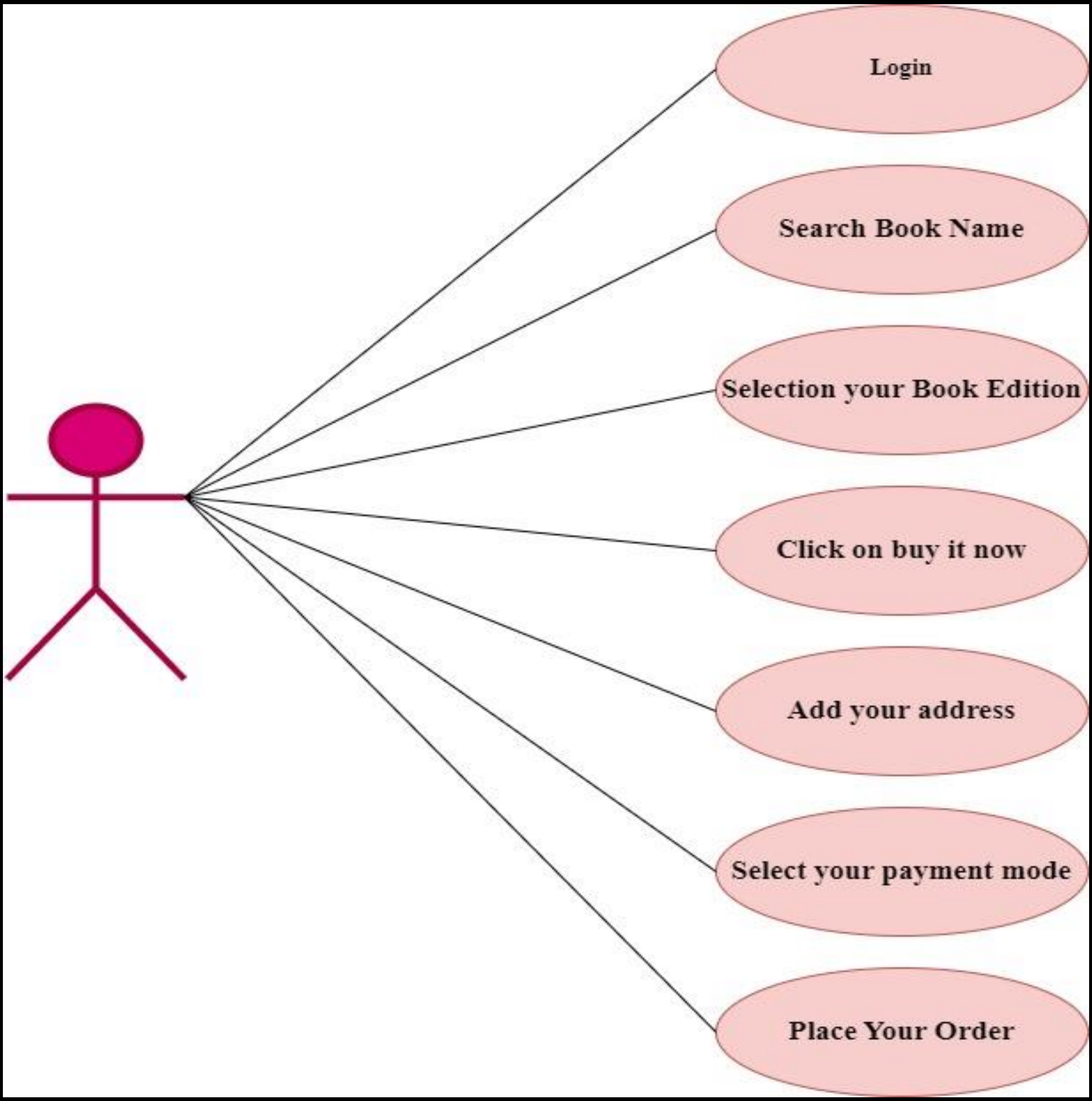
→ 3 TYPES OF MAINTENANCE: -

- CORRECTIVE MAINTENANCE: - Identifying and repairing defects.
- ADAPTIVE MAINTENANCE: - Adapting the existing solution for the new platforms.
- PERFECTIVE MAINTENANCE: - Implementing the new requirements.

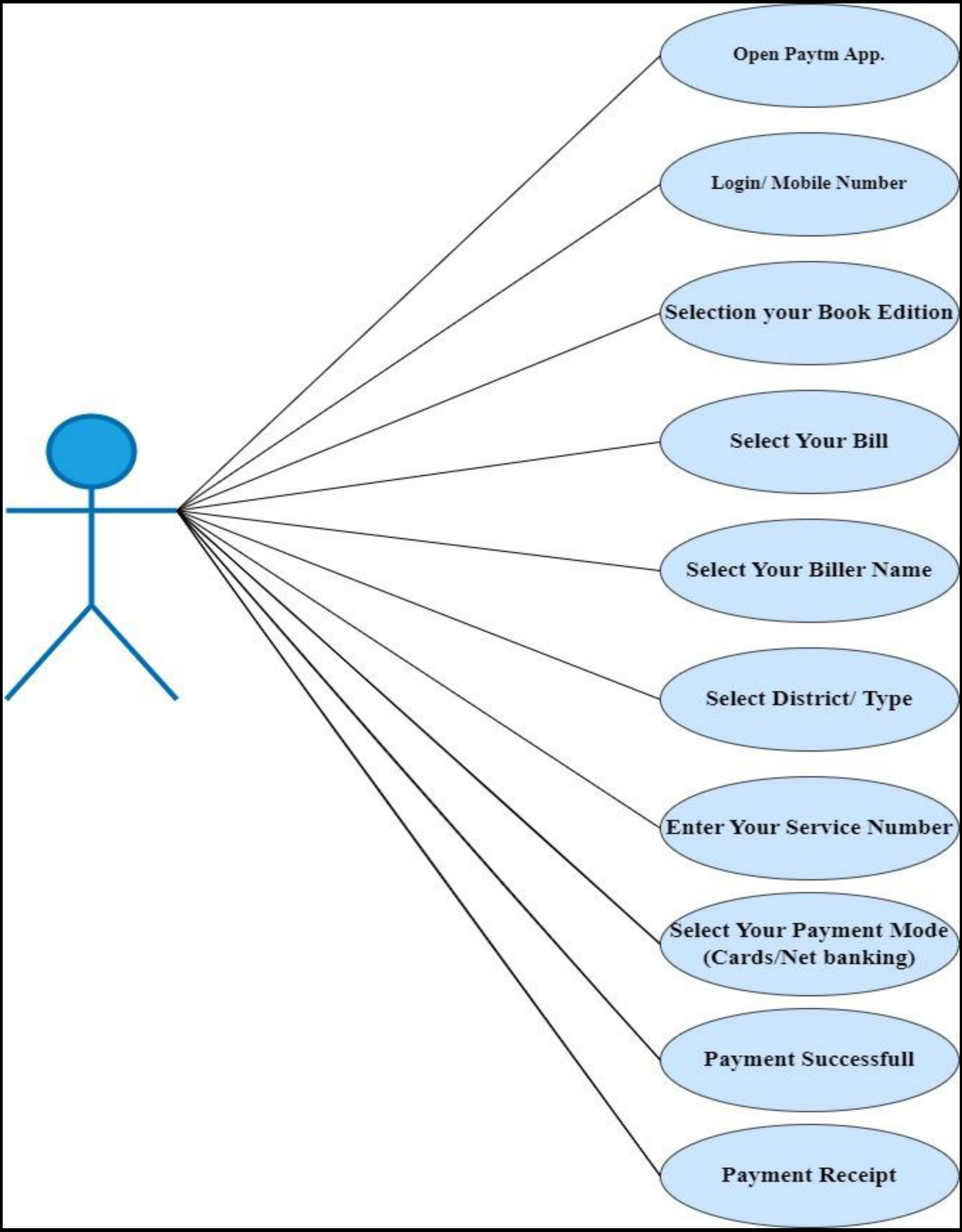
6.WHEN TO STOP SOFTWARE TESTING?

- All the high priority bugs are fixed.
- The rate at which bugs are found is too small.
- The testing budget is exhausted.
- The project duration is completed.
- The risk in the project is under acceptable limit.
- Measuring Test Coverage.
- Number of test cycles.
- Number of high priority bugs.

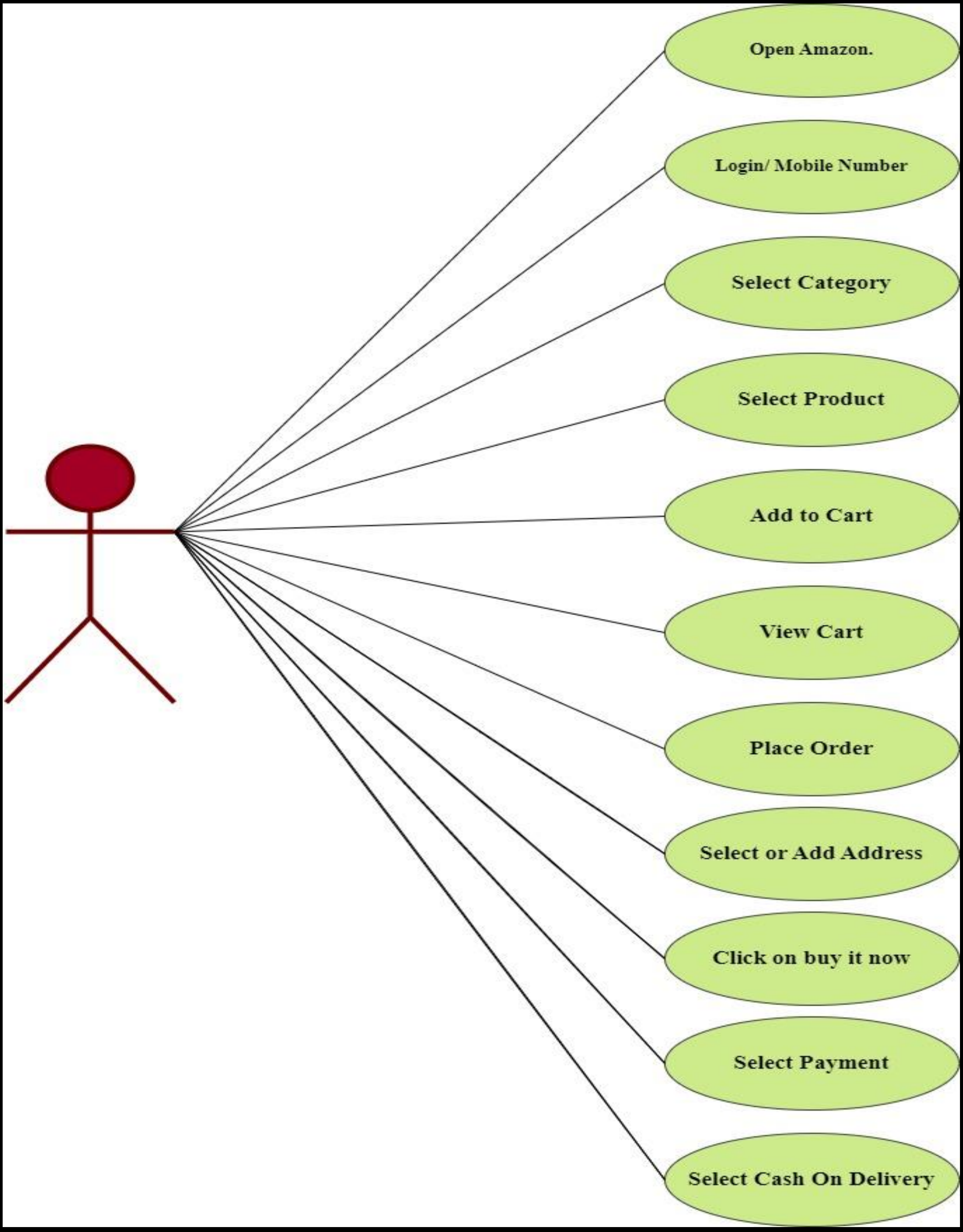
7. DRAW USECASE ON ONLINE BOOK SHOPPING.



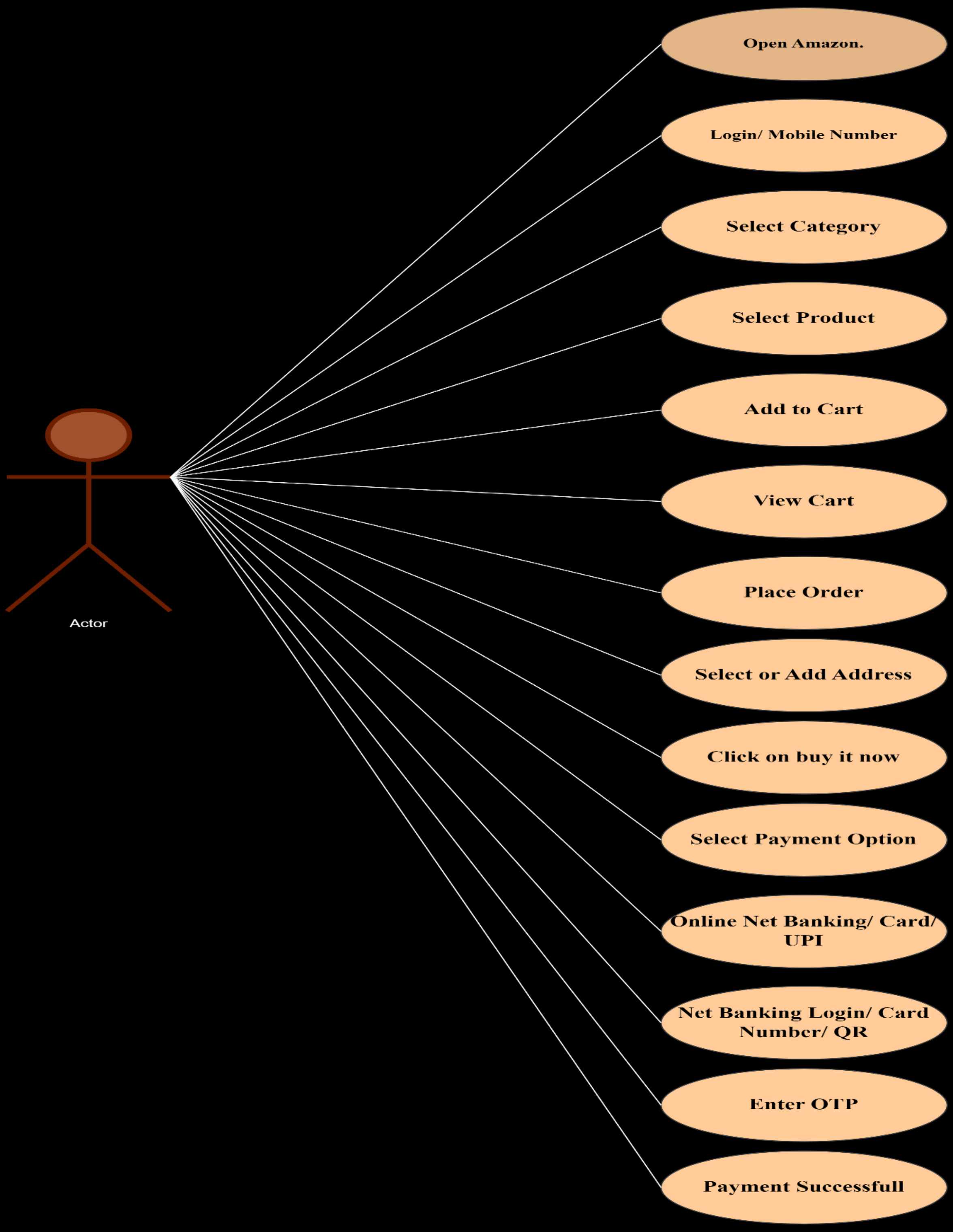
8. DRAW USECASE ON ONLINE BILL PAYMENT SYSTEM(PAYTM).



9. DRAW USECASE ON ONLINE SHOPPING PRODUCT USING COD.



10. DRAW USECASE ON ONLINE SHOPPING PRODUCT USING PAYMENT GATEWAY.



11. WHAT IS OOPS?

- Object oriented programming is viewed as a collection of objects.
- It is used to structure the software programme into single reusable code.
- Here it is referred as Functional testing or Black Box Testing.

12.WRITE BASICS CONCEPT OOPS?

1. CLASS
2. OBJECT
3. ENCAPSULATION
4. INHERITANCE
5. POLYMORPHISM
6. ABSTRACTION

13.WHAT IS CLASS?

- Blueprint / Template which is a collection of data members and member functions.
- Class is a collection of a data member (variables) and member function with its behaviour.

14.WHAT IS OBJECTS?

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

15.WHAT IS ENCAPSULATION?

- A wrapping up of data and function into a single until called Encapsulation.
- It hides private access of data member &member function.
- Data hiding.

16.WHAT IS POLYMORPHISM?

- An ability to take one name having many different forms.

1. Compile time Polymorphism: (Operator Overloading)

- Method name should be same in single class, but its behaviour (Arguments & Data type) is different.

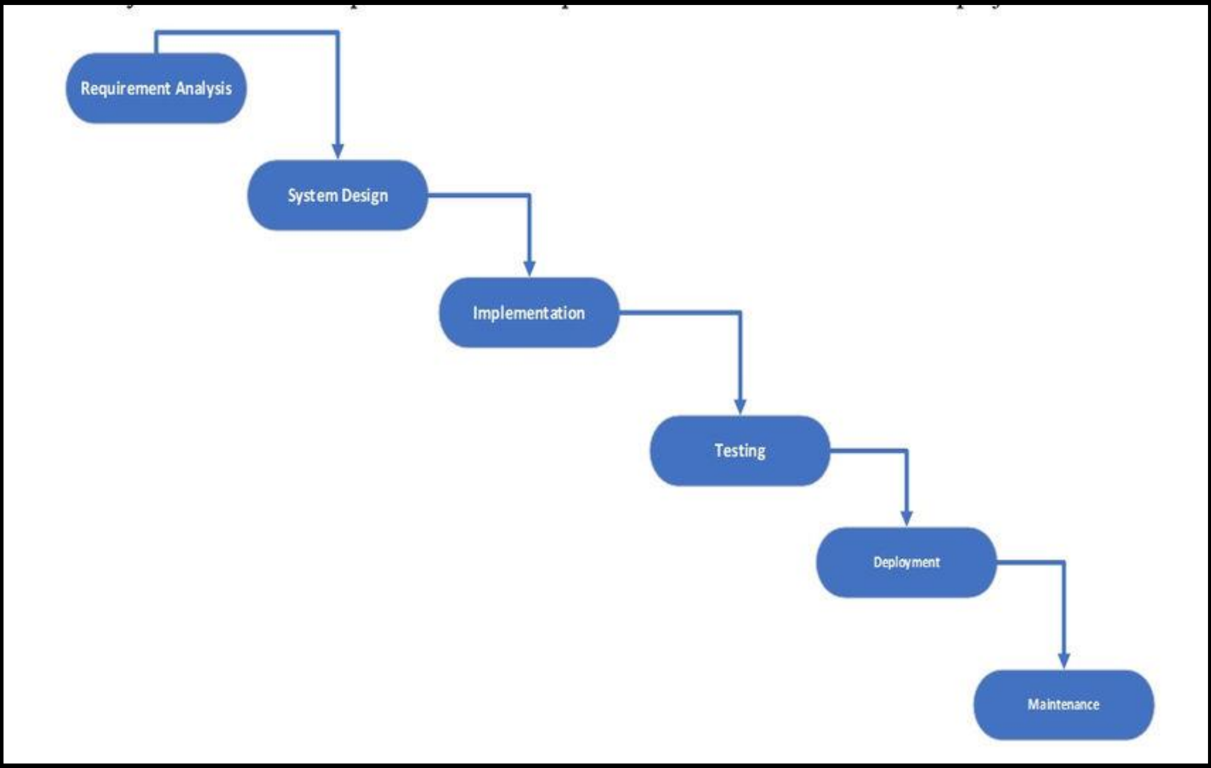
2. Run time Polymorphism: (Operator Overriding)

- Method should be same in super class and sub class, but its behaviour is different.

17.WHAT IS INHERITANCE?

- One class (Super, Base) inherits the properties of another class (Sub, Derived).
- Types of Inheritance: -
    1. Single Inheritance
    2. Multilevel Inheritance
    3. Hierarchical Inheritance
    4. Hybrid Inheritance
    5. Multiple Inheritance

18.EXPLAIN PHASES OF THE WATERFALL MODEL?



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

→Application (When to use?): -

- Requirements are very well documents, clear & fixed.
- Product definition is stable.
- Technology is understood & not dynamic.
- The project is short.

→Pros: -

1. Simple and easy to understand and use.
2. Phases are processed and completed one at time.
3. Works well for smaller projects where requirements are very well understood.
4. Process and results are very well documented.
5. Clearly defined stages.
6. Easy to arrange task.

→Cons: -

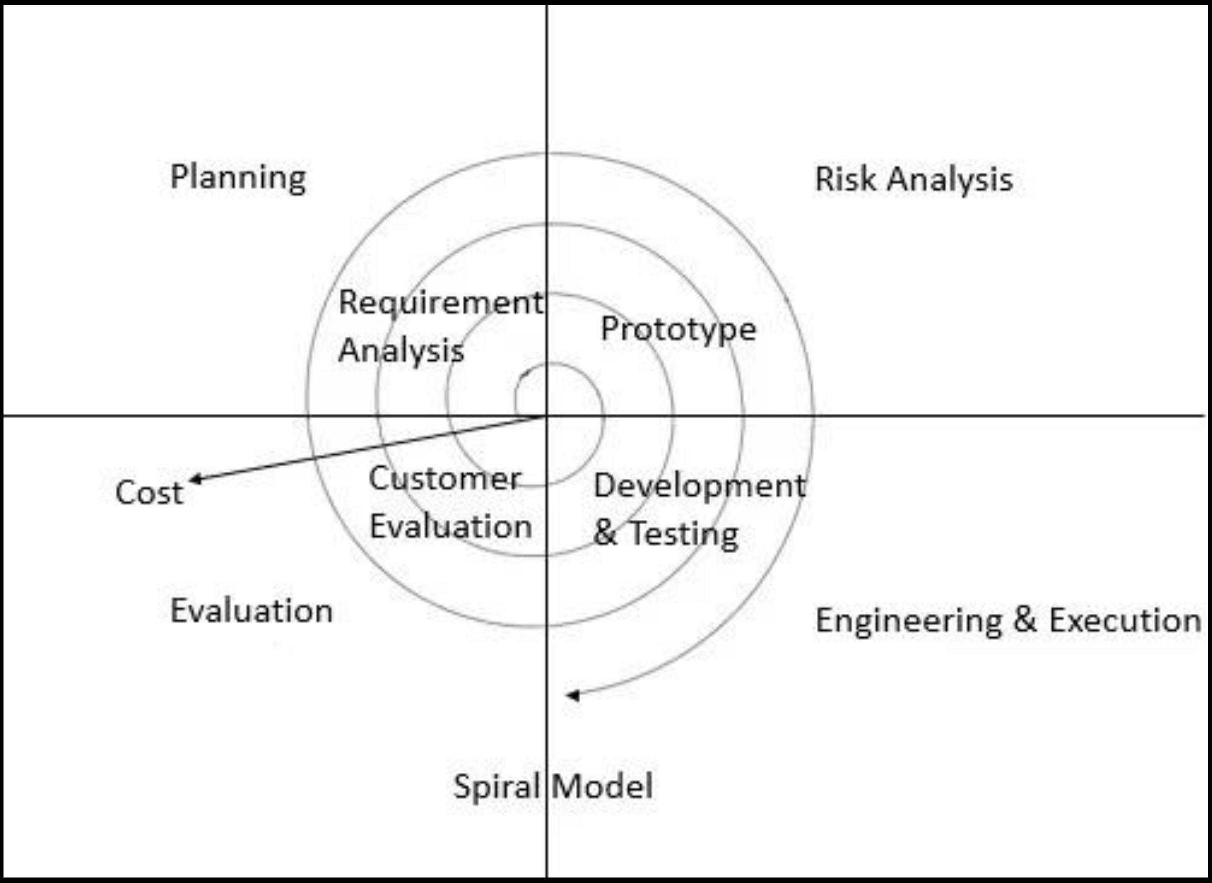
1. Lack of flexibility & adaptability (No changing requirements & no parallel work).
2. High amounts of risk and uncertainly.
3. Poor model for long and on-going projects.
4. Not good model for complex & object oriented projects.

19.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.

- The Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement.

20.WRITE PHASES OF SPIRAL MODEL?



1. Planning
2. Risk Analysis
3. Engineering
4. Customer Evaluation

1. Planning: -

   Determination of objectives, alternatives and Constraints.

2. Risk Analysis: -

   Analysis of alternatives and identification / resolution of risks.

3. Engineering: -

   Development of the next level product.

4. Customer Evaluation: -

   Assessment of the result of Engineering.

21.WRITE AGILE MANIFESTO PRINCIPLES?

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

22.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.
- Pros: -
  1. Very realistic approach.
  2. Rapid delivery.
  3. Functionality can be developed rapidly.
  4. Resource requirements are premium.
  5. Little or no planning required.
  6. Promotes teamwork and cross training.
  7. Suitable for fixed or changing requirements.
  8. Gives flexibility to developers.
- Cons: -
  1. More risk of sustainability, maintainability and extensibility.
  2. Depends heavily on customer interactions.
  3. Very high individual dependency.
  4. Minimum documentation generated.
  5. Not useful for small projects.
  6. Not suitable for handling complex dependencies.