ASSIGNMENT — 1

MODULE – 1 (INTRODUCTION AND FUNDAMENTAL)

1) What is software testing?

→ Software Testing is a process to verify that user Requirements are fulfil or not.

→ Software testing is a process to identify the correctness,completeness, and quality of the developed software

→ There are two types of testing

1. Manual testing
2. Automation testing

❖ Manual testing :- to execute the test case manually by the test engineer that is called manual testing.

❖ Automation testing :- to execute the test case by the test engineer with using automation tool that is called automation testing.

100% automation is not possible.

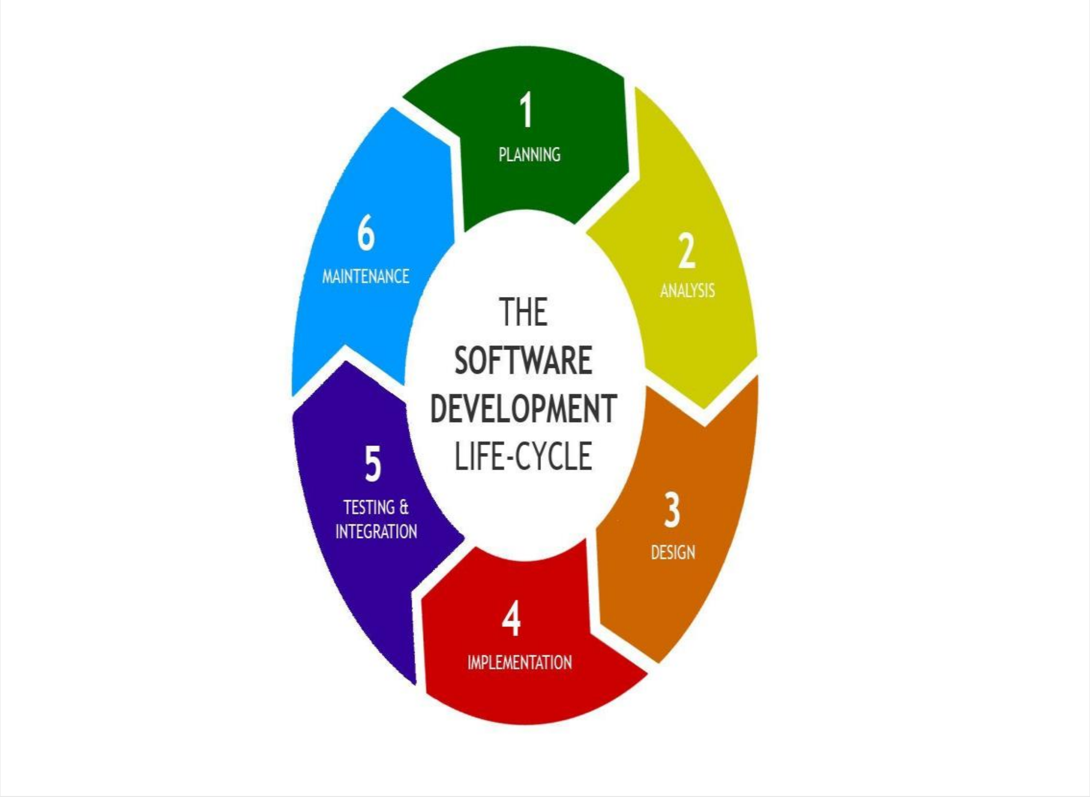
2) WHAT IS SDLC?

→ SDLC is stand for software development life cycle.

→ A project is seccessfull if it is done on schedule, within budget,

And meets the good quality.

- Here are the diagram of SDLC(Software development life cycle)



→ There are six phases of SDLC(Software Development Life cycle)

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

3) What is SRS?

→ A software requirements specification (SRS) is a complete description of the behavior 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.

→ Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints)

4) Write SDLC phses with basic introduction?

- There are six phases of SDLC.
  1. Planning
  2. Analysis
  3. Design
  4. Implementation
  5. Testing
  6. Maintenance

1.planning :-

- Lack of clarity
- Requirement confusion (Functional/ Non-Functional)
- Requirement amalgamation

2.Analysis :-

- How the requirements can be executed
- Test analysis involves studying software requirements and specifications to identify testable functions and create test conditions

3.Designing :-

- Visualize the software/ system by designing
  Like DFD – data flow diagram

Use case diagram
ER – entity relationship diagram etc.

4.Implementation :-

- ❖ Software implemented by the technology like java, python, PHP etc
- ❖ Full fill the requiremets of user

5.Testing :-

- ❖ Verify that the user requirements fullfill or not
- ❖ All resources are working or not

6.Maintenance :-

- ❖ There are three types of Maintenance
  - o Corrective maintenance – identifying and repair the defects
  - o Adaptive maintenance – adapting the existing solution to the new platforms.
  - o Perfective maintenance – implementing the new requirements.

5) WHAT IS OOPS ?

→ oops is stand for object oriented programming language

→ Set of instructions that can be executed by the developer.

→ Identifying objects and assigning responsibilities to these objects.

→ objects communicate other objects by sending messages. Messages are received by the methods of and object.

→ objects of a program interact by sending messages to each other.

→ Basic concepts of oops :-

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

6) what is class?

→ a class is a template/blueprint that includes a set of data and functions for creating objects

→ a class is way to define an object by outlining its features and actions.

→ example:-

- ❖ Class – car
- ❖ Data member – model,year,company name etc
- ❖ Member function – display_info,start_engine,stop_engine

7) what is object ?

→ An object is an instance of a class, which represents a real-world entity or concept.

→ It has its own set of attributes(data) & methods(function) that describe & define its behavior.

→ In object-oriented programming, the basic unit is the object, which is an instance of a class; the object allocates memory based on the class definition and always represents that class's structure and behavior.

- ❖ Example :- fruit is class
    - Mango is object

8) 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 Methods break the product into small incremental builds.

→ Agile model believes that every project needs to be handled differently and the existing methods divided into small time frames to deliver specific feature of a release.

9) write basic concepts of oops?

1.class

2.object

3.encapsulation

4.inheritance

5.polymorphism

Overloading

Overriding

6.abstraction

10) what is encapsulation?

→ wrapping up data members and member functions into a single unit is known encapsulation

→ encapsulation enables data hiding

11) what is inheritance?

→ Inheritance allows one class to inherit properties and features from another class, promoting code reusability.

- ❖ Types of Inheritance:
  1. Single Inh.
  2. Multilevel Inh.
  3. Multiple Inh.
  4. Heirarchical Inh.
  5. Hybrid Inh.

→ main class :- parent class/base class/super class

→ another class from main :- child class/derived class/sub class

12) what is polymorphism?

→ polymorphism lets the same method or function work in different ways for different objects.

→ It allows objects of different types to be treated through a common interface.

→ polymorphism has two types :-

1) overloading

2) overriding

13) explain phases of waterfall model?

1. **Requirements**: Understand and write down exactly what the project needs to do. This is where you figure out what the final product should be like.
2. **Design**: Plan how to create the project. This involves deciding how everything will be built and how the parts will work together.
3. **Implementation**: Build the project according to the design. This is where the actual coding or construction happens.
4. **Testing**: Check the project to make sure it works as expected. This is where you find and fix any problems.
5. **Deployment**: Release the project for use by people. This is when it goes live and users can start using it.
6. **Maintenance**: Keep the project running smoothly after it's released. This includes fixing any new issues that come up and making updates if needed.

14) Explain working methodology of agile model and, also write pros and cons?

→ The Agile SDLC model combines iterative and incremental approaches to ensure flexibility and quick delivery.

→ It divides the product into small, manageable parts called iterations, each lasting 1-3 weeks. During each iteration, cross-functional teams handle planning, design, coding, and testing simultaneously.

→ At the end of every iteration, a working version of the product is presented to customers and stakeholders for feedback.

❖ Pros :-

- Is a very realistic approach to software development
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.

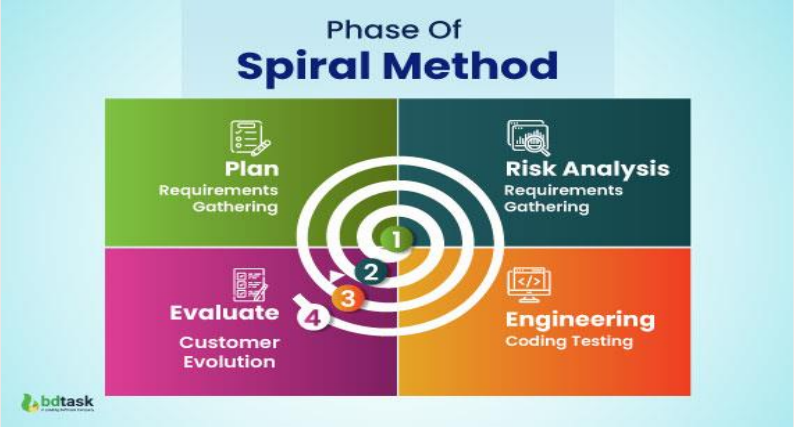
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required
- Easy to manage
- Gives flexibility to developers

❖ 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 documentation

15) write phases of spiral model?

1. Planning: Define project goals, scope, and deliverables.
2. Risk Analysis: Identify potential risks and how to address them.
3. Testing: Check the product to find and fix issues.
4. Evaluation: Review progress and gather feedback to plan the next cycle.



16) write agile manifesto principles

→ **Customer Satisfaction**: Keep the customer happy by delivering useful software early and often.

→ **Welcome Change**: Embrace changes in requirements, even late in the project, to keep the product relevant.

→ **Frequent Delivery**: Deliver working software regularly, with shorter times between releases.

→ **Team Collaboration**: Work closely with business people and developers throughout the project.

→ **Motivated Teams**: Build projects around motivated individuals and give them the tools and environment they need to do their best work.

→ **Face-to-Face Communication**: The best way to convey information is through face-to-face conversations.

→ **Working Software**: The main measure of progress is delivering working software.

→ **Sustainable Pace**: Work at a steady pace that can be maintained indefinitely, avoiding burnout.

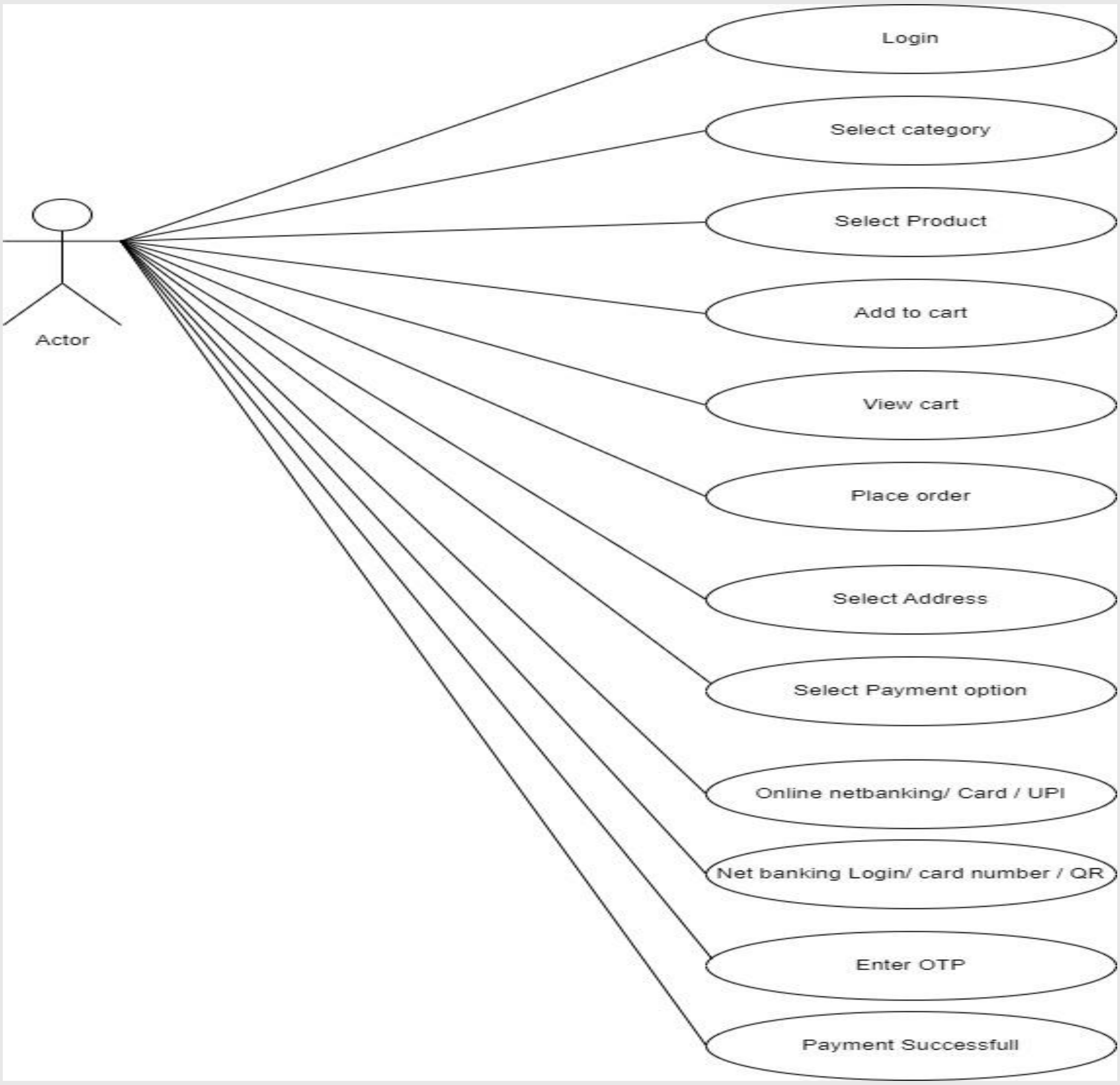
→ **Technical Excellence**: Focus on good design and technical excellence to enhance agility.

→ **Simplicity**: Keep things simple and focus only on what is needed right now.

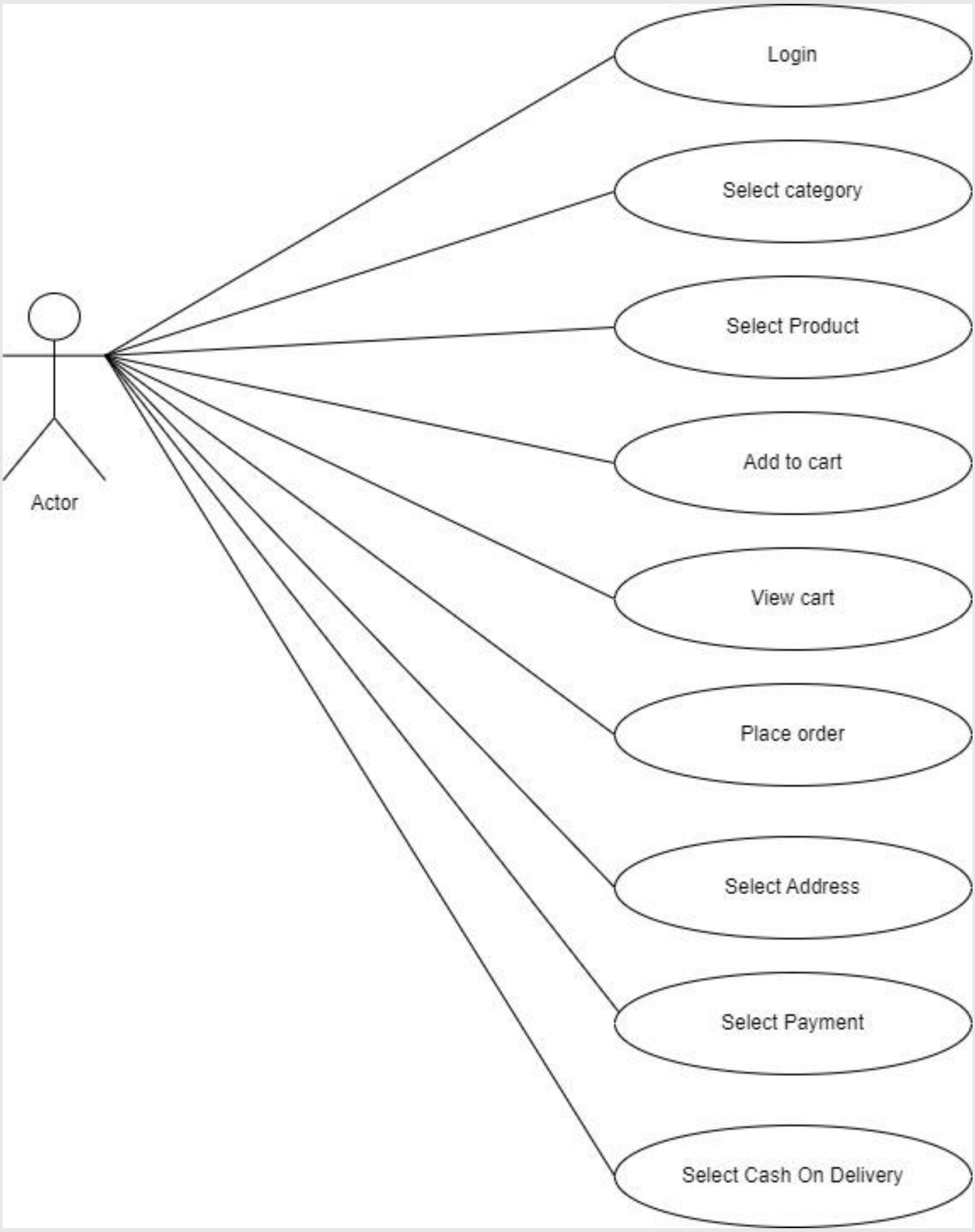
→ **Self-Organizing Teams**: Encourage teams to organize their own work for better results.

→ **Regular Reflection**: Reflect on how to become more effective and adjust behaviors accordingly.

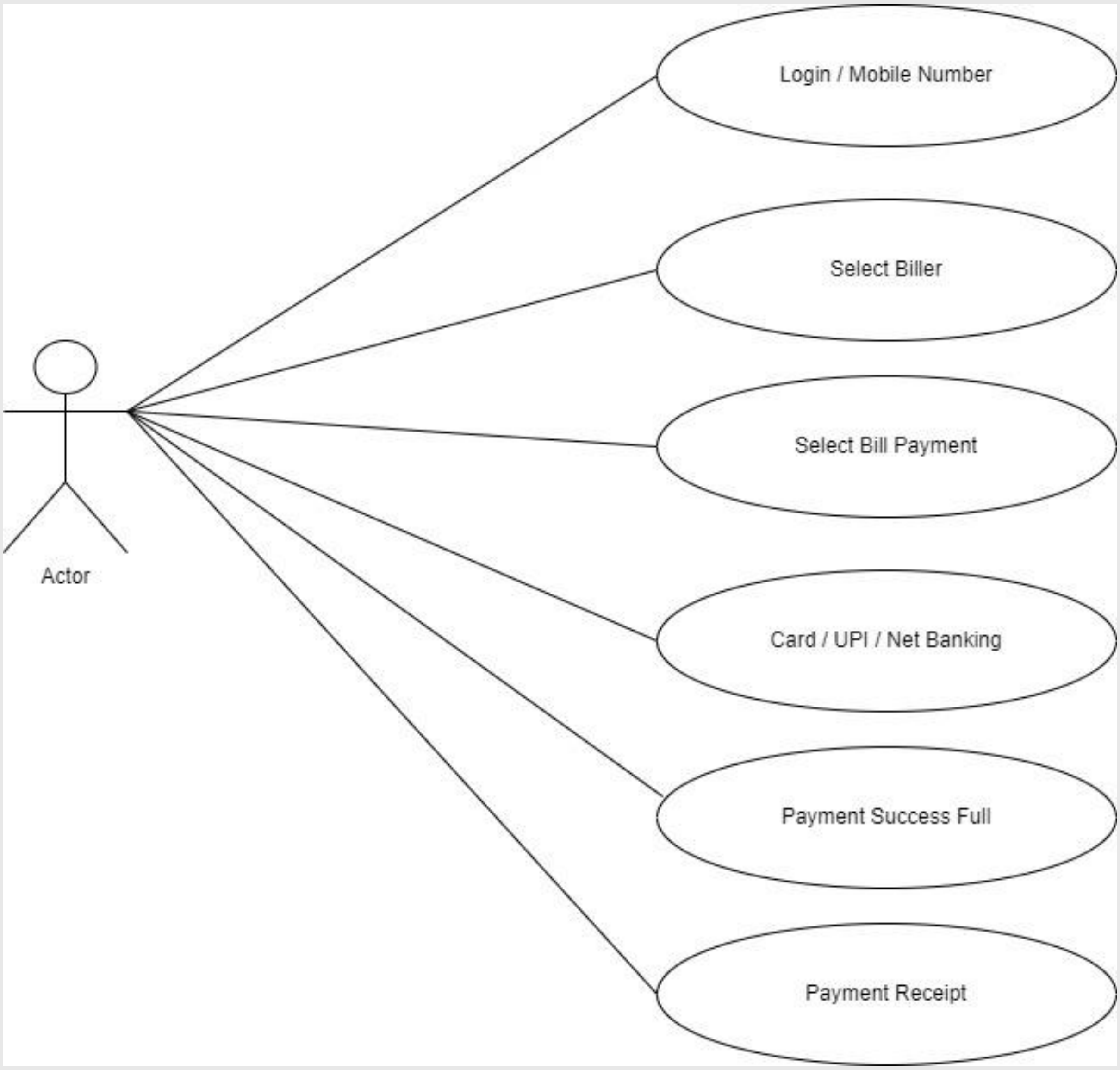
17) Draw use case on online shopping product using payment gateway.



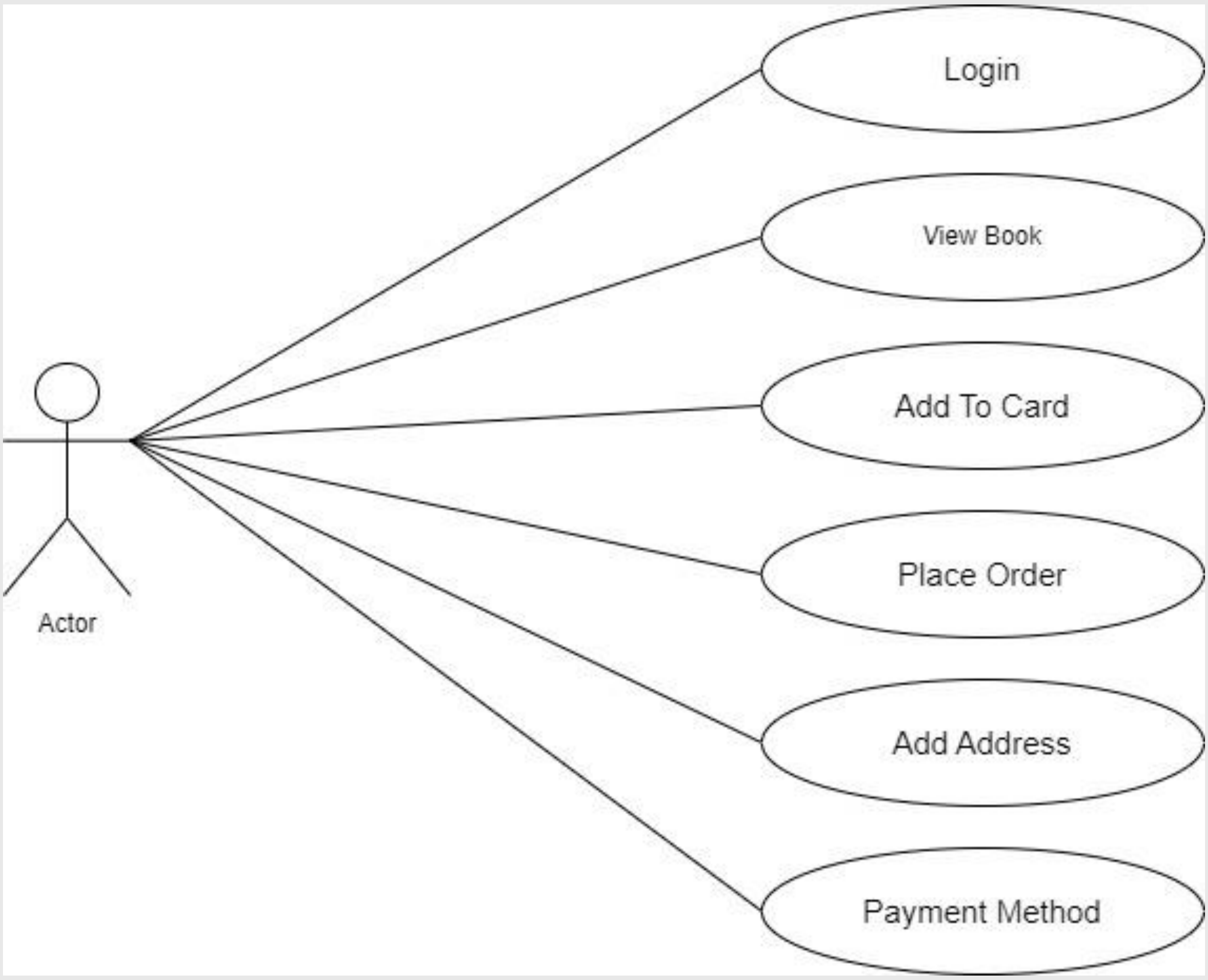
18) Draw use case on online shopping product using COD.



19) Draw use case on online bill payment system (paytm).



20) Draw use case on online book shopping.


Login
View Book
Add To Card
Place Order
Add Address
Payment Method
Actor