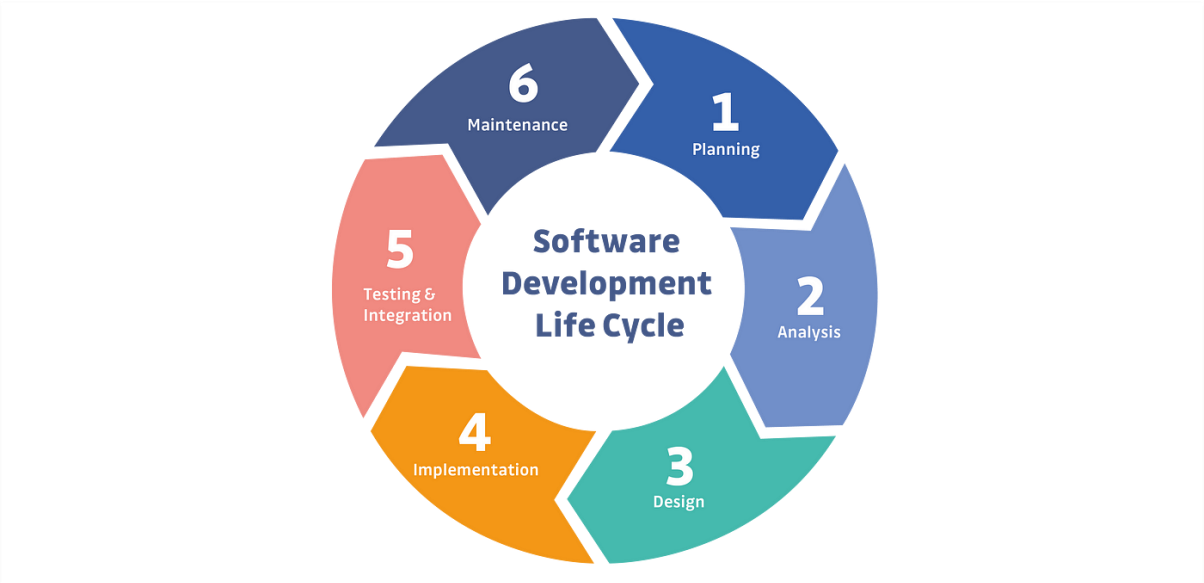
NAME: KSHATRIYA RONAK OMPRAKASH

COURSE: SOFTWARE TESTING (MANUAL AND AUTOMATION)

ASSIGNMENT: INTRODUCTION IN SOFTWARE TESTING

1. **What is SDLC?**

SDLC is a step by step approach to develop any software/ product with high quality, within the time and within the cost by defining the phases like Planning, Analysis, Design, Implementation, Testing & Integration, Maintenance.



2. **What is software testing?**

Software testing is a process to verify that the requirements are fulfilled or not.Software testing is a process which is used to identify the correctness, completeness, quality of the developed software. Two types of software testing are there

1. Manual Testing
2. Automation Testing

3. **What is SRS?**

Softwere Requirements Specification (SRS) is a document that describes what a softwere product will do and how it will perform.

It includes a set of use cases that describe all of the interection that the users will have with the softwere.

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 objective.

.To prevent defect to be entered into Project Life Cycle.

.Reviews of documents throughout the lifecycle.

5. **Write SDLC phases with basic introduction?**

1) Planning / Requirement Gathering (What)

Problems can be Raised while gathering the requirements:

--> Lack of clarity

--> Requirement confusion (functional/ non-functional)

--> Requirement Amalgamation (group)

2) Analysis (How)

How the requirements can be executed.

SRS (Software Requirement Specification) : Document

3) Designing:

-Visualize the software/ system by designing.

E.g.: DFD : Data Flow Diagram

ER: Entity Relationship

4) Implementation /Building/ coding

Software can be implemented by the technology (Java, Python, PHP).

5) Testing (QA)

Verify & validate the software.

6) Maintenance (Deployment):

Sales after service:

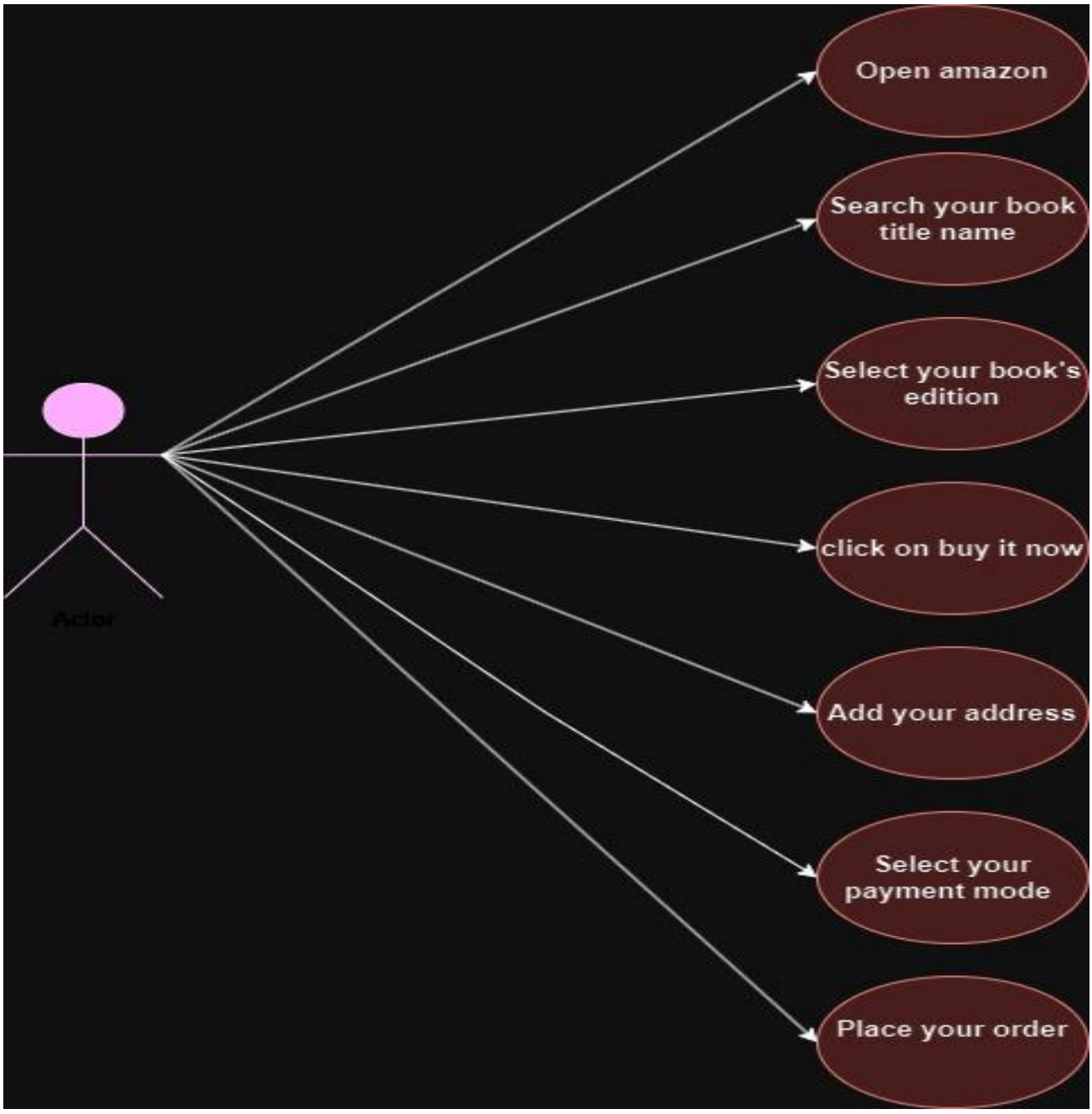
3 types of Maintenance :

-Corrective maintenance: Identifying and repairing defects

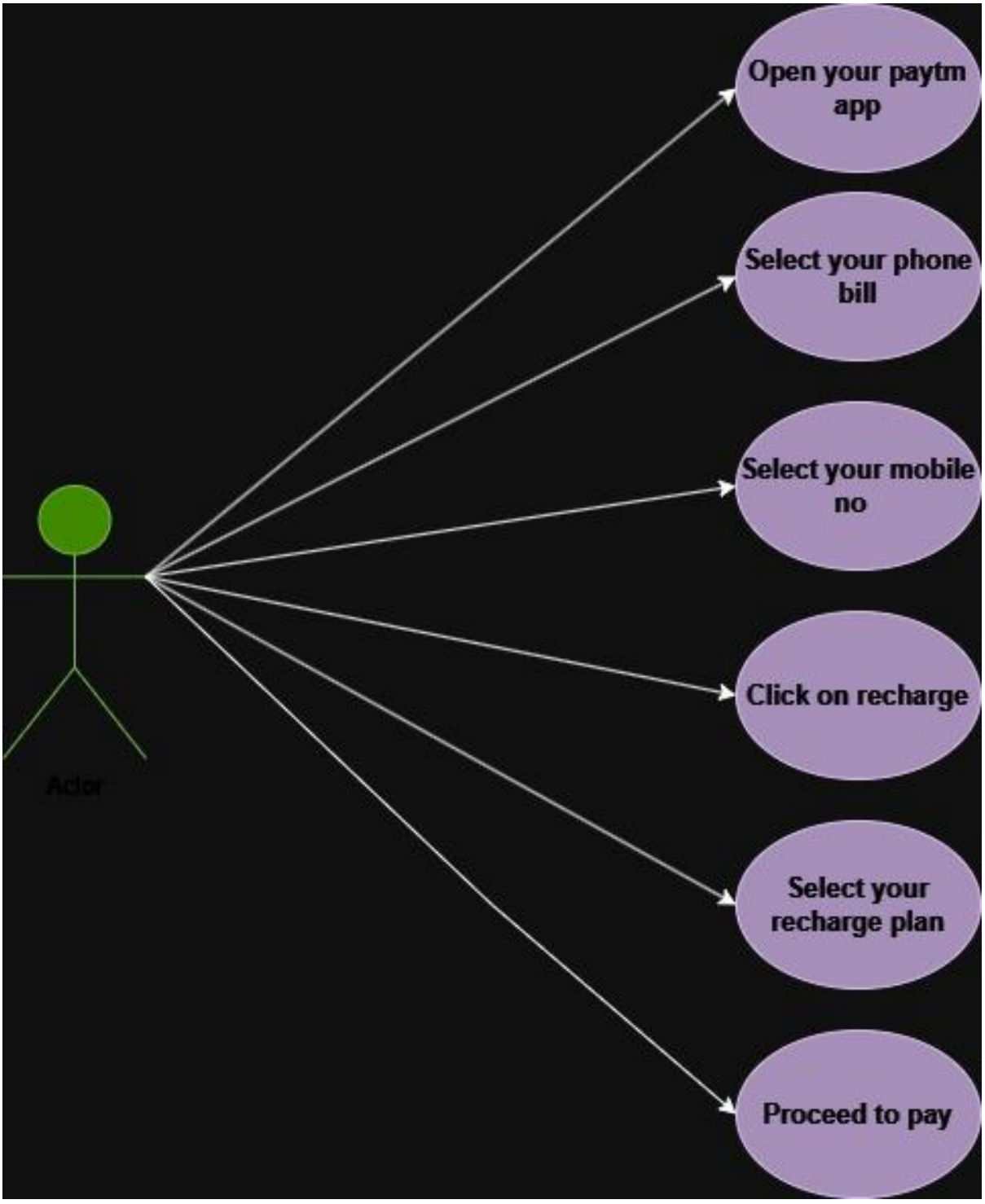
- Adaptive maintenance: Adapting the existing solution to the new platforms.

-Perfective Maintenance: Implementing the new requirements.

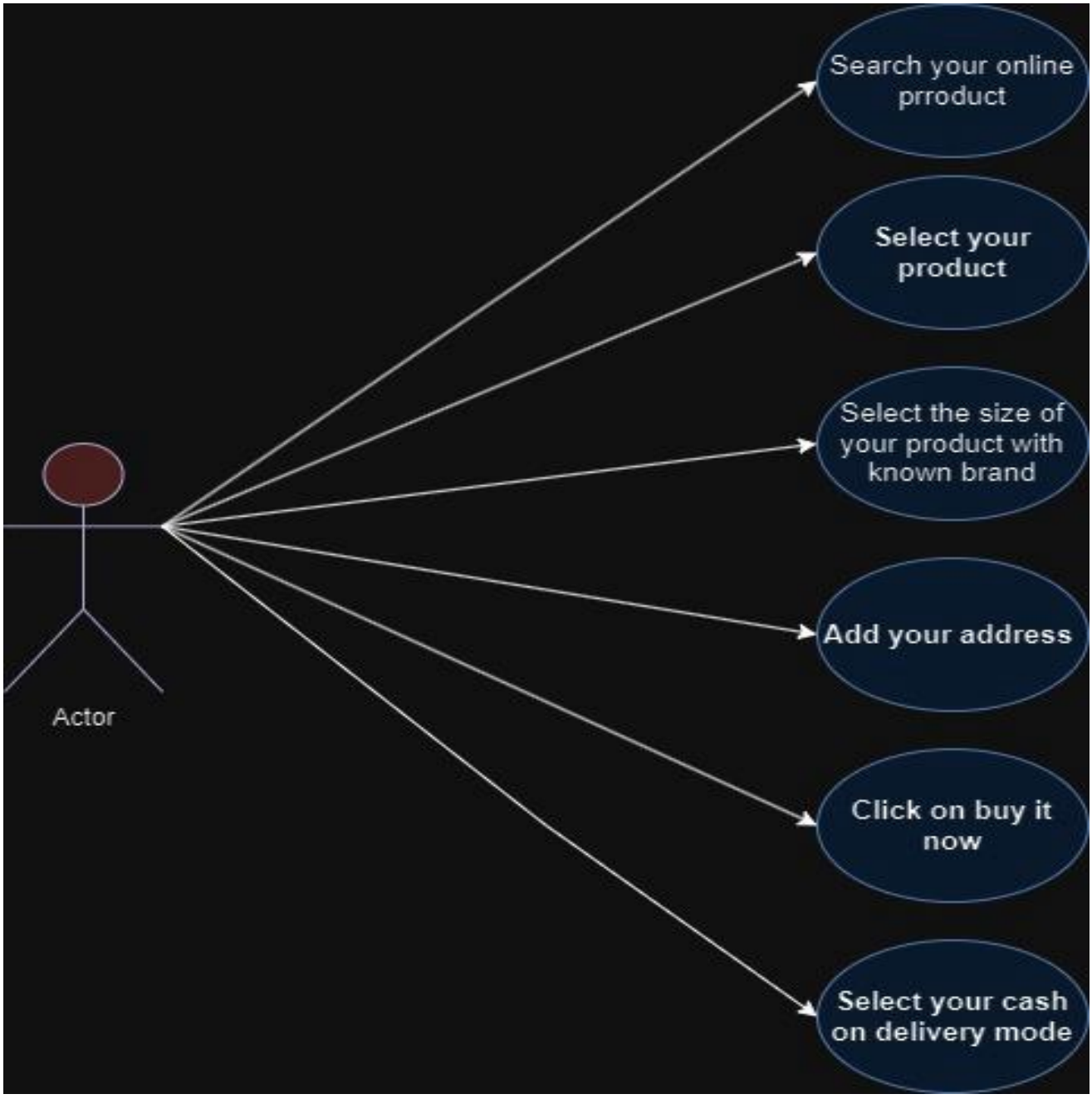
6. **Draw Usecase on Online book shopping**.



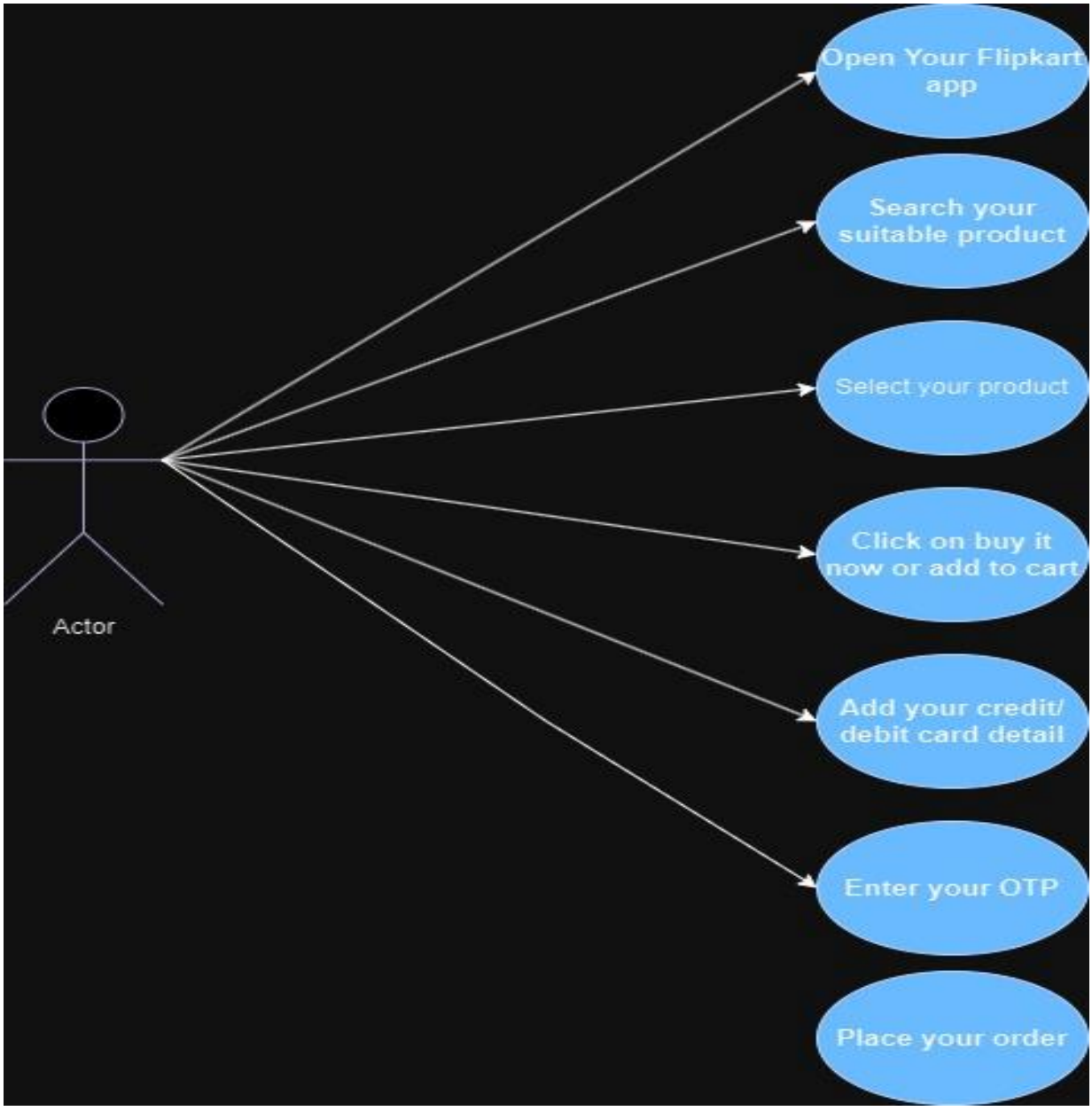
7. Draw Use case on online bill payment system (paytm).



8. Draw usecase on Online shopping product using COD.



9. Draw usecase on Online shopping product using payment gateway.



10. 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

Programming: set of instructions that can be executed by the developer.

OOPs (python, java, react)

11. Write Basic Concepts of oops.

There are basically six concept of oops which are:

Basic Concepts of OOP:

. Class

. Object

. Encapsulation

. Inheritance

. Polymorphism

. Abstraction

12. What is class?

Class is a collection of a data member and member function with its behaviour. Blueprint / template which is a collection of data members and member function.

Class: Employee

Data members : Emp_id, Emp_name, Salary, Designation, Dept, City, DOJ

Member Function (Method) : join_emp(), get_commision(),

13. What is 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.

Object will always represent the relevant class.

Fruit is class.

Apple is an object.

Employee is a class.

Mr. X is an object.

Student is a class.

Piyush is an object

class Employee

{

data members: Emp_id, Emp_name, Salary, Designation, Dept, City, DOJ

member function : join_emp(), get_commision(),

}

14. What is encapsulation?

A wrapping up of data and functions into a single unit is called Encapsulation. IT hide/include private access of data member & member function.

15. What is inheritance?

To derive the properties/features/attributes of One class to another class.

To reusability of code.

Main class: Parent Class / Base Class / Super Class

Another class from Main: Child class / Derived Class / Sub class

```
class Per_student
{
    rollno, name, course, std..
}
class Edu_student
{
    Marks, per, grade, per
}
```

-Types of Inheritance:

1. Single Inh.
2. Multilevel Inh.
3. Multiple Inh.
4. Heirarchical Inh.
5. Hybrid Inh.

16. What is polymorphism?

An ability to take one name having many different forms.

1) Compile time / Static binding / Overloading

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

eg class ABC

{

a, b, c, d

add(a, b)

add(c, d)

add(a, b, c)

add(a,b,c,d)

}

2. Run time / Dynamic binding/ Overriding (Always done with Inheritance)

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

Eg.

class A

{

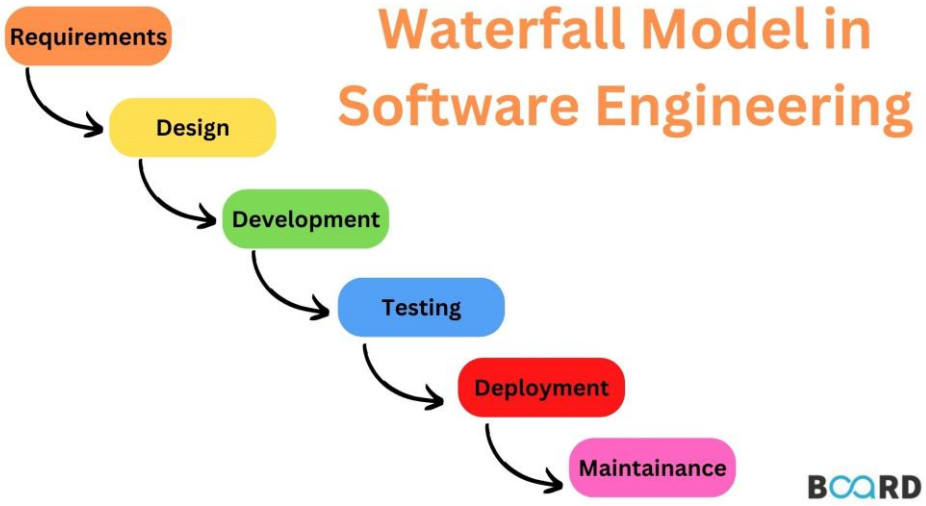
add()

}

17. 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.”

18. Explain Phases of the waterfall model.



- Planning,
- Analysis,
- Design,
- Implementation,
- Testing, Maintenance

19. Write phases of spiral model.

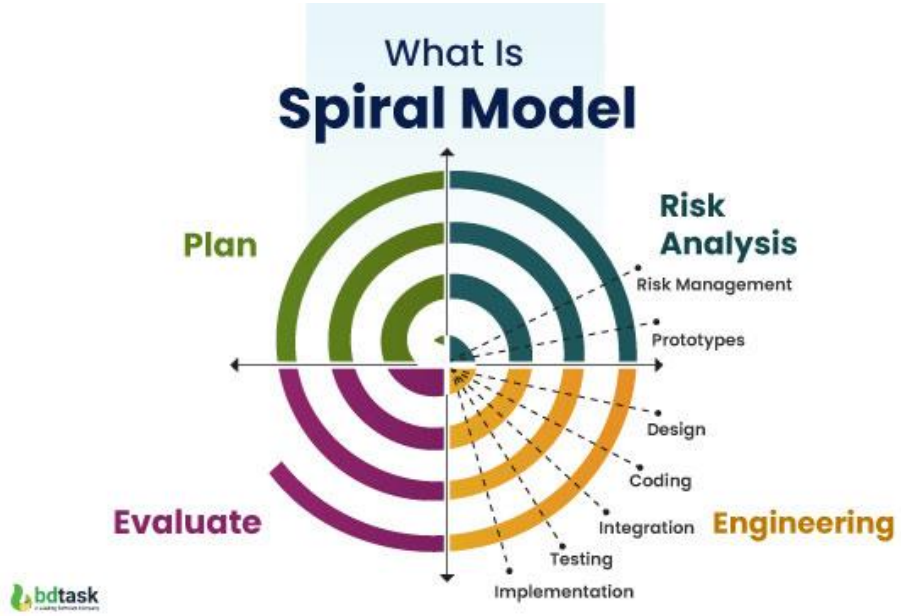
Spiral Model is very widely used in the software industry as it is in synch with the natural development process of any product learning with maturity also involves minimum risk for the customer as well as the development firms.

A. Planning / Requirement Gathering / Feasibility Study

B. Risk Analysis / Design

C. Engineering / Coding

D. Customer Evaluation / Testing



20. Write agile manifesto principles.

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

21. 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.
- Agile Methods break the product into small incremental builds
- These builds are provided in iterations.
- Each iteration typically lasts from about one to three weeks.

Pros:

1. Very realistic approach
2. Rapid delivery.
3. Functionality can be developed rapidly
4. Resource requirements are minimum.
5. Little or no planning required

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