

# **Assignment – 1**

## **( Module – 1 )**

### **1. What is SDLC ?**

Ans. SDLC means software development life cycle which means 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 planning, analysis , design, coding and implementation , testing and maintenance.

### **2. What is software testing ?**

Ans. Software testing is a process used to identify the correctness, completeness, and quality of developed computer software. In other words software testing is process of validating and verifying.

### **3. What is Agile methodology ?**

Ans. 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.

### **4. What is SRS ?**

Ans. A software requirements specification is a complete description of the behavior of the system to be developed.

This standard describes possible structures, desirable and qualities of a software requirements specification.

## **5. What is oops ?**

Ans. Oops is for identifying and assigning responsibilities to objects.

Object oriented programming has a web of interacting objects, each house keeping its own state.

Objects of program interact by sending messages to each other.

## **6. Write basic concepts of oops.**

Ans. - Object

- Class
- Encapsulation
- Inheritance
- Polymorphism ( overriding , overloading )
- Abstraction

## **7. What is object.**

Ans. Object is the basic unit of object oriented programming.

An “object” is anything to which a concept applies.

That is both data and function that operate on data are bundled as a unit called as object.

## **8. What is class.**

Ans. A class represents an abstraction of the object and abstracts the properties and behavior of the object.

Class can be considered as the blue print for an object and describe the properties and behavior of the object, but without any actual existence.

An object is a particular instance of a class which has actual existence and there can be many objects for a class.

## **9. What is encapsulation.**

Ans. Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.

Encapsulation is placing the data and the functions that work on that data in the same place. While working with procedural languages, it is not always clear which functions work on which variables but object oriented programming provides you framework to place the data and the relevant functions together in the same project.

## **10. What is inheritance.**

Ans. Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship.

Inheritance describes the relationship between base class and derived class.

A class can get some of its characteristics from a parent class and then add unique features of its own.

## **11. What is polymorphism.**

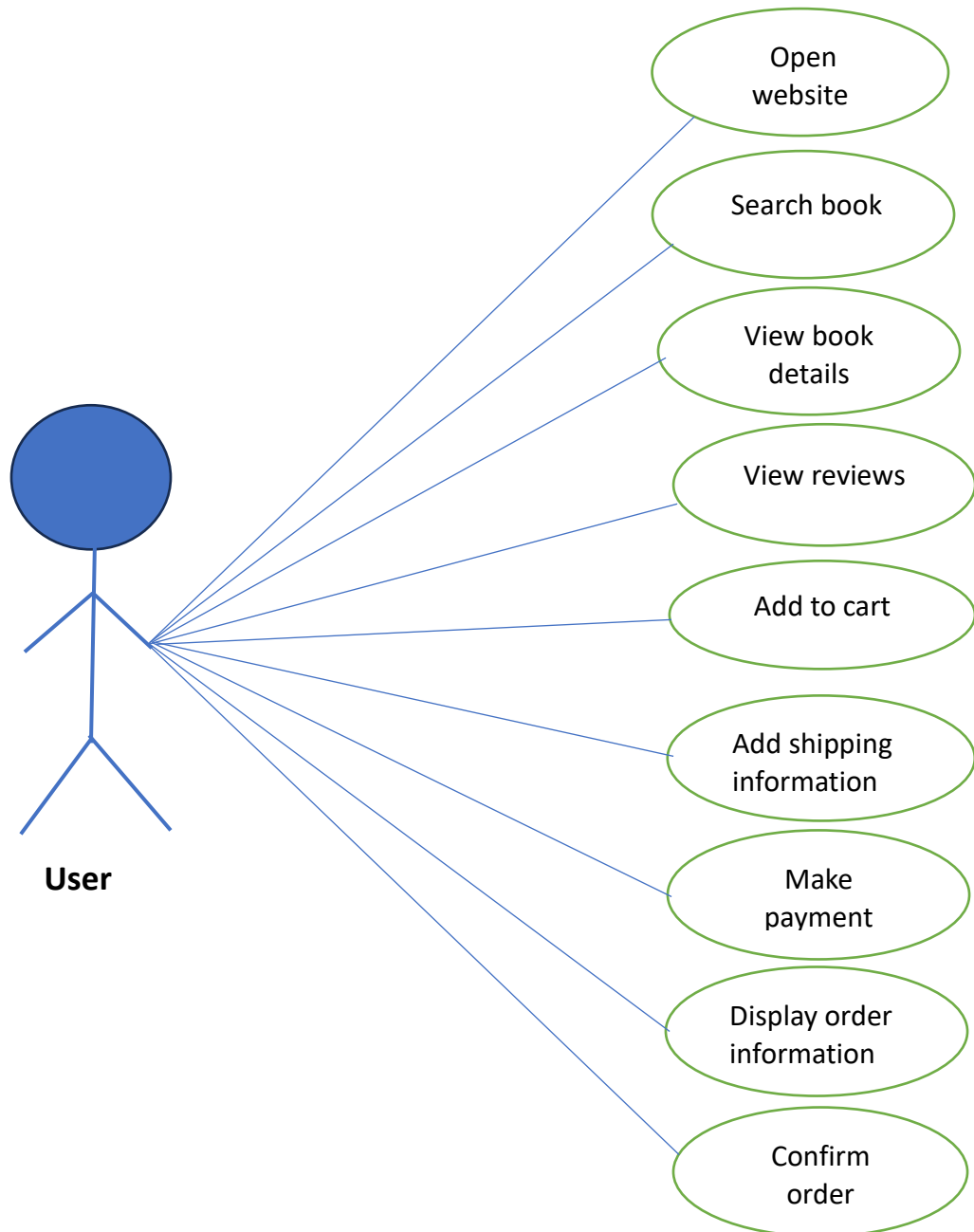
Ans. Polymorphism means “having many forms”.

It allows different objects to respond to the same message in different ways, the response specific to the type of object.

poly refers to many. This is a single function or an operator functioning in many way different upon the usage is called polymorphism.

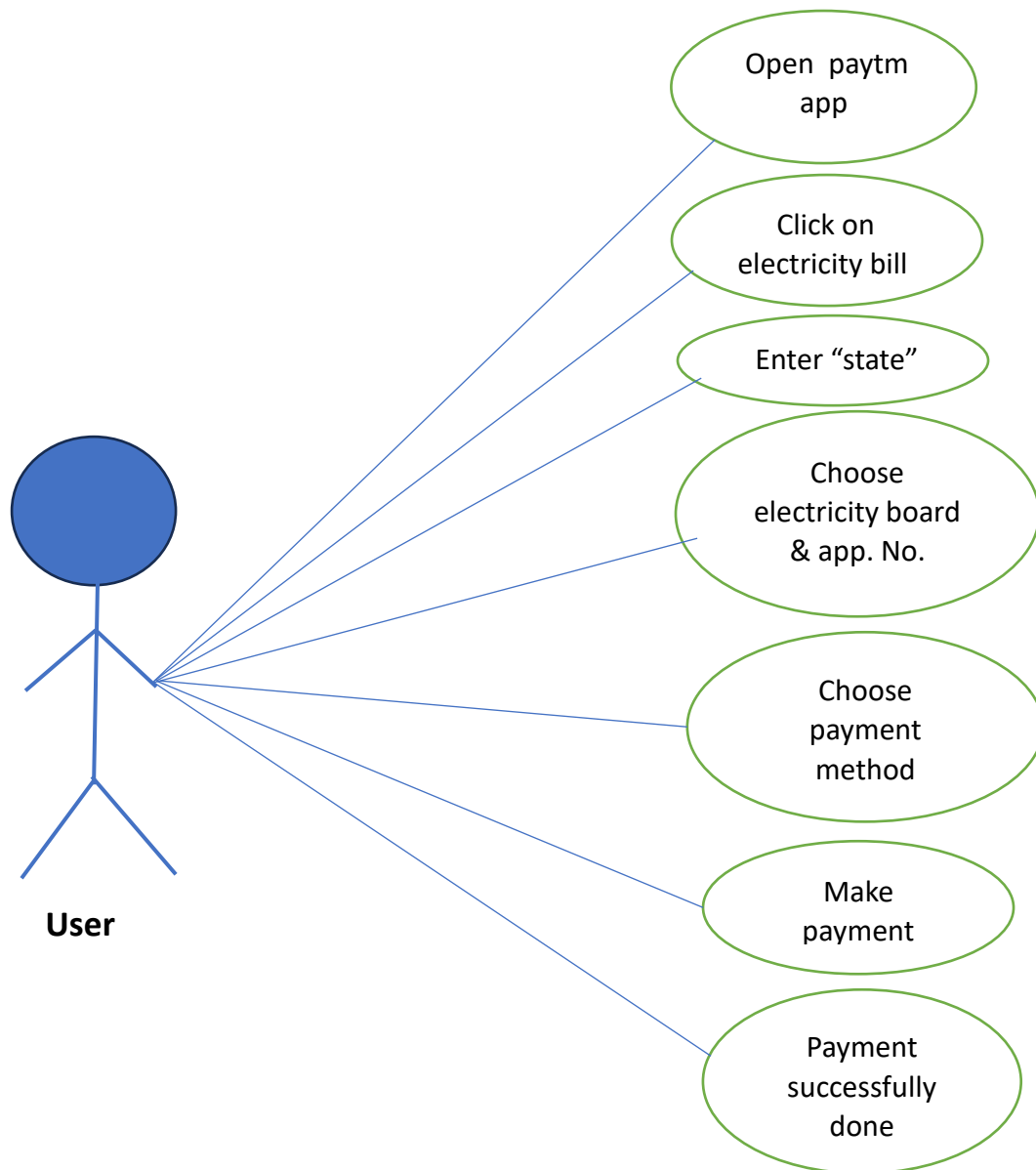
## 12. Draw usecase on online book shopping.

Ans.



### 13. Write usecase on online bill payment system. ( paytm )

Ans. Here is a example of electricity bill payment through paytm.



## **14. Write SDLC phases with basic information.**

### **Ans. 1. Requirement Gathering**

In this phase we got requirements may be documented in written form and they may be incomplete or even incorrect . using this scenarios we can establish the customer needs.

So after this all requirement gathering we can do planning about the project what the customer need.

### **2. Analysis phase**

This phase is about analysis all the required information what we have collected. ideally the result at the end we get is a requirement document and this documents in a clear way is to be built and this document is the architecture of software.

After completed this phase we get details about computer programming language, algorithms, memory size and many other details

### **3. Design phase**

In this phase we design the architecture document.

Design team will expand the information established in the required document and design team will also converts the typical scenarios into test plan.

### **4. Implementation phase**

As we got requirement document from analysis phase and architecture of document from design phase. Now the team should build exactly what has been requested.

Implementation phase deals with issues of quality , performance and debugging.

Inthis phase still a room for innovation and flexibility like we want narrowly designed component or it be more general to satisfy the guideline.

So in the end of this we got the product itself .

### **5. Testing phase**

This phase is a differently performed after the implementation phase.

We take any kind of software, first the quality of the software is very important. If the customer is satisfied with the quality of the software then they will wait for the next version of the software, otherwise not.

So in this phase testing the software with different kind of testing like unit testing, application testing, stress testing so we confirmed that the product what we have built is now ready to launch.

## **6. Maintenance phase**

It's about enhancing the deployed software and this phase comes when the product has already launched in the market.

The developing team will have some mechanism to track the defect and deficiencies and the team will fix defects.

In short it's about repair defects and adapt the solution to the new requirements.

## **15. Explain phases of the waterfall model.**

Ans. 6 phases of waterfall model as below.

### **1. Requirement Gathering and analysis**

All possible requirements of the system to be developed are captured in this phase and documented in a requirements specification document.

### **2. System design**

The requirements specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system.

### **3. Implementation**

With inputs from the system design, the system is first developed in small programs called unit, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as unit testing.

### **4. Integration and testing**

All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

## **5. Deployment of system**

Once the functional and non-functional testing is done , the product is deployed in the customer environment or releases into the market.

## **6. Maintenance**

There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

## **16. Write phases of spiral model.**

Ans. Spiral model has four phases

### **1. planning**

This phase includes requirement gathering and analysis based on the requirements , objectives are defined and different alternate solutions are proposed.

### **2. risk analysis and resolving**

In this quadrant, all the proposed solutions are analyzed and any potential risk is identified , analyzed and resolved.

### **3. Develop and testing**

This phase includes the actual implementation of the different features. All the implemented features are then verified with thorough testing.

### **4. evalution and planning of the next phase**

In this phase the software is evaluated by the customer. It also includes risk identification and monitoring like cost overrun or schedule slip page and after that planning of the next phase is started.



## 17. Write agile manifesto principles.

Ans. Four values of agile manifesto :

Individuals and interactions over process and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

Agile manifesto principles are below.

- To satisfy the customer through the early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity: the art of maximizing the amount of work not done is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

## **18. Explain working methodology of agile model and also write pros and cons.**

Ans. **Agile methodology** :- Agile SDLC model is a combination of iterative and incremental process models with focus on 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.

Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis design, coding, unit testing and acceptance testing.

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

**Pros. :-** Very realistic approach.

- Rapid delivery.

- Functionality can be developed rapidly.

- Resource requirements are minimum.

- Little or no planning required.

- promotes team work and cross training .

- suitable for fixed or changing requirements.

- gives flexibility to developers.

**Cons. :-** More risk of sustainability, maintainability and extensibility.

- Depends heavily on customer interactions.

- Very high individual dependency.

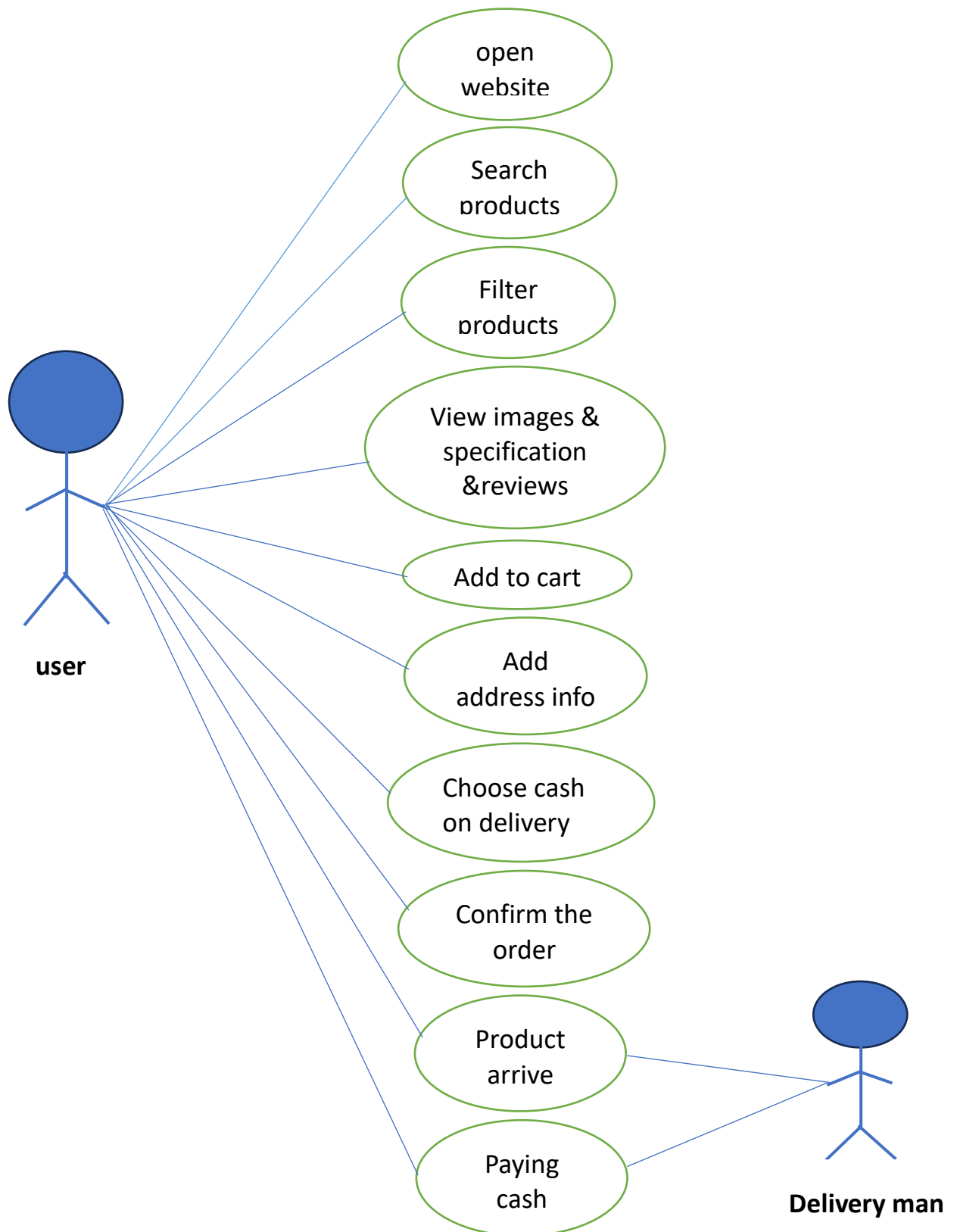
- Minimum documentation generated.

- Not useful for small projects.

- Not suitable for handling complex dependencies.

## 19. Draw usecase on online shopping product using COD.

Ans.



## 20. Draw usecase on online shopping product using payment gateway.

Ans.

