**Assignment : Module 1**

1. **What is SDLC?**

**Ans:** SDLC is (Software Development Life Cycle ) it is a structure imposed on the development of a software product that defines the process for planning , implementation, testing, documentation , and on going maintenance and support.

1. **What is software testing?**

**Ans:**  Software Testing is a process used to identity the correctness, completeness, and quality Of developed computer software. There are two types of testing.

1. Static testing
2. Dynamic testing

* **Static testing**: It is the process to find the defects without interacting with the software, Application or without executing the code. Static testing done under the process of verification.
* **Dynamic Testing**: It is the process of testing the software with the intergration of software or executing the code. It is done during the process of validation

.

1. **What is Agile Methodology ?**

**Ans:** Agile Methodology model is a combination of interative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

1. **What is SRS?**

**Ans: Software Requirements Specification.** It is a compete description of the behaviour of the system to be developed. There is two types of requirements ?

1. Fuctional
2. Non functional

* **Functional :** Functional requirements are very important system requirement in the system design process. These requirement are the technical specification, systm design parameters and guidelines , data processing and calculation modules ets, of the proposed systm.
* **Non functional** : Non functional reqierements are specify criteria that can be used to judge the operation of a system , rather than specific behaviors.

1. **What is OOPS ?**

**Ans : Object Oriented Programing.** It is focus on object/data rather than process. An object is a like Black Box. The internal details are hidden. Object a derived from abstact data type.

* **For example : Python, Java, Ruby, C#, etc.**

1. **What is Basic concept of OOP?**

**Ans : There are 6 basic concept of OOP.**

1. **CLASS :** It is a blueprint / collection for an object.
2. **Object :** It is a part / example / instance representative entity of class.

**“ class can have many object but only relute to a single class “**

**Example** : Apple is a fruit

* There are Apple is an Object and Fruit is a Class**.**

1. **Encapsulation :** Wrapping up of data in to a a single unit.
2. **Inheritance :**  Ability to adapt to behaviour of parent class to child class.

* Here two or more class are in parent child relation**.**

1. **Polymorphim :** “ Ability to represent in different way”
2. **Absraction :** Abstraction is the representation of the essential features of anobject.
3. **What is Object ?**

**Ans :** An object represent an individual , identifiable, unit, or entity, either real or abstract, with a well defined role in the problem domain. An object is anything to which a concept applies. An object has the responsibility to know the responsibility to do.

1. **What is class ?**

**Ans:**  A class represent an abstraction of the object and abstract the properties and behaviour of the object. An object is a particular instance of a class which has actual existence and there can be many object for a class. We do not actually buy these blueprint but the actual object.

1. **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 object. Encapsulation is placing the data and the functions that work on that data in the same place.

1. **What is inheritance ?**

**Ans** : Inheritance means that one class inherits the characteristics of another class. This is also called a is a relationship. This is a very important concept of object oriented programming since this feature helps to reduce the code size. Inheritance describes the relationship between two classes.

1. **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 the object. The most important aspect of an object is its behaviour .

1. **Write SDLC phases with basic introduction .**

**Ans :** A Software Development Life Cycle is essentially a series of steps or phases , that provide a model for the development and lifecycle management of an application piece of software. SDLC is a structure imposed on the development of a software product that defines the process for planning, implemention, testing, documentation, deployment, and ongoing maintenance and support.

* **There is some phases of SDLC.**

1. **Requirements** : Establish customer needs
2. **Analysis :** Model and specify the requirements**.**
3. **Design :** Model and specify a solution { Make a design of requirements}
4. **Implementation :** Construct a solution in Software.
5. **Testing** : Validate the solution against the requiremenrts**.**
6. **Maintenance** : Repair defects and adapt the solution to the new Requirements.
7. **Explain Phase of the Waterfall model.**

**Ans :** The waterfall model was the first process model to be introduced. It is also referred to as a linear sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall model is the earliest SDLC approach that was used for software development. The whole process of software development is divided in to seprate phases. In this waterfall model typically the outcomes of one phase acts as the input for the next phase sequentially.

* **There is some phases of waterfall model.**

1. **Requirement Analysis :** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
2. **System Design** : The requirement specification from first phase are studied in this phase and the system design is prepared. This system designs helps in specifying hardware and system requirements and helps in defining the overall system architecture.
3. **Implementation** : With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as unit testing.
4. **Intergration and testing** : All the units developed in the implementation phase are intergrated in to system after testing of each unit. Post intergration the entires 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 released 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 version are released.
7. **Write phases of spiral model ?**

**Ans :** Spiral model is very widely used in the software industry as it is in synch with the natural development process of any product i.e. learning with maturity and also involves minimum risk for the customer as well as the development firms. The spiral model is a software development model designed to control risk. The spiral model has FOUR phases . planning, risk analysis, product development, and nexr phase planning or evaluation.

1. **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 spiral.
2. **Risk Analysis :** In the risk analysis phase the risk associated with the project are identified and evaluated.
3. **Engineering :** In the engineering phase the software is developed based on the requirements gathered in the previous iteration.
4. **Evaluation** : In the evaluation phase the software is evaluated to determine if it meets the customer’s requirements and it is of high quality.
5. **Write agile manifesto principles.**

**Ans : There** are some manifesto principle of Agile methodology below.

1. **Individual and Interactions Over Processes and Tools**  
   The first value in the Agile Manifesto is “Individuals and interactions over processes and tools process or the tools drive development, the team is less responsive to change and less likely to meet customer needs. Communication is an example of the difference between valuing individuals versus process. In the case of individuals, communication is fluid and happens when a need arises.
2. **Working Software Over Comprehensive Documentation**

. Agile documents requirements as user stories, which are sufficient for a software developer to begin the task of building a new function. Technical specifications, technical requirements, technical prospectus, interface design documents, test plans, documentation plans, and approvals required for each. The list was extensive and was a cause for the long delays in development.

1. **Customer Collaboration Over Contract Negotiation**

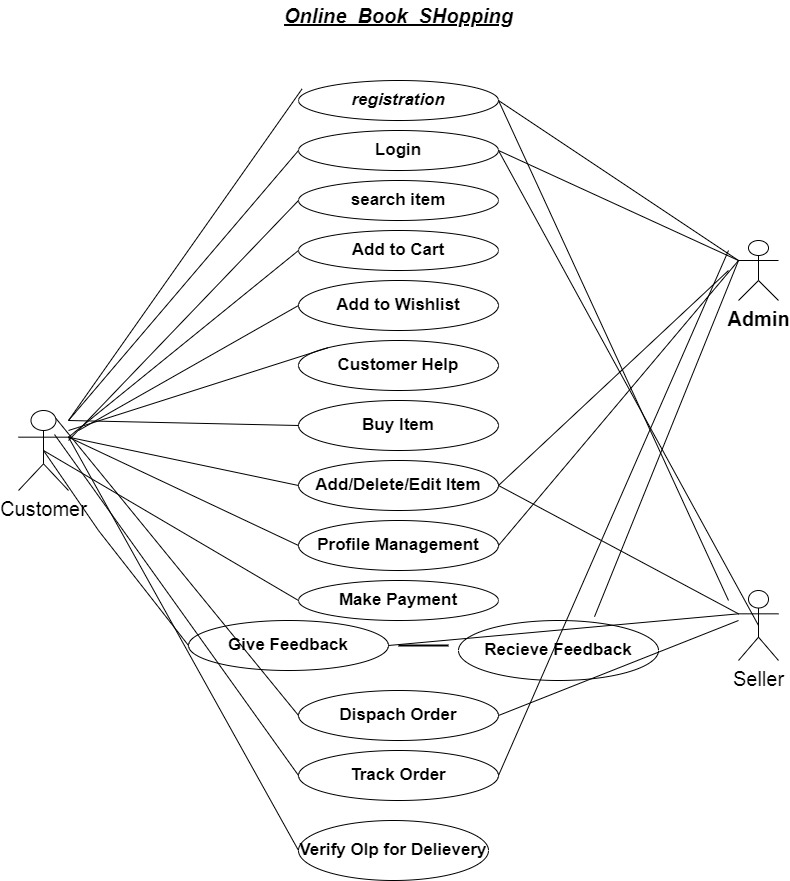
Negotiation is the period when the customer and the product manager work out the details of a delivery, with points along the way where the details may be renegotiated. Collaboration is a different creature entirely. With development models such as Waterfall, customers negotiate the requirements for the product, often in great detail, prior to any work starting.

1. **Responding to Change Over Following a Plan** With Agile, the shortness of an iteration means priorities can be shifted from iteration to iteration and new features can be added into the next iteration. Agile’s view is that changes always improve a project; changes provide additional value.
2. **Explain working methodology of Agile model and also write pros and cons?**

**Ans :**  The meaning of agile is swift or versatile. Agile SDLC model is a combination of iterative and incremental process model 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.

* **Agile model Pros.**
* Functionality can be developed rapidly and demonstrated.
* Promotes teamwork and cross training.
* Resource requirements are minimum.
* Suitable for fixed or changing requirements
* Easy to manage Gives flexibility to developers.
* **Agile model Cons**
* Not suitable for handling complex dependencies.
* More risk of sustainability, maintainability and extensibility.
* 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.

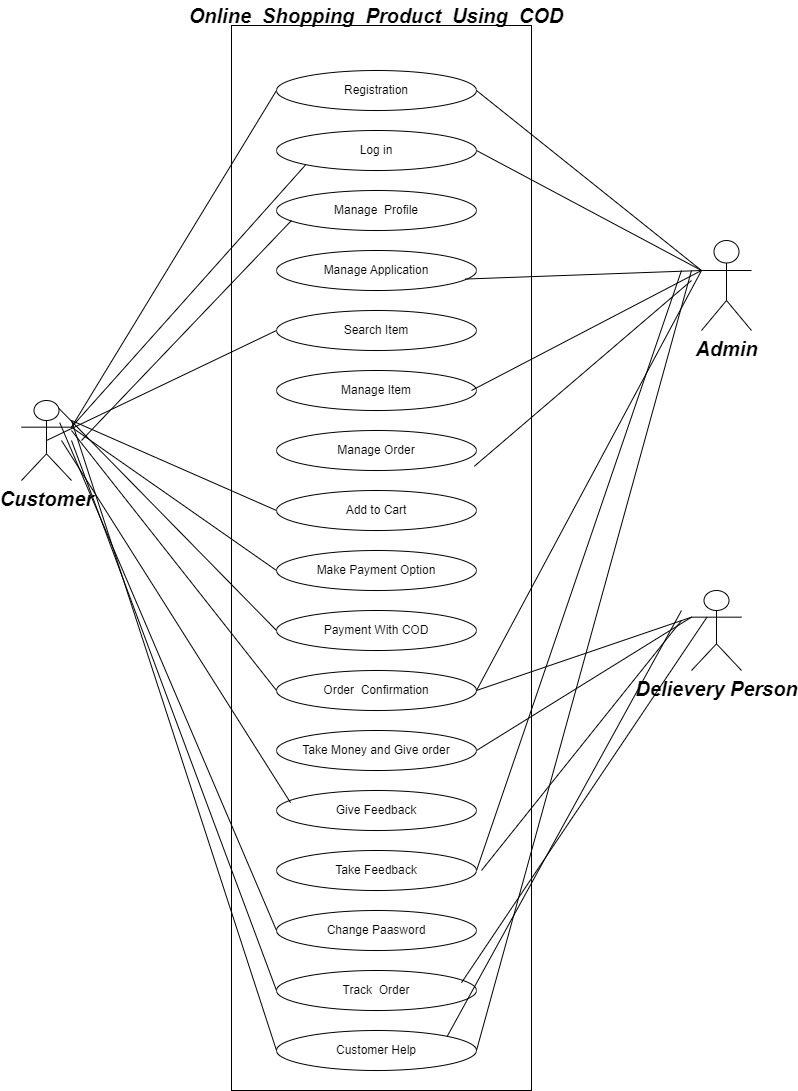
1. **Draw usecase on online book shopping.**



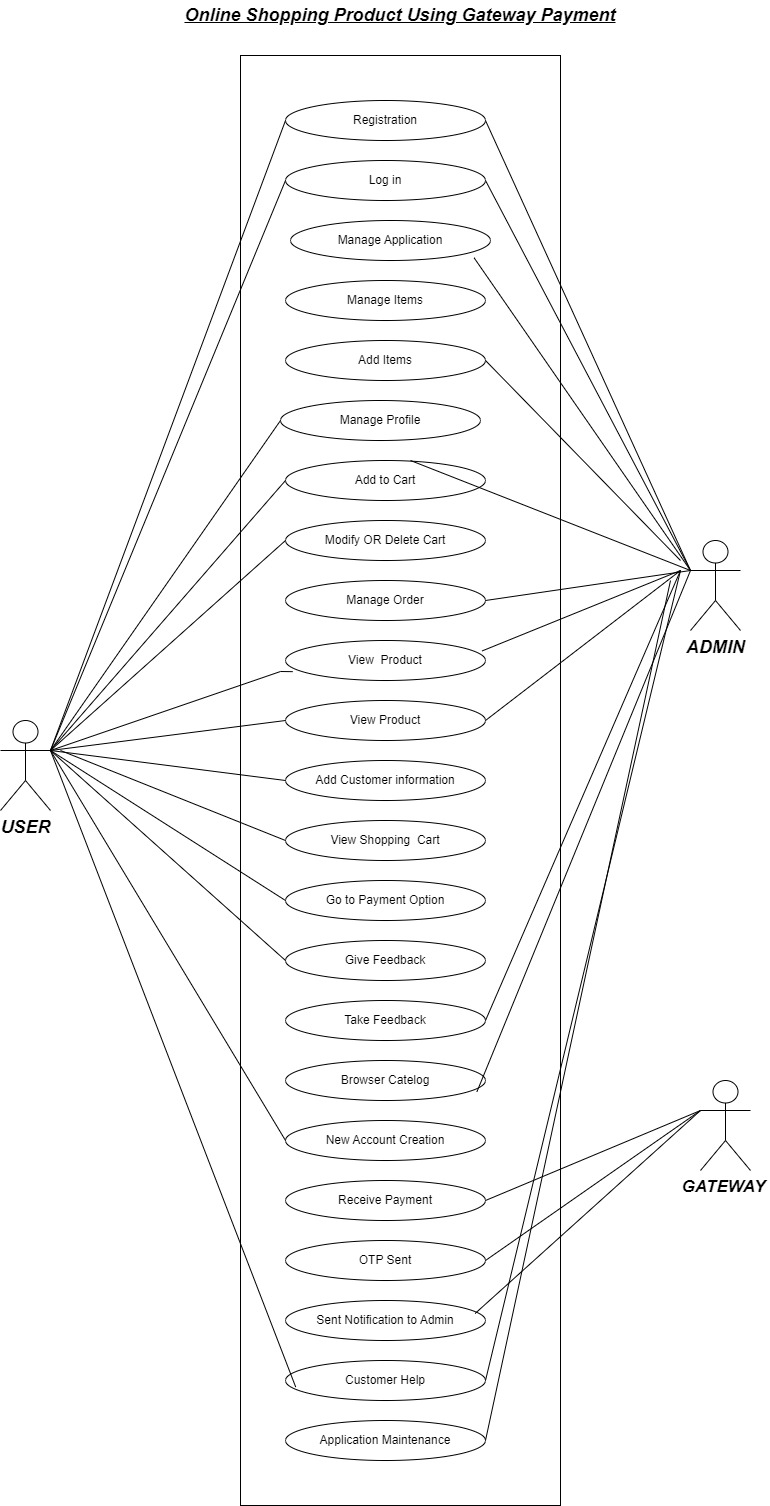
1. **Draw Usecase on online bill payment system {PAYTM}**

****

1. **Draw Usecase on online shopping Product using COD.**

****

1. **Draw Usecase on online shopping product using payment gateway.**

****