**Software Testing**

• **What is SDLC?**

SDLC = Software Development Life Cycle

|  |  |
| --- | --- |
| Requirements collection/ Gathering | Establish customer needs. |
| Analysis | Model and specify the requirements – “What”. |
| Design | Model and specify a solution – “Why”. |
| Implementation | Construct a solution in software |
| Testing | Validate the solution against the requirements. |
| Maintenance | Repair defects and adapt the solution to the new requirements. |

**• What is software testing ?**

Testing is the process of evaluating a system or its component with the intent to find that whether it satisfies the specified requirements or not.

Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

**• What is agile methodology?**

Agile SDLC model is a combination of iterative and incremental process models.

It divides the software into small incremental builds, this build are provided in iterations, that means the big projects are divided into small chunks.

**• What is SRS?**

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

Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional requirements.

**• What is oops?**

An object is like a black box.

The internal details are hidden.

**• Write Basic Concepts of oops?**

Object

Class

Encapsulation

Inheritance

Polymorphism

Overriding

Overloading

Abstraction

**• What is object?**

An object represents an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain.

**• What is class?**

Class is a collection of data member and member function.

**• What is encapsulation?**

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.

**• What is inheritance?**

Making a class from an existing class. Deriving the attribute of some other class.

**• What is polymorphism?**

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.

**Type : Overloading**

In method overloading, multiple methods having same name can appear in a class, but with different signature.

**Overriding**

Here, the actual method called will depend on the object at runtime, not the reference type.

**• Write SDLC phases with basic introduction?**

|  |  |
| --- | --- |
| Requirements collection/ Gathering | Establish customer needs. |
| Analysis | Model and specify the requirements – “What”. |
| Design | Model and specify a solution – “Why”. |
| Implementation | Construct a solution in software |
| Testing | Validate the solution against the requirements. |
| Maintenance | Repair defects and adapt the solution to the new requirements. |

**• 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.

**• Write agile manifesto principles?**

**Individuals and interactions** - in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

**Working software** - Demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.

**Customer collaboration** - As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements..

**Responding to change** - agile development is focused on quick responses to change and continuous development.

**• 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.

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

**Pros**

Good model for environments that change steadily.

Minimal rules, documentation easily employed.

Little or no planning required .

Easy to manage .

Gives flexibility to developers.

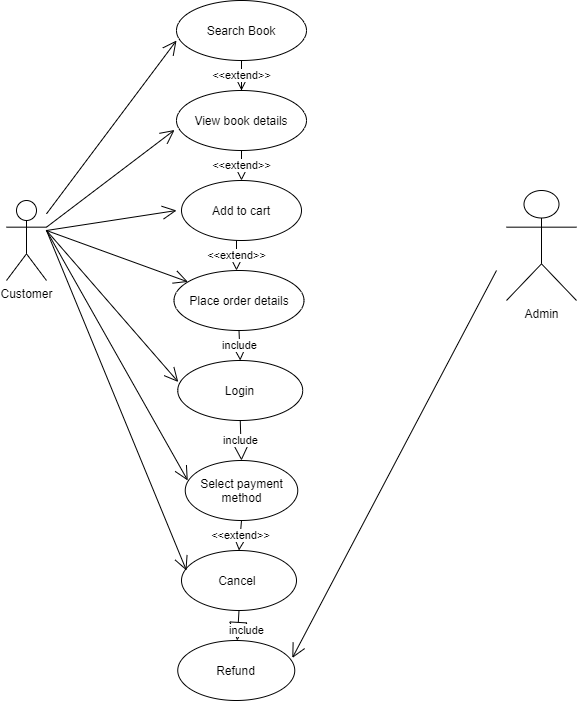
**Cons**

Not suitable for handling complex dependencies.

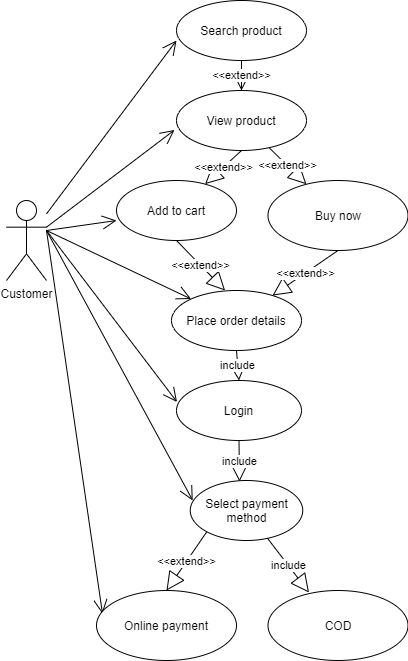
More risk of sustainability, maintainability and extensibility.

There is very high individual dependency, since there is minimum documentation generated.

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

****

**• Draw usecase on Online shopping product using COD.**

****

**• 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 i.e. learning with maturity and also involves minimum risk.

**Planning :** Determination of objectives, alternatives and constrains.

**Risk Analysis :** Analysis of alternatives and identification/ resolution of risk.

**Customer Evaluation** : Assessment of the result of engineering.

**Engineering** : Development of the “next level” product.

**• Explain Phases of the waterfall model.**

The waterfall is unrealistic for many reasons, especially:

Requirements must be “frozen” to early in the life cycle.

Requirements are validated too late.

**Requirement Gathering :**

Features

Usage scenarios

Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.

Requirements will Change!

Inadequately captured or expressed in the first place

User and business needs change during the project .

**Analysis Phase :**

This phase defines the problem that the customer is trying to solve.

The deliverable result at the end of this phase is a requirement document.

Ideally, this document states in a clear and precise fashion what is to be built.

This analysis represents the “what” phase.

**Design Phase :**

Design Architecture Document.

Implementation Plan.

Critical Priority Analysis.

Performance Analysis.

Test Plan .

**Implementation Phase :**

In the implementation phase, the team builds the components either from scratch or by composition.

Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility.

**Testing Phase :**

Quality is a distinguishing attribute of a system indicating the degree of excellence.

Regression Testing.

Internal Testing. Unit Testing.

Application Testing.

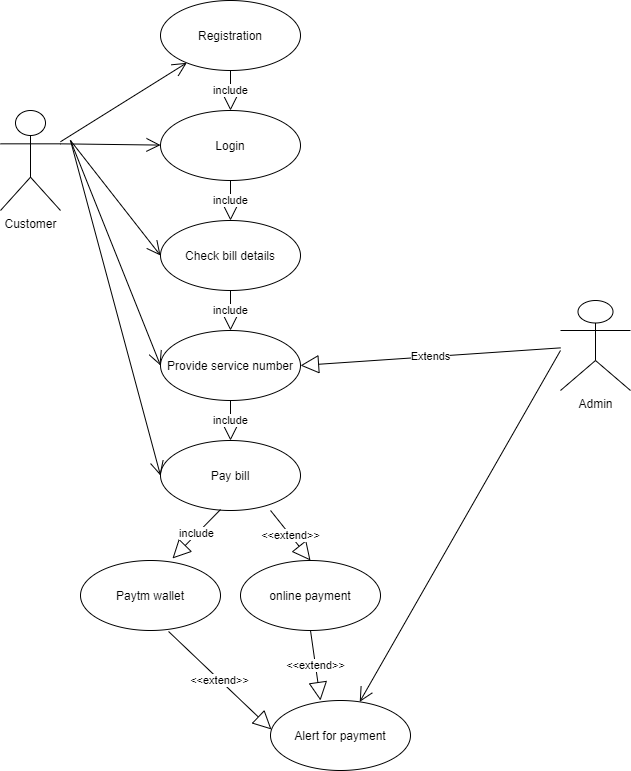
Stress Testing .

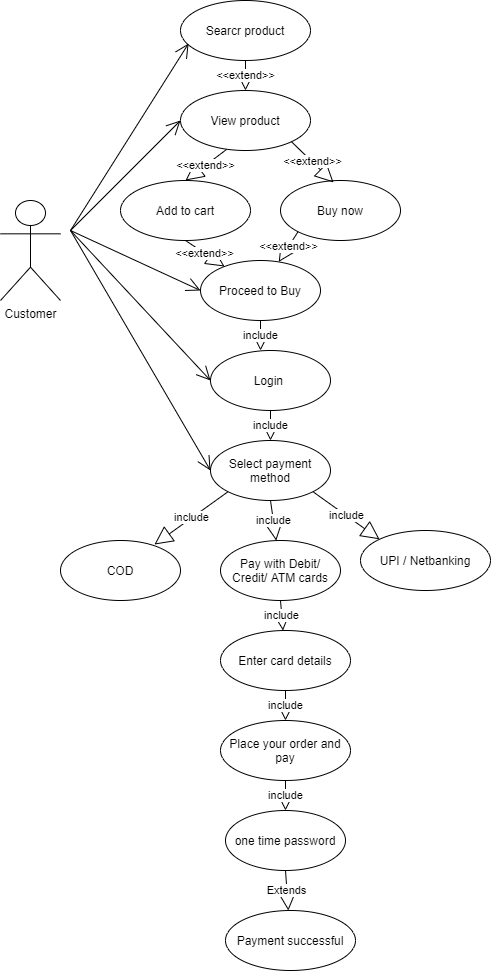
**Maintenance Phase :**

Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects.

Software maintenance is also one of the phases in the System Development Life Cycle (SDLC), as it applies to software development. The maintenance phase is the phase which comes after deployment of the software into the field.

**• Draw Usecase on online bill payment system (paytm).**

**• Draw usecase on Online shopping product using payment gateway.**

****