**Software Testing Assignment**

**Module-I (Fundamental)**

1. **What is SDLC?**

SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment and ongoing maintenance and support.

1. **What is Software testing?**

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

1. **What is agile methodology?**

Agile is a 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.

1. **What is SRS?**

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

1. **What is OOPS?**

OOPS (Object Oriented Programming System) is a programming paradigm based on the concept of objects ,which can contain data and code: data in the form of fields and code in the form of procedures(method).

1. **Write basic concept of oops.**

* Class
* Object
* Encapsulation
* Inheritance
* Polimorphism
* Abstraction

1. **What is object?**

Object is an instances of a class.

1. **What is class?**

Class is a collection of data members (variables) and member functions (process,method) with its behaviors.

1. **What is encapsulation?**

Encapsulation is the process of wrapping up of data (properties) and behavior (method) of an object into a single unit.

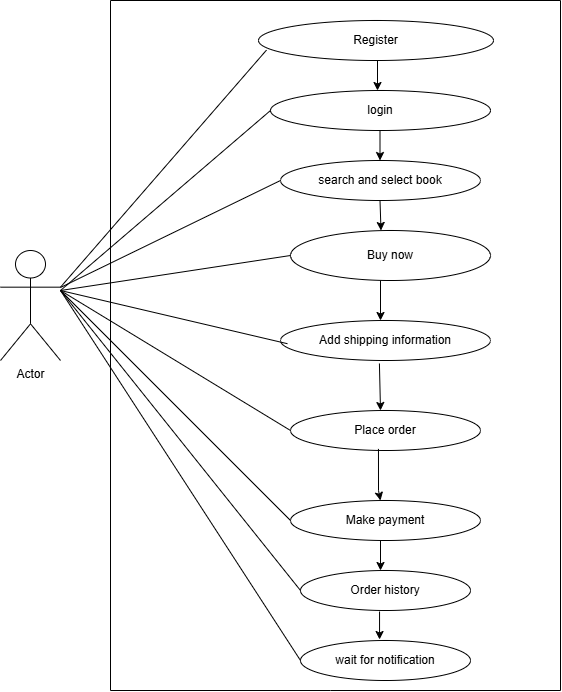
1. **What is inheritance?**

Inheritance is the properties of parent class extends into child class.

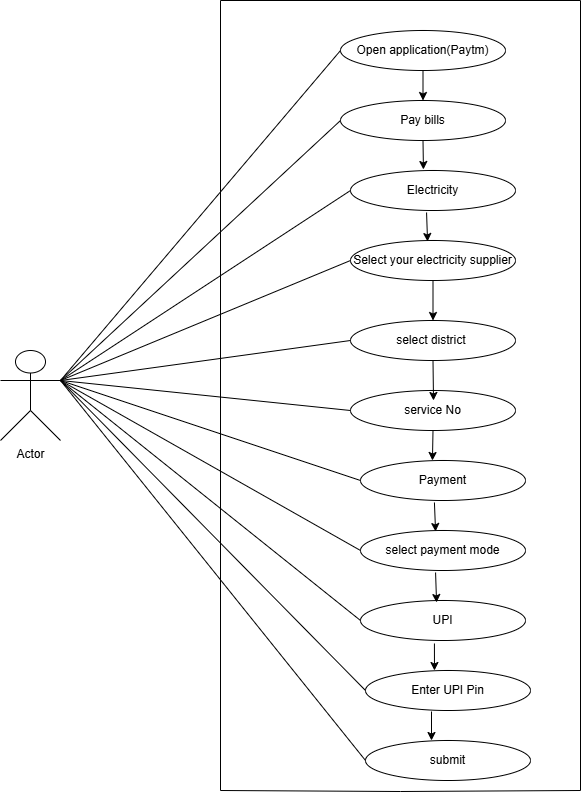
1. **What is polymorphism?**

Polimorphism is the ability to take one name having many forms or multiple forms.

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

****

1. **Draw usecase on online bill payment system (paytm).**

****

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

**Requirements Gathering:**

* Features
* Usage scenarios
* Requirements consist of natural language and supplemented by diagram and tables.
* Three types of problem arise:

1. Lack of clarity
2. Requirements confusion
3. Requirements amalgamation

**Analysis:**

* Analysis phase defines the requirement of the system
* Analysis represents the ‘What” phase.
* Requirement documentaries to capture the requirements from the customers.

**Design Phase:**

* Design Architecture document
* Implementation plan
* Teat plan
* Design team can expand the information established in the requirement document.

**Implementation Phase:**

* Implementation code
* Critical Error removal
* It deals with issue of quality, performance, baselines and debuggigng.

**Testing Phase:**

* Validate the solution against the requirements.
* To taste the product quality.

**Maintenance:**

* Maintenance is the process of changing the system after it has been deployed.
* Repair defects and adapt solution to the new requirement

**Three types:**

**Corrective Maintenance**

Identifying and repairing defects

**Adaptive Maintenance**

Adapting the existing solution to the new platform.

**Perfective Maintenance:**

Implementing the new software at a particula level of quality with known defects and deficiencies.

1. **Explain phases of water fall model?**

The classical software lifecycle models the software development as a step by step “waterfall” between the various development phases.

The waterfall is unrealistic for many reasons:

* Requirements must be “frozen”to early in the lifecycle
* Requirements are validated too late

**Applications**

* Requirements are very well documented,clear and fixed
* Product definition is stable
* Technology is not dynamic
* The project is short

**Pros**

* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model.
* Works well for smaller projects where requirements are very well understood.
* Process and results are well documented.

**Cons**

* Not suitable for the projects where requirements are at moderate to high risk of changing.
* No working software is produced until late during the lifecycle.
* Not a good model for comlex and object oriented projects.
* Poor model for long and ongoing projects.

1. **Write phases of spiral model?**

* **Planning:** Determination of objectives, alternatives and constrains.(initial requirements are necessary)
* **Risk Analysis:** Analysis of alternatives and resolution of risk. If don’t go use first prototype.
* **Engineering:** Development of next level product.
* **Customer Evaluation**: Assesment of result of engineering

1. **Write Agile manifesto principles?**

* **Individual and interaction:**

If any issue found in software search for another process or tool while in Agile, it is preferable to interact with client , manager or team regarding issue and make sure that issue get resolved.

* **Customer Collaboration:**

Customer collaboration is more important than contract negotiation.

* **Working Software:**

Agile is not saying that documentation is not needed but working software is much needed.

* **Responding to change:**

You need to implement new requirements in the middle of software, need to be a versatile to make changes in a software.

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

**Methodology of agile model**

* Agile methods break the product into small incremental builds.
* These builds are provided in iterations.
* Each iteration takes one to three weeks.
* Each iteration involves cross functional teams working on various areas like planning, requirement analysis, designing, coding, unit testing, acceptance testing.
* At the end of the iteration a working product is displayed to the customer and stakeholders

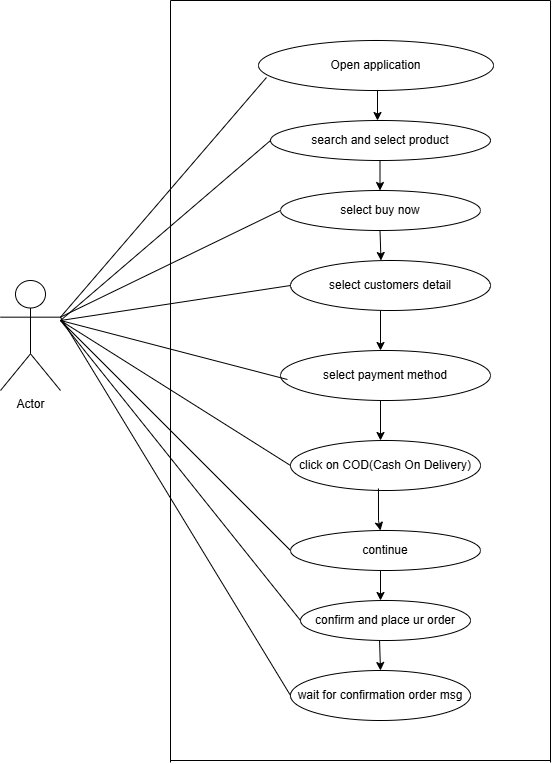
**Pros**

* Suitable for fixed or changing requirements
  + Resource requirements are minimum
  + Promote team work and cross training
  + Little or no planning required
  + Gives flexibility to the developers

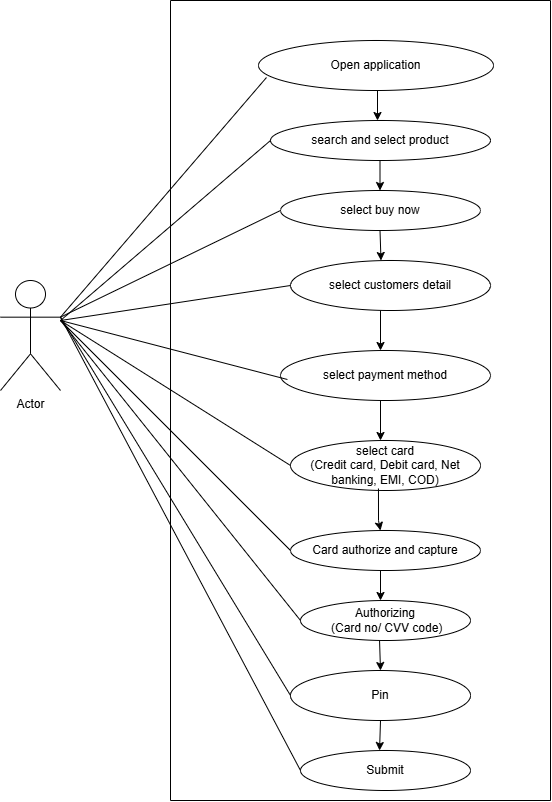
**Cons**

* More risk of sustainability, maintainability and extensibility
* Depends on customer interaction, so if customer is not clear, team can go in wrong direction
* Not suitable for handling complex dependancies
* Minimum documentation generated because of high indivisual dependency
* Technology is transfer to the new team members may be challenging due to lack of documentation usecase.

1. **Draw usecase on online shopping product using COD.**

****

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

****