**Software Testing Assignment**

**Module–1(Fundamental)**

* **What is SDLC**

Ans : **SDLC (Software Development Life Cycle)** is a step-by-step process used to design, develop, test, and deploy software systems. It provides a structured framework that ensures software is developed in a systematic, efficient, and high-quality way.

* **What is software testing?**

Ans**: Software Testing** is the process of evaluating a software application to check whether it meets the specified requirements and works as expected.

**software testing ensures that the software is free of bugs, errors, or defects and delivers the correct output.**

**Main Types of Testing:**

1. **Manual Testing** – Testers execute test cases without using automation tools.
2. **Automation Testing** – Using tools (like Selenium, JUnit, TestNG, etc.) to run test cases automatically.

* **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.

Agile Methods break the product into small incremental builds. These builds are provided in iterations.

* **What is SRS**

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

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

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

* **What is oops**

**Ans:**

Programming is like writing.

If you can write a demonstration, you can make a program.

So, programming is also easy.

But, actually, programming is not so easy, because a real good program is not easily programmed. It needs the programmers’ lots of wisdom, lots of knowledge about programming and lots of experience.

**OOPs (Object-Oriented Programming System)** is a **programming paradigm** based on the concept of **objects**, which can contain data (attributes) and methods (functions) that operate on that data.

It helps in writing **modular, reusable, and maintainable code** by organizing programs around real-world entities like *Student, Employee, Car, etc.*

* **Write Basic Concepts of oops**

**Ans: Class** – A blueprint for creating objects that defines attributes and behaviors.

**Object** – An instance of a class representing real-world entities.

**Encapsulation** – Binding data and methods together while restricting direct access.

**Abstraction** – Hiding implementation details and showing only essential features.

**Inheritance** – Acquiring properties and behaviors of one class into another.

**Polymorphism** – The ability of the same function or operator to behave differently.

* **What is object**

Ans: An object is an instance of a class that represents a real-world entity with state (data) and behavior (methods).

*Example:* A car object has color (data) and start() (behavior).

* **What is class**

Ans**:** A **class** is a blueprint or template used to create objects, containing **fields (attributes)** and **methods (functions)**.

*Example:* class Car { color; start(); stop(); }

* **What is encapsulation**

Ans: Encapsulation is the process of binding data (variables) and methods (functions) into a single unit and restricting direct access to the data using access modifiers.

*Example:* Private variables with getters and setters.

* **What is inheritance**

**Ans : Inheritance** allows a class (child/derived) to acquire the properties and methods of another class (parent/base), promoting **code reusability**.

*Example:* class Dog extends Animal {}

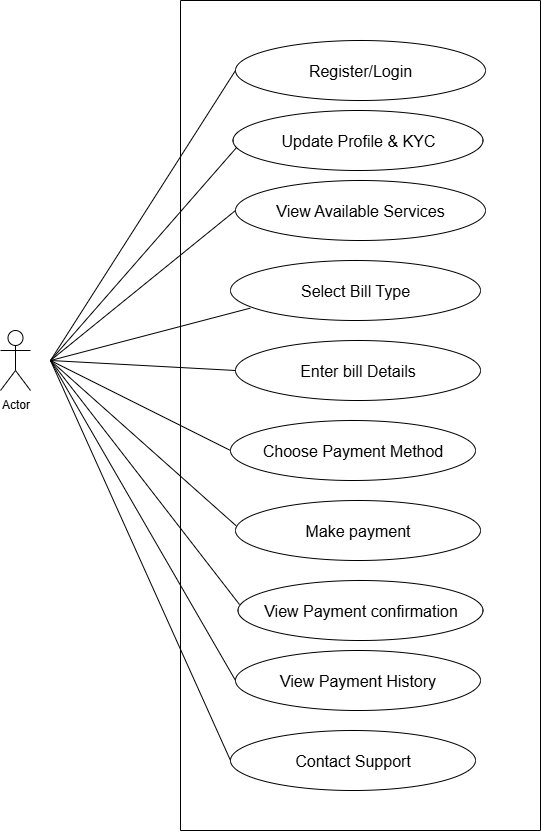
* **What is polymorphism**

Ans**: Polymorphism** means “many forms” – it allows the same method or operator to behave differently depending on the context.

Types:

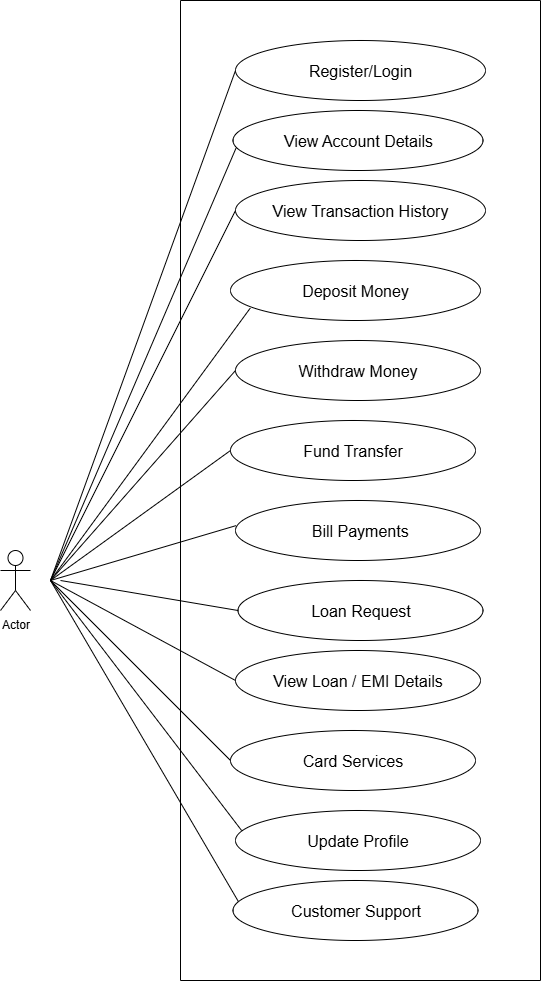
* **Compile-time (Overloading)** – Same method name, different parameters.
* **Run-time (Overriding)** – Subclass redefines a parent method.
* **Draw Usecase on online bill payment system (paytm)**

Ans:

****

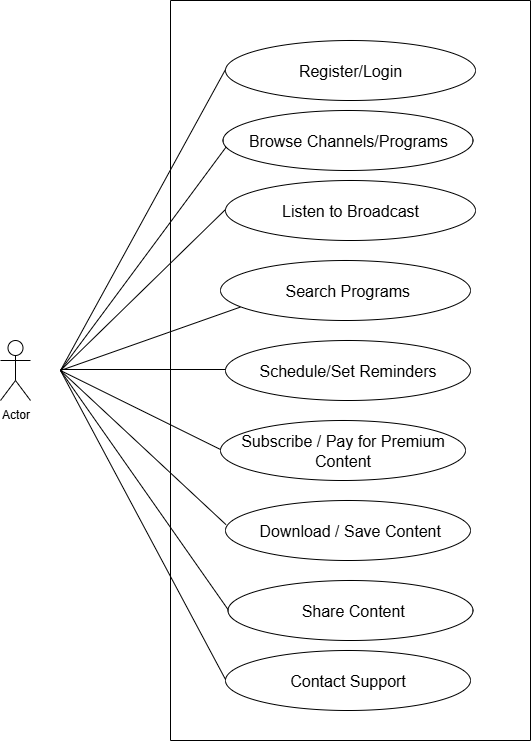
* **Draw Usecase on banking system for customers.**

**Ans:**

****

* **Draw Usecase on Broadcasting System.**

**Ans:**

****

* **Write SDLC phases with basic introduction**

Ans **:** 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.

There are a number of different development models.

Phases of SDLC:

1. Requirement Analysis

In this phase, stakeholders and business analysts gather requirements from clients or end-users.

The goal is to clearly understand what the software should do.

2. Planning

Defines the project scope, resources, cost estimation, and timeline.

Helps create a roadmap for the development process.

3. Design

System architecture and design documents are prepared.

Focuses on how the software will look and function (both high-level design and detailed design).

4. Development (Coding)

Actual coding is done based on the design documents.

Developers write code in the chosen programming language.

5. Testing

The software is tested to find and fix bugs.

Ensures the product works as expected and meets requirements.

6. Deployment

The software is released to the production environment.

Can be a full release or phased rollout depending on the strategy.

7. Maintenance

Continuous support after deployment.

Fixes bugs, updates features, and ensures smooth performance.

* **Explain Phases of the waterfall model**

Ans**:** Phases of the Waterfall Model

1. Requirement Analysis

All the software requirements are collected from the customer.

The scope of the project is clearly documented.

No coding starts until requirements are finalized.

2. System Design

Based on requirements, the overall system architecture and design are created.

Includes High-Level Design (HLD) (overall system structure) and Low-Level Design (LLD) (detailed design of modules).

3. Implementation (Coding)

Developers write code based on the design documents.

Each module is built and integrated to form the complete system.

4. Integration and Testing

All modules are combined and tested as a complete system.

Testing ensures that the software is free from defects and meets user requirements.

5. Deployment

The final product is delivered to the customer or installed in the production environment.

Users start working with the software.

6. Maintenance

After deployment, software may need updates, bug fixes, or enhancements.

This phase ensures the system continues to run smoothly.

* **Write phases of spiral model**

Ans: The Spiral Model combines iterative development (prototyping) and systematic phases of the waterfall model. It is risk-driven and best for large, high-risk projects.

Phases:

1. Planning – Identify objectives, alternatives, and constraints.

2. Risk Analysis – Evaluate risks and prepare strategies to reduce them.

3. Engineering (Development & Testing) – Develop the software in small increments and test it.

4. Evaluation (Customer Feedback) – Get feedback from stakeholders and decide whether to continue, modify, or stop the project.

* **Write agile manifesto principles**

Ans: 1. Customer satisfaction through early and continuous software delivery.

2. Welcome changing requirements, even late in development.

3. Deliver working software frequently (weeks rather than months).

4. Business people and developers must work together daily.

5. Build projects around motivated individuals.

6. Prefer face-to-face communication for effectiveness.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development.

9. Continuous attention to technical excellence and design improves agility.

10. Simplicity—the art of maximizing the amount of work not done—is essential.

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

**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.

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.

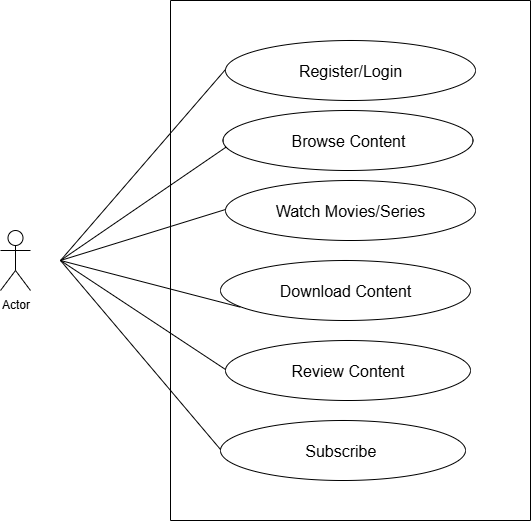
**Pros**

* Is a very realistic approach to software development Promotes teamwork and cross training.
* Functionality can be developed rapidly and demonstrated. Resource requirements are minimum.
* Suitable for fixed or changing requirements Delivers early partial working solutions.
* Good model for environments that change steadily. Minimal rules, documentation easily employed.
* Enables concurrent development and delivery within an overall
* planned context.
* Little or no planning required Easy to manage
* Gives flexibility to developers

**Cons**

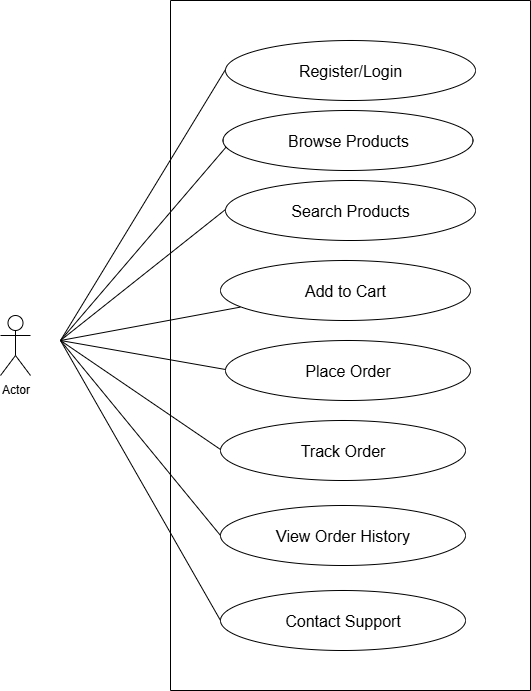
* Not suitable for handling complex dependencies.
* More risk of sustainability, maintainability and extensibility.
* An overall plan, an agile leader and agile PM practice is a must without which it will not work.
* Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
* Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
* 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.
* **Draw usecase on OTT Platform.**

**Ans:**

****

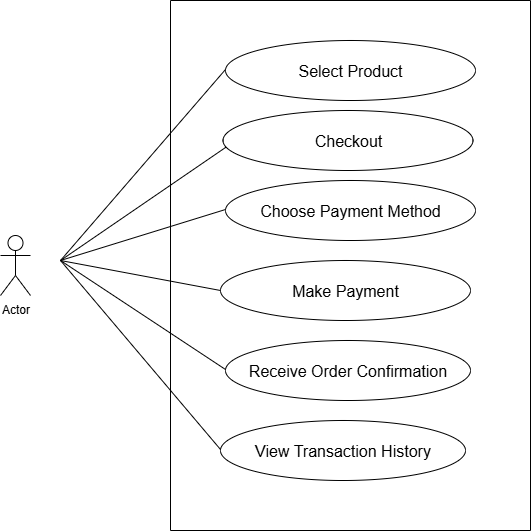
* **Draw usecase on E-commerce application**

**Ans:**

****

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

Ans:

****