1] What is SDLC?

=> SDLC is stands for software development life cycle. Which include various phases to develop software product…

=>requirement gathering, analysis, designing, implementation of codding, testing, deployment, maintenance

2] What is software Testing?

=> Software testing is term to evaluate whether given requirement is satisfied by software or not.

3] What is agile methodology?

=> Agile methodology in testing is an iterative and collaborative approach that integrates testing throughout the software development lifecycle. It emphasizes continuous testing, frequent feedback, and close cooperation between developers and testers to ensure high-quality software delivery. Testing in agile is performed in short cycles (sprints), allowing for quick identification and resolution of issues, adapting to changes, and ensuring the software meets user needs.

4] What is SRS?

=> Software Requirement Specification

=> It is a detailed document that describes the software system to be developed. It serves as a blueprint for both the development team and stakeholders, outlining the functional and non-functional requirements of the system. The SRS provides a comprehensive description of the system’s behavior, features, constraints, and interactions with other systems.

5] What is oops?

=> Object-Oriented Programming (OOP) is a programming paradigm based on the concept of "objects." These objects can contain data, in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). OOP is designed to facilitate the development of complex software systems by promoting the principles of modularity, reusability, and maintainability.

6] Write Basic Concepts of oops?

=>**1. Class:**

A class is a blueprint or template for creating objects.

=>**2. Object:**

An object is an instance of a class. It represents a specific entity with attributes and behavior defined by its class. Each object has its own state and can interact with other objects.

=>**3. Encapsulation:**

Encapsulation is the bundling of data (attributes) and methods (functions) that operate on the data into a single unit or class.

=>**4. Inheritance:**

Inheritance is a mechanism by which one class (the child or subclass) inherits the properties and behaviors (methods) of another class (the parent or superclass). This promotes code reuse and establishes a hierarchical relationship between classes.

=>**5. Polymorphism:**

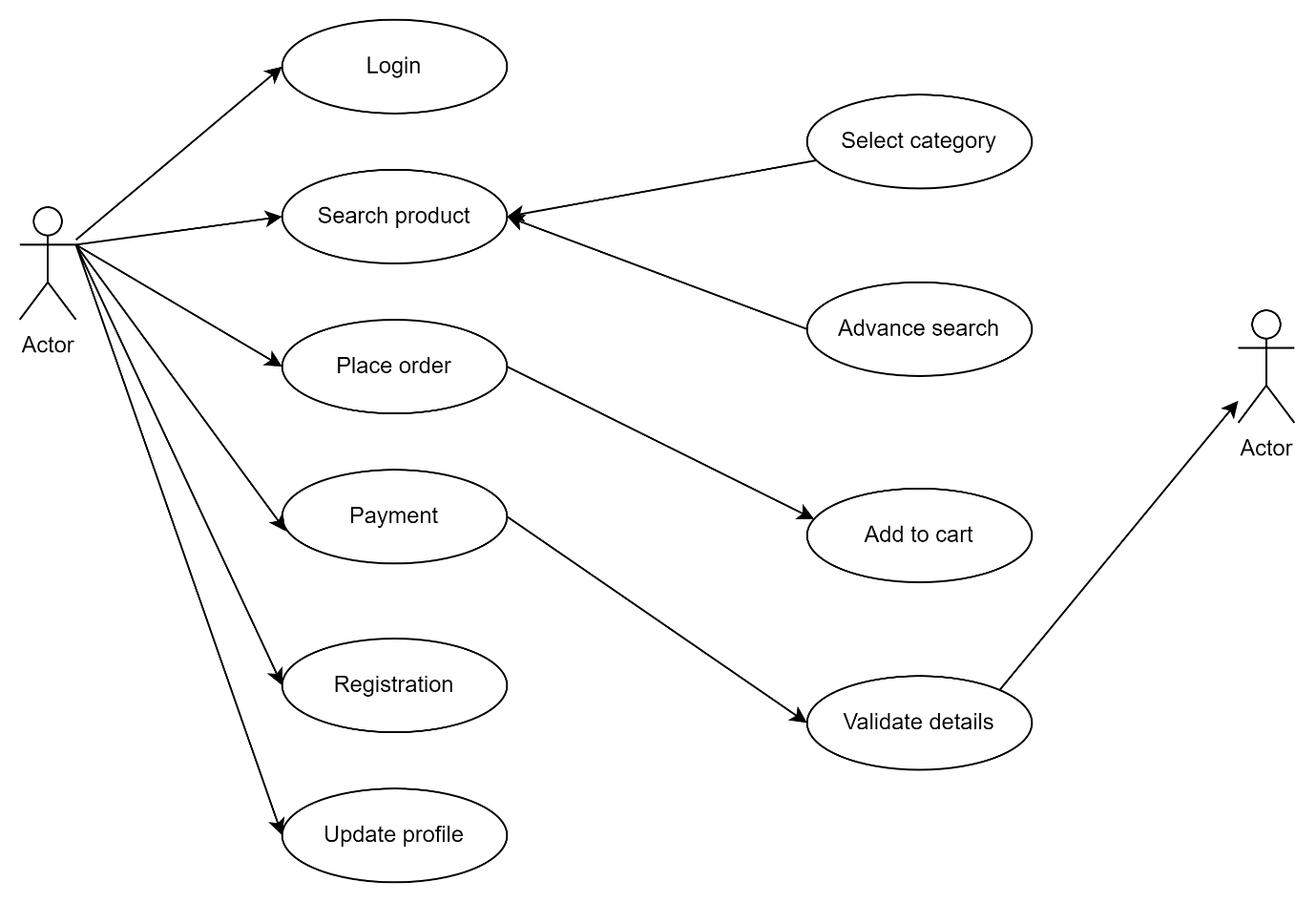
Polymorphism allows objects of different classes to be treated as objects of a common superclass. It enables a single interface to represent different underlying forms (data types). There are two types of polymorphism:

* **Compile-time polymorphism** (method overloading): Multiple methods with the same name but different signatures (parameter lists) in the same class.
* **Run-time polymorphism** (method overriding): A subclass provides a specific implementation of a method that is already defined in its superclass.

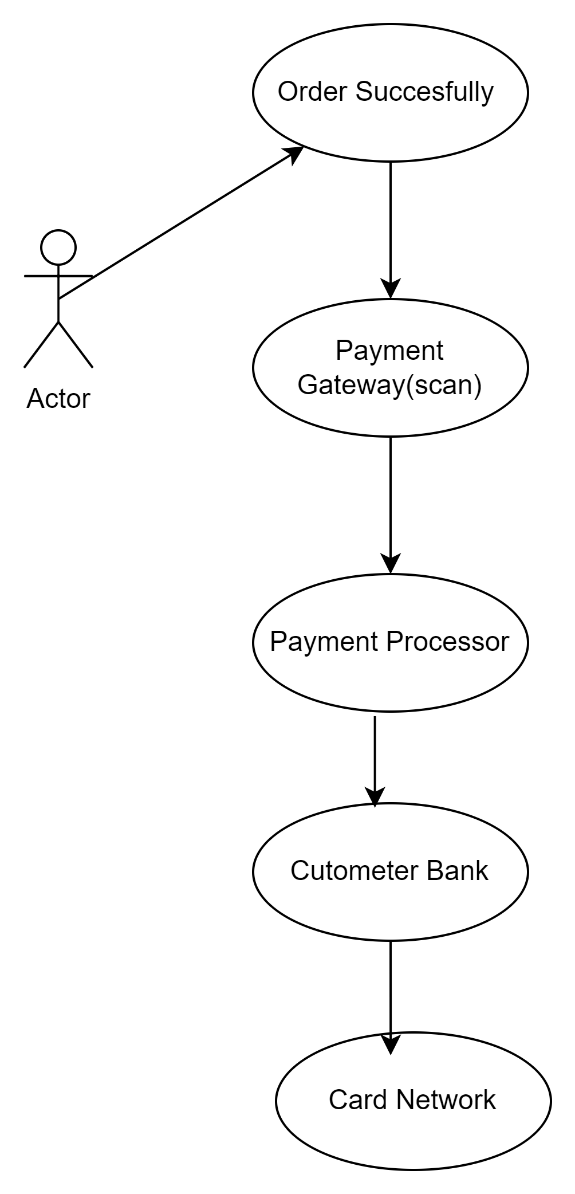
=>**6. Abstraction:**

Abstraction involves hiding complex implementation details and showing only the essential features of an object. This helps to reduce complexity and allows the programmer to focus on interactions at a higher level. Abstraction can be achieved using abstract classes and interfaces.

7] Draw Usecase on Online book shopping.



8] Draw Usecase on online bill payment system (paytm)



9] Write SDLC phases with basic introduction.

=>he Software Development Life Cycle (SDLC) is a structured process used for developing software applications. It ensures that high-quality software is delivered efficiently and effectively. The SDLC consists of several distinct phases, each with specific activities and deliverables. Here’s an overview of the main phases of the SDLC:

**1. Planning**

**Introduction:** The planning phase is the foundation of the software development process. It involves defining the project's scope, identifying objectives, and determining the feasibility of the project. This phase sets the stage for the entire project by outlining the project plan, resource allocation, and risk management strategies.

**2.Requirements Analysis**

**Introduction:** In the requirements analysis phase, the needs and expectations of the end users and stakeholders are gathered and documented. This phase is crucial for understanding what the software is supposed to achieve.

**3.Design**

**Introduction:** The design phase focuses on creating the architecture of the software. This includes defining the overall system architecture, database design, user interfaces, and other critical components.

**4.Implementation (or Coding)**

**Introduction:** During the implementation phase, the actual code for the software is written based on the design documents. This phase transforms the design into a functional software product.

**5.Testing**

**Introduction:** The testing phase is dedicated to verifying and validating the software to ensure it meets the specified requirements and is free of defects. Various testing methods are employed to identify and fix bugs.

**6.Deployment**

**Introduction:** In the deployment phase, the software is delivered to the end-users or made live in the production environment. This phase ensures that the software is properly installed, configured, and operational.

**7. Maintenance**

**Introduction:** The maintenance phase involves ongoing support and maintenance of the software after it has been deployed. This phase addresses any issues, enhancements, and updates needed to keep the software functioning optimally.

10] Explain Phases of the waterfall model.

=>**1. Requirements Gathering and Analysis**

**Introduction:** The first phase involves gathering detailed requirements from stakeholders to understand what the software should do.

**2. System Design**

**Introduction:** The system design phase translates the requirements into a blueprint for building the software. It includes both high-level and detailed design.

**3. Implementation (or Coding)**

**Introduction:** In this phase, developers write the actual code based on the design documents.

**4. Integration and Testing**

**Introduction:** The integration and testing phase involves combining all modules and testing the complete system to ensure it meets the requirements.

**5. Deployment**

**Introduction:** Once the system has passed all tests, it is deployed to the production environment for use by the end-users.

**6. Maintenance**

**Introduction:** The final phase is ongoing and involves maintaining the software to ensure it continues to perform as expected. This includes fixing bugs, making improvements, and updating the system as needed

11] Write phases of spiral model.

=>**1. Planning**

**Introduction:** The planning phase is the initial step in each cycle of the Spiral model, where objectives, alternatives, and constraints are identified.

**2. Risk Analysis**

**Introduction:** In the risk analysis phase, potential risks are identified and analyzed to mitigate their impact on the project.

**3. Engineering**

**Introduction:** The engineering phase involves the actual development and implementation of the chosen solution. It includes design, coding, and testing.

**4. Evaluation**

**Introduction:** The evaluation phase focuses on reviewing the outcomes of the current cycle and planning the next iteration.

**Repetition of Phases**

The Spiral model's strength lies in its iterative nature. Each cycle (or spiral) passes through these four phases, progressively refining and expanding the project. The repetition continues until the project is complete, with each cycle producing a more refined and tested version of the software.

12] Write agile manifesto principles.

=> 1. Customer Satisfaction

2. Welcome Changing Requirements

3. Deliver Working Software Frequently

4. Business People and Developers Must Work Together

5. Build Projects Around Motivated Individuals

6. Face-to-Face Communication

7. Working Software is the Primary Measure of Progress

8. Maintain a Sustainable Pace

**9. Continuous Attention to Technical Excellence**

12. Regular Reflection and Adaptation

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

=>**Pros:**

1. **Flexibility and Adaptability:**
   * Agile can quickly adapt to changing requirements and market conditions, making it ideal for projects with evolving needs.
2. **Customer Satisfaction:**
   * Frequent delivery of working software and regular feedback ensure that the product aligns with customer needs and expectations.
3. **Early and Continuous Delivery:**
   * Working software is delivered early and often, allowing for regular validation and feedback, which helps in identifying and addressing issues early.
4. **Improved Collaboration:**
   * Agile promotes collaboration between cross-functional teams and stakeholders, leading to better communication and shared understanding.
5. **Enhanced Quality:**
   * Continuous testing and integration help in identifying defects early and improving overall product quality.
6. **Increased Motivation and Engagement:**
   * Empowering teams to self-organize and make decisions increases motivation and engagement among team members.

**Cons:**

1. **Scope Creep:**
   * The flexibility of Agile can lead to scope creep if not managed carefully. Continuous changes and additions may impact project timelines and resources.
2. **Requires Close Collaboration:**
   * Agile depends heavily on effective communication and collaboration. If team members or stakeholders are not fully engaged, it can hinder progress.
3. **Less Predictability:**
   * The iterative nature of Agile can make it challenging to predict final project costs, timelines, and deliverables.
4. **Potential for Incomplete Documentation:**
   * Agile focuses on working software over comprehensive documentation, which can lead to insufficient documentation and challenges in maintaining the system.
5. **Challenges with Large Teams:**
   * Scaling Agile practices to large teams or complex projects can be challenging and may require additional frameworks or coordination.
6. **Requires Experienced Team Members:**
   * Agile methodologies work best with experienced and self-motivated team members who can adapt to the dynamic nature of the process.