# SE Assignment

# Module-1(SDLC)

• What is software? What is software engineering?

Software is a program or set of programs containing instructions that provide desired functionality. And Engineering is the process of designing and building something that serves a particular purpose and finds a cost-effective solution to problems.

Software engineering is the process of designing, developing, testing, and maintaining software. It is a systematic and disciplined approach to software development that aims to create high-quality, reliable, and maintainable software. Software engineering includes a variety of techniques, tools, and methodologies, including requirements analysis, design, testing, and maintenance.

Software engineering is the process of analyzing user needs and designing, constructing, and testing end-user applications that will satisfy these needs through the use of software programming languages. It is the application of engineering principles to software development.

• Explain types of software?

**Application Software :**

It is a type of software application that helps in the automation of the task based on the Users Input. It can perform single or multiple tasks at the same period of time. There are the different application which helps us in our daily life to process our instructions based on certain rules and regulations. Application Software helps in providing a graphical user interface to the user to operate the computer for different functionality.

The user may use the computer for browsing the internet, accessing to email service, attending meetings, and playing games. Different high-level languages are used to build application software.

**Types of Application Software**

* Application software
* System software
* Driver software
* Middleware
* Programming software
* **Application Software**

The most common type of software, application software is a computer software package that performs a specific function for a user, or in some cases, for another application. An application can be self-contained, or it can be a group of programs that run the application for the user.

* Examples of Modern Applications :

include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.

* **System Software**

These software programs are designed to run a computer's application programs and hardware. System software coordinates the activities and functions of the hardware and software. It controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in. The OS is the best example of system software; it manages all the other computer programs.

* Example: Notepad, Calculator etc.
* **Driver Software**

Driver Software Also known as device drivers, this software is often considered a type of system software. Device drivers control the devices and peripherals connected to a computer, enabling them to perform their specific tasks. Every device that is connected to a computer needs at least one device driver to function.

* Example: Audio Driver, Video Driver etc...
* **Middleware**

The term middleware describes software that mediates between application and system software or between two different kinds of application software.

* For example :

middleware enables Microsoft Windows to talk to Excel and Word. It is also used to send a remote work request from an application in a computer that has one kind of OS, to an application in a computer with a different OS. It also enables newer applications to work with legacy ones.

* **Programming Software**

Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and debug other software programs.

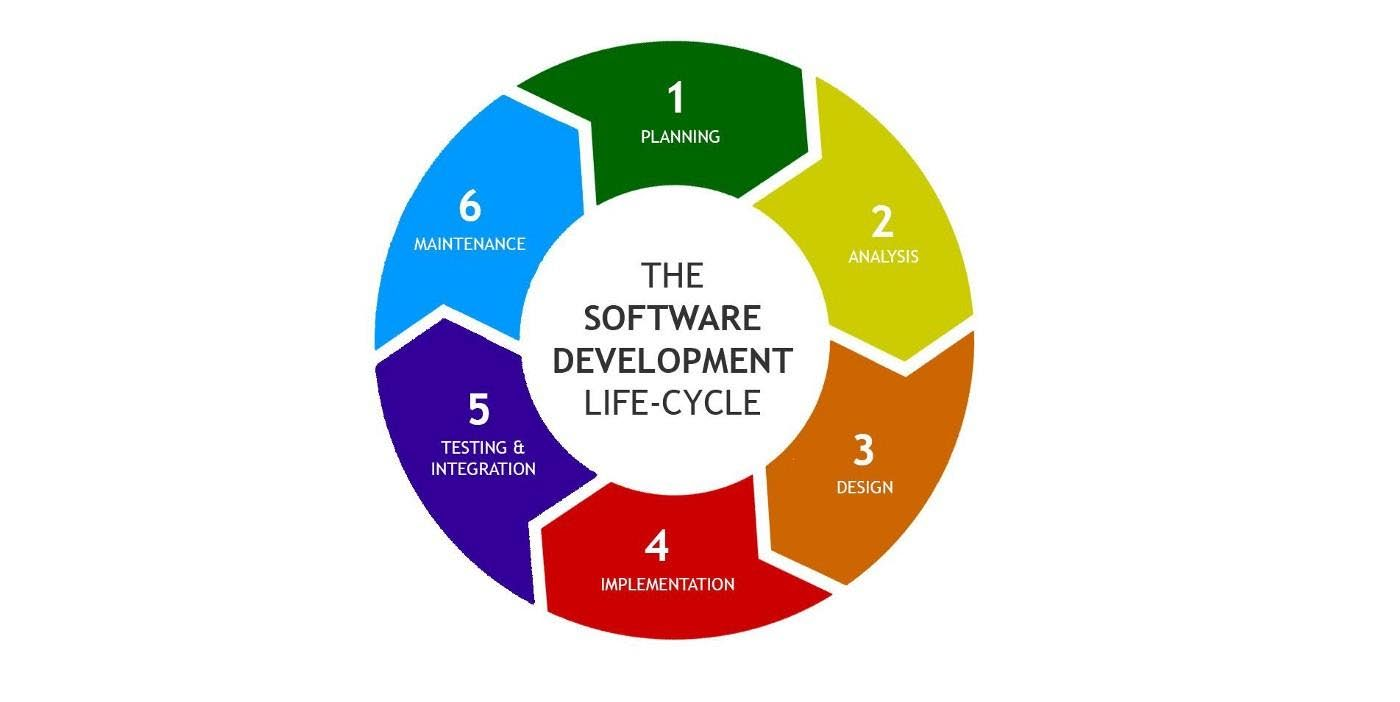
* Examples : Turbo c, Eclipse, Sublime etc..

• What is SDLC? Explain each phase of SDLC ?

The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. in detail, the SDLC methodology focuses on the following phases of

* **software development:**

1. Requirement Gathering
2. Analysis
3. Designing
4. Implementation
5. Testing
6. Maintenance



**1. Requirements gathering and analysis:** This phase involves gathering information about the software requirements from stakeholders, such as customers, end-users, and business analysts.

**2. Design:** In this phase, the software design is created, which includes the overall architecture of the software, data structures, and interfaces. It has two steps:

**High-level design (HLD):** It gives the architecture of software products.

**Low-level design (LLD):** It describes how each and every feature in the product should work and every component.

**3. Implementation or coding:** The design is then implemented in code, usually in several iterations, and this phase is also called as Development. things you need to know about this phase:

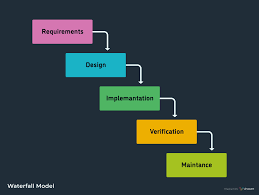
This is the longest phase in SDLC model.

This phase consists of Front end + Middleware + Back-end.

**4. Testing:** The software is thoroughly tested to ensure that it meets the requirements and works correctly.

**5. Deployment:** After successful testing, The software is deployed to a production environment and made available to end-users.

**6. Maintenance:** This phase includes ongoing support, bug fixes, and updates to the software.



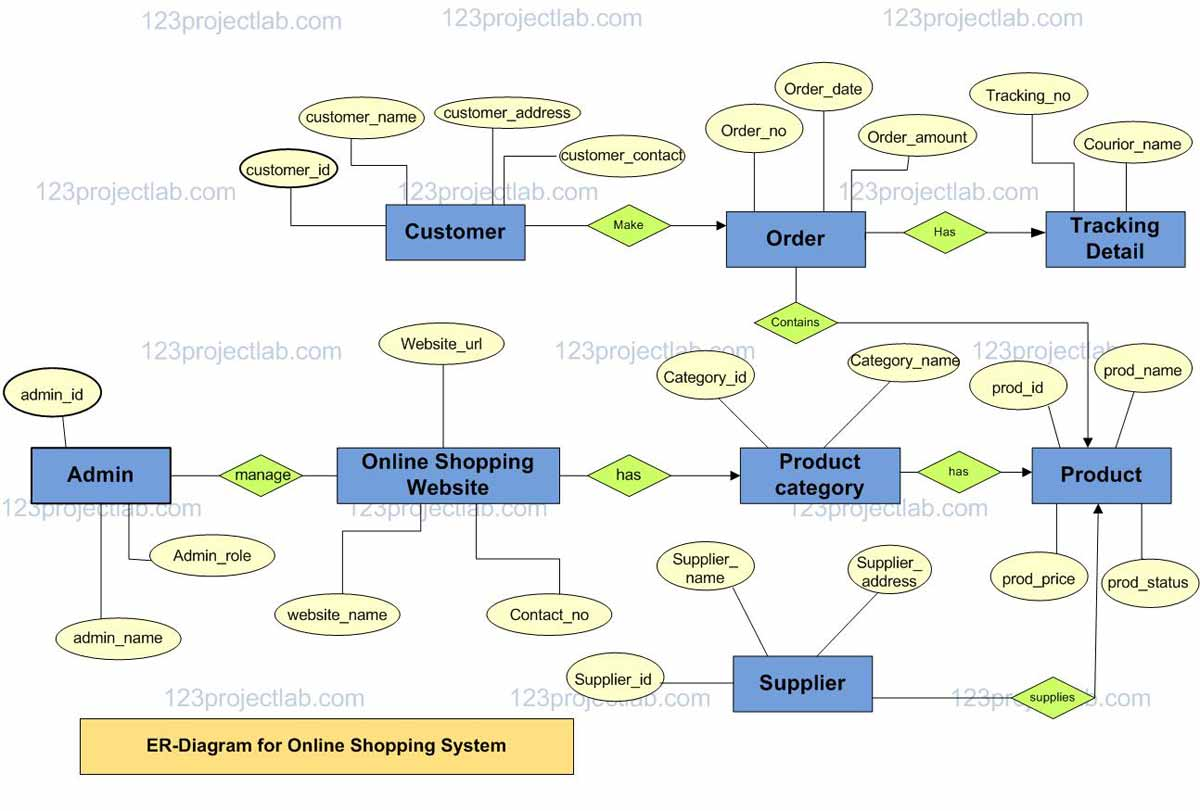
• What is DFD? Create a DFD diagram on Flipkart

DFD stands for Data Flow Diagram, which is a graphical representation of how data flows within a system. It is commonly used in software engineering and system analysis to understand and document the flow of information and processes.

Creating a DFD diagram for Flipkart would require a detailed understanding of its internal systems and processes, which are not publicly available. However, I can provide you with a general example of a high-level DFD diagram for an e-commerce platform like Flipkart. Please note that this example is a simplified representation and may not accurately reflect the actual systems and processes of Flipkart.

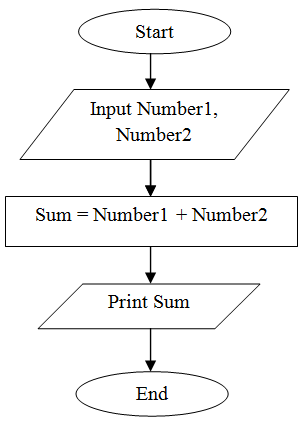
In this example, the diagram shows various components involved in the e-commerce process. The customer interacts with the system by searching and browsing for products, adding them to the shopping cart, and making a payment through a payment gateway. The order system processes the order, confirms it, and manages the delivery. The inventory management system keeps track of available products, and the customer can provide feedback on their experience.

Again, please note that this is a simplified example, and the actual DFD diagram for Flipkart would be more complex and detailed, involving numerous subsystems and processes.



• What is Flow chart? Create a flowchart to make addition of two numbers

A flowchart is a graphical representation of a process or algorithm that uses different symbols to depict the various steps and decisions involved. It allows for a visual representation of the flow of actions or information within a system.



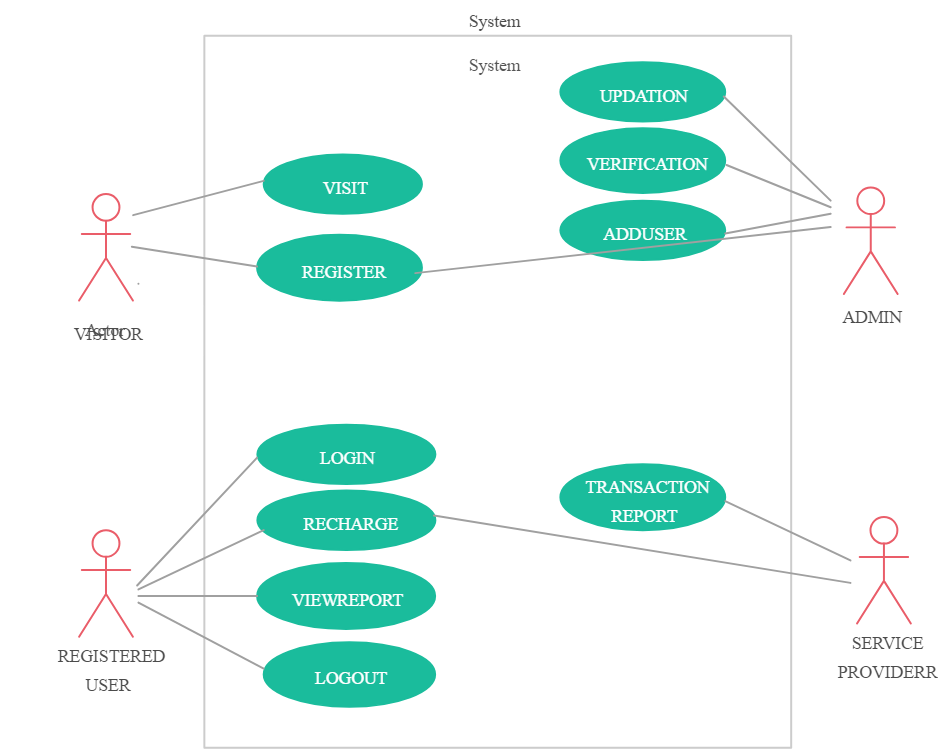
In this flowchart:

* "Start" represents the beginning of the process.
* "Input Number 1" and "Input Number 2" are the steps where you input the two numbers you want to add.
* "Add Number 1 and Number 2" is the step where the addition of the two numbers is performed.
* "Display the Result" is the step where the sum of the two numbers is shown.
* "Stop" represents the end of the process.

• What is Use case Diagram? Create a use-case on bill payment on paytm.

A use case diagram is a visual representation of the interactions between system users (actors) and the system itself. It illustrates the functionalities or features of a system from the user's perspective. Use case diagrams are commonly used in software development to capture and communicate the intended behaviour of a system.

Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process.



In this use case diagram:

* "User" represents the actor (system user) who interacts with the Paytm system.
* "Paytm" is the system itself, which provides various functionalities, including bill payment.
* "Payment Options" is a use case that allows the user to select the desired payment method.
* "Bill Information" is a use case that provides details about the bill to be paid.
* "Bill Payment" is the main use case where the user initiates the payment process for a bill.