

Software Engineering Assignment

MODULE: 1 (SDLC)

1 What is software? What is software engineering?

Ans. 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.

• **Software engineering:**

Software engineering is defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements.

-Let's look at the various definitions of software engineering:

- IEEE, in its standard 610.12-1990, defines software engineering as the application of a systematic, disciplined, which is a computable approach for the development, operation, and maintenance of software.
- Fritz Bauer defined it as 'the establishment and used standard engineering principles. It helps you to obtain, economically, software which is reliable and works efficiently on the real machines'.
- Boehm defines software engineering, which involves, 'the practical application of scientific knowledge to the creative design and building of computer programs. It also includes associated documentation needed for developing, operating, and maintaining them.'

2. Explain types of software.

Ans.

❖ Types of Application Software

1. Application software
2. System software
3. Driver software
4. Middleware
5. Programming Software

1. 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.

Example: Microsoft Office, Paint, Power point etc.

2. 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.
- Other examples of system software include the firmware, computer language translators and system utilities.

Example: Notepad, Calculator etc.

3. 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.
- Examples include software that comes with any nonstandard hardware, including special game controllers, as well as the software that enables standard hardware, such as USB storage devices, keyboards, headphones and printers.

Example: Audio Driver, Video Driver etc.

4. 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.

Example: database middleware, application server middleware

5. 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 of programming software include assemblers, compilers, debuggers and interpreters.

Examples : Turbo c ,Eclipse, Sublime etc.

3. What is SDLC? Explain each phase of SDLC.

Ans: - The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software.

- This process is about creating a detailed plan to guide the development of the product and then breaking down the development process into smaller modules that can be assigned, completed, and measured to make the whole thing more manageable.

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

2. Analysis: The first phase of the SDLC is the project planning stage where you are gathering business requirements from your client or stakeholders. This phase is when you evaluate the feasibility of creating the product, revenue potential, the cost of production, the needs of the end-users, etc.

3. Designing: 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.

4. Implementation: 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.

In front-end: Development of coding is done even SEO settings are done.

In Middleware: They connect both the front end and back end.

In the back-end: A database is created.

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

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

There are different methodologies that organizations can use to implement the SDLC, such as Waterfall, Agile, Scrum, V-Model and DevOps.

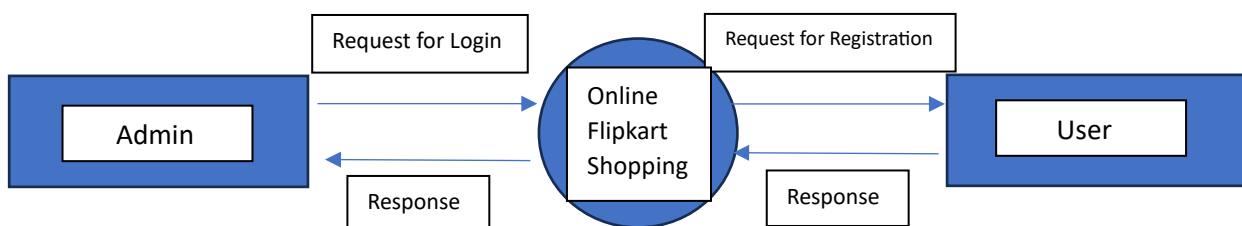
4. What is DFD? Create a DFD diagram on Flipkart.

Ans.-A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

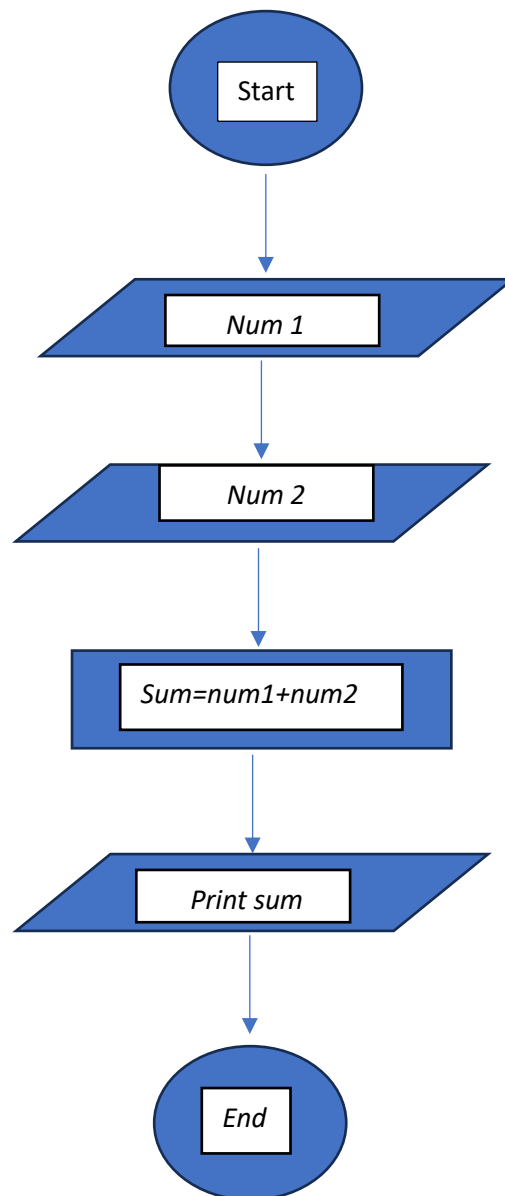
DFD Diagram on Flipkart:



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

Ans: - A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

Flowchart: -

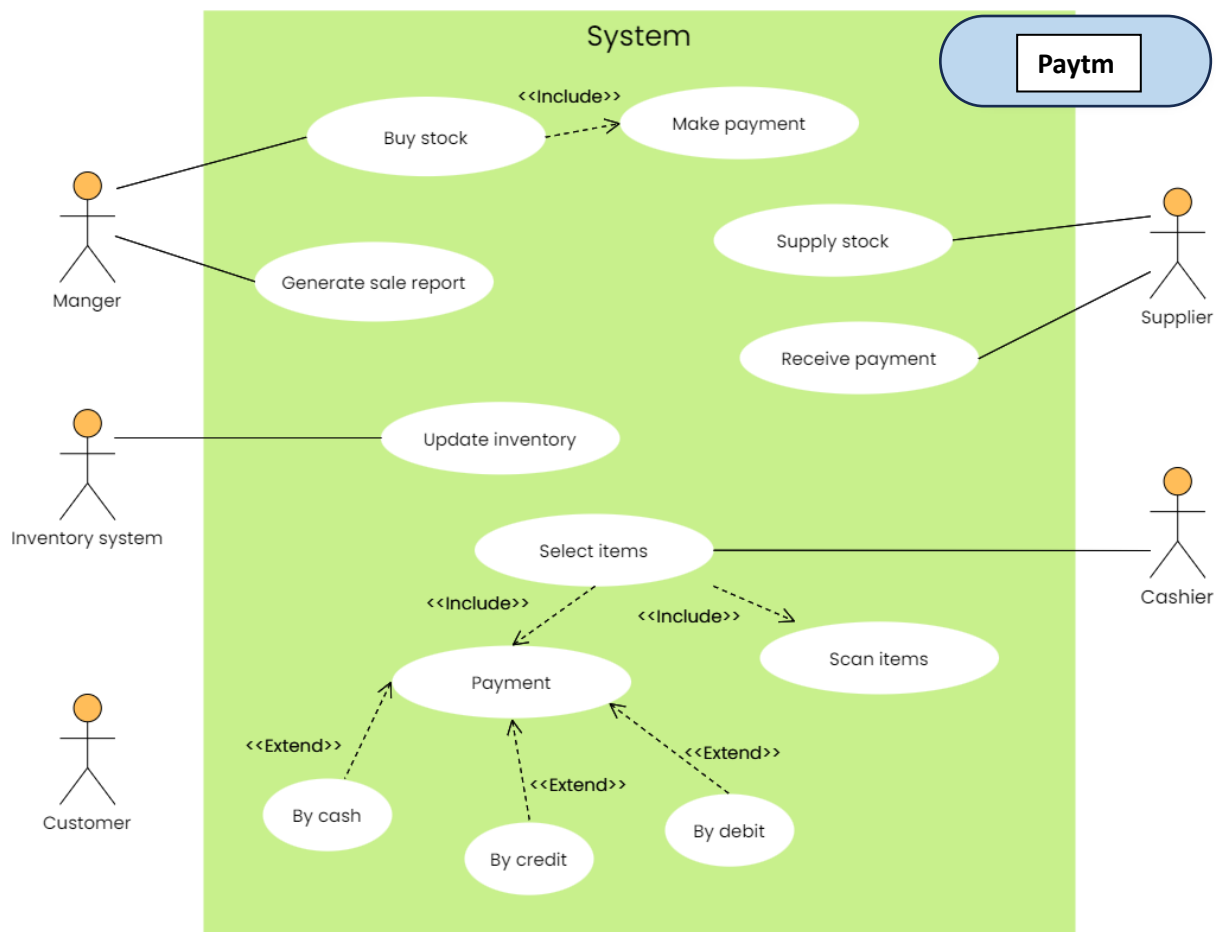


6. What is Use case Diagram? Create a use-case on bill payment on Paytm.

Ans: - A use case diagram is used to represent the dynamic behaviour of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and

elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.



The End
