Software Engineering

1) 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.

2) Explain types of software?

Among the various categories of software, the most common types include the following:

- **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. In addition, it controls

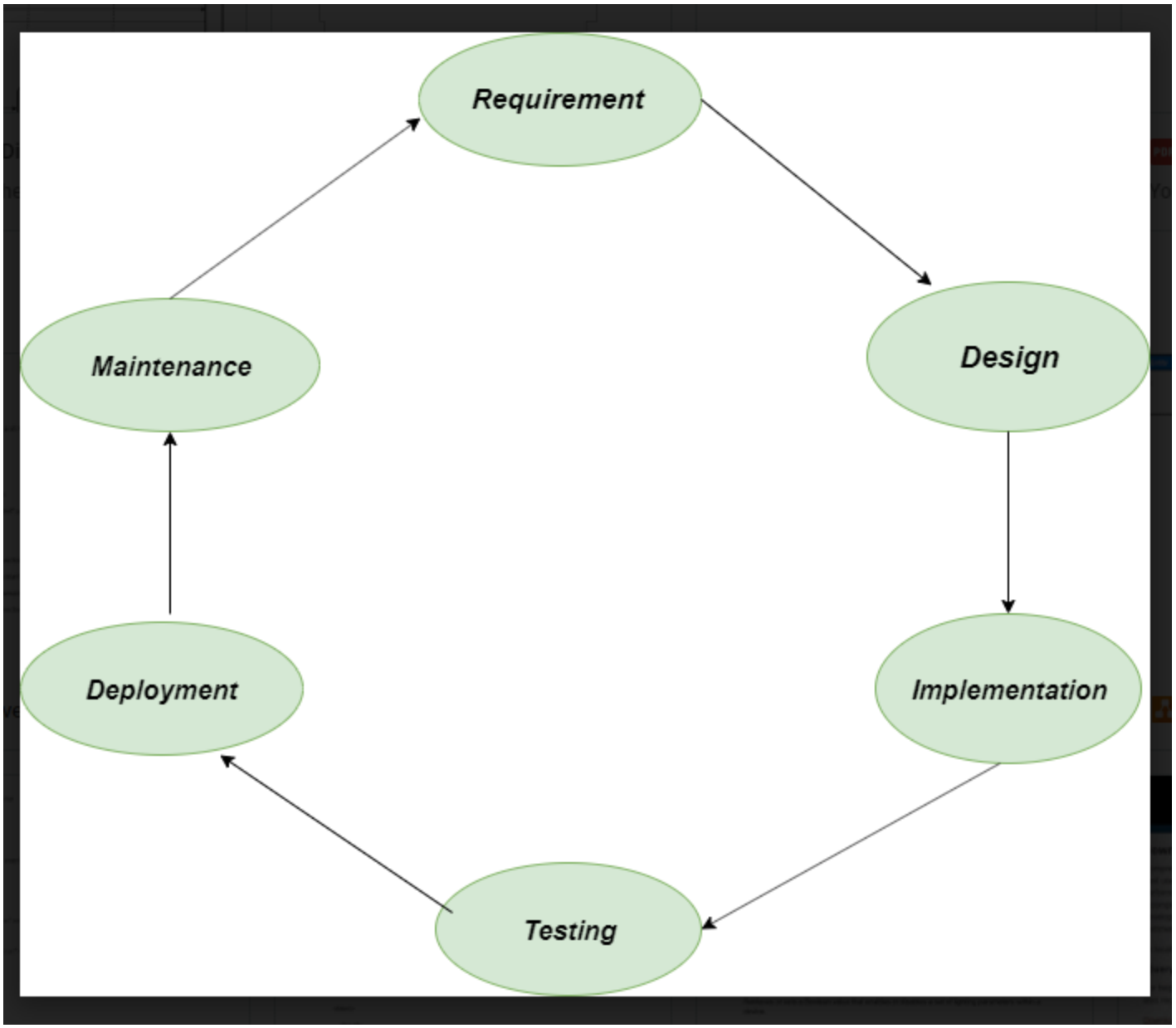
the operations of the computer hardware and provides an environment or platform for all the other types of software to work in. The OS is the best example of system software; it manages all the other computer programs. Other examples of system software include the firmware, computer language translators and system utilities.

- **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.
- **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 of programming software include assemblers, compilers, debuggers and interpreters.

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

SDLC (Software Development Life Cycle) is used in Every Software Development Company because it is the root of the Development Cycle, if that model would not exist in the world, firstly no software can build secondly if any how it would be made, it's not going to succeed it has no use, because of no maintenance, but Luckily SDLC model exist in Tech world **But why we need it Actually!**
There are several reasons why organizations use the Software Development Life Cycle (SDLC) when developing software applications:

To provide a **structured and organized approach** to software development: The SDLC provides a framework for managing the software development process, which helps to ensure that all necessary steps are taken and that the final product meets the requirements.
1.To ensure that the software is of high quality: The SDLC includes testing and quality assurance phases, which help to ensure that the software is free of bugs and that it meets the requirements.
2.To manage risks and costs: The SDLC helps organizations to identify and manage risks early in the development process, which can help to reduce costs and minimize the impact of any issues that do arise.
3.To improve communication and collaboration: The SDLC helps to ensure that all stakeholders, including customers, end-users, and developers, are involved in the development process and that their needs are taken into account.
4.To improve efficiency and productivity: The SDLC helps organizations to optimize the use of resources and to streamline the development process, which can improve efficiency and productivity.
5.To increase the likelihood of a successful project outcome: Following a well-defined SDLC process can greatly increase the chances of success of the project, as the process guides the team towards the goal in a systematic and efficient way.
Overall, the **SDLC is a valuable tool** for organizations to use when developing software applications, as it helps **to ensure that the final product is of high quality, meets the requirements**, and is delivered on time and within budget.
The SDLC typically includes the following phases:



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.
- **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.

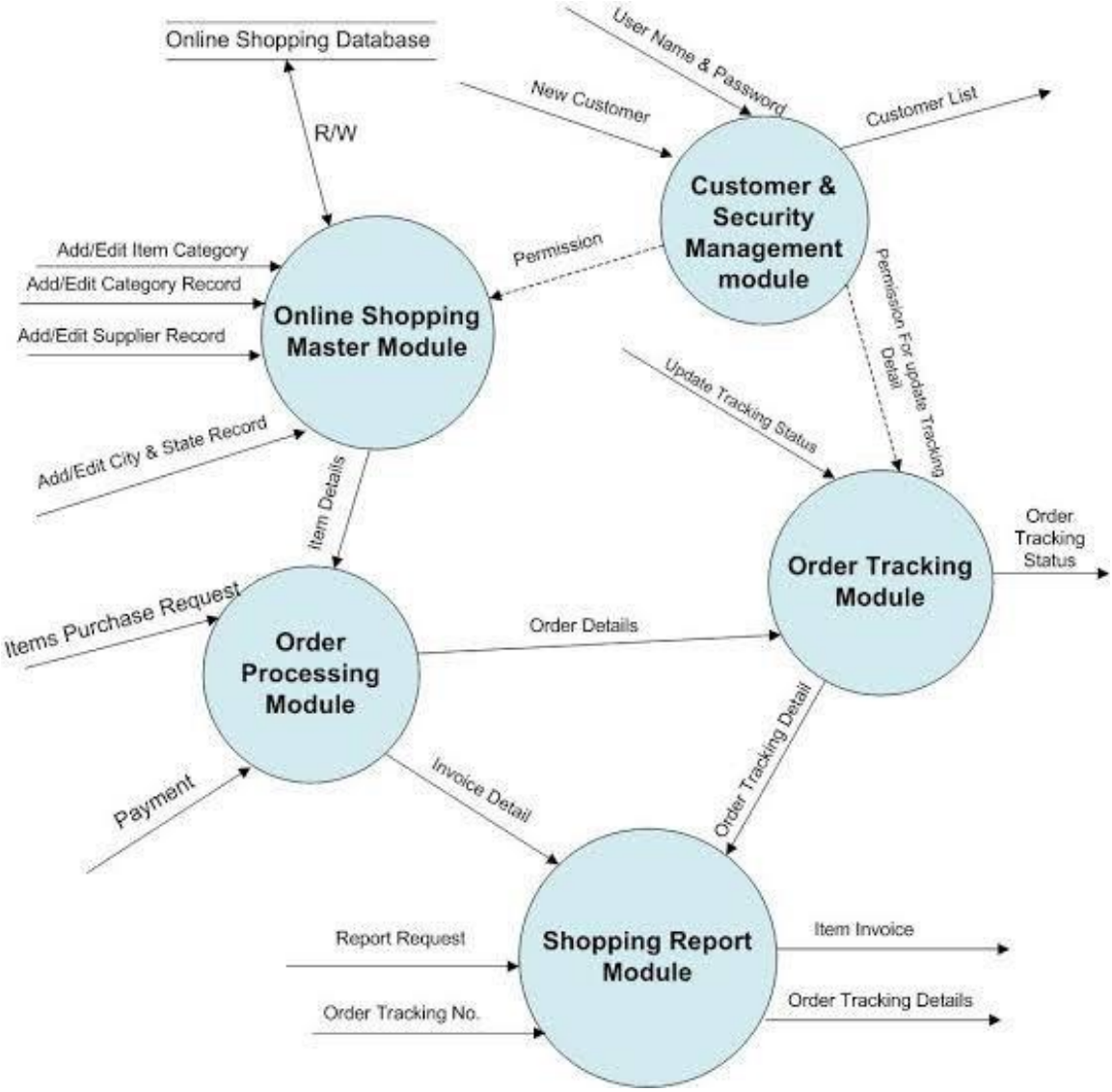
4. Testing: The software is thoroughly tested to ensure that it meets the requirements and works correctly.
5. Deployment: After successful testing, the software is deployed to a production environment and made available to end-users.
6. Maintenance: This phase includes ongoing support, bug fixes, and updates to the software.
There are **different methodologies** that organizations can use to implement the SDLC, such as **Waterfall, Agile, Scrum, V-Model** and **DevOps.**

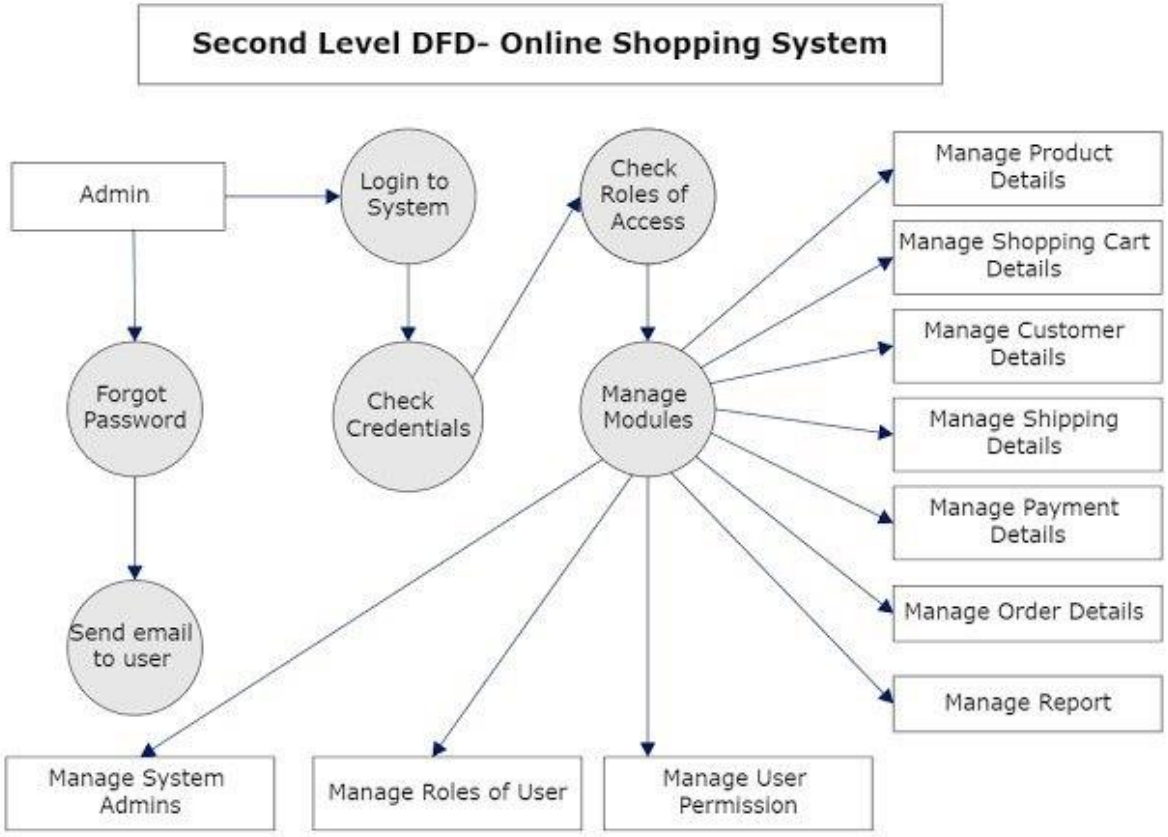
4) What is DFD? Create a DFD diagram on flipkart?

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.


Online Shopping Database
R/W
User Name & Password
New Customer
Customer List
Customer & Security Management module
Permission
Permission For update Tracking Detail
Add/Edit Item Category
Add/Edit Category Record
Add/Edit Supplier Record
Online Shopping Master Module
Add/Edit City & State Record
Item Details
Update Tracking Status
Order Tracking Module
Order Tracking Status
Items Purchase Request
Order Processing Module
Order Details
Payment
Invoice Detail
Order Tracking Detail
Report Request
Order Tracking No.
Shopping Report Module
Item Invoice
Order Tracking Details


Second Level DFD- Online Shopping System
Admin
Login to System
Check Roles of Access
Manage Product Details
Manage Shopping Cart Details
Manage Customer Details
Manage Shipping Details
Manage Payment Details
Manage Order Details
Manage Report
Forgot Password
Check Credentials
Manage Modules
Send email to user
Manage System Admins
Manage Roles of User
Manage User Permission

5) What is flow chart? Create a flowchart to make addition of two numbers?

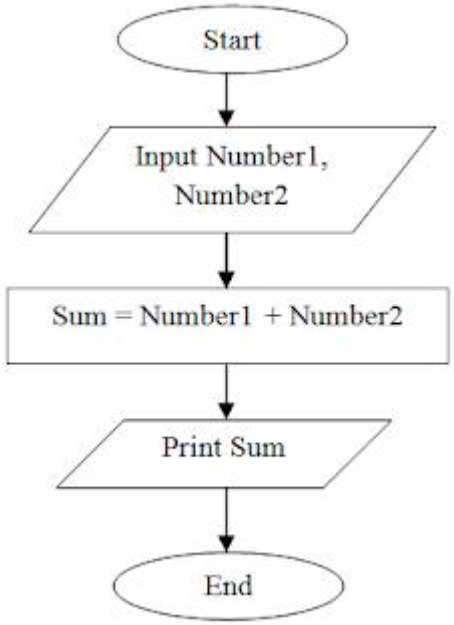
A flowchart is a diagrammatic representation of steps that we have taken to solve the problem.

Below is the flowchart to find the addition of two numbers.

Explanation:

Creating a Flowchart

1. A flowchart begins and ends with an oval symbol that is called the start and end symbol.
2. Input and output are marked by drawing a parallelogram.
3. Any process like addition or logical operation is shown by a rectangular box.
4. If you are making a decision, then show this by a diamond box.
5. All the symbols are connected via the arrow lines.
6. Memorize all these conventions and logically connect all the steps. Your flowchart is ready.



6) What is use case diagram? Create a use-case on bill payment on paytm?

Use Case Diagram

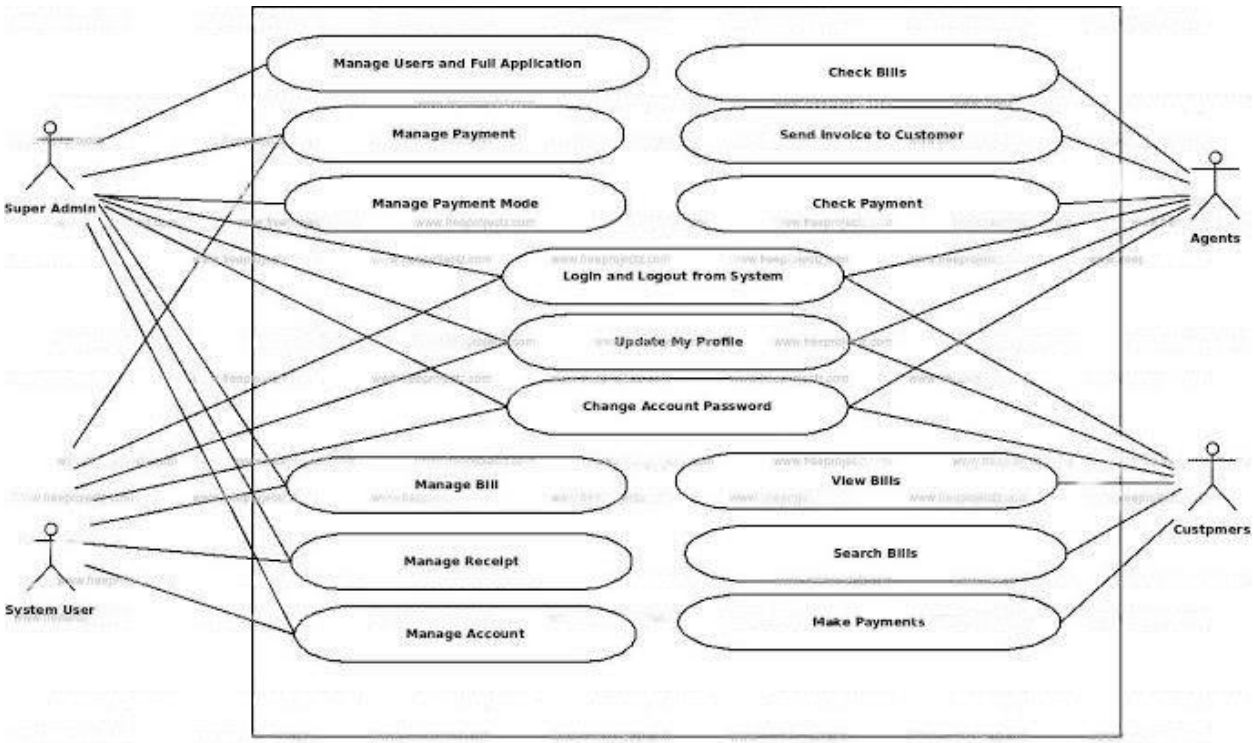
A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality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

Purpose of Use Case Diagrams

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.

Following are the purposes of a use case diagram given below:

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.



Usecase Diagram of Payment Management System