**ASSIGNMENT – 1**

**Q1. What is software? What is software engineering?**

* **Software** is a collection of instructions, data, or computer programs that are used to run machines and perform particular activities.
* In a computer system, the software is basically a set of instructions or commands that tell a computer what to do and how to do.
* **Software engineering** is the process of designing, developing, testing, and maintaining software.
* Software engineering is an art of developing quality software on the time within budget.
* **principles of Software Engineering:**

1) Efficiency

2) Reliability

3) Usability

4) Maintainability

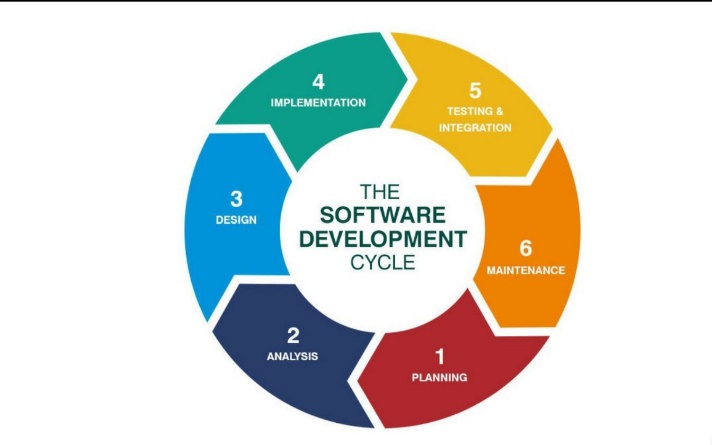
5) Portability

**Q2) Explain types of software**

* **System Software:** This type of software directly operates the computer hardware and provides the basic functionality to the users as well as to the other software to operate smoothly. It includes the operating system and device drivers.
* **Application Software:** These are programs designed to perform specific tasks for users. Examples include word processors, spreadsheets, and web browsers.
* **Operating System:** This is the main program of a computer system that manages all the resources such as computer memory, CPU, printer, hard disk, etc., and provides an interface to the user.
* **Device Driver:** A device driver is a program or software that controls a device and helps that device to perform its functions.
* **Language Processor:** This type of software converts programs written in high-level programming languages into sets of instructions that are easily readable by machines

**Q3) What is SDLC? Explain each phase of SDLC.**

* SDLC – Software Development Life Cycle.
* It's a structured process used by software developers to design, develop, test, and deploy software applications. The SDLC involves a series of phases that guide the development process from the initial planning stages to the final deployment and maintenance of the software. These phases typically include:
* **1. Requirements gathering:** Defining the scope, objectives, and requirements of the project. In this phase, all the relevant requirements for the software are gathered from the client or end-users. This is done through various data gathering techniques like brainstorming, interviews, surveys, etc
* **2. Analysis:** In the Analysis phase, the requirements gathered in the Requirement Gathering phase are studied and analysed to make them clear and well-documented. gathering and analysing requirements from stakeholders to understand what the software needs to accomplish.
* **3. Design:** Creating a blueprint for how the software will be structured and how its components will interact. In the Design phase, the software solution is designed based on the requirements defined in the Software Requirement Specification (SRS) document from the Analysis phase. The main goal of this phase is to transform the requirements into a structure that the developers can use to create the software. This involves designing the software architecture, database, user interfaces, and system interfaces.
* **4. Implementation:** Writing the code and building the software according to the design specifications. This phase is also known as the Coding or Development phase. The main goal of this phase is to translate the design of the software into code in a programming language. The developers must follow the coding guidelines, use programming tools and follow the best practices for developing high quality code.
* **5. Testing:** The testing phase is a critical part of the SDLC. It’s where the developed software is tested to ensure it functions as expected and meets the requirements specified in the initial stages of the project. Verifying that the software functions correctly and meets the specified requirements through various testing methods.
* **6. Maintenance:** Providing ongoing support, updates, and enhancements to ensure the software remains functional and efficient throughout its lifecycle. This phase begins after deployment and includes post-deployment support, issue tracking, updates and enhancements, performance optimization, preventive and corrective maintenance, and documentation. The goal is to ensure the software continues to deliver value.



**Q4) What is DFD? Create a DFD diagram on Flipkart**

* DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. It is a graphical tool, useful for communicating with users, managers and other personnel. it is useful for analysing existing as well as proposed system.

**-> In DFD there are mainly three level**

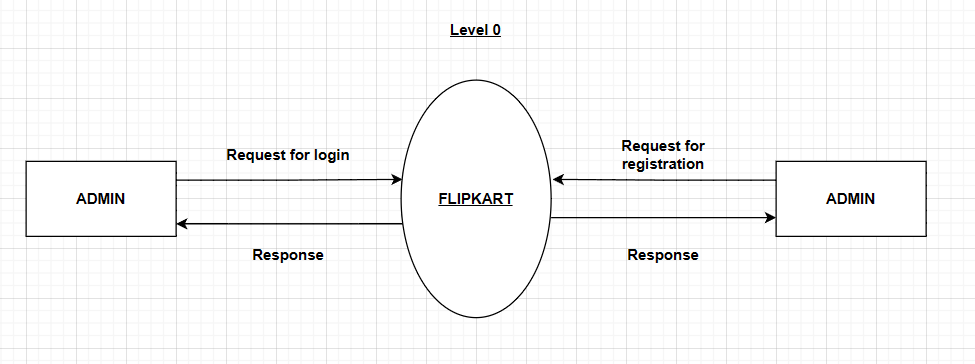
**1) 0 level**

**2) 1 level**

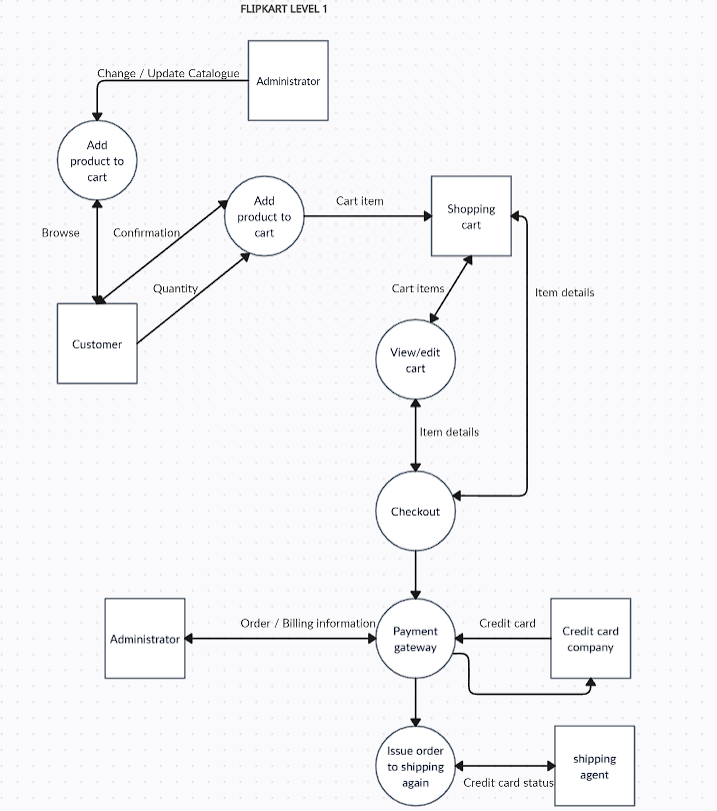
**3) 2 level**

**4) 3 level**

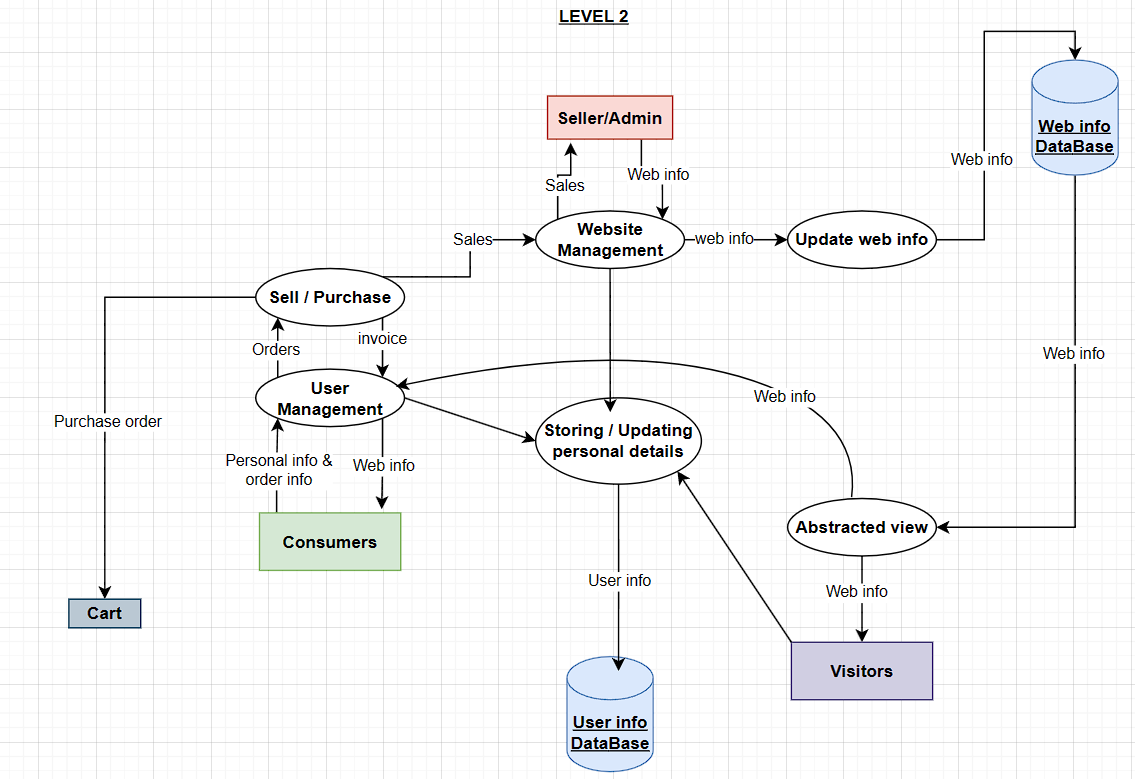
* **Level 0:** is also called a Context Diagram. It’s a basic overview of the whole system or process being analysed or modelled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

****

* **Level 1:** Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its subprocesses



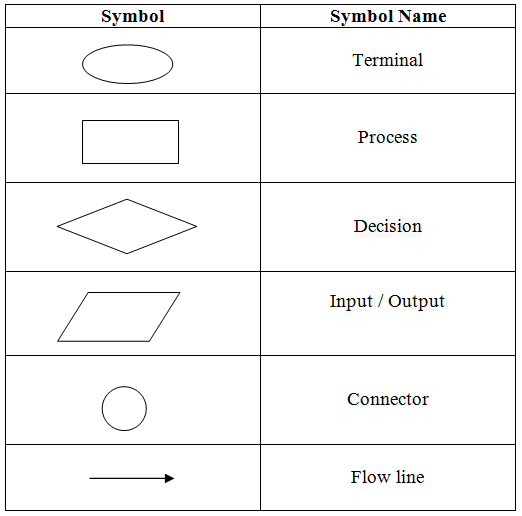
* **Level 2:** then goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system’s functioning.



* **Level 3:** Progression to Levels 3 beyond is possible, but going beyond Level 3 is uncommon. Doing so can create complexity that makes it difficult to communicate, compare or model effectively.

**Q5) What is Flow chart? Create a flowchart to make addition of two numbers**

* A flowchart is a graphical representation of an algorithm or a step-by-step approach to solving a task. It uses symbols connected by arrows to indicate the flow of information and processing. Flowcharts are used in various fields for analysing, designing, documenting, or managing a process or program. Flowchart is created on the base of algorithm



**Algorithm of sum of 2 numbers:**

**Step 1: Start**

**Step 2: Declare three variables: num1, num2, and sum**

**Step 3: Read the first number (num1) from the user**

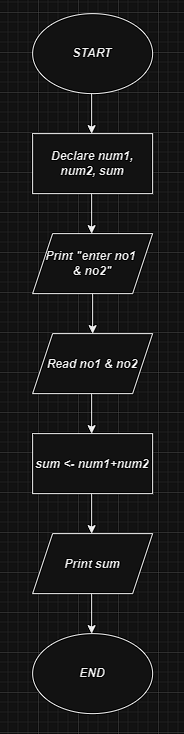
**Step 4: Read the second number (num2) from the user**

**Step 5: Add num1 and num2 and store the result in the sum variable (sum = num1 + num2)**

**Step 6: Display the value of the sum**

**Step 7: End**

**Flowchart of sum of 2 numbers:**

****

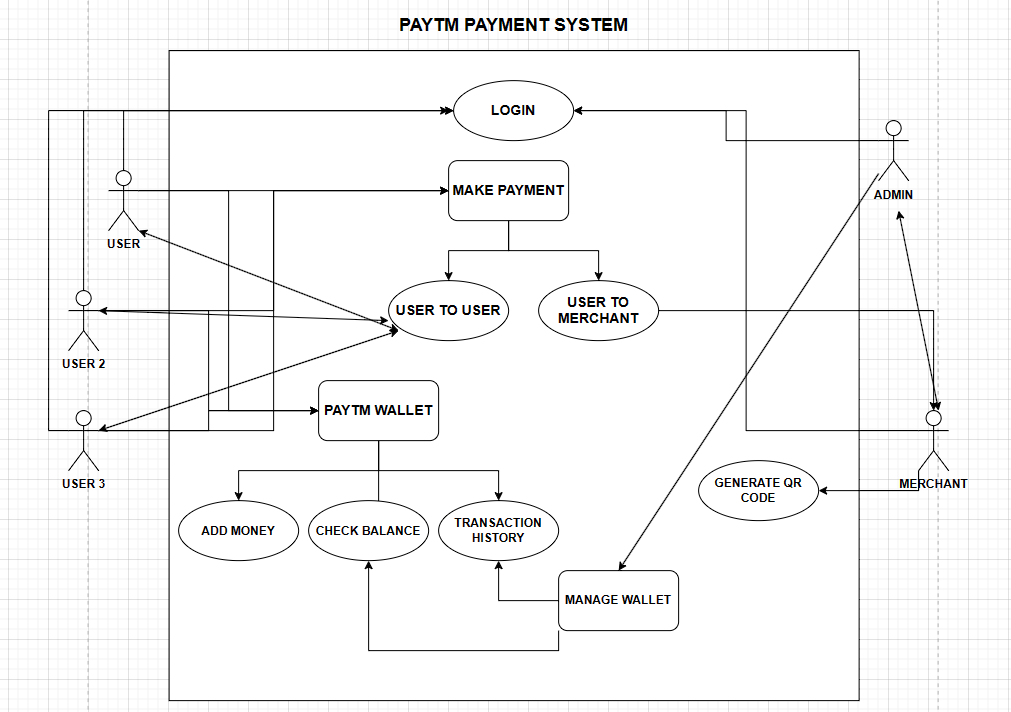
**Q5) What is Use case Diagram? Create a use-case of paytm**

* **Use case diagram:** A Use Case Diagram is a vital tool in system design, it provides a visual representation of how users interact with a system. It serves as a blueprint for understanding the functional requirements of a system from a user’s perspective, aiding in the communication between stakeholders and guiding the development process.
* **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.***

****