**SE – Overview of IT Industry**

**Assignment -1**

**1. What is software? What is software engineering?**

* **Software** is a set of instructions, also called a program that tells a computer how to perform tasks.

It is a key component of a computer system that directs hardware on what to do and how to perform tasks.

Software is divided into two main categories:-

1) system software 2) application software.

* **Software 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**

**1. System Software**

System software serves as the foundation for application software, managing

the hardware and creating an environment for applications to run.

**2. Application Software**

Application software is designed to help users perform specific task.

like a database software,Multimedia Software,Communication Software

**3. Programming Software**

Programming software includes tools and applications used by developers to create, test, and maintain software.

exa. Vscode , IntelliJ IDEA

**4. Middleware**

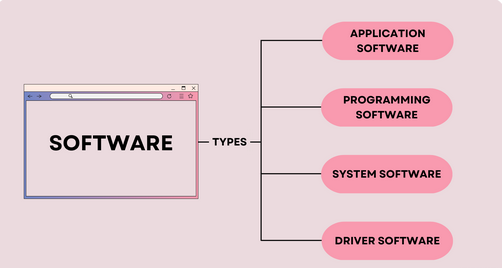
Middleware is software that acts as a bridge between system software and application software, enabling communication and data management.

exa. ODBC & JDBC

**5. Driver Software**

Specialized programs that act as a translator between a hardware device and the operating system, facilitating communication and control.

The primary purpose of driver software is to ensure that the hardware device functions properly with the computer’s operating system.

****

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

The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond

* **Here are some benefits of SDLC:**
* Increased visibility of the development process for all stakeholders involved
* Efficient estimation, planning, and scheduling
* Improved risk management and cost estimation
* Systematic software delivery and better customer satisfaction
* **The details of the SDLC process vary for different teams. However, we outline some common SDLC phases below.**

**1.** Planning & 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.

2. Define Requirements

* This phase is critical for converting the information gathered during the planning and analysis phase into clear requirements for the development team. This process guides the development of several important documents: a software requirement specification (SRS) or product specification, a Use Case document, and a Requirement Traceability Matrix document.

3. Design

* The design phase is where you put pen to paper—so to speak. The original plan and vision are elaborated into a software design document (SDD) that includes the system design, programming language, templates, platform to use, and application security measures. This is also where you can flowchart how the software responds to user actions.

4. Development

* This SDLC phase can take quite a lot of time and specialized development tools. It’s important to have a set timeline and milestones so the software developers understand the expectations and you can keep track of the progress in this stage.

5. Testing

* Before getting the software product out the door to the production environment, it’s important to have your quality assurance team perform validation testing to make sure it is functioning properly and does what it’s meant to do. The testing process can also help hash out any major user experience issues and security issues.

6. Deployment

* During the deployment phase, your final product is delivered to your intended user. You can automate this process and schedule your deployment depending on the type. For example, if you are only deploying a feature update, you can do so with a small number of users (canary release). If you are creating brand-new software, you can learn more about the different stages of the software release life cycle (SRLC).

7. Maintenance

* The maintenance phase is the final stage of the SDLC if you’re following the waterfall structure of the software development process. However, the industry is moving towards a more agile software development approach where maintenance is only a stage for further improvement.

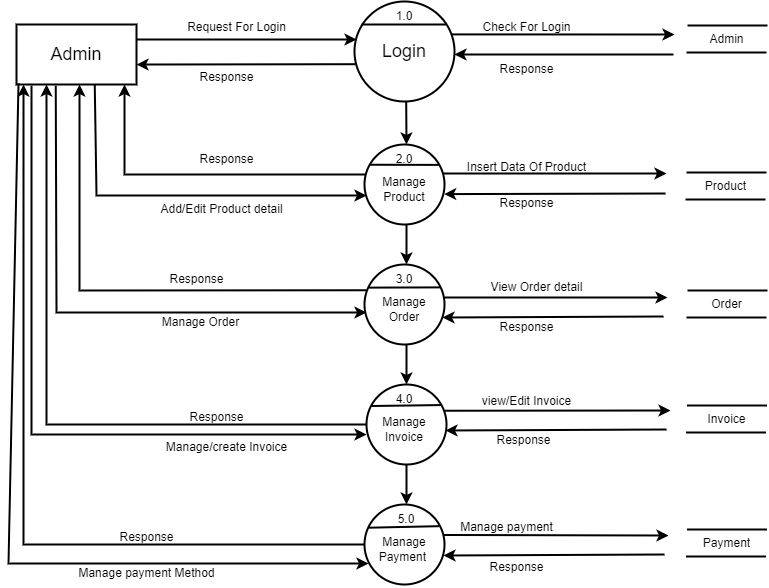
**4. What is DFD? Create a DFD diagram on Flipkart**

**DFD :-** Data Flow Diagram (DFD) represents the flow of data within information systems. Data Flow Diagrams (DFD) provide a graphical representation of the data flow of a system that can be understood by both technical and non-technical users. The models enable software engineers, customers, and users to work together effectively during the analysis and specification of requirements.

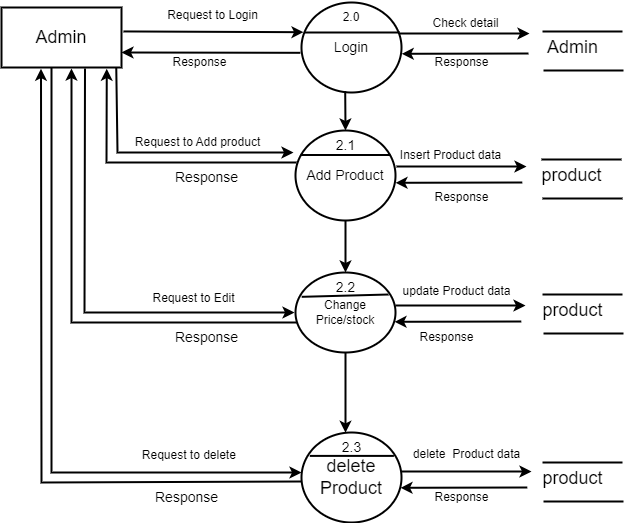
* **0 – level DFD**



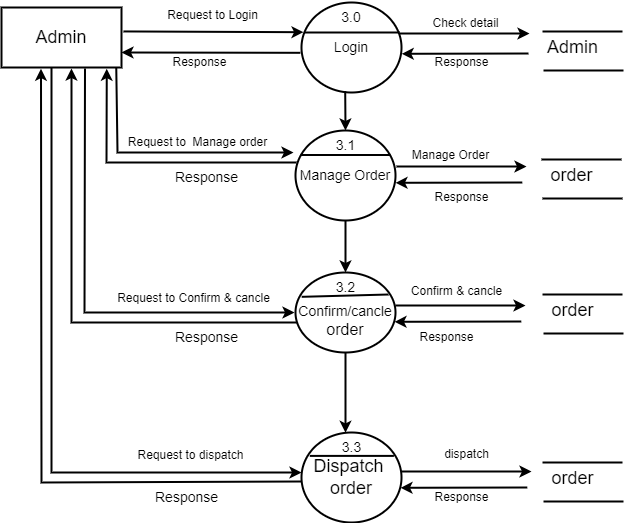
* **1 – level DFD [Admin side]**



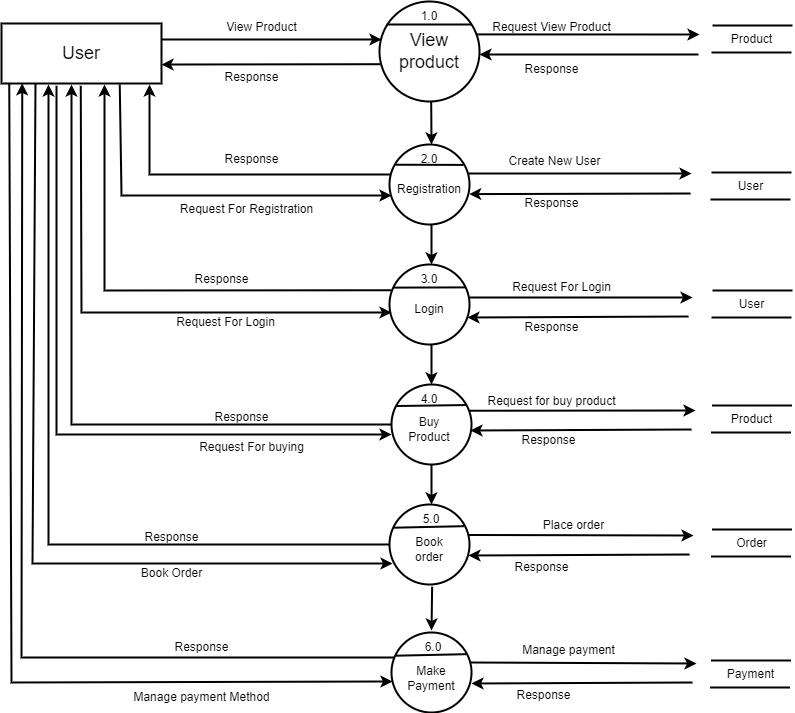
* **2 - level DFD Admin side[2.0]**



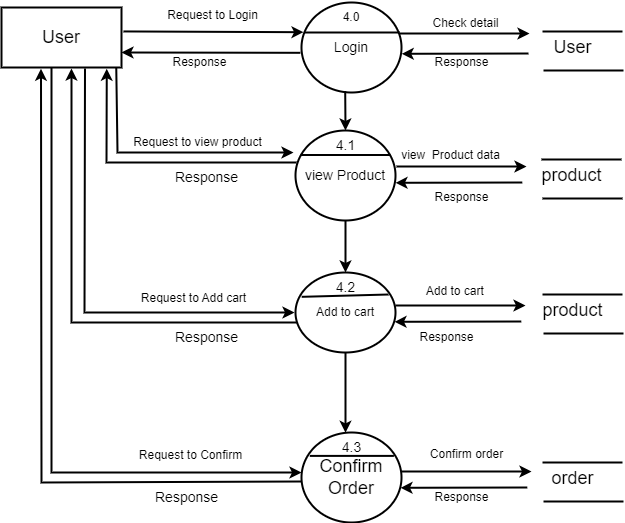
* **2 – level DFD Admin side [3.0]**



* **1 – level DFD [User Side]**



* **2 - level DFD User side[4.0]**



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

**FlowChart :-** A flow chart is a type of diagram that represents a workflow or process. A flowchart can also defined as a diagrammatic representation of an algorithm.

**Algorithm :-**  A Set of finite rules or instruction to be followed in calculation or other problem-solving operation

* **Program:**

**#include<stdio.h>**

**void main()**

**{**

**int n1,n2,ans;**

**printf("Enter 2 value:");**

**scanf("%d %d ",&n1,n2);**

**ans = n1+n2;**

**printf("%d",ans);**

**}**

* **algorithm:**

**1) START**

**2) declare variable n1,n2,ans**

**3) display enter 2 value**

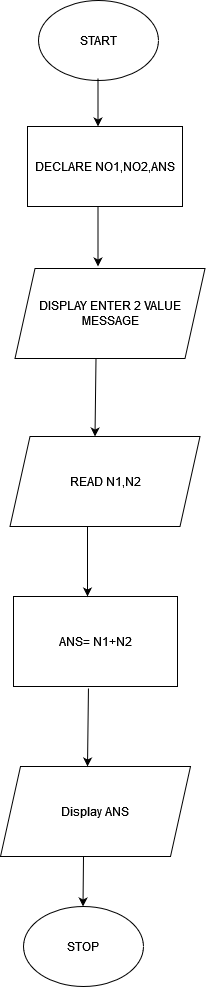
**4) read no1 and no2 from user**

**5) ans->no1+no2**

**6) display ans**

**7) STOP**

* **Flow Chart :**

****

**6) What is Use case Diagram? Create a use-case on bill payment on paytm.**

* 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.
* Use-case diagram:

