ASSIGNMENT

Module 1

1) What is software? What is software engineering?

Ans :

- Software refers to the collection of programs, data, documents, and other resources that enable a computer to perform specific tasks or functions. It includes all the intangible components that make a computer system useful and functional, such as applications, operating systems, utilities, and middleware.
- Software engineering, on the other hand, is the process of developing, implementing, and maintaining software. It applies engineering principles and techniques to software development processes to ensure that software systems are reliable, scalable, efficient, and maintainable Software engineering includes activities such as requirements analysis, . of planning, regulation, testing, deployment, and maintenance, all aimed at creating high-quality software It meets the needs of users and stakeholders.

2) Explain types of software

Ans :

Software can be categorized into several types based on various criteria, including its purpose, functionality, and how it is delivered. Here are some common types of software:

1. **System Software**: System software is designed to manage and control computer hardware and provide a platform for running application software. Examples include operating systems (e.g., Windows, macOS, Linux), device drivers, firmware, and utility programs like disk management tools and antivirus software.
2. **Application Software**: Application software is designed to perform specific tasks or functions for end-users. This category encompasses a wide range of software tailored to different needs and industries. Examples include word processors (e.g., Microsoft Word, Google Docs), spreadsheets (e.g., Microsoft Excel, Google Sheets), web browsers (e.g., Google Chrome, Mozilla Firefox), email clients, graphics design software (e.g., Adobe Photoshop), video editing software (e.g., Adobe Premiere Pro), and many more.

3. **Programming Software**: Programming software, also known as development tools or software development kits (SDKs), are used by software developers to create, debug, and maintain software applications. Examples include integrated development environments (IDEs) like Visual

Studio, Eclipse, and Xcode, as well as compilers, interpreters, debuggers, and version control systems.

4. **Middleware**: Middleware refers to software that acts as an intermediary between different applications or systems, facilitating communication and data exchange. It enables integration, interoperability, and scalability in distributed computing environments. Examples include database management systems (DBMS), message-oriented middleware (MOM), and application servers.

5. **Driver Software :** It is 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.

3) What is SLDC? Explain Each Phase of SLDC.

Ans :

SLDC stands for Software Development Life Cycle. It is a structured process used by software development teams to design, develop, test, and maintain software systems. The SDLC consists of several phases, each with its own set of activities and deliverables.

The common phases of the SDLC are:

1. **Requirement gathering:** In this phase, developers gather and analyze requirements from stakeholders to understand the needs of the software system. This involves conducting interviews, workshops, and surveys with end-users and stakeholders to document functional and non-functional requirements. The output of this phase is a detailed requirements specification document.

2. **Analysis :** This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation and Feasibility Phases. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed.

3. **Design**: Once requirements are gathered, the system design phase begins. In this phase, developers create a high-level design that outlines the architecture and structure of the software system. This includes defining system components, data models, interfaces, and algorithms. The design may be represented using diagrams such as UML (Unified Modeling Language) diagrams. The goal is to create a blueprint for implementing the software.

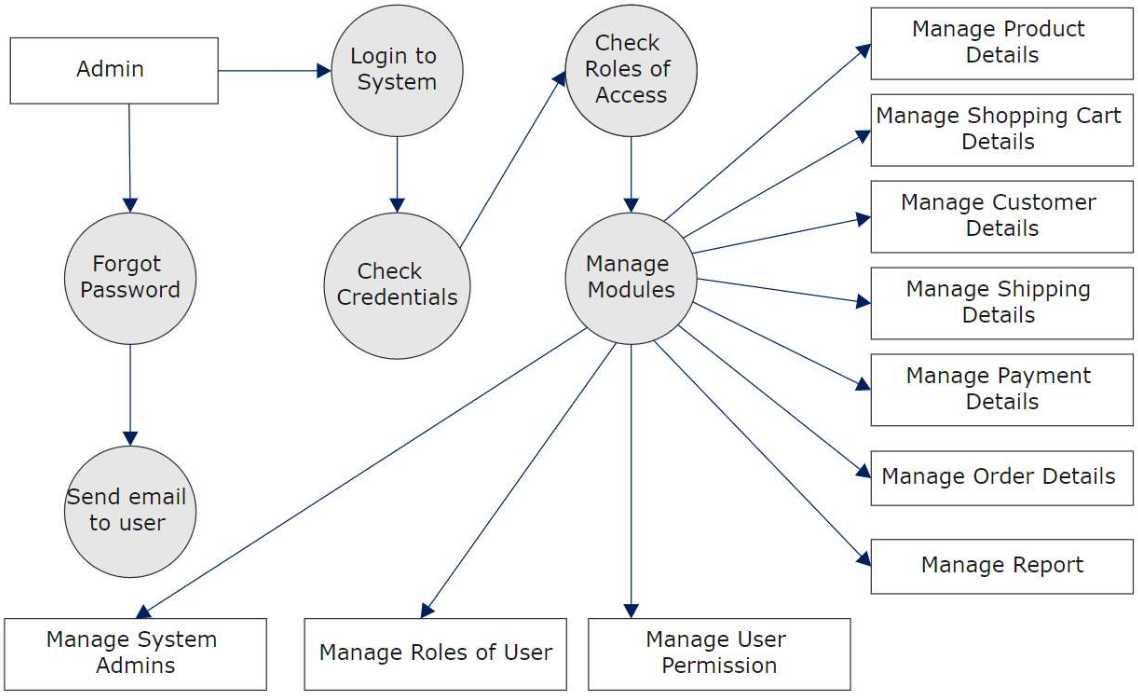
4. **Implementation**: The implementation phase involves actual coding and programming of the software system based on the design specifications. Developers write code according to coding standards and best practices, utilizing programming languages and development frameworks. This phase may also involve integrating third-party libraries, components, or APIs. The output of this phase is the executable software application.

5. **Testing**: In the testing phase, the software is tested to ensure that it meets the specified requirements and quality standards. Various types of testing are performed, including unit testing, integration testing, system testing, and acceptance testing. Testing helps identify and fix defects or bugs in the software before it is deployed to production. Test cases and test plans are created based on requirements and design specifications.

6. **Maintenance**: The maintenance phase involves ongoing support and maintenance of the software system after it has been deployed. This includes fixing bugs, addressing user feedback, implementing enhancements or new features, and ensuring the software remains compatible with evolving technologies and environments. Maintenance activities aim to optimize the performance, reliability, and security of the software over its lifecycle.

4) What is DFD? Create DFD Diagram on Flipkart.

Ans:

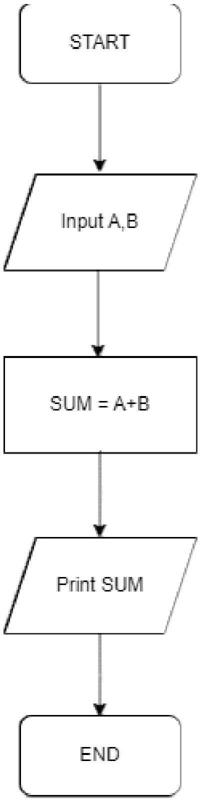


5) What is Flowchart? Create a flowchart to make addition of two numbers.

Ans :

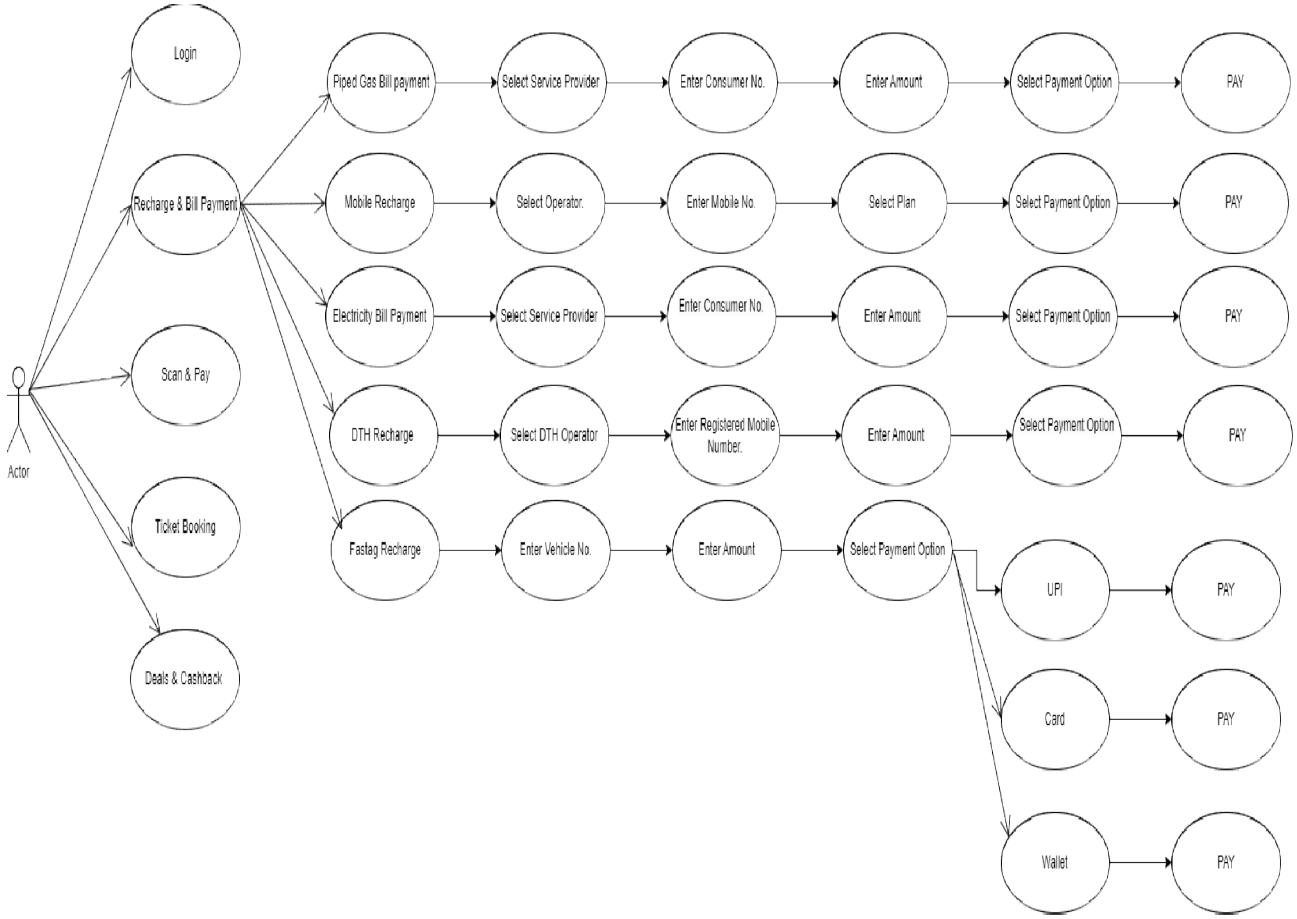
A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

Here is a Flowchart for addition of Two numbers A and B.



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

Ans : A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.


Actor
Login
Recharge & Bill Payment
Scan & Pay
Ticket Booking
Deals & Cashback
Piped Gas Bill payment
Select Service Provider
Enter Consumer No.
Enter Amount
Select Payment Option
PAY
Mobile Recharge
Select Operator.
Enter Mobile No.
Select Plan
Select Payment Option
PAY
Electricity Bill Payment
Select Service Provider
Enter Consumer No.
Enter Amount
Select Payment Option
PAY
DTH Recharge
Select DTH Operator
Enter Registered Mobile Number.
Enter Amount
Select Payment Option
PAY
Fastag Recharge
Enter Vehicle No.
Enter Amount
Select Payment Option
UPI
PAY
Card
PAY
Wallet
PAY