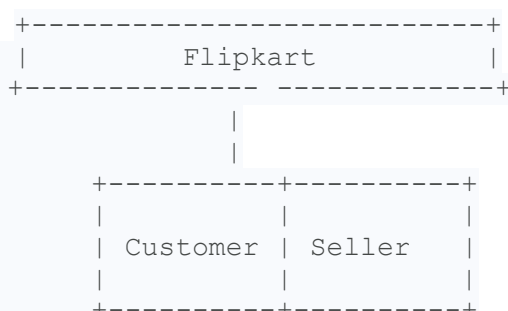
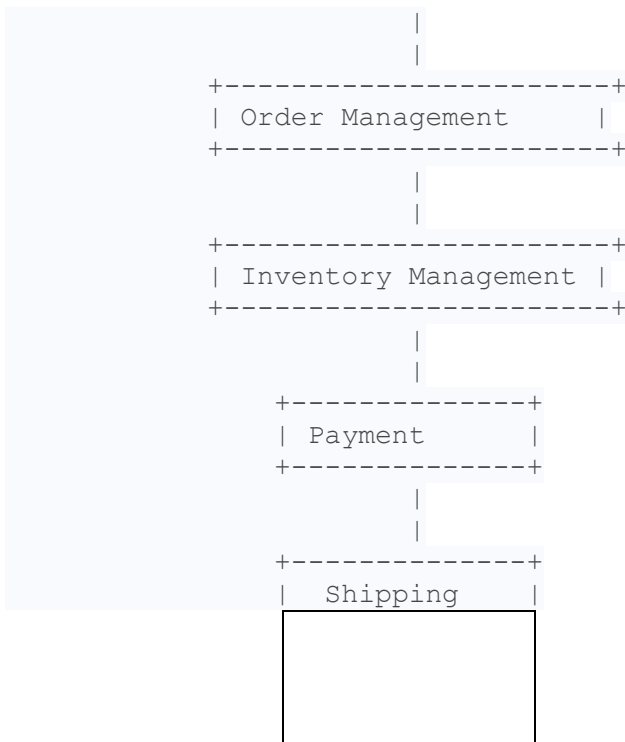
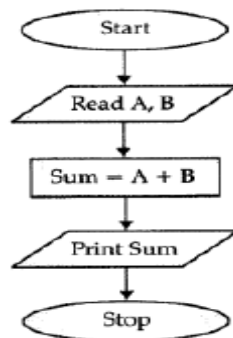


- What is software? What is software engineering?
 - Software is a set of instructions or programs that tells a computer what to do. It is intangible, meaning that it cannot be touched or seen, but it is essential for the operation of all computers and electronic devices. Software is typically written in a programming language, such as Java, Python, or C++.
 - Software engineering is the application of engineering principles and techniques to the development of software. It is a systematic approach to the design, development, testing, and maintenance of software systems. Software engineers use their knowledge of computer science and engineering to create software that is reliable, efficient, and easy to use.
- Explain types of software
 - Software is a set of instructions that tells a computer what to do. There are two main types of software:
 - System software: Controls the computer's hardware and provides a platform for applications to run on. Examples include operating systems, device drivers, and utilities.
 - Application software: Performs specific tasks for users. Examples include word processors, web browsers, spreadsheets, and video games.
 - Other types of software include programming software, middleware, and firmware.
- What is SDLC? Explain each phase of SDLC
 - the Software Development Life Cycle (SDLC) is a framework for developing software that meets the needs of its users and stakeholders. It is a systematic approach to the design, development, testing, and deployment of software systems.
 - The SDLC typically consists of the following phases:
 - Requirements gathering and analysis: This phase involves understanding the needs of the users and stakeholders of the software system, and documenting those requirements in a clear and concise way.
 - Design: This phase involves creating a high-level plan for the software system, including its architecture, components, and interfaces.
 - Development: This phase involves writing the code for the software system, and testing it to ensure that it meets the requirements.
 - Testing: This phase involves testing the software system to find and fix any errors or defects.
 - Deployment: This phase involves making the software system available to its users.
 - Maintenance: This phase involves fixing any bugs or defects that are found in the software system after it has been deployed.
- What is DFD? Create a DFD
 - A Data Flow Diagram (DFD) is a graphical representation of the flow of data through a system. It shows the inputs and outputs of the system, and the processes that transform the data. DFDs are used to document and analyze existing systems, and to design new systems. diagram on Flipkart
 - Here is a simplified DFD diagram of Flipkart:





- What is Flow chart? Create a flowchart to make addition of two numbers
 - A flowchart is a visual representation of a process. It uses shapes and arrows to show the steps in a process and the flow of data between the steps. Flowcharts are used to document and analyze existing processes, and to design new processes.
 -



- What is Use case Diagram? Create a use-case on bill payment on paytm.
 - A use case diagram is a type of Unified Modeling Language (UML) diagram that shows the interactions between a system and its users. It is used to identify the different ways that a system can be used, and to model the relationships between the different actors and use cases.
 - A use case diagram for bill payment on Paytm would typically include the following actors:
 - Customer: The person who is paying the bill.
 - Paytm: The online payment system.
 - The use cases for bill payment on Paytm would typically include:
 - Pay bill: The customer enters the bill amount and selects the payment method. Paytm processes the payment and sends a confirmation to the customer.
 - View bill history: The customer can view a history of their paid bills.

- Set up auto-pay: The customer can set up automatic payments for recurring bills.
- The use case diagram for bill payment on Paytm would show the relationships between the actors and use cases. For example, the customer actor would be associated with the pay bill and view bill history use cases. The Paytm actor would be associated with all of the use cases.
- Here is a diagram for bill payment on Paytm:

