

Software Engineering Assignment

MODULE: 1 (SDLC)

• What is software? What is software engineering?



Software:-

- Software, is a collection of computer programs and related data that provide the instructions for telling a computer what to do and how to do it.



Software engineering: -

- Software Engineering is an engineering branch related to the evolution of software product using well-defined scientific principles, techniques, and procedures. The result of software engineering is an effective and reliable software product.

• Explain types of software.

- There are three types of software.

1. System Software
2. Programming Software
3. Application Software

1) System Software

- System software provides the basic functions for computer usage and help run the computer hardware and system.
- Example: - Windows, Android, Linux, Mac.

2) Programming Software

- Programming is the process of designing, writing, testing, debugging, and maintaining the source code of computer programs. This source code is written in a programming language. The purpose of programming is to create a program that exhibits a certain desired behavior.
- Example: - Compiler

3) Application Software

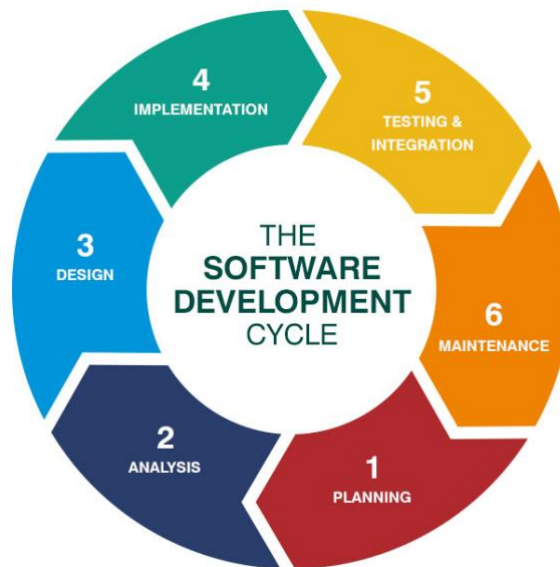
- Application software is the general designation of computer programs for performing user tasks.
- Types of Application Software:
 - ❖ Mobile App:
 - ✓ Application that runs on mobile platform.
 - ✓ Example: Instagram app,
 - ❖ Desktop App:
 - ✓ Application that runs stand-alone in a desktop or laptop computer.
 - ✓ Example: Microsoft Word, Notepad

❖ Web App:

- ✓ Apps that run on a web browser (Mozilla, Google Chrome etc.)
- ✓ Example: www.facebook.com, www.google.com

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

- The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software.



- SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.
- The following figure is a graphical representation of the various stages of a typical SDLC.
- A typical Software Development Life Cycle consists of the following stages –

Stage 1: Planning and Requirement Analysis

- Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.
- Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

- Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts.

This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

- SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS – Design Document Specification.
- This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.
- A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

- In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.
- Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

- This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

- Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).
- Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

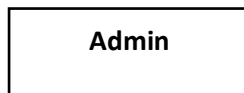
• What is DFD? Create a DFD diagram on Flipkart.

- DFD- Data Flow Diagrams
- Graphical representation of flow of data inside application.
- Used for visualization and data processing.

- DFD elements are:
1. External Entity
 2. Process
 3. Data Flow
 4. Data Store

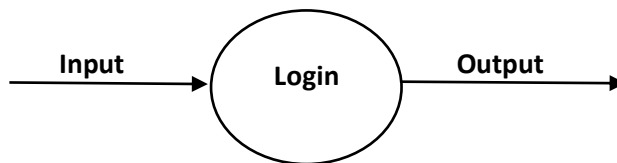
1) External entity:

Can be user or external system that performs some process or activity in project Symbolized with rectangle. If we have entity 'admin' then symbol will be



2) Process:

Work or action taken on incoming data to product output Each process must have input and output Symbolized as



3) Data Flow

Can be used to show input and output of data Should be named uniquely and don't include word 'data' Names can be 'payment', 'order', 'complaint' etc. Symbolized as

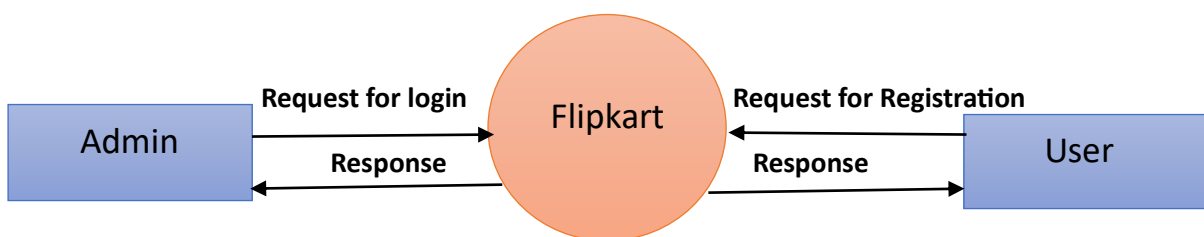


4) Data Store

Can be used to show database tables Only process may connect data stores There can be two or more process sharing same data store Symbolized as

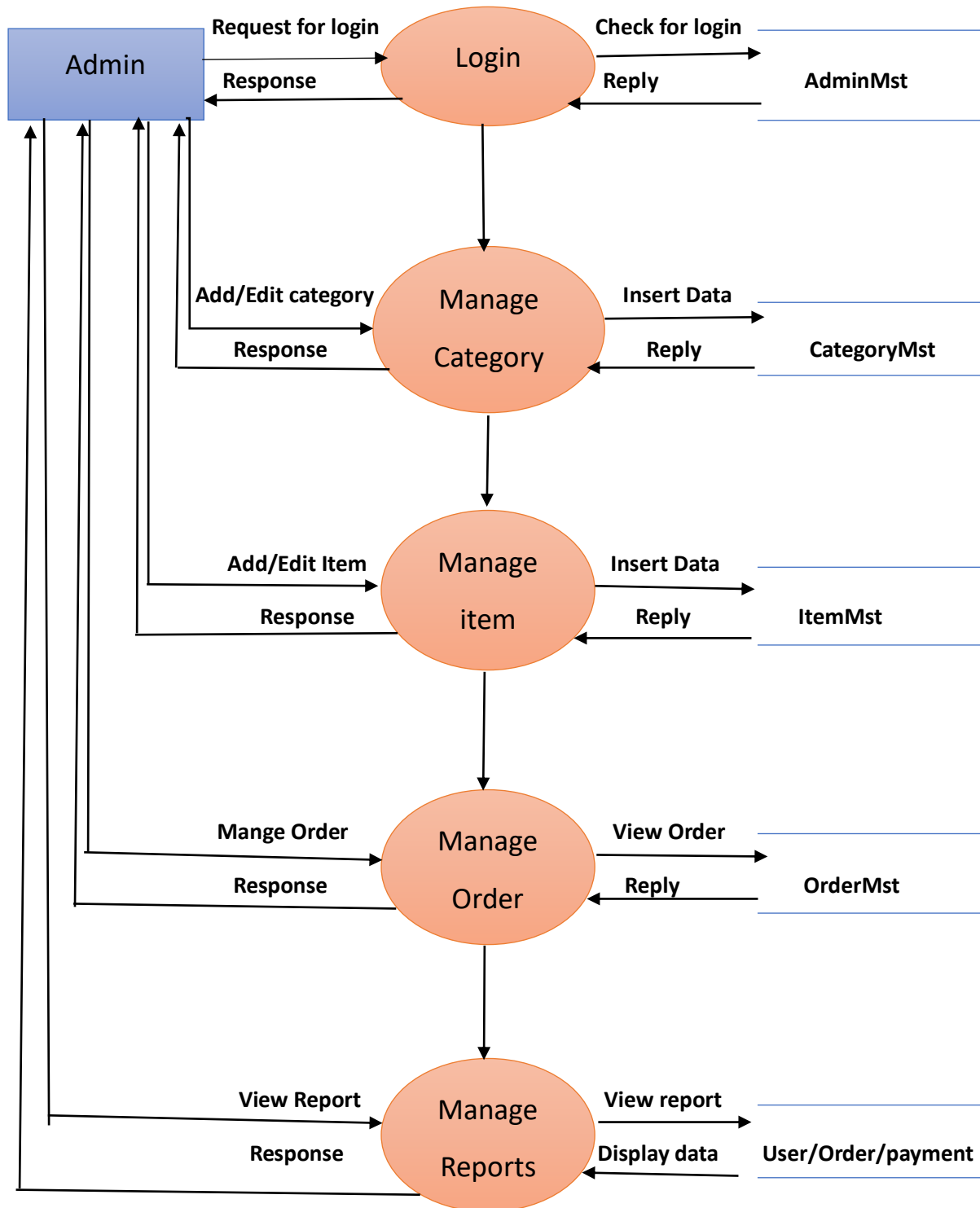


DFD diagram on Flipkart.

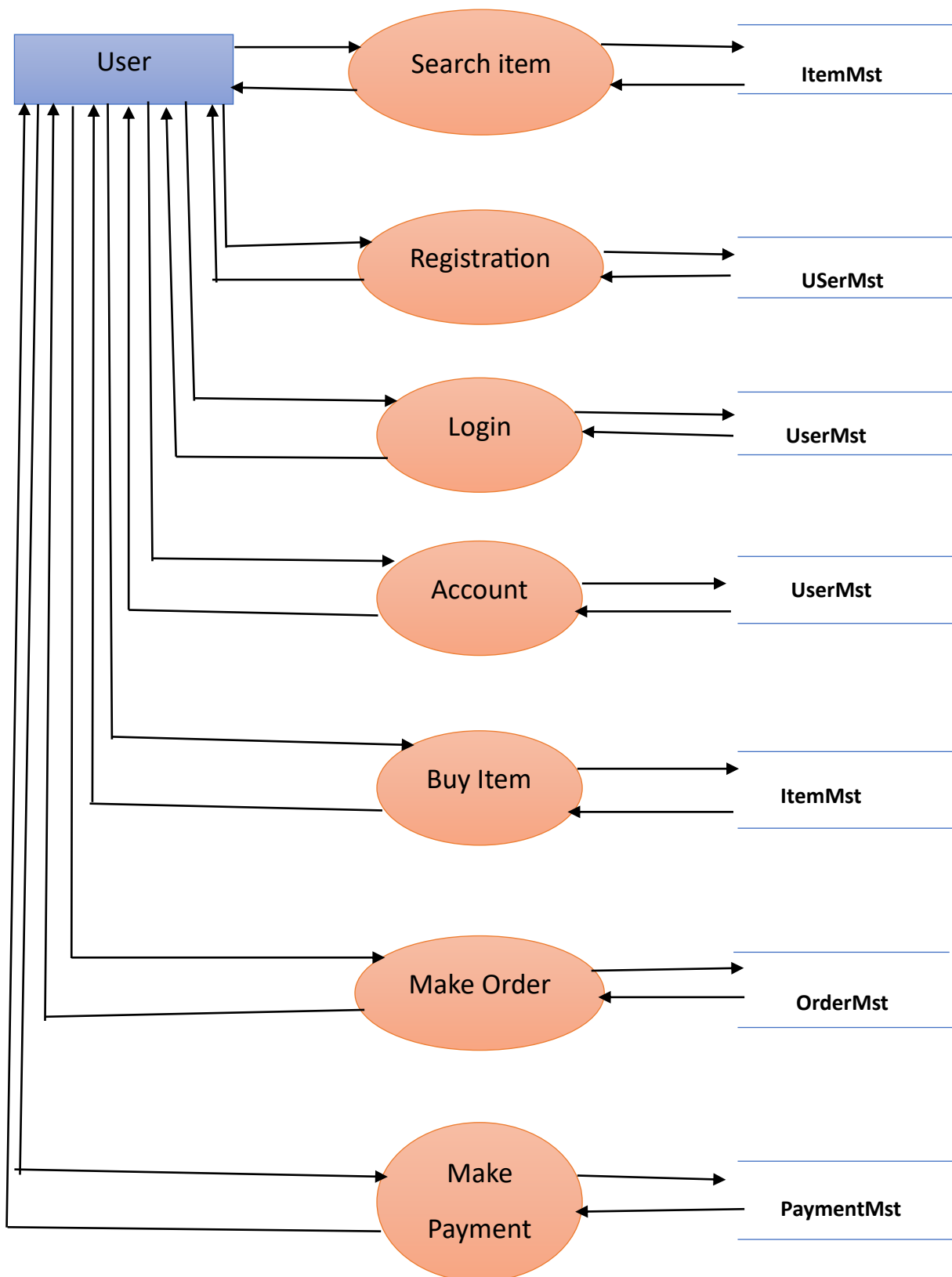


DFD zero level

Admin side DFD- 1 Level



User side DFD- 1 Level



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

- Used to show algorithm or process. Can give step solution to the problem
- The first flow chart was made by John Von Newman in 1945
- Pictorial view of process
- Flowcharts are generally drawn in the early stages of formulating computer solutions.
- Flowcharts facilitate communication between programmers and business people. These flowcharts play a vital role in the programming of a problem and are quite helpful in understanding the logic of complicated and lengthy problems.
- Once the flowchart is drawn, it becomes easy to write the program in any high-level language.
- Often, we see how flowcharts are helpful in explaining the program to others. Hence, it is correct to say that a flowchart is a must for the better documentation of a complex program. by step

Flowchart Symbol

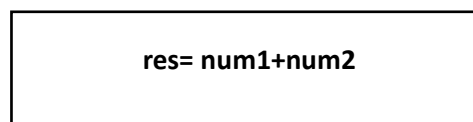
1) Start Or End:

Show starting or ending of any flow chart Symbolized as:



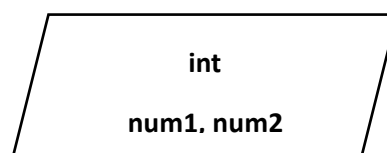
2) Process:

Defines a process like defining variables or initializing variable or performing any computation Symbolized as



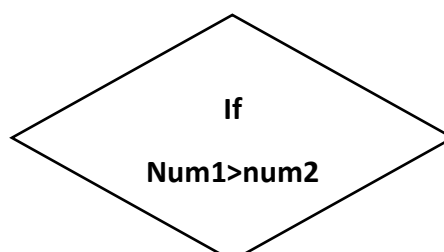
3) Input or Output

Used when user have to get or initialize any variable Like get num1 and num2 from user Symbolized as



4) Decision Making

For checking condition this symbol can be used Like if num1 is greater than num2 Can be symbolized as

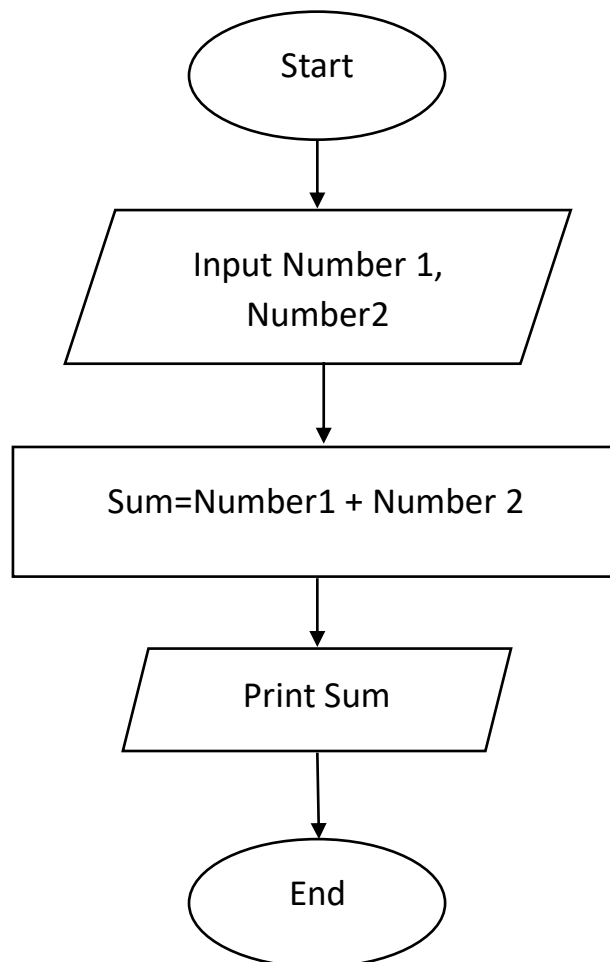


5) Flowlines Lines

showing flow of data and process Showing flow of instructions also Can be symbolized as



Flowchart to Add two number: -



• What is Use case Diagram? Create a use-case on bill payment on Paytm.

- 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.

use-case on bill payment on Paytm

