

MODULE 1(SDLC)

QUE:1 What is software? What is software engineering?

ANS:

- **Software** is a collection of codes, documents, and triggers that does a specific job and fills a specific requirement.
- It is a branch of engineering that deals with the development of software products. It operates within a set of principles, best practices, and methods that have been carefully honed throughout the years, changing as software and technology change.
- Software engineering leads to a product that is reliable, efficient, and effective at what it does. While software engineering can lead to products that do not do this, the product will almost always go back into the production stage.

QUE:2 Explain types of software.

ANS:

There are two types of software –

- System Software
- Application Software

1. System Software

- The system software is a collection of programs designed to operate, control, and extend the processing capabilities of the computer itself. System software is generally prepared by the computer manufacturers.
- These software products comprise of programs written in low-level languages, which interact with the hardware at a very basic level. System software serves as the interface between the hardware and the end users.
- Some examples of system software are Operating System, Compilers, Interpreter, Assemblers, etc.

2. Application Software

- Application software products are designed to satisfy a particular need of a particular environment. All software applications prepared in the computer lab can come under the category of Application software.
- Application software may consist of a single program, such as Microsoft's notepad for writing and editing a simple text. It may also consist of a collection of programs, often called a software package, which work together to accomplish a task, such as a spreadsheet package.

QUE:3 What is SDLC? Explain each phase of SDLC.

ANS:

SDLC: Software development life cycle.

SDLC is structure imposed on the software products that defines the planning, testing, implementation & support.

There are 6 types phases of SDLC.

1. Requirement collection
2. Analysis
3. Designing
4. Implementation
5. Testing
6. Maintenance

1. Requirement collection

In this phase we collect all necessary information about that software which is developed by developer. i.e., which kind of software is this? What to do with that software? How many components is required in that software? Etc.

2. Analysis

In this phase we define the technical requirements of the system so one can be addressed properly.

This phase is also allowed you to receive feedback and support from relevant internal and external stakeholder.

Also, during this phase the team defines the inputs and outputs of the dataflow in and out of the system by undertaking a through system analysis of business processes that need to be covered and solved by future system.

3. Designing

In this phase, all the documentation that the team created in analysis phase is used to develop the actual technical documentation of the project.

In the design phase, project members define the structure of a project components as well as key elements of the system by defining the interfaces that will exchange data within the workflow.

4. Implementation

In this phase, we implement the design into source code using coding. We also combine all the modules together into a training environment that detects errors and defects.

A test report which contains errors is prepared through test plan that includes test related task such as test case generation, testing criteria, and resource allocation for testing.

5. Testing

This phase is crucial as it directly impacts the quality of the outputs as it is where the Quality Assurance (QA) team takes assertive steps to verify and validate the elements of the information system across multiple testing scenarios.

In this phase, the QA team also helps improve code coverage through automated tests and using resources from both the back-end and the front-end of the system. Here, the QA team also carries out trial runs to collect system behaviour data for insights on what can be improved or tweaked for a superior user and system experience.

6. Maintenance

In maintenance phase where the project teams carefully assess the system to help reduce the cost of operation and maintenance through several methods like feedback collection, error detection and elimination, and optimal performance standards.

For example, a system's maintenance may also include the tracking and monitoring of the system's security, eliminating potential risks and threats, assembling a list of functionalities that need to be updated, adapting the system to environment changes and new business requirements, and more.

QUE:4 What is DFD? Create a DFD diagram on Flipkart

ANS:

DFD stands for “DATA FLOW DIAGRAM”

Through which we can represent flow of data graphically in an information system.

By using DFD we can easily understand the overall functionality of system because diagram represent the incoming data flow, outgoing dataflow, and store data in a graphical form.

It describes how the data is processed in a system in term of input & output.

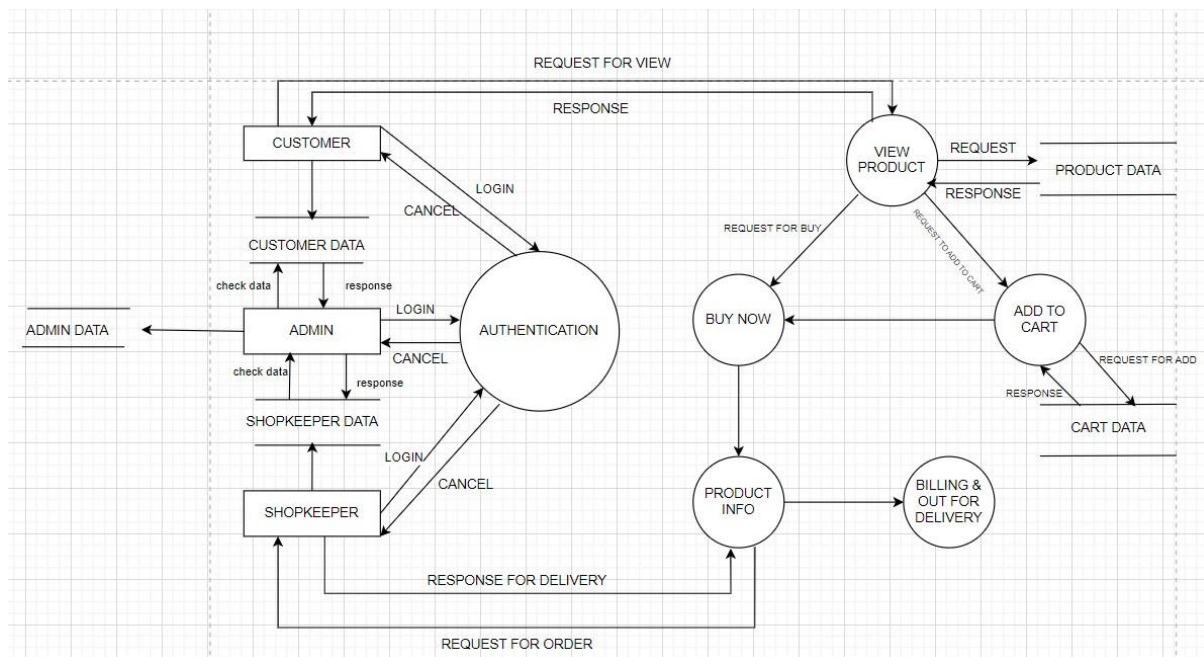


FIG: - DFD DIAGRAM OF FLIPKART

QUE:5 What is flow chart? Create a flow chart to make an addition of two numbers

ANS:

Flow chart:

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.

This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting, or managing a process or program in various fields.

Exp; Flow chart of addition of two numbers.

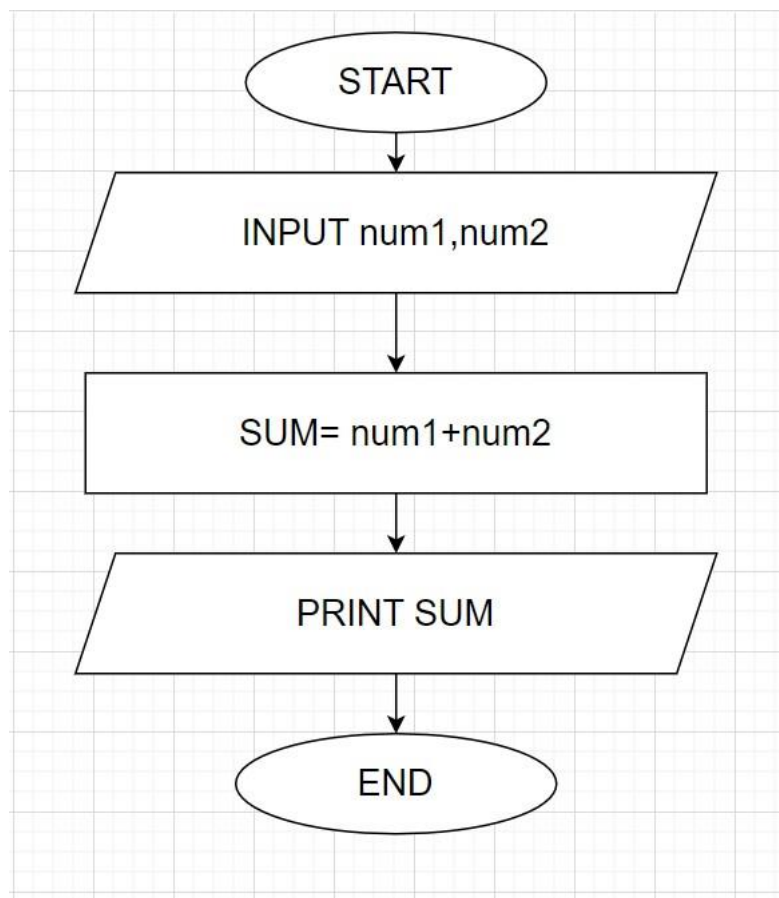


FIG: - FLOW CHART OF ADDITION OF TWO INTEGER

QUE:6 What is USE-CASE diagram? Create a use-case diagram on bill payment on Paytm.

ANS:

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

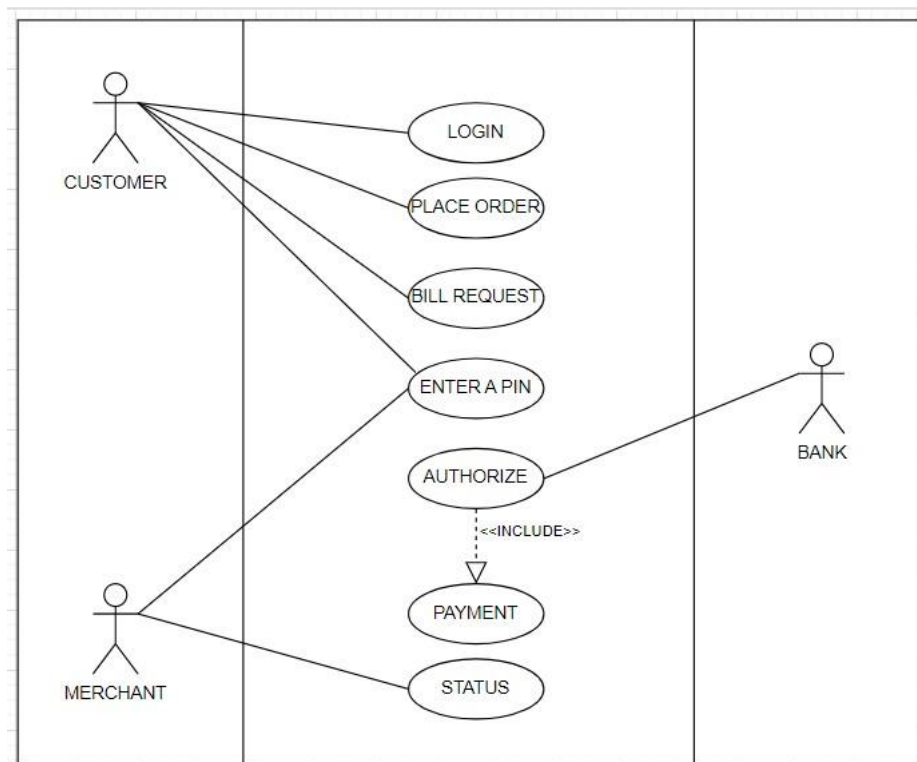


FIG: - USE-CASE DIAGRAM OF PAYTM