**Module – 1**

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

**Software :**

**Software is the language of computer**.

It 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 a systematic approach to the design, development, operation, and maintenance of a software system.

Software engineering is the art of developing quality software on time and within budget.‖

1. **Explain types of software** :
2. **System software**
3. **Application software**
4. **Driver software**
5. **Middleware**
6. **Programming software**
7. **System software** :

These software programs are designed **to run a computer's application programs and hardware**. System software coordinates the activities and functions of the hardware and software. In addition, it controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in**. The OS is the best example of system software; it manages all the other computer programs**. Other examples of system software include the firmware, computer language translators and system utilities.

**Ex. : Settings**

1. **Application software** :

The most common type of software, application software is a computer software package that performs a specific function for a user, or in some cases, for another application. An application can be self-contained, or it can be a group of programs that run the application for the user. Examples of modern applications include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.

**Ex. : paint, insta, facebook, whatsapp**

1. **Driver software :**

**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**.

**Ex. : audio drive, video drive , mp3**

1. **Middleware :**

The term middleware **describes software that mediates between application and system software or between two different kinds of application software**. For example, middleware enables Microsoft Windows to talk to Excel and Word. It is also used to send a remote work request from an application in a computer that has one kind of OS, to an application in a computer with a different OS. It also enables newer applications to work with legacy ones.

**Ex. : IOS, Windos**

1. **Programming software :**

Computer programmers use programming software **to write a code**. Programming software and programming tools enable developers to develop, write, test and debug other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

**Ex. : Dev C++,Turbo**

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

**SDLC** is a structure imposed on the development of a software product that defines the **process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support**. There are a number of different development models.

A **S**oftware **D**evelopment **L**ife **C**ycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

**Phase of SDLC :**

**1.Requirement**:

In this phase, all the requirements are collected from the customer/client. They are provided in a document called Businessmen requirement specification (BRS) and System requirement specification (SRS). All the details are discussed with the customer/client in detail.

**2.Analysis Phase :**

The analysis phase defines the requirements of the system, independent of how these Requirements will be accomplished.

This phase defines the problem that the customer is trying to solve.

The deliverable result at the end of this phase is a requirement document.

Ideally, this document states in a clear and precise fashion what is to be built.

**This analysis represents the ―what” phase.**

The requirement documentaries to capture the requirements from the customer's perspective by defining goals.

This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture.

The architecture defines the components, their interfaces and behaviors.

The deliverable design document is the architecture.

This phase represents the ―how” phase.

Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established.

The design may include the usage of existing components.

**3.Design**:

Design Architecture Document

Implementation Plan

Critical Priority Analysis

Performance Analysis

Test Plan

The Design team can now expand upon the information in the requirement Document.

The requirement document must guide this decision process. Analyzing the trade-offs of necessary complexity allows for many things to remain simple which, in turn, will eventually lead to a higher quality product. The architecture team also converts the typical scenarios into a test plan.

**4.Implementation Phase :**

In the implementation phase, the team builds the components either from scratch or by composition.

Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility.

The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging. The end deliverable is the product itself. There are already many established techniques associated with implementation.

**5.Testing:**

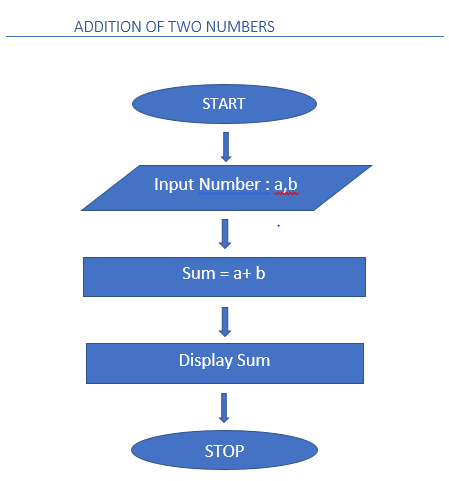
Testing is carried out to verify the entire system. The aim of the tester is to find out the gaps and defects within the system and also to check whether the system is running according to the requirement of the customer/client.

**6. Maintenance:**

Once the product has been delivered to the client a task of maintenance starts as when the client will come up with an error the issue should be fixed from time to time.

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

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 **.**

****

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

**DFD:**

* DFD stand of “Data Flow Diagram “. It is also known as a “Bubble Chart”.
* Through which we can represent the flow of data graphically on an Information system.
* By using DFD we can easily understand the overall functionality of system because diagram represents the incoming data flow, outgoing data flow and store data in a graphically form.
* It describe how data is processed in a system in term of input and output.

**COMPONUNTS:**

1. External Entity :

(INPUT) (OUTPUT)

1. Data Flow:
2. Process Or Bubble :
3. Data Store:

**Rules of DFD:**

1. Each process should have at least one input and one Output.
2. Each data store should have at least one data flow in and one data flow out.
3. All process in a DFD go to either another process or data store.
4. All the external entity must be connected through a process and entity can provide something to software (S/W) as well as entity can consume some data from the software(S/W).

**Types Of DFD:**

1. **0th Level DFD:**

It is a diagram which provide the entire system data flows and processing with a single process is called as context.

input PROCESS

1. **1st Level DFD:**

This is a more detailed version of the previous level that includes the database and version important unit.

Input process 1

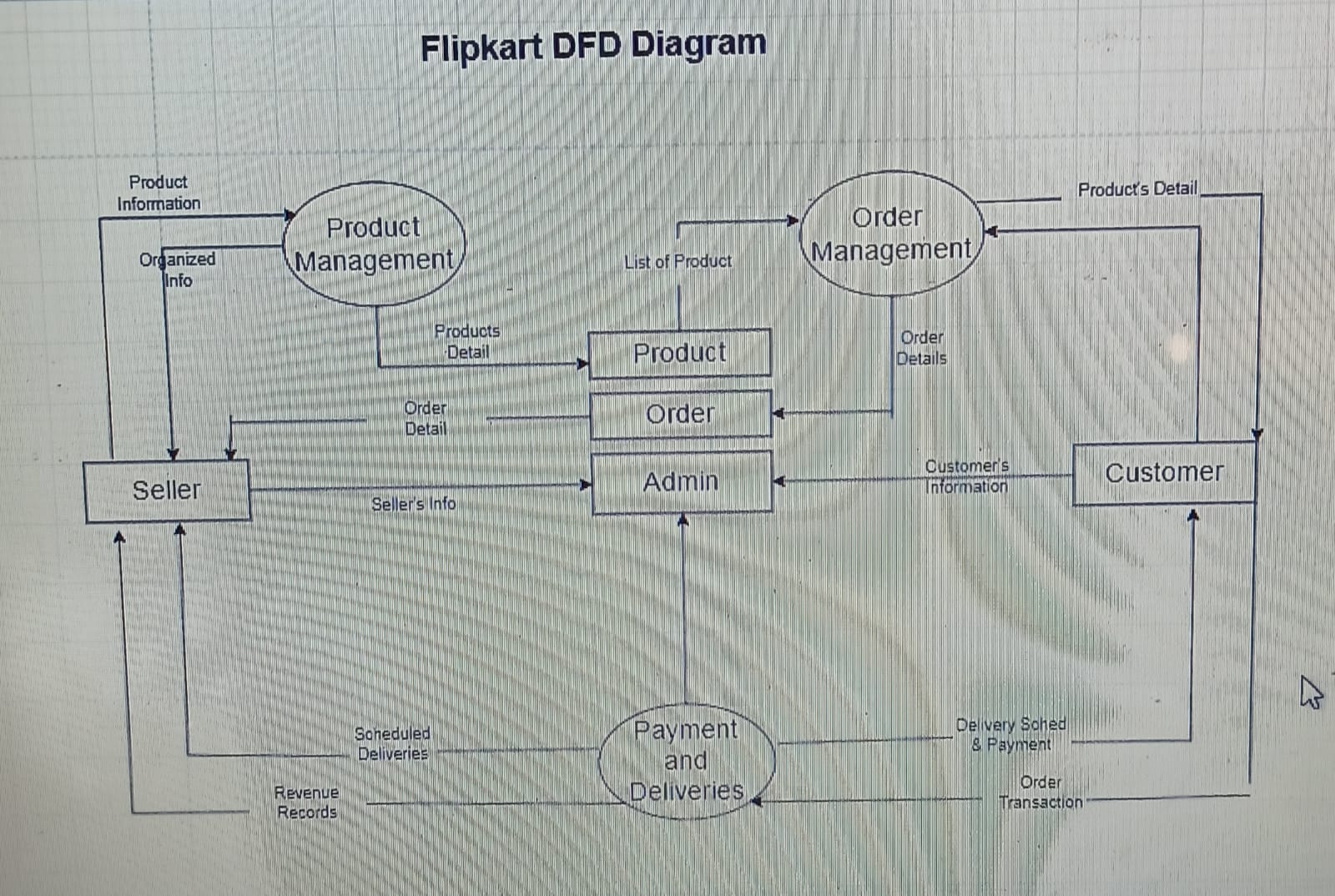
Data store Process 2

**Advantages of DFD :**

* It helps us to understand the functioning and the limits of a system.
* It is a graphical representation which is very easy to understand as it helps visualize contents.
* Data Flow Diagram represent detailed and well explained diagram of system components.
* It is used as the part of system documentation file.
* Data Flow Diagrams can be understood by both technical and nontechnical person because they are very easy to understand.

**Disadvantages of DFD :**

* At times DFD can confuse the programmers regarding the system.
* Data Flow Diagram takes long time to be generated, and many times due to this reasons analysts are denied permission to work on it.

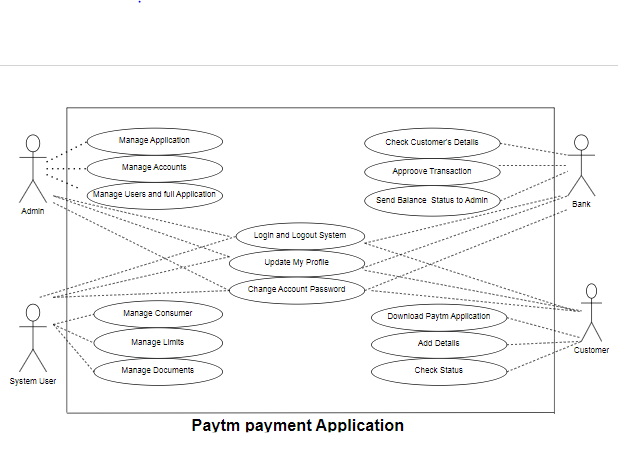


**6.. What is Use Case Diagram ? Create a use case on bill payment on paytm.**

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.**

****