**MODULE : 1**

**Software Engineering - Overview of IT Industry**

**1. What is Software ? What is Software Engineering ?**

**Ans.** Software refers to a set of instructions, programs, or data that tell a computer or device how to perform specific tasks. It includes everything from operating systems and applications to games and utilities.

The software can be best defined as a set of instructions, technically referred to as programs, that perform operations and specific tasks based on the commands of the user. Every single task that a user intends to perform is regulated by software. Made of binary language (ones and zeroes), there is a variety of software for different tasks.

Software refers to the opposite term of Hardware, which refers to the physical aspects of the device. Software refers to the objects inside the system , i.e. applications, programs, data, instructions, etc. Software can be categorised into many categories based on their functionality, purpose and uses.

* **Software Engineering :**

**Assignment**

Software Engineering is a systematic, disciplined and quantifiable study and approach to the design, development, operation and maintenance of a software system. It can also be described as a systematic approach to developing, designing, maintaining, and managing software systems. It involves applying engineering principles and practices to the entire software development process, from initial concept and requirements gathering to design, implementation, testing, deployment, and maintenance.

Software engineers use various methodologies, tools, and techniques to create high-quality software that meets the needs of users or organizations. This includes activities such as:

1. **Requirements Analysis**
2. **Design**
3. **Implementation**
4. **Testing**
5. **Deployment**
6. **Maintenance**

Overall, software engineering is about applying systematic approaches to software development to produce high-quality, reliable, and maintainable software solutions that meet the needs of users and stakeholders.

**2 . Explain Types of Software.**

**Ans.** Software can be categorized into several types based on various conditions, including its purpose, functionality, and distribution method. Here are some common types of software:

**A. System Software :**

This type of software manages and controls computer hardware so that other application software can perform tasks. These softwares are designed to handle and co-ordinate with hardware of computer so that other applications can run smoothly. The perfect example of system softwares are operating systems. Example, Windows, macOs, Linux, etc.

System software also includes applications that come pre-installed in computer and other devices. Some of the examples of these applications are calculator, calendar, antivirus programs, disk management tools and etc.

**B. Application Softwares :**

Application software is designed to perform specific tasks or solve particular problems for end-users. It includes a wide range of programs such as word processors, spreadsheet applications, graphic design software, web browsers, email clients, and video editing tools. These type of softwares are the most common type of softwares found on the computer and other devices. The primary examples of these application softwares are Microsoft Excel, Powerpoint, Word, Apple Pages, Paint, etc.

**C. Driver Softwares :**

Driver software, often simply referred to as drivers, is a type of system software that facilitates communication between hardware devices and the operating system (OS) of a computer or other electronic devices. Drivers act as translators or in simple words as a middlemen, allowing the OS to understand and interact with hardware components such as printers, graphics cards, network adapters, and storage devices. There are certain conditions for these drivers, such as compatibility, functionality, updates, installation, etc. Driver softwares are one of the most important softwares in computer and other devices. Examples of driver softwares are video driver, audio driver, keyboard driver, storage drivers, etc.

**D. Middleware :**

Middleware softwares are like courier service of computer and other devices,

These software do the work of connecting one software to another. For example, allowing Windows to Tak to Excel or powerpoint. Middleware is software that serves as an intermediary between different applications or systems, enabling communication and data exchange. It facilitates integration between software components and often provides additional services such as security, transaction management, and messaging. Example: database middleware, application server middleware, etc.

**E. Programming Softwares :**

Programming softwares refer to those softwares which allows developers to write various codes, which allows them to create different kinds of apps and games and other softwares also. These programming softwares allow developers to create, debug, improve and maintain softwares. Examples, IDEs like Android Studio, VScode, Xcode or Compilers like Dev C++, Turbo C++, etc.

**F. Enterprise Softwares :**

Enterprise Softwares Are softwares which are used by organizations, not Individuals. These softwares are used to manage accounting, sales, marketing, and human resources, analyse data, handle payments and projects, and develop custom applications. There is a wide range of enterprise software solutions available on the market. Among them are universal software, which entrepreneurs can use regardless of the specifics of their operations, as well as specialized programs dedicated to specific industries. Example, WordPress, AWS( Amazon Web Services), etc.

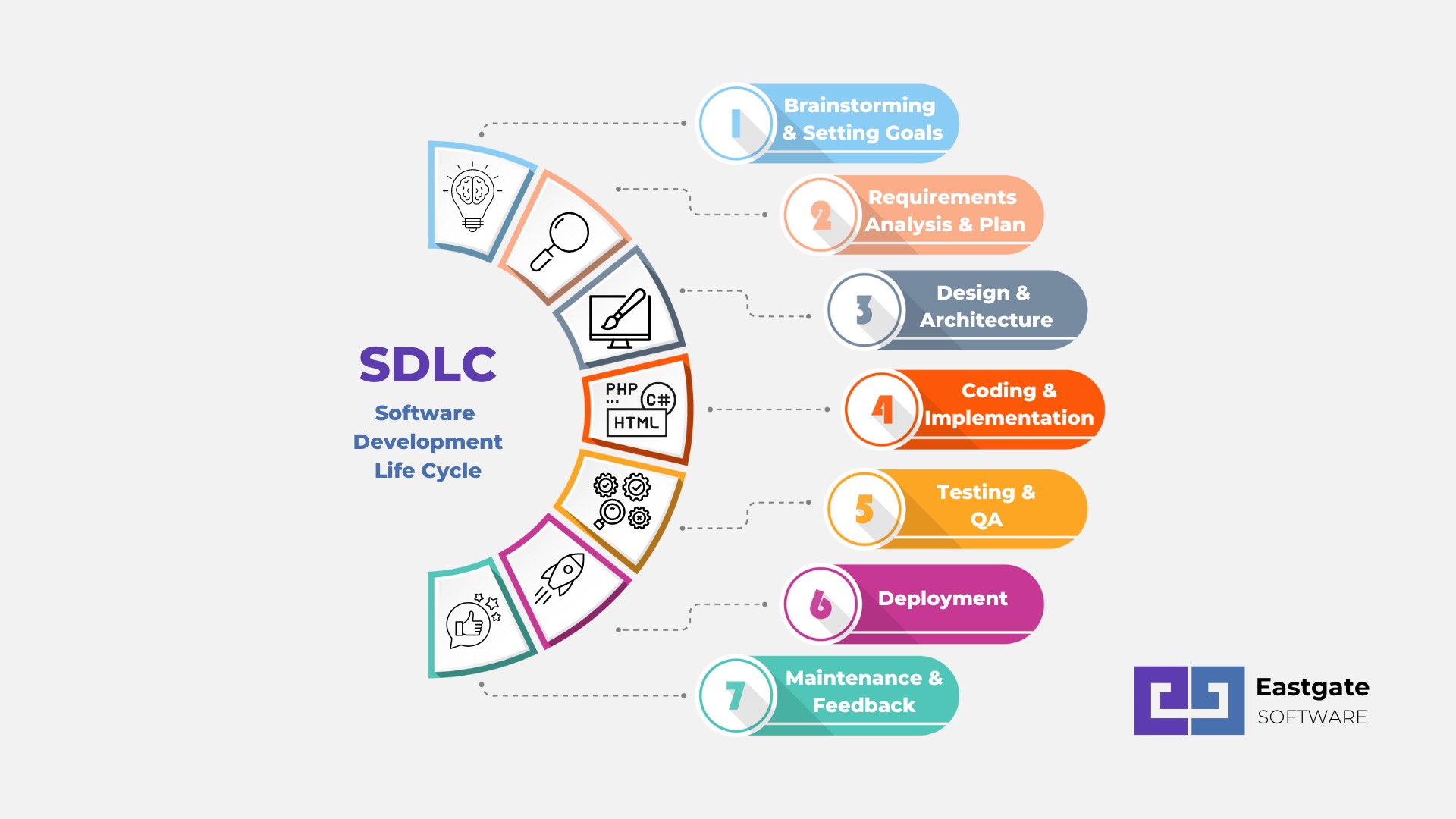
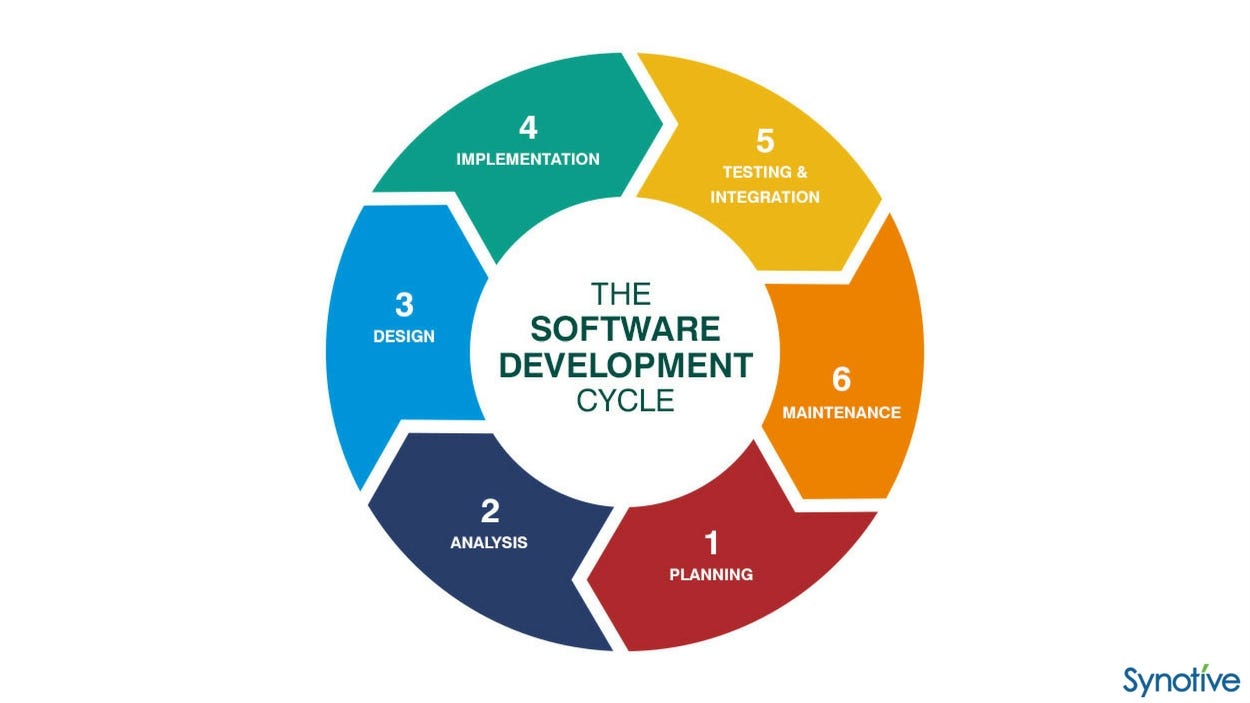
**3. What is SDLC ? Explain each phase of SDLC.**

**Ans.** SDLC stands for Software Development Life Cycle. As it’s name suggests SDLC is the whole process from thought of creating a software to its implementation to its maintenance. In other words, It's a framework that defines the processes used by organizations to design, develop, test, and deploy software applications. The main objective of the SDLC is to produce high-quality software that meets or exceeds customer expectations, is delivered on time, and stays within budget.

SDLC can be divided into 7 basic phases. Lets see each phase below :

* **Planning**
* **Analysis**
* **Design**
* **Implementation**
* **Testing**
* **Deployment**
* **Maintenance**

For better understanding of SDLC, observe the photos below



Lets study this process in detail, each phase is explained below :

1. **Planning :**
   * Identify the need for software.
   * Define project scope, goals, and objectives.
   * Gather requirements from stakeholders.
   * Create a project plan, including timelines, budget, and resources.

**B. Analysis :**

* + Analyze and refine gathered requirements.
  + Understand the Scope of the project.
  + Create detailed requirement specifications.
  + Conduct feasibility studies to ensure the project is viable.

**C. Design :**

* + Design the architecture of the software.
  + Create system and database designs.
  + Develop wireframes or prototypes to visualise the user interface.
  + Define technical specifications for developers to follow.

**D. Implementation :**

* + Write code based on the design specifications.
  + Develop software modules or components.
  + Follow coding standards and best practices.
  + Perform unit testing to ensure individual components work as expected.

**E. Testing :**

* + Test the software to identify and fix defects.
  + Conduct various types of testing such as :
    - * Unit Testing : Testing individual components.
      * Integration Testing : Testing interactions between components.
      * System Testing : Testing the entire system.
      * User Acceptance Testing (UAT) : Testing with end users to ensure it meets their requirements.
      * Perform regression testing to ensure new changes don’t introduce new defects.

**F. Deployment :**

* + Deploy the tested software to the production environment.
  + Ensure proper configuration and setup.
  + Train end-users if necessary.
  + Monitor the deployment for any issues and address them properly.

**G. Maintenance :**

* + Provide ongoing support and maintenance.
  + Fix bugs and issues reported by users.
  + Enhance the software by adding new features or improvements.
  + Perform regular updates to keep the software secure and up-to-date.

This whole process of SDLC requires proper communication between the developer team and the stakeholder. Proper communication and regular testing makes the process of SDLC easier. The process of SDLC is similar to the food chain of nature. The process of SDLC is followed by each and every software.

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

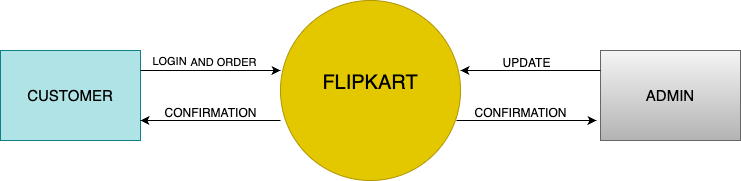
**Ans.** DFD stands for “Data flow Diagram”. DFDs are used to explain the structure and the ways of working of a particular organisation in the form of a diagram. DFD is a visual representation that illustrates how the data is transferred from one end to other, how the processes are done, so basically explaining how information and data flow in systems or between systems. DFDs make the job of understanding the structure of particular data easy, which results in creation of systems that are smooth and efficient. DFD consists of certain levels, which are not always necessary. The creation of these multiple level DFDs depends on the purpose and the audience we are targeting. These levels are explained in short below :

1. **Level 0 DFD :**

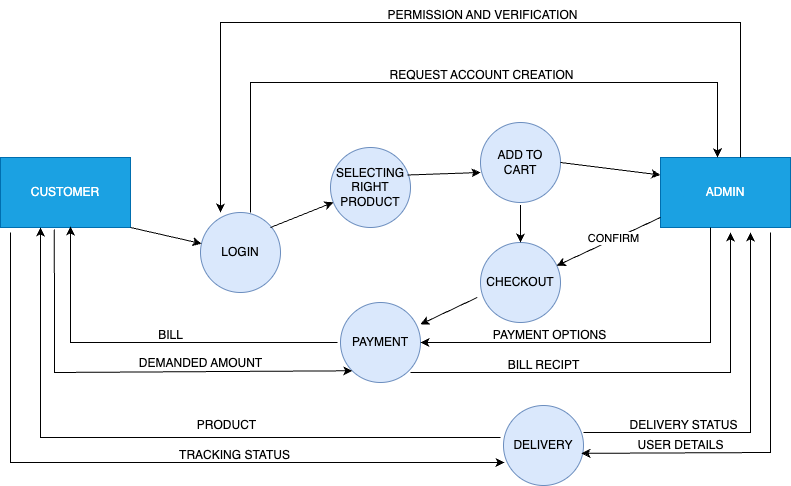
Level 0 DFDs are considered the highest level, as they provide broad overview of the system. It typically shows the main processes and external entities interacting with the system. Stakeholders often observe the level 0 DFDs to understand the scope and boundaries of a particular system.

**B. Lower Level DFDs( Level 1, Level 2, etc. ) :**

Lower level DFDs are used to describe the level 0 DFDs in more detail and these DFDs include more information in depth about the points discussed in the level 0 DFD. These DFDs are very useful for explaining the functions of a complex systems, as these DFDs use more data. The example of these complex system include E-Commerce websites like Flipkart, which we will use an example for creating a DFD.

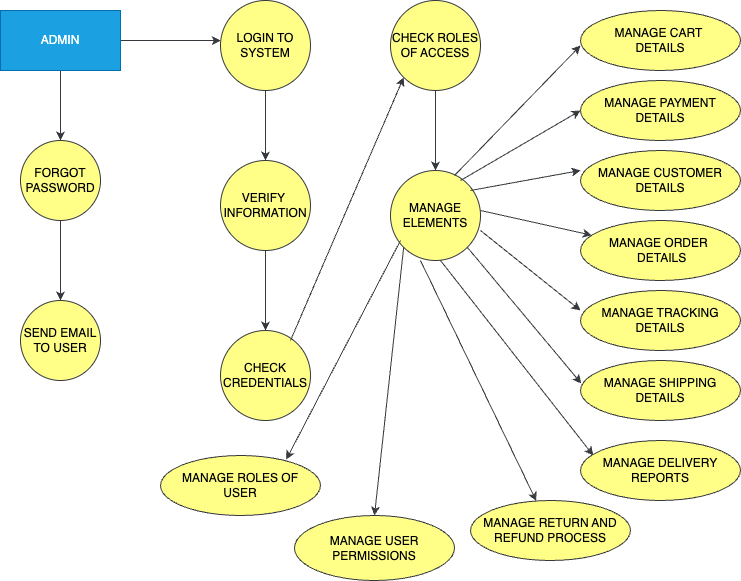
****

**DFD Level 0**

****

**DFD Level 1**

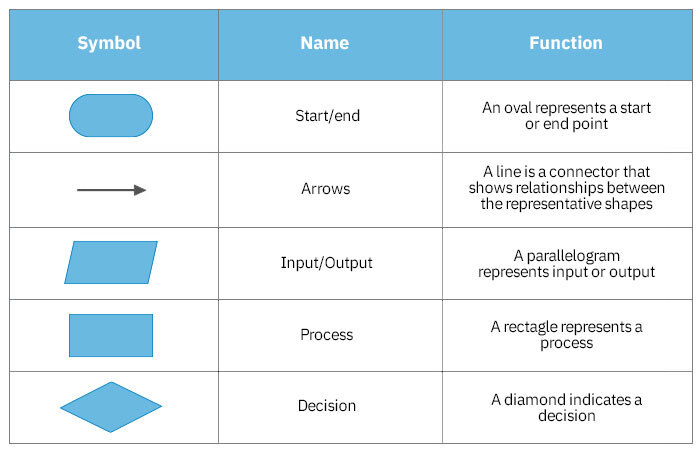
For further explanation of DFD i.e. Level 2 DFD we will use the entity ADMIN.



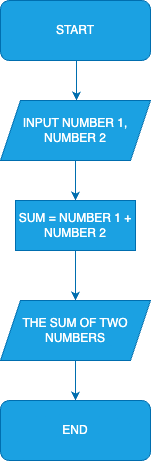
**DFD Level 2**

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

**Ans.** Flowchart is a part of the diagram family. Flowchart is a type of diagram which indicates or represents workflow or process. Flowchart is a graphical representation of a particular system in a step-by-step diagram. For showcasing the step-by-step process it uses certain shapes and symbols. These shapes and symbols are given in the table below :



We will create a flowchart for understanding how and for what it is actually used. Lets make a flowchart to make addition of two numbers



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

**Ans.** A use case diagram is a part of the diagram family. It is a type of behavioural diagram in the Unified Modelling Language (UML) that illustrates the interactions between users (actors) and a system to accomplish specific goals or tasks. Use case provides a high level functionalities and information about particular system. A use case diagram has some key components, some of them are given below :

1. **Actor :**

Actor is not any particular person or entity, but the one who is communicating with the system. It can be a person, a system talking to system and etc.

**B. Use Case :**

A use case in a use case diagram is a visual representation of the tasks an user can perform or interact with a particular system. Each use case

describes a particular action or sequence between user and the system.

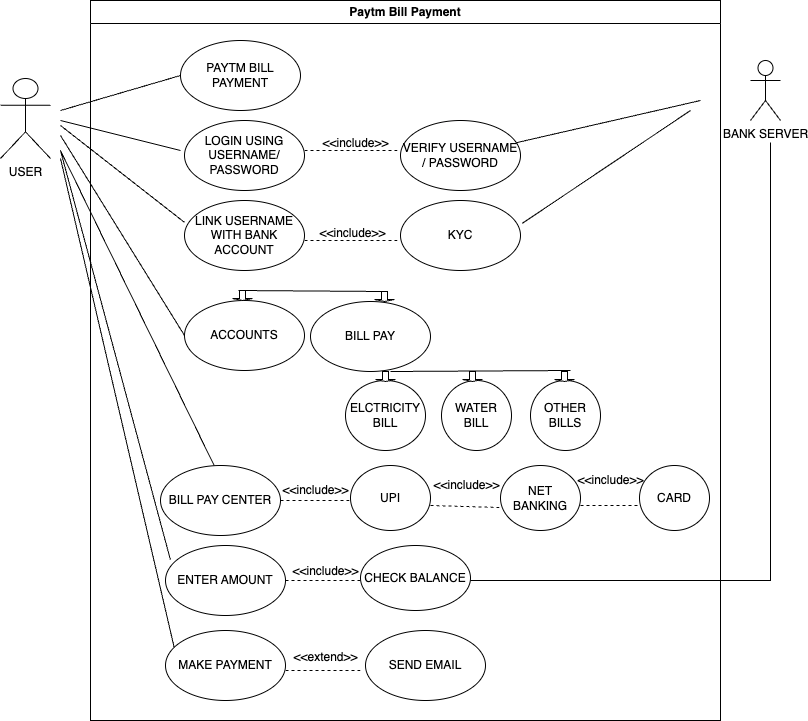
**C. Relationships :**

Use Case Diagrams show relationships between actors and use cases, such as associations, generalizations, and includes/extends relationships between use cases. Arrows are used for showing relationships between use cases . Include is used for use cases which are necessary and exclude is used for use case which in optional.

**D. System Boundary :**

System boundary is as name suggests the boundary inside which all the use cases and relationships are included or shown, while making a use case diagram. A system boundary defines the scope of what a system will be. A system cannot have infinite functionality. So, it follows that use cases also need to have definitive limits defined.

Lets create a use case diagram on Paytm bill payment, so that we can understand use case properly and learn how it works,

**Over**