

# What is System Design?

**Ans:** System Design is defined as a process of creating an architecture for different components, interfaces, and modules of the system and providing corresponding data helpful in implementing such elements in systems.

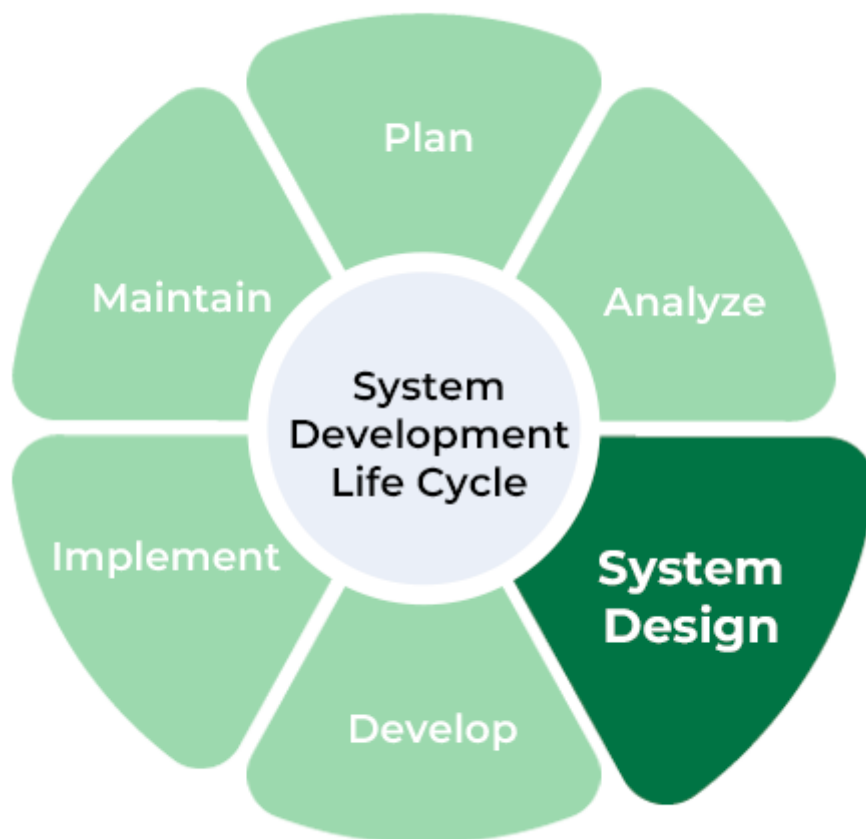
---

## Why learn System Design?

**Ans:** In any development process, be it Software or any other tech, the most important stage is **Design**. Without the designing phase, you cannot jump to the implementation or the testing part. The same is the case with the System as well.

The **stages in System Development Life Cycle** are as follows:

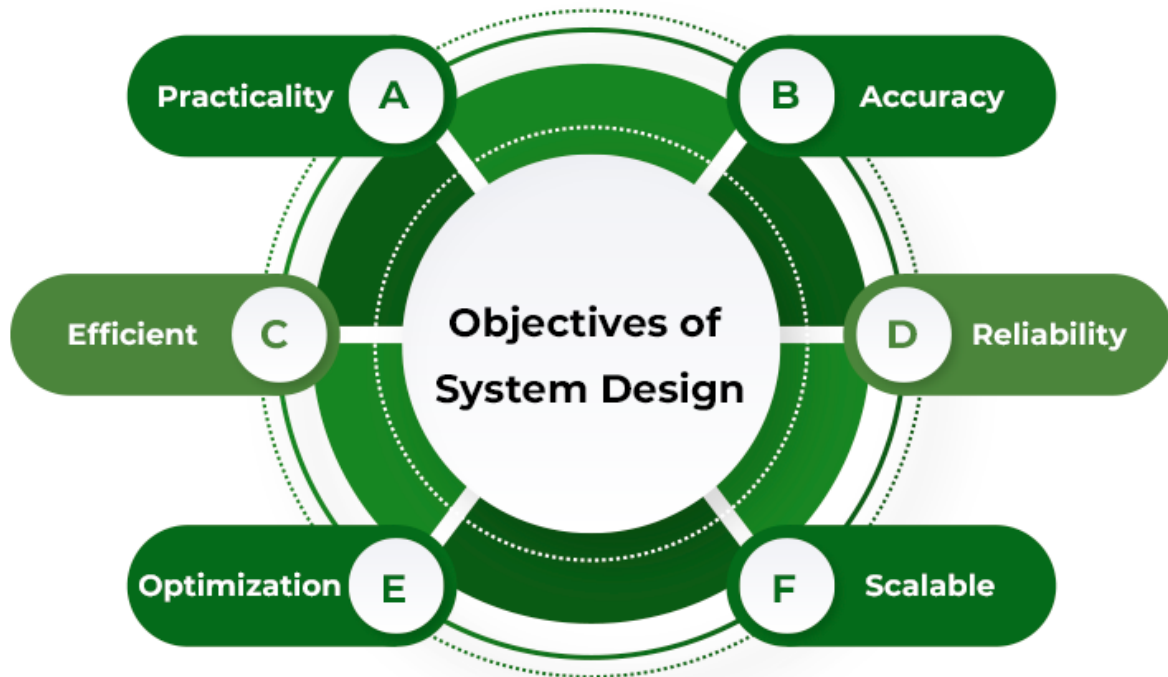
1. Plan
2. Analyse
3. Design
4. Develop
5. Implement
6. Maintain



---

# Objectives of System Design

Ans:



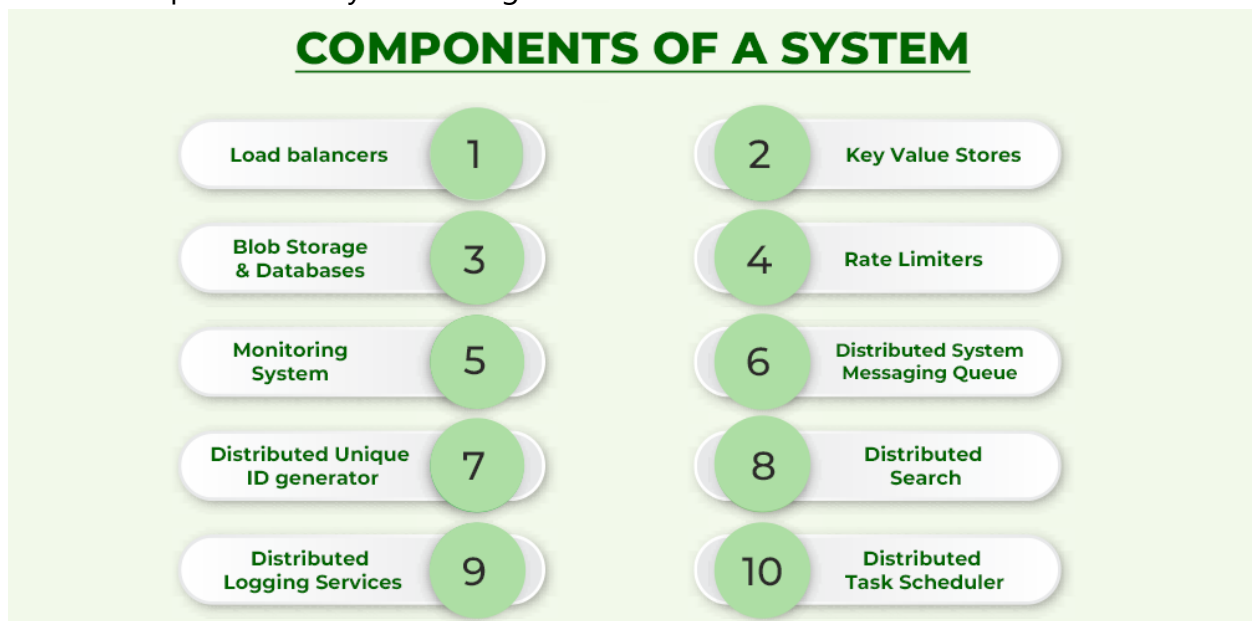
1. **Practicality:** We need a system that should be targeting the set of audiences(users) corresponding to which they are designing.
2. **Accuracy:** Above system design should be designed in such a way it fulfills nearly all requirements around which it is designed be it functional o non-functional requirements.
3. **Completeness:** System design should meet all user requirements
4. **Efficient:** The system design should be such that it should not overuse surpassing the cost of resources nor under use as it will by now we know will result in low thorough put (output) and less response time(latency).
5. **Reliability:** The system designed should be in proximity to a failure-free environment for a certain period of time.
6. **Optimization:** Time and space are just likely what we do for code chunks for individual components to work in a system.
7. **Scalable(flexibility):** System design should be adaptable with time as per different user needs of customers which we know will keep on changing on time.

---

## Components of System Design

**Ans:** Below are some of the major components of the System Design. discussed in brief. The detailed version of this will be discussed in different posts:

1. **Load balancers:** Most crucial component for scalability, availability, and performance measures for systems.
2. **Key Value Stores:** It is a storage system similar to hashtables where key-value stores are distributed hash tables.
3. **Blob Storage:** Blob stands for binary large objects, as the name suggests is storage for unstructured data such as YouTube, and Netflix.
4. **Databases:** It is an organized collection of data so that they can be easily accessed and modified.
5. **Rate Limiters:** These sets the maximum number of requests a service can fulfill.
6. **Monitoring System:** These are basically software where system administrator monitor infrastructures such as bandwidth, CPU, routers, switches, etc.
7. **Distributed System Messaging Queue:** Transaction medium between producers and consumers.
8. **Distributed Unique ID generator:** In the case of large distributed systems, every moment multiple tasks are occurring so in order to distinguish it assign a tag corresponding to every event.
9. **Distributed Search:** Over every website, crucial information that visitors will seek is put into the search bar.
10. **Distributed Logging Services:** Tracing sequences of events from end to end.
11. **Distributed Task Scheduler:** Computational resources such as CPU, memory, storage, etc.## Components of System Design



---

## System Architecture

Software architecture is a way in which we define how the components of a design are depicted design and deployment of software.

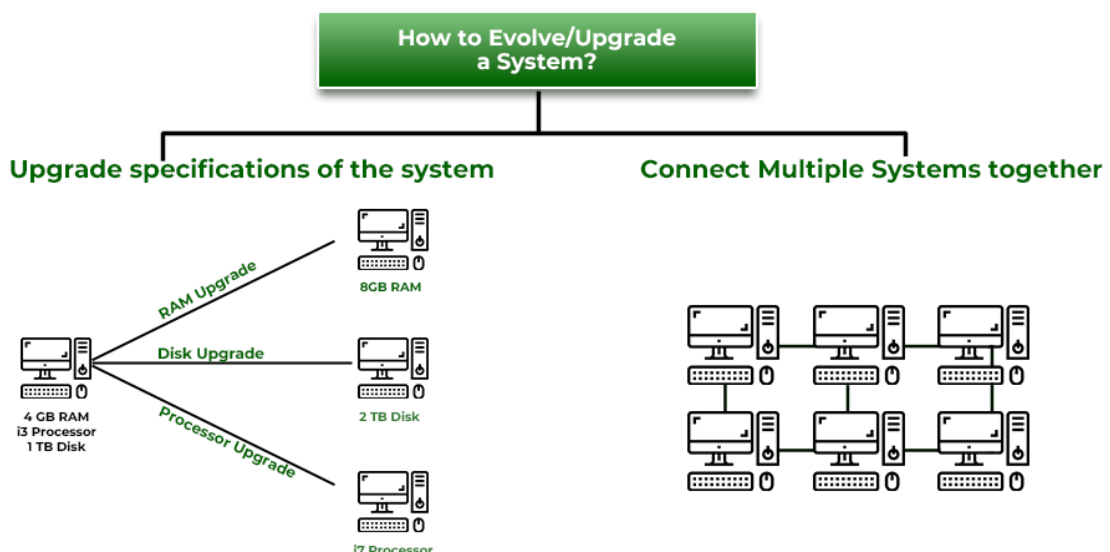
1. Layered Pattern
2. Client-Server Pattern
3. Monolithic Pattern
4. Microkernel Pattern
5. Microservices Pattern

## Evolution/Upgrade/Scale of an Existing System

With the increase in tech usage, be it offline or online, it is now a must for every developer to design and create a **scalable system**. If the system is not scalable, with the increase in users, it is very likely that the system will crash. Hence the concept of scaling comes into play.

Suppose there is a system with configurations of specific disk and RAM which was handling tasks. Now if we need to evolve our system or scale up, we have two options with us.

1. **Upgrade Specifications of existing system:** We are simply improving the processor by upgrading the RAM and disk size and many other components. Note that here we are not caring about the scalability and availability of network bandwidth. Here as per evolution we are working over the availability factor only considering scalability will be maintained. This is known as vertical scaling.
2. **Create a Distributed System by connecting multiple systems together:** We see above that if scalability is not up to mark then we need multiple systems for this measure as availability measures do have a limitation. In order to scale up, we need more systems (more chunks of blocks) and this is known as horizontal scaling.



# Advantages of System Design

- Reduces the design cost of a product.
- Speedy software development process
- Saves overall time in SDLC (software dev life cycle)
- Increases efficiency and consistency of a programmer.
- Saves resources