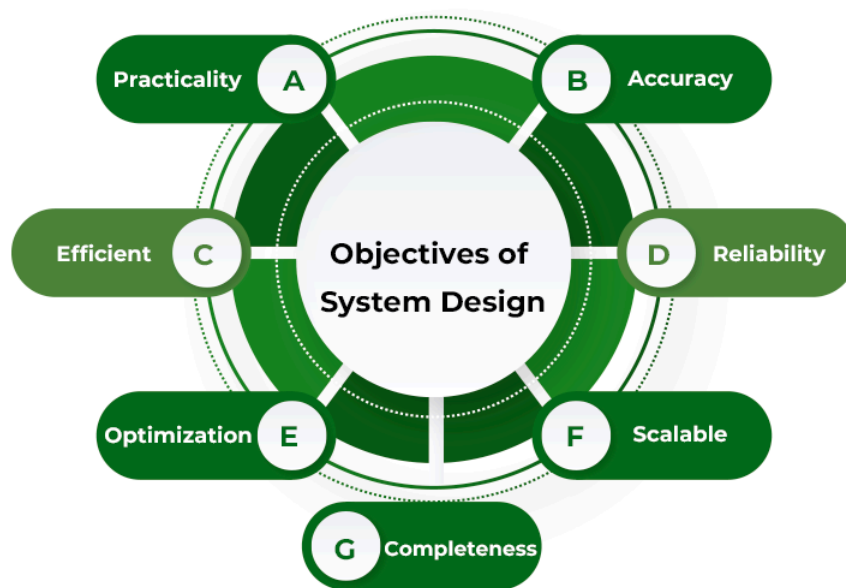


# Goals and Objectives of System Design

---

The objective of system design is to create a plan for a software or hardware system that meets the needs and requirements of a customer or user. This plan typically includes **detailed specifications** for the system, including its architecture, components, and interfaces.

System design is an important step in the development process of any system, as it serves as the foundation for the implementation and deployment of the system. A well-designed system can help ensure that the system is **reliable**, **efficient**, and **user-friendly**.



Objectives of System Design

## Objectives of System Design

---

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 or 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 throughput (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. The best example here out is the well-known firm: Nokia. It is the most important aspect while designing systems and is the result of why 1 of 100 startups succeed over the long run, the best example here out is GeeksforGeeks.

There are several key objectives of system design, including:

1. **Identify the needs and requirements of the user or customer:** The first step in the system design process is to understand the needs and requirements of the user or customer. This involves gathering information about the user's goals, needs, and expectations for the system, as well as any constraints or limitations that need to be taken into account.
2. **Develop a plan for the system:** Once the needs and requirements of the user have been identified, the next step is to develop a plan for the system. This plan should detail the architecture, components, and interfaces of the system, as well as any algorithms, data structures, or other components that will be used.
3. **Ensure that the system is reliable and efficient:** One of the key objectives of system design is to ensure that the system is reliable and efficient. This means designing the system in a way that minimizes downtime and errors and maximizes performance and speed.
4. **Make the system user-friendly:** Another important objective of system design is to make the system user-friendly. This means designing the system in a way that is intuitive and easy to use, with a user interface that is clear and straightforward.
5. **Take into account any constraints or limitations:** In addition to meeting the needs and requirements of the user, the system design must also take into account any constraints or limitations. For example, the system may need to be designed to run on specific hardware or software platforms, or to comply with certain standards or regulations.

**Conclusion:** Designing system's objective is eccentric to create a plan for a system that meets the needs and requirements of the user or customer, while also ensuring that the system is reliable, efficient, and user-friendly. By following a systematic and comprehensive design process, developers can create systems that are well-suited to their intended purpose and users.

**Example:** Design a system for a small e-commerce website that allows customers to browse and purchase products online.

**Step 1:** Identify the functional requirements of the system

- Customers should be able to browse through a catalog of products and view product details.
- Customers should be able to add products to a shopping cart and place an order.

- The system should track the status of each order and send updates to the customer.
- The system should process payments and handle returns and refunds.

**Step 2:** Identify the non-functional requirements of the system

- The system should be able to handle high traffic and large numbers of concurrent users.
- The system should have a fast response time and able to handle rapid updates to the product catalog.
- The system should be secure, with some measure in place to protect against unauthorized access and data breaches.

**Step 3:** Design the high-level architecture of the system

- The system will consist of a web frontend that allows customers to browse and purchase products, and a backend that handles orders, payments, and products management.
- The backend will consist of a database for storing customer and order information, and a set of microservices that handles different functions such as payment processing and other fulfillment.
- The frontend and backend will communicate through API's(Application Programming Interface).

**Step 4:** Design the detailed architecture of system

- The web frontend will be built using a modern web framework such as React or Angular. It will make API calls to the backend to retrieve product information and place orders.
- The backend will consist of a database such as MySQL or PostgreSQL for storing customer and order information. It will also include microservices for payment processing(using a service such as Stripe), order fulfillment, and email notifications. These microservices will communicate with each other and the database through API's.
- The system will also include security measures such as HTTPS encryption, authentication, and authorization.

**Step 5:** Implement and test the system

- The various components of the system will be developed and tested separately before being integrated and tested as a whole.
- The system will be deployed to a production environment, such as cloud platform like AWS(Amazon web services) or GCP, and monitored for performance and reliability.