System Design Interviews

7 types of **non-functional requirements** you must cover.

DesignGurus.org

What is the **difference** between **functional** and **non-functional** requirements?

Functional Requirements

Functional requirements help understand the functions of the system. They define what the system must do.

Example: “Send email when a customer makes a purchase.”

Non-functional Requirements

Non-functional requirements help understand the quality of the system. They describe how the system should behave and what limits there are on its functionality.

Examples: "Emails should be sent with a latency of no greater than 12 hours“ or “Each page must load within 2 seconds.”

So what are the different **types** of **non-functional requirements** you should consider covering in a system design interview?

DesignGurus.org

1. **Availability**

Availability defines the uptime requirements of the system. For how much time a system's functionalities and services are available for use.

Example: "Customer's dashboard must be available 99.98 percent of the time every month during business hours."

2. **Scalability**

The ability for the system to grow and perform as the number of users, requests, or data increases.

Example: "The system must be scalable enough to support one million page visits at any time while maintaining optimal performance.”

3. **Usability**

Usability defines how easy it is for a user to interact with the system. The user should be able to operate the system safely, effectively, and efficiently while enjoying the experience.

Example: "Customer should be able to disable the popup offers."

4. **Extensibility**

Extensibility defines the ability of a system to extend its functionality. This extension can be through adding new functionality or modifying existing functionality.

Example: "The web crawler should be able to parse XML documents in the future."

5. **Resiliency**

Resiliency defines how a system can gracefully handle and recover from accidental or malicious failures.

Example: "Upon a critical failure, the system should be able to roll back within 10 minutes."

6. **Reliability**

Reliability specifies how likely the system will work without failure for a given period.

Example: "The system must perform without failure in 95 percent of use cases during a month.”

7. **Security**

Security requirements define how well the system and its data are protected against attacks or unauthorized access.

Example: "The payment processing gateway must be PCI DSS compliant."

➡ A system can turn a non-functional requirement into a functional requirement by making it a core feature.

➡ Other examples of non-functional requirements are Durability, Maintainability, and Performance.

➡ Preparing for System Design Interview, check "**Grokking the System Design Interview**" at DesignGurus.org

DesignGurus.org