System Design Interviews

7 characteristics that will distinguish you from others.

1. Eliminating vagueness

What does it mean?
- Asking questions to clear the boundary of the system.
- Identifying the limits and scope of the system.

Examples:
- Are we going to support one million users?
- Will there be any push notifications for updates?

2. Being able to ask the right questions

Examples:

- If we are going for lower latency, what minimum throughput should we expect from the system?

- Since we will be storing data for global users, should we store data near the user's location to reduce latency?

3. Making reasonable assumptions

Examples:

- We will prioritize high throughput and a reasonable latency.
- We will adopt a pull data model for less active users and switch to a hybrid (push & pull) model for highly active (or celebrity) users.

4. Identifying and prioritizing the core requirements of the system

Examples:
- Users should be able to upload videos and attach them to their tweets.
- It is important for the system to be highly available, but consistency can take a hit. The user should always be able to access the system, but it's okay if they sometimes see inconsistent data.

5. Analyzing different solutions and tradeoffs

Examples:
- SQL vs. NoSQL
- Availability vs. Consistency
- Throughput vs. Latency
- Pull vs. Push

6. Making reasonable choices

Examples:

- Since we will be storing a massive amount of data, we will partition our data to distribute it to multiple servers.
- Since we need low latency, we'll store the data near the user's location.

7. Identifying and resolving bottlenecks

Examples:

- Is there any single point of failure in our system? What are we doing to mitigate it?
- Do we have enough service replicas running such that a few failures will not cause a total system shutdown?

➡ Practice these techniques to distinguish yourself from others!

➡ These approaches are discussed in detail in "**Grokking the System Design Interview**."