

# How to Succeed in a System Design Interview

System design questions can often be as general as “How would you design Product X?”



Pramp · [Follow](#)

Published in Pramp Blog | Coding Interview & Job Search Resources for Developers ·

7 min read · Aug 13, 2018



3.7K



4





“Oh, you mean what would be the best way to design a product that has had hundreds of software developers working on it for a decade? Sure, let me answer that in 45 minutes or less!”

Real-world system design is challenging and complex, especially for large systems with varying requirements and constraints. Truth be told, an unbelievable amount of complexity can also be found beneath something as simple as visiting Google in your browser. While the end user is blind to much of this complexity, you must face it head-on as a system designer.

Given the complexity of system design, it's impossible to cover all the various topics and tradeoffs in under an hour. Moreover, unlike data structures and algorithms questions, there is more than one optimal solution. Instead, different interviewers can conduct vastly different interviews based on the same question by focusing on various aspects of the system.

Think of a system design interview as a brainstorming session driven by open-ended questions in which you'll be expected to competently discuss a complex distributed system. Consider it an opportunity to work with your interviewer, just as two team members would, to solve a real problem related to the company's goals.

The good news is that you don't need to know everything about system design! Your final design is less important than the thought process behind your design choices. After all, this reflects the experience of actually working at many tech companies. Engineers have a tremendous amount of freedom; we aren't asked to implement fully-specified features but rather to take ownership of open-ended problems and come up with the best solutions.

To develop the skills needed to ace a system design interview, you must familiarize yourself with the topics that system design interview questions might involve and how to approach them. You'll also need to be able to methodically explore different directions, ask the right questions, and understand best practices and common pitfalls of modern distributed systems.

Actual experience with a wide range of tools and systems is an advantage, but being able to identify a need and suggest a standard solution for it would get you a long way, even if you've never used it yourself. For instance,

recognizing that your distributed system needs load balancing and naming NGINX as a popular choice is more important than being able to drill down into the details of how to configure one product or another to perform as a load balancer.

In these cases, the magic ingredient is honesty — you should always be confident in saying, “While I’ve never used technology X, I know it’s a common solution to problem Y.” The combination of honesty, confidence, and a willingness to learn will leave a much better impression on your interviewer than throwing around incoherent tidbits about a product you’ve



Search Medium



Write



BLACKBOX AI



## **A Step-by-Step Approach to Acing your System Design Interview**

While every system design interview is different, there are some common steps you should cover, even if the conversation might not be as sequential as your ideal thought process.

The system design interview is an open-ended conversation you'll be expected to lead. Try using the following steps to guide your discussion:

## Step 1 — Understand the Goals

Clarifying ambiguities early in the system design interview is critical.

Candidates who spend time clearly defining the end goals of the system have a better chance of success than those who don't. For that reason, ensure you understand the basic requirements and ask clarification questions. Start with the most basic assumptions:

- What is the goal of the system?
- Who are the users of the system? What do they need it for? How are they going to use it?
- What are the inputs and outputs of the system?

Even if you're asked about a well-known product, you should still share your assumptions about it with your interviewer. You may find that you and your interviewer don't have the same assumptions about products like Twitter, Facebook, or Reddit. In addition to helping you focus, it demonstrates product sensibility and good teamwork.

## Step 2 — Establish the Scope

Now that you understand the system, you should describe the feature set you'll discuss. Try to define all the features and all the components you think

of in order of importance to the user. You don't have to get it right on your first attempt, but make sure you and your interviewer agree.

Ask clarifying questions, such as:

- Do we want to discuss the end-to-end experience or just the API?
- What clients do we like to support (mobile, web, etc.)?
- Do we require authentication? Analytics? Integrating with existing systems?

Take a few minutes to discuss this with your interviewer, and write it down.

### **Step 3 — Design for the Right Scale**

The same feature set requires a very different approach for different scales. It's essential to determine the scale to know whether your data can fit on one machine or whether you need to scale the reads. You might ask:

- What is the expected read-to-write ratio?
- How many concurrent requests should we expect?
- What's the average expected response time?
- What's the limit of the data we allow users to provide?



Different answers require very different designs, so getting the scale right is key to success.

#### **Step 4 — Start High-Level, then Drill-Down**

Start with covering the end-to-end process based on the goals you've established. This might include detailing different clients, APIs, backend services, offline processes, network architecture, and data stores and how they all come together to meet the requirements.

This is also an excellent point to identify the system's entry points, such as:

- User interaction
- External API calls
- Offline processes

This will allow the conversation to drill down into potential performance bottlenecks and decisions about separating responsibilities. Whichever approach you choose to start with, remember to always **start simple and iterate**.

#### **Step 5 — Data Structures and Algorithms (DS&A)**

Wait, this again? Even in a system design interview? Yes! Turns out, this is actually important in designing software systems.

URL shortener? Makes me think of a hashing function. Oh, you need it to scale? Sharding might help. Concurrency? Redundancy? Generating keys becomes even more complicated. The same goes for designing an analytics system, a news feed, or a Q&A forum, each with its own set of common DS&A.

Don't forget to account for your scaling requirements, where analyzing runtime and memory complexity becomes handy.

## **Step 6 — Tradeoffs**

When it comes to system design interviews, almost every decision you'll need to make will involve a tradeoff. Being able to describe them in real-time, as you're suggesting solutions, shows that you understand that complex systems often require compromises and allow you to demonstrate your knowledge regarding the pros and cons of different approaches. Since there's no one correct answer, having this discussion will give your interviewer the impression that you're practical and will use the right tool for the job.

A few common examples might include the following:

- What type of database would you use and why?

- What caching solutions are out there? Which would you choose and why?
- What frameworks can we use as infrastructure in your ecosystem of choice?

Practice system design interviews on Pramp →

## Practice is Key

While you can definitely hone down the theory by yourself, the last piece of the puzzle is practice. Make sure to apply these steps time and time again, answering questions about modern system design with real peers.

It's the combination of knowledge, technique, and practice, that'll enable you to land your dream job.

To see you off, check out this system design interview example from our friends at [Exponent](#):

## Additional Resources

### Articles/Blogs:

- [Exponent's System Design Interview Prep Course](#) — An online course with mock interview videos and lessons to help you ace the system design interview
- [Exponent's Engineering Manager Interview Prep Course](#) — The world's first ever engineering manager interview course

- [System Design Cheat Sheet](#) — A guide on the key topics within system design
- [The Log: What every software engineer should know about real-time data's unifying abstraction](#) — A lengthy article about logs and tradeoffs
- [High Scalability](#) — A blog about how real-world systems are designed
- [The Architecture of Open Source Applications \(Volume 2\): Scalable Web Architecture and Distributed Systems](#) — An article about some of the key issues to consider when designing large websites
- [Distributed Systems Reading List](#) — A list of articles on distributed systems to help change the way you think

### **Videos:**

- [System Design Mock Interview: Instagram](#) — YouTube walkthrough of a System Design mock interview about designing Instagram
- [System Design Mock Interview: Facebook Messenger](#) — YouTube walkthrough of a System Design mock interview about designing Facebook
- [Intro to Architecture and Systems Design Interviews](#) — YouTube tutorial made by an ex-Facebook engineer about system design interviews

- System Design Introduction For Interview. — YouTube tutorial made by Tushar Roy about system design interviews

. . .

→ **The Complete System Design Interviewer Guide**

Interview

Interviewing

Interview Questions

Interview Tips

Coding



**Written by Pramp**

3K Followers · Editor for Pramp Blog | Coding Interview & Job Search Resources for Dev...

The Pramp Team

Follow





✓ Pr... in Pramp Blog | Coding Interview & Job Search...

## The Anatomy Of A System Design Interview

System design is the process of defining the architecture, components, data, interfaces,...

2 min read · Aug 13, 2018

👏 112 💬

🔖+ ...

✓ Pr... in Pramp Blog | Coding Interview & Job Search...

## How to Ace Front-End Interview Questions

Front-end developer interviews are particularly challenging because you need t...

12 min read · Aug 14, 2018

👏 2.4K 💬 4

🔖+ ...

✓ Pr... in Pramp Blog | Coding Interview & Job Searc...

## How to Succeed in Data Structures and Algorithms Interview

To develop the skills needed to ace a DS&A interview, you need to familiarize yourself wi...

12 min read · Nov 8, 2018

 388  1

See all from  
Pramp



✓ Pr... in Pramp Blog | Coding Interview & Job Searc...

## How to Succeed in A Data Science Interview

Most data science interviews are unable to explore the wide range of topics necessary t...

14 min read · Oct 25, 2018

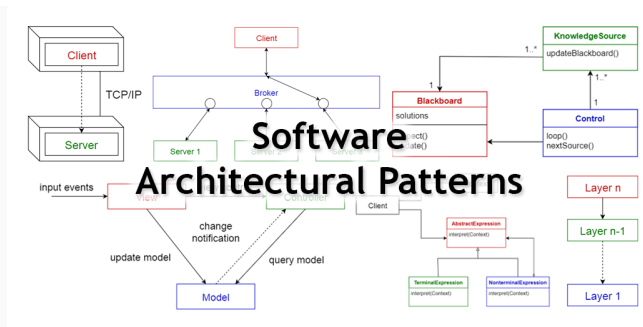
 248  5

See all from Pramp Blog | Coding Interview & Job Search Resources for  
Developers

## Recommended from Medium





Vijini Mallawaarachchi in Towards Data Science

## 10 Common Software Architectural Patterns in a nutshell

Ever wondered how large enterprise scale systems are designed? Before major softwa...

★ · 5 min read · Sep 4, 2017



39K



126



Arslan Ahmad in Level Up Coding

## Don't Just LeetCode; Follow the Coding Patterns Instead

What if you don't like to practice 100s of coding questions before the interview?

5 min read · Sep 16, 2022



14



### Lists



#### Stories to Help You Grow as a Software Developer

19 stories · 295 saves

#### The New Chatbots: ChatGPT, Bard, and Beyond


13 stories · 91 saves

#### General Coding Knowledge

20 stories · 235 saves

#### Icon Design

30 stories · 59 saves

 Anthony D. Mays

## How to Practice LeetCode Problems (The Right Way)


tl;dr: You're doing it wrong. Use "The Six Steps" any time you practice LeetCode...

★ · 13 min read · May 10, 2022



9



 Denat Hoxha

## Sharing Data Between Microservices


Robust distributed systems embrace eventual consistency to share data between...

7 min read · Oct 24, 2022




26



 Damian Gryski

## Consistent Hashing: Algorithmic Tradeoffs

 Adham Dannaway in UX Planet

## 16 little UI design rules that make a big impact

Like this article? Buy me a coffee.

12 min read · Apr 2, 2018



22



A UI design case study to redesign an example user interface using logical rules or...

14 min read · Mar 14



110



See more recommendations

[Help](#) [Status](#) [Writers](#) [Blog](#) [Careers](#) [Privacy](#) [Terms](#) [About](#) [Text to speech](#) [Teams](#)