Google

Preparing for the System Design Interview

What is a system design question and why is it important?

System design questions are used to assess a candidate's ability to combine knowledge, theory, experience and judgement toward solving a real-world engineering problem. Sample topics include feature sets, interfaces, class hierarchies, constraints, simplicity, robustness and tradeoffs. The interview will assess your deep understanding of how the internet works and familiarity with the various pieces (routers, domain name servers, load balancers, firewalls, etc.).

Interview Tips and Expectations:

1. The system design question will ask you to take an abstract question in a formerly unseen problem and present a high level framework to solve the presented problem. In doing so, follow these steps:
    - Gather key requirements
    - Identify key design elements
    - Identify tradeoffs of different decisions
    - Dive deep into a specific sub-problem AND/OR demonstrate what process to use to arrive at a solution

2. Identify the major components of an overall system design and deeply describe at least two of them. Suggest which technologies and systems should be used to solve a relatively common problem (i.e. relational database, document-based database, etc.), and if relevant, share a story about a time you have solved a problem using these technologies.

3. When defining a system, include as many non-functional requirements as possible. For example: performance, latency, legal, privacy, maintainability, capacity, security, geographic, and cross-DC consideration.

4. Develop a framework you can use to answer most questions.

5. **Approach the problem with an open mind:** Be mindful that your familiarity (if any) with the problem may make you take shortcuts (even verbal ones when explaining your approach); the interviewer may not share the same context as you. If there is a more succinct approach, feel free to explain to the interviewer the route you're taking and the context behind why.

6. **Problem Solve:** The interviewers are looking for problem solving and approach as much as they are assessing your solution. So, ask clarifying questions about the problem, express ideas verbally, and think out loud. Constantly challenge your own design. Be prepared to justify every decision you make. Practice going through problems with a friend.

7. **Listen:** Design questions are typically open-ended and the problem may be ambiguous. If the interviewer gives a hint or asks a question while you are solving the problem, listen intently and do not ignore. They are most likely trying to guide you or looking for a particular awareness.

Google

Role-related design questions

Due to the wide variety of projects we offer at Google, it is important our design questions are relevant to both your background and the role you will be exploring. Be sure to discuss with your recruiter the role you're pursuing and which system design question is best suited to assess your skills. Depending on your area of expertise, one of the following types of design questions will be asked:

Distributed System Design

System design is a general category which means "big enough systems that you have to operate on a high level of abstraction to get the basic architecture down." A distributed system question is related to problems of coordination between autonomous systems while system design can include problems constrained to a single execution.

Distributed systems questions focus on cross-task coordination, communication, synchronization, and failure modes. System design questions are more weighted toward managing the complexity of large bodies of software. You can find some of the Google research around distributed systems here.

Example Distributed System Design: Design a distributed unique ID service.

Low Level System Design

This type of system design question is typically asked when exploring our embedded software engineering roles here at Google. Expect a very broad question that doesn't have a single right answer but can go in a number of directions. Your interviewer will probably drill down in more detail in a few areas.

A low level system design interview covers areas of design that span issues of scaling, details of integration with hardware and integration with higher level components. You might come across questions that relate to OTAs, boot, reliability, power management, the kinds of issues that any low-level team will be dealing with on a constant basis. The interview will assess understanding of high level concepts, file storage, encryption, signing, how devices boot, how disks are laid out, how power management functions, etc.

Example Low Level Design: Design an automated failover system for a replicated database.

Machine Learning Design

Candidates who have experience with Machine Learning technologies may be asked a Machine Learning design question. These questions are generally open ended so you can dive deeper into the solution. It is important to design a solution thinking about data analysis, potential use cases, the users, scalability, limitations and tradeoffs. The Machine Learning Design question may focus on one specialization within machine learning. These may include computer vision, deep learning/neural networks, recommendations/ranking systems, natural language processing, speech/audio or reinforcement learning.

Example Machine Learning Design: Design a near-duplicate image detection system.

Mobile Application System Design

Mobile application design questions are targeted toward mobile focused candidates (iOS and Android developers). A mobile application design interview gives the candidate a chance to display their design skills within this domain. The goal of a design interview is to assess your

ability to combine knowledge, theory, experience and judgement toward solving a real-world engineering problem. Interviewers may ask the candidate to specify the design of an entire mobile application, subsystems within a specific application or a solution to a difficult aspect of mobile application development.

Reflect on the design decisions you've made in the past and think about the justifications and rationale for those design details. Study any published and available designs for mobile application problems. Think in some detail about how you would architect some of the applications that you use often and are familiar with. Practice with top-down and bottom-up design approaches and consider how various layers of a system interact with each other.

Example Application System Design: How would you speed up the startup time for the YouTube app?

Front End UI Web Design

A front end design interview will typically involve problems that require an understanding of key front end concepts such as web serving systems, client-side technologies and languages, and common design patterns for UI development.

The problem analysis could involve a break-down into sub-problems, identification of areas of risk, or a discussion of requirements and goals. The solution could involve (but not limited to!) system diagrams, state diagrams, data-structures, or algorithms. Explaining the solution crisply is as important as coming up with the solution, and informs us how well candidates are able to collaborate. Code may be involved to describe a solution, but this may not be the objective. In some cases, a portion of a design problem may translate into a coding a problem.

Example Front End UI Web Design: Design a web system to sell concert tickets.

Interview Resources

Books
Cracking the Coding Interview
Gayle Laakmann McDowell
Programming Interviews Exposed: Secrets to Landing Your Next Job
John Mongan, Eric Giguere, Noah Suojanen, Noah Kindler
Programming Pearls
Jon Bentley
Introduction to Algorithms
Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein

Interview Prep
How we hire
Interviewing @ Google
Candidate Coaching Session:Tech Interviewing
CodeJam: Practice & Learn
Technical Development Guide

About Google
Company - Google
The Google story
Life @ Google
Google Developers
Open Source Projects
Github: Google Style Guide

Google Publications
The Google File System
Bigtable
MapReduce
Google Spanner
Google Chubby