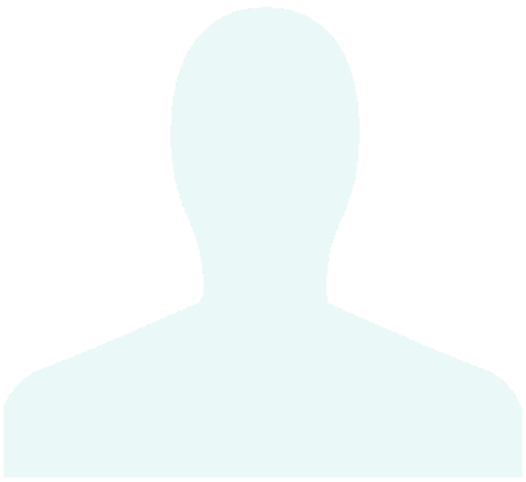
Grokking the System Design Interview
Search

LearnTeach



- My Profile
  - View
  - Edit
- Logout

Why System Design Interviews?

1. What are system design interviews?

In system design interviews, candidates are required to show their ability to develop a high-level architecture of a large system. Designing software systems is a very broad topic and even a software engineer having years of experience at a top software company may not claim to be an expert on system design. During such interviews, candidates have 30 to 40 minutes to answer questions like, "How to design a cloud storage system like Dropbox?" or "How to design a search engine" etc. In real life, companies spend not weeks but months and hire a big team of software engineers to build such systems. Given this, then how can a person answer such a question in 40 minutes? Moreover, there is no set pattern of such questions. Questions are flexible, unpredictable, usually open-ended, and have no standard or squarely correct answer.

Unlike coding interviews where problem solving ability of the candidates is evaluated, design interviews consist of complicated and fuzzy questions which aim at testing the candidate's ability to analyze a vague and complicated problem, their compatibility with building large systems and how do they present their solution. Such interviews also look into a candidate's competence in guiding and moving the conversation forward.

These days, companies are least bothered about your pedigree, where you studied or where you come from but surely concerned about what you can do on the job. For them, the most important thing is your thought process and your mindset to look into and handle problems. For these counts, candidates are generally scared of the design interviews. But despite all this, I believe there is no need of scaring off. You need to get into what the companies want to know about you during these 40 minutes, which is basically "your approach and strategy to handle a problem" and how organized, disciplined, systematic, and professional you are at solving it. What is your capacity to analyze an issue and your level of professional mechanics to solve it step by step?

In short, system design interview is, just understanding it from interviewer's perspective. During the whole process, it is your discussion with the interviewer that is of core importance.

2. How to give system design interview?

There is no strictly defined process to system design interview. Secondly, there are so many things inherently unclear about large systems that without clarifying at least a few of them in the beginning, it would be impossible to go for a solution. Any candidate who does not realize this fact would fall into the trap of quickly jumping onto finding a solution.

For instance, the questions can be like:

- Design a URL shortening service like TinyURL.
- How would you build a social network like Facebook and implement a feature where one user receives notifications when their friends 'like' the same things as they do?
- How to design a ride-sharing service like Uber, which connects passengers who need a ride with drivers who have a car.

Any candidate who does not have experience in building systems might think such question grossly unfair. On top of that, there generally isn't any one correct answer to such questions. The way you are answering the question would sufficiently tell upon your professional competence and background experience. That is the thing which the interviewer will evaluate you on.

Since the questions are intentionally weakly defined, jumping into designing the solution immediately without understanding them properly is liable to get you in trouble. Spend a few minutes questioning the interviewer to comprehend the full scope of the system. Never assume things that are not explicitly stated. For instance, the "URL shortening service" could be serving just a few thousand users, but each could be sharing millions of URLs. It could also mean to handle millions of clicks on the shortened URLs or just a few dozens. The service may also require providing extensive statistics about each shortened URL (which will increase our data size), or statistics may not be a requirement at all. Therefore, don't forget to make sure you gather all the requirements as the interviewer would not be listing them out for you in advance.

The point I want to make is that the main difference between design interviews and the rest is that you are not presented with the full detail of the problem at the outset. Rather you are required to scale the breadth and depth of a blurred and indistinct problem. You are supposed to take the details and interrogate the crux of the issue by putting judicious questions yourself. Your questions and points of interest to clarify the problem presented go a long way in evaluating your ability and competence as an asset to the company.

In design and architecture interviews the problems presented are quite big. They definitely cannot be solved in 40 minutes' time implying that the objective is to test the technical depth and diversity the interviewee invokes during the interview. That also speaks strongly for your would be 'level' in the company. And your level in the company should come from your analytical ability to sort out the problem besides your ability to work in a team (your behavioral and background side of the interview), and your capacity to perform as a strong technical leader. In a nutshell, the basic idea of hiring at a level is to scale a person's ability to contribute value to the company's wants and needs. For that, you must exhibit your strengths by showing reasonable technical breadth.

Try to learn from the existing systems: How have these been designed? Another important point to be kept in mind is that, the interviewer expects that candidate's analytical ability and questioning on the problem must comparable to his/her experience. If you have a few years of software development experience, you are expected to have certain knowledge and should avoid divulging into asking basic questions that might have been appropriate coming from a fresh graduate. For that, you should prepare sufficiently ahead of time. Try to go through real projects and practices, well in advance of the interview as most questions are based on real-life products, issues and challenges.

Leading the conversation: It is not the ultimate solution to the problem, rather the discussion process itself that is important in the interview. And it is the candidate who should lead the conversation going both broad and deep into the components of the problem. Hence, take the interviewer along with you during the course of solving the problem by communicating with him/her step by step as you move along

Solving by breaking down: Design questions at first might look complex and intimidating. But whatever the complexity level of the problem, a top-down and modularization approach can help a lot in solving the problem. Therefore, you should break the problem into modules and then tackle each of them independently. Subsequently, each component can be solved as a sub problem by reducing it to the level of a known algorithm. This strategy will not only make the design much clearer to you and your interviewer but make evaluation much easier for the interviewer. However, while doing so, keep this thing in mind that mostly the problems presented in high skill design interviews don't have the solutions. The most important thing is the way how you make progress tackling the problem, and the strategies you make use of.

Dealing with the bottlenecks: Working on the solution, you might confront some bottlenecks. This is very normal. When confronting bottlenecks, your system might require a load balancer with many machines behind it to handle the user requests or the data might be so huge that you need to distribute your database on multiple servers. It might also be possible that the interviewer wants to take the interview in a particular direction. If that is the case, you are supposed to move in that direction and should go deep leaving everything else aside. If you feel stuck somewhere, you can ask for a hint so that you may keep going. Keep in mind that each solution is a kind of trade-off; hence, changing something may worsen something else. Here, the important thing is your ability to talk about these trade-offs and to measure their impact on the system keeping all the constraints and use cases in mind. After finishing with your high-level design and making sure that the interviewer is ok with it, you can go for making it more detailed. Usually, that means making your system scale.

3. Summary

Solving system design questions could be broken down into three steps:

- **Scoping the problem:** Don't make assumptions; Ask clarifying questions to understand the constraints and use cases.
- **Sketching up an abstract design** Illustrating the building blocks of the system and the relationships between them.
- **Identifying and addressing the bottlenecks** by using the fundamental principles of scalable system design.

4. Conclusion

Design interviews are formidable, open-ended problems that cannot be solved in the allotted time. Therefore, you should try to understand what your interviewer intends to focus on and spend sufficient time on it. Be well aware of the fact that the discussion on system design problem could go in different directions depending on the preferences of the interviewer. The interviewers might be unwilling to see how you create a high-level architecture covering all aspects of the system or they could be interested in looking for specific areas and diving deep into them. This means that you must deal with the situation strategically as there are chances of even the good candidates failing the interview, not because they don't have the knowledge, but because they lack the ability to focus on the right things while discussing the problem.

If you have no idea how to solve these kinds of problems, you can familiarize yourself with the common patterns of system design by reading diversely from the blogs, watching videos of tech talks from conferences. It is also advisable to arrange discussions and even mock interviews with experienced engineers at big tech companies.

Remember there is no ONE right answer to the question because any system can be built in different ways. **The only thing that is going to be looked into is your ability to rationalize ideas and inputs.**

Mark as completed
Next →System Design Basics
Send feedback or ask a question

32 recommendations

- Home
- Featured
- Team
- Blog
- FAQ
- Terms of Service
- Contact Us