Why System Design Interviews?

1. What are system design interviews?

In system design interviews, candidates are required to show their ability to d
of a large system. Designing software systems is a very broad topic and even a
of experience at a top software company may not claim to be an expert on sys
interviews, candidates have 30 to 40 minutes to answer questions like, "How t
like Dropbox?" or "How to design a search engine" etc. In real life, companies
and hire a big team of software engineers to build such systems. Given this, th
such a question in 40 minutes? Moreover, there is no set pattern of such quest
unpredictable, usually open-ended, and have no standard or squarely correct

Unlike coding interviews where problem solving ability of the candidates is ev
consist of complicated and fuzzy questions which aim at testing the candidate
complicated problem, their compatibility with building large systems and how

but surely concerned about what you can do on the job. For them, the most im
process and your mindset to look into and handle problems. For these counts,
of the design interviews. But despite all this, I believe there is no need of scari
the companies want to know about you during these 40 minutes, which is bas
strategy to handle a problem" and how organized, disciplined, systematic, and
it. What is your capacity to analyze an issue and your level of professional me

In short, system design interview is, just understanding it from interviewer's p
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
unclear about large systems that without clarifying at least a few of them in th
impossible to go for a solution. Any candidate who does not realize this fact w
jumping onto finding a solution.

For instance, the questions can be like:

- Design a URL shortening service like TinyURL.

- How to design a ride-sharing service like Uber, which connects passenge
who have a car.

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

Since the questions are intentionally weakly defined, jumping into designing t
without understanding them properly is liable to get you in trouble. Spend a f
interviewer to comprehend the full scope of the system. Never assume things
instance, the "URL shortening service" could be serving just a few thousand us
millions of URLs. It could also mean to handle millions of clicks on the shorter
The service may also require providing extensive statistics about each shorter
data size), or statistics may not be a requirement at all. Therefore, don't forget
requirements as the interviewer would not be listing them out for you in adva

The point I want to make is that the main difference between design interview
not presented with the full detail of the problem at the outset. Rather you are

problem presented go a long way in evaluating your ability and competence a

In design and architecture interviews the problems presented are quite big. T
in 40 minutes' time implying that the objective is to test the technical depth ar
invokes during the interview. That also speaks strongly for your would be 'lev
level in the company should come from your analytical ability to sort out the p
work in a team (your behavioral and background side of the interview), and y
strong technical leader. In a nutshell, the basic idea of hiring at a level is to sc
contribute value to the company's wants and needs. For that, you must exhibi
reasonable technical breadth.

Try to learn from the existing systems: How have these been designed? Ano
in mind is that, the interviewer expects that candidate's analytical ability and
must comparable to his/her experience. If you have a few years of software de
expected to have certain knowledge and should avoid divulging into asking ba
been appropriate coming from a fresh graduate. For that, you should prepare
to go through real projects and practices, well in advance of the interview as r
real-life products, issues and challenges.

Leading the conversation: It is not the ultimate solution to the problem, rathe

of solving the problem by communicating with him/her step by step as you m

Solving by breaking down: Design questions at first might look complex and i
complexity level of the problem, a top-down and modularization approach ca
problem. Therefore, you should break the problem into modules and then tac
Subsequently, each component can be solved as a sub problem by reducing it
algorithm. This strategy will not only make the design much clearer to you an
evaluation much easier for the interviewer. However, while doing so, keep th
problems presented in high skill design interviews don't have the solutions. T
way how you make progress tackling the problem, and the strategies you mak

Dealing with the bottlenecks: Working on the solution, you might confront s
normal. When confronting bottlenecks, your system might require a load bala
behind it to handle the user requests or the data might be so huge that you ne
on multiple servers. It might also be possible that the interviewer wants to tak
direction. If that is the case, you are supposed to move in that direction and sh
else aside. If you feel stuck somewhere, you can ask for a hint so that you may
each solution is a kind of trade-off; hence, changing something may worsen so
important thing is your ability to talk about these trade-offs and to measure th
keeping all the constraints and use cases in mind. After finishing with your hi

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
and use cases.
- **Sketching up an abstract design** Illustrating the building blocks of the s
between them.
- **Identifying and addressing the bottlenecks** by using the fundamental p
design.

4. Conclusion

Design interviews are formidable, open-ended problems that cannot be solved
you should try to understand what your interviewer intends to focus on and s
well aware of the fact that the discussion on system design problem could go i
on the preferences of the interviewer. The interviewers might be unwilling to
architecture covering all aspects of the system or they could be interested in l

lack the ability to focus on the right things while discussing the problem.

It is also advisable to arrange discussions and even mock interviews with exp
companies.

Remember there is no ONE right answer to the question because any system c
The only thing that is going to be looked into is your ability to rationalize i

Mark as completed

Next →
System Design Basics