

System Design Approach

System Design

General Approach and Strategy

Goals

- ▶ Present general set of tools to tackle system design question.

System Design

General Approach and Strategy

Goals

- ▶ Present general set of tools to tackle system design question.
- ▶ Tactics to leverage current knowledge and apply to system design.

System Design

General Approach and Strategy

Goals

- ▶ Present general set of tools to tackle system design question.
- ▶ Tactics to leverage current knowledge and apply to system design.
- ▶ Provide further resources and information on system design interviews.

Attributions

- ▶ Content based on Crack the System Design interview: tips from a Twitter software engineer

Clarification

- ▶ You are given a purposefully vague question. Part of the assessment of the interviewer is to see how well you handle vague requirements.

Clarification

- ▶ You are given a purposefully vague question. Part of the assessment of the interviewer is to see how well you handle vague requirements.
- ▶ Similar to a more standard technical interview question, you should clarify ambiguous points with your interviewer.

Clarification

- ▶ You are given a purposefully vague question. Part of the assessment of the interviewer is to see how well you handle vague requirements.
- ▶ Similar to a more standard technical interview question, you should clarify ambiguous points with your interviewer.
- ▶ This gives them a good picture as to how you will deal with these scenarios on the job.

Asking Good Questions

- ▶ The purpose of asking clarifying questions is to dispel the ambiguity and to uncover an actionable piece of the puzzle.

Asking Good Questions

- ▶ The purpose of asking clarifying questions is to dispel the ambiguity and to uncover an actionable piece of the puzzle.
- ▶ A good question will give direction and determine the needs of the customer.

Asking Good Questions

- ▶ The purpose of asking clarifying questions is to dispel the ambiguity and to uncover an actionable piece of the puzzle.
- ▶ A good question will give direction and determine the needs of the customer.
- ▶ Implementing something completely different than what the customer wants certainly has bad implications for the company. Try to whittle away at the problem statement until the scope, motivation, and direction is clear.

Asking Good Questions

- ▶ The purpose of asking clarifying questions is to dispel the ambiguity and to uncover an actionable piece of the puzzle.
- ▶ A good question will give direction and determine the needs of the customer.
- ▶ Implementing something completely different than what the customer wants certainly has bad implications for the company. Try to whittle away at the problem statement until the scope, motivation, and direction is clear.
- ▶ It is better to ask a question that seems trivial than to assume the intent and as a result implement the wrong solution for the problem statement.

Understanding the Problem

- ▶ Ensure that prior to proposing any solution or code that you understand the problem statement and can reiterate it back to the interviewer.

Understanding the Problem

- ▶ Ensure that prior to proposing any solution or code that you understand the problem statement and can reiterate it back to the interviewer.
- ▶ Furthermore, make sure that you have a solid grasp on the constraints, bottlenecks, scope, goals, etc. of the problem you are attempting to solve.

Understanding the Problem

- ▶ Ensure that prior to proposing any solution or code that you understand the problem statement and can reiterate it back to the interviewer.
- ▶ Furthermore, make sure that you have a solid grasp on the constraints, bottlenecks, scope, goals, etc. of the problem you are attempting to solve.
- ▶ Understanding all of these components will make solving the problem itself much less daunting.

Focusing Scope

- ▶ Depending on your background if asked to design a large software service, like Facebook for instance, your response will most likely be based on your area of expertise.

Focusing Scope

- ▶ Depending on your background if asked to design a large software service, like Facebook for instance, your response will most likely be based on your area of expertise.
- ▶ For instance, if you are a front-end developer, you may spend more time on designing the user interface and if you are a back-end developer, you may decide to speak more on that side. The focus also of course depends on the role to which you are applying for.

Focusing Scope

- ▶ Depending on your background if asked to design a large software service, like Facebook for instance, your response will most likely be based on your area of expertise.
- ▶ For instance, if you are a front-end developer, you may spend more time on designing the user interface and if you are a back-end developer, you may decide to speak more on that side. The focus also of course depends on the role to which you are applying for.
- ▶ Since system design interviews are generally given to candidates who have had previous experience, you can showcase your specialty and make a strong case for what your skills specifically bring to the table.

Tackling the Problem

- ▶ Once you and the interviewer are on the same page as to what the goals, scope, etc. of the project is, it's time to drill down and be more specific with how you will go about solving the problems at hand.

Tackling the Problem

- ▶ Once you and the interviewer are on the same page as to what the goals, scope, etc. of the project is, it's time to drill down and be more specific with how you will go about solving the problems at hand.
- ▶ Typically, abstracting certain pieces of functionality into some type of high-level API is a way in which to organize the major components of the system.

Tackling the Problem

- ▶ Being able to intelligently talk about the trade-offs of different database technology is a major plus. For instance, knowing the difference between a SQL and NoSQL database and why you might want to use either one for the task at hand helps guide your design process.

Tackling the Problem

- ▶ Being able to intelligently talk about the trade-offs of different database technology is a major plus. For instance, knowing the difference between a SQL and NoSQL database and why you might want to use either one for the task at hand helps guide your design process.
- ▶ The technologies of SQL and NoSQL in their entirety can be daunting, but for the purpose of this interview, unless you're interviewing for a database position, it will typically be fine to just have a high-level understanding of the technology, trade-offs, etc.

Further Resources

- ▶ Sam of Byte-by-Byte has a great video 3 Ways to Ace Your System Design Video
- ▶ Sam has been a guest on this channel, and you can check out a podcast we did together.
- ▶ Simply searching in YouTube for “system design interviews” yields a large
- ▶ If you prefer book content, the best book I have seen on system design is Designing Data-Intensive Applications written by Martin Kleppmann.