SYSTEM DESIGN GUIDE

What this guide is and whom it's for

Candidates often get overwhelmed with system design. I don’t blame them. **There are literally 100 of topics you can study while preparing for an interview. Does that mean you should stop everything and study all of them? No**, not at all. It’s important to **learn the basics** really well.

Learning the basics well is very important in system design. If you're new to the topic, studying too much in a short time can actually hurt your chances in an interview. At the start, knowing more and having more options can be helpful, but it might also overwhelm you with too many possible answers, making you overthink and choose complicated solutions.

We’ve found that 80% of system design interviews focus on just 20% of the concepts. This guide will cover those key concepts in detail to help you perform better in your interviews.

It's designed for back end-focused engineers preparing for mid to senior-level roles. Whether you’re new to system design interviews or an experienced interviewer looking to sharpen your skills, this guide has you covered.

How we made this guide

We created this guide by diving deep into the world of system design. We started by going through over 80 hours of real system design interviews and lessons, looking for common patterns and challenges people face. To make it even better, we collected feedback from 100+ top interviewers with lots of experience.

Then, we tested our ideas with engineers who were new to system design. These mock interviews helped us improve the guide, making sure it's practical, easy to understand, and helpful.

Every topic in this guide is chosen carefully, focusing on what you're most likely to see in 80% of mid to senior-level interviews. It's not just theory-it's a practical resource built by experts who know both sides of the interview process.

This guide is more than just information, it's a tool designed to help you succeed, tested and proven in the real world.

How to use this guide

This guide is structured in 3 parts

1. System Design Key topics to understand.
2. How to approach a system design interview question.
3. Design 100+ popular systems from scratch, and learn how to get unstuck.

For Interview Do I need to know everything?

No, You don't need to know everything to prepare for the interview.

What you're asked in a system design interview depends on several factors, such as:

1. Your experience level
2. Your technical background
3. The role you're applying for
4. The company you're interviewing with
5. Luck

Experienced candidates are usually expected to know more about system design. For example, architects or team leads may need a deeper understanding than individual contributors. Big tech companies often include one or more system design rounds in their interviews.

A good approach is to start with broad knowledge and dive deeper into a few key areas. Learn a little about important system design topics and adjust your preparation based on your time, experience, and the job you're applying for.

- **Short timeline:** Aim for **breadth** with system design topics. Practice by solving **some** interview questions.
- **Medium timeline:** Aim for **breadth** and some **depth** with system design topics. Practice by solving **many** interview questions.
- **Long timeline:** Aim for **breadth** and more **depth** with system design topics. Practice by solving **most** interview questions.

Topics	Short	Medium	Long
Learn the basics: Read about system design topics to understand how systems work.	Yes	Yes	Yes
Explore company blogs: Check out engineering blogs of the companies you're interviewing with.	Some	Some	Yes
Study real-world examples: Look at real-world system architectures to see how they're designed.	Some	Yes	Yes
Understand the process: Learn how to approach a system design interview question.	Yes	Yes	Yes
Work through **System Design Interview Questions** with solutions	Some	Many	Most
Work through **Object-oriented Design Interview Questions** with solutions	Some	Many	Most
Review **Additional System Design Interview Questions**	Some	Many	Most

Rule of Thumb:

If you have an upcoming system design interview and you're vastly unprepared, **the best thing you can do is reschedule your interview.** The bigger the company, the less they care about rescheduling.

SYSTEM DESIGN KEY TOPICS TO UNDERSTAND

Introduction to System Design

You might be reading this guide because you recently didn't do well in a system design interview. Maybe you watched a YouTube video that made system design seem too complicated to understand. Or perhaps, despite having years of experience, you find it hard to show your skills in the short time of a system design interview.

Do not panic!

You don't need professional experience with distributed systems to do well in system design interviews. Even if you have that experience, many skilled engineers still find system design interviews challenging. **How you do in an interview doesn't reflect your overall value as a software engineer-it shows how well you can handle system design interviews.** While they are related, being a great programmer doesn't guarantee success in these interviews.

What is System Design?

System design is about understanding and planning how different components of a system work together to solve a problem. It involves designing scalable, efficient, and reliable systems, taking into account factors like performance, load balancing, and fault tolerance.

In a system design interview, you are expected to **think through the architecture of a system,** break it down into **smaller parts,** and explain **how those parts will work together to meet the requirements.**

Why is System Design Important in Interviews?

System design interviews test **how well you can approach complex problems and structure your thoughts in a clear and logical manner.** They evaluate your ability to solve real-world problems that require scalable solutions, something that every software engineer faces at some point in their career.

Even if you don't have direct experience with large systems, the ability to think through these problems is what the interviewer is assessing.

Remember

You can pass system design interviews even if you've never designed distributed systems before.

If you have copied files between machines with drag-and-drop, you are halfway there. If you implemented clients or servers or have opened network connections, you've got this. This guide will teach you the most important 20% of information that will appear 80% of the time in system design interviews. By the end of this guide you won't be an expert, but you'll be well on your way to being a better engineer and a much better interview candidate.

The difference between Engineering problems and Design problems

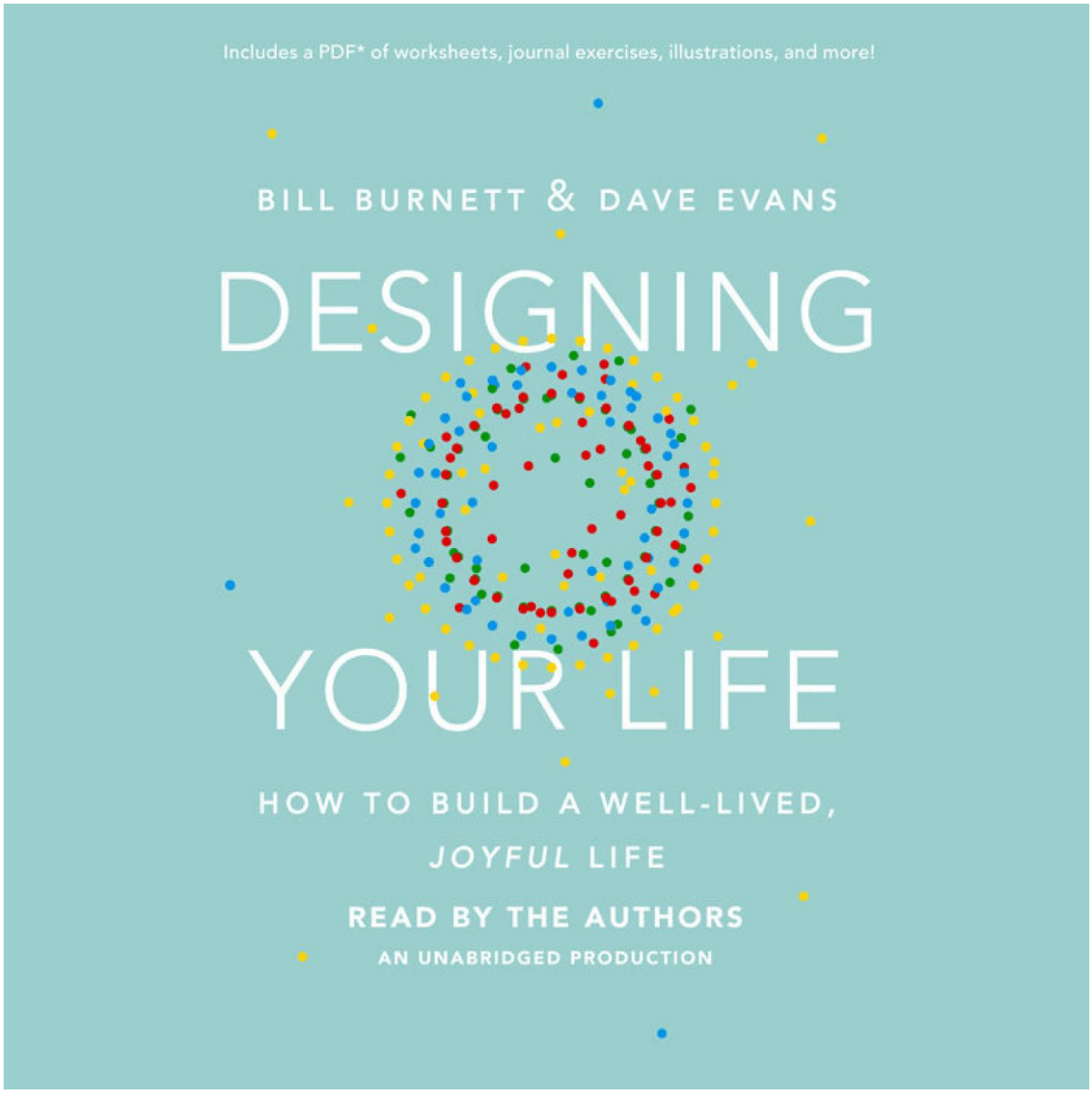
In this excerpt taken from Design Your Life, written by two Stanford professors and engineers, you'll get a better understanding of how different problems require different approaches. We bolded to emphasize the parts that are most important to note.

There's a difference between design problems and engineering problems... **Engineering is a good approach to solving a problem when you can get a great deal of data and you're sure there is one best solution.** Bill [one of the authors] worked on the problem of engineering the hinges on Apple's first laptops, and the solution he and his team came up with made those laptops some of the most reliable on the market. The solution required many prototypes and lots and lots of testing, similar to the design process, but the goal of creating hinges that would last five years (or opening and closing ten thousand times) was fixed, and his team tested many different mechanical solutions until they met their goal. **Once this goal was met, the solution could be reproduced millions of times. It was a good engineering problem.**

"Compare this with the problem of designing the first laptop that had a 'built in mouse'. Because Apple's computers relied on the mouse to do almost everything, building a laptop that required you to be wired up to a regular mouse was unacceptable. **This was a design problem. There was no precedent to design toward, there was no fixed or predetermined outcome,** there were plenty of ideas floating around the lab, and a number of different designs were tested, but nothing was working. Then along came an engineer named Jon Krakower. Jon had been tinkering around with miniaturized trackballs, and had the crazy idea to push the keyboard to the back of the unit, leaving just enough room to squeeze in this tiny pointing device. This turned out to be the big breakthrough everyone had been looking for, and has been part of the signature look of Apple laptops ever since.

When you have a desired outcome (a truly portable laptop computer) but no clear solution in sight, that's when you brainstorm, try crazy stuff, improvise, and

keep 'building your way forward' until you come up with something that works. You know it when you see it. A great design comes together in a way that can't be solved with equations and spreadsheets and data analysis. It has a look and feel all of its own - a beautiful aesthetic that speaks to you."



This is one reason engineers new to system design can bomb their first couple of system design interviews spectacularly: They approach a design problem as if it's an engineering problem.

There is not a single "best" solution to a system design problem. There are no predetermined outcomes. **The less code you write in a system design interview, the better.**

To succeed in a system design interview, you want to collaborate with your interviewer, try crazy stuff, and try more crazy stuff until the design "feels right."