**Chapter 3**

A Framework for System Design Interviews

**Understanding system design interviews**

What does a good interviewer look for?

* Technical design skills
* Ability to collaborate
* Work under pressure
* Resolve ambiguity constructively
* Ability to ask good questions
* Write down assumptions

Red flags?

* Over-engineering – unaware of the compounding costs of over-engineered systems
* Narrow mindedness
* Stubbornness
* Answering without a thorough understanding of the requirements

Traits of a great system design interview

* Question is open-ended
* No one-size-fits-all solution

**Step 1 – Understand the problem and establish design scope**

* Ask questions to understand the problem and clarify ambiguities
* Questions to ask
* What specific features are we going to build?
* How many users does the product have?
* How fast does the company anticipate to scale up? What are the anticipated scales in 3 months, 6 months, and a year?
* What is the company’s technology stack? What existing services you might leverage to simplify the design?

**Step 2 – Propose high-level design and get buy-in**

* Start with a blueprint design, ask for feedback, and treat the interviewer as a teammate and work together
* Draw box diagrams, including
  + Clients (mobile/web)
  + APIs
  + Web servers
  + Data stores & Caches
  + CDN
  + Message queue
* Go through concrete use cases if possible
* Ask if you should include API endpoints and database schema

**Step 3 – Design deep dive**

* By this step, you should have achieved:
  + Agreed on the overall goals and featured scope
  + Sketched out a high-level blueprint for the overall design
  + Obtained feedback from the interviewer on the high-level design
  + Had some initial ideas about areas to focus on in deep dive based on feedback
* Every interviewer is different and may ask to focus on specific points of interest
* Time management is important – avoid focusing on unnecessary details that does not prove your ability to design a scalable system

**Step 4 – Wrap up**

* The interviewer may ask follow-up questions or give you the freedom to continue
* Some useful directions
  + Identify the system bottlenecks and discuss potential improvements
  + Recap the design, especially when multiple solutions were suggested
  + Error cases (server failure, network loss, etc.)
  + Operation issues – monitoring metrics and error logs
  + Deployment - how to roll out the system?
  + How to scale the system?

**Summary**

* Dos
  + Ask for clarifications. Do not assume your assumptions are correct
  + Understand the requirements
  + There is neither a right nor best answer. A solution designed to solve the problems of a young startup is different from that of an established company
  + Communicate - let the interviewer understand your thinking process
  + Suggest multiple approaches if possible
  + Once a general blueprint is agreed, focus on a specific detail
  + Design the most critical components first
  + Bounce ideas off the interviewer. A good interviewer works with you as a teammate
  + If stuck, ask for hints
* Don’ts
  + Be unprepared for typical interview questions
  + Jump into a solution without clarifications the requirements or assumptions
  + Go into too much details on a single component in the beginning
  + Think in silence

**Time Allocation on each step**

* In 45 minutes

May vary depending on the question, interviewer and time given

|  |  |
| --- | --- |
| Understand the problem and establish design scope | 3-10 minutes |
| Propose high-level design and get buy-in | 10-15 minutes |
| Design deep dive | 10-25 minutes |
| Wrap up | 3-5 minutes |