**Product System Design Interview (45 minutes)**

**What is a design interview?**

We will test how you build a product or API at scale that supports an end user product or service, think of this as an “API with input and output” interview.

**What are we looking for?**

* Assume this will be given to a junior engineer to design. Ask questions regarding the requirements of the system, then take the initiative to drive the discussion. The interviewer will ask specific questions as needed and they can validate/clarify question as well.
* Articulate any assumptions you make about the system (users size, frequency of use, etc.)
* Weigh multiple possible approaches and reflect on your design regarding tradeoffs such as:
  + Easy to build APIs vs Long term APIs
  + UI complexity vs server complexity
  + Payload size vs performance
* Anticipate bottlenecks and other problems
* Respond to changing requirements

**What topics to cover?**

|  |  |  |
| --- | --- | --- |
| Topics to Review | These topics are fair game in the design interview: | You will NOT be penalized for not for having **in-depth** experience with these topics: |
| * Storage data models * Scalability * Design patterns * Data ownership * Protocols * Data formats | * Client-server design * Designing for long term vs complexity * Scale in terms of team developers not requests * Accommodating possible product changes | * Distributing data/requests and load balancers * QPS capacity or machine estimates * Sharding * Cross-dc network traffic considerations * Byte size estimation |
|  |
|  |  |

This isn’t an interview where you’ll apply an existing design or one you’ve made already in the past. They will ask you to step out of your comfort zone to design a system outside of your domain knowledge. Although you won’t be given a specific number, you can make assumptions and have them validated by the interviewer as you design your system.

**How to prepare?**

1. Review design topics
2. Practice the below questions with a friend with a whiteboard
   1. Design a service or a product API (ie: Twitter, Netflix, etc.)
   2. Designing a chatting service or a feed API