**What is a system design interview?**

* You would be asked a question to design a solution to a real world engineering problem
* It can cover anything from large service architecture (gmail or gmaps), to algorithms, to hardware constraints.
* It’ll be more of a conversation with you taking the lead - if you have in depth knowledge of a certain area, take the opportunity to showcase your depth of knowledge to the interviewer!

**What are our interviewers looking out for?**

* Your ability to provide a concrete, realistic and quantitative design (i.e. Does your design work for that problem? You might have to go into specific details like numbers - i.e. numbers of machines required, possible queries per second, etc)
* Your ability to implement API and interface design (i.e. whether its easy to use and works well with other teams in the organization, or with existing parts of the system)
* Your ability to discuss the properties of the system (i.e. Given a certain architecture, does it favor latency?)
* Your ability to discuss the tradeoffs and options ( i.e. Don’t just mention that you will use a database. Explain why you choose that? What are the alternative options? Talk through the options and explain why you’ve made that choice)
* How do you productionize: Reliability and scale

**How can I prepare for a system design interview?**

* Review distributed systems concepts
* Consider the entire spectrum of requirement gathering: time, cost, throughput, users, latency, storage, etc
* Think about: How would I determine what is a good latency? How robust and how scalable can it be? How do I productionize it and scale it in a manner that caters to the users?
* Think about: How will you build a flexible design? What components will you keep variable, and what components will you set in stone?

**Approach you can adopt for a system design interview**

1. **Scope the requirements**
   * Always ask clarifying questions! The problem statement will be incomplete, so ask for solid requirements
2. **Design the solution** 
   * Feel free to start with something simple!
   * Be flexible! Don’t be afraid to make changes on some of the designs you have laid out. Interviewer might be giving you hints and guiding you in a certain direction.
3. **Deep Dive in sub-areas**
4. **Iterate**
   * Improve and adapt. Consider scalability, reliability, flexibility and maintainability
   * Always think about backup plans - On scalability, have you designed it in such a way that it can handle data center outage?

**Useful resources**

* [Latency numbers every programmer should know](https://gist.github.com/jboner/2841832)
* [YouTube: Building software systems at Google and Lessons Learned](https://www.youtube.com/watch?v=modXC5IWTJI)
* [Software Engineering advice from building large-scale distributed systems](http://static.googleusercontent.com/media/research.google.com/en//people/jeff/stanford-295-talk.pdf)
* [How to ace a systems design interview](https://www.palantir.com/2011/10/how-to-ace-a-systems-design-interview/)

**Other resources for practice/ refresher**

* [System Design Interview](https://github.com/checkcheckzz/system-design-interview)

**Reading materials if you have time:**

* [Distributed systems and parallel computing](https://ai.google/research/pubs?area=DistributedSystemsandParallelComputing)
* [MapReduce: Simplified Data Processing on Large Clusters](https://ai.google/research/pubs/pub62)
* [Bigtable: A Distributed Storage System for Structured Data](https://ai.google/research/pubs/pub27898)
* [The Google File System](https://ai.google/research/pubs/pub51)

**You might want to prepare on the following topics (not all will be assessed):**

* API design
* Systems architecture
* Capacity/ latency/ throughput
* Scalability
* Network Design
* Fault tolerance
* System level interactions and trade-offs