

CheatSheet: System Design For Job Interview

INTERVIEW

- PDF Link: [cheatsheet-systemdesign-A4.pdf](#), Category: interview
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-systemdesign-A4>
- Related posts: CheatSheet: Leetcode For Code Interview, [#denny-cheatsheets](#)

File me Issues or star this repo.

1.1 Process Of System Design

Num	Name	Summary
1	Outline use cases: List major and focus on some	Show good sense. The questions you asked define your level
2	Understand scale: Data + Traffic	Estimate the volume of the problem space
3	Abstract design	Sketch main components, explain workflow, avoid too deep for details
4	Bottlenecks: Key challenges + Trade-Offs	Valid reasoning to justify your proposals
5	Scale your design	Availability, Scalability, Reliability, Security, etc

1.2 Typical Trade-Off

Num	Name	Summary
1	Performance vs Scalability	
2	Latency vs Throughput	
3	Availability vs Consistency	Brewer's CAP theorem

1.3 Classic Design Problems - Design A Complex Product

Num	Name	Summary
1	Design: TinyURL - A URL Shortener Service	
2	Design: A Parking Lot Service	link
3	Design: An Elevator Service	
4	Design: Uber Backend	
5	Design twitter news feed	link
6	Design web crawler	
7	Design K/V DB	
8	Design: Google Suggestion Service	
9	Design a payment processor	
10	Design a voice conference system	
11	Design google doc	
12	Design gmail	
13	Design instagram, a photo sharing app	
14	Design Yelp, a location-based system	
15	Design an API gateway	
16	Design amazon book recommendation system	
17	Design leetcode	
18	Google autocomplete	
19	Design: A URL Redirecting Feature	
20	Design Google PageRank	
21	Design messaging/notification system	
22	Design search post system	
23	Design memcache	
24	Design typeahead	

1.4 Classic Design Problems - Design A Technical Feature

CheatSheet: Feature Design For Job Interview

1.5 Engineering Of Well-Known Products

Name	Summary
Google	Link: Google Architecture
Facebook	Link: Facebook Live Streams
Twitter	Link: Twitter Image Service, YouTube: Timelines at Scale
Uber	Link: Lessons Learned From Scaling Uber
Tumblr	Link: Tumblr Architecture
StackOverflow	Link: Stack Overflow Architecture

1.6 Grow Design Expertise In Daily Work

Num	Name	Summary
1	Products under the hood	Complicated and high-scale systems which you might use every day
2	Read engineering blogs	Especially for big companies
3	Tools under the hood	Common tools/frameworks
4	Try tools	Use cases; Alternatives; Pros and Cons
5	Read papers	Best practices in papers
6	Try new things	Gain hands-on experience; evaluate alternatives
7	Datastore & OS	Learn how databases and operating systems work
8	Deep dive into daily work	Unify and normalize problems from daily work

1.7 Engineering Blogs/Websites

Name	Summary
Compnay Tech Blog	Website: Facebook Engineering, Website: Google Developers
Compnay Tech Blog	Medium: Netflix Blog, Medium: Airbnb Engineering & Data Science
Individual Tech Blog	Blog: All Things Distributed - Amazon CTO, Blog: highscalability
Papers	Github: papers-we-love
Website	Website: interviewing.io, Website: interviewbit.com
Website	Website: hiredintech - System Design, Github: system-design-primer
YouTube	YouTube: Intro to Architecture and Systems Design Interviews
YouTube	YouTube Channel: Success in Tech, YouTube: Scalability Harvard Web Development
Reference	Link: Preparing for your Software Engineering Interview at Facebook
Reference	Link: The System Design Process
Cheatsheet	Code problems for #oodesign, CheatSheet: System Design For Job Interview
Cheatsheet	CheatSheet: Leetcode For Code Interview
Cheatsheet	CheatSheet: Behavior Questions For Coder Interview

1.8 More Resources

License: Code is licensed under MIT License.

<https://github.com/binhnguyennus/awesome-scalability>