

What? Interview coaching from Googlers!

I want to know more (http://www.gainlo.co/?utm_source=blog&utm_medium=banner <http%3A//blog.gainlo.co/index.php/2016>



Create a TinyURL System

If you have started preparing for system design interviews, you must have heard of one of the most popular question – create a TinyURL (<http://tinyurl.com/>) system.

This week, Gainlo (http://www.gainlo.co/?utm_source=blog&utm_medium=create%20a%20tinyurl%20system&utm_campaign=blog) team has brainstormed tons of ideas about this interview question. I'd like to summarize how we'll analyze this problem with high-level thoughts here.

If you are new to system design interview, I'd highly recommend you take a look at 8 Things You Need to Know Before a System Design Interview (<http://blog.gainlo.co/index.php/2015/10/22/8-things-you-need-to-know-before-system-design-interviews/>) first, which provides general strategies that can help you get started.

Question

How to create a TinyURL (<http://tinyurl.com/>) system?

If you are not familiar with TinyURL, I'll briefly explain here. Basically, TinyURL is a URL shortening service, a web service that provides short aliases for redirection of long URLs. There are many other similar services like Google URL Shortener (<https://goo.gl/>), Bitly (<http://bit.ly>) etc..

For example, URL <http://blog.gainlo.co/index.php/2015/10/22/8-things-you-need-to-know-before-system-design-interviews/> (<http://blog.gainlo.co/index.php/2015/10/22/8-things-you-need-to-know-before-system-design-interviews/>) is long and hard to remember, TinyURL can create an alias for it – <http://tinyurl.com/j7ve58y> (<http://tinyurl.com/j7ve58y>). If you click the alias, it'll redirect you to the original URL.

So if you would design this system that allows people input URLs with short alias URLs generated, how would you do it?

High-level idea



Let's get started with basic and high-level solutions and we can keep optimizing it later on.

At first glance, each long URL and the corresponding alias form a key-value pair. I would expect you to think about something related to hash (https://en.wikipedia.org/wiki/Hash_function) immediately.

Therefore, the question can be simplified like this – given a URL, how can we find hash function F that maps URL to a short alias:

$$F(\text{URL}) = \text{alias}$$

and satisfies following condition:s

1. Each URL can only be mapped to a unique alias
2. Each alias can be mapped back to a unique URL **easily**

The second condition is the core as in the run time, the system should look up by alias and redirect to the corresponding URL quickly.

Basic solution

To make things easier, we can assume the alias is something like `http://tinyurl.com/<alias_hash>` and `alias_hash` is a fixed length string.

If the length is 7 containing [A-Z, a-z, 0-9], we can serve $62^7 \approx 3500$ billion URLs. It's said that there are ~644 million URLs (<https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=how%20many%20urls%20are%20there>) at the time of this writing.

To begin with, let's store all the mappings in a single database. A straightforward approach is using `alias_hash` as the ID of each mapping, which can be generated as a random string of length 7.

Therefore, we can first just store `<ID, URL>`. When a user inputs a long URL "http://www.gainlo.co", the system creates a random 7-character string like "abcd123" as ID and inserts entry `<"abcd123", "http://www.gainlo.co">` into the database.

In the run time, when someone visits `http://tinyurl.com/abcd123`, we look up by ID "abcd123" and redirect to the corresponding URL "http://www.gainlo.co (http://www.gainlo.co)".

Performance VS flexibility



There are quite a few follow-up questions for this problem. One thing I'd like to further discuss here is that by using GUID (https://en.wikipedia.org/wiki/Globally_unique_identifier) (Globally Unique Identifier) as the entry ID, what would be pros/cons versus incremental ID in this problem?

If you dig into the insert/query process, you will notice that using random string as IDs may sacrifice performance a little bit. More specifically, when you already have millions of records, insertion can be costly. Since IDs are not sequential, so every time a new record is inserted, the database needs to go look at the correct page for this ID. However, when using incremental IDs, insertion can be much easier – just go to the last page.

So one way to optimize this is to use incremental IDs. Every time a new URL is inserted, we increment the ID by 1 for the new entry. We also need a hash function that maps each integer ID to a 7-character string. If we think each string as a 62-base numeric, the mapping should be easy (Of course, there are other ways).

On the flip side, using incremental IDs will make the mapping less flexible. For instance, if the system allows users to set custom short URL, apparently GUID solution is easier because for whatever custom short URL, we can just calculate the corresponding hash as the entry ID.

Note: In this case, we may not use random generated key but a better hash function that maps any short URL into an ID, e.g. some traditional hash functions like CRC32 (<http://www.accuhash.com/what-is-crc32.html>), SHA-1 (<https://en.wikipedia.org/wiki/SHA-1>) etc..

Cost

I can hardly not ask about how to evaluate the cost of the system. For insert/query, we've already discussed above. So I'll focus more on storage cost.

Each entry is stored as <ID, URL> where ID is a 7-character string. Assuming max URL length is 2083 characters (<https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=max%20url%20length>), then each entry takes $7 * 4 \text{ bytes} + 2083 * 4 \text{ bytes} = 8.4 \text{ KB}$. If we store a million URL mappings, we need around 8.4G storage.

If we consider the size of database index (https://en.wikipedia.org/wiki/Database_index) and we may also store other information like user ID, date each entry is inserted etc., it definitely requires much more storage.

Multiple machines

Apparently, when the system has developed to certain scale, a single machine is not capable to store all the mappings. How do we scale with multiple instances?

The more general problem is how to store hash mapping across multiple machines. If you know distributed key-value store (https://en.wikipedia.org/wiki/Distributed_data_store), you should know that this can be a very complicated problem. I will discuss only high-level ideas here and if you are interested in all those details, I'd recommend you read papers like Dynamo: Amazon's Highly Available Key-value Store (<http://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf>).

In a nutshell, if you want to store a huge amount of key-value pairs across multiple instances, you need to design a lookup algorithm that allows you to find the corresponding machine for a given lookup key.

For example, if the incoming short alias is <http://tinyurl.com/abcd123>, based on key "abcd123" the system should know which machine stores the database that contains entry for this key. This is exactly the same idea of database sharding ([https://en.wikipedia.org/wiki/Shard_\(database_architecture\)](https://en.wikipedia.org/wiki/Shard_(database_architecture))).

A common approach is to have machines that act as a proxy, which is responsible for dispatching requests to corresponding backend stores based on the lookup key. Backend stores are actually databases that store the mapping. They can be split by various ways like use $\text{hash}(\text{key}) \% 1024$ to divide mappings to 1024 stores.

There are tons of details that can make the system complicated, I'll just name a few here:

- Replication ([https://en.wikipedia.org/wiki/Replication_\(computing\)](https://en.wikipedia.org/wiki/Replication_(computing))). Data stores can crash for various random reasons, therefore a common solution is having multiple replicas for each database. There can be many problems here: how to replicate instances? How to recover fast? How to keep read/write consistent?
- Resharding. When the system scales to another level, the original sharding algorithm might not work well. We may need to use a new hash algorithm to reshard the system. How to reshard the database while keeping the system running can be an extremely difficult problem.
- Concurrency ([https://en.wikipedia.org/wiki/Concurrency_\(computer_science\)](https://en.wikipedia.org/wiki/Concurrency_(computer_science))). There can be multiple users inserting the same URL or editing the same alias at the same time. With a single machine, you can control this with a lock. However, things become much more complicated when you scale to multiple instances.

Summary

I think you've already realized that the problem is not complicated at first glance, but when you dig into more details especially consider scale issues, there can be a lot of follow-up problems.

As we said in earlier post (<http://blog.gainlo.co/index.php/2016/02/17/system-design-interview-question-how-to-design-twitter-part-1/>), specific techniques used is not that important. What really matters is the high-level idea of how to solve the problem. That's why I don't mention things like Redis (<http://redis.io/>), RebornDb (<http://reborndb>) etc..

Again, there are infinite ways to extend this problem further. I'd like to use this post as a starting point for you to get familiar with system design interview.

If you find this post helpful, I would really appreciate if you can share it with your friends. Also you can check more system design interview questions (<http://blog.gainlo.co/index.php/category/system-design-interview-questions/>) and analysis here.

The post is written by **Gainlo** (http://www.gainlo.co/?utm_source=blog&utm_medium=footer-link http%3A//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/&utm_campaign=blog) - a platform that allows you to have mock interviews with employees from Google, Amazon etc..

I'd like to learn more (http://www.gainlo.co/?utm_source=blog&utm_medium=footer <http%3A//blog.gainlo.co/index.php/2016>




[http://www.facebook.com/sharer.php?u=http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-](http://www.facebook.com/sharer.php?u=http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/)

[create-tinyurl-system/](http://www.facebook.com/sharer.php?u=http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/); [http://twitter.com/share?url=http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-](http://twitter.com/share?url=http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/)

question-create-tinyurl-system/&text=Create+a+TinyURL+System+>  ([http://www.linkedin.com/shareArticle?](http://www.linkedin.com/shareArticle?mini=true&url=http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/)

[mini=true&url=http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/](http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/);  10 

[http://reddit.com/submit?url=http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/&title=Create a TinyURL System](http://reddit.com/submit?url=http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/&title=Create+a+TinyURL+System;); 

Subscribe

We'll email you when there are new posts here.

Related Posts:

1. **Create a Photo Sharing App** (<http://blog.gainlo.co/index.php/2016/03/01/system-design-interview-question-create-a-photo-sharing-app/>)
2. **Design a Cache System** (<http://blog.gainlo.co/index.php/2016/05/17/design-a-cache-system/>)
3. **Design a Key-Value Store (Part I)** (<http://blog.gainlo.co/index.php/2016/06/14/design-a-key-value-store-part-i/>)
4. **Random ID Generator** (<http://blog.gainlo.co/index.php/2016/06/07/random-id-generator/>)

22 thoughts on "Create a TinyURL System"



RAJESH

March 9, 2016 at 8:19 pm (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-427>)

Another well written design post. A complex problem explained in easy to understand terms.

Keep these ones coming, Jake. 😊

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replyto=427#respond>



JAKE

March 9, 2016 at 11:54 pm (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-428>)

Thanks Rajesh. More posts in this series are coming. Stay tuned!

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replyto=428#respond>



SAMUEL

March 11, 2016 at 3:42 pm (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-440>)

There are a few points in your analysis that are incorrect. An interviewer familiar with this test will beat on those:

- * A random string is not a hash. The fundamental property of hashes is that they are functions from a domain (say strings) to a smaller domain (say strings of 5 characters). The same input produces always the same output.
- * Neither hashes nor random strings of 7 characters would work in practice. You calculated how many URLs you can store, but that assumes you can efficiently use all that space. Once you have a few urls stored in your system, you will have clashes. Avoiding those

clashes is non trivial in a globally distributed system. Even worse, random strings will hit that limit much earlier if the system is popular since every requests generates a new key. Conversely, with actual hashes, the likelihood of clashes depends on the amount of urls you store. Since you have chosen a fairly long identifier, you 'd be "more or less" safe until you store about 100.000.000 URLs, but that goes against the URL being "tiny". You could store the 644000000 urls using 5 character identifiers, so your identifiers are currently too big.

* When you suggest using a "better function" pointing out to actual hashes. Both functions that you suggest are bad examples. CRC is designed for cheap error detection, it's distribution is less than ideal and is easy to attack (an attacker could very easily generate many different urls that hash to the same crc). SHA-1 is better, but has also know attacks.

* I doubt there is a cost penalty for the database depending on which type of ID you use. The cost of random IDs is that you need to check for clashes which requires a transaction, which is not really possible in a globally distributed system.

* I got confused when you introduced the idea of using GUIDs. GUIDs won't work, they are simply too large (to guarantee uniqueness). But then you move to incremental ids, I assume that you refer to a counter that increments per time you write an entry. While that could work, there is much more to that. Generating a unique sequence of ids in a distributed system is not trivial either, you'll need to find your ways around that. However, once you do that, why are you adding an extra hash to the ID? It is already unique.

* You don't only need distribution due to scalability issues (which I doubt would be much of a problem here) but due to both availability and latency. If the system is global, you don't want people in Australia connecting to some datacenter in Ireland, plus you need to cope with network and server failures. Distributing the solution _is_ the meat of this problem, if you could solve it with only one computer is pretty trivial.

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=440#respond>



DAVID DEMARTINI ([HTTP://WWW.OUTSPOKEN.NINJA](http://www.outspoken.ninja))

March 14, 2016 at 4:20 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-463>)

SAMUEL,

Thank you for saving me the time of writing up a detailed reply, you covered all my points and brought up a couple I was not immediately considering.

Well said.

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=463#respond>



NAMELESS

April 9, 2016 at 10:14 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-576>)

Can you please write an article on How to Design a rate limiter?.....I was asked a question in my Google Interview which was a slight modification of rate limiter but sadly couldn't answer it.

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=576#respond>



CHAITANYA

April 24, 2016 at 2:12 pm (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-665>)

Hi Samuel,

In Your second point "random strings will hit that limit much earlier if the system is popular since every requests generates a new key." can you please explain this point in detail as i could not able to understand.

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=665#respond>

**SHILPA**May 6, 2016 at 4:26 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-747>)

Hi,

Doesn't 1 char take 1 byte of memory?

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=747#respond>**DVN SRAVAN KUMAR**May 30, 2016 at 2:43 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-984>)

thanks for the article !

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=984#respond>**CHUCK (HTTP://REZSVSDIH.COM)**October 29, 2016 at 4:07 pm (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-3196>)

I can't beivle I've been going for years without knowing that.

blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=3196#respond**BRAVE**May 31, 2016 at 1:05 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-990>)

644 million URLs seems to be incorrect. The link says that there are 644 million active websites, not URLs. Facebook alone would have many more than 644 million publicly accessible URLs.

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=990#respond>**SEGGIOLINO AUTO (HTTP://LIJIN338.COM)**August 6, 2016 at 10:44 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-1816>)

Hello, I think your site might be having browser compatibility issues.

When I look at your blog site in Firefox, it looks fine but when opening in Internet Explorer, it has some overlapping. I just wanted to give you a quick heads up! Other then that, wonderful blog!

<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=1816#respond>**JAKE**August 8, 2016 at 6:20 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-1837>)

Thanks Seggiolino. Our engineer team is looking at the issue 😊

blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=1837#respond

**LONG UNDERWEAR (HTTP://WWW.HUE.COM)**October 11, 2016 at 4:51 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-2910>)

I am continuously looking online for posts that can benefit me.
Thank you!

//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=2910#respond)

**GANGSTAR VEGAS INFINITE MONEY CHEAT (HTTPS://GANGSTARVEGASCHEATS.WORDPRESS.COM/)**October 13, 2016 at 12:08 pm (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-2937>)

Deference to article author, some superb information.

//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=2937#respond)

**CASSIE (HTTP://GTESBMEELDB.COM)**October 29, 2016 at 3:41 pm (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-3162>)

What I find so inntiestreg is you could never find this anywhere else.

blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=3162#respond)

**RMN**December 1, 2016 at 5:16 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-3795>)

It would be really nice if you could talk about some useful concepts instead of just stating the problem.

For example:

"How to reshard the database while keeping the system running can be an extremely difficult problem"

Why not mention consistent hashing?

//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=3795#respond)

**OALA SUB PRESIUNE BUN (HTTP://GAMMADOSPORCORRIDA.COM)**December 6, 2016 at 3:17 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-3882>)

If some one wishes to be updated with latest technologies then he must be pay a
quick visit this web page and be up to date everyday.

//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=3882#respond)

**SERVICE REPAR MANUALS (HTTP://WWW.REPAIRLOADER.NET/)**December 6, 2016 at 8:54 pm (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-3890>)

I think that what you posted was actually very logical.

However, what about this? what if you added a little content?

I mean, I don't wish to tell you how to run your website,

but suppose you added a title that makes people desire more?

I mean Create a TinyURL System is a little vanilla. You ought to peek at Yahoo's home page and see how they write article titles to grab viewers to click.

You might add a video or a pic or two to get people excited about what you've got to say. In my opinion, it might make your blog a little livelier.

//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=3890#respond)



COMPARATIF COLONNE DE DOUCHE (HTTP://INTERESTCORNER.COM)

December 10, 2016 at 6:19 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-3940>)

This design is steller! You most certainly know how to keep a reader entertained. Between your wit and your videos, I was almost moved to start my own blog (well, almost...HaHa!) Excellent job.

I really loved what you had to say, and more than that, how you presented it. Too cool!

//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=3940#respond)



SMERIGLIATRICE (HTTP://ESTONIAN-EURO.COM)

December 18, 2016 at 11:46 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-4101>)

Excellent, what a web site it is! This webpage presents helpful data to us, keep it up.

//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=4101#respond)



MEJOR AMOLADORA (HTTP://SKYWAYJUMPER.COM/)

December 27, 2016 at 7:08 pm (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-4337>)

Excellent article. I will be going through some of these issues as well..

//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=4337#respond)



ALBELLI FOTOALBUM (HTTP://WWW.COUPONPROMOTIE.COM/)

January 9, 2017 at 1:35 am (<http://blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/#comment-4623>)

I'd like to thank you for the efforts you've put in writing this website. I am hoping to view the same high-grade content by you in the future as well. In fact, your creative writing abilities has motivated me to get my very own blog now 😊

//blog.gainlo.co/index.php/2016/03/08/system-design-interview-question-create-tinyurl-system/?replytocom=4623#respond)

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

[◀ Create a Photo Sharing App \(http://blog.gainlo.co/index.php/2016/03/01/system-design-interview-question-create-a-photo-sharing-app/\)](http://blog.gainlo.co/index.php/2016/03/01/system-design-interview-question-create-a-photo-sharing-app/)[9 Tips to Prepare Your Coding Interviews In One Week ▶ \(http://blog.gainlo.co/index.php/2016/03/15/9-tips-to-prepare-your-coding-interviews-in-one-week/\)](http://blog.gainlo.co/index.php/2016/03/15/9-tips-to-prepare-your-coding-interviews-in-one-week/)

CATEGORIES

[Coding Interview Questions \(http://blog.gainlo.co/index.php/category/coding-interview-questions/\)](http://blog.gainlo.co/index.php/category/coding-interview-questions/)[Common Pitfalls \(http://blog.gainlo.co/index.php/category/common-pitfalls/\)](http://blog.gainlo.co/index.php/category/common-pitfalls/)[Common Questions \(http://blog.gainlo.co/index.php/category/common-questions/\)](http://blog.gainlo.co/index.php/category/common-questions/)[Facebook Interview Questions \(http://blog.gainlo.co/index.php/category/facebook-interview-questions/\)](http://blog.gainlo.co/index.php/category/facebook-interview-questions/)[Mock Interview Thoughts \(http://blog.gainlo.co/index.php/category/mock-interview-thoughts/\)](http://blog.gainlo.co/index.php/category/mock-interview-thoughts/)[Others \(http://blog.gainlo.co/index.php/category/others/\)](http://blog.gainlo.co/index.php/category/others/)[System Design Interview Questions \(http://blog.gainlo.co/index.php/category/system-design-interview-questions/\)](http://blog.gainlo.co/index.php/category/system-design-interview-questions/)[The Complete Guide to Google Interview Preparation \(http://blog.gainlo.co/index.php/category/google-interview-preparation/\)](http://blog.gainlo.co/index.php/category/google-interview-preparation/)[Uber Interview Questions \(http://blog.gainlo.co/index.php/category/uber-interview-questions/\)](http://blog.gainlo.co/index.php/category/uber-interview-questions/)

Gainlo - Mock Interview With Professionals

[Visit Gainlo \(http://www.gainlo.co/?utm_source=blog&utm_medium=site-footer http%3A//blog.gainlo.co/index.php](http://www.gainlo.co/?utm_source=blog&utm_medium=site-footer)

