**System Design:**

DB First vs API First:

<https://softwareengineering.stackexchange.com/questions/264379/code-first-vs-database-first/264389#264389?s=21ca9403dc494760bcc06ecb5a16980d>

<https://www.ryadel.com/en/code-first-model-first-database-first-vs-comparison-orm-asp-net-core-entity-framework-ef-data/>

<https://dzone.com/articles/an-api-first-development-approach-1>

[API](https://dzone.com/articles/versioning-restful-services-with-spring-boot) [Versioning](http://www.springboottutorial.com/spring-boot-versioning-for-rest-services) – [Should](https://dzone.com/articles/versioning-rest-api-with-spring-boot-and-swagger) be avoided as much as possible. There are various of doing API versioning – URI Versioning, Header Versioning, Request Parameter Versioning, Media Type Versioning

**DB Related Theory:**

**SQL** – ideal for Transactional Storage (booking, payment etc.), **NoSQL** – Non-Transactional Storage (comments, reviews, history data etc.)

Surrogate key vs Natural business key - [here](https://stackoverflow.com/questions/2186260/when-to-use-an-auto-incremented-primary-key-and-when-not-to) [here](https://stackoverflow.com/questions/63090/surrogate-vs-natural-business-keys) [here](https://stackoverflow.com/questions/707657/picking-the-best-primary-key-numbering-system) (very nice)

If you are using surrogate keys then don’t forget to key create unique constraint on the natural business key - [here](https://stackoverflow.com/questions/337503/whats-the-best-practice-for-primary-keys-in-tables)

Usually the rule of thumb is that never use meaningful information in primary keys (like Social Security number or barcode). Just plain auto incremented integer. However constant the data seems - it may change at one point (new legislation comes and all SSNs are recalculated). Integer comparisons are much cheaper than string comparisons, and it will occupy less space in the DB- [here](https://stackoverflow.com/questions/506164/use-item-specific-prefixes-and-autonumber-for-primary-keys)

**Common Mistakes on DB Design** - [here](https://stackoverflow.com/questions/621884/database-development-mistakes-made-by-application-developers?page=1&tab=votes#tab-top)

Load Balancers – Software and Hardware

Layer 4 vs Layer 7 load balancing

JVM Profilers - [here](https://www.baeldung.com/java-profilers), [Uber JVM Profiler](https://github.com/uber-common/jvm-profiler/tree/master/src/main/java/com/uber/profiling/profilers)

**Courses to do:**

Apache Kafka course (Udemy)

**Approach for System Design:**

1. Ask for Requirements
2. State assumptions if any and confirm from them
3. Think out loud
4. Think of Design Patterns you can use

File System Design : [here](https://leetcode.com/articles/design-in-memory-file-system/)

10 Common Design Questions ([here](https://www.evernote.com/client/web?login=true&paywallUrlName=WEB-INF%2Fjsp%2Fredesign%2Fglobal%2Flayout.jsp#?anb=true&n=1fb605f7-d935-4d66-a97e-050555b997e2&s=s703&)):

1. Vending Machine (State Pattern)

<https://www.youtube.com/watch?v=VIzWKPRG9lo>

<https://www.youtube.com/watch?v=MGEx35FjBuo>

1. URL Shortner
2. Score Card (Subscriber/Publisher Pattern)
3. Online Gaming (Chess, Snakes&Ladder etc)
4. Elevator/Parking System, Scheduler design, Text Editor, File System Design
5. Messaging FrameWork (Whatsapp, Messenger)
6. Social Networking Site (Facebook)
7. Ticket Booking App (Oyo, Redbus, IRCTC)
8. Storage Application (Instagram)
9. Tax booking Apps (Ola, Uber)
10. Snakes And [Ladders](http://www.cvc.uab.es/shared/teach/a21291/temes/object_oriented_design/materials_adicionals/ladders_and_snakes/ladders_and_snakes.pdf)
11. Miscellaneous – File Converter, Compiler

Top 10 Design Questions – [Link1](https://hackernoon.com/top-10-system-design-interview-questions-for-software-engineers-8561290f0444) [Link2](https://www.hackingnote.com/en/interview/system-design-interview-questions)

Design Cheat Sheet – [Link2](https://gist.github.com/vasanthk/485d1c25737e8e72759f)

Consistent Hashing – [Link1](https://www.youtube.com/watch?v=viaNG1zyx1g) [Link2](https://www.youtube.com/watch?v=bBK_So1u9ew)

All About RabbitMQ - [here](https://www.cloudamqp.com/blog/2017-12-29-part1-rabbitmq-best-practice.html) [here](https://www.rabbitmq.com/confirms.html)

Text Editor - [here](https://www.eduonix.com/dashboard/learn-how-to-create-a-texteditor-with-java/lecture/15909916), [here](https://zatackcoder.com/creating-text-editor-java/) (Done using Java swing, so not refer this- useful only for learning swing)

Useful Links:

<http://highscalability.com/>

Implement a priority queue with dynamic priorities

Distributed Cache: [here](http://blog.gainlo.co/index.php/2016/05/17/design-a-cache-system/?utm_campaign=quora&utm_medium=What+is+distributed+caching?&utm_source=quora)

1. Which algorithm to use? LRU
2. What is the eviction policy?
3. How to handle concurrency updates?
4. How to implement distributed Cache?

TinyBit URL:

1. Size of the URL u want to convert to?
2. How to handle collisions?

Elevator Design:

Book My Show:

1. Fd

Document Store Design:

1. Fd

Parking Lot: