@priyavajpeyi

TOP 10 System Design Interview Questions

DID YOU KNOW?

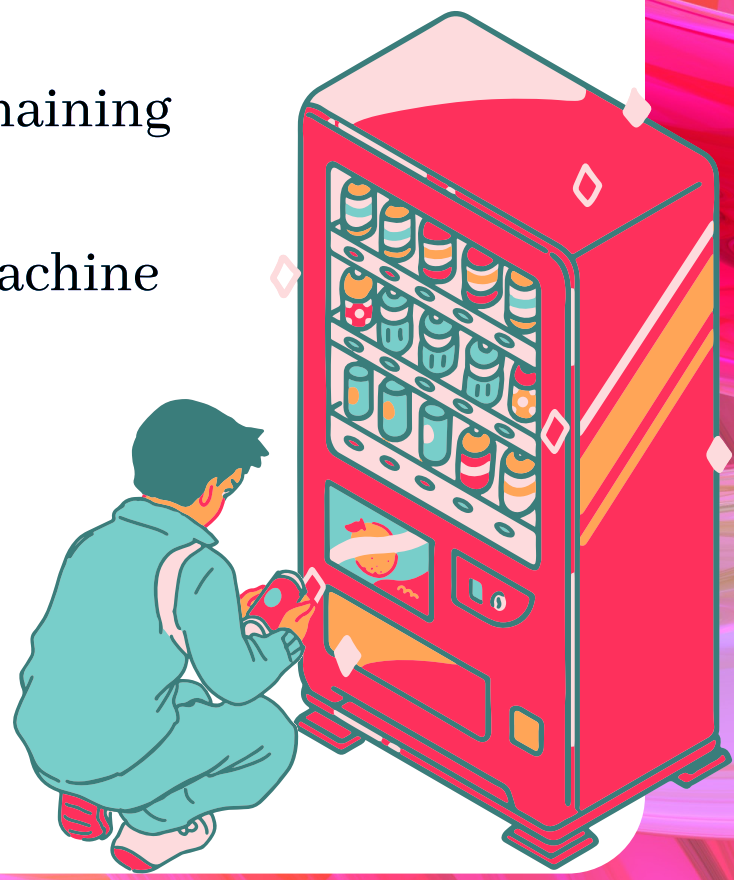
1. How do you design the Vending Machine?

Problem Statement

You need to design a Vending Machine which:

1. Accepts coins of 1,5,10,25 Cents i.e. penny, nickel, dime, and quarter.
2. Allow user to select products Coke(25), Pepsi(35), Soda(45)
3. Allow users to take a refund by canceling the request.
4. Return the selected product and remaining change if any
5. Allow reset operation for vending machine supplier.

Solution



2. Design a URL Shortening service like goo.gl or bit.ly?

Problem Statement

Functional Requirements:

1. Given a URL, our service should generate a shorter and unique alias of it (short link).
2. When users access a short link, our service should redirect them to the original link.
3. Users should optionally be able to pick a custom short link for their URL.
4. Links will expire after a standard default timespan. Users should be able to specify the expiration time.

Non-Functional Requirements:

1. The system should be highly available. This is required because, if our service is down, all the URL redirections will start failing.
2. URL redirection should happen in real-time with minimal latency.
3. Shortened links should not be guessable (not predictable).

Solution

3. How would you create your own Instagram?

Problem Statement

Functional Requirements:

1. Users should be able to upload/download/view photos.
2. Users can perform searches based on photo/video titles.
3. Users can follow other users.
4. The system should generate and display a user's News Feed consisting of top photos from all the people the user follows.

Non-Functional Requirements:

1. Our service needs to be highly available.
2. The acceptable latency of the system is 200ms for News Feed generation.
3. Consistency can take a hit (in the interest of availability) if a user doesn't see a photo for a while; it should be fine.
4. The system should be highly reliable; any uploaded photo or video should never be lost.

Solution

4. How do you design global file sharing & storage apps like Google Drive or Dropbox?

Problem Statement

Functional Requirements:

1. Users should be able to upload and download files from any device. And the files will be synchronized in all the devices that the user is logged in.

Non-Functional Requirements:

1. Users can upload and download files from any device. The service should support storing a single large file up to 1 GB. Service should synchronize automatically between devices; if one file is uploaded from a device, it should be synced on all devices that the user is logged in.

Solution

5. How do you design a Twitter Clone?

Problem Statement

Functional Requirements:

1. The user should be able to tweet as fast as possible
2. The user should be able to see Tweet Timeline(s)
3. User timeline: Displaying user's tweets and tweets user retweet
4. Home timeline: Displaying Tweets from people users follow
5. Search timeline: Display search results based on #tags or search keyword
6. The user should be able to follow another user
7. Users should be able to tweet millions of followers within a few seconds (5 seconds)
8. The user should see trends

Solution

6. How to design a global video streaming service like YouTube or NetFlix?

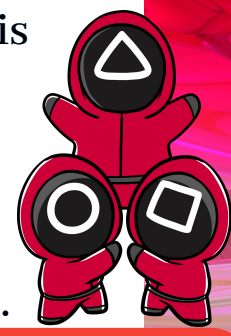
Problem Statement

Functional Requirements:

1. Content creators can upload videos.
2. Viewers can watch videos on different devices (mobile, TV, etc.).
3. Users can search videos by their titles.
4. Users can like/dislike or comment on videos.
5. The system can store likes, dislikes and number of views to display these stats to the users.

Non-Functional Requirements:

1. There should be no buffering so the viewers can have a real-time experience when watching videos.
2. The system should have low latency and high availability. Consistency is of secondary importance in this case, since it's acceptable if a newly uploaded video is not available to a user for a while.
3. Video storage should be reliable. Uploaded videos should not be lost.
4. The system should be able to scale with the increasing number of users.



Solution

7. How to design a Web Crawler like Google?

Problem Statement

Functional Requirements:

1. Crawling Frequency: Also known as crawl rate, or crawl frequency refers to how often you want to crawl a website. You can have different crawl rates for different websites. For example, news websites might need to be crawled more often.
2. Dedup: Where multiple crawlers are used, they may add duplicate links to the same URL pool. Dedup or duplicate detection involves the use of a space-efficient system, like Bloom Filter, to detect duplicate links, so your design isn't crawling the same sites.
3. Protocols: Think about the protocols that your crawler will cater to. A basic crawler can handle HTTP links, but you can also modify the application to work over STMP or FTP.
4. Capacity: Each page that is crawled will carry several URLs to index.

Solution

8. How to design a global ride-hailing service e.g. Uber, Grab, or Ola backend?

Problem Statement

Functional Requirements:

1. Drivers must be able to frequently notify the service regarding their current location and availability
2. Passengers should be able to see all nearby drivers in real-time
3. Customers can request a ride using a destination and pickup time.
4. Nearby drivers should be notified when a customer needs to be picked up.
5. Once a ride is accepted, both the driver and customer must see the other's current location for the entire duration of the trip.
6. Once the drive is complete, the driver completes the ride and should then be available for another customer.

Solution

9. How would you design a Parking Lot system?

Problem Statement

Functional Requirements:

1. The parking lot should have multiple floors.
2. The parking lot should have multiple entry and exit points.
3. Customers can collect a parking ticket from the entry points and can pay the parking fee at the exit points on their way out.
4. Customers can pay the tickets at the automated exit panel or to the parking attendant.
5. Customers can pay via both cash and credit cards.
6. Customers should also be able to pay the parking fee at the customer's info portal on each floor. If the customer has paid at the info portal, they don't have to pay at the exit.
7. The system should not allow more vehicles than the maximum capacity of the parking lot. If the parking is full, the system should be able to show a message at the entrance panel and on the parking display board on the ground floor.
8. Each parking floor will have many parking spots. The system should support multiple types of parking spots such as Compact, Large, Handicapped, Motorcycle, etc.
9. The Parking lot should have some parking spots specified for electric cars. These spots should have an electric panel through which customers can pay and charge their vehicles.
10. The system should support parking for different types of vehicles like car, truck, van, motorcycle, etc.
11. Each parking floor should have a display board showing any free parking spot for each spot type.
12. The system should support a per-hour parking fee model. For example, customers have to pay $4 for the first hour, $3.5 for the second and third hours, and $2.5 for all the remaining hours.

Solution

10. How do you design an Elevator of the Lift system?

Problem Statement

Functional Requirements:

1. An elevator can move up, down or standstill.
2. An elevator can transfer passengers from one floor in a building to another floor in the minimum time possible.
3. The elevator door can only open when it is standstill in a floor (i.e. not moving).
4. Let's first discuss the number of floors that a building can have. The tallest building in the world is Burj Khalifa and it has around 163 floors. Let's just assume that our system has 200 floors, which is still far greater than what the tallest building has. We cannot just assume some arbitrarily large number. When we are gathering requirements, we need to make sure that we are still collecting some reasonable requirements.
5. Burj Khalifa has around 57 elevators. We can assume in our system we have around 100 elevators.
6. Then there are some requirements related to each elevator like:
    a. Number of passengers / Load on the elevator
    b. Moving speed of the elevator
    c. Now the fourth requirement is about what we would like to minimize.
    d. Do we like to minimize the wait time of the passengers?
    e. Or do we like to minimize the overall wait time in the system?

Solution