**System Design Components:**

1. DNS
2. Load Balancer
3. API Gateway
4. CDN
5. Forward Proxy vs Reverse Proxy
6. Caching
7. Data Partitioning
8. Database Replication
9. Distributed Message Queue
10. Microservice
11. NoSQL Database
12. Database Index
13. Distributed File Systems
14. Notification System
15. Distributed Coordination Services
16. Heartbeat
17. Checksum
18. SQL vs. NoSQL
19. Forward and reverse proxies
20. API design
21. Performance measurement of scalable systems
22. Scalability, availability, and reliability of large systems
23. Data partitioning techniques

**Top system design interview questions for software developers**

* Design a global file storage and sharing service like Dropbox or Google Drive.
* Design a social network like Facebook, Instagram, or Twitter.
* Design a global chat service like Facebook Messenger or WhatsApp.
* Design a global video streaming service like YouTube or Netflix.
* Design a ride-hailing service like Uber or Lyft.
* Design a web crawler.
* Design Typeahead search.
* Design Ticketmaster.
* Design a key-value store.
* Design an API rate limiter.
* Design URL shortening service.
* Design a distributed messaging system.
* Design an elevator system.
* Design a parking lot system.
* Design a game of chess.
* Design a movie ticket booking system.
* Design a library management system.
* Design an e-commerce store like Amazon

**Frontend System Design**

HLD:

* Requirements
* Functional
* Non-functional
* Scoping
* Tech Choices
* Component Architecture
* Data Model
* Data API

LLD:

* Tech Choices
* Component Architecture
* Date Model
* Data API

**FACEBOOK NEWS FEED**

**Requirements:**

**Functional:**

* Add Post
* Image
* Video
* Audio
* Link
* Text
* Formatting Text
* View Post
* Save
* Report
* Edit (own post)
* Liked & comment, share on a post

**Non-functional**:

* Mobile/Desktop
* Responsive/Adaptive
* Rendering – CSR/SSR/SSG/ISR
* SPA/MPA
* Authentication
* Authorization
* Profile Management
* Localization
* Asset Optimization (CSS, JS, Images)
* Internationalization
* Accessibility
* Performance (FCP, LCP, TTI, Web Vitals)
* Versioning
* CI/CD
* Caching
* Optimization
* Security
* Logging/Monitoring/Analytics
* PWA

**Teck Stack**:

* Framework
* State Management
* Routing
* Folder Structure
* Design system vs custom components
* REST vs GraphQL
* Rendering: CSR/SSR/SSG/ISR
* Mono vs Multi Repo
* Micro-frontend
* Build Tools
* CI/CD
* Docker & Kubernetes

**DATA PROTOCOL:**

* REST/GraphQL/SSE/rPF
* JSON/Protocol Buffer

**DATA API:**

* getPosts(): Post[]
* addPost(text,media,tags):Boolean
* like(postcftId): Boolean
* share(postId): Boolean
* report(postId): Boolean
* save(postId): Boolean

DATA MODEL:

* **Posts:{**

Id:ID,

text:string,

media:blob,

tags:[string],

likes:[ID:user],

comments:[Comment],

shares:[ID:user],

reports:[ID:user],

createdAt:DateTime,

createdBy:User

**}**

* **User:{**

id:ID,

location:string,

favouriteTags:[string],

createdAt:DateTime

**}**

* **Comment:{**

id:ID,

text:string,

media:blob,

replies:[Comment],

createdAt:DateTime,

createdBy:User

**}**

**Resources:**

<https://www.designgurus.io/blog/system-design-interview-fundamentals>

<https://www.designgurus.io/blog/system-design-interview-guide>