Front-End System Design Interview Guide

JSer

1. Understand what it is

What are expected to create?
Are we talking about the same service.

2. Decide the scope that suits

Not a perfect service, but the core parts that
Could appeal to interviewer, within 45 minutes

List up the TODO and NOT-TODO.

Check with interviewer

If system design

1. What is the basic goal of the feature
2. What is your non-functional goal of the feature
3. What is the data flow (api) / user flow
4. What is the MVP
5. What is the state of the UI component
6. How would you separate them in parts and put then together (UI/logic)
7. What is the core spec

If Product design

1. The goal of the web service
2. Relation with native apps? Replica? Or lite version? PWA needed?
3. Target platform, mobile? Desktop?
4. Mobile first, needs design for Desktop?
5. Is SEO a concern? SSR needed? SPA enough

If Product design

6. Volume of the service, team members.
7. What is the MVP, core features
8. What is the shining point, from service & DX
9. What is the future roadmap

3. Assumptions on background

Suppose the DAU/MAU of the service
Suppose how many interactions occurs in a day
Suppose 300KB is tolerable
Suppose average api response is 100ms
.etc

4. Big Picture

Draw a diagram or list up the outline
Data flow / User interaction flow
Check with interviewer

5. Key challenges, bottleneck

Basically it should be tuning on

1. Smoothness
2. Speed

Shine yourself here!

Smoothness(jank-free)

1. Instant go back(Page Stack/global state/api caching)
2. Instant go forward (Skeleton / loading indicator / above-the-fold)
3. Instant interaction response (A11y, passive listener, design guidlines)
4. Native-like Animation/Transitions/Gestures
5. Native-like UI components

Speed/Performance

1. preload / prefetch
2. Code splitting(Skeleton)
3. Caching / CDN
4. Service worker/offline
5. Lazy-load
6. Auto pager
7. Infinite scroll
8. SSR/initial data feed
9. Within viewport update(API .etc)

About Images

1. Compress
2. Lazy Load / placeholder
3. Progressive images
4. Use SVG for icons
5. Caching / http2

About API

1. Poll/Web Socket/SSE
2. BBF (api aggregating)
3. GraphQL
4. Caching / http2

RAIL model

1. Response (100ms)
2. Animation(frame within 10ms)
3. Idle (use idle time, 50ms)
4. Load (5 seconds)

Matrix

1. DOMContentLoaded
2. Load
3. First Contentful Paint
4. First Meaning Paint (deprecated)
5. Speed Index
6. First CPU Idle (ready to interact, deprecated)
7. Time To (fully) Interactive:
8. First Input delay
9. Total Blocking Time(from FCP to TTI)
10. Largest Contentful Paint (2.5s)

Stick to core values

Example from Facebook

Move fast

Be Bold

Focus On Impact

Be Open

Build Social Value

6. Trade-off, alternatives, TODO

Nothing is perfect.
Try to list up possible improvement ideas,
And things you wanana do if more time given.