2020 02 15 - Handout – System Design

Q1. Designing Twitter

Let's design a Twitter-like social networking service. Users of the service will be able to post tweets, follow other people, and favorite tweets.

Q2. Designing Yelp or Nearby Friends

Let's design a Yelp like service, where users can search for nearby places like restaurants, theaters, or shopping malls, etc., and can also add/view reviews of places.

Approaching System Design Questions

(Credits - Cracking the Amazon System Design Interview @ Amazon Bellevue)

Preface

System design questions cover fairly large and complex systems. It is impossible to enumerate all scenarios, tradeoffs etc in the limited time. Unlike the coding interview questions, system design questions do not have one optimal solution. You should think of a system design interview as if it were a brainstorming session. The interviewer will kick it off with vague, open ended requirements, and you the candidate will drive the conversation by asking relevant questions. The interviewer typically wear the hat of the "client/product manager" making decisions when asked. Sometimes, the interviewer will wear the hat of a peer reviewing your design, often pointing at weaknesses in your design. Always remember that the goal of this interview is to assess your ability to drive this discussion, the final product you design may not be as relevant. Consider feedback on your design an opportunity to explore alternatives. You don't need to be an expert in this function area. Focus on skills that you will need to develop to ace a system design interview.

9-Step process:

Step 0: Play back the question: Repeat the question and confirm the requirements.

Step 1: Requirements clarifications: Always ask questions to find the exact scope of the problem you are solving.

Step 2: System interface definition: Define what APIs are expected from the system. This'll also ensure if you haven't gotten any of the requirements wrong.

Step 3: Back-of-the-envelope estimation: It's always a good idea to estimate the scale of the system you are going to design.

Step 4: Define data model: Although it's not required early on, this will clarify how data will flow among different components of the system and later will also guide you towards data partitioning.

Step 5: High-level design: Draw a block diagram with 5-6 boxes representing core components of your system.

Step 6: Detailed component design: Dig deeper into 2-3 components; interviewers feedback should always guide you towards which parts of the system they wants you to explain further.

Step 7: Bottlenecks. Try to discuss as many bottlenecks as possible, and different approaches to mitigate them.

Step 8: Design Verification: Make sure your design works for things you set out to solve.