

3장. 시스템 설계 면접 공략법

🕒 Created	@September 21, 2022 8:46 PM
📌 Progress	DONE

- 3.1. 효과적인 면접을 위한 4단계 접근법
- 3.1.1. 1단계: 문제 이해 및 설계 범위 확정
 - 3.1.2. 2단계: 개략적인 설계안 제시 및 동의 구하기
 - 3.1.3. 3단계: 상세 설계
 - 3.1.4. 4단계: 마무리
 - 3.1.5. 면접에서 해야 할 것과 하지 말아야 할 것들

시스템 설계 면접은 두 명의 동료가 모호한 문제를 풀기 위해 협력하여 그 해결책을 찾아내는 과정에 대한 시뮬레이션이다.



시스템 설계 면접

- 설계 기술을 시연
- 설계 과정에서 내린 결정들에 대한 방어 능력
- 면접관의 피드백을 건설적인 방식으로 처리할 자질



면접관의 목표

- 설계 능력의 기술적 평가
- 협력에 적합한 사람인지 평가
- 압박이 심한 상황도 잘 헤쳐 나갈 자질이 있는지 평가
- 모호한 문제를 건설적으로 해결할 능력이 있는지 평가
- 부정적 신호(red flag)
 - 설계의 순수성(purity)에 집착 → 나머지 타협적 결정(tradeoff)을 도외시하고 과도한 엔지니어링(over-engineering)을 하는가
 - 완고함
 - 편협함



학습 TODO list



- 3.1. 효과적인 면접을 위한 4단계 접근법
- 3.1.1. 1단계: 문제 이해 및 설계 범위 확정
 - 3.1.2. 2단계: 개략적인 설계안 제시 및 동의 구하기
 - 3.1.3. 3단계: 상세 설계
 - 3.1.4. 4단계: 마무리
 - 3.1.5. 면접에서 해야 할 것과 하지 말아야 할 것들

3.1. 효과적인 면접을 위한 4단계 접근법

3.1.1. 1단계: 문제 이해 및 설계 범위 확정

요구사항을 완전히 이해하지 않고 답을 내놓는 행위는 아주 엄청난 부정적 신호(red flag)다.

엔지니어가 가져야 할 가장 중요한 기술 중 하나는 올바른 질문을 하는 것, 적절한 가정을 하는 것, 그리고 시스템 구축에 필요한 정보를 모으는 것이다.



요구사항을 정확히 이해하는 데 필요한 질문들

- 구체적으로 어떤 기능들을 만들어야 하나?
- 제품 사용자 수는 얼마나 되나?
- 회사의 규모는 얼마나 빨리 커지리라 예상하나? 석 달, 여섯 달, 일년 뒤의 규모는 얼마가 되리라 예상하는가?
- 회사가 주로 사용하는 기술 스택(technology stack)은 무엇인가? 설계를 단순화하기 위해 활용할 수 있는 기존 서비스로는 어떤 것들이 있는가?

- 예제: 뉴스피드(news feed) 시스템 설계



모바일 앱과 웹 가운데 어느 쪽을 지원해야 하나요? 아니면 둘 다일까요?



둘 다 지원해야 합니다.



가장 중요한 기능은 무엇인가요?



새로운 포스트(post)를 올리고, 다른 친구의 뉴스 피드를 볼 수 있도록 하는 기능입니다.



이 뉴스 피드는 시간 역순으로 정렬되어야 하나요? 아니면 다른 특별한 정렬 기준이 있습니까? 제가 특별한 정렬 기준이 있느냐고 묻는 이유는, 피드에 올라갈 포스트마다 다른 가중치가 부여되어야 하는지 알고 싶어서 인데요. 가령 친구의 포스트가 사용자 그룹(user group)에 올라가는 포스트보다 더 중요하다거나.



문제를 단순하게 만들기 위해, 일단 시간 역순으로 정렬된다고 가정합시다.



한 사용자는 최대 몇 명의 사용자와 친구를 맺을 수 있나요?



5000명입니다.



사이트로 오는 트래픽 규모는 어느 정도입니까?



일간 능동 사용자(daily active user, DAU)는 천만 명입니다.



피드에 이미지나 비디오도 올라올 수 있나요? 아니면 포스트는 그저 텍스트입니까?



이미지나 비디오 같은 파일도 포스트 할 수 있어야 합니다.

3.1.2. 2단계: 개략적인 설계안 제시 및 동의 구하기

개략적인 설계안을 제시하고 면접관의 동의를 얻어라.

- 면접관을 마치 팀원인 것처럼 대하면서 설계안에 대한 최초 청사진을 제시하고 의견을 구해야 함
- 화이트보드나 종이에 핵심 컴포넌트(클라이언트(모바일/웹), API, 웹 서버, 데이터 저장소, 캐시, CDN, 메시지 큐 등)를 포함하는 다이어그램을 그려야 함
- 면접관에게 개략적 추정이 필요한지 미리 확인하고, 필요하다면 최초 설계안이 시스템 규모에 관계된 제약사항들을 만족하는지를 개략적으로 소리 내어 설명하며 계산해 볼 것
- 시스템의 구체적 사용 사례를 살펴보며 개략적 설계안을 잡아 나가고 미처 고려하지 못한 에지 케이스(edge case)를 발견할 수 있음
- 큰 규모의 설계 문제는 API 엔드포인트(endpoint)나 데이터베이스 스키마도 보일 필요 없음 → 면접관의 의견을 물어보는 것도 좋음



1-2단계에서 달성해야 하는 것

- 시스템에서 전반적으로 달성해야 할 목표와 기능 범위 확인
- 전체 설계의 개략적 청사진 마련
- 해당 청사진에 대한 면접관의 의견 청취
- 상세 설계에서 집중해야 할 영역들 확인

- 예제: 뉴스피드(news feed) 시스템 설계

1. 개략적 설계: 이 설계는 두 가지 처리 플로(flow)로 나눠 생각해 볼 수 있음

- **피드 발행(feed publishing)** : 사용자가 포스트를 올리면 관련된 데이터가 캐시/데이터베이스에 기록되고, 해당 사용자의 친구(friend) 뉴스 피드에 뜨게 됨
- **피드 생성(feed building)** : 어떤 사용자의 뉴스 피드는 해당 사용자 친구들의 포스트를 시간 역순으로(최신 포스트부터 오래된 포스트 순으로) 정렬하여 만듦

2. 각 플로의 개략적 설계안 그리기

3.1.3. 3단계: 상세 설계

- 설계 대상 컴포넌트 사이의 우선순위 정하는 것
 - 면접관이 집중 했으면 하는 영역을 알려주기도 함
- 선임급 개발자 면접의 경우, 시스템의 성능 특성에 대한 질문을 할 것
 - 질문 내용이 시스템의 병목 구간이나 자원 요구량 추정치에 초점이 맞춰져 있음
- 면접관은 특정 시스템 컴포넌트의 세부사항을 깊이 있게 설명하는 것을 보길 원함
 - ex. 단축 URL 생성기(URL shortener) 설계 → 해시 함수의 설계를 구체적으로 설명하는 것을 듣고 싶어 할 것
 - ex. 채팅 시스템 설계 → 어떻게 하면 지연시간(latency)을 줄이고 사용자의 온/오프라인 상태를 표시할 것인지 듣고 싶어 할 것
- 시간관리: 긍정적 신호(signal)를 전달하는 데 집중하고 불필요한 세부사항에 시간을 쓰지 말 것
 - ex. 페이스북에서 뉴스 피드의 순위를 매기는 데 사용되는 EdgeRank 알고리즘에 대해 이야기하는 것을 바람직하지 않음 → **규모 확장 가능한 시스템을 설계할 능력이 있다는 것을 입증하는게 중요**

- 예제: 뉴스피드(news feed) 시스템 설계

1. 2단계에서 개략적으로 설계한 플로(flow)를 더 깊이 탐구하기

- **피드 발행(feed publishing)**
- **피드 생성(feed building)** : 뉴스 피드 가져오기(news feed retrieval)

2. 각 플로에 대한 상세 설계안 그리기

3.1.4. 4단계: 마무리

- 면접관이 설계 결과물에 관련된 몇 가지 후속 질문을 던질 수 있음(follow-up questions)

- 스스로 추가 논의를 진행할 수 있음



4단계 지침

- 면접관이 시스템 병목구간, 혹은 좀 더 개선 가능한 지점을 찾아내라 주문할 수 있음
 - 설계가 완벽하다거나 개선할 부분이 없다는 답을 하지 말 것
 - 비판적 사고 능력을 보이고 좋은 인상을 남길 기회
- 만든 설계를 한번 다시 요약하는 것도 도움이 될 수 있음
 - 여러 해결책을 제시한 경우 특히 중요함
- 오류가 발생하면 무슨 일이 생기는지(서버 오류, 네트워크 장애 등)를 따져보는 것도 좋음
- 운영 이슈도 논의할 가치가 있음
 - 메트릭, 로그는 어떻게 수집하고 모니터링할 것인가
 - 시스템은 어떻게 배포해(roll-out) 나갈 것인가
- 미래에 닥칠 규모 확장 요구에 어떻게 대처할 것인지 논의하는 것도 좋음
 - ex. 현재 설계로 백만 사용자는 능히 감당할 수 있다고 해 보자. 천만 사용자를 감당하려면 어떻게 해야 하는가?
- 시간이 좀 남을 경우: 필요하지만 다루지 못했던 세부적 개선사항들을 제안할 수 있음

3.1.5. 면접에서 해야 할 것과 하지 말아야 할 것들



해야 할 것

- 질문을 통해 확인(clarification)
- 문제의 요구사항 이해
- 정답이나 최선의 답안은 없음
- 면접관이 사고 흐름을 이해할 수 있게 할 것
- 여러 해법을 함께 제시
- 개략적 설계에 면접관이 동의하면 가장 중요한 컴포넌트부터 세부사항을 설명
- 면접관의 아이디어를 이끌어 낼 것
- 포기하지 말 것



하지 말아야 할 것

- 전형적인 문제도 대비하지 않고 면접장에 가지 말 것
- 요구사항이나 가정들이 분명하지 않은 상태에서 설계를 제시하지 말 것
- 처음부터 특정 컴포넌트의 디테일에 집착하지 말 것
- 힌트 청하기와 소통을 주저하지 말 것
- 설계안을 내놓는다고 면접이 끝이 아님



시간 배분

- 문제 이해 및 설계 범위 확정: 3분~10분
- 개략적 설계안 제시 및 동의 구하기: 10분~15분
- 상세 설계: 10분~25분
- 마무리: 3분~5분