

# 3장 - 시스템 설계 면접 공략법

시스템 설계 면접은 일반적으로 **널리 알려진 제품 X를 설계해 보라** 라는 식이다.

널리 알려진 제품을 만들기 위해 수백, 수천 그 이상의 엔지니어가 투입되었는데, 한 시간에 그걸 그대로 설계하는 것은 불가능하다.

시스템 설계 문제를 내는 이유는 다음과 같다.

## 시스템 설계 문제를 내는 이유.

- 두 명의 동료는 모호한 문제를 풀기 위해 협력하여 그 해결책을 찾아내는 과정에 대한 시뮬레이션이다.

이 문제에는 정해진 **결말도 없고, 정답도 없다.** (ex. **스케일 업**, **스케일 아웃** 모두 문제를 해결할 수도 있음)

최종적으로 도출된 설계안은 설계 과정에 들인 노력에 비해 그다지 중요하지 않다.

이 면접은 **설계 기술을 시연하는 자리** 이고, **설계 과정에서 내린 결정들에 대한 방어 능력을 보이는 자리** 이며,

**면접관의 피드백을 건설적인 방식으로 처리할 자질** 이 있음을 보이는 자리이다.

- 시스템 설계 면접은 실제 지원자의 **설계 능력의 기술적 측면을 평가하는 자리 그 이상** 이다.

시스템 설계 면접이 잘 징해되면, 지원자가 **협력에 적합한 사람 인지, 압박이 심한 상황도 잘 헤쳐 나갈 자질** 이 있는지, **협력에 적합한 사람 인지, 압박이 심한 상황도 잘 헤쳐 나갈 자질** 이 있는지, **모호한 문제를 건설적으로 해결할 능력** 이 있는지 확인할 수 있다. 이때 **좋은 질문을 던질 능력** 이 있는지도 중요하다.

- 좋은 면접관은 **부정적 신호(red flag)**도 놓치지 않는다.

설계의 순수성(purity)에 집착한 나머지 **타협적 결정(tradeoff)**을 도외시하고 **과도한 엔지니어링(over-engineering)**을 하고 마는 엔지니어들이 현업에도 많다.

그런 엔지니어들은 과도한 엔지니어링의 결과로 시스템 전반의 비용이 올라간다는 사실을 알아채지 못하는 일이 많은데, 그 결과로 상당수 회사들은 값비싼 대가를 치르게 된다.

면접관에게 이런 엔지니어들과 같은 경향이 있다고 보이지 말자.

부정적 신호 중 **오버 엔지니어링** 말고, **완고함**, **편협함** 같은 것들도 있다. 이런 부분들을 조심하자.

## 효과적인 면접을 위한 4단계 접근법

여러번 반복하지만 시스템 설계 면접은 전부 제각각이고, 좋은 설계 면접은 정해진 결말도 정답도 없다.

하지만 그

절차나 범위에는 공통적인 부분이 있다.

### 1단계 - 문제 이해 및 설계 범위 확정

"호랑이는 왜 그르렁댈까요?"

뒷자리에서 손이 번쩍 올라갔다.

"그래, 지미?" 선생이 대답했다.

"배가 고파서 입니다."

"잘했어요, 지미."

지미의 어린 시절은 질문에 맨 처음 손을 드는 학생이고, 답을 알든 모르든 문제에 덤비는 것을 좋아하는 학생이다.

또한 우수한 학생이었어서 모든 문제에 대한 답을 빨리 알아내는데 자부심이 있다.

모든 문제를 1등으로 풀고 시험장을 나서는 학생이 바로 지미였다. (경진대회도 마찬가지)

하지만 **면접장에서는 지미가 되면 안 된다.**

**시스템 설계 면접을 볼 때는 생각 없이 바로 답을 내서는 좋은 점수를 받을 수 없다.**

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

면접은 퀴즈 쇼가 아니며, 정답 따위는 없다는 걸 상기하자.

그러니까 답부터 들이밀지 말라. 속도를 늦추고, 깊이 생각하고 질문하여 요구사항과 가정들을 분명히 하자.

저자는 이 부분에 대한 중요성을 강조 강조 강조해도 모자람이 없다고 한다.

엔지니어가 가져야 할 중요한 기술 중 하나는 올바른 질문 을 하는 것, 적절한 가정 을 하는 것,

시스템 구축에 필요한 정보 를 모으는 것이다.

면접관은 지원자에게 질문을 던지면 질문에 대한 답을 바로 내놓거나, 아니면 여러분 스스로 어떤 가정을 하기를 주문할 것이다. 후자의 경우에는 그 가정을 화이트보드나 종이에 적어두어야 한다. (나중에 필요해질 수 있음)

## | 어떤 질문을 해야 될까?

- 요구사항을 정확히 이해하는 데 필요한 질문을 해라. (아래 질문과 같은)
  - 구체적으로 어떤 기능들을 만들어야 하나?
  - 제품 사용자 수는 얼마나 되나?
  - 회사의 규모는 얼마나 빨리 커지리라 예상하나? 석 달, 여섯 달, 일년 뒤의 규모는 얼마나 되리라 예상?
  - 회사가 주로 사용하는 기술 스택은 무엇인가?
    - 설계를 단순화하기 위해 사용할 수 있는 기존 서비스로는 어떤 것들이 있는가?
- ▼ 개인적인 의견으로 설계하는데 이게 맞나? 싶은 부분들이 있을 때 물어봐야 될 신호라고 생각된다.

- 캐시 TTL 정책 을 어떻게 할 것인지에 대해 질문

(1) 대략 2초를 생각한다. 라고 답변한 뒤 →

(2) 이 서비스를 사용하는 사용자가 많으려나? 라는 생각을 하면서  
사용자가 그렇게 많지 않을 것 같은데 너무 적게 줬나? 싶어서 →

(3) 10초로 수정해도 될 것 같다.

10 초사이의 update가 발생하지 않을 것 같다. 등 이상한 소리를 한 전적이 있음

애초에 (2) 번에 관련된 생각이 들었을 때 즉, 이게 맞나? 라는 듯한 물음표가

들었을 때  
질문을 해야되는 신호라고 생각된다.

## | (1단계, 문제 이해 및 설계 범위 확정) 예제

뉴스 피드(news feed) 시스템을 설계하라는 요구를 받았다면 요구사항을 분명히 하기 위한 질문을 던져야 된다.

지원자 : 모바일 앱과 웹 앱 가운데 어느 쪽을 지원해야 하나요? 아니면 둘 다

면접관 : 둘 다 지원해야 됩니다.

지원자 : 가장 중요한 기능은 무엇인가요?

면접관 : 새로운 포스트(post)를 올리고, 다른 친구의 뉴스 피드를 볼 수 있도록

지원자 : 이 뉴스 피드는 시간 역순으로 정렬되어야 하나요? 아니면 다른 특별한 정렬 기준을 묻는 이유는, 피드에 올라갈 포스트마다 다른 가중치가 부여되어야 하는지 알고 싶어서 인데요.  
가령 가까운 친구의 포스트 사용자 그룹(user group)에 올라가는 포스트

면접관 : 문제를 단순하게 만들기 위해, 일단 시간 역순으로 정렬된다고 가정합시다.

지원자 : 한 사용자는 최대 몇 명의 사용자와 친구를 맺을 수 있나요?

면접관 : 5000명 입니다.

지원자 : 사이트로 오는 트래픽 규모는 어느 정도입니까? (멘토링때 대규모라고

면접관 : 일간 능동 사용자(DAU)는 천만 명입니다.

지원자 : 피드에 이미지나 비디오도 올라올 수 있나요? 아니면 포스트는 그저 텍스트

면접관 : 이미지나 비디오 같은 미디어 파일도 포스트 할 수 있어야 합니다.

## 2단계 : 개략적인 설계안 제시 및 동의 구하기

이번 단계에서 초점을 맞추어야 할 것은 개략적인 설계안을 제시하고, 면접관의 동의를 얻는 것이다.

이 과정은 면접관과 협력하며 진행하면 좋다.

- 설계안에 대한 최소 청사진을 제시하고 의견을 구하라.
  - 면접관을 마치 팀원인 것처럼 대하자.
  - 좋은 면접관들은 지원자들과 대화하고 설계 과정에 개입하기를 즐긴다.
- 화이트보드나 종이에 핵심 컴포넌트를 포함하는 다이어그램을 그려라.
  - 클라이언트(모바일/웹), API, 웹 서버, 데이터 저장소, 캐시, CDN, 메시지 큐 같은 것들이 포함된다.
- 최초 설계안이 시스템 규모에 관계된 제약사항들을 만족하는지를 개략적으로 계산해보자.
  - 계산 과정은 소리내어 설명하자.
  - 이런 개략적 추정이 필요한지는 면접관에게 미리 물어보도록 하자.

가능하다면 시스템의 구체적 사용 사례도 몇 가지 살펴보자.

구체적 사용 사례를 통해

미처 고려하지 못한 엣지 케이스(edge case)를 발견할 가능성이 있음

이 단계에서 **API 엔드포인트(endpoint)**나 **데이터베이스 스키마**도 보여야 하는가?

- 질문에 따라 다르지만, **구글 검색 엔진을 설계하라** 와 같은

큰 규모의 설계 문제라면 이 단계에서 다루기에는 지나치게 세부적인 내용일 것이다.

하지만 멀티 플레이어 포커 게임의 백엔드를 설계하라는 질문이라면 괜찮을 수 있다.

면접관의 의견을 물어보자.

(2단계, 개략적인 설계안 제시 및 동의 구하기) 예제

뉴스 피드 시스템을 설계하라는 질문을 그대로 활용하여, 개략적 설계는 어떻게 만드는지 살펴본다.

지금 당장에 이해할 필요는 없다.

**상세한 내용은 11장에서 설명한다.**

개략적으로 보자면 이 설계는 **두 가지 처리 플로(flow)**로 나눠 생각해 볼 수 있다.

**피드 발행(feed publishing)**과 **피드 생성(feed building)**이 그것이다.

- **피드 발행**

- 사용자가 포스트를 올리면 관련된 데이터가 캐시/데이터베이스에 기록되고, 해당 사용자의 친구 뉴스 피드에 뜨게 된다.

- **피드 생성**

- 어떤 사용자의 뉴스 피드는 해당 사용자 친구들의 포스트를 시간 역순으로(최신 포스트부터 오래된 포스트 순으로) 정렬하여 만든다.

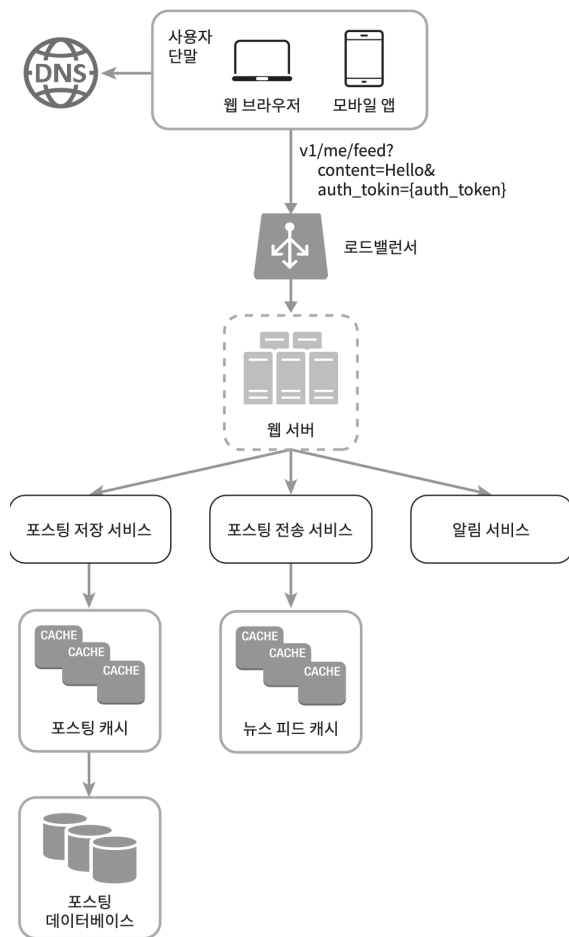


그림 3-1  
피드 발행 플로우

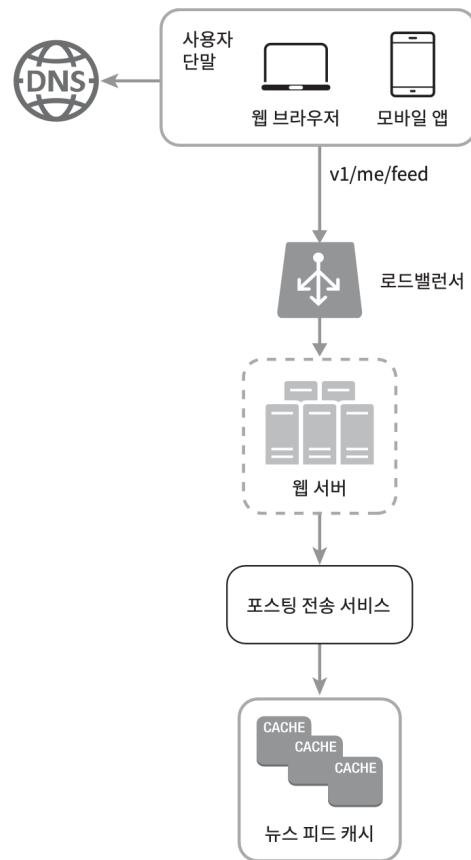


그림 3-2  
피드 생성 플로우

### 3단계 : 상세 설계

이 단계로 왔다면 면접관과 다음 목표는 달성한 상태이다.

- 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

이제 **목표**로 해야 될 일은 면접관과 함께 **설계 대상 컴포넌트 사이의 우선순위를** 정하는 것이다.

똑같은 면접이랑 있을 순 없다. (예제를 많이 공부해도 면접관의 스타일에 따라 다름)

특히 선임급 개발자라면 시스템 성능 특성에 대한 질문이 있을 수 있고, 시스템의 병목 구간이나 자원 요구량 추정치에 초점이 맞춰질 수 있다.

하지만, **대부분의 경우** 면접관은 **특정 시스템 컴포넌트들의 세부 사항을 깊이 있게 설명**하는 것을 원한다.

ex)

**단축 URL 생성기** 설계에 관련한 것이라면 **해시 함수의 설계** 를 구체적으로 설명하는 것을 원할 수 있음

**채팅 시스템** 에 관한 문제라면 어떻게 **지연 시간을 줄이고 사용자의 온/오프라인 상태를 표시** 할 것인지에 대해

시간 관리에도 신경써야 된다. 사소한 세부사항을 설명하느라 시간이 다 지날 수도 있다.

면접관에게 긍정적 신호를 전달하는 데 집중해야 된다.

ex) 메타에서 뉴스 피드의 순위를 매기는 데 사용되는 EdgeRank 알고리즘에 대해 이야기 하는 것은 좋지 않다.

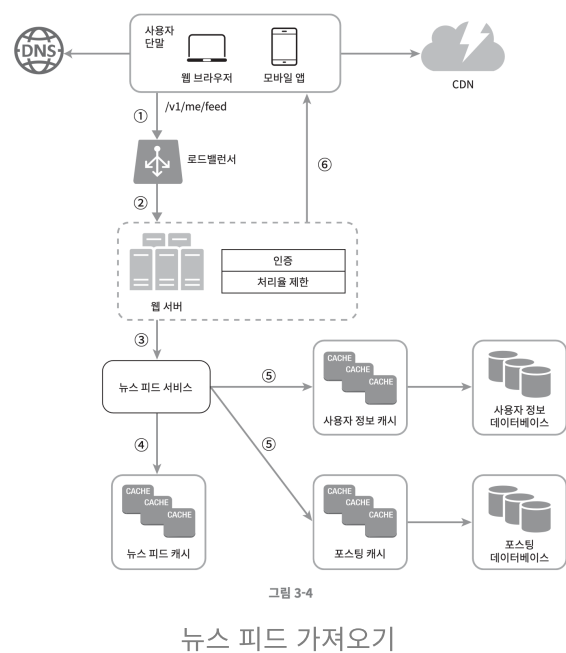
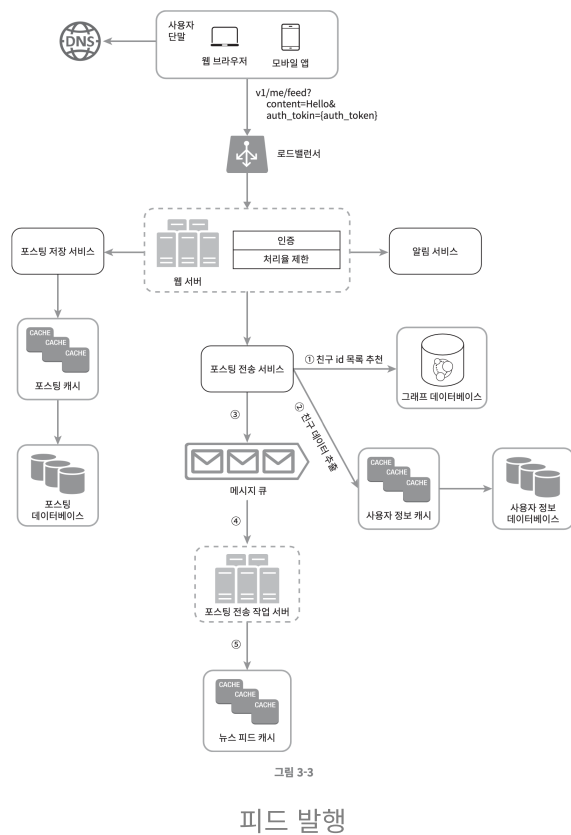
## |(3단계, 상세 설계) 예제

뉴스 피드 시스템에 대한 개략적 설계를 2단계까지 마쳤고, 면접관도 해당 설계에 만족하고 있다.

다음 두 가지 중요한 용례를 보다 깊이 탐구해야 한다. (11장에서 자세히 살펴보자)

1. 피드 발행
2. 뉴스 피드 가져오기(news feed retrieval)





## 4단계 : 마무리

이 단계에서 면접관은 실제 결과물에 관련된 몇 가지 후속 질문을 던질 수 있고(follow-up questions)

스스로 추가 논의를 진행하도록 할 수도 있다. 다음 지침을 활용하자.

- 면접관이 시스템 병목구간, 혹은 좀 더 개선 가능한 지점을 찾아내라 주문할 수 있다.  
여기서 설계가 완벽하다거나 개선할 부분이 없다는 답은 하지 말자.  
언제나 개선할 점은 있다. 이런 질문은 비판적 사고 능력을 보이고, 마지막으로 좋은 인상을 보일 수 있다.
- 만든 설계를 다시 요약해주는 것도 도움이 될 수 있다.  
여러 해결책을 제시한 경우에는 특히 중요하다.  
긴 면접 세션이 끝난 뒤 면접관의 기억을 환기시켜주는 효과가 있다.
- 오류가 발생하면 무슨 일이 생기는지(서버 오류, 네트워크 장애 등) 따져보면 흥미롭다.
- 미래에 닥칠 규모 확장 요구에 따라 어떻게 대처할 것인지도 흥미로운 주제다.

ex) 현재 설계로 백만 사용자는 감당할 수 있는데, 천만 사용자를 감당하려면 어떻게 할 수 있을까?

- 시간이 좀 남았다면 필요하지만 다루지 못했던 **세부적 개선사항**을 제안할 수 있다.

## 해야할 것

- **질문을 통해 확인하라(clarification)**. 스스로 내린 가정이 옳다 믿고 진행하지 말라.

- 문제의 요구사항을 이해하라.

- 정답이나 최선의 답안 같은 것은 없다는 점을 명심하라.

스타트업을 위한 설계안과 수백만 사용자를 지원해야 하는 중견 기업을 위한 설계안이 같을리 없다.

요구사항을 정확하게 이해했는지 다시 확인하라.

- 면접관이 여러분의 사고 흐름을 이해할 수 있도록 하라. **면접관과 소통**하라.

- 가능하다면 여러 해법을 함께 제시하라.

- **개략적 설계에 면접관이 동의하면**, 각 **컴포넌트의 세부사항을 설명**하기 시작하라.

이때

**가장 중요한 컴포넌트부터 진행**하라.

- 면접관의 아이디어를 이끌어 내라. 좋은 면접관은 지원자와 같은 팀원처럼 협력하난.

- 포기하지 말라.

## 하지 말아야 할 것

- 전형적인 면접 문제들에도 대비하지 않은 상태에서 면접장에 가지 말라.
- 요구사항이나 가정들을 분명히 하지 않은 상태에서 설계를 제시하지 말라.
- **처음부터 특정 컴포넌트의 세부사항을 너무 깊이 설명하지 말라.**

**개략적 설계를 마친 뒤 세부사항으로 나아가라.**

- 진행 중에 막혔다면, **힌트를 청하기를 주저하지 말라.**
- 다시 말하지만, **소통을 주저하지 말라.** 침묵 속에 설계를 진행하지 말라.

- 설계안을 내놓는 순간 면접이 끝난다고 생각하지 말라.  
면접관이 끝났다고 말하기 전까지는 끝난 것이 아니다.  
**의견을 일찍, 자주 구하라.**

## 시간 배분

시스템 설계 면접은 보통 매우 광범위한 영역이라 45분 또는 한 시간은 충분하지 않을 수 있다.

따라서 시간 관리를 잘 하는 것이 중요하다.

### 상술한 각 단계에 어느 정도의 시간을 배분하는 것이 좋을까? - 45분의 시간이 주어진다고 가정

- 1단계 - 문제 이해 및 설계 범위 확정 : 3 ~ 10분
- 2단계 - 개략적 설계안 제시 및 동의 구하기 : 10 ~ 15분
- 3단계 - 상세 설계 : 10 ~ 25분
- 4단계 - 마무리 : 3 ~ 5분