문제 1

문제 1. 다음과 같은 조건의 CRUD API 스펙을 정의하세요.

> a. 사용자 정보 관리 하는 API 입니다.
> b. 사용자 속성은 id/pw/name/role 입니다.
> c. 삭제는 조건을 가질 수 있습니다.
> d. API 스펙은 아래 내용에 작성하세요.

- API 스펙 : default URL : api/v1

전체 조회	URL	
	params	
	body	
단건 조회	URL	
	params	
	body	
추가	URL	
	params	
	body	
수정	URL	
	params	
	body	
삭제	URL	
	params	
	body	

답변1

1. 전체 조회

1. Method : GET
2. URL : /api/v1/users
3. params:
    1. name (%string%) : like 검색을 위한 이름 문자열
    2. role (List) : 해당하는 역할 배열 (디폴트: 전체 검색)
    3. page (int) : 페이지 번호 (디폴트:1)

    4. limit (page) : 페이지당 유저 수 (디폴트: 보통 10~20이지만 프론트와 협의에 따라 다름)
    5. sort {key(name,role,id) : desc | asc} : (디폴트: 보통 등록 최신수 기준, 프론트와 협의에 따라 다름)
    6. stream (boolean) : SSE 로 전달받을지 여부
4. Body: empty

2. 단건 조회

1. Method : GET
2. URL : /api/v1/users/{id}
3. params : null
4. body : empty

3. 추가

1. Method : POST
2. URL : /api/v1/users
3. params : null
4. body :

```
{
 name : string, // not null
 role : List<String>, // not null, empty 여부는 기본 역할을 어떻게 부여할지 정책에 따라 다를듯
 pw : string // not null, 경우에 따라 비밀번호는 자동 생성해서 nullabe 로 설정도 가능
}
```

4. 수정

> **Info**
>
> 정석적인 경우에는 일부 수정을 PATCH 를 사용하는 것이지만, 현업에서는 방화벽에서 잘 사용하지 않는 메서드는 차단하는 경우가 많기에 전체 수정, 일부 수정도 PUT 으로 통일하는게 좋음

1. Method : PUT
2. URL : /api/v1/users/{id}
3. body :

```
{
 name : string, // nullable
 role : List<string, // nullable
```

```
  pw : string // nullable, 보통 비밀번호를 변경하는건 API 자체를 분리해서 사용하긴함
}
```

5. 삭제

> **Info**
>
> 예전에 대학병원 서버로 개발할때 개인정보 보호때문에 GET, POST 만 허용하는 경우도 있음 그럴 때는 메서드를 POST 로 사용

1. Method : DELETE
2. URL : /api/v1/users/{id}
3. body : empty

문제 2

문제 2. 기존의 코드를 유지보수(기능 추가 / 수정 / bug-fix 등)를 할 때 고려 할 사항을 경험 중심으로 작성해 주세요.

a. 코드는 초기 개발 모델 코드 입니다. b. 코드 중 일부는 공통 모듈로 개발 되어 있습니다.

> **Info**
>
> 현 회사에 처음 업무 투입되었을때 10년 넘은 레거시 프로젝트에 2주만에 신규 기능을 추가했었는데 그 때 경험을 나열함

1. 영향 범위 파악
    1. 서비스 파악 :
        1. 프론트 페이지가 있다면 유지보수와 관련된 기능을 테스트 아이디로 직접 사용해야함, API 만 있다면 관련된 API 들을 파악해야함
        2. 관련된 API 의 사용처를 미리 파악해 영향 범위를 고려해야함
    2. 코드베이스 :
        1. API : 전체를 파악할 필요는 없고 API 작성 코드를 참조하는 함수, 모듈등을 파악해야함 (문서화가 잘 되어 있다면 문서를 참고하는 것이 좋지만 개인적으로 코드가 가장 정확한 문서라고 생각)
        2. 코드 사용처 파악 : 코드가 maven 같은 사내 라이브러리 서버에 올라가 있고 이를 사용하는 곳이 다양하다면, 이를 어디서 사용하는지 파악해야함

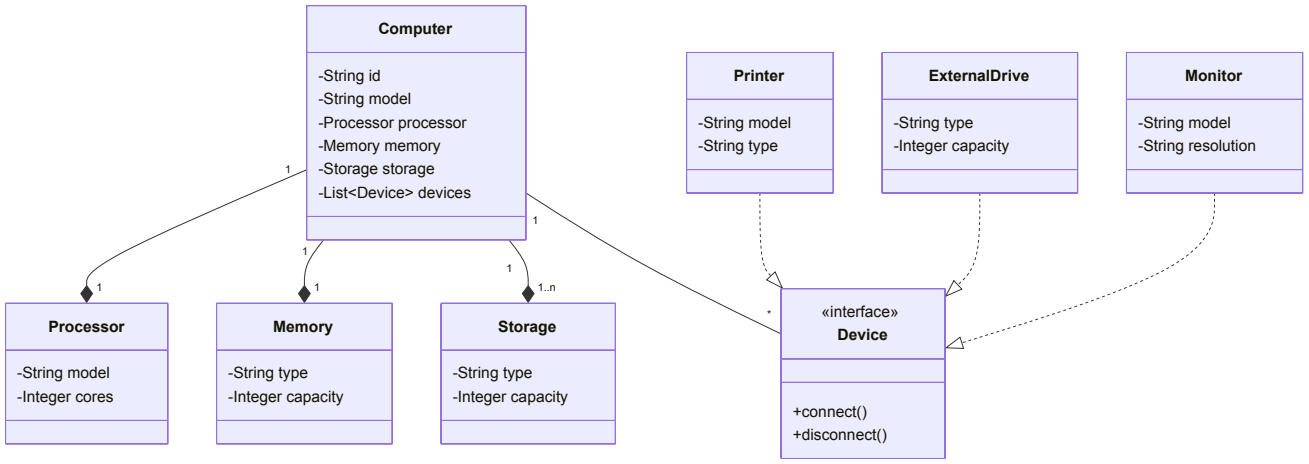
3. 인프라 : 코드와 관련된 DB, 메시징 시스템, 데몬 등 관련된 인프라에 대해서 파악

2. 개발
    1. 계획서 작성
        1. 파악한 영향 범위를 토대로 개발 계획서를 작성해 담당자 공유
        2. 이 과정에서 공통 모듈 코드를 수정하면 영향이 커서 독립적인 모듈에 유지 보수 기능을 추가해야한다는 등 구체적인 개발 계획이 수립 됨
    2. 계획서를 바탕으로 개발
3. 배포
    1. 영향 범위 모니터링 : 배포시에는 처음에 분석한 영향 범위를 바탕으로 혹시 장애가 생길지 모니터링하는 과정이 필요

문제 3

문제 3. 아래 주제 중 1개를 선택하여 데이터 객체를 모델링해주세요.

Computer / Server / Switch / LB / SmartPhone



문제 4

문제 4. 조건부 Rendering을 할 때 사용하는 방법을 설명해 주세요.

a. 화면에 제공해야 되는 정보는 티켓 종류, 진행 상태에 따라 제공되는 정보가 변경됩니다.
b. 언어적 제한은 없습니다. (React, VueJs, JSP, jQuery 등)
c. 경험 위주의 Rendering 방법을 설명해 주세요

- 티켓 종류와 상태를 변수로 저장
- 변수 조합에 해당하는 데이터 확인
- 데이터 없는 경우 서버에서 fetch 요청
- fetch 중 로딩 상태 표시
- 데이터 수신 후 화면에 결과 렌더링 후 변수 조합에 따라 랜더링 데이터 저장
- 변수 조합에 해당하는 렌더링 데이터가 있는 경우 fetch 없이 표기
- 필요에 따라 새로고침 버튼 등을 추가

문제 5

문제 5. A시스템은 API 호출 시 다음과 같은 제약 사항을 가지고 있습니다. Token을 갱신하기 위한 방법에 대해서 설명해 주세요.

a. 사내의 통합정보 조회 시스템에 A시스템을 연동하여 사용자들에게 정보를 제공해야 됩니다.
b. A시스템은 로그인 시마다 인증 로그가 남아 Token 방식을 통해 정보를 조회해야 됩니다.
c. A시스템은 로그인 시 Token이 발급됩니다.
d. 발급된 Token은 5분 단위로 refresh API 호출 하여 갱신해야 됩니다.

Info

사내 통합 정보 조회 시스템과 A 시스템의 연동이 토큰 발급에 어떤 영향을 끼치는지 파악이 안되므로, A 시스템 토큰 갱신에만 집중해서 답변을 작성했습니다.

1. 클라이언트 최초 인증 시 A 시스템에 로그인해 토큰을 발급
2. 클라이언트가 A 시스템과 작업을 계속 진행하는 경우 API 요청에서 401 과같은 토큰 만료 응답이 오면 refresh API 를 호출해 토큰을 재발급하고 다시 요청, (개인적으로 이를 구현할때는 axios 인터셉터를 사용해 자동으로 토큰을 재발행(최대 3회시도))

문제 6

문제 6. 운영 중인 API 서버에서 아래와 같은 현상을 보이고 있습니다. 지원자 분은 어떻게 조치를 하시겠습니까? 조치 순서와 방법을 설명해 주세요. (원인분석 방법, 문제해결 방법 등 모든 행위를 포함합니다.)

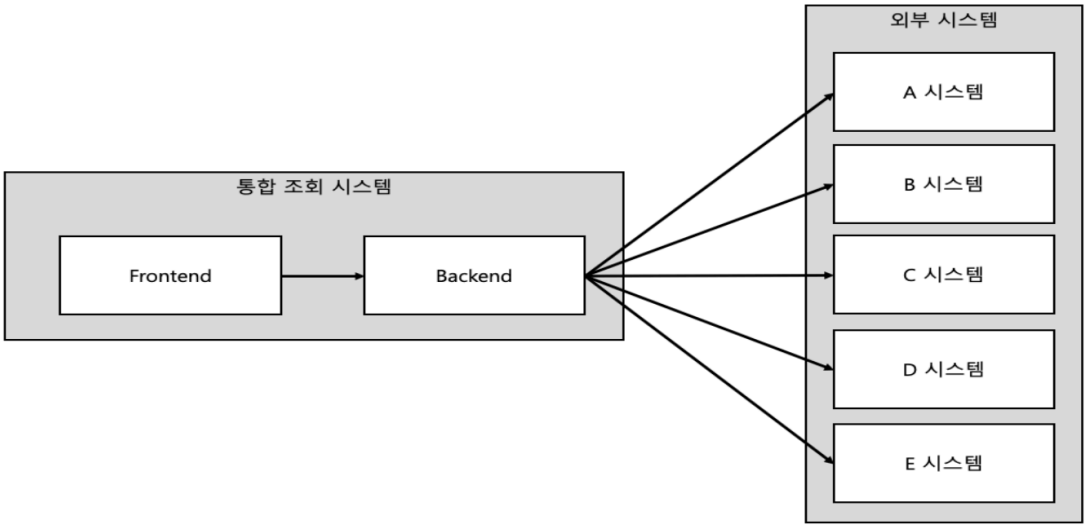
a. 모든 응답에서 Timeout이 발생하고 있음. b. CPU 사용량이 평균 80% 이상을 유지하고 있음 c. Memory 사용량은 변화가 없음.

- 원인 분석
  - 서버에 접속해 top 를 찍어 CPU 사용률이 높은 프로세스 확인 (보통 이런 경우 서버 프로세스가 문제)
  - 사용률이 높아지는 시기를 확인
  - 사용률이 높아질때 서버 로그를 확인, 사용률의 원인이되는 API 나 CRON을 추적, 모든 응답에서 Timeout 이 발생한다면, 각 요청이 공용으로 사용하는 부분에서 병목이 있을 가능성이 매우 큼 ex. 커넥션 풀이 모두 active 한 상태 (실제 경험)
- 문제해결 방법
  - 문제 해결 방법은 원인에 따라 상이함
  - 만약 데드락, 커넥션풀 병목 등 서버 자체의 이슈라면 긴급 재실행을 통해 초기화
  - 외부 API 또는 인프라 장애 (ex. 사내 API, Kafka , DB 타임 아웃)인 경우 담당 부서에 즉시 공유해야함

문제 7

문제 7. 여러 시스템에 접속하기 위한 GW API 서버를 개발할 때 고려해야 되는 항목과 이유에 대해서 설명해 주세요.

a. 제시된 시스템의 아키텍처를 변경해도 됩니다.
b. 개발시 고려 할것, 효율적인 개발 방법 등 모든 것을 작성해주세요



1. GW API 가 여러 시스템에 연결되어있는 만큼 트래픽을 감당할 수 있는지 고려
2. 트래픽 부하때문에 GW API 는 스케일 아웃을 적용할 가능성이 높으며 이에 따라 Stateless 기반의 설계를 하는 것이 중요 ex. 인증 방식을 세션 아이디에서 JWT 등을 사용
3. 여러 외부 시스템을 통합해 사용하는 만큼 기능마다 Network I/O 가 많이 발생하고 속도 이슈가 생길 것 같아 캐싱 기능을 적극적으로 도입
