

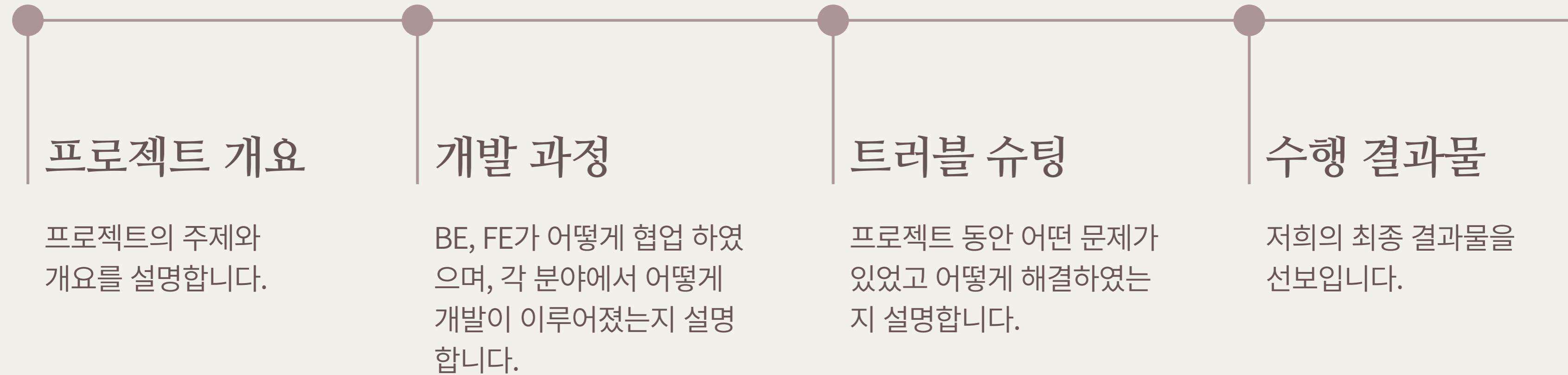


팬트리(Pantry)

Pantry

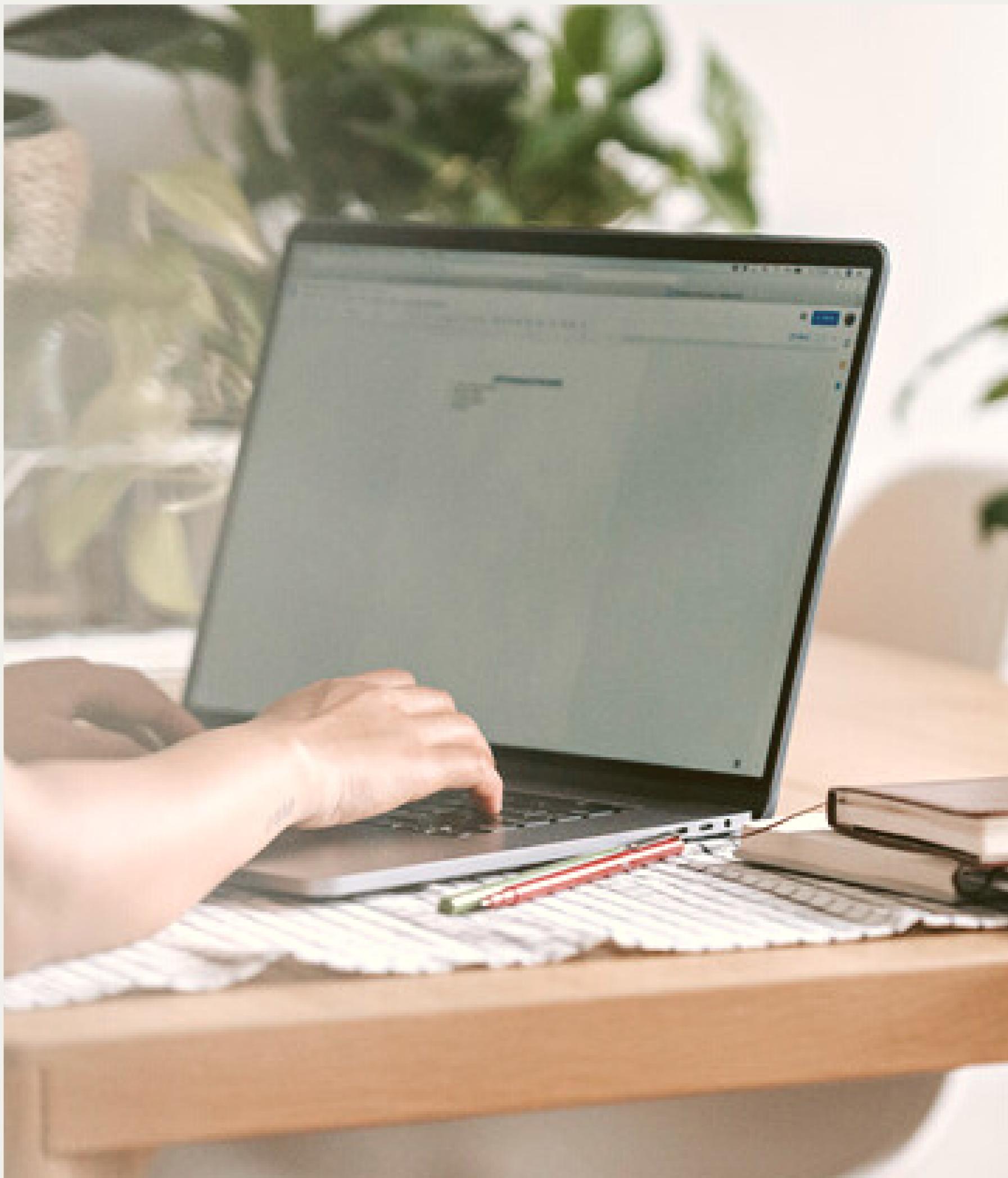
미니 프로젝트 6조

CONTENTS



프로젝트 개요

- 프로젝트의 주제
 - 연차와 당직을 관리
 - 새로운 유저의 유입을 검토하고 관리
- 프로젝트 개요
 - 팬트리(Pantry) -> 탕비실
 - 회사원의 휴식 공간인 탕비실의 이미지를 기반으로 커피의 색상을 사용하여 디자인



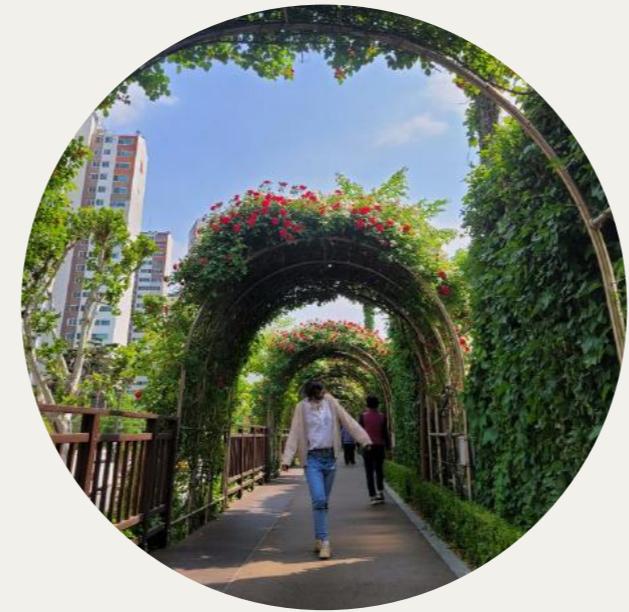
Front Engineer



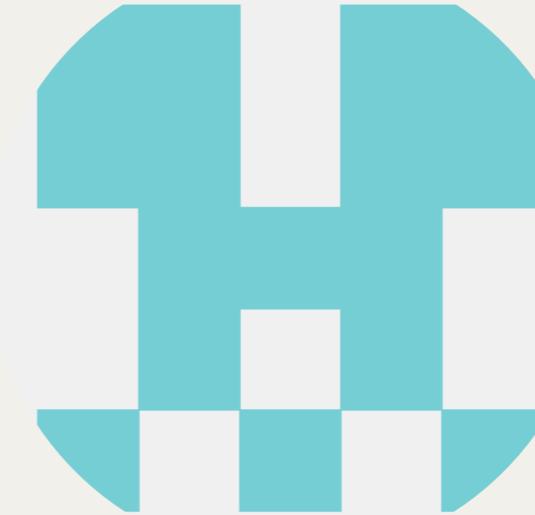
박희수(조장)



임성열



김선미



김민기



정윤조

메인 페이지
캘린더 모달 컴포넌트
페이지 배포

유저 관리 페이지
유저 권한 페이지
모달 컴포넌트

연차/당직
승인 및 내역 페이지

전체 레이아웃
마이 페이지

로그인 페이지
회원가입 페이지
에러 페이지

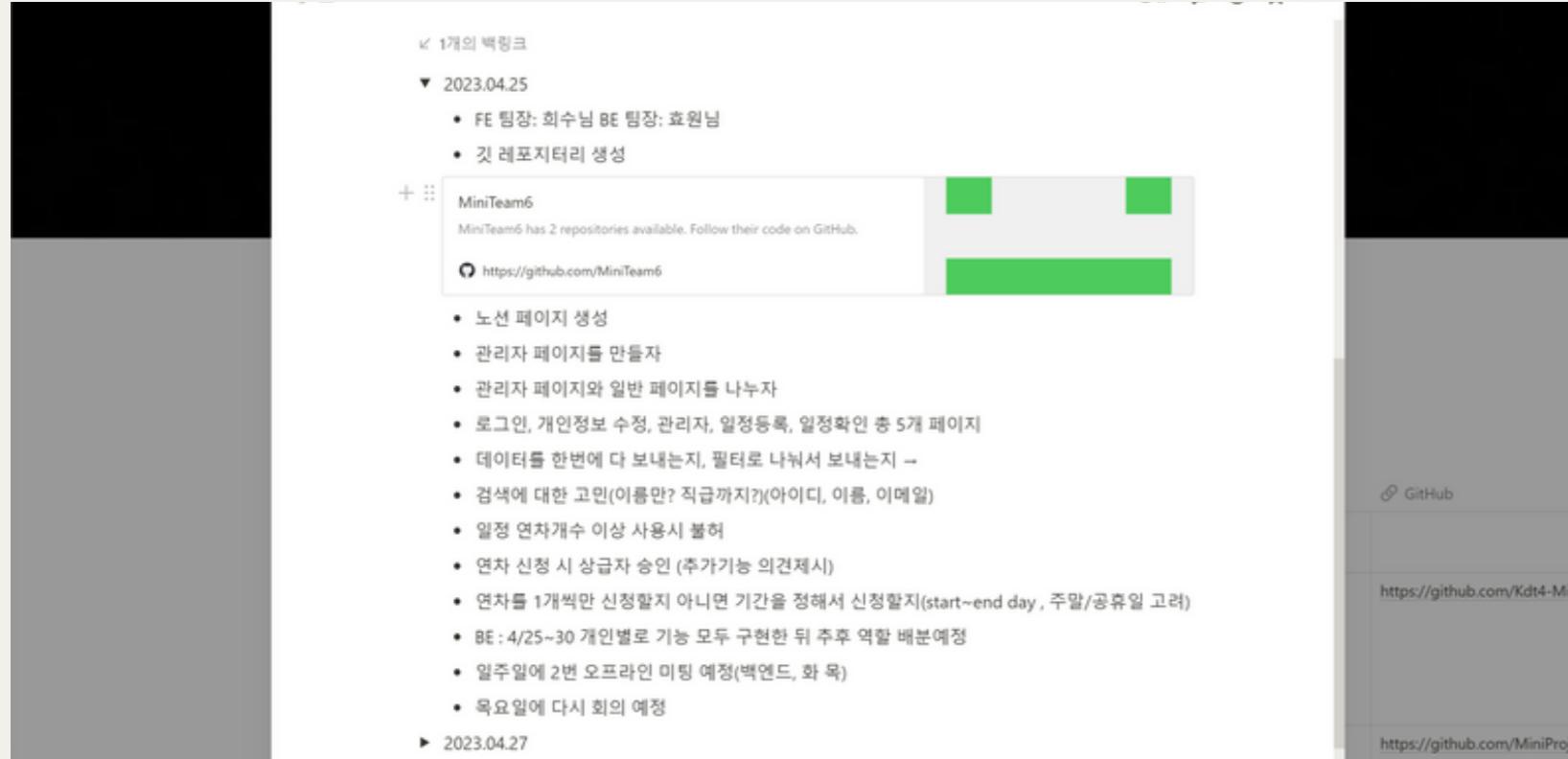
기술 스택(FE)

- React
 - 프로젝트 요구사항
- Typescript
 - 프로젝트 요구사항
- 개발 환경 세팅
 - eslint
 - prettier
- 비동기 작업
 - Axios
 - 작업 간소화 및 관리 편의성
 - react-query
 - 쿼리키로 나누어 컴포넌트간의 refetch 관리
 - 페이지네이션 구현 용이
- 인증 관리
 - react-cookie
 - 인증 토큰 관리
- 라우팅
 - react-router-dom
 - React로 SPA를 구현하기 때문에 북 마크나 히스토리 기능 필요
 - 보안 라우팅 개발 용이(authHook)
- 달력
 - react-big-calendar
 - 비교적 작은 번들링 사이즈
 - 프로젝트 요구 기능에 맞춰 커스텀화 가능
 - react-datepicker
 - 매우 작은 번들링 사이즈
 - 프로젝트 요구 기능에 필요한 최소 기능만 있음

기술 스택(FE)

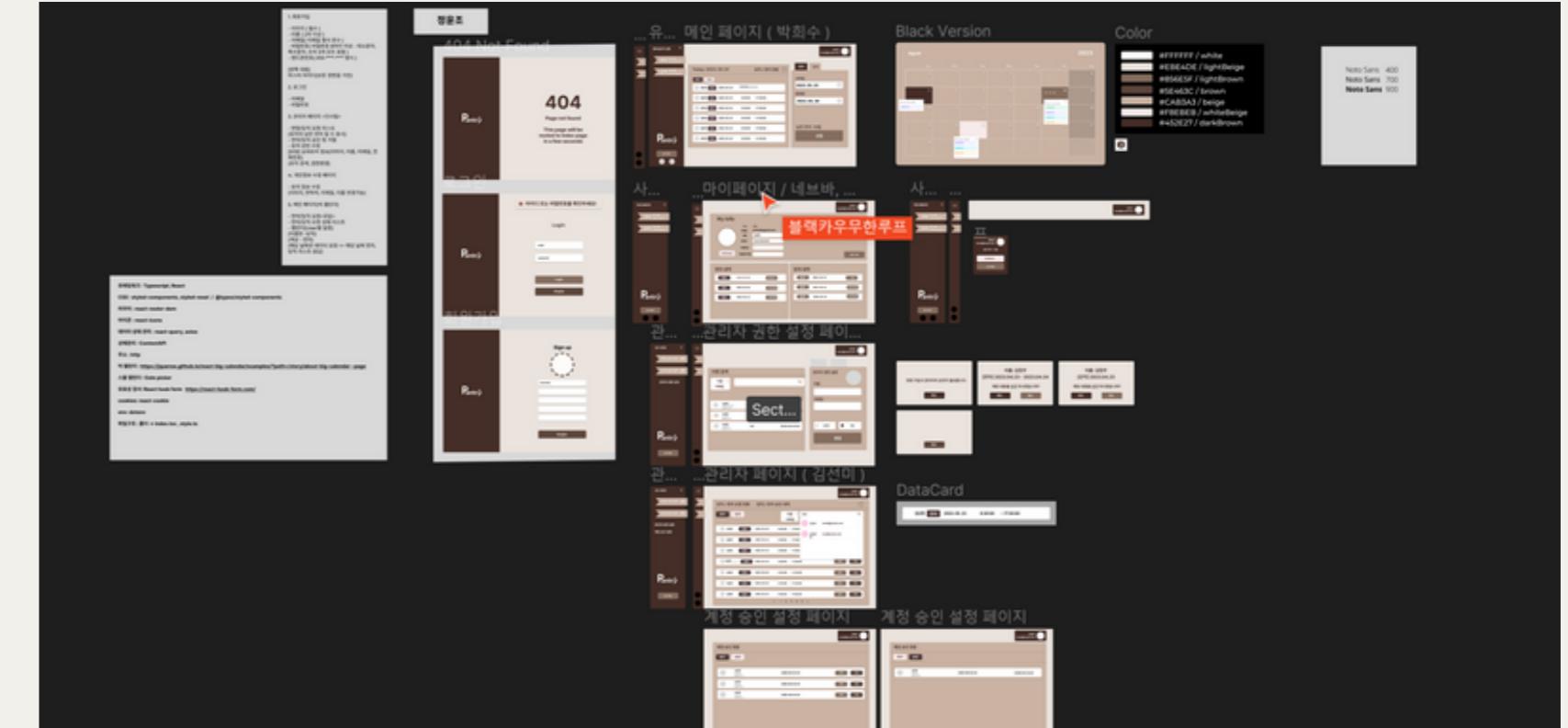
- 날짜 계산
 - moment
 - react-big-calendar의 의존성 패키지로 연결됨
 - 연차/당직 신청 시 날짜를 비교하여 필터링하는 기능이 필요
 - 기존의 new Date보다 로직이 간단
- Form
 - react-hook-form with yup
 - yup으로 schema를 설계해서 진행하면 개발 편의성
 - 입력 검사 시 동적인 UX를 구현하기 용이함
- 아이콘
 - react-icons
 - 지원하는 아이콘 개수가 많음
 - CSS 적용 가능
 - tree shaking 지원
- CSS in JS
 - styled-components
 - props와 동적으로 스타일을 조정하는데 편의성이 있음
- alert 모달
 - react-tostify
 - 동적인 alert창을 보여주고, 간단한 alert창 디자인을 지원함.

개발 과정



페이지 구성 회의

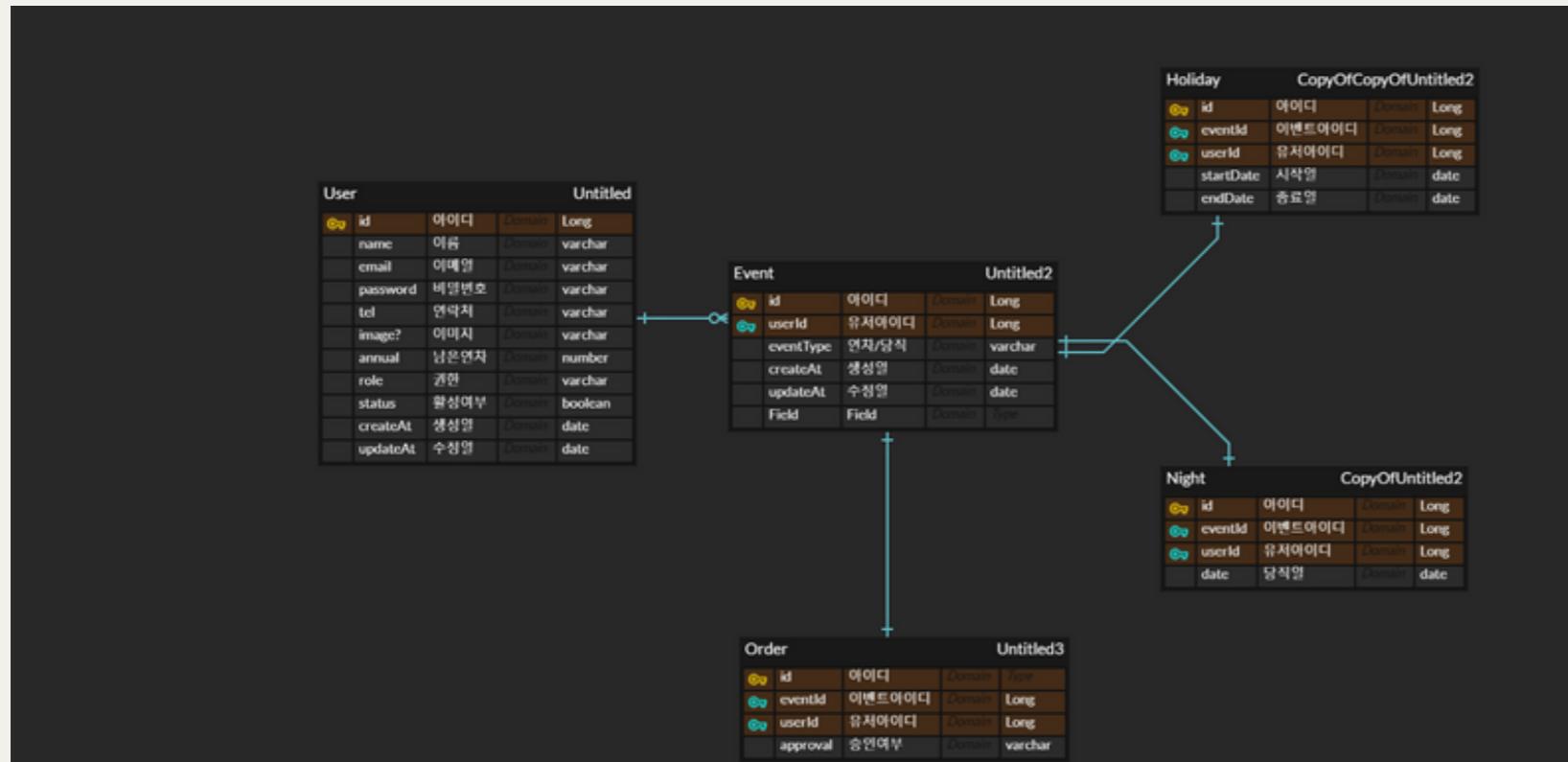
페이지의 구성과
해당 페이지에 어떤 기능이 들어갈 것인지
기본적인 틀을 잡습니다.



디자인 작업

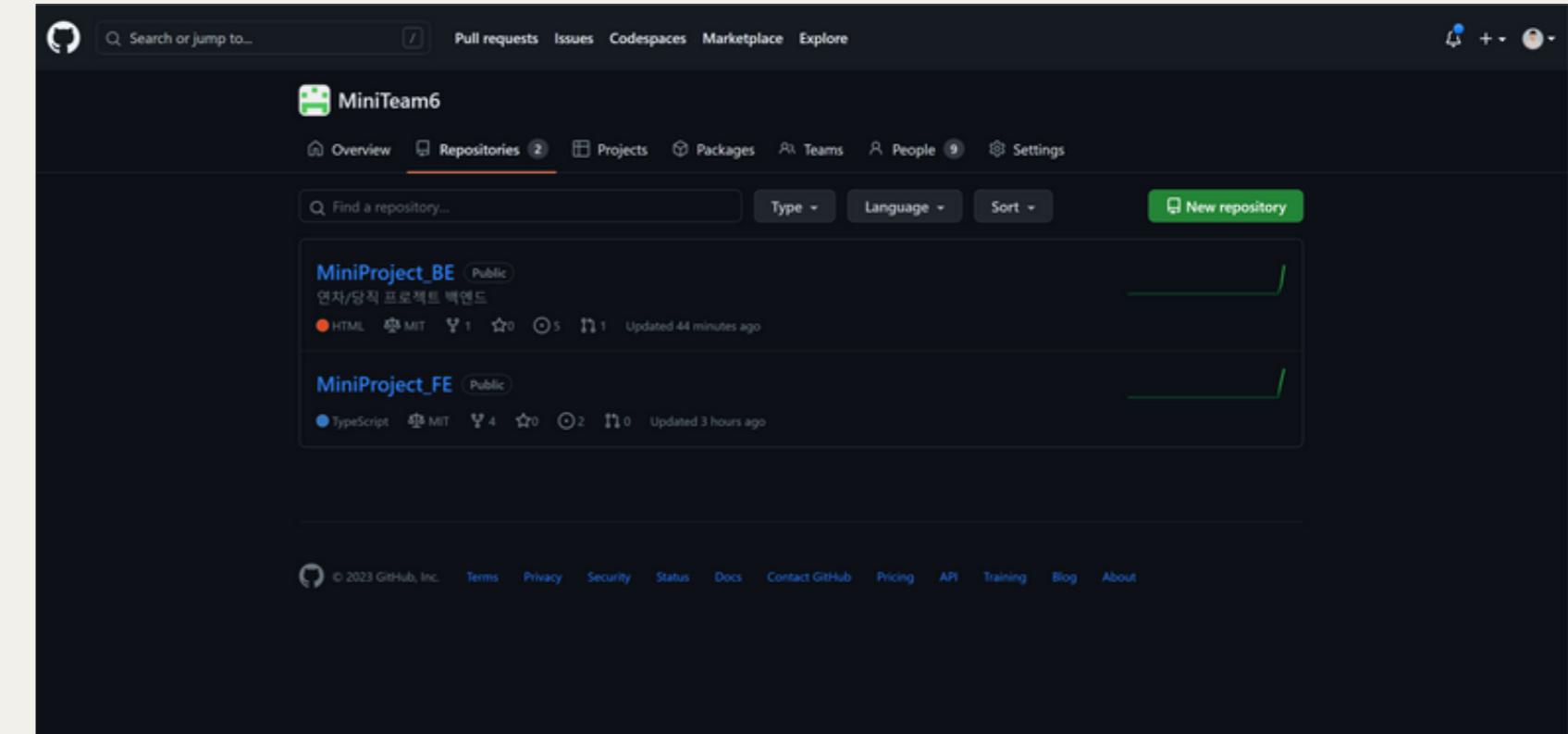
피그마를 이용하여
기본적인 와이어 프레임과
디자인을 작성합니다.

개발 과정



데이터 테이블 작성

데이터 베이스를 어떻게 연결하고
어떤 식으로 관리할지 작성합니다.



Github Repo 생성

FE, BE 간의 사용 언어, 명세서 등.
정보를 공유하고 소통한 후
팀 repo를 생성하고,
본격적인 개발을 시작합니다.

개발 과정(FE)



역할 분담 및 레이아웃 구성

페이지 단위로 역할을 배분하고,
로고 제작 및 기본적인 파일 구성과
레이아웃 구조를 잡습니다.

(컴포넌트 별로 구성하려 하였으나, 시간 관계상
페이지로 역할을 나눴습니다)



UI/UX 구현

백엔드의 API가 완성되는 동안
UI/UX 적인 디자인을 먼저 수행합니다.
백엔드의 개발 시간을 고려하여,
MSW로 가짜 데이터를 가지고
구현하기도 했습니다.

개발 과정(FE)

Frontend screenshot showing a list of vehicle/shift requests. The table includes columns for Name, Date, and Status. The status column shows '승인' (Approved) for all listed items.

Name	Date	Status
코스	2023-06-17	승인
박희수	2023-06-02	승인
코스	2023-06-01	승인
임성열	2023-05-23	승인
임성열	2023-05-22	승인
코스	2023-05-20	승인

완성된 데이터 요청

완성된 API에 데이터를 요청하고,
응답 받은 데이터를 기반으로
사용자에게 보여줍니다!
(많은 시행착오가 있었답니다.)

GitHub pull request list showing multiple PRs merged yesterday:

- #29 by Nevacat was merged yesterday • Review required
- #28 by Nevacat was merged yesterday • Review required 5 tasks done
- #27 by sung34 was merged yesterday • Review required
- #26 by seon-mikim was merged yesterday • Review required 5 tasks done
- #25 by minki-dev was merged yesterday • Review required
- #24 by sung34 was merged yesterday • Review required
- #23 by jjj1111 was merged yesterday • Review required
- #22 by sung34 was merged 3 days ago • Review required
- #21 by sung34 was merged 3 days ago • Review required
- #20 by sung34 was closed 3 days ago • Review required
- #19 by seon-mikim was merged 4 days ago • Review required 5 tasks done

본격적인 병합

이제 각자 맡았던 페이지를 병합하고,
공통으로 사용할 수 있는 컴포넌트는
공통 컴포넌트로 빼서 코드를 수정합니다.
그리고 전체적인 구조를 수정합니다.
(초반 레이아웃을 제대로 잡지 않아서
많은 문제가 발생했습니다.)

개발 과정(FE)

The screenshot shows the Pantry FE application interface. On the left, there's a sidebar with a logo and navigation links for '로그아웃' and '로그인'. The main area has a header '연차/당직 신청' with a back arrow. Below it, a sidebar shows '연차 D-1 2023-05-17' and '당직 D-4 2023-05-20'. The main content area displays a list of leave requests for today (2023-05-16) under 'TODAY: 2023.05.16'. Each request includes the user's name, email, leave type ('연차'), duration ('4일'), and dates ('2023-05-26 (금) ~ 2023-05-29 (월)'). To the right, a modal window titled '당직 / 연차 현황' shows a calendar for May 2023 with the 16th highlighted. It also includes fields for '신청일' (2023/05/16), '종료일' (2023/05/16), and '총: 1 일'. A large '신청' button is at the bottom of the modal.

프로젝트 배포

QA를 거쳐서 오류와 레이아웃 수정 후
어느정도 완성 된 프로젝트를
AWS를 통해 배포 합니다.

The screenshot shows the Pantry FE application login page. It features a dark brown header with the 'Pantry' logo. Below it, there are two input fields: one for 'Email' containing 'test@naver.com' and another for 'Password' with placeholder '*****'. At the bottom, there are two buttons: 'Login' in white and 'SignUp' in light blue.

에러 핸들링

기간 내내 많은 시행착오와 시간에 쫓기며,
마무리 짓지 못했던 에러에 대응하는
코드를 추가했습니다.
그 이후 코드를 다시 검토합니다.

FE 트러블 슈팅



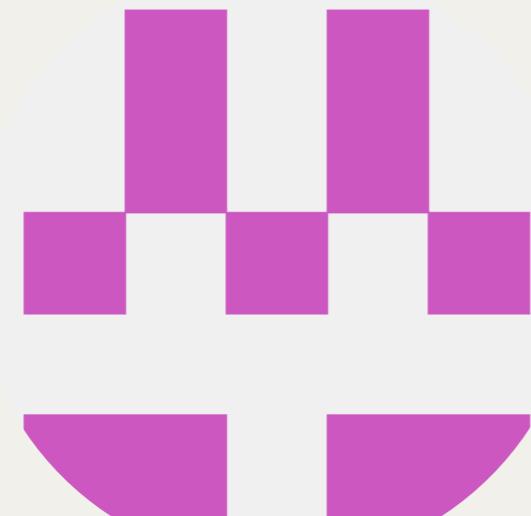
문제

1. 부족한 레이아웃 구성으로 인해 병합 시 구조 붕괴
2. AWS 연동 github 자동 배포 관련 이슈
3. 메뉴를 닫았음에도 계속 화면 구성에 포함되어 클릭이 되는 이슈(opacity 속성)
4. 회원가입시 signupDTO를 application/json 으로만 하면 오류
5. props drilling 해결을 위해 contextAPI로 useState Hook을 전달하였으나, 잣은 리렌더링 발생
6. 회원가입시 관리자에게 회원가입 요청은 갔으나 axioserror가 떠서 회원가입 페이지에서 실패가 뜨는 오류
7. moment의 기본값을 설정해주지 않아서 요청 데이터에 현재 시간보다 12시간 늦은 시간이 입력됨

해결

1. 제대로 된 레이아웃 구조 수정 후 전체 페이지 조정
2. 미해결
3. pointer-event를 none으로 설정하여 클릭이 무시되도록 함(트랜지션 효과 유지 가능)
4. signupDTO를 application/json한 것을 blob객체로 만들어서 전송해서 성공.
5. useReducer Hook으로 stateContext와 dispatchContext로 나누어 공급, 필요한 부분에 memo 도입
6. axios.timeout의 한도를 늘려서 해결하려 했으나 미해결.
7. 코드 가장 상단에 moment.tz.setDefault('Asia/Seoul')선언

Back Engineer



신효원(조장)

User 관련 기능 구현
배포, 스토리지 등 인프라 관리



이지훈

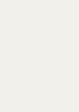
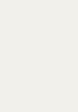
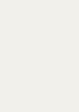
프로젝트 설계 및 확장
CI/CD, 로깅 등 인프라 구축
테스트 기반 API 문서화



고예림

Admin 관련 기능 구현
Admin 테스트코드 작성
에러/예외 처리

기술 스택(BE)

- 웹 서버 (프로젝트 요구사항)
 - Spring boot 2.7.11 
 - JDK 11 
 - Gradle 7.6.1 
- 환경 세팅
 - Docker 
- 데이터 베이스
 - MySQL 5.7 
 - H2(테스트) 
- 이미지 처리
 - AWS S3 
 - Thumbnailator 
- 인증 관리
 - JWT 
- 배포
 - AWS EC2 
- CI/CD
 - Github Actions 
- 테스트
 - JUnit 
 - Mockito 
- 문서화
 - Asciidoctor 
 - Rest Docs 
- 로그 모니터링
 - Sentry 

개발 과정



요구사항 정리

FE팀원들과의 싱크업 미팅을 통해
프로젝트 요구사항과 FE/BE 간의 요구사항을
토대로 개발 기능의 범위를 결정하였습니다.



컨벤션 협의

프로젝트에서 필요한 컨벤션들을 협의하였습니다.
특히 GitHub에서 PR요청을 하고
팀원 1명 이상 Review를 받아야
Merge할 수 있도록 설정하였습니다.

개발 과정



도메인 설계

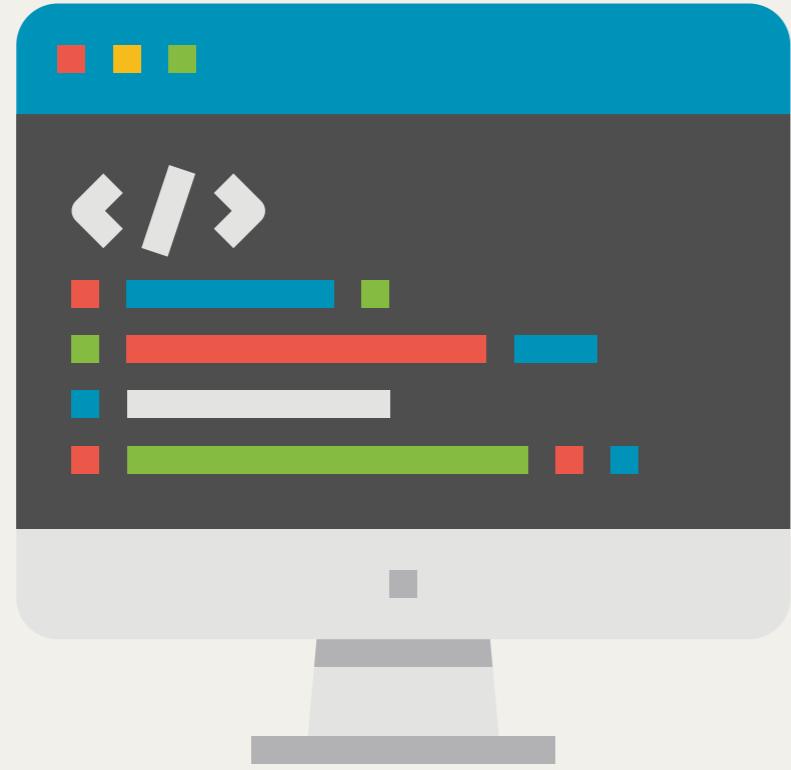
ERD를 바탕으로 도메인 설계를 진행하였습니다.

Aa Name	Text	Tags	비고	Request	Res
/api/signup	회원가입	POST	<p>status: 기본값 False(관리자 승인 때문에)</p> <ul style="list-style-type: none">- 이미지 (필수)- 이름 (2자 이상)- 이메일 (이메일 형식 준수)- 비밀번호 (8자리 이상, 대소문자/특수문자/숫자 3개 모두 포함)- 핸드폰번호 (010-****-**** 형식)	{ "image": <이미지파일첨부>, "signupInDTO": { "username": "김김김", "password": "aaaa1234@@@", "email": "email@email.com", "phone": "010-1111-1111" } }	{ "status": "ok", "msg": "user", "id": "user", "email": "email", "phonenumber": "phonenumber", "image": "image", "thumb": "thumb" }
/api/emailcheck	이메일 중복 확인	GET	이메일 존재하면 트루		{ "exists": true }

1차 API 명세서 작성 / 역할분담

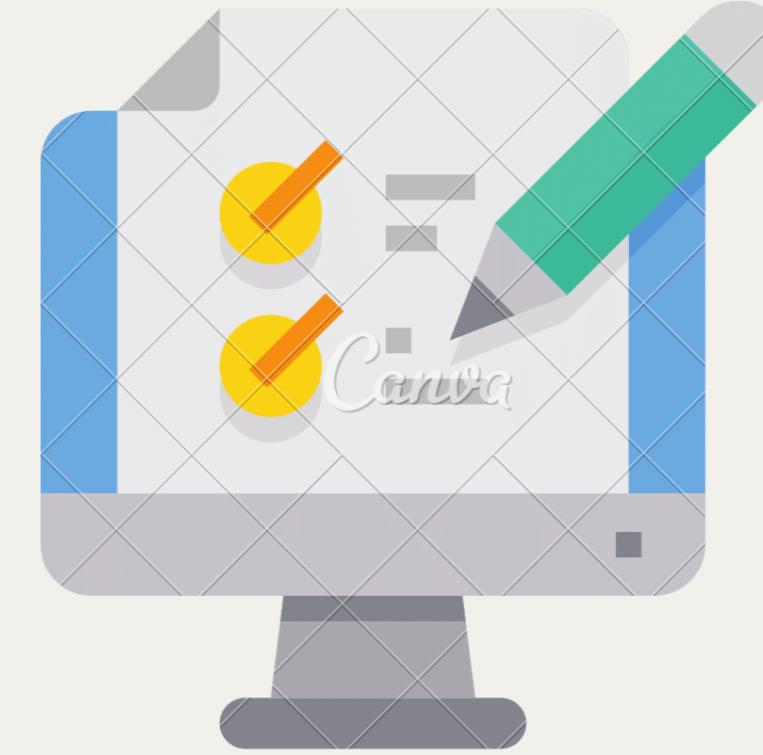
FE의 와이어프레임을 바탕으로
1차 API명세서를 작성하고, 역할분담을 하였습니다.

개발 과정



기능구현

요구사항과 API명세서를 토대로
기능을 구현하였습니다.



테스트코드 작성 및 최종 API 명세 문서화

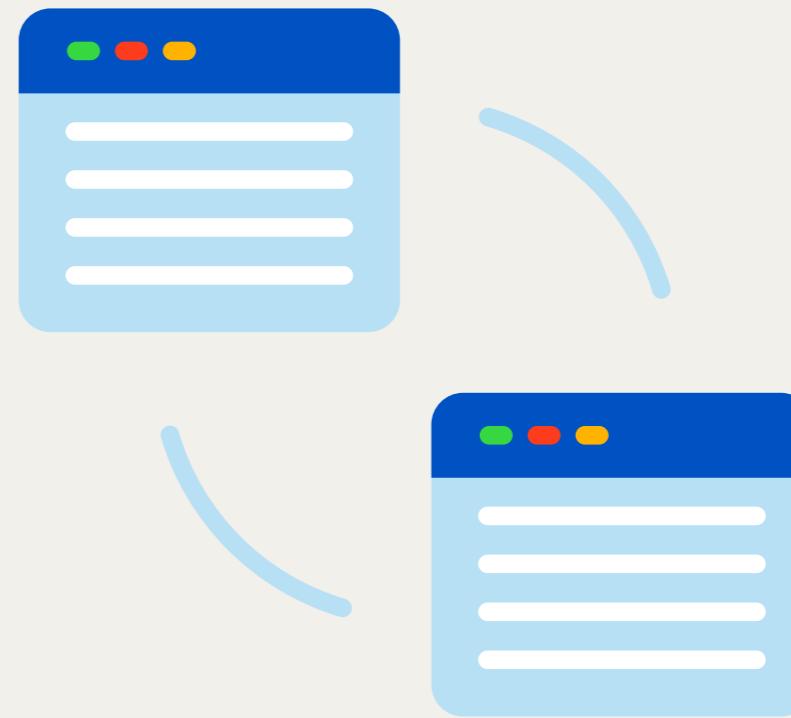
구현한 기능들이 정상적으로 작동을 하는지
테스트하기 위해 단위테스트 및 통합테스트를 진행하였고
이 과정에서 오류가 발생한 부분은 수정하였으며
REST Docs로 API명세서를 문서화하였습니다.

개발 과정



인프라 구축

효율적인 개발 환경과
변경에 유연한 소프트웨어를 만들기 위해
인프라 구축을 하였습니다.



리팩토링

FE와 통합 후 FE의 요구사항 반영 및
부족한 기능들을 보완, 예외/에러 처리 등
리팩토링 작업을 진행하였습니다.

컨벤션 협의

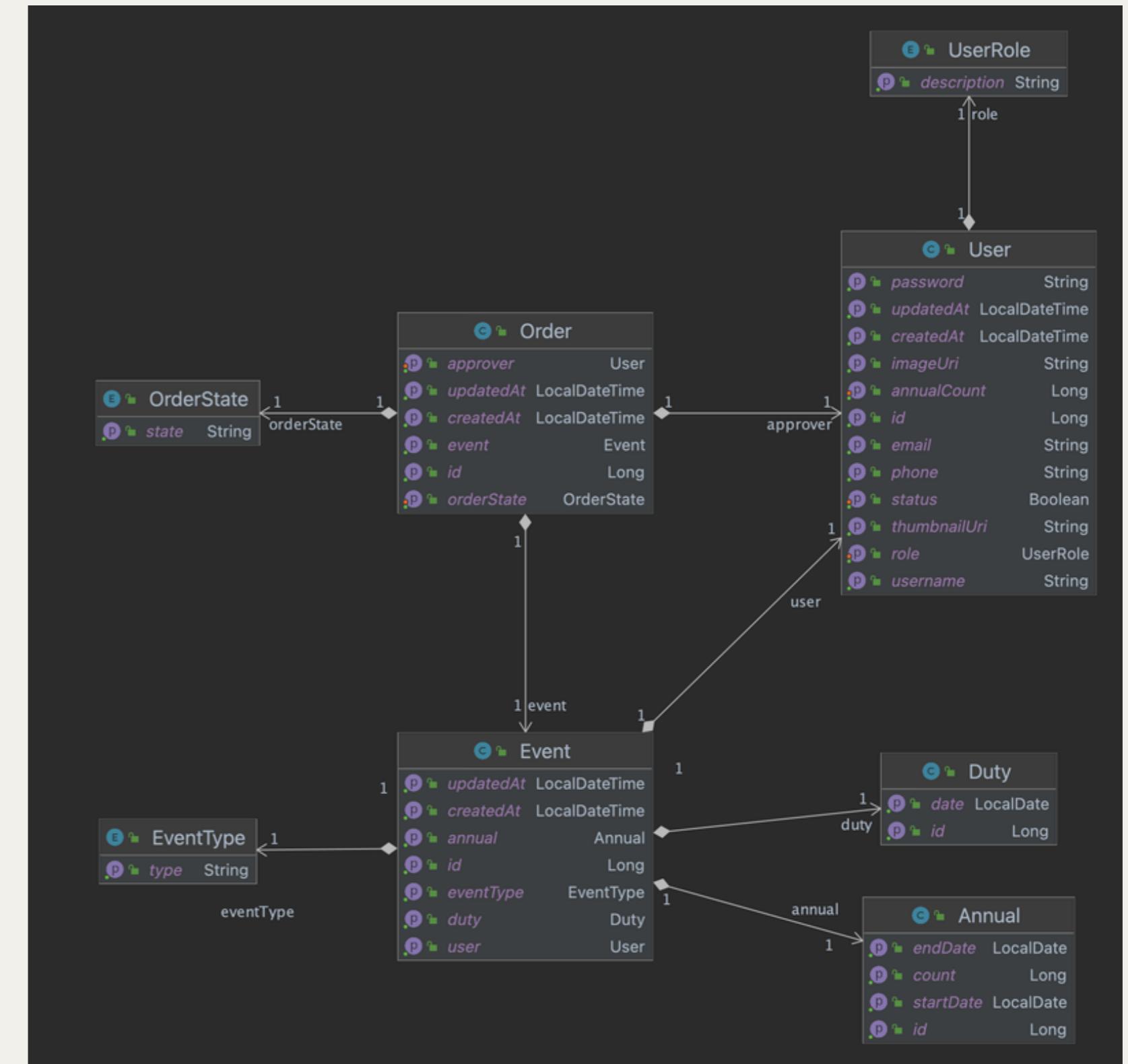
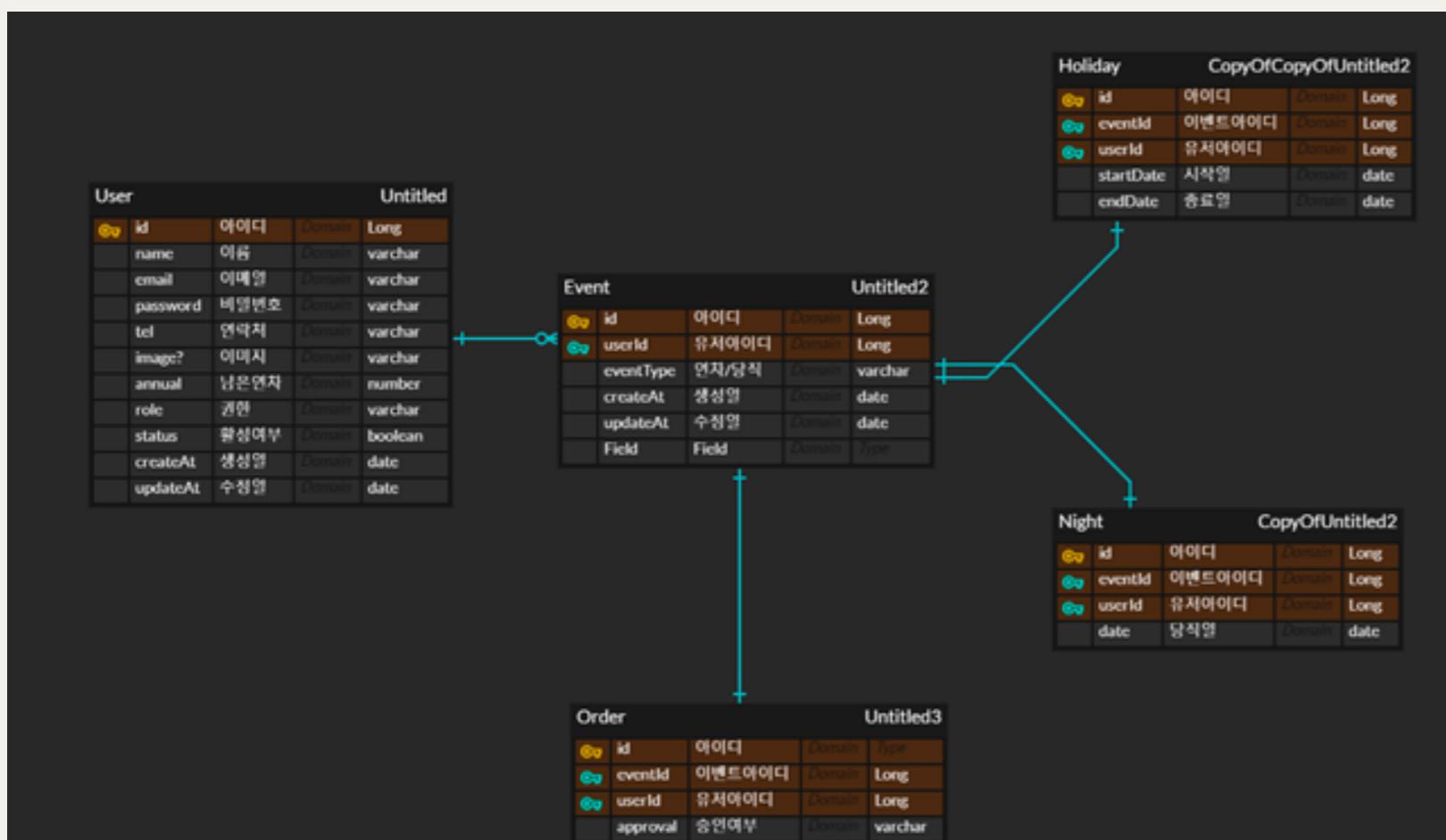
코드 컨벤션

- Java
 - 구글 자바 스타일 가이드 참고
- DB
 - 스네이크 케이스
- 프로젝트 구조
 - MVC 패턴

Git 브랜치 전략과 룰

- 3인이라 포크는 하지 않았고, 커밋/이슈/PR 컨벤션 등은 조금 자유롭게 진행
- Main, Dev, Feature 로 나누어 진행
- Dev 를 최신화하고 Feature 브랜치 만 들어 작업
- Dev로 PR, 테스트 패스 및 리뷰 후 머지
- 배포 시점마다 Main 으로 머지
- Dev는 1인 이상 승인해야 머지 가능

도메인 설계



User

- id
- username
- email
- password
- phone
- imgSrc
- status
 - 회원상태관리
- thumbnailSrc
- role
 - USER
 - ADMIN
- createdAt
- updatedAt

Event

- id
- eventType
 - DUTY
 - ANNUAL
- annual
 - One to One
- duty
 - One to One
- user
 - One to One
 - 신청자

Annual

- id
- startDate
- endDate

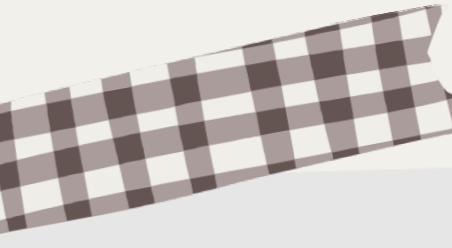
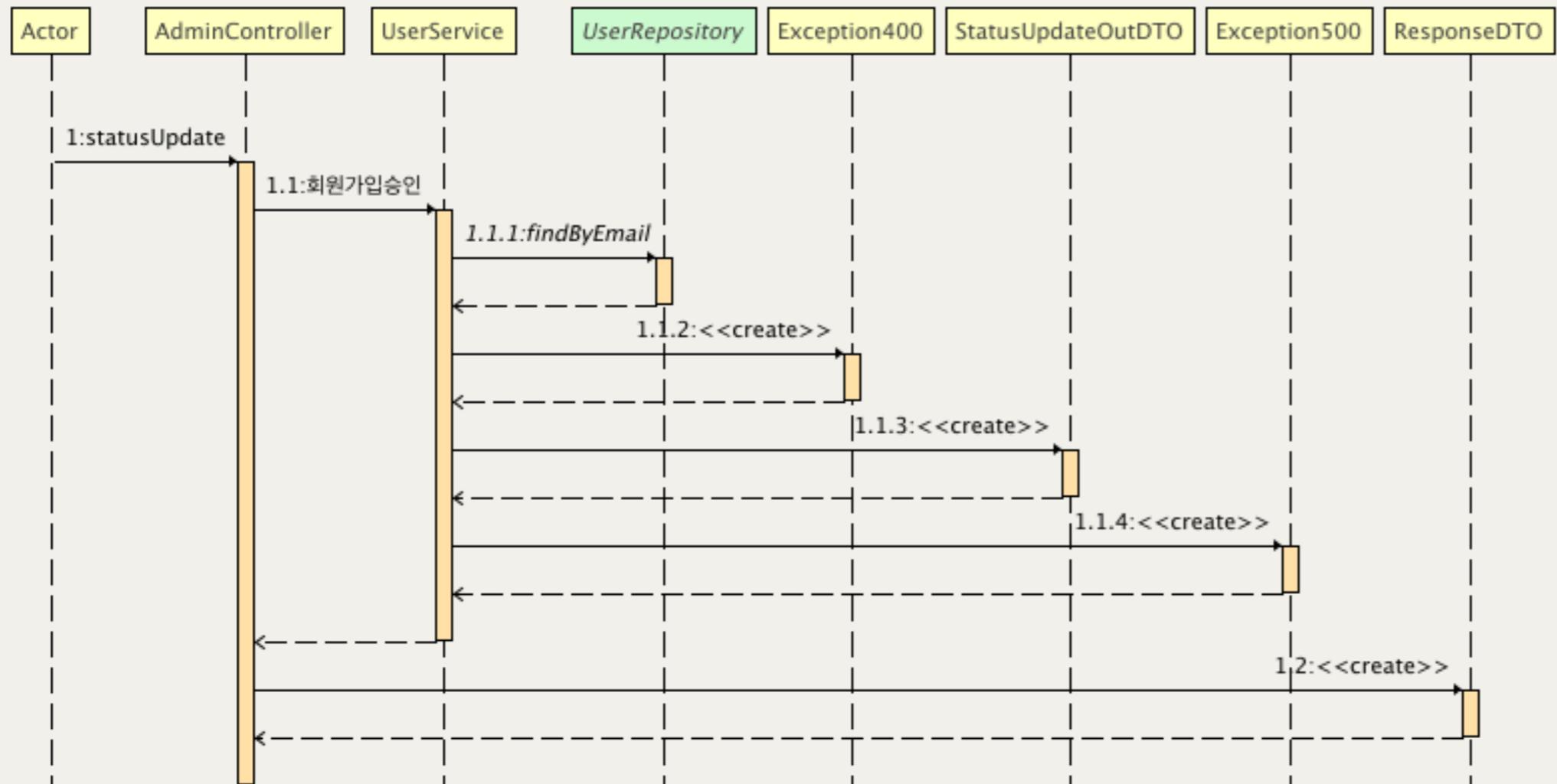
Duty

- id
- date

Order

- id
- approver
 - 승인자
 - One to One
- event
 - One to One
- orderState
 - APPROVALED
 - REJECTED
- createdAt
- updatedAt

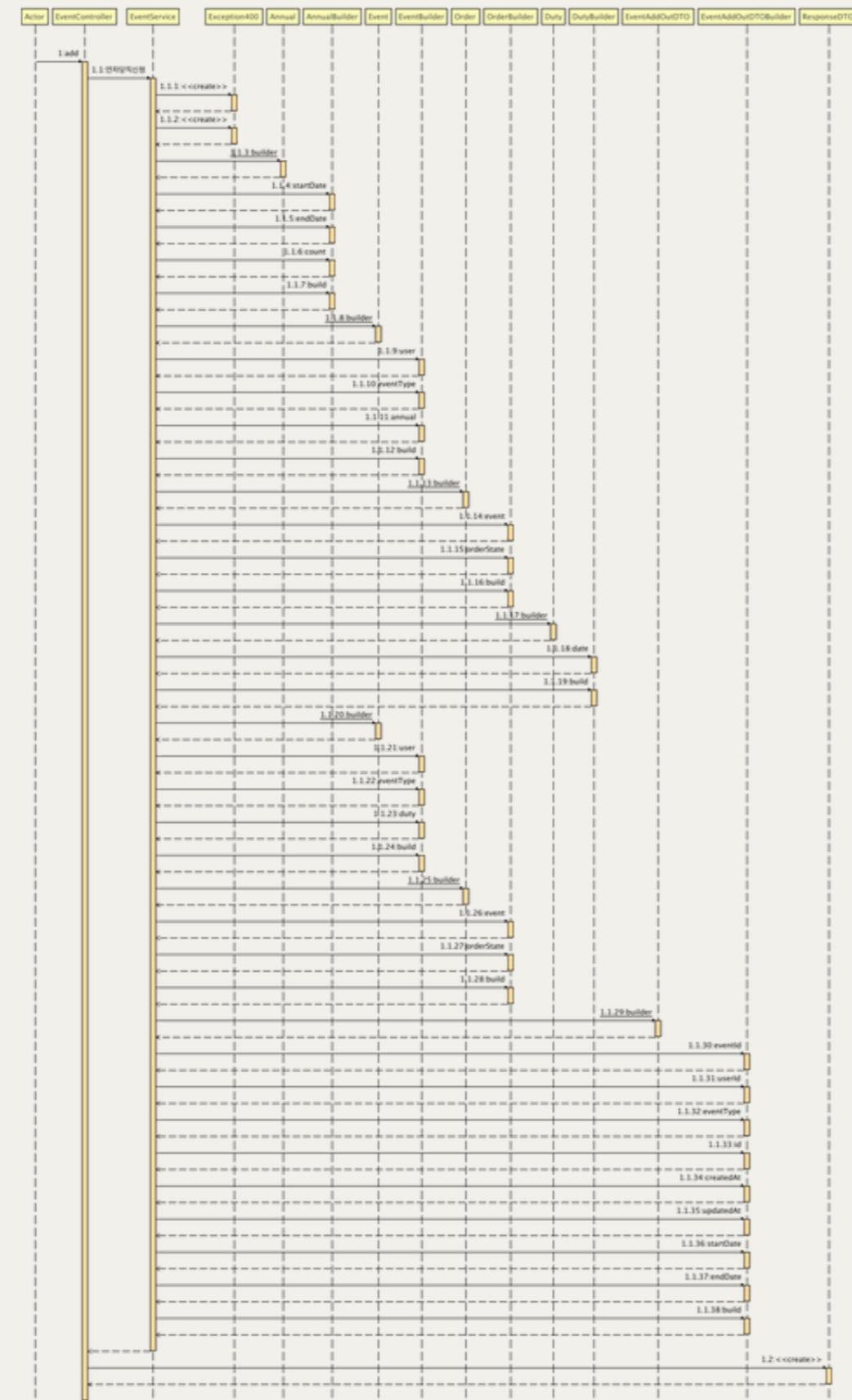
주요 기능



가입승인

- 회사에서 사용할 시스템인것을 감안해 회원가입을 한 User의 status는 모두 false
- Admin user의 승인을 받아야 status -> true로 변경
- status가 true인 User만 로그인 가능

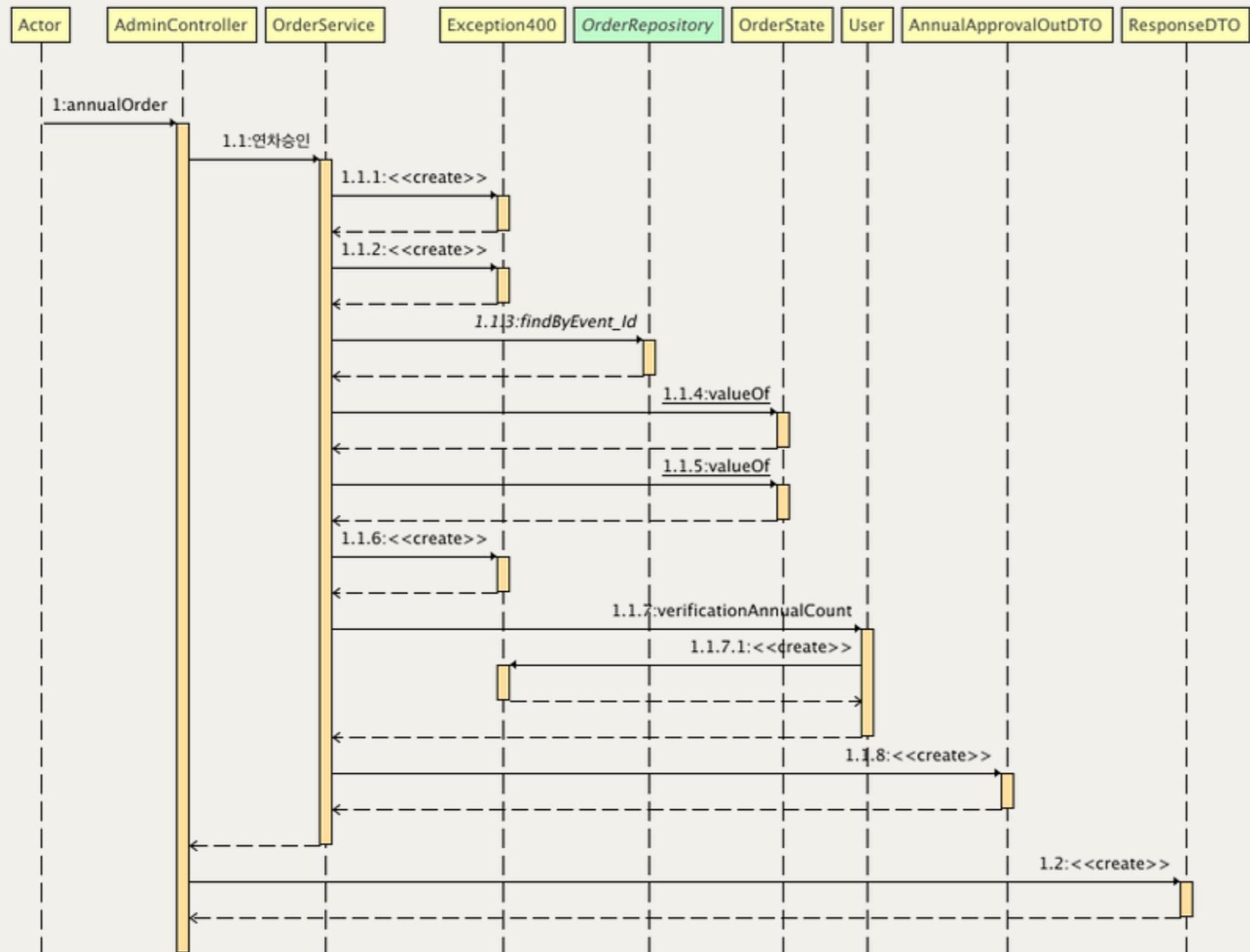
주요 기능



연차/당직 신청

- Event테이블에는 신청자인 User와 EventType에 따라 Anuual OR Duty와 단방향 OneToOne 연관관계를 맺는다.
- 연차/당직 신청시 Order에는 OrderState는 Waiting으로 , Event와 단방향 OneToOne 연관관계를 맺는다.

주요 기능



연차/당직 결재

- ADMIN만 연차/당직 결재가능
- OrderState가 Waiting인 연차/당직 신청내역들을 대상으로 결재를 할 수 있다.
- 결재상태는 OrderState의 APPROVALED(승인) / REJECTED (거절)로 구분한다.
- 결재시 Order의 User에는 승인자의 정보가 들어간다.

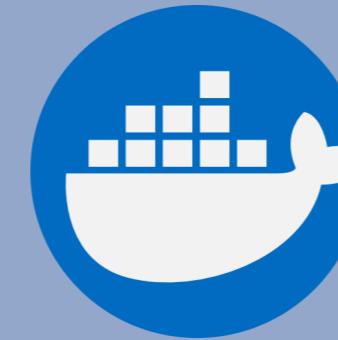
인프라 구축

Sentry



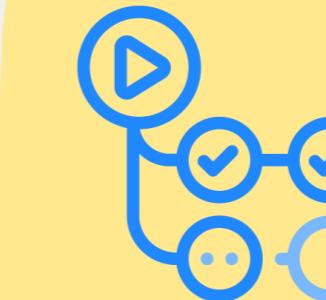
- 로그 관리
- WebHook 설정 통해 슬랙으로 알림이 오도록 설정

Docker



- 컨테이너 기반 애플리케이션 실행 환경 관리
- 호스트 환경에 구애받지 않고 독립적인 환경에서의 실행 가능

GitHub Actions



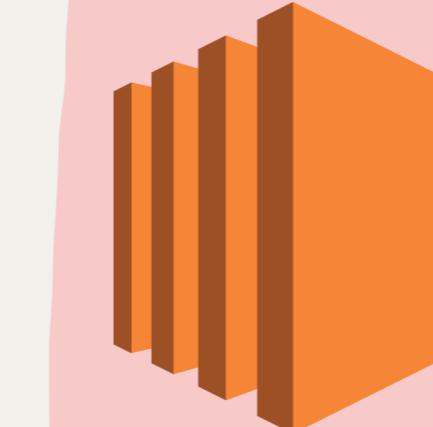
- CI/CD
- Main 브랜치에 Merge 하거나 PR 요청 시 액션 실행

S3



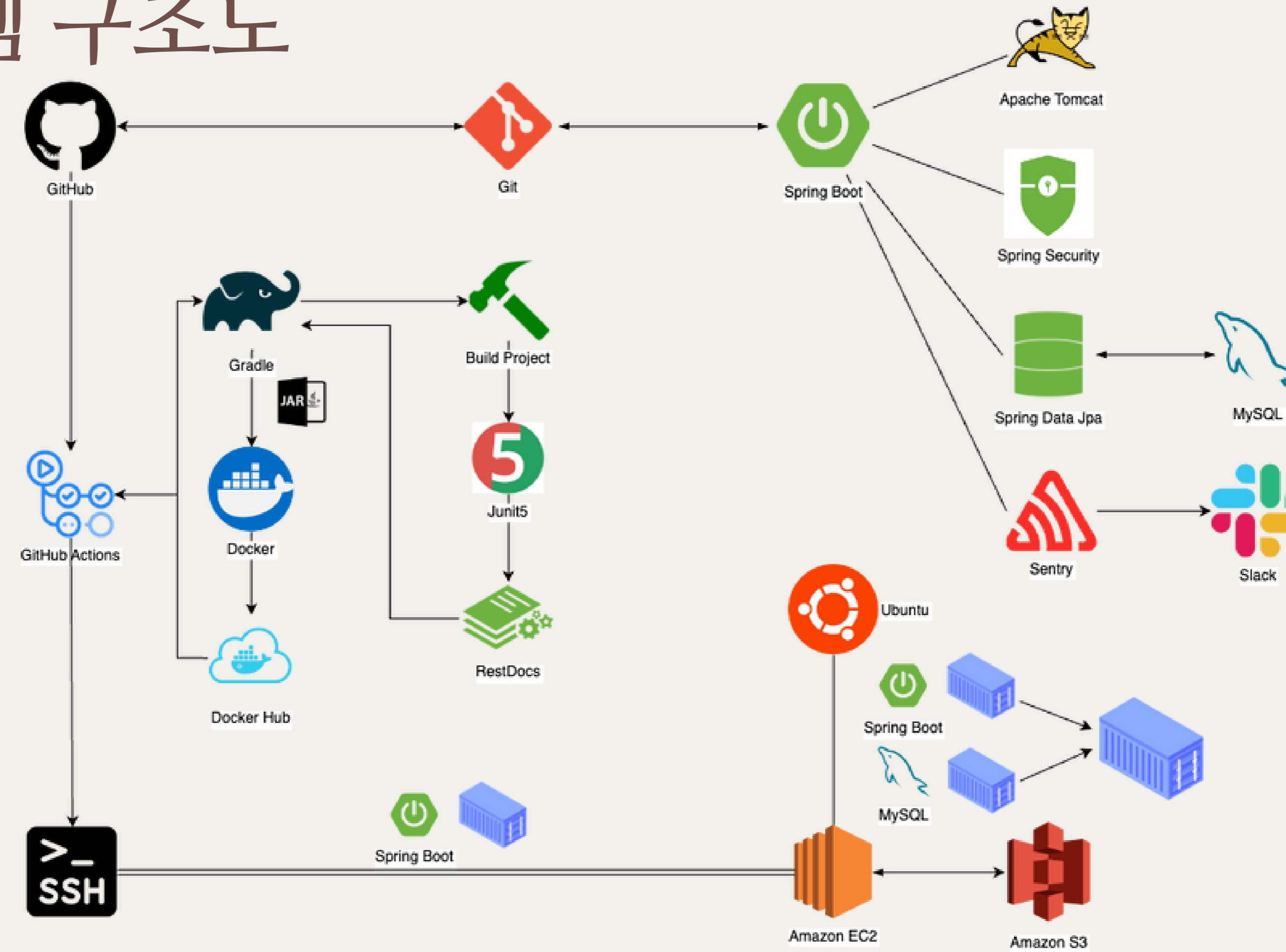
- 클라우드 스토리지 서비스
- 회원 사진 파일 저장/관리

EC2



- 클라우드 컴퓨팅 서비스
- DB와 WAS를 구동하기 위한 가상 환경 및 리소스 제공

시스템 구조도



M4 mine ⓘ

- [Issues](#)
- [Projects](#)
- [Performance](#)
- [Profiling](#)
- [Replays](#)
- [Crons](#)
- [Alerts](#)
- [Discover](#)
- [Dashboards](#)
- [Releases](#)
- [User Feedback](#)
- [Stats](#)
- [Settings](#)

Quick Start
5 Remaining tasks

- [My Sentry Trial](#)
- [Help](#)
- [What's new](#)

< Collapse

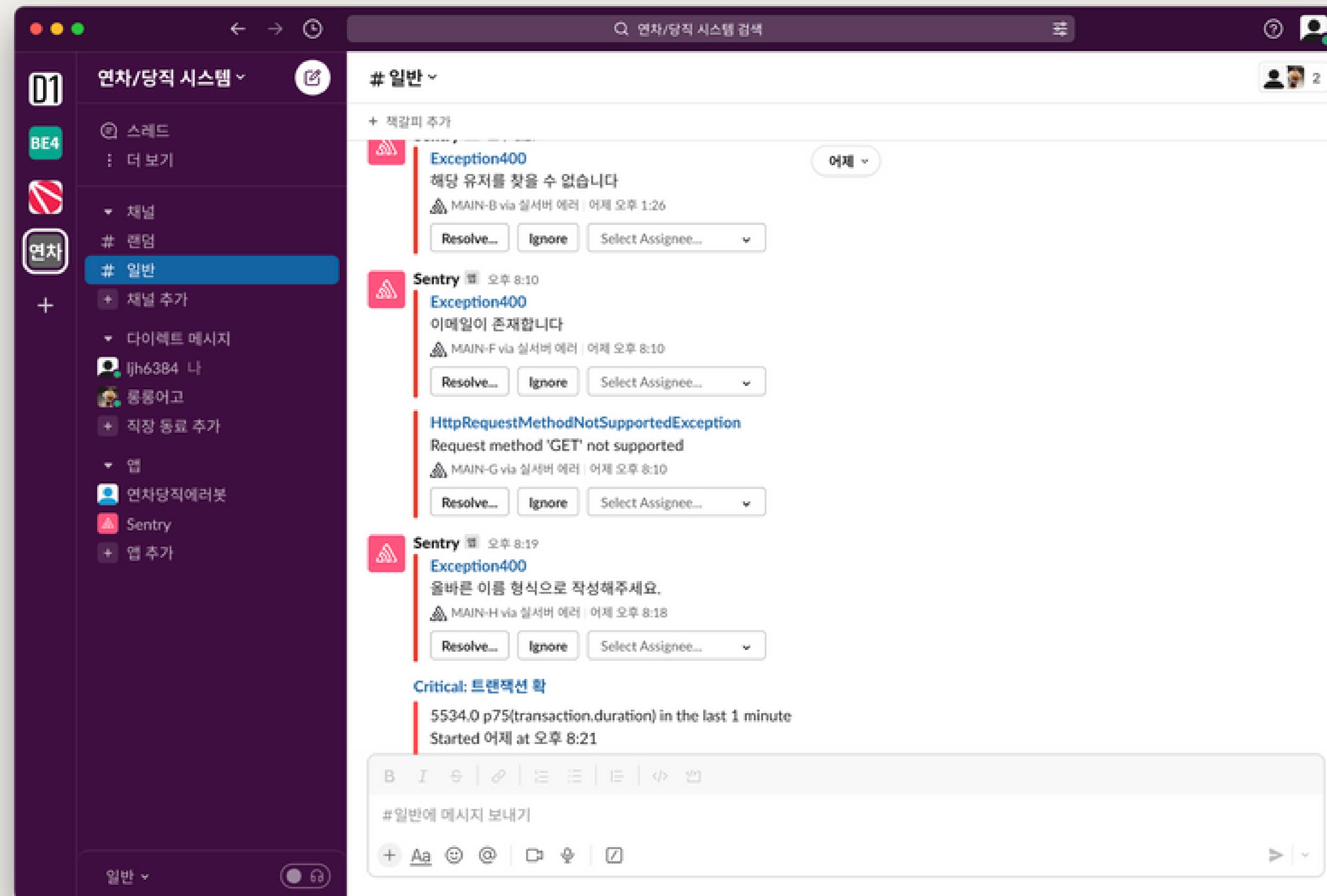
Issues ⓘ

All Unresolved 11 For Review 11 Ignored

My Projects production 14d All Categories Custom Search is:unresolved

Resolve Ignore Mark Reviewed Merge ... Last Seen GRAPH 24h 14d EVENTS USERS ASSIGNEE

Issue Type	Description	Severity	Last Seen	Graph	Events	Users	Assignee
Exception400	POST /api/signup 이메일이 존재합니다	Medium	10min ago 15hr old		9	0	
Exception401	POST /api/login 인증되지 않았습니다	Medium	9hr ago 1d old		35	0	
Exception400	GET /api/user/myinfo 해당 유저를 찾을 수 없습니다	Medium	9hr ago 22hr old		6	0	
Exception400	GET /api/user/nextevent 해당 유저를 찾을 수 없습니다	Medium	9hr ago 9hr old		1	0	
Exception404	POST /api/admin/role/update 동일한 권한으로 변경할 수 없습니다.	Medium	15hr ago 15hr old		2	0	
토큰 만료됨	POST /api/signup shop.mtcoding.restdemo.core.auth.jwt.MyJwtAuthorizationFilter	Medium	15hr ago 15hr old		2	0	
HttpMediaTypeNotSupportedException	POST /api/signup Content type 'application/octet-stream' not supported	Medium	15hr ago 15hr old		1	0	
MultipartException	POST /api/signup Failed to parse multipart servlet request; nested exception is java.io.IOException: org.apache...	Medium	15hr ago 15hr old		1	0	



BE 트러블 슈팅

문제

1. 키 관리 문제
2. 백엔드 개발이 속도가 늦어서 프론트엔드가 기다려야 하는 문제
3. 새로운 기술 사용하는데 어려운 문제
4. Multipart-formdata 테스트 문제
5. EC2 기본 사용자 문제
6. AOP 적용안되는 문제
7. 가장 빠른 연차 테스트 매번 달라지는 문제
8. S3 서버와 EC2 인스턴스 느린 문제
9. 깃허브액션에서 도커 다운로드 중 에러 발생하는 문제
10. 모든 응답을 일관성있게 노력했지만, 예측 불가능한 응답이 발생

해결

1. 파일에서 파일로 전달하거나 이미지 빌드시 깃허브 시크릿, ARG, ENV 등을 이용해 키 전달
2. 임시 API를 만들어서 전달해서 해결
3. 열심히 공부해서 해결
4. 직접 document를 작성해서 해결
5. 아마존 리눅스 사용시엔 EC2-User, 우분투를 사용했기 때문에 ubuntu인걸 파악해서 해결
6. @Valid 사용시 바로 뒤에 Errors를 붙여야 한다는걸 공유해서 해결
7. 테스트에서만 따로 시간을 관리해서 항상 테스트가 통과되도록 설정
8. 프리티어 요금제여서 그런걸로 추정. 프론트에서 대기시간을 늘려서 임시해결
9. 실패시 retry 설정하려고 했으나 기능을 지원안해서 서버가 돌아올때까지 재시도
10. 미해결

프로젝트 결과

TEAM
PANTRY

배포 사이트
Pantry.

깃허브
FRONT
BACK



THANK YOU

감사합니다

