



대보정보통신

최종 프로젝트

- TIKITAKA -

팀 - 불4조

이재진 황승준 서지효 정준열

목차

프로젝트 소개

분석 / 설계

시연

팀 소개 및 소감



프로젝트 소개

- 프로젝트 주제 선정
- 프로젝트 기획
- 벤치마킹

협업을 위한 통합 웹사이트 **TIKITAKA**

01

소셜 로그인을 통한 간편 로그인
어렵게 가입하지 않고 편하게 가입하자

02

한번에 관리하는 프로젝트
프로젝트를 생성하고 모든 프로젝트를 관리

03

한번에 하는 워크스페이스와 채팅
워크스페이스와 채팅의 결합

프로젝트 기획

협회에서 교육을 들으며 3번의 프로젝트를 진행했습니다.

팀이 바뀌고 프로젝트를 진행할 때마다 매번 노션 워크스페이스를 새로 만들어 초대를 하고 카카오톡 팀 방을 만들어 모든 팀원들을 초대하고 노션을 썼다가 카카오톡으로 소통했다가 하는 어려움이 있었습니다. 이에 모든 기능을 가지고 있는 웹사이트가 있으면 좋겠다는 생각을 하여 해당 프로젝트를 기획하게 되었으며 화상채팅 및 화면 공유가 되면 더욱더 사이트를 고도화 할 수 있을 것이라고 생각하여 해당 기능을 추가하였습니다.





분석 / 설계

- 린캔버스 및 요구사항
- 아키텍쳐
- DB모델링 / 다이어그램
- 스토리보드
- 형상관리

린캔버스 및 요구사항

팀 명 : 블4조 (열정을 불태우며 아침까지 코딩하는 4조)

프로젝트 명 : 티키타카(Tiki-Taka)

| 문제 | 솔루션 | 고유가치 제안(Needs) | 기술적경쟁 우위 | 고객군 |
|---|---|--|---|---|
| <p>문제</p> <ol style="list-style-type: none">프로젝트를 진행할 때마다 노션, 슬랙, 카카오톡 등 여러 어플 및 사이트에 가입을 해야함.프로젝트 협업을 하나의 사이트에서 진행 할 수 없고 여러 사이트의 기능을 같이 사용해야 함.여러 사이트의 기능을 사용하다 보니 자료 공유 및 확인에 불편함이 있음 <p>기존 대안</p> <ol style="list-style-type: none">문서공유<ul style="list-style-type: none">- 카카오톡, 슬랙등 메신저를 통한 문서의 공유- 노션을 통한 문서 작성채팅<ul style="list-style-type: none">- 카카오톡, 슬랙 등 메신저 사용- 노션에 글을 남겨두고 확인- 디스코드 음성, 화상 채팅칸반보드<ul style="list-style-type: none">- 노션을 통한 칸반보드 활용 | <p>솔루션</p> <p>기존에 프로젝트를 진행할 시 필요한 기능은 여러가지이지만 하나의 어플, 사이트에서 모든 기능을 제공하는 곳이 없음.</p> <p>협업에 필요한 정보 공유, 문서 공유, 칸반 보드, 채팅 기능이 한 곳에 있는 사이트가 필요함.</p> <p>프로젝트 별로 초대코드를 만들어 인원을 관리 할 수 있게함</p> <p>프로젝트 별 포함된 인원들끼리 어느 페이지에서도 실시간 채팅이 가능한 기능을 제공</p> <p>각 프로젝트 기능 및 역할마다 담당자를 선택하여 칸반보드를 생성, 확인, 변경할 수 있는 기능 제공</p> <p>각 인원이 원하는 정보, 문서를 공유, 작성할 수 있는 기능 제공</p> <p>프로젝트에 들어온 사람들이 함께 얘기할 수 있는 음성 기능과 회의할 수 있는 화상채팅 기능 제공</p> | <p>고유가치 제안(Needs)</p> <p>한 곳에서 모든 프로젝트를 관리하고 협업할 수 있는 협업 관리 사이트</p> <p>“이거 만들고 이거 가입하고 이쪽 저쪽 많이 힘들었지? TIKITAKA가 답이다!! 여러 프로젝트?! 공유!! 채팅!! 칸반보드!! 모두 한곳에서!!!”</p> <p>상위 개념</p> <p>노션(Notion)</p> <ul style="list-style-type: none">- 개인 및 팀 워크스페이스 <p>카카오톡</p> <ul style="list-style-type: none">- 실시간 채팅 및 정보 공유 <p>슬랙</p> <ul style="list-style-type: none">- 실시간 채팅 및 정보 공유- 협업을 위한 팀 문서 관리 <p>디스코드</p> <ul style="list-style-type: none">- 음성 및 화상채팅 기능 | <p>기술적경쟁 우위</p> <p>노션, 슬랙, 카카오톡 등 여러 어플 및 사이트를 사용하지 않고 한 사이트에서 문서 공유 및 채팅, 회의록 등 작성을 할 수 있음</p> <p>프로젝트 인원들만 사용할 수 있는 독립적인 공간으로 활용 가능</p> <p>개인 프로젝트 관리에도 유용한 사이트</p> <p>프로젝트 인원끼리의 정보 공유에 탁월함</p> <p>채널(마케팅)</p> <p>인스타그램 (Instagram)</p> <ul style="list-style-type: none">- Meta DMP를 통한 인스타그램 특정 타겟팅 마케팅 <p>유튜브 (Youtube)</p> <ul style="list-style-type: none">- Youtube 검색 알고리즘을 통한 Youtube 광고 마케팅 <p>디스콰이엇 (Disquiet)</p> <ul style="list-style-type: none">- 프로젝트 연결 및 커뮤티니 플랫폼인 디스콰이엇 배너 광고 | <p>고객군</p> <ol style="list-style-type: none">프로젝트를 진행하는 모든 개발자, 기획자, 디자이너 등 협업을 진행하는 모든 직군함께 공부하며 정보를 공유해야 하는 중, 고, 대학생칸반보드 및 채팅을 통해 소통해야 하는 모든 업무를 가진 직군스스로 공부하며 To Do List, 문서를 저장하고 사용해야하는 일반인 등회의를 진행해야 하는 모든 사람들음성채팅을 하며 하나의 화면을 공유해야 하는 사람들 |
| <p>비용</p> <ul style="list-style-type: none">- 인건비- 서버비용- 서비스 마케팅을 위한 마케팅 비용(Instagram, Youtube 광고 등) | | <p>수익</p> <ul style="list-style-type: none">- 프로젝트 개설 수에 따른 수익 발생 (ex: 개설한 채널 5개 초과시 유료화)- 프로젝트 인원수에 따른 수익 발생 (ex: 10인 초과 프로젝트 유료화)- 프로젝트 칸반보드 섹터이 따른 수익발생(ex: 칸반보드 섹터 5개 초과시 유료화)- 프로젝트 사용 용량에 따른 수익 발생(ex: 프로젝트 문서 페이지 1G 이상 시 유료화) | | |

린캔버스 및 요구사항

| No | 요구사항ID | 요구사항명 | 기능API | 기능명 | 상세설명 |
|----|---------|----------|--------------------------------|---------------|--|
| 1 | BEFORE | 기능 기획 | 프로젝트 기획 | 프로젝트 기획 | 프로젝트 주제 및 기획의도 |
| 2 | | | 프로젝트 요구사항 정의 | 프로젝트 요구사항 정의 | 프로젝트에 필요한 요구사항 정의 |
| 3 | | 로그인 | / | 메인페이지 | 로그인화면이 떠있는 메인페이지 |
| 4 | | | /login | 자체 로그인 기능 | TikiTaka의 회원가입을 통해 아이디, 비밀번호 생성 후 로그인 하는 기능(JWT) |
| 5 | | | /login/kakao | 카카오 로그인 기능 | 카카오 아이디로 TikiTaka에 로그인 하는 기능 |
| 6 | | | /login/google | 구글 로그인 기능 | 구글 아이디로 TikiTaka에 로그인 하는 기능 |
| 7 | | | /login/naver | 네이버 로그인 기능 | 네이버 아이디로 TikiTaka에 로그인 하는 기능 |
| 8 | | | /login/findId | 아이디 찾기 | 아이디를 잊어버렸을 시 본인확인을 통해 비밀번호를 재설정하는 기능 |
| 9 | | | /login/findPwd | 비밀번호 재설정 | 비밀번호를 잊어버렸을 시 본인확인을 통해 비밀번호를 재설정하는 기능 |
| 10 | | 회원가입 | /signUp | 회원가입페이지 이동 | 회원가입 페이지로 이동하는 기능 |
| 11 | | | /signUp/model | 회원가입 | TikiTaka에 인증을 통해 회원가입을 하는 기능 |
| 12 | | | /signUp/Auth | 이메일 인증 | 이메일 인증을 통해 유저를 인증하는 기능 |
| 13 | | | /signUp/profile | 프로필 업로드 | 프로필 사진 파일을 업로드하고 이미지 패스 등을 DB에 저장하는 기능 |
| 14 | | | /signUp/profile | 썸네일 생성 | 프로필 사진 파일을 등록시 작은 size로 썸네일 이미지 파일을 제작 및 저장하는 기능 |
| 15 | PROJECT | 마이페이지 | /myPage | 회원정보 확인 | 유저의 이메일, 전화번호, 주소 등 개인정보 및 프로필을 확인하는 기능 |
| 16 | | | /myPage/edit | 회원정보 수정 | 유저의 이메일, 전화번호, 주소 등 개인정보 및 프로필을 변경하는 기능 |
| 17 | | 헤더 | - | 소리 조절(듣기 말하기) | 모든 페이지에서 공유되는 부분(ex: 헤더나 푸터)에서 전체 소리를 조절하는 기능 |
| 18 | | | /webcam/connect | 화상채팅 | 화상채팅을 실행시키는 기능 |
| 19 | | 프로젝트 리스트 | /project/list | 프로젝트 리스트 | 현재 자신이 입장되어 있는 프로젝트 리스트를 보여주는 기능 |
| 20 | | | /project/search/{requestParam} | 프로젝트 검색 | 프로젝트 리스트에서 원하는 단어로 프로젝트를 검색하는 기능 |
| 21 | | | /project/creation | 프로젝트 만들기 | 프로젝트를 만드는 기능, 초대코드 생성 |
| 22 | | | /project/hidden | 프로젝트 숨기기 | 업데이트가 없는 프로젝트 등을 숨겨서 화면 내에 보여주지 않는 기능 (visible을 바꾸는 기능) |
| 23 | | | /project/{requestParam} | 프로젝트 입장하기 | 프로젝트 주인에게 받은 초대코드를 입력하여 프로젝트에 입장하는 기능 |
| 24 | | | /project/invitation | 프로젝트 초대하기 | 다른 사람이 프로젝트에 입장할 수 있는 초대 코드를 생성 혹은 보내는 기능 |
| 25 | | 프로젝트 페이지 | /chat/ | 실시간 채팅 기능 | 프로젝트 내 사람들과 자유롭게 실시간 채팅을 사용 할 수 있는 기능으로 프로젝트 들어올시 웹소켓 연결 |

린캔버스 및 요구사항

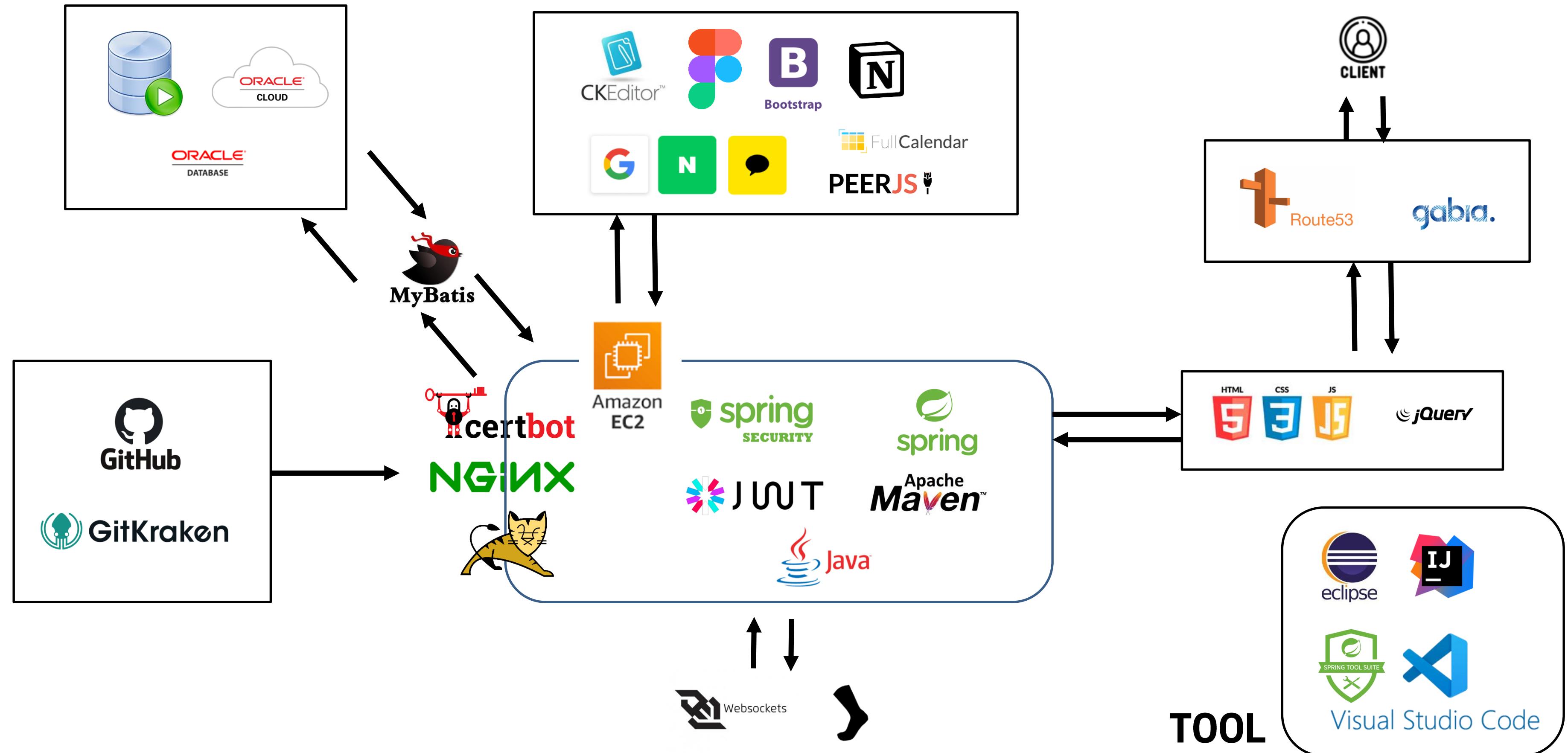
| | | | | | | |
|----|---------|-----------|---------------------------|------------|--|--|
| 25 | DOC | 프로젝트 환경설정 | /chat/ | 실시간 채팅 기능 | 프로젝트 내 사람들과 자유롭게 실시간 채팅을 사용 할 수 있는 기능으로 프로젝트 들어올시 웹소켓 연결 | |
| 26 | | | /alert | 알림기능 | 프로젝트 내 업데이트(신규 채팅 내용, 파일 전송 등)가 발생 시 프로젝트를 구독한 모든 인원들에게 알림 보내기 | |
| 27 | | | /project/modi | 프로젝트 수정 하기 | 프로젝트 주인만 가능한 기능으로 프로젝트명 등을 수정할 수 있게 함 | |
| 28 | | | /project/deletion | 프로젝트 삭제 하기 | 프로젝트 주인이 이제 사용하지 않을 프로젝트를 삭제 하는 기능으로 프로젝트 주인만 가능하게 함 | |
| 29 | | 문서페이지 | /documents/list | 문서리스트 | 프로젝트 내의 문서의 리스트를 보여줌 | |
| 30 | | | /documents/creation | 문서생성 | 프로젝트 내의 새로운 문서를 생성 할 수 있음 | |
| 31 | | | /documents/{pathVariable} | 문서 상세페이지 | 프로젝트내의 문서 상세 내용을 보여줌 | |
| 32 | | | /documents/edit | 문서수정 | 프로젝트내의 문서를 수정할 수 있음 | |
| 33 | | | /documents/deletion | 문서삭제 | 프로젝트내 안쓰는 문서를 삭제 할 수 있음 | |
| 34 | | | /board/list | 칸반보드 리스트 | 프로젝트 내의 칸반보드의 리스트를 보여줌 | |
| 35 | KANBAN | 칸반보드 | /board/sectors/creation | 섹터 생성 | 프로젝트 내의 새로운 칸반보드 섹터를 생성 할 수 있는 기능 | |
| 36 | | | /board/notes/creation | 노트 생성 | 프로젝트 내의 칸반보드 섹터의 노트를 생성 할 수 있는 기능 | |
| 37 | | | /board/sectors/edit | 섹터 수정 | 프로젝트 내의 새로운 칸반보드 섹터를 생성 할 수 있는 기능 | |
| 38 | | | /board/notes/edit | 노트 수정 | 프로젝트 내의 칸반보드 섹터의 노트를 수정 할 수 있는 기능 | |
| 39 | | | /board/sectors/deletion | 섹터 삭제 | 프로젝트 내의 새로운 칸반보드 섹터를 삭제 할 수 있는 기능 | |
| 40 | | | /board/notes/deletion | 노트 삭제 | 프로젝트 내의 칸반보드 섹터의 노트를 삭제 할 수 있는 기능 | |
| 41 | | 일정 페이지 | /schedule | 일정 조회 | 프로젝트 내의 일정을 조회 할 수 있는 기능 | |
| 42 | SCEDULE | | /schedule/add | 일정 추가 | 프로젝트 내의 일정을 추가할 수 있는 기능 | |
| 43 | | | /schedule/edit | 일정 수정 | 프로젝트 내의 일정을 수정할 수 있는 기능 | |
| 44 | | | /schedule/deletion | 일정 삭제 | 프로젝트 내의 일정을 삭제 할 수 있는 기능 | |
| 45 | VOICE | 새로 넣을 기능 | /voiceTalking | 음성채팅 | 프로젝트 내 사람들을 초대해서 음성통화 및 채팅을 할 수 있는 기능 | |
| 46 | SHARE | | /screenSharing | 화면공유 | 프로젝트 내 사람들을 초대해서 자신의 화면을 공유하여 보여주는 기능 | |
| 47 | VIDEO | | /videoConference | 화상회의 | 프로젝트 내 사람들을 초대해서 화상통화 및 채팅을 할 수 있는 기능 | |
| 48 | PRESEN | 최종발표준비 | 발표자료 준비 | | 발표에 필요한 문서작업 및 PPT 제작 | |
| 49 | | | 시연영상 준비 | | 시연 불가시 보여드릴 시연영상 제작 | |
| | | | 발표연습 | | 발표연습 | |
| 50 | | | 최종발표 | | 최종발표 | |

린캔버스 및 요구사항

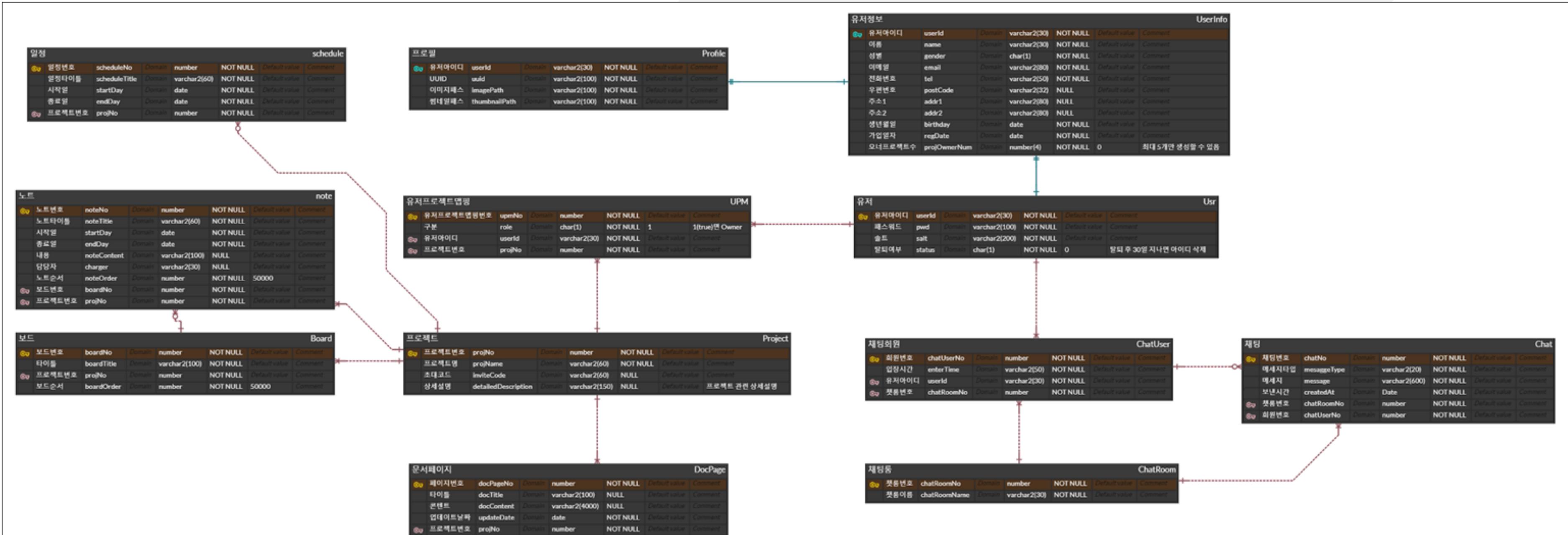
GANNT CHART



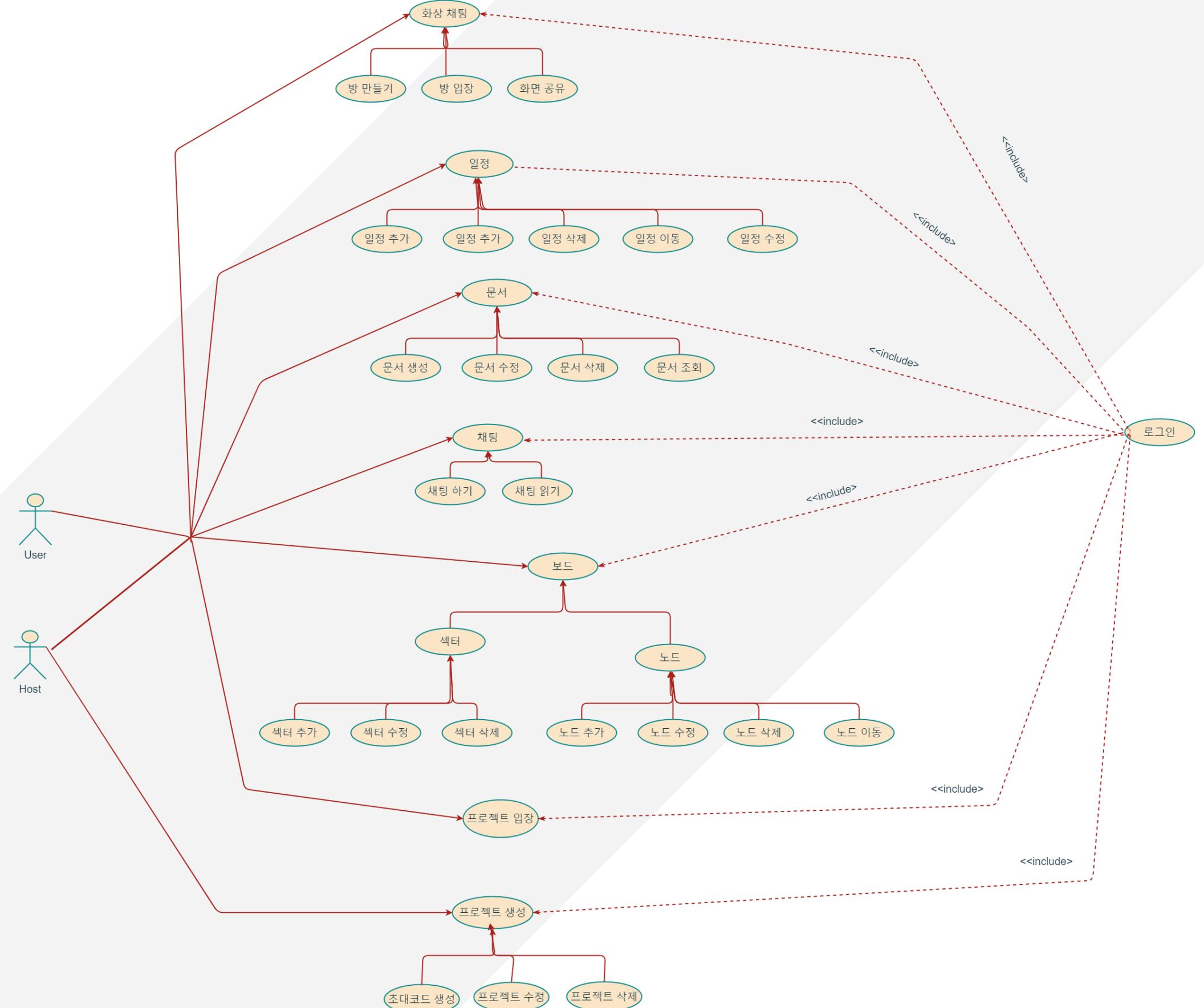
아키텍쳐



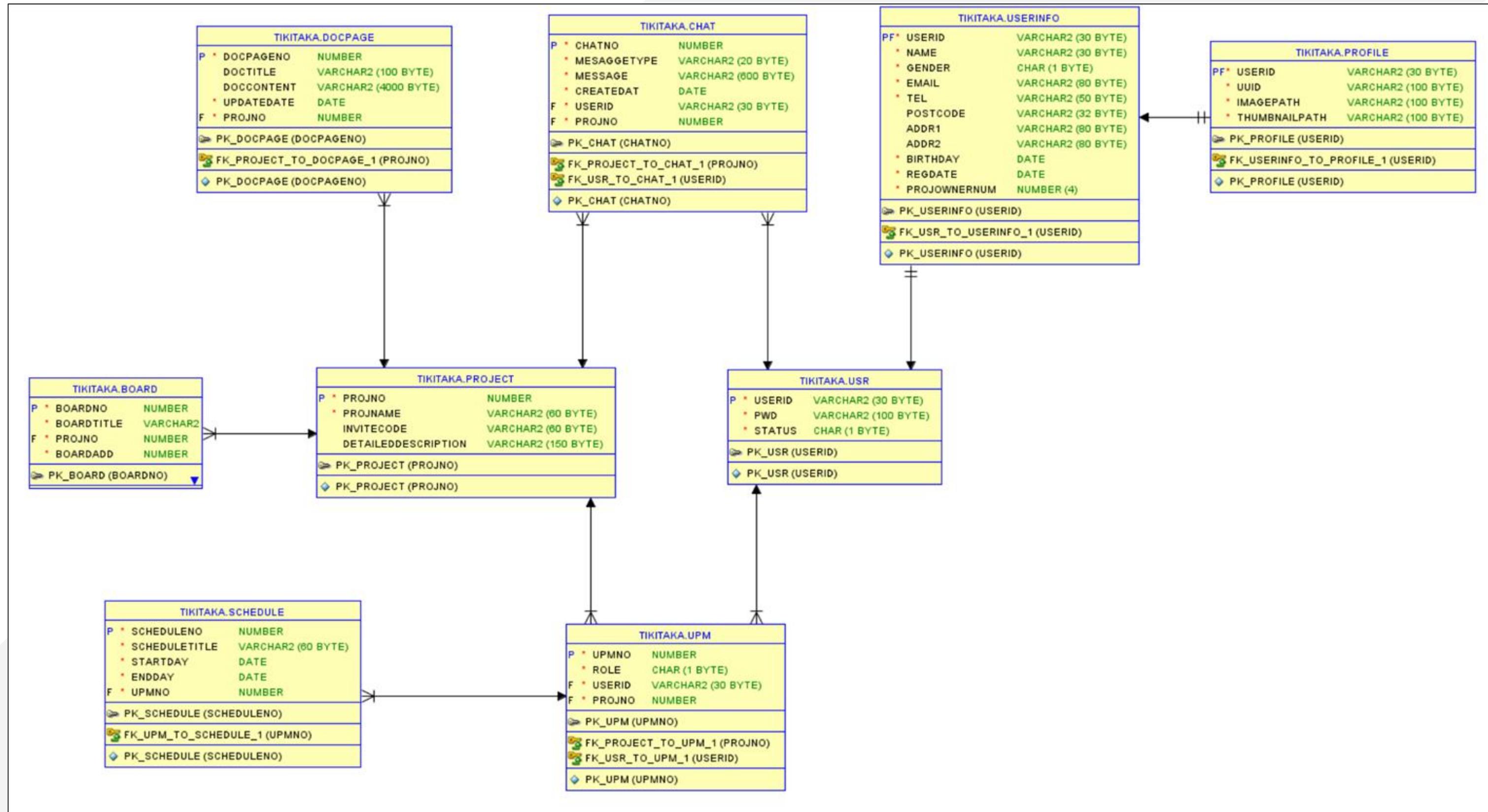
DB모델링 / 다이어그램



유즈케이스 다이어그램



클래스 다이어그램



스토리보드

프로젝트 관련 기능

- 프로젝트 리스트 (메인)
- 프로젝트 리스트 (방 입장 모달)
- 프로젝트 리스트 (방 생성)
- 프로젝트 상세

프로젝트 리스트 (메인)

Description(화면설명)

① 참여 중인 방 검색 기능
- 현재 참여 중인 방을 이름으로 검색할 수 있습니다.

② 방 만들기 / 방 입장하기 버튼
- 사용자가 새로운 방을 만들거나, 코드나 링크를 통해 이미 생성된 방에 참여할 수 있습니다.

③ 옵션 버튼
- 옵션 버튼을 클릭하면 프로젝트 방의 제목, 태그를 수정하거나 방을 삭제할 수 있습니다. 수정할 수 있는 항목은 마스터에게만 주어집니다.

화면명 프로젝트 리스트 (메인 화면) 화면번호 1 J 마이페이지 로그아웃

Logo

① 방 만들기 ② 입장하기

피그마 사용 방법 공유

#프로젝트 #스토리보드

2023. 08. 19

③ 수정하기 삭제하기

프로젝트 리스트 (방 입장 모달)

Description(화면설명)

① 프로젝트 생성 창
- 생성할 프로젝트의 제목과 태그를 작성할 수 있습니다.

② 코드 입력 창
- 유저만든 코드나 링크를 입력해 진행 중인 프로젝트에 참여할 수 있습니다.

③ 생성 / 입장하기 버튼
- 생성 / 입장하기 버튼으로 초대받은 프로젝트에 입장하거나 새로운 프로젝트를 생성할 수 있습니다.

화면명 프로젝트 리스트 (방 입장 모달) 화면번호 2 마이페이지 로그아웃

Logo

① 방 만들기 ② 입장하기

피그마 사용 방법 공유

#프로젝트 #스토리보드

2023. 08. 19

방 만들기

① #프로젝트 #방제작

방 입장하기

② #프로젝트 #방입장

취소 설정하기

취소 설정하기

프로젝트 리스트 (방 생성)

Description(화면설명)

① 새로운 방 생성
- 생성과 입장을 통해 새로운 프로젝트 공유방이 생긴 것을 확인할 수 있습니다.

화면명 프로젝트 리스트 (방 생성 원료) 화면번호 3 J 마이페이지 로그아웃

Logo

① 방 만들기 ② 입장하기

피그마 사용 방법 공유

#프로젝트 #스토리보드

2023. 08. 19

스프링 프로젝트

#프로젝트 #방입장

2023. 08. 19

프로젝트 상세

Description(화면설명)

① 방
- 템에서 메인, 문서, 보드, 일정을 선택해 다른 창으로 넘어갈 수 있습니다.

② 날짜와 로그인 정보
- 오늘 날짜와 로그인 회원의 정보를 출력합니다.

③ 프로젝트 소개
- 위에서부터 방을 생성할 때 작성했던 프로젝트의 제목, 태그, 소개글이 출력됩니다.

④ 오늘 할 일 / D - 1
- 현재 프로젝트에서 오늘 할 일을 마감해야 할 일을 목록형으로 나타냅니다.

⑤ 채팅
- 현재 프로젝트의 참가 중인 유저들과 채팅하여 실시간 소통이 가능합니다.

화면명 프로젝트에 접속한 화면 화면번호 4 그룹 제작

① 메인 문 보드 일정

② 23년 08월 18일 (금) ③ 스프링 프로젝트

○○님 환영합니다!
상태 메시지
상태 설정

#프로젝트 #업업들
진행 중인 프로젝트를 팀원들과 공유해요.

전체 프로젝트 현황

○○님 프로젝트 현황

현재 접속 중 (1) J

오늘 할 일

D - 1

○○님 프로젝트 현황

현재 접속 중 (1) J

서명고
자리 비움 (0)

Recent messages...

스토리보드

문서 관련된 기능

- 문서 생성

- 문서 조회

- 문서 수정

- 문서 삭제

- 채팅 기능 우측 고정

The screenshot displays a user interface for document management. At the top, there's a navigation bar with tabs: '문서' (Document), '화면명' (Screen Name), '문서 공유 화면' (Document Share Screen), '화면번호' (Screen Number) set to 5, and a '그룹 채팅' (Group Chat) section indicating 'Today'. Below the navigation is a 'Description(화면설명)' (Description(Screen Description)) panel containing two items: '문서 목록' (Document List) and '문서 작성창' (Document Creation Window). Item 1 is described as '작성된 문서 목록을 확인할 수 있습니다.' (You can check the list of created documents). Item 2 is described as '문서를 작성할 수 있습니다.' (You can create documents). To the right of this panel is a 'Animation' section with a sub-section 'Overview'. It includes a note about 'Please note' regarding the transition from v0.10 to an older Animation API, a 'Code snippet' for importing React and Framer, and a sample component code. A sidebar on the left shows a '문서 목록' (Document List) with several placeholder messages: 'Lo dolor sit amet consectetur.', 'Lorem ipsum', 'Lorem dolor sit amet consectetur.', 'Lorem ipsum dolor sit amet consectetur.', and 'Lorem ipsum dolor sit amet consectetur.'. A 'Send Message...' button is at the bottom right of the sidebar.

스토리보드

보드 관련된 기능

- 보드 메인페이지

- 보드 노트 추가

- 보드 섹터 추가

- 보드 노트 수정, 삭제

- 보드 섹터 수정, 삭제

- 채팅 기능 우측 고정

보드 메인

Description(화면설명)

1 할 일 목록 상태
- 보드에서는 진행 중인 프로젝트의 할 일, 진행 중인 일, 완료한 일을 나누어 목록을 작성할 수 있습니다.
오른쪽 끝의 위치한 + 버튼을 통해 사용자가 지정한 상태의 섹션을 추가할 수 있습니다.

2 버튼
- + 버튼을 통해 할 일 카드를 추가할 수 있습니다.

화면명 보드 (할 일 목록 화면) 화면번호 6

비번 문서 보드 일정

할 일 진행 중인 일 완료한 일

그룹 채팅

보드 1

Description(화면설명)

1 추가된 카드 수정, 삭제
- 계획을 클릭하면 실제 정보 수정이 가능한 모달 창이 됩니다. - 아이콘을 통해 삭제가 가능합니다.

화면명 보드 할 일 추가 화면 화면번호 7

비번 문서 보드 일정

할 일 진행 중인 일 완료한 일

그룹 채팅

보드 2

Description(화면설명)

1 할 일 모달창 제목 수정
- 할 일 제목을 수정할 수 있습니다.

2 날짜 지정 기능
- 할 일의 날짜, 기간을 지정합니다.

3 추가 내용
- 부가적인 요소들을 추가, 삭제, 완료 합니다.

화면명 할 일 추가 모달창 화면번호 8

비번 문서 보드 일정

할 일 진행 중인 일 완료한 일

그룹 채팅

1 할 일을 추가하세요.
2 할 일
3 할 일을 추가하세요.

스토리보드

일정 관련된 기능

- 전체 일정 조회 기능
- 칸반보드의 일정 DB에서 가져오는 기능
- 일정을 추가하는 기능
- 일정을 상세 조회하는 기능
- 채팅 기능 우측 고정

1. 일정 (Calendar)

Description(화면설명)
일정 확인
1. 보드에서 추가한 할 일 목록들을 일정 항목에서 달력형으로 확인이 가능합니다.

화면명 일정 확인 화면 화면번호 1

2. 일정 상세

Description(화면설명)
일정 추가
1. 일정 추가 버튼을 누르면 모달창을 통해 날짜를 직접 선택하고, 일정을 추가할 수 있습니다.

화면명 일정 추가 화면 화면번호 10

3. 일정 수정 삭제

Description(화면설명)
일정 수정, 삭제
1. 해당 날짜의 일정을 누르면 일정을 수정하거나, 삭제할 수 있는 모달창이 됩니다.

화면명 일정 수정, 삭제 화면 화면번호 11

4. 일정 수정, 삭제 화면

Description(화면설명)
일정을 수정하세요.
1. 일정 수정 화면
2. 일정 삭제 화면

화면명 일정 수정, 삭제 화면 화면번호 11

스토리보드

로그인 기능

- 회원가입 기능

- 로그인 기능

- 소셜 로그인 기능

- JWT 토큰 기능

The storyboard illustrates the user interface for login and registration. It consists of two main panels: '로그인' (Login) on the left and '회원가입' (Registration) on the right.

로그인 (Login Panel):

- Description:** 화면설명 (Description)
- 화면명:** 로그인 (Login)
- 화면번호:** 1
- Content:** Includes a sidebar with 4 steps:
 - ① 로그인: 아이디 패스워드 입력시 로그인이 가능합니다.
 - ② 회원가입: 회원이 아닐시 회원가입버튼을 누르면 회원가입 창으로 넘어갑니다.
 - ③ 비밀번호 찾기: 아이디, 비밀번호를 잊어버렸을 때 찾을 수 있습니다.
 - ④ 간편로그인: 기존 로그인 창고 네이버, 카카오, 구글 간편 로그인이 가능합니다.
- Input Fields:** ID, Password
- Buttons:** ① 로그인, ② 아직회원이 아니신가요? 회원가입, ③ 비밀번호를 잊어버렸어요, ④ 간편 로그인 (with icons for Naver, Kakao, and Google).

회원가입 (Registration Panel):

- Description:** 화면설명 (Description)
- 화면명:** 회원가입 (Registration)
- 화면번호:** 2
- Content:** Includes a sidebar with 1 step:
 - ① 회원가입: 회원가입 창목을 모두 입력하에 유효성 검사가 통과되면 회원가입이 됩니다.
- Input Fields:** ID, P/W, P/W 확인, Email, 휴대전화, 주 소, 주소찾기, 상세주소, 가입완료, 취소.

형상관리



Feat : tiles 프레임워크 적용 #5

Merged yellowrel merged 9 commits into main from SJ last week

Conversation 3 Commits 9 Checks 0 Files changed 330 +49,855 -7

yellowrel commented last week

Feat : tiles 프레임워크 적용
Design : 메티리얼 테마 적용
Fix : 오류수정

yellowrel and others added 9 commits last week

- tiles 프레임워크 적용
- Merge remote-tracking branch 'origin/main' into SJ
- 오류수정
- Feat : tiles 프레임워크 적용

Reviewers: jaen9010, seo-jihyo, jjungyul

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

jjungyul approved these changes last week

View reviewed changes

jjungyul left a comment

확인했습니다.

seo-jihyo approved these changes last week

View reviewed changes

seo-jihyo left a comment

확인했습니다.

yellowrel merged commit dcbfce4 into main last week

Revert



시연

- 기술소개
- 시연
- 트러블슈팅

스프링 시큐리티

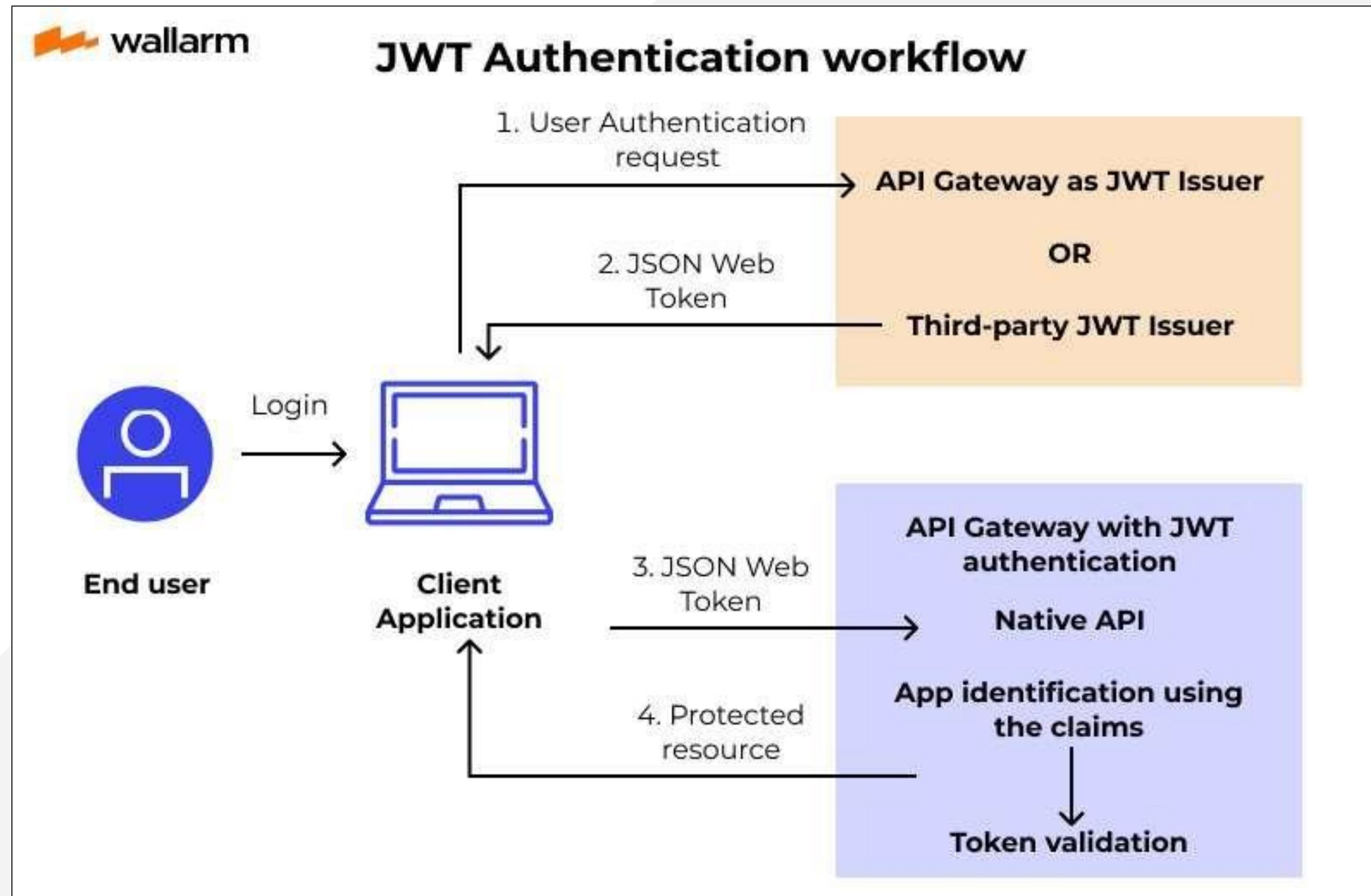
- 📁 com.kosa.tikitaka.security
- 📁 com.kosa.tikitaka.security.filter
- 📁 com.kosa.tikitaka.security.jwt
- 📁 com.kosa.tikitaka.security.provider

- ⌄ com.kosa.tikitaka.security
 - > 📄 FilterSkipMatcher.java
 - > 📄 FormLoginSuccessHandler.java
 - > 📄 UserDetailsImpl.java
 - > 📄 UserDetailsServiceImpl.java
 - > 📄 WebSecurityConfig.java

```
13
14 public class FormLoginSuccessHandler extends SavedRequestAwareAuthenticationSuccessHandler {
15     public static final String AUTH_HEADER = "Authorization";
16     public static final String TOKEN_TYPE = "BEARER ";
17
18     @Override
19     public void onAuthenticationSuccess(HttpServletRequest request, HttpServletResponse response,
20             Authentication authentication) throws ServletException, IOException {
21         final UserDetailsImpl userDetails = (UserDetailsImpl) authentication.getPrincipal();
22         final String token = JwtTokenUtils.generateJwtToken(userDetails);
23         response.addHeader(AUTH_HEADER, TOKEN_TYPE + token);
24     }
25 }
```

```
28 @Configuration
29 @EnableWebSecurity
30 @EnableGlobalMethodSecurity(securedEnabled = true)
31 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
32     private final JWTAuthProvider jwtAuthProvider;
33     private final HeaderTokenExtractor headerTokenExtractor;
34
35     public WebSecurityConfig(JWTAuthProvider jwtAuthProvider, HeaderTokenExtractor headerTokenExtractor) {
36         this.headerTokenExtractor = headerTokenExtractor;
37         this.jwtAuthProvider = jwtAuthProvider;
38     }
39
40     @Bean
41     public BCryptPasswordEncoder encodePassword() {
42         return new BCryptPasswordEncoder();
43     }
44
45     @Override
46     protected void configure(HttpSecurity http) throws Exception {
47         http.csrf().disable();
48
49         //서버에서 인증은 JWT로 인증하기 때문에 Session의 생성을 막습니다.
50         http.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS);
51         http.addFilterBefore(formLoginFilter(), UsernamePasswordAuthenticationFilter.class)
52             .addFilterBefore(jwtFilter(), UsernamePasswordAuthenticationFilter.class);
53
54         http.authorizeRequests()
55             .antMatchers("/motion/signUp", "/user/**", "/js/**", "/css/**", "/img/**", "/ws-stomp/**")
56                 .permitAll()
57             .and()
58                 .authorizeRequests()
59                     .anyRequest()
60                     .authenticated()
61             .and()
62                 .logout()
63                 .logoutUrl("/logout")
64                 .logoutSuccessUrl("/motion/login")
65             .and()
66                 .exceptionHandling()
67                 .accessDeniedPage("/forbidden")
68             .and()
69                 .formLogin()
70                 .loginPage("/motion/login");
71     }
72 }
```

JWT TOKEN 인증방식



스프링 시큐리티

```
com.kosa.tikitaka.security.filter
  FormLoginFilter.java
  JwtAuthFilter.java
```

```
com.kosa.tikitaka.security.jwt
  HeaderTokenExtractor.java
  JwtDecoder.java
  JwtPreProcessingToken.java
  JwtTokenUtils.java
```

```
com.kosa.tikitaka.security.provider
  FormLoginAuthProvider.java
  JWTAuthProvider.java
```

```
13@Component
14public class JwtDecoder {
15  public String decodeUsername(String token) {
16    DecodedJWT decodedJWT = isValidToken(token).orElseThrow(() -> new IllegalArgumentException("유효한 토큰이 아닙니다."));
17
18    Date expiredDate = decodedJWT
19      .getClaim(JwtTokenUtils.CLAIM_EXPIRED_DATE)
20      .asDate();
21
22    Date now = new Date();
23    if(expiredDate.before(now)) {
24      throw new IllegalArgumentException("유효한 토큰이 아닙니다.");
25    }
26}
```

```
12public class JwtTokenUtils {
13
14  private static final int SEC = 1;
15  private static final int MINUTE = 60 * SEC;
16  private static final int HOUR = 60 * MINUTE;
17  private static final int DAY = 24 * HOUR;
18
19  //JWT 토큰의 유효기간 : 1일 (단위: seconds)
20  private static final int JWT_TOKEN_VALID_SEC = 1 * DAY;
21  //JWT 토큰의 유효기간 : 1일 (단위: milliseconds)
22  private static final int JWT_TOKEN_VALID_MILLI_SEC = JWT_TOKEN_VALID_SEC * 1000;
23
24  public static final String CLAIM_EXPIRED_DATE = "EXPIRED_DATE";
25  public static final String CLAIM_USER_NAME = "USER_NAME";
26  public static final String JWT_SECRET = "jwt_secret_!@#$%";
27
28  public static String generateJwtToken(UserDetailsImpl userDetails) {
29    String token = null;
```

```
public static String generateJwtToken(UserDetailsImpl userDetails) {
  String token = null;
  try {
    token = JWT.create()
      .withIssuer("tikitaka")
      .withClaim(CLAIM_USER_NAME, userDetails.getUsername())
      .withClaim(CLAIM_EXPIRED_DATE, new Date())
      .sign(generatedAlgorithm());
  } catch (Exception e) {
    e.printStackTrace();
  }
  return token;
}

private static Algorithm generatedAlgorithm() {
  return Algorithm.HMAC256(JWT_SECRET);
}
```

소셜로그인 API 사용

TIKITAKA

개요 API 설정 멤버관리 로그인 통계 API 통계 Playground(Beta)

네이버 로그인 서비스에 한하여 검수가 진행됩니다.

승인 최종 검수 완료일: 2023.09.20

소명 내용 입력

제공 정보 활용처 수정하였습니다.
이메일을 통해 초대코드를 보내고 프로젝트에 입장하는 로직에 사용 됩니다.
회원 이름을 통해 채팅을 하거나 담당자를 정해 업무 분담을 할 수 있습니다.

104/500

검수 상태

파일 첨부 (추가 제출 서류 또는 소명 자료)

https://developers.naver.com/doc/review_202309/t4OWgJ98r5KZ...

https://developers.naver.com/doc/review_202309/t4OWgJ98r5KZ...

- pdf, jpg, gif, png 및 오피스 파일 형식을 첨부하실 수 있습니다.
- 5MB 이하의 파일을 최대 5개까지 업로드 할 수 있습니다.
- [네이버 로그인 검수 가이드](#)를 참고하세요.

사유에 따른 적절한 조치 이후 검수를 다시 요청하실 수 있습니다.

아래 정보들은 [API 설정] 탭에서 조회할 정보로 선택하신 항목입니다.

실시간 채팅

The screenshot shows a web-based real-time messaging application. At the top, there's a browser header with tabs for '메인 페이지' and 'localhost:8080/tikitaka/project/4'. Below the header, the page title is 'TIKITAKA'.

The main navigation menu includes links for '메인' (Main), '문서' (Document), '보드' (Board), '일정' (Schedule), '화상채팅' (Video Chat), and '마이페이지' (My Page). A sidebar on the left displays the date '23년 09월 26일' and several progress bars for different users: 정준열님 (Progress: ~10%), 이재진님 (Progress: ~10%), 서지효님 (Progress: ~10%), and 황승준님 (Progress: ~10%).

The central content area features a '스프링 프로젝트' (Spring Project) section with a subtitle '#프로젝트 #협업툴' and a message '진행 중인 프로젝트를 팀원들과 공유해요'. Below this, there are two columns: '오늘 할 일' (Today's Tasks) and 'D - 1'.

The '오늘 할 일' column contains three items:

- 메인페이지 뷰 만들기: 메인페이지 뷰는 완성해야 스크립트나 다른 기능은 구현할 수 있음. 구현 필요.
- 프로젝트 이동 기능: 각 프로젝트 번호를 가지고 프로젝트 메인페이지로 이동하는 기능 구현 필요.
- 프로젝트 생성 기능: 이메일을 통해 프로젝트 초대코드를 초대할 사람의 이메일로 보내 초대하는 기능.

The 'D - 1' column contains two items:

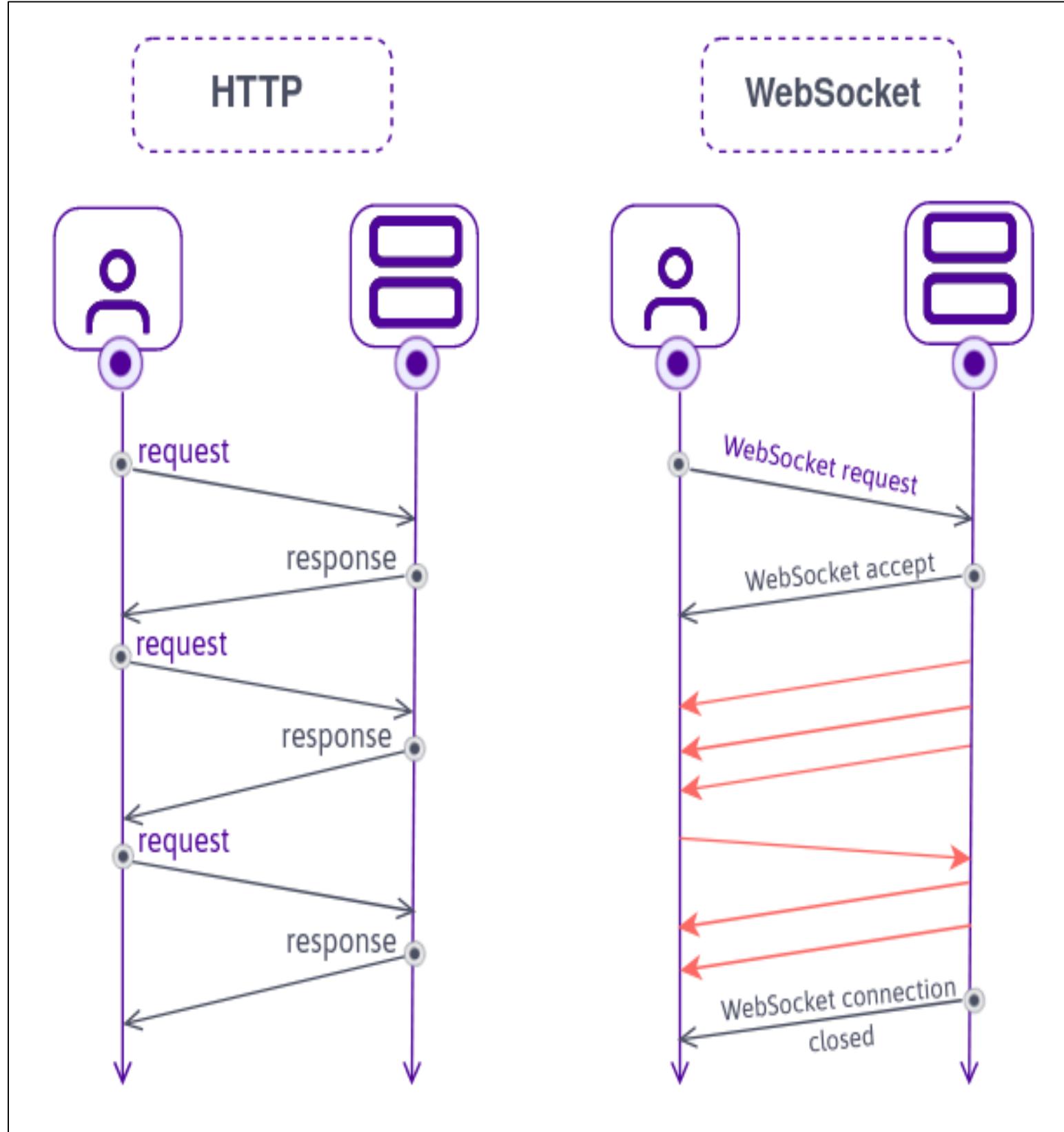
- 프로젝트 수정 기능: 프로젝트의 상세설명이나 타이틀 등을 수정할 수 있는 기능을 구현.
- 프로젝트 삭제 기능: 프로젝트 번호를 가지고 프로젝트를 삭제할 수 있는 기능 필요. 삭제 기능 구현.

On the right side, there's a real-time chat window showing messages between 'T' (Team) and '정준열' (Jung Junyeol). The messages are:

- T: 안녕하세요!
- 정준열: 저는 정준열입니다!
- T: 우리 프로젝트 같이 한번 잘해봐요!!!!
- 정준열: 좋습니다!
- T: 팀명은 뭐로 지을까요?
- 정준열: 불4조 어때요?
- T: 굳굳!!!! 1안읽음

At the bottom, there's a message input field with a placeholder '메세지를 입력하세요.' and a send button. The footer includes system status icons like weather (23°C), search, and various application icons.

실시간 채팅

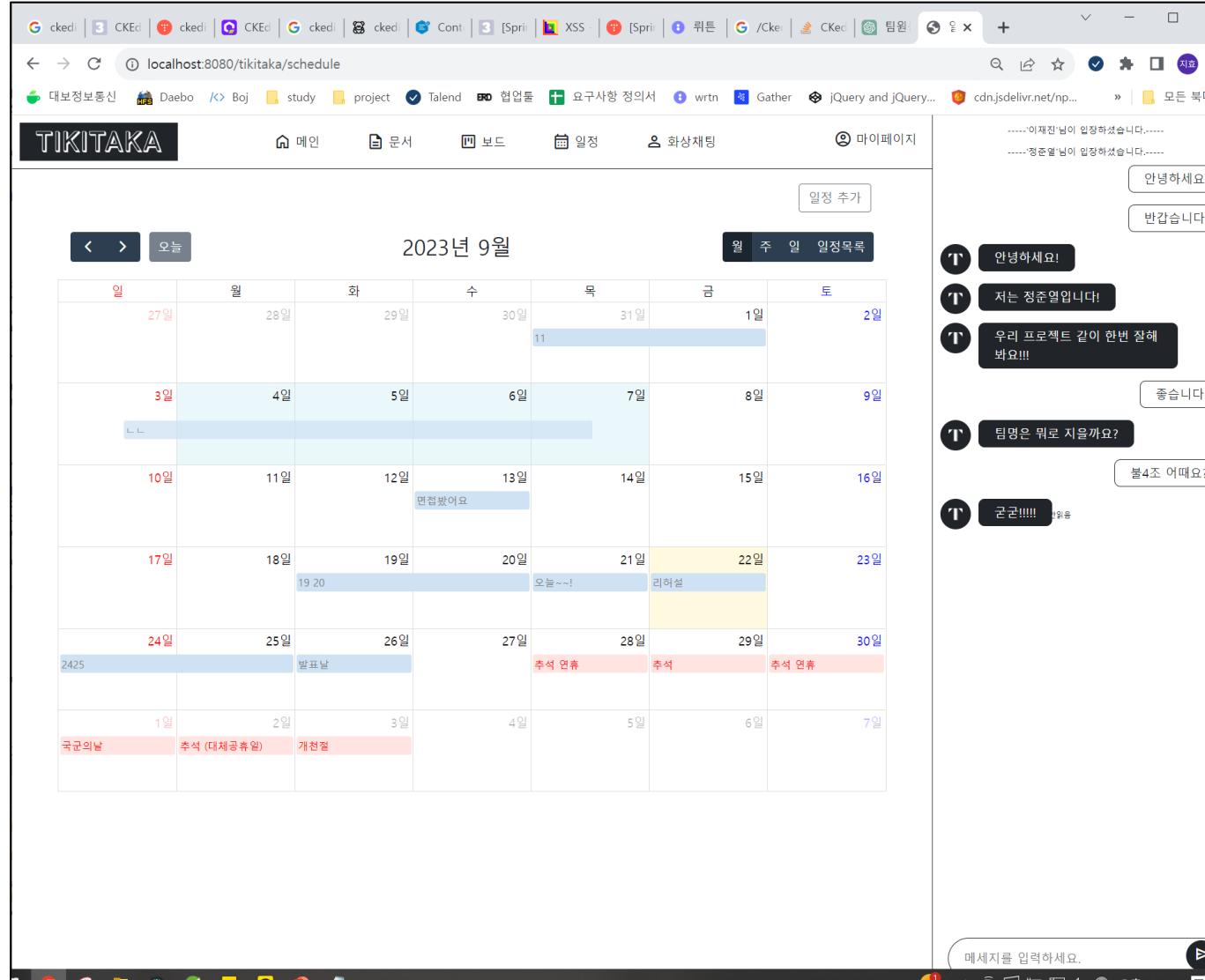


```
9 @Configuration
10 @EnableWebSocketMessageBroker
11 public class WebSocketConfig implements WebSocketMessageBrokerConfigurer {
12
13     @Override
14     public void registerStompEndpoints(StompEndpointRegistry registry) {
15         // 인자로 들어가는는 "/ws-stomp"는 첫 websocket 3-way handshake를
16         // 하기 위한 주소
17         // cors 설정 과 sockJS설정을 할 수 있다.
18         // handler를 설정안해도 됨
19         registry.addEndpoint("/ws-stomp").setAllowedOrigins("*").withSockJS();
20     }
21
22     @Override
23     public void configureMessageBroker(MessageBrokerRegistry registry) {
24         // enableSimpleBroker spring에서 제공하는 내장브로커 사용
25         // 해당 파라미터를 prefix로 받고 있는 경우 그 메세지를 브로커가 처리하겠다
26         // "/queue"는 1대1 메세지 , "/topic" 1대 다로 메세지를 처리할때
27         registry.enableSimpleBroker("/sub");
28         // setApplicationDestinationPrefixes
29         // 바로 브로커로 가는 경우가 아니라 메세지가 가공이 필요할때 handler로 메세지가
30         // 라우팅 되도록 하는 설정
31         registry.setApplicationDestinationPrefixes("/pub");
32     }
}
```

/ws-stomp를 endPoint로 잡아 ws://호스트이름/ws-stomp로 들어오는 요청에 대해 cors설정을 진행하고 sockJS를 사용하였습니다.

/pub으로 들어오는 모든 메세지에 대해 /sub하고 있는 유저들에게 메세지를 뿐여주었습니다.

실시간 채팅



```
//drag & drop으로 일정 수정
eventDrop: function(info) {
    var start = info.event.start;
    var end = info.event.end - 1;
    var eventData = {
        scheduleNo: info.event.id,
        scheduleTitle: info.event.title,
        startDay: moment(start).format('YYYY-MM-DD'),
        endDay: end ? moment(end).format('YYYY-MM-DD') : moment(start).format('YYYY-MM-DD')
    };

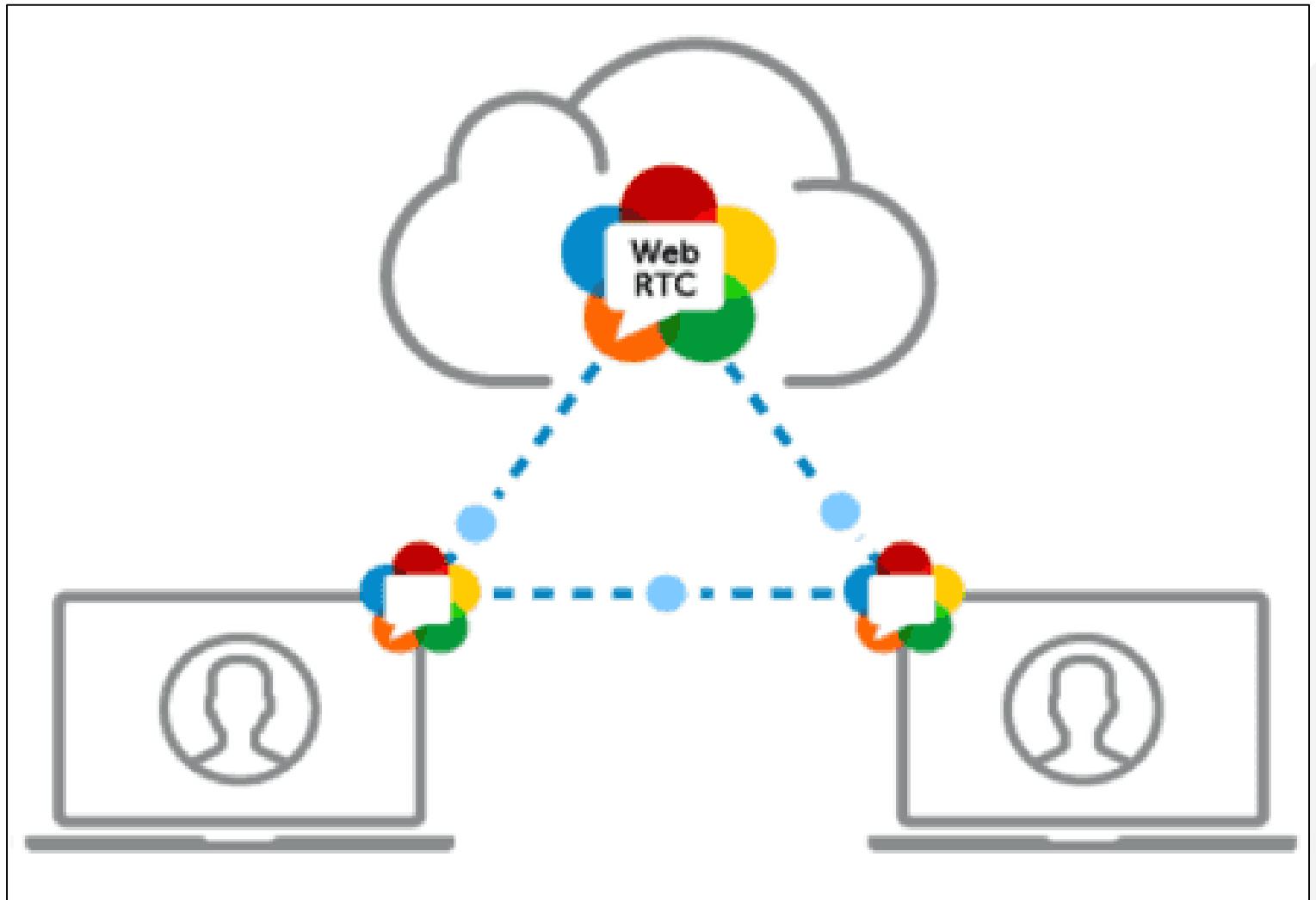
    $.ajax({
        url: "schedule/edit",
        type: "POST",
        contentType: "application/json",
        data: JSON.stringify(eventData),
        success: function(response) {
            if(response.status === 'success') {
                alert("일정이 수정되었습니다.");
                location.reload();
            } else {
                alert("일정 수정에 실패했습니다.");
            }
        }
    });
},
```

FullCalendar 라이브러리에서 제공하는 eventDrop기능이 일정을 드래그 앤 드롭하여 이동할 때 실행됩니다.

일정의 시작(start)과 종료(end) 날짜 정보 및 타이틀과 아이디값을 추출해 info 객체에 저장하고 종료 날짜를 YYYY-MM-DD 형식으로 바꿉니다.

end 값이 없는 경우 (하루) 시작 날짜와 동일하게 설정하고 \$.ajax 함수를 사용하여 서버로 일정 수정 요청을 보냅니다.

WebRTC & PEERJS



PEERJS 

```
function createRoom() : void {  
    console.log("Creating Room")  
    document.getElementById( elementId: "local-video").hidden = false;  
    let room = document.getElementById( elementId: "room-input").value;  
    if (room == " " || room == "") {  
        alert("방 번호를 입력해주세요.")  
        return;  
    }  
    room_id = PRE + room + SUF;  
    peer = new Peer(room_id)  
    peer.on('open', (id) : void => {  
        console.log("Peer Connected with ID: ", id)  
        hideModal()  
        getUserMedia({ video: true, audio: true }, (stream) : void => {  
            local_stream = stream;  
            setLocalStream(local_stream)  
        }, (err) : void => {  
            console.log(err)  
        })  
        notify( msg: "다른 사람이 입장할때까지 대기해주세요.")  
    })  
    peer.on('call', (call) : void => {  
        call.answer(local_stream);  
        call.on('stream', (stream) : void => {  
            setRemoteStream(stream)  
        })  
        currentPeer = call;  
    })  
}
```

WebRTC (Web Real Time Communication)

웹 브라우저 간에
플러그인의 도움 없이 서로
통신할 수 있도록 설계된
API입니다.
음성 통화, 영상 통화, P2P
파일 공유 등으로 활용될
수 있습니다.

PeerJS는 WebRTC를
사용하기 편하게
Wrapping 해둔
라이브러리입니다.