

# toPangyo팀 아키텍처

팀장

FE 우명규

FE 류나연

BE 선원종

BE 전해지

BE 최성민

# 목차

**01** 프로젝트 소개와 기능

**02** 프론트엔드 구조 및 flow

**03** 전체 아키텍처 및 개별 아키텍처

**04** DB Schema / ERD

**05** API 명세서 및 사용 API

**06** 업무분담

# 프로젝트 소개 및 기능



## 플레이스 매칭 커뮤니티 플랫폼

---

위치를 기반으로 하여 유저간 매칭을 통해  
유저들이 플레이스를 직접 선택하여  
함께 즐길 수 있는 콘텐츠 제공

‘진짜 서울’ + 채팅 기능

# 프로젝트 주제

## 이 프로젝트를 선택한 이유

### 프론트엔드

1. 지도 및 위치 기능을 사용하고 가독성 있게 UI를 배치
2. 오류없이 계획대로 기능을 잘 구현하는 것

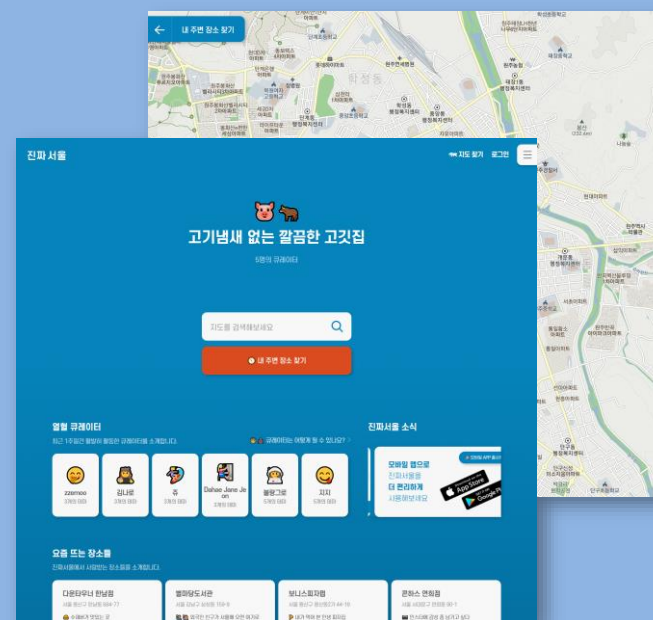


모두의 니즈를 충족?

### 백엔드

1. 직접 경험해보지 않았던 통신 소켓 프로젝트 경험
2. 디스코드와 같은 메신저를 만들어보는 것

## 진짜 서울

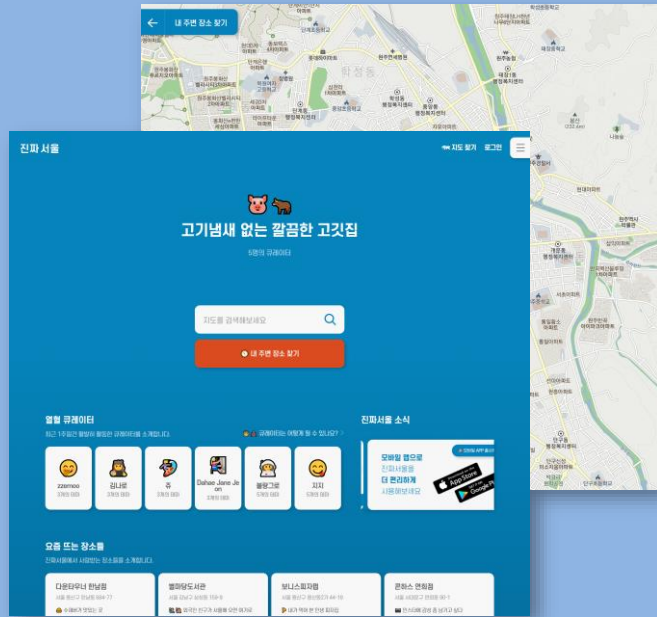


# 프로젝트 주제

## 양방향 소통의 부재

단순히 플레이스에 대한 코멘트와 그 코멘트에 대한 의견만 남길 수 있을 뿐  
유저 - 유저간 양방향 소통을 할 수 있는 기능이 없어 단지 지도만 보는 용도

## 진짜 서울



+

채팅 기능  
매칭 기능

=

동세권

## 동세권 기능

위치 기반 모집글  
리스트 생성(+재검색)  
카테고리 분류  
검색

매칭

메시지 전송  
메시지 받기  
방장 권한  
퇴장 기능

채팅

알림 전송  
알림 받기

알림

로그인  
회원가입  
프로필 설정

유저

글쓰기  
글 수정 / 삭제  
시간 및 장소 설정  
신청 기능

포스팅

프론트엔드 구조  
및  
주요 부분 FLOW



# 프론트엔드 컴포넌트 구조

## <Navbar>

- 현재 로그인한 사용자의 닉네임을 표시해주고 동세권 로고, 여러 메뉴 표시

## /signin <SignInPage>

- <SignInBox> : 사용자의 id, password를 입력

## /signup <SignUpPage>

- <SignUpBox> : 사용자의 닉네임, id, password, 생년월일, 전화번호, 성별, 이메일의 정보를 입력

## /mypage <Mypage>

- <Profile> : 나의 프로필사진, 닉네임 등의 정보를 확인할 수 있으며 회원정보를 수정할 수 있는 버튼을 표시
- <Alert> : 매칭성공/실패여부, 채팅 알림을 확인
- <History> : 내가 매칭되었던 내역을 확인

## /mypage/edit <EditMyPage>

- 프로필사진과 유저의 정보를 수정할 수 있는 컴포넌트

## / <HomePage>

- <Map> : 지도와 유저의 현재위치를 표시
- <MatchingPost> : 신청한 모집글이 대기 중인지 매칭완료된건지 표시
- <Toolbar> : 검색, 카테고리 찾기, 모집글 리스트 숨기기 기능
- <PostList> : 모집 글들을 리스트로 표시
  - <Post> : 모집글 제목, 내용, 인원, 시간, 모집상태 등 정보를 카드형태로 표시
- <WritePost> : 모집글을 작성
- <DetailPost> : 모집글 카드클릭 시 모집글의 상세정보 표시
- <ApplyPost> : 신청버튼 클릭시 신청여부 확인
- <MatchingDetail> : 매칭이 완료되어 매칭된 사람들과 채팅, 유저확인, 시간/장소 설정(방장-참여자 간 표시되는게 다르게)
  - <MatchingChat> : 매칭완료된 사람들과 채팅
  - <MatchingUser> : 매칭완료된 사람들을 확인
  - <MatchingPlace> : 매칭완료된 사람들과 장소를 결정
  - <MatchingTime> : 매칭완료된 사람들과 시간을 결정
- <CheckGather> : 매칭완료된 유저들이 만났는지 여부 확인
  - <UserRating> : 모임 이후 유저들 평가
  - <FailReason> : 모임에 실패하였을 때 이유 확인

## /users/{userId} <UserPage>

- <Profile> : id에 맞는 유저의 프로필사진, 닉네임 등 유저에 대한 정보를 확인
- <History> : 해당 유저가 매칭되었던 내역을 확인

# 프론트엔드 주요 Page

## 로그인

동세권

id

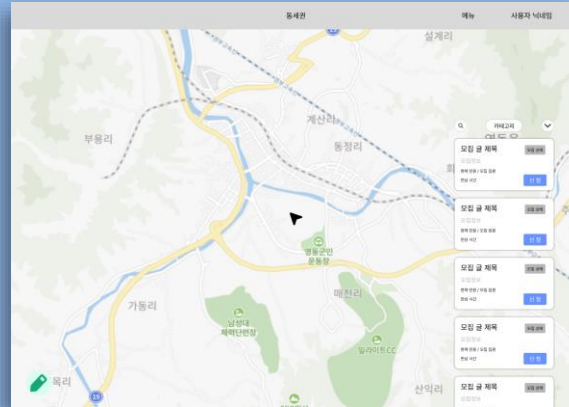
password

로그인

Don't have an account? [Sign Up](#)

[Forgot Password!](#)

## 메인



## 매칭

매칭 페이지 디자인. '모집글 제목'과 '모집글 내용'을 입력하고, '매칭' 버튼을 클릭하면, '매칭 결과' 테이블이 표시됩니다.

유저 1	유저 2	유저 3	유저 4
유저 1	유저 2	유저 3	유저 4
유저 1	유저 2	유저 3	유저 4
유저 1	유저 2	유저 3	유저 4

## 회원가입

회원가입

nickname

name

id

비밀번호를 입력해주세요 (8~16자, 영문, 숫자)

가입하기

Already have an account? [Sign In](#)

## 마이

마이 페이지 디자인. '내 정보'와 '내 정보 수정' 버튼을 클릭하면, '내 정보' 테이블이 표시됩니다.

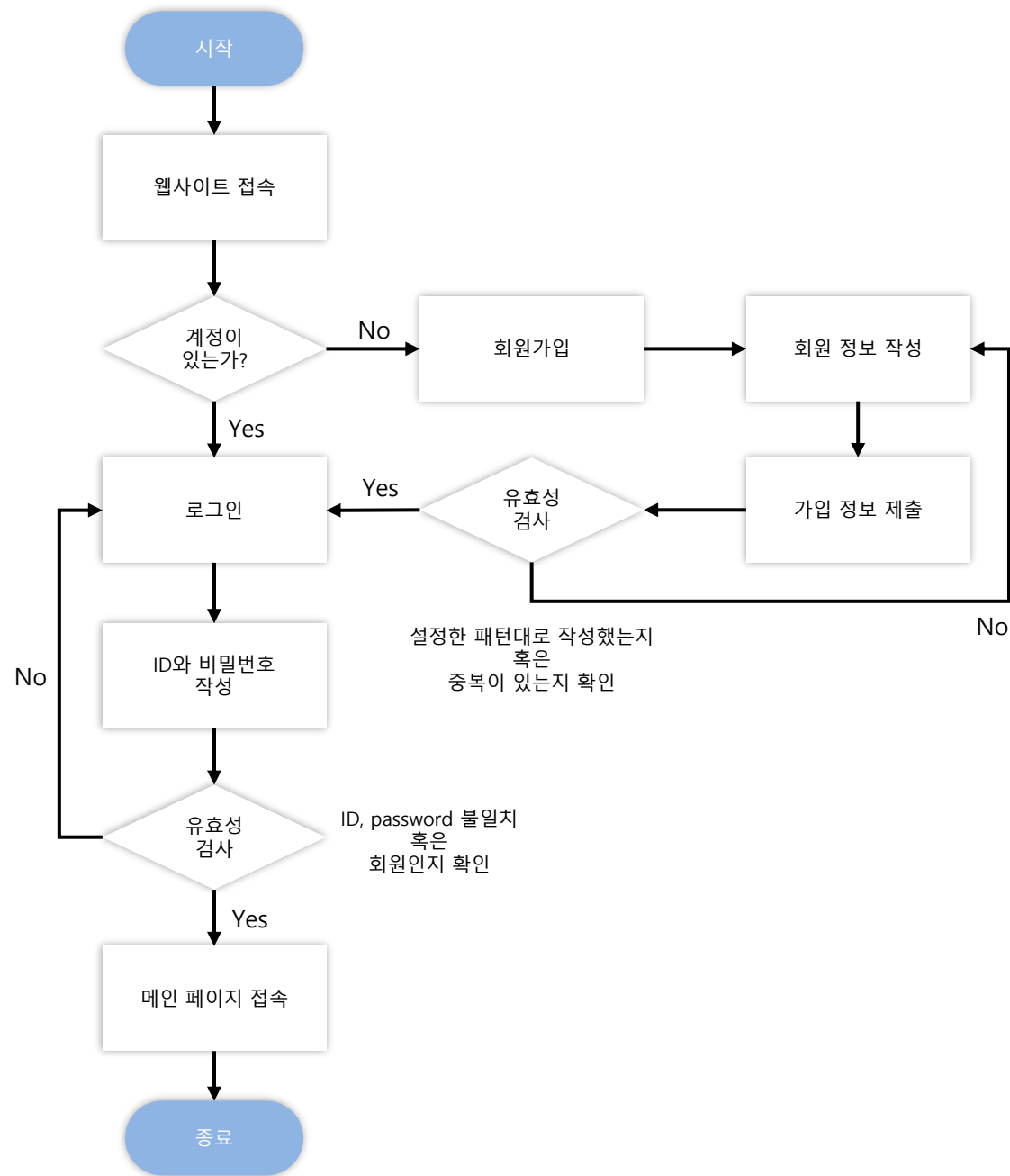
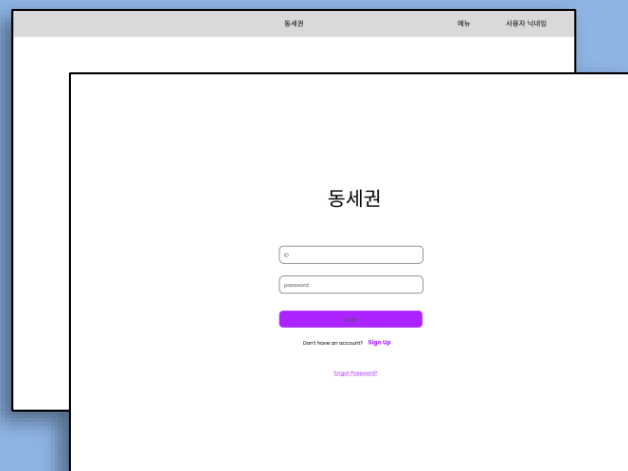
유저 1	유저 2	유저 3	유저 4
유저 1	유저 2	유저 3	유저 4
유저 1	유저 2	유저 3	유저 4
유저 1	유저 2	유저 3	유저 4

## 모집 글

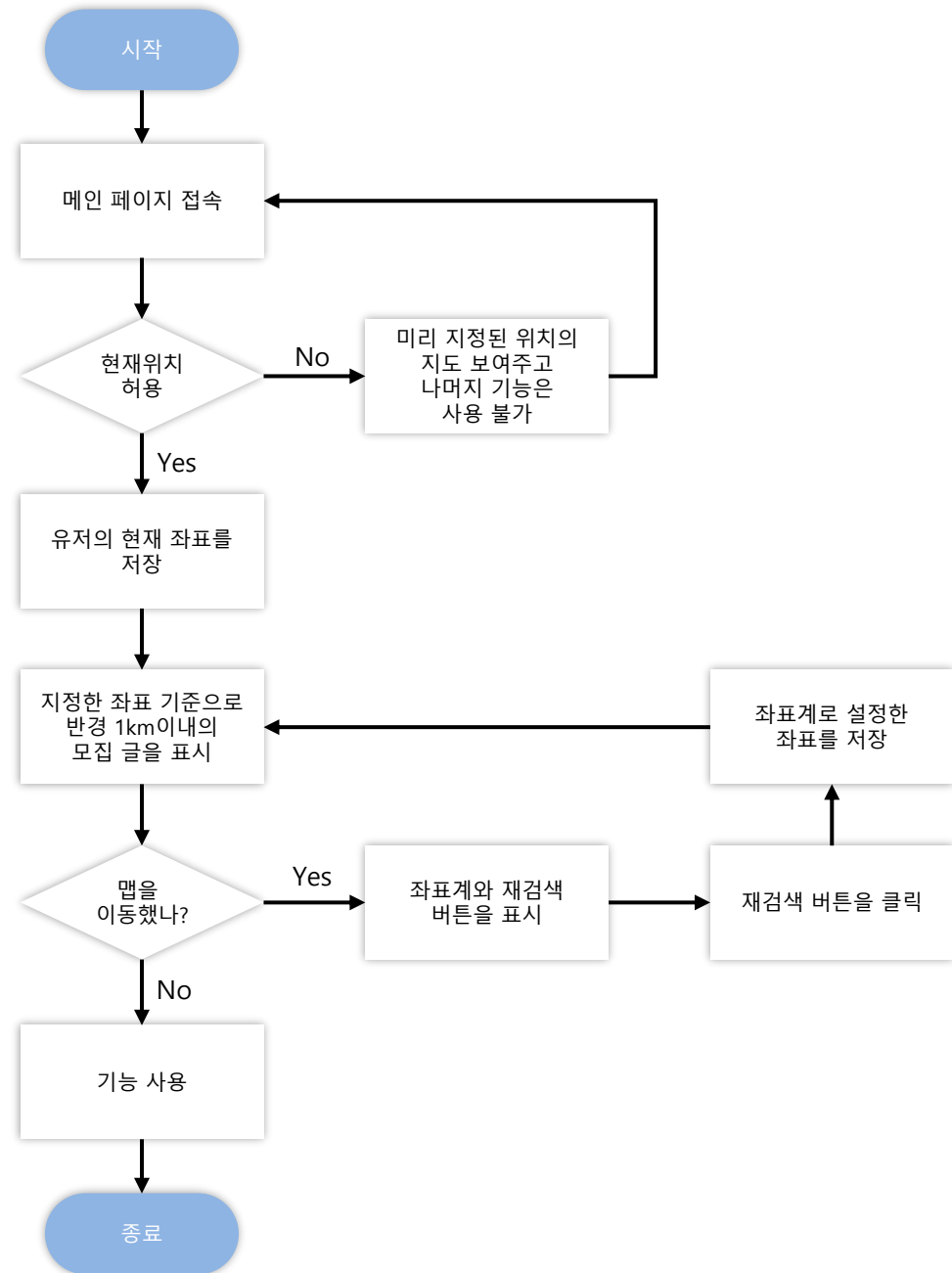
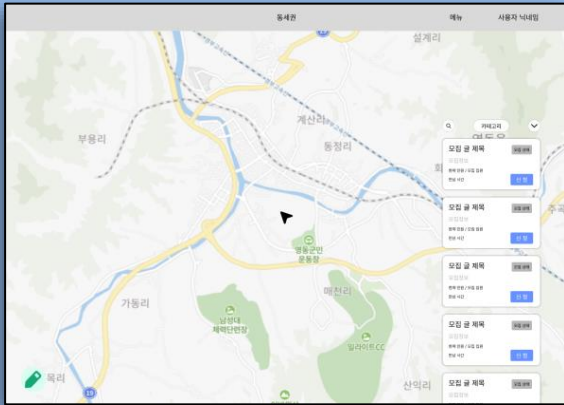
모집글 페이지 디자인. '모집글 제목'과 '모집글 내용'을 입력하고, '모집' 버튼을 클릭하면, '모집 결과' 테이블이 표시됩니다.

유저 1	유저 2	유저 3	유저 4
유저 1	유저 2	유저 3	유저 4
유저 1	유저 2	유저 3	유저 4
유저 1	유저 2	유저 3	유저 4

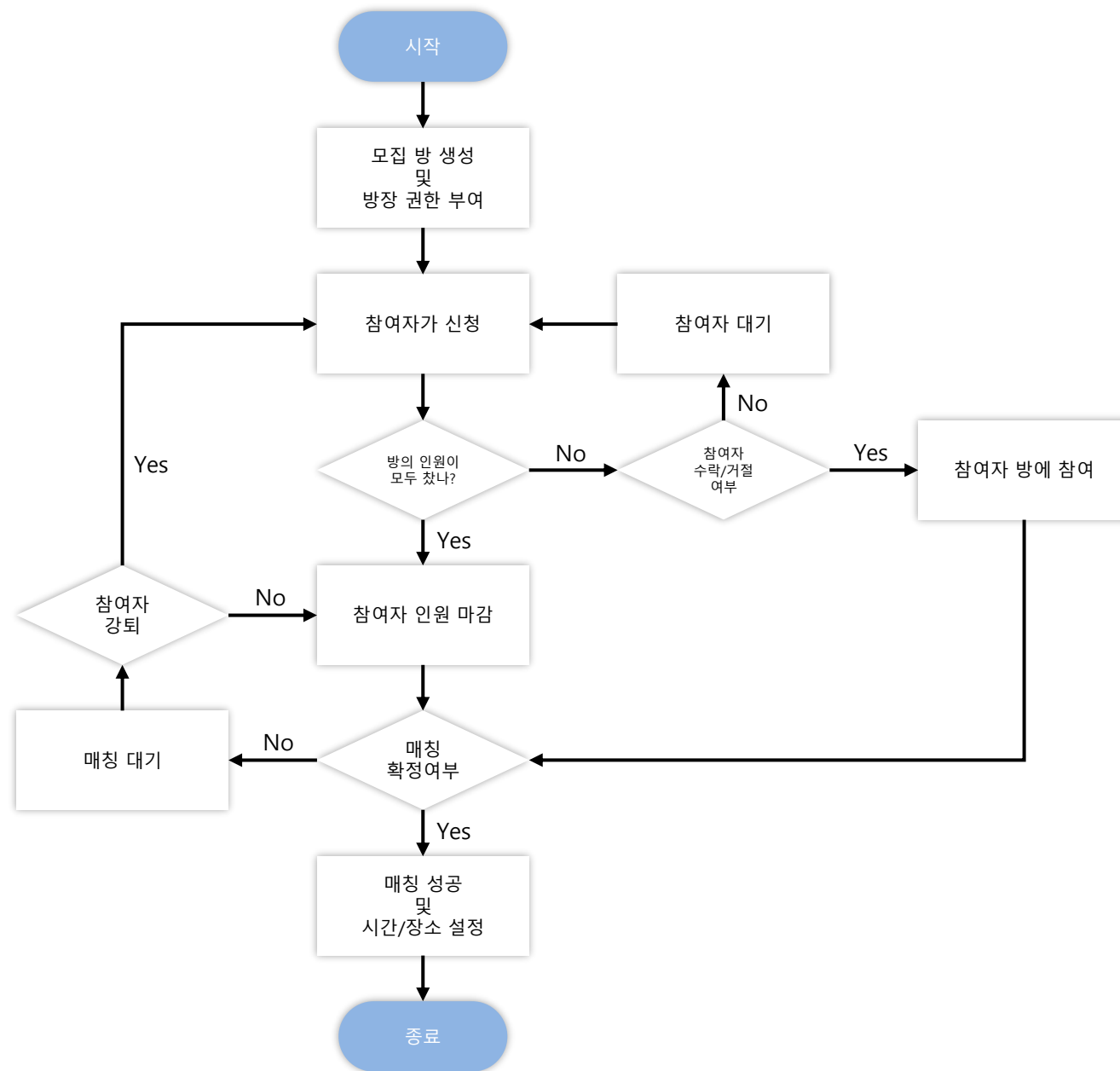
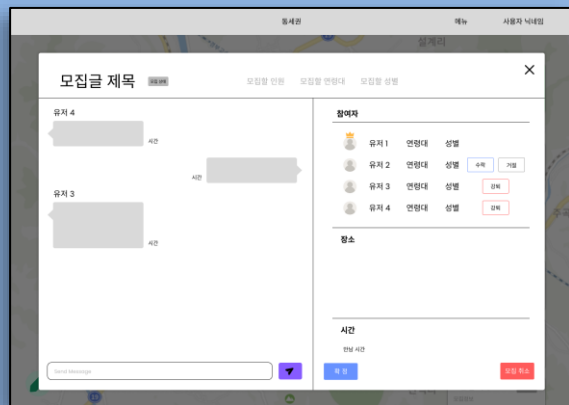
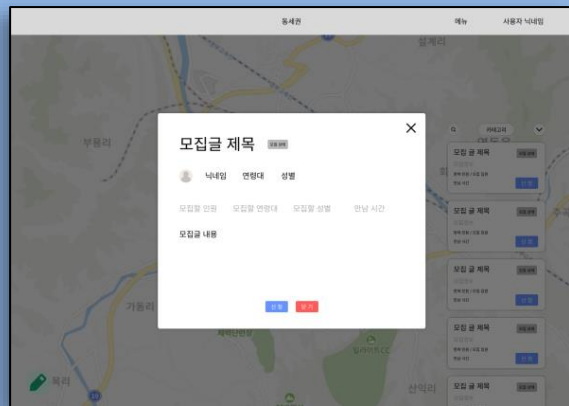
# 로그인 & 회원가입 주요 Flow



# 메인 페이지 주요 Flow

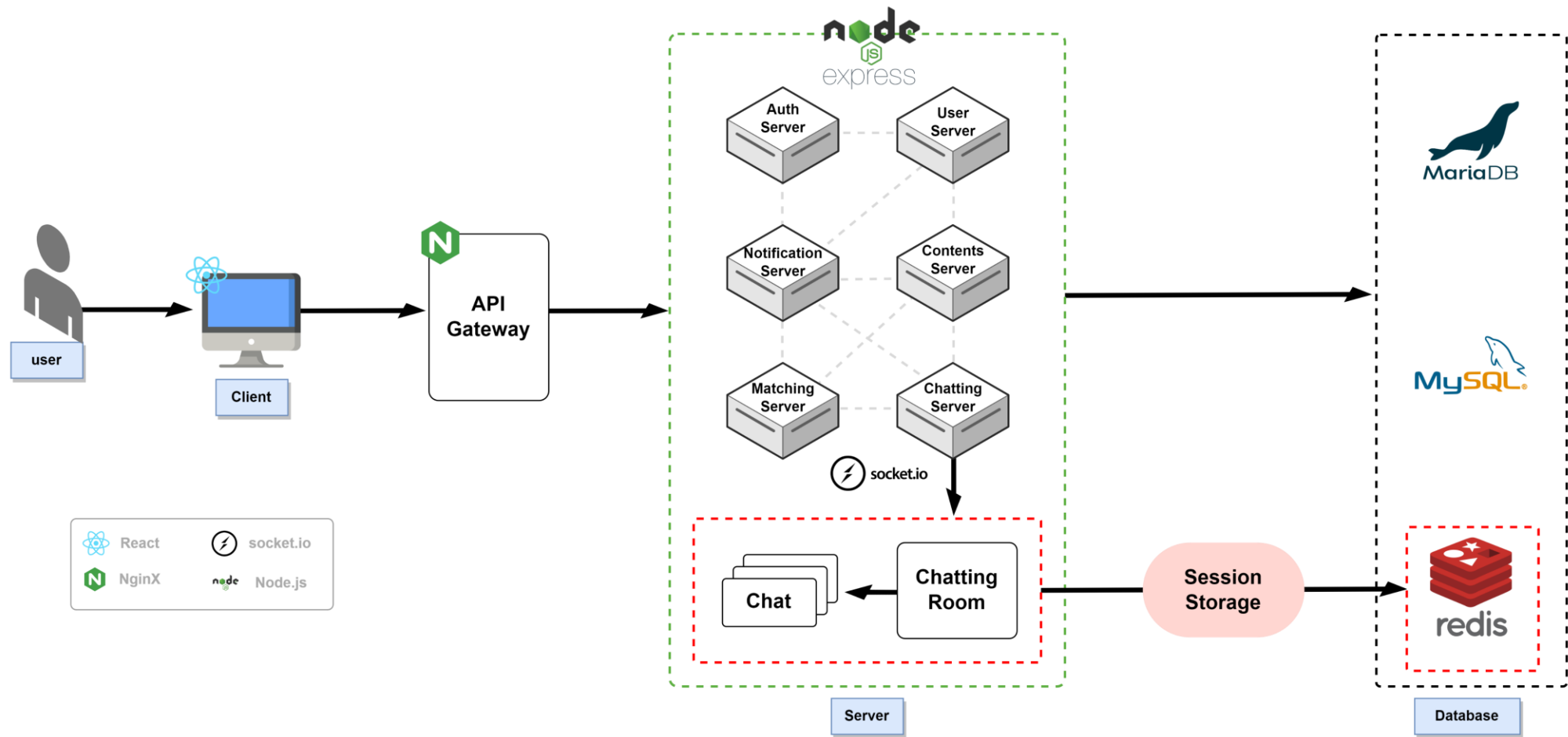


## 매칭 & 포스팅 주요 Flow

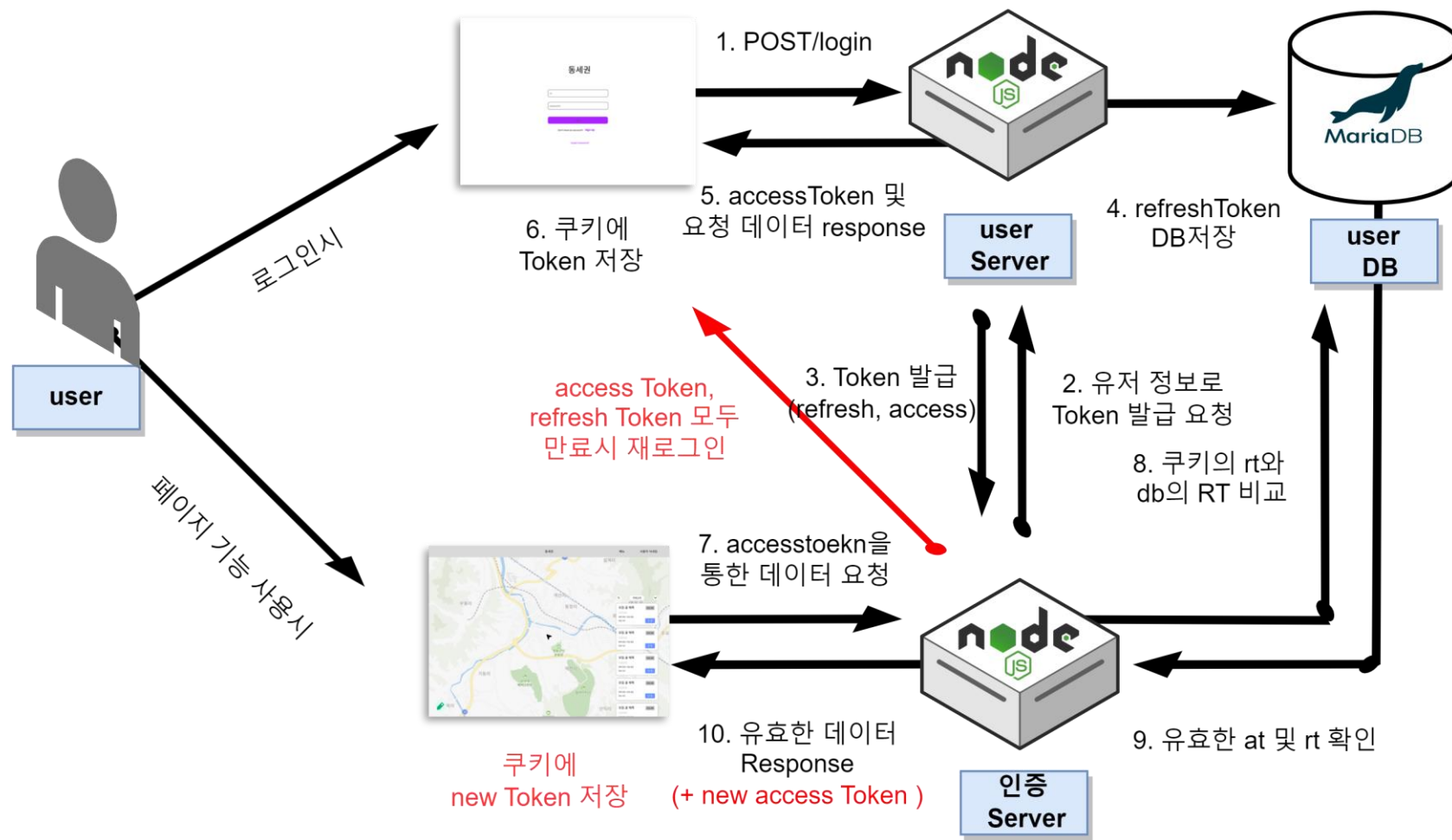


# 전체 아키텍처 구조 및 개별 서버 아키텍처

# 전체 아키텍처 구조

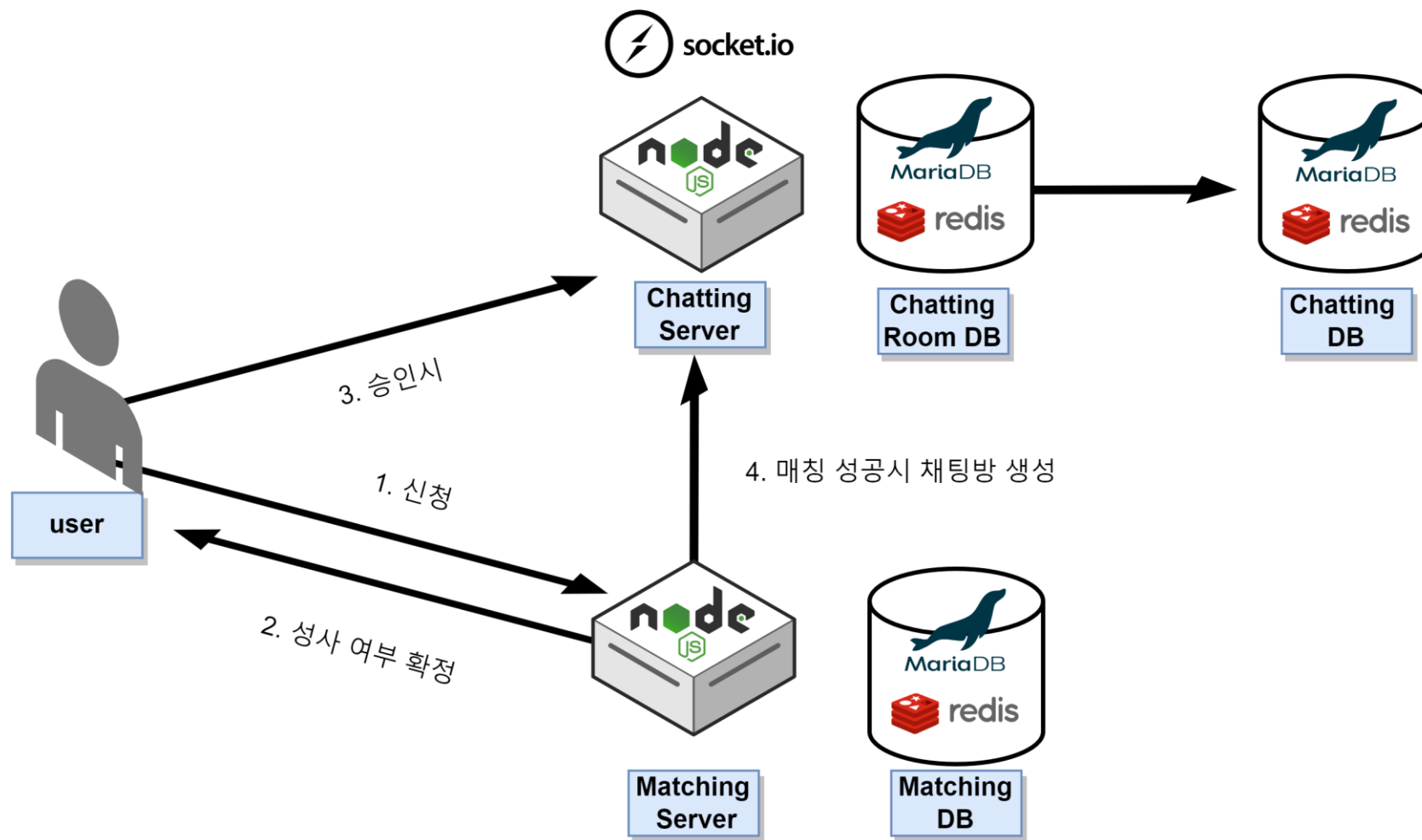


## 개별 서버 아키텍처 – User(주요기능) & Auth server

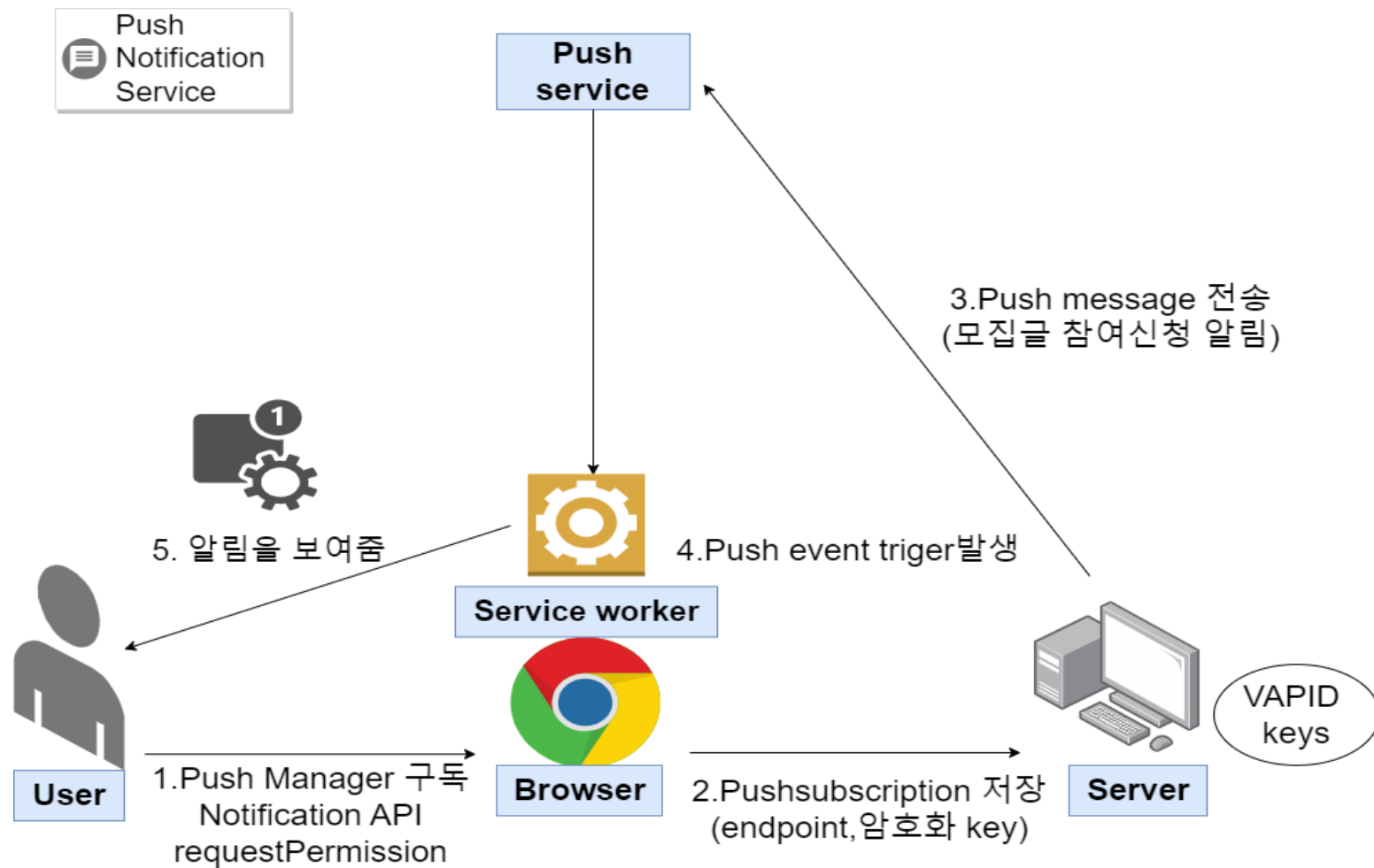




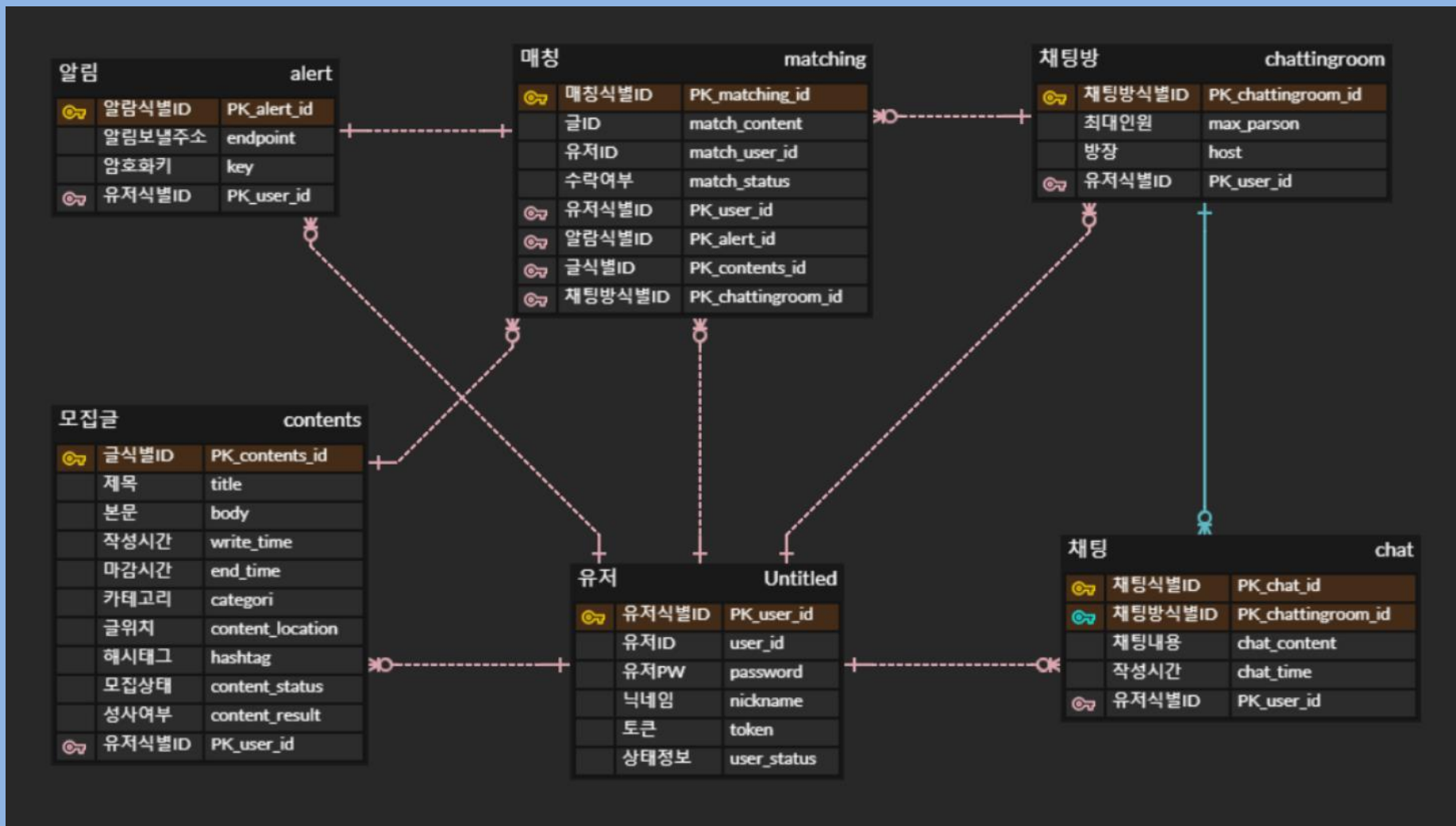
## 개별 서버 아키텍처 – Chatting & Matching server



## 개별 서버 아키텍처 – Notification server



## DB schema / ERD



# API 명세서 및 사용 API

# API 명세서

200 - OK	데이터 요청 정상
400 - Bad Request	요청이 비정상이므로 응답 불가
401 - Unauthorized	올바른 API가 아님
402 - Request Failed	어떠한 요인으로 요청 실패
403 - Forbidden	API를 요청을 수행할 수 있는 권한이 없음
404 - Not Found	요청한 리소스가 없음
429 - Too Many Requests	너무 많은 API요청이 있는 것
500, 502, 503, 504 - Server Errors	서버에 비정상적인 에러

POST	/users/login	로그인
POST	/users/register	회원가입
POST	/users/logout	로그아웃
GET	/users/{userId}	유저의 정보 가져오기
GET	/users/{userId}/mypage	현재 로그인 된 사용자 정보 가져오기
PUT	/users/{userId}/mypage	현재 로그인 된 사용자 정보 저장하기
GET	/api/post	모집글 가져오기
POST	/api/post	모집글 작성하기
DELETE	/api/post	모집글 삭제하기
GET	/api/post/{postId}	모집글 상세 정보 가져오기
PUT	/api/post/{postId}/apply	모집글 신청하기
GET	/api/post/search	맵에 지정한 좌표계 위치에서 재검색하여 모집글 가져오기
DELETE	/api/post/{postId}/kick/{userId}	참여자 강퇴
PUT	/api/post/{postId}/accessed	참여자 참여 수락
PUT	/api/post/{postId}/denied	참여자 참여 거절
PUT	/api/post/{postId}/success-matching	매칭 확정
GET	/api/{alertId}/{userId}	알림 가져오기
DELETE	/api/{alertId}/{userId}	알림 삭제하기



## 왜 socket.io인가?



채팅은 대용량 작업보다는 세밀한 관리를 요하는 기능이라고 판단하여  
더욱 직관적이고 특히 우리 프로젝트에서 중요한 room기능을 사용할 수 있는

**socket.io를 채택**

# 지도 API

지도 생성  
중심좌표 설정  
확대 레벨 설정



카카오 지도 api

좌표 생성  
마커 찍기

현재 위치 요청



현재 위치 제공



IP Geolocation API

IP Geolocation API is a free real-time IP to Geolocation JSON API  
with detailed countries data integration.

Geolocation api

업무 분담

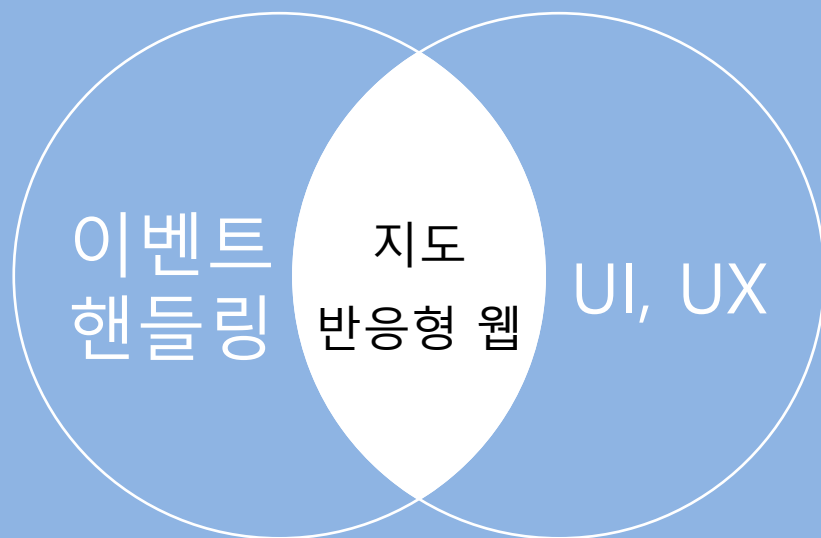


# 업무 분담

## 프론트엔드

우명규

류나연



## 백엔드

전혜지

매칭  
알림

최성민

포스팅  
채팅

선원종

유저  
인증

감사합니다