

2022 SmileGate Winter DevCamp 2th

B-Tree

중간발표

김상수 - Android / Chat server

정진우 - Data Server

신승용 - Board Server



목차

중간 발표 순서

- 팀 소개
- 아키텍쳐
- 팀 빌딩
- 개인 목표
- 개발 계획
- 일정 관리

팀 소개

팀 및 팀원 소개

충청 지역에서 활동하는 'B-TREE' 입니다.
균형잡힌 B-TREE의 B를 전원이 백엔드
기술스택으로 희망을 했기 때문에
Backend로 하기로 하여 backend-tree를
줄여서 B-TREE입니다.

김상수

충북대학교 소프트웨어학과 3학년

다양한 백엔드 기술들을 익히고 실력을 늘리기 위해 참여했습니다

정진우

충남대학교 컴퓨터융합학부 3학년

추후 입사할 회사에 눈도장을 찍기 위해서 참여했습니다

신승용

충북대학교 소프트웨어학과 2학년

토이 프로젝트가 아닌 대형 프로젝트를 경험하고 싶어서 참여했습니다

팀 소개

주제 소개

협업과 실력 성장에 초점을 맞춘



개인의 성장

개인 목표를 모두 달성가능한 주제

구현 가능성

개인의 성장

스토브 플랫폼으로 주제를 정하였지만
기술적으로나 벡앤드 면에서 보아서 성장할 방향이 좋지 않다고
생각되어서 플랫폼 형태를 유지 하되 많은 기능을 구현할 수 있는 당근
마켓 클론코딩으로 주제를 변경함

개인 목표를 모두 달성가능한 주제

팀원 개인의 목표를 달성 가능한 플랫폼 형태의 서비스

구현 가능성

기존에 있는 기능 뿐만 아니라
여러가지 기능을 유동적으로 추가 할 수 있는 잠재력이 있음

당근마켓이란?

중고거래 플랫폼

안 쓰는 물품을 가까운 동네 이웃과 나눔하는 중고거래 플랫폼
'당신 근처에서 만나는 마켓'의 준말

The image shows a mobile application interface for Daangin Market. At the top, there are location filters for '역삼동' (Yeoksam-dong) and '서초동' (Seocho-dong). The search bar contains the text '가게를 찾아보세요' (Search for stores). Below the search bar are several product cards:

- 플로어 스탠드** (Floor Stand) located in Yeoksam 2-dong, Bongeum, priced at 10,000원. Status: 예약중 (On Hold). Likes: 3, Comments: 6.
- 암체어** (Armchair) located in Yeoksam 1-dong, Bongeum, priced at 80,000원.
- 무쇠냄비** (Cast Iron Skillet) located in Gyeopho 2-dong, Bongeum, priced at 100,000원.
- 원룸이사 해요** (Moving service for studio apartment) located in Gangnam-gu, Dodeok 1-dong, regional advertising, priced at 35,000원.

On the right side of the screen, there are category filters: 에어컨 (Air Conditioner), 용달 (Delivery), 이사 (Relocation), 중고차 (Used Car), 카페 (Cafe), 농수산물 (Agricultural products), and 부동산 (Real Estate). A speech bubble says '||를 확인해보세요!' (Check it out!).

기술 스택

사용언어



python



java



kotlin

프레임워크



fastApi



Spring MVC



Spring webflux



Android compose

데이터베이스



Elastic search (ELK Stack)

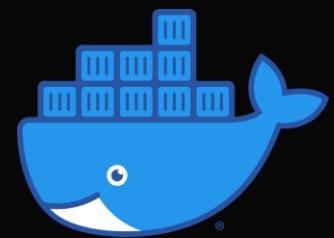


redis

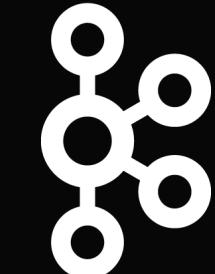


MySQL

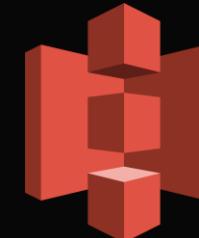
devOps



docker



kafka

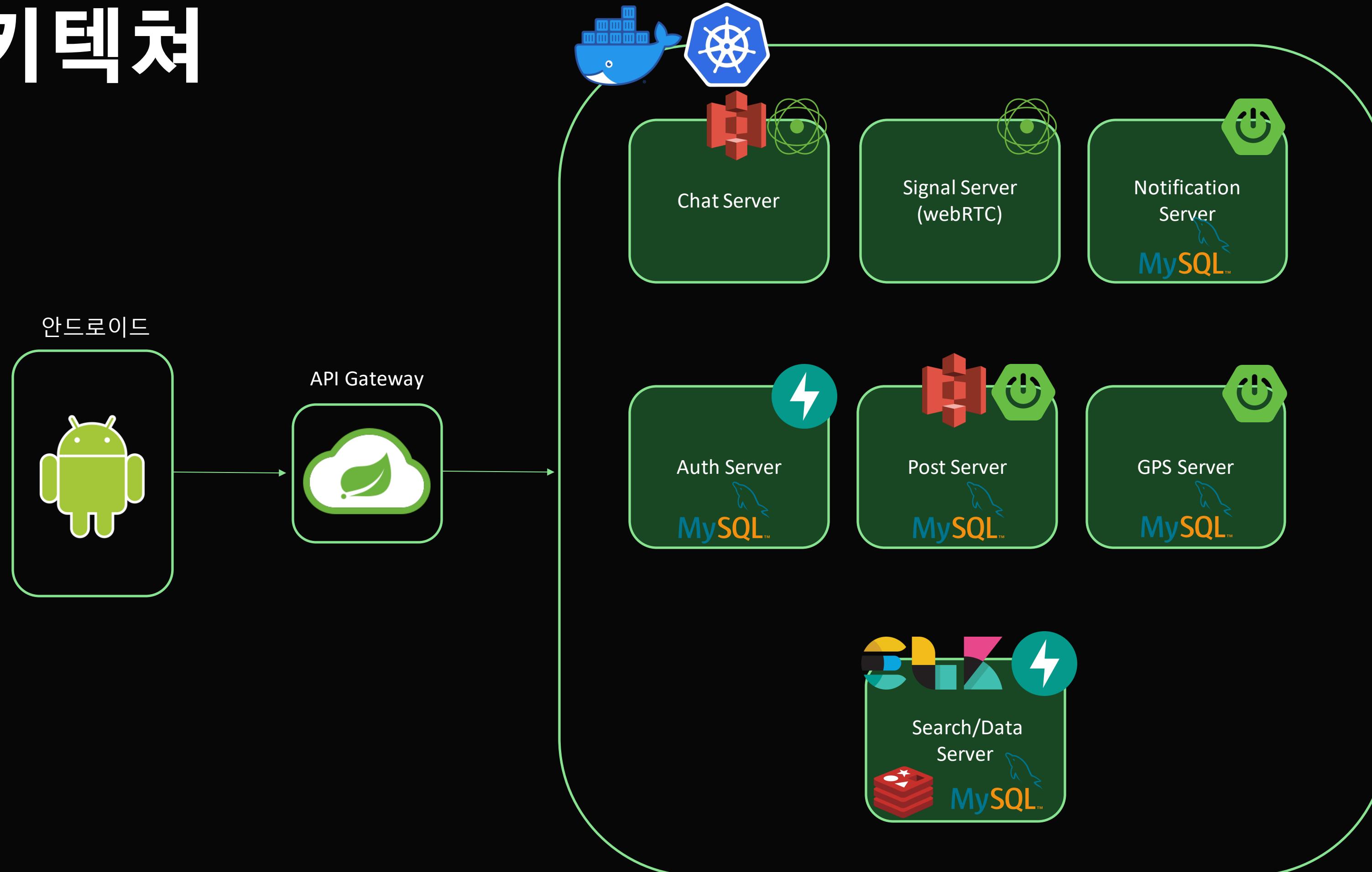


Aws S3



k8s

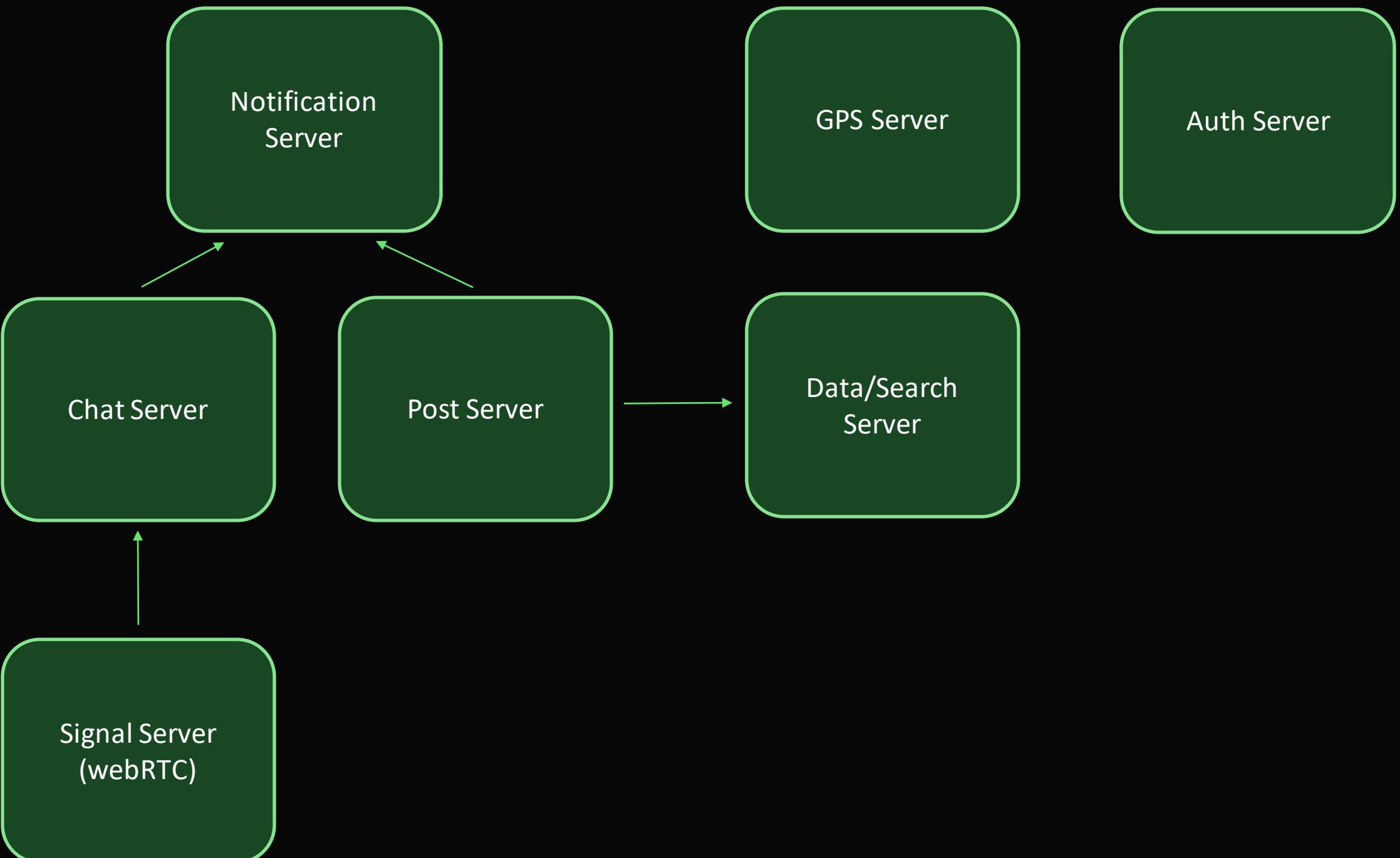
아키텍쳐



MSA 간 통신



메시지 브로커로써
카프카를 이용한
통신



* msa별 각 DB를 가진다.

팀 빌딩

철학의 공유와 예측과 관리가 용이한
팀을 위한 계획

일정의 예측 가능성

각자의 일정을 공유할 수 있는 캘린더를 노션에서 관리
코어 타임을 지정하여 최소한의 개발시간 확보
진행 사항 및 이슈를 코어 타임 시작 할 때 가지는 회의를 통해서
공유를 하는 스크럼 회의 진행
주기적인 프로젝트 테스팅을 통한 개발 지연 최소화 및 유연한 개발 목표

팀의 실력 향상

팀의 성장이 개인의 성장이란 철학 공유
서로의 성장을 위해 개발 내용 문서화를 통한 지식 공유
스크럼 회의에서 이슈에 대해 같이 고민
다같이 정한 그라운드룰을 엄격하게 따르기

팀 빌딩

팀 목표

- 체계적인 관리를 통한 협업 진행
- 효율적인 개발과 추후 참고를 위한 문서 작성
- 스프린트 주기별로 프로젝트 테스트 진행
- 프로젝트 평가에서 1등하기

팀 빌딩

업무 분담

김상수

채팅 및 알림

일반 채팅 서버
음성 채팅 서버
알림 서버
안드로이드

신승용

게시글 및 GPS 기능

홈 (판매글) 및 동네생활
매너 온도 및 감사인사(리뷰)
신고기능
GPS 기능을 통한 사용 가능 영
역 설정
AWS S3와 같은 파일 호스팅
서비스 서버 구현

정진우

검색 및 추천

카테고리 설정 기능
검색 기능
추천 기능
회원가입/로그인/로그아웃
• 프로필 조회
• 나의거래
(판매/구매/관심)

팀 빌딩 workflow

협업하는 방식

스크럼 회의 : 코어 타임 전 짧게 (최대 15분)

- 개인 목표와 팀 목표를 되새기는 시간 가지기(?)
- 오늘 뭐할지 구체적으로 정하기
- 개발하면서 생긴 이슈를 공유하고 같이 해결 방법 도출

코어 타임 : 10:00 ~ 12:00 / 13:00 ~ 15:00

- 12:00 ~ 13:00 점심시간
- 의사소통이 필요한 업무 위주 진행
- 슬랙을 통한 즉각적인 의사소통 진행

스프린트 주기 및 점검 : 10일

- 목표 수정사항 반영이 된 스토리보드, PMP, 아키텍쳐 작성
- 프로젝트 테스팅 진행

협업 툴

- 자료 정리 : Notion
- 커뮤니케이션 툴: Slack
- 버전 관리 Git



팀 빌딩 workflow

GROUND RULE

- 의사 결정 방식
 - 최대한 모든 의견을 맞추기! 안되면 다수결!
- 서기
 - 순서가 정해져 있는 주기적인 방식
- 이슈 해결 방법
 - 스크럼 회의 때 공유 -> 같이 해결 방법 도출 -> 구글링 해도 안되는 경우 익일 캠프장님께 문의
- 불만 해결 방법
 - 스크럼 회의 때 대화로 해결
- 깃 및 깃허브 사용방식
 - 고정된 형식의 깃허브 로그
 - git flow를 통한 브랜치 관리

개인 목표 - 김상수

기능적, 비기능적인 실력 향상

코틀린을 사용한 스프링과
안드로이드에서 시스템과 상황을
고려한 비동기적인 처리를 할 수
있는 수준에 이르기

- 코틀린 스프링 mvc, webflux 학습
- 안드로이드 compose 학습
- 코틀린에서의 다양한 비동기 처리법 문서화
- 스프링과 안드로이드의 시스템 적인 비동기 처리 조사
- 프로젝트에 적용

MVC와 MVVM 을 적용할 때
폴더링과 기능 분할에 대한 습관
만들어 시작하기

- MVC, MVVM 에 대한 조사 및 차이점 비교
- 스프링의 폴더링 전략과 기능 조사
- 안드로이드의 폴더링 전략과 기능 조사
- 프로젝트에 적용하면서 수정 할 점 리스트 작성

개인 목표 - 정진우

AI High performer 가 되기 위한 작은 첫 걸음

- 수집한 데이터를 바탕으로 서비스에 적용할 수 있는 적절한 모델을 빠르게 개발하는 개발자가 되기
 - 추천 시스템에 사용 할 수 있는 알고리즘을 리스트업 해보기
 - 결과물을 도출할 수 있는 모델을 여러개 만들어서 오프라인 테스트를 진행해 여러 평가지표들의 값을 정리해서 서비스에 적합한 모델을 정해보기
- 꾸준한 스크럼 회의를 통해 현업에 가서도 적용할 수 있는 커뮤니케이션 기술을 가진 개발자가 되기.
 - 이유없이 불참하지 않기, 불참한다면 사유를 말하는가?
 - 현재 처한 상황을 명확하게 말했는지? 그렇지 않았다면 무엇이 문제였는지 등, 본인 스스로가 그라운드 룰을 잘 지키고 있는지 자가점검을 진행하고 뿐만 아니라 꾸준하게 팀원들과 의사소통하며 부족한 점이 없는지 확인하기

개인 목표 - 정진우

AI High performer 가 되기 위한 작은 첫 걸음

데이터를 수집하여 분석/가공 할 수 있는 능력을 가진

개발자가 되기

- 1개 서버에 ELK Stack 설치& 구현 확인해보기
- 각 MSA에 Logstash를 설치해서 로그수집하여 kibana로 로그분석해보기
- 데이터를 10000개 크롤링 후 DB에 저장하고 쿼리를 날렸을때 의도적으로 잘작동하는지 확인하기
- RDBMS에서 ES로 10000개 Indexing 해보기
- 사용할 때 DB쿼리와 ES쿼리의 속도를 측정해보기
- 수집한 데이터를 인공지능에 활용할 수 있는 데이터로 가공하기

개인 목표 - 신승용

모바일 기반의 어플리케이션 개발 프로젝트 경험하고 회고해보기

- 게시글 형태의 안드로이드 기반 앱을 2~3개 클론코딩 해보며 안드로이드와 Spring의 전반적인 데이터 흐름을 익힌다.
 - ✓ 클론코딩을 해본 후 코드를 보고 역으로 아키텍처를 스스로 설계하여 흐름을 이해했는지 점검
- 프로젝트 구현 단계일 때 클론코딩을 통해 학습한 앱 개발 구조 및 데이터의 흐름을 통해 실제 코드를 구현한다.
 - ✓ 코드를 구현하면서 웹 어플리케이션과 앱 개발을 비교하여 노션에 문서화 하기
- 프로그램 구현이 끝나면 앱 설계 구조와 실제코드를 보며 자문자답을 해보면서 내가 진짜 이해를 했다고 할 수 있는지 점검한다.
 - ✓ 구현한 모바일 앱 어플리케이션이 어떻게 동작하고 데이터가 흐르는지 노션에 문서화 하여 기록하기.
 - ✓ 다음 앱 개발 프로젝트를 할 때에는 어떤 기능을 위해 어떤 구조를 설계해야 할지 별도의 검색 없이도 인지하는 상태에 도달하기

개발 계획(1월 초, 중순) - 김상수

- ✓ 코틀린을 사용한 스프링 인터넷 강의 시청
 - ✓ 안드로이드 컴포즈 구글 코드랩을 통한 학습
-

- 코틀린에서의 비동기 처리 방법 조사 및 문서화 (ex 코루틴, RxJava)
 - MVC, MVVM 에 대해서 조사 및 차이점 정리
 - 스프링과 안드로이드에서의 시스템적인 비동기 처리 조사
 - 스프링과 안드로이드에서의 폴더링 전략과 기능 조사
 - 안드로이드 UI 개발을 통해 mvvm 패턴 적용
-

개발 계획(1~2월) - 김상수

- 채팅, 알림 서버 mvc 조사한 것을 토대로 구현
 - 코루틴, flux 등의 여러 방식 적용 해보기
-

- 안드로이드에서 webRTC 적용
-

- 안드로이드, 서버 연결
 - 데이터 처리, 네트워크 통신, UI 갱신에 있어서의 비동기 처리 유의하면서 연결
-

- 디자인 패턴 적용 사항에 대해서 불편했던 점 리스트 작성
 - 테스팅을 통해 비동기 처리 방식에 따른 성능 비교
-

개발 계획 - 정진우

Auth, User 시스템 설계

- JWT 기반 로그인 기능
 - Redis 이용 토큰 작업 및 Refresh 토큰 기능 적용
 - USER 데이터 베이스 구현 및 USER CRUD
 - E-mail 인증 회원가입 구현
 - 나의 거래 (판매/구매/관심 목록)
 - 매너온도
-

ES 기반 검색 시스템 설계

- Crawling 데이터 10000개 수집하고 DB 등록
- 로그 수집후 Kibana로 로그분석
- Mysql DB -> ES로 Indexing 구현
- 실시간 데이터 인덱싱
- 검색 알고리즘 설계

개발 계획 - 정진우

ML기반 추천 시스템 설계

- DB에 있는 Data를 AI에 적용하기 위한 데이터 가공
 - 추천 시스템에서의 적합한 알고리즘 리스트업
 - 적합한 추천 알고리즘을 정하고 성능 테스트하기
 - 학습 한 모델을 적용 시킬 추천 서버 구현하기
 - 실제 알고리즘 적용 후 성능 테스트하기
-

시스템 성능 평가 및 개선

- 전체적인 시스템 성능 평가
- Docker Container 형태로 배포해보기

개발 계획 - 신승용

- 모바일 기반 어플리케이션 백엔드 클론코딩

- ✓ 모바일 개발 흐름 감잡기, 2~3개 클론코딩 진행
 - ✓ ~1/20 까지 진행
-

- 판매글 게시판 및 동네생활 게시판 구현하기

- ✓ CRUD를 기반으로 한 게시판 기능 구현, 공유 및 신고 기능 구현
 - ✓ ~1/27 까지 진행
-

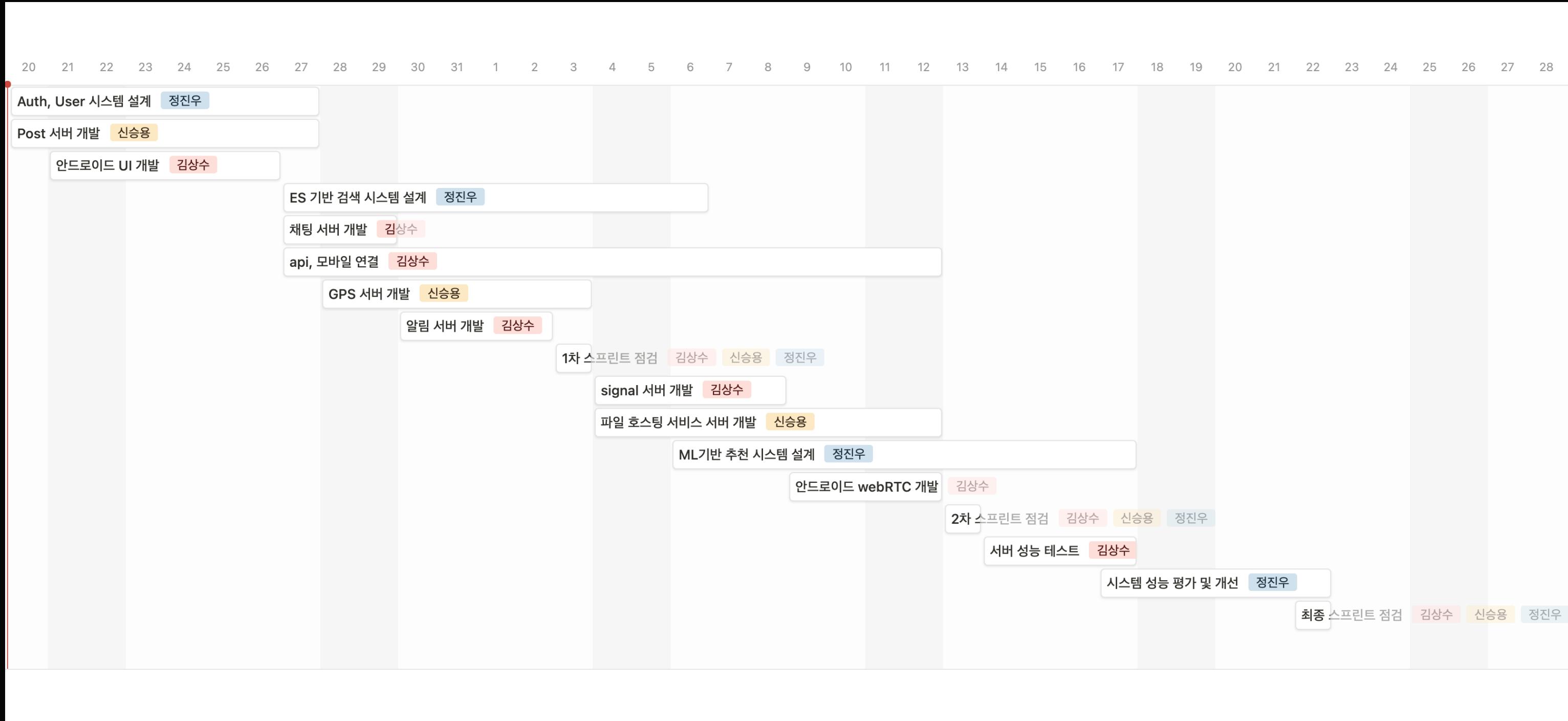
- GPS 기능 구현하기

- ✓ 지역 설정을 통한 게시글 노출 범위 설정
 - ✓ ~2/3 까지 진행
-

- AWS S3와 같은 파일 호스팅 서비스 서버 구현하기

- ✓ 최종 발표 전까지 진행
-

개발 계획 - 간트





감사합니다!

이상으로 B-TREE 였습니다

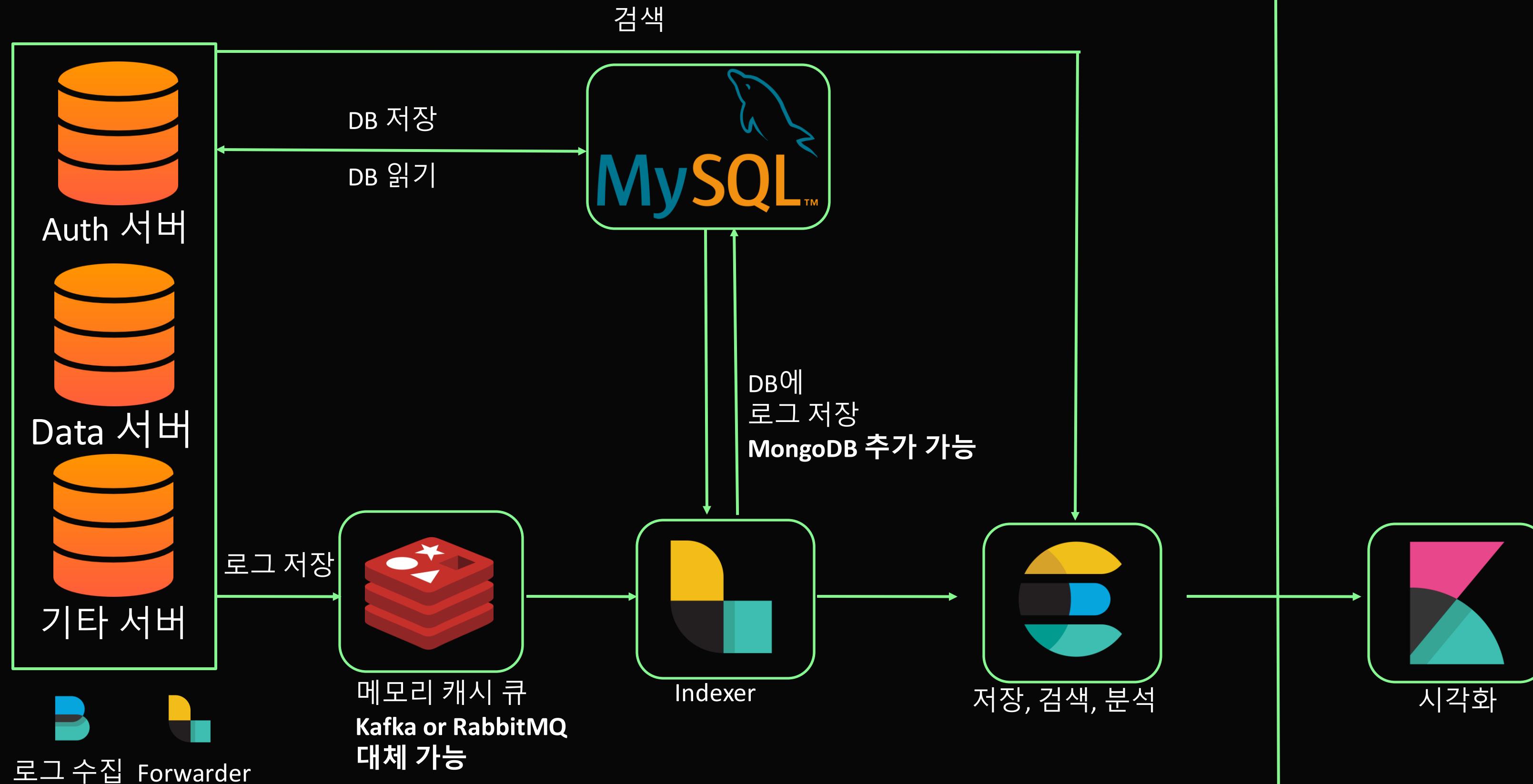
*이후는 부록입니다.

당근마켓

회원가입
이메일
인증
앱 시작
로그인

홈	게시글 목록	게시글 추가	상세 페이지	거래상태 변경	활동 알림	동네 설정	검색	검색 결과	키워드 알림	중고거래 카테고리	카테고리 설정
카테고리	알림	동네생활	게시글 목록	게시글 추가	댓글	활동 알림	댓글	활동 알림	키워드 알림	텍스트	사진
알림	검색	동네생활	알림	검색	나가기	신고	나가기	신고	키워드 알림	위치	약속
채팅	채팅방	알림	입력	전화	활동 알림	나가기	신고	활동 알림	키워드 알림	텍스트	사진
나의 당근	판매 내역	당근 가계부	구매 내역	프로필	나의 당근	동네 설정	알림 키워드 설정	동네 인증	나의 당근	동네 설정	알림 키워드 설정

로그 및 데이터 수집



크롤링 구조

- 웹 당근마켓에서 HTML Tag를 기준으로 데이터 파싱을 진행
- 파싱된 데이터를 데이터 구조에 맞게 변환하여 표현
- 표현된 데이터를 MySQL Board Table에 저장
- MySQL에 저장된 데이터를 Indexer를 이용하여 ES DB에 저장

검색 구조

- 사용자는 키워드(물품/동네)로 검색이 가능함.
- 키워드를 ES에 입력하여 게시글 정보를 받아서 반환

추천 구조

- 사용자는 임의의 게시글을 클릭한다.
- 클릭한 게시글의 정보를 Data Server로 보낸다.
- Data Server에서 게시글의 정보를 특정 기준을 적용하여 유사한 정보들을 추출해낸다.
- 추출한 정보들을 기준으로 Search Server에서 적절한 게시글을 몇 개 검색하여 반환한다.
- 게시글들의 정보를 기반으로 지도 학습 또는 비지도 학습을 한다.
- 추출한 정보들을 기준으로 인공지능 알고리즘을 통해 적절한 게시글을 몇 개 검색하여 반환한다.

깃 및 깃허브 사용방식

Git flow - 5가지 브랜치 전략



```
<type>(<module>): <subject>
<BLANK LINE>
<body>
<BLANK LINE>
<footer>
```

- Types
 - **feat**: A new feature
 - **fix**: A bug fix
 - **docs**: Documentation only changes
 - **style**: Changes that do not affect the meaning of the code. Such as white-space, formatting, missing semi-colons, etc... It also includes changes to JSHint, JSCS rules of the code.
 - **refactor**: A code change that neither fixes a bug nor adds a feature
 - **test**: Adding missing tests. Changing tests.
 - **chore**: Changes to the build process or tools and libraries such as documentation generation

당근마켓 유저 로그인 / 회원가입 절차

The screenshots illustrate the user flow for creating a new account or logging in:

- Step 1: Location Selection**
- Step 2: Phone Number Entry**
- Step 3: Verification Code Input**
- Step 4: Account Creation Confirmation**
- Step 5: Final Activation Confirmation**

Database Diagram:

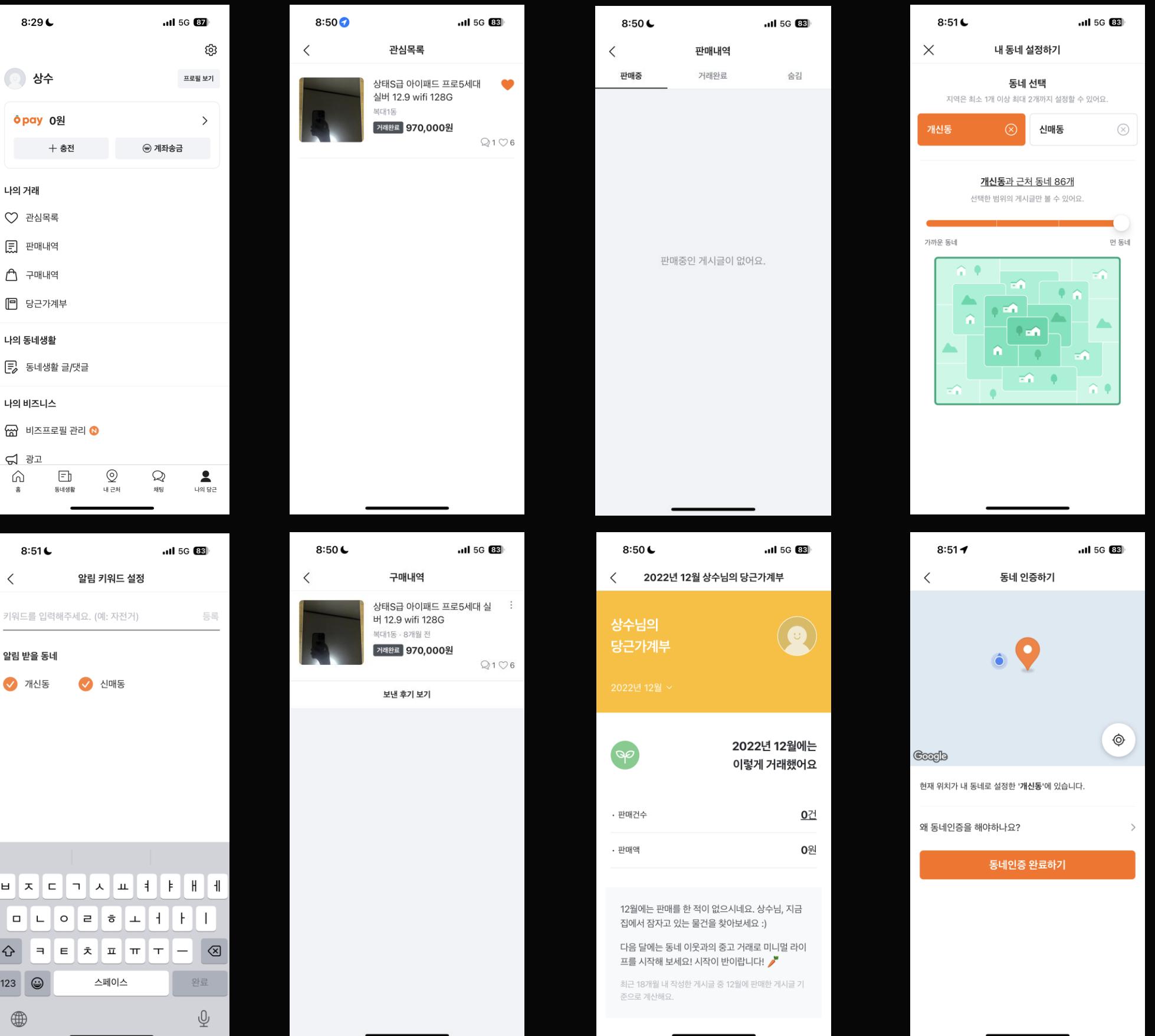
```

    graph LR
        locations[locations] --- Authenticate_1[Authenticate_1]
        locations[locations] --- Authenticate_2[Authenticate_2]
        Authenticate_1 --- UserNumber[UserNumber]
        Authenticate_2 --- UserNumber[UserNumber]
        UserNumber --- user[user]
    
```

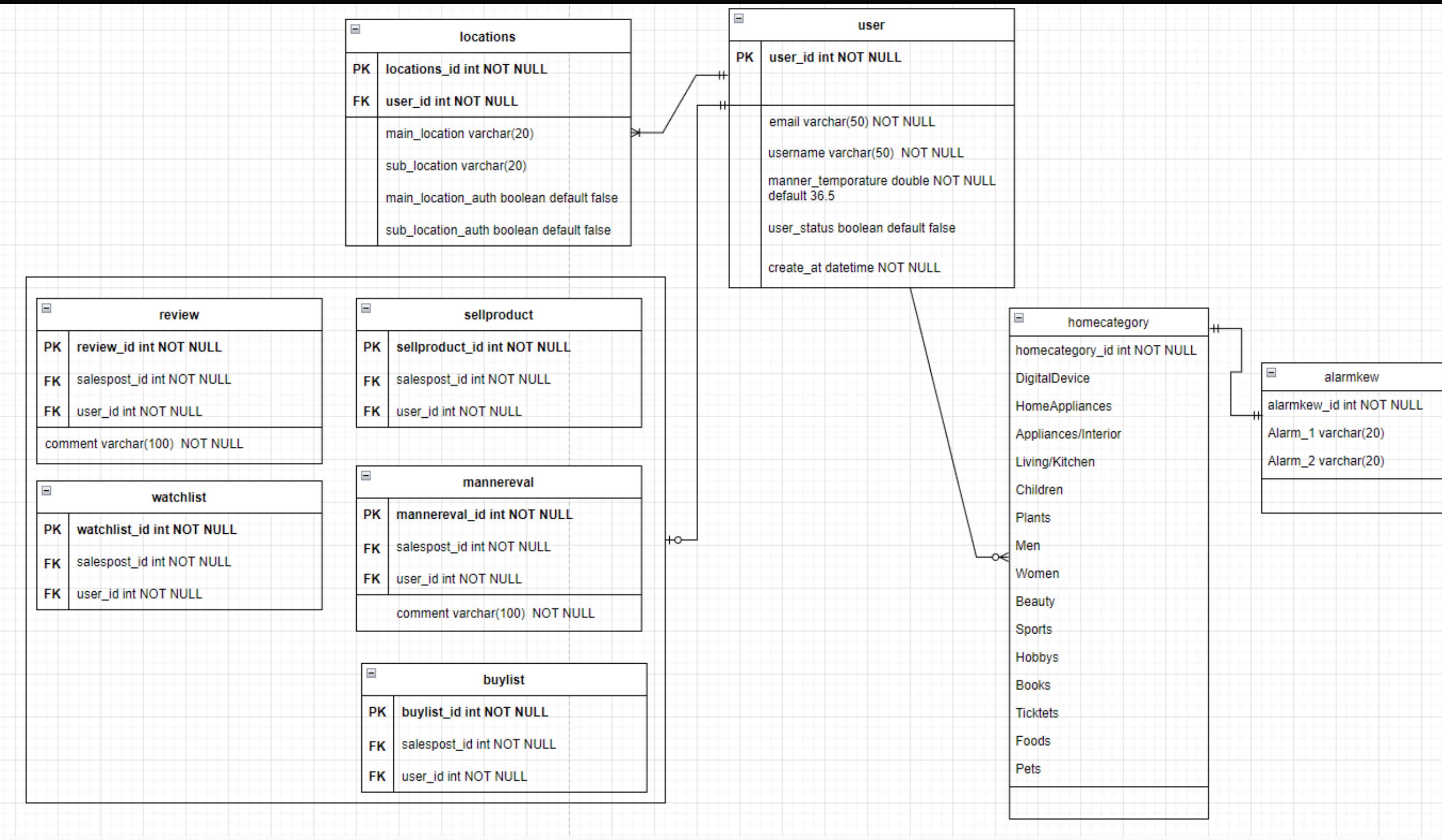
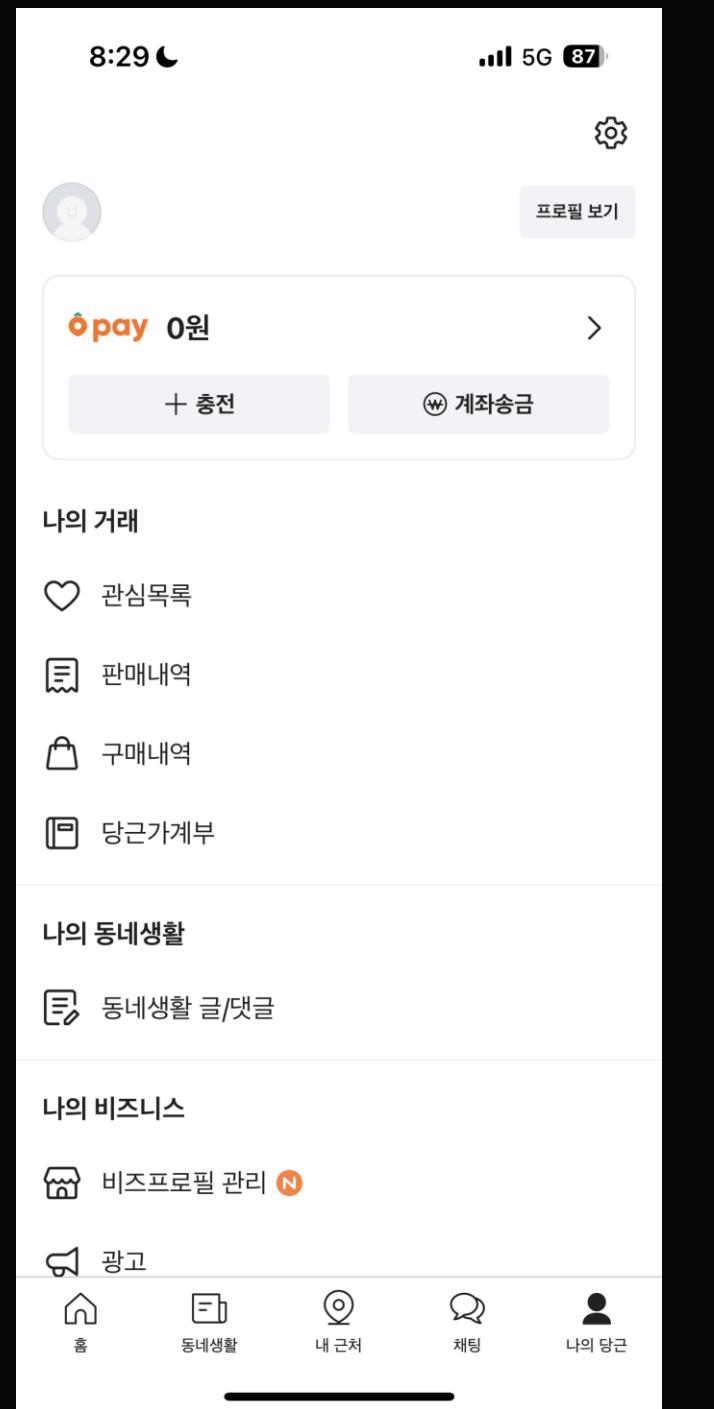
Table Definitions:

- locations** (Left):
 - locations_id
 - Locales_1
 - Locales_2
 - Authenticate_1
 - * Authenticate_2
 - UserNumber
- user** (Right):
 - user_id
 - email
 - nickname
 - mannertemperature
 - locations
 - sellingproduct
 - mannerevaluations
 - transcationreview
 - homecategory
 - alarmkeyword

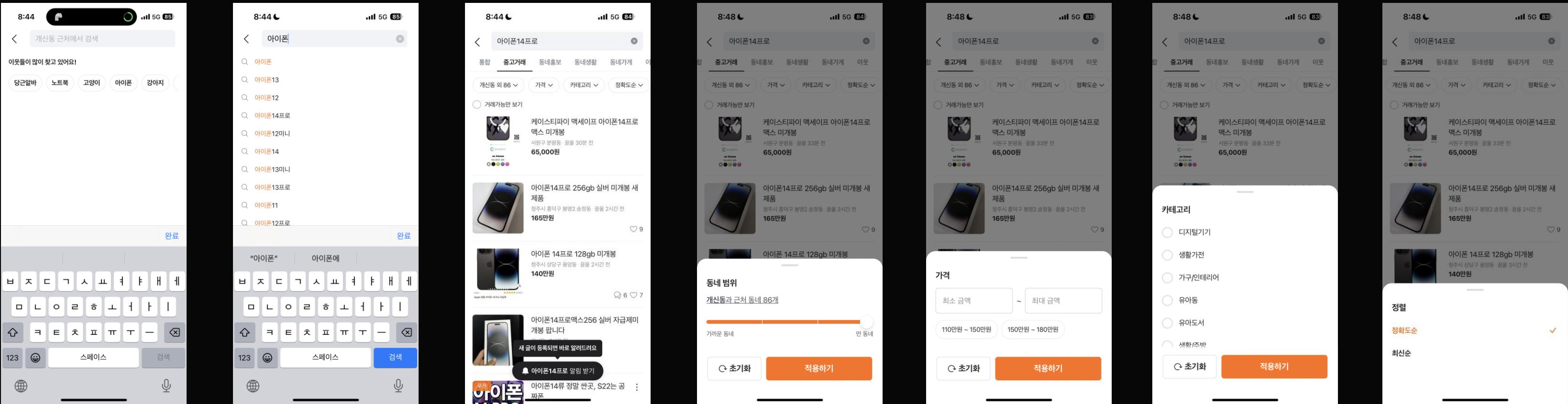
당근마켓 유저 마이페이지



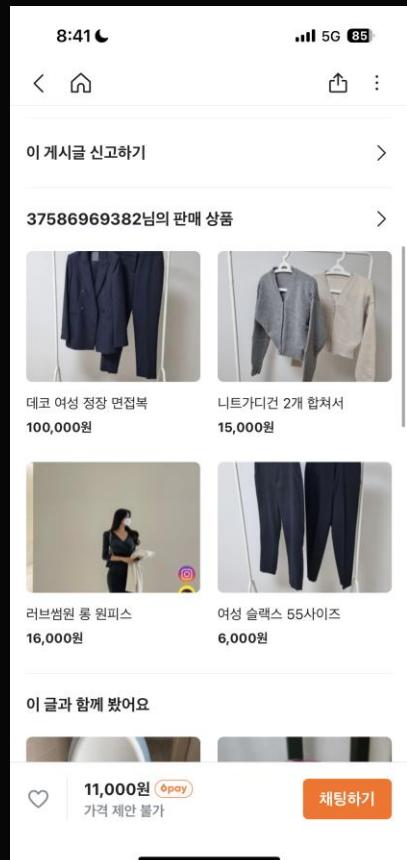
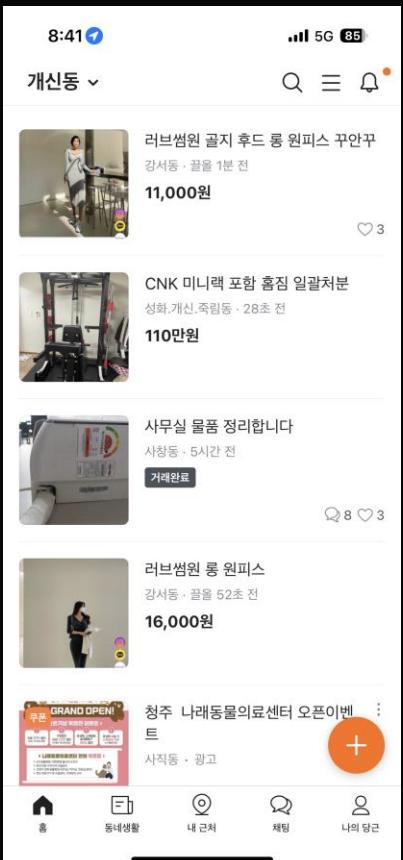
유저 데이터베이스 구조



당근마켓 검색

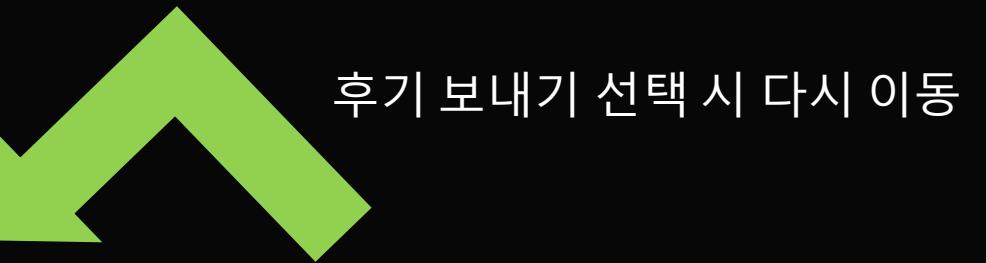


당근마켓 홈(판매글)

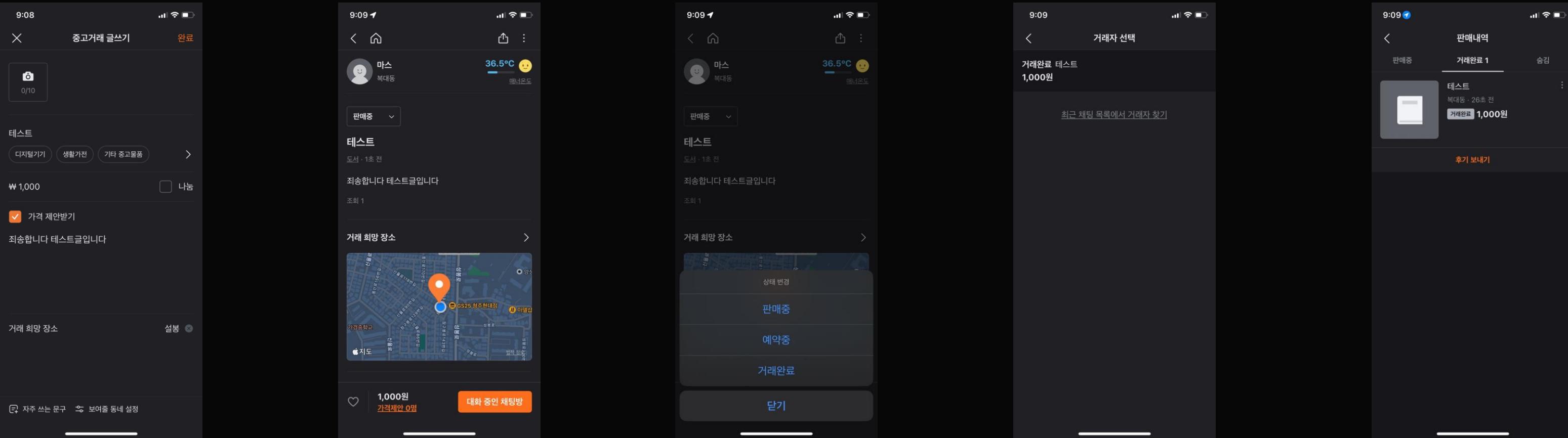


Salespost	
PK	salespost_id int NOT NULL
	title varchar(50) NOT NULL
	content text NOT NULL
	salesimg JSON
	price int
	username varchar(50) NOT NULL
	category varchar(50) NOT NULL
	locate varchar(50) NOT NULL
	update datetime NOT NULL
	likenum int
	chatnum int

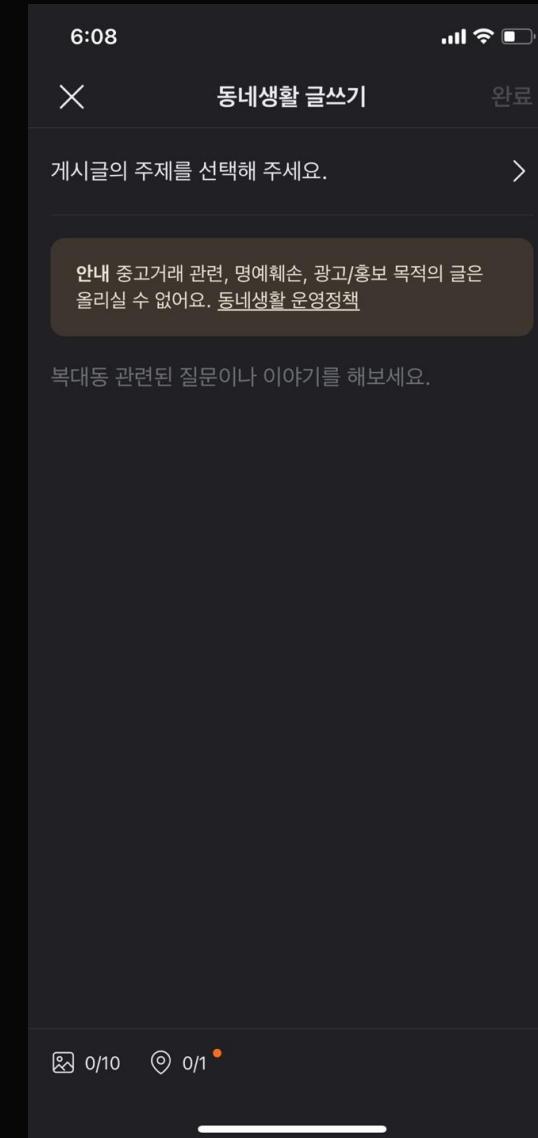
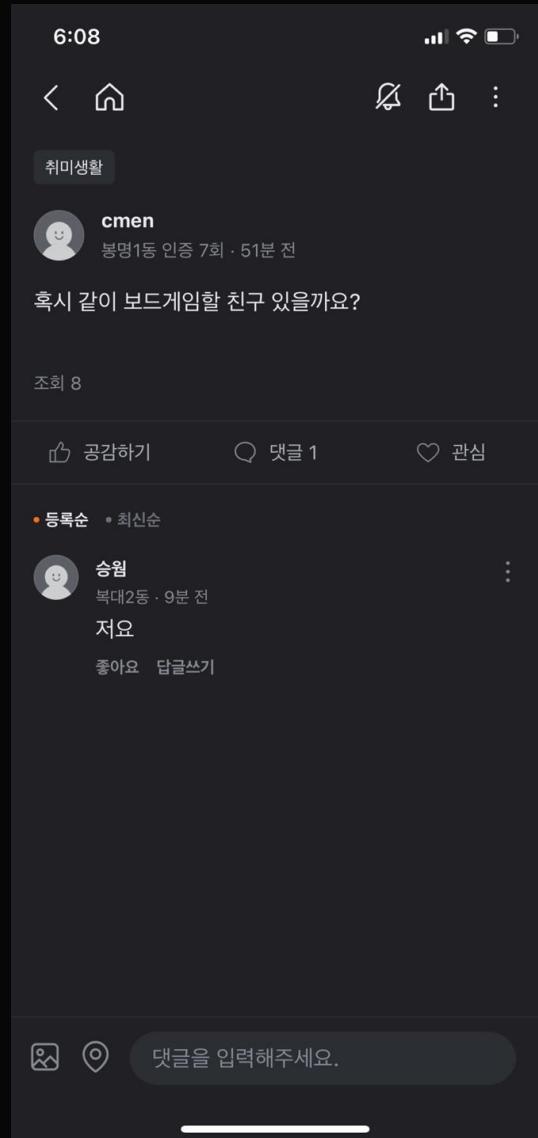
당근마켓 판매 flow



후기 보내기 선택 시 다시 이동



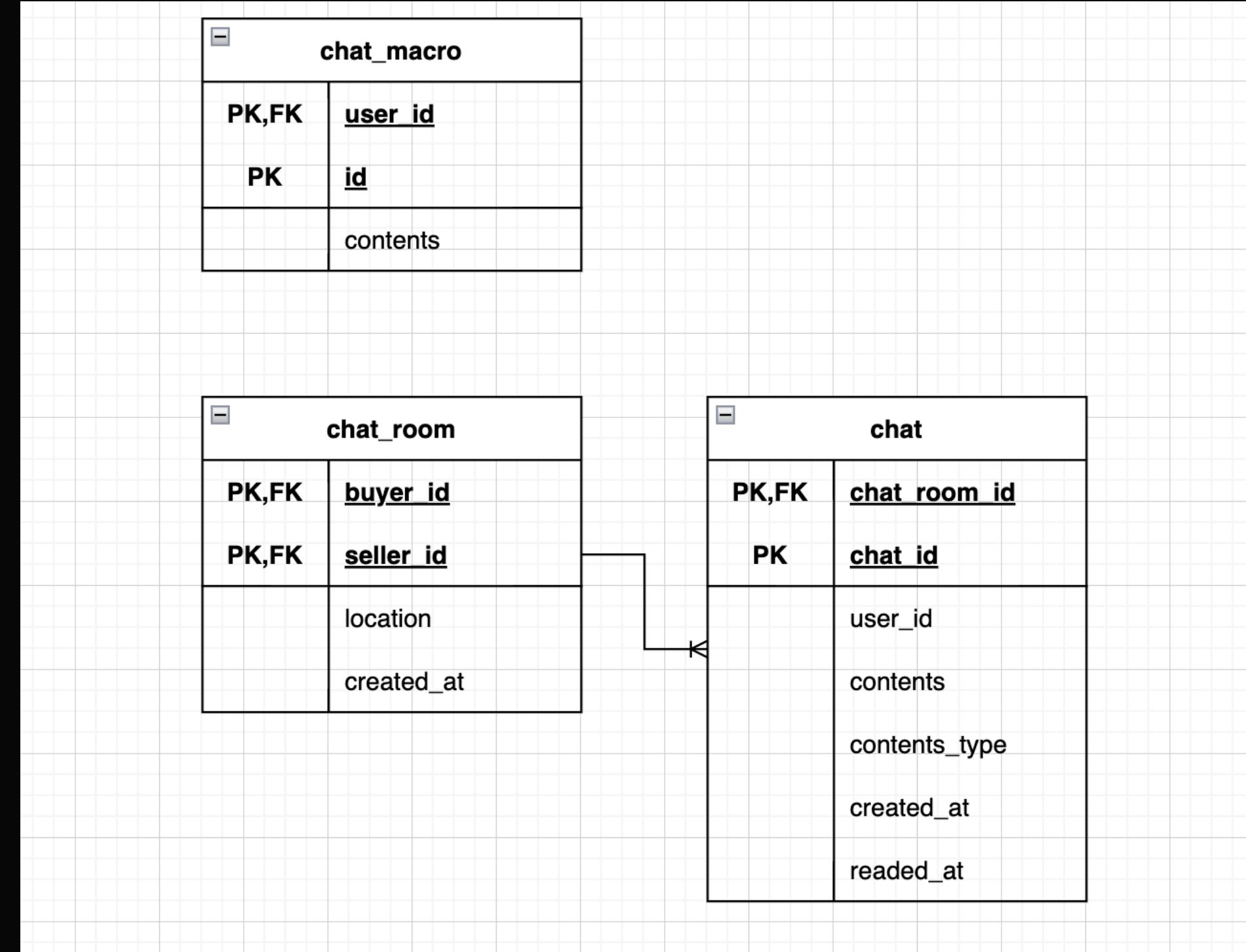
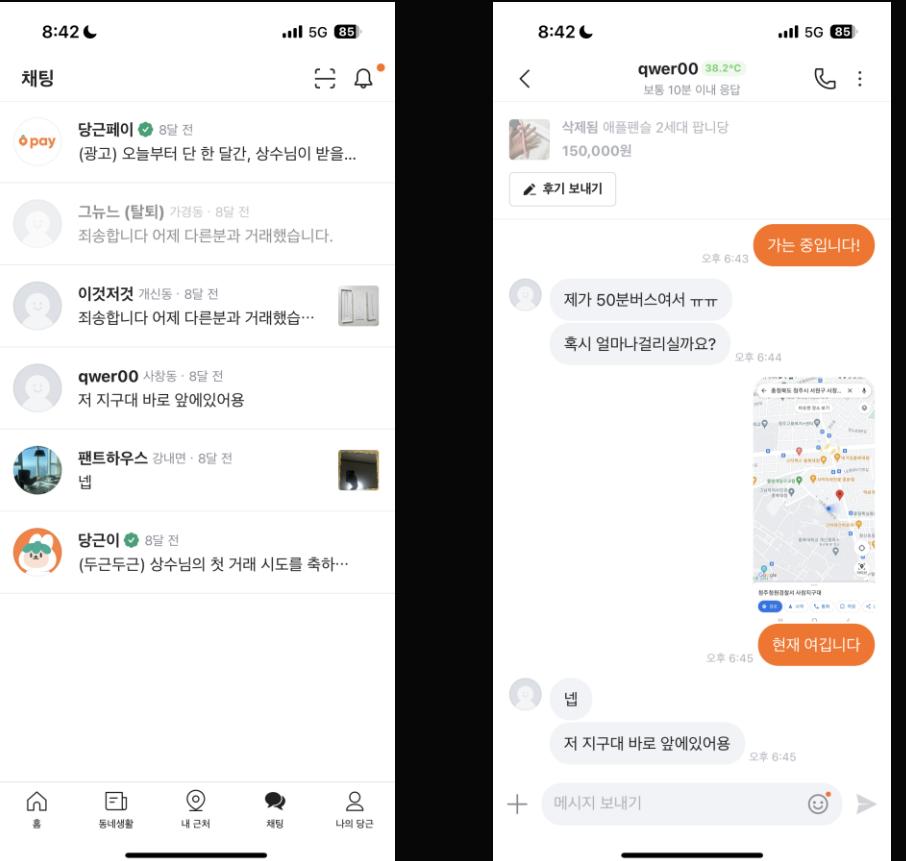
당근마켓 동네생활



PK	boardpost_id int NOT NULL
	content text NOT NULL
	contentimg JSON
	username varchar(50) NOT NULL
	comment_id int NOT NULL
	update datetime NOT NULL
	locate varchar(50) NOT NULL

PK	comment_id int NOT NULL
	content varchar(50) NOT NULL
	boardpost_id int NOT NULL
	username varchar(50) NOT NULL
	date datetime NOT NULL

당근마켓 채팅



당근마켓 알림

