

보유스킬

프로그래밍 언어 및 개발도구

- Kotlin
- Dart
- Java
- Android Studio
- Spring boot

데이터베이스 및 서버

- Firebase
- Room
- SharedPreferences
- MySQL
- JPA

아키텍처 및 디자인 패턴

- MVVM
- Repository Pattern
- Clean Architecture
- Spring MVC

버전 관리 및 협업툴

- PostMan
- Github
- Notion
- Slack

라이브러리 및 프레임워크

- Retrofit2
- Okhttp3
- Glide
- Jetpack Components
[LiveData, Flow, Coroutine, Hilt, ViewModel, Navigation Components]
- Flutter Riverpod [StateNotifier, ProviderScope]

보안 및 인증

- Spring Security
- JWT

프로젝트

개인 프로젝트

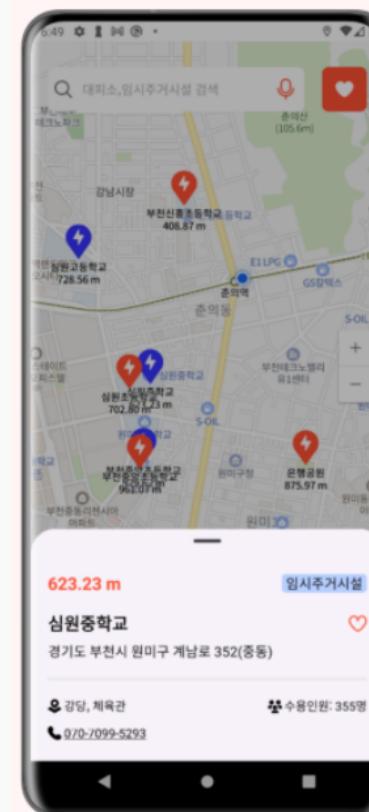
2025.11.11 ~ 11.17



내가 읽은 책을 책장에 저장하고,
책의 페이지수와 리뷰를 기록할 수 있는 앱

[출시완료] 개인 프로젝트

2024.08.16 ~ 10.04
/ 2025.04 ~ 06



재난발생시 주변 대피소 정보를 한눈에 확인할 수 있는
안전지킴이 앱

프로젝트

프로젝트

[출시완료]

팀 프로젝트

2024.05.27 ~ 07.05



지역별 특산물을 활용한
요리레시피 제공 앱

팀 프로젝트

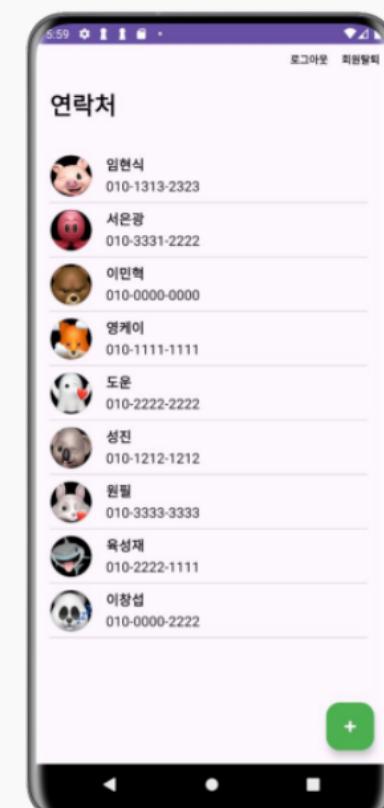
2024.05.13 ~ 05.23



카테고리별로 모아보는 나만의
MiniYoutube 앱

개인 프로젝트

2025.01.03 ~ 03.05



Spring Boot
REST API 서버연동을 통한
간단한 연락처 앱

프로젝트 소개



손쉽게 기록하는 독서기록 앱

개발기간	2025. 11. 11 ~ 11. 17
개발인원	개인 프로젝트



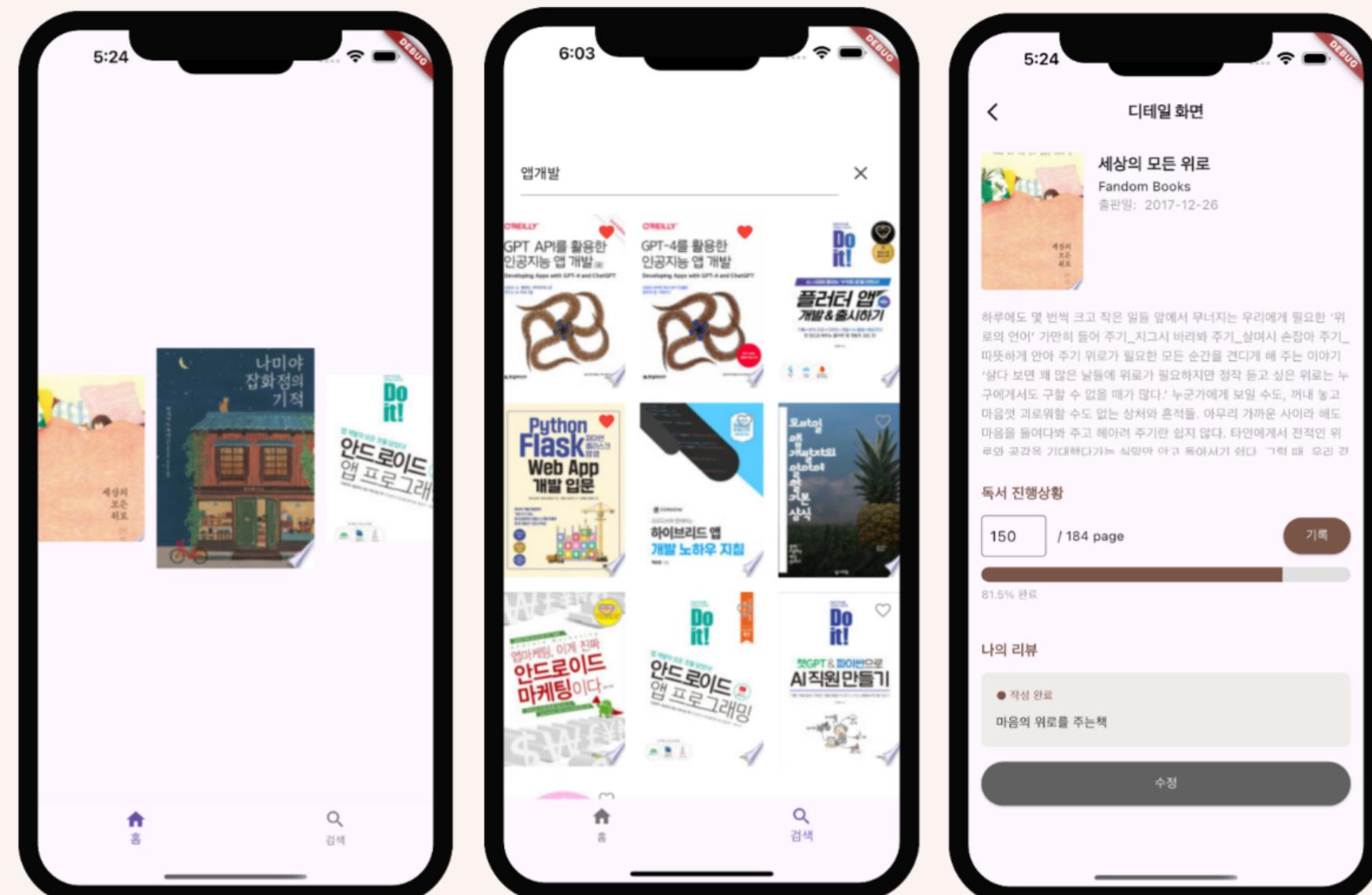
[깃허브 링크](#)

사용기술

언어	Dart
디자인패턴	MVVM + Repository Pattern
상태관리 및 라이브러리	Riverpod , google-books api, carousel_slider 라이브러리
데이터베이스 및 서버	Firebase - Realtime database

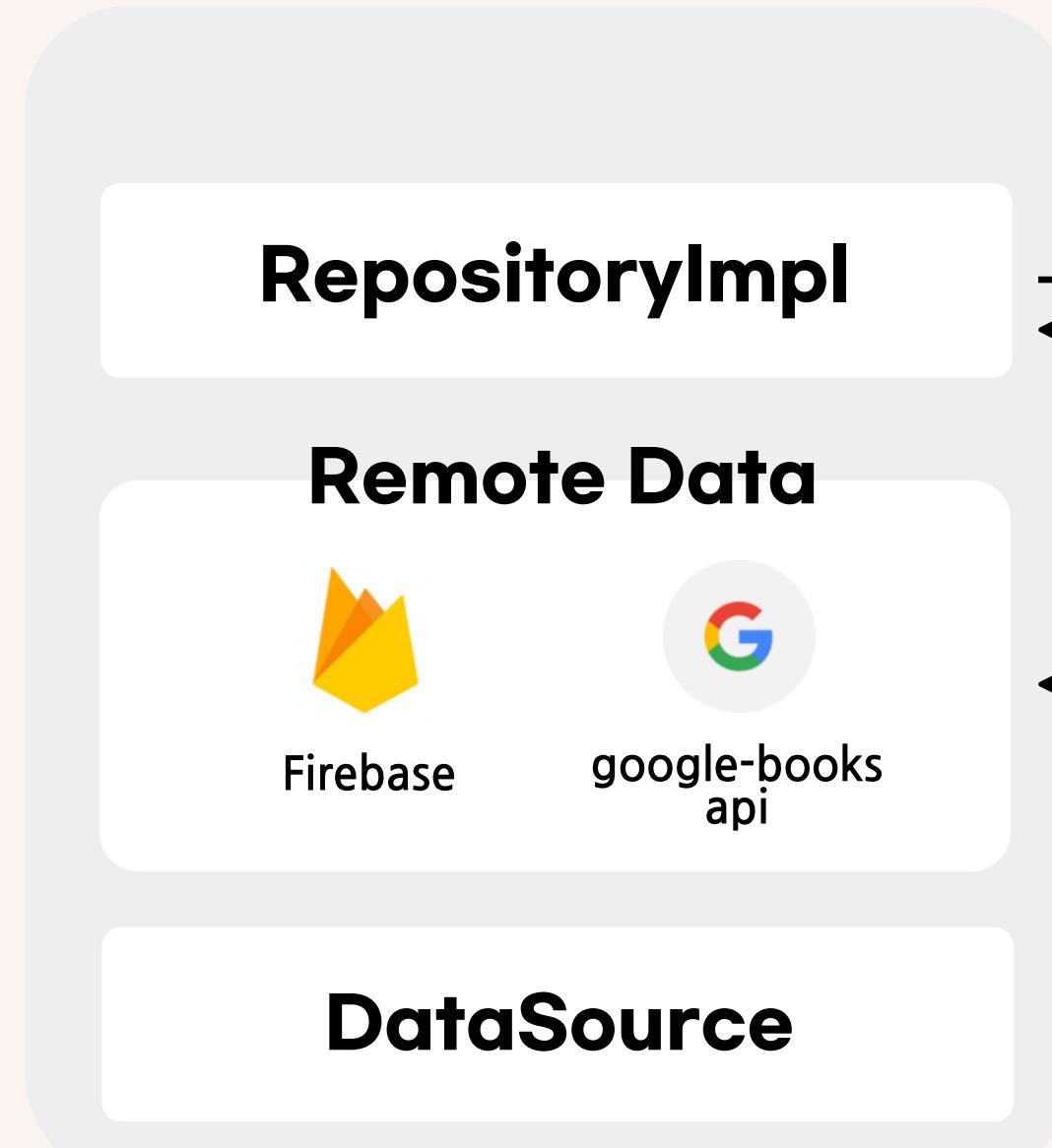
주요기능

책장에 좋아요한 책 저장, 책검색 기능, 책 세부정보 확인,
독서 페이지 기록 & 리뷰 기록

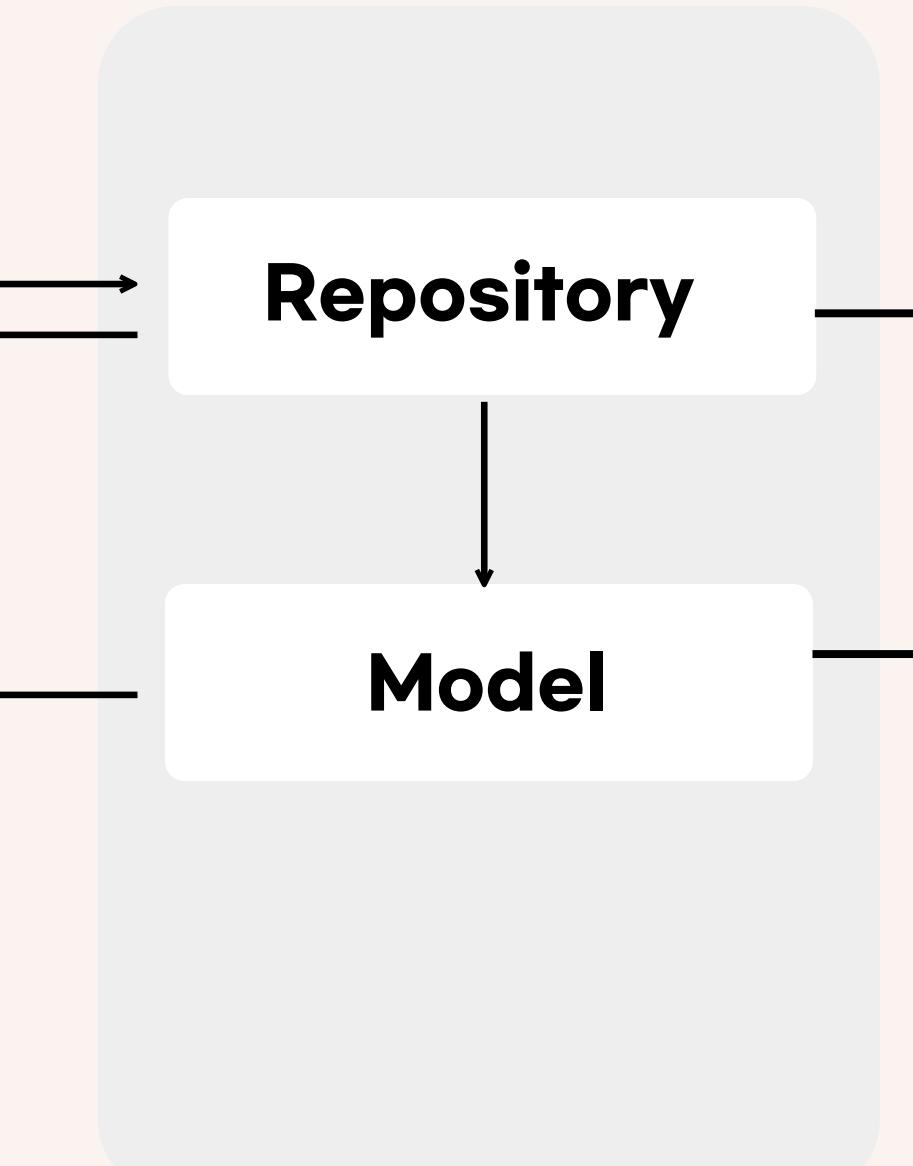




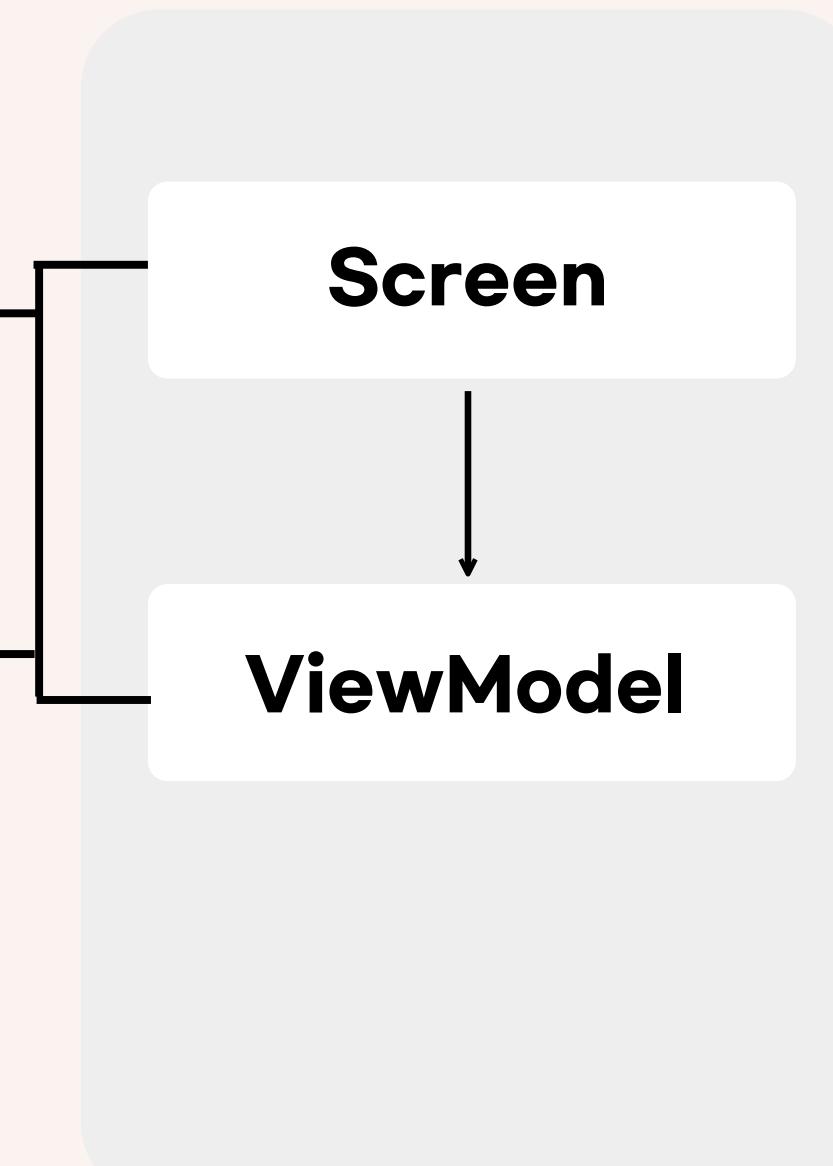
Data Layer



Domain Layer



View Layer



MVVM + Repository Pattern



앱 주요기능 (1/3) - 책장 화면

좋아요를 눌러, 읽은 책을 책장에 담을 수 있습니다.
책장의 책을 클릭하면 바로 상세화면으로 이동이 가능합니다.



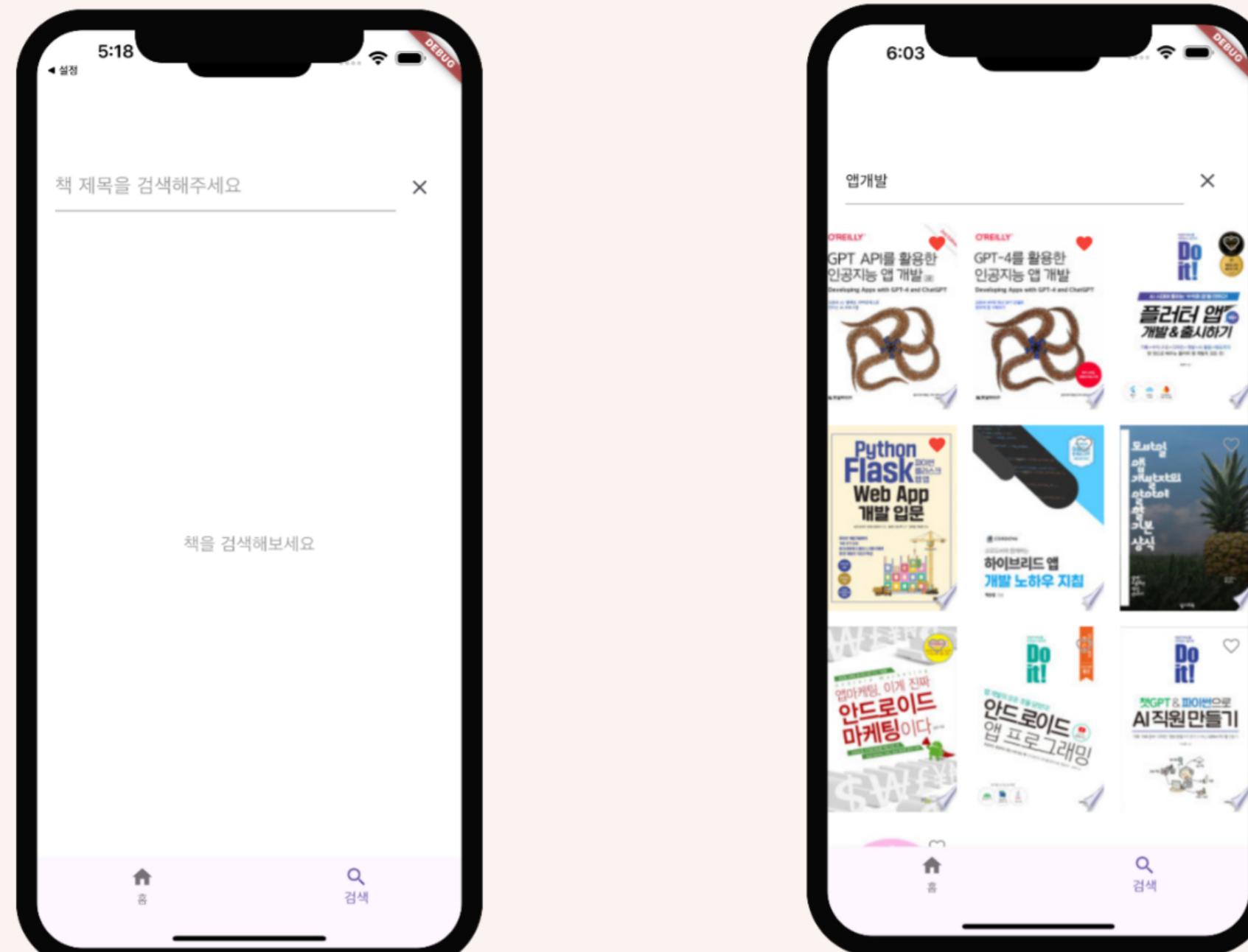
책장에 추가한 책이 없을때



추가한 책이 있을때

readway | 앱 주요기능 (2/3) - 검색 화면

다양한 책을 검색이 가능하고, 클릭시 상세화면으로 넘어갈 수 있습니다.
좋아요를 누르면 책장에 책이 담깁니다.



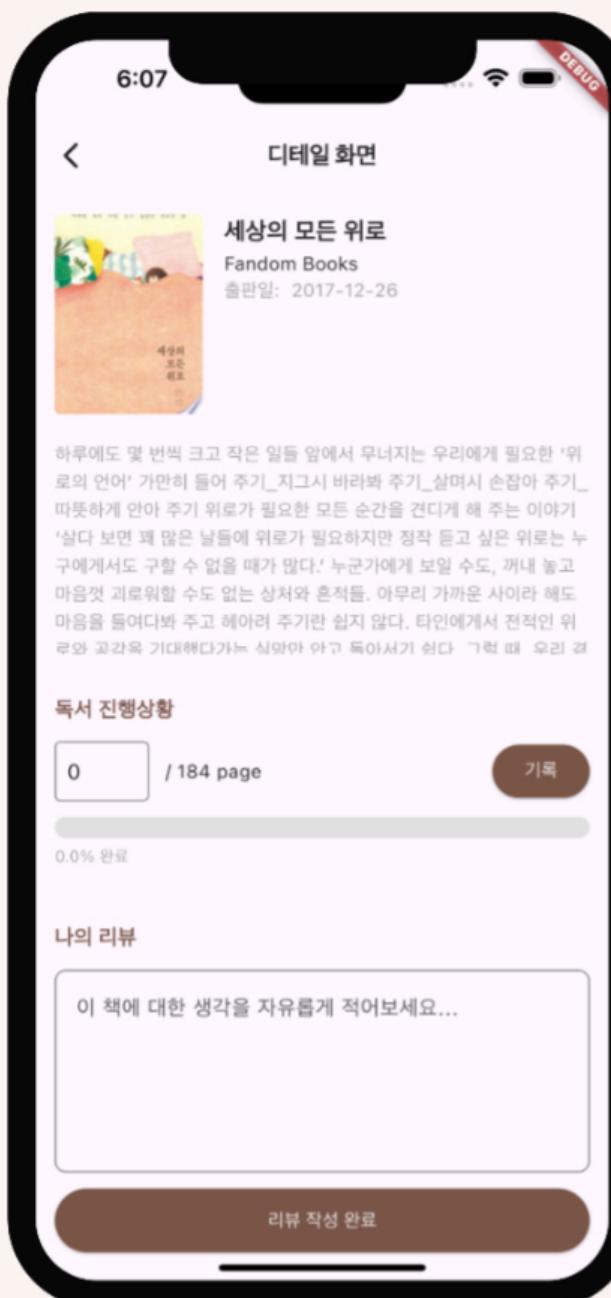
검색 화면



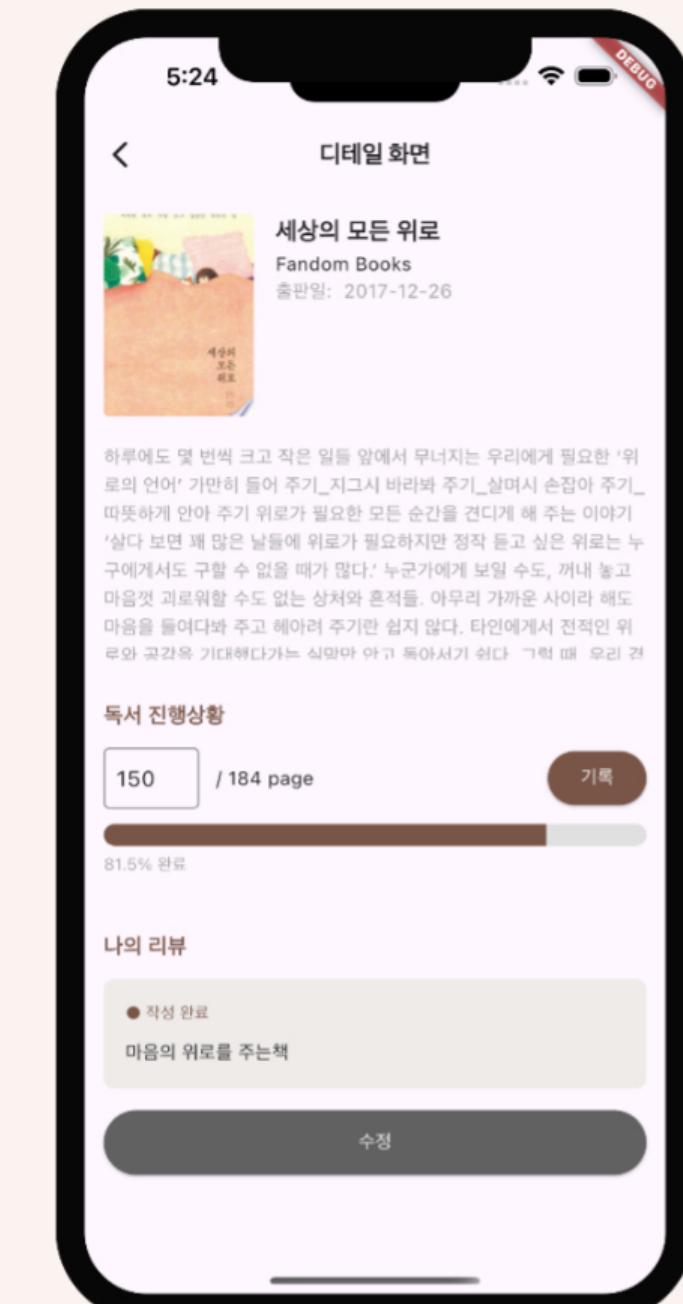
readway

앱 주요기능 (3/3) - 상세 화면 / 다국어 지원

상세화면에서는 페이지를 기록하고, 리뷰를 기록&수정이 가능합니다
언어는 한국어와 영어를 지원합니다.



기록하지 않았을때



페이지 & 리뷰 기록했을때



지키미

재난발생시 주변 대피소정보를 한눈에 확인할 수 있는 안전지킴이 앱

개발기간	2024.08.16 ~ 10.04 / 2025.04 ~ 진행중
개발인원	개인 프로젝트

[플레이스토어 링크](#)

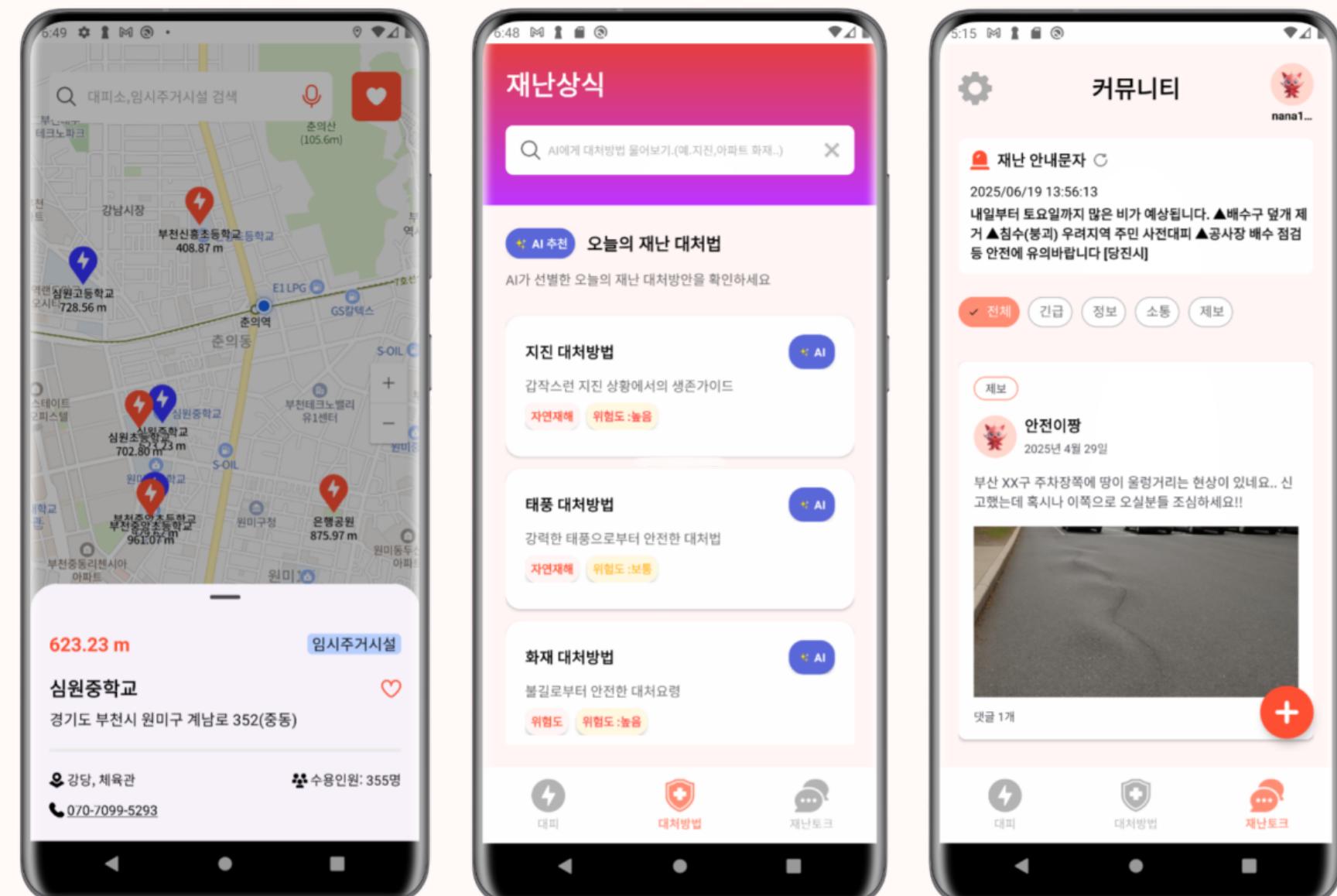
[깃허브 링크](#)

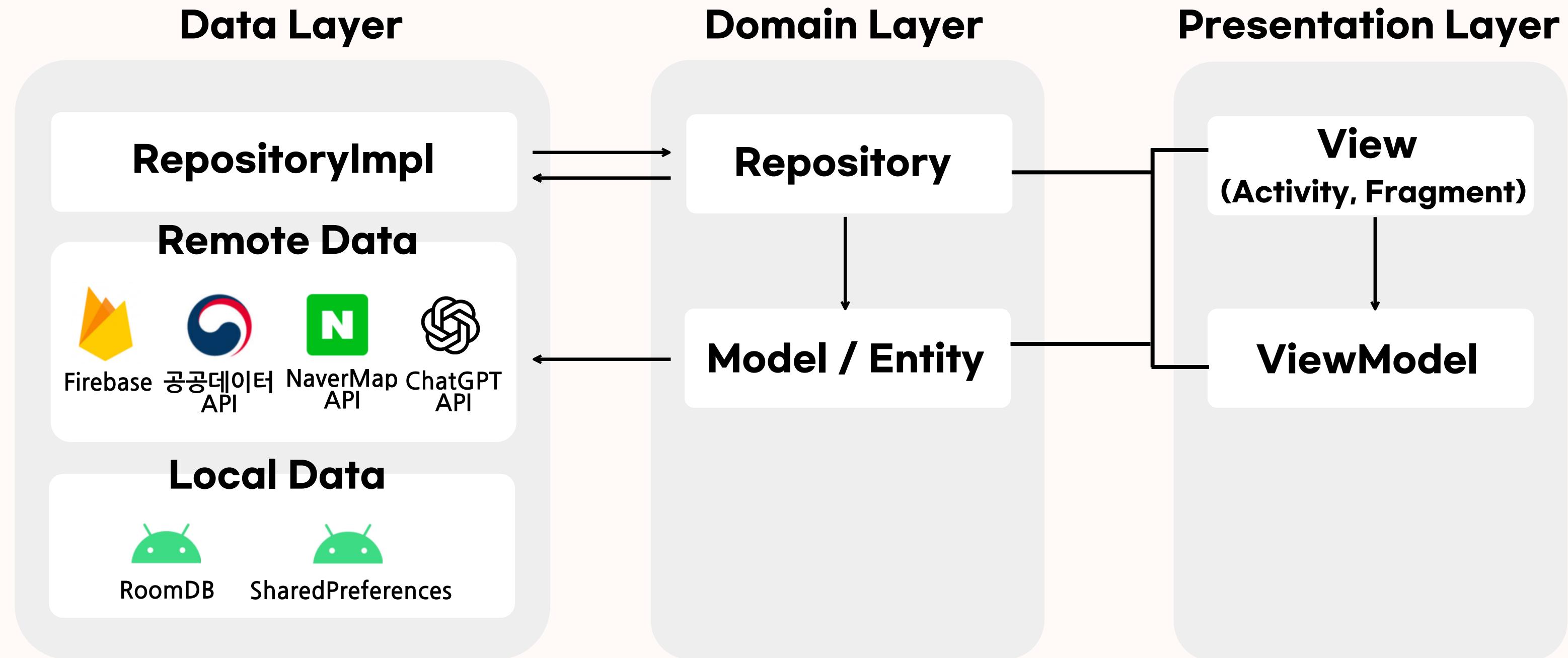
사용기술

언어	Kotlin
디자인패턴	MVVM + Repository Pattern
라이브러리 및 프레임워크	Retrofit2, Okhttp3, Glide, Flow, Coroutine, Hilt, Diffutil, NaverMap, XML, Navigation, Material3 Design
데이터베이스 및 서버	Firebase, RoomDB, SharedPreferences

현재위치 반경 5km 이내의 대피소 정보확인 / 좋아요를 통한 대피소 저장 / 텍스트&음성검색 기능/ 재난 대처법 제공 / 재난 커뮤니티 기능 / 로그인&회원가입

주요기능





MVVM + Repository Pattern



지키미

앱 주요기능 (1/5) - 로그인 / 회원가입

회원가입/로그인을 통해 지키미의 모든 기능을 사용할 수 있습니다.
이용약관에 모두 동의해야만 회원가입이 완료됩니다.

The screenshots show the following steps:

- 회원가입 (Sign Up):** Shows fields for Email (jikimi@naver.com), Nickname (지키미), Password (*****), and Password Confirmation (*****). A red error message "비밀번호가 일치하지 않습니다" (Passwords do not match) is displayed below the confirmation field. Below the form are two checkboxes: "이용약관 동의 [필수]" (Agree to Terms of Service [Required]) and "위치서비스 이용약관 동의 [필수]" (Agree to Location Service Terms of Service [Required]).
- 회원가입 (Sign Up):** Similar to the first screenshot, but the password confirmation field is empty. A small note "위치서비스 이용약관에 동의해주세요" (Agree to Location Service Terms of Service) appears above the "회원가입" button.
- 로그인 (Login):** Shows fields for Email and Password. Both fields have red error messages: "이메일을 입력하세요" (Please enter an email) and "비밀번호를 입력하세요" (Please enter a password). Below the form is a note "계정이 없으신가요? 회원가입" (Don't you have an account? Sign up).
- 로그인 (Login):** Similar to the third screenshot, but the password field is empty. A note "계정이 없으신가요? 회원가입" (Don't you have an account? Sign up) is also present.

회원가입 화면

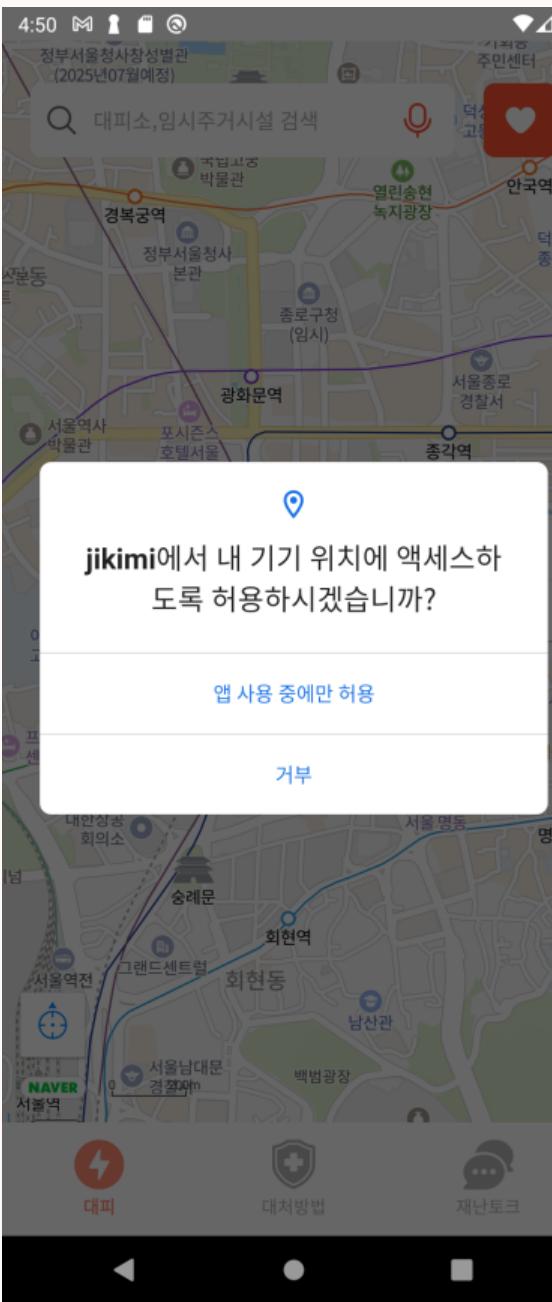
로그인 화면



지키미

앱 주요기능 (2/5) - 현재위치 반경 5km이내의 대피소 정보확인

사용자가 위치정보 접근에 동의하면, 현재 위치 반경 5km이내의 대피소(임시주거시설,야외대피장소)를 모두 가져와 보여줍니다.
대피소 마커를 클릭하면 거리,주소,수용인원 등 대피소의 상세정보를 쉽게 확인 할 수 있습니다.

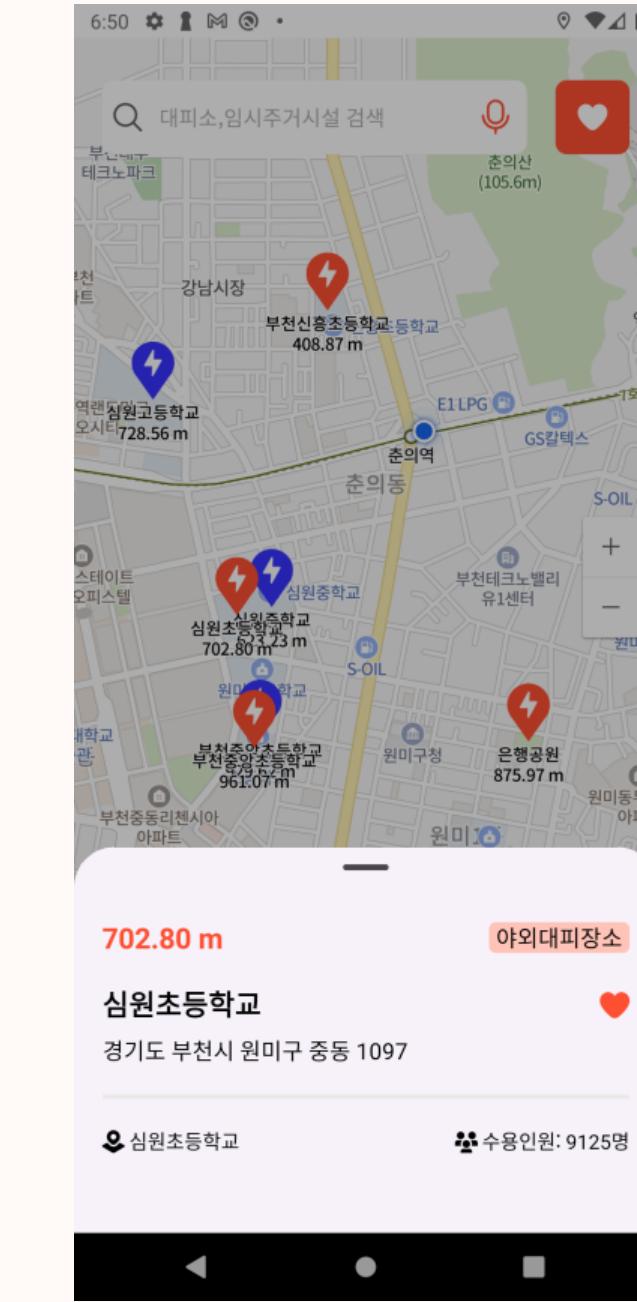


위치정보 접근 허용화면



전화번호 클릭시
전화앱으로 이동 →

임시주거시설,야외대피장소의 상세정보 확인화면





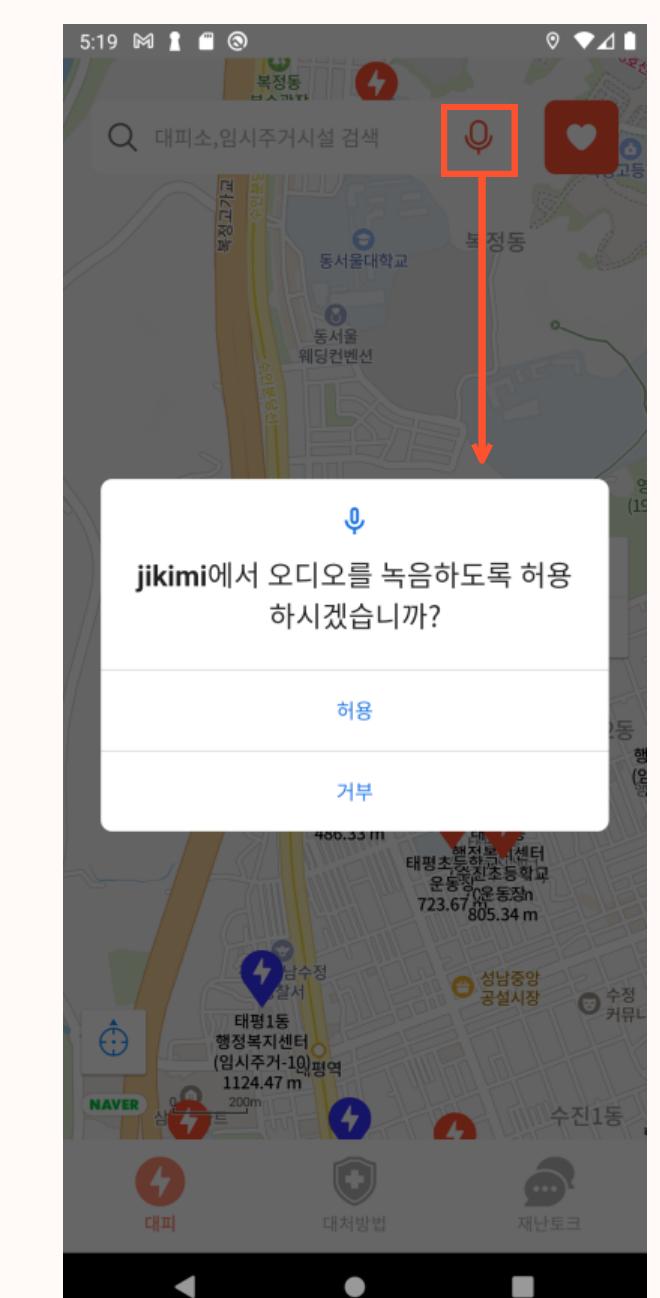
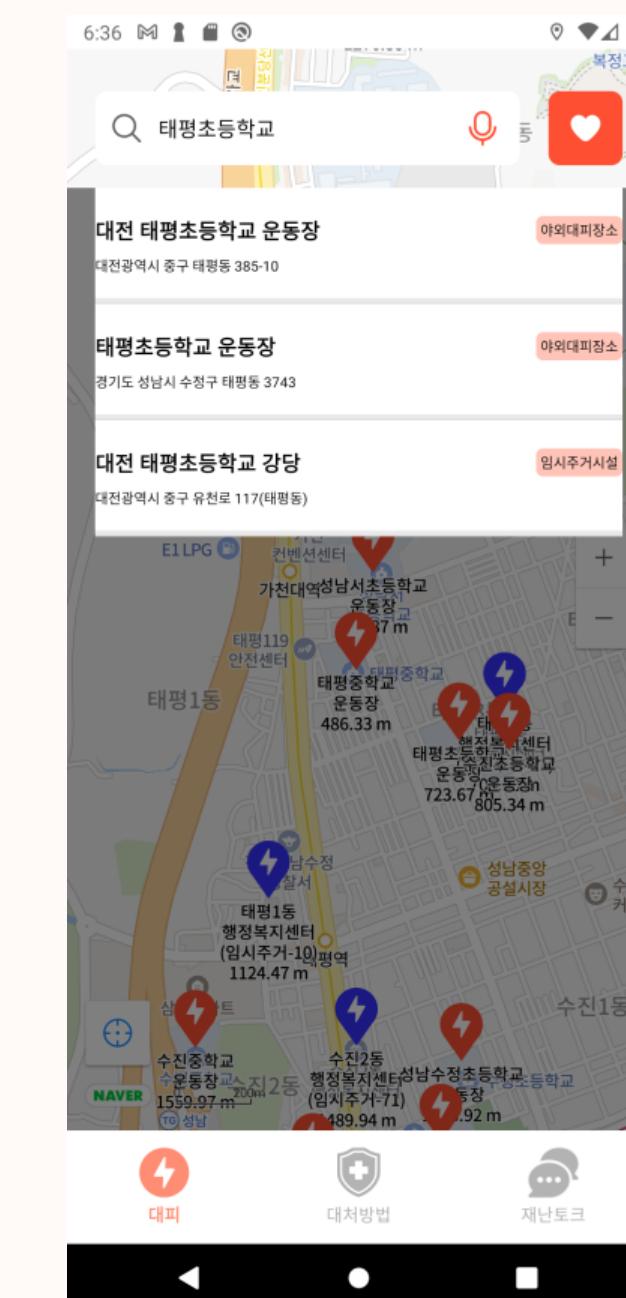
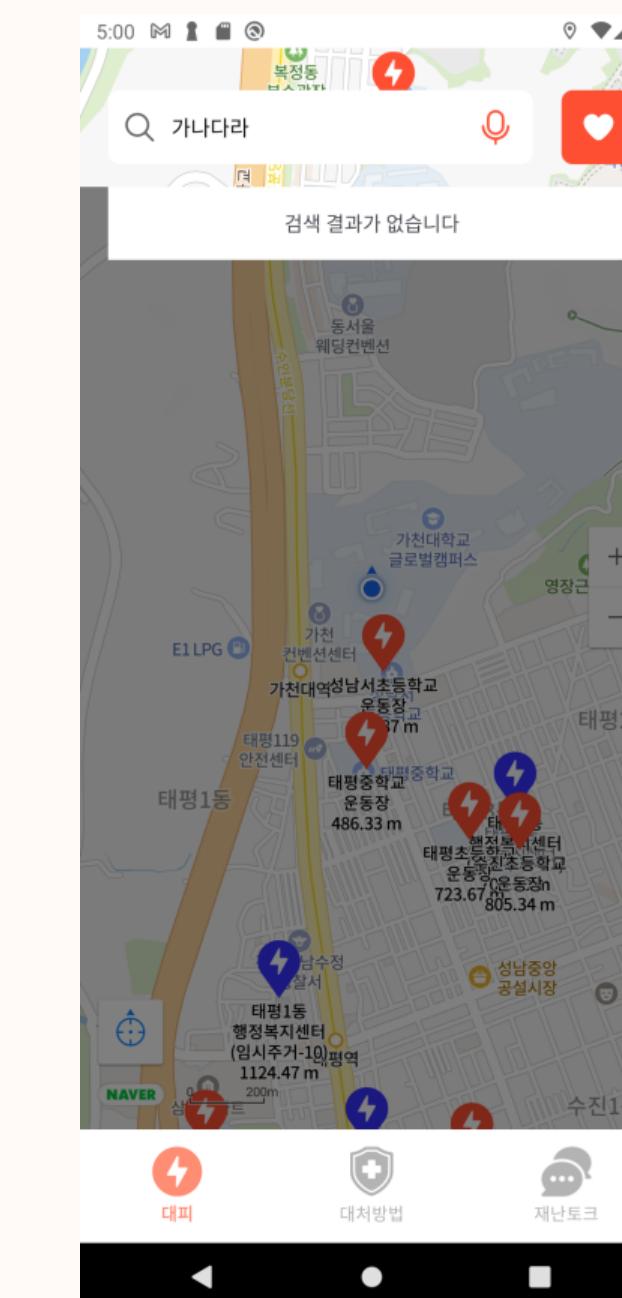
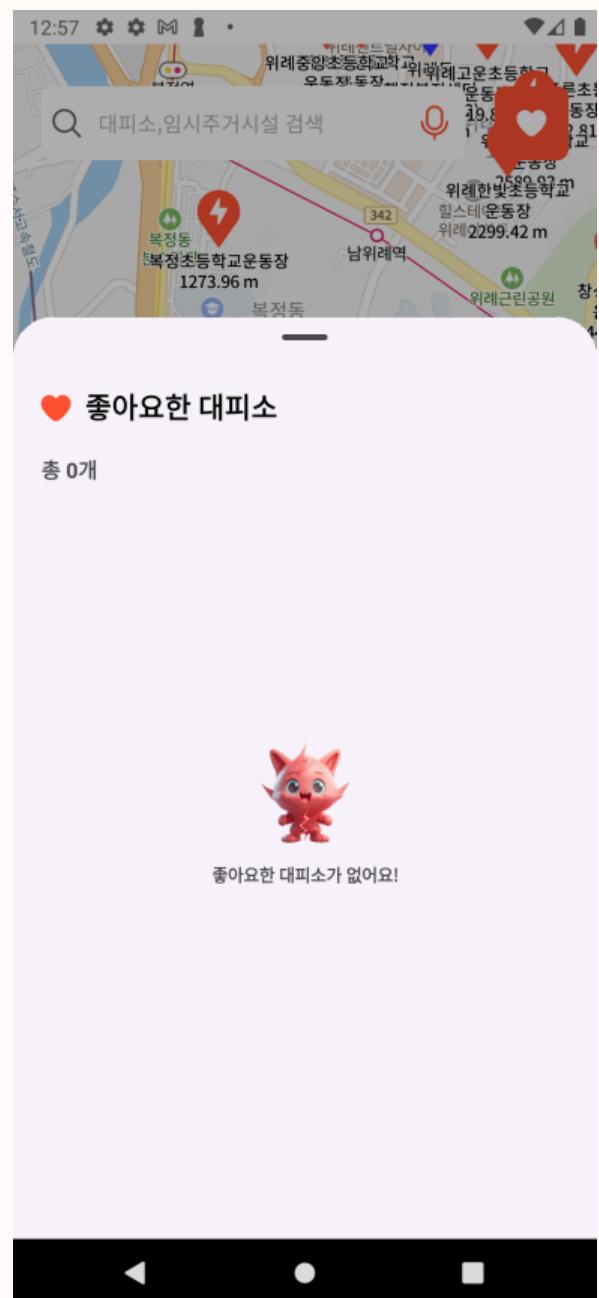
지키미

앱 주요기능 (3/5) - 좋아요를 통한 대피소 저장 / 텍스트 & 음성검색 기능

대피소 상세화면에서 원하는 대피소를 좋아요를 통해 저장할 수 있습니다.

좋아요한 대피소 클릭시 해당 대피소 위치로 이동 / 롱클릭시 좋아요한 대피소를 삭제할수있습니다.

또한 대피장소의 이름을 통해 검색할 수 있습니다. 마이크 버튼을 눌러 음성검색도 가능합니다.



저장한 대피소가 없을때

저장한 대피소가 있을때

검색결과가 없을때

검색결과가 있을때

음성녹음 허용화면

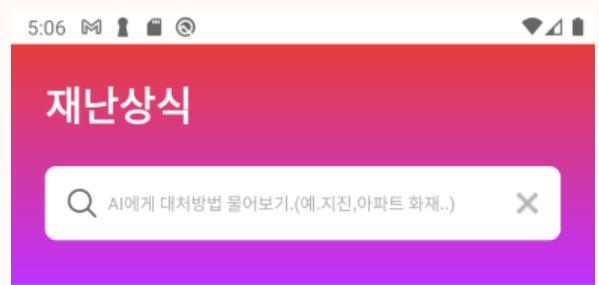


지키미

앱 주요기능 (4/5) - AI가 알려주는 재난 대처방법 학습

AI가 재난 대처법을 랜덤으로 추천해줍니다.

재난 키워드 검색을 통해, 원하는 재난상황별 대처방법을 확인할 수 있습니다.



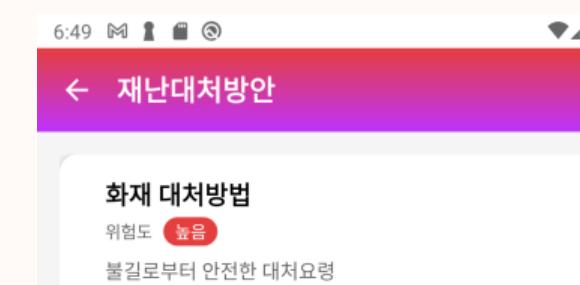
오늘의 재난 대처법

AI가 선별한 오늘의 재난 대처방안을 확인하세요



오늘의 재난 대처법

AI가 선별한 오늘의 재난 대처방안을 확인하세요



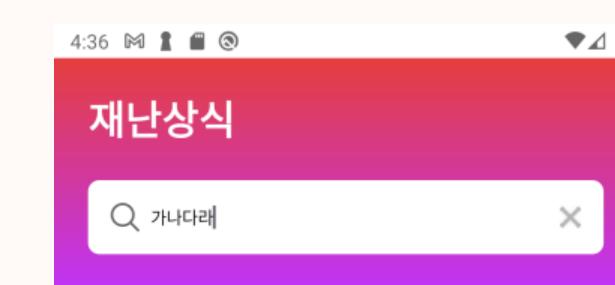
화재 대처방법

위험도
높음

불길로부터 안전한 대처요령

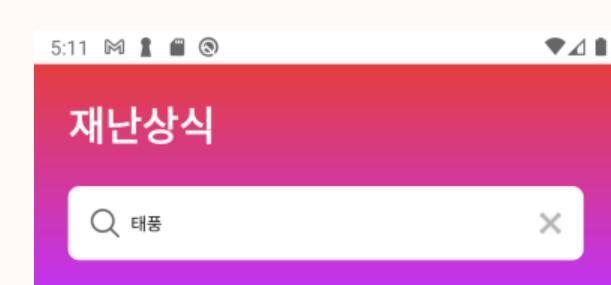
AI 생성 콘텐츠

이 내용은 AI가 분석하여 생성한 맞춤형 대처방안입니다. 실제 상황에서는 전문가의 조언을 우선시 바랍니다.



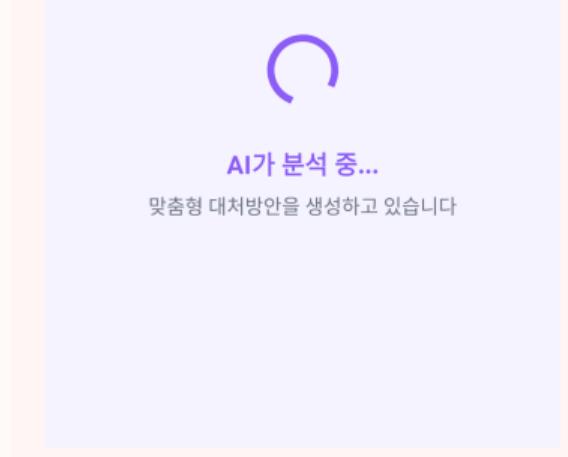
'가나다라' 검색결과

AI가 선별한 오늘의 재난 대처방안을 확인하세요



'태풍' 검색결과

AI가 선별한 오늘의 재난 대처방안을 확인하세요



AI가 분석 중...

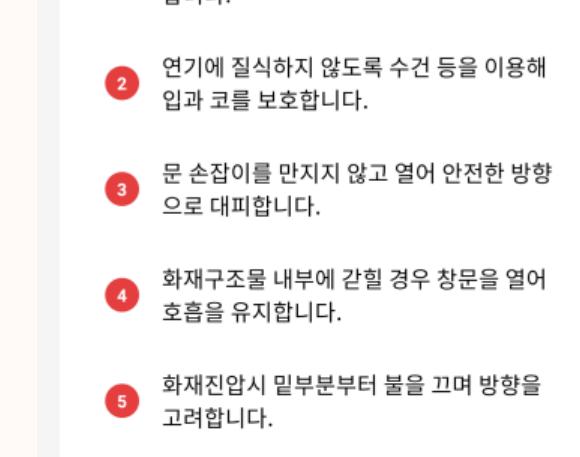
맞춤형 대처방안을 생성하고 있습니다



태풍 대처방법

강력한 태풍으로부터 안전한 대처법

자연재해 위험도: 보통

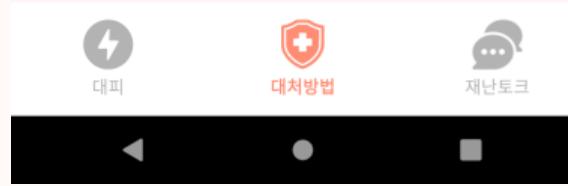


화재 대처방법

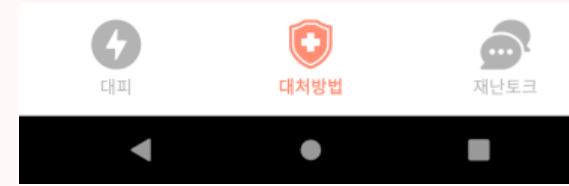
불길로부터 안전한 대처요령

위험도
높음

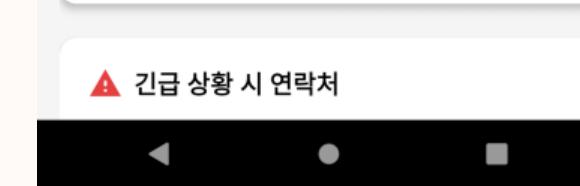
긴급 상황 시 연락처



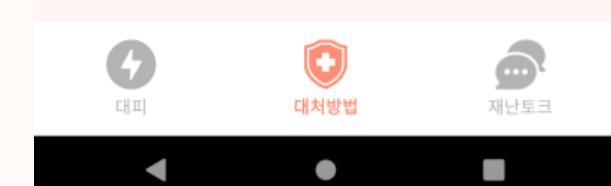
로딩중일때



오늘의 재난대처법 추천



상세 대처방안

재난과 관련없는
키워드 검색할때

재난대처방법 검색



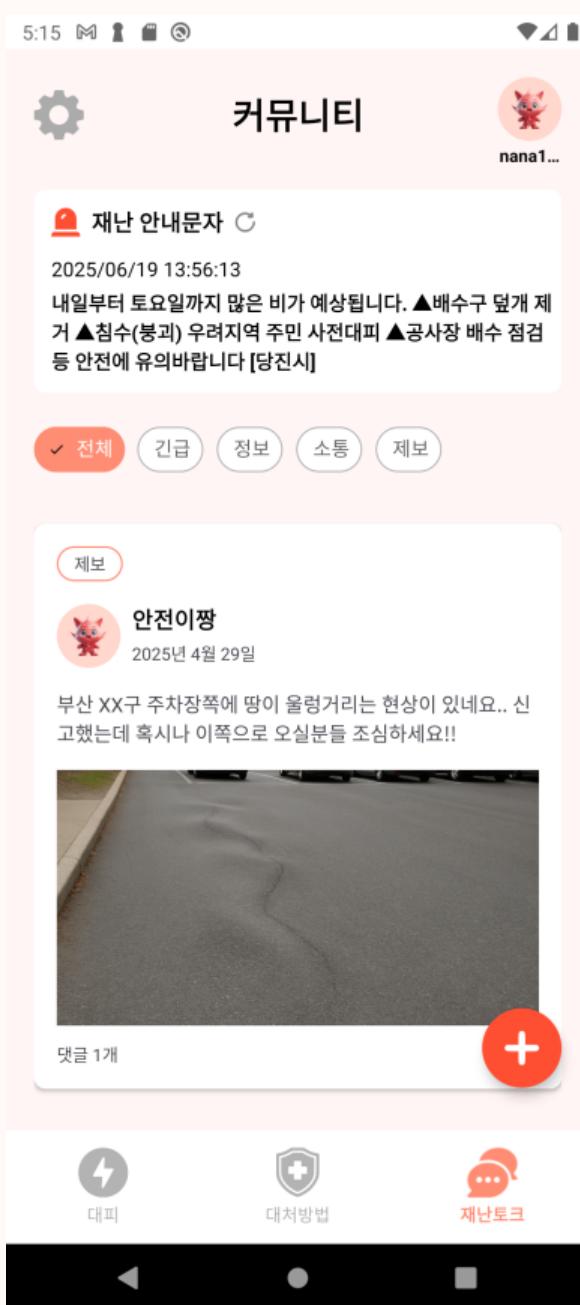
지키미

앱 주요기능 (5/5) - 재난 커뮤니티 기능

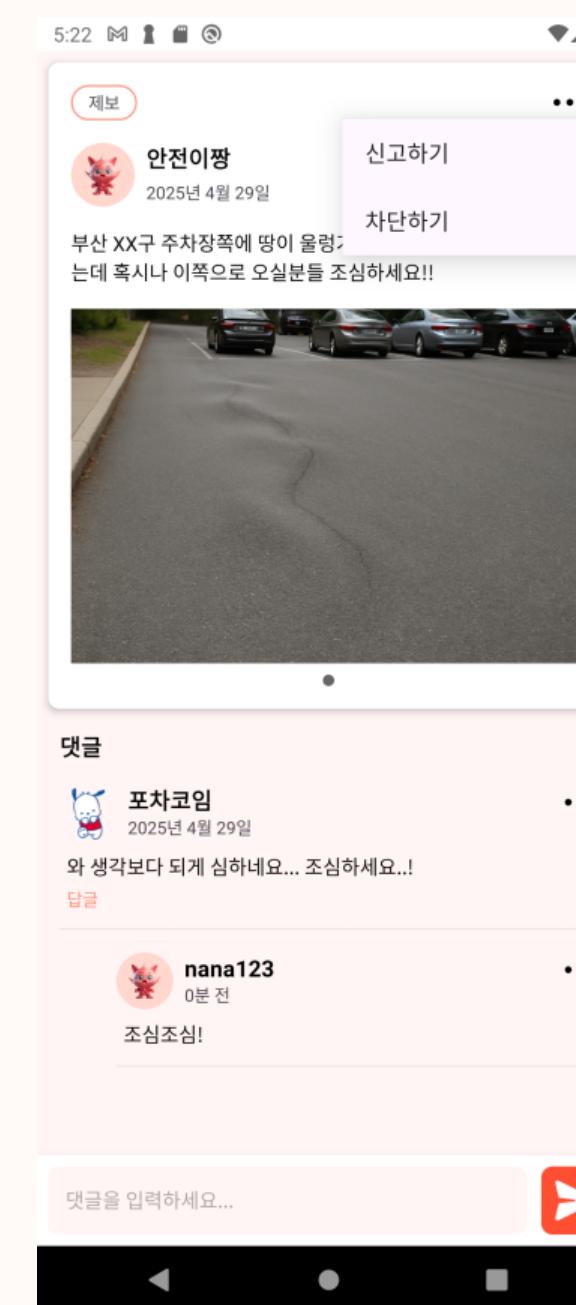
재난커뮤니티 기능을 통해 실시간으로 소통할 수 있습니다.

내가 작성한 게시물은 수정, 삭제를 할수있고 내가 작성한 댓글은 삭제만 가능합니다.

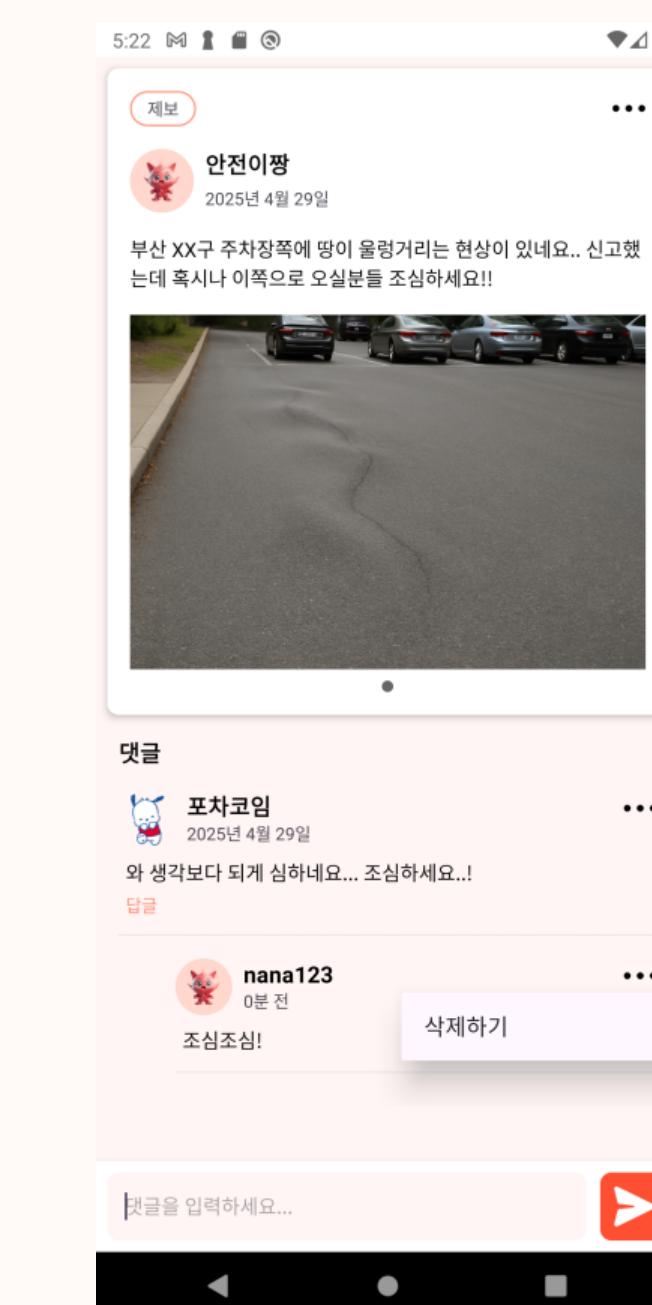
다른 사람의 게시물과 댓글이 부적절한 내용을 포함하고 있을 경우 신고, 차단이 가능합니다.[신고-개발자에게 신고내역 접수/ 차단-즉시 사용자에게 안보임]



게시물 화면



다른 사람의 게시물 & 댓글 [신고, 차단 기능]



내가 작성한 게시물 [수정, 삭제 기능] 내가 작성한 댓글 [삭제 기능]



프로필 편집기능



지키미

사용자 피드백 반영사항

카테고리별 반영

카테고리별로 볼수있으면 좋겠다는
사용자의 의견반영



회원가입, 로그인 진입시 로딩문제 해결

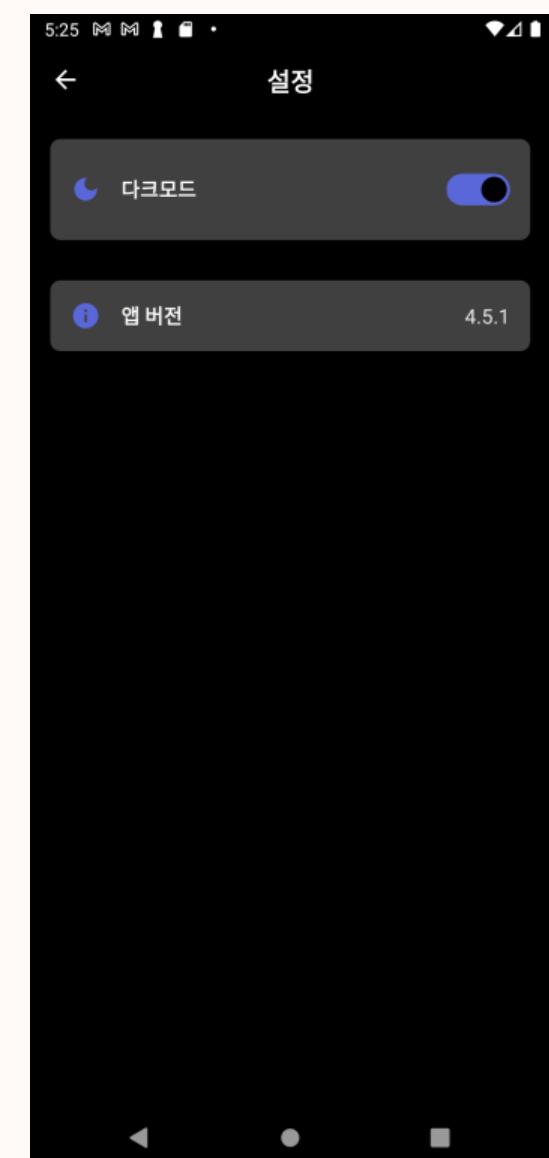
StateFlow의 초기값이 Resource.Loading(null)로 설정되어 있어서,
Fragment가 생성되자마자 바로 로딩상태가 되는 문제 발생

StateFlow의 초기값을
MutableStateFlow<Resource<User>?>(null)로 변경해준뒤,
Loading상태일때와 null상태일때를 람다함수로 분기처리하여 해결

```
// viewModel에서 StateFlow로 회원가입 상태를 관찰
private fun setupObservers() {
    viewLifecycleOwner.lifecycleScope.launch {
        viewLifecycleOwner.repeatOnLifecycle(Lifecycle.State.STARTED) {
            viewModel.signupStatus.collect { resource ->
                when (resource) {
                    is Resource.Loading -> {
                        binding.progressBar.visibility = View.VISIBLE
                    }
                    is Resource.Success -> {
                        binding.progressBar.visibility = View.GONE
                        (activity as MainActivity).showToast("회원가입 성공")
                        findNavController().navigate(R.id.loginFragment)
                    }
                    is Resource.Error -> {
                        binding.progressBar.visibility = View.GONE
                        (activity as MainActivity).showToast(resource.message ?: "회원가입 실패")
                    }
                    null ->{
                        // 초기 상태
                        binding.progressBar.visibility = View.GONE
                    }
                }
            }
        }
    }
}
```

다크모드 반영

다크모드 기능 사용자가 생각보다 많음을 느끼고,
다크모드 기능 구현해 반영

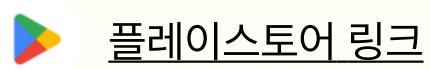




맛고

지역별 특산물을 활용한 요리레시피 제공 앱

개발기간	2024.05.27 ~ 07.05
개발인원	팀 프로젝트 [4명]
담당역할	지도기반 지역별 특산물 정보확인 [기여도: 100%] 레시피 북마크 기능 [기여도: 100%] 로그인 & 회원가입 [기여도: 50%] 서비스 기획 및 디자인 [기여도: 80%]



깃허브 링크

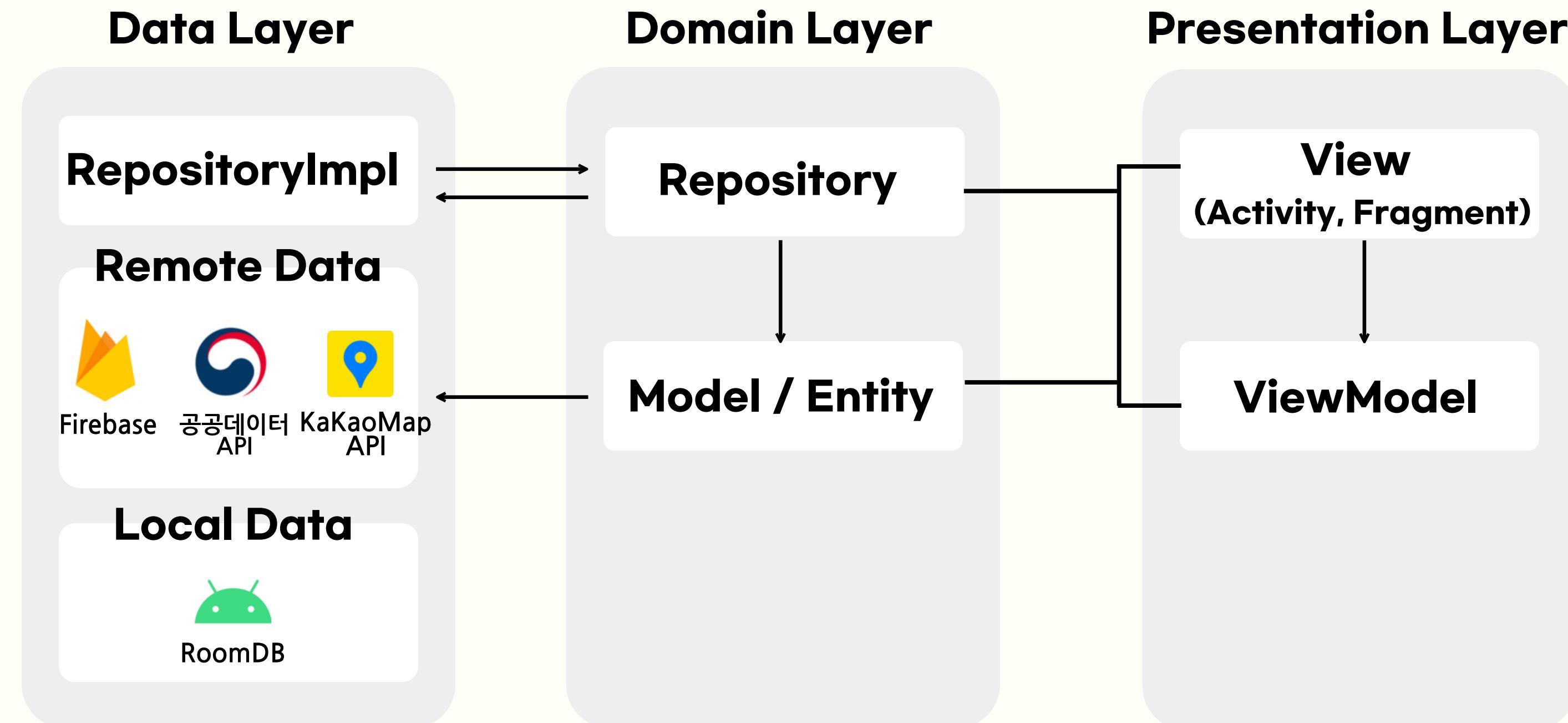
사용기술

언어	Kotlin
디자인패턴	MVVM + Repository Pattern
라이브러리 및 프레임워크	Retrofit2, Okhttp3, Glide, Flow, Coroutine, Hilt DiffUtil, Jsoup, KakaoMap, XML, Navigation
데이터베이스 및 서버	Firebase, RoomDB

주요기능

지도기반 지역별 특산물 정보확인[기여도:100%] / 특산물을 활용한 레시피 제공 / 레시피 작성 / 북마크 기능[기여도:100%] / 로그인&회원가입[기여도:50%]





MVVM + Repository Pattern

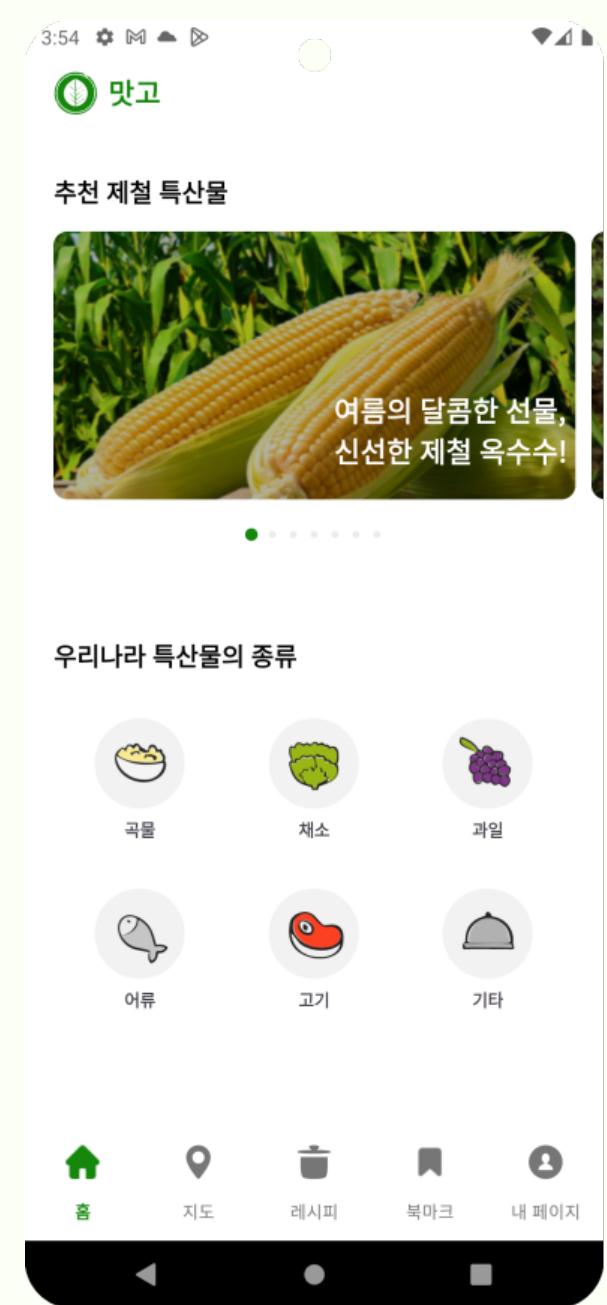
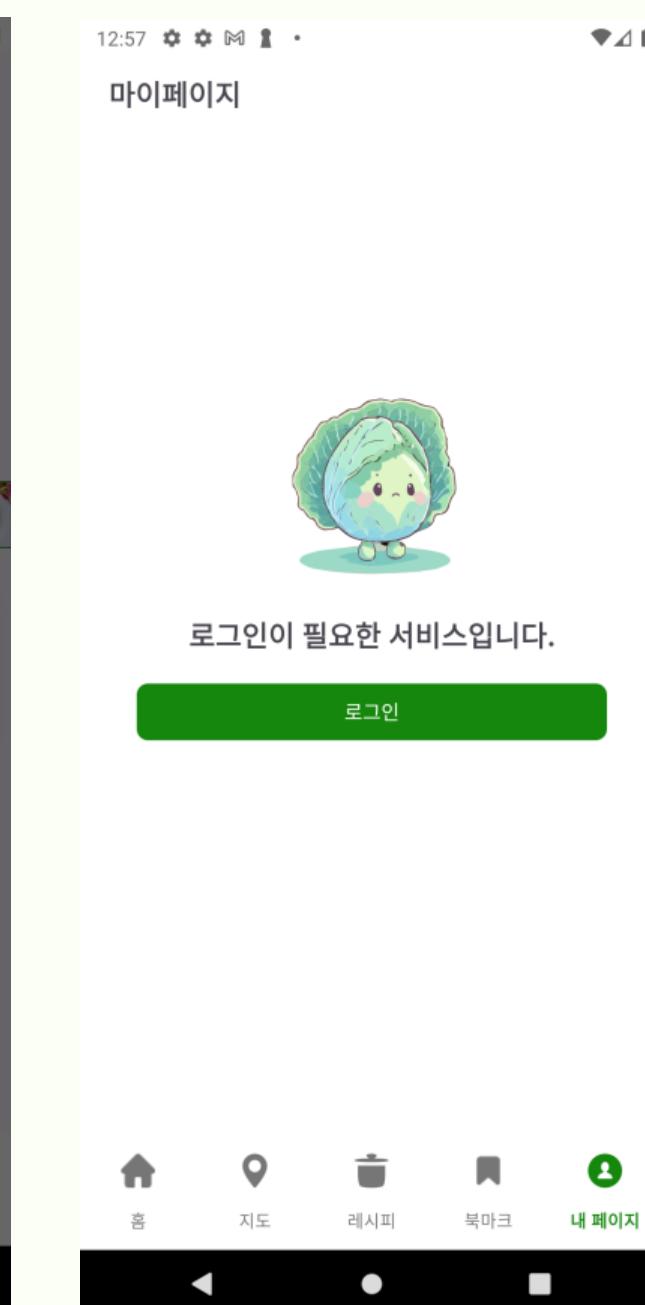
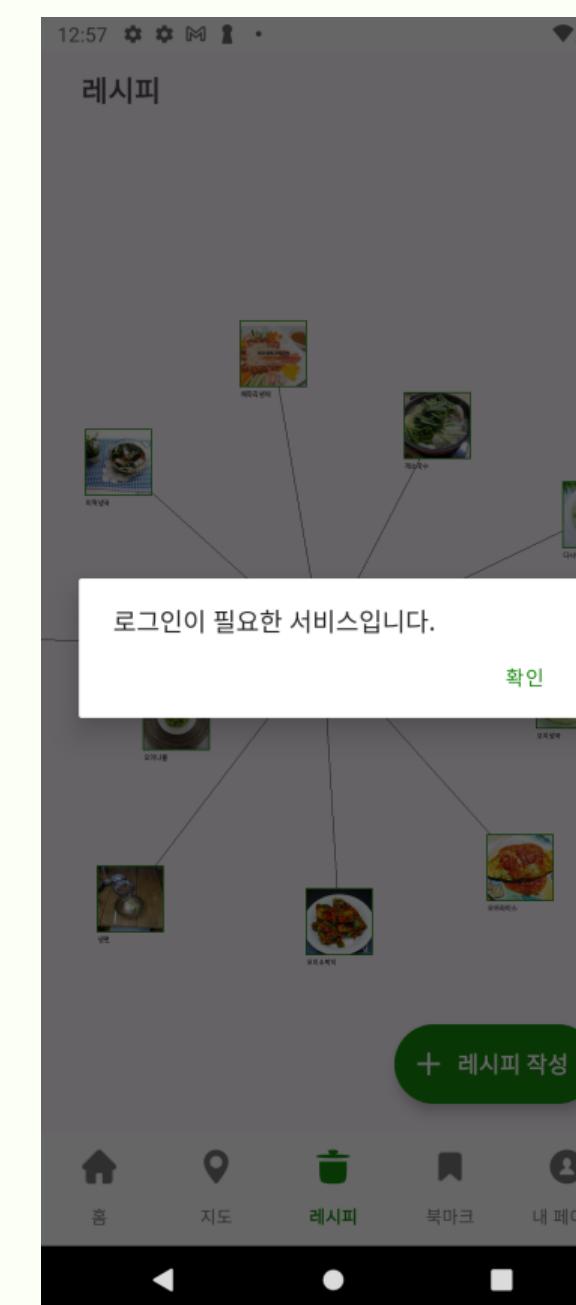
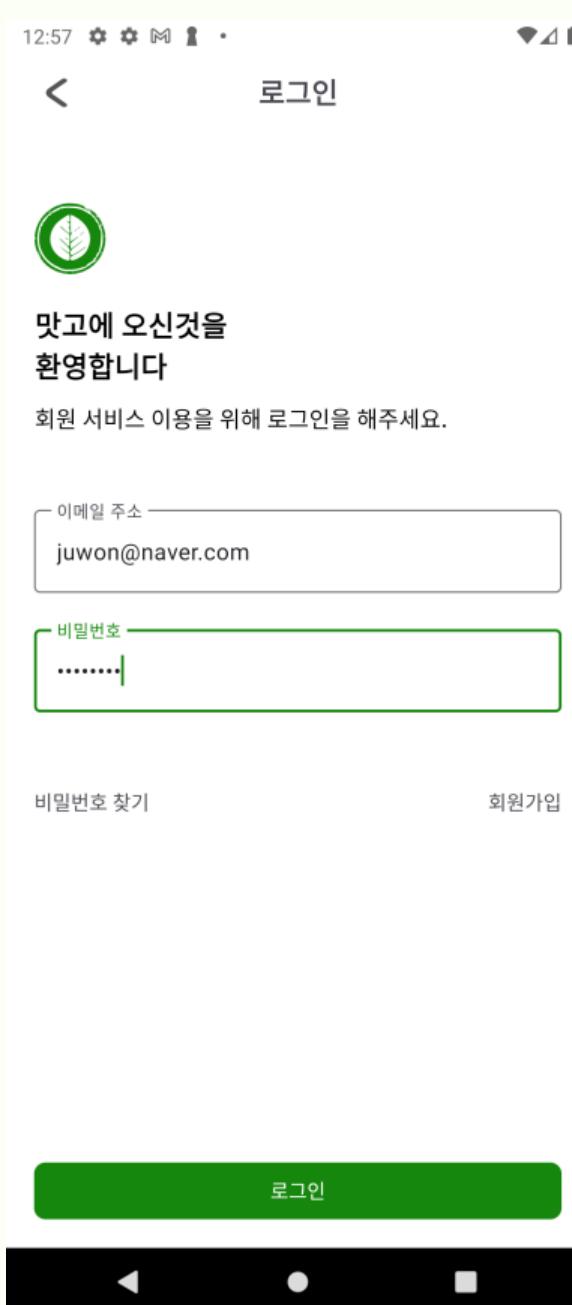
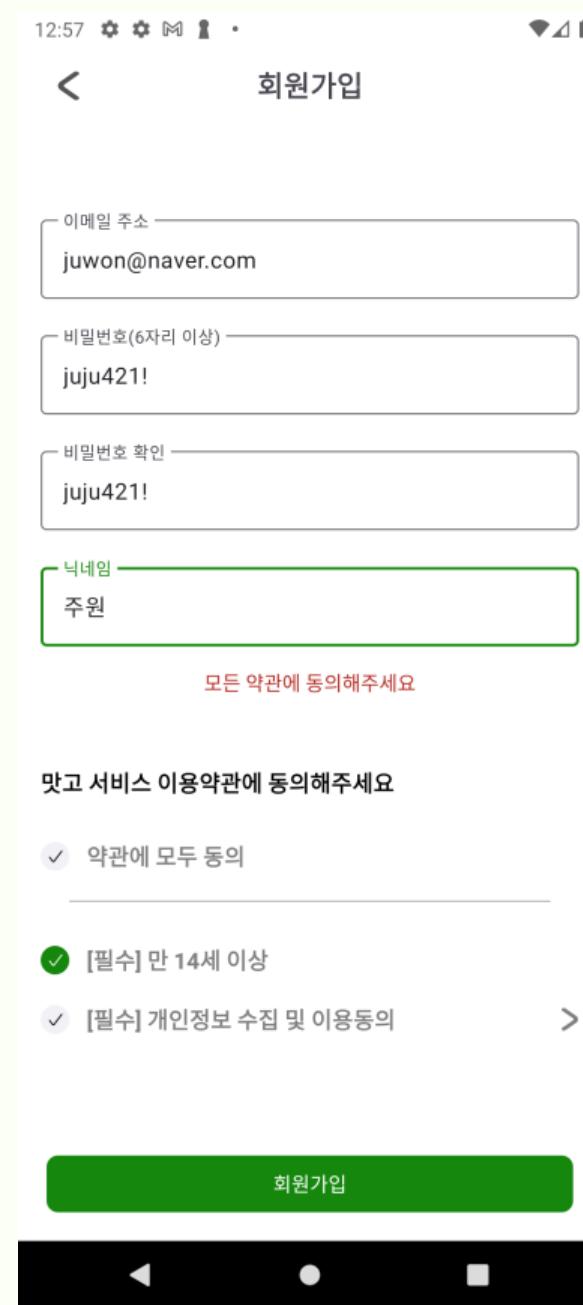


맛고

앱 주요기능(1/4) - 로그인 & 회원가입 / 홈화면

이용약관에 모두 동의해야, 회원가입이 완료됩니다.
로그인을 하지 않아도 레시피작성, 내 페이지를 제외한 기능을 사용할 수 있습니다.

홈화면에서는 제철 특산물 배너와 특산물의 종류를 확인할 수 있습니다.



회원가입 화면

로그인 화면

로그인이 필요한 화면

홈화면

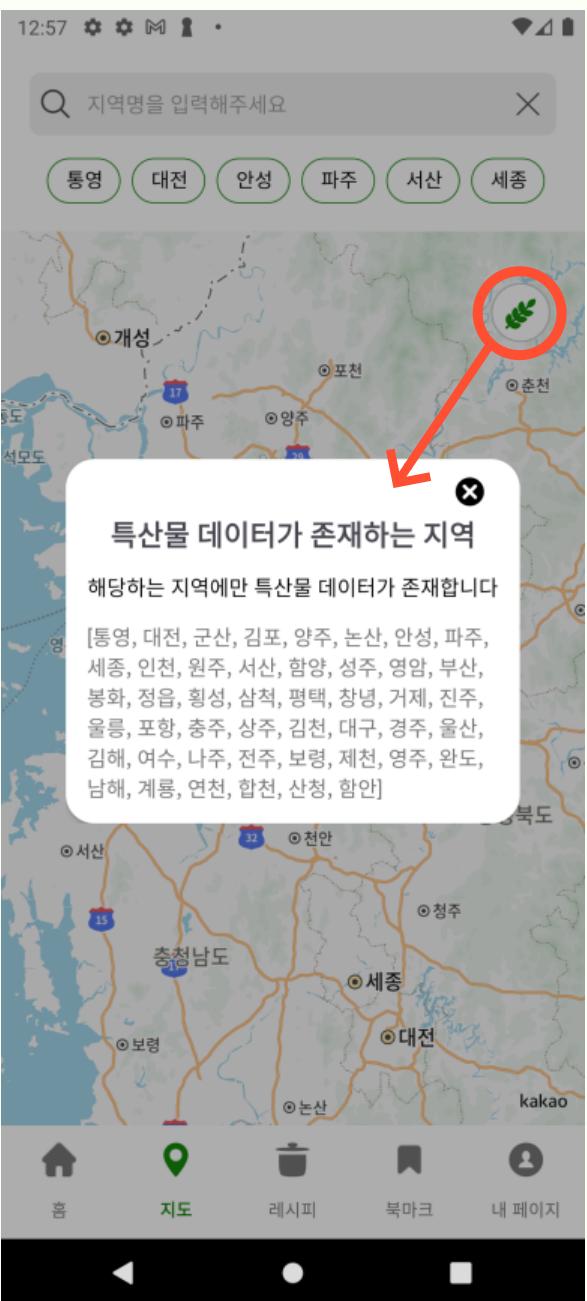


맛고

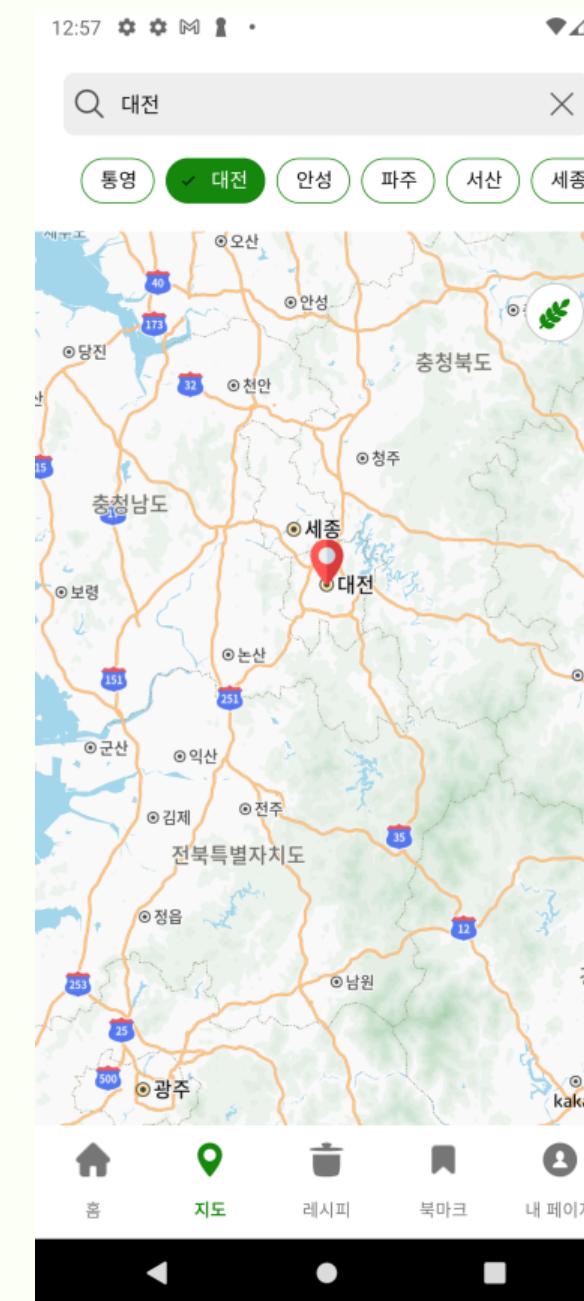
앱 주요기능(2/4) - 지도기반 지역별 특산물 정보확인

지도를 통해 해당지역의 특산물을 확인할 수 있습니다.

다이얼로그를 통해 특산물 데이터가 존재하는 지역을 제공합니다.



특산물 데이터 존재지역
다이얼로그



검색 or 칩버튼을 통한
지역검색



지역 특산물 제공화면



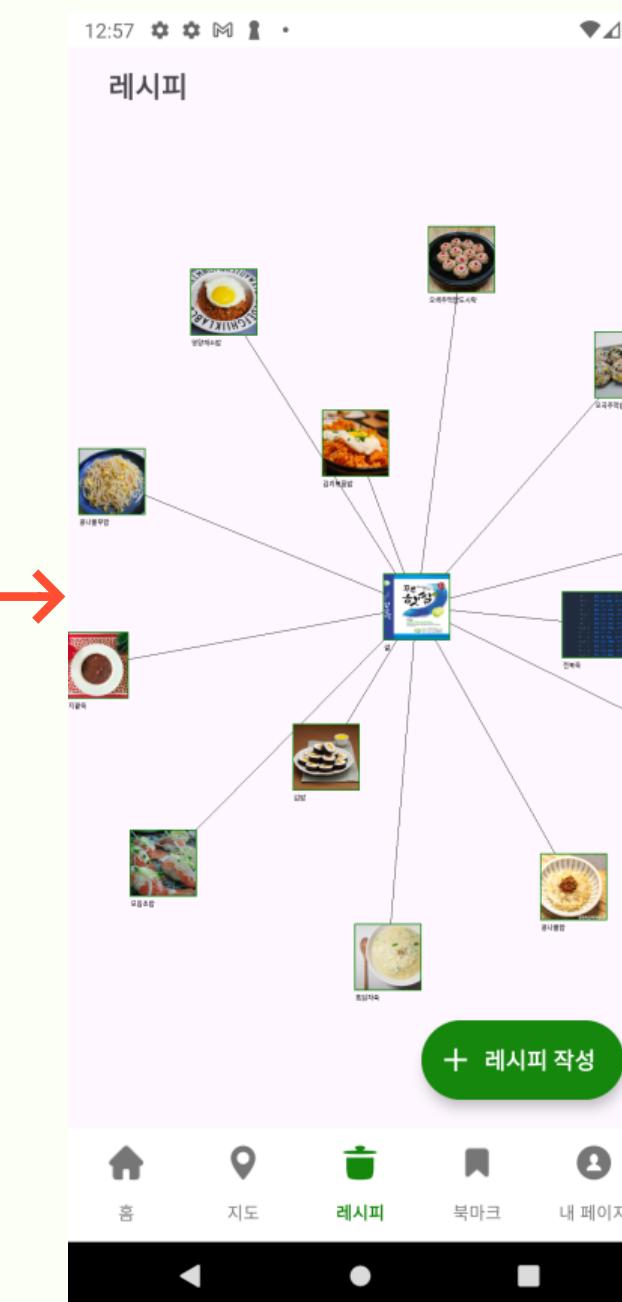
맛고

앱 주요기능(3/4) - 특산물을 활용한 레시피제공 및 레시피 작성기능

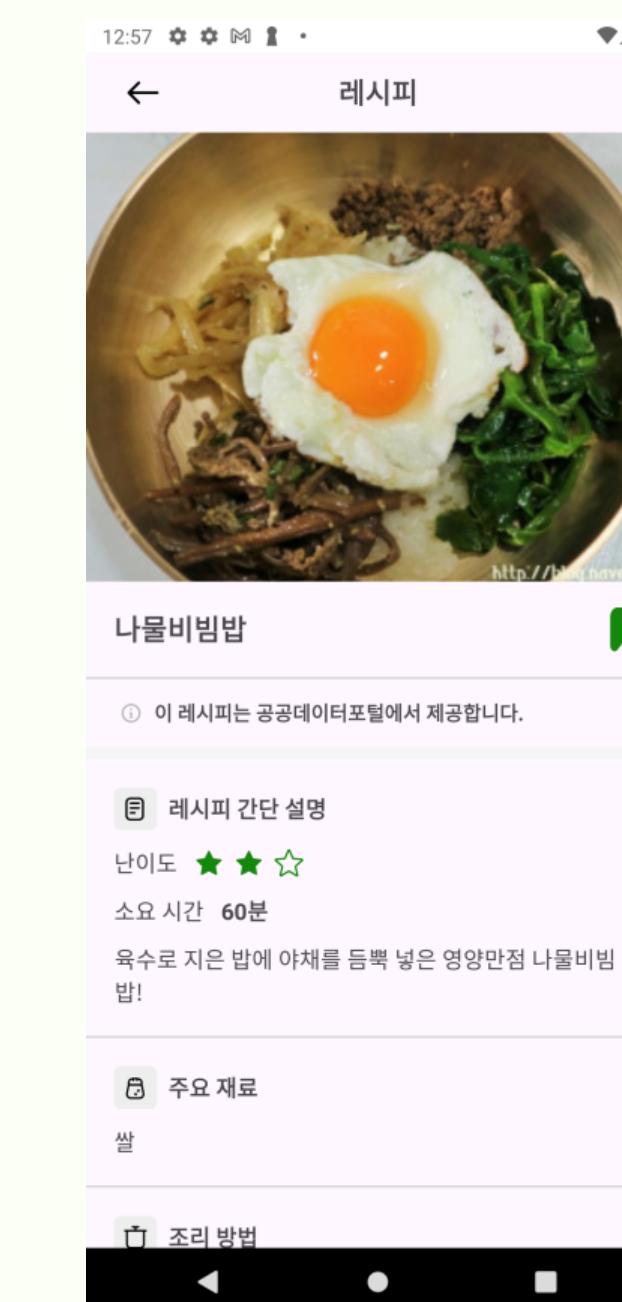
레시피화면에서 특산물을 길게클릭하면, 해당 특산물과 관련한 레시피가 마인드맵 형식으로 제공됩니다.
레시피를 누르면 상세레시피를 확인할 수 있습니다.
사용자가 직접 레시피를 작성할수도 있습니다.



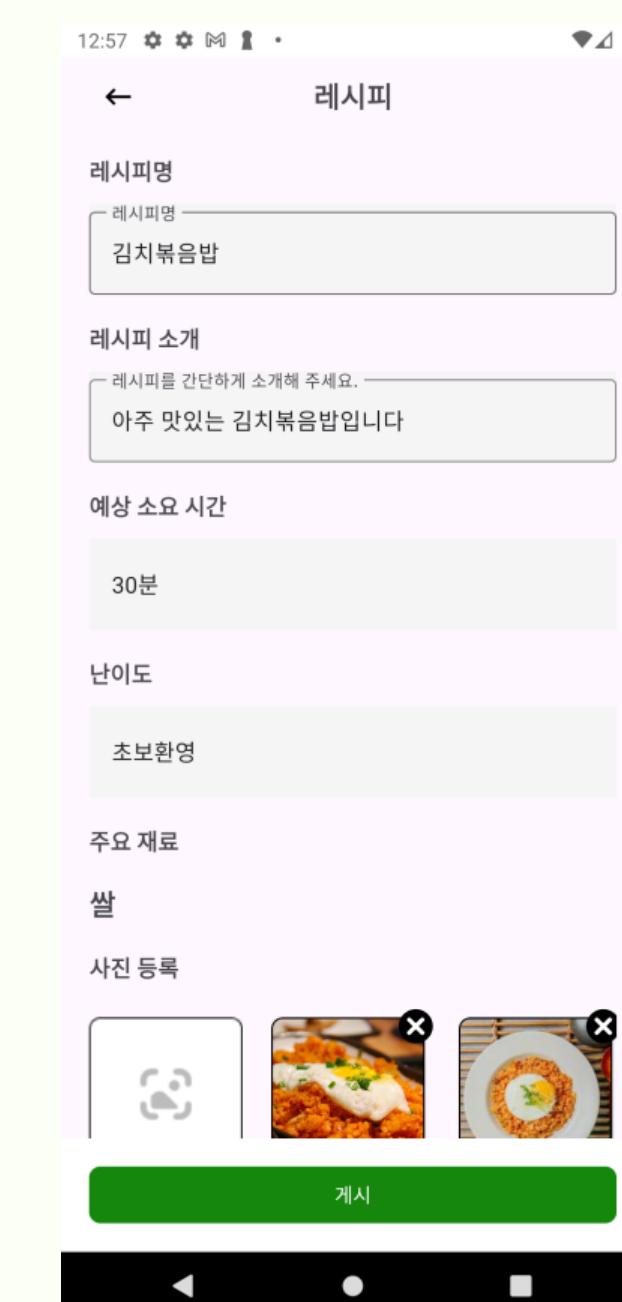
레시피 로딩 중 화면



레시피 로딩 완료 화면



레시피 상세화면



레시피 작성화면

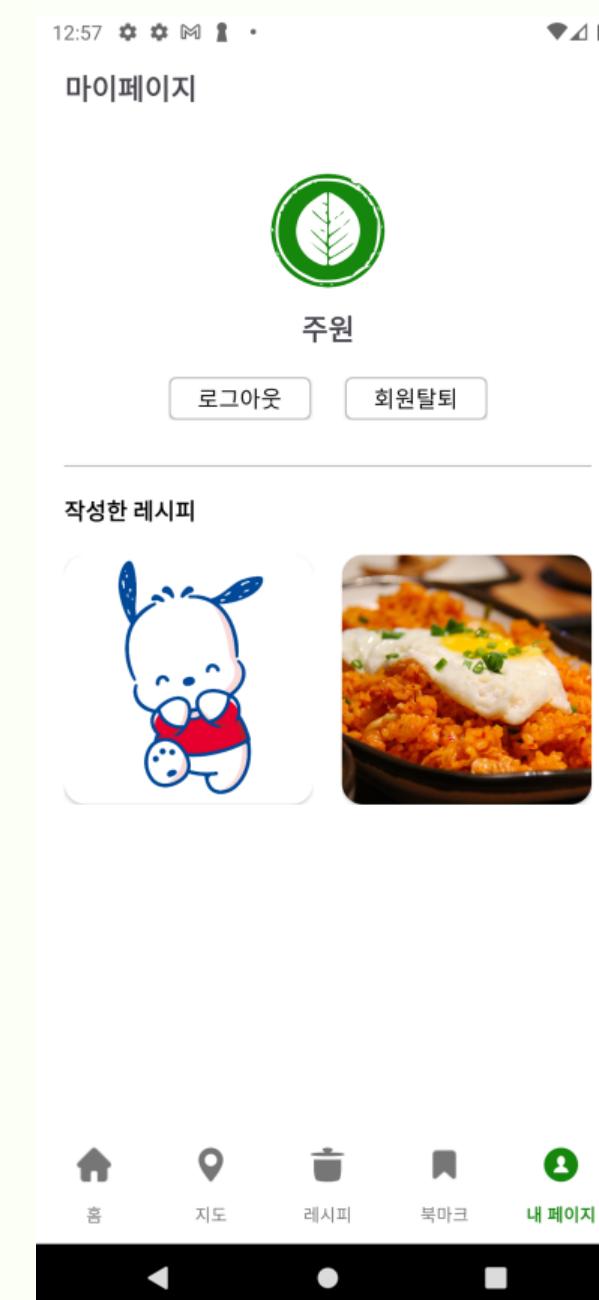


맛고

앱 주요기능(4/4) - 레시피 북마크 기능 / 마이페이지

레시피 상세화면에서 북마크를 통해 레시피를 저장할 수 있습니다.
 북마크탭에서 레시피를 모아볼 수 있고, 롱클릭시 북마크를 삭제할 수 있습니다.

마이페이지에서는 내가 작성한 레시피를 모아서 확인할 수 있습니다.

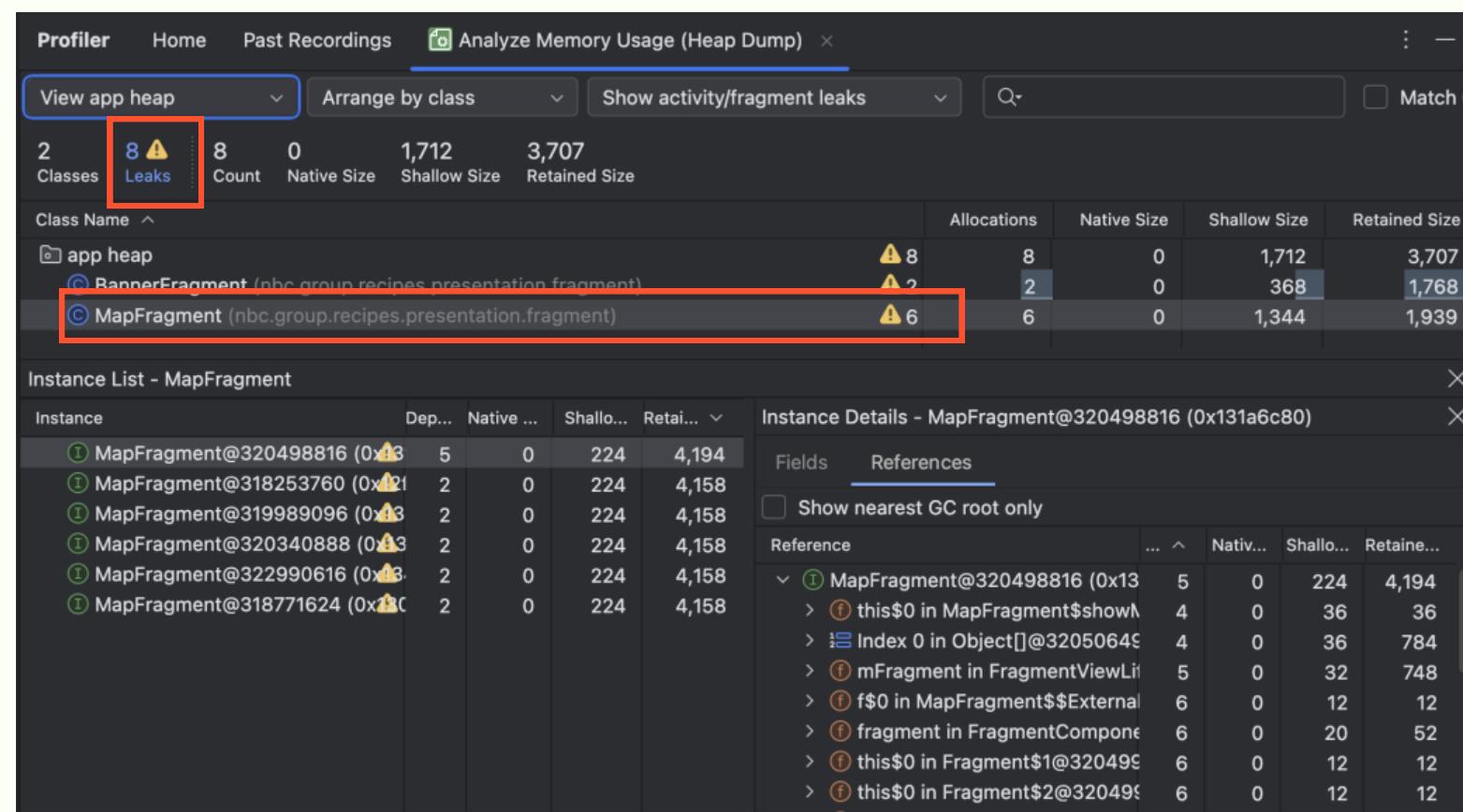




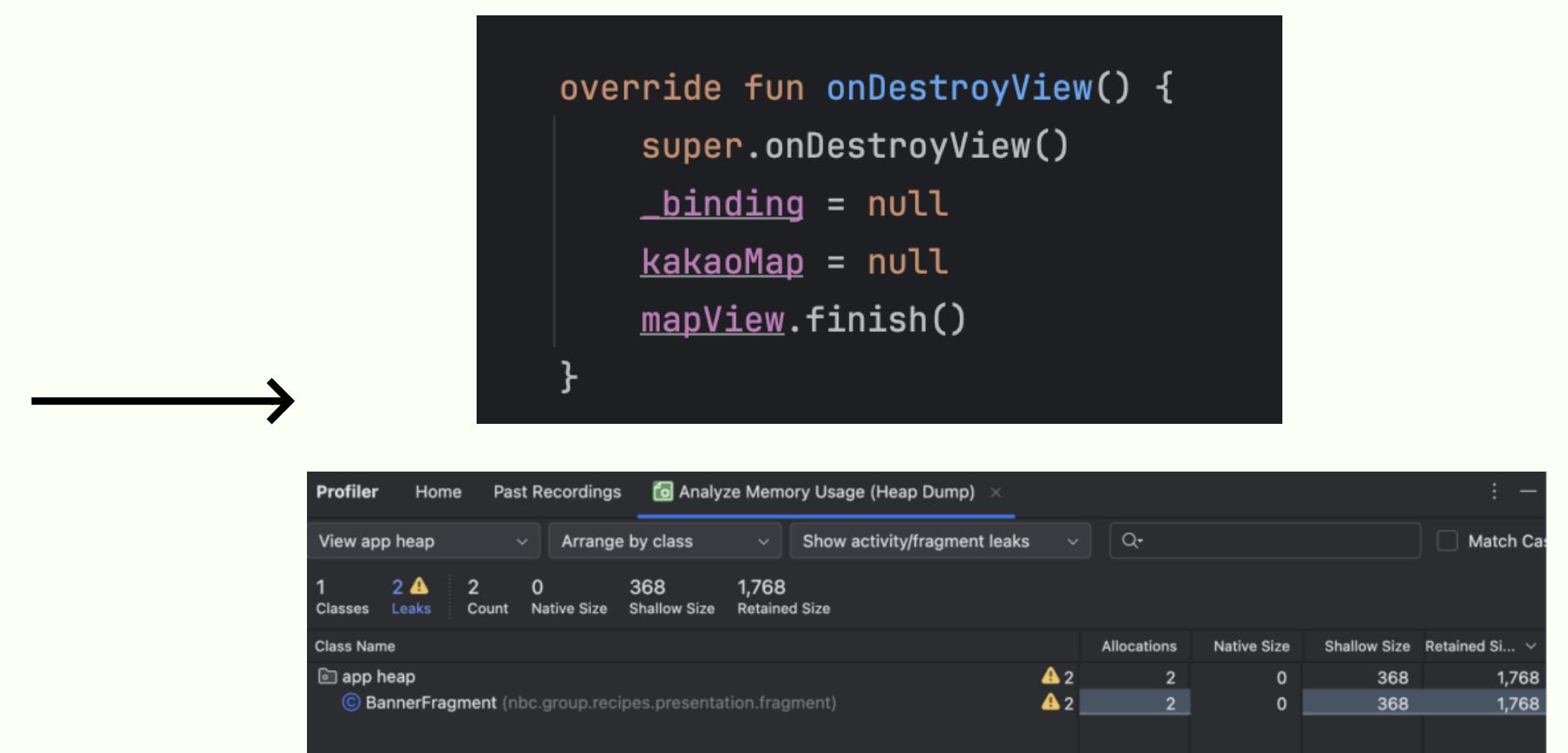
맛고

Fragment 생명주기를 고려해, 메모리누수 3배 이상 감소

Profiler를 사용하여 메모리 누수를 확인한 결과 MapFragment에서 약 100MB의 메모리 누수가 발생했고, 이로 인해 Fragment 반복호출 시, Fragment에 메모리가 지속적으로 누적되어 앱이 종료되는 문제가 발생했습니다. 이를 해결하기 위해 Fragment의 생명주기를 고려하여, `MapView.finish()`를 `onDestroyView()` 상태에서 호출함으로써 메모리 누수를 약 30MB로 줄였습니다. 이 결과 메모리 누수를 약 3배 이상 감소시키며 사용자 경험을 개선했습니다.



MapFragment에서 메모리누수 발생



`onDestroyView()`에서
`mapView.finish()`를 호출함으로써
메모리누수 해결

프로젝트 소개



MiniYoutube | 카테고리별로 모아보는 나만의 MiniYoutube앱

개발기간	2024.05.13 ~ 05.23
개발인원	팀 프로젝트 [4명] - 팀장
담당역할	YoutubePlayer 라이브러리를 통한 영상재생 [기여도:100%] 영상 저장 & 공유기능 [기여도:100%] 서비스 기획 및 디자인 [기여도:90%]



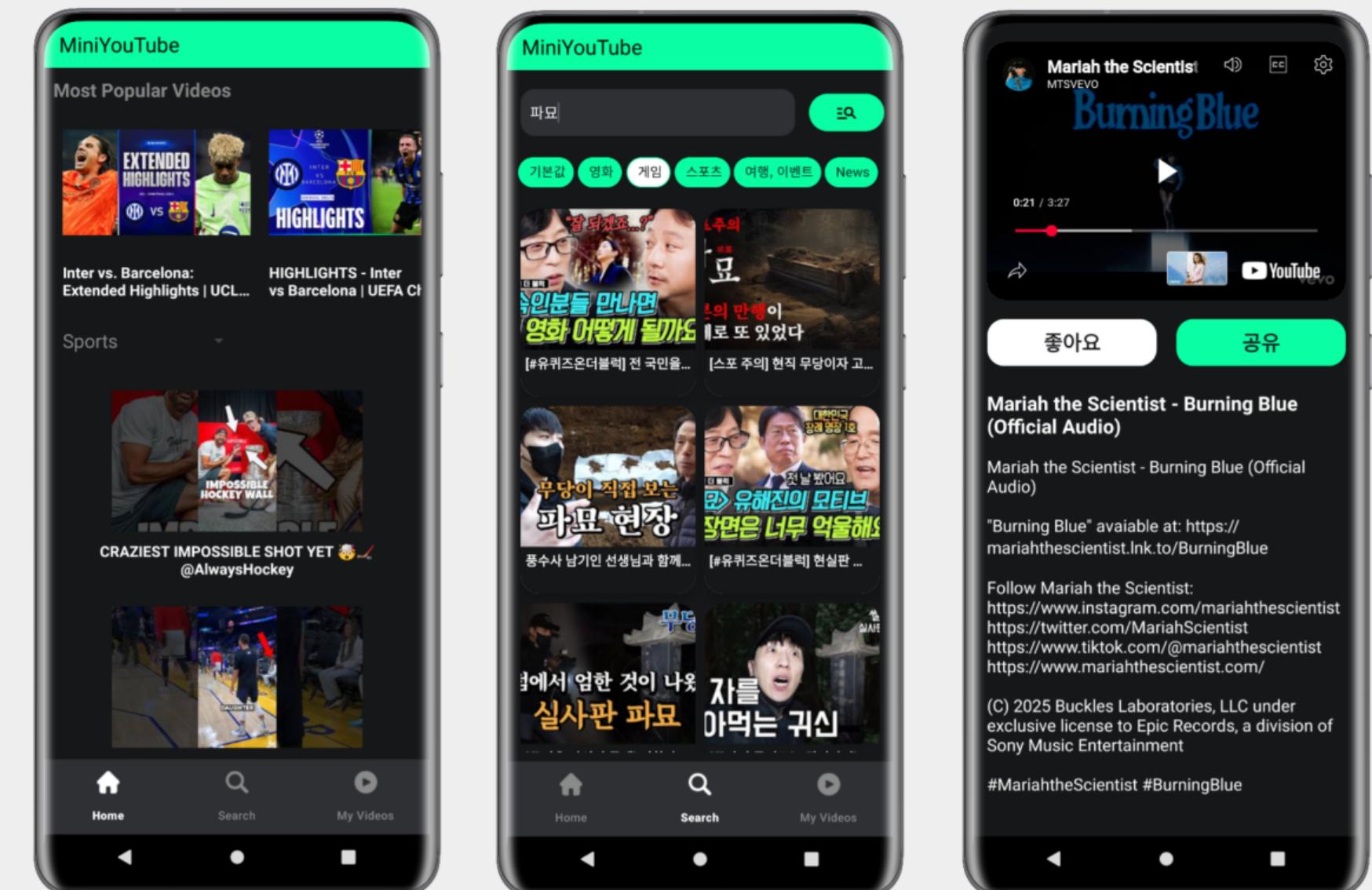
[깃허브 링크](#)

사용기술

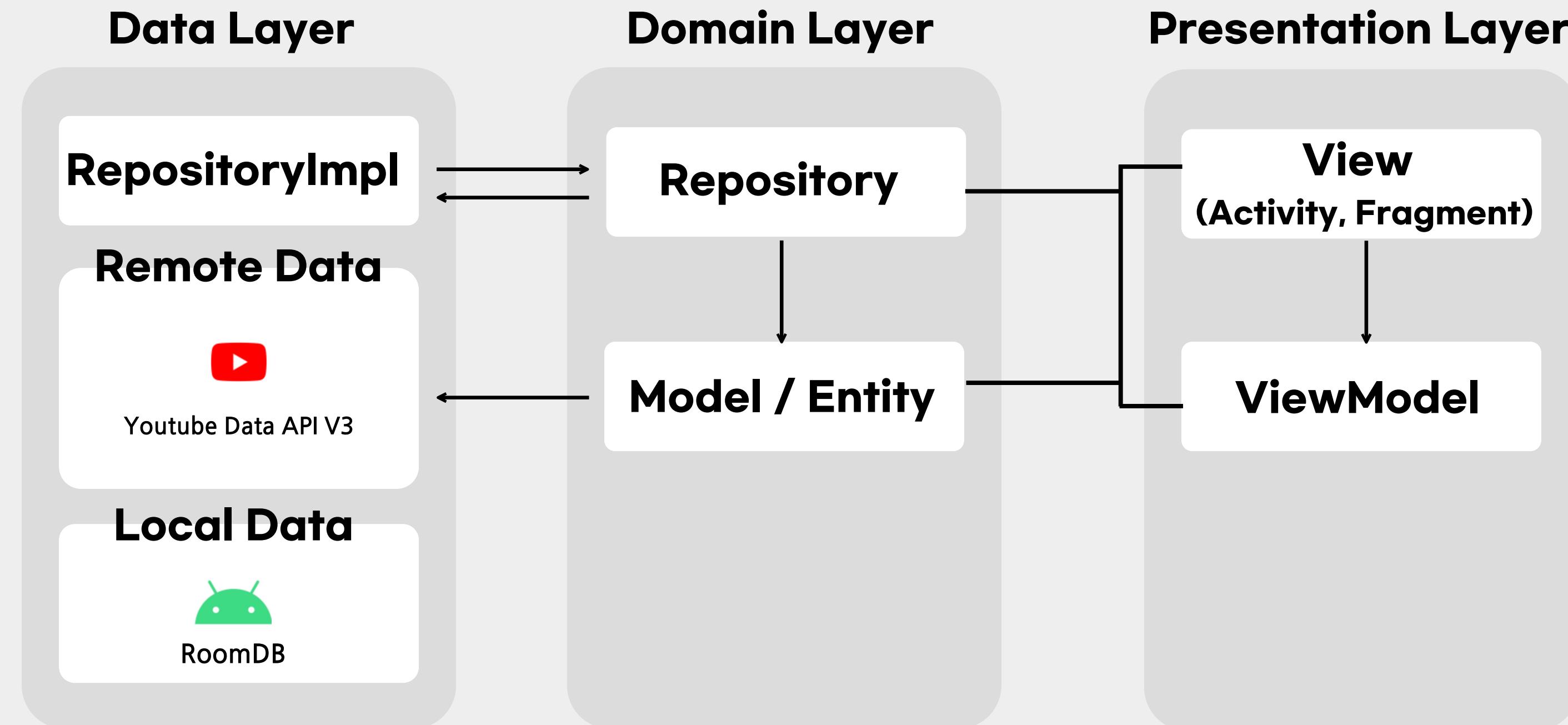
언어	Kotlin
디자인패턴	MVVM + Repository Pattern
라이브러리 및 프레임워크	Retrofit2, Okhttp3, Glide, LiveData, Coroutine, Hilt , DiffUtil, YouyubePlayer 라이브러리, XML
데이터베이스 및 서버	RoomDB

주요기능

인기&다양한 카테고리별 영상제공 / 영상 검색기능 / YoutubePlayer 라이브러리를 통한 영상재생
[기여도:100%] / 영상 저장&공유 기능[기여도:100%]



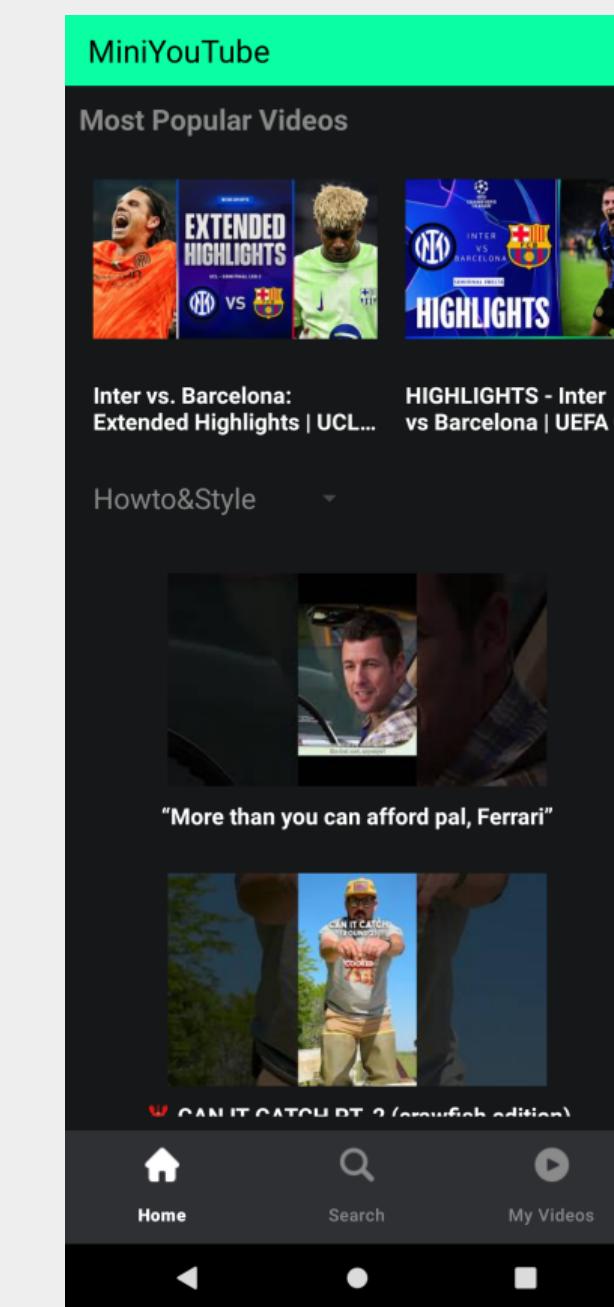
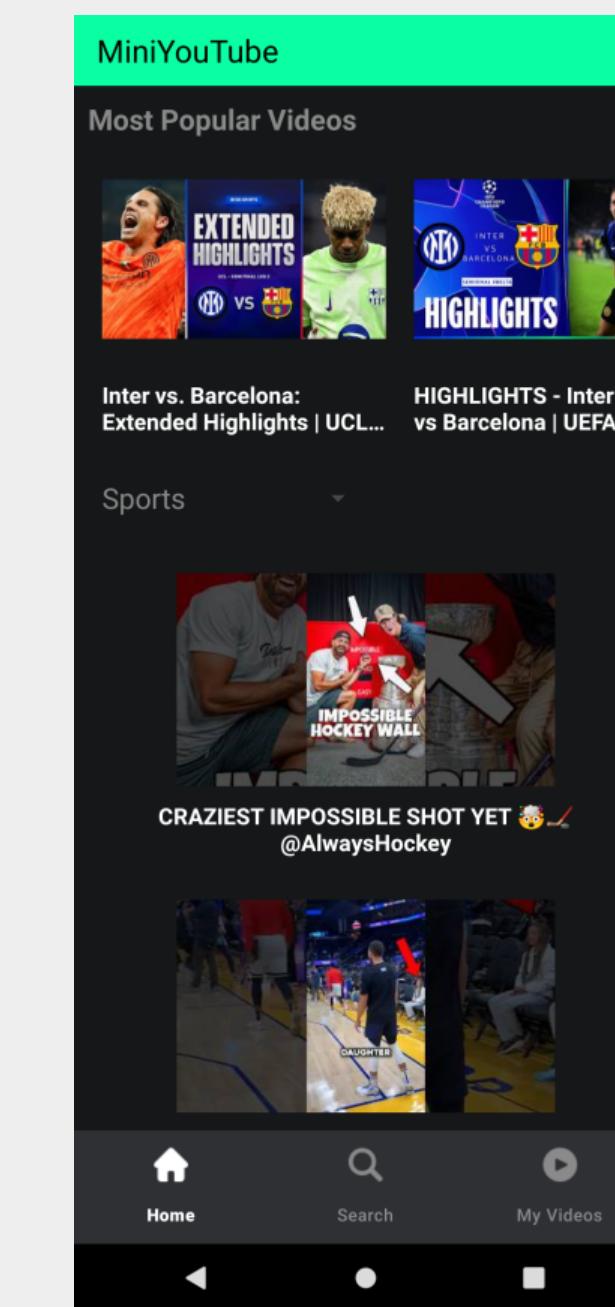
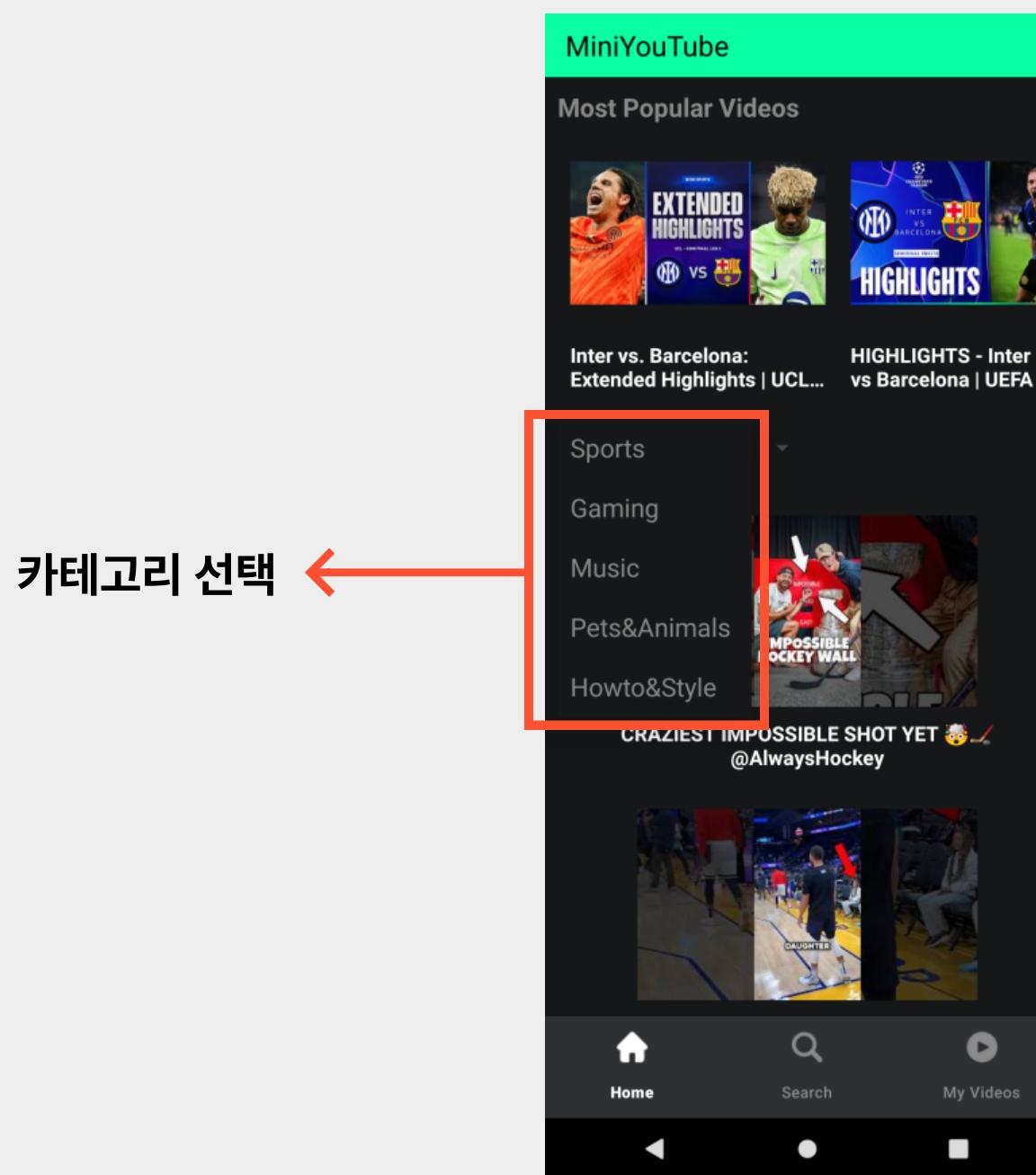
M MiniYoutube | 서비스 구조



MVVM + Repository Pattern

M MiniYoutube | 앱 주요기능 (1/3) - 홈화면

최상단에는 현재 가장 인기있는 영상 5개를 순서대로 보여줍니다.
카테고리를 선택하면, 카테고리와 관련된 인기있는 영상을 보여줍니다.



Sports 카테고리 선택

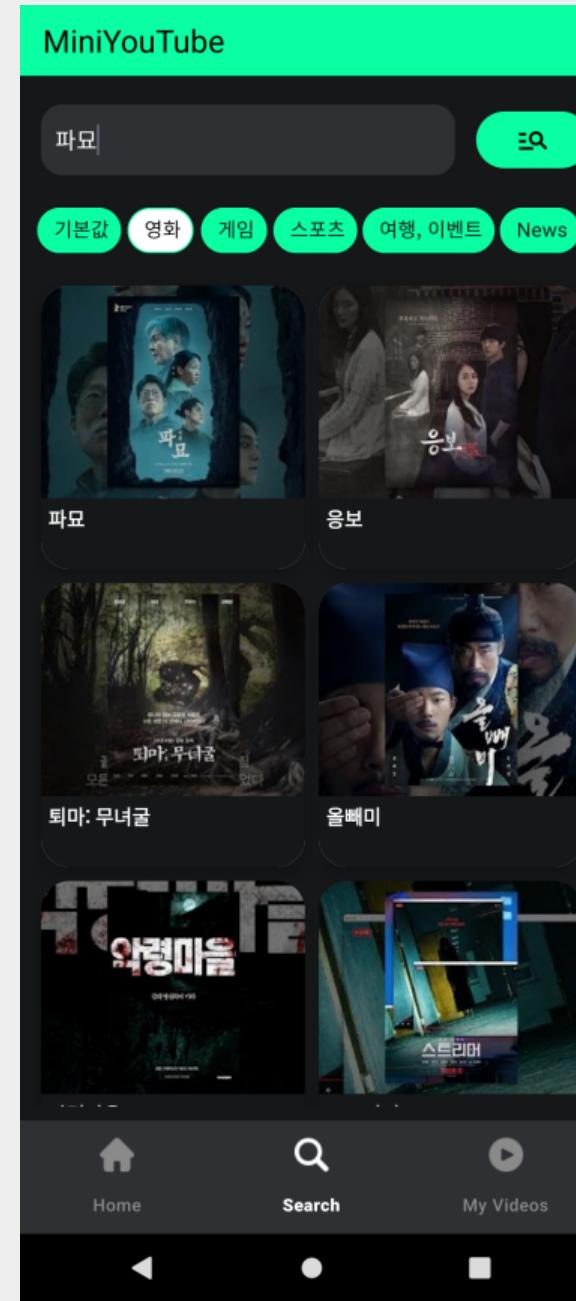
Howto&Style 카테고리 선택

M MiniYoutube | 앱 주요기능 (2/3) - 검색기능

원하는 영상을 검색한뒤, 다양한 카테고리별로 영상을 확인할 수 있습니다.



기본값 카테고리



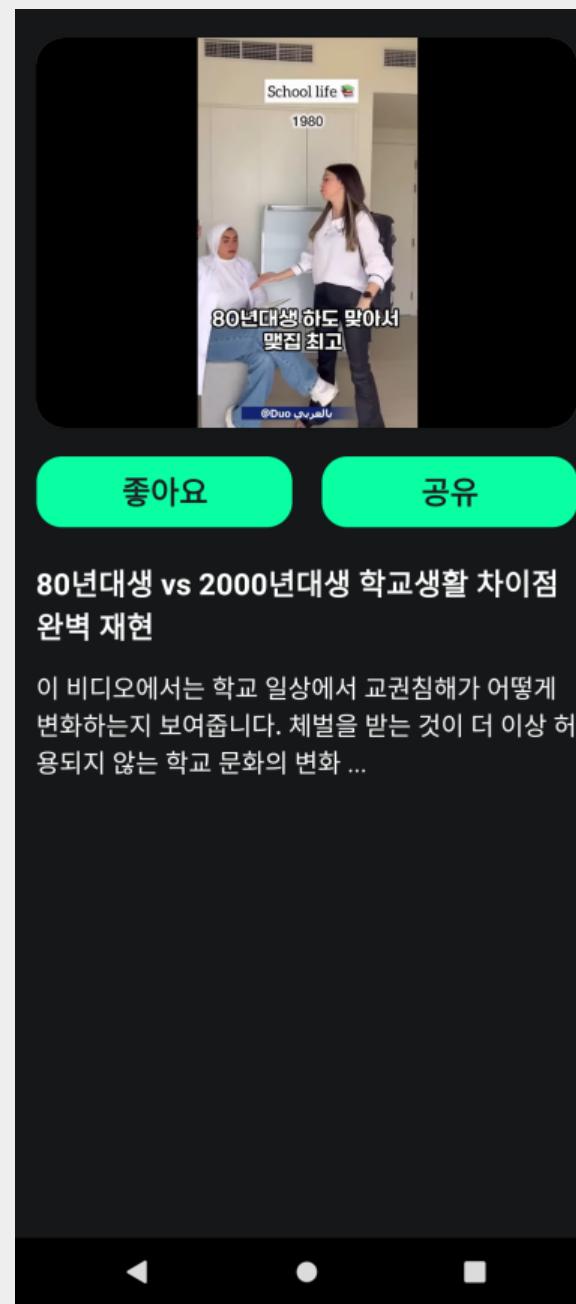
영화 카테고리



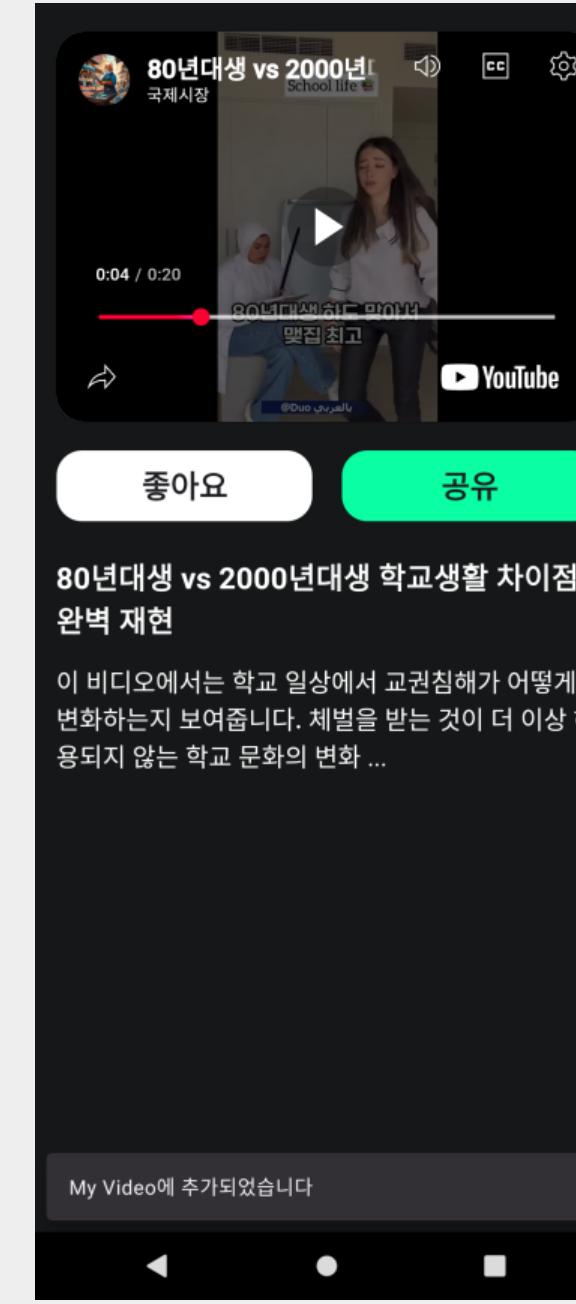
News 카테고리

M MiniYoutube | 앱 주요기능 (3/3) - 디테일 화면 & 마이비디오 화면

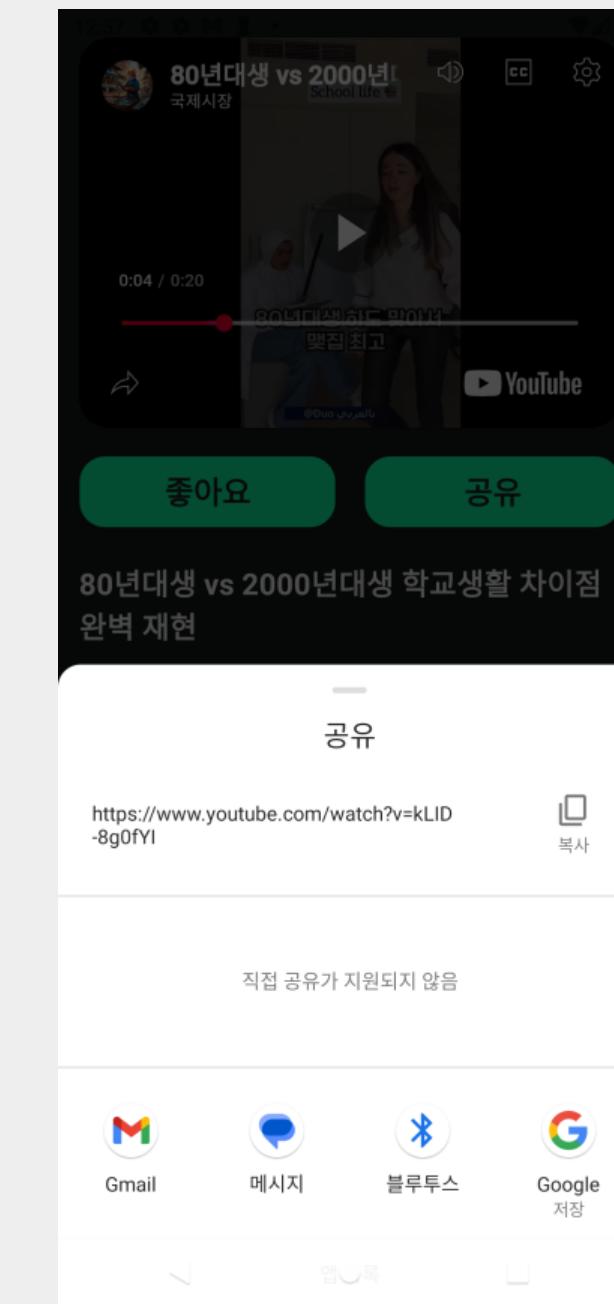
디테일 화면에서는 YoutubePlayer라이브러리를 통해 바로 영상시청을 할 수 있습니다. 또한 저장, 공유 기능도 제공합니다.
마이비디오 화면에서는 내가 저장한 영상을 따로 확인하고 시청할 수 있습니다.



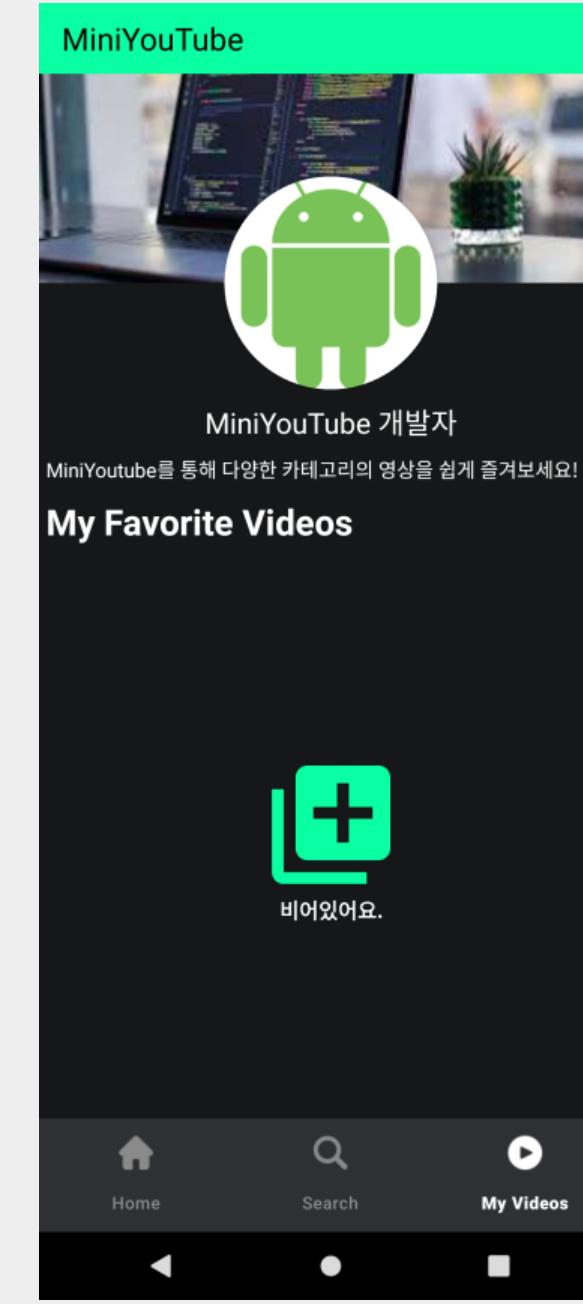
영상 시청



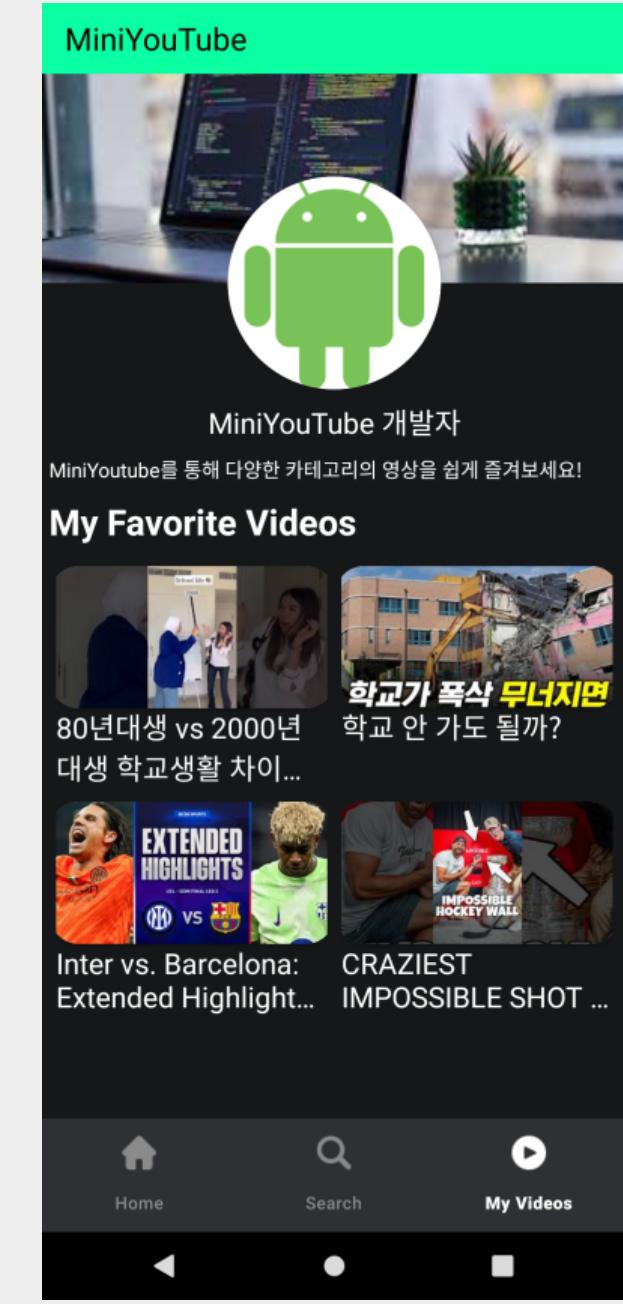
좋아요 기능



공유 기능



저장한 영상이 없는상태



저장한 영상이 있는상태

스키마값 구조 변경에 따른 RoomDB 오류 해결

- 해결방법1) 앱 자체를 삭제해서 데이터를 지우는 방법
- 해결방법2) 데이터베이스의 버전을 증가시키고, 마이그레이션 코드를 작성해주는 방법
→ 해당 프로젝트에서는 해당 오류를 앱 삭제로 간단하게 해결했으며,
배포앱인 지키미앱 개발시에는 데이터베이스 버전 증가와 마이그레이션 코드를 작성해 해결했습니다.

```
@Database(entities = [LikeEntity::class, ShelterEntity::class], version = 2)
abstract class AppDatabase : RoomDatabase(){
    abstract fun shelterDao() : ShelterDao
}
```

버전증가

```
@Module
@InstallIn(SingletonComponent::class)
object RoomModule {
    @Singleton
    @Provides
    fun provideAppDatabase(@ApplicationContext context: Context): AppDatabase = Room.databaseBuilder(
        context,
        AppDatabase::class.java,
        name: "app.db"
    )
        .addMigrations(MIGRATION_1_2)
        .build()

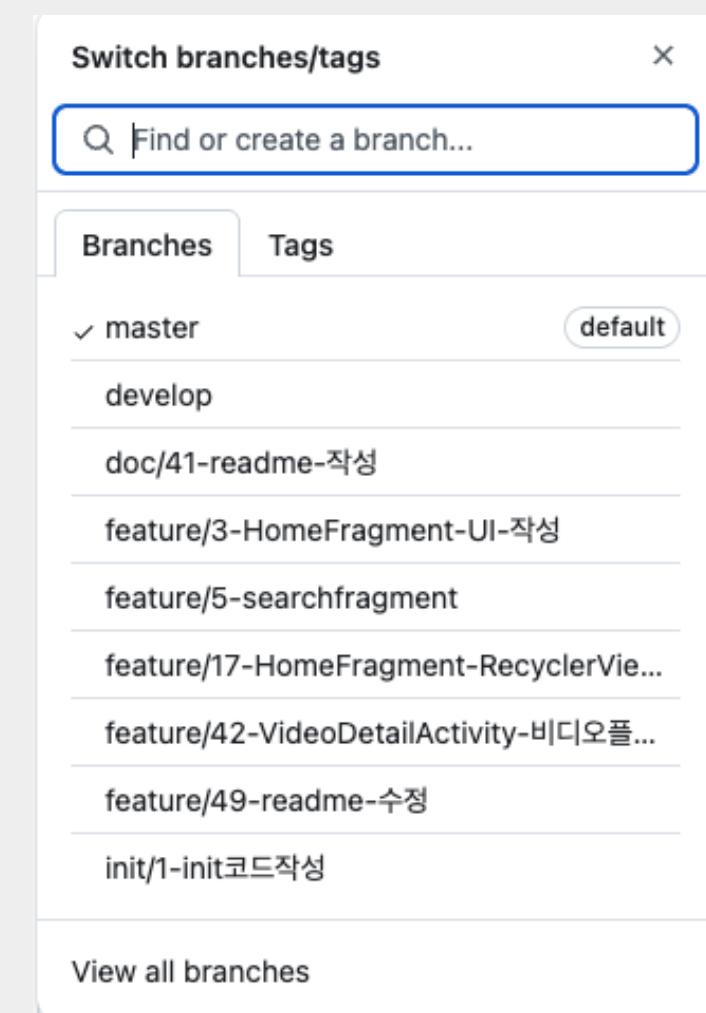
    @Singleton
    @Provides
    fun provideShelterDao(appDatabase: AppDatabase) : ShelterDao = appDatabase.shelterDao()

    private val MIGRATION_1_2 = object : Migration(1, 2){
        override fun migrate(database: SupportSQLiteDatabase){
            // 테이블이 이미 올바르게 존재하는 경우 구조적 변경이 필요하지 않음
            // 테이블을 수정해야 하는 경우 여기에서 수정할 수 있음
        }
    }
}
```

마이그레이션

Git-flow 전략 도입으로, 협업 및 코드관리 최적화

Git-flow를 적용하여 이슈별로 브랜치를 분리하고, 브랜치 이름을 기능에 맞게 설정했습니다.
이를 통해 각 작업을 명확히 정의할 수 있었으며, 관련된 이슈와 기능을 쉽게 파악할 수 있었습니다.
또한 팀원들이 각자의 브랜치에서 독립적으로 작업해 충돌을 최소화하여, 진행 중인 작업을 원활하게 공유할 수 있었습니다.



Branch 분리

<input type="checkbox"/>	<input checked="" type="radio"/> [feat] searchFragment -> VideoDetailActivity 연결
#23	- by kory0115 was closed on May 16, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [feat] MyVideo화면 추가 구현 및 수정
#20	- by yoonchanchoi was closed on May 16, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [fix] RoomDB 수정
#18	- by yoonchanchoi was closed on May 16, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [feat] HomeFragment RecyclerView 구현
#17	- by lany159 was closed on May 17, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [feat] VideoDetailActivity 데이터 적용
#16	- by wndnjs00 was closed on May 16, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [feat] RoomDB, Hilt 적용
#14	- by yoonchanchoi was closed on May 14, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [feat] SearchViewModel
#13	- by kory0115 was closed on May 16, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [feat] MyVideoFragment UI 작성
#6	- by yoonchanchoi was closed on May 14, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [feat] SearchFragment UI 작성
#5	- by kory0115 was closed on May 14, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [feat] MainActivity, VideoDetailActivity UI 작성
#4	- by wndnjs00 was closed on May 14, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [feat] HomeFragment UI 작성
#3	- by lany159 was closed on May 14, 2024
<input type="checkbox"/>	<input checked="" type="radio"/> [init] init 코드 구현
#1	- by wndnjs00 was closed on May 13, 2024

이슈관리



연락처앱

Spring Boot REST API서버로 구현한 연락처 앱

Spring Boot에 대한 기초 지식이 없는 상태에서 해당 프로젝트를 통해 REST API 서버 개발의 기본을 학습했습니다.

Spring MVC 구조 (Controller, Service, Repository)를 활용하여 서버를 구현하였으며, REST API 서버와 Android를 연동한 간단한 연락처 앱을 성공적으로 개발했습니다.

개발기간	2025.01.03 ~ 03.05
개발인원	개인 프로젝트

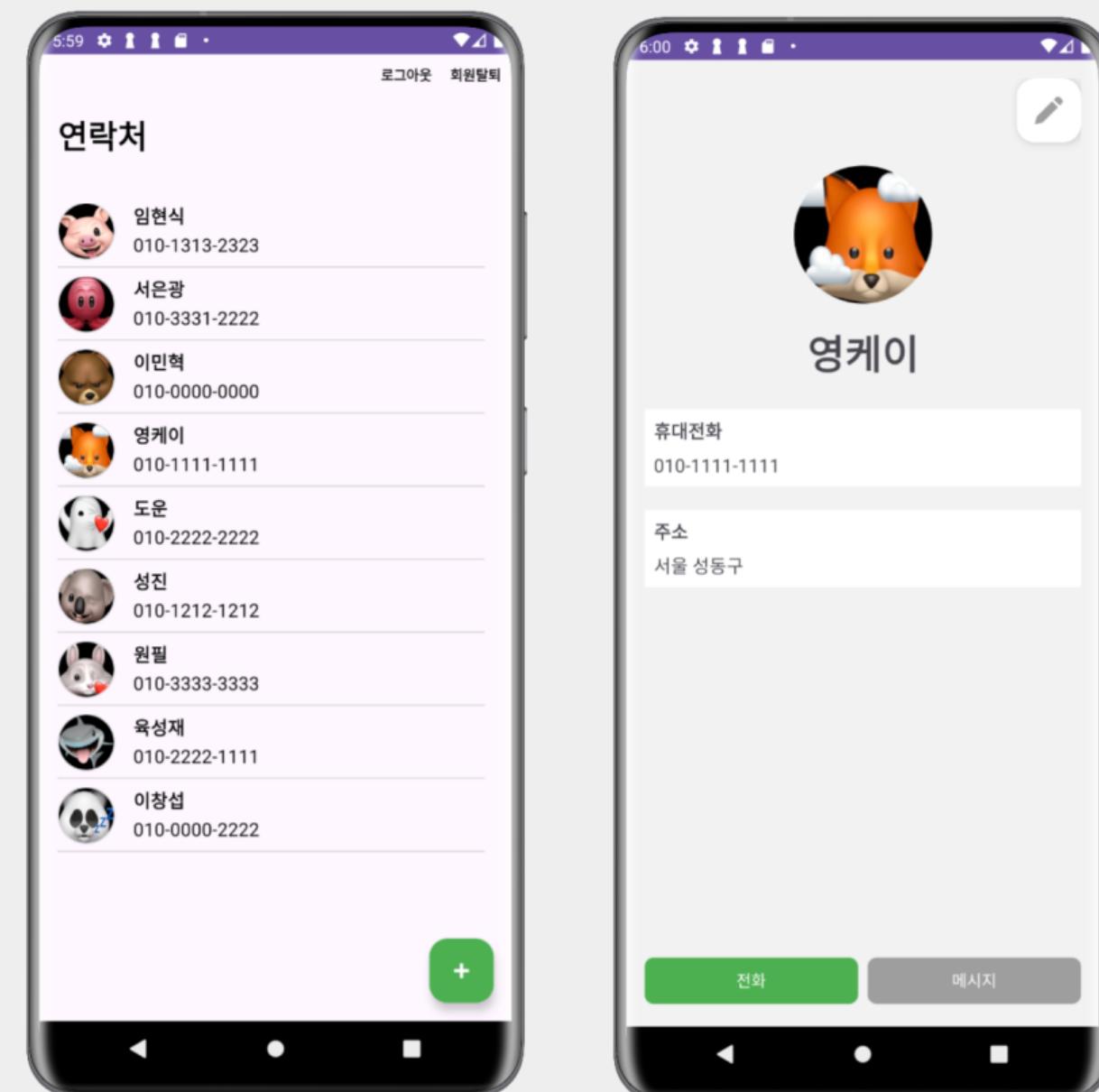

[깃허브 링크 \(Android\)](#)

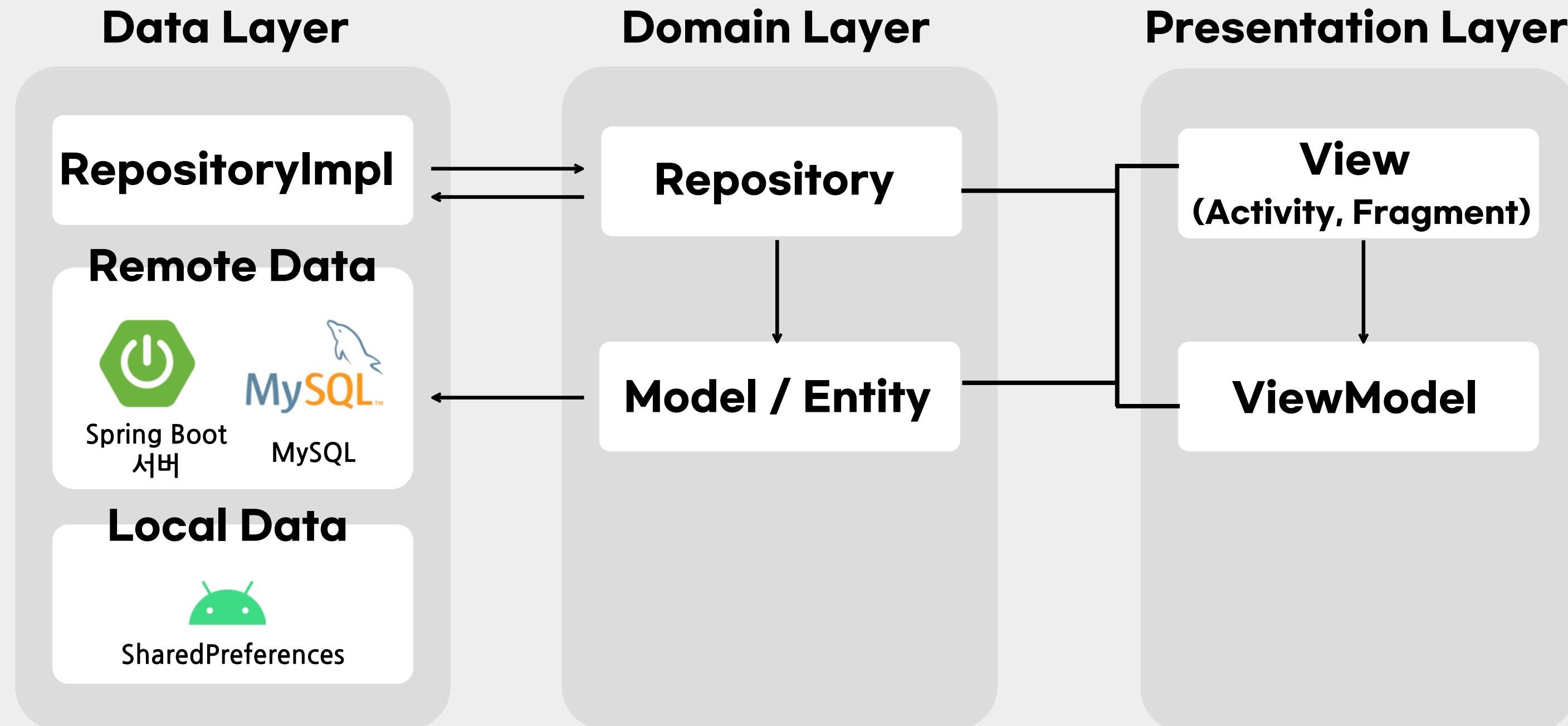
[깃허브 링크 \(Spring Boot Server\)](#)

사용기술

언어	Java
디자인패턴	MVVM + Repository Pattern / Spring MVC
라이브러리 및 프레임워크	Retrofit2, Okhttp3, ViewModel, LiveData, Coroutine, Hilt DiffUtil, XML, Spring Security, JWT
데이터베이스 및 서버	MySQL, JPA, SharedPreferences

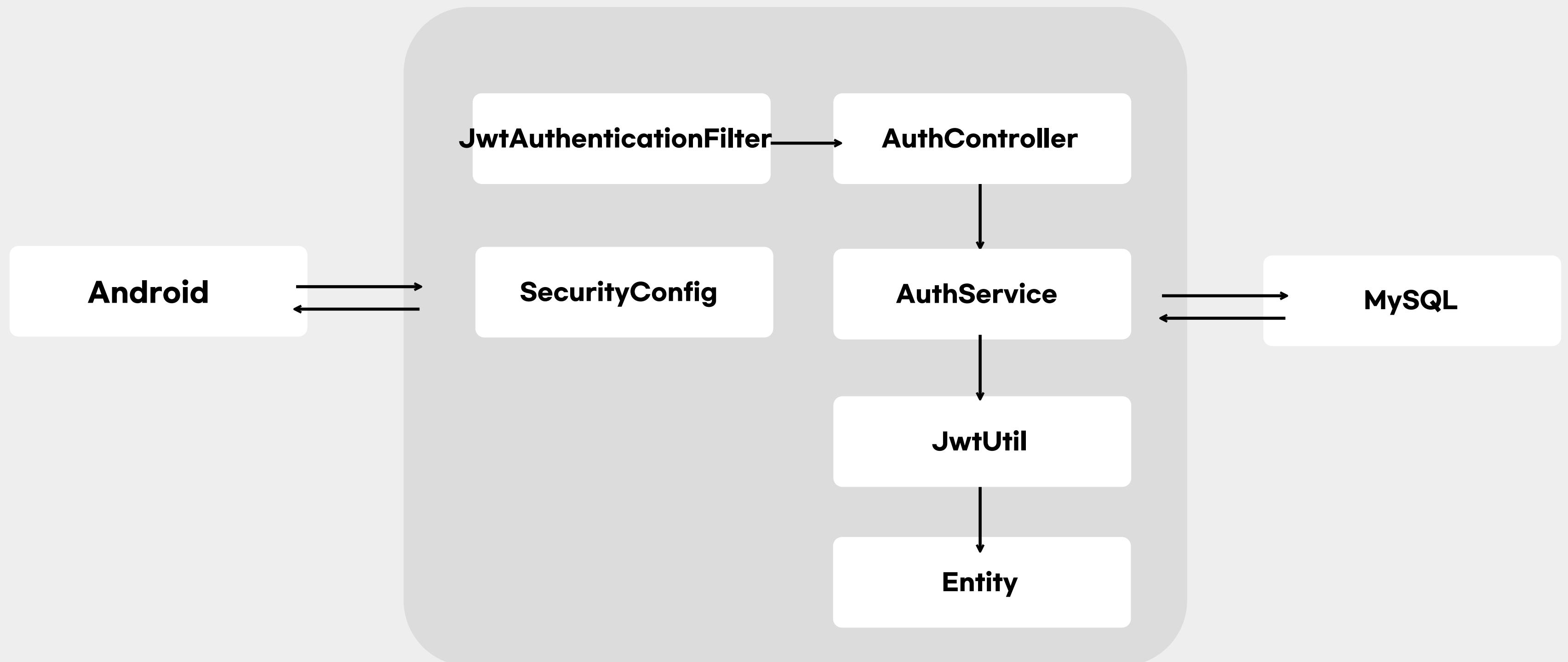
주요기능
회원 연락처 조회, 등록, 수정, 삭제 / 전화&메시지 기능 / 로그인&회원가입&로그아웃&회원탈퇴



**MVVM + Repository Pattern**



Access Token: 30분 유효
Refresh Token: 1주일 유효

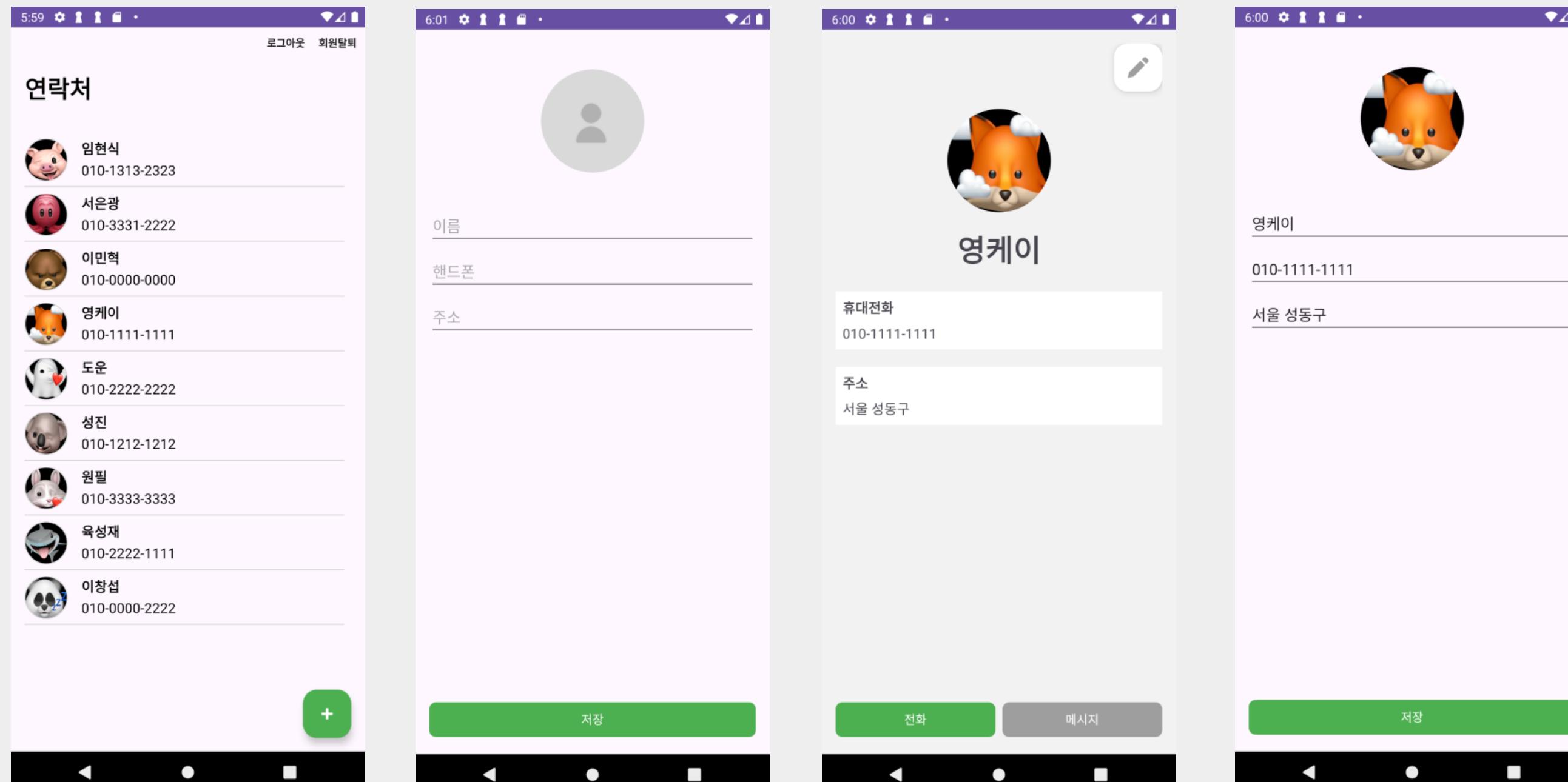




연락처앱

앱 주요기능 (1/2) - 연락처화면

연락처 목록을 확인하고 추가, 수정, 삭제를 할 수 있습니다. 상세페이지에서는 전화, 문자가 가능합니다.
삭제하고 싶은 연락처 목록을 길게 누르면 삭제가 가능합니다.



연락처 목록 확인

연락처 추가

연락처 상세확인

연락처 수정



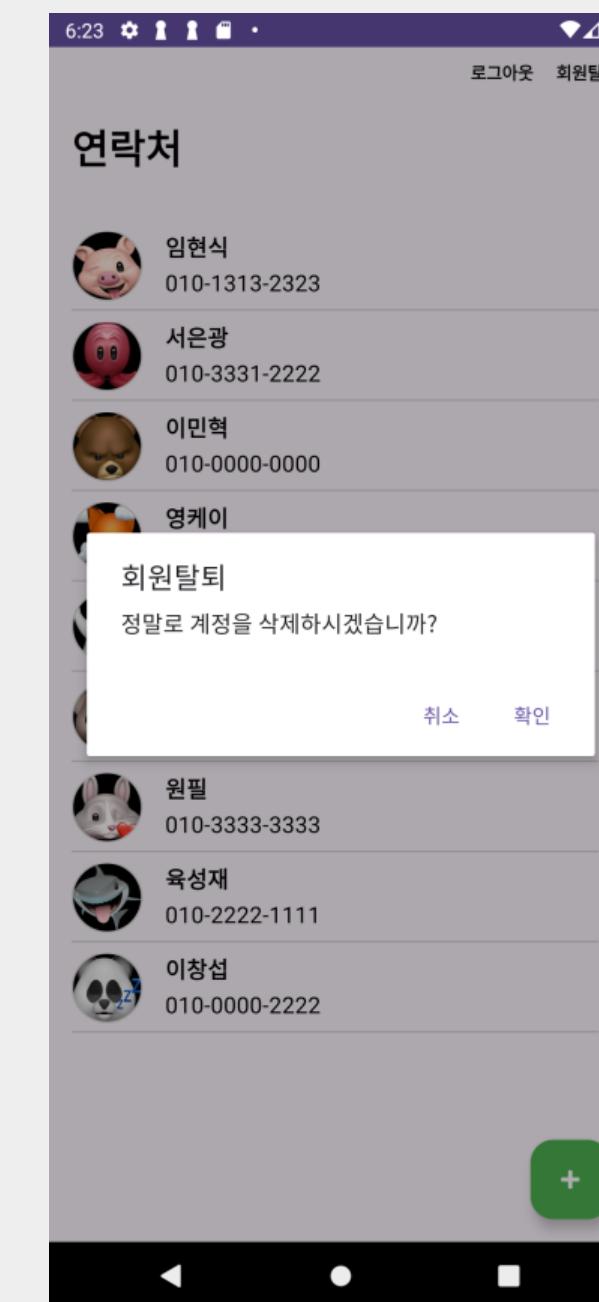
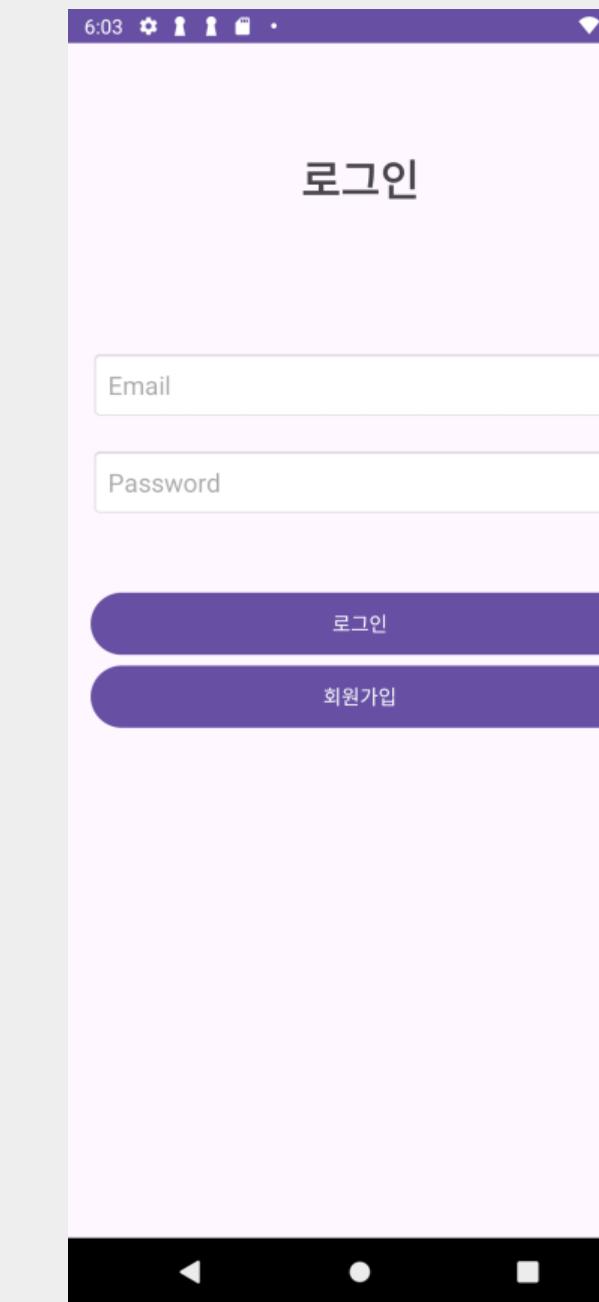
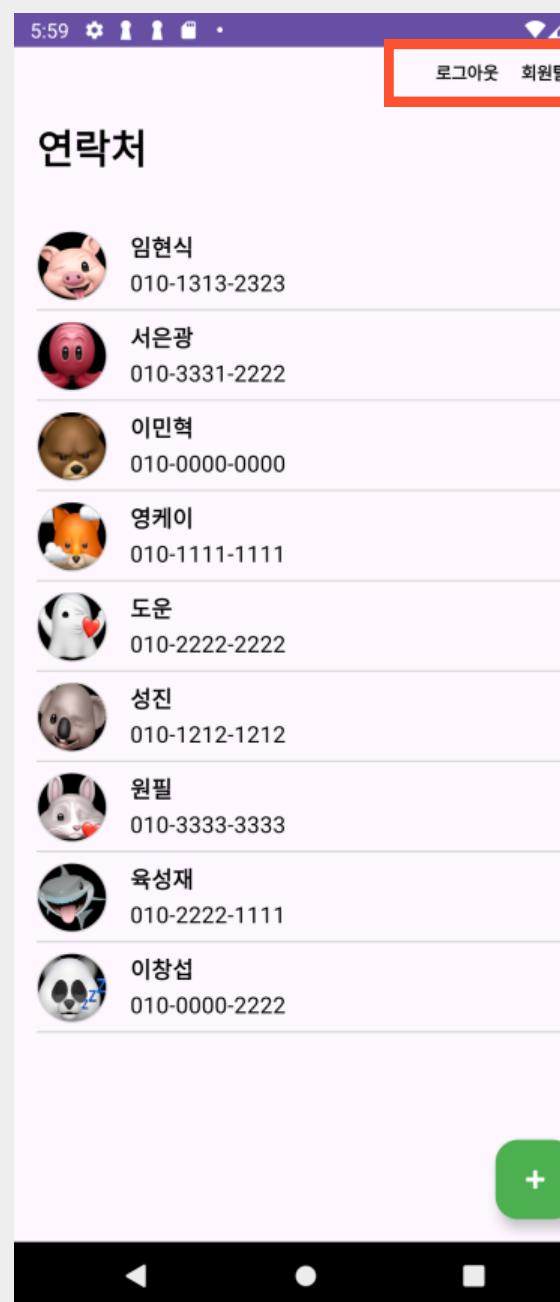
연락처앱

앱 주요기능 (2/2) - 로그아웃, 회원탈퇴

연락처 목록에서 로그아웃, 회원탈퇴가 가능합니다.

로그아웃을 하면 로그인화면으로 돌아가며, 재로그인을 해야합니다. 회원탈퇴시에는 서버에 있는 회원의 정보가 삭제됩니다.

Access Token이 10분뒤에 만료되며, Refresh Token이 발급됩니다. 일주일뒤에 Refresh Token이 만료되면 로그아웃이 되어 보안성이 강화됩니다.



로그아웃 했을때

회원탈퇴

감사합니다!