



개발자 포트폴리오

E-mail

|

TEL

|

Github

|

최



목 차

1

보유 기술

| Native App, Web, Back-end, 형상관리

2

안드로이드 앱 1: 내 노래방책

| 개요, 기획의도, 화면소개, 개발 중 고민한 점

3

안드로이드 앱 2: 디코챗

| 개요, 기획의도, 화면소개, 개발 중 고민한 점

보유 기술

<h2>Android Native App</h2>	<h3><u>Android</u></h3> <ul style="list-style-type: none"> Android 4대 컴포넌트의 특성 이해 Lifecycle을 고려한 Callback 활용 RecyclerView로 데이터 표시 BottomNavigationView를 이용한 Fragment 간 화면 전환 카카오, 네이버, 구글의 지도 및 로그인 API 접근 및 활용 공공 API 접근 및 활용 Retrofit을 이용한 HTTP통신 및 JSON parsing Push Notification 활용 ResultLauncher를 이용한 동적 Permission 요청 앱 기획, 개발, PlayStore 출시, 유지보수 등 앱 개발 전체 과정 진행 	<h3><u>Kotlin & Java</u></h3> <ul style="list-style-type: none"> Collection Framework 활용 익명클래스를 이용한 인터페이스 구현 람다 표현식과 SAM 변환기술 적용 Thread 활용 및 비동기 처리 구현 예외 처리 I/O Stream을 활용한 파일 입출력
<h2>Web</h2>	<h3><u>HTML5 + CSS3</u></h3> <ul style="list-style-type: none"> MediaQuery와 Flex스타일의 반응형 웹 구현 Bootstrap 라이브러리 활용 	<h3><u>JavaScript</u></h3> <ul style="list-style-type: none"> ES6 스타일 코딩 (let, class, arrow function, module) AJAX 기술을 이용한 비동기 통신 DOM, BOM 트리를 활용한 요소 제어 JSON 파싱
<h2>Back-end</h2>	<h3><u>Apache-PHP-MySQL 웹 서버</u></h3> <ul style="list-style-type: none"> GET / POST request 데이터의 저장과 처리 서버DB의 스키마 설계 및 CRUD 작업 JSON, HTML 형식으로 response 출력 	<h3><u>Firebase</u></h3> <ul style="list-style-type: none"> Cloud Storage을 이용한 파일 업로드 및 다운로드 Firestore를 통한 NoSQL DB 활용 Cloud Messaging으로 푸시 서비스 이용
<h2>형상관리</h2>	<h3><u>Git</u></h3> <ul style="list-style-type: none"> Branch별 소스코드 버전 관리 및 팀 프로젝트 적용 	



프로젝트 기술서 (Native App)

1. 내 노래방책

- 개발기간: 2022.10.24. ~ 12.2. (6주)
- 개발인원: 1명

주요 기능

- Open API에서 Retrofit으로 노래방 반주곡 JSON 데이터를 받아와서 RecyclerView에 표시
- 노래방 웹사이트를 JSoup로 크롤링하여 RecyclerView에 표시
- 기기 내 SQLite DB에 곡 북마크 저장
- Shared Preference에 설정 저장
- Naver Maps, Naver Search API를 통해 현재 위치와 가장 가까운 노래방 위치를 지도에 표시

개발 환경

OS	Windows 10	언어	Kotlin
개발 툴	Android Studio	DBMS	SQLite

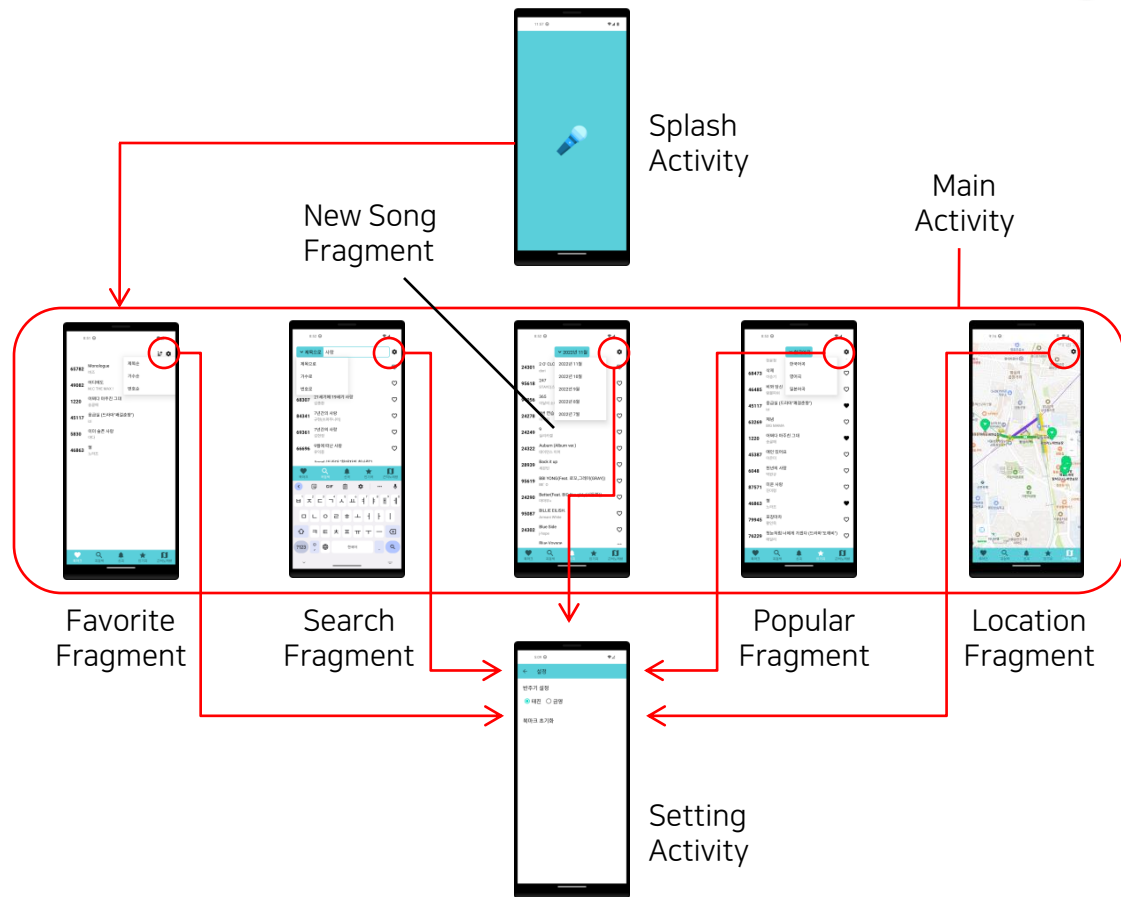


와이어프레임



기획 의도

현재 플레이스토어에는 노래방 반주기에 등록된 노래 검색기능과 즐겨찾기 등록 기능을 제공하는 앱들이 이미 많이 올라와 있다. 그러나 기존의 앱들에는 즐겨찾기 등록된 곡에 메모를 남기는 기능, 근처 노래방 검색 기능 등 몇몇 기능이 부족하다고 느꼈다. 그래서 이런 기능을 보완한 앱을 새로 만들어 보고자 이 기획을 시작하였다.



스플래시 & 바텀 네비게이션

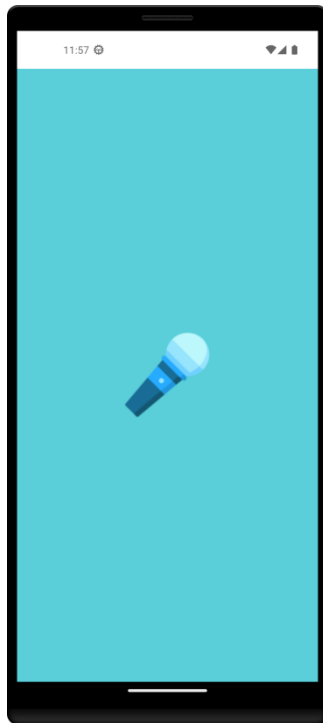


Shared
Preference

저장된
유저 설정
불러오기

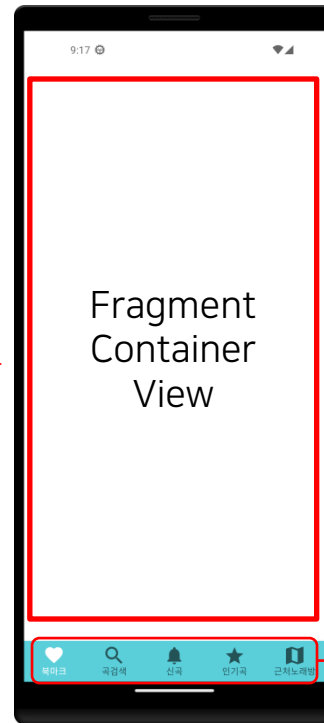


앱 작동시
첫 화면



Splash Activity

Handler를 통해
1.5초 후 Main
Activity로 화면 전환



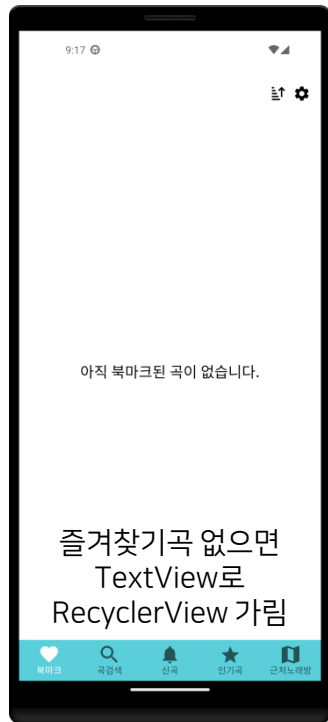
Main Activity

Fragment
Container
View

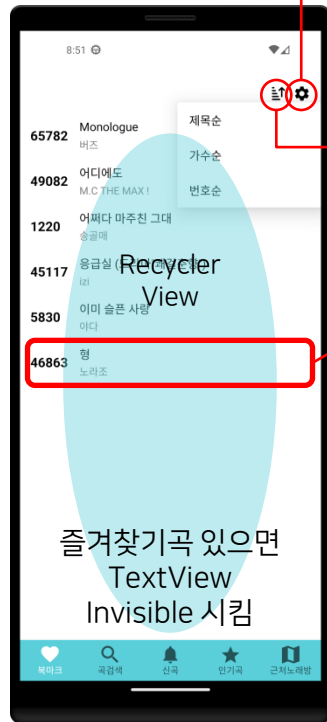
BottomNavigationView에
OnItemSelectedListener
붙여서 각 프래그먼트를
show() / hide()

Bottom
Navigation View

곡 즐겨찾기 화면



Favorite Fragment

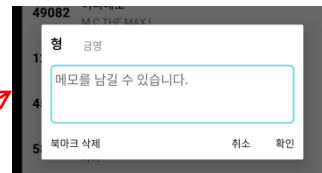


PopupMenu로
리스트 정렬
기준 선택

RecyclerView

46863 형 노래조

즐거찾기곡 있으면
TextView
Invisible 시킴



아이템 클릭시 메모 작성
다이얼로그 표시



DB에서
즐거찾기곡
SELECT

데이터베이스: favorite

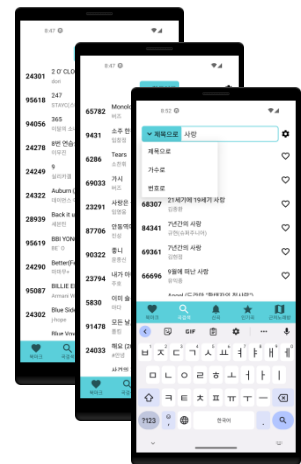
id	brand	_no	제목	singer	_release	memo
1	kumyoung	65782	Monologue	버즈	NULL	NULL
2	kumyoung	5830	이미 슬픈 사람	아다	NULL	NULL
3	kumyoung	49082	어디에도	M.C THE MAXI	NULL	NULL
4	kumyoung	45117	응급실 (도라지'액절충량)	izi	NULL	NULL
5	kumyoung	1220	어쩌다 마주친 그대	송골매	NULL	NULL
6	kumyoung	46863	형	노라조	NULL	NULL

*DB Browser

화면 전환



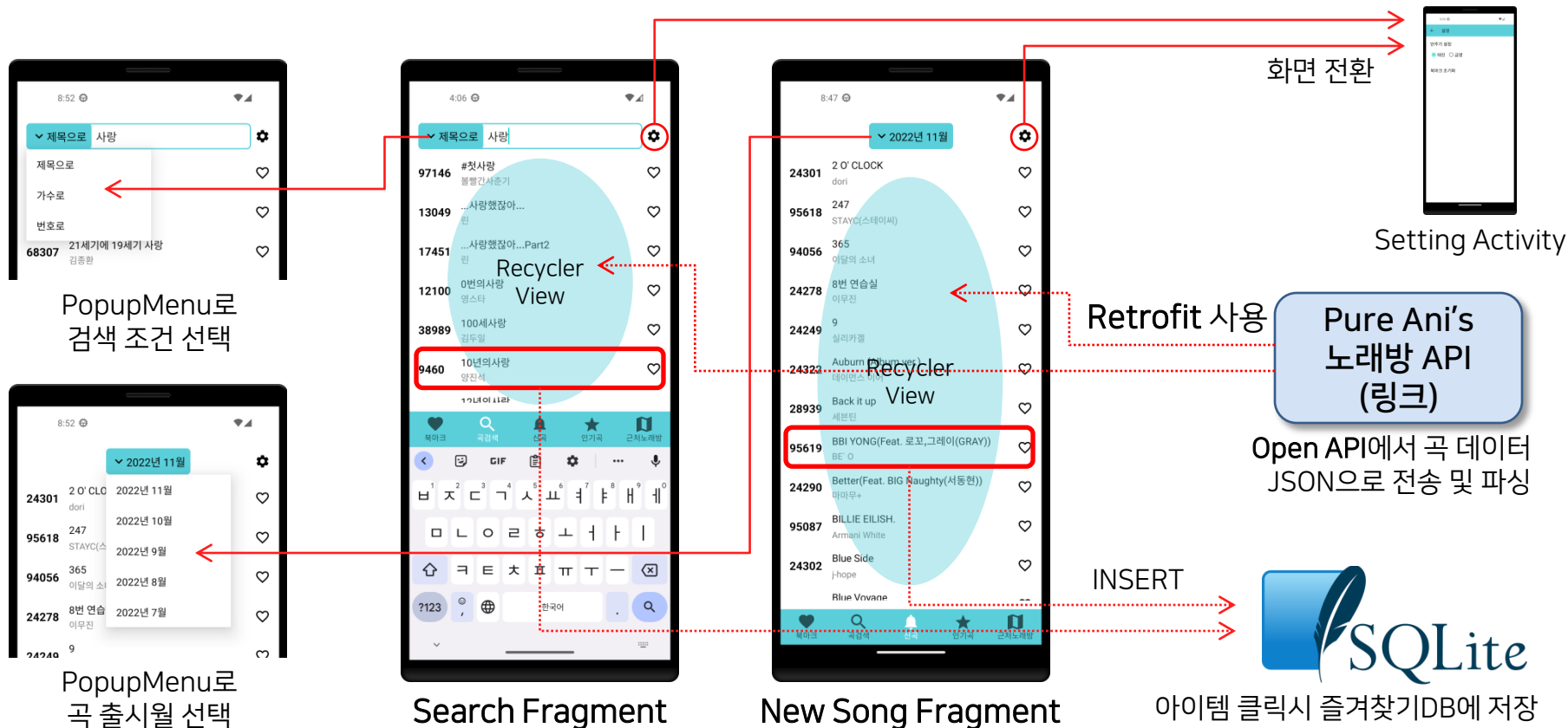
Setting Activity



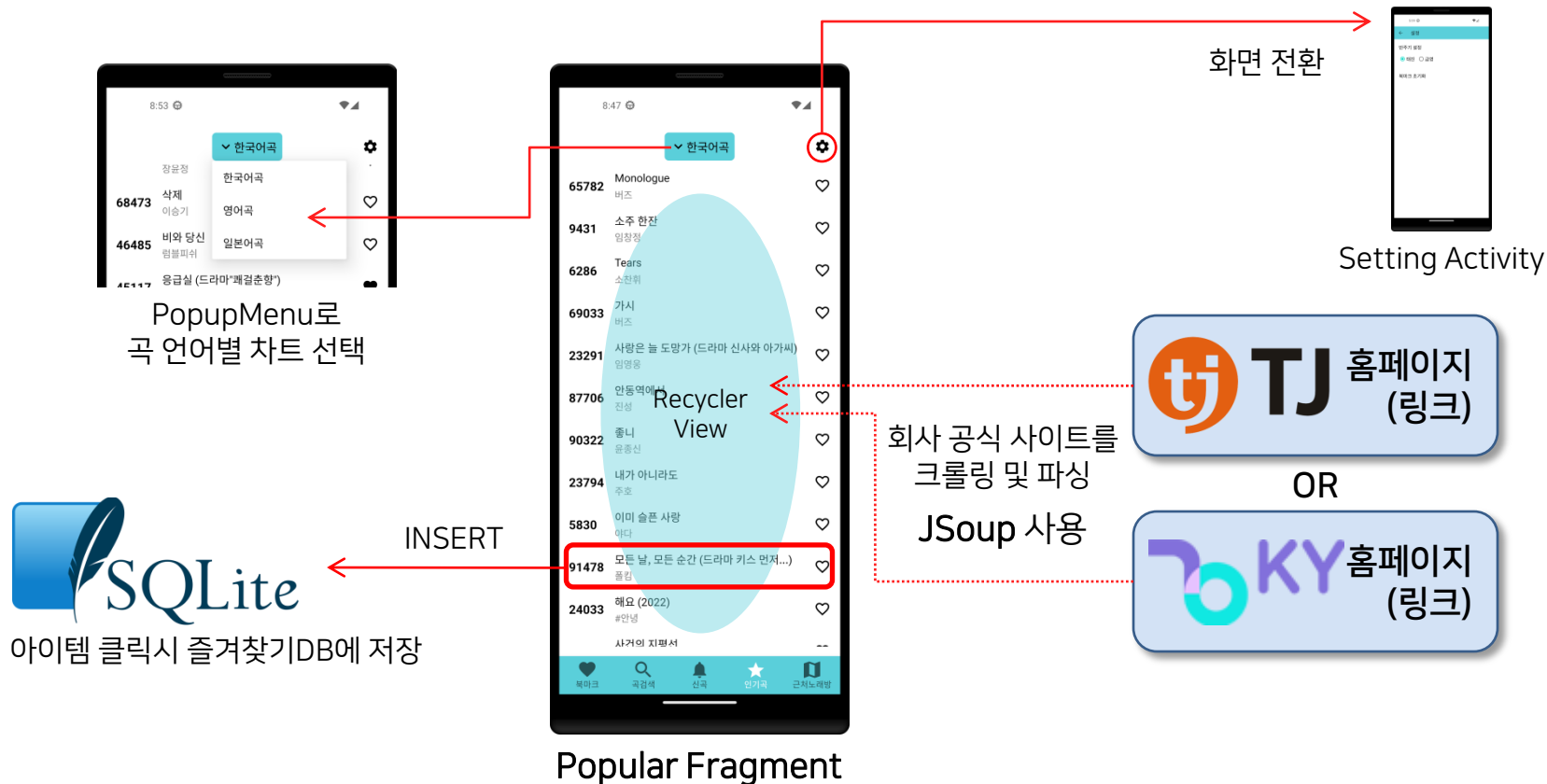
다른 프래그먼트에서
즐거찾기곡을 DB에 저장

INSERT

곡 검색 화면 & 신곡 조회 화면



인기곡 차트 조회 화면



근처 노래방 검색 화면 & 설정 화면



기기 위치정보:
Google Play
Location Service

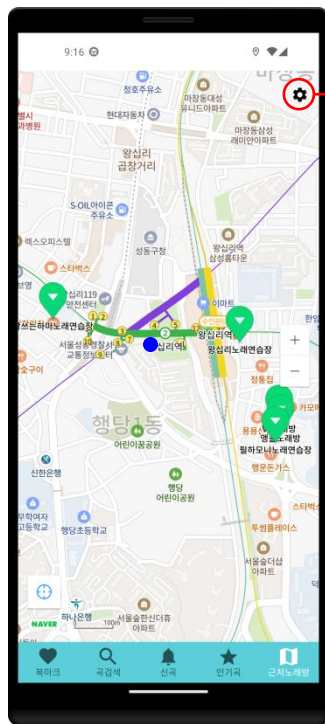


지도:
Naver Maps API



노래방 위치정보:
Naver Search API

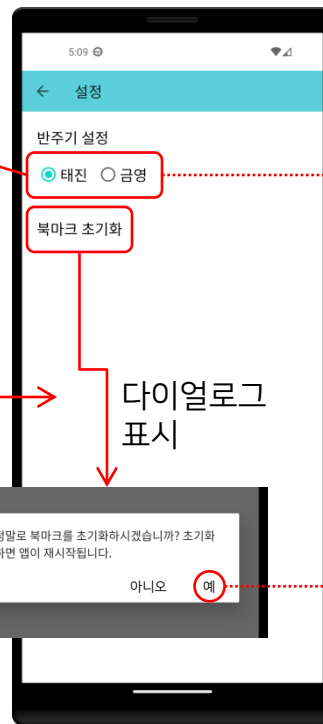
기기의 위치정보를 받아서
지도에 현재 위치와 가장
가까운 노래방 위치를 표시



Location Fragment

탭 레벨 프로퍼티로
저장해서 모든
검색결과에 반영

화면 전환



Setting Activity

설정값
저장



Shared
Preference

다이얼로그
표시

DROP



DB상의 테이블 지우고
앱 재시작

(다시 DB에 접근시
테이블 자동 재생성)



개발 과정에서 고민한 점

• 프래그먼트간 전환시 show() / hide()를 쓴 이유

처음엔 JetPack Navigation을 써서 코드를 간략화하려고 했다. 그러나 이를 적용하는 과정에서 해당 라이브러리가 프래그먼트 전환시 replace()를 한다는 것을 알게 되었다. 프래그먼트를 넘나들 때마다 각 화면의 검색 결과가 초기화된다면 유저 입장에서 불편할 거라고 생각해서, Bottom Navigation View에 리스너를 달아 show() / hide() 명령을 내리는 방식으로 코딩을 다시 했다.

• 개인 API와 공식사이트 데이터를 섞어 쓴 이유

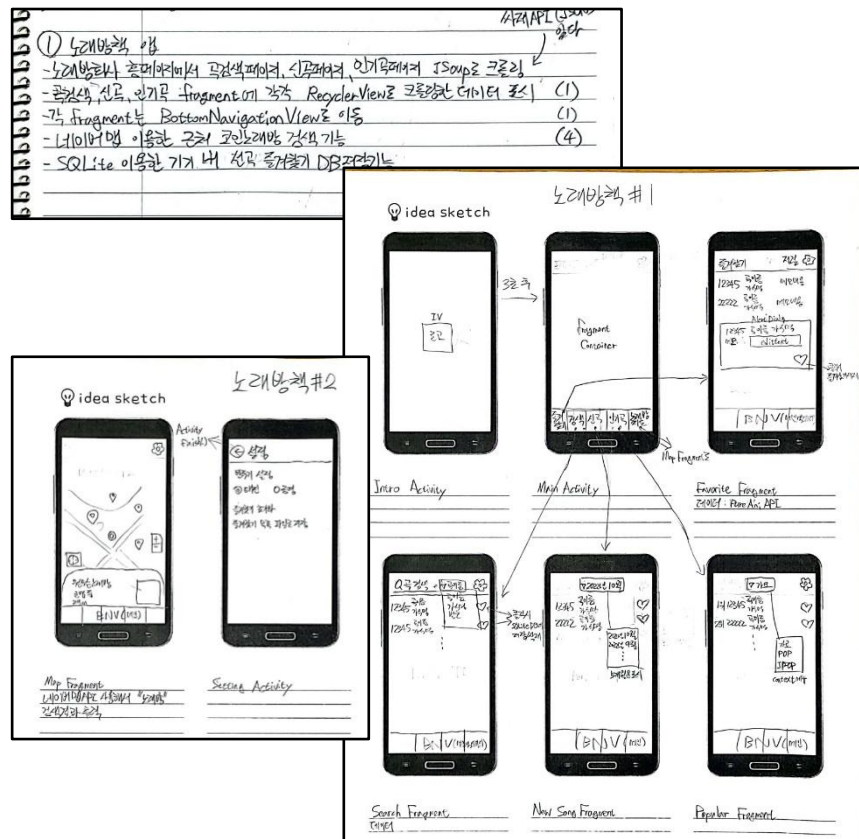
개인 API가 잘 만들어져 있어서 처음엔 곡 검색, 신곡 조회, 인기곡 조회 모두 개인 API에서 데이터를 받아오려고 했다. 그러나 개발 진행 중 알아보니 개인 API는 인기곡 조회를 지원하지 않았고, 어쩔 수 없이 이 기능만 공식사이트를 크롤링하는 것으로 변경했다.

• Retrofit과 JSoup을 쓴 이유

대표적인 HTTP Client Library로 Retrofit(+OkHttp)과 Volley가 있다고 알고 있다. 그런데 Volley는 이미 deprecated 됐다는 이야기도 있었고, 무엇보다도 Retrofit이 훨씬 속도가 빠르다고 하여 Retrofit을 선택했다. Retrofit을 쓰고 있으니 HTML 파싱도 컨버터만 추가해서 하는 쪽이 더 편했을 수도 있지만, JSoup을 써본 적이 없어서 한번 공부해보자 하는 마음으로 JSoup을 이용했다.

• 네이버 맵, 네이버 검색 API를 쓴 이유

구글, 카카오, 네이버 중 어떤 회사 API를 사용해도 결과가 크게 다르지 않을 것 같았고, 학원 실습 때 구글과 카카오 API의 사용법은 배웠지만 네이버 API는 자칫 잘못하면 이용 요금이 과다 청구될 위험이 있다고 하여 배운 적이 없었다. 그래서 이번 기회에 연습을 해보고자 네이버 API를 사용했다.





프로젝트 기술서 (Native App)

2. 디코챗

- 개발기간: 2022.10.24. ~ 12.2. (6주)
- 개발인원: 1명

주요 기능

- 웹호스팅 서버를 통한 채팅, 유저 등록, 친구 추가, 채팅방 개설 및 초대, 멘션 기능
- Firebase Cloud Message 푸시 서버 및 Service 컴포넌트를 활용한 실시간 채팅 구현
- Google, Kakao 로그인 API를 통한 간편로그인
- 갤러리 앱의 ContentProvider 컴포넌트에 접근하여 사진 업로드

개발 환경

OS	Windows 10	언어	Kotlin, PHP
개발 툴	Android Studio Visual Studio Code	DBMS	MySQL



링크



링크

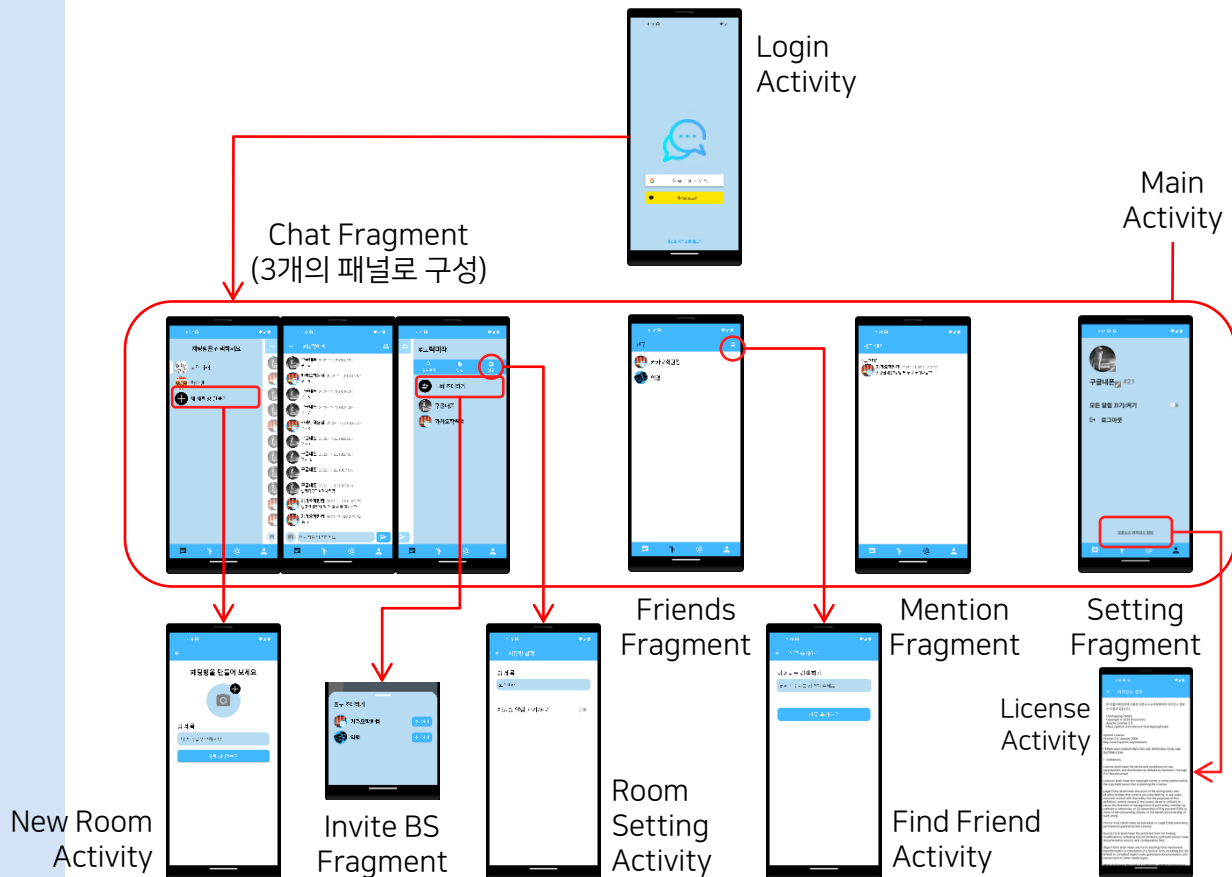
와이어프레임



기획 의도

10월 카카오 서버 화재 사건 때, 학원에서 배운 채팅 앱 예제를 친구들에게 배포해서 카카오톡 대신 이용했던 경험 있다.

이 경험을 통해 채팅 앱이 얼마나 사회에 중요한 역할을 하는가를 깨달았고, 이에 유명한 채팅 앱 '디스코드'의 레이아웃을 벤치마킹하여 채팅 앱을 한번 만들어 보자는 기획을 시작하게 되었다.



웹서버 정보



[닷홈 무료 호스팅 이용 \(링크\)](#)

- Apache 2.4
- PHP 7.4
- MySQL 8.0

*서버와의 HTTP통신에는
Retrofit 라이브러리 사용

웹서버 DB 스키마



유저 테이블

maratangsoft dicochat_users
user_no : int
google_id : varchar(30)
kakao_id : varchar(30)
fcm_token : text
nickname : text
user_img : varchar(80)

친구 테이블

maratangsoft dicochat_friend
user_no : int
friend_no : int

채팅 테이블

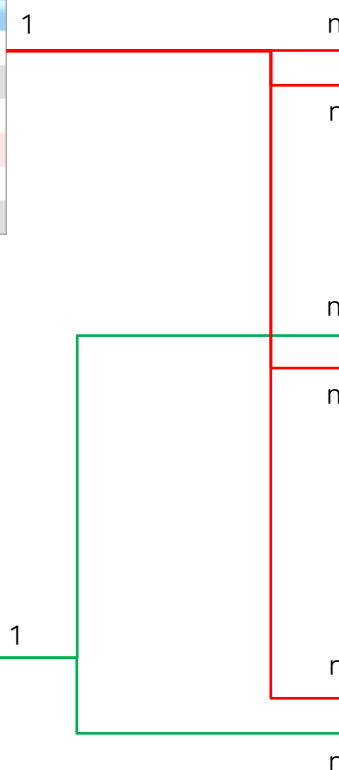
maratangsoft dicochat_chats
chat_no : int
room_no : int
user_no : int
message : text
file_url : varchar(80)
mentioned : int
write_date : datetime

채팅방 테이블

maratangsoft dicochat_rooms
room_no : int
room_title : varchar(30)
room_img : varchar(80)
create_date : datetime

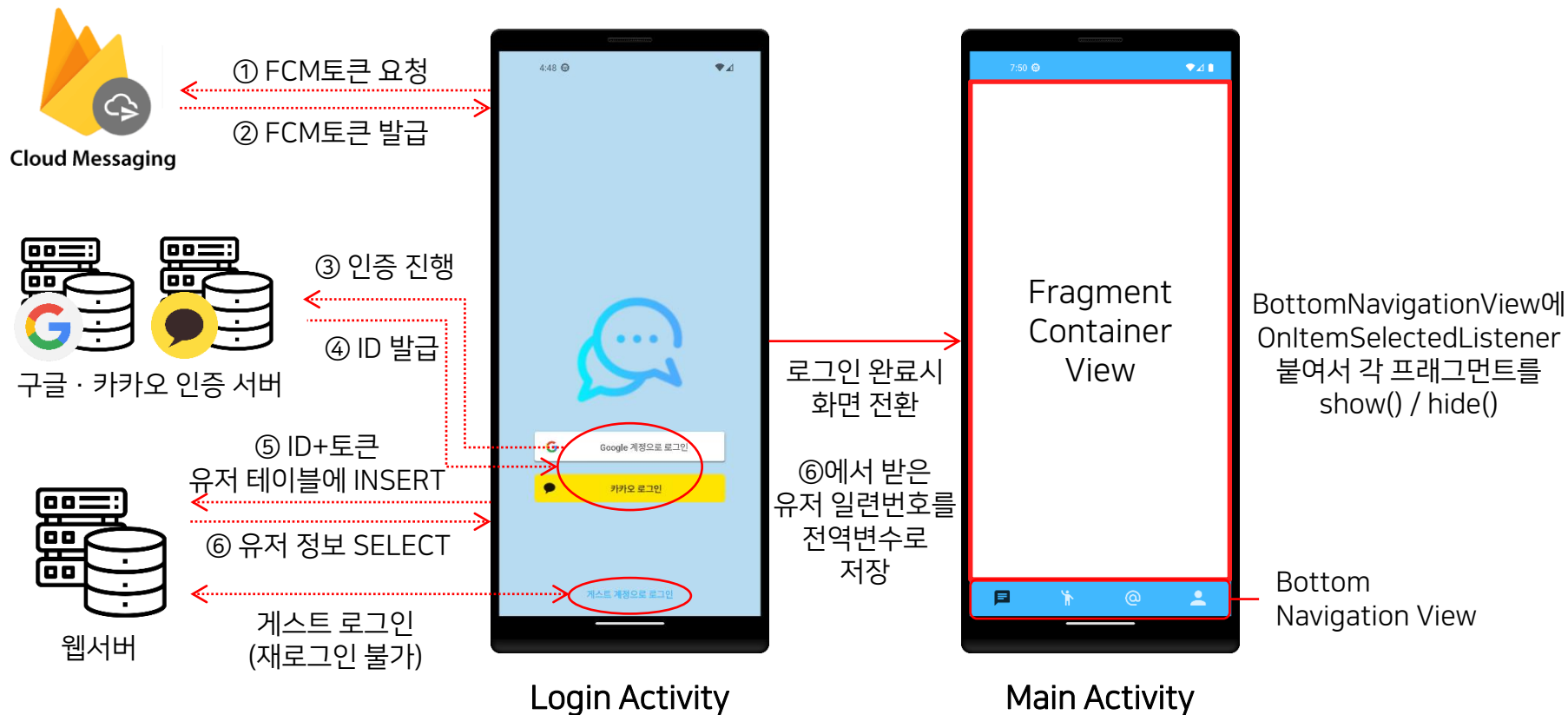
채팅방 멤버 테이블

maratangsoft dicochat_room_member
user_no : int
room_no : int
enter_date : datetime





로그인 화면 & 바텀 내비게이션



Chat Fragment 패널 구조



왼쪽 패널
(채팅방 목록, 첫 화면)

중앙 패널
(채팅 화면)

오른쪽 패널
(채팅방 정보)

 **Overlapping Panels**

디스코드에서 Github에
공개한 레이아웃
라이브러리 사용 (링크)

Apache License 2.0



채팅방 목록 화면 & 채팅방 생성 화면

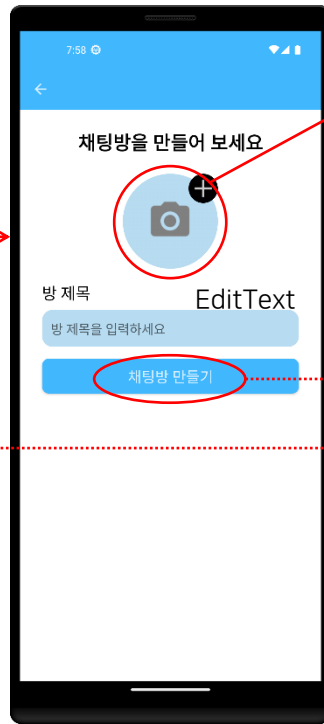
채팅방 선택해야
채팅 화면, 채팅방
정보 화면으로
이동 가능

채팅방 선택시
해당 채팅방 ID를
전역변수로 저장



Chat Fragment
(왼쪽 패널)

클릭시
화면 전환



New Room Activity

Intent로 갤러리 앱
연결해서 사진 선택

사진, 방 제목
선택 후 버튼 클릭시
DB 채팅방 테이블에
INSERT

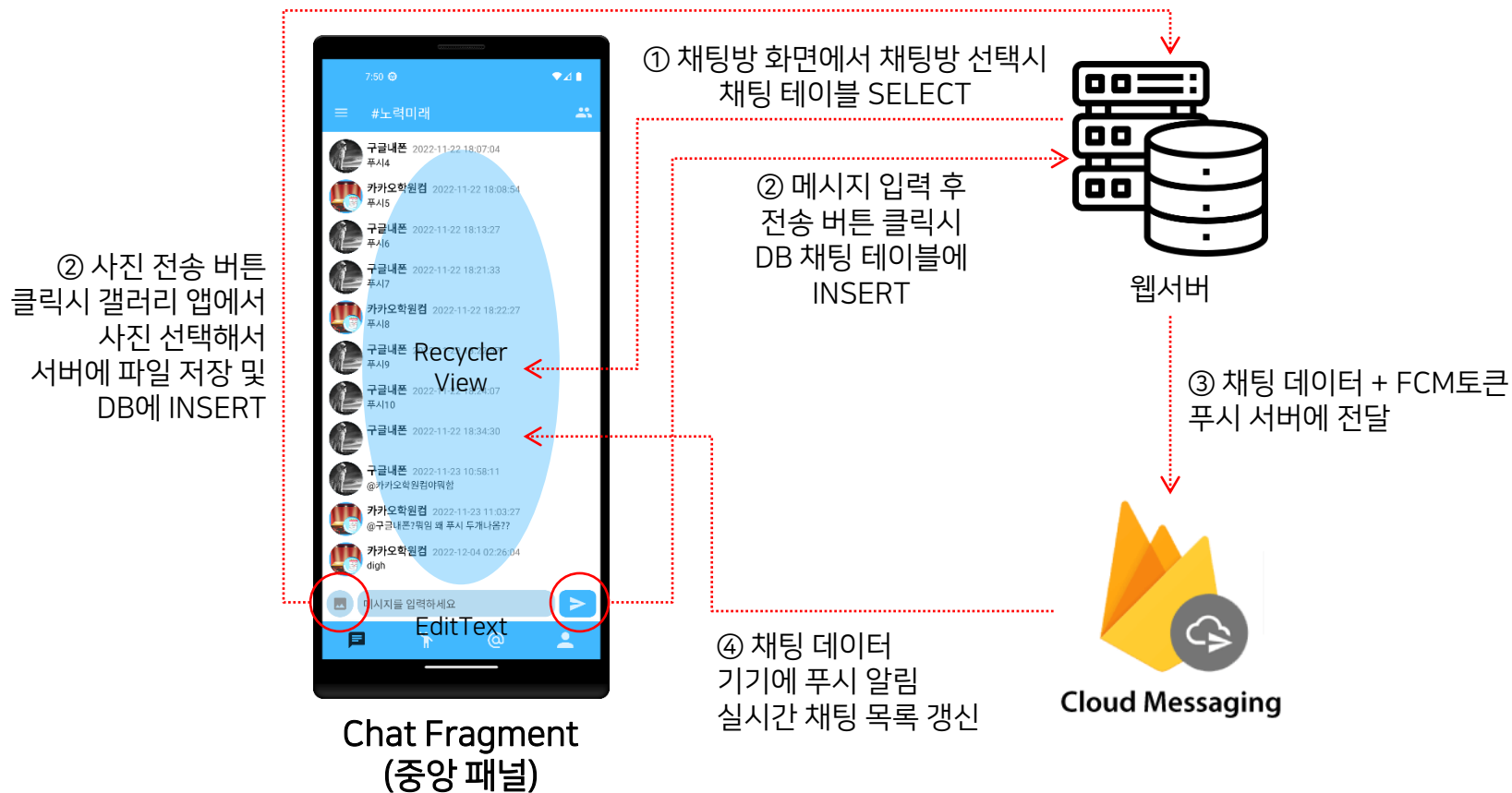
채팅방 테이블 SELECT
후 JSON 전송 및 파싱



웹서버

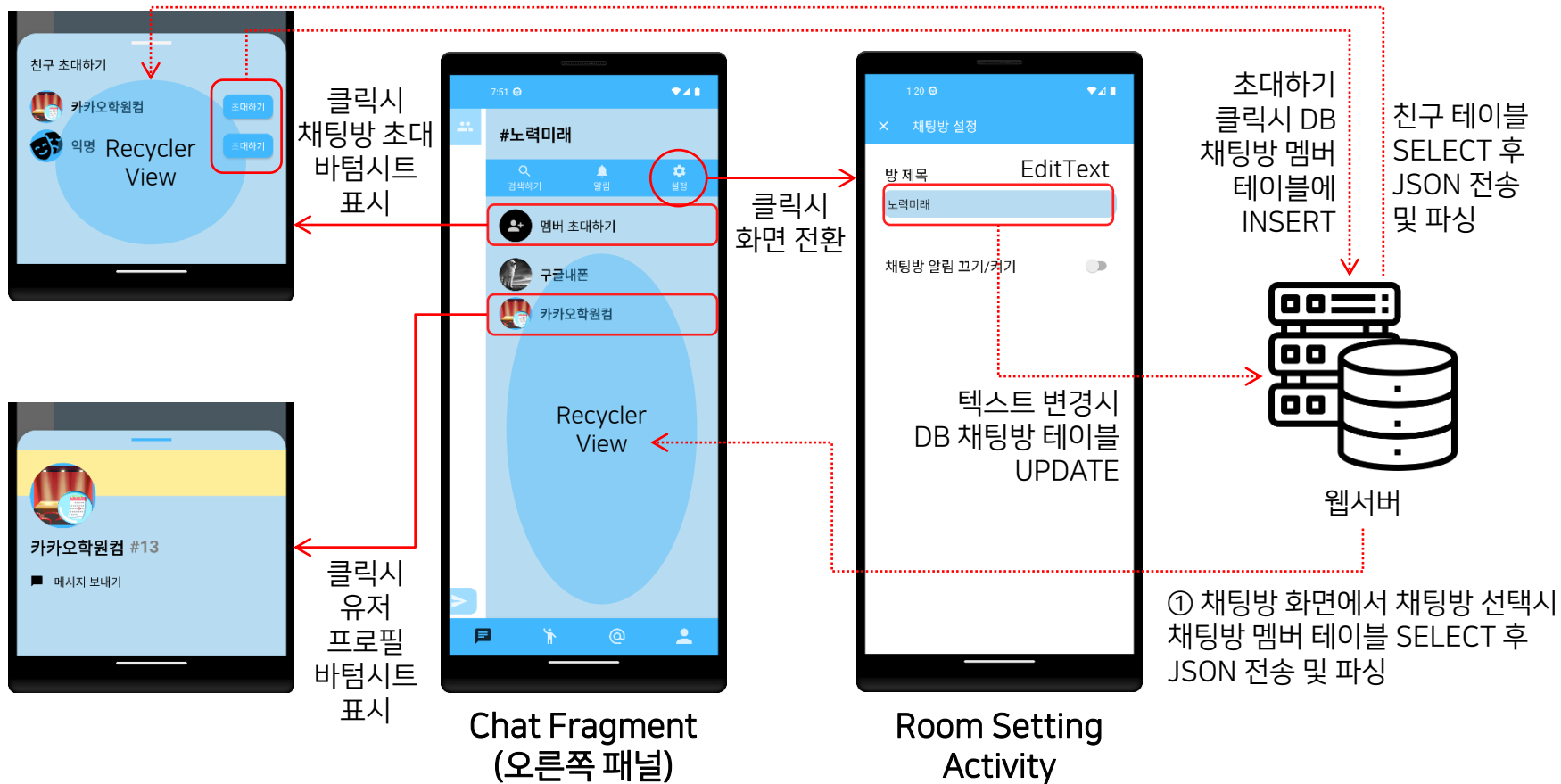


채팅 화면



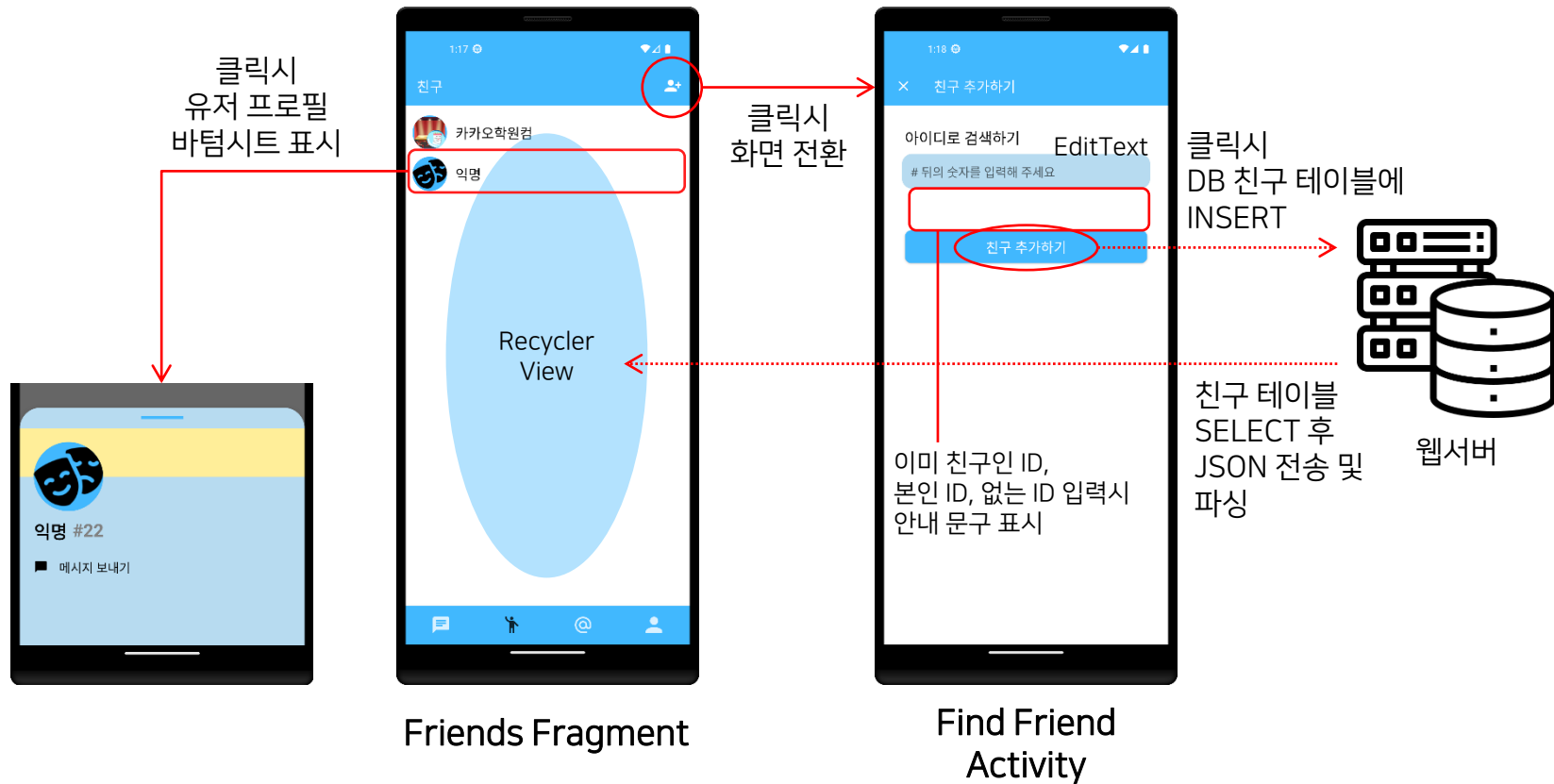


채팅방 정보 화면 & 채팅방 설정 화면



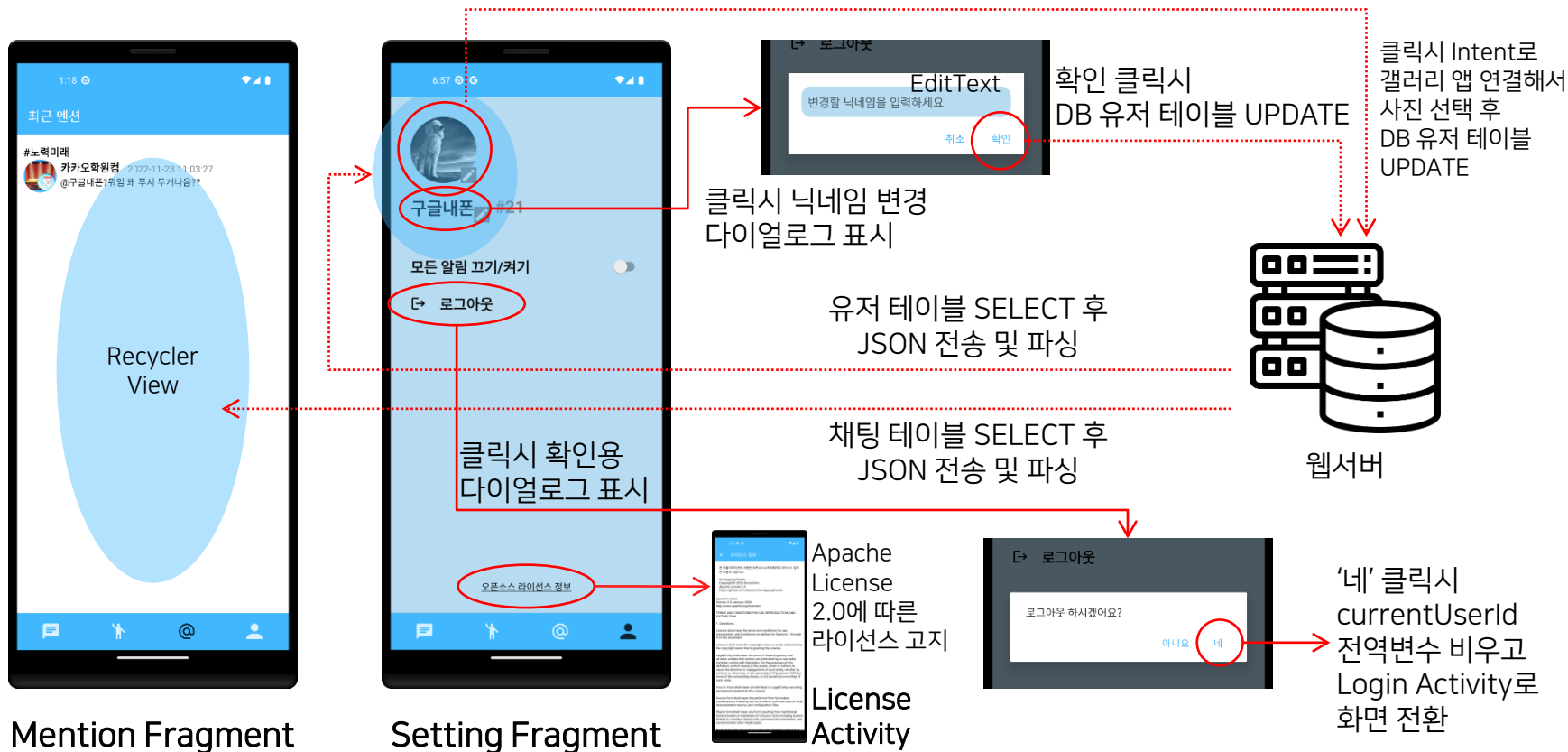


친구 목록 화면 & 친구 추가 화면





멘션 목록 화면 & 프로필 설정 화면



[illegible]