

PORTFOLIO

박진솔

Android Developer

GitHub : <https://github.com/jjeansp9>

Tel : 010-2074-3853

E-Mail : jjeansp9@gmail.com



목차



1. 기술소개

Android

Kotlin, Java, XML

C/C++

HTML5/CSS, JavaScript

PHP, MySQL, Git/GitHub

2. Native App

TRAVIN - 여행 정보공유를 위한 SNS

1인 개발

Login REST API 활용 - Kakao

DB 활용 - PHP, MySQL, Firebase

Activity에 Fragment Layout 구현

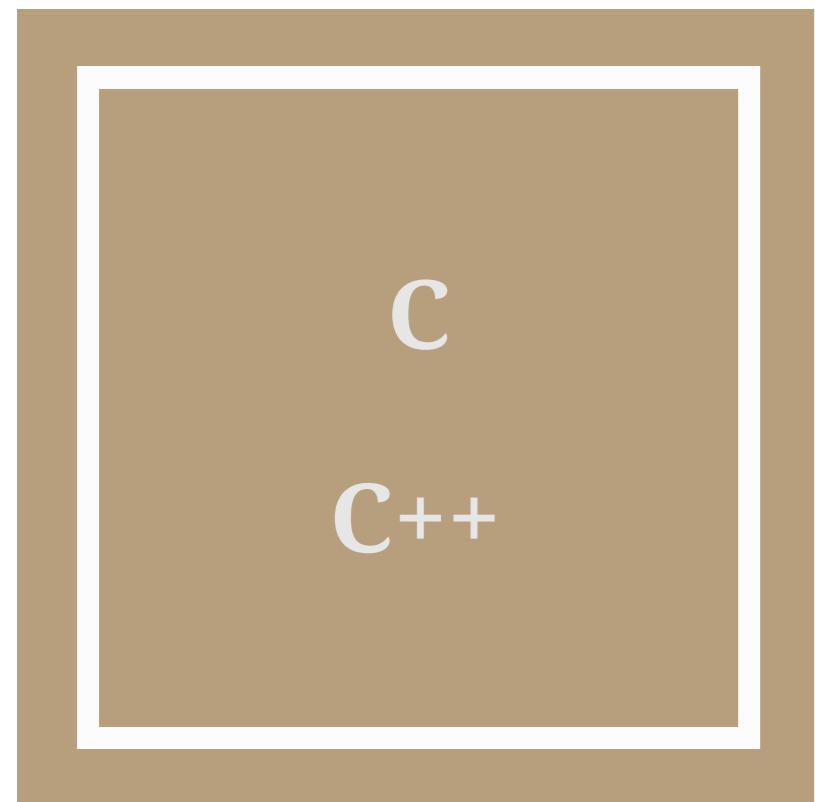
기술소개



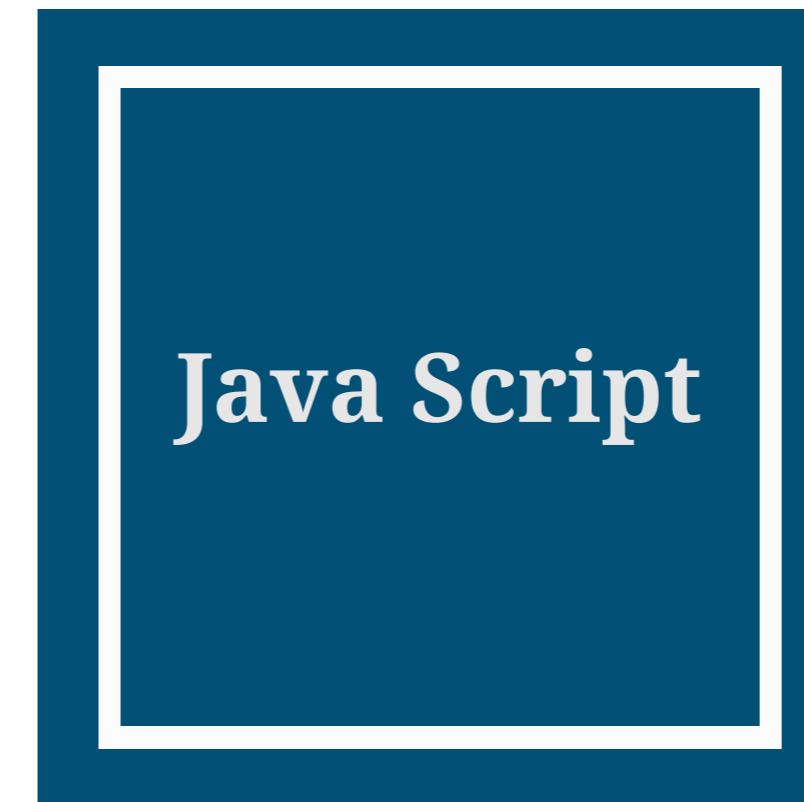
Kotlin 문법 및 함수형 프로그래밍 기법 이해 및 활용
상속 관계 클래스의 함수, 변수 오버라이딩 학습
Java 문법 및 객체지향 프로그래밍 기법 이해 및 활용
XML을 이용해 앱의 화면 Layout 구성



HTML5의 기본태그 사용방법 숙지 및 활용
HTML5를 이용해 웹브라우저의 Layout 구성
CSS의 Style 태그를 이용하여 웹사이트 디자인
미디어 쿼리를 이용하여 화면 크기별 리사이즈 구현



절차지향 프로그래밍 기법을 이해하고 숙지
다중 상속 및 객체지향 프로그래밍 기법 활용
알고리즘에 대한 이해 및 다양한 연산자 등 프로그래밍 기본 문법



JavaScript를 이용하여 웹사이트의 기능 구현
ajax통신을 통해 자바스크립트 비동기 처리
화살표 함수 표기법을 통한 코드 간소화방법 이해 및 숙지
ECMAScript 6버전에 있는 let, const 변수키워드 숙지

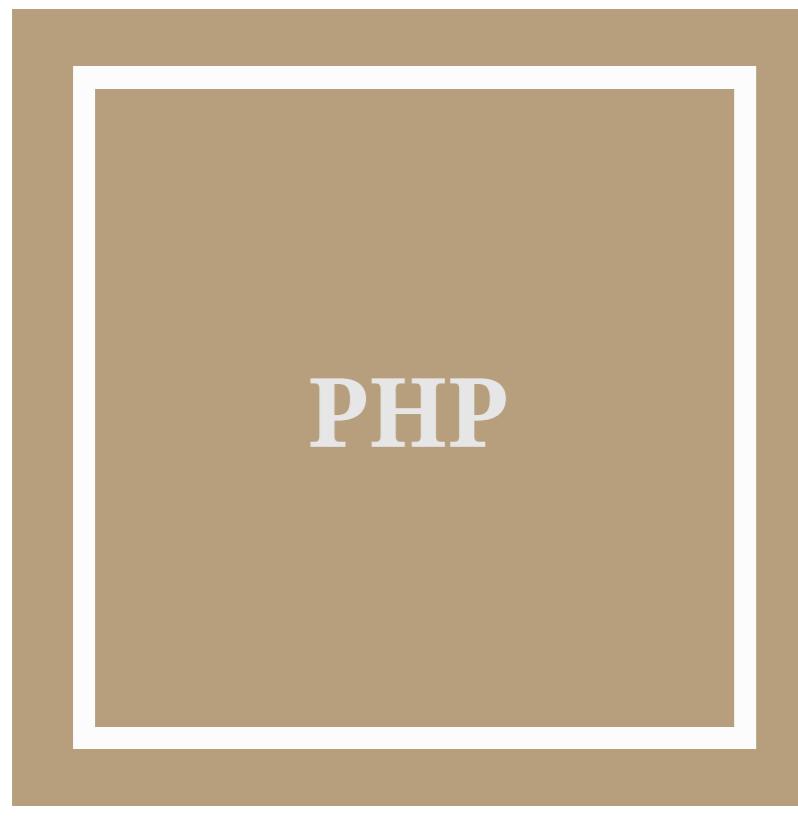
기술소개



안드로이드 4대 컴포넌트 구성요소 활용방법 숙지
4대컴포넌트 중 Activity, Content Provider를 사용하여 앱개발
Google Play Store에 앱 출시, 배포 및 버전관리
다양한 데이터들의 저장을 위해 Firebase 연동
HTTP 통신을 위해 Retrofit2 Library를 사용하여 Json 문서
파싱



Git 시스템을 활용하여 Github 사이트 저장소에 파일 저장
하는 법 숙지
VScode 개발환경에서 Git 형상관리 프로그램을 이용하여
Clone, Create Branch, Commit, Push 등을 사용하여 반
응형 웹개발을 위해 버전관리, 소통하며 팀프로젝트 경험



PHP 문법을 이용해 MySQL(DBMS)에 액세스
MySQL로 DB 입출력을 위한 쿼리문 요청방법 숙지



PHP에서 작성한 쿼리문 요청을 통해 전달받은 DB를 테이블
생성 후 테이블에 저장
GUI환경에서 SQL문 질의를 통해 CRUD 작업

NATIVE APP

프로젝트 개요

프로젝트명 Travin(트래빈) - 여행 정보공유를 위한 SNS

담당역할 기획, 디자인, 개발, 출시 및 업데이트

개발인원 1명

개발기간 2022.10 ~ 2022.11 (4주)

개발환경 OS - Windows 10

Development Tool - Android Studio, Visual Studio Code

Language - Kotlin, XML, PHP

Database - MySQL, Firebase

Hosting - Dothome WebHosting

프로젝트 설명

사용자들과 여행 정보교류를 위한 커뮤니케이션 SNS입니다. 키워드를 선택해 여행스타일이 비슷한 사용자들끼리 친구를 맺고 메세지 대화를 할 수 있습니다. 글을 업로드하고 다른 사용자들은 글에 댓글 및 좋아요를 누를 수 있습니다. 주변에 숙소를 찾고있는 경우, 주변 숙소찾기를 이용하여 숙소의 위치를 알아낼 수 있습니다.

GitHub

<https://github.com/jjeansp9/Travin>

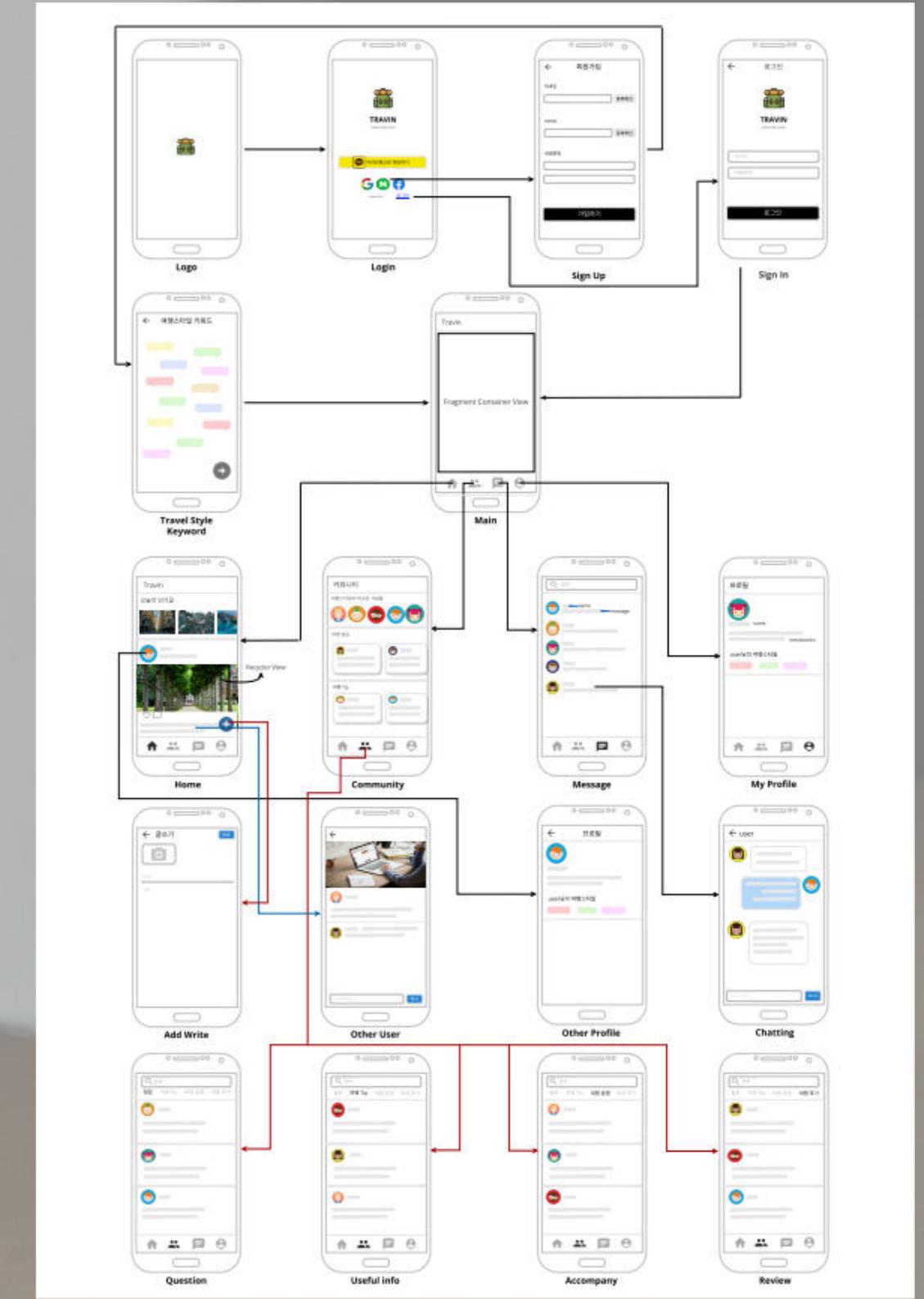
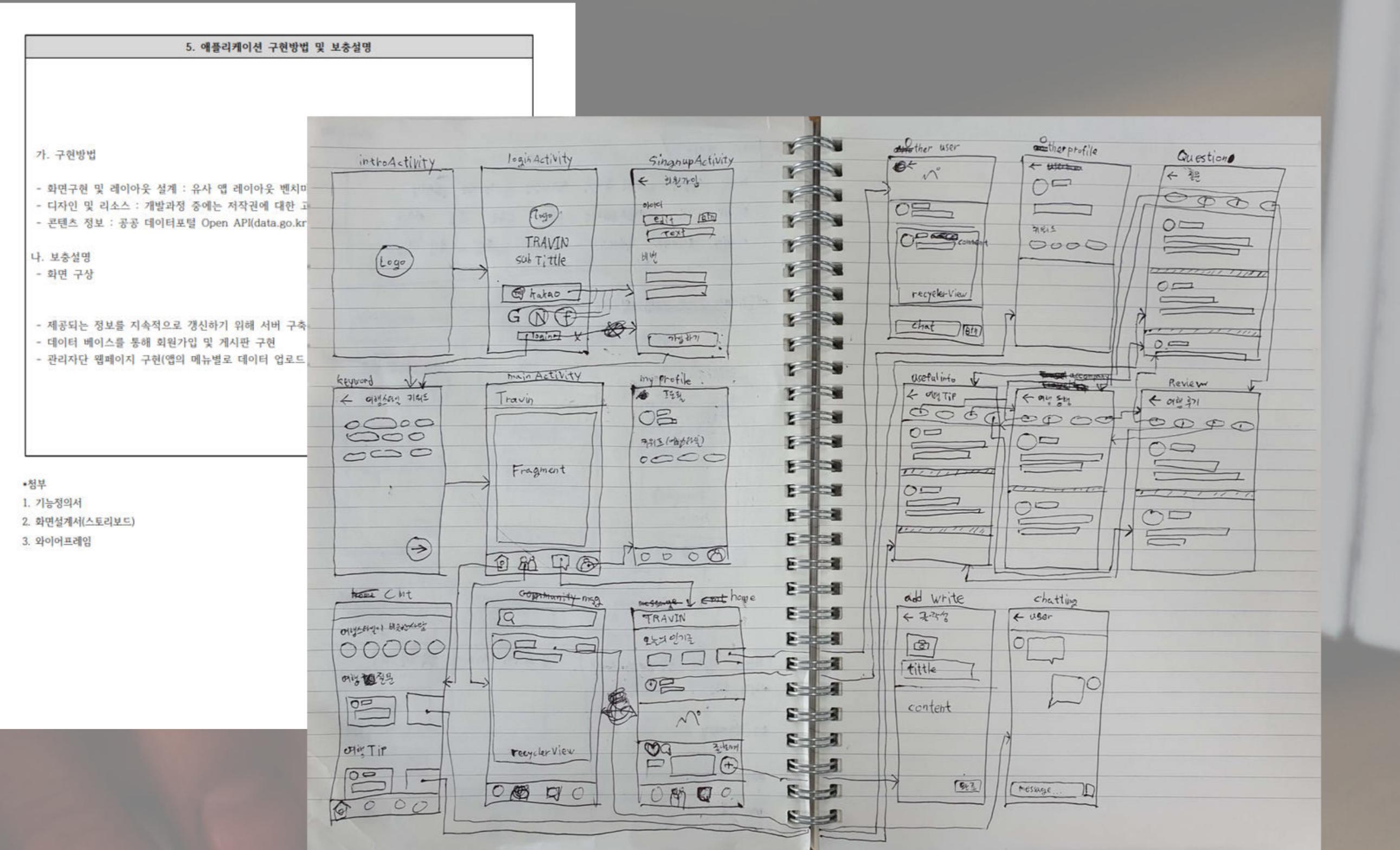
PlayStore

<https://play.google.com/store/apps/details?id=com.jspstudio.travin>

프로젝트 소개

프로젝트 기획 과정

[모바일 앱 기획서 양식]			
훈련과정명	안드로이드 네이티브 & 하이브리드 앱 개발자 양성	작성자	박진솔
팀 구성	1인 개발	팀장	박진솔 - 총괄 제작
		팀원-server	박진솔 - 서버, DB 및 관리자단 페이지 설계
		팀원-client	박진솔 - 앱 기능구현 및 레이아웃
1. 애플리케이션 명			
Travin - 여행정보 공유를 위한 SNS			
2. 애플리케이션 기획 배경(의도)			
<p>인터넷검색을 통해 기대를 갖고 여행지를 찾지만 예상한 생각과 다르게 정보부족으로 인해 실망한 적이 있다. 특히 혼자 여행을 다니는 경우 숙소를 이용할 때 발생하는 금액문제, 혼자 식당을 이용해야하는 불편함, 여행을 다니면서 느끼는 기쁨, 슬픔, 행복 등의 감정들을 나눌 사람이 없는 경우 또한 사회적인 인간에게는 큰 불편함으로 다가올 수 있다.</p> <p>이러한 문제들을 해결하기 위해 다양한 여행자들과 서로 소통하며 여러 감정들을 나누며, 여행스타일이 맞는 여행자들끼리 동행도 하면 기쁨이 두배가 될 거라는 생각도 든다.</p> <p>또한 한국관광데이터앱, 통계청에 따르면 국내여행 횟수, 1인가구 인구수가 과거부터 현재까지 증가했으며 앞으로도 꾸준히 증가할 것으로 보인다. 이러한 통계자료들을 참고해보면 여행을 즐기는 인구수가 많아질수록 여행관련 앱 이용자수가 증가할 것이라는 미래에 대한 추측을 해볼 수 있다.</p>			
3. 애플리케이션 주요기능			
<p>여행과 관련된 정보 및 소통을 하는 소셜 앱이며 아래와 같은 기능들이 있다.</p> <ol style="list-style-type: none"> 여행과 관련된 사진이나 글을 작성하여 업로드하여 서로 정보를 공유 여행스타일이 비슷한 사람들끼리 소통할 수 있게 메시지 기능 제공 사용자가 여행할 지역에 대해 질문할 수 있게 여행질문 공간 마련 여행하고자 하는 지역 주변 숙소찾기 기능 제공 			
4. 주요 사용대상(タ겟)			
○ 다수의 사람들과의 소통, 정보공유를 원하는 국내 및 해외 여행객			



기획서 작성

앱개발을 시작하기 전 만들고자 하는 앱의 주제 선정

주제를 정한 후 주제에 관련된 나의 니즈파악 등 기획 의도 구체화

어떤 기능들을 넣을지 생각하고 기획서에 작성

와이어프레임

기획서를 작성한 후 기획한 앱의 화면설계를 위해 와이어프레임 스케치

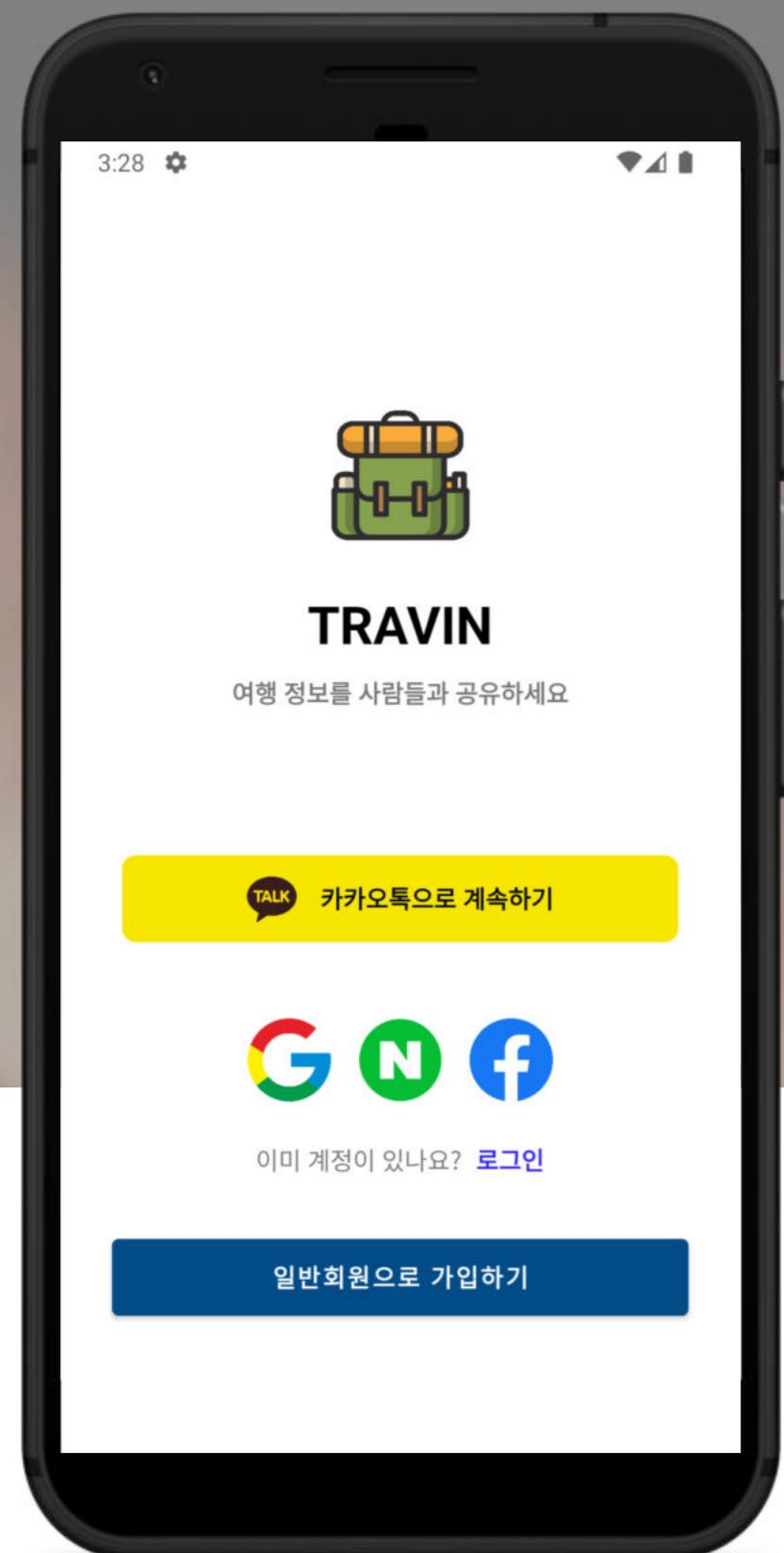
각각의 화면마다 Activity 혹은 Fragment 이름 정의

작성한 기획서, 스케치한 와이어프레임을 기반으로 그래픽 형태의 와이어프레임을 설계

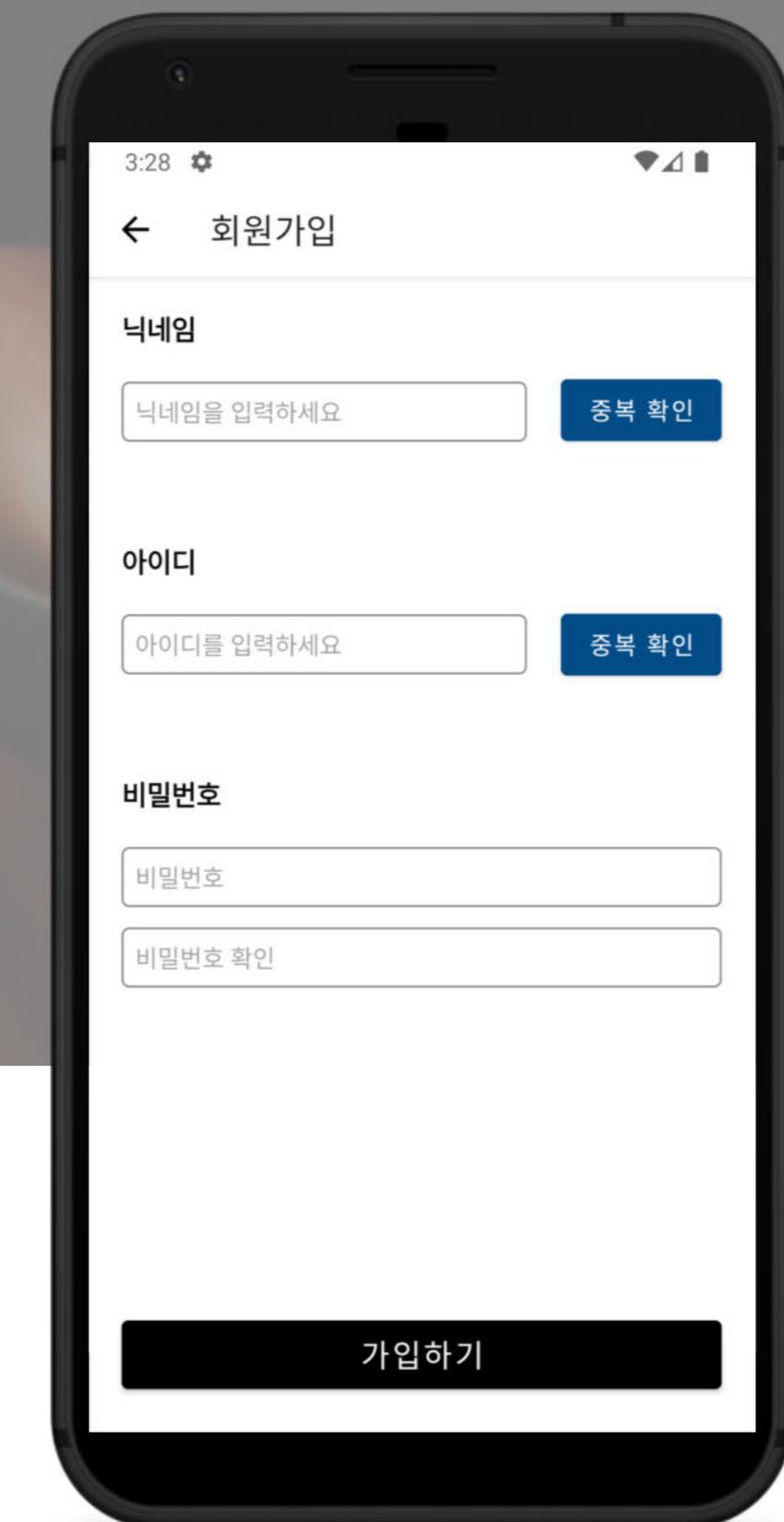
프로젝트 소개

주요 화면구성

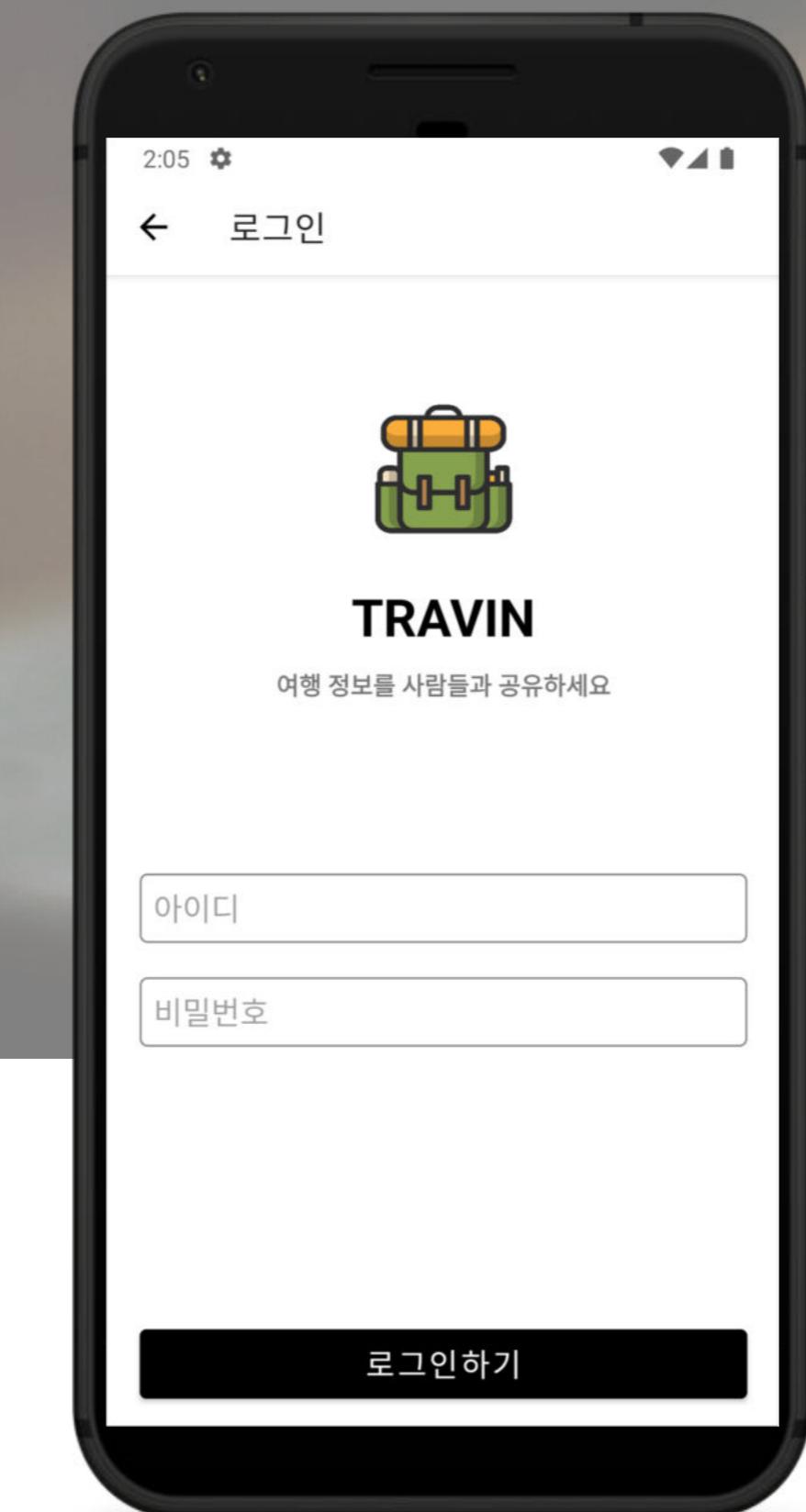
TRAVIN APP



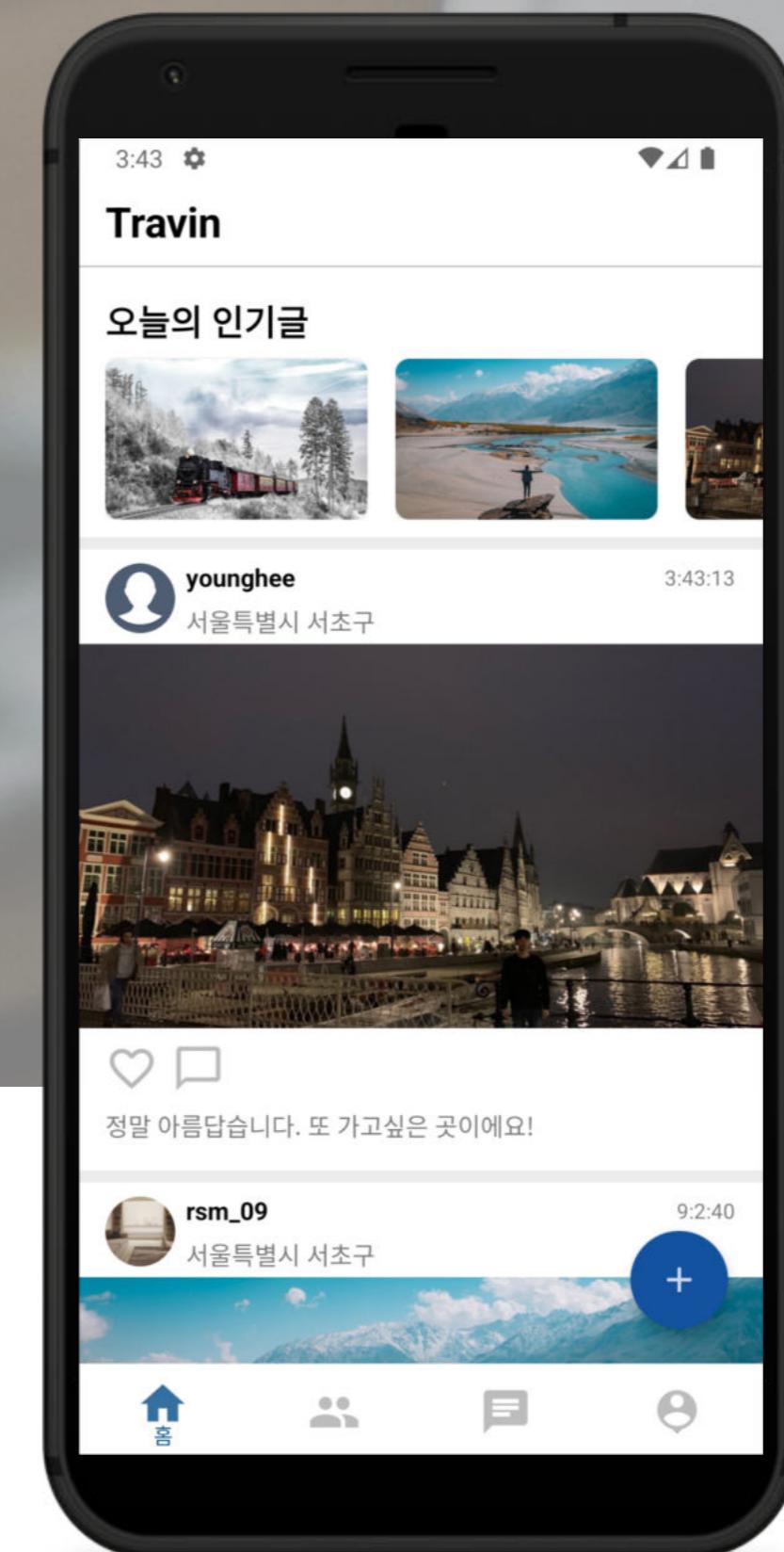
Login Activity
(로그인화면)



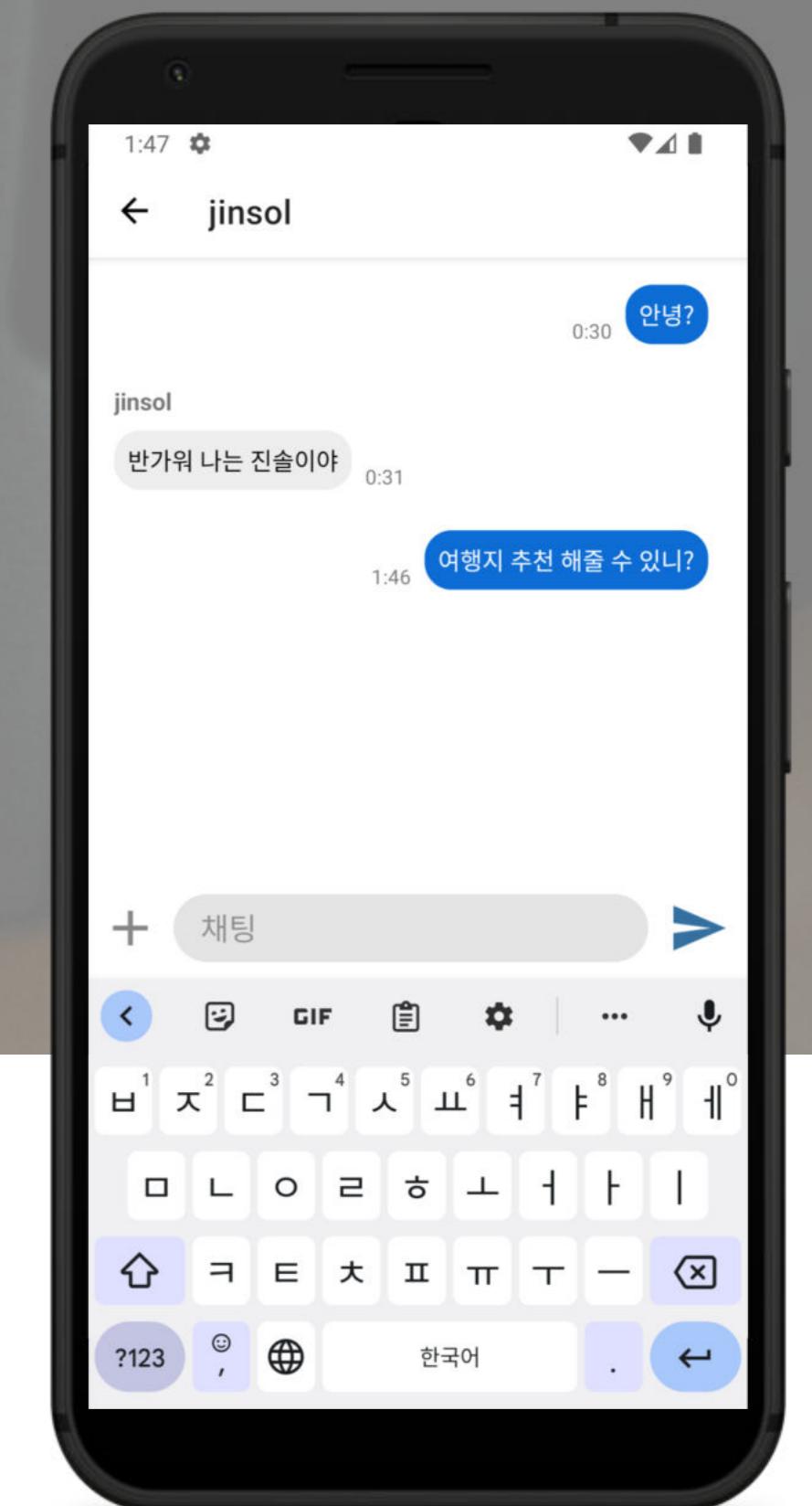
SignUp Activity
(회원가입 화면)



SignIn Activity
(일반계정 로그인화면)



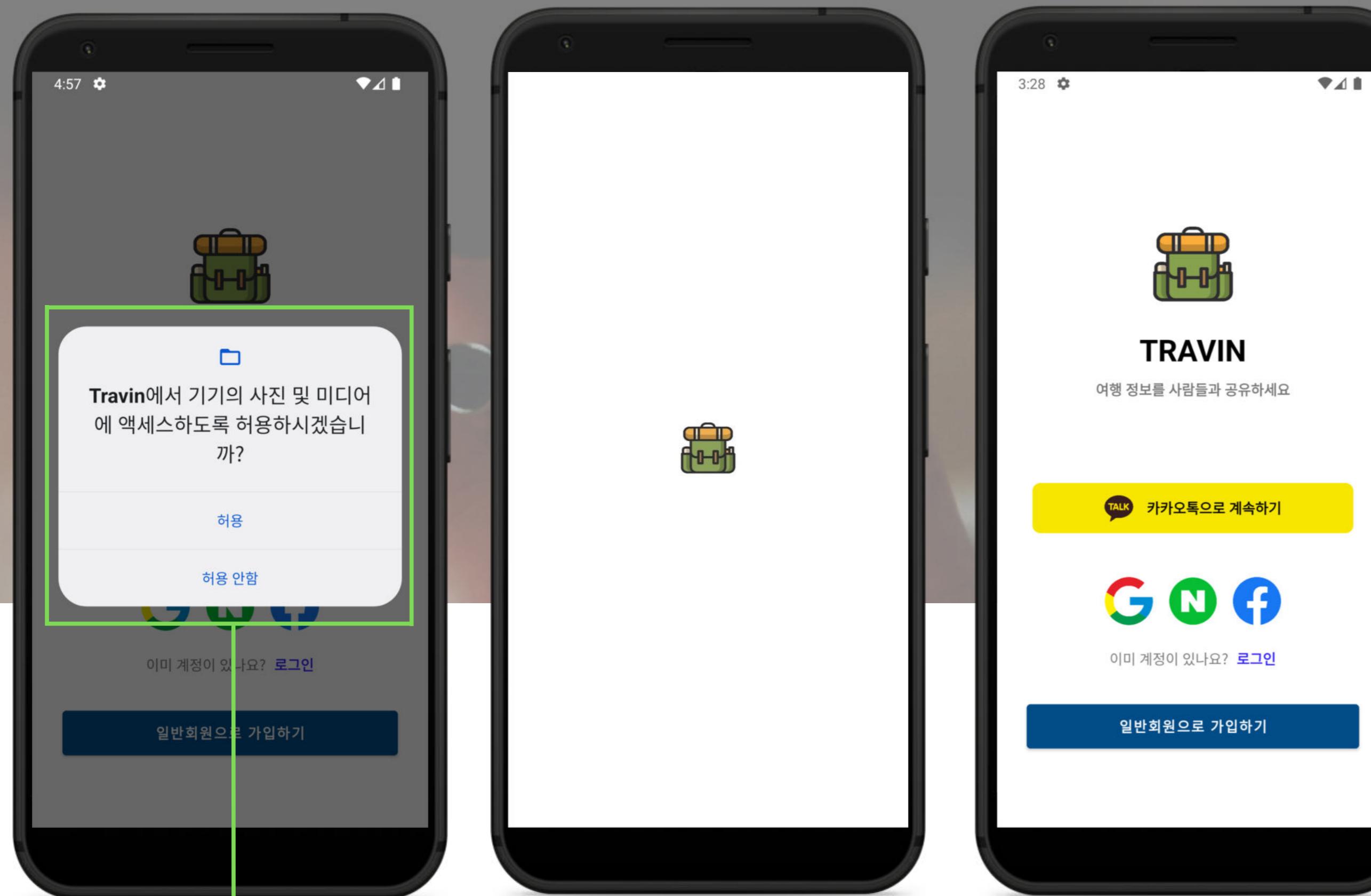
Home Fragment
(홈 화면)



Chatting Activity
(채팅화면)

프로젝트 소개

TRAVIN APP



사용자의 사진저장앱에 접근하기 위해 Manifest에서 WRITE_EXTERNAL_STORAGE / READ_EXTERNAL_STORAGE 퍼미션을 추가하고, requestPermissions()을 사용하여 사진앱에 접근여부에 대해 권한 요청하였습니다

Intro Activity
(인트로 화면)

Login Activity
(계정 로그인화면)

화면설명

앱설치하고 처음 실행하면 사진앱 액세스 권한에 대해 요청합니다 사진권한을 선택한 후부터 앱을 실행하면 IntroActivity(인트로 화면)이 실행되고, 1초 후에 Login Activity(로그인 화면)으로 이동합니다

Login Activity(로그인 화면)에서 로그인 버튼을 클릭하면 Intent를 사용하여 SignInActivity로 화면전환됩니다

회원가입 버튼을 클릭하면 Intent를 사용하여 SignUpActivity로 화면전환됩니다

각 SNS계정 아이콘을 클릭하면 해당 계정으로 로그인하여 MainActivity로 이동합니다

기술설명

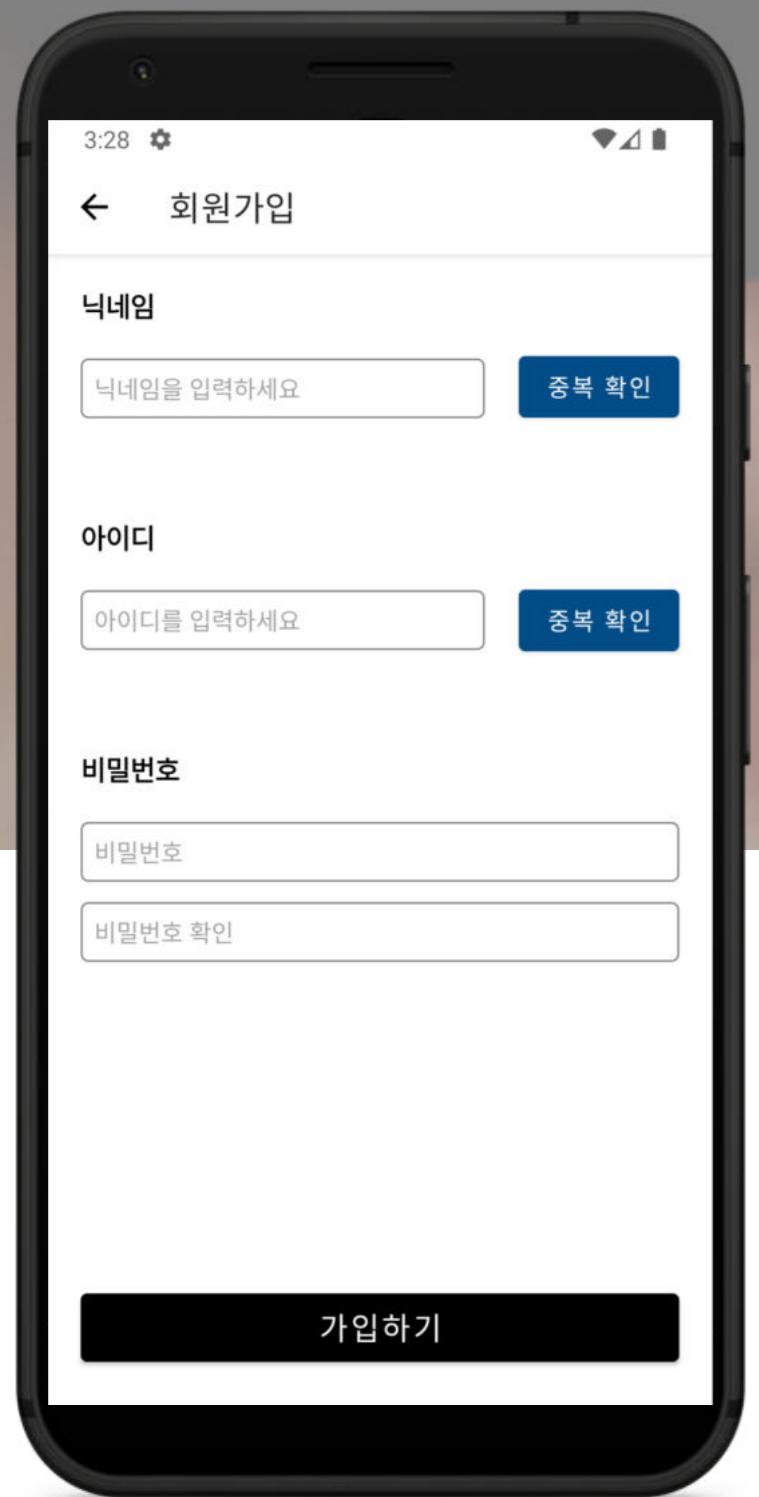
사진권한 선택 후 앱을 실행하면 Intro Activity(인트로 화면)이 실행되며 Handler 객체가 가지고있는 postDelayed() 함수에 Intent객체를 이용해 화면전환하게 하고, delay 파라미터에 1000을 입력하여 1초후에 작동하게 합니다. 코틀린의 특징인 SAM 변환 문법으로 간결하게 코드작성했습니다

Kakao for Developer 사이트에서 제공하는 Kakao SDK for Android를 사용하여 카오 로그인을 구현하였습니다

[일반회원으로 가입하기] 버튼을 클릭하면 Intent객체를 사용하여 SignUpActivity(회원가입 화면)으로 이동합니다

프로젝트 소개

TRAVIN APP



SignUp Activity
(회원가입 화면)

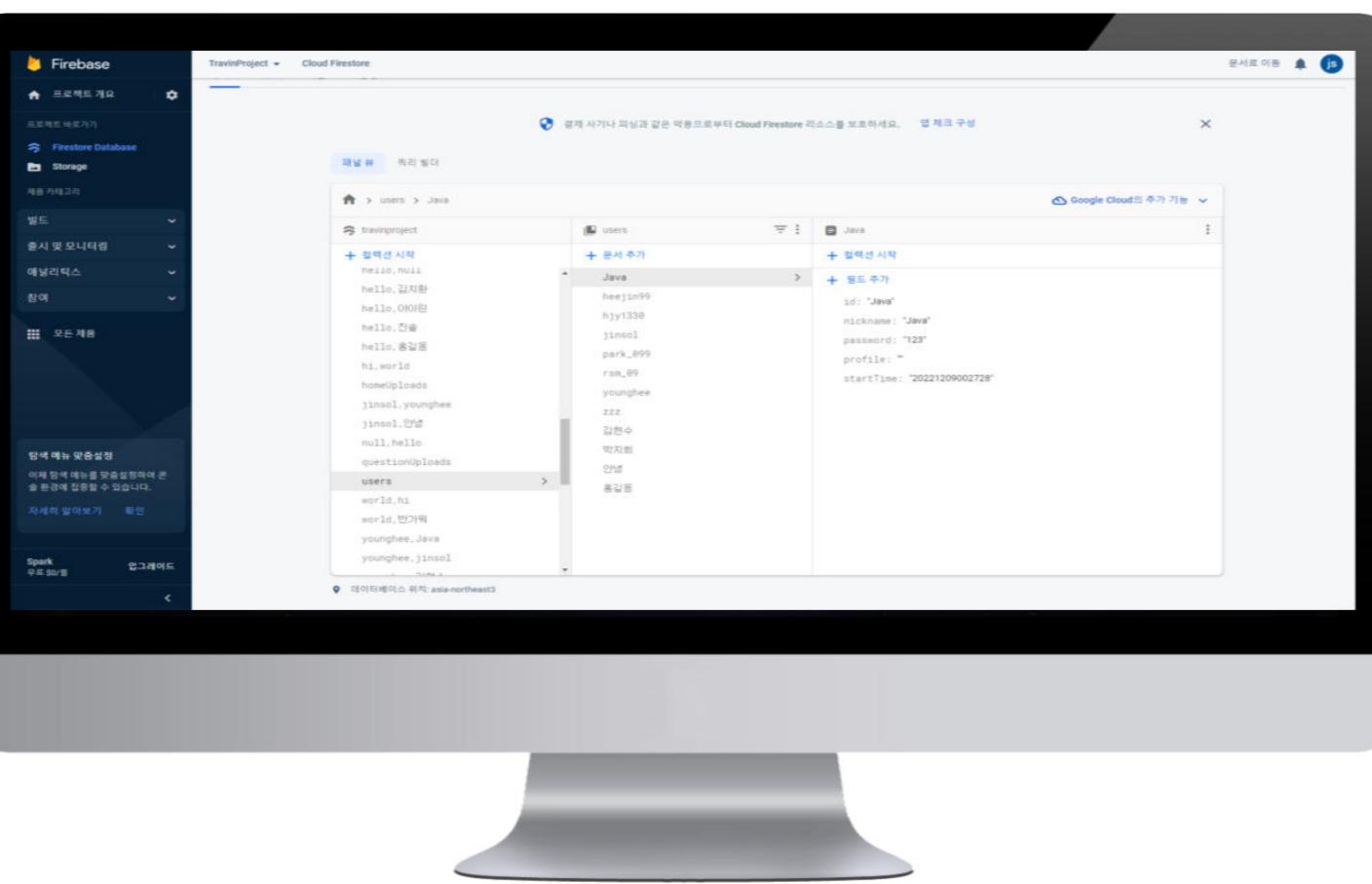
VSCODE 개발환경에서 작성한 PHP문서

```

Travin > TravinDB > insertUserData.php
1 <?php
2     header('Content-Type:application/plain; charset=utf-8');
3
4     //파티클래프로 전달된 POST방식의 데이터를
5     $nickname= $_POST['nickname'];
6     $id= $_POST['id'];
7     $password= $_POST['password'];
8     $now= date("Y-m-d H:i:s");
9
10
11
12
13     // MySQL DB에 데이터 저장 ( 테이블명 : travin_user_data )
14     $db= mysqli_connect("localhost", "jspstudio", "qawsedfrg#3", "jspstudio"); //host, id, password, filename
15     mysqli_set_charset($db, "utf8");
16
17     // 데이터들을 MySQL travin_user_data 테이블에 데이터를 삽입하는 쿼리문
18     $sql= "INSERT INTO travin(user_id, id, password, date) VALUES('$nickname', '$id', '$password', '$now')";
19     $result= mysqli_query($db, $sql);
20
21     if($result) echo "회원가입 완료";
22     else echo "회원가입에 실패했습니다. 다시 시도해주세요";
23
24     mysqli_close($db);
25 ?>

```

닷홈 호스팅에서 제공하는 MySQL 사용



Firebase에서 Travin 앱 프로젝트 생성, 패키지 등록 후 users라는 이름의 컬렉션 생성 후 Document에 회원정보 데이터들을 저장

화면설명

닉네임, 아이디, 비밀번호, 비밀번호 확인란을 입력하여 회원가입을 합니다

회원가입에 성공할 경우 회원정보 데이터들이 저장되며, 다음 화면으로 이동합니다

동일한 닉네임, 아이디가 있는 경우 및 비밀번호, 비밀번호확인이 동일하지 않은경우 회원가입이 불가능합니다

기술설명

회원정보데이터들을 저장하기위해 두가지 방법으로 구현했습니다

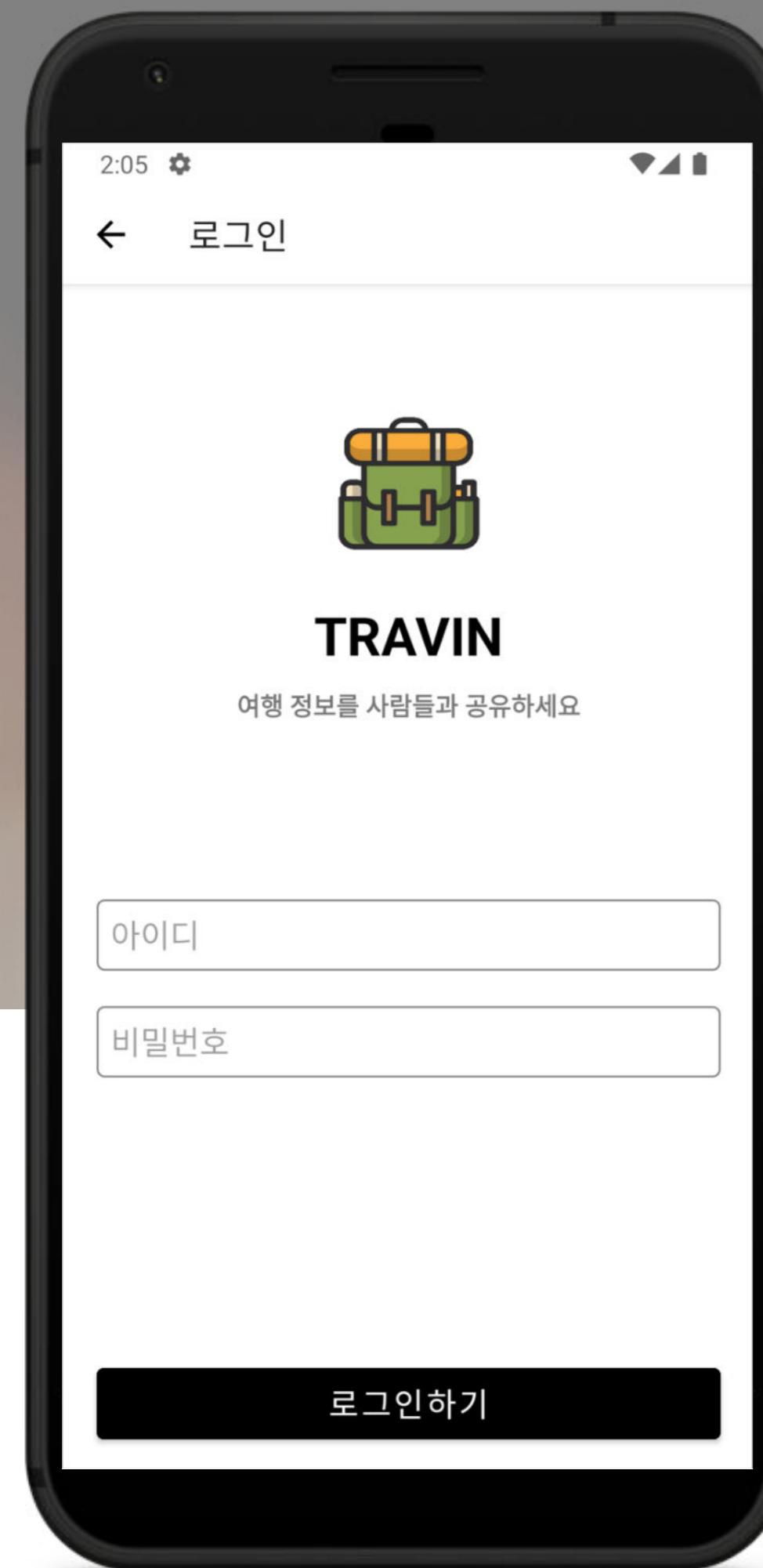
(1. Firebase Firetore / 2. REST API 사용을 위해 Retrofit2 Library 추가)

1. 회원가입 성공 시 회원정보데이터 저장을 위해 입력한 정보들을 문자열 형태로 받아오고, 가입하기 버튼을 누르면 입력한 회원 정보들을 Firebase Firestore에 Collection 생성후, Document에 저장했습니다

2. 가입하기 버튼을 누르면 회원정보 저장을 위해 REST API를 이용하였고, 쉽게 이용하기 위해 Retrofit2 Library를 추가하여 사용했습니다, Interface 를 구현한다음 POST방식을 사용하였으며 json 문서형태로 회원정보를 PHP문서로 보낸 후 PHP문서에서 쿼리문을 작성하여 MySQL 테이블 생성 및 회원정보 데이터를 추가하였습니다

프로젝트 소개

TRAVIN APP



SignIn Activity
(일반 로그인화면)

화면설명

아이디와 비밀번호를 입력하여 로그인합니다

로그인에 성공한 경우 Main Activity(홈 화면)으로 이동합니다

뒤로가기 아이콘모양의 버튼을 누르면 이전화면(Login Activity)으로 돌아갑니다

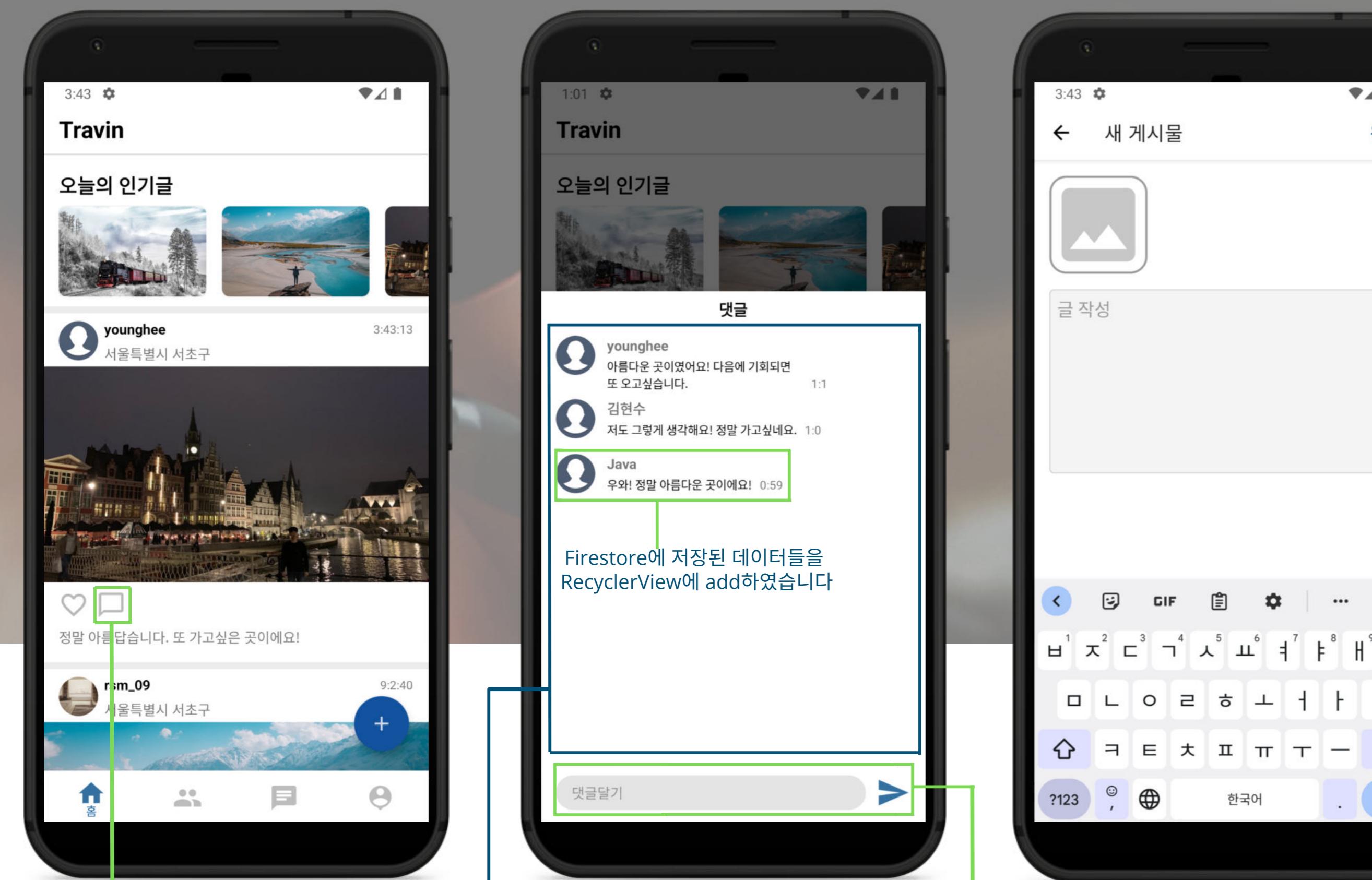
기술설명

로그인하기 버튼을 클릭하면 EditText에 입력한 아이디, 비밀번호를 입력한 문자열을 받아오고, Firebase Firestore에 저장된 회원정보들을 불러와 if문으로 비교하여 문자열이 같은 경우 로그인되며 Intent를 이용해서 Main Activity(홈 화면)로 이동합니다

getSupportActionBar()를 사용하여 뒤로가기버튼을 생성, Override되어있는 onSupportNavigateUp()에 onBackPressed()를 추가하여 뒤로가기버튼 기능을 구현했습니다

프로젝트 소개

TRAVIN APP



해당 아이콘을 클릭하면 XML로 설계한 Bottomsheet.show()가 동작합니다

Home Fragment
(홈 화면)

XML에 RecyclerView로 화면 설계 후 Home Fragment에 bottomSheetDialog.show()를 사용하여 바텀시트를 생성하였습니다.

Upload Activity
(업로드 화면)

EditText에 입력한 문자열을 입력한 후 보내기 아이콘을 클릭하면 Firestore에 입력한 문자열을 저장합니다.

화면설명

업로드된 글 중 좋아요를 가장 많이 받은 글의 이미지가 오늘의 인기글란에 등록되어 보여집니다

FloatingActionButton 버튼을 클릭하면 글 업로드화면이 열리고, 업로드화면에서 이미지, 글을 입력한 후 완료버튼을 눌러 업로드할 수 있습니다

업로드되어있는 각 글들마다 댓글모양 아이콘을 클릭하면 댓글창이 열립니다. 댓글창에서 업로드된 글에 대한 댓글을 남길 수 있습니다

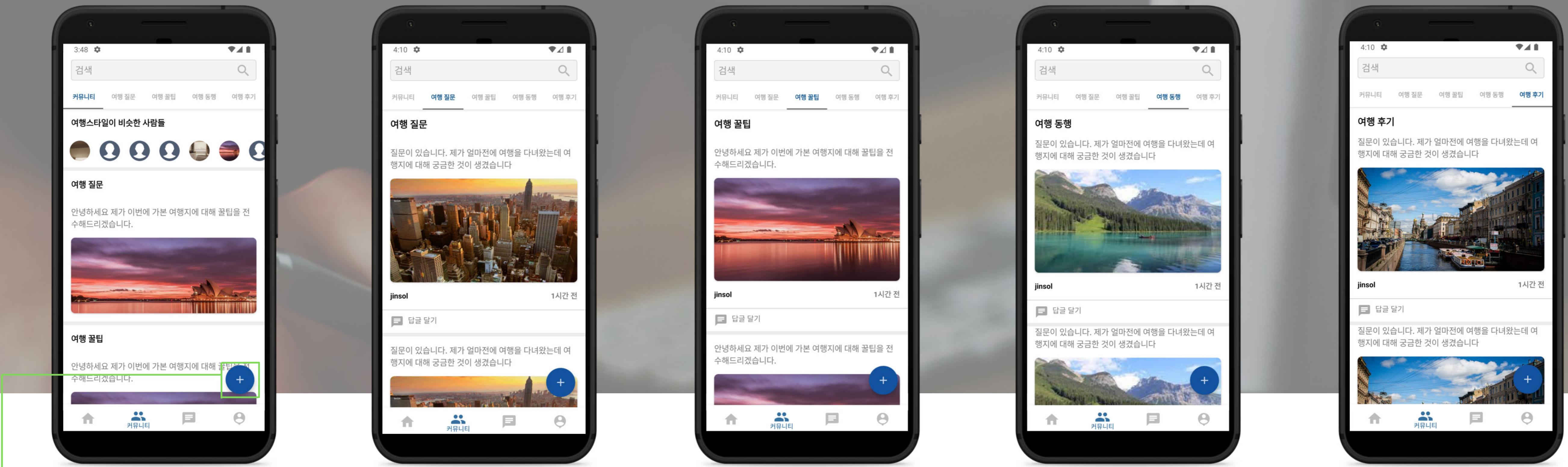
기술설명

Upload Activity(업로드화면)에서 글과 이미지를 업로드하기 위해 Manifest에서 Write, Read 퍼미션 추가 후 Intent 객체안에 ACTION PICK을 사용하여 사진에 접근하여 사진을 업로드 할 수 있게 구현하였고, 글 입력 후 완료버튼을 누를 시 Firestore에 이미지, 글 작성란에 입력한 문자열이 저장됩니다.

업로드화면에 입력하고 완료버튼을 눌러 Firestore에 저장된 데이터들을 Home Fragment(홈화면)에 addSnapshotListener 사용하여 전역변수로 만든 배열변수에 저장한 후 add하여 업로드된 모든 데이터들을 RecyclerView 형태로 화면에 보여줍니다

프로젝트 소개

TRAVIN APP



→ XML문서로 별도의 화면설계 후 Popupmenu를 이용하여 4개의 탭메뉴 이름이 적혀진 Popupmenu 창을 엽니다. 해당 문자열을 putExtra로 업로드화면으로 넘기면서 Intent 객체로 화면전환합니다.

화면설명 커뮤니티 - ViewPager2에 놓여진 5개중 4개의 탭메뉴들을 Vertical 방향으로 보여줍니다. 왼쪽으로 슬라이드하여 업로드된 글들을 볼 수 있습니다

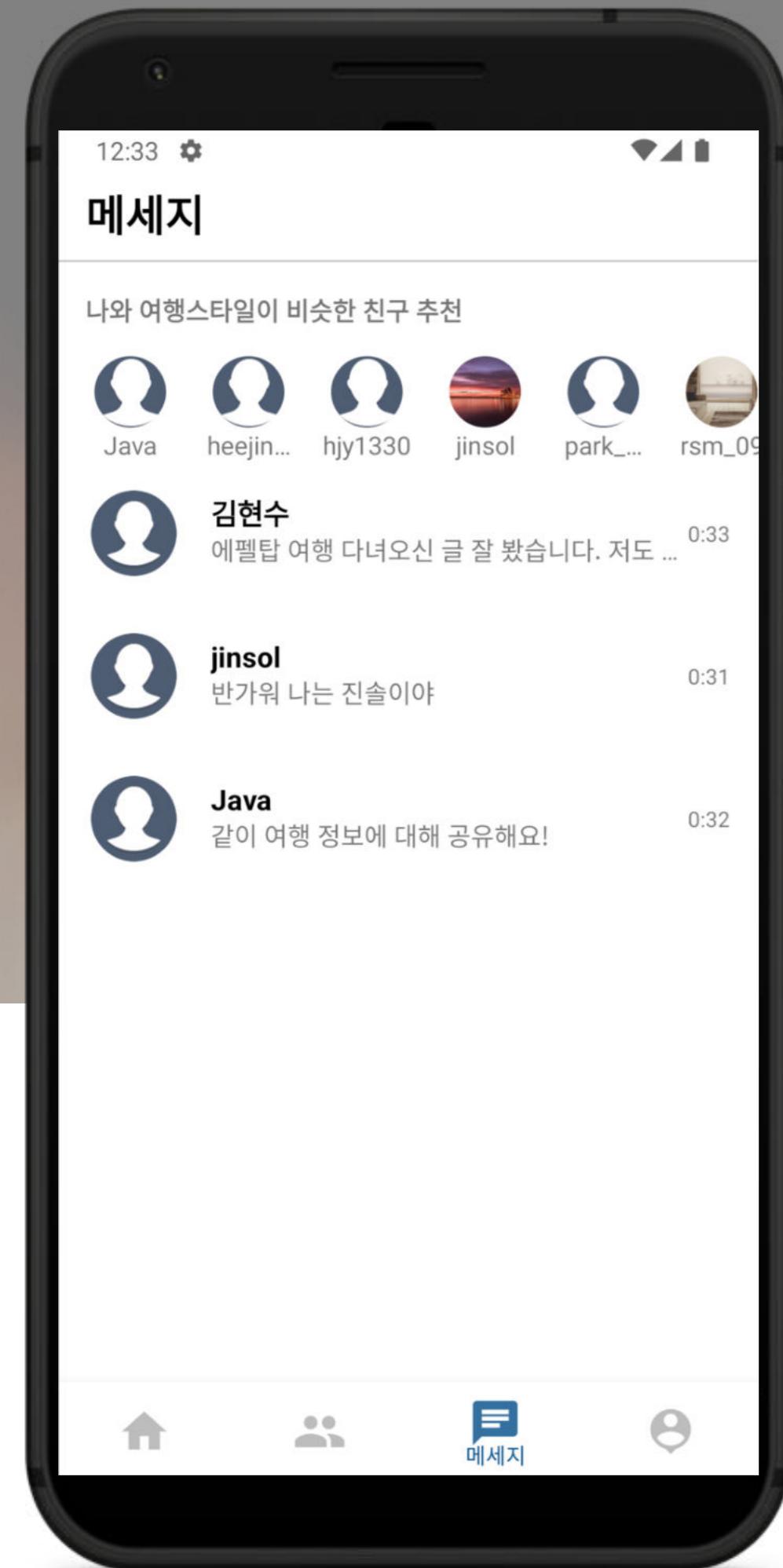
여행질문, 여행꿀팁, 여행동행, 여행후기 - 각 탭마다 질문, 꿀팁, 동행자찾기, 여행후기 남기기 등 글을 업로드하여 소통할 수 있습니다

Community Fragment
(커뮤니티 화면)

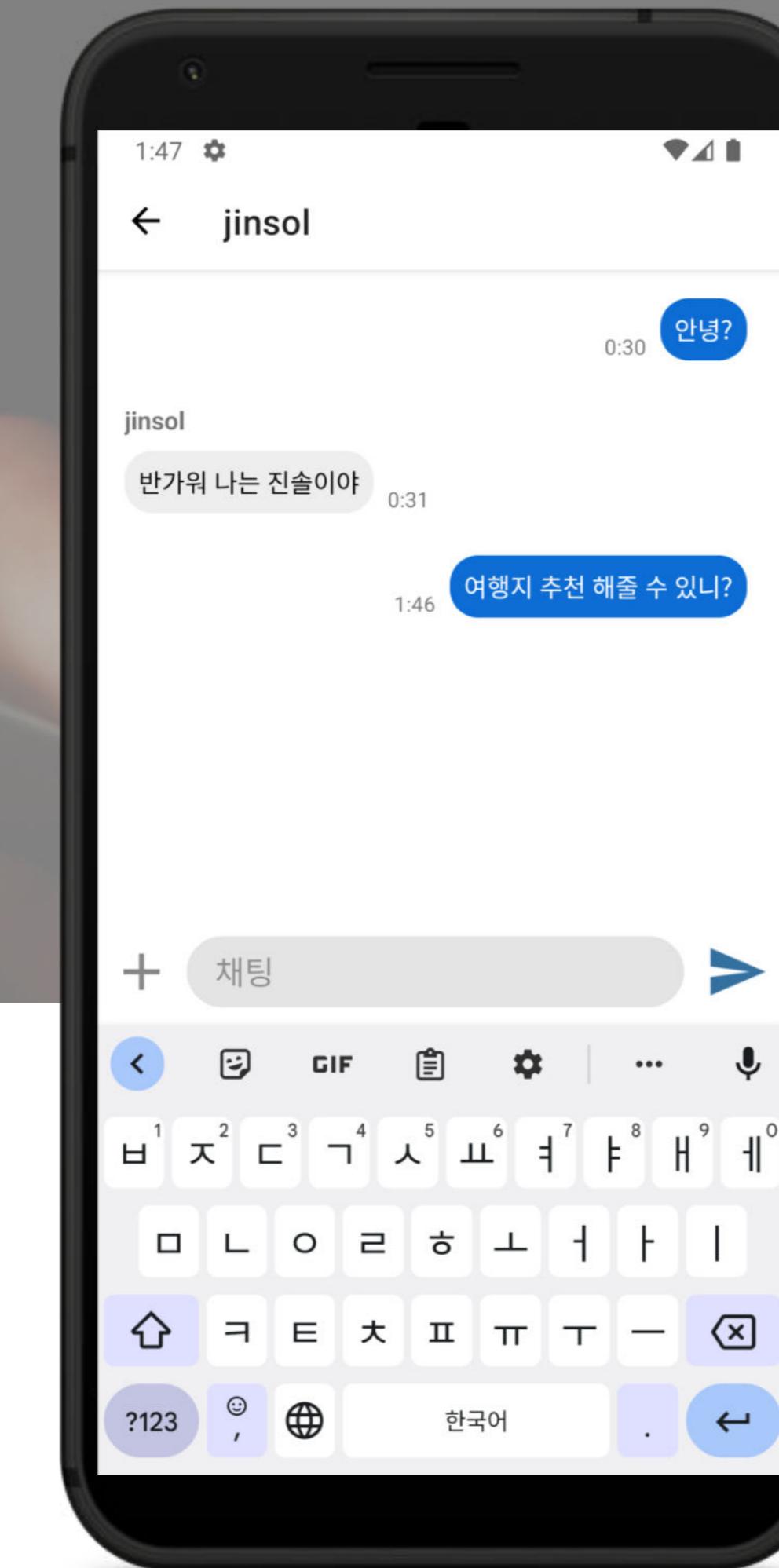
기술설명 Community Fragment에서 ViewPager2, Adapter, TabLayoutMediator를 이용하여 pager 5개를 연동하였습니다
각각의 pager들을 연동하기 위해 Fragment 4개를 추가한 후 Adapter를 사용했고, Fragment마다 RecyclerView를 활용해 글 업로드된 데이터들을 보여줍니다

프로젝트 소개

TRAVIN APP



Message Fragment
(메세지 화면)



Chatting Activity
(채팅 화면)

화면설명

Message Fragment(메세지 화면) 상단에는 회원가입한 유저들의 프로필이미지, 닉네임 정보 목록을 볼 수 있습니다

나에게 채팅을 보낸 사용자를 클릭하면 Chatting Activity(채팅 화면)으로 화면전환 됩니다

Chatting Activity(채팅 화면)에서는 채팅을 입력한 후, 오른쪽의 보내기버튼을 눌러 상대방과 채팅을 할 수 있습니다

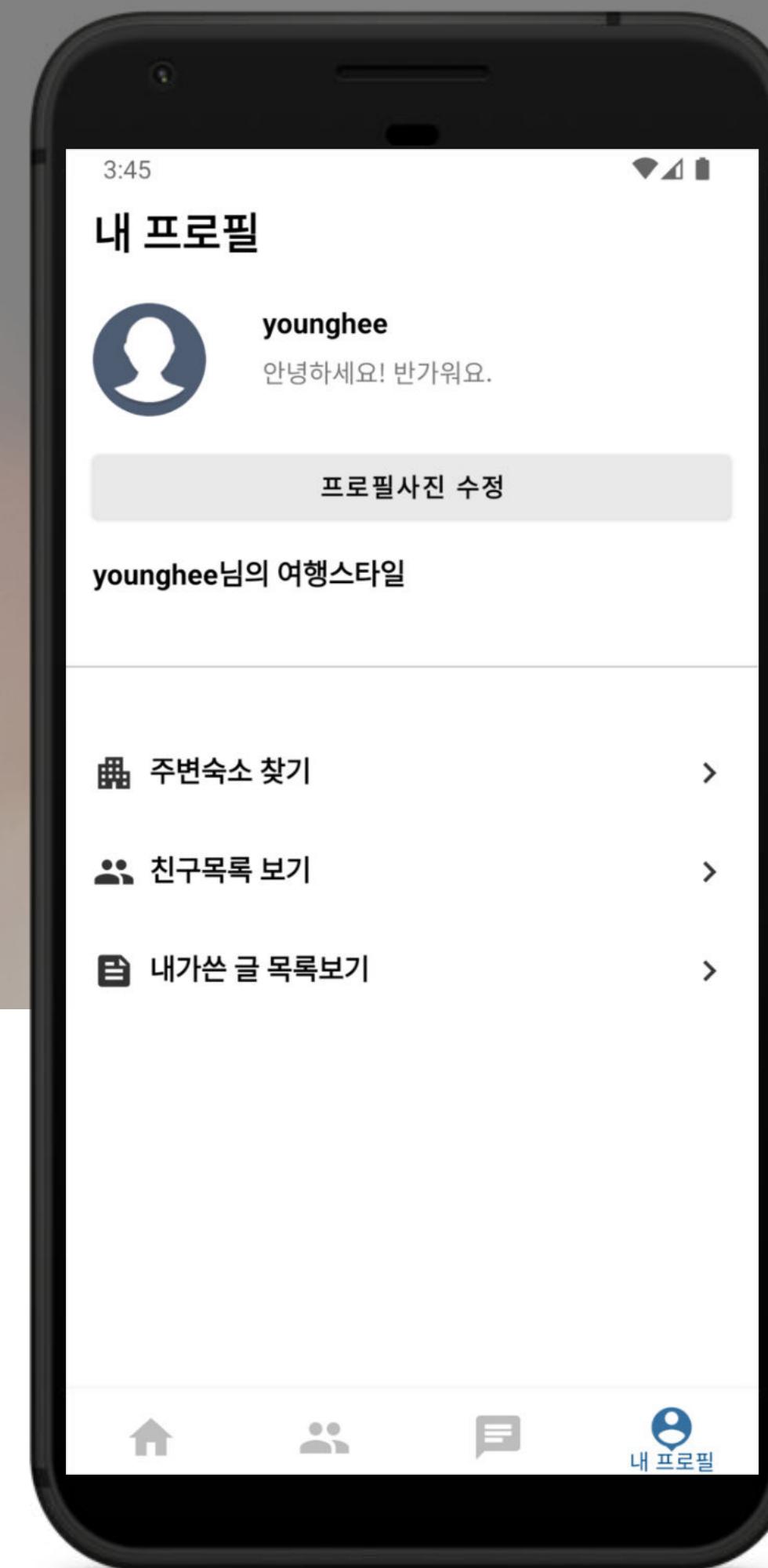
기술설명

Message Fragment(메세지 화면) 상단에 각 사용자들의 프로필이미지를 클릭하면 친구요청 AlertDialog 를 띄워 확인버튼을 누르면 해당 사용자에게 친구요청을 할 수 있습니다. Adapter에 별도의 Listener Interface를 구현하여 각각의 RecyclerView의 Item마다 클릭이벤트 처리를 구현하였습니다

Message Fragment(메세지 화면)에서 메세지목록에 있는 친구를 클릭하면 RecyclerView item 클릭이벤트 처리 구현하였으며, Intent 객체를 사용해 Chatting Activity(채팅 화면)으로 화면전환 되고, 2개의 RecyclerView의 화면을 설계했고, Adapter에 if문으로 Shared Preferences에 저장된 내닉네임의 문자열을 비교하는 구문을 추가하여 각각의 RecyclerView에 연결했습니다

프로젝트 소개

TRAVIN APP



Account Fragment
(내프로필 화면)

화면설명

내 닉네임과 내 프로필 상태를 볼 수 있습니다.

[프로필사진 수정] 버튼을 누르면 프로필사진 변경을 위해 Upload Profile Activity(프로필사진 업로드화면)으로 화면전환 됩니다

[친구목록 보기]를 누르면 Account Friend List Activity(친구목록 화면) 화면으로 전환됩니다

기술설명

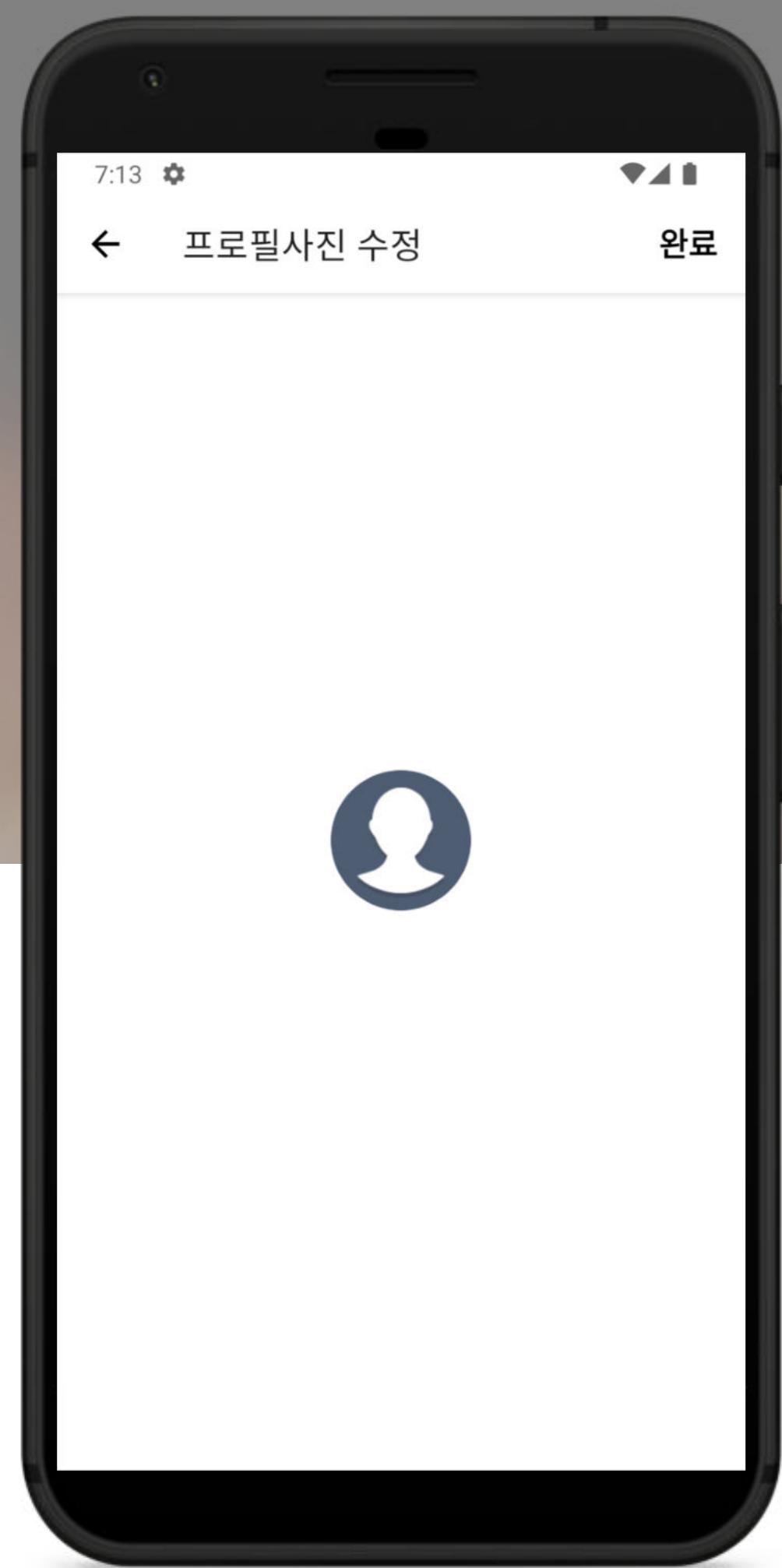
XML에서 화면 설계를 하였고 각각의 아이템마다 id를 줘서 kotlin문서에서 viewBinding하였습니다. viewBinding된 아이템마다 setOnCLickListener 객체를 추가하였고, 안에 함수는 별도로 작성해놓은 함수명을 입력하여 구현했습니다

화면에 보여지는 내 프로필 이미지는 프로필사진 수정을 통해 이미지가 변경됩니다. 프로필사진수정 화면에서 이미지수정 완료버튼을 누를 경우 Firestore에 저장되어있는 프로필Url 문자열을 받아와 Glide Lib를 사용하여 프로필이미지를 화면에 보여줍니다

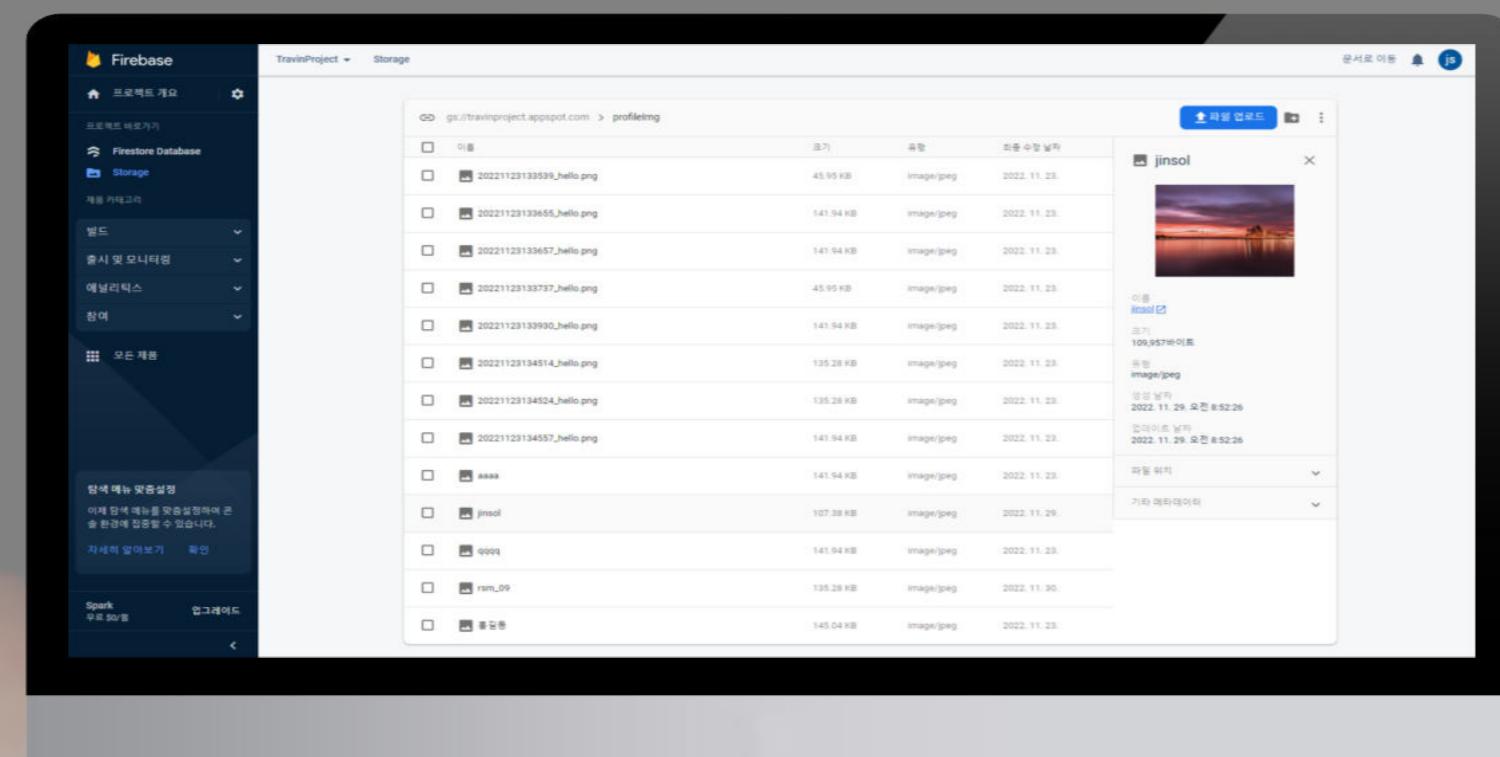
[주변숙소 찾기]는 Account Hotel List Activity(숙소리스트 화면), [친구목록 보기]는 Account Friend List Activity(친구목록 화면)으로 Intent 객체를 이용하여 화면전환합니다

프로젝트 소개

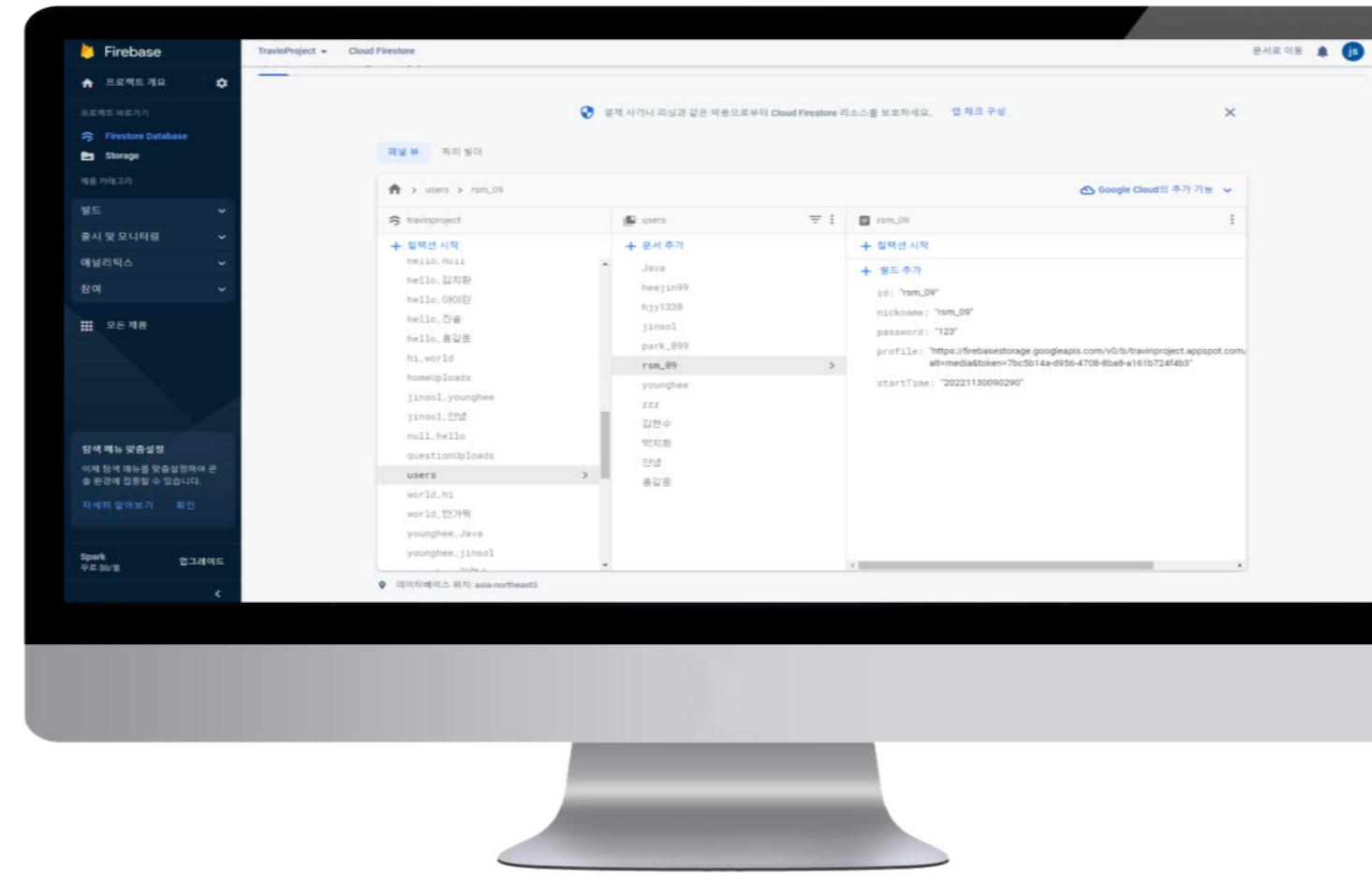
TRAVIN APP



UploadProfile Activity
(프로필사진 업로드화면)



Firebase Storage에 저장된 Profile Images



Firebase Firestore Database에 있는 user Document에 Profile Image Url 추가

화면설명

화면 가운데 프로필이미지를 클릭하여 사용자의 휴대폰에 접근하여 사진이 저장된 앱을 엽니다

사진앱에 저장된 사진을 클릭하면 UploadProfile Activity(프로필사진 업로드 화면)의 화면 가운데 이미지가 선택한 사진으로 바뀝니다

완료버튼을 누르면 선택한 이미지로 프로필사진을 변경합니다

기술설명

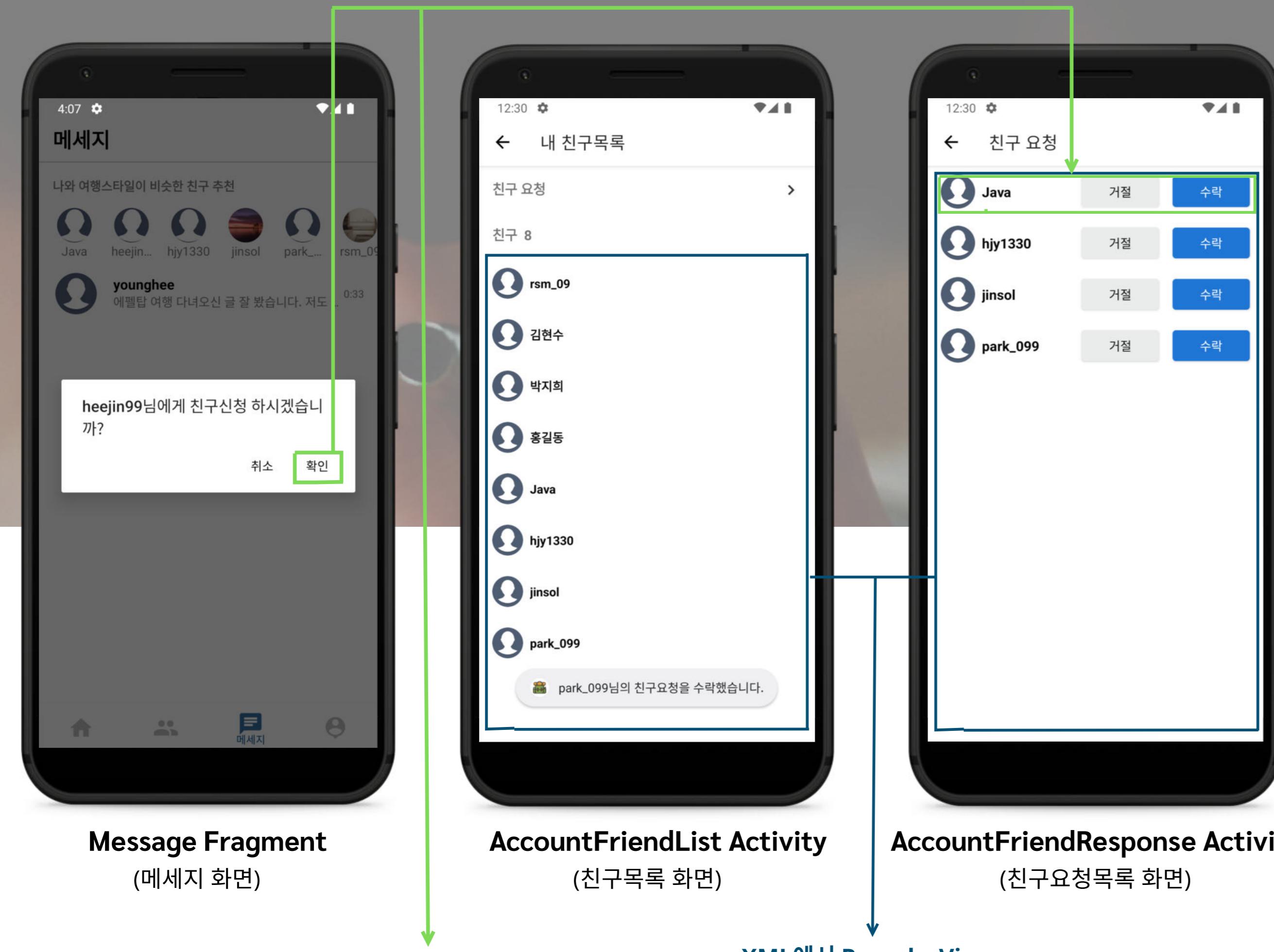
화면 가운데 사진을 클릭하면 Intent객체를 이용해 ACTION_PICK기능을 사용하여 사진앨범을 호출하고, ActivityResultContracts 객체의 startActivityForResult 함수를 사용해 사진 이미지를 가져옵니다

fireStorage.getReference()를 사용하여 profileImg의 폴더를 만들고 UserNickname으로 문서를 만들고 putFile()을 사용하여 사진을 저장합니다

Storage에 내 닉네임으로 저장되어있는 사진경로를 getReference().downloadUrl을 사용하여 접근한 다음 문자열로 받아와 Firestore.getInstance() 한 후 users 컬렉션에 접근하여 내 닉네임으로 된 Document에 저장합니다

프로젝트 소개

TRAVIN APP



여행스타일이 비슷한 친구추천 하단에 있는 유저들의 프로필사진 중 하나를 클릭하면 AlertDialog가 동작합니다 Dialog에 확인을 클릭해서 친구신청 합니다. 친구신청을 받은 사용자에게는 친구요청목록에서 응답할 수 있습니다

화면설명

AccountFriendList Activity(친구목록 화면)에서 [친구요청 >] 을 클릭하면 (친구요청목록 화면)으로 이동합니다

AccountFriendResponse Activity(친구요청목록 화면)에서 나에게 친구요청을 보낸 목록들을 봅니다. 수락을 누르면 친구목록에 추가되며, AccountFriendList Activity(친구목록 화면)에 보여집니다

AccountFriendList Activity(친구목록 화면)에 등록된 친구를 클릭하면 클릭한 친구와 채팅하기 위해 MessageChatting Activity(채팅 화면)를 엽니다

기술설명

[친구요청 >] 을 클릭하면 Intent 객체를 통해 AccountFriendResponse Activity(친구요청목록 화면)으로 이동합니다

MessageFragment(메세지화면)에서 친구요청을 하면 Firebase Firestore에 친구 요청받은 유저의 닉네임으로 Collection을 생성하고 친구요청자 닉네임으로 Document를 생성하고 value엔 요청자 닉네임을 문자열로 대입합니다

AccountFriendResponse Activity(친구요청목록 화면)에는 Collection().addSnapshotListener()를 사용하여 자신의 닉네임을 찾습니다. 있는 경우 Document의 값을 찾아서 Document 개수만큼 RecyclerView에 배열로 add합니다