

|  |
| --- |
| **Alpaca : 자율주행 택시 서비스** |
| **포팅 매뉴얼 (모바일)** |
|  |
|  |
|  |
|  |
|  |

목차

[1. Gitlab 소스코드 클론 이후 빌드 및 배포할 수 있도록 정리 3](#_Toc111754981)

[A. 개발환경 3](#_Toc111754982)

[Frontend 3](#_Toc111754983)

[Backend 3](#_Toc111754984)

[AOS 3](#_Toc111754985)

[배포 3](#_Toc111754986)

[IDE 3](#_Toc111754987)

[B. 빌드 & 배포 3](#_Toc111754993)

[I. Android 3](#_Toc111754994)

[2. 프로젝트에서 사용하는 외부 서비스 정보를 정리한 문서 6](#_Toc111754997)

[A. 외부 서비스 정보 6](#_Toc111754998)

[3. 시연 시나리오 8](#_Toc111754999)

[A. 목적지 선택 8](#_Toc111755000)

[B. 택시 호출하기 8](#_Toc111755001)

[C. 택시 운행 9](#_Toc111755002)

[D. 기타 정보 10](#_Toc111755003)

# 1. Gitlab 소스코드 클론 이후 빌드 및 배포할 수 있도록 정리

## A. 개발환경

### Frontend

* HTML / CSS
* JavaScript
* Node.js
* Vue.js : Vue 2

### Backend

* Firebase : 28.0.1

### AOS

* Kotlin : 1.7.10
* SDK : 31
* Gradle : 7.3.3
* JDK : 1.8

### IDE

* Android Studio : Chipmuck 2021.2.1 Patch 1

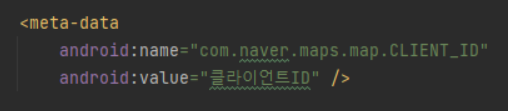
## B. 빌드 & 배포

* https://lab.ssafy.com/s07-mobility-autodriving-sub2/S07P22D208

### I. Android

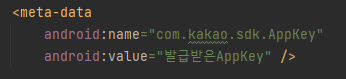
**빌드**

* NaverMapClientId 추가



AndroidManifest.xml 파일에서 클라이언트ID 부분에 추가

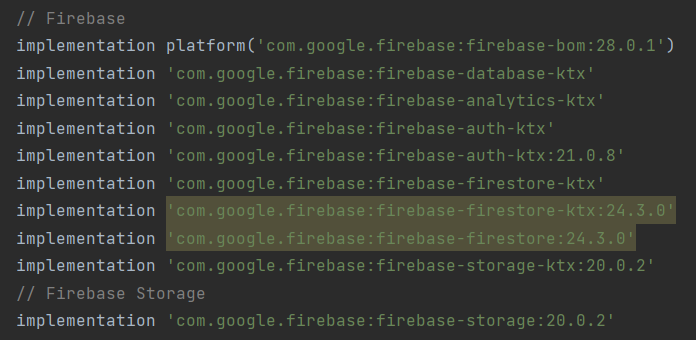
* 카카오맵 로컬 AppKey 추가



AndroidManifest.xml 파일에서 발급받은AppKey 부분에 추가

* Firebase 설정

1. 파이어베이스 콘솔에서 프로젝트 생성
2. 좌측 사이드바에서 Cloud Messaging 클릭
3. Android 앱 등록
4. google-service.json 다운로드 후 Project/app 밑에 복사



App 수준 gradle에 dependency 추가

* Build 버튼 클릭

**배포**

1. 상단 메뉴 Build -> Generate Signed Bundle/APK 클릭
2. Android App Bundle 체크 -> Next 클릭
3. Key store path 선택, key store password, alias, password 작성 후 Next 클릭
4. Release 체크 -> Finish 클릭
5. app/release 경로에 abb 파일 생성 완료

# ****2. 프로젝트에서 사용하는 외부 서비스 정보를 정리한 문서****

## A. 외부 서비스 정보

**Naver Map API** – 네이버 지도 API

URL:  [네이버 지도 API 관련 문서 링크](https://navermaps.github.io/android-map-sdk/guide-ko/0.html)

FCM – Firebase 클라우드 메시징

URL: [FCM API 관련 문서 링크](https://firebase.google.com/docs/cloud-messaging/android/client?hl=ko)

Firebase – Realtime Database

URL: [Realtime DB API 관련 문서 링크](https://firebase.google.com/docs/database?hl=ko)

Firebase – FireStore Database

URL: [Firestore DB API 관련 문서 링크](https://firebase.google.com/docs/firestore?hl=ko)

Firebase – Storage

URL: [Storage API 관련 문서 링크](https://firebase.google.com/docs/storage?hl=ko)

Firebase – Auth

URL: [Auth API 관련 문서 링크](https://firebase.google.com/docs/auth?hl=ko)

**Kakao 로컬 API** – 카카오맵 로컬 데이터

URL:  [Kakao Map local API 관련 문서 링크](https://developers.kakao.com/docs/latest/ko/local/common)

**Google Login API – 구글 로그인**– 구글 로그인

URL:  [Google Login API 관련 문서 링크](https://cloud.google.com/identity-platform/docs/use-rest-api?hl=ko)

**Facebook Login API** – 페이스북 로그인

URL:  [Facebook Login API 관련 문서 링크](https://developers.facebook.com/docs/facebook-login/web?locale=ko_KR)

# 3. 시연 시나리오

## A. 목적지 선택

- **유저 홈 화면**

앱을 실행시키면 가장 먼저 홈 화면이 보인다.

앱 중단의 즐겨찾기 RacyclerView와 자주 가는 목적지 RecyclerView의 아이템을 선택하여 목적지 명, 목적지 주소, 목적지의 좌표 등의 정보를 불러올 수 있다.

출발지 선택하기 버튼을 클릭하여 출발지 선택화면으로 전환한다.

**- 출발지 선택 화면**

네이버맵을 통해 현재 유저의 위치를 기준으로 맵을 포커싱하고 현재 위치에 오버레이를 설정한다.

화면 가운데 마커를 생성하여 해당 마커의 위치를 출발지로 선택할 수 있게 한다.

마커를 이동시켜 해당 위치의 정보를 불러온다.

목적지 선택하기 버튼을 클릭하여 불러온 정보를 출발지로 저장하고 목적지 선택화면으로 전환한다.

**- 목적지 선택 화면**

앱 상단의 출발지, 목적지 선택 TextView를 선택하여 목적지를 검색할 수 있다.

목적지 검색의 경우, 카카오맵 로컬 데이터를 기반으로 해당 텍스트뷰의 글을 통해 관련 정보를 RecyclerView에 불러온다.

RecyclerView의 아이템을 선택하여 목적지 정보를 가져온다.

앱 중단의 즐겨찾기 RacyclerView와 자주 가는 목적지 RecyclerView의 아이템을 선택하여 목적지 명, 목적지 주소, 목적지의 좌표 등의 정보를 불러올 수 있다.

택시 호출하기 버튼을 선택하여 목적지와 출발지 정보를 저장한다.

## B. 택시 호출하기

**- 경로 설정 화면**

네이버 맵을 통해 출발지와 목적지를 각각 노란색, 초록색 마커로 표시한다.

목적지와 출발지 사이의 경로를 ROS에서 다익스트라 알고리즘을 통해 최적화하여 Firebase에 저장한다.

저장된 경로 정보를 불러와, Polyline을 통해 경로를 그린다.

현재 작성된 경로의 길이를 표기한다.

해당 거리를 운행하는 예상 요금을 표기한다.

후불 결제 버튼을 통해 택시 대기 화면으로 전환한다.

**- 택시 대기 화면**

택시를 호출하는 화면으로, 현재 위치에서 가장 가까운 택시부터 운행여부 및 현재 택시 상태를 판단하여 가장 적합한 자율주행 택시를 호출한다.

**- 택시 확인 화면**

앱 중단에 호출한 차량의 사진과 차량 번호를 제공한다.

승차감과 청결도 상태 등 해당 차량의 정보를 제공한다.

확인 버튼을 통해 택시 호출 화면으로 전환한다.

**- 택시 호출 화면**

네이버 맵을 통해 출발지를 노란색 마커로 표기하며, 현재 택시의 위치를 차량 사진을 이용한 마커를 생성하여 표기한다..

목적지와 출발지 사이의 경로를 ROS에서 다익스트라 알고리즘을 통해 최적화하여 Firebase에 저장한다.

저장된 경로 정보를 불러와, Polyline을 통해 경로를 그린다.

InfoWindow를 통해 현재 차량과 출발지 사이의 거리와 도착까지 남은 시간을 제공한다.

앱 하단에 차량의 정보와 현재 차량의 위치, 돌방상황을 대비하여 차주와 통화, 문자가 가능하다.

택시가 출발지에 도착하면 출발전 화면으로 전환한다.

## C. 택시 운행

**- 출발 전 화면**

앱 중단에 해당 차량의 사진과 번호를 제공한다.

앱 하단 왼쪽에 탑승하기 버튼을 통해 차문을 열고 닫을 수 있다.

앱 하단 오른쪽에 사진찍기 버튼을 통해 출발 전 차량 상태 저장 화면으로 전환한다.

앱 하단에 이동하기 버튼을 통해 택시 운행 화면으로 전환한다.

**- 출발 전 차량 상태 저장 화면**

그리드뷰로 표시된 플러스 버튼을 선택하여 카메라에 접근한다.

이때, SDK 21 이상의 기기만 퍼미션 허용 여부를 묻고 이용할 수 있다.

사진은 최대 8개까지 찍을 수 있다.

사진 저장하기 버튼을 선택하여 해당 사진을 Firebase Storage를 통해 사진을 저장한다.

이후, 저장된 사진의 정보를 Firestore DB에 저장한다.

저장이 완료되면 출발 전 화면으로 돌아간다.

**- 택시 운행 화면**

네이버 맵을 통해 목적지를 노란색 마커로 표기하며, 현재 택시의 위치를 차량 사진을 이용한 마커를 생성하여 표기한다..

목적지와 출발지 사이의 경로를 ROS에서 다익스트라 알고리즘을 통해 최적화하여 Firebase에 저장한다.

저장된 경로 정보를 불러와, Polyline을 통해 경로를 그린다.

InfoWindow를 통해 도착까지 남은 시간을 제공한다.

앱 하단에 차량의 정보와 현재 차량과 목적지 사이의 거리, 현재 요금을 제공한다.

**- 도착 후 화면**

앱 중단에 해당 차량의 사진과 번호를 제공한다.

앱 하단 왼쪽에 해당 차량의 승차감과 청결도를 판단하여 ratingBar로 평가할 수 있다..

앱 하단 오른쪽에 사진찍기 버튼을 통해 도착 후 차량 상태 저장 화면으로 전환한다.

앱 하단에 결제하기 버튼을 통해 택시 운행 화면으로 전환한다.

**- 도착 후 차량 상태 저장 화면**

그리드뷰로 표시된 플러스 버튼을 선택하여 카메라에 접근한다.

이때, SDK 21 이상의 기기만 퍼미션 허용 여부를 묻고 이용할 수 있다.

사진은 최대 8개까지 찍을 수 있다.

사진 저장하기 버튼을 선택하여 해당 사진을 Firebase Storage를 통해 사진을 저장한다.

이후, 저장된 사진의 정보를 Firestore DB에 저장한다.

저장이 완료되면 도착 후 화면으로 돌아간다.

**- 결제하기 화면**

BootPay를 통해 결제를 구현한다.

신용카드, 체크카드 등 카드로 결제가 가능하며 카카오페이, 네이버페이 등 간편결제도 가능하다.

결제가 완료되면 유저 홈 화면으로 이동한다

## D. 기타 정보

**- 로그인 화면**

앱 중단에 아이디와 비밀번호를 누르고 로그인 버튼을 누르면 로그인이 가능하다.

로그인에 성공하면 유저 홈 화면으로 이동한다.

앱 하단의 Github, Facebook, Google 버튼을 누르면 SNS 로그인이 가능하다.

SNS 로그인 시 가입 이력이 있다면 유저 홈화면, 가입 이력이 없다면 회원가입 화면으로 이동한다.

앱 하단의 비밀번호 찾기 버튼을 누르면 비밀번호 찾기 화면으로 이동한다.

앱 하단의 회원가입 버튼을 누르면 회원가입 화면으로 이동한다.

**- 비밀번호 찾기 화면**

앱 중단에 가입되어있는 이메일 아이디를 적고 앱 하단의 비밀번호 찾기 버튼을 누르면 해당 이메일로 비밀번호 찾기 메일이 발송된다.

**- 회원가입 화면**

앱 상단부터 순서대로 프로필사진, 이메일 아이디, 비밀번호, 전화번호, 집 주소, 회사 주소 입력이 가능하다.

SNS 로그인을 통해 회원가입 화면으로 이동하면 프로필 사진, 이메일 아이디, 이름은 자동으로 입력이 되어있고, 비밀번호 입력 창은 비활성화된다.

앱 상단 프로필 사진의 카메라 버튼을 누르면 갤러리에서 프로필 사진을 설정할 수 있다.

아이디와 이름은 최초 가입시에만 설정이 가능하다.

전화번호를 입력하고 우측의 인증 버튼을 눌러 인증했을 시에만 회원가입이 가능하다.

집/회사 주소는 선택 사항이고 Kakao 주소 API 검색 기능을 지원한다.

앱 하단의 회원 가입 버튼을 누르면 등록이 가능하다.

**- 제공자 등록 화면**

앱 상단부터 순서대로 차량 사진, 차량 이름, 차량 번호 입력이 가능하다.

앱 상단 프로필 사진의 카메라 버튼을 누르면 갤러리에서 차량 사진을 설정할 수 있다.

앱 하단의 차량 제공자 등록 버튼을 누르면 등록이 가능하다.

**- 마이 페이지 화면**

앱 상단에 사용자의 사진과 이름을 제공하고, 이용 회수에 따라 등급이 나뉘어 진다.

앱 중단에 즐겨찾기 수정, 회원정보 수정, 비밀번호 변경, 로그아웃, 회원탈퇴로 5개의 메뉴가 있다.

로그아웃 버튼을 누르면 로그인 화면으로 전환하게 되고 나머지 버튼을 누르면 해당 버튼에 맞는 호면으로 이동한다.

앱 하단에는 탑승했던 택시 정보가 Recycler View로 출력되고 버튼을 누르면 상세 정보가 출력된다.

앱 하단 오른쪽에 챗봇 버튼을 누르면 챗봇 화면으로 전환된다.

**- 즐겨찾기 수정 화면**

앱 상단에 즐겨찾기 등록을 위한 주소 검색이 가능하고 앱 하단의 추가하기 버튼을 누르면 즐겨 찾기 등록이 된다.

앱 상단에 주소 검색 시 Kakao 주소 API로 검색 기능을 지원한다.

앱 중단에 등록된 즐겨찾기 목록을 Recycler View로 출력되고 Recycler View Item 우측의 별 아이콘을 누르면 삭제를 위한 Dialog가 출력된다.

**- 회원정보 수정 화면**

앱 상단부터 순서대로 프로필사진, 전화번호, 집 주소, 회사 주소 수정이 가능하다.

앱 상단 프로필 사진의 카메라 버튼을 누르면 갤러리에서 프로필 사진을 설정할 수 있다.

아이디와 이름은 최초 가입시에만 설정이 가능하다.

전화번호 변경 시 우측의 편집 버튼을 누르면 변경이 가능하다.

전화번호 변경 시 우측의 인증 버튼을 눌러 인증했을 시에만 변경이 가능하다.

집/회사 주소 변경 시 우측의 편집 버튼을 누르면 주소 입력 Dialog가 출력되고 Kakao 주소 API 검색 기능을 지원한다.

**- 비밀번호 변경 화면**

앱 중단에 기존 비밀번호를 입력해 재검증을 성공했을 시에만 비밀번호가 변경 가능하다.

앱 중하단에 변경 비밀번호를 2번 입력해서 동일한 값일 때 변경이 가능하다.

**- 회원탈퇴 화면**

앱 우측 아래에 스위치 버튼을 누르고 앱 하단의 회원탈퇴 버튼을 누르면 확인 Dialog가 출력된다.

회원탈퇴 시 Firabase의 Auth, Firastore, Storage 내 관련정보가 모두 삭제된다.

**- 탑승했던 택시 화면**

탑승 택시 목록이 Recycler View로 출력이 되며 Recycler View Item을 누르면 차량 상세 내용으로 이동한다.

**- 차량 상세 조회 화면**

앱 상단에 차량 사진, 차량 번호가 출련된다.

앱 중단에 위에서부터 차례대로 승차감/청결도 평점, 출발지/목적지, 소요시간, 운행 거리, 요금이 출력된다.

앱 중단의 승차감/청결도 평점을 누르면 평점 재변경이 가능하다.

앱 하단의 1:1 채팅 버튼을 누르면 해당 차량 제공자와의 1:1 채팅화면으로 이동한다.

**- 1:1 채팅 화면**

앱 상단에 상대방의 고유 아이디를 출력한다.

유저가 보낸 메시지는 우측, 상대가 보낸 메세지는 좌측에 Recycler View로 출력된다.

앱 하단에 메시지를 입력하고 앱 하단 오른쪽의 전송 버튼을 누르면 메시지가 전송된다.

메시지가 전송되면 자동으로 하단으로 스크롤 된다.

**- 챗봇 화면**

앱 화면에 챗봇 기본 목록이 Recycler View로 출력된다.

챗복 목록을 클릭하면 해당 하위 목록들이 출력된다.

앱 하단 왼쪽의 “처음으로” 버튼을 누르면 최상위 목록(처음 목록)들이 출력된다.

앱 하단 중앙의 “이전으로” 버튼을 누르면 이전 목록들이 출력된다.

앱 하단 오른쪽의 “상담사 연결” 버튼을 누르면 상담사 전화번호가 출력된다.

메시지가 전송되면 자동으로 하단으로 스크롤 된다.

**- 제공자 화면**

앱 상단에 유저 사진, 이름, 수익 금액이 출력된다.

앱 중단에 제공자 차량 사진, 차량 번호가 출력된다.

앱 하단에 차량 평점, 운행 여부, 마감 시간이 출력된다.

운행 여부 스위치를 통해 운행 여부를 정할 수 있다.

앱 하단에 이용자 정보가 Recycler View로 출력되고 버튼을 누르면 이용자 정보 화면으로 이동한다.

**- 이용자 정보 화면**

앱 상단에 이용자의 프로필 사진과 이름이 출력된다.

앱 중단에 출발 전/후 차량 내부 사진을 출력한다.

앱 하단에 차량 평점을 출력한다.

앱 하단의 1:1 채팅 버튼을 누르면 해당 이용자와 1:1 채팅화면으로 이동한다.