

# Walking Mate

## project.D

2016314670 문정연

2014314143 김규용

2018311304 성보경

2016311561 송유호

## Contents

1 Entity Type & Key

---

2 Technical Study

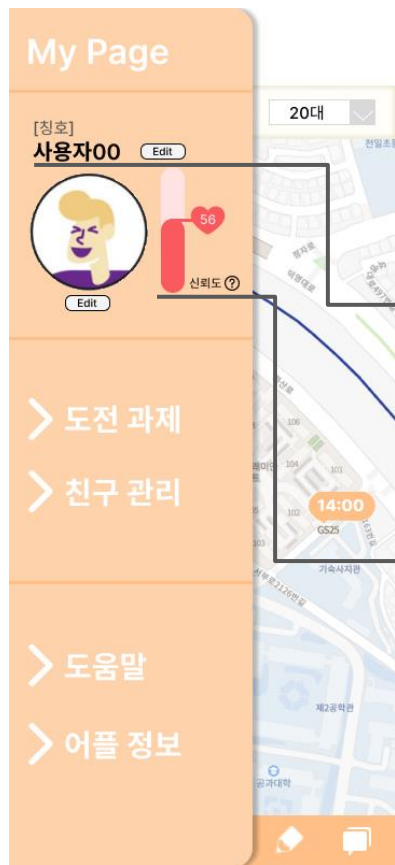
---

3 Planning

---



# Entity Type & Key

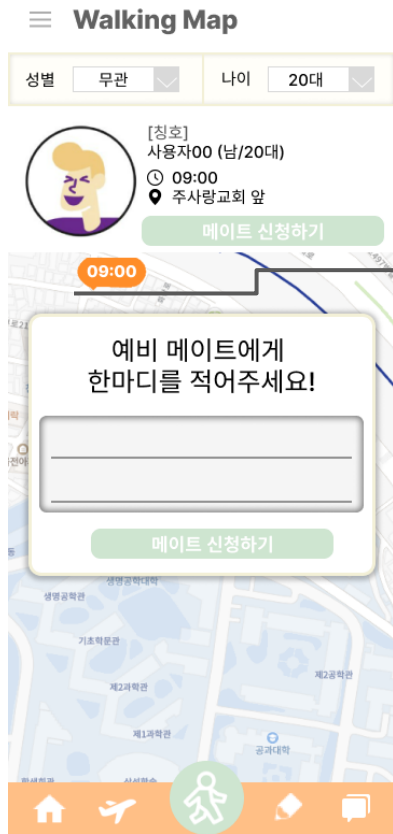


고객 정보

ID, 이름, 성별, 나이, 닉네임

고객\_신뢰도

ID, 제재 내역, 신뢰도



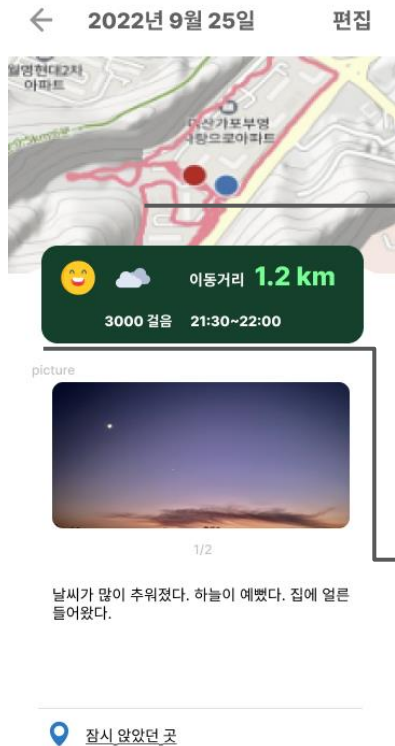
산책 핀

ID, 출발 시간, 출발 위치, 노출 필터링,  
성사 여부

게시판

ID, 기간, 동선(좌표set), 노출 필터링,  
성사 여부, 제목, 내용





추적

ID, 출발 시간, 도착 시간, 출발 좌표, 도착 좌표,  
마커, 이동 거리, 이동 시간, 이동 좌표set

게시판

ID, 날짜, 동선(이미지), 마커, 감정, 날씨,  
이동 시간, 이동 거리, 제목, 내용

## 채팅

	사용자1	10/02	20분 전
	message		
	사용자2	10/11	1시간 전
	message		
	사용자3	10/08	3시간 전
	message		

채팅

ID1, ID2, 메시지 내용, 메시지 시간

도전  
과제ID, 거리 항목, 신뢰도 항목, 친구 항목,  
연속 항목, 선택 칭호

## ← 도전, 과제



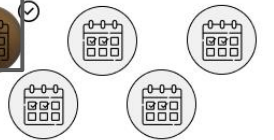
건강한 워커 (5/5)



믿음직한 워커 (3/5)



꾸준한 워커 (1/5)

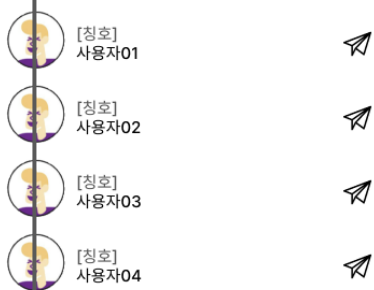


사교적인 워커 (4/5)



← 친구 관리 

목록(4) 요청(1)



← 친구 관리 

목록(4) 요청(1)

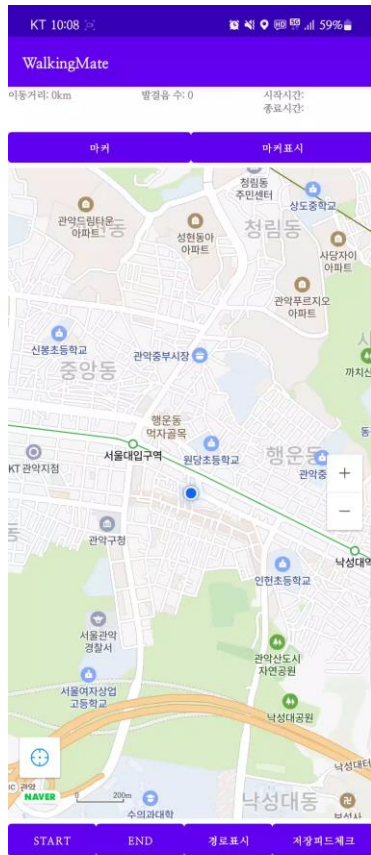


친구

ID, 친구 목록(IDset), 받은 요청(IDset)

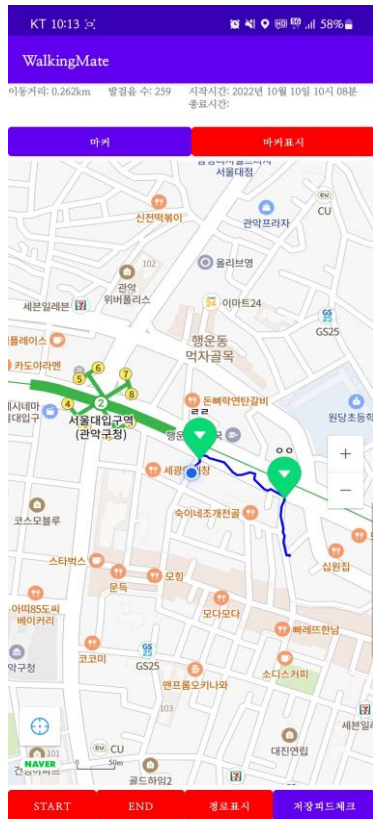


# Technical Study



- `sensorManager`: 만보기 기능 제공
- `addOnLocationChangeListener`: 좌표가 바뀔 때마다 이 함수를 호출하여 최신의 좌표 데이터를 수집함
- `coordList`: 경로추적을 진행하며 받아온 모든 좌표
  - > 좌표마다 선을 이어서 경로를 표시하고 거리를 계산함
  - > 때문에 1 초과해야 한다는 조건이 따름
- `cameraUpdate`: 내 위치로 카메라를 실시간으로 이동
- `pathOverlay`: 좌표set을 이용하여 지도에 경로를 기록
- `FeedData(coordList, markerList, timecheck, step, displaycement)`
  - > 객체를 String 형태로 바꿔서 저장
  - > Data를 다시 불러올 때는 decode 과정 필요

- sensorManager: 만보기 기능 제공
- addOnLocationChangeListener: 좌표가 바뀔 때마다 이 함수를 호출하여 최신의 좌표 데이터를 수집함
- coordList: 경로추적을 진행하며 받아온 모든 좌표
  - > 좌표마다 선을 이어서 경로를 표시하고 거리를 계산함
  - > 때문에 1 초과해야 한다는 조건이 따름
- cameraUpdate: 내 위치로 카메라를 실시간으로 이동
- pathOverlay: 좌표set을 이용하여 지도에 경로를 기록
- FeedData(coordList, markerList, timecheck, step, displaycement)
  - > 객체를 String 형태로 바꿔서 저장
  - > Data를 다시 불러올 때는 decode 과정 필요



- `sensorManager`: 만보기 기능 제공
- `addOnLocationChangeListener`: 좌표가 바뀔 때마다 이 함수를 호출하여 최신의 좌표 데이터를 수집함
- `coordList`: 경로추적을 진행하며 받아온 모든 좌표
  - > 좌표마다 선을 이어서 경로를 표시하고 거리를 계산함
  - > 때문에 1 초과해야 한다는 조건이 따름
- `cameraUpdate`: 내 위치로 카메라를 실시간으로 이동
- `pathOverlay`: 좌표set을 이용하여 지도에 경로를 기록
- `FeedData(coordList, markerList, timecheck, step, displacement)`
  - > 객체를 String 형태로 바꿔서 저장
  - > Data를 다시 불러올 때는 decode 과정 필요

## 위젯

- 시작 지점에서 start/end 버튼을 누르면 거리, 이동경로로 측정 시작
- 종료 지점에서 start/end 버튼을 누르면 측정 종료 및 종료 시 액티비티 활성화
- 지점 마킹이 필요한 경우 mark 버튼을 누르면 어플을 실행시켜 마커 및 메모 기능 활성화

## 노티피케이션 바 (위젯과 디자인 유사)

- 어플 내 혹은 위젯으로 시작 버튼을 누르면 활성화
- 활성화 시 잠금화면에서도 조작가능하게끔 설정할 예정
- 종료 지점에서 버튼을 누르면 측정 종료 및 종료 시 액티비티 활성화



기록 전 위젯



기록 중 위젯

## 파이어베이스

The screenshot displays the Android Studio interface. The main editor shows the `build.gradle (app)` file for the `WalkingMate` project. The file content is as follows:

```
sourceCompatibility JavaVersion.VERSION_1_8
targetCompatibility JavaVersion.VERSION_1_8

dependencies {
    implementation 'com.naver.maps:map-sdk:3.15.0'

    implementation 'com.google.android.gms:play-services-location:20.0.0'
    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material:1.6.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.firebase:firebase-firestore:24.3.1'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
    implementation 'com.google.firebase:firebase-analytics:21.1.1'
    implementation 'com.google.firebase:firebase-core:21.1.1'
    implementation 'com.google.firebase:firebase-database:20.0.6'
}
```














The right-hand Assistant pane displays the 'Get started with Cloud Firestore' guide. It includes the following steps and status:

- 1 Connect your app to Firebase  
✓ Connected
- 2 Add Cloud Firestore to your app  
✓ Dependencies set up correctly
- 3 Initialize Cloud Firestore

A note is also present: "NOTE: After adding the SDK, here are some other helpful configurations to consider: Do you want an easier way to manage library versions? You can use the Firebase Android BoM to manage your Firebase library versions and ensure that your..."

## 파이어베이스

- 클라우드 데이터베이스 연동 활성화  
(데이터의 사유화를 막기 위해 어플 데이터 관리자용 구글 계정 생성)
- 이후 실시간 채팅기능 활성화를 위한 인앱 메시징과 리얼타임 데이터베이스 기능, 피드 및 게시판 데이터를 위한 클라우드 스토리지 활성화 예정
- 현재 소셜 로그인 기능의 경우, 파이어베이스와 연동이 쉬운 구글과 페이스북에서는 성별과 연령대를 수집하지 않기 때문에 이 두 로그인을 이용하는 사용자의 경우 추가 데이터를 입력하게 하거나, 이 소셜로그인을 사용하지 않을 예정
- 네이버 로그인과 카카오 로그인의 경우 연령대와 성별을 수집할 수 있어서 이 데이터를 받을 수 있게 설정

- ▶  **Analytics**  
Track and analyze user behavior in your app. [More info](#)
- ▶  **Authentication**  
Sign in and manage users with ease using popular login providers like Google, Facebook, and others. You can even use a custom authentication system. [More info](#)
- ▶  **Realtime Database**  
Store and sync data with this cloud-hosted NoSQL database. Data is synced across all clients in realtime and remains available when your app goes offline. [More info](#)
- ▶  **Cloud Firestore**  
Store and sync your app data with this flexible, scalable NoSQL cloud-hosted database. [More info](#)
- ▶  **Cloud Storage for Firebase**  
Store and retrieve large files like images, audio, and video without writing server-side code. [More info](#)
- ▶  **Cloud Functions for Firebase**  
Automatically run backend code in response to events triggered by Firebase features and HTTPS requests. [More info](#)
- ▶  **Firebase ML**  
Firebase ML is a mobile SDK that brings Google's machine learning expertise to Android and iOS apps in a powerful yet easy-to-use package. [More info](#)
- ▶  **Crashlytics**  
Get clear, actionable insight into app issues that erode your app quality. [More info](#)
- ▶  **Performance Monitoring**  
Gain insight into the performance characteristics of your app. [More info](#)
- ▶  **Test Lab**  
Test your apps against a wide range of physical devices hosted in Google's cloud. [More info](#)
- ▶  **App Distribution**  
Distribute pre-release versions of your app to your trusted testers. [More info](#)
- ▶  **Cloud Messaging**  
Deliver and receive messages and notifications reliably across cloud and device. [More info](#)
- ▶  **In-App Messaging**  
Engage your app's active users by sending them targeted, contextual messages that encourage them to use key app features. [More info](#)

**Planning**



1

앱 디자인 적용 및 테스트

---

2

소셜 로그인 기능 등록

---

3

실시간 채팅 기능 구현

---

**Thank you!**