



2020년 캡스톤 디자인 프로젝트

최종 보고서

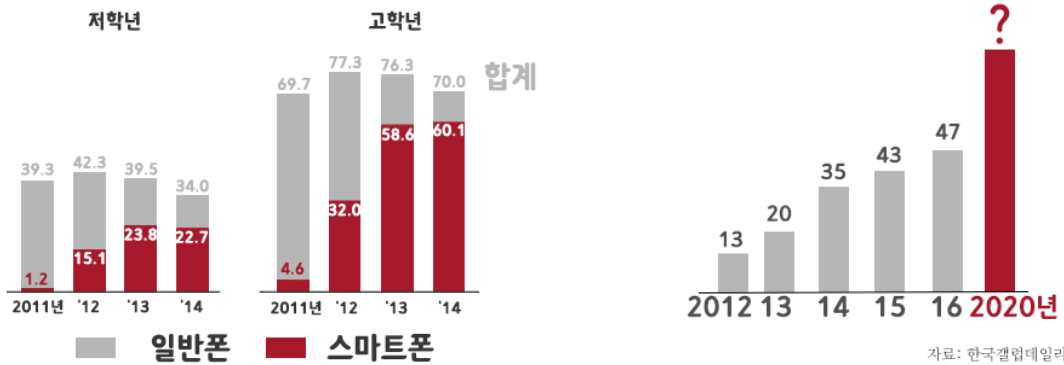
성공회대학교 컴퓨터공학과

팀 이름: 엄마, 길 알려줘

팀원: 안민영, 전지윤, 이애나

1. 소개 및 개발 배경

초등학생 스마트폰 등 휴대폰 보유 비율(단위: %) 60대 이상 노인들의 스마트폰 사용률(단위: %)



▲초등학생 스마트폰 보유 비율

▲60대 이상 노인들의 스마트폰 사용률

많은 연구 및 기술개발에 힘입어, 현재 보행자를 위한 길찾기 서비스는 상당한 수준의 정확도를 갖춘 경로안내를 제공하는 수준에 이르렀다. 그러나 보행자용 내비게이션 서비스에 있어서 지금까지 충분히 논의되지 않았던 사실 중 하나는, 보행에 어려움을 겪는 사람, 즉 '교통약자'일수록 이 서비스를 더욱 필요로 한다는 점이다.¹

2012년 SK텔레콤에 따르면 올해 어르신 휴대폰 활용 교육생을 대상으로 스마트폰 이용 행태 설문조사를 실시한 결과, 전체 응답자 125명 중 스마트폰 사용자가 52%(72명)로 이미 상당수가 스마트폰을 사용하고 있는 것으로 조사됐다. 어르신들이 통화 기능 외에 가장 많이 사용하는 스마트폰 기능은 '애플리케이션'이 58%(42명)로 가장 많았으며, 주로 사용하는 애플리케이션은 길 찾기/대중교통 정보 검색 55%(23명)로 가장 많았다. 반면 어르신들이 스마트폰 사용 시 불편한 점으로는 '스마트폰의 복잡한 기능'을 1위(35%, 25명)로 꼽았다.²

교통약자가 목적지까지 길을 찾아가는 과정에서 발생할 수 있는 문제를 최소화하기 위해 이 어플리케이션을 제작하였다. 이 어플리케이션의 목적은 보호자가 직접 길 찾기 서비스를 설정하고 교통약자의 실시간 위치를 확인하여 교통약자들이 편리하게 지도어플을 사용할 수 있도록 도왔다. 타겟층을 초등학생을 비롯한 자녀, 60대 이상 노인, 장애인으로 설정하였다.

¹ 방윤식,『교통약자를 위한 공간정보 플랫폼 기술 및 앱 개발』,『Science & Technology Policy』,2016년,46쪽

² 박철근, "노인들이 자주 쓰는 앱은 '길 찾기'", 이투데이, 2012.06.28(<https://www.etoday.co.kr/news/view/602620>)

2. 개발 환경

A. 소프트웨어

- i. Server : Spring Tool Suite 4.6 + MySQL
- ii. Android : Android Studio 4.0
- iii. iOS : xcode

B. 개발 언어

- i. Server : Java 8.0
- ii. Android : Java 8.0
- iii. iOS : swift 5.0

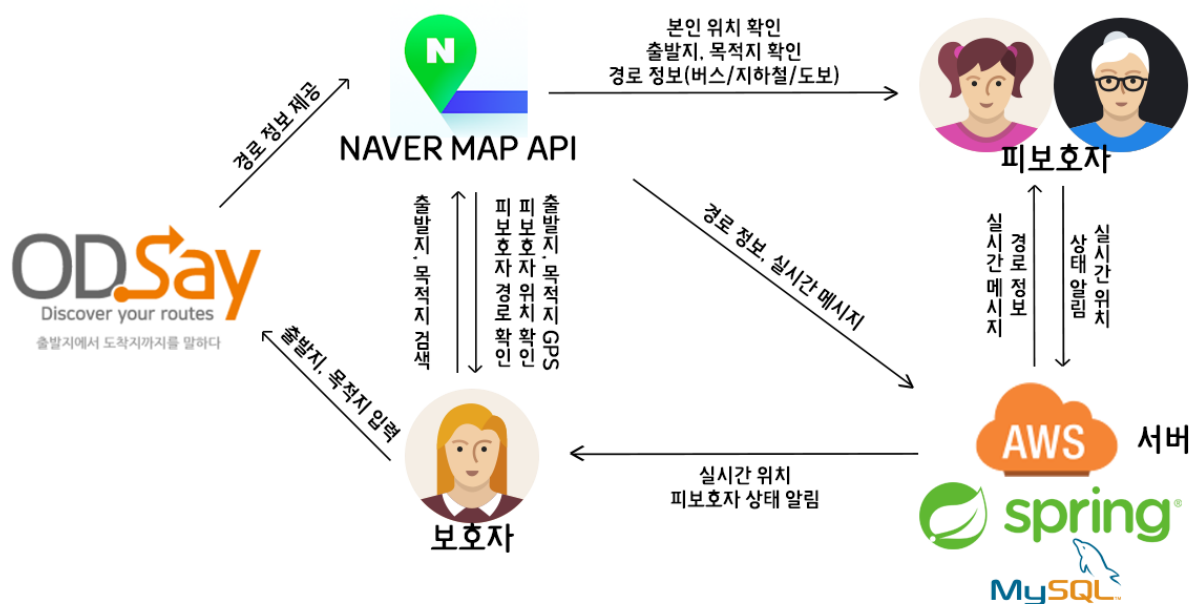
C. Framework

- i. Server : Spring JPA

D. 타겟 플랫폼

- i. Android : API 24 (Nougat) 이상
- ii. iOS : 13.0 이상

3. 어플리케이션 흐름도

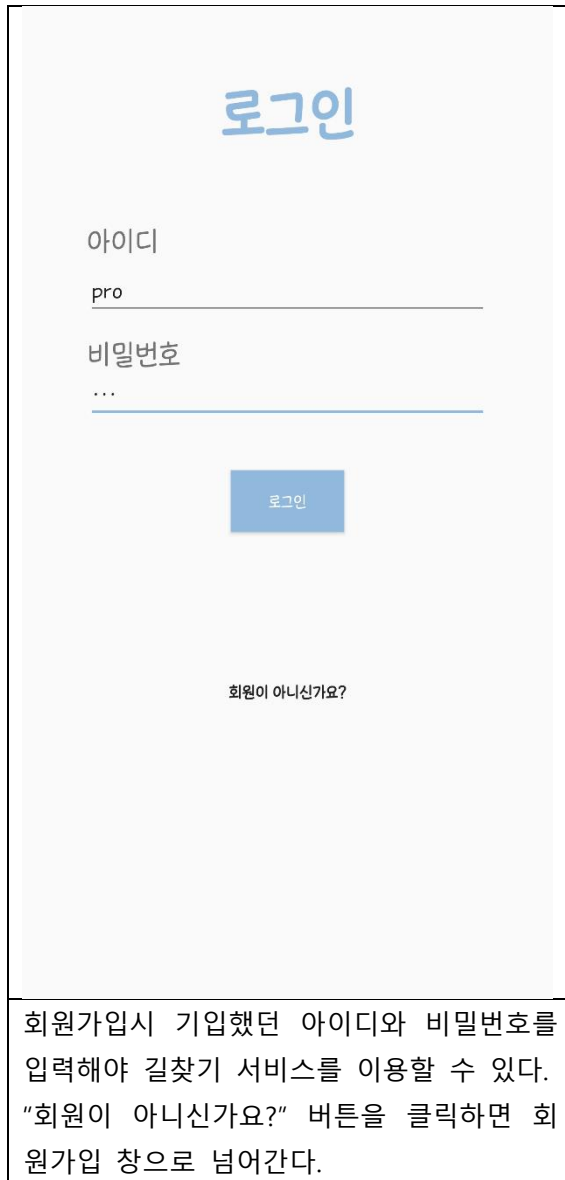


4. 코드 / 구현 기능 및 사용자 매뉴얼

A. [공통]

i. 로그인 / 로그아웃

1. 로그인



The image shows a login form with a light gray background. At the top, the word '로그인' (Login) is written in a large, blue, sans-serif font. Below it, there are two input fields. The first is labeled '아이디' (ID) and contains the text 'pro'. The second is labeled '비밀번호' (Password) and contains three dots '...'. Below the password field is a blue rectangular button with the text '로그인' (Login) in white. At the bottom of the form, there is a link that says '회원이 아니신가요?' (Are you not a member?).

로그인

아이디
pro

비밀번호
...

로그인

회원이 아니신가요?

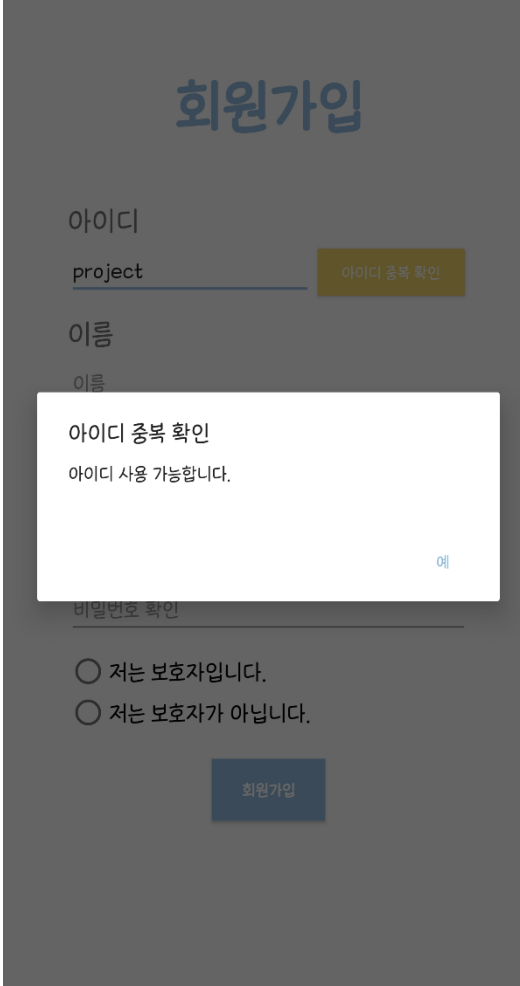

회원이입시 기입했던 아이디와 비밀번호를
입력해야 길찾기 서비스를 이용할 수 있다.
“회원이 아니신가요?” 버튼을 클릭하면 회
원가입 창으로 넘어간다.

2. 로그아웃

로그인을 하고 메뉴화면에서 로그아웃 버튼을 클릭하면 로그아웃이 된다.

ii. 회원가입 / 회원탈퇴


1. 회원가입

	
<p>회원가입시 회원의 개체 무결성을 위해서 중복확인을 꼭 해야 회원가입이 진행이 된다.</p>	<p>회원가입시 요구하는 정보는 아이디, 이름, 비밀번호가 있으며, 본인이 보호자인지 피보호자인지 결정해야만 회원가입이 진행된다.</p>

2. 회원탈퇴

로그인을 하고 메뉴화면에서 회원탈퇴 버튼을 클릭하면 탈퇴할 수 있다.


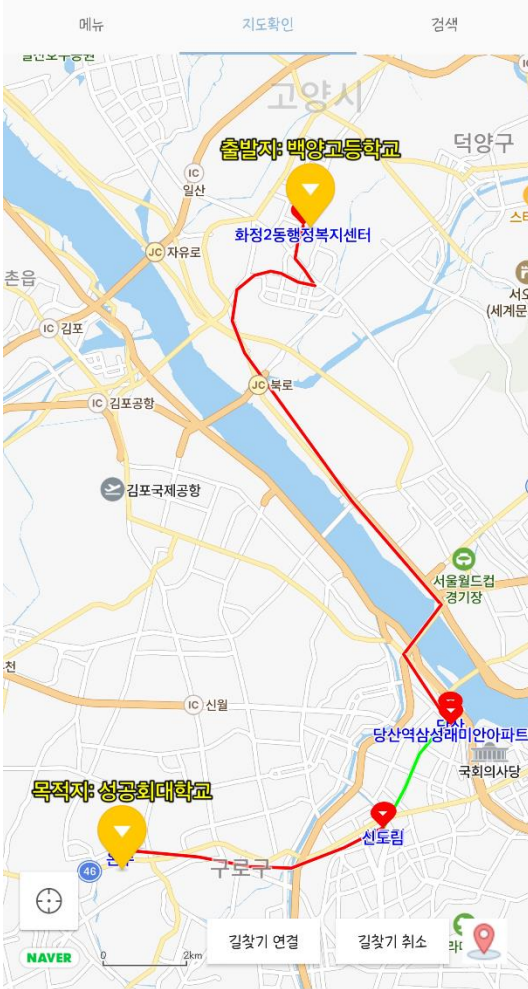
iii. 피보호자 / 보호자 연결

	<p>본인이 보호자일 경우, 피보호자의 아이디를 입력하고 반대로 본인이 피보호자일 경우, 보호자의 아이디를 입력하면 서로를 연결할 수 있다.</p>
---	--

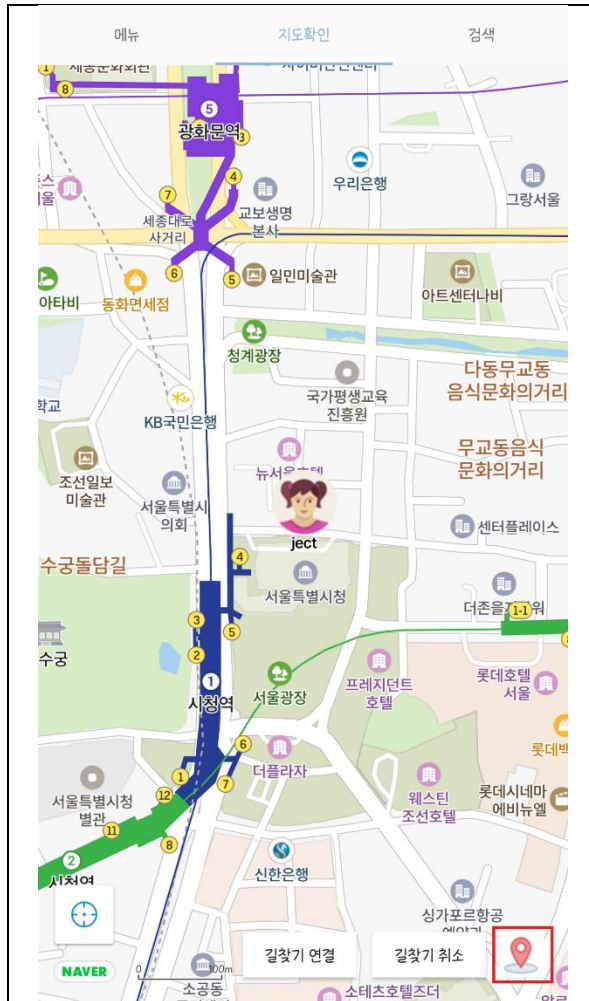

```

        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            /*코드 생략*/
        }
    })
    .show();
    return true;
}
} else { // 로그인이 되어있지 않고 피보호자 연결을 하지 않았다면
    Toast.makeText(MapFragmentActivity.this, "로그인하고 피보호자와 연결하
    신 후에\n길찾기 서비스를 이용할 수 있습니다.", Toast.LENGTH_SHORT).show();
    return false;
}
return false;
}
});
//[END]

```

	
<p>검색 페이지에서 “검색” 버튼을 클릭하면 하단에 길찾기 정보를 제공받을 수 있다. (버스, 지하철, 도보)</p>	<p>지도에선 피보호자가 환승할 곳(승하차 포함)을 확인할 수 있다.</p>
<pre> //[START] 검색버튼을 누르면 길찾기 서비스를 이용 btSearch.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { /* 코드 생략 */ odsayService.requestSearchPubTransPath(String.valueOf(start.longitude), String.valueOf(start.latitude), String.valueOf(destination.longitude), String.valueOf(destination.latitude), "0", "0", "0", onResultCallbackListener); } }); //[END] </pre>	

ii. 피보호자의 위치 확인

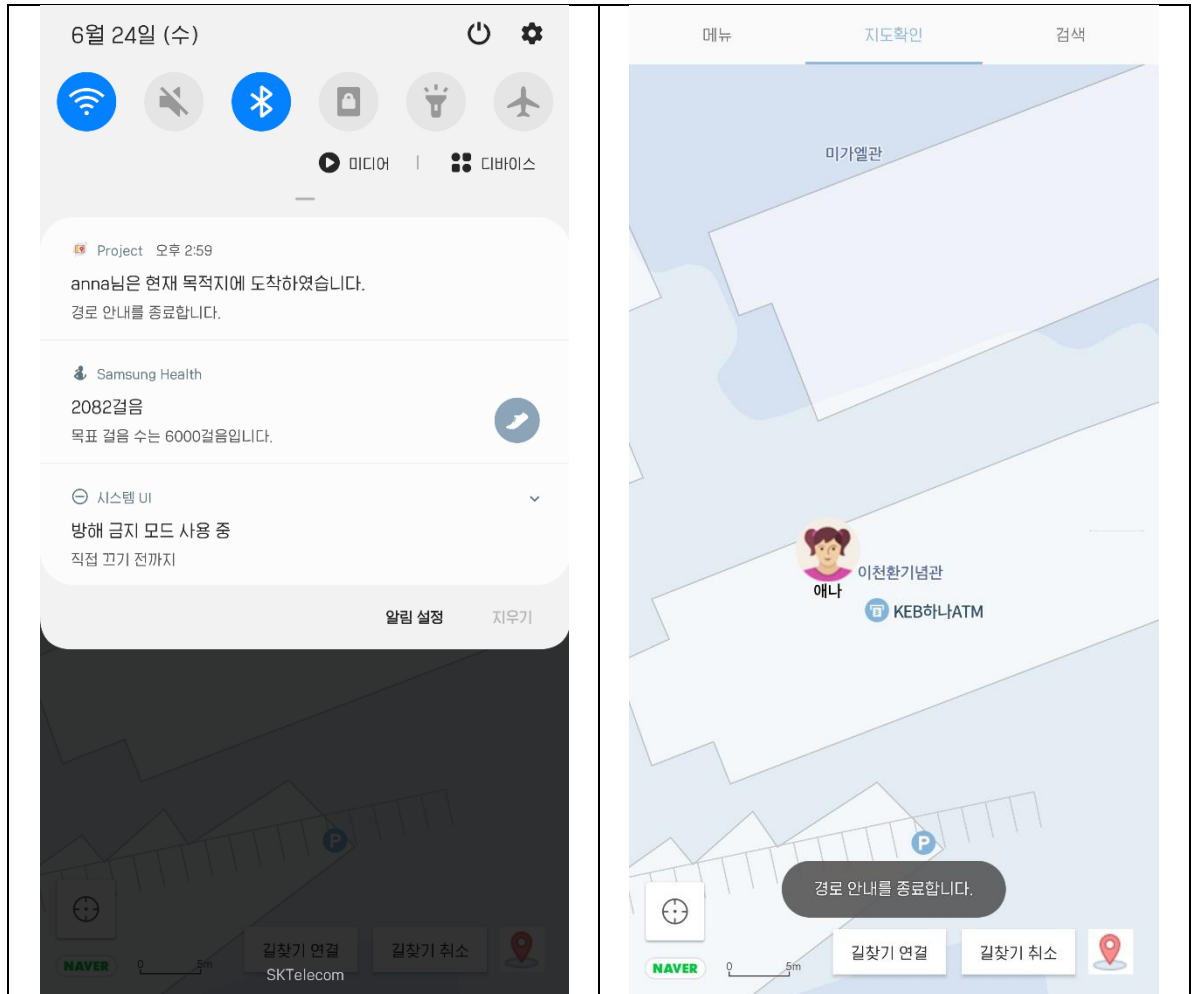


기본적으로 보호자는 피보호자의 위치를 계속 확인할 수 있다.

빨간색으로 표시한 버튼을 누르면 카메라의 위치가 피보호자의 위치로 이동한다.

```
// [START] 카메라 버튼 -> 피보호자의 위치 카메라로 보여줌
btCamera.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (wardLatLng != null) { //wardLatLng은 피보호자의 위치 변수
            CameraUpdate camera = CameraUpdate.scrollTo(wardLatLng);
            naverMap.moveCamera(camera);
        }
    }
});
// [END]
```

iii. 피보호자가 목적지 근처에 왔을 때 푸시 알림



보호자는 목적지와 피보호자의 위치가 100m 이하일 때 push알림을 받는다.

보호자에게 온 푸시 알림을 클릭하면 자동으로 길찾기가 취소된다.

```

if (destination != null) {
    if (destination.distanceTo(wardLatLng) <= 100) { // [START] 목적지와 피보호자의 위치가 100m 이하일 때

        // [START] 목적지 근처에 피보호자가 도착하면 푸시 알림
        pushNotification("목적지 도착", user.getWardId() + "님은 현재 목적지에 도착 하였습니다.", "경로 안내를 종료합니다.");
        // [END]

        // [START] Driving database 삭제하고 초기화 시키고 종료
        dataService.delete.deleteDriving(user.getId()).enqueue(new
        Callback<ResponseBody>() {
            @Override
            public void onResponse(Call<ResponseBody> call,
            Response<ResponseBody> response) {
                Toast.makeText(MapFragmentActivity.this, "경로 안내를 종료합니
다.", Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onFailure(Call<ResponseBody> call, Throwable t) {
                Log.d("경로안내", "종료 실패");
            }
        });
        // [END]

        // [START] 위치정보 삭제
    
```

```

startMarker.setMap(null);
destinationMarker.setMap(null);
setTransitMarkerNull();

startMarker = new Marker();
destinationMarker = new Marker();
start = null;
destination = null;
// [END]
}
}
// [START] 푸시 알림
public void pushNotification(String ticker, String title, String text) {
    final NotificationManager notificationManager = (NotificationManager)
    MapFragmentActivity.this.getSystemService(MapFragmentActivity.this.NOTIFICATIO
    N_SERVICE);

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        notificationManager.createNotificationChannel(new
        NotificationChannel(CHANNEL_ID, CHANEL_NAME,
        NotificationManager.IMPORTANCE_DEFAULT));
        builder = new NotificationCompat.Builder(MapFragmentActivity.this,
        CHANNEL_ID); // 하위 버전일 경우
    } else {
        builder = new NotificationCompat.Builder(MapFragmentActivity.this,
        CHANNEL_ID);
    }

    final Intent intent = new
    Intent(MapFragmentActivity.this.getApplicationContext(),
    MapFragmentActivity.class);

    // 푸시 알림을 터치하여 실행할 작업에 대한 Flag 설정 (현재 액티비티를 최상단으로
    올린다 | 최상단 액티비티를 제외하고 모든 액티비티를 제거한다)
    intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP |
    Intent.FLAG_ACTIVITY_CLEAR_TOP);

    // 앞서 생성한 작업 내용을 Notification 객체에 담기 위한 PendingIntent 객체 생성
    PendingIntent pendnoti = PendingIntent.getActivity(MapFragmentActivity.this,
    0, intent, PendingIntent.FLAG_UPDATE_CURRENT);
    // 푸시 알림 설정

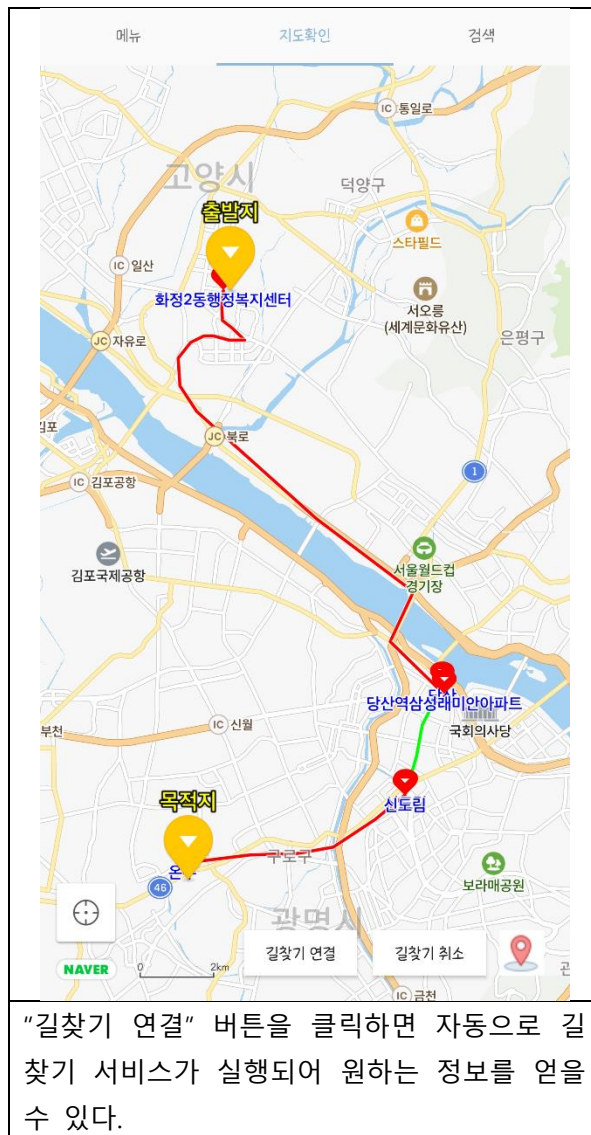
    builder.setSmallIcon(R.drawable.icon).setTicker(ticker).setWhen(System.current
    TimeMillis())
        .setNumber(1).setContentTitle(title).setContentText(text)
        .setDefaults(Notification.DEFAULT_SOUND |
        Notification.DEFAULT_VIBRATE).setContentIntent(pendnoti).setAutoCancel(true).s
        etOngoing(true);

    // NotificationManager를 이용하여 푸시 알림 보내기
    notificationManager.notify(1, builder.build());
}
// [END]

```

C. [피보호자]

- i. 보호자가 찾아 놓은 길을 연결하여 지도에 표시



ii. 본인 위치 확인 및 서버에 전송

본인 위치를 확인할 수 있으며, 서버에 계속 전송되어 보호자는 피보호자의 위치를 계속해서 확인할 수 있다.

```
naverMap.addOnLocationChangeListener(new NaverMap.OnLocationChangeListener() {
    @Override
    public void onLocationChange(@NonNull Location location) {
        if (user.getId() != null && user.getWardId() != null &&
            user.getId().equals(user.getWardId())) { // 피보호자라면
            // 위치 정보를 가져온다.
            double latitude = location.getLatitude();
            double longitude = location.getLongitude();
            wardLatLng = new LatLng(latitude, longitude);
            Map<String, Double> map = new HashMap<>();
            map.put("latitude", latitude);
            map.put("longitude", longitude);

            // 서버에 저장
            dataService.update.updateLocation(user.getId(), map).enqueue(new
            Callback<Member>() {
                @Override
                public void onResponse(Call<Member> call, Response<Member>
                response) {

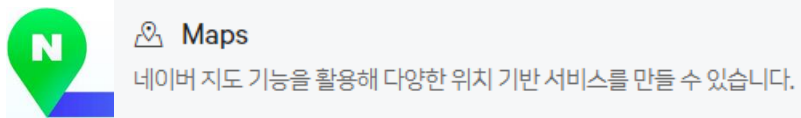
                }

                @Override
                public void onFailure(Call<Member> call, Throwable t) {

                }
            });
        }
    }
});
```

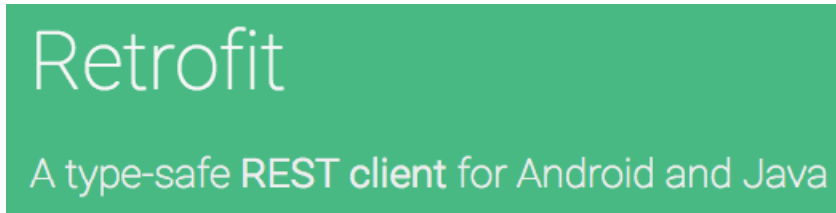
5. 활용한 라이브러리, open API, Framework

A. Naver Cloud Maps - Mobile Dynamic Map



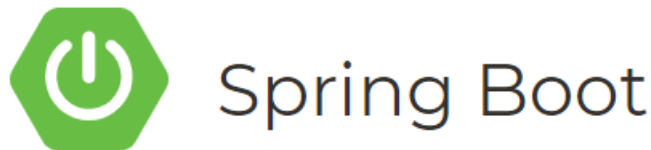
최신 지도 서비스를 제공하는 네이버 지도 API가 제공하는 지도와 좌표에서부터 카메라, 사용자 인터페이스, 위치, 오버레이, 마커, 경로선등을 사용하였다.

B. Java retrofit2



Square사에서 제공하는 오픈소스 라이브러리이다. REST API에서 사용하는 CRUD기능을 모두 제공한다. POST(create), GET(read), PUT(update), DELETE(delete) 또한 annotation(@)을 활용하여 손쉽게 통신 코드를 작성할 수 있다.

C. Spring Boot JPA



Java Persistence API (자바 ORM 기술에 대한 API 표준 명세)이다. 한마디로 ORM을 사용하기 위한 인터페이스를 모아둔 것 이라고 볼 수 있다. 자바 어플리케이션에서 관계형 데이터베이스를 사용하는 방식을 정의한 인터페이스이다.

D. ODsay



대중교통 길찾기 API는 출발점의 경위도 좌표(SX, SY)에서 도착점의 경위도 좌표(EX, EY)로 가는 대중교통 결과를 반환한다. 따라서 경위도 좌표(SX,SY,EX,EY)가 대중교통 길찾기 API의 필수 파라미터이다. 필수 파라미터인 경위도 좌표는 대중교통 정류장 검색 API, 반경내 대중교통 POI 검색 API, 지도 위 대중교통 POI검색 API의 결과에서 표출되는 좌표를 이용하거나 Naver, Daum, Google을 포함한 모든 Map API 이용하여 지도 위에서 좌표를 취득할 수 있다.

6. 일정 및 회고

A. 업무 분담

담당	이름	담당 업무
팀장	안민영	<ul style="list-style-type: none"> - 어플리케이션 기획 - Android 어플리케이션 개발 (지도 구현, 길 찾기) - 프로젝트 발표 자료 작성
팀원	전지윤	<ul style="list-style-type: none"> - 어플리케이션 설계 - 웹 서버 설계 및 구축 (AWS EC2 + Spring JPA) - 데이터 베이스 설계 및 구축 (MySQL) - Android 어플리케이션 개발 (로그인, 회원가입, 길 찾기)
팀원	이애나	<ul style="list-style-type: none"> - 어플리케이션 화면 설계 - Android 어플리케이션 개발 (길 찾기) - iOS 어플리케이션 개발

B. 프로젝트 일정

	3월		4월				5월				6월			
주제 선정														
웹 서버 설계 및 구축														
로그인/회원가입														
지도 서비스														
보호자/피보호자 연결														
피보호자 위치 서비스														
경로 서비스														
발표자료 제작														

C. 한계 및 향후계획

i. 한계

1. 지하에 있는 경우 정확한 위치 파악이 불가능하다.
2. 백그라운드에서 실행되는 GPS 기능이 계속 사용되는 경우 배터리 소모량이 크다.
3. 네이버 Search Places 지원 중지로 장소 검색이 불가능하다.

ii. 향후계획

1. iOS 개발 예정 - 현재 android 개발 진행중에 있다.
2. 여러 개의 경로를 선택할 수 있는 창을 추가한다