

프로젝트 보고서

프로젝트 명: 미세먼지 정보제공 App ‘먼지아나’



과 목 명	소프트웨어 공학
담당교수	강동현 교수님
학 번 / 이 름	20203041/공태현 20183098/최진서
조 편 성	M 조

1. 프로젝트 개요.....	3
1.1 프로젝트 명.....	3
1.2 진행 동기.....	3
1.3 개발 목적.....	4
1.4 동작환경 및 개발도구.....	4
2. 유사 서비스.....	4
2.1 유사 서비스와의 차이점 및 장·단점.....	5
3. 요구사항.....	5
3.1 기능적 요구사항.....	5
3.2 비 기능적 요구사항.....	6
3.3 도메인 요구사항.....	6
3.4 구조적 명세.....	6
4. Use-Case Diagram.....	9
4.1 Use-Case description.....	9
5. Class Diagram.....	11
5.1 MainActivity Class.....	11
5.2 CountySelect Class.....	13
5.3 ProvinceSelect Class.....	15
5.4 ListViewAdapter Class.....	17
5.5 Splash Class.....	18
5.6 MapActivity Class.....	19
5.7 CountyDustInfo Class.....	21
5.8 AdjacentStationList Class.....	22
5.9 StationLog Class.....	23
5.10 StationActivity Class.....	24
5.11 ViewHolder Class.....	25
5.12 TMapView Class.....	25
5.13 TMapData Class.....	26

5.14 TMapPoint Class.....	27
5.15 TMapMarkerItem Class.....	28
5.16 TMapPOIItem Class.....	28
5.17 CoordPoint Class.....	29
5.18 TransCoord Class.....	30
6. Sequence Diagram.....	30
6.1. 지역설정.....	30
6.2. 지역별 미세먼지 정보조회.....	31
6.3. 주변 측정소 조회.....	33
6.4. 측정소별 주간 예보.....	35
7. App UI.....	38
8. Test.....	39
8.1 기능테스트.....	39
8.1.1 지역별 미세먼지 정보조회.....	39
8.1.2 지역설정.....	41
8.1.3 주변 측정소 조회.....	43
8.1.4 측정소별 주간 예보조회.....	44
8.2 실행 테스트.....	47

1. 프로젝트 개요

1.1 프로젝트 명

미세먼지 정보제공 App ‘먼지아나’

1.2 진행 동기

중국에서의 코로나 19 확산세가 진정되고 공장이 재가동 하기 시작하면서 그에 따라 발생한 미세먼지가 겨울철 시베리아 기단을 타고 남하하여 본격적으로 국내 대기에 중국 발 미세먼지 유입이 증가하고 있다. 미세먼지는 주로 석탄이나 석유 같은 화석연료의 연소과정에서 배출되며 세계보건기구(WHO) 산하 국제 암연구소에 의해 1군 발암물질로 지정되어 있다. 직경 10 μ m 이하의 먼지를 미세먼지라 정의하는데, 이는 머리카락보다

훨씬 작은 크기로 숨을 쉴 때 폐 속 깊숙이 흡착된다. 따라서 지속적으로 미세먼지에 노출되면 천식, 알레르기 비염 등 호흡기 질환을 유발하고 악화시킬 수 있으며 영아나 노인의 폐 발달을 저해하거나 만성적으로 폐 기능 저하를 유발할 수 있다. 특히 중국은 대기환경 배출허용 규제가 모호하고 대기오염기준이 우리나라의 기준치 보다 높게 책정되어 있기 때문에 겨울철 중국 발 미세먼지는 국민의 건강에 상당한 악영향을 끼친다고 할 수 있다. 따라서 개개인의 호흡기 건강을 유지하기 위해 지역별 미세먼지 정보를 제공하는 서비스의 필요성을 느끼게 되었고 개발을 진행하고자 한다.

1.3 개발 목적

미세먼지 정보제공 App ‘먼지아나’는 사용자에게 실시간으로 지역별 미세먼지 수치에 대한 정보를 제공하는 모바일 애플리케이션이다. 미세먼지 정보는 한국 환경공단과 aqicn.org(World Air Quality Index Project)에서 제공하는 정보를 활용하여 언제 어디서든지 사용자에게 필요한 미세먼지 정보를 제공할 수 있다. 사용자는 거주하고 있는 지역의 행정구역에 대한 평균 미세먼지 수치와 등급을 확인할 수 있고 App 지도상에 해당 행정구역 내 대기 질 측정소의 위치와 측정소별 미세먼지 예보를 조회할 수 있다. 사용자는 이러한 정보를 통해 외출여부를 실내에서 판단할 수 있으며 야외 활동을 하기 전 미세먼지 등급에 따라 호흡기 보호를 위한 대책을 강구할 수 있도록 하여 국민 개개인의 호흡기 건강을 보호하는데 도움을 주는 것을 개발 목적으로 한다.

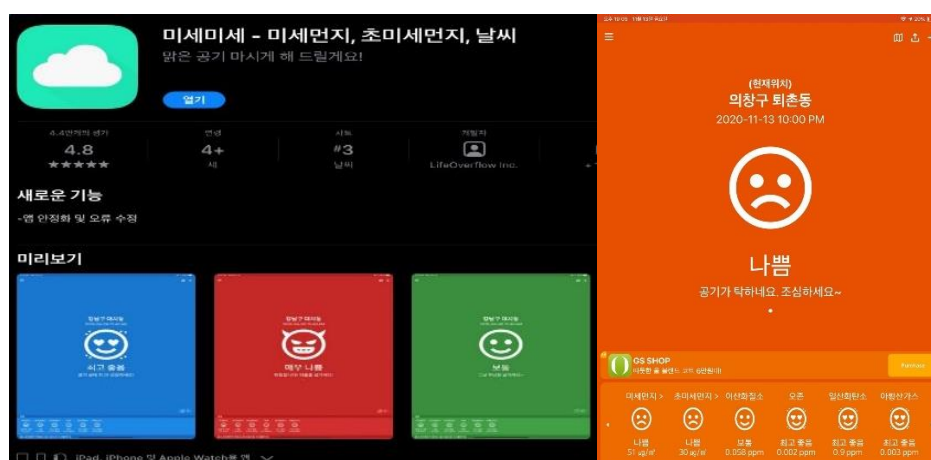
1.4 동작환경 및 개발도구

동작환경: Android 9.0(Pie) 이상

개발언어: Java(OpenJDK 13)

개발도구: Android Studio Android SDK

2. 유사 서비스



[그림 2] 미세먼지 정보제공 App ‘미세미세’

현재 앱스토어에 등록되어 App 중 구현하고자 하는 서비스와 가장 유사한 App 인 ‘미세미세’는 아래와 같은 기능을 가지고 있다. 따라서 개발 시, 실제 서비스 중인 App 이 기능을 어떤 방식으로 구현하였는지 참고하고자 한다.

- 앱을 실행하면 사용자 현재위치의 미세먼지 수치와 등급을 제공한다.
- 대한민국 내 지역별 최소 행정구역 단위로 미세먼지 수치를 조회할 수 있다.
- 사용자가 원하는 지역에 대한 정보를 ‘즐거찾기’ 기능으로 관리할 수 있다.
- 사용자의 현재 위치의 미세먼지 등급이 변경되면 푸시 알림을 제공한다.

2.1 유사 서비스와의 차이점 및 장·단점

구현하고자 하는 App 은 유사 서비스와 다르게 사용자의 세부적인 위치를 기기의 GPS 를 통해 조회하지 않고 사용자로부터 직접 행정구역 단위 까지만 입력 받는다. 또한 입력 받은 정보를 따로 DB 에 저장하지 않으므로 개인정보 유출에 대한 우려가 적다. 그리고 지역별로 대기 질 측정소의 위치와 측정소별 주간 미세먼지 수치예보를 제공하여 거주하고 있는 지역에 대해 보다 정확한 정보를 제공한다는 장점이 있다. 하지만 미세먼지 등급에 대한 정보를 색깔과 텍스트로만 표현하여 등급별로 이모티콘을 사용해 표현한 유사 서비스에 비해 직관성이 떨어진다. 또한 미세먼지 등급 변동에 따른 알림 기능이 없어 사용자가 직접 조회하기 전까지 미세먼지 악화에 대한 선제적인 대응이 불가능하다는 단점도 존재한다.

3. 요구사항

개발하고자 하는 App 의 요구사항은 기능, 제약조건, 실행환경에 따라 크게 3 부분으로 나누고 구조적 명세를 표준화된 방식으로 서술하여 구현하고자 하는 App 을 구체화하였다.

3.1 기능적 요구사항

- 사용자는 원하는 지역의 시간대별 최신 미세먼지 수치 평균값과 등급을 확인할 수 있다.
- 사용자는 GUI 를 통해 미세먼지 수치를 조회하고자 하는 행정구역(시·군·구)을 설정할 수 있다.
- 사용자가 지역설정을 완료하면 선택한 지역의 관공서(시청, 군청, 구청, 행정복지센터)를 중심으로 T-map 지도가 생성되고 지역별 미세먼지 수치와 등급을 조회할 수 있다.
- 사용자는 ‘주변측정소 조회’ 버튼을 누르면 해당 지역의 대기 질 측정소의 위치가 T-Map 상에 마커가 생성된다.
- 사용자는 T-map 상에 생성된 마커를 선택하면 해당 마커 위치에 해당하는 대기 질 측정소의 주소와 주간 미세먼지 수치 예보를 조회할 수 있다.

3.2 비 기능적 요구사항

- 미세먼지 정보는 사용자가 조회하고자 하는 행정구역의 선택을 완료한 시점에 서버로 데이터를 요청한다.
- 지역별 미세먼지 수치와 등급을 조회하기 전 지역선택이 선행되어야 한다.
- ‘주변측정소 조회’ 버튼은 지역선택이 선행되어야 생성된다.
- T-map API 가 해당 지역 관공서를 검색할 수 없으면 가장 근접한 결과에 대한 위치로 지도 중심점을 맞춘다.
- 대기 질 측정소의 미세먼지 주간 예보는 사용자가 마커를 선택한 시점에 서버로 데이터를 요청한다.
- 언어는 한국어만 지원한다.

3.3 도메인 요구사항

- App 은 Android 기반 스마트폰에서 동작한다.
- App 내 지역설정은 대한민국만 사용 가능하다.
- App 은 기기가 인터넷에 연결되어 있어야만 동작한다.
- App 은 한국환경공단 Open API 와 Air Quality Forecast API 를 사용하여 미세먼지 정보를 제공받는다.
- 지도 API 는 T-map API 를 사용한다.

3.4 구조적 명세

‘먼지아나’ /미세먼지 정보제공 App	
기능	지역별 미세먼지 수치와 등급 조회
설명	사용자가 원하는 대한민국의 시, 군, 구 별 행정구역 단위의 최신 미세먼지 수치와 등급 정보를 제공한다.
입력	GUI 를 통한(시·군·구) 단위의 대한민국 행정구역 주소
출처	UI 로부터 발생한 이벤트
출력	해당지역의 T-map 지도, 미세먼지 수치($\mu\text{g}/\text{m}^3$), 등급
동작	사용자가 지역설정을 완료하면, 해당 지역의 관공서(시청, 군청, 구청, 행정복지센터)를 중심으로 T-map 지도가 화면에 출력된다. 선택한 행정구역 정보를 사용하여 한국환경공단 API 를 통해 해당지역의 미세먼지 수치 데이터를 요청하여 반환된 값과 등급을 선택한 지역과 함께 화면하단 배너에 텍스트로 출력한다. 이때 미세먼지 등급이 ‘좋음’ 이면 파란색, ‘보통’ 이면 녹색, ‘나쁨’ 이면 주황색, ‘매우 나쁨’ 이면 빨간색을 출력한다.
필요사항	시·군·구 단위의 행정구역 정보.
사전조건	사용자가 미세먼지를 정보를 조회할 지역을 GUI 로 선택해야 한다.

사후조건	사용자가 GUI 를 통해 입력한 지역정보를 사용하여 미세먼지 데이터를 요청하고 해당지역 T-map 지도를 생성하여 화면에 출력한다.
부작용	T-map 데이터에 해당지역 관공서에 대한 검색 결과가 존재하지 않을 때 검색결과 값과 가장 연관성이 높은 위치를 중심으로 맞춘다.

[표 3] ‘지역별 미세먼지 수치와 등급조회’ 요구사항에 대한 구조적 명세

‘먼지아나’ /미세먼지 정보제공 App	
기능	지역설정
설명	사용자가 미세먼지 정보를 조회하고자 할 지역을 GUI 를 통해 행정구역 단위로 선택한다.
입력	행정구역 버튼 리스트 선택
출처	UI 로부터 발생한 이벤트
출력	단계별 행정구역 명칭 리스트 (도, 광역시→ 시, 군, 구), 선택한 지역명에 대한 토스트 메시지
동작	사용자가 GUI 를 통해 단계별 행정구역을 선택한다. 최상위 행정구역 리스트에서 특정 지역을 선택하면 화면이 전환되어 선택한 행정구역의 하위 행정구역 리스트가 생성된다. 시, 군, 구 단위까지 같은 과정을 반복한다. (예: 경상남도 → 창원시 → 의창구)
필요사항	선택할 지역에 대한 상위 행정구역 명칭
사전조건	없음
사후조건	사용자가 지역을 선택할 때까지 대기한다.
부작용	없음

[표 3.1] ‘지역설정’ 요구사항에 대한 구조적 명세

‘먼지아나’ /미세먼지 정보제공 App	
기능	주변 측정소 조회
설명	사용자가 선택한 지역의 대기 질 측정소의 위치를 T-map 지도상에 마커로 표시한다.
입력	‘인접 측정소 조회’ 버튼 선택
출처	UI 로부터 발생한 이벤트
출력	화면의 T-map 지도상에서 측정소에 해당하는 부분에 마커 생성
동작	지도의 중심점(관공서) 좌표를 사용하여 한국 환경공단 API 를 통해 해당좌표에서 인접한 측정소 주소데이터를 요청하고 반환된 주소데이터를 T-map API 를 통해 주소 값을 좌표 값으로 변환하여 지도상에 해당 좌표에 해당하는 위치에 마커를 생성한다.

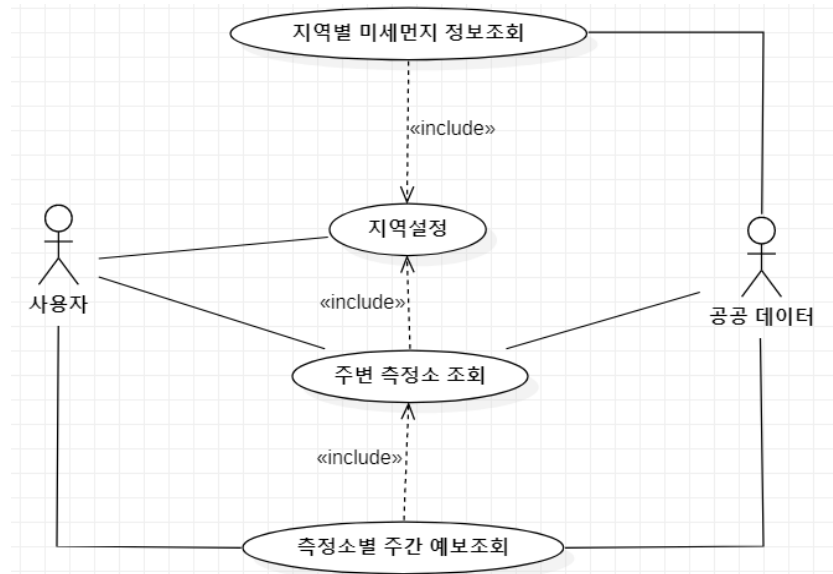
필요사항	지역 정보.
사전조건	사용자가 미세먼지를 정보를 조회할 지역을 GUI 를 통해 선택하고 T-map 지도가 생성되어 있어야 한다.
사후조건	생성된 지도의 중심점이 관공서의 좌표를 기준으로 인접한 대기 질 측정소의 위치를 지도상에 마커로 표시한다.
부작용	없음

[표 3.2] ‘주변 측정소 조회’ 요구사항에 대한 구조적 명세

‘먼지아나’ /미세먼지 정보제공 App	
기능	측정소별 주간 예보조회
설명	대기 질 측정소 조회 기능으로 지도상에 생성된 마커를 선택했을 때 해당 측정소에 대한 위치 정보와 미세먼지 예보데이터가 출력된다.
입력	지도상의 마커 선택
출처	UI 로부터 발생한 이벤트
출력	측정소 지역정보, 주간 미세먼지 수치예보, 측정소 좌표
동작	사용자가 마커를 선택하면 해당 마커에 대한 좌표를 토스트 메시지로 화면에 출력하고 측정소 좌표를 사용하여 Air Quality Forecast API 를 통해 좌표 값에 해당하는 측정소 미세먼지 주간예보 데이터를 요청하고 반환된 주간 예보데이터를 측정소 지역정보와 함께 텍스트로 출력한다.
필요사항	지역 정보.
사전조건	대기 질 측정소 위치조회
사후조건	측정소 정보, 예보데이터 출력
부작용	없음

[표 3.3] ‘측정소별 주간 예보조회’ 요구사항에 대한 구조적 명세

4. Use-Case Diagram



[그림 5] 미세먼지 정보제공 App ‘먼지아나’ Use-Case Diagram

- 사용자는 지역설정, 주변측정소 조회, 측정소별 주간 예보조회 기능에 대해 직접적으로 상호작용할 수 있다.
- 지역별 미세먼지 정보 조회는 지역설정 기능이 먼저 선행되고 지역설정이 완료되면 즉시 실행된다.
- 사용자가 지정한 지역의 미세먼지 데이터는 한국 환경공단 API 를 통해 데이터를 전달받는다.
- 측정소별 주간 예보조회 데이터는 Air Quality Forecast API 를 통해 전달받는다.

4.1 Use-Case description

실시간 미세먼지 정보 조회	
액터	공공데이터
설명	사용자는 원하는 지역에 대한 미세먼지 정보를 조회할 수 있다. 미세먼지에 대한 정보는 한국환경공단 API 로부터 전달받는다.
데이터	미세먼지 수치, 미세먼지 등급, 설정한 지역에 대한 정보
자극	지역설정
반응	화면에 선택한 지역의 관공서를 중심으로 한 T-map 지도 생성, 선택지역 미세먼지 정보, 등급 출력
주석	미세먼지 데이터는 한국 환경공단 서버에 저장되어 있는 값 중 가장 최근 값을 사용한다. 사용자로부터 지역설정이 선행되어야 한다.

[표 4.1] ‘지역별 미세먼지 정보조회’ Description

지역설정	
액터	사용자
설명	미세먼지 정보를 조회하고자 하는 지역을 선택한다.
데이터	선택한 행정구역에 대한 문자열 데이터
자극	조회하고자 하는 지역(시·군·구)에 해당하는 버튼을 선택한다.
반응	화면에 T-map 지도가 출력되고 지도의 중심점은 해당지역의 관공서이다.
주석	입력이 잘못되었을 때 ‘뒤로가기’ 버튼을 선택하면 이전화면으로 전환되고 다시 지역을 선택할 수 있다.

[표 4.2] ‘지역설정’ Description

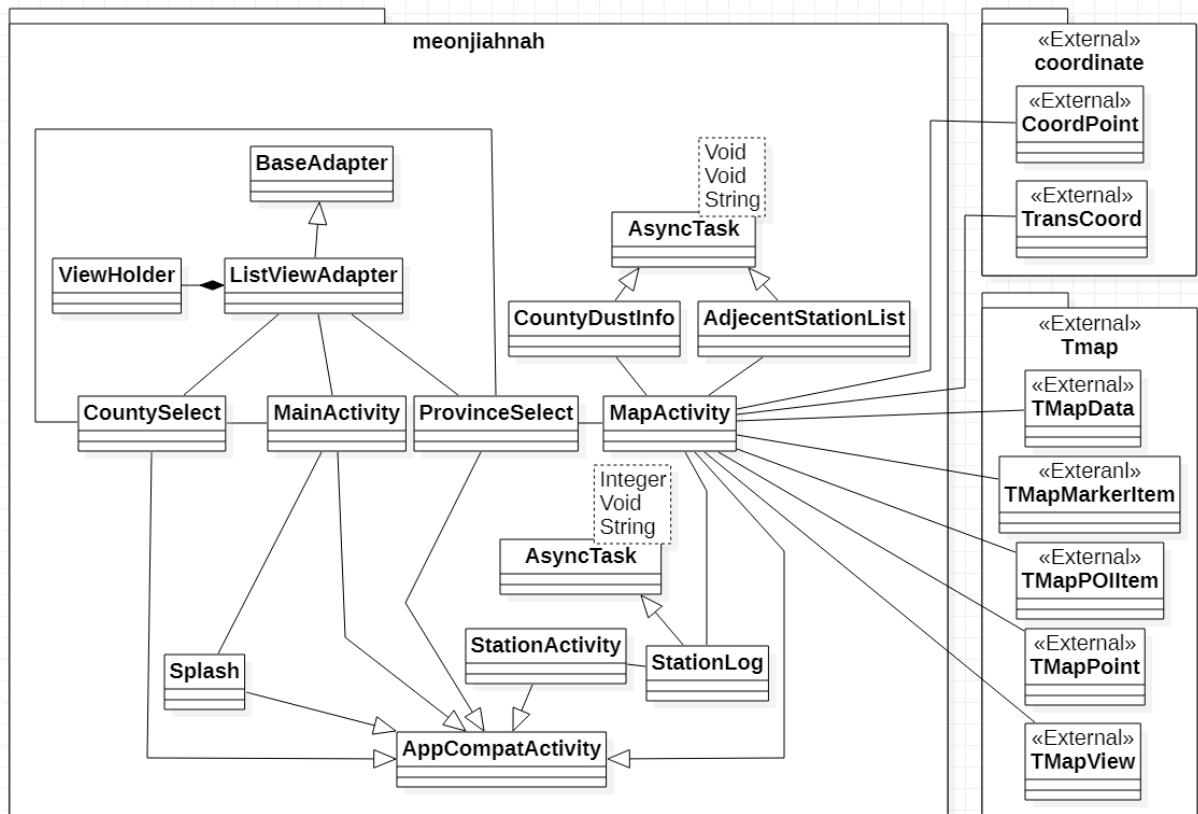
지역예보 조회	
액터	사용자, 공공 데이터
설명	사용자가 미세먼지 정보를 조회한 지역의 관공서에서 인접한 미세먼지 측정소를 조회하여 지도상에서 해당 위치에 마커 생성
데이터	인접 측정소 주소데이터, 좌표 값
자극	‘주변 측정소 조회’ 버튼 선택
반응	T-map 지도상에 인접 측정소의 좌표 값을 기준으로 마커 생성
주석	인접 측정소 주소 데이터는 한국 환경공단 API 를 사용하여 데이터를 가져온다.

[표 4.3] ‘주변 측정소 조회’ Description

지역예보 조회	
액터	사용자, 공공 데이터
설명	사용자가 선택한 지역의 대기 질 측정소의 주간 미세먼지 수치예보를 조회한다.
데이터	일주일 단위의 미세먼지 수치, 측정소 주소, 좌표 값
자극	지도상에 존재하는 마커를 선택한다.
반응	지도에서 마커에 해당하는 위치의 좌표 값을 토스트 메시지로 출력하고 해당 측정소의 주간 미세먼지 예보 데이터를 텍스트 형식으로 화면에 출력한다.
주석	반드시 ‘주변측정소 조회’ 버튼을 누른 다음 생성된 마커를 선택해야 측정소별 예보 조회 기능을 사용할 수 있으며 지역예보 조회 기능은 일주일 단위의 미세먼지 정보와 측정소의 위치에 대한 정보를 제공한다.

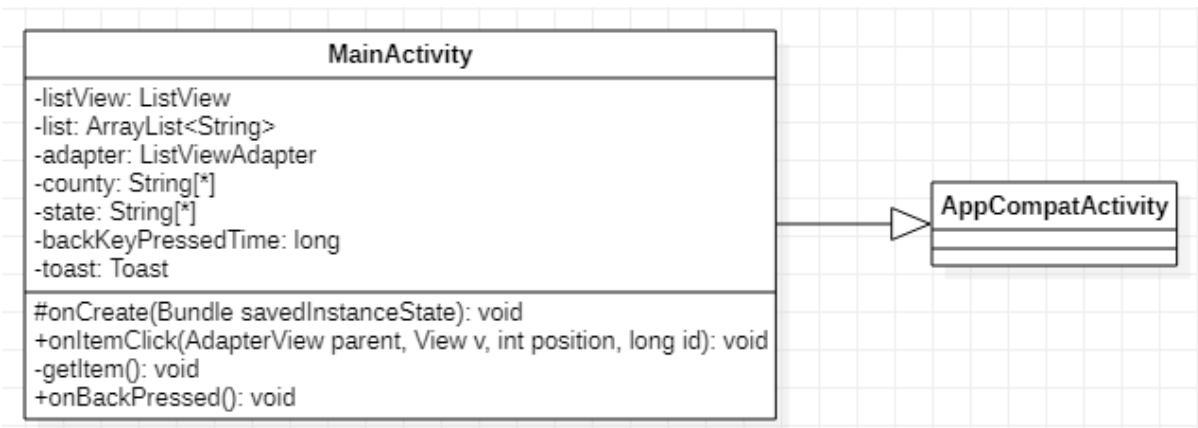
[표 4.4] ‘측정소별 주간 예보조회’ Description

5. Class Diagram



[그림 5] Class Diagram

5.1 MainActivity Class



[그림 5.1] MainActivity Class

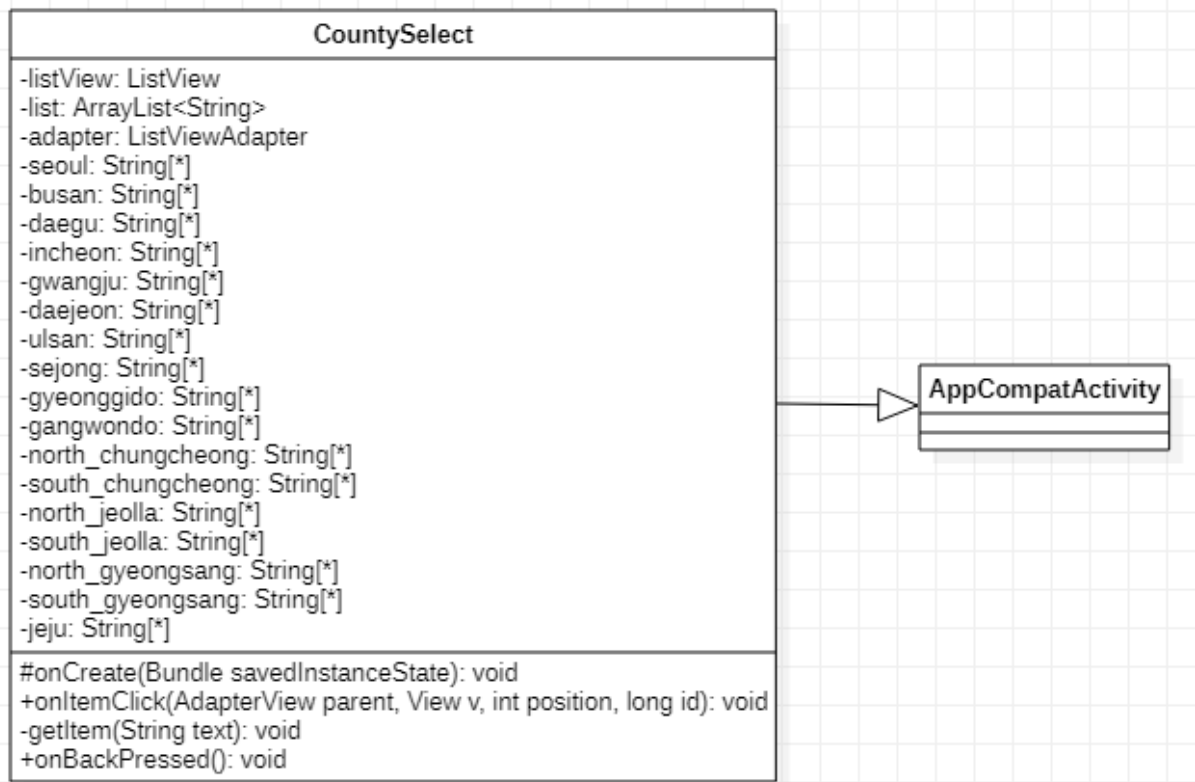
5.1.1 MainActivity Class description

MainActivity Class
상속 클래스

AppCompatActivity
멤버변수
private ListView listView 화면의 전환을 위한 뷰를 리스트 타입으로 저장하기 위한 변수.
private ArrayList<String> list 데이터를 배열의 리스트로 저장하기 위한 변수.
private ListViewAdapter adapter 리스트 뷰를 각 클래스마다 객체화 될 때 XML 파일과 연결하기 위한 변수.
private String[] county 특별시나 광역시 단위로 명칭을 저장하기 위한 변수.
private String[] state 도 단위로 명칭을 저장하기 위한 변수.
private long backKeyPressedTime 첫 번째 뒤로 가기 버튼을 누를 때 시간을 저장하기 위한 변수.
private Toast toast 토스트 메시지를 출력하기 위한 변수.
메서드
@Override protected void onCreate(Bundle savedInstanceState) 액티비티가 최초 생성될 때 호출되는 메서드. 초기화 작업이 진행되며 행정구역 별로 터치할 화면을 생성한다.
@Override public void onItemClick(AdapterView parent, View v, int position, long id) 지역의 명칭을 토스트 메시지로 출력하는 것과 동시에 행정구역의 이름을 터치하면 화면이 전환되며 해당하는 행정구역의 하위 단위의 행정구역의 리스트가 출력된다.
private void getItem() 화면에 출력될 행정구역 단위의 이름을 화면에 리스트 형식으로 추가하는 메서드.
@Override public void onBackPressed() onBackPressed 메서드는 '뒤로 가기' 버튼을 한번 누르면 화면이 이전으로 전환되는 이벤트 처리 메서드로 '뒤로 가기' 버튼을 한번 누른 상태에서 2.5 초 내에 '뒤로가기' 버튼을 한번 더 선택하면 앱이 종료되도록 구현되어 있다.

[표 5] MainActivity Class description

5.2 CountySelect Class



[그림 5.2] CountySelect Class description

5.2.1 CountySelect description

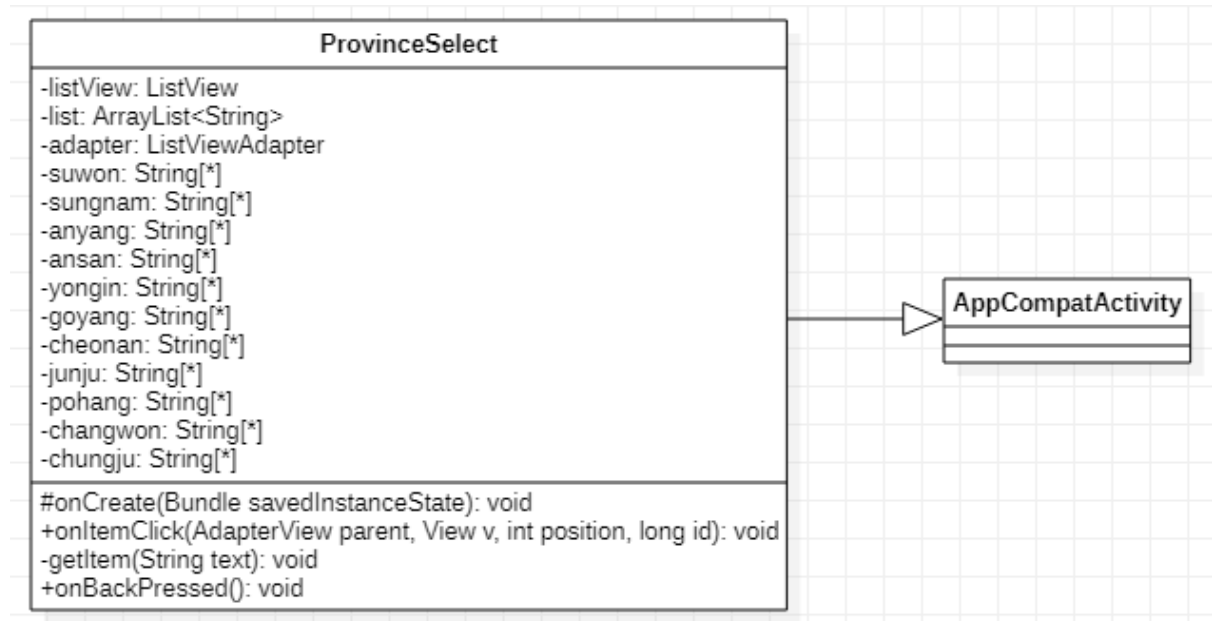
CountySelect
상속 클래스
AppCompatActivity
멤버 변수
private ListView listView 화면의 전환을 위한 뷰를 리스트 타입으로 저장하기 위한 변수.
private ArrayList<String> list String 타입의 데이터를 배열의 리스트로 저장하기 위한 변수.
private ListViewAdapter adapter 리스트 뷰를 각 클래스마다 객체화 될 때 XML 파일과 연결하기 위한 변수.
private String[] seoul 서울 특별시에 위치하는 모든 구의 명칭을 저장하기 변수.
private String[] busan 부산 광역시에 위치하는 모든 구의 명칭을 저장하기 위한 변수.

private String[] daegu 대구 광역시에 위치하는 모든 구의 명칭을 저장하기 위한 변수.
private String[] incheon 인천 광역시에 위치하는 모든 구의 명칭을 저장하기 위한 변수.
private String[] gwangju 광주 광역시에 위치하는 모든 구의 명칭을 저장하기 위한 변수.
private String[] daejeon 대전 광역시에 위치하는 모든 구의 명칭을 저장하기 위한 변수.
private String[] ulsan 울산 광역시에 위치하는 모든 구의 명칭을 저장하기 위한 변수.
private String[] sejong 세종 특별자치시에 위치하는 모든 면과 읍의 명칭을 저장하기 변수.
private String[] gyeonggido 경기도에 위치하는 모든 시의 명칭을 저장하기 위한 변수.
private String[] gangwondo 강원도에 위치하는 모든 시와 군의 명칭을 저장하기 위한 변수.
private String[] north_chungcheong 충청북도에 위치하는 모든 시와 군의 명칭을 저장하기 위한 변수.
private String[] south_chungcheong 충청남도에 위치하는 모든 시와 군의 명칭을 저장하기 위한 변수.
private String[] north_jeolla 전라북도에 위치하는 모든 시와 군의 명칭을 저장하기 위한 변수.
private String[] south_jeolla 전라남도에 위치하는 모든 시와 군의 명칭을 저장하기 위한 변수.
private String[] north_gyeongsang 경상북도에 위치하는 모든 시와 군의 명칭을 저장하기 위한 변수.
private String[] south_gyeongsang 경상남도에 위치하는 모든 시와 군의 명칭을 저장하기 위한 변수.
private String[] jeju 제주 특별시에 위치하는 모든 시의 명칭을 저장하기 위한 변수.
메서드
@Override protected void onCreate(Bundle savedInstanceState) 특별시나 광역시 또는 특별자치시나 도를 터치했을 때 해당하는 행정구역에 소속되어 있는 (시·구·군·면)을 선택할 수 있는 목록과 화면을 생성하는 메서드.
@Override public void onItemClick(AdapterView parent, View v, int position, long id) (시·구·군·면)을 터치했을 때 발생하는 이벤트에 대한 내용을 구현한 메서드.
private void getItem(String text) 앞에서 선택한 행정구역에 따라서 해당하는 행정구역에 소속되어 있는 (시·구·군·면) 목록을 화면에 생성하는 메서드.

```
@Override
public void onBackPressed()
‘뒤로 가기’ 버튼을 누르면 해당하는 화면이 종료되고 이전의 화면으로 전환되는
메서드.
```

[표 5.1] CountySelect description

5.3 ProvinceSelect Class



[그림 5.3] ProvinceSelect Class

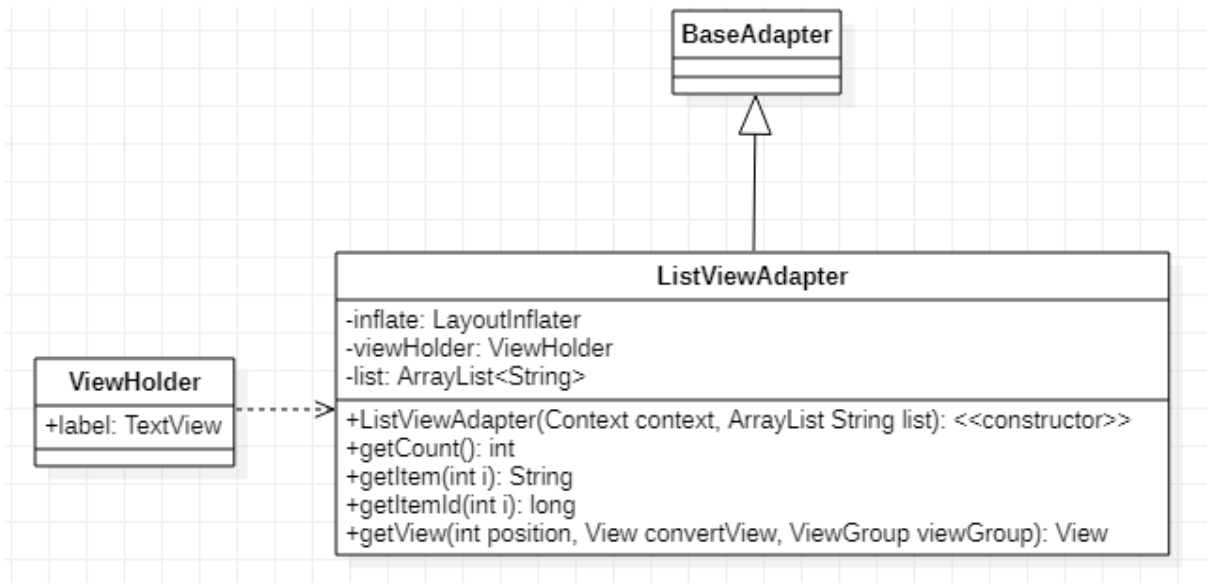
5.3.1 ProvinceSelect Class description

ProvinceSelect
상속 클래스
AppCompatActivity
멤버변수
private ListView listView 화면의 전환을 위한 뷰를 리스트 타입으로 저장하기위한 변수.
private ArrayList<String> list 데이터를 배열의 리스트로 저장하기위한 변수.
private ListViewAdapter adapter 리스트 뷰를 각 클래스마다 객체화 될 때 XML 파일과 연결하기 위한 변수.
private String[] suwon 수원에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.

private String[] sunnam 성남에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
private String[] anyang 안양에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
private String[] ansan 안산에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
private String[] yongin 용인에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
private String[] goyang 고양에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
private String[] cheonan 천안에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
private String[] junju 전주에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
private String[] pohang 포항에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
private String[] changwon 창원에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
private String[] chungju 충주에 소속 되어있는 전체 구의 명칭을 저장하기위한 변수.
메서드
@Override protected void onCreate(Bundle savedInstanceState) 앞서 특별시나 광역시 또는 특별자치시나 도에 해당하는 행정구역을 선택했을 때 해당하는 행정구역에 소속되어 있는 구를 선택할 수 있는 목록과 화면을 생성하는 메서드.
@Override public void onItemClick(AdapterView parent, View v, int position, long id) 앞서 선택한 지역의 구를 선택했을 때 발생하는 이벤트에 대한 내용을 구현한 메서드.
private void getItem(String text) 앞서 선택한 행정구역에 따라 해당하는 행정구역에 소속 되어있는 구의 목록을 생성하여 화면에 출력하는 메서드다.
@Override public void onBackPressed() ‘뒤로가기’ 버튼을 누르면 해당하는 화면이 종료되고 이전의 화면으로 전환되는 메서드.

[표 5.2] ProvinceSelect Class description

5.4 ListViewAdapter Class



[그림 5.4] ListViewAdapter Class

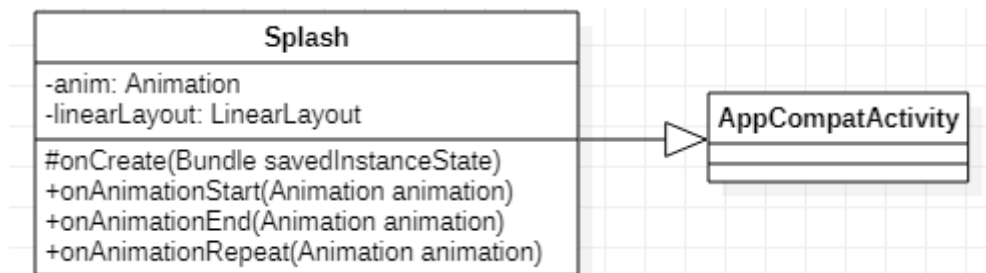
5.4.1 ListViewAdapter Class description

ListViewAdapter
상속 클래스
BaseAdapter
멤버변수
private LayoutInflater inflate XML 에 정의된 Resource 그려니까 데이터를 View 객체로 반환해주는 역할을 하는 변수.
private ViewHolder viewHolder ListView 를 생성할 때 마다 매번 findViewById() 함수를 호출하지 않고 ViewHolder 를 통해서 각 위젯을 필요할 때마다 재사용 하여 각 행을 구현하기 위한 변수.
private ArrayList<String> list 데이터를 ArrayList 로 저장하기위한 변수.
메서드
public ListViewAdapter(Context context, ArrayList<String> list) MainActivity 에서 데이터 리스트를 넘겨받아 해당하는 ListViewAdapter 객체에 저장하기위한 생성자.
@Override public int getCount() list 의 크기를 반환하기 위한 메서드.

<pre>@Override public String getItem(int i)</pre> <p>list 에 저장되어 있는 데이터를 해당하는 인덱스의 데이터를 반환하기 위한 메서드.</p>
<pre>@Override public long getItemId(int i)</pre> <p>BaseAdapter 클래스의 추상 메서드를 구현하기 위한 메서드.</p>
<pre>@Override public View getView(int position, View convertView, ViewGroup view- Group)</pre> <p>ListView 에 표현하기 위한 뷰를 생성하는 함수. Inflater 를 통해 행단위의 뷰의 내용을 파싱하여 뷰들을 객체화 한다. 그리고 각 위젯의 참조를 얻어와 현재 표현하고자 하는 데이터 값들을 지정해주기 위한 메서드.</p>

[표 5.3] ListViewAdapter Class description

5.5 Splash Class



[그림 5.5] Splash Class

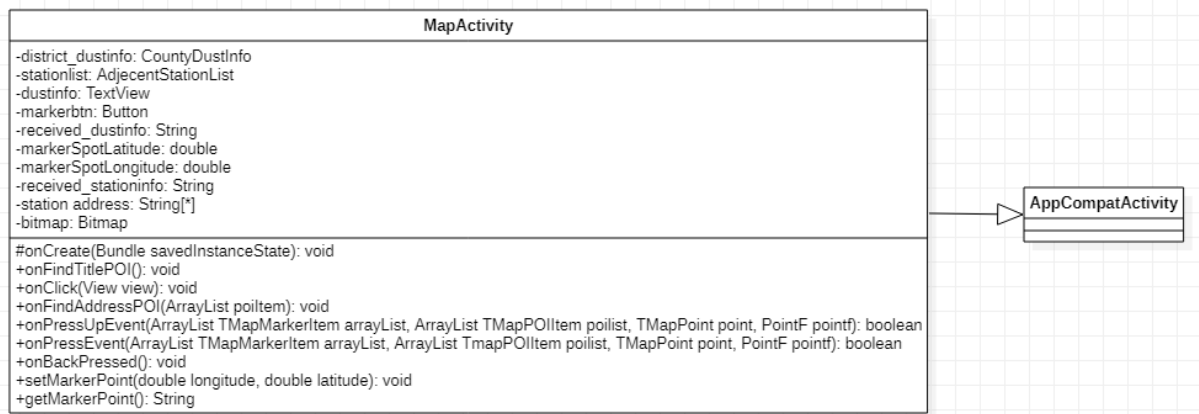
5.5.1 Splash Class description

Splash
상속 클래스
AppCompatActivity
멤버변수
Animation anim 앱을 시작할 때 발생하는 애니메이션에 대한 데이터를 저장하기 위한 변수.
LinearLayout linearLayout 애니메이션을 LinearLayout 형태로 출력하기 위한 변수.
메서드
<pre>@Override protected void onCreate(Bundle savedInstanceState)</pre> <p>앱을 실행할 때 발생하는 애니메이션에 대한 이벤트를 구현한 메서드.</p>

@Override public void onAnimationStart(Animation animation) 애니메이션 이벤트를 구현하기 위한 추상 메서드를 구현하는 메서드. 앱 동작 중에 호출되지 않는다.
@Override public void onAnimationEnd(Animation animation) 앱이 실행될 때 애니메이션을 실행하고 애니메이션이 종료될 때 발생하는 이벤트를 구현한 메서드.
@Override public void onAnimationRepeat(Animation animation) 애니메이션 이벤트를 구현하기 위한 추상 메서드를 구현하는 메서드. 앱 동작 중에 호출되지 않는다.

[표 5.4] Splash Class description

5.6 MapActivity Class



[그림 5.6] MapActivity Class

5.6.1 MapActivity Class description

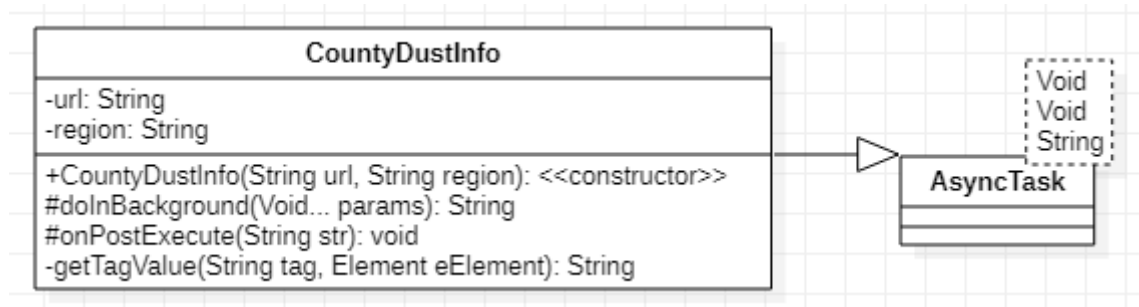
MapActivity
상속 클래스
AppCompatActivity
멤버변수
private CountyDustInfo district_dustinfo OpenAPI의 공공 데이터를 사용하기 위한 변수다.
private AdjacentStationList stationlist 측정소의 목록을 저장하기 위한 변수다.
private TextView dustinfo 해당하는 지역의 정보를 출력하기 위한 텍스트를 출력하기 위한 텍스트 뷰 변수.

private Button markerbtn ‘주변측정소 조회’ 버튼에 대한 데이터를 저장하기 위한 변수다.
private String received_dustinfo 해당하는 지역에 대한 정보를 요청하여 전송받은 데이터를 저장하기 위한 변수.
private double markerSpotLatitude T-map 을 터치했을 때 생성되는 마커의 위도 값을 저장하기위한 변수다.
private double markerSpotLongitude T-map 을 터치했을 때 생성되는 마커의 경도 값을 저장하기위한 변수다.
private String received_stationinfo 앞서 선택한 지역의 주변 측정소에 대한 정보를 공공데이터에 요청하였을 때 해당하는 측정소에 대한 데이터를 저장하기 변수.
private String station_address[] 해당 지역의 주변 측정소의 주소를 저장하기위한 변수.
Bitmap bitmap T-map 을 출력할 때 마커의 데이터를 저장할 변수.
<div>메서드</div>
@Override protected void onCreate(Bundle savedInstanceState) 앞서 선택한 지역에 따라 해당하는 위치를 T-map 지도를 생성하고 선택한 지역에 따라 공공 데이터에서 해당하는 지역에 대한 데이터를 스레드를 사용해 앱 실행 Flow 와 병렬적으로 동작하는데 API 를 통해 사용자가 선택한 지역에 대한 미세먼지 정보를 요청하여 전송받은 데이터를 텍스트 뷰에 출력하는 메서드.
@Override public void onFindTitlePOI(ArrayList poiItem) 사용자가 선택한 지역의 관공서 이름에 대한 문자열 데이터를 T-map API 를 사용해 검색하여 해당 위치에 대한 위도와 경도 값을 반환 받아 해당 좌표를 중심으로 지도를 생성하는 메서드다. 그리고 좌표 값을 WGS84 좌표계에서 TM 좌표계로 변환하여 공공 데이터 API 로 주변 측정소 정보를 요청하여 반환 받은 측정소의 주소 값을 측정소 별로 분할하여 String 타입의 배열 변수에 저장하는 메서드.
@Override public void onClick(View view) ‘주변측정소 조회’ 버튼을 선택했을 경우에 발생하는 이벤트 메서드.
@Override public void onFindAddressPOI(ArrayList poiItem) 앞서 저장한 주변 측정소 주소를 바탕으로 T - Map API 를 통해 측정소의 위치를 검색하는 메서드이며 반환된 좌표 값을 저장하고 지도상에 각 좌표 값에 해당하는 위치에 마커를 생성한다.
@Override public boolean onPressUpEvent(ArrayList<TMapMarkerItem>arrayList, ArrayList<TMapPOIItem> poilist, TMapPoint point, PointF pointf) 앞서 생성된 마커를 눌렀을 경우에 발생하는 이벤트를 구현한 메서드이며 마커를 누르게 되면 화면이 전환되어 앞서 선택한 지역의 정보와 주변에 위치하는 측정소의 시간대별 미세먼지 정보를 출력한다.

<pre>@Override public boolean onPressEvent(ArrayList<TMapMarkerItem>arrayList, ArrayList<TMapPOIItem> poilist, TMapPoint point, PointF pointf)</pre> <p>추상 클래스의 메서드를 구현하기 위해 별다른 의미 없이 구현한 메서드.</p>
<pre>@Override public void onBackPressed()</pre> <p>T-map 이 출력된 상태에서 뒤로가기 버튼을 눌렀을 때 발생하는 이벤트를 구현한 메서드.</p>
<pre>public void setMarkerPoint(double longitude, double latitude)</pre> <p>해당하는 화면을 출력하는 객체에 마커가 생성된 위치의 좌표 값을 저장하는 메서드.</p>
<pre>public String getMarkerPoint()</pre> <p>마커가 생성된 위치의 Double 타입의 좌표 값을 변수에 저장하여 반환하는 메서드.</p>

[표 5.5] MapActivity Class description

5.7 CountyDustInfo Class



[그림 5.7] CountyDustInfo Class

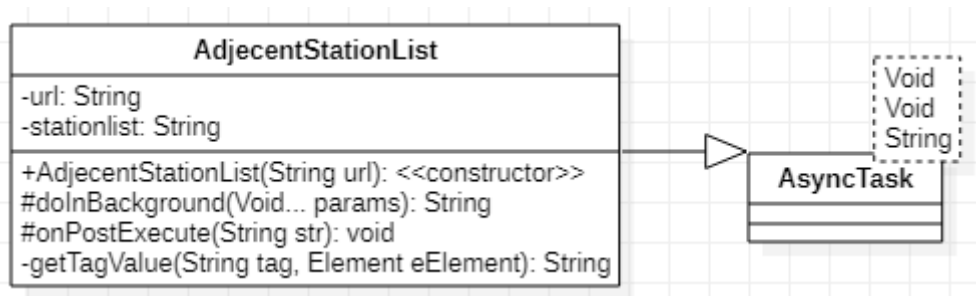
5.7.1 CountyDustInfo Class description

CountyDustInfo
상속 클래스
AsyncTask<Void, Void, String>
멤버변수
<pre>private String url</pre> <p>데이터의 url 을 저장하기 위한 변수.</p>
<pre>private String region</pre> <p>앞에서 선택한 지역의 명칭을 저장하기 위한 변수.</p>
메서드
<pre>public CountyDustInfo(String url,String region)</pre> <p>해당하는 위치의 데이터를 요청 하기위한 url 과 해당하는 위치의 명칭을 객체의 url 변수와 region 변수에 저장하기위한 생성자.</p>

<pre>@Override protected String doInBackground(Void... params)</pre> <p>API 키를 포함해서 앞서 선택한 지역에 해당하는 데이터를 parsing 할 url 을 지정하는 메서드이며 그리고 parsing 한 데이터를 로그에 출력하는 메서드.</p>
<pre>@Override protected void onPostExecute(String str)</pre> <p>쓰레드가 종료되면 결과를 반환하면서 그 반환 값을 통해 발생하는 이벤트를 구현한 메서드.</p>
<pre>private String getTagValue(String tag, Element eElement)</pre> <p>태그와 element 를 매개변수로 전달받아 노드를 찾아서 해당하는 노드의 값을 반환하는 메서드.</p>

[표 6.6] CountyDustInfo Class description

5.8 AdjacentStationList Class



[그림 5.8] AdjacentStationList Class

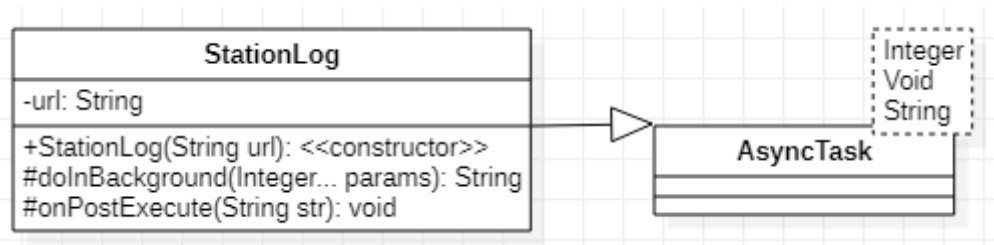
5.8.1 AdjacentStationList Class description

AdjacentStationList
상속 클래스
AsyncTask<Void, Void, String>
멤버변수
<pre>private String url</pre> <p>데이터의 url 을 저장하기 위한 변수.</p>
<pre>private String stationlist</pre> <p>측정소에 대한 명칭을 저장하기 위한 변수.</p>
메서드
<pre>public AdjacentStationList(String url)</pre> <p>해당하는 측정소에 대한 데이터를 요청하기위한 url 을 객체의 url 변수에 저장하기위한 생성자.</p>

<pre>@Override protected String doInBackground(Void... params)</pre> <p>API 키를 포함해서 앞서 선택한 지역에 해당하는 측정소에 대한 데이터를 parsing 할 url 을 지정하는 메서드다. 그리고 parsing 한 데이터를 로그에 출력하는 메서드.</p>
<pre>@Override protected void onPostExecute(String str)</pre> <p>쓰레드가 종료되면 결과를 반환하면서 그 반환 값을 통해 발생하는 이벤트를 구현한 메서드.</p>
<pre>private String getTagValue(String tag, Element eElement)</pre> <p>태그와 element 를 매개변수로 전달받아 매개변수들을 바탕으로 노드를 찾아서 해당하는 노드의 값을 반환하는 메서드.</p>

[표 5.7] AdjacentStationList Class description

5.9 StationLog Class



[그림 5.9] StationLog Class

5.9.1 StationLog Class description

StationLog
상속 클래스
AsyncTask<Integer, Void, String>
멤버변수
<pre>private String url</pre> <p>데이터의 url 을 저장하기 위한 변수.</p>
메서드
<pre>public StationLog(String url)</pre> <p>해당하는 측정소에 대한 데이터를 요청하기위한 url 을 객체의 url 변수에 저장하기위한 생성자.</p>
<pre>@Override protected String doInBackground(Void... params)</pre> <p>API 키를 포함해서 앞서 선택한 지역에 해당하는 측정소에 대한 데이터를 parsing 할 url 을 지정하는 메서드다. 그리고 parsing 한 데이터를 로그에 출력하는 메서드.</p>

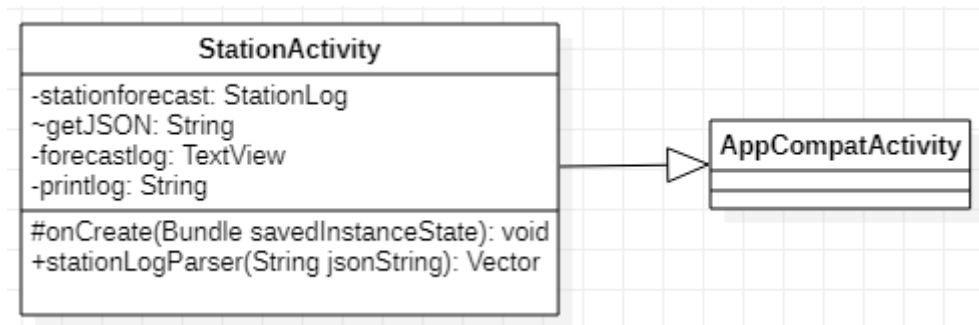
@Override

protected void onPostExecute(String str)

스레드가 종료되면 파라미터를 반환하면서 그 반환 값을 통해 발생하는 이벤트를 구현하는 메서드. 별다른 의미는 없는 메서드.

[표 5.8] StationLog Class description

5.10 StationActivity Class



[그림 5.10] StationActivity Class

5.10.1 StationActivity Class description

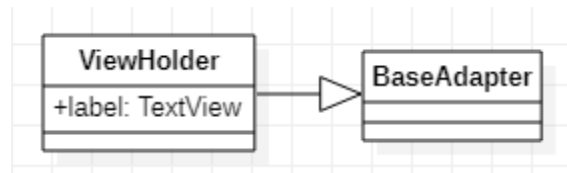
StationActivity
상속 클래스
AppCompatActivity
멤버변수
private StationLog stationforecastst 앞서 선택한 지역의 측정소에서 데이터를 파싱하기위한 변수.
private String getJSON 해당하는 지역의 측정소에서 가져온 데이터를 String 타입으로 저장하기위한 변수.
private TextView forecastlog 주간 예보를 화면에 출력하기위한 변수
private String printlog API 에서 가져온 데이터를 저장하기위한 변수.
메서드
@Override protected void onCreate(Bundle savedInstanceState) 앞에서 선택한 지역의 측정소에서 데이터를 요청하여 변수에 저장하는 메서드이며 그리고 가공된 JSON 데이터가 저장 되어있는 변수를 반환 받는 메서드를 호출하여 해당하는 측정소의 가공된 데이터를 텍스트 뷰에 출력하는 메서드.


```
public Vector stationLogParser(String jsonString)
```

전송받은 JSON 데이터가 저장된 변수의 데이터를 가공하여 화면에 출력하기 위하여 변수에 저장하여 반환하는 메서드.

[표 5.9] StationActivity Class description

5.11 ViewHolder Class



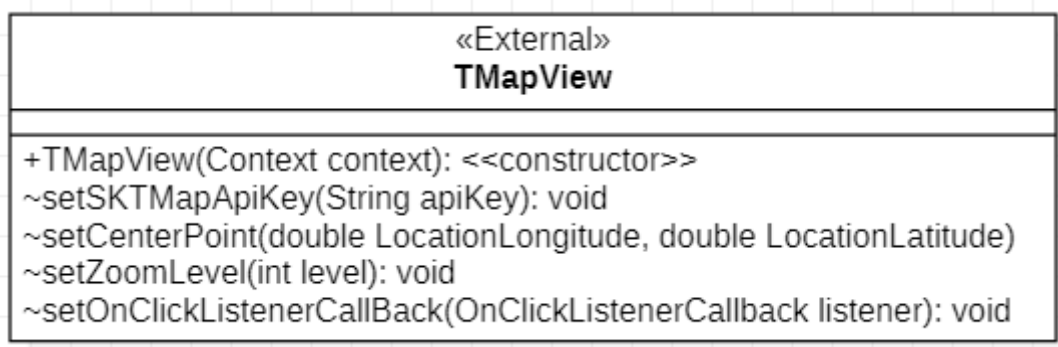
[그림 5.11] ViewHolder Class

5.11.1 ViewHolder Class description

ViewHolder
상속 클래스
AppCompatActivity
소속 클래스
ListViewAdapter
멤버변수
<pre>public TextView label</pre> 출력된 뷰의 데이터를 저장하기 위한 변수.

[표 5.10] ViewHolder Class description

5.12 TMapView Class



[그림 5.12] TMapView Class

5.12.1 TMapView Class description

<<External>> TMapView
소속 패키지
TMap
메서드
public TMapView(Context context) 매개변수가 Centex 로 구성된 TMapView 타입의 객체를 생성하기 위한 생성자 보통 TMapView(this)로 객체를 생성한다.
void setSKTMapApiKey(String apiKey) 라이브러리 사용을 위해 할당 받은 등록된 키를 설정한다.
void setCenterPoint(double LocationLongitude, double LocationLatitude) 매개변수로 전달받은 좌표값을 바탕으로 T - Map 의 중심좌표를 이동하는 메서드.
void setZoomLevel(int level) T - Map 의 축척 레벨을 설정하는 메서드.
void setOnClickListenerCallBack(OnClickListenerCallBack listener) 매개변수로 전달받은 인터페이스 내부에 인터페이스 메서드를 재정의 하여 클릭 이벤트가 발생할 때 인터페이스 메서드를 별도로 추가하여 구현하는 메서드.

[표 5.11] TMapView Class description

5.13 TMapData Class

«External» TMapData
+TmapData(): <<constructor>> ~findTitlePOI(final String data, final FindTitlePOIListenerCallback findTitlePOIListener): void ~findAddressPOI(final String data, final FindAddressPOIListenerCallback findAddressPOIListener): void

[그림 5.13] TMapData Class

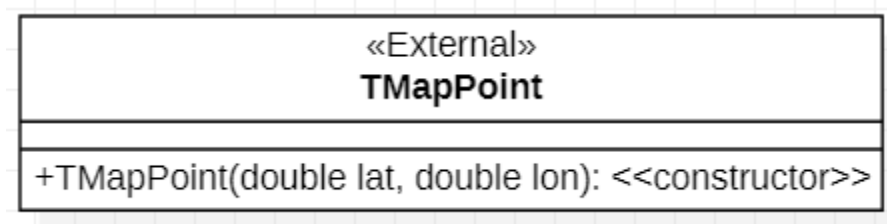
5.13.1 TMapView Class description

<<External>> TMapData
소속 패키지
TMap

메서드
public TMapData() POI 검색, 경로검색 등의 지도 데이터를 관리하는 TMapData 타입의 객체를 생성하는 생성자.
void findTitlePOI (final String data, final FindTitlePOIListenerCallback findTitlePOIListener) 메서드 내부에 인터페이스 함수를 따로 재정의하여 앞서 선택한 지역의 명칭을 바탕으로 T - Map 에 표시된 상점이나 건물 또는 관심지점의 데이터를 공공데이터에 요청하여 받아오는 메서드다.
void findAddressPOI(final String data, final FindAddressPOIListenerCallback findAddressPOIListener) 메서드 내부에서 인터페이스 함수를 따로 재정의 하여 앞서 저장한 측정소의 주소를 바탕으로 측정소의 좌표값을 검색하는 메서드.

[표 5.12] TMapView Class description

5.14 TMapPoint Class



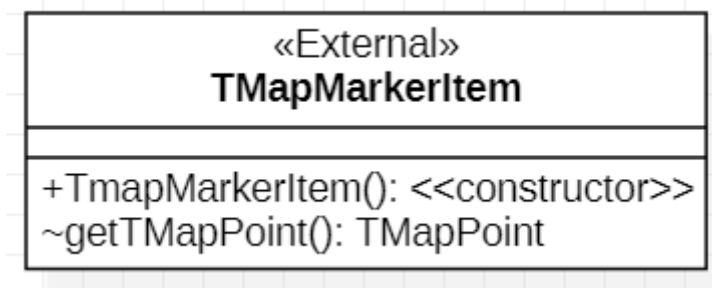
[그림 5.14] TMapPoint Class

5.14.1 TMapPoint Class description

<<External>> TMapPoint
소속 패키지
TMap
메서드
public TMapPoint(double lat, double lon) 위도, 경도 좌표를 나타내기 위해 TMapPoint 타입의 객체를 생성하는 생성자.

[표 5.13] TMapPoint Class description

5.15 TMapMarkerItem Class



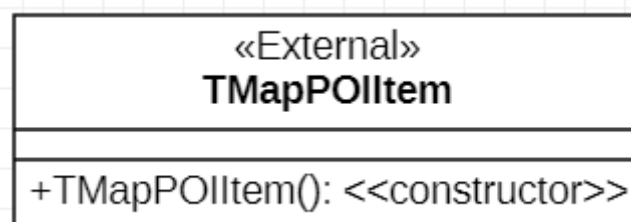
[그림 5.15] TMapMarkerItem Class

5.15.1 TMapMarkerItem Class description

«External» TMapMarkerItem
소속 패키지
TMap
메서드
public TMapMarkerItem() T - Map 에 마커를 표시하기 위한 TMapMarkerItem 타입의 객체를 생성하는 생성자.
TMapPoint getTMapPoint() T - Map 에 생성된 마커의 위도, 경도 좌표를 반환하는 메서드.

[표 5.14] TMapMarkerItem Class description

5.16 TMapPOIItem Class



[그림 6.16] TMapPOIItem Class

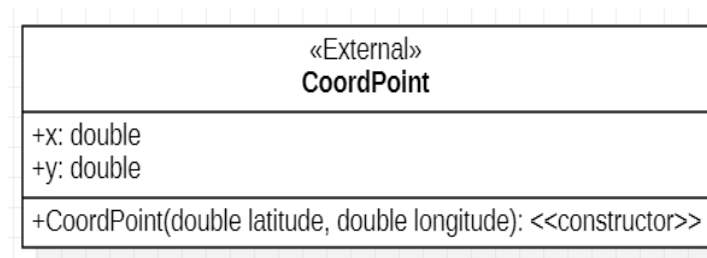
5.16.1 TMapPOIItem Class description

«External» TMapPOIItem

소속 패키지
TMap
메서드
public TMapPOIItem() T - Map 에 표시된 상점이나 건물 또는 관심지점에 대한 데이터를 관리하기 위한 TMapPOIItem 타입의 객체를 생성하는 생성자.

[표 5.15] TMapPOIItem Class description

5.17 CoordPoint Class



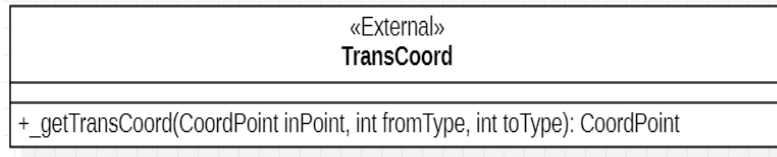
[그림 5.17] CoordPoint Class

5.17.1 CoordPoint Class description

<<External>> CoordPoint
소속 패키지
coordinate
멤버 변수
public double x 변환한 좌표값을 x 좌표의 형태로 저장하는 변수.
public double y 변환한 좌표값을 y 좌표의 형태로 저장하는 변수.
메서드
public CoordPoint(double latitude, double longitude) 해당하는 좌표값을 다른 타입의 좌표로 변환하기 위한 CoordPoint 타입의 객체를 생성하는 생성자.

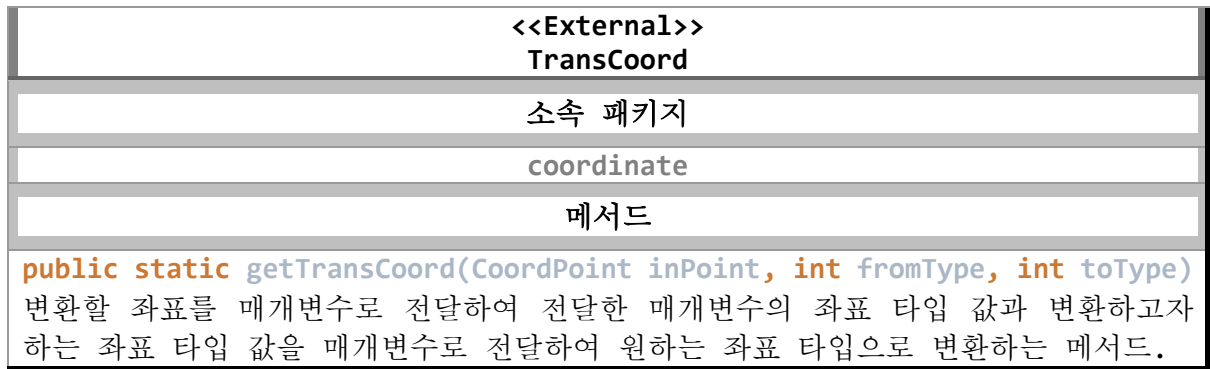
[표 5.16] CoordPoint Class description

5.18 TransCoord Class



[그림 6.18] TransCoord Class

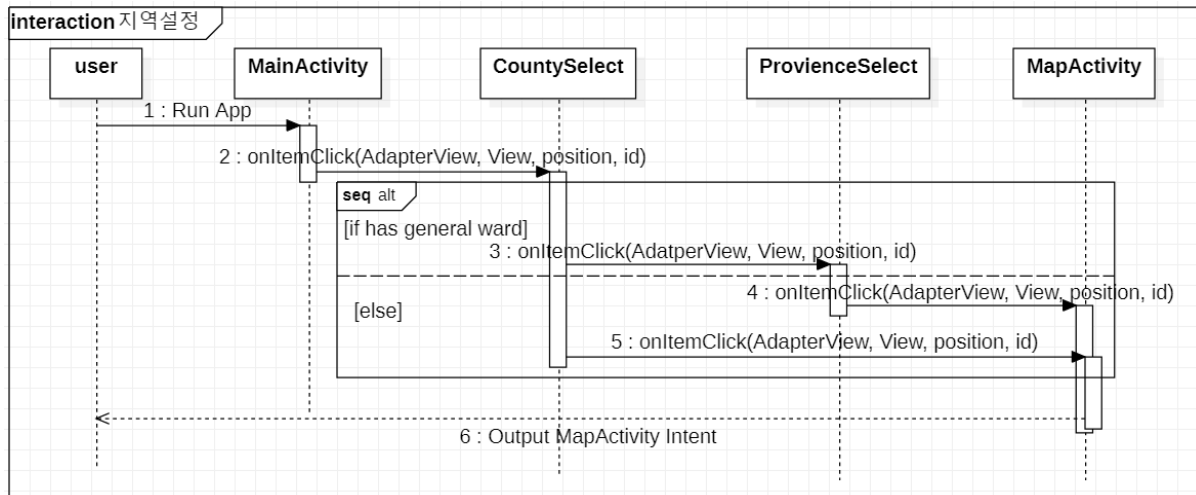
5.18.1 TransCoord Class description



[표 5.17] TransCoord Class description

6. Sequence Diagram

6.1. 지역설정



[그림 6.1] 지역설정 Sequence Diagram

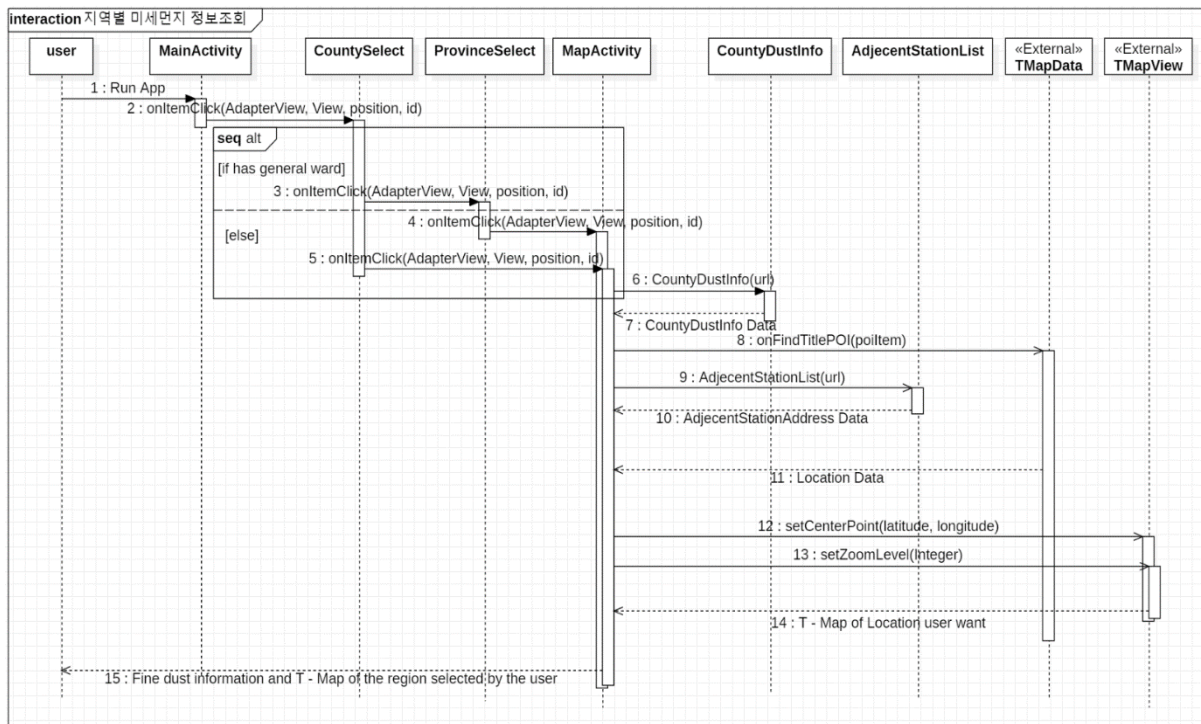
6.1.1 지역설정 description

1. 사용자가 앱을 실행한다.
2. 앱을 실행하게 되면 제일 먼저 MainActivity 클래스의 화면이 출력되는데 생성되어있는 리스트 목록 중 하나(광역시 또는 도 단위)를 터치하면 터치한 목록의

정보가 onItemClick() 메서드의 매개변수로 전달되어 다음 화면이 생성되어 출력된다.

3. 앞서 선택한 지역에 따라 생성되는 화면이 다르며 만약 사용자가 여기서 일반구가 있는 지역(시 단위)을 선택했다면 한번더 onItemClick() 메서드의 매개변수로 터치한 목록의 정보가 매개변수로 전달되어 구 단위의 지역을 선택하기 위해 ProvinceSelect 클래스의 화면이 생성되어 출력된다.
4. 앞서 선택한 지역에 따라 출력되는 지역이 다르며 GUI 로 구현한 최소 행정단위인 구를 선택한다면 선택한 목록의 정보가 매개변수로 전달되어 MapActivity 클래스의 화면이 생성되어 출력된다.
5. 만약에 사용자가 CountySelect 클래스의 화면에서 일반 구가 없는 시, 예를 들어 서울 특별시를 선택했다고 가정하면 선택한 목록의 정보가 매개변수로 전달되어 ProvinceSelect 클래스의 화면을 거치지 않고 바로 MapActivity 클래스의 화면이 생성되어 출력된다.
6. 사용자는 MapActivty 클래스의 Intent 그러니까 MapActivity 클래스의 생성된 화면을 볼 수 있다.

6.2. 지역별 미세먼지 정보조회



[그림 6.2] 지역별 미세먼지 정보조회 Sequence Diagram

6.2.1 지역별 미세먼지 정보조회 description

1. 사용자가 앱을 실행한다.

2. 앱을 실행하게 되면 제일 먼저 MainActivity 클래스의 화면이 출력되는데 생성되어있는 리스트 목록 중 하나(광역시 또는 도 단위)를 터치하면 터치한 목록의 정보가 onItemClick() 메서드의 매개변수로 전달되어 다음 화면이 생성되어 출력된다.
3. 앞서 선택한 지역에 따라 생성되는 화면이 다르며 만약 사용자가 여기서 일반구가 있는 지역(시 단위)을 선택했다면 한번더 onItemClick() 메서드의 매개변수로 터치한 목록의 정보가 매개변수로 전달되어 구 단위의 지역을 선택하기 위해 ProvienceSelect 클래스의 화면이 생성되어 출력된다.
4. 앞서 선택한 지역에 따라 출력되는 지역이 다르며 GUI 로 구현한 최소 행정단위인 구를 선택한다면 선택한 목록의 정보가 매개변수로 전달되어 MapActivity 클래스의 화면이 생성되어 출력된다.
5. 만약에 사용자가 CountySelect 클래스의 화면에서 일반 구가 없는 시, 예를 들어 서울 특별시를 선택했다고 가정하면 선택한 목록의 정보가 매개변수로 전달되어 ProvienceSelect 클래스의 화면을 거치지 않고 바로 MapActivity 클래스의 화면이 생성되어 출력된다.
6. MapActivity 클래스에서 CountyDustInfo 클래스의 생성자를 매개변수로 앞서 선택한 지역의 정보를 포함한 url 을 매개변수로 전달받아 생성된 CountyDustinfo 객체로 미세먼지 정보를 요청한다.
7. CountyDustInfo 클래스는 쓰레드를 상속받은 클래스 이므로 쓰레드를 이용하여 미세먼지 정보를 공공데이터에서 전달받는다.
8. 앞서 선택한 지역의 명칭을 바탕으로 좌표값을 검색하는 TMapData 클래스의 메서드를 호출한다.
9. AdjacentStationList 클래스는 쓰레드를 상속받은 클래스다. 객체를 생성하여 쓰레드를 이용하여 앞서 선택한 지역의 좌표값을 바탕으로 공공데이터 주변의 측정소에 대한 정보를 요청한다.
10. 요청한 주변의 측정소에 대한 정보를 쓰레드를 통해 전달받는다.
11. 8 번에서 호출했던 TMapView 클래스의 onFindTitlePOI() 메서드에서 반환된 앞서 선택한 지역의 정보를 전달받는다.
12. 전달받은 지역의 좌표값을 TMapView 클래스의 SetCenterPoint() 메서드를 이용하여 T - Map 의 초기 위치를 정한다.
13. TMapView 클래스의 setZoomLevel() 메서드를 이용하여 출력할 T - Map 의 줌 정도를 설정한다.
14. 사용자가 앞서 선택한 지역이 T - Map 에 출력된다.
15. 사용자는 앞서 선택한 지역에 대한 미세먼지 정보와 선택한 지역을 T - Map 에서 볼 수 있다.

[illegible]

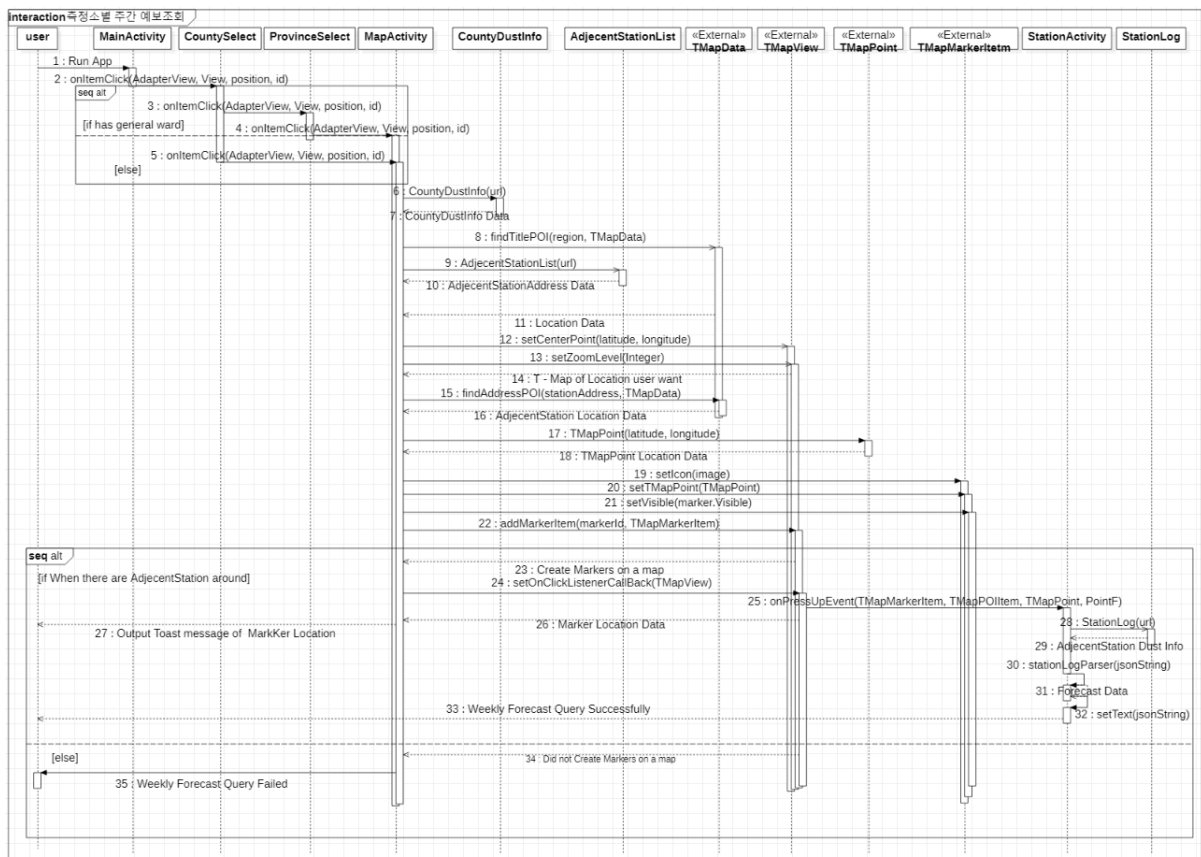
6.3.1 주변 측정소 조회 description

- 33

6. MapActivity 클래스에서 CountyDustInfo 클래스의 생성자를 매개변수로 앞서 선택한 지역의 정보를 포함한 url 을 매개변수로 전달받아 생성된 CountyDustinfo 객체로 미세먼지 정보를 요청한다.
7. CountyDustInfo 클래스는 쓰레드를 상속받은 클래스 이므로 쓰레드를 이용하여 미세먼지 정보를 공공데이터에서 전달받는다.
8. 앞서 선택한 지역의 명칭을 바탕으로 좌표 값을 검색하는 TMapData 클래스의 메서드를 호출한다.
9. AdjacentStationList 클래스는 쓰레드를 상속받은 클래스다. 객체를 생성하여 쓰레드를 이용하여 앞서 선택한 지역의 좌표 값을 바탕으로 공공데이터 주변의 측정소에 대한 정보를 요청한다.
10. 요청한 주변의 측정소에 대한 정보를 쓰레드를 통해 전달받는다.
11. 8 번에서 호출했던 TMapView 클래스의 onFindTitlePOI() 메서드에서 반환된 앞서 선택한 지역의 정보를 전달받는다.
12. 전달받은 지역의 좌표 값을 TMapView 클래스의 SetCenterPoint() 메서드를 이용하여 T - Map 의 초기 위치를 정한다.
13. TMapView 클래스의 setZoomLevel() 메서드를 이용하여 출력할 T - Map 의 줌 정도를 설정한다.
14. 사용자가 앞서 선택한 지역이 T - Map 에 출력된다.
15. 앞서 전달받은 주변에 위치한 측정소들의 주소를 바탕으로 TMapData 클래스의 findAddressPOI() 메서드를 이용하여 측정소의 좌표를 검색한다.
16. 검색한 측정소의 좌표 값을 전달받는다.
17. 측정소의 위치에 마커를 생성하기위해 TMapPoint 객체를 생성한다.
18. 생성한 TMapPoint 객체의 좌표 값을 전달받는다.
19. TMapMarkerItem 클래스의 setIcon() 메서드를 호출하여 생성할 마커의 이미지를 정한다.
20. TMapMarkerItem 클래스의 setTMapPoint() 메서드를 호출하여 앞서 생성한 좌표값이 저장되어있는 객체를 매개변수로 전달해 마커를 생성하고자 하는 위치를 지정한다.
21. TMapMarkerItem 클래스의 setVisible() 메서드를 호출하여 마커가 화면에 보이도록 설정한다.
22. 화면에다가 마커를 추가하기 위해 TMapView 클래스의 addMarKetItem() 메소드를 호출한다.
23. 마커가 정상적으로 T - Map 에 생성되었다.
24. 사용자는 측정소의 위치를 조회할 수 있다.

25. 사용자가 앞서 선택한 위치의 주변에 측정소가 없는 경우가 있다. 이런 경우에는 마커가 T - Map에 생성되지 않는다.
26. 사용자는 측정소의 위치를 조회하지 못하였다.

6.4. 측정소별 주간 예보



[그림 6.4] 측정소별 주간 예보 Sequence Diagram

6.4.1 측정소별 주간 예보 description

1. 사용자가 앱을 실행한다.
2. 앱을 실행하게 되면 제일 먼저 MainActivity 클래스의 화면이 출력되는데 생성되어있는 리스트 목록 중 하나(광역시 또는 도 단위)를 터치하면 터치한 목록의 정보가 onItemClick() 메서드의 매개변수로 전달되어 다음 화면이 생성되어 출력된다.
3. 앞서 선택한 지역에 따라 생성되는 화면이 다르며 만약 사용자가 여기서 일반구가 있는 지역(시 단위)을 선택했다면 한번더 onItemClick() 메서드의 매개변수로 터치한 목록의 정보가 매개변수로 전달되어 구 단위의 지역을 선택하기 위해 ProvinceSelect 클래스의 화면이 생성되어 출력된다.

4. 앞서 선택한 지역에 따라 출력되는 지역이 다르며 GUI 로 구현한 최소 행정단위인 구를 선택한다면 선택한 목록의 정보가 매개변수로 전달되어 MapActivity 클래스의 화면이 생성되어 출력된다.
5. 만약에 사용자가 CountySelect 클래스의 화면에서 일반 구가 없는 시, 예를 들어 서울 특별시를 선택했다고 가정하면 선택한 목록의 정보가 매개변수로 전달되어 ProvinceSelect 클래스의 화면을 거치지 않고 바로 MapActivity 클래스의 화면이 생성되어 출력된다.
6. MapActivity 클래스에서 CountyDustInfo 클래스의 생성자를 매개변수로 앞서 선택한 지역의 정보를 포함한 url 을 매개변수로 전달받아 생성된 CountyDustInfo 객체로 미세먼지 정보를 요청한다.
7. CountyDustInfo 클래스는 쓰레드를 상속받은 클래스 이므로 쓰레드를 이용하여 미세먼지 정보를 공공데이터에서 전달받는다.
8. 앞서 선택한 지역의 명칭을 바탕으로 좌표 값을 검색하는 TMapData 클래스의 메서드를 호출한다.
9. AdjacentStationList 클래스는 쓰레드를 상속받은 클래스다. 객체를 생성하여 쓰레드를 이용하여 앞서 선택한 지역의 좌표 값을 바탕으로 공공데이터 주변의 측정소에 대한 정보를 요청한다.
10. 요청한 주변의 측정소에 대한 정보를 쓰레드를 통해 전달받는다.
11. 8 번에서 호출했던 TMapView 클래스의 onFindTitlePOI() 메서드에서 반환된 앞서 선택한 지역의 정보를 전달받는다.
12. 전달받은 지역의 좌표 값을 TMapView 클래스의 SetCenterPoint() 메서드를 이용하여 T - Map 의 초기 위치를 정한다.
13. TMapView 클래스의 setZoomLevel() 메서드를 이용하여 출력할 T - Map 의 줌 정도를 설정한다.
14. 사용자가 앞서 선택한 지역이 T - Map 에 출력된다.
15. 앞서 전달받은 주변에 위치한 측정소들의 주소를 바탕으로 TMapData 클래스의 findAddressPOI() 메서드를 이용하여 측정소의 좌표를 검색한다.
16. 검색한 측정소의 좌표 값을 전달받는다.
17. 측정소의 위치에 마커를 생성하기위해 TMapPoint 객체를 생성한다.
18. 생성한 TMapPoint 객체의 좌표 값을 전달받는다.
19. TMapMarkerItem 클래스의 setIcon() 메서드를 호출하여 생성할 마커의 이미지를 정한다.
20. TMapMarkerItem 클래스의 setTMapPoint() 메서드를 호출하여 앞서 생성한 좌표값이 저장되어있는 객체를 매개변수로 전달해 마커를 생성하고자 하는 위치를 지정한다.

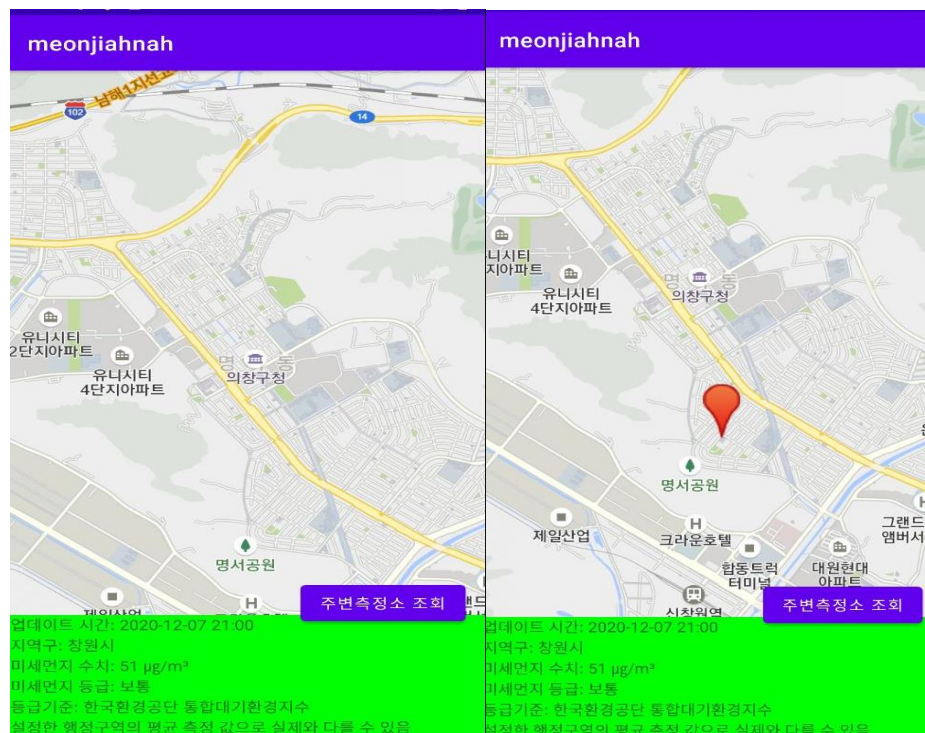
21. TMapMarkerItem 클래스의 setVisible() 메서드를 호출하여 마커가 화면에 보이도록 설정한다.
22. 화면에다가 마커를 추가하기 위해 TMapView 클래스의 addMarketItem() 메소드를 호출한다.
23. 마커가 정상적으로 T - Map 에 생성되었다.
24. TMapView 클래스의 setOnClickListenerCallBack() 메서드를 호출하여 생성된 마커가 한번더 눌러졌을 때 발생하는 이벤트를 처리한다.
25. TMapView 클래스의 OnPressUpEvent() 메서드를 호출하여 눌러졌던 마커가 눌러지지 않은 상태가 되었을 때 발생하는 이벤트를 처리한다. 그리고 StationActivity 클래스의 화면이 생성되어 출력된다.
26. 생성된 마커의 위치 정보를 가져온다.
27. 토스트 메시지를 이용하여 사용자에게 측정소의 좌표 값을 화면에 출력한다.
28. StationLog 클래스는 스레드 클래스를 상속받은 클래스이므로 객체를 생성하여 쓰레드를 이용하여 해당하는 측정소의 위치 정보를 포함한 url 로 공공데이터를 요청한다.
29. 앞서 요청했던 측정소의 정보를 가져온다.
30. StationActivity 의 stationLogParser() 메소드를 이용하여 앞서 저장한 측정소의 정보를 StationActivity 클래스의 화면에 출력하기 위하여 가공한다.
31. 앞서 가공한 데이터를 가져온다.
32. 가공한 데이터를 StationActivity 클래스의 화면에 출력하기 위해 setText() 메서드를 호출한다.
33. 사용자는 측정소별 주간 예보조회를 할 수 있다.
34. 사용자가 앞서 선택한 위치의 주변에 측정소가 없는 경우가 있다. 이런 경우에는 마커가 T - Map 에 생성되지 않는다.
35. 사용자는 측정소별 주간 예보조회를 하지 못하였다.

7. App UI

meonjiahnah	meonjiahnah	meonjiahnah
서울특별시	창원시	의창구
부산광역시	진주시	성산구
대구광역시	통영시	마산합포구
인천광역시	사천시	마산회원구
광주광역시	김해시	진해구
대전광역시	밀양시	
울산광역시	거제시	
세종특별자치시	양산시	

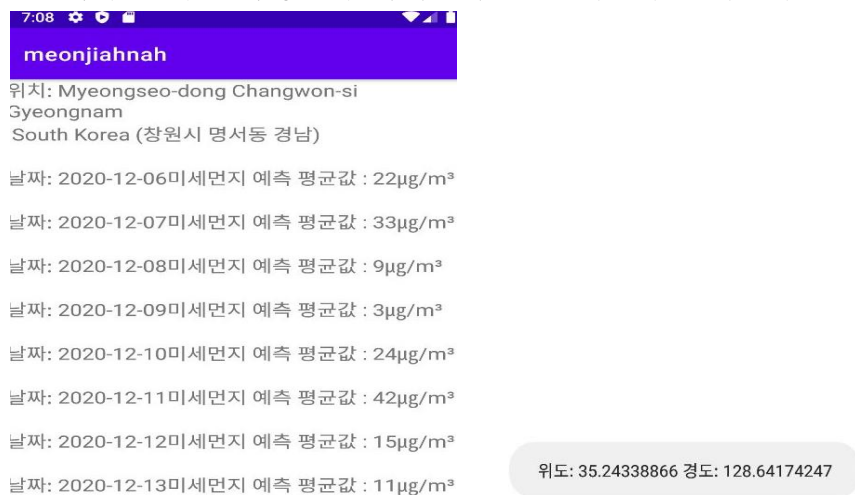
[그림 7] 지역선택 UI

최초 실행 시 위와 같이 광역시, 도, 자치 시 단위로 리스트가 생성되고 리스트 중 하나를 선택하면 화면이 전환되면서 선택한 지역의 하위 행정구역 리스트로 변경된다. 동일한 방법으로 시, 군, 구 단위 행정구역까지 선택을 완료하면 해당 지역에 대한 실시간 미세먼지 정보와 등급을 출력한다.



[그림 7.1] 지역별 미세먼지 수치, 등급, 인접 측정소 마커

위와 같이 선택한 지역의 관공서를 중심으로 T-map 지도가 생성되면서 하단 배너에 해당지역의 미세먼지 수치와 등급, 업데이트 시간이 텍스트 형식으로 출력되어 있는 모습을 확인할 수 있다. 배너의 경우 미세먼지 등급에 따라 색깔이 바뀌게 된다. 이 상태에서 주변 측정소 버튼을 누르게 되면 인접한 대기 질 측정소 위치에 마커가 생성된다. 이때 마커를 선택하면 해당 측정소의 미세먼지 주간 예보가 출력된다.



[그림 7.2] 측정소별 미세먼지 주간예보, 측정소 위치 토스트 메시지

위와 같이 마커를 선택하면 토스트 메시지로 해당 측정소의 좌표를 출력하고 측정소 위치정보, 미세먼지 예보를 출력하는 모습을 확인할 수 있다.

8. Test

유스케이스에서 정의한 기능이 잘 적용되었는지 검증하기 위한 기능 테스트와, 다양한 환경에서 실행여부를 검증하기 위해 실행테스트를 수행한다.

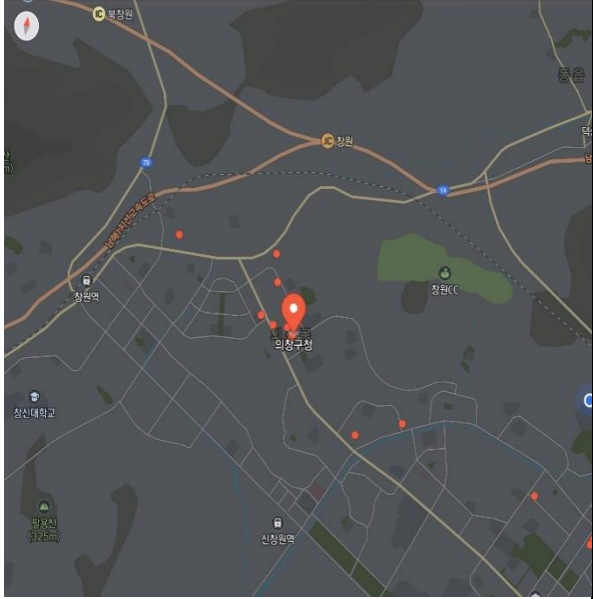

8.1 기능테스트

8.1.1 지역별 미세먼지 정보조회

테스트 시나리오	테스트 횟수	예상결과
지도의 중심점이 선택한 지역의 관공서의 위치가 서비스 중인 T-Map 과 일치하는지 확인	20	일치
선택한 지역의 미세먼지 수치가 한국 환경공단에서 제공하는 서비스의 데이터와 일치하는지 확인	10	일치


[표 8] 지역별 미세먼지 정보조회 테스트

iOS T-map	먼지아나
-----------	------

	
결과	<p>실제 서비스 중인 T-map App 으로 창원시 의창구청에 대한 검색 결과와 먼지아나 App 에서 의창구청을 선택했을 때 생성되는 지도의 위치가 같다.</p>

[표 8.1] 지역별 미세먼지 정보조회 테스트

T-map	먼지아나
-------	------

<div> <div>측정소</div> <div>21시</div> <div>[경남 창원시]경화동</div> <div>14</div> <div>[경남 창원시]명서동</div> <div>27</div> <div>[경남 창원시]봉암동</div> <div>17</div> <div>[경남 창원시]사파동</div> <div>14</div> <div>[경남 창원시]성주동</div> <div>27</div> <div>[경남 창원시]용지동</div> <div>21</div> <div>[경남 창원시]웅남동</div> <div>28</div> <div>[경남 창원시]월영동</div> <div>19</div> <div>[경남 창원시]회원동</div> <div>28</div> </div>	<p>에어 코리아(한국 환경공단)에서 제공하는 창원지역 21:00 기준 측정소별 미세먼지 수치이다. 위 값의 평균값은</p> $\frac{14+27+17+14+27+21+28+19+28}{9} = 21.666666 \approx 22(\mu\text{g}/\text{m}^3)$ <p>이다. ‘먼지아나’ 에서 제공하는 값과 근사한 것을 확인할 수 있다.</p>
	 <p>주변측정소 조회</p> <p>업데이트 시간: 2020-12-08 21:00 지역구: 창원시 미세먼지 수치: 22 $\mu\text{g}/\text{m}^3$ 미세먼지 등급: 좋음 등급기준: 한국환경공단 통합대기환경지수 설정한 행정구역의 평균 측정 값으로 실제와 다를 수 있음</p> <p>텍스트 뷰에 출력되는 데이터는 21:00 기준 창원 내 측정소의 21:00 기준 미세먼지 수치의 평균값이다.</p>
결과	<p>실제 서비스 중인 에어코리아의 창원지역 미세먼지 시간대별 평균값과 먼지아나에서 제공하는 창원지역 미세먼지 평균값은 일치한다.</p>

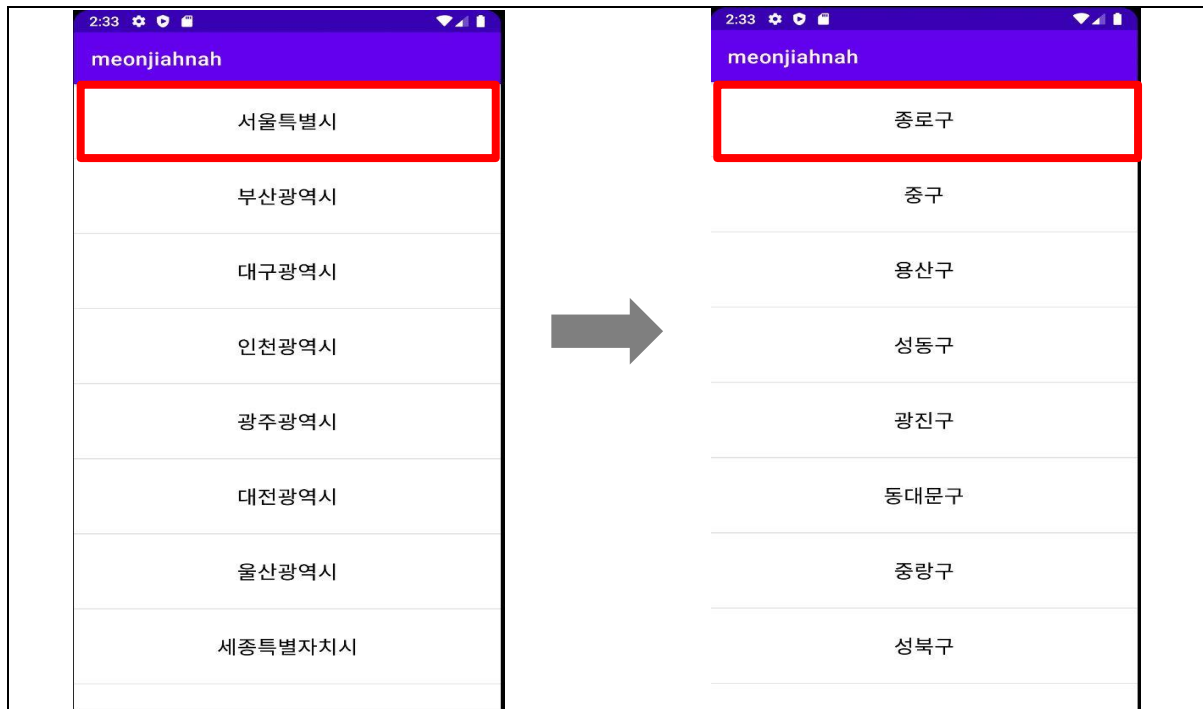
[표 8.2] 미세먼지 데이터 일치 테스트

8.1.2 지역설정

테스트 시나리오	테스트 횟수	예상결과
GUI 로 지역선택 시 선택한 지역과 T-map 지도 생성에 필요한 지역 파라미터 값이 일치하는지 확인	30	일치

[표 8.3] 지역설정 테스트 시나리오

먼지아나



GUI 를 통해 서울특별시와 종로구를 선택한다.

Console Log

```
String input = "";
if(region.length == 3){// 일반구가 있는 시를 선택했을 경우
    input = selectstate+"청";
    System.out.println("input"+input);//테스트 로그
}
else {
    char last = region[1].charAt(region[1].length() - 1);//최소행정구역 끝자리 구분
    input = selectstate;
    System.out.println("input"+input);//테스트 로그
    if(last=='구' || last=='군' || last=='시') {
        input = selectstate + "청";
        System.out.println("input"+input);//테스트 로그
    }
}
```

/*지역명 검색으로 좌표값을 반환하는 메소드*/

```
tmapdata.findTitlePOI(input, new TMapData.FindTitlePOIListenerCallback() {
```

위 코드는 GUI 로부터 입력 받은 값을 문자열 데이터로 저장하여 지역 관공서를 검색하기 위해 가공하는 과정을 나타낸다. 마지막 행의 코드는 지도를 생성하기 위한 T-map API 에서 제공하는 메서드로 앞서 가공한 input 이라는 문자열 변수를

파라미터로 가진다. 해당 코드에서 변수 값을 콘솔로 출력했을 때 아래와 같다.

```
I/System.out: 스레드 내에서 호출됨:2020-12-08 23:00 종로구 27
I/System.out: 스레드 리턴값: 2020-12-08 23:00 종로구 27
    input서울특별시 종로구
    input서울특별시 종로구청
```

결과	GUI 로 선택한 지역정보와 실제 지도를 생성하는데 사용되는 파라미터 값이 같은 것을 확인할 수 있다.
----	-----------------------------------------------------------

[표 8.4] 지역설정 테스트

9.1.3 주변 측정소 조회

테스트 시나리오	테스트 횟수	예상결과
지도상에 생성된 마커가 실제 대기 질 측정소 위치와 일치하는지 확인	30	일치

[표 8.5] 주변 측정소 조회 시나리오

Console Log
<pre>Element eElement = (Element) nNode; Log.d(tag: "AdjacentStationList", msg: "station name : " + getTagValue(tag: "stationName", eElement)); Log.d(tag: "AdjacentStationList", msg: "addr : " + getTagValue(tag: "addr", eElement)); stationlist+=getTagValue(tag: "addr", eElement)+",";</pre> <p>위 코드는 ‘주변 측정소 조회’ 코드를 선택했을 때 지도 중심점과 인접한 측정소의 위치 데이터를 XML 형식으로 받아서 가공하는 코드의 일부이다. 이 부분에 콘솔 로그를 출력하는 코드를 삽입해서 로그에 인접 측정소 좌표를 출력하도록 구성한다. 그 다음 App 을 실행하여 로드해 생성되는 측정소 주소를 에어코리아에서 제공하는 측정소 정보와 대조시킨다.</p> <pre>D/AdjacentStationList: station name : 중구 addr : 서울 중구 덕수궁길 15시청서소문별관 3동 station name : 종로 addr : 서울 종로구 종로 169(종묘주차장 앞)</pre> <p>Appd 을 실행시키고 서울특별시 종로구를 선택했을 때 위와 같은 로그가 출력된다.</p>
면지아나

측정소명	중구 측정소 (도시대기)
운영기관	서울특별시보건환경연구원
주소	서울 중구 덕수궁길 15 시청서소문별관 3동
설치년도	1995
측정항목	SO ₂ CO O ₃ NO ₂ PM ₁₀ PM _{2.5}
측정소명	종로 측정소 (도로변대기)
운영기관	서울특별시보건환경연구원
주소	서울 종로구 종로 169 (종묘주차장 앞)
설치년도	2008
측정항목	SO ₂ CO O ₃ NO ₂ PM ₁₀ PM _{2.5}

위 사진은 한국환경공단에서 운영하는 에어코리아 라는 미세먼지 정보제공 사이트에서 서울 특별시 중구와 종로구의 측정소 위치정보를 조회한 결과이다 먼지아나의 마커를 생성하는 로그에서 확인한 주소정보와 동일한 것을 확인할 수 있다.

결과	일치함
----	-----

[표 8.6] 지역설정 테스트

8.1.4 측정소별 주간 예보조회

테스트 시나리오	테스트 횟수	예상결과
웹 브라우저에서 Air Quality Forecast API 에 App 과 동일한 측정소에 대한 쿼리를 요청해 데이터 일치여부를 확인	10	일치

[표 8.7] 지역별 미세먼지 정보조회 시나리오

먼지아나

meonjiahnah

위치: Gaeumjeong-dong Changwon-si
Gyeongnam
South Korea (창원시 가음정동 경남)

날짜: 2020-12-07미세먼지 예측 평균값 : 32 $\mu\text{g}/\text{m}^3$

날짜: 2020-12-08미세먼지 예측 평균값 : 11 $\mu\text{g}/\text{m}^3$

날짜: 2020-12-09미세먼지 예측 평균값 : 6 $\mu\text{g}/\text{m}^3$

날짜: 2020-12-10미세먼지 예측 평균값 : 21 $\mu\text{g}/\text{m}^3$

날짜: 2020-12-11미세먼지 예측 평균값 : 44 $\mu\text{g}/\text{m}^3$

날짜: 2020-12-12미세먼지 예측 평균값 : 20 $\mu\text{g}/\text{m}^3$

날짜: 2020-12-13미세먼지 예측 평균값 : 27 $\mu\text{g}/\text{m}^3$

날짜: 2020-12-14미세먼지 예측 평균값 : 3 $\mu\text{g}/\text{m}^3$

JSON 데이터

해당 로그정보로 웹 브라우저에서 Air Quality Forecast 서버에 쿼리를 요청하면 아래와 같은 JSON 데이터가 출력된다.

```
← → ↺ 🏠 https://api.waqi.info/feed/geo:35.19931153;128.71095984/?token=2cb3f47c5dc26210f66b28597dd5e845338c6396

{"status":"ok","data":{"aqi":70,"idx":4019,"attributions":[{"url":"https://www.airkorea.or.kr/","name":"South Air Korea Environment Corporation"}, {"url":"https://aqicn.org/city/korea/gyeongnam/changwon-si/gaeumjeong-dong","name":"South Air Korea Environment Corporation"}, {"url":"https://aqicn.org/city/korea/gyeongnam/changwon-si/gaeumjeong-dong","name":"South Air Korea Environment Corporation"}],"dominantpol":"pm25","time":{"tz":"+09:00","v":1607475600,"iso":"2020-12-09T01:00:00+09:00"},"forecast":{"daily":{"o3":{"avg":3,"day":"2020-12-07","max":14,"min":12,"max":4,"min":1},"pm10":{"avg":32,"day":"2020-12-07","max":51,"min":18},"pm25":{"avg":11,"day":"2020-12-07","max":49,"min":4},"avg":6,"day":"2020-12-08","max":19,"min":0},"pm25":{"avg":101,"day":"2020-12-07","max":137,"min":67},"avg":37,"day":"2020-12-08","max":128,"min":0},"pm25":{"avg":22,"day":"2020-12-14","max":63,"min":12},"uvi":{"avg":1,"day":"2020-12-08","max":2,"min":0},"debug":{"sync":"2020-12-09T01:20:46+09:00"}}
```

해당 데이터를 웹 상에서 제공되는 JSON 변환기를 사용했을 때 아래와 같은 구조를 가지고 있는 것을 확인할 수 있는데 먼지아나의 화면에 출력되는 정보는 'Pm10'에 해당하는 데이터인데 먼지아나의 값과 대조해 봤을 때 일치하는 것을 확인할 수 있다.

object ▶ data ▶ forecast ▶ daily ▶ pm10 ▶ 2			
▼ object {2}			
status : ok			
▼ data {9}			
aqi : 70			
idx : 4019			
▶ attributions [2]			
▼ city {3}			
▶ geo [2]			
name : Gaeumjeong-dong, Changwon-si, Gyeongnam, South Korea (창원시 가음정동 경남)			
url : https://aqicn.org/city/korea/gyeongnam/changwon-si/gaeumjeong-dong			
dominentpol : pm25			
▶ iaqi {12}			
▶ time {4}			
▼ forecast {1}			
▼ daily {4}			
▶ o3 [6]			
▼ pm10 [9]			
avg : 32			▼ 5 {4}
day : 2020-12-07			avg : 44
max : 51			day : 2020-12-11
min : 18			max : 45
▼ 1 {3}			min : 27
avg : 11			▼ 6 {4}
day : 2020-12-08			avg : 20
max : 49			day : 2020-12-12
2 : 4			max : 45
▼ 3 {4}			min : 13
avg : 6			▼ 7 {4}
day : 2020-12-09			avg : 27
max : 9			day : 2020-12-13
min : 0			max : 55
▼ 4 {4}			min : 18
avg : 21			▼ 8 {4}
day : 2020-12-10			avg : 3
max : 27			day : 2020-12-14
min : 9			max : 19

위 그림은 창원시 성산구 주변 측정소의 미세먼지 예측 평균값이다. 해당 주의 월요일부터 차주 월요일까지의 예측 정보를 보여준다.

35.19931153 128.71095984

미세먼지 예보 API스레드 리턴값: {"status":"ok","data":{"aqi":70,"idx":4019,"attributions":[{"url":"https://www.airkorea.or.kr/"}]}

위 코드는 마커를 선택했을 때 마커의 좌표값과 Air Quality Forecast Api 를 사용해 쿼리를 요청할 때 발생하는 로그이다.

결과	일치함
----	-----

[표 8.8] 지역별 미세먼지 정보조회 테스트

8.2 실행 테스트

Android Studio 를 사용하여 제작한 App 을 배포형식인 .apk 파일을 생성하여 기기 테스트를 진행했다. 테스트 환경은 Google Firebase TestLab 에서 제공하는 가상기기와 스마트 모바일 앱 개발 지원센터에서 제공하는 테스트 도구를 사용하여 원격으로 실제 기기에 접속하여 테스트를 진행했다. TestLab 에서의 실행 테스트는 기기에 앱을 설치하고 App UI 복잡도에 따라 자동화된 프로그램이 UI 간 상호작용 테스트를 진행한다. 테스트 진행 결과는 아래와 같다.

0:01 기본 활동 시작	0:16 요소 탭함 android.widget.LinearLayout
0:04 요소 탭함 android.widget.LinearLayout	0:16 뒤로 버튼 누름
0:05 요소 탭함 android.widget.LinearLayout	0:17 요소 탭함 android.widget.LinearLayout
0:07 요소 길게 누름 android.opengl.GLSurfaceView	0:19 뒤로 버튼 누름
0:08 요소 탭함 android.opengl.GLSurfaceView	0:19 요소 탭함 android.widget.LinearLayout
0:08 {TARGET} 탭함 button3	0:20 뒤로 버튼 누름
0:09 뒤로 버튼 누름	0:20 요소 탭함 android.widget.LinearLayout
0:09 요소 탭함 android.widget.LinearLayout	0:22 뒤로 버튼 누름
0:10 뒤로 버튼 누름	0:22 요소 탭함 android.widget.LinearLayout
0:11 요소 탭함 android.widget.LinearLayout	

[그림 8] 자동화 프로그램 App 동작 시나리오 캡처

1 분동안 UI 상호작용 테스트를 진행하고 응답시간과 기기자원 사용량을 기록한다. 테스트에서 지역을 선택하고 지도가 로딩되는 구간에서 응답지연시간이 가장 크고 CPU, RAM 사용량이 가장 많았던 것을 확인할 수 있다.



[그림 8.1] App 기기자원 사용량 캡처

스마트 모바일 앱 개발 지원센터에서 제공하는 원격으로 실제 기기에 접속하여 App 실행 테스트 환경을 제공하는 서비스로 테스트한 기기는 6 종류이고 기기정보는 아래와 같다.

제조사	모델명	운영체제 버전	해상도	디스플레이 크기(인치)	프로세서
Samsung	SM-N971N	9.0	1080 X 2280	6.3	Samsung Exynos Modem 5100
Samsung	SM-A908	9.0	1080 X 2400	6.7	퀄컴 스냅드래곤 855 SM8150
Samsung	SM-G977	9.0	1440 X 3040	6.7	삼성 엑시노스 9 Series (9820) SoC
Samsung	SM-G960N	9.0	1440 X 2960	5.8	삼성 엑시노스 9 Series (9810) SoC
LG	LM-V500N	9.0	1440 X 3120	6.4	퀄컴 스냅드래곤 855 SM8150
LG	LM-V510N	9.0	1080 X 2340	6.4	퀄컴 스냅드래곤 855 SM8150

[표 8.9] 테스트 기기정보

✓ 앱 설치/정상실행 테스트 * 앱 실행 체크 시간 설정 : 시작 후 초

✓ 앱 삭제 테스트

✓ 앱 실행 전/후 CPU 비교테스트

✓ 앱 실행 전/후 Memory 비교테스트

수행 횟수 : 회

Log Level :

[그림 8.2] 테스트 종류 정보

기기 테스트 기준은 위 그림과 같다. 각 기기별로 3 번에 걸쳐 설치, 실행, 삭제를 수행한다. 앱 실행 후 5 초간 대기하여 실행여부를 판단하는데 실행 테스트 중 로그가 발생했을 때 실행 실패를 판단하는 로그 레벨은 아파치 Log4j 기준 Error 로그가 발생하면 실행 실패라고 판단한다.

meonjiahnah			
서울특별시	[12-08 11:01:22.963]	W	632:632 keymaster_tee [WRN]begin req PARAMS: A32 B2 P64
	[12-08 11:01:22.964]	I	959:959 keystore] 1 0
	[12-08 11:01:22.965]	I	959:959 KeystoreOperation] Block api uploadOpAsProto, wasSuccessful
	[12-08 11:01:22.966]	W	959:959 keystore] ENTER remove operation from uid 1000
	[12-08 11:01:22.967]	W	632:632 keymaster_tee [WRN]start nwd_get_key_characteristics
부산광역시	[12-08 11:01:22.968]	W	632:632 keymaster_tee [WRN]ret OK PARAMS: A32 P0 P1 B2 P64 AR1 S
	[12-08 11:01:22.969]	W	632:632 keymaster_tee [WRN]start nwd_get_key_characteristics
	[12-08 11:01:22.970]	W	632:632 keymaster_tee [WRN]ret OK PARAMS: A32 P0 P1 B2 P64 AR1 S
	[12-08 11:01:22.971]	W	632:632 keymaster_tee [WRN]start nwd_get_key_characteristics
	[12-08 11:01:22.972]	W	632:632 keymaster_tee [WRN]ret OK PARAMS: A32 P0 P1 B2 P64 AR1 S
대구광역시	[12-08 11:01:22.973]	W	632:632 keymaster_tee [WRN]begin req PARAMS: A32 B2 P64
	[12-08 11:01:22.974]	I	959:959 keystore] 1 0
	[12-08 11:01:22.975]	I	959:959 KeystoreOperation] Block api uploadOpAsProto, wasSuccessful
	[12-08 11:01:22.976]	W	959:959 keystore] ENTER remove operation from uid 1000
	[12-08 11:01:22.977]	W	959:959 keystore] ----- beginning of system
인천광역시	[12-08 11:01:23.134]	W	959:959 keystore] ----- beginning of system
	[12-08 11:01:23.135]	I	1340:1340 Telecom:SamsungPrebindingServiceImpl] isSkTPhone
	[12-08 11:01:23.136]	I	1340:1340 Telecom:SamsungPrebindingServiceImpl] needToKeep
	[12-08 11:01:23.470]	I	1340:1688 WifiVendorHal] getWifiLinkLayerStats(.13140) before call
	[12-08 11:01:23.483]	I	1340:1688 WifiVendorHal] getWifiLinkLayerStats(.13140) after call
광주광역시	[12-08 11:01:23.500]	I	1340:1688 WifiStateMachine] checkScoreBasedQuality - mPrevio
	[12-08 11:01:23.823]	I	1340:1807 ActivityManager] [ML]Checking for the Machine Learn
	[12-08 11:01:24.137]	I	813:813 SurfaceFlinger] EffectController destroyed!
	[12-08 11:01:24.138]	I	813:813 DisplayDevice] DisplayDevice destroyed : DisplayDevice=phoner
	[12-08 11:01:24.230]	E	810:1491 NativeSemDvfsCpuManager] release: Start
대전광역시	[12-08 11:01:24.231]	E	810:1491 NativeSemDvfsCpuManager] release(): mIsAcquired = 1, mNan
	[12-08 11:01:24.232]	E	810:1491 NativeCustomFrequencyManager] [NativeCFMS] bpCustomFrequ
	[12-08 11:01:24.233]	E	810:1491 NativeSemDvfsCpuManager] release: End
	[12-08 11:01:24.268]	I	813:813 SurfaceFlinger] EffectController destroyed!
	[12-08 11:01:24.270]	I	813:813 DisplayDevice] DisplayDevice destroyed : DisplayDevice=phoner
세종특별자치시	[12-08 11:01:24.405]	I	765:25840 sensors-hal] handle_sns_std_sensor_event:91, light_ct
	[12-08 11:01:24.515]	I	1340:2436 SamsungAlarmManager] setLocked to kernel - T:2 / 2
	[12-08 11:01:24.520]	I	813:813 SurfaceFlinger] EffectController destroyed!
	[12-08 11:01:24.521]	I	813:813 DisplayDevice] DisplayDevice destroyed : DisplayDevice=phoner
	[12-08 11:01:24.688]	I	1340:2436 SamsungAlarmManager] setLocked to kernel - T:2 / 2
경기도	[12-08 11:01:24.752]	I	813:813 SurfaceFlinger] EffectController destroyed!
	[12-08 11:01:24.753]	I	813:813 DisplayDevice] DisplayDevice destroyed : DisplayDevice=phoner
	[12-08 11:01:24.956]	I	2113:2856 lmsUtil] fragment is null. get simSlot from priority po
	[12-08 11:01:24.963]	E	2113:2856 CapabilityDiscModule<0>] checkModuleReady: mCon
	[12-08 11:01:24.964]	E	25798:26095 ORC/RcsCapabilityManager] queryOwnCapability, curs
강원도	[12-08 11:01:24.986]	I	813:813 SurfaceFlinger] EffectController destroyed!
	[12-08 11:01:24.987]	I	813:813 DisplayDevice] DisplayDevice destroyed : DisplayDevice=phoner

[표 8.10] 기기 테스트 화면과 로그 일부 캡처(Galaxy A90)

단말기 정보	실행 횟수	결과	결과 상세
단말기명: Galaxy A90 모델명: SM-G960N	3	성공	App 실행 전/후 리소스 변화 전체: CPU(7%), Memory(2,098,020KB) 전체: CPU(21%), Memory(2,241,344KB)
단말기명: Galaxy Note 10 모델명: SM-N971N	3	성공	App 실행 전/후 리소스 변화 전체: CPU(8%), Memory(4,205,024KB) 전체: CPU(21%), Memory(4,255,948KB)
단말기명: Galaxy S10 5G 모델명: SM-G977	3	성공	App 실행 전/후 리소스 변화 전체: CPU(14%), Memory(2,808,960KB) 전체: CPU(22%), Memory(2,974,020KB)
단말기명: Galaxy S9 모델명: SM-G960N	3	성공	App 실행 전/후 리소스 변화 전체: CPU(10%), Memory(2,532,708KB) 전체: CPU(44%), Memory(2,551,744KB)
단말기명: V50 ThinQ 모델명: LM-V500N	3	성공	App 실행 전/후 리소스 변화 전체: CPU(16%), Memory(3,210,572KB) 전체: CPU(21%), Memory(3,216,864KB)
단말기명: V50S ThinQ 모델명: LM-V510N	3	성공	App 실행 전/후 리소스 변화 전체: CPU(13%), Memory(3,522,972KB) 전체: CPU(23%), Memory(3,711,528KB)

[그림 8.11] App 실행 테스트 결과