**모바일 프로그래밍 결과 보고서**

**16조**

201819176 이용현

202012173 김상은

202012252 지수영

202018417 황인효

**<목차>**

**1. 서론**

**1.1 프로젝트 주제**

**1.2 개발 목적 및 관련 기본 앱들의 보완점**

**2. 앱 기능**

**2.1 위치**

**2.2 날씨**

**2.3 옷 추천**

**2.4 음식 추천**

**2.5 위치 재설정**

**3. 결론**

**3.1 개선해야할 사항**

**3.2 역할분담**

**3.3 후기**

**1. 서론**

**1.1 프로젝트 주제**

현재의 날씨 상태와 거기에 맞춰 옷과 음식을 추천해주는 날씨 앱

**1.2 개발 목적 및 관련 기본 앱들의 보완점**

선택한 지역의 날씨 예보를 받아와서 강수량, 하늘 상태, 기온 등을 표출해 현재 날씨 상황을 파악할 수 있는 앱을 제작하였다. 기존의 앱들과는 다르게 기온에 걸맞는 옷차림을 추천하는 기능을 추가했으며 메뉴 선정에 어려움을 느끼는 사람들을 위한 음식 추천 기능도 추가했다.

**2. 앱 기능**

**2.1 위치**

**[UI]**



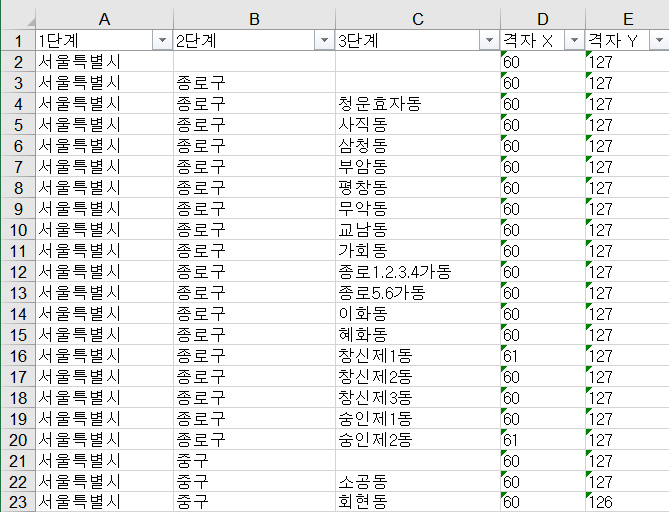
[초기화면]

앱을 구동하면 다이얼로그 창이 실행되면서 콤보박스(스피너)에서 현재 날씨를 알고 싶은 지역을 선택할 수 있다. 첫 번째 콤보박스는 도/특별시/광역시를 선택할 수 있고 선택된 항목에 따라 두 번째 콤보박스의 항목들이 표시가 된다. 두 번째 콤보박스는 시/군/구를 선택할 수 있으며 첫 번째와 마찬가지로 선택된 항목에 따라 3번째 항목들이 표시된다. 지역 선택을 마쳤으면 하단의 확인 버튼을 클릭해 위치 값을 전달한다.

**[코드]**

MainActivity에서는 지역명을 전달받고 location.xls 파일에서 해당 지역의 좌표 값을 찾는다. 엑셀 파일을 사용하기 위해 Gradle 파일에 을 추가했다.

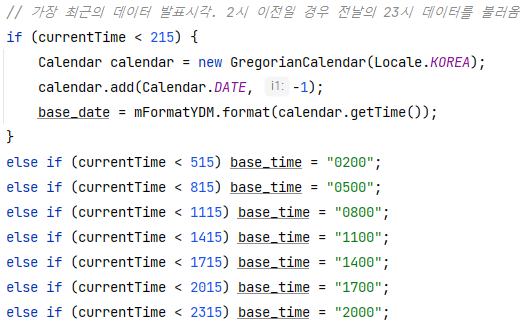
앱 구동과 동시에 Spinner가 실행되지 않도록 초기 isFirstSelected 값을 true로 설정해 놓는다. 시와 동을 선택하는 스피너에도 동일하게 적용한다.



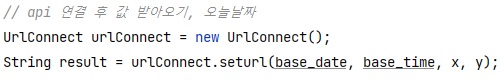
findLactionFirst 일부 발췌

위의 findViewFirst 함수는 도/특별시/광역시 값을 전달받아 엑셀 파일에서 시/군/구를 찾아 SecondSelect ArrayList 배열에 저장한다. 저장된 값은 스피너의 항목으로 들어간다.

확인 버튼을 누르면 WeatherData의 func함수에 좌표값을 전달한다. 좌표값이 전달되어 결과 값을 잘 받아오면 “00”이라는 resultCode가 반환되고 activity\_main 화면으로 전환된다. Move.putExtra 함수를 사용해서 인텐트에 지역명과 날씨 상태를 전달한다.

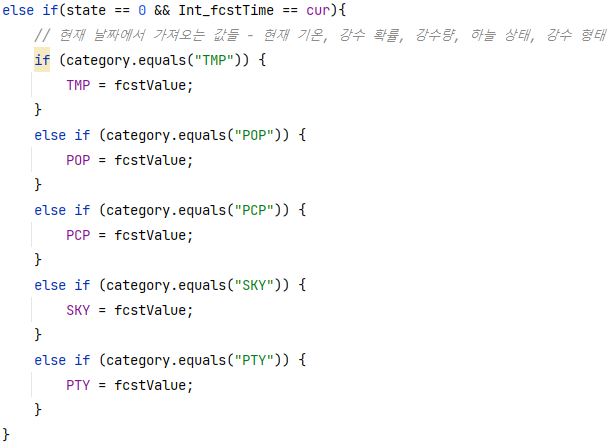
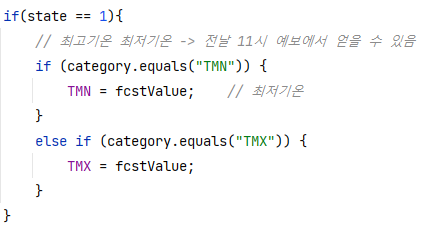
****

WeatherData 클래스에서는 현재 시간을 구하고 API에서 값을 받아오기 위해 base\_time을 형식에 맞게 바꾸어준다. 우리 조가 사용했던 API는 3시간 단위로 예보가 발표되기 때문에 우측 코드와 같은 조건문을 사용해 base\_time을 설정했다.

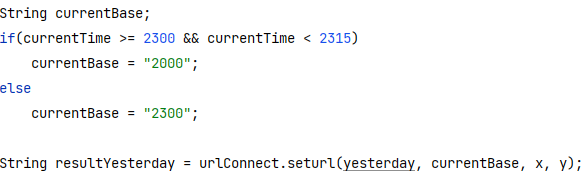


UrlConnect - 현재 날자와 형식에 맞는 base\_time을 urlConnect에 전달해 예보를 받아온다. 받아오는 자료의 타입은 json이며, API는 공공데이터 포털에서 제공하는 단기예보를 사용했다. URL에 접근하기 때문에 manifest 파일에 아래와 같은 권한을 추가해야 한다.

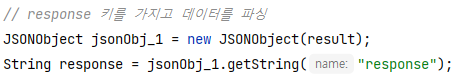


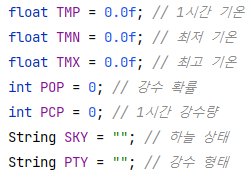
 

ParseData 클래스에서 날씨 정보를 파싱할 때 WeatherData에서 전달받은 state 값에 따라 어떤 값을 추출할 지 결정된다. State가 1이면 전날 23시에 발표된 예보를 가져와 현재 날짜의 최고 기온과 최저기온을 추출할 수 있다. (현재 날짜의 최고기온과 최저기온은 전날 23시 예보에서만 가져올 수 있다.) State가 0이면 현재 날짜를 base\_date로 가져온 것으로 파싱한 fcstTime 중에서 현재 시간과 동일한 시간일 경우에 나머지 값들을 추출한다. (조건식이 없다면 파싱한 예보 중 10시간 이상 차이 나는 미래의 예보를 가져올 수도 있다.)



* 최저 기온과 최고 기온 추출을 위해 전날 날짜를 전달하는 데 23시부터 23시 15분 사이에는 전날 23시 예보를 가져올 수 없다. (기상청 API는 최근 1일간의 정보만 제공하고 예보는 23시 15분에 발표되기 때문) 그렇기 때문에 이 사이 시간대(23시~23시 15분)에는 현재 날짜의 20시에 발표된 예보에서 값을 추출한다.



예보를 받으면 ParseData 클래스를 통해 필요한 값을 추출하고 필요한 형태로 형 변환을 한다. 이 앱에서 사용한 값은 최근 기온과 하루 최저, 최고 기온, 강수 확률, 1시간 강수량, 하늘 상태, 강수 형태가 있다.

강수량을 나타내는 변수 PCP는 “강수 없음” 표시될 때가 있는데 이럴 경우에는 PCP의 값을 0으로 설정해서 전달한다. 하늘 상태를 나타내는 변수 SKY는 상태에 따라 1, 3, 4의 코드 형태로 출력되는데 각각 맑음, 구름 많음, 흐림 상태임을 표현한다. PTY의 경우 강수 형태를 나타내는데 1, 2, 3, 4 등의 코드로 나타나며 마찬가지로 조건문을 통해 각각 비, 비/눈, 눈, 소나기 등으로 변수를 재설정한다.

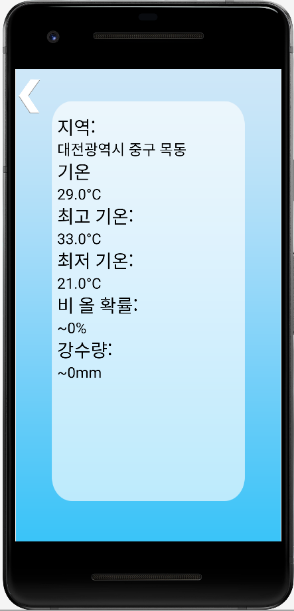
**2.2 날씨**

왼쪽: 낮 시간대 화면, 오른쪽: 밤 시간대 화면



**[UI]**

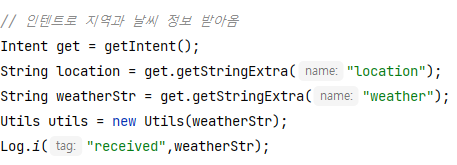
 좌표 값을 전달받으면 weather\_rec 클래스가 실행되면서 액티비티 화면이 전환된다. 상단에는 지역명과 현재 하늘 상태가 표시되고 하단에는 세 개의 이미지 버튼이 생성된다. 첫 번째 이미지 버튼은 기온과 비올 확률, 강수량이 표시되어 있으며 클릭했을 때 sulmyung 액티비티로 전환된다. 화면 전환 시에 인텐트를 통해 현재 날씨와 지역명도 함께 전달된다. 두 번째 이미지 버튼은 옷 추천 이미지 버튼으로 현재 기온에 맞는 옷차림을 추천해준다. 세 번째 이미지 버튼을 누르면 음식 추천 액티비티로 전환되며 랜덤으로 음식을 추천받을 수 있다.



sulmyung.xml

Sulmyung 액티비티는 날씨 정보를 더 상세히 가져와 출력한다. 기온을 비롯해 지역명, 기온, 최고기온, 최저기온, 비 올 확률, 강수량이 표시된다.

**[코드]**



인텐트를 통해 지역명과 날씨 정보를 받아와 화면에 출력한다.

**2.3 옷 추천**

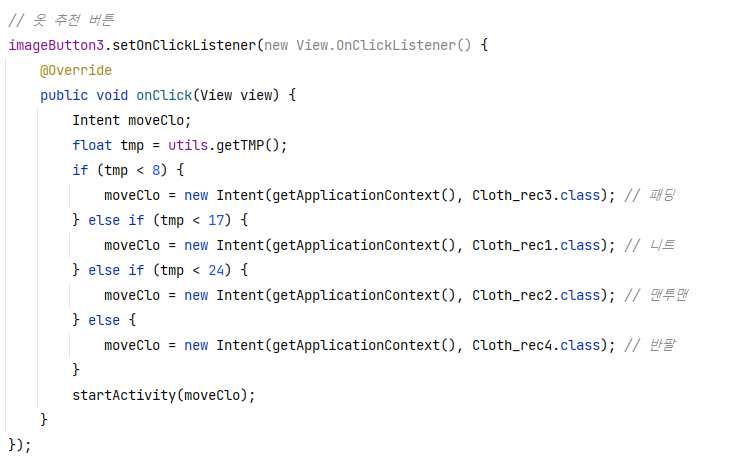
**[UI]**



Cloth\_rec.xml

현재 기온에 따라 Cloth\_rec1~Cloth\_rec4 까지 중 하나의 클래스의 액티비티가 생성된다. 기온이 8도 미만이라면 패딩 이미지가 있는 Cloth\_rec3의 클래스의 액티비티를 생성하고 17도 미만이라면 rec1, 24도 미만이라면 rec2 그 외의 기온에는 Cloth\_rec4 클래스의 액티비티를 생성한다.

**[코드]**

****

기온 값에 따라 다른 인텐트를 생성한다.

**2.4 음식 추천**

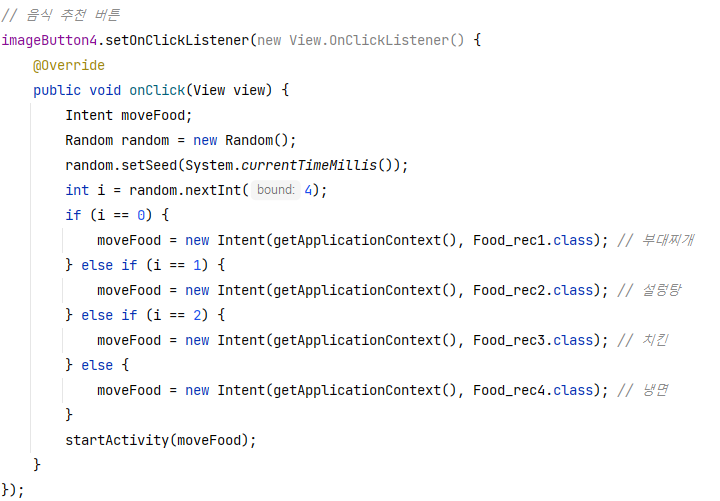
**[UI]**

****

Food\_rec.xml

음식 추천은 랜덤 함수를 발생시켜서 1부터 4까지의 값 중 하나를 받고 해당하는 조건의 클래스들(Food\_rec1~Food\_rec4) 중 하나의 클래스의 액티비티를 랜덤하게 실행시킨다. 음식 추천 종류에는 부대찌개, 치킨, 설렁탕, 냉면이 있다.

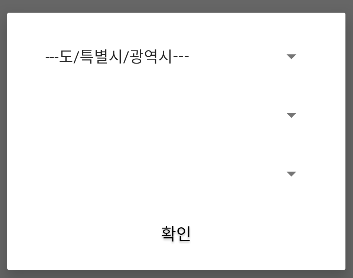
**[코드]**

****

랜덤 함수를 발생시켜 해당하는 숫자의 조건식의 인텐트를 생성한다.

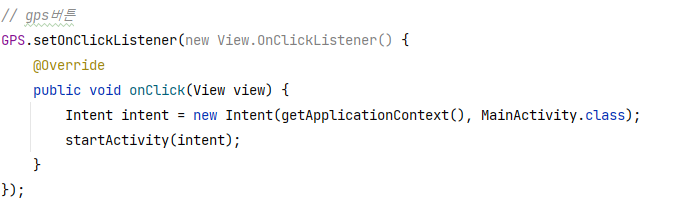
**2.5 위치 재설정**

**[UI]**

****

Activity\_main의 화면 우측 상단에 위치를 재설정할 수 있는 이미지 버튼을 생성하고 버튼을 눌렀을 때 초기 화면으로 전환된다. 초기화면에서 다시 위치를 설정해 날씨 예보를 불러올 수 있다.

**[코드]**

****

MainActiviy 클래스의 액티비티를 불러와 초기화면으로 전환한다.

**3. 결론**

**3.1 개선해야할 사항**

앱을 실행시킬 때마다 지역을 재설정해야 하고 앱을 나갔다 들어왔을 때 현재 시간의 예보가 반영되지 않기 때문에 지역 설정부터 다시 해야한다는 번거로움이 있다. 데이터 베이스를 활용해 자주 확인하는 지역 정보를 저장하여 앱을 재시작 하더라도 바로 원하는 지역의 날씨 상태를 확인할 수 있도록 하고 시간을 업데이트 할 수 있는 버튼을 추가해 현재 시간에 맞는 예보를 표출할 수 있도록 보완해야 한다.

**3.2 역할분담**

UI – 김상은, 이용현

소스코드 – 지수영, 황인효

**3.3 후기**

앱을 만드는 과정에서 API에 접근해 데이터를 파싱해오고 엑셀 파일에서 값을 불러오는 등 여러 형태의 파일을 사용하여 코드를 구성하는 방법을 알 수 있었고 파싱한 데이터를 코드 내부에 원하는 형태로 가공해 적용해볼 수 있었다. 처음 구상했던 것보다 기능이 축소된 것 같아 아쉬움은 남지만 부족한 점을 보완해 앱을 더 기능성 있게 수정해보고 나아가 다른 API를 활용해 새로운 앱을 제작해보고 싶다.

**유튜브 링크**

[**https://youtu.be/w4\_q80CF73w**](https://youtu.be/w4_q80CF73w)