<모바일응용 소프트웨어 설계>

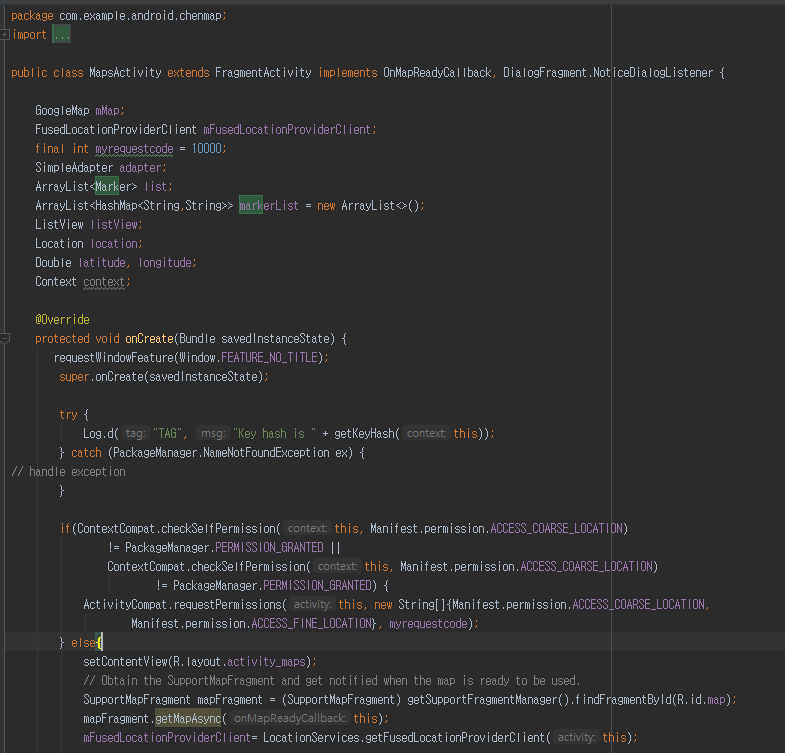
# 4주차 실습 과제

정보통신공학과

12161774 이 채 은

**[소스 코드]**

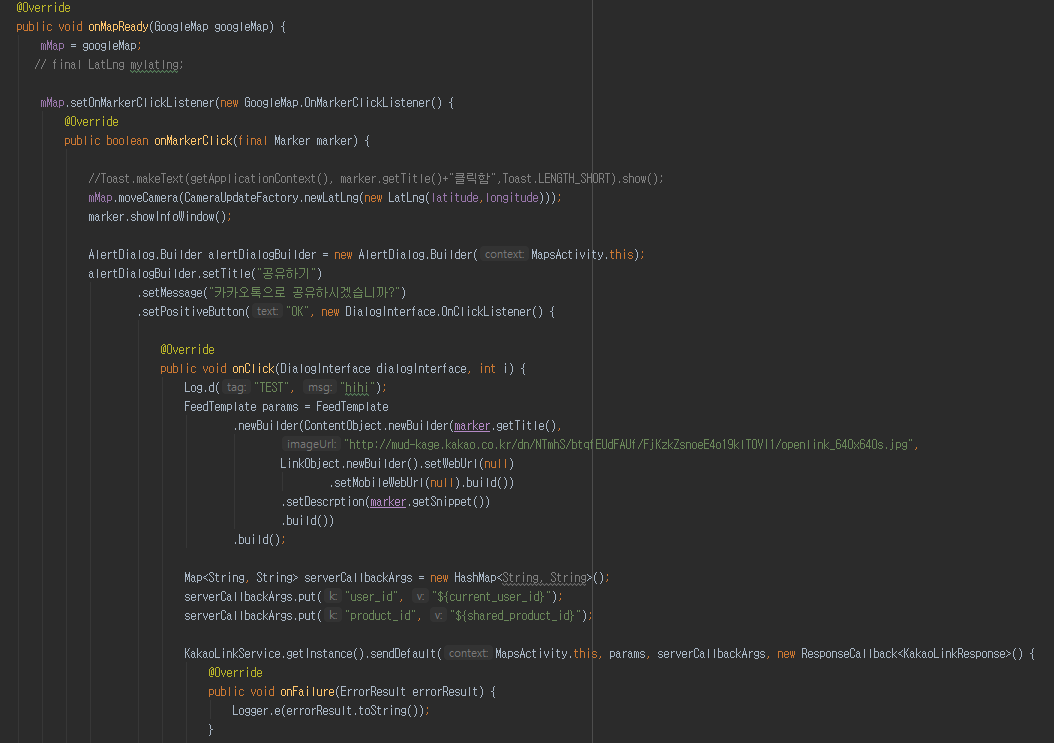
1. MapsActivity.java

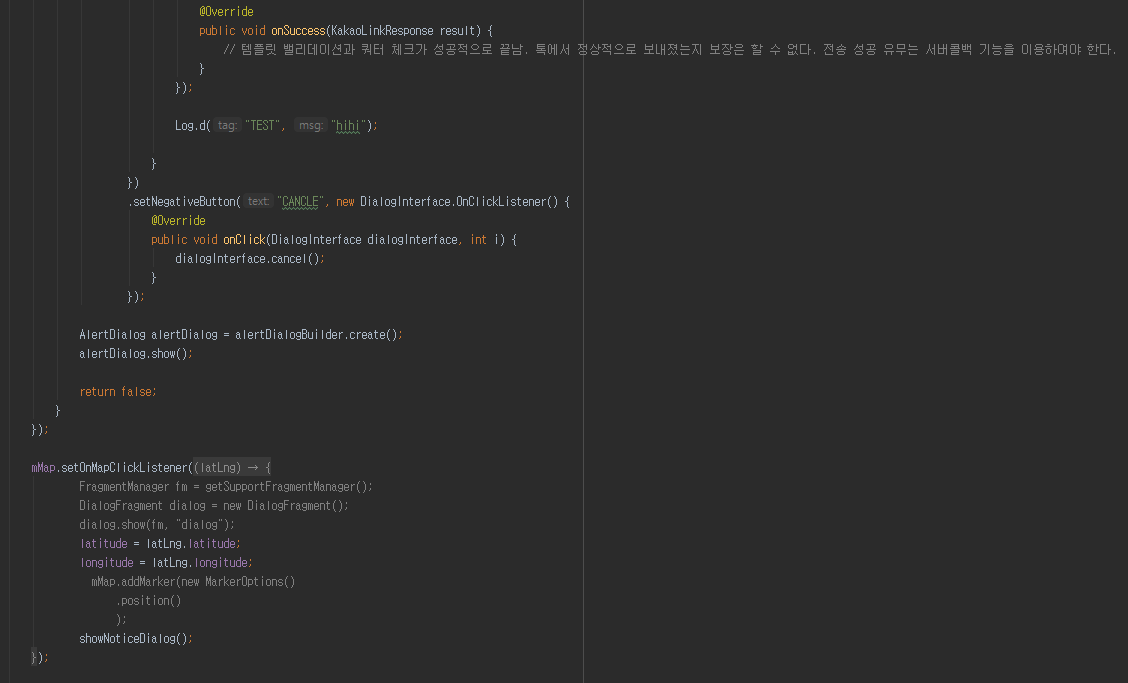


=> 먼저 사용하는 변수들을 선언해주고, OnCreate 에서 런타임 퍼미션을 설정해준다. GPS와 지도를 사용하기 때문에 COARSE, FINE LICATION 관련 퍼미션을 설정한다. 설정이 되어 있다면 else문이 실행되어 SupportMapFragment를 이용해 mapfragment를 불러온다.

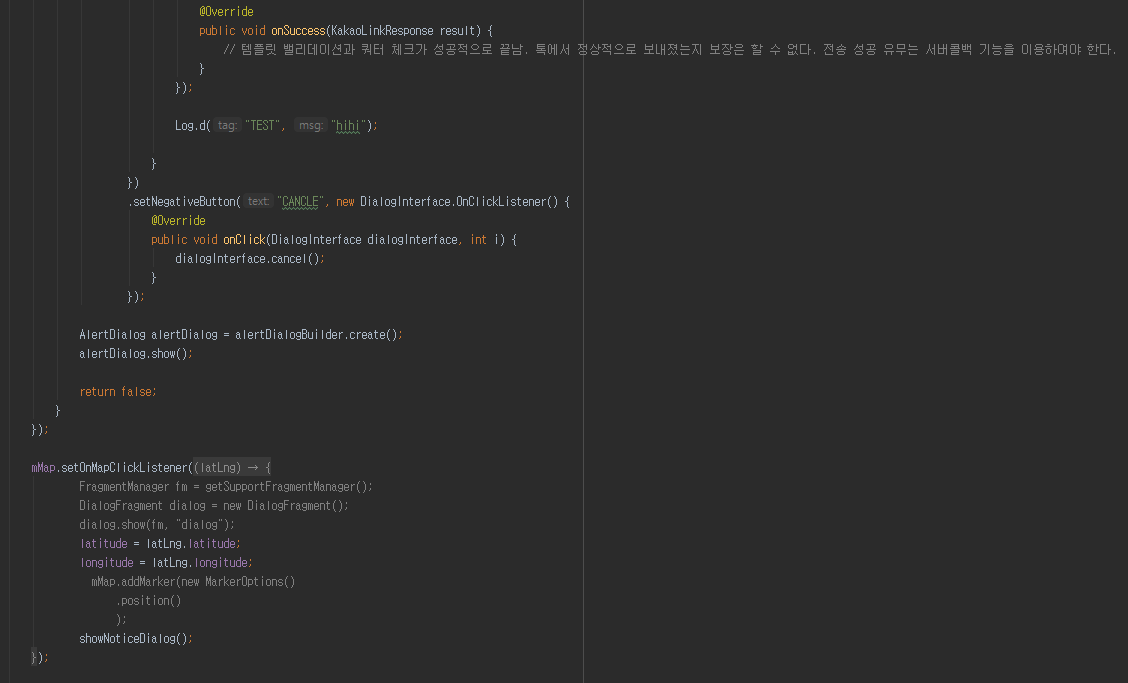


=> listView를 사용하기 위해 listview 객체를 선언한 뒤 listview를 findViewById로 찾아와서 연결해주고, 리스트뷰에 추가된 아이템을 누르면 camera가 그 위치로 이동하도록 moveCamera를 설정해준다.





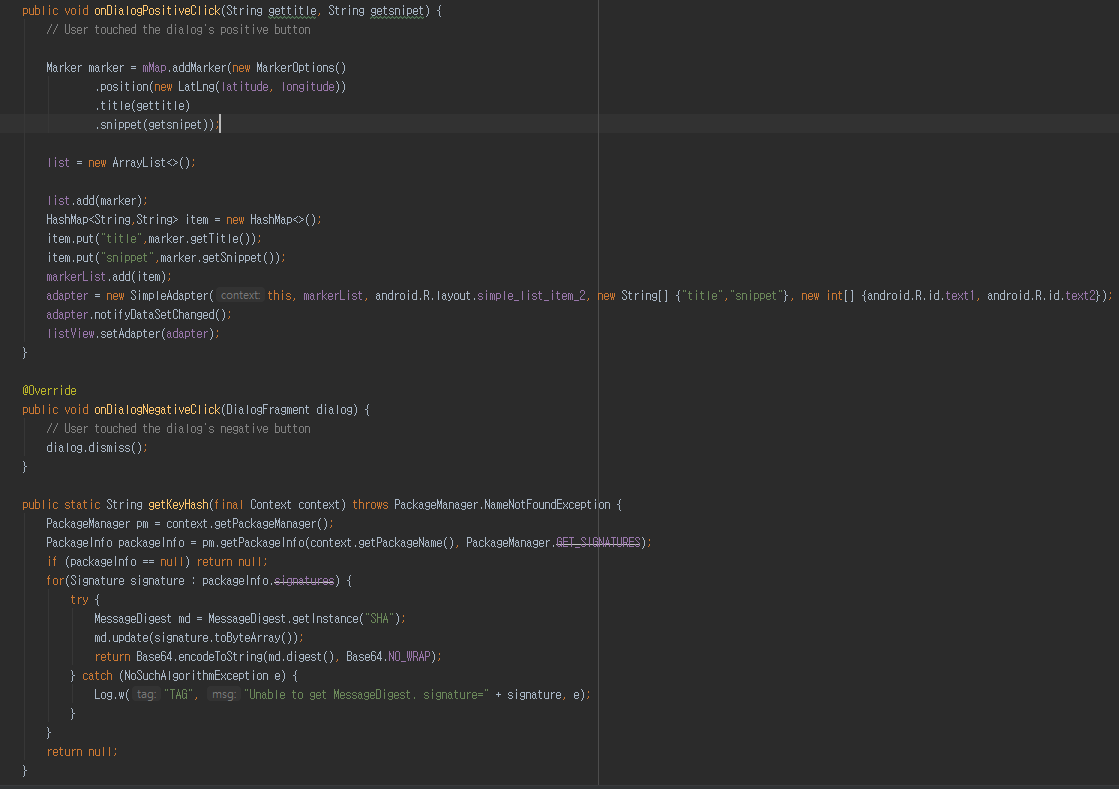
=>map 이 실행되면 onMapReady가 뜬다. 따라서 먼저 mMap이라는 객체로 googleMap을 받아온다. 그 후 마커 클릭 리스너를 설정해서, 마커를 클릭했을 때 이벤트를 정의한다. 마커가 클릭되면 moveCamera를 이용해 그 마커가 화면의 중간에 오도록한다. 또한 AlertDialog를 Builder로 만들어서 공유를 할 수 있도록 dialog를 띄워준다. dialog에는 "OK", "CANCLE" 버튼이 있는데, "OK"를 누르면 설정한 링크로 공유가 된다. 그리고 "CANCLE" 버튼을 누르면 공유가 취소되고, 다시 dialog가 뜨기 전으로 돌아간다.



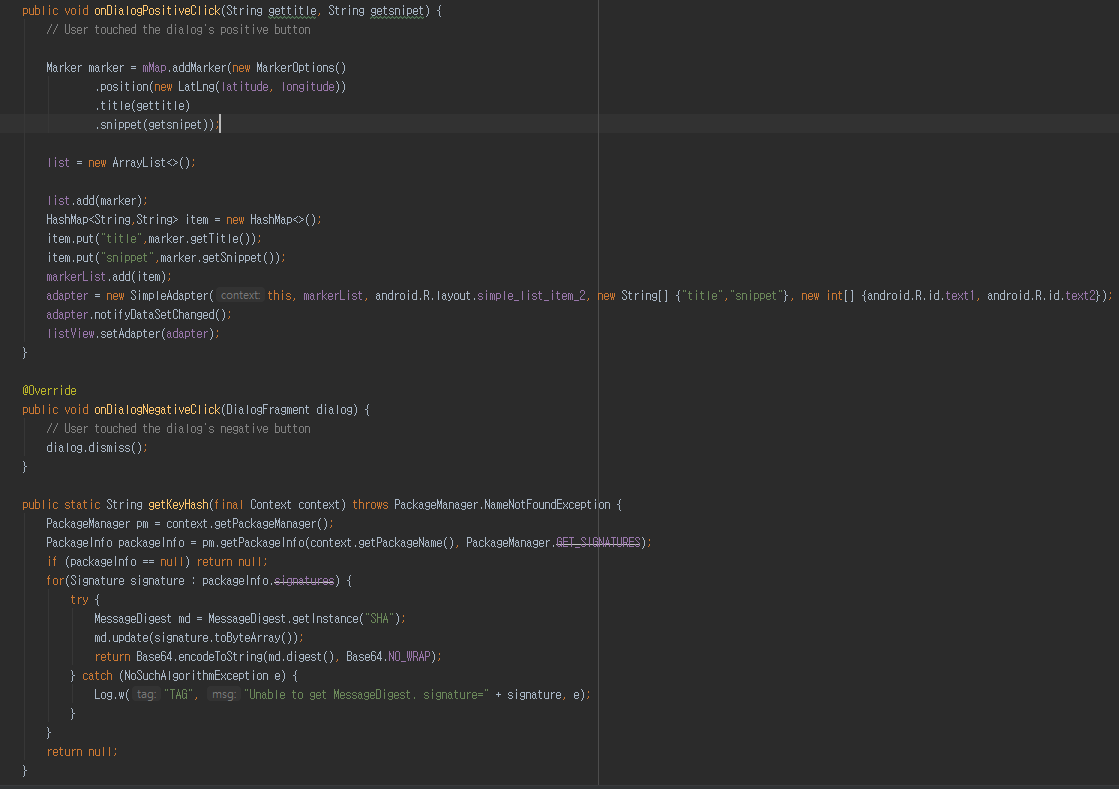
=> 마커가 클릭 됐을 때 또 다른 이벤트로 title, snipet을 설정해주고 리스트뷰에 추가해주기 위한 코드이다. 현재 마커가 클릭된 곳의 latitude, longitude를 받아오고 showNoticeDialog() 메소드를 실행시킨다.



=> showNoticeDialog 함수는 다음과 같다. DialogFragment로 설정된 프래그먼트의 dialog를 불러와서 show 를 통해 보여준다.



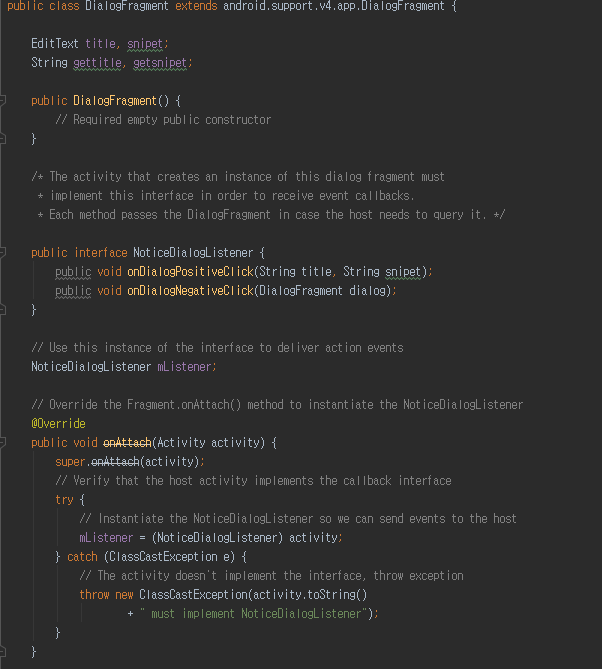
=> DialogPositiveClick 메소드를 만들어서 dialog의 추가 버튼을 눌렀을 때 실행될 부분을 설정해준다. 이 때 DialogPositiveClick, DialogNegativeClick 은 모두 DialogFragment에 이미 설정된 함수로, Main Activity 역할을 하는 MapsActivity와 Dialogfragment 가 콜백 메소드로 연결되어 있기 때문에, dialog의 title edittext, snipet edittext에 내가 직접 입력해준 값을 넘겨 받아와서 사용해줄 수 있다. 따라서 dialog의 추가 버튼을 누르면 우선 marker가 추가된다. Title과 snippet은 내가 설정해준 text로 받아 오기 위해서 DialogFragment에서 받아온 gettitle, getsnippet으로 설정해준다. 그리고 HashMap으로 item을 받아온다. Title과 snippet 둘 다를 한번에 리스트뷰에 보여주기 위해 사용했다. 그리고 마지막에 notifyDataSetChanged를 이용해서 리스트뷰에 추가된 아이템들을 업데이트 해준다.



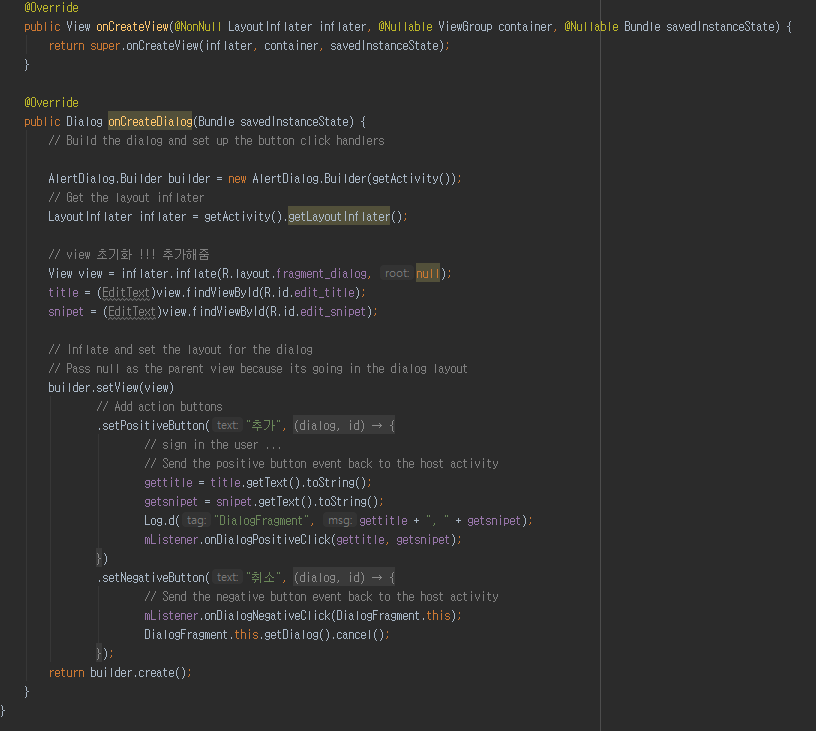
=> onDialogNegativeClick은 취소 버튼을 눌렀을 때 실행되는 내용으로, dialog.dismiss를 실행해서 다이얼로그를 닫아준다.

=> getKeyHash는 해쉬 키를 받아 오기 위해 만들어준 메소드이다.

2. DialogFragment.java

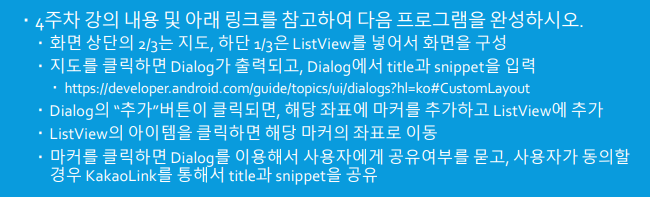


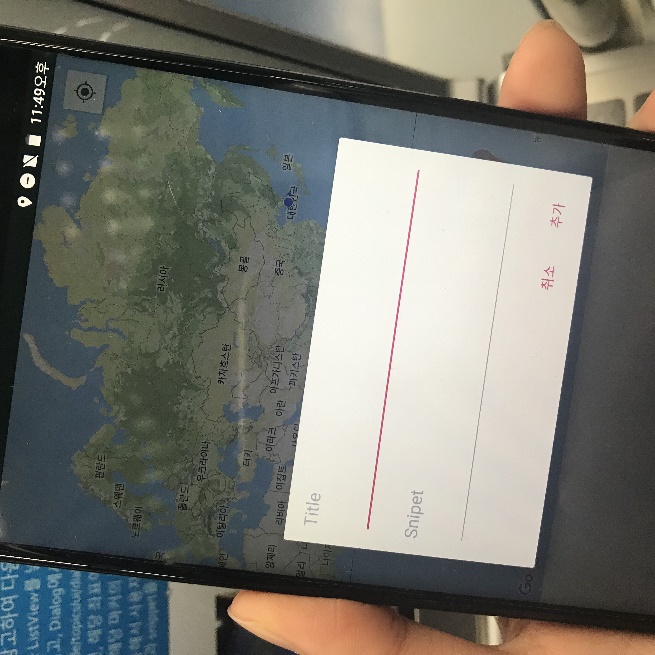
=> NoticeDialogListener mListener로 리스너를 설정해준다. MapsActivity와 정보를 주고 받기 위해서 콜백 리스너를 설정해준것이다. 그리고 onAttach에서 이 리스너를 초기화해서 이벤트를 host activity로 전달할 수 있도록 했다.

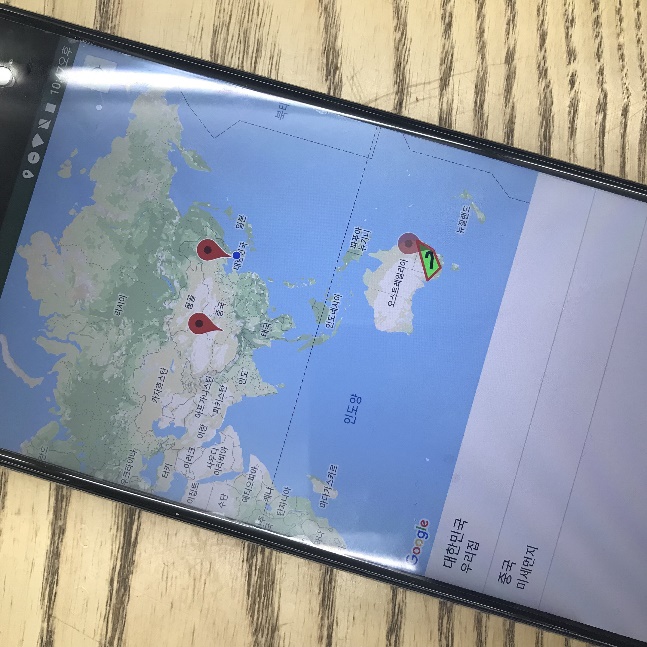


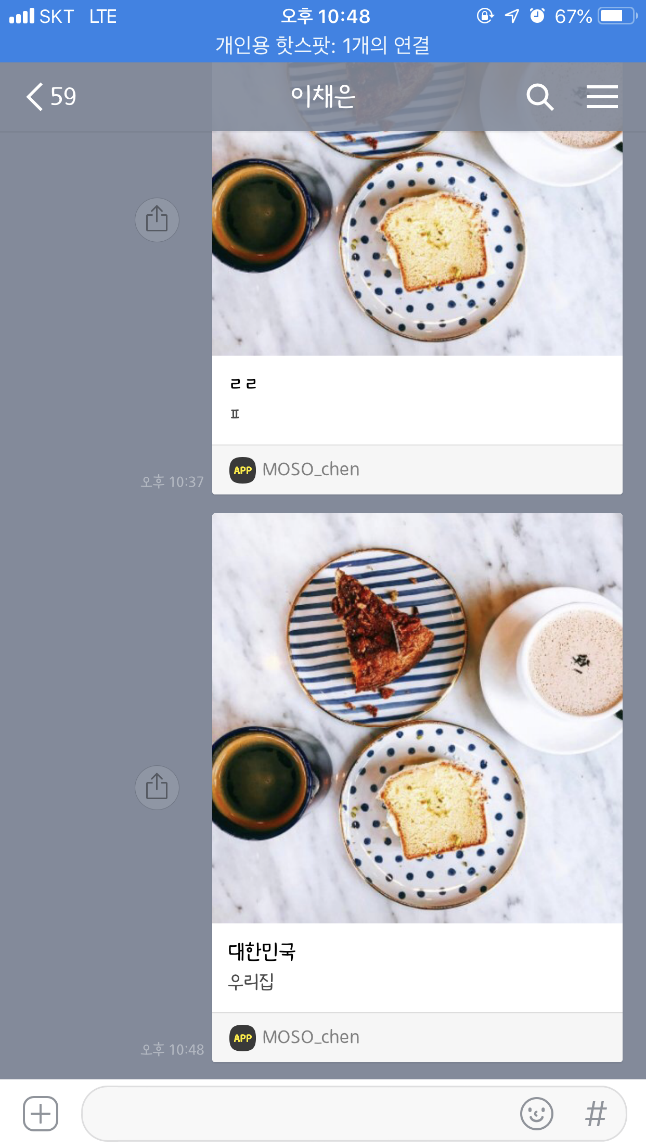
=> onCreateDialog를 통해서 dialog를 만들어준다. Builder로 dialog를 생성하고, inflater로 layoutinflater를 설정해준다. 그리고 view들을 초기화하고 setView를 해서 이 view를 사용한다. setPositiveButton, setNegativeButton을 이용해 추가 / 취소 버튼을 만들고, 각 버튼이 눌리면 실행될 이벤트를 설정해준다. 추가 버튼을 누르면 edittext에 입력된 값을 getText를 이용해서 string으로 받아온다. 이 값은 콜백 리스너인 mListener 를 이용해서 host activity에 전달되어 사용된다. 취소 버튼을 누르면 dialog가 종료된다.

**[실습 결과 및 고찰]**



=> 다음은 실제 실행 결과이다.





=> 앱에 들어가서 장소를 클릭하면 dialog가 뜨면서 title, snipet을 설정할 수 있도록 한다.

그 후 추가 버튼을 누르면 listview에 입력한 정보가 순서대로 추가된다. 그리고 마커를 누르면 마커가 중앙으로 이동하고, 공유하기를 묻는 dialog가 뜬다.

이 dialog에서 공유하기를 누르면 선택한 마커의 title, snippet이 카카오톡으로 공유된다.

=> 이번 과제에서는 google map을 이용해서 마커를 추가하고, 마커에 등록된 정보를 리스트뷰에 나타냈다. 그리고 리스트뷰에 저장된 아이템을 클릭하면 OPEN API를 이용해서 카카오톡 링크로 보내질 수 있도록 했다. OPEN API 와 MAP을 활용해볼 수 있는 과제였다.