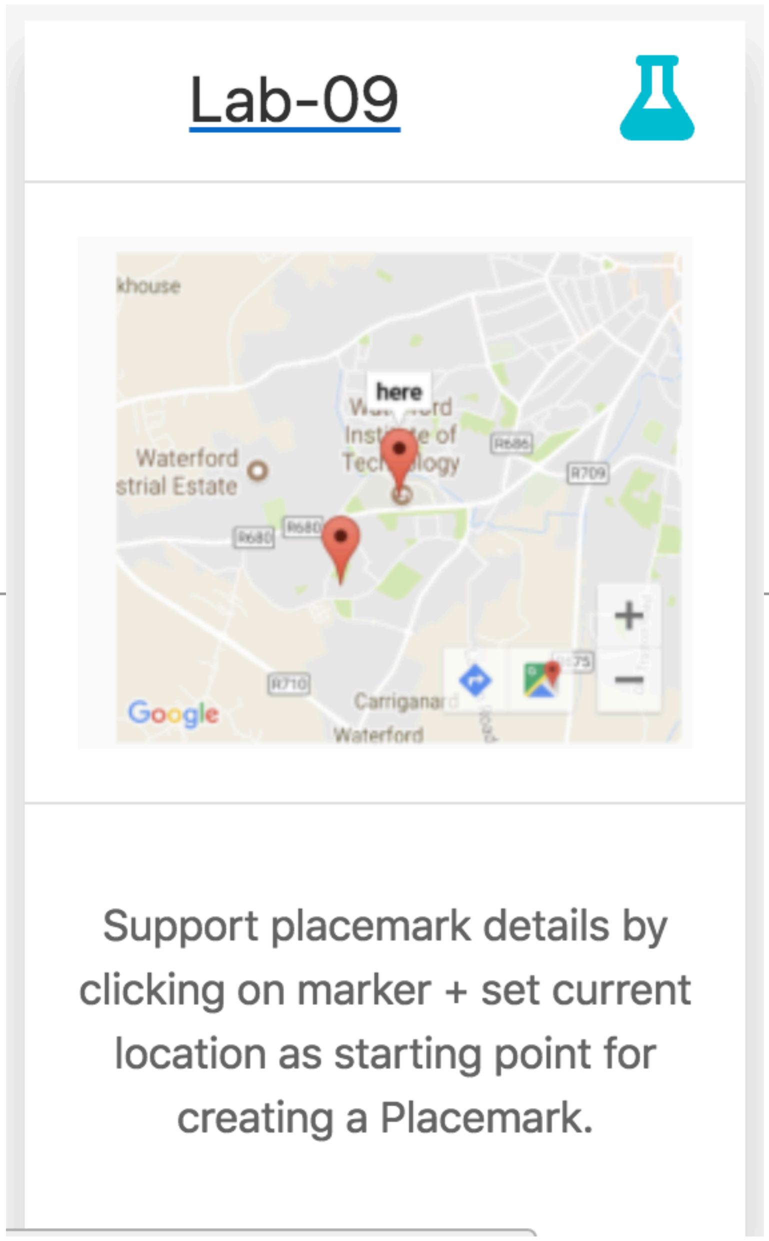
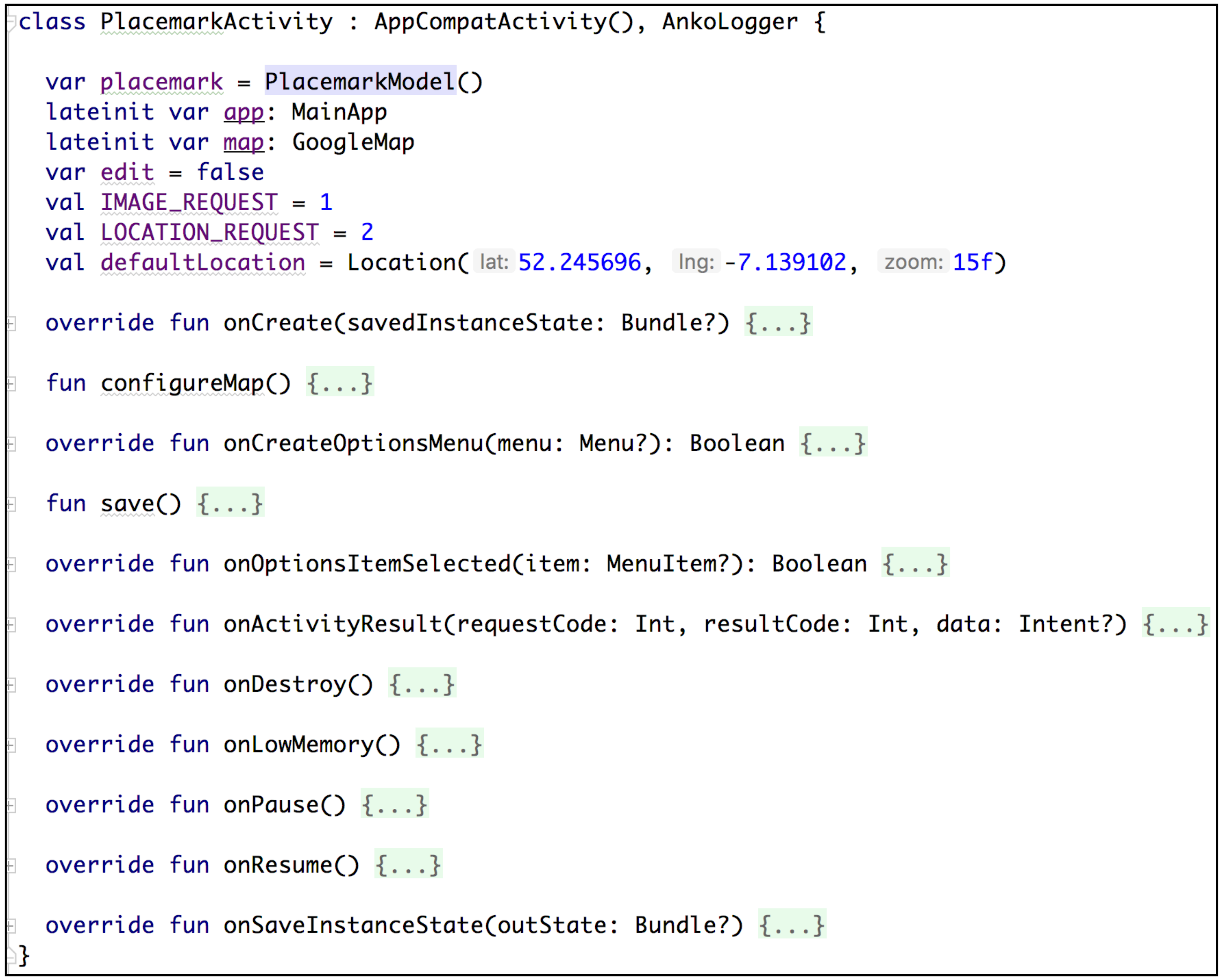
Maps - Current Location


here
Waterford Institute of Technology
Waterford Industrial Estate
Carriganard
Waterford
Google

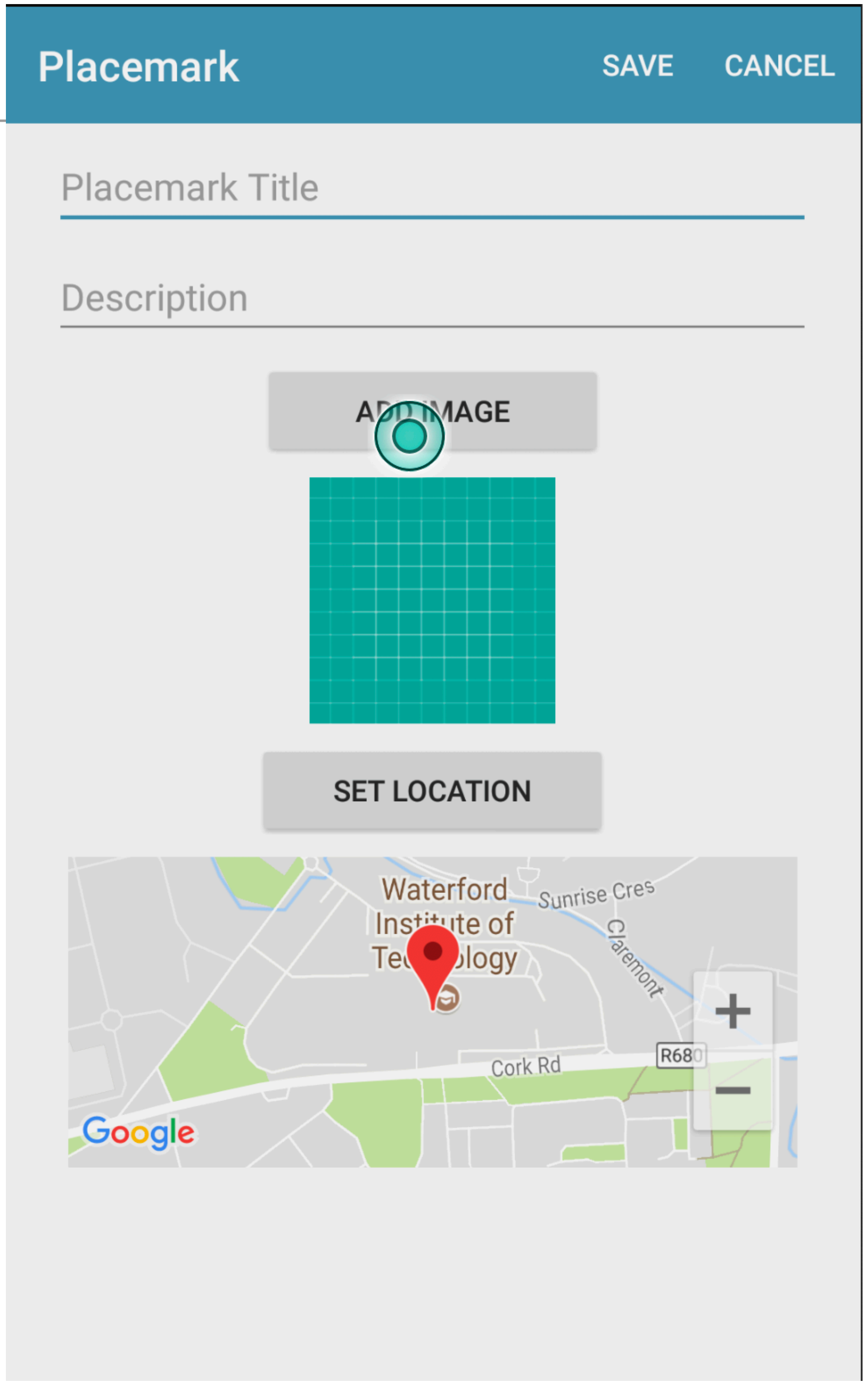
Lab-09

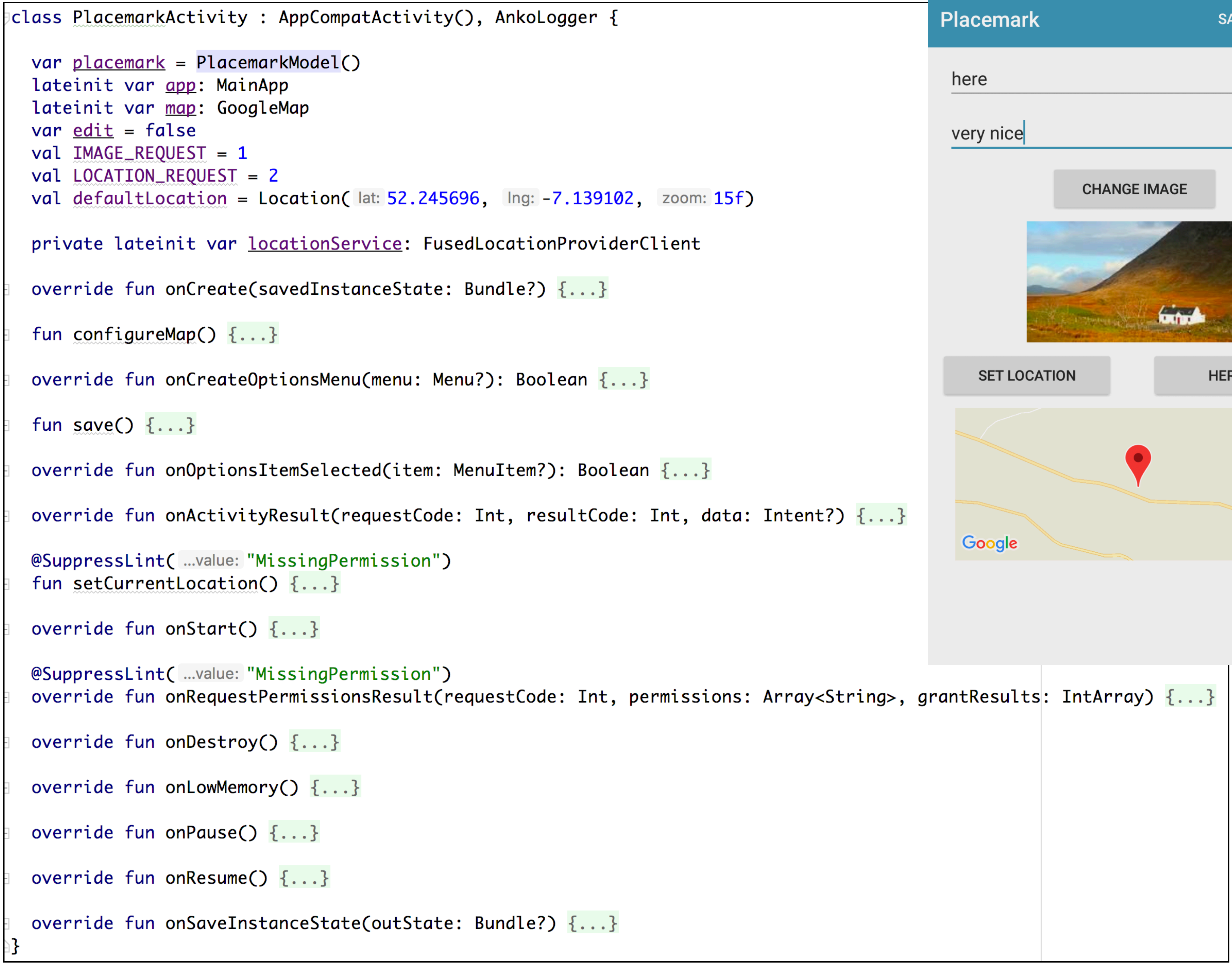
Support placemark details by clicking on marker + set current location as starting point for creating a Placemark.

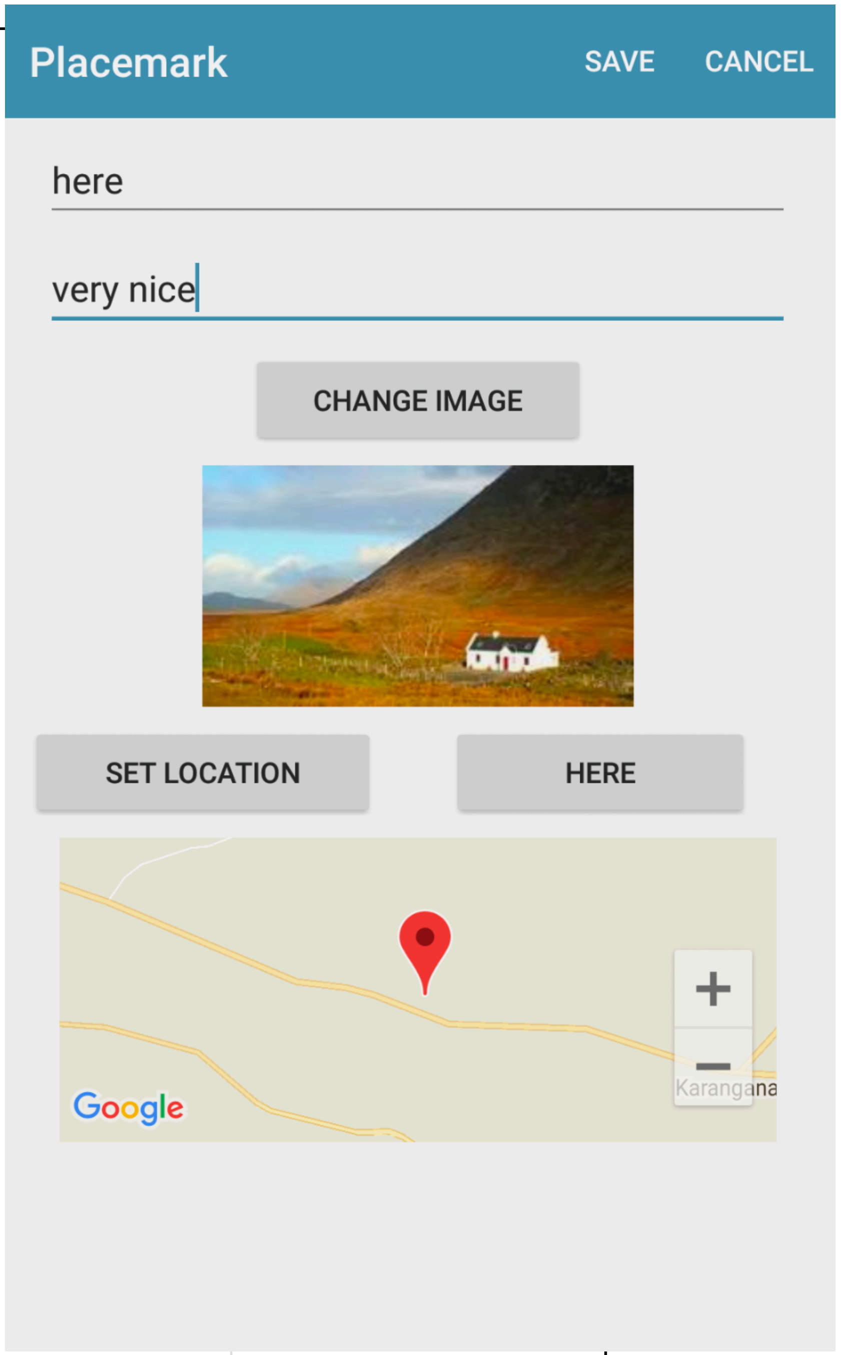
PlacemarkActivity



```
class PlacemarkActivity : AppCompatActivity(), AnkoLogger {

  var placemark = PlacemarkModel()
  lateinit var app: MainApp
  lateinit var map: GoogleMap
  var edit = false
  val IMAGE_REQUEST = 1
  val LOCATION_REQUEST = 2
  val defaultLocation = Location( lat: 52.245696,  lng: -7.139102,  zoom: 15f)

  override fun onCreate(savedInstanceState: Bundle?) {...}

  fun configureMap() {...}

  override fun onCreateOptionsMenu(menu: Menu?): Boolean {...}

  fun save() {...}

  override fun onOptionsItemSelected(item: MenuItem?): Boolean {...}

  override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {...}

  override fun onDestroy() {...}

  override fun onLowMemory() {...}

  override fun onPause() {...}

  override fun onResume() {...}

  override fun onSaveInstanceState(outState: Bundle?) {...}
}
```




```
class PlacemarkActivity : AppCompatActivity(), AnkoLogger {

  var placemark = PlacemarkModel()
  lateinit var app: MainApp
  lateinit var map: GoogleMap
  var edit = false
  val IMAGE_REQUEST = 1
  val LOCATION_REQUEST = 2
  val defaultLocation = Location( lat: 52.245696,  lng: -7.139102,  zoom: 15f)

  private lateinit var locationService: FusedLocationProviderClient

  override fun onCreate(savedInstanceState: Bundle?) {...}

  fun configureMap() {...}

  override fun onCreateOptionsMenu(menu: Menu?): Boolean {...}

  fun save() {...}

  override fun onOptionsItemSelected(item: MenuItem?): Boolean {...}

  override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {...}

  @SuppressLint( ...value: "MissingPermission")
  fun setCurrentLocation() {...}

  override fun onStart() {...}

  @SuppressLint( ...value: "MissingPermission")
  override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<String>, grantResults: IntArray) {...}

  override fun onDestroy() {...}

  override fun onLowMemory() {...}

  override fun onPause() {...}

  override fun onResume() {...}

  override fun onSaveInstanceState(outState: Bundle?) {...}
}
```



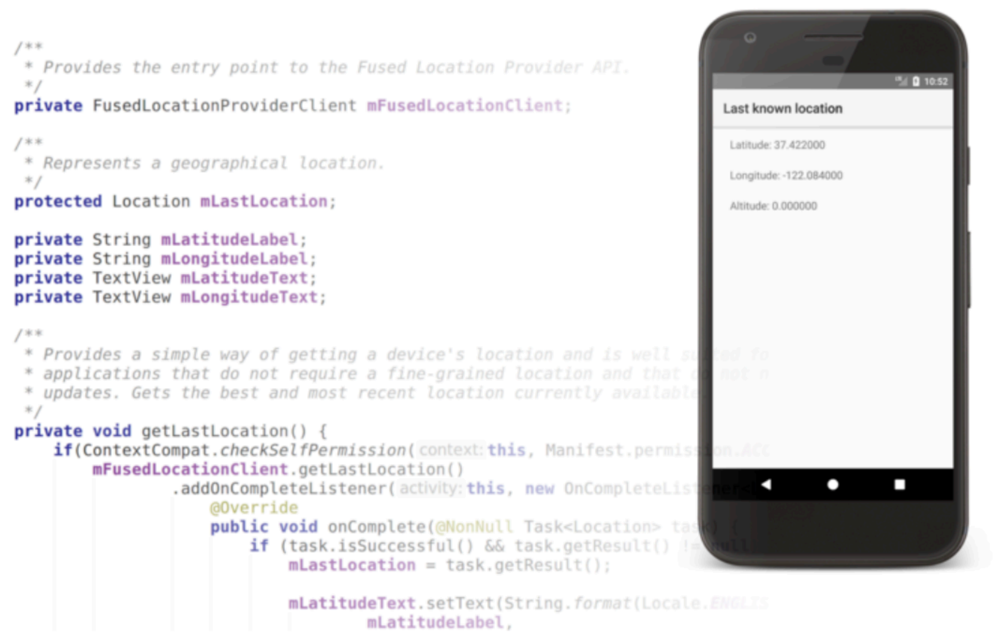
Fused Location Provider API   Search   ALL PRODUCTS

Simple, battery-efficient location API for Android

Apps can take advantage of the signals provided by multiple sensors in the device to determine device location. However, choosing the right combination of signals for a specific task in different conditions is not simple. Finding a solution that is also battery-efficient is even more complicated.

The fused location provider is a location API in Google Play services that intelligently combines different signals to provide the location information that your app needs.

The fused location provider manages the underlying location technologies, such as GPS and Wi-Fi, and provides a simple API that you can use to specify the required quality of service. For example, you can request the most accurate data available, or the best accuracy possible with no additional power consumption.



```
/**
 * Provides the entry point to the Fused Location Provider API.
 */
private FusedLocationProviderClient mFusedLocationClient;

/**
 * Represents a geographical location.
 */
protected Location mLastLocation;

private String mLatitudeLabel;
private String mLongitudeLabel;
private TextView mLatitudeText;
private TextView mLongitudeText;

/**
 * Provides a simple way of getting a device's location and is well su
 * applications that do not require a fine-grained location and that
 * updates. Gets the best and most recent location currently availabl
 */
private void getLastLocation() {
    if(ContextCompat.checkSelfPermission( context: this, Manifest.permis
        mFusedLocationClient.getLastLocation()
                .addOnCompleteListener( activity: this, new OnCompleteLis
                    @Override
                    public void onComplete(@NonNull Task<Location> ta
                        if (task.isSuccessful() && task.getResult() !
                            mLastLocation = task.getResult();

                            mLatitudeText.setText(String.format(Locale.
                                    mLatitudeLabel,
```

Last known location

Latitude: 37.422000

Longitude: -122.084000

Altitude: 0.000000

https://developers.google.com/location-context/fused-location-provider/

Support for common location scenarios

Last Known Location

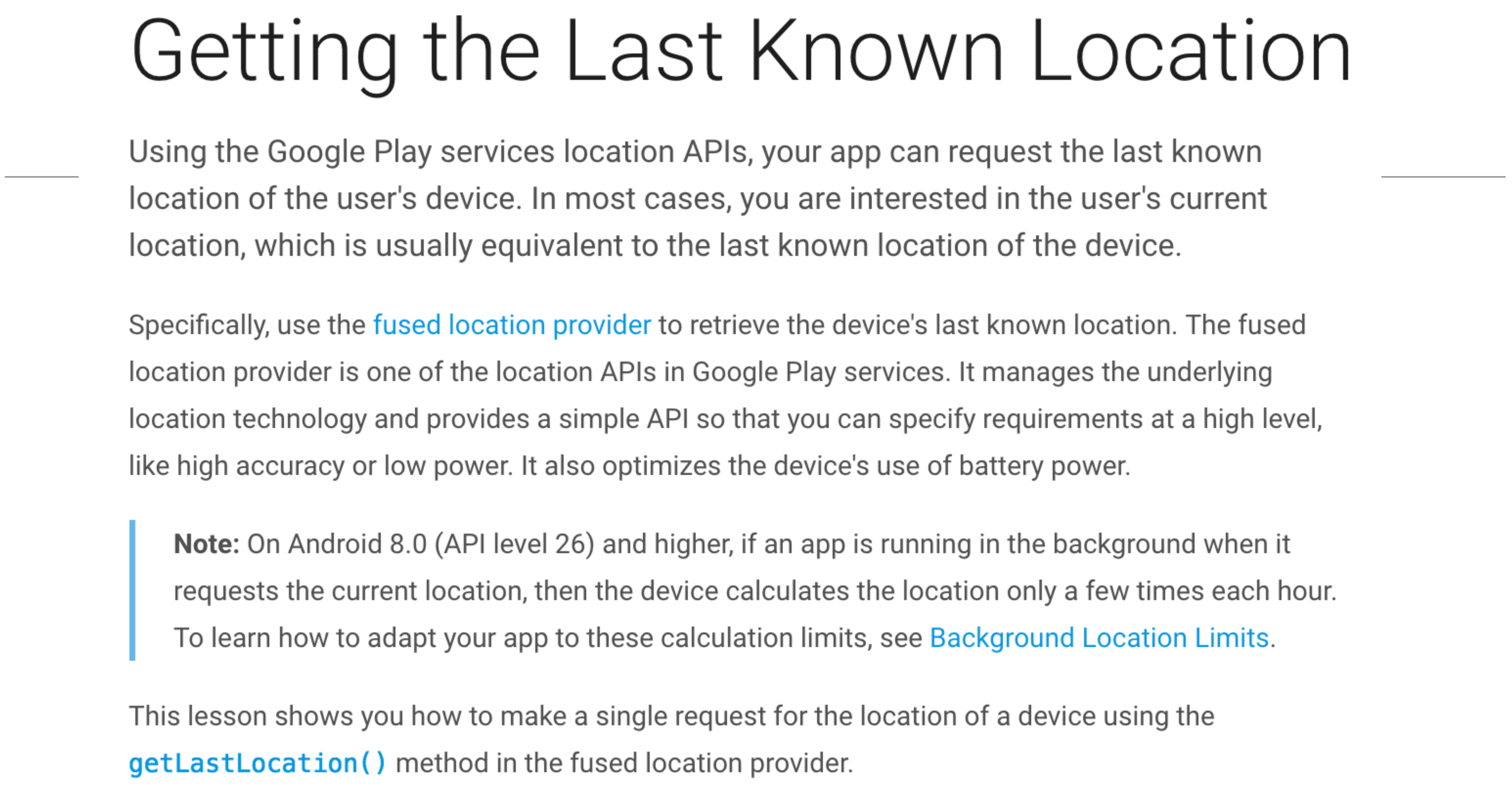
Using the fused location provider API, your app can request the last known location of the user's device. Getting the last known location is usually a good starting point for apps that require location information.

Location Settings

When requesting location information many different location sources, such as GPS and Wi-Fi, are used. Deciding which sources to use can be challenging, but the fused location provider API removes the guesswork by automatically changing the appropriate system settings. All your app must do is specify the desired level of service.

Location Updates

In addition to the last known location, the fused location provider API can deliver location updates to a callback in your app at specific intervals. You can specify the desired interval as a parameter of the quality of service. By using location updates, your app can provide additional information such as direction and velocity.



Getting the Last Known Location

Using the Google Play services location APIs, your app can request the last known location of the user's device. In most cases, you are interested in the user's current location, which is usually equivalent to the last known location of the device.

Specifically, use the fused location provider to retrieve the device's last known location. The fused location provider is one of the location APIs in Google Play services. It manages the underlying location technology and provides a simple API so that you can specify requirements at a high level, like high accuracy or low power. It also optimizes the device's use of battery power.

> **Note:** On Android 8.0 (API level 26) and higher, if an app is running in the background when it requests the current location, then the device calculates the location only a few times each hour. To learn how to adapt your app to these calculation limits, see Background Location Limits.

This lesson shows you how to make a single request for the location of a device using the `getLastLocation()` method in the fused location provider.

https://developer.android.com/training/location/retrieve-current.html

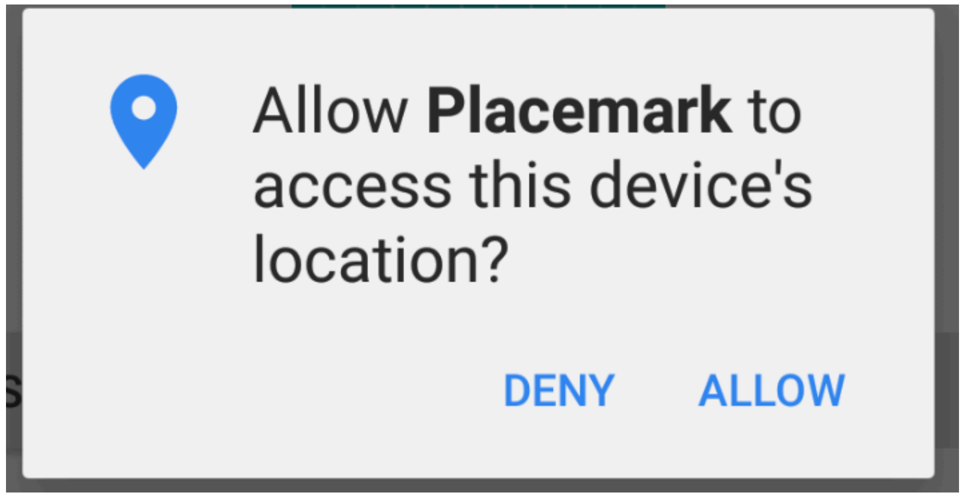
Location Permissions

If an app is to use the users location, there are 2 permission steps required

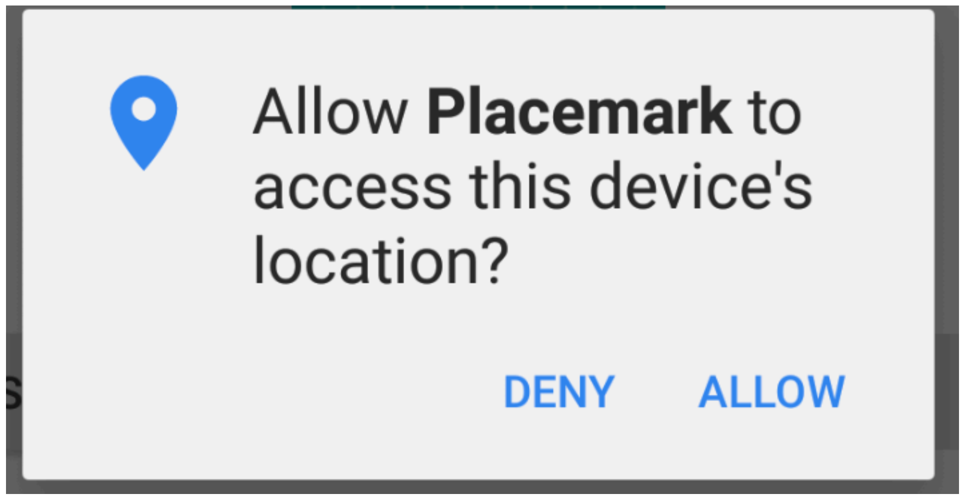
1: AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="org.wit.placemark">

  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

...
```

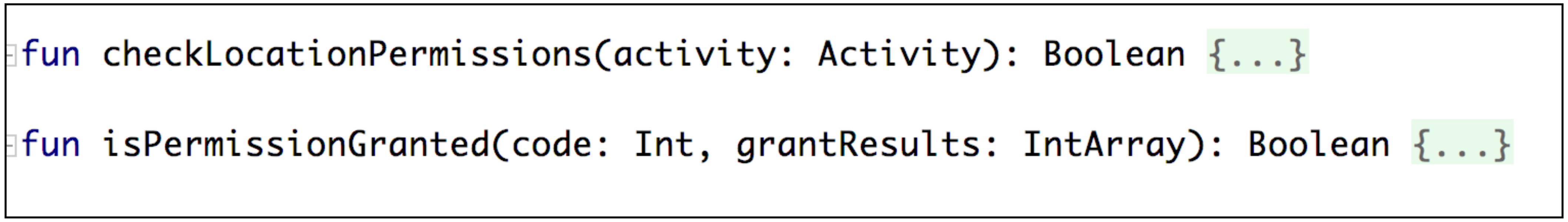
2: Dialog Directly with the user


Allow **Placemark** to access this device's location?

2: Dialog Directly with the user


Allow **Placemark** to access this device's location?

New Helper functions to support this interaction:

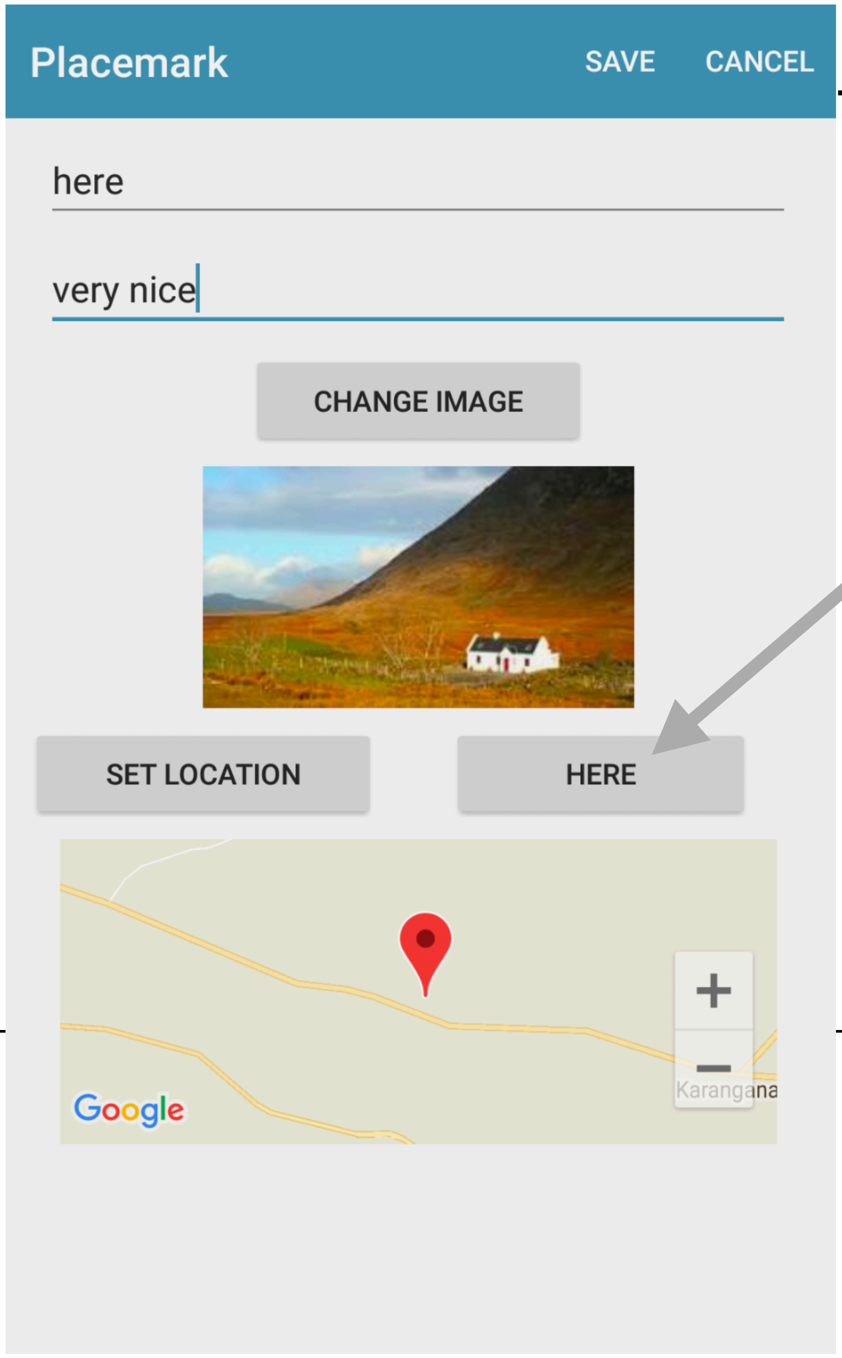

```
fun checkLocationPermissions(activity: Activity): Boolean {...}

fun isPermissionGranted(code: Int, grantResults: IntArray): Boolean {...}
```

LocationHelper

PlacemarkActivity

onStart - ask helper to see if permissions have been granted already.

If they have - enable **Here**

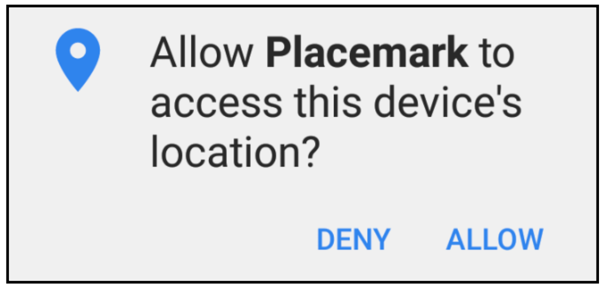


The **Here** butto only enabled if permissions granted

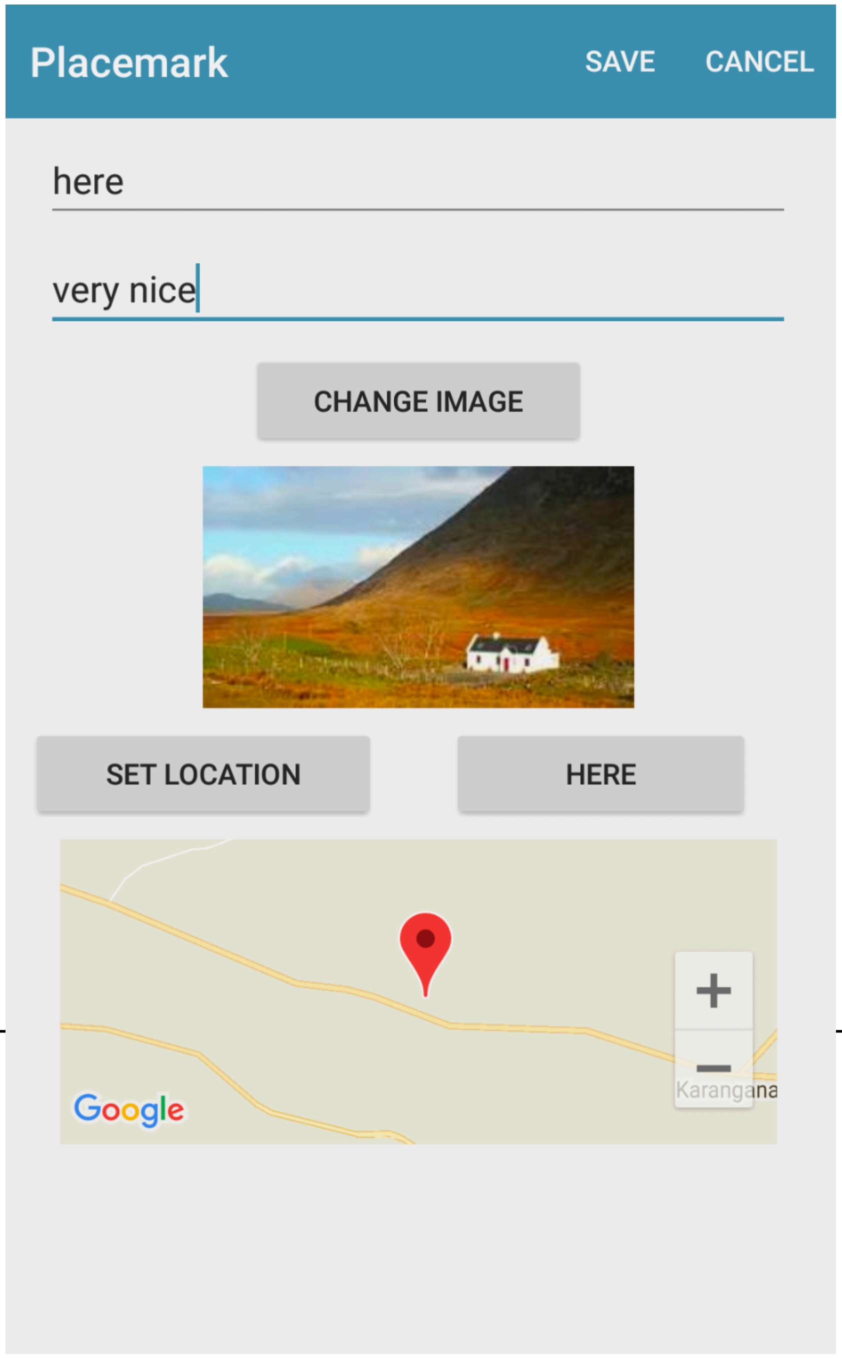
```
override fun onStart() {
  super.onStart()
  if (checkLocationPermissions( activity: this)) {
    btnHere.isEnabled = true
  }
}

@SuppressLint( ...value: "MissingPermission")
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<String>, grantResults: IntArray) {
  if (isPermissionGranted(requestCode, grantResults)) {
    btnHere.isEnabled = true
  }
}
```

PlacemarkActivity

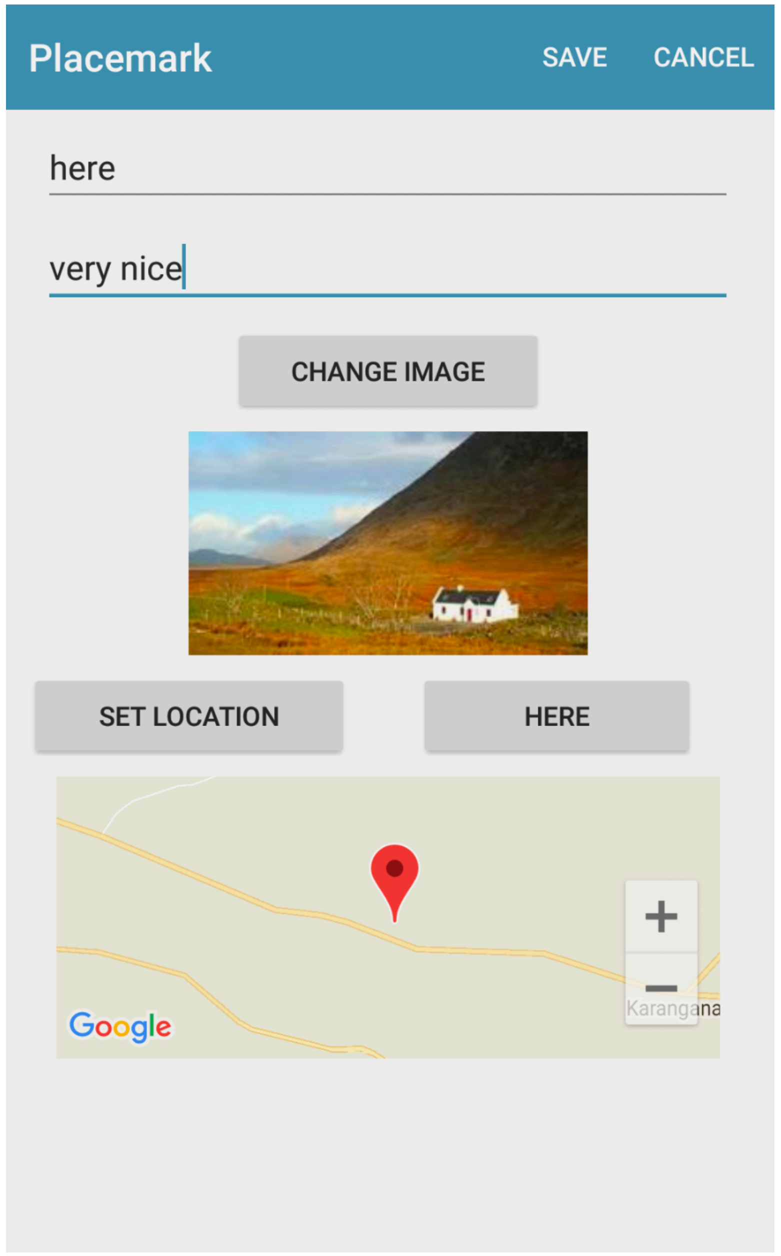
If not enabled already, dialog presented and return from dialog sent to *onRequestPermissionResult*



Allow **Placemark** to access this device's location?



```
override fun onStart() {
  super.onStart()
  if (checkLocationPermissions( activity: this)) {
    btnHere.isEnabled = true
  }
}

@SuppressLint( ...value: "MissingPermission")
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<String>, grantResults: IntArray) {
  if (isPermissionGranted(requestCode, grantResults)) {
    btnHere.isEnabled = true
  }
}
```



new field in PlacemarkActivity class

```
private lateinit var locationService: FusedLocationProviderClient
```

initialise in onCreate

```
locationService = LocationServices.getFusedLocationProviderClient(this)
```

Here event handler

```
btnHere.setOnClickListener {
  setCurrentLocation()
}
```

Recover last known location

```
@SuppressLint("MissingPermission")
fun setCurrentLocation() {
  locationService.lastLocation.addOnSuccessListener {
    defaultLocation.lat = it.latitude
    defaultLocation.lng = it.longitude
    placemark.lat = it.latitude
    placemark.lng = it.longitude
    configureMap()
  }
}
```

… and update map

Recover last known location

```
@SuppressLint("MissingPermission")
fun setCurrentLocation() {
  locationService.lastLocation.addOnSuccessListener {
    defaultLocation.lat = it.latitude
    defaultLocation.lng = it.longitude
    placemark.lat = it.latitude
    placemark.lng = it.longitude
    configureMap()
  }
}
```

Asynchronous request to location service

Location passed as default parameter **it** to callback

Recover lat/lng from **it**
Update map accordingly

… and update map