

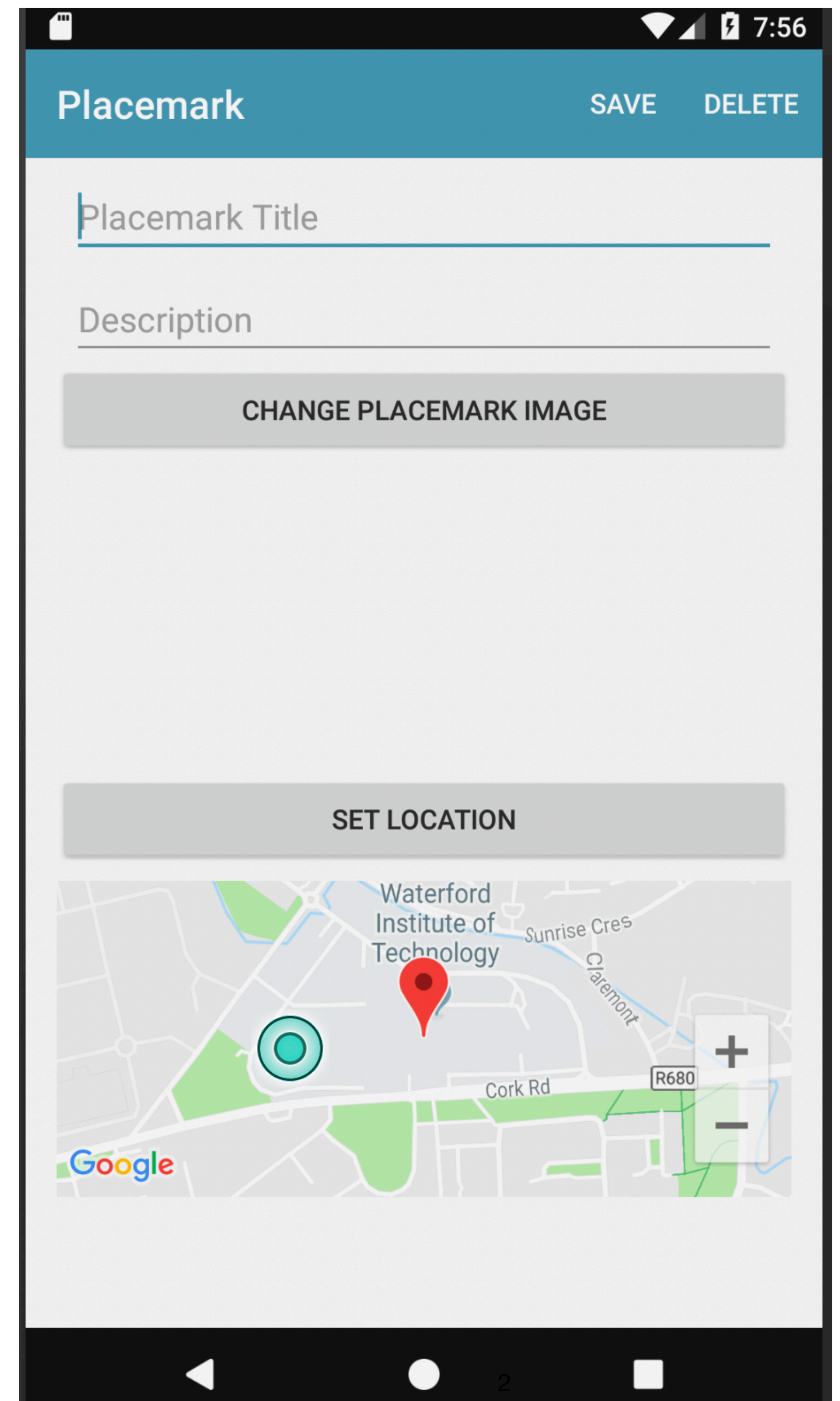
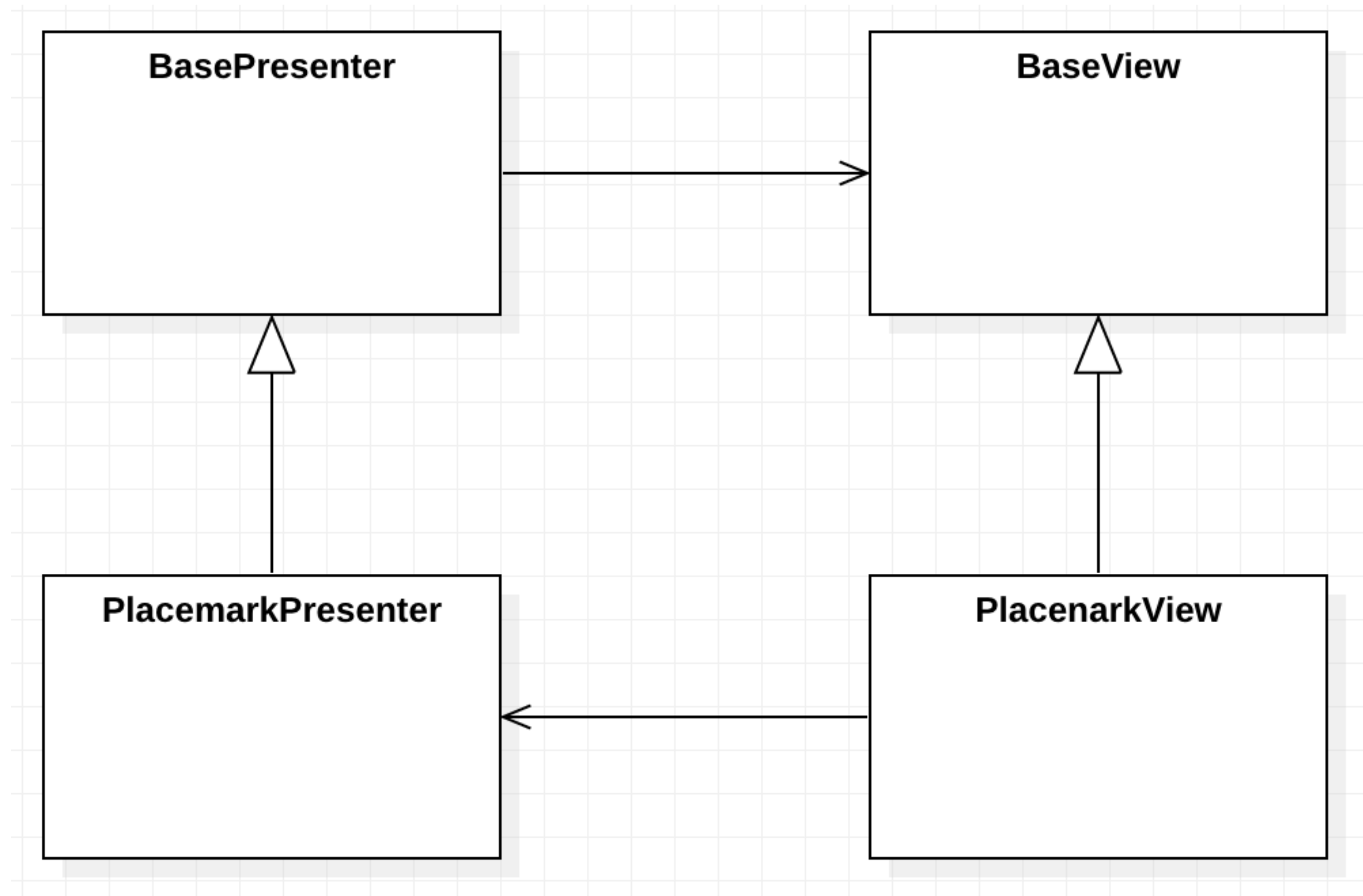
Current Location

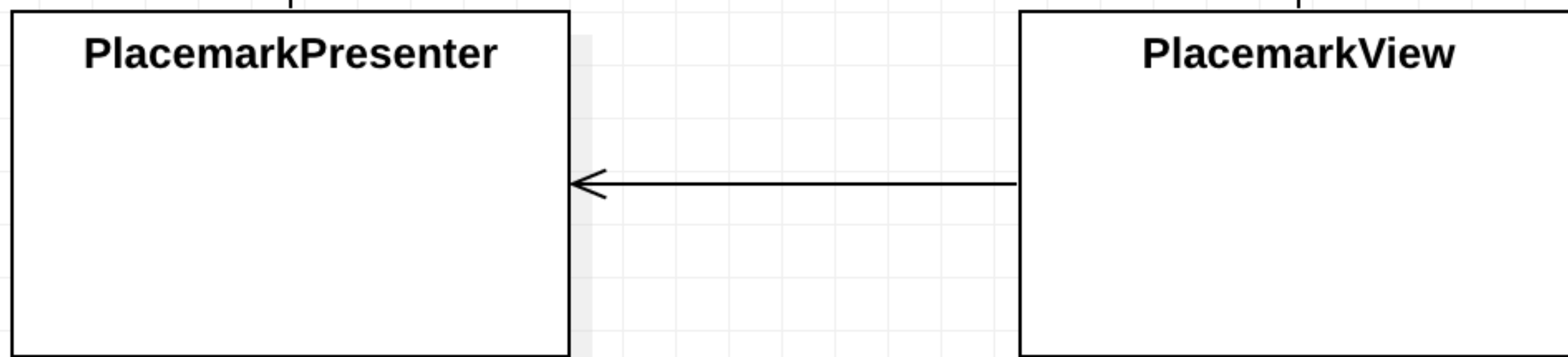
Current
Location



Using the Fused Location
Provider to recover the last
known location

PlacemarkView/Presenter





```
class PlacemarkPresenter(view: BaseView) : BasePresenter(view) {

    var map: GoogleMap? = null
    var placemark = PlacemarkModel()
    var defaultLocation = Location( lat: 52.245696, lng: -7.139102, zoom: 15f)
    var edit = false;

    init {...}

    fun doConfigureMap(m: GoogleMap) {...}

    fun locationUpdate(lat: Double, lng: Double) {...}

    fun doAddOrSave(title: String, description: String) {...}

    fun doCancel() {...}

    fun doDelete() {...}

    fun doSelectImage() {...}

    fun doSetLocation() {...}

    override fun doActivityResult(requestCode: Int, resultCode: Int, data: Intent) {...}
}
```

```
class PlacemarkView : BaseView(), AnkoLogger {

    lateinit var presenter: PlacemarkPresenter
    var placemark = PlacemarkModel()

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

    override fun showPlacemark(placemark: PlacemarkModel) {...}

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

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

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

    override fun onBackPressed() {...}

    override fun onDestroy() {...}

    override fun onLowMemory() {...}

    override fun onPause() {...}

    override fun onResume() {...}

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



```

class PlacemarkPresenter(view: BaseView) : BasePresenter(view) {

    var map: GoogleMap? = null
    var placemark = PlacemarkModel()
    var defaultLocation = Location(52.245696, -7.139102, 15f)
    var edit = false;

    init {
        if (view.intent.hasExtra("placemark_edit")) {
            edit = true
            placemark = view.intent.extras.getParcelable<PlacemarkModel>("placemark_
view.showPlacemark(placemark)
        } else {
            placemark.lat = defaultLocation.lat
            placemark.lng = defaultLocation.lng
        }
    }

    fun doConfigureMap(m: GoogleMap) {
        map = m
        locationUpdate(placemark.lat, placemark.lng)
    }

    fun locationUpdate(lat: Double, lng: Double) {
        placemark.lat = lat
        placemark.lng = lng
        placemark.zoom = 15f
        map?.clear()
        map?.uiSettings?.setZoomControlsEnabled(true)
        val options = MarkerOptions().title(placemark.title).position(LatLng(placemark.lat, placemark.lng))
        map?.addMarker(options)
        map?.moveCamera(CameraUpdateFactory.newLatLngZoom(LatLng(placemark.lat, placemark.lng), placemark.zoom))
        view?.showPlacemark(placemark)
    }
    ...
}

```

```

class PlacemarkView : BaseView(), AnkoLogger {

    lateinit var presenter: PlacemarkPresenter
    var placemark = PlacemarkModel()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_placemark)

        init(toolbarAdd)

        presenter = initPresenter (PlacemarkPresenter(this)) as PlacemarkPresenter

        mapView.onCreate(savedInstanceState);
        mapView.getMapAsync {
            presenter.doConfigureMap(it)
        }

        chooseImage.setOnClickListener { presenter.doSelectImage() }

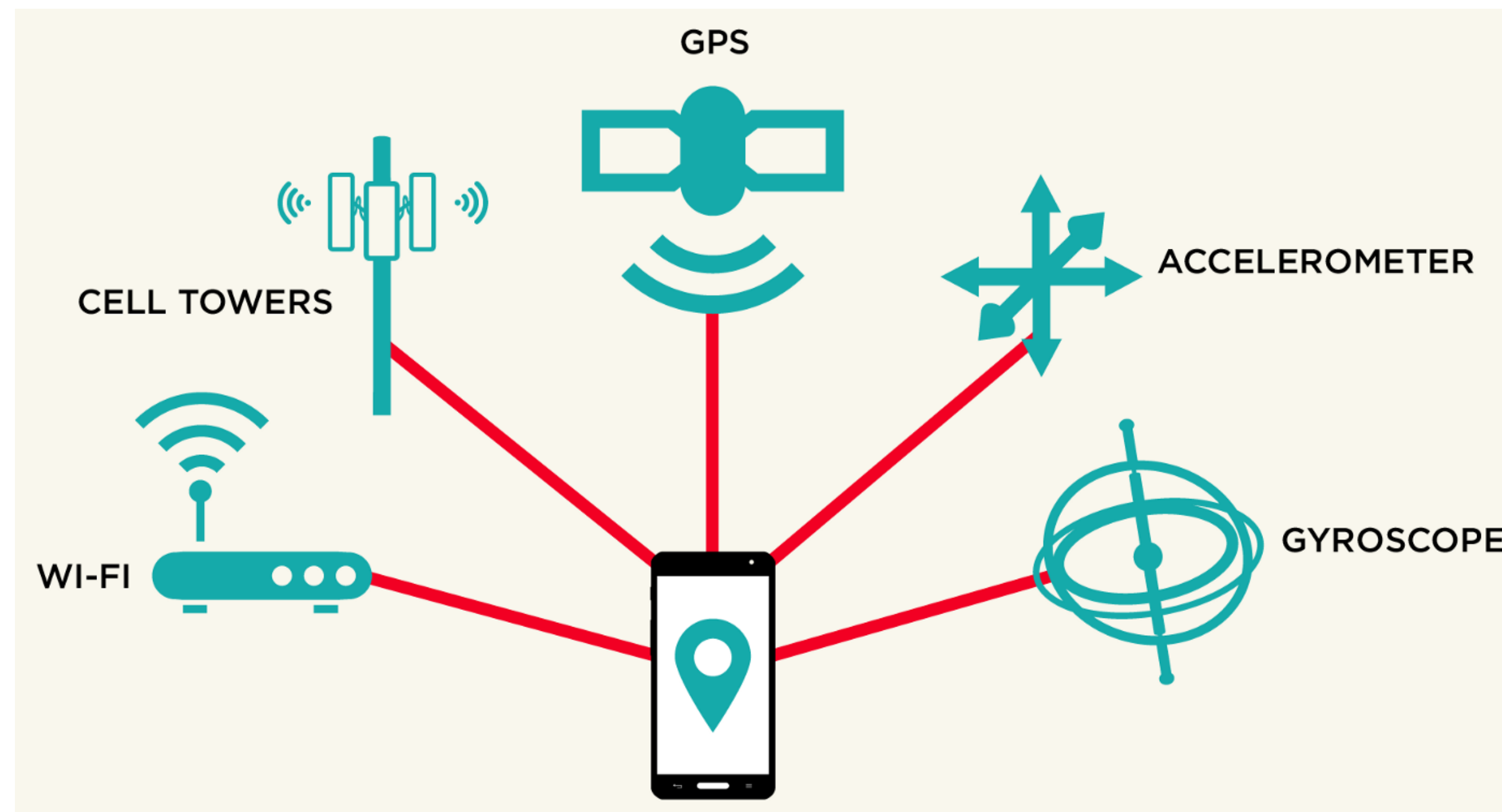
        placemarkLocation.setOnClickListener { presenter.doSetLocation() }
    }
    ...
}

```

build.gradle

```
implementation 'com.google.android.gms:play-services-location:17.0.0'
```

play-service-location Library



Location Permissions

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

1: AndroidManifest.xml

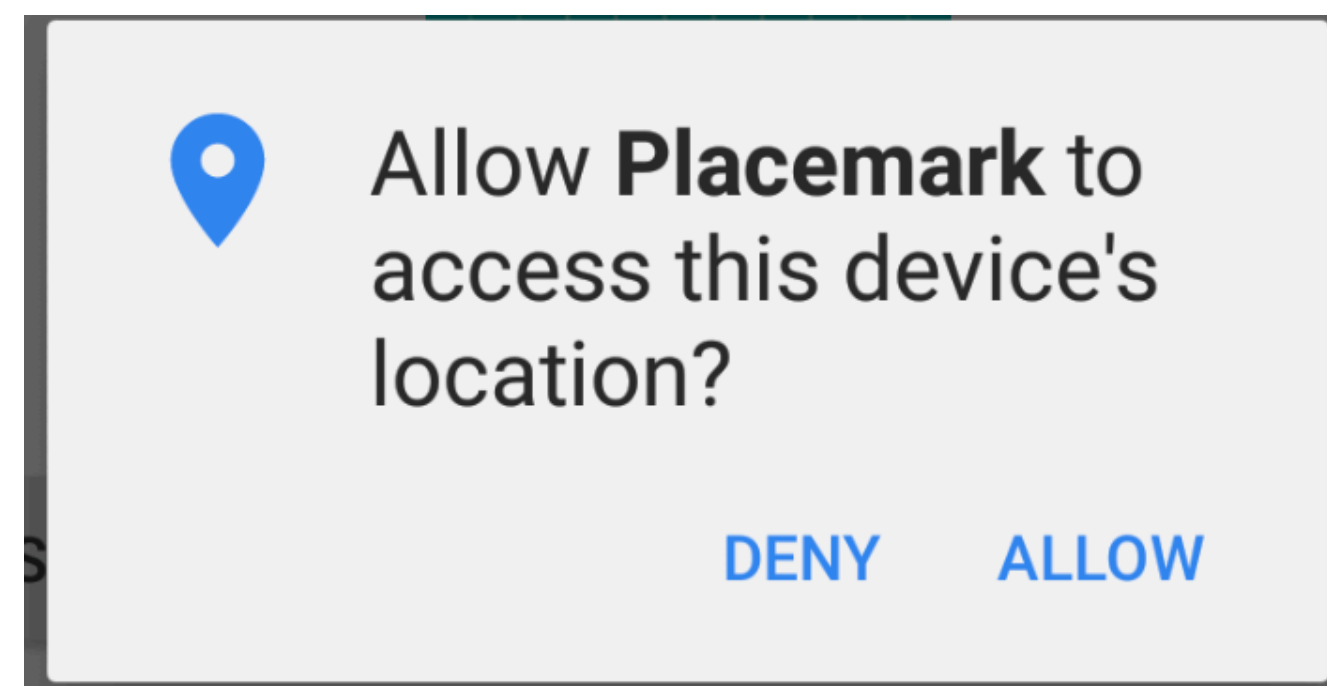
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.wit.placemark">

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

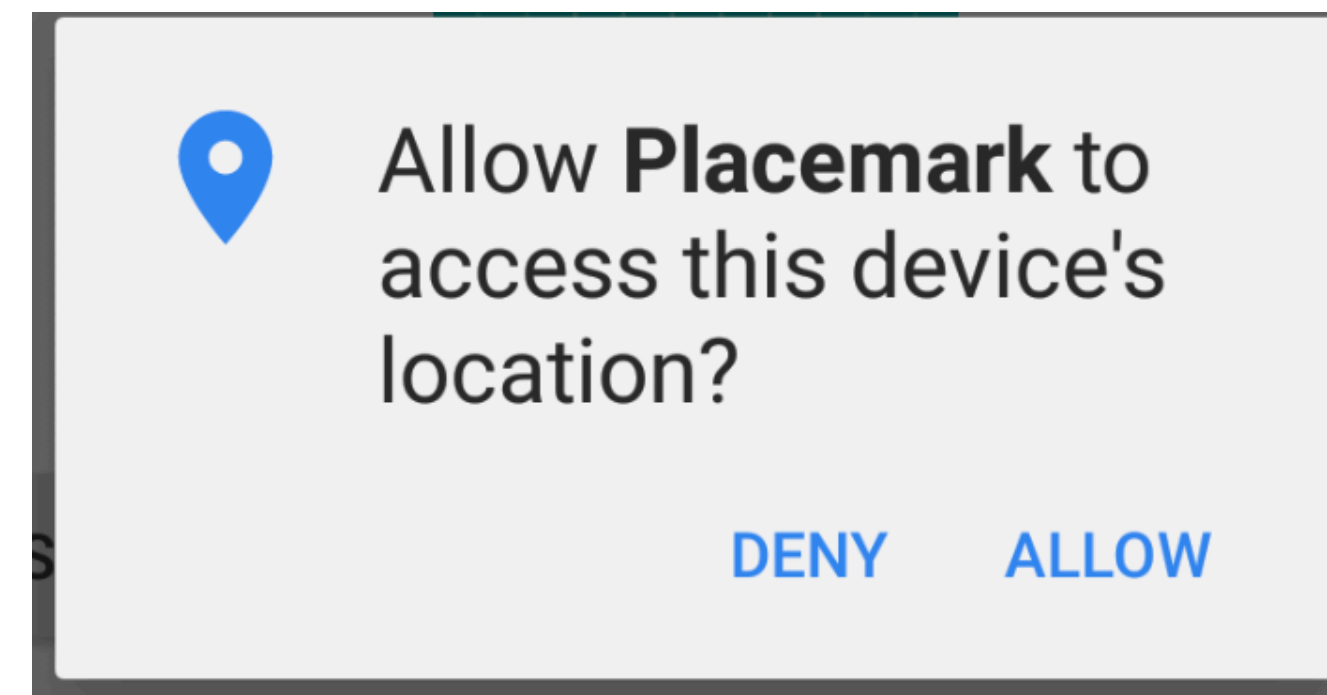
    ...

```

2: Dialog Directly with the user



2: Dialog Directly with the user



New Helper functions to support this interaction:

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

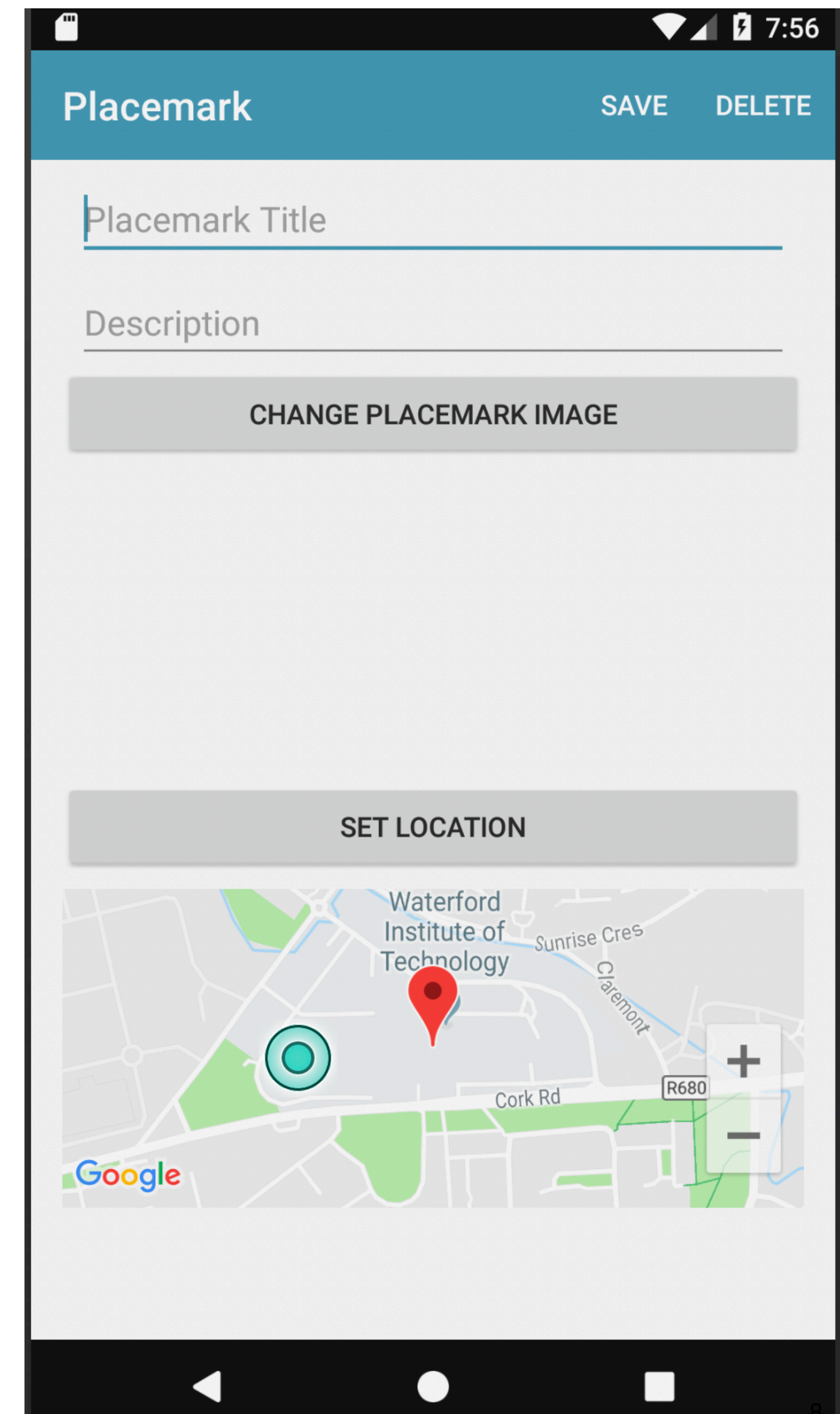
LocationHelper

PlacemarkPresenter - Acquire Permissions

```
class PlacemarkPresenter(view: BaseView) : BasePresenter(view) {
    ...

    init {
        if (view.intent.hasExtra("placemark_edit")) {
            edit = true
            placemark = view.intent.extras.getParcelable<PlacemarkModel>("placemark_edit")
            view.showPlacemark(placemark)
        } else {
            if (checkLocationPermissions(view)) {
                // get current location
            }
        }
    }

    override fun doRequestPermissionsResult(requestCode: Int, permissions: Array<String>,
                                             grantResults: IntArray) {
        if (isPermissionGranted(requestCode, grantResults)) {
            // get current Location
        } else {
            locationUpdate(defaultLocation.lat, defaultLocation.lng)
        }
    }
    ...
}
```



PlacemarkPresenter - Acquire Permissions

```
class PlacemarkPresenter(view: BaseView) : BasePresenter(view) {  
    ...  
  
    init {  
        if (view.intent.hasExtra("placemark_edit")) {  
            edit = true  
            placemark = view.intent.extras.getParcelable<PlacemarkModel>("placemark_edit")  
            view.showPlacemark(placemark)  
        } else {  
            if (checkLocationPermissions(view)) {  
                // get current location  
            }  
        }  
    }  
}  
  
override fun doRequestPermissionsResult(requestCode: Int, permissions: Array<String>,  
                                         grantResults: IntArray) {  
    if (isPermissionGranted(requestCode, grantResults)) {  
        // get current Location  
    } else {  
        locationUpdate(defaultLocation.lat, defaultLocation.lng)  
    }  
}  
...  
}
```



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

DENY

ALLOW

PlacemarkPresenter - Acquire Permissions

```
class PlacemarkPresenter(view: BaseView) : BasePresenter(view) {
    ...

    init {
        if (view.intent.hasExtra("placemark_edit")) {
            edit = true
            placemark = view.intent.extras.getParcelable<PlacemarkModel>("placemark_edit")
            view.showPlacemark(placemark)
        } else {
            if (checkLocationPermissions(view)) {
                // get current location
            }
        }
    }

    override fun doRequestPermissionsResult(requestCode: Int, permissions: Array<String>,
                                             grantResults: IntArray) {
        if (isPermissionGranted(requestCode, grantResults)) {
            // get current Location
        } else {
            locationUpdate(defaultLocation.lat, defaultLocation.lng)
        }
    }
    ...
}
```



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

DENY

ALLOW

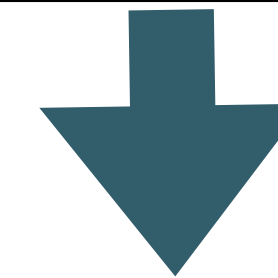
Acquire current Location
(permissions previously
granted)

Acquire current Location
(permissions granted just
now)

Permission Denied, show
default location

Fused Location Client

Acquire a reference to
the Fused location
provider



```
class PlacemarkPresenter(view: BaseView) : BasePresenter(view) {
```

```
    var locationService: FusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(view)
```

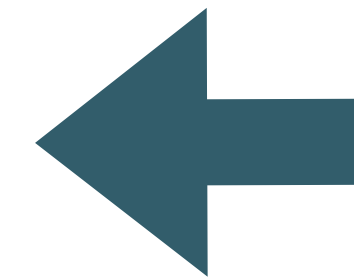
```
    ...
```

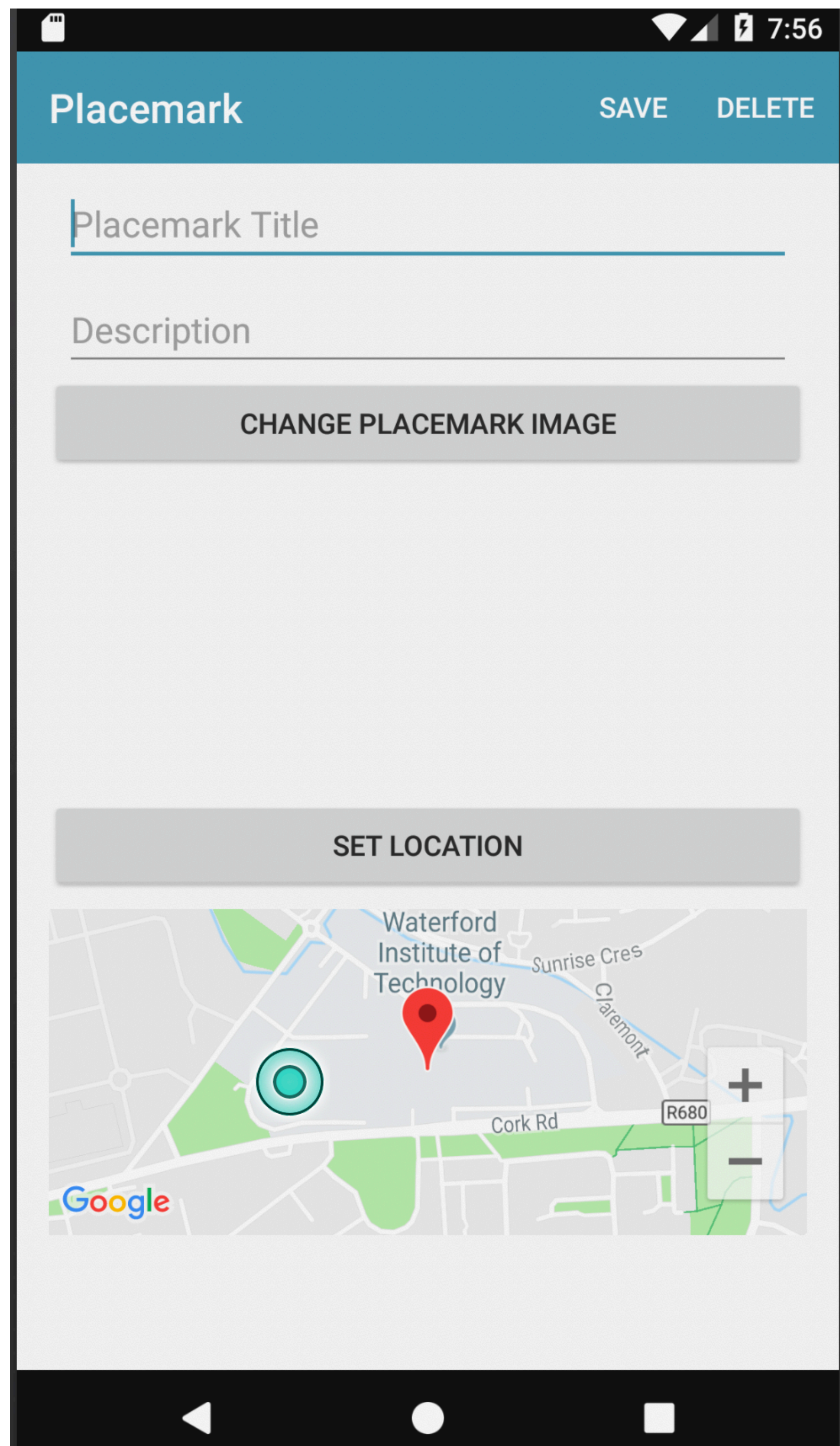
```
    @SuppressWarnings("MissingPermission")
    fun doSetCurrentLocation() {
        locationService.lastLocation.addOnSuccessListener {
            locationUpdate(Location(it.latitude, it.longitude))
        }
    }
}
```

```
    ...
```

```
}
```

New method to
request current
location





```
class PlacemarkPresenter(view: BaseView) : BasePresenter(view) {

    var map: GoogleMap? = null
    var placemark = PlacemarkModel()
    var defaultLocation = Location(52.245696, -7.139102, 15f)
    var edit = false;
    var locationService: FusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(view)

    init {
        if (view.intent.hasExtra("placemark_edit")) {
            edit = true
            placemark = view.intent.extras.getParcelable<PlacemarkModel>("placemark_edit")
            view.showPlacemark(placemark)
        } else {
            if (checkLocationPermissions(view)) {
                doSetCurrentLocation()
            }
        }
    }

    @SuppressWarnings("MissingPermission")
    fun doSetCurrentLocation() {
        locationService.lastLocation.addOnSuccessListener {
            locationUpdate(it.latitude, it.longitude)
        }
    }

    override fun doRequestPermissionsResult(requestCode: Int, permissions: Array<String>, grantResults: IntArray) {
        if (isPermissionGranted(requestCode, grantResults)) {
            doSetCurrentLocation()
        } else {
            locationUpdate(defaultLocation.lat, defaultLocation.lng)
        }
    }

    fun locationUpdate(lat: Double, lng: Double) {
        placemark.lat = lat
        placemark.lng = lng
        placemark.zoom = 15f
        map?.clear()
        map?.uiSettings?.setZoomControlsEnabled(true)
        val options = MarkerOptions().title(placemark.title).position(LatLng(placemark.lat, placemark.lng))
        map?.addMarker(options)
        map?.moveCamera(CameraUpdateFactory.newLatLngZoom(LatLng(placemark.lat, placemark.lng), placemark.zoom))
        view?.showPlacemark(placemark)
    }
    ...
}
```

Recover last known location

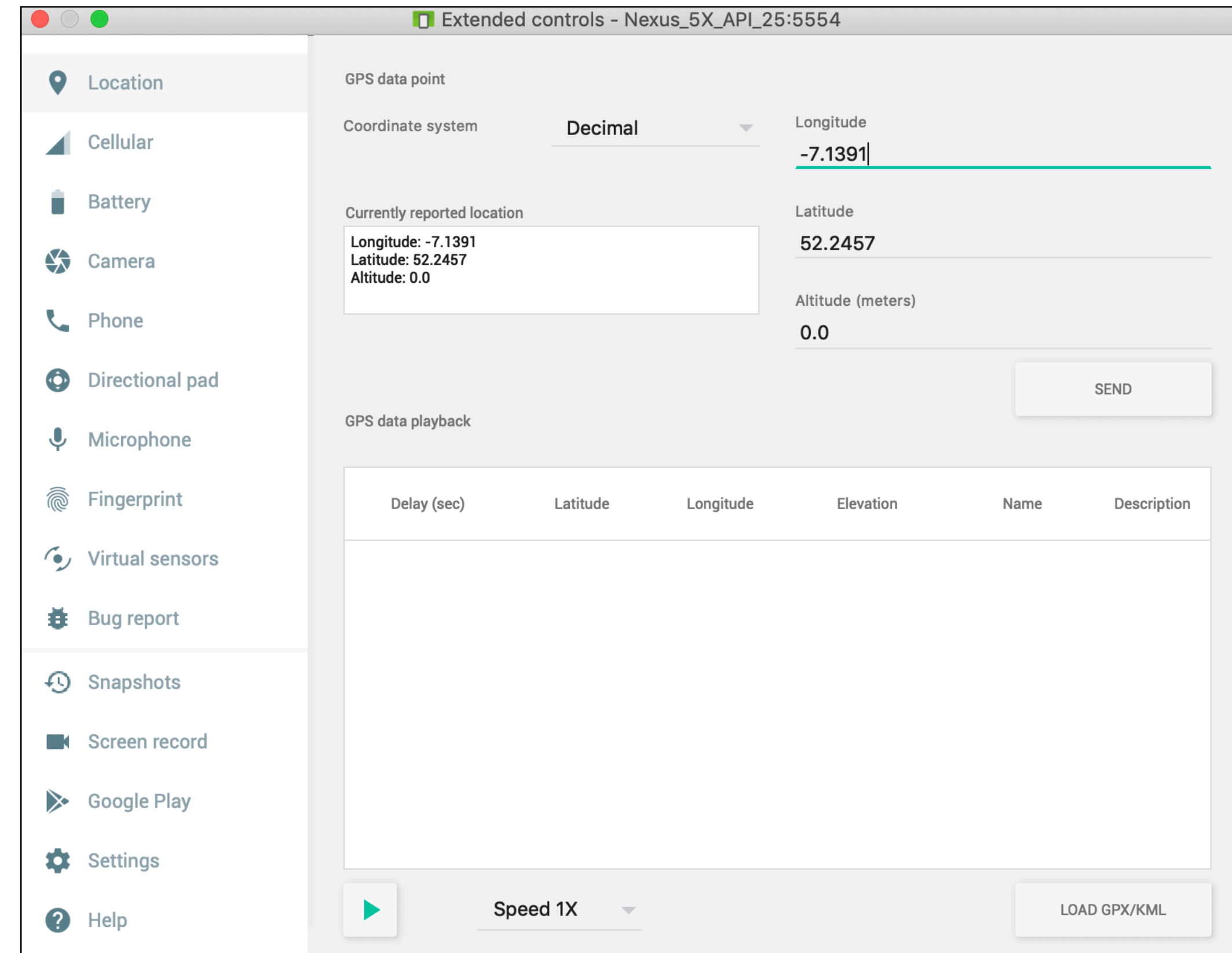
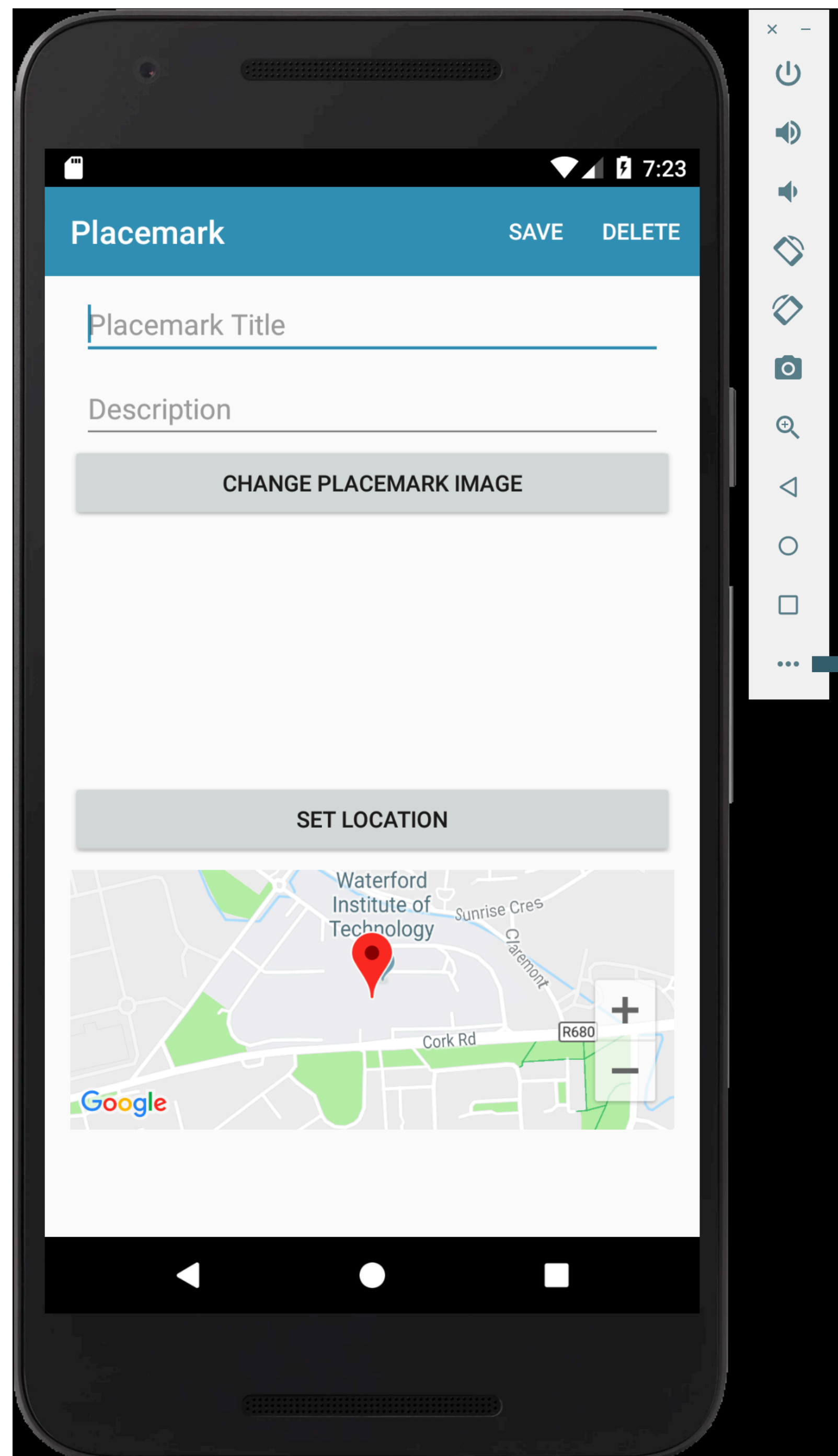
Suppress Permissions
warnings from compiler

Asynchronous request to
location service

```
@SuppressWarnings("MissingPermission")
fun doSetCurrentLocation() {
    locationService.lastLocation.addOnSuccessListener {
        locationUpdate(it.latitude, it.longitude)
    }
}
```

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

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



For '**lastLocation**' there is a noticeable lag
- up to several minutes - for the location
changes to percolate into the simulator