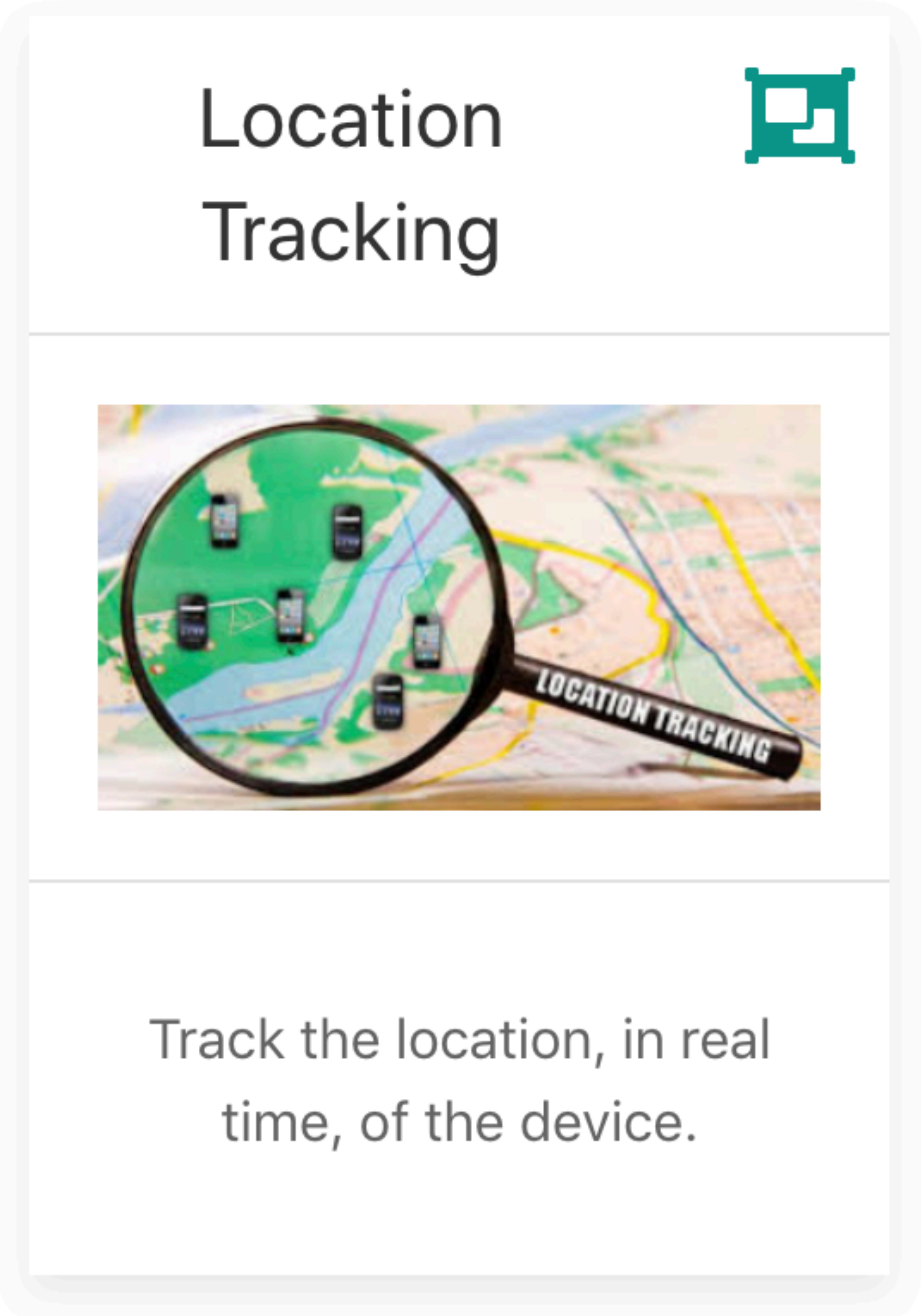
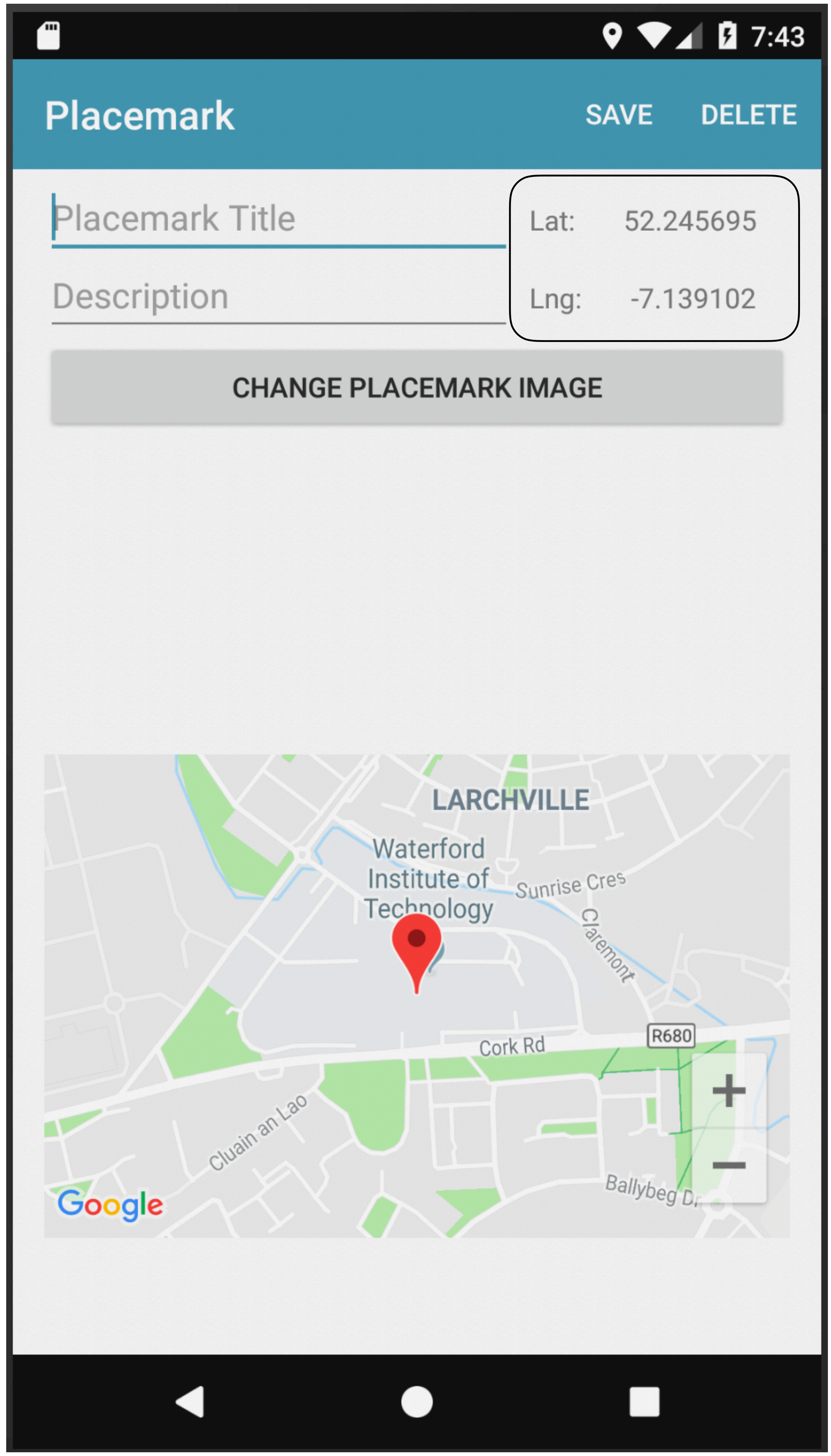
Location Tracking



Location Tracking

Track the location, in real time, of the device.

“Live” Current Lat/Lng



7:43

Placemark SAVE DELETE

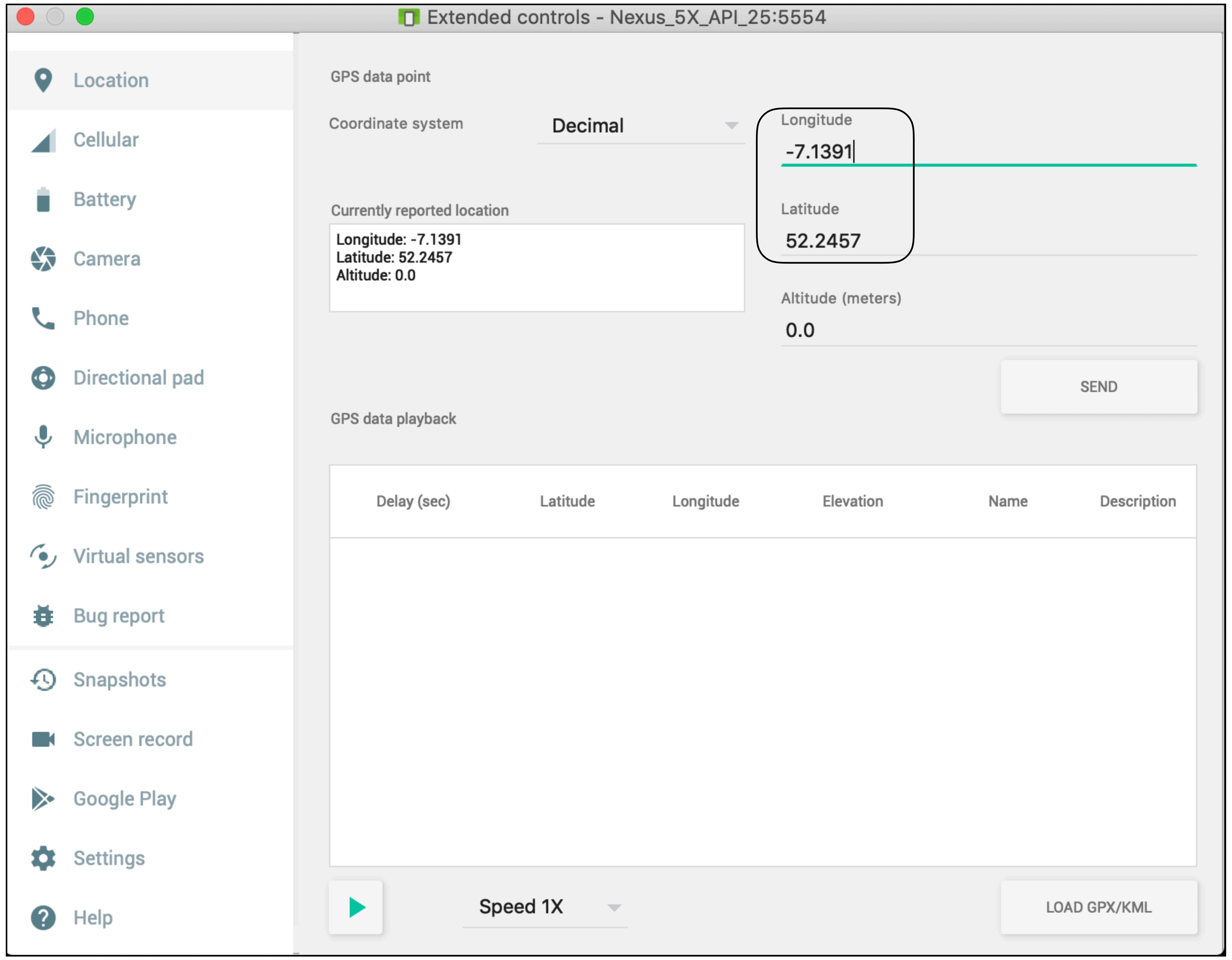
Placemark Title

Description

Lat: 52.245695

Lng: -7.139102

CHANGE PLACEMARK IMAGE



Extended controls - Nexus_5X_API_25:5554

GPS data point

Coordinate system: Decimal

Longitude: -7.1391

Latitude: 52.2457

Altitude (meters): 0.0

Currently reported location

Longitude: -7.1391
Latitude: 52.2457
Altitude: 0.0

SEND

GPS data playback

Delay (sec)	Latitude	Longitude	Elevation	Name	Description

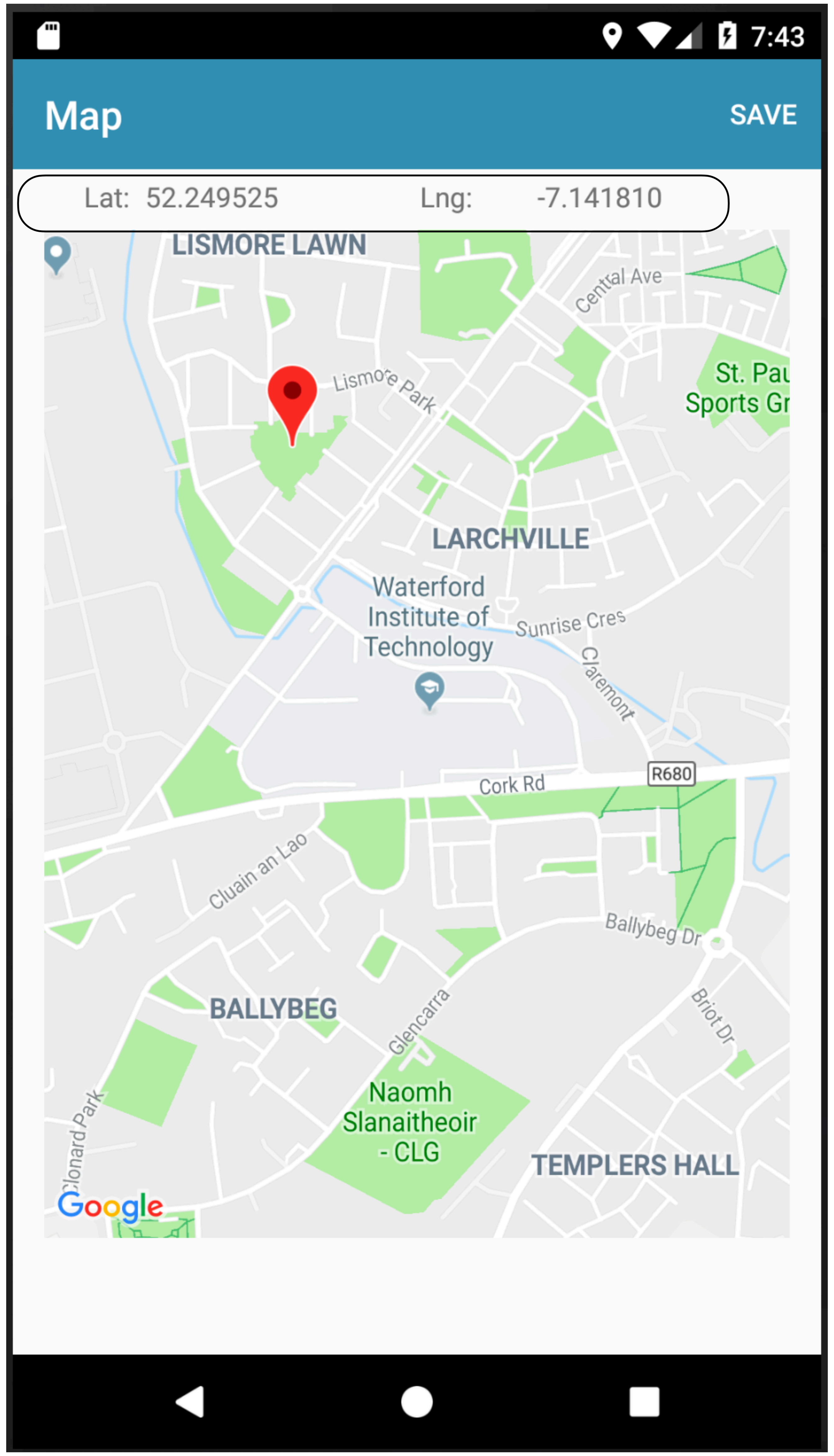
Speed 1X

LOAD GPX/KML

Changes in simulator to be reflected immediately in app

Display Current Marker Lat/Lng

Modify EditLocation to display lat/lng for marker as dragged around map

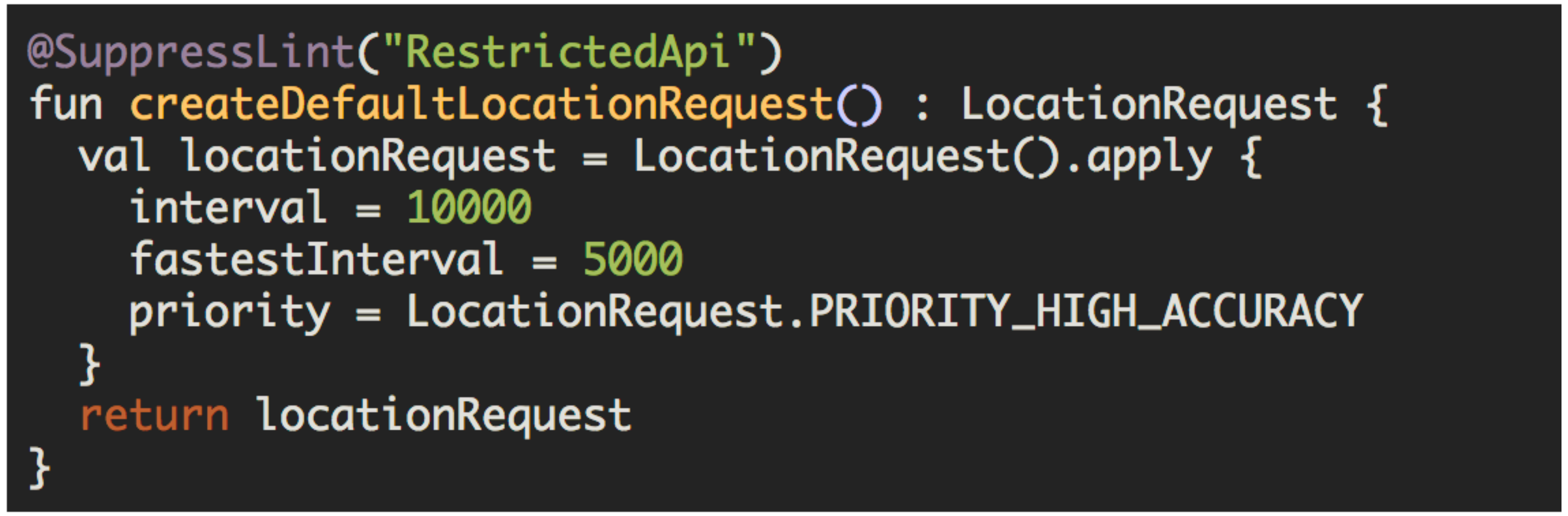


Revised Approach: Request Location Update Stream

To store parameters for requests to the fused location provider, create a LocationRequest.

The parameters determine the level of accuracy for location requests - including the update interval, fastest update interval, and priority

New Helper function to create a location request object

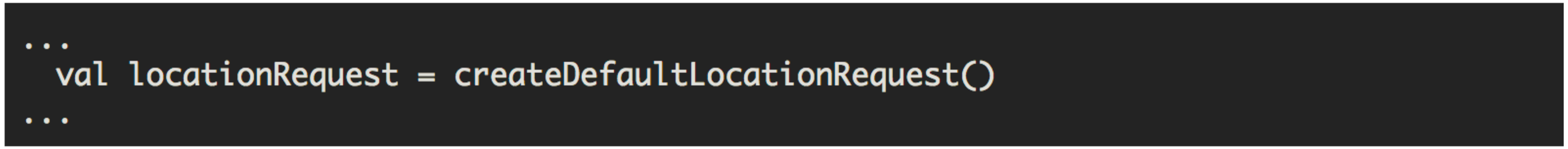


```
@SuppressLint("RestrictedApi")
fun createDefaultLocationRequest() : LocationRequest {
  val locationRequest = LocationRequest().apply {
    interval = 10000
    fastestInterval = 5000
    priority = LocationRequest.PRIORITY_HIGH_ACCURACY
  }
  return locationRequest
}
```

https://developer.android.com/training/location/change-location-settings.html#location-reques

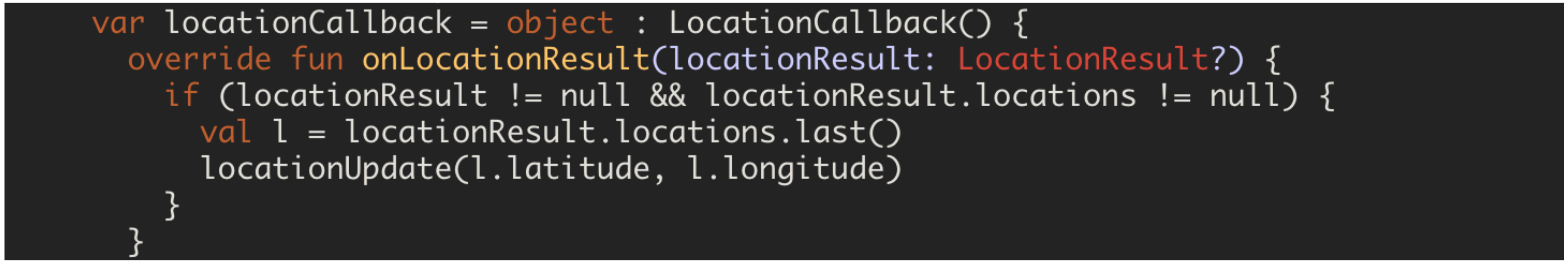
Revised PlacemarkPresenter

location request parameters in new attribute



```
...
  val locationRequest = createDefaultLocationRequest()
...
```

A call-back function which will be triggered by the fused location service



```
    var locationCallback = object : LocationCallback() {
      override fun onLocationResult(locationResult: LocationResult?) {
        if (locationResult != null && locationResult.locations != null) {
          val l = locationResult.locations.last()
          locationUpdate(l.latitude, l.longitude)
        }
      }
```

Revised PlacemarkPresenter

```
@SuppressLint("MissingPermission")
fun doResartLocationUpdates() {

  var locationCallback = object : LocationCallback() {
    override fun onLocationResult(locationResult: LocationResult?) {
      if (locationResult != null && locationResult.locations != null) {
        val l = locationResult.locations.last()
        locationUpdate(Location(l.latitude, l.longitude))
      }
    }
  }

  if (!edit) {
    locationService.requestLocationUpdates(locationRequest, locationCallback, null)
  }
}
```

Encapsulated callback

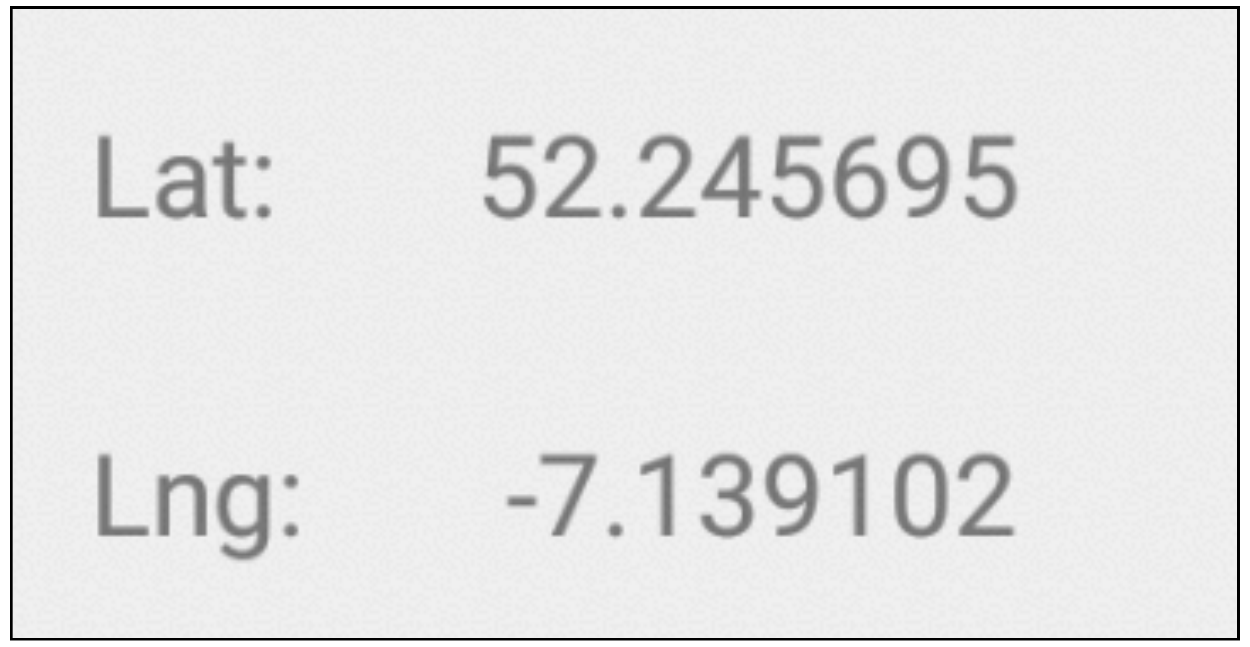
If in edit mode (i.e. this is a new placemark), request a location update stream

Revised PlacemarkPresenter

PlacemarkPresenter

```
if (locationResult != null && locationResult.locations != null) {
  val l = locationResult.locations.last()
  locationUpdate(Location(l.latitude, l.longitude))
}
```

```
fun locationUpdate(location: Location) {
  //…
  view?.showPlacemark(placemark)
}
```



Lat: 52.245695

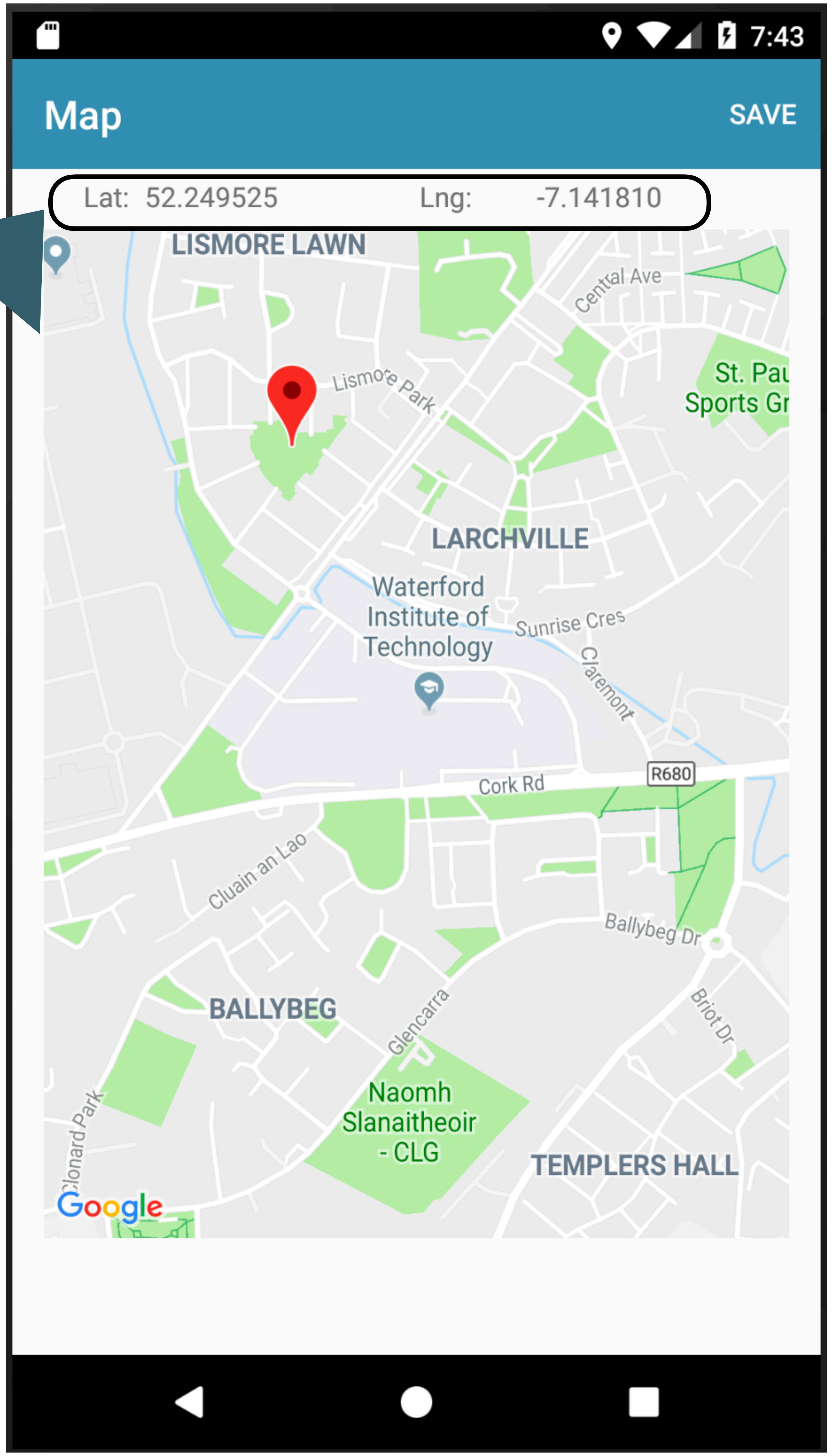
Lng: -7.139102

```
override fun showPlacemark(placemark: PlacemarkModel) {
  ...
  lat.setText("%.6f".format(placemark.location.lat))
  lng.setText("%.6f".format(placemark.location.lng))
}
```

PlacemarkView

EditLocationView

```
class EditLocationView : BaseView(), GoogleMap.OnMarkerDragListener {

  ...

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

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

  override fun onMarkerDrag(marker: Marker) {
    lat.setText("%.6f".format(marker.position.latitude))
    lng.setText("%.6f".format(marker.position.longitude))
  }

  ...
}
```



Map SAVE

Lat: 52.249525 Lng: -7.141810