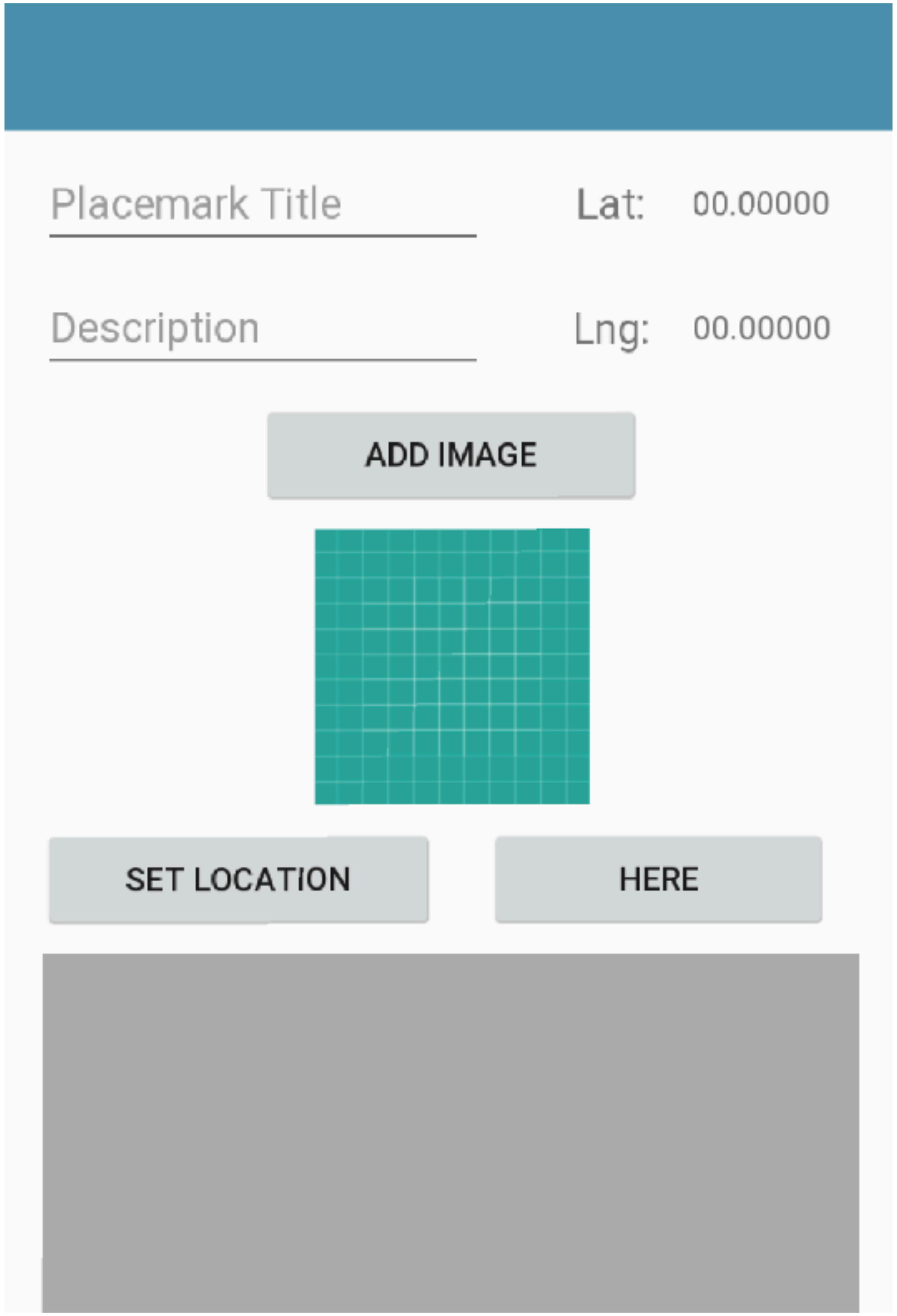
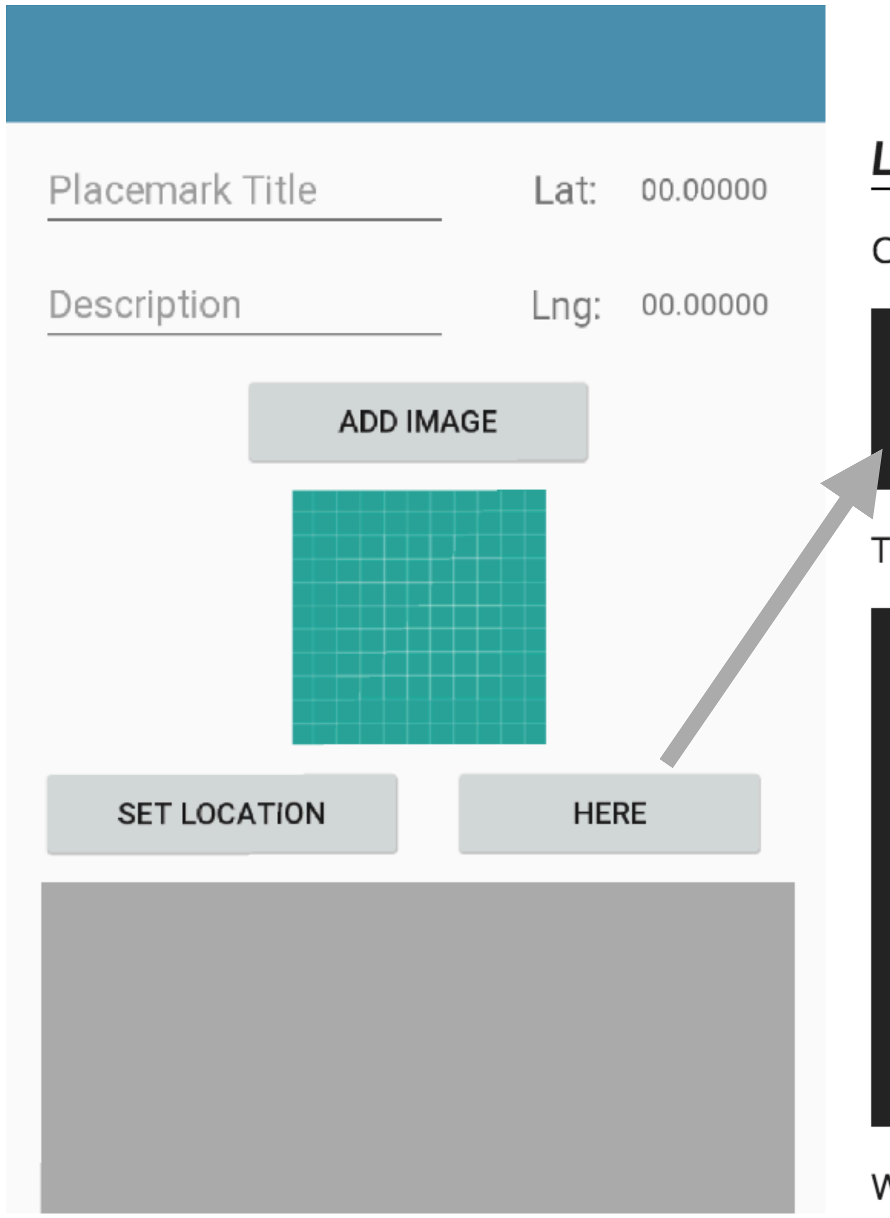
Location Tracking

Display Current Lat/Lng on Placemark Activity





Current Approach - Request location if button pressed

Location Requests

Currently we request the last known location when the user presses the `Here` button:

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

This calls this method:

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

Which performs an asynchronous request to get the latest known location. The configureMap method then ensures a marker is placed at this location:

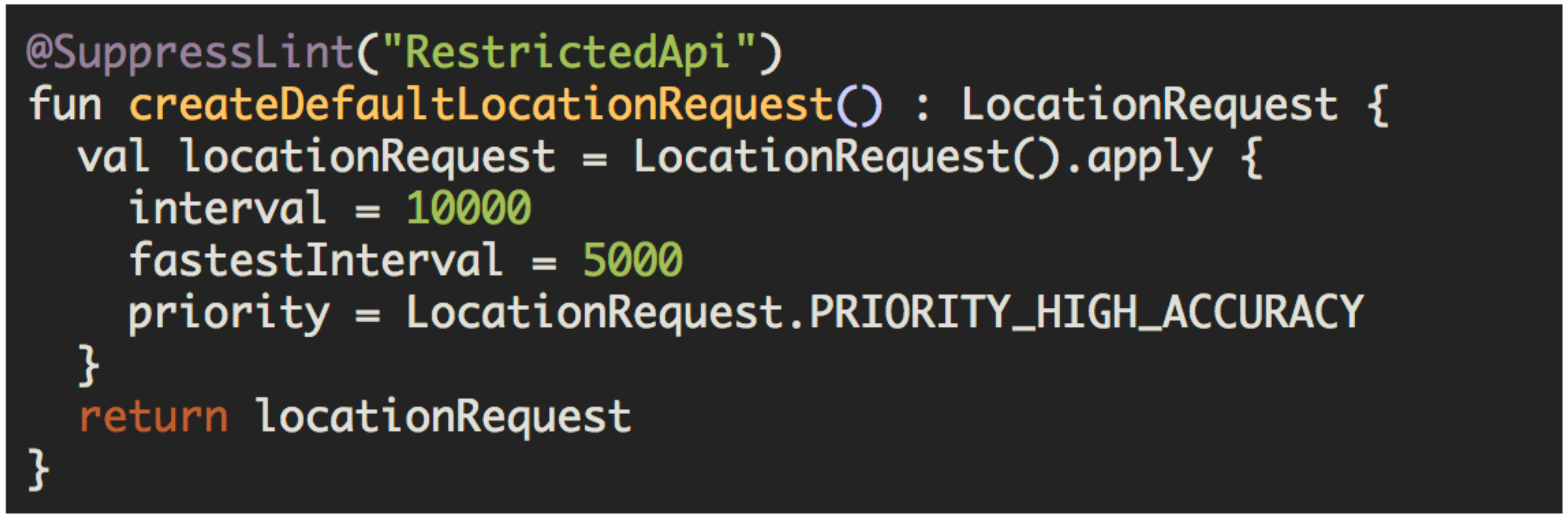
```
  fun configureMap() {
    map.uiSettings.setZoomControlsEnabled(true)
    val loc = LatLng(placemark.lat, placemark.lng)
    val options = MarkerOptions().title(placemark.title).position(loc)
    map.addMarker(options)
    map.moveCamera(CameraUpdateFactory.newLatLngZoom(loc, placemark.zoom))
  }
```

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 PlacemarkActivity

location request parameters in new field

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

Start the location update stream

```
  @SuppressLint("MissingPermission")
  private fun startLocationUpdates() {
    locationService.requestLocationUpdates(locationRequest, locationCallback, null)
  }
```

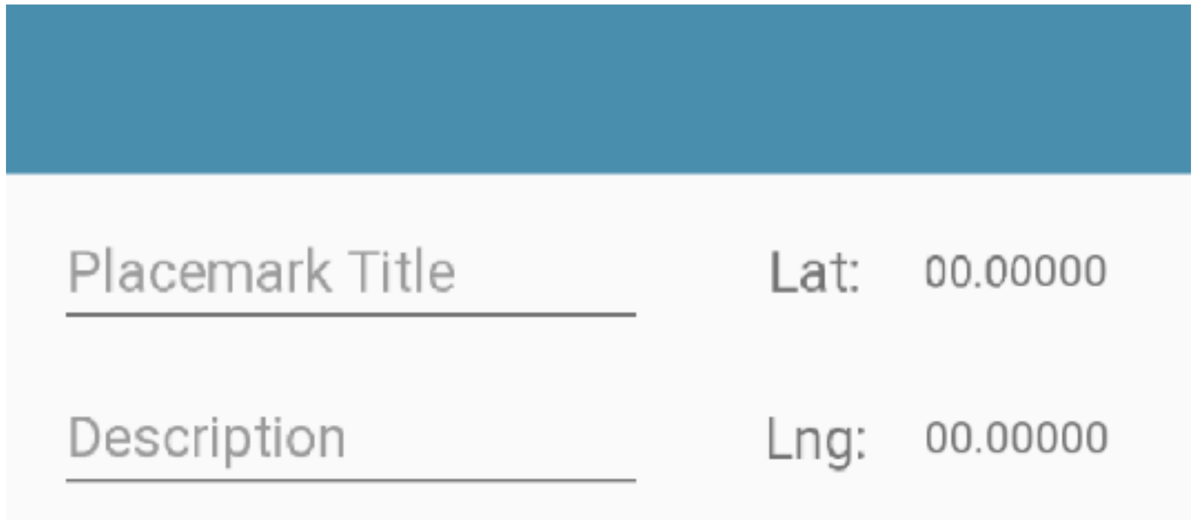
This must be resumed whenever the application is started

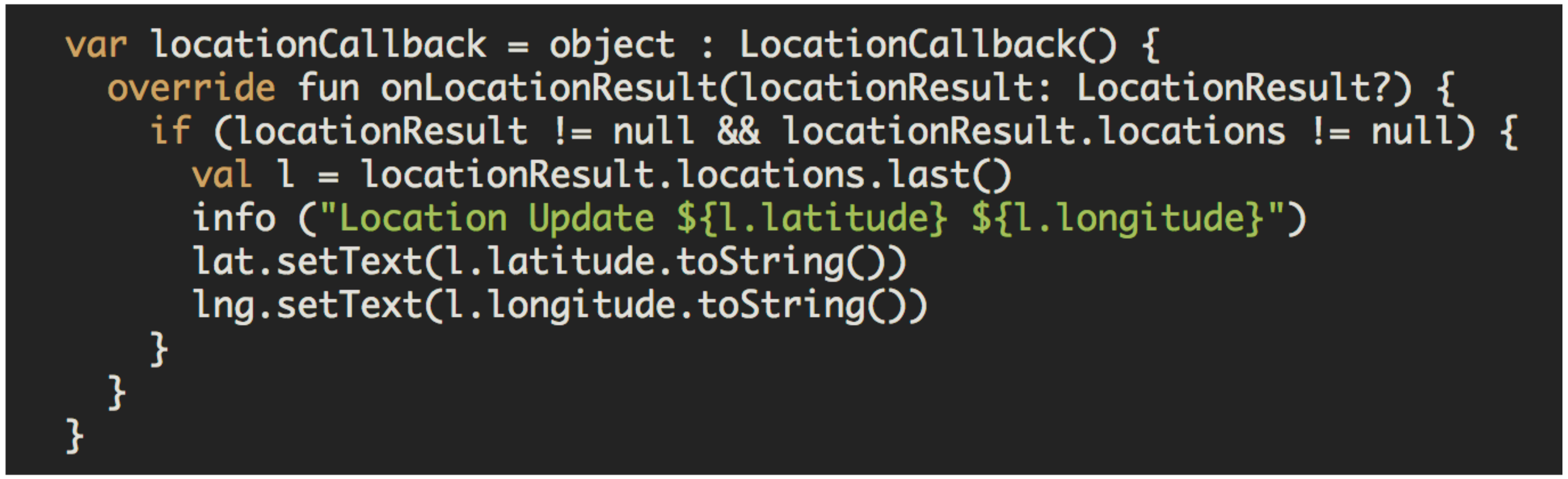
```
  override fun onResume() {
    super.onResume()
    mapView.onResume()
    startLocationUpdates()
  }
```

Location Stream

location callback - called at set interval with new location information

```
  var locationCallback = object : LocationCallback() {
    override fun onLocationResult(locationResult: LocationResult?) {
      info("Location Update")
    }
  }
```





```
var locationCallback = object : LocationCallback() {
  override fun onLocationResult(locationResult: LocationResult?) {
    if (locationResult != null && locationResult.locations != null) {
      val l = locationResult.locations.last()
      info ("Location Update ${l.latitude} ${l.longitude}")
      lat.setText(l.latitude.toString())
      lng.setText(l.longitude.toString())
    }
  }
}
```

In callback, retrieve lat/lng and send to layout