SETUP MOBILE APPLICATION ENVIRONMENT

Project Date	**September2022**
Team ID	**PNT2022TMID13786**
Project Name	**Containment Zone Alerting Application**

Geofence in Android App:

```
/*
* Copyright (C) 2014 The Android Open Source Project
 *
* Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
 *
* http://www.apache.org/licenses/LICENSE-2.0
 *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and * limitations under the License.
 */

package com.example.android.wearable.geofencing;

import static com.example.android.wearable.geofencing.Constants.ANDROID_BUILDING_ID;

import static com.example.android.wearable.geofencing.Constants.ANDROID_BUILDING_LATITUDE;
```

```
import static com.example.android.wearable.geofencing.Constants.ANDROID_BUILDING_LONGITUDE;

import static com.example.android.wearable.geofencing.Constants.ANDROID_BUILDING_RADIUS_METERS;

import static com.example.android.wearable.geofencing.Constants.CONNECTION_FAILURE_RESOLUTION_REQUEST;

import static com.example.android.wearable.geofencing.Constants.GEOFENCE_EXPIRATION_TIME;

 import static com.example.android.wearable.geofencing.Constants.TAG;   import static

com.example.android.wearable.geofencing.Constants.YERBA_BUENA_ID;

import static com.example.android.wearable.geofencing.Constants.YERBA_BUENA_LATITUDE;

import static com.example.android.wearable.geofencing.Constants.YERBA_BUENA_LONGITUDE;

import static com.example.android.wearable.geofencing.Constants.YERBA_BUENA_RADIUS_METERS;


 import android.app.Activity;

import android.app.PendingIntent;

import android.content.Intent;

import android.content.IntentSender;

import android.os.Bundle;   import

android.util.Log;   import

android.widget.Toast;


import com.google.android.gms.common.ConnectionResult;
```

```
 import com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks;
import com.google.android.gms.common.GooglePlayServicesUtil;   import
com.google.android.gms.common.api.GoogleApiClient;
 import com.google.android.gms.common.api.GoogleApiClient.OnConnectionFailedListener;
import com.google.android.gms.location.Geofence;   import
com.google.android.gms.location.LocationServices;

 import java.util.ArrayList;
import java.util.List;

 public class MainActivity extends Activity implements ConnectionCallbacks,
OnConnectionFailedListener {

    // Internal List of Geofence objects. In a real app, these might be provided by an API based on
// locations within the user's proximity.
    List<Geofence> mGeofenceList;

    // These will store hard-coded geofences in this sample app.
    private SimpleGeofence mAndroidBuildingGeofence;
private SimpleGeofence mYerbaBuenaGeofence;

    // Persistent storage for geofences.
    private SimpleGeofenceStore mGeofenceStorage;

    private LocationServices mLocationService;
    // Stores the PendingIntent used to request geofence monitoring.
private PendingIntent mGeofenceRequestIntent;
   private GoogleApiClient mApiClient;
```

```
    // Defines the allowable request types (in this example, we only add geofences).
    private enum REQUEST_TYPE {ADD}
private REQUEST_TYPE mRequestType;

    @Override       protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);
        // Rather than displayng this activity, simply display a toast indicating that the geofence
// service is being created. This should happen in less than a second.           if
(!isGooglePlayServicesAvailable()) {
            Log.e(TAG, "Google Play services unavailable.");
finish();              return;
        }

        mApiClient = new GoogleApiClient.Builder(this)
                .addApi(LocationServices.API)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this)
                .build();

        mApiClient.connect();

        // Instantiate a new geofence storage area.
        mGeofenceStorage = new SimpleGeofenceStore(this);
// Instantiate the current List of geofences.
       mGeofenceList = new ArrayList<Geofence>();
        createGeofences();
```

```
    }


    /**
     * In this sample, the geofences are predetermined and are hard-coded here. A real app might
* dynamically create geofences based on the user's location.
     */
    public void createGeofences() {
        // Create internal "flattened" objects containing the geofence data.
        mAndroidBuildingGeofence = new SimpleGeofence(
                ANDROID_BUILDING_ID,                // geofenceId.
                ANDROID_BUILDING_LATITUDE,
                ANDROID_BUILDING_LONGITUDE,
                ANDROID_BUILDING_RADIUS_METERS,
                GEOFENCE_EXPIRATION_TIME,
                Geofence.GEOFENCE_TRANSITION_ENTER | Geofence.GEOFENCE_TRANSITI
ON_EXIT
        );
        mYerbaBuenaGeofence = new SimpleGeofence(
                YERBA_BUENA_ID,                // geofenceId.
                YERBA_BUENA_LATITUDE,
                YERBA_BUENA_LONGITUDE,
                YERBA_BUENA_RADIUS_METERS,
                GEOFENCE_EXPIRATION_TIME,
                Geofence.GEOFENCE_TRANSITION_ENTER | Geofence.GEOFENCE_TRANSITI
ON_EXIT
        );


        // Store these flat versions in SharedPreferences and add them to the geofence list.
mGeofenceStorage.setGeofence(ANDROID_BUILDING_ID, mAndroidBuildingGeofence
);
```

```
        mGeofenceStorage.setGeofence(YERBA_BUENA_ID, mYerbaBuenaGeofence);
mGeofenceList.add(mAndroidBuildingGeofence.toGeofence());
mGeofenceList.add(mYerbaBuenaGeofence.toGeofence());
    }


    @Override       public void onConnectionFailed(ConnectionResult
connectionResult) {
        // If the error has a resolution, start a Google Play services activity to resolve it.
if (connectionResult.hasResolution()) {              try {
              connectionResult.startResolutionForResult(this,
                    CONNECTION_FAILURE_RESOLUTION_REQUEST);
           } catch (IntentSender.SendIntentException e) {
              Log.e(TAG, "Exception while resolving connection error.", e);
           }
        } else {
           int errorCode = connectionResult.getErrorCode();
           Log.e(TAG, "Connection to Google Play services failed with error code " + errorCode);
        }
    }

    /**
     * Once the connection is available, send a request to add the Geofences.
     */
    @Override
    public void onConnected(Bundle connectionHint) {
        // Get the PendingIntent for the geofenc…
```