Date	15 November 2022
Team ID	PNT2022TMID31668
Project Name	CONTAINMENT ZONE ALERTING

Geofence in Android App:

Code:

```
package com.example.finalgeofence;

import android.app.IntentService;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.TaskStackBuilder;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.text.TextUtils;
import android.util.Log;

import androidx.core.app.NotificationCompat;

import com.google.android.gms.location.Geofence;
import com.google.android.gms.location.GeofenceStatusCodes;
import com.google.android.gms.location.GeofencingEvent;

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


public class
```

```
GeofenceTrasitionService extends IntentService {

    private static final String TAG = GeofenceTrasitionService.class.getSimpleName();

    public static final int GEOFENCE_NOTIFICATION_ID = 0;

    public GeofenceTrasitionService() {
        super(TAG);
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        GeofencingEvent geofencingEvent = GeofencingEvent.fromIntent(intent);
        // Handling errors
        if ( geofencingEvent.hasError() ) {
            String errorMsg = getErrorString(geofencingEvent.getErrorCode() );
            Log.e( TAG, errorMsg );
            return;
        }

        int geoFenceTransition = geofencingEvent.getGeofenceTransition();
        // Check if the transition type is of interest
        if ( geoFenceTransition == Geofence.GEOFENCE_TRANSITION_ENTER ||
                geoFenceTransition == Geofence.GEOFENCE_TRANSITION_EXIT ) {
            // Get the geofence that were triggered
            List<Geofence> triggeringGeofences = geofencingEvent.getTriggeringGeofences();

            String geofenceTransitionDetails = getGeofenceTrasitionDetails(geoFenceTransition, triggeringGeofences );

            // Send notification details as a String
            sendNotification( geofenceTransitionDetails );
        }
```

```
}


private String getGeofenceTrasitionDetails(int geoFenceTransition, List<Geofence> triggeringGeofences) {
    // get the ID of each geofence triggered
    ArrayList<String> triggeringGeofencesList = new ArrayList<>();
    for ( Geofence geofence : triggeringGeofences ) {
        triggeringGeofencesList.add( geofence.getRequestId() );
    }

    String status = null;
    if ( geoFenceTransition == Geofence.GEOFENCE_TRANSITION_ENTER )
        status = "Entering ";
    else if ( geoFenceTransition == Geofence.GEOFENCE_TRANSITION_EXIT )
        status = "Exiting ";
    return status + TextUtils.join( ", ", triggeringGeofencesList);
}

private void sendNotification( String msg ) {
    Log.i(TAG, "sendNotification: " + msg );

    // Intent to start the main Activity
    Intent notificationIntent = MainActivity.makeNotificationIntent(
            getApplicationContext(), msg
    );

    TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
    stackBuilder.addParentStack(MainActivity.class);
    stackBuilder.addNextIntent(notificationIntent);
    PendingIntent notificationPendingIntent = stackBuilder.getPendingIntent(0, PendingIntent.FLAG_UPDATE_CURRENT);
```

```
        // Creating and sending Notification
        NotificationManager notificatioMng =
                (NotificationManager) getSystemService( Context.NOTIFICATION_SERVICE );
        notificatioMng.notify(
                GEOFENCE_NOTIFICATION_ID,
                createNotification(msg, notificationPendingIntent));

    }

    // Create notification
    private Notification createNotification(String msg, PendingIntent notificationPendingIntent) {
        NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(this);
        notificationBuilder
                .setSmallIcon(R.drawable.ic_action_location)
                .setColor(Color.RED)
                .setContentTitle(msg)
                .setContentText("Geofence Notification!")
                .setContentIntent(notificationPendingIntent)
                .setDefaults(Notification.DEFAULT_LIGHTS | Notification.DEFAULT_VIBRATE | Notification.DEFAULT_SOUND)
                .setAutoCancel(true);
        return notificationBuilder.build();
    }


    private static String getErrorString(int errorCode) {
        switch (errorCode) {
            case GeofenceStatusCodes.GEOFENCE_NOT_AVAILABLE:
                return "GeoFence not available";
            case GeofenceStatusCodes.GEOFENCE_TOO_MANY_GEOFENCES:
                return "Too many GeoFences";
            case GeofenceStatusCodes.GEOFENCE_TOO_MANY_PENDING_INTENTS:
                return "Too many pending intents";
```

```
            default:
                return "Unknown error.";
        }
    }
}
```