Sprint 4

Date	11 November 2022
Team ID	PNT2022TMID14122
Project Name	IoT Based Safety Gadget for Child Safety Monitoring and Notification

Sprint 4 is send alert notification when entered and exited the geofence **Coding:**

```java
package com.example.geofence;

import
android.content.BroadcastReceiver; import
android.content.Context; import
android.content.Intent; import
android.location.Location; import
android.os.CountDownTimer; import
android.util.Log; import
android.widget.Toast;

import com.google.android.gms.location.Geofence
; import com.google.android.gms.location.GeofencingEvent;
import java.util.List
; import android.os.Handler;

public class GeofenceBroadcastReceiver extends BroadcastReceiver {

    private static final String TAG = "GeofenceBroadcastReceiv";
    @Override      public void onReceive(Context context,
Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
// an Intent broadcast
        //.
      /*Toast.makeText(context, "GEOFENCE_ENTERED",
Toast.LENGTH_SHORT).show();

        final Toast mToastToShow;
```

```java
        int toastDurationInMilliSeconds = 1200000;        mToastToShow =
Toast.makeText(context, "GEOFENCE_EXITED", Toast.LENGTH_LONG);



        // Set the countdown to display the toast        CountDownTimer toastCountDown;
toastCountDown = new CountDownTimer(toastDurationInMilliSeconds,
100000) {            public void onTick(long millisUntilFinished) {
mToastToShow.show();
        }

        public void onFinish() {
mToastToShow.cancel();            }
      };

      // Show the toast and starts the countdown
mToastToShow.show();        toastCountDown.start();*/


    NotificationHelper notificationHelper = new
NotificationHelper(context);

notificationHelper.sendHighPriorityNotification("GEOFENCE_TRANSITION_ENTER" , "",
MapsActivity.class);

    GeofencingEvent geofencingEvent = GeofencingEvent.fromIntent(intent);

      if (geofencingEvent.hasError()) {
        Log.d(TAG, "onReceive: Error receiving geofence event...");            return;
      }

      List<Geofence> geofenceList =
geofencingEvent.getTriggeringGeofences();        for (Geofence
geofence: geofenceList) {
        Log.d(TAG, "onReceive: " + geofence.getRequestId());        }
//      Location location = geofencingEvent.getTriggeringLocation();        int transitionType =
geofencingEvent.getGeofenceTransition();
      switch (transitionType) {            case
Geofence.GEOFENCE_TRANSITION_ENTER:
```

```
                notificationHelper.sendHighPriorityNotification("Entered the Location", "", MapsActivity.class);
break;                case Geofence.GEOFENCE_TRANSITION_EXIT:


                notificationHelper.sendHighPriorityNotification("Exited      the         Location     ",     "",
MapsActivity.class);            break;
    }


  }
 }
```

Notification Class:

```
package com.example.geofence;

import android.app.Notification; import
android.app.NotificationChannel; import
android.app.NotificationManager; import
android.app.PendingIntent; import
android.content.Context; import
android.content.ContextWrapper; import
android.content.Intent; import android.graphics.Color;
import android.os.Build;

import androidx.annotation.RequiresApi; import
androidx.core.app.NotificationCompat; import
androidx.core.app.NotificationManagerCompat;

import java.util.Random;

public class NotificationHelper extends ContextWrapper {

  private static final String TAG = "NotificationHelper";

  public NotificationHelper(Context base) {        super(base);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {          createChannels();
    }
  }
  private String CHANNEL_NAME = "High priority channel";
  private String CHANNEL_ID = "com.example.geofence" + CHANNEL_NAME;
```

```java
@RequiresApi(api = Build.VERSION_CODES.O)    private void createChannels()
{

    NotificationChannel notificationChannel = new
NotificationChannel(CHANNEL_ID, CHANNEL_NAME, NotificationManager.IMPORTANCE_HIGH);
notificationChannel.enableLights(true);        notificationChannel.enableVibration(true);
notificationChannel.setDescription("this is the description of the channel.");
notificationChannel.setLightColor(Color.RED);


notificationChannel.setLockscreenVisibility(Notification.VISIBILITY_PUBLIC) ;        NotificationManager
manager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
manager.createNotificationChannel(notificationChannel);    }
    public void sendHighPriorityNotification(String title, String body, Class activityName) {


    Intent intent = new Intent(this, activityName);
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 267, intent,
PendingIntent.FLAG_UPDATE_CURRENT);
    Notification notification = new NotificationCompat.Builder(this, CHANNEL_ID)
//          .setContentTitle(title)
//          .setContentText(body)
        .setSmallIcon(R.drawable.ic_launcher_background)
        .setPriority(NotificationCompat.PRIORITY_HIGH)             .setStyle(new
NotificationCompat.BigTextStyle().setSummaryText("summary").setBigContentTi
tle(title).bigText(body))
        .setContentIntent(pendingIntent)
        .setAutoCancel(true)
        .build();


    NotificationManagerCompat.from(this).notify(new Random().nextInt(), notification);
}
}
```

Output :