SPRINT 4

TEAM ID : PNT2022TMID32082

TEAM LEADER : T.A.Kaviprakash

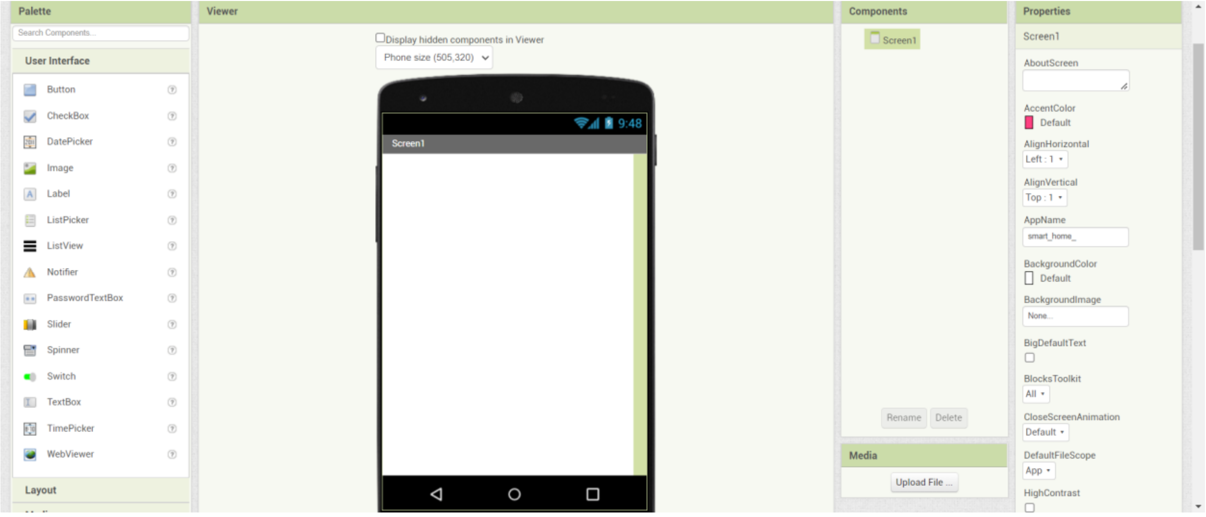
TEAM MEMBER 1: B.Chiradeep.

TEAM MEMBER 2: R.Kumaravel.

TEAM MEMBER 3: S.poovarasan

CREATING THE MIT APPLICATION AND SHOWING THE CHILD'S LOCATION

Create App in the MIT App inventor:



AboutScreen

AccentColor: Default

AlignHorizontal: Left : 1

AlignVertical: Top : 1

AppName: smart_home_

BackgroundColor: Default

BackgroundImage: None...

BigDefaultText: □

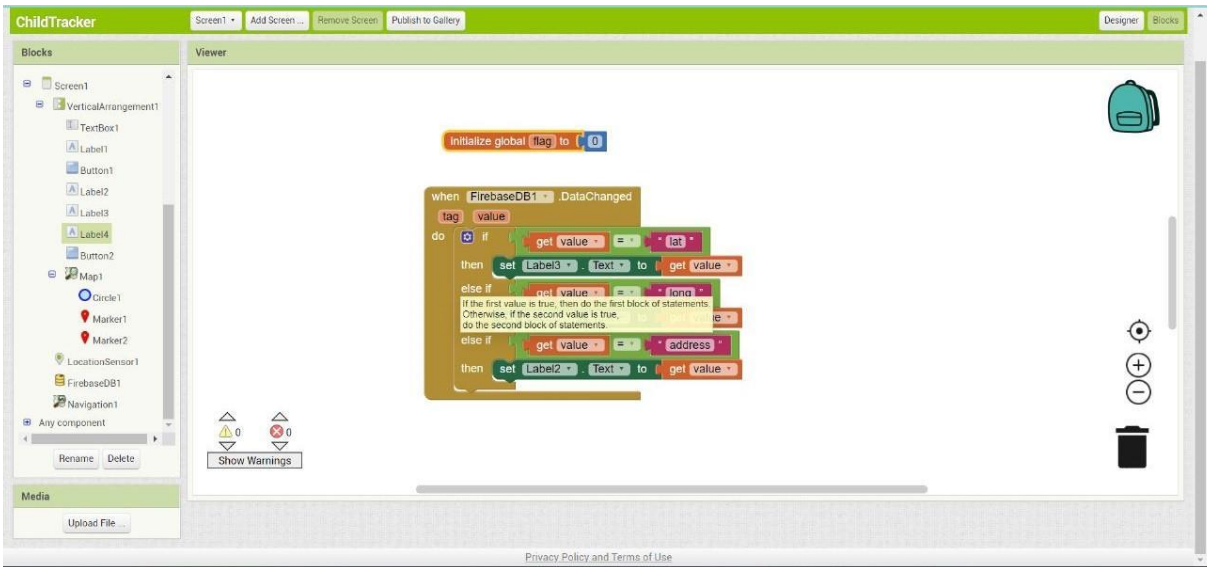
BlocksToolkit: All

CloseScreenAnimation: Default

DefaultFileScope: App

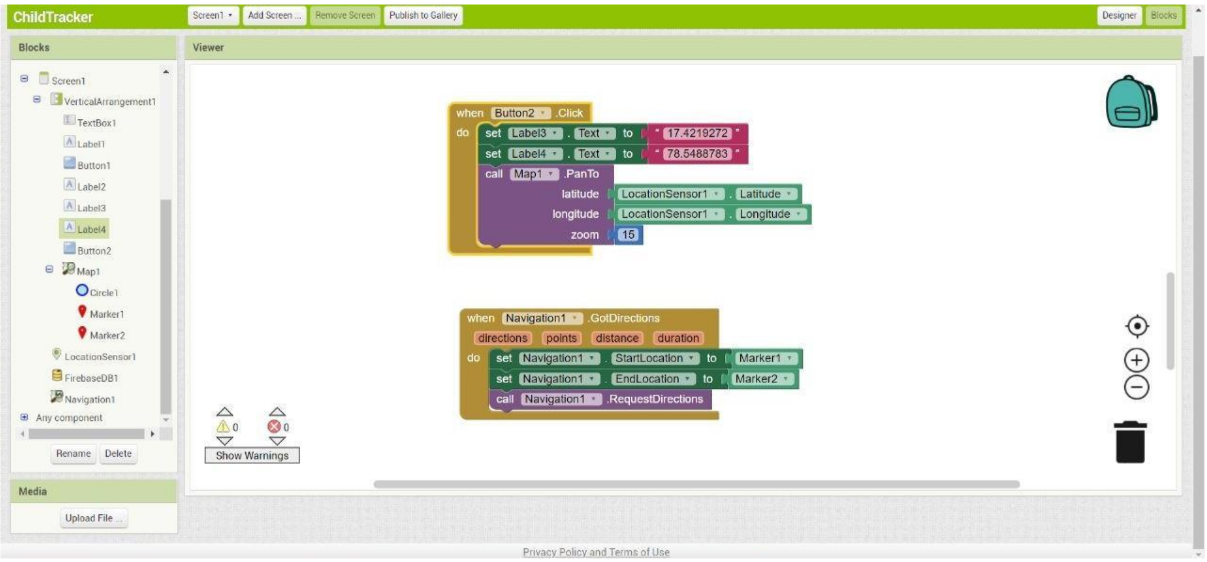
HighContrast: □

Block configuration:

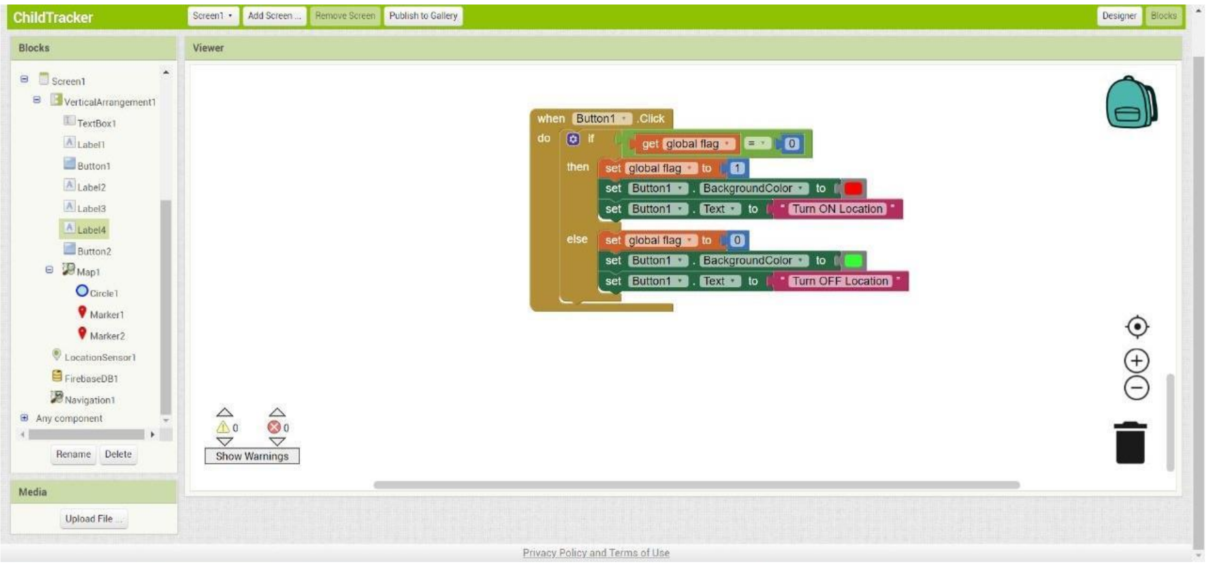


```
initialize global flag to 0

when FirebaseDB1 .DataChanged
 tag  value
do  if      get value = " lat "
    then    set Label3 . Text to get value
    else if get value = " long "
    ...e
    else if get value = " address "
    then    set Label2 . Text to get value
```

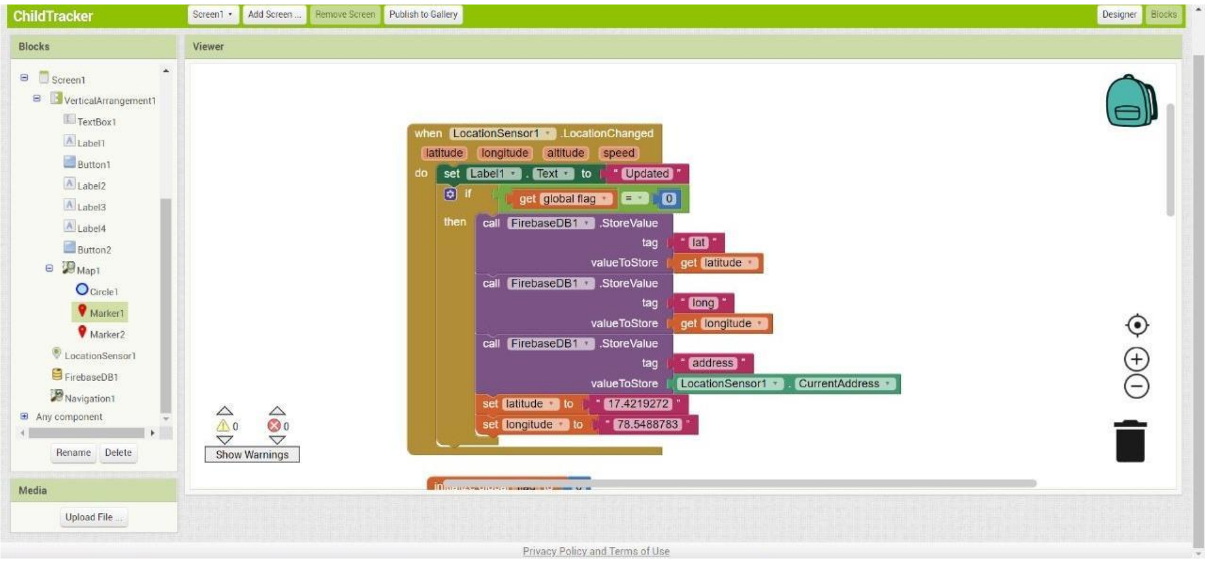
If the first value is true, then do the first block of statements.
Otherwise, if the second value is true,
do the second block of statements.



```
when Button2 .Click
do  set Label3 . Text to " 17.4219272 "
    set Label4 . Text to " 78.5488783 "
    call Map1 .PanTo
              latitude  LocationSensor1 . Latitude
              longitude LocationSensor1 . Longitude
              zoom      15

when Navigation1 .GotDirections
 directions  points  distance  duration
do  set Navigation1 . StartLocation to Marker1
    set Navigation1 . EndLocation to Marker2
    call Navigation1 .RequestDirections
```

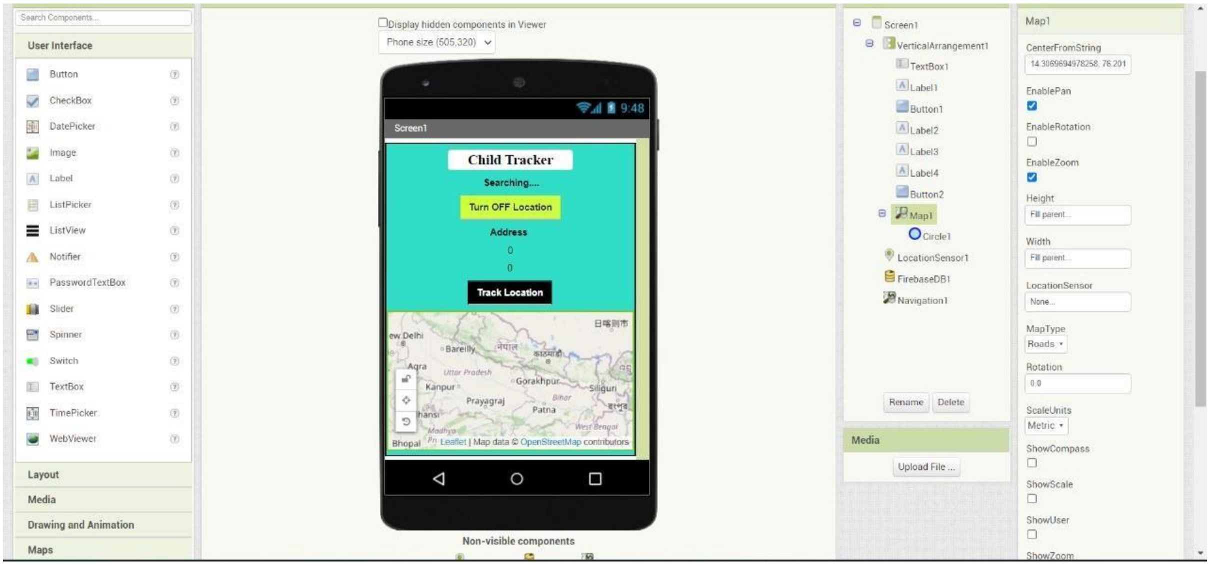


```
when Button1 .Click
do  if   get global flag = 0
    then set global flag to 1
         set Button1 . BackgroundColor to 🟥
         set Button1 . Text to " Turn ON Location "
    else set global flag to 0
         set Button1 . BackgroundColor to 🟩
         set Button1 . Text to " Turn OFF Location "
```



```
when LocationSensor1 .LocationChanged
  latitude  longitude  altitude  speed
do  set Label1 . Text to " Updated "
    if  get global flag = 0
    then  call FirebaseDB1 .StoreValue
                           tag " lat "
                  valueToStore get latitude
          call FirebaseDB1 .StoreValue
                           tag " long "
                  valueToStore get longitude
          call FirebaseDB1 .StoreValue
                           tag " address "
                  valueToStore LocationSensor1 . CurrentAddress
          set latitude to " 17.4219272 "
          set longitude to " 78.5488783 "
```

OUTPUT(APPINVENTOR:



CenterFromString: 14.3069694978258, 76.201

EnablePan: ☑

EnableRotation: □

EnableZoom: ☑

Height: Fill parent...

Width: Fill parent...

LocationSensor: None...

MapType: Roads

Rotation: 0.0

ScaleUnits: Metric

ShowCompass: □

ShowScale: □

ShowUser: □

ShowZoom

CODE:

geofence:

package com.example.geofence;

import android.app.PendingIntent;

import android.content.Context;

import android.content.ContextWrapper;

import android.content.Intent;

```
import android.widget.Toast;

import com.google.android.gms.common.api.ApiException;

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

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

import com.google.android.gms.location.GeofencingRequest;

import com.google.android.gms.maps.model.LatLng;

public class GeofenceHelper extends ContextWrapper {

 private static final String TAG = "GeofenceHelper";

 PendingIntent pendingIntent;

 public GeofenceHelper(Context base) { super(base);

 }

 public GeofencingRequest getGeofencingRequest(Geofence geofence) { return new

GeofencingRequest.Builder()

 .addGeofence(geofence)

 .setInitialTrigger(GeofencingRequest.INITIAL_TRIGGER_ENTER)

 .build();

 }

 public Geofence getGeofence(String ID, LatLng latLng, float radius, int transitionTypes) { return new

Geofence.Builder()

 .setCircularRegion(latLng.latitude, latLng.longitude, radius) .setRequestId(ID)

 .setTransitionTypes(transitionTypes)

 .setLoiteringDelay(5000)

 .setExpirationDuration(Geofence.NEVER_EXPIRE)

 .build();

 }

 public PendingIntent getPendingIntent() { if (pendingIntent != null) {

return pendingIntent;
}
 Intent intent = new Intent(this, GeofenceBroadcastReceiver.class);
 pendingIntent = PendingIntent.getBroadcast(this, 2607, intent, PendingIntent.FLAG_IMMUTABLE);
 return pendingIntent;
 }
 public String getErrorString(Exception e) {
if (e instanceof ApiException) {
 ApiException apiException = (ApiException) e;
switch(apiException.getStatusCode())
{
case GeofenceStatusCodes
.GEOFENCE_NOT_AVAILABLE:
return "GEOFENCE_NOT_AVAILABLE";
case
GeofenceStatusCodes
 .GEOFENCE_TOO_MANY_GEOFENCES:
return "GEOFENCE_TOO_MANY_GEOFENCES"; case
GeofenceStatusCodes
 .GEOFENCE_TOO_MANY_PENDING_INTENTS:
 return "GEOFENCE_TOO_MANY_PENDING_INTENTS";
 } }
 return e.getLocalizedMessage();
 }
}
```

ALERT & NOTIFICATION:

```
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;
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:

```
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;

 @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);

}

}
```