CS8611 - MINI PROJECT

GEOFENCING

ALGORITHM:

STEP-1: The first step is opening the geofence project in android studio.

STEP-2: The second step is creating the source code according to the geofence activities first step in requesting geofence monitoring is to request the necessary permissions. To use geofencing, your app must request the following:

- ACCESS FINE LOCATION
- ACCESS BACKGROUND LOCATION

STEP-3: The third step is to create and add geofences using the location API's builder class for creating Geofence objects, and the convenience class for adding them. Also, to handle the intents sent from Location Services when geofence transitions occur, you can define a PendingIntent.

STEP-4: The fourth step is to use Geofence.Builder to create a geofence, setting the desired radius, duration, and transition types for the geofence.

STEP-5: The fifth step is to uses the GeofencingRequest class and its nested GeofencingRequestBuilder class to specify the geofences to monitor and to set how related geofence events are triggered.

STEP-6: The sixth step is to define the broadcast Receiver foe geofence transitions. An Intent sent from Location Services can trigger various actions in your app, but you should not have it start an activity or fragment, because components should only become visible in response to a user action. In

many cases, a BroadcastReceiver is a good way to handle a geofence transition. A BroadcastReceiver gets updates when an event occurs, such as a transition into or out of a geofence, and can start long-running background work.

STEP-7: The seventh step is to add geofences, use the GeofencingClient.addGeofences() method. Provide the GeofencingRequest object, and the PendingIntent.

STEP-8: The eight step is to handle geofence transitions. When Location Services detects that the user has entered or exited a geofence, it sends out the Intent contained in the PendingIntent you included in the request to add geofences. A broadcast receiver like GeofenceBroadcastReceiver notices that the Intent was invoked and can then obtain the geofencing event from the intent, determine the type of Geofence transition(s), and determine which of the defined geofences was triggered. The broadcast receiver can direct an app to start performing background work or, if desired, send a notification as output.

STEP-9: The ninth step is to stop the geofence monitoring. Stopping geofence monitoring when it is no longer needed or desired can help save battery power and CPU cycles on the device. You can stop geofence monitoring in the main activity used to add and remove geofences; removing a geofence stops it immediately. The API provides methods to remove geofences either by request IDs, or by removing geofences associated with a given PendingIntent.

STEP-10: After the successful completion of the Geofencing project close the android studio.