TASK 3 - the deadline for submitting the task is 30.01.2020

The aim of the exercise is to get acquainted with the following issues:
- network services available in Android
- parsing XML/JSON responses,
- storing data in the memory of the device (e.g. in a private directory of applications)
- working with multimodules project
- creating with ui and unit tests
- working with database and API (ex. Firebase, azure)
- using Service, Location Service API and Activity Recognition Transition API

Configuration view

The app will load the required configuration from a HTTP endpoint.

`URL: https://backed-url?id=223344`

The ID in the query string must be provided through the UI.

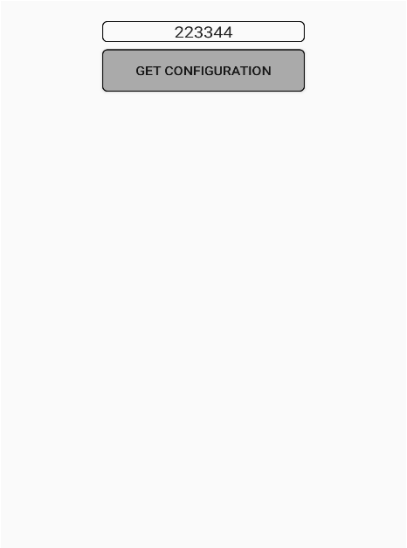
Example configuration response JSON:

```
{
"name": "Test 1",
    "token": " eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
Ww7TXlTYWZldHkifQ.6VDCmHZ_jGYhfpB8G1FBsVaFMKlIwCwH-HzhCyE5bUU",
    "trackedObjectId": "232a834e-a8a5-67cb-aee4-015bed6a5442",
    "positionIntervalInMinutes": 15

}
```

UI mock-up

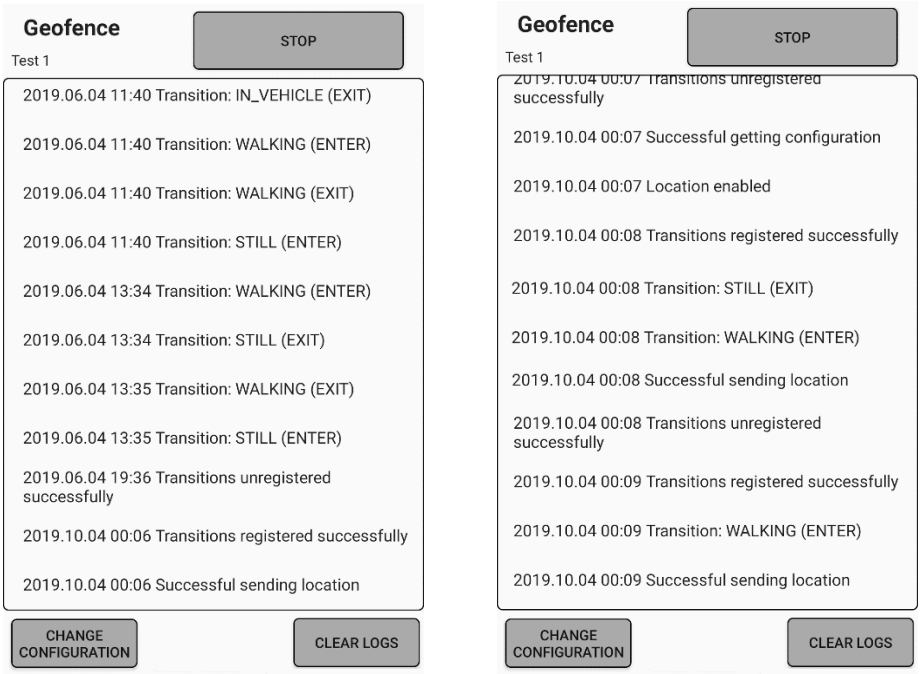
The configuration ID must be provided through the UI. If the configuration could not be loaded / parsed, feedback must be provided in the UI.



If the configuration is loaded, the following elements must be visible in the UI.

- Name of the loaded configuration

- Start / stop button
- Log of events since location tracking was started with timestamps
- Button to change configuration



HTTP call to backend

The app will make HTTP POST calls to the backend whenever a geofence is entered or exited. See cURL example below for HTTP POST call headers and body.

```
curl -X POST \
  https://backend-url/report \
  -H 'Cache-Control: no-cache' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <token>' \
  -d '{
       "location":{
                       "acc": <accuracy>,
                       "alt": <altitude>,
                       "bea": 0,
                       "lat": <latitude>,
                       "long": <longitude>,
                       "prov": "GPS",
                       "spd": 0,
                       "sat": 0,
                       "time": "<timestamp>",
                       "serial": "",
                       "tid": "<trackedObjectId>",
                       "plat": "Android",
                       "platVer": "1.0",
                       "bat": 1

       }
}'


Request body properties:

<token>              JWT from configuration
```

<accuracy>	GPS position accuracy
<altitude>	GPS position altitude
<latitude>	GPS position latitude
<longitude>	GPS position longitude
<timestamp>	Current device time in UTC ex 2018-09-04T12:00:19.131Z
<trackedObjectId>	From configuration

The remaining properties like bearing (bea), number of satellites (sat), speed (spd) and battery level (bat) can be provided if easily available but are not required.

Application behaviour

1. The application must request all required permissions. Results of permission requests must be logged.
2. Events like permission requests, HTTP calls, activity transition, problems with the Internet and GPS, etc must be logged to a persisted log with timestamps (local device time). It must be possible to clear the log manually.
3. The application must be able to report location data even when the application is not active / in the foreground. Running a foreground service seems to be the best way of ensuring that the application is not suspended by the OS.
4. The application must use the Location Service API to periodically get the device location and send it to the server. The frequency of the updates will be configured through the `positionIntervalInMinutes` property.
5. High accuracy priority must be used to get the highest possible location accuracy. If possible, a GPS location must be forced (not network or WiFi location).
6. To reduce battery consumption, the Activity Recognition Transition API must be used. If the current activity is STILL, location updates should not be requested or sent to the server.
7. The application sends device locations to the server (coordinates).
8. Activity transitions (STILL, WALKING, etc) must be logged.
9. Separate bussines logic into second module and provide unit and ui tests

Application compatibility

Android version 5 and above must be supported. Before submitting test your application on Android 8.1 or newer