



Mobile App Development 2

Study diary

Eric Brown

SISÄLLYS

[1 Week exercises 3](#_Toc86919927)

[1.1 Taskulamppusovellus (Camera HW API) 3](#_Toc86919928)

[1.2 Vatupassi (Sensor API) 3](#_Toc86919929)

[2 Week excercises 4](#_Toc86919930)

[3 Week exercises 5](#_Toc86919931)

[4 Week exercises 6](#_Toc86919932)

[5 Week exercises 7](#_Toc86919933)

[6 Week exercises 8](#_Toc86919934)

[7 Week exercises 9](#_Toc86919935)

[8 Week exercises 10](#_Toc86919936)

[9 Week exercises 11](#_Toc86919937)

[10 Week exercises 12](#_Toc86919938)

[Final project (not mandatory but required for best grades) 13](#_Toc86919939)

[Sources used with exercises 14](#_Toc86919940)

# Week exercises

## Android BMI App

Copy/paste your relevant source, screenshots etc. here.

Source: <https://github.com/evvic/mobile_app_development/tree/main/BMI_app>

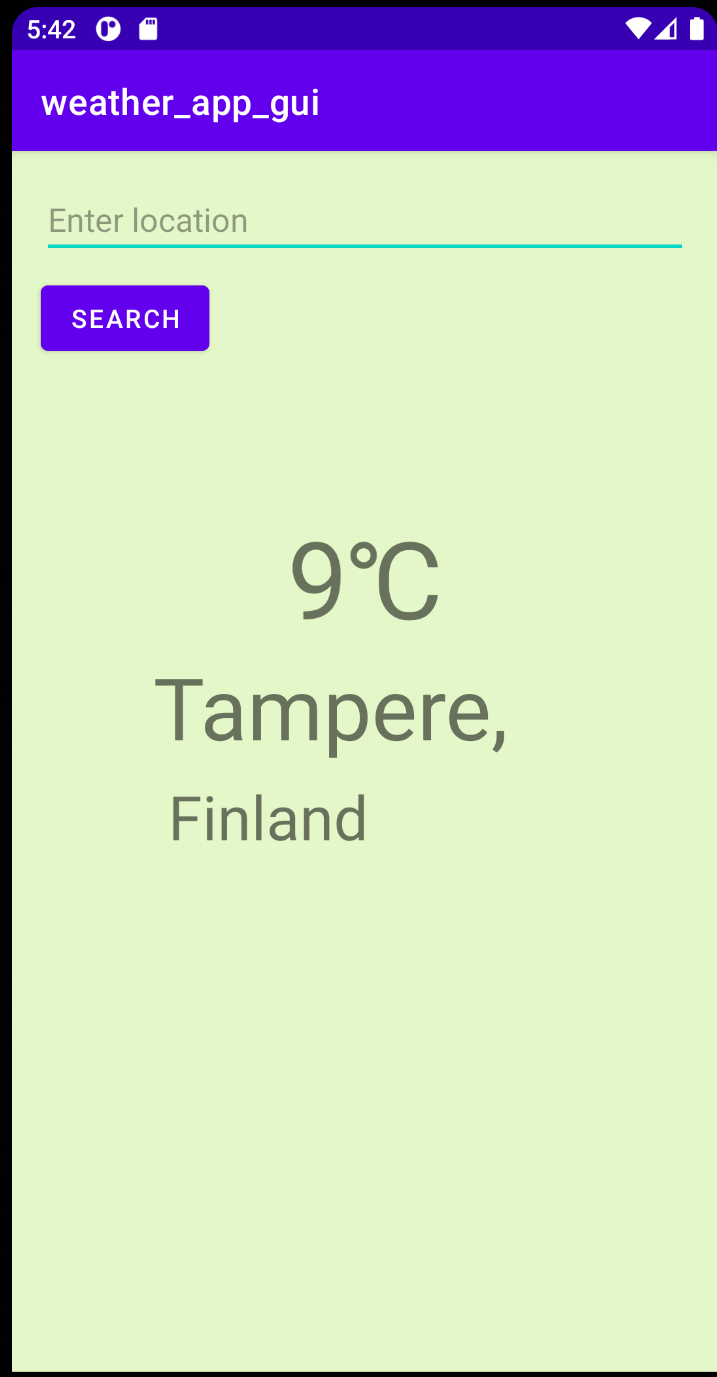
I followed along in class when designing this app. My app looks the exact same as the professors, where when the button is pressed, the BMI is calculated and displayed above the button, near the bottom of the screen.



## Android Weather App GUI

Copy/paste your source, screenshots etc. here

Source:<https://github.com/evvic/mobile_app_development/tree/main/weather_app_gui>

I created the weather GUI design thinking about how a typical weather app may look. Usually very simple and clean, depending on the informatics needed. Basically it could use the user’s location to determine the location and temperature, or the user can enter a location and the app would show the current temperature of that given location.

XML code for project:

<?xml version="1.0" encoding="utf-8"?><androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:background="#AED6F3AE" tools:context=".MainActivity"> <Button android:id="@+id/button" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="16dp" android:layout\_marginTop="8dp" android:text="Search" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/searcch" /> <TextView android:id="@+id/locationCity" android:layout\_width="236dp" android:layout\_height="70dp" android:layout\_marginStart="87dp" android:layout\_marginEnd="87dp" android:text="Tampere," android:textSize="48sp" app:layout\_constraintBottom\_toTopOf="@+id/country" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" /> <TextView android:id="@+id/country" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="87dp" android:layout\_marginBottom="286dp" android:text="Finland" android:textSize="34sp" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintStart\_toStartOf="parent" /> <TextView android:id="@+id/temperature" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="162dp" android:layout\_marginEnd="162dp" android:text="9℃" android:textSize="60sp" app:layout\_constraintBottom\_toTopOf="@+id/locationCity" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" /> <EditText android:id="@+id/searcch" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="16dp" android:layout\_marginTop="16dp" android:layout\_marginEnd="16dp" android:ems="10" android:hint="Enter location" android:inputType="textPersonName" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" /></androidx.constraintlayout.widget.ConstraintLayout>

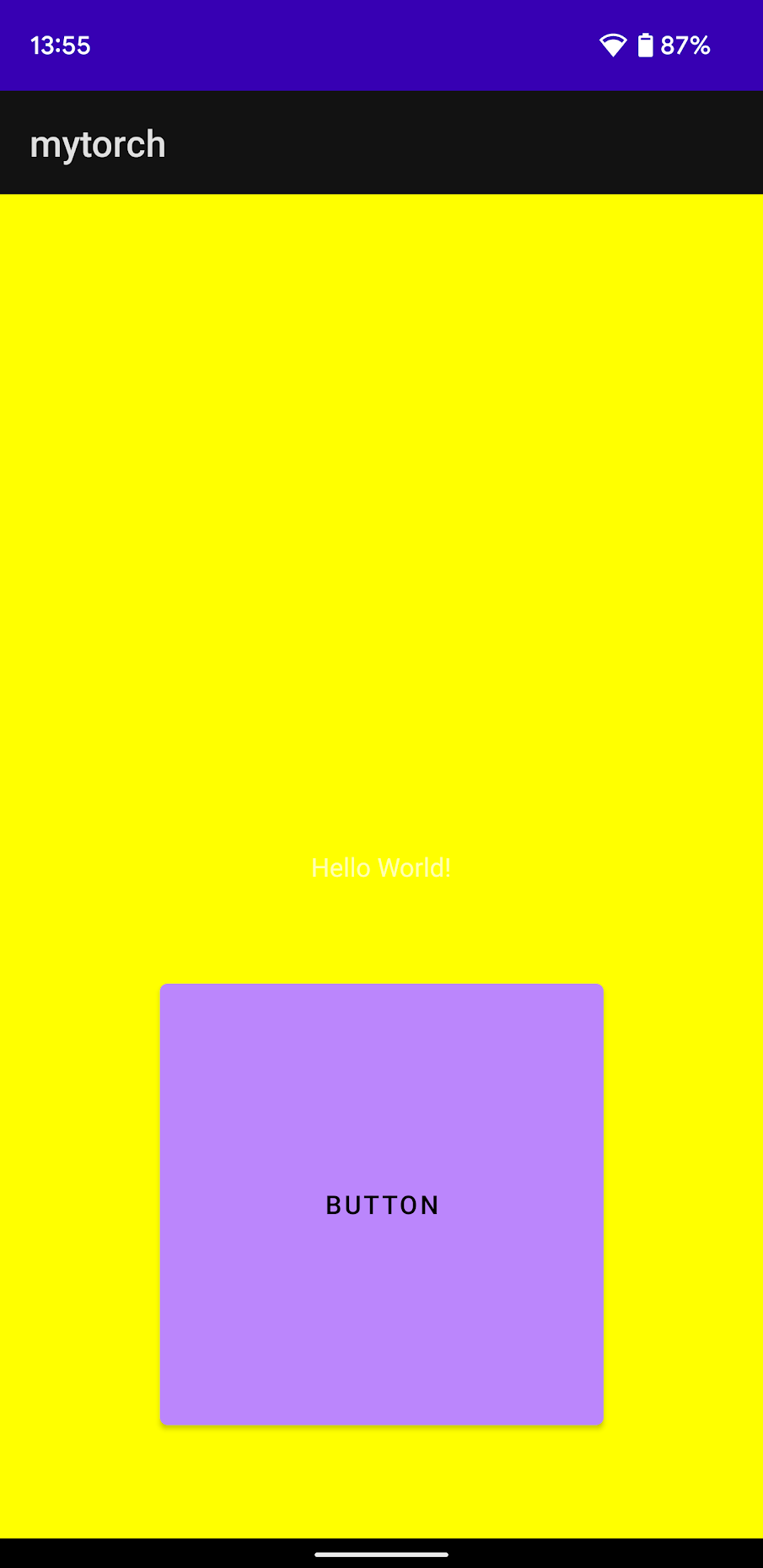
My Gradle wasn’t building correctly so I had to add these extra lines to my build.gradle(:app): https://stackoverflow.com/a/69043734

# Week excercises

**2.1: Synchronous API call example - Camera HW API - "Flashlight APP"**

Source: https://github.com/evvic/mobile\_app\_development/tree/main/mytorch

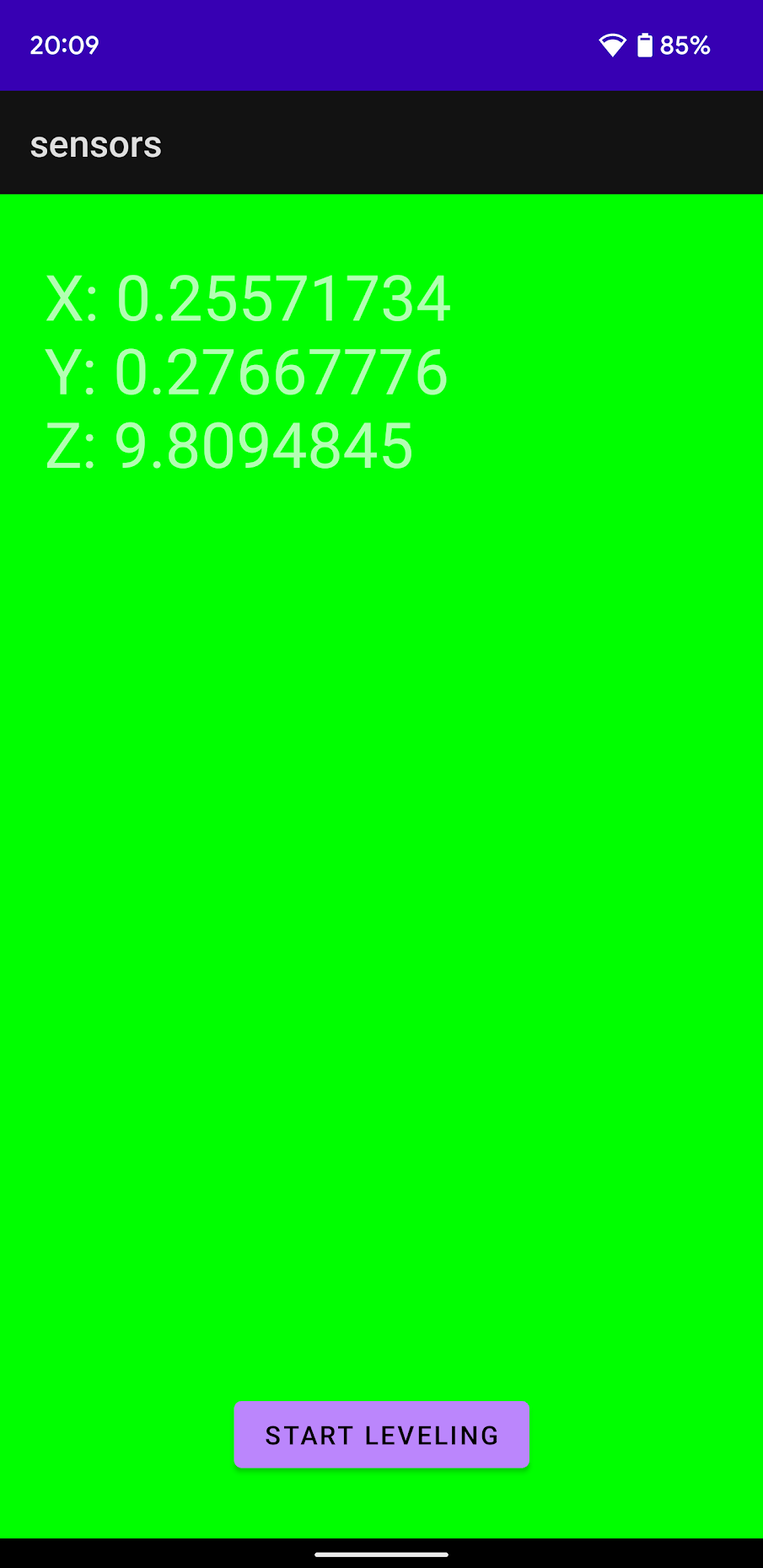
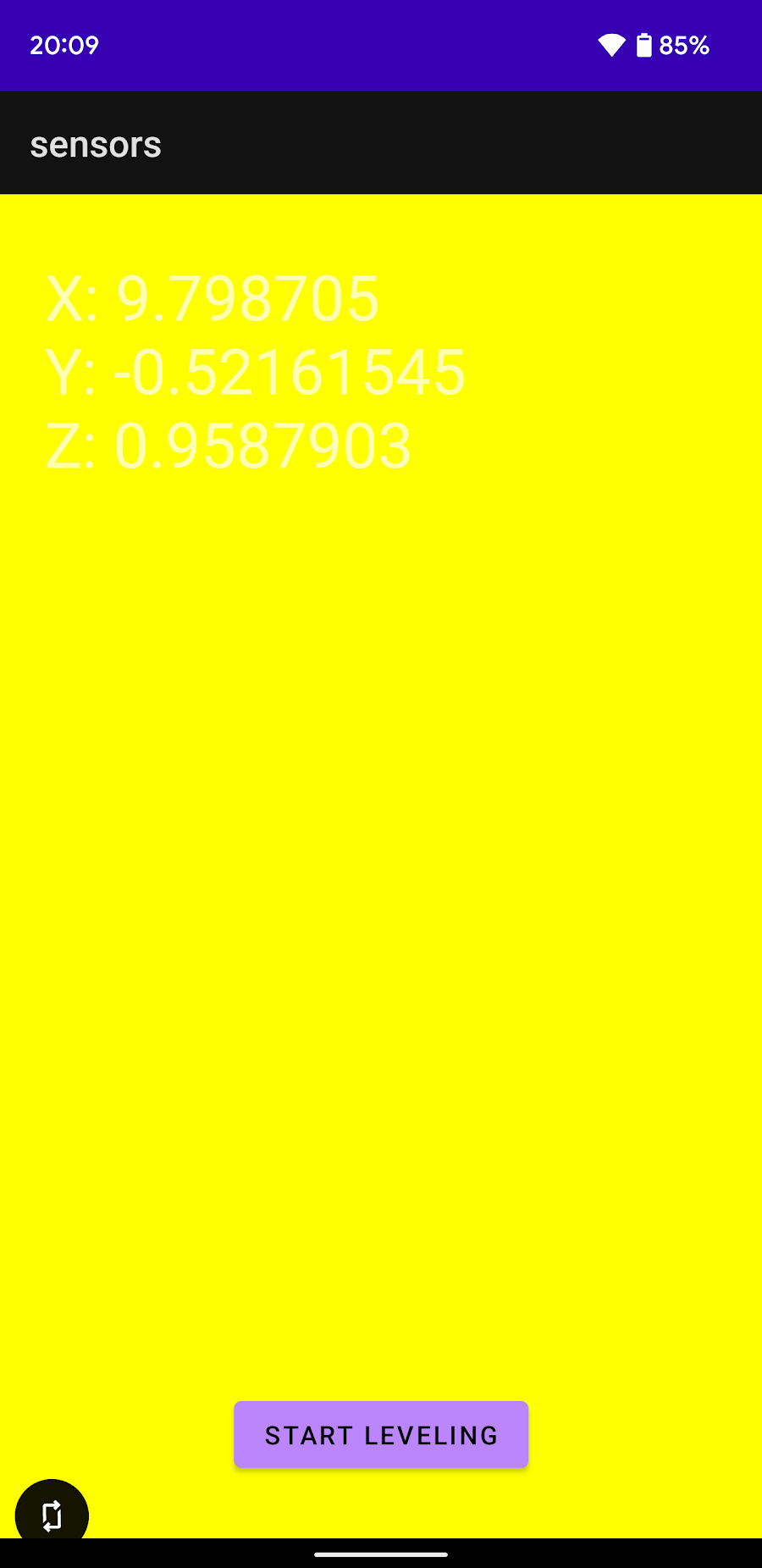
I created a button that toggles the flashlight attached to my back camera to turn on and off. Also when the light is off, the background is black, but when the light is turned on, the background turns yellow.

****

# **2.2: Asynchronous API call example (listener) - Sensor API and "Level APP"**

Source: https://github.com/evvic/mobile\_app\_development/tree/main/sensors

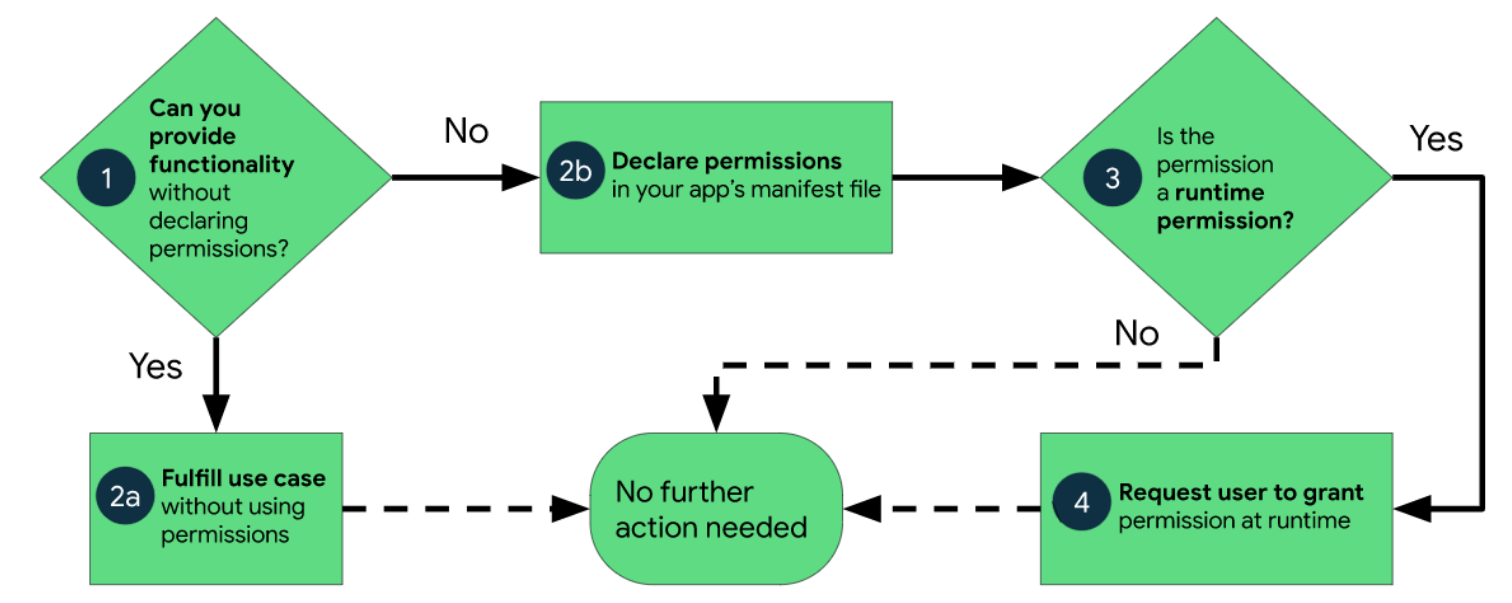
This app uses a listener of the accelerometer to update whenever the phones axis’ change. When the X-axis nears 10, basically the phone is horizontal, the background changes from the default blue to yellow. Then, if the phone is flat on a desk, the x-axis and y-axis are near zero, the background turns green.



# Week exercises

# **Written questions and answers**

The Android App permission work flow:



Basically this flow diagram shows the steps necessary for the program to have access rights to using a user’s devices API. Depending on how sensitive the data is towards the user, the program may need extra permissions before accessing that specific API. The program might not even need to declare the permission in the manifest file if it’s not sensitive data. Or if the permission needs to be declared, it might still need the user’s explicit concent before the API can be accessed by the program.

For example, to access a user’s general and specific location, the manifest file must coontaint the following lines inside the <manifest> tags:

**<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"></uses-permission>  
<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"></uses-permission>**

Then, because this is sensitive information, before being able to access the device’s location, it must be granted access by the user through code such as:

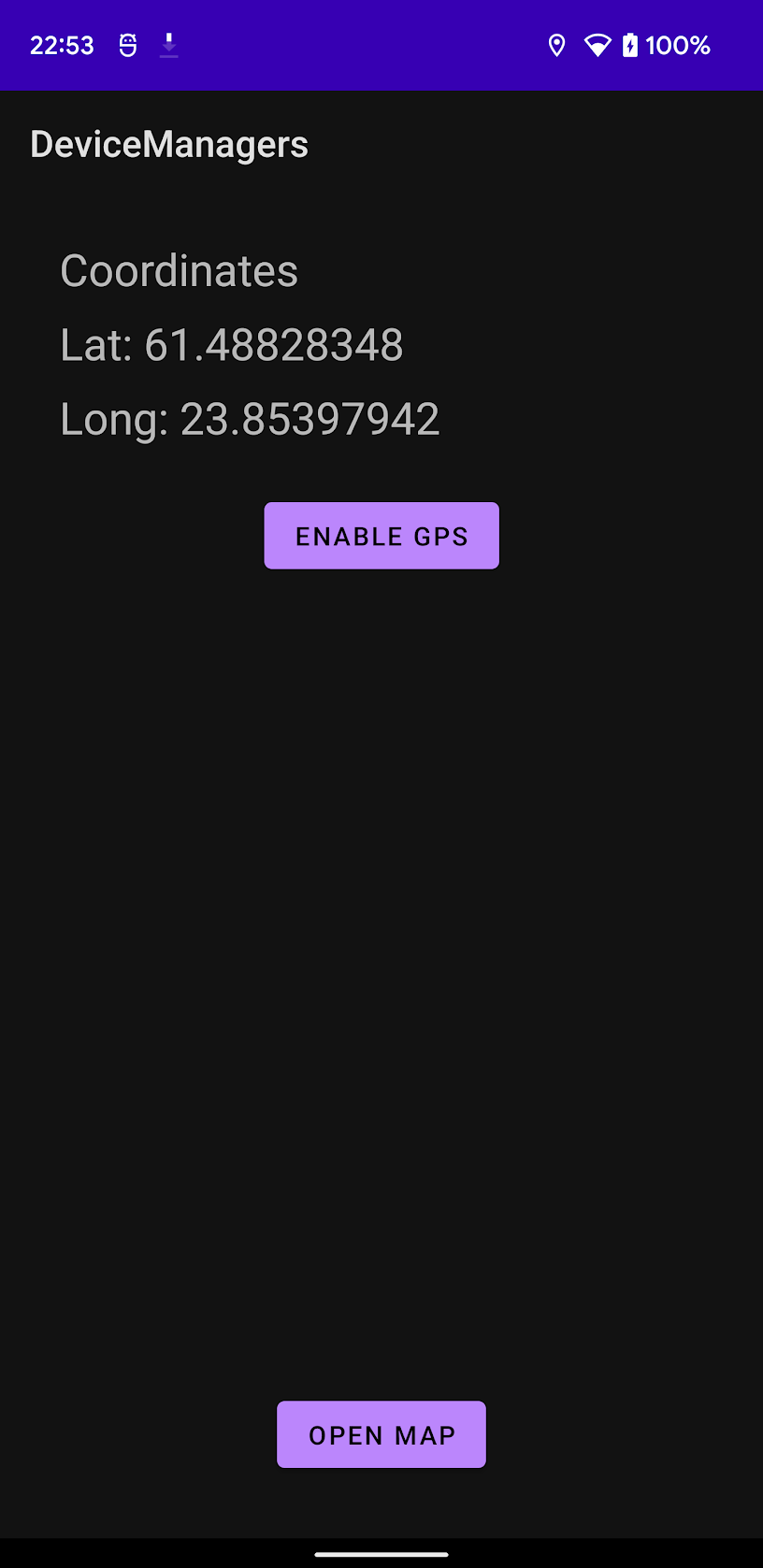
**if(checkSelfPermission(Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*  
&& checkSelfPermission(Manifest.permission.*ACCESS\_COARSE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*)  
{  
 ActivityCompat.requestPermissions(this,  
 *arrayOf*(Manifest.permission.*ACCESS\_FINE\_LOCATION*, Manifest.permission.*ACCESS\_COARSE\_LOCATION*),  
 0)****}**

The Manifest is used to check if permission has been granted (as shown in the if statement), and if not, a permission request is sent that the user must approve of before the specific device API can be used.

# **Code Lab**

I created the simple app that asks the user for permission to use their specific and general location, then uses that data from the devices API and updates to give the current latitude and longitude of the device.

Source code: <https://github.com/evvic/mobile_app_development/tree/main/DeviceManagers>\

Clicking ”ENABLE GPS” asks the user for location permissions and then begins a listener to any GPS location updates, which updates the UI.

Then clicking open map sends the intent to Google Maps with the current devices location.

# Week exercises

# Week exercises

# Week exercises

# Week exercises

# Week exercises

# Week exercises

# Final project (not mandatory but required for best grades)

Sources used with exercises

List here the possible sources you’ve used with exercises (e.g. stackoverflow, tutorialspoint, github etc.)