



Mobile App Development 2

Study diary

Jaakko Virtanen

SISÄLLYS

[1 Week exercises 3](#_Toc87035283)

[1.1 Android BMI App 3](#_Toc87035284)

[1.2 Android Weather App GUI 3](#_Toc87035285)

[2 Week excercises 4](#_Toc87035286)

[3 Week exercises 5](#_Toc87035287)

[4 Week exercises 6](#_Toc87035288)

[5 Week exercises 7](#_Toc87035289)

[6 Week exercises 8](#_Toc87035290)

[7 Week exercises 9](#_Toc87035291)

[8 Week exercises 10](#_Toc87035292)

[9 Week exercises 11](#_Toc87035293)

[10 Week exercises 12](#_Toc87035294)

[Final project (not mandatory but required for best grades) 13](#_Toc87035295)

[Sources used with exercises 14](#_Toc87035296)

# Week exercises

## Android BMI App

<https://github.com/jaakko13/mobileAppDev2/tree/master/1.1BmiApp>

No real challenges in this work. Very simple introductory assignment.

## Android Weather App GUI

<https://github.com/jaakko13/mobileAppDev2/tree/master/12_weatherAppGUI>

<?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"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/location"  
 android:textSize="24dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="164dp"  
 android:layout\_marginTop="98dp"  
 android:text="Tampere"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/locationText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="168dp"  
 android:layout\_marginTop="128dp"  
 android:text="Location"  
 android:textSize="16dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/temperature"  
 android:textSize="24dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="164dp"  
 android:layout\_marginTop="64dp"  
 android:text="10 Celsius"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/location" />  
  
 <TextView  
 android:id="@+id/temperatureText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="164dp"  
 android:layout\_marginTop="96dp"  
 android:text="Temperature"  
 android:textSize="16dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/location" />  
  
 <TextView  
 android:id="@+id/percipitationText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="56dp"  
 android:layout\_marginTop="212dp"  
 android:text="Percipitation"  
 android:textSize="16dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/location" />  
  
 <TextView  
 android:id="@+id/HumidityText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="240dp"  
 android:layout\_marginTop="212dp"  
 android:text="Humidity"  
 android:textSize="16dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/location" />  
  
 <TextView  
 android:id="@+id/percipitation"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="56dp"  
 android:layout\_marginBottom="165dp"  
 android:text="80 Percent"  
 android:textSize="24dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/button"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <TextView  
 android:id="@+id/Humidity"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="56dp"  
 android:layout\_marginBottom="164dp"  
 android:text="10 Percent"  
 android:textSize="24dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/button"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <Button  
 android:id="@+id/addLocationButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="158dp"  
 android:layout\_marginEnd="159dp"  
 android:layout\_marginBottom="174dp"  
 android:text="Add Location"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <Button  
 android:id="@+id/refreshButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="75dp"  
 android:layout\_marginTop="98dp"  
 android:layout\_marginEnd="16dp"  
 android:text="Refresh"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/location"  
 app:layout\_constraintTop\_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>

Again pretty simple introductory assignment.

# Week excercises

2.1 <https://github.com/jaakko13/mobileAppDev2/tree/master/flashlight>

fun switchFlashlight(view: android.view.View){  
 val cameraManager = getSystemService(*CAMERA\_SERVICE*) as CameraManager  
  
 for( id in cameraManager.*cameraIdList*){  
 if(cameraManager.getCameraCharacteristics(id).get(CameraCharacteristics.*FLASH\_INFO\_AVAILABLE*) == true){  
 torch = !torch  
 cameraManager.setTorchMode(id, torch)  
 }  
  
 val background = findViewById<ConstraintLayout>(R.id.*backgroundConstraintLayout*)  
 if(torch){  
 background.setBackgroundColor(Color.*YELLOW*)  
 }  
 else{  
 background.setBackgroundColor(Color.*BLACK*)  
 }  
 }  
   
}

Enjoyed learning the best practices of accessing the sensors. Farly easy overall but useful info. Opens the door for a lot of functionality for future apps and has already given me some ideas. No real challenges in the exercise.

2.2 <https://github.com/jaakko13/mobileAppDev2/tree/master/level>

fun leveler(view: android.view.View){  
 val sensorList = sensorManager.getSensorList(Sensor.*TYPE\_ALL*)  
  
 for( sensor in sensorList){  
 Toast.makeText(this, sensor.*name*, Toast.*LENGTH\_SHORT*).show()  
 }  
  
 val accelerometer = sensorManager.getDefaultSensor(Sensor.*TYPE\_ACCELEROMETER*)  
  
 if(accelerometer != null) {  
 sensorManager.registerListener(this, accelerometer, SensorManager.*SENSOR\_DELAY\_NORMAL*)  
 levelingStarted = true  
 }  
}

Very interesting learning how to access all the different sensor. The function above is how we got the data. But most of the visual code was in the onSensorChanged function as it updated the data on the screen. Not too complicated overall and didn’t face any big challenges.

# Week exercises

3.1

The figure shows the steps a developer should take when thinking about adding permission to their application. First step is to ask yourself if the permissions is really needed. You should not have unecessary permissions in your application. Mainly for security reasons. If possible to create functionality without the use of permissions then do so and that’s it. But if you need permissions then you should declare them in the manifest file of the app. Finally if the permission is a runtime permission then you have to request the access from the user but if not then there is nothing else to do.

if(ActivityCompat.checkSelfPermission(this, android.Manifest.permission.ACCESS\_FINE\_LOCATION) //Make sure permissions are good  
 != PackageManager.PERMISSION\_GRANTED && ActivityCompat  
 .checkSelfPermission(this, android.Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED  
){  
 ActivityCompat.requestPermissions(this, arrayOf(android.Manifest.permission.ACCESS\_FINE\_LOCATION), 101)  
 return  
}

The above piece of code is used to check the permissions and ask for them if they have not been granted yet. The initial if statement is what does the checking. While the ActivityCompat.requestPermission inside the brackets of the if statement is what requests permission from the user if the program does not have the required permissions.

3.2 Github: <https://github.com/jaakko13/mobileAppDev2/tree/master/codelab>

Getting coordinates was easy as I’ve don’t this before for a final project and also during my last internship. Worked with maps a lot… The location listener was new and a bit more of a challenge. Very useful as I had just used a button to update the location but now it updates constantly. Also opening the map was fairly simple using intent.

# Week exercises

# 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.)