Tampereen ammattikorkeakoulu

Mobile App Development 1

Study diary

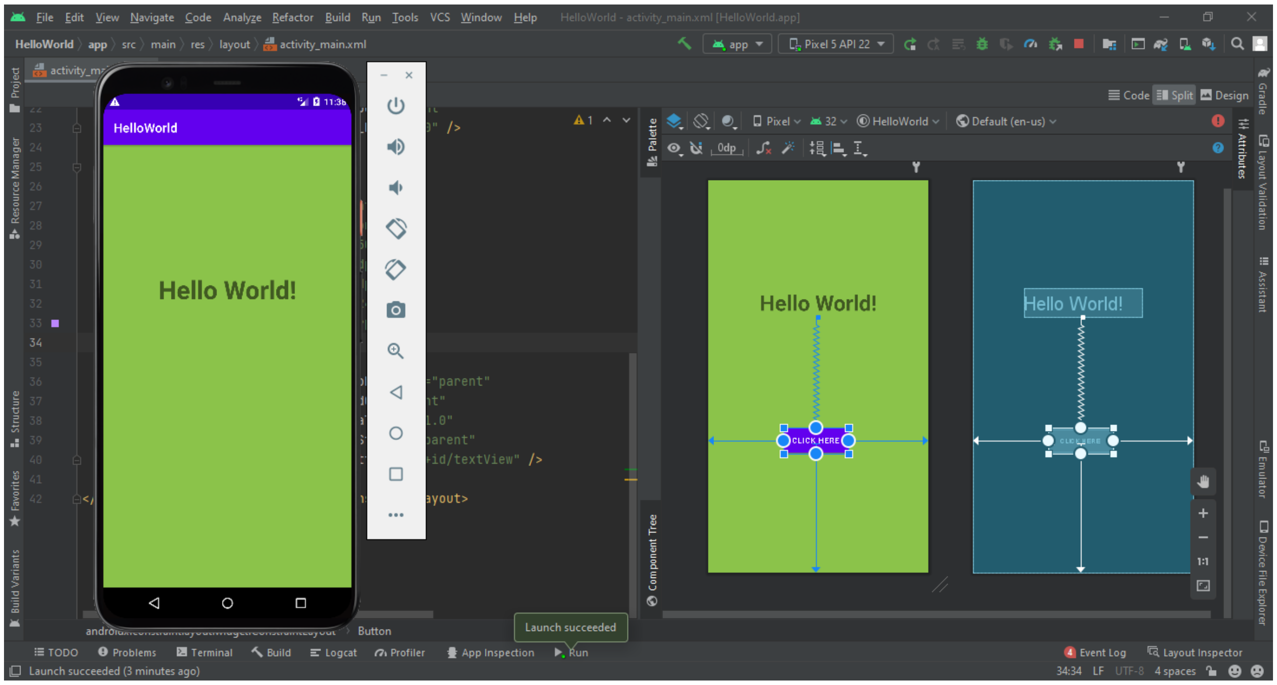
O R Imon

Contents

Week exercises - 01

1.1 Yes . I have completed my first task.

1.2 Android Studio setup and Hello World


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HelloWorld - activity_main.xml [HelloWorld.app]
HelloWorld 〉 app 〉 src 〉 main 〉 res 〉 layout 〉 activity_main.xml
app
Pixel 5 API 22
HelloWorld
Hello World!
Code Split Design
Pixel 32 HelloWorld Default (en-us)
0dp
CLICK HERE
Palette
Component Tree
Attributes
Button
Launch succeeded
TODO Problems Terminal Build Logcat Profiler App Inspection Run
Event Log Layout Inspector
Launch succeeded (3 minutes ago)
34:34 LF UTF-8 4 spaces

Code:

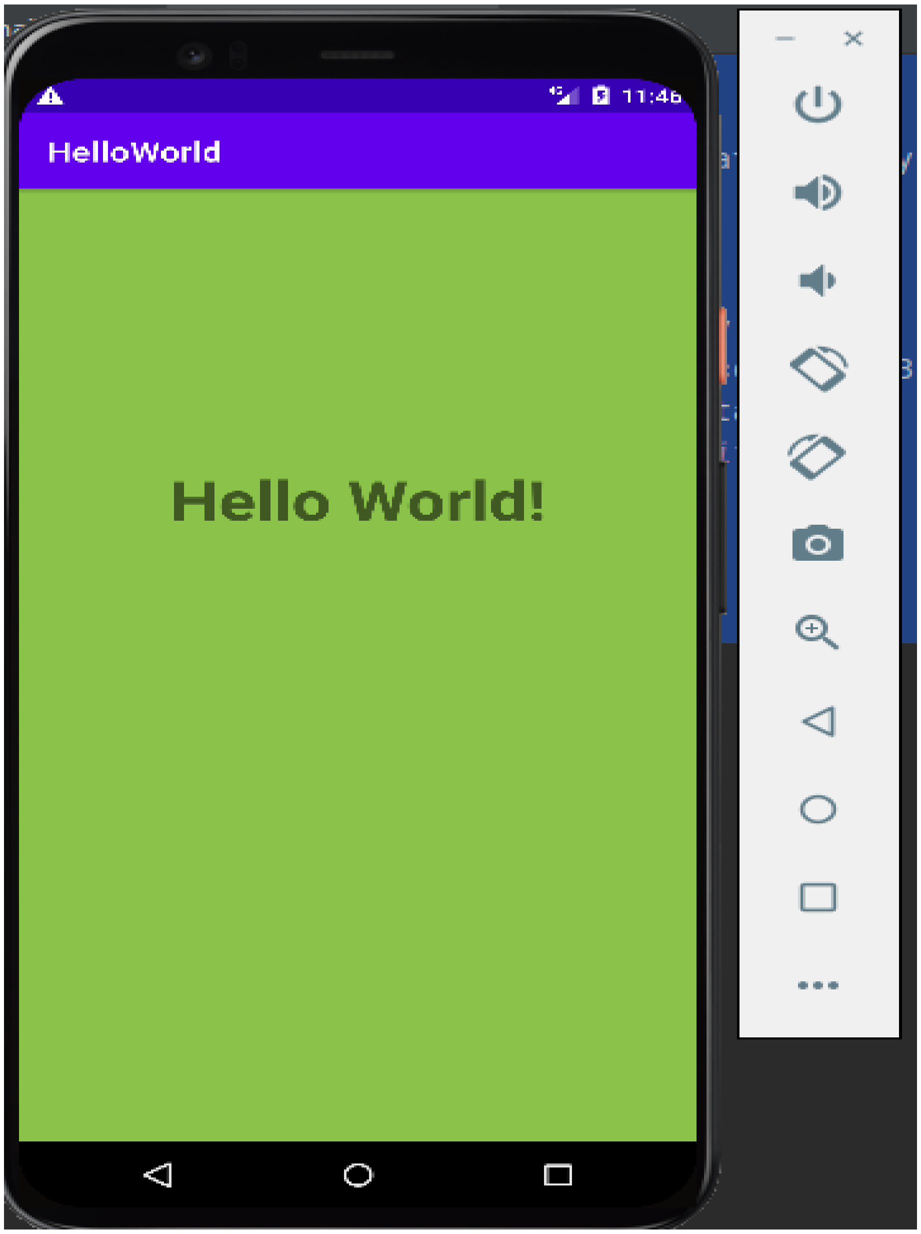
Activity_main.xml:

```
<?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="#8BC34A"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="201dp"
        android:text="Hello World!"
        android:textSize="40sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.502"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="150dp"
        android:layout_marginTop="223dp"
        android:layout_marginEnd="150dp"
        android:layout_marginBottom="240dp"
        android:background="@color/purple_200"
        android:onClick="sayHello"
        android:text="Click Here"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity:

```
package com.example.helloworld

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    fun sayHello(view: View) {}
}
```

Output:


11:46
HelloWorld
Hello World!

Exercice 2.1 What is an activity and what is the role of XML and Java files when implementing an GUI?

An activity represents a single screen with a user interface just like window or frame of Java. Android activity is the subclass of ContextThemeWrapper class.

Using XML layouts has many advantages over Java code. You get easy references to strings, drawables, dimensions, themes, etc. But the biggest advantage is automatic support for multiple configurations. Without changing your code you can have different layouts for landscape and portrait by just having an XML layout in layout-land/ and layout-port/. And you can do the same to adapt the layout to different resolutions, languages, etc.

In Java applications, the components that comprise a GUI (Graphical User Interface) are stored in containers called forms. The Java language provides a set of user interface components from which GUI forms can be built.

2.2 What is meant by an activity lifecycle? When is activity created/ closed in Android?

Activity Lifecycle: Activity is one of the building blocks of Android OS. In simple words Activity is a screen that user interact with. Every Activity in android has lifecycle like created, started, resumed, paused, stopped, or destroyed. These different states are known as Activity Lifecycle.

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.

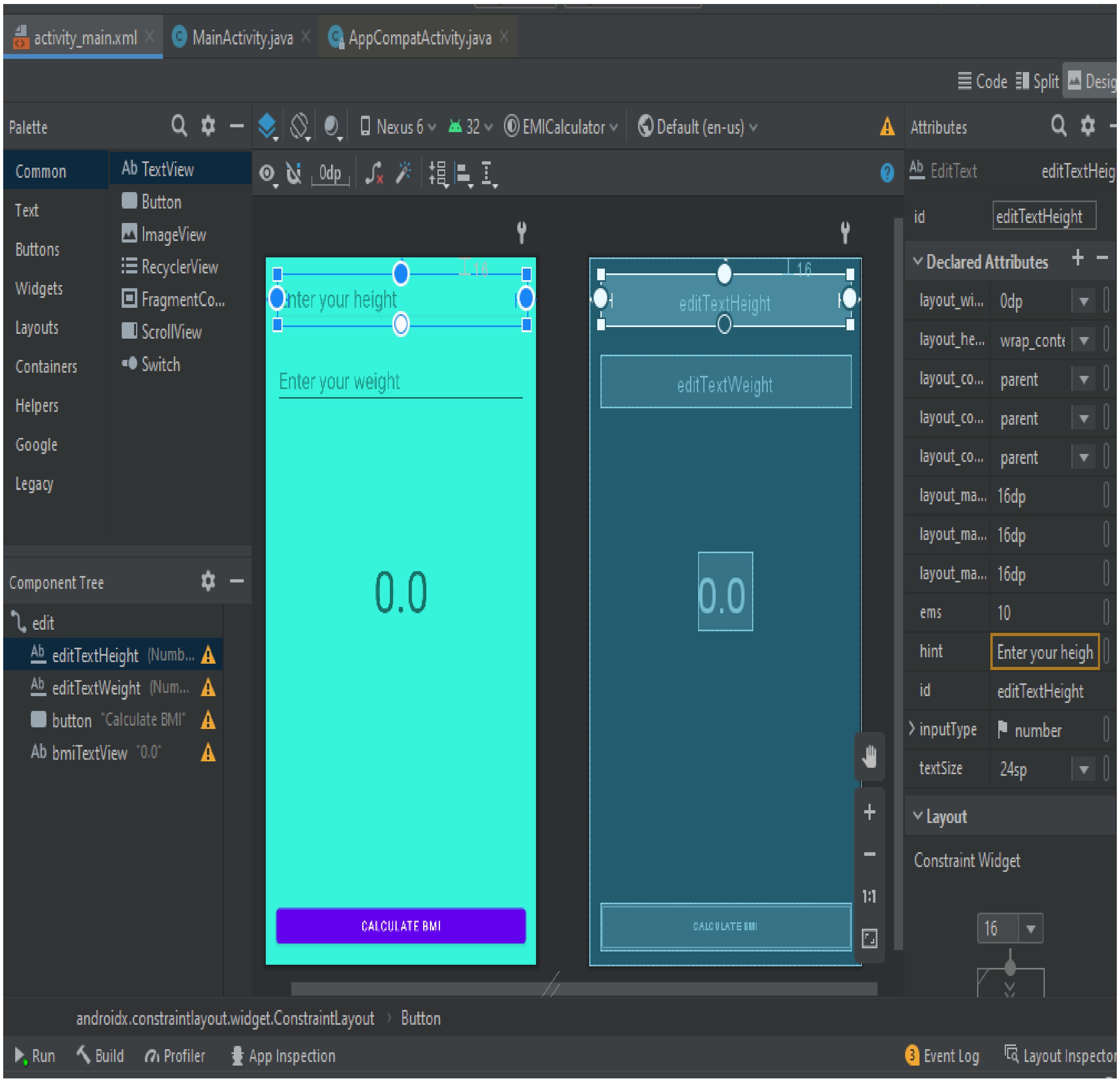
An activity represents a single screen with a user interface just like window or frame of Java. Android activity is the **subclass** of ContextThemeWrapper class. The Activity class defines the following call backs i.e. events.

I can use this.finish() if you want to close current activity.

```
this.finish()
startActivity(i);
finish();
```

2. BMI App

Activity_main.xml



id: editTextHeight

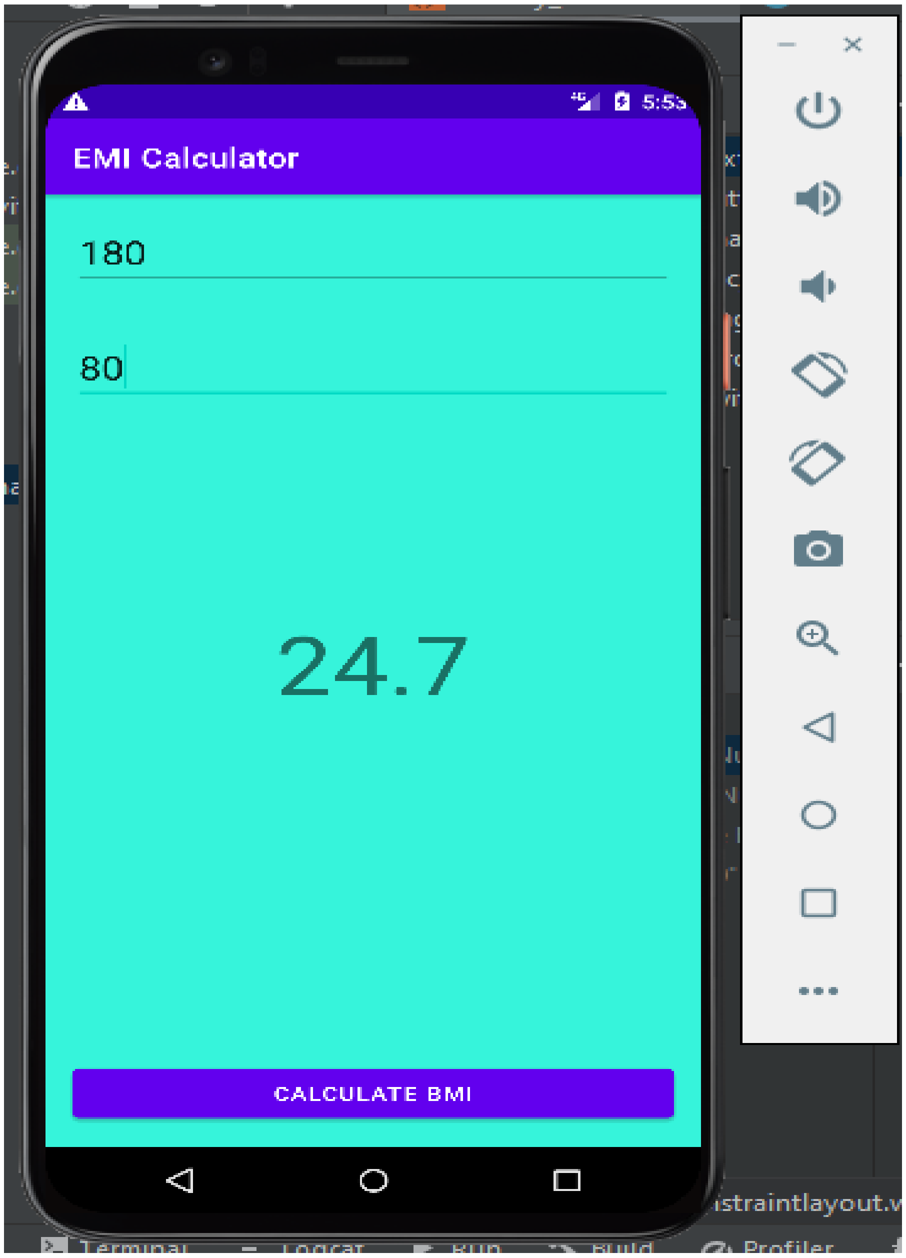
Declared Attributes

layout_wi...	0dp
layout_he...	wrap_conte
layout_co...	parent
layout_co...	parent
layout_co...	parent
layout_ma...	16dp
layout_ma...	16dp
layout_ma...	16dp
ems	10
hint	Enter your heigh
id	editTextHeight
inputType	number
textSize	24sp

MainActivity.java

```java
package com.example.emicalculator;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void calculateBMI(View v){
        //1. Read weight and height from editTexts
        EditText editTextHeight
=findViewById(R.id.editTextHeight);
        EditText editTextWeight
=findViewById(R.id.editTextWeight);

       //2. Read the input text as string (it's always string
even through it has number only)
        String
heightString=editTextHeight.getText().toString();// "180"
        String weightString=editTextWeight.getText().toString();
//"82"

       //3. convert the texts in double
        double height=Double.parseDouble(heightString);
        double weight=Double.parseDouble(weightString);

       //4. Calculate the BMI as BMI=weight /(height/100 *
height/100)

        double BMI= weight /(height/100 * height/100);

        //5. Convert the result double to string with 1 decimal
        TextView bmiTextView=findViewById(R.id.bmiTextView);
        bmiTextView.setText(String.format("%.1f",BMI));
    }

}
```

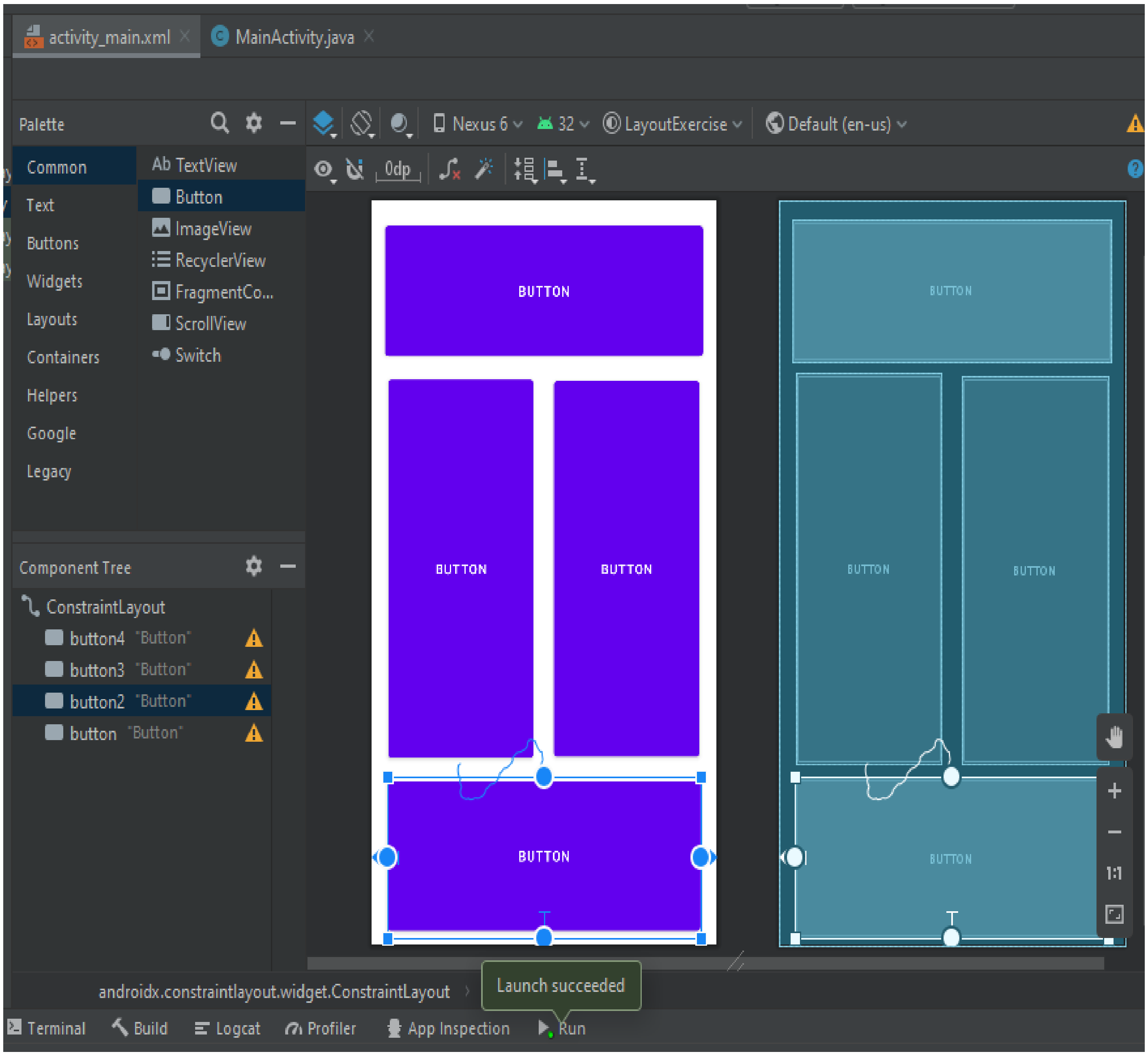
BMI Calculator:



3. Creating GUI layouts with Constraint layout

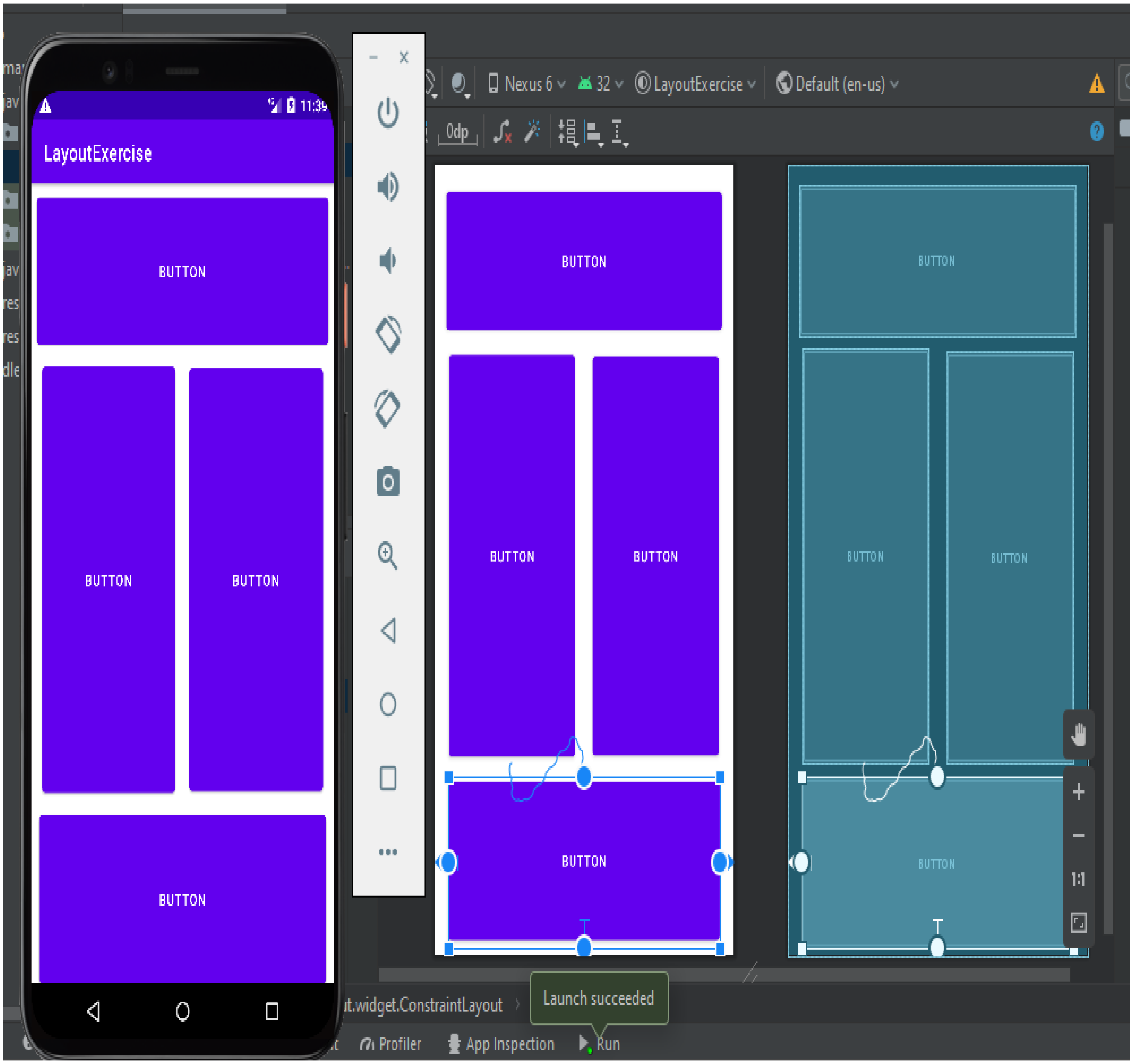
Create the following layout by using Android Designer. The elements below (blue) can be buttons.

Activity_main.xml:

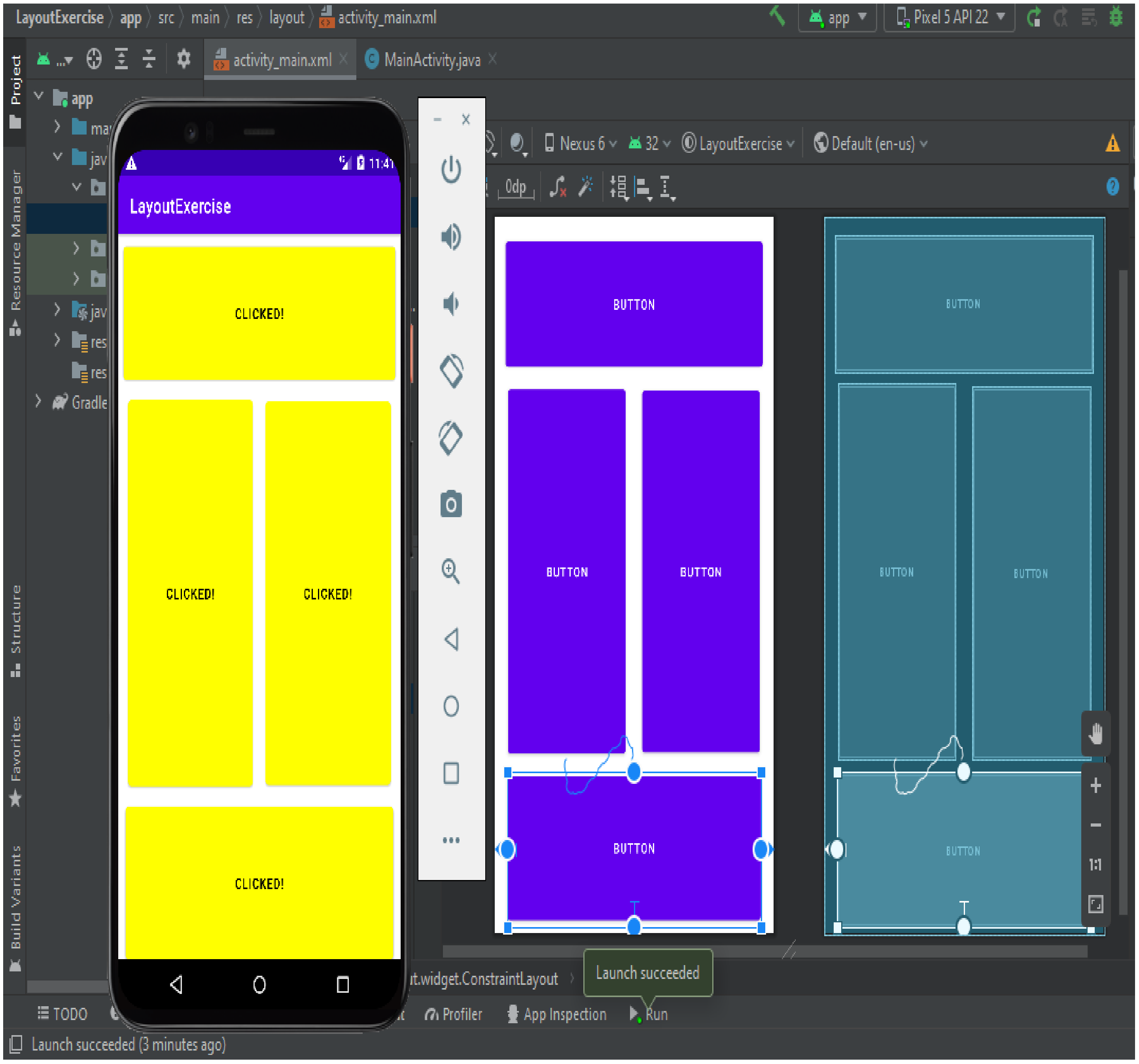


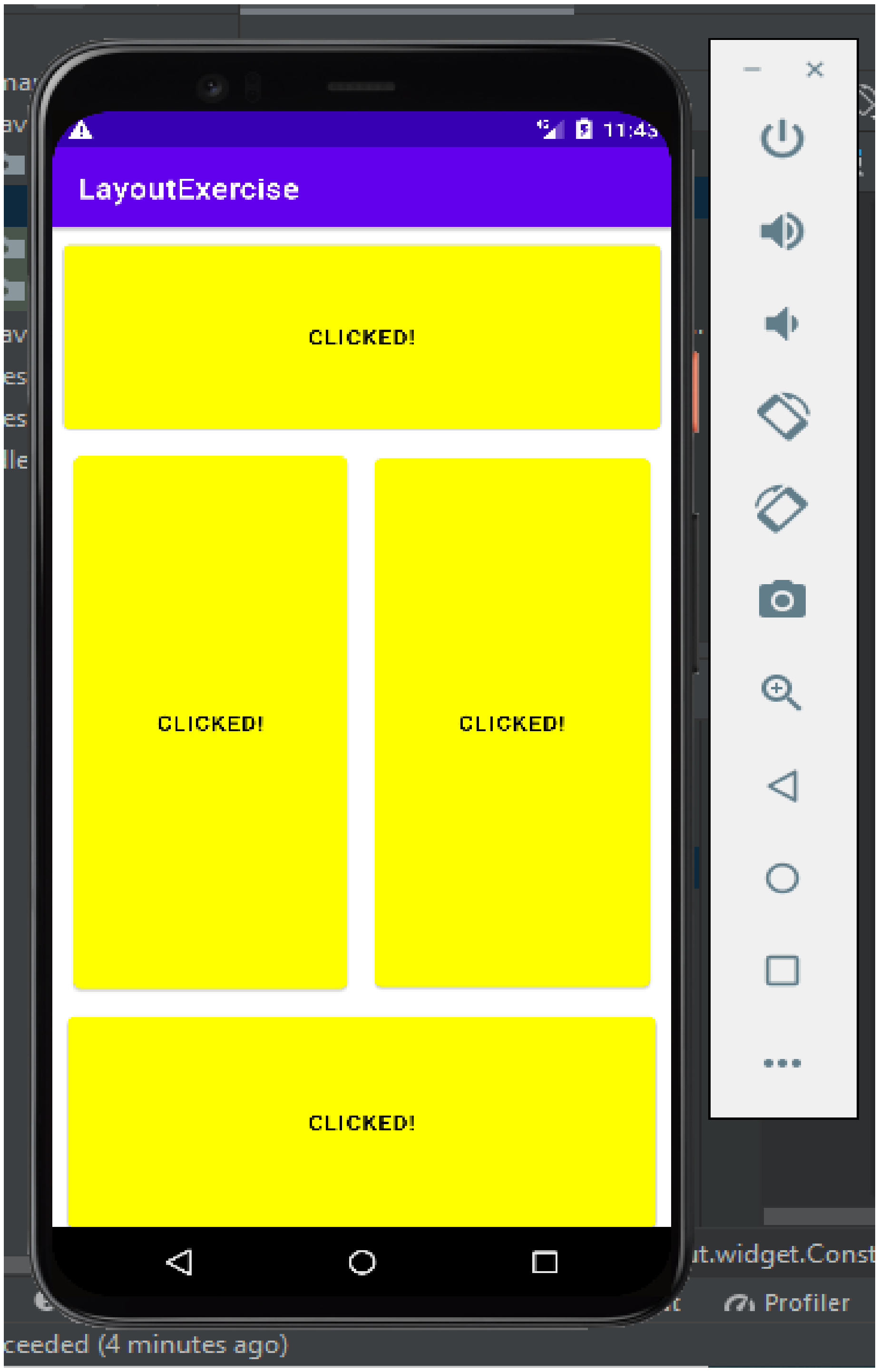
MainActivity.java

```java
package com.example.layoutexercise;
import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void changeButtonColor(View view) {
        //view is the GUI element that sent the event
        Button button=(Button)view;
        button.setBackgroundColor(Color.YELLOW);
        button.setTextColor(Color.BLACK);
        button.setText("Clicked!");
    }
}
```


LayoutExercise
BUTTON
BUTTON
BUTTON
BUTTON
Nexus 6
32
LayoutExercise
Default (en-us)
0dp
BUTTON
BUTTON
BUTTON
BUTTON
BUTTON
BUTTON
BUTTON
BUTTON
Launch succeeded
Profiler
App Inspection
Run

Implement the elements so that when clicked, they turn their color to Yellow.




11:43
LayoutExercise
CLICKED!
CLICKED!
CLICKED!
CLICKED!
Profiler
ceeded (4 minutes ago)

Exercises - 03

What advantages you see in defining user interfaces in UI resource files (.xml). instead of coding the UI (e.g. with Java)?

Read a layout XML. parsing it using XMLPullParser (Android provides three types of XML parsers which are DOM, SAX and XMLPullParser. Among all of them android recommend XMLPullParser because it is efficient and easy to use).

Making Java (or Kotlin) View object to create the UI (ie viewgroup for container and views for widget) from the parsed XML layout.

And that explains the secret behind the implemented code (the example is written in JAVA) we find when we create an Activity in Android Studio.

Separation of logic from presentation. This does help on larger projects when you need to refactor some code. If it is tightly coupled to the UI this job can become time consuming very easily if much of the presentation logic is in code.

The structure of XML correlates nicely to the structure of a user interface, i.e., a tree like structure

In Android studio, how localization and language variants are created for an Android

project?

Android is used on a wide range of devices in a variety of countries. Your app should handle text, audio files, numbers, money, and visuals in ways that are acceptable to the locations where your app is utilized to reach the most customers.

This document explains how to localize Android apps in the best way possible.

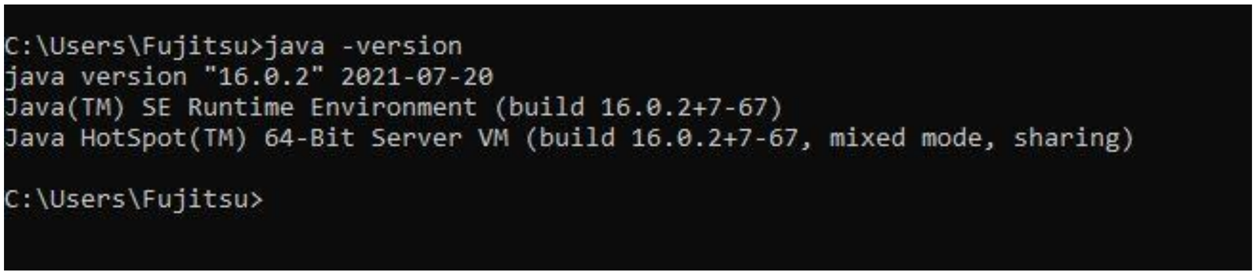
You should have a working knowledge of either Kotlin or Java, as well as knowledge of Android resource loading, XML declaration of user interface elements, development issues such as activity lifecycle, and general internationalization and localization concepts.

Use of resource files and localizing your app

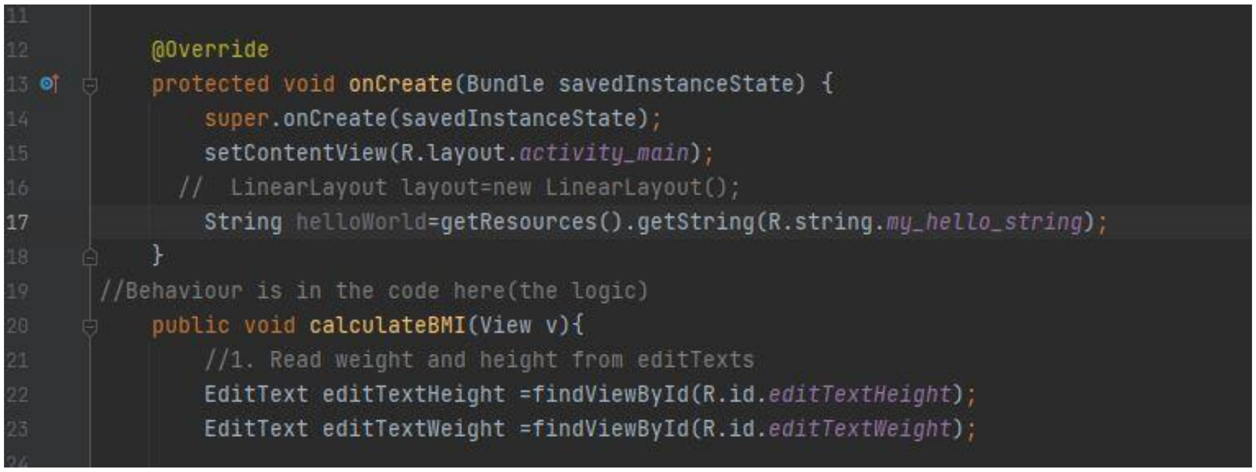
Android is used on a wide range of devices in a variety of countries. Your app should handle text, audio files, numbers, money, and visuals in ways that are acceptable to the locations where your app is utilized to reach the most customers.

This document explains how to localize Android apps in the best way possible.

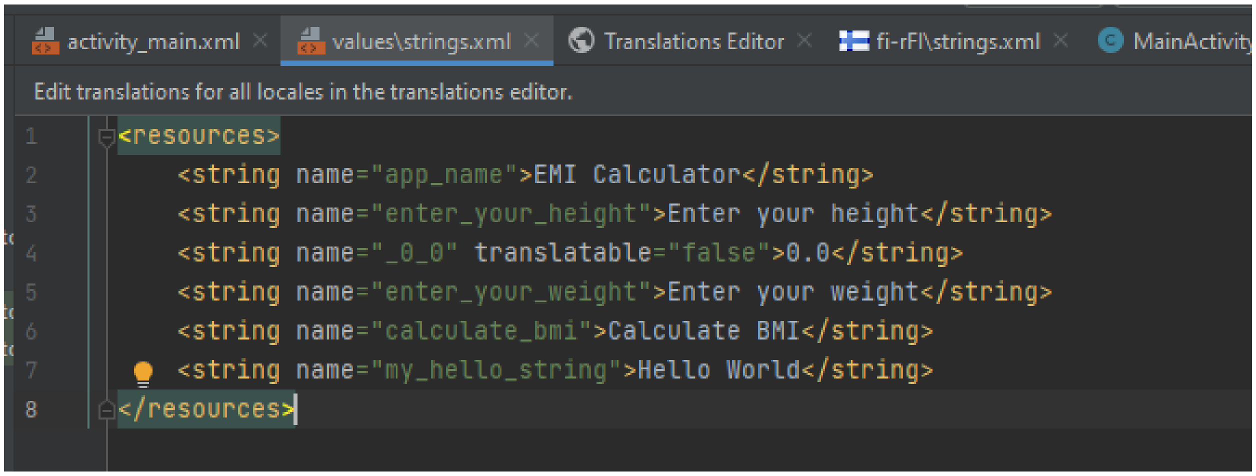
You should have a working knowledge of either Kotlin or Java, as well as knowledge of Android resource loading, XML declaration of user interface elements, development issues such as activity lifecycle, and general internationalization and localization concepts.



```
C:\Users\Fujitsu>java -version
java version "16.0.2" 2021-07-20
Java(TM) SE Runtime Environment (build 16.0.2+7-67)
Java HotSpot(TM) 64-Bit Server VM (build 16.0.2+7-67, mixed mode, sharing)

C:\Users\Fujitsu>
```

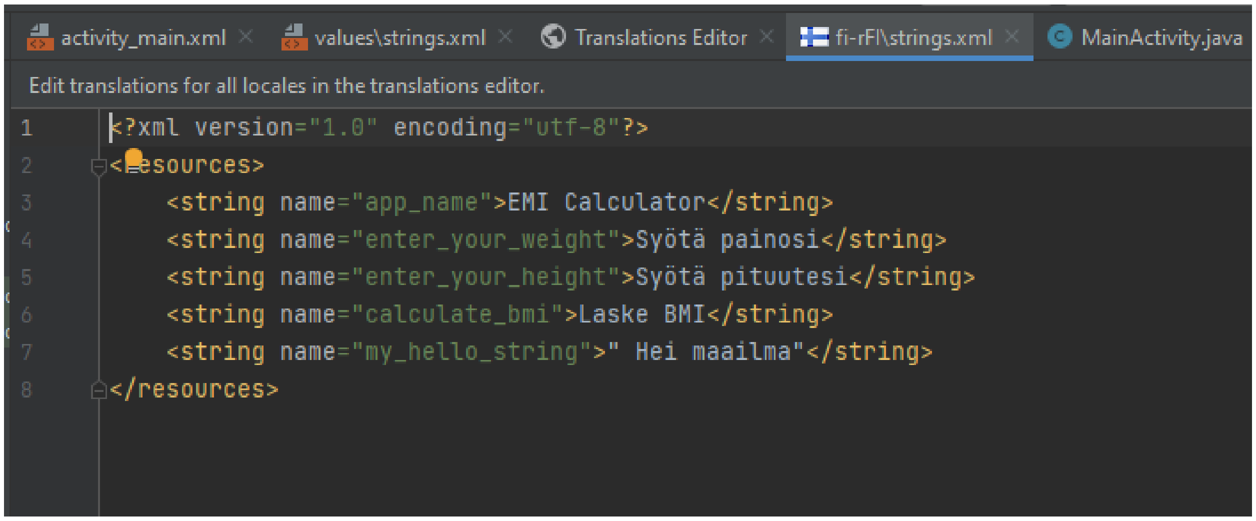


```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
      //  LinearLayout layout=new LinearLayout();
        String helloWorld=getResources().getString(R.string.my_hello_string);
    }
//Behaviour is in the code here(the logic)
    public void calculateBMI(View v){
        //1. Read weight and height from editTexts
        EditText editTextHeight =findViewById(R.id.editTextHeight);
        EditText editTextWeight =findViewById(R.id.editTextWeight);
```


activity_main.xml | values\strings.xml | Translations Editor | fi-rFI\strings.xml | MainActivity

Edit translations for all locales in the translations editor.

```
<resources>
    <string name="app_name">EMI Calculator</string>
    <string name="enter_your_height">Enter your height</string>
    <string name="_0_0" translatable="false">0.0</string>
    <string name="enter_your_weight">Enter your weight</string>
    <string name="calculate_bmi">Calculate BMI</string>
    <string name="my_hello_string">Hello World</string>
</resources>
```


activity_main.xml | values\strings.xml | Translations Editor | fi-rFI\strings.xml | MainActivity.java

Edit translations for all locales in the translations editor.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">EMI Calculator</string>
    <string name="enter_your_weight">Syötä painosi</string>
    <string name="enter_your_height">Syötä pituutesi</string>
    <string name="calculate_bmi">Laske BMI</string>
    <string name="my_hello_string">" Hei maailma"</string>
</resources>
```

MainActivity.java

```
package com.example.emicalculator;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
      //  LinearLayout layout=new LinearLayout();
        String
helloWorld=getResources().getString(R.string.my_hello_string);
    }
//Behaviour is in the code here(the logic)
    public void calculateBMI(View v){
        //1. Read weight and height from editTexts
        EditText editTextHeight
=findViewById(R.id.editTextHeight);
        EditText editTextWeight
=findViewById(R.id.editTextWeight);

       //2. Read the input text as string (it's always string
even through it has number only)
        String
heightString=editTextHeight.getText().toString();// "180"
        String weightString=editTextWeight.getText().toString();
//"82"

       //3. convert the texts in double
        double height=Double.parseDouble(heightString);
        double weight=Double.parseDouble(weightString);

       //4. Calculate the BMI as BMI=weight /(height/100 *
height/100)

        double BMI= weight /(height/100 * height/100);

        //5. Convert the result double to string with 1 decimal
        TextView bmiTextView=findViewById(R.id.bmiTextView);
        bmiTextView.setText(String.format("%.1f",BMI));



    }


}
```

Exercise 04

1.

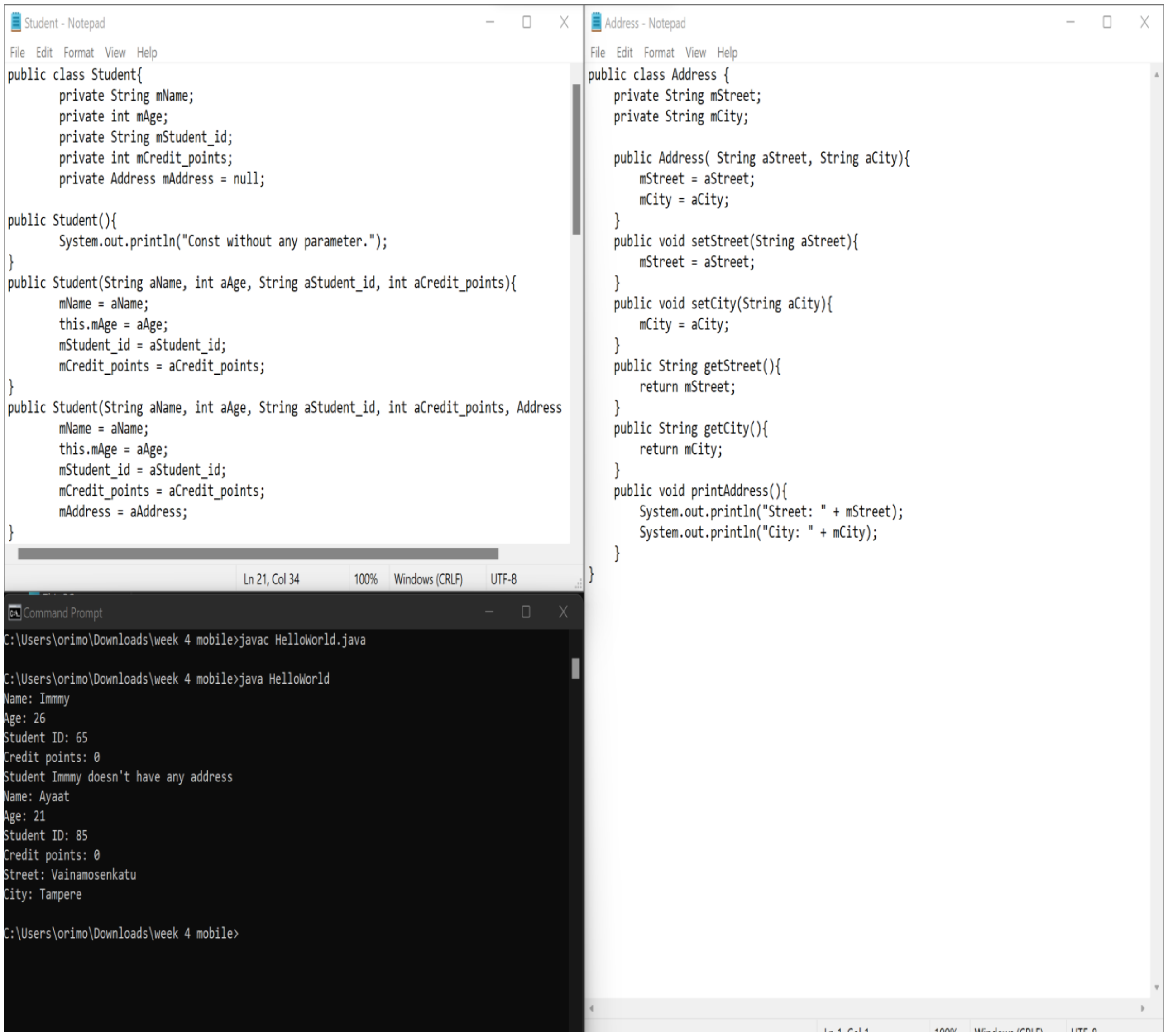
Add Address to a student. Create a new class Address. Address should have 2 member variables (and appropriate constructors, getters, and setters.

1.Street Address (String)

2. City (String)

Add a new 4 parameter constructor for the student class the parameters should be name, age, student ID and address, which will be initialized when student is created.

Create some students in your main method. Some students might have an address, some students don't so use.


Student - Notepad
File Edit Format View Help
Ln 21, Col 34 100% Windows (CRLF) UTF-8
Address - Notepad
File Edit Format View Help
Command Prompt

```
public class Student{
        private String mName;
        private int mAge;
        private String mStudent_id;
        private int mCredit_points;
        private Address mAddress = null;

public Student(){
        System.out.println("Const without any parameter.");
}
public Student(String aName, int aAge, String aStudent_id, int aCredit_points){
        mName = aName;
        this.mAge = aAge;
        mStudent_id = aStudent_id;
        mCredit_points = aCredit_points;
}
public Student(String aName, int aAge, String aStudent_id, int aCredit_points, Address
        mName = aName;
        this.mAge = aAge;
        mStudent_id = aStudent_id;
        mCredit_points = aCredit_points;
        mAddress = aAddress;
}
```

```
public class Address {
    private String mStreet;
    private String mCity;

    public Address( String aStreet, String aCity){
        mStreet = aStreet;
        mCity = aCity;
    }
    public void setStreet(String aStreet){
        mStreet = aStreet;
    }
    public void setCity(String aCity){
        mCity = aCity;
    }
    public String getStreet(){
        return mStreet;
    }
    public String getCity(){
        return mCity;
    }
    public void printAddress(){
        System.out.println("Street: " + mStreet);
        System.out.println("City: " + mCity);
    }
}
```

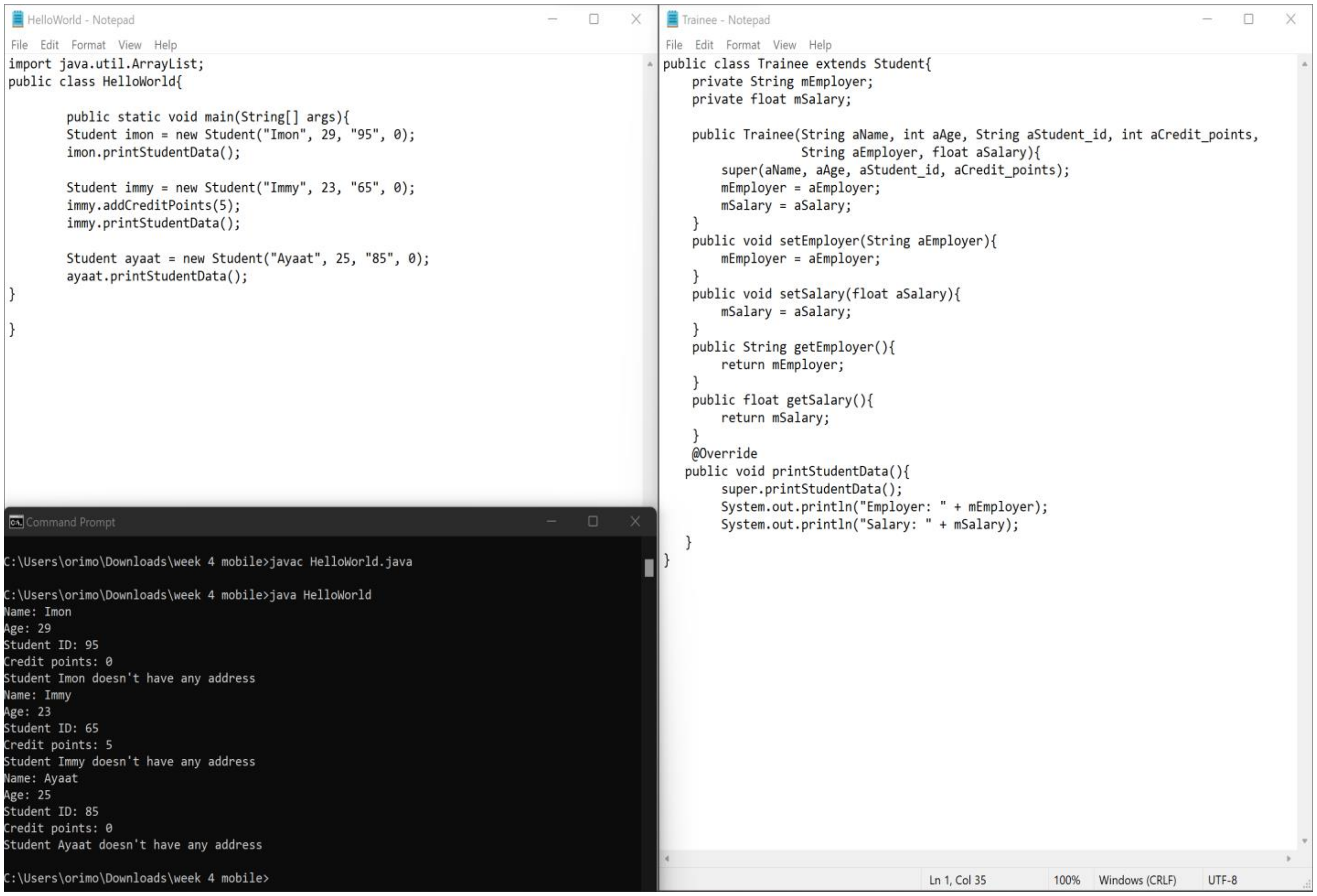
```
C:\Users\orimo\Downloads\week 4 mobile>javac HelloWorld.java

C:\Users\orimo\Downloads\week 4 mobile>java HelloWorld
Name: Immmy
Age: 26
Student ID: 65
Credit points: 0
Student Immmy doesn't have any address
Name: Ayaat
Age: 21
Student ID: 85
Credit points: 0
Street: Vainamosenkatu
City: Tampere

C:\Users\orimo\Downloads\week 4 mobile>
```

2.

Extend student class to create a new class “Trainee”. Trainee is a student who has a employer (String) and a salary (float). To Trainee class, implement constructor, where you can enter employer and salary (and other student data like name, age, student ID – address not necessary).

• Implement getters and setters for the salary and employer info.

• Override printStudentData in your Trainee class so that it prints all student information including employer and salary.



HelloWorld - Notepad

```java
import java.util.ArrayList;
public class HelloWorld{

        public static void main(String[] args){
        Student imon = new Student("Imon", 29, "95", 0);
        imon.printStudentData();

        Student immy = new Student("Immy", 23, "65", 0);
        immy.addCreditPoints(5);
        immy.printStudentData();

        Student ayaat = new Student("Ayaat", 25, "85", 0);
        ayaat.printStudentData();
}

}
```

Command Prompt

```
C:\Users\orimo\Downloads\week 4 mobile>javac HelloWorld.java

C:\Users\orimo\Downloads\week 4 mobile>java HelloWorld
Name: Imon
Age: 29
Student ID: 95
Credit points: 0
Student Imon doesn't have any address
Name: Immy
Age: 23
Student ID: 65
Credit points: 5
Student Immy doesn't have any address
Name: Ayaat
Age: 25
Student ID: 85
Credit points: 0
Student Ayaat doesn't have any address

C:\Users\orimo\Downloads\week 4 mobile>
```

Trainee - Notepad

```java
public class Trainee extends Student{
    private String mEmployer;
    private float mSalary;

    public Trainee(String aName, int aAge, String aStudent_id, int aCredit_points,
                   String aEmployer, float aSalary){
        super(aName, aAge, aStudent_id, aCredit_points);
        mEmployer = aEmployer;
        mSalary = aSalary;
    }
    public void setEmployer(String aEmployer){
        mEmployer = aEmployer;
    }
    public void setSalary(float aSalary){
        mSalary = aSalary;
    }
    public String getEmployer(){
        return mEmployer;
    }
    public float getSalary(){
        return mSalary;
    }
    @Override
   public void printStudentData(){
        super.printStudentData();
        System.out.println("Employer: " + mEmployer);
        System.out.println("Salary: " + mSalary);
   }
}
```

3.

In your main method, create an arraylist (dynamic array) of students:

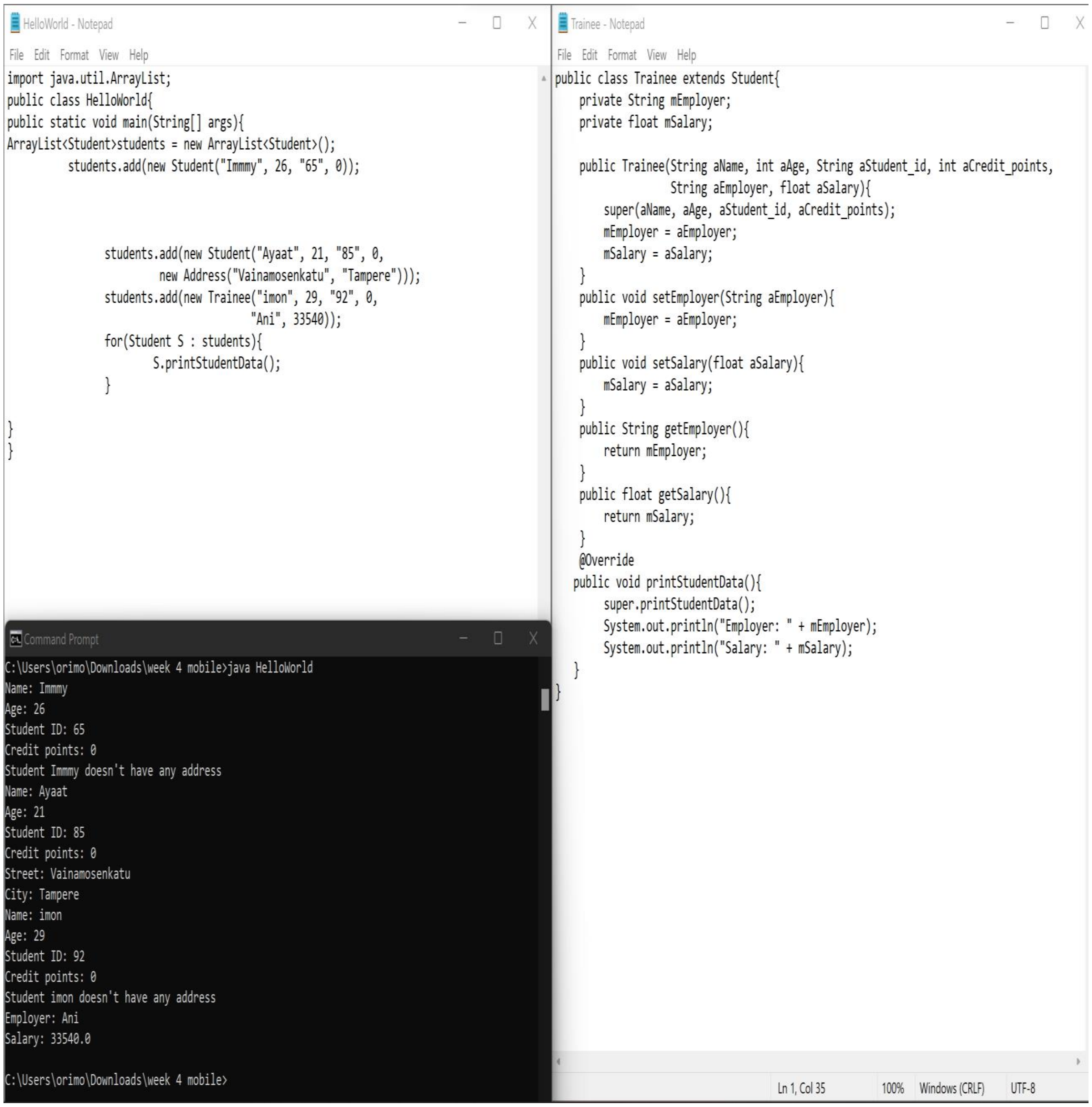
ArrayList<Student> students = new ArrayList<Student>();

Add some random Students and Trainees to the array list

students.add(new Student(“Kalle”, 20, “12345”));

students.add(new Trainee(“Maija”, 22, “11122”, “Solita”, 2500));

Write a for loop, where you print all their data by calling printStudentData method.



```java
import java.util.ArrayList;
public class HelloWorld{
public static void main(String[] args){
ArrayList<Student>students = new ArrayList<Student>();
          students.add(new Student("Immy", 26, "65", 0));



               students.add(new Student("Ayaat", 21, "85", 0,
                        new Address("Vainamosenkatu", "Tampere")));
               students.add(new Trainee("imon", 29, "92", 0,
                                    "Ani", 33540));
               for(Student S : students){
                       S.printStudentData();
               }

}
}
```

```
C:\Users\orimo\Downloads\week 4 mobile>java HelloWorld
Name: Immy
Age: 26
Student ID: 65
Credit points: 0
Student Immy doesn't have any address
Name: Ayaat
Age: 21
Student ID: 85
Credit points: 0
Street: Vainamosenkatu
City: Tampere
Name: imon
Age: 29
Student ID: 92
Credit points: 0
Student imon doesn't have any address
Employer: Ani
Salary: 33540.0

C:\Users\orimo\Downloads\week 4 mobile>
```

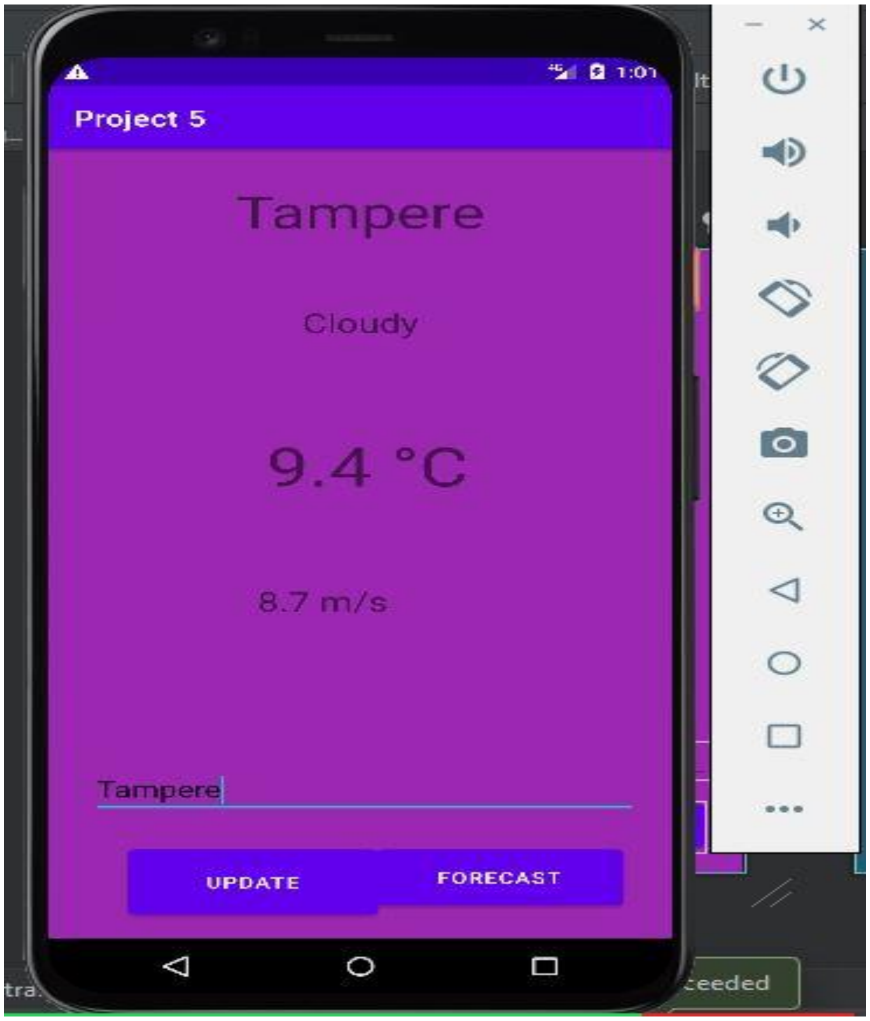
```java
public class Trainee extends Student{
    private String mEmployer;
    private float mSalary;

    public Trainee(String aName, int aAge, String aStudent_id, int aCredit_points,
                   String aEmployer, float aSalary){
        super(aName, aAge, aStudent_id, aCredit_points);
        mEmployer = aEmployer;
        mSalary = aSalary;
    }
    public void setEmployer(String aEmployer){
        mEmployer = aEmployer;
    }
    public void setSalary(float aSalary){
        mSalary = aSalary;
    }
    public String getEmployer(){
        return mEmployer;
    }
    public float getSalary(){
        return mSalary;
    }
    @Override
   public void printStudentData(){
        super.printStudentData();
        System.out.println("Employer: " + mEmployer);
        System.out.println("Salary: " + mSalary);
   }
}
```

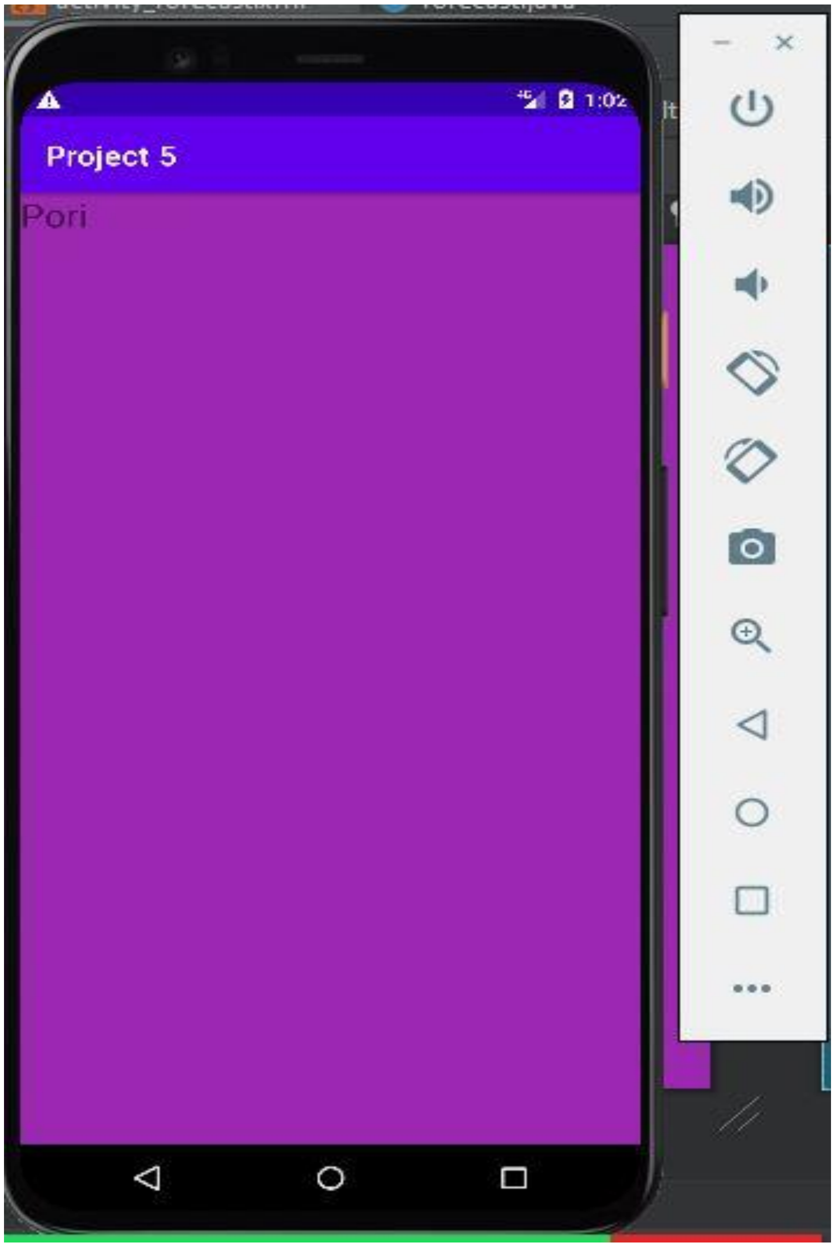
Exercise 05

Implement a simple App GUI with 2 screens and some data passing between the screens (data from 1st activity to 2nd activity). You can choose your own example app but here is an example:

• Weather app showing weather conditions like temperature, wind and overall weather in the 1st screen. Then another screen for weather forecast. The 1st screen should contain a button, which upon clicking, opens the 2nd screen with an intent.

• Add some data to the 2nd screen by sending it via the Intent from the 1st screen. Show the data in the 2nd screen in some Widget, e.g. TextView.

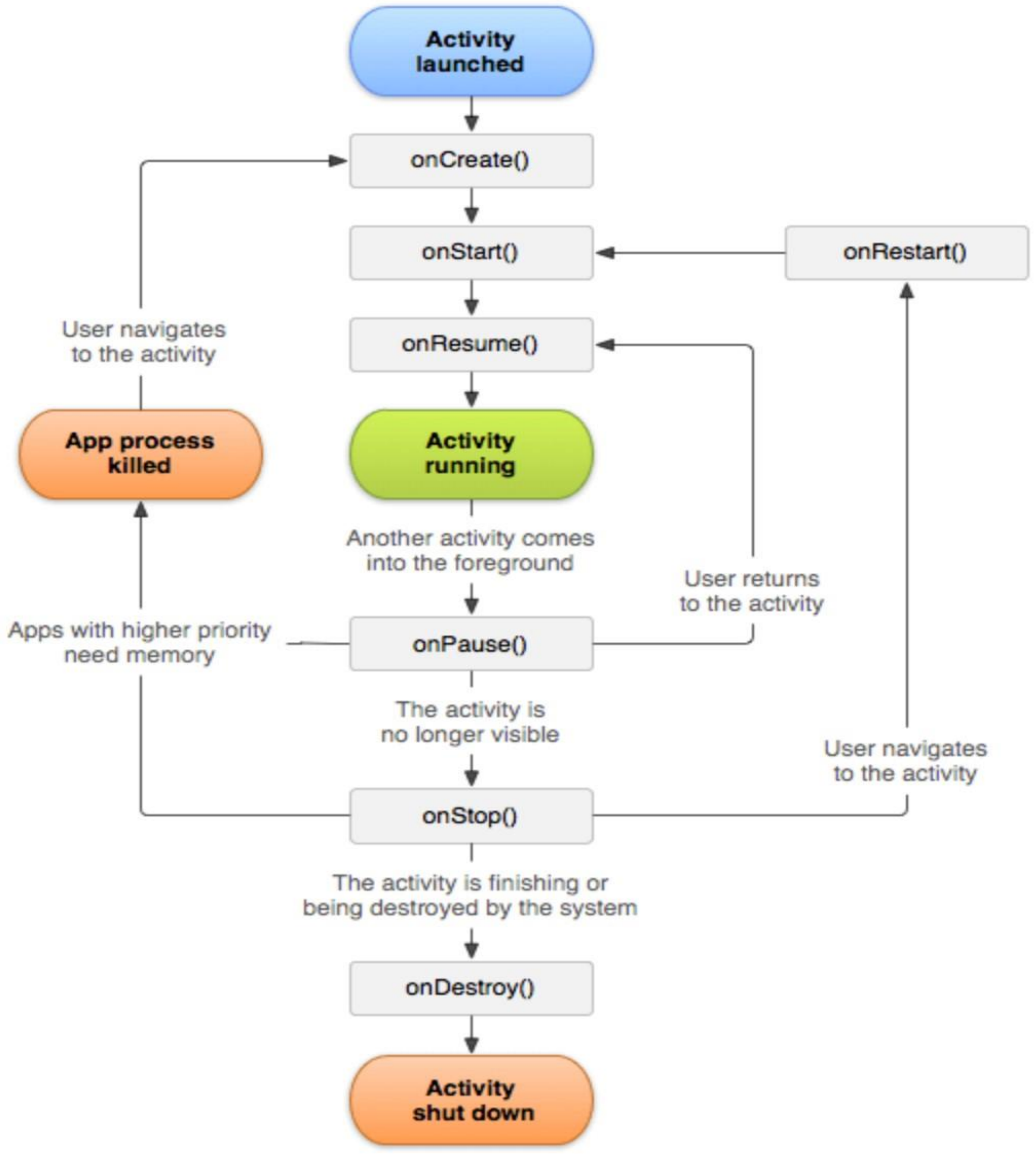



1:02
Project 5
Pori

Explain what an Activity is and explain the activity lifecycle.

Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class. The android Activity is the subclass of ContextThemeWrapper class. An activity is the single screen in android. It is like window or frame of Java. By the help of activity, you can place all your UI components or widgets in a single screen.

The 7-lifecycle method of Activity describes how activity will behave at different states.


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

• What is an Intent in Android? How are intents used to switch between application screens? How do you pass data between the activities with intents?

An intent is a message object that you can use to ask another app component to perform a task. Although intentions help components communicate in a variety of ways, there are three main applications: Begin a new activity. A single screen in an app is represented by an Activity.

I'm new in the android develop world. I created simple application and created a simple GUI with one button. If the user presses this button, I want to change the screen to show some other GUI.

When making your Android apps you may get to the point where you need to create multiple Activities for different screens within your app. When you have multiple Activities, you will need a way to transition between them in a way that doesn't break the back button functionality.

When making your Android apps you may get to the point where you need to create multiple Activities for different screens within your app. When you have multiple Activities, you will need a way to transition between them in a way that doesn't break the back button functionality.

To switch between Activities in Android you will need to follow these steps:

- Create the Activities
- Add the Activities to the app's Manifest
- Create an Intent referencing the Activity class you want to switch to
- Call the startActivity(Intent) method to switch to the Activity
- Create a back button on the new Activity and call the finish() method on an Activity when the back button is pressed

Exercises 6

Explain options and procedure when you need to get REST data from a HTTP server.

In REST OPTIONS is a method level annotation, this annotation indicates that the following method will respond to the HTTP OPTIONS request only. It is used to request, for information about the communication option available for a resource.

This method allows the client of the REST API to determine, which HTTP method (GET, HEAD, POST, PUT, DELETE) can be used for a resource identified by requested URI, without initiating a resource request by using any particular HTTP method. Response to this method are not cacheable.

A 200 OK response should include any header fields like Allow, that determine the communication option implemented by the server and are applicable for a resource identified by requested URI. The response body (if any) should also include information about the communication option available for a resource. The format for such a body is not defined by the specification, but might be defined by future extensions to HTTP. If response body is not included then value of Content-Length header field should be 0.

Example :-

200 OK

Allow: HEAD, GET, PUT, DELETE,OPTIONS

If request URI is an asterisk sign (*), then the OPTIONS method request applies to the server rather than to a specific resource. It can be used as a ping to server for checking its capabilities.

REST OPTIONS method is also used for CORS (Cross-Origin Resource Sharing) request.

What are the options for that in Android? How is fetching done with Android Volley Library?

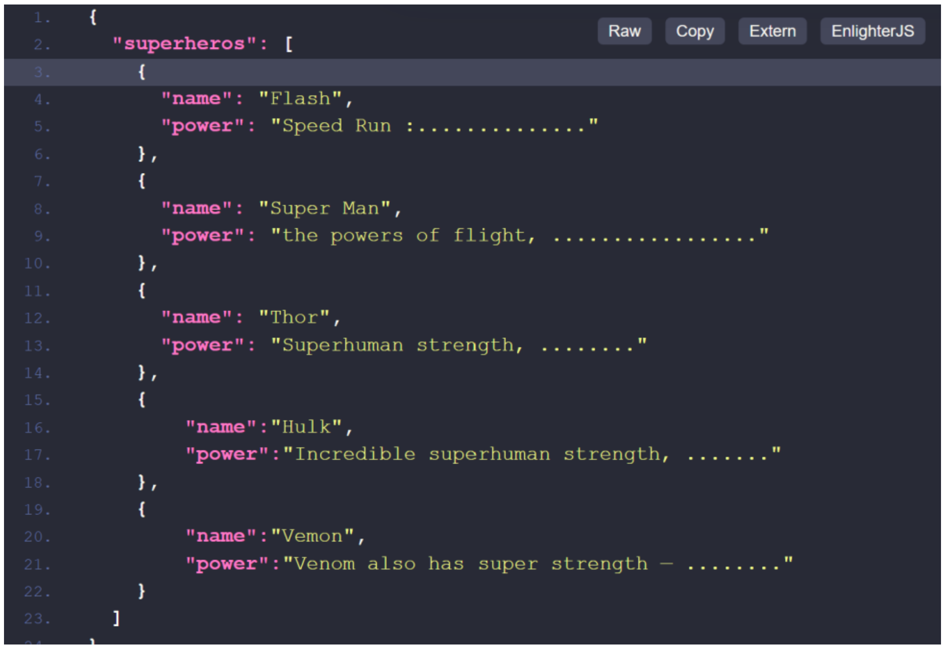
Volley is an HTTP library that makes networking very easy and fast, for Android application development.

Now it's time for implement library, As i have told you above that i have a URL that contain JSON Data Format.

https://protocoderspoint.com/jsondata/superheros.json

Then if you are implementing this project then you can use same URL for your project Practice or Create your own JSON Data.

If you open the above URL, you will see the following JSON DATA .



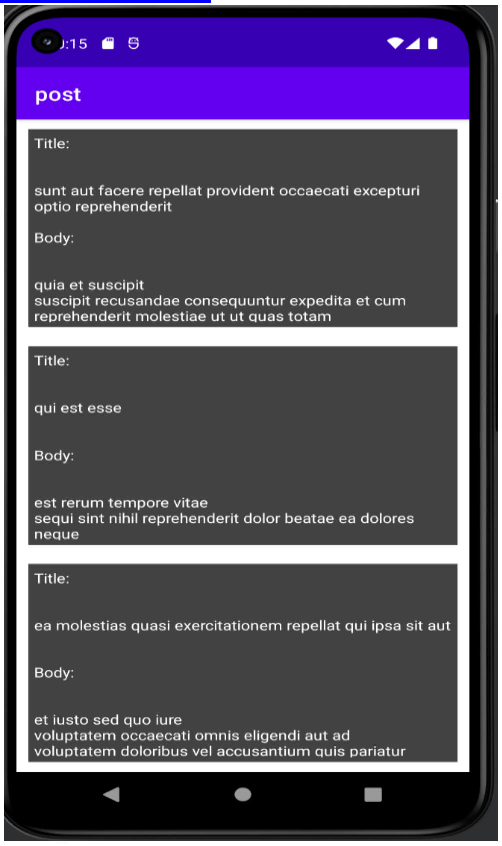
```
{
  "superheros": [
    {
      "name": "Flash",
      "power": "Speed Run :.............."
    },
    {
      "name": "Super Man",
      "power": "the powers of flight, ................."
    },
    {
      "name": "Thor",
      "power": "Superhuman strength, ........"
    },
    {
        "name":"Hulk",
        "power":"Incredible superhuman strength, ......."
    },
    {
        "name":"Vemon",
        "power":"Venom also has super strength — ........"
    }
  ]
```

How is JSON parsed in Android/Java. Explain the API for that.

JSON (JavaScript Object Notation) is a straightforward data exchange format to interchange the server's data, and it is a better alternative for XML. This is because JSON is a lightweight and structured language. Android supports all the JSON classes such as JSONStringer, JSONObject, JSONArray, and all other forms to parse the JSON data and fetch the required information by the program. JSON's main advantage is that it is a language-independent, and the JSON object will contain data like a key/value pair. In general, JSON nodes will start with a square bracket ([) or with a curly bracket ({). The square and curly bracket's primary difference is that the square bracket ([) represents the beginning of a JSONArray node. Whereas, the curly bracket ({) represents a JSONObject. So one needs to call the appropriate method to get the data. Sometimes JSON data start with [. We then need to use the getJSONArray() method to get the data. Similarly, if it starts with {, then we need to use the getJSONobject() method. The syntax of the JSON file is as following:

```
{
"Name": "Imon",
"Estd": 2022,
"age": 10,
"address": {
  "buildingAddress": "5th & 6th Floor Royal Kapsons, A- 118",
  "city": "Sector- 136, Finland",
  "state": "Tampere (111222)",
  "postalCode": "33540"
},
```

I use jsonplaceholder in this site. JSONPlaceholder comes with a set of resources. I use in the posts, resources have relations. For example: posts have many comments. I can use http or https https://jsonplaceholder.typicode.com/All HTTP methods are supported for your requests.



Title:

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

Body:

quia et suscipit
suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam

Title:

qui est esse

Body:

est rerum tempore vitae
sequi sint nihil reprehenderit dolor beatae ea dolores neque

Title:

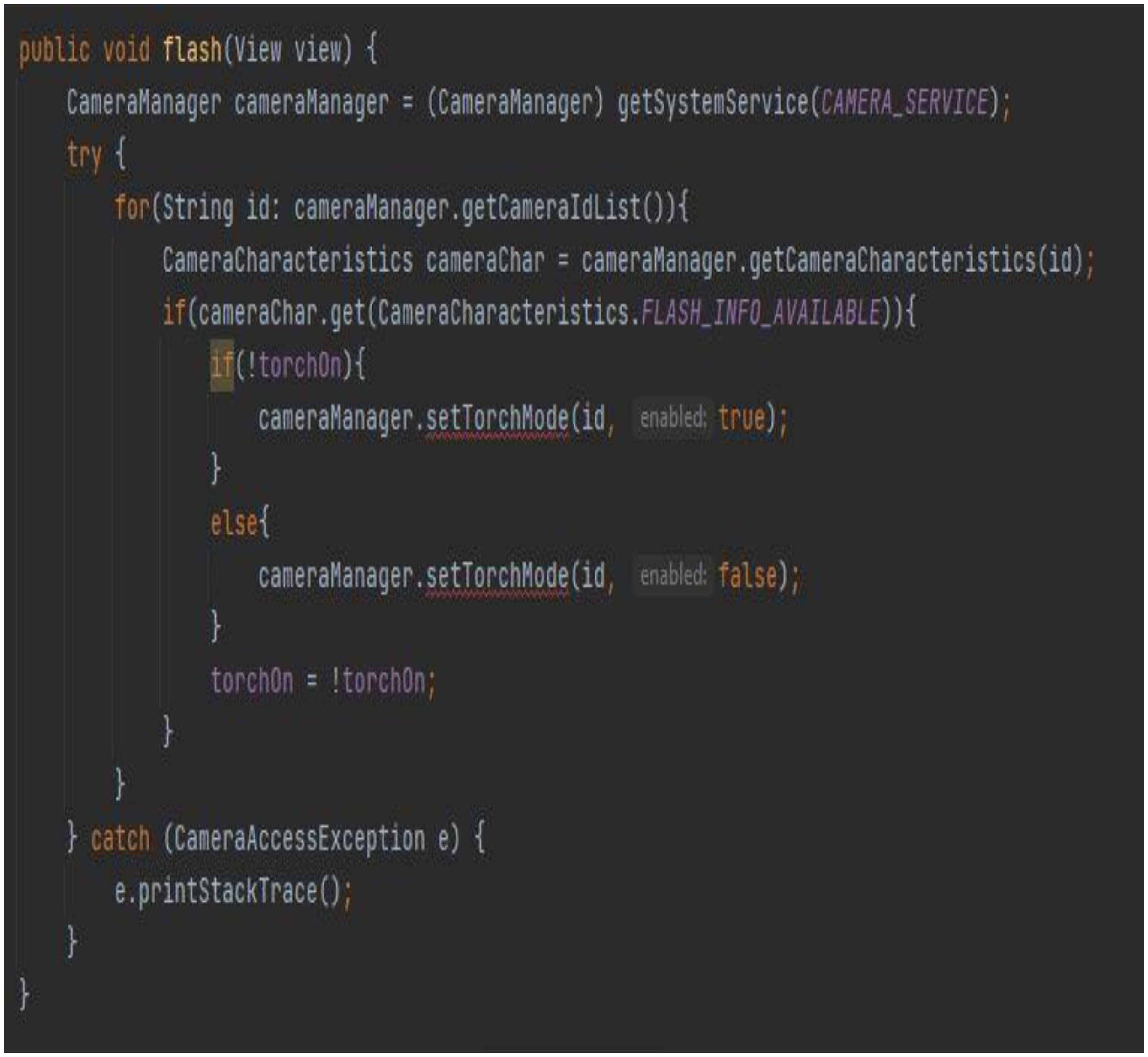
ea molestias quasi exercitationem repellat qui ipsa sit aut

Body:

et iusto sed quo iure
voluptatem occaecati omnis eligendi aut ad
voluptatem doloribus vel accusantium quis pariatur

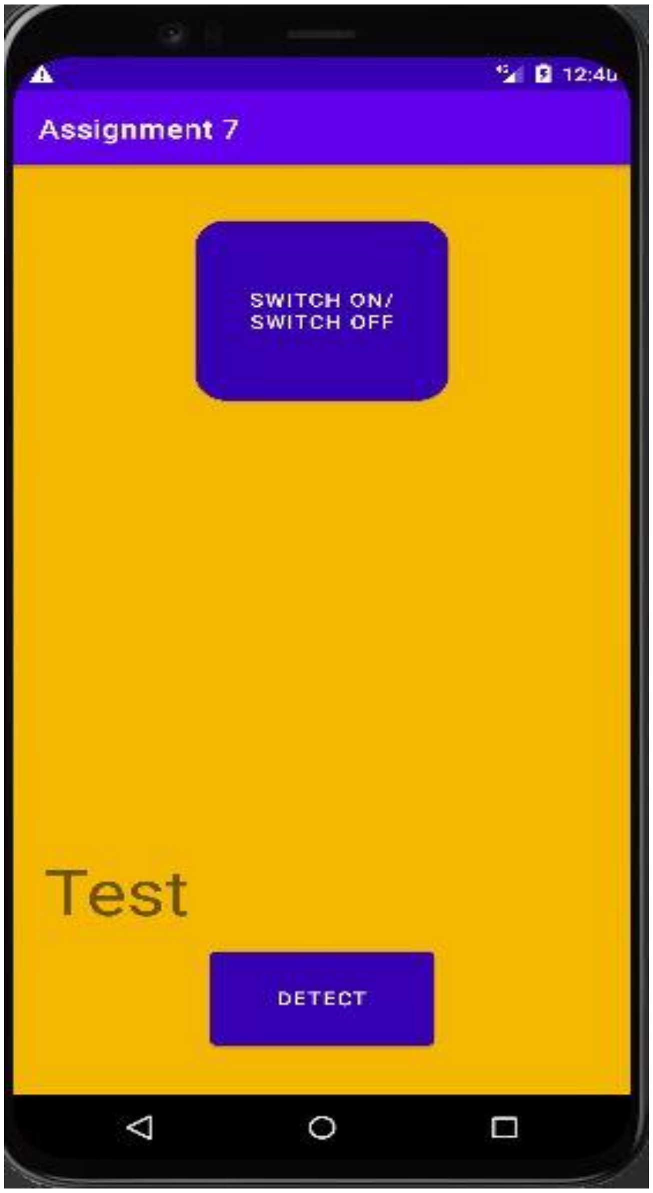
Exercises 7

Synchronous device API Example – Camera HW and "torch app"

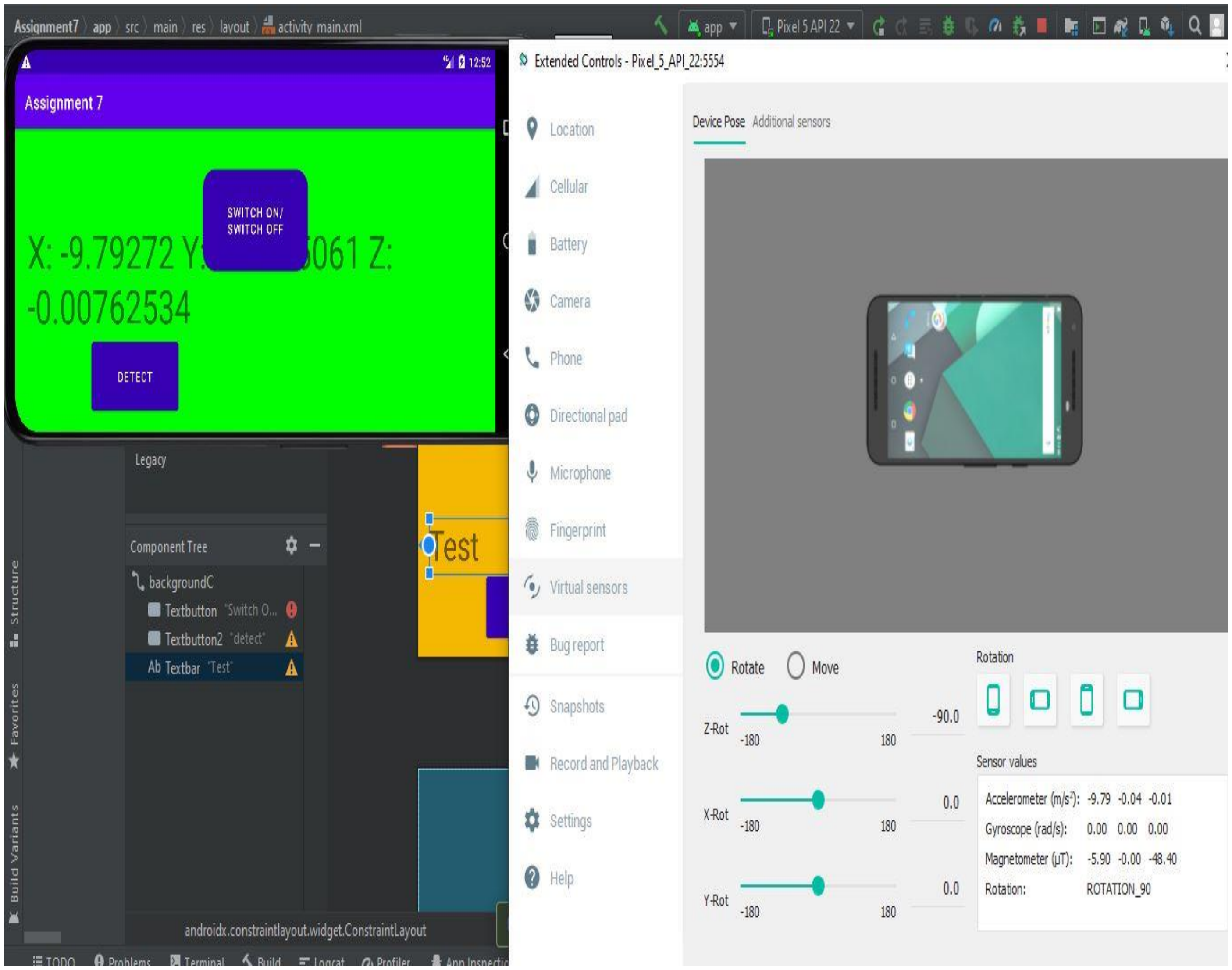


```java
public void flash(View view) {
    CameraManager cameraManager = (CameraManager) getSystemService(CAMERA_SERVICE);
    try {
        for(String id: cameraManager.getCameraIdList()){
            CameraCharacteristics cameraChar = cameraManager.getCameraCharacteristics(id);
            if(cameraChar.get(CameraCharacteristics.FLASH_INFO_AVAILABLE)){
                if(!torchOn){
                    cameraManager.setTorchMode(id,  enabled: true);
                }
                else{
                    cameraManager.setTorchMode(id,  enabled: false);
                }
                torchOn = !torchOn;
            }
        }
    } catch (CameraAccessException e) {
        e.printStackTrace();
    }
}
```

Implement your own Flashlight APP for Android. Implement a a simple user interface (like clickable switch, button or imageview)

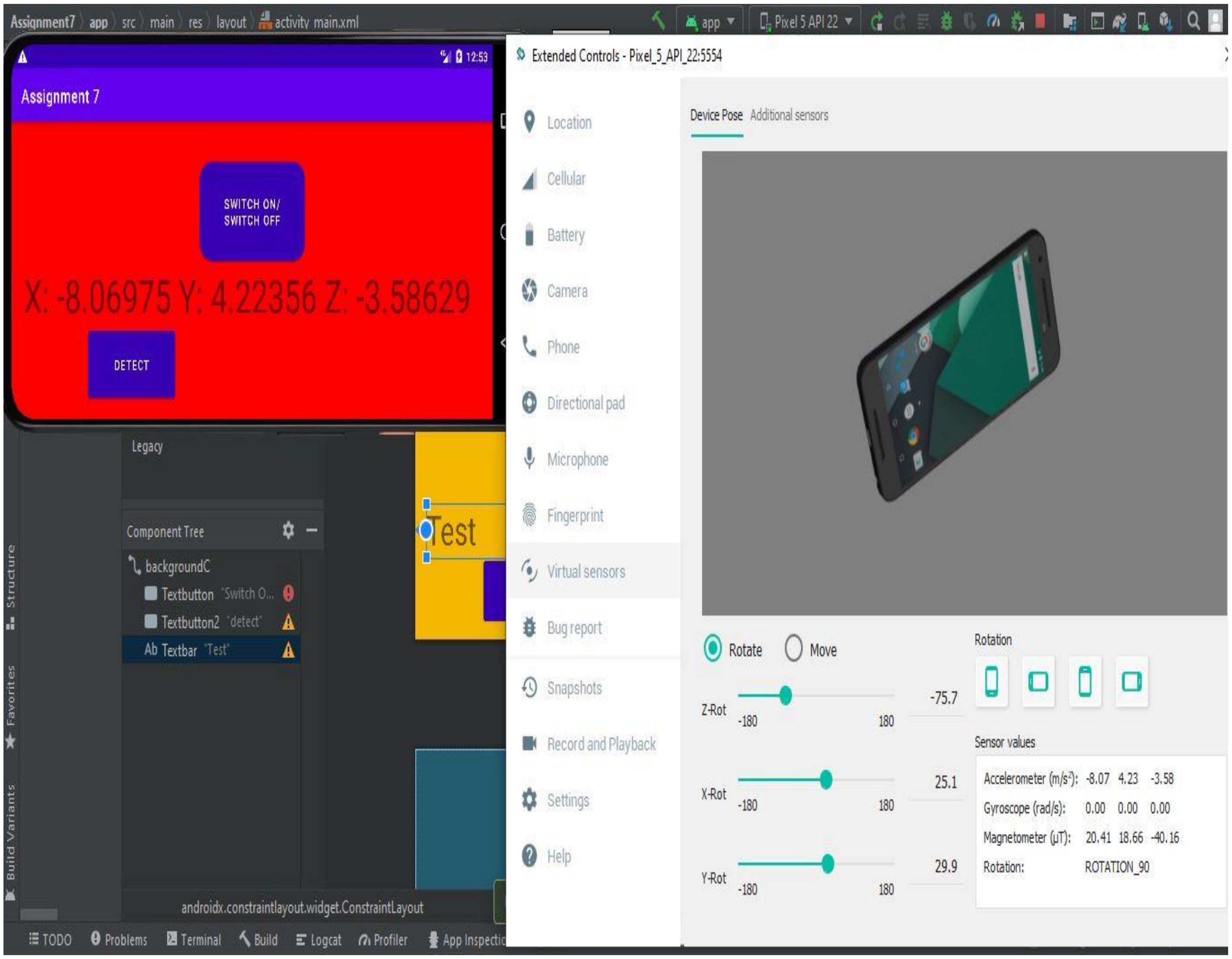


Implement a simple "Level tool" APP, where you can measure if an object is leveled. Make a simple GUI e.g. by showing the level with numbers or background colours (or imageview).



Accelerometer (m/s²): -9.79 -0.04 -0.01
Gyroscope (rad/s): 0.00 0.00 0.00
Magnetometer (μT): -5.90 -0.00 -48.40
Rotation: ROTATION_90

Here you can use Accelerometer Sensor API and listen e.g. xCoordinate. When the xCoordinate is close to zero, the object is leveled.



X: -8.06975 Y: 4.22356 Z: -3.58629

Z-Rot -180 180 -75.7
X-Rot -180 180 25.1
Y-Rot -180 180 29.9

```
Accelerometer (m/s²):  -8.07   4.23   -3.58
Gyroscope (rad/s):      0.00   0.00    0.00
Magnetometer (μT):     20.41  18.66  -40.16
Rotation:              ROTATION_90
```

Exercise 08

1. To meet a use case, almost every software accesses restricted data or performs limited actions. In that situation, some permissions must be declared. Some rights are set up at installation time, while others are set up at runtime. The permissions method shown here is a great example.

Explanation: Without expressing permission, the developer can do the task. However, if he does, he must indicate rights in the manifest file. He selects a permission to be granted during runtime. As a result, the program now requests permission at runtime. The user can fulfill the uses by gaining access to the authorization. Aside from that, no.

• The manifest file declares the permission kinds. The permission kinds can be defined here.

• The main activity file contains the user request. The permission box appears on the screen, and you must accept it.

2. Code lab

MainActivity.java

```
package com.example.studentproject;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements
LocationListener {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void calling(View view) {
        //todo: start listing tp users location through location
manager
        //todo: location permission specified in android
manifest file
        //todo: ask user's permission run-time before accessing
GPS (dangerous permission)
        //todo: Update the latitude and longitude in the UI

        LocationManager locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);
        //todo:2 check if the user granted permission for GPS
        //todo: if not, let's prompt and ask the user to give it

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            //todo we don't have permission so ask it
            ActivityCompat.requestPermissions(this,
                    new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION},
                    0
            );
            return;
        }

locationManager.requestLocationUpdates(LocationManager.GPS_PROVI
DER, 0, 0, this);

    }

    public void docall(View view) {


        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.CALL_PHONE}, 0);
            return;
        }
        //todo we don't have permission so ask it
        Intent intent = new Intent(Intent.ACTION_CALL);
        intent.setData(Uri.parse("tel:0468840334"));
        if (intent.resolveActivity(getPackageManager()) != null)
{
            startActivity(intent);
        }
    }

    public void mapbegin(View view) {
//        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED){
//            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.CALL_PHONE}, 0);
//            return;
//        }
        Intent intent = new Intent(Intent.ACTION_VIEW);
        intent.setData(Uri.parse("geo:23.810331,90.412521"));
        if (intent.resolveActivity(getPackageManager()) != null)
{
            startActivity(intent);
        }
    }

    @Override
    public void onLocationChanged(@NonNull Location location) {
        //todo: now we can read the latitude and longitude from
"location" parameter

        double latitude = location.getLatitude();
        double longitude = location.getLongitude();
        //todo: update the UI
        TextView gpsTextView = findViewById(R.id.beginText);
        gpsTextView.setText("Lat: " + latitude + "Long: " +
longitude);
    }
}
```

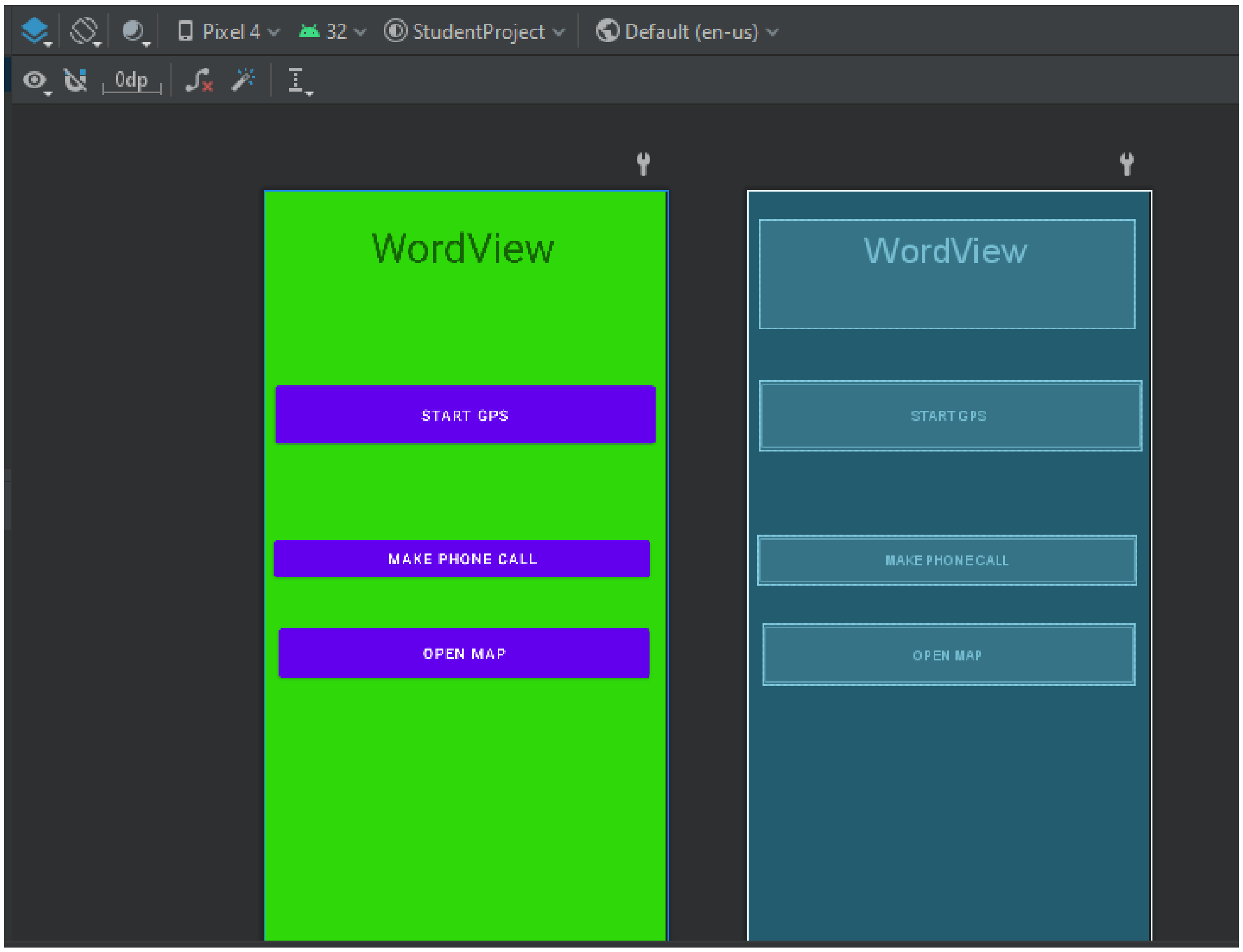
AndroidManifest.xml

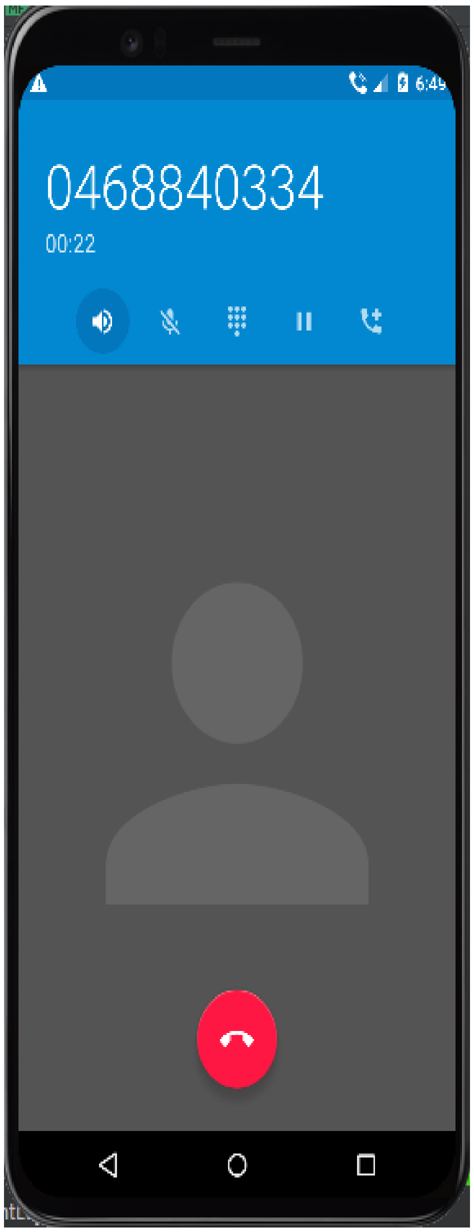
```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.studentproject">

    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission
android:name="android.permission.CALL_PHONE" />


    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.StudentProject">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

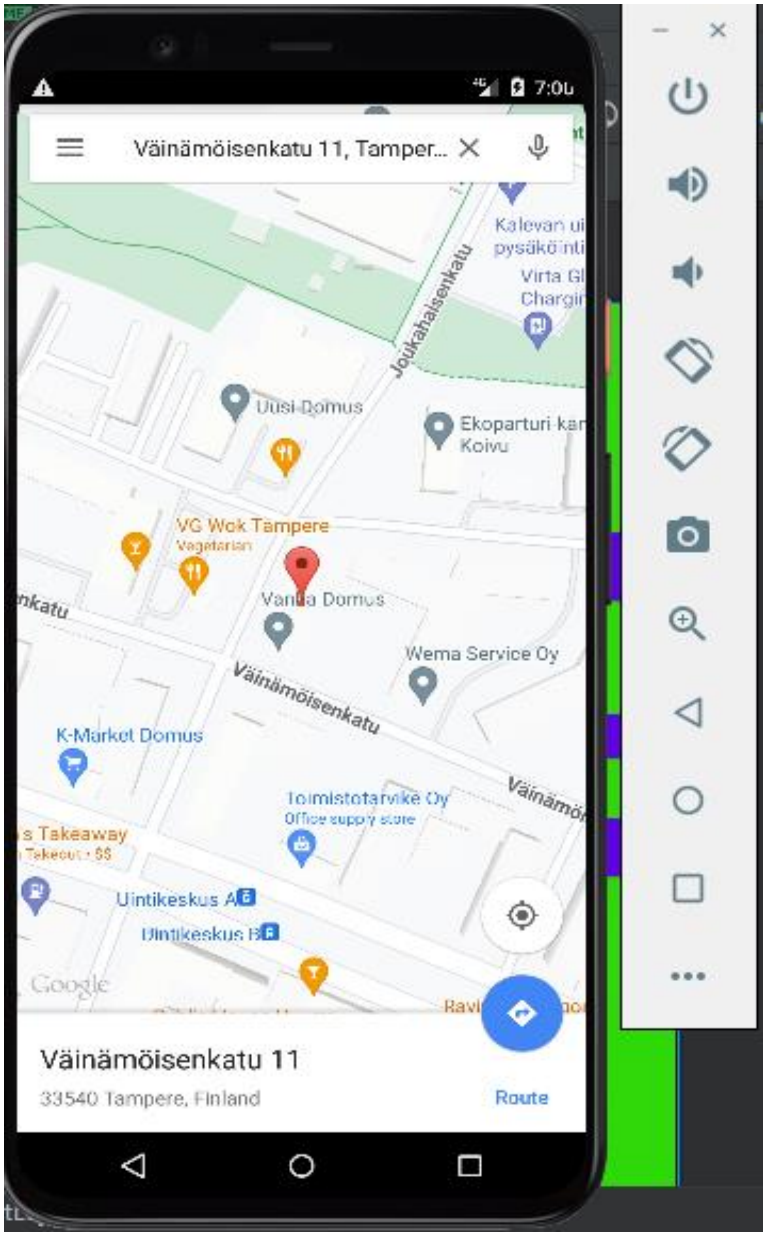
Activity_main.xml

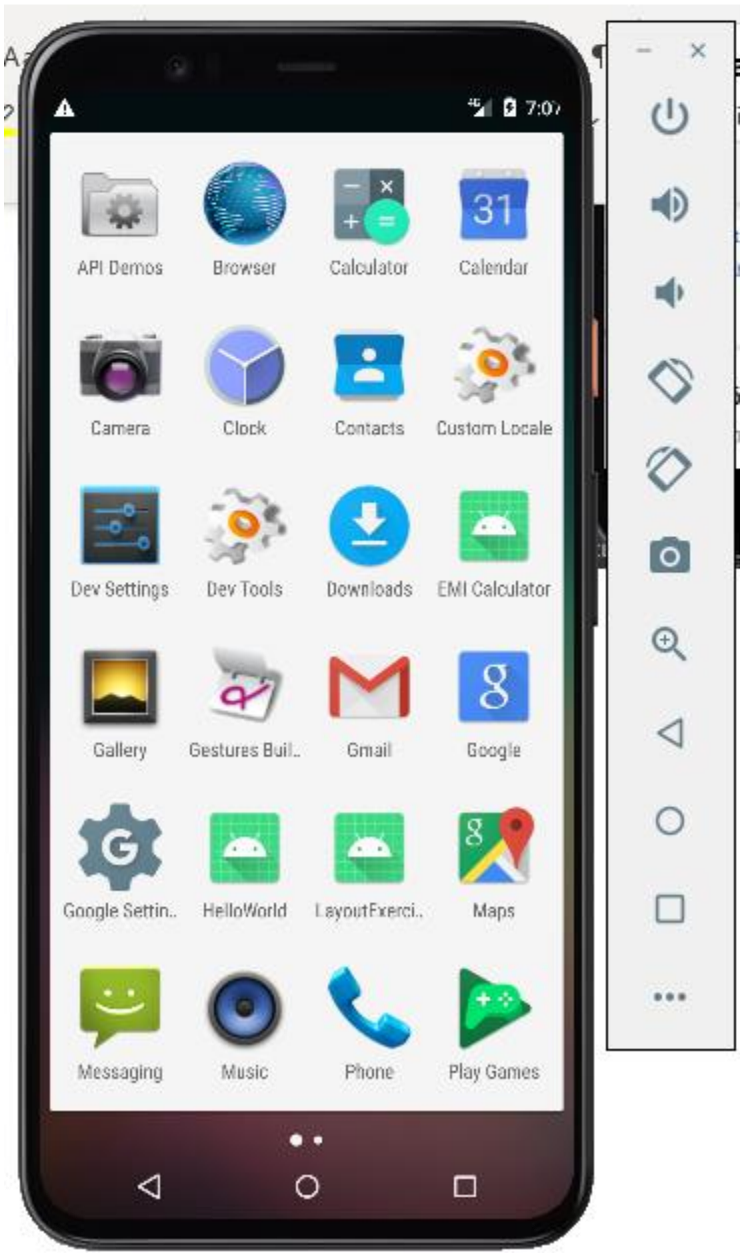
```xml
<?xml version="1.0" encoding="utf-8"?>

<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="#B83737"
    android:backgroundTint="#2ED809"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/BeginText"
        android:layout_width="365dp"
        android:layout_height="105dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="28dp"
        android:layout_marginEnd="40dp"
        android:text="WordView"
        android:textAlignment="center"
        android:textSize="40sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.142"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/sense"
        android:layout_width="372dp"
        android:layout_height="68dp"
        android:layout_marginStart="15dp"
        android:layout_marginTop="52dp"
        android:layout_marginEnd="15dp"
        android:onClick="calling"
        android:text="Start GPS"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.476"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/BeginText" />

    <Button
        android:id="@+id/phonecall"
        android:layout_width="368dp"
        android:layout_height="48dp"
        android:layout_marginStart="15dp"
        android:layout_marginTop="44dp"
        android:layout_marginEnd="15dp"
        android:onClick="docall"
        android:text="Make phone call"
        app:layout_constraintBottom_toTopOf="@+id/openmap"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/sense" />

    <Button
        android:id="@+id/openmap"
        android:layout_width="362dp"
        android:layout_height="60dp"
        android:layout_marginStart="15dp"
        android:layout_marginEnd="15dp"
        android:layout_marginBottom="348dp"
        android:onClick="mapbegin"
        android:text="open map"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```


Pixel 4
32
StudentProject
Default (en-us)
0dp
WordView
START GPS
MAKE PHONE CALL
OPEN MAP
WordView
START GPS
MAKE PHONE CALL
OPEN MAP


0468840334
00:22
6:49

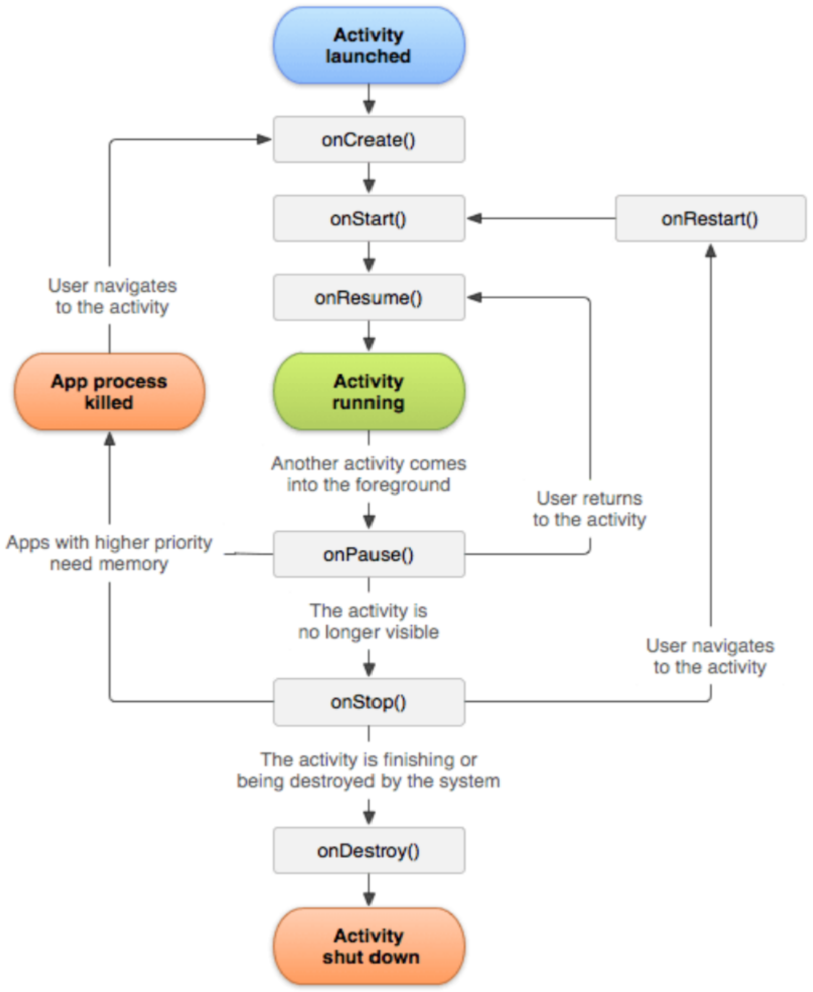

7:00
Väinämöisenkatu 11, Tamper...
Joukahaisenkatu
Uusi Domus
Ekoparturi Koivu
VG Wok Tampere
Vegetarian
Vanha Domus
Wema Service Oy
Väinämöisenkatu
K-Market Domus
Toimistotarvike Oy
Office supply store
Uintikeskus A
Uintikeskus B
Google
Väinämöisenkatu 11
33540 Tampere, Finland
Route


7:07
API Demos
Browser
Calculator
Calendar
Camera
Clock
Contacts
Custom Locale
Dev Settings
Dev Tools
Downloads
EMI Calculator
Gallery
Gestures Buil..
Gmail
Google
Google Settin..
HelloWorld
LayoutExerci..
Maps
Messaging
Music
Phone
Play Games

Exercise 09

Explain Android Activity lifecycle states.

An Android activity goes through six major lifecycle stages or callbacks. These are: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Give an example of each state and when it is entered. Also explain activity lifecycle callbacks.

1. Changing the Orientation of the App

Let's say the user is viewing the app in portrait mode and suddenly the user changes the orientation of the application to landscape mode. Since changing the orientation of the application will eventually change the view, which will lead to a configuration change in the app and the activity associated with the portrait mode is destroyed and new activity associated with the landscape mode is created. The running/active activity associated with the portrait mode will have onPause(), onStop(), and onDestroy() callbacks triggered. When the user changes the orientation, the new activity is created and new activity will have onCreate(), onStart(), and onResume() callbacks triggered.

2. Switching Between the Apps In Multiwindow Screen

As it is known that android has launched a multi-window feature for the Android 7 for the apps with API level 24 and higher, so when any configuration changes of the android app, the android system notify such events and change the lifecycle states of the activity according to the conditions in which configuration of the application have changed. The common example of changing the configuration of the app is when the user resizes one activity with respect to another activity. Your activity can handle the configuration to change itself, or it can allow the system to destroy the activity and recreate it with the new dimensions.

In multi-window mode, although there are two apps that are visible to the user, there is only the one app with which the user is interacting is in the foreground and has focus is in the onResume() state and another app that is in the background and visible to the user is in a Paused state. So it can be said that the activity with which the user is interacting is the only activity which is in the resumed state, while all the other activities are started but not resumed. When the user switches from one app to another app, the system calls onPause() lifecycle state on the running app, and onResume() lifecycle state on another app which is previously not in the active state. It switches between these two methods each time the user toggles between apps.

3. Activity or dialog appears in the foreground

Let’s suppose when an activity is running and is in progress, but at the same time, a new activity appears in the foreground, and partially covering the activity in progress, and take the focus and causing the running activity to enter into the Paused state by calling onPause() on the running activity and the new activity enter into the onResume() state. When the covered activity returns to the foreground and regains focus, it calls onResume(), whereas the running activity again enters into the onPause() state.

If a new activity or dialog appears in the foreground, taking focus and completely covering the activity in progress, the covered activity loses focus and enters the Stopped state. The android operating system then immediately calls onPause() and onStop() in succession. When the same instance of the covered activity comes back to the foreground, the system calls onRestart(), onStart(), and onResume() on the activity. If it is a new instance of the covered activity that comes to the background, the system does not call onRestart(), only calling onStart() and onResume().

4. The user taps the Back button

If an activity is in the foreground and is in running state, and the user taps the Back button, the activity transitions through the onPause(), onStop(), and onDestroy() callbacks. In addition to being destroyed, the activity has also removed the stack, which is used for storing the activities in order of the time put in the stack. It is important to note that, by default, the onSaveInstanceState() callback is not called in this case as it is assumed that the user tapped the Back button with no expectation of returning to the same instance of the activity and so there is no need to save the instance of the activity. If the user overrides the onBackPressed() method, it is still highly recommended that the user should invoke super.onBackPressed() from the overridden method, else if super.onBackPressed() is not invoked then it may lead to ambiguous behavior of the application and may lead to bad user experience.

What is meant by Activity data Bundle? How do you save activity state (member data of your activity)?

onCreate() :

When: it is called when an activity is first created

What: all the necessary UI elements should be initialized

Is activity visible: not yet

onStart():

When: an activity becomes visible to the user

What: code that maintains the UI, start async tasks (get data from API or database), register listeners

Is activity visible: yes

onResume():

When: is called before the user starts interacting with the activity

What: start animations

Is activity visible: yes

onPause():

When: user is leaving the activity

What: stop animations

Is activity visible: partially invisible

onStop():

When: the activity is no longer visible to the user

What: unregister listeners and all resources allocated in onStart()

Is activity visible: no

onRestart():

When: the activity in the stopped state is about to start again (on back click)

What: the cursor should be required

Is activity visible: no

onDestroy():

When: the activity is destroyed from the memory

What: should never be overridden

Is activity visible: no

Steps to implement this behaviour:

Activity1 => startActivityForResult() applied to an Intent with putExtra containing a String

Activity2 => get the String from the bundle and validate data

Activity2 => setResult() and finish current activity

Activity1 => get the result from Activity2 by overriding onActivityResult()

What happens with activity lifecycle when you change the orientation of the device from portrait to landscape when your activity is visible and active?

Lock screen orientation

To lock the screen orientation change of any screen (activity) of your android application makes your activity display only in one mode i.e. either Landscape or Portrait. This is the simplest way to handle screen orientation but not generally recommended.

For this you need to add below line in your projects AndroidManifest.xml. Add the below line along with your activity entry in the AndroidManifest file.

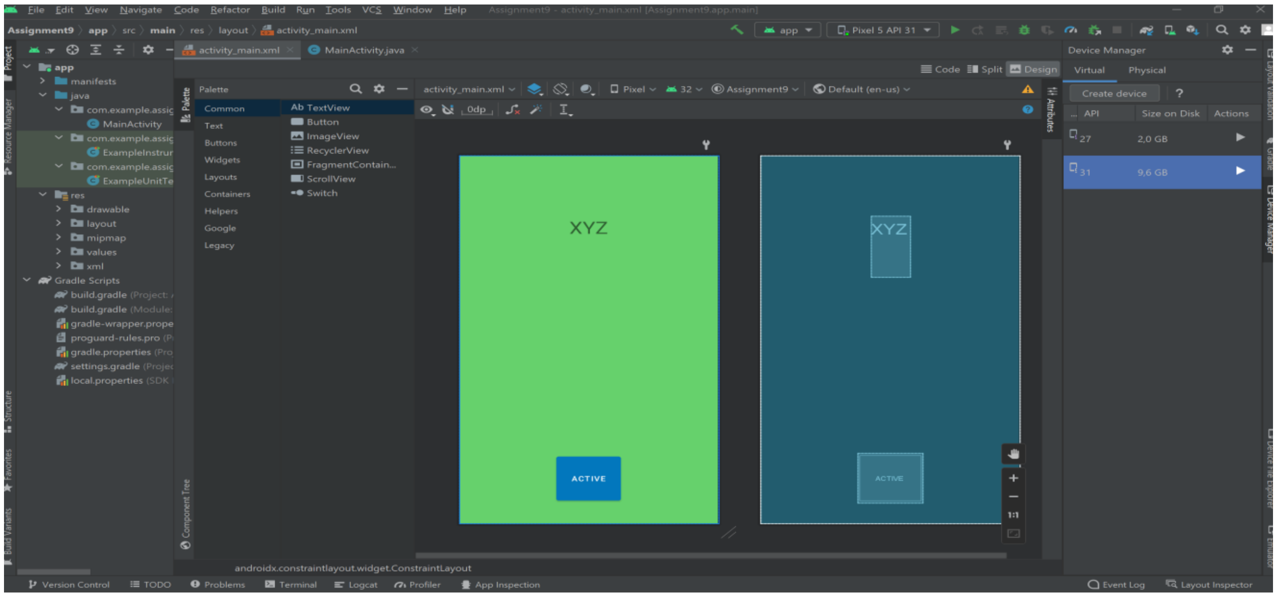
Prevent Activity to recreated

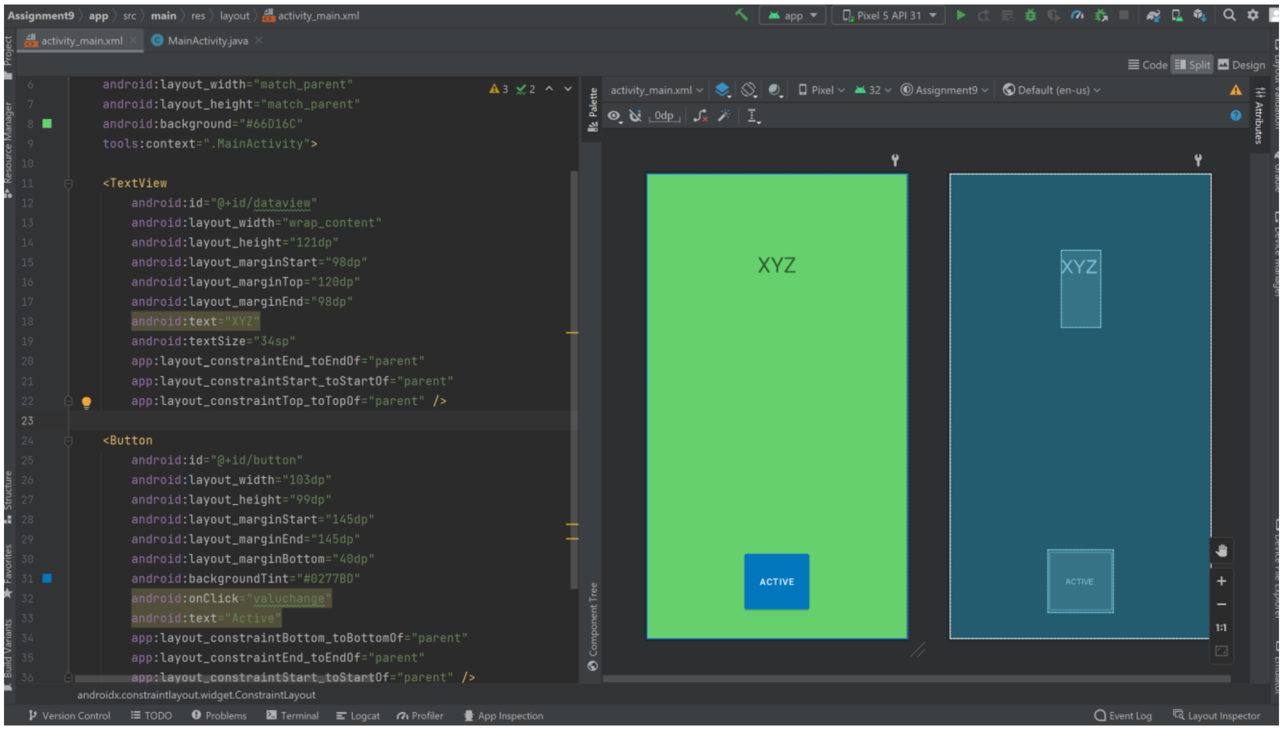
Another most common solution to dealing with orientation changes by setting the android:configChanges flag on your Activity in AndroidManifest.xml. Using this attribute your Activities won’t be recreated and all your views and data will still be there after orientation change.

Save basic state

This is the most common situation to save the basic data of your Activity or Fragment during orientation change. You can save Primitive data such as String, Boolean, Integers or Parcelable objects in a Bundle during the orientation change and read the same data when Activity recreated.

Saving and restoring the data works using two Activity lifecycle methods called onSaveInstanceState() and onRestoreInstanceState().





```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#66D16C"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/dataview"
        android:layout_width="wrap_content"
        android:layout_height="121dp"
        android:layout_marginStart="98dp"
        android:layout_marginTop="120dp"
        android:layout_marginEnd="98dp"
        android:text="XYZ"
        android:textSize="34sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="103dp"
        android:layout_height="99dp"
        android:layout_marginStart="145dp"
        android:layout_marginEnd="145dp"
        android:layout_marginBottom="40dp"
        android:backgroundTint="#0277BD"
        android:onClick="valuchange"
        android:text="Active"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
```

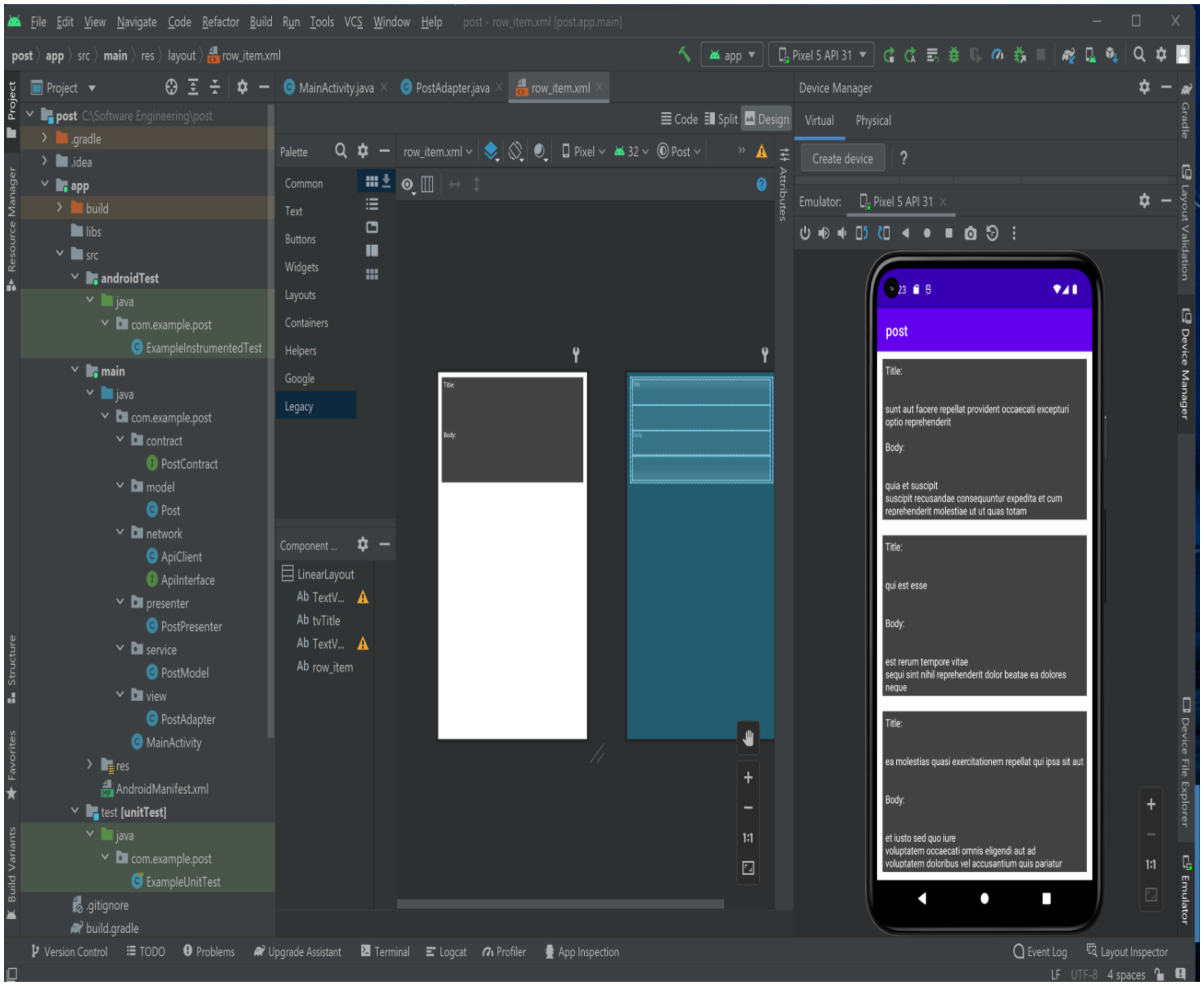
```java
package com.example.assignment9;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

import android.content.Context;
import android.graphics.Color;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements
SensorEventListener  {
    private float x = 0;
    private float y = 0;
    private float z = 0;
    private SensorManager sensorManager;
    private Sensor accelerometer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView sensorValue = findViewById(R.id.dataview);
        sensorValue.setText("X:  " + x + " Y:  " + y + " Z:  " + z);
        SensorManager sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        Sensor accelerometer =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        sensorManager.registerListener((SensorEventListener) this,
accelerometer, SensorManager.SENSOR_DELAY_NORMAL);
    }
    @Override
    protected void onSaveInstanceState(Bundle savedInstanceState){
        super.onSaveInstanceState(savedInstanceState);
        savedInstanceState.putFloat("X1", x);
        savedInstanceState.putFloat("Y1", y);
        savedInstanceState.putFloat("Z1", z);
    }
    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState){
        super.onRestoreInstanceState(savedInstanceState);
        x = savedInstanceState.getFloat("X1");
        y = savedInstanceState.getFloat("Y1");
        z =  savedInstanceState.getFloat("Z1");
        TextView sensorValue = findViewById(R.id.dataview);
        sensorValue.setText("X:  " + x + " Y:  " + y + " Z:  " + z);
    }


    @Override
    protected void onStart() {
        super.onStart();
    }
    @Override
    protected void onPause() {
        super.onPause();
        sensorManager.unregisterListener(this,accelerometer);
    }
    @Override
    protected void onStop() {
        super.onStop();
    }
    @Override
    protected void onResume() {
        super.onResume();
        sensorManager.registerListener(this,accelerometer,
SensorManager.SENSOR_DELAY_NORMAL);
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
    }


    public void valuchange(View view) {
        SensorManager sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        Sensor accelerometer =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        sensorManager.registerListener((SensorEventListener) this,
accelerometer, SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    public void onSensorChanged(SensorEvent sensorEvent) {
        //we get sensor events in sensorevent
        //read the sensor xyz values and show in the textview
        x = sensorEvent.values[0];
        y = sensorEvent.values[1];
        z = sensorEvent.values[2];

        //update the ui
        TextView sensorValue = findViewById(R.id.dataview);
        sensorValue.setText("X: " + x + " Y: " + y + " Z: " + z);

//        ConstraintLayout backgroudlayout = findViewById(R.id.backgroundC);
//        if(z > -0.3 & z < 0.3){
//            //if device is on level change the background to green to
//            backgroudlayout.setBackgroundColor(Color.GREEN);
//        }
//        else{
//            backgroudlayout.setBackgroundColor(Color.RED);
//        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int i) {

    }
}
```

Final Android Project

My task in this Android is to implement an Android application, which utilizes some exiting open REST Api (HTTP/JSON) and some device API. I use two activities and intents between activities. Properly implemented Activity lifecycle.

MVP is a user interface architectural pattern engineered to facilitate automated unit testing and improve the separation of concerns in presentation logic: The model is an interface defining the data to be displayed or otherwise acted upon in the user interface.

Model–view–presenter is a derivation of the model–view–controller architectural pattern and is used mostly for building user interfaces. It has contact in PostContract, model in Post, network in ApiClient, ApiInterface and presenter in PostPresenter, Service in PostModel, view in PostAdapter, MainActivity.



post

Title:

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

Body:

quia et suscipit
suscipit recusandae consequuntur expedita et cum
reprehenderit molestiae ut ut quas totam

Title:

qui est esse

Body:

est rerum tempore vitae
sequi sint nihil reprehenderit dolor beatae ea dolores
neque

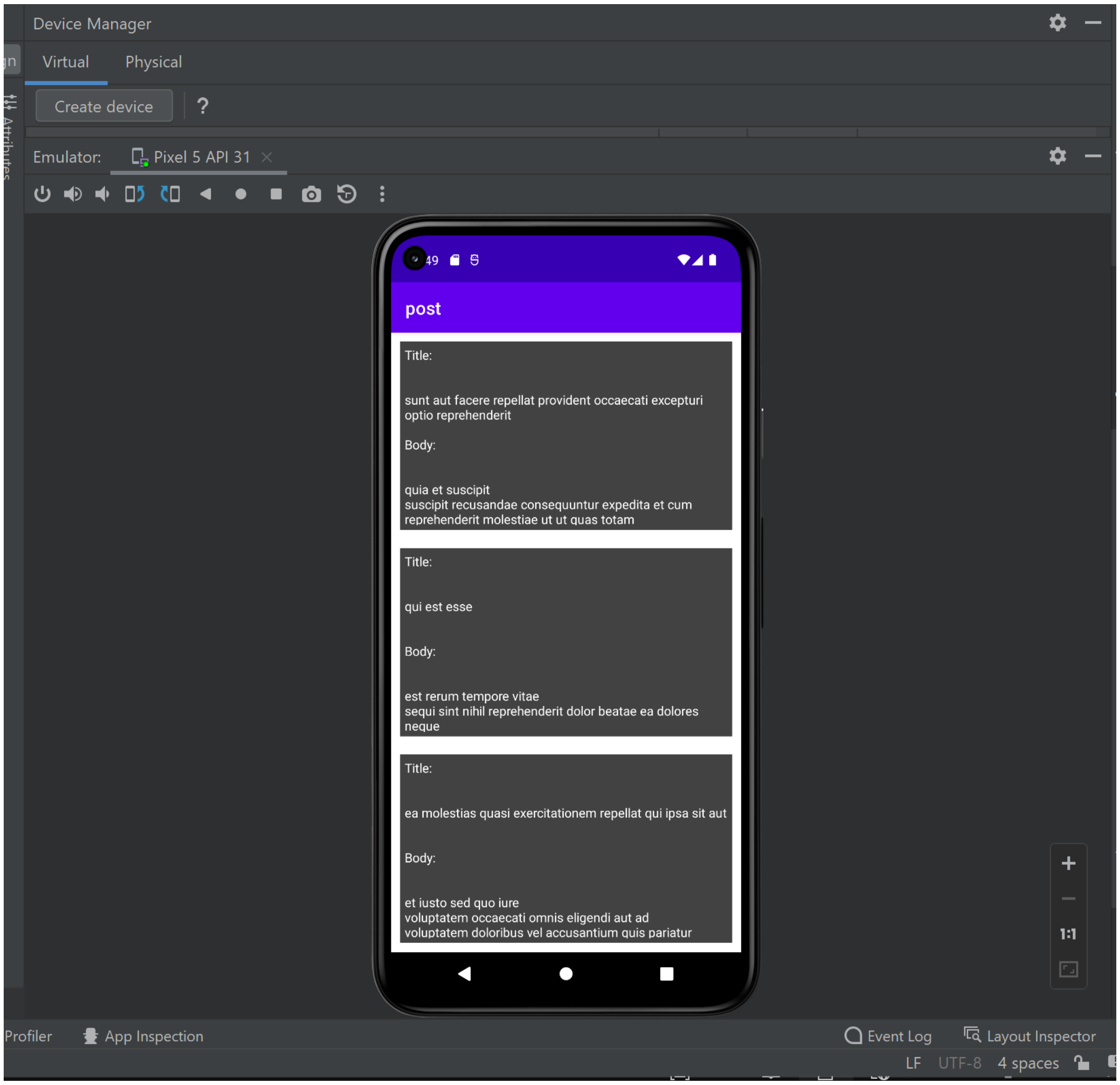
Title:

ea molestias quasi exercitationem repellat qui ipsa sit aut

Body:

et iusto sed quo iure
voluptatem occaecati omnis eligendi aut ad
voluptatem doloribus vel accusantium quis pariatur

I use jsonplaceholder in this site. JSONPlaceholder comes with a set of 6 common resources. I use in the posts, resources have relations. For example: posts have many comments. All HTTP methods are supported. I can use http or https for your requests.
https://jsonplaceholder.typicode.com/



post

Title:

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

Body:

quia et suscipit
suscipit recusandae consequuntur expedita et cum
reprehenderit molestiae ut ut quas totam

Title:

qui est esse

Body:

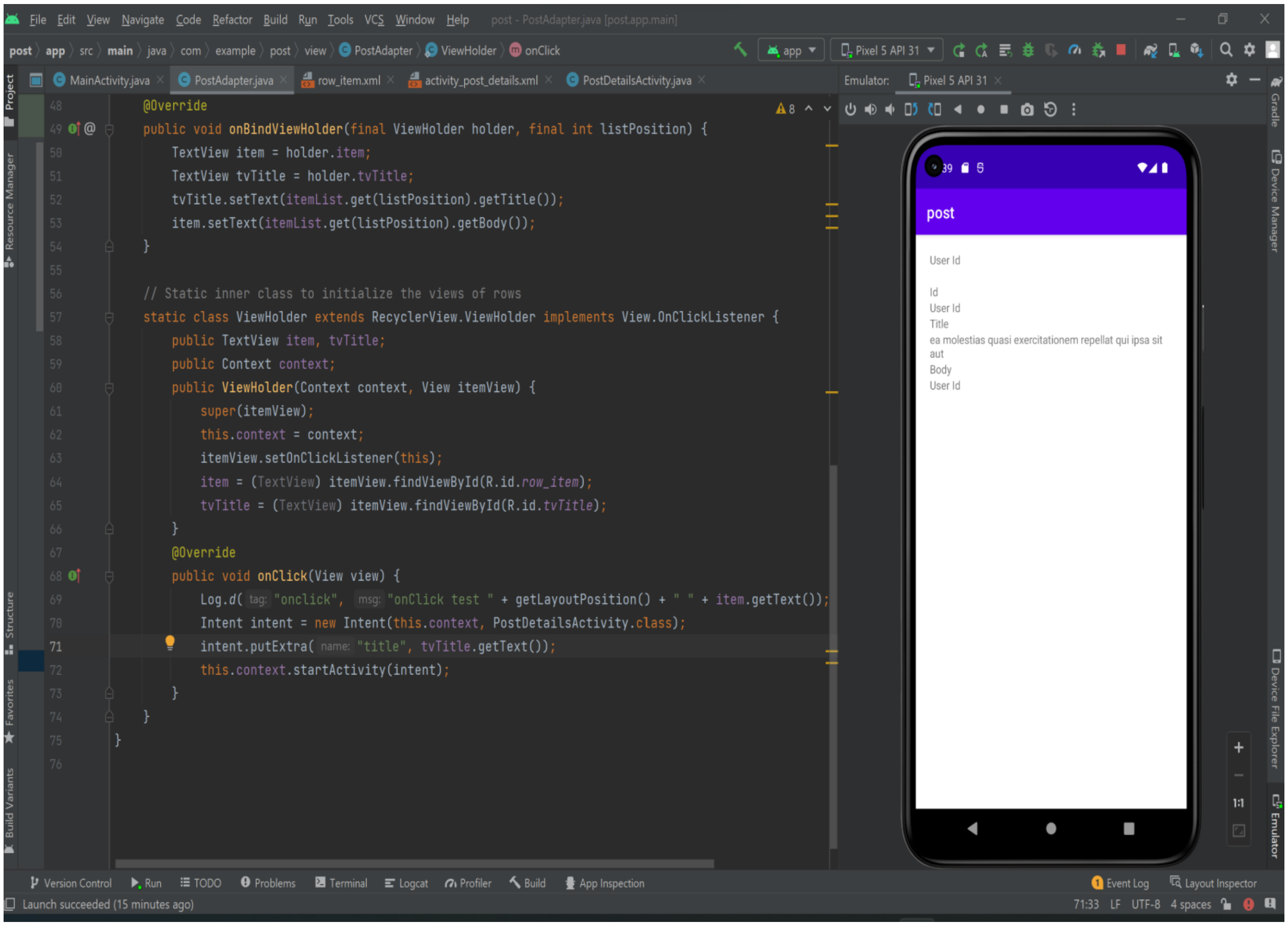
est rerum tempore vitae
sequi sint nihil reprehenderit dolor beatae ea dolores
neque

Title:

ea molestias quasi exercitationem repellat qui ipsa sit aut

Body:

et iusto sed quo iure
voluptatem occaecati omnis eligendi aut ad
voluptatem doloribus vel accusantium quis pariatur

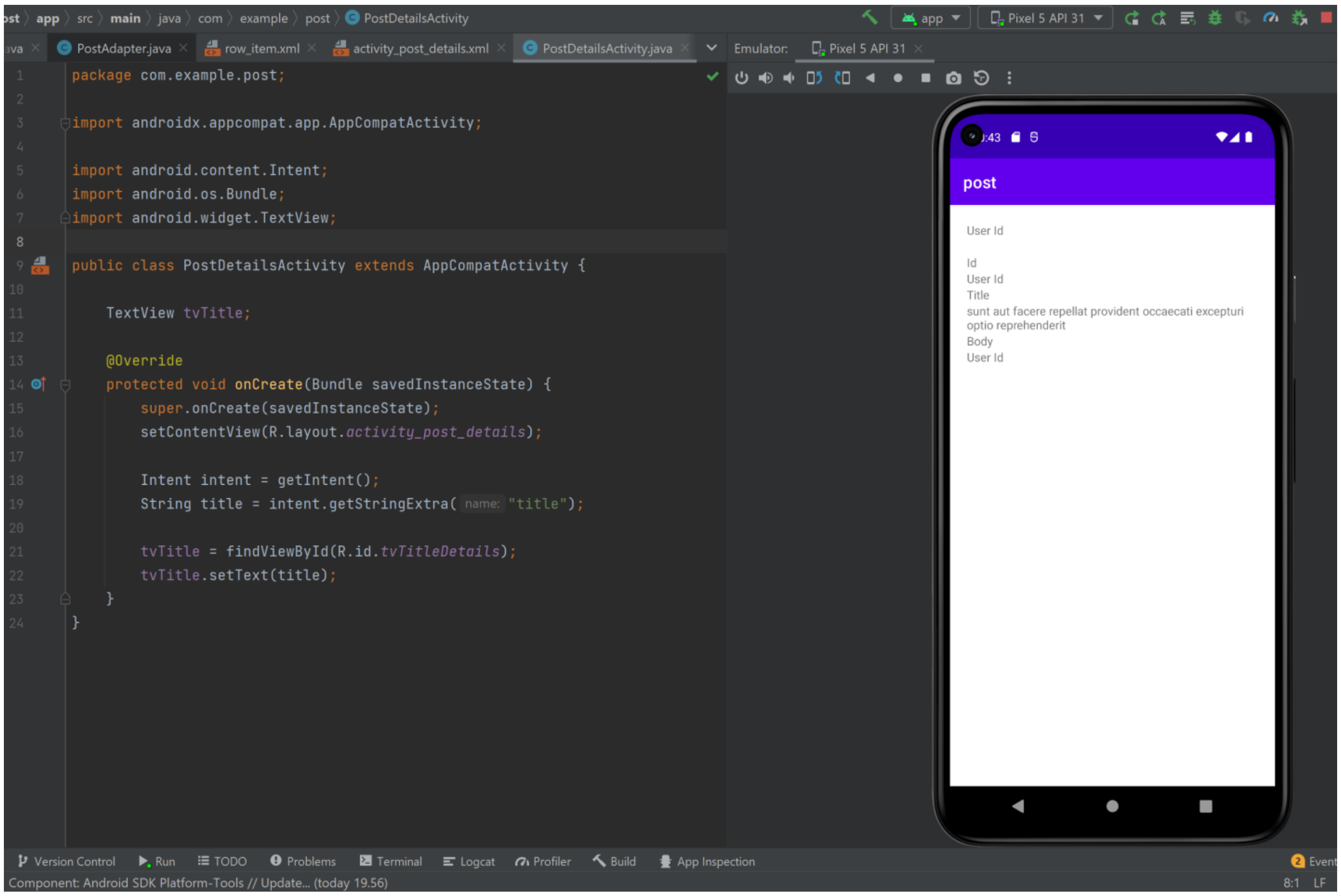


```java
    @Override
    public void onBindViewHolder(final ViewHolder holder, final int listPosition) {
        TextView item = holder.item;
        TextView tvTitle = holder.tvTitle;
        tvTitle.setText(itemList.get(listPosition).getTitle());
        item.setText(itemList.get(listPosition).getBody());
    }

    // Static inner class to initialize the views of rows
    static class ViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {
        public TextView item, tvTitle;
        public Context context;
        public ViewHolder(Context context, View itemView) {
            super(itemView);
            this.context = context;
            itemView.setOnClickListener(this);
            item = (TextView) itemView.findViewById(R.id.row_item);
            tvTitle = (TextView) itemView.findViewById(R.id.tvTitle);
        }
        @Override
        public void onClick(View view) {
            Log.d( tag: "onclick",  msg: "onClick test " + getLayoutPosition() + " " + item.getText());
            Intent intent = new Intent(this.context, PostDetailsActivity.class);
            intent.putExtra( name: "title", tvTitle.getText());
            this.context.startActivity(intent);
        }
    }
}
```

```
post

User Id

Id
User Id
Title
ea molestias quasi exercitationem repellat qui ipsa sit
aut
Body
User Id
```

Launch succeeded (15 minutes ago)

I also add second the Json title with post , link https://jsonplaceholder.typicode.com/



```java
package com.example.post;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

public class PostDetailsActivity extends AppCompatActivity {

    TextView tvTitle;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_post_details);

        Intent intent = getIntent();
        String title = intent.getStringExtra( name: "title");

        tvTitle = findViewById(R.id.tvTitleDetails);
        tvTitle.setText(title);
    }
}
```

```
post

User Id

Id
User Id
Title
sunt aut facere repellat provident occaecati excepturi
optio reprehenderit
Body
User Id
```

Summertime is always the best of what might be.

END