Learn to create a BMI Calculator App for Android

The Body Mass Index (BMI) or Quetelet index is a value derived from the mass weight and height of an individual. The BMI is defined as the body mass divided by the square of the body height, and is universally expressed in units of kg/m². By analyzing the BMI value, we can determine a diagnostic.

First step for the application is to define a layout letting users to enter weight and height values to calculate the BMI index. Besides, we will need a button to launch the BMI calculation and also a TextView to display the result.

The layout will have the following form **activity_main.xml** :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout  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:orientation="vertical"
   tools:context=".MainActivity">

//to show text “Weight”
   <TextView
        android:text="@string/weight"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:gravity="center_horizontal"
        android:layout_marginTop="50dp"
        android:textSize="20sp"/>

//input to insert value “Weight”
   <EditText
        android:id="@+id/weight"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="10dp"
        android:ems="6"
        android:inputType="number|numberDecimal"
        android:textSize="20sp"/>

//to show text “Height”
   <TextView
        android:text="@string/height"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:gravity="center_horizontal"
        android:layout_marginTop="50dp"
        android:textSize="20sp"/>
```

```
//input to insert value “Height”
    <EditText
        android:id="@+id/height"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="10dp"
        android:ems="6"
        android:inputType="number|numberDecimal"
        android:textSize="20sp"/>

//to calculate BMI
    <Button
        android:id="@+id/calc"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="25dp"
        android:onClick="calculateBMI"
        android:text="@string/calculateBMI"
        />

//to show result BMI
    <TextView
        android:id="@+id/result"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:gravity="center_horizontal"
        android:layout_marginTop="25dp"
        android:textSize="20sp"/>

</LinearLayout >
```

Write the Java code in our **strings.xml** file:

```
//to declare text
<resources>
    <string name="app_name">My Application</string>

    <string name="weight">Weight</string>
    <string name="height">Height</string>
    <string name="calculateBMI">Calculate BMI</string>

    <string name="very_severely_underweight">very severely underweight</string>
    <string name="severely_underweight">severely underweight</string>
    <string name="underweight">underweight</string>
    <string name="normal">normal</string>
    <string name="overweight">overweight</string>
    <string name="obese_class_i">obese_class_i</string>
    <string name="obese_class_ii">obese_class_ii</string>
    <string name="obese_class_iii">obese_class_iii</string>

</resources>
```

Now, we can write the Java code in our Main Activity **MainActivity.java** :

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {

    private EditText height;
    private EditText weight;
    private TextView result;

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

        height = (EditText) findViewById(R.id.height);
        weight = (EditText) findViewById(R.id.weight);
        result = (TextView) findViewById(R.id.result);
    }
```

```
//function calculate BMI

    public void calculateBMI(View v) {
        String heightStr = height.getText().toString();
        String weightStr = weight.getText().toString();

        if (heightStr != null && !"".equals(heightStr)
                && weightStr != null  &&  !"".equals(weightStr)) {
            float heightValue = Float.parseFloat(heightStr) / 100;
            float weightValue = Float.parseFloat(weightStr);

            float bmi = weightValue / (heightValue * heightValue);

            displayBMI(bmi);
        }
    }

//to display BMI result

    private void displayBMI(float bmi) {
        String bmiLabel = "";

        if (Float.compare(bmi, 15f) <= 0) {
            bmiLabel = getString(R.string.very_severely_underweight);
        } else if (Float.compare(bmi, 15f) > 0  &&  Float.compare(bmi, 16f) <=
0) {
            bmiLabel = getString(R.string.severely_underweight);
        } else if (Float.compare(bmi, 16f) > 0  &&  Float.compare(bmi, 18.5f)
<= 0) {
            bmiLabel = getString(R.string.underweight);
        } else if (Float.compare(bmi, 18.5f) > 0  &&  Float.compare(bmi, 25f)
<= 0) {
            bmiLabel = getString(R.string.normal);
        } else if (Float.compare(bmi, 25f) > 0  &&  Float.compare(bmi, 30f) <=
0) {
            bmiLabel = getString(R.string.overweight);
        } else if (Float.compare(bmi, 30f) > 0  &&  Float.compare(bmi, 35f) <=
0) {
            bmiLabel = getString(R.string.obese_class_i);
        } else if (Float.compare(bmi, 35f) > 0  &&  Float.compare(bmi, 40f) <=
0) {
            bmiLabel = getString(R.string.obese_class_ii);
        } else {
            bmiLabel = getString(R.string.obese_class_iii);
        }

        bmiLabel = bmi + "\n\n" + bmiLabel;
        result.setText(bmiLabel);
    }
}
```

When a user click on the calculate BMI button, the calculateBMI method is called. We get the values entered by the user for the weight and the height. The height is entered in

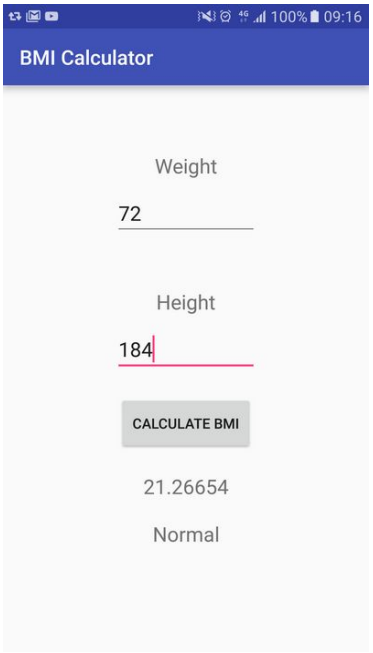
centimeter. For the formula, we need to have a height in meter. So, we divide the value entered by 100. Then, we apply the formula to calculate the BMI :

float bmi = weightValue / (heightValue * heightValue);

With the BMI value, we can display the result on the user interface. We define the displayBMI method for that. To determine the diagnostic associated to the BMI value, we are going to use the following table :

Category	BMI (kg/m^2)		BMI Prime	
	from	to	from	to
Very severely underweight		15.0		0.60
Severely underweight	15	16	0.60	0.64
Underweight	16	18.5	0.64	0.74
Normal (healthy weight)	18.5	25	0.74	1.0
Overweight	25	30	1.0	1.2
Obese Class I (Moderately obese)	30	35	1.2	1.4
Obese Class II (Severely obese)	35	40	1.4	1.6
Obese Class III (Very severely obese)	40		1.6	

The last step is just to display the BMI value and the diagnostic in the result TextView. After that, you can try the application and enjoy the BMI Calculator in action :


100% 09:16
BMI Calculator
Weight
72
Height
184
CALCULATE BMI
21.26654
Normal