Design and Implementation of Body Quality Index App Based on Android

Zhao Limei¹*, Ma Xiaotie²
¹College of Information Engineering, Beijing Institute of Fashion Technology, Beijing, 100029, China
²College of Information Engineering, Beijing Institute of Fashion Technology, Beijing, 100029, China
*Corresponding author’s e-mail: 1067403637@qq.com

Abstract. The body quality index is a commonly used standard for measuring the fatness and health of the human body at home and abroad, referred to as the BMI index. The design is based on the android software development platform, which has developed an app that can help people measure their body quality index. The design module consists of height and weight input module, calculation module, query result module and BMI common sense module. In the development process, with the help of Android studio software, the interface of the app is laid out and designed through the use of xml layout language. The java programming language is used to show users the five functions of body mass index, body weight status, standard weight, healthy weight range and BMI common sense.

1. Introduction
At present, with the diversification of diet and the increase of contemporary work pressure, the per capita weight in China is rising, the proportion of obese people is increasing year by year, and the phenomenon of sudden disease has become a major concern. Now everyone pays special attention to their physical health. To determine whether a person has reached the standard weight, the BMI index of mobile phone software has played a useful role in solving this problem. Whether in domestic or foreign, the BMI index is an important indicator of the health of a person based on the relationship between height and weight. Through the development of this app, users can query their BMI index and health status at any time.

2. App development design
In this design, the body quality index software has two interfaces, the main interface and the query interface. The main interface is to enter the height, weight and gender selection, and the user can edit this. The query interface is calculated by the relevant algorithm, so that the user's BMI index value, physical health status, standard weight, weight range under health, and related knowledge about the BMI index are reflected in the interface. The overall functional block diagram of the App is shown in figure 1.
3. App interface setting

3.1. Layout method
This article mainly applies linear layout. Linear layout is the more common layout method in Android, using the LinearLayout tab. The layout is divided into a vertical layout and a horizontal layout. And there are two types of controls: vertical and horizontal. The vertical layout has only one control per line and the horizontal layout has only one row, and all the controls are arranged from left to right.

3.2. Main page layout
In the created Bmi project, the main layout file is activity_main.xml under the app/res/layout folder. The parent layout of the page adopts the vertical layout in the linear layout, which is realized by setting android:orientation="vertical". The input of height, weight and gender in the submodule adopts horizontal layout and is realized by setting android:orientation="horizontal". The page background is implemented by setting android:background="@drawable/background". PaddingLeft, paddingRight and paddingTop are the distances of the internal text from the left border, the right border and the top [1].

```xml
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</LinearLayout>
```

The layout_width and layout_height properties are used to set the width and height respectively.

3.2.1. Layout control. Each control of the UI layout control has its corresponding properties. By selecting different properties and giving them values, different effects can be achieved. The layout of the article mainly uses three major controls.

1. Text representation control. The text presentation control is what presents to the user. In the main page, such as the text of height, weight and gender, their function is the content displayed on the mobile phone, users can not edit.

```xml
<TextView/>
```

2. Edit text control. The function is to let the user input the content required by the text, which needs to be edited by the user. Taking the height text as an example, the layout xml statement is as follows.

```xml
<EditText
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</LinearLayout>
```
By setting the android: id="@+id/height" attribute, an identifier named "height" is created, which adds a new id to the control. This id is automatically generated in the R.java file. It is convenient to use the "R.id.height" to refer to the height object in the main program. And through the android: hint property setting to get the "Please enter the height" prompt [2].

3. Button control. Button controls are divided into two main controls: radio buttons and command buttons.

Radio button controls are applied to the gender selection module. This is because men and women are mutually exclusive, and only one of the radio buttons in the same group can be selected. Radio button controls must be nested under the RadioGroup container.

Among them, the selected state of the radio button control, in the xml file can be set using android: checked="", set to true if selected, set to false if not selected. This design sets the male to true, that is, after the page is launched, the gender selection defaults to male.

Command button controls are used to respond to the user's click operation. When receiving the response command, it jumps to the next page for corresponding processing. Buttons can be represented in text or image. In this paper, the calculation button and the clear button are designed in text form, and the BMI value, health status, standard weight and healthy weight range are obtained by calculating the button. The trigger event of the calculation button is implemented by setting the android: onClick="calculator" property.

The clear button is a one-key clear when the height and weight values are entered incorrectly, allowing the user to re-enter the value. The android: text="clear" is used to set the button text name, and the trigger event of the clear button is implemented by android: onClick="clear".

3.3. Result page layout
The activity_second.xml layout file is created under the app/res/layout folder to display the app's query results. The parent layout of the page adopts the vertical layout, the same as the main page. There are five modules on the results page, which are weight status, BMI index, standard weight, healthy weight range, and related BMI index.

Among them, weight status, BMI, standard weight and healthy weight range were all laid out in the same way. And they all adopted the horizontal layout in the linear layout. And set the property android: layout_gravity to left, so that the position displayed by the control is at the leftmost end of the interface.

In the results page, the layout control is applied to the text representation control, which is used to render the calculated result. Take the weight status as an example, set the android: id="@+id/weightcondition" attribute in the TextView. It is convenient to use the "R.id.weightcondition" to refer to the object in the Activity program [3]. The BMI index, standard weight and healthy weight range are the same as above and add the new android: id="@+id/myBMI" attribute, android: id="@+id/standard_weight" attribute and android: id="@+id/health_weight" attribute in the text representation control to add new the id.

The final introduction to the BMI index common sense is implemented by the android: text=" " attribute in the text representation control.

4. Main program design
MainActivity.java is the main program file. The onCreate method has been automatically called by the system when the activity is created. It is the beginning of the activity life cycle. The programming uses the java programming language and uses the BMI algorithm to calculate the BMI value. The BMI value is compared with the index reference standard to determine the weight status. Finally, the standard weight and the range of healthy weight is calculated according to the relevant algorithm and the derived inverse formula. The specific flow chart is shown in figure 2.
4.1. Initialization settings
First, declare the user height and user weight in the MainActivity, rewrite the parent class through @override, and use the onCreated method to complete the initialization work when the activity starts.

The main program needs to continue adding the findViewById method in the default onCreate method to initialize the height and weight input box control. Among them, R.id.height indicates that the id of the height input box is height, which is mentioned in the height layout file in 3.2.1.

4.2. Button listener implementation
Implementing a button listener event can be done by declaring the button's id in the activity_main file and setting the setOnClickListener click event for it in MainActivity.java. Or declare the button property setting in the activity_main file, and set its method in MainActivity.java [4]. This article uses the second method, does not require id and initialization, directly in the layout file by setting android:onClick="calculator" and android:onClick="clear" attribute, to achieve the triggering event of the calculation and clear button.

4.3. BMI index calculation and display

4.3.1. BMI index calculation. First, the algorithm needs to obtain the user's height and weight values. And use the getText() method to convert the user's height text content into a string and assign it to the weight, and convert the user's weight text content into a string and assign it to the weight.

Second, convert the height and weight variables of String type into double type and represent them with h and w variables respectively. Convert by applying the Double.parseDouble method.
Finally, the BMI index is equal to the weight divided by the square of the height. The algorithm formula is BMI index = weight (kg) / (height * height (m)). In the case where the height h is not zero, the calculation result is assigned to the result variable.

double result = w / (h * h)

4.3.2. Starting SecondActivity. First, generate an intent object in MainActivity, call the setClass method to set the SecondActivity to be started, call the startActivity method to start the Activity. After that, the activity needs to add the passed data, use bmi to receive the value of result, and height to receive the input height value.

Finally, register the SecondActivity in the app/manifests/AndroidManifest.xml manifest file.

4.3.3. BMI index display. In the activity_second.xml layout file, the text indicates that the control has set the android: id="@+id/myBMI" attribute, and the myBMI identifier is created. This requires that the application in secondactivity.java file use R.id.myBMI to get the interface component.

In SecondActivity.java, the interface of the Activity is displayed using the activity_second.xml layout file, which is implemented by the code setContentView(R.layout.activity_second). The findViewById method is used to initialize the text representation control that outputs BMI value, which is implemented by myBMI = (TextView) findViewById(R.id.myBMI). After that, BMI index data is extracted by getDoubleExtra method. Finally, the calculated BMI value is displayed by the setText method. And the BMI value retains two decimal places.

    int v = (int)(value*100+0.5);
    myBMI.setText(v/100.0+ " (21-22) is the best");

In the process of BMI calculation, two cases need to be paid attention to. One is to determine whether the edited text control has numeric input. If the text of one of the controls does not input a value, then an error message will appear, and get a message prompt for height or weight please input data. The second is to determine whether the input height value is zero. If it is zero, the text content is cleared by the userheight.setText(""), and the user needs to re-enter the height.

4.4. Weight status judgment and display

The weight status is classified according to the size of the BMI index. The smaller the index, the lighter the weight, the bigger the index and the higher the weight. The specific classification criteria in this article is shown in Table 1.

| Guideline | BMI classification | Health status |
|-----------|--------------------|---------------|
| BMI<18.5  | Too light          | Need nutrition|
| 18.5<=BMI<24 | normal             | Continue to keep|
| 24<=BMI<28 | overweight         | Healthy weight loss|
| 28<=BMI<30 | Grade 1 obesity    | Exercise       |
| 30<=BMI<40 | Grade 2 obesity    | Exercise       |
| BMI>=40   | Grade 3 obesity    | Exercise       |

First, the output of the weightCondition text represents the control initialization, implemented by the code weightCondition = (TextView) findViewById (R. id. weightcondition). According to the BMI classification of Table 1, the following code is written to realize the situation judgment and displayed in the text representation control with the id weightCondition.

4.5. Standard weight and healthy weight calculation and display

The standard weight calculation is only related to the user's height value. When the user's height value is obtained, the code is calculated according to the summarized promotion formula to calculate the
standard weight. Standard weight (kg) = [height (m) - 1] * 0.9, the formula is universal, and the results of the calculation can be applied to all Asians.

doUBLE height=intent.getDoubleExtra("height",-1);
double w=(int)((height-1) * 0.9);
standard_ weight.setText(w*100+"”);

The healthy weight range refers to the state of weight under normal conditions, that is, the BMI value is between 18.5 and 24. By inversely deriving the formula for calculating the BMI index, it is concluded that when the BMI value is 18.5, the body weight is at a normal minimum. Min=18.5* height* height. When the BMI value is 24, it is the normal maximum body weight, Max = 24 * height * height. The definition of a healthy weight range is between the minimum and maximum values.

doUBLE min=(int)((18.5*height*height)*100+0.5)/100;
doUBLE max=(int)((24*height*height)*100+0.5)/100;
height_ weight. setText(min+ “” +max);

4.6. App physical map

Figure 3. Main interface. Figure 4. Query interface.

5. Conclusion
The design of this App is mainly accomplished through two parts: interface setting and main program implementation. Through the familiarity of the xml layout language and the java language, an App with five functions of displaying weight status, BMI index, standard weight, healthy weight range and BMI common sense is realized. The advantage of the design is in the development process, through the derivation of formula calculation, constantly modify the source code and repeatedly run debugging, so as to complete the function of the test. The disadvantage is that it only realizes the basic functions of BMI, and it still needs further study and optimization of the code to achieve more functions.

References
[1] Wang Xianghui, Zhang Guoyin, Shen Jie. (2010) Android application development. Tsinghua university press, Beijing.
[2] Dong Minghua, li Hongwei. (2017) Android entry program development-BMI calculator. Computer knowledge and technology, 163: 51–59.
[3] Yin Mengzheng. (2016) Overview of android-based APP development platform. Communications power technology, 33:154-155+213.
[4] Yao Haozhe, Liu Lijuan. (2016) Construction of an App for personal health promotion based on Android platform. Computer era, 11: 28-31.