



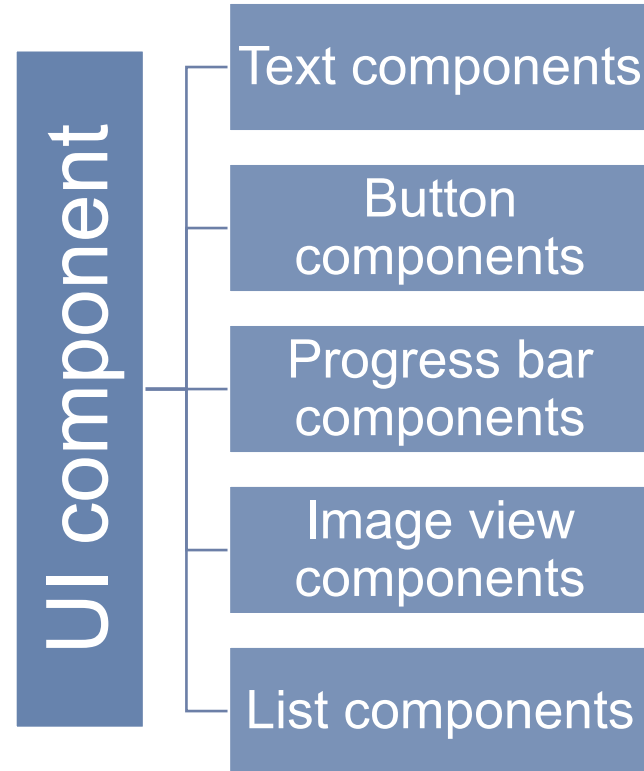
移动应用开发

张昕

zhangxin@cust.edu.cn

2022

UI components



Text components

■ **TextView**

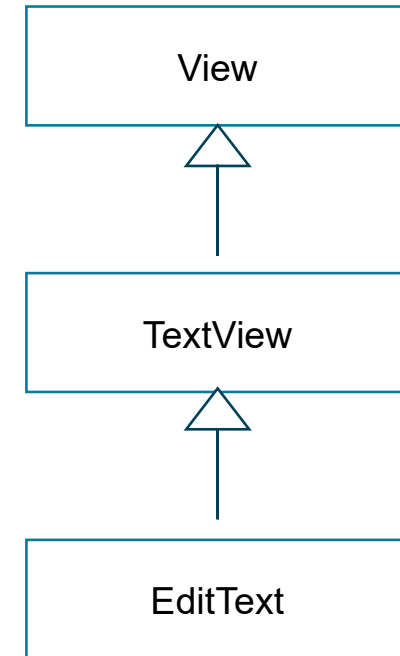
- Displays text
- Read-only

■ **EditText**

- Also displays text
- Read and edit

■ **TextView can be added with either XML file or Java file**

- XML highly recommended

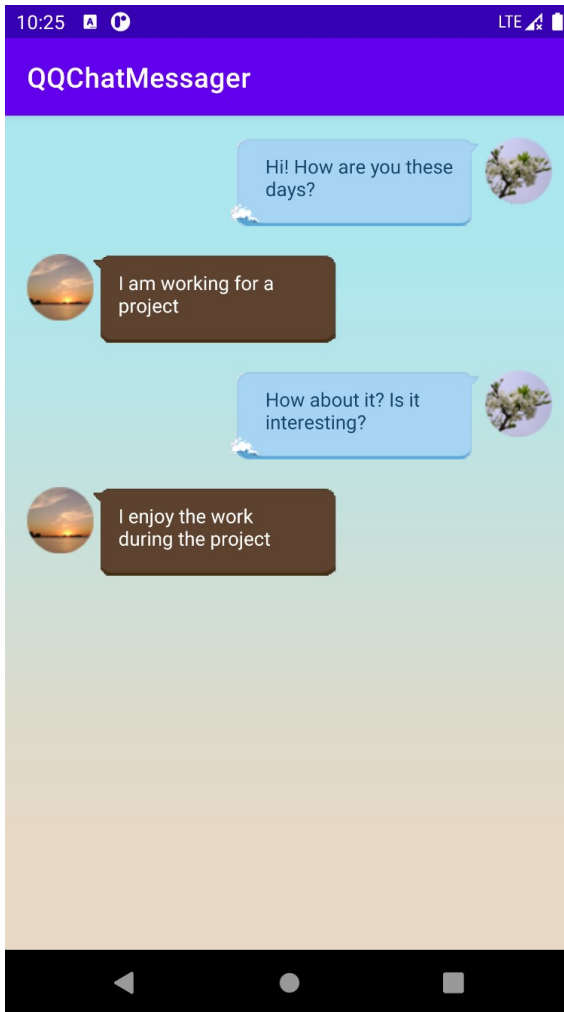


TextView

```
<TextView  
    Attribute list  
>  
</TextView>
```

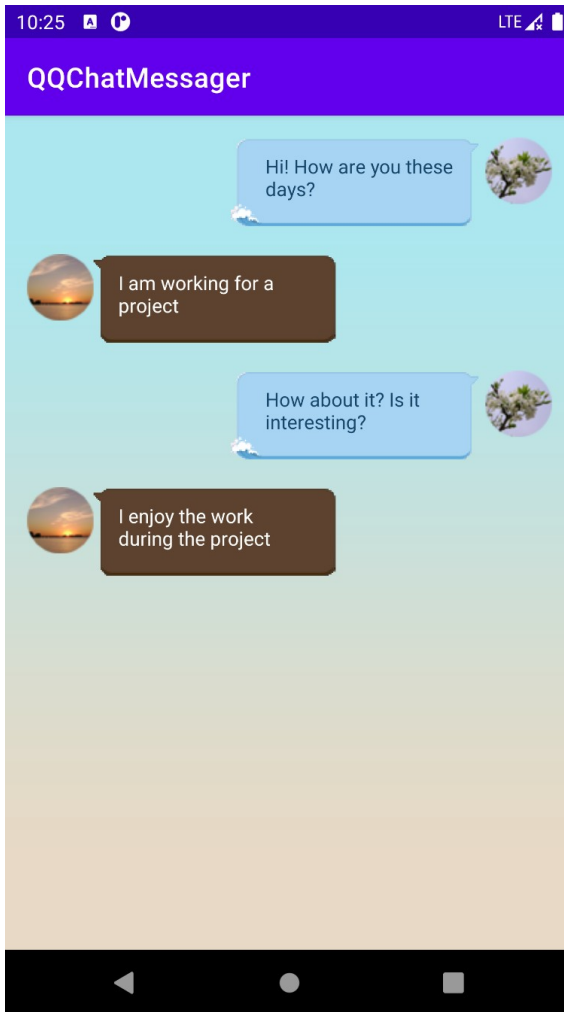
Attribute	Methods
android:autoLink	Specifies whether to convert the text in the particular format into “hyperlink” (none, web, email, phone, map, all)
android:drawableBottom	Draws the specific image at the bottom area within textview. Images are located in res\mipmap and pointed by “@mipmap/filename (without extension name)”
android:drawableLeft	Draws the specific image at the left area within textview. Images are located in res\mipmap and pointed by “@mipmap/filename (without extension name)”
android:gravity	Specifies the alignment of text within textview. The values could be top, bottom, left, right, center_vertical, fill_vertical, center_horizontal, fill_horizontal, center, fill, clip_vertical, clip_horizontal. Multiple values could be mentioned with “ ”, e.g., “right bottom”
android:inputType	Specifies the text type within textview, i.e., textPassword, textEmailAddress, phone, date. “ ” as separator.
android:singleLine	Specifies whether the text is displayed as single line. “true” means the text will not be broken into lines. Multiple lines by default.
android:text	Displays the text displayed in textview.

Small Demo



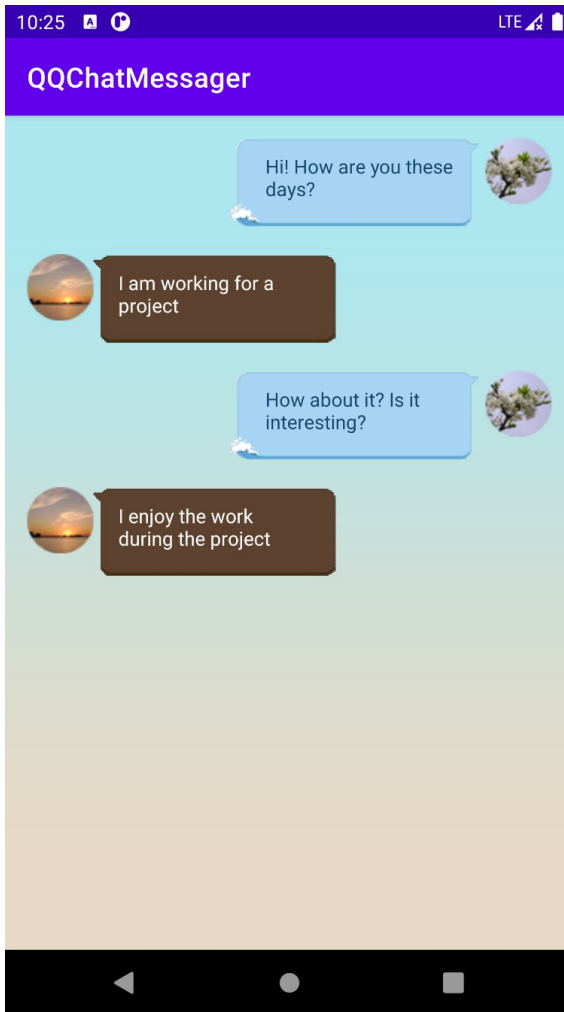
```
<GridLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/bg"
android:columnCount="6"
android:paddingLeft="16dp"
android:paddingTop="16dp"
android:paddingRight="16dp"
android:paddingBottom="16dp"
tools:context="com.example.qqchatmessenger.MainActivity">
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_row="0"
    android:layout_column="1"
    android:layout_columnSpan="4"
    android:layout_gravity="end"
    android:layout_marginRight="5dp"
    android:layout_marginBottom="20dp"
    android:background="@drawable/bg_textview"
    android:maxWidth="180dp"
    android:text="Hi! How are you these days?"
    android:textColor="#16476B"
    android:textSize="14sp" />
```

Small Demo



```
<ImageView
    android:id="@+id/ico1"
    android:layout_row="0"
    android:layout_column="5"
    android:layout_columnSpan="1"
    android:layout_gravity="top"
    android:src="@drawable/ico2" />
<ImageView
    android:id="@+id/ico2"
    android:layout_row="1"
    android:layout_column="1"
    android:layout_gravity="top"
    android:src="@drawable/ico1" />
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_row="1"
    android:layout_marginBottom="20dp"
    android:background="@drawable/bg_textview2"
    android:maxWidth="180dp"
    android:text="I am working for a project"
    android:textColor="#FFFFFF"
    android:textSize="14sp" />
```

Small Demo



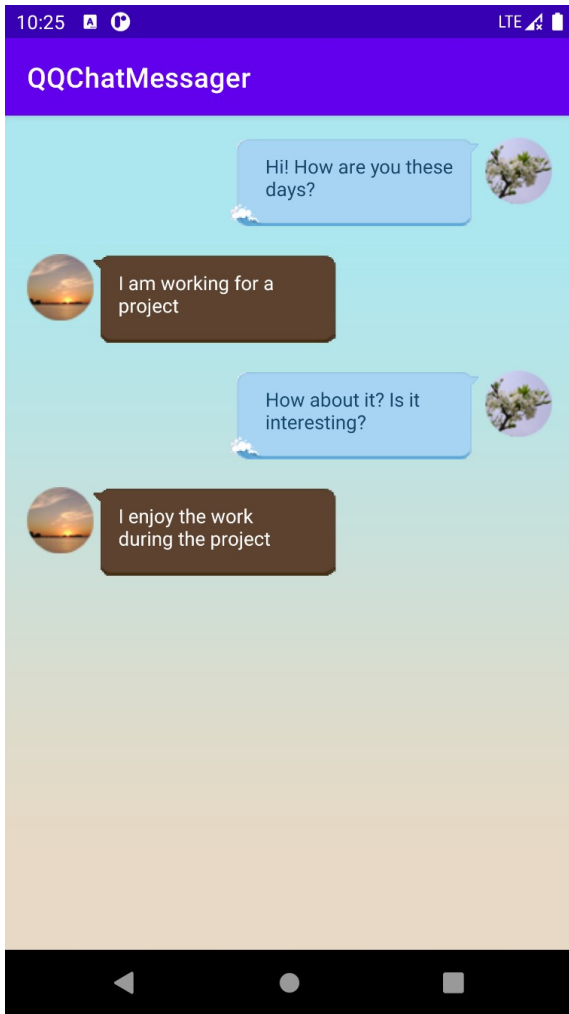
<TextView

```
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_row="2"
android:layout_column="1"
android:layout_columnSpan="4"
android:layout_gravity="end"
android:layout_marginRight="5dp"
android:layout_marginBottom="20dp"
android:background="@drawable/bg_textview"
android:maxLength="180dp"
android:text="How about it? Is it interesting?"
android:textColor="#16476B"
android:textSize="14sp" />
```

<ImageView

```
android:id="@+id/ico3"
android:layout_row="2"
android:layout_column="5"
android:layout_columnSpan="1"
android:layout_gravity="top"
android:src="@drawable/ico2" />
```

Small Demo



```
<ImageView
    android:id="@+id/ico4"
    android:layout_row="3"
    android:layout_column="1"
    android:layout_gravity="top"
    android:src="@drawable/ico1" />
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_row="3"
    android:layout_marginBottom="20dp"
    android:background="@drawable/bg_textview2"
    android:maxLength="180dp"
    android:text="I enjoy the work during the project"
    android:textColor="#FFFFFF"
    android:textSize="14sp" />
</GridLayout>
```




EditText

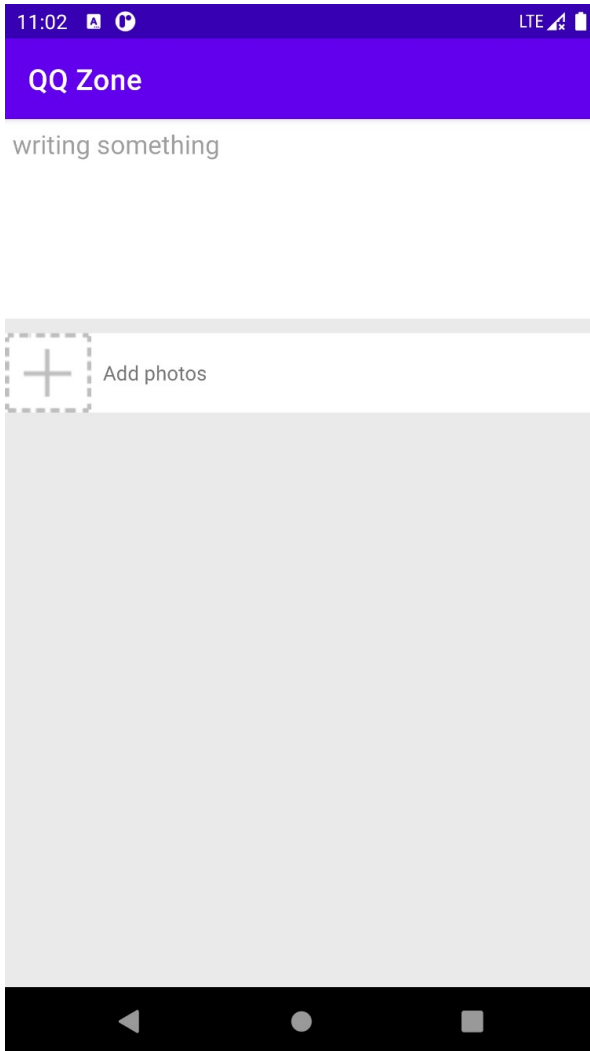
```
<EditText  
    Attribute list  
>  
</EditText>
```

- ❑ EditText is the subclass of TextView class, and it could possess all the acts of TextView.
- ❑ To obtain the content of EditText component, the getText() method in Java can be employed.

```
EditText login = (EditText)findViewById(R.id.login); //component id is login  
String loginText = login.getText().toString();
```

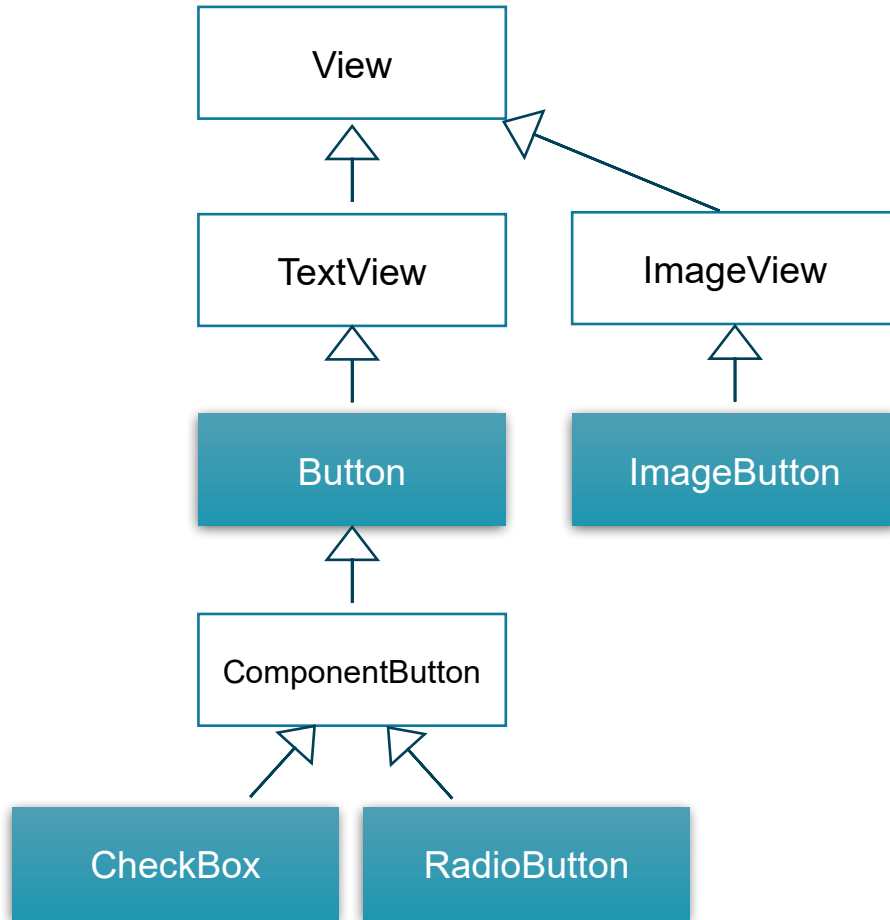


Small Demo



```
<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"
    android:paddingBottom="0dp"
    android:paddingTop="0dp"
    android:paddingRight="0dp"
    android:paddingLeft="0dp"
    android:background="#EAEAEA"
    tools:context="com.example.qqzone.MainActivity">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText1"
        android:lines="6"
        android:hint="writing something"
        android:padding="5dp"
        android:background="#FFFFFF"
        android:gravity="top"
        android:layout_marginBottom="10dp"
        android:inputType="textMultiLine"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView1"
        android:text="Add photos"
        android:drawableLeft="@mipmap/addpicture"
        android:drawablePadding="8dp"
        android:gravity="center_vertical"
        android:background="#FFFFFF"
        android:textColor="#767676" />
</LinearLayout>
```

Button component



Button

```
<Button
    android:id="@+id/ButtonID"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="Button displayed text"
>
</Button>
```

- ❑ After adding Button in layout, the corresponding event handling with event listeners would be employed to make buttons interact with users
- ❑ Solution 1: within MainActivity.java

```
import android.view.View.OnClickListener;
import android.widget.Button;
Button login = (Button)findViewById(R.id.login);
login.setOnClickListener(new OnClickListener(){
    public void onClick(View v){
        //code invoked by event occurrence
    }
});
```

Button

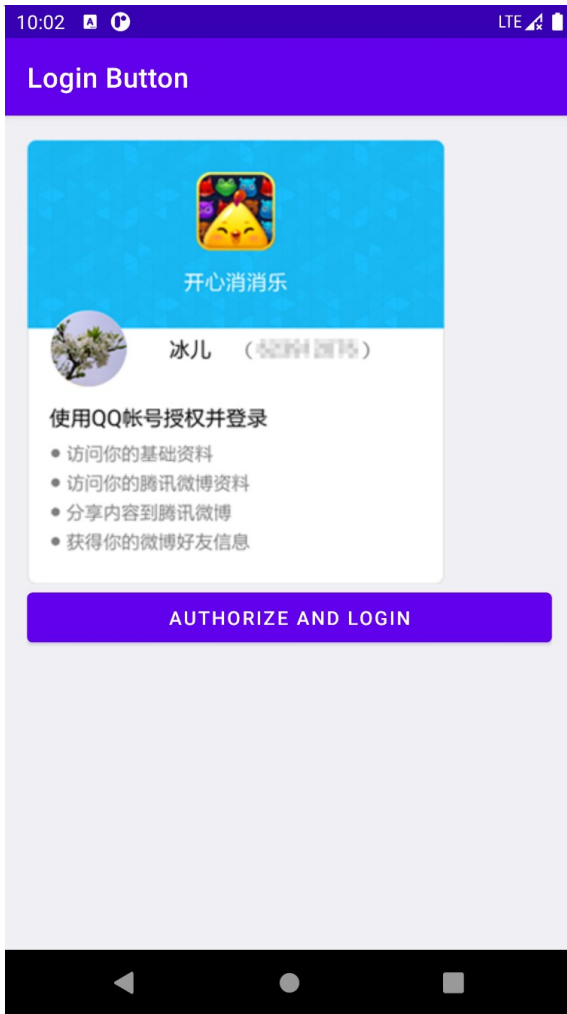
```
<Button
    android:id="@+id/ButtonID"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="Button displayed text"
>
</Button>
```

- ❑ After adding Button in layout, the corresponding event handling with event listeners would be employed to make buttons interact with users
- ❑ Solution 2:
 - ❑ create a method with argument of View object and settle the invoked actions within the method
 - ❑ Set android:onClick attribute to point the specific method (mentioned above) for calling the actions

```
public void myClick(View view){
    //the code of invoked actions
}
```

```
<Layout...
    android:onClick="myClick"
>
```

Small Demo



```
<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"
android:background="#EFFFF4"
android:paddingBottom="16dp"
android:paddingLeft="16dp"
android:paddingRight="16dp"
android:paddingTop="16dp"
tools:context="com.example.loginbutton.MainActivity">
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageView1"
    android:scaleType="fitEnd"
    android:src="@drawable/top"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/button1"
    android:background="@drawable/shape"
    android:text="Authorize and login"
    android:textColor="#FFFFFF"/>
</LinearLayout>
```

ImageButton

<ImageButton

```
android:id="@+id/ImageButtonID"  
android:layout_height="wrap_content"  
android:layout_width="wrap_content"  
android:src="@mipmap/ImageFileName"  
android:scaleType="Zoom method"
```

>

</ImageButton>

▣ scaleType attribute value

Value	Description
matrix	Zoom image under matrix pattern
fitXY	Zoom image in horizontal and vertical direction separately to fill ImageButton, which would cause the original aspect ratio changed
fitStart	Preserve aspect ratio till the image could fill ImageButton, and the image is located at top-left corner of ImageButton
fitCenter	Preserve aspect ratio till the image could fill ImageButton, and the image is located at middle of ImageButton
fitEnd	Preserve aspect ratio till the image could fill ImageButton, and the image is located at bottom-right corner.
center	Locate the image at the center of ImageButton without zoom
centerCrop	Preserve aspect ratio to zoom image and make image totally cover ImageButton
centerInside	Preserve aspect ratio to zoom image and make ImageButton totally display image

Small Demo



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@mipmap/bg"
    android:gravity="bottom|center_horizontal"
    android:orientation="vertical"
    android:paddingBottom="20dp"
    tools:context="com.example.startgamebutton.MainActivity">
    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/start"
        android:background="#0000"
        android:src="@mipmap/bt_start"/>
    <ImageButton
        android:id="@+id/switch1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#0000"
        android:src="@mipmap/bt_switch"
        android:layout_marginTop="10dp"/>
</LinearLayout>
```


Small Demo



```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
            WindowManager.LayoutParams.FLAG_FULLSCREEN);  
        ImageButton st = (ImageButton)findViewById(R.id.start);  
        st.setOnClickListener(new View.OnClickListener(){  
            @Override  
            public void onClick(View v) {  
                Toast.makeText(MainActivity.this,  
                    "You have pressed start button",  
                    Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}
```



RadioButton

<RadioButton

android:text="displayed text"
android:id="@+id/RadioButtonID"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:checked="true|false"

>

</RadioButton>

<RadioGroup

android:id="@+id/RadioGroupID"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:orientation="horizontal"

>

<!--Multiple radio buttons-->

</RadioGroup>

Small Demo

1:00

LTE

Radio Button

One day, one customer came into John's shop, and he picked up a product of \$25 and paid \$100.
However, John didn't have enough change, and he went to the neighbour shop owned by Simon to make the \$100 into changes and paid \$75 to the customer.
A moment later, Simon came to John's and told that the \$100 bill is fake. John thus offered another \$100 to Simon. Please answer how much money did John lose?

☐ A: \$125

☒ B: \$100

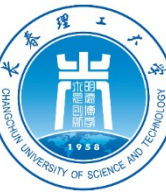
☐ C: \$175

☐ D: \$200

SUBMIT

Correct!

```
<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="com.example.radiobutton.MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=" One day, one customer came into John's shop, and he picked up a product of $25 and paid $100.\n However, John
didn't have enough change, and he went to the neighbour shop owned by Simon to make the $100 into changes
and paid $75 to the customer.\n A moment later, Simon came to John's and told that the $100 bill is fake. John thus offered another
$100 to Simon. Please answer how much money did John lose?"
        android:textSize="16sp"/>
    <RadioGroup
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/radioGroup1">
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/rb_a"
            android:text="A: $125"/>
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/rb_b"
            android:text="B: $100"/>
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/rb_c"
            android:text="C: $175"/>
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/rb_d"
            android:text="D: $200"/>
    </RadioGroup>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/bt_submit"
        android:text="Submit"/>
</LinearLayout>
```



Small Demo

1:00

LTE

Radio Button

One day, one customer came into John's shop, and he picked up a product of \$25 and paid \$100.
However, John didn't have enough change, and he went to the neighbour shop owned by Simon to make the \$100 into changes and paid \$75 to the customer.
A moment later, Simon came to John's and told that the \$100 bill is fake. John thus offered another \$100 to Simon. Please answer how much money did John lose?

☐ A: \$125

☒ B: \$100

☐ C: \$175

☐ D: \$200

SUBMIT

Correct!

```
public class MainActivity extends AppCompatActivity {
    Button bt_submit;
    RadioGroup rg;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        bt_submit = (Button)findViewById(R.id.bt_submit);
        rg = (RadioGroup)findViewById(R.id.radioGroup1);
        bt_submit.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                for(int i = 0; i < rg.getChildCount(); i++){
                    RadioButton rb = (RadioButton)rg.getChildAt(i);
                    if(rb.isChecked()){
                        if(rb.getText().equals("B: $100")){
                            Toast.makeText(MainActivity.this, "Correct!", Toast.LENGTH_LONG).show();
                        }
                        else{
                            AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
                            builder.setMessage("Wrong Answer!");
                            builder.setPositiveButton("OK", null).show();
                        }
                        break;
                    }
                }
            }
        });
    }
}
```



CheckBox

<CheckBox

android:text="displayed text"

android:id="@+id/CheckBoxID"

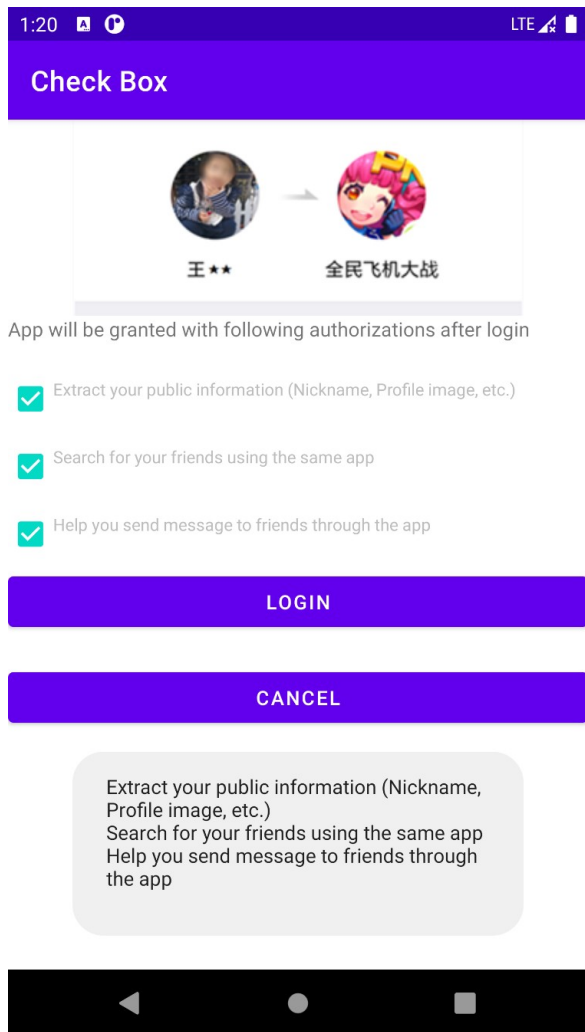
android:layout_height="wrap_content"

android:layout_width="wrap_content"

>

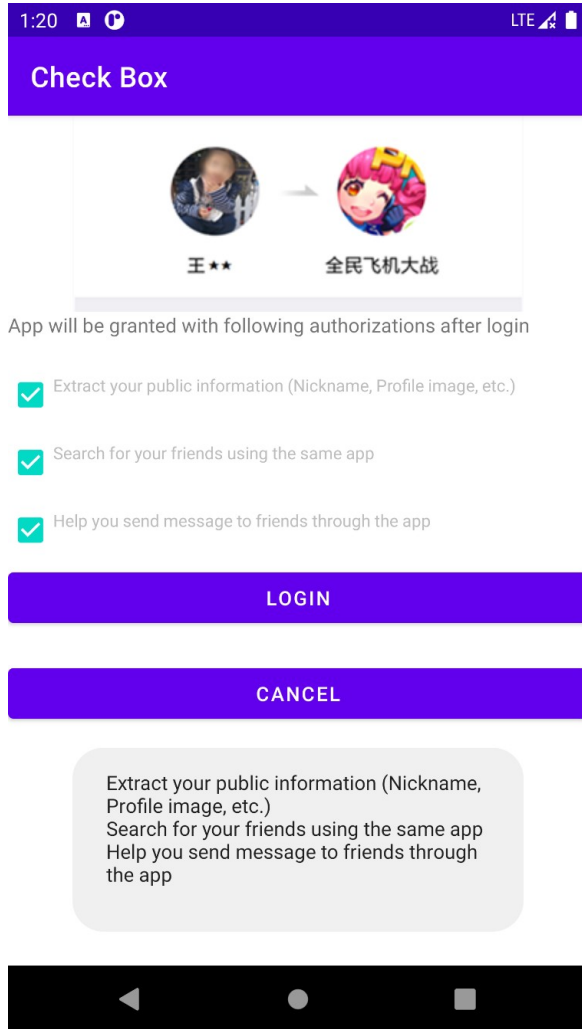
</CheckBox>

Small Demo



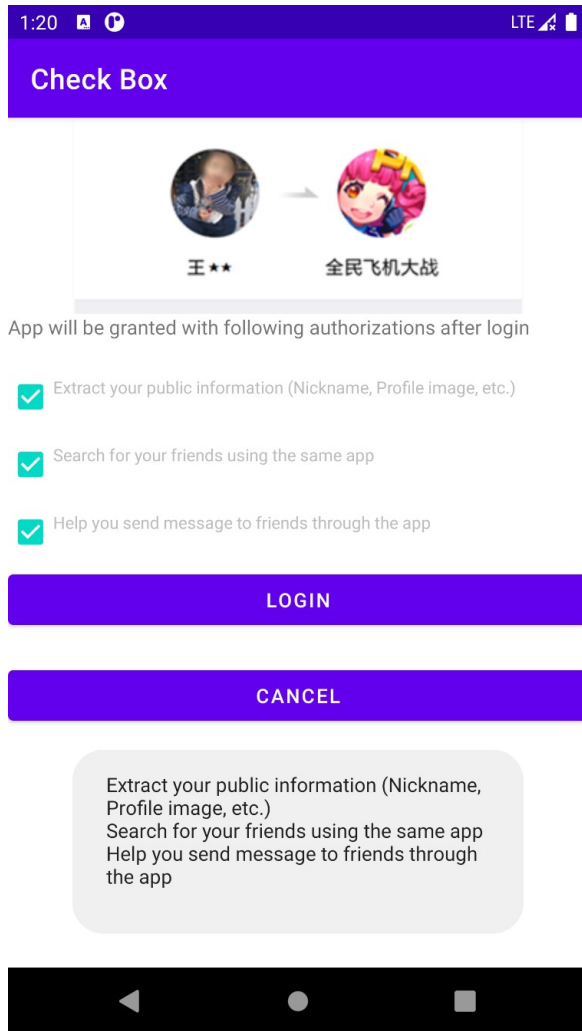
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.checkbox.MainActivity">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/feiji_top"
    />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="App will be granted with following authorizations after login\n"
        android:textSize="14sp"
    />
    <CheckBox
        android:id="@+id/checkbox1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Extract your public information (Nickname, Profile image, etc.)\n"
        android:checked="true"
        android:textSize="12sp"
    android:textColor="#BDBDBD"/>
    <CheckBox
        android:id="@+id/checkbox2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Search for your friends using the same app\n"
        android:checked="true"
        android:textSize="12sp"
        android:textColor="#BDBDBD"/>
    <CheckBox
        android:id="@+id/checkbox3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Help you send message to friends through the app\n"
        android:checked="true"
        android:textSize="12sp"
    android:textColor="#BDBDBD"/>
```

Small Demo



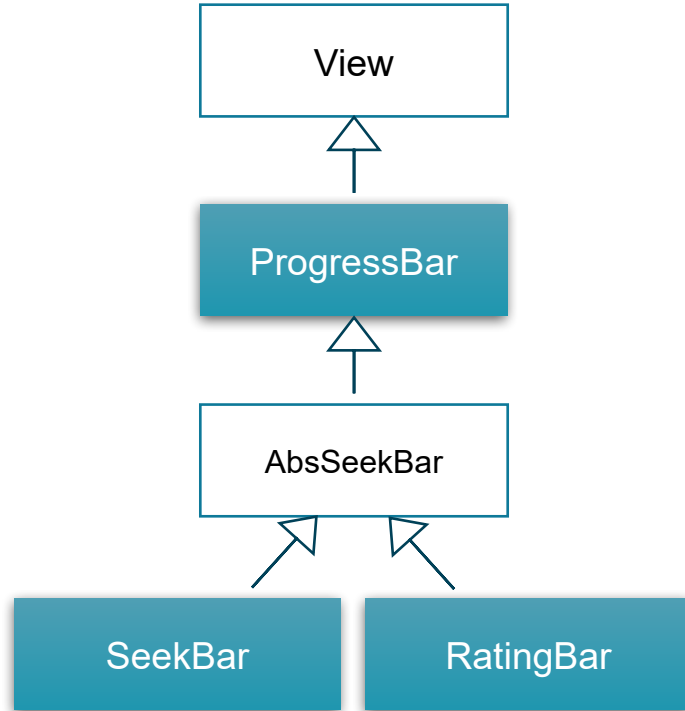
```
<Button
    android:id="@+id/btn_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#009688"
    android:text="Login"/>
<Button
    android:layout_marginTop="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF" android:text="Cancel"/>
</LinearLayout>
```

Small Demo



```
public class MainActivity extends AppCompatActivity {
    Button btn_login;
    CheckBox checkBox1, checkBox2, checkBox3;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn_login = (Button) findViewById(R.id.btn_login);
        checkBox1 = (CheckBox) findViewById(R.id.checkbox1);
        checkBox2 = (CheckBox) findViewById(R.id.checkbox2);
        checkBox3 = (CheckBox) findViewById(R.id.checkbox3);
        btn_login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String checked = "";
                if (checkBox1.isChecked()) {
                    checked += checkBox1.getText().toString();
                }
                if (checkBox2.isChecked()) {
                    checked += checkBox2.getText().toString();
                }
                if (checkBox3.isChecked()) {
                    checked += checkBox3.getText().toString();
                }
                Toast.makeText(MainActivity.this, checked, Toast.LENGTH_LONG).show();
            }
        });
    }
}
```


ProgressBar component



ProgressBar

```
<ProgressBar  
    attribute list  
>  
</ProgressBar>
```

Attribute	Description
android:max	Sets the maximum value of progress bar
android:progress	Specifies the passed progress value
android:progressDrawable	Sets the drawable style of the progress bar channel

□ Style attributes of ProgressBar

Attribute	Description
?android:attr/progressBarStyleHorizontal	Thin horizontal progress bar
?android:attr/progressBarStyleLarge	Big round progress bar
?android:attr/progressBarStyleSmall	Small round progress bar
@android:style/Widget.ProgressBar.Large	Big interval round progress bar with rolling pattern
@android:style/Widget.ProgressBar.Small	Small interval round progress bar with rolling pattern
@android:style/Widget.ProgressBar.Horizontal	Thick horizontal progress bar

Small Demo



```
<RelativeLayout 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="@mipmap/xxll"
    android:paddingBottom="16dp"
    android:paddingTop="16dp"
    android:paddingRight="16dp"
    android:paddingLeft="16dp"
    tools:context="com.example.horizontalprogressbar.MainActivity">
    <ProgressBar
        android:layout_width="match_parent"
        android:layout_height="25dp"
        android:id="@+id/progressBar1"
        style="@android:style/Widget.ProgressBar.Horizontal"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="60dp"
        android:max="100"/>
</RelativeLayout>
```

Small Demo



```
public class MainActivity extends Activity {
    private ProgressBar horizontalPro;
    private int mProgressStatus = 0;
    private Handler mHandler;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        horizontalPro = (ProgressBar)findViewById(R.id.progressBar1);
        mHandler = new Handler(){
            public void handleMessage(Message msg){
                if(msg.what == 0x111){
                    horizontalPro.setProgress(mProgressStatus);
                }else{
                    Toast.makeText(MainActivity.this, "The progress has been finished!",
                        Toast.LENGTH_SHORT).show();
                    horizontalPro.setVisibility(View.GONE);
                }
            }
        };
    }
}
```

```
new Thread(new Runnable() {
    @Override
    public void run() {
        while(true){
            mProgressStatus = doWork();
            Message m = new Message();
            if(mProgressStatus < 100){
                m.what = 0x111;
                mHandler.sendMessage(m);
            }else{
                m.what = 0x110;
                mHandler.sendMessage(m);
                break;
            }
        }
    }
    private int doWork(){
        mProgressStatus += Math.random()*10;
        try{
            Thread.sleep(200);
        }catch(InterruptedException e){
            e.printStackTrace();
        }
        return mProgressStatus;
    }
}).start();
}
```

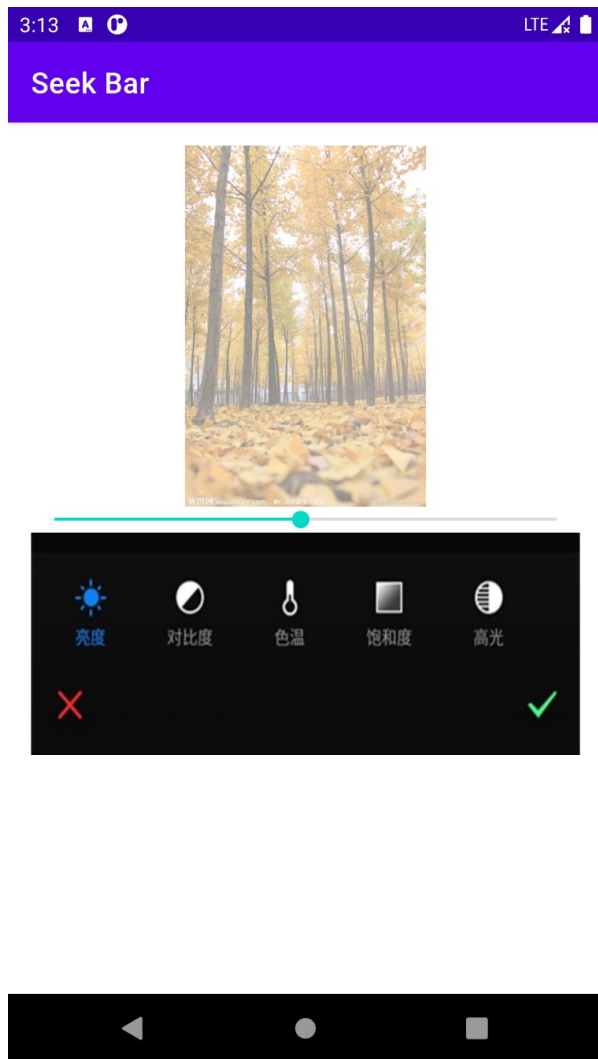
SeekBar

```
<SeekBar  
    android:layout_height="wrap_content"  
    android:id="@+id/seekBarID"  
    android:layout_width="match_parent"  
>  
</SeekBar>
```

- ❑ SeekBar enables users to alternate the outlook with adjusting “android:thumb” attribute
 - ❑ The attribute value is a Drawable object that provides the customized sliders

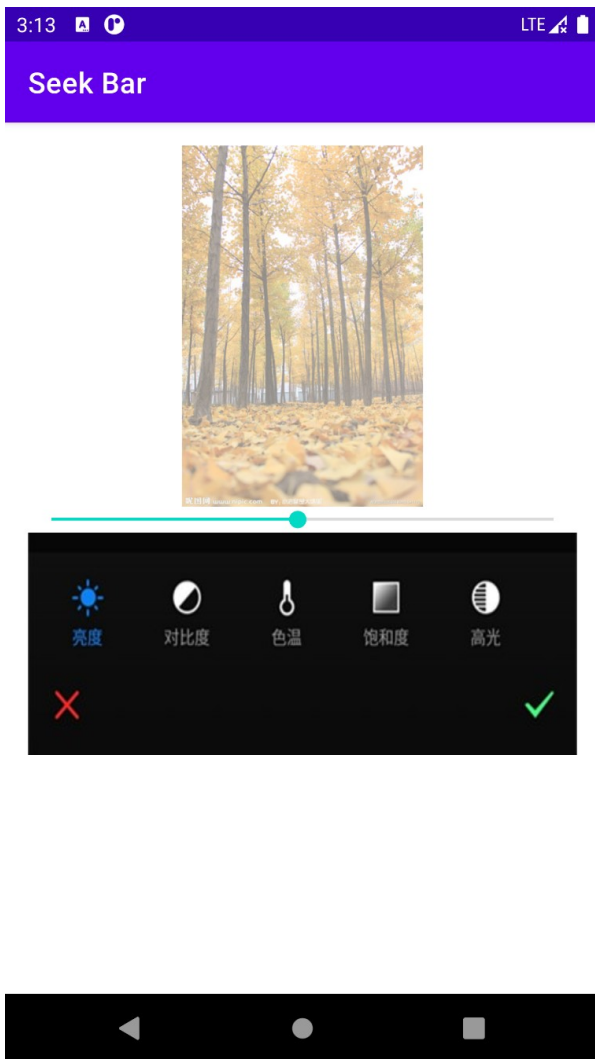
```
SeekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener(){  
    public void onStopTrackingTouch(SeekBar seekBar){  
        //code to execute  
    }  
  
    public void onStartTrackingTouch(SeekBar seekBar){  
        //code to execute  
    }  
  
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser){  
        //code to execute  
    }  
}
```

Small Demo



```
<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:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.seekbar.MainActivity">
    <ImageView
        android:id="@+id/image"
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:src="@mipmap/lijiang"/>
    <SeekBar
        android:id="@+id/seekbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:max="255"
        android:progress="255" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="fitXY"
        android:src="@mipmap/meitu"/>
</LinearLayout>
```


Small Demo



```
public class MainActivity extends AppCompatActivity {
    private ImageView image;
    private SeekBar seekBar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        image = (ImageView)findViewById(R.id.image);
        seekBar = (SeekBar)findViewById(R.id.seekbar);
        seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                image.setImageAlpha(progress);
            }
            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
            }
            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
            }
        });
    }
}
```

RatingBar

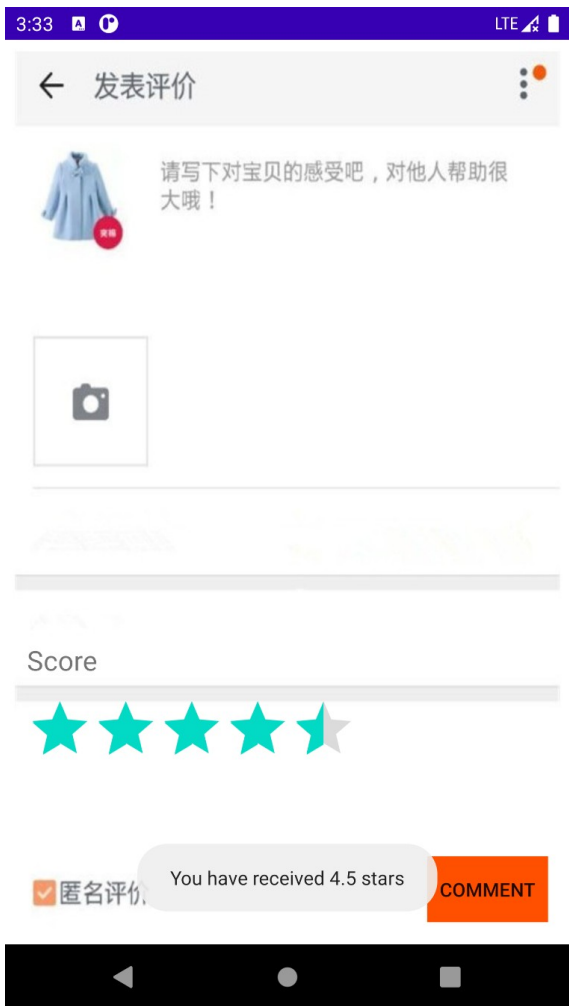
```
<RatingBar  
    Attribute list  
>  
</RatingBar>
```

Attribute	Description
android:isIndicator	Specifies the bar could be changed by users, true: not allow to change
android:numStars	Specifies the number of stars
android:rating	Specifies the default number of stars
android:stepSize	Specifies the interval of adjusting levels, 0.5 star by default

□ Common methods

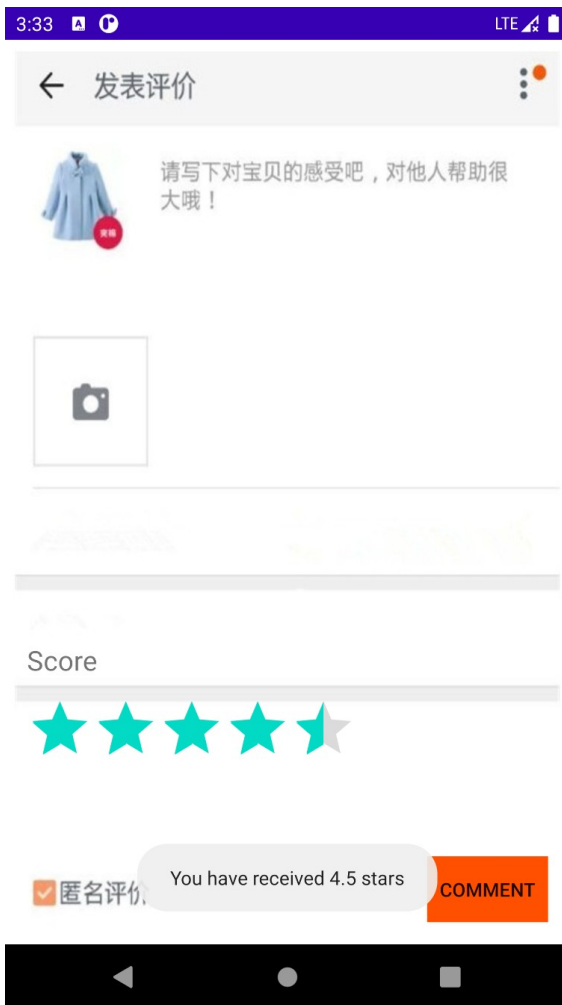
- `getRating()`
- `getStepSize()`
- `getProgress()`

Small Demo



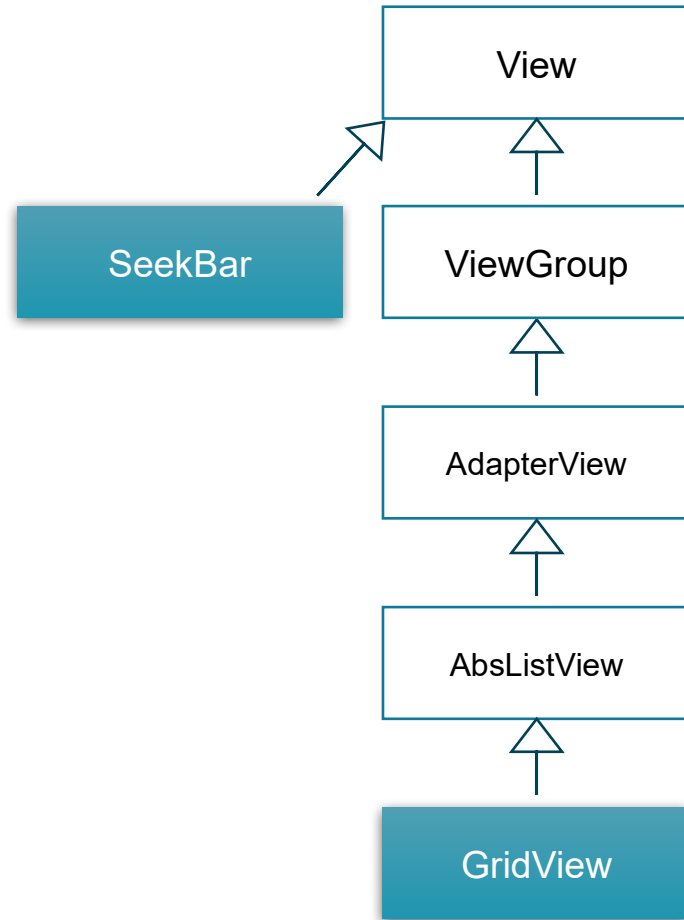
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@mipmap/xing1"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.starrating.MainActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/btn"
        android:layout_marginBottom="130dp"
        android:text="Score"
        android:textSize="20sp" />
    <RatingBar
        android:id="@+id/ratingBar1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/btn"
        android:layout_marginBottom="60dp"
        android:numStars="5"
        android:rating="0" />
    <Button
        android:id="@+id/btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:background="#FF5000"
        android:text="Comment" />
</RelativeLayout>
```

Small Demo



```
public class MainActivity extends Activity {
    private RatingBar ratingbar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ratingbar = (RatingBar)findViewById(R.id.ratingBar1);
        Button button = (Button)findViewById(R.id.btn);
        button.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                int result = ratingbar.getProgress();
                float rating = ratingbar.getRating();
                float step = ratingbar.getStepSize();
                Log.i("Star rating bar", "step="+step+" result="+result+" rating="+rating);
                Toast.makeText(MainActivity.this,
                    "You have received "+rating+ " stars", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Image View Component



ImageView

```
<ImageView  
    Attribute list  
>  
</ImageView>
```

Attribute	Description
android:adjustViewBounds	Sets whether to preserve the aspect ratio through adjusting the bounds of ImageView
android:maxHeight/ android:maxWidth	Sets maximum height/width in condition that android:adjustViewBounds=true, otherwise the attribute will not work
android:scaleType	Set zooming and locating pattern to display ImageView in proper size. Attribute value could be: maxtrix, fitXY, fitStart, fitCenter, fitEnd, center, centerCrop
android:src	Sets the drawable object ID to be displayed, e.g., android:src="@drawable/photo"
android:tint	Makes color onto image. Attribute value could be written in form of #rgb, #argb, #rrggbb or #aarrggbb

Small Demo



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.imageview">
    <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.UIComponents">
        <activity android:name=".MainActivity" android:screenOrientation="landscape">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```
<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:paddingLeft="16dp"
    android:paddingTop="16dp"
    android:paddingRight="16dp"
    android:paddingBottom="16dp"
    tools:context="com.example.imageview.MainActivity">
    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:src="@mipmap/flower" />
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:adjustViewBounds="true"
        android:maxWidth="90dp"
        android:maxHeight="90dp"
        android:src="@mipmap/flower" />
    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="90dp"
        android:layout_height="90dp"
        android:layout_margin="5dp"
        android:scaleType="fitEnd"
        android:src="@mipmap/flower" />
    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="90dp"
        android:layout_height="90dp"
        android:src="@mipmap/flower"
        android:tint="#77ff0000" />
</LinearLayout>
```

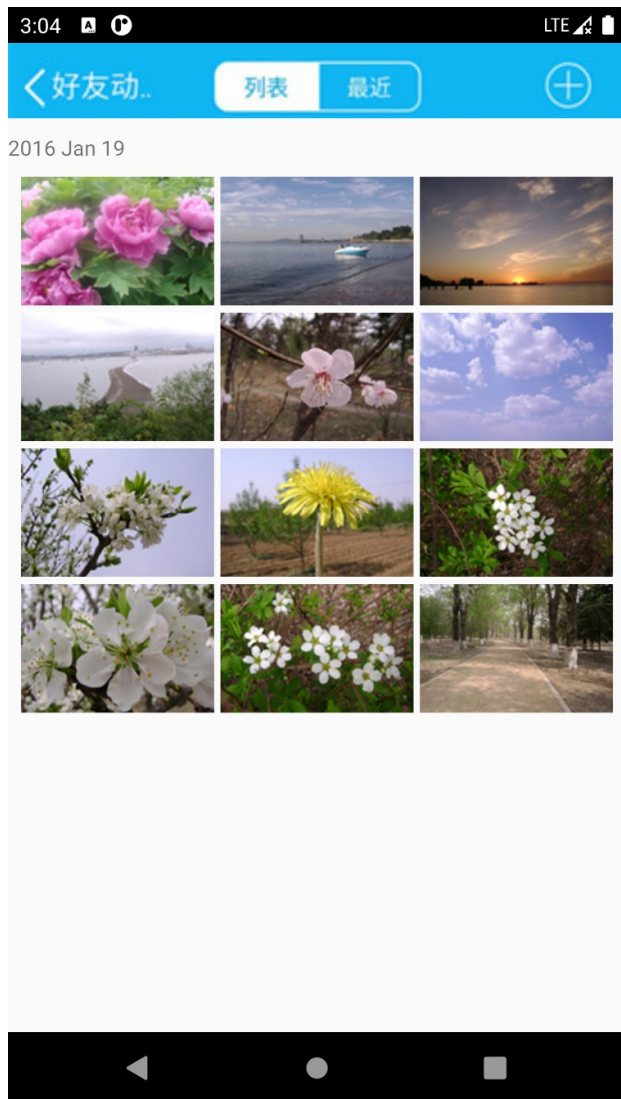
GridView

```
<GridView  
    Attribute list  
>  
</GridView>
```

Attribute	Description
android:columnWidth	Sets width of columns
android:gravity	Sets alignment pattern
android:horizontalSpacing	Sets the horizontal space between the components
android:numColumns	Sets the number of columns. If it is set as 1, you should use ListView instead of GridView
android:stretchMode	Sets the pattern of stretching view. The value could be none, spacingWidth (i.e., space between components), columnWidth (i.e., grid cells), spacingWidthUniform (i.e., both space and cell)

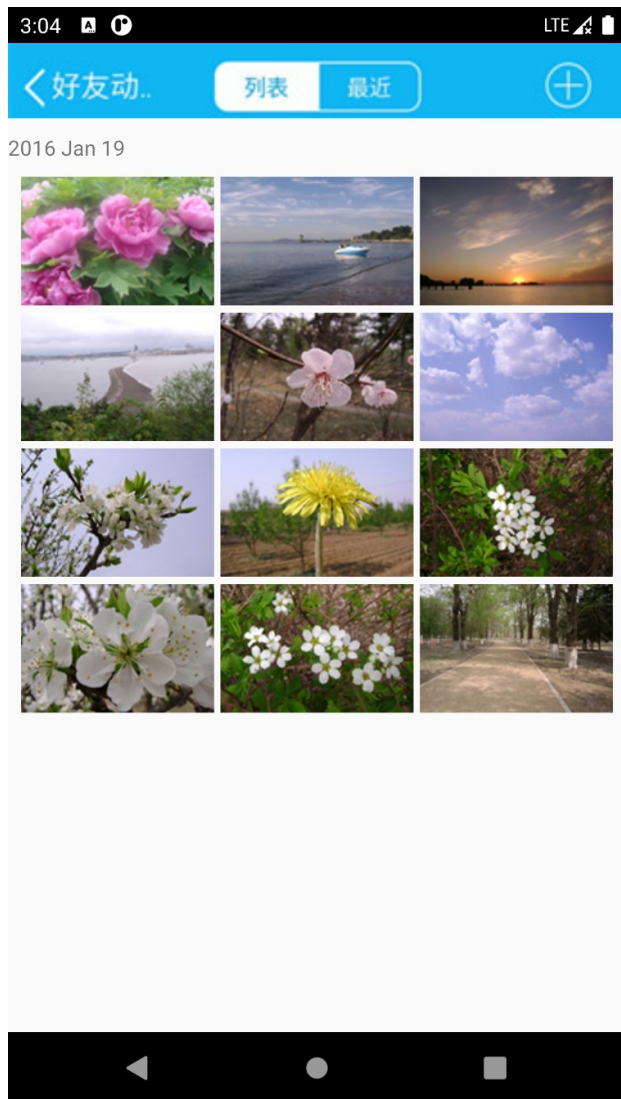
- When using GridView, Adapter class (it is in fact an interface) is often employed to offer the relevant data, the implementing classes are:
 - ArrayAdapter: wraps multiple values of array into several list items and only displays one-line text
 - SimpleAdapter: wraps multiple values of List set into several list items. It is powerful and easy-to-customized
 - SimpleCursorAdapter: must align the Cursor attribute to component ID. It could display the content in database as list

Small Demo



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.qqalbum.MainActivity">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/qqxiang"
        android:scaleType="centerCrop"
        android:adjustViewBounds="true"
        android:maxLength="1440dp" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:text="2016 Jan 19" />
    <GridView
        android:id="@+id/gridview1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:columnWidth="120dp"
        android:gravity="center"
        android:numColumns="auto_fit"
        android:stretchMode="spacingWidthUniform"
        android:verticalSpacing="5dp">
    </GridView>
</LinearLayout>
```


Small Demo



```
public class MainActivity extends Activity {
    private Integer[] picture = {R.mipmap.img01,R.mipmap.img02,R.mipmap.img03,R.mipmap.img04,
        R.mipmap.img05,R.mipmap.img06,R.mipmap.img07,R.mipmap.img08,R.mipmap.img09,
        R.mipmap.img10,R.mipmap.img11,R.mipmap.img12,};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        GridView gridView = (GridView)findViewById(R.id.gridview1);
        gridView.setAdapter(new ImageAdapter(this));
    }

    public class ImageAdapter extends BaseAdapter{
        private Context mContext;
        public ImageAdapter(Context c){
            mContext = c;
        }
        public int getCount(){
            return picture.length;
        }
        public Object getItem(int position){
            return null;
        }
        public long getItemId(int position){
            return 0;
        }
        public View getView(int position, View convertView, ViewGroup parent){
            ImageView imageView;
            if(convertView == null){
                imageView = new ImageView(mContext);
                imageView.setLayoutParams(new ViewGroup.LayoutParams(450,300));
                imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
            }
            else{
                imageView = (ImageView)convertView;
            }
            imageView.setImageResource(picture[position]);
            return imageView;
        }
    }
}
```



Listview

<ListView

Attribute list

>

</ListView>

❑ Load data in ListView

❑ Solution 1: In res/values folder, you could establish a xml file containing string array to load entries in ListView

❑ Solution 2:

❑ Create Adapter object. ArrayAdapter object is often used to load list of text.

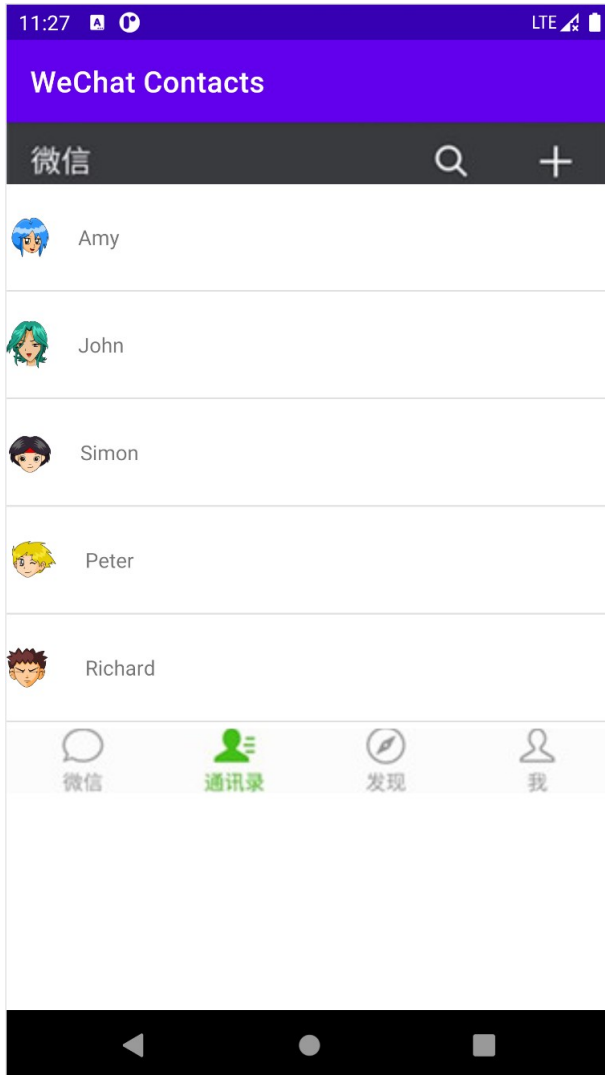
❑ simple_list_item_1, simple_list_item_2, simple_list_item_checked, simple_list_item_multiple_choice, simple_list_item_single_choice

❑ Link the created adapter to the ListView object with setAdapter() method.

❑ ListView.setAdapter(adapter);

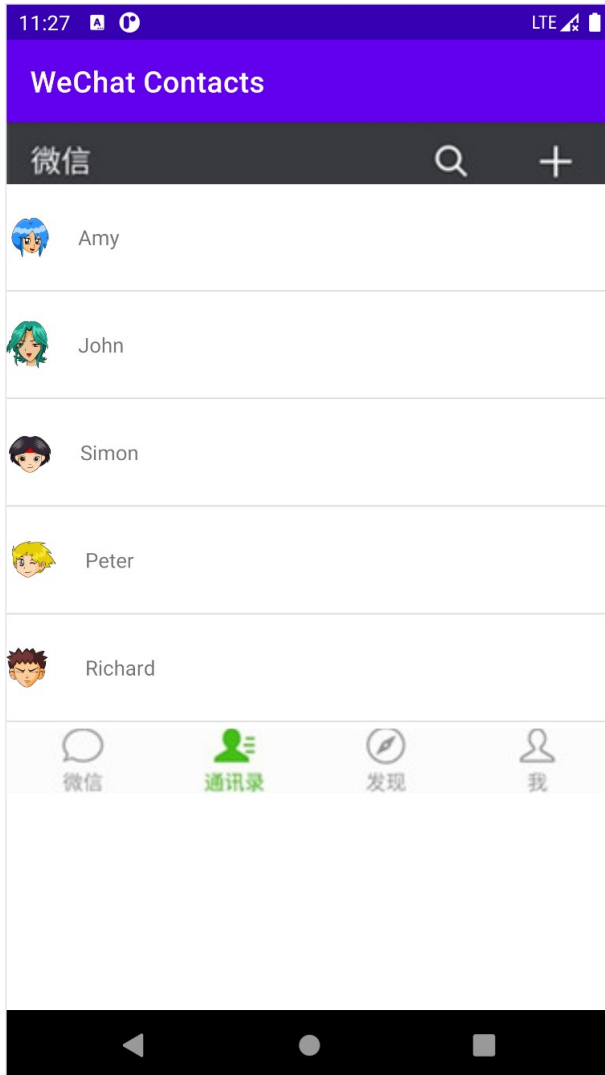
Attribute	Description
android:divider	Sets divider pattern in ListView, the divider can be either colored line or resource from Drawable folder
android:dividerHeight	Sets the height of divider
android:entries	Sets items in list from parsing array resource
android:footerDividersEnabled	Sets whether to draw dividers before rendering footer view, true by default. When using this attribute, you can configure footer view with calling addFooterView() method of ListView class.
android:headerDividersEnabled	Sets whether to draw dividers before rendering header view, true by default. When using this attribute, you can configure footer view with calling addHeaderView() method of ListView class.

Small Demo



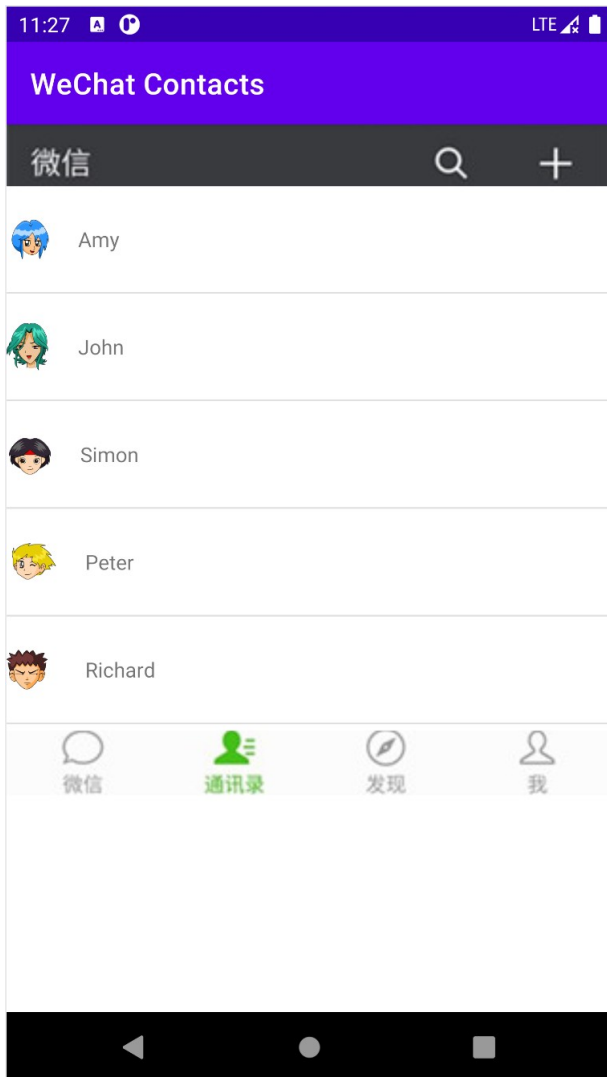
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:id="@+id/image"
        android:paddingRight="10dp"
        android:paddingTop="20dp"
        android:paddingBottom="20dp"
        android:adjustViewBounds="true"
        android:maxLength="72dp"
        android:maxLength="72dp"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_gravity="center"
        android:id="@+id/title"
    />
</LinearLayout>
```

Small Demo



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.wechatcontacts.MainActivity">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/wei_top"
        android:scaleType="centerCrop"/>
    <ListView
        android:id="@+id/listview"
        android:layout_width="match_parent"
        android:layout_height="370dp">
    </ListView>
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/wei_down"
        android:scaleType="centerCrop"/>
</LinearLayout>
```

Small Demo



```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ListView listview = (ListView) findViewById(R.id.listview);
        int[] imageId = new int[]{R.mipmap.img01, R.mipmap.img02, R.mipmap.img03,
            R.mipmap.img04, R.mipmap.img05, R.mipmap.img06,
            R.mipmap.img07, R.mipmap.img08, R.mipmap.img09,
        };
        String[] title = new String[]{"Amy", "John", "Simon", "Peter", "Richard",
            "Lee", "Eva", "Catherine", "Mozart"};
        List<Map<String, Object>> listItems = new ArrayList<Map<String, Object>>();

        for (int i = 0; i < imageId.length; i++) {
            Map<String, Object> map = new HashMap<String, Object>();
            map.put("image", imageId[i]);
            map.put("Name", title[i]);
            listItems.add(map);
        }
        SimpleAdapter adapter = new SimpleAdapter(this, listItems,
            R.layout.main, new String[]{"Name", "image"}, new int[]{
                R.id.title, R.id.image});
        listview.setAdapter(adapter);
        listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Map<String, Object> map = (Map<String, Object>) parent.getItemAtPosition(position);
                Toast.makeText(MainActivity.this, map.get("Name").
                    toString(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

