

I. GIỚI THIỆU

1. FrameLayout

Đơn giản là vùng hiển thị 1 nội dung cụ thể nào đó. Nó chứa trong nó 1 thành phần view khác như 1 hình ảnh, 1 nút nhấn, 1 nhãn...

Phần tử con mặc định hiển thị ở góc trên trái của FrameLayout.

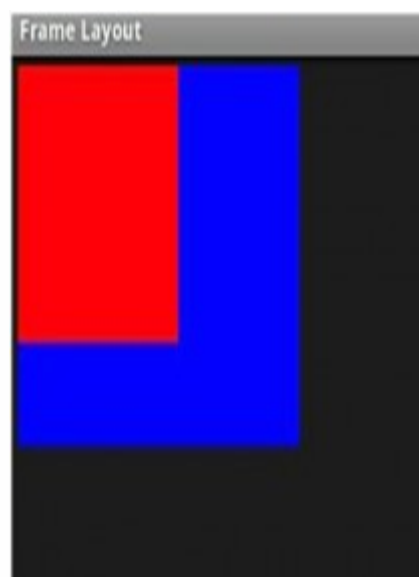
Các phần tử “sau” sẽ nằm chồng lên phần tử “trước” (sau, trước theo thứ tự trong tập tin xml).

```
<FrameLayout android:id="@+id/mainlayout" android:layout_height="match_parent"
android:layout_width="match_parent" android:orientation="vertical"
xmlns:android=http://schemas.android.com/apk/res/android

<ImageView android:layout_height="wrap_content" android:layout_width="wrap_content"
android:padding="5px" android:src="@drawable/blue"/>

<ImageView android:layout_height="wrap_content" android:layout_width="wrap_content"
android:padding="5px" android:src="@drawable/red"/>

</FrameLayout>
```



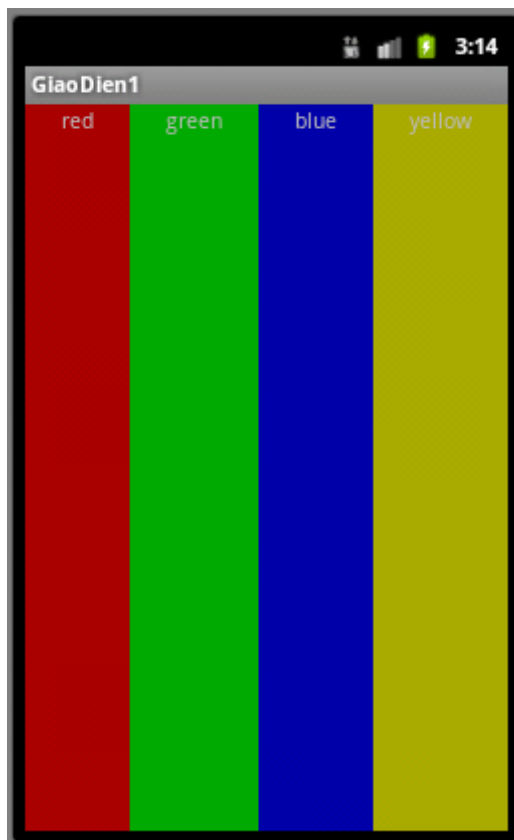
Chú ý 2 dòng lệnh bên dưới:

- android:src="@drawable/**blue**"
- android:src="@drawable/**red**"

Là do ta kéo 2 cái hình tên là **blue** và **red** vào thư mục drawable của ứng dụng.

2. LinearLayout

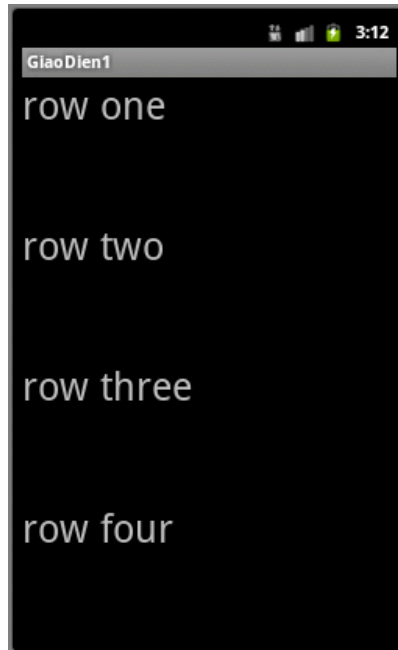
Layout này hay được sử dụng nhiều nhất khi làm ứng dụng. Nó đơn giản cho phép ta sắp xếp các phần tử trong nó theo dạng danh sách dọc hoặc ngang. Để chỉ LinearLayout này bố trí theo chiều dọc hay chiều ngang ta sử dụng thuộc tính **android:orientation** với các giá trị lần lượt là **vertical** hoặc **horizontal**. Để các phần tử con nằm trong LinearLayout có độ rộng tương đối so với nhau ta sử dụng thuộc tính android: **layout_weight** và thiết lập giá trị là con số tương ứng. ***Ví dụ sau tạo một giao diện gồm 4 cột có kích thước bằng nhau và có màu sắc khác nhau:***



```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1">
    <TextView
        android:text="red"
        android:gravity="center_horizontal"
        android:background="#aa0000"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"/>
    <TextView
        android:text="green"
        android:gravity="center_horizontal"
        android:background="#00aa00"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"/>
    <TextView
```

```
        android:text="blue"
        android:gravity="center_horizontal"
        android:background="#0000aa"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"/>
<TextView
    android:text="yellow"
    android:gravity="center_horizontal"
    android:background="#aaaa00"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"/>
</LinearLayout>
```

Ví dụ tiếp sau đây tạo một giao diện gồm 4 hàng, mỗi hàng là 1 dòng chữ:

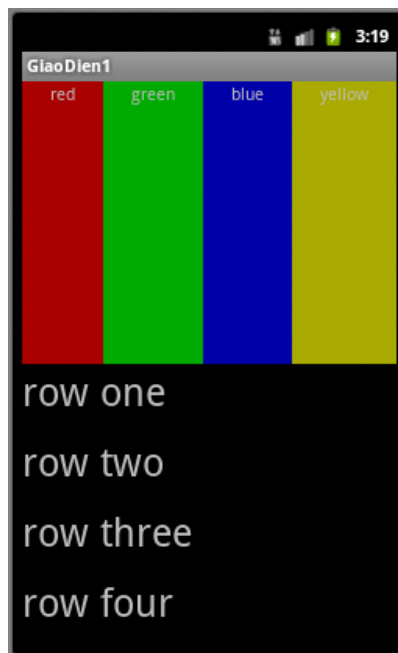


```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1">
    <TextView
        android:text="row one"
        android:textSize="15pt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"/>
    <TextView
        android:text="row two"
        android:textSize="15pt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"/>
    <TextView
        android:text="row three"
        android:textSize="15pt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:layout_weight="1"/>
    <TextView
        android:text="row four"
        android:textSize="15pt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"/>
</LinearLayout>
```

Chúng ta cũng có thể lồng LinearLayout này vào LinearLayout kia để tạo ra một giao diện kết hợp phức tạp hơn.

Ví dụ sau sẽ đặt 2 LinearLayout nằm trong 1 LinearLayout khác



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1">
```

Nội dung tài liệu có tham khảo từ nhiều nguồn từ internet

```
<TextView
    android:text="red"
    android:gravity="center_horizontal"
    android:background="#aa0000"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"/>
<TextView
    android:text="green"
    android:gravity="center_horizontal"
    android:background="#00aa00"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"/>
<TextView
    android:text="blue"
    android:gravity="center_horizontal"
    android:background="#0000aa"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"/>
<TextView
    android:text="yellow"
    android:gravity="center_horizontal"
    android:background="#aaaa00"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"/>
</LinearLayout>

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1">
    <TextView
        android:text="row one"
        android:textSize="15pt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"/>
    <TextView
        android:text="row two"
        android:textSize="15pt"
        android:layout_width="match_parent
```

Nội dung tài liệu có tham khảo từ nhiều nguồn từ internet

```
        android:layout_height="wrap_content"
        android:layout_weight="1"/>
<TextView
    android:text="row three"
    android:textSize="15pt"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"/>
<TextView
    android:text="row four"
    android:textSize="15pt"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"/>
</LinearLayout>
</LinearLayout>
```

Một số giải thích cho các thuộc tính của các thành phần trong 3 ví dụ trên:

1.	android:gravity	canh nội dung nằm trong 1 thành phần
2.	android:layout_width	độ rộng của thành phần
3.	android:layout_height	độ cao của thành phần
4.	android:layout_weight	độ “rộng” của thành phần so với thành phần khác

Một số giải thích cho các giá trị thuộc tính được sử dụng trong 3 ví dụ trên:

1.	match_parent	Giãn rộng bằng layout cha chứa nó
2.	wrap_content	Co nhỏ vừa nội dung nó chứa
3.	Về layout_weight, giả sử ta có 2 cột mà muốn cột thứ nhất rộng gấp đôi cột thứ 2 thì ta sẽ thiết lập để giá trị layout_weight của cột thứ nhất lớn gấp đôi của cột thứ 2 (ví dụ cột thứ nhất là 2, cột thứ 2 là 1. Hoặc của cột thứ nhất là 4, của cột thứ 2 là 2).	

3. TableLayout

Kết hợp với thẻ TableRow để tạo ra các hàng các cột cho layout. Cách sử dụng cũng khá đơn giản. Khi dùng TabLayout, mặc định sẽ tạo ra các đường viền xung quanh các ô. Các ô có thể chứa nội dung là bất kỳ thành phần view nào khác

Nội dung tài liệu có tham khảo từ nhiều nguồn từ internet

(có thể chứa LinearLayout, FrameLayout hoặc TableLayout cũng được). Mỗi thẻ TableRow nằm trong thẻ TableLayout tạo ra một hàng. Thẻ này sẽ thực sự sẽ chứa đựng các thành phần view khác.

Để xem cách dùng, ta sẽ thử làm ví dụ sau để tạo ra một menu cho ứng dụng bằng TableLayout:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="1">
    <TableRow>
        <TextView
            android:text="Open..."
            android:padding="3dip" />
        <TextView
            android:text="Save AS..."
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>

    <TableRow>
        <TextView
            android:text="Close..."
            android:padding="3dip" />
        <TextView
            android:text="More..."
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
</TableLayout>
```

Ta được kết quả hiển thị như sau:



4. RelativeLayout

Đây là loại Layout cho phép chúng ta thiết lập mối liên hệ hiển thị giữa các thành phần con với nhau. Nói cách khác, một thành phần con có thể được định vị vị trí của nó so với thành phần chứa nó hoặc các thành phần cạnh nó.

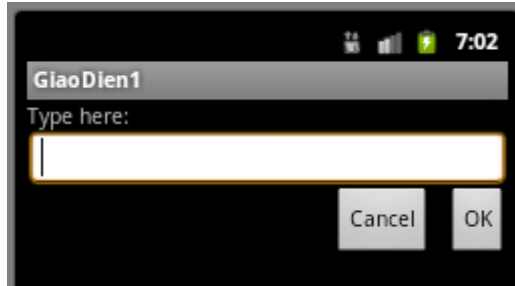
Ví dụ đơn giản sau đây sẽ sử dụng một thuộc tính xml để canh một nút nhấn sang bên phải thành phần đang chứa nó.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:id="@+id/label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Type here:"/>
    <EditText
        android:id="@+id/entry"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@android:drawable/editbox_background"
        android:layout_below="@id/label"/>
    <Button
        android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/entry"
        android:layout_alignParentRight="true"
        android:layout_marginLeft="10dip">
```

Nội dung tài liệu có tham khảo từ nhiều nguồn từ internet

```
        android:text="OK" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toLeftOf="@id/ok"
        android:layout_alignTop="@id/ok"
        android:text="Cancel" />
</RelativeLayout>
```

Chúng ta có kết quả hiển thị như sau:



Trong ví dụ đơn giản trên. Chúng ta khai báo 1 RelativeLayout, trong đó đặt 4 thành phần con gồm 1 TextView, 1 EditText, và 2 Button. Ở EditText chúng ta có thiết lập thuộc tính:

```
android:layout_below="@id/label"
```

Thuộc tính này khai báo rằng EditText sẽ xuất hiện kế tiếp sau phần tử có id là **label**

Ở Button thứ nhất (có nhãn là Ok) chúng ta có thiết lập thuộc các thuộc tính:

```
android:layout_below="@id/entry"  
android:layout_alignParentRight="true"
```

Thuộc tính này làm cho nút Ok xuất hiện ngay sau phần tử có id là entry và đồng thời bị kéo sang bên phải của RelativeLayout.

Bảng danh sách các thuộc tính canh phần tử xuất hiện như thế nào so với các phần tử khác được liệt kê phía sau:

Tên thuộc tính	Mô tả
android:layout_above	Đặt phần tử hiện tại nằm kế sau phần tử có id được chỉ ra
android:layout_alignBaseline	Đặt phần tử này lên cùng dòng với phần tử có id được chỉ ra
android:layout_alignBottom	Canh sao cho đáy của phần tử hiện tại trùng với đáy của phần tử có id được chỉ ra
android:layout_alignLeft	Đặt cạnh trái của phần tử hiện tại trùng với cạnh trái

Nội dung tài liệu có tham khảo từ nhiều nguồn từ internet

	của phần tử có id được chỉ ra
android:layout_alignParentBottom	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh xuống đáy của phần tử chứa nó
android:layout_alignParentLeft	Nếu được thiết lập là true thì phần tử hiện thời sẽ canh trái so với phần tử chứa nó
android:layout_alignParentRight	Nếu được thiết lập là true thì phần tử hiện thời sẽ canh phải so với phần tử chứa nó
android:layout_alignParentTop	Nếu được thiết lập là true thì phần tử hiện thời sẽ canh lên đỉnh phần tử chứa nó
android:layout_alignRight	Canh cạnh phải của phần tử hiện thời trùng với cạnh phải của phần tử có id được chỉ ra
android:layout_alignTop	Canh đỉnh của phần tử hiện thời trùng với đỉnh của phần tử có id được chỉ ra
android:layout_alignWithParentIfMissing	Nếu thiết lập là true, thì phần tử sẽ được canh theo phần tử chứa nó nếu các thuộc tính canh của phần tử không có.
android:layout_below	Đặt phần tử hiện thời ngay sau phần tử có id được chỉ ra.
android:layout_centerHorizontal	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh giữa theo chiều ngang phần tử chứa nó.
android:layout_centerInParent	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh chính giữa theo chiều phải trái và trên dưới so với phần tử chứa nó.
android:layout_centerVertical	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh chính giữa theo chiều dọc phần tử chứa nó.

android:layout_toLeftOf	Đặt cạnh phải của phần tử hiện thời trùng với cạnh trái của phần tử có id được chỉ ra.
android:layout_toRightOf	Đặt cạnh trái của phần tử hiện thời trùng với cạnh phải của phần tử có id được chỉ ra.

II. THỰC HÀNH

1. LinearLayout

a) Khởi tạo LinearLayout bằng code

Khai báo biến:

```
private LinearLayout llNameContainer, llAddressContainer, llParentContainer;
```

Định nghĩa các hàm khởi tạo LinearLayout

```
private void createNameContainer() {
    llNameContainer = new LinearLayout(this);
    llNameContainer.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT,
        LayoutParams.WRAP_CONTENT));
    llNameContainer.setOrientation(LinearLayout.HORIZONTAL);

    TextView tvName = new TextView(this);
    tvName.setText("Name: ");
    llNameContainer.addView(tvName);

    TextView tvNameValue = new TextView(this);
    tvNameValue.setText("John Doe");
    llNameContainer.addView(tvNameValue);
}

private void createAddressContainer() {
    llAddressContainer = new LinearLayout(this);
    llAddressContainer.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT,
        LayoutParams.WRAP_CONTENT));
    llAddressContainer.setOrientation(LinearLayout.HORIZONTAL);

    TextView tvAddress = new TextView(this);
    tvAddress.setText("Address:");
    llAddressContainer.addView(tvAddress);

    TextView tvAddressValue = new TextView(this);
    tvAddressValue.setText("911 Hollywood Blvd");
    llAddressContainer.addView(tvAddressValue);
}
```

Nội dung tài liệu có tham khảo từ nhiều nguồn từ internet

```
private void createParentContainer() {  
    llParentContainer = new LinearLayout(this);  
  
    llParentContainer.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT,  
        LayoutParams.MATCH_PARENT));  
    llParentContainer.setOrientation(LinearLayout.VERTICAL);  
  
    llParentContainer.addView(llNameContainer);  
    llParentContainer.addView(llAddressContainer);  
}
```

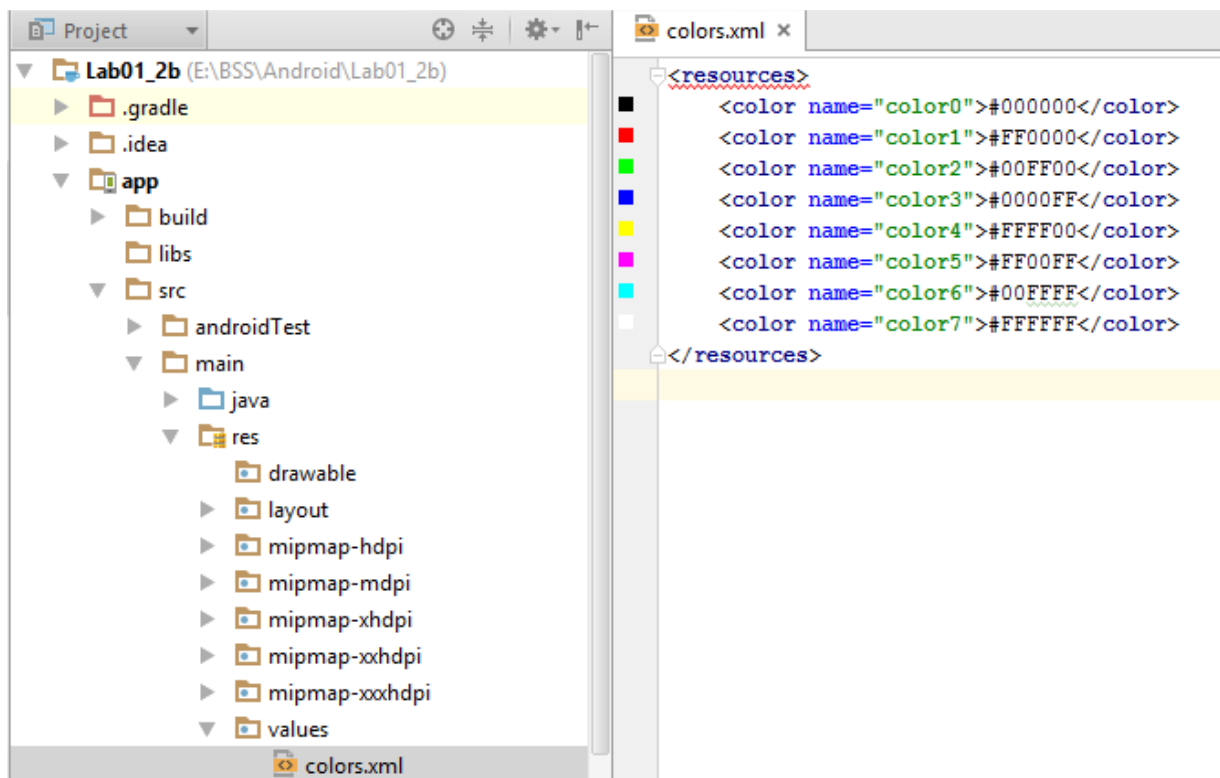
Gọi các hàm vừa tạo trong hàm onCreate()

```
createNameContainer();  
createAddressContainer();  
createParentContainer();  
setContentView(llParentContainer);
```

Chạy ứng dụng và xem kết quả.

b) Sử dụng LinearLayout trong file xml

Tạo project mới, tạo file colors.xml trong thư mục value với nội dung như sau:



Sử dụng đoạn code xml sau để tạo giao diện cho MainActivity:

```
<?xml version="1.0" encoding="utf-8"?>
<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="vn.edu.uit.lab01_2b.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="@color/color0"
        android:orientation="horizontal"></LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2"
        android:orientation="horizontal">

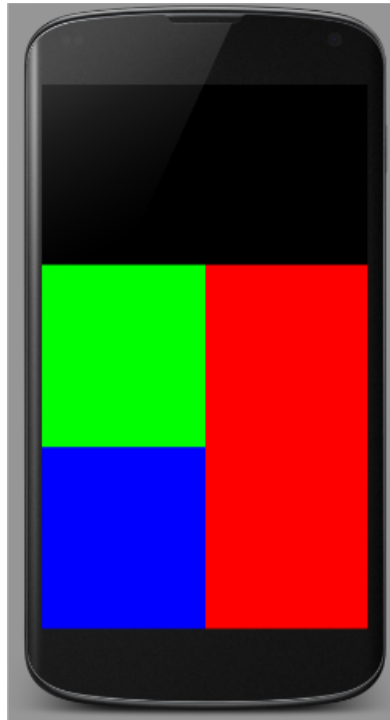
        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_weight="1"
                android:background="@color/color2"
                android:orientation="horizontal"></LinearLayout>

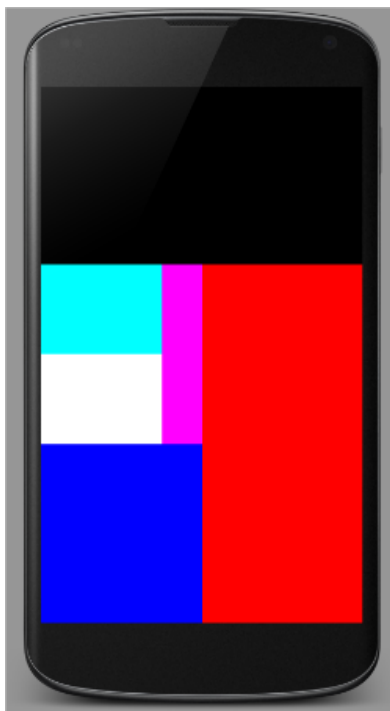
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_weight="1"
                android:background="@color/color3"
                android:orientation="horizontal"></LinearLayout>
            </LinearLayout>
        </LinearLayout>

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="@color/color1"
            android:orientation="vertical"></LinearLayout>
    </LinearLayout>
</LinearLayout>
```


Chạy ứng dụng, ta thấy giao diện hiện tại như sau:



Sửa đổi (bổ sung) nội dung xml để ứng dụng có được giao diện như hình:



2. RelativeLayout

Tạo project mới, nội dung file strings.xml:

```
<resources>
    <string name="app_name">Lab01_2c</string>
    <string name="sign_in">Sign in</string>
    <string name="username">Username: </string>
    <string name="password">Password: </string>
    <string name="signup">Sign up</string>
</resources>
```

Nội dung file dims.xml

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="margin_base">5dp</dimen>
    <dimen name="margin_basex2">10dp</dimen>
    <dimen name="margin_basex3">15dp</dimen>
    <dimen name="margin_basex4">20dp</dimen>
    <dimen name="text_small">16sp</dimen>
    <dimen name="text_medium">18sp</dimen>
    <dimen name="text_large">20sp</dimen>
</resources>
```

Nội dung file activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <TextView
        android:id="@+id/tv_sign_in"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/sign_in"
        android:textAllCaps="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="@dimen/margin_basex4"
        android:textSize="@dimen/text_large"
        />

    <TextView
        android:layout_marginTop="@dimen/margin_basex4"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="@string/username"
```

Nội dung tài liệu có tham khảo từ nhiều nguồn từ internet

```
        android:id="@+id/tv_user_name"
        android:layout_below="@+id/tv_sign_in"
        android:layout_alignParentLeft="true" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/et_username"
        android:hint="name@gmail.com"
        android:layout_alignTop="@+id/tv_user_name"
        android:layout_toRightOf="@+id/tv_user_name"
        />

    <TextView
        android:layout_marginTop="@dimen/margin_base"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="@string/password"
        android:id="@+id/tv_password"
        android:hint="*****"
        android:layout_below="@+id/et_username"
        android:layout_alignParentLeft="true" />

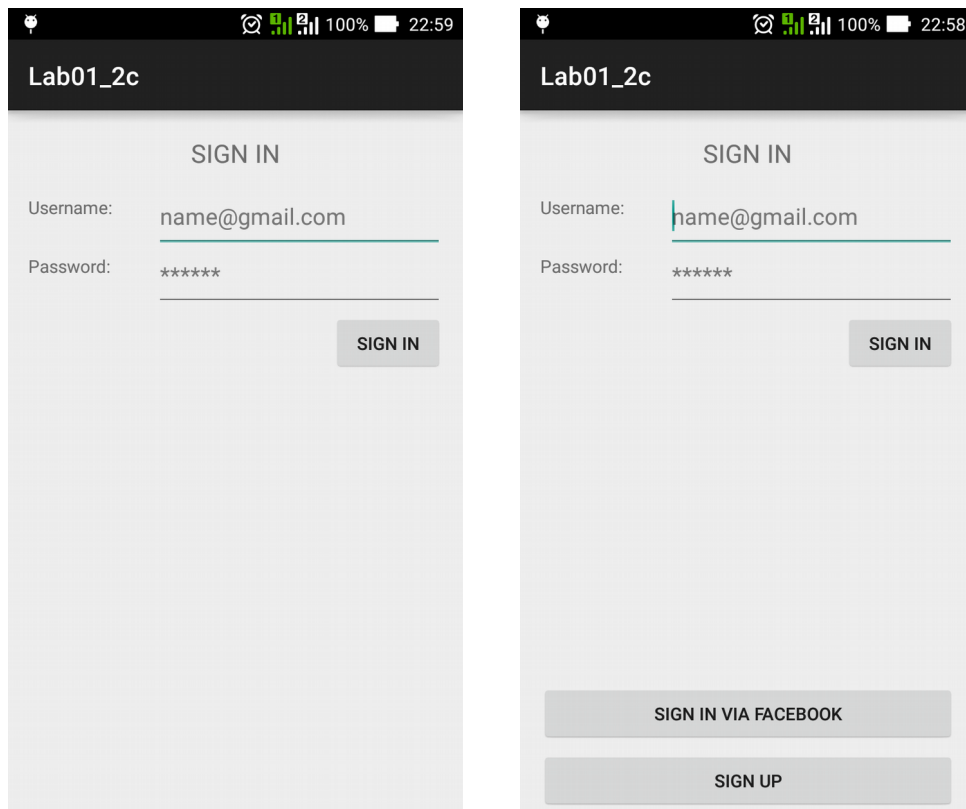
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/et_password"
        android:password="true"
        android:layout_alignTop="@+id/tv_password"
        android:layout_toRightOf="@+id/tv_password"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="@dimen/margin_base"
        android:text="@string/sign_in"
        android:id="@+id/btn_sign_in"
        android:layout_marginTop="@dimen/margin_base"
        android:layout_below="@+id/et_password"
        android:layout_alignParentRight="true" />

</RelativeLayout>
```

Chạy thử chương trình được giao diện như hình bên trái. Điều chỉnh, bổ sung để ứng dụng có giao diện như hình bên phải:

III.



THAM KHẢO THÊM (Tự xem thêm)

1. @Annotation

Giảm 1/3 lượng code, các link tham khảo:

- <https://github.com/excilys/androidannotations/wiki>
- <https://github.com/excilys/androidannotations/wiki/Configuration>
- <https://github.com/excilys/androidannotations/wiki/FirstActivity>

➔ Cố gắng sử dụng các @annotation sau: @Eactivity; @ViewById; @AfterViews; @Click; @LongClick; @ItemClick; @ItemSelect; @Extra; @FragmentArg; @OnActivityResult

2. Retrofit

- <http://square.github.io/retrofit/>