

Lập trình thiết bị di động **LAYOUT VÀ VIEW (P1)**

ThS. Nguyễn Quang Phúc
phucnq.edu@gmail.com

NỘI DUNG

1. Layout

- *Frame Layout*
- *Linear Layout*
- *Table Layout*
- *Relative Layout*
- *Constraint Layout*

2. View cơ bản

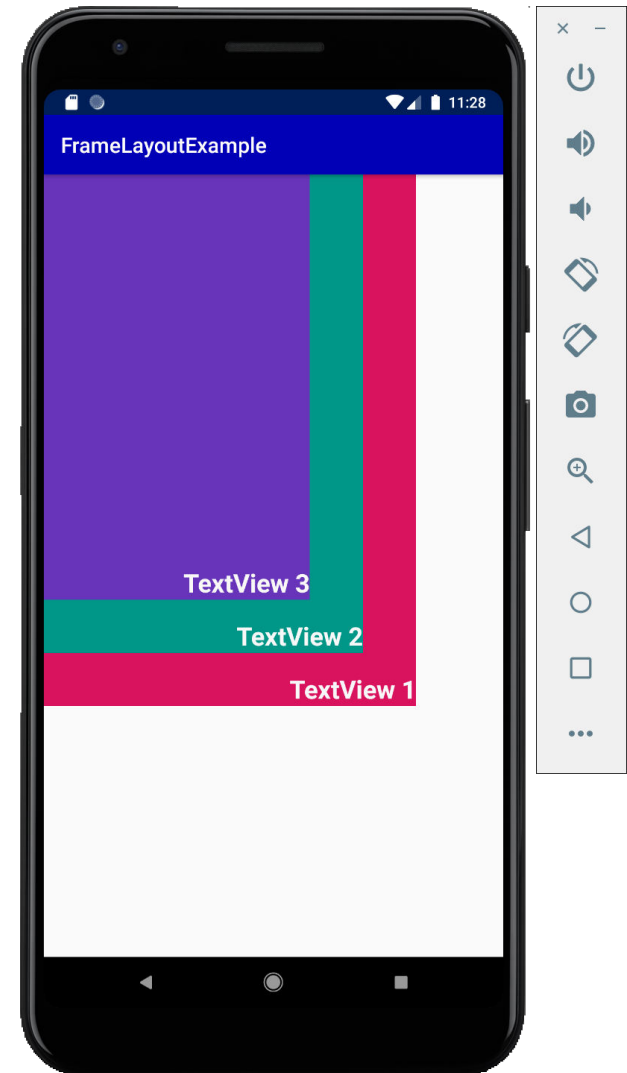
- *TextView*
- *EditText*
- *Button*
- *CheckBox, RadioButton*
- *ImageButton*
- *ImageView*

3. Bài tập rèn luyện

1. Thiết kế giao diện với Layout

➤ Frame Layout

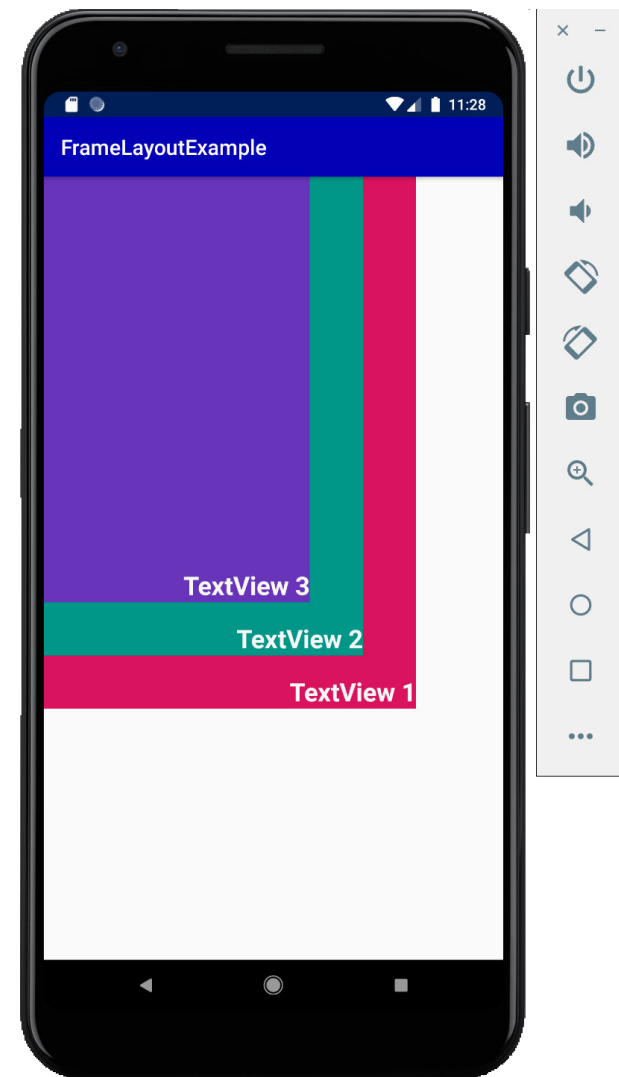
- ✓ *Thường được sử dụng để xây dựng bố cục tổ chức hiển thị một đối tượng duy nhất.*
- ✓ *Các view được thêm vào FrameLayout sẽ mặc định được neo ở vị trí Top-Left, view thêm sau sẽ chồng lên view đã thêm trước đó.*



1. Thiết kế giao diện với Layout

➤ Frame Layout

```
MainActivity.java x activity_main.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   tools:context=".MainActivity">
7   <TextView
8     android:id="@+id/textView"
9     android:layout_width="350dp"
10    android:layout_height="500dp"
11    android:background="@color/colorAccent"
12    android:gravity="bottom|end"
13    android:text="@string/text_view_1"
14    android:textColor="@color/colorText"
15    android:textSize="24sp"
16    android:textStyle="bold" />
17   <TextView
18     android:id="@+id/textView2"
19     android:layout_width="300dp"
20     android:layout_height="450dp"
21     android:background="#009688"
22     android:gravity="bottom|end"
23     android:text="@string/text_view_2"
24     android:textColor="@color/colorText"
25     android:textSize="24sp"
26     android:textStyle="bold" />
27   <TextView
28     android:id="@+id/textView3"
29     android:layout_width="250dp"
30     android:layout_height="400dp"
31     android:background="#673AB7"
32     android:gravity="bottom|end"
33     android:text="@string/text_view_3"
34     android:textColor="@color/colorText"
35     android:textSize="24sp"
36     android:textStyle="bold" />
37 </FrameLayout>
```



1. Thiết kế giao diện với Layout

➤ Linear Layout

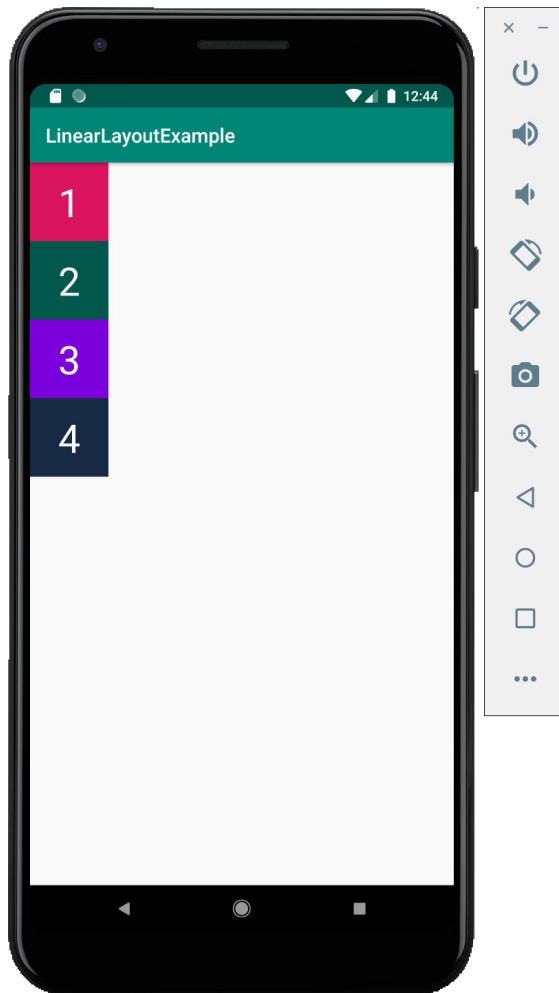
- ✓ Thường được sử dụng để xây dựng bố cục tổ chức hiển thị các đối tượng theo một chiều duy nhất (dọc hoặc ngang).
- ✓ Các view sẽ được bố trí theo dạng khối và không đè lên nhau, có thể sử dụng thuộc tính Gravity thiết lập lại vị trí.

```
<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"
  tools:context=".MainActivity"
  android:orientation="vertical">
```

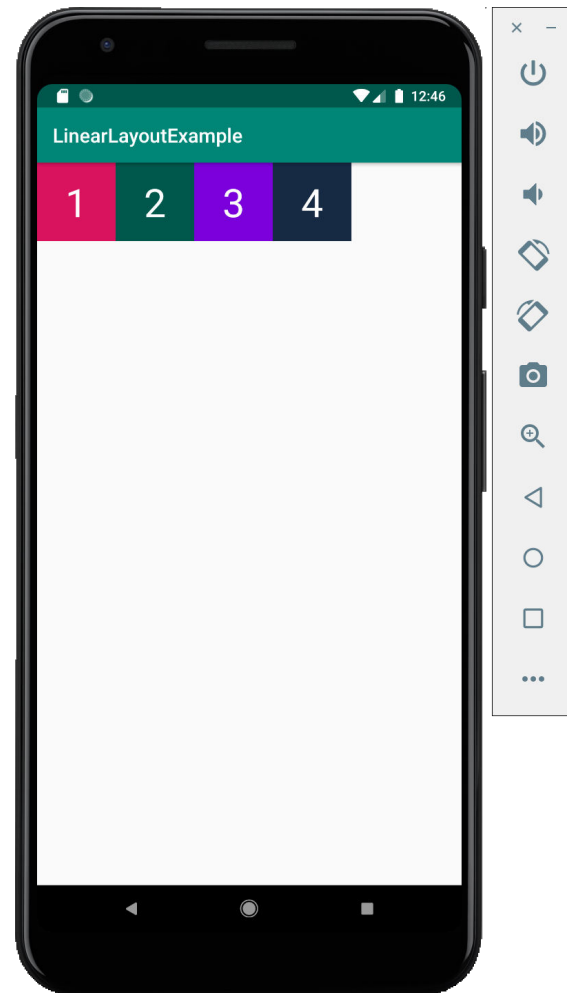
```
</LinearLayout>
```

1. Thiết kế giao diện với Layout

➤ Linear Layout



Vertical

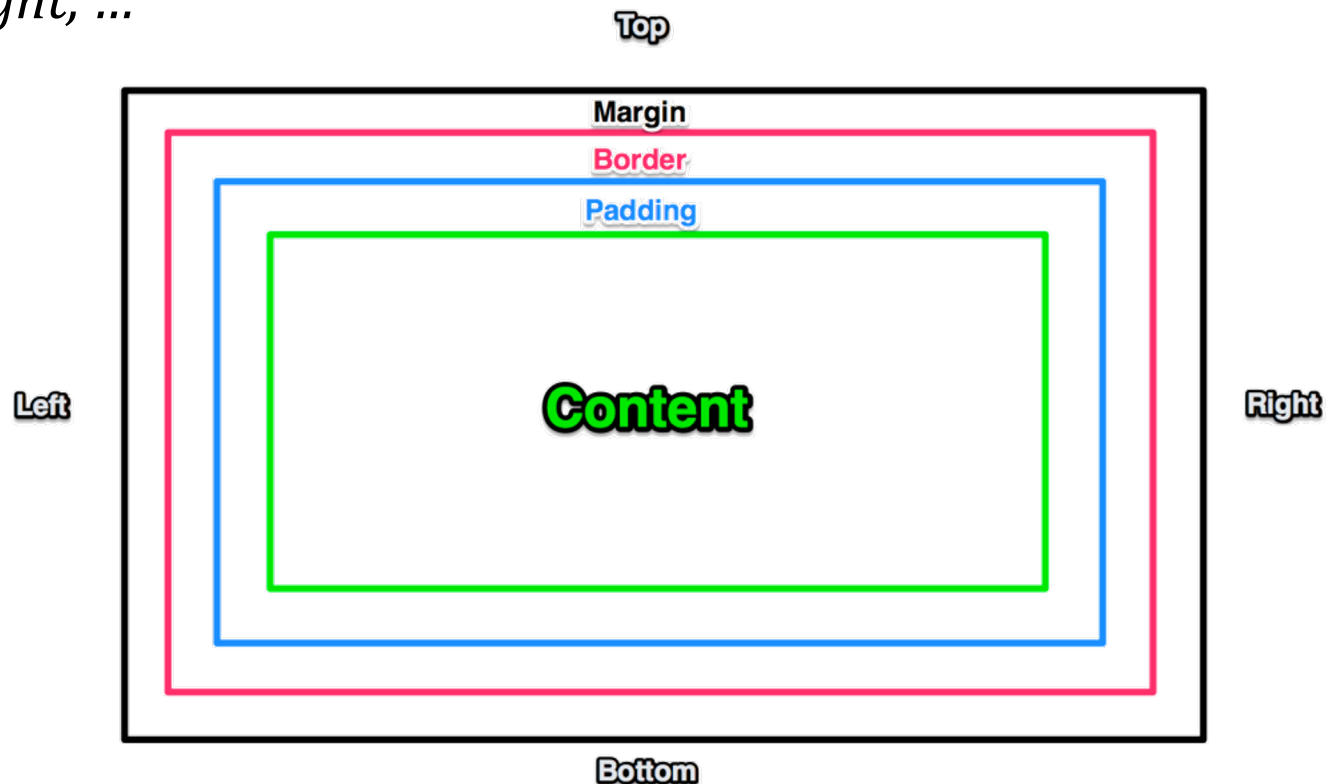


Horizontal

1. Thiết kế giao diện với Layout

➤ Linear Layout

- ✓ Một số thuộc tính hỗ trợ thiết kế: *margin, padding, gravity, weight, ...*



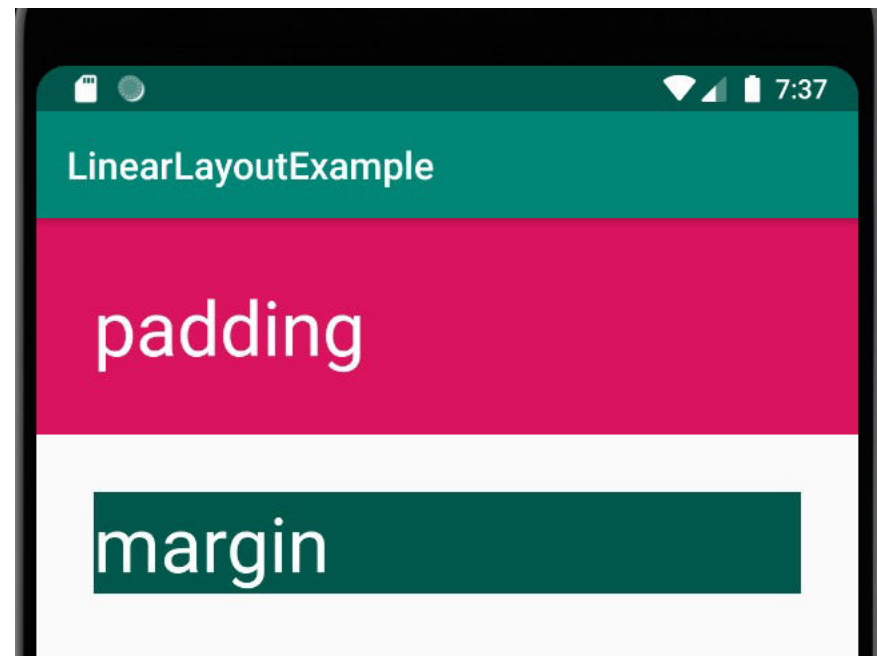
1. Thiết kế giao diện với Layout

➤ Linear Layout

✓ Thuộc tính hỗ trợ thiết kế: *margin, padding*

➔ Padding (internal spacing – khoảng cách giữa nội dung bên trong so với đường viền của control)

➔ Margin (external spacing – khoảng cách giữa control này với control khác)



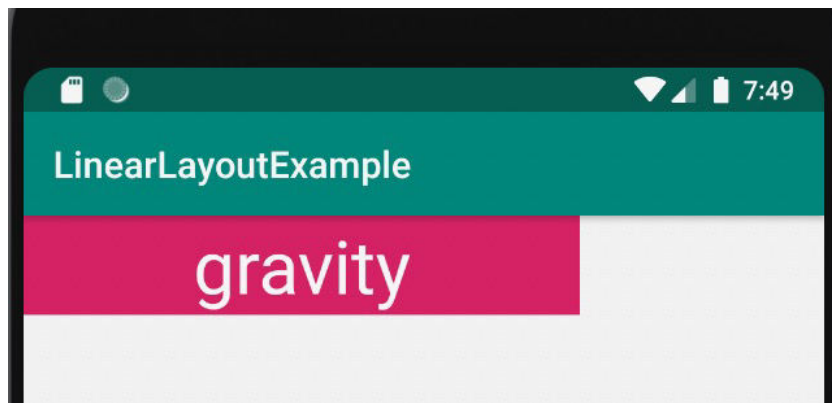
```
android:padding="30dp"
```

```
android:layout_margin="30dp"
```

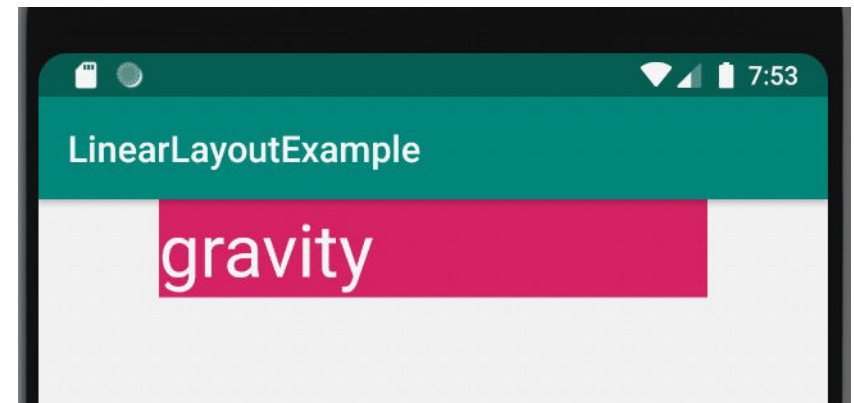

1. Thiết kế giao diện với Layout

➤ Linear Layout

✓ Thuộc tính hỗ trợ thiết kế: *gravity*



```
android:gravity="center"
```

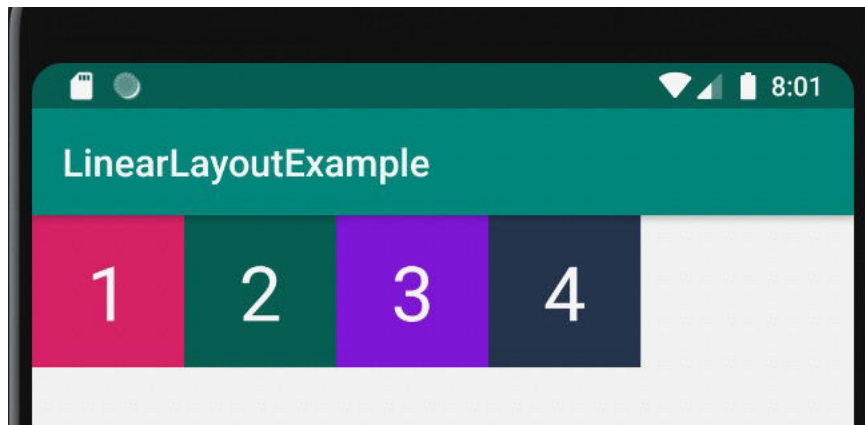


```
android:layout_gravity="center"
```

1. Thiết kế giao diện với Layout

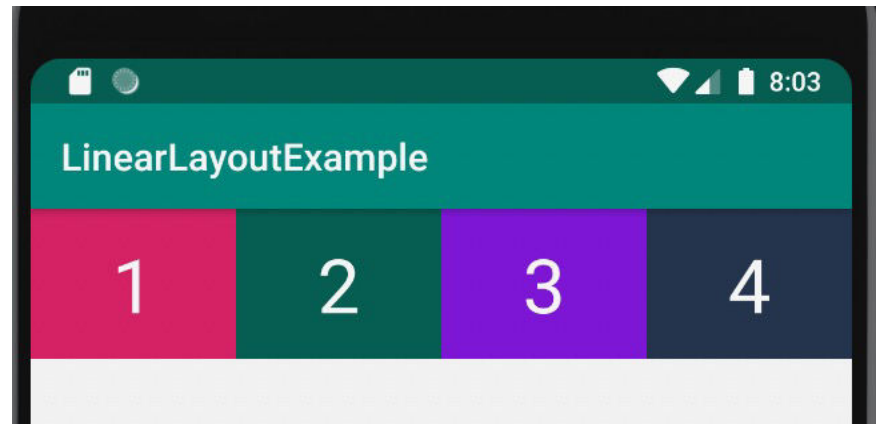
➤ Linear Layout

✓ Thuộc tính hỗ trợ thiết kế: *weight*



`android:layout_weight="1"`

➔ Thuộc tính “`layout_weight`” giúp chia đều tỉ lệ hiển thị của các view.



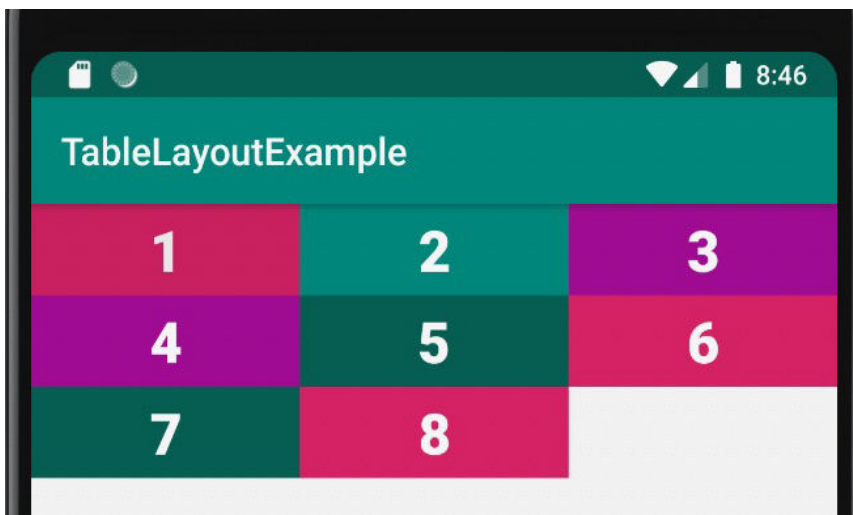
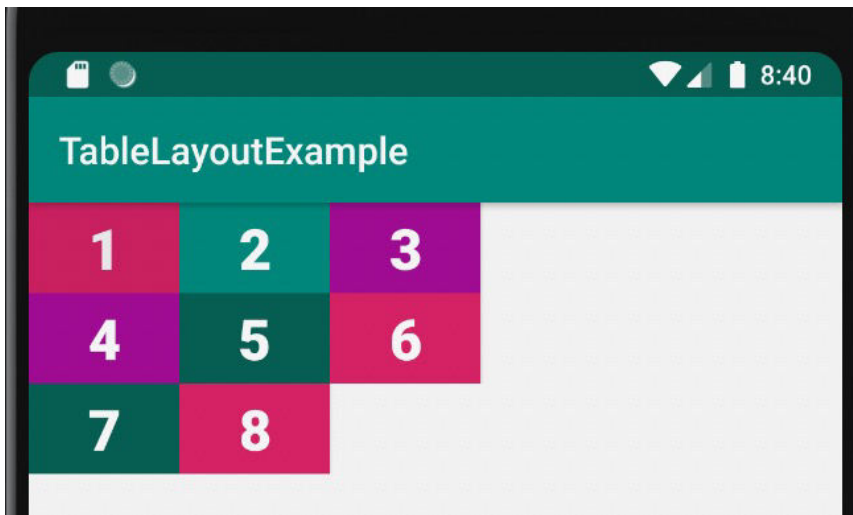
1. Thiết kế giao diện với Layout

➤ Table Layout

- ✓ *TableLayout kế thừa từ LinearLayout, cho phép hiển thị các đối tượng theo dạng bảng. Đối tượng view TableRow đại diện cho mỗi dòng, bên trong TableRow chứa các View con, mỗi View con này nằm ở vị trí một ô bảng (cell).*
- ✓ *Cột / dòng trong TableLayout tính từ vị trí số 0.*
- ✓ *TableLayout sẽ xem dòng nào có số lượng view nhiều nhất để xác định rằng nó có bao nhiêu cột.*

1. Thiết kế giao diện với Layout

➤ Table Layout



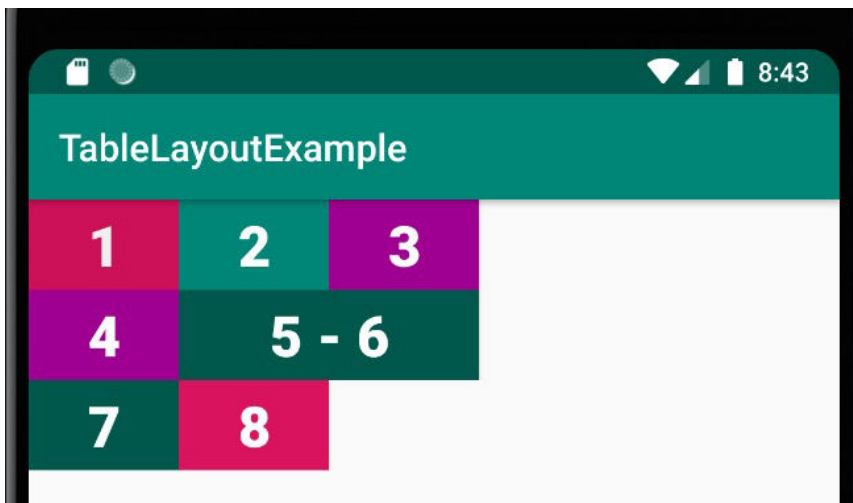
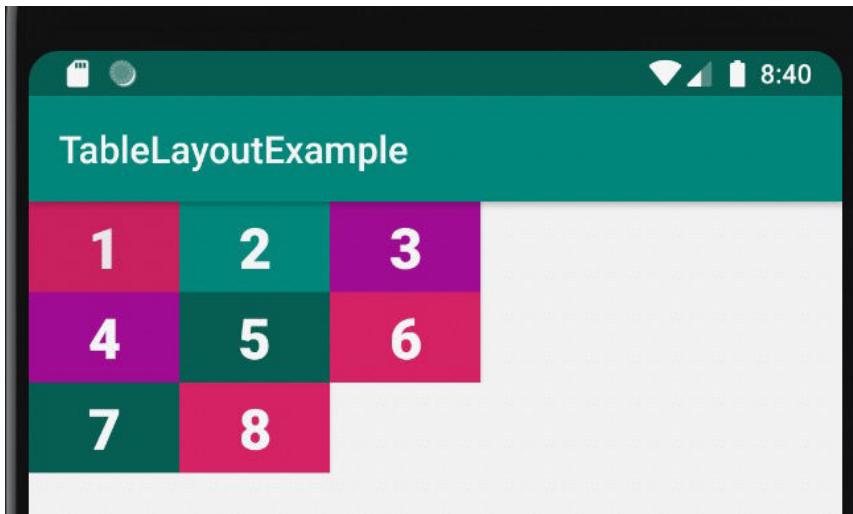
```
<TableLayout
xmlns:android="http://schemas.android.
com/apk/res/android"
xmlns:tools="http://schemas.android.co
m/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:stretchColumns="*">

</TableLayout>
```



1. Thiết kế giao diện với Layout

➤ Table Layout



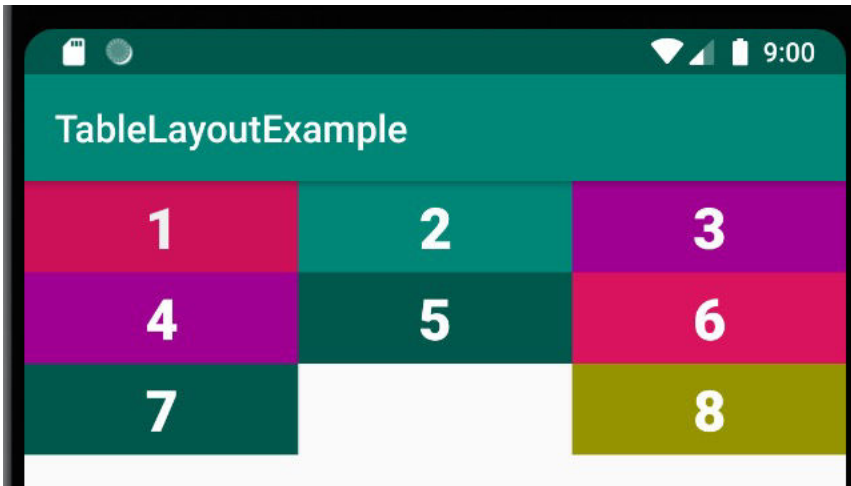
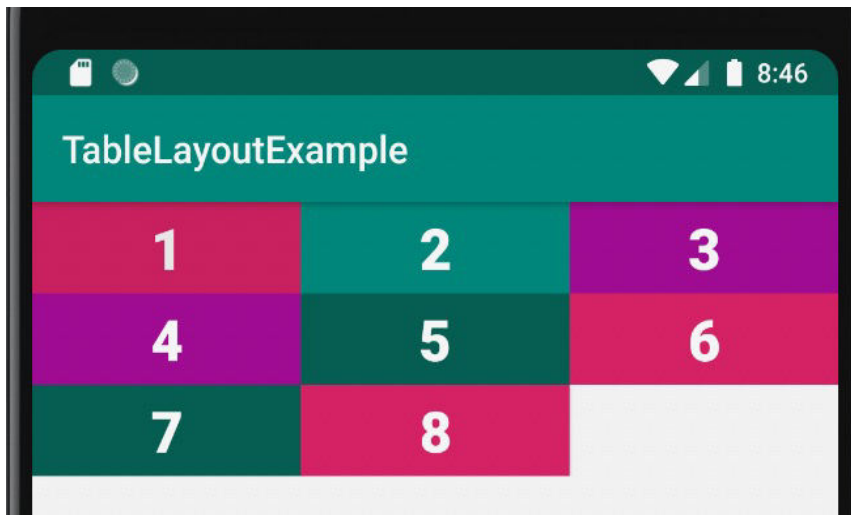
<Button

```
android:id="@+id/button5"  
android:layout_width="80dp"  
android:layout_height="wrap_content"  
android:background="@color/..."  
android:text="5 - 6"  
android:textColor="@color/mauTrang"  
android:textSize="30sp"  
android:textStyle="bold"  
android:layout_span="2"/>
```



1. Thiết kế giao diện với Layout

➤ Table Layout



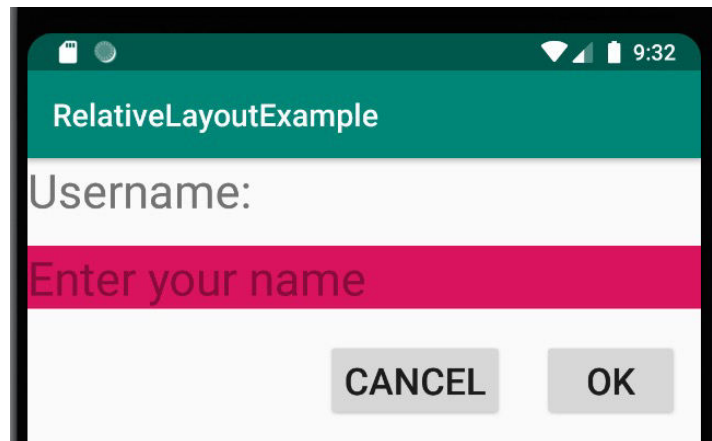
<Button

```
android:id="@+id/button8"  
android:layout_width="80dp"  
android:layout_height="wrap_content"  
android:layout_column="2"  
android:background="#959307"  
android:text="8"  
android:textColor="@color/mauTrang"  
android:textSize="30sp"  
android:textStyle="bold" />
```

1. Thiết kế giao diện với Layout

➤ Relative Layout

- ✓ *RelativeLayout là loại Layout mà trong đó vị trí của mỗi view con sẽ được xác định so với view khác hoặc so với thành phần cha của chúng thông qua ID.*



```
<Button
    android:id="@+id/btnOk"
    .....
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="116dp"
    android:layout_marginEnd="14dp"
    ...../>
```

1. Thiết kế giao diện với Layout

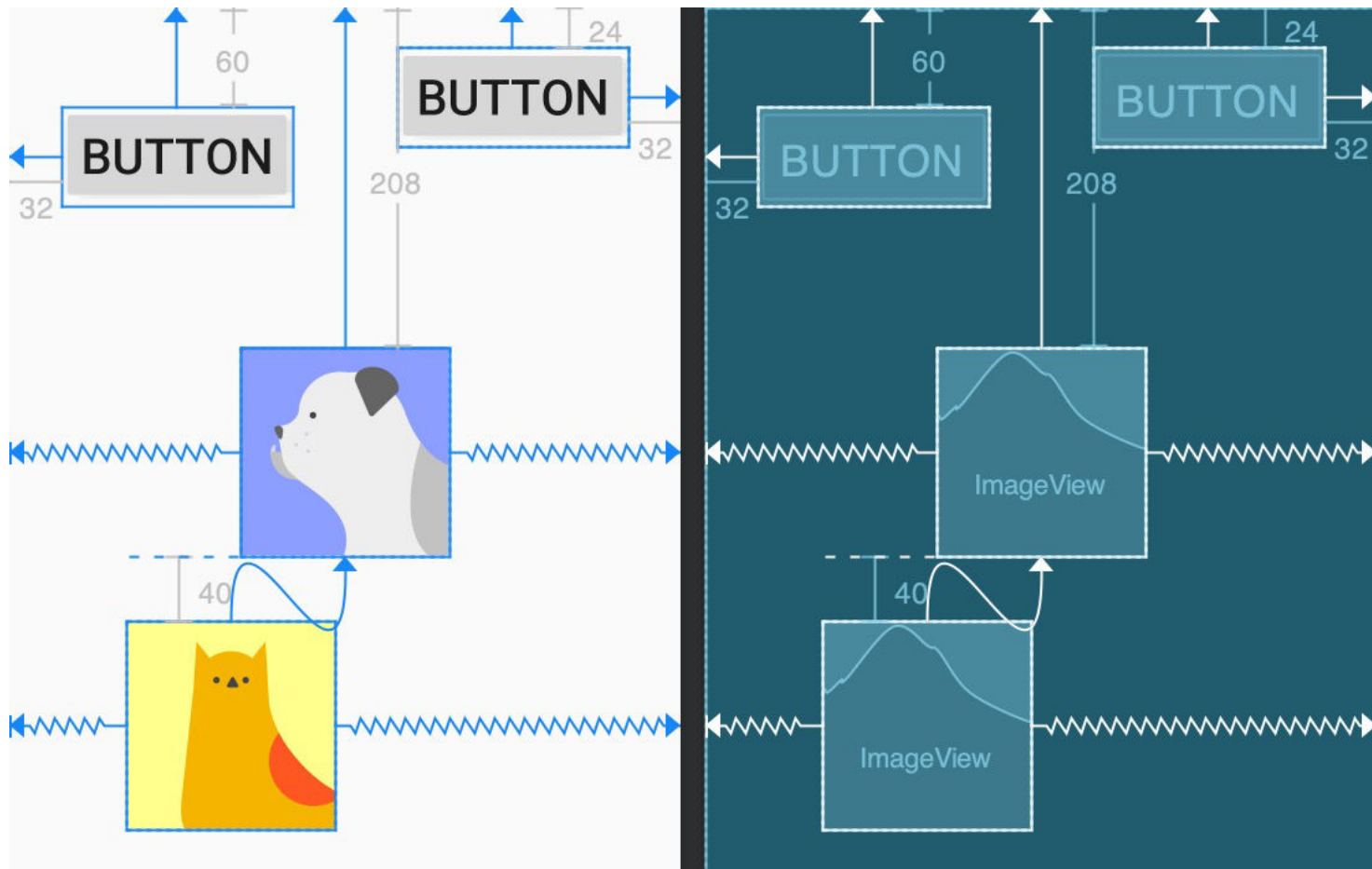
➤ Constraint Layout

- ✓ *ConstraintLayout giúp thiết kế giao diện với các ràng buộc, nguyên tắc liên kết nhất định giữa các thành phần (view).*
- ✓ *Mỗi một view phải có ít nhất một điểm neo theo chiều ngang và một điểm neo theo chiều dọc.*

1. Thiết kế giao diện với Layout

➤ Constraint Layout

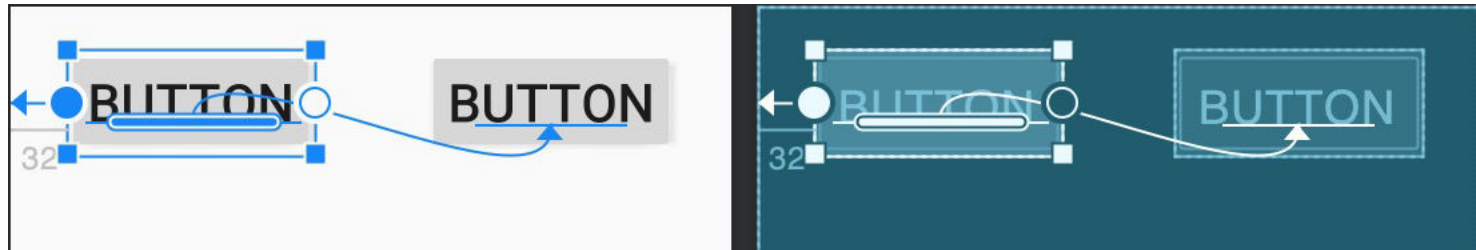
✓ Tạo constraint



1. Thiết kế giao diện với Layout

➤ Constraint Layout

- ✓ Tạo baseline constraint

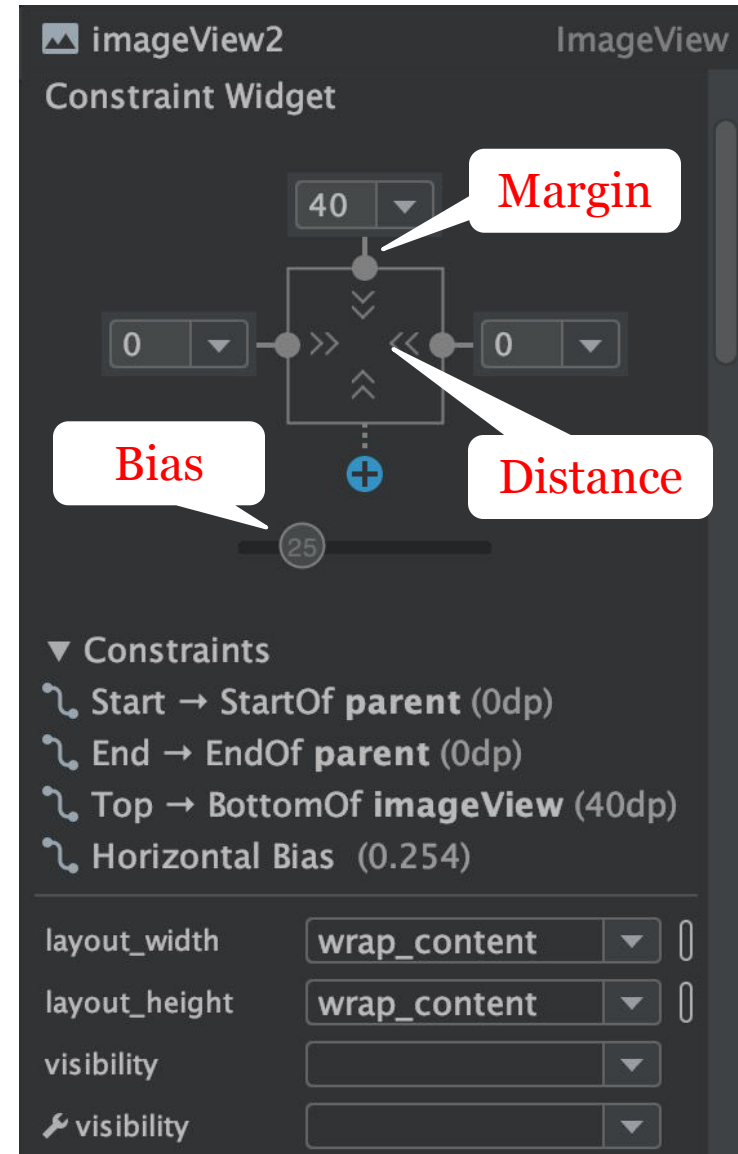
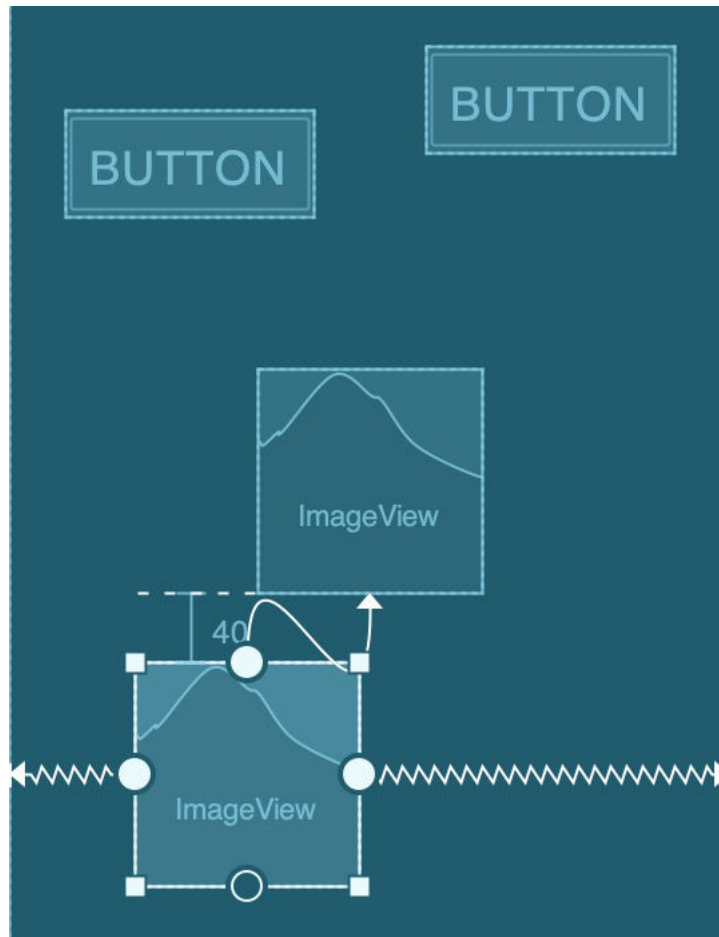


➔ *Baseline constraint giúp canh chỉnh vị trí các view dựa theo text bên trong, phù hợp cho các view như: TextView, EditText, Button, ...*

1. Thiết kế giao diện với Layout

➤ Constraint Layout

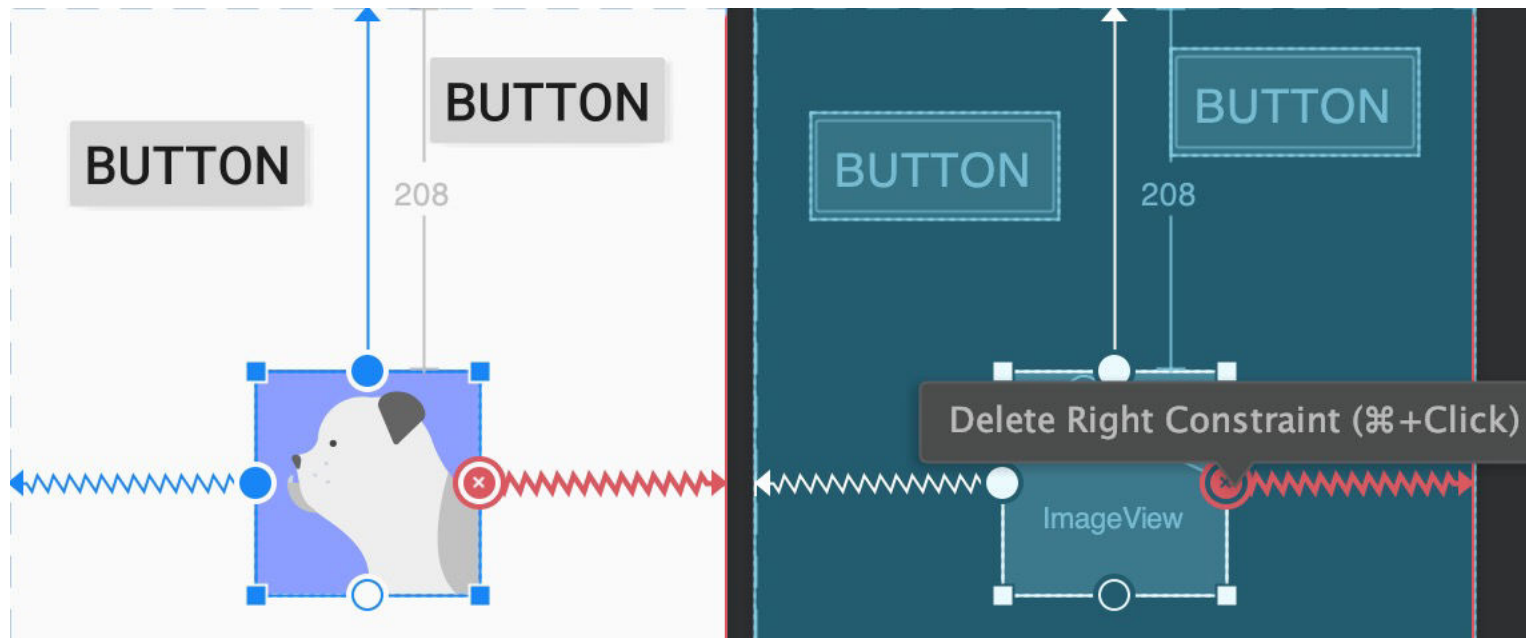
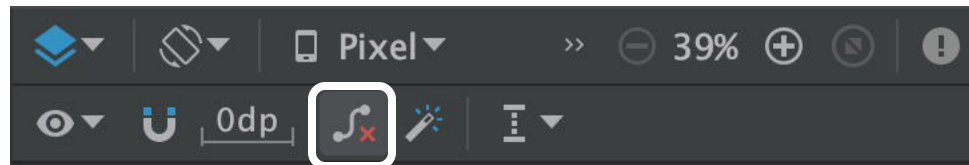
✓ Tạo constraint



1. Thiết kế giao diện với Layout

➤ Constraint Layout

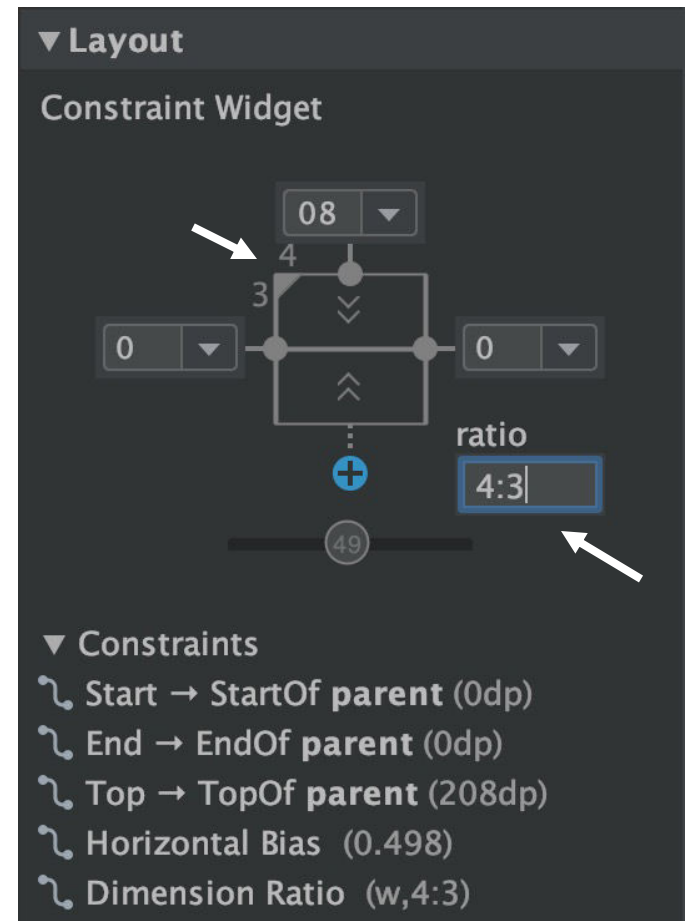
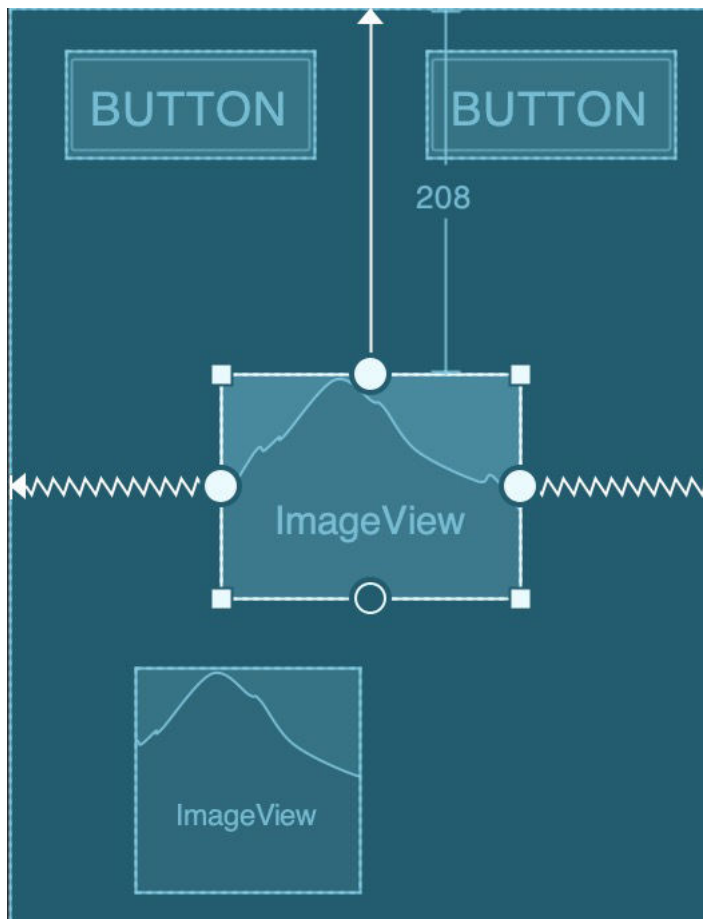
✓ Xóa constraint



1. Thiết kế giao diện với Layout

➤ Constraint Layout

- ✓ Ratio -> điều chỉnh kích thước view theo tỷ lệ

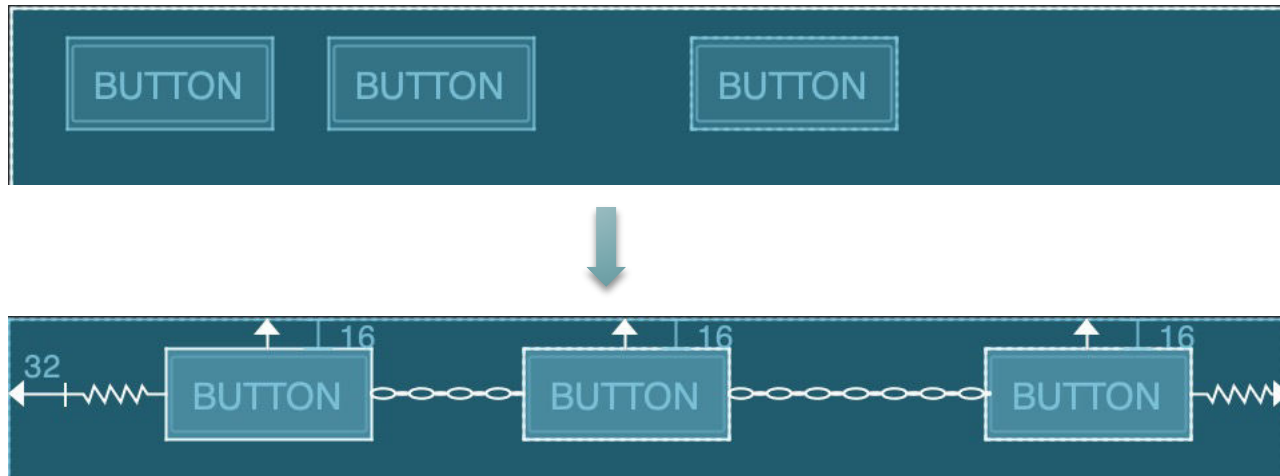


1. Thiết kế giao diện với Layout

➤ Constraint Layout

✓ Chain -> xâu chuỗi (xích) các view

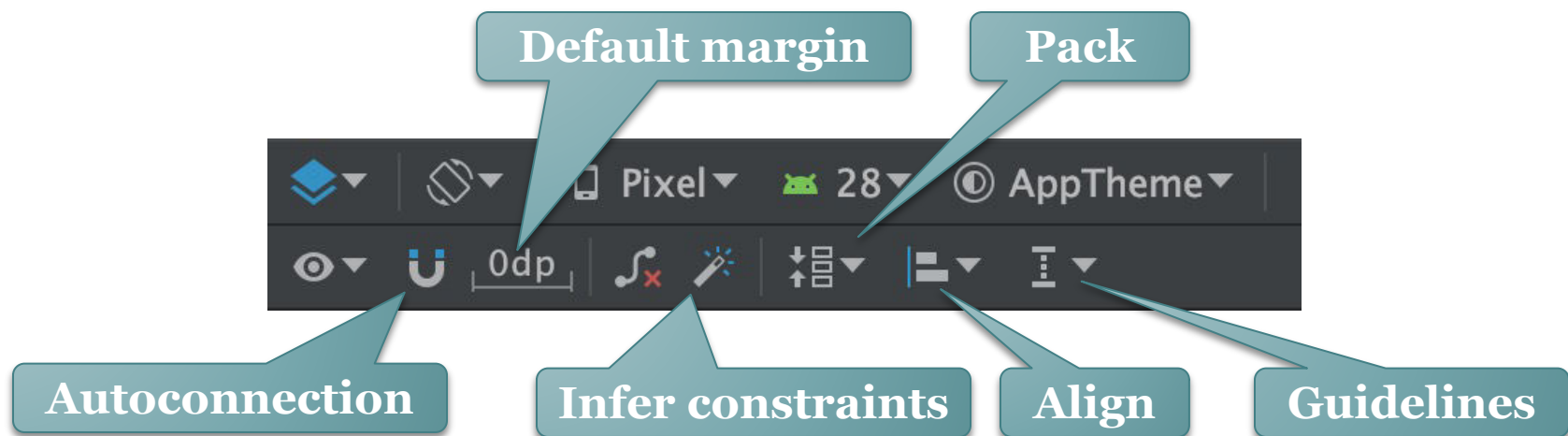
Tạo chain: chọn tất cả các View → click phải chuột → Chains → Create Horizontal/Vertical Chain



1. Thiết kế giao diện với Layout

➤ Constraint Layout

- ✓ Một số tính năng khác



2. View (control) cơ bản

➤ **TextView**

- ✓ *Dùng hiển thị dữ liệu văn bản, không cho phép chỉnh sửa, thường được đặt tên với prefix **txt**.*
- ✓ *Thiết lập nội dung hiển thị:*
 - *Trong Java code (Activity):*

```
textView.setText("TextView");
```

- *Trong XML:*

```
android:text="TextView"
```


2. View (control) cơ bản

➤ TextView

The screenshot displays the Android Studio interface during the setup of a TextView widget. On the left, the **Palette** window shows the **Ab TextView** widget selected under the **Text** category. Below it, the **Component Tree** shows the widget being added to the **LinearLayout(horizontal)** container. The central canvas shows a visual representation of the TextView. On the right, the **Attributes** panel is open, showing the configuration for the **Ab textView** widget. The **Declared Attributes** section shows **layout_width** and **layout_height** set to **wrap_content**. The **Common Attributes** section shows **text** set to **TextView**, **textSize** set to **30sp**, and **textAlignment** set to **center**.

Attributes

Ab textView

id: textView

Declared Attributes

Layout

layout_width: wrap_content

layout_height: wrap_content

layout_weight: 1

visibility:

visibility:

Common Attributes

text: TextView

text:

contentDescripti...:

textAppearance: @android:style/'

fontFamily: sans-serif

typeface:

textSize: 30sp

lineSpacingExtra:

textColor: @android:color/se

textStyle: B I T

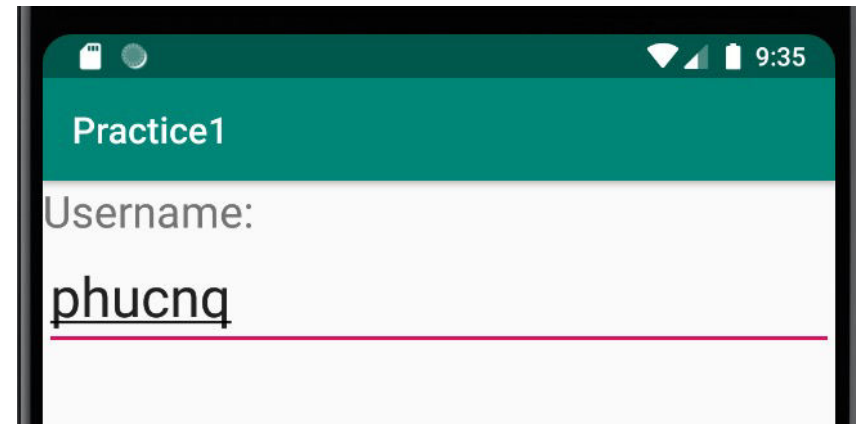
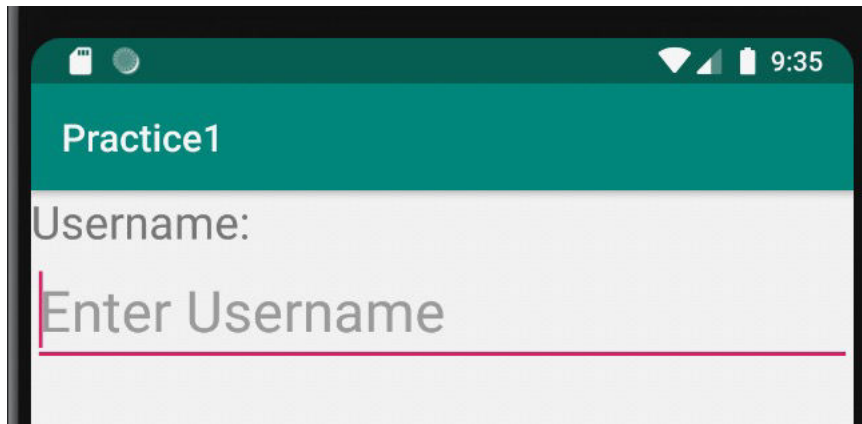
textAlignment: | | | | |

alpha:

2. View (control) cơ bản

➤ EditText

- ✓ Dùng hiển thị và cho phép thay đổi dữ liệu văn bản, thường được đặt tên với prefix *edt*.
- ✓ Có các thuộc tính tương tự như *TextView*.



2. View (control) cơ bản

➤ EditText

The screenshot displays the Android Studio interface with the following components:

- Palette:** The 'Text' category is selected, and 'Ab Plain Text' is highlighted.
- Component Tree:** The hierarchy shows 'LinearLayout(vertical)' containing 'Ab textView- "Username:"' and 'Ab editText(Plain Text)'.
- Design View:** A visual representation of the layout with the text 'Username:' and an input field containing 'Enter Username'.
- Attributes Panel:** The 'editText' widget's properties are listed, including:
 - Declared Attributes:** None.
 - Layout:** None.
 - Common Attributes:**
 - `inputType`: `textPersonName`
 - `hint`: `Enter Username`
 - `style`: `@style/Widget.A`
 - `singleLine`: `True`
 - `selectAllOnFocus`: `True`
 - `text`:
 - `textAppearance`: `@android:style/`
 - `fontFamily`: `sans-serif`
 - `typeface`:
 - `textSize`: `30sp`
 - `lineSpacingExtra`:
 - `textColor`: `@android:color/te`
 - `textStyle`: `B I T`
 - `textAlignment`: `Left`
 - `alpha`:

2. View (control) cơ bản

➤ Button

- ✓ *Cho phép hiển thị dữ liệu văn bản, hình ảnh; thường được đặt tên với prefix **btn**.*
- ✓ *Nhận và phản hồi tương tác “nhấn” từ người dùng.*
- ✓ *Lắng nghe sự kiện:*
 - *Trong Java code:*

```
btnSubmit = findViewById(R.id.btnSubmit);  
btnSubmit.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        //Xử lý ...  
    }  
});
```

- *Trong XML:*

```
android:onClick="MethodName"
```

2. View (control) cơ bản

➤ Button

The screenshot displays the Android Studio IDE with a design view of a login form. The form consists of a text input field labeled "Username:" and a "SUBMIT" button. The "Button" widget is selected in the Palette, and its attributes are shown in the Attributes panel.

Palette:

- Common: Button
- Text: ImageButton
- Buttons: ChipGroup, Chip
- Widgets: CheckBox
- Layouts: RadioGroup, RadioButton
- Containers: ToggleButton
- Google: Switch
- Legacy: FloatingActionButton

Component Tree:

- LinearLayout(vertical)
 - Ab textView- "Username:"
 - Ab editText(Plain Text)
 - btnSubmit- "Submit"

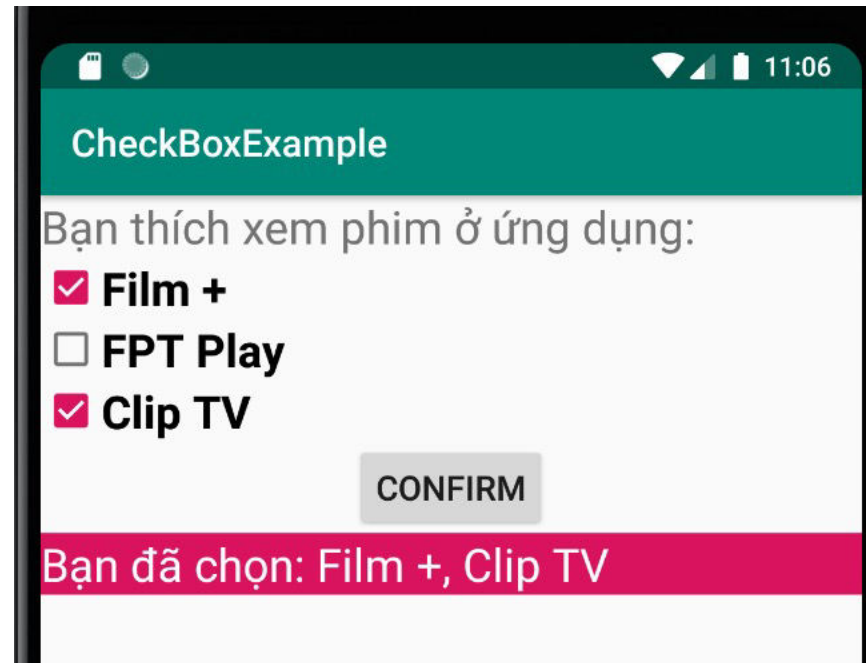
Attributes:

- btnSubmit (Button)
- id: btnSubmit
- Declared Attributes: + -
- Layout: + -
- Common Attributes:
 - style: @android:style/
 - onClick:
 - background: @android:drawable/
 - text: Submit
 - textAppearance: @android:style/
 - fontFamily: sans-serif-med
 - typeface:
 - textSize: 18sp
 - lineSpacingExtra:
 - textColor: @android:color/te
 - textStyle: B I T
 - textAlignment: | | | | |
 - alpha:
- All Attributes: accessibilityLiveR

2. View (control) cơ bản

➤ CheckBox

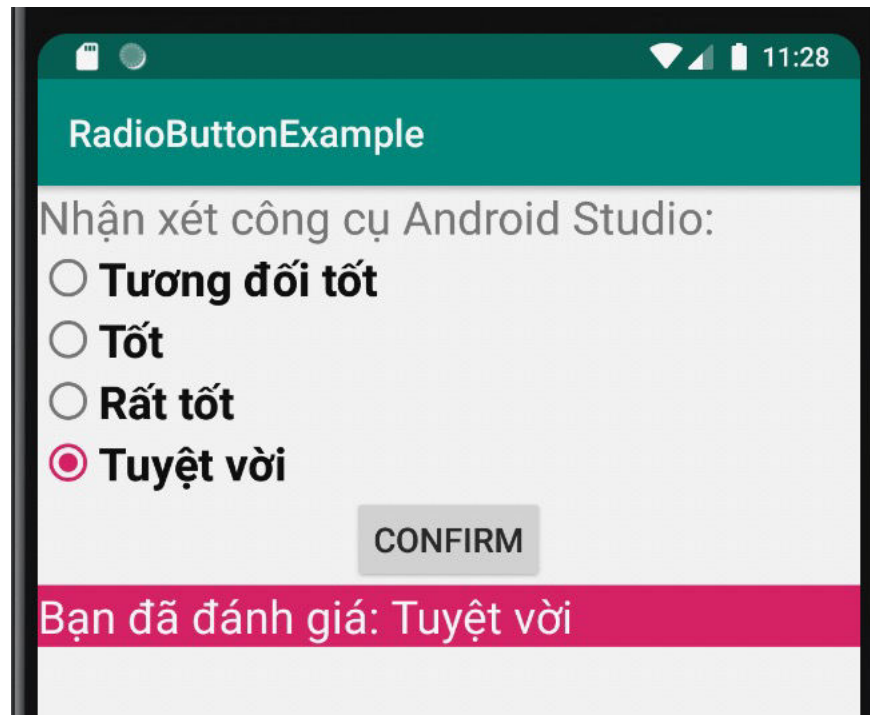
- ✓ *Cho phép chọn nhiều tùy chọn, thường được đặt tên với prefix **chk**.*



2. View (control) cơ bản

➤ RadioButton

- ✓ *Cho phép chọn duy nhất một tùy chọn, thường được đặt tên với prefix **rad**.*



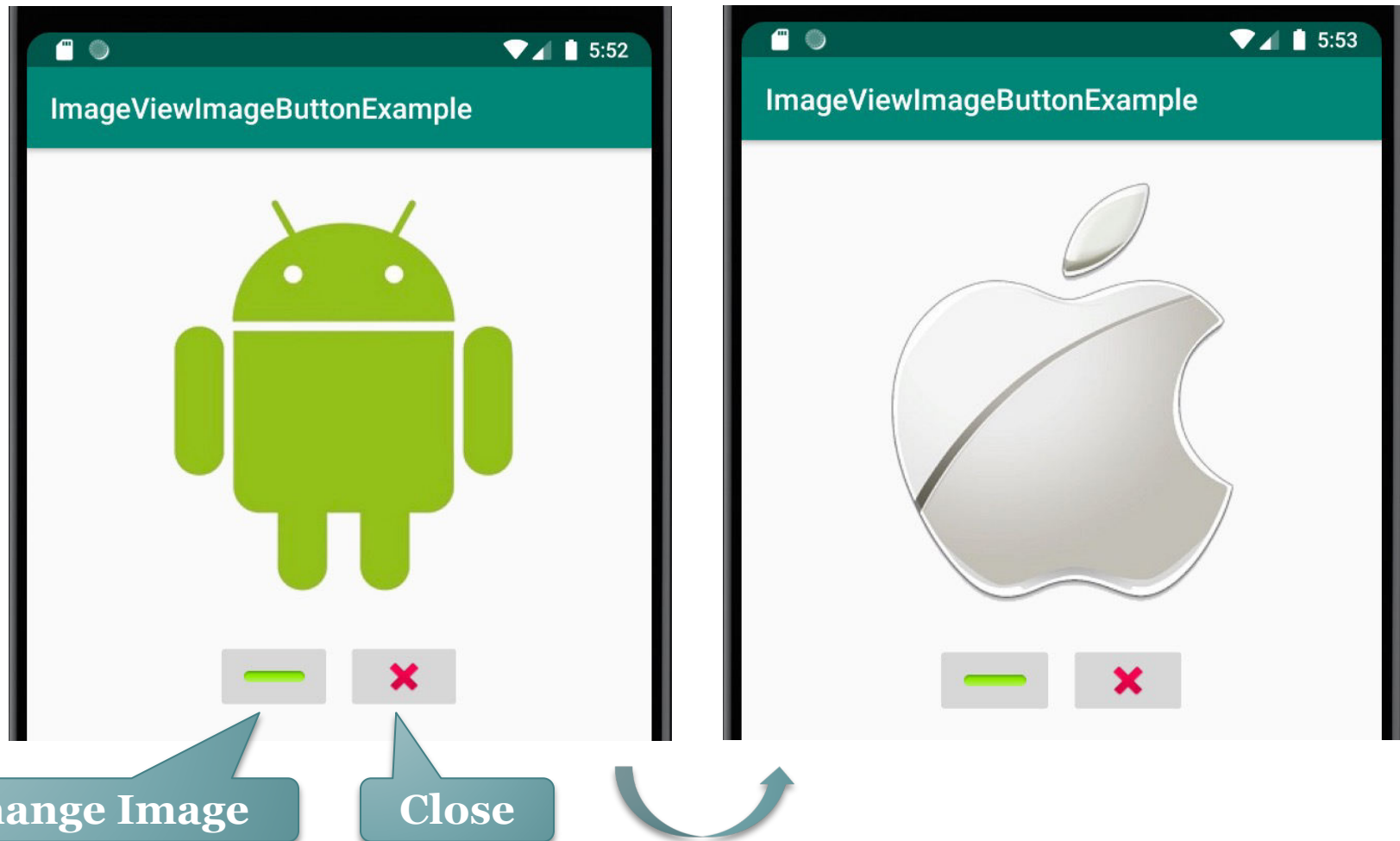
2. View (control) cơ bản

➤ **ImageButton, ImageView**

- ✓ *ImageButton* là một button cho phép chứa hình ảnh, thường được đặt tên với prefix *btn*, *imgBtn*.
- ✓ *ImageView* dùng để hiển thị hình ảnh, cho phép tương tác; thường được đặt tên với prefix *imv*.

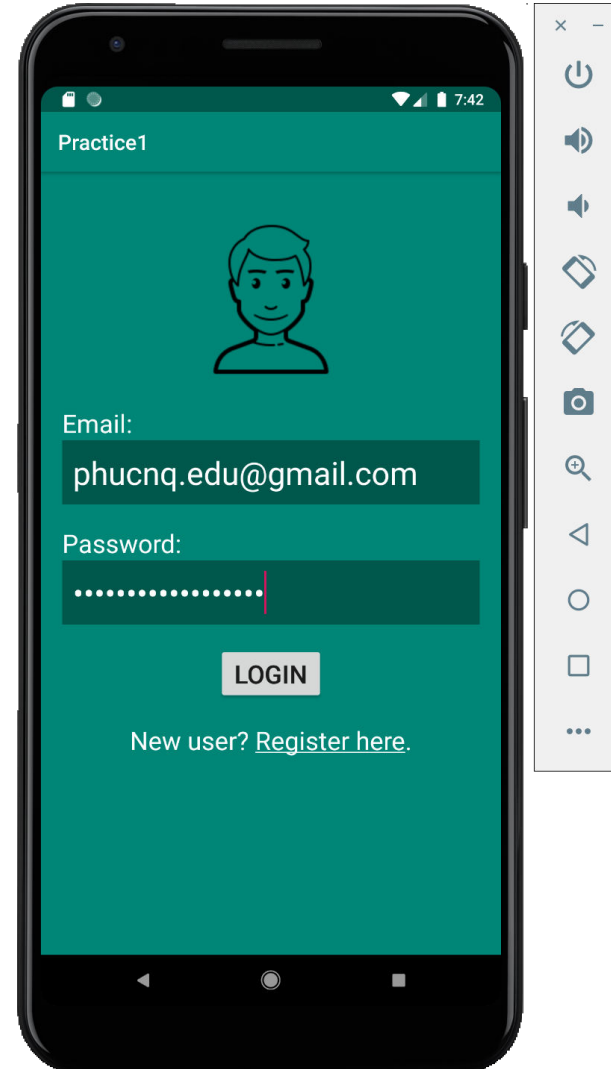
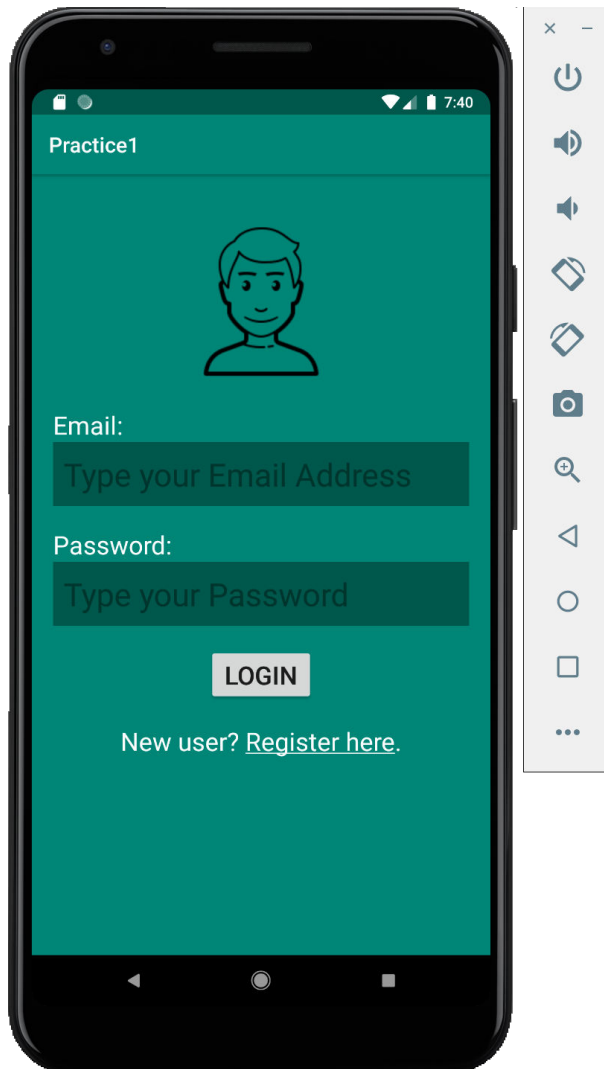
2. View (control) cơ bản

➤ ImageButton, ImageView



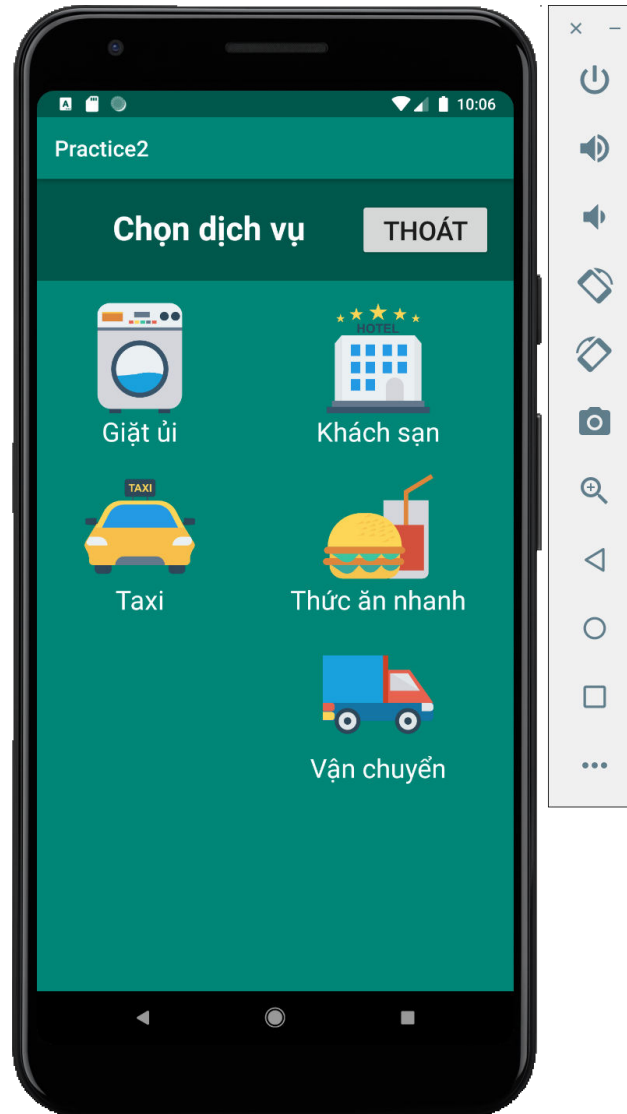
3. Bài tập rèn luyện

➤ Bài tập 1



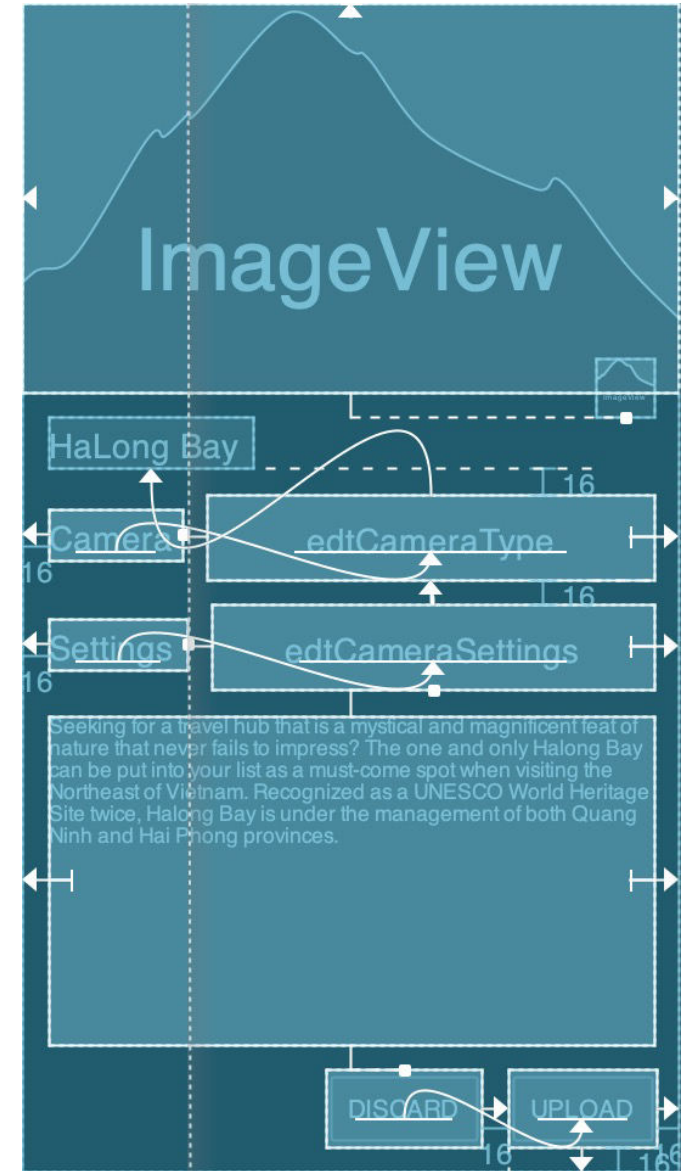
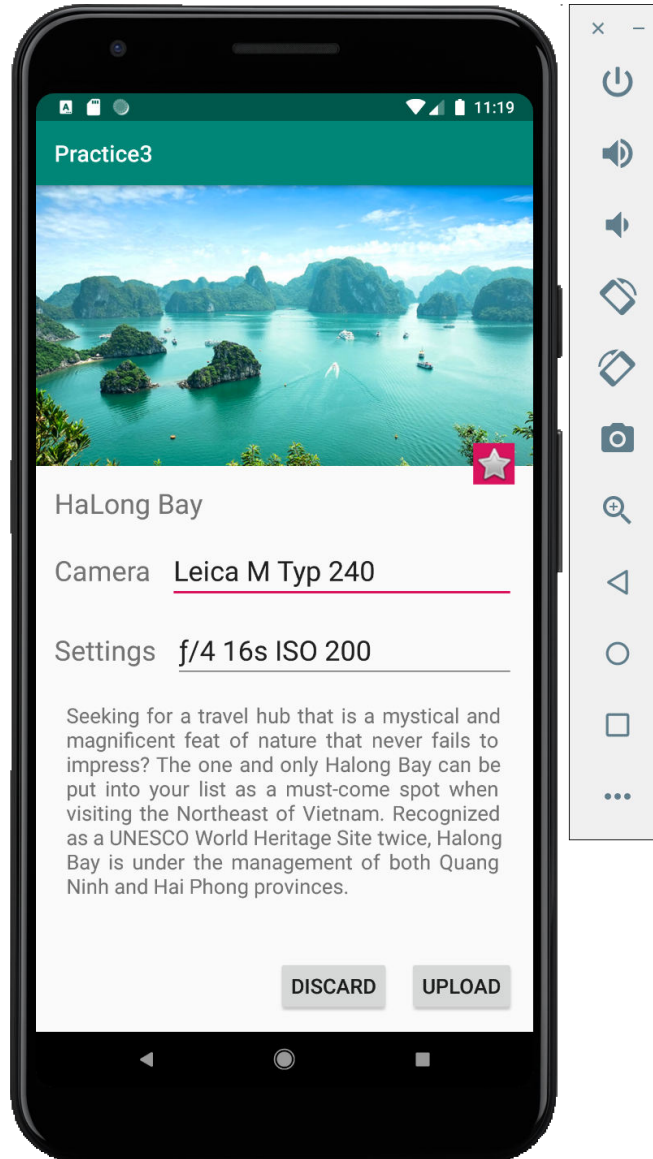
3. Bài tập rèn luyện

➤ Bài tập 2



3. Bài tập rèn luyện

➤ Bài tập 3



Q & A