Fpt University

FPT POLYTECHNIC

LẬP TRÌNH ANDROID CƠ BẢN

BÀI 3: LAYOUT

www.poly.edu.vn

Mục tiêu

- Kết thúc bài học này bạn có khả năng
  - Sử dụng Layout của ứng dụng Android
  - Sử dụng các widget cơ bản

NỘI DUNG

- Phần I: Layout của ứng dụng Android
  - RelativeLayout
  - LinearLayout
- Phần II: Các widget cơ bản
  - TextView
  - EditText
  - Button
  - Image
  - ListView

Fpt University

FPT POLYTECHNIC

BÀI 3: LAYOUT

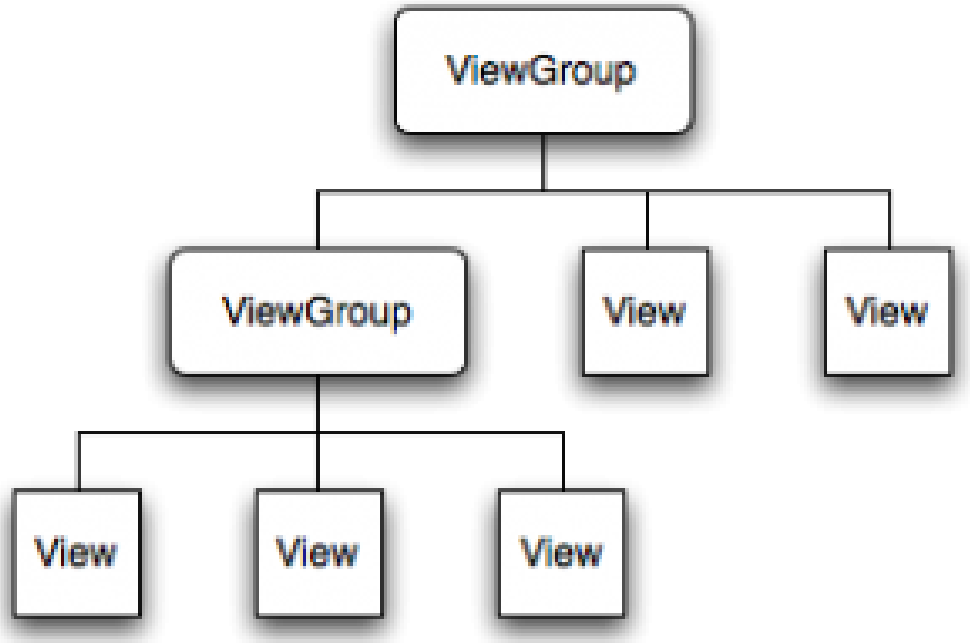
PHẦN I: LAYOUT

www.poly.edu.vn

Thiết kế giao diện người dùng

- Càng đơn giản càng tốt
- Dành nhiều thời gian tìm hiểu nhu cầu của khách hàng về giao diện
- Sử dụng điều khiển giao diện chuẩn

Cây phân cấp View (View Hierarchy)

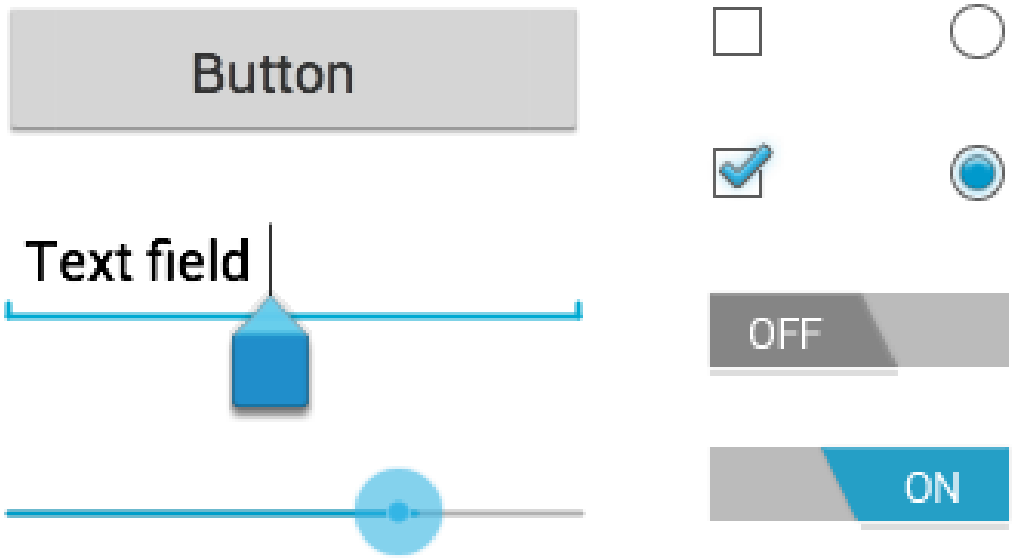


Cây phân cấp View (View Hierarchy)

- View: đơn vị cơ bản của giao diện người dùng
  - Widgets: android.widget.*
  - Là lá của cây phân cấp View
- ViewGroup: định nghĩa layout
  - Nằm trong android.widget.*
  - Định nghĩa nơi chứa các Views (hoặc View Group) con

Ví dụ Widget (view)

- Button
- EditText
- CheckBox và RadioButton
- TextView, ImageView,…



Ví dụ về Layout (ViewGroup)

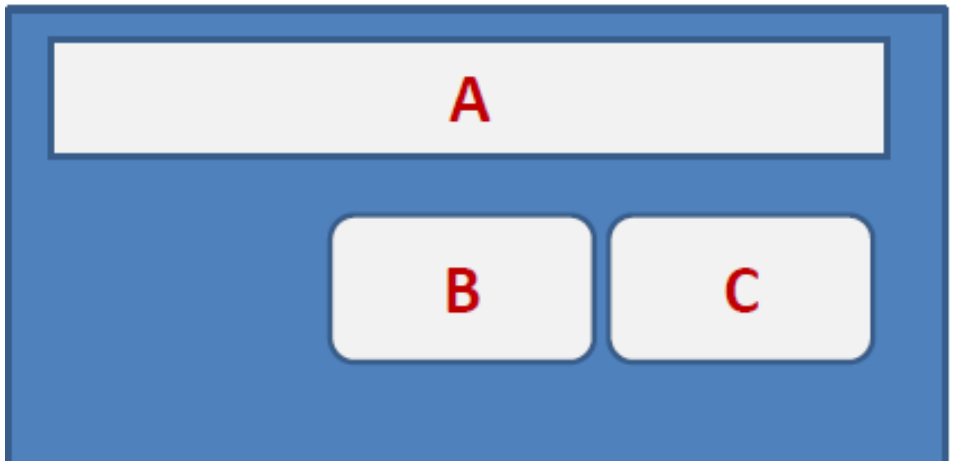
- LinearLayout
- RelativeLayout
- FrameLayout
- TableLayout
- Absolute Layout
- ScrollView Layout
- ....

RELATIVELAYOUT

- RelativeLayout cho phép sắp xếp các control theo vị trí tương đối giữa các control khác trên giao diện (kể cả control chứa nó).
- Thường ta sẽ dựa vào Id của các control khác để sắp xếp theo vị trí tương đối.
- Do đó khi làm RelativeLayout phải chú ý là đặt Id control cho chuẩn xác, nếu sau khi Layout xong mà lại đổi Id của các control thì giao diện sẽ bị xáo trộn (do đó nếu đổi ID thì phải đổi luôn các tham chiếu khác sao cho khớp với Id mới đổi).

Ví dụ:

- A đứng trên đầu ,
- C bên dưới A và ở phía bên phải,
- B bên dưới A và bên trái C



Một số thuộc tính sắp xếp widget với layout chứa nó:

- **android:layout_alignParentTop:** chỉ ra rằng widget phải được đặt ở đầu của layout mà nó nằm.
- **android:layout_alignParentBottom** đặt ở dưới cùng
- **android:layout_alignParentLeft** đặt ở bên trái
- **android:layout_alignParentRight** : đặt ở bên phải
- **android:layout_centerInParent** : đặt ở trung tâm
- **android:layout_centerHorizontal:** đặt ở trung tâm theo chiều ngang
- **android:layout_centerVertical:** đặt ở trung tâm theo chiều dọc

Một số thuộc tính sắp xếp widget với các widget khác:

- **android:layout_above** chỉ ra rằng widget phải được đặt ở trên của widget tham chiếu.
- **android:layout_below** chỉ ra rằng widget phải được đặt ở dưới của widget tham chiếu.
- **android:layout_toLeftOf** chỉ ra rằng widget phải được đặt ở bên trái của widget tham chiếu.
- **android:layout_toRightOf** chỉ ra rằng widget phải được đặt ở bên phải của widget tham chiếu.

RELATIVELAYOUT

- **android:layout_alignTop:** làm cho top của widget này căn bằng với top của widget tham chiếu
- **android:layout_alignBottom** làm cho cạnh dưới của widget này căn bằng với cạnh dưới của widget tham chiếu
- **android:layout_alignLeft** làm cho cạnh trái của widget này căn bằng với cạnh trái của widget tham chiếu
- **android:layout_alignRight** làm cho cạnh phải của widget này căn bằng với cạnh phải của widget tham chiếu

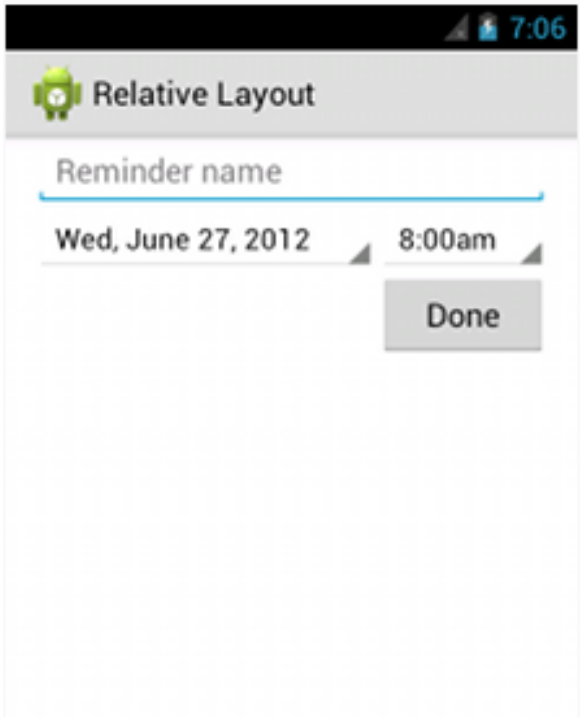
Để sắp xếp các Widget, ta cần phải làm những bước sau:

- Gán Id cho tất cả các phần tử control (android:id)
- Cú pháp: @+id/… để đặt id cho từng control, ví dụ cho thẻ EditText là android:id = “@+id/editUserName”
- Để bố trí control khác liên quan đến control này, ta cũng sẽ sử dụng giá trị (@+id/…). Ví dụ đặt một control dưới hộp EditText ở trên ta có câu lệnh:

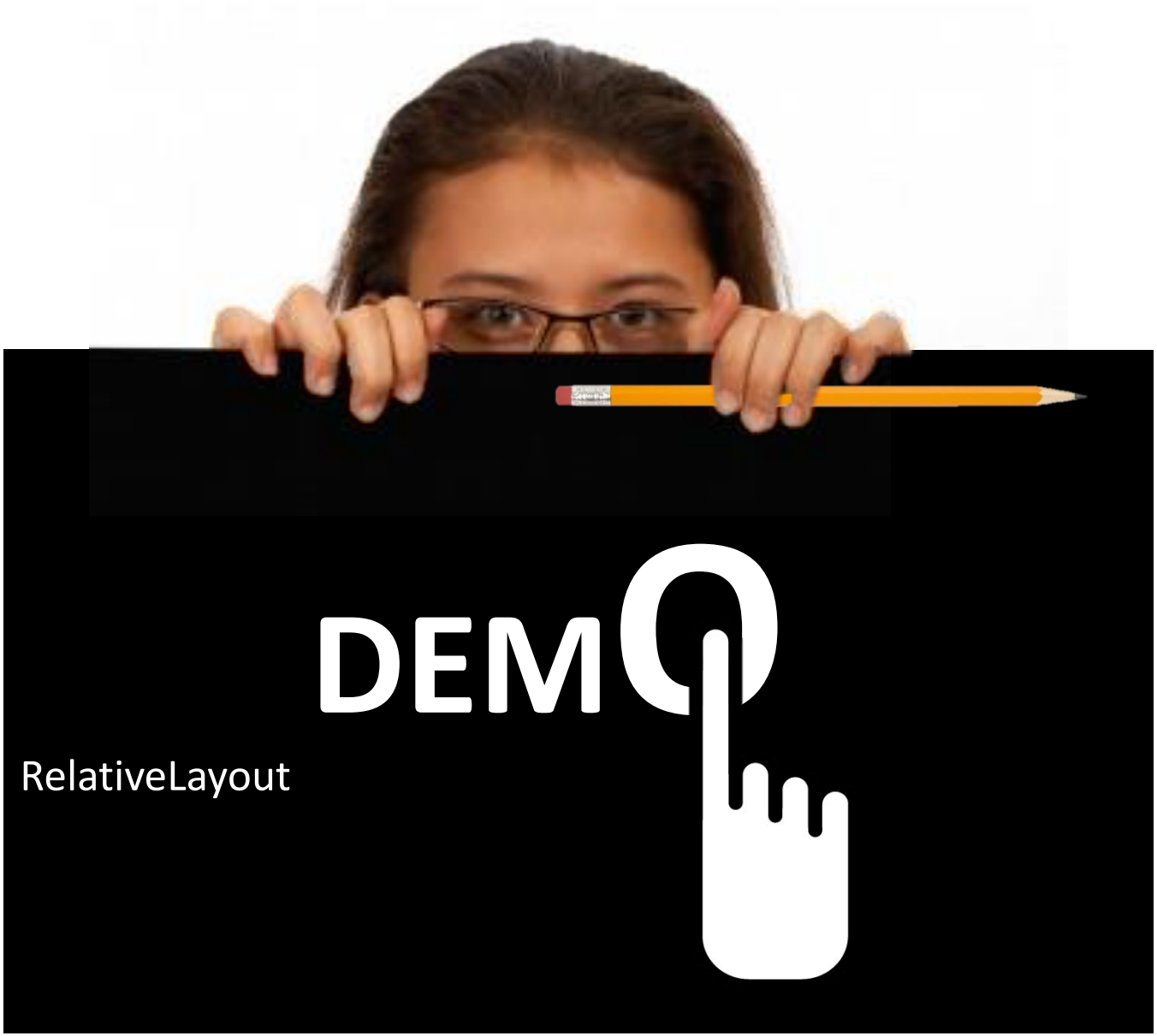
android: layout_below = “@+id/ediUserName”

RELATIVELAYOUT

```
<?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"
   android:paddingLeft="16dp"
   android:paddingRight="16dp" >
   <EditText
      android:id="@+id/name"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:hint="@string/reminder" />
   <Spinner
      android:id="@+id/dates"
      android:layout_width="0dp"
      android:layout_height="wrap_cotent"
      android:layout_below="@id/name"
      android:layout_alignParentLeft="true"
      android:layout_toLeftOf="@+id/times" />
   <Spinner
      android:id="@id/times"
      android:layout_width="96dp"
      android:layout_height="wrap_content"
      android:layout_below="@id/name"
      android:layout_alignParentRight="true" />
   <Button
         android:layout_width="96dp"
         android:layout_height="wrap_content"
         android:layout_below="@id/times"
         android:layout_alignParentRight="true"
         android:text="@string/done" />
   </RelativeLayout>
```

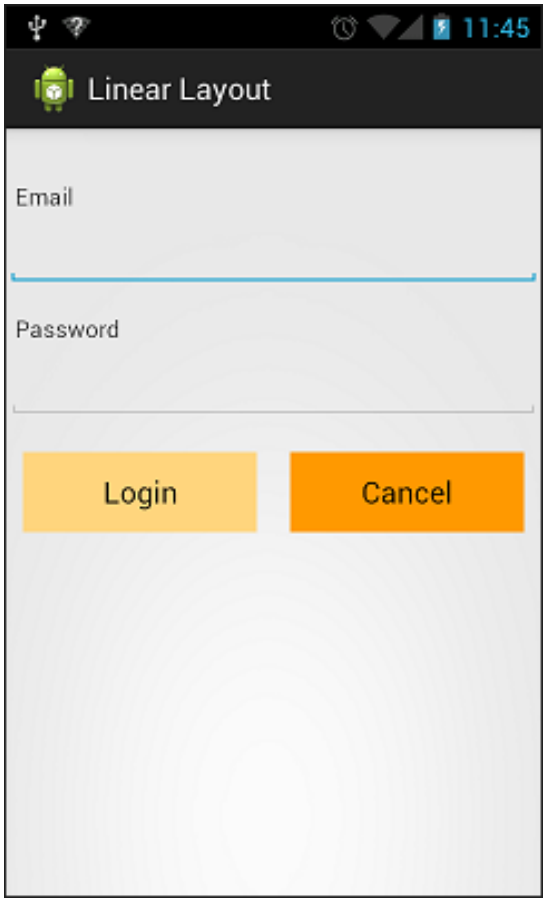


RELATIVELAYOUT



LinearLayout

- Là layout sắp xếp các View con trong nó theo duy nhất một chiều, ngang hoặc dọc theo giá trị của thuộc tính android:orientation
  - Orientation = vertical hoặc horizontal
- Có thể lồng nhiều layout phức tạp
  - Thông thường sử dụng cho form nhỏ



LinearLayout lồng nhau

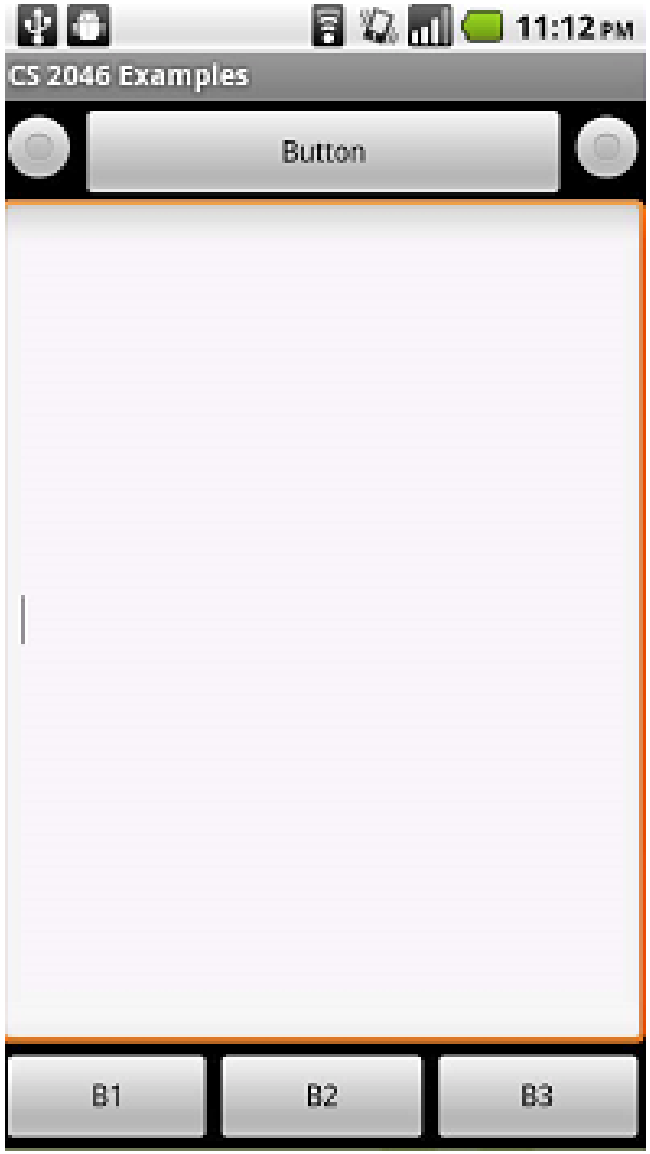
- LinearLayout lồng nhau là một cách để dễ dàng tạo các giao diện chung
- Chú ý: nếu lồng nhau mà số cấp lớn hơn hoặc bằng 5 sẽ làm cho việc tải giao diện chậm hơn

Trọng lượng của Layout (layout weight)

- Trọng lượng cho phép tạo LinearLayout với cỡ cân đối
- Default = 0 – không gian tối thiểu để hiển thị tất cả nội dung

Ví dụ

- Chúng ta sẽ định nghĩa layout cho giao diện sau như thế nào?



Fpt University

FPT POLYTECHNIC

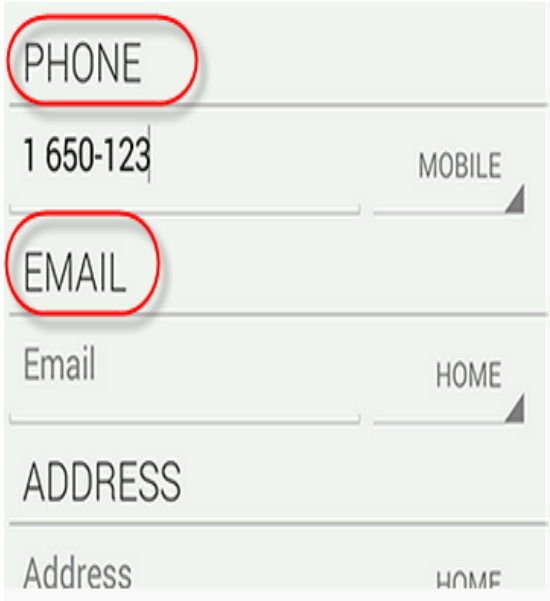
BÀI 3: LAYOUT

PHẦN II: CÁC WIDGET CƠ BẢN

www.poly.edu.vn

1. TextView

- Trong Android, một label được sử dụng là TextView.
- Một TextView dùng để hiển thị thông tin và không cho phép người dùng chỉnh sửa.
- Một số thuộc tính của control:
  - Cần thiết lập id cho control.
  - layout_width, layout_height cũng nên thiết lập cho control (bắt buộc)
  - Để thay đổi màu nền dùng background, thay đổi màu chữ dùng textColor…



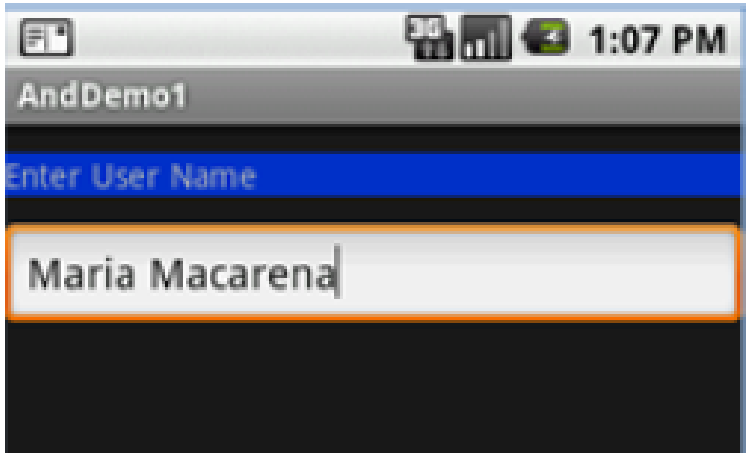
CÁC WIDGET CƠ BẢN

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
android:id="@+id/absLayout"
android:layout_width="fill_parent" android:layout_height="fill_parent"
xmlns:android="http://schemas.android.com/apk/res/android"
>
<TextView
android:id="@+id/myTextView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:background="#ff0000ff"
android:padding="3px"
android:text="Enter User Name"
android:textSize="16sp"
android:textStyle="bold"
android:gravity="center"
android:layout_x="20px"
android:layout_y="22px“ >
</TextView>
</AbsoluteLayout>
```

2. EditText

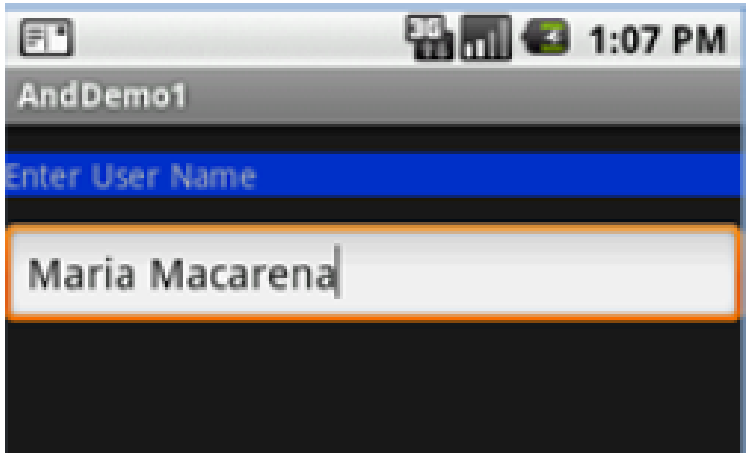
- Thẻ EdtitText (hoặc textBox) là một trường hợp mở rộng của TextView, cho phép cập nhật, chỉnh sửa dữ liệu

```
<EditText
android:id="@+id/txtUserName"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:textSize="18sp"
android:autoText="true"
android:capitalize="words"
android:hint="First Last Name"
>
</EditText>
```



CÁC WIDGET CƠ BẢN

```
<EditText
android:id="@+id/txtUserName"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:textSize="18sp"
android:autoText="true"
android:capitalize="words"
android:hint="First Last Name"
>
</EditText>
```



- Lấy control theo Id:

  EditText txtbox=(EditText) findViewById(R.id.editText1);

- Thiết lập giá trị cho EditText

  txtBox.setText("some text")

- Lấy dữ liệu bên trong EditText:

  String msg=txtBox.getText().toString()

3. Button

- Thẻ Button cho phép người dùng thực hiện một hành động click vào giao diện.
- Là một lớp con của TextView, do đó về cơ bản các thuộc tính cũng tương tự như TextView được liệt kê trang trước.
- Một button có thể chứa text, hoặc icon, hoặc cả text cả icon.

- Với text, sử dụng thẻ button:

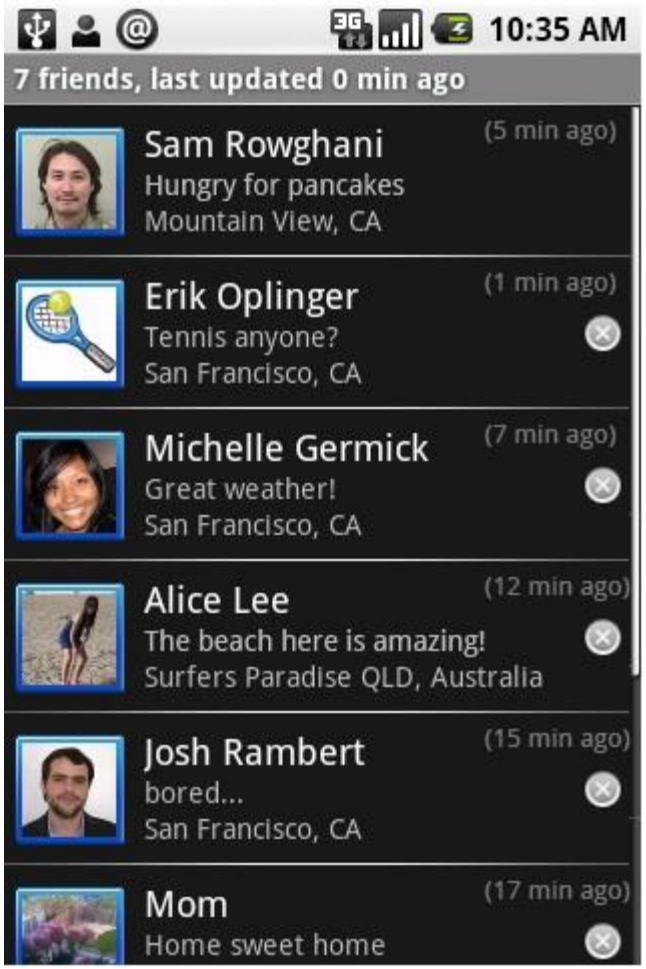
```
<Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_text"
        ... />
```

4. Image

- ImageView và ImageButton thường được dùng để nhúng hình ảnh cho ứng dụng.
- Cả hai thẻ này đều tương tự như TextView và Button.
- Các thẻ này đều sử dụng thuộc tính android:scr hoặc android:background để hiển thị ảnh.
- Ảnh được sử dụng sẽ được tham chiếu tới thư mục /res/*drawable/*
- ImageButton là một lớp con của ImageView và thêm một số chuẩn của Button để thực hiện các sự kiện Click

ListView

- ViewGroup chứa danh sách các View
- Có thể định nghĩa một View để hiển thị khi List rỗng sử dụng setEmptyView
- Mỗi dòng mặc định là TextView, có thể tùy biến
- Thông thường được load dữ liệu động



ListView Adapter

- Adapter – ràng buộc nội dung động vào View trong ListView

  Ví dụ ArrayAdapter đối với mảng
- Đơn giản – ràng buộc giá trị text vào text field trong ListView
- Phức tạp hơn – tùy biến ListView row, đối tượng tùy biến được ràng buộc vào View

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>
(this,android.R.layout.simple_list_item_1, Subjects);
```

Các bước tạo ListView

- Tạo Listview trong Layout

- Tạo Array hoặc ArrayList chứa dữ liệu

```
String[] arr = new String[]{“Apple”,”Samsung”,”LG”}
```

- Tạo Adapter ràng buộc nội dung động vào Array

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>
(this,android.R.layout.simple_list_item_1, arr);
```

- Show dữ liệu trong mảng lên ListView

```
ListView.setAdapter(adapter)
```


DEMO
- Sử dụng Adapter cho Listview

TỔNG KẾT NỘI DUNG BÀI HỌC

- Phần I: Layout của ứng dụng Android
  - RelativeLayout
  - LinearLayout
- Phần II: Các widget cơ bản
  - TextView
  - EditText
  - Button
  - Image
  - ListView


TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
LUẬT KINH TẾ
KỸ THUẬT SOẠN THẢO VÀ TRÌNH BÀY VĂN BẢN
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
Cảm ơn