Lập trình ứng dụng trên đầu cuối di động

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG DI ĐỘNG ANDROID

Nội dung

1. Lập trình quản lý vòng đời ứng dụng di động
2. Lập trình giao diện ứng dụng di động
3. Lập trình quản lý dữ liệu
4. Lập trình kết nối mạng
5. Lập trình chức năng cuộc gọi
6. Lập trình chức năng nhắn tin ngắn (SMS)
7. Lập trình xử lý đa phương tiện
8. Lập trình điều khiển các thiết bị phần cứng tích hợp trên đầu cuối di động

1. Lập trình quản lý vòng đời ứng dụng di động Android

Tiến trình (process)

- Android cung cấp máy ảo Dalvik cho phép tất cả các ứng dụng chạy trong tiến trình riêng của nó. Đồng thời Android có cơ chế quản lý các tiến trình (process) riêng này theo chế độ ưu tiên (priority). Các tiến trình có độ ưu tiên thấp sẽ bị Android giải phóng mà không hề cảnh báo nhằm đảm bảo tài nguyên.

- Thời gian sống của tiến trình ứng dụng không được điều khiển trực tiếp bởi chính nó. Thay vào đó, nó được xác định bởi hệ thống qua sự kết hợp của:
  - Những phần của ứng dụng mà hệ thống biết đang chạy.
  - Những phần đó quan trọng như thế nào đối với người dùng.
  - Bao nhiêu vùng nhớ chiếm lĩnh trong hệ thống.

Các loại tiến trình(process) của Android

- **Active process:** là process của ứng dụng hiện thời đang được người dùng tương tác. Active process bao gồm:
  - Các Activity đang ở trạng thái hoạt động (active).
  - Broadcast Receivers đang thực thi sự kiện onReceive.
  - Các service đang thực thi các sự kiện onStart, onCreate, onDestroy.
  - Các service được chỉ định để chạy ở chế độ hiển thị trên màn hình (foreground).

- **Visible process:** là process của ứng dụng mà Activity đang hiển thị đối với người dùng, nhưng không chạy ở chế độ foreground cũng như không nhận tương tác từ người dùng. Điều này xảy ra khi Activity không hiển thị toàn màn hình hoặc trong suốt với người dùng (Khi sự kiện OnPause của Activity được gọi).

- **Started Service process:** là service đang chạy. Những service này bị Android giải phóng khi active process hoặc visible process cần sử dụng tài nguyên của hệ thống

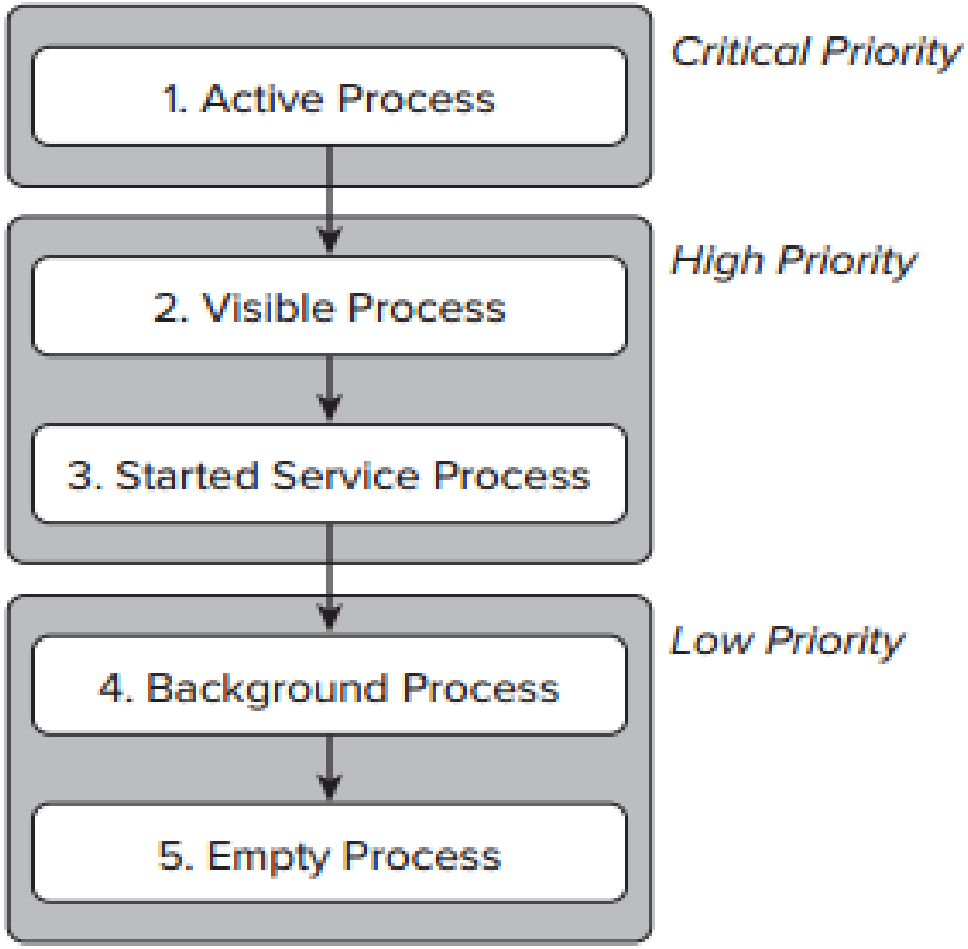
Các loại tiến trình(process) của Android

- **Background process:** là process của ứng dụng mà các activity của nó ko hiển thị với người dùng và không có bất kỳ service nào đang chạy (Khi sự kiện OnStop của Activity được gọi).

- **Empty process:** process không có bất cứ 1 thành phần nào active.
  - Để cải thiện hiệu suất tổng thể hệ thống, Android thường giữ lại một ứng dụng trong bộ nhớ ngay cả khi nó không còn sử dụng, nhằm duy trì vùng nhớ tạm thời để quá trình khởi động lại ứng dụng được nhanh chóng. Các process này sẽ bị giải phóng khi có yêu cầu.

Thứ tự ưu tiên của các process

Theo chế độ ưu tiên thì khi cần tài nguyên, Android sẽ tự động giải phóng process có độ ưu tiên từ thấp lên cao, trước tiên là các empty process.



Chu kỳ sống của ứng dụng Android

- Mỗi ứng dụng chạy trong tiến trình riêng của mình do máy ảo của Android quản lý đều có chu kỳ sống của nó.

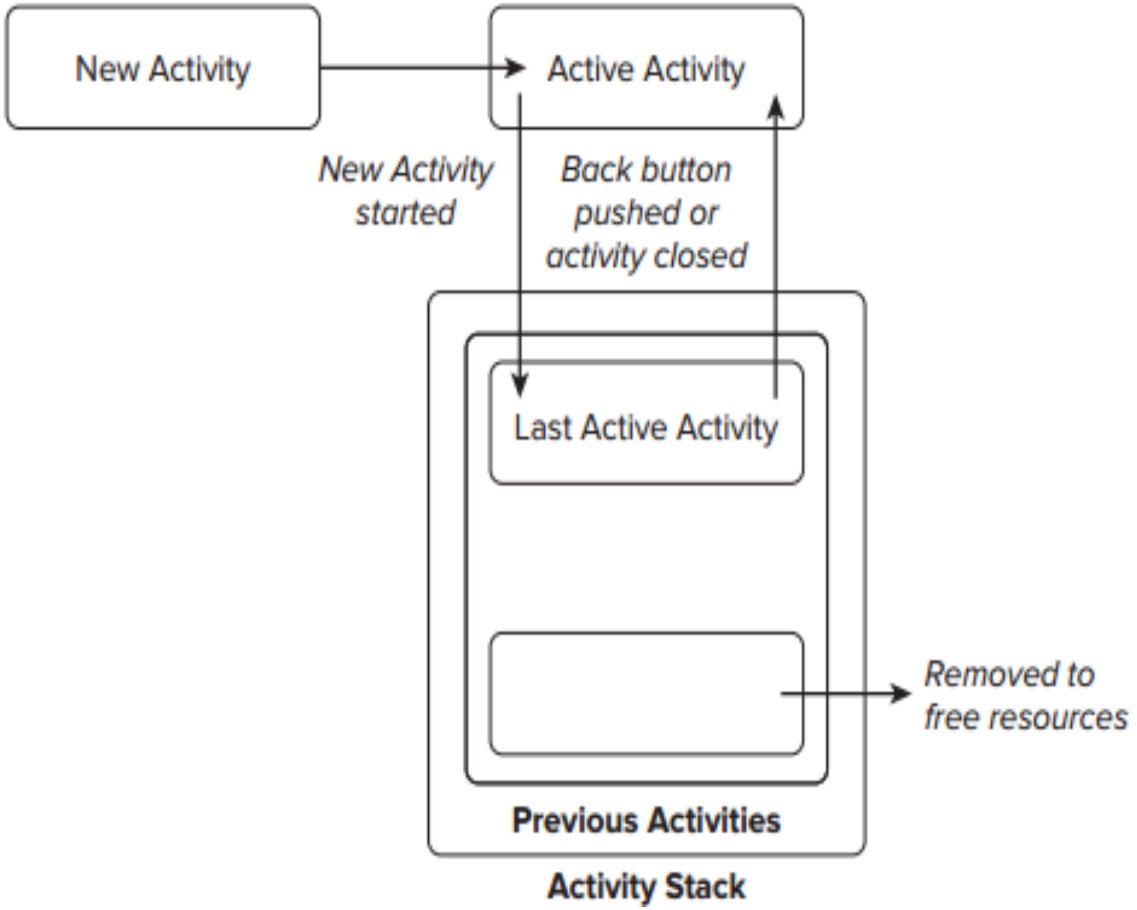
- Chu kỳ sống của ứng dụng Android phụ thuộc vào chu kỳ sống của các thành phần tạo nên ứng dụng Android, mỗi thành phần trong ứng dụng Android đều có một chu kỳ sống riêng.

- Các ứng dụng chỉ được gọi là kết thúc khi tất cả các thành phần trong ứng dụng kết thúc.

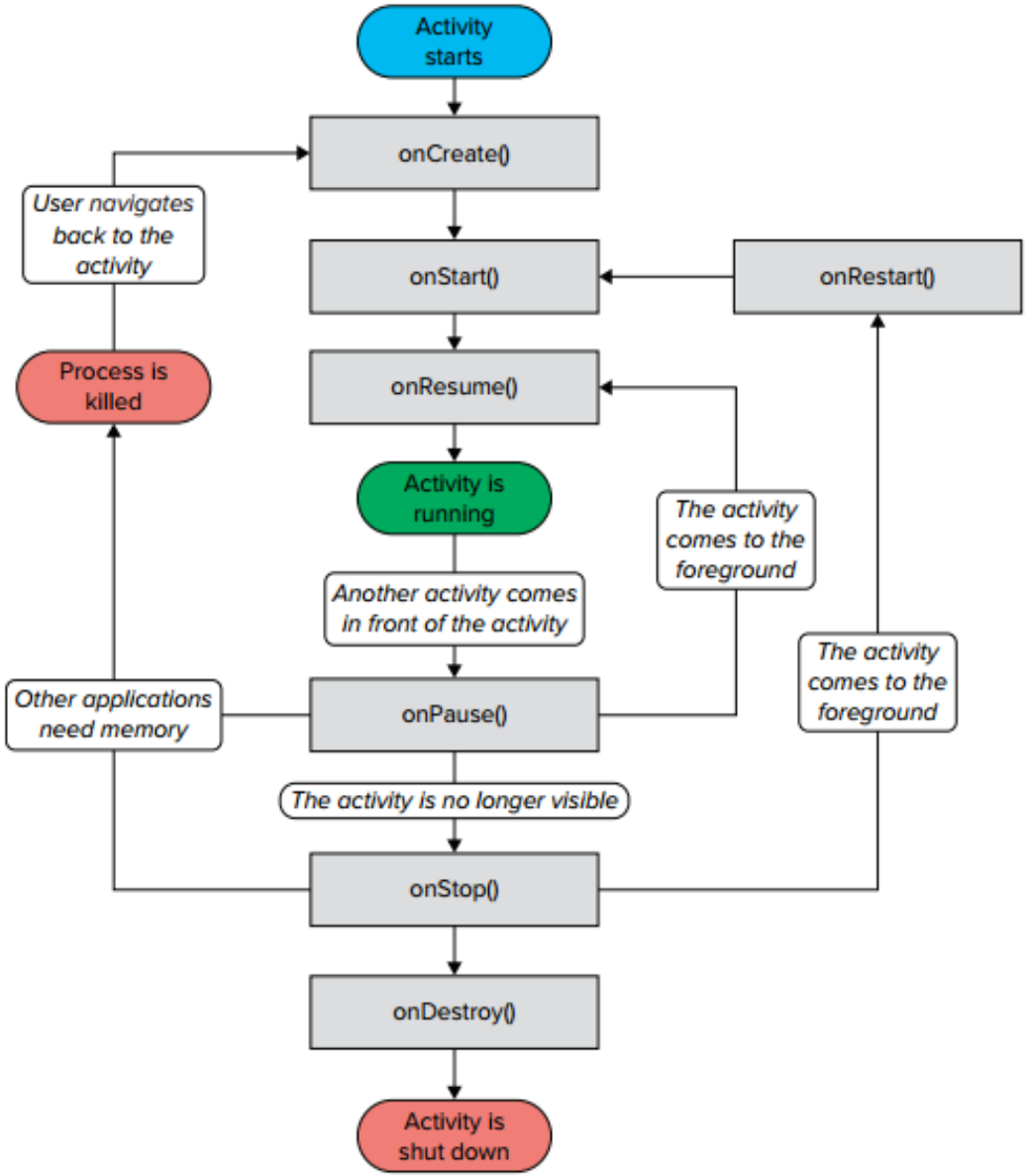
- Trong các thành phần tạo nên ứng dụng Android thì một thành phần tối quan trọng của bất kỳ một ứng dụng Android là Activity

Activity

- Thông thường trong một ứng dụng sẽ có một hoặc nhiều Activity. Mỗi một Activity này sẽ có một vòng đời riêng độc lập hoàn toàn với các Activity khác
- **Activity stack**



Mô hình vòng đời của Activity



Lập trình quản lý vòng đời của Activity trong ứng dụng Android

- ✓ Bước 1: Tạo Android project có tên Activity101
- ✓ Bước 2: Tạo một activity mới
- ✓ Bước 3: Quản lý vòng đời Activity qua các hàm sự kiện trong mô hình vòng đời Activity.
- ✓ Bước 4: Bấm F11 để debug ứng dụng trên Emulator

BTVN 1

- Tạo ứng dụng CheckLifeTimeCycle
  - Start: onCreate, onStart, onResume
  - Bấm nút Home: onPause, onStop
  - Bấm nút Back: onPause, onStop, onDestroyed
  - Có ứng dụng khác kích hoạt: onPause, onStop
  - Xoay màn hình: onP, onS, onD, onC, onStart, onRe
  - Tắt nguồn: onP, onStop

- Chọn menu Source/ Override/Implement methods để override các phương thức onStart, onRestart, onResume,onPause, onStop, onDestroy

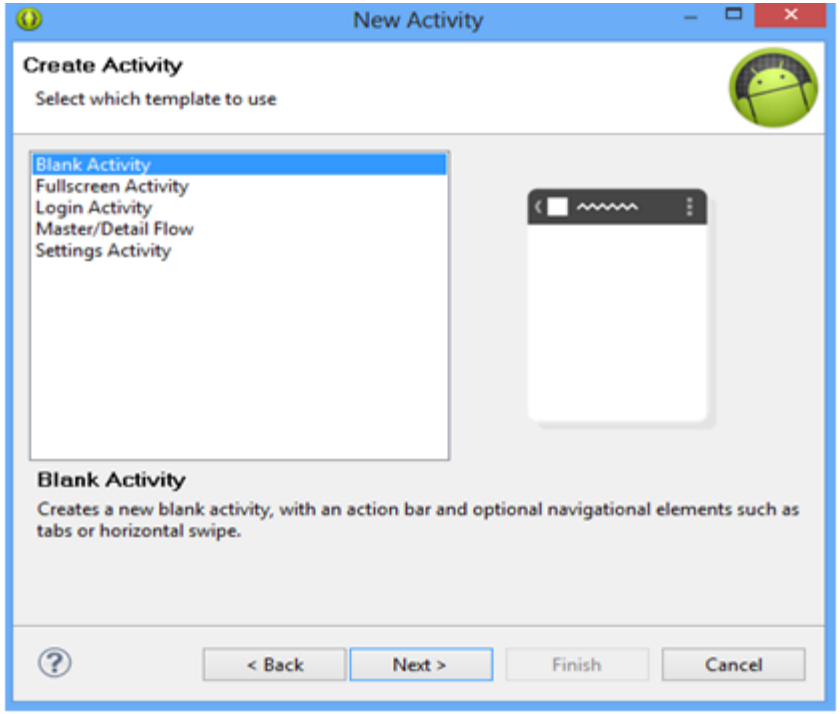
- Trong các hàm trên Viết lệnh hiển thị message cho mỗi sự kiện

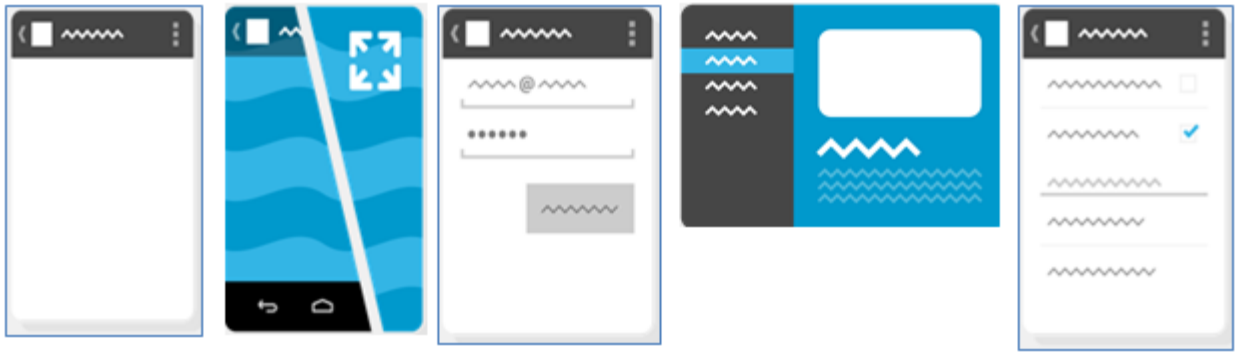
```
Toast.makeText(this,”onResume”, Toast.LENGTH_SHORT).show();
```

2. Lập trình giao diện ứng dụng di động Android

GUI Template /Android

- 5 mẫu giao diện cho Activity:
  - Blank Activity
  - Fullscreen Activity
  - Login Activity
  - Master/Detail Flow
  - Settings Activity





Hình 3.6. Các mẫu giao diện ứng dụng Android

Blank, Fullscreen, Login, Master/Detail Flow, Settings

ACTIVITY

- Activity dùng để hiển thị màn hình giao diện ứng dụng Android
- Giao diện ứng dụng được xây dựng từ các đối tượng View và ViewGroup
- Có nhiều kiểu View và ViewGroup. Mỗi một kiểu là một hậu duệ của class View và tất cả các kiểu đó được gọi là các widget.
- Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, background, kích thước, lề…

Thuộc tính	Miêu tả
layout_width	Chiều rộng của View/ ViewGroup
layout_height	Chiều cao của View/ ViewGroup
layout_marginTop	Khoảng cách trốngtính từ mép phía trên của View/ ViewGroup
layout_marginBottom	Khoảng cách trống tính từ mép phía dưới của View/ ViewGroup
layout_marginLeft	Khoảng cách trống tính từ mép bên trái của View/ ViewGroup
layout_marginRight	Khoảng cách trống tính từ mép bên phải của View/ ViewGroup
layout_gravity	Qui định vị trí của View đặt bên trong so với ViewLayout
layout_weight	Khoảng cách trống của layout cấp phát cho View
layout_x	Tọa độ x của View/ ViewGroup
layout_y	Tọa độ y của View/ ViewGroup

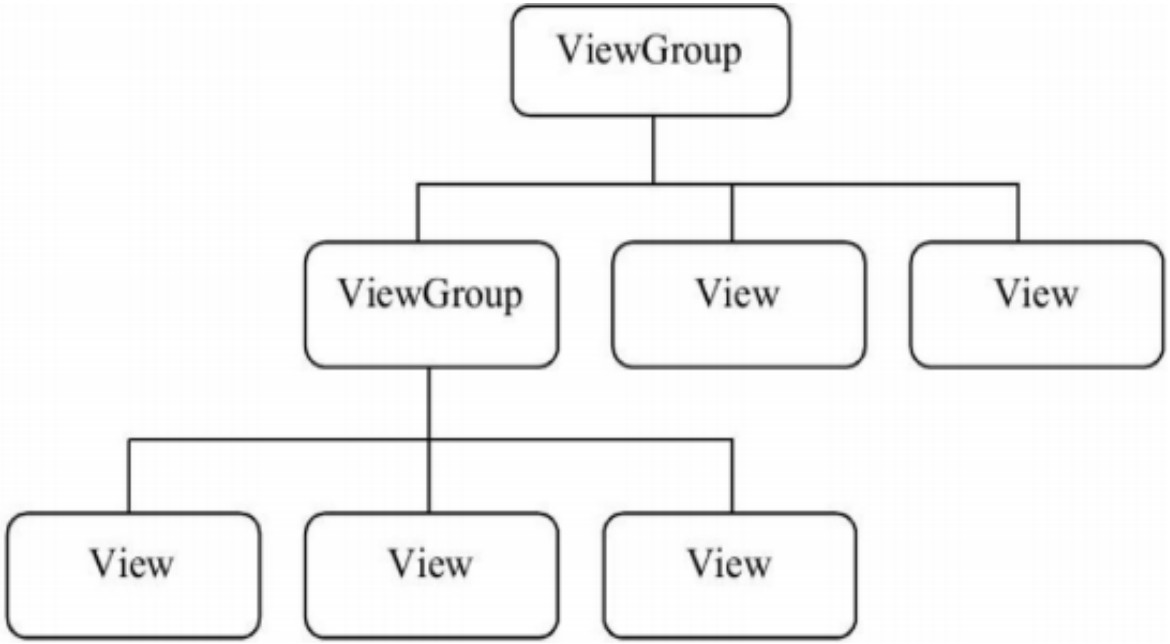
Các thuộc tính khoảng cách có giá trị với đơn vị đo lường thuộc 1 trong các loại sau:

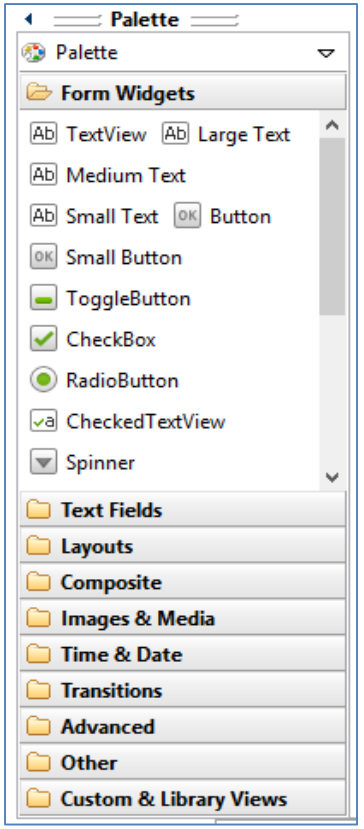
- dp (Density-independent pixel): đây là đơn vị đo lường được khuyến cáo sử dụng. 1dp tương ứng với 1 pixel trên màn hình phổ biến của thiết bị Android 160 dpi.
- sp (Scale-independent pixel): khuyến cáo sử dụng làm đơn vị của kích cỡ font.
- pt (Point): 1pt = 1/72 inch.
- px (Pixel): Đơn vị đo số lượng điểm ảnh thực tế trên màn hình thiết bị. Đơn vị này không được khuyến cáo sử dụng vì kích cỡ các phần tử trên giao diện ứng dụng có thể không cố định trên các thiết bị có độ phân giải màn hình khác nhau.

- Chú ý rằng chia thiết bị Android thành 4 loại với mật độ điểm ảnh trên màn hình khác nhau:

  - ✓ Màn hình có mật độ điểm ảnh thấp (ldpi) -120 dpi.
  - ✓ Màn hình có mật độ điểm ảnh trung bình (mdpi) -160 dpi.
  - ✓ Màn hình có mật độ điểm ảnh cao (hdpi) -240 dpi.
  - ✓ Màn hình có mật độ điểm ảnh cao nhất (xhdpi) -320 dpi.

Cấu trúc một giao diện ứng dụng Android





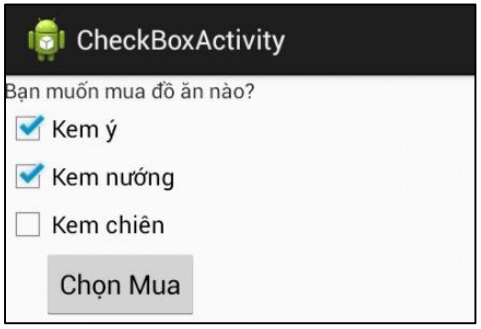
View

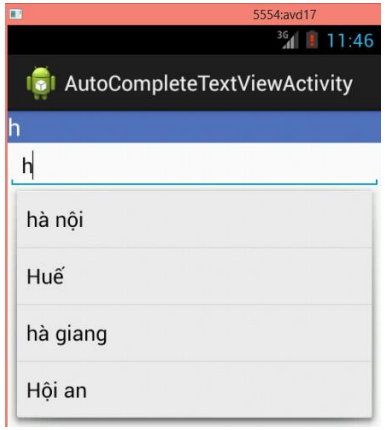
- Các view trong Android được định nghĩa sẵn trong lớp thư viện android.view.View.
- Các view này được chia thành 3loại chính sau :
  - Các view cơ bản
  - Các picker view
  - Các view hiển thị danh sách

View

-- Các view cơ bản --

- Các view cơ bản trong Android nhằm hiển thị thông tin văn bản hoặc cho phép lựa chọn thông tin
- Bao gồm : TextView, EditText, Button, ImageButton, CheckBox, ToggleButton, RadioButton, RadioGroup, Progressbar, AutoCompleteTextView

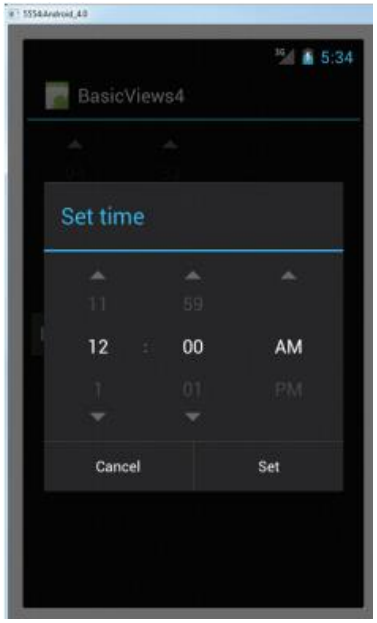


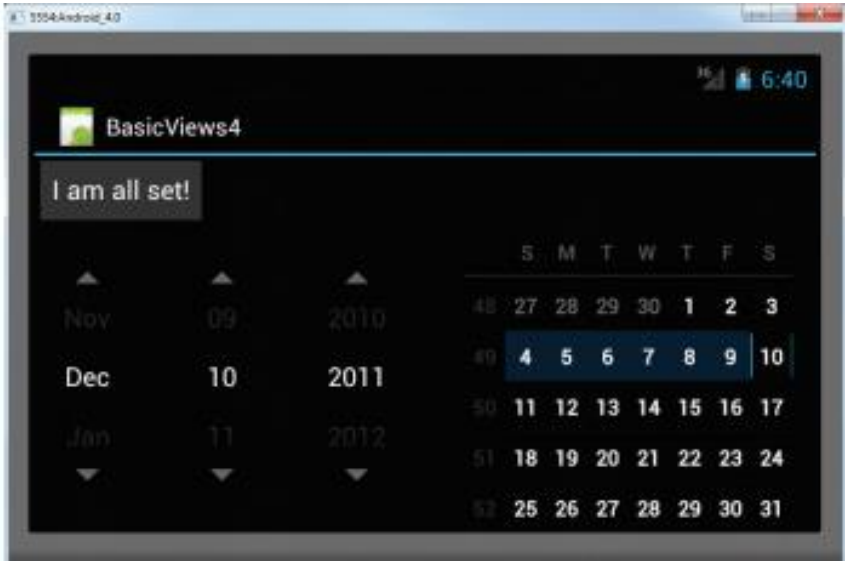


View

-- Các picker view --

- Picker được chia làm 2 loại : TimePicker, DatePicker. Việc sử dụng 2 loại picker này cho phép người dùng chọn thời gian, ngày tháng năm trên giao diện ứng dụng Android.

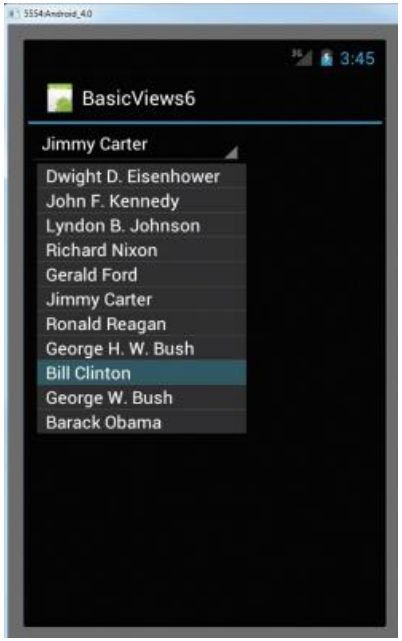




View

-- Các view hiển thị danh sách--

- Nhằm hiển thị danh sách mục thông tin.
- Có 2 loại view hiển thị danh sách: ListView, SpinnerView.



Xử lý sự kiện khi người dùng tương tác với các View

- Các view nhận được các tương tác từ người dùng sẽ xử lý các sự kiện tương ứng.
- Các sự kiện này được Android Framework quản lý bởi 1 hàng đợi (queue) hoạt động theo cơ chế FIFO (First In First Out).
- Các bước cần làm để xử lý sự kiện cho các đối tượng giao diện:
  - Đăng ký lắng nghe sự kiện (Event Listenters Registration).
  - Khai báo đối tượng nhận thông báo khi có sự kiện xuất hiện (Event Listenters).
  - Xử lý sự kiện (Event Handlers)

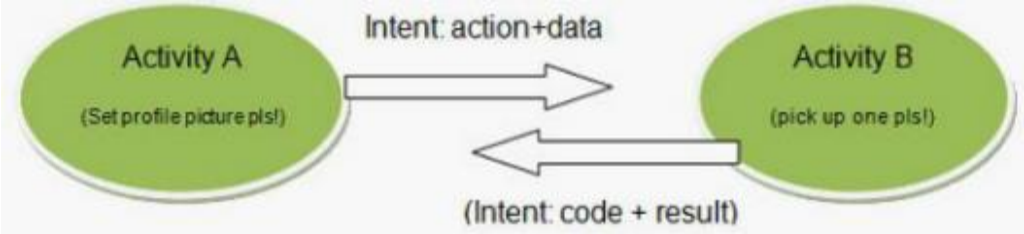
ViewGroup

- ViewGroup thực ra chính là View hay nói đúng hơn thì ViewGroup chính là các widget layout được dùng để bố trí các đối tượng khác trong một màn hình giao diện.
- Các ViewGroup trong Android được định nghĩa sẵn trong lớp thư viện android.view.ViewGroup.
- Các ViewGroup này được chia thành các loại chính sau :
  - LinearLayout
  - AbsoluteLayout
  - TableLayout
  - RelativeLayout
  - FrameLayout
  - ScrollView
- Có thể kết hợp các loại ViewGroup khác nhau để tạo ra giao diện ứng dụng Android.

INTENT

- Ứng dụng Android thường bao gồm nhiều Activity, mỗi Activity hoạt động độc lập với nhau và thực hiện những công việc khác nhau.

- Intent giống như người đưa thư, giúp Activity này có thể triệu gọi cũng như truyền dữ liệu cần thiết tới một Activiy khác.

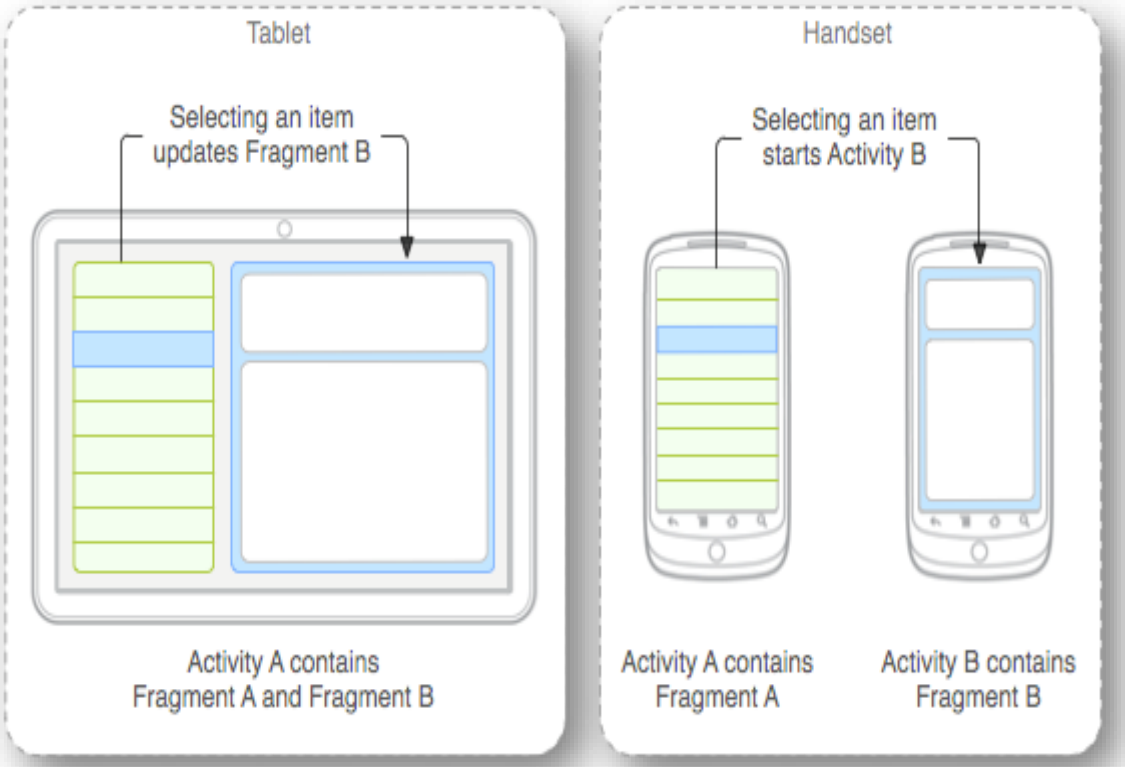


Hình 3.28. Dùng Intent để truyền dữ liệu giữa 2 Activity

FRAGMENT

Fragment là một thành phần giao diện được nhúng vào Activity cho phép thiết kế Activity linh động.

Fragment có thể coi như **sub-activity.**



Hình 3.29. Ví dụ kết hợp các Fragment trên một Activity Giao diện ứng dụng trên máy tính bảng (trái) và điện thoại di động (phải)

❖ Phân loại Fragment:

- ListFragment : dùng để hiển thị danh sách mục tin.
- DialogFragment : fragment hiển thị dưới dạng dialog động phía trên Activity.
- PreferenceFragment : dùng để hiển thị phân cấp các đối tượng.

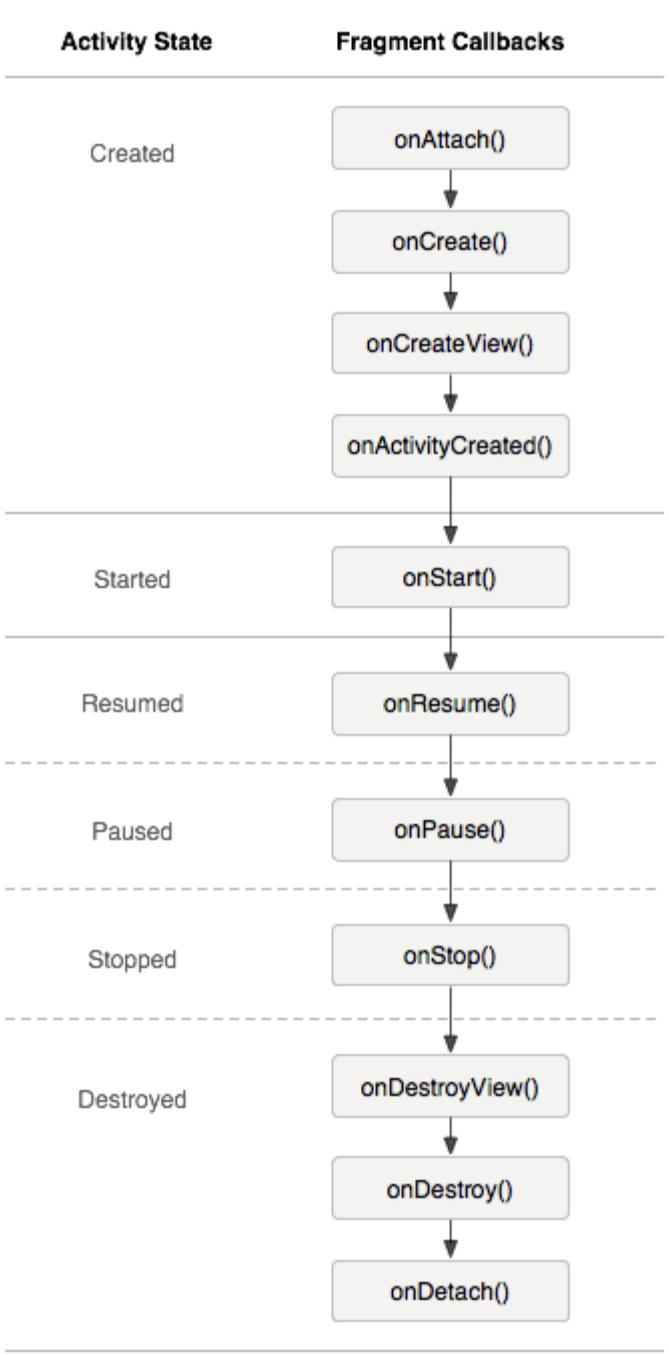
❖ Đặc điểm của Fragment:

- Fragment có layout và vòng đời hoạt động riêng. Vòng đời hoạt động của Fragment phụ thuộc vào vòng đời của Activity chứa nó.
- Fragment có thể được thêm vào hoặc gỡ bỏ khỏi Activity trong khi Activity đang hoạt động.
- Một Activity có thể được hình thành từ nhiều Fragment, một Fragment có thể được sử dụng bởi nhiều Activity.
- Fragment được đưa vào từ Android API 11.

❖ Cách tạo và sử dụng Fragment:

- Tạo Fragment bằng cách tạo 1 lớp kế thừa từ class Fragment cơ sở.
- Tạo giao diện cho Fragment dưới dạng mã XML
- Nhúng Fragment vào Activity bằng cách khai báo thành phần <fragment> trong file giao diện của Activity.

Vòng đời fragment



SV chia nhóm báo cáo

Trình bày cách thức sử dụng + ví dụ về:

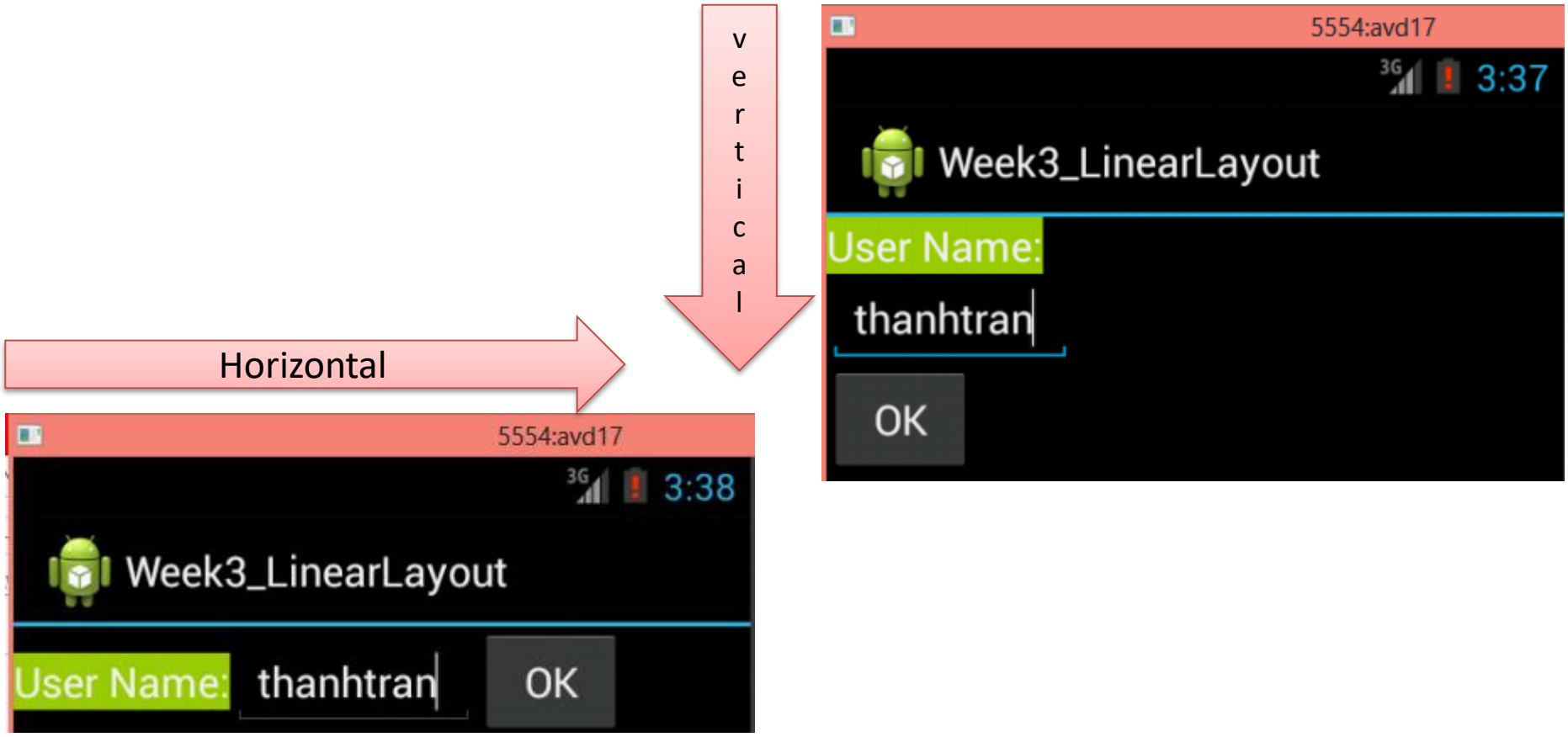
1. Layout.
2. Fragment, TabSelection.
3. Intent
4. TextView, EditText.
5. Button, ImageButton.
6. CheckBox, ToggleButton.
7. RadioButton, RadioGroup.
8. TimePicker, DatePicker.
9. ScrollView, ImageView.
10. ListView, Spinner.
11. AutoCompleteTextView, Gallery.
12. SlideDrawer, Menu.
13. Xử lý sự kiện khi người dùng tương tác với các View

Bài tập 1: Layout

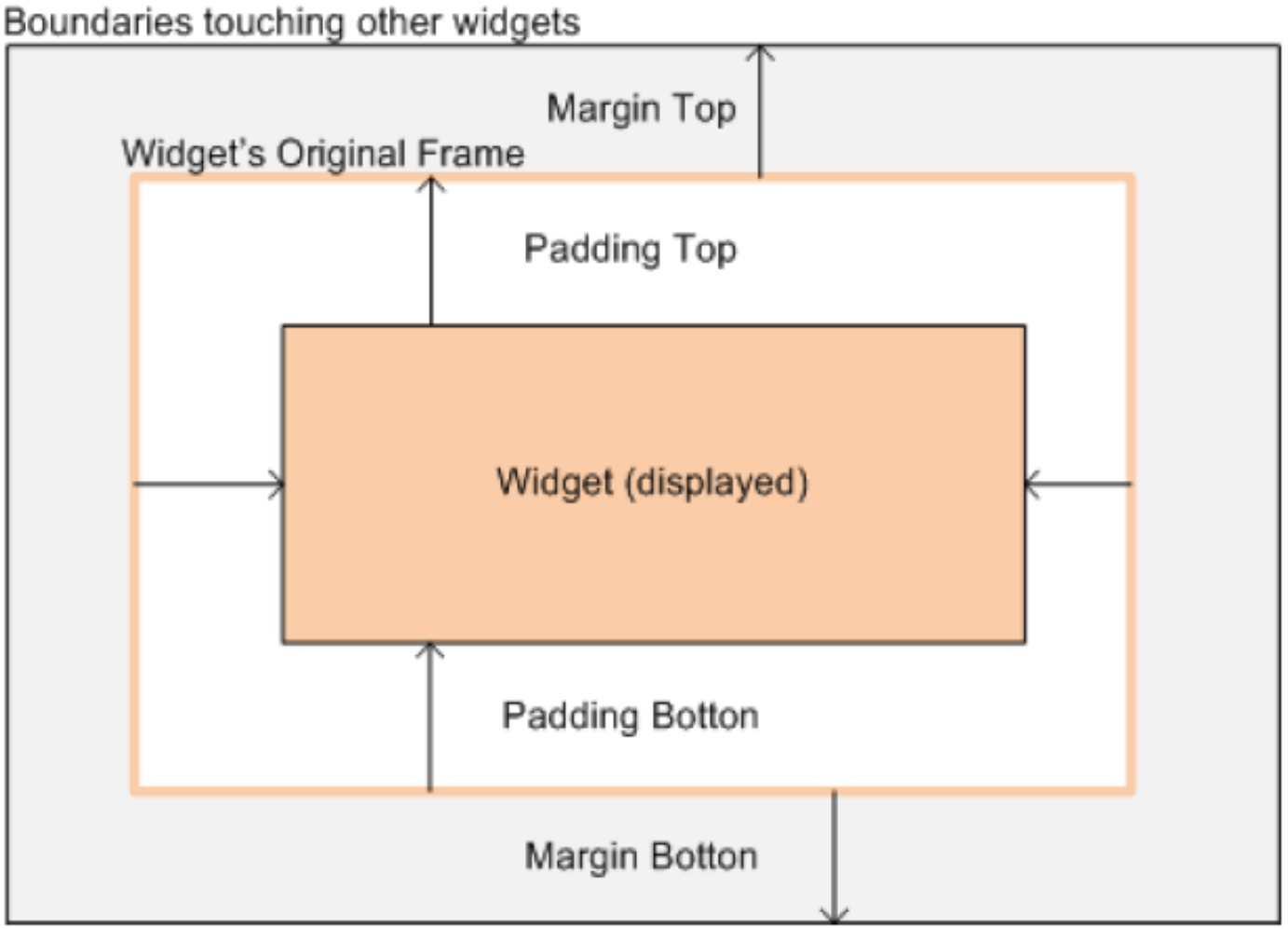
ViewGroup

- ✓ LinearLayout
- ✓ AbsoluteLayout
- ✓ TableLayout
- ✓ RelativeLayout
- ✓ FrameLayout

LinearLayout



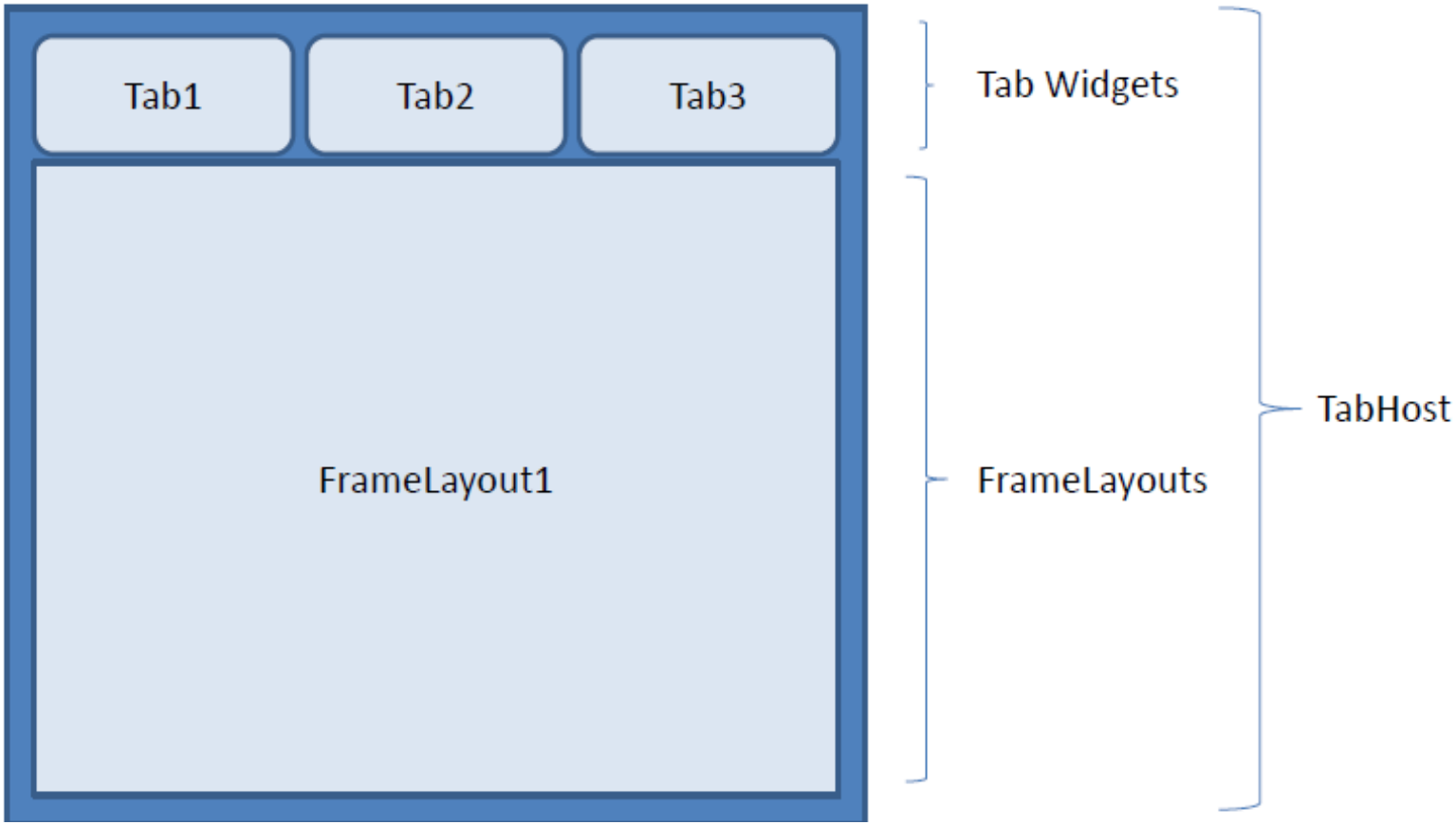
LinearLayout



Bài tập 2: Fragment, TabSelection

Tab Selector

Components



Bài tập 3: TextView, EditText

TextView

- Cách tạo:
  - Kéo thả TextView từ Palette.
- Sử dụng:
  - Sau khi kéo thả TextView từ Palette => trong activity_main.xml xuất hiện Tab:

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="154dp"
    android:text="TextView" />
```

TextView

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
TextView tv1;
tv1 = (TextView) findViewById(R.id.textView1);
tv1.setText("Hello! How are you?");
```

EditText

- Cách tạo:
  - Kéo thả EditText (hay Plain Text) từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="36dp"
    android:ems="10" >
```

EditText

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
EditText ed1;
ed1 = (EditText) findViewById(R.id.editText1);
String t = ed1.getText().toString(); //Lấy nội dung trong EditText.

//Nếu muốn convert sang số nguyên (nếu dữ liệu là dạng số):
int i = Integer.parseInt(ed1.getText().toString());
```

Bài tập 4: Button, ImageButton

Button

- Cách tạo:
  - Kéo thả Button từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editText1"
    android:layout_centerHorizontal="true"
    android:text="Button" />
```

Button

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
Button bt1;
bt1 = (Button) findViewById(R.id.button1);
bt1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
        // TODO Auto-generated method stub
                tv1.setText(""+ed1.getText().toString());
        }
});

//Lưu ý: khi đó TextView tv1 và EditText ed1 phải có tính chất final.
```

ImageButton

- Cách tạo:
  - Kéo thả ImageButton từ Palette => chọn Create New Icon => đặt tên cho Icon Name
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<ImageButton
    android:id="@+id/imageButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="61dp"
    android:src="@drawable/imagebtn" />
```

ImageButton

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
ImageButton imgbt1;
imgbt1 = (ImageButton) findViewById(R.id.button1);
imgbt1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
        // TODO Auto-generated method stub
                tv1.setText(""+ed1.getText().toString());
        }
});

//Lưu ý: khi đó TextView tv1 và EditText ed1 phải có tính chất final.
```

Bài tập 5: CheckBox, ToggleButton

CheckBox

- Cách tạo:
  - Kéo thả CheckBox từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<CheckBox
    android:id="@+id/checkBox1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/imageButton1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="23dp"
    android:text="CheckBox" />
```

CheckBox

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
CheckBox chk = (CheckBox) findViewById(R.id.checkBox1);

if (chk.isChecked()){
        tv1.setText("CheckBox is checked");
}else
        tv1.setText("CheckBox is UNchecked");
```

ToggleButton

- Cách tạo:
  - Kéo thả ToggleButton từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<ToggleButton
    android:id="@+id/toggleButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/button1"
    android:layout_below="@+id/checkBox1"
    android:text="ToggleButton" />
```

ToggleButton

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
ToggleButton tgbt1 = (ToggleButton) findViewById
(R.id.toggleButton1);
if (tgbt.isChecked()){
        tgb1.setText(" ToggleButton is checked");
}else
        tgb1.setText(" ToggleButton is UNchecked");
```

Bài tập 6: RadioButton, RadioGroup

RadioGroup

- Cách tạo:
  - Kéo thả RadioGroup từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<RadioGroup
    android:id="@+id/radioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/imageButton1"
    android:layout_below="@+id/radioButton1"
    android:layout_marginTop="16dp" >

    <RadioButton
        android:id="@+id/radio0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="RadioButton" />

    <RadioButton
        android:id="@+id/radio1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="RadioButton1" />

    <RadioButton
        android:id="@+id/radio2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="RadioButton2" />
</RadioGroup>
```

RadioGroup

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
RadioGroup rdgroup = (RadioGroup)findViewById(R.id.radioGroup1);
int idChecked = rdgroup.getCheckedRadioButtonId();

if (idChecked == R.id.radio0){
   tv2.setText("Radio0 is checked");
}
if (idChecked == R.id.radio1){
   tv2.setText("Radio1 is checked");
}
if (idChecked == R.id.radio2){
   tv2.setText("Radio2 is checked");
}
```

Bài tập 7: TimePicker, DatePicker

TimePicker

- Cách tạo:
  - Kéo thả TimePicker từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<TimePicker
    android:id="@+id/timePicker1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="22dp" />
```

TimePicker

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
TimePicker tpk1 = (TimePicker) findViewById(R.id.timePicker1);
tpk1.setIs24HourView(true);
Toast.makeText(getBaseContext(),"Time
"+tpk1.getCurrentHour()+":"+tpk1.getCurrentMinute(),
Toast.LENGTH_SHORT).show();
```

DatePicker

- Cách tạo:
  - Kéo thả TimePicker từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<DatePicker
        android:id="@+id/datePicker1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true" />
```

DatePicker

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
DatePicker dpk1 = (DatePicker)findViewById(R.id.datePicker1);
Toast.makeText(getApplicationContext(), "Date:
"+dpk1.getDayOfMonth()+
"/"+dpk1.getMonth()+"/"+dpk1.getYear(),
Toast.LENGTH_SHORT).show();
```

Bài tập 8: ScrollView, ImageView

ScrollView

- Cách tạo:
  - Kéo thả ScrollView từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:orientation="vertical" >

            <TextView
                android:id="@+id/textView2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Large Text"
                android:textAppearance="?android:attr/textAppearanceLarge" />
            <Button
                android:id="@+id/button1"
                style="?android:attr/buttonStyleSmall"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Button" />
        </LinearLayout>
    </ScrollView>
```

ImageView

- Cách tạo:
  - Copy (hoặc link) file hình vào trong các thư mục drawable.
  - Kéo thả ImageView từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong activity_main.xml xuất hiện Tab:

```
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button3_2"
    android:layout_alignParentBottom="true"
    android:src="@drawable/vd" />
```

ImageView

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
ImageView imgview =
(ImageView)findViewById(R.id.imageView2);
```

Bài tập 9: ListView, Spinner

ListView

- Cách tạo:
  - Kéo thả ListView từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong <file>.xml xuất hiện Tab:

```
<ListView
        android:id="@+id/listView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="59dp" >
</ListView>
```

ListView

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…)):

```
String[] students = {"Nam","Lan","Hoa","Hanh","Thanh",
                                        "Trung","Son","Dung","Van","Manh","Thang"};

ListView lv = (ListView)findViewById(R.id.listView1);

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

lv.setAdapter(adapter);

lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        Toast.makeText(getBaseContext(), "Position :"+arg2+" - Value = "+students[arg2],
        Toast.LENGTH_SHORT).show();
        }
});
```

Spinner

- Cách tạo:
  - Kéo thả Spinner từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong <file>.xml xuất hiện Tab:

```
<Spinner
    android:id="@+id/spinner1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="124dp" />
```

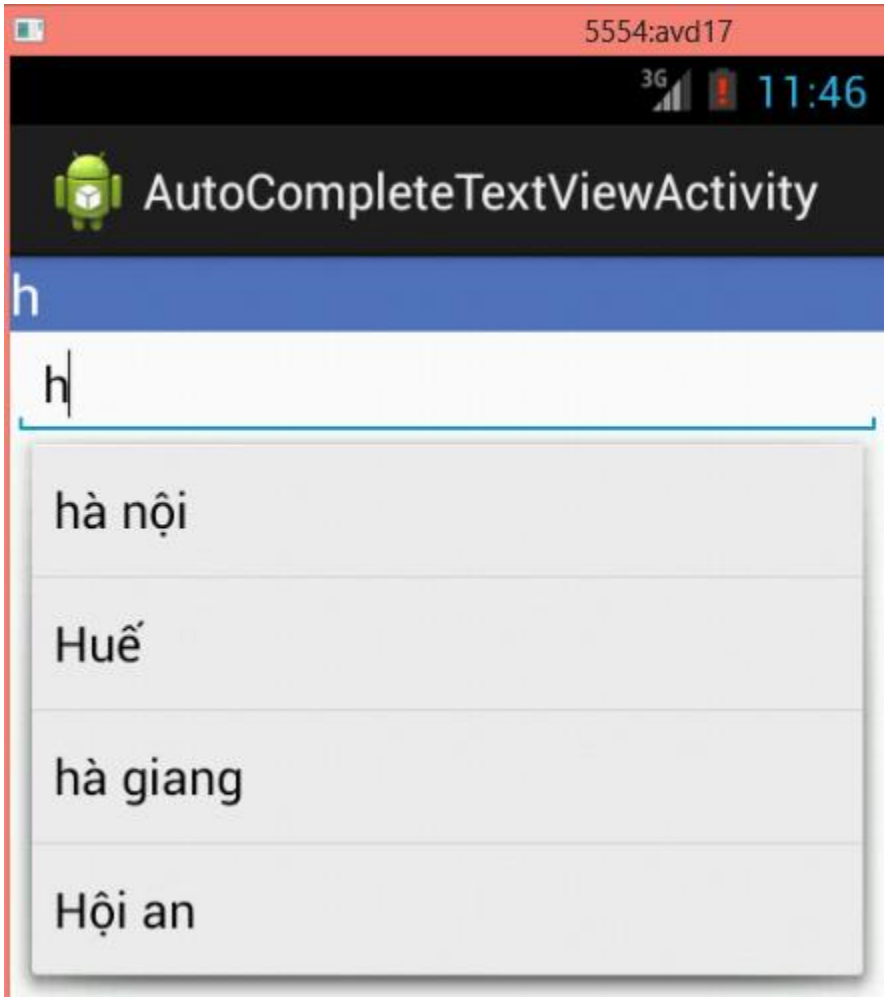
Spinner

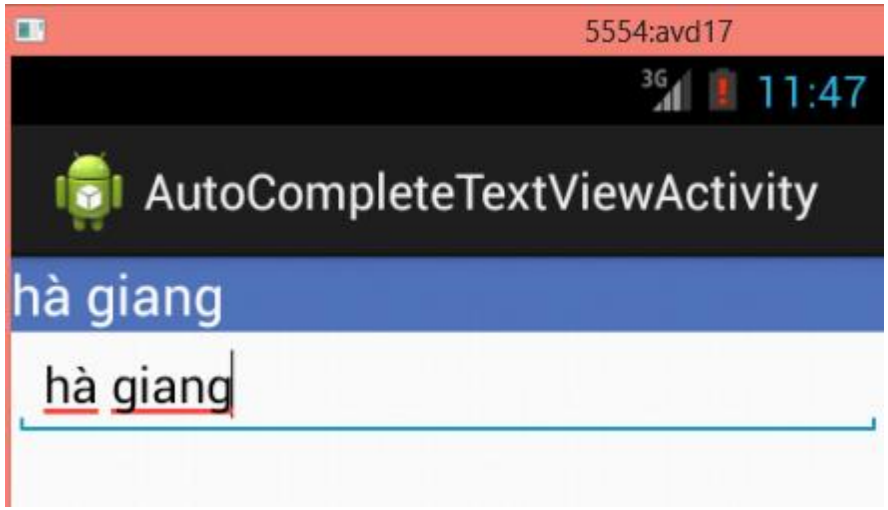
- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…):

```
String goods[] = {"TV","Labtop","SmartPhone“, "Watch","Book"};

Spinner spin = (Spinner) findViewById(R.id.spinner1);

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

adapter.setDropDownViewResource(android.R.layout.simple_list_item_single_choice);

spin.setAdapter(adapter);

spin.setOnItemSelectedListener(new OnItemSelectedListener() {

        @Override
        public void onItemSelected(AdapterView<?> arg0, View arg1,int arg2, long arg3) {
        Toast.makeText(getBaseContext(), "Position: "+arg2+" - Selected Value:"+goods[arg2],Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onNothingSelected(AdapterView<?> arg0) {
        // TODO Auto-generated method stub

        }
});
```

Bài tập 10:
AutoCompleteTextView, Gallery

AutocompleteTextView





AutoCompleteTextView

- Cách tạo:
  - Kéo thả AutoCompleteTextView và MultiAutoCompleteTextView từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong <file>.xml xuất hiện Tab:

```
<AutoCompleteTextView
    android:id="@+id/autoCompleteTextView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="90dp"
    android:ems="10“
    android:completionThreshold="1"
    android:text="" />

<MultiAutoCompleteTextView
    android:id="@+id/multiAutoCompleteTextView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/autoCompleteTextView1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="49dp"
    android:ems="10“
    android:completionThreshold="1"
    android:text="" />
```

AutoCompleteTextView

- Sử dụng:
  - Khai báo trong MainActivity.java (ví dụ trong hàm onCreate(…)):

```
String[] students =
{"Nam","Lan","Hoa","Hanh","Thanh","Trung","Son","Dung","Van","Manh","Thang"};

AutoCompleteTextView autoComTV =
(AutoCompleteTextView)findViewById(R.id.autoCompleteTextView1);
autoComTV.addTextChangedListener(this);
autoComTV.setAdapter(new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,students));

MultiAutoCompleteTextView mulComTV = (MultiAutoCompleteTextView)findViewById(R.id.multiAutoCompleteTextView1);
mulComTV.setAdapter(new ArrayAdapter<String>(this,android.R.layout.simple_dropdown_item_1line,students));
mulComTV.setTokenizer(new MultiAutoCompleteTextView.CommaTokenizer());
```

Picture Gallery

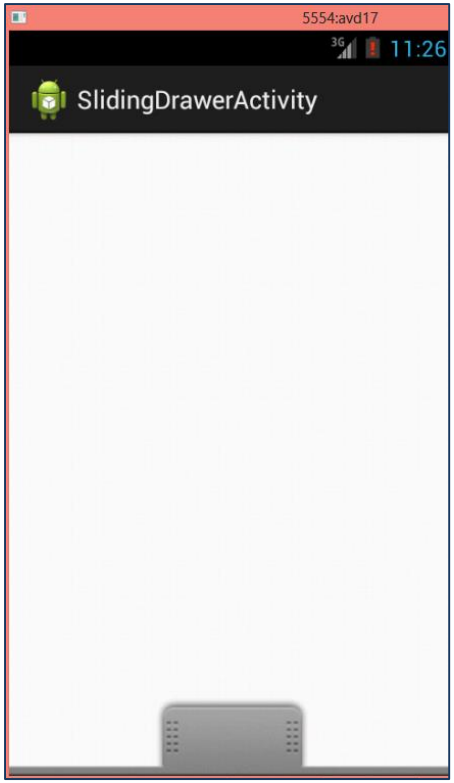
Picture Gallery

- Cách tạo:
  - Kéo thả ImageView và Gallery từ Palette.
- Sử dụng:
  - Sau khi kéo thả Palette => trong <file>.xml xuất hiện Tab:

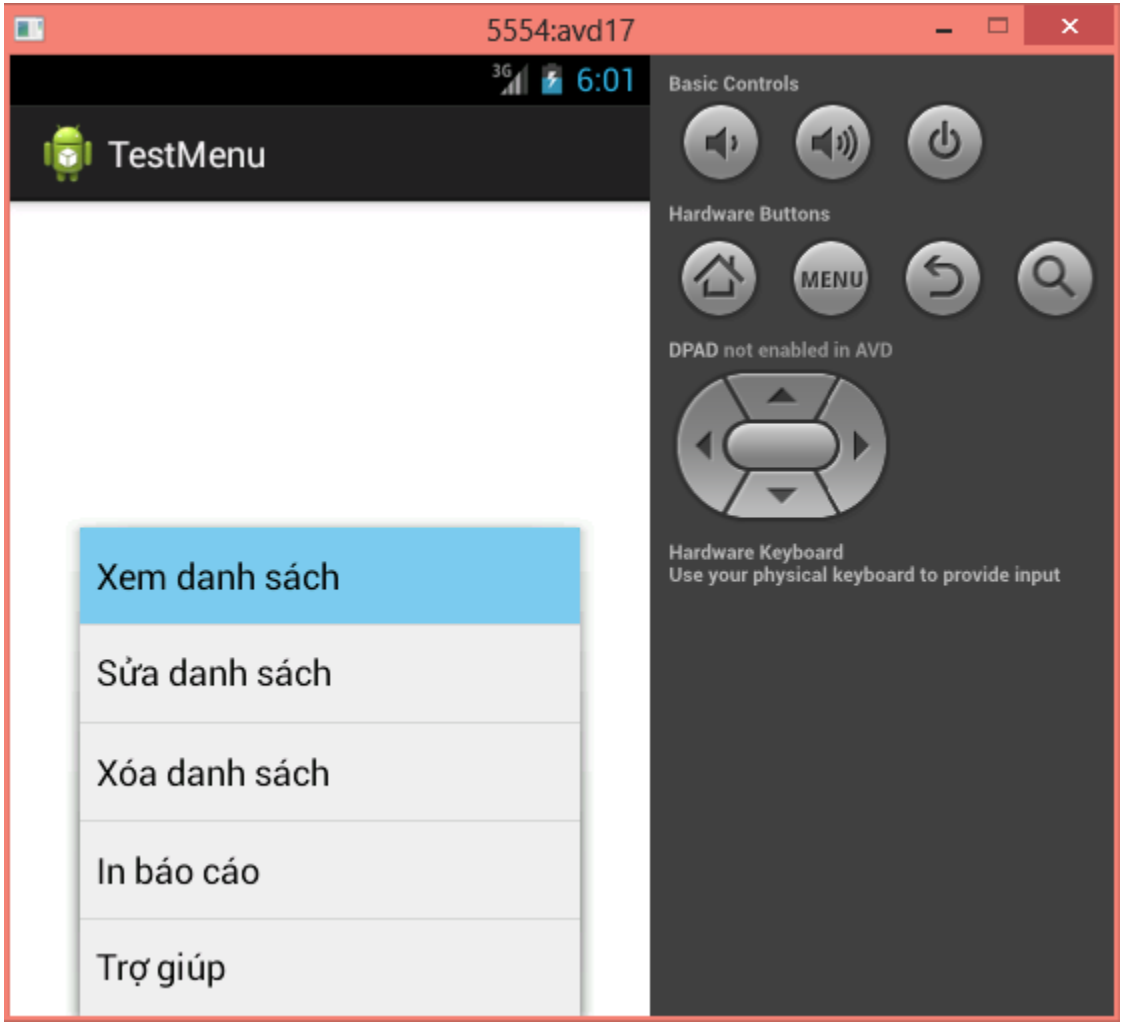
```
<Gallery
    android:id="@+id/Gallery01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" >
</Gallery>

<ImageView
    android:id="@+id/ImageView01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >
</ImageView>
```

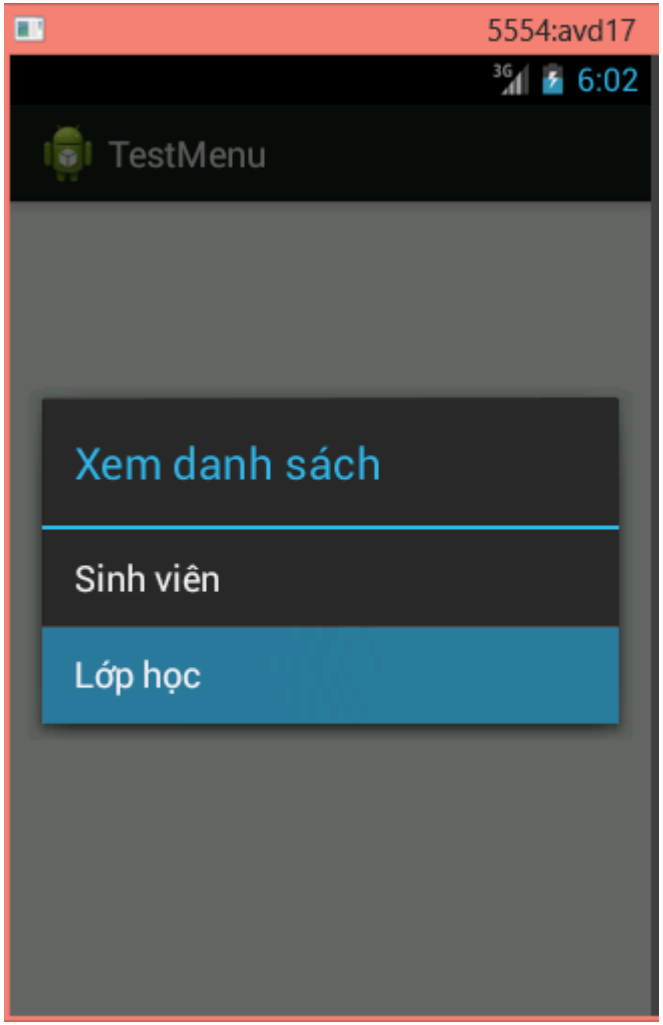
Bài tập 11: SlideDrawer, Menu

SlideDrawer



Menu


5554:avd17
6:01
TestMenu
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input
Xem danh sách
Sửa danh sách
Xóa danh sách
In báo cáo
Trợ giúp


5554:avd17
6:02
TestMenu
Xem danh sách
Sinh viên
Lớp học

BTVN: Trình bày về quản lý dữ liệu và đưa ra ví dụ cụ thể

1. Quản lý dữ liệu dạng key-value
   - Tạo và kết nối tới file SharedPreferences.
   - Ghi dữ liệu vào SharePreferences.
   - Đọc dữ liệu từ SharePreferences.

2. Quản lý dữ liệu từ file trong bộ nhớ thiết bị
   - Đọc, ghi vào file (bộ nhớ trong).
   - Đọc, ghi vào file (bộ nhớ ngoài).

3. Lập trình quản lý dữ liệu

Các cách quản lý dữ liệu cho ứng dụng Android

1. Quản lý dữ liệu với SharedPreferences
2. Quản lý dữ liệu từ file trong bộ nhớ thiết bị
3. Quản lý dữ liệu với cơ sở dữ liệu quan hệ SQLite