



**FPT POLYTECHNIC**

THIẾT KẾ GIAO DIỆN TRÊN  
ANDROID

**Bài 5: Pickers-Tabs-  
Sliding**

---

[www.poly.edu.vn](http://www.poly.edu.vn)

---

## Nội dung bài học

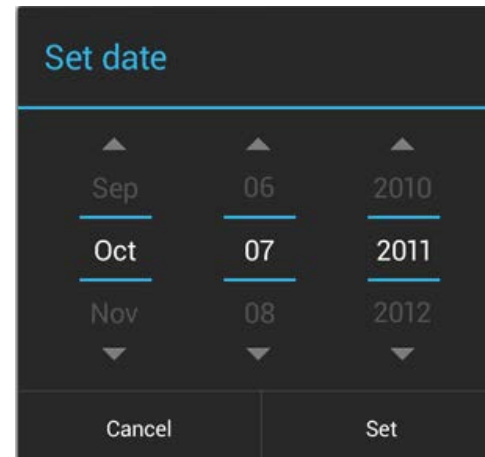
1. Pickers Widget
2. Tabs selector
3. SlidingDrawer



# 1. Pickers Widgets

## Date

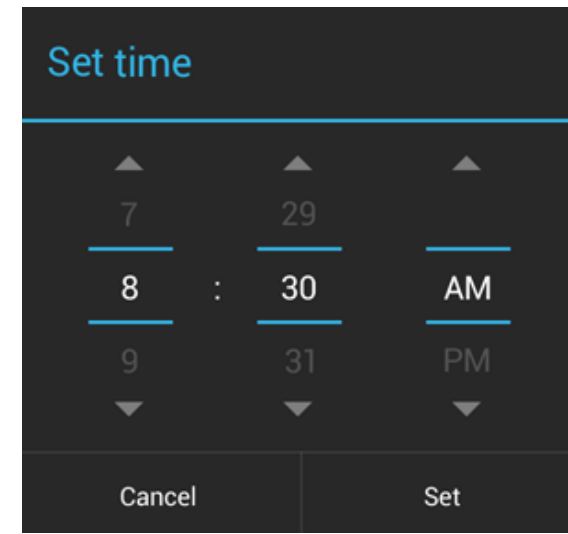
- Android cung cấp công cụ widgets (DatePicker, TimePicker) và dialogs (DatePickerDialog, TimePickerDialog) để người dùng nhập và hiển thị ngày, tháng.
- Trong đó DatePicker và DatePickerDialog dùng để thiết lập ngày, tháng, năm. Giá trị tháng chạy từ 0 (January) đến 11(December)
- Hai hàm sử dụng để gọi : **OnDateChangeListener** và **OnDateSetListener**



# 1. Pickers Widgets

## Time.

- Trong đó TimePicker và TimePickerDialog dùng để thiết lập giờ, phút, giây. Giá trị giờ (từ 0-23) và phút (0-59).
- Có thể chọn một trong hai dạng hiển thị 12-hour mode (AM/PM) hoặc 24-hour mode
- Hai hàm sử dụng để xử lý thông tin khi người dùng thay đổi thông tin thời gian: **OnTimeChangeListener** và **OnTimeSetListener**



# 1. Pickers Widgets

## Tạo Time Picker

- Để hiển thị `TimePickerDialog` sử dụng `DialogFragment`, cần định nghĩa lớp `fragment`, là một lớp kế thừa của `DialogFragment` và trả lại `TimePickerDialog` bằng hàm `onCreateDialog()`.
- Để định nghĩa `DialogFragment` cho `TimePickerDialog` cần:
  - Định nghĩa hàm `onCreateDialog()` được trả lại bởi một đối tượng của `TimePickerDialog`.
  - Thực hiện cuộc gọi `TimePickerDialog.OnTimeSetListener`

# 1. Pickers Widgets

- Ví dụ về kế thừa DialogFragment

```
public static class TimePickerFragment extends DialogFragment
    implements TimePickerDialog.OnTimeSetListener {

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // Use the current time as the default values for the picker
        final Calendar c = Calendar.getInstance();
        int hour = c.get(Calendar.HOUR_OF_DAY);
        int minute = c.get(Calendar.MINUTE);
        // Create a new instance of TimePickerDialog and return it
        return new TimePickerDialog(getActivity(), this, hour, minute,
            DateFormat.is24HourFormat(getActivity()));
    }

    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
        // Do something with the time chosen by the user
    }
}
```

# 1. Pickers Widgets

- Để sử dụng time picker, ta thực hiện như sau:
  - Thiết kế một button có gọi hàm showTimeDialog như sau:

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/pick_time"  
    android:onClick="showTimePickerDialog" />
```

- Khi click vào button, hệ thống sẽ gọi hàm sau:

```
public void showTimePickerDialog(View v) {  
    DialogFragment newFragment = new TimePickerFragment();  
    newFragment.show(getSupportFragmentManager(), "timePicker");  
}
```

# 1. Pickers Widgets

## Tạo Date Picker

- Để hiển thị DatePickerDialog sử dụng DialogFragment, cần định nghĩa lớp fragment, là một lớp kế thừa của DialogFragment và trả lại DatePickerDialog bằng hàm onCreateDialog().
- Để định nghĩa DialogFragment cho DatePickerDialog cần:
  - Định nghĩa hàm onCreateDialog() được trả lại bởi một đối tượng của DatePickerDialog.
  - Thực hiện lời gọi DatePickerDialog.OnDateSetListener



# 1. Pickers Widgets

## ● Ví dụ về kế thừa DialogFragment

```
public static class DatePickerFragment extends DialogFragment
    implements DatePickerDialog.OnDateSetListener {

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // Use the current date as the default date in the picker
        final Calendar c = Calendar.getInstance();
        int year = c.get(Calendar.YEAR);
        int month = c.get(Calendar.MONTH);
        int day = c.get(Calendar.DAY_OF_MONTH);
        // Create a new instance of DatePickerDialog and return it
        return new DatePickerDialog(getActivity(), this, year, month, day);
    }
    public void onDateSet(DatePicker view, int year, int month, int day) {
        // Do something with the date chosen by the user
    }
}
```

# 1. Pickers Widgets

- Để sử dụng date picker, ta thực hiện như sau:
  - Thiết kế một button có gọi hàm showDateDialog như sau:

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/pick_date"  
    android:onClick="showDatePickerDialog" />
```

- Khi click vào button, hệ thống sẽ gọi hàm sau:

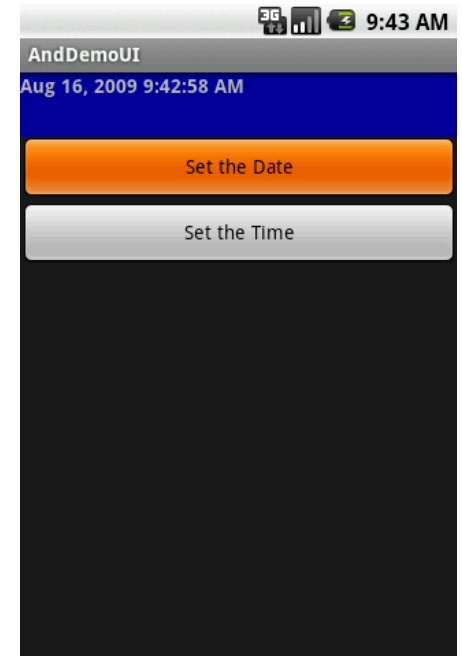
```
public void showDatePickerDialog(View v) {  
    DialogFragment newFragment = new DatePickerFragment();  
    newFragment.show(getSupportFragmentManager(), "datePicker");  
}
```

# 1. Pickers Widgets

## ● Ví dụ:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/widget28"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android"
>
<TextView
    android:id="@+id/lblDateAndTime"
    android:layout_width="fill_parent"
    android:layout_height="47px"
    android:background="#ff000099"
    android:textStyle="bold"
>
</TextView>
```

```
</Button>
<Button android:id="@+id/btnTime"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Set the Time"
>
</Button>
</LinearLayout>
```



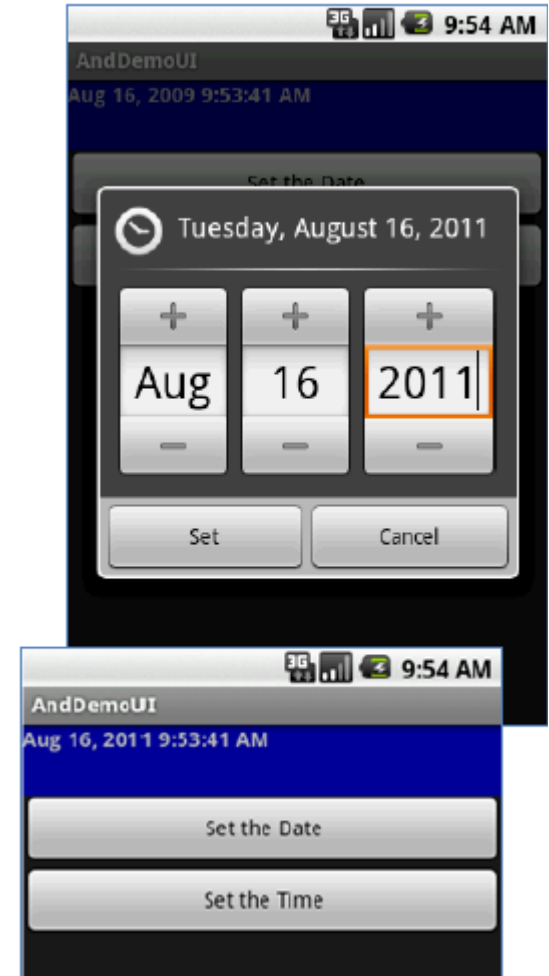
# 1. Pickers Widgets

Ví dụ:

```
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TimePicker;
import android.widget.TextView;
import java.text.DateFormat;
import java.util.Calendar;
public class AndDemoUI extends Activity {
    DateFormat fmtDateAndTime = DateFormat.getDateTimeInstance();
    TextView lblDateAndTime;
    Calendar myCalendar = Calendar.getInstance();
    DatePickerDialog.OnDateSetListener d = new DatePickerDialog.OnDateSetListener()
    {
        public void onDateSet(DatePicker view, int year,
                               int monthOfYear, int dayOfMonth) {
            myCalendar.set(Calendar.YEAR, year);
            myCalendar.set(Calendar.MONTH, monthOfYear);
            myCalendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
            updateLabel();}};
```

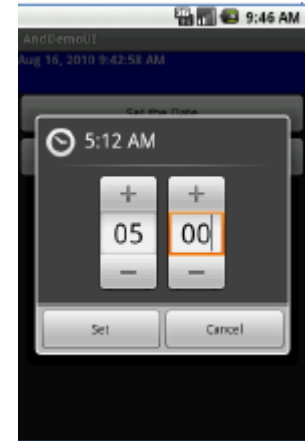
# 1. Pickers Widgets

```
DatePickerDialog.OnDateSetListener d = new
DatePickerDialog.OnDateSetListener()
{
    public void onDateSet(DatePicker view, int year, int
    monthOfYear, int dayOfMonth) {
        myCalendar.set(Calendar.YEAR, year);
        myCalendar.set(Calendar.MONTH, monthOfYear);
        myCalendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
        updateLabel();
    }
};
private void updateLabel() {
    lblDateAndTime.setText(fmtDateAndTime.format
                           (myCalendar.getTime()));
}
```



# 1. Pickers Widgets

```
@Override
public void onCreate(Bundle icle) {
    super.onCreate(icle);
    setContentView(R.layout.main);
    lblDateAndTime = (TextView) findViewById(R.id.lblDateAndTime);
    Button btnDate = (Button) findViewById(R.id.btnDate);
    btnDate.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            new DatePickerDialog(AndDemoUI.this, d, myCalendar.get(Calendar.YEAR),
                                myCalendar.get(Calendar.MONTH),
                                myCalendar.get(Calendar.DAY_OF_MONTH)).show();
        }
    });
    Button btnTime = (Button) findViewById(R.id.btnTime);
    btnTime.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            new TimePickerDialog(AndDemoUI.this, t,
                                myCalendar.get(Calendar.HOUR_OF_DAY),
                                myCalendar.get(Calendar.MINUTE), true).show();
        }
    });
    updateLabel();
} // onCreate
} // class
```

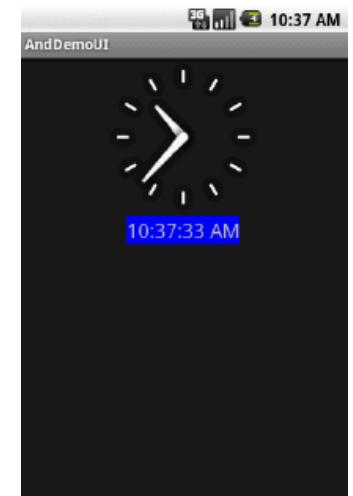


# 1. Pickers Widgets

- **Android cung cấp các widgets về thời gian: DigitalClock và AnalogClock**

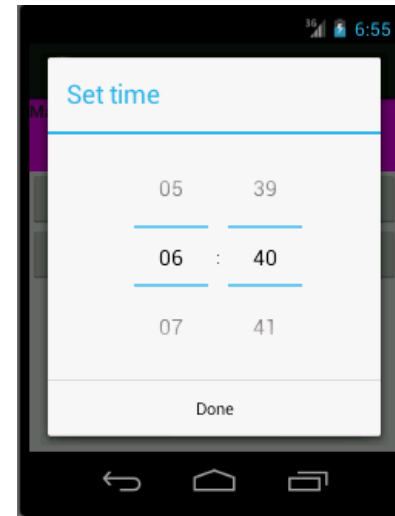
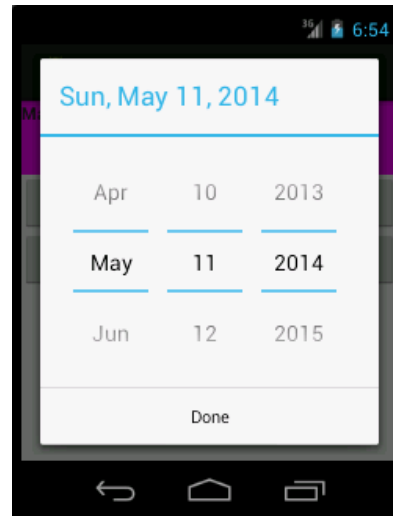
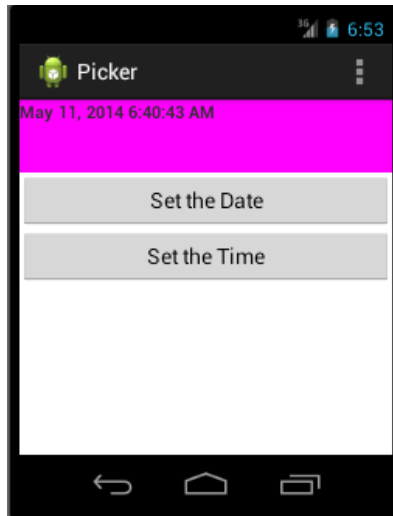
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:id="@+id/widget34"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/a
        pk/res/android">
>
<DigitalClock
    android:id="@+id/digital"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#ff0000ff"
    android:textSize="20px"
    android:layout_below="@+id/analog"
    android:layout_centerHorizontal="true"
>
```

```
</DigitalClock>
<AnalogClock
    android:id="@+id/analog"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
>
</AnalogClock>
</RelativeLayout>
```



# 1. Pickers Widgets

- Demo → Picker



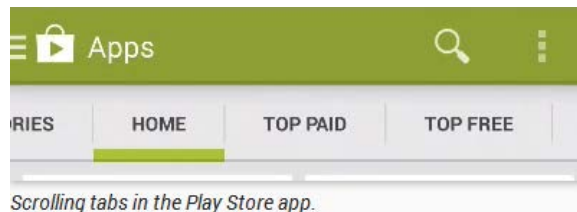


## 2. Tab Selector

- Tab Selector giống như Property Sheet trong Visual C, giống như Tab Control C#
- Đối với Android, mỗi Tab nên thiết kế trên một Layout khác nhau, trong Main Layout include các tab đó vào (tức là nếu như ứng dụng có 2 Tab con thì sẽ tạo 2 Layout khác nhau rồi include chúng vào Main layout)

- Có hai loại Tab:

- Scrollable Tabs:

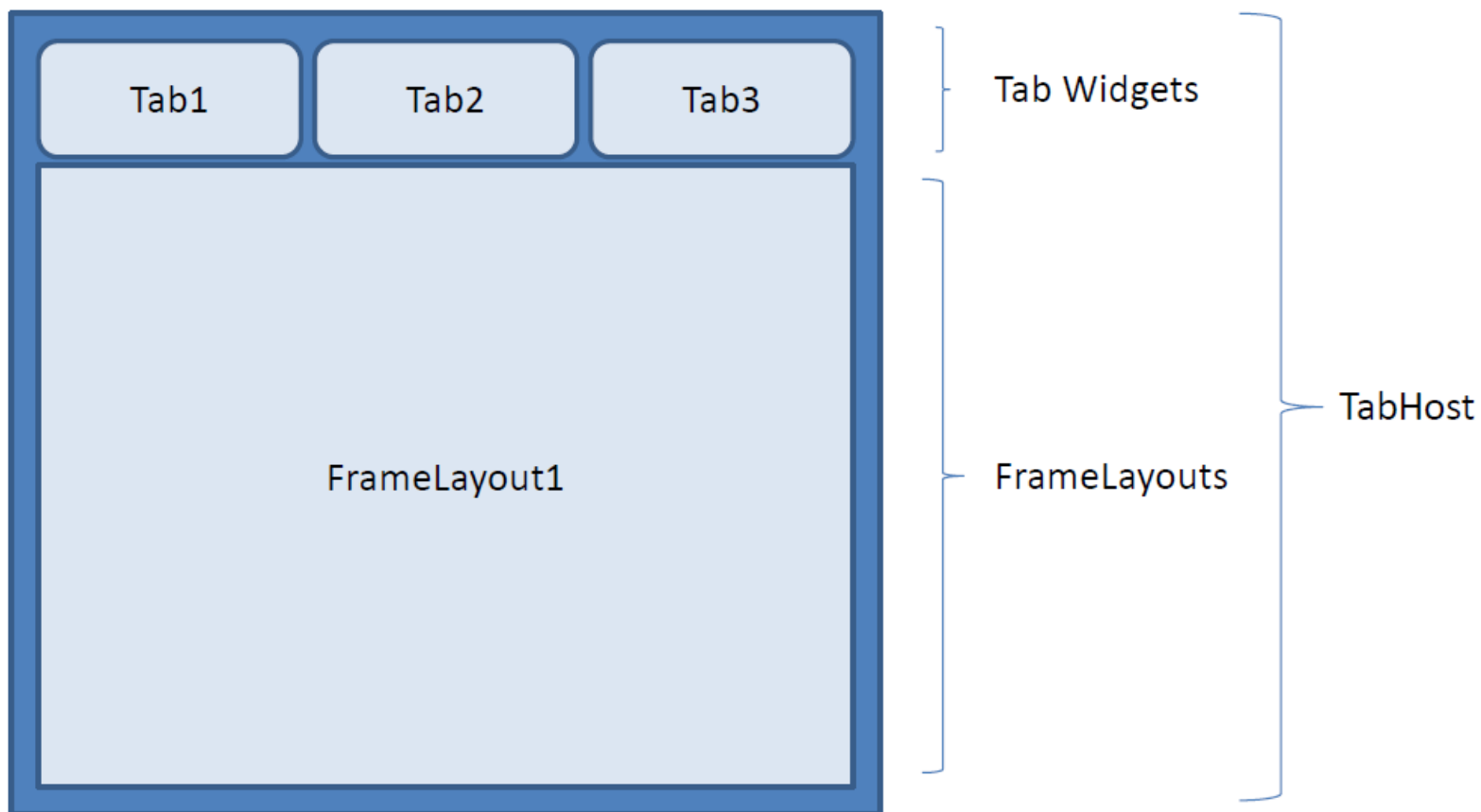


- Fixed Tabs:



## 2. Tab Selector

- Các thành phần của Tab Selector gồm:



## 2. Tab Selector

Các thành phần của Tab Selector gồm:

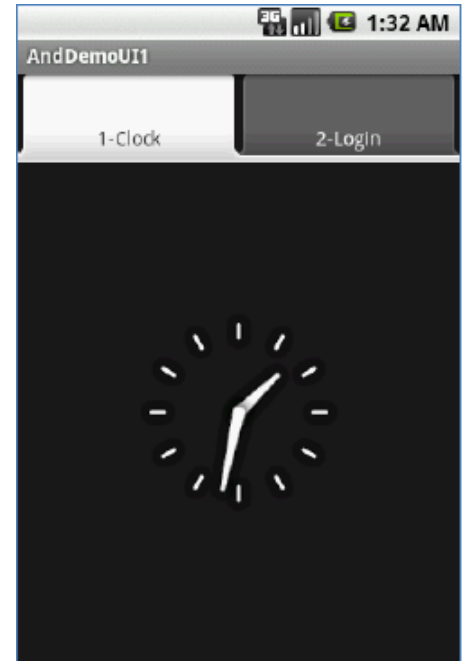
- TabHost: Là Container chính chứa các Tab buttons và Tab contents
- TabWidget: Để định dạng cho các Tab buttons : Nhãn, Icon...
- FrameLayout: là Container để chứa các layout cho Tab contents, ta chỉ có thể dùng FrameLayout cho Tab contents, không thể dùng các loại Layout khác.

## 2. Tab Selector

### ● Ví dụ sử dụng Tabs:

```
<?xml version="1.0" encoding="utf8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TabHost android:id="@+id/tabhost"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"/>
        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:paddingTop="62px">
            <AnalogClock android:id="@+id/tab1"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_centerHorizontal="true"/>
```

```
<FrameLayout
    android:id="@+id/tab2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <include layout="@layout/screen2" />
</FrameLayout>
</FrameLayout>
</TabHost>
</LinearLayout>
```

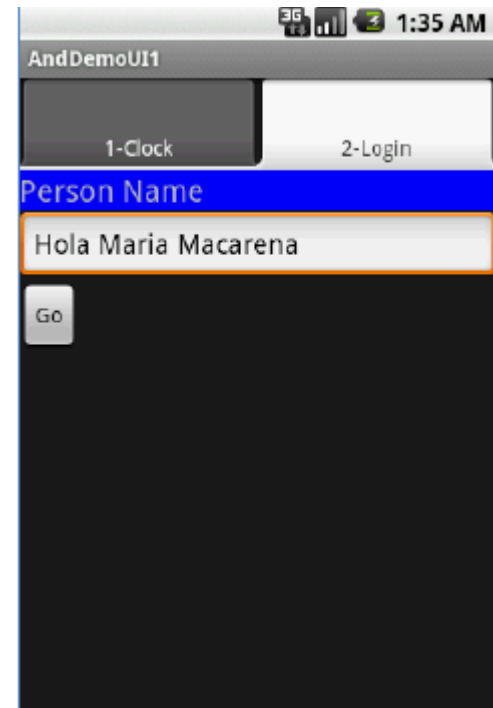


## 2. Tab Selector

### /res/layout/screen2.xml

```
<LinearLayout
    android:id="@+id/tab2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com
/apk/res/android">
    <TextView
        android:id="@+id/caption1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#ff0000ff"
        android:text="Person Name"
        android:textSize="20px">
    </TextView>
    <EditText
        android:id="@+id/txtPerson"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="txtPerson"
        android:textSize="18sp">
    </EditText>
```

```
<Button
    android:id="@+id/btnGo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Go">
</Button>
</LinearLayout>
```

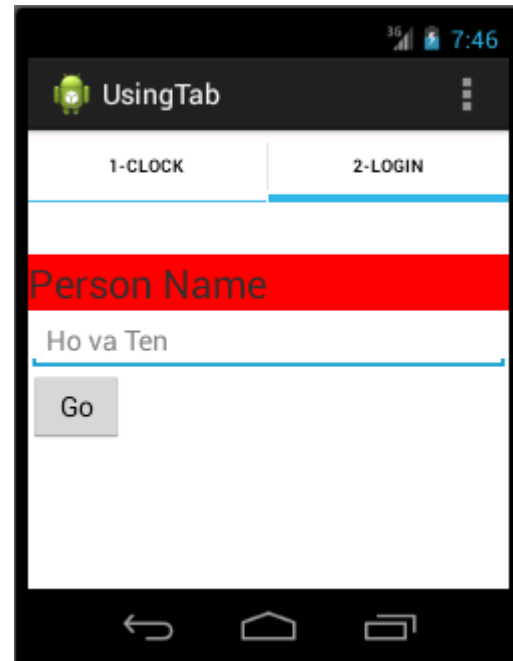
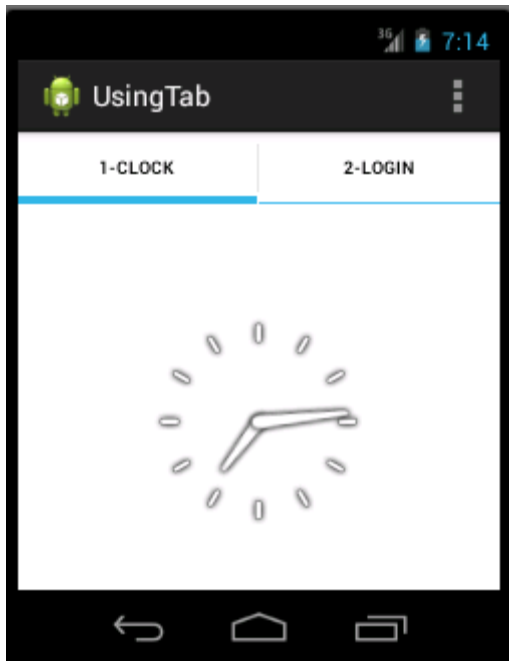


## 2. Tab Selector

```
Packageexample.selectionwidgets;
importandroid.app.Activity;
importandroid.os.Bundle;
importandroid.view.View;
importandroid.view.View.OnClickListener;
importandroid.widget.Button;
importandroid.widget.EditText;
importandroid.widget.TabHost;
publicclassAndDemoUI1 extendsActivity {
    @Override
    publicvoidonCreate(Bundle icle) {
        super.onCreate(icle);
        setContentView(R.layout.main);
        TabHost tabs=(TabHost)findViewById(R.id.tabhost);
        tabs.setup();
        TabHost.TabSpecspec;
        spec =tabs.newTabSpec("tag1");
        spec.setContent(R.id.tab1);
        spec.setIndicator("1-Clock");
        tabs.addTab(spec);
        spec=tabs.newTabSpec("tag2");
        spec.setContent(R.id.tab2);
        spec.setIndicator("2-Login");
        tabs.addTab(spec);
        tabs.setCurrentTab(0);
        Button btnGo=(Button)findViewById(R.id.btnGo);
        btnGo.setOnClickListener(newOnClickListener() {
            @Override
            publicvoidonClick(View arg0) {
                EditTexttxtPerson=
                (EditText)findViewById(R.id.txtPerson);
                String theUser= txtPerson.getText().toString();
                txtPerson.setText("Hello"+ theUser);
            }
        });
    }
}
```

## 2. Tab Selector

- Demo → Tab Selector



### 3. SlidingDrawer

- SlidingDrawer chứa nội dung ẩn trong màn hình, cho phép người dùng kéo điều khiển để đưa nội dung ra màn hình.
- SlidingDrawer có thể đặt theo chiều dọc hoặc chiều ngang.
- SlidingDrawer chỉ được sử dụng bên trong FrameLayout hoặc RelativeLayout
- Kích thước của SlidingDrawer được xác định bằng không gian nội dung chiếm khi SlidingDrawer trượt ra, vì vậy khi thiết kế ở cả hai chiều đều sử dụng thuộc tính *fill\_parent*



### 3. SlidingDrawer

- Ví dụ: SlidingDrawer được sử dụng trên giao diện Andoid, trượt ra để truy cập các ứng dụng như hình dưới:



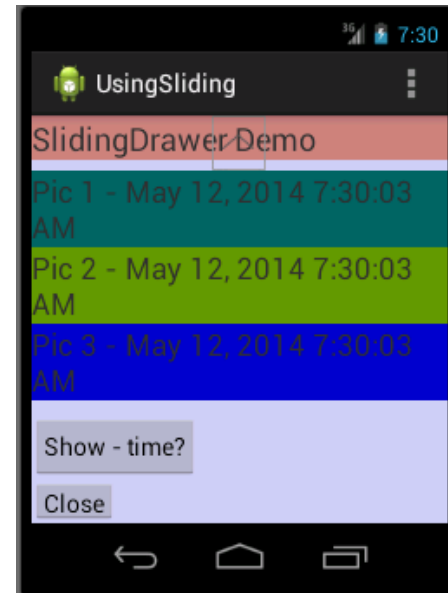
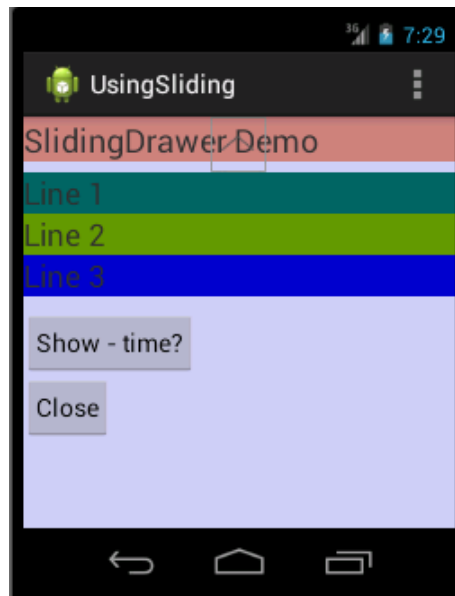
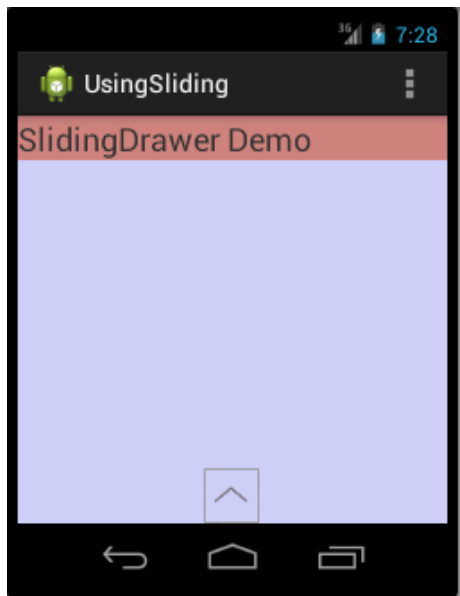
### 3. SlidingDrawer

- Trong XML layout, SlidingDrawer phải xác định id của **handle** và **content**: trong đó handle có thể là ảnh nhỏ để đóng và mở ứng dụng; content chứa nội dung, các control khác.

```
<SlidingDrawer
    android:id="@+id/drawer"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:handle="@+id/handle"
    android:content="@+id/content">
    <ImageView
        android:id="@id/handle"
        android:layout_width="88dip"
        android:layout_height="44dip" />
    <GridView android:id="@id/content"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
</SlidingDrawer>
```

### 3. SlidingDrawer

- Ví dụ: Thẻ TextView màu đỏ nằm trên cùng với UI chính, khi SlidingDrawer trượt ra, một màn hình mới hiển thị: 3 textview khác và 2 button



### 3. SlidingDrawer

- SlidingDrawer XML layout:

```
<?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:background="#FF4444CC >
```

```
    <TextView android:id="@+id/label0"
```

```
        android:layout_alignParentTop="true"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#ffcc3300"
        android:text="SlidingDrawer Demo"
        android:textSize="24sp" />
```

```
    <SlidingDrawer android:id="@+id/drawer"
```

```
        android:layout_alignParentBottom="true"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
```

```
        android:handle="@+id/handle"
```

```
        android:content="@+id/content" >
```

```
    <ImageView
```

```
        android:id="@+id/handle"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/tray_handle_normal"
    />
```

```
    <LinearLayout
```

```
        android:id="@+id/content"
```

```
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >
```

```
        <TextView
```

```
            android:id="@+id/label1"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:background="#ff006666"
            android:text="Line 1" android:textSize="22sp"
        />
```

### 3. SlidingDrawer

#### <TextView

```
android:id="@+id/label2"  
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:background="#ff669900"  
android:text="Line 2" android:textSize="22sp"/>
```

#### <TextView

```
android:id="@+id/label3"  
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:background="#ff0000cc"  
android:text="Line 3" android:textSize="22sp"/>
```

#### <TextView

```
android:id="@+id/filler1"  
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:textSize="6sp" />
```

#### <Button

```
android:id="@+id/btn1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:padding="4px"  
android:text=" btn1 - time? " />
```

#### <Button

```
android:id="@+id/btn2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:padding="4px"  
android:text=" btn2 - close " />
```

</LinearLayout>

</SlidingDrawer>

</RelativeLayout>

### 3. SlidingDrawer

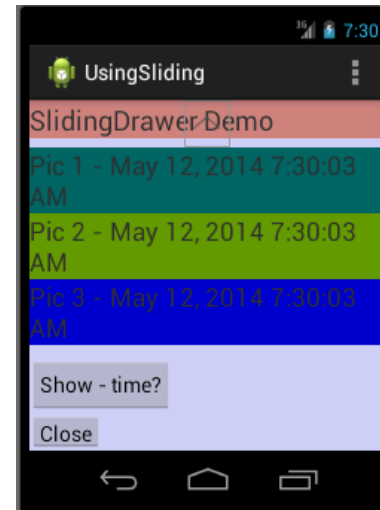
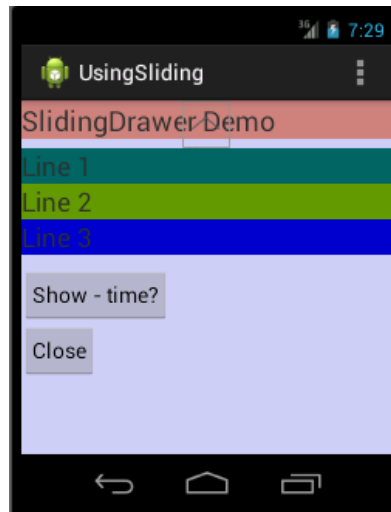
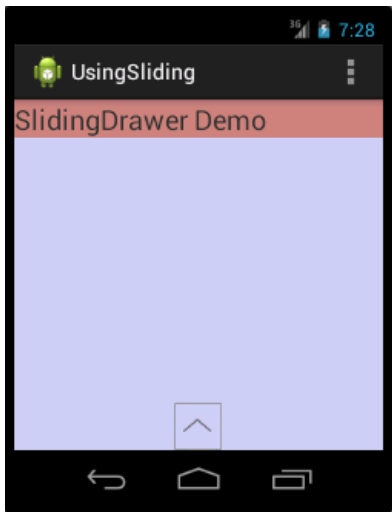
#### ● SlidingDrawer- Android Activity:

```
package cis493.slidingdreawerdemo;
import java.util.Date;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;
public class SlidingDrawerDemo extends Activity
{
    Button btn1; Button btn2; TextView label1;
    TextView label2; TextView label3;
    SlidingDrawer myDrawer;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        myDrawer =
        (SlidingDrawer)findViewById(R.id.drawer);
        btn1 = (Button)findViewById(R.id.btn1);
```

```
        btn2 = (Button)findViewById(R.id.btn2);
        label1 = (TextView)findViewById(R.id.label1);
        label2 = (TextView)findViewById(R.id.label2);
        label3 = (TextView)findViewById(R.id.label3);
        btn1.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                Date dt = new Date();
                String now = dt.toLocaleString();
                label1.setText("111 - Hola amigos " + now);
                label2.setText("222 - Hola amigos " + now);
                label3.setText("333 - Hola amigos " + now);
            }
        });
        btn2.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                myDrawer.animateClose();
            }
        });
    } //onCreate
} // class
```

### 3. SlidingDrawer

- Demo → SlidingDrawer



# Tổng kết bài học

Tìm hiểu cách sử dụng:

- Pickers Widget
- Tabs selector
- SlidingDrawer



**Kết thúc!**

