

Best Practice trong Android

Best Practice là tập hợp những kỹ thuật và cách làm mà đã được nghiên cứu, chứng minh trong thực tế, khi áp dụng nó sẽ làm cho sản phẩm của chúng ta tốt hơn, tránh được rất nhiều các vấn đề mà người khác đã gặp phải.

Best Practice bao gồm các đặc điểm thiết kế có tính tương tác, hiệu suất, bảo mật, tương thích, thử nghiệm, phân phối, ... Chúng được thu gọn và được liệt kê như dưới đây.

Best Practices - User input

Mỗi trường text là dụng ý cho một công việc khác nhau. Ví dụ, một số trường là text và một số trường khác là number. Nếu nó là cho number thì tốt hơn hết là để hiển thị bàn phím số khi trường text đó cần nhập. Cú pháp như sau:

```
<EditText android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/editText"
android:layout_alignParentRight="true" android:layout_alignParentEnd="true"
android:hint="User Name" android:layout_below="@+id/imageView"
android:layout_alignLeft="@+id/imageView"
android:layout_alignStart="@+id/imageView" android:numeric="integer" />
```

Một điều khác nữa là nếu trường của bạn là cho password, thì nó phải hiển thị một mặt khẩu gợi ý để mà người dùng có thể dễ dàng ghi nhớ mật khẩu. Điều này có thể thực hiện như:

```
<EditText android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/editText2"
android:layout_alignLeft="@+id/editText"
android:layout_alignStart="@+id/editText" android:hint="Pass Word"
android:layout_below="@+id/editText"
android:layout_alignRight="@+id/editText"
android:layout_alignEnd="@+id/editText" android:password="true" />
```

Best Practice: Tác vụ ở Background

Có một số tác vụ cụ thể trong một ứng dụng mà đang chạy trong một background. Tác vụ của chúng có thể để lấy một số thứ từ internet, chơi nhạc, ... Nó được đề nghị là các tác vụ chờ đợi trong thời gian dài không nên được thực hiện trong UI thread và thay vào đó là trong Background bằng các Service hoặc AsyncTask.

Service vs AsyncTask

Cả hai được sử dụng để thực hiện các tác vụ ở background, nhưng Service không bị ảnh hưởng bởi hầu hết các sự kiện về vòng đời của giao diện UI, vì thế nó tiếp tục chạy trong các tình huống có thể shut down một AsyncTask.

Best Practice: Hiệu suất

Hiệu suất ứng dụng nên được đánh dấu. Nhưng nó nên là khác nhau không chỉ trên front end, mà còn trên back end khi thiết bị được kết nối với nguồn điện hoặc charging. Charging có thể từ USB và từ Wire cable.

Khi thiết bị tự nạp, cần cập nhật các thiết lập ứng dụng, nếu chưa có, như tối đa hóa refresh rate bất cứ khi nào thiết bị được kết nối. Điều này thực hiện như sau:

```
IntentFilter ifilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED); Intent
batteryStatus = context.registerReceiver(null, ifilter); // Are we charging /
charged? Full or charging. int status =
batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1); // How are we
charging? From AC or USB. int chargePlug =
batteryStatus.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1);
```

Best Practice: Bảo mật và Riêng tư

Việc bảo mật không chỉ với ứng dụng và với dữ liệu người dùng là cực kỳ quan trọng. Tính an toàn có thể được tăng cao bởi các yếu tố sau:

- Sử dụng storage nội bộ thay vì ngoại vi để lưu giữ cho các file ứng dụng.
- Sử dụng Content Provider bất cứ khi nào có thể.
- Sử dụng SSI khi kết nối tới Web.
- Sử dụng permission thích hợp để truy cập các tính năng khác nhau của thiết bị.

Ví

Ví dụ sau minh họa sự sử dụng của một số Best Practice bạn nên dùng khi phát triển các ứng dụng Android. Ví dụ này tạo một ứng dụng cơ bản cho phép bạn xác định cách sử dụng các trường Text và cách tăng hiệu suất bằng việc kiểm tra các trạng thái charging cho điện thoại.

Để thử nghiệm, bạn cần chạy ví dụ trên một thiết bị thực sự.

Sau đây là nội dung của **src/MainActivity.java**

```
package com.example.sairamkrishna.myapplication; import
android.bluetooth.BluetoothAdapter; import android.bluetooth.BluetoothDevice;
import android.content.DialogInterface; import android.content.Intent; import
android.content.IntentFilter; import android.os.BatteryManager; import
android.support.v7.app.ActionBarActivity; import android.os.Bundle; import
android.view.Menu; import android.view.MenuItem; import android.view.View;
import android.widget.AdapterView; import android.widget.Button; import
android.widget.EditText; import android.widget.ListView; import
android.widget.TextView; import android.widget.Toast; import
java.util.ArrayList; import java.util.Set; public class MainActivity extends
ActionBarActivity { EditText ed1,ed2; Button b1; @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
ed1=(EditText)findViewById(R.id.editText);
ed2=(EditText)findViewById(R.id.editText2);
b1=(Button)findViewById(R.id.button); b1.setOnClickListener(new
View.OnClickListener() { @Override public void onClick(View
v) { IntentFilter ifilter = new
IntentFilter(Intent.ACTION_BATTERY_CHANGED); Intent batteryStatus =
registerReceiver(null, ifilter); int status =
batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1); boolean
isCharging = status == BatteryManager.BATTERY_STATUS_CHARGING ||
status == BatteryManager.BATTERY_STATUS_FULL; int
chargePlug = batteryStatus.getIntExtra(BatteryManager.EXTRA_PLUGGED,-1);
boolean usbCharge = chargePlug == BatteryManager.BATTERY_PLUGGED_USB;
boolean acCharge = chargePlug == BatteryManager.BATTERY_PLUGGED_AC;
if(usbCharge){ Toast.makeText(getApplicationContext(),"Mobile is
charging on USB",Toast.LENGTH_LONG).show(); } else
{ Toast.makeText(getApplicationContext(),"Mobile is charging on
AC",Toast.LENGTH_LONG).show(); } } });
@Override protected void onDestroy() { super.onDestroy(); }
@Override public boolean onCreateOptionsMenu(Menu menu) { // Inflate
the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.menu_main, menu); return true; }
@Override public boolean onOptionsItemSelected(MenuItem item) { //
Handle action bar item clicks here. The action bar will // automatically
handle clicks on the Home/Up button, so long // as you specify a parent
activity in AndroidManifest.xml. int id = item.getItemId();
//noinspection SimplifiableIfStatement if (id == R.id.action_settings) {
return true; } return super.onOptionsItemSelected(item); } }
```

Sau đây là nội dung của **activity_main.xml**

```
<RelativeLayout 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:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
```

```
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity"> <TextView android:text="Bluetooth
Example" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/textview"
android:textSize="35dp" android:layout_alignParentTop="true"
android:layout_centerHorizontal="true" /> <TextView
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="Tutorials point" android:id="@+id/textView"
android:layout_below="@+id/textview"
android:layout_centerHorizontal="true" android:textColor="#ff7aff24"
android:textSize="35dp" /> <ImageView
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:id="@+id/imageView" android:src="@drawable/abc"
android:layout_below="@+id/textView"
android:layout_centerHorizontal="true" /> <EditText
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:id="@+id/editText" android:layout_alignParentRight="true"
android:layout_alignParentEnd="true" android:hint="User Name"
android:layout_below="@+id/imageView"
android:layout_alignLeft="@+id/imageView"
android:layout_alignStart="@+id/imageView" android:numeric="integer" />
<EditText android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/editText2"
android:layout_alignLeft="@+id/editText"
android:layout_alignStart="@+id/editText" android:hint="Pass Word"
android:layout_below="@+id/editText"
android:layout_alignRight="@+id/editText"
android:layout_alignEnd="@+id/editText" android:password="true" />
<Button android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Check"
android:id="@+id/button" android:layout_below="@+id/editText2"
android:layout_centerHorizontal="true" /> </RelativeLayout>
```

Sau đây là nội dung của **Strings.xml**

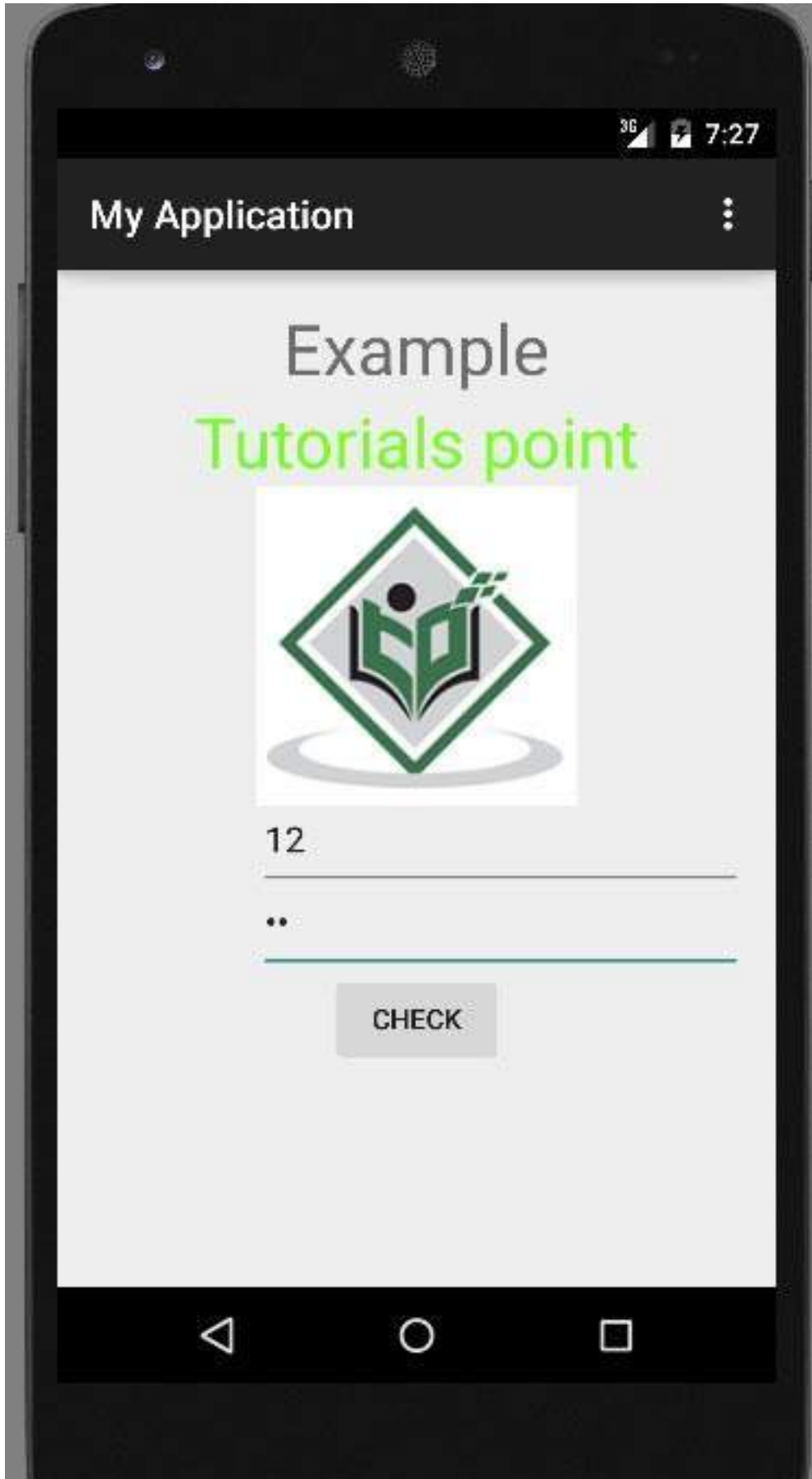
```
<resources> <string name="app_name">My Application</string> <string
name="hello_world">Hello world!</string> <string
name="action_settings">Settings</string> </resources>
```

Sau đây là nội dung của **AndroidManifest.xml**

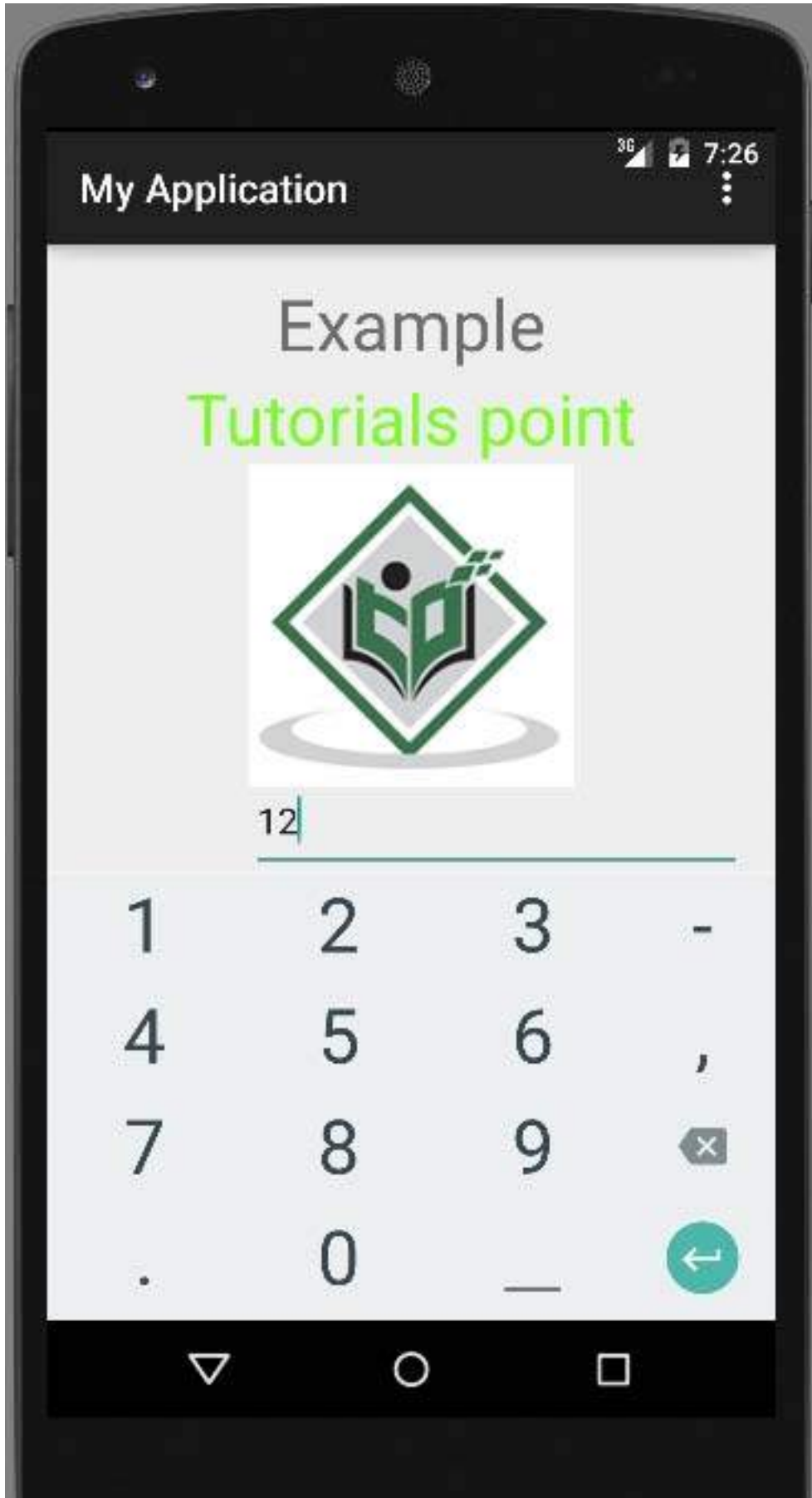
```
<?xml version="1.0" encoding="utf-8"?> <manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.bestpractices" android:versionCode="1"
android:versionName="1.0" > <application android:allowBackup="true"
android:icon="@drawable/ic_launcher" android:label="@string/app_name"
android:theme="@style/AppTheme" > <activity
android:name="com.example.bestpractices.MainActivity"
android:label="@string/app_name" > <intent-filter>
<action android:name="android.intent.action.MAIN" /> <category
```

```
android:name="android.intent.category.LAUNCHER" />          </intent-filter>  
</activity>          </application> </manifest>
```

Cuối cùng, bạn chạy ứng dụng Android vừa sửa đổi ở trên.



Hình ảnh trên hiển thị một output của ứng dụng.



Bây giờ gõ trên trường username và bạn sẽ thấy các gợi ý đã được xây dựng sẵn trong Android từ từ điển.



Trong trường password, nó sẽ biến mất ngay sau khi bạn nhập text vào trường này.

