



GIỚI THIỆU

Anh Lê Training hoạt động trong lĩnh vực đào tạo, hiện tại chúng tôi đang triển khai các chương trình đào tạo trong lĩnh vực Marketing, Sale, Công nghệ thông tin, Đồ họa máy tính, Quản trị doanh nghiệp, Kỹ năng mềm và Ứng dụng công nghệ trực tuyến vào hoạt động doanh nghiệp.

Thời gian qua, Android đã thu hút rất nhiều nhà phát triển từ khắp mọi nơi khi Google phân phối miễn phí Android và các đoạn mã nguồn mở cho bất cứ ai muốn tham gia phát triển phần mềm cho Android. Vì thế, nếu nắm trong tay những kiến thức cơ bản về hệ điều hành này, bạn có thể dễ dàng đưa ra các phần mềm cho Android và có thể tùy chỉnh theo ý riêng của mình.

Khóa học lập trình ứng dụng Android tại Anh Lê Training chính là nơi đem lại cho bạn những công cụ cần thiết, những kiến thức cơ bản, giúp bạn tiến gần hơn tới vị trí của một lập trình viên, một nhà phát triển chuyên nghiệp, hoặc chỉ đơn giản là thỏa sức sáng tạo ra những ứng dụng hữu ích cho cuộc sống.

Sau khóa học “Lập trình ứng dụng Android” bạn sẽ thành thạo các công việc sau:

- Tự thiết lập môi trường để phát triển ứng dụng cho Android.
- Áp dụng được quy trình phát triển ứng dụng cho điện thoại Android.
- Tự phát triển ứng dụng Android hoàn chỉnh, theo mô hình hướng đối tượng, tích hợp các module của Google, của nhà phát triển thứ 3 vào ứng dụng của bạn.
- Hiểu biết sâu sắc về Android và những công nghệ mới của Google.
- Tự phát triển ứng dụng Android hoàn chỉnh tích hợp các module của Google vào ứng dụng, thành thạo trong việc sử dụng web services.

Giáo trình “**Lập trình ứng dụng Adnroid**” được viết cho các học viên của Anh Lê Training. Trong quá trình biên soạn không tránh khỏi những khiếm khuyết, chúng tôi rất mong và xin cảm ơn các đồng nghiệp, các bạn có những đóng góp để cuốn sách ngày càng hoàn thiện.

ANH LÊ TRAINING



MỤC LỤC

TỔNG QUAN VỀ ANDROID.....	3
CÀI ĐẶT CÔNG CỤ LẬP TRÌNH ANDROID	7
TEXT FIELDS & FORM WIDGETS	19
MATH-DATE-REGULAREXPRESSION-STRING	24
BẢNG SỐ XE	27
SPINNER	31
CHECKBOX.....	36
RADIOBUTTON	41
PASSWORD	46
TOGGLEBUTTON.....	50
ALERTDIALOG.....	54
GRIDVIEW.....	56
RATINGBAR.....	59
LAYOUT	63
MENU	76
CHỌN VÀ CHƠI NHẠC MP3	85
GOOGLE MAP	92
ANALOGCLOCK & DIGITALCLOCK.....	98
DATE PICKER	100
TOAST	105
PROGRESSBAR.....	108
ACTIVITY	114
SQlite	130
CÀI ĐẶT & QUẢN TRỊ SQlite TRONG FIREFOX.....	145
SEND EMAIL	150



TỔNG QUAN VỀ ANDROID

1. Android là gì ?

Android là một hệ điều hành dành cho thiết bị di động như điện thoại, máy tính bảng và netbooks. Android được phát triển bởi Google, dựa trên nền tảng Linux kernel và các phần mềm nguồn mở. Ban đầu nó được phát triển bởi Android Inc (sau đó được Google mua lại) và gần đây nó trở thành một trong những phần mềm đứng đầu của liên minh OHA (Open Handset Alliance - với khoảng 78 thành viên bao gồm cả nhà sản xuất, nhà phát triển ứng dụng... cho thiết bị di động mà dẫn đầu là Google). Andorid được phát triển nhằm cạnh tranh với các hệ điều hành di động khác như iOS (Apple), BlackBerry OS, Windows Mobile (Microsoft), Symbian (Nokia), Samsung (Bada), WebOS (Palm)... Theo thống kê trong quý II năm 2010 tại Mỹ, hệ điều hành Android chiếm thị phần 33% (cao nhất) trong tổng số các hệ điều hành di động được bán ra, thứ 2 là BlackBerry OS 28% và iOS (Apple) xếp thứ 3 với 22%.

Android có một cộng đồng phát triển ứng dụng rất lớn, hiện có khoảng hơn 70.000 ứng dụng có sẵn cho Android và đang liên tục được cập nhật. Ứng dụng được phát triển bằng ngôn ngữ Java kết hợp với thư viện Java có sẵn của Google. Các nhà phát triển ứng dụng có thể sử dụng máy tính chạy hệ điều hành Windows hoặc MacOS hoặc Linux kết hợp với Android SDK để phát triển ứng dụng cho Android. Hệ điều hành Android bao gồm 12.000.000 dòng mã trong đó có 3.000.000 dòng XML, 2.800.000 dòng C, 2.100.000 dòng Java, và 1.750.000 dòng C++.

Để phát triển phần mềm trên Android, các lập trình viên có thể tải về bộ công cụ phát triển (Android SDK). Bộ công cụ bao gồm các công cụ và các API cần thiết, sử dụng ngôn ngữ Java để lập trình.

2. Các chức năng chính của Android.

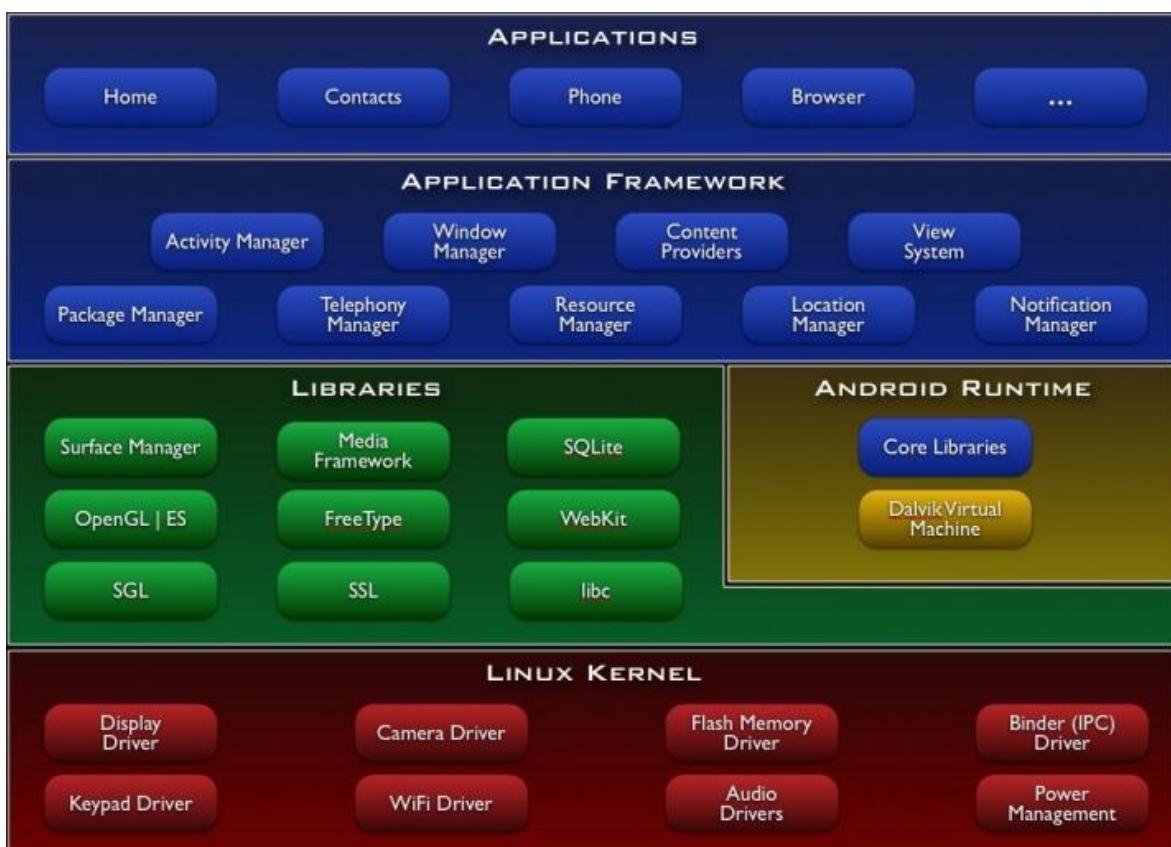
Các chức năng mà Android cung cấp:

- Android cung cấp framework ứng dụng cho phép việc tái sử dụng và thay thế mã nguồn ở dạng component một cách dễ dàng.
- Cung cấp máy ảo Dalvik được tối ưu cho các thiết bị di động.
- Trình duyệt Web dựa trên engine mã nguồn mở Webkit.



- Các tính năng đồ họa được tối ưu bởi một thư viện đồ họa 2D bên dưới; đối với đồ họa 3D, Android sử dụng thư viện OpenGL ES 1.0 nếu thiết bị có hỗ trợ.
- Sử dụng SQLite để lưu trữ dữ liệu có cấu trúc.
- Hỗ trợ các định dạng hình ảnh, âm thanh, video phổ biến như MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF.
- Hỗ trợ băng tần GSM (tùy vào phần cứng thiết bị).
- Hỗ trợ Bluetooth, EDGE, 3G, WiFi (tùy vào phần cứng thiết bị).
- Ngoài ra còn có khả năng của các thiết bị như máy chụp ảnh, thiết bị định vị toàn cầu, la bàn, và bộ cảm biến gia tốc.
- Cung cấp môi trường phát triển phần mềm đầy đủ, bao gồm phần mềm giả lập thiết bị, các công cụ gỡ rối (debugging), theo dõi bộ nhớ và năng suất hoạt động, plugin cho Eclipse IDE.

3. Kiến trúc của Android



Tầng ứng dụng – Applications:

Tầng ứng dụng của Android bao gồm các ứng dụng cốt lõi như:

- Email client
- Chương trình SMS



- Lịch
- Bản đồ
- Trình duyệt
- Danh bạ
- Các ứng dụng khác.

Tầng Application Framework:

Ở tầng này, các nhà phát triển ứng dụng có thể truy cập vào phần cứng thiết bị, thông tin định vị vị trí, chạy các dịch vụ nền, đặt các cảnh báo, thông báo vào thanh trạng thái, v.v.... và quan trọng nhất, đó là các API của framework.

Phía dưới tất cả các ứng dụng là một tập các dịch vụ hệ thống bao gồm:

- Một tập các đối tượng View có thể được mở rộng để xây dựng một ứng dụng, gồm có List, Grid, TextBox, Button, và WebBrowser.
- Các đối tượng ContentProvider cho phép các ứng dụng truy cập vào dữ liệu của các ứng dụng khác (chẳng hạn như truy cập danh bạ), hoặc để chia sẻ dữ liệu với nhau.
- Một trình quản lý tài nguyên, cho phép truy xuất các tài nguyên không phải mã nguồn như các chuỗi đã được bản địa hóa, các tập tin đồ họa và giao diện.
- Trình quản lý thông báo cho phép tất cả các ứng dụng hiển thị các cảnh báo lên thanh trạng thái.
- Trình quản lý các đối tượng Activity dùng để quản lý vòng đời của các ứng dụng cung cấp các chức năng điều hướng.

Tầng Libraries – Runtime:

Tầng này cung cấp các thư viện media dựa trên thư viện PacketVideo's OpenCORE; các thư viện này hỗ trợ khả năng playback và thu lại nhiều định dạng âm thanh, hình ảnh thông dụng như MPEG4, H.264, MP3, AAC, AMR, JPG, PNG.

Kèm theo đó là thư viện SQLite, một hệ quản trị cơ sở dữ liệu nhỏ nhẹ và mạnh mẽ được cung cấp cho tất cả các ứng dụng.

Ở Runtime, Android cung cấp máy ảo Dalvik dùng để thực thi các file định dạng Dalvik Executable (.dex) đã được tối ưu hóa cho các thiết bị có bộ nhớ nhỏ. Máy ảo Dalvik chỉ chạy các class đã được đăng ký và được biên dịch bởi một trình biên dịch Java đi kèm theo bộ SDK (dx tool). Ngoài ra Dalvik còn sử dụng nhân Linux để quản lý các tính năng ở mức thấp



và các tác vụ chạy theo luồng. Mọi ứng dụng Android chạy trên một tiến trình riêng cùng với một instance riêng của máy ảo Dalvik. Dalvik đã được tối ưu sao cho một thiết bị có thể chạy nhiều mày ảo cùng lúc một cách hiệu quả.

Tầng Linux kernel:

Android được phát triển dựa trên các dịch vụ hệ thống cốt lõi của Linux phiên bản 2.6, bao gồm các module:

- Security
- Memory management
- Process management
- Network stack
- Driver model

Tầng kernel hoạt động như một lớp trùu tượng giữa lớp phần cứng và phần mềm.

4. Cấu trúc một dự án Android

Khi tạo mới một dự án Android, bạn cần lưu ý tới các mục sau:

- AndroidManifest.xml, là một file XML miêu tả ứng dụng được xây dựng và các thành phần kèm theo như các Activities, các Services được cung cấp bởi ứng dụng đó.
- Thư mục libs/ chứa các thư viện Java ở dạng JAR của các hãng thứ 3 mà ứng dụng cần để chạy.
- Thư mục src/ chứa mã nguồn Java cho ứng dụng.
- Thư mục res/ chứa các tài nguyên chẵng hạn như các biểu tượng, giao diện, ... được đóng gói kèm theo khi biên dịch ứng dụng.
- Thư mục assets/ chứa các file tĩnh mà bạn muốn cài đặt lên hệ thống



CÀI ĐẶT CÔNG CỤ LẬP TRÌNH ANDROID

1. Cài đặt JDK.

JDK (Java Development Kit) bao gồm JRE (Java Runtime Environment) nó là bộ thư viện nền tảng để chúng ta có thể lập trình java.

- Để download JDK truy cập vào địa chỉ:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Click vào hình của mục Java Platform (JDK) như sau:



- Tiếp theo, Click vào radiobutton “Accept License Agreement”. Chọn vào file .exe của mục Windows x86(32-bit). (Hoặc Windows x86(64-bit)).



Java SE Development Kit 7u4

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux x86 (32-bit)	64.00 MB	jdk-7u4-linux-i586.rpm
Linux x86 (32-bit)	79.09 MB	jdk-7u4-linux-i586.tar.gz
Linux x64 (64-bit)	64.93 MB	jdk-7u4-linux-x64.rpm
Linux x64 (64-bit)	77.67 MB	jdk-7u4-linux-x64.tar.gz
Mac OS X (64-bit)	97.28 MB	jdk-7u4-macosx-x64.dmg
Solaris x86 (32-bit)	137.42 MB	jdk-7u4-solaris-i586.tar.Z
Solaris x86 (32-bit)	82.01 MB	jdk-7u4-solaris-i586.tar.gz
Solaris SPARC (32-bit)	140.37 MB	jdk-7u4-solaris-sparc.tar.Z
Solaris SPARC (32-bit)	86.73 MB	jdk-7u4-solaris-sparc.tar.gz
Solaris SPARC (64-bit)	16.45 MB	jdk-7u4-solaris-sparcv9.tar.Z
Solaris SPARC (64-bit)	12.55 MB	jdk-7u4-solaris-sparcv9.tar.gz
Solaris x64 (64-bit)	14.95 MB	jdk-7u4-solaris-x64.tar.Z
Solaris x64 (64-bit)	9.54 MB	jdk-7u4-solaris-x64.tar.gz
Windows x86 (32-bit)	87.97 MB	jdk-7u4-windows-i586.exe
Windows x64 (64-bit)	92.33 MB	jdk-7u4-windows-x64.exe

- Chạy file download về để cài đặt

2. Cài đặt Eclipse.

- Là công cụ soạn thảo, biên dịch và chạy ứng dụng được viết bằng java (trong đó có ứng dụng Android)
- Download, truy cập từ địa chỉ:
<http://www.eclipse.org/downloads/>
- Trên trang web ta chọn download bản Eclipse IDE for Java Developers (128M), (bảng 32bit hay 64bit tùy theo máy của bạn)

The screenshot shows the Eclipse Downloads page at <http://www.eclipse.org/downloads/>. The main heading is "Eclipse Downloads". Below it, there are three tabs: "Packages", "Developer Builds", and "Projects". The "Packages" tab is selected. A dropdown menu shows "Eclipse Indigo (3.7.2) Packages for Windows". Under this, there are two options: "Eclipse IDE for Java EE Developers" (212 MB) and "Eclipse Classic 3.7.2" (174 MB). Both have download counts and "Details" links. At the bottom, there is another option: "Eclipse IDE for Java Developers" (128 MB), which is highlighted with a red box and has download counts. To its right, there are "Windows 32 Bit" and "Windows 64 Bit" download links, each with a green arrow icon.



- Giải nén file mới tải (vd: eclipse-java-indigo-SR2-win32.zip) thành thư mục Eclipse
- Chạy chương trình Eclipse bằng cách nhấp đúp file eclipse.exe trong thư mục vừa Eclipse. (không cần cài đặt).

3. Cài đặt Android SDK

- Android SDK (Android Software Development Kit) là bộ thư viện nền tảng để lập trình Android.
- Truy cập địa chỉ:
<http://developer.android.com/sdk/index.html> và download phiên bản mới nhất về cài đặt

The screenshot shows the 'Download the Android SDK' section of the Android Developers website. On the left, there's a sidebar with links like 'Android SDK Starter Package', 'Download', 'Installing the SDK', 'Downloadable SDK Packages', and 'SDK Tools, r19'. The main content area has a heading 'Download the Android SDK' and a welcome message. Below it is a table with download links for different platforms. A red arrow points to the 'installer_r18-windows.exe (Recommended)' link in the Windows row.

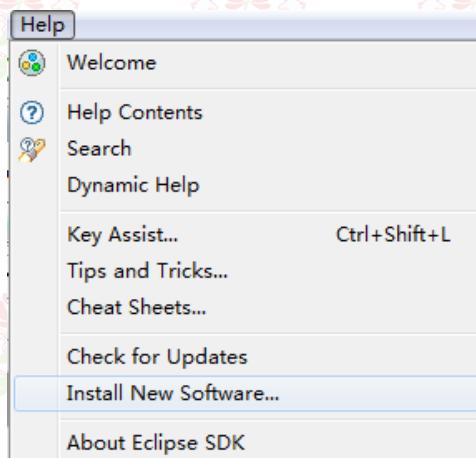
Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r18-windows.zip	37440775 bytes	bfbfdf8
	installer_r18-windows.exe (Recommended)	37416234 bytes	40b1fe
Mac OS X (intel)	android-sdk_r18-macosx.zip	33903758 bytes	8328e8
Linux (i386)	android-sdk_r18-linux.tgz	29731463 bytes	6cd716

- Sau khi cài xong sẽ có thư mục C:\Program Files\Android.

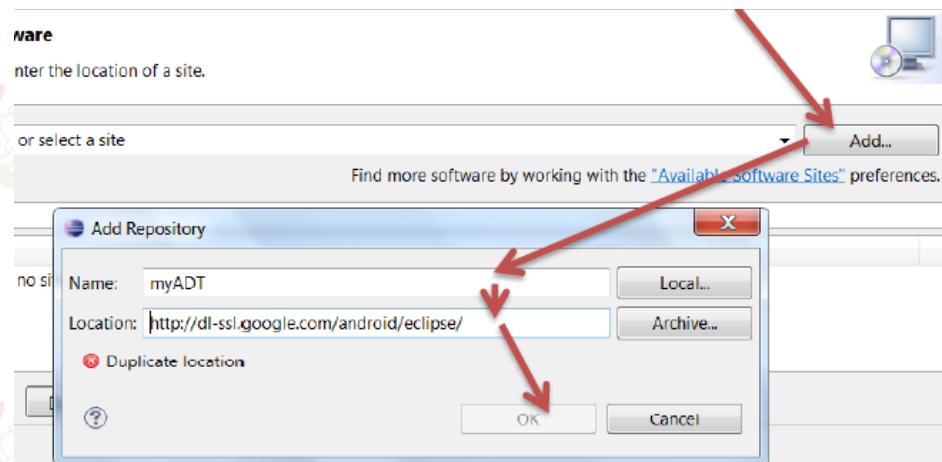
4. Cài đặt ADT

Android Development Tools (ADT) là plugin của Eclipse IDE được thiết kế nhằm cung cấp cho lập trình viên môi trường tích hợp mạnh mẽ để xây dựng các chương trình Android trên Eclipse.

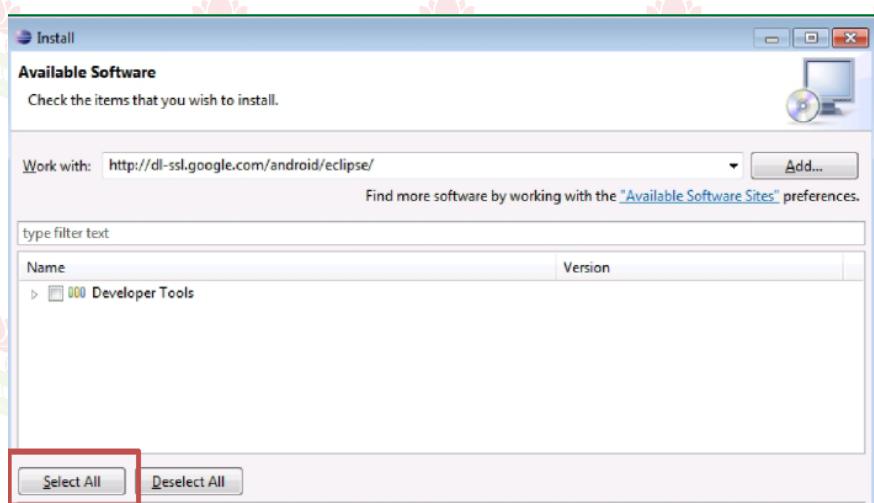
- Mở Eclipse chọn Help -> Install new software.



- Chọn nút “Add” để mở Hộp thoại “Add Repository”.

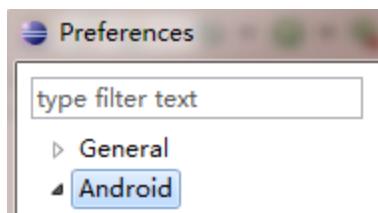
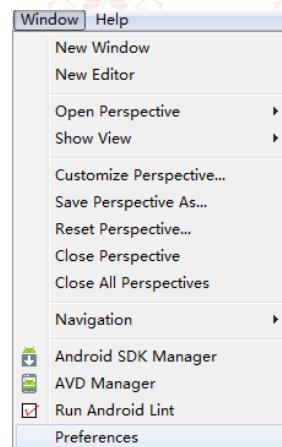


- Trong hộp thoại vừa mở, mục name nhập vào một cái tên. Vd: myADT. Mục Location nhập vào địa chỉ như bên dưới rồi chọn OK
http://dl-ssl.google.com/android/eclipse/
- Quay về Hộp thoại Install. Chọn nút “Select All” sau đó next tới và finish.

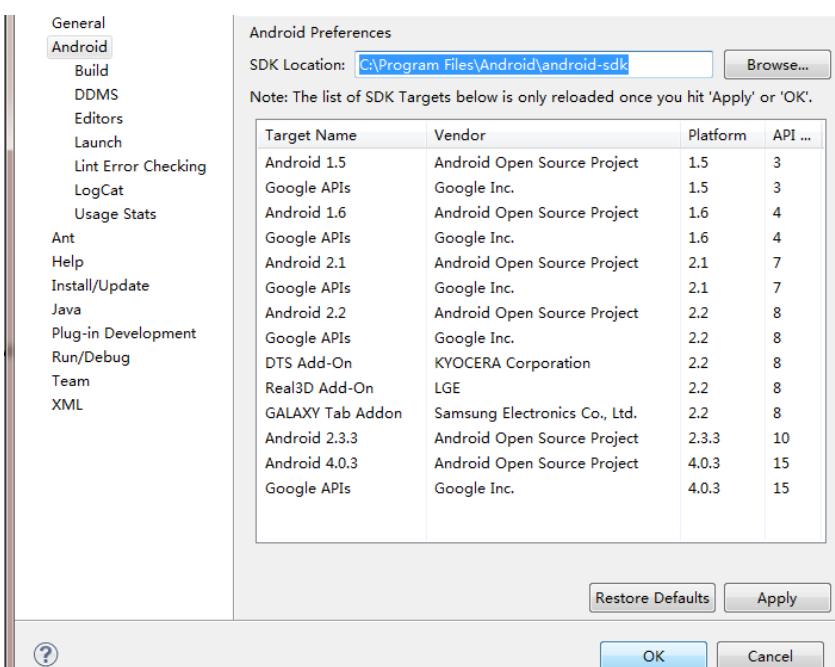


5. Kết nối Eclipse và Android

- Mở Eclipse Chọn Window -> Preferences -> Android

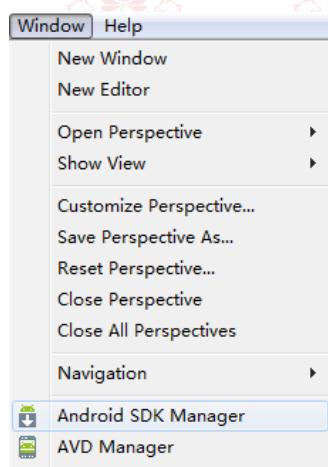


- Tại SDK Location: Chọn mục chứa android_sdk, sau sau đó chọn OK

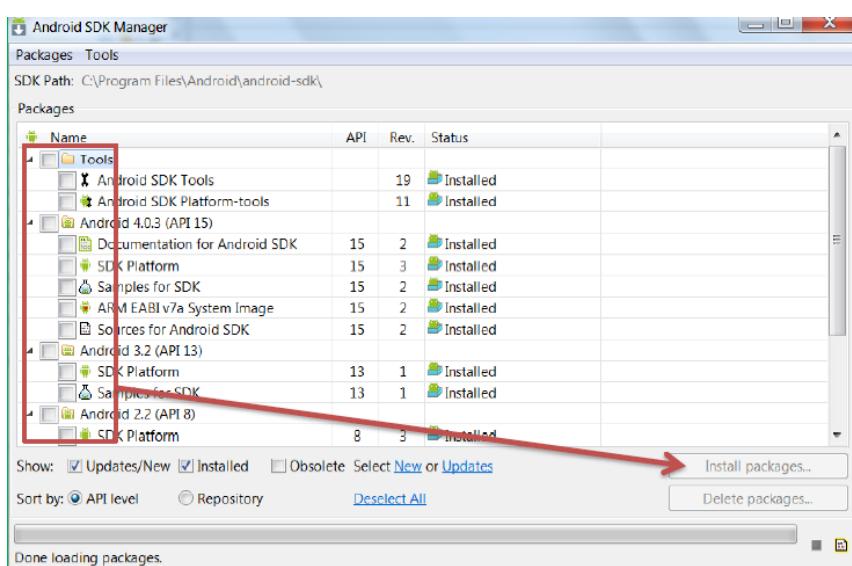


6. Cài các phiên bản Android để lập trình

- Trong Eclipse chọn Window -> Android SDK Manager.



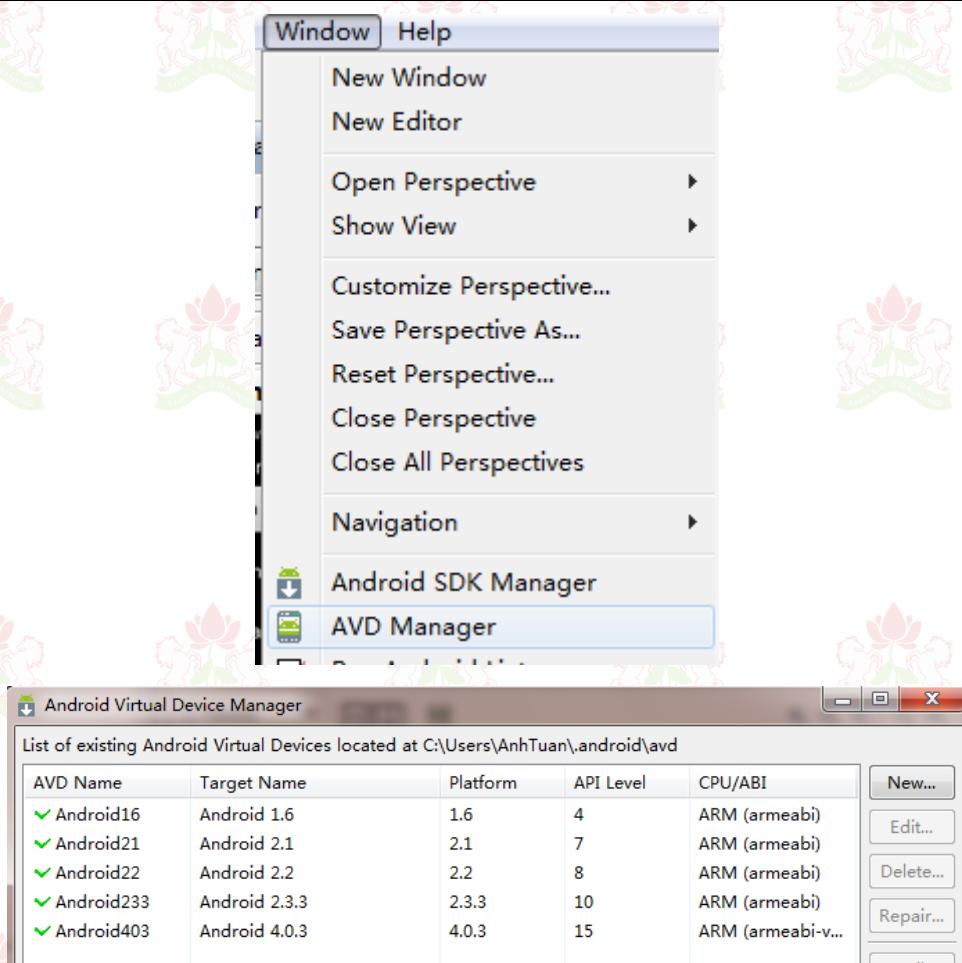
- Trong hộp thoại check các phiên bản muốn cài rồi chọn nút Install packages



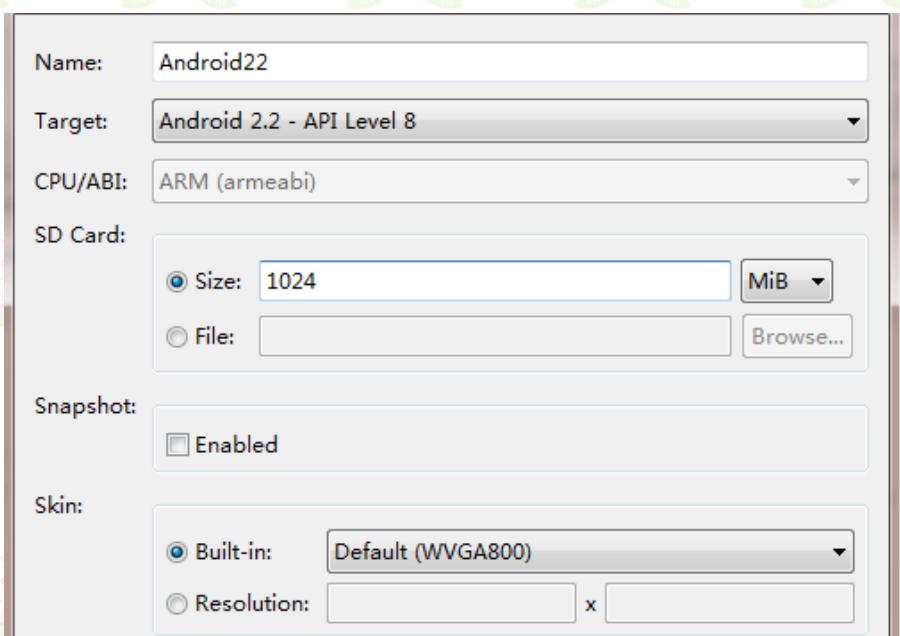
7. Cài đặt AVD

Android Virtual Device (AVD) là các máy ảo android. Cho phép ta chạy một máy giả lập Android trên PC để phục vụ cho việc kiểm tra ứng dụng ta đang viết.

- Trong Eclipse chọn Window -> AVD Manager.

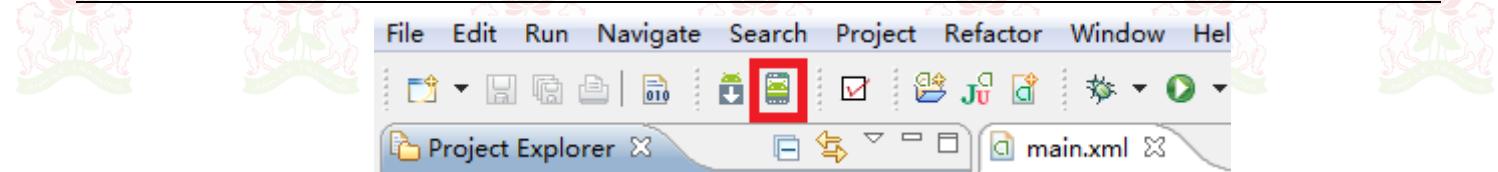


- Chọn nút “New” để tạo một máy ảo mới và nhập vào một số thông tin

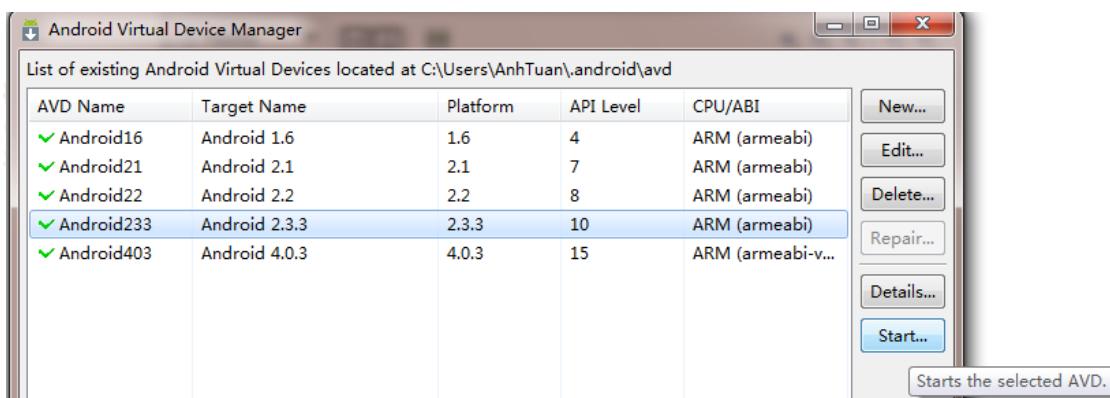


8. Chạy Thủ Máy Ảo

- Trên Eclipse chọn Window -> AVD Manager hoặc chọn biểu tượng trên thanh công cụ để mở AVD Manager.



- Chọn tên máy ảo và chọn nút Start

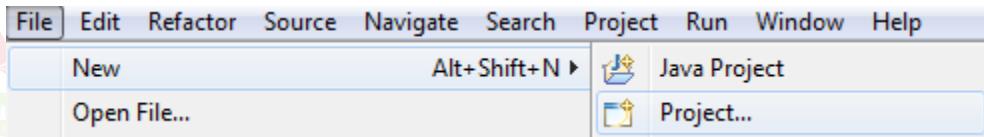




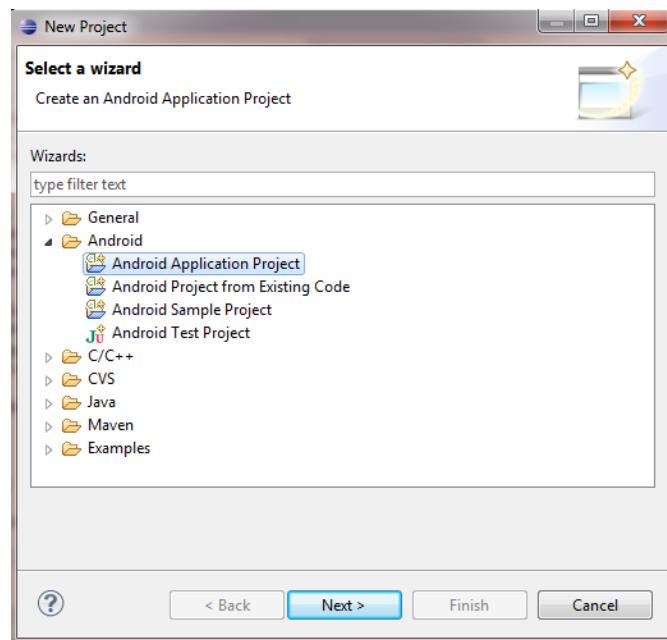
HELLO

1. Tạo project mới

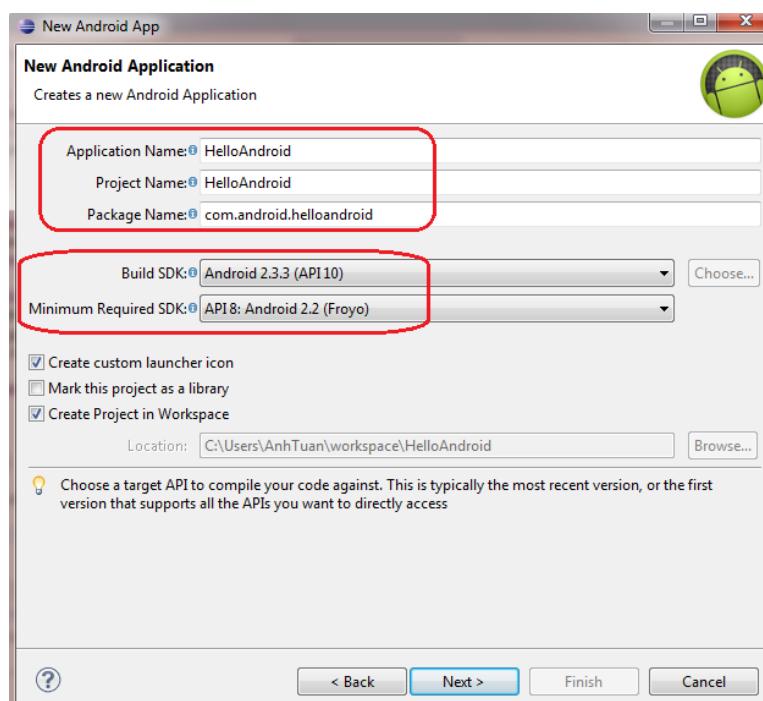
- Chọn File, New, Project ...



- Chọn Android Application Project và chọn Next

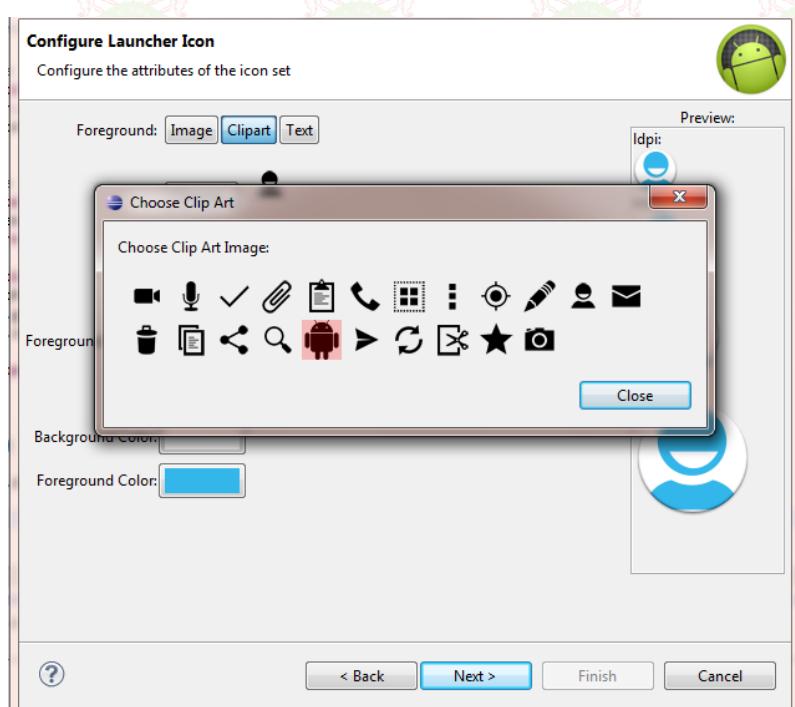


- Điền và chọn một số thuộc tính và chọn Next

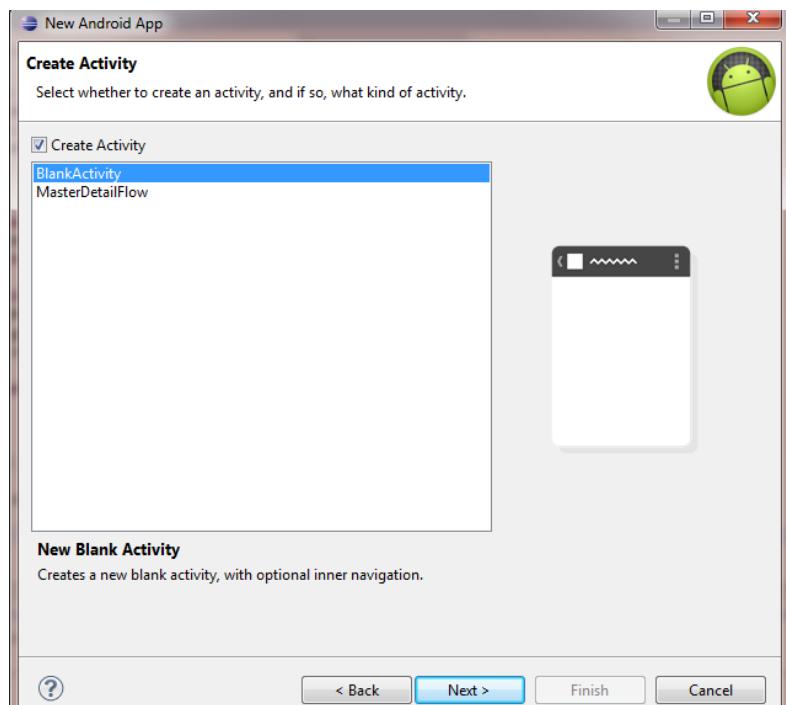




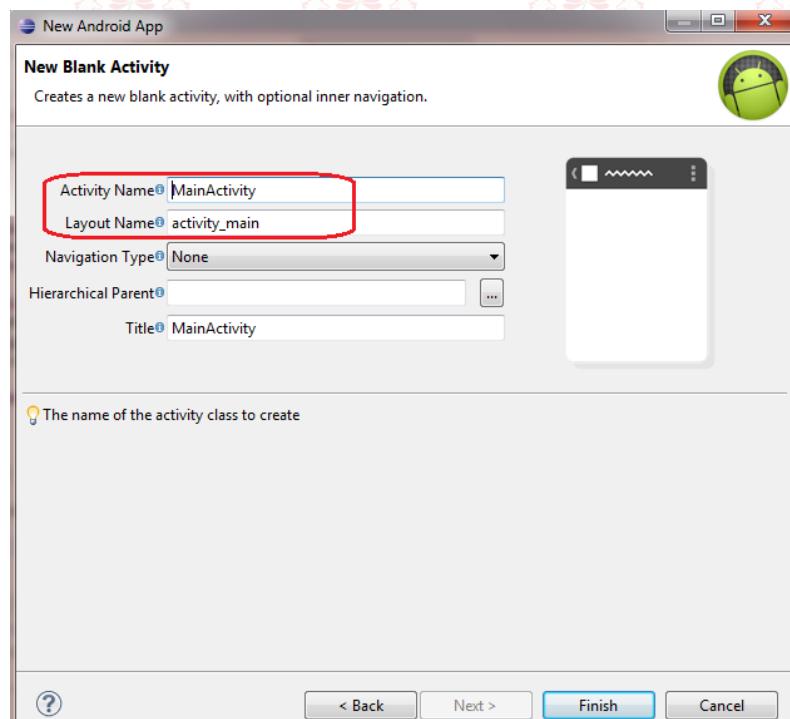
- Chọn icon



- Tạo Activity

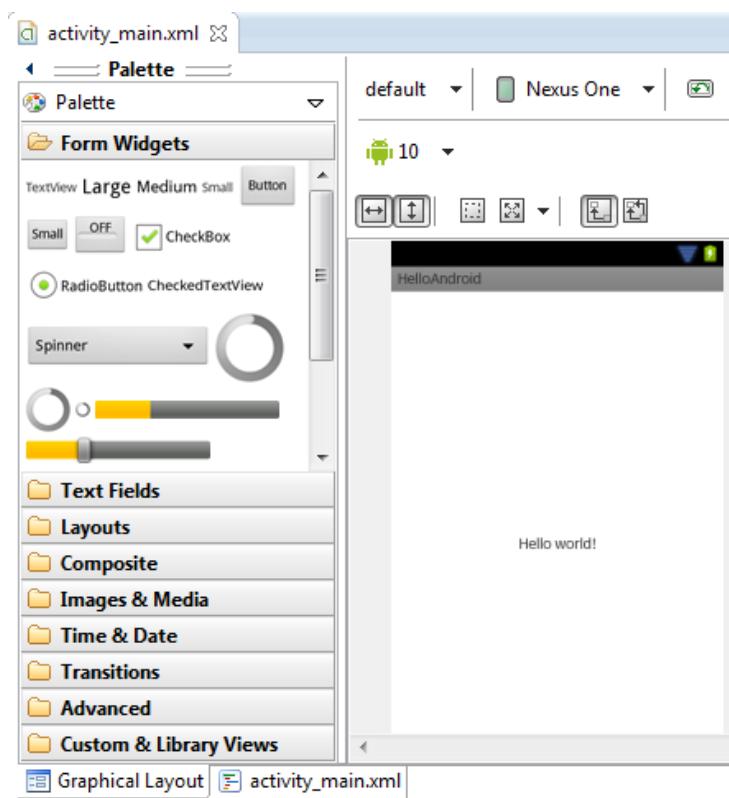


- Đặt tên cho Activity và Layout, chọn finish



2. Cấu trúc thư mục project:

- Layout của project



- và cấu trúc XML của layout



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" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/hello_world"
        tools:context=".MainActivity" />

</RelativeLayout>
```

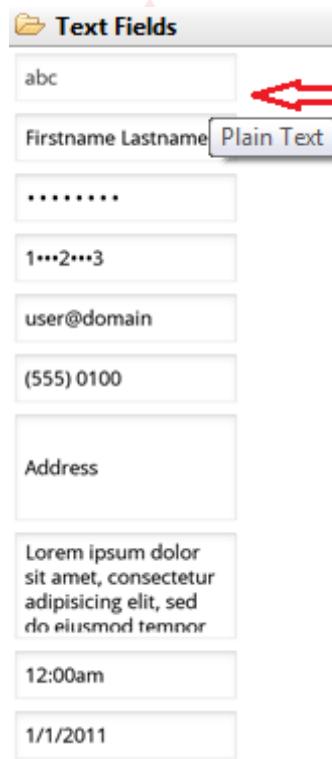


TEXT FIELDS & FORM WIDGETS

1. Tạo project mới

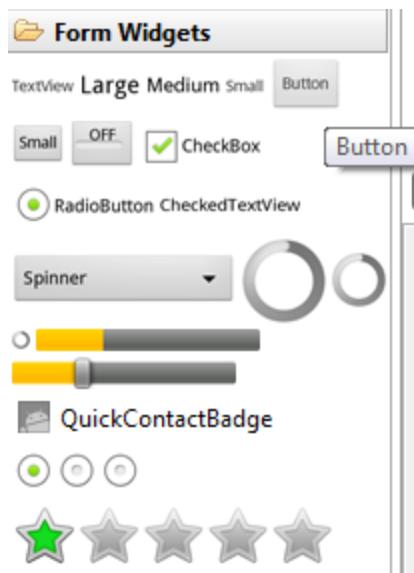
2. Tạo 2 EditText

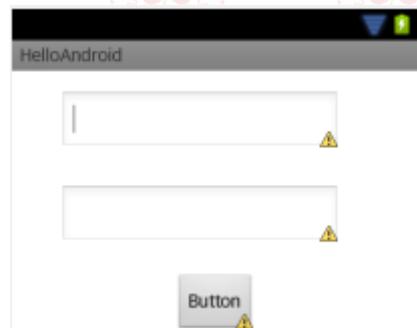
- Chọn editText từ TextFields và kéo sang layout



3. Tạo Button

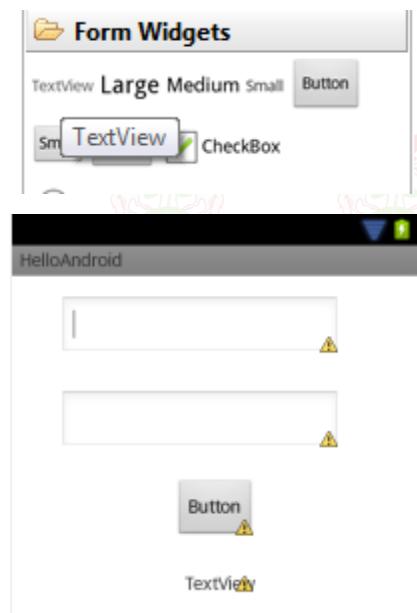
- Chọn Button từ FormWidgets và kéo sang layout





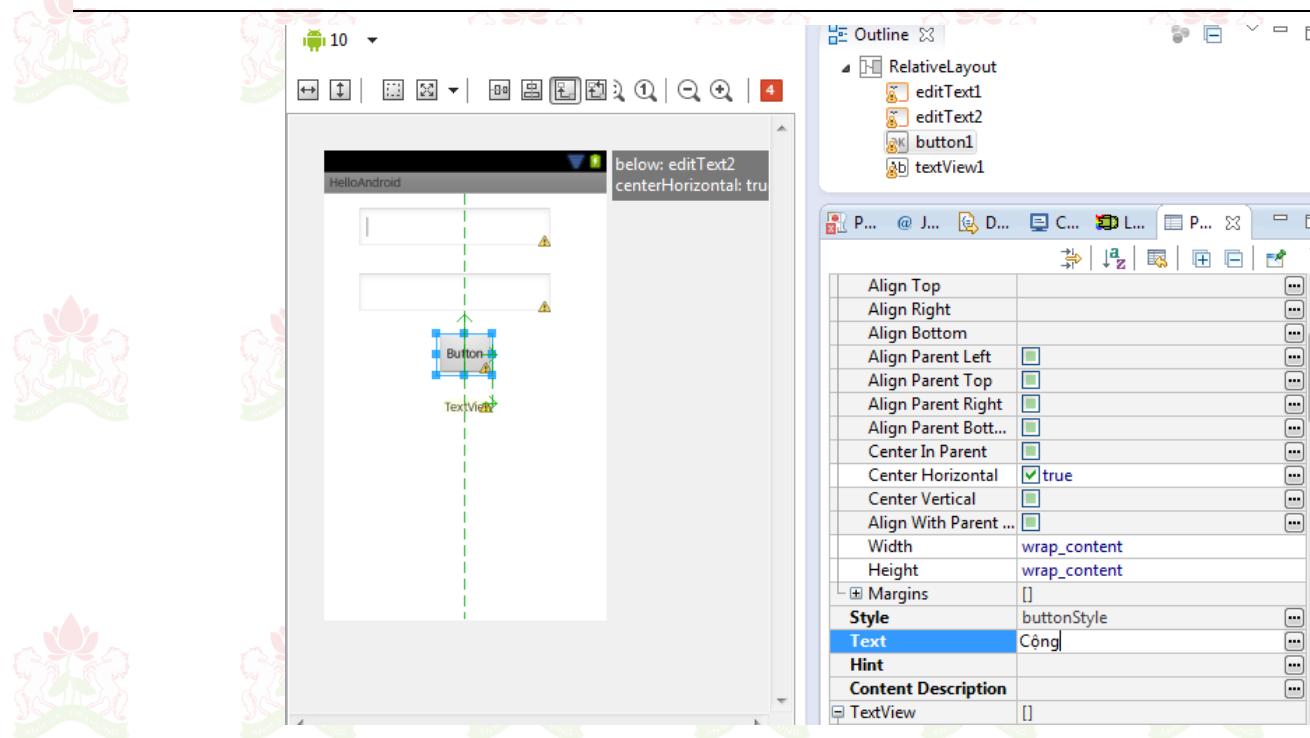
4. Tạo TextView

- Chọn TextView từ Form Widgets và kéo sang layout



5. Thay đổi thuộc tính cho đối tượng

- Chọn đối tượng
- Trong cửa sổ properties, chúng ta thay đổi các thuộc tính cần thay đổi

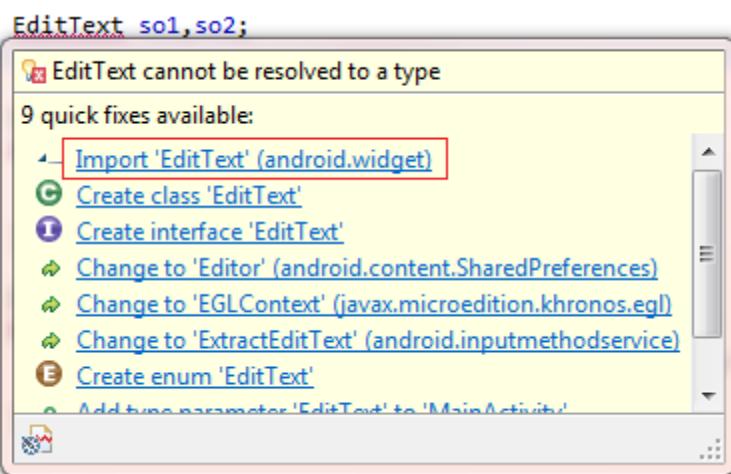


6. Viết thêm code trong MainActivity.java

- Gõ vào đoạn code sau:

```
public class MainActivity extends Activity {  
    EditText so1,so2;  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        getMenuInflater().inflate(R.menu.activity_main, menu);  
        return true;  
    }  
}
```

- Chọn import thư viện của đối tượng





- Tương tự cho các đối tượng khác, chúng ta được như sau:

```
package com.android.helloandroid;  
  
import android.os.Bundle;  
import android.app.Activity;  
import android.view.Menu;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
  
public class MainActivity extends Activity {  
  
    EditText so1,so2;  
    Button bt;  
    TextView kq;
```

- Viết code tiếp tục

```
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    so1=(EditText) findViewById(R.id.editText1);  
    so2=(EditText) findViewById(R.id.editText2);  
    kq=(TextView) findViewById(R.id.textView1);  
    bt=(Button) findViewById(R.id.button1);  
  
    bt.setOnClickListener(new View.OnClickListener() {  
  
        public void onClick(View v) {  
            // TODO Auto-generated method stub  
            int s1=Integer.parseInt(so1.getText().toString());  
            int s2=Integer.parseInt(so2.getText().toString());  
            int s3=s1+s2;  
            kq.setText("Kết quả là: "+s3);  
        }  
    });  
}
```

7. Chạy ứng dụng

- Chọn vào icon sau



- Kết quả khi chạy





MATH-DATE-REGULAREXPRESSION-STRING

1. Math class:

Chứa các phương thức và hằng số tĩnh (static) phục vụ cho toán học.

Các phương thức thông dụng:

Phương thức	Mô tả	Vd
abs(x)	Trả về trị tuyệt đối	Math.abs(-5) ;// 5
ceil(x)	Làm tròn lên	Math.ceil(5.1); //6.0
floor(x)	Làm tròn xuống	Math.floor(5.9); //5.0
round(x)	Làm tròn cân bằng (0->cận 5 tròn xuống(5->trở lên làm tròn lên	Math.round(5.1); //5.0 Math.round(5.5); //6.0
pow(x,y)	x^y	Math.pow(5,3); //125.0
sqrt(x)	Rút căn	Math.sqrt(9); //3.0
sin(x) / cos(x) / tan(x)	Lấy sin, cos tan	
random()	Số ngẫu nhiên từ 0.0 đến 1.0	Math.random()
PI	Trả về số PI	Math.PI //3.1416.....

2. Date class:

Giúp chúng ta xử lý dữ liệu dạng ngày, tháng, năm, giờ, phút, giây.....

Muốn sử dụng phải import java.util.Date;

Các hàm thông dụng để lấy và gán dữ liệu cho Date

getMonth() / setMonth(int months)	Đọc và ghi tháng (0-11)
getYear() / setYear(int years)	Đọc và ghi năm (100-2000) (phải cộng thêm 1900)
getHours() /setHours(int hours)	Đọc và ghi giờ (0-59)
getMinutes() / setMinutes(int minutes)	Đọc và ghi phút (0-59)
getSeconds() / setSeconds(int seconds)	Đọc và ghi giây (0-59)
getTime() / setTime(int miliseconds)	Đọc và ghi thời gian mili giây

3. String class:

Chuỗi là một dãy các ký tự. Có thể dùng kí hiệu “ + “ để nối chuỗi với nhau. Khởi tạo chuỗi đơn giản nhất như sau:

String ten_biem="Chuỗi";

Các phương thức thông dụng của chuỗi.



Tên phương thức	Mô Tả
length()	Trả về chiều dài chuỗi
toLowerCase()	Đổi sang chữ thường
toUpperCase()	Đổi sang chữ hoa
trim()	Cắt bỏ kí tự trắng ở 2 đầu của chuỗi
equals(String chuoi2)	So sánh chuỗi với chuoi2 . Phân biệt Hoa thường
equalsIgnoreCase(String chuoi2)	So sánh chuỗi với chuoi2. Không phân biệt Hoa thường
startsWith(String chuoi2)	So sánh chuỗi có bắt đầu là chuoi2 không
endsWith(String chuoi2)	So sánh chuỗi có kết thúc bằng chuoi2 không
indexOf(String s)	Tìm vị trí đầu tiên xuất hiện của s
lastIndexOf(String s)	Tìm ngược từ dưới lên vị trí đầu tiên xuất hiện của s
split(String mau)	Cắt chuỗi thành mảng dựa theo mẫu
substring(int dau,int cuoi)	Lấy chuỗi con trong chuỗi
getBytes()	Chuyển chuỗi sang mảng byte
String.format()	Định dạng lại chuỗi
String.valueOf(Object a)	Ép 1 dữ liệu sang kiểu chuỗi

4. RegularExpression (Biểu thức chính qui)

Dùng để so khớp một chuỗi với một mẫu xem chuỗi có thỏa theo chuẩn của mẫu hay không.

Vd: so 1 chuỗi có phải là email hay không, so 1 dãy xem có phải là số xe TPHCM không.

Muốn xây dựng mẫu phải dựa vào các kí tự, mỗi kí tự có ý nghĩa riêng

Một số kí tự thường dùng

Kí tự	Mô tả
.	Đại diện cho 1 kí tự bất kì trừ kí tự xuống dòng \n
\d	Kí tự chữ số tương đương [0-9] (một kí số)
\D	Kí tự không phải một kí số
\s	Đại diện cho kí tự trắng gồm cả tab, xuống dòng, về đầu dòng .Tương đương [\f\n\r\t\v]
\S	Kí tự không phải khoảng trắng [^ \f\n\r\t\v]
\w	Kí tự chữ cái, chữ số, dấu_. Tương đương [a-zA-Z0-9_]
\W	Tương đương [^a-zA-Z0-9_]



^	Bắt đầu 1 chuỗi hay 1 dòng
\$	Kết thúc 1 chuỗi hay 1 dòng
	Cái này hoặc cái kia . Tương đương với or
[]	Đại diện 1 kí tự được liệt kê trong []
[^]	Đại diện 1 kí tự không được liệt kê trong []
()	Phân nhóm vd: ((a(b))c) sẽ khớp với b, ab, abc
?	Tương đương {0,1}. Đứng trước từ 0 hay 1 lần. vd: A?B khớp với B hay AB
*	Tương đương {0,} . Đứng trước từ 0 hoặc nhiều lần. A*B khớp với B, AB, AAB, AAAB.....
+	Tương đương {1,}. Đứng trước từ 1 lần trở lên. A+B khớp với AB, AAB, AAAB.....
{n}	Kí tự trước nó xuất hiện đúng n lần
{min,max}	Kí tự trước nó xuất hiện ít nhất min lần nhiều nhất max lần

Ví dụ:

- Email đơn giản:

String regex="\\w+@\\w+\\.\\w+";

- Số CMND:

String regex="\d{10}";

Số điện thoại di động:

String regex="0\\d{8,9}";

Số xe máy TPHCM

String regex = "5\\d-[A-Z]\\d-\\d{3}.{2}";

- Địa chỉ web:

String regex="http://www.(\\w+\\.\\.\\.)\\.\\w+";

Khi sử dụng ta phải dùng với class Pattern và class Matcher như ví dụ bên dưới

```

String soxe=et_soixe.getText().toString();
String chuoi_mau="5[0-9]-[A-Z][0-9]-[0-9]{3}\\.[0-9]{2}";

Pattern patter=Pattern.compile(chuoi_mau);
Matcher matcher=patter.matcher(soxe);

if(matcher.matches()==true)
    tv_kq.setText("Số xe hợp lệ TPHCM");
else
    tv_kq.setText("Số xe không hợp lệ");

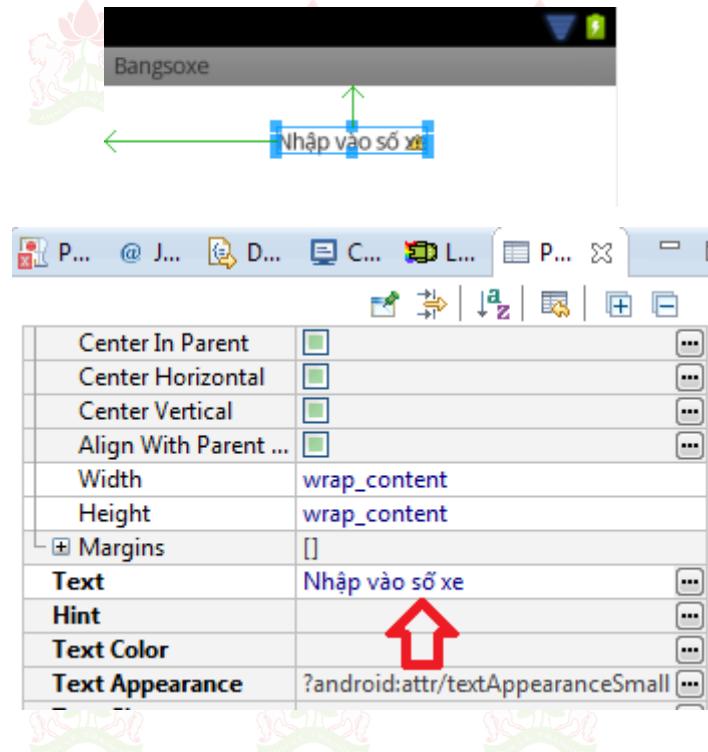
```



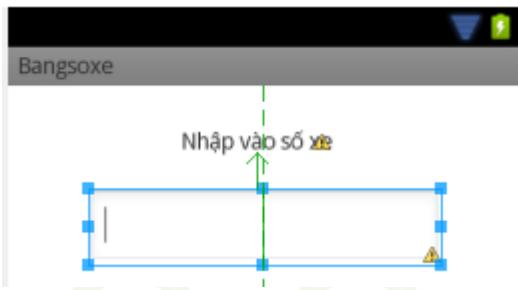
BẢNG SỐ XE

1. Tạo project mới
2. Tạo các đối tượng

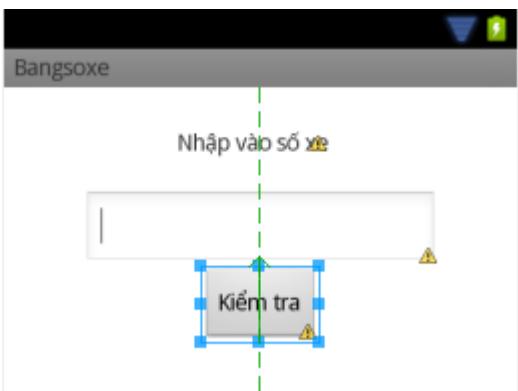
- Tạo TextView

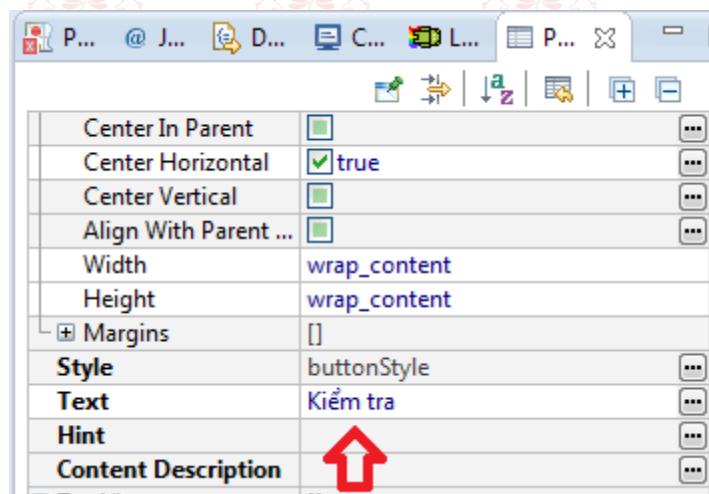


- Tạo EditText

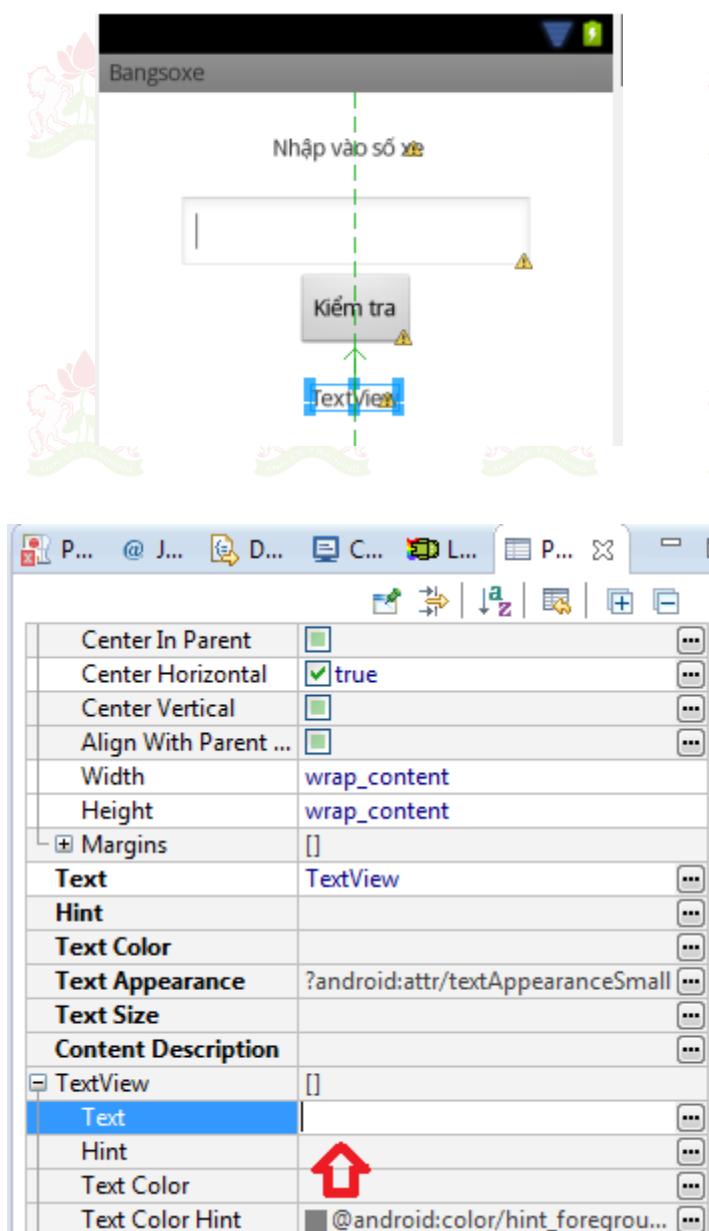


- Tạo Button



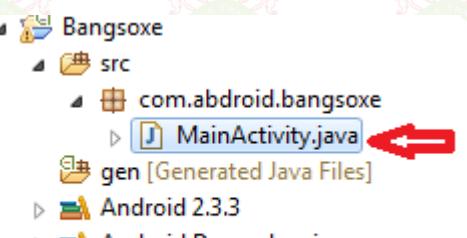


- Tạo TextView (để báo cáo kết quả kiểm tra)





3. Viết code cho MainActivity.java trong thư mục src



```
public class MainActivity extends Activity {

    Matcher matcher;
    TextView kq;
    EditText bs;
    Button bt;
    String s;
    String mau="5[0-9]-[A-Z][0-9]-[0-9]{3}\\.[0-9]{2}";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        bs=(EditText)findViewById(R.id.editText1);
        kq=(TextView)findViewById(R.id.textView2);
        bt=(Button)findViewById(R.id.button1);

        bt.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {
                // TODO Auto-generated method stub
                Pattern pt=Pattern.compile(mau);
                s=bs.getText().toString();
                matcher=pt.matcher(s);

                if (matcher.matches()==true){
                    kq.setText("Bảng số đúng dạng của TP.HCM");
                }
                else
                {
                    kq.setText("Bảng số không đúng dạng của TP.HCM");
                }
            }
        });
    }
}
```



4. Chạy thử ứng dụng

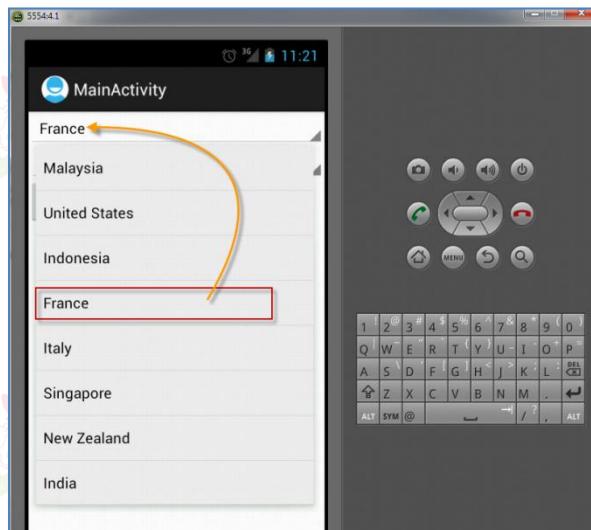




SPINNER

1. Giới thiệu

- Chọn các Spinner → Submit → Giá trị Spinner sẽ hiển thị như sau.

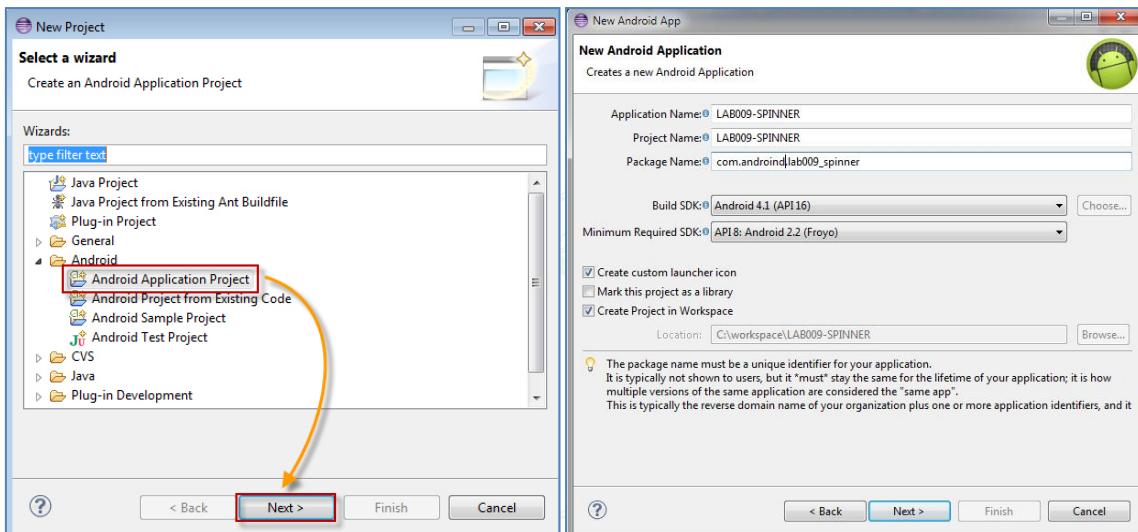


2. Tạo Project

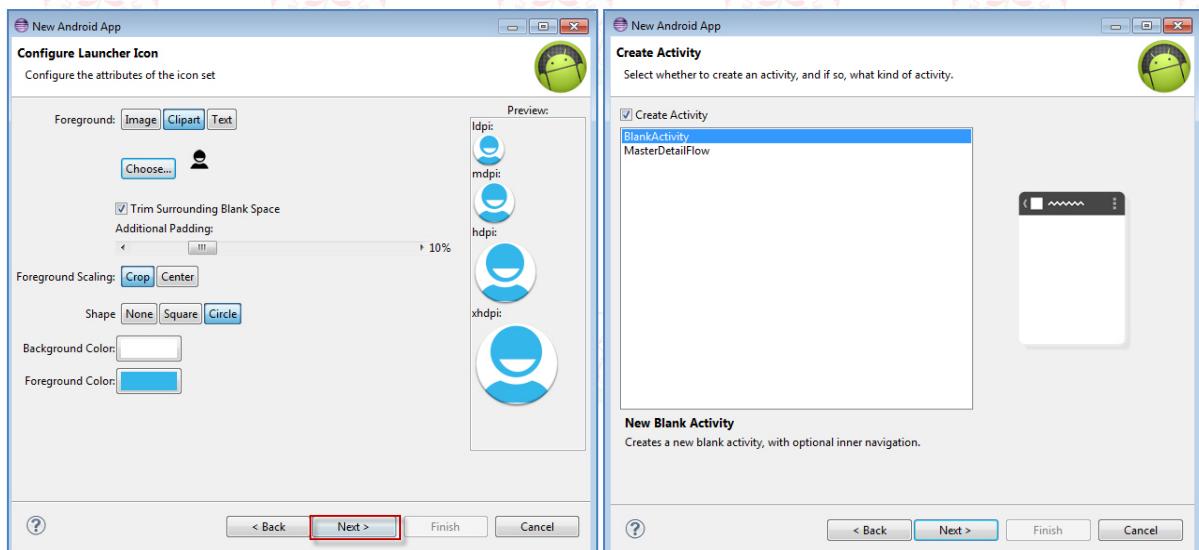
- Chọn menu : File → New → Project.



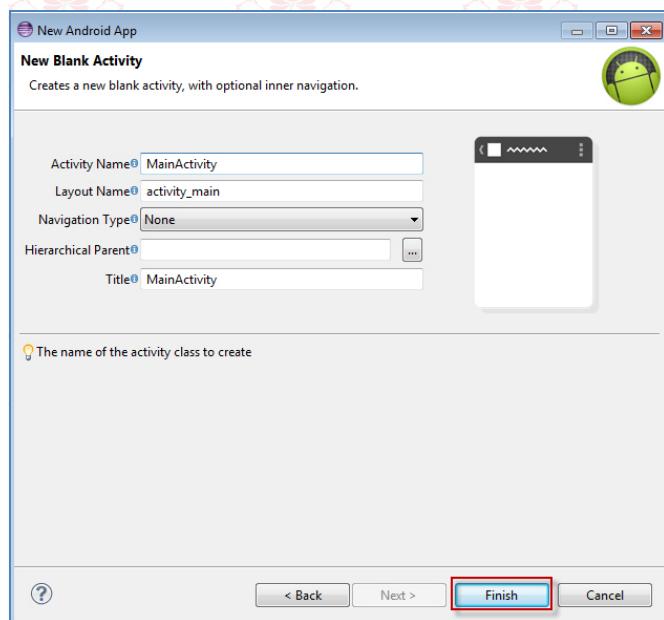
- Chọn Androind Application project → Next → Khai báo Project như sau.



- Hộp thoại Configuration Laucher Icon → Tùy chọn Icon → Next.
- Hộp thoại Create Activity : Chọn Activity : Blank Activity → Next.



- Khai báo Activity chính.



3. Thiết kế giao diện

- Viết code file strings.xml như sau.

```

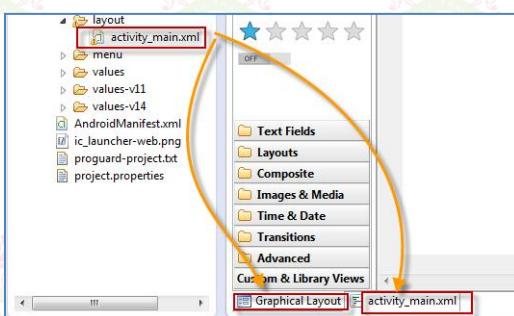
activity_main.xml strings.xml MainActivity.java
<resources>
    <string name="app_name">LAB009-SPINNER</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>
    <string name="title_activity_main">MainActivity</string>
    <string name="country_prompt">Choose a country</string>
    <string-array name="country_arrays">
        <item>Malaysia</item>
        <item>United States</item>
        <item>Indonesia</item>
        <item>France</item>
        <item>Italy</item>
        <item>Singapore</item>
        <item>New Zealand</item>
        <item>India</item>
    </string-array>
</resources>

```

- Sử dụng file : activity_main.xml trong thư mục layout để thiết kế giao diện.



- Có hai dạng : Graphical layout (Giao diện đồ họa) và *.xml (Code)



- Thiết kế giao diện theo code và kết quả như sau.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:entries="@array/country_arrays"
        android:prompt="@string/country_prompt" />

    <Spinner
        android:id="@+id/spinner2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/btnSubmit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit" />

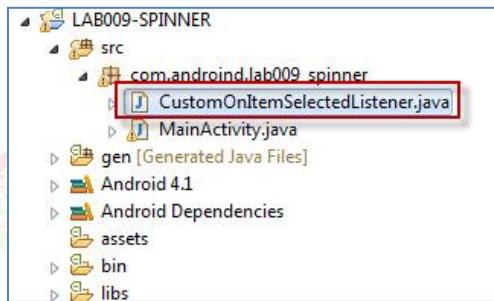
</LinearLayout>
```



4. Viết code trang Activity.java

```
activity_main.xml>MainActivity.java
import java.util.ArrayList;
import java.util.List;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;
public class MainActivity extends Activity {
    private Spinner spinner1, spinner2; private Button btnSubmit;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addItemsOnSpinner2(); addListenerOnButton(); addListenerOnSpinnerItemSelection();
    }
    private void addItemsOnSpinner2() {
        spinner2 = (Spinner) findViewById(R.id.spinner2);
        List<String> list = new ArrayList<String>();
        list.add("list 1"); list.add("list 2"); list.add("list 3");
        ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_spinner_item, list);
        dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner2.setAdapter(dataAdapter);
    }
    private void addListenerOnButton() {
        spinner1 = (Spinner) findViewById(R.id.spinner1);
        spinner2 = (Spinner) findViewById(R.id.spinner2);
        btnSubmit = (Button) findViewById(R.id.btnSubmit);
        btnSubmit.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                Toast.makeText(MainActivity.this,"OnClickListener : " +
                    "\nSpinner 1 : "+ String.valueOf(spinner1.getSelectedItem()) +
                    "\nSpinner 2 : "+ String.valueOf(spinner2.getSelectedItem()),Toast.LENGTH_SHORT).show();
            }
        });
    }
    private void addListenerOnSpinnerItemSelection() {
        spinner1 = (Spinner) findViewById(R.id.spinner1);
        spinner1.setOnItemSelectedListener(new CustomOnItemSelectedListener());
    }
}
```

- Trong quá trình viết code thông báo lỗi → Tạo thêm file CustomOnItemSelectedListener.java như sau.



- Viết code trang CustomOnItemSelectedListener.java



```
activity_main.xml strings.xml MainActivity.java CustomOnItemSelectedListener.java
package com.androind.lab009_spinner;

import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.Toast;

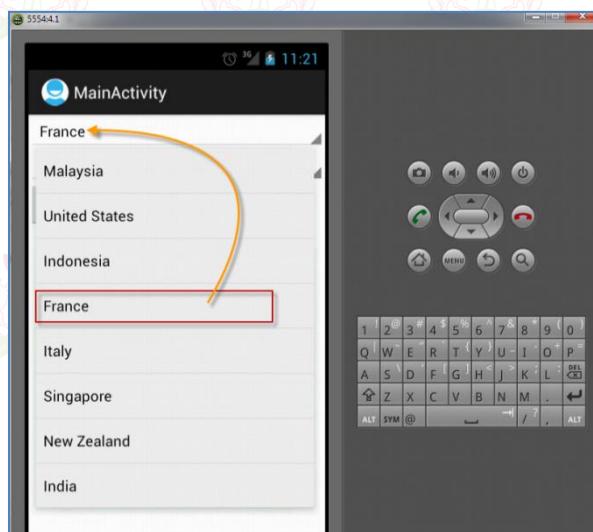
public class CustomOnItemSelectedListener implements OnItemSelectedListener {

    public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {
        Toast.makeText(parent.getContext(),
                "OnItemSelectedListener : " + parent.getItemAtPosition(pos).toString(),
                Toast.LENGTH_SHORT).show();
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}
```

5. Run ứng dụng

- Chọn các Spinner → Submit → Giá trị Spinner sẽ hiển thị như sau.

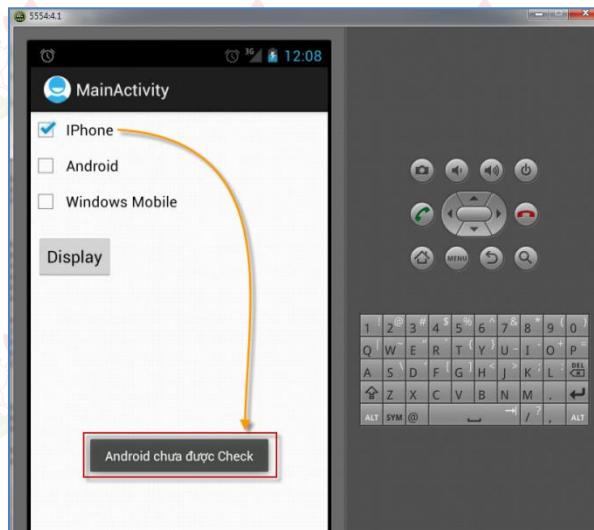




CHECKBOX

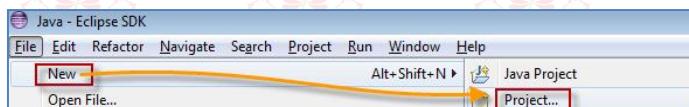
1. Giới thiệu

- Khi check vào iPhone → Xuất hiện đoạn Text thông báo : Androind chưa được Check
- Khi check vào Androind → Click Display → Hộp thoại thông báo : Androind check : true;

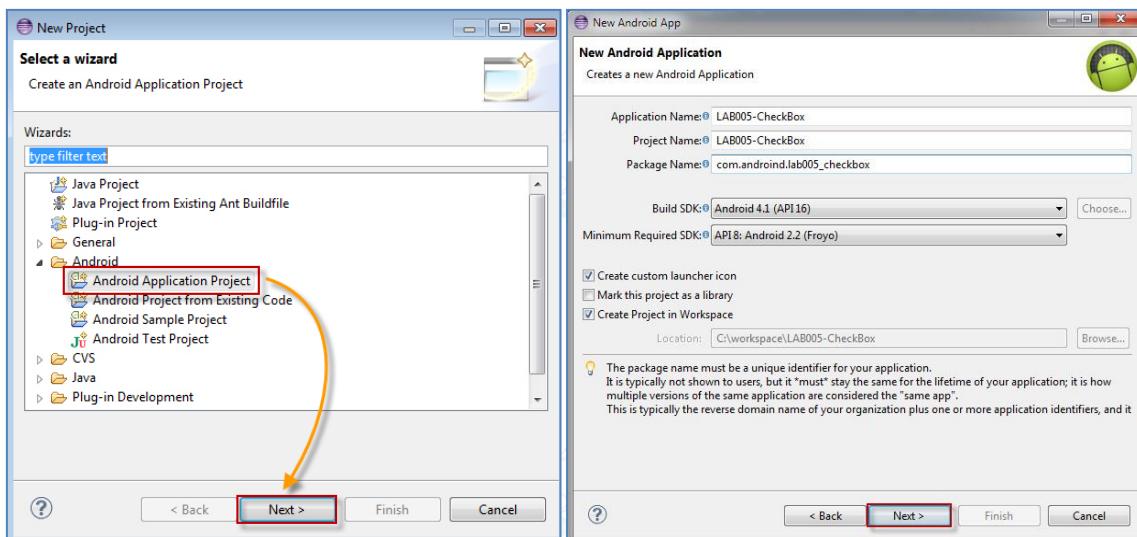


2. Tạo Project

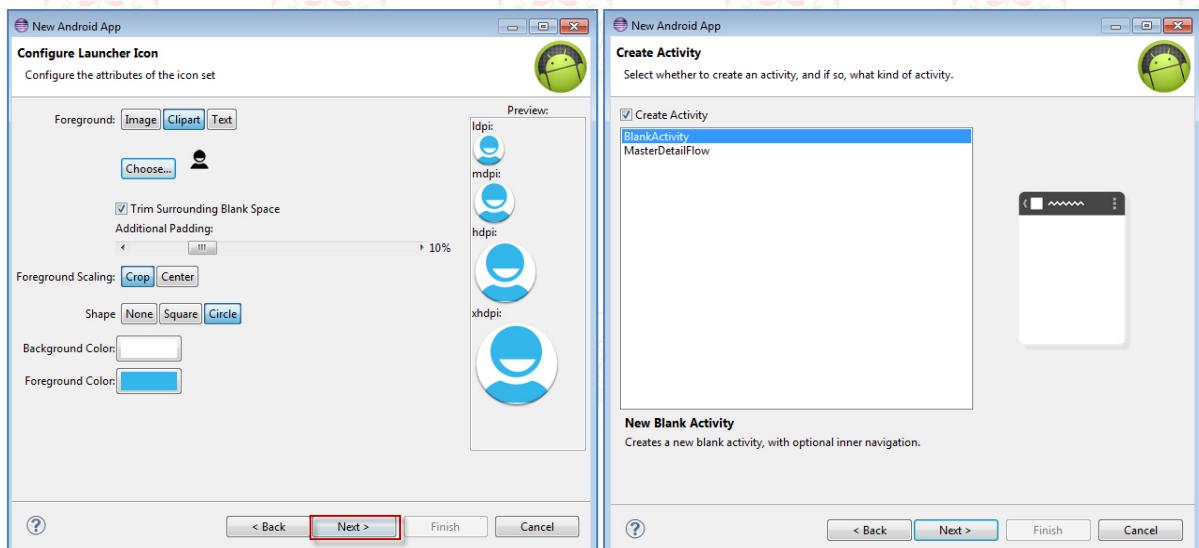
- Chọn menu : File → New → Project.



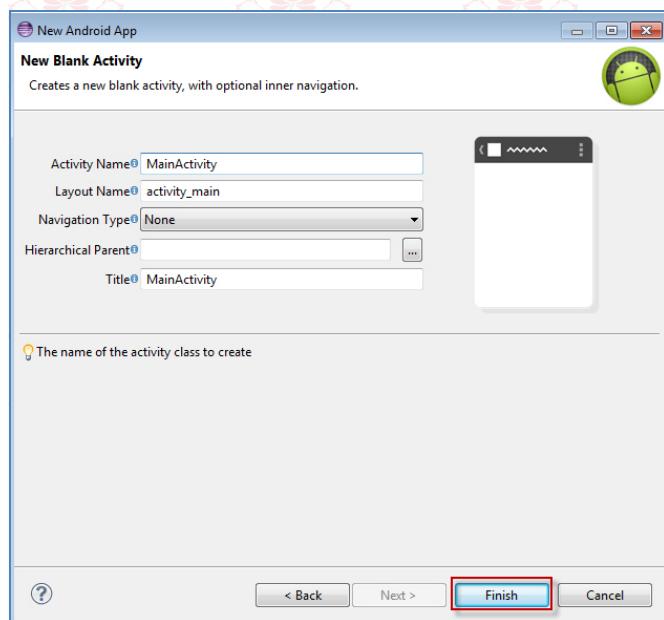
- Chọn Androind Application project → Next → Khai báo Project như sau.



- Hộp thoại Configuration Laucher Icon → Tùy chọn Icon → Next.
- Hộp thoại Create Activity : Chọn Activity : Blank Activity → Next.



- Khai báo Activity chính.



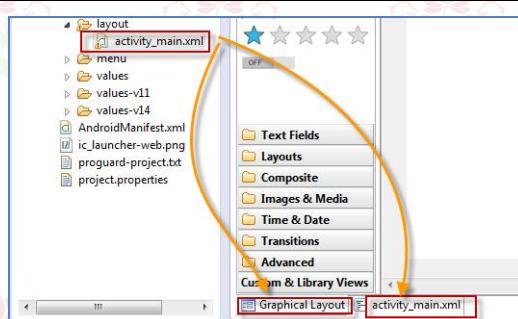
3. Thiết kế giao diện

- Viết code trang string.xml như sau.

The image shows the 'strings.xml' file in the Android Studio resources tab. The XML code defines several string resources:

```
<resources>
    <string name="app_name">LAB005-CheckBox</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>
    <string name="title_activity_main">MainActivity</string>
    <string name="chk_ios">iPhone</string>
    <string name="chk_android">Android</string>
    <string name="chk_windows">Windows Mobile</string>
    <string name="btn_display">Display</string>
</resources>
```

- Sử dụng file : activity_main.xml trong thư mục layout để thiết kế giao diện.
- Có hai dạng : Graphical layout (Giao diện đồ họa) và *.xml (Code)



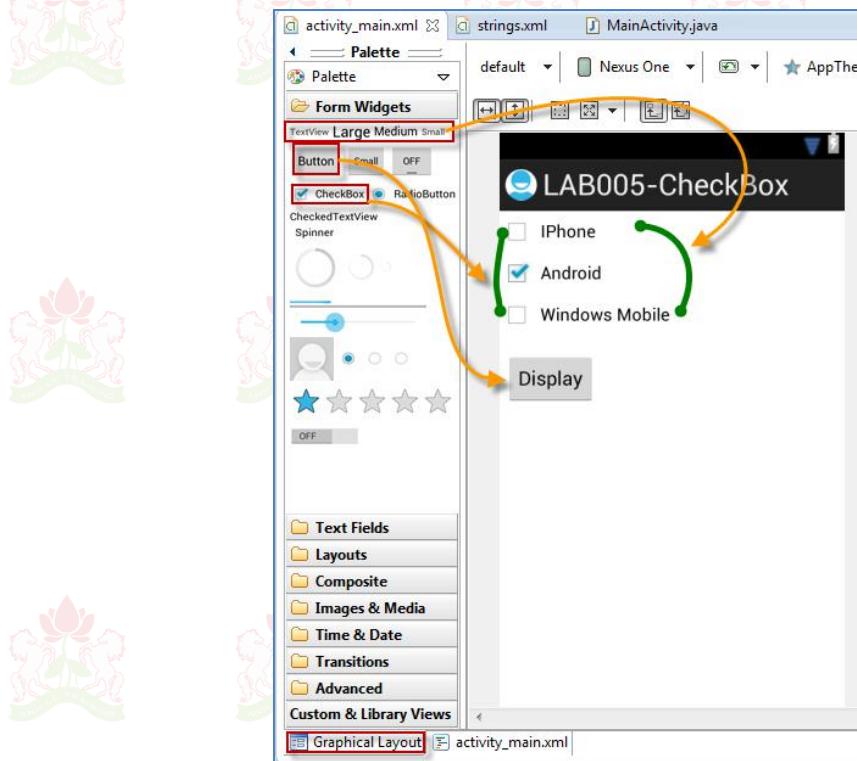
- Thiết kế giao diện theo code như sau.

```
<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" >

    <CheckBox
        android:id="@+id/chkIos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/chk_ios" />
    <CheckBox
        android:id="@+id/chkAndroid"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/chkIos"
        android:checked="true"
        android:text="@string/chk_android" />
    <CheckBox
        android:id="@+id/chkWindows"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/chkAndroid"
        android:text="@string/chk_windows" />
    <Button
        android:id="@+id/btnDisplay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/chkIos"
        android:layout_below="@+id/chkWindows"
        android:layout_marginTop="16dp"
        android:text="@string/btn_display" />
</RelativeLayout>
```

- Hoặc có thể tạo giao diện theo đồ họa như sau.





4. Viết code trang Activity.java

```
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;

public class MainActivity extends Activity {
    private CheckBox chkIos, chkAndroid, chkWindows; private Button btnDisplay;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addListenerOnChkIos();
        addListenerOnButton();
    }
    private void addListenerOnButton() {
        chkIos = (CheckBox) findViewById(R.id.chkIos);
        chkAndroid = (CheckBox) findViewById(R.id.chkAndroid);
        chkWindows = (CheckBox) findViewById(R.id.chkWindows);
        btnDisplay = (Button) findViewById(R.id.btnDisplay);
        btnDisplay.setOnClickListener(new OnClickListener() { //Run when button is clicked
            public void onClick(View v) {
                StringBuffer result = new StringBuffer();
                result.append("iPhone check : ").append(chkIos.isChecked());
                result.append("\nAndroid check : ").append(chkAndroid.isChecked());
                result.append("\nWindows Mobile check : ").append(chkWindows.isChecked());
                Toast.makeText(MainActivity.this, result.toString(), Toast.LENGTH_LONG).show();
            }
        });
    }
    private void addListenerOnChkIos() {
        chkIos = (CheckBox) findViewById(R.id.chkIos);
        chkIos.setOnClickListener(new OnClickListener() {
            public void onClick(View v) { //is chkIos checked?
                if (((CheckBox) v).isChecked()) {
                    Toast.makeText(MainActivity.this, "Android chưa được Check", Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
```

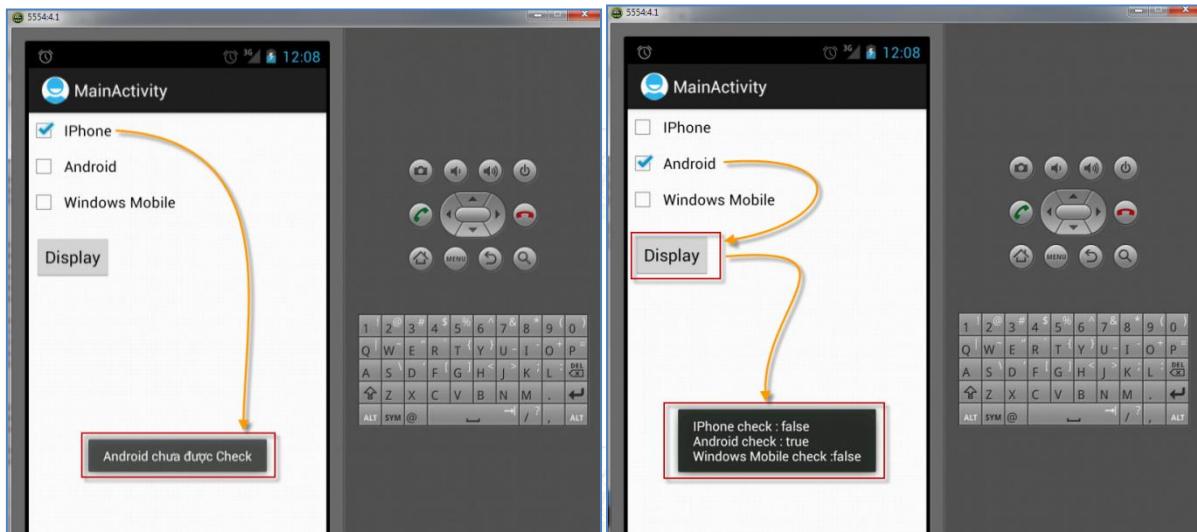
.public void onClick(View v){...} : Click vào CheckBox.

.((CheckBox) v).isChecked() : Kiểm tra đã check vào CheckBox



5. Run ứng dụng

- Khi check vào iPhone → Xuất hiện đoạn Text thông báo : Androind chưa được Check
- Khi check vào Androind → Click Display → Hộp thoại thông báo : Androind check : true;

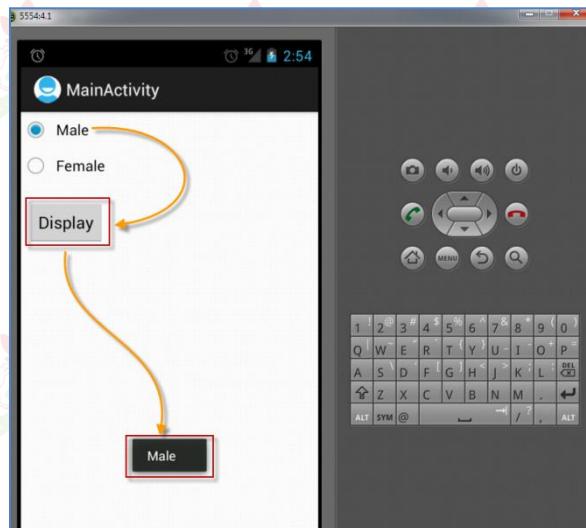




RADIOBUTTON

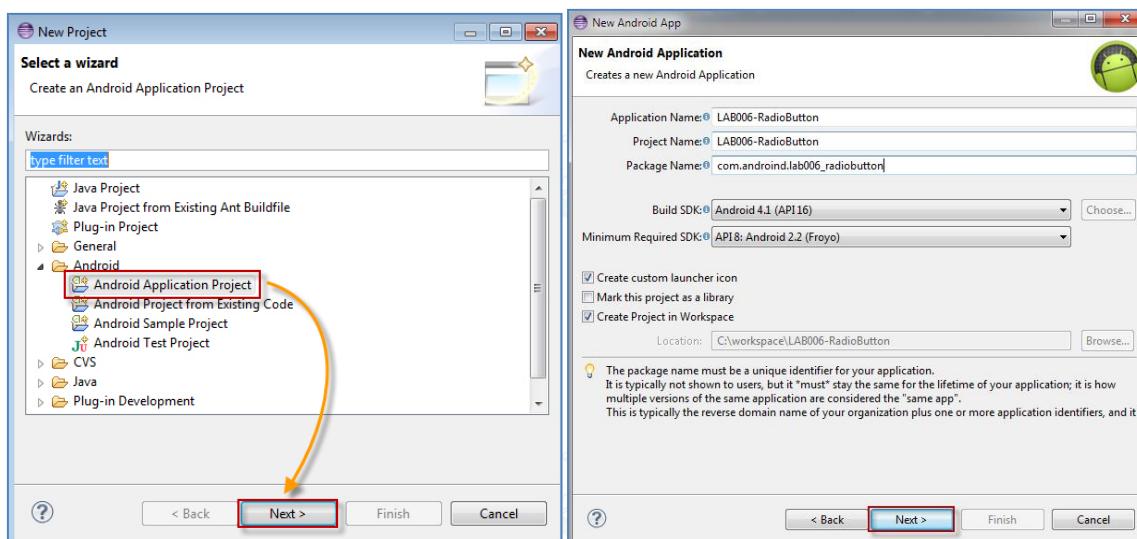
1. Giới thiệu

- Khi check vào Male → Click Display → Hộp thoại thông báo : Male
- Khi check vào Female → Click Display → Hộp thoại thông báo : Female

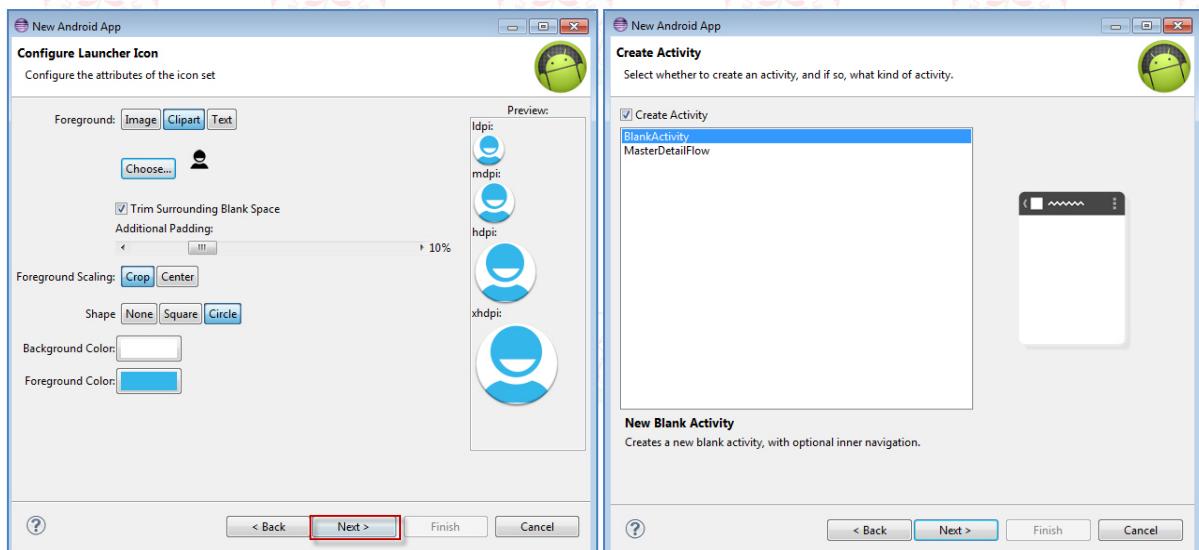


2. Tạo Project

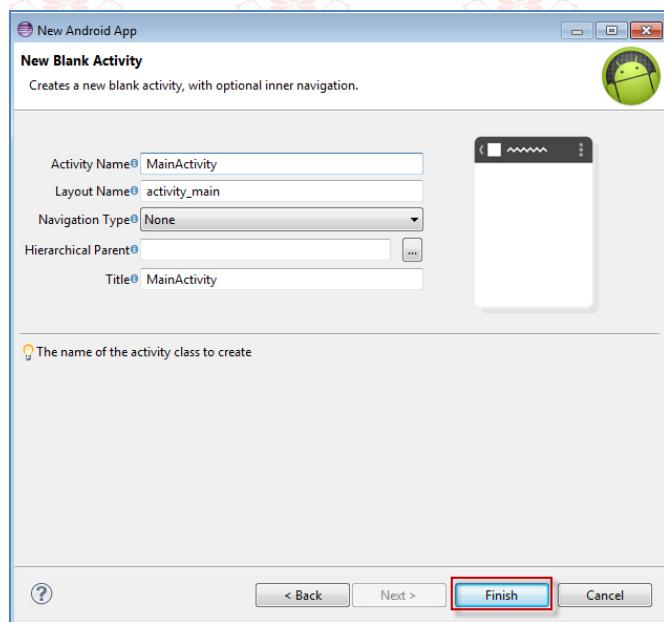
- Chọn menu : File → New → Project.
- Chọn Androind Application project → Next → Khai báo Project như sau.



- Hộp thoại Configuration Laucher Icon → Tùy chọn Icon → Next.
- Hộp thoại Create Activity : Chọn Activity : Blank Activity → Next.



- Khai báo Activity chính.



3. Thiết kế giao diện

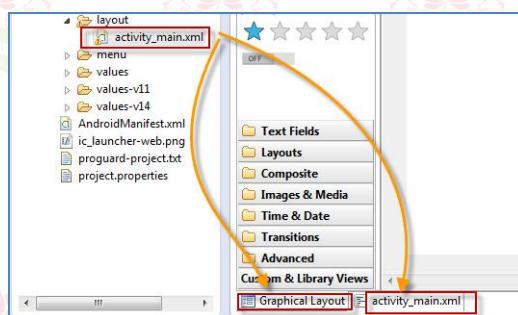
- Viết code trang string.xml như sau.

```

<resources>
    <string name="app_name">LAB006-RadioButton</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>
    <string name="title_activity_main">MainActivity</string>
    <string name="radio_male">Male</string>
    <string name="radio_female">Female</string>
    <string name="btn_display">Display</string>
</resources>

```

- Sử dụng file : activity_main.xml trong thư mục layout để thiết kế giao diện.
- Có hai dạng : Graphical layout (Giao diện đồ họa) và *.xml (Code)



- Thiết kế giao diện theo code như sau.

```
<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" >

    <RadioGroup
        android:id="@+id/radioSex"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

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

        <RadioButton
            android:id="@+id/radioFemale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/radio_female" />

    </RadioGroup>

    <Button
        android:id="@+id/btnDisplay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/radioSex"
        android:layout_below="@+id/radioSex"
        android:layout_marginTop="18dp"
        android:text="@string/btn_display" />

</RelativeLayout>
```

- Hoặc có thể tạo giao diện theo đồ họa như sau.



4. Viết code trang Activity.java

```
activity_main.xml strings.xml MainActivity.java

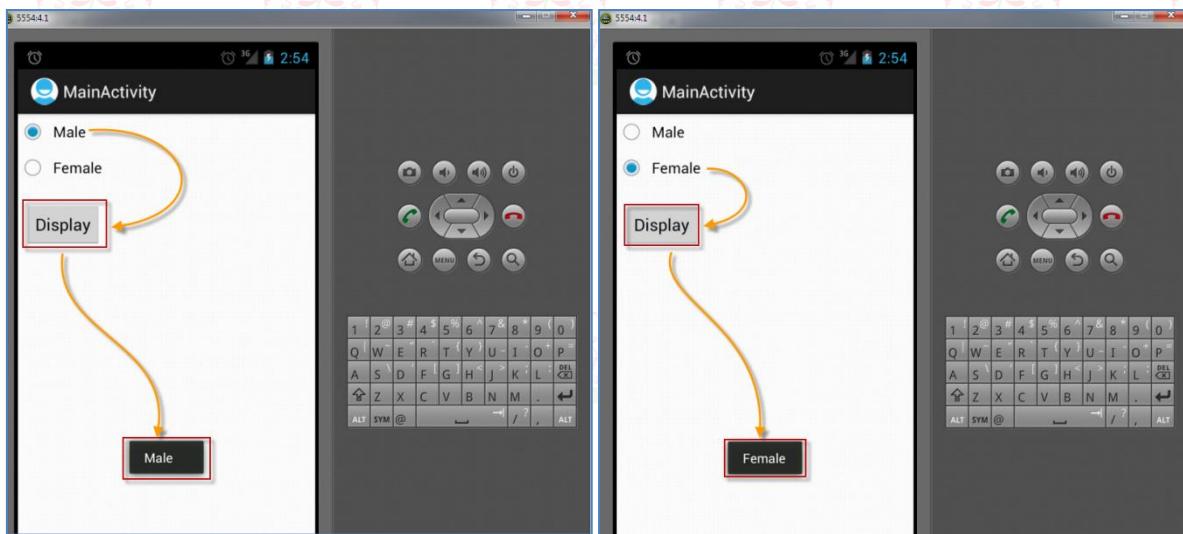
package com.androind.lab006_radiobutton;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;

public class MainActivity extends Activity {
    private RadioGroup radioSexGroup;
    private RadioButton radioSexButton;
    private Button btnDisplay;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addListenerOnButton();
    }
    private void addListenerOnButton() {
        radioSexGroup = (RadioGroup) findViewById(R.id.radioSex);
        btnDisplay = (Button) findViewById(R.id.btnDisplay);
        btnDisplay.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                // get selected radio button from radioGroup
                int selectedId = radioSexGroup.getCheckedRadioButtonId();
                // find the radiobutton by returned id
                radioSexButton = (RadioButton) findViewById(selectedId);
                Toast.makeText(MainActivity.this,
                        radioSexButton.getText(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

5. Run ứng dụng

- Khi check vào Male → Click Display → Hộp thoại thông báo : Male
- Khi check vào Female → Click Display → Hộp thoại thông báo : Female

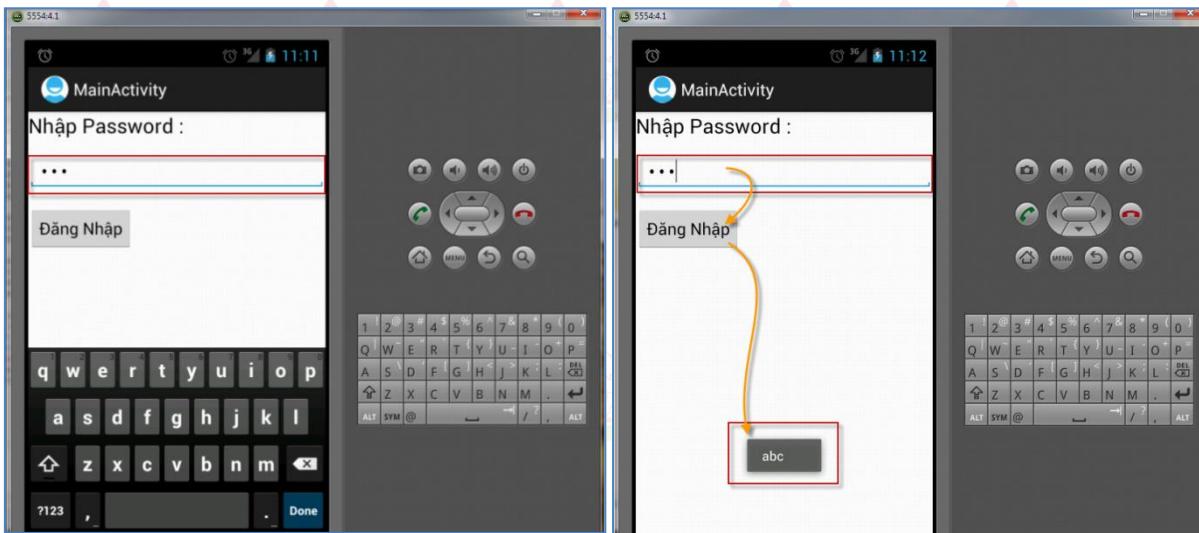




PASSWORD

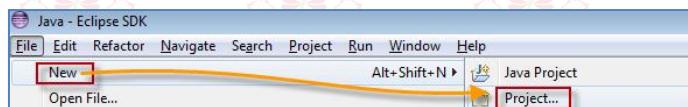
1. Giới thiệu

- Khi nhập Password vào EditText → Click Đăng Nhập → Đoạn Text nhập từ EditText sẽ được hiển thị ra như sau.

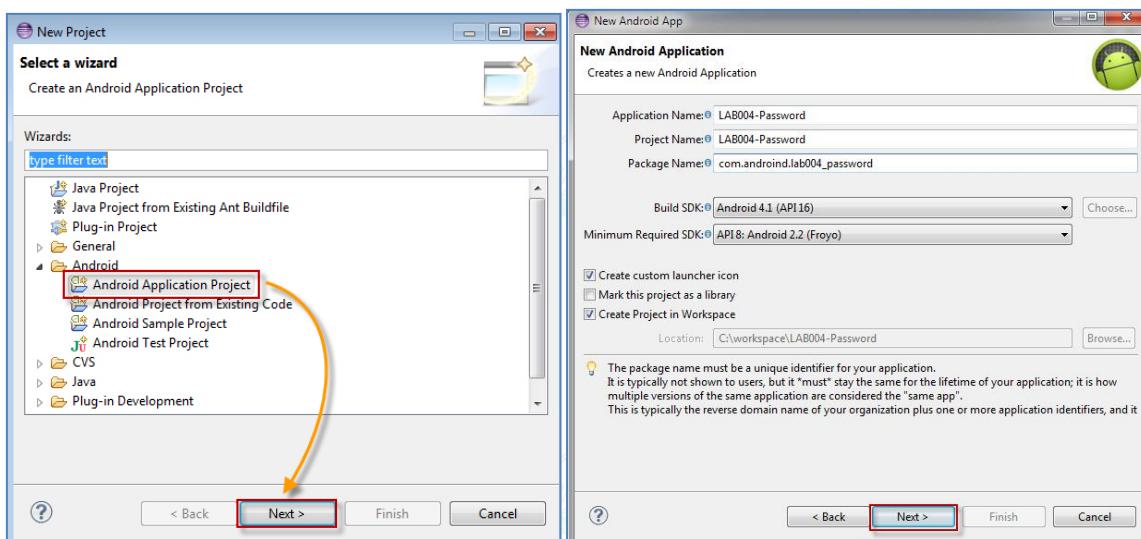


2. Tạo Project

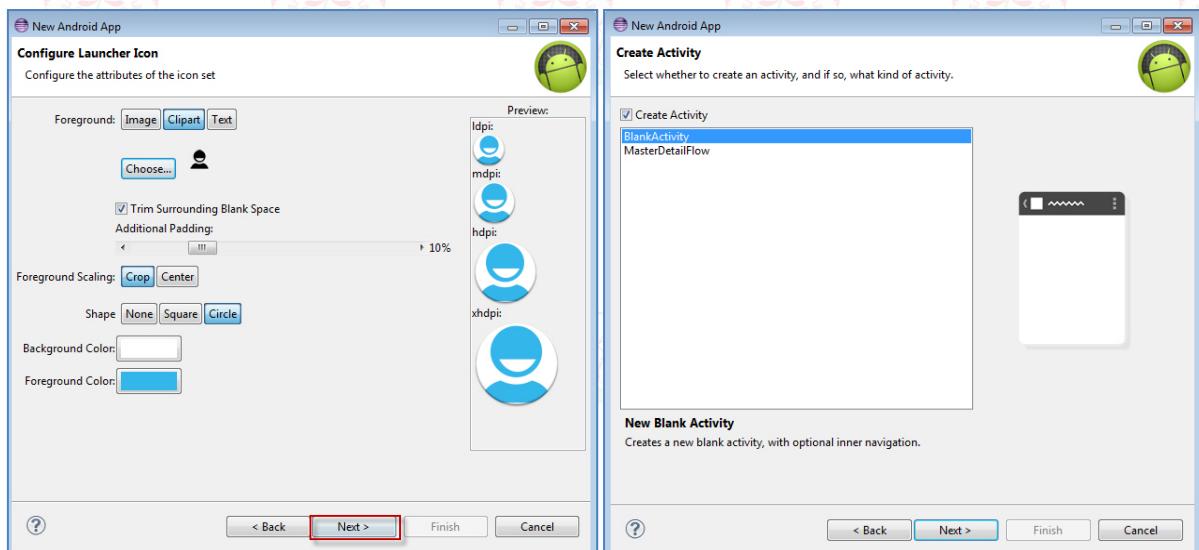
- Chọn menu : File → New → Project.



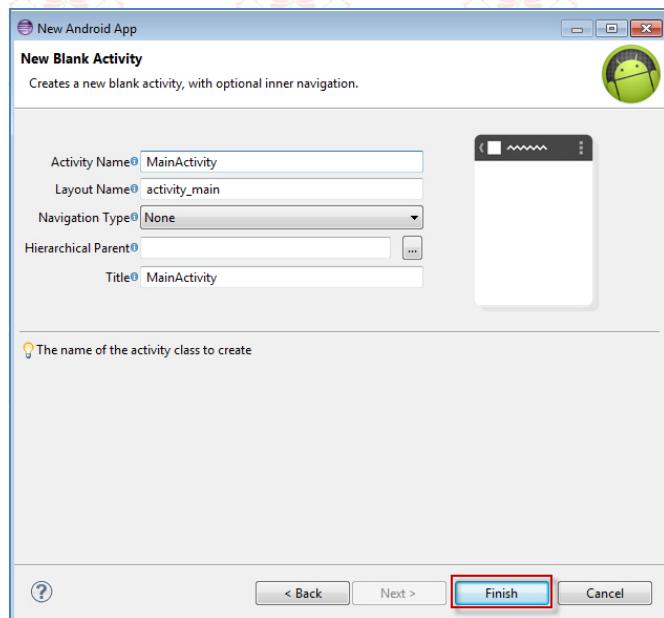
- Chọn Androind Application project → Next → Khai báo Project như sau.



- Hộp thoại Configuration Laucher Icon → Tùy chọn Icon → Next.
- Hộp thoại Create Activity : Chọn Activity : Blank Activity → Next.

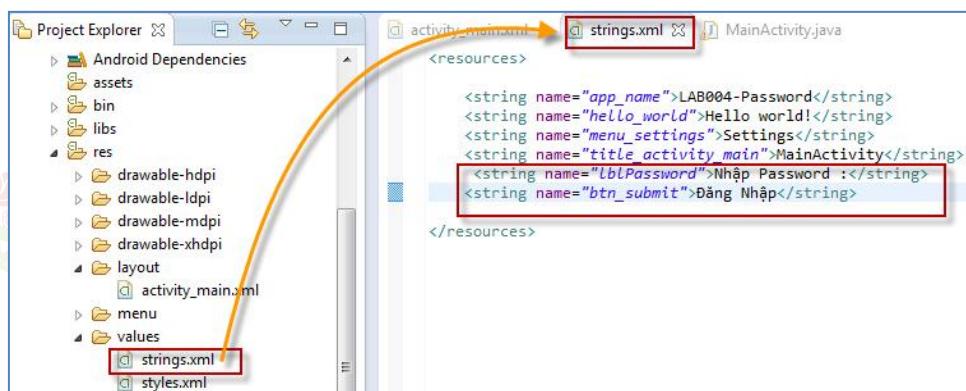


- Khai báo Activity chính.

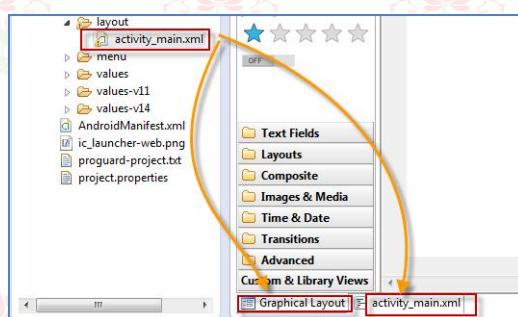


3. Thiết kế giao diện

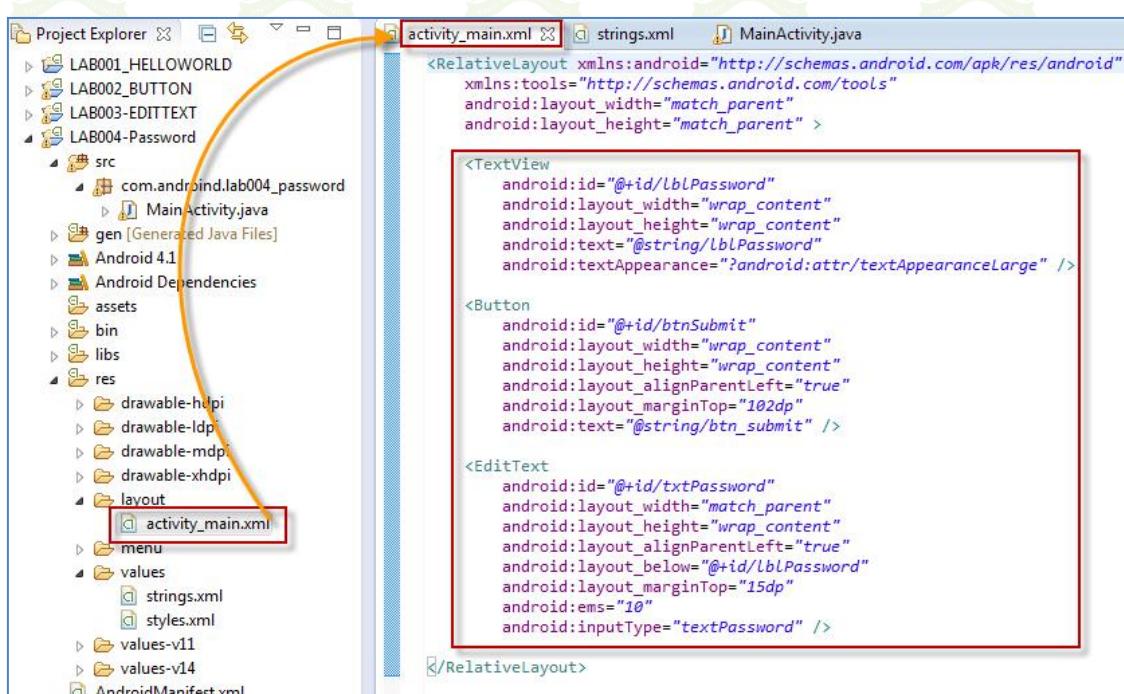
- Viết code trang string.xml như sau.



- Sử dụng file : activity_main.xml trong thư mục layout để thiết kế giao diện.
- Có hai dạng : Graphical layout (Giao diện đồ họa) và *.xml (Code)



- Thiết kế giao diện theo code như sau.



- Hoặc có thể tạo giao diện theo đồ họa như sau.





4. Viết code trang Activity.java

```

Project Explorer
LAB001_HELLOWORLD
LAB002_BUTTON
LAB003-EDITTEXT
LAB004-Password
src
com.androidlab004_password
MainActivity.java
gen [Generated Java Files]
Android 4.1
Android Dependencies
assets
bin
libs
res
drawable-hdpi
drawable-ldpi
drawable-mdpi
drawable-xhdpi
layout
activity_main.xml
menu
values
strings.xml
styles.xml
values-v11
values-v14
AndroidManifest.xml

activity_main.xml strings.xml *MainActivity.java
package com.androidlab004_password;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {
    private EditText password;
    private Button btnSubmit;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addListenerOnButton();
    }

    private void addListenerOnButton() {
        password = (EditText) findViewById(R.id.txtPassword);
        btnSubmit = (Button) findViewById(R.id.btnSubmit);
        btnSubmit.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                Toast.makeText(MainActivity.this, password.getText(),
                        Toast.LENGTH_SHORT).show();
            }
        });
    }
}

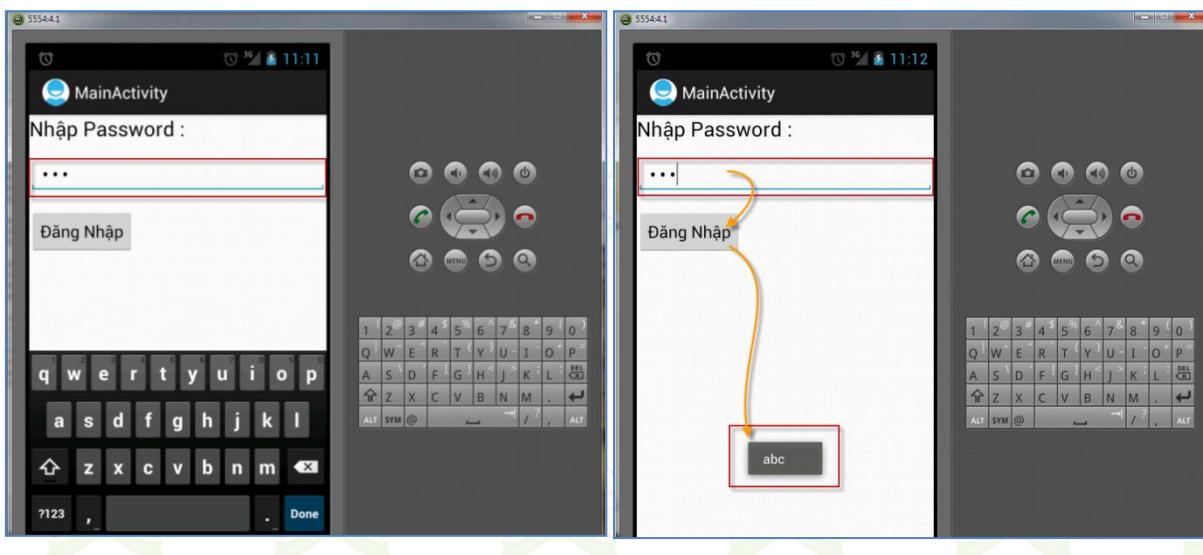
```

.Toast.makeText(...).show() : Show dữ liệu.

.password.getText() : lấy dữ liệu từ EditText (password)

5. Run ứng dụng

- Khi nhập Password vào EditText → Click Đăng Nhập → Đoạn Text nhập từ EditText sẽ được hiển thị ra như sau.

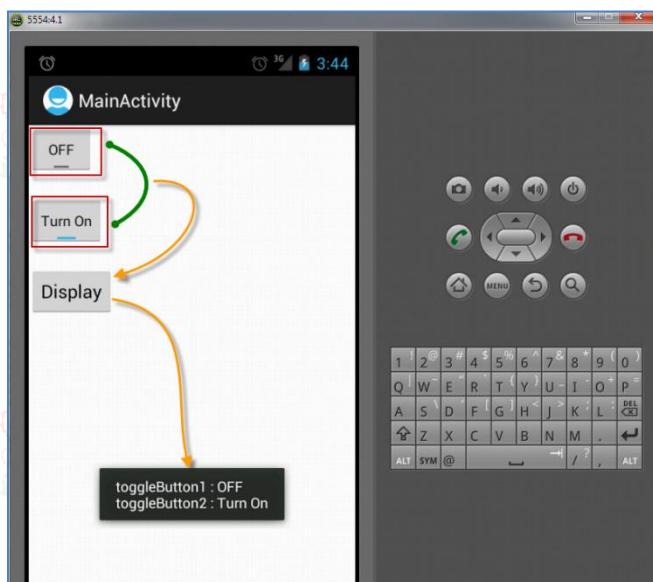




TOGGLEBUTTON

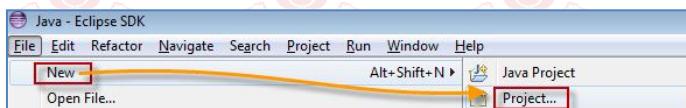
1. Giới thiệu

- Khi Click vào Dispaly → kết quả giá trị hiển thị được nhận từ giá của hai ToggleButton.

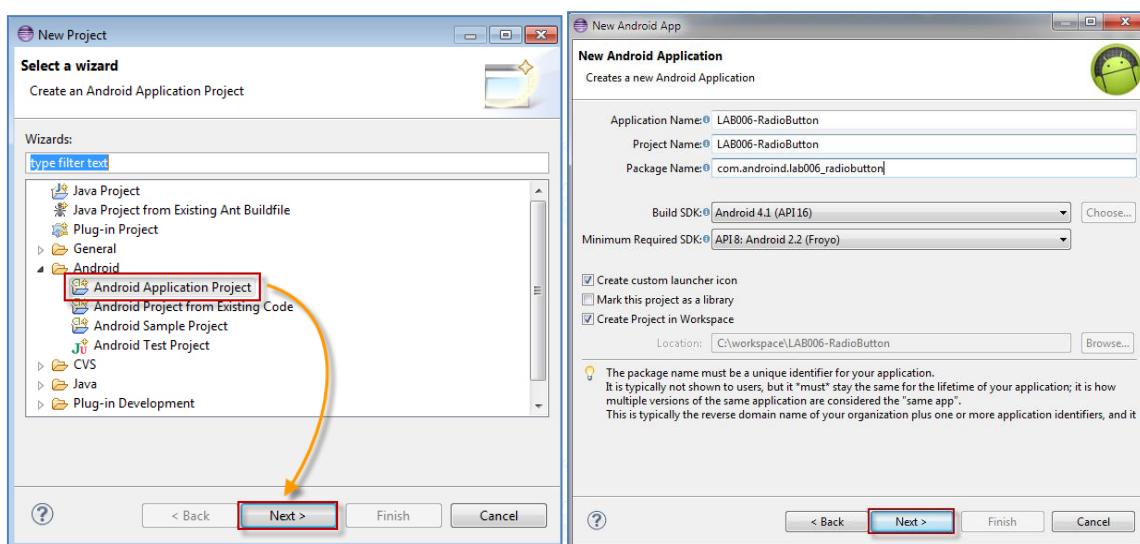


2. Tạo Project

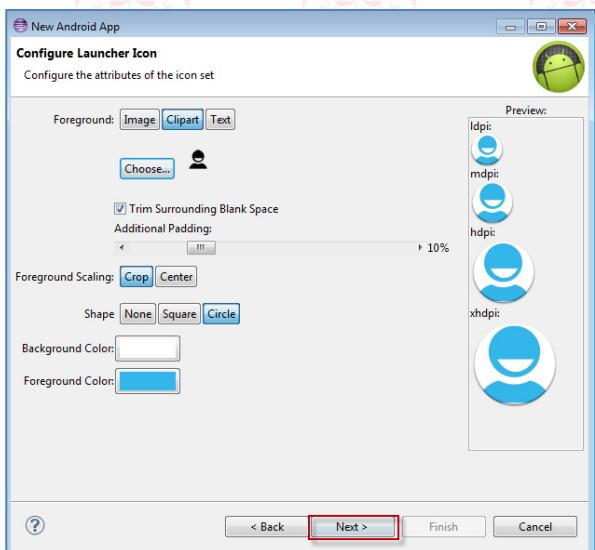
- Chọn menu : File → New → Project.



- Chọn Androind Application project → Next → Khai báo Project nhu sau.



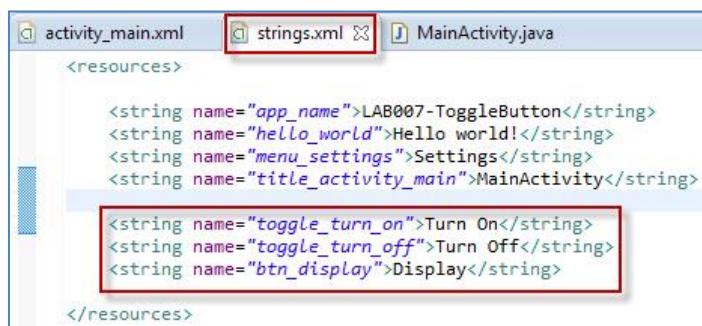
- Hộp thoại Configuration Laucher Icon → Tùy chọn Icon → Next.
- Hộp thoại Create Activity : Chọn Activity : Blank Activity → Next.



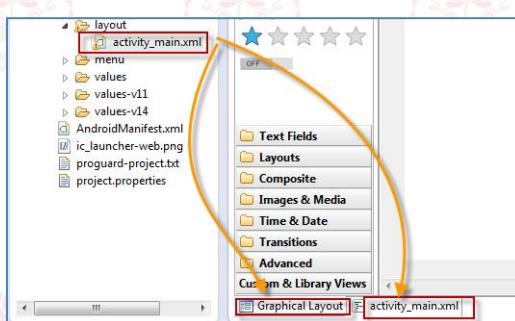
- Khai báo Activity chính.

3. Thiết kế giao diện

- Viết code trang string.xml như sau.



- Sử dụng file : activity_main.xml trong thư mục layout để thiết kế giao diện.
- Có hai dạng : Graphical layout (Giao diện đồ họa) và *.xml (Code)



- Thiết kế giao diện theo code như sau.



```
<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" >

    <ToggleButton
        android:id="@+id/toggleButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ToggleButton" />

    <ToggleButton
        android:id="@+id/toggleButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/toggleButton1"
        android:layout_marginTop="22dp"
        android:checked="true"
        android:textOff="@string/toggle_turn_off"
        android:textOn="@string/toggle_turn_on" />

    <Button
        android:id="@+id	btnDisplay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/toggleButton2"
        android:layout_marginTop="21dp"
        android:text="@string/btn_display" />

</RelativeLayout>
```

- Hoặc có thể tạo giao diện theo đồ họa như sau.





4. Viết code trang Activity.java

```
activity_main.xml strings.xml MainActivity.java
package com.androidlab007_togglebutton;

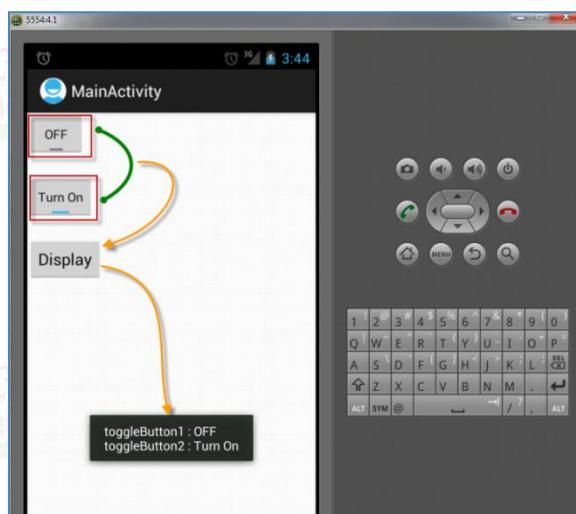
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;
import android.widget.ToggleButton;

public class MainActivity extends Activity {
    private ToggleButton toggleButton1, toggleButton2;
    private Button btnDisplay;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addListenerOnButton();
    }
    private void addListenerOnButton() {
        toggleButton1 = (ToggleButton) findViewById(R.id.toggleButton1);
        toggleButton2 = (ToggleButton) findViewById(R.id.toggleButton2);
        btnDisplay = (Button) findViewById(R.id.btnDisplay);
        btnDisplay.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                StringBuffer result = new StringBuffer();
                result.append("toggleButton1 : ").append(toggleButton1.getText());
                result.append("\ntoggleButton2 : ").append(toggleButton2.getText());
                Toast.makeText(MainActivity.this, result.toString(),
                        Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

- StringBuffer result : nhận giá trị trả về.
- Result.append : giá trị trả về sẽ ghi nối vào.
- Toast.makeText(...).show() : xuất ra khung dữ liệu.

5. Run ứng dụng

- Khi Click vào Dispaly □ kết quả giá trị hiển thị được nhận từ giá của hai ToggleButton.



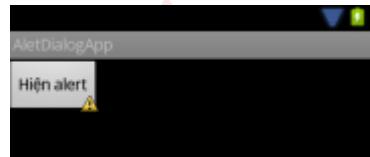


ALERTDIALOG

1. Tạo project mới

2. Tạo layout

- Tạo button cho layout



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

    <Button
        android:id="@+id/buttonAlert"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hiện alert" />

</LinearLayout>
```

3. Viết code cho MainActivity.java trong src

```
public class MainActivity extends Activity {
    final Context context = this;
    private Button button;
```



```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    button = (Button) findViewById(R.id.buttonAlert);
    button.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(context);
            // Dat title
            alertDialogBuilder.setTitle("Đây là title");
            // Dat dialog
            alertDialogBuilder
                .setMessage("Chọn Yes để thoát!")
                .setCancelable(false)
                .setPositiveButton("Yes",
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id)
                        {
                            MainActivity.this.finish(); //đóng ứng dụng
                        }
                    })
                .setNegativeButton("No",
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog,int id)
                        {
                            dialog.cancel(); //đóng dialog
                        }
                    });
            // Tao alertDialog
            AlertDialog alertDialog = alertDialogBuilder.create();
            // hien alertDialog
            alertDialog.show();
        }
    });
}
```

4. Kết quả chạy thử

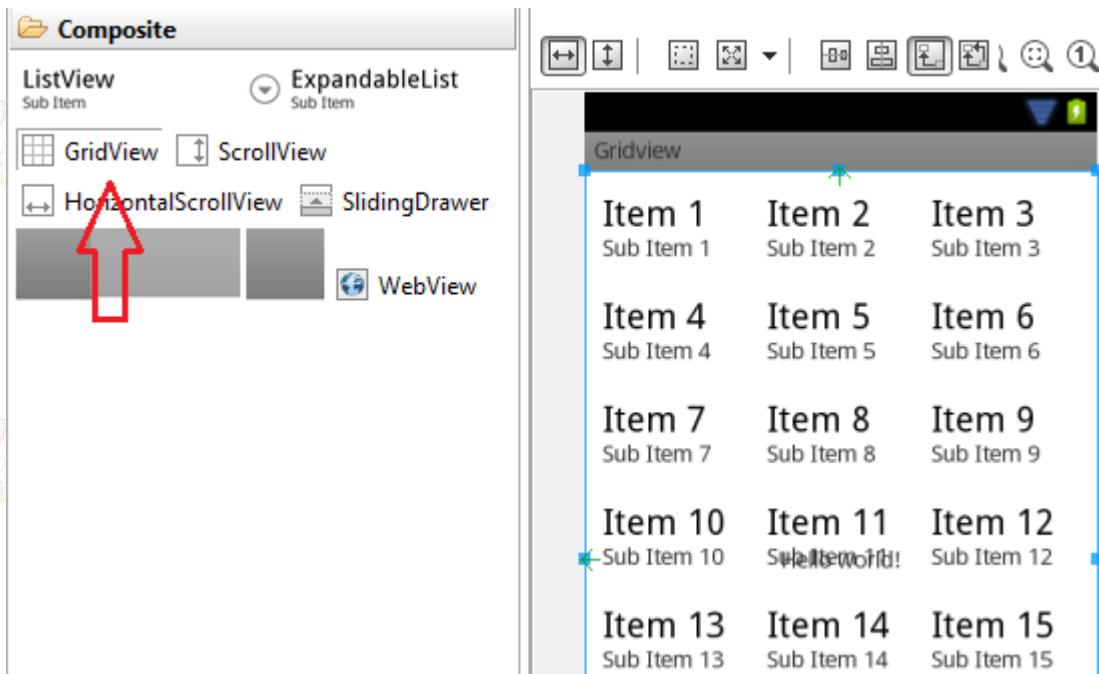




GRIDVIEW

1. Tạo project mới

2. Tạo đối tượng GridView cho layout



3. Viết code cho MainActivity.java trong src

```
public class MainActivity extends Activity {  
  
    GridView gridView;  
  
    static final String[] numbers = new String[] {  
        "A", "B", "C", "D", "E",  
        "F", "G", "H", "I", "J",  
        "K", "L", "M", "N", "O",  
        "P", "Q", "R", "S", "T",  
        "U", "V", "W", "X", "Y", "Z"};
```



```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    gridView = (GridView) findViewById(R.id.gridView1);

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

    gridView.setAdapter(adapter);

    gridView.setOnItemClickListener(new OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View v,
            int position, long id) {
            Toast.makeText(getApplicationContext(),
                ((TextView) v).getText(), Toast.LENGTH_SHORT).show();
        }
    });
}
```

4. Chạy thử ứng dụng

MainActivity						
A	B	C	D	E	F	
G	H	I	J	K	L	
M	N	O	P	Q	R	
S	T	U	V	W	X	
Y	Z					

- Khi click vào một ký tự thì



MainActivity

A B C D E F

G H I J K L

M N O P Q R

S T U V W X

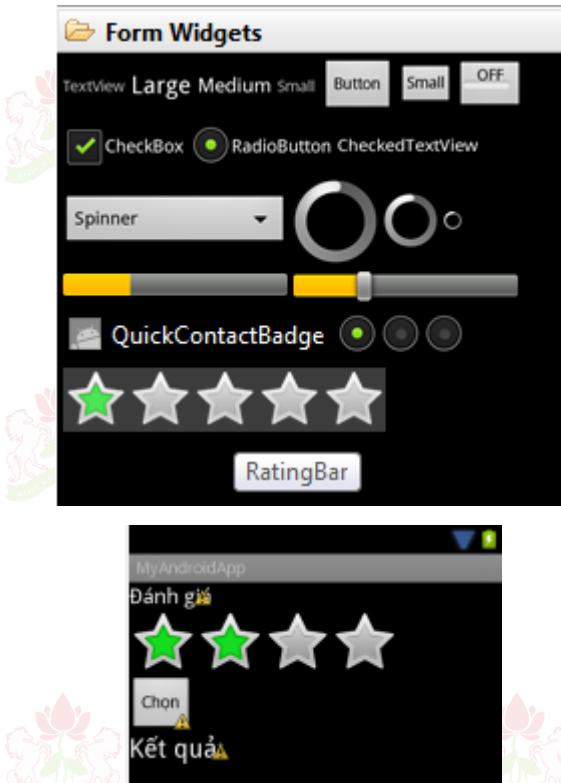
Y Z

T



RATINGBAR

1. Tao project mới
2. Tạo các đối tượng cho layout



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/LblRateMe"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Đánh giá"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <RatingBar
        android:id="@+id/ratingBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:numStars="4"
        android:stepSize="1.0"
        android:rating="2.0" />

    <Button
        android:id="@+id/btnSubmit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Chon" />
```



```
<LinearLayout
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/lblResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Kết quả: "
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:id="@+id/txtRatingValue"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceSmall" />

</LinearLayout>
</LinearLayout>
```

3. Viết code cho Activity

```
public class MyAndroidAppActivity extends Activity {

    private RatingBar ratingBar;
    private TextView txtRatingValue;
    private Button btnSubmit;

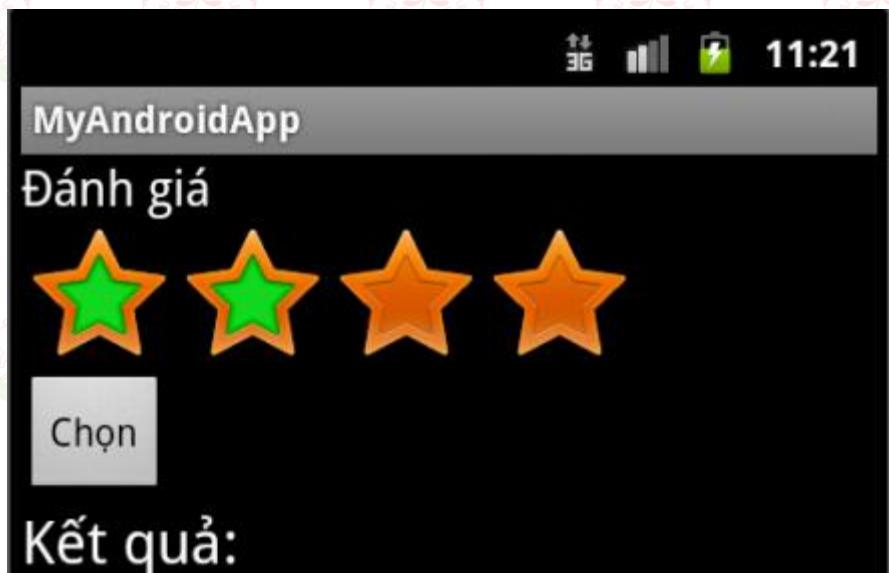
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        addListenerOnRatingBar();
        addListenerOnButton();
    }
}
```



```
public void addListenerOnRatingBar() {  
  
    ratingBar = (RatingBar) findViewById(R.id.ratingBar);  
    txtRatingValue = (TextView) findViewById(R.id.txtRatingValue);  
  
    ratingBar.setOnRatingBarChangeListener(new OnRatingBarChangeListener() {  
        public void onRatingChanged(RatingBar ratingBar, float rating,  
                                    boolean fromUser) {  
            txtRatingValue.setText(String.valueOf(rating));  
        }  
    });  
  
    public void addListenerOnButton() {  
  
        ratingBar = (RatingBar) findViewById(R.id.ratingBar);  
        btnSubmit = (Button) findViewById(R.id.btnSubmit);  
  
        btnSubmit.setOnClickListener(new OnClickListener() {  
  
            public void onClick(View v) {  
  
                Toast.makeText(MyAndroidAppActivity.this,  
                               String.valueOf(ratingBar.getRating()),  
                               Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}  
}
```

4. Kết quả chạy thử





Đánh giá



Chọn

Kết quả: 3.0

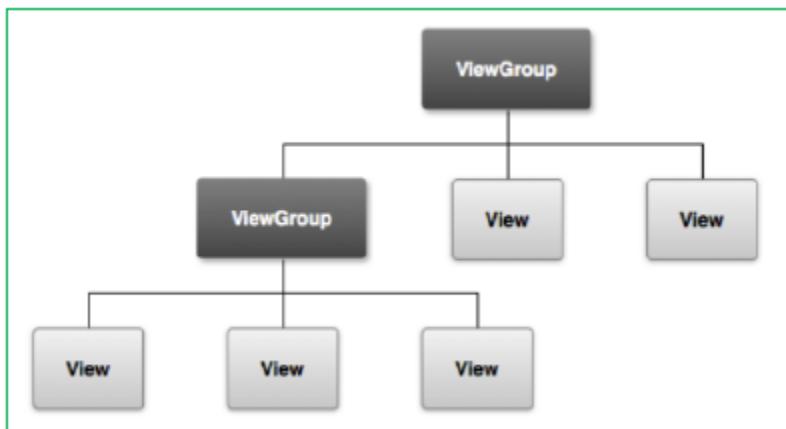
3.0



LAYOUT

1. Khái niệm layout

Giao diện người dùng của Android được định nghĩa như một cây phân cấp của các View và ViewGroup. ViewGroup là một container có thể chứa các thành phần khác và là lớp cơ sở cho các class Layout kế thừa từ nó.



Android cung cấp nhiều loại layout cho phép chứa đựng và tổ chức các thành phần của một activity như button, checkbox, ... và những view khác.

Các layout cho phép kết hợp và lồng vào nhau để tạo nên giao diện phức tạp và đẹp hơn.

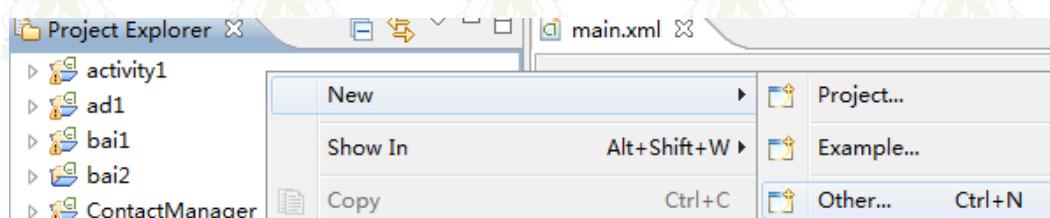
Các layout phần lớn được định nghĩa trong các file XML. Mặc định Eclipse phát sinh ra 1 layout : res/layout/main.xml dùng cho Activity đầu tiên.

Ta có thể tự định ra các layout mới cho một Activity mới, và trong file java của Activity ta dùng setContentView(R.layout.tenlayout) để chỉ định Activity sẽ dùng file .xml nào làm Layout cho nó.

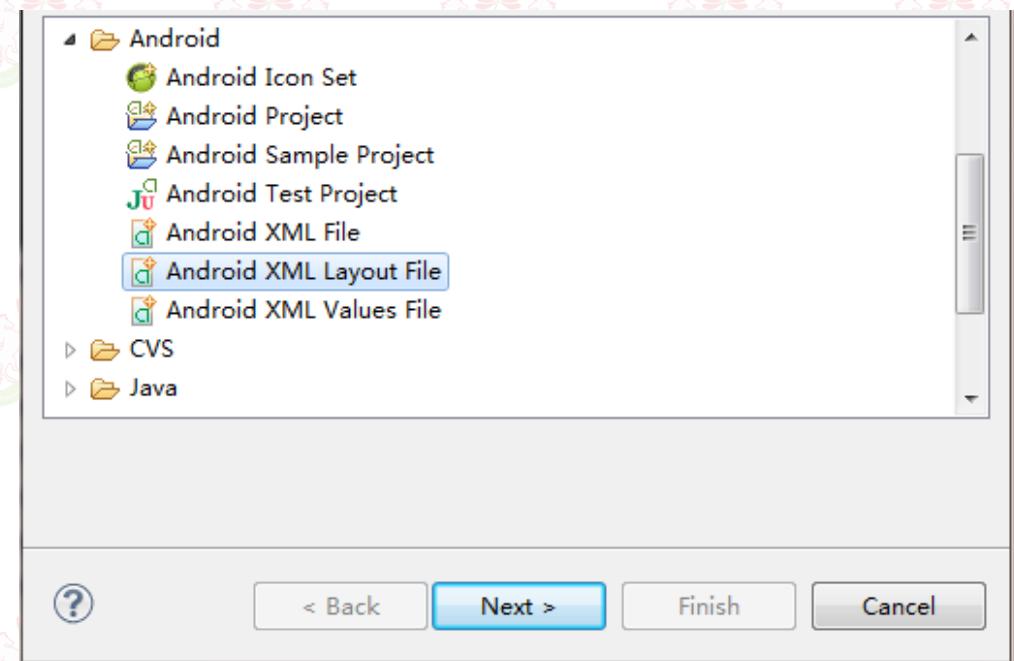
Có một số layout cơ bản như: LinearLayout, RelativeLayout, TableLayout....

2. Cách tạo một layout mới

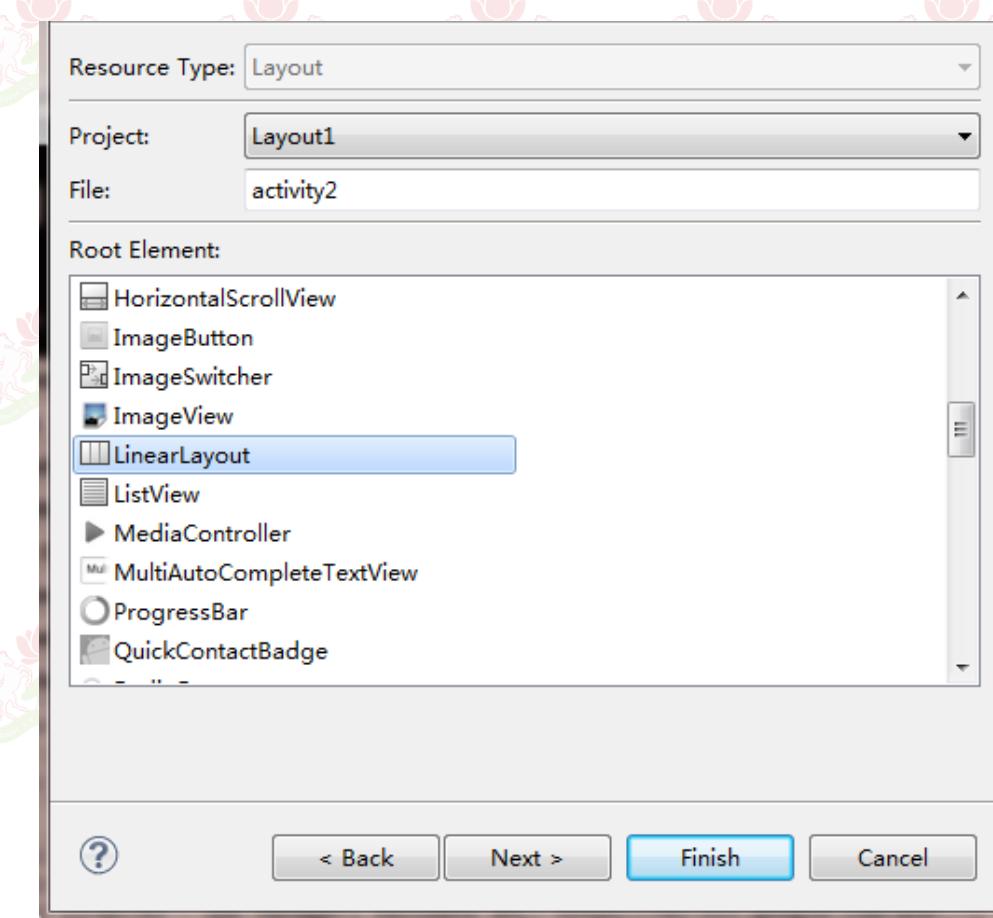
- Click phải chuột vào cửa sổ “Project Explorer”
- Chọn New, Other ...



- Chọn Android XML Layout File, chọn Next



- Chọn tên Project mà layout được tạo thêm
- Đặt tên cho layout (chú ý: dùng chữ thường)
- Chọn loại layout (ví dụ như: LinearLayout) hoặc element nào đó
- Chọn Finish



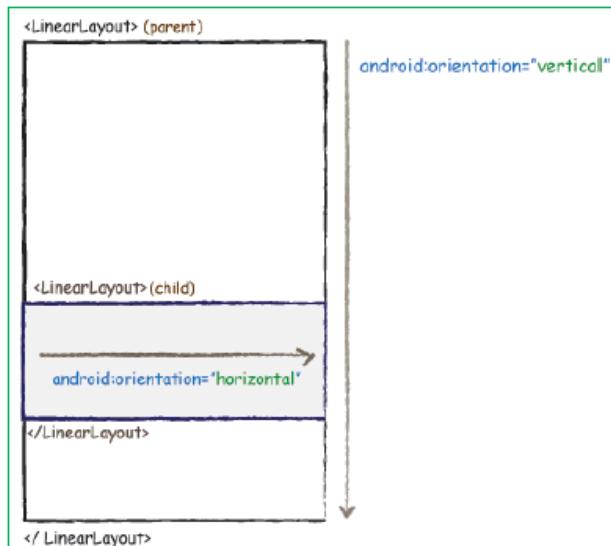


3. Các loại Layout

3.1 Linear Layout

Giống như tên gọi của nó, LinearLayout cho phép các thành phần bên trong nó hiển thị tuyến tính hoặc theo chiều ngang (Horizontally) hoặc theo chiều dọc (Vertically).

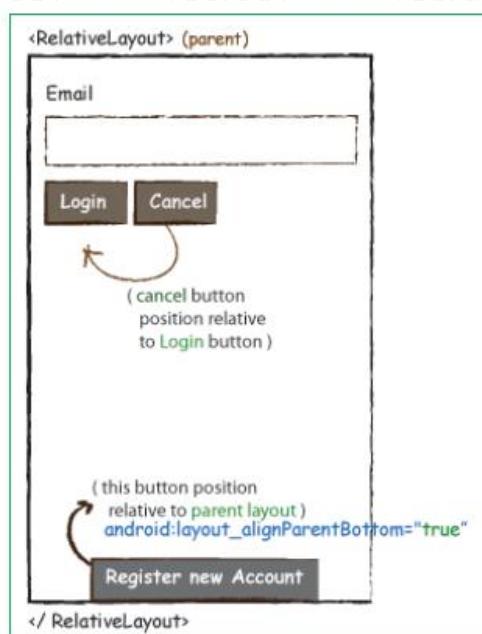
Chiều ngang hay dọc của LinearLayout được quyết định nhờ thuộc tính android:orientation trong thẻ LinearLayout.



3.2 Relative Layout

Trong Relative Layout mỗi thành phần được định vị tương đối với cách thành phần khác hoặc với thành phần cha.

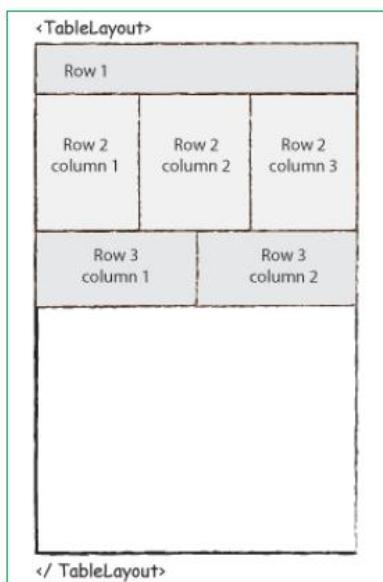
Vd: hình bên dưới



Đầu tiên là `textView` “Email”, tiếp theo là một `EditText` có vị trí “`layout_below`” so với `TextView` “Email”. Nút “Cancel” nằm tương đối bên “`layout_toRightOf`” của nút “Login” và nút “Register new Account” có vị trí là “`layout_alignParentBottom`”.

3.3 Table Layout

`TableLayout` làm việc giống cách `table layout` của `HTML`. Ta có thể chia layout ra theo dòng và cột. Nhưng khác `HTML` là nó chỉ dùng thẻ `TableRow` để chia dòng.

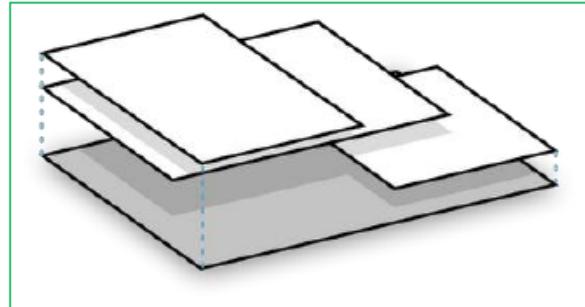
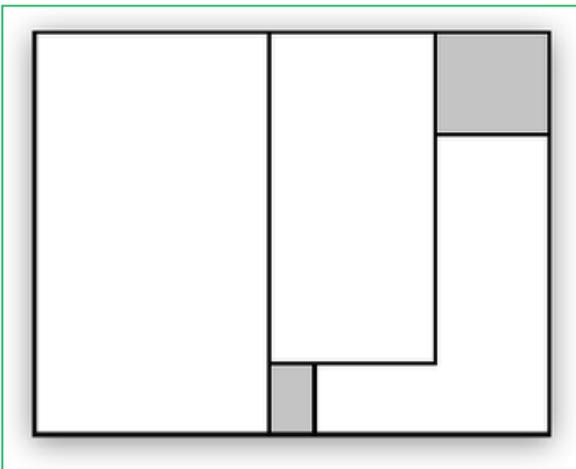


3.4 Frame Layout

Cách bố trí trong `FrameLayout` theo từng khung (frame) nghĩa là các thành phần bên trong layout này có thể chồng lên nhau.

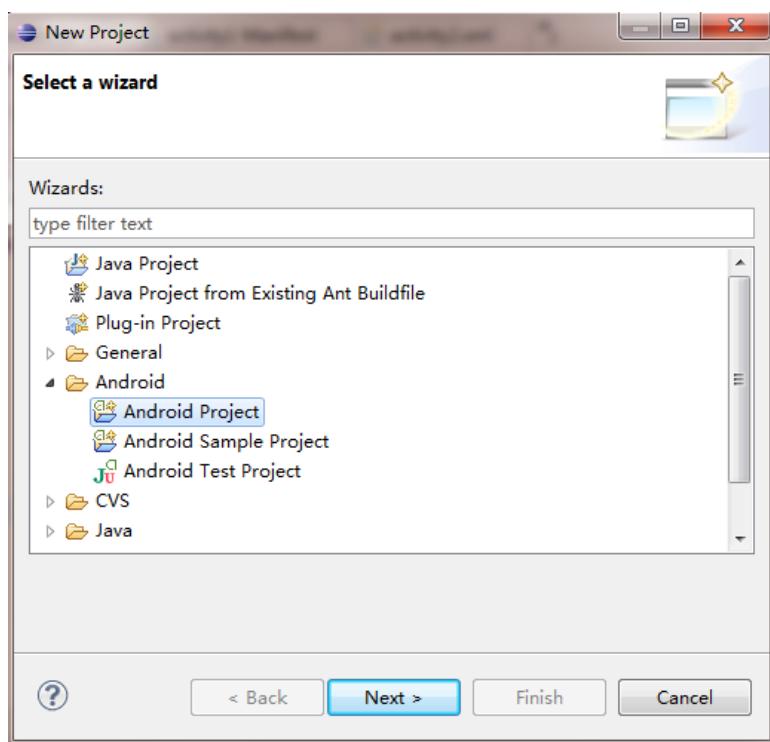
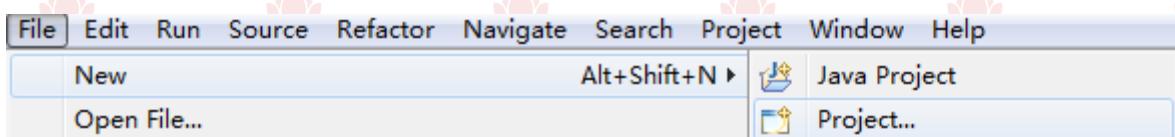


Khi các thành phần chồng lên nhau thì dựa vào vị trí để biết thành phần nào nằm trên thành phần nào. Nghĩa là theo mã lệnh thành phần viết trước sẽ bị chồng lên bởi thành phần viết sau và cứ tiếp tục như thế.



4. Ví dụ (Tablelayout)

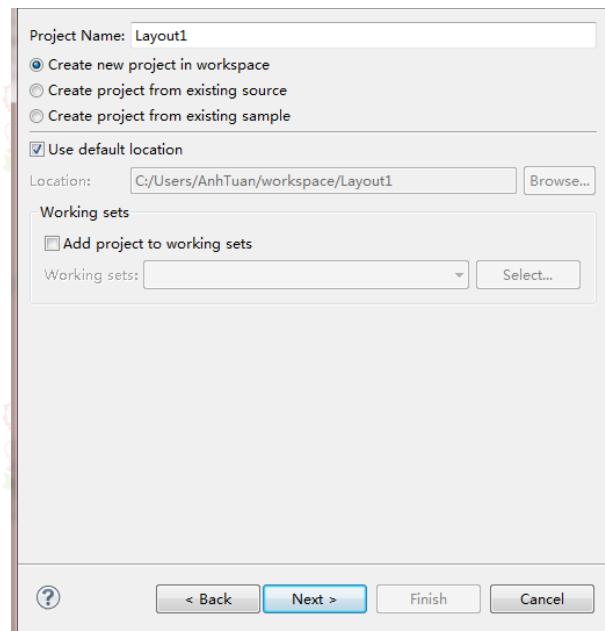
- Bước 1: Tạo project mới
 - + Chọn File, New, Project ...
 - + Chọn Android Project và chọn Next



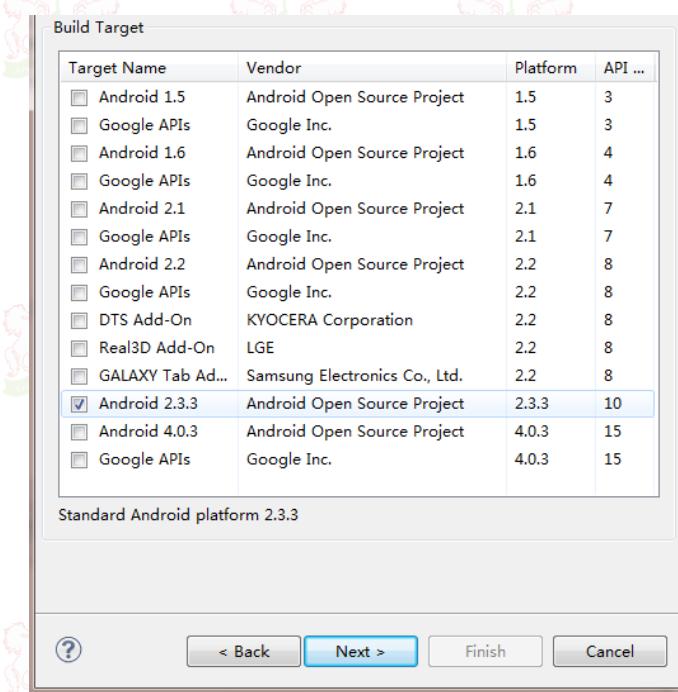


- + Tại Project name: Đặt tên cho project
- + Tại Location: Chọn thư mục chứa project hoặc chọn thư mục mặc định

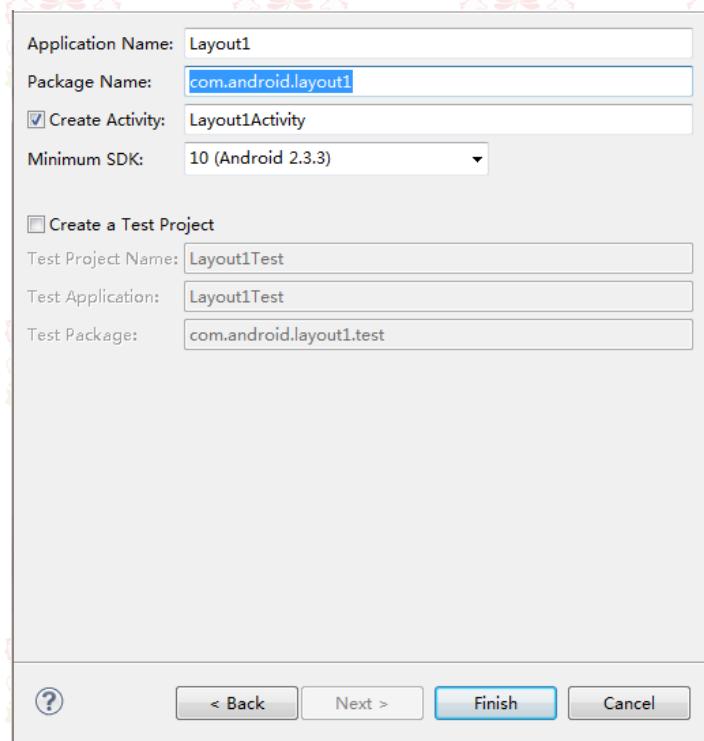
Sau đó chọn Next



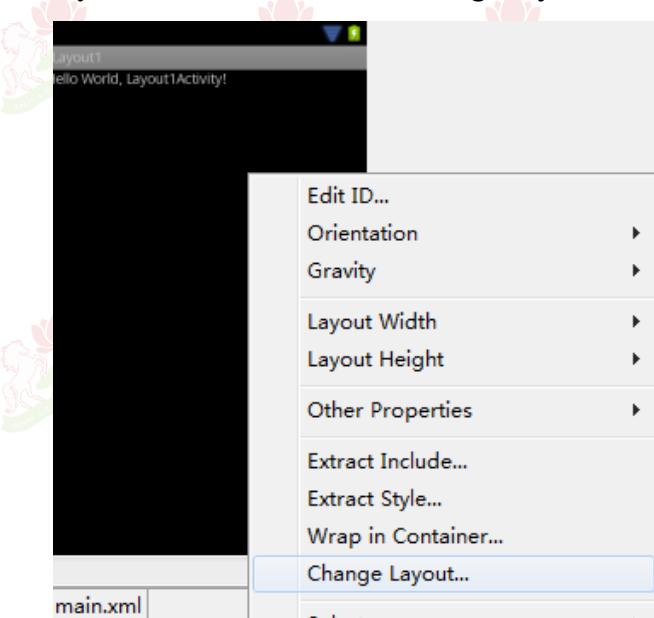
- + Chọn phiên bản (version) của Android và chọn Next



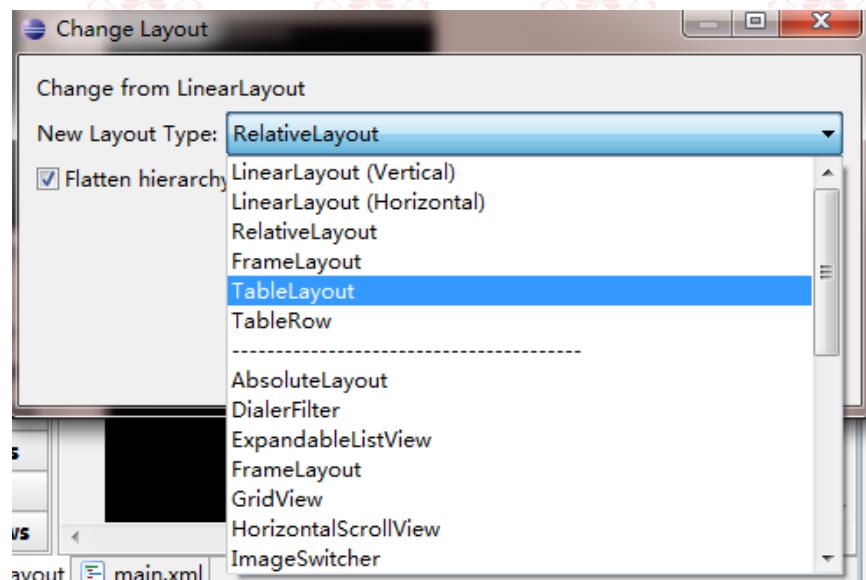
- + Tại Package Name: Gõ vào class cho ứng dụng, sau đó chọn Finish



- Bước 2: Thay đổi layout main.xml
 - + Mở layout main.xml
 - + Click phải chuột trên layout main.xml và chọn change layout...



- + Chọn loại layout muốn thay đổi và chọn OK



- + Thay đổi layout như sau:

```
<?xml version="1.0" encoding="utf-8"?>

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:id="@+id/tableLayout1"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent" >

    <!-- 2 columns -->

    <TableRow

        android:id="@+id/tableRow1"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:padding="5dip" >

        <TextView

            android:id="@+id/textView1"

            android:text="Column 1"

            android:textAppearance="?android:attr/textAppearanceLarge" />

        <Button

            android:id="@+id/button1"

            android:text="Column 2" />

    </TableRow>
```



<!-- edittext span 2 column -->

<TableRow>

```
    android:id="@+id/tableRow2"  
  
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
  
    android:padding="5dip" >
```

<EditText>

```
        android:id="@+id/editText1"  
  
        android:layout_span="2"  
  
        android:text="Column 1 & 2" />
```

</TableRow>

<!-- just draw a red line -->

<View>

```
    android:layout_height="2dip"  
  
    android:background="#FF0000" />
```

<!-- 3 columns -->

<TableRow>

```
    android:id="@+id/tableRow3"  
  
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
  
    android:padding="5dip" >
```

<TextView>

```
        android:id="@+id/textView2"  
  
        android:text="Column 1"  
  
        android:textAppearance="?android:attr/textAppearanceLarge" />
```

<Button>

```
        android:id="@+id/button2"
```



```
        android:text="Column 2" />
```

```
<Button
```

```
    android:id="@+id/button3"
```

```
    android:text="Column 3" />
```

```
</TableRow>
```

```
<!-- display this button in 3rd column via layout_column(zero based) -->
```

```
<TableRow
```

```
    android:id="@+id/tableRow4"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:padding="5dip" >
```

```
<Button
```

```
    android:id="@+id/button4"
```

```
    android:layout_column="2"
```

```
    android:text="Column 3" />
```

```
</TableRow>
```

```
<!-- display this button in 2nd column via layout_column(zero based) -->
```

```
<TableRow
```

```
    android:id="@+id/tableRow5"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:padding="5dip" >
```

```
<Button
```

```
    android:id="@+id/button5"
```

```
    android:layout_column="1"
```

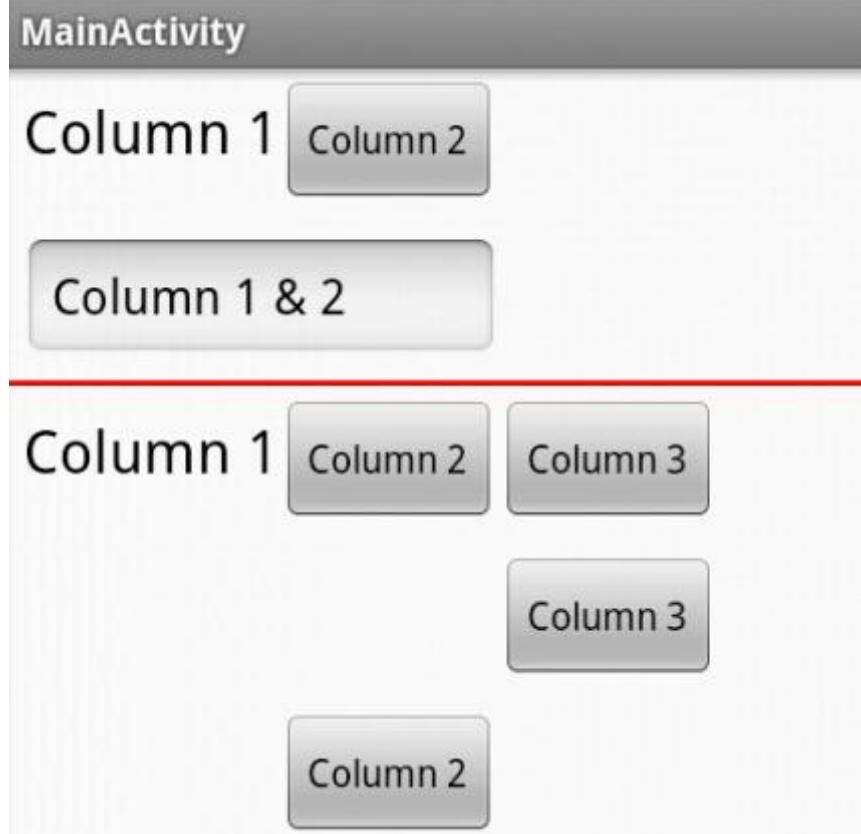
```
    android:text="Column 2" />
```

```
</TableRow>
```

```
</TableLayout>
```



- + Kết quả khi chạy



5. Ví dụ (Relativelayout)

```
<?xml version="1.0" encoding="utf-8"?>

<?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" >

    <Button

        android:id="@+id	btnButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"/>

    <Button

        android:id="@+id	btnButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 2"/>

```



```
        android:text="Button 2"

        android:layout_toRightOf="@+id	btnButton1"/>

<Button

    android:id="@+id	btnButton3"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Button 3"

    android:layout_below="@+id	btnButton1"/>

<TextView

    android:id="@+id	textView1"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_below="@+id	btnButton3"

    android:layout_marginTop="94dp"

    android:text="User :"

    android:textAppearance="?android:attr/textAppearanceLarge" />

<EditText

    android:id="@+id/editText1"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignParentRight="true"

    android:layout_alignTop="@+id/textView1"

    android:layout_toRightOf="@+id	btnButton3" />

<Button

    android:id="@+id	btnSubmit"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignParentRight="true"

    android:layout_below="@+id/editText1"
```



```
        android:text="Submit" />  
</RelativeLayout>
```

- Khi chạy kết quả:





MENU

1. Menu

Là một thành phần giao diện người dùng quan trọng trong nhiều loại ứng dụng.

Trong Android có 3 dạng Menu là Option Menu, Context Menu và Popup Menu.

Ta có thể định nghĩa menu trong XML hoặc trong code java nhưng nên dùng XML vì:

- Trong XML sẽ dễ dàng hình dung cấu trúc của menu.
- Tách riêng phần layout trong file XML và phần xử lý trong file java sẽ làm mã lệnh dễ nhìn hơn.

2. Định nghĩa Menu trong XML

Android cung cấp một định dạng XML chuẩn để định nghĩa cho tất cả loại menu.

Dùng XML định nghĩa menu và tất cả những mục con bên trong nó trong một file resource. Sau đó ta có thể nạp nó như một đối tượng Menu vào trong Activity.

Để định nghĩa menu, tạo 1 file XML trong thư mục res/menu/ và xây dựng với các thẻ:

- <menu> : Là node gốc của file dùng định nghĩa một menu. Bên trong nó là các thẻ <item> và <group>
- <item>: Dùng định nghĩa một mục con của menu. Nó có thể chứa một thẻ <menu> bên trong để tạo các menu con.
- <group>: Là thẻ tùy chọn dùng để nhóm các mục con.

Ví dụ:

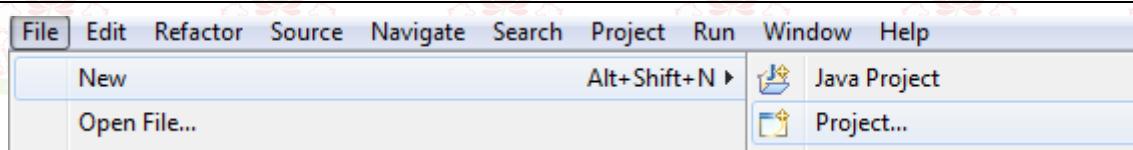
```
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:id="@+id/menu_insert"
          android:title="Thêm" />
    <item android:id="@+id/menu_update"
          android:title="Sửa" />
    <item android:id="@+id/menu_delete"
          android:title="Xóa" />

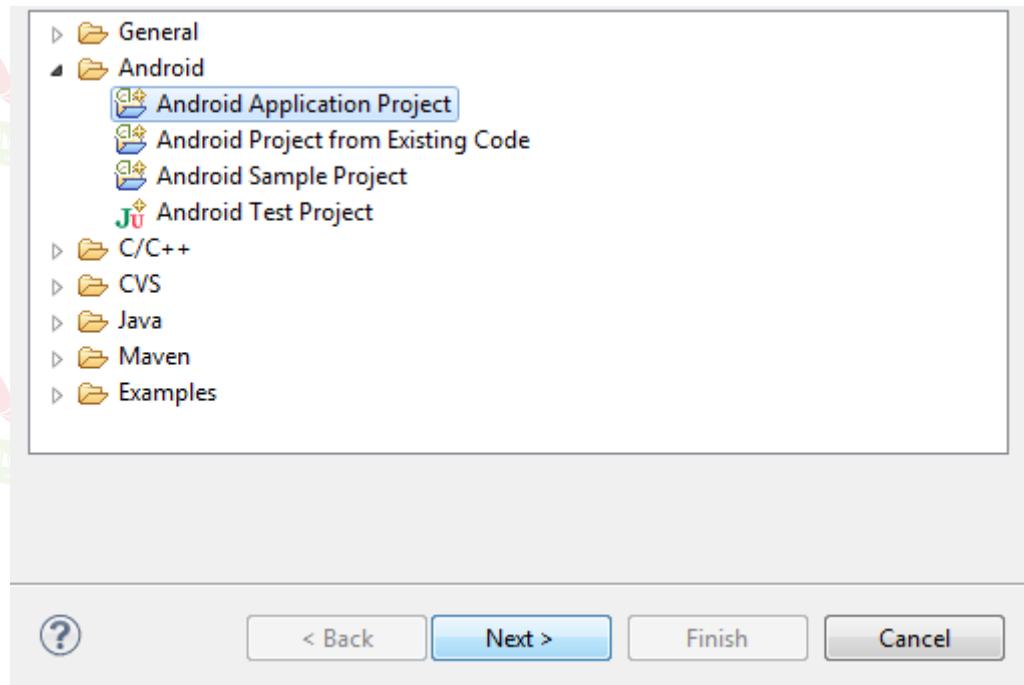
</menu>
```

3. Option menu

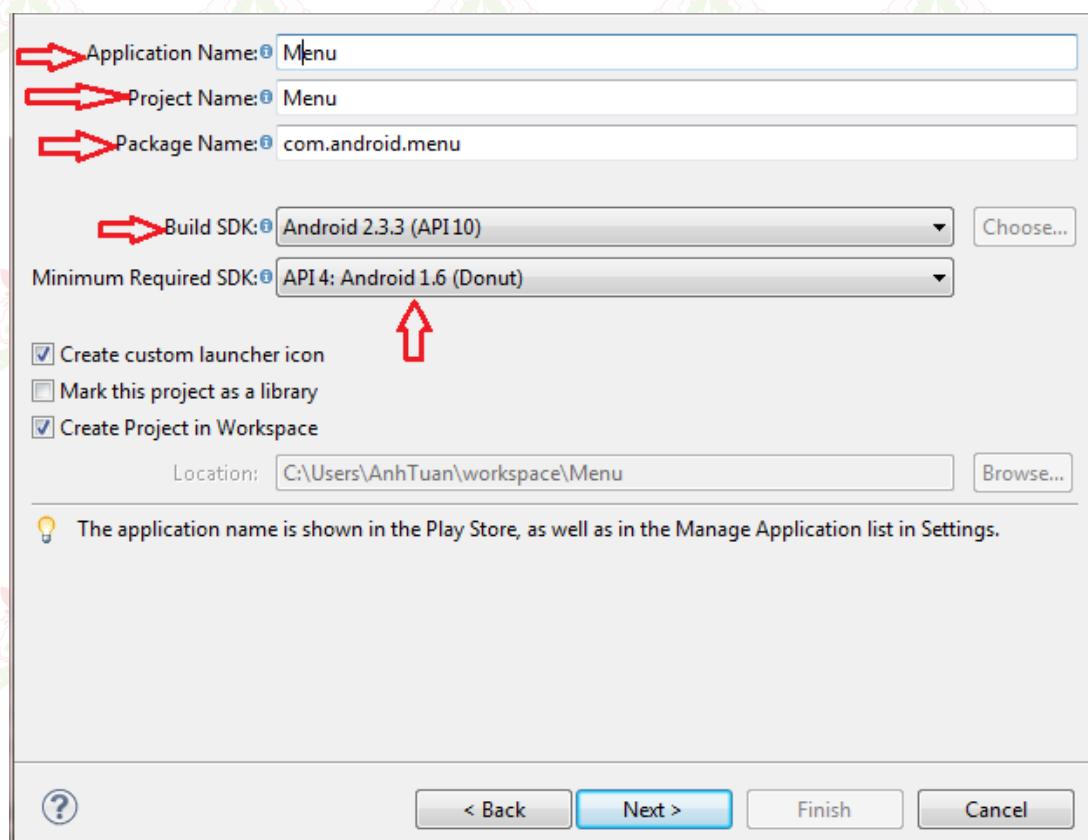
- Bước 1: Tạo project mới
 - + File, New, Project



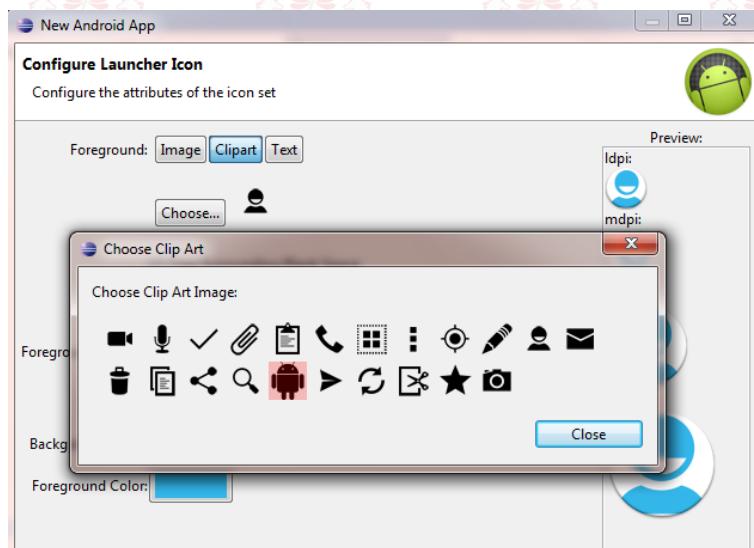
- + Chọn Android Application Project và chọn Next



- + Gõ vào và chọn các tham số thích hợp, sau đó chọn next



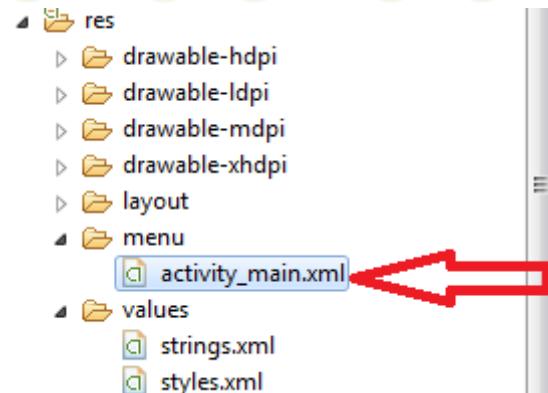
- + Chọn icon cho ứng dụng



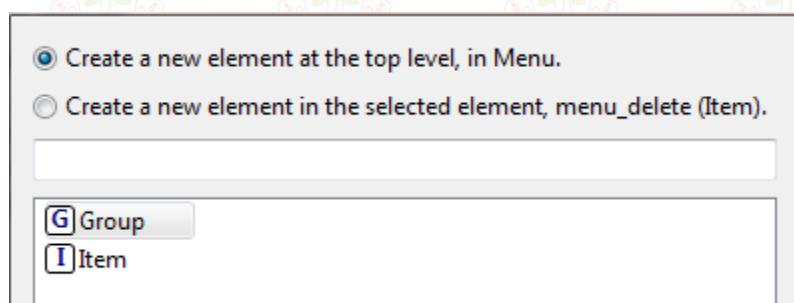
- Bước 2: Tạo menu

- + Tạo file .xml (activity_main.xml trong res/menu)

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
  
    <item android:id="@+id/menu_insert"  
        android:title="Thêm" />  
    <item android:id="@+id/menu_update"  
        android:title="Sửa" />  
    <item android:id="@+id/menu_delete"  
        android:title="Xóa" />  
  
</menu>
```

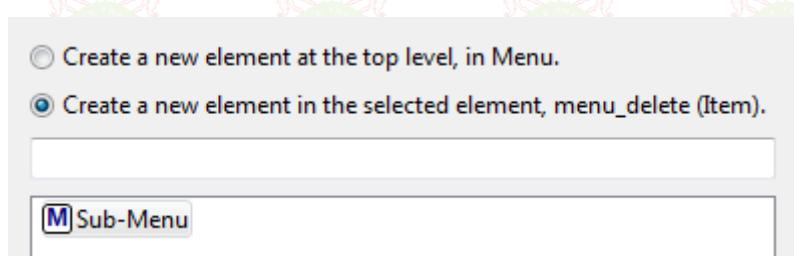


- + Hoặc dùng giao diện tạo menu như sau:
- + chọn nút Add...
- + Tạo menu cùng cấp





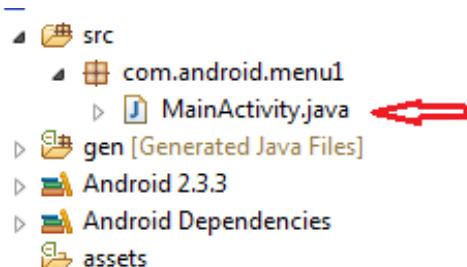
- + Tạo menu con



- + Nhập các thuộc tính vào

Id	@+id/menu_delete	Browse...
Menu category	▼	
Order in category		
Title	Xóa	Browse...
Title condensed	Browse...	
Icon		
Alphabetic shortcut	Browse...	
Numeric shortcut	Browse...	
Checkable	▼	
Checked	▼	
Visible	▼	
Enabled	▼	

- Bước 3: Viết code trong file MainActivity.java trong thư mục src





```
package com.android.menu1;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

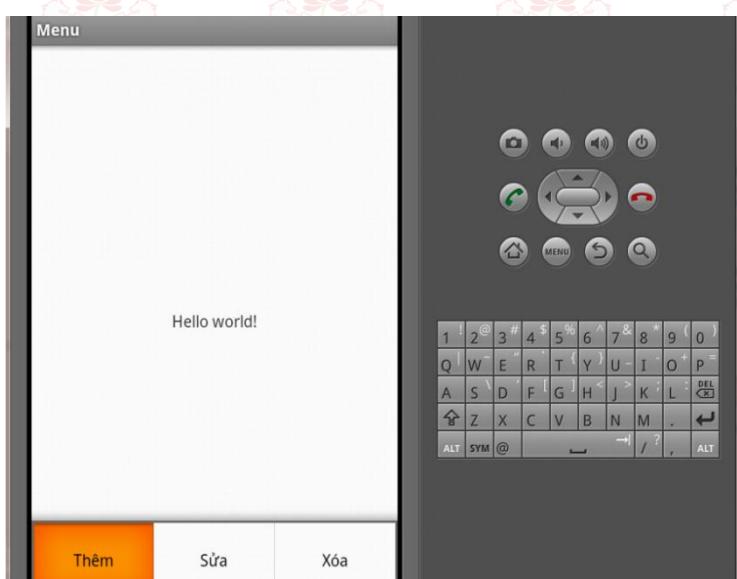
public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return super.onCreateOptionsMenu(menu);
    }
}
```



- Bước 4: Chạy thử



- Bước 5: Viết thêm đoạn code sau

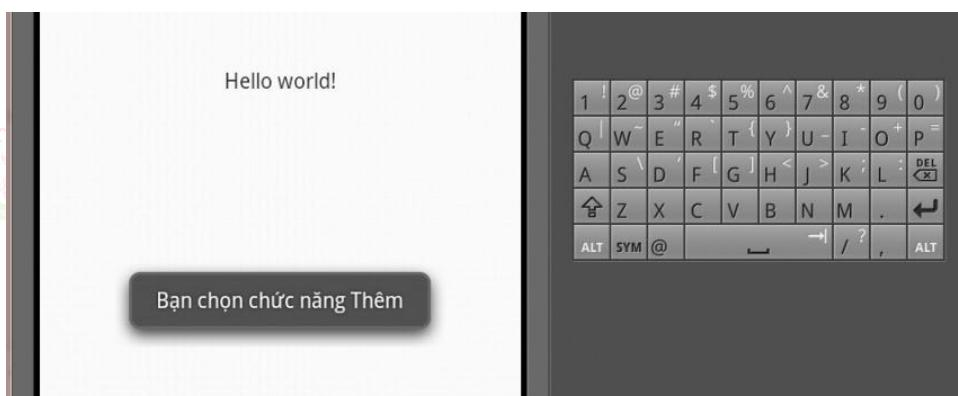


```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.activity_main, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item){
    String mess="";
    switch (item.getItemId()){
        case R.id.menu_insert: mess="Bạn chọn chức năng Thêm";
            break;
        case R.id.menu_update: mess="Bạn chọn chức năng Sửa";
            break;
        case R.id.menu_delete: mess="Bạn chọn chức năng Xóa";
            break;
    }
    Toast.makeText(MainActivity.this, mess, Toast.LENGTH_SHORT).show();
    return super.onOptionsItemSelected(item);
}
```

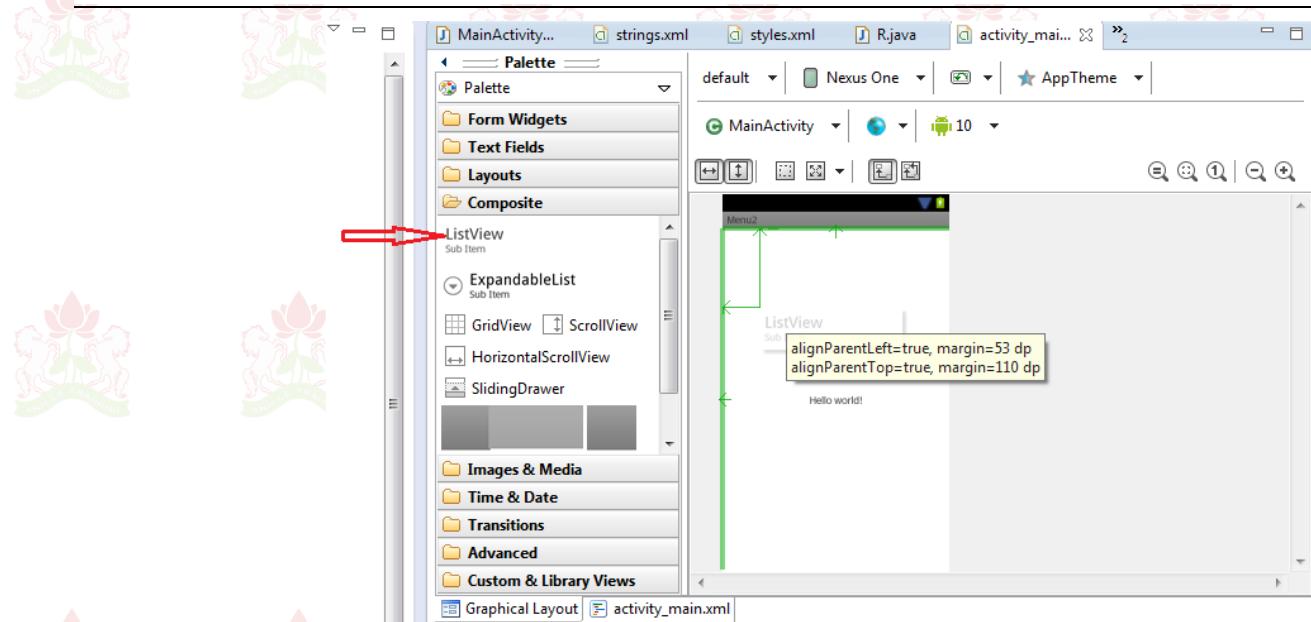


- Bước 6: Chạy thử ứng dụng
 - + Khi chọn vào menu “Thêm”

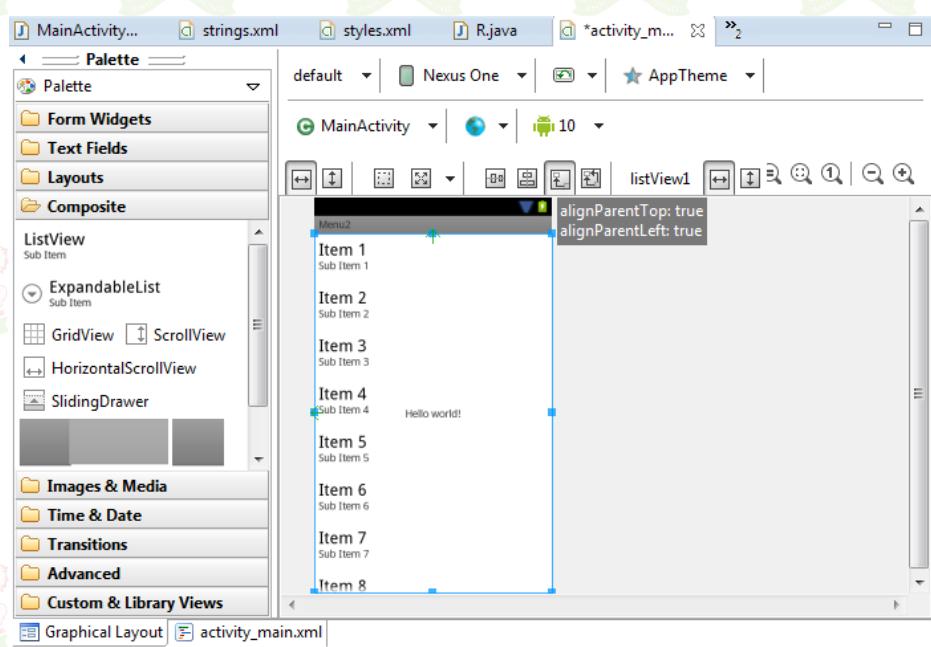


4. Context menu

- Bước 1: Tạo project
- Bước 2: Tạo ListView
 - + Chọn ListView trong “Composite” và kéo sang Layout



- + Hiện chỉnh ListView lại như sau



- Bước 3: Thêm đoạn code sau



```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.ListView;

public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ListView lv=(ListView) findViewById(R.id.listView1);
        String[] list=new String[] {"Apple","Google","Microsoft","FaceBook"};
        BaseAdapter adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,list);
        lv.setAdapter(adapter);
    }
}
```

và chạy thử ứng dụng

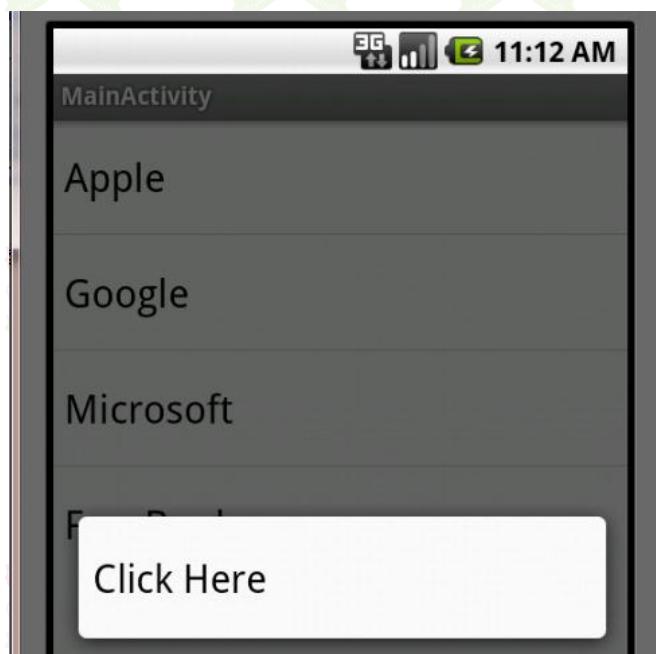


- Bước 4: Tiếp tục thêm đoạn code sau



```
registerForContextMenu(lv);  
}  
  
@Override  
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuItemInfo menuInfo)  
{  
    super.onCreateContextMenu(menu, v, menuInfo);  
    menu.add("Click Here");  
}  
  
@Override  
public boolean onContextItemSelected(MenuItem item){  
  
    super.onContextItemSelected(item);  
    if (item.getTitle() == "Click Here"){  
        Toast.makeText(MainActivity.this, "Chào bạn!", Toast.LENGTH_SHORT).show();  
    }  
  
    return true;  
}
```

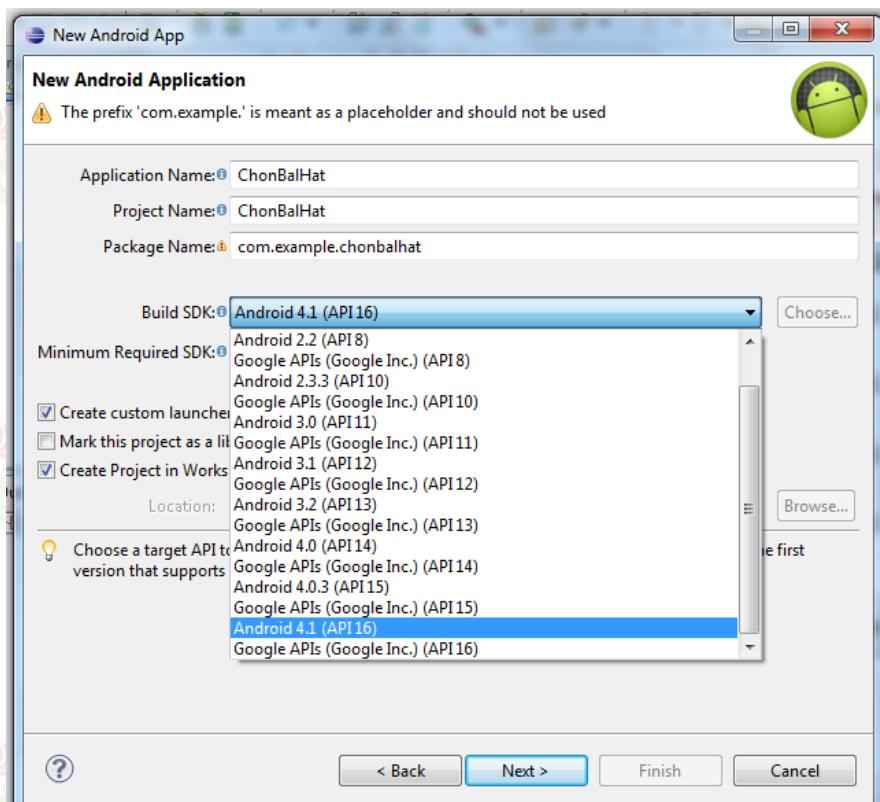
- Bước 5: Chạy ứng dụng



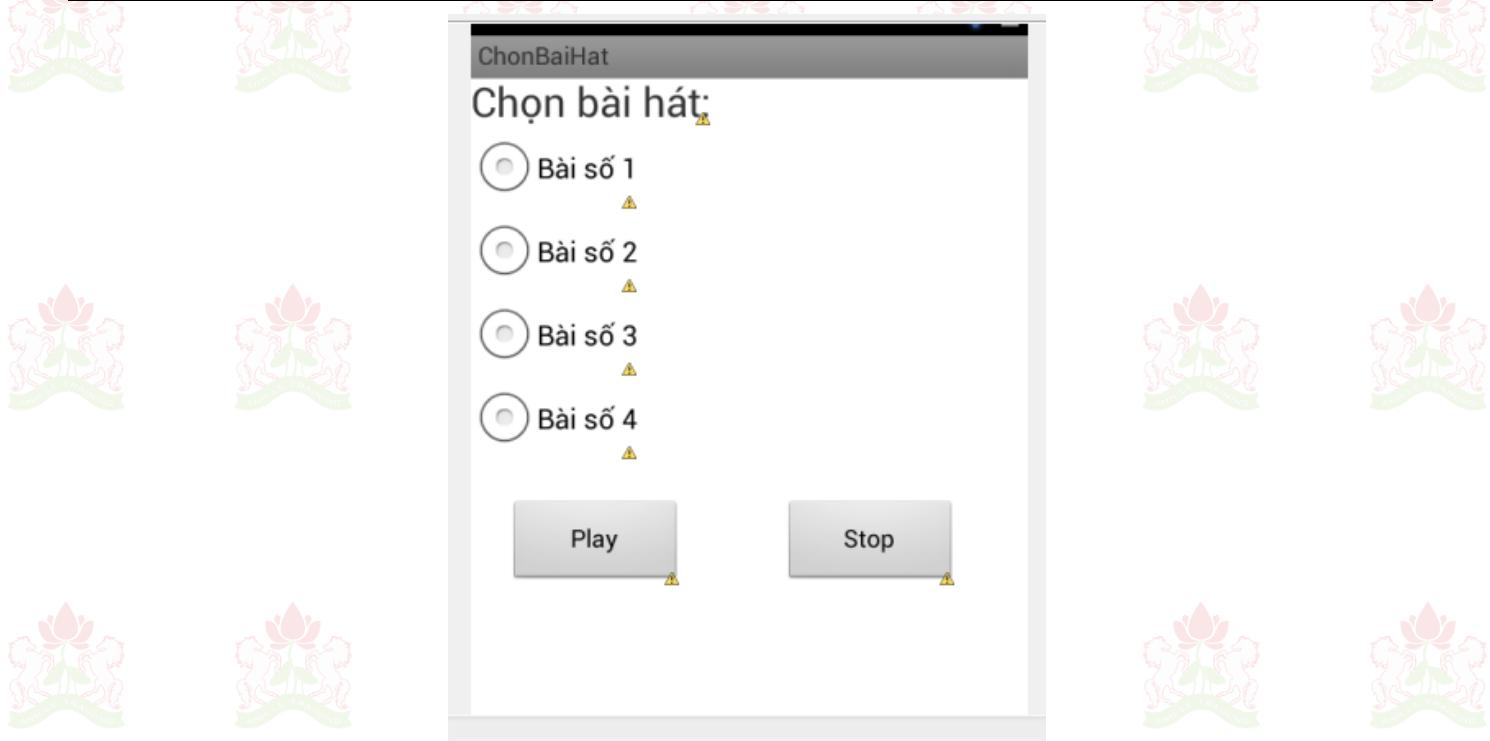


CHỌN VÀ CHƠI NHẠC MP3

- **Bước 1:** Tạo project mới



- **Bước 2:** tạo giao diện như sau:



```
MainActivity.java  main.xml x
<RadioGroup
    android:id="@+id/groupRadio"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:orientation="vertical" >

    <RadioButton
        android:id="@+id/radioButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_below="@+id/textView1"
        android:layout_marginTop="28dp"
        android:text="Bài số 1" />

    <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/radioButton1"
        android:layout_below="@+id/radioButton1"
        android:text="Bài số 2" />

    <RadioButton
        android:id="@+id/radioButton3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/radioButton2"
        android:layout_below="@+id/radioButton2"
        android:text="Bài số 3" />

    <RadioButton
        android:id="@+id/radioButton4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/radioButton3"
        android:layout_below="@+id/radioButton3"
        android:text="Bài số 4" />
```



```
mainActivity.java  main.xml X

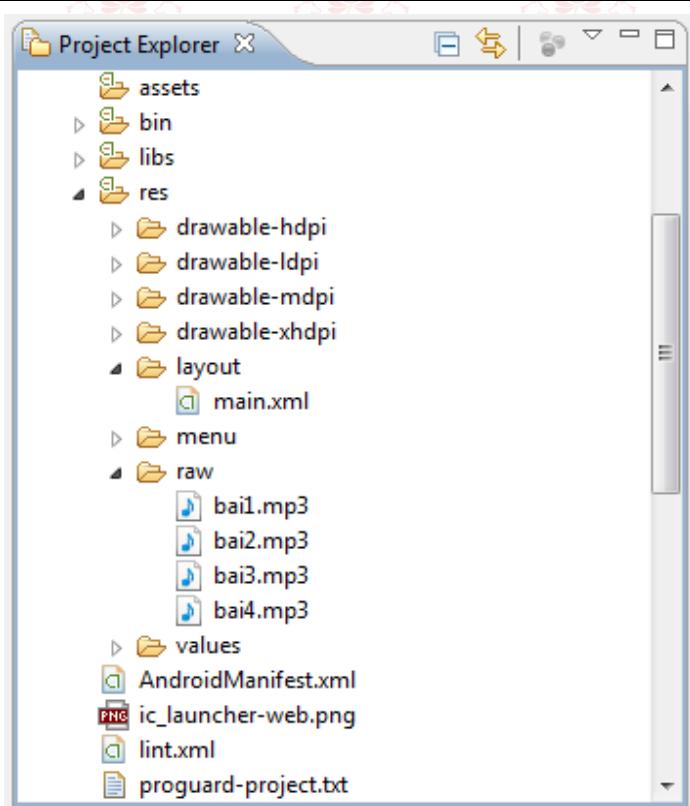
<RadioButton
    android:id="@+id/radioButton4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id.radioButton3"
    android:layout_alignTop="@+id/LinearLayout1"
    android:text="Bài số 4" />
</RadioGroup>

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:text="Chọn bài hát:"
    android:textSize="10pt" />

<Button
    android:id="@+id/button1"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/LinearLayout1"
    android:layout_marginLeft="21dp"
    android:text="Play" />

<Button
    android:id="@+id/button2"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:layout_below="@+id/LinearLayout1"
    android:layout_marginLeft="19dp"
    android:layout_toRightOf="@+id/LinearLayout1"
    android:text="Stop" />
```

- **Bước 3:** Tạo thư mục raw trong res và chép các bài hát vào thư mục này (thư mục raw dùng chứa các file media)



- **Bước 4:** Mở file MainActivity.java và import các thư viện hỗ trợ như trong hình

```
package com.example.chonbaihat;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import android.os.Bundle;
import android.os.Environment;
import android.annotation.TargetApi;
import android.app.Activity;
import android.view.Menu;
import android.media.MediaPlayer;
import android.view.View;
import android.widget.Button;
import android.widget.RadioGroup;
import android.widget.RadioGroup.OnCheckedChangeListener;
import android.widget.Toast;
```

- **Bước 5:** Khai báo các tham số liên quan



```
@TargetApi(8)
public class MainActivity extends Activity implements OnCheckedChangeListener{
    MediaPlayer song1,song2,song3,song4;

    int whatsong;
    int Rsong;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        whatsong=0;
        RadioGroup rgMusic=(RadioGroup) findViewById(R.id.groupRadio);
        rgMusic.setOnCheckedChangeListener(this);

        //Thiet lap cac bai hat
        song1=MediaPlayer.create(this,R.raw.bai1);
        song2=MediaPlayer.create(this,R.raw.bai2);
        song3=MediaPlayer.create(this,R.raw.bai3);
        song4=MediaPlayer.create(this,R.raw.bai4);
        Button bPlay=(Button) findViewById(R.id.button1);
        Button bStop=(Button) findViewById(R.id.button2);
    }
}
```

- + Code thay đổi bài hát khi một nút radio được chọn

```
//thay doi bai hat khi mot nut radio duoc chon

public void onCheckedChanged(RadioGroup arg0, int arg1){
    switch(arg1)
    {
        case R.id.radioButton1:
            whatsong=1;break;
        case R.id.radioButton2: whatsong=2;break;
        case R.id.radioButton3: whatsong=3;break;
        case R.id.radioButton4: whatsong=4;break;
    }
}
```

- + Code cho nút Play:



MainActivity.java X main.xml

```
// Code cho nút Play
bPlay.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        if(song1.isPlaying()){
            song1.pause();
            song1.seekTo(0);
        }
        if(song2.isPlaying()){
            song2.pause();
            song2.seekTo(0);
        }
        if(song3.isPlaying()){
            song3.pause();
            song3.seekTo(0);
        }
        if(song4.isPlaying()){
            song4.pause();
            song4.seekTo(0);
        }
        switch(whatsong){
            case 0: Toast noSong=Toast.makeText(MainActivity.this,"Hay chon bai hat", Toast.LENGTH_SHORT);
                      noSong.show();break;
            case 1: song1.start();break;
            case 2: song2.start();break;
            case 3: song3.start();break;
            case 4: song4.start();break;
        }
    }
});
```

+ Code cho nút Stop:

MainActivity.java X main.xml

```
// Code cho nút Stop
bStop.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        if(song1.isPlaying()){
            song1.pause();
            song1.seekTo(0);
        }else if(song2.isPlaying()){
            song2.pause();
            song2.seekTo(0);
        }else if(song3.isPlaying()){
            song3.pause();
            song3.seekTo(0);
        }else if(song4.isPlaying()){
            song4.pause();
            song4.seekTo(0);
        }
    }
});
```

+ Code để download bài hát về máy:



```
[MainActivity.java] [main.xml]

//Tạo sự kiện nhấn giữ phím để down load bài hát về máy
bPlay.setOnLongClickListener(new View.OnLongClickListener() {
    public boolean onLongClick(View v) {
        // TODO Auto-generated method stub

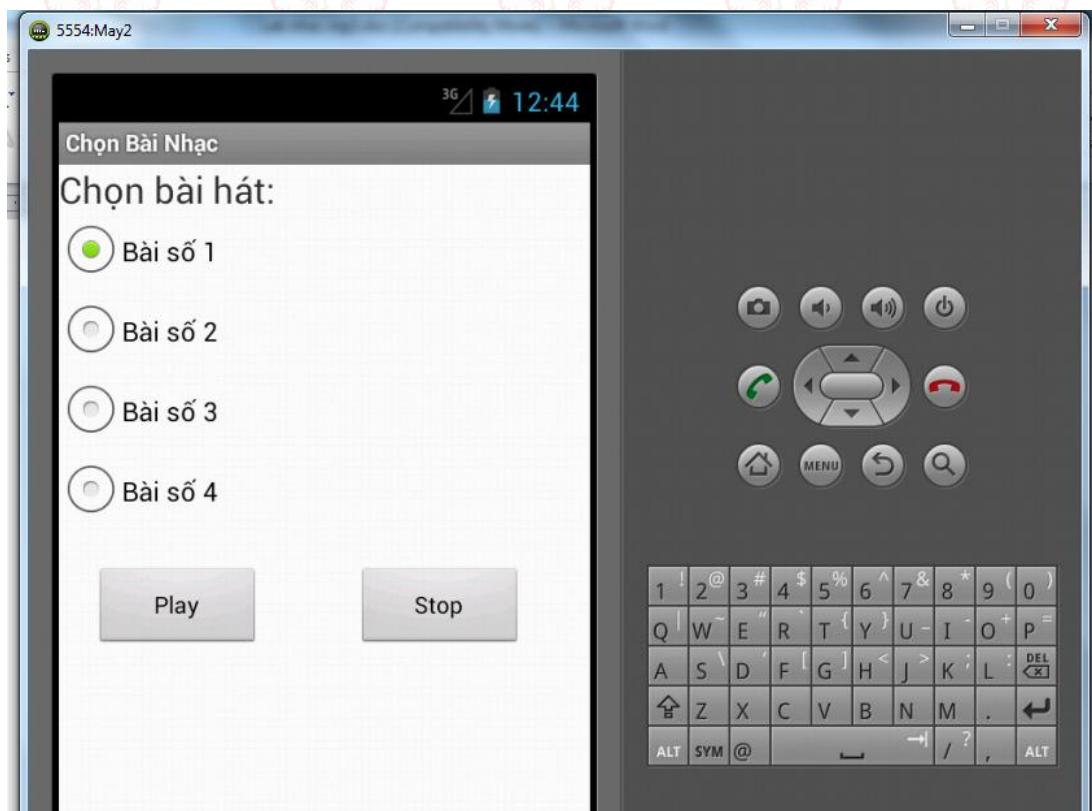
        File path=Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS);
        File file=new File(path,"bai"+whatsong+".mp3");
        try{
            switch(whatsong)
            {
                case 1: Rsong=R.raw.bai1;break;
                case 2: Rsong=R.raw.bai2;break;
                case 3: Rsong=R.raw.bai3;break;
                case 4: Rsong=R.raw.bai4;break;
            }

            InputStream is=getResources().openRawResource(Rsong);
            FileOutputStream os= new FileOutputStream(file);
            byte[] data=new byte[is.available()];
            is.read(data);
            os.write(data);

            is.close();
            os.close();
        }

        }catch(IOException e){
            Toast fail=Toast.makeText(MainActivity.this, "Download that bai!", Toast.LENGTH_SHORT);
            fail.show();
        }
        return false;
    }
});
```

Bước 6: Chạy chương trình để test

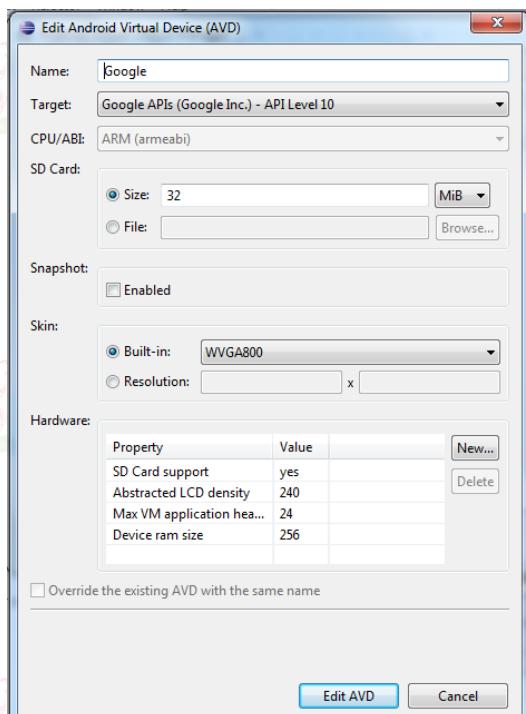




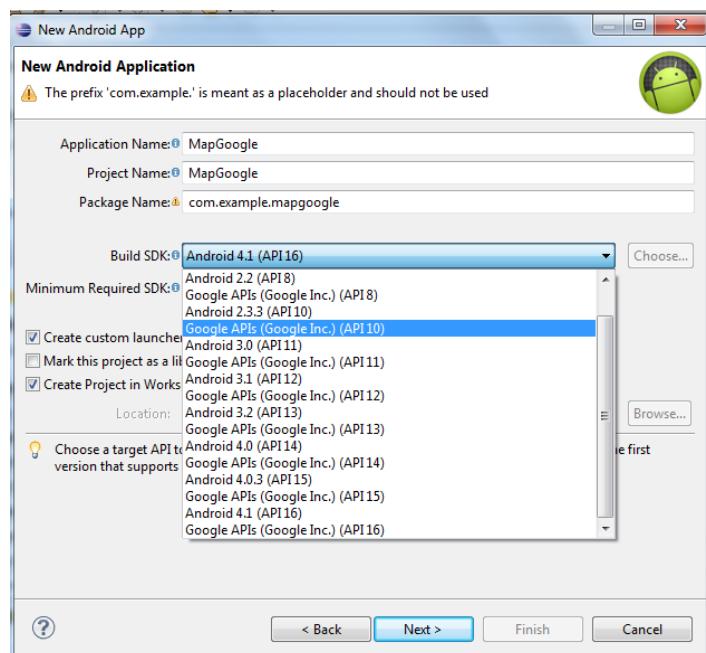
GOOGLE MAP

- **Bước 1:** Tạo máy ảo mới để chạy ứng dụng

Vào AVD Manager tiến hành tạo máy ảo như hình:



- **Bước 2:** Tạo Project mới:



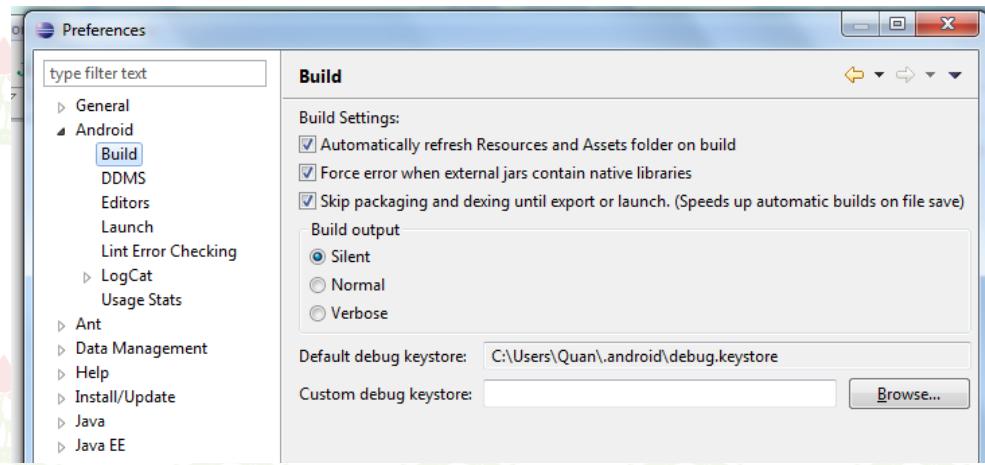
Chọn SDK nhu hình để hỗ trợ các thư viện của google.



- **Bước 3: Đăng ký và lấy key Google Map API**

- + Lấy code MD5 fingerprint trên máy của bạn:

Trước hết, ta phải xác định đường dẫn đến file debug.keystore. Vào Window -> Preferences để lấy đường như hình:



Vào cmd gõ lệnh sau để lấy mã MD5 fingerprint:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Quan>cd c:\
c:>cd Program Files (x86)
c:\Program Files (x86)>cd Java
c:\Program Files (x86)\Java>cd jre6
c:\Program Files (x86)\Java\jre6>cd bin
c:\Program Files (x86)\Java\jre6\bin>keytool -list -alias androiddebugkey -keystore "C:\Users\Quan\.android\debug.keystore" -storepass android -keypass android
androiddebugkey, Feb 28, 2012, PrivateKeyEntry,
Certificate fingerprint (MD5): 3B:72:4D:33:A6:96:49:41:26:7E:22:1A:ED:4D:DD:AD
c:\Program Files (x86)\Java\jre6\bin>
```

Trong đó đường dẫn “c:\Program Files (x86)\Java\jre6\bin” là đường dẫn đến thư mục bin của Java trên máy của bạn

Như ví dụ trên thì mã MD5 sẽ là: 3B:72:4D:33:A6:96:49:41:26:7E:22:1A:ED:4D:DD:AD

- + Lấy key Google Map API:

Vào link: <https://developers.google.com/android/maps-api-signup?hl=vi>

Copy mã và paste như hình dưới đồng thời nhấn nút Generate API Key:



Android Maps APIs Terms of Service

Last Updated: October 13, 2008

Thanks for your interest in the Android Maps APIs. The Android Maps APIs are a collection of services (including, but not limited to, the "com.google.android.maps.MapView" and "android.location.Geocoder" classes) that allow you to include maps, geocoding, and other content from Google and its content providers in your Android applications. The Android Maps APIs explicitly do not include any driving directions data or local search data that may be owned or licensed by Google.

1. Your relationship with Google.

1.1. Your use of any of the Android Maps APIs (referred to in

I have read and agree with the terms and conditions ([printable version](#))

My certificate's MD5 fingerprint: 3B:72:4D:33:A6:96:49:41:26:7E:22:1A:ED:4D:DD:AD

[Generate API Key](#)

Kết quả sẽ như hình:

Thank you for signing up for an Android Maps API key!

Your key is:

OG-Lf_Yd0tS1DIG_OEJNv5x4_mt9vj80jQA76dQ

This key is good for all apps signed with your certificate whose fingerprint is:

3B:72:4D:33:A6:96:49:41:26:7E:22:1A:ED:4D:DD:AD

Here is an example xml layout to get you started on your way to mapping glory:

```
<com.google.android.maps.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey="0G-Lf_Yd0tS1DIG_OEJNv5x4_mt9vj80jQA76dQ"
    />
```

Check out the [API documentation](#) for more information.

Copy đoạn mã

```
<com.google.android.maps.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey="0G-Lf_Yd0tS1DIG_OEJNv5x4_mt9vj80jQA76dQ"
    />
```

Vào file layout chính của bạn (trong bài lab này là file main.xml)



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <!-- Code của google sẽ paste vào đây -->
    <com.google.android.maps.MapView
        android:id="@+id/mapView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey="0G-Lf_Yd0tS1DIG_0EJNv5x4_mt9vj80jQA76dQ"
    />
</RelativeLayout>
```

- **Bước 4:** Để chạy Google Map trên ứng dụng, ta khai báo trong file AndroidManifest.xml dòng:

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Và dòng:

```
<uses-library android:name="com.google.android.maps" />
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.gonmap"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="7"
        android:targetSdkVersion="15" />
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <uses-library android:name="com.google.android.maps" />

        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

Trong file MainActivity.java, ta phải Extend MapsActivity và Overrides phương thức isRouteDisplayed():



```
//import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends MapActivity {

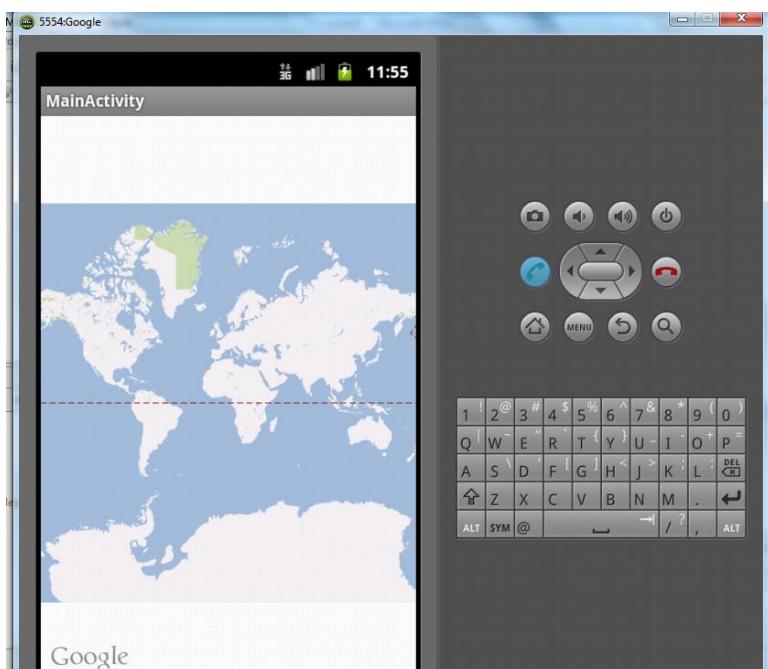
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    protected boolean isRouteDisplayed() {
        // TODO Auto-generated method stub
        return false;
    }
}
```

- **Bước 5:** Chạy ứng dụng để test



- **Bước 6:** Tạo Zoom Control cho Google map:

Thêm các lệnh sau vào MainActivity.java:



```
//import android.app.Activity;
+ import android.os.Bundle;[]

public class MainActivity extends MapActivity {

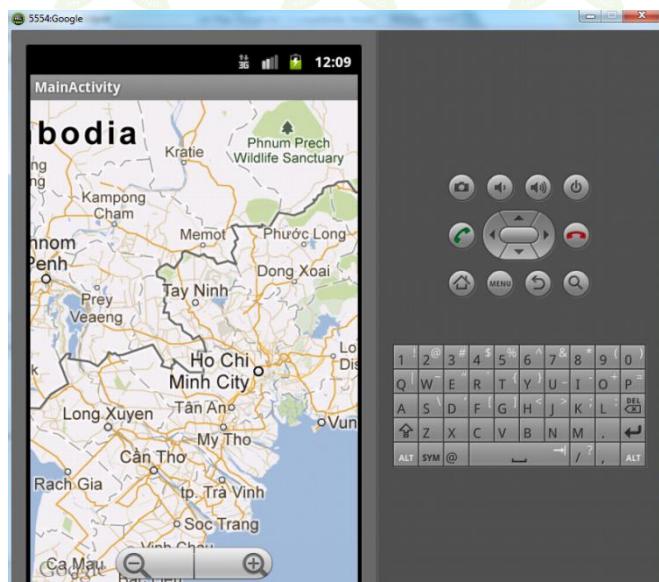
    MapView mapView;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mapView = (MapView) findViewById(R.id.mapView);
        mapView.displayZoomControls(true);
        mapView.setBuiltInZoomControls(true);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    protected boolean isRouteDisplayed() {
        // TODO Auto-generated method stub
        return false;
    }
}
```

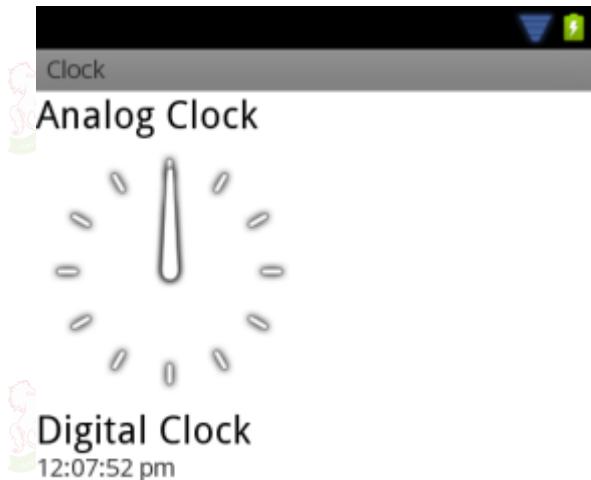
Bước 7: Chạy ứng dụng để test





ANALOGCLOCK & DIGITALCLOCK

- **Bước 1:** Tạo project mới
- **Bước 2:** Tạo layout



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Analog Clock"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <AnalogClock
        android:id="@+id/analogClock1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Digital Clock"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <DigitalClock
        android:id="@+id/digitalClock1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DigitalClock" />

</LinearLayout>
```

- **Bước 3:** Viết thêm code vào src/MainActivity.java



```
package com.android.clock;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.AnalogClock;
import android.widget.DigitalClock;

public class MainActivity extends Activity {

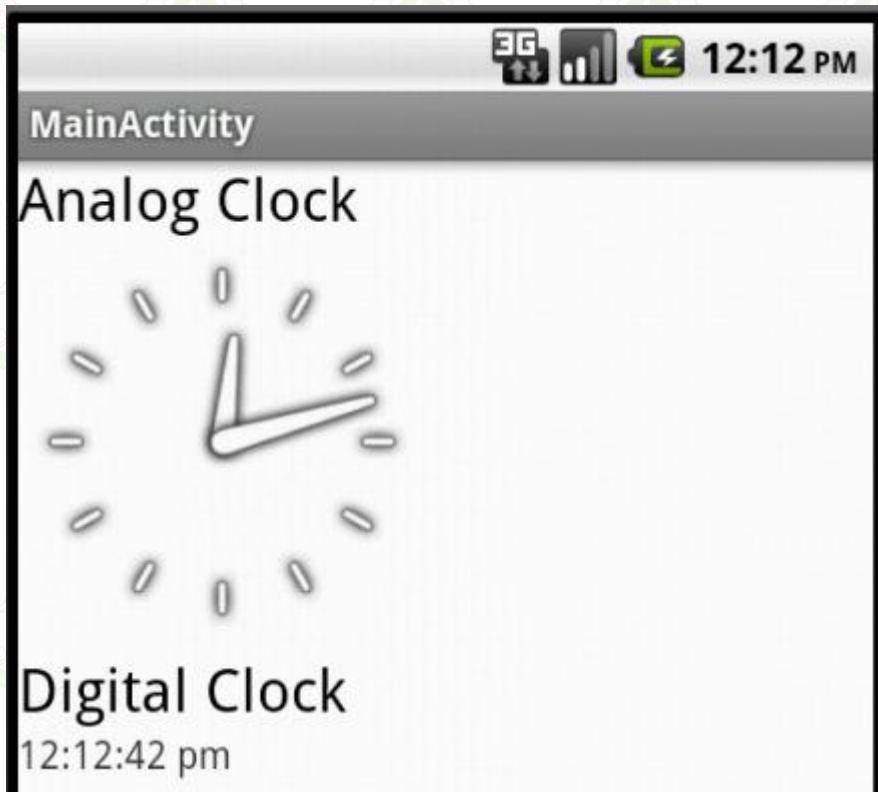
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        AnalogClock ac = (AnalogClock) findViewById(R.id.analogClock1);

        DigitalClock dc = (DigitalClock) findViewById(R.id.digitalClock1);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

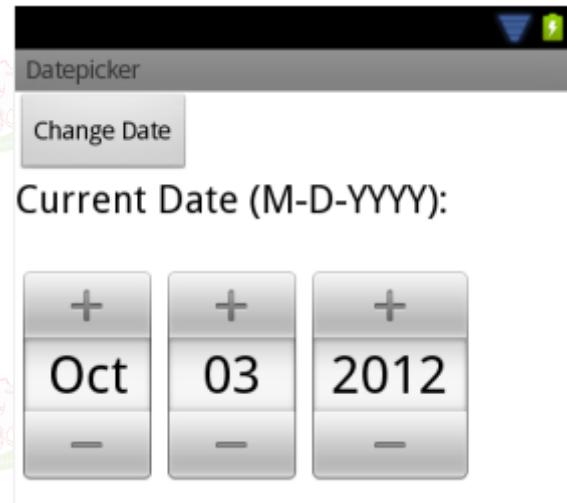
Bước 4: Chạy ứng dụng





DATE PICKER

- **Bước 1:** Tạo project mới
- **Bước 2:** Tạo layout



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/btnChangeDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Change Date" />

    <TextView
        android:id="@+id/lblDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Current Date (M-D-YYYY): "
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:id="@+id/tvDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <DatePicker
        android:id="@+id/dpResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

- **Bước 3:** Thêm code vào src/MainActivity.java
 - + import các thư viện



```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import java.util.Calendar;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TextView;
```

+ Chèn thêm các code:

```
public class MainActivity extends Activity {

    private TextView tvDisplayDate;
    private DatePicker dpResult;
    private Button btnChangeDate;

    private int year;
    private int month;
    private int day;

    static final int DATE_DIALOG_ID = 999;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        setCurrentDateOnView();
        addListenerOnButton();
    }

    // display current date
    public void setCurrentDateOnView() {}

    public void addListenerOnButton() {}

    protected Dialog onCreateDialog(int id) {}

    private DatePickerDialog.OnDateSetListener datePickerListener = new OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker view, int year, int month, int day) {
            String date = year + "-" + (month + 1) + "-" + day;
            tvDisplayDate.setText(date);
        }
    };

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

Trong đó:



```
// display current date
public void setCurrentDateOnView() {

    tvDisplayDate = (TextView) findViewById(R.id.tvDate);
    dpResult = (DatePicker) findViewById(R.id.dpResult);

    final Calendar c = Calendar.getInstance();
    year = c.get(Calendar.YEAR);
    month = c.get(Calendar.MONTH);
    day = c.get(Calendar.DAY_OF_MONTH);

    // set current date into textview
    tvDisplayDate.setText(new StringBuilder()
        // Month is 0 based, just add 1
        .append(month + 1).append("-").append(day).append("-")
        .append(year).append(" "));

    // set current date into datepicker
    dpResult.init(year, month, day, null);

}

public void addListenerOnButton() {

    btnChangeDate = (Button) findViewById(R.id.btnChangeDate);
    btnChangeDate.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {
            showDialog(DATE_DIALOG_ID);
        }
    });
}

@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
    case DATE_DIALOG_ID:
        // set date picker as current date
        return new DatePickerDialog(this, datePickerListener,
            year, month, day);
    }
    return null;
}
```



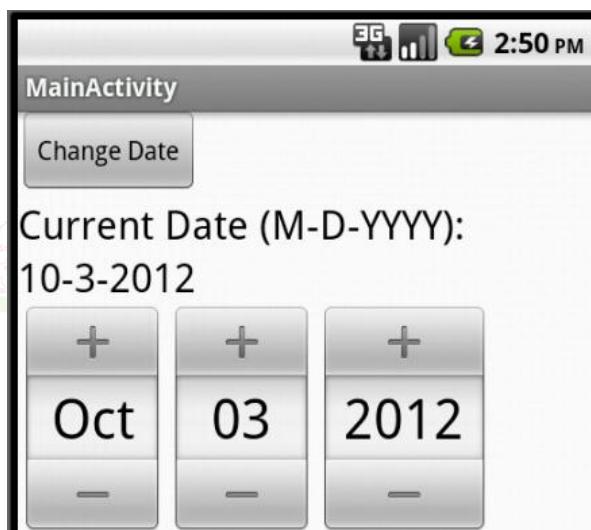
```
private DatePickerDialog.OnDateSetListener datePickerListener
= new DatePickerDialog.OnDateSetListener() {

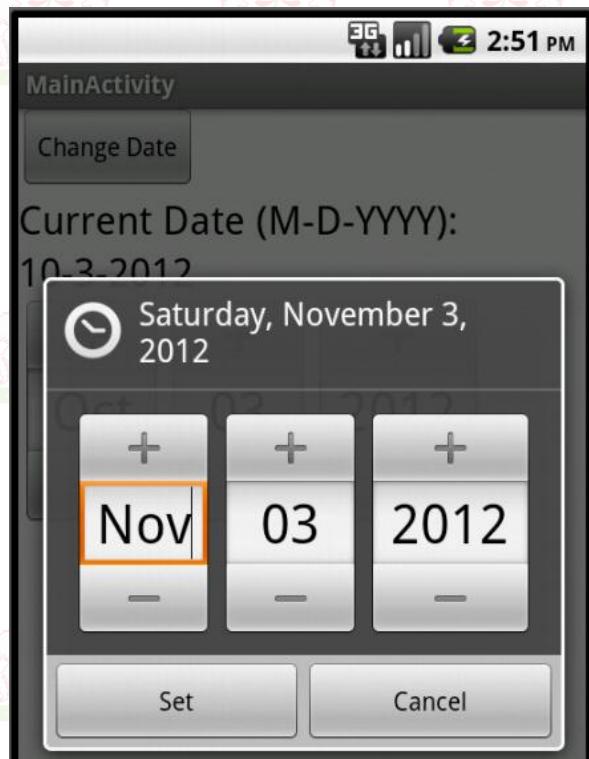
    // when dialog box is closed, below method will be called.
    public void onDateSet(DatePicker view, int selectedYear,
                          int selectedMonth, int selectedDay) {
        year = selectedYear;
        month = selectedMonth;
        day = selectedDay;

        // set selected date into textview
        tvDisplayDate.setText(new StringBuilder().append(month + 1)
            .append("-").append(day).append("-").append(year)
            .append(" "));

        // set selected date into datepicker also
        dpResult.init(year, month, day, null);
    }
};
```

- **Bước 4:** Chạy ứng dụng







TOAST

- **Bước 1:** Tạo project mới
- **Bước 2:** Thiết kế layout



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/buttonToast"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Toast" />

</LinearLayout>
```

- **Bước 3:** Chèn thêm các code sau vào src/M(MainActivity.java)



```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;
```

```
public class MainActivity extends Activity {

    private Button button;

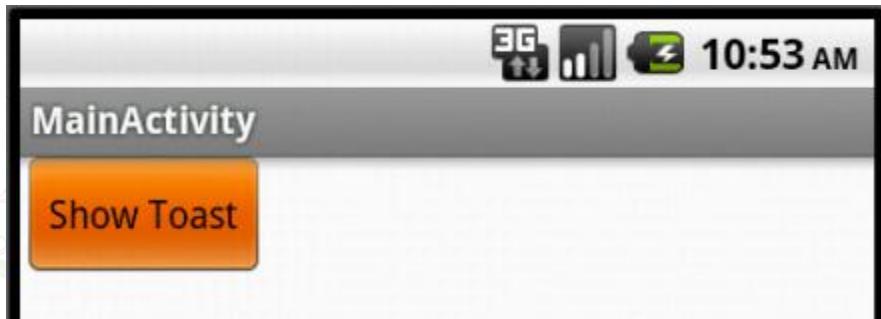
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button = (Button) findViewById(R.id.buttonToast);

        button.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                Toast.makeText(getApplicationContext(),
                        "Button is clicked", Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

- **Bước 4:** Chạy ứng dụng







PROGRESSBAR

- **Bước 1:** Tạo project mới
- **Bước 2:** Tạo layout



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/btnStartProgress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Download File" />

</LinearLayout>
```

- **Bước 3:** Viết thêm code vào src/MainActivity.java



```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.app.ProgressDialog;
import android.os.Handler;
import android.widget.Button;
import android.view.View;
import android.view.View.OnClickListener;

public class MainActivity extends Activity {
    Button btnStartProgress;
    ProgressDialog progressBar;
    private int progressBarStatus = 0;
    private Handler progressBarHandler = new Handler();
    private long fileSize = 0;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        addListenerOnButton();
    }

    public void addListenerOnButton() {
        // file download simulator... a really simple
    }

    public int doSomeTasks() {
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

Trong đó:



```
public void addListenerOnButton() {  
  
    btnStartProgress = (Button) findViewById(R.id.btnStartProgress);  
    btnStartProgress.setOnClickListener(  
        new OnClickListener() {  
  
            public void onClick(View v) {  
  
                // prepare for a progress bar dialog  
                progressBar = new ProgressDialog(v.getContext());  
                progressBar.setCancelable(true);  
                progressBar.setMessage("File downloading ...");  
                progressBar.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);  
                progressBar.setProgress(0);  
                progressBar.setMax(100);  
                progressBar.show();  
  
                //reset progress bar status  
                progressBarStatus = 0;  
  
                //reset filesize  
                fileSize = 0;  
  
                new Thread(new Runnable() {  
                    public void run() {}  
                }).start();  
  
            }  
  
        });  
}
```



```
new Thread(new Runnable() {
    public void run() {
        while (progressBarStatus < 100) {
            // process some tasks
            progressBarStatus = doSomeTasks();
            // your computer is too fast, sleep 1 second
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            // Update the progress bar
            progressBarHandler.post(new Runnable() {
                public void run() {
                    progressBar.setProgress(progressBarStatus);
                }
            });
        }
        // ok, file is downloaded,
        if (progressBarStatus >= 100) {

            // sleep 2 seconds, so that you can see the 100%
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            // close the progress bar dialog
            progressBar.dismiss();
        }
    }).start();
});
});
```

Và:

```
// file download simulator... a really simple
public int doSomeTasks() {

    while (fileSize <= 1000000) {

        fileSize++;

        if (fileSize == 100000) {
            return 10;
        } else if (fileSize == 200000) {
            return 20;
        } else if (fileSize == 300000) {
            return 30;
        }
        // ...add your own

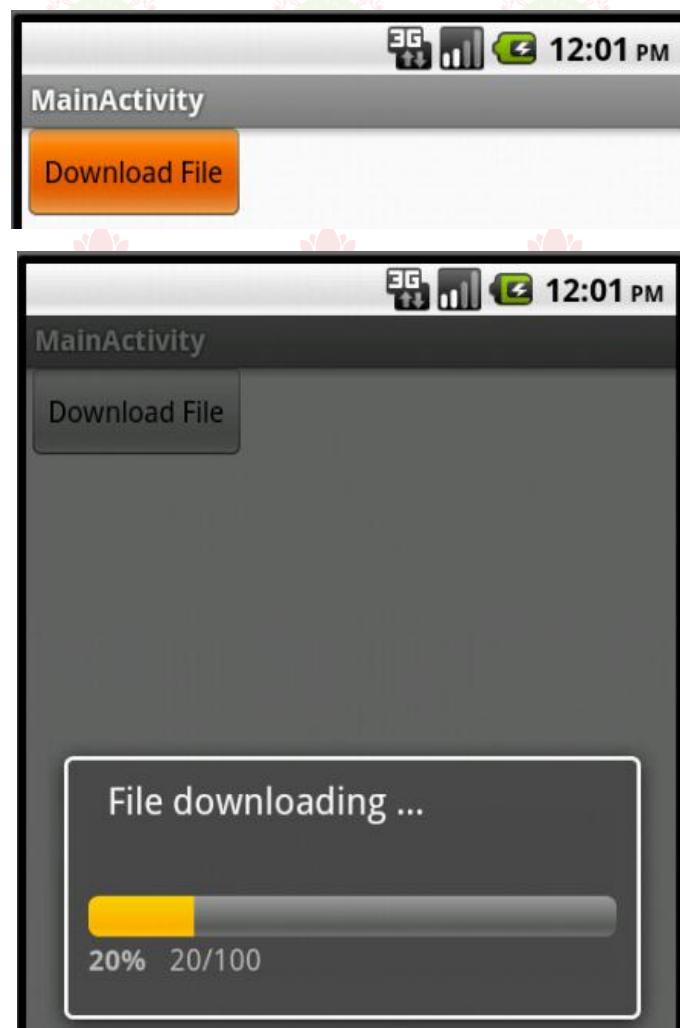
    }

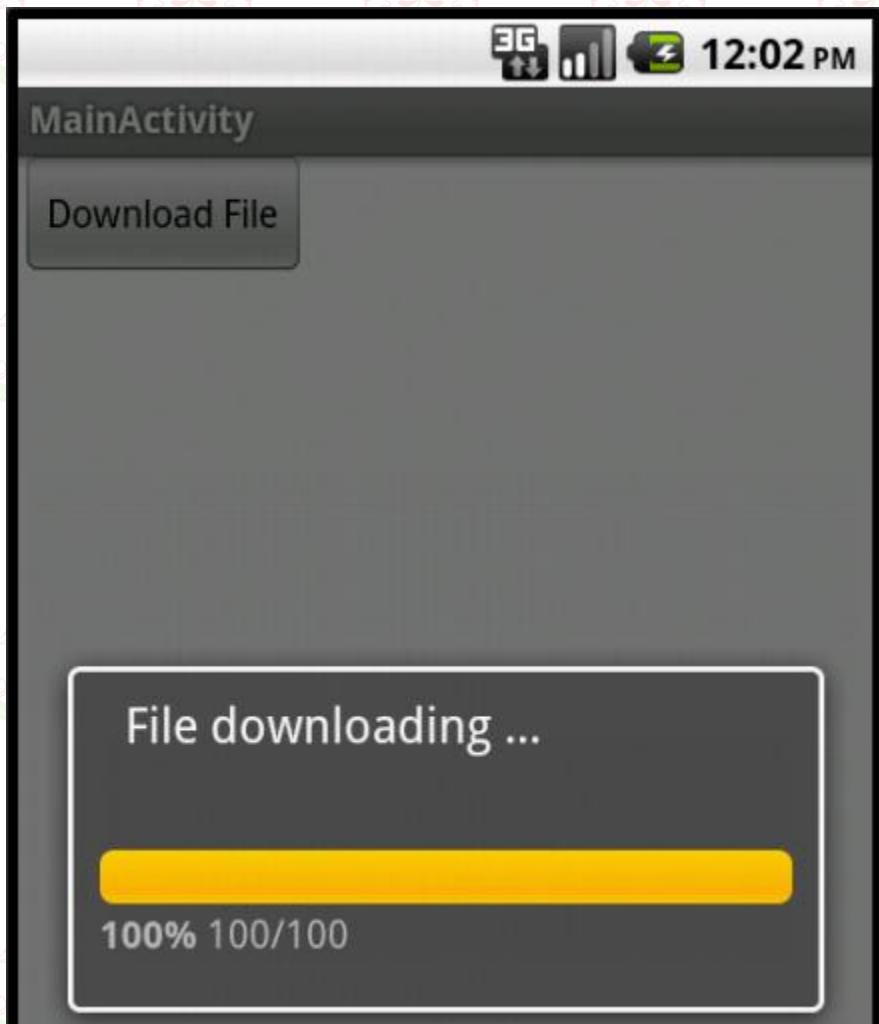
    return 100;

}
```



- **Bước 4: Chạy ứng dụng**





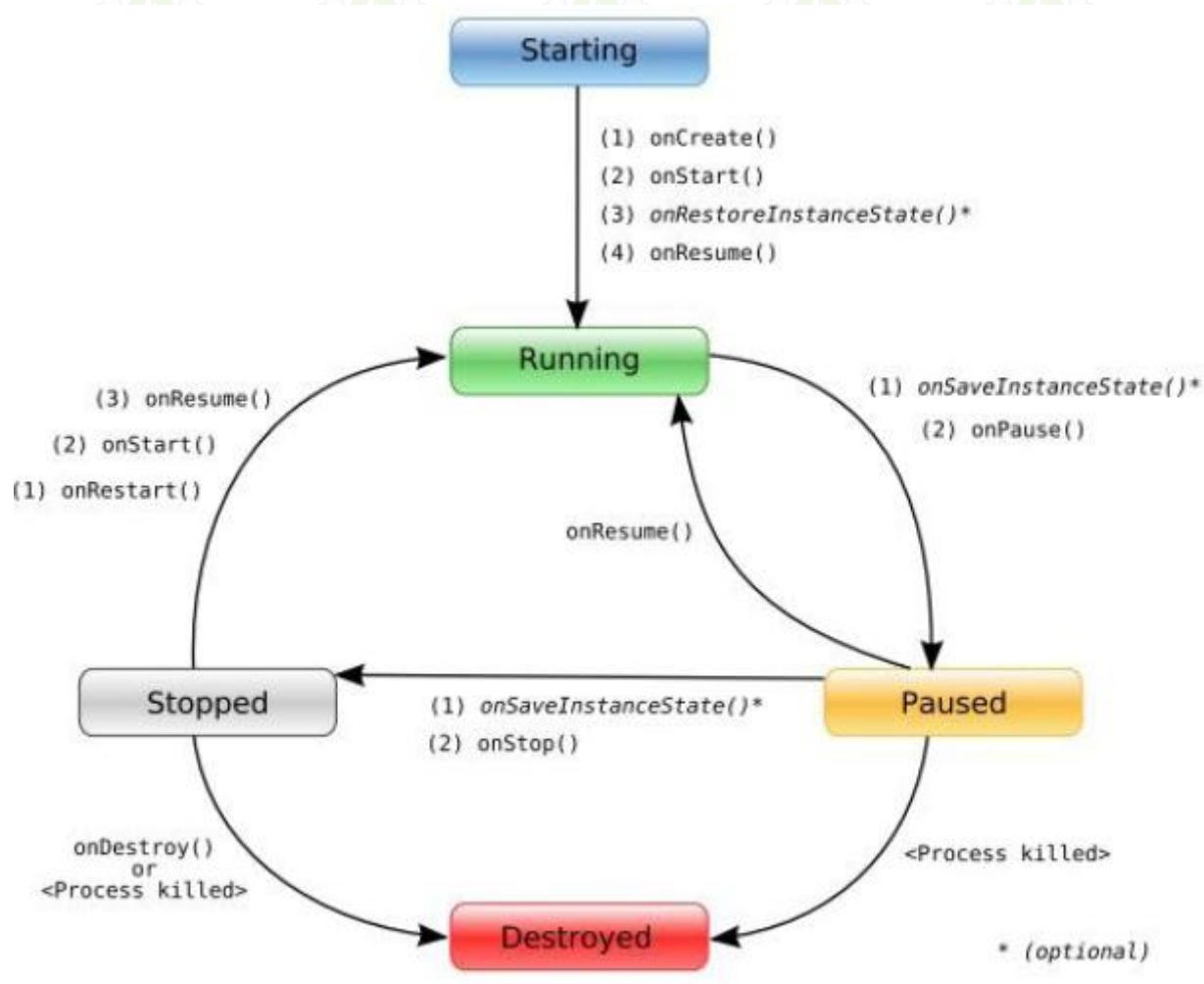


ACTIVITY

1. Activity

- Activity là presentation layer cho ứng dụng Android. Hiểu đơn giản nó một màn hình hiển thị.
- Nó chiếm toàn bộ vùng drawable của màn hình (trừ phần status và title bar).
- Một ứng dụng có thể có nhiều activity.
- Các activity được quản lý trong AndroidManifest.xml
- Có thể chuyển từ một activity sang 1 activity khác. Muốn thực hiện việc chuyển ta phải thông qua Intent.

2. Activity life cycle



Trạng thái Starting (Đang bắt đầu):

- Khi 1 Activity chưa tồn tại trong bộ nhớ nó ở trạng thái Starting (đang bắt đầu).



- Trong khi nó đang bắt đầu nó sẽ gọi các phương thức onCreate, onStart, onResume để chạy và Activity chuyển sang trạng thái Running.
- Trong quá trình chuyển đổi từ Starting sang Running nó sẽ tiêu tốn rất nhiều tài nguyên thiết bị, làm ảnh hưởng trực tiếp đến pin, đó là lý do tại sao ta không tự động hủy các activity dù nó không còn hiển thị.

Trạng thái Running (đang chạy):

- Activity ở trạng thái này là activity đang hiển thị trên màn hình và tương tác với người sử dụng. Ta có thể xem như Activity đang nhận Focus.
- Tại một thời điểm chỉ có 1 Activity đang ở trạng thái Running.
- Activity đang ở trạng thái này được ưu tiên về bộ nhớ và tài nguyên để chạy càng nhanh càng tốt

Trạng thái Paused (tạm nghỉ):

- Khi Activity bị mất Focus (không còn tương tác với người dùng) nhưng vẫn Visible trên màn hình thì nó ở trạng thái Paused.
- Để thấy nhất là khi một Dialogbox được bật lên, Activity nằm bên dưới bị mờ đi (trạng thái tạm nghỉ) nhưng không dừng lại (không qua trạng thái stop), khi dialogbox thực hiện xong thì Activity lại tiếp tục.
- Activity ở trạng thái này vẫn được ưu tiên về bộ nhớ và tài nguyên vì nó vẫn còn chạy, chỉ tạm nghỉ thôi.

Trạng thái Stopped (dừng):

- Khi một activity không còn được visible nhưng vẫn còn trong bộ nhớ, đó là trạng thái Stopped.
- Ở trạng thái này Activity có thể chuyển trở lại trạng thái Running hoặc có thể bị hủy và xóa khỏi bộ nhớ.

Hệ thống giữ các Activity trong trạng thái Stopped vì có thể người dùng sẽ mở trở lại Activity. Tiêu tốn tài nguyên để chuyển từ Stopped sang Running ít hơn nhiều so với chạy lại từ đầu vì dữ liệu vẫn còn trong bộ nhớ và sẵn sàng hoạt động.

- Các Activity ở trạng thái stopped có thể được xóa khỏi bộ nhớ ở bất kỳ thời điểm nào ví dụ như khi hết bộ nhớ hệ thống Android sẽ tự hủy các Activity đang ở trạng thái Stopped để lấy thêm bộ nhớ.



Trạng thái Destroyed (hủy):

- Activity được gọi là Destroyed khi nó không còn nằm trong bộ nhớ.
- Hệ thống Activity Manager sẽ quyết định xem Activity nào không còn cần thiết nữa để xóa nó ra khỏi bộ nhớ.
- Trước khi bị hủy Activity có thể thực hiện một số hoạt động ví dụ lưu dữ liệu. Tuy nhiên không đảm bảo rằng các hoạt động đó thực hiện được trước khi bị hủy. Do đó, tốt nhất là thực hiện các việc quan trọng ví dụ như lưu dữ liệu tại trạng thái Paused chứ không đợi đến trạng thái Destroyed mới thực hiện.

3. Tạo một Activity mới

Mỗi project được tạo ra đều có 1 Activity mặc định. Ta có thể tạo thêm các activity theo các bước:

- Tạo một file layout mới cho activity mới đặt tên vd: activitymoi.xml
- Tạo một class mới cho project đặt tên vd: activitymoi.java
- Cho class vừa tạo kế thừa từ lớp Activity và override hàm onCreate.
- Khai báo activity vừa tạo trong file AndroidManifest.xml.
- Trong hàm onCreate. Dùng hàm setContentView để chỉ ra layout cho file java.

4. Intent

- Intent là thông điệp bất đồng bộ, nó cho phép ứng dụng yêu cầu các chức năng từ những component khác trong hệ thống Android như các activity hay các service.
- Nói đơn giản intent là cầu nối giữa Activity này và Activity kia hoặc giữa Activity với một component khác.
- Intent có 2 dạng là
- Explicit Intent
- Implicit Intent

5. Explicit Intent (intent tường minh):

- Khi gọi nó xác định rõ component sẽ nhận để xử lý.
- Dùng nó để 1 activity gọi 1 activity khác.

Intent i=new Intent(view.getContext(),Activity2.class);

6. Implicit Intent (intent không tường minh)

- Khi gọi không xác định rõ component sẽ nhận để xử lý.
- Dựa vào một số thông tin mà hệ thống sẽ biết dùng component nào để xử lý hoặc cho người dùng chọn component để xử lý (nếu có nhiều component xử lý được tác vụ đó).



```
Intent i=new Intent(Intent.ACTION_VIEW,Uri.parse("http://www.google.com"));
```

7. Chạy và kết thúc Activity

- Sau khi có Intent. Đối với dạng Activity đơn giản (Activity này mở Activity khác). Ta dùng hàm startActivity(), truyền vào intent để chuyển qua Activity khác.
- Trong Activity để kết thúc hoàn toàn Activity ta dùng hàm finish().
- Nếu không dùng hàm finish() thì Activity đó vẫn còn cho dù đã qua Activity khác.

8. Truyền dữ liệu

- Khi Activity1 gọi Activity2 ta có thể truyền dữ liệu từ Activity1 qua cho Activity2.
- Khi Activity2 kết thúc có thể trả ngược dữ liệu về cho Activity1.
- Cả Explicit Intent và Implicit Intent đều có thể truyền dữ liệu.
- Implicit Intent còn dựa vào dữ liệu truyền để quyết định dùng component nào để mở.
- Khi có nhiều chương trình có thể phục vụ được yêu cầu thì Android sẽ hiện hộp thoại cho phép người sử dụng chọn lựa dùng chương trình nào để mở.
- Ta có thể khai báo trong AndroidManifest.xml để đăng kí với hệ thống biết ta có thể phục vụ cho 1 yêu cầu.

VD: Chương trình của ta có thể xem hình ảnh, ta đăng kí với hệ thống. Khi có 1 chương trình nào cần gọi chương trình xem ảnh, khi đó hệ thống sẽ đưa ta vào danh sách phục vụ cho người dùng chọn.

9. Truyền dữ liệu từ 1 Activity sang 1 Activity khác.

Sau khi khởi tạo Intent ta có thể đưa dữ liệu cho Intent để gửi qua Activity khác bằng hàm putExtra như sau:

```
intent.putExtra("gia_tri_1", new String("xin chao ban"));
intent.putExtra("gia_tri_2", new Integer(20));
```

Bên Activity2 tức là Activity nhận ta có thể dùng getIntent().getExtras() cùng với 1 hàm getString() hoặc getInt()... (đúng kiểu dữ liệu được gửi) để nhận dữ liệu.

```
String gt1=getIntent().getExtras().getString("gia_tri_1");
int gt2=getIntent().getExtras().getInt("gia_tri_2");
```

10. Trả dữ liệu về từ Activity bị gọi cho Activity gọi.

Khi Activity1 gọi Activity2, sau khi Activity2 xử lý xong có thể trả dữ liệu ngược trở về cho Activity1 trước khi kết thúc. Thực hiện như sau:



- Trên Activity1 khi dùng Intent ta dùng hàm startActivityForResult và truyền vào intent, requestCode.

```
Intent i=new Intent(arg0.getContext(),activity2.class);  
//gửi kèm requestCode để biết gửi cho ai để xét ai trả về.  
startActivityForResult(i,999);
```

- Trên Activity2 ta tạo 1 intent, đưa dữ liệu vào cho intent và dùng hàm setResult và truyền vào intent đã tạo.

```
String tuoi=et.getText().toString();  
Intent intent_tra_ve=new Intent();  
intent_tra_ve.putExtra("tuoi", tuoi);  
setResult(RESULT_OK, intent_tra_ve);
```

- Quay lại Activity1 ta override lên hàm onActivityResult. Bên trong hàm ta xét requestCode để xác định ai trả về, resultCode để xác định trạng thái trả về. Sau đó ta có thể lấy dữ liệu trả về như sau:

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    // TODO Auto-generated method stub  
    super.onActivityResult(requestCode, resultCode, data);  
  
    if(requestCode==999 && resultCode==RESULT_OK)  
    {  
        String kq=data.getStringExtra("tuoi").toString();  
        tv.setText("tuoi ban la: " + kq);  
    }  
}
```

11. Gửi dữ liệu của Implicit Intent

- Implicit Intent cũng gửi dữ liệu cho 1 Intent không xác định trước.
- Cung cấp loại hành động, loại dữ liệu, thao tác thuộc nhóm nào... hệ thống sẽ tự tìm activity đã đăng ký để kích hoạt.

Implicit Intent truyền vào 2 đối số: đối số 1 chỉ ra loại hành động, đối số thứ 2 là dữ liệu dạng Uri

```
Intent i=new Intent(Intent.ACTION_VIEW,Uri.parse("http://www.google.com"));
```

Ta cũng có thể dùng hàm setData để đưa dữ liệu vào cho Intent



```
Intent j=new Intent(Intent.ACTION_VIEW);
j.setData(Uri.parse("http://www.google.com"));
```

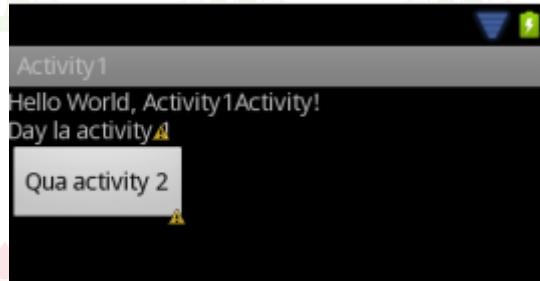
Hoặc dùng setType để chỉ ra loại dữ liệu

```
Intent i=new Intent(Intent.ACTION_GET_CONTENT);
i.setType("image/*");
```

12. Một số Lab

12.1 Lab 1:

- Bước 1: Tạo project mới
- Bước 2: Tạo layout cho Activity1



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

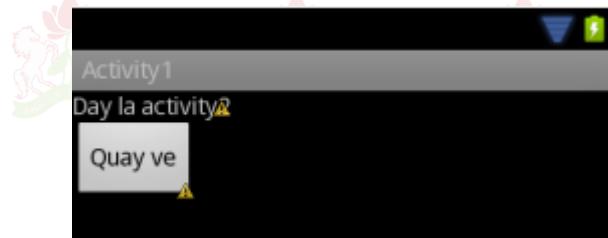
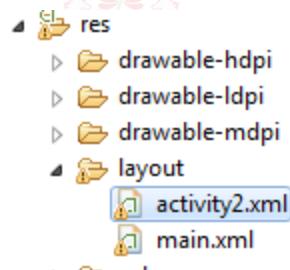
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Day la activity 1" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Qua activity 2" />

</LinearLayout>
```

- Bước 3: Tạo layout cho Activity2 (res/layout/activity2.xml)



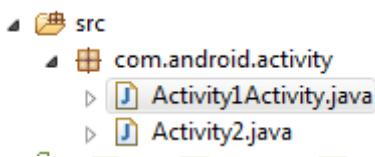
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Day la activity2" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Quay ve" />

</LinearLayout>
```

- Bước 4: Viết code cho activity1





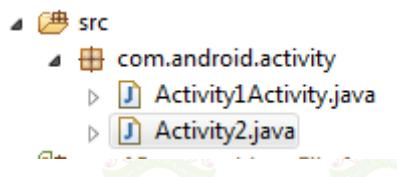
```
import android.app.Activity;
import android.view.View;
import android.widget.Button;
import android.os.Bundle;
import android.content.Intent;

public class Activity1Activity extends Activity {
    /** Called when the activity is first created. */
    Button bt;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        bt=(Button)findViewById(R.id.button1);
        bt.setOnClickListener(new View.OnClickListener(){

            public void onClick(View v){
                Intent i=new Intent(v.getContext(),Activity2.class);
                startActivity(i);
                //finish(); // Dung lenh nay khi quay ve activity 1 la ket thuc luon
            }
        });
    }
}
```

- Bước 5: Viết code cho activity2



```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.app.Activity;

public class Activity2 extends Activity {

    Button bt;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity2);

        bt=(Button)findViewById(R.id.button1);
        bt.setOnClickListener(new View.OnClickListener(){

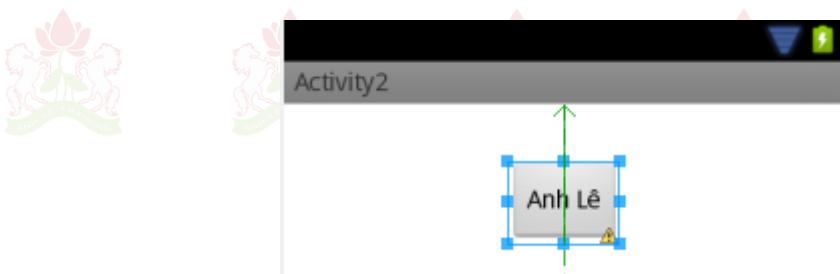
            public void onClick(View v){
                finish();
            }
        });
    }
}
```

- Bước 6: Chạy ứng dụng



12.2 Lab 2: (Dùng Implicit Intent)

- Bước 1: Tạo project
- Bước 2: Tạo layout



```
<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" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp"
        android:text="Anh Lê" />

</RelativeLayout>
```



- Bước 3: Viết code

```
import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {
    Button bt;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

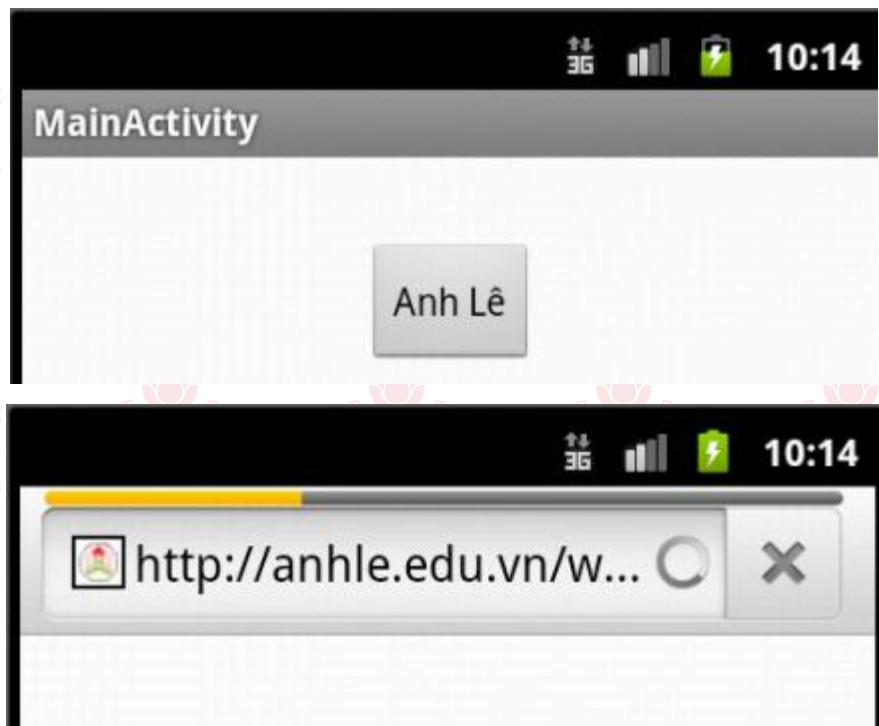
        bt=(Button)findViewById(R.id.button1);
        bt.setOnClickListener(new View.OnClickListener(){

            public void onClick(View v){
                Intent i=new Intent(Intent.ACTION_VIEW);
                i.setData(Uri.parse("http://anhle.edu.vn"));
                startActivity(i);

            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

- Bước 4: Chạy ứng dụng





12.3 Lab 3 (Truyền dữ liệu giữa các Activity)

- Bước 1: Tạo project mới
- Bước 2: Tạo layout chính



Năm Sinh Của Bạn

Xem Tuổi Theo Can Chi



```
<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" >

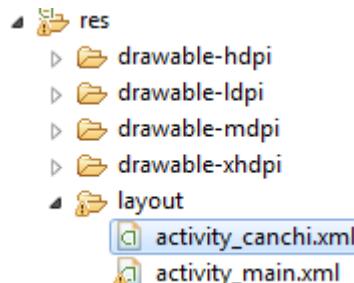
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="71dp"
        android:ems="10" />

    <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:layout_marginTop="28dp"
        android:text="Xem Tuổi Theo Can Chi" />

    <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="40dp"
        android:text="Năm Sinh Của Bạn" />

</RelativeLayout>
```

- Bước 3: Tạo layout mới



Tuổi Theo Can Chi Của bạn

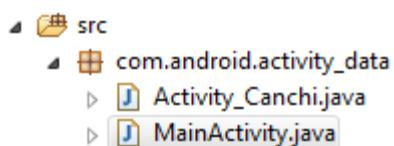


```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/RelativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <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="75dp"
        android:text="Tuổi Theo Can Chi Của ban" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="23dp" />
</RelativeLayout>
```

- Bước 4: Viết code cho MainActivity.java





```
package com.android.activity_data;

import android.os.Bundle;

public class MainActivity extends Activity {

    Button bt;
    EditText ns;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

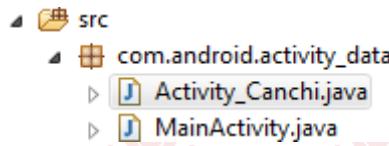
        bt=(Button)findViewById(R.id.button1);
        ns=(EditText) findViewById(R.id.editText1);

        bt.setOnClickListener(new View.OnClickListener(){
            public void onClick(View v){
                //finish();
                Intent i=new Intent(v.getContext(),Activity_Canchi.class);
                i.putExtra("tuoi", ns.getText().toString());
                startActivity(i);

            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

- Bước 5: Viết code cho Activity_Canchi.java





```
package com.android.activity_data;

import android.os.Bundle;

public class Activity_Canchi extends Activity {

    TextView txt1;
    String chi, can;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_canchi);

        txt1=(TextView)findViewById(R.id.textView2);

        int nam=Integer.parseInt(getIntent().getExtras().getString("tuoi").toString());

        switch(nam%10)
        {
            case 0: can="Canh"; break;
            case 1: can="Tân"; break;
            case 2: can="Nhâm"; break;
            case 3: can="Quý"; break;
            case 4: can="Giáp"; break;
            case 5: can="Ất"; break;
            case 6: can="Bính"; break;
            case 7: can="Đinh"; break;
            case 8: can="Mậu"; break;
            case 9: can="Kỷ"; break;
        }

    }

    switch(nam%12)
    {
        case 0: chi="Thân"; break;
        case 1: chi="Dậu"; break;
        case 2: chi="Tuất"; break;
        case 3: chi="Hợi"; break;
        case 4: chi="Tí"; break;
        case 5: chi="Sửu"; break;
        case 6: chi="Dần"; break;
        case 7: chi="Mão"; break;
        case 8: chi="Thìn"; break;
        case 9: chi="Tị"; break;
        case 10: chi="Ngọ"; break;
        case 11: chi="Mùi"; break;
    }
    txt1.setText(can+" "+chi);
}
```

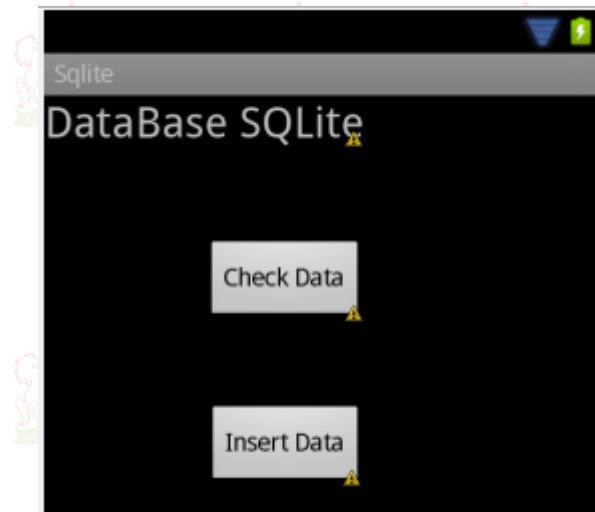
- Bước 6: Chạy ứng dụng





SQLITE

- Bước 1: Tạo project mới
- Bước 2: Tạo layout chính



Activity_main.xml:

```
<?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" >

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="DataBase SQLite"
        android:textSize="24.5sp" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/text"
        android:layout_below="@+id/text"
        android:layout_marginTop="52dp"
        android:text="Check Data" />

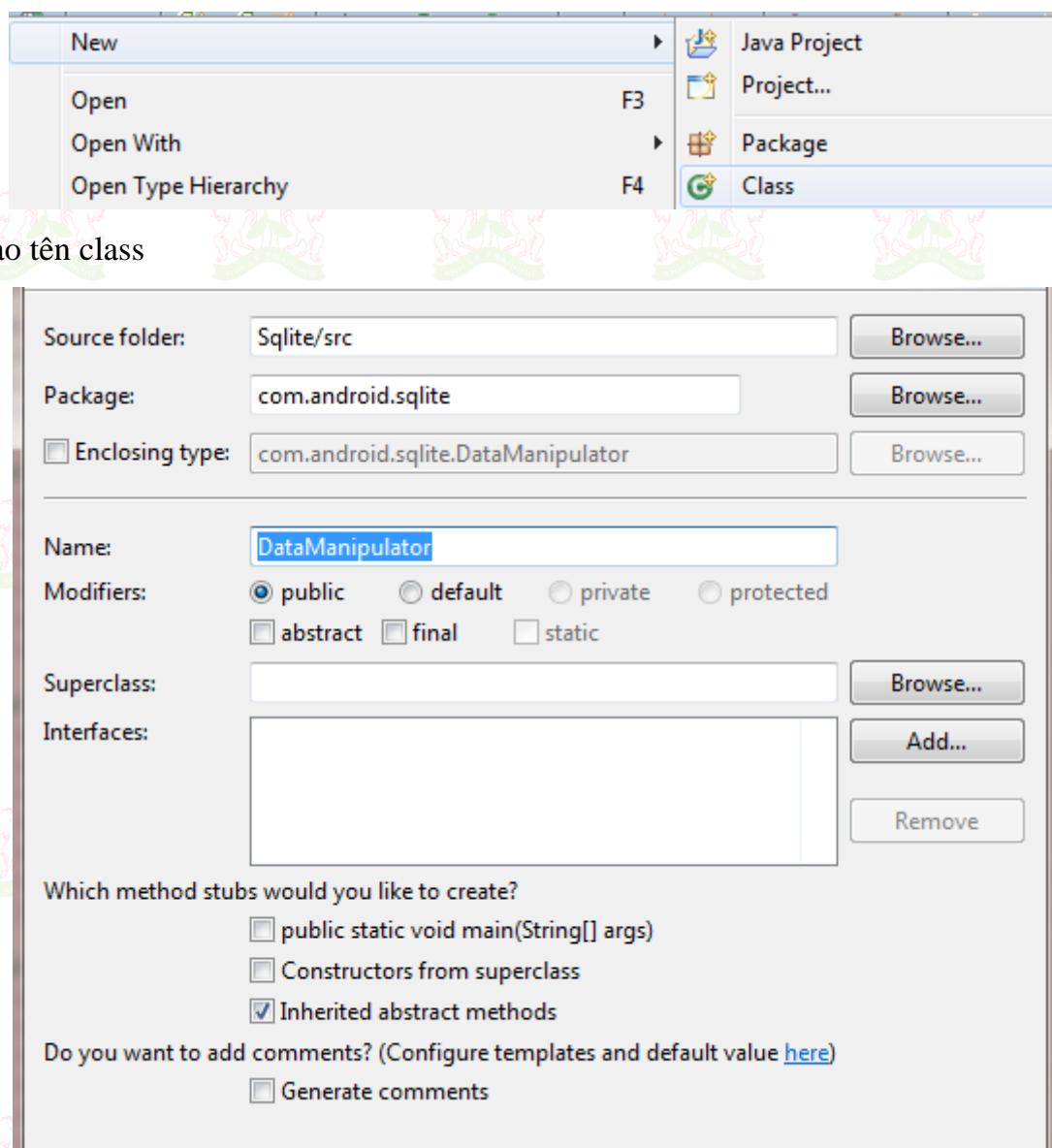
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/button2"
        android:layout_below="@+id/button2"
        android:layout_marginTop="46dp"
        android:text="Insert Data" />

</RelativeLayout>
```

- Bước 3: Tạo class có tên file là DataManipulator.java



- + Click phải chuột tại thư mục src/com.android.sqlite
- + Chọn New, Class



- + Gõ vào code cho class DataManipulator.class



```
package com.android.sqlite;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteStatement;
import java.util.ArrayList;
import java.util.List;

public class DataManipulator {

    private static final String DATABASE_NAME = "mydatabase.db";
    private static final int DATABASE_VERSION = 1;
    static final String TABLE_NAME = "newtable";
    private static Context context;
    static SQLiteDatabase db;

    private SQLiteStatement insertStmt;

    private static final String INSERT = "insert into " + TABLE_NAME +
            " (name,number,skypeId,address) values (?,?,?,?,?)";
    public DataManipulator(Context context) {}
    public long insert(String name, String number, String skypeId, String address) {}

    public void deleteAll() {}

    public List<String[]> selectAll() {}

    public void delete(int rowId) {}

    private static class OpenHelper extends SQLiteOpenHelper {}

}
```

Trong đó các function:

```
public DataManipulator(Context context) {
    DataManipulator.context = context;
    OpenHelper openHelper = new OpenHelper(DataManipulator.context);
    DataManipulator.db = openHelper.getWritableDatabase();
    this.insertStmt = DataManipulator.db.compileStatement(INSERT);
}

public long insert(String name, String number, String skypeId, String address) {
    this.insertStmt.bindString(1, name);
    this.insertStmt.bindString(2, number);
    this.insertStmt.bindString(3, skypeId);
    this.insertStmt.bindString(4, address);
    return this.insertStmt.executeInsert();
}

public void deleteAll() {
    db.delete(TABLE_NAME, null, null);
}
```



```
public List<String[]> selectAll()
{
    List<String[]> list = new ArrayList<String[]>();
    Cursor cursor = db.query(TABLE_NAME, new String[] { "id", "name", "number", "skypeId", "address" },
                           null, null, null, null, "name asc");
    int x=0;
    if (cursor.moveToFirst()) {
        do {
            String[] b1=new String[]{cursor.getString(0),cursor.getString(1),cursor.getString(2),
cursor.getString(3),cursor.getString(4)};
            list.add(b1);
            x=x+1;
        } while (cursor.moveToNext());
    }
    if (cursor != null && !cursor.isClosed()) {
        cursor.close();
    }
    cursor.close();
    return list;
}

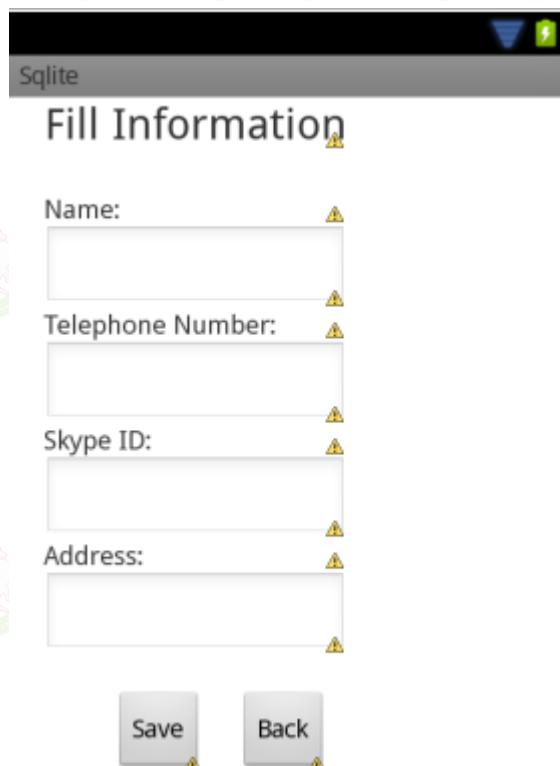
public void delete(int rowId) {
    db.delete(TABLE_NAME, null, null);
}

private static class OpenHelper extends SQLiteOpenHelper {
    OpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE_NAME + " (id INTEGER PRIMARY KEY, name TEXT, " +
                "number TEXT, skypeId TEXT, address TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
    {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}
```



- Bước 4: Tạo layout mới res/layout/save.xml



File save.xml:



```
<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout01"
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="20sp">
    <TextView
        android:id="@+id/button1_label"
        android:layout_height="fill_parent"
        android:layout_width="wrap_content"
        android:text="Fill Information"
        android:textSize="24.5sp"
        android:layout_gravity="center"
        android:layout_marginBottom="25dip" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Name:"
        android:layout_gravity="center"
        android:textSize="14.5sp" />
    <EditText
        android:id="@+id/name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Telephone Number:"
        android:layout_gravity="center"
        android:textSize="14.5sp" />
    <EditText
        android:id="@+id/number"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Skype ID:"
        android:layout_gravity="center"
        android:textSize="14.5sp" />
    <EditText
        android:id="@+id/skypeId"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Address:"
        android:layout_gravity="center"
        android:textSize="14.5sp" />
    <EditText
        android:id="@+id/address"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <LinearLayout
        android:id="@+id/LinearLayout02"
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="20sp">
```



```
<Button
    android:text="Save"
    android:id="@+id/Button01add"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20sp"
    android:layout_marginLeft="20sp">
</Button>
<Button
    android:text="Back"
    android:id="@+id/Button01home"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20sp"
    android:layout_marginLeft="20sp">
</Button>
</LinearLayout>
</LinearLayout>
```

- Bước 5: Tạo Activity SaveData.java để lưu thông tin

```
package com.android.sqlite;

import android.app.Activity;
import android.view.View;
import android.view.View.OnClickListener;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.widget.EditText;
import android.widget.TextView;

public class SaveData extends Activity implements OnClickListener {
    private DataManipulator dh;
    static final int DIALOG_ID = 0;

    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.save);
        View add = findViewById(R.id.Button01add);
        add.setOnClickListener(this);
        View home = findViewById(R.id.Button01home);
        home.setOnClickListener(this);
    }

    public void onClick(View v){}

    protected final Dialog onCreateDialog(final int id) {
```

và:



```
public void onClick(View v){
    switch(v.getId()){
        case R.id.Button01home:
            Intent i = new Intent(this, MainActivity.class);
            startActivity(i);
            break;
        case R.id.Button01add:
            View editText1 = (EditText) findViewById(R.id.name);
            View editText2 = (EditText) findViewById(R.id.number);
            View editText3 = (EditText) findViewById(R.id.skypeId);
            View editText4 = (EditText) findViewById(R.id.address);
            String myEditText1=((TextView) editText1.getText()).toString();
            String myEditText2=((TextView) editText2.getText()).toString();
            String myEditText3=((TextView) editText3.getText()).toString();
            String myEditText4=((TextView) editText4.getText()).toString();
            this.dh = new DataManipulator(this);
            this.dh.insert(myEditText1,myEditText2,myEditText3,myEditText4);
            showDialog(DIALOG_ID);
            break;
    }
}

protected final Dialog onCreateDialog(final int id) {
    Dialog dialog = null;
    switch(id) {
        case DIALOG_ID:
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setMessage("Information saved successfully ! Add Another Info?");
            builder.setCancelable(false).setPositiveButton("No",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        SaveData.this.finish();
                    }
                }).setNegativeButton("Yes", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        dialog.cancel();
                    }
                });
            AlertDialog alert = builder.create();
            dialog = alert;
            break;
        default:
    }
    return dialog;
}
```

- Bước 6: Tạo layout mới res/layout/check.xml



Item 1

Sub Item 1

Item 2

Sub Item 2

Item 3

Sub Item 3



check.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
<TextView
    android:id="@+id/selection2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="NAME - TELPHONE NO - SKYPE ID- ADDRESS" />
<TextView
    android:id="@+id/selection"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<ListView
    android:id="@+id/list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:drawSelectorOnTop="false"
    android:textSize="3dip" />
</LinearLayout>
```

- Bước 7: Tạo Activity CheckData.java

```
package com.android.sqlite;

import java.util.ArrayList;
import java.util.List;
import android.app.ListActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

public class CheckData extends ListActivity {
    TextView selection;
    public int idToModify;
    DataManipulator dm;

    List<String[]> list = new ArrayList<String[]>();
    List<String[]> names2 = null ;
    String[] stg1;
    protected void onCreate(Bundle savedInstanceState){}

    public void onListItemClick(ListView parent, View v, int position, long id) {}
```

và:



```
protected void onCreate(Bundle savedInstanceState){  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.check);  
    dm = new DataManipulator(this);  
    names2 = dm.selectAll();  
  
    stg1=new String[names2.size()];  
    int x=0;  
    String stg;  
  
    for (String[] name : names2) {  
        stg = name[1] + " - " + name[2] + " - " + name[3] + " - " + name[4];  
        stg1[x]=stg;  
        x++;  
    }  
  
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,  
        android.R.layout.simple_list_item_1,stg1);  
    this.setListAdapter(adapter);  
    selection=(TextView)findViewById(R.id.selection);  
}  
  
public void onListItemClick(ListView parent, View v, int position, long id) {  
    selection.setText(stg1[position]);  
}
```



- Bước 8: Viết code cho MainActivity.java

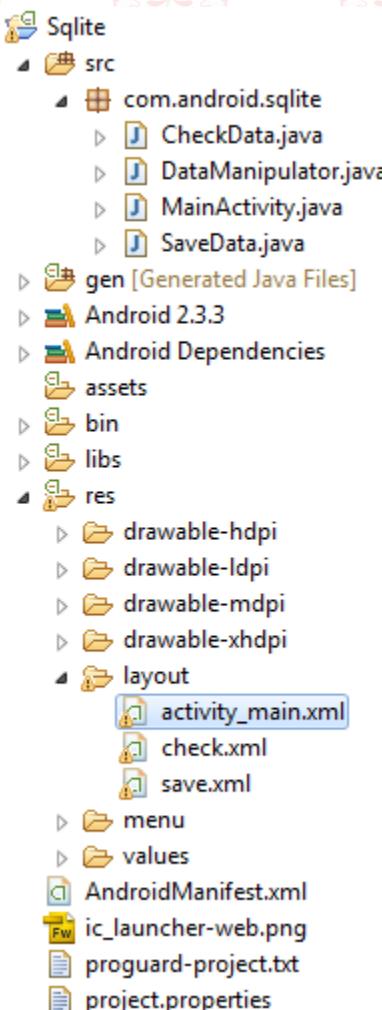
```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;

public class MainActivity extends Activity implements OnClickListener {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        View button1Click = findViewById(R.id.button1);
        button1Click.setOnClickListener(this);
        View button2Click = findViewById(R.id.button2);
        button2Click.setOnClickListener(this);
    }

    public void onClick(View v) {
        // TODO Auto-generated method stub
        switch(v.getId()){
            case R.id.button1:
                Intent i = new Intent(this, SaveData.class);
                startActivity(i);
                break;
            case R.id.button2:
                Intent ii = new Intent(this, CheckData.class);
                startActivity(ii);
                break;
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

- + Ứng dụng sau khi hoàn thành có cấu trúc thư mục như sau:





- Bước 9: Thêm tham số vào AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.sqlite"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".SaveData" android:label="@string/app_name">
        </activity>
        <activity android:name=".CheckData" android:label="@string/app_name">
        </activity>
    </application>

</manifest>
```

- Bước 10: Chạy ứng dụng



- + Chọn "Insert Data"



3G 10:54 AM

Sqlite

Fill Information

Name:

Telephone Number:

Skype ID:

Address:

AB 10:53 AM

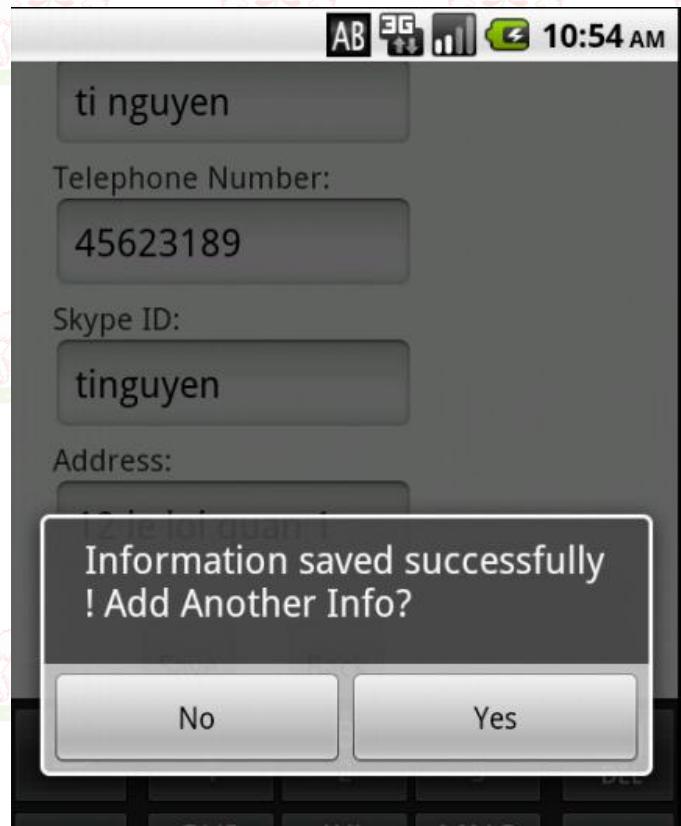
ti nguyen

Telephone Number:

Skype ID:

Address:

1
2
3



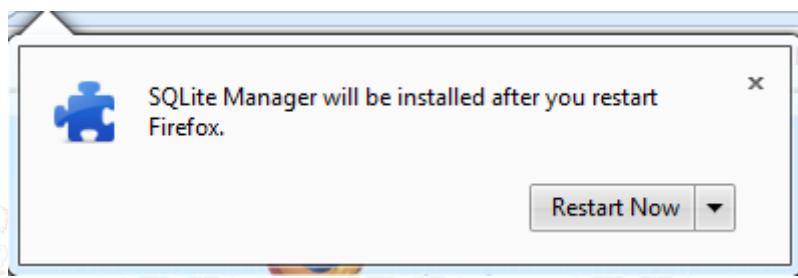
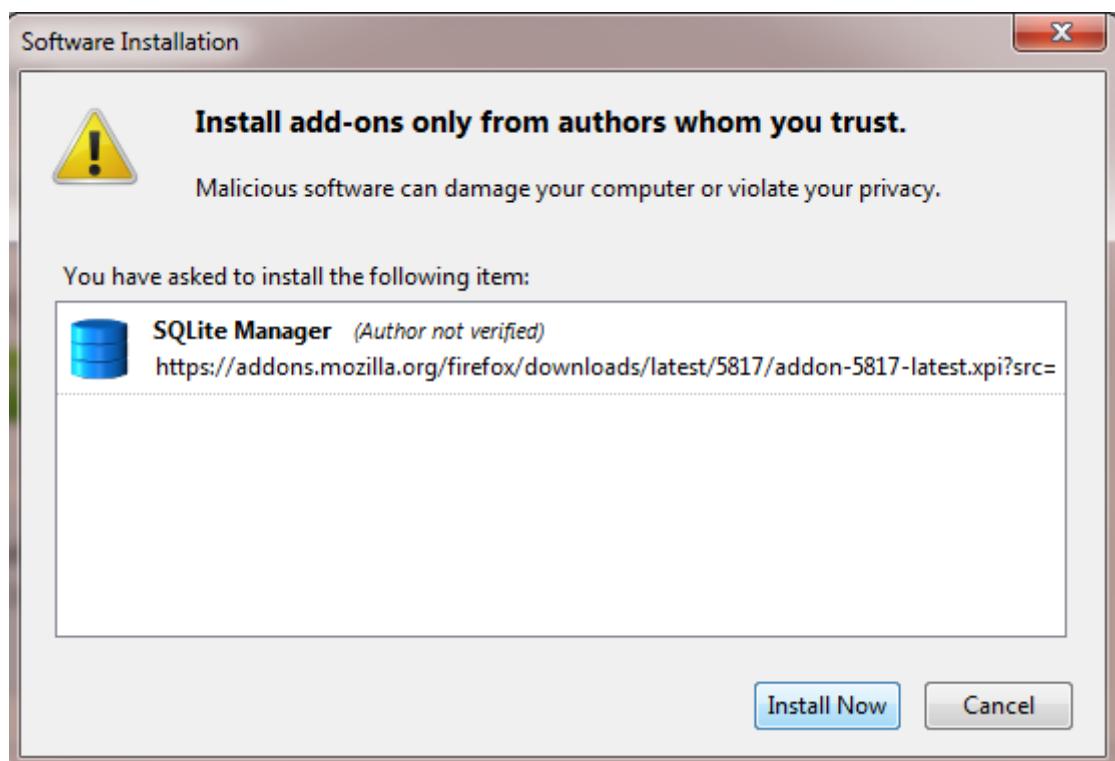
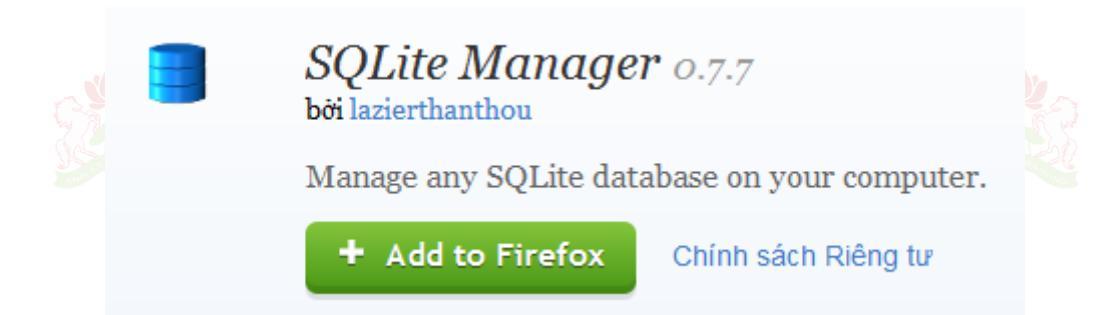
+ Chọn “Check Data”

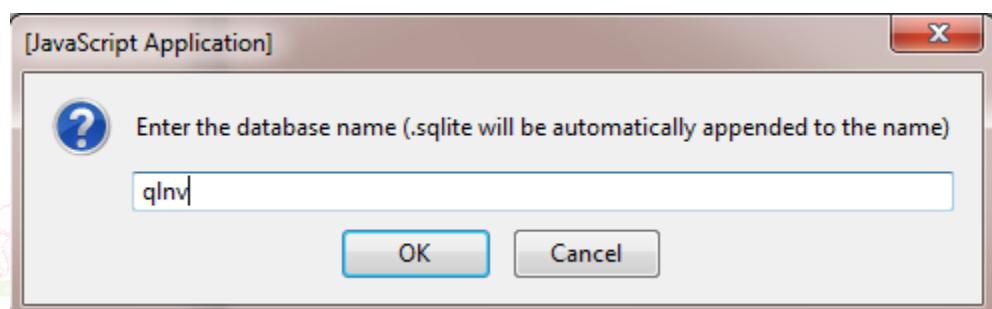
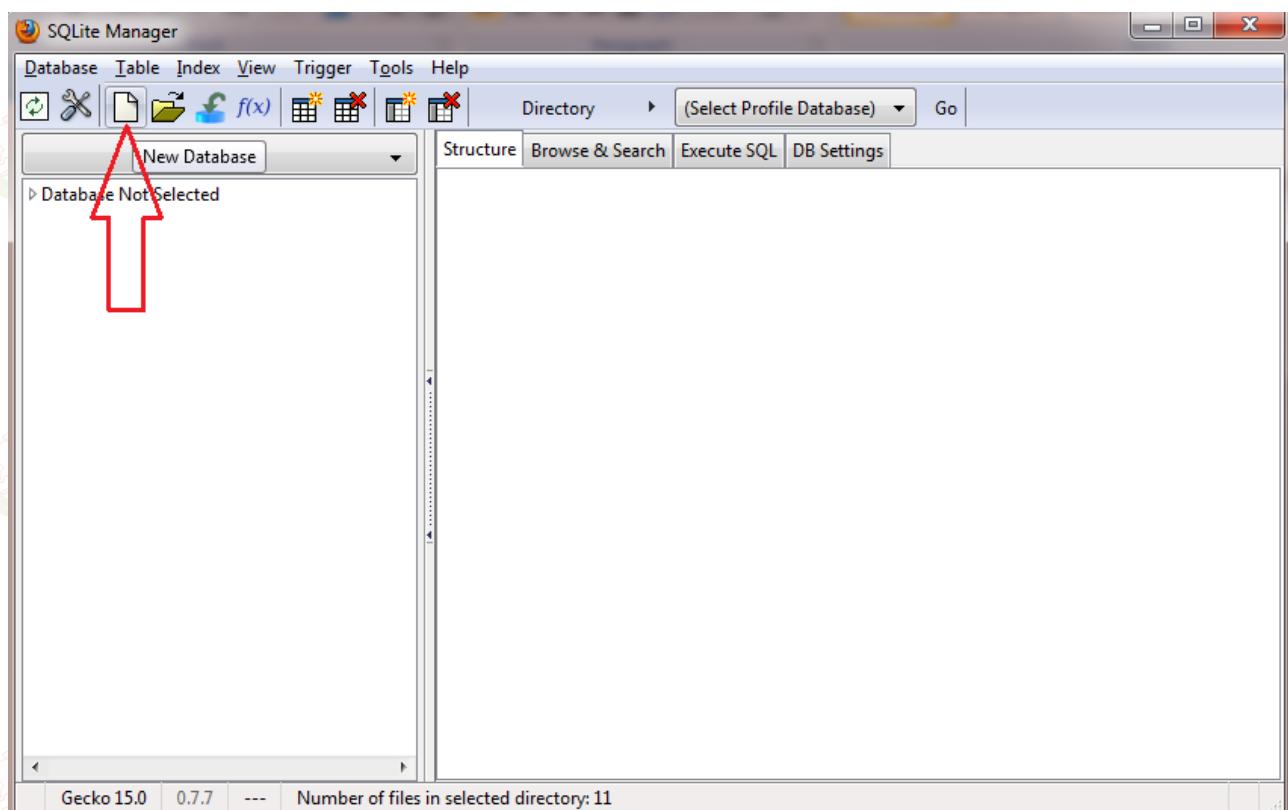
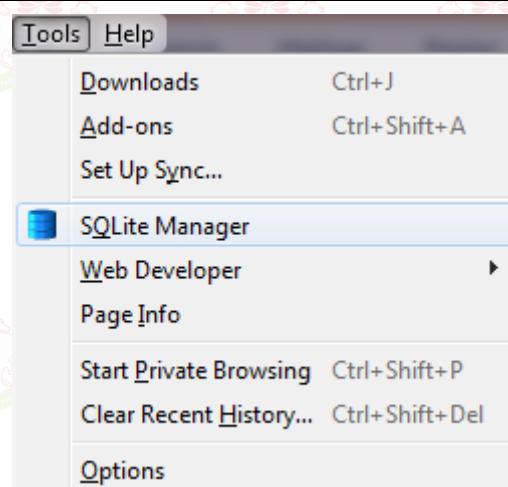




CÀI ĐẶT & QUẢN TRỊ SQLITE TRONG FIREFOX

Vào <https://addons.mozilla.org/vi/firefox/addon/sqlite-manager/>







The screenshot shows the SQLite Manager interface. At the top, a sidebar lists the database 'qlnv.sqlite' and its contents: 'Master Table (1)', 'Tables (0)', 'Views (0)', 'Indexes (0)', and 'Triggers (0)'. A 'Create Table' button is highlighted in the 'Tables (0)' section. Below this, a modal window titled 'SQLite Manager - Create Table' is open. It displays the 'Database' dropdown set to 'main' and the 'Table Name' input field set to 'nhanvien'. Underneath, there are two checkboxes: 'Temporary table' and 'If Not Exists'. The main area is titled 'Define Columns' and contains a table with six columns: 'Column Name', 'Data Type', 'Primary Key?', 'Autoinc?', 'Allow Null?', 'Unique?', and 'Default Value'. The columns defined are:

Column Name	Data Type	Primary Key?	Autoinc?	Allow Null?	Unique?	Default Value
manv	VARCHAR	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	
hoten	VARCHAR	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
ngaysinh	DATETIME	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
noisinh	VARCHAR	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
hinh	VARCHAR	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
		<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	

At the bottom right of the modal are 'OK' and 'Cancel' buttons.



SQLite Manager - C:\android\qlnv.sqlite

Database Table Index View Trigger Tools Help

qlnv.sqlite

Master Table (1)

Tables (1) nhanvien

Views (0)

Indexes (1)

Triggers (0)

Structure Browse & Search Execute SQL DB Settings

TABLE nhanvien Search Show All Add Duplicate

rowid manv hoten ngaysinh Add noisinh

SQLite 3.7.12.1 Gecko 15.0 0.7.7 Exclusive Number of files in selected directory: 11 ET: 0 ms

Add New Record

Table Name: nhanvien

Enter Field Values

1. manv (VARCHAR)	001	↶ ↻
2. hoten (VARCHAR)	Nguyễn Văn Tèo	↶ ↻
3. ngaysinh (DATETIME)	12-12-1992	↶ ↻
4. noisinh (VARCHAR)	tphcm	↶ ↻
5. hinh (VARCHAR)	teo.png	↶ ↻

OK Cancel



SQLite Manager - C:\android\qlnv.sqlite

Database Table Index View Trigger Tools Help

qinv.sqlite

Master Table (1)
Tables (1)
nhanvien
Views (0)
Indexes (1)
Triggers (0)

Structure Browse & Search Execute SQL DB Settings

TABLE nhanvien Search Show All Add Duplicate Edit Delete

rowid	manv	hoten	ngaysinh	noisinh	hinh
1	001	Nguyễn Văn Tèo	12-12-1992	tphcm	teo.png
2	002	Nguyễn Văn Tí	8-9-1998	hà nội	ti.png

1 to 2 of 2

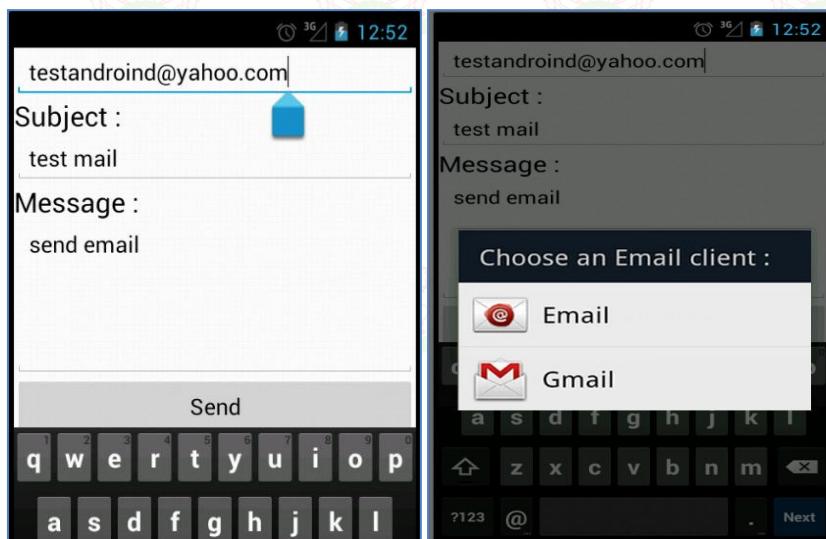
SQLite 3.7.12.1 Gecko 15.0 0.7.7 Exclusive Number of files in selected directory: 11 ET: 0 ms



SEND EMAIL

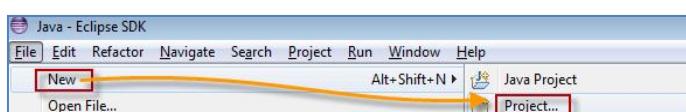
1. Giới thiệu

- Ứng dụng yêu cầu chúng ta phải có Mail Client để gửi mail đến Mail Server ngoài Internet.
- Dữ liệu nhập sẽ gửi đến Mail Client và Mail Client sẽ gửi mail đi.

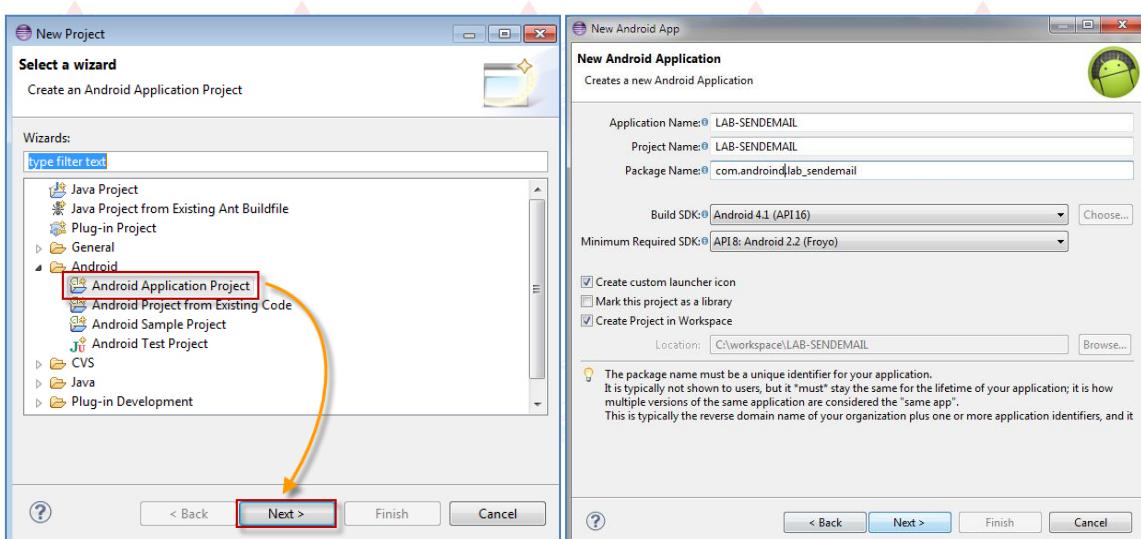


2. Tạo Project

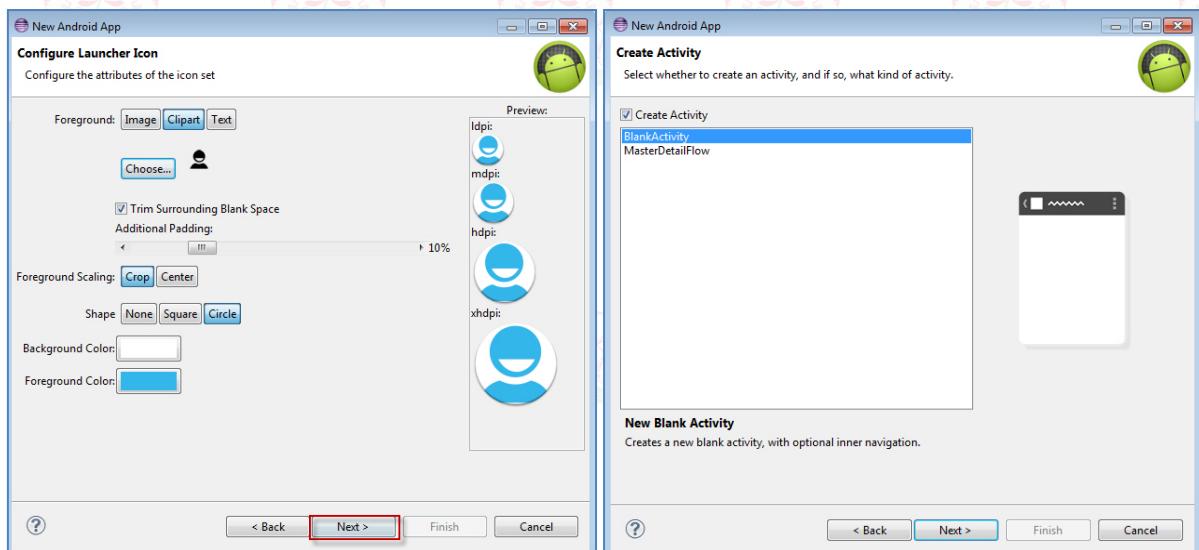
- Chọn menu : File → New → Project.



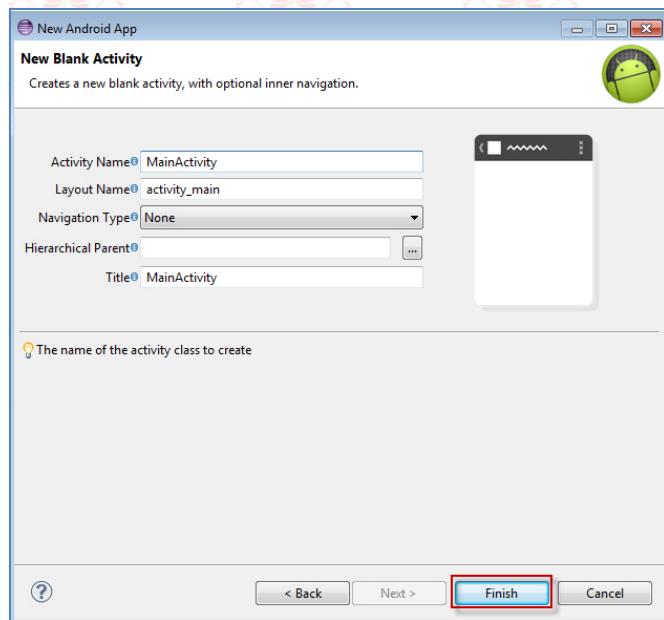
- Chọn Androind Application project → Next → Khai báo Project như sau.



- Hộp thoại Configuration Laucher Icon → Tùy chọn Icon → Next.
- Hộp thoại Create Activity : Chọn Activity : Blank Activity → Next.

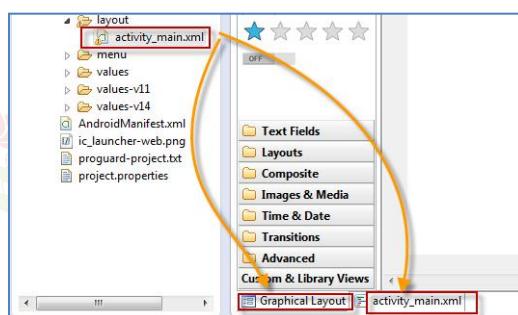


- Khai báo Activity chính.

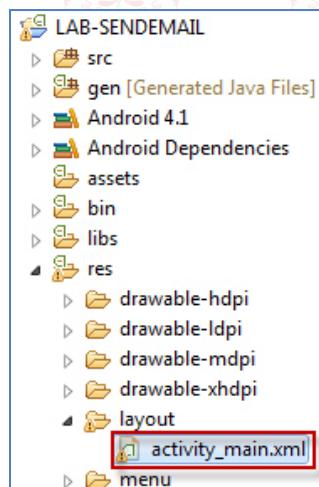


3. Thiết kế giao diện

- Sử dụng file : activity_main.xml trong thư mục layout để thiết kế giao diện.
- Có hai dạng : Graphical layout (Giao diện đồ họa) và *.xml (Code).



- Thiết kế giao diện theo code như sau (Mở file main.xml).



- Viết code trang main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/linearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textViewPhoneNo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="To : "
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <EditText
        android:id="@+id/editTextTo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress" >
        <requestFocus />
    </EditText>

    <TextView
        android:id="@+id/textViewSubject"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Subject : "
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <EditText
        android:id="@+id/editTextSubject"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        >
    </EditText>

    <TextView
        android:id="@+id/textViewMessage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

```



```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Message :"
        android:textAppearance="?android:attr/textAppearanceLarge" />
```

<EditText

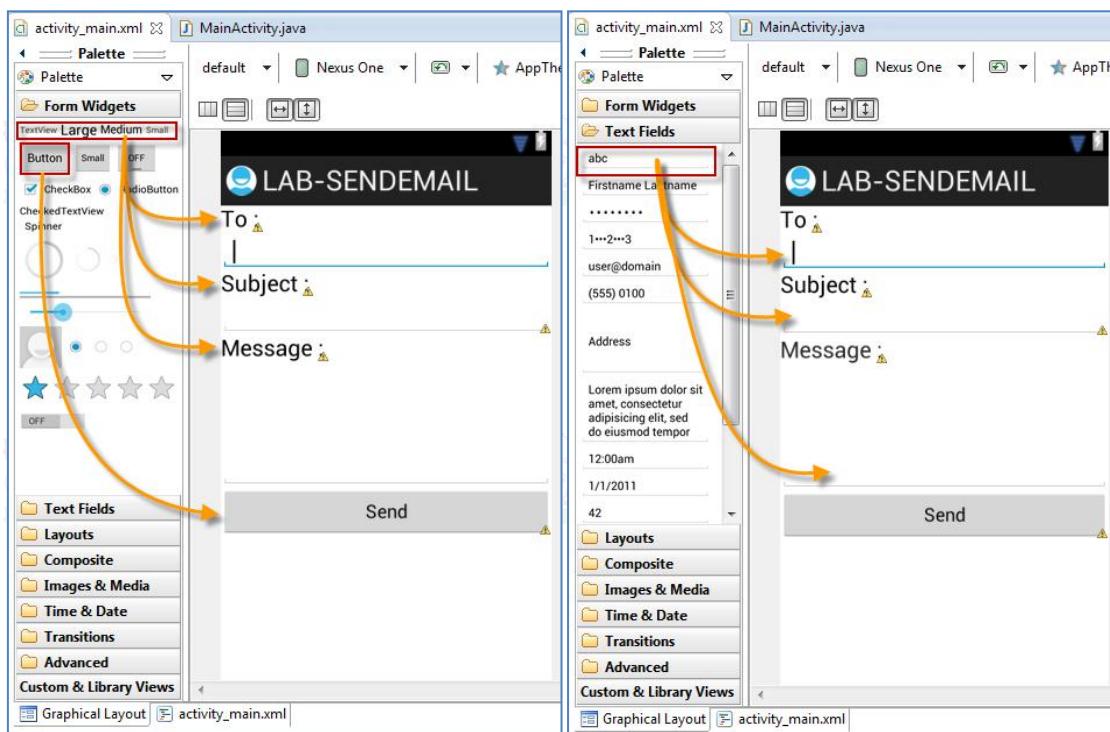
```
    android:id="@+id/editTextMessage"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="top"
    android:inputType="textMultiLine"
    android:lines="5" />
```

<Button

```
    android:id="@+id/buttonSend"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Send" />
```

</LinearLayout>

- Hoặc có thể tạo giao diện theo đồ họa như sau.





4. Viết code trang Activity.java

```
activity_main.xml>MainActivity.java
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends Activity {
    Button buttonSend;
    EditText textTo;
    EditText textSubject;
    EditText textMessage;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        buttonSend = (Button) findViewById(R.id.buttonSend);
        textTo = (EditText) findViewById(R.id.editTextTo);
        textSubject = (EditText) findViewById(R.id.editTextSubject);
        textMessage = (EditText) findViewById(R.id.editTextMessage);
        buttonSend.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                String to = textTo.getText().toString();
                String subject = textSubject.getText().toString();
                String message = textMessage.getText().toString();
                Intent email = new Intent(Intent.ACTION_SEND);
                email.putExtra(Intent.EXTRA_EMAIL, new String[]{to});
                //email.putExtra(Intent.EXTRA_CC, new String[]{to});
                //email.putExtra(Intent.EXTRA_BCC, new String[]{to});
                email.putExtra(Intent.EXTRA_SUBJECT, subject);
                email.putExtra(Intent.EXTRA_TEXT, message);
                //need this to prompts email client only
                email.setType("message/rfc822");
                startActivity(Intent.createChooser(email, "Choose an Email client :"));
            }
        });
    }
}
```

5. Run ứng dụng

- Ứng dụng yêu cầu chúng ta phải có Mail Client để gửi mail đến Mail Server ngoài Internet.
- Dữ liệu nhập sẽ gửi đến Mail Client và Mail Client sẽ gửi mail đi.

