**Task 4. Tìm hiểu các đối tượng Giao diện Android Phần 3.**

GV. Nguyễn Văn Thành

**I. TÌM HIỂU CÁC ĐỐI TƯỢNG**

**1.** **AndroidManifest.xml**

a. Có ý nghĩa như thế nào trong project Android?

**AndroidManifest** là file để bản kê khai trình bày những thông tin thiết yếu về ứng dụng của bạn với hệ thống Android, thông tin mà hệ thống phải có trước khi có thể chạy bất kỳ mã nào của ứng dụng.Hay nói cách khác đây là file dùng để config những thuộc tính cho ứng dụng mà khi ứng dụng khởi chạy hệ điều hành có thể hiểu được và xử lí.

**Những công việc mà tập tin manifest cần phải thực hiện :**

– Khai báo tên của package cho ứng dụng. Tên của package là định danh duy nhất của ứng dụng

– Mô tả các thành phần (component) của ứng dụng: activity, services, broadcast receivers, và content providers mà ứng dụng có. Khai báo tên các lớp mà cài đặt các thành phần này và các tác vụ mà các thành phần này có thể xử lý (ví dụ các thông điệp một Intent có thể xử lý). Những khai báo này cho phép hệ điều hành Android biết các thành phần này có thể có thể thực hiện được tác vụ gì và điều kiện để chúng có thể thực thi.

– Xác định tiến trình (process) sẽ chứa những thành phần (component) nào.

– Khai báo các quyền mà ứng dụng cần phải có để truy xuất các thành phần của API và giao tiếp với các ứng dụng khác.

– Khai báo các quyền mà yêu cầu ứng dụng khác cần phải có để có thể giao tiếp được với các component của ứng dụng này.

– Mô tả phiên bản API tối thiểu mà ứng dụng có thể chạy.

– Liệt kê các thư viện mà ứng dụng cần phải kết nối.

b. Thuộc tính **package="……….."** có ý nghĩa gì?

**package:** chính là package name của project. Mặc định package này sẽ là giá trị giống như thuộc tính applicationId trong file build.gradle ở cấp độ module.

c. Thẻ **<uses-permission android:name= “……..” />** có ý nghĩa gì? Hãy viết dòng code báo quyền truy cập INTERNET bên dưới?

**<uses-permission />**: đây là cú pháp lệnh để xin cấp một quyền gì đó khi ứng dụng bạn tương tác với dữ liệu nào đó mà google không cho phép dùng tùy ý.Cú pháp xin cấp quyền như sau :

<uses-permission android:name="…" />

Ví dụ cú pháp lệnh xin cấp quyền truy cập Internet như sau :

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

d. Thẻ **<application>…..</application>** được sử dụng để làm gì?

**<application>… …</application>**: thẻ này chứa một số thuộc tính liên quan đến cấu hình của ứng dụng (đúng như tên gọi của nó)

e. Thuộc tính **android:icon="........"** trong thẻ **<application>** có ý nghĩa gì?

**android:icon** – Thuộc tính này là nơi bạn thiết lập icon cho ứng dụng. Icon này sẽ xuất hiện trên màn hình chính của thiết bị. Hiện tại thì icon này đang được set mặc định là file **ic\_launcher.png** để trong thư mục **res/mipmap/**.

f. Thuộc tính **android:label="........"** trong thẻ **<application>** có ý nghĩa gì?

**android:label** – Thuộc tính này trỏ đến một giá trị **string** trong resource, hoàn toàn tương tự như cách khai báo text cho **TextView**. String này chính là tên ứng dụng của chúng ta.

g. Thuộc tính **android:** **theme="........"** trong thẻ **<application>** có ý nghĩa gì?

**android:theme** – Thuộc tính này khá hay, nó giúp chúng ta định nghĩa *“giao diện chủ đề”* cho ứng dụng

*Copy một đoạn code hoàn chỉnh của AndroidManifest sau khi bạn đã sửa lại tên, icon và theme cho chương trình của mình theo yêu cầu sau:*

* Tên app: AssignmentTask4
* Theme: NoActionBar.

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

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

package="com.example.assignmenttask4">

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

<resources>

<!-- Base application theme. -->

<style name="AppTheme" parent="Theme.AppCompat.NoActionBar">

<!-- Customize your theme here. -->

<item name="colorPrimary">@color/colorPrimary</item>

<item name="colorPrimaryDark">@color/colorPrimaryDark</item>

<item name="colorAccent">@color/colorAccent</item>

</style>

</resources>

**2.** **Thẻ <style> trong thư mục res/values**

a. Ý nghĩa của thẻ <style> là gì?

***Style – Phong cách***, Phong Cách ở đây là muốn nói về cách mà bạn tạo ra một sự đồng nhất về giao diện cho ứng dụng. Như ví dụ về ***Button***  bạn muốn tạo ra các ***Button*** giống nhau ở các màn hình, để cho nó có sự đồng nhất. Khi đó việc thiết kế cho ***Button*** sao cho chúng giống nhau như vậy khiến bạn cảm thấy nhàm chán vì mất khá nhiều thời gian. Lúc đó bạn sẽ cần đến một nơi định nghĩa hết tất tần tật những thuộc tính của ***Button*** trên, rồi đến với từng ***Button*** bạn chỉ việc mang định nghĩa đó ra dùng. Thì khi đó Style chính là kiểu resource mà bạn đang tìm

b. Thuộc tính **“name”** của thẻ <style> có ý nghĩa gì?

- Đây là tên của Style

c. Thuộc tính **“parent”** của thẻ <style> có ý nghĩa gì?

- Thuộc tính này không bắt buộc. Nhưng nếu bạn muốn Style của mình được kế thừa từ một Style có sẵn (do bạn tạo ra trước đó, hay từ Style của một thư viện nào đó, hoặc Style của hệ thống), việc kế thừa này giúp bạn tận dụng lại những định nghĩa từ Style gốc, và tạo ra các định nghĩa mới bổ sung cho Style gốc còn thiếu.

d. Các thẻ <item> con nằm trong thẻ <style> để làm gì?

- Mỗi thẻ item như vậy định nghĩa ra một loại phong cách nào đó, trong đó tên của loại phong cách đó được miêu tả trong thuộc tính ***name*** của thẻ ***Style***

e. App đang dùng thẻ <style> nào? Hãy viết thêm 1 thẻ <item> để set màu nền mặc định là màu #FFFF00 cho toàn bộ các layout của app, kiểm tra lại ở các layout xem đã có sự thay đổi màu nền chưa?

**- Thẻ style được dùng trong app :**

<resources>

<!-- Base application theme. -->

<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

<!-- Customize your theme here. -->

<item name="colorPrimary">@color/colorPrimary</item>

<item name="colorPrimaryDark">@color/colorPrimaryDark</item>

<item name="colorAccent">@color/colorAccent</item>

</style>

</resources>

**- Set màu nền mặc định là màu #FFFF00 cho toàn bộ các layout của app**

<style name="backgroundColor">

<item name="android:background">#FFFF00</item>

<style>

**3. Context trong android**

a. Context trong Android là gì? Có mấy loại Context trong Android?

Context là trạng thái của app của bạn ở bất kỳ thời điểm nào.

Các loại Context trong Android :

* - Application Context
* - Activity Context

b. Activity là loại Context được dùng để làm gì?

Loại context này ở trong Activity và nó cũng gắn liền với vòng đời của Activity, nên sử dụng loại context này khi bạn truyền nó sử dụng trong phạm vi activity hoặc bạn cần context cho công việc mà gắn liền với vòng đời hiện tại của activity

c. Sử dụng Activity như thế nào? Trong chương trình đang sử dụng mặc định Activity nào?

[***Activity***](https://developer.android.com/reference/android/app/Activity.html?hl=vi) là một thành phần ứng dụng cung cấp một màn hình mà với nó người dùng có thể tương tác để thực hiện một điều gì đó.

Sử dụng Activity để bắt sự kiện khi bấm vào ***View***

Hãy vận dụng các kiến thức tìm hiểu để:

* Tạo một giao diện ***layout1.xml*** có nền màu xanh dương và hiển thị nó lên màn hình máy ảo (Sử dụng activity mặc định đã có sẵn khi tạo app (Hình như là MainActivity)).

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

<android.support.constraint.ConstraintLayout

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@android:color/holo\_blue\_dark"

tools:context=".MainActivity">

</android.support.constraint.ConstraintLayout>

* Tạo một class activity mới để hiển thị ***layout1.xml*** *(không sử dụng MainActivity.class nữa).*

package com.example.layouttask4\_hstc;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.layout1);

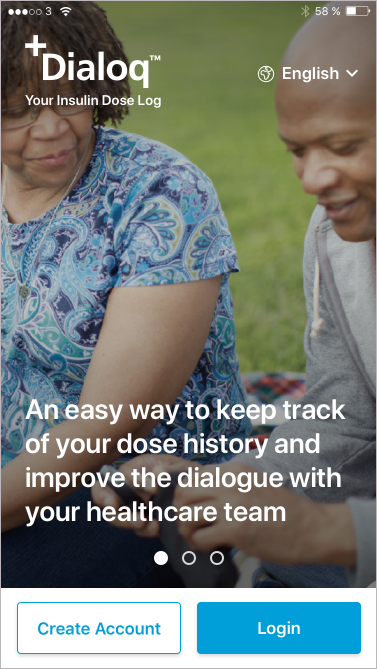
}

}

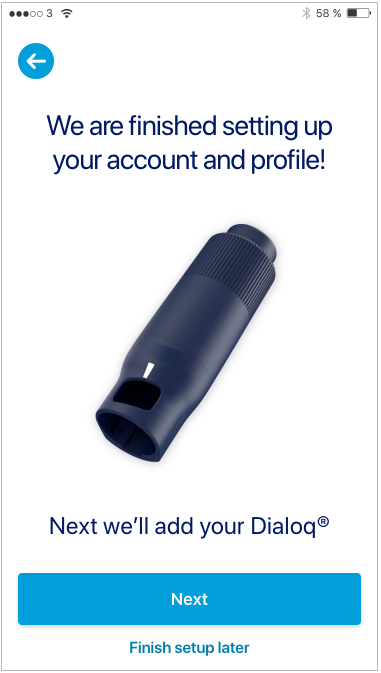
**II. THỰC HÀNH XÂY DỰNG GIAO DIỆN**

**- Chú ý: Các layout sau và hiển thị lên máy ảo, mỗi giao diện sẽ đi kèm với 1 activitiy**

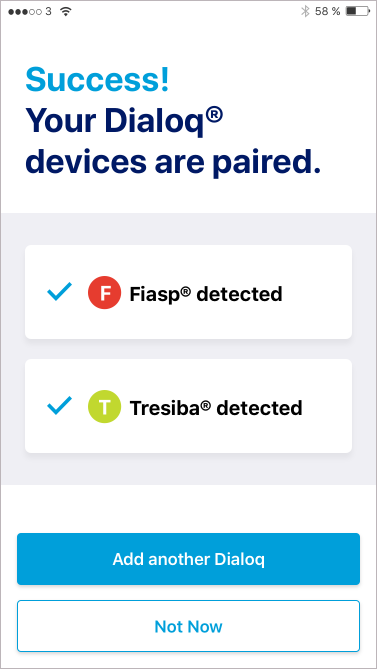
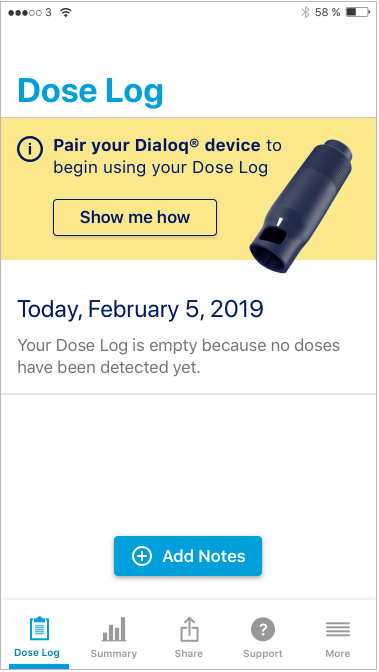
a. act\_landing\_page.xml



b. frg\_introduction.xml c.frg\_account\_setup.xml

d. frg\_pair\_success.xml e. frg\_not\_pair.xml

**III. Nhóm trưởng tạo respository mới cho buổi học 09/05 và các thành viên clone về, tạo nhánh sẵn sàng để học.**

**IV. QUY ĐỊNH BÁO CÁO**

**1. Bảng báo cáo (Làm bằng excel) các hạng mục trong Task 3 như sau:**

**Team…..**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hạng mục Task3** | **Chịu trách nhiệm chính** | **Tình trạng** | **Output**  **(tên nhánh làm)** | **Vấn đề gặp phải** |
| Tìm hiểu đối tượng TextView-EditText-Button | Tất cả | Hoàn thành | Bảng đặc tả (docx) | Không |
| Giao diện note.xml | Hoàng Đức Mạnh | Hoàn thành | feature\_task3/manhhd | Không |
|  |  |  |  |  |

**2. Nộp như sau:**

**-** Truy cập link: <https://drive.google.com/drive/folders/1C9nKkbug__buVZNhElF0v9HFvypr0-L4?usp=sharing>

- Tạo 1 folder cho team mình và upload output mà các em làm lên đó nhé.

Chúc các em thành công

GV. Thành Nguyễn