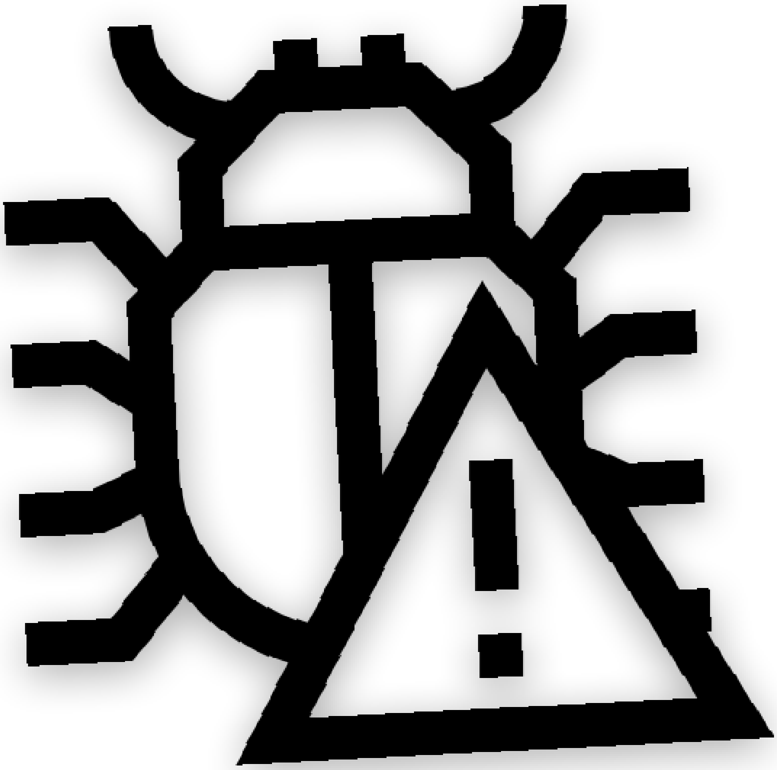
Bug Report



Generated by : Charchit Subedi

Date : 2022/june/05

Time : 9:39 am

Application : Tianmei

Content

Pg.No

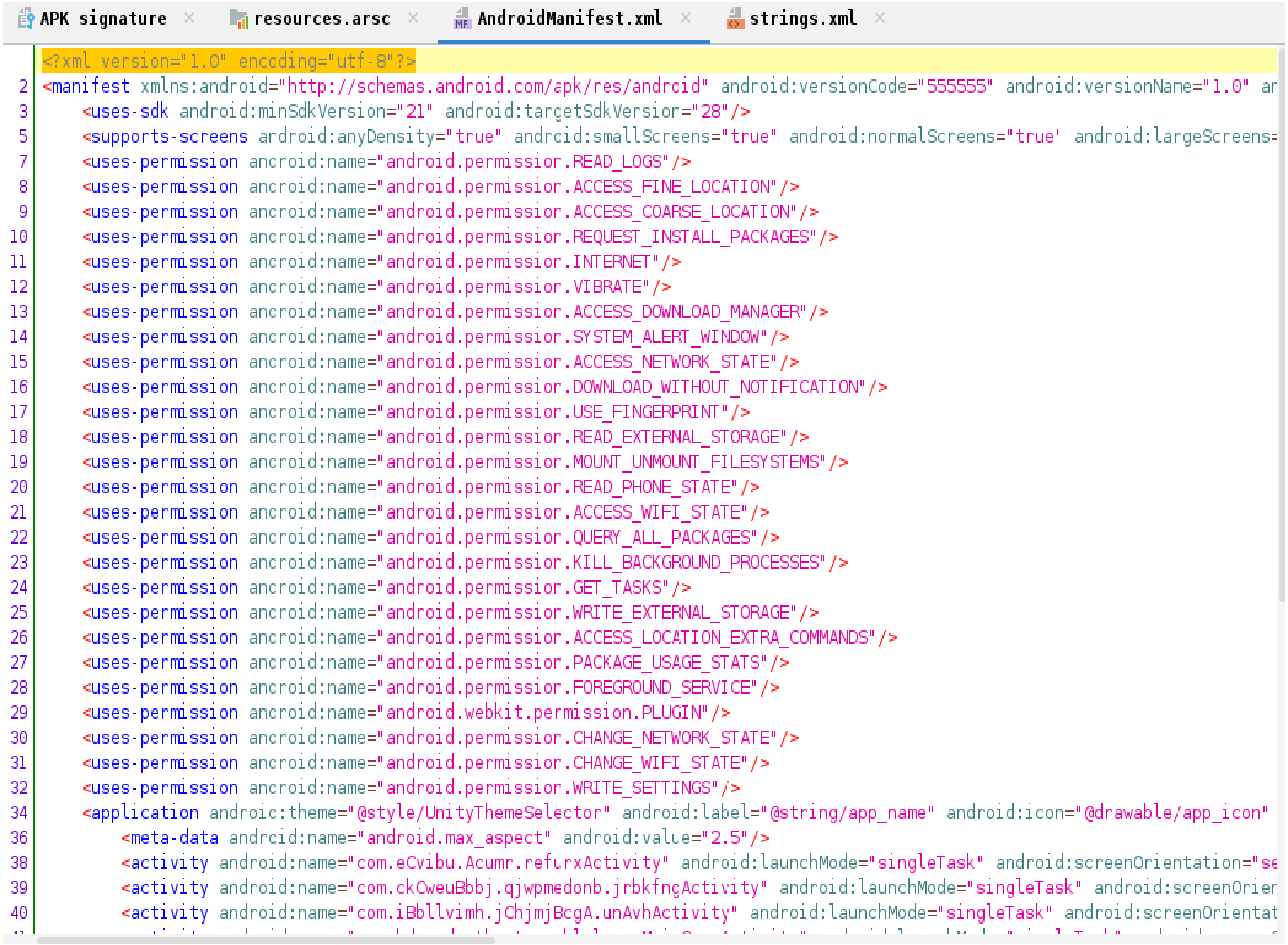
Introduction to Tianmei

It is the chiniese application , which is hosted by the **87715.xyz**

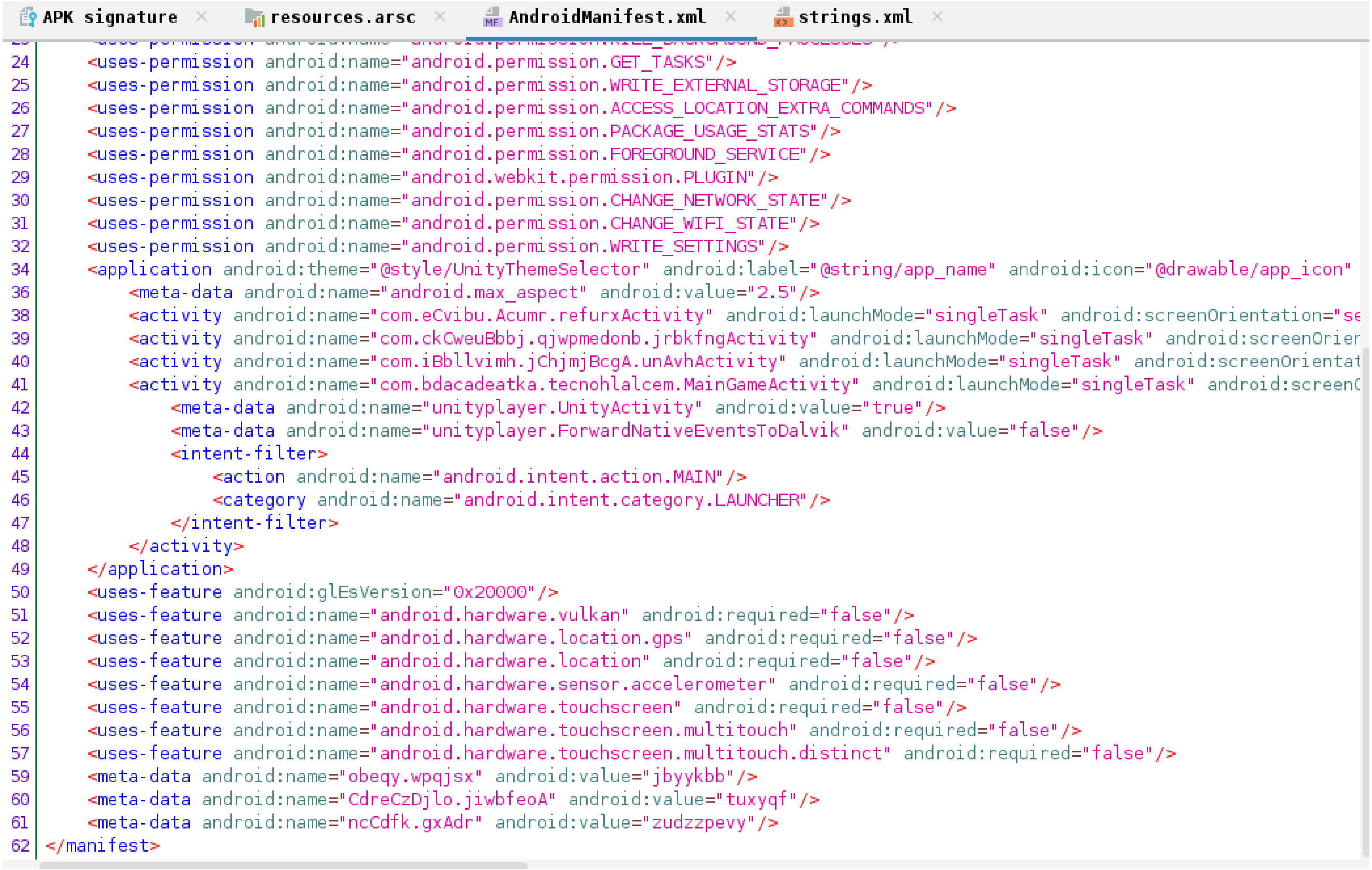
Introduction to JADX

JADX is a decompilation tool that can produce Java Source code from Dex and Apk files, being capable of providing human-readable java classes, it reverses AndroidManifest.xml files which contains all configuration details for the app & many other resources present as part of the '.apk'.

USE OF JADX Tools

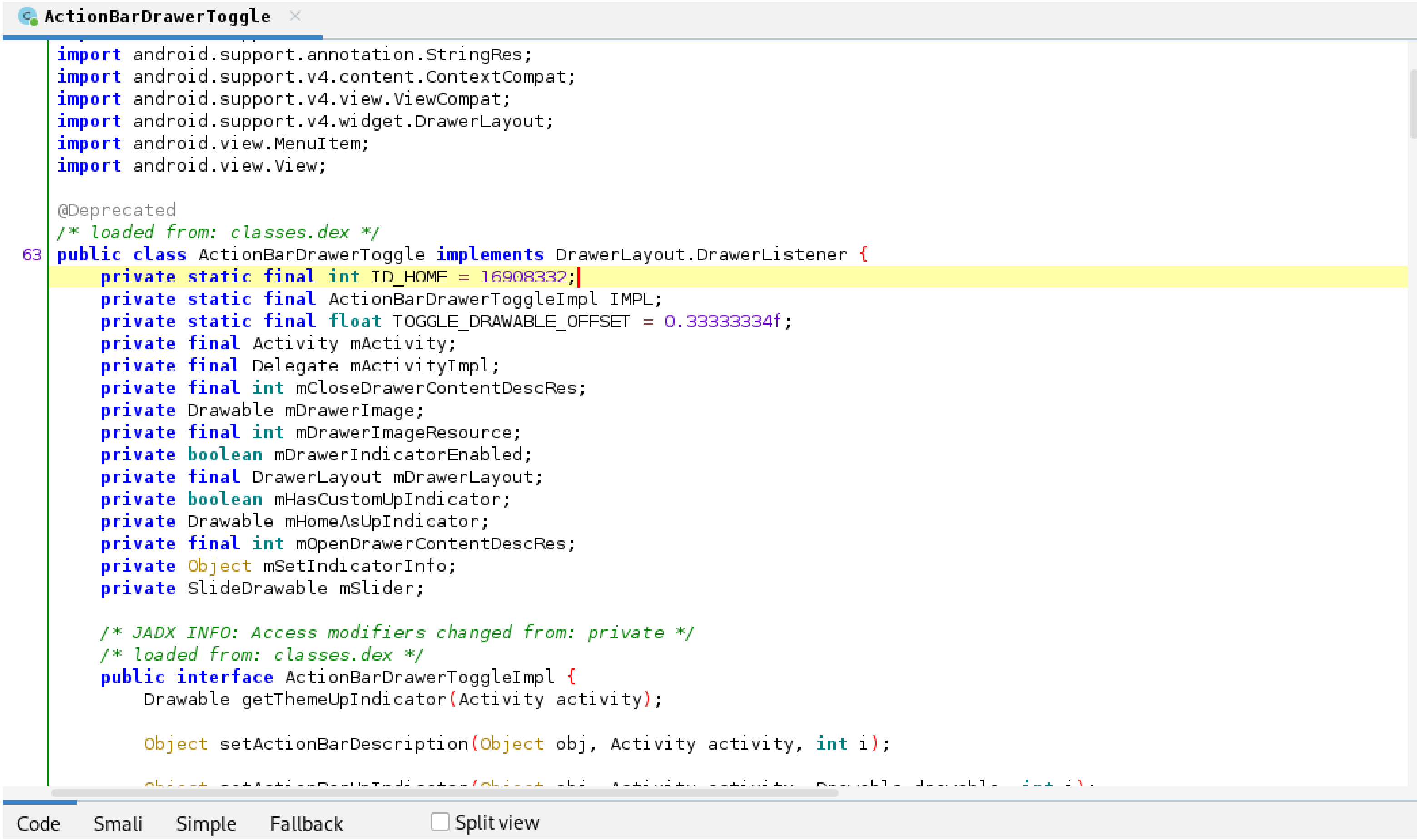


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="555555" android:versionName="1.0" ar
    <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="28"/>
    <supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true" android:largeScreens=
    <uses-permission android:name="android.permission.READ_LOGS"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.ACCESS_DOWNLOAD_MANAGER"/>
    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION"/>
    <uses-permission android:name="android.permission.USE_FINGERPRINT"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.QUERY_ALL_PACKAGES"/>
    <uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
    <uses-permission android:name="android.permission.GET_TASKS"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"/>
    <uses-permission android:name="android.permission.PACKAGE_USAGE_STATS"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.webkit.permission.PLUGIN"/>
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
    <application android:theme="@style/UnityThemeSelector" android:label="@string/app_name" android:icon="@drawable/app_icon"
        <meta-data android:name="android.max_aspect" android:value="2.5"/>
        <activity android:name="com.eCvibu.Acumr.refurxActivity" android:launchMode="singleTask" android:screenOrientation="se
        <activity android:name="com.ckCweuBbbj.qjwpmedonb.jrbkfngActivity" android:launchMode="singleTask" android:screenOrier
        <activity android:name="com.iBbllvimh.jChjmjBcgA.unAvhActivity" android:launchMode="singleTask" android:screenOrientat
```



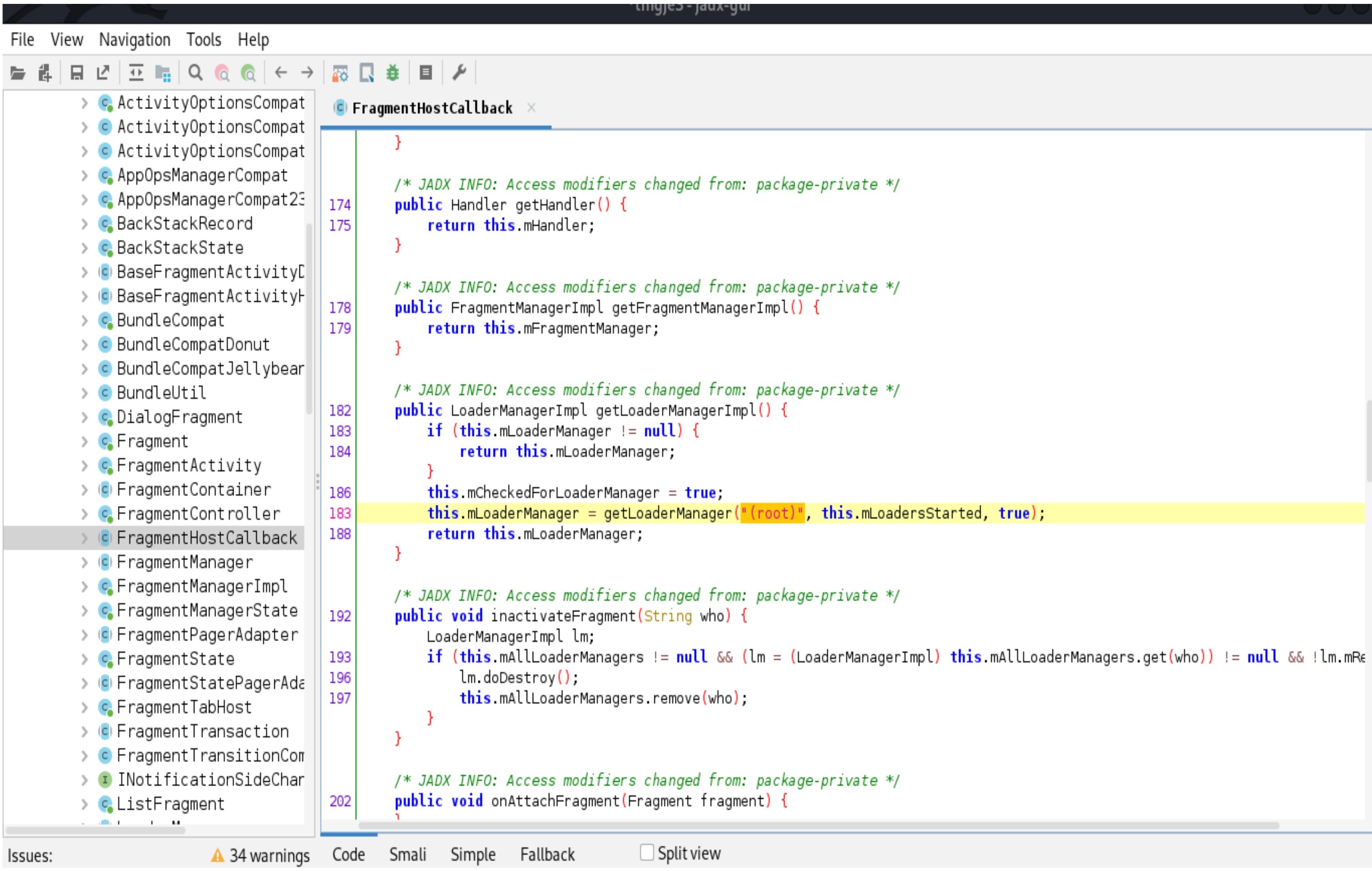
```xml
    <uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
    <uses-permission android:name="android.permission.GET_TASKS"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"/>
    <uses-permission android:name="android.permission.PACKAGE_USAGE_STATS"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.webkit.permission.PLUGIN"/>
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
    <application android:theme="@style/UnityThemeSelector" android:label="@string/app_name" android:icon="@drawable/app_icon"
        <meta-data android:name="android.max_aspect" android:value="2.5"/>
        <activity android:name="com.eCvibu.Acumr.refurxActivity" android:launchMode="singleTask" android:screenOrientation="se
        <activity android:name="com.ckCweuBbbj.qjwpmedonb.jrbkfngActivity" android:launchMode="singleTask" android:screenOrier
        <activity android:name="com.iBbllvimh.jChjmjBcgA.unAvhActivity" android:launchMode="singleTask" android:screenOrientat
        <activity android:name="com.bdacadeatka.tecnohlalcem.MainGameActivity" android:launchMode="singleTask" android:screenC
            <meta-data android:name="unityplayer.UnityActivity" android:value="true"/>
            <meta-data android:name="unityplayer.ForwardNativeEventsToDalvik" android:value="false"/>
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
    <uses-feature android:glEsVersion="0x20000"/>
    <uses-feature android:name="android.hardware.vulkan" android:required="false"/>
    <uses-feature android:name="android.hardware.location.gps" android:required="false"/>
    <uses-feature android:name="android.hardware.location" android:required="false"/>
    <uses-feature android:name="android.hardware.sensor.accelerometer" android:required="false"/>
    <uses-feature android:name="android.hardware.touchscreen" android:required="false"/>
    <uses-feature android:name="android.hardware.touchscreen.multitouch" android:required="false"/>
    <uses-feature android:name="android.hardware.touchscreen.multitouch.distinct" android:required="false"/>
    <meta-data android:name="obeqy.wpqjsx" android:value="jbyykbb"/>
    <meta-data android:name="CdreCzDjlo.jiwbfeoA" android:value="tuxyqf"/>
    <meta-data android:name="ncCdfk.gxAdr" android:value="zudzzpevy"/>
</manifest>
```

As we can see that the application is taking all the permission from our android device which is not necessary , If the application is taking Permission whithout any reason then we can say that this is Malicious Application . As we can see that the application is asking for download without permission which is not activated in any geniun application. The application is also taking our hardware permission which may cause our phone damage.



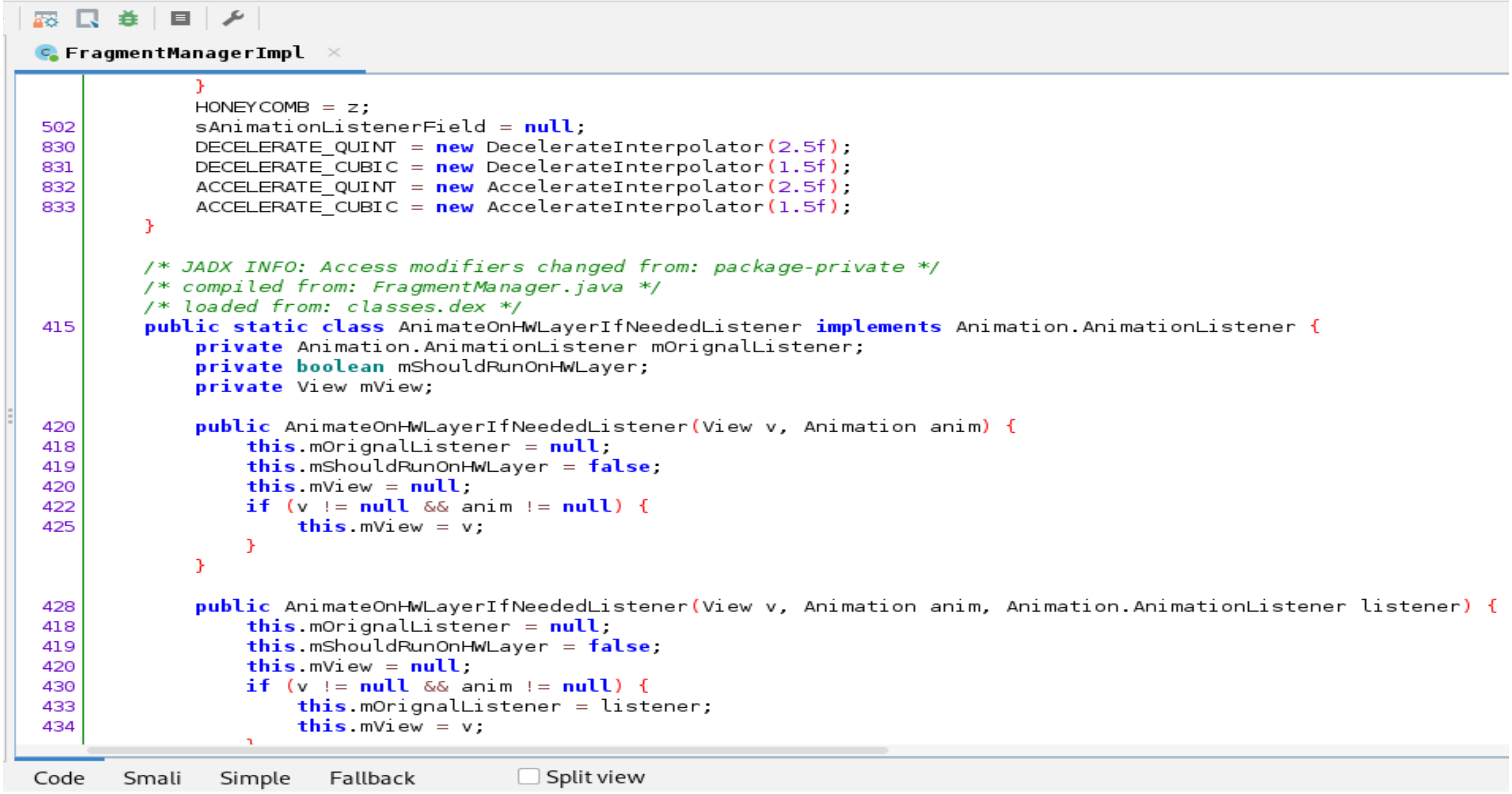
```
import android.support.annotation.StringRes;
import android.support.v4.content.ContextCompat;
import android.support.v4.view.ViewCompat;
import android.support.v4.widget.DrawerLayout;
import android.view.MenuItem;
import android.view.View;

@Deprecated
/* loaded from: classes.dex */
public class ActionBarDrawerToggle implements DrawerLayout.DrawerListener {
    private static final int ID_HOME = 16908332;
    private static final ActionBarDrawerToggleImpl IMPL;
    private static final float TOGGLE_DRAWABLE_OFFSET = 0.33333334f;
    private final Activity mActivity;
    private final Delegate mActivityImpl;
    private final int mCloseDrawerContentDescRes;
    private Drawable mDrawerImage;
    private final int mDrawerImageResource;
    private boolean mDrawerIndicatorEnabled;
    private final DrawerLayout mDrawerLayout;
    private boolean mHasCustomUpIndicator;
    private Drawable mHomeAsUpIndicator;
    private final int mOpenDrawerContentDescRes;
    private Object mSetIndicatorInfo;
    private SlideDrawable mSlider;

    /* JADX INFO: Access modifiers changed from: private */
    /* loaded from: classes.dex */
    public interface ActionBarDrawerToggleImpl {
        Drawable getThemeUpIndicator(Activity activity);

        Object setActionBarDescription(Object obj, Activity activity, int i);

        Object setActionBarUpIndicator(Object obj, Activity activity, Drawable drawable, int i);
```

In the aboove picture the Home_Id is given which is 16908332.



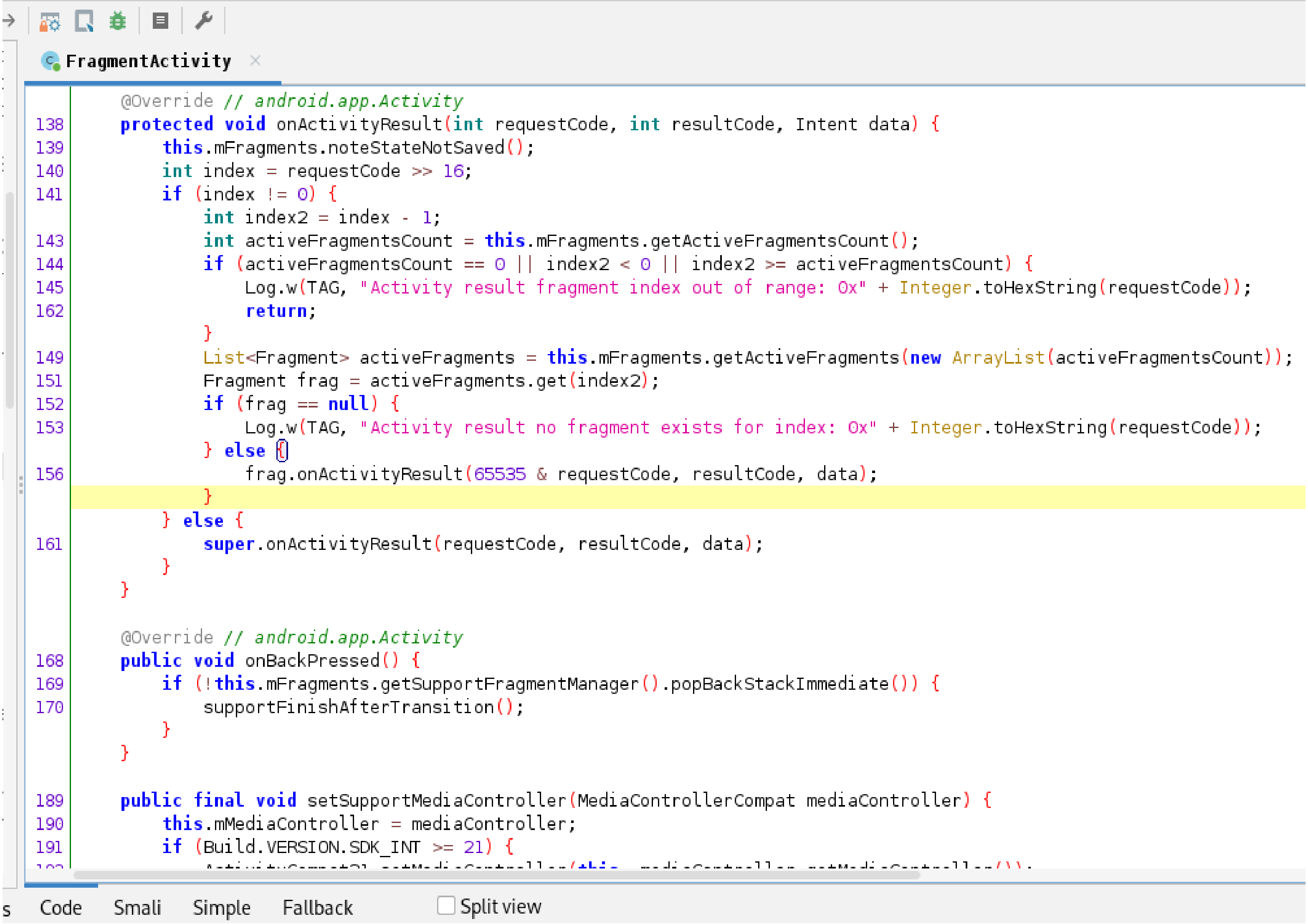
```
    }

    /* JADX INFO: Access modifiers changed from: package-private */
    public Handler getHandler() {
        return this.mHandler;
    }

    /* JADX INFO: Access modifiers changed from: package-private */
    public FragmentManagerImpl getFragmentManagerImpl() {
        return this.mFragmentManager;
    }

    /* JADX INFO: Access modifiers changed from: package-private */
    public LoaderManagerImpl getLoaderManagerImpl() {
        if (this.mLoaderManager != null) {
            return this.mLoaderManager;
        }
        this.mCheckedForLoaderManager = true;
        this.mLoaderManager = getLoaderManager("(root)", this.mLoadersStarted, true);
        return this.mLoaderManager;
    }

    /* JADX INFO: Access modifiers changed from: package-private */
    public void inactivateFragment(String who) {
        LoaderManagerImpl lm;
        if (this.mAllLoaderManagers != null && (lm = (LoaderManagerImpl) this.mAllLoaderManagers.get(who)) != null && !lm.mRe
            lm.doDestroy();
            this.mAllLoaderManagers.remove(who);
        }
    }

    /* JADX INFO: Access modifiers changed from: package-private */
    public void onAttachFragment(Fragment fragment) {
```

The apllication is taking root permission also .



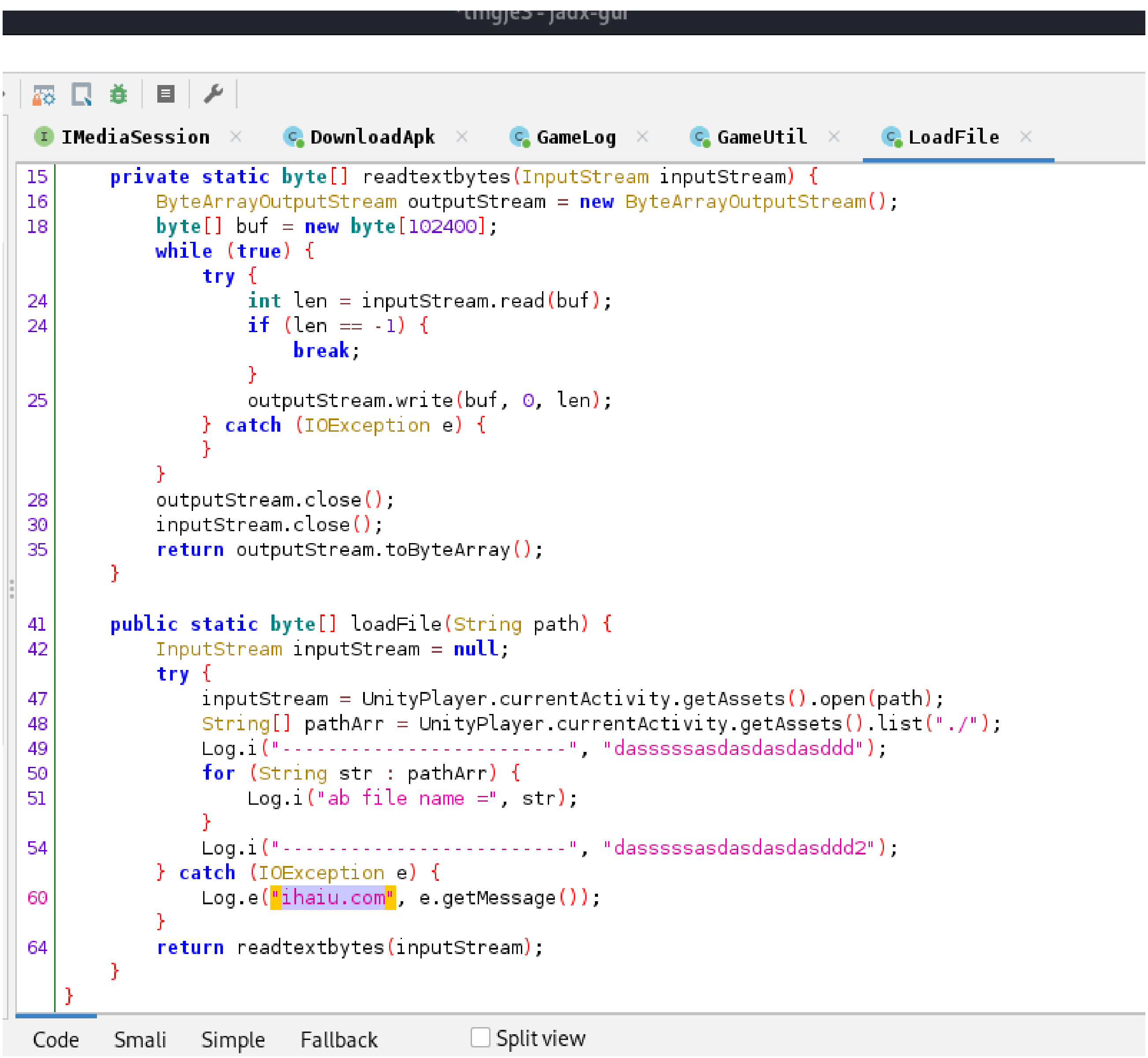
```
            }
            HONEYCOMB = z;
            sAnimationListenerField = null;
            DECELERATE_QUINT = new DecelerateInterpolator(2.5f);
            DECELERATE_CUBIC = new DecelerateInterpolator(1.5f);
            ACCELERATE_QUINT = new AccelerateInterpolator(2.5f);
            ACCELERATE_CUBIC = new AccelerateInterpolator(1.5f);
        }

        /* JADX INFO: Access modifiers changed from: package-private */
        /* compiled from: FragmentManager.java */
        /* loaded from: classes.dex */
        public static class AnimateOnHWLayerIfNeededListener implements Animation.AnimationListener {
            private Animation.AnimationListener mOrignalListener;
            private boolean mShouldRunOnHWLayer;
            private View mView;

            public AnimateOnHWLayerIfNeededListener(View v, Animation anim) {
                this.mOrignalListener = null;
                this.mShouldRunOnHWLayer = false;
                this.mView = null;
                if (v != null && anim != null) {
                    this.mView = v;
                }
            }

            public AnimateOnHWLayerIfNeededListener(View v, Animation anim, Animation.AnimationListener listener) {
                this.mOrignalListener = null;
                this.mShouldRunOnHWLayer = false;
                this.mView = null;
                if (v != null && anim != null) {
                    this.mOrignalListener = listener;
                    this.mView = v;
```

As we can see that the listener is also mentioned in the code.



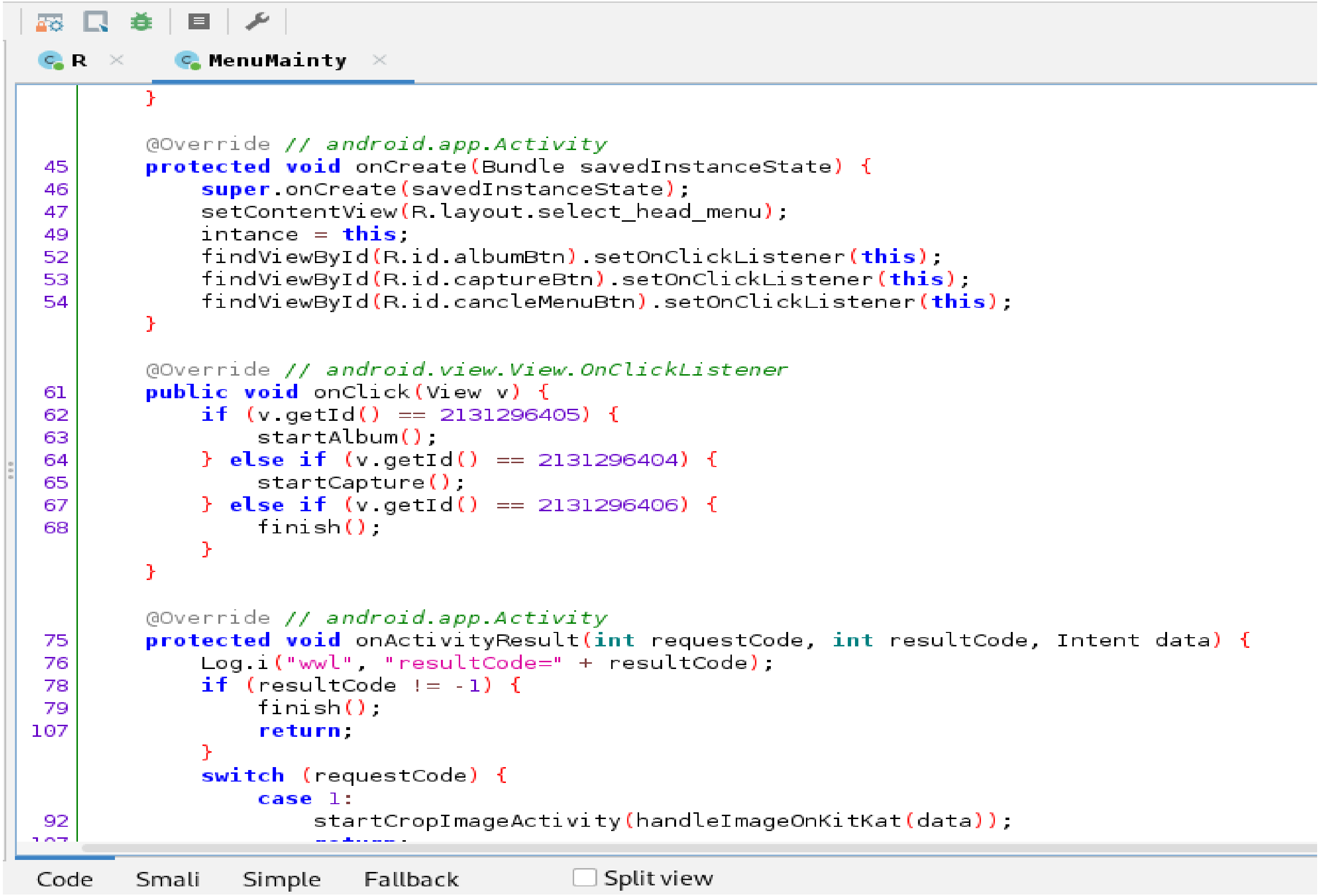
```
    @Override // android.app.Activity
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        this.mFragments.noteStateNotSaved();
        int index = requestCode >> 16;
        if (index != 0) {
            int index2 = index - 1;
            int activeFragmentsCount = this.mFragments.getActiveFragmentsCount();
            if (activeFragmentsCount == 0 || index2 < 0 || index2 >= activeFragmentsCount) {
                Log.w(TAG, "Activity result fragment index out of range: 0x" + Integer.toHexString(requestCode));
                return;
            }
            List<Fragment> activeFragments = this.mFragments.getActiveFragments(new ArrayList(activeFragmentsCount));
            Fragment frag = activeFragments.get(index2);
            if (frag == null) {
                Log.w(TAG, "Activity result no fragment exists for index: 0x" + Integer.toHexString(requestCode));
            } else {
                frag.onActivityResult(65535 & requestCode, resultCode, data);
            }
        } else {
            super.onActivityResult(requestCode, resultCode, data);
        }
    }

    @Override // android.app.Activity
    public void onBackPressed() {
        if (!this.mFragments.getSupportFragmentManager().popBackStackImmediate()) {
            supportFinishAfterTransition();
        }
    }

    public final void setSupportMediaController(MediaControllerCompat mediaController) {
        this.mMediaController = mediaController;
        if (Build.VERSION.SDK_INT >= 21) {
            ActivityCompat21.setMediaController(this, mediaController.getMediaController());
```

As we can see that the listener is activated on port 65535.



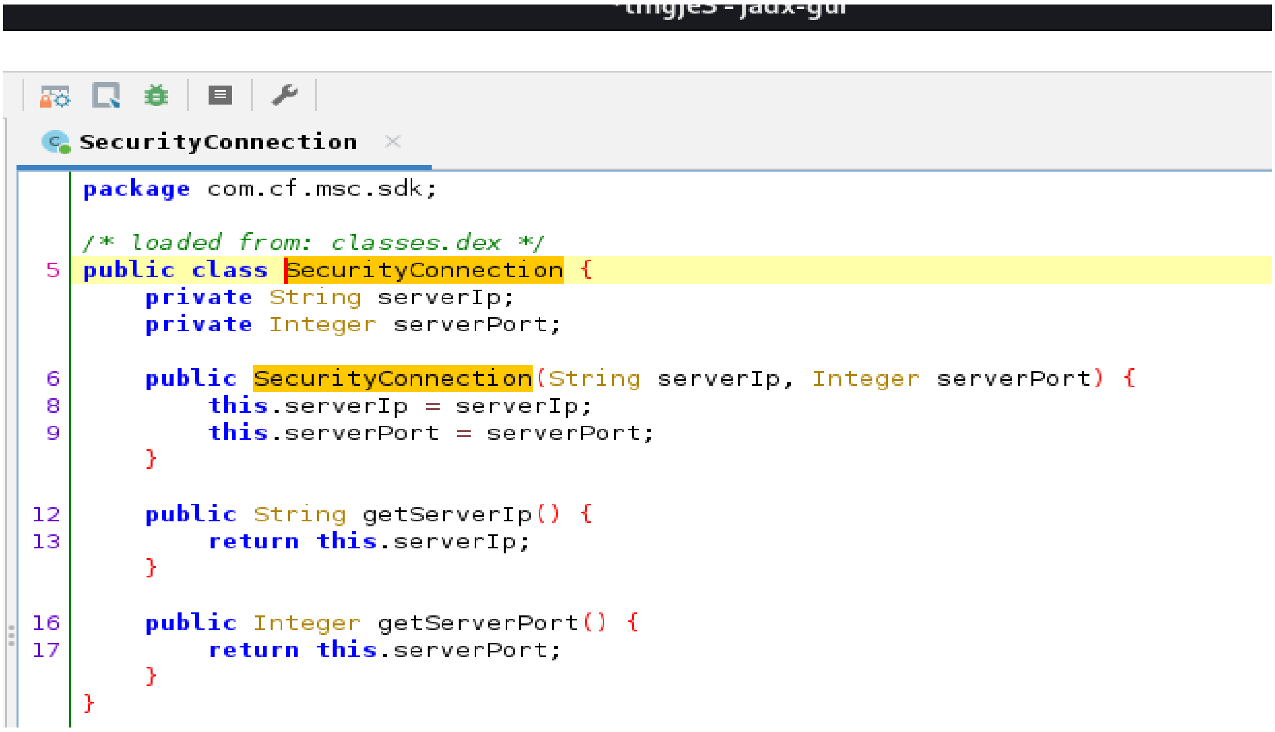
```java
    private static byte[] readtextbytes(InputStream inputStream) {
        ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
        byte[] buf = new byte[102400];
        while (true) {
            try {
                int len = inputStream.read(buf);
                if (len == -1) {
                    break;
                }
                outputStream.write(buf, 0, len);
            } catch (IOException e) {
            }
        }
        outputStream.close();
        inputStream.close();
        return outputStream.toByteArray();
    }

    public static byte[] loadFile(String path) {
        InputStream inputStream = null;
        try {
            inputStream = UnityPlayer.currentActivity.getAssets().open(path);
            String[] pathArr = UnityPlayer.currentActivity.getAssets().list("./");
            Log.i("-------------------------", "dasssssasdasdasdasddd");
            for (String str : pathArr) {
                Log.i("ab file name =", str);
            }
            Log.i("-------------------------", "dasssssasdasdasdasddd2");
        } catch (IOException e) {
            Log.e("ihaiu.com", e.getMessage());
        }
        return readtextbytes(inputStream);
    }
}
```

We can see that there one link is given which is ihaiu.com you can go to link by clicking the url up.



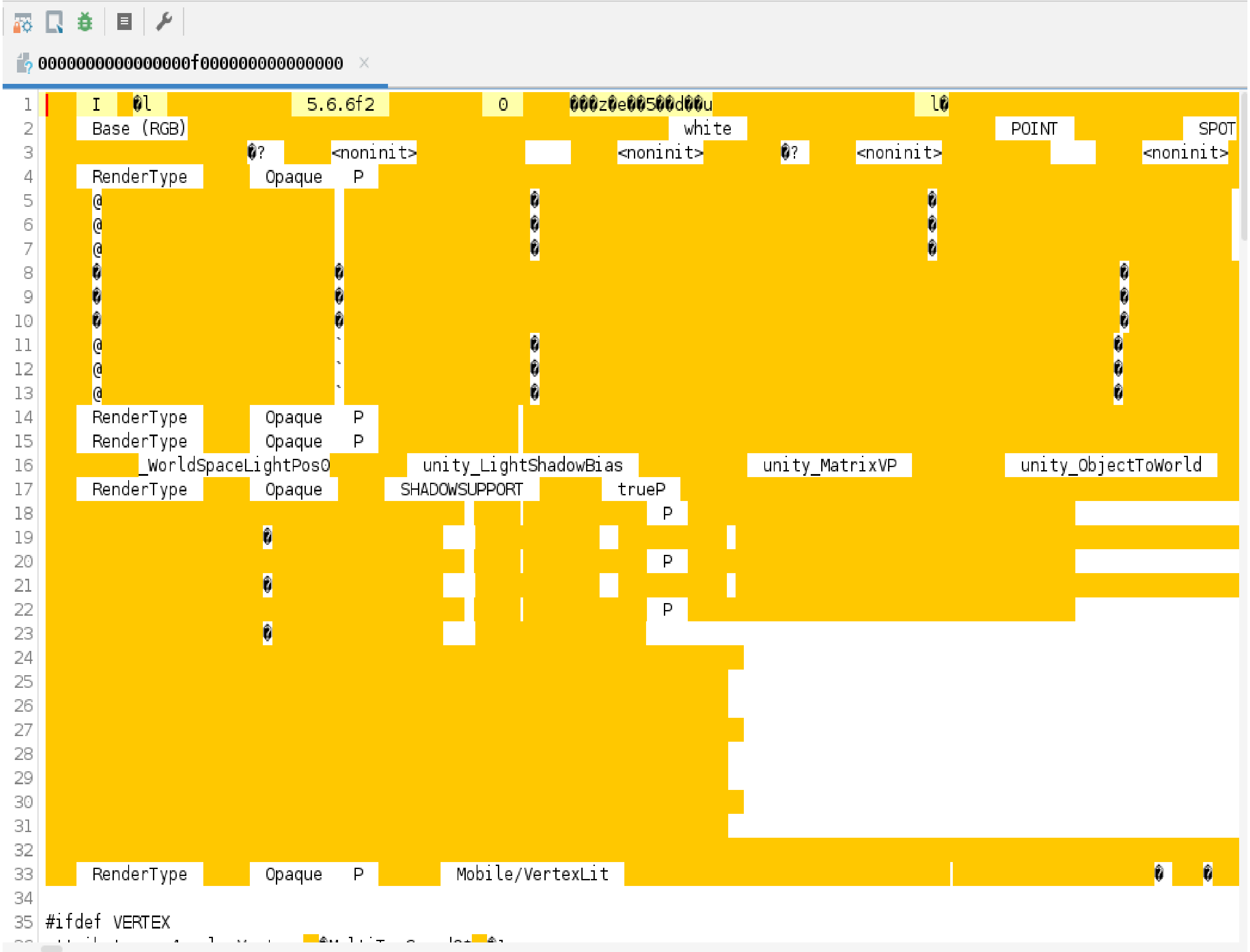
```
}

@Override // android.app.Activity
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.select_head_menu);
    intance = this;
    findViewById(R.id.albumBtn).setOnClickListener(this);
    findViewById(R.id.captureBtn).setOnClickListener(this);
    findViewById(R.id.cancleMenuBtn).setOnClickListener(this);
}

@Override // android.view.View.OnClickListener
public void onClick(View v) {
    if (v.getId() == 2131296405) {
        startAlbum();
    } else if (v.getId() == 2131296404) {
        startCapture();
    } else if (v.getId() == 2131296406) {
        finish();
    }
}

@Override // android.app.Activity
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    Log.i("wwl", "resultCode=" + resultCode);
    if (resultCode != -1) {
        finish();
        return;
    }
    switch (requestCode) {
        case 1:
            startCropImageActivity(handleImageOnKitKat(data));
            return;
```

We can see that there is uid is given in the code by which we can collect some information.



```
package com.cf.msc.sdk;

/* loaded from: classes.dex */
public class SecurityConnection {
    private String serverIp;
    private Integer serverPort;

    public SecurityConnection(String serverIp, Integer serverPort) {
        this.serverIp = serverIp;
        this.serverPort = serverPort;
    }

    public String getServerIp() {
        return this.serverIp;
    }

    public Integer getServerPort() {
        return this.serverPort;
    }
}
```

We can clearly see that the The application is forwarding our port and ip to the attacker server.



```
0000000000000000f0000000000000000

 I  �l         5.6.6f2      0     ���z�e��5��d��u               l�
 Base (RGB)                                 white                  POINT     SPOT
              �?   <noninit>           <noninit>     �?     <noninit>          <noninit>
 RenderType   Opaque  P
 @                         �                          �
 @                         �                          �
 @                         �                          �
 �               �                                                      �
 �               �                                                      �
 �               �                                                      �
 @               `         �                                           �
 @               `         �                                           �
 @               `         �                                           �
 RenderType   Opaque  P
 RenderType   Opaque  P
    _WorldSpaceLightPos0    unity_LightShadowBias       unity_MatrixVP        unity_ObjectToWorld
 RenderType   Opaque     SHADOWSUPPORT    trueP
                                              P
             �
                                              P
             �
                                              P
             �

 RenderType   Opaque  P    Mobile/VertexLit                                            �  �

#ifdef VERTEX
```

As we can see that the code is Encrypted and we can only see some text only.

CONCLUSION

Hence , We can say that the application is a malicious application. This type of application should be banned and user also should not install application from unknown source. The attacker may attack and access to the user android through this app.

At last I want to say that This application is malicious and cause serious damage to phone and people life .