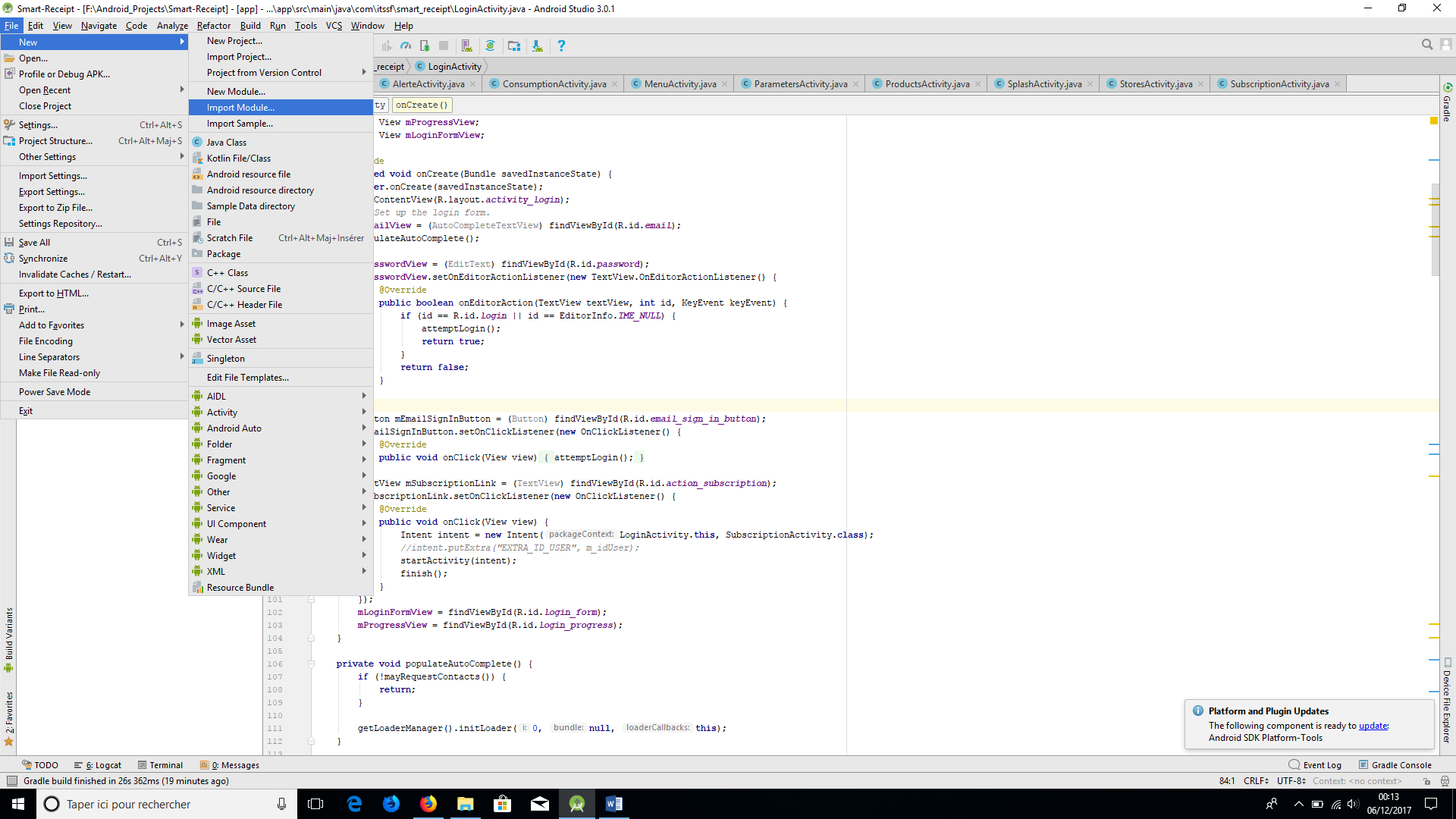
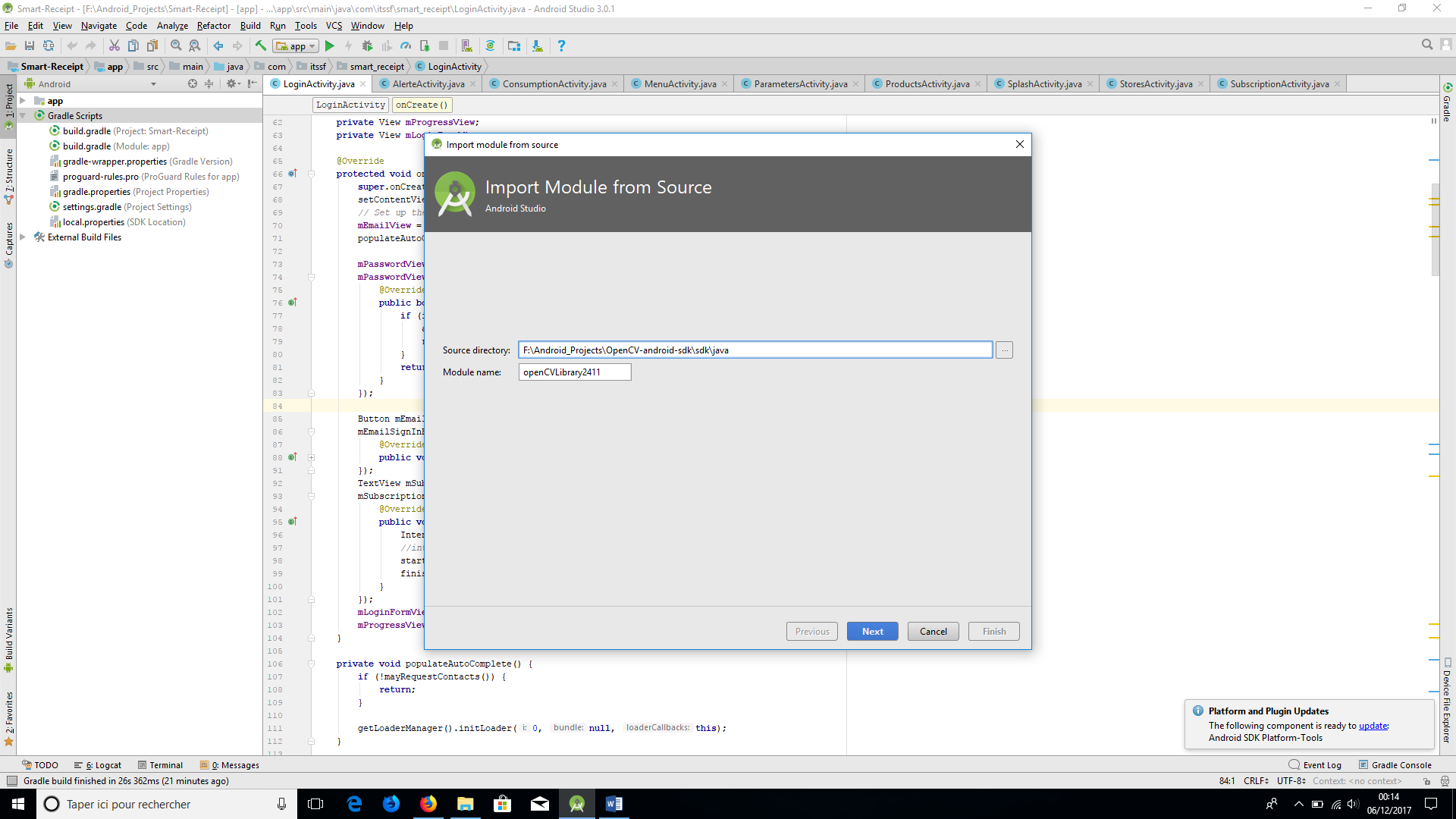
Télécharger OpenCV à l’adresse suivante : <http://sourceforge.net/projects/opencvlibrary/files/opencv-android/2.4.11/OpenCV-2.4.11-android-sdk.zip/download>

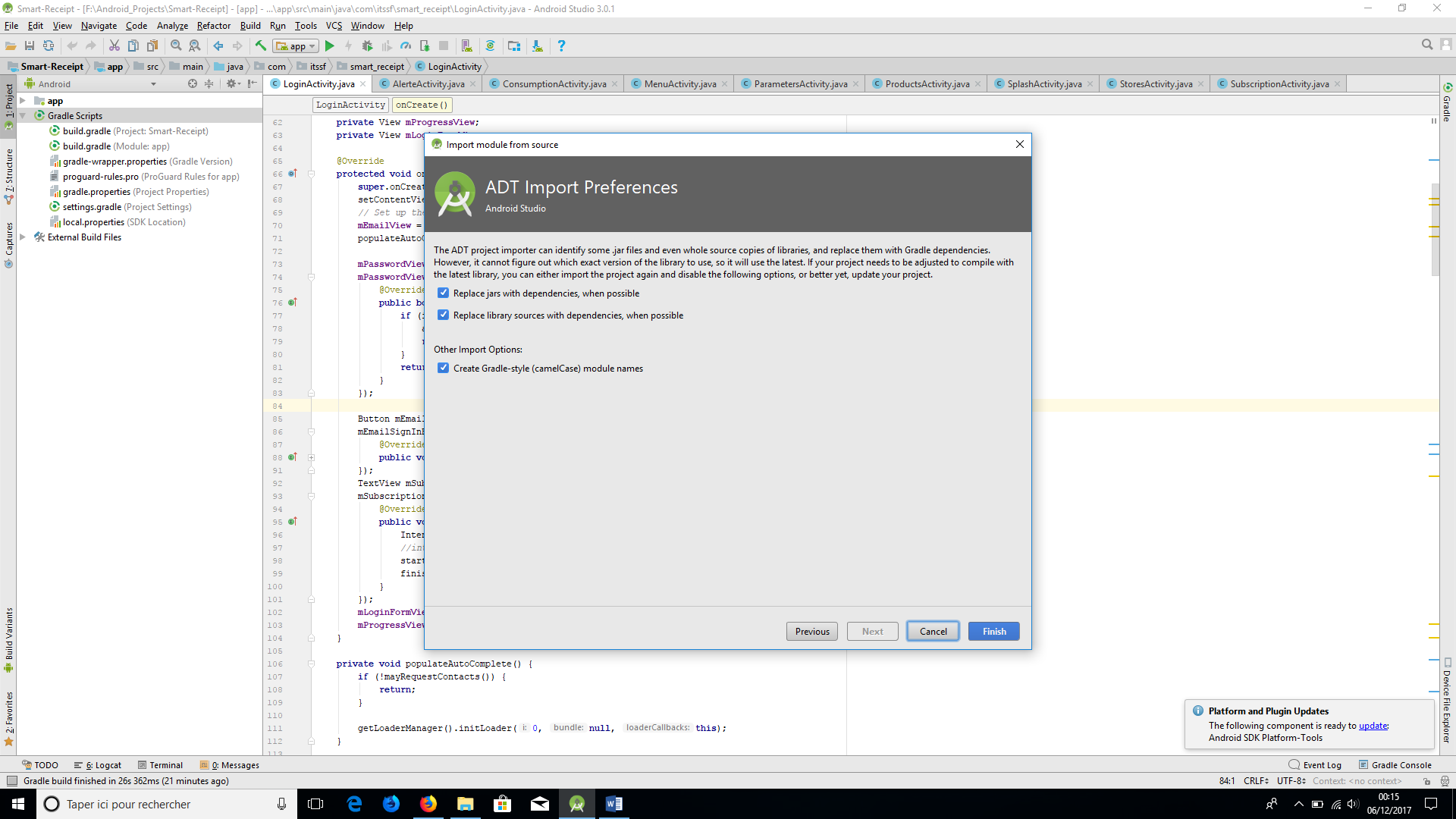
Extraire l’archive sur votre poste

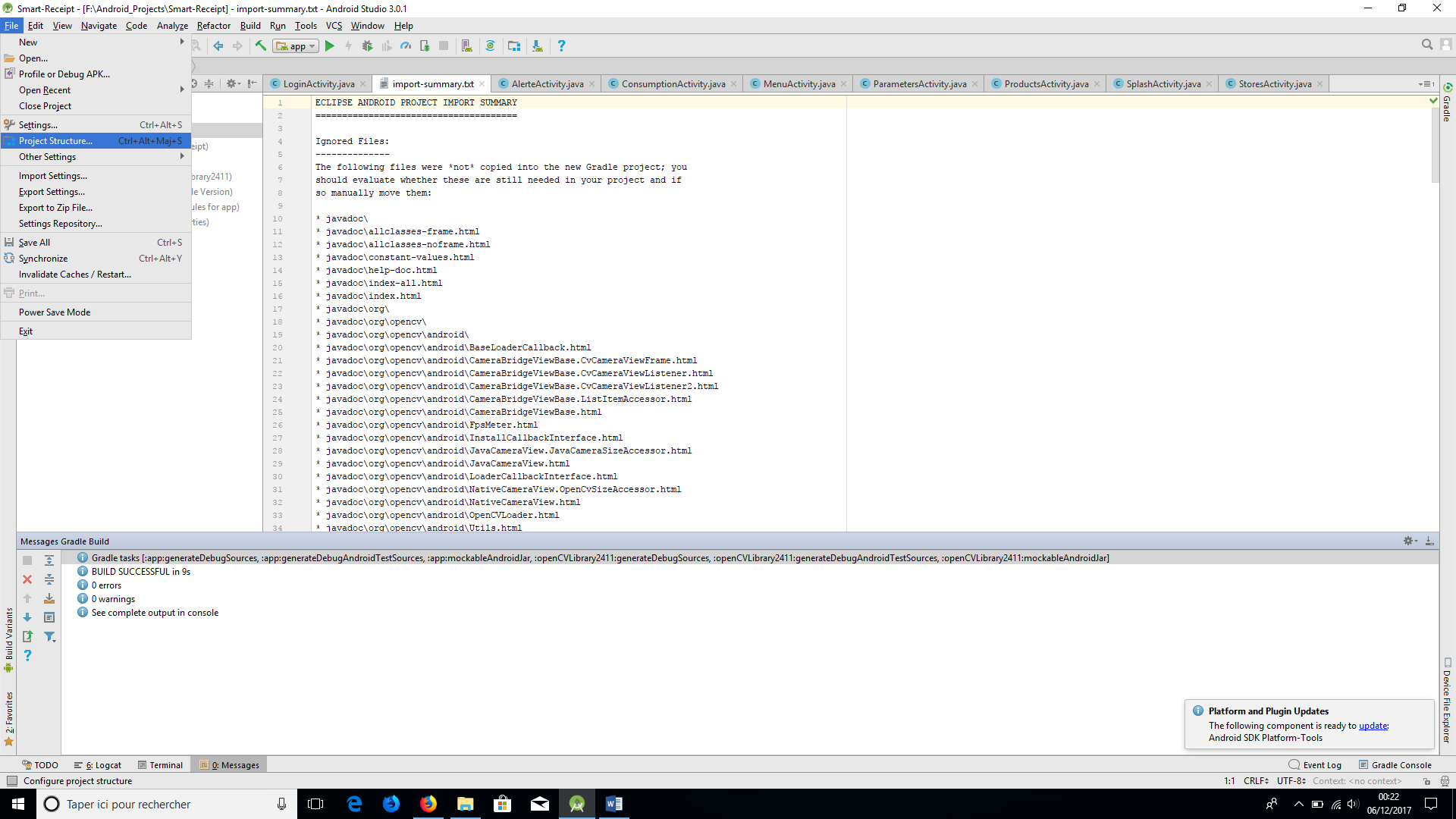
L’achive a cette structure :

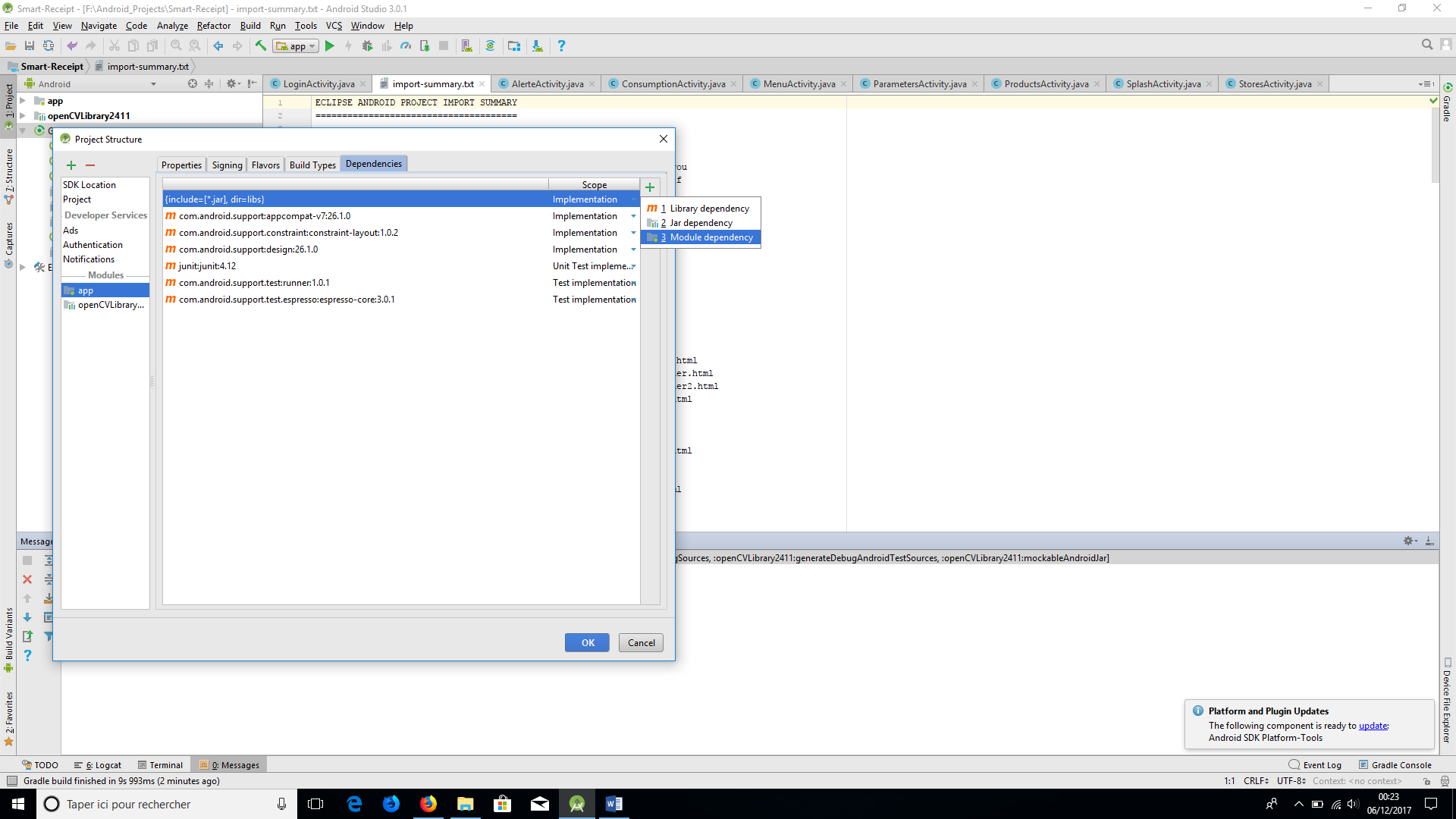
….

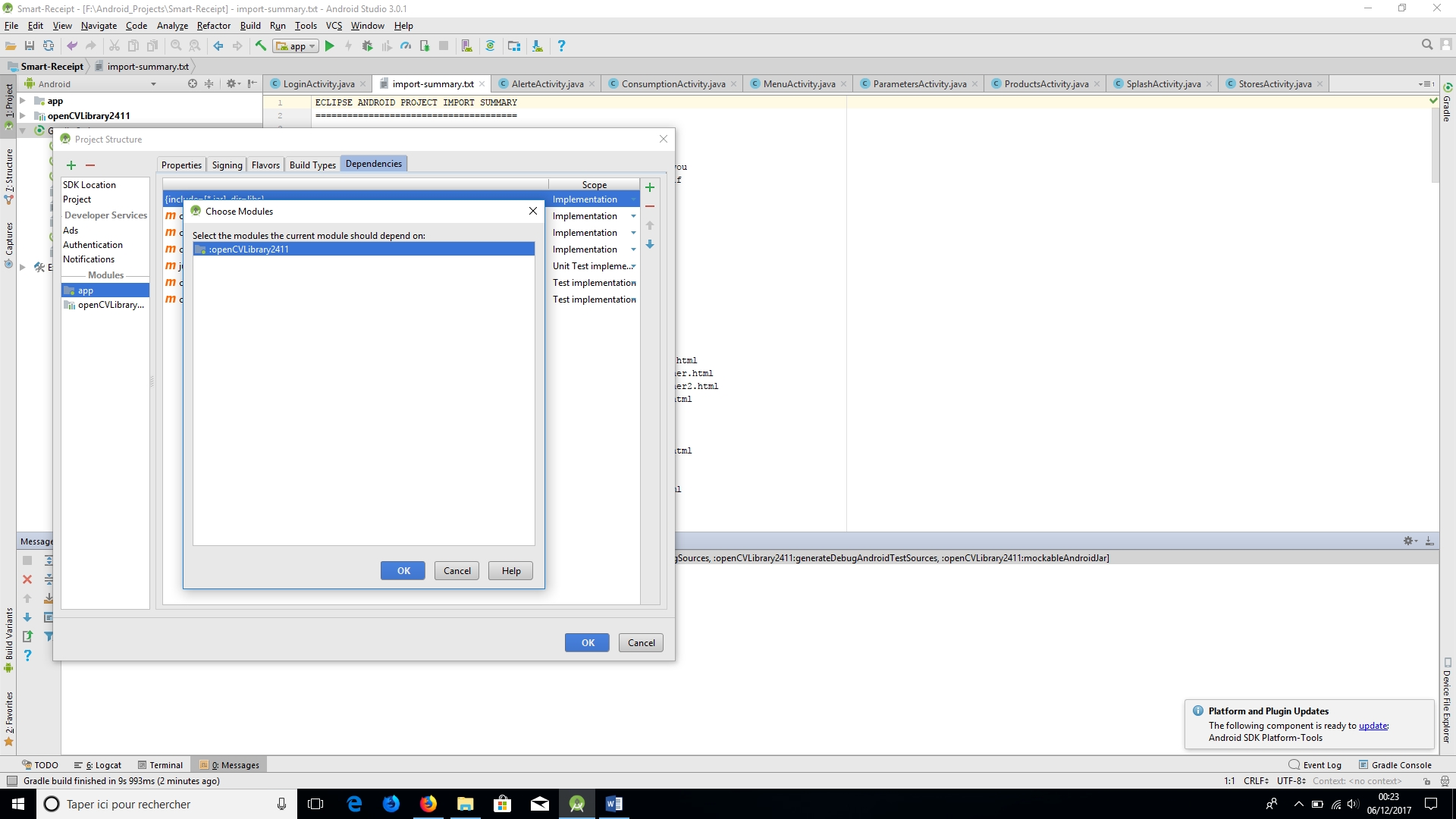


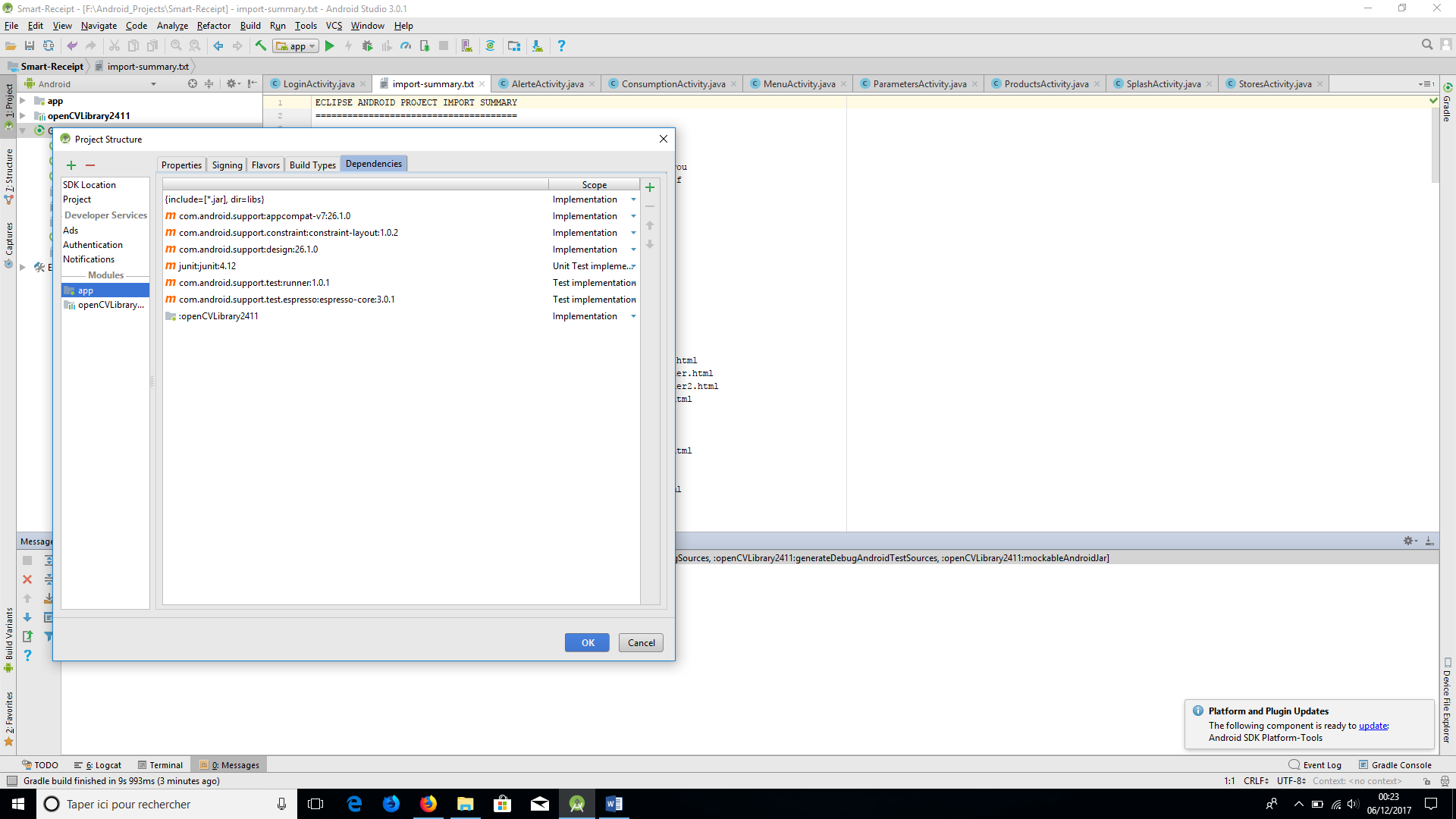












Créer le dossier jiniLibs dans le dossier src/main de votre application.

Suivre le dossier **OpenCV3-android-sdk/sdk/native/libs/**

Et créer les dossiers correspondants aux arches dans le dossier jniLibs précédant et copier les fichiers « libopencv\_java.so » dans les dossiers d’arch correspondant

Créer le dossier jni dans le dossier src/main

Dans ce dossier créer le fichier Android.mk

LOCAL\_PATH := $(call my-dir)

include $(CLEAR\_VARS)

OPENCV\_ROOT:=F:/Android\_Projects/OpenCV-android-sdk

OPENCV\_CAMERA\_MODULES:=on

OPENCV\_INSTALL\_MODULES:=on

OPENCV\_LIB\_TYPE:=SHARED

include ${OPENCV\_ROOT}/sdk/native/jni/OpenCV.mk

NDK\_MODULE\_PATH=F:/Users/miche/AppData/Local/Android/sdk/ndk-bundle

LOCAL\_ARM\_NEON := true

LOCAL\_SRC\_FILES := ScannerActivity.cpp

LOCAL\_CPPFLAGS := -std=gnu++0x

LOCAL\_CFLAGS += -O2

LOCAL\_LDLIBS += -llog -ldl

LOCAL\_MODULE := native

include $(BUILD\_SHARED\_LIBRARY)

Modifier OPENCV\_ROOT et NDK\_MODULE\_PATH avec vos valeurs

Créer le fichier ScannerActivity.cpp dans le dossier jni

Créer le Application.mk dans le dossier jni

APP\_STL := gnustl\_static

APP\_CPPFLAGS := -frtti -fexceptions

NDK\_TOOLCHAIN\_VERSION=4.9

APP\_ABI := armeabi armeabi-v7a x86 mips

APP\_PLATFORM := android-16

APP\_OPTIM := release

Modifier build.gradle

apply plugin: 'com.android.application'

android {

compileSdkVersion 26

defaultConfig {

applicationId "com.itssf.smart\_receipt"

minSdkVersion 20

targetSdkVersion 26

versionCode 1

versionName "1.0"

testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

externalNativeBuild {

cmake {

cppFlags ""

}

}

}

sourceSets.main.jni.srcDirs = []

task ndkBuild(type: Exec, description: 'Compile JNI source via NDK') {

def ndkDir = "/home/wenfahu/NVPACK/android-sdk-linux/ndk-bundle"

commandLine "$ndkDir/ndk-build",

'NDK\_PROJECT\_PATH=build/intermediates/ndk',

'NDK\_LIBS\_OUT=src/main/jniLibs',

'APP\_BUILD\_SCRIPT=src/main/jni/Android.mk',

'NDK\_APPLICATION\_MK=src/main/jni/Application.mk'

}

tasks.withType(JavaCompile) {

compileTask -> compileTask.dependsOn ndkBuild

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

debug {

debuggable = true

jniDebuggable true

}

}

externalNativeBuild {

cmake {

path "CMakeLists.txt"

}

}

}

dependencies {

implementation fileTree(include: ['\*.jar'], dir: 'libs')

implementation 'com.android.support:appcompat-v7:26.1.0'

implementation 'com.android.support.constraint:constraint-layout:1.0.2'

implementation 'com.android.support:design:26.1.0'

testImplementation 'junit:junit:4.12'

androidTestImplementation 'com.android.support.test:runner:1.0.1'

androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'

implementation project(':openCVLibrary2411')

}

android.defaultConfig.vectorDrawables.useSupportLibrary = true

Ajouter la nouvelle task dans build.gradle

task ndkBuild(type: Exec, description: 'Compile JNI source via NDK') {

def ndkDir = "/home/wenfahu/NVPACK/android-sdk-linux/ndk-bundle"

commandLine "$ndkDir/ndk-build",

'NDK\_PROJECT\_PATH=build/intermediates/ndk',

'NDK\_LIBS\_OUT=src/main/jniLibs',

'APP\_BUILD\_SCRIPT=src/main/jni/Android.mk',

'NDK\_APPLICATION\_MK=src/main/jni/Application.mk'

}

