Read Me

Enclosed is the requested artifacts for the C196 Performance Assessment. The application was built using Android Studio 3.21 targeting a minimum Android SDK 24.

The application was built using a 7.1.1 Emulator and a Pie Physical device.

The application makes use of the *Android Architecture Components* Room, ViewModel and LiveData.

https://developer.android.com/topic/libraries/architecture/

Due solely to a need for Lambda expressions, this application was built using Java 8 language features.

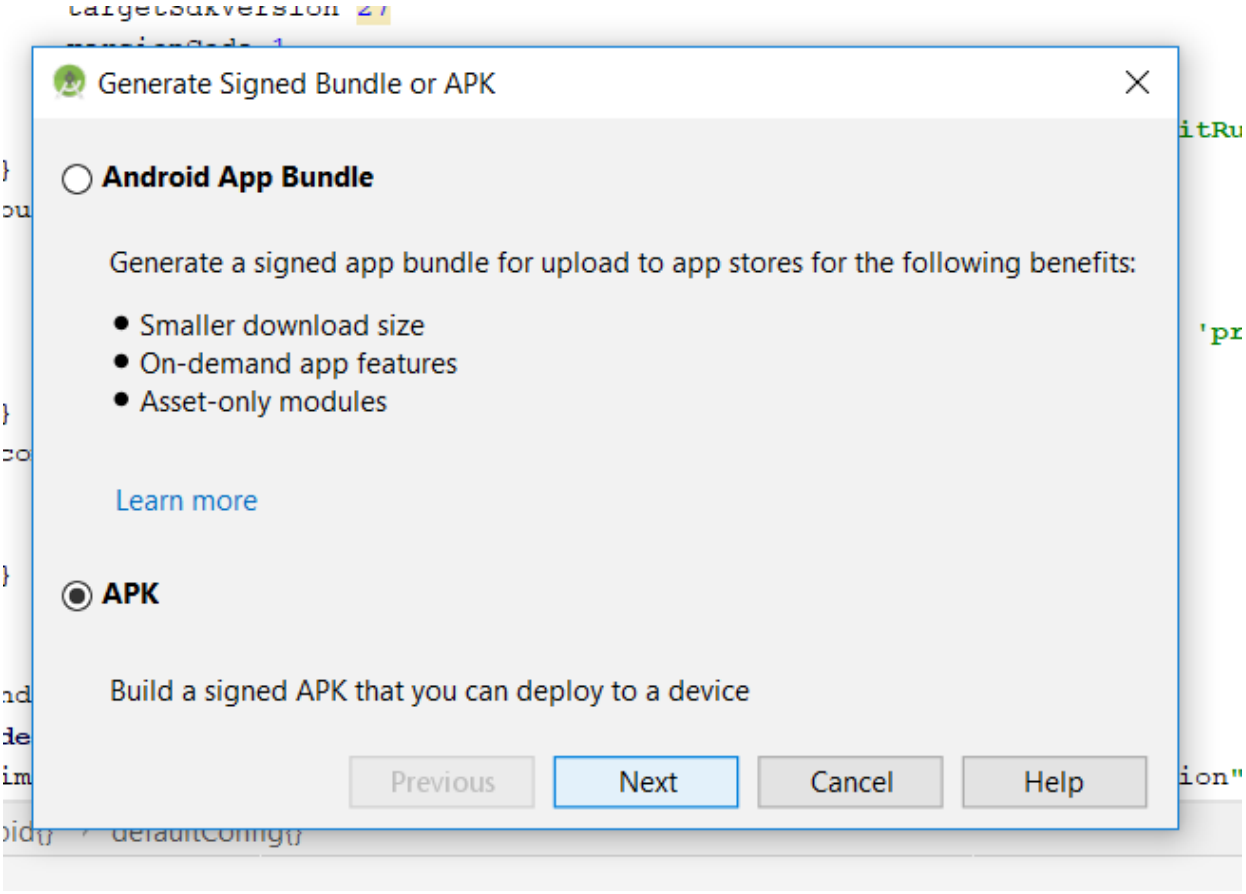
Appropriate help can be found while running the application by clicking the  icon in the toolbar.

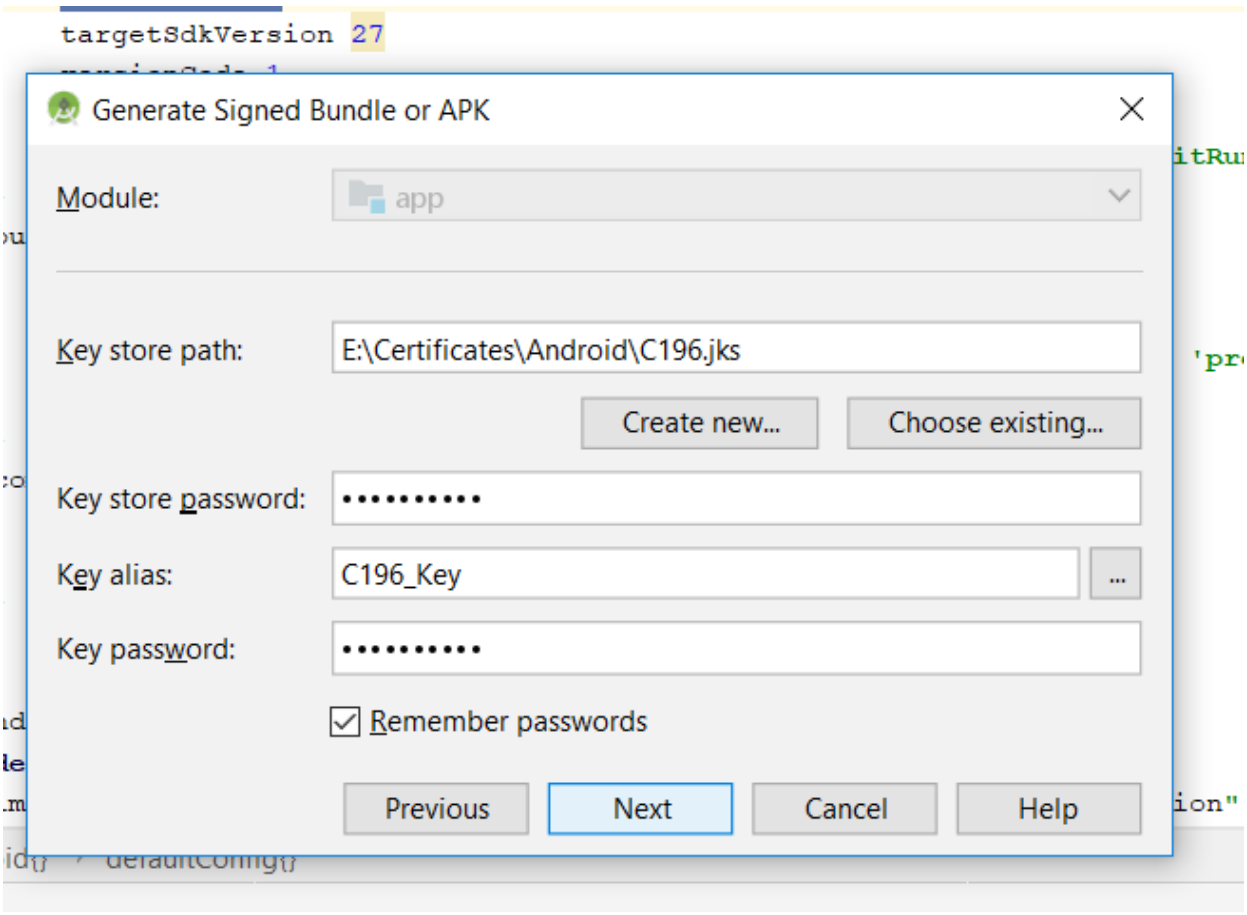
Most all of the UI was created declaratively using resource files, however the menus are created programmatically in the class BaseActivity using the onPrepareOptionsMenu override.

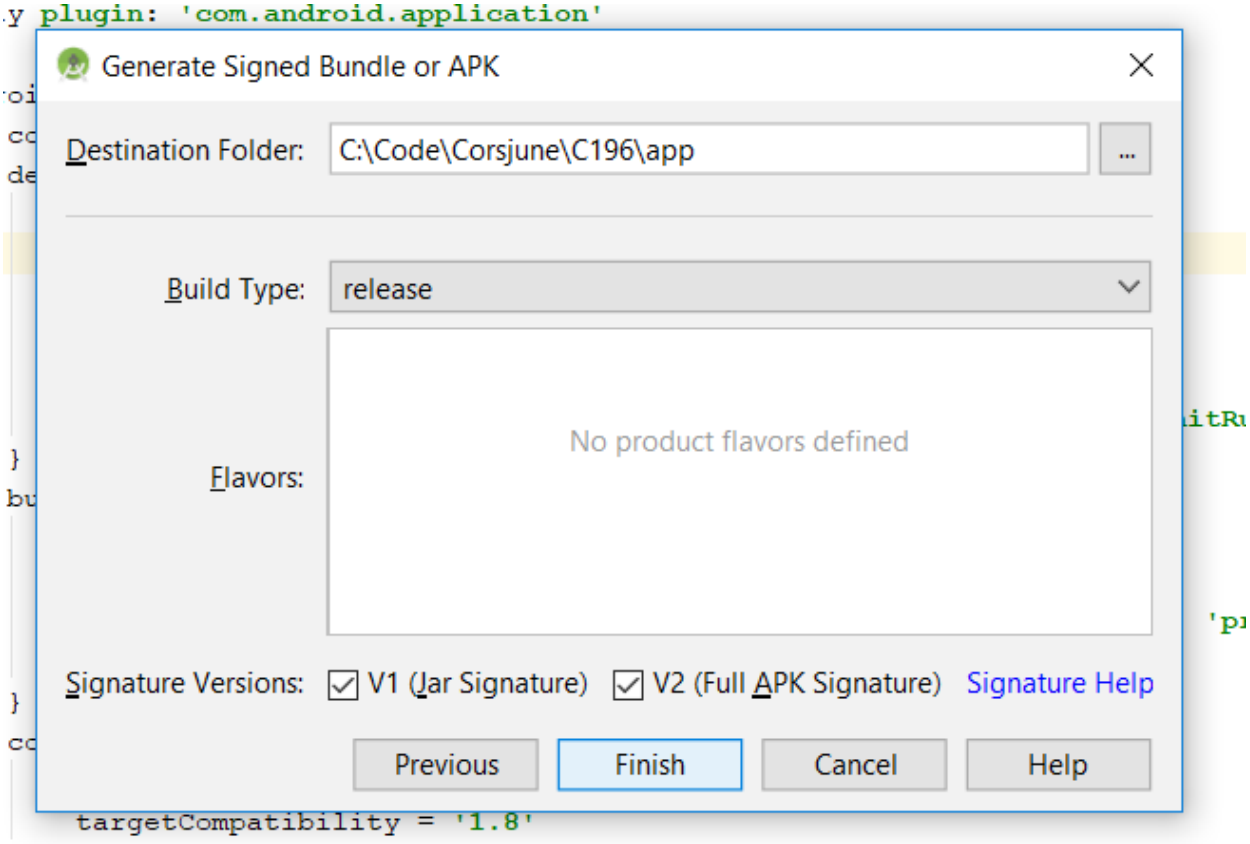
Notes can be shared on the Course Edit screen provided that notes have been entered. This can be done by clicking the Shared icon in the toolbar.

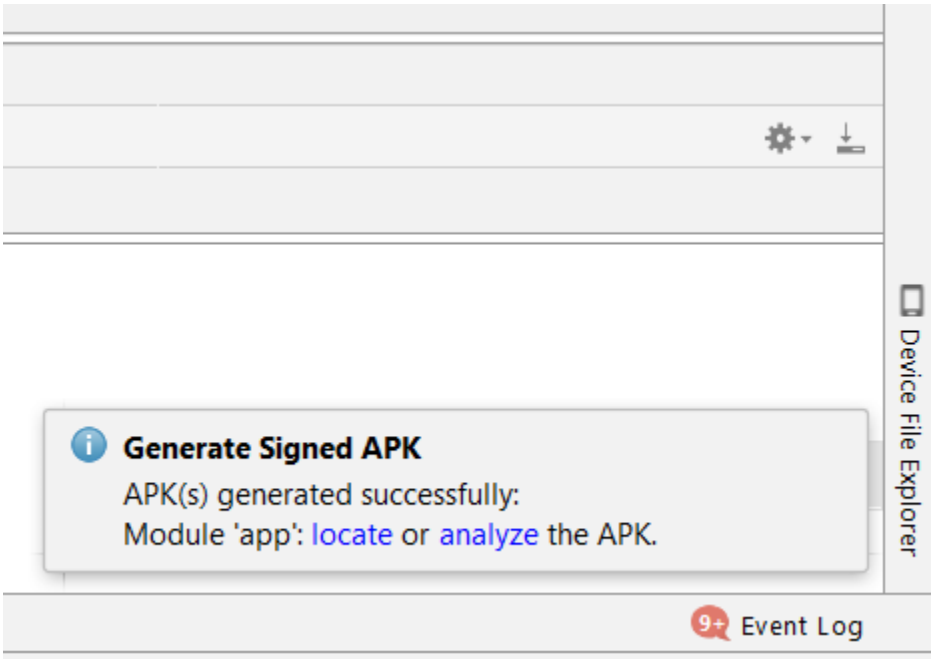
Signed APK

Below is a screenshot from signing the APK.


Generate Signed Bundle or APK
Android App Bundle
Generate a signed app bundle for upload to app stores for the following benefits:
Smaller download size
On-demand app features
Asset-only modules
Learn more
APK
Build a signed APK that you can deploy to a device
Previous
Next
Cancel
Help


targetSdkVersion 27
Generate Signed Bundle or APK
Module:
app
Key store path:
E:\Certificates\Android\C196.jks
Create new...
Choose existing...
Key store password:
Key alias:
C196_Key
Key password:
Remember passwords
Previous
Next
Cancel
Help


plugin: 'com.android.application'
Generate Signed Bundle or APK
Destination Folder:
C:\Code\Corsjune\C196\app
...
Build Type:
release
Flavors:
No product flavors defined
Signature Versions:
V1 (Jar Signature)
V2 (Full APK Signature)
Signature Help
Previous
Finish
Cancel
Help
targetCompatibility = '1.8'


Generate Signed APK
APK(s) generated successfully:
Module 'app': locate or analyze the APK.
Device File Explorer
Event Log

1. *Explain mobile application development through the context of the architecture involved, including hardware and software capabilities and limitations.*

Some of the difficulties encountered with mobile development is the following:

Development occurs on a machine with an entirely different processor set compared to the target. This requires emulation which runs slow.

Mobile targets are of varied hardware and viewport sizes. As such, you need to test against a variety of targets.

Mobile targets have a smaller memory and storage footprint requiring one to be cognizant of such. Also with this in mind, you will need to guard against the OS closing your application or application objects when memory is pressured or focus is given to a different application.

a. Identify the version of the operating system your application was developed under and is compatible with.

The application targets SDK 27 but can run under a minimum SDK of 24.

2. *Describe* **(suggested length of 1–2 paragraphs)** *the challenges you faced during the development of the mobile application.*

Due to rapid pace of development in the Android space, a lot of the development content found was stale.

It was difficult to determine what architectural patterns was best to work with on an Android App. (i.e. MVC, MVP, MVVM).

3. Describe **(suggested length of 1–2 paragraphs)** *how you overcame* **each** *challenge discussed in part F2.*

I researched various options and determined that MVVM as described in https://developer.android.com/topic/libraries/architecture/ was the best fit for this development.

I was able to find relevant discussions via Stack Overflow and Google Searchs.

Describe **(suggested length of 1–2 paragraphs)** *what you would do differently if you did the project again*

One of my biggest complaints of this development was the amount of boilerplate and plumbing code that was required to be written to complete this effort. I would explore third party libraries to see if any offered abstractions that could alleviate the plumbing code.

I would look into Kotlin in lieu of Java for a future effort. IMHO, Java required too much verbosity.

I have some code smells between the viewmodal and view. I would take another look there.

I would explore creating Generics to replace the 101 uses of RecycleView Adapters. The Adapters were a lot of cut and paste, Find and Replace.

5. Describe how emulators are used and the pros and cons of using an emulator vs. using a development device.

Emulators allow you to emulate a target device. During the development process, you can use emulators to target a variety of target devices which may have varied screen sizes, hardware platforms and features. This can be done without possessing said devices. However, two concerns that one may have with the usage of emulators is that emulators may suffer slow performance compared to an actual device and/also emulator may not perfectly reproduce the target's capabilities, software or hardware.