**F. C196**

1. Android development takes getting used to for the architecture. Having to deal with all sorts of screen sizes, orientations, and limited hardware resources. There are so many different versions of the Android OS. The various ways to architect a GUI is numerous. The development architecture for GUI I used were fragments and using a controller to navigate between the fragments. I used a single activity and that had an area for the fragments to be contained in. There are much more dependent things and a whole new process of gradles compared to other processes. Some hardware limitations for mobile development is the different screen aspects and small sizes. Having to strategically place things on the screen so that things look uniform and not clutter but be easily readable. I would say a software limitation is the operating system. The number of versions for everything are numerous and what could work in one, could break in another or not even be supported. I feel that limitations are the learning curve, it seems so complex for what is involved. All the dependencies, versions, manifests, and gradles. It’s a lot to take on and to figure out what does what.

1a. I developed under Android 5.1. I chose this because it will run on about 92.3% of devices. It is supported at minimum of Android 5.1 to maximum Android 10.

1. I struggled with this project. I didn’t understand how to build the GUI and where to start. It is completely different than IntelliJ with SceneBuilder. I don’t know if I got so used to that and struggled to adapt. I also struggled with the RecyclerView. I couldn’t figure out how to get my items to populate in the list.
2. I eventually figured out a way to build a GUI using fragments after constant research on android’s developer site and youtube. This allowed me to figure out a starting point and what to do. The more and more I did, the more adapted I became to android studio. I got RecyclerView to finally populate my items after looking at several recourses. I found an article on google that really helped me figure out what I was doing wrong. Apparently, I was populating the items, but my layout for the items didn’t have a fit to contents constraint for height. This caused things to populate but not show up properly in the view.
3. Since I have knowledge of how Android Studio works, I would understand the templates better when you first create a project. It would make sense to me how those templates apply. If I did the project again, I would choose a different activity template, but still use the fragments. I finally got adjusted to the fragments and how to work them.
4. Emulators create a realistic looking device on the screen to test the apps and are used to troubleshoot. They provide a virtual real-world representation. It’s a little more difficult because the habit to scroll versus clicking on the screen to navigate takes some getting used to. Having a physical device would be easier to get a true experience. I would say that’s the downfall for the emulator. There isn’t a true experience. The downfall for a physical device though is the price. An emulator can you give every single device possible, but realistically it would be way too expensive to buy all those devices. Not including the inconvenience of having to hook them up.