Musings on Mobile App Development

WGU C196 – Mobile Application Development

By Wesley Choate

1. When designing for mobile applications, there are many factors to consider related to the hardware and software limitations but also advantages. In the case of this course, the software limitation is making the applications run on the Android operating system. And the decision of which version to develop for is a big decision that must be made up front because of the many things that are not backwards compatible. For this project, the process used for scheduling notifications in the future is very different depending on the version of Android used. From the hardware side, having a touch screen makes the user interaction different. In particular, instead of double clicking, you normally will use either a single click or long press. Additionally, the smaller screen requires attention related to the amount of information on the page, and the size of user interface components. Too much information can be overwhelming and hard to consume, and if the button and other user interaction components are too small, they can be difficult to use.

1a. For this project, I used a minimum Android version of 27, with a target of 29.

2. Some of the biggest problems that I encountered were learning the tooling to development for the Android platform. This was my first usage of Android Studio, so there was some general ramp up there, and some problems probably caused just by my inexperience. Additionally, I have not really done any kind of front-end mobile application development. Trying to understand the workflows, design pattern, and limitations caused some problems at times during the process. Additionally, when I tried to deploy to my Amazon Fire tablet at the end, I found out it was on a version of Android older than Oreo, so I could not test on a real device.

3. Trying to overcome my lack of experience in both mobile application development and usage of the Android Studio took quite a bit of reading and research. I spent a lot of time looking through the Google Material Design documentation, and even more time digging through the various Android development guides and tutorials that google offers. In addition to that, there was a lot of trial and error followed by refactoring once I would solve something. As for the lack of an actual device, due to this already being done on an extension, I do not have time to order a new device. Therefore, I was not able to overcome this problem, and only tested on emulators.

4. I would learn to use the actual visual screen design and workflow tools that are in Android Studio better. I did most of the layout work manually entering and editing the xml files, not using the graphical tools. I think if I had used those my development time would have been much less and more consistent. Also, I used the Android Studio template (which as far as I know was ok to use) to build the initial skeleton of the application. But that left me feeling a little reversed on the activity and fragment workflow. I would have rather had the lists (recyclerview) be the parent (activity) and the details by the child (fragment). But the template built the options from the drawer as fragments, and I was not comfortable enough at the start to tear all that down. So learned some lessons there for the future.

5. Emulators are great for testing since they can so easily be launched right from Android Studio, and do not require any kind of external device to be attached. After using one, it seems like they are not quite as smooth to interact with a real device. And certain user interactions like swiping and rotating the device are not exactly the same. The ability to test on different sizes and operating system levels is really useful and help for multiple device verification. I would have liked to test on a real device for this task, but I am an Apple user in real life, so I only had my old Fire tablet, and it was not able to upgrade to Android Oreo or higher.