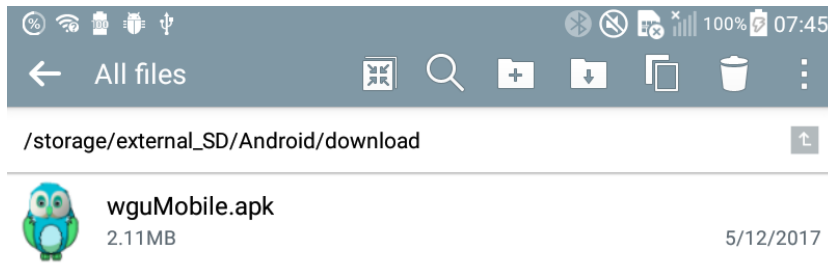


Android Project Reflection:

D) StoryBoard.

# E) Screenshot of APK



F.1) For mobile application development, there is an added layer of complexity in that your deployment environment is not your development environment. The limitations of this is that you have to either use a physical device while debugging, or an emulated device or a mix of both. I overcame this limitation by using a mix of 2 virtual devices, and 2 physical devices.

F.1.A) I developed this under Windows 10, using Android Studio. The base level of Android I aimed for was Android API 15 (ICS, AKA 4.0.1) and since I didn't have a physical ICS device that was one of my emulated devices. The mix of devices I used for this were, a virtual Nexus 7 running API 15, a physical LG 7 tablet on API 16, a virtual Nexus 6 running API 17 and my phone which is LG G5 running API 18. I hoped this mix would provide good coverage, but it can't possibly cover every circumstance. Every device is going to behave a little bit differently.

F.2 and 3) I ran into several challenges during the project, the first of which was time. I had under two months before the end of my term when I started this process back in early April. The way I overcame this obstacle, was to use the resources I had just learned in project management. I made a list of what I needed to accomplish, broke it down into easily digestible chunks, then tried to set timelines on what I needed to do next. I reviewed my progress every week, and used the rubric as a guideline to make sure I was staying on track for my time and what I needed to accomplish.

Another challenge I ran into was getting more than one ListView to function in my application. The example from David Gasner (Lynda.com resource video in course) uses a method that only allows one ListView in the application. It hard locked the thread to the table. After doing some research, I figured out how to fix that, using a different database call, and creating adapter classes for every ListView and Spinner.

Another monumental challenge was getting the camera to work, and to return the file and path that it saved. Across even my four devices, I found that the camera behaved differently and sometimes the bundled intent didn't include the path to the file that was just created. It worked fine on my virtual device, my Android 6.0 emulator, but when I tried it on 7.0 on my physical device it no longer worked.

I overcame all of these challenges in the same way. I first googled the problem I was running in to, usually found the exact problem on StackOverflow (see G.) and implemented the example solutions in to my code. If I couldn't find the solution easily on StackOverflow, I reach out to the Mentor. After detailing exactly what my problem was in the email to the mentor, the solution became easier for me to find online.

F.4) What would I do differently?

I would have simplified my CursorAdapter and Database class structure. It uses a List class to allow setting and getting property resources. This is really unnecessary and I could've just used the Database class definition as the List definition as well. Another thing I would like to have done differently is test the application across more physical devices. While researching my problems, it became apparent that the Android fragmentation problem is very real and code can behave very differently across different devices and versions of Android.

F.5) an emulator is software that emulates hardware, and software for that hardware. It's used when you don't have physical access to that hardware, but need to perform system testing on that hardware. When I was doing testing I found it far easier to use the emulators for testing, than using a physical device. I have two monitors, and setup the emulator on the second monitor. This allowed me to control inputs into the emulator, without switching my focus or having to pick up one of the physical devices I

have. The other pro in my case is that neither of my physical devices are rooted, or able to be rooted. When needing to debug the device to make sure things are being saved where I expect that is crucial.

The downside of using an emulator is that an emulator, no matter how good does not behave exactly like the actual device. Another problem I noticed is rendering bugs. When emulating on my virtual ICS device on a virtual 7" screen, the rendering did not look correct, even though it was correct.

G) APA references:

**Android EditText, soft keyboard show/hide event? (2011, June 12). Retrieved May 7, 2017, from <http://stackoverflow.com/questions/3793093/android-edittext-soft-keyboard-show-hide-event/6324350#6324350>  
hide on screen objects when the keyboard is visible**

**Felker, D. (2011, February 20). Android ACTION\_IMAGE\_CAPTURE Intent. Retrieved May 7, 2017, from <http://stackoverflow.com/questions/1910608/android-action-image-capture-intent/5054673#5054673>  
Camera storage and access**

**Gasner/Lynda, D. (2015, May 29). Building a Note-Taking App for Android. Retrieved April 1, 2017, from <https://www.lynda.com/course/sharerredirect/377485?org=wgu.edu>  
General Resource for Android Development**