Greg Snyder
c196 Reflection Paper

1. Mobile application development has its challenges, but it's these challenges which make mobile platforms so popular. Developing a mobile application takes specialized planning for it's small form factor. Screen size, RAM, storage capacity and battery life are all challenges needed to be overcome while developing a mobile application.

Mobile devices come in all shapes and sizes. From a wide range of phone/phablets to multiple tablet sizes. This means developers and designers have to account for each of these device sizes and make sure their applications work flawlessly on each. A developer has to be sure their application runs smoothly on high end devices and lower end as well. A designer has to take into account the rotation of the device and making sure all the content is clear and not cramped on the screen. Having a cluttered interface is one of the biggest errors you can make when designing your application. "Clutter is one of the worst enemies of good design. By cluttering your interface, you overload users with too much information: Every added button, image and icon makes the screen more complicated" (Esplin, 2017).

Developing apps for different platforms is also another hurdle to overcome when developing mobile applications. Each of these platforms requires its own code base and specialists to manage them. Not only are their different but then each phone manufacturer has their designs and specifications. This requires a very delicate coop which needs to constantly be tested and maintained.

Battery life for mobile devices have been a major issue from their inception. These devices can only fit so much within their frames. This has gotten better over the years especially with phones becoming larger, but this also means more powerful devices and larger, brighter screens. All of these design points have to be balanced to get the most out of the battery life of a mobile device.

1a. The application was developed for Android with a target API of 26

2. One challenge I faced from a hardware standpoint is the use of Android Studio and my PC. I actually created three different versions of the applications due to different issues before a finished application. I must have had a memory leak in the first application because I could not design the XML files in design mode. It would freeze my whole computer and I would have to restart the whole machine.

I had more trouble with the design aspect of developing the application than anything. Using the constraint layout was a bigger issue than it seemed like it should have been. I also had trouble with the relationship between mentors and the courses. I could not get the mentors to display on the courses screen. The courses under each term worked fine but I think it was because mentors and assessments were both associated with a course.

3. Due to being new to Android Studio, I had to recreate the project as I spent too many hours trying to find the memory leak. I also watched and read tutorials on designing the layout of the application. They helped, but I still felt the constraint layout cubersone. I ended up doing a lot of the layouts in linear format for a simply designed app. Due to lack of time before the semester ended, I ended up adding the mentors as a field under courses.

4. Overall, the project went well and development was smooth. There are a few things I would have done differently if I was doing the project over. I would have used butter knife from the start. The first and second project I started I was using the normal `findViewById`. It wasn't until my third attempt when I decided to use butterknife it helped with the development time. I would have used fragments instead of creating an activity for each view. This would have allowed for more flexibility when developing the application. Looking back on it, I would have created mentors as its own table. This was my original plan but I ran out of time before the semester.

5. Emulators are used as a way to test your application without connecting a physical device. There are pros and cons for both emulators and physical devices.

<u>Physical device</u>

Pros:

- You are able to get a feel for how the application will respond in hand as the finished project would. is. Easier to see how display reacts to touch, rotation, responsiveness.

Cons:

- Have to have a physical device handy and multiple if you want to test different screen sizes.
- Have to troubleshoot shoot connection issues.

<u>Emulator</u>

Pros:

- You are able to have multiple device sizes installed to test your application on a range of device types.

Cons:

- Lose some aspects of testing such as vibration and physically moving the device which can help while developing certain types of applications.
- Emulators are a but sluggish although this depends on your machine. It takes a lot of computing power to simulate the hardware and software of another device.
- Using a mouse for interaction on a mobile device does not replicate the real world feel of a physical device

References

Esplin, C. (2017, April 7). A Comprehensive Guide To Mobile App Design Retrieved from https://www.smashingmagazine.com/2018/02/comprehensive-guide-to-mobile-app-design/