Section F

## Explain how your application would be different if it were developed for a tablet rather than a phone, including a discussion of fragments and layouts.

* Developing an app for a tablet instead of a phone would be easier in some regards and more difficult in others. In my experience, there are less manufacturers for tablets than phones, or at least there are less options when shopping in person. This should limit the amount of devices an app should be compatible with. Many people also decide that they don’t need a tablet, whereas nearly everyone has a smart phone nowadays. For this specific application, many people would choose to download it on their phones so they can see the notifications while they are away from home, since they almost always have their phone. Most people don’t take their tablet with them everywhere.
* However, phones have a fairly standard screen size. Looking at Samsung’s Galaxy S22 line, the S22 screen is 149.9mm, the S22+ screen is 162.1 mm, and the S22 Ultra is 172.5 mm (S22, S22+, and S22 Ultra Comparison, 2023). As screen size is measured diagonally, 23 mm or so between the S22 and S22 ultra doesn’t make much of a difference in terms of app layout. Other phone manufacturers will make their screens to be about the same size. Google’s Pixel 7 has a screen size of 160 mm and the Pixel 7 Pro’s screen is 170.2 mm (Google Store's Phone Category, 2023). This makes it easier to develop an app because you can estimate the screen size of most of the market. My wife and I have had several tablets over the years. Some have been able to fit in the pockets of some of my pants, while others have been the same size or larger than a laptop screen. This can make the GUI look really cramped on smaller devices, or really empty on larger devices. A good application should look presentable on all screen sizes.

## Identify the minimum and target operating system your application was developed under and is compatible with.

* The application minimum OS is API 26, and the target OS is API 33

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

* I encountered a few issues during development, mostly with variable types. I like things to be stored in their data type, i.e., dates stored as a LocalDate. That way if someone enters a non-date, it should throw an error. Since the entities were straightforward, I had mass created them at the beginning. I also opted to use ENUMs for the assessment types and course status. When I finally got everything setup and the database portion created, I ran my code and it failed. It took me a while to figure out why it was failing, and then longer to fix related errors, especially with Intents.

### Describe (suggested length of 1–2 paragraphs) how you overcame each challenge discussed in part F3.

* SQLite doesn’t accept dates as their own data type, so I had to change the dates to Text/Strings (SQLite Datatypes, 2023). It adds some complexity when you are already partway through the project, but if I had known from the start it would have been easier. For the date, I opted to just use strings since it was easy to convert it into a date type if I needed to. Again, with the ENUMs, SQLite wanted them as Text/Strings. I could have just changed the ENUMs to regular strings and hard coded it in for the spinners, but I’ve found I learn more trying to make things work in my own ’best practice’ methodology. To get the intent to function properly, I had to use something like this: **(AssessmentType) (getIntent().getSerializableExtra("type")).** I’m not a big fan of the type casting, but I think it’s better in the long term to keep the ENUMs in rather than use strings.

### Discuss (suggested length of 1–2 paragraphs) what you would do differently if you did the project again.

* I tried to work on the project while I had down time at work. If I did it again, I would try to do it in longer sessions and do it on my home computer where I wouldn’t have as many distractions and an easier time navigating through everything because I’d have more screen space. I also would know about what I need so I don’t spend time changing data types mid project. Some of my code also follows the structure shown in the webinars. I don’t like some of the ways it was built, so now that I have a better idea of what to do I could make it more personal

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

* Emulators are nice because you can easily test an app on a wide variety of devices. If the OS becomes unstable for some reason, it’s trivial to reset. Emulators are also free. Emulators also allow quick switching between multiple API versions. One downside is emulators can take a lot of computer resources to run, so it is hard to test response times. Also, some features are unavailable, like Bluetooth. Others aren’t as effective, like location.
* Development devices let you use all the features of the device. You can also test responsiveness. However, they can be quite expensive. They also take up physical space, so there is a limit to how many you can have. Computers also have a limit on how many USB ports they have. While you can expand them, eventually there will be too many. It isn’t easy to control multiple devices at once either. If you only have one device, you can’t easily test the application on different APIs. You would need to flash the device each time you want to try a different API.

# References

*Google Store's Phone Category*. (2023, 01 24). Retrieved from Google Store: https://store.google.com/category/phones?hl=en-US

*S22, S22+, and S22 Ultra Comparison*. (2023, 01 24). Retrieved from Samsung: https://www.samsung.com/us/compare/#category/N0002101/products/SM-S901ULVEVZW,SM-S906ULBEXAA,SM-S908UDREXAA

*SQLite Datatypes*. (2023, 01 24). Retrieved from SQLite: https://www.sqlite.org/datatype3.html