Thomas Miller - C196 - Student ID=010243852

1. 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.

If this was designed on a tablet, I’d be redesigning the items within the RecyclerView fragments. Currently, because of space constraints, it only displays the Title/Name of the item, but a larger screen could allow for presenting the start/end dates and the Course Status/Assessment Type for the Course/Assessment list views respectively. The Add/Edit/View fragments could largely remain untouched, with the exception of the Assessments button, when editing a Course, as it currently overflows. The larger screen allows the layout itself to be larger, where the text would naturally not overflow.

I’d also consider revamping the Home view a bit, as it’s currently a text entry, and two RecyclerViews for course/assessment shortened lists. I’d instead have the two lists split the layout in half, left being for courses, right being for assessments tied to the selected course. (I’d still keep the last Upcoming Assessments list on the bottom, as those may not be tied to any course/term. These RecyclerView Item fragments could also include the start/end date, as well as small icons for the course/assessment enum’s.

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

This is compiling on API 44, but targeting Android 13 (API Level 33). The minimum is also 33.

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

I had two problems: one was updating the recyclerview whenever a new object was added to the datamanager (essentially the in-memory db). I primarily ran into this issue when adding an item, then deleting it, the item still existed within the recyclerview. The other issue I had was connecting the dialog to an item onclick within a recyclerview. I needed this to be able to find/open a model to pre-fill the edit dialog fields.

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

To have the recyclerview update itself on an action, I had to make a callback from the add/edit fragments back into the recyclerview, to update their views. To allow editing an object from a recyclerview, I had to make a interface on the top fragment to pass the position from the recyclerview, into the add/edit dialog (with edit mode being set to true).

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

Currently viewing a model is the same as Editing, but I’d update the dialogs to have separate states, having the view mode disable the forms editable state (and hide the delete/save buttons). Having them combined may be functional, but to a user, would be unexpected.

I’d also spend time theming the app, refining margins, text sizes, buttons, and lists, to not simply be the default Material Design defaults. The space between the header and top of content within the primary fragments, I’d also like to put something there to fill in the space, but I don’t know what.

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

I actually didn’t use an emulator at all throughout development, for there being extreme performance issues even booting the emulated device on my computers. If I were to set one up (outside of android studio, their emulator is currently broken for me on new versions of OS’s), I’d set up WayDroid.

Here are the pros and cons of using an emulator.

Cons:

* The ARM->X86/64 translation layer (usually via QEMU), can introduce extreme performance impacts.
* A normal android phone may have a lot of various apps/services running in the background, including manufacturer services. Those have potential to introduce bugs into code normally targeting AOSP, and using an emulator will not help catch these.

Pros:

* Testing repeatability. When the emulator is reset upon every test, bugs caught within an emulator can create accurate regression tests
* Accessibility, particularly on various screen sizes: It isn’t feasible to purchase multiple devices to cover various screen sizes (example: foldables, tall aspect ratio, and small phones). The emulator can emulate different screen sizes to simulate the experience of people with these phones/tablets.