1.

Android mobile application development is conducted with Android Studio which utilizes a pattern of Activities and a ViewModel architecture design. Android Studio uses an Android phone emulator that can run many different types of Android devices and Android versions.

There are many hardware limitations and capabilities to consider such as screen size, memory, and storage size. There are hardware limitations with the amount of memory and storage you have to work with so the app needs to be architected in such a way to get around this such as only querying the data you absolutely need to display.

The software capabilities are can also be limited compared to a desktop app. One example of this is using SQLite compared to a full sized SQL database. There are limitations with using SQLite such as limited data types and no user management.

1a.

For my project, I used a Pixel 2 device.

The minimum SDK is 17 and the target SDK is 29

2.

Development was a challenge for me since I wasn't familiar with any Android devices or development. I think the most challenging issues I faced were not understanding the different APIs being used and how certain examples online would be using incompatible APIs than mine which caused a lot of headaches. There was a period of time at the beginning that I wanted to give up due to trying to use the wrong API and having constant build failures.

I also had some challenges with getting to know the Android emulator and how to appropriately set it up for my project. I had issues with the time being different and not matching my system clock so setting the alarms was inconsistent.

It was challenging to implement allowing the user to enter a new mentor after the design of the app initially was to insert a list of mentors into the database on app load. The user would then select the mentor from a list. It was difficult to add this ability in a way that would integrate properly in my initial design and not cause errors.

3.

I was able to overcome a lot of the problems with the API version by using the exact API used in the plain ol' notes app tutorial. Once I followed the video and copied his setup verbatim, I was able to properly build my app.

I was able to correct the time issues on the Android emulator by remembering to explicitly set the clock to my current time. I had to do this every time the emulator restarted.

In the course edit screen, I added three text edit fields and a button to save a new mentor. The mentor would then be available for selection in the mentor select dropdown. Once selected, the mentor would be assigned to the course per the usual mentor select design.

4.

If I did this project differently, I would definitely not try to use a different API than the one used in plain ol' notes. Without knowing anything about Android development, it lead to too many unknown errors that I wasn't able to figure out myself.

I would also probably finish out the entire course edit section, before moving on to the assessment section so that I could copy and paste code instead of having to fix issues in both layouts at the end.

5.

Emulators are run on a host computer and used to mock a mobile device in order to run and debug your mobile application. Being a Mac/iPhone user, this allowed me to build the project. This method worked pretty well throughout.

Generally, emulators will run slower than using a development device. The pros of using a development device is that you are running it on dedicated hardware that will allow your app to run smoother and faster. The cons of running a development device is that you need to own this device which not everyone can afford. The main con when running via an emulator is that your host computer's hardware is being used to run the emulator which can lead to the emulator not having the processing power required to run your app smoothly. The advantages of an emulator is that you can mock the device and run your application regardless of whether you own the device or not.