Michael Irick  
C196

The development of the mobile application for C196 was filled with challenges. The Android version targeted is 8.0, but the application should be compatible with those versions higher. This placed a few limitations on development. I do not have a compatible device to test with(my phone is 5.0 and functions perfectly as a phone, not much else) and so needed to rely heavily on the emulator. Using the emulator prevents me from reaping the benefits of an actual device, such as running on real hardware with resource limitations found in the wild, experiencing the application as a user would, and finding performance issues(battery life, storage access, etc); on the other hand, it is vastly less expensive to use an emulator for both acquisition and maintenance costs for the vast range of possible Android devices. The Android version also prevented me from using some of the newer features of Java, lambda expressions in particular. The minimum SDK version is 26 and the target SDK version is 28.

However, neither of those problems compared to the utter pain that the Java language is to develop in. The Java Virtual Machine is a master work of architecture and computer engineering but it is quite wasted by assumptions and paradigms imposed by the programming language. Given the choice, I would have preferred to write the application in Kotlin which is compiled into the same bytecode and would work perfectly. Unfortunately, there is no other solution to this problem other than wrestle with the beast and press on. I attempted to factor out as much of the verbose boilerplate code into a class hierarchy utilizing generics and inheritance where possible, and while my attempts to produce a stable, maintainable frame on which to build were largely successful, if this were a professional project, there would be no hope for profit.

One interesting challenge I faced was the sheer amount of information that got put into some of the views. The course view, for example, had the course information plus a section for the assessments, the alerts, and the notes. To resolve this, I implemented a series of collapsable panels that display the panel title and a toggle button to expand the view.

If I were to do this project again, I would choose to build a webview and wrap a web application using any decent framework; Ruby-on-Rails, Django, or even Wordpress would provide a better, faster, more complete solution to the business problem the project seeks to solve than a native application.