Zachary Timmerman

5/1/2017

Project 3: Map O Shame

Map O Shame is an app primarily designed for one purpose: to get butts in seats in lecture halls. Its methodology is simple: no one likes being found out. We don’t like being ashamed, and we’ll do things we’d rather not to avoid shame. This idea of shame as a motivator has been used in other apps before. BetterMe, an alarm app for iPhone with Facebook integration[[1]](#footnote-1), posts how you didn’t feel like getting out of bed every time you hit snooze. Other software on the same concept are Morning Routine, which does not shut off its alarm until you scan the barcode of household items[[2]](#footnote-2), and Gym-Pact, an app that attaches to your bank account and removes money when you fail to show up to the gym[[3]](#footnote-3). The key behind all three is the concept of shame and consequence. For many, it’s difficult to forget the long-term impact of short-term actions, and these apps can provide negative reinforcement to give that little push. The app is primarily designed around college students, who are getting used to the idea of being wholly responsible for themselves, with no more Mom and Dad to tell them what they must do. While we can skip out on class, it usually ends up biting us in the long term. While the apps are good for their specific categories, this app is designed to tailor more to the needs of such students.

Features implemented include a half-baked attempt at an AlarmManager, a NotificationManager, and a broadcast receiver to start the notification manager from the prompting of the alarm manager. The idea behind the system was that the alarm manager would be set for a given class when the course was added. The alarm manager would then set off a broadcast receiver, which would initialize a notification with the important details of the class’s name and location into an intent, which would then, when the notification was clicked, launch the Twitter API to post how hard you failed to make it to class to Twitter. That was the idea, as Rube Goldbergian as it was. The API used was the Fabric API’s TwitterKit, which links to Twitter’s API and provides a simple interface. Ultimately, though, there were problems.

Firstly, development was non-existent on the app. I pushed aside the project due to poor planning of projects for other classes, so this project wasn’t given much mind while I tried to head off the other fires that were supposed to be decent projects. Then even more got in the way, and suddenly all that free time I’d planned to work on the project was gone, giving only a day or so. Compounding this, I’d failed to estimate just how incomplete the project 1 application was in the first place. Only somewhat workable and labyrinthine in its construction, it wasn’t easy to follow just what actually did what. To top it off, there is still a persistent bug that I have not been able to nail down. Adding a course will, without fail, crash the emulator’s runtime of the app. However, the course is still added to the ContentProvider. At first, I thought it was due to the activity attempting to launch another activity, but that was not the case. The TwitterKit API was also ill-suited for what I wanted it to do, since the interface was designed under the understanding that the user would want to be prompted for every Twitter post the app could make. Not the best solution to getting a user shamed. As time wore down, I planned to scale things down and simply push a notification based on the alarm manager that would ask if the student really was in class or not, and would call them a liar if they said that they were actually in class. Simpler, since I could not figure out how the location system in the original app worked in the slightest.

The timing system already present in the app was much weirder than I had expected, so “linking up” the app’s data to a new alarm manager proved far more difficult than I thought it would be, leading to its failure. Ultimately, this app is simply a case of getting in way over my head and thinking that I could whip up a quick solution in a couple nights. It’s that Android development simply doesn’t work that way, as every module needs to be carefully looked at to figure out what exactly it does, anyway. While very powerful, the endless maze of dependencies and black-box modules gets very confusing very quickly.

1. <https://itunes.apple.com/us/app/betterme/id593717331?mt=8> [↑](#footnote-ref-1)
2. <https://play.google.com/store/apps/details?id=net.havchr.mr2> [↑](#footnote-ref-2)
3. <http://www.gym-pact.com/> [↑](#footnote-ref-3)