P7

Ian Davis

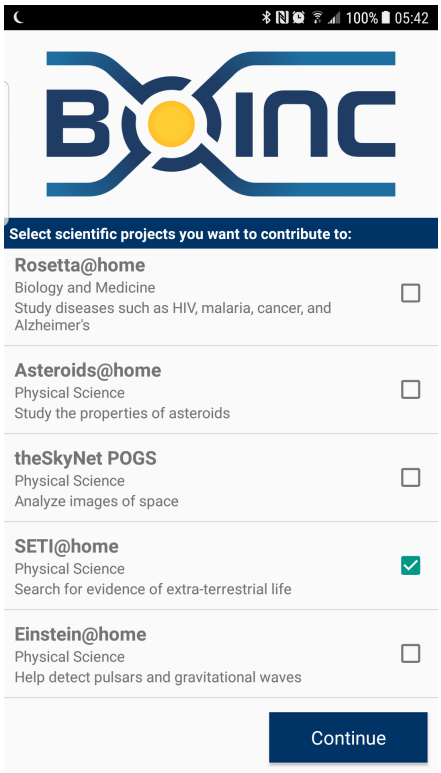
U00723728

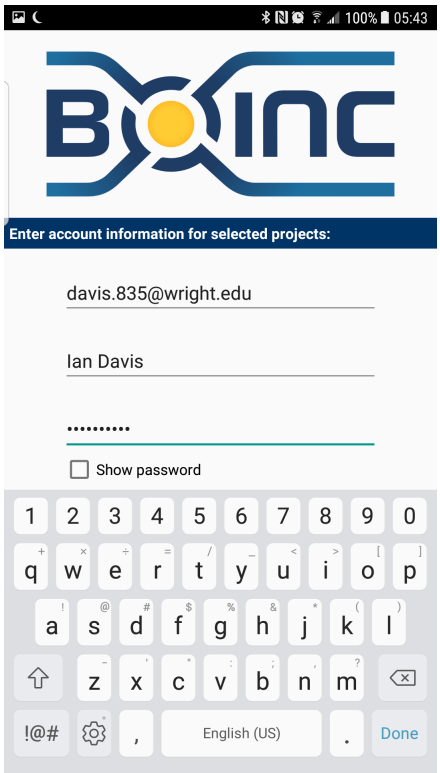
Github link: https://github.com/IanDavis1995/P7

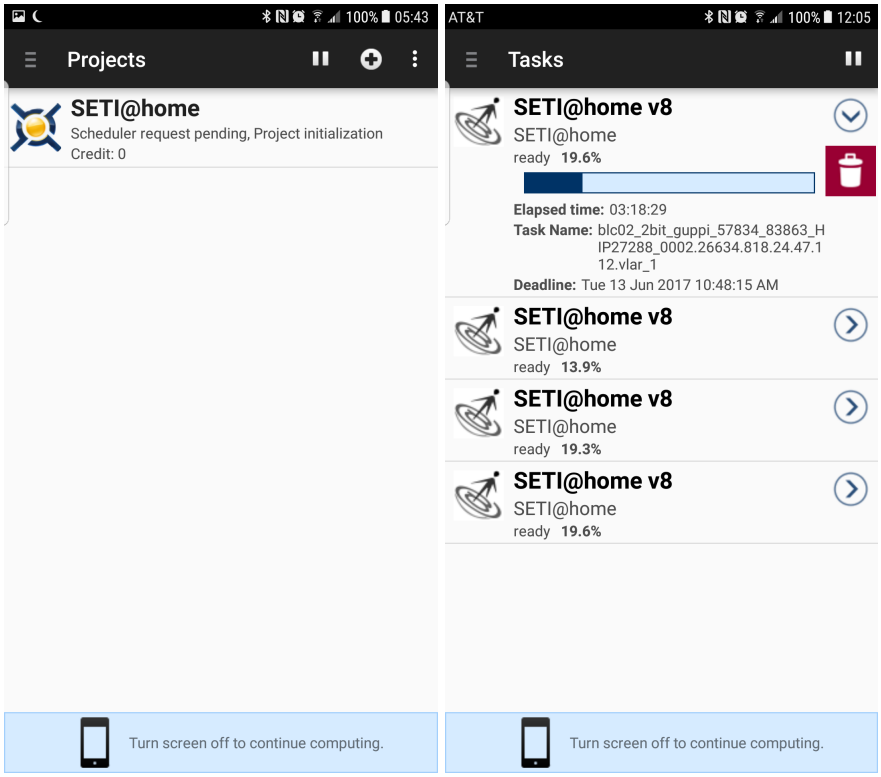
Task 1

- Screenshots pg. 1-2
- Status Report pg. 2
- Experience Report pg. 3

Screenshots





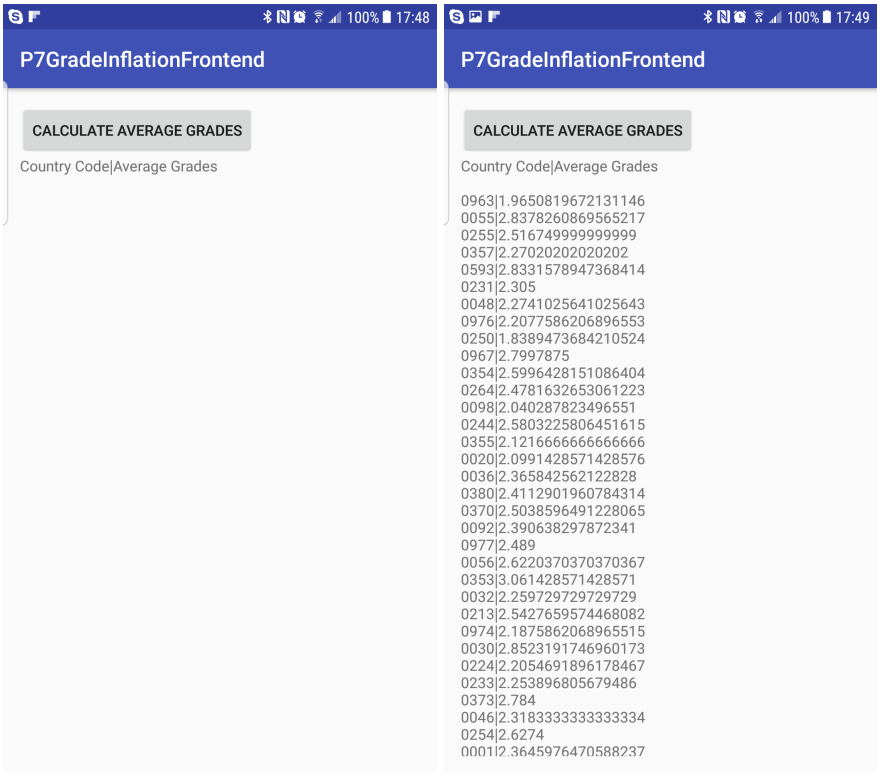


Status Report

The BOINC app was incredibly easy to install and setup, and SETI@home was up and running in seconds. The user interface for the project is a little lackluster, it shows you what tasks are running, a percentage complete (?) and how long they've been running, but that's about it. It would be nice to have more feedback on statistics such as CPU Cycles that I've contributed, and maybe some more information on what it's doing with those CPU Cycles as well. Overall, for an end user, this seems like a great tool to put your phone to use when you're not using it, but it would be nice to get some additional feedback.

Task 2

Screenshots

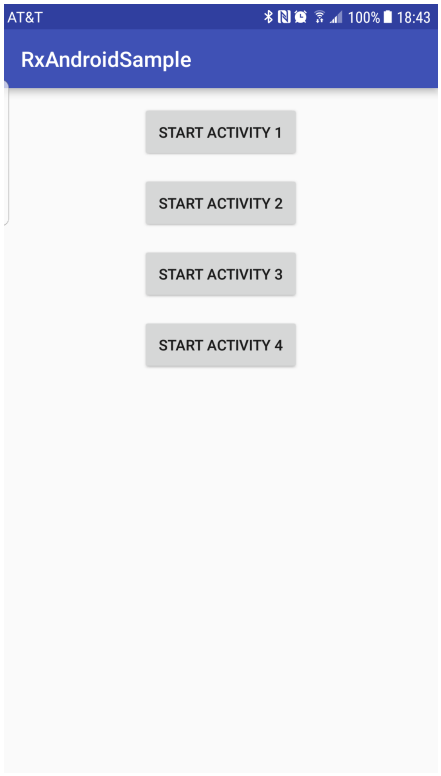


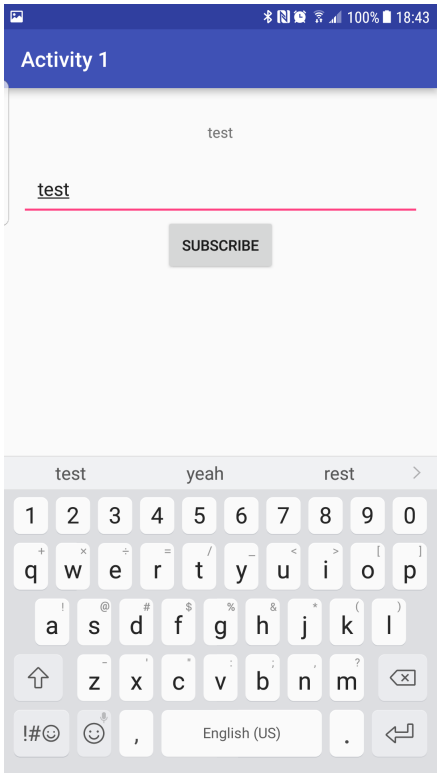
Status Report

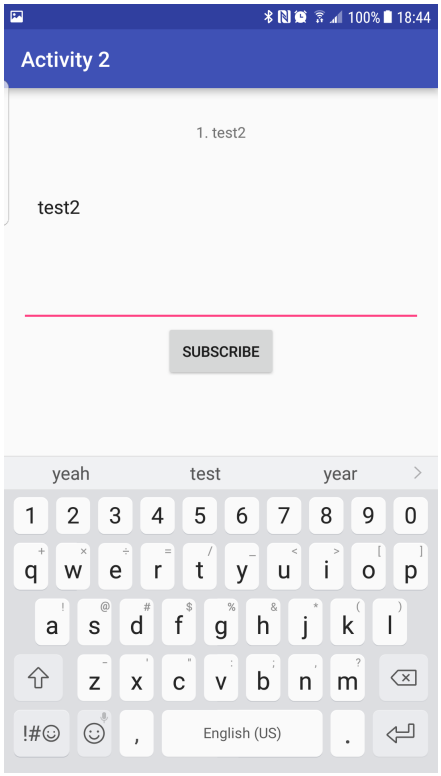
This application takes a decent amount of time to calculate the averages for all countries, but there are no issues. I Implemented a socket server in an Amazon EC2 instance that reduces a stream of all the files in the Grades directory and calculates averages of all students' grades. It then writes a list of all the country codes mapped to the average grades for that country back on the socket that sent the request. This list is then displayed on the android app.

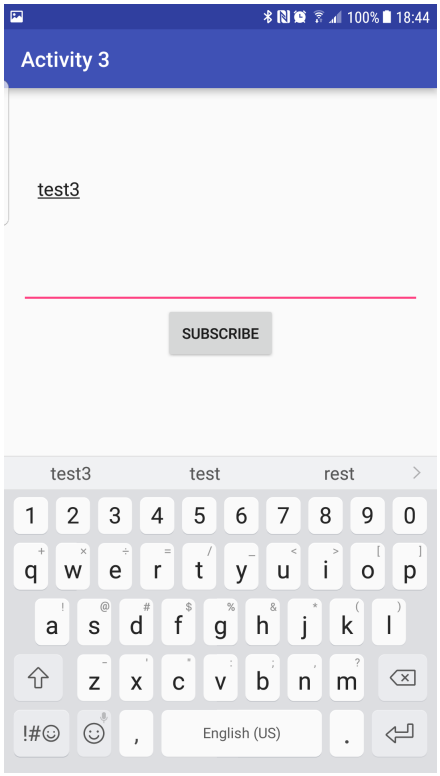
Task 3

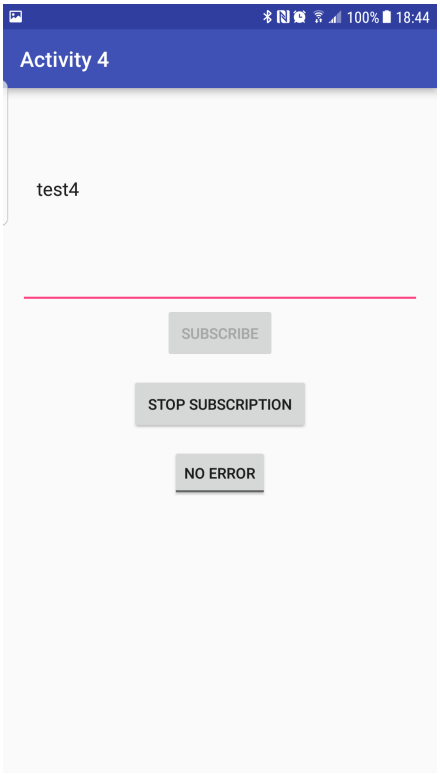
Screenshots

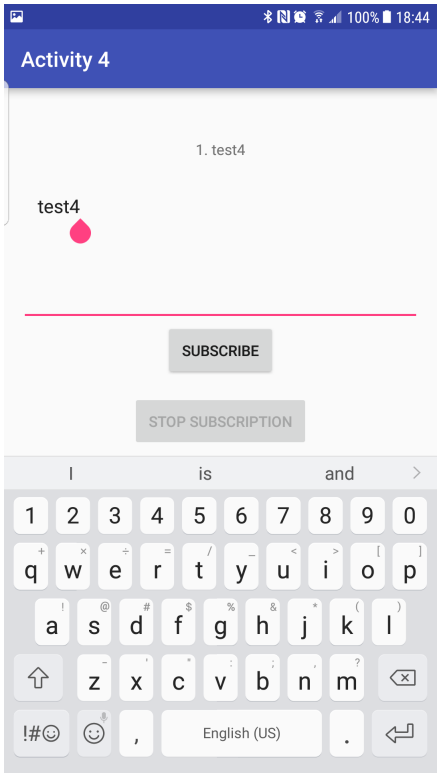










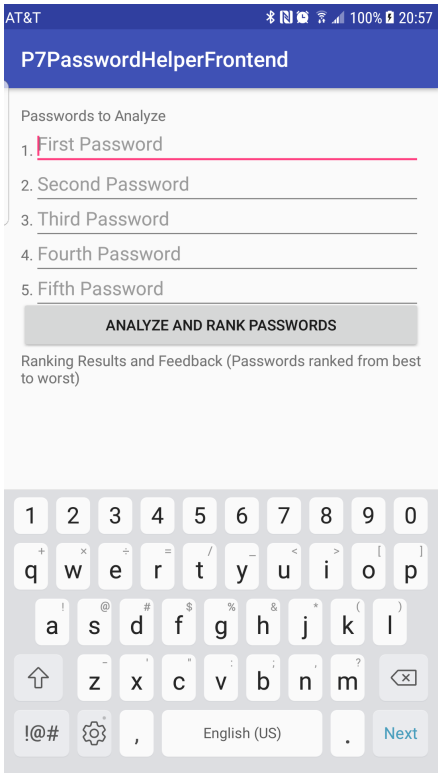


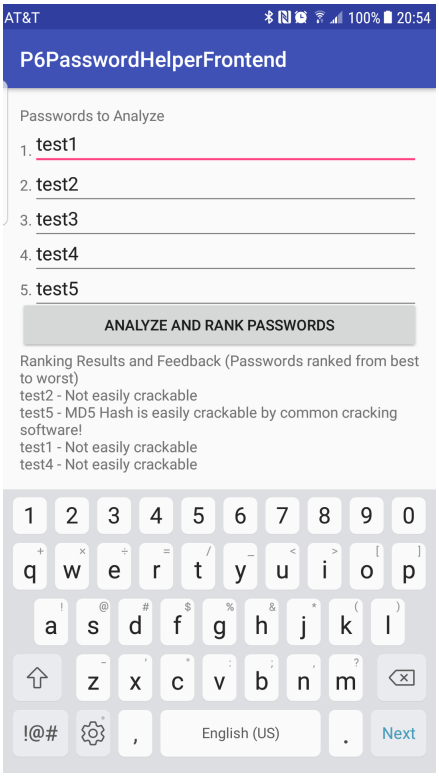
Status Report

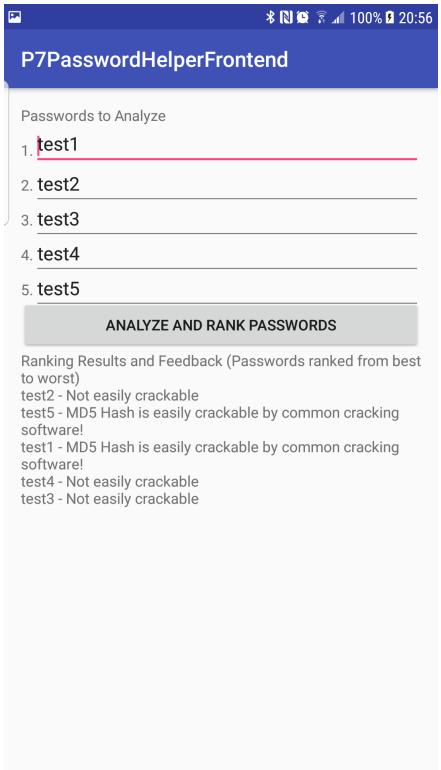
I was unsure in what way were intended to modify this apk

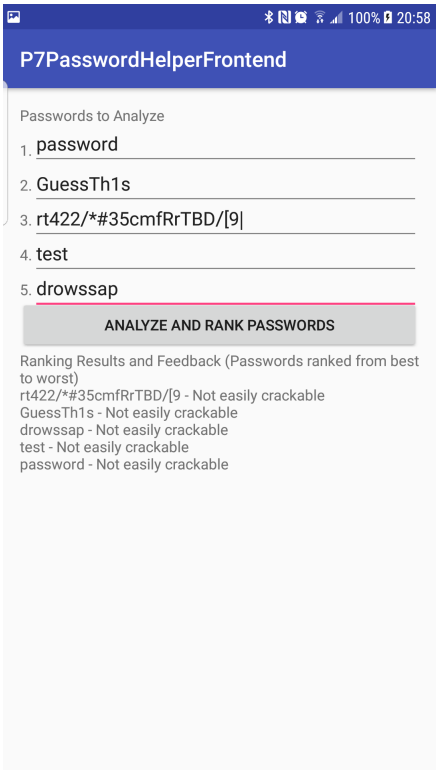
Task 5

Screenshots









Status Report

For Task 5, I added in hashcat support to my apk for P6. For each password entered, the backend (running in Amazon EC2) calculates the MD5 hash, then runs this through hashcat to determine if it can crack it. Hashcat running in the cloud is not working well with this though, as very generic passwords such as “password” and “test” are for some reason not cracked. I tried several different methods and key space rules for the hashcat command to see if that would have any affect, but to no avail. There also seems to be no rhyme or reason as to what passwords will be cracked and when. I.E in some tests the password “test5” would be cracked by hashcat and no other would (as in “test1”-“test4”) whereas in other tests, the password “test3” would be cracked, and no others. I am not sure what would be the cause of this, as even a simple word list would presumably be able to crack these passwords. I am using the rockyou.txt wordlist as in P6.