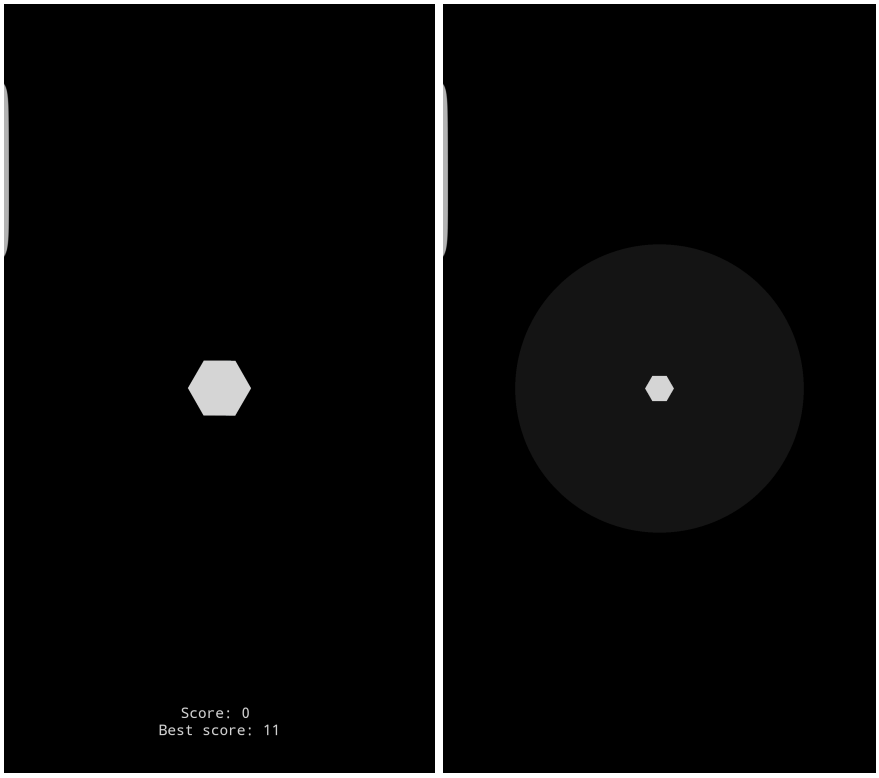
P5

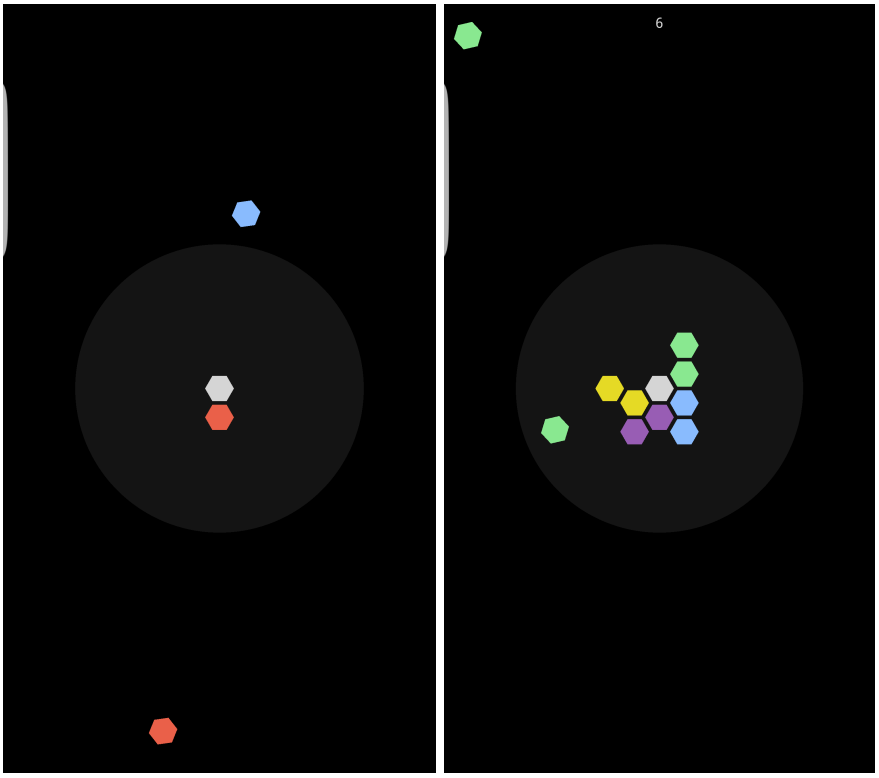
Ian Davis

W002icd

Github URL: https://github.com/IanDavis1995/P5

Task 1


Score: 0
Best score: 11


6

Status Report

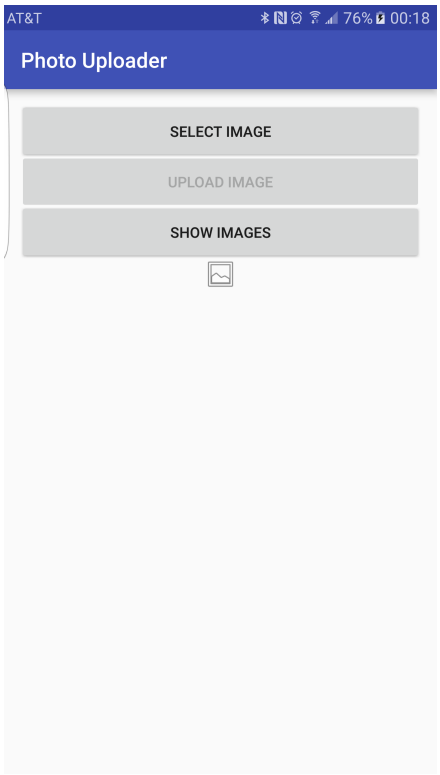
Some documentation may not be completely accurate, and some areas are lacking/vague in detail. Overall, mostly complete. The user's manual is rather short, as the general functioning of the game seemed self-explanatory.

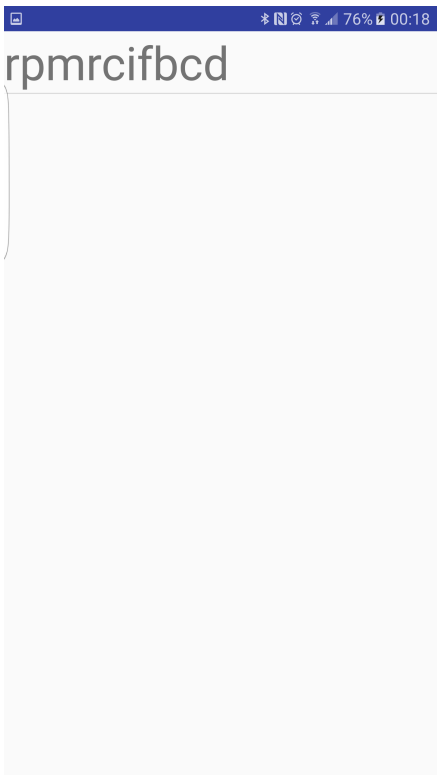
Experience Report

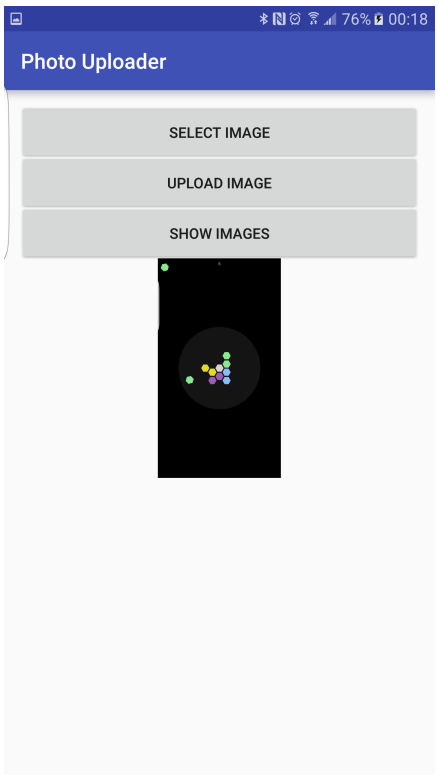
Most of this code was easy to follow, however I struggled to put together the functioning of the GraphGameHexagon, GameHexagon and GameField classes at first. Some debugging/stepping through update calls shed some light on how this all worked together and helped flesh out some documentation on this code. Being thorough with this required a lot of effort. I did not do very much with the MyGLRenderer class and Hexagon base class as a lot of this was some complicated OpenGL code with vertex shaders and projection matrices and buffers galore. I gave it my best effort but was not sure where to begin with understanding what all was exactly going on besides a very high level overview of what it seemed to do.

Task 2

Screenshots







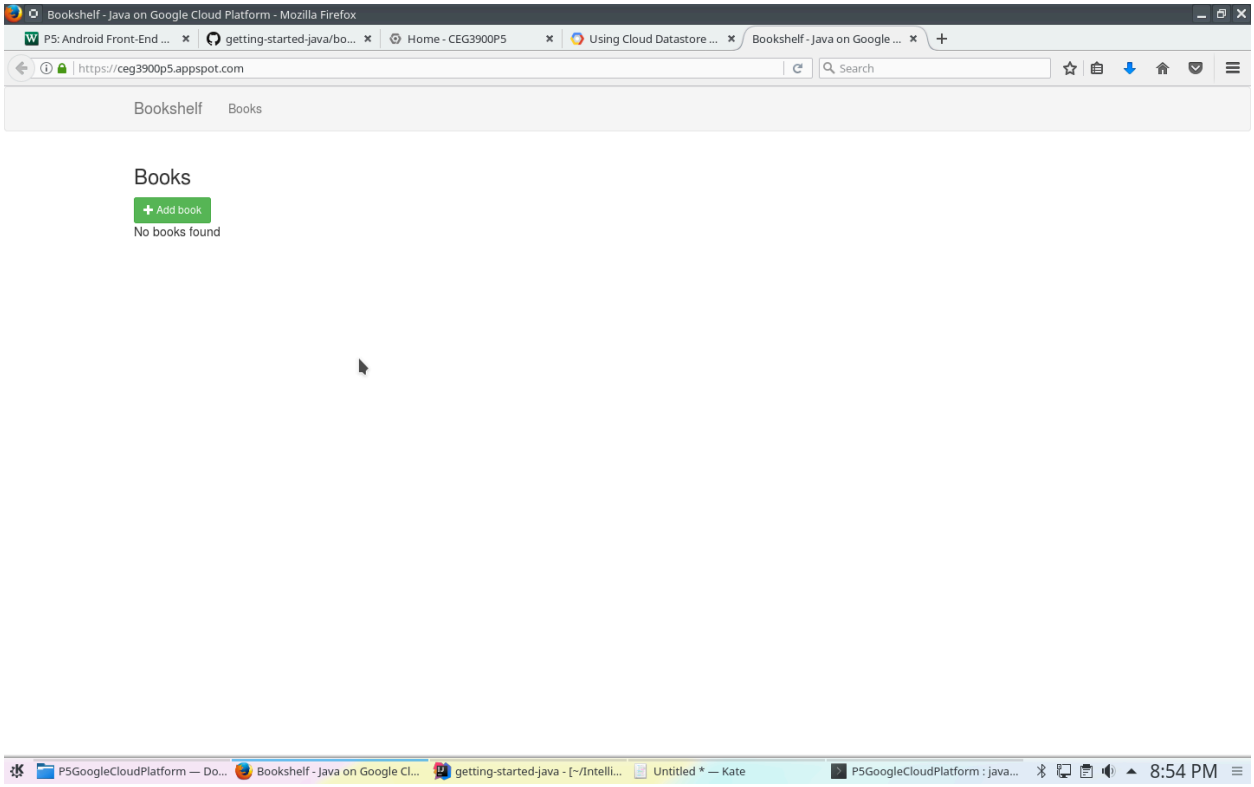
Status Report

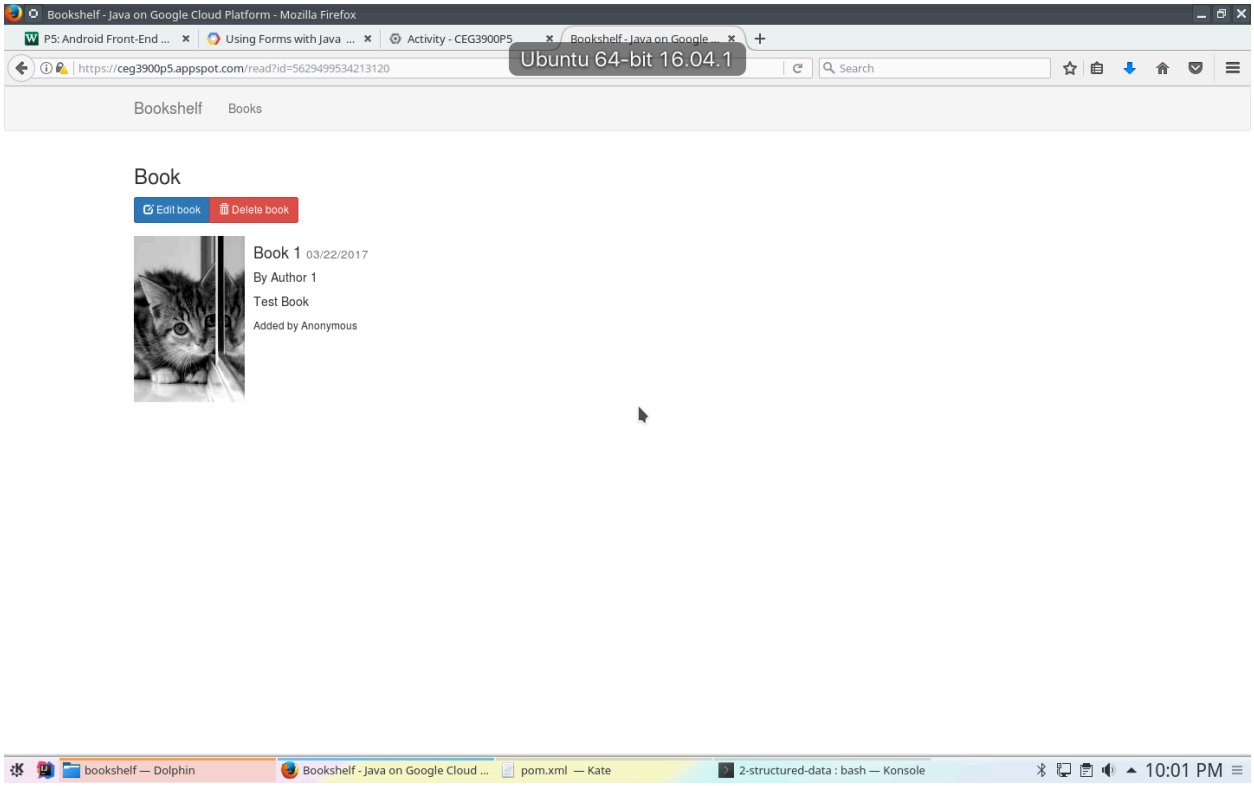
Fully Working, no issues found.

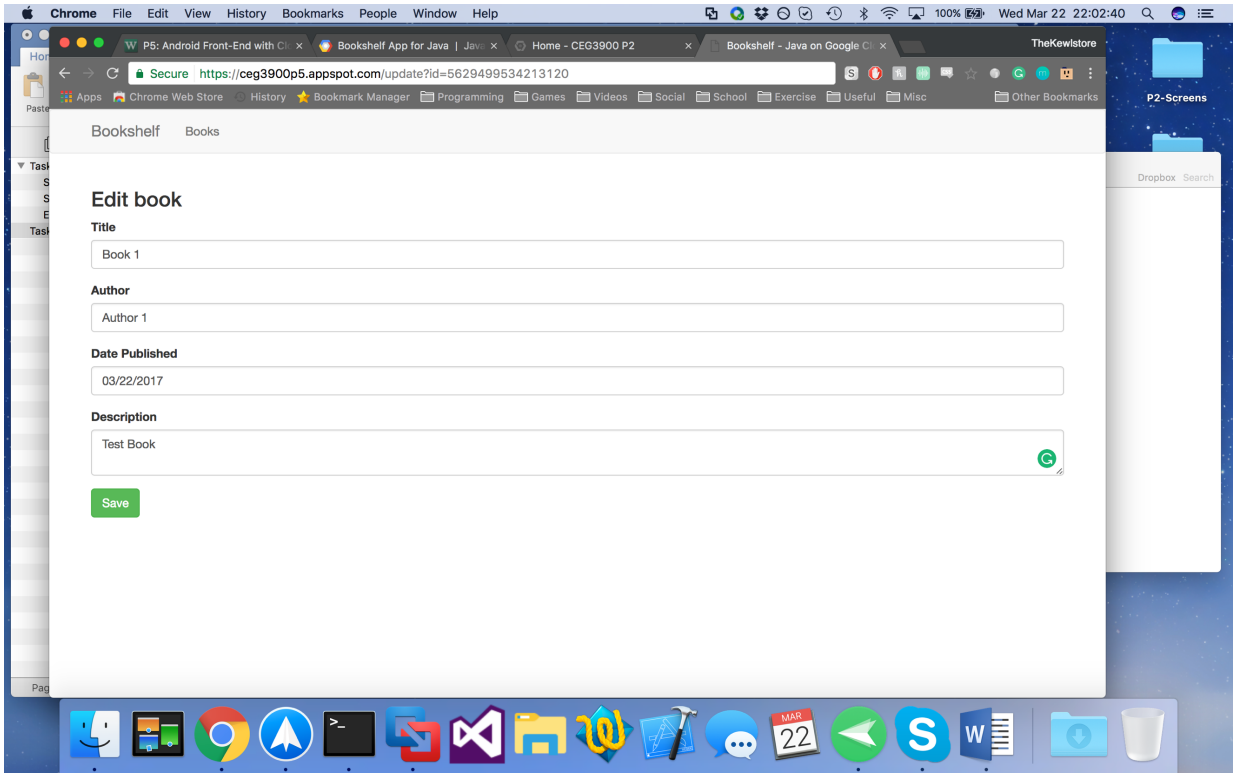
Experience Report

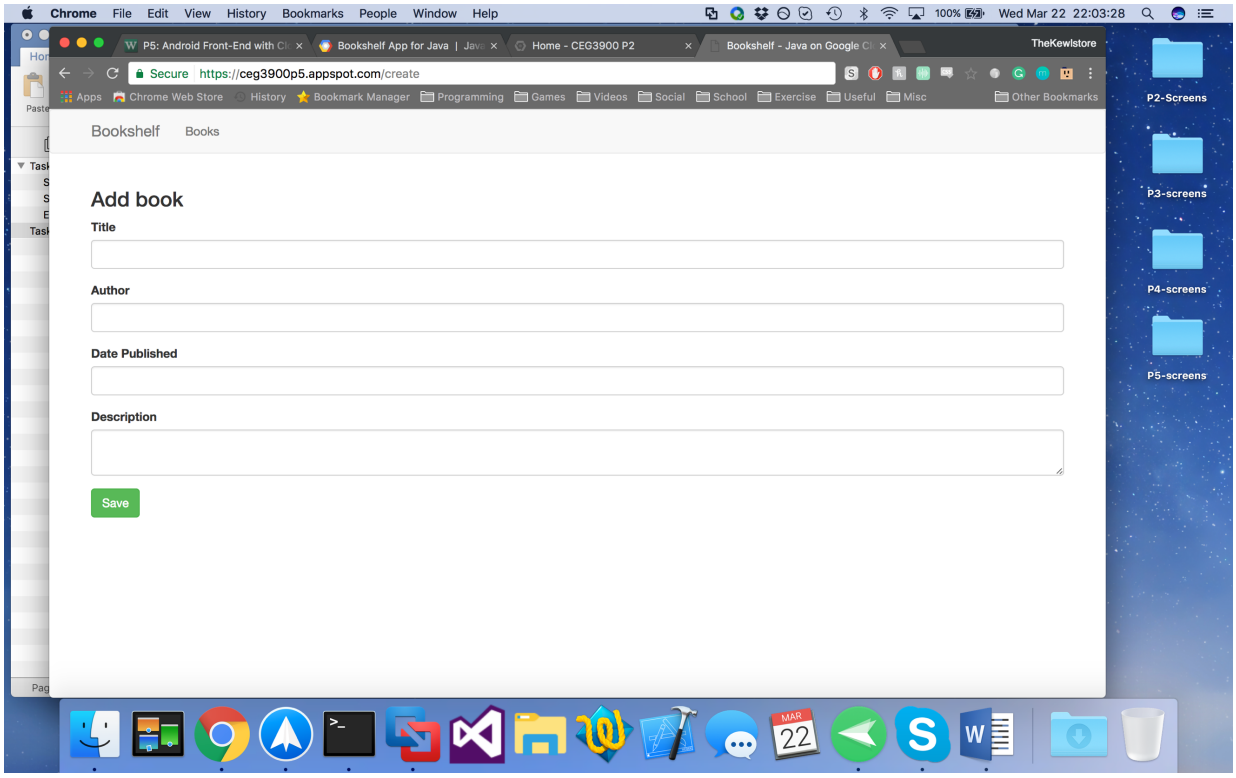
Setting up this APK and the Azure Blob Storage was very easy and user friendly. The tutorial didn't do the best job explaining that the ACCOUNT refers to a Blob Storage account on the Azure Portal, but it wasn't difficult to figure out.

Task 3


Bookshelf - Java on Google Cloud Platform - Mozilla Firefox
P5: Android Front-End ...
getting-started-java/bo...
Home - CEG3900P5
Using Cloud Datastore ...
Bookshelf - Java on Google ...
https://ceg3900p5.appspot.com
Search
Bookshelf
Books
Books
+ Add book
No books found
P5GoogleCloudPlatform — Do...
Bookshelf - Java on Google Cl...
getting-started-java - [~/Intelli...
Untitled * — Kate
P5GoogleCloudPlatform : java...
8:54 PM


Bookshelf - Java on Google Cloud Platform - Mozilla Firefox
P5: Android Front-End ...
Using Forms with Java ...
Activity - CEG3900P5
Bookshelf - Java on Google ...
Ubuntu 64-bit 16.04.1
https://ceg3900p5.appspot.com/read?id=5629499534213120
Search
Bookshelf
Books
Book
Edit book
Delete book
Book 1 03/22/2017
By Author 1
Test Book
Added by Anonymous
bookshelf — Dolphin
Bookshelf - Java on Google Cloud ...
pom.xml — Kate
2-structured-data : bash — Konsole
10:01 PM


Chrome File Edit View History Bookmarks People Window Help
Wed Mar 22 22:02:40
P5: Android Front-End with Cl
Bookshelf App for Java | Jav
Home - CEG3900 P2
Bookshelf - Java on Google Cl
TheKewlstore
Secure https://ceg3900p5.appspot.com/update?id=5629499534213120
Apps Chrome Web Store History Bookmark Manager Programming Games Videos Social School Exercise Useful Misc Other Bookmarks
Bookshelf Books
Edit book
Title
Book 1
Author
Author 1
Date Published
03/22/2017
Description
Test Book
Save
P2-Screens


Chrome File Edit View History Bookmarks People Window Help
Wed Mar 22 22:03:28
P5: Android Front-End with Cl
Bookshelf App for Java | Jav
Home - CEG3900 P2
Bookshelf - Java on Google Cl
TheKewlstore
Secure https://ceg3900p5.appspot.com/create
Apps Chrome Web Store History Bookmark Manager Programming Games Videos Social School Exercise Useful Misc Other Bookmarks
Bookshelf Books
Add book
Title
Author
Date Published
Description
Save
P2-Screens
P3-screens
P4-screens
P5-screens

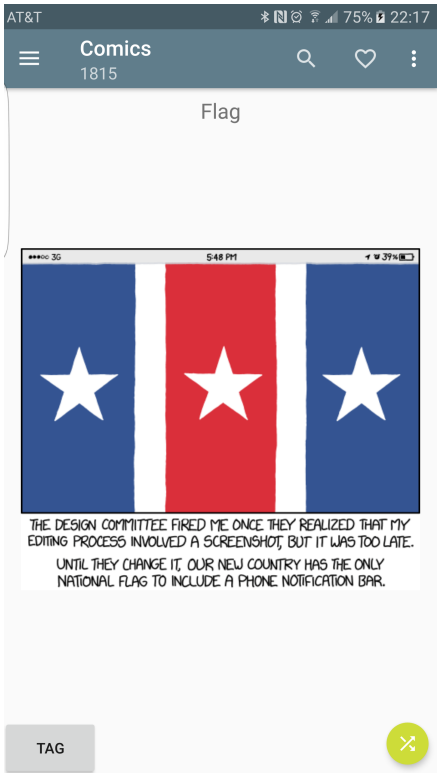
Status Report

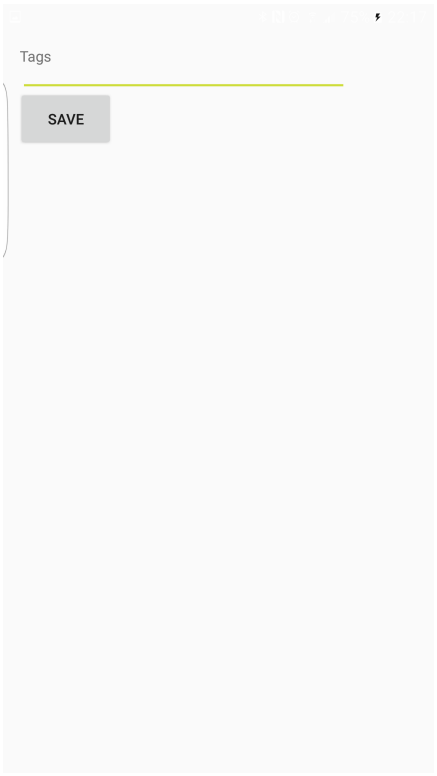
Mostly working, but images cannot be added to new books.

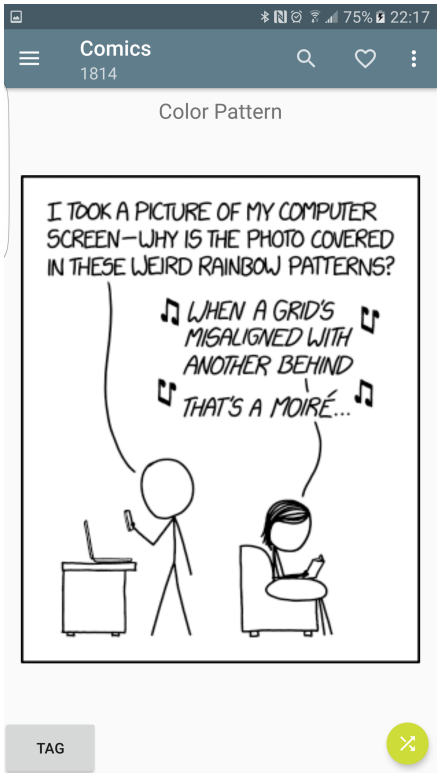
Experience Report

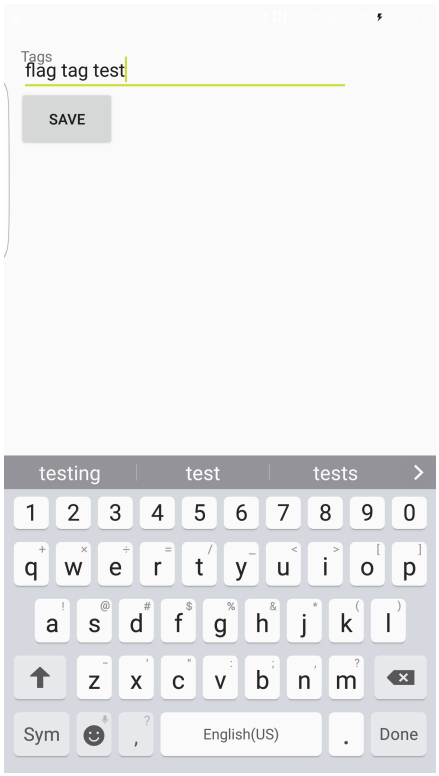
I struggled understanding how to get the permissions of the Blob Storage with the Client ID and Client Secret, the only way I could manage to successfully create new books was to set the default permissions to read/write. I also couldn't get the application to deploy locally, as I kept getting a weird chain of errors. Deploying it to the web worked throughout, however.

Task 4


AT&T
75% 22:17
Comics
1815
Flag
THE DESIGN COMMITTEE FIRED ME ONCE THEY REALIZED THAT MY EDITING PROCESS INVOLVED A SCREENSHOT, BUT IT WAS TOO LATE.
UNTIL THEY CHANGE IT, OUR NEW COUNTRY HAS THE ONLY NATIONAL FLAG TO INCLUDE A PHONE NOTIFICATION BAR.
TAG


Tags
SAVE


75% 22:17
Comics
1814
Color Pattern
I TOOK A PICTURE OF MY COMPUTER SCREEN—WHY IS THE PHOTO COVERED IN THESE WEIRD RAINBOW PATTERNS?
WHEN A GRID'S MISALIGNED WITH ANOTHER BEHIND
THAT'S A MOIRÉ...
TAG


Tags
flag tag test
SAVE
testing
test
tests
English(US)
Done

Status Report

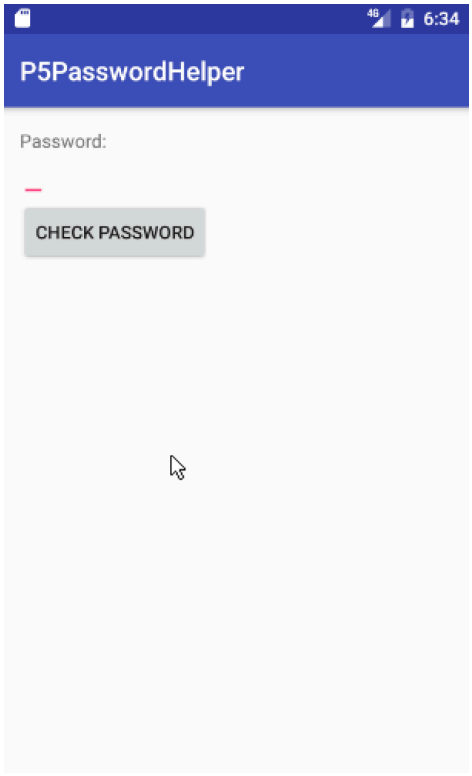
Mostly Functional. Tags are stored as a string for each comic id and loaded in when the tag button is pressed on a comic. They are saved when the save button is pressed. Tags are only stored on a per comic level, and comics cannot be looked up based on tags.

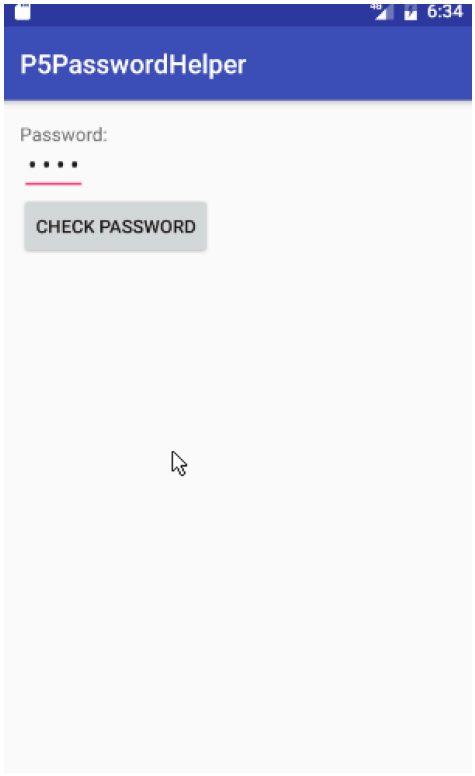
Experience Report

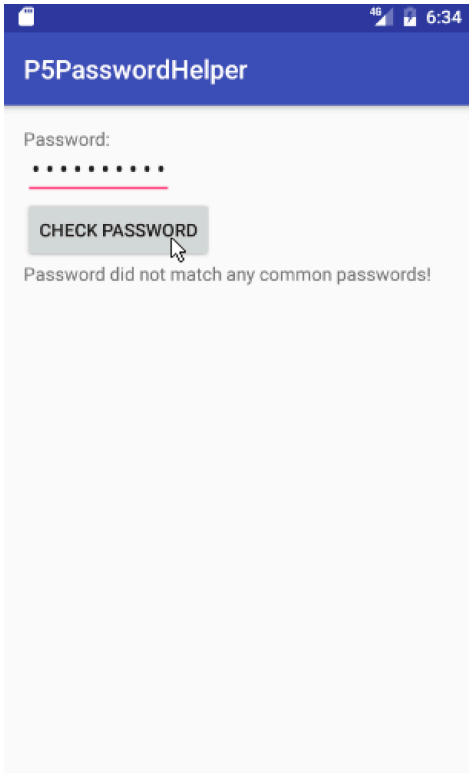
I had some trouble with getting the Easy XKCD Source up and running. The Android appcompat library was complaining about MenuItemCompat.setOnActionExpandListener being deprecated, even though all online references suggested using exactly that. I'm not sure if this was because of versioning or what, but ultimately, I had to comment out these calls. This did not seem to alter any noticeable behavior of the options menu, however. After these build issues were resolved, this task was straightforward. I chose to implement it with a Firebase database, storing tags as a string mapped to their xkcd comic id. Other than some issues getting the firebase libraries added into the easy xkcd project properly, this was a pretty smooth implementation. I struggled slightly to figure out how the xkcd viewer implemented the actual comic viewing pane through its fragments, as it was far from a trivial implementation of the UI.

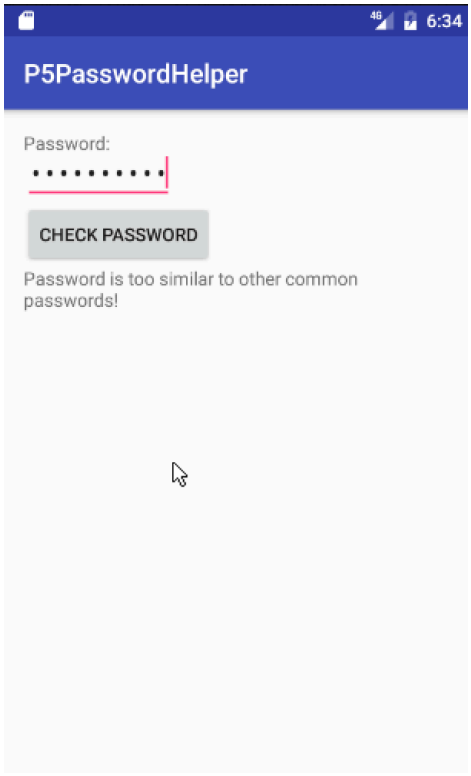
Task 5

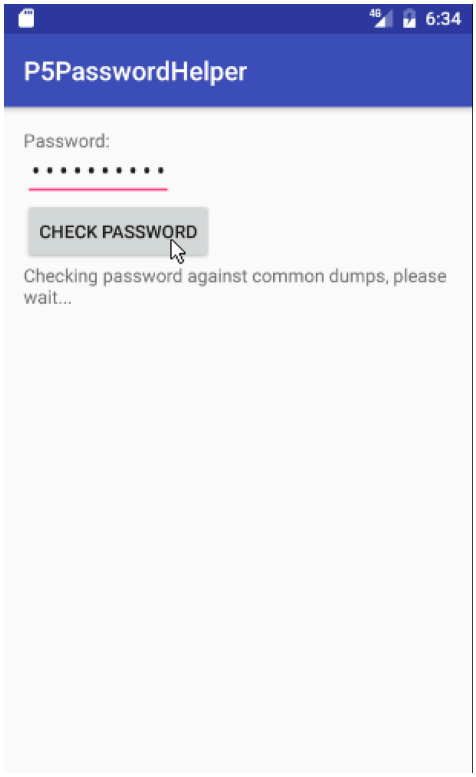
Screenshots











Status Report

Functional, not well optimized. Worst case computation time is about 6 seconds for a password with no commonality. Some of the most "common" phrases/words will be caught quickly, however.

Experience Report

I implemented this app using a google app engine HttpServlet backend, and sent Http get requests along, passing the password entered from the android application. The backend had a resource file of the top 1 million common passwords taken from the SecList github repository. It parses this file, reading in each password and checks for similarities based on the rules defined in the Definition of Similarity section of this report. I had a lot of issues trying to get the app engine

servlet up and running, and had to hunt down lots of tutorials of how to set everything up properly. I considered several different options for implementing this in the cloud, but decided on google app engine servlets as it seemed the most flexible to fit this specific need. The other main choice was Amazon EMR with an Apache Hadoop job, but I was even more lost on where to begin with that implementation.

Definition of Similarity

For this APK, I defined a password as being too similar based on the following rules

(NOTE: For this list, a "common password" is used to mean a password found in the common password dumps available in the SecList github repository):

- A common password exists such that 60% of its characters exist (in the same order) as the password given
- Said common password is no longer than the password given
- Said common password is no more than 3 characters shorter than the password given.

Several different tweaks of these rules were tested, but this gave the best balance between being decently "tough" on password uniqueness, but also not matching any conceivable password possible, without getting into some much more complex comparison algorithms.

Description of APK Benefits/Help

Beyond simply giving suggestions about the security of certain passwords, this app could also be useful to determine if a password already in use were already cracked or compromised on a public password dump. As exploits like this are increasingly common, an app that could scour some of the most common password dumps and determine if your password is already cracked, or even further, how easy it would be to crack, could be invaluable to end users.