CS 307 Sprint 2 Retrospective

Team 7: Michael Hockerman, Tylor Garrett, Nick Stanish, Travis Coria, Trevor Coria, Kyle Potts

Implemented and Working

- Allows users to add metadata for taks
  - Metadata successfully added to the tak, which is then successfully added to the server. Currently, the user has no way to access the metadata, which will be covered in Sprint 3.
- Connect Map View with correct Taks
  - On the website, there is an option next to the map to open up a new page which lists all the taks inside the map with their relative information.
- Web Logout
  - A user can successfully log out of the application. The logout sets the user to be logged out in the database as well.
- Google OAUTH Signon
  - When opening the application the user is presented with the option to authenticate with the application using the Google account already signed-in on their phone. We have both sign-in and sign-out implemented, though the exact placement on the UI might change in the coming sprint.
- JSON API Parsing
  - We parse JSON from the server to get updated map and tak IDs. It is done on a background thread and works as well as we would expect.
- QR Code Scanning
  - If the user has a QR code scanning application on their phone, our app uses it to scan QR codes generated by our server, then reach out to the server to download the map requested.
- QR Code Generation
  - The website generates a QR code for each map consisting of the map's unique ID. This QR code is displayed on the map details page for each map, and could be printed and/or accessed as the user desires.
- Maps JS library
  - Maptak.js wraps the functionality of google maps for easy reuse on multiple pages, and loads Google's code asynchronously

Implemented and Not Working Well

- Set Maps as public or private
  - The user is able to select the private status of their map, which is then stored in the MapObject in the app. This is not currently synced with the database. We are planning to complete this in Sprint 3.
- Set administrators for Maps
  - The user is currently able to add an arbitrary number of administrators for each map, which is then stored in the MapObject in the app. This is not

    - currently synced with the database. We are planning to complete this in Sprint 3.
- More Robust API
    - In creating the android application, the api was made for robust (checking for errors, etc). However this made the API code much more complicated and messy.
- Main Menu
    - Functionality for all of the buttons on the main menu is present. The UI is not presentable as a final product and will need to be refined in Sprint 3.
- Network Async Tasks for Addings Tasks and Maps
    - The currently Async tasks hang on the ui thread. This is due to the fact that we must wait for the server to respond with the newly created object's id on the server. Once it gets the id the async tasks returns and the ui thread is open to use again. This can be fixed by creating a database call that will have the ability to update a current map's id in the database.
- Search
    - Search functionality on Android is very rudimentary. It only searches the local cache of maps on the user's phone, and does nothing beyond just a String.equals() on the map name. This will be improved during sprint three to allow for more free-form searching, and possibly even searching through maps stored on the server.
- (Web) Editing maps
    - Users can create and delete maps, and add taks to maps, but they still can't rename maps or delete taks individually. This will be fixed in sprint 3.
- (Web) Editing taks
    - Users can change the title of a tak, but they cannot yet change the location. This will be fixed in sprint 3.

Not Implemented

- Solve multiple tak issue
    - This wasn't resolved and we could not find an efficient way doing it.This will most likely be removed from the requirements as the idea is not well formed.
- Database optimizations (Android)
    - Database performance was satisfactory enough during Sprint 2 to not warrant time invested in optimizing it beyond what is stupidly obvious. Moving forward, we will keep an eye on this and make optimizations, such as moving all database transactions off the UI thread, as necessary.