Sprint 2 Planning Document

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

Sprint 2 User Stories

- As a user, I would like the maps I create on the Android app to sync with the backend server.
- As a user, I would like the Taks I add on the Android app to sync with the backend server
- As a user, I would like to set maps as public or private.
- As a user, I would like to scan a QR code in the Android app to download a specific map.
- As a user, I would like the Taks I create on the website to belong to a specific map.
- As a developer, I would like an API to be available which, given a Google account OAUTH token, allows me to access that users' maps/Taks, add new maps for that user, and add Taks to existing maps belonging to that user.

Sprint 2 Tasks

- Travis Coria
    - (Android) Set maps as public or private (10 hours)
        - Functionality in CreateMap which would allow users to check whether they want their maps to be created as public or private. Private maps are only viewable and editable by the owner. Public maps are editable by the owner and viewable by anyone who possesses the MapID (which can be found through QR code scanning, searching, or other means). This information is synced to the local database cache and also to the server.
    - (Android) Set administrators for maps (15 hours)
        - Functionality in CreateMap which would allow map "administrators" to grant map edit privileges to other MapTak users. Administrators could enter users' email addresses, and these users would then have access and edit privilege to the map regardless of whether it is set to public or private. This information is synced with the local database cache and to the server.
- Trevor Coria
    - (Android) Allow users to set metadata for Taks (15 hours)
        - Functionality in AddTak which allows users to create arbitrary key:value pairs for each Tak. Each key:value pair is then synced to the local database and to the server. The keys can be whatever the user desires. Functionality might also be included in which a dropdown box

        - is provided of all the keys the user has already set for other Taks in a given map, so each Tak can have the same keys should the user desire.
    - (Android) Main Menu (10 hours)
        - A fragment which acts as the initial landing page for users in the application. Initially, this will only include a sign-in link. Once signed in, the user can, from this page, access CreateMap, MapList, QR code scanning, and search functionality.
        - Design will be important eventually, but for sprint 2 functionality is paramount.
- Tylor Garrett
    - (Android) QR Code Scanning (12 hours)
        - Application feature which uses an undecided external library to access the camera on most Android devices and recognize QR codes. These QR codes would contain a server-generated MapID. We would then use the MapID to access the MapTak API, download the appropriate map, and include in in the users' map list.
    - (Android+Server) QR Code Generation (12 hours)
        - Application feature which generates QR codes for maps on the user's Android phone. They can then export these QR codes using a standard Android share intent.
    - (Android) Search Interface (5 hours)
        - Class (fragment) which allows the user to enter a search query. This query is then sent to the server through an AsyncTask. The fragment should also be able to handle interpreting the search results from the server, though search functionality server-side will not be complete until sprint 3.
- Michael Hockerman
    - (Android) JSON API Parsing (10 hours)
        - Class which Takes in a JSONObject returned by the NetworkAsyncTask, parses it appropriately, and adds the containing information to the android application's sqlite database. This will all be done on a non-UI thread to ensure the application has optimal performance.
    - (Android) Google OAUTH Signon (10 hours)
        - Functionality which allows the user to sign-in to the application with their google account. If possible, we will use the android credential library to use the google account already signed in to their phone. This information will be stored somewhere in the app and included in API calls to the backend when accessing maps/Taks.
    - (Android) Database Optimizations (10 hours)
        - Currently, database calls are all done on the UI thread and thus significantly slow down performance. This will be changed so all

resource-heavy database calls are spun off into their own thread automatically. Especially for database queries, this will also likely require callbacks from the database to alert calling code that the transaction is complete.

- Kyle Potts
    - (Android) Network AsyncTask (12 hours)
        - We need a class in Android which, when triggered, will download all the necessary data from the MapTak API and present this JSON file to the JSON parser for parsing into the sqlite database. This class will extend Android's AsyncTask class. It might also consist of two classes: One for downloading information, and one for uploading new data to the server, depending on design decision which will become more obvious as the sprint progresses.

    - (Web) Connect Map View with with correct Taks(6 hours)
        - Currently the map view is not connected with the Tak view. We need to have it so when a user click on a map the correct Tak view(a list of all the Taks in the clicked map) it loaded. This include a change that when a Tak is added, its id is added to a list of Tak id's in the map object.
    - (Web) Account Logout(5 hours)
        - Currently there is no way to sign out of MapTak. We need to create a way to signout (with existing Google Sign in). Signing out should change the website view. All maps and Taks previously viewable should not be viewable. This should update the Account object in the NDB to show that the Account is logged out.

    - Make REST API more robust (5 Hours).
        - Currently the rest API is very messy and not well defined. We should create make the rest API more robust and make sure the API returns a correct JSON response when necessary.
- Nick Stanish
    - (Web + API) Edit Tak Information (7 hours)
        - Individual Tak information should be easy to update on the web site. The user must have editing permissions to edit (e.g. be either the creator or a manager). It should be as user-friendly as possible and allow for editing the title, coordinates, and meta-information. Delete option should also be supported but limited to the creator of the Tak or admin of the map.
    - (Web + API) Edit Map Information (6 hours)
        - Maps should be updateable from the website and have the necessary server hooks in the API. There should be a website form for easily adding Taks to a map, renaming the map title, removing Taks from a

map, moving Taks from one map to another, and adding/removing managers.

- (Web + API) Solve multiple Tak issue (5 hours)
  - At some point a user may want one Tak to be in multiple maps. For example, a museum could be part of both an “Indianapolis” map and an “Indiana” map. One map may be shared with some people and the other with different people. The creator of the map may not want one map to update another so we will create a new instance of the Tak when it is added to a new map. This will also be useful in maps with different privacy settings so that the Tak cannot be accessed if the user doesn’t have the permissions to view it.
- (Web + API) Homepage login (8+ hours)
  - The homepage currently does not update (other than a username appearing). To improve the user experience as well as increase usability, the homepage will be updated to a useful hub of different information for the user. Some information would be maps that the person owns, maps the user has permissions to edit, recently created Taks, and a search bar of their Taks.
- (Web) Maps Javascript library (5 hours)
  - Google maps are going to be displayed on many of the pages of our website. In order to make javascript calls easier and cleaner, we will have a static library of common methods for adding Taks, editing Taks, displaying many Taks, generating static images of Taks and facilitating changes to maps.