Realised the initial plan I made was closer to a waterfall approach. Did a reworked plan using an excel template that is split down into releases with each release having 4 sprints.

04/11/2019

LOGIN

Created initial single page setup

I used jquery mobile and jquery to create a single page setup. I had a number of issues getting jquery mobile to work with new version of jquery I have reverted to an older version for now. Will concider updgrading if I come across functionality I need. Page candy can be done using jquery mobile.

Made the login authenticator redirect to a login page instead of a popup as the popup UX on mobile is not very good

Notifications

Created a set of custom elements to allow notification data relating to the user to render. This is current unordered and will show every notification ever.

Intially had an issue with getting connection call-back to fire, this was a problem because I had misnamed the element registered to the custom registry. Found out I can use attribute change call-backs to modify displayed data on elements. This is particularly useful when dealing with asynchronous calls, like the promised used by firebase.

Created team invite accept mechanic to add new teams to the users own team collection. This will potentially be an issue when doing permissions. The user needs access to team data in order to add a reference in there own collection, permission may be achieved by adding a user reference in the team data someone to check against.

05/11/19

Meeting with Jarod confirmed I am on the right track with planning. Raised concerns of implementation without proper design. After I have proven that some of the higher risk elements of this release are possible I will review and enrich wireframes and use cases previously done.

I had to add these additional tasks to the current sprint as its imperative this is done early to reduce the change of UI redesign. Fortunately contingency and task over-estimation should extend the amount of work this sprint beyond workable limits.

Also raised concerns with GDPR particularly where sending personally identifiable information like names. This has made me rethink the invite team process as the ability to search for users by name or synonym will violate some GDPR principles.

06/11/19

Enriched some of the wireframes initial created as part of this process I created a new set of use cases and created flow diagrams to think about screens and process flow. This was inspired by a conversation with Jared about the different types of users considering company structure. I created a use case for a more vertical type of company where creation of tasks and team invites is reserved for management roles. Created flow diagrams to help inform the design of screens and application flow.

Refactored the notification class extracting query control and snapshot CRUD esc functionality into a separate class. This should be enough to create any updatable list from. This was helpful and help clean up the notification class. I did this in order to make a team custom element list.

The team list is another example of a query-controlled snapshot list. This query creates a list of divs from an accounts team data. This displays any teams that the user is subscribed to by having it in the collection under an accounts document. This may have potential issues with permissions, I believe access to the team document can be controlled in this way.

Created functionality to allow testing of notification invite system. This is just a form on the team row that “sends an invite” to any player. This is done by creating a team invite notification for the desired account, on accepting the account will then have a record of the current team. This allows permission control for both R/W access of a team as well as simplifying the query to access all team data.