Sprint Two Retrospective

20/09/2019

Attendees: Austen McClernon

After a successful sprint one, sprint two looked to expand the product from a single artifact registry usable for only one group of people, to allow users to have multiple groups which share artifacts. These groups were called circles and represent a network of people who have shared interest in a set of artifacts. One person is allowed to be a part of multiple ‘circles’. The acceptance requirements for this sprint are outlined below:

Requirement	**Result**
■ Be able to create a circle which allows for restrictive access based on user permissions.	**Requirement Met ->** backend API endpoints allows to create a new circle. The front end renders the circle available to the user. Conditional rendering ensures permission based access.
■ Be able to add members to a circle and remove members from a circle based on an admin type permission system.	**Requirement Met ->** API endpoints for adding and removing members from a circle successfully implemented and integrated with middleware and frontend. Permission authentication ensures that only admins can add and remove members, as per requirements.
■ Have a clean and simple landing page for when users first arrive to the at the website.	**Requirement Met ->** Landing page created successfully and the routing is set-up to ensure it is met upon arrival to the website. **Improvements ->** Work can be done to clean up and improve the UI and design of the landing page to make it more aesthetic.
■ Ensure that users can be members to multiple circles and that they should have access to	**Requirement Met ->** Backend modeling for circles and frontend rendering ensures that users have access to all of the circles that they are a member of. This is accessed by clicking on the circle button

the artifacts in the circle that they are apart of.	on the navigation bar.
■ User feedback extended for all actions, not only delete artifact, remove artifact and update artifact.	**Requirement Met ->** Snackbar component changed through the use of the package notistack, which allows for easy implementation of snackbar feedback and extends the usability of the snackbar to provide a more extensive range of user feedback.

Additional Non-Sprint Requirements Met / In Progress

Darkmode
Beta in progress for inbuilt instant message system.

Technologies Introduced During this Sprint

Client Dependencies Added	
1. react-animations	2. radium
3. notistack	4. react-autosuggest
5. react-test-renderer	6. redux-mock-store
7. socket-ioclient	

Server Dependencies Added	
8. Jest	

Post Sprint Team Discussion (SWOT)

Strengths	Weaknesses	Opportunities	Threats
Ability to extend the project to allow for circles.	Documentation has fallen a bit behind since implementation has become a priority.	Creating documentation for each process to allow the project to be technically understandable and make joining the team easier.	Project loses maintainability
Ability to meet deadlines & requirements	Testing. Bugs in the program is symptomatic of sub-standard testing.	Implementing a full testing suit can make the code more robust and ensure that there are no bugs.	Evergreen testing may seem make a testing suit appear effective but still allows for bugs particularly for edge-cases.
All members have been able to contribute effectively to the project			Feature-creep. One of the strengths has been the ability to introduce features to the project effectively. However, given that the next sprint is the last, feature creep may result in there being bugs in the final product.
The connection of the backend APIs with the frontend is smooth, maintainable and extendable.			
The structure, design and backbone of the			

product is fully implemented, meaning that the final sprint can be used to clean, and fortify the product.			

Post Sprint Review:

1. Looking through the weaknesses from the last sprint, it is great to see that most of the weaknesses were taken on board, and strategies were successfully implemented to fix them. In particular:
    a. Code Quality: Eslint was introduced to ensure a high standard of code-quality. This is shown through the A-rating from codacy. THis has helped to encourage active contribution from all members to the code and ensure that the project has been made maintainable, readable and extendable.
    b. Hard for all members to contribute to features: The implementation of peer-programming on a Friday 11-1 has ensured contribution from all members to the project.