

Team PorkBelly
Sprint 2 Planning
15th September, 2021

Sprint 2 planning was composed alongside a meeting held at 8pm of the day. The goal of this sprint is to develop Tags functionality and styling the product to create a ship-able version.

The team would like to allow themselves three weeks to complete this sprint, as this would coincide with ending on the 9th week of Semester. It would also be concluded just in time for the scheduled second review with the client on the 10th week of Semester (the week of October 4th).

User stories addressed:

The Tags functionality has been broken down into user stories: 10, 11, 12, 13, and 14. User should be able to attach / detach certain tags to their cards (10), filter their cards given a selection of tags (11), create (12), delete (13) and edit new tags (14) associated with their account.

This sprint also aims to style the then complete product as close as to the Figma prototypes as possible.

Tasks Extracted from the Product Backlog:

Task Number	Task Name	Relevant User Story(s)
T14	Implementing tag creation (PUT) endpoint	12
T15	Implementing tag update (PATCH) endpoint	14
T16	Implementing tag removal (DELETE) endpoint	13

Additional Tasks Created:

The additional tasks were created to complement those present in the Product Backlog. These includes implementing modular components to construct the functional interfaces for these user stories, extensions to the current code infrastructure to support these new features, as well as performance improvements to existing features.

These tasks were identified during a size estimation of the remaining open issues on the repository, where the team was engaged in splitting up the user stories in portions that could be estimated. These tasks were further split into issues on Github for which their sizes could be estimated and realistically completed. A transcription of the size estimation event can be found here: https://github.com/chomosuke/IT-PROJECT-PorkBellyPro/blob/docs/docs/team_minutes/sprint2Poker.pdf

Task Number	Task Name	Task Description	Relevant User Story(s)	Estimated Size
T17	Extending application infrastructure to support Tag Filter	Defining the software interface to declare: the fields of a tag controller functions for a tag	10, 11, 12, 13, 14	*

T17A	Backend support for Tags	Implementing back-endpoints for Tag CRUD operations	^	3
T17B	Extend application life cycle to support tag filter	Extending the application life cycle to hold a selection of tags to filter cards.	^	3
T17C	Declaring the Tag Component properties	Defining the component properties to display a tag's information	^	2
T18	Implement the controller methods for a Tag	Implementation of the Tag interface's methods to make the appropriate Api calls and its interactions with the front-end's application cycle.	12, 13, 14	5
T19	Creating a 'Tag' UI Component	Implementation of component(s) to display a tag on our user interface	10, 11, 12, 13, 14	1
T20	Creating a 'Tag Picker' UI Component	Implementation of component(s) that construct the UI for users to attach tags to cards, create, edit, and delete tags	10, 12, 13, 14	5
T21	Extend home page to consume the tag filter	Extending the card filtration logic to filter additional on the tags supplied	11	1
T22	Creating a 'Tag List' UI Component	Implementation of a horizontal bar of the user's tags on the Home page for them to select which tags to filter cards on.	11	1
T23	Extending the Header's search bar to include the tags selected by the user	Extending the search bar to show which tags are being used to filter the cards on to allow users to remove those tags from the filter.	11	2
T24	Visual Styling	Extract appropriate measurements and asset properties from the Figma prototype. These will be used to construct a Style Guide to aid everyone in styling all the	N/A	7

		components in the front end. Each member will be able to select which components to style		
T25	Change api to transfer image hashes	A team member has noticed that the browser makes an excess calling image retrieval due to lack of image equivalence checking. This task aims to let the browser fetch a card's image only when it knows the current image needs to be changed.	6	3
T26	Locking the selected card in the viewport display	Aims to ensure that a card being viewed remains in view despite the expansion of the details panel or screen resizing. If the card list has been scrolled, it would fix its scroll percentage on these actions.	6	2
T27	Implement parallel image processing	Aims to delegate the image processing tasks to separate thread so that the application's main rendering thread is unaffected.	3, 4	3
*: Split into smaller tasks during estimation ^: See above task				