

# Critika

CS 30700: Software Engineering 1  
Sprint 2 Retrospective

**Team 8:**

Brandon Sung  
Connor Todd  
Cyrus Santiago  
Keith Tan  
Ken Sodetz  
Julien San Diego

## WHAT WENT WELL

In general, we set up a lot of the necessary building blocks for our product to run. We made a lot of important backend routes and features, and most of the UI for the website. We also set up the necessary foundation on the frontend to easily integrate the API into the application. Building new features will be a lot easier now with the backend and frontend set up, and we should be able to work faster and implement even more in the following sprints.

As a user, I would like to be able to login and use my credentials for actions

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Implement frontend authentication	8	Frontend	Keith
2	Implement a listener so we can determine if a user is logged in	4	Frontend	Keith
3	Save authentication and username to cookies	2	Frontend	Keith
4	Automatically assign authentication token to all routes that require it	5	Frontend	Keith
5	Integrate backend route to login using frontend form input	3	Frontend	Keith

Users can successfully login into an unique account through our UI. They first must provide a username and password that they registered earlier. Then each time they start a new session (login), they are assigned an auth token for that session, and the auth token stays with them for all routes.

User Story #2 - 20 hrs

As a developer, I would like the frontend and backend to be integrated.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Integrate Add Submissions Page --	4	Front end	Brandon

2	Integrate Edit Submission Page	4	Front end	Brandon
3	Integrate Login Page ----	3	Front End	Julien
4	Integrate Reset Password Page	3	Front End	Julien
5	Integrate Edit Profile Page	3	Front End	Julien
6	Integrate Register Page ----	3	Front End	Julien

Currently, all of the components were integrated at some point (before or after we refactored). We integrated most of our components from sprint 1. However, we refactored too late, and were not able to integrate everything in our new version of the front end. Catching up shouldn't take too long next sprint, however we need to integrate all components in our newly refactored front end.

#### User Story #5 - 15.5 hrs

As a user, I would like to be able to view and utilize my Critika Coins

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Add Critika Coin counter with icon that persists on Navbar	1	Frontend	Brandon
2	Add Critika Coin counter to profile screen	1	Frontend	Brandon
3	Add option to spend critika coin on user's current submissions	1	Frontend	Brandon
4	Add option to spend critika coin on newly uploaded submissions on the user's "create submissions" page	1	Frontend	Brandon
5	Initialize 5 coins upon account creation	.5--	Backend	Keith
6	Create route that handles the accumulation of coins and reflect the change in the database	2--	Backend	Keith
7	Create route that handles	2--	Backend	Keith

	the spending of coins and reflect the change in the database			
8	Create Unit tests for Backend	2	Backend	Keith
9	Create tests for frontend	2	Frontend	Brandon
10	Integrate frontend and backend	3	Frontend/Backend	Brandon

The backend routes have been created and a solid front end UI also exists to go with it. We implemented Critika Coins as a base for a workaround to giving feedback. Each user is initialized with 5 coins, then they can spend their coins to post instantly instead of waiting to give 5 feedbacks. Each time a user adds feedback, they lose a coin.

User Story #6 - 7 hrs

As a user, I would like to be able to easily view all communities

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create page for viewing all communities in more detail	2	Front End	Cyrus
2	Generate thumbnails for each category	1	Front End	Cyrus
3	Include short description of each category	1	Front End	Cyrus
4	Make clickable link for each category	1	Front End	Cyrus
6	Create tests for frontend	2	Frontend	Cyrus

We all discussed how we want our communities system to and agreed upon a solid system. Communities are essentially like subreddits, with subscribers, posts, and categories (flairs). Right now we have a route that can simply show a user all communities ever created. We have backend routes for searching through communities and linking to communities. However, they still need to be integrated together.

As a user, I would like to be able to easily see submissions from other users I can give feedback to.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI carousels on dashboard page to display queue of user submissions	1	Frontend	Brandon
2	Create UI carousels on categories and explore screens to display trending or popular user submissions	1	Frontend	Brandon
3	Design algorithm for deciding who's queues a user's submission should be displayed on if they spent a Critika Coin	1.5	Backend	Connor
4	Create a backend route to pull submissions for display	2	Backend	Keith
5	Create unit tests for backend	2	Backend	Connor
6	Create tests for frontend	3	Frontend	Ken
7	Integrate Frontend and Backend	3	Frontend/Backend	Keith

Right now, there is a route in the backend “submission/all” that returns in the response all submissions ever submitted. There are more specific ones (/submission/mine and submission/all-community) that return submissions from a user or community. The frontend UI has components for these as well.

## WHAT DID NOT GO WELL

User Story #3 - 14.5hrs

As a user, I would like to be able to critique and give feedback to other users' submissions

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Add a critique button to user submissions and Generate a "Critique Submission" screen	0.5	Frontend	Julien
2	Include UI that displays the submission name, thumbnail, and type of feedback requested, along with a form to input text to critique the submission	1	Frontend	Julien
3	Update notifications screen of user whose submission was critiqued	1	Frontend	Julien
4	Give 1 Critika Coin to user who provided feedback	1	Backend	Keith
5	Create a backend route to handle the given critiques and add to the database	3	Backend	Ken
6	Create Unit tests for Backend	2	Backend	Connor
7	Create tests for frontend	2	Frontend	Julien
8	Integrate frontend and	4	Frontend/Backend	Keith

	backend			
--	---------	--	--	--

Because we had to refactor, We were not able to complete this user story, so we will have to push it to the next sprint. The backend route to give feedback to a submission is functional. Each feedback maps to a submission.

#### User Story #4 - 13.5 hrs

As a user, I would like to be able to rate the feedback I've been given.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Generate a "Rate Feedback" screen	1	Frontend	Julien
2	Include form to rate feedback on scale of 1-5 and to include text input fields for comments/rationale	1	Frontend	Julien
3	Update notifications screen of user whose feedback was rated	1	Frontend	Julien
4	Adjust critika score on profile page of user whose feedback was rated	1	Frontend	Julien
5	Create a route to handle the rating for the particular piece of feedback and update the rating in the database	2.5	Backend	Keith
6	Create Unit tests for Backend	2	Backend	Connor
7	Create tests for frontend	2	Frontend	Julien
8	Integrate frontend and backend	3	Frontend/Backend	Keith

This is another case where the backend is ahead of the frontend. Due to our major refactor, we were not able to get to integrate this route, however both components work independently, and we will finish this before starting sprint 3 user stories.

#### User Story #8 - 14.5 hrs

As a user, I would like to be able to report bad feedback.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Generate a "Report feedback" screen	0.5	Frontend	Julien
2	Include form to specify reason for report via generic dropdown options along with text input field to further specify rationale for report	1	Frontend	Julien
3	Update notifications screen of user whose feedback was reported	1	Frontend	Julien
4	Update notifications screen of admins to alert them of the report	1	Frontend	Julien
5	Update reports screen for admins to display the reported feedback	1	Frontend	Julien
6	Create a route to report poor feedback and mark the feedback for review in the backend	2	Backend	Ken
7	Create Unit tests for Backend	2	Backend	Connor
8	Create tests for frontend	2	Frontend	Julien
9	Integrate frontend and backend	4	Frontend/Backend	Ken

Similarly to the user story above, this route is implemented in the backend and the component is made, however we just need to send a request in the frontend and handle the response and populate the page before sprint 3.

User Story #9 - 13.5 hrs

As a user, I would like to be able to report bad comments.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Generate a "Report	0.5	Frontend	Julien



	comment” screen			
2	Include form to specify reason for report via generic dropdown options along with text input field to further specify rationale for report	1	Frontend	Julien
3	Update notifications screen of user whose comment was reported	1	Frontend	Julien
4	Update notifications screen of admins to alert them of the report	1	Frontend	Julien
5	Update reports screen for admins to display the reported comment	1	Frontend	Julien
6	Create a route to report poor comments and mark the comment for review in the backend	2	Backend	Ken
7	Create Unit tests for Backend	2	Backend	Connor
8	Create tests for frontend	2	Frontend	Ken
9	Integrate frontend and backend	3	Frontend/Backend	Julien

Similarly to the user story above, this route is implemented in the backend and the component is made, however we just need to send a request in the frontend and handle the response and populate the page before sprint 3. Each submission has an array of comment objects, which contain the comment text, username of commenter, and a flag for reported (along with a reason for being reported and a message).

#### User Story #10 - 15 hrs

As an admin, I would like to be able to unban users, and make another user an admin.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a UI that displays a form for which user to unban	3	Frontend	Cyrus

2	Create a UI that displays the users to make an admin	3	Frontend	Cyrus
3	Create route to unban a user	2	Backend	Connor
4	Create route to make another user an admin	3	Backen	Connor
5	Create Unit tests for Backend	2	Backend	Connor
6	Create tests for frontend	2	Frontend	Cyrus
7	Integrate frontend and backend	3	Frontend/Backend	Julien

We have made the backend routes for everything that will be on our Admin page. However we have not yet created a secure login page on the front end for admins. We plan on integrating a ton of features already built in the backend for Admins to ban and unban users along with looking at reported subs and comments.

User Story #11 - 13 hrs

As a user, I would like to be able to create a new community

#	Task Description	Estimated Time (Hours)	Team	Owner
1	On the categories screen a button should be accessible for a user to propose a new category	0.5	Frontend	Cyrus
2	Create a "create community screen"	0.5	Frontend	Cyrus
3	Include a form on propose category screen to indicate the community name, rationale for why category should be included, and a button to submit proposal	1	Frontend	Cyrus
4	Update notifications screen of admin whenever a user submits a proposal	1	Frontend	Cyrus

5	Create a route to submit the new category for review	2	Backend	Ken
6	Create Unit tests for Backend	3	Backend	Ken
7	Create tests for frontend	2	Frontend	Cyrus
8	Integrate frontend and backend	3	Frontend/backend	Ken

This will also be connected to the Admins page, so that Admins will be able to approve suggested communities and user will be able to use a UI to make that suggestion. Backend routes have been created to add new communities to our database.

#### User Story #12 - 11 hrs

As a user, when I'm critiquing, I would like the option to have my information anonymous from the person I'm critiquing

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Include radio button to indicate desire for anonymity on Critique Submission screen	0.5	Frontend	Cyrus
2	Hide user info from notifications screen and feedback received screen	1	Frontend	Cyrus
3	Create a route to switch anonymous boolean for a particular instance of feedback	1.5	Backend	Ken
4	Create unit tests for backend	3	Backend	Ken
5	Create tests for frontend	2	Frontend	Cyrus
6	Integrate frontend and backend	3	Frontend/Backend	Ken

Back end route has been created to ensure user anonymity, but we haven't implemented a critiquing submission UI yet, which we will add a checkbox for this user story as well as integrate that system to make changes in the database.

## **HOW WE COULD IMPROVE**

One of our biggest shortcomings in the previous sprint was that we were unable to implement frontend to backend integration due to lack of knowledge among our team with the original framework. So our main focus for this sprint (and first user story), was to integrate all of our sprint 1 routes and components. For the frontend components, we initially (at the beginning of the semester) decided to use React. However, none of us have had previous experience building a client server project using React. This is one reason why we were not able to integrate during

our first sprint, and this issue continued over to sprint 2. We had to refactor the entire frontend, because we did not properly prepare and organize our files and software. Better communication between all teams and the scrum master earlier on would have helped, and this would also have helped us organize better. To add on to that, our decision making could also be improved. We could have done some more research on redux to see the challenges, and maybe figure out a way to get through it earlier. In all, we think our communication, decision making, and organization could be improved. That being said, by the end of the sprint, we did a nice job catching everyone up and integrating (better communication and organization).

We think another improvement could have been our choice of user stories. We believe some of the user stories we chose to implement for sprint 2 were not as integral to our product as some others. To improve this, we should have an in person (not over the internet) meeting to select the most important stories. Again, the theme for this shortcoming is communication.

Additionally, since there are six members in the team, meeting conflicts is inevitable. However, during the first sprint, there was a number of meetings in which multiple team members were not present, which potentially caused some fallbacks in developing the product. Another way to further improve is making sure there's a time period in which everyone is committed in attending a group meeting. That way, discussions regarding the project's progression and development is relayed in a timely and efficient manner.

On the development side, there is also room for improvement. Our refactoring was a long process that set us back a little bit, and there is still a lot to accomplish on the UI side. We think doing this earlier would have helped, so completing user stories sooner will definitely help us in sprint 3.