

Rooted

Sprint 3 Retrospective

December 9, 2019

Team 2

Julien San Diego, asandieg@purdue.edu

Cyrus San Santiago, santiac@purdue.edu

Ken Sodetz, sodetz@purdue.edu

Ben Malone, malone74@purdue.edu

Abigehl Slattery, aslatter@purdue.edu

What Went Well

Last sprint, we ran into a lot of issues with integration, as there was much confusion and complications when trying to hook up the frontend and backend together. To combat that problem this sprint, we had two approaches for user stories. Both were successful. The first was to implement features and stories full-stack, assigning one person to see the work to completion from start to finish. Our second approach was paired programming. We ensured that if we were going to split a story's work between one frontend person and one backend person that those two individuals must sit next to each other in person to work out any kinks. This avoided a majority of the problems we faced last sprint. Additionally, we were also much more successful with our time management. Last sprint, there were many features that we tried to complete the night before our presentation. This resulted in faulty code and merge conflicts that broke a lot of our site's functionality, disabling us from demoing it properly. To avoid this, we agreed that no changes should be pushed too close to our deadline, and that we should rehearse and be familiar with our program at least two days in advance.

Unfinished Sprint 2 User stories

User Story #	User Story
1	As a user, I would like to be able to search for groups
3	As a user, I would like to be able to respond to invites to groups

Spillover Story #1:

As a logged in user on any page of the site, there is a “Search” option visible on the site’s navbar. Clicking this will navigate the user to a search page on which the user can enter any search term(s) and hit search and be displayed any trees which match that term. If there are no trees matching that term, the message “No tree contains this phrase” will be displayed.

Spillover Story #2:

If a user is invited to a group by an admin, they will receive a notification on their profile page. This notification has an accept and reject button on it. If the user accepts the notification, they will be added to the group and the notification will be removed from their notification list. If the user chooses to reject the notification, they will not be added to the group and the notification will be removed from the notification list.

User Story #1 (Backlog #42)

Estimated Hours: 17

[Group Member Announcement Request]

As a group member, I would like to be able to request for an announcement to be displayed to the group admin team.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create button to request an announcement to be added on the group page	1	Frontend	Ben
2	Create form to input details about announcement	2	Frontend	Ben
3	Create confirmation pop-up to submit request	2	Frontend	Ben
4	Create API route to request an announcement	4	Backend	Ken
5	Integration	4	Both	Ken
6	Unit tests	4	Both	Ben

Completed:

Under the announcements tab on the tree page, a group member can use the “Request An Announcement” button to request an announcement to display to the group admins. The popup box allows the user to enter any text they would like and hit send to submit their request. This request is then displayed on the “Pending Announcements” tab on the admin dashboard, giving the admins the option to accept the request, which displays the announcement on the group’s announcements tab, or reject the request, which removes the request from the pending announcements page.

User Story #2 (Backlog #34)

Estimated Hours: 19

[Approve/Reject Announcement Requests]

As a group admin, I would like to be able to approve or reject announcement requests

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI to display requested announcement requests	4	Frontend	Ben
2	Create API route to approve announcement requests	4	Backend	Ken
3	Create API route to reject announcement requests	4	Backend	Ken
4	Integration	6	Both	Abigehl
5	Create Unit Tests	2	Both	Cyrus

Completed:

On the admin dashboard under the pending announcements, all announcement requests from group users are displayed. Each request has an approve and reject button on it. If an admin chooses to approve the announcement, they will be prompted to confirm their decision then the announcement will be removed from the pending announcements tab and displayed on the group announcements page instead. Approved announcements are visible to all users who are part of a group. If the admin decides instead to reject the announcement, they will be prompted to confirm their decision, then the announcement will be removed from the pending announcements tab.

User Story #3 (Backlog #43)

Estimated Hours: 16

[Anonymous Message to Admins]

As a group member, I would like to be able to anonymously submit messages to the group moderation team.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create button on group page to submit anonymous message	2	Frontend	Ben
2	Create pop-up form to input message	2	Frontend	Ben
3	Create confirmation to submit form	2	Frontend	Ben
4	Create UI display on admin dashboard that shows messages without usernames	2	Frontend	Ben
5	Create API route to submit messages	2	Backend	Julien
6	Integration	4	Both	Julien
7	Create Unit Tests	2	Both	Cyrus

Completed:

On the group page, using the “Message The Admins” page, a group member can send any message to the groups admins. The button opens a popup window with a text box in which the member can write their message. Once the message is sent, it is listed anonymously on the admins’ “Messages” page.

User Story #4 (Backlog #31)

Estimated Hours: 21

[Group Admin | Add Announcement]

As a group admin, I would like to add announcements.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI display for an announcement	2	Frontend	Cyrus
2	Create a button to create a new announcement	1	Frontend	Cyrus
3	Create a form to take in announcement info	3	Frontend	Cyrus
4	Create a confirmation pop-up	1	Frontend	Cyrus
5	Create API route to create an announcement	4	Backend	Julien
6	Create API route to retrieve announcements	4	Backend	Julien
7	Integration	4	Both	Cyrus
8	Create Unit Tests	2	Both	Cyrus

Completed:

Similar to how group members can add announcements, the admins can use either the “Request An Announcement” button on the tree page or the “Add Announcement” button on the admin page to add their own announcements. Since admins do not need their announcements approved, entering a message and hitting send on either page will immediately display their announcement on the group’s announcements page.

User Story #5 (Backlog #32)

Estimated Hours: 11

[Group Admin | Remove Announcement]

As a group admin, I would like to remove announcements.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI component indicating removal of announcement	1	Frontend	Cyrus
2	Create UI component confirming removal of an announcement	1	Frontend	Cyrus
3	Create API route for removal of announcement	4	Backend	Julien
4	Integration	4	Both	Cyrus
5	Create Unit Tests	2	Both	Cyrus

Completed:

On a group's admin page, under the announcements tab, admins can view all the currently approved announcements along with a remove button. If the admin would like to remove an announcement, they can simply click the remove button, click on the confirmation popup, and the announcement will be permanently removed from the groups announcement tab.

User Story #6 (Backlog #33)

Estimated Hours: 23

[Add Non-Rooted Member]

As a group admin or moderator, I would like to be able to add non-rooted members.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a button for adding a non-rooted member	1	Frontend	Cyrus
2	Create UI component for making a non-rooted member	5	Frontend	Cyrus
3	Create API route for creating a non-rooted member	6	Backend	Ken
4	Create API route for retrieving a non-rooted member	4	Backend	Ken
5	Integration	5	Both	Cyrus
6	Create Unit Tests	2	Both	Cyrus

Completed:

Using the “Add Member” button on the groups homepage, admins can add any non-rooted member of their choice. This box allows the user to enter a name for the non-rooted member and optionally an email if they would like the invitee to eventually become a rooted member. Once the the “Add User” button is sent, the non-rooted member is immediately displayed on the homepage of the tree because the admin suggested this user, meaning that nobody else needs to approve it.

User Story #7 (Backlog #38)

Estimated Hours: 18

[Distinction For Rooted/Non-Rooted]

As a group admin or moderator, I would like to be able to visibly establish a distinction between non-rooted members and rooted members

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI for displaying non-rooted members	3	Frontend	Ben
2	Create route to establish rooted status	3	Backend	Julien
3	Create route to establish non-rooted status	3	Backend	Julien
4	Integration	4	Both	Julien
5	Unit tests	5	Both	Julien

Completed:

On a group's tree page all of the group's members are displayed under the heading Tree Members. The distinction between rooted and non-rooted members is made by showing rooted members with a green profile picture, and a clickable link to their profile, and non-rooted members with a black profile picture with no link to their profile.

User Story #8 (Backlog #41)

Estimated Hours: 17

[Group Member Add Non-Rooted Member]

As a member of a group, I would like to be able to add certain non-rooted-members to the group.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create button on group page to add new non-rooted members	1	Frontend	Ben
2	Create pop up form to input info about said members	2	Frontend	Ben
3	Create confirmation pop-up to add said member	2	Frontend	Ben
4	Create UI display on admin dashboard to notify them	2	Frontend	Ben
5	Create route to add non-rooted members	2	Backend	Abigehl
6	Integration	4	Both	Julien
7	Create unit tests	2	Both	Ben

Completed:

Using the add member button on the group page, under the non-rooted member section there are boxes to enter both a name of a non rooted member and their email. The user must enter at least a name and optionally an email. Once submitted, this suggested user will be displayed to the page's admins for approval. Once approved, the user will be displayed on the tree page.

User Story #9 (Backlog #21)

Estimated Hours : 18

[Sitewide Admin Close Group]

As a sitewide admin, I would like to be able to close any group.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a button to close/delete a group for sitewide admins	1	Frontend	Ben
2	Modify close/delete button to only appear for site admin	2	Frontend	Ben
3	Create a confirmation dialog that prompts the admin to confirm the action	4	Frontend	Ben
4	Edit API route close/delete a group	4	Backend	Ken
5	Integration	4	Both	Abigehl
6	Create unit tests	3	Both	Cyrus

Completed:

On the sitewide admin dashboard, all of the site's trees are displayed to the admins along with a "Delete Tree" button. If an admin clicks this button for any tree, they will be shown a popup asking them to confirm the permanent deletion of this tree. Once confirmed, the tree will be deleted from the database and no longer viewable on the site.

User Story #10 (Backlog #22)

Estimated Hours: 18

[Sitewide Admin Global Ban]

As a sitewide admin, I would like to be able to globally ban any user

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create ban button on user specific profile for sitewide admins	1	Frontend	Abigehl
2	Make button visible to sitewide admins only	4	Frontend	Abigehl
3	Create API route to globally ban users	5	Backend	Julien
4	Integration	5	Both	Cyrus
5	Create unit tests	3	Both	Ben

Completed:

On the sitewide admin dashboard under the users tab, all of the site's users are displayed to the admins along with a "Ban" button. If the admin decides to ban a user, they can click the ban button then hit ok on the confirmation popup. After this, the user will be added to the banned users page and no longer able to login to the site.

User Story #11 (Backlog #23)

Estimated Hours: 27

[Sitewide Admin | Global Unban]

As a sitewide admin, I would like to be able to globally unban any user

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a display for all banned users on the site	4	Frontend	Ben
2	Make display visible to only sitewide admins	4	Frontend	Ben
3	Create button to unban users	1	Frontend	Abigehl
4	Create API route to display globally banned users	5	Backend	Julien
5	Create API route to globally unban users	5	Backend	Ken
6	Integration	5	Both	Abigehl
7	Unit tests	3	Both	Cyrus

Completed:

On the sitewide admin dashboard under the banned users tab, all of the site's users are displayed to the admins along with an "Unban" button. If the admin decides to unban a user, they can click the unban button then hit ok on the confirmation popup. After this, the user will be removed from the banned users tab of the site and be added back to the users list. The user will once again be able to use the site as they previously could before being banned.

User Story #12 (Backlog #39)

Estimated Hours: 45

[Rooted Group Member Add/Edit Involvement/Year]

As a rooted member of a group, I would like to be able to add/edit my involvement/year.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Alter schema for tree so that it contains a field for user involvement and year	5	Backend	Ken
2	Create UI hover for user involvement/year	5	Frontend	Abigehl
3	Create UI component for creating user involvement/year	3	Frontend	Abigehl
4	Create API route for creating user involvement/year	4	Backend	Ken
5	Create UI component for editing involvement/year	6	Frontend	Abigehl
6	Create API route for editing user involvement/year	5	Backend	Ken
7	Alter API route for retrieving tree members	3	Backend	Ken
8	Integration	10	Both	Ken
9	Create Unit Tests	4	Both	Ben

Completed:

As a member/admin of a tree clicking the “Edit Your Involvement” button will allow a user to update the year they joined and left that specific group. The popup box provides two fields for entering this information, the user can enter either or both values. If they choose to enter a value for year joined, the tree members display on the main page of the group will be updated to reflect the year they entered.

User Story #13 (Backlog #36)

Estimated Hours: 10

[Kick Group Member]

As a group admin, I would like to be able to kick a member out of the group.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI component that allows an admin to remove a user from a given group	1	Frontend	Ben
2	Create API route to remove someone from a group	4	Backend	Julien
3	Integration	3	Both	Julien
4	Create Unit Tests	2	Both	Cyrus

Completed:

On the admin dashboard of a group, under the users tab, all members of that group will be listed for the admins to see. Alongside the names of the users, will be a kick button, clicking this will bring up a dialog box asking the admin to confirm the permanent removal of this member from the group. Hitting ok will remove the user from the group entirely, preventing them from seeing any of the groups information. Kicking differs from banning users in that banned users are still in the group but can no longer interact with the group.

Issues:

We encountered a bug which allowed you to kick yourself from a group. We were able to solve this issue however this fix was not pushed to our repository prior to our final presentation. It was fixed by ensuring the kick button was not displayed on the name which matched that of the current user.

User Story #14 (Backlog #30)

Estimated Hours: 23

[Group Admin | Color Scheme]

As a group admin or moderator, I would like to be able to customize the color scheme of my group page

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Add color scheme component to tree schema	1	Frontend	Abigehl
2	Add button to initiate edit color scheme form	1	Frontend	Abigehl
3	Add pop up form to input new color	4	Frontend	Abigehl
4	Add confirmation pop up	1	Frontend	Abigehl
5	Create API route to set color scheme	4	Backend	Abigehl
6	Create API route to get color scheme	4	Backend	Abigehl
7	Integration	6	Both	Abigehl
8	Create Unit Tests	2	Both	Ben

Completed:

On the admin dashboard, clicking the “Edit Tree” button will bring the admin to the customization page for their tree. If they would like to change the color of the group’s header and buttons, they can paste a hex value of their choice in the :”Edit Tree Color” field and hit submit. Once the value is submitted, the previously mentioned displays will now be the color the admin chose.

User Story #15 (Backlog #44)

Estimated Hours: 14

[Remove Group Photos]

As a group admin, I would like to remove photos from the group photos page.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI to select photo(s) for deletion	3	Frontend	Cyrus
2	Create a delete button	1	Frontend	Cyrus
3	Create a pop-up confirmation	1	Frontend	Cyrus
4	Remove photo from display on group photo library	2	Frontend	Cyrus
5	Create API route to delete photo	2	Backend	Julien
6	Integration	3	Both	Julien
7	Create unit tests	2	Both	Julien

Completed:

Under the photos tab on any group page is a location to display pictures uploaded by members of the group. In order to control the content posted there, admins have the ability to delete any images they may find inappropriate. They can do this by clicking on any photo, indicated by a green box around the image, and hitting the “Delete Picture” button which is only visible to admins. Clicking the button brings up a popup confirming the deletion, if the admin hits yes, the photo is permanently deleted from the group’s photos page.

What Didn't Go Well

In terms of what did not go well, we mostly lacked in our design choices. Due to time constraints, or difficulty of implementation, we had to make sacrifices and compromises to be able to deliver certain features. As an example, one of our stories allows users to change the color scheme of their page. We initially agreed that a user would be able to select an RGB value from either a color spectrum or palette of colors, but in the end our best solution was to have a form that takes in hex codes. Additionally, this sprint ended up being our largest in terms of how much work needed to be done. Because of the sheer volume of stories, it was sometimes difficult to determine who should work on what. We struggled and stressed over this for a while, until we eventually decided to use a to-do board. This board ended up working out great for us, as it gave us structure and organization on how to chunk up the tasks at hand.

How Should We Improve?

While we made significant improvements from Sprint 2 to Sprint 3, there is of course always more that can be achieved. For starters, if we were to continue this project, future goals would definitely involve redesigning features to meet the criteria of our initial expectations. Since we made a lot of sacrifices and compromises, it would be nice to go back and flesh out some stories to be even greater. Additionally, there are many features that could definitely be added to the site on top of what we have already developed. A suggestion from professor Dunsmore was that we could give users the ability to sort members in a tree by categories, instead of just by the year they joined. Lastly, there are many stories that could have had better visual aspects with cleaner and more user-friendly HTML/CSS. However, because of time constraints, we sometimes settled for less, as long as they functioned properly. We could certainly improve upon such stories to make our site as best as it can be.