Rooted

Sprint 2 Retrospective
November 4, 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

For this sprint, we assigned ourselves a more manageable amount of user stories which allowed us to finish a majority of them in addition to other stories which were not in our original sprint 2 plan. We were able to better estimate how long each task will take, which allowed us to spare time to solve any issues we encountered throughout the sprint. Thanks to our much more diligent planning and scheduling we were able to implement all of our overflow features from sprint 1 and a majority of what needed to be done for this sprint. Diligent planning required each member of our team to put in extra work to constantly communicate what they were working on and when they could get it done by. Every member stepped up and put in the extra effort to stay in contact with the others which ensured our planning didn't go to waste. Our work this sprint leaves us in a good position for sprint 3 to smooth out the rough edges of the website as well as add any additional cornerstone features to ensure that we meet our goals for the website. While we do have a significant amount of work ahead of us, if we use what we learned during the sprint and build upon it, we should be in a good place by the end.

User Story #2
Estimated Hours: 27

As a user, I would like to be able to request to join a group

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a button to request a join a group	1	Frontend	Cyrus
2	Create a confirmation dialog to confirm that the request has been sent	4	Frontend	Abigehl
3	Create notification display	4	Frontend	Ben
4	Create API route to send a request to join a group	5	Backend	Ken
5	Create API route for notifications	5	Backend	Ken
6	Integration	7	Both	Cyrus
7	Create unit tests	3	Both	Benx

Completed:
As an already registered Rooted user, you cano view the tree pages for other groups, with the amount of information displayed depending on whether the group is public or private. Regardless of the groups privacy status, a button displays "Request to Join Group". Clicking this button will send a request to the admins of that group, which they can then approve or deny. When a user clicks the request button, a request is properly sent to the admins which they can view on the admin page.

User Story #3
Estimated Hours: 23

As a user, I would like to be able to respond to invites to groups

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a page for users to view all invites to groups	5	Frontend	Abigehl
2	Create a button to accept an invitation to a group	1	Frontend	Abigehl
3	Create a button to reject an invitation to a group	1	Frontend	Abigehl
4	Create API route to accept an invitation to a group	4	Backend	Julien
5	Create API route to reject an invitation to a group	4	Backend	Julien
6	Integration	5	Both	Cyrus
7	Create unit tests	3	Both	Ben

Completed:
When a user invites another to a group using the "Add User" button on a group page, this sends a request to the admins of the group to invite the suggested user. If the admin agrees to send the invite, the user will receive a notification, viewable on their profile page, which will allow them to either accept or deny the invite to the group. Accepting the invite will add the user to the group, allowing them to view the group's contents.

Issues:

While we were able to implement the majority of this feature, we did not successfully complete the rejecting invite feature. This means that when the user rejects an invitation, the rejection goes through but the notification is not removed from the array and is still visible on the user's notifications page. This rejected notification should no longer be visible in the notification page.

User Story #4

Estimated Hours: 18

As a group admin, I would like to be able to close/delete my own group

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

Completed:

Admins of a group have the ability to view the admin's page for a group which provides them with many options regarding the group. One of these is to permanently delete the group and its contents. To do so, the admin simply needs to click the "Delete This Tree" button which brings up a popup asking the admin to confirm their decision, here they can either confirm the deletion or back out. If they choose to delete the group, the group is permanently removed from the database and will not be displayed on any user's homepage. Backing out of the deletion popup will simply leave the group as is.

User Story #5
Estimated Hours: 22

As a group admin, I would like to be able to make my group public or private

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI drop down button that for public or private	3	Frontend	Ben
2	Implement UI indication to show if a group is private or public	1	Frontend	Ben
3	Create API route to make a group private	5	Backend	Ken
4	Create API route to make a group public	5	Backend	Julien
5	Integration	5	Both	Cyrus
6	Create unit tests	3	Both	Ken

Completed:
After integration of both frontend and backend, if a user of a group is an admin, he or she is able to go to the admin dashboard and make a group private or public. If a group is private, only members of the group are allowed to view the group. If a non-member navigates to the group page, a message will show that the group is private and shows a button that allows the non-member to request to join the group.

User Story #6
Estimated Hours: 23

As a group admin or moderator, I would like to be able to invite non-members to join my group

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a button to invite a non-member	1	Frontend	Abigehl
2	Create a UI form to send an invite to an individual without a Rooted account	4	Frontend	Ben
3	Create an API route to invite non-members in Rooted to join a group	5	Backend	Ken
4	Create an API route to send invites to individuals without a Rooted account	5	Backend	Julien
5	Integration	5	Both	Cyrus
6	Create unit tests	3	Both	Ken

Completed:
When logged in as an admin/moderator, a user is able to click the invite user button on a group page. A popup will request a username. Once the moderator/admin has typed in a username and clicked submit, the web app will send an invite to the user specified. If the user does not exist or the username is incorrect, the admin will be notified via a popup. The invite will be sent to the user's notifications and if the user clicks accept on that notification, they will be added to the group.

User Story #7
Estimated Hours: 25

As a group admin or moderator, I would like to be able to approve or reject non-member requests to join my group

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a page for groups to view all requests from non-members	5	Frontend	Abigehl
2	Create a button to accept a request	1	Frontend	Ben
3	Create a button to reject a request	1	Frontend	Ben
4	Create API route to accept a request from a non-member	5	Backend	Julien
5	Create API route to reject a request from a non-member	5	Backend	Julien
6	Integration	5	Both	Cyrus
7	Create unit tests	3	Both	Ken

Completed:
When a user requests to join a group they will be added to the pending users list found on the admin dashboard. The admin dashboard is accessible to admins of a group and can be found by pressing the admin dashboard button on a group page. When viewing the list of pending users, an admin will be able to accept or reject a user. Accepting a request will add the requested user to the group while rejecting a request will remove a user from the list.

User Story #8
Estimated Hours: 18

As a group admin or moderator, I would like to be able to edit our about page

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a UI form for editing Group Information	4	Frontend	Abigehl
2	Add submit button	1	Frontend	Ben
3	Create API route to edit Group	5	Backend	Ken
4	Integration	5	Both	Cyrus
5	Create Unit tests	3	Both	Ken

Completed:
After integration of both frontend and backend, if a user of a group is an admin, he or she is able to go to the admin dashboard and edit the group page, such as the name of the tree, the image of the tree, the description of the tree, and the bio of the tree. Any changes made will be reflected on the group page. The “bio” section of the edit page directly correlates to the about page which is displayed on the public tree. This can be changed so that bio is called “about” so that there is no confusion.

User Story #9
Estimated Hours: 18

As a group admin or moderator, I would like to be able to ban users from my group

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create ban button on user specific profile	1	Frontend	Abigehl
2	Make button visible to group admin and moderators only	4	Frontend	Abigehl
3	Create API route to edit Group	5	Backend	Julien
4	Integration	5	Both	Cyrus
5	Create unit tests	3	Both	Ben

Completed:
After integration of both frontend and backend, if a user of a group is an admin, he or she is able to go to the admin dashboard and ban a user by clicking ‘ban user’. If the user is banned, the user’s name will appear on the banned users list. Once the user is banned from the group they should no longer be banned and the ban button should no longer appear.

User Story #10
Estimated Hours: 27

As a group admin or moderator, I would like to be able to unban users from my group

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a display for all banned users of a group	4	Frontend	Ben
2	Make display visible to only group admin or moderators	4	Frontend	Ben
3	Create button to unban users	1	Frontend	Abigehl
4	Create API route to display banned users	5	Backend	Julien
5	Create API route to unban users	5	Backend	Ken
6	Integration	5	Both	Abigehl
7	Unit tests	3	Both	Cyrus

Completed:
After integration of both frontend and backend, if a user of a group is an admin, he or she is able to go to the admin dashboard and unban a user by clicking ‘unban user’. If the user is banned, the user’s name will appear on the banned users list. The admin can navigate to the banned users lists and click the ‘unban user’ button. After a user is unbanned, the user will be removed from the banned users list.

Additional Sprint 2 User stories

#	User Story
11	As a group admin, I would like to be able to add other users to be an admin **Completed:** When a group is created, the user who created the group is automatically assigned an admin role. The admin can decide how many members the admin team should consist of. The admin can go to the admin dashboard, look at the list of users, and click on the button 'Make admin'. Upon adding a selected user as an admin, the user will have admin privileges.
12	As a member of a group, I would like to be able to upload photos to my page within the group **Completed:** Each group has a photo album where users can upload images that all other members are able to see. Any member of the group can simply go on the 'Photos' tab and see all pictures uploaded by any members or upload a photo by clicking the 'Upload' button. The user will then be prompted to choose an image from his or her files.
13	As a member of a group, I would like to be able to request to the group moderation team that certain non-members be invited to the group **Completed:** Not every individual in a group is immediately added to a group as the number of members may be too big for admins to remember to add everyone. Members of the group have the ability to ask the admin team if a certain non-member can be invited to a group. A user will click 'Request member to be invited' button where an input box will show prompting the user to add the non-member's username. After clicking 'Request', the admin team will get a notification showing which members are asked to be requested. If approved, the admin team will send the requested user an invite.
14	As a user, I would like to be able to receive notifications **Completed:** If a user is invited to join a group, the user will receive a notification saying that the admin team of a group has invited him or her to join the group. The user has the option to accept or reject the invite. If a user chooses to accept the invite, the user will automatically be added to the group.

What Didn’t Go Well

While our additional work in planning the sprint did benefit us quite a bit, we still ran into some issues with completing all of our work. These issues mostly stemmed from the fact that sometimes we failed to focus on completing the most important work first and instead getting sidetracked to work on extra features which may not be necessary. If we had ensured every story on our planning document was completed and functioning properly before we moved on to additional tasks, we likely would have been more successful this sprint.

Another issue we faced was working on getting tasks done much too late into the sprint. Only hours before our sprint review we raced to complete some user stories which ended with us fixing them but then subsequently breaking them right before we were due to present. If we had prevented ourselves from touching any code before the review, our presentation as a whole would have gone much smoother. This issue also was due to a breakdown in communication where we were unable to solve inconsistencies between frontend and backend implementation in a timely manner, leaving us to do it right before the sprint review.

User Story #1
Estimated Hours: 26

As a user, I would like to be able to search for groups

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a page for search and display findings	5	Frontend	Abigehl
2	Add a UI search bar to search for groups	4	Frontend	Abigehl
3	Add a link or button to navigate to group page	1	Frontend	Abigehl
4	Create API route to retrieve all groups	4	Backend	Julien
5	Create API route to retrieve specific group information	4	Backend	Julien
6	Itegaration	5	Both	Cyrus
7	Create unit tests	3	Both	Ben

Incomplete :

We currently display a search bar on all of the pages of our web app but it does not function. Instead of being able to search for groups, clicking the search button just displays all the groups that user is a part of. The goal for this feature was to enter a search term and find all groups which fit that term.

How Should We Improve?

In the future, communication between team members could be improved. It is understandable that each member has other commitments besides this class, but it is important to let members know if he or she is unavailable. There has been occasions where a team member needs to discuss something with another member, but there’s a disconnection in communication. We believe that this can be improved by simply letting team members know if they are unavailable, and if so, let the rest know when they will be available to discuss. In addition to such communication, we should also discuss and communicate design decisions as a team to remove any confusion when it comes to implementing. The logic of both frontend and backend should be heavily discussed and agreed upon, so that when implementing a feature, the design decision is set and individuals working on the frontend and backend are aware of how things should be implemented.

As a team, we should better set up standards on how we work to create a better workflow. For example, when implementing a new feature, a new branch should be made. Upon completion of a feature, we should be ensuring that new code does not break the old code before merging with our main branch. This would make it easier to detect which merge created problems and if needed, the main branch can be easily reverted. In the future, we should set up a soft deadline. By setting up a soft deadline, we are giving ourselves extra time in case an issue arises and prevents us from pushing broken code at the last minute.

In our next sprint, if we better structure our standards and how we work collectively, we will be able to reduce merge conflicts, issues with broken code, and have a better workflow.