Rooted

Sprint 1 Retrospective
October 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

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. 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. Now that all of our group members have had time and experience working with our project's technologies, it should be easier for us to assign tasks as we understand what everyone is interested in and best at working on. This should greatly benefit our productivity as we can work independently and not have to spend as much time learning the technologies.

User Story #1
Estimated Hours: 29 hrs
As a user, I would like to be able to register and verify a Rooted account.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Set up Back End server	3	Back End	Ken
2	Set up Mongo database	2	Back End	Ken
3	Set up deployment server	2	Back End	Ken
4	Create UI form to enter information about user	3	Front End	Ben
5	Create button to send information to Back End to register	1	Front End	Ben
6	Create API route to register user and add them to database	6	Back End	Ken
7	Implement API route to generate a verification token	6	Back End	Ken
8	Implement mailer to send verification token	3	Back End	Ken
9	Create Unit Tests	3	Front End/Back End	Cyrus

Completed:

The servers and database setup took a little longer than anticipated, but they were the first things completed for this entire project. The UI form to sign up and register were done without any issue, and the API routes (/register and /verify-email) were made. The mailer sends an email to the email provided by the user with a verification token that they must use to verify the email, along with a message from the team. The backend has extensive error checking, and the unit tests made check all possible registration issues (register without username, email, invalid password, etc.). All unit tests were passed.

User Story #2

Estimated Hours: 16
As an existing user, I would like to be able to login

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI form to Login	2	Front End	Abigehl
2	Create button to send information to Back End to login	1	Front End	Abigehl
3	Create API route to login a user	2	Back End	Ken
4	Set up error handling	2	Front End	Cyrus
5	Set up a user's homepage	7	Front End	Abigehl
6	Create unit tests	2	Front End/Backend	Cyrus

Completed:

After you register for an account, the obvious next step is to sign in. The sign-in form for the UI prompts a user for their username and password, and the backend does a check to make sure the user has a registered account with the provided password (with the /login route). Once signed in, the backend also creates an authentication token (self made), which the server uses to check to see if the user logs in. This token is used as a check in any function involving the user, and the token is returned as a header once the user successfully signs in. The unit tests check for possible issues, especially with the authentication token, and the tests all pass.

User Story #3
Estimated Hours: 7
As an existing user, I would like to be able to logout.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a navigational banner	3	Front End	Abigehl
2	Add a logout button to the navigational banner	1	Front End	Abigehl
3	Create API route to log out a user and clear tokens	1	Back End	Ken
4	Create unit tests	2	Front End/Back End	Cyrus

Completed:

After you login to an account, a user should be able to log out. On the navbar, there's a logout button that allows a user to click and log out of the account previously on. Once the logout button is pressed, the user is redirected back to the login page.

User Story #4
Estimated Hours: 9
As a user, I would like to be able to edit my email address.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Add an edit account button to users homepage	1	Front End	Ben
2	Create UI form to enter/show information about user	2	Front End	Ben
3	Create button to send information to Back End to register	1	Front End	Ben
4	Create API route to change a user's email address	3	Back End	Julien
5	Create unit tests	2	Front End/Back End	Cyrus

Completed:
After you have registered for an account and have signed in, you at some point may desire to change which e-mail address is associated with your account. In the navbar, when you click on your username, a drop down with options related to your account is displayed. Upon clicking the option to 'edit email address', a UI form pops up, with fields to input a new e-mail address. Once submitted, the backend updates the user's profile within the database, replacing their old e-mail address with the new one. Our unit tests model such a scenario, and pass.

User Story #5
Estimated Hours: 8
As a user, I would like my password to be reset if I forget it.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Add “Forgot password?” button to login form	1	Front End	Abigehl
2	Create forgot password popup with UI form for resetting password	2	Front End	Abigehl
3	Create API route to send a random 6 digit auth code to a specified user’s email address and submit a new password when the code is found in the request body	3	Backend	Julien
4	Create unit tests	2	Front End/Back End	Cyrus

Completed:
After you have registered for an account, you at some point may forget your password. On the sign-in form, there is a blue link that says ‘forgot password?’. Upon clicking this link, a UI form pops up, with fields to input the e-mail address associated with your account. Once submitted, the backend generates an e-mail with a 6 digit code that is sent to the address the user provided. This code serves as a temporary password. Our unit tests model such a scenario, and pass.

User Story #6
Estimated Hours: 6
As a user, I would like to be able to change my password.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Alter UI form to enter/show information, adding a password entity	1	Front End	Ben
2	Create API route to change a user's password	3	Back End	Julien
3	Create unit tests	2	Front End/Back End	Cyrus

Completed:

After you have registered for an account and have signed in, you at some point may desire to change your password. In the navbar, when you click on your username, a drop down with options related to your account is displayed. Upon clicking the option to 'change password', a UI form pops up, with fields to input a new password. Once submitted, the backend updates the user's profile within the database, replacing their old password with the new one. Our unit tests model such a scenario, and pass.

User Story #11
Estimated Hours: 19
As a user, I would like to be able to create a group and be granted group creator privileges

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI form to enter information about group	5	Front End	Ben
2	Create a Page to display Group	5	Front End	Ben
3	Create API route to create a group	3	Back End	Julien
4	Create API route to grant group creator privileges	2	Back End	Julien
5	Create a display on a user's profile page to reflect their privileges	2	Front End	Ben
6	Create unit tests	2	Front End/Back End	Cyrus

Completed:

As a user of the site, should a group not exist, one would want the ability to create it. From the page that displays groups, there is a button to create a new group. Upon clicking this button, a form pops up prompting the user for information about the group, such as its name, a description, and an image. After submitting this form the backend has a route to generate a new group using the given information as its metadata. The particular user that created the group is then identified on the group page as the founder, and is granted certain privileges. Our unit tests model this group creation process.

User Story #13
Estimated Hours: 10
As a general user, I would like to be able to report a group to the sitewide admins

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a UI button to report users	1	Front End	Abigehl
2	Create a UI form to report reason as to why a user is being reported	2	Front End	Abigehl
3	Create a confirmation dialog to prompt user to confirm report action	2	Front End	Abigehl
4	Create API route to report a group	3	Backend	Julien
5	Create unit tests	2	Front End/Back End	Cyrus

Completed:

As a user of the site, should a group exhibit inappropriate activity, one would desire the ability to report them to the moderation team. When navigating to a group's page, there can be found a button to report the group. Upon clicking the button, a form is displayed with fields to allow the user to describe their reason for reporting the group. There is then a confirmation pop up that follows. After confirming their report, a backend call is made to generate a report that will be visible to the moderation team. Our unit tests model this report procedure.

User Story #14
Estimated Hours: 12
As a user, I would like to be able to view all groups that I am a part of from my general profile page

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a portion of the User's profile page that reflects their involvement in groups	2	Front End	Abigehl
2	Create a group specific page for a user	4	Front End	Ben
3	Create API route to view all groups within general profile page	4	Backend	Julien
4	Create unit tests	2	Front End/Back End	Cyrus

Completed:

As an active user of the site, one would want to be able to view what trees they belong to. This information can be accessed via a user's profile. Upon navigating to one's profile page, on the right hand side a list is generated, containing any tree a user is involved in. This is achieved via a backend call that checks the database to retrieve all trees that the user is a member of.

What Didn’t Go Well

We assigned ourselves an excessive amount of user stories because we felt that our teams prior experience in building projects would allow us to get more done. In actuality, we ran into issues with stories which we thought would be simple which held us back from completing the extra stories we had added. By incorrectly estimating the time it would take to complete our stories, we had created a large body of work for ourselves which we only realized once we got further into the project.

User Story #7
Estimated Hours: 14
As a user, I would like to be able to create and edit a general profile page, with info such as my name, birth year, bio, links to social media, etc.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Alter UI form to enter/show information about user	2	Front End	Ben
2	Create API route to create a general profile page	3	Backend	Julien
3	Create API route to retrieve profile information	3	Backend	Julien
4	Create API route to update information	4	Backend	Julien
5	Create unit tests	2	Front End/Back End	Cyrus

Incomplete:

In this user story we were able to complete all of the individual components laid out in the breakdown of the story. The issue we ran into with this story is the communication between the front and backend. A user is able to edit their information from their basic profile page but they are unable to view any of the updated information. This is due to a bug in the communications between the front and backend. The webpage is supposed to receive all the details on the user when their profile page is accessed, and when you print out the response that the server-side is sending it prints out the correct response. This implies that there is an error in the way that the frontend section of the profile page, however, when using postman to show the communications between the front and back end, it shows that the message being received is only partial. This was realized last minute after hours of debugging and we were unable to find what is causing the disconnect between what is being printed out and what is being grabbed by the front end. As this is a vital piece of the program we will need to fix this at the beginning of the next sprint.

User Story #8
Estimated Hours: 8
As a user, I would like to be able to upload a profile image.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Add a UI button to upload a photo	2	Front End	Abigehl
2	Create Drag functionality for uploading picture	2	Front End	Abigehl
3	Create display for image on user's profile page	1	Front End	Abigehl
4	Create API route to add a photo to a user	2	Back End	Julien
5	Create unit tests	2	Front End/Back End	Cyrus

Incomplete:

For this story, we wanted to give users the ability to upload their own image of choice as their profile picture. Although we have image uploading functionally implemented for groups, we struggled to incorporate this for individual users. The problem lies within the fact that we wanted to host actual images instead of links to images. Group photo uploads are imageURLs, which we are able to store and display properly. We thought that this logic would easily transfer when implementing a file-upload button for user profile pictures, however this was not the case. We ran into many errors and put in hours and hours of debugging. The result is that we have a front-end button that upon clicking will initiate a prompt for file-uploading. Submission of a file does not actually store anything in the backend. However, if we hardcode an image as a user's profile picture, this can properly be displayed.

User Story #9
Estimated Hours: 11
As a user, I would like to be able to selectively hide or display specific info on my profile page.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Add drop down next to a text field to indicate if a user wants to hide or display information	2	Front End	Ben
2	Create API route to hide specific information on profile page	2	Back End	Julien
3	Create API route to display specific information on profile page	2	Back End	Julien
4	Create API route to retrieve profile information	3	Back End	Julien
5	Create unit tests	2	Front End/Back End	Cyrus

Incomplete:

Due to the issues in user story #7 this was also not completed. While the value for this can be updated in the database, the user is unable to view the update of information. This is due to an underlying error in communication between the back and front end. This is also a rather vital component of the project so this will also need to be fixed at the beginning of sprint two.

User Story #10
Estimated Hours: 17
As a user, I would like to be able to upload photos to my profile page's photo library.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Add a UI button to multiple upload photos	2	Front End	Abigehl
2	Create Drag functionality for uploading multiple images	3	Front End	Abigehl
3	Integrate UI button with backend implementation	3	Back End	Julien
4	Create API route to upload photos to a profile page's photo library	4	Back End	Julien
5	Create a display on the user's profile f1or photo library	3	Front End	Abigehl
6	Create unit tests	2	Front End/Back End	Cyrus

Incomplete:

For this story, we wanted to give users the ability to upload photos to their own personal photo library, and for them to be able to view those images on a page dedicated to this library. The way this would work, is that each user object would have an array of images associated with their account, and the photo library page would simply iterate through the array to display each photo on the page. An 'upload photo' button would be in place to take user input and add it to this array. Our app successfully retrieves a user's photo array, and even successfully displays photos from this array onto the site. However, we were unable to complete functionality for adding photos to the array. We attempted to implement two different approaches. Approach #1 was to have the photo array contain image urls, and upon clicking the button the user would be able to paste in a new url to add to the array. Approach #2 was to have the button actually initiate a file upload. While we were able to accomplish front-end functionality for both approaches, we were unsuccessful in getting the photos to actually be appended to the array.

User Story #12
Estimated Hours: 9
As a general user, I would like to be able to report users of a group to the group's moderation team

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a UI button to report users	1	Front end	Ben
2	Create a UI form to report reason as to why a user is being reported	2	Front End	Ben
3	Create a confirmation dialog to prompt user to confirm report action	1	Front End	Ben
4	Create API route to report users	3	Back End	Julien
5	Create unit tests	2	Front End/Back End	Cyrus

Incomplete:

For this story, we were unable to fully implement the functionality due to an issue retrieving the reported user's username. From a rooted page, the current user can see members of a group and click on their name to choose either to navigate to their profile or to report them. If the user chooses to report them, they should be redirected to the report page and can report the user. While our site can bring the user to the report page, we are unable to store the chosen username for the user being reported, this prevents the proper user from being reported. The backend for this system works properly, if you hardcode a username value then the user will be properly reported. Once we are able to determine which user's name has been clicked on and is supposed to be reported, this story will be completed.

User Story #15
Estimated Hours: 12
As a user, I would like to be able to view other user's profile pages.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a clickable link to a person's profile on group page	2	Front End	Ben
2	Create hover functionality for a person's name in a group	3	Front End	Abigehl
3	Create a non-user version of profile page	2	Front End	Ben
4	Create API route to view another user's profile pages	3	Back End	Julien
5	Create unit tests	2	Front End/Back End	Cyrus

Incomplete:

For this story, we wanted to give users the ability to navigate to another user's profile, in an effort to give them opportunities for networking. We successfully are able to generate profile pages for each user, and are also successfully able to populate a tree's page with its users and their usernames. However, due to time constraints we were unable to provide a link that navigates to a user's profile page upon clicking on their username on a tree page. Currently, clicking on another user's username will instead link to your own profile page. This is a relatively small error, and its cause is known. Essentially, because we are not generating a unique id for each profile in the backend, the route directs a user to their own profile page instead of the profile page of the user they clicked on. Because we chose to focus on other features, this went unresolved.

User Story #16
Estimated Hours: 13
As a user, I would like to be able to directly message other users.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	On non-user profile create a button to message user	1	Front End	Cyrus
2	Create a UI for hosting messaging	5	Front End	Ben
3	Create API route to send messages	5	Back End	Julien
4	Create unit tests	2	Front End/Back End	Cyrus

Incomplete:

Due to issues in other parts of our project we were unable to get to the frontend portion of this user story. The backend of this story should be already completely implemented, we just need to create a frontend and intermediate layer that connects the two. This is the only user story that we did not complete due to lack of time. The other issues we encountered were due to unexpected, difficult to find bugs.

How Should We Improve?

In the future we need to better communicate solid deadlines for steps. In addition we need to schedule ourselves more time to deal with debugging the user stories we complete, as well as more time to complete the user stories as we severely underestimated the length of time each user story would take. We need to communicate our bugs more effectively and work as a team to correct the bugs. If we are able to improve both our team coordination and communication, it should allow us to work much more efficiently on what needs to be done and not have to run too close to deadlines. By better structuring our sprint and understanding exactly what needs to be done each week, we should be in a much better spot moving forward.