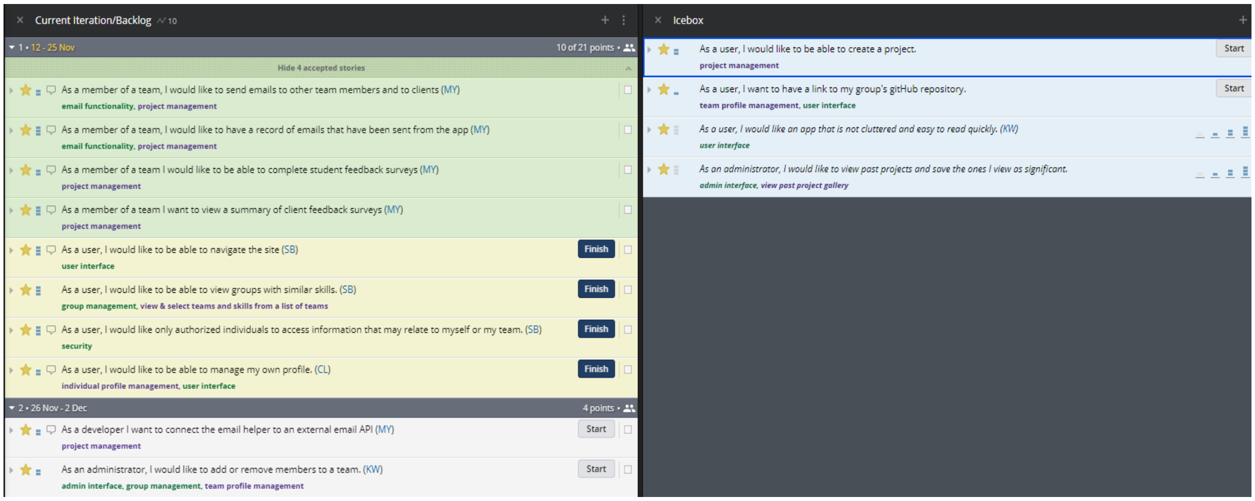
Sprint 2 – End of Sprint Deliverables

Group C – Project #3

User Stories Planned vs Implemented

Pivotal ID	Description	Implemented?
#161696993	Send email to client or others	Yes
#161703978	View previously-sent emails for a project	Yes
#161868036	Connect email functionality to an external email API	No
#161683917	Site navigation	Yes
#161667403	View teams and skills	Yes
#161672173	Login page and authorization	Yes
#161867769	Submit student feedback surveys	Yes
#161867775	View client feedback surveys	Yes
#161696996	Add and remove team members	Yes
#161706502	Individual profile management	Yes

Pivotal Tracker



As a member of a team, I would like to send emails to other team members and to clients (MY)
email functionality, project management

As a member of a team, I would like to have a record of emails that have been sent from the app (MY)
email functionality, project management

As a member of a team I would like to be able to complete student feedback surveys (MY)
project management

As a member of a team I want to view a summary of client feedback surveys (MY)
project management

As a user, I would like to be able to navigate the site (SB)
user interface

As a user, I would like to be able to view groups with similar skills. (SB)
group management, view & select teams and skills from a list of teams

As a user, I would like only authorized individuals to access information that may relate to myself or my team. (SB)
security

As a user, I would like to be able to manage my own profile. (CL)
individual profile management, user interface

As a developer I want to connect the email helper to an external email API (MY)
project management

As an administrator, I would like to add or remove members to a team. (KW)
admin interface, group management, team profile management

As a user, I would like to be able to create a project.
project management

As a user, I want to have a link to my group's gitHub repository.
team profile management, user interface

As a user, I would like an app that is not cluttered and easy to read quickly. (KW)
user interface

As an administrator, I would like to view past projects and save the ones I view as significant.
admin interface, view past project gallery

Individual Contributions - Steven Ballios (Product Manager)

https://github.com/kaboom18 kaboom18

https://www.pivotaltracker.com/n/projects/2209974

- **Description of Work Completed:**
    - **list of features implemented (completion status) or % of contribution (Identify User Stories by name, Pivotal Tracker ID#, and GitHub Branch)**

      Navigation #161683917, Master

      view skills and teams, #161667403, view-team-skills

      authorization, #161672173, private

    - **Description of testing completed**

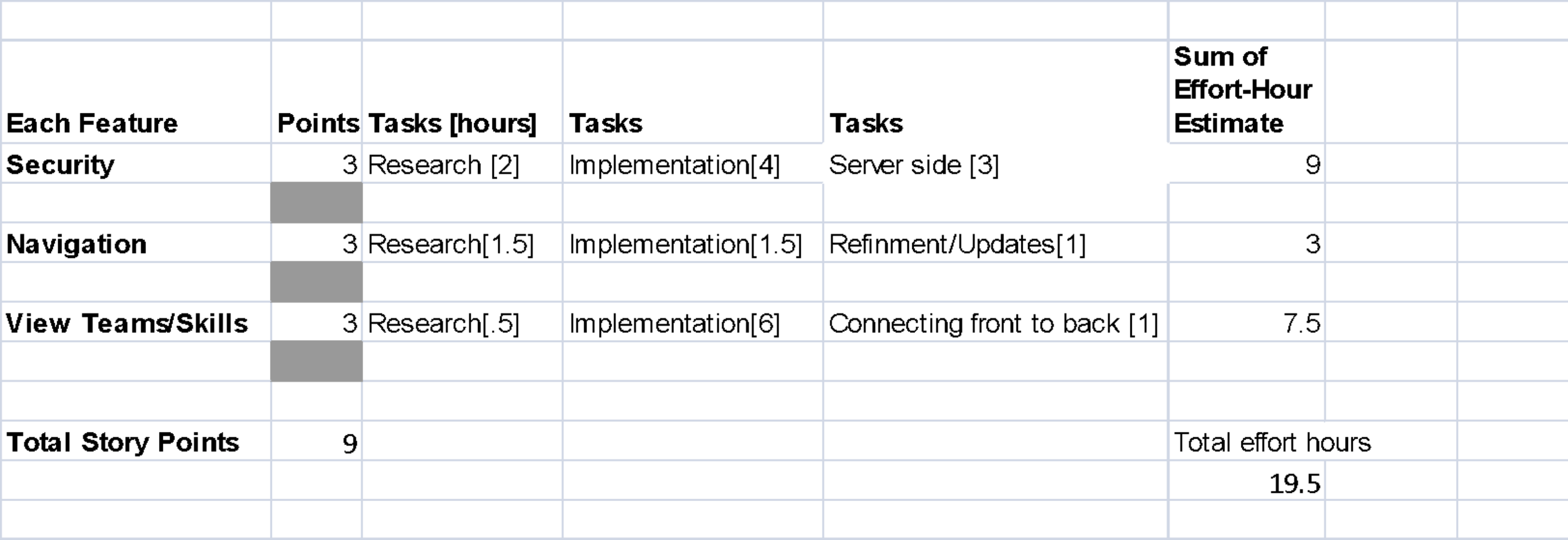
      -I have added basic navigation on the client

      -Made login page and is functioning with server side validation

      -Teams skills page proof of concept complete

      -Created the mongo database, Created user model and added test data

    - **Include Individual Taskboard & burndown chart**



Each Feature	**Points**	**Tasks [hours]**	**Tasks**	**Tasks**	**Sum of Effort-Hour Estimate**
Security	3	Research [2]	Implementation[4]	Server side [3]	9
Navigation	3	Research[1.5]	Implementation[1.5]	Refinment/Updates[1]	3
View Teams/Skills	3	Research[.5]	Implementation[6]	Connecting front to back [1]	7.5
Total Story Points	9				Total effort hours
					19.5

- **Other information: This is your opportunity to document whatever you have done for credit**
  - Managing product
  - 4 User Stories written/modified
  - Balsamiq account created, wireframes created
  - some team documents /scheduling meetings

- **Brief description tutorials completed, article read, or resources used**
  - React.js documentation

Individual Contributions - Craig Lange (Dev Team Member)

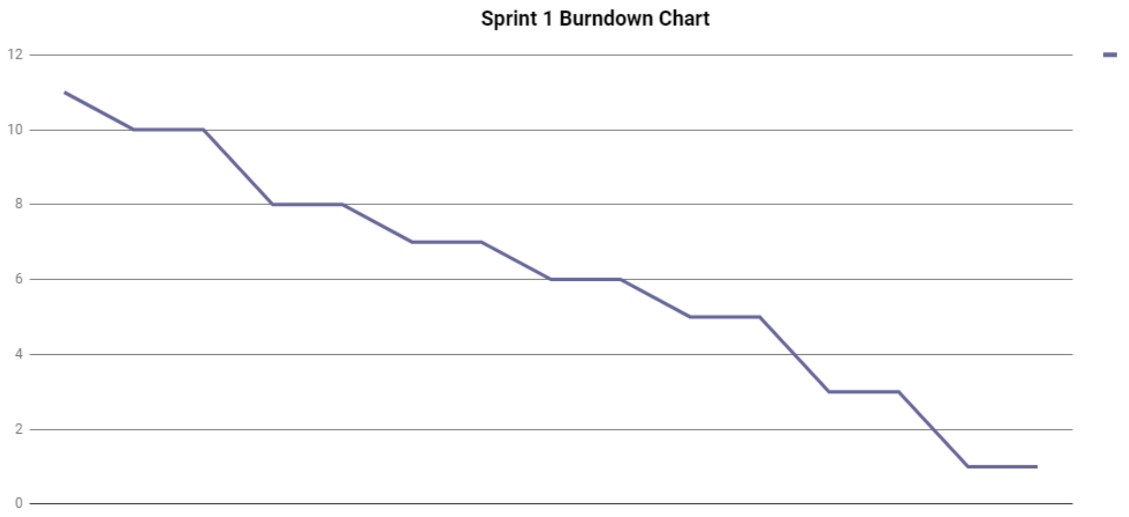
User stories written: #161706502
User stories completed/contributed:
#161706502 - “As a user, I would like to be able to manage my own profile” - Github branch IndivProfile (now merged with master). Story is ⅔ complete. Created the individual user page, as well as the create new user page, and the backend functions and routing for both. Tested layout using the web browser and tested the APIs using calls from Postman. Created wireframes for both pages.
#161683917 - “As a user, I would like to be able to navigate the site” - Small additions made were part of branch IndivProfile. Helped refine layout and file structure for the front end to allow easier addition of new pages/features and more streamlined navigation.
Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2209974
GitHub ID: clange9530
Team GitHub page: https://github.com/UFO-CEN-Group-C/student-portal-app

Tutorials Completed/Articles Read: https://scotch.io/tutorials/how-to-use-the-javascript-fetch-api-to-get-data
https://medium.com/@pshrmn/a-simple-react-router-v4-tutorial-7f23ff27adf
https://reacttraining.com/react-router/web/api/Route/component
The above articles covered how to handle calls to an external API in react, as well as creating routes and links to other pages. They were all very helpful and I used what I learned to handle fetching data from the database to display on the user page, as well as saving data to the database from the create new user page.

Assistance: Helped to refine overall layout/file structure of front end after SB created the initial structure.

Summary: Researched routing and using the fetch API with React. Helped create navbar and layout/navigation of front end. Created individual user page and create new user page.Created backend functions and express routing for both.

Sprint 1 Backlog			Sprint 1 (2 Weeks)													
Product Backlog Item ID	User Stories (Features)	**Initial Estimate**	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
#161706502	Create indiv. User page	8	7	7	6	6	5	5	4	4	3	3	2	2	1	1
#161706502	Create new user page	2	2	2	2	2	2	2	2	2	2	2	1	1	0	0
#161683917	Refine navigation, folder structure	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	Total Effort	11	10	10	8	8	7	7	6	6	5	5	3	3	1	1



Note: Burndown chart includes features planned from original backlog. Recent feedback from the client means that there is much more refinement and work to do on the user pages.

Individual Contributions – Kenneth Wilkinson (Scrum Master)

GitHub: https://github.com/UFO-CEN-Group-C/student-portal-app

GitHub ID: kwilkinson1

Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2209974

User Stories Written: #162074492, #162074461, #162074450, #162074440, #162074427

User Stories Contributed: #162074461 (60%), #161696996 (70%), #161697014 (50%), #161696996 (70%)

Testing: Testing was accomplished both by use of postman for router functionality and google chrome web developer tools for front end.

Tutorials Completed:

1. Express Essential Training: https://www.lynda.com/Node-js-tutorials/Express-EssentialTraining/679637-2.html

    a. Used to understand basic functionality of Express router commands

2. Designing RESTful APIs: https://www.lynda.com/Web-Development-tutorials/DesigningRESTful-APIs/642497-2.html

    a. This tutorial helped me understand how to organize a scalable API to use with the group

3. GIT Essential Training: https://www.lynda.com/Git-tutorials/Git-EssentialTraining/100222-2.html

    a. Prior to this semester, I had not used Git, so this helped me understand how to merge, branch, fetch, and pull.

    b. This tutorial also helped me understand how to comment appropriately.

4. Building a REST API with Node.js: https://www.youtube.com/watch?v=0oXYLzuucwE&list=PL55RiY5tL51q4DB63KBnygU6opNPFk_q

    a. This tutorial was very helpful in understanding the basics of a REST API. I think this really hammered home how to get the back end functionality working properly. Personally, I think these videos would help others in class understand scalable applications.

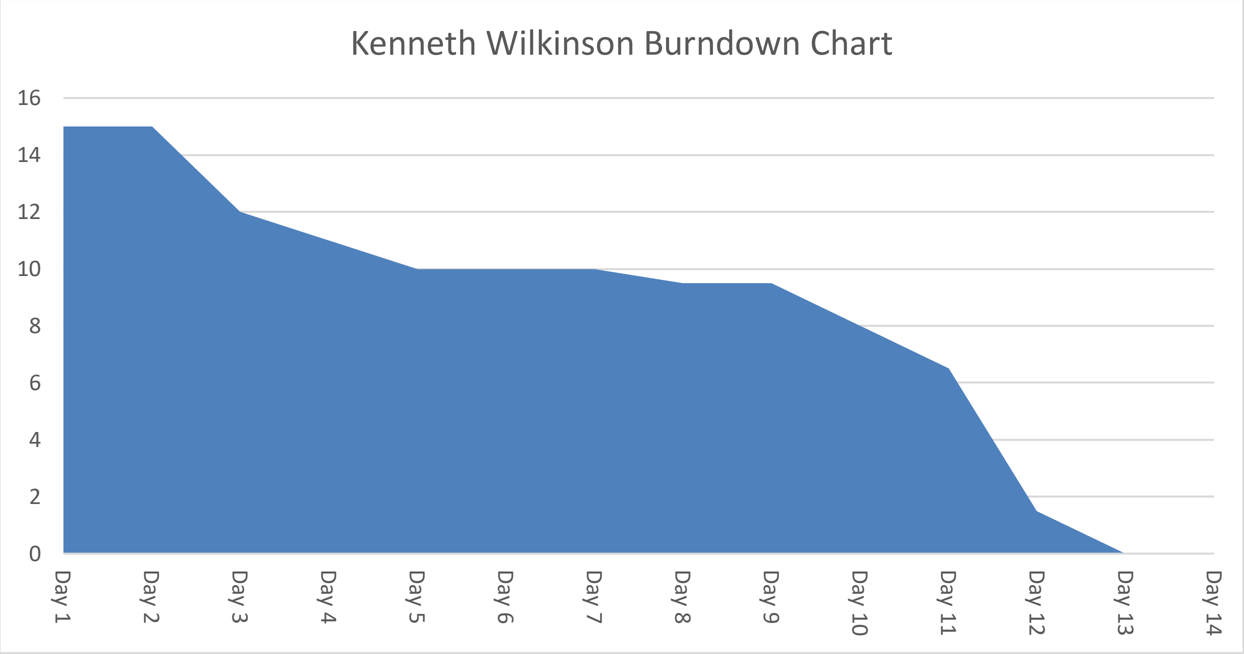
5. AWS S3 with Node.js: https://www.youtube.com/watch?v=joXy_OTCO_E

    a. This video was helpful in understanding how to set up Amazon S3 bucket for file sharing.

Assistance: Corrected issue with navbar from merge conflict.

Summary: For this sprint, I worked on both tidying up the file structure and viewing the team profile page. I have also begun research into using Amazon's S3 bucket service to upload and download images for user profile pictures and team logos. I added some functionality to only update those fields which are changed, and also prevented addition of fields which are not defined in the schema. I also ensured that the team profiles are stored properly in the database.

User Story	Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
As an admin, I would like to view team profile information	Design: Add team router functionality	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	Create team schema	0	0	1	0	0	0	0	0	0	0	0	0	0.5	0
	Create team profile page	0	0	2	0	0	0	0	0	0	0	0	0	1	0
	Format router and add controller file	0	0	0	0	0	0	0	0	0	0	0	2	0	0
As a user, I would like to upload / edit a profile picture	Research Amazon Web Services	0	0	0	1	1	0	0	0	0	0	0	0	0	0
As an administrator, I would like to search for a specific team profile.	Fix router	0	0	0	0	0	0	0	0	0	0	2	0	0	0
	Fix schema (add _id field generated by mongoose)	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0
As an administrator, I would like to add / remove members from a team.	Add patch functionality to router	0	0	0	0	0	0	0	0	0	0	0	2	0	0
	Research how to limit updates based on schema and prevent corruption	0	0	0	0	0	0	0	0	0	0	0	1	0	0
As a user, I would like to navigate the site easily	Fixed navbar error	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0
Subtotal		**0.00**	**0.00**	**3.00**	**1.00**	**1.00**	**0.00**	**0.00**	**0.50**	**0.00**	**1.00**	**2.50**	**5.00**	**1.50**	**0.00**



Individual Contributions – Max Yeste (Dev Team)

GitHub ID: mrmvy
Team GitHub: https://github.com/UFO-CEN-Group-C/student-portal-app
Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2209974

Work Completed

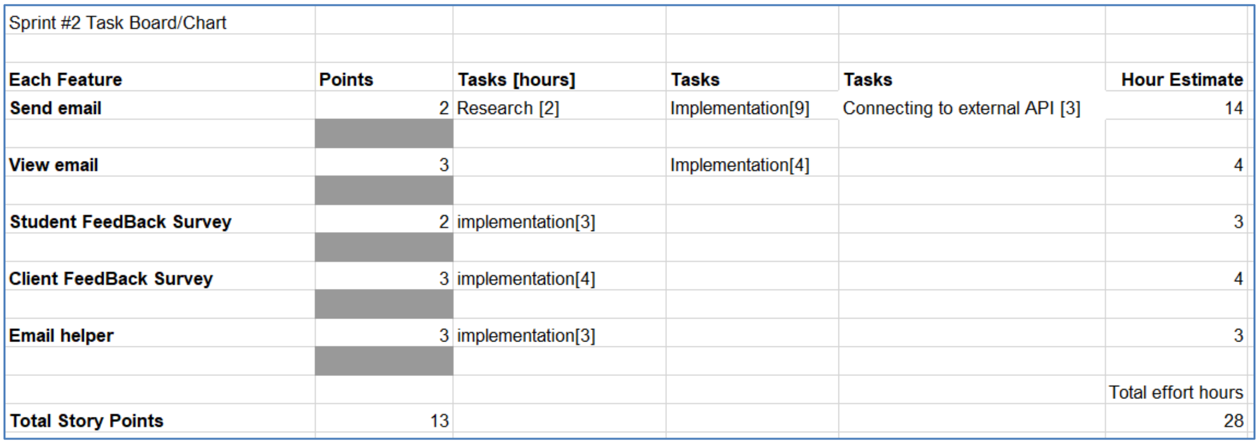
Features Implemented

- Send Email (completed)
  - Pivotal Tracker ID: #161696993
  - GitHub Pull Request: https://github.com/UFO-CEN-Group-C/student-portal-app/pull/4
- View previously-sent emails (completed)
  - Pivotal Tracker ID: #161703978
  - GitHub Pull Request: https://github.com/UFO-CEN-Group-C/student-portal-app/pull/7
- Submit Student Survey (completed)
  - Pivotal Tracker ID: #161867769
  - GitHub Pull Request: https://github.com/UFO-CEN-Group-C/student-portal-app/pull/8
- View Client Survey (completed)
  - Pivotal Tracker ID: #161867775
  - GitHub Pull Request: https://github.com/UFO-CEN-Group-C/student-portal-app/pull/9
- Connect Email Functionality to External API (attempted, not completed)
  - Pivotal Tracker ID: #161868036
  - GitHub Pull Request: N/A

Testing Completed

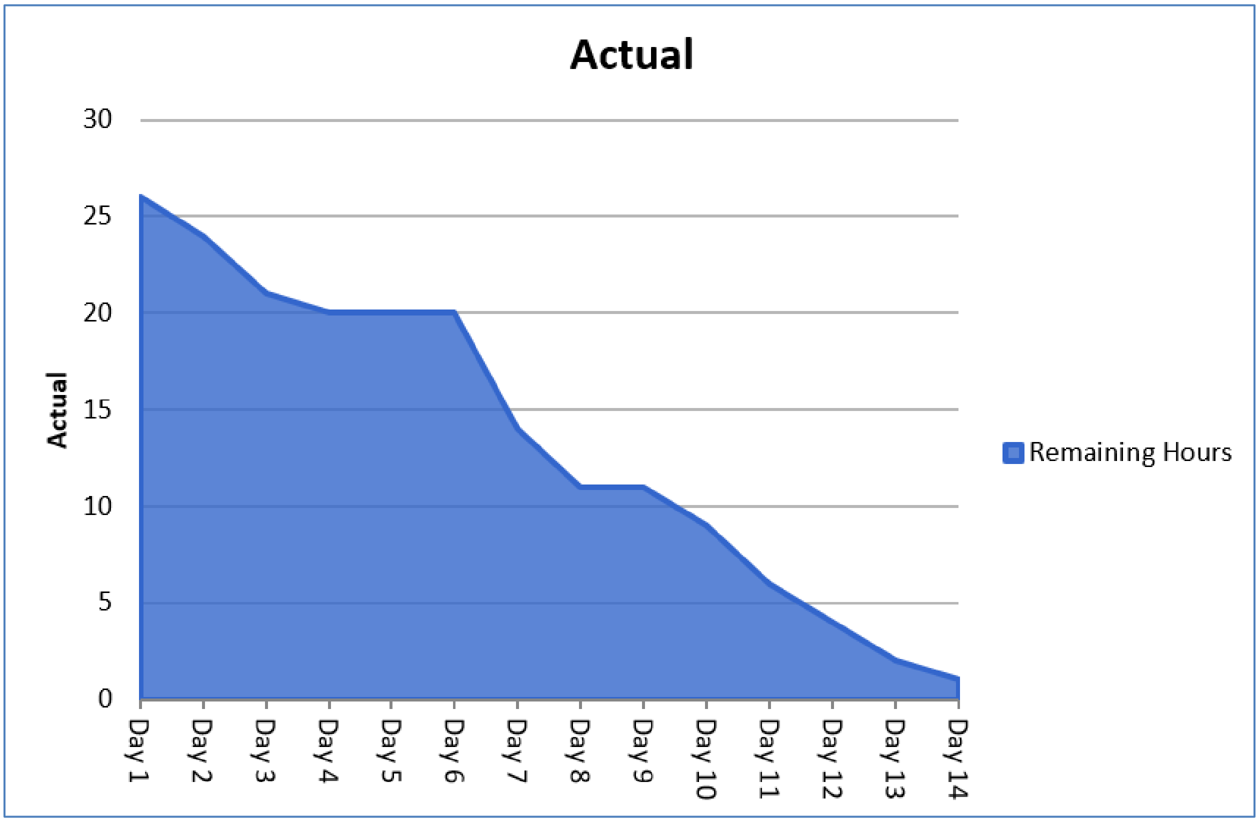
- For the four completed features above, I manually tested the front-end using a web browser. Testing included interacting with the application as well as inspecting the state of the application at different points in time by using console.log to write information to the console in the browser's developer tools.
- I tested the back-end API for each of the four features by submitting requests to the API using Postman.

Individual Taskboard



Sprint #2 Task Board/Chart					
Each Feature	**Points**	**Tasks [hours]**	**Tasks**	**Tasks**	**Hour Estimate**
Send email	2	Research [2]	Implementation[9]	Connecting to external API [3]	14
View email	3		Implementation[4]		4
Student FeedBack Survey	2	implementation[3]			3
Client FeedBack Survey	3	implementation[4]			4
Email helper	3	implementation[3]			3
					Total effort hours
Total Story Points	13				28

Individual Burndown Chart



Other Information

User Stories written

- View previously-sent emails
  - Pivotal Tracker ID: #161703978
- Submit Student Survey
  - Pivotal Tracker ID: #161867769
- View Client Survey
  - Pivotal Tracker ID: #161867775
- Connect Email Functionality to External API
  - Pivotal Tracker ID: #161868036

User Stories Modified

- Send Email (originally created by Ken, but extensively modified by me)
  - Pivotal Tracker ID: #161696993

Epics Written

- Individual Profile Management
  - Pivotal Tracker ID: ##4142290
- Project Management
  - Pivotal Tracker ID: ##4142287
- Team Profile Management
  - Pivotal Tracker ID: ##4142284
- View & Select teams and skills from a list of teams
  - Pivotal Tracker ID: ##4147487
- View Past Project Gallery
  - Pivotal Tracker ID: ##4147486

Wireframes Created

- Project Email Functionality
  - Balsamiq Link: https://balsamiq.cloud/sxfcogu/pg8qwoq/r2A5D
- Survey Functionality (Submit Student Survey, View Client Survey)
  - Balsamiq Link: https://balsamiq.cloud/sxfcogu/pg8qwoq/r45CF

Database Schema

Created schema for the following tables/MongoDB collections:

- clientsurveyresponses
- emails
- studentsurveyresponses
- studentsurveys

Documentation/Presentations

- Created PowerPoint slide deck for Sprint 2 End of Sprint Demo

Tutorials/Articles/Resources

Making dynamic form inputs with React

- **URL:** https://goshakkk.name/array-form-inputs/
- **Description:** For the student and client surveys I needed to be able to add a series of controls (labels and textboxes in this case) to the web page, but the questions for the survey are stored in the MongoDB database which means that at design time I would not know the number of controls needed or the specific text for each label. Also, I needed to bind these controls to an array that is stored in the React component's state but, as with the controls, I do not know the number of elements in the array at design time. Dynamically adding the controls to the web page was not difficult as this is similar to work we had done in the Bootcamp assignments with adding buildings to a grid for the UF Directory app. However, I did not know how to ensure that each of the dynamically created controls (each one is an HTML input tag) was properly bound to the correct array element. Although my final code is set up a bit differently than the sample code on this web page, it helped me understand how to handle binding the dynamically created controls to the array that was created at runtime.
- **Recommendation:** I recommend this resource for other students with similar needs.

Create React App with an Express Backend

- **URL:** https://daveceddia.com/create-react-app-express-backend/
- **Description:** It was not immediately clear to me how to set up a React front-end to work with an Express back-end API. In our Bootcamp assignments we had a React application with no back-end, an Express back-end with no front-end, and an Express back-end with an Angular front-end. This tutorial helped me understand how to use the React componentDidMount lifecycle method together with fetch() to call the back-end API and process the returned data. I used this in all of the features that I worked on.
- **Recommendation:** I recommend this resource for other students with similar needs.

Email Services

I explored several email services and encountered challenges with each of them.

SendPulse (https://sendpulse.com/)

This initially seemed to be the most promising of email services that I found by performing a Google search. This service offers a free account that allows up to 12,000 emails to be sent each month through their SMTP service. The SMTP service also offers a REST API and a Node.js library that can be used to send interact with the REST API. Initially my account with SendPulse was disabled without explanation. An inquiry to their support team revealed (after waiting a couple of days) that this was because I had used a disposable email address (https://sneakemail.com) with my account (I generally use disposable email addresses as a spam-management technique). I subsequently created a new account using a regular email address that I created for this purpose. However, I then found that SendPulse's API documentation was sorely lacking. It took quite a bit of experimentation with Postman to figure out how to get an authentication token from the API. Once I finally did that, attempts to send an email message using the API resulted in an error with the code "8" and the message "no data". SendPulse did not provide any explanation for this error and some Google searching was also not fruitful. I then tried just sending an email directly from our Express back-end application using SendPulse's Node.js library. However, once I got this code mostly working (again, challenging due to lack of good documentation from SendPulse), the SendPulse API returned an error that said, "554 5.9.2 Sender domain is not valid." This error occurred despite the fact that I had copied my code, including the sample message JSON, from an example supplied by SendPulse and also despite the fact that my sample message included a sender with a valid domain. **Recommendation:** At this time I cannot recommend SendPulse to others.

Mailjet (https://www.mailjet.com)

Mailjet also initially seemed promising with two different REST APIs available for sending email. However, as I studied the API documentation for Mailjet I discovered that they do not allow arbitrary senders for emails sent through their API. Instead, each sender must be pre-defined and set up in the Mailjet account. This makes it impractical for our app as we would not know who the email sender is until a user creates an email message. **Recommendation:** At this time I cannot recommend Mailjet to others.

Mailgun (https://www.mailgun.com)

Mailgun is another service that offers a free, entry-level service and a REST API. However, Mailgun requires that you pre-define senders for emails that you will be sending (although they do not require that the "From" that the recipient sees is the same as the pre-defined sender). However, Mailgun does require that you are able to make DNS entries for the sender's domain, making it impractical for our project. **Recommendation:** At this time I cannot recommend Mailgun to others.

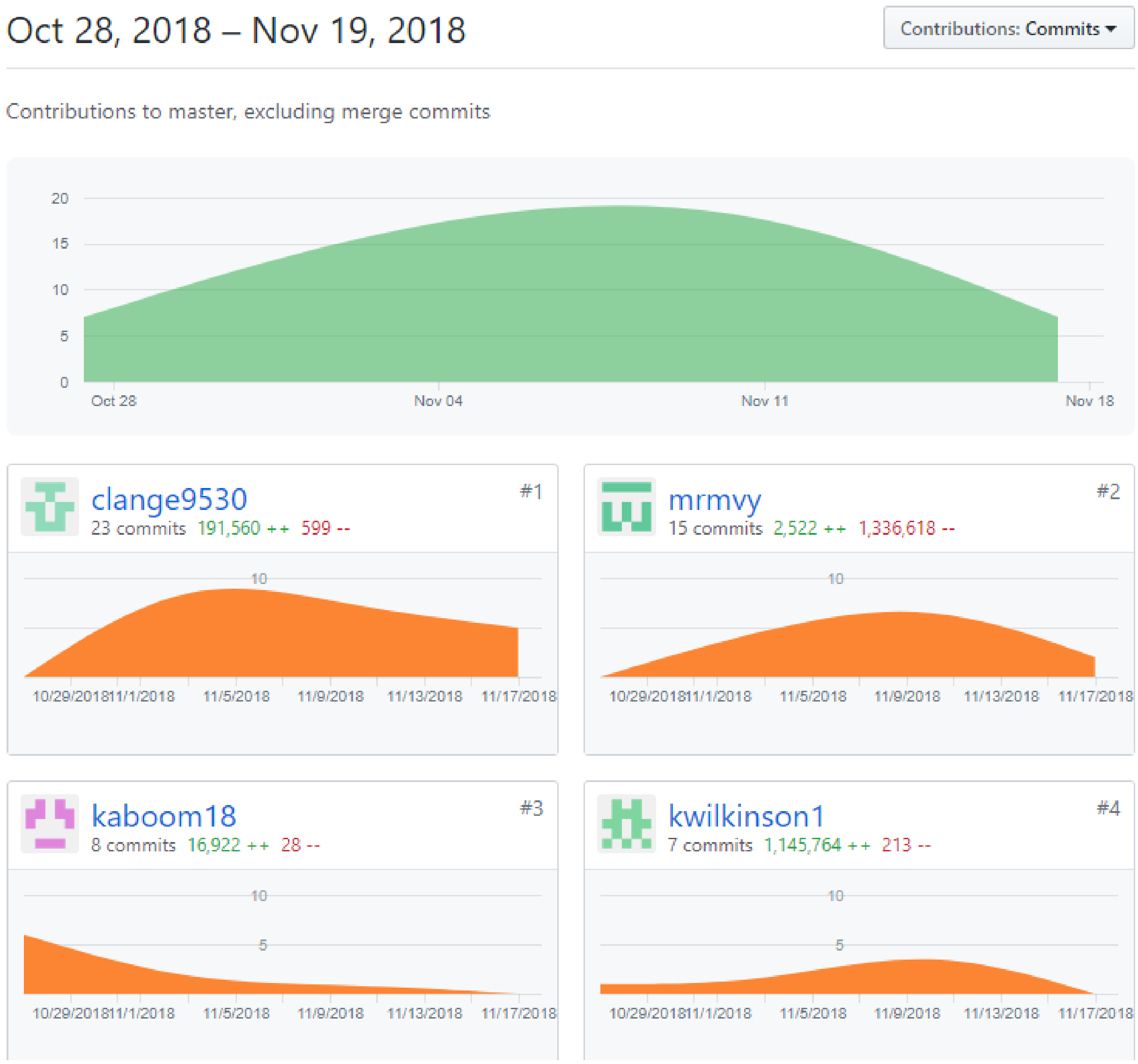
SocketLabs (https://www.socketlabs.com/)

SocketLabs is yet another company that offers a free, entry-level email service and a REST API. However, SocketLabs requires you to specify a company name and website URL during account creation. I tried using University of Florida and www.ufl.edu for these two items but my account was disabled the day after I created it. Since I did not have better options for a company name or website, I did not make any further attempts with SocketLabs. **Recommendation:** At this time I cannot recommend SocketLabs to others.

Amazon Simple Email Service (https://aws.amazon.com/ses/)

Unsurprisingly, Amazon also offers email service as part of their AWS suite of services. They also have special student-oriented accounts that offer free usage of many AWS services. However, like some of the other available services I found, Amazon Simple Email Service requires you to pre-define and verify an email sender and/or domain. **Recommendation:** At this time I cannot recommend Amazon SES to others.

GitHub Contributions



Oct 28, 2018 – Nov 19, 2018

Contributions: Commits ▾

Contributions to master, excluding merge commits

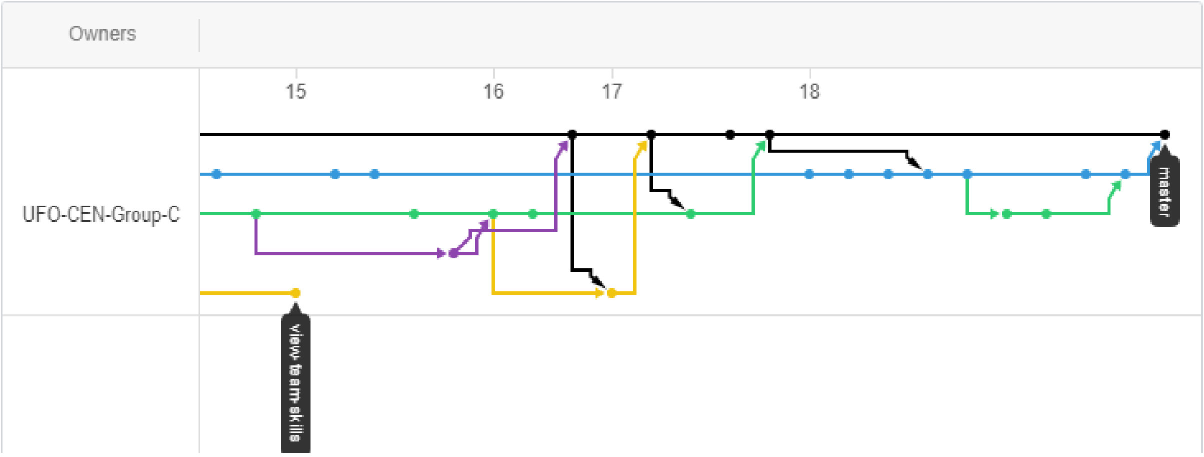
clange9530 #1
23 commits 191,560 ++ 599 --

mrmvy #2
15 commits 2,522 ++ 1,336,618 --

kaboom18 #3
8 commits 16,922 ++ 28 --

kwilkinson1 #4
7 commits 1,145,764 ++ 213 --

GitHub Network Chart



Group Confirmation

All members of Team #, as indicated by our initials at the bottom of the statement of work, agree with the work described and contribution of each team member. Any discrepancies have been discussed and revised in this document.

SB CL

MY KW