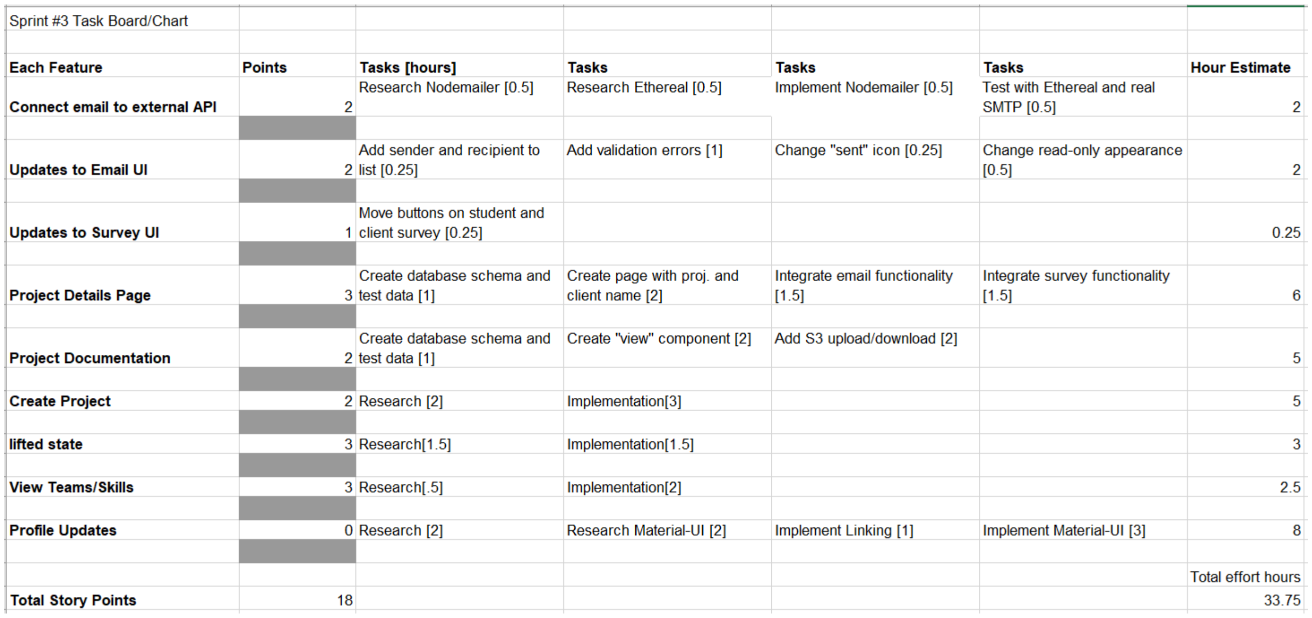
Sprint 3 – Work Distribution

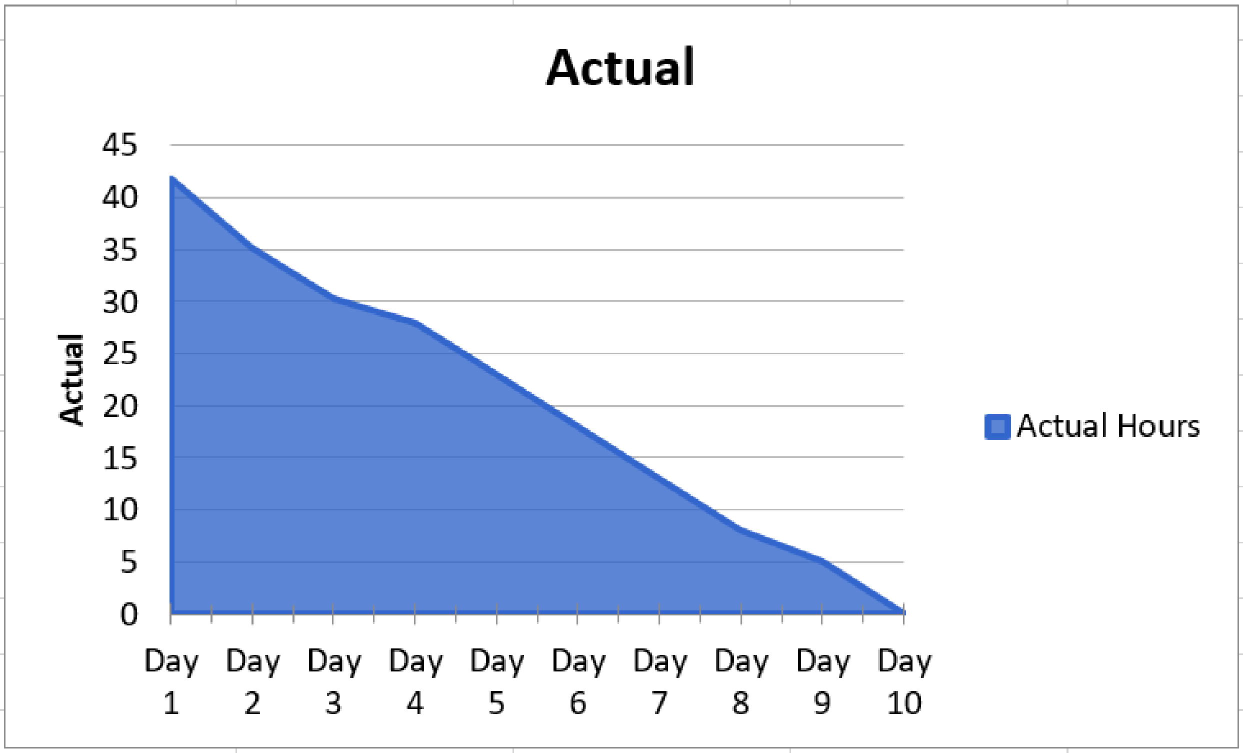
Group C – Project #3

Group Task Board



Sprint #3 Task Board/Chart						
Each Feature	**Points**	**Tasks [hours]**	**Tasks**	**Tasks**	**Tasks**	**Hour Estimate**
Connect email to external API	2	Research Nodemailer [0.5]	Research Ethereal [0.5]	Implement Nodemailer [0.5]	Test with Ethereal and real SMTP [0.5]	2
Updates to Email UI	2	Add sender and recipient to list [0.25]	Add validation errors [1]	Change "sent" icon [0.25]	Change read-only appearance [0.5]	2
Updates to Survey UI	1	Move buttons on student and client survey [0.25]				0.25
Project Details Page	3	Create database schema and test data [1]	Create page with proj. and client name [2]	Integrate email functionality [1.5]	Integrate survey functionality [1.5]	6
Project Documentation	2	Create database schema and test data [1]	Create "view" component [2]	Add S3 upload/download [2]		5
Create Project	2	Research [2]	Implementation[3]			5
lifted state	3	Research[1.5]	Implementation[1.5]			3
View Teams/Skills	3	Research[.5]	Implementation[2]			2.5
Profile Updates	0	Research [2]	Research Material-UI [2]	Implement Linking [1]	Implement Material-UI [3]	8
						Total effort hours
Total Story Points	18					33.75

Group Burndown Chart



Individual Contributions - Steven Ballios (Product Manager)

Please see Steven’s submission for the **Sprint #3 – Individual Contributions** assignment.

Individual Contributions - Craig Lange (Dev Team Member)

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 profile-updates, last-minute, and final-ui-updates (now merged with master). Integrated Materal-UI into Individual User and Create/Edit User page. Linked User Page to Team Page. Tested layout using the web browser and tested the APIs using calls from Postman as well as form data from the browser..

#161683917 - “As a user, I would like to be able to navigate the site” - Integrated Material-UI into the NavBar to help tie the aesthetic of the app together.

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

https://github.com/clange9530/student-portal-app

Branches: profile-updates, last-minute, and final-ui-updates.

Tutorials Completed/Articles Read:

https://material-ui.com/

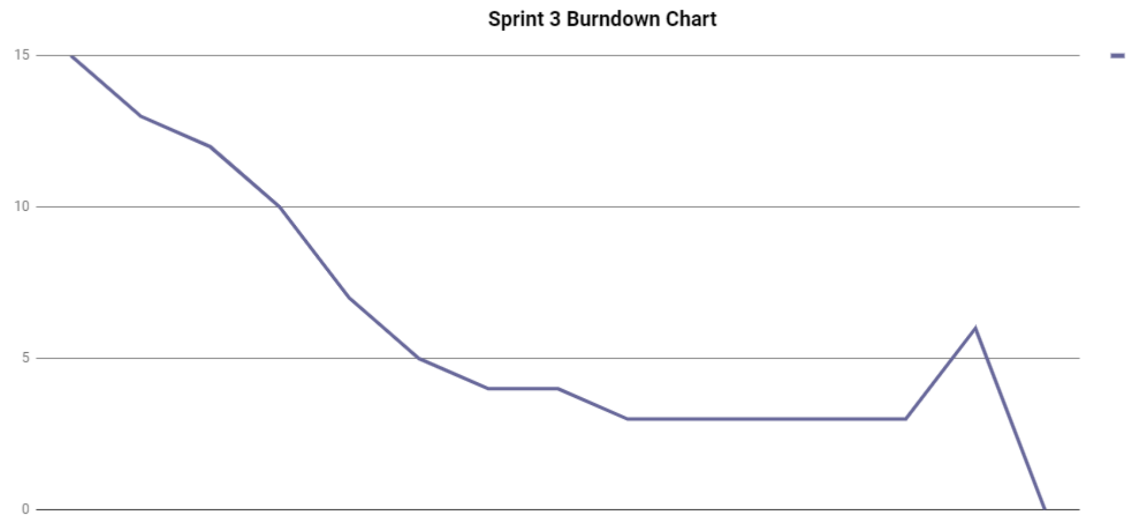
Read a lot of the documentation and looked at a lot of the component demos on the Material-UI website.

Assistance: Helped with the styling of the Teams page and making the routing for it dynamic.

Summary: Continued working on the pages from the last sprint, tying in Material-UI into all of the components that I worked on to bring the app’s visual theme together. Fixed bugs and functionality issues on the User and CreateProfile pages.

Sprint 3 Backlog			Sprint 3													
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	Improve Individual User Page	6	4	3	1	1	1	1	1	1	1	1	1	1	1	0
#161706502	Integrate User Page with Edit User Page	3	3	3	3	3	1	0	0	0	0	0	0	0	0	0
#161706502	Integrate User Page with Team Page	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0
#161706502	Fix bugs on Edit/Create User Page	3	3	3	3	3	3	3	3	2	2	2	2	2	2	0
#161683917	Integrate Material-UI into NavBar - Not originally planned	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
	Total Effort	15	13	12	10	7	5	4	4	3	3	3	3	3	6	0

Sprint Burndown Chart


Sprint 3 Burndown Chart
15
10
5
0

Individual Contributions – Kenneth Wilkinson (Scrum Master)

GitHubID: kwilkinson1

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

Team GitHub: https://github.com/clange9530/student-portal-app

User Stories Contributed/completed:

- "As a user, I would like to add a profile picture." (Completed)
    - Pivotal Tracker ID: #162403656
    - GitHub Pull Request:
    - Added back end router for aws functionality
        - I personally had several issues getting response from router. At first, I believed it to be an issue with the permissions for the bucket, but was unable to determine what the issue could be. I was unable to even elicit an error message. Craig helped by testing the code I had written, and was able to utilize it on his local repository.
    - Utilized middleware to dynamically upload user profile picture to aws
        - As I was still unable to elicit an error message/response from my router on my computer, I eventually researched more and found react-s3, a middleware that performs upload/delete functionality for aws s3.
    - Created coding to dynamically update profile picture on user page, and update "ProfilePicURL" property to reflect URL from aws s3 bucket.
    - Performed styling to make HTML input component analogous to other components on user page

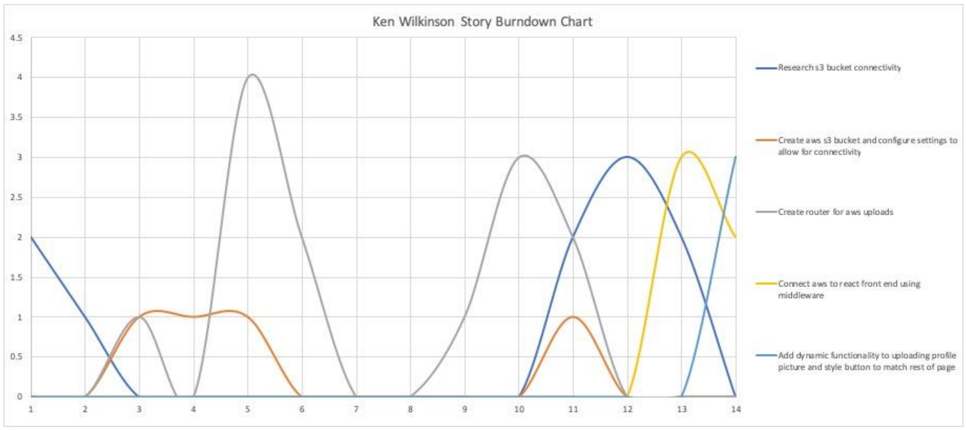
Testing Completed:

- All testing for router functionality was performed via PostMan
- All testing for front end functionality was performed via manual means (utilizing components and google web developer tools)

Tutorials Completed/Read:

- **Simple Image Upload with Node on Amazon S3** (https://www.youtube.com/watch?v=ASuU4km3VHE)
    - This tutorial helped to understand the best way to use multer and aws-sdk to create a router to post files to Amazon's aws
- **Upload Files from React to AWS S3 in 6 Minutes (https://www.youtube.com/watch?v=fXjU8jTjHB4)**
    - This tutorial demonstrated the middleware I eventually used to perform the s3 uploads. However, the material is dated and react-s3 must be used with new function calls.
- **NPM React-S3** (https://www.npmjs.com/package/react-s3)
    - This is the documentation for React-S3, which allows for easy uploads and deletes to aws buckets.

Individual Burndown Chart



Individual Taskboard

Sprint 3 Backlog		Sprint 3													
Backlog Item ID	Description	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
162403656	Research s3 bucket connectivity	2	1	0	0	0	0	0	0	0	0	2	3	2	0
162403656	Create aws s3 bucket and configure settings to allow for connectivity	0	0	1	1	1	0	0	0	0	0	1	0	0	0
162403656	Create router for aws uploads	0	0	1	0	4	2	0	0	1	3	2	0	0	0
162403656	Connect aws to react front end using middleware	0	0	0	0	0	0	0	0	0	0	0	0	3	2
162403656	Add dynamic functionality to uploading profile picture and style button to match rest of page	0	0	0	0	0	0	0	0	0	0	0	0	0	3
	Total	33	32	30	29	24	22	22	22	21	18	13	10	5	0

Individual Contributions – Max Yeste (Dev Team)

GitHub ID: mrmvy
Team GitHub: https://github.com/clange9530/student-portal-app
Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2209974

Work Completed

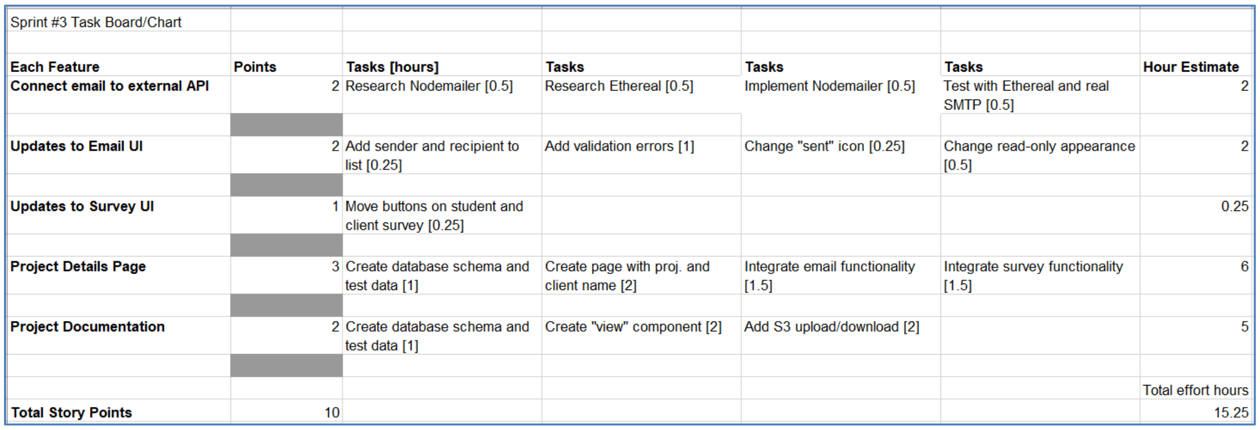
Features Implemented

- Connect email to external API (completed)
    - Pivotal Tracker ID: #161868036
    - GitHub Pull Request: https://github.com/UFO-CEN-Group-C/student-portal-app/pull/11
- Updates to Email UI
    - Pivotal Tracker ID: #162174014
    - GitHub Pull Request: https://github.com/UFO-CEN-Group-C/student-portal-app/pull/13
- Updates to Survey UI
    - Pivotal Tracker ID: #162174181
    - GitHub Pull Request: https://github.com/UFO-CEN-Group-C/student-portal-app/pull/14
- Project Details Page
    - Pivotal Tracker ID: #162179725
    - GitHub Pull Request: https://github.com/UFO-CEN-Group-C/student-portal-app/pull/15
- Project Documentation
    - Pivotal Tracker ID: #162179729
    - GitHub Pull Request: https://github.com/UFO-CEN-Group-C/student-portal-app/pull/18

Testing Completed

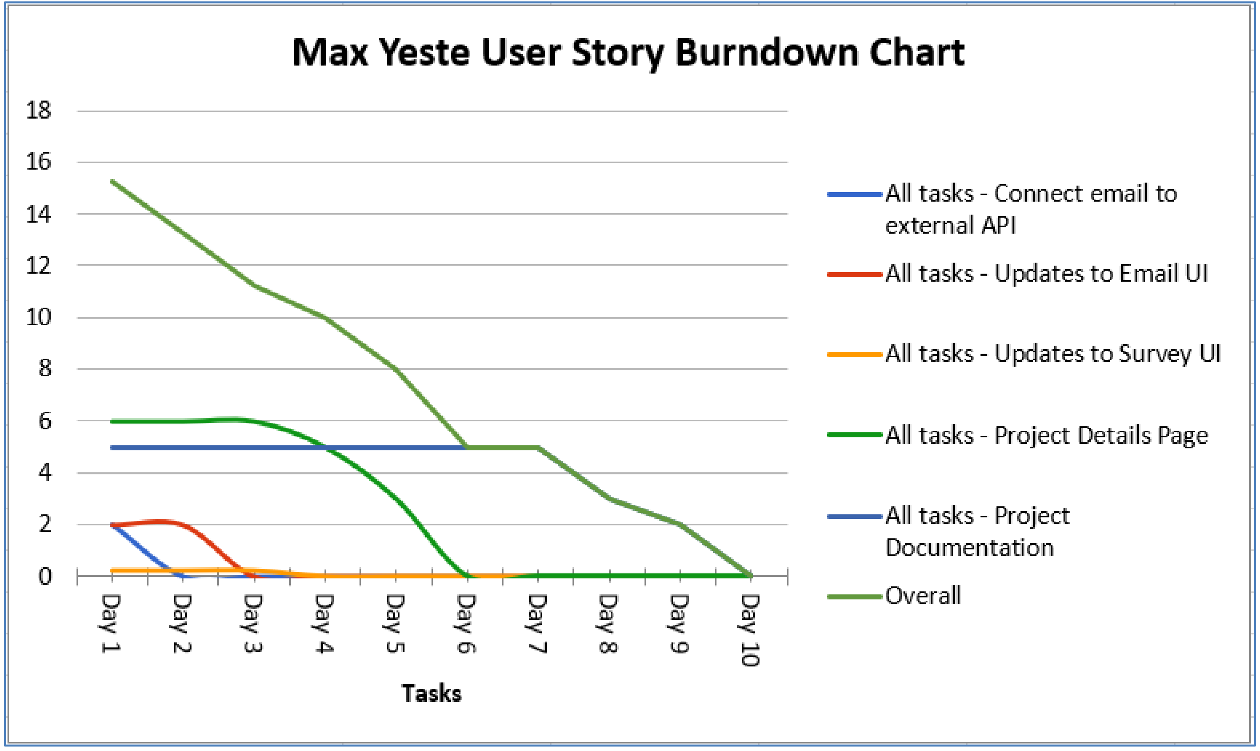
- For the five completed features above, I manually tested the front-end using a web browser. Testing included interacting with the application as well as inspecting the state of the application at different points in time by using console.log to write information to the console in the browser's developer tools.
- I tested the back-end API for each of the five features by submitting requests to the API using Postman.
- I tested the email integration with an external API/service using two different SMTP services:
    - Ethereal – a simulated SMTP service
    - The SMTP server associated with one of my personal email accounts

Individual Taskboard



Sprint #3 Task Board/Chart						
Each Feature	**Points**	**Tasks [hours]**	**Tasks**	**Tasks**	**Tasks**	**Hour Estimate**
Connect email to external API	2	Research Nodemailer [0.5]	Research Ethereal [0.5]	Implement Nodemailer [0.5]	Test with Ethereal and real SMTP [0.5]	2
Updates to Email UI	2	Add sender and recipient to list [0.25]	Add validation errors [1]	Change "sent" icon [0.25]	Change read-only appearance [0.5]	2
Updates to Survey UI	1	Move buttons on student and client survey [0.25]				0.25
Project Details Page	3	Create database schema and test data [1]	Create page with proj. and client name [2]	Integrate email functionality [1.5]	Integrate survey functionality [1.5]	6
Project Documentation	2	Create database schema and test data [1]	Create "view" component [2]	Add S3 upload/download [2]		5
						Total effort hours
Total Story Points	10					15.25

Individual Burndown Chart



Other Information

User Stories written

- Updates to Email UI
    - Pivotal Tracker ID: #162174014
- Updates to Survey UI
    - Pivotal Tracker ID: #162174181
- Project Details Page
    - Pivotal Tracker ID: #162179725
- Project Documentation
    - Pivotal Tracker ID: #162179729

Wireframes Created

- Project Details Page
    - Balsamiq Link: https://balsamiq.cloud/sxfcogu/pg8qwoq/rB23D

Database Schema

Created schema for the following tables/MongoDB collections:

- projects
- projectdocumentation

Documentation/Presentations

- Created PowerPoint slide deck for Sprint 3 End of Sprint Demo
- Created Sprint Planning Board for Sprint 3
- Created final Readme file for application
- Created Client Documentation

Test Data Utility

- In this sprint I extended and modified JSONToMongo.js from Bootcamp Assignment 3. I introduced some changes that make it easier to add new data to be inserted into the MongoDB by minimizing the amount of additional code that is needed to do so. I also introduced the ability to make the insertion of test data idempotent by setting a property called "overwrite" in the JSON file that contains the test data.

Tutorials/Articles/Resources

Email Services

In this sprint I abandoned my previous effort to find an email service that would allow us to send emails from the app using either SMTP or a REST API. Instead, I focused on integrating Nodemailer (https://nodemailer.com/about/) into the app. Nodemailer made it easy to integrate SMTP functionality into the back-end Node.js server app. **Recommendation:** I highly recommend Nodemailer for others that are interested in integrating SMTP functionality into a Node.js application.

However, the integration of Nodemailer still requires an actual SMTP server if it is to send email messages. Fortunately, the Nodemailer website has a link to a simulated SMTP service called Ethereal (https://ethereal.email/). Ethereal provides a free SMTP server that functions just like a real SMTP server except that it does not actually deliver email. This makes it ideal for the development and testing of SMTP functionality in an application that is under development. Ethereal offers lightweight account creation and the ability to view the content of a message that was “sent” using the Ethereal SMTP server. **Recommendation:** I highly recommend Ethereal for development of SMTP functionality in an application.