



Imperium

Team 15

Jeremy Potratz
CJ Enright
Anthony Popilov
Peter Reckamp
Jacob Szarzynski

What went well

This sprint we were able to accomplish all of our user stories and tasks assigned to us, so we were able to be very productive throughout the sprint. We were also able to chip away at some of the things in our non functional user stories (like the design of our site), which will help to make an overall better end product. As a whole, our team was able to use git pretty consistently with only a few minor issues. Git proved invaluable when we wanted to reference deleted code that suddenly become relevant again.

Backend/frontend error handling

We were able to get errors communicated from the backend to the frontend in a pretty generic way that we hope will work for most situations. If there was an error, the JSON response sent to the front end will have a non-null error field with the error to show the user. So far this have proven to be easy to work with and flexible enough to work on all of our views.

Communication

We used a Facebook Messenger group chat to coordinate, and while that's a simple method of communication it ended up being very effective. We always knew when meetings were going to happen, and were always able to hold each other accountable.

User Stories

User Story 1

As a user, I would like to be able to create an account for Imperium so that I can find a job or so that I can recruit.

Description	Assignee	Work Estimate
Create initial backend server	CJ	6 hours
Create initial Firebase database	Jeremy/Peter	7 hours
Create initial React app with router	Jacob	5 hours
Create signup view to send login requests to the backend	Anthony	5 hours
Write controller for user logic	CJ	5 hours
Integrate password hashing library with user controller	CJ	4 hours

Write authentication middleware using JWT	CJ	6 hours
Create server side logic to store user information in firebase	Jeremy	5 hours
Write tests to ensure the user controller functions correctly under expected input	Peter	4 hours

Completed: A user can create an account, and all of the “initial” parts have been completed. We ended up finding that Firebase can do a lot of the account management functionality, however we decided to continue to roll our own authentication to have greater control over it.

User Story 2

As a user, I would like to be able to reset my password if I forget it so that I won't be locked out of my Imperium account.

Description	Assignee	Work Estimate
Create password reset view that can send a password reset request to the backend	Anthony/Jeremy	4 hours
Add “forgot password?” button to login/signup forms	Jacob	2 hours
Write handler for password reset in user controller	Jacob	4 hours
Write code to email user their new, random, temporary password	Anthony	4 hours
Write tests to ensure that the password was successfully changed	Peter	3 hours

Completed: When a user makes a password reset request, node-mailer will send the email with the new random password, and that password is updated in the database. When considering the “magic link” way of resetting passwords, we ended up sticking to this version as it aligns more to what most users are used to. That is, logging in with a password and username as you normally would.

User Story 3

I would like to be able to login and manage my Imperium account so that I can use Imperium as effectively as possible

Description	Assignee	Work Estimate
Create login view that can send a login request to the backend	Jacob	4 hours

Write handler for logging in in user controller	Jeremy	3 hours
Create account management view where users can update their preferences	Anthony	5 hours
Write handler for updating user preferences	Jeremy	4 hours
Create server side logic to edit and update user information in firebase	Anthony	4 hours
Create tests to ensure logging in only works for a correct username and password	Peter	3 hours
Create server side logic to store and update user files in firebase	Peter/Jeremy	6 hours

Completed: Users are able to edit their account and have it update in the database. One nicety we added was the preferences autofilling to show what they are currently set to. We also ended up having most preferences able to be set at the time of account creation on the signup page.

User Story 4

As a user, I would like to delete my account so that I can no longer be associated with the service.

Description	Assignee	Work Estimate
Add delete user section to user preferences view	Jacob	3 hours
Write handler for deleting an account in user controller	Anthony	4 hours
Write tests that creates an creates an account, sends a request to the delete account endpoint, and makes sure that account no longer exists	Peter	2 hours

Completed: We considered not actually deleting accounts and keeping them stored in a separate database in case a user would like to restore their account. It seems most tech companies do this so they have more data and records to work off of. However, that seemed somewhat unethical. Instead of that, we just remove all their data without keeping a trace of it.

User Story 5

As a user, I would like to logout of my account so that others cannot potentially hijack my account.

Description	Assignee	Work Estimate
Logout user after they have been inactive for too long	CJ	3 hours

Add logout button on user view after login	Jacob	3 hour
Write handler for logging a user out in the user controller	Jacob	4 hour
Write tests to ensure that any request after a logout request no longer has a valid, authenticated header	Peter	2 hours

Completed: The authentication part of this ended up being just removing the JWT from local storage, so we were able to simplify the process a lot. To log out users after a certain amount of time we set the expiration time of the JWT to a configurable amount of time after it was generated.

User Story 6

As a user, I would like to be able to easily navigate through the website so that I can efficiently find jobs or potential employees.

Description	Assignee	Work Estimate
Create a hamburger menu to navigate to different pages	Jacob	5 hours
Set up building and minifying our react app with webpack	CJ	5 hours
Write test cases to ensure that your have transitioned to the correct page	Peter	3 hours

Completed: The hamburger menu was made with the help of a react library, and is about as performant as it can be. We ended up finding out that create-react-app already provides a way to create a minified build with “npm run build”, so instead we used that time to add CSS styling to the web app.

User Story 7

I would like to have the option to remember my email for easier login

Description	Assignee	Work Estimate
Prompt a pop up asking if the user would like the information remembered after logging in for the first time	Anthony	3 hours
Write handler for remembering login information in user controller	Peter	3 hours
Create server side logic that enables the user to access a forgotten password	Jeremy	4 hours
Write test cases to ensure users properly access their forgotten password	Peter	2 hours

Completed: We originally had this as “remember password”, however that was already implemented by the JWT storing session info until it expires. We moved to storing a user’s email in localStorage, which makes logging in a lot more convenient.

User Story 8

I would like to create a bio about myself so that companies can quickly understand who I am

Description	Assignee	Work Estimate
Add bio field to user preferences view	Anthony	2 hour
Add bio field to sign up view	Jacob	1 hour
Update sign up handler to handle setting user bio	CJ	1 hour
Update preferences handler to handle updating user bio	Jeremy	2 hours
Write tests to ensure updating bio actually updates	Peter	2 hours

Completed: When a user sets or updates their bio it is immediately available in the database, which should help for viewing others profiles later.

What did not go well?

Part of the reason that we were able to finish all of our user stories this sprint could have been due to us not taking on as many user stories as the guidelines suggest. We didn’t take as many this sprint because the ones did choose were large scale stories that future stories all rely on. For future sprints we will be closer to the suggested number of stories.

Git

On a few occasions we ran into some merge conflicts. Of course, these are to be expected and can be dealt with quickly, but some of those times we accidentally took the incorrect action and deleted the HEAD version instead of the outdated local version. While this was easy to fix, it took us a bit to find out what had happened and rectify the situation.

How should you improve?

1. Take on more user stories so that we can have a good number of features by the time sprint 3 comes to a close. Last time, we missed a point for not having enough user stories. While we did have more than enough hours to keep us occupied, this should be remedied. In the future, we must include more user stories to combat this issue.
2. Take care when dealing with git merge conflicts to avoid headaches of accidentally deleted code. Otherwise, we will waste time figuring out exactly what went wrong and which set of code to keep/delete. This is annoying and can be solved if we are just careful with pulling and pushing to GitHub, while also coordinating with fellow teammates to make sure everything goes swimmingly.
3. Use meetings more effectively to coordinate and set the pace for development. Last sprint, we procrastinated a little too much on finalizing our user stories and creating the testing document. As a result, our final meeting took much longer than it should have and we crammed too much work into one occasion. With more effective meetings, we can schedule work more efficiently and prevent a hefty last-minute workload.