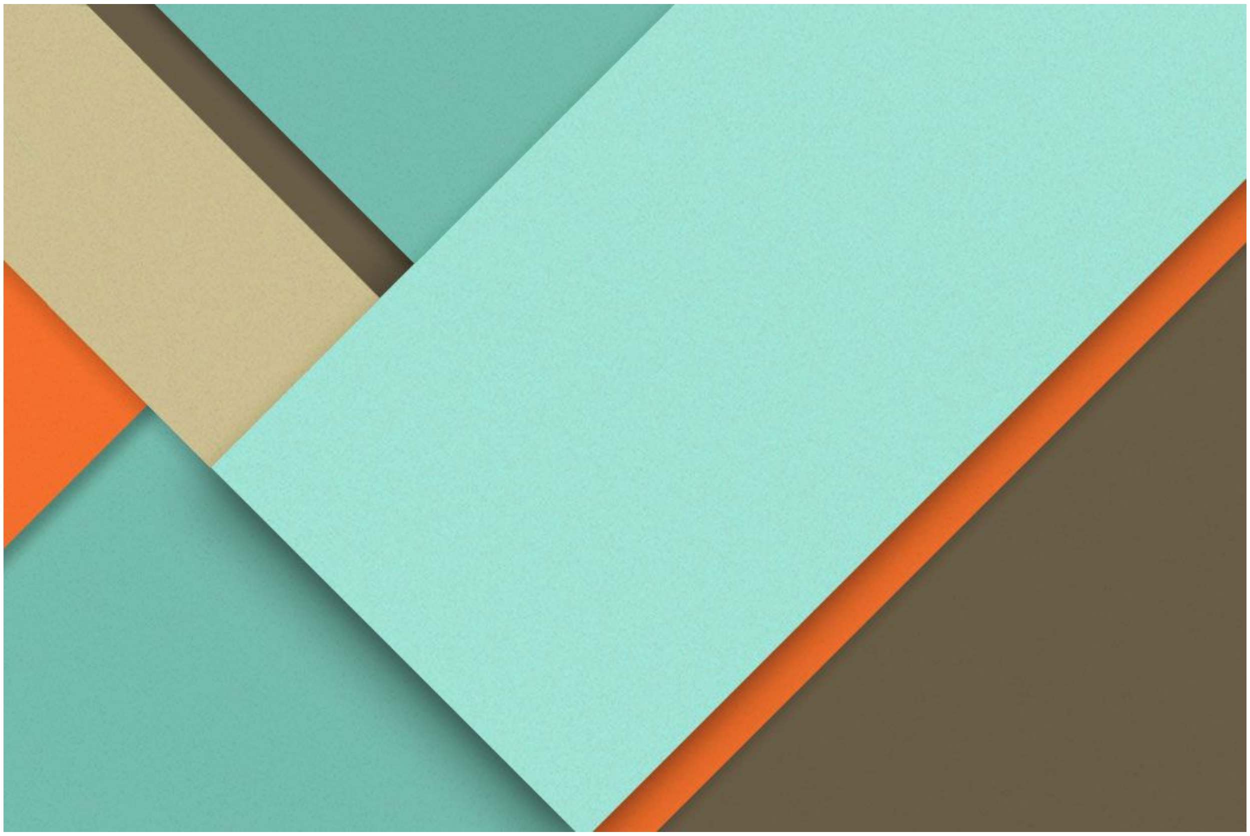

Imperium

Team 15

Jeremy Potratz
CJ Enright
Anthony Popilov
Peter Reckamp
Jacob Szarzynski

What went well

This sprint we were able to complete all of our user stories, and well as the task that we had failed to complete correctly last sprint. We also put a lot of work into making our website look presentable and stable, in preparation for the final presentation. As of now we are content with the current state of the app.

CSS changes

We used a CSS library called Bulma to style our pages. There's a react implementation of that library that makes it a bit easier to use. Using this library let us make all components of our site have a cohesive design. It also helped us to make the website mobile friendly and quick to load

Firebase

Allowing employers to create multiple job postings was a major overhaul of our application and the way we structured data in our database. Luckily, firebase has no strict schema or table structures, and instead just uses JSON for its internal storage. This meant we didn't have many hurdles to clear in order to follow through on this transition. Simply changing our backend code was enough to get our database into a workable state.

User Stories

User Story 1

As an employer I would like a history of past actions (like/dislike) I've performed on students

Description	Assignee	Work Estimate
Add history to company home	Jacob	1 hour
Update action handlers to set history in database	CJ	2 hours
Add view that shows history to user	CJ	4 hours
Write tests to ensure history is correctly stored for liking/disliking/favoriting	Peter	3 hours

Completed: An employer is able to view their past actions of liking and disliking students. They are easily viewable in the history tab on our hamburger menu and shows a good overview of things like, when a user was like/disliked, their profile, their email, and a chronological order of likes/dislikes. We accomplish this through storing the various actions in Firebase then pulling those actions from the database when it is time for viewing.

User Story 2

As a student I would like a history of past actions (like/dislike/favorite) I've performed on employers

Description	Assignee	Work Estimate
Update student handlers to add action history to database	Anthony	2 hours
Update history view for user specific actions (for example, favoriting)	Anthony	2 hours
Write tests to ensure favoriting is recorded correctly	Peter	1 hour

Completed: Similar to how an employer is able to view a history of actions performed on the student, a student is also able to access a history of their actions on employers. We show this in a similar way as with employers, providing the job that you acted upon, what action you actually used, and chronological order with time stamps of when you acted upon the job. This is done the same way in the database as employers, where we store the actions in Firebase and pull them when needed for viewing.

User Story 3

I would like to chat with students I've matched with so that I can find employees to hire

Description	Assignee	Work Estimate
Add chat option to matches page	Anthony	1 hr
Create chatting view	Jacob	5 hours
Write frontend logic to send/receive messages to/from the backend	CJ	6 hours

Write backend logic to store/send chat messages	Jeremy	4 hours
Write tests to ensure messages are sent and received correctly	Peter	3 hours

Completed: We accomplished sending chats through a variety of different processes. All in all the chat looks very stylish and enables a good connection to be made between an employer and student. It also enables students to be able to hold different conversations with different employers as well as employers to speak with different students looking for jobs. We accomplished the chat through storing messages in Firebase with an id for the sender and receiver of the message, and pulling the messages as the chat view is opened by either messaging party.

User Story 4

I would like to hide my profile from other users when I am no longer seeking employment/employees

Description	Assignee	Work Estimate
Add a checkbox to the options page for allowing you to hide your profile	Anthony	1 hour
Create handler for hiding accounts in the backend	Jeremy	2 hours
Write backend logic to store that an account is hidden	Jeremy	3 hours
Write tests to ensure the account is hidden	Peter	3 hours

Completed: A new boolean, ‘isHidden’, was created. If set to true, the frontend will not show that specific user. A new button was added in preferences to change whether a profile is hidden or not, and can be toggled whenever.

User Story 5

I would like to easily schedule interviews with matched students so that I can continue the relationship past our initial meeting

Description	Assignee	Work Estimate
Create button and view on chat view for interviewers to	Anthony	3 hour

create calendar events		
Write javascript code that generates files for students to automatically add the interview events to their calendar	Anthony	6 hours
Write tests to ensure that the created calendar events match with their expected values	Peter	3 hours

Completed: In order to implement this we first had to implement the chat, which is a previous user story. This was implemented on the frontend and is only viewable by employers in their chat menu, and most of the major calendars are included in the options. When an interview is sent, a student can click on the link and it will autofill information in their preferred calendar.

User Story 6

As an employer I would like to be able to see when a student has favorited me

Description	Assignee	Work Estimate
Have an indicator on a student card when a student has favorited the user	Jacob	1 hour
Write a handler for showing being favorited in the backend	Jeremy	2 hours
Write backend logic to display on cards that a user has favorited a company	Jeremy	3 hours
Create tests to ensure that an employer can successfully see that a student has favorited them	Anthony	3 hours

Completed: A new notification was created to notify an employer when they have been favorited. The ability to turn on or off this notification was created in the preferences page. Being favorited will also show up in the user's history, much like their other actions. The company and student will still match correctly if a favorite is used instead of a like.

User Story 7

I would like to receive email notifications that can give me updates about my profile if time allows

Description	Assignee	Work Estimate
The user will get an email notification when they get a match and or a chat message	Jacob	1 hour

Create handler for email notifications in the backend	Jeremy	2 hour
Write backend logic to send email notifications when necessary	Jeremy	4 hour
Write test cases that properly ensure the user was notified via email	Peter	2 hour

Completed: In order to complete this user story, we added an option to the account settings in which a user can enable/disable email notifications. When liking/disliking profiles, a user will receive an email notification if they like a profile that has previously liked them. When a user receives a new message, they are also notified via email.

User Story 8

I would like to get desktop notifications when i get matched or have new messages if time allows

Description	Assignee	Work Estimate
Update react app to allow for showing notifications	Jacob	3 hour
Create handler for desktop notifications in the backend	Anthony	2 hours
Write backend logic to send desktop notifications when necessary	Jeremy	3 hour
Write test cases that properly ensure the user was notified via desktop notification	Peter	3 hours

Not Completed: This user story, as well as number 7, were to be completed if time allows. Although we were able to implement email notifications, we did not have time to create the desktop notification system. This is the only user story listed that was not completed.

User Story 9

I would like to disable certain notifications

Description	Assignee	Work Estimate
Add a checkbox to the preferences page that will let the user turn notifications on or off	Anthony	1 hour
Create handler for notifications being on or off in the	Jeremy	1 hour

backend		
Write backend logic to keep track of whether notifications are on or off (and act on that accordingly)	Jeremy	5 hours
Write test cases to ensure that a user gets notifications at based on their preference	Peter	3 hours

Completed: This user story was implemented in the Account settings, in which a user is able to toggle various notifications on/off. In this case, a user is able to hide/unhide their profile, as well as turning email notifications on/off. This would have also applied to desktop notifications had that user story been completed.

User Story 10

I would like to receive a notification when another user favorites me

Description	Assignee	Work Estimate
The user will get a email notification when they get favorited	Anthony	3 hours
Create handler for favorite notifications	Jeremy	2 hours
Write backend logic to send favorite notifications when necessary	CJ	4 hours
Write test cases to ensure that a user gets favorite notifications	Peter	3 hours

Completed: This was implemented in a similar way to the previous notifications features. It required a special setting for job postings to be able to disable favorite notifications. We use the nodemailer package to send emails, and gmail hosts our actual email server.

User Story 11

I would like to have a main page for my company

Description	Assignee	Work Estimate
Add ability to navigate to and around a company's home page	Jacob	2 hour
Create frontend view to show company home and their job listings	Jacob	5 hours

Create a handler for retrieving home page data	CJ	3 hours
Write logic to display job postings	CJ	4 hours
Write tests to ensure that the company can successfully see all of their job postings	Peter	3 hours

Completed: This was a rather involved change that fundamentally changed how we store our data. Luckily, firebase stores data very flexibly so that change was mostly straightforward. Now, employers create job postings which are stored in a separate table than users are stored in, and are addressed by the company's email followed by a number

User Story 12

I would like to create separate job postings so that I can more effectively find potential employees

Description	Assignee	Work Estimate
Have a button that will take you to a new page to create a new job posting	Jacob	1 hour
Create view for creating new job postings	CJ	4 hours
Write backend logic to create and store job postings	CJ	5 hours
Write tests to ensure that a company can successfully create job postings	Peter	3 hours

Completed: Similar to above, this involved changing a lot of our UI elements to work nicely with this feature. An employer can create a new job at the company home page, and view their current postings there as well. Creating a posting is similar to creating an employer account before, but now the resulting object is owned by the employer who created the posting.

User Story 13

I would like to be able to navigate between my seperate job postings and search for students within those postings.

Description	Assignee	Work Estimate
Be able to navigate back to the company home page	Anthony	1 hours
Write frontend view to allow for searching for students within a job	Jacob	3 hours

Once a certain job profile has been selected, a user can go to the normal home page and see potential matches for that job	Jacob	4 hours
Write tests that ensure that a user can successfully navigate between job postings	Jacob	4 hours

Completed: To accomplish this we updated the hamburger menu to show only the company home page as well as the signout button. To navigate between job postings you return to the company home and can click on buttons that are shown for each job posting you've created. For example, the edit job button will take you to a view where you can edit the preferences for a job posting.

Sprint 2 remaining tasks

User Story 11

I would like to 'favorite' up to 3 companies to show that I am extra interested

Description	Assignee	Work Estimate
Write logic to keep track that only 3 companies may be favorited at a time	CJ	3 hour

Completed: This was a simple fix that we finished pretty easily. We just needed to show on the frontend that a user was not able to Favorite after their designated three favorites. We did this by hiding the Favorite button in the Matches page after the student hit their Favorite limit.

How should you improve?

1. We did not do a good enough job on finding small design flaws in the application. The styling on some of the pages was not as desirable as we would have liked, specifically the view pages. It was rather bland and did not have the outline that the cards had on other page. In the future we need to be more thorough with checking out our work as this could have easily been caught and fixed with a good check through the front end. Additionally we can add more test and get more thorough with the testing, especially towards the end of the sprint.
2. This sprint we failed to optimally take advantage of group meetings. Most weeks we ended up not meeting and when we did meet up some group members were not

able to attend. Consequently we ended up having some small miscommunication issues on who should work on what and had a little bit of wasted time and merge conflicts due to members working on the same thing. Additionally we had members trying to work on a user story but they did not know if they can start and as a result put off the task till later. In the future we need to have mandatory group meetings. that work for everyone where we can have detailed discussions on what needs to get done and who should do it.

3. By using git branches we could improve our productivity as a team. We could create a feature for each branch that isn’t merged into master until we’re confident in its stability. Not only would this make our master branch more stable, but it would also make developing the app easier and more clear. That clarity will come from knowing who’s assigned to which task, and exactly what they have to do for that task before it will be merged into master.