Design Decisions

User Testing Feedback

Summarize the feedback from user testing, noting what you found most surprising/interesting about the feedback

Note each step that you asked the participant to take in its own paragraph with heading. There should be no more than 8 steps that the user takes. Then, write one paragraph summarizing your reflection about what was surprising. Each paragraph should be no more than 3 sentences.

1. User Creates an Account

Our user immediately figured out the tab system for our login/sign up pop up. However she expected to be redirected to the homescreen, so remaining on the profile page was unintuitive.

This is interesting since we didn't want to navigate users away from their task when designing our pop up system, but this test suggests that we should redirect users when their account is created.

2. User Creates a Freet.

Our user immediately found the Create button in the navbar, started typing out a Freet, and posted. She then found the Freet on the Explore page. The whole process went very smoothly, and she seemed to intuitively know what to do.

Perhaps it was a little surprising that she was able to figure out how to create a Freet so quickly, since the Create button doesn't specify what it is creating. It's possible that the appearance of a new button after logging in made the Create button particularly conspicuous and noticeable.

3. User Edits a Freet

Our user found the pencil icon on a Freet she made, clicked it, and saw that the Freet content turned purple. She then guessed that it was editable, added some text, and clicked out the box. Afterwards, she said, "Oh, okay!", indicating that her intuition was correct and that editing a Freet seemed natural to her.

It is a little surprising that she was able to figure out that the Freet's content was directly re-editable as there are very few indicators of such, but it makes sense since the color of the text changes.

4. User Creates a Refreet

Our user hovered over the Refreet icon, clicked it, and immediately typed in a comment, then moved on to making a Refreet chain. She was confused initially that the Refreet viewer only

showed immediate children of Refreets, but after viewing several Freets in the Refreet chain, she understood how it worked.

This is a bit surprising since we assumed that indentation and the purple line stretching under direct Refreets would provide enough clues to indicate this property. However, this test indicates that we need to provide more signs that the Refreet column only shows direct Refreets. We could potentially do this with a message on the right hand panel before a Freet is selected.

5. User Deletes a Freet

Our user hovered over the trashcan until the tooltip showed, clicked it and the confirmation message after, and then seemed unsure if her message was actually deleted.

This was a bit surprising since we designed the feed to automatically update once the user deletes a Freet, and we also colored the user's Freets differently to help further distinguish them from other Freets. However, the repeating color scheme still lends itself to look repetitive so that consecutive Freets look similar to each other.

6. User Follows another User

Our user was able to follow themself after posting a freet. They were surprised to learn that you cannot follow users directly from their profile.

This is interesting because we believed that there might be use cases in which a user would want to be able to follow themself (such as being able to see their own posts in their feed). However, it might seem unintuitive or surprising to some users that they can follow themselves.

7. User Changes Username

Our user tried to change their username to that of an existing user, and failed in doing so. She then proceeded to try to change their username to be similar to that of an existing user, except with the first letter capitalized, and succeeded in doing so. She then wondered if she could delete that other user's posts due to having a similar username, but was unable to do so.

This is an interesting case that we did not consider, as the user intentionally tried to masquerade as another user and eventually partially succeeded in doing so. It is intriguing to consider this negative consequence of the fact that we allow multiple usernames that are identical but have different capitalization.

8. User Re-Sorts Freets

Our user clicked the three sort options on the feed, but because her feed was nearly empty, the sort options didn't do much. She seemed confused about what each of the icons did, and she remarked on her confusion around how the Freets were sorted.

This is interesting because we expected users to have a lot of Freets in their feed, and so clicking on one of the sort options would yield more obvious results. This test suggests that we should put a header over the sort options, specifying their functionality.

Sketches and Finished Frontend

1. Displaying Children Refreets (Sketch + Implementation Decision)

We added a feature which allows you to view all freets which have refreeted a specific freet. This allows the creator of the freet to view all responses to their post, allowing them to see how others have engaged with this content. We believe this decision will significantly increase user engagement in our site, allowing more ways for users to interact with each other, such as asking questions or relating to other people's content and receiving responses from those users.

However, the limitations of this feature are that users might be more ambivalent to refreet other's content in fear of negative backlash. It might even lead to heated discussions and back-and-forths in chains of refreets, which is a more negative atmosphere than we would like. We mitigate this by only displaying one layer of children refreet, which makes it much less likely for such lengthy chains to be created. Furthermore, refreets are created as actual freets rather than something similar to a comment, so such chains would end up cluttering one's content feed, which could reflect negatively on the user and thus users would be less likely to do.

One alternative we considered was not just displaying the immediate refreets, but recursively the entire chain of all refreets of those refreets, etc. Though this can make it easier to engage in chains of discussion, this can also lead to heated discussions which creates a more negative atmosphere than we would like. We also considered an alternative of not displaying any of the refreets at all, though this limits users' ability to engage with each other and doesn't even completely eliminate the possibility of a back-and-forth between users as they can still refreet each others' content.

2. Displaying Follower and Like Count (Sketch Decision)

Our design displays the total like count on each freet, and the total follower count on each user's profile. We believe that seeing that users seeing likes increase on their posts and seeing their numbers of followers increasing will give them positive reinforcement, and encourage them to post more content and engage with their followers. However, one limitation of this is that it might be harder for lesser known users to become popular as they start out with low follower and like counts, which might make other users less willing to view their content and might discourage them from continuing to post content.

One alternative we considered was not just displaying the total like and follow count, but the actual list of users who have liked a freet or followed a user. While this may build a sense of community amongst people who like similar content and follow similar users, it can also build a sense of hostility between users, especially users who know each other in real life, when they don't follow each other or like each other's content. It might also discourage people from supporting content which they might like, in fear that they will be judged for their publicly-displayed interests.

Another alternative we considered was displaying the follower and like count of posts only to the owner rather than publicly to all users. This maintains the upside of positive reinforcement and the downside of negative reinforcement. However, the main difference with this alternative is that users will no longer get a sense of how good or bad content is based on the number of likes or followers that it has. This alternative is less desirable, though, as it prevents users from being able to filter out extremely unwanted content (indicated by having near-zero follows or likes).

3. Separating Profile and Settings (Sketch Decision)

Our design has a profile page for each user, and a user can go to their own profile page to see all of their freets. However, in order to perform actions directly on their account such as changing their username and password, they utilize a settings popup that is separate from their profile. This settings popup is available by clicking on the appropriate icon on the Nav Bar from anywhere on the site, allowing a user quick, convenient, and easy access to their account settings. One limitation of this choice, though, is that it might take a while for a user to discover that these settings exist.

One alternative we considered was integrating the user settings directly into the profile (for example, having an edit button next to your username indicating that you can edit your username). However, this feature might be somewhat unintuitive, as you can view other users' profiles (but not have the ability to edit their account settings). Furthermore, it would also just clutter one's own profile. Very rarely will a user want to both view their freets and also adjust their account settings, so in order to keep our interface cleaner in both use cases we have kept the profile and settings separate.

Another possible alternative could have been, rather than having a settings popup accessible from anywhere on the site, we could have a settings page (entirely separate from the profile page). However, by having to navigate to an entirely different page, the user's experience would be interrupted, i.e. losing their position in the feed. Our settings popup allows the user to take a brief detour to adjust a setting, such as changing their username or password, and then immediately return to what they were doing, resulting in a smoother user experience.

4. Sorting Freets (Sketch Decision)

We give users the ability to sort freets by newest and by popular. We chose this over the alternative of not allowing the user to choose their sort (defaulting to the same sort each time which we may even be able to curate to try to optimize the user experience) because we would like to give users greater agency over the type of content they would like to see - they can use the newest sort to view more recent content, or the popular sort to view more popular content which is more likely to interest the user. Thus, the order of freets displayed is more likely to satisfy their personal preferences.

However, one limitation of this choice is that it is still harder for users to discover lesser-known content producers, especially those that post at different times than the user is checking Fritter. We have mitigated this by adding a random sort feature, which retrieves a feed consisting of randomly selected freets, which gives users more opportunities to explore the wealth of content shared on our site.

We also considered having a larger variety of sort options to accommodate more possible ways for users to explore Fritter. However, this could lead to a more cluttered interface, and so we decided to limit the sort options to a select few that we believed to be the most useful - Newest, Popular, Random. We could also have an advanced sort options menu which the user presses a button to open up, but this would add minimal benefit at the cost of more complexity to our interface than we would like.

5. Autofill Username (Implementation Decision)

Our Nav Bar includes a user search bar, which allows users to search up usernames of existing users in order to be directed to their profile. We also have an autofill feature, which displays a few usernames which begin with the string that the user is searching. Thus, a user might only have to type a few characters of the user's username before the autofill dropdown shows an option containing their full username and provides a link to their profile. This is especially useful if the user they are searching for has a particularly long username, or a sequence of characters at the end that might be difficult to remember exactly such as a sequence of numbers. One limitation of this design choice is that our autofill currently checks for users that begin with the search string, so it doesn't correct for typos in the middle of the string.

One alternative would have been to have the user search direct to a separate page containing a list of all users that share a similar username to what the user searched for. However, this would not save as much time as our autofill feature, as the user first has to search the string, navigating to this page, and search this page to identify the user they are looking for and finally click on the link. The autofill has the benefit of providing immediate feedback if no users begin with the string the user is searching, indicating that the user might have made a typo and allowing them to correct for it quickly and effortlessly (rather than having to perform multiple searches).

Another alternative could have been to just require search phrases to be exact usernames, and if a username did not exist then the search would direct the user to a page notifying them of such. Since we already link to a user's profile on their freets, the only benefit of this search feature would be if the user learns of another user's username through alternative means, such as being friends in real life. However, even in this case users might be prone to forgetting small details of the username, especially numbers that might follow the username, so the autofill would be helpful.

6. Tooltips for Icon Buttons (Implementation Decision)

Our website has most functionality represented by small icons. Clicking on these icons activates the corresponding functionality, such as editing or refreeting. However, as the functionality might not be immediately obvious from the icon (especially to less tech-savvy users), we display a tooltip indicating the functionality whenever a user hovers over an icon (i.e. hovering over the heart button on a freet will either display “Like” or “Unlike”). Thus, if a user is not certain yet what clicking on a certain icon does, they can always hover over it to find out. Though, one limitation of this choice is that it takes a little bit of time for users to recognize the icons to be able to comfortably use the site’s functionalities.

One alternative we considered was having the name of the functionality just displayed next to the icon, or even just having the name of the functionality in place of the icon. However, while this would make it easier for newer users to understand the full functionality of the site immediately, it would result in a very clunky interface. Given that we have very few kinds of icons, these icons are very representative of their function, and that we expect these icons to be used with high frequency, it should be easy for a user to quickly gain the ability to recognize the functionality of all the icons without having to hover over and read the tooltip.

Another alternative might be having some sort of tutorial pop up when users first use the site to learn all the functionality, which the user can return to whenever they want. While this may help less tech-savvy users learn the functionality, it may prove to be obstructive to more tech-savvy users who immediately recognize the functionality of most of the icons and do not need such a tutorial. Furthermore, it is more likely the case that a user will forget the functionality for a single icon rather than for all of the icons on the site, so having to go through a tutorial or search through a table of contents on a help page to find a small piece of information would be very inconvenient - it is better to just provide this information right next to the icon.