

Flow

Streamlining your creative process so you can pursue your passions.

Hi-Fidelity Prototype Report

Genie Hyatt: *Team Leader and Documentation*
Trijeet Mukhopadhyay: *Designer and Developer*
Ken Chhan: *Developer and User Testing*

Problem and Solution Overview:

Finishing a creative endeavor can be extremely daunting, difficult, and discouraging, and when this is combined with the many barriers one faces when attempting to finish their creative project, the individual often gives up, or puts their project on the back burner indefinitely.

Our product will help individuals to streamline their creative process through organizational tools such as a and map, to make each task seem less intimidating and more manageable. It will also provide mechanisms such as reminders and deadlines for the individual and others to hold themselves accountable, and a networking aspect that will allow them to share their progress so that they can receive feedback from other creators. We believe that providing organization, accountability tools, and more readily available feedback will significantly increase individuals' motivation, resilience, and likeliness that they will complete their project.

Tasks

1. Quickly add a note about inspiration (simple)

Our contextual inquiry participants noted that they believed inspiration could come from anywhere, and that they often had great ideas while simply going about their day. However, in the rush of every day life it is not always easy to jot down these ideas and be able to return to them later, so we thought it important to provide a quick and easy way to take a snapshot of an idea and file it away with all of the others for ease of access at a later time.

2. Create a new project (moderate)

To finish a project, an individual must first start one. This task was to create a brand new project with a title and a description that they could start working on. We can also have them move over an idea from their idea board into this project, since projects are often started because of ideas that the individual has had.

3. Create a task and set a deadline (complex)

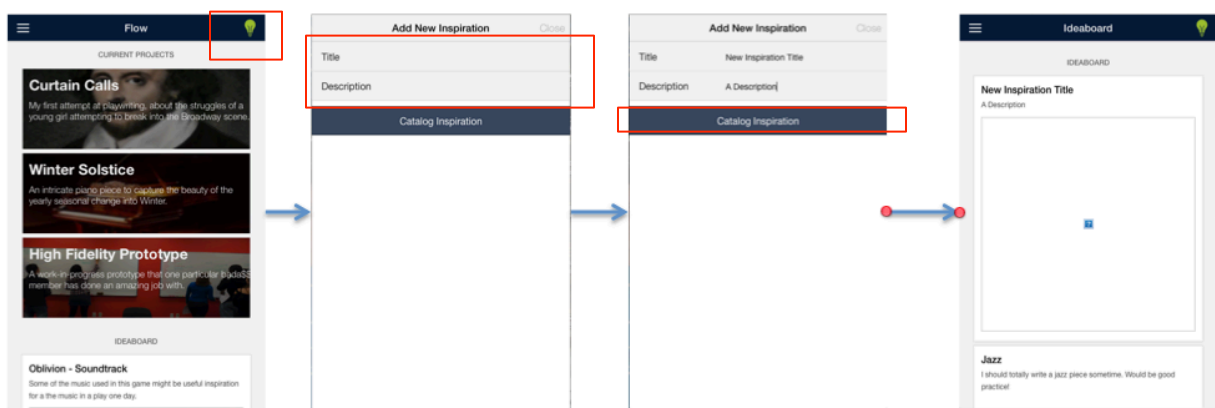
One of the core functions of our app was to facilitate the breaking down of large, overwhelming projects into smaller, bite-sized chunks. This task asked a user to go to their project, and then give themselves a task with a deadline, breaking up their project and making a commitment to themselves to work on it.

4. Finish a task and share the work (simple)

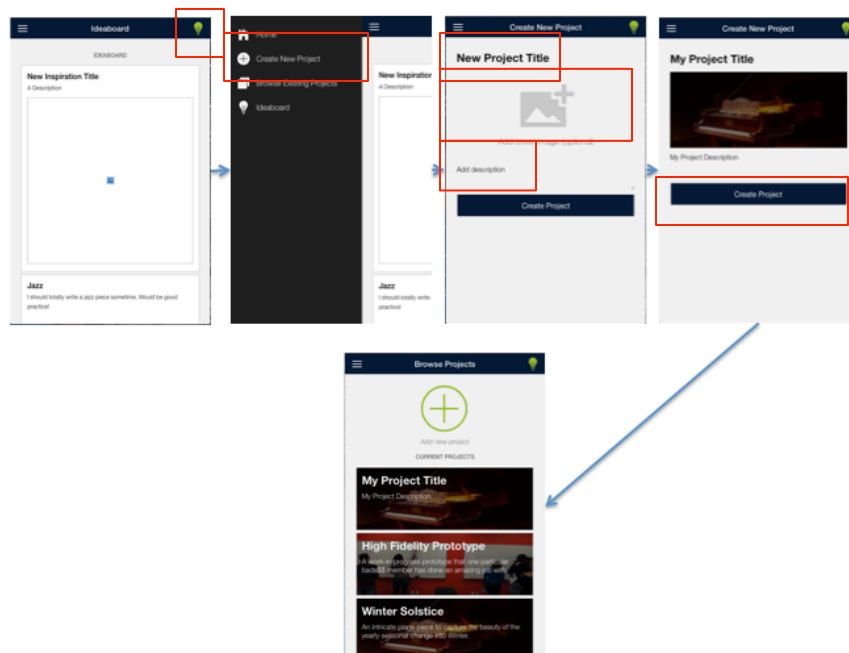
Finally, our research indicated that feedback was both extremely important and extremely hard to get while working on a creative project. After working on a task that the user has assigned to themselves, they should upload some of their work and share it with their community and network of friends. After sharing their work, they should check the comments and feedback to see how their work was received and what they should continue with.

Storyboarded Scenarios

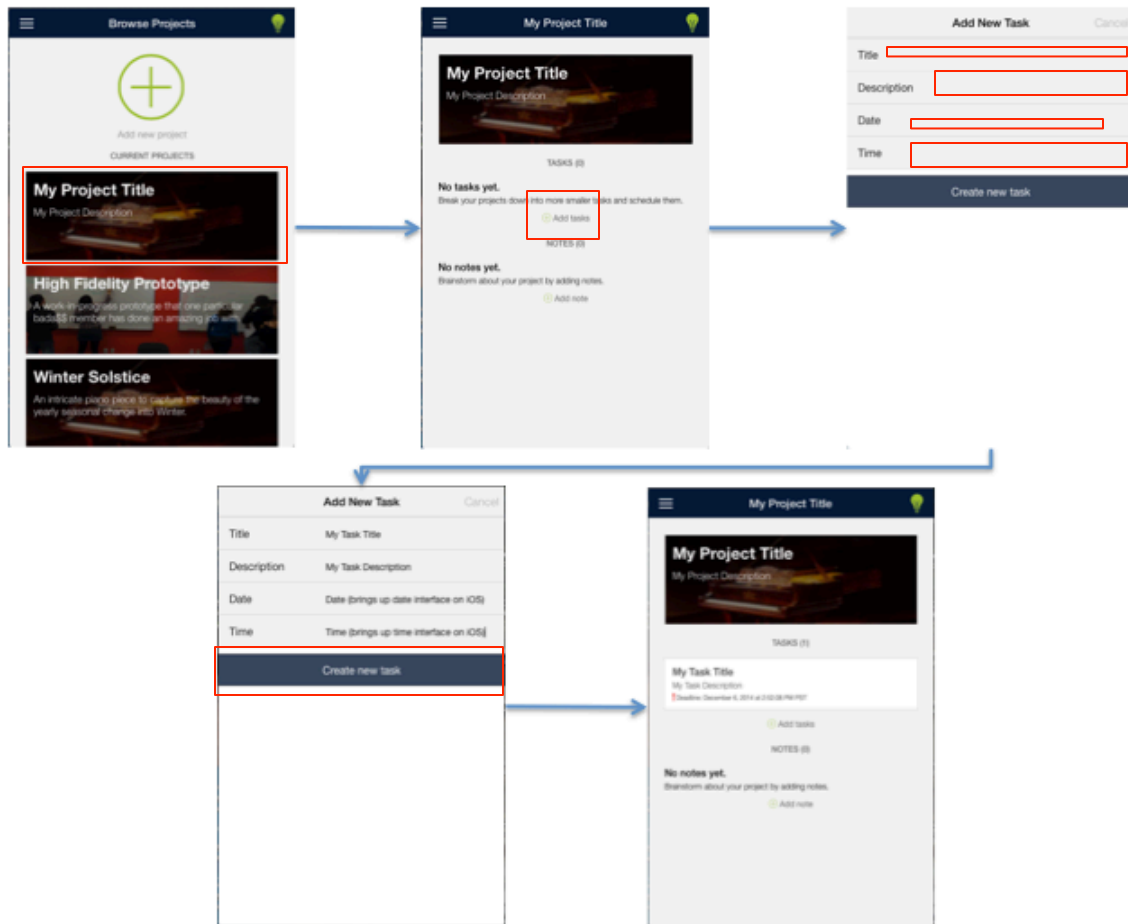
1. Quickly add a note about inspiration (simple)



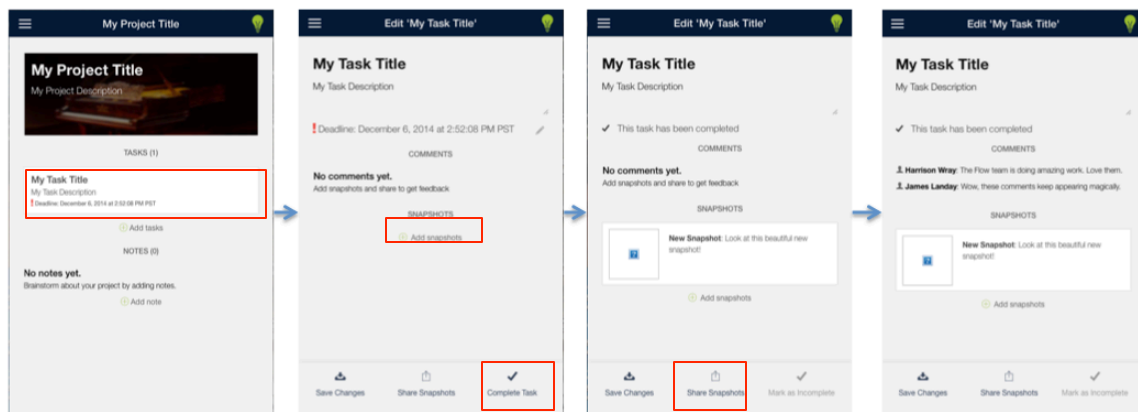
2. Create a new project (moderate)



3. Create a task and set a deadline (complex)



4. Finish a task and share the work (simple)



Major Usability Problems:

1. [H2-1] [S: 3]: *There should be a way to move an idea from the idea board into a project.*

Implementing this was an idea that we had had for awhile but that proved more difficult than anticipated with the prototyping tool InVision, in our hi-fi prototype, you can access the option shown in Figure 1 by tapping on an idea in the idea board and choosing which project to move it to.

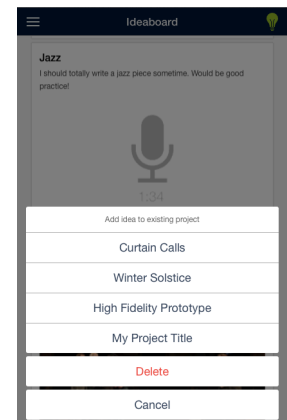


Figure 1

2. [H2-4] [S: 3]: *The “+” symbol in the upper RH corner could mean add different things, and it’s not clear that it’s only for adding inspiration to the idea board.*

This problem had not occurred to us while creating the button, but on receiving the feedback we changed the symbol to that of a light bulb as shown in Figure 2 because it seemed to be more explicit about adding just an idea.

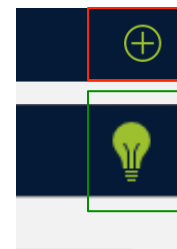


Figure 2

3. [H2-1] [S: 4]: *Unclear whether the main screen for a project should be the edit-project view or the web. It’s also not immediately clear how to switch back into the web view.*

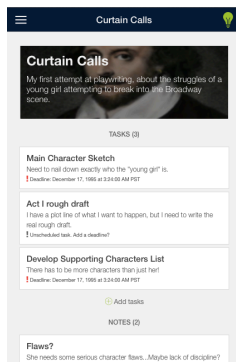


Figure 3

This was not a usability problem that we were able to completely tackle in our hi-fi prototype due to our inability to program the web-view as we had wanted. Instead we changed our web view to more of a list view (Figure 3) and added a back motion (mentioned in the next comment) that we think helped relieve this confusion.

4. [H2-3] [S: 4]: *There are many places in which there is no immediate back button in which it would be very useful.*

This was an important problem, and in our hi-fidelity prototype we added the ability to go back by swiping to the right. This was simple and didn't make our interface look any different, but we do think it was a very valuable change.

5. [H2-3] [S: 4]: No way to exit the 'comments' screen on the web-style overview, not even through tapping a menu again

This was human error in creating the mid-fi prototype, and was different in the hi-fi prototype because the comments were viewed differently. Without the web-view, comments were simply seen on a task edit page in a list (Figure 4)

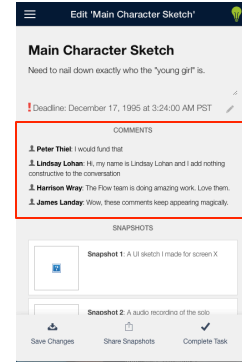
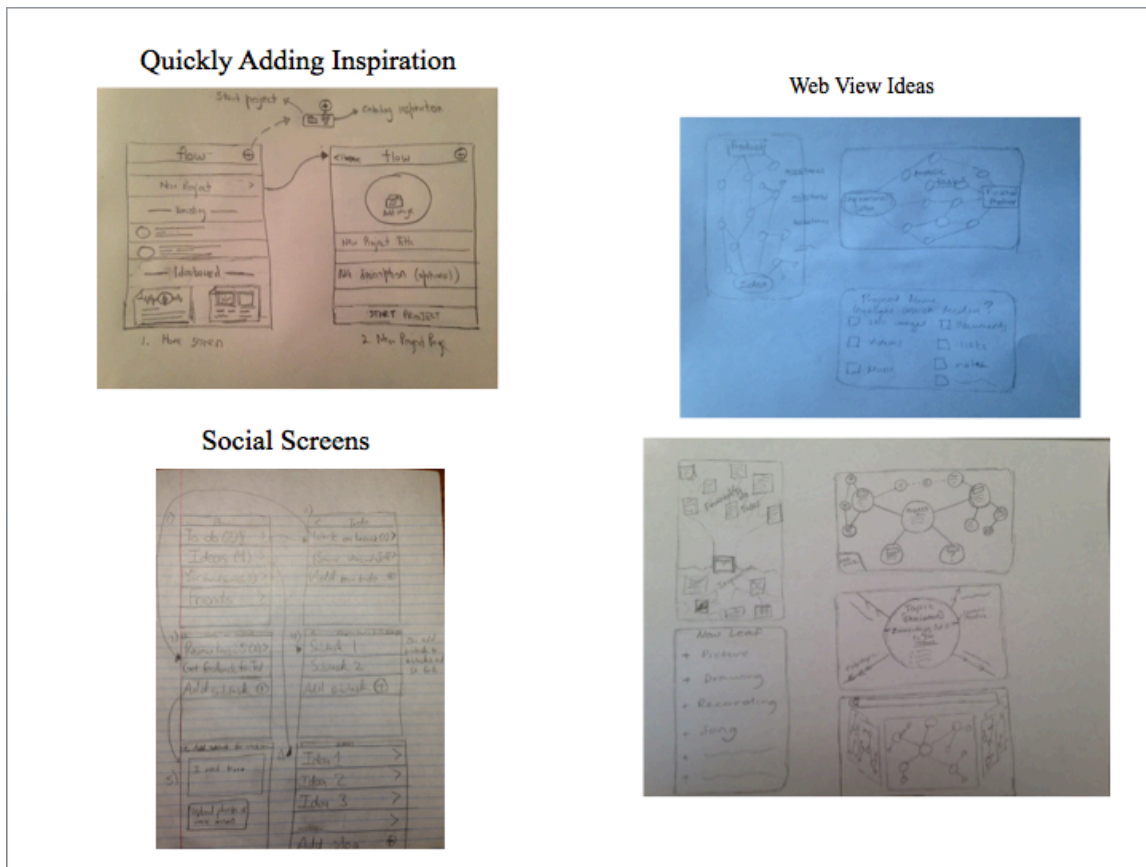


Figure 4

Design Evolution:

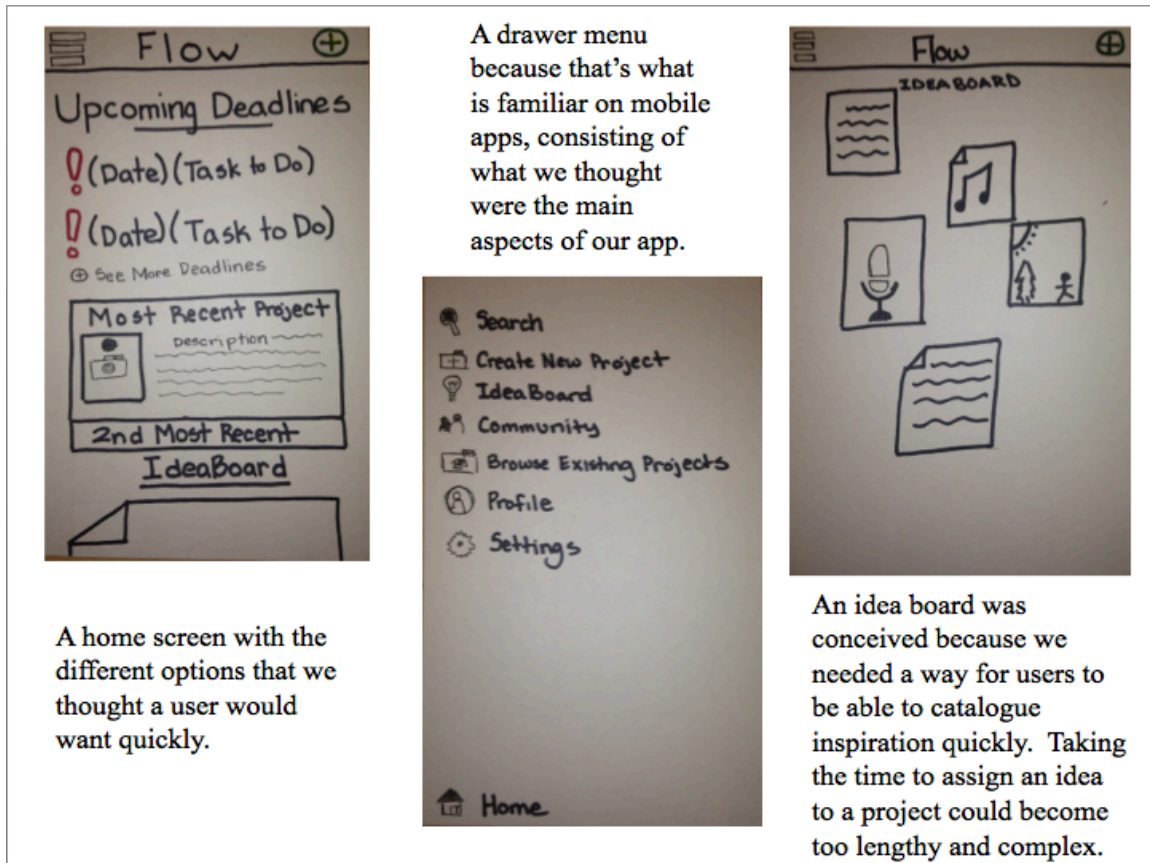
Initial Sketches

These are initial, quick ideas about how the user should interact with the app and how data should be displayed and organized. We based our sketches on findings from out contextual inquiry, such as the need to quickly jot down inspiration, and for organization.



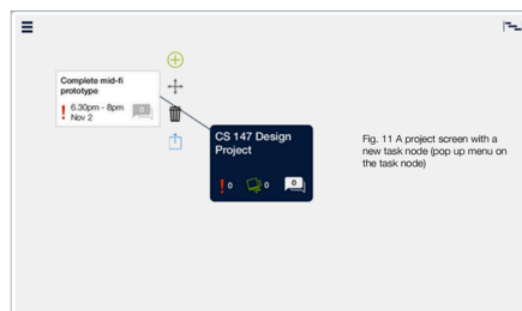
Paper Prototype

Our paper prototype was a conglomeration of the initial sketches into a more organized and detailed set up. Such things as menus, and placement of buttons had to be figured out, as well as an overarching idea about how we wanted our app to flow.



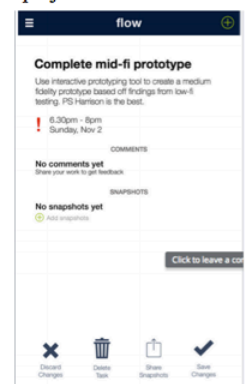
Medium-Fidelity Prototype

Using results from user testing on the paper prototype, we knew we had to make a clearer map/web-view so that users knew that the nodes were interactive. We also knew we needed the data more easily accessible, and not so hard to find. For example, a user should be able to click on a node and see all the information on that node that they need, without making too many more clicks.



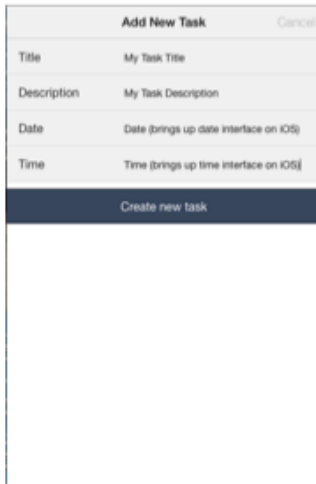
A web view of one of the projects with a node and a pop up menu next to the node. We had to think of different touch actions that could be used to interact with the web easily and intuitively, as well as what each node should display. Because of the limited real estate of a node, we developed icons.

A view to edit and see the details of a task, as well as the comments. We wanted comments to be easily accessible as feedback was an important part of our project.

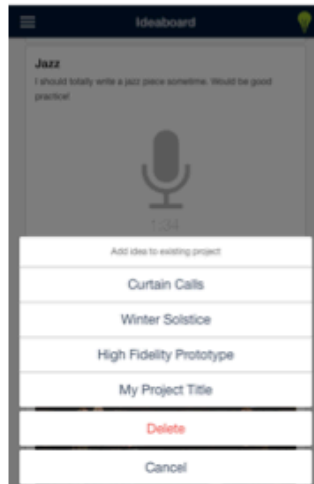


High-Fidelity Prototype

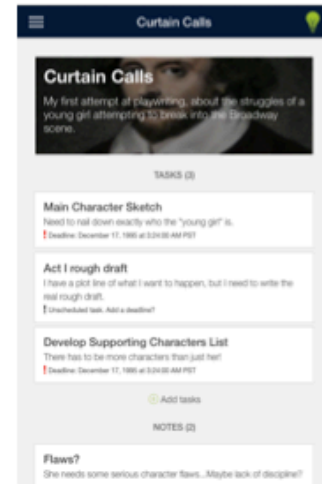
The high-fidelity prototype used results from a heuristic evaluation of the medium-fidelity prototype. Some major issues that were fixed were a lack of interaction between the idea board and the projects, as well as issues with adding things. Modals helped to make what was being added more clear to the user, and the “+” icon was changed to a light bulb so that its purpose was more clear. We also made navigation between screens easier with the help of a back motion.



Modals for some interactions were added to make them slicker and more intuitive, rather than entirely separate pages.



More seamless interaction between the idea board and the projects.



A list view rather than a web view of the project due to technical limitations, though deadlines are listed first as they are more pressing than inspirational ideas.

Prototype Implementation:

Tools

To implement our hi-fi prototype, we used a combination of AngularJS and Ionic. This allowed us to create our screens and controllers using HTML, SCSS, and JavaScript, while still creating a product that looked and felt like a real mobile application. Our original intention was to use Xcode and Interface Builder, but both the learning curve of native iOS programming and the difficulties of getting the right permissions to effectively test an Xcode project were crippling.

AngularJS and Ionic allowed us to use tools that were already familiar such as HTML and SCSS to create interfaces that actually looked how we wanted them too. They also made it possible to test the application quickly and easily in a web browser,

reducing the frustration and difficulty of testing on an actual phone while we were struggling with UDIDs and Provisioning Profiles. Finally, one of our group members did not own a mac, making it impossible for him to work with Xcode, but completely possible for him to work with AngularJS and Ionic.

Although these tools were more familiar to us than native iOS programming, we were not masters at them, and creating the dynamic and interactive web view proved to be far too complex given our skill level and time frame. This was very frustrating, as it was a central component of our interface design.

Wizard of OZ

Our most obvious wizardry was our implementation of the commenting. Our app had no true back-end or networking capability, but we still wanted our users to get “feedback” on their completed tasks. Our solution was to create hardcoded comments that were set on timers after a snapshot was “shared”. After the user shared a snapshot, comments would slowly start to appear on the screen, as though people had been looking at the work and commenting naturally.

Hard Coded Data

Because our app is meant to be used over a long period of time, and hold lots of relevant data, we created three hardcoded projects in various stages of progress. This was so that a the user could flip through the app and see what it would look like and how you could interact with it if you had already been using and adding things to it for quite some time. We also added a couple hard coded inspiration notes to the idea board. Without these hardcoded elements, the app would have consisted mostly of blank screens, and that’s not how a user would ordinarily see it.

We also hardcoded the comments, since there was no actual networking aspect, and we had to Wizard of Oz the social interaction part.

Finally, a few pictures were hardcoded into the projects that the user’s created, because the app could not actually access the phone camera or photo reel.

Missing Features

To have truly created the app we want, we would have needed a back end and a networking aspect, but either of those are for a finished project, and not necessarily a hi-fi prototype. What we were truly missing was the web-view.

Our project hinged around this digitized mind-map, brainstorming, and organization tool, and we were not truly able to test our interface without it. Something of that complexity is difficult, and I do not believe it was possible for us at that time to implement in code, but our prototype was much weaker without it.