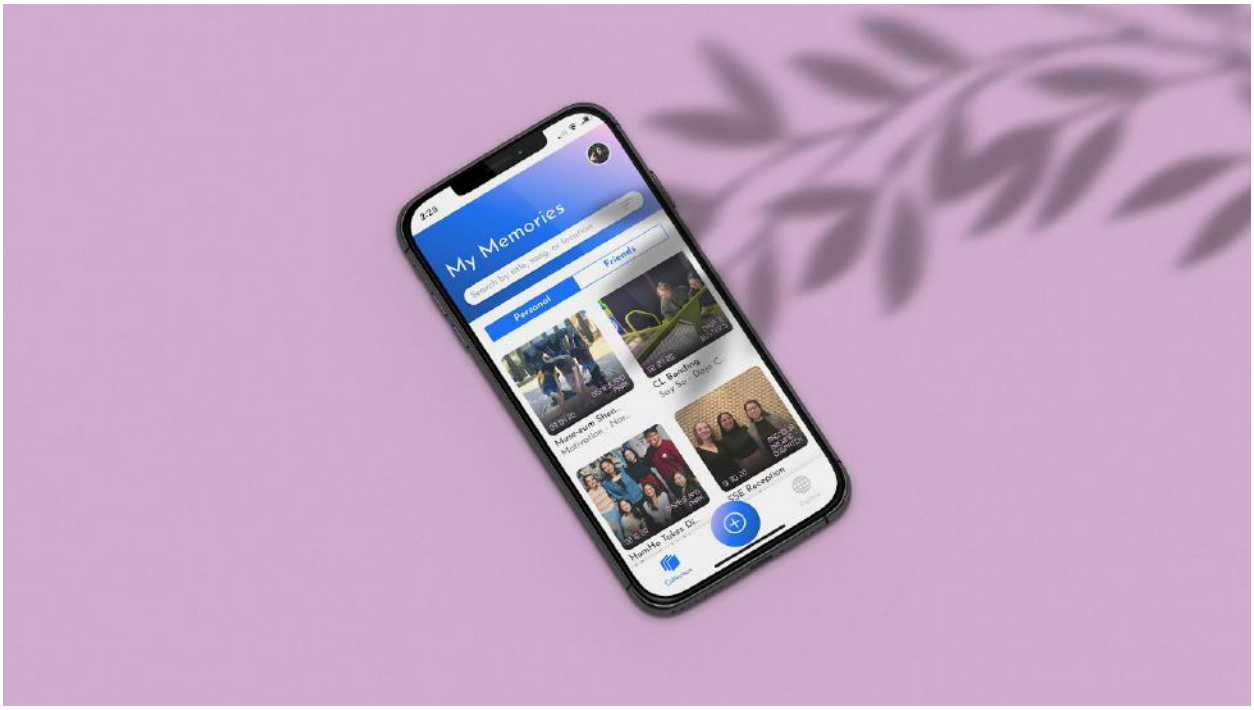
MUSE



Final report + documentation
CS 194H Winter 2020

Vincent Nicandro, *UX Designer/Web Developer*
Tiffany Manuel, *App Developer/UX Designer*
Alema Fitisemanu, *UX Designer*
Nylah DePass, *App Developer/UX Designer*

Table of contents

Problem + solution overview

There's something about memories that we naturally want to have a record of. But current methods to keeping these memories have some drawbacks: journaling is time-consuming, and photos only capture visual details. Instead of simply storing one-dimensional mementos of the past, we aim to recreate the environment of the memory itself.

Muse interweaves music with memories to make memories a delight, giving users a better way to remember by recording the location and song associated to a memory. In this way, we not only remember these experiences; we can Muse about them.

Figure 1. Core screens of the app.

Tasks

Create a memory

Complex task

The user taps on the + icon on the bottom menu bar, taking them to a screen where they can fill in all relevant information for their new memory (title, song, location, date, photo/video cover, and memory description). After filling in all the fields, the user taps the Save button on the center bottom of the screen to save the memory to their collection.

View a memory

Simple task

The user taps on a memory card in their collection, taking them to a memory screen which features all relevant information for that memory. The user can scroll to read the memory description. If they so choose, the user can click on the play button next to the song, and the song will launch outside of the app onto Spotify.

Share a memory

Medium task

The user taps on a memory card in their collection, taking them to a memory screen which features all relevant information for that memory. The user then taps the Share icon on the bottom right corner of the screen (a box with an arrow pointing upwards). The user has the option to either share to a friend on the app or to the public collections on the Explore tab.

Explore memories

Medium task

The user taps on the Explore tab, taking them to a curated stream of memories based on their most recent memory's related location and song. The user can use the search bar for specific locations and songs if they so choose. Users can view memories that appear in these location/song-specific collections (see **View a memory**).

Task flows

Create a memory

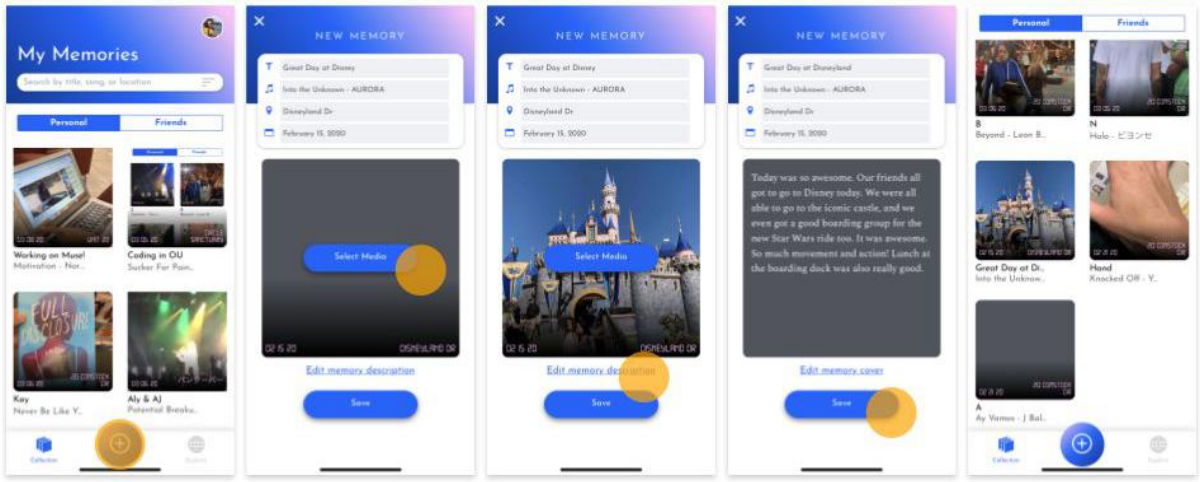


Figure 2. Task flow for creating a memory.

View a memory

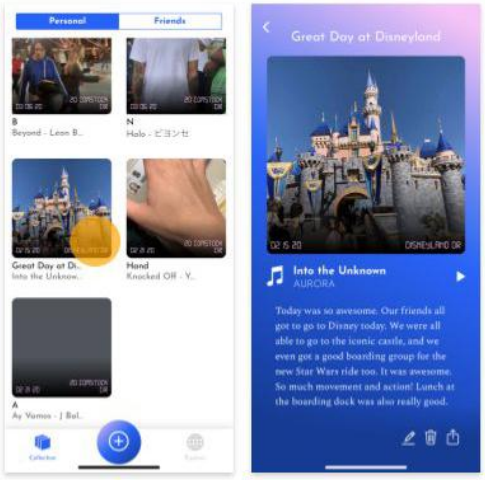


Figure 3. Task flow for viewing a memory.

Share a memory

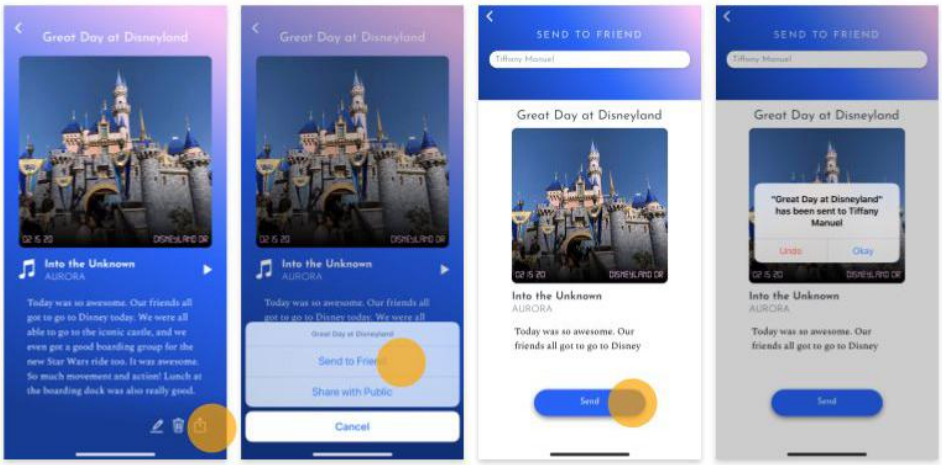


Figure 4. Task flow for sharing a memory with a friend.

Figure 5. Task flow for sharing a memory with a public collection.

Explore memories

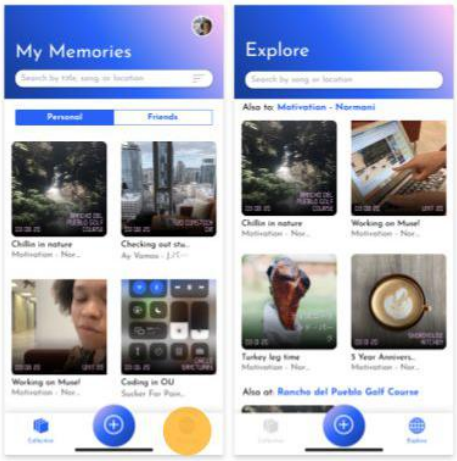


Figure 6. Task flow for exploring memories.

Design evolution

Muse went through many changes, not least of which a significant name change from Rechords to Muse. From here up until the second high-fidelity prototype, the project was known as Rechords throughout CS 147—regardless, we will continue to refer to the project as Muse throughout all points.

Sketches + low-fidelity prototype

The design process began with initial sketches of different ways we could execute our idea. One of them utilized a smartwatch interface, which we converted into the mobile app interface that you see in Figure 7 below.

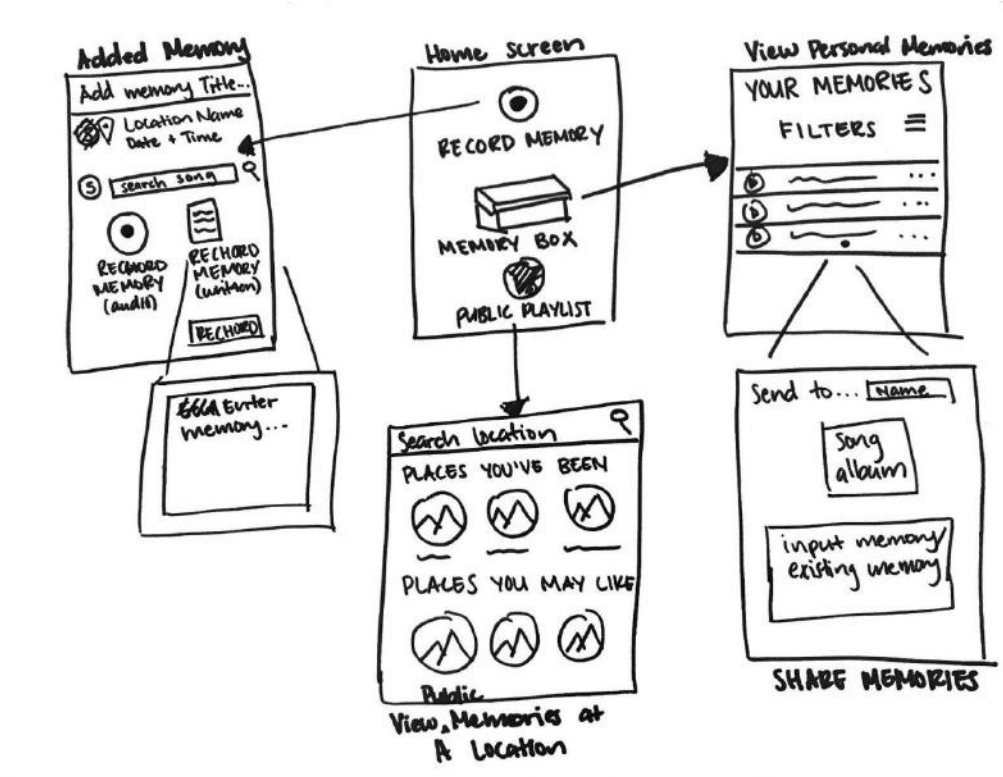


Figure 7. Sketches for wearable watch prototype.

We were drawn to this portable concept because it allowed a greater range of options for capturing memories (by audio, text, search, and Shazam) while also increasing ease of filtering memories for easier access. From this initial UI, users could input all of the information related to the memory that they are trying to record (top left). After recording a specific memory, users could look back and reflect on them through the Memory Box tab (top right), and also access others' memories at a location through the Public Playlist section (bottom center).

From this concept, we translated the UI into a low-fidelity paper prototype intended for a mobile application. The following are the task flows for the lo-fi prototype in detail.

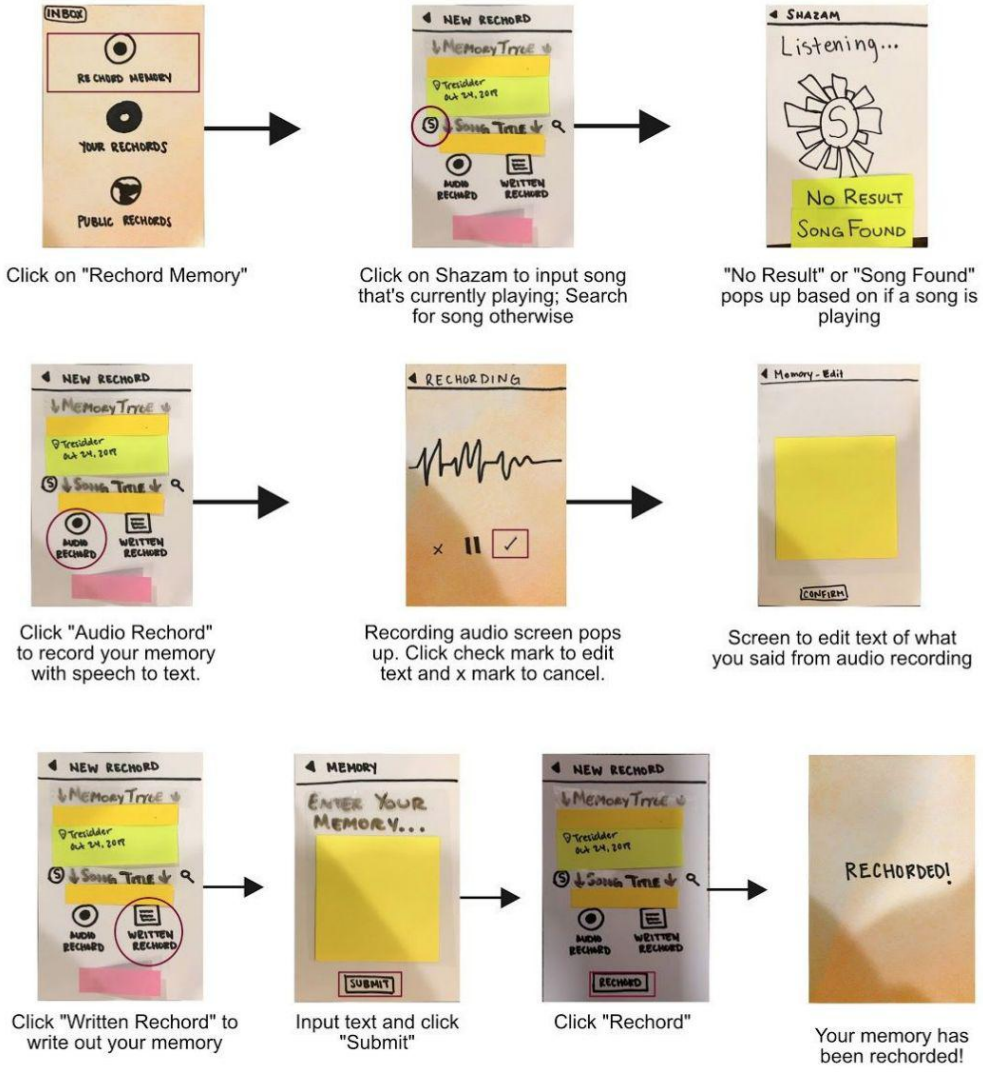


Figure 8. Task flow for creating and saving a rechord (memory).

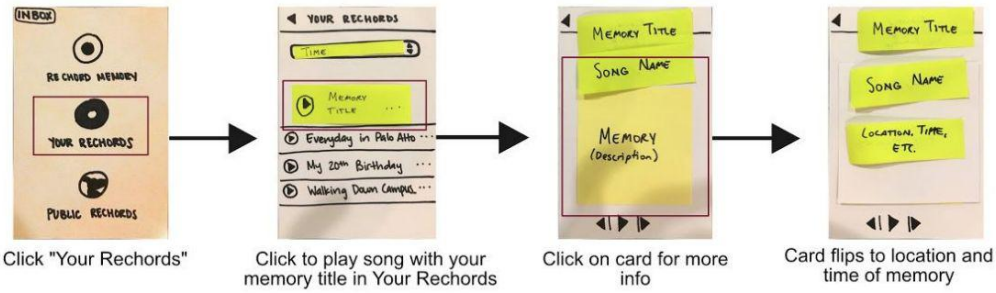


Figure 9. Task flow for viewing a recently made memory.

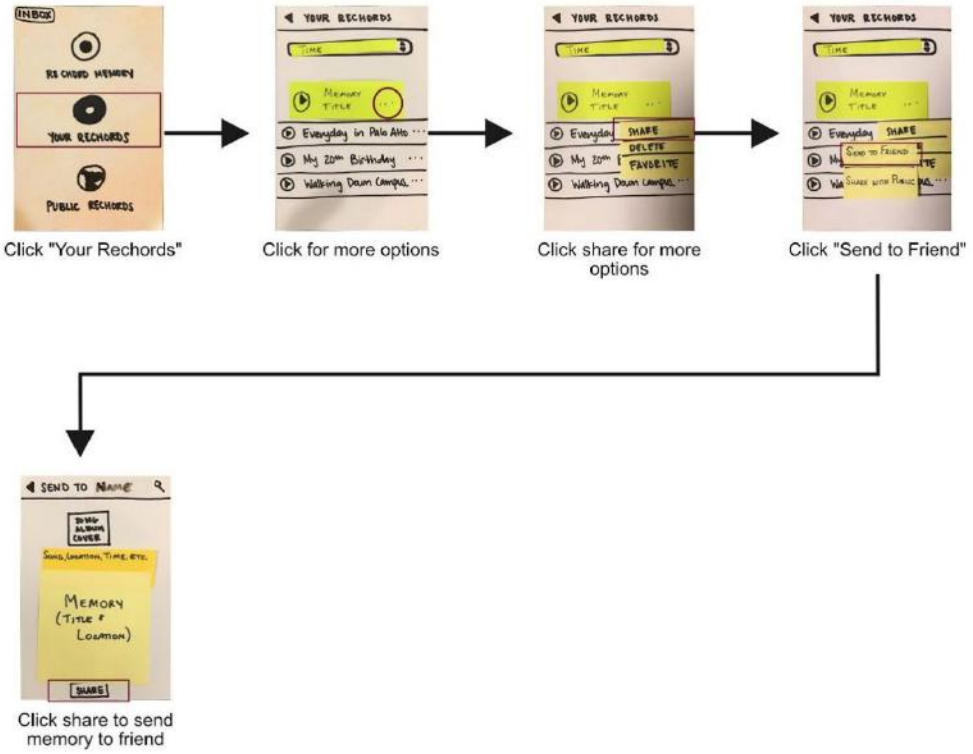


Figure 10. Task flow for sharing a memory with a friend.

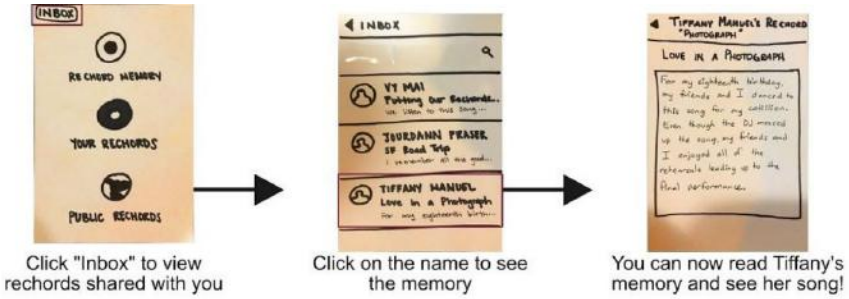


Figure 11. Task flow for viewing a friend's memory.

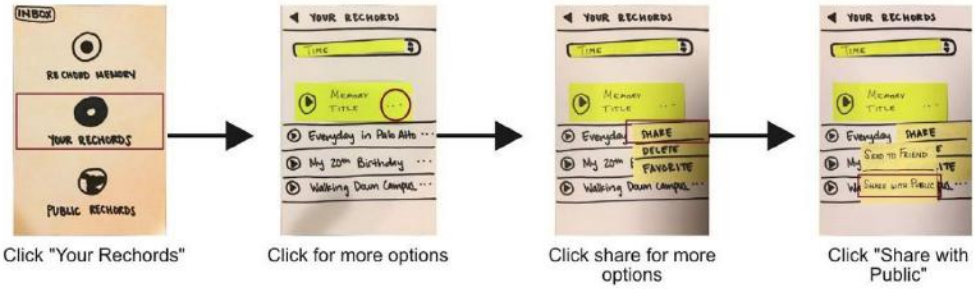


Figure 12. Task flow for sharing a memory to the public.

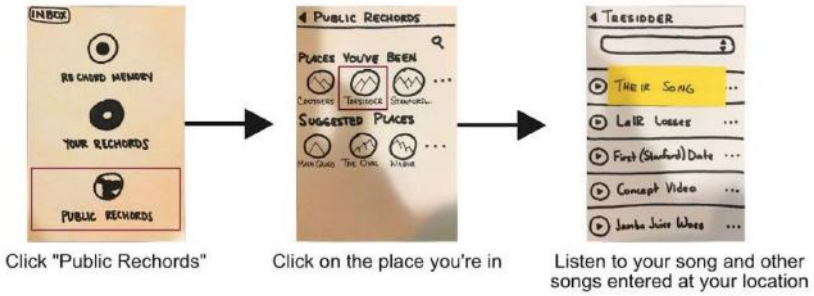


Figure 13. Task flow for viewing public memories from your current location.

From here, we conducted a pilot usability test.

Usability testing feedback

The pilot usability test we conducted from the lo-fi test brought out a lot of relevant points to consider. Our results from this test are summarized in Figure 14 below.

Participant #	**1**	**2**	**3**
Create + save a memory	2:30	5:00	5:07
View a memory you created	1:40	0:30	3:00
Share a memory to a friend	1:40	2:55 ****incomplete*	2:10 ****incomplete*
Share a memory to the public	0:20	2:30 ****incomplete*	1:09
View a friend's memory shared to you	2:30 ****incomplete*	0:45	1:34
View a public memory at your current location	0:10	0:20	1:05

Figure 14. Task execution timing from usability testing.

Even though the testers like the idea of the app, they became frustrated with the UI and process of creating a memory. Participants seemed less excited about the potential functionality of our app and more focused on simply completing the task. Only one participant wanted to explore the app more in depth.

Based on this feedback, we adjusted our sketches as follows.

Transforming the inbox into a Friends tab

Figure 15. Sketches of change from inbox to friends' memories tab.

Testers didn't like the idea of an inbox because it was too formal and hard to find, leading to the following changes: getting rid of the inbox on the homepage, moved friends' memories from inbox to My Rechords, replaced the Search feature with a Sort By feature, and used a dropdown to view descriptions.

Redone flow for creating a new memory

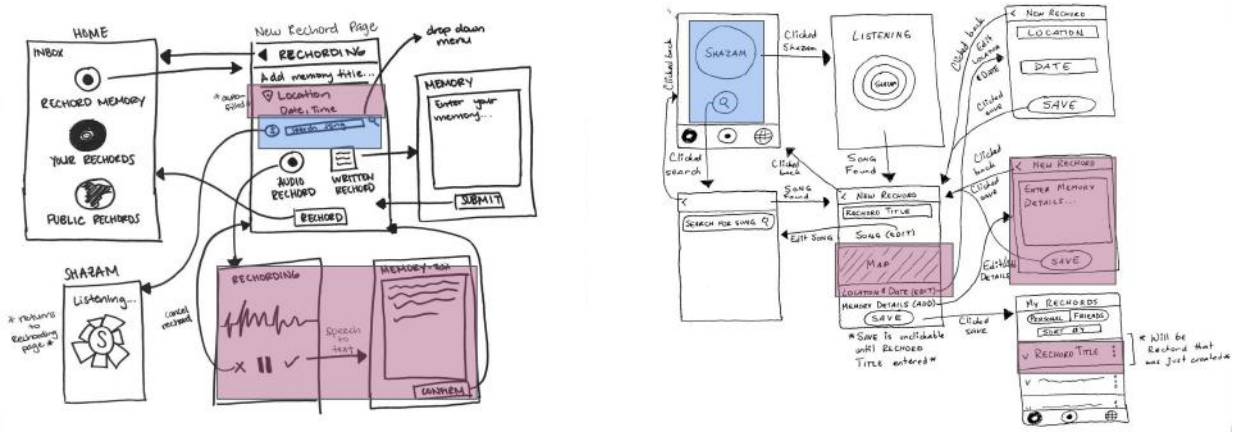


Figure 16. Sketches of change in new flow for new memory.

The old process for creating a new memory was difficult for our testers to navigate. They thought capturing a song first was more intuitive, so we made the following changes based on that feedback: made creating a memory the default home screen, captured a song with Shazam or search first, added a location map of location added, allowed potential for speech-to-text via keyboard, made memory details optional, and once saved, automatically takes you to the memory you just created in your personal playlist.

Changing the navigation

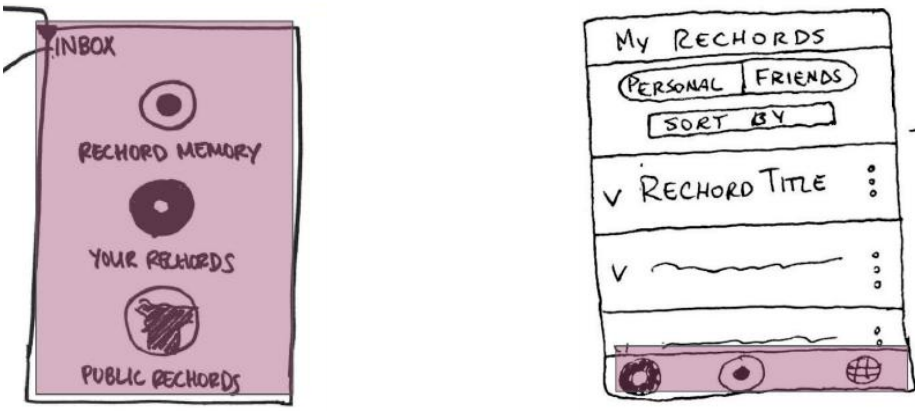


Figure 17. Sketches of change from screen menu to bar navigation.

We decided to make this change to make it easier for the user to navigate between different screens. Additionally, the change emphasizes the main function of our app: creating and saving personal memories.

Medium-fidelity prototype

The following are the task flows of the medium-fidelity prototype in detail, taking into account the changes we made during usability testing.

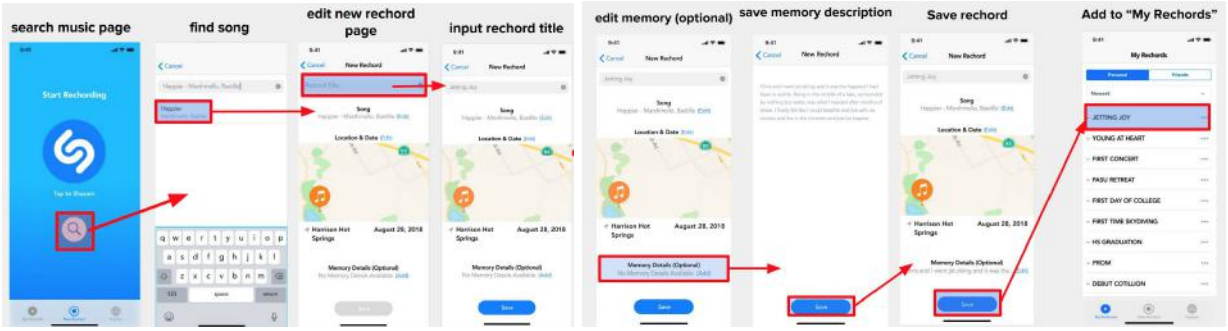


Figure 18. task flow for creating and saving a memory.

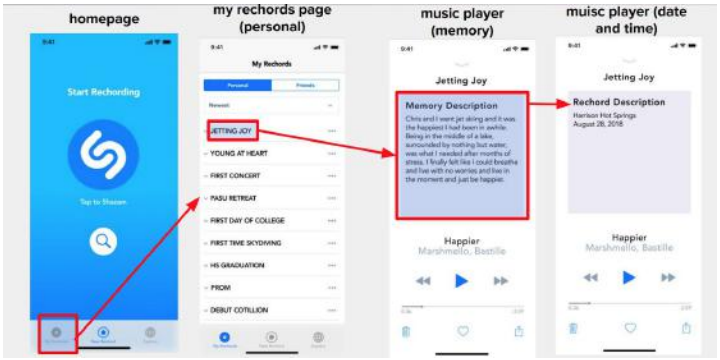


Figure 19. Task flow for viewing a memory.

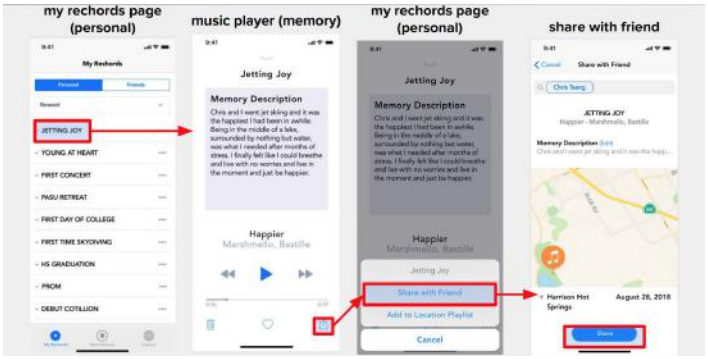


Figure 20. Task flow for sharing a memory to a friend.

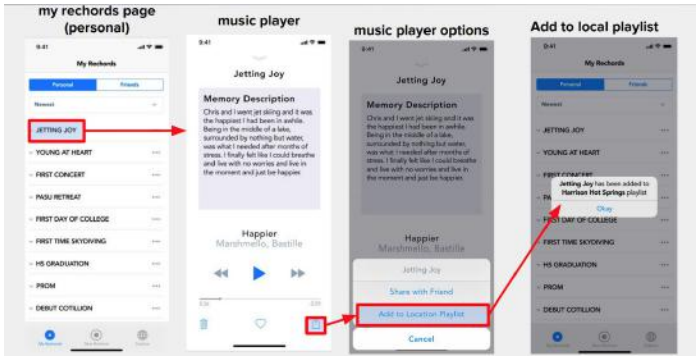


Figure 21. Task flow for sharing a memory to the public.

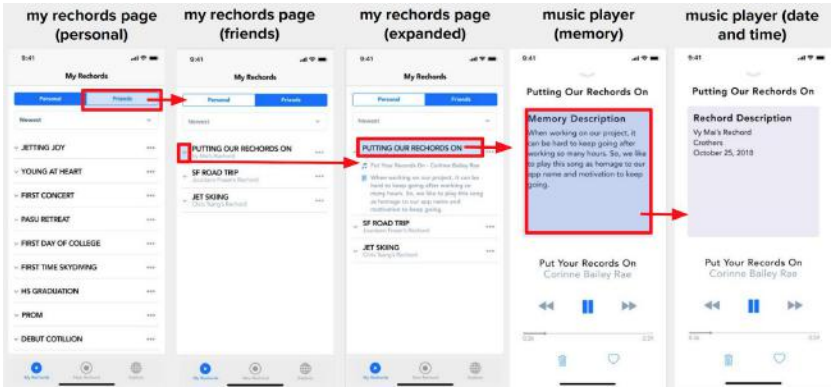


Figure 22. Task flow for viewing a friend's memory shared to you.

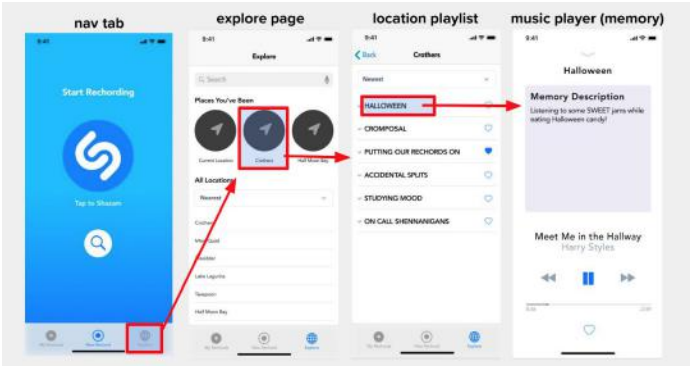


Figure 23. Task flow for viewing a public memory at your current location.

From our medium-fidelity prototype, we went into a heuristic evaluation, which would suggest even more changes come the high-fidelity prototype.

High-fidelity prototype #1

From our heuristic evaluation, we addressed a few problems with our med-fi prototype and adjusted accordingly.

Moving from playlist design to album design

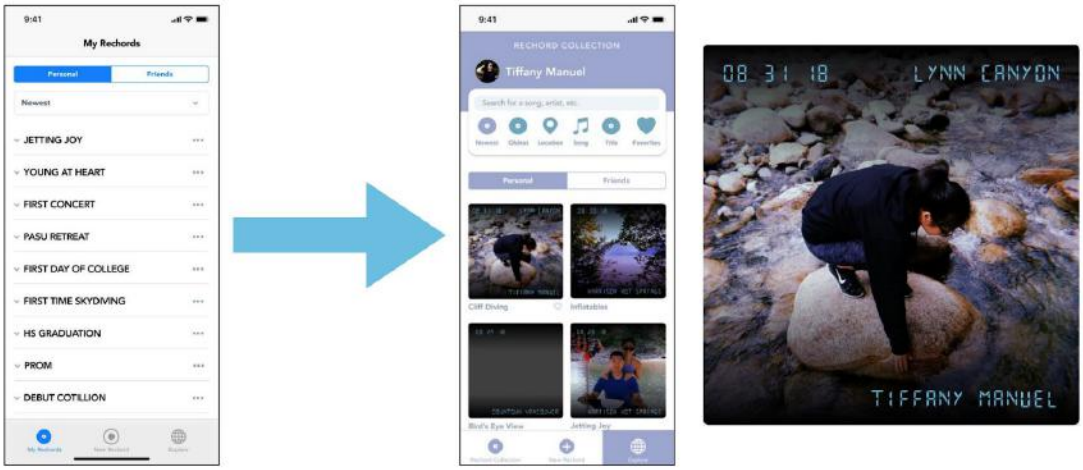


Figure 24. Change from playlist to album cover design.

Since we were already pulling information like photos, location, and songs, it no longer made sense aesthetically or heuristically to keep the playlist organization. We moved to the album design to emphasize the multi-dimensional facets of memory Muse was emphasizing.

Changing the New Memory screen

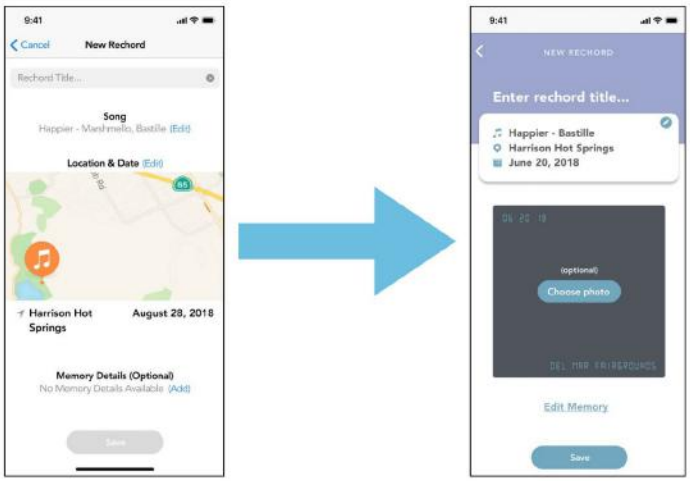


Figure 25. Change from location-focused to album-focused new memory.

In applying the new album-focused design across the app, we took this opportunity to revise the new memory screen, grouping required information together at the top and optional information at the bottom. We also took the time to introduce some whimsy by the album flip animation when the memory description is updated and edited.

Changing the Explore screen

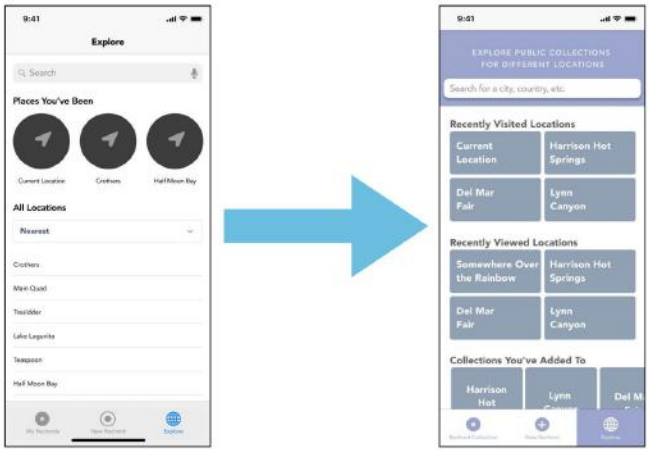


Figure 26. Change from location-dependent playlist to general explore tab for collections.

In addition to the New Memory creen, we also applied the redesign to the Explore tab to make it consistent with the feel of the rest of the app. We feel that this new Collections view is easier to look at and less cluttered.

Streamlining the View Memory screen

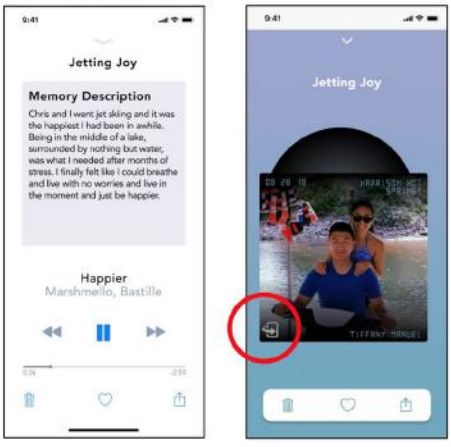


Figure 27. Addition of a flip icon to reflect flippable album cover.

Most critically, users could not find the memory description; we added a flip icon to the cover to remedy this issue.

High-fidelity prototype #2

After our first high-fidelity prototype, we did a round of lab usability testing; our second high-fidelity prototype addressed issues we found during these tests. In addition, it's here that we pivoted from Rechords to Muse in full and started to introduce the new design aesthetic.

Revising the My Collections screen

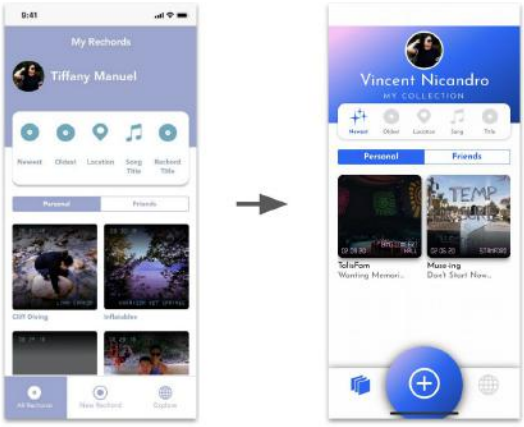


Figure 28. Restructuring of the My Collections screen.

The My Collections screen is now the de facto home tab for the app, now that the process for creating a new memory is different in that we no longer have users Shazam the song. Note the change in colors to reflect more active and bright contemplation, the new icons, and the redesign of the album covers, as well as adding the song just below the memory title. In addition, information hierarchy is redone such that the top header is much more compressed and takes up less space, allowing for more room for memories.

Tweaking the New Memory screen

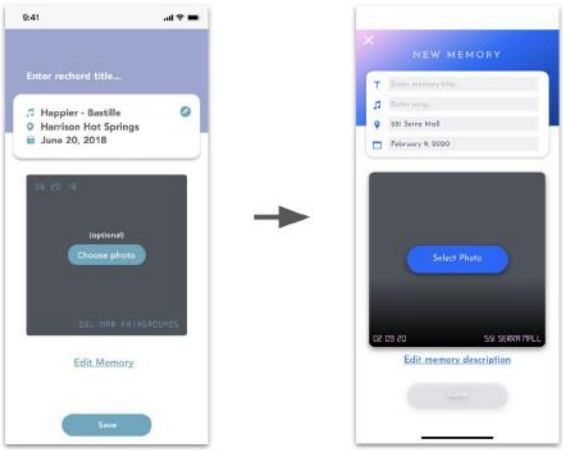


Figure 29. Restructuring of the My Collections screen.

The biggest change here in addition to the aesthetic updates is the moving of the Title field from outside the info box to inside, as our lab test participants consistently missed the title in the previous screen. Otherwise, all other things remain relatively the same, albeit one thing: how users get to this screen.

Revising the New Memory task flow

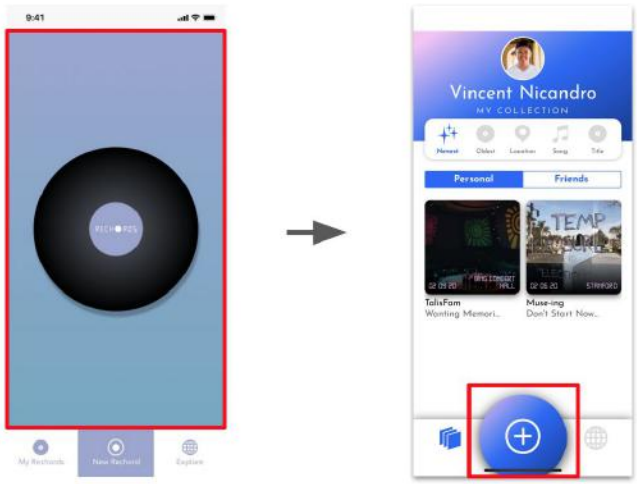


Figure 30. How to get to the New Memory screen.

We see that originally, New Memory took users to a page where they had to Shazam a song - our users didn't see this as intuitive from testing, so instead we have users create a new memory now by clicking the large plus button at the bottom of the screen.

High-fidelity prototype #3

Between our second and third hi-fi prototype, we did a round of field usability testing; our third high-fidelity prototype addressed issues we observed during these tests plus adding critical functionality that was still missing.

Final revising for the View Memory screen

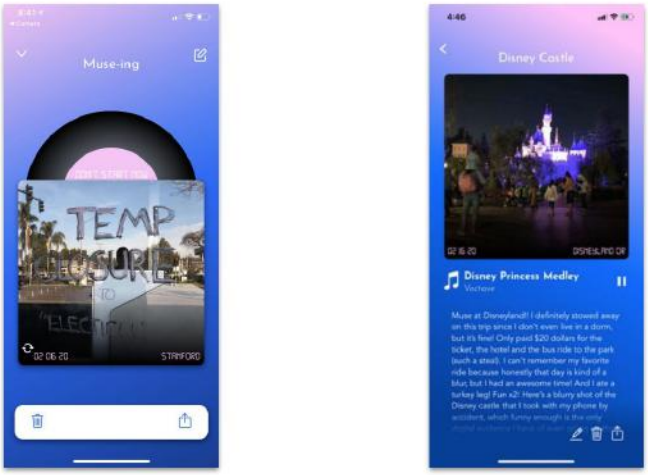


Figure 31. Moving away from album flip to memory view.

Most users were confused with how to view the content in a memory because it was obscured in the flip interaction. As a result, we made the difficult decision of removing the album flip; the screen now displays the cover, song, and memory as separate components to interact with. We feel we compensated the loss of the album flip whimsy by introducing the opportunity for video album covers, which add a level of kinetic energy to the app.

Populating the Explore screen without hard-coding

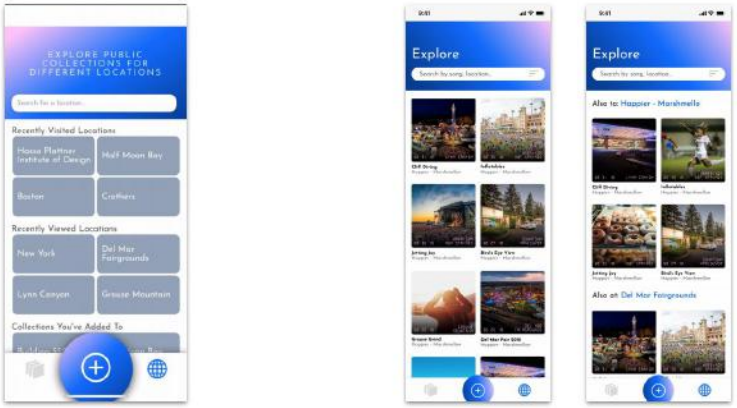


Figure 32. Difference between hard-coded locations to true Explore page.

We noticed that many users didn't know how to interact with the Explore screen because there wasn't enough reason to, largely owing to the fact that the Explore screen was ugly and hardcoded. To rectify this, the screen is now populated with the 10 most recently shared memories from the global community, and updates to relevant memories based on your location and chosen song.

Playing music through Spotify

In the most exciting development, the app can now play music! We've changed the user flow for logging in by having users log in with Spotify, which opens the app up behind the scenes of Muse. When users click on a memory, the song starts playing automatically behind the app. This tweak was perhaps the most important, as it increased the fidelity of the app to its intended purpose - turning music into memories.

Cleaning up the My Collections screen

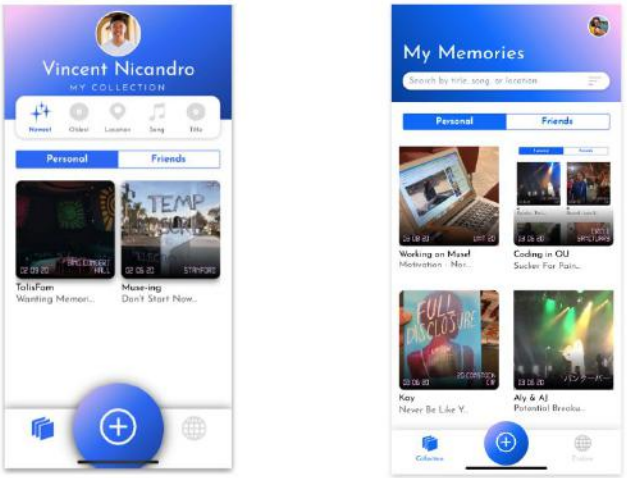


Figure 33. Changes in the header and bottom navigation bar.

With the help of instructors, guest speakers, and peers alike, we streamlined the design a bit more of the home screen, reducing the size of the navigation bar dramatically by reducing the size of the New Memory button, adding labels, and dropping the filters into the search bar functionality. Profile photo prominence was also decreased substantially.

Evaluation techniques

Of the techniques we conducted over the past two quarters, by far the most effective was the field usability study. Our team was incredibly intentional in casting a wide net both in and out of Stanford (even conducting tests while at Disneyland!) and in doing so we were able to get a lot of helpful feedback across demographics related to our app. Most interestingly, we were able to get this feedback while we were in situations that were most conducive to using the app anyway (think Disneyland again!), which resulted in feedback we could trust for the most part.

It was in the field study that we found out that what users could put in an album cover correlated somewhat to how interested they would be to using the app; that insight alone drove the research into adding support for video album covers, which adds so much to the kinetic nature of Muse now. We also learned a lot about how people intended to use the app (the Shazam feature we'd been toying with was a plus, but not integral; people wanted to explore memories by both song and location), and participants were happy and willing to give feedback.

Final interface

The final interface for Muse allows users to create new memories which are associated with a song, date, location, and entry. These memories can then be looked back upon for further reflection, or be shared to a user's friends and the public.

Users can view their friends' memories that are shared to them; in addition, users can go to the Explore tab and find memories at locations and with songs they desire. When users view a memory, they can play the song by clicking the play button, which will take them out of the app and into Spotify to play the song.

Greater detail about the final interface can be found by looking at the Tasks and Task flows sections.

To be implemented

In terms of friend management, we've yet to implement a more robust form of adding a friend beyond searching for their name and adding them one-way. We thought the feature wasn't relevant to our main tasks, and so it stayed at a relatively medium-fidelity level; in the future, we hope to make it more robust through the use of friend requests, confirming friends both ways so unsuspecting users aren't spammed with memories from users they don't know.

Additionally, we'd like a more advanced means of playing the song by having the song play without leaving the app. Spotify API makes this particular point difficult, and so while we were able to have the song play in the app background through Spotify, it was quite finicky and ultimately not at a point we could release the app faithfully on the App Store. We compromised by having the app take us to Spotify to play the song, which isn't ideal, but users can still go back to the app and read their memory while the song plays in the background.

Lastly, we want to explore the bounds of what an album cover could entail. We'd like to expand users' ability to add photos and videos to the memory cover by introducing slideshows for multiple photos as well as being able to upload GIFs.

Features left out

In addition to the compromises we made above (which we recognize as versions of features we implemented in part but left out on the whole), we also left out the more curatorial features related to Explore in favor of an automatically generated list of content that appears curated to the user. We see an opportunity for the Explore tab to have more curated content by an editorial team (see the business model section later in the report).

Hard-coded + WoZ

Nothing in the app is hard-coded or uses Wizard of Oz techniques. However, we pre-populated all content/memories into the app; all content was created by the team while on the app as opposed to pulling from a back-end database. We did this so that users who joined the app would actually have memories to explore; these memories are live and real, related to our team mates.

Tools

Muse's mobile application was built using React Native, Expo, and Firebase, with additional use of both Google Maps API (for automatic location detection and geotagging) as well as Spotify API (for music playing and its robust song database). On the design side, we used Figma to document and decide upon UI changes before implementing them into the mobile app. Lastly, we used GitHub to keep track of the different versions of the app while developing it.

React Native was particularly good for Muse as the core functionality of the app was already built onto React Native during the first half of Muse's development, chosen for its modularity and ease of learning, in addition to flexibility in cross-platform implementation. Half of our team were primarily in the role of app developer, so going into Muse through React Native was a bit of a learning curve but not terribly so.

Working with Google Maps and Spotify APIs was a bit of a pain, largely due to the time between when we last touched the app last year and when we picked it up at the beginning of this year. Seemingly intuitive things like playing music actually became a nightmare to pull off, and things that worked before (such as auto-detection for location) seemingly broke at some point, which was frustrating.

Expo was helpful in simulating the app, but was limited in that it could only really work as an option for Android devices as opposed to iOS, which was more involved and is bound to the code running on our laptop.

Firebase was especially useful for running our app backend; the information and content being hosted in the cloud through Firebase made it such that the app felt real and lived in (mostly because it was through our own use); nevertheless, like React Native, Firebase had a bit of a learning curve that took a bit of time to acclimate to.

For the other half of the team that focused on design (with input from the app developers), Figma was a fantastic online, cloud-based collaborative tool to make changes to the UI on the fly. What was especially helpful in developing Muse was that we did our best to have 1:1 parity between the Figma file and the React Native app, so any future changes we made on Figma would be like changing the current state of the mobile app.

Finally, Github was useful for ease of version tracking, allowing our app developers to work independently before combining and pushing.

Start musing

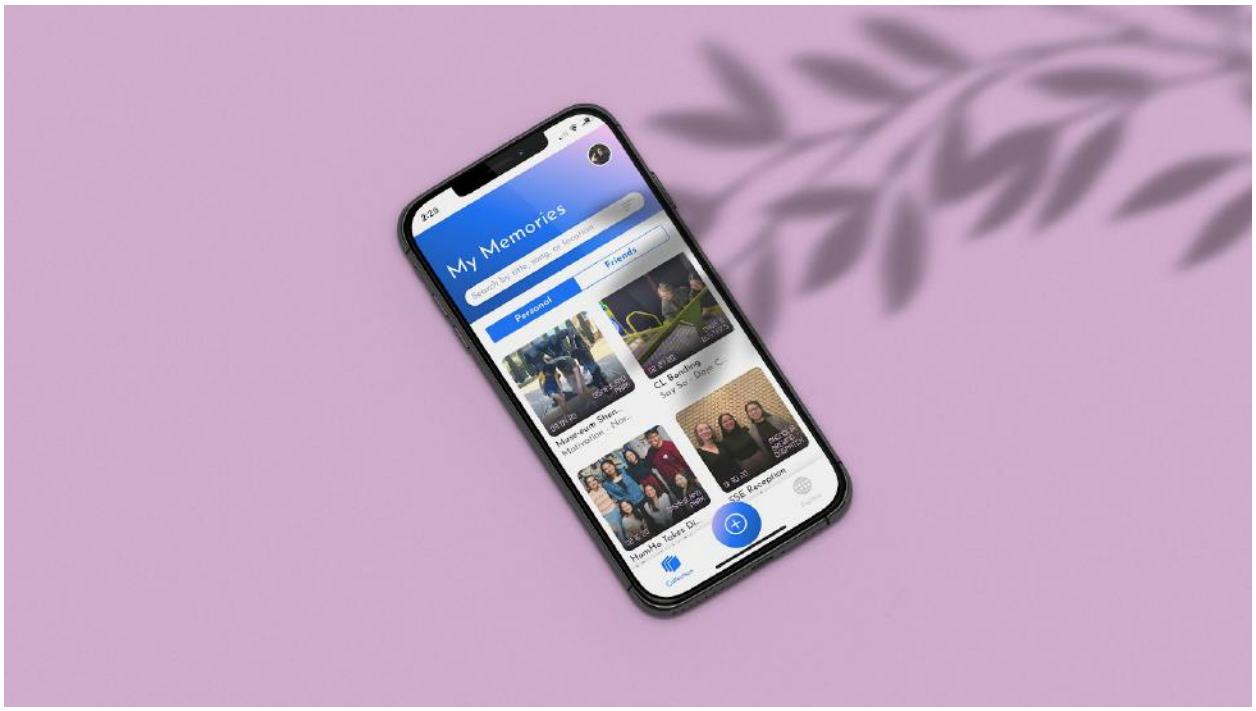


Figure 34. Muse.

We are currently enrolling in the Apple Developers program to put Muse on the App Store; however, we're currently experiencing delays through that avenue. In the meantime, try out our second high-fidelity prototype by following these steps.

To use our app on your **Android** phone, please use the following instructions:

1. Download the Expo app on your Android phone.
2. Go to https://expo.io/@tiffanymanuel/muse and scan the QR code.
3. Press the push notification that pops up.

Unfortunately, due to Expo's limitations regarding access on Apple devices, devices running iOS aren't able to run this prototype. However, you can still test our prototype by using the following instructions:

1. Go to https://expo.io/@tiffanymanuel/muse and click "Open project in the browser" (under the QR code).
2. *(Optional)* If you have an Appetize code, enter the code into the provided space.
3. Click "Open Project" to run the app in your browser.

Congrats! Go on and start Muse-ing.

Making it real

Meet the team

Alema F.
UX Designer

Tiffany M.
App Developer

Nylah D.
App Developer

Vincent N.
UX Designer

Alema Fitisemanu is a junior studying Science, Technology and Society, with a concentration in Human-Computer Interaction, with a minor in Art Practice. His work includes UI/UX design, visual design and graphic design. In the future, he hopes to design awesome educational experiences for Pacific Island youth.

Tiffany Manuel is a senior studying Computer Science on the Human-Computer Interaction track. She has experience with UI/UX design, frontend development (primarily with mobile applications), and product management. She is currently looking for opportunities that allow her to further explore and utilize her design and development skills.

Nylah DePass is a coterm studying Computer Science with a concentration in Human-Computer Interaction. Her work experience includes backend, full stack, and mobile engineering. In the future, she hopes to design products that are humanitarian and accessible to all.

Vincent Nicandro is a senior and incoming coterm studying Computer Science and Classics at Stanford, concentrating in Human-Computer Interaction. He has worked as a UI/UX designer, web developer, and UX researcher, and he is interested in finding novel and exciting ways to share engaging stories with people.

Business model

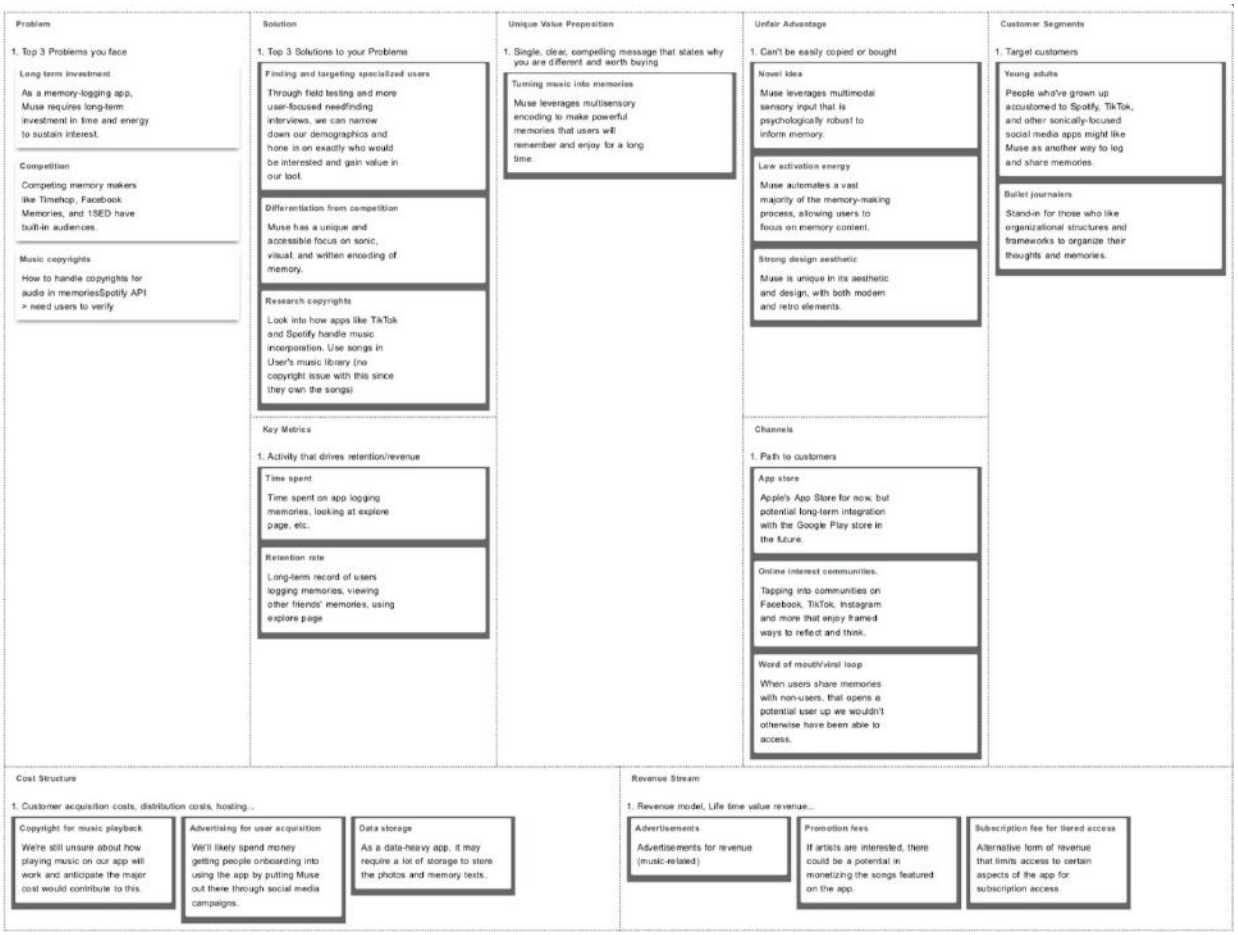


Figure 35. Lean business model, as constructed in Canvanizer.

Our intended customer base is young adults who would like to integrate more contemplation and self-reflection in their life. Muse is especially targeted to those exposed to Spotify, TikTok, and other apps which have users already acclimated to music-related content. In addition, we use the term "bullet journalers" to describe the population of people who would appreciate Muse - we believe that Muse's key appeal is its offering of an organizational structure for thoughts, memories, and emotions. With hundreds of thousands of users that use memory-logging and sharing apps like Timehop, Facebook Memories, and One Second Everyday, Muse has the opportunity to enter the market with a unique take on memory-keeping.

Our business model relies specifically on two things: a tiered membership plan and opportunity for sponsored content. With regards to the subscription plan, Muse would always remain an app that is free to use - however, specific features like having slideshow and GIF album covers for memories or exporting your memories' songs into a playlist would come with an additional membership. Sponsored content is where we expect a majority of our revenue to come from, particularly with the potential for companies to sponsor curated memories from their locations

or for artists to sponsor curated memories from their songs in the Explore tab. In addition, we would also like to host and sponsor music events of our own in the future, bringing artists to users to boost the creation of new memories.

In the long-term, we believe that Muse has the chance to bring self-reflection back into the lives of young adults in a fresh and fun way. We all make so many memories each and every day of our lives; we hope that Muse can be a means to make those memories stronger, creating records for us to look back on and enjoy.

Summary

Muse interweaves music with memories to make memories a delight, addressing everyone's need to reflect and contemplate quickly and easily. We often remember memories through more than one sense, but current forms of memory-keeping are often limited to one-dimension. Our key impact is changing the perception of self-meditating and self-reflecting within young people, leveraging the power of music to help people remember stronger and more vividly.

Figure 36. Design evolution of Muse.