

## Hi Fi Prototype



### Rechords

Turn Music into Memories

**Jourdann F.**

*Web Developer and UX Designer*

**Tiffany M.**

*Mobile App Developer and UX Designer*

**Vy M.**

*Mobile App Developer and UX Designer*

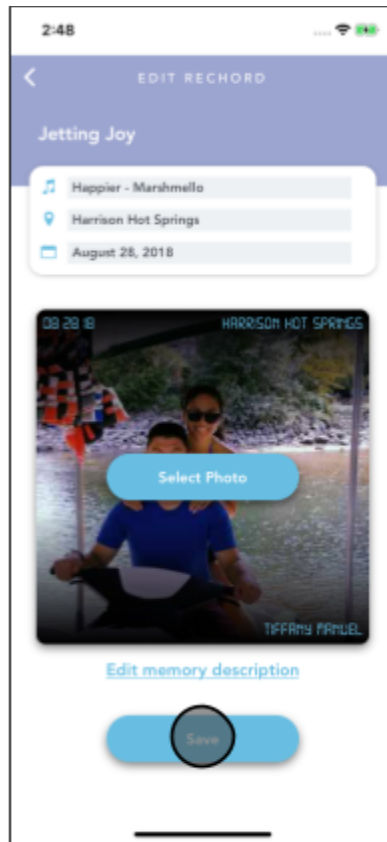
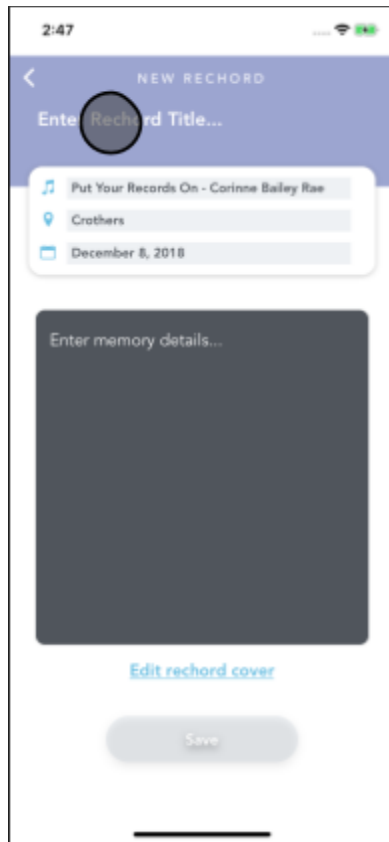
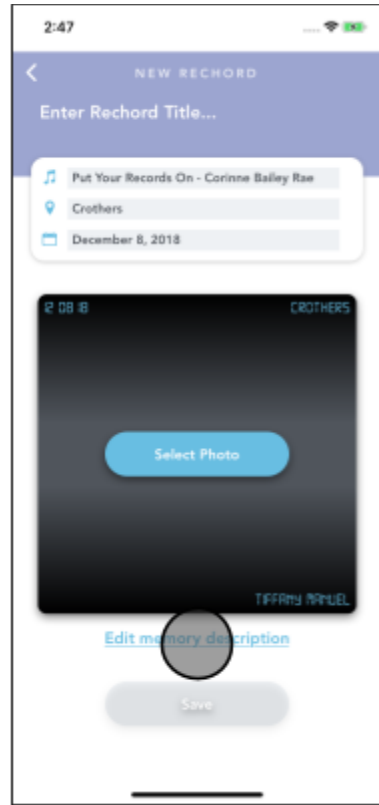
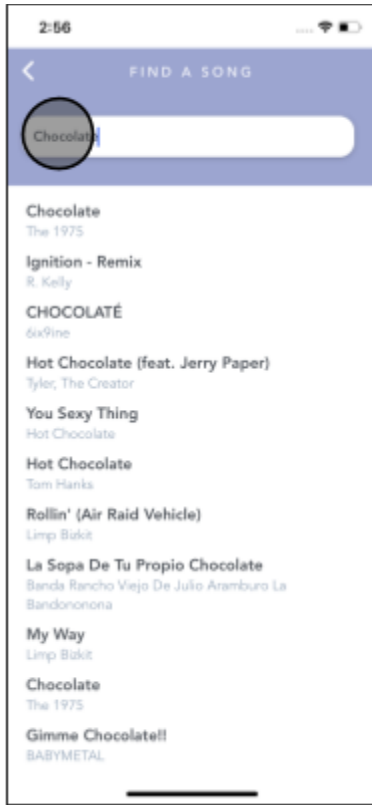
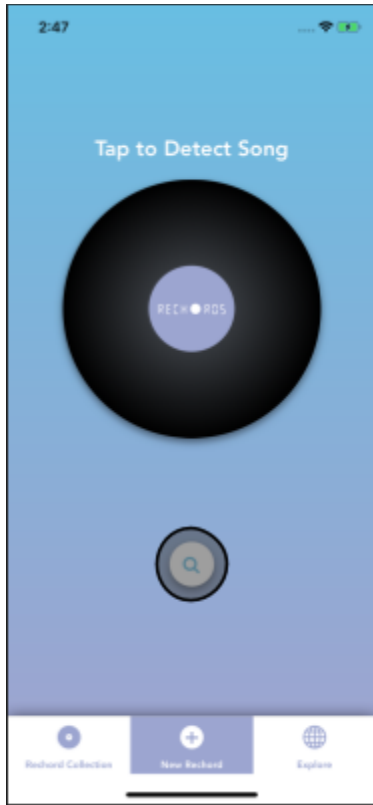
## Problem/Solution Overview

When it comes to memory-keeping, journaling is time-consuming, and photos only capture visual details. We aim to recreate the environment of the memory itself and streamline the process of memory-keeping. Rechords uses the connection between music and memories to give users a better way to remember.

## Tasks & Final Interface Scenarios

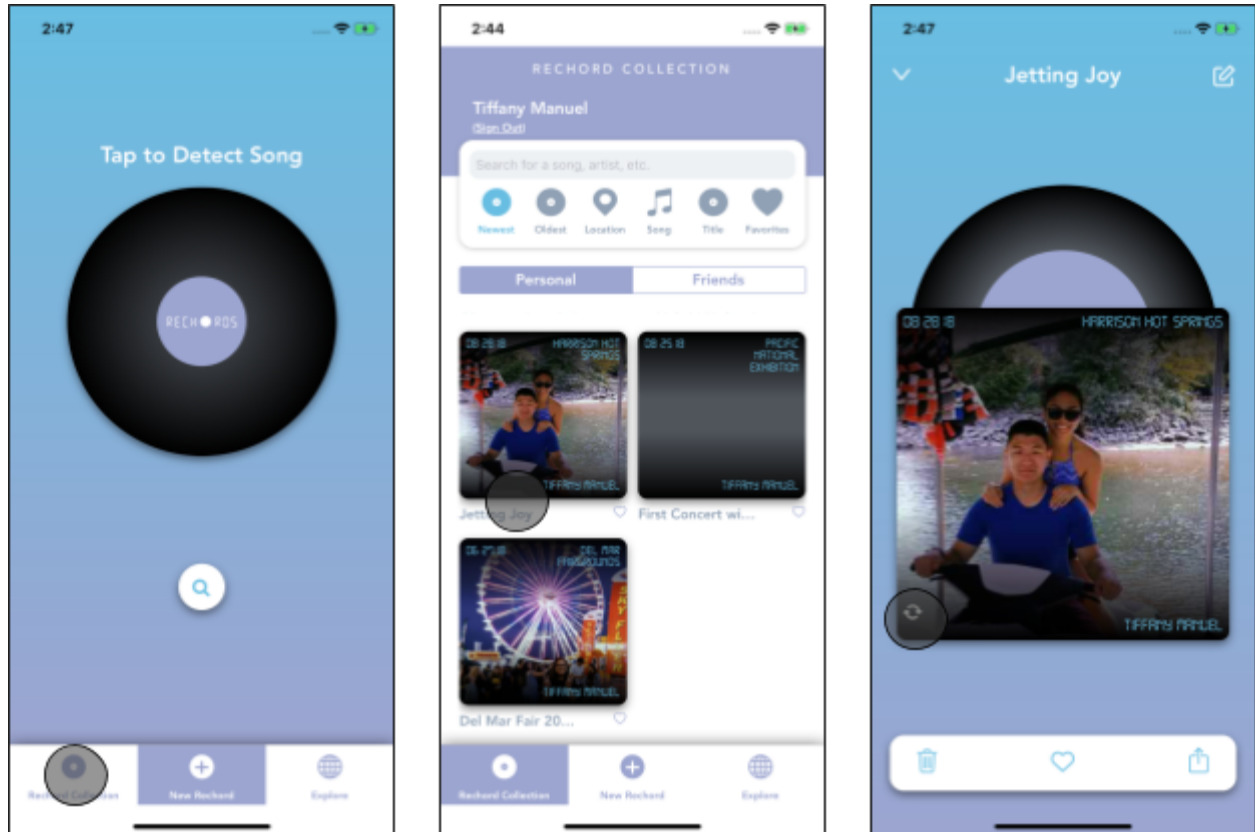
[Simple] Create and save your own rechord (memory and song)

The user taps on the Rechord logo to detect a song or the search button to manually search for a song. If the user decides to search for a song, a list of suggested song names will pop up as the user enters the name of the song. Once the user chooses a song, they can 1) edit the location and date and 2) add a title, photo, or memory description. As long as the user adds a title, they can click the save button to save the memory; the photo and details are optional. This task was chosen because it gives people a way to store their memories.



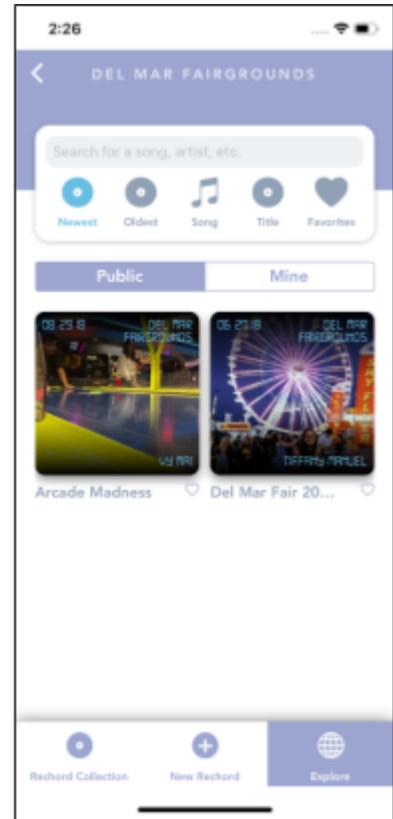
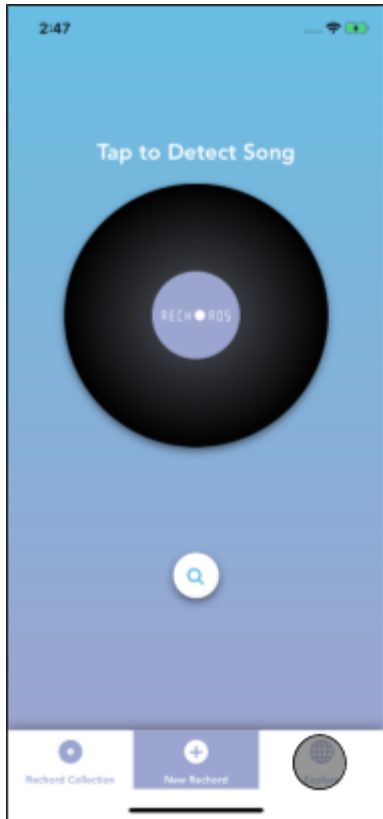
### [Simple] View your own records

The user taps on the Record Collection tab to get to a screen with all of their saved records. From here, they can click on any saved record and view the attached song. They can also view the details of their memory by clicking the icon near the bottom left of the record cover. This task was chosen because it gives the user a way to review their memories when they're in a reflective or nostalgic mood.



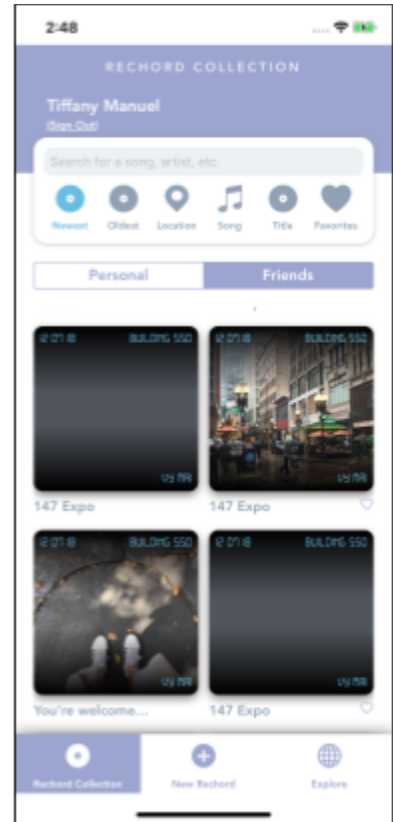
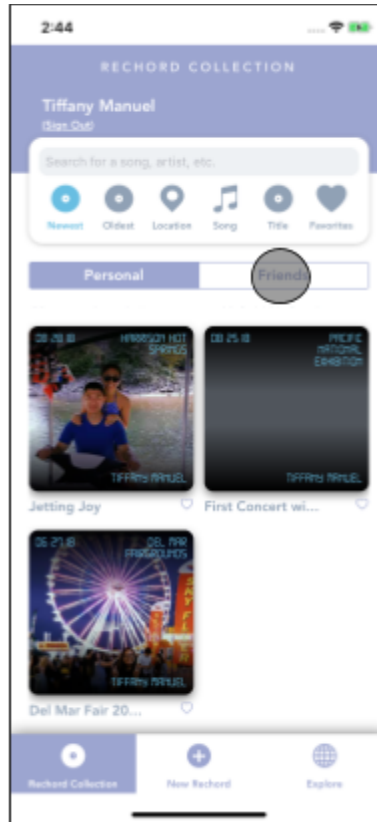
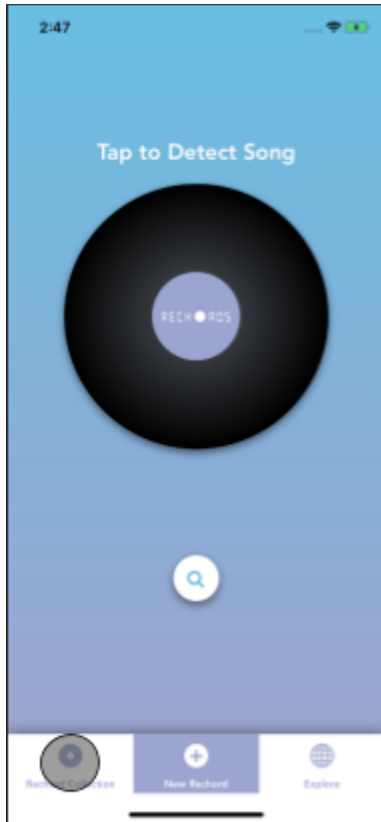
### [Medium] View records from a public collection

The user clicks on the Explore tab. From here, they can click on any of the locations on the page or search for a location. When they click on a location, they are shown all of the records shared by users for that location. This task was chosen because our users showed that they're interested in learning about others' stories that were made at the same place.



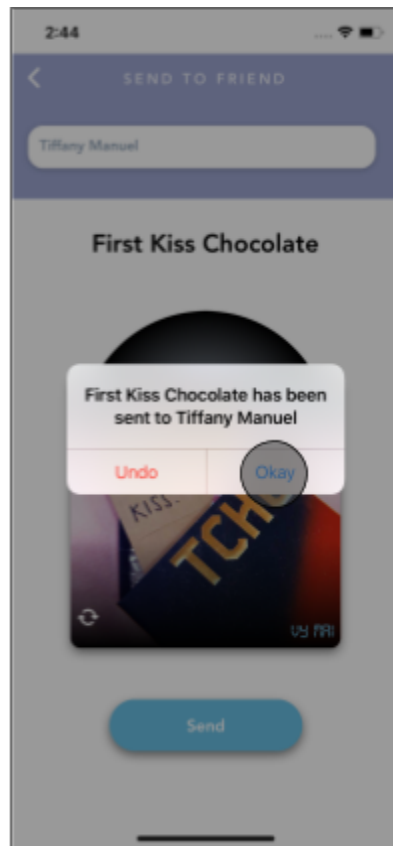
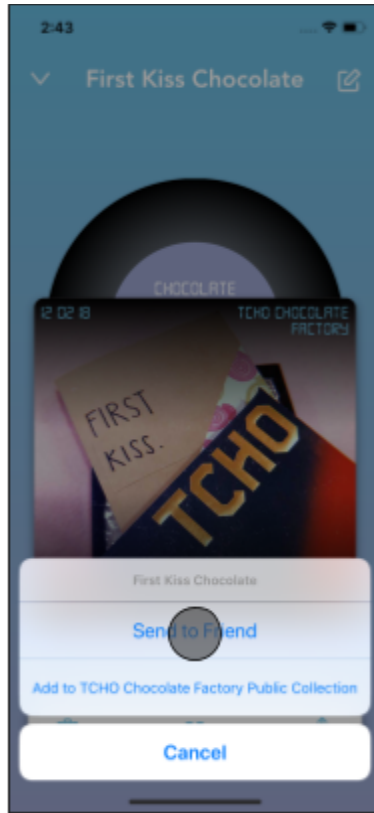
### [Medium] View records from your friends

The user clicks on the Record Collection tab. From here, the user can press the friends tab to view the records that have been shared with them from friends. This task is really important because memories aren't always experienced by one person. This task allows user to receive memories that they experienced with someone else.



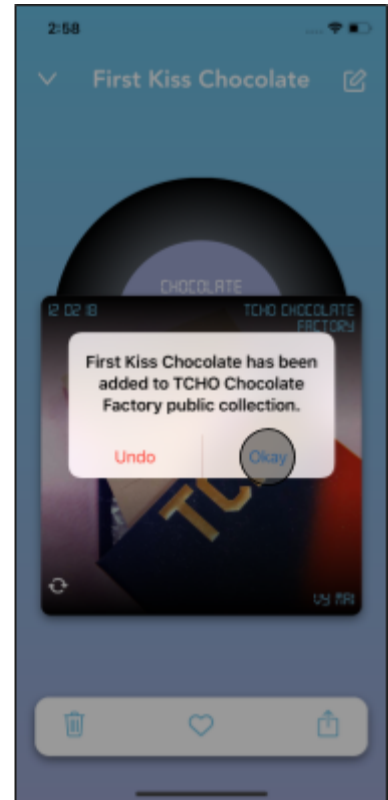
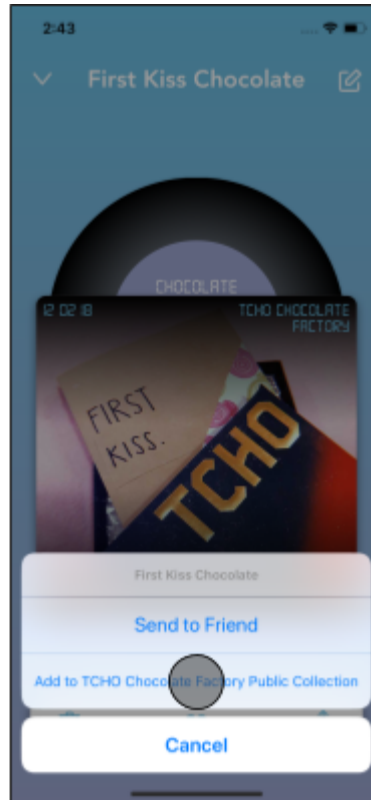
### [Complex] Share a rechord with your friends

After the user views their rechord, they can click the share button on the white action bar. An action sheet will pop up, and the user clicks on the Share to Friend option. The user then searches for a friend they want to share the rechord with and may review the rechord to make sure everything is correct. Once the user clicks Send, they are prompted to either undo or continue to send the rechord. Similar to the last task, this task is really important because memories aren't always experienced by one person. This task allows user to share memories that they experienced with someone else.



[Complex] Share a record to a public collection at a specific location

The user clicks on the share button on the white action bar. Then the user selects Add to [location name] Public Collection and is prompted to make sure the user actually wants to share the record to the public collection. Again, this task was chosen because it allows a user to share their experiences with others.



# Design Evolution

## Initial UI Sketch

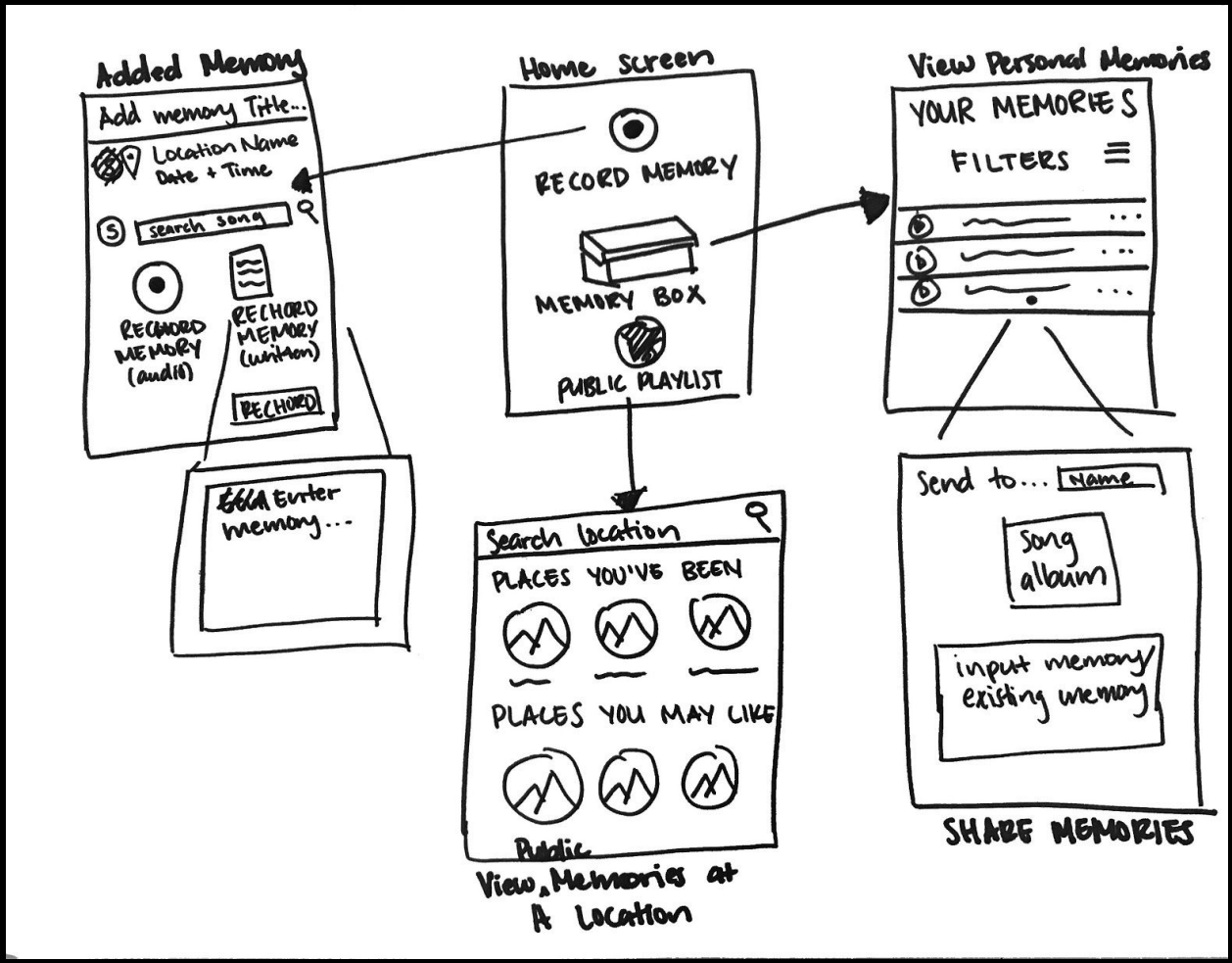


Figure 1: Initial UI Sketch



# Low-Fidelity Prototype

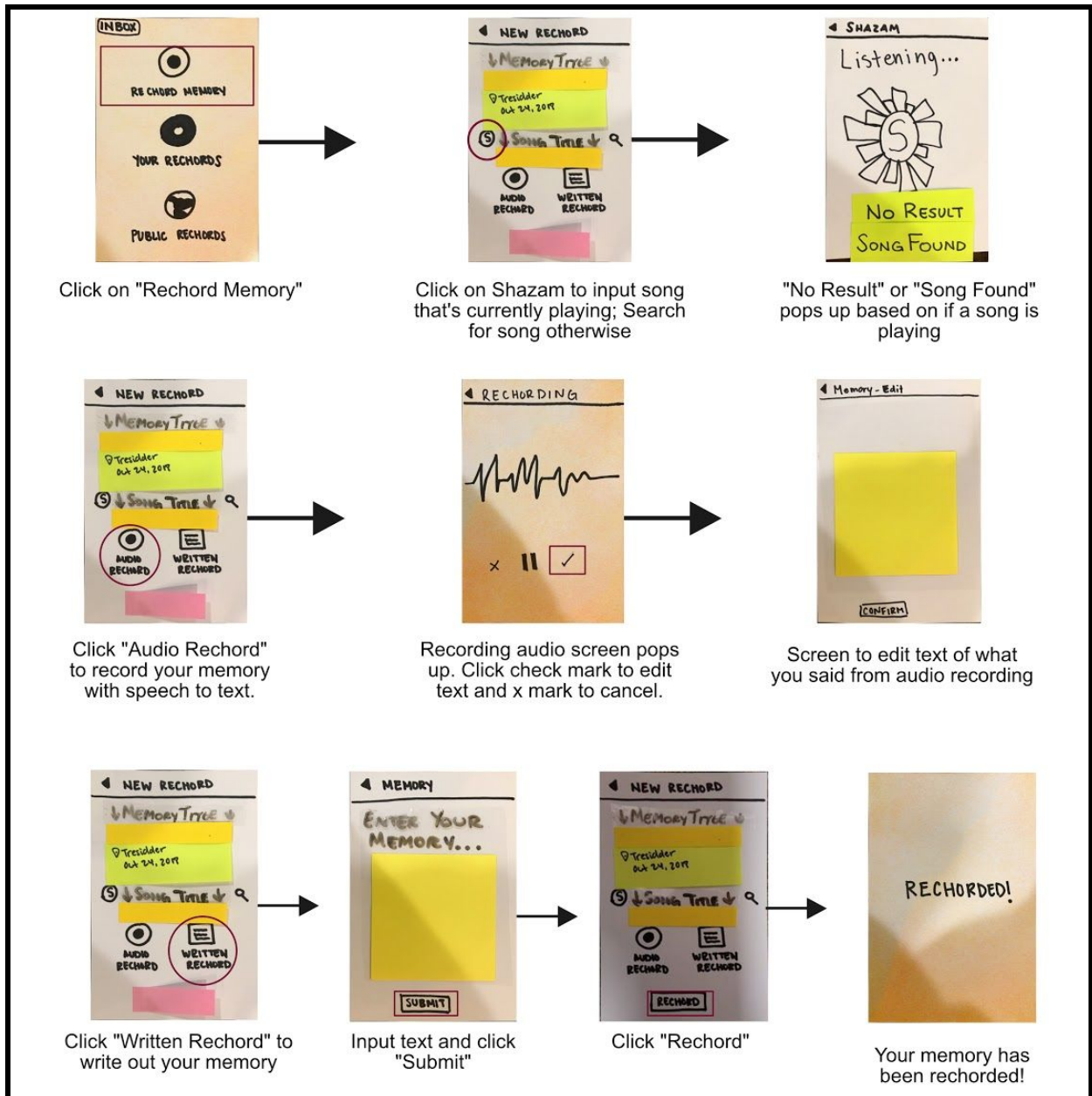


Figure 2: Low-fi Prototype - UI flow for first task (creating a rechord)

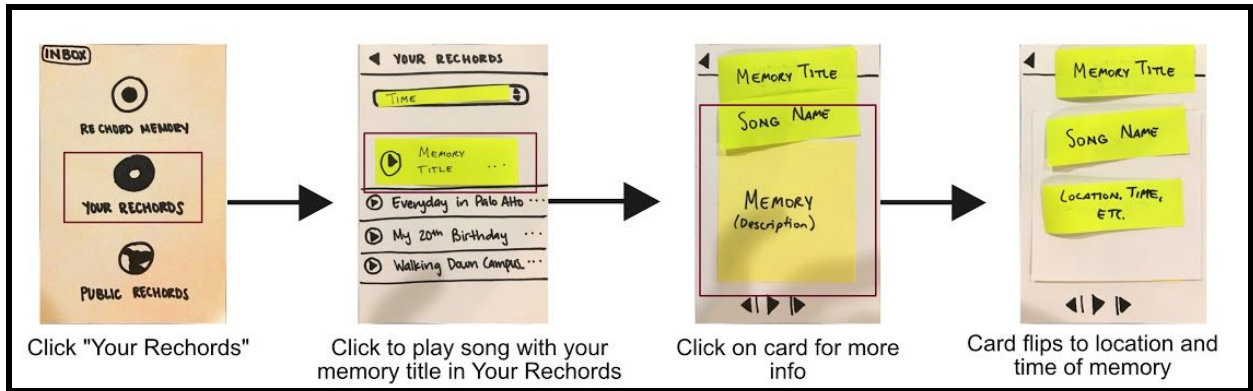


Figure 3: Low-fi Prototype - UI Flow for second task (view your record)

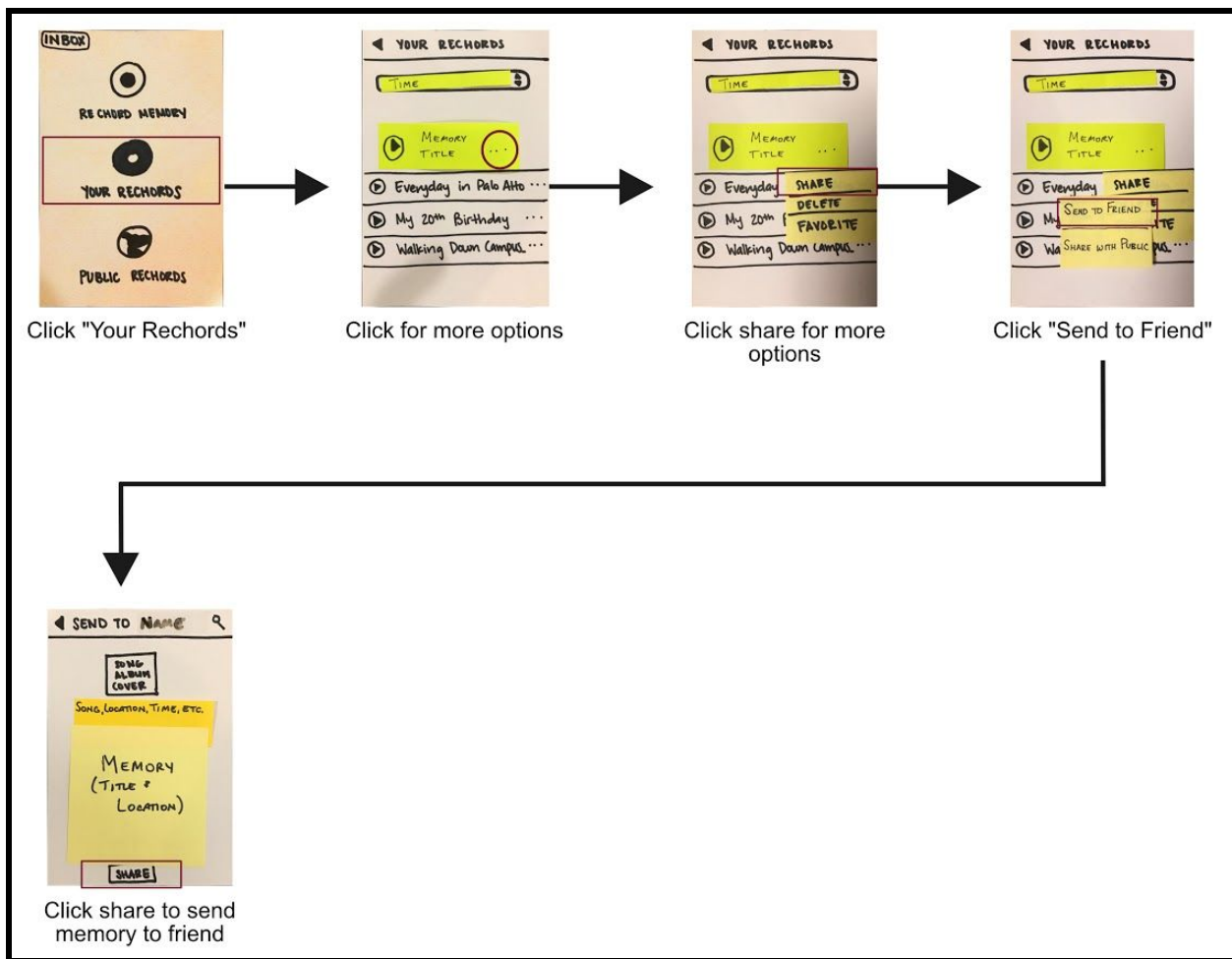


Figure 4: Low-fi Prototype - UI flow for third task (share record with friend)

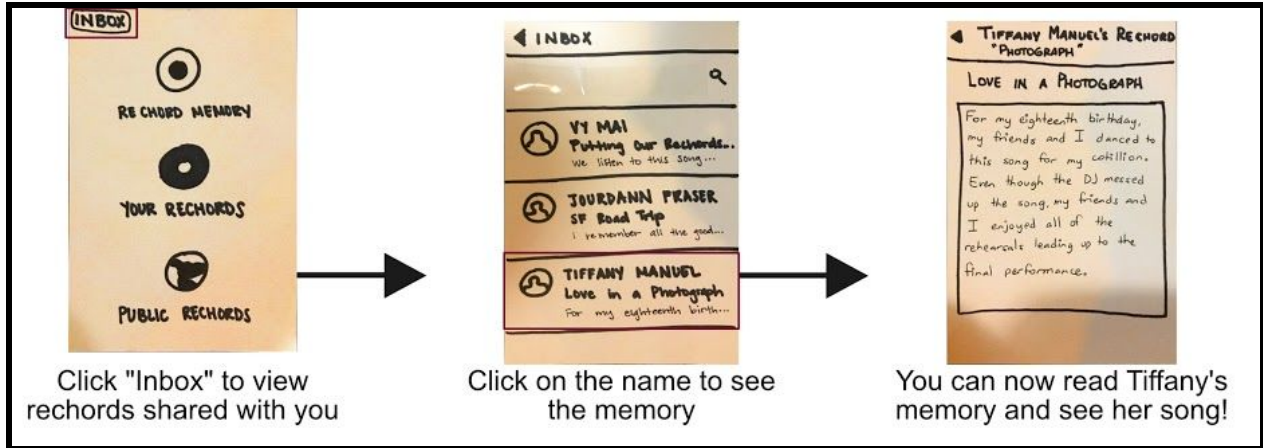


Figure 5: Low-fi Prototype - UI flow for fourth task (view a shared record from a friend)

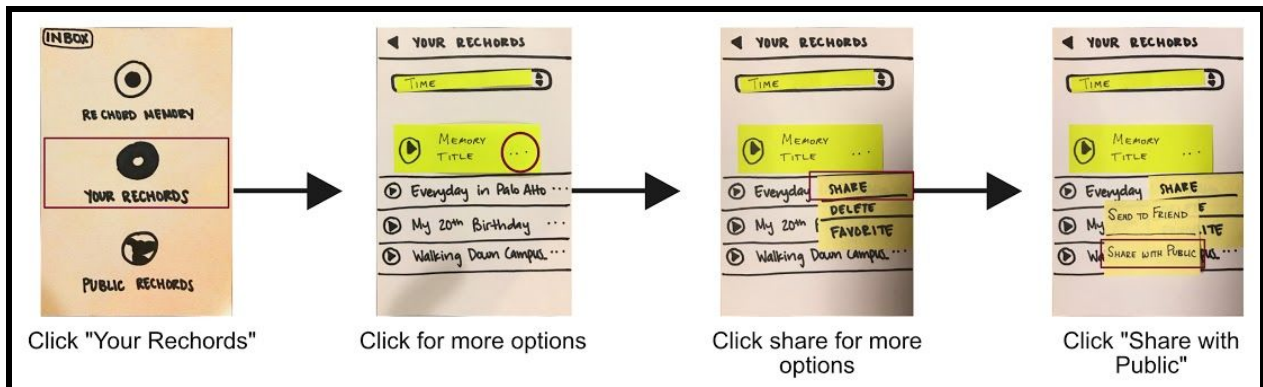


Figure 6: Low-fi Prototype - UI flow for fifth task (share record to public location playlist)

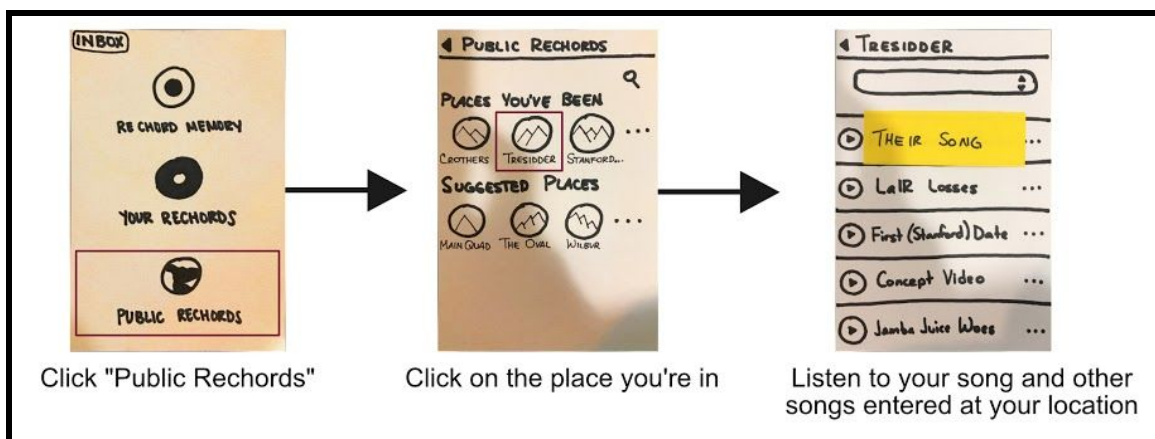
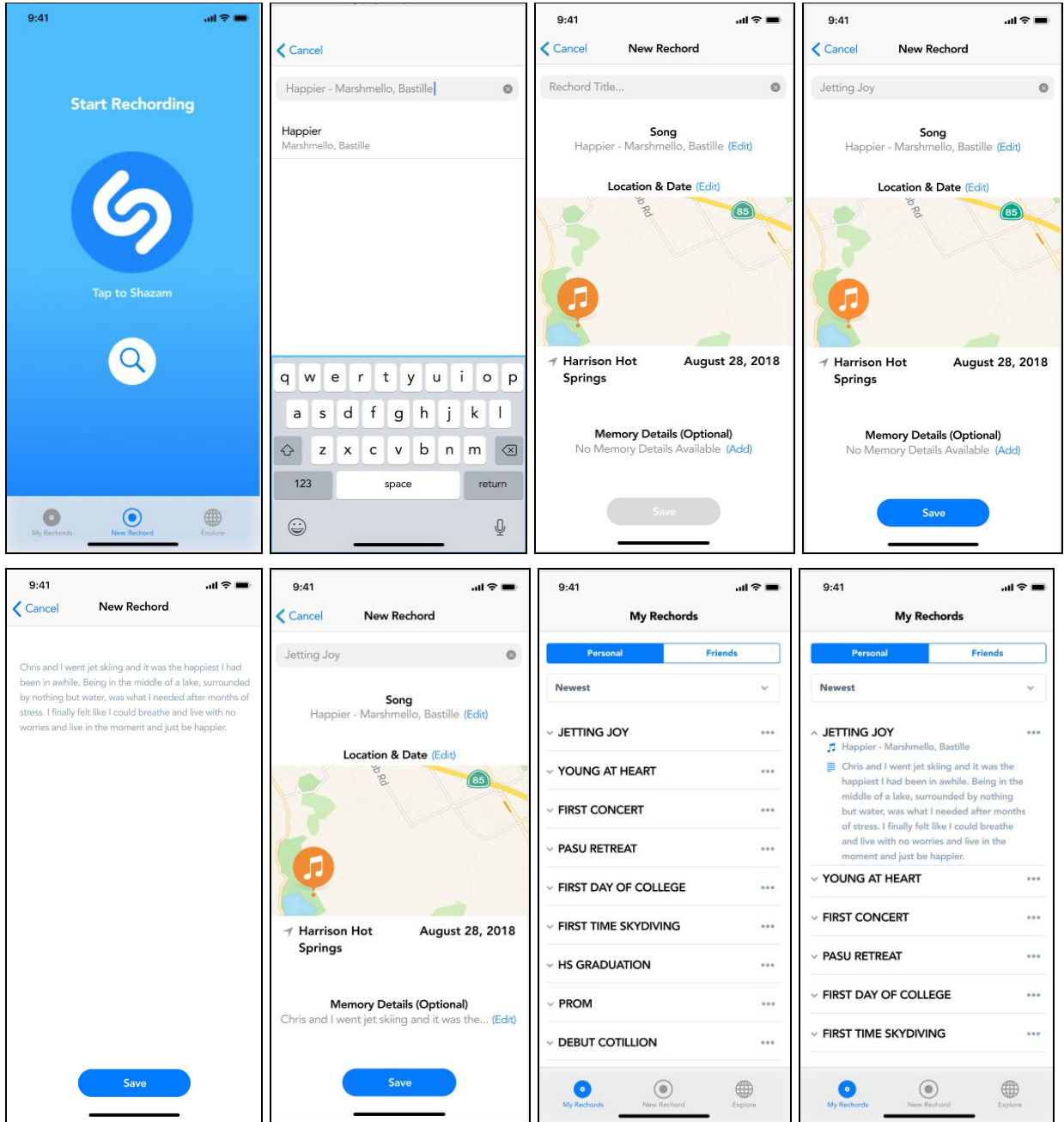


Figure 7: Low-fi Prototype - UI flow for sixth task (view public records at a location)

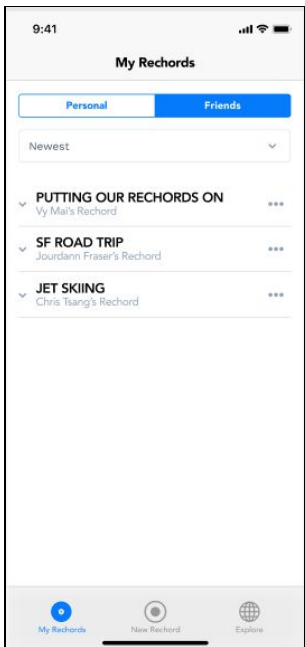
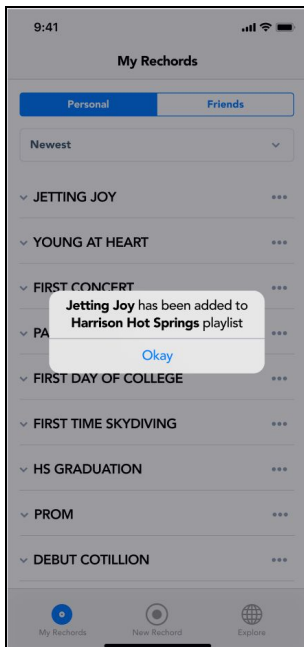
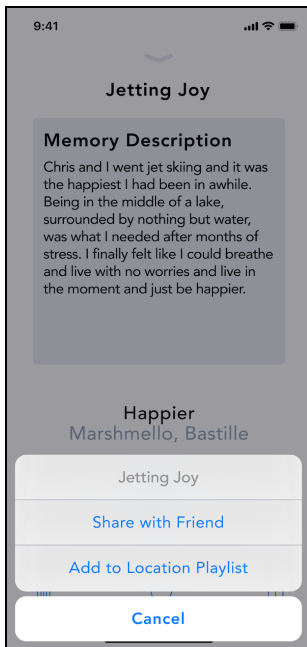
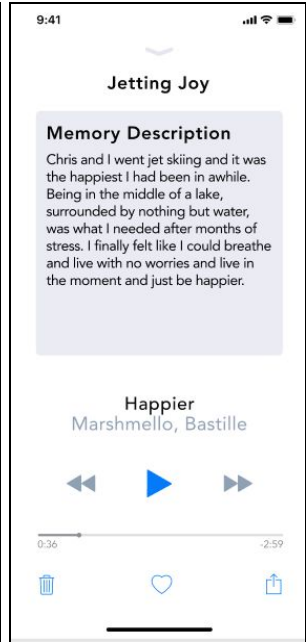
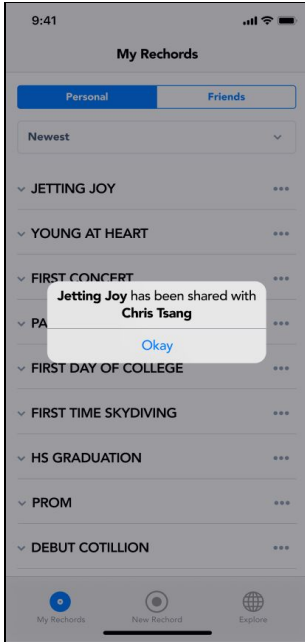
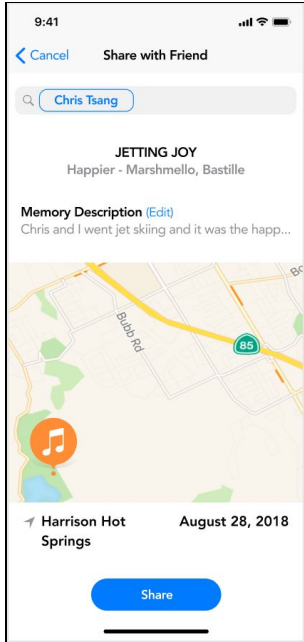
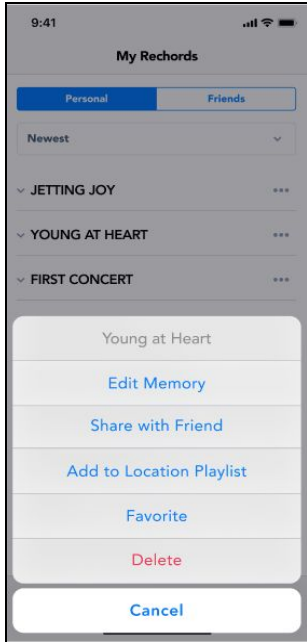
For the initial UI sketch, we decided to create a home screen with three major pages you can navigate to, aligned in the center of the screen (seen in Figure 1). The three major pages are Record Memory, Your Records, and Public Records. We also added an

inbox to the home screen, so people could view records sent from their friends. We kept this design for our lo-fi prototype.

# Medium-Fi Prototype







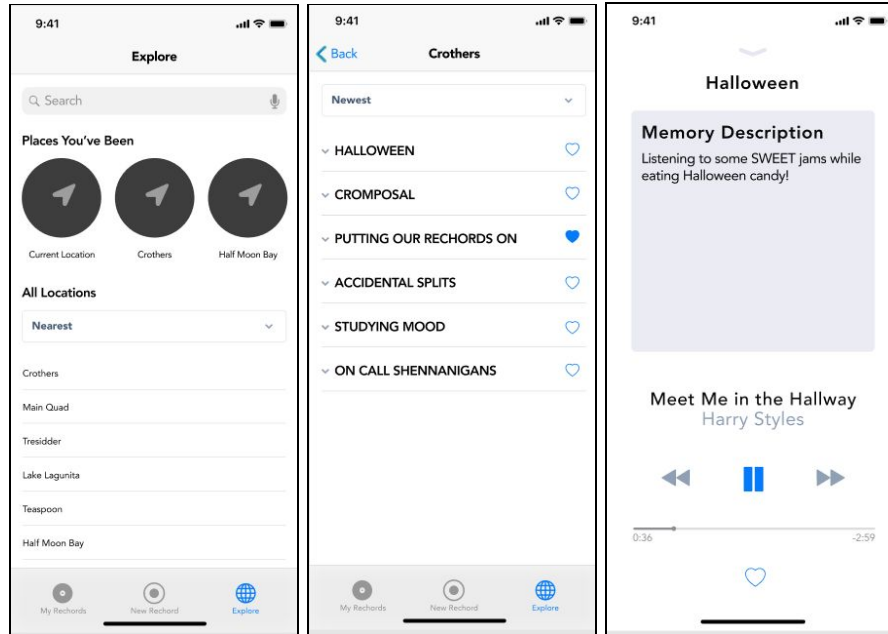
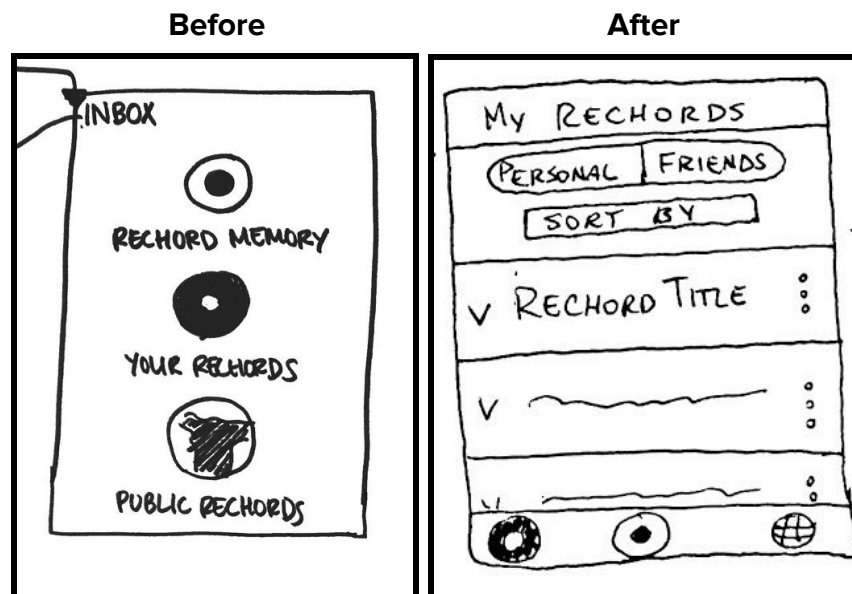


Figure 8: Medium-Fi Prototype

Based on the feedback from our lo-fi prototype, we decided to make the following changes:

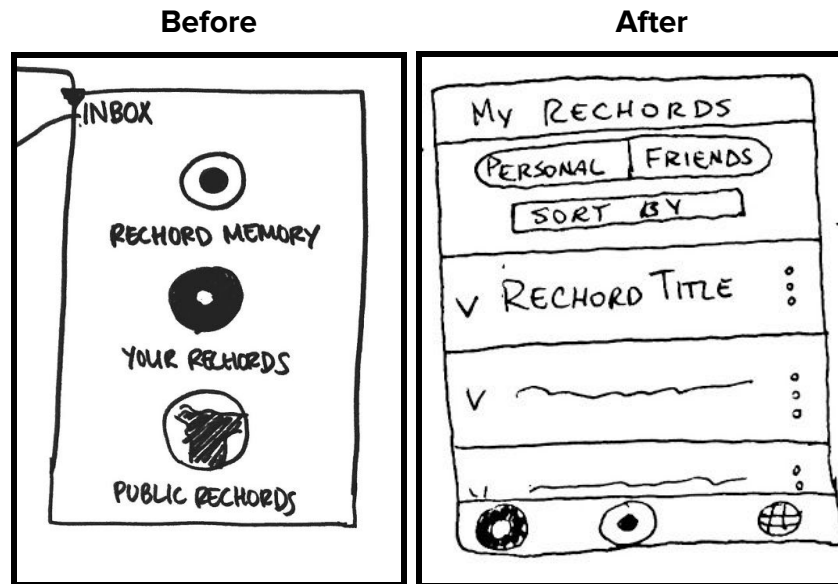
### 1. Replacing the Home Screen

We decided to get rid of the home screen and replaced it with tab navigation. We made this change because we thought it would make it easier for users to navigate between screens. Additionally, this change allows for the focus of our app, creating and saving records, to be highlighted.



## 2. Getting Rid of Inbox

We got rid of the inbox on the home screen and replaced it with a tab in the my records screen that toggles between a user's personal records and records shared with them. During our lo-fi testing, users had problems finding the inbox on the home page and found the name confusing. So we also changed the name of inbox to friends' records.

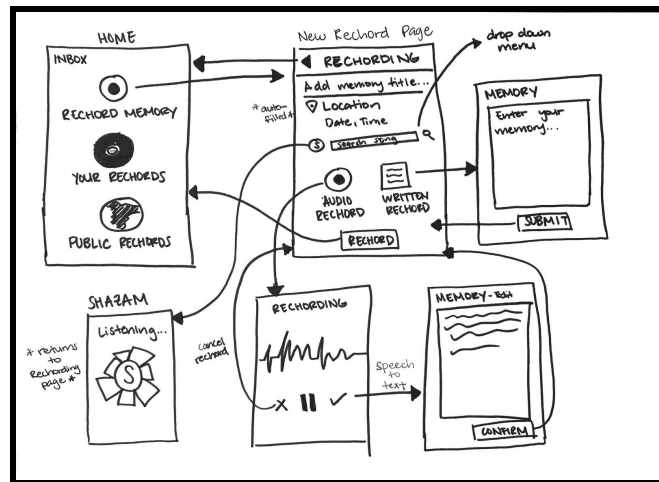


## 3. Streamlining Creating a Record

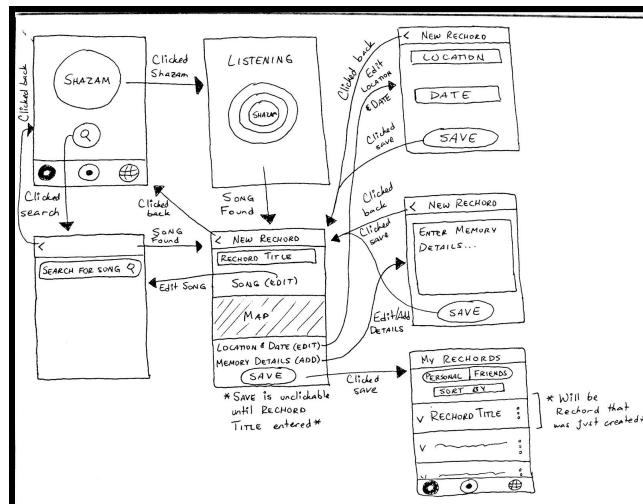
We change the process for creating a record so that a song could be recognized by shazam or searched first. A map of the location was added and speech-to-text can be done on the keyboard. We also make adding the details to a memory optional. We decided to make these changes because it was difficult for our users to create a record, and we thought this process seemed more intuitive.



## Before



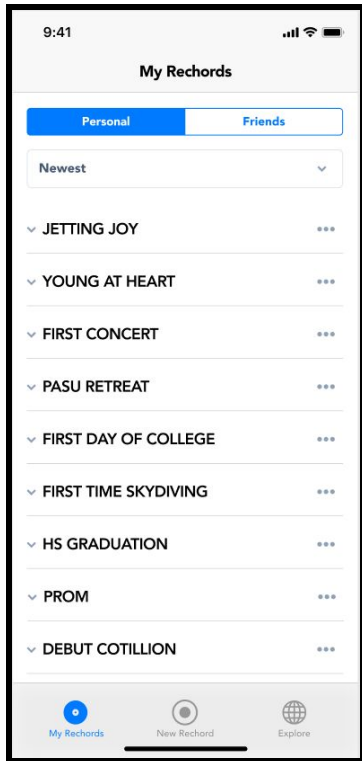
## After



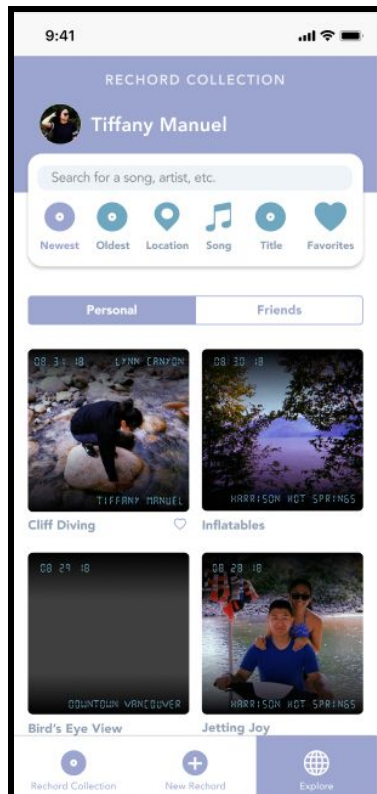
## Major Usability Problems Addressed

Prior to receiving the results of our heuristic evaluations, we began redesigning the general UI of our project, since we did not focus on the visual aspect of the project when we created the medium-fi prototype. The most prominent change that we made was changing from a “playlist” design to a “collection” design. Rather than listing all of the user’s record titles in a playlist format, we decided to add a visual aspect to a record, allowing users to customize the album covers for their records. This change allowed us to display more details of each record without overwhelming the user.

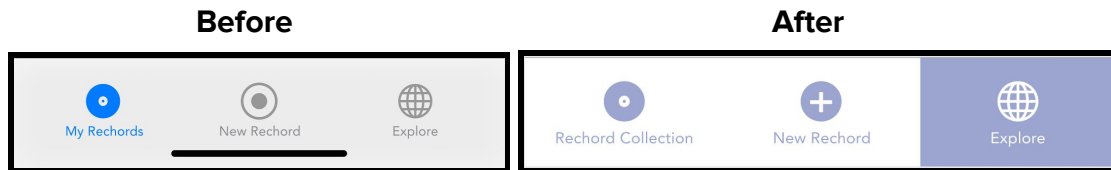
Before



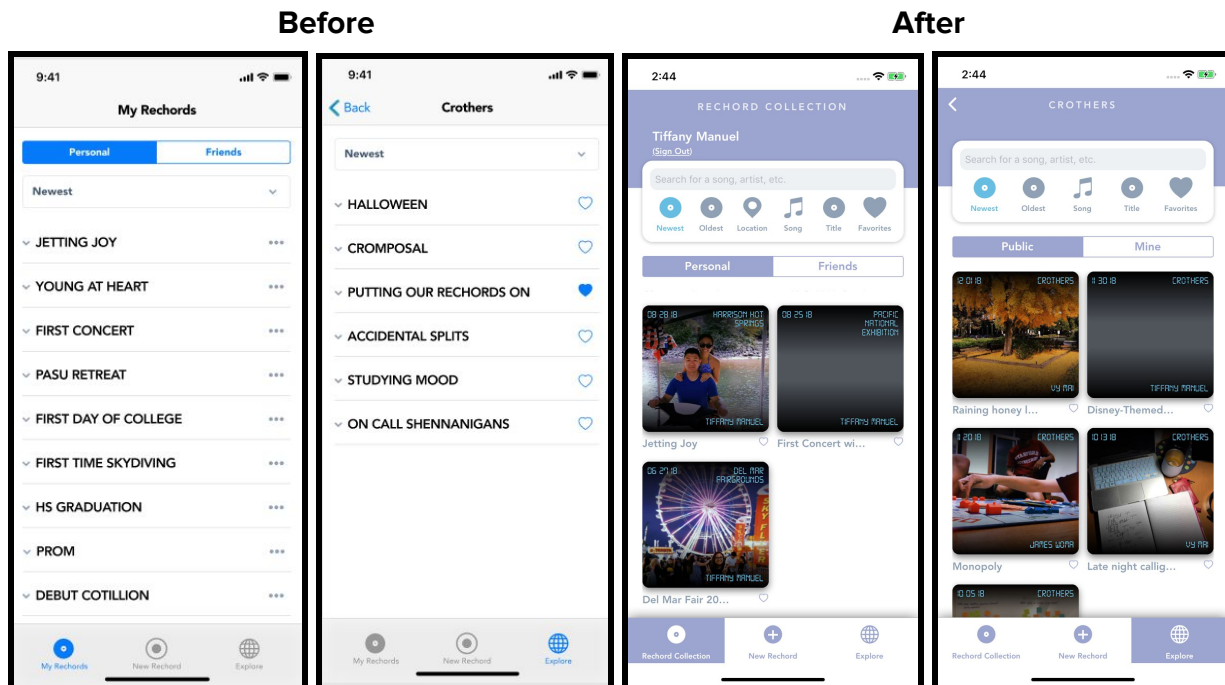
After



This change carried through to the terminology we used in the app. We now refer to personal collections, friends collections, and public collections. This change in terminology helped to address one of the issues brought up in our heuristic evaluation results, which stated that labelling the left-most navigation option as “My Rechords” made it unclear that rechords shared to you by friends could be found under this tab. We changed this navigation option to “Rechord Collection” to address this concern.

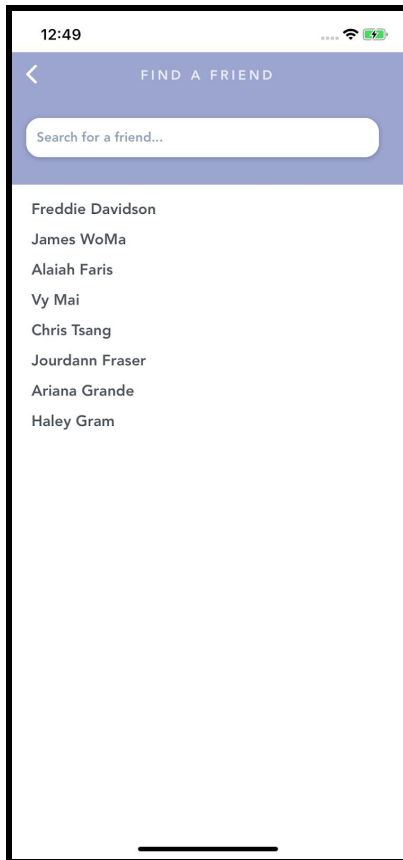


This overall redesign also allowed us to make collection screens more consistent with one another. One of the issues found during the heuristic evaluation was that of being able to favorite a rechord in the playlist screen for location playlists, but not being able to do so in personal playlists. We now allow the users to favorite rechords in all collection screens.

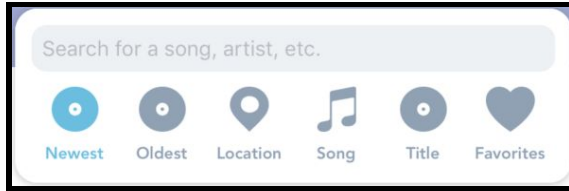


**Severity 3 and 4 Violations** (Note: Some violations were combined)

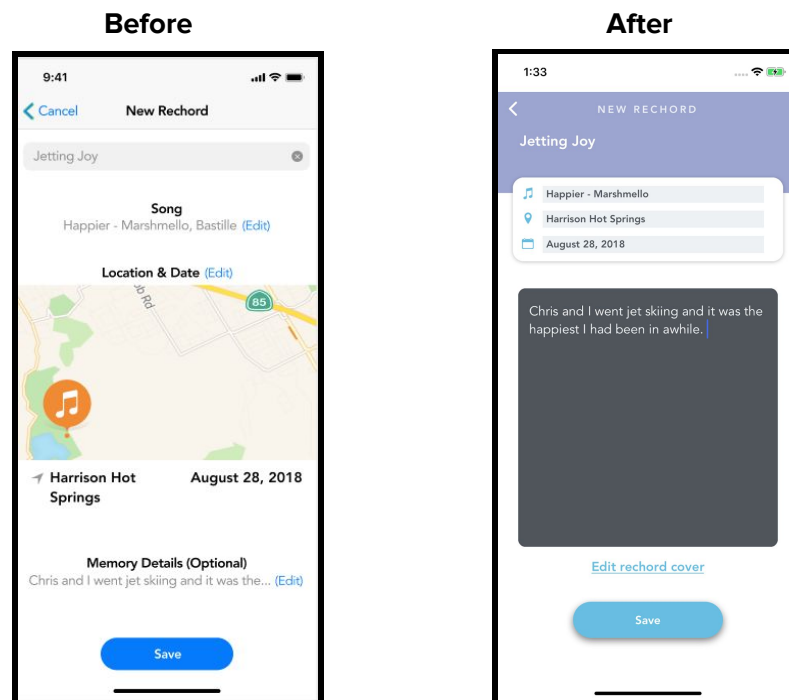
- 1. No “Add Friend” option.** In our medium-fi prototype, users were able to share rechords with friends, but it was unclear how users would add friends. Due to the time limitations of this class and the scope of this project, we did not implement a friend database and, therefore, an “add friend” option. However, users are still able to share with “friends.” This is currently done by allowing users to share with all other users, so, in a way, our user database can be considered a “friend” database. (Note: We did not think of how the “friend search” functionality would be implemented in the medium-fi prototype, so there are no before photos.)



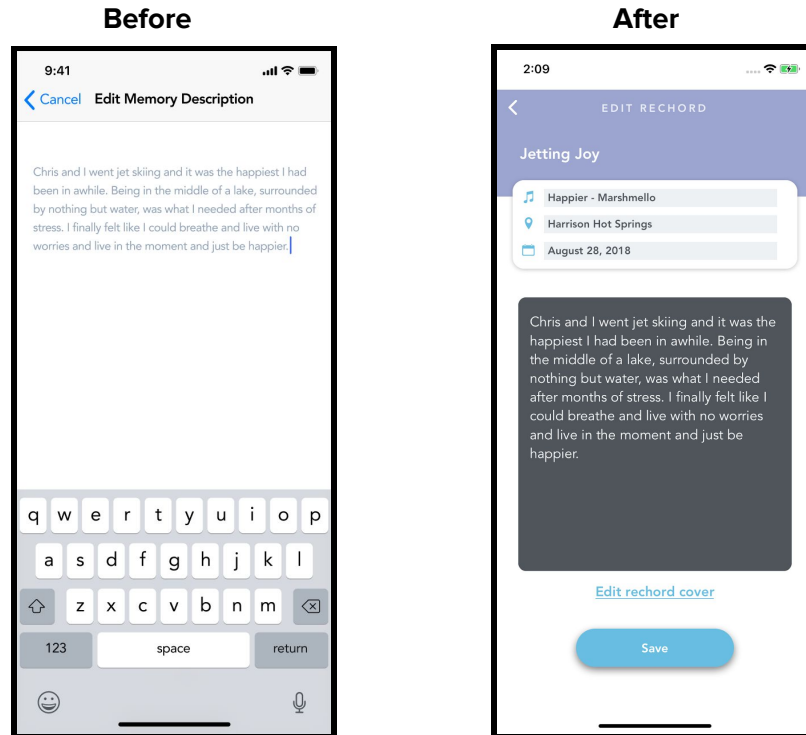
- 2. Results of favoriting a rechord were unclear.** In our medium-fi prototype, users had the option to favorite each rechord, but it was unclear what the consequences of doing this were. Users can now sort by their favorites. We decided to add “favorites” as an option for sorting because we believed it would indicate to users that rechords they have favorited will only be visible to them and that the act of favoriting was included to make it easier for them to access their favorite memories, rather than having to scroll through all of their rechords to find them. (Note: In our overall redesign of the UI, we also changed the look of the “Sort By” options. In our medium-fi prototype, the options were available in a drop-down menu, so we do not have a before picture of the previous sort by options as this menu was not implemented in the medium-fi prototype.)



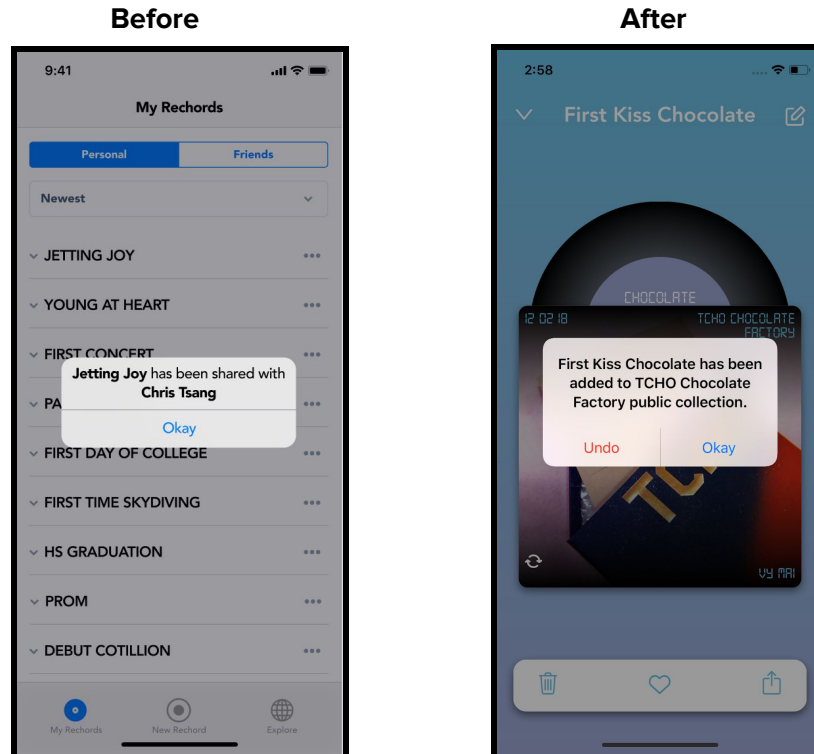
3. **Proofreading “Memory Details” was difficult.** In our medium-fi prototype, after the user entered the additional details for the memory, there was no way for them to proofread the entire description without having to click on the “edit” button and be taken to another screen. After our overall redesign of the UI, we put the memory details/description on the “back” of the album cover, which allows the user to edit the memory description/details on the same page that they edit all other parts of the rechord. Since the user now edits everything on a single page, they will no longer have to go to a different screen to view the full details/description.



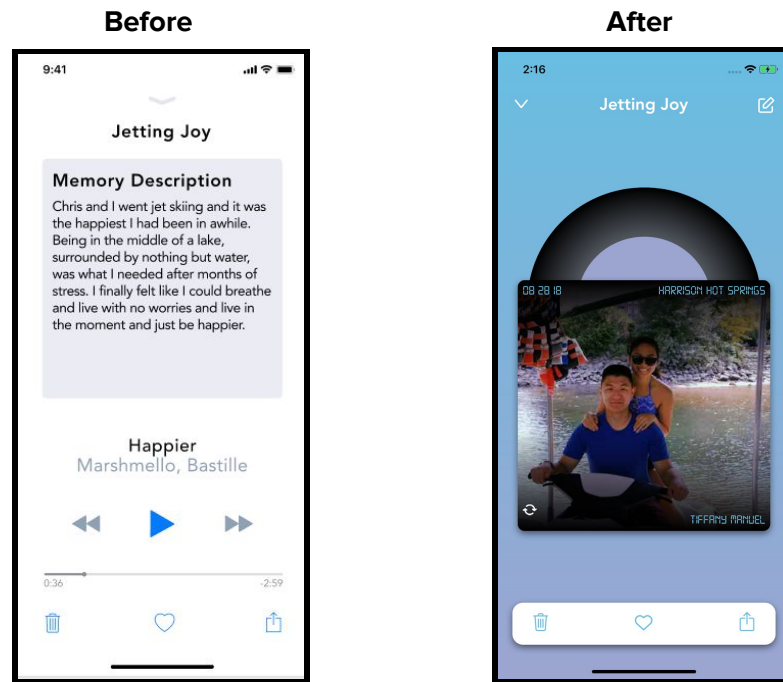
4. **Unable to edit song and location.** In our medium-fi prototype, we only allowed the user to edit the memory details. Rather than limiting the user to only being able to edit the memory details, we now allow them to change all parts of the rechord. Clicking on the “Edit” option goes to a screen similar to that used for creating a new rechord, allowing the song, location, rechord title, date, record cover image, and memory details to be changed.



5. **No “Undo” option after sending to friend or adding to public playlist.** In our medium-fi prototype, after the user sent their rechord to a friend or added it to the public location playlist, there was no way for them to undo either of these actions. We included an “undo” option to the notification/alert pop-up that appears after the user presses “send” or “add to \_\_\_ public collection.



- 6. Unclear that record album is clickable/flippable.** In our medium-fi prototype, there was no clear indication that the record being viewed was clickable/flippable. On the "New Record" screen, we have added a button that says "Edit memory description." When the user clicks on this button, we animate the record to flip, so the user is exposed to this functionality from the beginning. We also added a "flip" icon to the record cover and flip animation will be used when the user clicks on this icon.



## Prototype Implementation

### Tools

We built our prototype using React Native and Expo. Having taken CS 47, our developers were able to replicate much of the UI by creating custom components. Expo also has a great deal of pre-made modules to access users' phone data, such as an ImagePicker, Location (to access current location), and Permissions API. However, Expo tended to crash unexpectedly as our project grew, and React Native did not behave uniformly across platforms. The Android version is significantly buggier, and the UI is not perfectly replicated. Thankfully, we designed for iOS.

We were able to auto-fill the location name with the Google Maps API as well as search for a song using the Spotify API. However, using the Spotify API proved challenging as CS 47 did not teach us much on how to make API calls and go through token authentication.

We also stored our data in a real, live database called Firebase. Users may create an account and save/delete their own records, send records to a friend, and add to a public collection at a specific location. Despite making much headway functionality-wise, the documentation for Firebase was not suited for React Native, and image support in Firebase is fairly new. Currently, android users cannot upload images to Firebase.



## Hardcoded Data

We were unable to implement detecting a song with the Spotify API as it requires a more difficult and rigorous user authentication process, suited for websites and not for React Native. Thus, we just played the music in the background during the demo and automatically set the song to “Put Your Records On” by Corinne Bailey Rae. We also hardcoded the locations you can find in the Explore page, though the hardcoded locations will still update in real time with records from other users. There will just be a specific set of locations you can look at. We also hardcoded the profile picture and sort bar as they were not important features. The records were automatically sorted by newest date.

## Future Updates

Unfortunately, we could not implement every feature, but if we had time, we would have implemented the following:

- Detecting a song with Spotify API or music recognition software
- Adding a message when sending a record to a friend
- Expanding use of Google Maps API to search for a location
- Updating the Explore page based on user search history
- Implementing functionality for deleting a friend record
- Implementing sort by/favoriting functionality
- Moving the record cover up when editing the description
- Be able to play the music when viewing a record

## Summary

Many people like to look back on their memories and relive a past experience. Yet journaling is hard to maintain, and even an advanced journaler can find difficulty in capturing their true feelings/emotions into words. And while a picture may be worth thousand words, sometimes a thousands words is not enough to capture complex feelings/emotions. Records adds a new dimension to memory-keeping, music, to give users an easier way to remember through sound. Throughout the quarter, we have iterated through multiple designs in order to make the process of accessing your memories as easy as possible. Through this class, we’ve learned how to find a real-world problem based on users needs, brainstorm both wild and tame design ideas, and then focus on creating a solution that will best fit our users’ needs. In the future, we’d love to continue to develop Records and make memory-keeping an easy process that anyone can do.