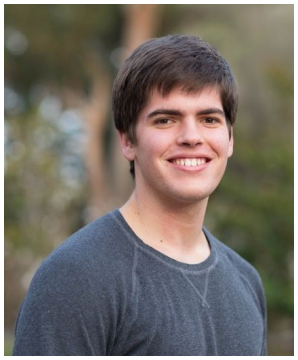
Hi-Fi Prototype

Assignment 8 • rekindl • Local Community (1:30PM)

Meet The Team



Ryan C.



Amanda L.



Sara V.



James C.

Introduction

Mission Statement: **Reignite** faded friendships.

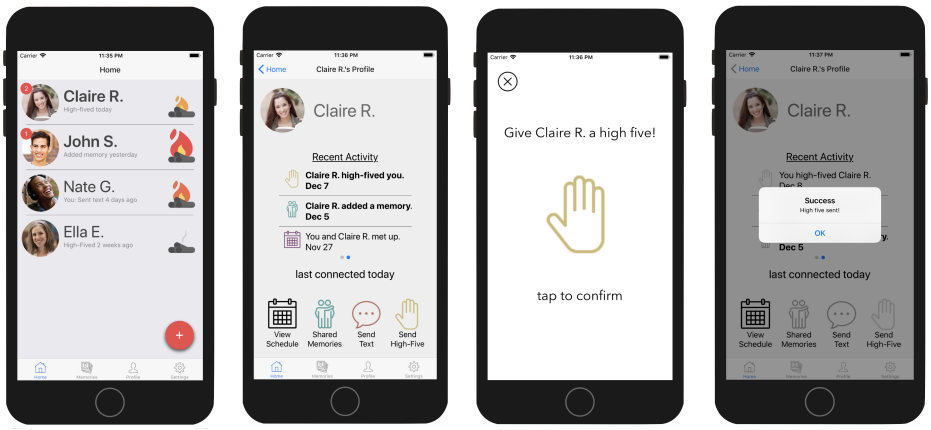
Problem Overview: Busy schedules and distance can cause even the closest of friends to **grow apart**. We often **forget to check in**, and with so much on our plates, it can be extremely **difficult to follow up** on those “we should get lunch sometime” or “it’s been too long” moments.

Solution Overview: Rekindl helps you keep up friendships by **reminding** you of connections that are becoming distant, **saving** memories of your meet ups, and **choosing** relationships you want to reignite.

Final Interface

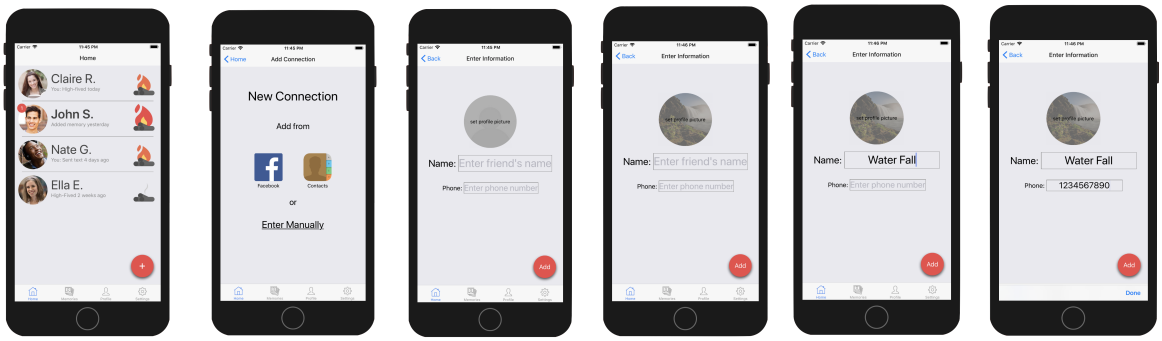
Task 1 (simple):

Get a notification about a faded relationship and send a text to that friend to reconnect. We chose this as a simple task because this is the core functionality of our app — to get reminders about faded friendships so that you can reignite them.



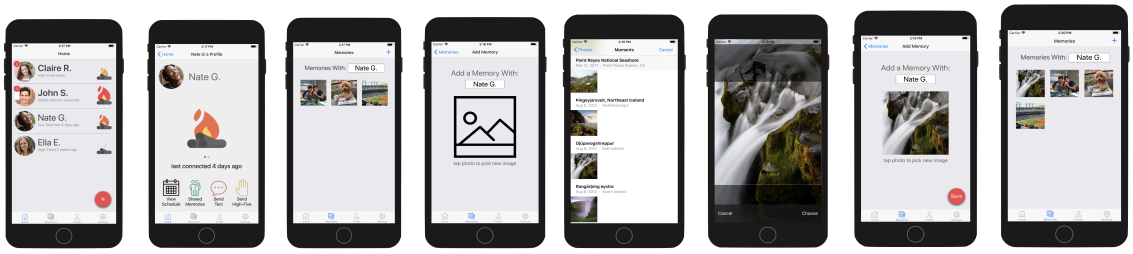
Task 2 (moderate):

Add a new connection to the app. We chose this as a moderate task because it is important to be able to add new friends to the app as you form new and stronger relationships, but this may not be a feature that you use every day



Task 3 (complex):

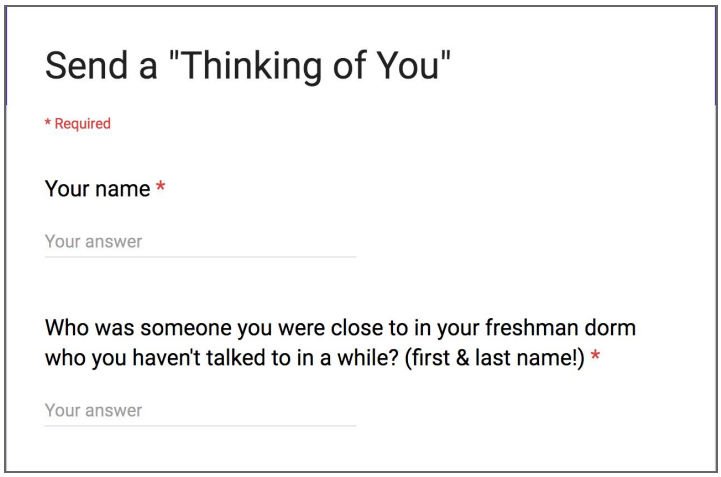
Add a picture you and a friend took at a recent meet-up to your memories with that friend. We chose this as a complex task because, while it takes a little more work and requires users to take pictures with one another, we believe it could really help strengthen relationships by adding another aspect that they can bond over. It also adds a way for the app to record physical meet-ups and reflect that in the fire icon size.



Design Evolution

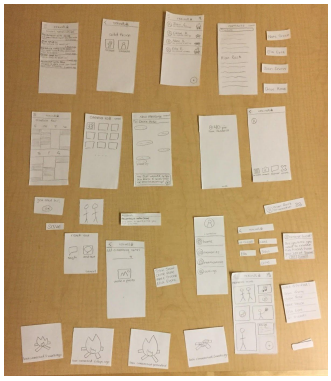
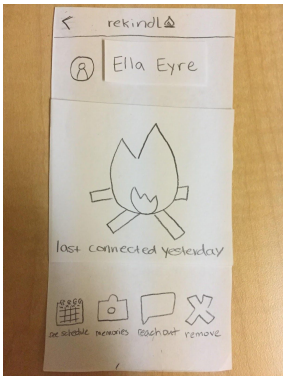
Experience Prototype:

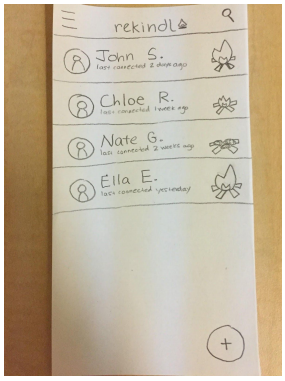
For one of our experience prototypes, we circulated a Google Form that allowed participants to check a box to have us send an email to one of their friends that they hadn't gotten in touch with for a while, letting the friend know that our participant was thinking of them. We got an overwhelmingly positive response, finding that nearly all our participants had friends who they truly valued, but struggled to find time to connect with them on a regular basis and thus had drifted away from. They were eager and willing to reach out to these friends when given an easy way to do so. These findings inspired our solution of **rekindl**, an app to help manage friendships and remind users to reach out to friends in order to maintain those connections.

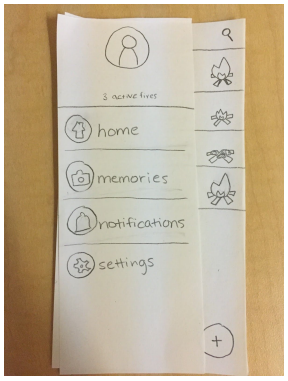


Lo-Fi Prototype:

After sketching out a few different implementations for our interface, we decided on a UI where users see a list of all their connections, and then can click on each individual connection to view more information on that relationship. We opted for this over UIs oriented around newsfeeds, schedules, or memories because we wanted the focus to be on the individual relationships. We still encorporated schedules and memories into the app, accessible through the menu and on an individual basis. After sketching many screens (some shown below), we tested it on a few participants, and found that overall, they liked it! There were, however, a few issues we found, mostly involving the menu and buttons on the detail pages for connections, that inspired changes moving forward to our med-fi prototype.

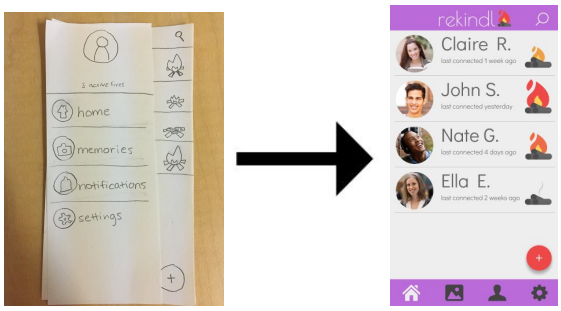




Med-Fi Prototype:

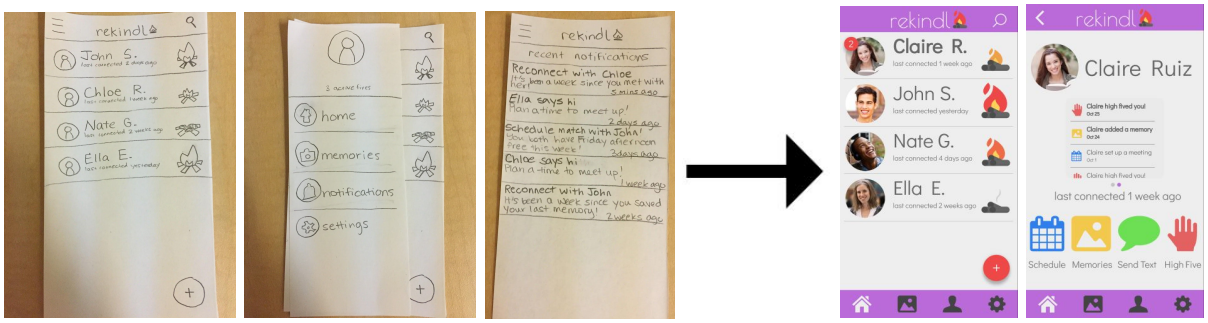
We designed screens for our med-fi prototype in Figma, and connected them together in InVision App. The biggest change in our design at this stage was changing our menu from a hamburger menu in the top left corner to be tabs at the bottom of the screen. We made this change because our participants for pilot testing of the lo-fi prototype found the hamburger menu less intuitive, and thought that it took too many taps to access key screens. Having the tabs allows for quick and easy access to the home screen, memories, your own profile, and settings. Other changes we made included getting rid of the delete button on the detail page for connections and splitting the connect button into a send text button and a high-five button. We did this because we received feedback that having the delete button that big made it look like a main functionality, while having extra clicks to get to each method of connecting was a pain. At this point, we also fleshed out more of what we wanted to show for the user profile page and the schedule view based on what our participants reported expected to see and be able to do on those pages.

Navigation Changes



Menu Tab

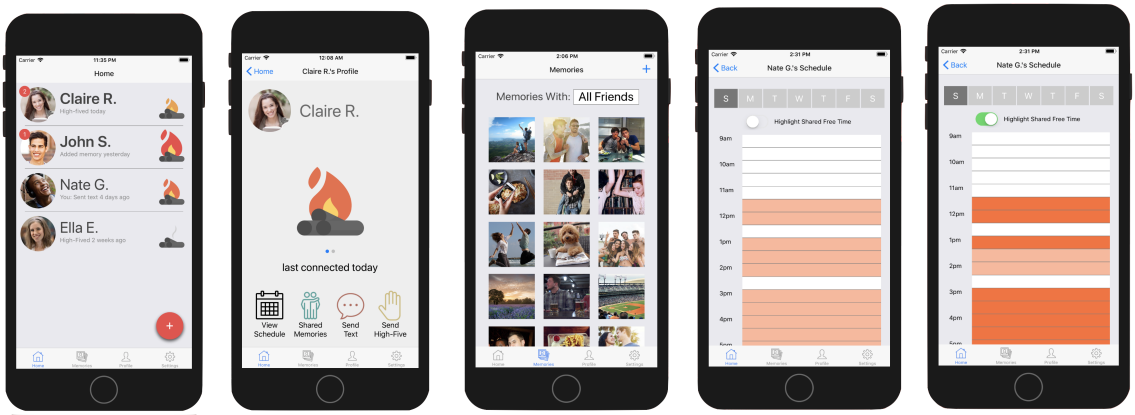
Notification Changes



Menu Navigation Notifications within connections

Hi-Fi Prototype:

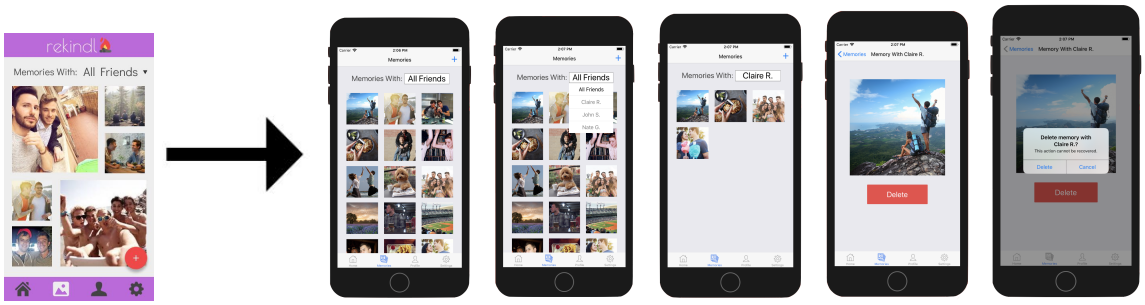
Overall, our hi-fi prototype was similar to our med-fi, but with some minor differences, and with some more thoroughly implemented features (mainly because there were many limitations on what we could do in the med-fi prototype without being able to save state. We received feedback on our med-fi prototype through a group Heuristic Evaluation. Results from this and the changes that it yielded are described in the next section. Details on the implementation of the prototype are described in the section after that.



Major Usability Problems

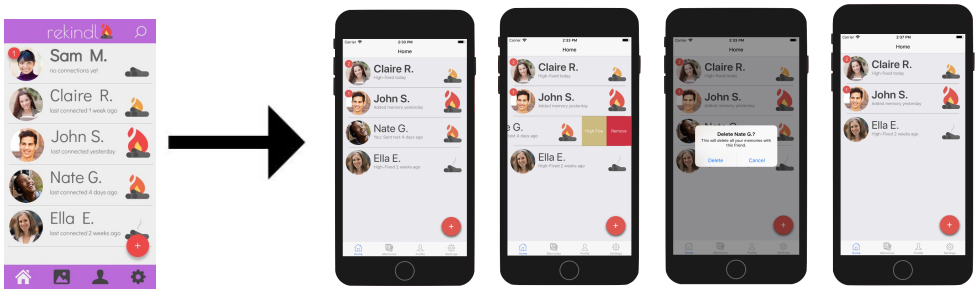
The main violations we found in our feedback from the HEuristic Evaluation centered around the Memories, Home, and Calendar Pages. We had 6 violations of severity 3, and none of severity 4. The severity 3 violations and our fixes for them are shown below.

1. There was no way to delete a memory once it had been added. To solve this, we added the ability to tap on a memory to navigate to a new page where the image is expanded and a delete button is shown beneath this larger image. This had two benefits: it allowed us to implement the 'delete memory' functionality in a way that didn't overcrowd the main Memories page, and also provided a way to see the images better.



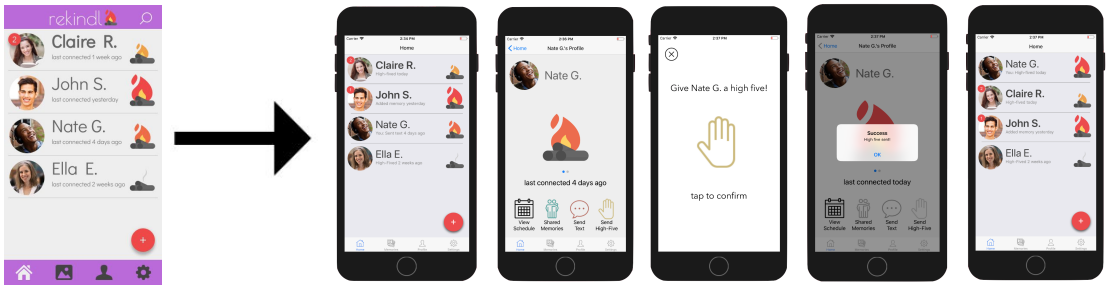
Before After: Navigate to a specific memory detail screen to delete it

2. There was no way to delete a connection once they had been added. To solve this, we followed the common iOS standard of allowing users to swipe on an item in the list to remove it from the list. Swiping left on a connection from the home screen gives the option to remove the friend, complete with a short confirmation message to make sure you don't accidentally remove friends.



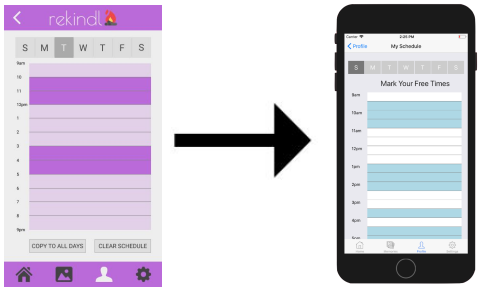
Before After: Deleting Nate G. removes him from main list

3. The ordering of connections on the home screen was also not clear to everyone. Our intention was to simply bring friends up to the top of the list whenever they had a notification, and allow them to fall back down as we got notifications from other friends. In our med-fi prototype, however, we did not update the “last connected” time with notifications, so this added to the confusion. In our hi-fi prototype, any action taken by either the user or one of their connections (high five, text, added memory, etc.) will update the type and time of the last connection and move that connection up to the top of the list. Any interaction will also immediately change the size of the fire for that friend.



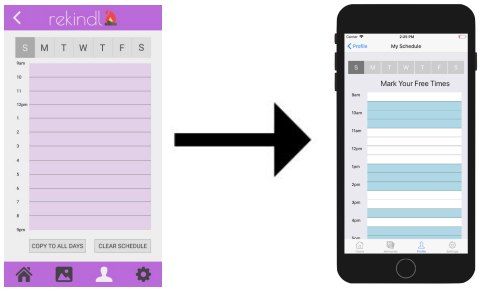
Before After: High-fiving Nate G. brings him to top/updates fire size

4. It was unclear whether users were supposed to mark their free or busy times on their own schedule. To clarify this, we added a header on the schedule page that directly prompts users to mark their free times.



Before After: Added header

5. In our med-fi prototype, we had a button below the schedule that said “Copy to all days”. This language was unclear for some, and we decided that having an identical schedule day-to-day within the same week was unlikely, so we just removed this button altogether.



Before After: Schedule only

6. Due to the inability to save states in the med-fi prototype, the only way to undo a change made to your schedule (i.e. untoggle a certain time-slot) was to click the button to clear everything. In the hi-fi prototype, each time-slot on the schedule toggles on and off when you click on it, allowing much more user control.



Before After: To deselect a time-slot, just click!

We also made a few other changes based on lower-severity violations from the Heuristic Evaluation, and just from things we noticed.

- We changed the general aesthetic of the app to the more clean look that React Native supports.
- We removed the “View All Memories” option from the user profile page, as that was redundant with the functionality of just selecting the Memories tab from the bar at the bottom of the screen.
- The appearance of adding memories is also slightly different, as it uses the actual Camera Roll from the user’s phone.
- React Native by default puts back buttons in the top right corner of screens, improving our navigation where we didn’t have back buttons.
- We moved the ‘+’ button to add a photo from the Memories page to the header, as it was blending in a little with the pictures around it in the bottom right.

- We changed the notification section in the detail screen for connections to be a log of all interactions (by the user and the connection) in order to keep track of both sides of the friendship.
- When you swipe over a connection in the list on the home screen, in addition to 'Remove' there is also an option to send a high five from there. This enables extreme/expert users to quickly send high fives.
- We took away the 'Edit Account' text on the user profile screen and just put a pencil next to the name, as the pencil is standard enough to indicate to users its functionality as an edit button.

Prototype Implementation

We built our hi-fi prototype as a "Create React Native App," using the React Native and Expo frameworks. React Native allowed us to program in Javascript without needing to learn Swift on the fly to develop for iOS. The familiar feel of JSX (Javascript's XML-like language) allowed us to use a format similar to HTML tags and CSS styling to design the app. Expo allowed for easy testing and the ability to quickly update changes to our code and view the most recent version of the app on our own phones. React Native and Expo both supplied us with built-in APIs for interfacing with the Camera Roll and other common styles of elements (lists, drop-down menus, swipeable items, etc.). A downside of it was that there were certain React Native add-ons that were only available if we were coding through XCode instead of Expo. This would have involved "ejecting" the app to use with XCode, and we no longer could have tested it through Expo. A member of our group does not have an Apple computer, so working with XCode would not have worked. There were also problems with generating a .ipa file to submit it.

We used a couple of Wizard of Oz features, mainly circling around the connections aspect of the app. High fives say they send, but do not actually have a recipient, and texts actually just send to a member of our group. Additionally, memories can only be seen by one user (the current user of the app) rather than being seen by the two parties the memory is "shared" with.

All of the initial data is hardcoded too, including the names, activities, and photos of the four connections that the user (also hardcoded) has to start. This was because we did not implement onboarding or user accounts, and it also showcases more of our features if the demo does not start from scratch.

Aside from onboarding and accounts, there were a couple of other features that we left unimplemented for now. We did not implement a search function on the home screen for the hi-fi prototype, as it would only be important for users with many connections, which is not in the scope of this prototype. We also only implemented adding new connections manually instead of trying to interface with APIs for Facebook and the default Contacts app. Currently, your own schedule is randomized and editing it does not save state. Additionally, the recent activity logs and fire sizes do not reflect adding a new memory. The Settings section also is not implemented, as this will probably just contain things like notification preferences and sending us feedback and things like that. The prototype also does not send push notifications at this point, as there are no accounts to be interacting with one another and sending each other notifications.

Summary

While much of the technology today aimed at bringing local communities together offline is focused on *finding* new friendships and relationships (e.g. Tinder, Bumble), we found that there was a lack of products that focused on *maintaining* these connections once they're formed. Through our interviews and experience prototype, we found this not only to be an extremely widespread problem, but one that most people were eager to fix once given an easy way to do so. Thus, **rekindl** seeks to fill this market gap by providing users with an elegant and simple way to keep track of their friendships and reconnect. Throughout the past quarter, we used an iterative design methodology in the creation of **rekindl**. At every stage of the design process from the initial needfinding interviews to the high-fidelity prototype presented at the end-of-quarter project fair, we continued to both expand and refine our ideas for moving forward with our product, then build and prototype. Although our "final" product still has much that has been left unimplemented and we still have many ideas for new features that would be added if we were to continue working on **rekindl**, we have developed a version of the app that fits within the scope of this class and that we are all proud of creating.