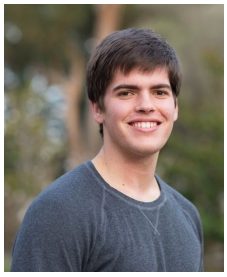
# Final Report

CS194H • Assignment 14 • Rekindl

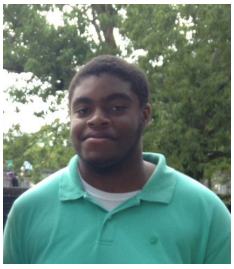
## Meet The Team



Ryan C.



David M.



Ikechi A.



Ben W.



Jesse C.

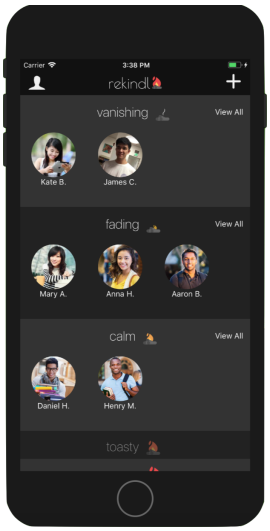
## Problem Description

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. Existing social media can encourage **breadth over depth** in friendships, and drive people to curate an **artificial** “public self” in order to try to impress people online.

## Solution Overview

Our mission is to **reignite** faded friendships.

Rekindl helps you keep up friendships by **reminding** you of connections that are becoming distant and allowing you to **choose** relationships you want to reignite. The app prioritizes friends who you have not connected with in a while and makes it easy to **get back in touch** with them.



# Tasks

**Task 1 - Set a status (simple):**
By setting a status from your profile screen, your friends who have you as a connection in their app will see your status. It can often be difficult to keep everyone up to date on developments in your life, but this helps you get the word out to your core group of close friends, and gives them a talking point with you to help break the ice and help you get started with rekindling your friendship. This is our simple task because it requires very little effort on the part of the user to enter a status/life update every week or two.

**Task 2 - Rekindle a vanishing friendship (moderate):**
Select a friend from one of the top categories on the home screen, whose fire is dying out because you have not reconnected for a while. From their detail screen, you can see their status if they have the app and easily shoot them a text or call them. Because different people have different definitions for what it takes to consider their friendship "rekindled" (text conversation vs. phone call vs. in-person meetup), to reset that connection's fire, you must then click the "Rekindle Friendship" button. This is probably the most important part of the app, because our goal is to get people to maintain their friendships. We designated it as our moderate task because, while we do what we can to make it easier for people to reconnect, it still requires some effort from the user.

**Task 3 - Add a new connection to the app (complex):**
Click the "+" in the top right corner and either select one of your contacts who you want to add to the app, or enter their information manually. This is an important task because you will need to add a few of your friends to the app when you initially create your account, and may also have reason to add friends later as you meet new people or begin to grow apart from old friends. We designated this as our complex task because it requires the user to decide how frequently they want to reconnect with the friend. We tried to highlight the selection field for this, because some users had struggled with making this important decision.

# Task Flows

**Task 1:**
Click profile icon in top left corner of home screen (Figure 1, screen 1). Then click the “Update Status” button (Figure 1, screen 2), and enter your new status and press “Update” (Figure 1, screen 3).

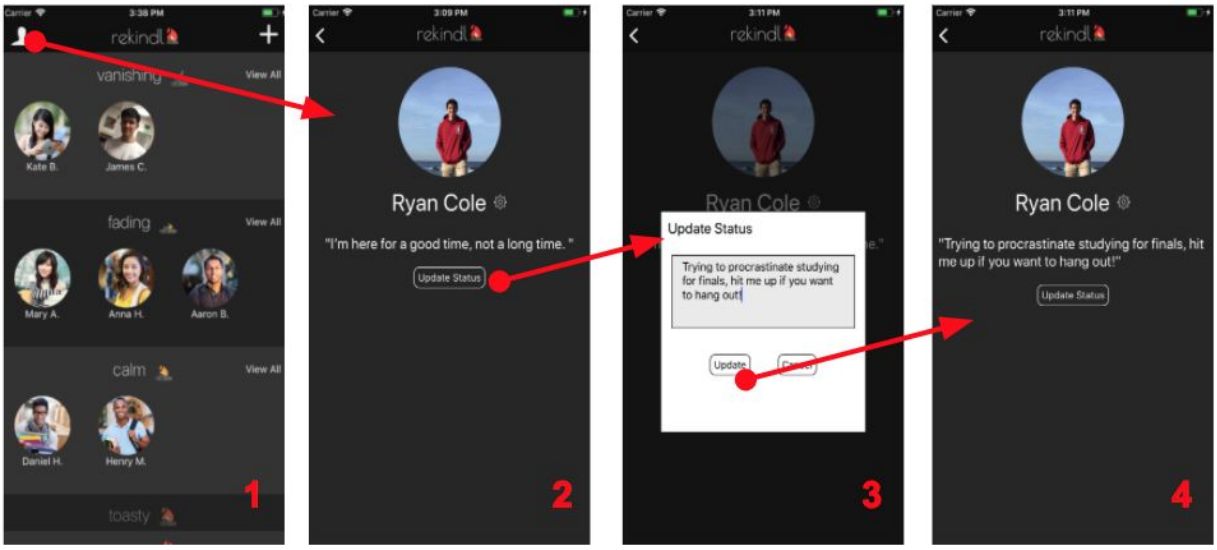



**Figure 1: Task 1 - Set a status**

**Task 2:**
Select a friend from the home screen (Figure 2, screen 1) and click the message icon (Figure 2, screen 2) to send them a text (Figure 2, screen 3). Once you are satisfied that you have reconnected sufficiently, click “Rekindl Friendship” (Figure 2, screen 4) and confirm (Figure 2, screen 5).

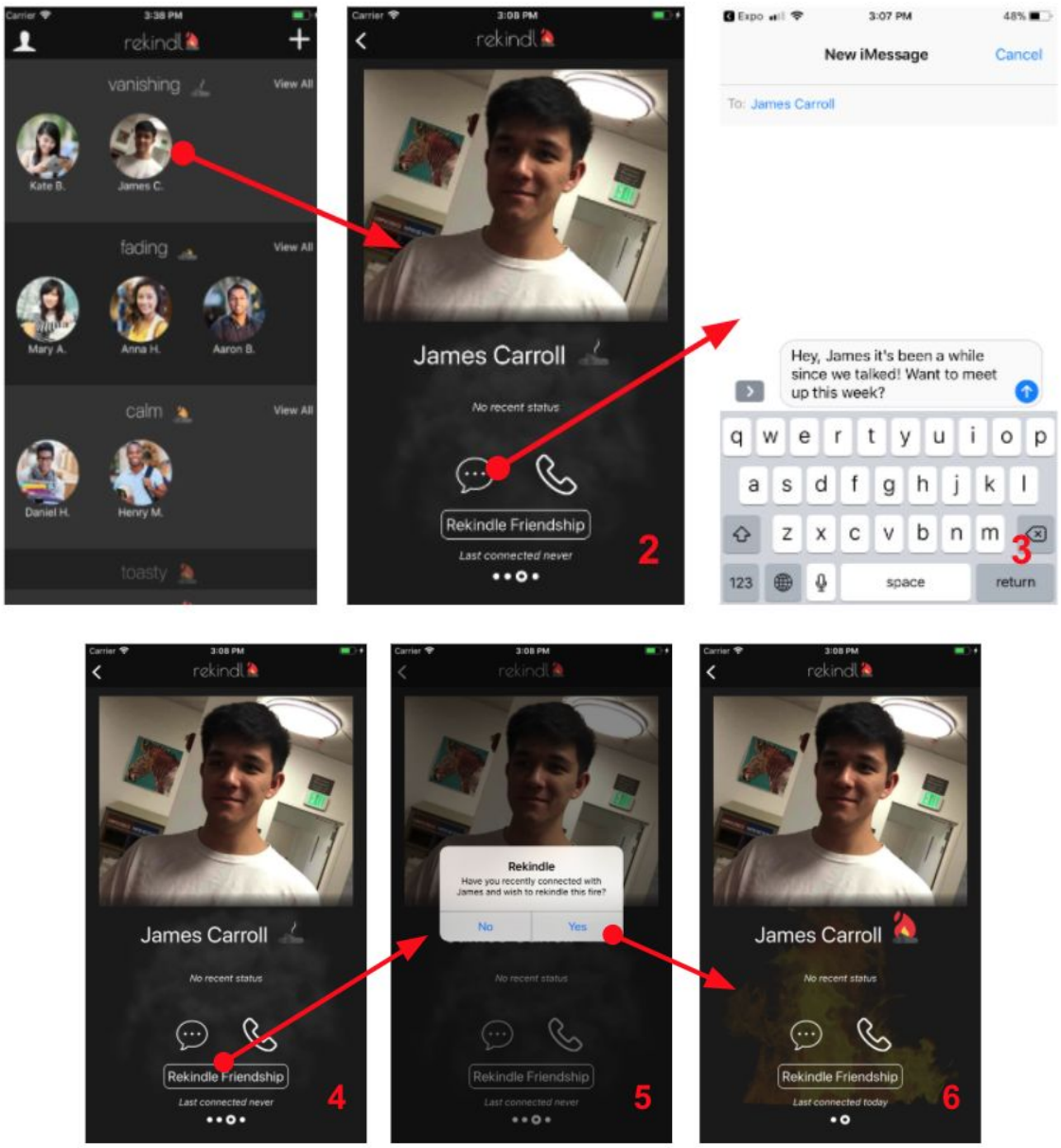



**Figure 2: Task 2 - Rekindle a vanishing friendship**

**Task 3:**

Click the “+” in the top right (Figure 3, screen 1) and select a friend from your contacts (Figure 3, screen 2). Update any of their information as needed (Figure 3, screens 3, 4), and click add (Figure 3, screen 5).

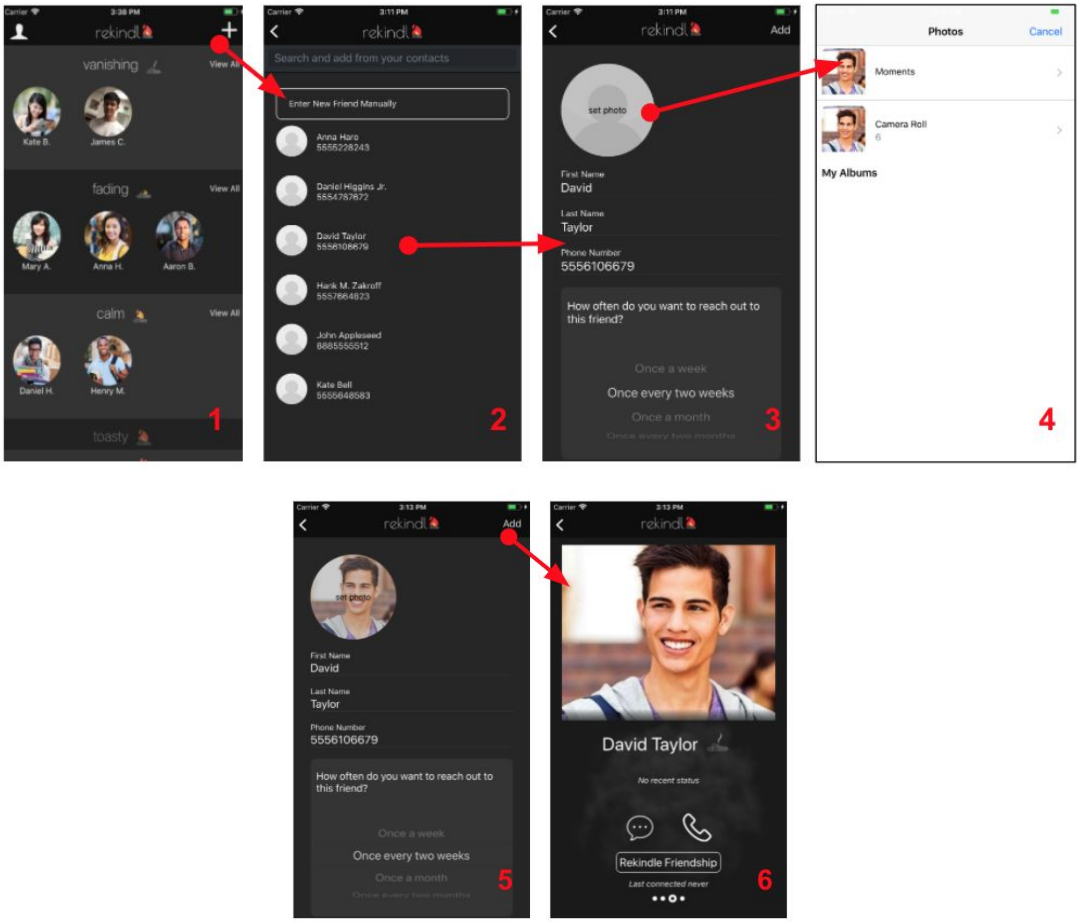



**Figure 3: Task 3 - Add a new connection**

## Design Evolution

**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.

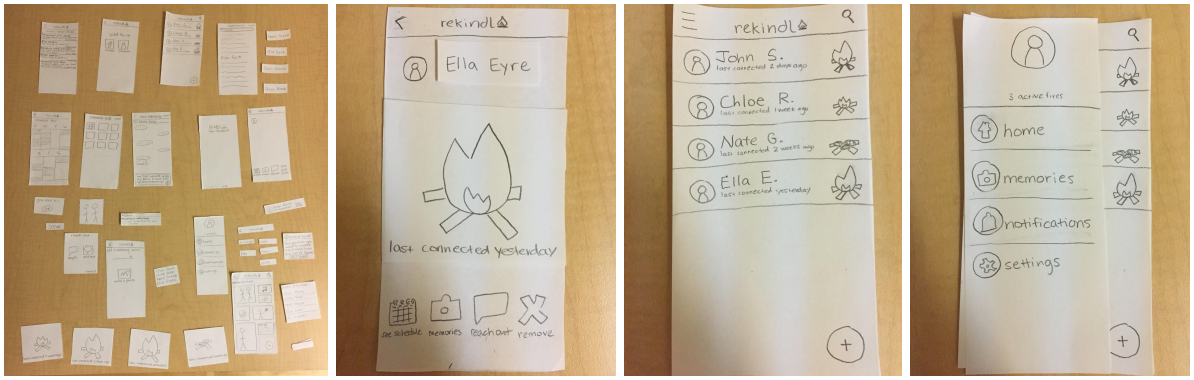



**Figure 4: Lo-Fi Prototype Screens**

**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. 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.

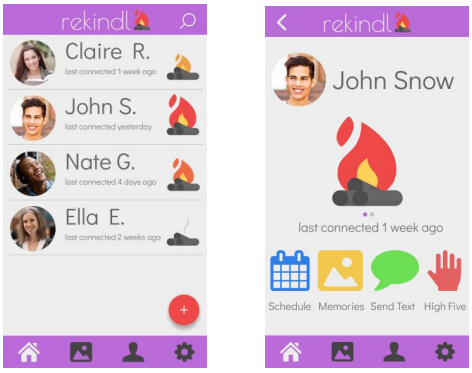



**Figure 5: Med-Fi Prototype Screens**

**Hi-Fi Prototype 1:**

Overall, our hi-fi prototype 1 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. Most of the changes we made were to flatten features out a little more and clean up the visual appearance of the UI.

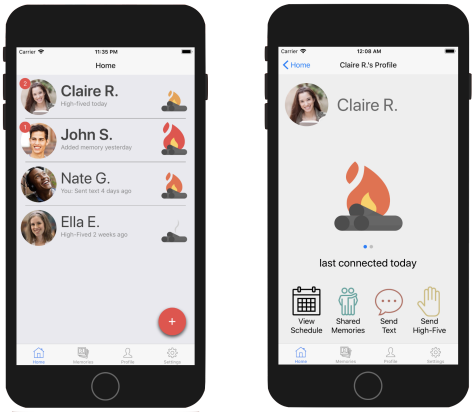



**Figure 6: Hi-Fi Prototype 1 Screens**

**Hi-Fi Prototype 2:**

Entering this quarter, with 4 new team members, we really started questioning whether our current design was the best way to cultivate deep, personal connections. We ultimately decided that the current list view we had as our home screen was too reminiscent of other existing interfaces in social media apps, which we believed were not ideal for deep connections. We mocked up some completely new redesigns for the UI of our app, while maintaining the same basic idea and functionality that we believed was core to the idea of Rekindl. We also decided to focus in on what was key to our app — helping users maintain their friendships. As a part of this, we cut out the memories, schedule, and high-five features that we had in our last prototype, because these felt like they only complicated the app and didn't provide any features that our users couldn't already get from Facebook or Google Calendar. While we didn't have this version of the prototype done for our lab usability testing, we showed our participants the mocks for the new design and got positive feedback, so we went ahead and implemented the designs for our hi-fi 2. By sorting friends into groups based on the size of their fire, we clearly prioritized friends whose fires were dying out. This new design was also more conducive to having more friends in the app as well, and also had a more safe feel to it.

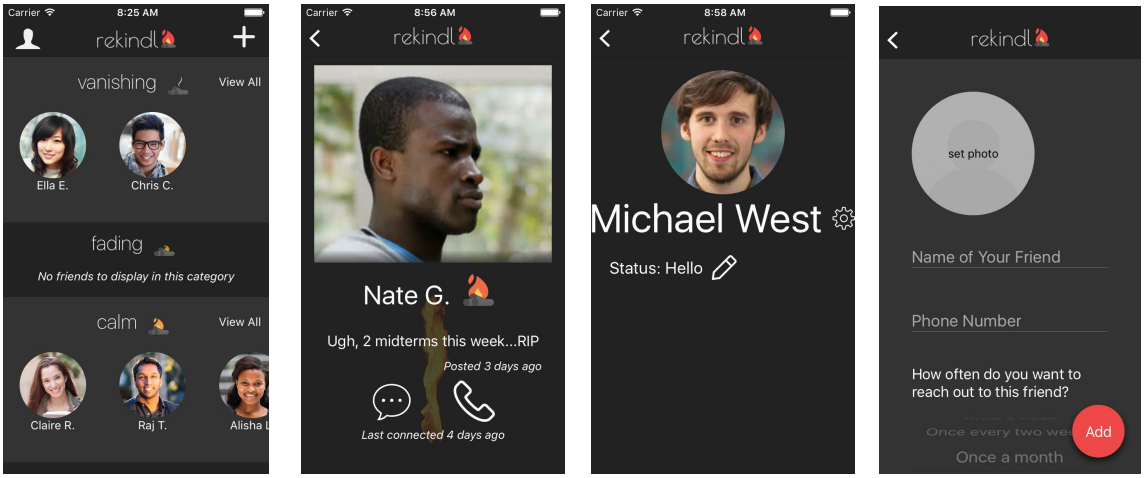



**Figure 7: Hi-Fi Prototype 2 Screens**

**Hi-Fi Prototype 3:**

I believe that our most important testing was our field usability testing, as that helped us fine tune our final hi-fi prototype. Through this testing, we found some issues with the visibility and usability of certain features, and with certain consistency standards that we were not sticking to. In particular, the status module was confusing for many of our participants, and many of them also overlooked the spinner for selecting how often they wanted to reach out to friends. Before testing, we implemented a sign up/sign in flow, using Firebase authentication to sign in users, but our clickstream logging revealed to us that users were spending too long on the sign up page, leading us to put more clear instructions about password requirements. In general, the biggest change from our 2nd hi-fi prototype to our 3rd was the ability to register users and have accounts backed by data stored in Firebase. However, we made many other small tweaks based on feedback from the class and from user testing. We also implemented our settings screen, where users could edit their connections and delete or change the information of their friends. Images of these screens can be seen in the next section (Figures 8-12).

# Final Implementation

**Final Interface:**

Our final UI design didn't change much through our past couple prototypes, but was a huge change from where we were last quarter. We ended up really liking our new design that helps users focus on developing deep connections and having meaningful

interactions with friends with whom they have not connected with in a while. We kept a consistent color scheme throughout the app for a clean feel and smooth transitions between screens.

The first big functionality that a user would do on downloading the app would be to create an account or sign in (Figure 8). Both signing up and signing in have smooth, intuitive flows. We added more descriptive placeholder text on the sign-up page to help minimize user errors, and added underlines to text fields to make it more clear that it was a field for them to fill out.

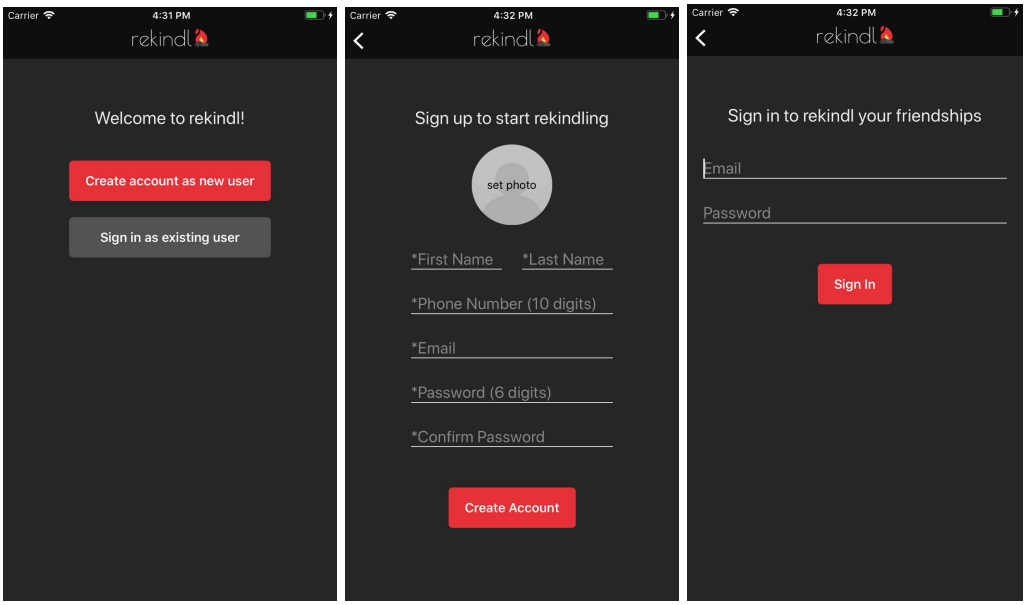



**Figure 8: Final Sign Up/Sign In Task Flow**

Once you sign in, you can perform a variety of operations. You can add a friend from your contacts (or enter their information manually), as shown in Figure 3. The app loads your contacts in from the Contacts app, and populates a list (alphabetical by first name) that you can also filter using the search bar at the top. After selecting a friend, it will pre-fill their information into the form, and you can edit anything you want and set how often you want to meet up with them, before adding them to your friends list.

You can select any of your connections to see more information about them, as shown in Figure 2. From there, you can text them, call them, or manually report that you have connected with them and your fire should be rekindled.

From your profile screen, you can set your status using the built in modal text window, as shown in Figure 1. You can also navigate to the settings page from your profile page, and manage your connections from there, as shown below in Figure 9.

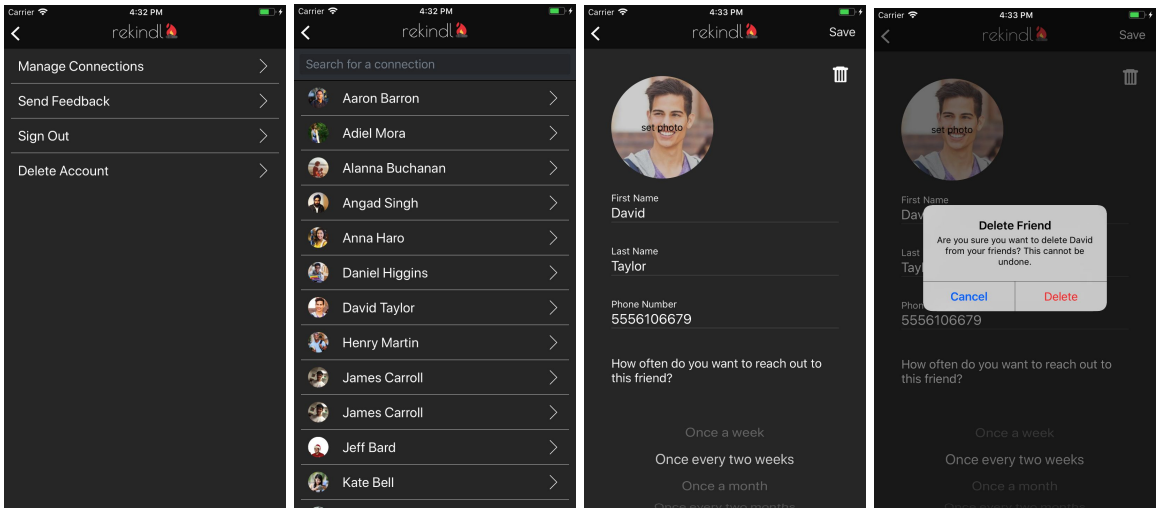



**Figure 9: Managing Connections**

The only other feature that may be worth mentioning in detail is the way the home screen works (Figure 10). When a user adds a friend, they select how often they want to reconnect with that friend. They will begin in the “vanishing” category, but reset to the “roaring” category whenever you Rekindle your friendship with them. The app will keep track of how long it has been since you last connected with each friend, and linearly move that friend back up through the categories as time passes. For example, a friend who you want to remember to reconnect with every week will progress through the categories much faster than a friend who you only want to reconnect with once a month.

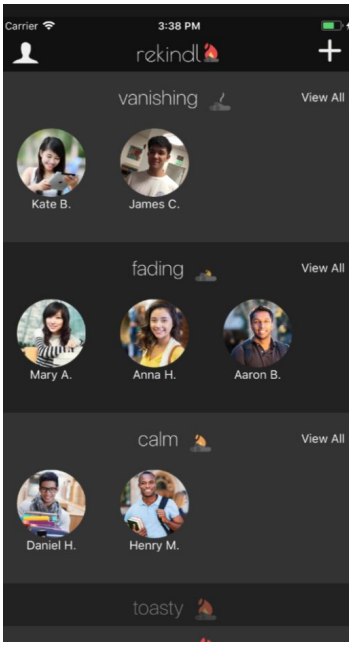



**Figure 10: Home Screen**

**Unimplemented Features:**
The greatest limitation with our final app is that, without our own back end server, there is not a good way for us to handle sending users push notifications. We configured Expo to be able to send notifications to the users, but did not find a way to do so without the user currently being in the app. We also looked into using Firebase, where we were storing our data, to send push notifications, but this feature of Firebase was not compatible with React Native apps still connected to Expo. If we were to continue with this project, we could eject the app from Expo and build it as a native iOS app and configure notifications that way, but that would involve changing many of our modules, as a good number of the elements we currently use are provided by the Expo framework and would not work if we ejected from Expo. There were no "Wizard of Oz" features in the app, with all accounts and friends implemented as would be the case in the production version of the app.

**Tools Used:**
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. There were also problems with generating a .ipa file to build it to potentially be put on the App Store.

We used Firebase for storing our data for all of our different users and their friends. This worked well because it allowed us to take advantage of Firebase's built-in user authentication and most features of it integrated well with React Native.

**Download Instructions:**
We built **rekindl** to be executed on an iPhone X running iOS 11. We did our best to make it responsive, but some layouts may not render quite as planned on older devices.

To open **rekindl** on mobile:

1) Download the Expo Client app from the App Store
2) Access the QR code to the prototype through this link (also available on our website): https://expo.io/@rcole34/rekindlapp
3) Scan the QR code in the Expo app, or press the "+" in the top right corner and enter "@rcole/rekindlapp".

## Making It Real

We are a team of Stanford undergraduate and graduate students studying computer science. We come from a variety of backgrounds and have a variety of interests, but we all have experienced the problem of growing apart from friends that we used to be close with, as time and busy schedules cause us to not always follow through on meeting back up with old friends. We brought together this shared experience and combined our passions for design, testing, and development to create Rekindl. We believe that this shared experience and our backgrounds in different fields of computer science have qualified us to create this app exploring the field of helping people to reignite their faded friendships.

Our full business model that we came up with for this class is shown below in Figure 11. To summarize some of the key points, our target market would be college students and recent graduates — people who have recently moved and have a lot going on with getting used a new style of life, who could easily otherwise lose track of old friends. These customers are also kind of people who would have smart phones and be accustomed to using them for other forms of social media. As we believe (and have confirmed through our testing) this is a very common problem for people in this market, there is a significant market size for our target demographic. While more ideas are detailed below, one of our favorite ideas for monetizing Rekindl is to sell ad space in the app to local restaurants, or other local establishments where people could go to meet back up with old friends. We believe we could have a big impact on our target demographic by helping people stay in touch with old friends and maintain a larger network, instead of falling out of touch with friends with whom we used to be close.

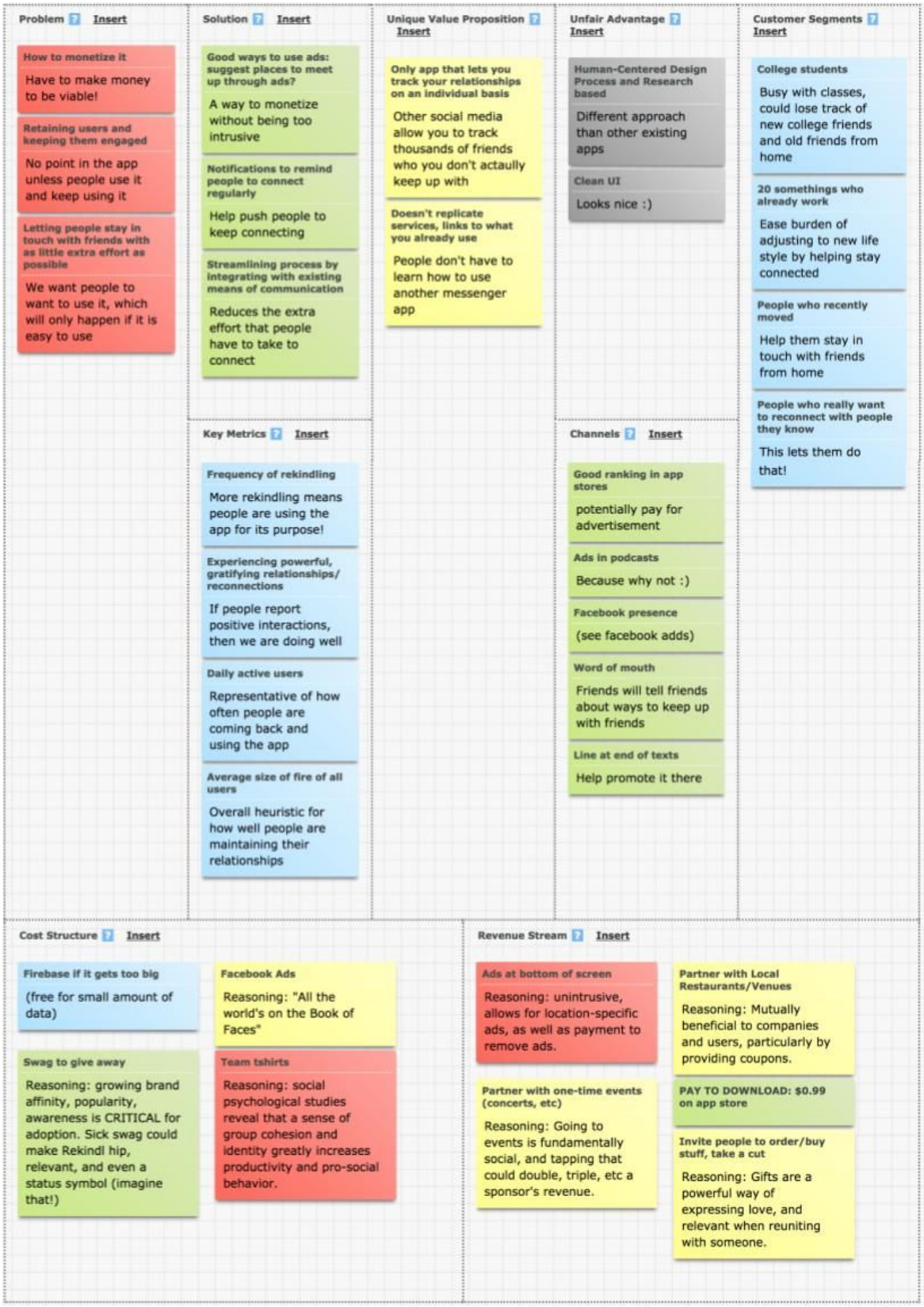



**Figure 11: Business Model Lean Canvas**

## Summary

Much of the technology today aimed at bringing local communities together offline is focused on *finding* new friendships and relationships (e.g. Tinder, Bumble), or on cultivating an artificial “public self” in order to extend the breadth of your network, rather than the depth of your existing relationships (e.g. Twitter, Instagram). 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 two quarters, 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 final high-fidelity prototype presented at the end-of-quarter project expo, we continued to both expand and refine our ideas for moving forward with our product, then build and prototype. Although our final product could still use a small amount of polishing in terms of implementing notifications and finding new components to use to be able to eject the app from Expo and build it for the App Store, we are proud of the polished product that we have created over the past two quarters.