

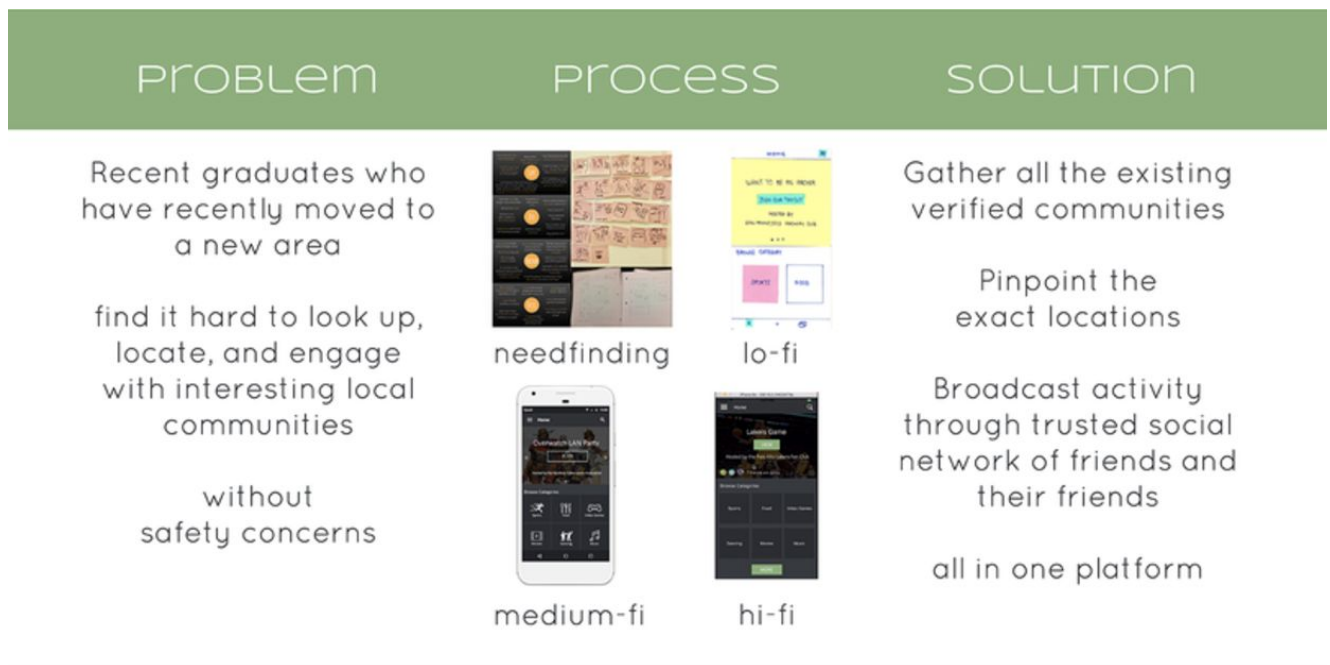
Interactive (High-fi) Prototype

Pakapark B. Stephanie T. Uzair I.



I. Problem and Solution Overview

Many people find it very hard to find, and participate in local communities that they're interested in without fear and concern of safety. With Quest, we ease all the process by gathering information on existing verified communities, pinpoint locations of meetings and events, and allowing users to broadcast their own activities or talents through their social circle all in one platform.



II. Final Tasks & Final Interface Scenarios

Task 1 (easy): Find existing events of interest

Users can search and select existing events of interest either by choosing the suggested overall popular activities or by category in the home screen. In case that users tap on the suggested activity, it will direct them to the detail of the event as well as a list of their friends going to event. Users can select the event by simply tap on 'going' button. On the other hand, in case that users tap on category of their interest, the program will list all subcategories in alphabetical order. Once users tap on one of the subcategories, the program will list all the activities in that

subcategories in an descending order of the number of friends going to the event. Once users choose the activity, it will direct to the activity detail page. In addition, users can unjoin the event by simply tapping on 'unjoin' button.

We select this task because it is the most simple and frequent problem our users have and it could be easily solved by the use of online platform.

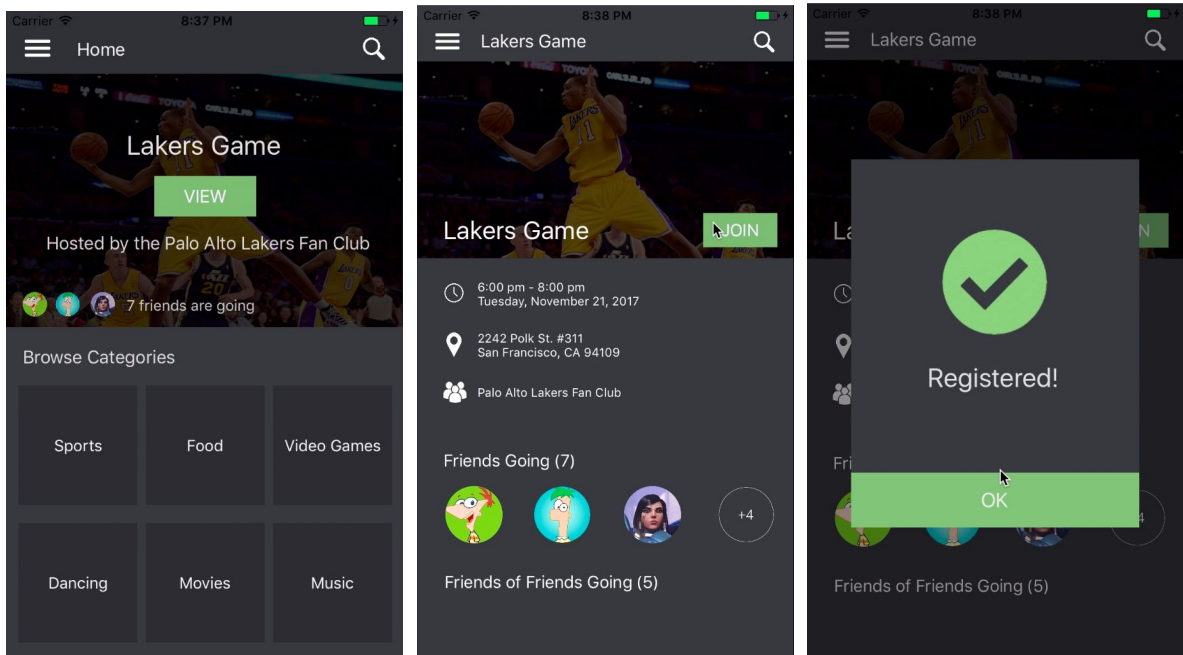


Figure 1: Simple Task Flow by tapping on the suggested activity

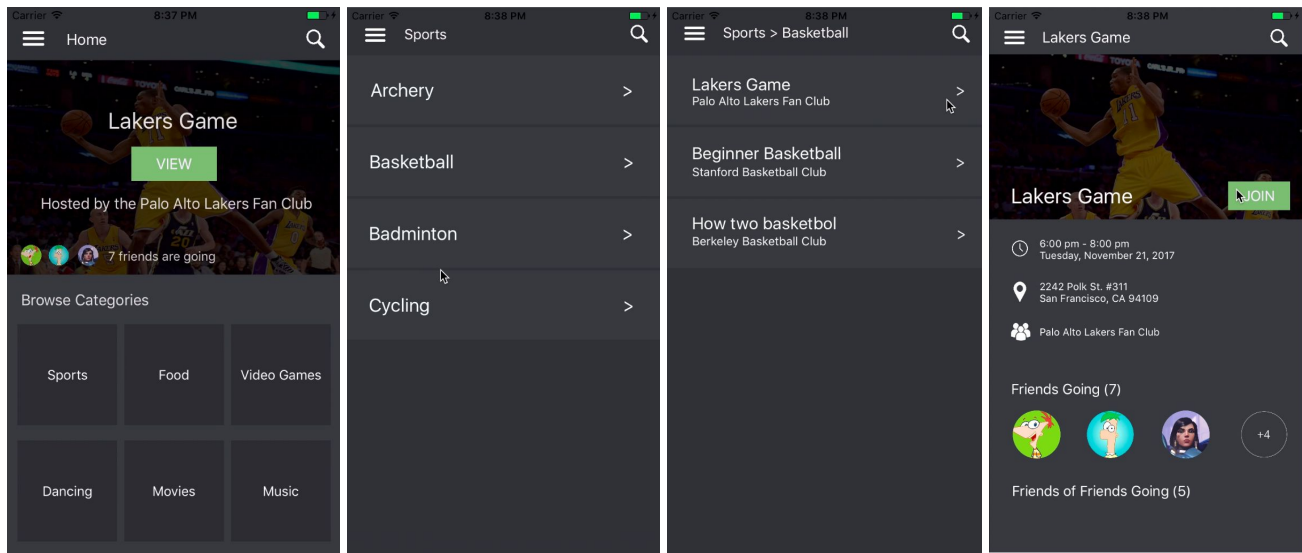


Figure 2: Simple Task Flow by selecting category of interest

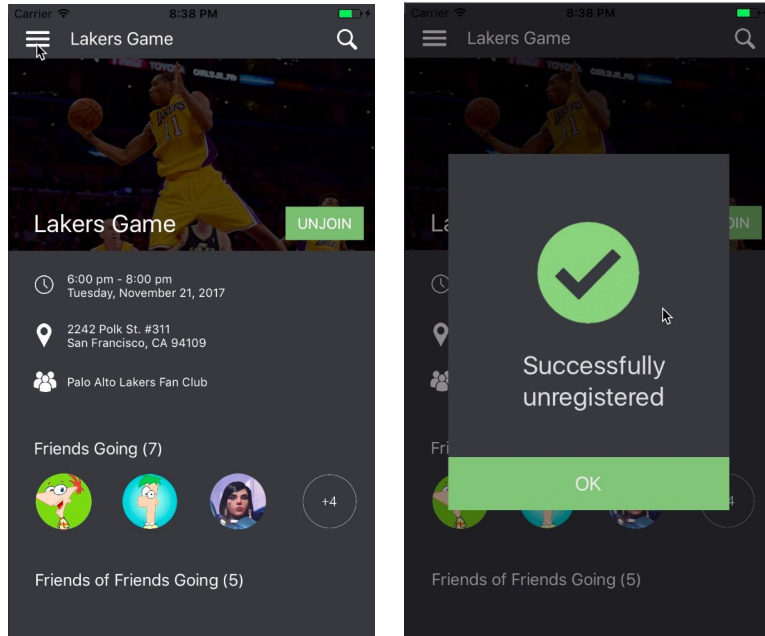


Figure 3: (Simple Task) Unjoin the activity

Task 2 (moderate): Tag along to clubs/events with friends

To tag along to an event with friends, users can tap on on the friend's profile on the event page which will have a pop up of basic detail of that friend. In this pop up, there are two buttons: 'view' and 'tag along' buttons. To tag along to the current event, users can tap on 'tag along' button and tap on 'yes' button in the verification page to confirm their action. On the other hand, if users want to see all events that a specific friend is going to, they can tap on 'view' button. This will direct users to a friend's profile page which contains a list of events that he or she plans to go. Users can simply tag along button in this page. In addition, users can cancel the tag along action at any time by clicking 'requested' button located at the exact location as 'tag along' button previously.

We select this task because it provides a shortcut for users to reach more clubs/activities in their local community and help users connect with each other in a very efficient way.

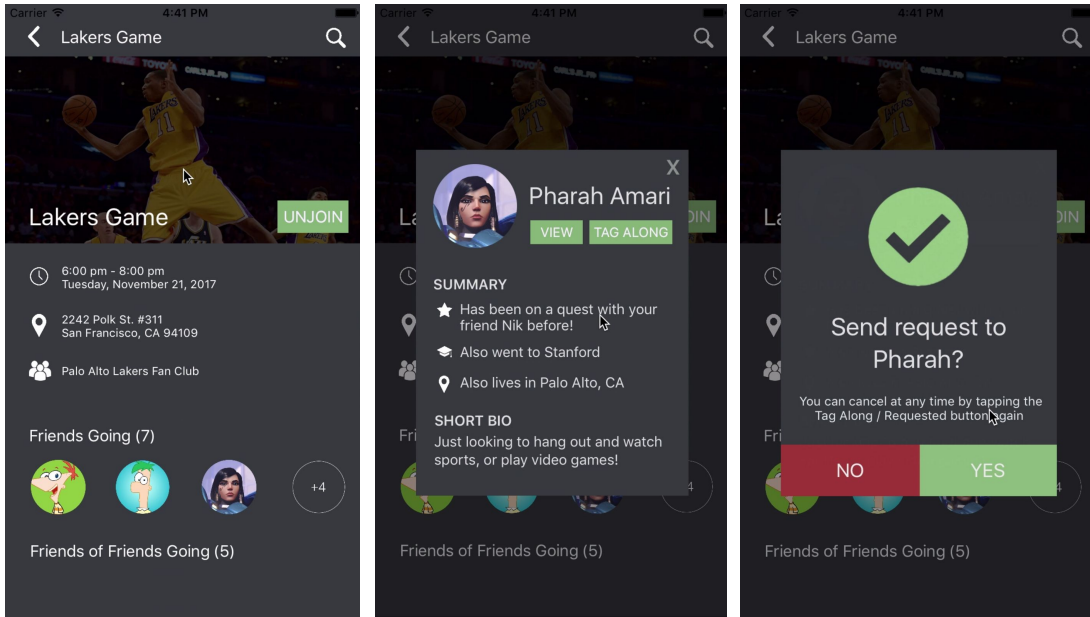


Figure 4: Moderate Task Flow by tag along with friends from event page

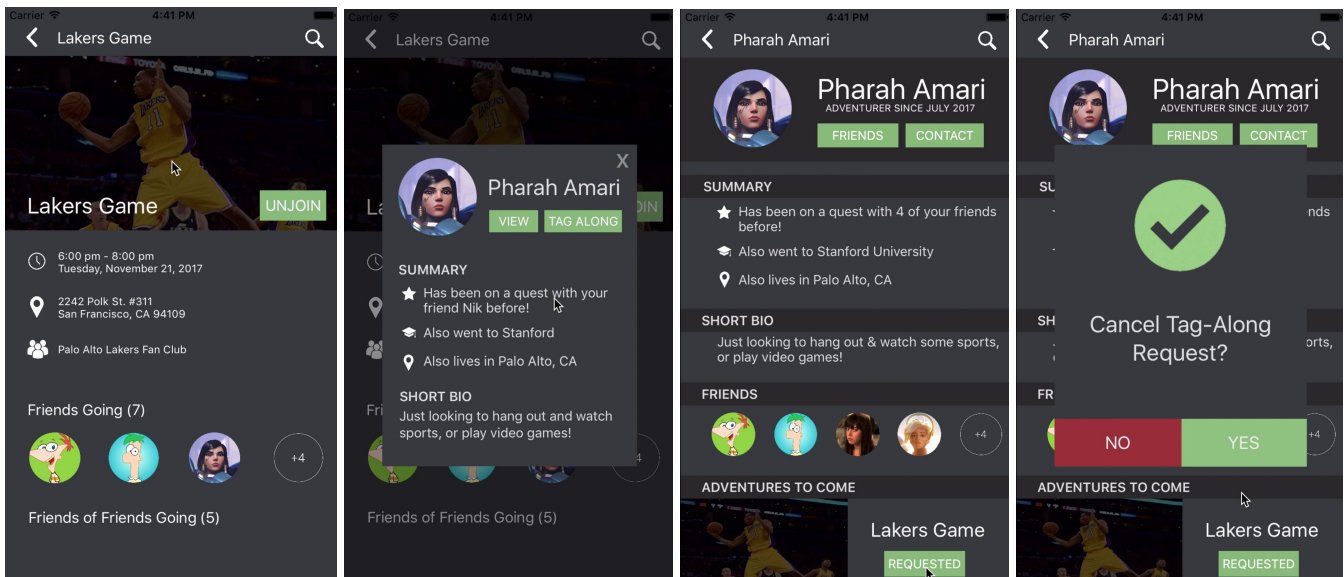


Figure 5: Moderate Task Flow by tag along with friends from friend's profile page

Task 3 (complex): Create your own club

Users can swipe to the right to open the menu bar. Then, to create a club, users tap on manage clubs which will direct to the page showing a list of clubs they currently have or request. In case that the users don't have a club, they can tap on the '+' button located either at the middle or the lower right corner of the page. This will direct users to the club creating request form. Simply fill in the information of the club including its name, description, location and image. Users must also select at least five friends from their friend lists for a request to be a club founder.

After finishing selecting the friend, users submit the request until reach the confirmation page. Then, the club request will appear in the manage club page.

This is not a frequent task as users are more likely to want to find activities and others to do activities with (i.e. task 2 and 3) than they feel the need to create their own club.

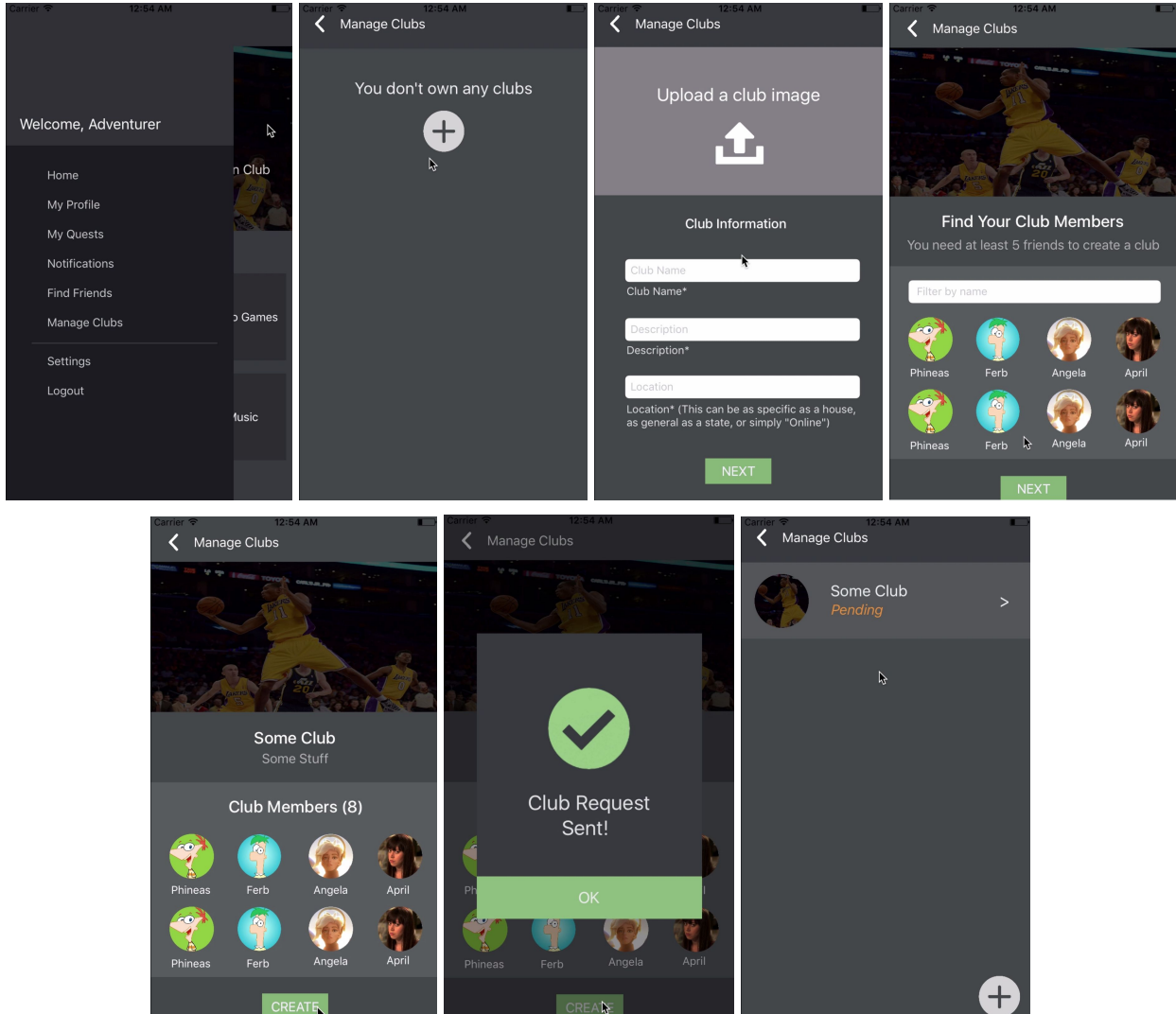


Figure 6: Complex Task Flow

III. Design Evolution

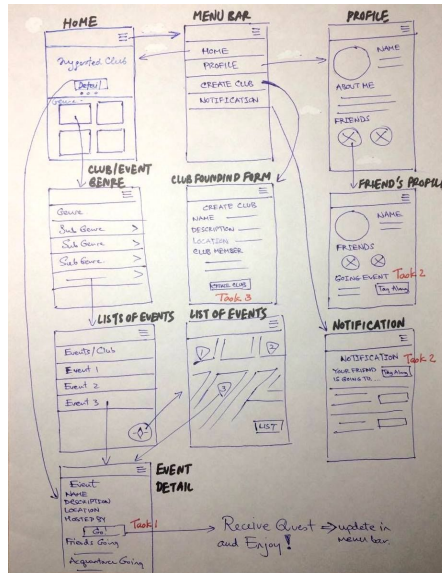


Figure 7: Initial UI Sketch

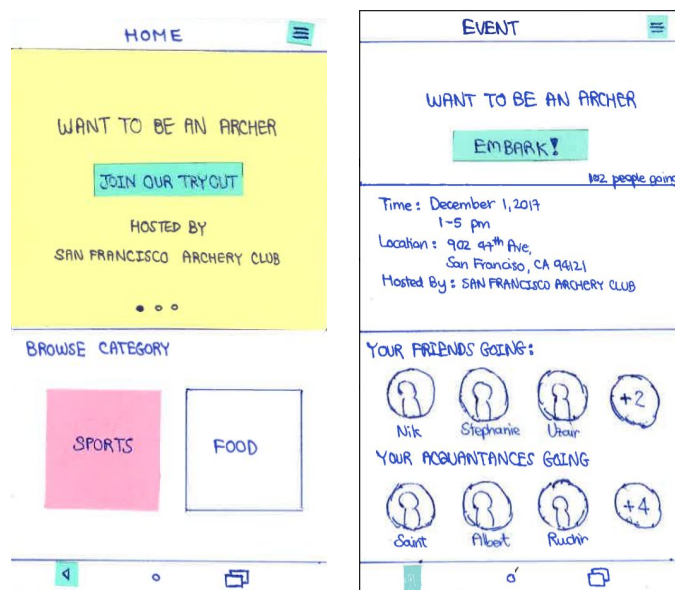


Figure 8: Simple Task Flow For Low-fi Prototype (by selecting the suggested activity)

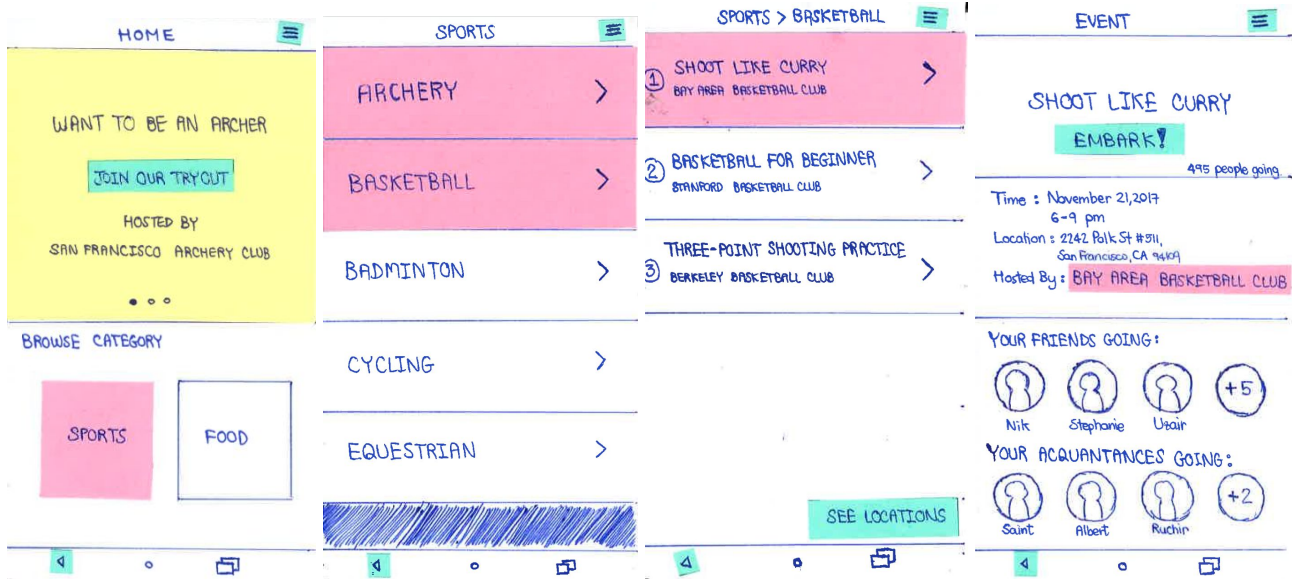


Figure 9: Simple Task Flow For Low-fi Prototype (by selecting category of interest)

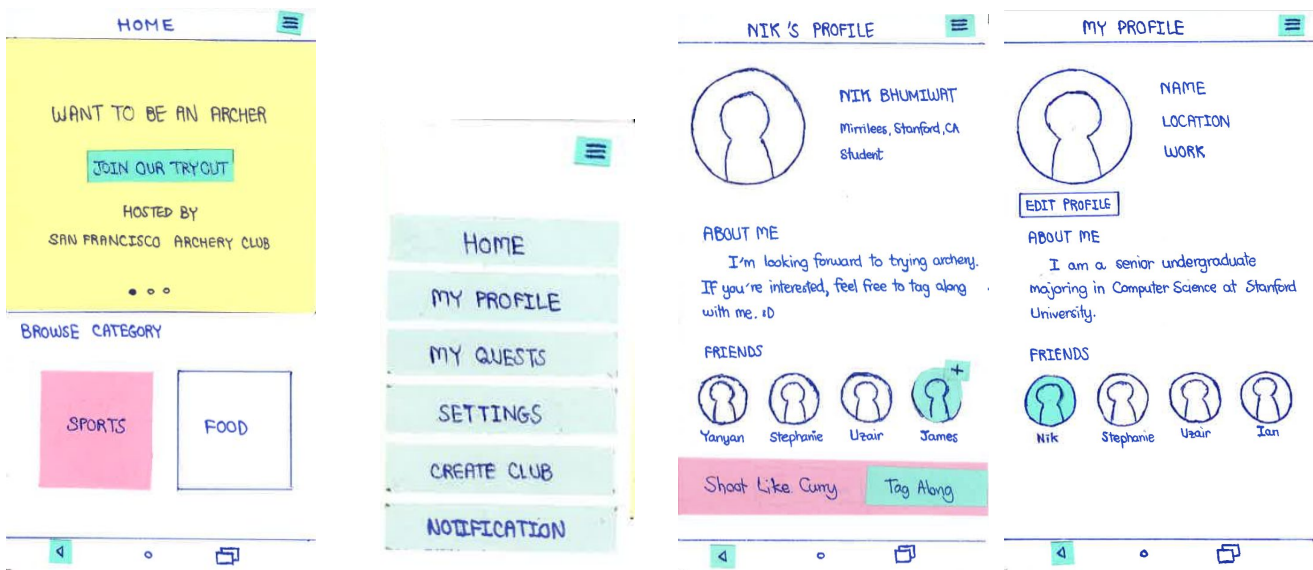


Figure 10: Moderate Task Flow For Low-fi Prototype (By Friends)



Figure 11: Moderate Task Flow For Low-fi Prototype (By Notification)

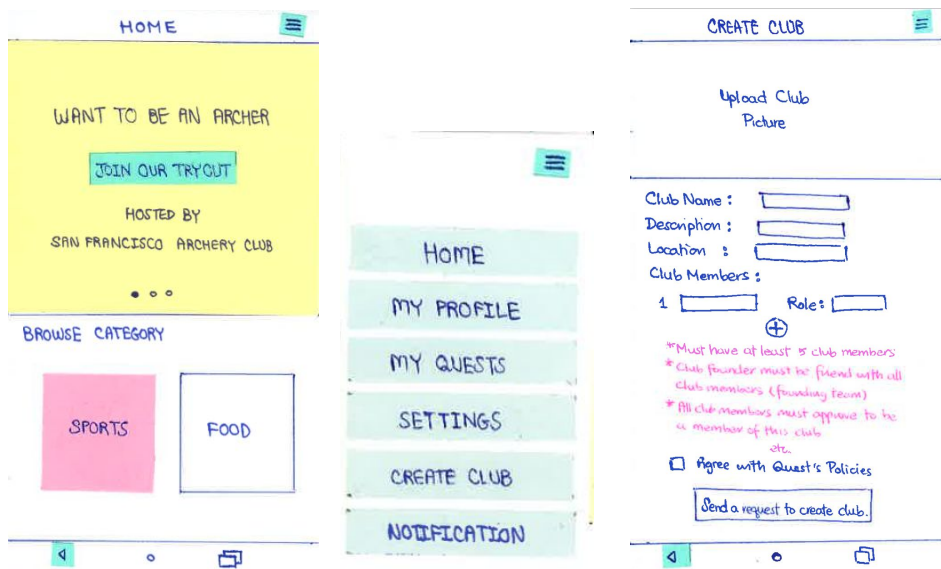


Figure 12: Complex Task Flow For Low-fi Prototype

In our initial UI sketch (Figure 7), we envisioned that our three tasks could be separated into three parts of the application. We decided that the simple task would be in the homepage since it's the most frequent task for most users. Also, we believed that it was important to allow users to switch across tasks and go to every other part of the application (moderate and complex tasks) by going to the menu bar. We keep this feature both in low-fi prototype and medium prototype as shown below.

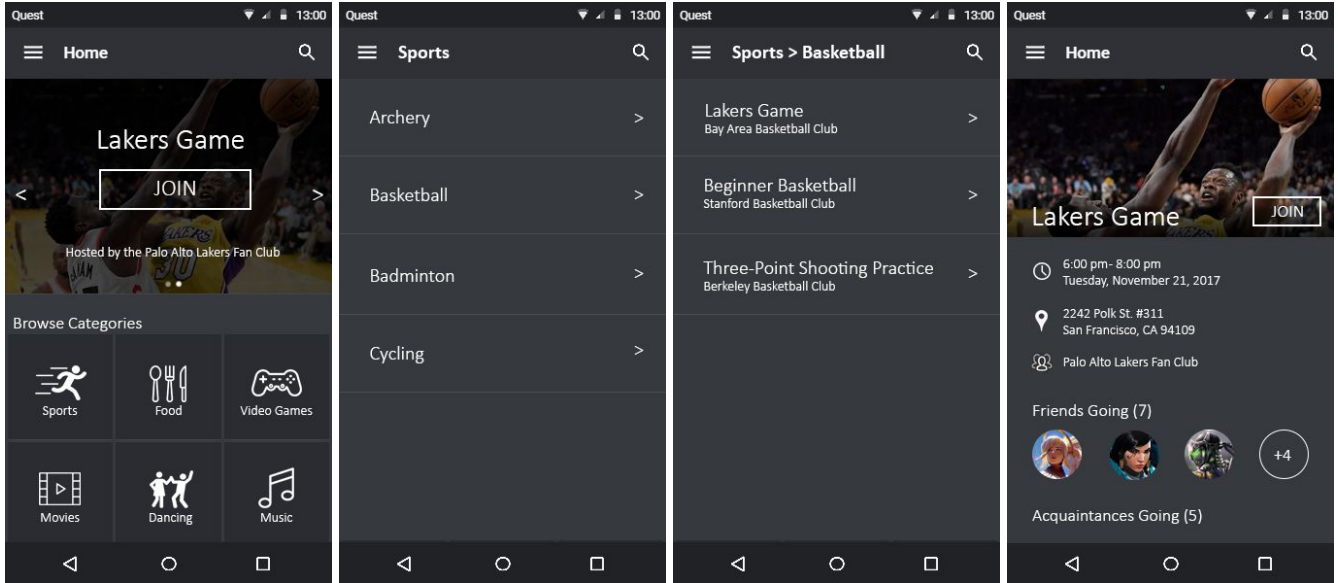


Figure 13: Simple Task Flow For Medium-Fi Prototype

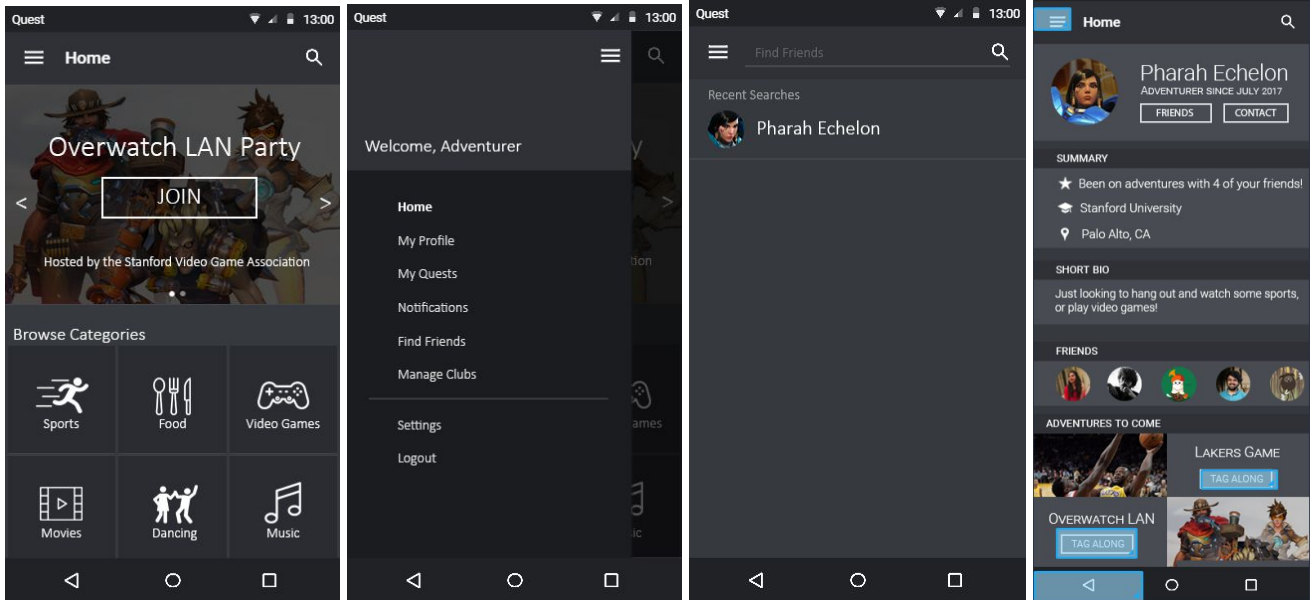


Figure 14: Moderate Task Flow For Medium-Fi Prototype (By Find Friend)

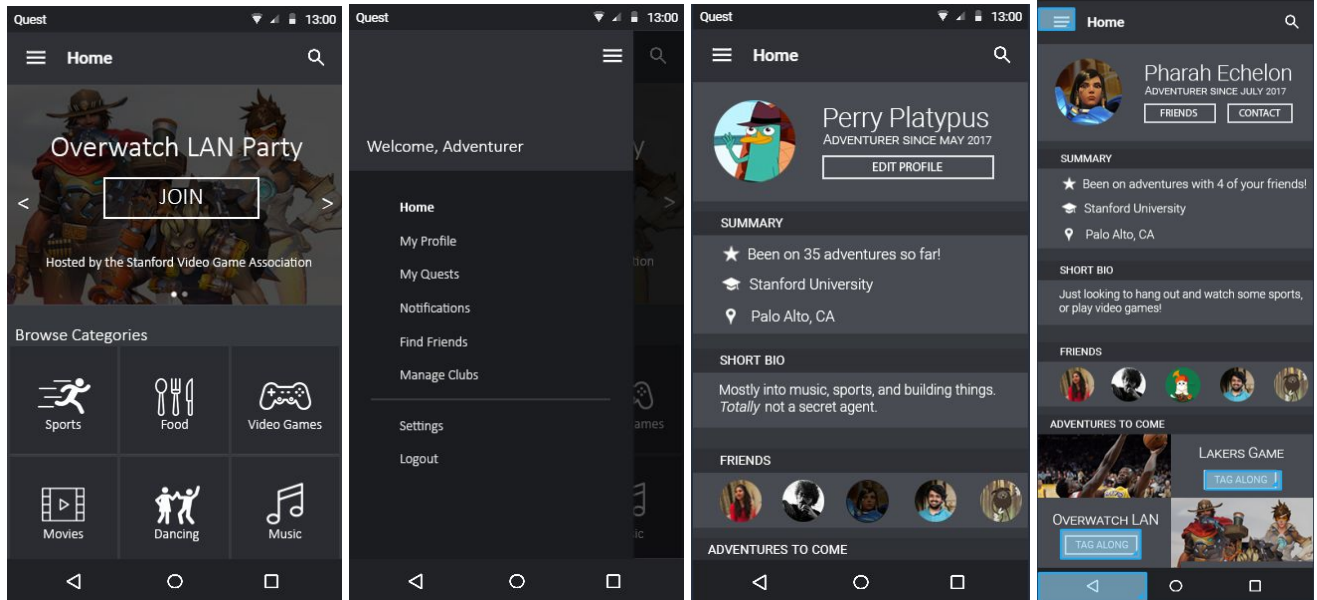


Figure 15: Moderate Task Flow For Medium-Fi Prototype (By My Profile)

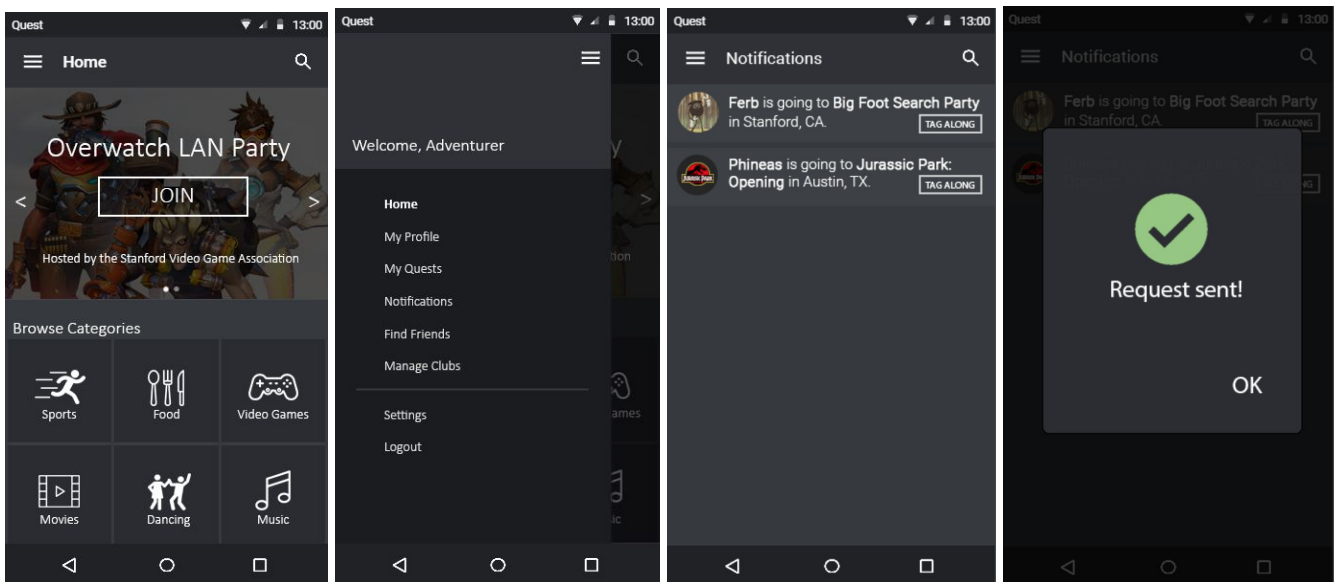
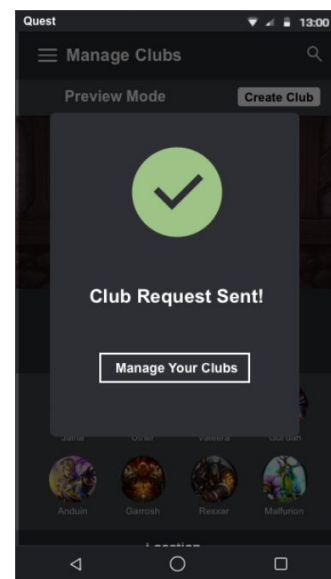
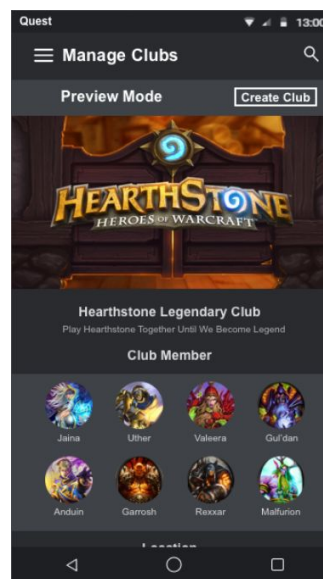
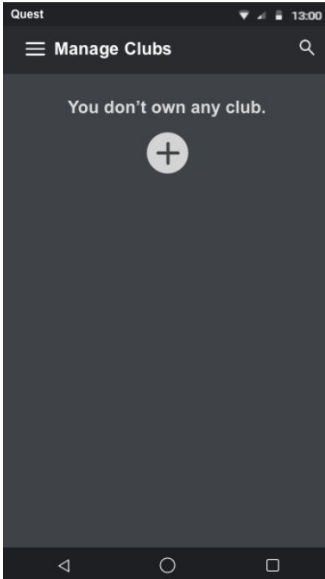
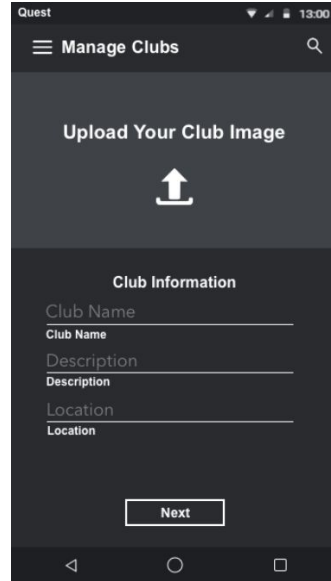
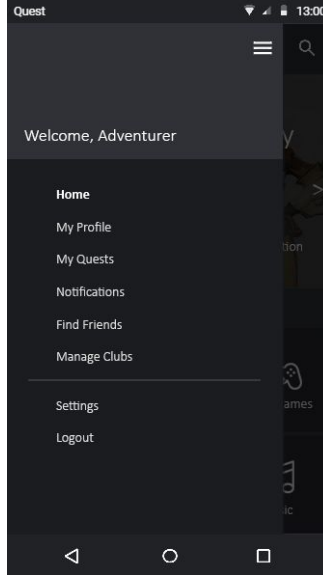
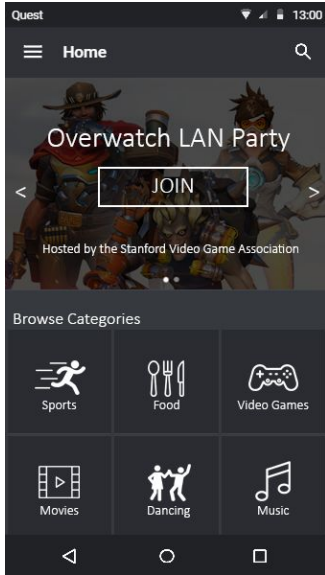


Figure 16: Moderate Task Flow For Medium-Fi Prototype (By Notifications)



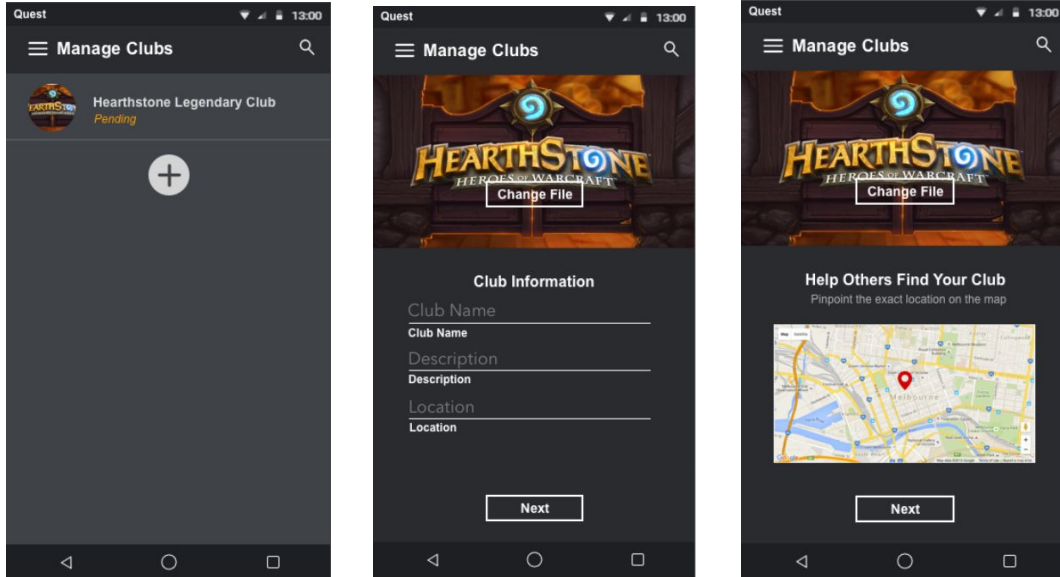


Figure 17: Complex Task Flow For Medium-Fi Prototype

IV. Design Changes (From Low-Fi Prototype to Medium-Fi Prototype)

We changed the design from the low-fi prototype to medium-fi prototype as follow:

1. Top Bar (Menu Button, Headline Text)

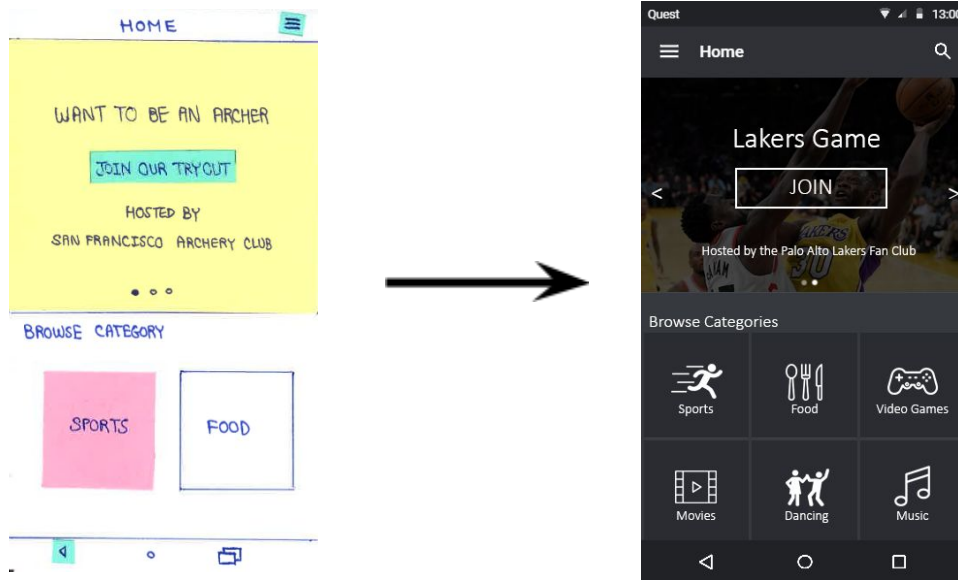


Figure 18: Top Bar Design Change – Menu Button and Headline Text

We observe that many recent applications put the menu bar on the left side of the screen because users typically read from left to right and up to down. Setting the menu bar and headline text at the top left corner is easier to find than putting in at the top right corner.

2. “Embark” Button to “Join” Button

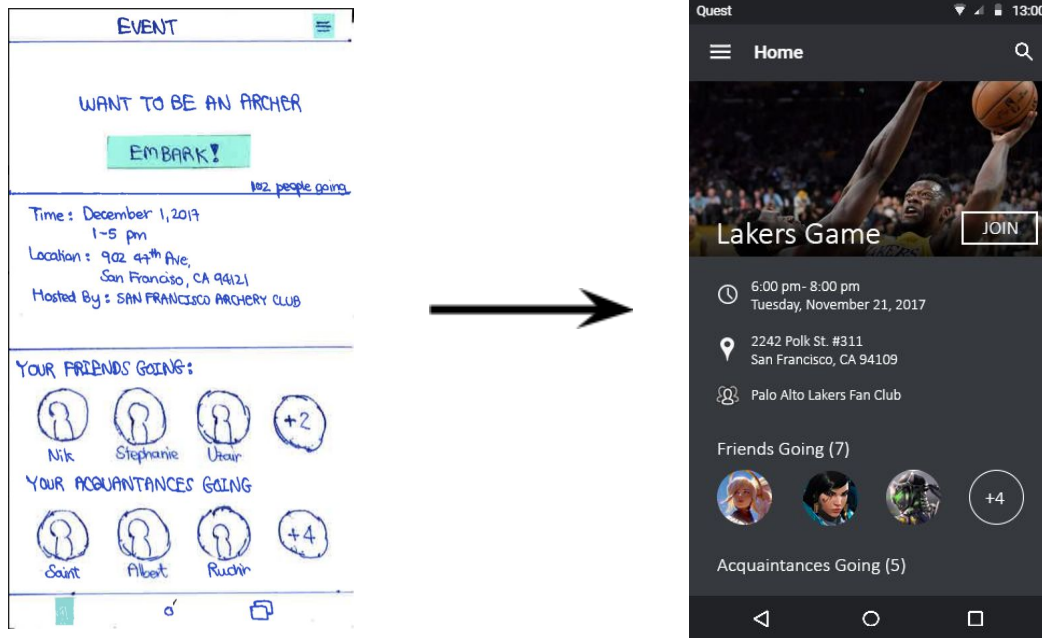


Figure 19: Button Change – Embark button to Join Button

From our low-fi prototype testing, almost all participants do not understand the meaning of “embark” in this context. This might be because this word sounds like a quest in the video game, not an activity quest in real life. Changing the word “Embark” to “Join” button make it much easier to understand the the general audience.

3. Remove Search By Location

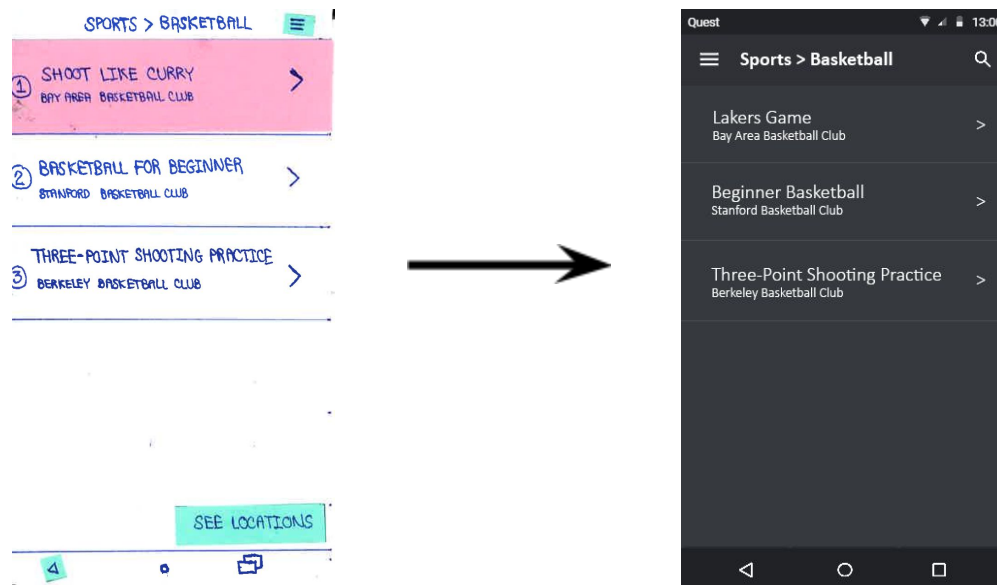


Figure 20: Usability Change – Remove Search Event By Location

This idea might originally sounded great since there were many research supporting that people usually wanted to see things visually much more than things as a text. However, during the low-fi prototype testing, it turned out that no participant decide to tap on location button since it's less intuitive to select the event by location than by interest.

4. More Menu Options

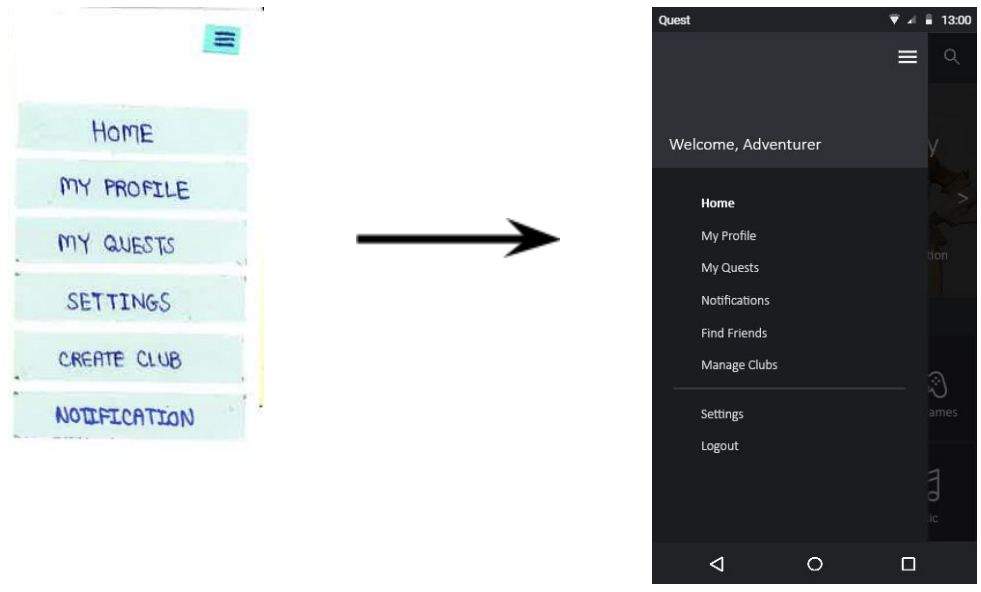


Figure 21: Menu Option Change – Add Find Friends and Logout

We also added 'Find Friends' and Logout' buttons in the menu bar. From our low-fi prototype testing, we observed that most participants could not find friend's profile page from the menu bar make it difficult to tag along to an event with friends. Adding this button makes the task more completion. Moreover, the 'Logout' button is important when multiple users are sharing a phone (as is especially common among families in emerging economies), or want to log into their profile using someone else's phone. Thus, adding these two options in the menu bar seemed to be an essential for our application.

V. Major Usability Problems Addressed

Our major usability problems (level 3 and 4):

1. User does not know which friends are on the service.

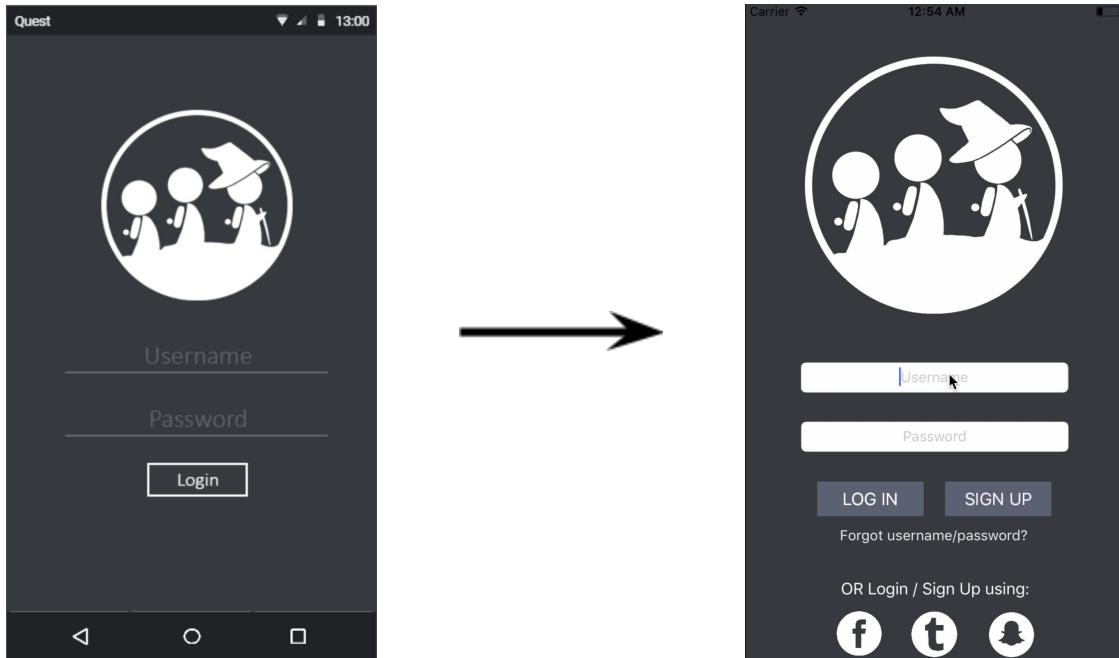


Figure 22: Add Login With Other Social Platform

Our medium-fi prototype does not allow users to login with other social media platform like Facebook, Twitter, and Snapchat. This makes it difficult to connect with friends that already had in the social media in our platform. In the hi-fi prototype, we add three more options as shown at the bottom of the login page: login with Facebook, login with Twitter, and login with Snapchat.

2. User cannot unjoin an event.

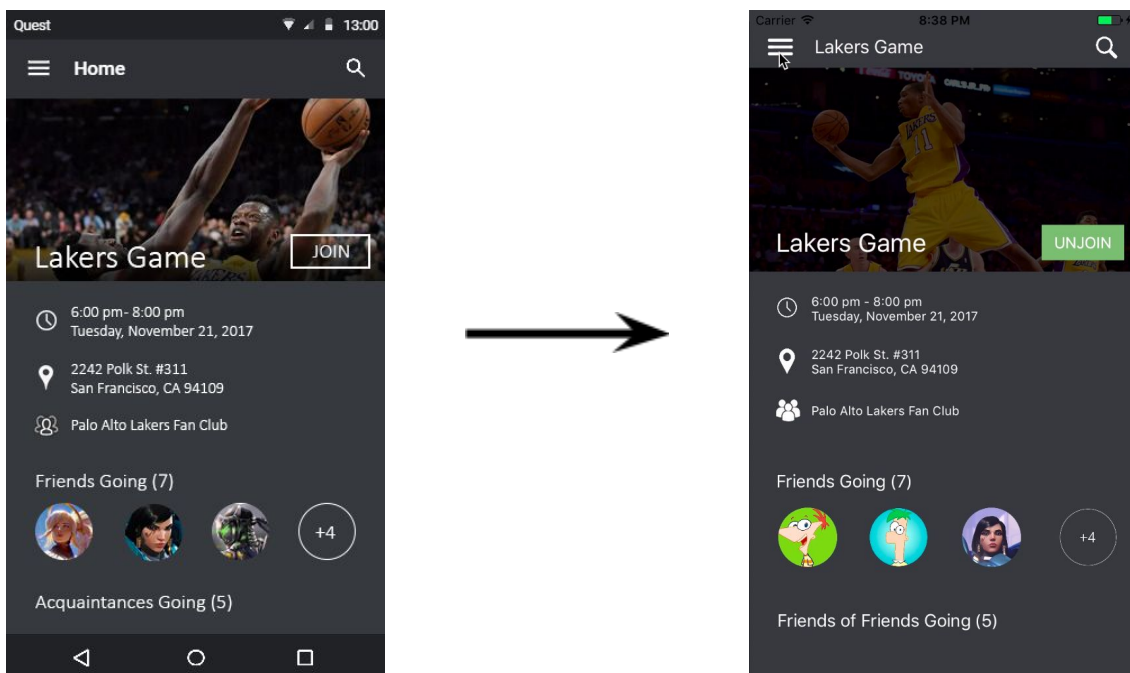


Figure 23: Allow Users To Unjoin An Event

In our medium-fi prototype, users are not allowed to unjoin the event. Thus, users cannot undo the action in case that they unintentionally tap on the join button. Moreover, users are not allowed to view a list of events that they are going. In response to these issues, our hi-fi prototype allows the users to unjoin the event at any time. However, we do not have a page that user can view all the events that they are going since it does not directly relevant to the tasks and not as important like Facebook's events.

3. User cannot join a club

In our low-fi prototype, our application allows users to join a club, but we deprecate this feature for the medium-fi prototype. We believe that in order for a user to join a club, it's a responsibility for club founders to decide. However, in the future, we plan to have a notification about a new upcoming event from a club that user already visited.

4. Other Changes Made

Another issue frequently brought up in the heuristic evaluation (severity 1 - 2) is the lack of confirmation screens and system information, which made it difficult for users to understand what happened if they misclicked buttons. The hi-fi prototype includes yes/no confirmation screens when sending and canceling tag along requests to friend (Figure 4). Additionally, the buttons also change label after an action has been made to indicate how to undo an action (Figure 5). We also made other labeling changes to use words users would be more familiar with, such as Friends of Friends instead of Acquaintances, as well as other button labels to be more representative of their actions (e.g. view featured event vs. join button). Profile wording is also more consistent across screens.

VI. Prototype Implementation

Tools:

- We primarily used XCode to design for iOS devices, and utilized Adobe Photoshop to tweak designs and icons wherever necessary
- **Helped:** XCode's native Interface Builder mode was very helpful because it allowed us to tweak the User Interface very closely, duplicate screens when the need arises, and to ensure that the presentation of the app was exactly as we wanted it
- **Helped:** XCode also makes it very easy to attach code to certain parts of the view (for example, you can tie a boolean variable to a button to represent whether or not it has been tapped). Since we do not have a server API or database, this made it easy to execute our "Wizard of Oz" technique for keeping track of whether or not the user has joined an event yet, and it also allowed us to pass data between views.
- **Did not help as much:** One way in which XCode did not help as much was that there are certain UI-related tasks which can oddly only be performed through Swift code attached to the view, for example, making the tappable area of each button bigger. Since none of us had used Swift before, this significantly slowed down

the creation of our views, at least initially, since many small changes unexpectedly did not have a front-end way of executing them.

Wizard of Oz Techniques:

- We assume we have some AI available to make initial recommendations for events based on interests.
- The app keeps track of whether a user has joined an event, or chosen to tag along with a friend, and then updates the buttons for these accordingly. For example “Join Event” changes to “Unjoin Event” and “Tag Along” changes to “Requested”
- When the user creates a club, the app stores the data they entered in a static variable, and uses this to display their new club in the “Manage Clubs” page.

Hard-coded Data:

- As we do not yet have an API server, or a database, all events and profiles in the prototype are hardcoded for now.

What is Missing and What Might you Add:

- In a fully-featured version of this app, we would also add club pages, the user’s own profile (including their activity log), and an initial on-boarding flow which explains ‘tagging along’ and accessing the ‘hamburger menu’.
- On the technical side, we would add an AI mechanism to recommend events, and an API/database to fetch and store data.

VII. Summary

Even though there are many social sites and platforms has been created during the past decades, many people from all walks of life still find it hard to look up, locate, and engage with engrossing local community without safety concerns. We hope that, with Quest, everyone would be able to easily look up local communities of interest, bridge the social gap among people in the same vicinity, and learn more about their fellow locals while bonding over shared interests with ease. Moving forward, we would like to make our application more complete and pick up the principles of UX, and design as well as the lessons in platform creation we have learned from creating Quest, to adapt and apply to our future works.