



Huddle Team



Mike Weingert
Developer



Nadav Lidor
Manager



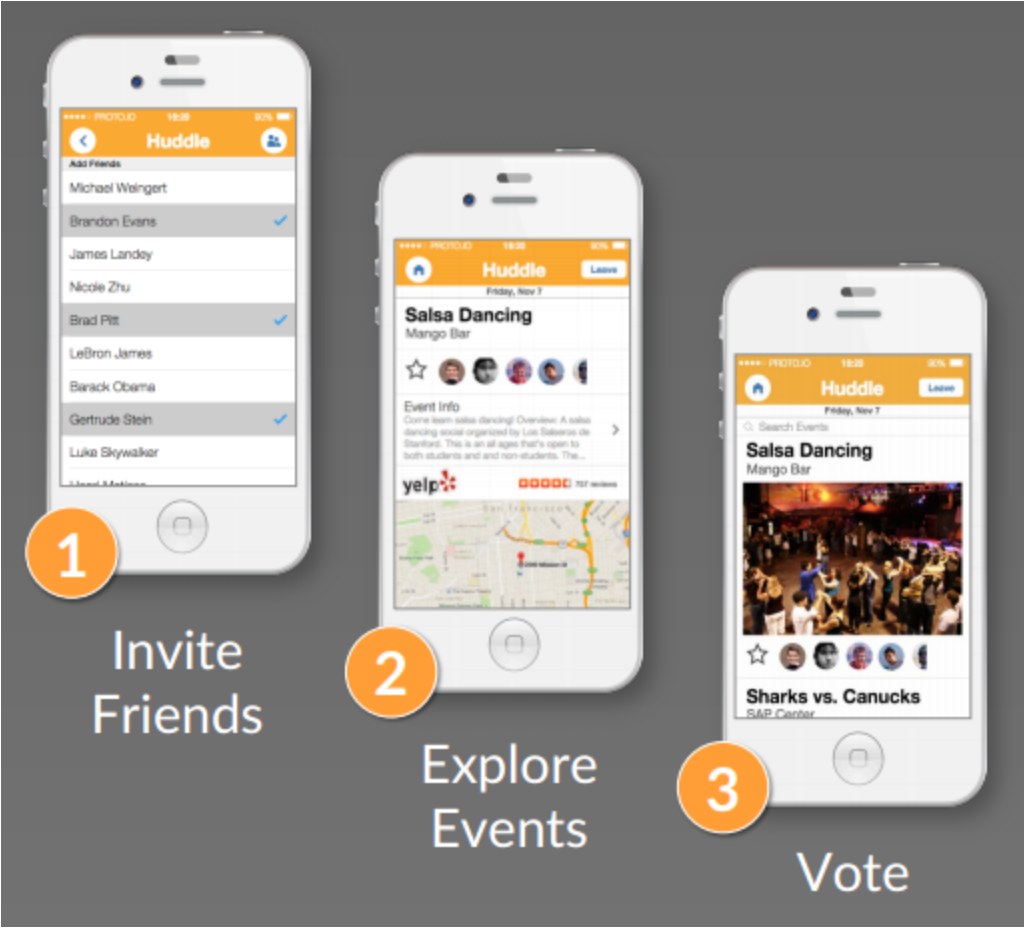
Joe Polin
User Testing /
Documentation



Brandon Evans
Design

Problem and Solution Overview

People get stuck in familiar routines because familiarity is comforting. As a result, they miss out on discovering new possibilities that may enrich their lives. This problem is compounded by the pains of planning and organizing. Scheduling friends, finding common availabilities, and booking reservations are just a few of the headaches that dissuade people from exploring and discovering new things. Huddle eliminates many of these headaches by proposing events based on a user's preferences and seamlessly organizing and scheduling outings with friends. The hassle of communicating with multiple friends and trying to get everyone to agree on one event becomes quick and painless because Huddle acts as a third party mediator of sorts. It guides everyone to a common decision through a structured and fair voting process.



Tasks & Final Interface Scenarios

Task 1. Coordinating and Scheduling a Group Event (Complex)

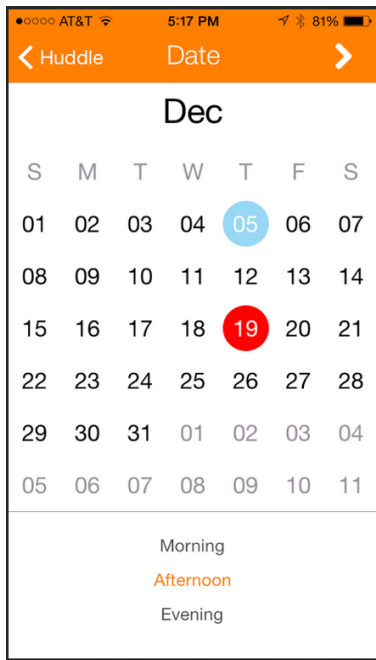
The user wants to hang out with his or her friends on a particular day and time. The user decides the day and time and invites friends to hang out. As a group, the user and his or her friends communicate their preferences for different activities. We chose this task because it's complex and requires interacting with almost all of the screens in our application.

Task 1 Storyboard

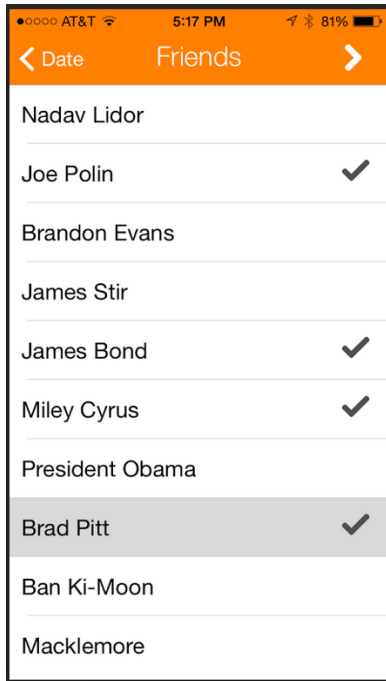
Step 1. Tap on the plus icon in the top-right corner.



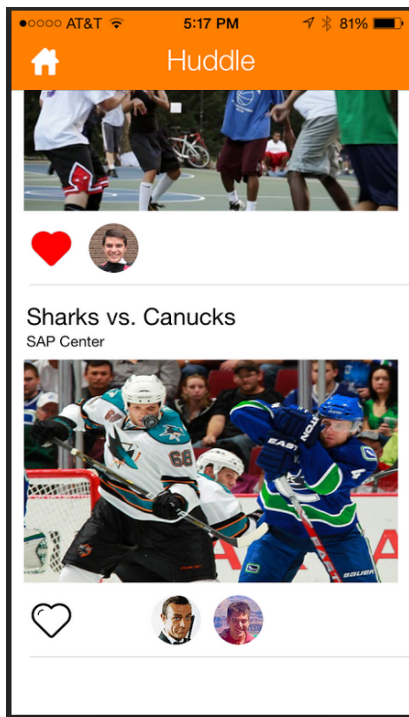
Step 2. Tap to select date and time, then tap arrow in top-right corner.



Step 3. Select which friends to invite and tap on the arrow in the top-right corner.



Step 4. Participate in voting by tapping the heart icon for activities you like.

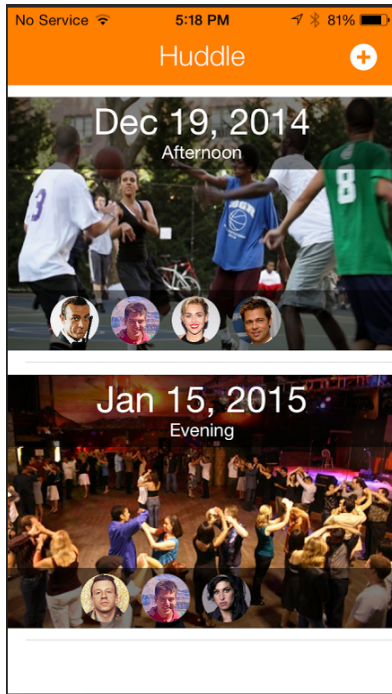


Task 2. Indicating Activity Preferences (Moderate)

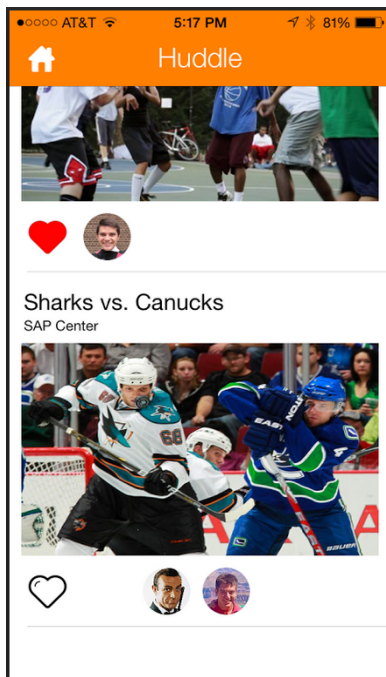
A user has been invited to hang out with a friend. The user indicates his or her preferences for activities. This task ensures that we also consider the experience of the users who receive invites to a Huddle. We need to make sure the voting process is straightforward and engaging.

Task 2 Storyboard

Step 1. Tap on corresponding Huddle tile.



Step 2. Tap heart icon for activities you like.



Task 3. Leaving a Huddle (Simple)

A user accepts an invitation to an outing. The user later decides not to participate and informs their friends that they will not be attending.

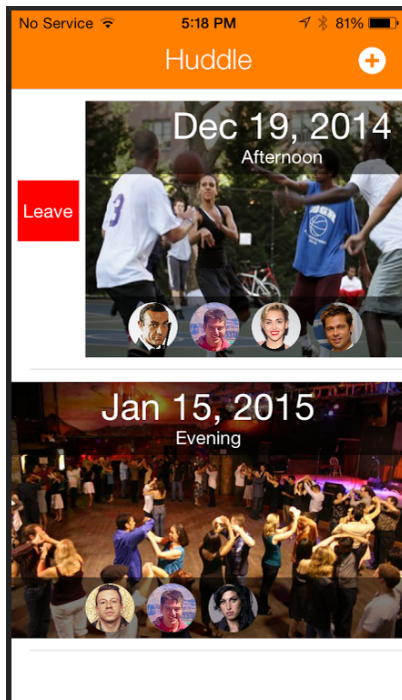
This task is a quick, simple task that does not require interacting with many screens. Implementing this task also completes the cycle of adding and deleting Huddles to the home screen, which significantly improves the smoothness/intuitiveness of the prototype.

Task 3 Storyboard

Step 1. Swipe right on Huddle tile.



Step 2. Tap 'leave'.



Major Usability Problems Addressed

Due to the simplicity of our design, the heuristic evaluation did not report many significant usability violations. The evaluators reported only two violations of severity level 3 and zero violations of severity level 4. In addition, we ultimately decided that neither level 3 violation necessitated a fix for the current prototype. Our high-fi prototype therefore focused on making the app more visually beautiful and engaging along with minor usability tweaks.

Lack of Friends List Organization - Severity 3

Our evaluators wanted the ability to organize friends, to make the process of finding and inviting friends quicker when you have a large list of friends. We ended up not addressing this complaint in our high-fidelity prototype because we felt that it was not a significant issue in terms of implementing our core tasks. Also, unlike Facebook, Huddle is an app primarily for communicating with friends you will actually interact and meet up with. We envision friends lists being primarily small lists of close friends. Therefore, the issue of navigating a massive list of friends is unlikely to come up for our typical user. That said, including a “best” and/or “recent” section for friends would make the process of inviting friends more efficient. It is a feature that we would add when further building out our high-fidelity prototype.

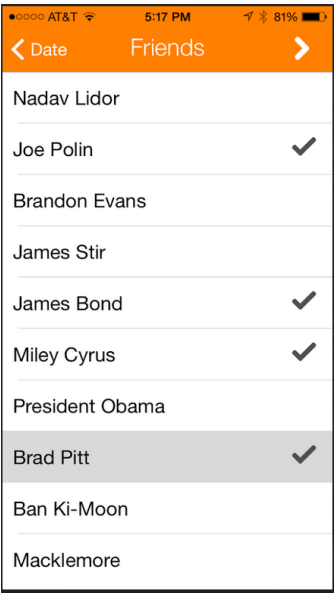


Figure 1. Friend Invite Screen.

Invite Uncertainty - Severity 3

Evaluators felt it was unclear what happens when a user gets invited to a Huddle. They wanted our prototype to include notification messages and to show things like whether a friend can decline or accept an invite. While this is definitely a nice feature for a future build of our prototype, we felt that it was not a necessary feature for the core functionality we envisioned for the high-fidelity prototype.

No Way of Un-leaving a Huddle - Severity 2

Our evaluators wanted a way to “undo” leaving a Huddle. Getting invited to a Huddle is the only way of joining a Huddle, so if a user leaves a Huddle and changes their mind later, there’s no easy way of re-joining.

We decided this was not a major issue because the actions needed to leave a Huddle make it unlikely that a user will accidentally leave a Huddle. In addition, the lack of a ‘rejoin’ feature enforces commitment. An issue that came up repeatedly during our contextual inquiry was the frustration people had when people failed to show up on time or were otherwise uncommitted to an outing. By not allowing people to change their minds later and rejoin a Huddle, we encourage commitment among the members of a Huddle.

Following that principle of encouraging people to stay committed to a Huddle, we changed the ‘leave’ button into something that shows up when you swipe right on a Huddle tile on the homepage. By not keeping the ‘leave’ button in plain site, we remove it as a constant visual presence that may prompt some users into leaving a huddle.

Unnecessary Warning Text for Leaving a Huddle - Severity 1

Heuristic evaluators felt that the warning dialog for leaving a Huddle was too verbose for such a simple and self-explanatory action. We decided to address this by removing the warning dialog and having changing the leave button into a hidden button that you reveal by swiping a Huddle tile to the right.

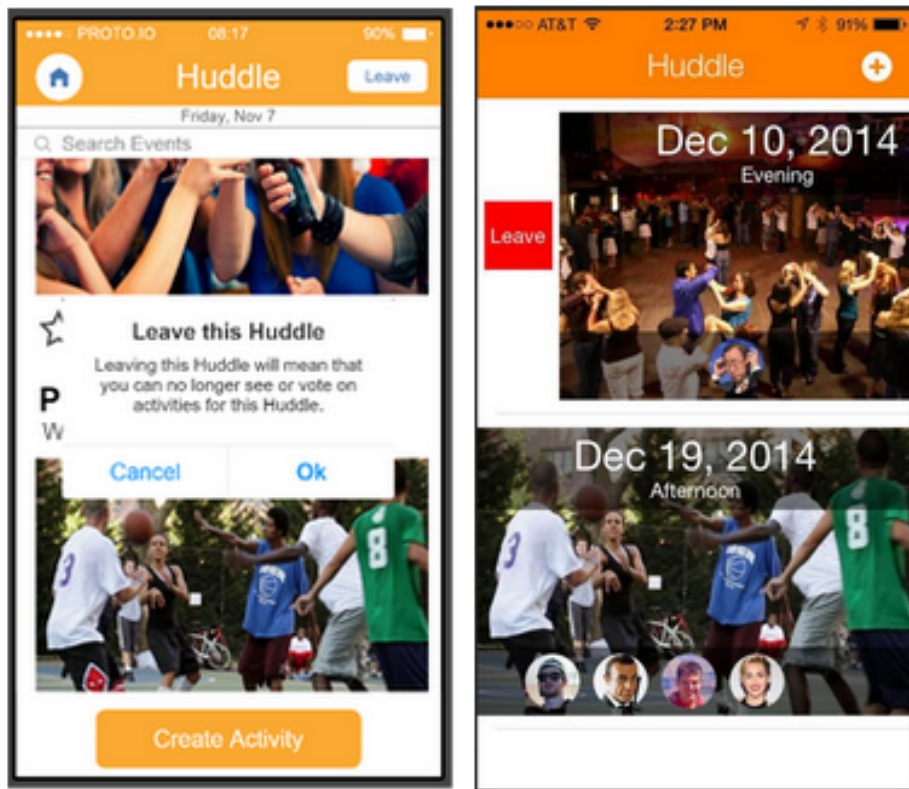


Figure 2. Old Leave Button (Left) Vs. New Leave Button (Right)

“Leading” and “Suggested” Activities - Severity 1

Evaluators did not see enough distinction between these sections to warrant their existence. We agreed and decided to remove these sections for our high-fidelity prototype. Our activities are now in a single, continuous list of image tiles. The screen is now simpler and more engaging.

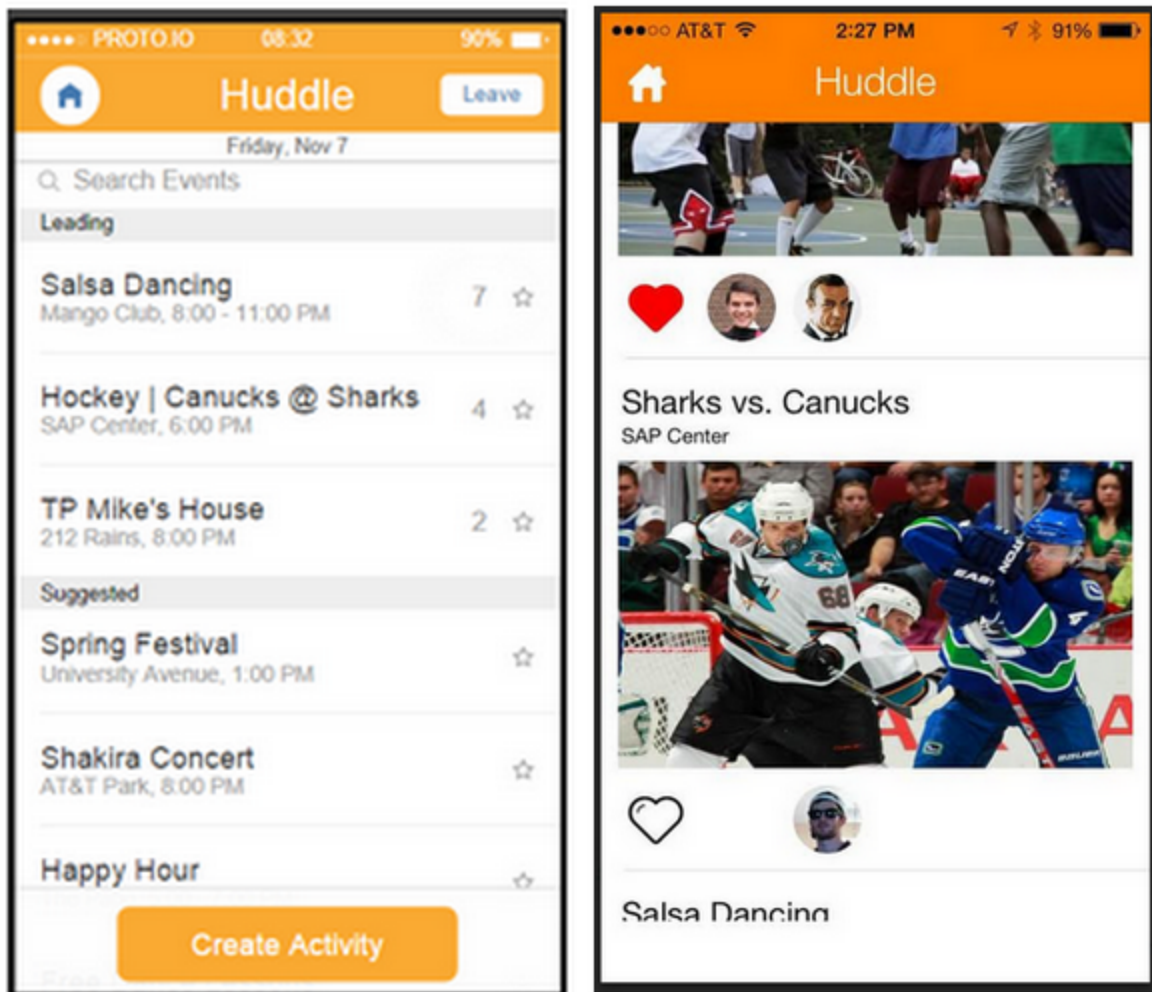


Figure 3. Old Activities Page (Left) Vs. New Activities Page (Right).

Inability to Specify Huddle Time - Severity 2

Evaluators wanted the ability to specify a time of day for a Huddle rather than only being able to select the date. We addressed this by adding buttons that allow the user to indicate whether the Huddle will occur during the morning, afternoon, or evening.

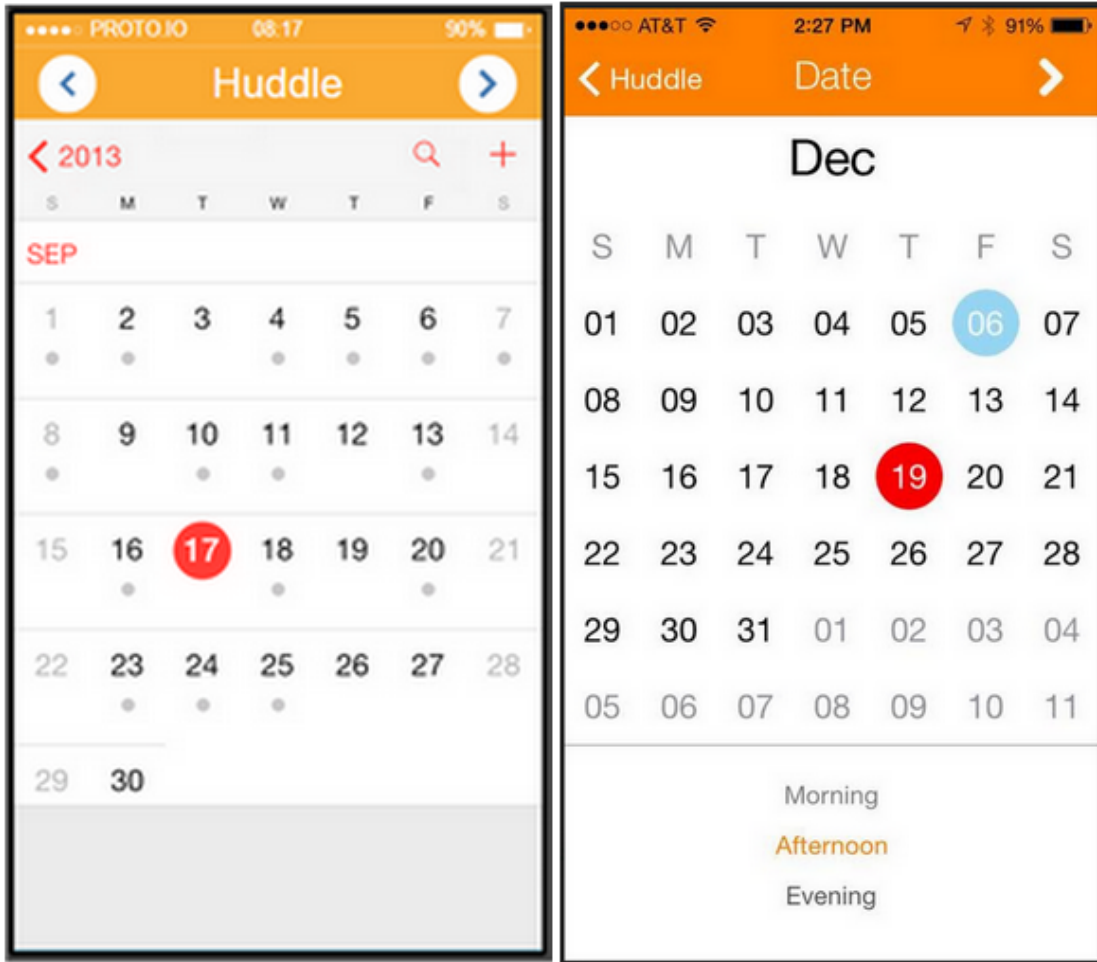


Figure 4. Old Calendar (Left) Vs. New Calendar (Right).

Home Screen Too Boring (Studio Leader Feedback)

Our original medium-fidelity home screen was a text list, similar to our original low-fidelity prototype sketches. Because our app is aimed at making people excited to try new things and go out with friends, we updated our design for our high-fidelity prototype to be more centered on interesting images.

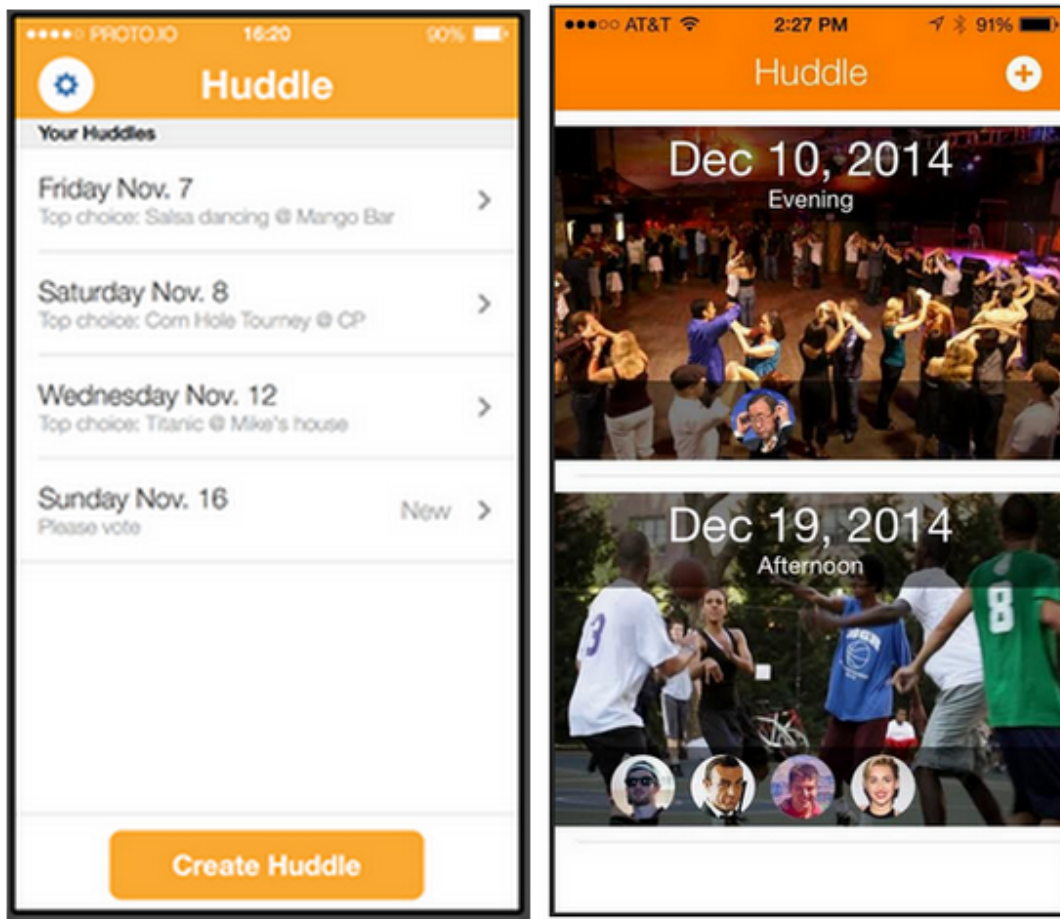


Figure 5. Old Home Screen (Left) Vs. New Home Screen (Right).

Event List Too Boring (Studio Leader Feedback)

We also received feedback on the lack of visual interest on our events list screen, so we updated it with a similar focus on large images. As with the home screen, we included avatar icons to indicate which friends voted on which activities. This was a more interesting and socially-compelling way of showing an activity's popularity. It also gives users more information at a glance, since some users may have preferences for and against hanging out with certain people.

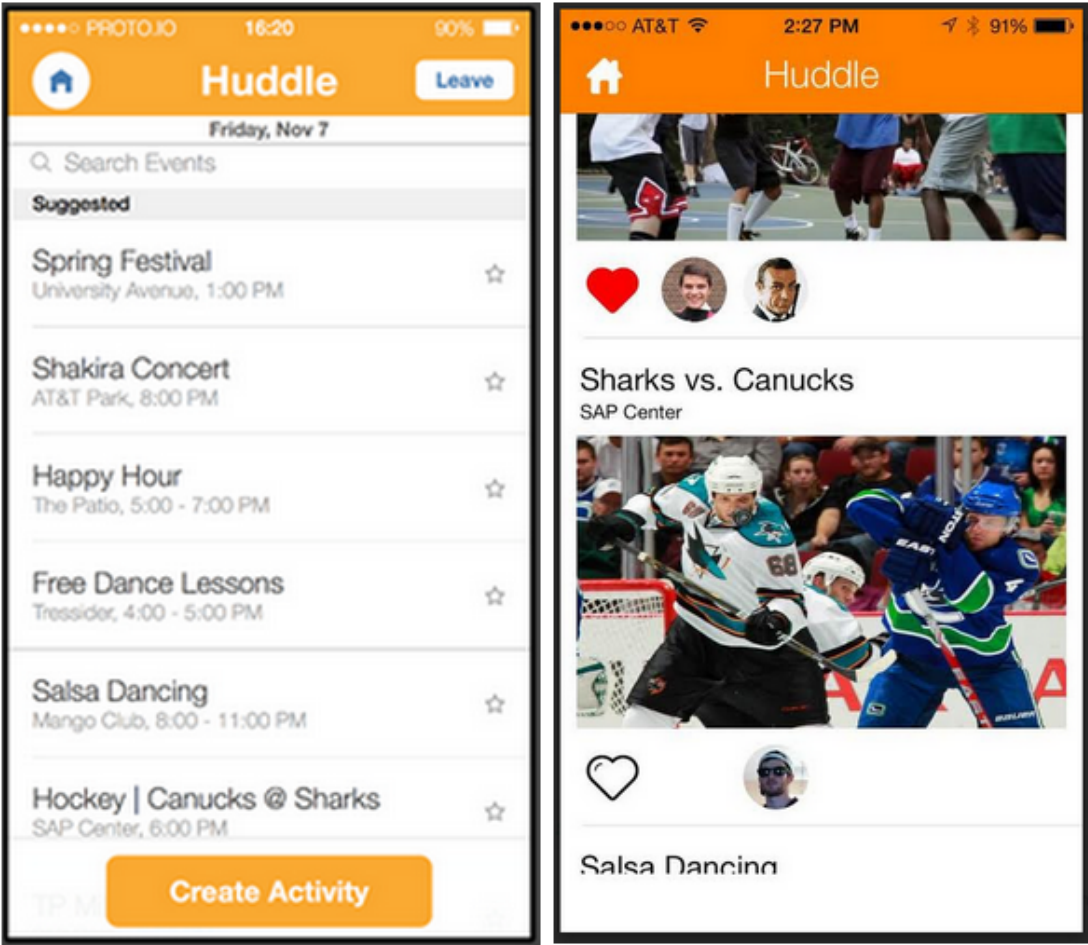


Figure 6. Old Activities Screen (Left) Vs. New Activities Screen (Right).

Design Evolution

Lo-Fi Paper Prototype

During user testing with this paper prototype, we found that users were satisfied with the overall structure and functionality with the app. We encountered minor issues where users were confused by some of the terminology we used and what certain buttons meant (“leave”, “vote”, “next”, “edit”).

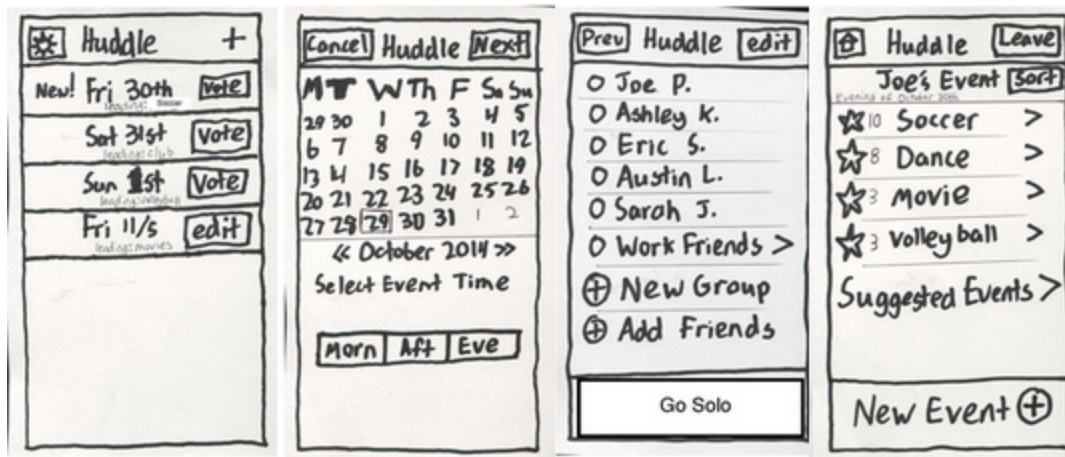


Figure 7. Low-fi Prototype Screens.

Medium-Fi Prototype

For this prototype, we focused on modifying the interface on multiple screens to reduce some of the confusion users had about what certain buttons did and what actions they could take on each screen. For example, we removed the “vote” buttons from the home page since users wondered what tapping that button actually meant and how it differed from tapping elsewhere on the listed Huddle (there was no difference). Instead, we simply had an arrow to indicate that tapping a Huddle takes you to another screen with more information. Our other changes primarily involved a similar approach where we simplified things to reduce confusion (removing the “new group” button from the friends list, combining “suggested events” with the main events page, etc).

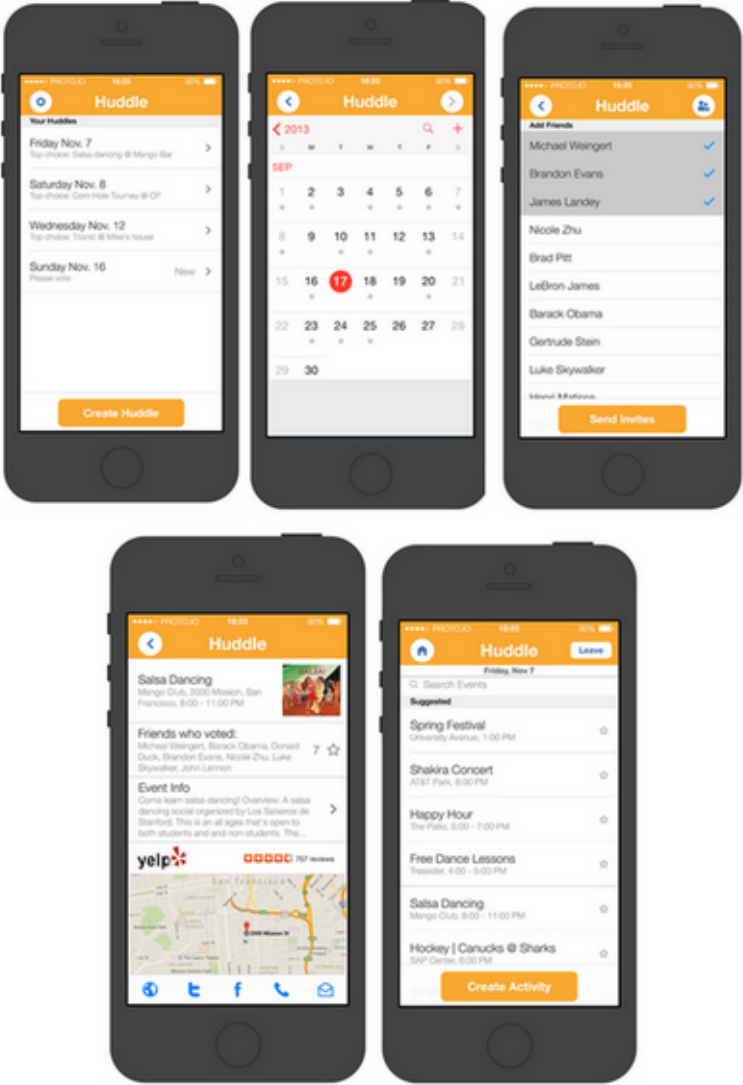


Figure 8. Medium-fi Prototype Screens

Medium-Fi Prototype 2.0

Our main goal with this updated version of the medium-fidelity prototype was to revise the main screens to be more visually interesting. Our original medium-fidelity prototype had a very straightforward set of text-centric screens that were functional but boring. We want users to be excited when they open the Huddle app and see the home screen. We want them to feel inspired to try new events when they see the images for those events in the voting screen. With that in mind, we replaced the old text lists for the Huddles and activities with big, vibrant images that capture the sense of adventure and fun we want people to feel when using Huddle.

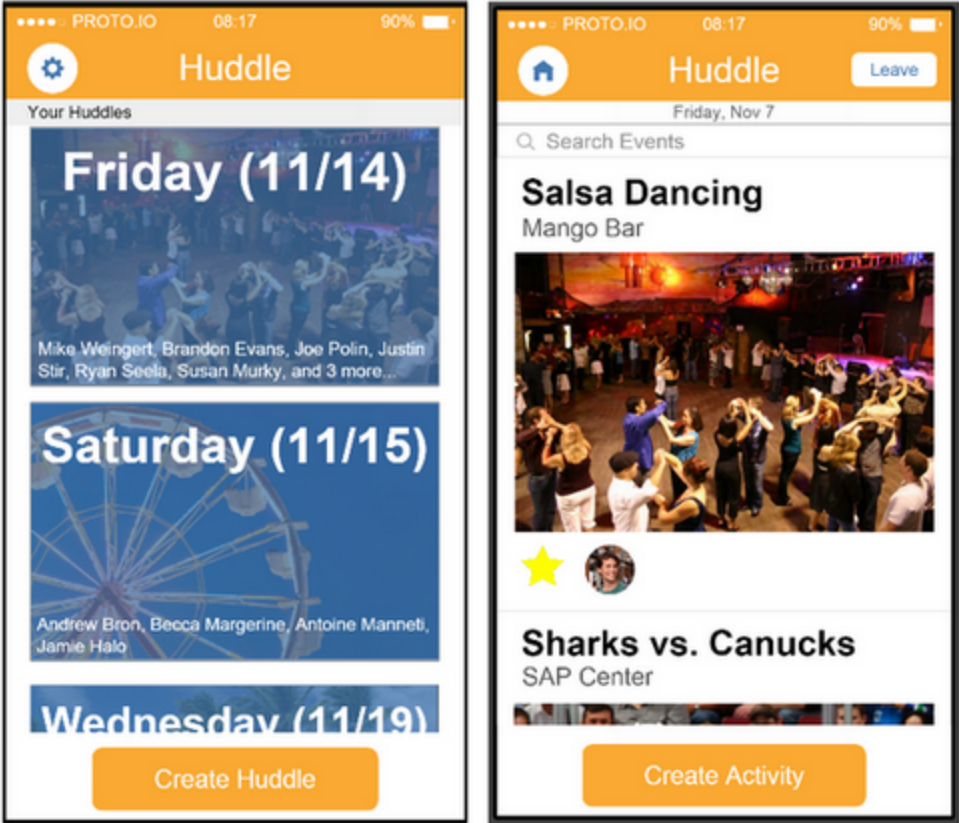


Figure 9. Medium-fi Prototype 2.0 Screens

Hi-Fi Prototype

Install Link: <http://install.diawi.com/GxiZh1>

For the high-fidelity prototype, we wanted to further refine the visual design to correct some of the issues that popped up during feedback sessions with our studio leader and from a meeting we had with someone from IDEO. The blue overlay we used in our medium-fidelity prototype ended up making the home screen's images less vibrant and interesting than they would be without the blue filter. We also decided to show which friends were in a Huddle through avatar icons rather than boring text names.

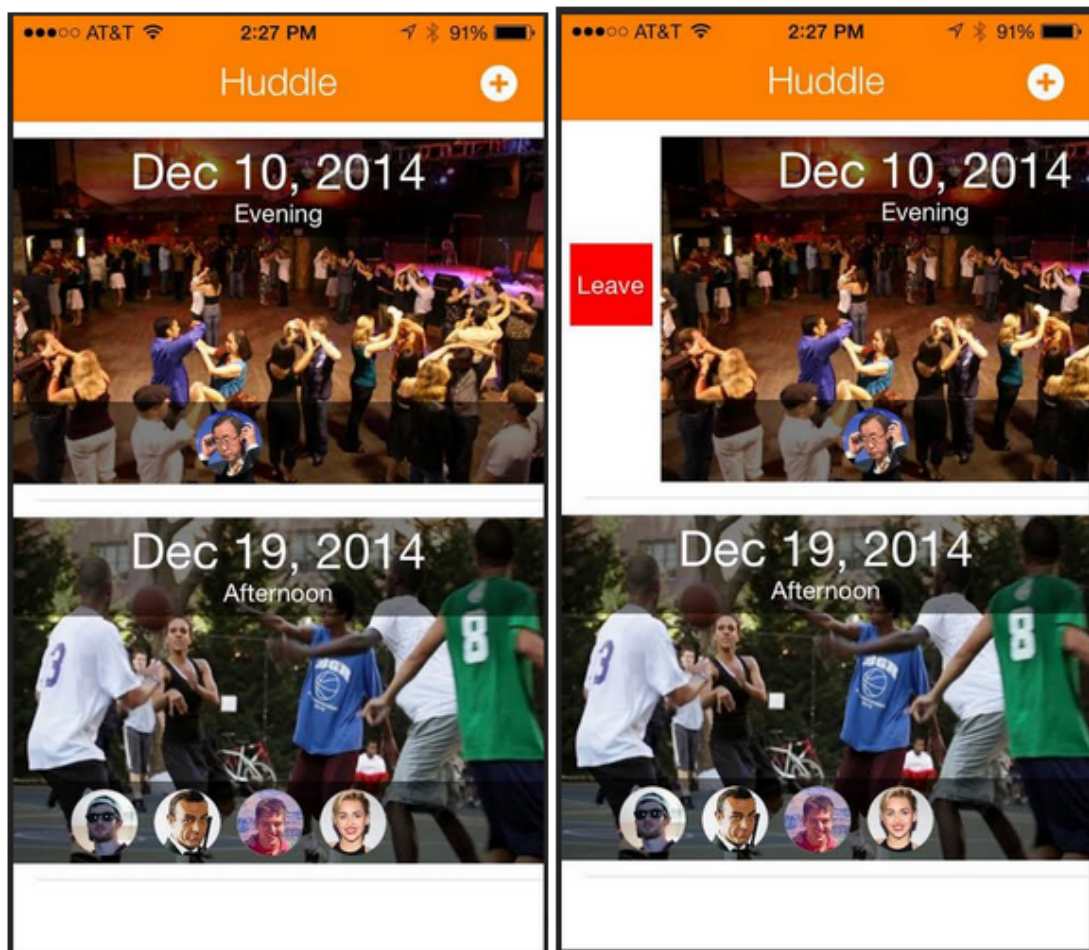


Figure 10. High-fi Home Screen With Hidden Leave Button



Figure 11. High-fi Calendar Screen with Selectable Times.

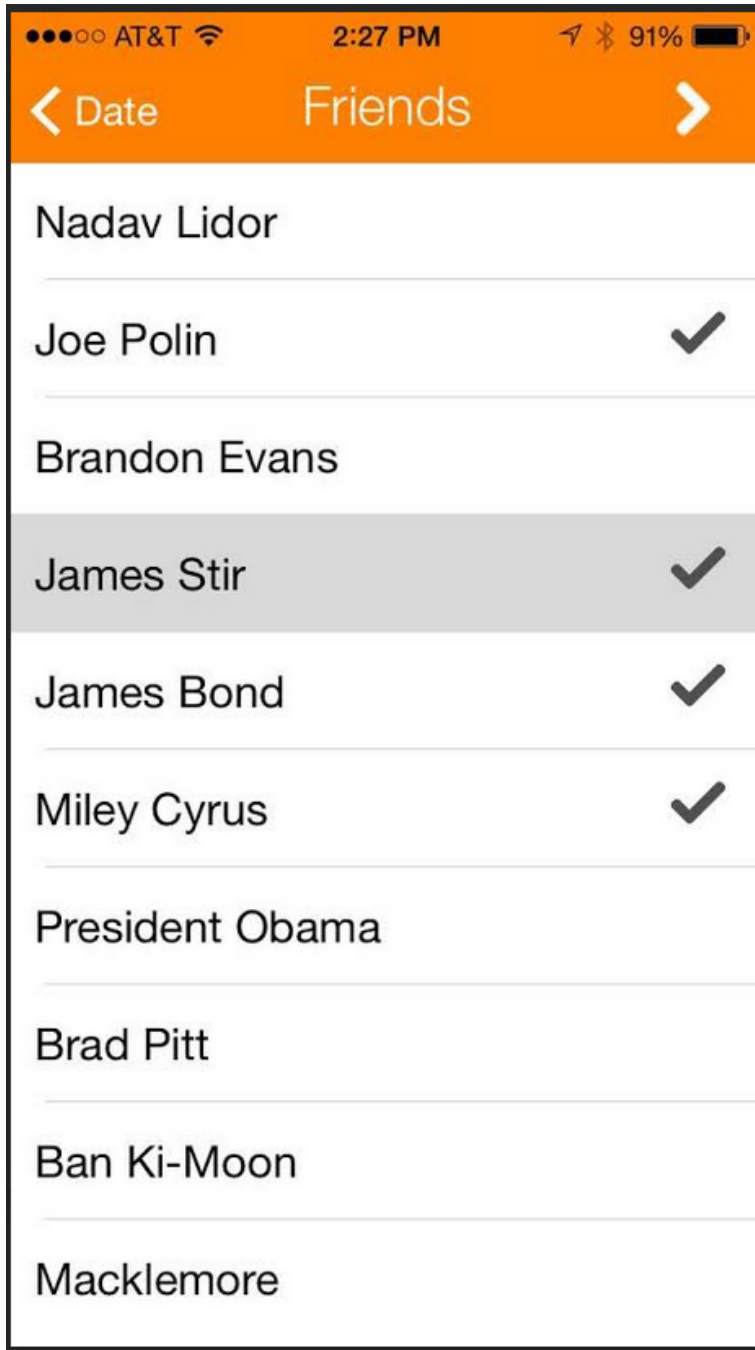


Figure 12. High-fi Friend Invite Screen.

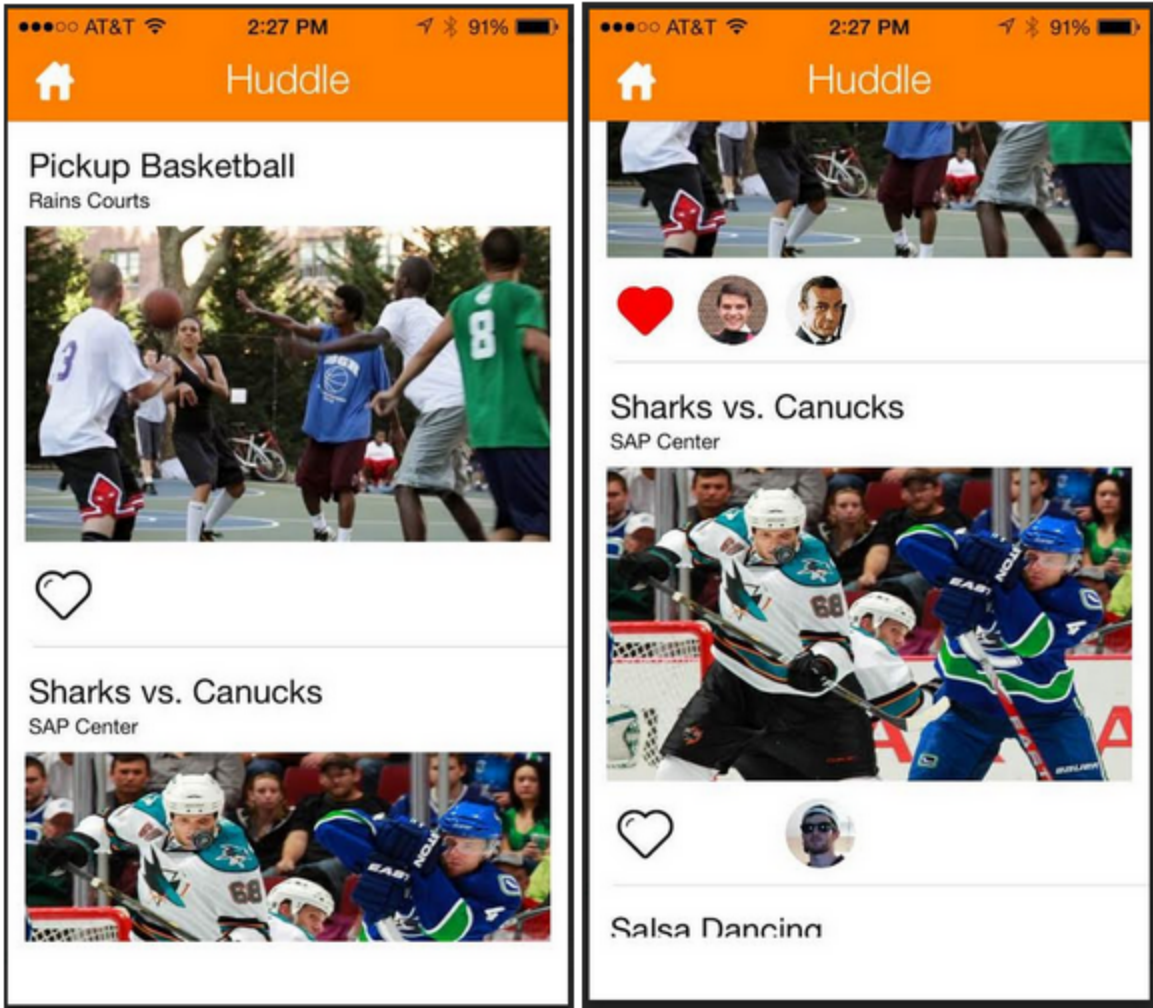


Figure 13. High-fi Activities Screen, With (Right) and Without (Left) Votes

Prototype Implementation

Tools

We coded the app in objective-c, so it runs natively on iOS. We used the xCode IDE for coding the app, and iPhone simulator was our main tool for testing. On the backend we used a SQL database to store all of the huddles, votes, dates, and other information as locally stored data on the iPhone.

To coordinate as a team, we used Github, which worked well for our source control needs. We used TestFlight to create builds of the app for team members to test on. TestFlight allowed for quick testing on multiple devices without requiring group members to build and deploy the code themselves.

Things that Did Not Go Well

Using a custom calendar in our app was challenging. We used an open-source calendar called JTCalendar that satisfied our requirements for the calendar page. However, we had trouble figuring out how to indicate to users that the calendar could be swiped to move to different months.

Objective-c has a steep learning curve that made it difficult to bring people who have not coded in it before up to speed.

Testing a SQL database on iOS was difficult because there's no easy way to dump table contents or to debug SQL problems. SQL databases are stored internally on iOS, which we do not have access to.

Code Design (Things that Worked Well)

1. Strong use of tableviews with custom UITableViewCells to create most of the screens.
2. Used some open source libraries (JTCalendar) to provide some extra functionality without having to code it ourselves.
3. Heavy reliance on SQL database to 'save' all work done (after any changes) to make sure that no work is lost.
4. Use of screen load (viewDidLoad / viewWillAppear) to create customization of screen based off of SQL database.

5. Use of global variables / settings to control things like color / font / table size (to allow for rapid iteration).
6. Separate view controllers for each page (to allow group members to work in parallel).

Wizard of Oz & Hard-coded Data

1. Event suggestions was hard-coded with a fixed number of events.
2. The list of friends to invite was also hard-coded.
3. We used a timer to Wizard of Oz the voting process. Votes from other friends appear after a certain time has elapsed rather than actually implementing this social feature.

What's Currently Missing

1. Integration with Google calendar for scheduling.
2. Integration with Facebook / Google for event curation.
3. Custom event creation.
4. Actual coordination with friends.
5. Login system.
6. Friends list organization with "recent" and "best" friends sections.
7. Event details screen.

Project Stats

Approximately 3000 lines of code were created, not including any open source libraries used.

Roughly 40 hours were spent coding the app, split across the team.