

# PickUp

## *Interactive Prototype*

[http://www.cs.washington.edu/education/courses/cse440/CurrentQtr/projects/pickup/docs/pickup\\_interactive\\_prototype.pdf](http://www.cs.washington.edu/education/courses/cse440/CurrentQtr/projects/pickup/docs/pickup_interactive_prototype.pdf)

CSE 440, Fall 2009

### Team Members

Ian Crofoot (Design Coordinator)  
Sunil Garg (Documentation Coordinator)  
Mitch Ishimitsu (Testing Coordinator)  
Daniel Swisher (Group Manager)

### Problem and Solution

As people plan casual sporting activities, one of their main tasks is to locate other people interested in participating at a mutually convenient time. At times, this becomes quite a challenge, since people generally depend on their existing social networks, which may or may not contain the people they're looking for. Additionally, coordinating such games with large groups of people is not a simple task. Our application seeks to improve upon the status quo by making it easier to find people with shared sporting interests and compatible scheduling and location preferences.

### Tasks

Our tasks provide coverage of the main functions of our application, as follows.

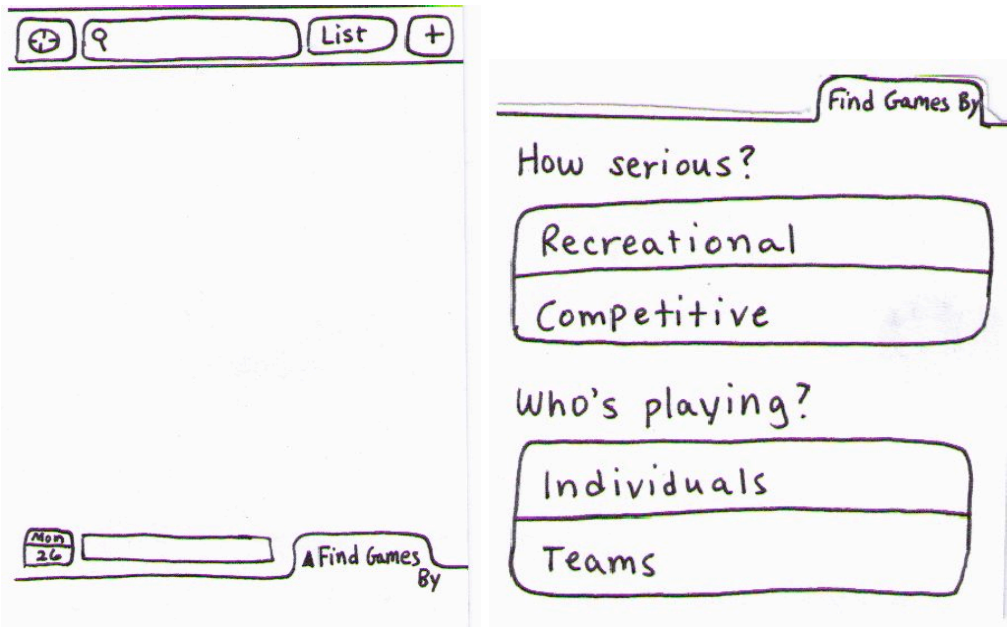
- **Easy task:** Our simplest task involves leaving feedback for another user after a game is done. This does not have an analogue in current sports practices, other than impressions formed of people via word of mouth.
- **Medium task:** Another task is to find a game of basketball and join it. This is a relatively simple task, requiring the user to locate a basketball game on the map, view details, and join it.
- **Hard task:** Our third task gives an alternate scenario to the one above: none of the basketball games look appealing to the user, so they must create their own.
- **Hardest task:** The most complex task we present is to first create a volleyball team and invite 3 friends to join that team. Once that is completed, we ask the user to find another volleyball team and challenge them to a game. This involves the user filling out their own scheduling info, sending out it out to other users (simulated), receiving responses about what times are ideal for others (simulated), and choosing the best time.

## Interface Revision Sketches

### Search and Filter options for finding a game

Before, we had two separate elements on the game screen for finding a game; one, a search bar, that allows users to type in what sport or other characteristic of the event that they want to find, and two, a filter tab with options for choosing between competitive and recreational games, and between games where individuals join in and where entire teams, or groups, join in. Having the two separate elements led to some confusion for users trying to find a game. In trying to figure out a way to combine the two elements, we simply decided to drop the filtering options, as there is no longer a limit on whether teams or individuals can sign up for a game (though games can be private – invite only), and decided that the simplicity of dropping the competitive/recreational distinction outweighed the strangeness of there being only one filtering option, as well as the fact that game descriptions themselves can contain info on intended seriousness or skill level.

Old Screens:



New Screen:



## Scheduling

Scheduling a game, and assisting a game creator in choosing a time, proved to be one of the most complex tasks. Before, we had the creator set a range of dates, and other participants filled out what times they are available across that entire range. We realized though, that at some point the creator will also have to enter in their availability info. So, to simplify things, we now have the creator enter in when they are available, to both establish the range and to constrain the total available time for other users to enter when they're available. We felt this constraint was worth the benefit of simplifying the setting of the range and the setting of availability info.

Old Screen:

Cancel New Game Done

Choose your Sport >

+ Invite player >

Privacy  
Public

Request schedules between:

Tue Oct 27, 2009 3:30 PM  
Fri Oct 30, 2009 5:00 PM

Sat Oct 24	1	00	
Sun Oct 25	2	15	AM
Mon Oct 26	3	30	PM
Tue Oct 27	4	45	
Wed Oct 28	5		

New Screen:

Change Location

Time

Set Time Now Or Get Schedules From Players First

◀	Thurs	Fri	Sat	Sun	▶	▲
	█					12 PM
	█			█		12 PM
			█	█		1:00 PM
			█			1:30 PM
						▼

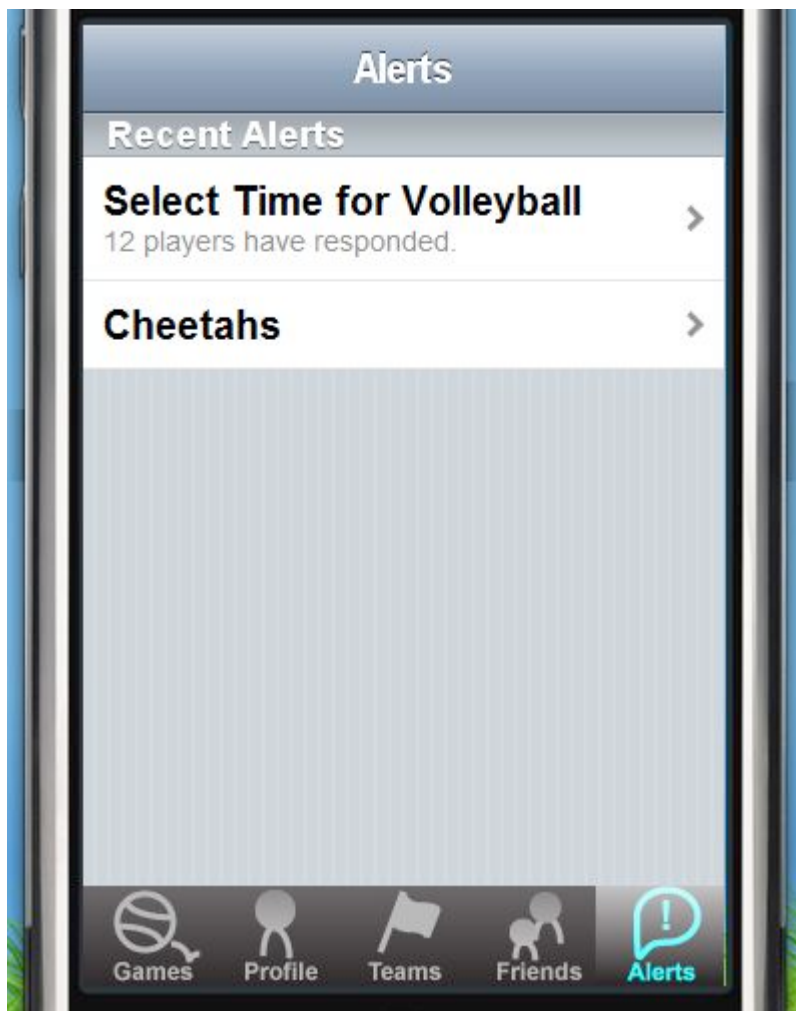
Games Profile Teams Friends Alerts

## Notifications Screen

Prior to our interactive prototype, we did not include a notifications screen. We had the program display notifications on startup if there were any new ones, but we decided that a user should have a place where they can refer back to them and view prior notifications. Thus, we added a notifications screen, accessible as a tab next to the other main tabs (profile, friends, games, teams). The simplest and most convenient way to display them, we figured, was a simple list, sorted by date, with the most recent notifications at the top, similar to text message or email systems.

Old Screen: (none)

New Screen:

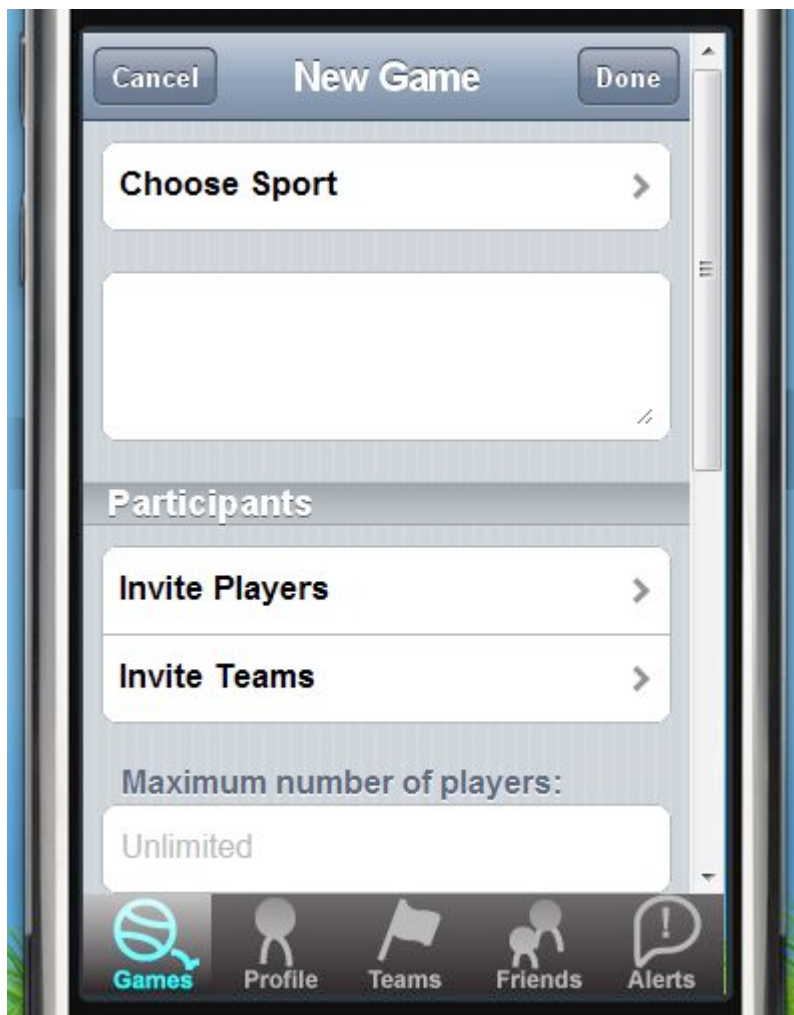


## Create a game screen

We also realized that several fields were missing from our game creation screens. We added items for entering a game description, setting the maximum number of players, and choosing a location. Other things we changed were 1) letting the creator invite individuals or teams, 2) making the wording clearer on invite-only vs. open games, and 3) letting the user choose between setting the time then or requesting schedule info from other players.

Old Screen: Refer to 'Scheduling', above

New Screen: Refer to 'Scheduling' above for view of requesting schedules portion; more below:





## Prototype Overview

### a) Overview of Implementation

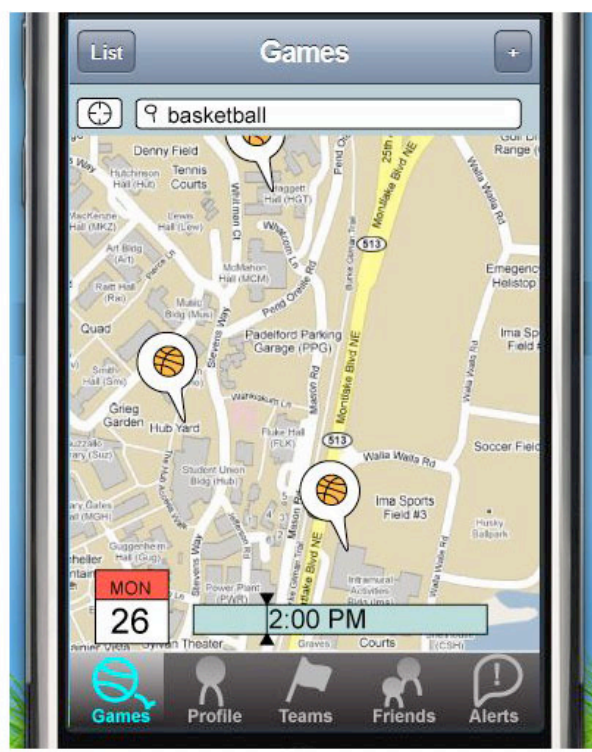
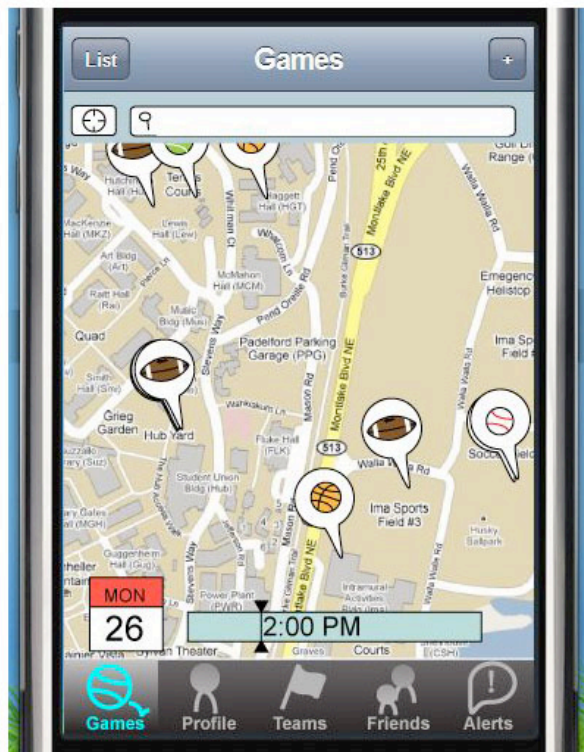
Our interface is designed with many non-standard graphical controls, like maps and time sliders, which made our interactive prototype a challenge. We decided that using images for graphical controls and sticking with a fixed path approach would be the only feasible implementation strategy. We were able to find a tool to help us make our prototype look and feel like an actual iPhone, however, this tool has limitations which forced us to make some parts of the interface, like the time selection page, out of screenshots captured from the iPhone. We did not encounter a need to use Wizard of Oz techniques, which we believe is good, because our application is able to run on the internet without needing us to facilitate in person. Our prototype demonstrates how our set of tasks can be completed, often in more than one way, but it does very little dynamically.

## b) Scenarios (with storyboards)

### Scenario 1a: Finding a basketball game

Step 1: Tap search bar, type basketball, hit enter.

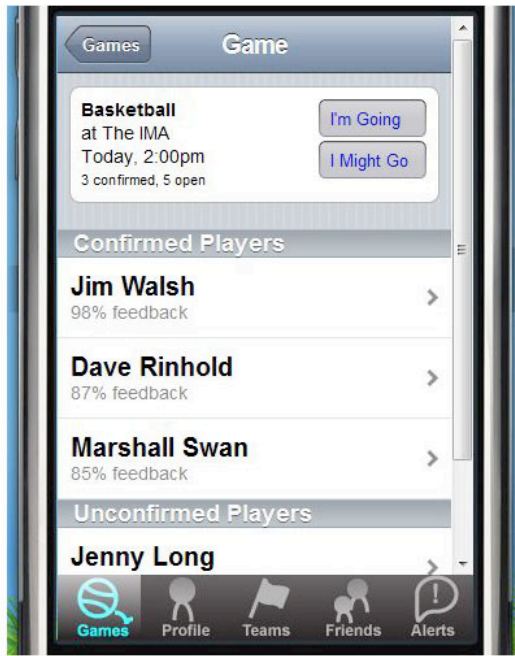
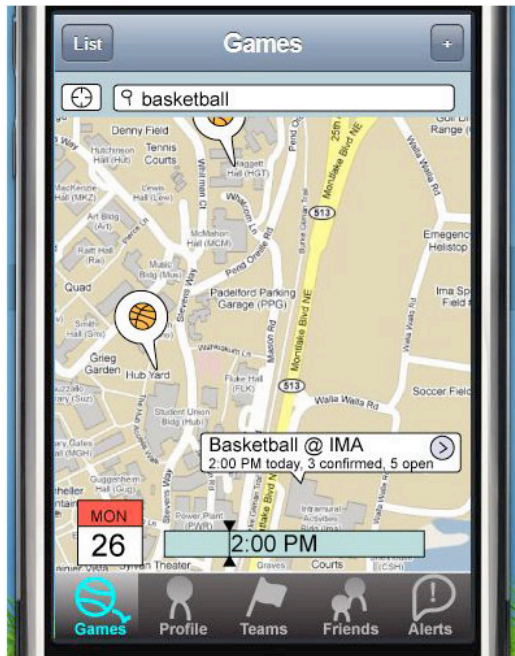
Step 2: Click icon at IMA to view summary





Step 3: Click arrow to view game details screen

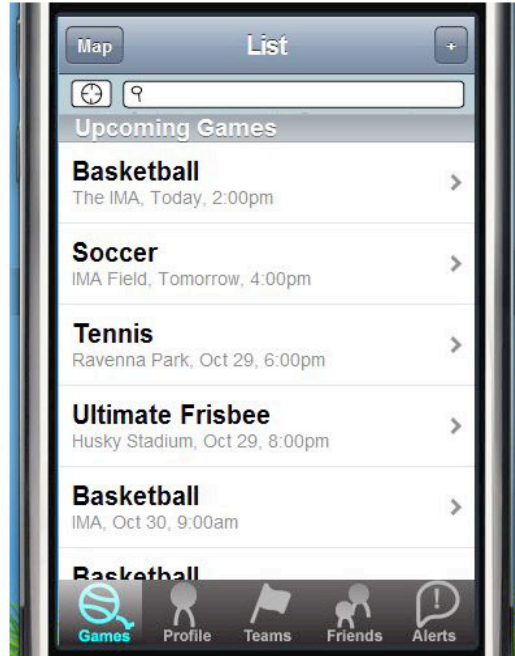
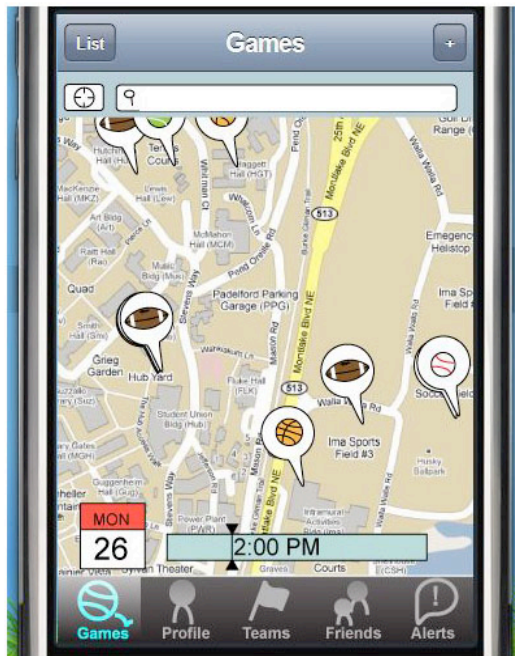
Step 4: Hit 'I'm Going'



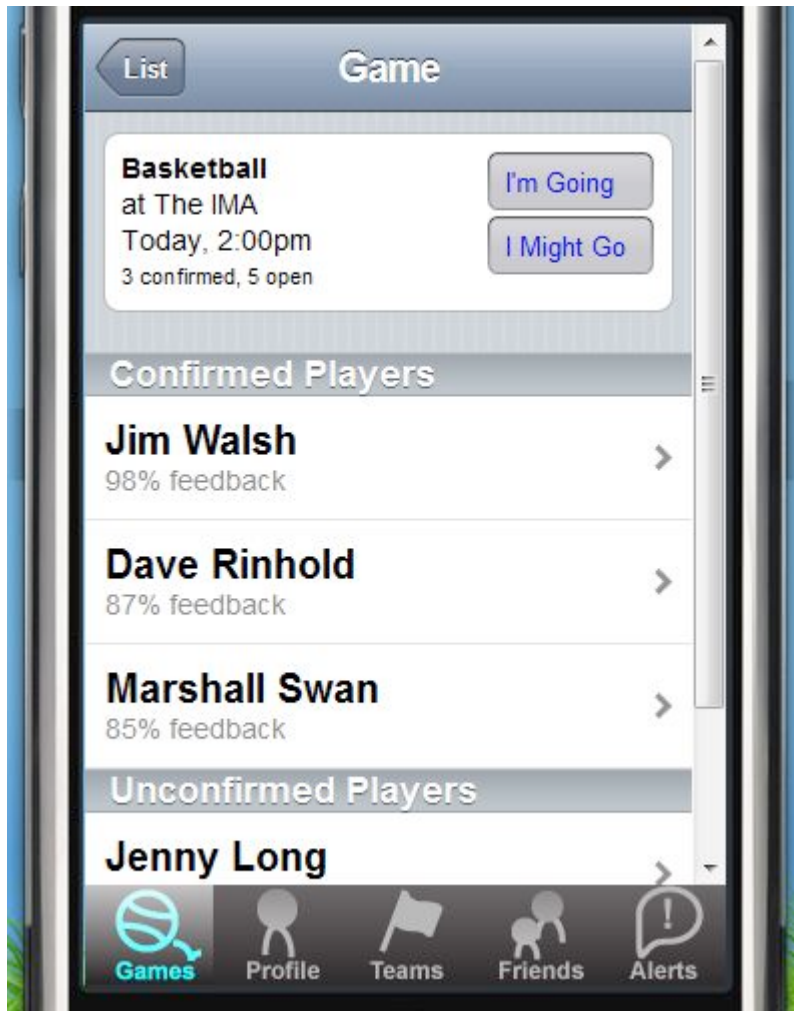
Scenario 1b: Finding a basketball game (alternate route)

Step 1: Click 'List'

Step 2: Click first game in list



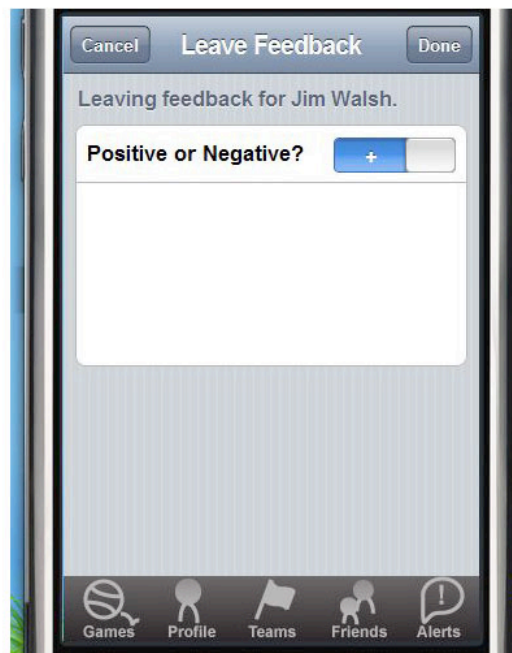
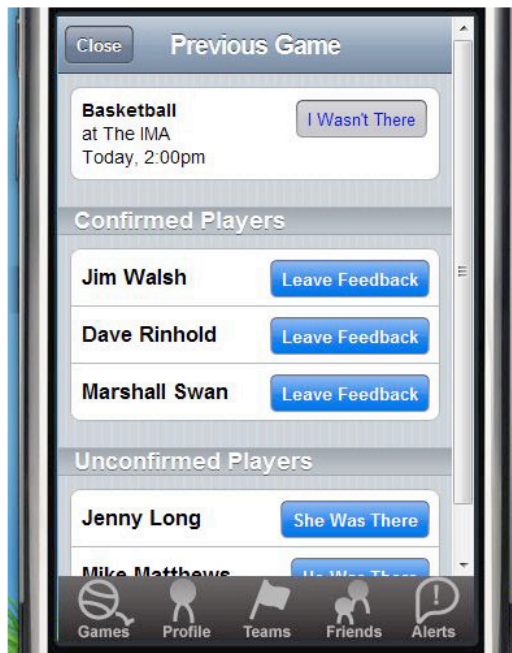
Step 3: Click 'I'm Going'



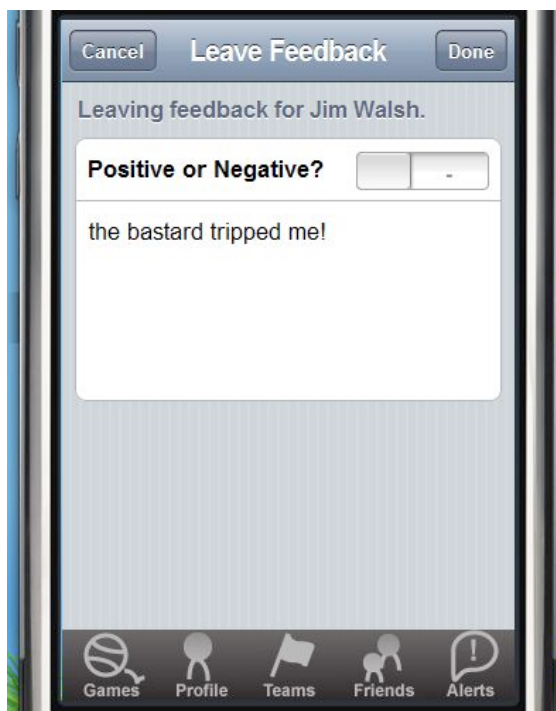
## Scenario 2: Leaving Feedback

Step 1: Previous game screen comes up on app startup, click 'Leave Feedback' for Jim Walsh

Step 2: Feedback screen

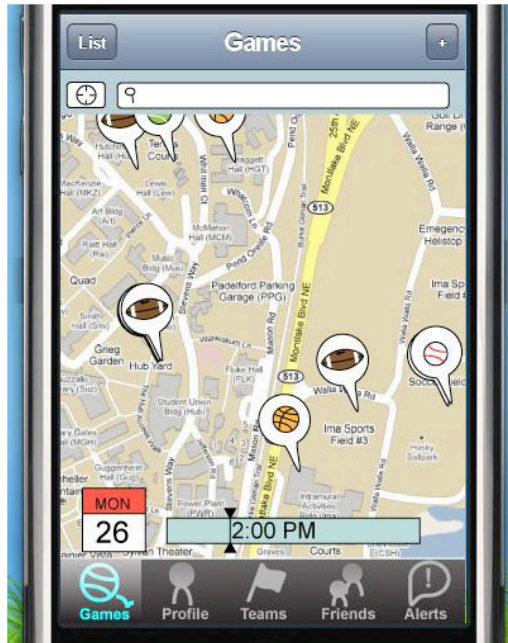


Step 3: Switch to negative (if desired), leave feedback, click 'Done'

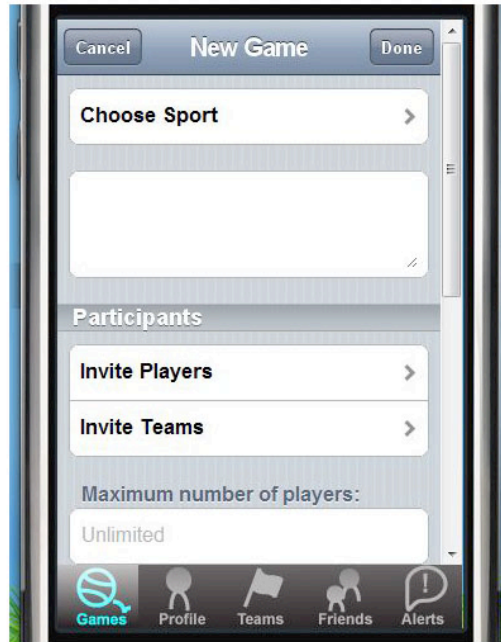


### Scenario 3: Creating a game

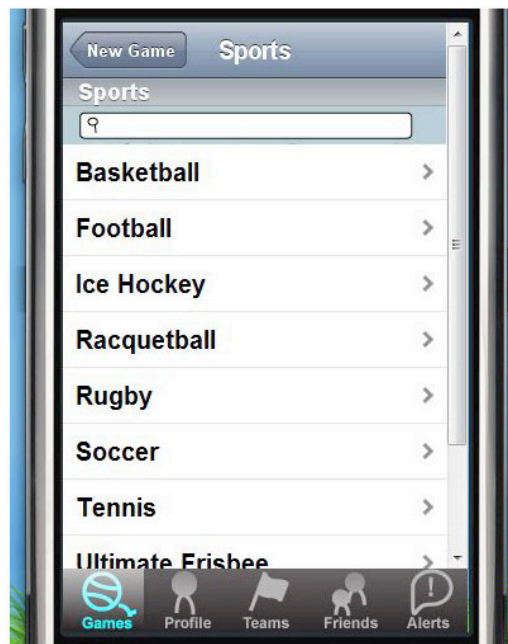
Step 1: Click '+' to create a new game



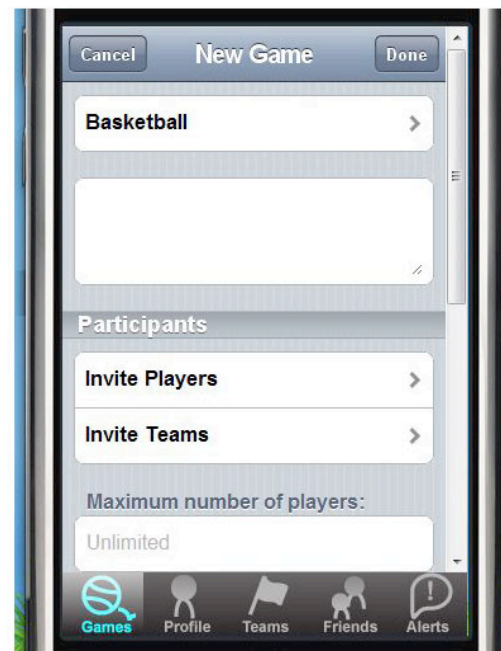
Step 2: Click 'Choose Sport'



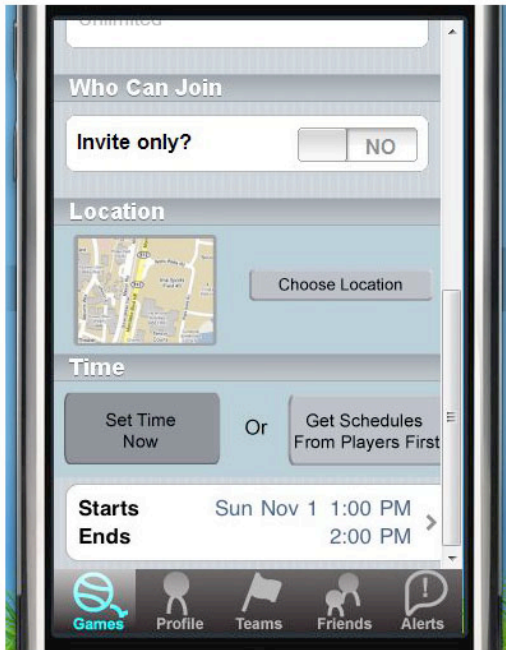
Step 3: Click 'Basketball'



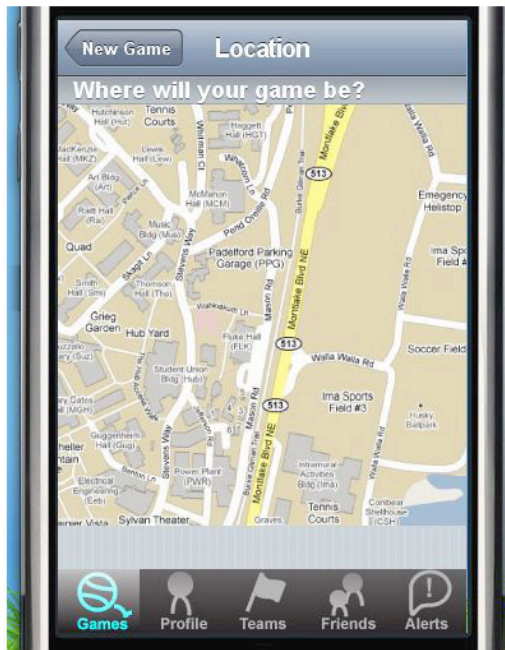
Step 4: Scroll down...



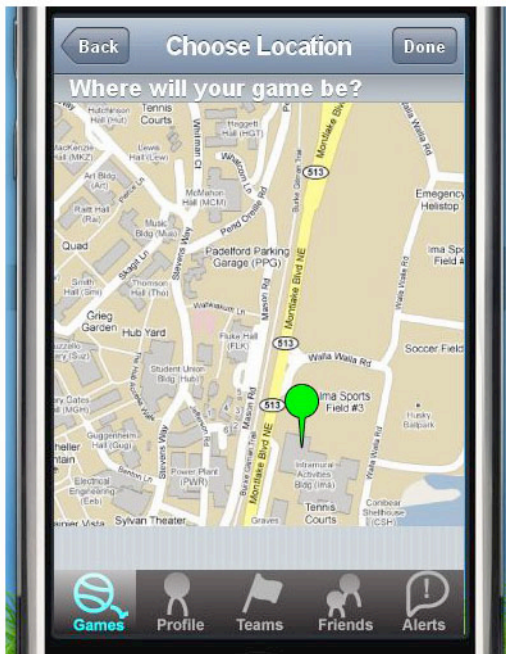
Step 5: Click 'Choose Location'



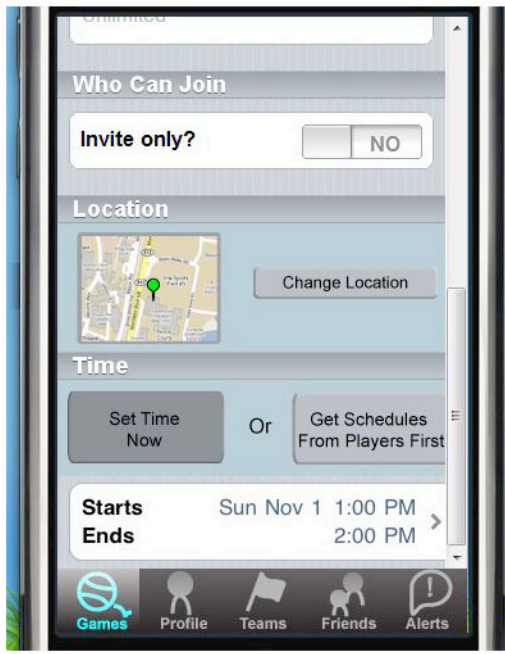
Step 6: Click on the IMA



Step 7: Click 'Done'

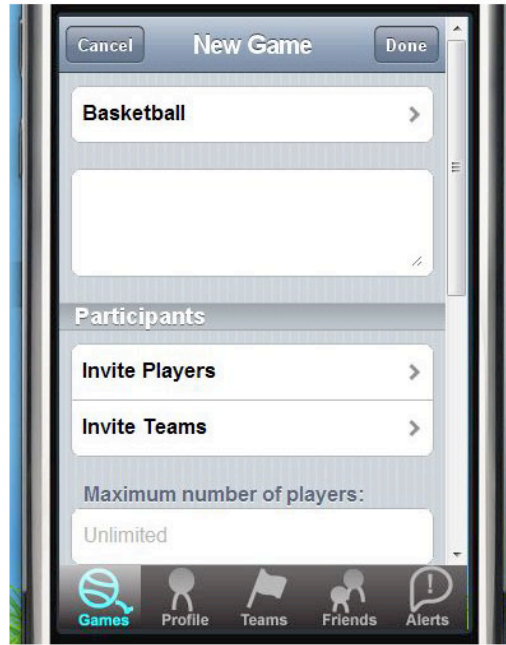
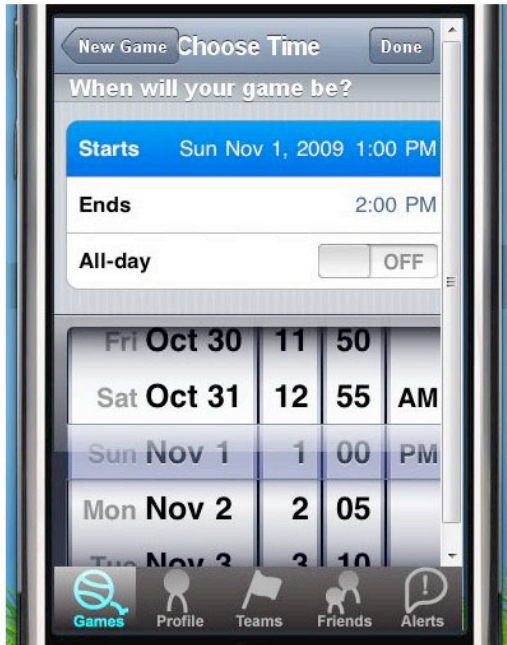


Step 8: Click Start and End times



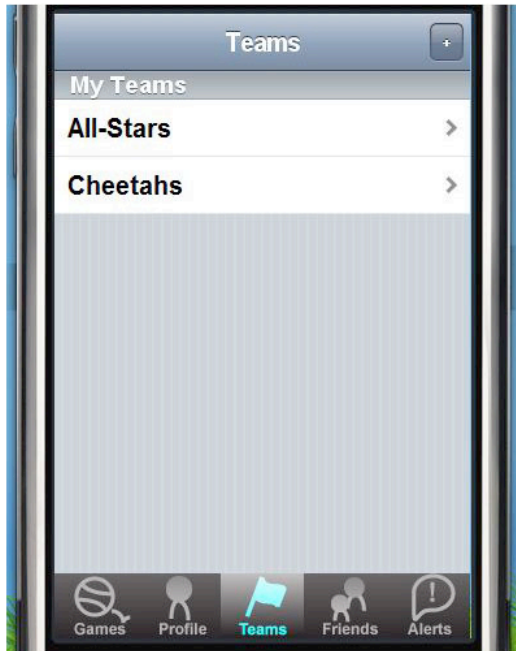
Step 9: Click 'Done' (we don't simulate this screen as it's a standard iphone widget)

Step 10: Scroll up and click 'Done'

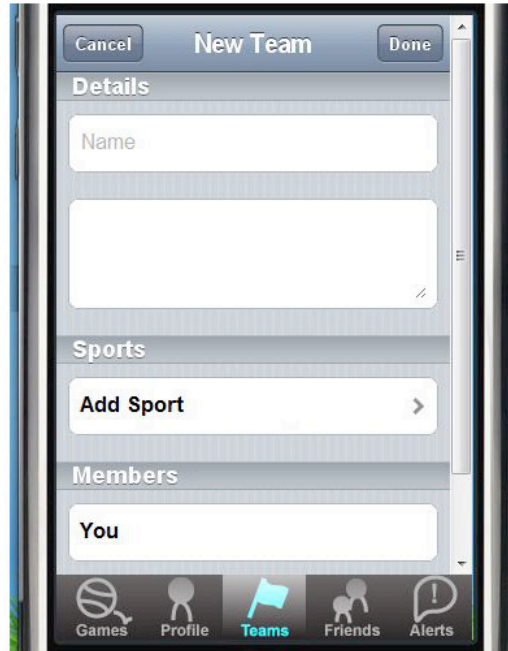


## Scenario 4, part 1: Creating a team

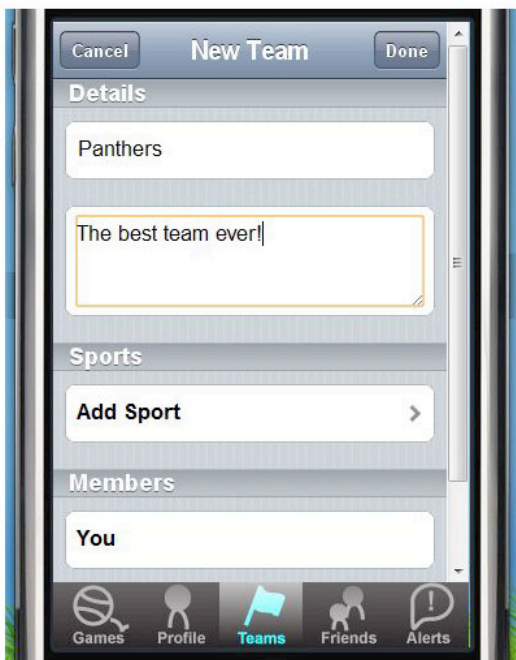
Step 1: Click '+' for new team



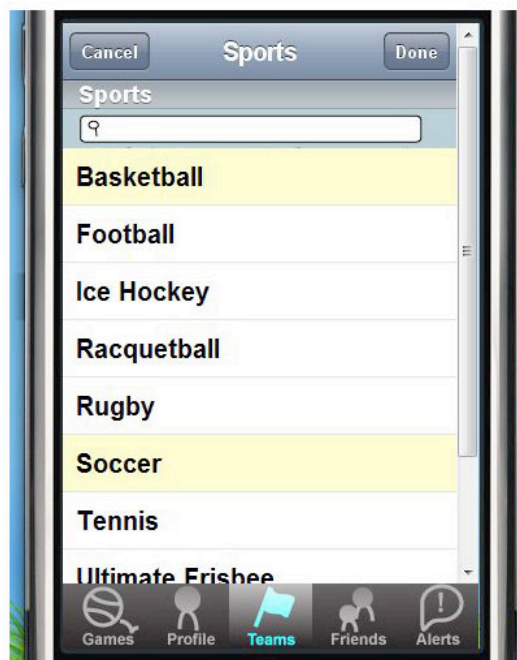
Step 2: Enter name and description



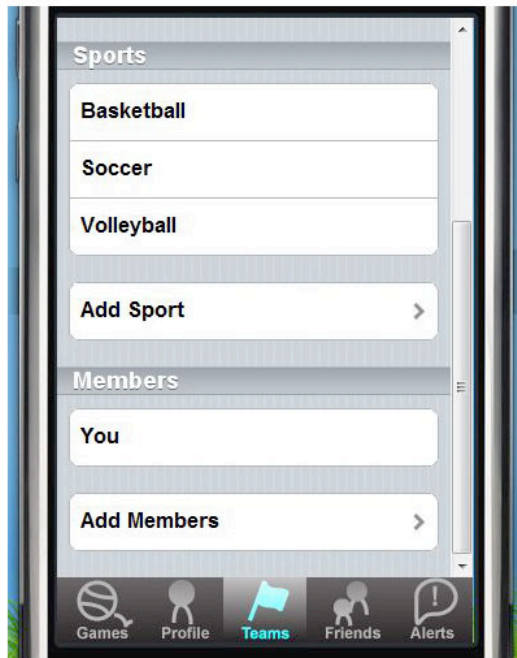
Step 3: Click 'Add Sport'



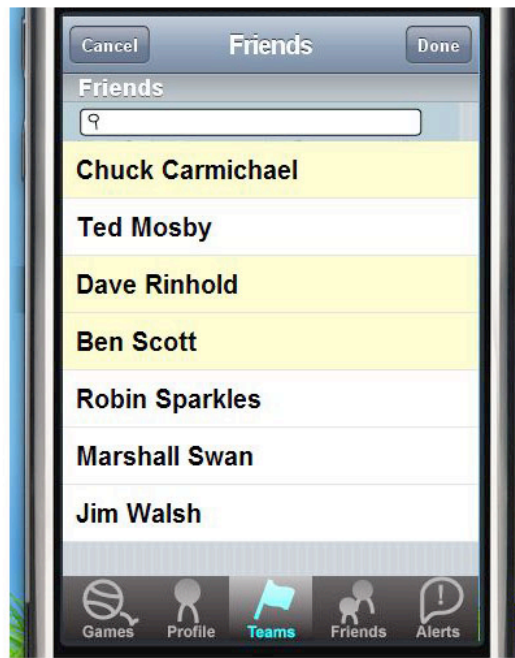
Step 4: Choose sports, click 'Done'



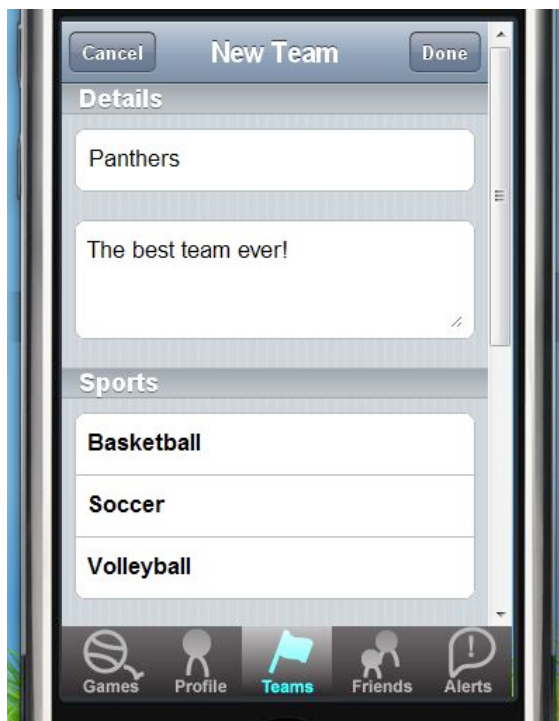
Step 5: Click 'Add Members'



Step 6: Choose 3 friends, click 'Done'



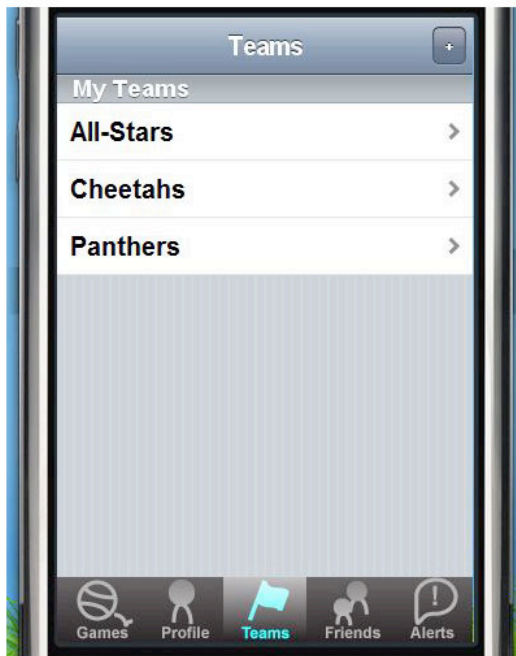
Step 7: Click 'Done'



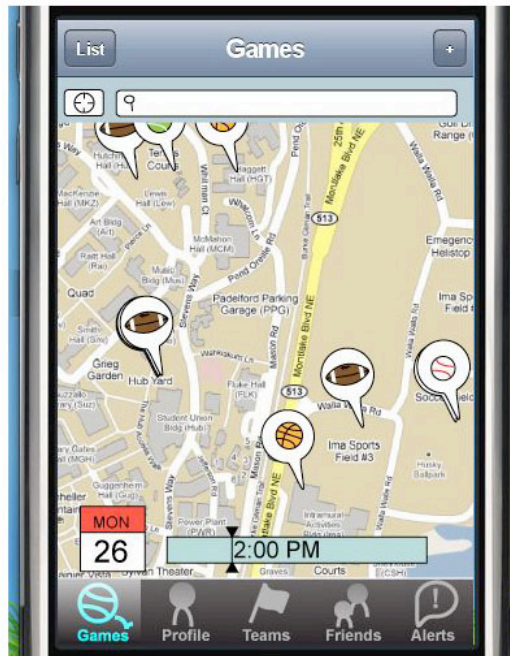


## Part 2: Creating game, requesting schedules

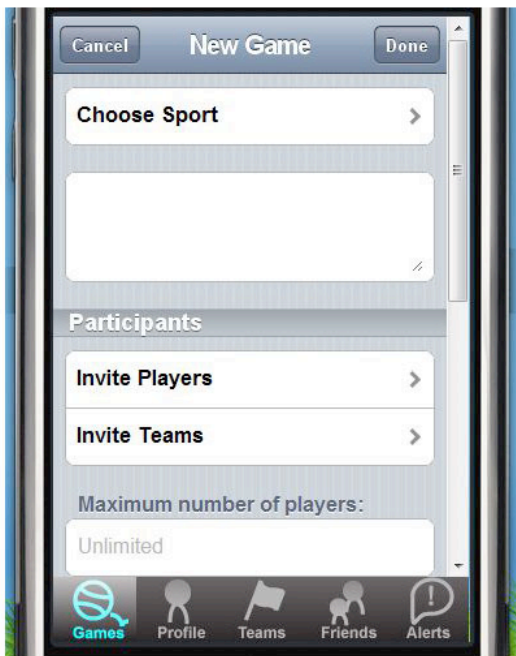
Step 9: Click 'Games' tab



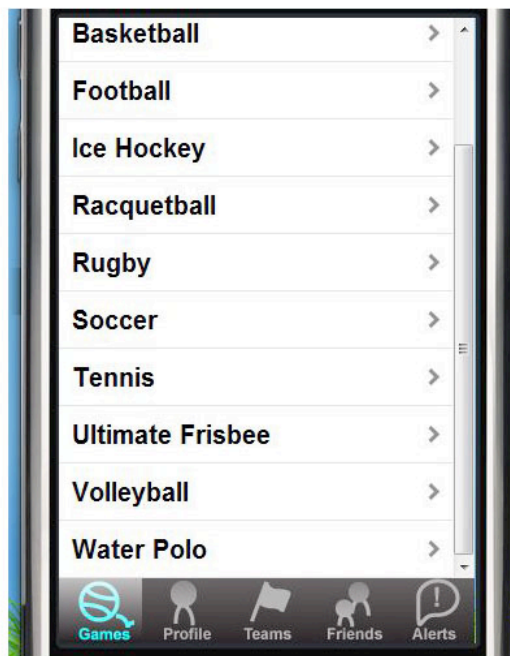
Step 10: Click '+' for new game



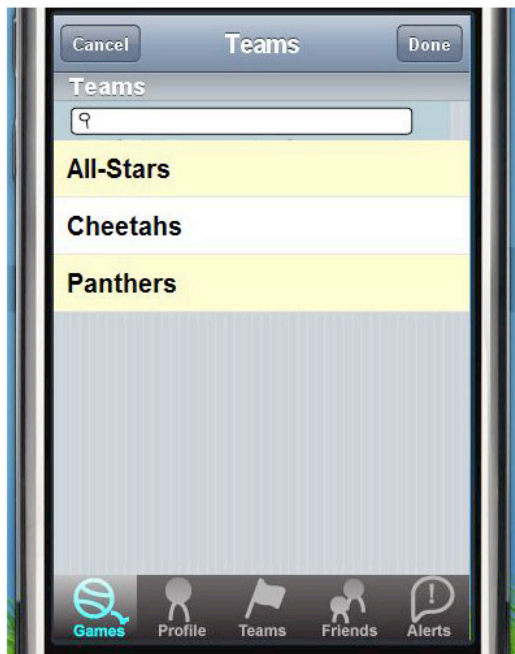
Step 11: Click 'Choose Sport'



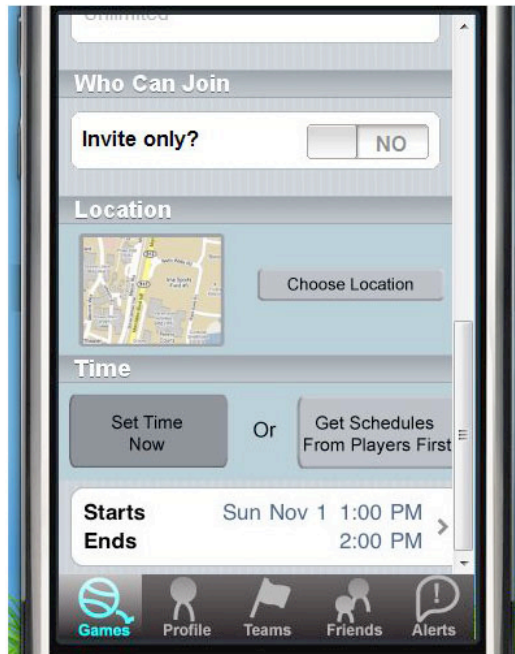
Step 12: Select 'Volleyball'



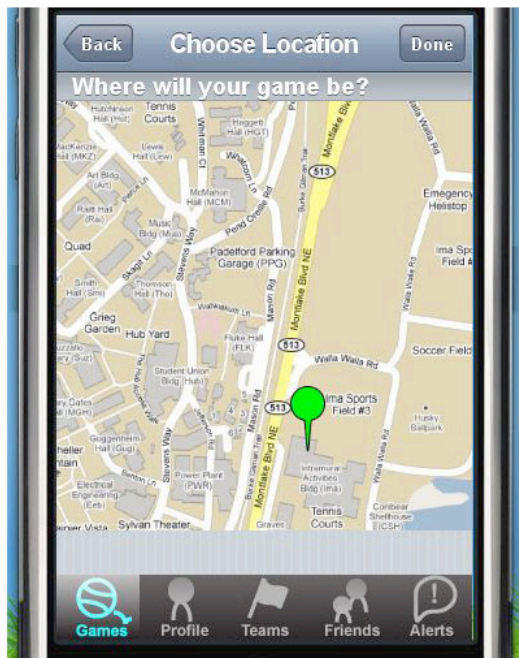
Step 13: Select 'Invite Teams' (from New Game screen), choose All-Stars and Panthers



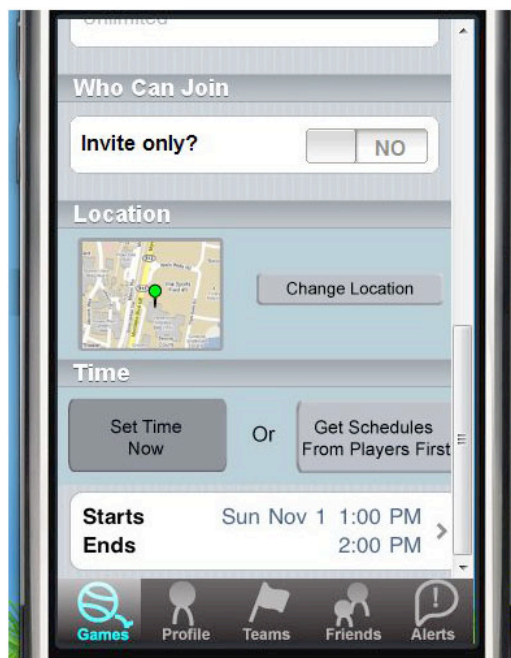
Step 14: Select 'Choose Location'



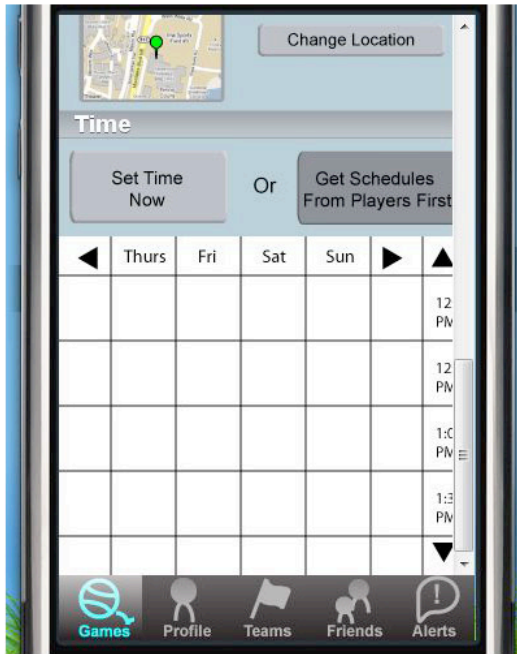
Step 15: Click on IMA, click 'Done'



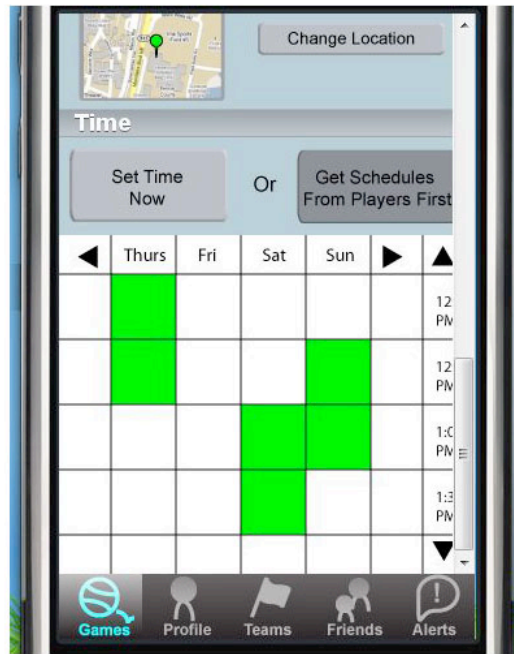
Step 16: Click 'Get Schedules From...'



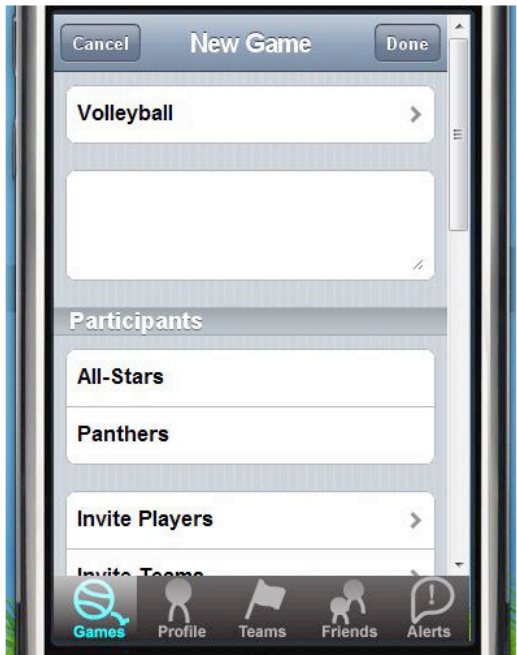
Step 17: Schedule pops up



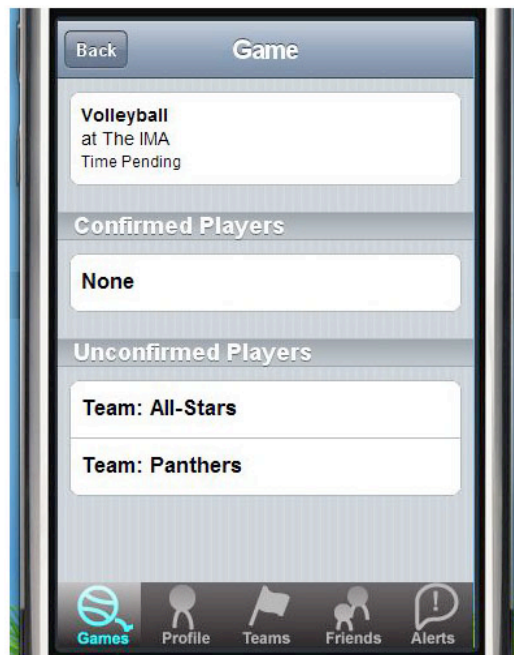
Step 18: Pick times you're available



Step 19: Click 'Done' screen

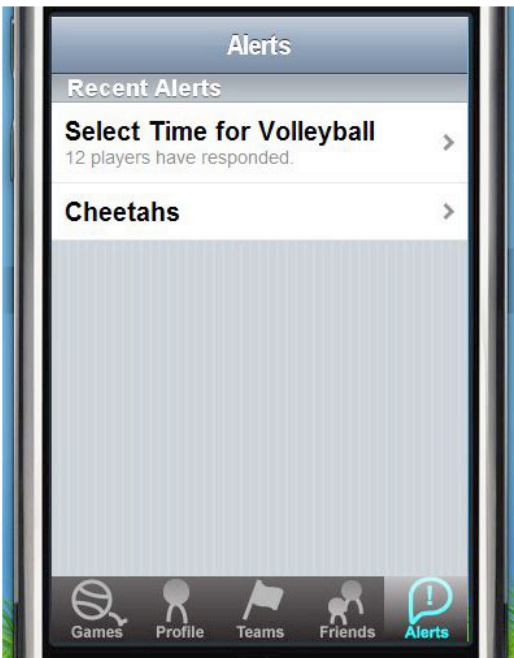


Step 20: Click 'Alerts' to go to that

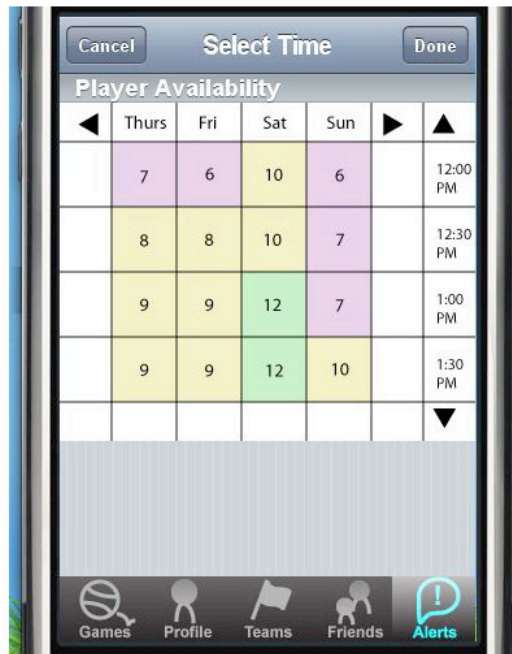


### Part 3: Choosing the best time

Step 21: Click the first alert



Step 22: Choose best time (green)



Step 23: Click 'Done'



### **c) Tools Used**

We used a Javascript framework called WebApp.Net (<http://webapp-net.com/>) to develop our prototype. It gave us the look and functionality of the standard iPhone interface, which is what we wanted to conform to. This saved us a lot of time that might have been spent manually recreating that style of interface. With many assets pre-made for us, and with an easy way to create lists and other interactive elements, this tool saved us a lot of time, and was perfect for developing our iPhone-style application.

One drawback to the system, however, is that a lot of the interface elements were limited. For example, we wanted to have the ability to toggle between "invite-only" and "anyone" in the game options, but the toggle widget only had enough room for "yes" and "no". This meant we had to reword the questions for that option, and was not our ideal solution.

Another drawback is that the prototype does not render entirely correctly in all web browsers. Things work well in every browser we have tested, except Internet Explorer. In IE, the prototype does not render at all. We have decided that the features of the WebApp.Net toolkit outweigh these compatibility issues, and have put a disclaimer on the website to alert IE users of the problem.

Aside from Javascript and the WebApp.Net framework, we used html and images to implement difficult controls.

### **d) What was left out and why**

Because this is a fixed-path prototype, it is only functional for the specific tasks and scripts that we have outlined. Thus, the user is not able to click all of the elements on any given page and see the expected action occur – in places where actions are not implemented, clicking an item will result in no change. This greatly simplified our development process, as we did not have to implement the logic to properly deal with what would happen when users deviate from the script.

In particular, one thing did not implement is the friends tab, because it is not directly involved in the tasks. Similarly, any other screens that were not required to successfully complete the given tasks were excluded. Also, any backend logic, such as databases to manage profiles, game information, and teams was excluded, due to the difficulty of implementation and the fact that it was not required for the fixed-path approach.

### **e) Wizard of Oz techniques**

Our prototype does not require any Wizard of Oz techniques for use.