

Player Tracker Final Report by Jared Alexander

*Note, Player Tracker was tested on Android 2.2, both on a device and emulator

I created a fantasy football player tracker application for users in Yahoo fantasy football leagues. It uses the Yahoo Fantasy Sports API to authorize a user, and download a list of players on one's team. It also takes advantage of the Twitter Search API, and requires the user to authorize oneself with a Twitter account login. I made a Twitter account necessary to use the app because it increases the number of search requests a user is allowed to make (150 per hour without logging in, 350 per hour when logged in). This should cut down on Twitter rate limits for those who make heavy use of the app.

There are many use cases for this app. Some people may just want to check tweets about their players on demand. For example, if they are monitoring if a player will start this week, many times tweets can paint a picture of the player's current situation. Other owners may want updates delivered to them automatically. For example, if they aren't able to watch the NFL game that their player is playing in, they might want to know the status of that player. Is he hurt? Has he scored a touchdown? By defining keywords (such as td, touchdown, injury, injure, broken, etc...), the app notifies users whether or not these words are occurring in tweets about their players. If they are, then the user is notified.

Getting Started

When users first run the app, they are required to log into both Yahoo and Twitter accounts. The app stores authorization tokens for both logins. Twitter's token does not expire. However, Yahoo's token expires after an hour. Users may be asked to reauthorize a Yahoo login if they decide to use Player Tracker for a different Yahoo fantasy team, or to refresh the roster for a given team. Users may wish to refresh a roster if they have used Yahoo's website to add/drop/trade players and the app has an out of date copy of their roster.

Player List Screen

After gaining authorizations, users select a league and team name within that league. They are then presented with a list of players on their team. By pressing a player name, users are presented with the most recent tweets mentioning this player. By checking the box next to the player name, users can enable background notifications for this player. If the notification calls for further action on a user's part (such as picking up the backup running back to replace an injured one), there is a button that will launch the official Yahoo Fantasy Football app to manage a user's roster. If it is not installed, then users will be directed to the Android Market listing for Yahoo's app.

Settings Screen

Notifications are managed in the settings screen. By pressing the settings button on the player list screen, the user is able to enable a notification background service, based on keyword matches. Users can add as many keywords as they like, and if a tweet occurs that has the player's name and any of the keywords, a notification will trigger, even if the app is closed. Keywords can also be removed. The settings screen lets users choose the update interval for the background service to trigger on. Additional settings include refreshing a team's roster, choosing a new team, or clearing Twitter authorization credentials.

Background Notifications

Notifications will appear at the top of the phone screen, even if the app is closed. By pressing on the player notification, the user is directed back to the app, and the most recent tweets are shown for that player. Tweets that contain one or more of the keywords are highlighted in red to stand out from the rest.

Problems Encountered

I was stuck for a while on getting OAuth to work for Yahoo. One of the problems was that the Java library I used (sign post) did not have any well-documented examples for authenticating against the Yahoo API. However, I was able to craft my own solution by piecing together parts from other sign post examples (such as Twitter). Other hurdles included learning how to interact with the Fantasy Football API, how to trigger a background service, and how to create and display notifications. These problems were solved by reading through the official documentation.

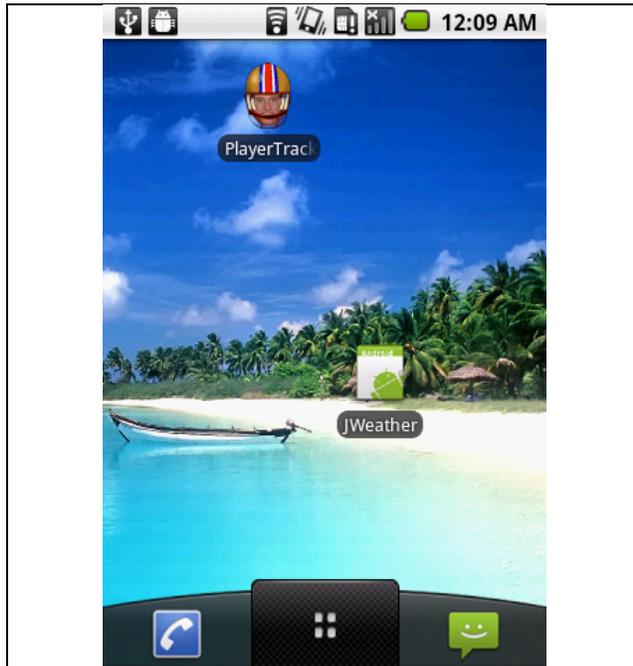
Reusable Classes

I developed a few reusable classes. HttpUtils is a utility class for connecting to the internet and downloading XML. TwitterOAuthActivity and YahooOAuthActivity can be used in the future for other apps that I make that need to authenticate users using Twitter and Yahoo. I can reuse my process for creating an Alarm Manager to trigger a background service at an interval to create and display notifications. This will definitely speed up app development in the future. Additionally, I created a few custom adapters for list views that can be reused in the future (such as a list view containing text and a checkbox).

Platform Specific Features

I used many Android specific features. Some of them included use of the Alarm Manager to schedule a background service to run, the use of a background service, creation and displaying of notifications, and the creation of an animation to display to the user.

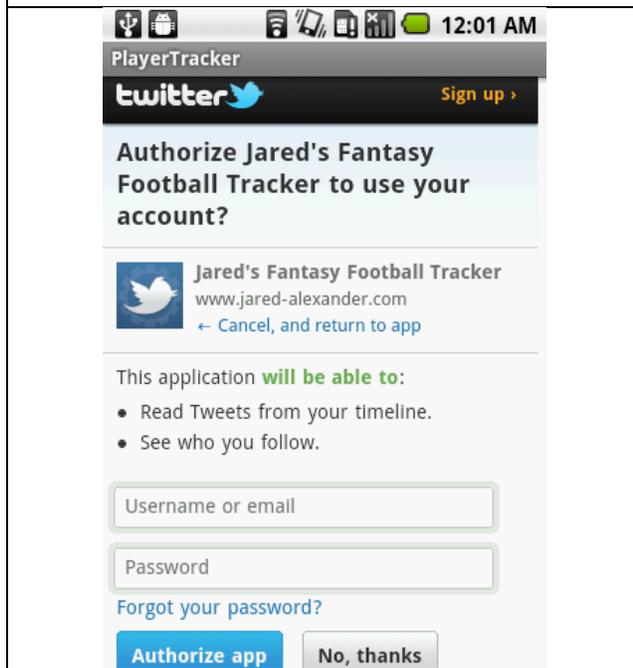
Screenshots



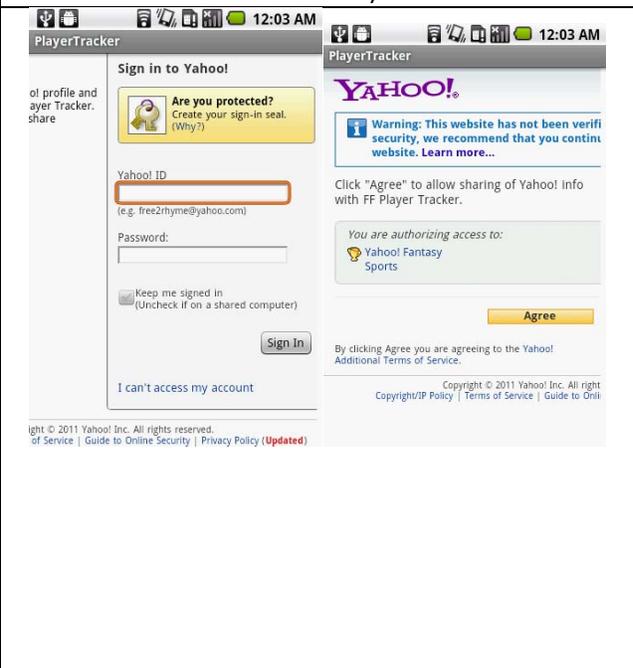
Icon on Android home screen (yeah that's me)



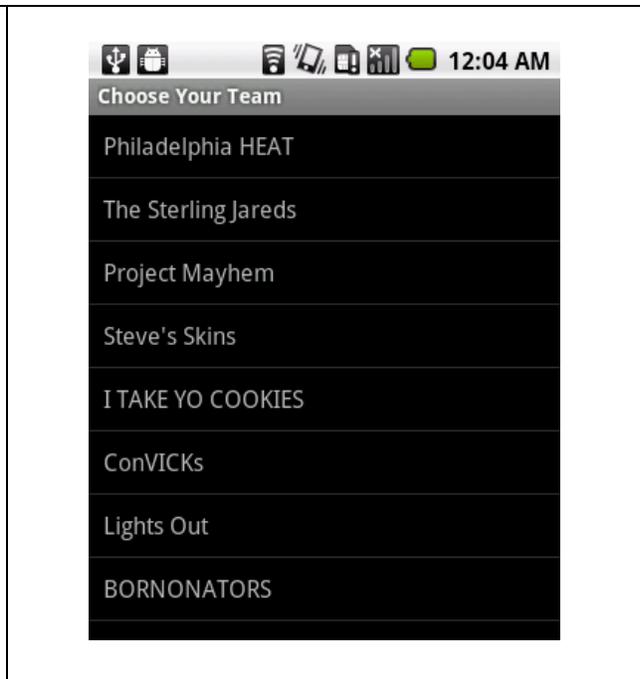
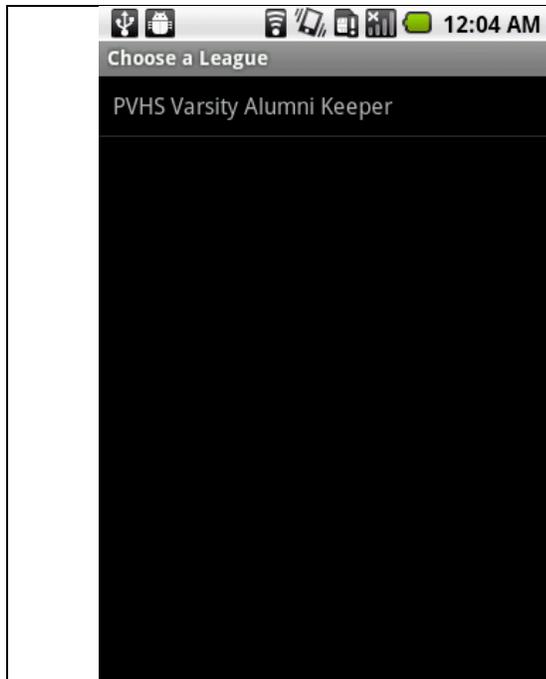
Splash screen animation when app loads (yeah that's me)



Twitter authorization

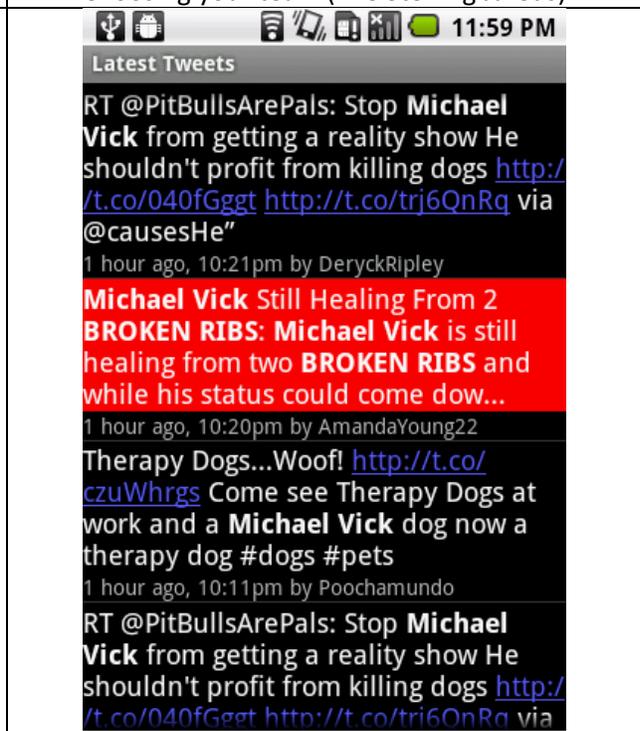
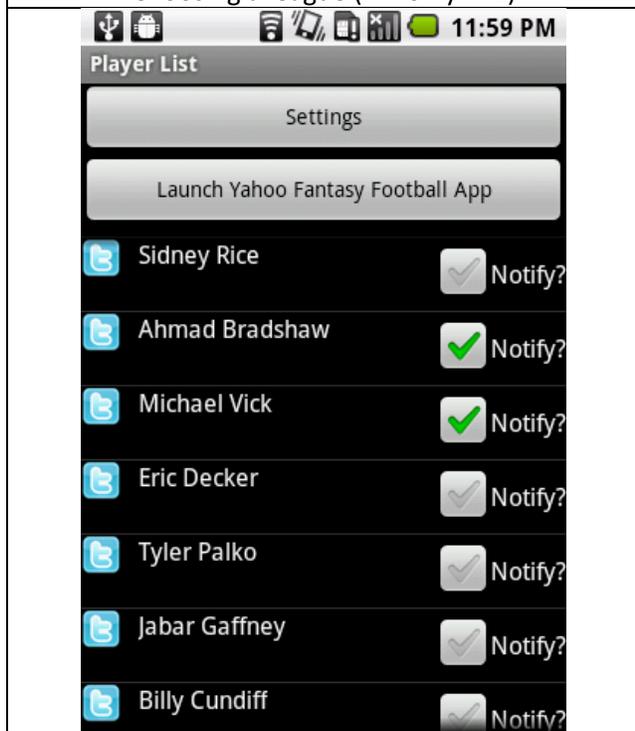


Yahoo authorization



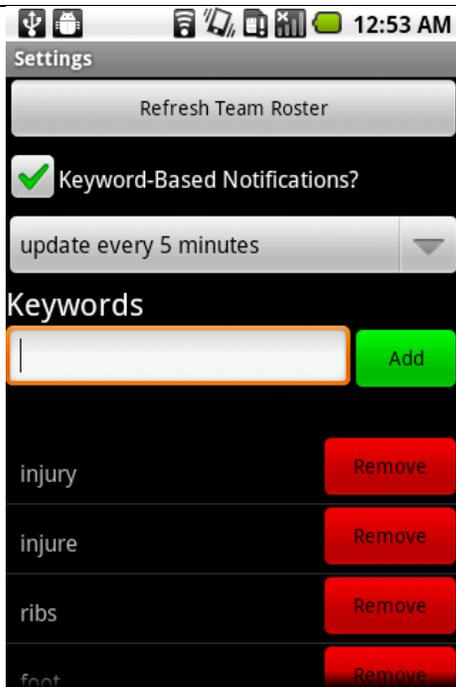
Choosing a league (I'm only in 1)

Choosing your team (The Sterling Jareds)

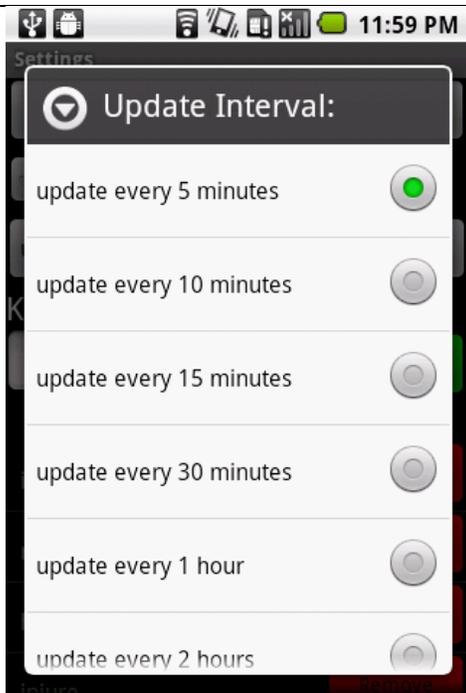


Player list

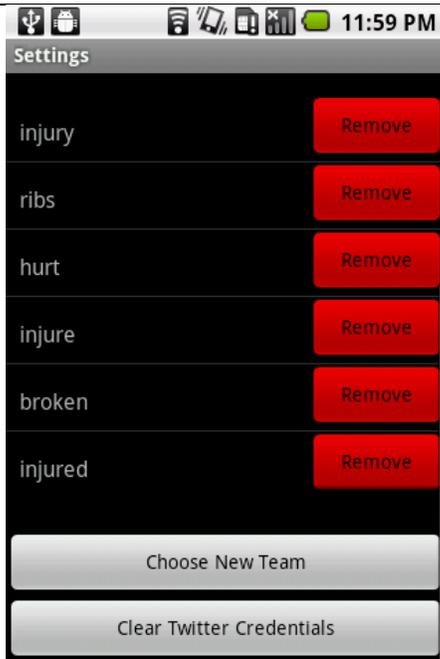
Example of on-demand tweet list



Settings screen



Adjusting the update interval for keyword-based notifications



Other settings at the bottom

<p>Example background notification</p>	<p>After tapping Ahmad Bradshaw, loading tweets</p>
<p>Tweet results, with red highlight of keyword that triggered notification</p>	