**Software Requirements and Design Document**

**For**

**Group 8**

Version 1.0

**Authors**:

Megan Achurra

Dylan Giesler

Daniel Lukish

Miguel Perez Ojito

Thomas Rudolph

1. **Overview**

Our project is a fantasy soccer website that is supposed to cater to newer players. It will have all the normal features expected from a fantasy sports site, such as drafts, trading, and of course a points system. Since we are attempting to also cater to newer players, we are adding in things like tool tips and suggestions that will make the system more accessible to those not familiar with fantasy sports or soccer.

Our drafts will allow the user and their friends to pick from the top players and assemble an all-star team based on informed decisions thanks to our player information system. Trades are also a very important part of any fantasy system, and ours will definitely have one. Knowing which data points are relevant and not is also important for a newcomer, and so we provide explanations for each along with how this can affect them. Additionally, while most fantasy sites use point systems that have all sorts of partial points, negative points, and blocked points, ours uses a simple point system. If your player scores a point, you gain a point. For every assist your player gets, they gain a point.

1. **Functional Requirements**
2. Create Leagues – High Priority

The ability to create a league and invite friends to it. Basically, each league will be a group of people competing with each other to see whose players can score more points. This is also how league owners are created.

1. Invite To League – High Priority

League owners should be able to invite any user to join their league.

1. Drafts – High Priority

League owners should be able to begin a draft when ready. This will begin a snake draft where the users are randomly assigned an order and given equal opportunity to choose players.

1. Point System – High Priority

The ability to gain points based on a user’s player’s performance. This is the primary metric by which players will compete.

1. Login – High Priority

Gives the ability to store information done on a user account. Also required for many of the functions listed above. This will be implemented with a database and user session to show a person is logged in between screens.

1. Sign Up – High Priority

Allows the users to create an account to start storing data. Gives access to the login functionality and all the functionality it provides.

1. Player information system - Medium Priority

Allows the user to quickly get a page full of stats and current news about a player so they can make better informed decisions about trades and picks.

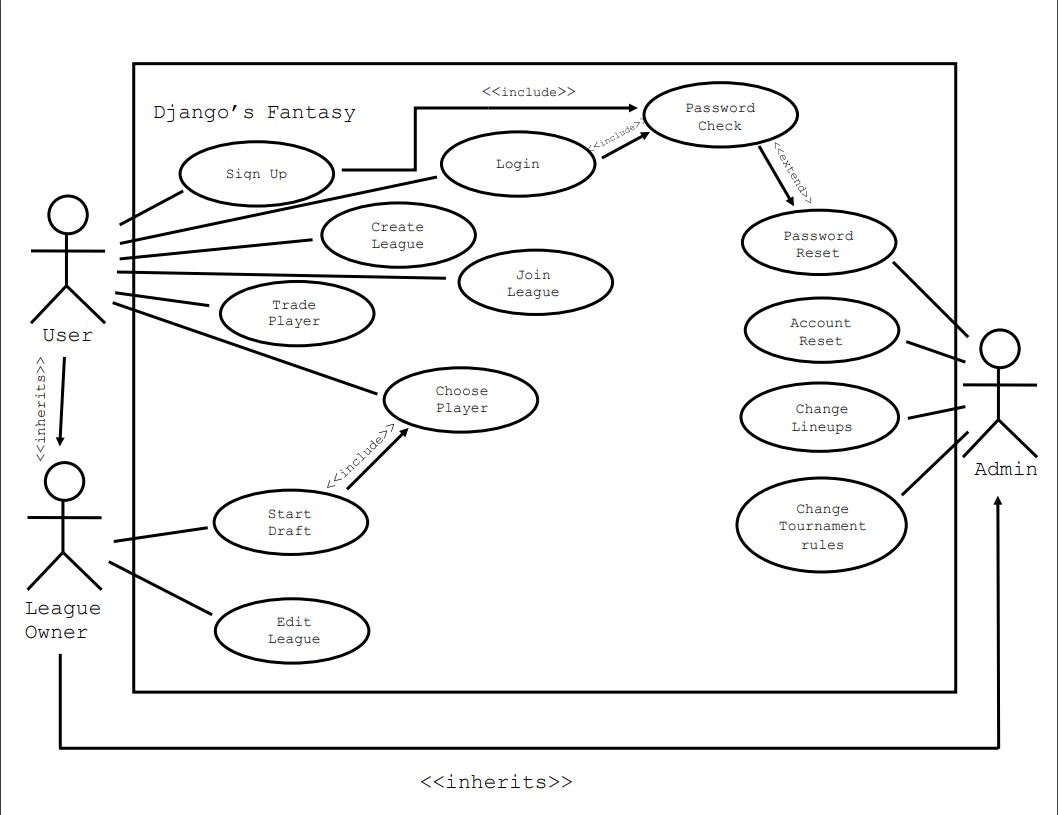
1. Trading – Medium Priority

Users should be able to trade their players with other users in their league.

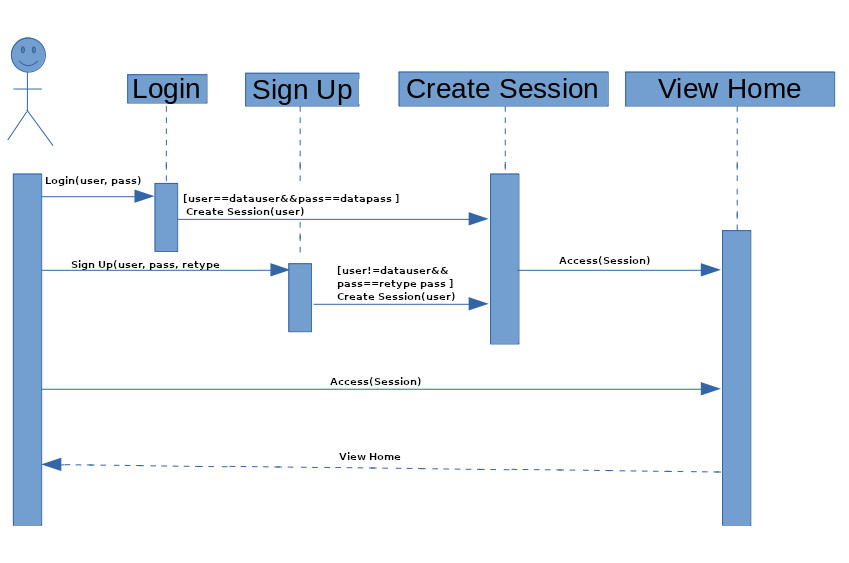
1. New Player Assistance – Low Priority

Users will be provided tools such as newsfeeds and annotated stats to make better decisions.

1. **Non-functional Requirements**
2. Hashing for Passwords – Many people, especially non I.T., will use the same passwords across many platforms, including their email. The reason this security is needed, so that if anyone with access to the database, imaginary employees, can’t just steal users' passwords and gain access to random accounts. Also, if someone manages to break security and gains the tables, it adds a little extra layer of safety. With the use of strong peppers, maybe slow down hash cracking tables.
3. Simple layout - The site is meant for people who are new to fantasy sports, and so it is our goal to make a site that keeps the pages simple and gives quick access to the most important parts of the game.
4. New player assistance - Again, the site is for those new to fantasy sports, or possibly sports in general, so we want to take every opportunity we can to better explain what is going on, why it’s important, and how it affects the game. This is listed as both a functional and non-functional requirement because while there will be usable elements in it, it is also kind of a design philosophy for our site.
5. Code Consistency - Ensuring that we all follow our coding guides so we can readily reference each other's work and guarantee uniformity across our pages
6. **Use Case Diagram**

****

1. **Class Diagram and/or Sequence Diagrams**



1. **Operating Environment**

Many of the environments being used are local hosts on each person’s machine. All of us use different operating systems such as Linux, Windows, and Mac. Those of us testing and building the backend (php) need to run a webserver. The web server runs the php before sending the resulting code to the clients. Apache is the web server on our local host. We will create a system to officially host the site for testing, and display. It uses a hosting site, named Dreamhost. Dreamhost provides us with shell, sftp, phpmyadmin (This is an easy to use interface for databases), and MySql databases. This allows us to test our code on machine that we all can use, creating a uniform resource.

1. **Assumptions and Dependencies**

One dependency is the use of embedded twitter frames. While it is unlikely for Twitter to change the way their frames work since it would break so many sites, it is still something we need to be aware of. Our home page also currently uses a recent games frame from sofascore.com. This could be something we do on our own in the future once our game database is working, but for now it works fine. Also on the home page, we currently have a stand-in chat system provided by minnit.chat, which is just standing in for the chat system we will make in the future. We are dependent on Sports Open Data API. However it has been running since 2016 so it is unlikely this will change.