**Software Requirements and Design Document**

**For**

**Group <8>**

Version 3.0

**Authors**:

Megan Achurra

Dylan Giesler

Daniel Lukish

Miguel A. Perez Ojito

Thomas Rudolph

# Overview

Django’s Fantasy is a fantasy soccer site focused on attracting newer players to the world of fantasy sports. We plan to offer exhilarating features such as: drafts, leagues, and a point system, but we will also lower the barrier to entry and soften the learning curve to help make things less daunting for new players. Tool tips and suggestions should help new players feel a little more informed, and comfortable when choosing a team or checking up on their progress. Knowing which data points are and are not relevant to a newcomer can be one of the most difficult tasks, so we plan to make a team information system that will help a person choose a team that is right for them by providing statistics along with context for those statistics. We will also be using a simplified point system where if your team wins a match you gain a point. That’s it, no frills. Just a simple, easy to use system that can help get your novice friends right into the action with more experienced players.

# Functional Requirements

1. Create Leagues – High Priority

Creates the league owner based on who creates the league. Allows the invitation of other users to the league.

2. Trades - High Priority

Allows users to trade teams with one another as long as they are in the same league.

2. Invite To League – High Priority

League owners can invite any other user to join their league.

3. Drafts – High Priority

League owners can begin a draft at any point after they have filled their league with users. Then they can begin a draft where users in the league choose teams in a snake draft order.

4. Point System – High Priority

The metric by which users are compared to each other. You gain a point when your team wins or when you correctly guess which team would lose.

5. Login – High Priority

Allows access to the site after signing up. Required to get anything useful from the site.

6. Sign Up – High Priority

Allows the users to create an account, which will allow them to login with the same credentials in the future.

7. Forgot Password - High/Medium Priority

Allows users to be able to get back into their account if they forgot their password. PHP was written to generate one valid reset link with a long string of random chars. This way, a request must be made, a link sent to the users email, and the link must match what we have in the database to work.

7. Team information system - Medium Priority

Allows the user to easily see a page full of statistics and explanations about all the teams in the premier league.

# Non-functional Requirements

1. Hashing Passwords - High Priority

For any kind of system that saves passwords, hashing should be included. This way if any databases get leaked, it makes it harder to compromise the other service accounts using the same email/password combo.

2. Trade System

Just because there was a big UI update I wanted to make sure that its addition would not break the rest of the site’s layout.

# Use Case Diagram

# Class Diagram and/or Sequence Diagrams

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 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.

# Assumptions and Dependencies

*List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*

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. The team information system also uses frames from footystats.org. It is also something we would change to our own system if we had more time, but the footystats frame gives all the information we wanted to display anyway. We are dependent on Sports Open Data API. However it has been running since 2016 so it is unlikely this will change. One last assumption is that people will also understand how the trades work because it is more picture oriented rather than overly explanatory. So what they want is on the left and what they’ll give for it is on the right.