**Software Implementation and Testing Document**

**For**

**Group <8>**

Version 3.0

**Authors**:

Megan Achurra

Dylan Giesler

Daniel Lukish

Miguel A. Perez Ojito

Thomas Rudolph

# Programming Languages

PHP: PHP, along with MySql, allows us to add a ton of functionality while being secure. Using this as a combo gives use the ability to store and use any data we choose. It will give us the ability to login, signup, create leagues, store login attempts, and even create a custom chat. PHP also gives use the ability to create a session, thus keeping us logged in from page to page. Also, because it is a server side language, the code is run before it is sent to the client. This means the users can’t see the PHP code, adding an extra layer of security.

HTML/CSS: These are being used for designing the different sites and layouts. So they determine what the colors are, what font we use, and what the overall site looks like.

Javascript: Being used sparingly and only for some basic scripting, like on the home page where you can select between the standings and your current league scores and for the tooltips. The main reason for the use of Javascript is due to how closely integrated Javascript is with any modern site.

Python: Removed.

# Platforms, APIs, Databases, and other technologies used

MYSQL: Using MYSQL was a complete freebie for us. The fundamental reasons for choosing MYSQL for our database are: Two of us just took the course this past summer and have it fresh in our minds, and PHP works perfectly with MYSQL out of the box so there is little to no configuration that we needed to do on our end.

API - API used: [sportsop-soccer-sports-open-data-v1.p.rapidapi.com](https://rapidapi.com/sportsop/api/soccer-sports-open-data)

We receive information on each teams: Wins, Losses, Draws, Points, Scores(Goals), Conceded(Goals), and number of matches played. Our API is limited to 100 requests a day, which should be no problem for our work.

# Execution-based Functional Testing

Password Reset - We generated many password reset links. We made sure that if a new password reset was generated, then all the old ones generated were invalidated. This was to make sure someone did not attempt to generate thousands of links to make it easier to set a new password. The link also requires an email to be passed in, We made sure that links only worked for emails that the password links belonged to. Last, we check the validation of the link to see that it can only be used one. We did this by trying multiple links. \*Note: one issue: the emails are generally marked as spam, and put in the spam folder, unfortunately.

Drafting - We checked to make sure that teams picked could not be re-picked by a different user in the same league. We wanted to make that a user in two different leagues could pick the same team. We tested this by creating many tests users who each belonged to different leagues of different sizes. We then proceeded to pick teams, thus testing the draft.

# Execution-based Non-Functional Testing

Loading the website on computers with different resolutions to make sure things don’t break at different sizes. We did this by resizing the page to extremes and used different monitor resolutions (1920x1080, 3440x1440, 1440x900) so that we could make sure everything stayed where it was supposed to be no matter what screen you were on. Loaded pages on

# Non-Execution-based Testing

Like in the previous iterations, we mostly just showed each other what we had been working on when we were around each other. This gave us a chance to show what we had been working on and how it works. We also individually went through each page in the sight and tested all the buttons and functionalities to try and catch any bugs. We were able to discover a handful of issues in this way most of which being quick fixes.