

Team 

Fantasy Sports App

Technical Design Document

Jon Warren, Katie Myers, Jack Tolton
11-1-2017

Part 1

Target platform: *Windows Desktop application and iOS Mobile application*

Development platform: *C# for desktop application, Swift for mobile application, MySQL for our back-end*

Source code hosting: *Source code will be hosted via git located at <https://github.com/Tolton/Fantasy-Sports-Application>*

Third-party APIs: *Player stats will be pulled from <https://www.mysportsfeeds.com> (free registration is required for use of the API)*

Other assets required: *Player mugshot pictures will be pulled via TSN's image gallery API using the following format:*

`http://tsnimages.tsn.ca/ImageProvider/PlayerHeadshot?seold=[name]`

Other images used for this project will be either be created personally by members of our group, or public domain images will be used.

Hosting: *We will be hosting our database with Amazon's Relational Database Service in the free-tier version (only accessible for 12 months).*

Part 2

Project Description

For our project, we are going to be designing a fantasy sports application. Our fantasy sports application is essentially an online game where competing participants will be able to assemble their own imaginary teams from one of the 4 major North American sports leagues (however, we will focus solely on NHL) using their selected players' actual statistics from real games as a means of scoring points.

The core application revolves around 3 key features that are essential to any fantasy sports application: the draft, the team and the league. Detailed explanations are given below:

The Draft

The draft is an event that takes place where all competing participants in a league select their players (in either a pre-determined or a randomized order) to their fantasy team. The event happens in real-time, and aside from league/team configuration, it's the starting for every fantasy league. Once a player has been drafted to a participant's team, that player is no longer eligible to be drafted to any other participant's team.

The Team

Each participant owns and operates their own team, consisting of a roster of their selected players from the draft. The way you play fantasy sports is by managing your team to make it as good as possible, by either adding players, dropping players or trading with other teams in the league to better your own team. Participants are only allowed a certain number of players on their team, so these player acquisitions and roster modifications are very important to a team's success.

The League

The league acts as a set of rules and guidelines that dictates how the game is played. League configuration dictates how players score points, how many players they are allowed to have on their team, how often they are allowed to add/drop/trade players. The league also keeps track of either weekly or seasonally team stats which in turn dictates the standings within the league. All teams within a league play by the same rules dictated by the league commissioner (the participant who creates the league and defines the rules of the league).

Requirements

Must Have	Should Have	Nice To Have
The ability to register an account and login to it	The ability to change your team name freely after it has been decided	A method to reset your password if it is forgotten
The ability to create a new league from your account and determine rules and scoring settings, as well as making the league private or public	The ability to send other participants within your league messages	Restricted offline play (be able to save information locally and view cached team/player information)
The ability to join a pre-existing league openly if public, or enter the league name and provide a password for joining a private league	The ability to have the league auto-select players if participants are not present for the draft	Determine a league treasurer and provide PayPal access so that participants can buy-in to the league if it's a cash league
The ability to create a team within a league and customize the team name	The ability to change league settings after a league has already been created	The ability to set up a real-time live draft involving all participants in the league
The ability to initially draft players for your team	The ability to trade players with other teams in your league	Provide player injury status (if the API provides this information)
The ability to add and drop players to a team	Optional notifications when your team gets points, or when another team requests a trade with you	The ability to veto trades between other teams in your league.
The ability to log out of your account after you're logged in as well as switch your currently viewed league	Player status updated at the end of the day after all games have concluded (11:59pm PDT or 2:59am EDT the following day)	The ability to add or remove teams after the draft has already taken place.

Milestones

Week 3 (Sept.22 – Sept.28) – Work on TDD

Week 4 (Sept. 29 – Oct. 5) – Graphics and Base GUI (front), Database Structuring (back)

- **Jon and Katie** will work on look of the application and agree upon how navigation around the app will take place. At the end of this week, all of our custom graphics should be complete, and navigation within the app should be for the near-fully functional (a lot of actual content will be missing, however).
- **Jack** will be designing data structures and tables within our database to store league information as well as team information. Player stats will not be stored on our back-end (as that would be redundant to pull from our API and upload to our server), however, information regarding the teams and leagues (such as team names, rosters and associated user, as well as league settings and standings) will need to be stored.

Week 5 (Oct. 6 – Oct. 12) – Registration and Login (front and back)

- Week 4 will be dedicated to creating our registration and login functionality.
- **Jon and Katie** will ensure that the front-end presents a clear and easily accessible method for both registering and logging in. They will also ensure that user information is passed to the server in the same manner to ensure consistency across both desktop and mobile platforms.
- **Jack** will ensure that user information passed to the server is handled in a secure manner, including the storing, hashing and salting of the user information on the server. He will also be responsible for protecting against collisions and potential SQL-injections.
- ****TIME PERMITTED****: We will also include the ability to reset your password either by email or by answering security questions.

Week 6 (Oct. 13 – Oct. 19) – Team & League Structure (front and back)

- **Jon and Katie** will be creating an interface for users to create and manage their leagues, as well as creating (but not yet managing) teams.
- **Jack** will ensure that information regarding leagues and teams is stored properly and protect against duplicate team names, invalid names and invalid setting choices. Jack will also ensure that the date is kept track of properly to ensure that player stats are updated at the correct time.
- After this week, users should be able to log in to their account, enter one of the 4 sports and either create or join a league. On top of that, players should be able to create a team with basic information (name and picture).
- ****TIME PERMITTED****: We will also include the ability for teams to send messages to one-another within the league.

Week 7 (Oct. 20 – Oct. 26) – Team Manipulation (front and back)

- Week 7 will be dedicated to ensuring that rosters are near-fully functional.
- **Jon, Katie and Jack** will work together to ensure that users are able to seamlessly alter their roster including both adding and dropping players (player trades will not yet be functional). Team rosters will be visible to all participants, and league standings should be visible to all participants as well. Live draft will not yet be included, and when a team is created, they will be able to pick up available players all at once that have not yet been selected by other teams.
- All 3 of us will also prepare for our upcoming Alpha demo on Thursday of this week.

ALPHA DEMO

Week 8 (Oct. 27 – Nov. 2) – Alpha regroup and fixup (front and back)

- **Jon, Katie and Jack** will work on completing any missing functionality from the Alpha presentation and ensure that their respective parts are up-to-date with respect to all milestones before the Alpha. All bugs (with respect to previous milestones) should be fixed. They will also all work together on the updated Technical Design Document revisions for the upcoming Thursday.

Week 9 (Nov. 3 – Nov. 9) – League Rules and Scoring (front and back)

- **Jon and Katie** will add the rules interface during the league creation screen. This should allow users to specify exactly how teams are to score points and other league rules (number of player adds per day/week, number of players on a roster, number of players per position and other rules as we see fit). On top of that, they will ensure that the rules are visible to all participants of the league and work with **Jack** to ensure that players are not able to break the rules.
- **Jack** will create a data structure to store the rules and scoring system of the league as well as provide the API for reading and interpreting those rules. He will also work with **Jon and Katie** to ensure that users will not be able to break the rules.

Week 10 (Nov. 10 – Nov. 16) – League privatization (front), League security (back)

- **Jon and Katie** will add the option for users to make their league either private or public. Public leagues will not require a password while private leagues will. On top of that, all leagues should be displayed when the user goes to Join a League (rather than entering the name), and only private leagues will prompt you for a password.
- **Jack** will ensure that our database is able to differentiate between private and public leagues, as well as storing the password for just private leagues. On top of that, he will ensure that league passwords are also salted and hashed (at Alpha, only user login information was salted + hashed).

Week 11 (Nov. 17 – Nov. 23) – League Standings (front and back)

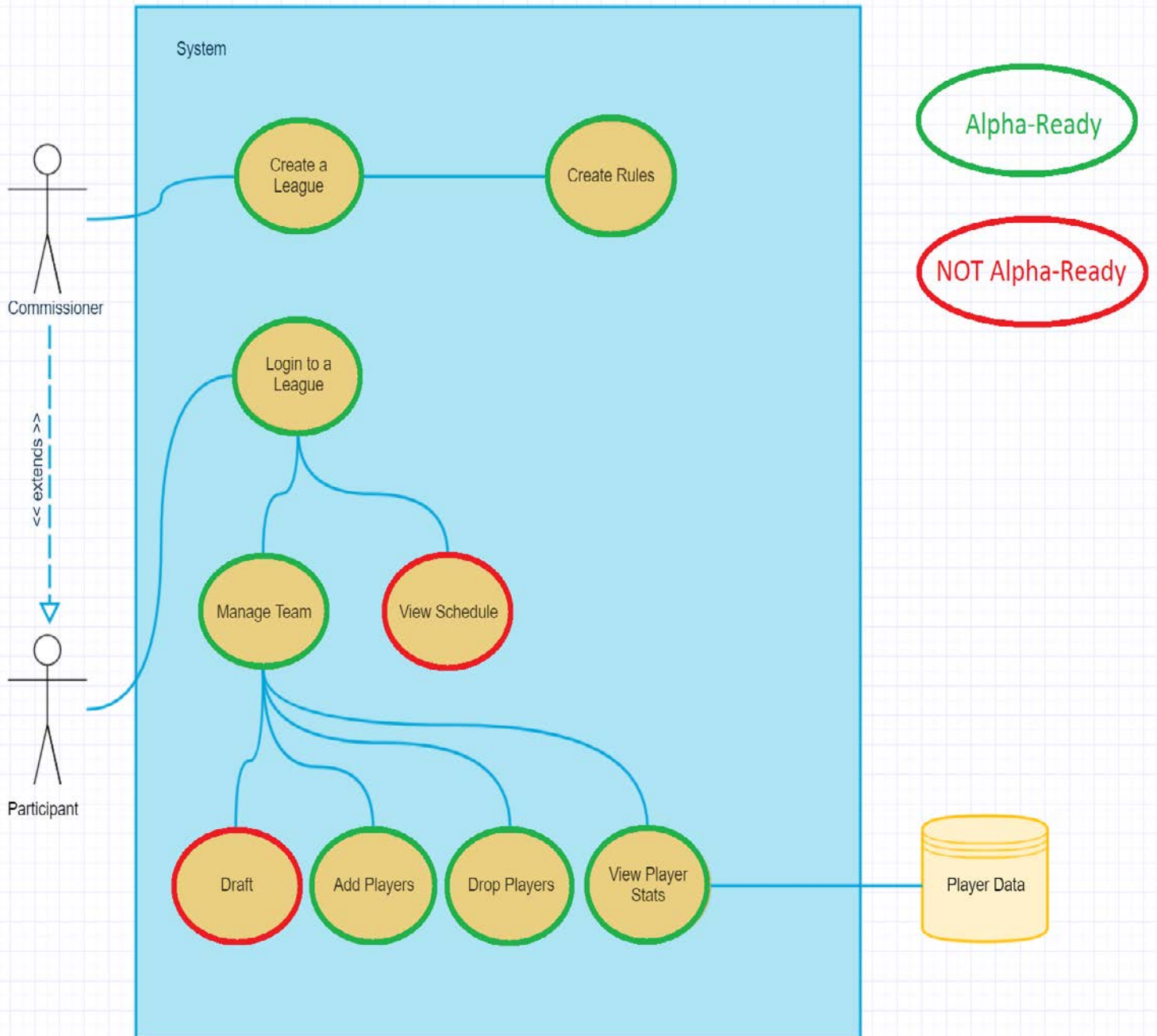
- **Jon and Katie** will ensure that the league standings are shown to reflect the rules put in place by the commissioner of the league. On top of that, stats relevant to the league rules will be shown when attempting to view player stats.
- **Jack** will ensure that league standings are updated on the backend depending on the date (as our free API access allows only last year's stats – so we need to treat it as a simulation). He also needs to ensure that dates that the player was acquired & dropped are stored in the database to ensure that participants only receive points from players who were picked up after their add date.

Week 12 (Nov. 24 – Nov. 30) – RC Demo Prep / Final touch-ups (front and back)

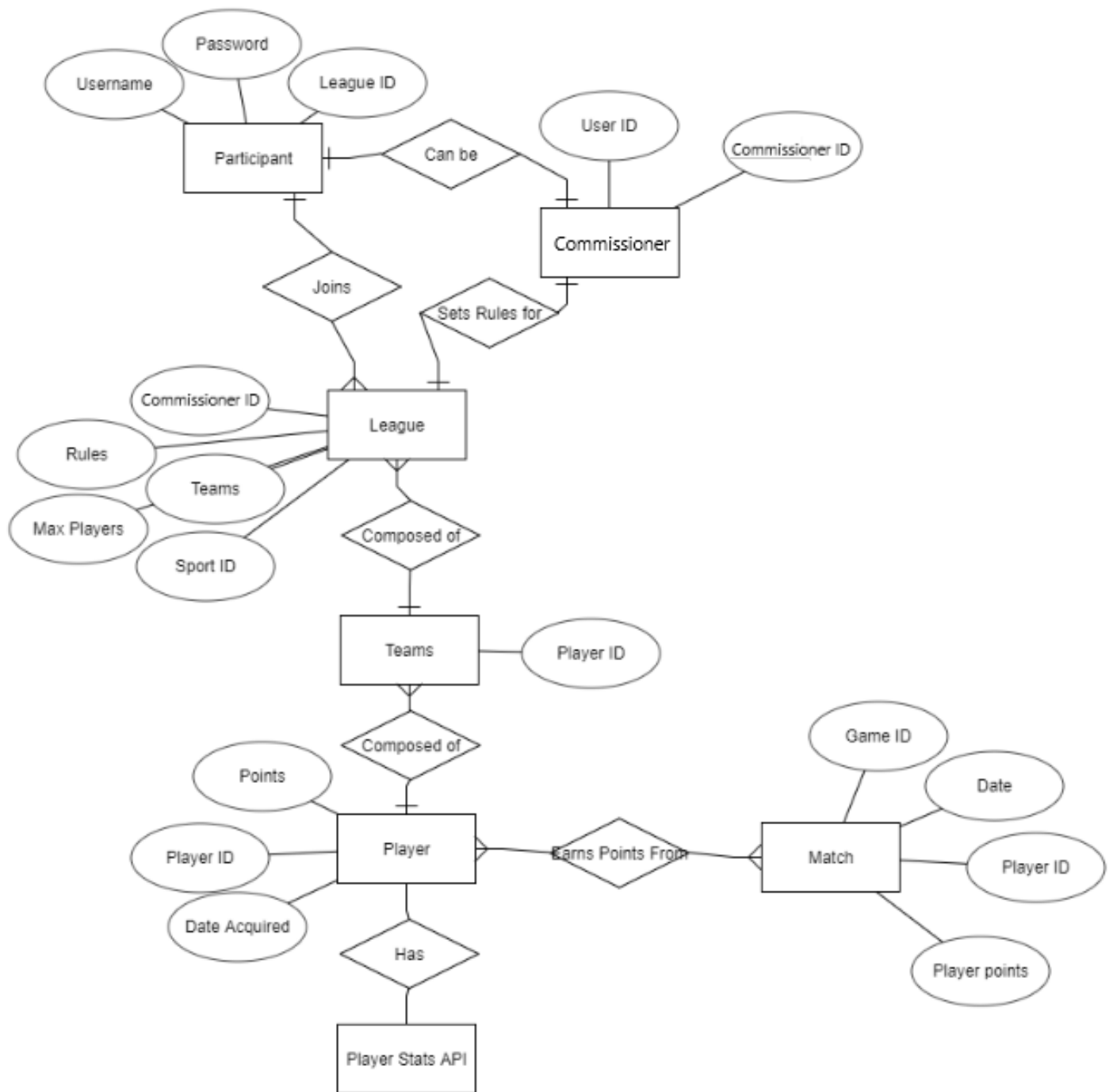
- **Jon, Katie and Jack** will ensure that the RC Demo is ready! Final touch-ups including thorough bug-testing, SQL injection tests and stress testing will be done to ensure there won't be any issues. **Jon and Katie** will perform prototyping sessions for their respective applications with colleagues in preparation for the demonstration. **Jack** will research potential SQL injection threats and will ensure that the program will be impenetrable in that regard.

Part 3

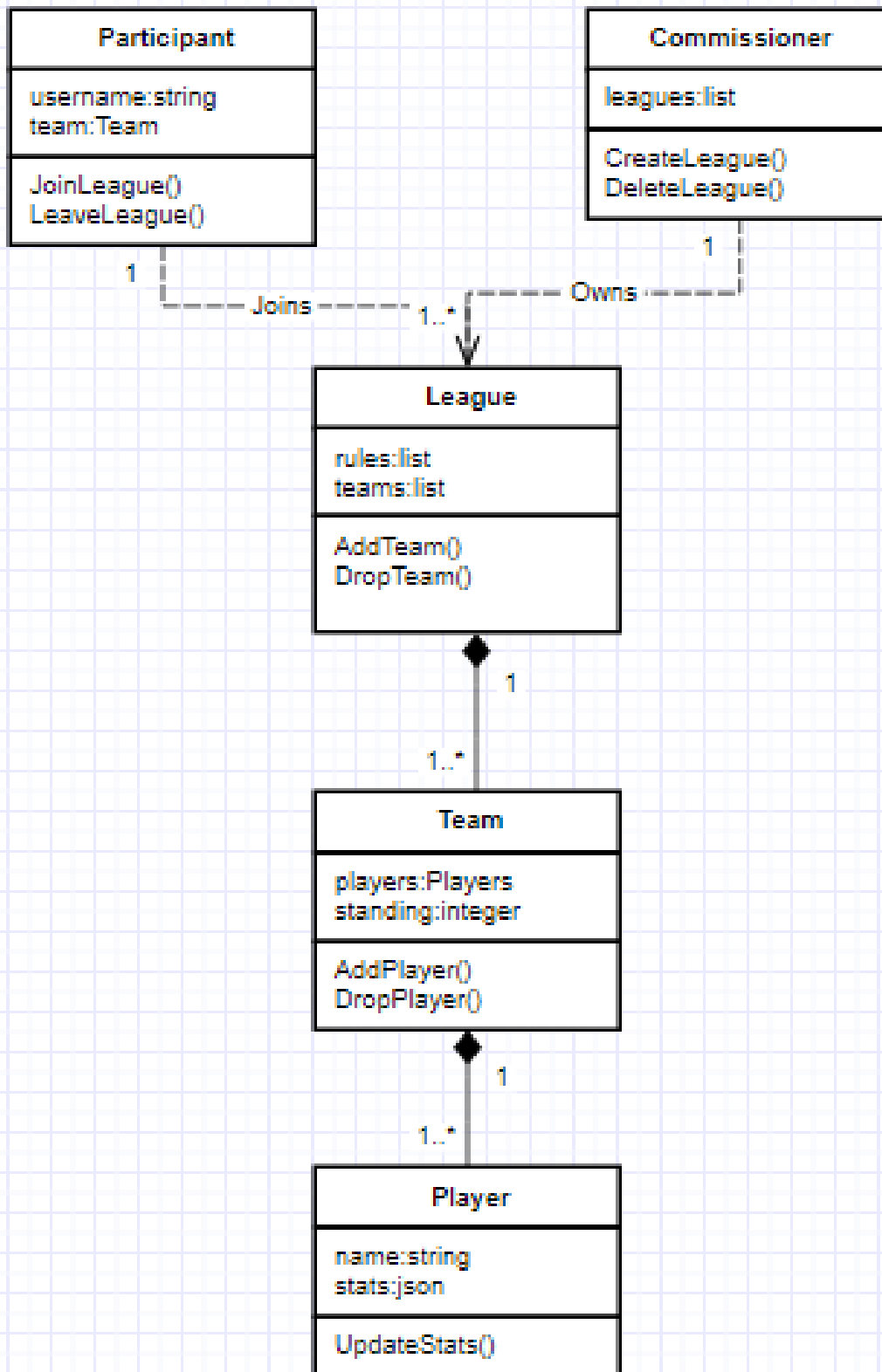
Use Case Diagram



Entity Diagram



Class Diagram



UI Mockups



Football



Hockey



Baseball



Basketball



Welcome, TestUser!

Log Out



Test Player
#99, Center
5'10", 200lbs

Player Name	Pos	G	A	Sh	Hits	PPP	W
Test Player	C	12	15	77	123	10	4
...							



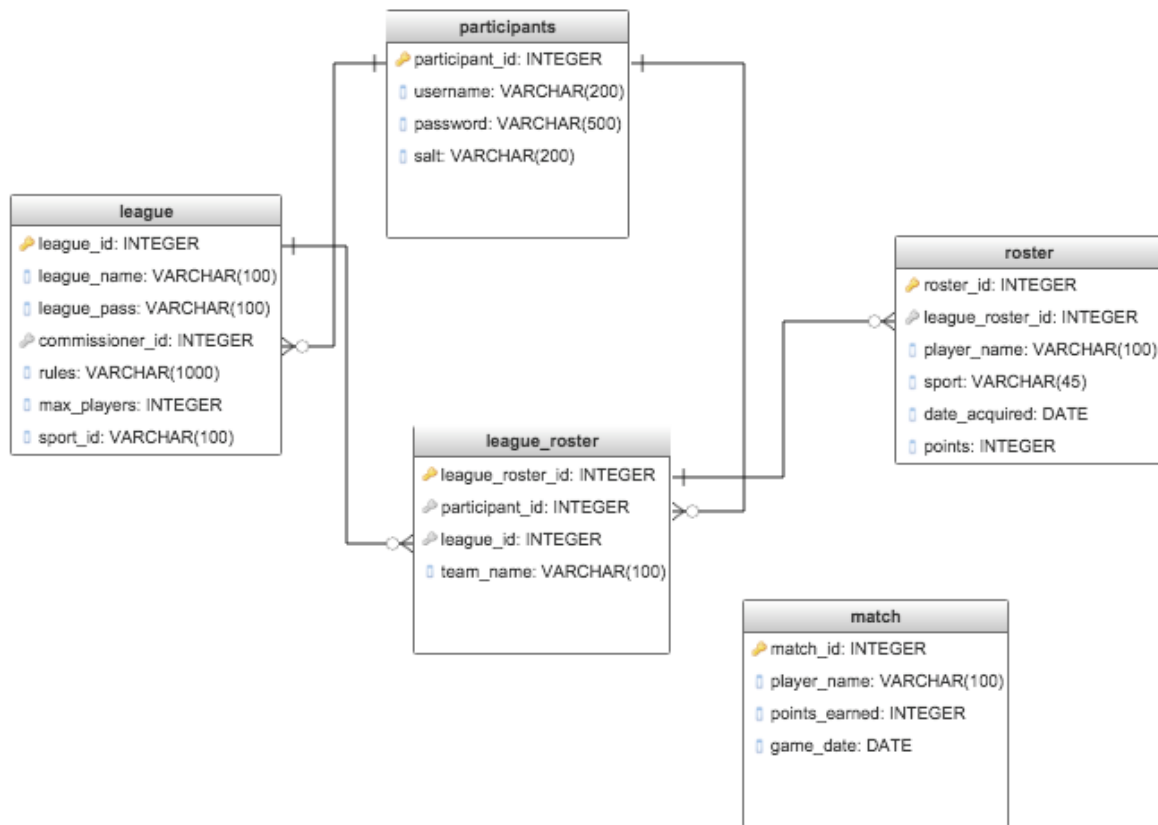
Welcome, TestUser!

Log Out

Week 13 (January 1 - January 7)

Test Team!! vs TestTeam2

Goals	4	5	★
Assists	8	12	★
Points	12	17	★
Shots	64 ★	55	
Hits	104 ★	89	
3		2	



Individual Write-Ups

“

I had a lot of fun with this project; it really motivated me to put a lot of work into this as it was a topic that we decided on doing rather than one assigned to us. I learned a lot along the way, with the number one thing being not to bite off more than I can chew. A lot of the milestones had me putting 1-2 hours of work into each day, and then putting 15-20 hours over the last 2 days of each week because it was going to be more work than I had anticipated. It really gave me a good perspective on setting reasonable timelines to accomplish milestones.

I also learned that documentation is a lot more important than I had initially given it credit for. As someone who absolutely hates writing up the documentation and would rather just try and code my way out of any issues, I was continually checking my milestones as well as class/entity diagrams to keep me on-track and to figure out what needed to be accomplished next and how. The documentation was also super helpful in maintaining consistency between the front and back-end code; I never had any issues connecting my front-end to the back end because the commenting on the back-end code was very-well commented and easy to port over into my front-end's native language (C#).

Finally, this course really taught me the value of designing and utilizing proper classes in object-oriented programming. This was my first experience using C#, and all of my previous object-oriented experience had been through Java (which I hate). I initially started coding this with a procedural-style and ended up having to redo most of my code later on in the course in order to accommodate for different classes. If I had used classes properly to begin with, I definitely could have saved a lot of time and effort by keeping it simpler from the start.

”

-Jon

“

I would say that the biggest learning point for me this semester was to write an entire project in a language that I had not worked in before. Having been familiar with Objective-C, it was quite a jump to move to Swift 4 given the new nuances of the languages, as well as re-learning how to use the Xcode environment. I can definitely say I appreciate it now that I'm more familiar with the structure of the language, however it did initially slow down progress. One of the biggest challenges I faced with Swift explicitly was the shift between versions, as much of the code online is for Swift 2, and the Xcode IDE does not always provide a solution to the errors it throws. I also learned more PHP this semester, having only worked once before with a small PHP script, so being able to write my own server was a very useful experience. I can also say that the documentation helped a lot, having defined milestones and constant miniature submissions really helped me keep the project on track, and I rarely found myself falling behind. It was a much better form of development than the typical last minute rush I find myself doing in other courses.

Many of my challenges this semester revolved around the iOS app interacting with the backend code that Jack wrote. Due to Apple's requirement of HTTPS endpoints, and most examples online including PHP endpoints, I ended up having to spin up a server to handle the API interactions. I also had issues with the company hosting my server, where they frequently went down or locked me out of my account. Thankfully, by the end of the semester, I managed to find a more reliable server, however this downtime cause a lot of issues during testing and demonstrations. Another challenge I found was the time I could dedicate to this project. I was taking six courses this semester, and could not dedicate the time this project needed for improving the design efficiency of the code, therefore the appearance of the app and the code itself feels very messy to me still. Overall though, a great learning experience for mobile development on iOS.

”

-Katie

“

A learning point for me this semester was how vital it is to plan out how the data structures need to be implemented and how they will interact with the other tables. It could get frustrating at times trying to truncate a table, only for the SQL query to fail due to a foreign key restraint. Also adding in other attributes to tables, but needing to have them set as a Non-null, then receiving an error due to everything already in the table getting set as a null for that column.

Furthermore, learning new ways of manipulating SQL queries, such as subqueries was also a new learning point and quite interesting seeing how you could make a complex query in order to get a specific set of data back. I had used Python to write a backend in a previous course, so for the most basic queries and other manipulation for things like json data, it was relatively easy. It was nice to be able to follow the documentation and use it as a checklist for things I had yet to do, as well as things I had completed. Although I still dislike documentation since usually when I start writing it out, I get eager to start coding and want the writing to be over with already.

The biggest challenge for me this semester was trying to find hosting solutions for the API and the interaction it has with the front ends. The requirements were heavy due to needing a Python interpreter as well as using HTTPS and of course being free was preferred. Overall, this was a very enjoyable project and helped me learn a lot about app development in small teams.

”

-Jack