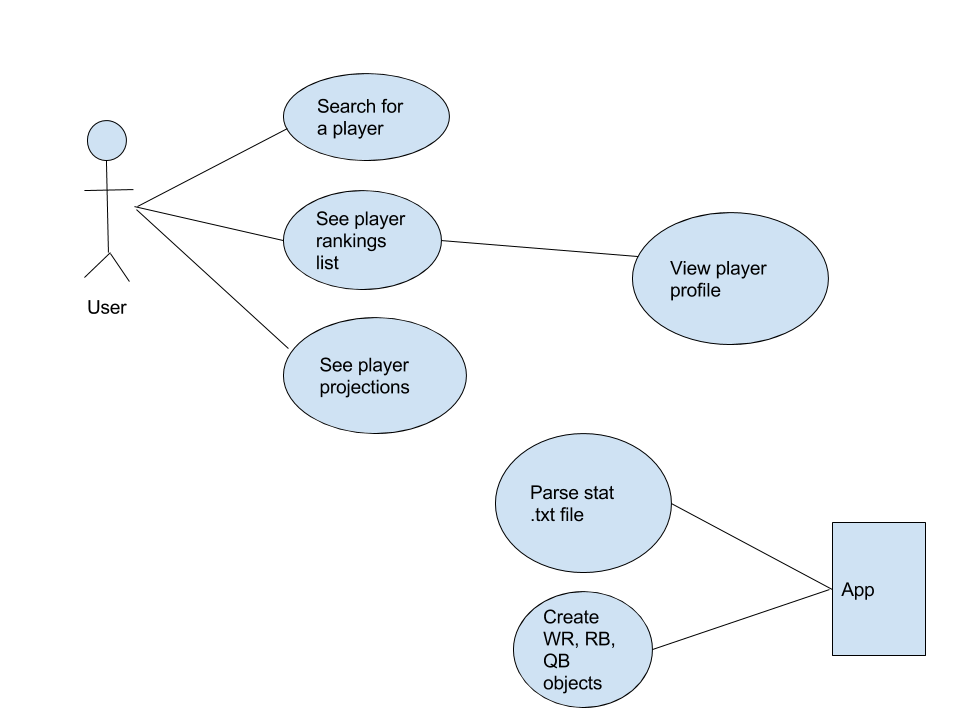
CJ Botlinger, Ryan Fritz, Drew Sadik

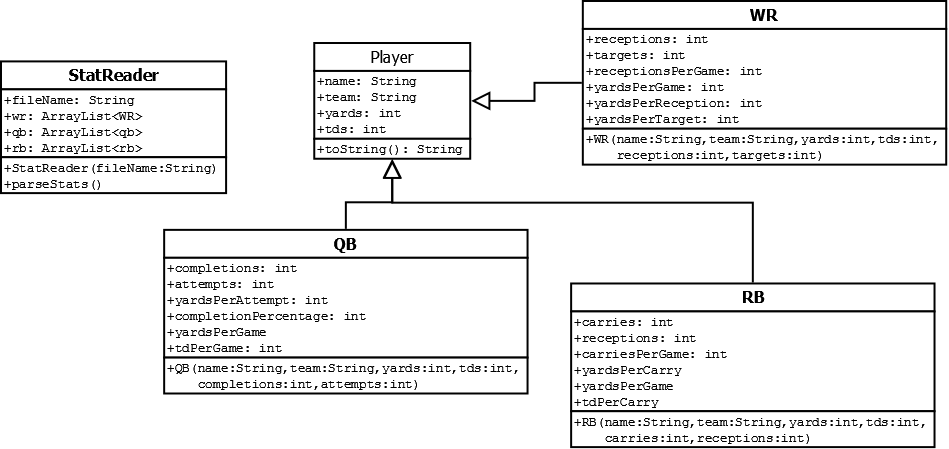
IST 311

Group 3

Fantasy Football App

Our application will be a one-stop shop for football fans who would like quick and easy access to interesting and relevant statistics. The application will read in basic stats for the top 20 WRs, RBs, and QBs, and store the data. Based on the basic stats received, it will calculate more complex efficiency-statistics that go a little deeper to show how a player has been performing. The menu will give the user the option to view a list of the top WRs, RBs, and/or QBs, and the player can look further into a single player’s stats by clicking on the player’s name and accessing his profile. If time permits, we would also like the application to take a look at the upcoming week of games, and recommend certain players to start based on matchups.





**Product backlog:**

Create Player class

Create WR class

Create RB class

Create QB class

Create Stats class

Convert to JavaFX and Create GUI Menu

Convert Console Data to Display on GUI Menu

Get Menu to Display Positional Stats

Create Buttons to Toggle between Positions

Create Player Profile GUI Menu

Show Menu Upon Clicking on Player

Store Schedule (Matchup) Data

Create Logic for Prediction

Create Scene for Weekly Prediction

**Sprint 1 backlog:**

Create Player class

Create WR class

Create RB class

Create QB class

Create Stats class

**Sprint 1 Report:**

Player classes and Stat Parser work as well if not better than initially expected.

A few bugs regarding calculated statistics were not working quite as planned, but do not crash the program at this point in time. We will have to take a look and fix them in Sprint 2.

**Sprint 2 Backlog:**

Convert to JavaFX and Create GUI Menu

Convert Console Data to Display on GUI Menu

**Sprint 2 Report:**

Creating a nice GUI menu proved to be a fairly simple task. Converting all the data to show up correctly in the menu was more difficult. On the console version, getters and setters were not needed, but we realized that for our FX App to populate the GUI Table with data, we needed to implement getters and setter. Once we figured that out, the rest was just working out some of the kinks.

**Sprint 2 Burndown:**

**Sprint 3 Backlog:**

Get Menu to Display Positional Stats

Create Buttons to Toggle Between Positions

**Sprint 3 Report:**

Having the ability to toggle between various positions wasn’t too hard, but it was very important for our program. At first, if the user clicked the WR Button while WR was already being displayed, the program would append a duplicate set of WR data to the bottom of the table. In order to remedy this we needed to move our setColumns() method outside of the action listener. The program worked as intended after that. A few minor bugs existed in certain columns of the table, but simple changes to variable scope squashed them.

**Sprint 3 Burndown:**