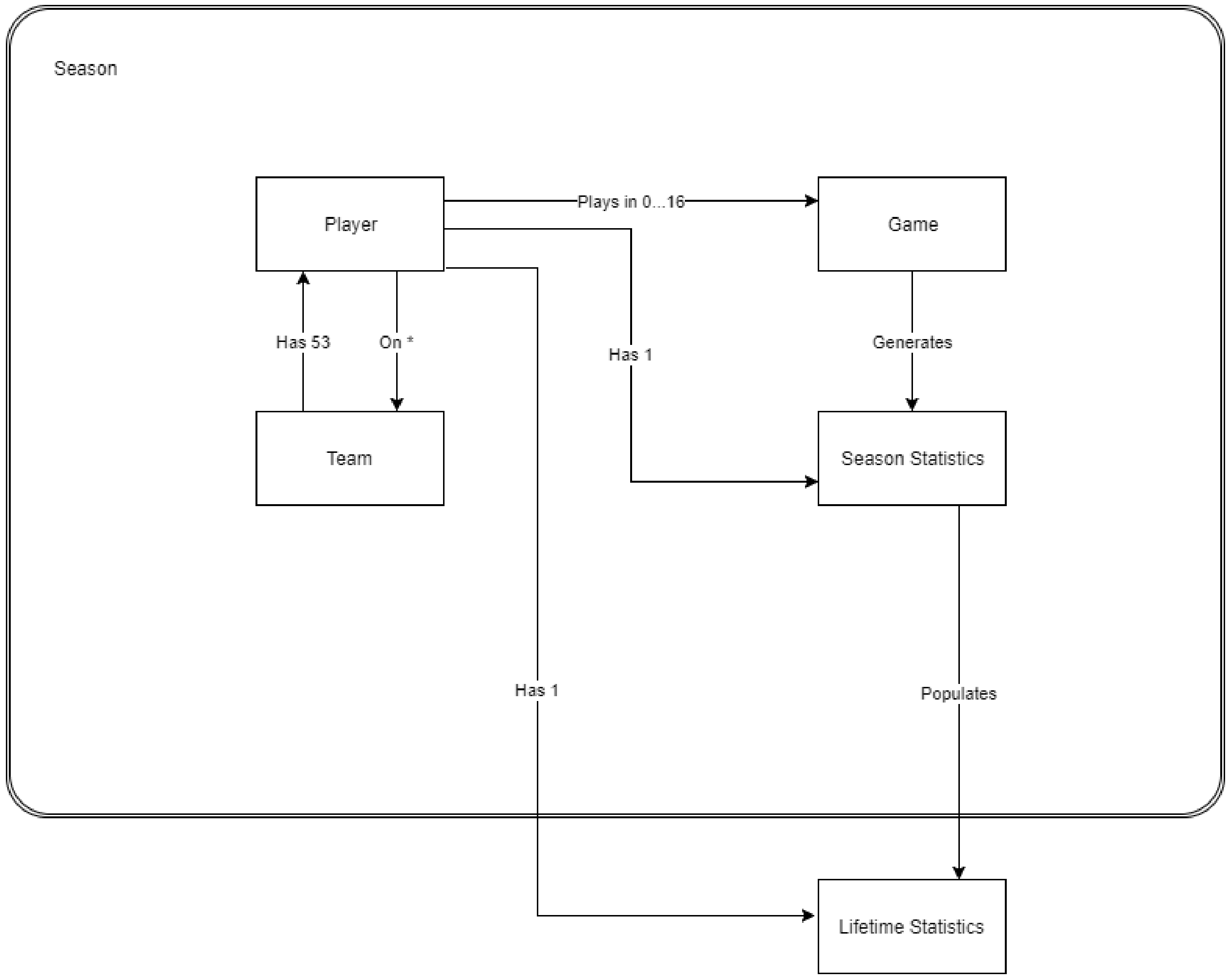
The Chumps
Sprint 4 Deliverables

Shane O’Brien, Jose Guvenilir, George Tsitsopoulos, Zachary Maurer

Domain Model



User Stories (Ask about specific names)

1. As someone looking to make a trade in fantasy football, I want to be able to get an in depth comparison of Tom Brady and Aaron Rodgers so that I can accept or decline this trade.

- We think that this user story fits very closely in line with our initial vision. Our plan is to bring up career, per season, and per game statistics of Tom Brady and Aaron Rodgers. Then, at the bottom, we will have fantasy point metrics. These fantasy point metrics will be our final say in who the user should take.
- Testing: We can test this by simply entering Tom Brady and Aaron Rodgers into our system and making sure our fantasy point metrics are visible on the screen. We would also like to see that any and all metric that could be used are easily seen and accessible. We will check for “season statistics”, “per game statistics”, “career statistics”, and “fantasy scoring”.

2. As someone who is trying to get into fantasy football, I want to have a website outline the important statistics in a wide receiver so I can know who to make trade offers with

- The team discussed this problem, and we feel that it is especially important. We want even a brand new fantasy football player to be able to use 120stat without issue.
- Testing: We have decided to accomplish this by utilizing our wide range of fantasy football experience. George and Shane have been playing fantasy football for years, while Jose and Zach have never played before. We want Jose and Zach to be able to understand all the football terms without much explanation from the user interface. Then, we will show the website to our friends who don’t play fantasy football, and see if they understand how to use the webapp. If not, we will adjust accordingly.

3. As someone who is in five fantasy football leagues, I want to spend less than 60 seconds looking up four possible trades so that I don’t spend too much of my day researching statistics

- As a team, we also understand the importance of this user story. If 120stat is cumbersome or slow, then there would be no point in its creation. We want 120stat to be an “all-in-one”, easy solution. The challenge, as we decided as a team, would be to give this user the metrics they want to see quickly and efficiently.
- Testing: We want to test this by creating a scenario in which a person has to look up four different trades and see how long it takes the person to get a conclusive answer on each one. We will physically time the user experience with a stopwatch. From the beginning of typing in the names to locating important information.

4. As someone who is new to fantasy football, I want to have a website tell me who has more fantasy value between LeVeon Bell and Antonio Brown

- We decided that this comparison could be a bit tough. LeVeon Bell and Antonio Brown are two different positions, so we would need to account for that when giving our final answer. We were thinking a conversion rate between the two positions could be used to determine our final metric.
- Testing: We want to test this simply by putting these two players into the webapp and making sure our final metric reflects their fantasy value. We will manually calculate

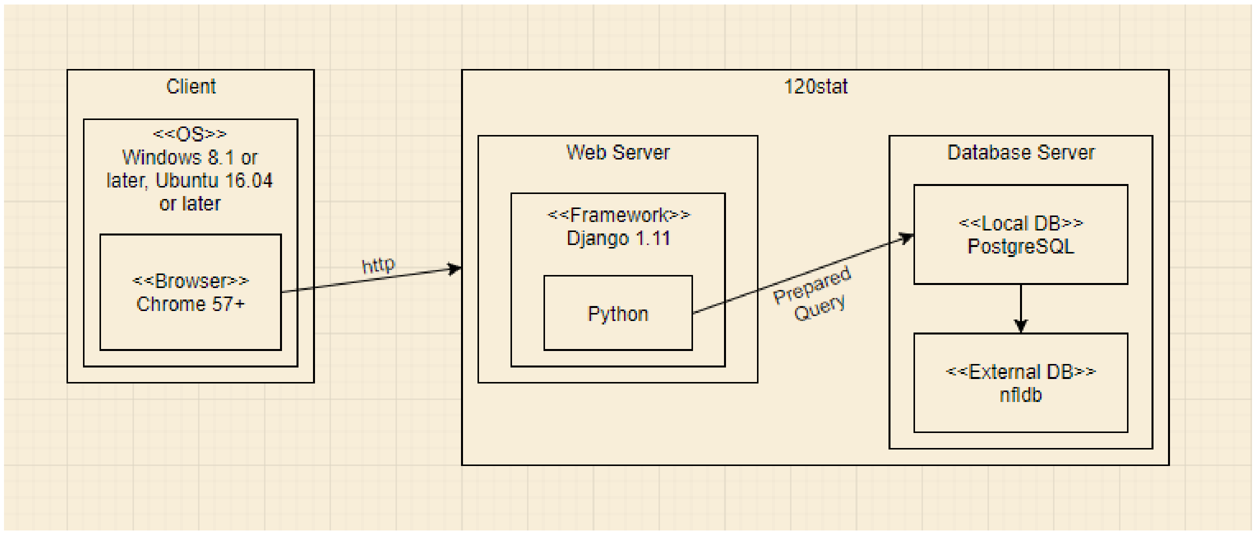
fantasy value over the last three years and compare that to our application. If our player evaluation shows that the fantasy value is correct, we are in the clear.

Supplemental Specifications

Must Have	
1.01	A user shall be able to search a player's name
1.02	A user shall be able to search for up to four players at once
1.03	A user shall be able to view cumulative yearly statistics
1.04	A user shall be able to game log statistics for a year's worth of games
1.05	The user shall be able to view fantasy football statistics for each player where the default scoring system is ESPN standard scoring
1.06	The website shall prevent SQL injections.
1.07	The website shall have an "About" page that will tell the user the how to use the website and talk about the project as a whole
1.08	The website shall work on Google Chrome
1.09	The UI shall display all possible basic statistical categories a general player may have regardless of position (Passing yards, completions, passing touchdowns, rushing yards, rushing attempts, rushing touchdowns, receiving yards, receptions, receiving touchdowns, field goals attempted, field goals made, interceptions, fumbles)
1.10	The website shall have a navigation bar at the top of each page and so the user can access the home page or about page
Should Have	
2.01	The UI shall show only passing and rushing statistics when viewing a quarterback's stats
2.02	The UI shall show only rushing and receiving statistics when viewing a runningback's stats
2.03	The UI shall show only receiving statistics when viewing a wide receiver's or tight end's stats

2.04	The UI shall show only kicking statistics when viewing a kicker's stats
2.05	A multi-player search across multiple positions shall display statistics only for relevant categories relating to each player
Could Have	
3.01	The user shall be able to click on a season and the view shall change to display the game statistics for that season
3.02	The website shall autofill the player name field with valid NFL player names as the user enters a name
3.03	The database storing player information will be updated at 5:00 AM ET daily
3.04	The user shall be able to view statistics from only a single-season (won't see all other seasons)
3.05	The user shall be able to view yearly cumulative statistics for the years since 2009 (won't include individual games)
Won't Have	
4.01	A user shall be able to filter statistics to show regular season and postseason statistics
4.02	A user shall be able to filter game statistics to show preseason, regular season, and postseason statistics
4.03	A user shall be able to filter game statistics to show postseason statistics

Deployment Diagram



Use Cases

Many of the possible use cases for 120stat are very similar. As a result, we picked three specific use cases that we feel cover most of the core feature of the webapp.

	Use Case: CompareTwoRBsSeasonStats
Identifier:	UC1
Description:	The CompareTwoRBsSeasonStats use case models a User comparing two RBs
Actors	120stat webapp User
Preconditions:	User is interested in comparing two RB for per season stats, and is on 120stat homepage
Flow of events:	1. User case starts when the User is on the homepage 2. User enters both running back names on the bottom of the homepage 3. User hits “Begin” 4. 120stat brings user to a new page 5. ● If 120stat finds the players, both RBs names are displayed on the screen, with their career stats,

	season stats, game stats, and fantasy stats below. The user now has easy access to the information they desire (season stats). ● Else, 120stat displays an error message and shows no info for the players. Gives a link back to the homepage.
Postconditions	The players' final information is displayed on the page, and the user has been given the information to compare seasons between two RBs

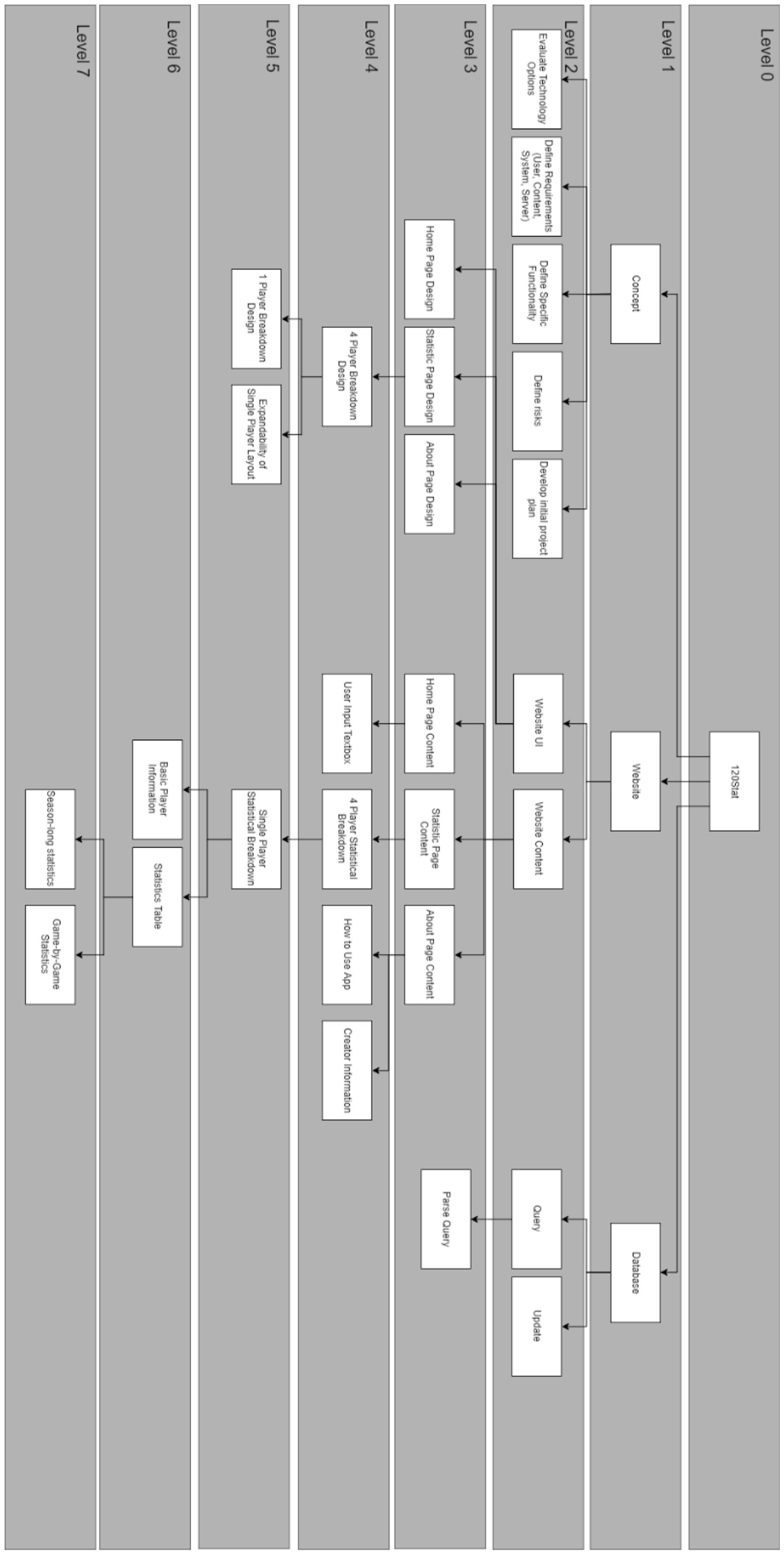
	Use Case: CompareWRRBPerGame
Identifier:	UC2
Description:	The CompareWRRBPerGame use case models a User comparing a RB and a WR
Actors	120stat webapp User
Preconditions:	User is interested in comparing RB and WR for per game yardage, and is on 120stat homepage
Flow of events:	1. User case starts when the User is on the homepage 2. User enters both running back names on the bottom of the homepage 3. User hits "Begin" 4. 120stat brings user to a new page 5. ● If 120stat finds the players, both the WR and RB names are displayed on the screen, with their career stats, game stats, and fantasy stats below. The user now has easy access to the information they desire (game stats). ● Else, 120stat displays an error message and

	shows no info for the players. Gives a link back to the homepage.
Postconditions	The players' final information is displayed on the page. The user has the information they came to 120stat for

	Use Case: CompareQBWRRB
Identifier:	UC3
Description:	The CompareTwoRBsSeasonStats use case models a User comparing a QB, a WR, and a RB for a trade (QB&WR for RB)
Actors	120stat webapp User
Preconditions:	User is interested in comparing a QB, a WR, and a RB for a trade, and is on 120stat homepage
Flow of events:	1. User case starts when the User is on the homepage 2. User enters the QB, the RB, and the WR name on the bottom of the homepage 3. User hits "Begin" 4. 120stat brings user to a new page 5. ● If 120stat finds the players, all players names are displayed on the screen, with their career stats, season stats, game stats, and fantasy stats below. The user now has easy access to the information they desire (fantasy stats). ● Else, 120stat displays an error message and

	shows no info for the players. Gives a link back to the homepage.
Postconditions	The players’ final information is displayed on the page, and the user has been given the information to compare fantasy value between the players they desire.

Work Breakdown Structure


Level 0
Level 1
Level 2
Level 3
Level 4
Level 5
Level 6
Level 7
120Stat
Concept
Website
Database
Evaluate Technology Options
Define Requirements (User, Content, System, Server)
Define Specific Functionality
Define risks
Develop initial project plan
Website UI
Website Content
Query
Update
Home Page Design
Statistic Page Design
About Page Design
Home Page Content
Statistic Page Content
About Page Content
Parse Query
4 Player Breakdown Design
User Input Textbox
4 Player Statistical Breakdown
How to Use App
Creator Information
1 Player Breakdown Design
Expandability of Single Player Layout
Single Player Statistical Breakdown
Basic Player Information
Statistics Table
Season-long statistics
Game-by-Game Statistics

UC1	**UC1** will involve many of the parts of the WBS. ● Need to design the home page and statistic page ○ Includes the 4 player breakdown design ■ includes the single player breakdown design ● Need to add content to the home page and statistic page ○ Includes setting up the user input textbox ○ Includes 2 player statistical breakdown ■ Includes single player breakdown, including basic player information and the statistics table ● Includes season long and game-by-game stats ● Need to set up the database and back end ○ This means setting up queries ○ This includes parsing queries to return valid outputs
UC2	**UC2** will involve all of the same things as **UC1**, however it will be comparing different positions and thus requiring different types of query returns.
UC3	**UC3** will involve all of the same things as **UC2** but will need to have the functionality to show 3 players' statistics at once. If **UC1** is implemented correctly, the functionality to expand our view to 4 players should already be completed.
Sprint 1	**Sprint 1** involved us laying down the base structure of the project ● Create a vision statement including an exectuve summary, problem statement, elevator summary, and business case. Also includes detailed description of project stakeholders and list of major risks and features ● Create some user scenarios ● Create an initial project schedule
Sprint 2	**Sprint 2** was simply aimed at setting up Django and familiarizing ourselves with the technology
Sprint 3	**Sprint 3** was the first time we had a website with data ● Create the initial Django layout ○ Create a home page ○ Create a statistics page ● Connect with the database ● Learn the API queries ● Use sample queries to hardcode statistics data

Updated Schedule

Sprint	Tasks	Dates
Sprint 1	Complete predetermined sprint 1 deliverables	Sep 25 - Oct 1
Sprint 2	Environment setup complete, tutorials started	Oct 2 - Oct 8
Sprint 3	Get a barebones site up, create example queries from database	Oct 9 - Oct 15
Sprint 4	Complete predetermined sprint 4 deliverables. Get data based on player name. UI design concept. Revise Sprint 1 Deliverables	Oct 16 - Oct 22
Sprint 5	Have the stats be determined by position. Begin UI implementation.	Oct 23 - Oct 29
Sprint 6	Complete predetermined sprint 6 deliverables, Extend functionality to two players	Oct 30 - Nov 5
Sprint 7	Add in fantasy football score weighting and other comparison options	Nov 6 - Nov 12
Sprint 8	Allow user to compare stats of 2-4 players	Nov 13 - Nov 19
Sprint 9	Polish up UI, make more intuitive and inviting	Nov 20 - Nov 26
Release	Make any minor finishing touches necessary	Nov 27 - Nov 29

Contribution Summary

For sprint 4 deliverables, Shane worked on user stories and use cases. George worked on supplement specifications and work breakdown structure. Jose worked on the deployment diagram, status report, and contribution summary. Zach worked on the domain model and updated schedule.

For the past couple of weeks, George and Zach have been working with the nfldb api. They determined that it was the best solution for both the data we needed and the format we needed it in. They ran test queries to understand how nfldb worked. Together they conceptualized and began working on classes that would handle getting and storing the data from this database, so that the front-end can be developed in relative isolation. They supplied Shane and Jose with sample queries for them to familiarize themselves with the data that will be displayed.

Jose and Shane have been focusing on going through Django tutorials and learning more about the front-end side of things. They have a barebones site up, and can display basic information on the page. They have also discussed and drawn out a potential UI, and decided on the site structure.

Status Report

When our group formed, we set up a means to communicate and set up a meeting time that we have followed consistently. We decided to use GitHub as a repository to track changes and bugs. After deciding to develop the site 120stat, we started researching the technologies we would use to build the site. We decided to primarily use the Django framework with nfldb to get player statistics. Since the last set of deliverables, we have continued accomplishing our goals. Everyone has their environment set up, and has done some amount of Django tutorials. The group has a very basic site up, and we can get data from nfldb using python. We have discussed and drawn out the UI for the site, and decided on the site structure. Finally, we have begun writing the classes which will handle getting and storing data for players, and we will soon combine the front end and the back end to display player statistics on the site.