Quikbet

Design & Planning Report

Paul Ballard, Nicholas Brough, Paul Wright

19062819, 18920577, 19359229

Contents

[**Introduction** 2](#_Toc10452091)

[**Competitor Analysis** 3](#_Toc10452092)

[**Squiggle** 3](#_Toc10452093)

[**The Arc** 4](#_Toc10452094)

[**Ladbrokes** 5](#_Toc10452095)

[**AFLalytics** 6](#_Toc10452096)

[**Footy Forecaster** 7](#_Toc10452097)

[**Live AFL Ladder** 8](#_Toc10452098)

[**Our Design** 9](#_Toc10452099)

[**Tech Stack** 9](#_Toc10452100)

[**Squiggle API** 9](#_Toc10452101)

[**Angular** 9](#_Toc10452102)

[**Bootstrap** 9](#_Toc10452103)

[**jQuery** 9](#_Toc10452104)

[**La Trobe** 9](#_Toc10452105)

[**BitBucket/Git** 9](#_Toc10452106)

[**Wireframe** 10](#_Toc10452107)

[**API Examples** 11](#_Toc10452108)

[**“As a fan, I want to see the results of all games my team has played this season”** 11](#_Toc10452109)

[**“As a fan, I want to see the prediction that my team will win their next game”** 11](#_Toc10452110)

[**“As a fan, I want to see opponent and games details for the next 5 matches my team will play”** 11](#_Toc10452111)

[**“As a fan, I want to see all the games and results so far this season for my team”** 12](#_Toc10452112)

[**"As a fan, I want to see all the games (hosting venue) and results so far this season for my team”** 12](#_Toc10452113)

[**“As a fan, I want to see the head-to-head games and if available, results between my team and my team’s rival <..> this season”** 12](#_Toc10452114)

[**"As a fan, I want to see the current ladder"** 13](#_Toc10452115)

[**“As a fan i want to see the predicted ladder”** 15](#_Toc10452116)

[**”As a fan I’d like to see my team’s history regarding W/L”** 16](#_Toc10452117)

[**Final Report** 17](#_Toc10452118)

[**System Design decisions** 17](#_Toc10452119)

[Wireframe 18](#_Toc10452120)

# **Introduction**

Design choices in creating a website are important to providing a user-friendly experience. These choices are based on visual design principles, subject to the norms of the type of website being created, as well as the tech stack it is created with. In this report we will design an AFL predictive modelling/statistics Single Page Application (SPA) called QuikBet focusing on user-friendly design.

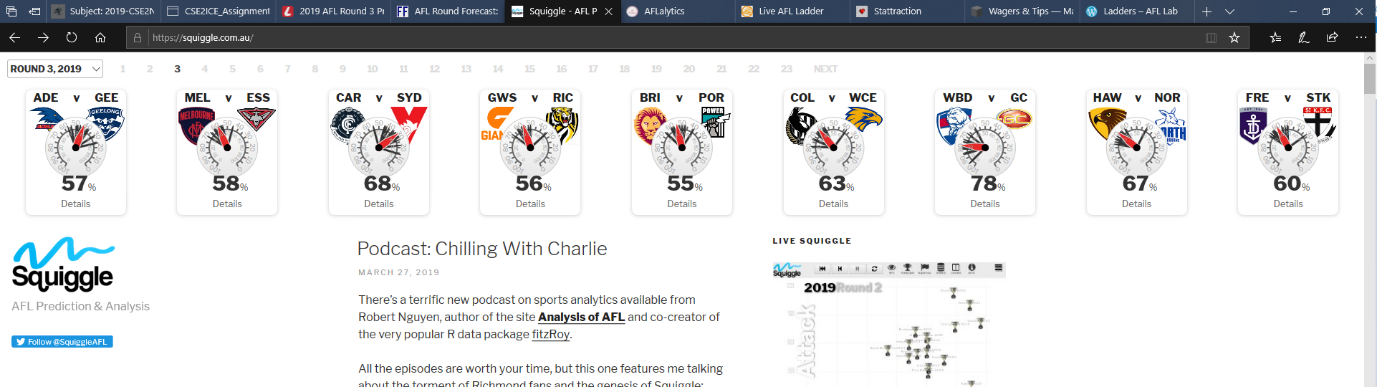
To better inform our design choices we will critique 5 competitor sites, each utilising statistics and predictive models, but each different in design execution for a variety of reasons, e.g. Main purpose of the website, the technical knowledge of the developer(s), etc.

In detailing our design, we will outline the tech stack used for the front and back-end, design a basic wireframe UI for visualisation and provide some API examples in order to demonstrate how the data will be selected from within the Squiggle API.

# **Competitor Analysis**

## **Squiggle**

(<https://squiggle.com.au/>)



Squiggle is a site for match predictions and statistics and is the home of the API we will be basing our website on.

**Design critique:**

Navigation – Relatively simple. Access to all matchup odds is available on the page via the nav bars and other data such as a predictive ladder is accessible in aside elements. The only negative apparent is in the blog-style of the main section requiring a large amount of scrolling

Symmetry – Very vertically consistent, although space between elements only consists of whitespace and can look a little muddled as a result

Alignment – There is a consistent division between the main articles and asides

Predictability – There is a header in the form of nav bars, consistently segmented elements spanning vertically, and questionably a footer containing a search bar. The logo forms part of the header, except in widescreen when it is fixed to the user’s scrolling

Economy of styles – Excluding some embedded apps and team colours the page is very economical in using primarily black text with sparingly accented blues

Proportions – The nav bars aren’t noticeably large or small, the central articles are given more focus than the asides and embedded apps contained all conform to the rectangular sections they are within

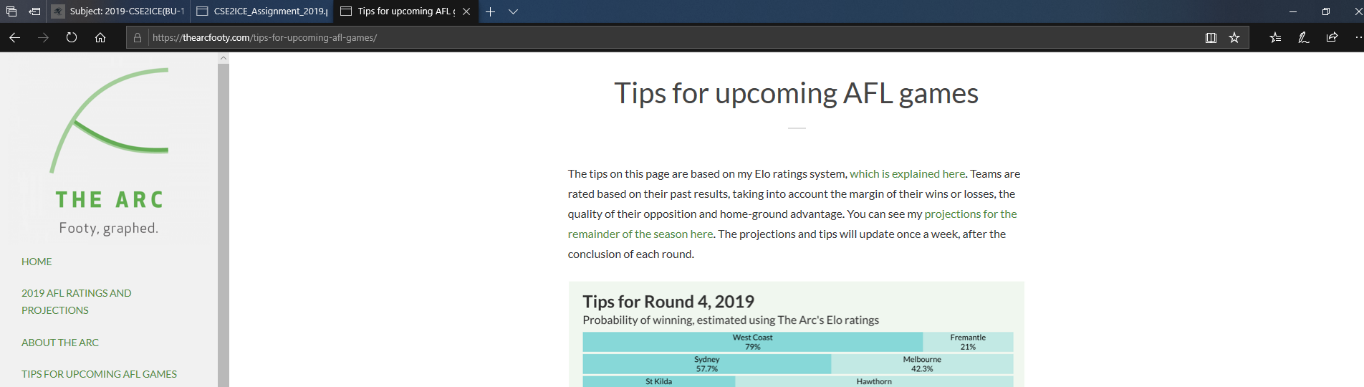
Layout – Head to head matchups are grouped together in the header, other data visualisations are grouped together in the aside elements and the blog style updates are the main articles

**Implications for our design decisions:**

1. We will aim to minimise scrolling. Our website won’t contain blog posts, so it will be less of an issue, nevertheless, we will opt for a design solution that selects specific data to display on screen, instead of large amounts of unfiltered text
2. Our design will use a more prominent banner logo as part of the header
3. Given the purpose of our site being stats focused, we will adopt a similar economy of style. A simple text design seems suitable for a statistics-based page

## **The Arc**

(https://thearcfooty.com/)



The Arc is an AFL projections website focusing on presenting data in various styles of graphs

**Design critique:**

Navigation – The title, navigation pane, keyword search, and social media links are combined in the form of a separately scrollable side bar. The scrolling is a little cumbersome and having general navigation near the top almost makes the scrolling unnecessary anyway.

Symmetry – Clean and consistent.

Alignment – Excluding the side bar, the page is consistently centre-aligned, and all of the elements conform nicely.

Predictability – The Arc follows some design conventions by including the title and navigation in a pane. There is no footer or separate aside element for secondary content and any content that may be suitable is placed in the navigation side bar.

Economy of styles – The Arc uses pale greens to highlight links and backgrounds of elements against the white space. Other colours are used only within data visualisations. The text fonts are basic and easy to read and are consistent except within table data. The overall style is light and not overbearing.

Proportions – Every element is given a reasonable amount of space, titles are easily differentiated

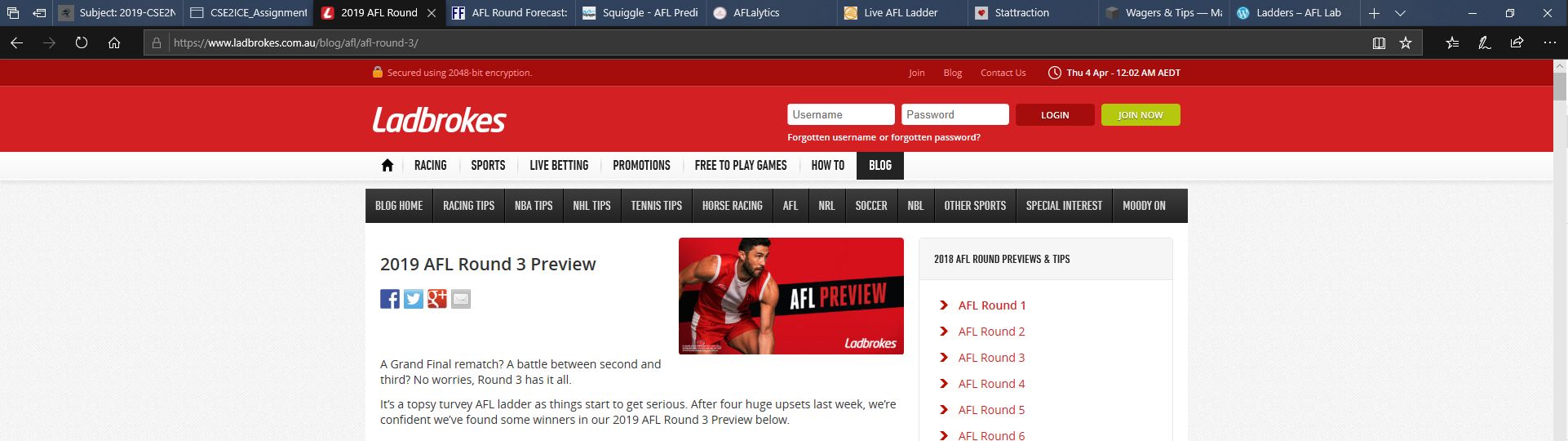
Layout – The layout isn’t complicated. The side bar contains all the navigation. Page content is placed sequentially down the page. Moving some of the content in the sidebar to a footer would likely improve the layout while simplifying the side bar.

**Implications for our design decisions:**

1. We will avoid using two scroll bars in our design.
2. Content will be presented in a predictable manner to limit cognitive load

## **Ladbrokes**

(<https://www.ladbrokes.com.au/blog/afl/>)



Ladbrokes is a sports betting site and as such focuses on functionality beyond sports prediction modelling. It is a service that allows betting on various sports, something that casts a much wider net than the SPA we are designing, however it does include predictions and some statistics for AFL so there is reason to critique it.

**Design critique:**

Navigation – General website navigation is available through the header. The predictive page for AFL can be found from Homepage > Blog > AFL, for instance. Links to each round can be selected from an aside section, and each game has a small write-up followed by a prediction within the main element which can make for a long page. There is also a ‘Betting Review’ page which breaks down the previous week. Getting to each section can be a little tedious, but for the amount of content not unreasonable.

Symmetry – Elements are consistently symmetrical.

Alignment – Elements are aligned uniformly and well defined within borders.

Predictability – Ladbrokes follows many typical website conventions. The header consists of the logo, account login details and a nav bar. Content is contained within well-defined elements and the website is closed by a footer.

Economy of styles – Ladbrokes uses black text on light grey and white backgrounds with red accents. Font is consistent throughout the website, with only the nav bar and logo differing. Overall, quite a common style, neither exceptional nor bad.

Proportions – All elements follow a basic rectangular form. The nav bar is quite compact. Main elements are given more weight than aside sections.

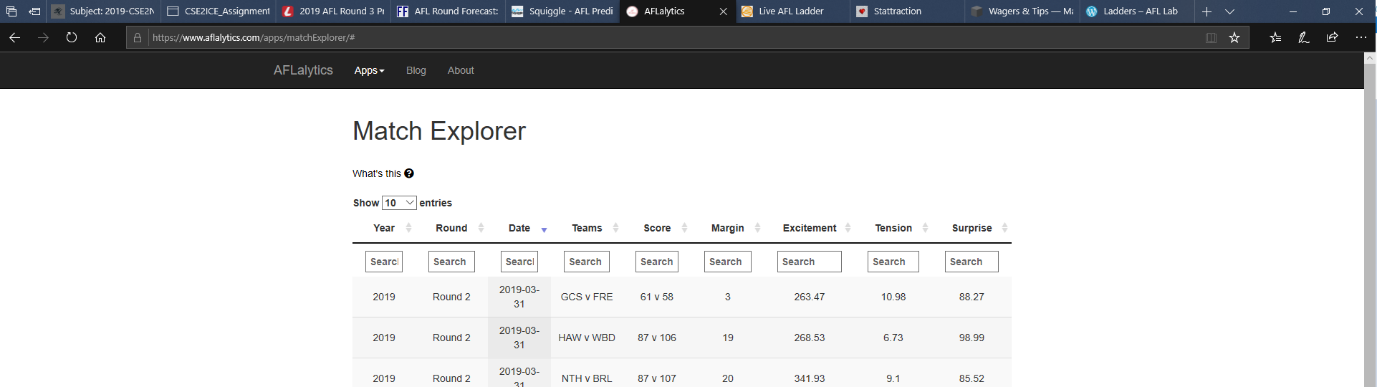
Layout – Easy to follow. Navigation links are grouped together, as are article posts.

**Implications for our design decisions:**

1. As in our critique of Squiggle, our design will avoid excessively long scrolling sections
2. QuikBet will be designed with similar clear regularity and unity of whitespace

## **AFLalytics**

(<https://www.aflalytics.com/apps/seasonForecast/>)



AFLalytics is a simple minimalistic site with AFL data and visualisations being the main focus.

**Design critique:**

Navigation – The entire sites navigation consists of 7 links from within the nav bar. Each page is compact having only a couple of pages requiring any scrolling. For each page containing AFL data, accessing it can also be narrowed through search functions.

Symmetry – The main elements are evenly balanced and clearly defined.

Alignment – Elements are well aligned. The page is centre-aligned and the title/nav bar is inline it.

Predictability – There isn’t much to critique. The only real common design patterns present are the nav bars and footers, so the website is predictable in that sense.

Economy of styles – Literally black and white. Grey backgrounds are used to define and break up white space. Use of an extra colour and

Proportions – This is hard to critique. The design has one focus, and that is the data and how its presented. Data visualisations are well proportioned, figures, text size and spacing are clear

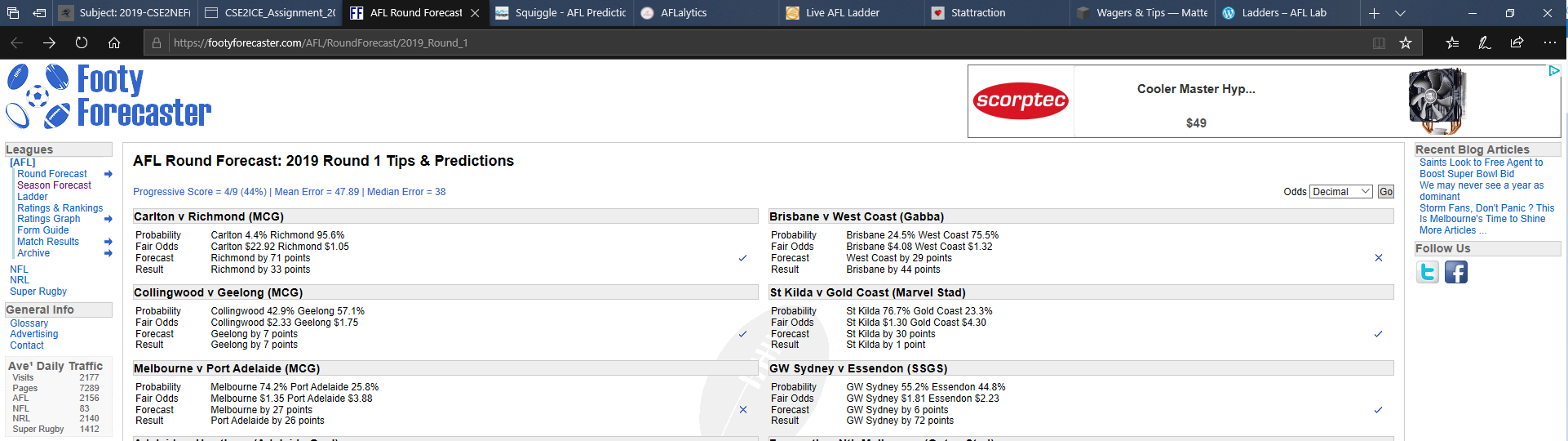
Layout – Again, difficult critique. There is a header, an element or two, and a footer.

**Implications for our design decisions:**

1. We would like to incorporate a similar emphasis on the data being succinct, but not so drastically, through the following:
   1. Our design will opt for sightly more colour than AFLalytics. It is too bland
   2. Instead of having all data selectors inside the data visualisations we will utilise space above as well as sliders on either side to spread out the design

## **Footy Forecaster**

(<https://footyforecaster.com/AFL/RoundForecast/2019_Round_3>)



Footy Forecaster is another example of a simple AFL data analysis website.

**Design critique:**

Navigation – This site has opted to use a sidebar as its main navigational pane. It is very easy to find specific pages through this manner and each page contains one sizeable table of data.

Symmetry – Lines are consistent.

Alignment – Logo is aligned left, banner advertisement is aligned right and besides the sidebars, all else is aligned centrally which is slightly jarring to look at.

Predictability – Besides lacking a top nav bar, the website does follow basic design patterns having a header, side navigation bar, central content and a footer.

Economy of styles – Similar to Squiggle with black text, blue accenting on links however this website has gone the ‘boxy squared borders’ route, where Squiggle opted for borderless with the latter looking cleaner as a result.

Proportions – The sidebars are small, particularly the cramped links within the right section. The scalability of this website leads excessive whitespace within the body elements.

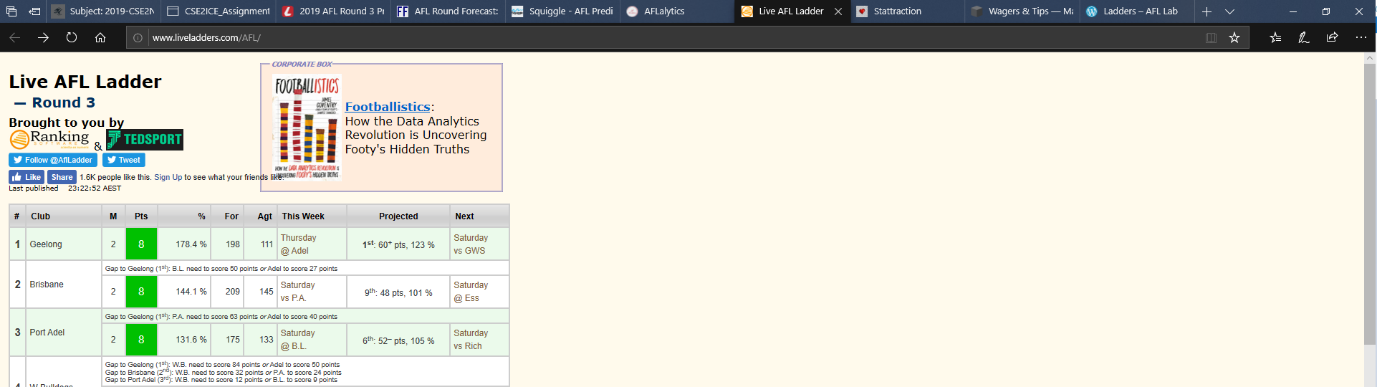
Layout – The logo, sidebars, footer and body are where they should be, however improvements could be made through utilising the header space better or changing the right aside section.

**Implications for our design decisions:**

1. Our website will incorporate a centre alignment throughout the page
2. Elements will be less blocky, using borders more sparingly

## **Live AFL Ladder**

(<http://www.liveladders.com/AFL/>)



Live AFL Ladder is a basic html page containing an AFL predictive ladder.

**Design critique:**

Navigation – There is nothing clickable besides social media links, partners and an advertisement. Navigation in this case is scrolling down the table.

Symmetry – There is symmetry within the table.

Alignment – Everything is aligned left which is an issue on widescreen monitors and is exacerbated by the hard borders used.

Predictability – This page consists of a title, a table and a footer. There are no forms of navigation.

Economy of styles –There are too many fill colours and the usage of one font in this case is bland.

Proportions – The title section is small and unremarkable. The cluttered links and text beneath it are squashed yet overflow into the advertisement. One simple improvement would be to move the “Brought to you by” section to the footer to declutter the layout somewhat.

Layout – The layout is as follows title>table>footer. Nothing major to really change without having more content.

**Implications for our design decisions:**

1. Our design will maintain adequate space between elements
2. We will utilise colours sparingly including whitespace and choose a more modern font style
3. As stated earlier, the page will be centre aligned
4. Various functional elements will be used to both choose data and provide data visualisation to our page, rather than having no interactivity

# **Our Design**

## **Tech Stack**

### **Squiggle API**

Squiggle API contains publicly available raw data for AFL statistics and predictions. QuikBet’s main purpose is to provide a user-friendly Single Page Application (SPA) to access this data.

### **Angular**

AngularJS is a powerful framework which provides dynamic web application functionality. It has been increasingly becoming an industry standard over the past few years and as such has large amounts of documentation. AngularJS is the ideal technology for designing QuikBet’s SPA functionality due not only to its prevalence in web design, but also its cross functionality with bootstrap and jQuery.

### **Bootstrap**

A HTML, CSS and JavaScript framework aimed at providing simple yet powerful web-design. It provides templates for styling the page, text and HTML elements, as well as a customisable Grid System to design a page layout. Bootstrap will be the main technology used to maintain uniform styles and create a simple, responsive layout due to its ability to streamline the design process.

### **jQuery**

jQuery is a library for simplifying coding in JavaScript. Aspects of JS programming such as DOM traversal and manipulation as well as animation, event handling and Ajax are some of the main focuses of the library. We will be using jQuery to simplify the JavaScript code for retrieval and parsing of data from Squiggle API.

### **La Trobe**

Server hosting will be maintained by La Trobe as it is a freely accessible hosting resource for our purposes.

### **BitBucket/Git**

BitBucket provides repository hosting services for development projects. Using git, BitBucket allows for version control of files, a great tool in allowing members of a project team to simultaneously work on development even in a decentralised manner. We will use BitBucket for version control hosting to provide more efficient collaboration.

## My First Document**Wireframe**

This is the wireframe for our site, we have tried to go for a simple yet elegant design. We intend to achieve this by using a simple and clean colour pallet as well as not cluttering the screen with unwanted information and trying to maintain a symmetrical balance across the screen.

We created a banner to sit at the top of the screen with a logo and short slogan that will collapse as you scroll down the page, so as not to use up excess screen real estate. We chose to use two sliders to set the start and end round that you want to return information from, the sliders are situated at different ends of the screen as when we read we start at the left and finish at the right which leads to a more intuitive interaction.

Radio buttons will be used to select the year you want to return data from, you will also be able to select multiple years. The sliders on the sides of the screen will display six teams at a time, when the arrow is clicked it will change all six teams. This makes the page more user friendly as it lessens the number of clicks required significantly, 3 clicks for a full rotation as opposed to 18.

The information returned will be displayed in two ways; if the game has been completed, it will display the points, date, location and the winning team. If the game has not yet been completed then it will display the names of the teams, who is tipped to win, the location of the game and the source the tip came from. If you then click on the box it will expand and have the betting information from the other sources, this allows the user to make a more informed betting decision.

Near the bottom of the page we have added a predicted ladder and a current ladder that will be completed using completed matches and predicted outcomes of further matches, this is added to give the user an idea of where their team might finish for the season. A footer has also been included to make the page look more professional and complete.

## **API Examples**

### **“As a fan, I want to see the results of all games my team has played this season”**

$.getJSON("https://api.squiggle.com.au/?q=games;format=json", function(data){

myTeam = "Collingwood";

myYear = 2019;

complete = 100;

for(i = 0; i < data.games.length; i++)

{

var victor = "";

if(data.games[i].hteam == myTeam && data.games[i].year == myYear && data.games[i].complete == complete)

{

var victor = ((data.games[i].winner == myTeam) ? 'won' : 'lossed');

console.log(myTeam + " played a home game at " + data.games[i].venue + " versing " + data.games[i].ateam + " in round " + data.games[i].round + " and " + victor);

}

else if(data.games[i].ateam == myTeam && data.games[i].year == myYear && data.games[i].complete == complete)

{

var victor = ((data.games[i].winner == myTeam) ? 'won' : 'lossed');

console.log(myTeam + " played a home game at " + data.games[i].venue + " versing " + data.games[i].ateam + " in round " + data.games[i].round + " and " + victor);

}

}

});

### **“As a fan, I want to see the prediction that my team will win their next game”**

$.getJSON("https://api.squiggle.com.au/?q=tips;format=json", function(data){

myTeam = "Collingwood";

year = 2019;

month = 4;

day = 3;

mySource = "Matter of Stats";

found = false

for(i = 0; i < data.tips.length && !found; i++)

{

var gameYear = parseInt(data.tips[i].date.substring(0,4));

var gameMonth = parseInt(data.tips[i].date.substring(5,7));

var gameDay = parseInt(data.tips[i].date.substring(8,10));

var futureGame = gameYear >= year && gameMonth >= month && gameDay >= day;

if(data.tips[i].hteam == myTeam && futureGame && data.tips[i].source == mySource)

{

console.log("Prediction that " + myTeam + " will win their next game is: " + data.tips[i].hconfidence);

found = true;

}

else if( data.tips[i].ateam == myTeam && futureGame && data.tips[i].source == mySource)

{

console.log("Prediction that " + myTeam + " will win their next game is: " + data.tips[i].aconfidence);

found = true;

}

}

});

### **“As a fan, I want to see opponent and games details for the next 5 matches my team will play”**

$.getJSON("https://api.squiggle.com.au/?q=games;format=json", function(data){

currentRound = 1;

roundlimit = currentRound + 5;

myTeam = "Collingwood";

myYear = 2019;

complete = 100;

for(i = 0; i < data.games.length; i++)

{

if(data.games[i].hteam == myTeam && data.games[i].year == myYear && data.games[i].round < roundlimit)

{

console.log(myTeam + " Is playing a home game at " + data.games[i].venue + " versing " + data.games[i].ateam + " in round " + data.games[i].round);

}

else if(data.games[i].ateam == myTeam && data.games[i].year == myYear && data.games[i].round < roundlimit)

{

console.log(myTeam + " Is playing an away game at " + data.games[i].venue + " versing " + data.games[i].hteam + " in round " + data.games[i].round);

}

}

});

### **“As a fan, I want to see all the games and results so far this season for my team”**

$.getJSON("https://api.squiggle.com.au/?q=games;format=json", function(data){

currentRound = 1;

roundlimit = currentRound + 5;

myTeam = "Collingwood";

myYear = 2019;

complete = 100;

for(i = 0; i < data.games.length; i++)

{

if((data.games[i].hteam == myTeam || data.games[i].ateam == myTeam) && data.games[i].year == myYear && data.games[i].complete == complete)

{

console.log(data.games[i]);

}

}

});

### **"As a fan, I want to see all the games (hosting venue) and results so far this season for my team”**

$.getJSON("https://api.squiggle.com.au/?q=games;format=json", function(data){

myTeam = "Collingwood";

myYear = 2019;

myVenue = "M.C.G."

for(i = 0; i < data.games.length; i++)

{

if(data.games[i].year == myYear && (data.games[i].hteam == myTeam || data.games[i].ateam == myTeam) && data.games[i].complete == 100 && data.games[i].venue == myVenue)

{

console.log(data.games[i]);

}

}

});

### **“As a fan, I want to see the head-to-head games and if available, results between my team and my team’s rival <..> this season”**

$.getJSON("https://api.squiggle.com.au/?q=games;format=json", function(data){

myTeam = "Collingwood";

myYear = 2019;

for(i = 0; i < data.games.length; i++)

{

if(data.games[i].year == myYear && data.games[i].hteam == myTeam)

{

if(data.games[i].complete == 0)

{

console.log(myTeam + " is playing a home game against " + data.games[i].ateam);

}

else

{

console.log(myTeam + " has played a home game against " + data.games[i].ateam);

// Insert desired game data as the game is completed.

}

}

else if (data.games[i].year == myYear && data.games[i].hteam == myTeam)

{

if(data.games[i].complete == 0)

{

console.log(myTeam + " is playing an away game against " + data.games[i].hteam);

}

else

{

console.log(myTeam + " has played an away game against " + data.games[i].hteam);

// Insert desired game data as the game is completed.

}

}

}

});

### **"As a fan, I want to see the current ladder"**

var teamNames = new Map([

["Adelaide", 1],

["Brisbane Lions", 2],

["Carlton", 3,],

["Collingwood", 4],

["Essendon", 5],

["Fremantle", 6],

["Geelong", 7],

["Gold Coast", 8],

["Greater Western Sydney", 9],

["Hawthorn", 10],

["Melbourne",11],

["North Melbourne", 12],

["Port Adelaide", 13],

["Richmond", 14],

["St Kilda", 15],

["Sydney", 16],

["West Coast", 17],

["Western Bulldogs", 18],

]);

var teams = new Array();

var team = {

name: "",

points: 0,

for: 0,

ag: 0,

percentage: 0,

};

//setup array

for (var [key, value] of teamNames) {

teams.push({name: key, points: 0, for: 0, against: 0, percentage: 0})

}

function searchTeam (teamName)

{

return teamNames.get(teamName)-1;

}

var hTeam = "";

var aTeam = "";

$.getJSON("https://api.squiggle.com.au/?q=games;format=json", function(data){

myYear = 2019;

complete = 100;

console.log("Scores \n \n \n ");

for(i = 0; i < data.games.length; i++)

{

if(data.games[i].year == myYear && data.games[i].complete == complete)

{

hTeam = data.games[i].hteam;

aTeam = data.games[i].ateam;

aTeamFor = data.games[i].ascore;

aTeamAgainst = data.games[i].hscore;

hTeamFor = data.games[i].hscore;

hTeamAgainst = data.games[i].ascore;

if (aTeamFor == hTeamFor)

{

teams[searchTeam(hTeam)].points += 2;

teams[searchTeam(aTeam)].points += 2;

}

else if (aTeamFor > hTeamFor)

{

teams[searchTeam(aTeam)].points += 4;

}

else{

teams[searchTeam(hTeam)].points += 4;

}

teams[searchTeam(hTeam)].for += data.games[i].hscore;

teams[searchTeam(hTeam)].against += data.games[i].ascore;

teams[searchTeam(aTeam)].for += data.games[i].ascore;

teams[searchTeam(aTeam)].against += data.games[i].hscore;

}

}

// calculate percentage

for (var i=0; i <= 17; i++)

{

teams[i].percentage = (teams[i].for / teams[i].against) \* 100;

}

for (var i=0; i <= 17; i++)

{

console.log(teams[i].name);

console.log(teams[i].points);

console.log(teams[i].for);

console.log(teams[i].against);

console.log(teams[i].percentage);

}

sortedTeams = teams.sort(function(a, b) {

if(a.points == b.points)

{

return a.percentage - b.percentage;

}

else

{

return a.points - b.points

}

});

for (var i=0; i <= 17; i++)

{

console.log(sortedTeams[i].name);

console.log(sortedTeams[i].points);

console.log(sortedTeams[i].for);

console.log(sortedTeams[i].against);

console.log(sortedTeams[i].percentage);

}

});

### **“As a fan i want to see the predicted ladder”**

\*Note: Predicted ladder won’t function correctly until the current ladder is implemented

$.getJSON("https://api.squiggle.com.au/?q=tips;format=json", function(data){

var year = 2019;

var month = 4;

var day = 14;

var gameYear = 0;

var gameMonth = 0;

var gameDay = 0;

var futureGame = false;

var mySource = "Matter of Stats";

function searchTeam (teamName)

{

return teamNames.get(teamName)-1;

}

var predictedWinner = "";

for(var i = 0; i < data.tips.length; i++)

{

gameYear = parseInt(data.tips[i].date.substring(0,4));

gameMonth = parseInt(data.tips[i].date.substring(5,7));

gameDay = parseInt(data.tips[i].date.substring(8,10));

futureGame = gameYear >= year && gameMonth >= month && gameDay >= day;

if(data.tips[i].source == mySource && futureGame)

{

teams[searchTeam(data.tips[i].tip)].points += 4;

}

}

predictedTeams = teams.sort(function(a, b) {

if(a.points == b.points)

{

return a.percentage - b.percentage;

}

else

{

return a.points - b.points

}

});

for (var i=0; i <= 17; i++)

{

console.log(predictedTeams[i].name);

console.log(predictedTeams[i].points);

console.log(sortedTeams[i].for);

console.log(sortedTeams[i].against);

console.log(predictedTeams[i].percentage);

}

});

### **”As a fan I’d like to see my team’s history regarding W/L”**

$.getJSON("https://api.squiggle.com.au/?q=games;format=json", function(data){

myTeam = "Collingwood";

myYear = 2019;

var winstreak = "streak";

for(i = 0; i < data.games.length; i++)

{

if(data.games[i].year == myYear && (data.games[i].hteam == myTeam || data.games[i].ateam == myTeam) && data.games[i].complete == 100)

{

if(data.games[i].winner == myTeam)

{

winstreak += "/ W ";

}

else{

winstreak += "/ L ";

}

}

}

console.log(winstreak);

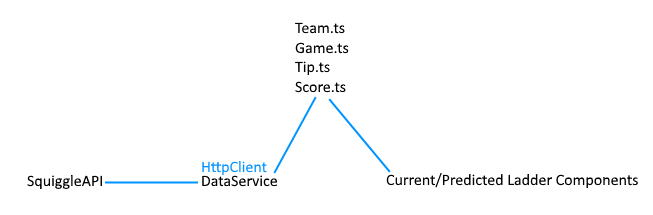
});

# **Final Report**

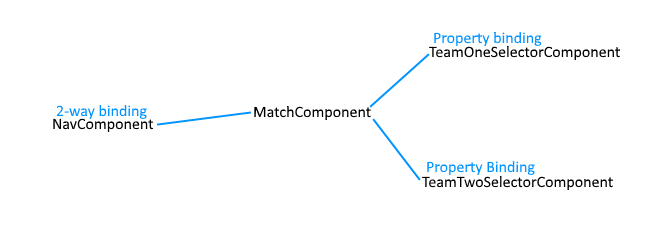
## **System Design decisions**

The tech stack for the front end development of QwikScore consists of the Angular framework for creating a structured web application, as well as Bootstrap and CSS for styling. Initially in part A we opted to develop our SPA(single page application) with Bootstrap, jQuery and Angular as the primary library and frameworks. In practice we never found ourselves using jQuery. The usefulness of Angular through its component system, TypeScript, ngModel and event binding provided significantly more functionality than jQuery could for the amount of effort and as such it became the core framework of our design.

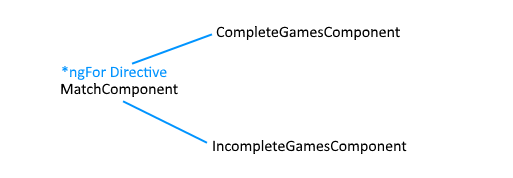
Our page design consists of several app-components. Besides the header/footer elements, each other app-component requires data to be parsed from the Squiggle API.

The current and predicated ladders receive their data from the DataService service which uses HTTPClient get methods to pull data from the SquiggleAPI. 

TeamSelector components are two carousels which both properties bind a team on click. The NavComponent utilises 2-way binding in selecting a range of rounds, a year and then OnChange requests data from the API with a range of rounds and the year to pull from the API.



This data is passed into the Match Component, with which it executes an ngFor loop directive displaying as either instances of the CompleteGamesComponent or the IncompleteGameComponent as in the wireframe



## Wireframe

Our initial design intention was to have a relatively simple web page with various visual tools to select data and have it displayed centrally. The first wireframe for QuikBet aimed to do so in a compact manner with minimal page redirection. The design has largely stayed the same. However, some design decisions and functionality were changed to help keep the page layout compact and make elements of qwikScore more accessible to the user. These changes will be outlined below.

The decision was made to rebrand this page to QwikScore as the focus of this specific page isn’t betting so much as informing the user of past and predictive scores.

In our original wireframe we planned for a banner that would collapse with scrolling but ultimately decided that floating a banner wasn’t necessary.

The team selection sliders have been changed from a vertical carousel to horizontal because we found it to be a faster method of cycling through teams than we had previously.

We made minor changes to the main central elements, reformatting the unplayed games to feature images of both teams as opposed to just the team tipped to win.

Considering the wide amount of data selectable at once, the decision was made to have all match data components be collapsible to prevent excessive scrolling for the user.