

FOOTBALL MATCH PREDICTION WEB APPLICATION - SURETHING

MARINA SHCHUKINA



A REPORT SUBMITTED AS PART OF THE REQUIREMENTS
FOR THE DEGREE OF COMPUTER SCIENCE
AT THE SCHOOL OF COMPUTING SCIENCE AND DIGITAL MEDIA
ROBERT GORDON UNIVERSITY, ABERDEEN, SCOTLAND

April 2015

Supervisor Dr. Robert McDermott, Dr. Richard Glassey

Abstract

The dissertation will describe the process of building a football betting simulator that uses win/loss points instead of money - SureThing. It is a web application where users can make their own match result predictions and place bets on real football matches in the English Premier League. The aim of the simulator is to show the influence of different factors on the match result and to encourage users to make more informed betting decisions.

All the objectives specified in the report were met and the project was successful.

Acknowledgements

I would like to acknowledge and extend my gratitude to the following people who have made the completion of this project possible:

- Dr. Roger McDermott for his support and guidance in this project
- Dr. Richard Glassey for his initial help and valuable advice
- My husband Murray and baby daughter Scarlett for their support and patience.

Declaration

I confirm that the work contained in this BSc (Hons) project report has been composed solely by myself and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged and all verbatim extracts are distinguished by quotation marks.

Signed Date

Marina Shchukina

Contents

Abstract	ii
Acknowledgements	iii
Declaration	iv
1 Introduction	1
1.1 Background	1
1.2 Proposed Web Application	3
1.3 Target Audience	3
1.4 Problems Addressed	3
1.5 Project Aims & Objectives	4
1.6 Report Structure	4
2 Legal, Social and Ethical Issues	6
2.1 Legal Issues	6
2.2 Social Issues	7
2.3 Ethical Issues	7
3 Requirement Analysis and Specification	8
3.1 Target Audience Questionnaire	8
3.2 Researching Current Solutions	12
3.2.1 Sports news websites	12
3.2.1.1 BBC Sport Football	13
3.2.2 Football statistics websites	14
3.2.2.1 WhoScored	14
3.2.2.2 Squawka	15
3.2.3 Bookmakers Websites	16
3.2.3.1 Bet365	16
3.2.4 Communities for sports fans	16

3.2.5	OLGB Betting Community	17
3.2.6	Black-box prediction applications	18
3.2.7	Conclusion	19
3.3	Requirements Specification	19
3.3.1	Definitions	20
3.3.2	Functional Requirements	21
3.3.2.1	Authentication and User Profile	21
3.3.2.2	Matches Overview	22
3.3.2.3	Prediction	22
3.3.2.4	Upcoming Match View	24
3.3.2.5	Played Match View	25
3.3.2.6	Dashboard	25
3.3.2.7	Notifications	26
3.3.2.8	Leaderboard	27
3.3.2.9	Optional requirements	27
3.3.3	Non-functional Requirements	28
3.4	Overall Architecture	28
3.5	Choice of Third Party API	29
3.6	Project Plan	30
4	Application Design	31
4.1	Application Structure	32
4.2	Branding and Visual Design	32
4.3	Analysis of the competitors experience	33
4.4	Wireframes	33
4.5	User Stories	39
4.6	Activity Diagrams	39
4.7	Database Schema	41
5	Implementation	43
5.1	Choice of Technologies	43
5.1.1	Front End	43
5.1.2	Back End	45
5.2	Application Architecture	46
5.3	Patterns And Conventions	47
5.3.1	Application Factory	48
5.3.2	Blueprints	48

5.3.3	Database Migrations	48
5.3.4	Exceptions	49
5.4	Other Implementation Processes	50
5.4.1	PEP8	50
5.4.2	PyLint	50
5.4.3	Version Control	50
5.4.4	Own Validation in Forms	51
5.4.5	Custom Macros	52
5.4.6	Integration with third-party API	52
5.4.7	Visual Effects	54
5.4.8	Responsive Design	56
5.5	Features Implementation	57
5.5.1	Authentication and User Profile	57
5.5.2	Matches Overview	61
5.5.3	Upcoming Match View	64
5.5.4	Prediction	68
5.5.5	Played Match View	72
5.5.6	Dashboard	74
5.5.7	Notifications	76
5.5.8	Leaderboard	77
5.6	Application Performance	78
5.7	Deploying the Application	78
6	Testing	79
6.1	Unit Testing	79
6.2	Continuous Integration with Travis CI	80
6.3	Use case Testing	81
6.3.1	Test 1 – Use Case 1 Testing	81
6.4	User Acceptance Testing	81
7	Conclusion	84
7.1	Evaluation	84
7.2	Personal Statement	85
7.3	Improvements and Future Work	86
A	Target Audience Questionnaire	88
A.1	Questions	88
A.2	Answers	90

B User Acceptance Questionnaire	93
B.1 Questions	93
B.2 Answers	94
C Installation Instructions	98
D Credits	99
E Project Management	100
F Presentation Slides	101

List of Tables

A.1	Table illustrating answers to questions 1-3 in the Target Audience Questionnaire . . .	91
A.2	Table illustrating answers to questions 4-7 in the Target Audience Questionnaire . . .	92
B.1	Table illustrating answers to questions 1-4 in the User Acceptance Questionnaire . . .	95
B.2	Table illustrating answers to questions 5-7 in the User Acceptance Questionnaire . . .	97

List of Figures

3.1	A pie chart illustrating the answers of the questionnaire respondents when asked how many times a week do they bet.	9
3.2	A pie chart displaying the answers of the respondents when asked how many sources of information (websites, newspapers, mobile apps) do they check before placing a bet.	9
3.3	A bar chart illustrating the answers of the respondents when asked to specify the sources of information that they use to support a betting decision.	10
3.4	A bar chart illustrating the answers of the respondents when asked to specify the factors they consider before placing a bet.	11
3.5	BBC Sport - Football section	14
3.6	Squawka	15
4.1	Mind map capturing the results of initial brainstorming on the application structure and navigation scenarios.	32
4.2	A wireframe of the main page of the website.	34
4.3	A wireframe of the dashboard of the website, displaying the overview of the saved matches. .	35
4.4	A wireframe of the dashboard of the website, displaying the prediction settings form. .	36
4.5	A wireframe displaying an example of an upcoming match.	37
4.6	A wireframe displaying the stats of a played match.	38
4.7	User Sign Up activity diagram.	40
4.8	Edit User Details activity diagram.	40
4.9	Bet on Match activity diagram.	41
4.10	A final diagram representing tables in the database.	42
5.1	A screenshot from the terminal output running Grunt.	44
5.2	Dashboard view function.	47
5.3	Function responsible for creating the application instance.	48
5.4	Blueprints registration.	48
5.5	GitHub allows developers to add custom labels.	50
5.6	GitHub. Issues related to the functionality of the application.	51

5.7	Validator function that checks if user with this username has already been registered with the SureThing application	52
5.8	Football API wrapper, fields	53
5.9	An example of an alert on the page appearing after user has logged out	54
5.10	The same alert fading out.	54
5.11	You have new messages. Desktop View.	55
5.12	You have new messages. Mobile View.	55
5.13	The user just read the last new message, the icon turns grey.	55
5.14	Dashboard view, DVGA screen size.	56
5.15	Module League Position in the Upcoming Match View, DVGA screen size.	57
5.16	SureThing offers registration form for the new users. However, the email address is expected to be unique.	58
5.17	User settings and the profile page can be accessed by clicking on the avatar icon located on the application navigation menu panel.	59
5.18	An example of a dummy avatar for user gannet , who is not registered with Gravatar services.	59
5.19	Users can change their passwords for security reasons.	60
5.20	After registration, users can share more information about themselves by editing their profile.	60
5.21	User profile page.	61
5.22	Matches in the overview are grouped by dates sorted in ascending order.	62
5.23	Example of an unplayed match displayed in the overview.	62
5.24	Example of a played match displayed in the overview.	63
5.25	Example of a live match displayed in the overview.	63
5.26	Upcoming Match View in the "read-only" mode: match header and the first prediction module, Module League Position.	65
5.27	Upcoming Match View in the "prediction" mode: match header and the first prediction module, Module League Position.	66
5.28	Module User Hunch.	67
5.29	Match result prediction.	67
5.30	An example of a prediction module in the Match Preview, user navigated from the main page.	68
5.31	Prediction values can be found on the panels for each module.	69
5.32	The calculation behind the prediction value for the League Position module.	70
5.33	The calculation behind the prediction value for the Form module.	70
5.34	The calculation behind the prediction value for the Home/Away module.	71
5.35	The table illustrates the calculation behind the prediction of the match result.	71

5.36 Played Match View, match header.	72
5.37 Played Match View, users' prediction statistics.	73
5.38 Prediction breakdown.	73
5.39 Played Match View, feedback provided for authenticated users.	73
5.40 Dashboard view.	74
5.41 An example of a Dashboard View with matches that have been committed and played.	75
5.42 Dashboard with no saved matches.	75
5.43 Dashboard menu.	76
5.44 An example of a message sent to the user.	76
5.45 Navigation menu panel with an inbox icon.	77
5.46 User inbox.	77
5.47 Leaderboard.	78
6.1 A screenshot of the terminal output showing the successfully passed test suite of the SureThing.	80
6.2 Travis CI configuration file.	80
6.3 An extract from README on GitHub. Travis status icon indicates that the last build passed.	81

Contents

Chapter 1

Introduction

1.1 Background

Gambling is surely one of the oldest forms of entertainment known to man. Wherever you may find a situation or event where the outcome is in doubt, you will find someone willing to bet on it. References to gambling go back to the beginning of recorded history and in nearly all cultures throughout the world. It is also frowned upon in many different religions and cultures but rarely banned outright, it has been around for a very long time and does not look like it will be disappearing any time soon. One of the oldest forms of gambling is betting on sporting events. The Romans bet on the outcome of chariot races and gladiator fights, the Greeks on the olympics, all the way through to today where you can not only bet on the outcome of every sporting event anywhere in the world but also, thanks in large part to the rise of internet gambling and live updates of events as they happen, you can bet on any aspect of the event - e.g. in the case of football matches you can bet on number of goals scored, who scored them and even the number of corners in a match - the possibilities are almost endless.

Online gambling really began to take off in 1994 when Antigua and Barbuda passed the Free Trade and Processing Act that entitled this Caribbean nation to grant licenses to those willing to open online casinos. The next industry development milestone was the formation of the Kahnawake Gaming Commission in 1996 in Canada. The commission issued gaming licenses to many online casinos and poker rooms [51].

Another important event contributing to the beginnings of the new industry was the creation of a software development company, Microgaming Software Systems, that built the first piece of functional software to be used in the online gaming sites. The company claims they released the first online casino in 1994. It is hard to verify that fact, but they were definitely one of the first. The technology behind their casino was a success and the company concentrated on developing various casino games

and supplying them to other online casinos [33] [15]. Since the mid 1990's the industry began to flourish. For example, there were only 15 online casinos in 1996. This number reached 200 sites by the end of 1997 [51].

Online sports betting industry took off shortly after the early online casinos and grew exceptionally quickly. Placing bets on sporting events online soon became very popular with gamblers all over the world. The existing bookmakers opened websites and many new companies joined the newly formed market [15]. Amongst the types of online sports gambling, betting on football is the most popular, leading the industry with a share about 70% with roughly half a million people betting weekly all over the world [25]. The popularity of betting on football, especially the so-called "in-play" betting (a type of betting when punters place bets as the event happens), was fueled by the increased coverage of the football matches by popular satellite television channels. According to Keogh and Rose [25], the popularity of football gambling skyrocketed before the 1998 World Cup and Euro 2000.

When it comes to any sports betting, football betting included, the player is trying to predict the result of the event and placing money on the outcome. This prediction can be made based on a "hunch" or by using logic and domain knowledge, in a lot of cases by both combined. This gave rise to a variety of betting software systems that attempt to predict the next match result. Most of the time these betting systems work as a "black box" not allowing the user to influence the prediction output and preventing the user from understanding the exact logic used inside the system.

People gambling on football can have various strategies when making a betting decision. As mentioned above, the user can buy prediction software and simply follow the tips suggested by that system. Another option is to make a decision influenced by the opinion of the other tipsters, experts' opinions or rumours. However, most experienced gamblers would likely ignore the tips and predictions of others and analyse the pure statistics themselves before trying to make their own prediction. According to Maguire [29], a contributor to the Encyclopedia Britannica, "One of the most popular forms of gambling is wagering on sports, which taps into the passion of sports fans. A bet placed on a race or a game allows fans to prove their knowledge of a sport or to show their loyalty to a particular team or competitor". In other words, it's not only about the money but also as a way to prove that your own judgement is superior to others. To make this happen, the gambler has to aggregate numerous pieces of information from various sources. This action has to be repeated for every single match.

From my experience, the necessity to repeat an action many times could lead to a creation of interesting software solutions. That is how I got inspired to create an application that would aggregate this information for the user and therefore act as an interactive decision supporting system.

1.2 Proposed Web Application

A web application aiming to forecast football match results is nothing new. However, most of the existing solutions either overwhelm their users with unstructured statistics or work as a self-contained black-box prediction tool preventing the users from participating in the process of prediction. The proposed system will provide its users with all the necessary information to support their betting decision and allow them to take full control over the prediction output. To the best of my knowledge no other software offers the same experience as the one proposed in this dissertation.

1.3 Target Audience

The intended users of the application will be people with an interest in football, more specifically in betting on football. With this in mind, the age range of potential users will be 18 and up.

1.4 Problems Addressed

Problem 1 - Before placing a bet, football punters have to aggregate different pieces of football statistics from various sources.

Most football punters will spend a lot of time gathering and analysing information about a football match. This can be anything from watching the previous games on television, reading expert analysis in newspapers to looking at specialised websites that show every possible statistic about a team or player. The application proposed in this dissertation will do this job for the user. It will aggregate all the data that may affect the outcome of an upcoming match and also make a comparison of the participating teams.

Among other benefits, the easy access to relevant data will hopefully encourage the application users to make betting decisions based on facts rather than an impulse or other punters' advice.

Problem 2 - Punters would like to be able to make their own prediction formula.

When betting on football, there is always a temptation to base a betting decision on a opinion of someone else or use prediction software and there are those who make a living off selling tips to punters. However, as mentioned previously, a serious sports gambler wants to be able to use their own knowledge and judgement to win, rather than rely on a third party "tipster".

The choice of factors that can influence a football match result can differ greatly for each punter. One may think that a team's form is the most important factor going into a match, while the next may think that the team's league position is the best judge of how they are going to do. The users of the proposed application will be able to create their own betting system by picking from the various factors and assigning them a weighting percentage, having the power to influence the prediction output.

Problem 3 - Football punters rarely monitor their betting results.

Learning from the past mistakes is a key to becoming successful in any activity. The application will offer a way to monitor the performance of user's betting system in order to be able to improve it over time.

1.5 Project Aims & Objectives

The overall aim of this project is to build a betting simulator that provides an effective solution to the above problems and enables its users to create their own, hopefully successful, betting system in the long run.

1.6 Report Structure

This report will describe the research, design and implementation of the application, as well as justify the decisions made throughout the project. The structure of the report is outlined below.

- Chapter 1 *Introduction*
- Chapter 2 *Background studies and Objectives* addresses the legal, social and ethical issues.
- Chapter 3 *Requirements Analysis* describes the processes occurring during the requirement analysis phase, such as target audience research and analysis of the existing solutions. The chapter also contains the final list of project mandatory and optional requirements.
- Chapter 4 *Application Prototype* describes the process of designing the application.
- Chapter 5 *Implementation* describes the process of building the application.
- Chapter 6 *Testing & Evaluation* evaluates the overall project and provides results of tests carried out.

- Chapter 7 *Conclusion*

Chapter 2

Legal, Social and Ethical Issues

In this chapter the legal, social and ethical issues relating to the application are addressed.

2.1 Legal Issues

There are a number of legal considerations to take into account when working on an application and also once it is a finished product to be used by the public.

The application will store some basic user data in its database. Therefore, the application must take all the necessary precautions to protect the stored data and sensitive information. The application will not disclose the personal data of its users to any third parties.

The final product must adhere to all the relevant UK legislation, this would include the Data Protection Act of 1998, which has in it many relevant laws pertaining to online products. In particular if the website was ever to make a profit either by advertising or sales or any other way, it would need to be registered with the appropriate government department. As it stands, it does not because it makes no money, but that would have to be looked into if the application was going to be developed along those lines. Also the usual anti discriminatory laws would have to be followed, meaning it cannot be biased against users because of their sex, nationality, colour or disabilities.

Less a legal and more a professional consideration is that the application should not try to mislead a user in any way, anything it can do should not be exaggerated, in this particular application there is nothing on the site saying that the predictions are going to be 100

2.2 Social Issues

There are several advantages of using the application for serious and not so serious gamblers alike. The main goal of using the application is to maximise the amount of correctly predicted results. Using and refining the prediction system should lead to more informed bets and hopefully remove any rash or less thought out bets that sometimes plague a gambler. Secondly, the use of the application will save what can be a large amount of time spent on gathering information before making a betting decision.

On the other hand, the application will hold some interest even for non gambling football fans. Every football fan has an opinion on who is going to win a match and more importantly why they are going to win, this application gives them a chance to prove it. By including a league table showing how the users rate against each other and in the future having more community focussed features mean that the application will offer something other than just a gambling tool.

2.3 Ethical Issues

There are a few ethical matters to consider when coming up with any new application, or indeed any new venture at all. For example, using illegal or pirated software, abusing open source code without proper accreditation or reverse engineering a piece of code to find out how it works then using that to improve your own software.

On this project there was no illegal software used and no reverse engineering done. Open source code that had a GPL (General Public License) was used. A GPL describes how an author would like the code to be distributed, copied or modified and was followed on this project correctly as well as all credit given where due.

There are ethical issues outside of the area of software development to consider as well, namely the ones relating to gambling in general. Gambling is not illegal but nevertheless it does have its share of controversy and is frowned upon in many circles. Gambling addiction is also a very big problem for a lot of people and feeding that addiction would be ethically wrong. However, this application does not require you to bet at all and does not link to any gambling websites. In general, the application is not aimed at promoting gambling. A user can get plenty of enjoyment and use out of it without ever risking any money. Moreover, it supports a more sensible and measured approach to football betting and by tracking a user's performance can show whether or not a person should be gambling in the first place.

Chapter 3

Requirement Analysis and Specification

Before designing a piece of software, it is important to have a good understanding of what the final product is supposed to do. During the requirements analysis phase, a number of steps were taken before assembling a list of requirements. First of all, a target audience questionnaire was produced and sent out to the potential application users. The results of the survey were then compiled and analysed. Secondly, a number of existing websites that could at least partly solve the problem outlined in the "Problem Statement" [1.4](#) were examined and evaluated. Finally, a list of functional and non-functional requirements covering all the aspects of the future web application was developed and documented in this report.

3.1 Target Audience Questionnaire

The target audience research aims to gather information on the way football punters make their betting decisions. Questions were specifically tailored to find out what kind of content would appeal to the potential users of the application [1.3](#). Due to the spread of target users, an online questionnaire was used to collect the answers. There were only 9 respondents due to the specificity of the topic. A full breakdown of the questions asked and the answers received can be found in the appendices [A](#).

How many times a week do you bet on football or other sporting events?

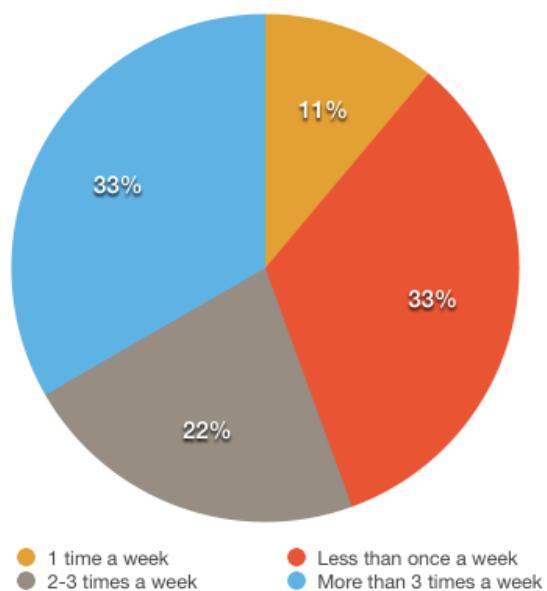


Figure 3.1: A pie chart illustrating the answers of the questionnaire respondents when asked how many times a week do they bet.

As it can be seen from figure 3.1, most of the survey participants are active punters, with the number of placed bets 2 or more per week.

How many sources of information (websites, newspapers, mobile apps) do you check before placing your bet?

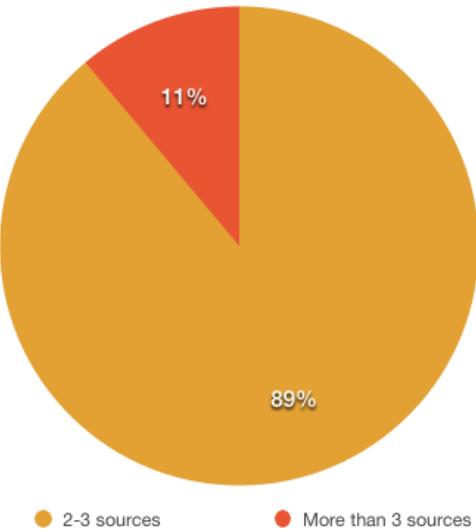


Figure 3.2: A pie chart displaying the answers of the respondents when asked how many sources of information (websites, newspapers, mobile apps) do they check before placing a bet.

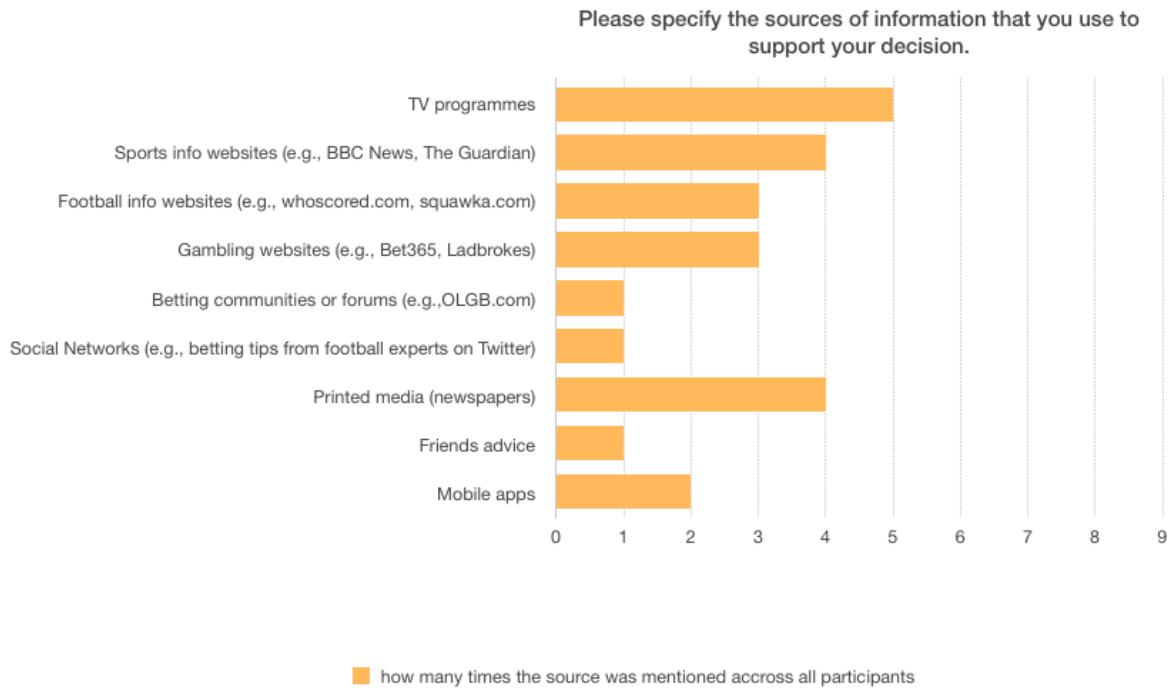


Figure 3.3: A bar chart illustrating the answers of the respondents when asked to specify the sources of information that they use to support a betting decision.

Charts in figures 3.2 and 3.3 demonstrate that punters tend to analyse data from several sources before placing their bets. This proves a need for an application that can reduce the amount of time people spend switching between different type of media to obtain all the information they need. Among the most popular sources were mentioned TV programmes, sports info websites and newspapers.

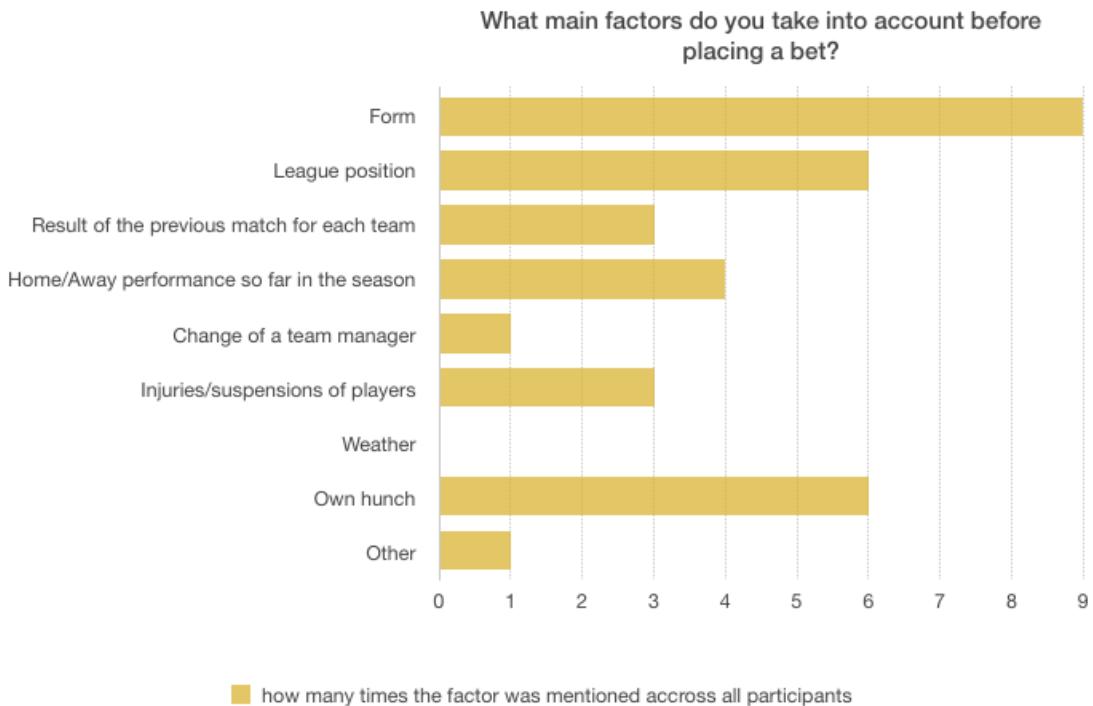


Figure 3.4: A bar chart illustrating the answers of the respondents when asked to specify the factors they consider before placing a bet.

The respondents were asked to list all the factors that they take into consideration before making a betting decision. They were provided with a long list to choose from and asked to specify their own factors if needed. The bar chart in figure 3.4 illustrates the answers. Three factors appeared to be clear winners: form, league position and home/away performance. An interesting point is that punters often use intuition in the decision-making process, as 66% of the respondents mentioned "own hunch" as one of the factors. Incorporating user hunch into the prediction formula will be a clear challenge for me as a developer, however, it looks like the potential application users would like to be able to include it into the calculation.

It was expected that more respondents will mention home/away performance as one of the top influences in making betting decisions. The relatively low interest in this factor can be possibly explained by the fact that the value representing home/away performance cannot be simply found in a standard league table, and a punter needs to make an extra effort to calculate it.

One of the respondents specified a factor that had not been considered in the field "other": "whether or not the odds appear to offer good value". Considering that most survey participants mentioned that they only use one favourite betting provider when placing a bet, this seems to be an interesting point. It looks like the future application will benefit from offering its users odds comparison and possibly a recommendation, such as "odds of the day".

Another interesting fact is that 100

Finally, most of the survey participants answered "yes" when asked whether they would find useful a "web application allowing you to participate in the prediction of a match result by making up your own prediction formula".

Despite the limited amount of respondents, the answers collected with the questionnaire appeared to be a very valuable input to the phase of the project planning.

3.2 Researching Current Solutions

Before gathering the project requirements, it is good practice to conduct research on what current websites are already available to football fans with interest in betting. The research can be a source of inspiration and could also help to avoid potential design mistakes. During the analysis, it is important to attempt to understand the main purpose of the analysed websites, as well as the way they present information to the user and communicate with them.

This section is concerned with websites that can be useful for predicting football results. In our context, these are the various online sources of information a football punter would turn to before making a betting decision unless the decision is based solely on intuition. In general, several different types of such websites can be found online, namely:

1. Sports news websites
2. Football statistics websites
3. Bookmakers websites
4. Communities for sports fans
5. "Black-box" prediction applications

This section of the report looks at one or two examples of each of the categories presented above, analysing the weak and strong points of the chosen website and discussing usefulness of the whole category from the point of view of a football punter.

3.2.1 Sports news websites

This category represents football news websites. Into this category fall both general sports websites with a football section and football news websites, such as:

- BBC Sport [2]
- The Guardian Sport News [18]
- Sky Sports [48]
- The Times Sport section [52]
- Football365.com [10]
- Goal.com [16]

The sports news websites aim to present the news alongside the essential football statistics. The information is usually not as detailed as on the football stats websites, however the very latest football news compensates for this drawback. Each of the *news* websites named above (BBC News, The Guardian, Sky Sports) have a football section that presents the reader with the combination of football news and stats. One of the most popular football news websites is BBC Sport Football.

3.2.1.1 BBC Sport Football

BBC Sport Football is a good quality sports news website. It offers a very neat and simple interface and does not overwhelm the reader with irrelevant graphics. Although, it almost looks too minimalistic, the user can still get all the most important information about football teams and players. The website provides automatically updated live scores across all featured football leagues.

BBC Sport has very impressive news coverage of both major and minor British football leagues, as well as the main European leagues. It can also take pride in high quality writers (journalists) contributing to the website and expert analysis from former players and managers. Being a part of the BBC, it can afford to pay for this extra content, giving it an advantage over most other sports websites. Another benefit is its ability to embed video from the television arm of the BBC directly on the website, both highlights of games and interviews with current players and managers, again something that most websites cannot do.

The main drawback is that the stats are kept to a minimum. This can be a problem for a serious football gambler that wants to analyse the details of the game from all possible angles. However, for the purpose of a fan this level of statistics are sufficient.

SPORT FOOTBALL

[Home](#) [Football](#) [Formula 1](#) [Cricket](#) [Rugby U](#) [Rugby L](#) [Tennis](#) [Golf](#) [Athletics](#) [All Sport](#)

[European Football](#) > [Champions League](#) > [Groups & Schedule](#) | [Results](#) | [Fixtures](#) | [Live Scores](#) | [All Teams](#) | [Leagues & Cups](#)



Top Picks



Reaction: PSG 1-1
Chelsea
16 Feb 2015 | FOOTBALL



I am the headmaster now - Sherwood
17 Feb 2015 | ASTON VILLA



Sportsday - rolling sports news
16 Feb 2015 | SPORTSDAY

By James McMath
BBC Sport

- Ivanovic scores first European goal since 2013
- Cavani scores sixth goal in seven matches to equalise
- David Luiz plays in midfield against former club
- Costa subdued on return from suspension

17 February 2015
Last updated at
22:21

4.1K
 Share

Chelsea withstood waves of Paris St-Germain attacks to claim a draw in the first leg of their Champions League last-16 tie.

Featured in this story

- [Football on the BBC Facebook](#)
- [World Football Podcast](#)
- [Match of the Day Twitter](#)
- [BBC Sport Facebook](#)
- [BBC Sport Twitter](#)
- [Uefa](#)
- [Uefa Champions League](#)

Latest Football

Last updated 17 Feb 2015 UK

PREVIOUS RESULTS

TUE 17 FEB 2015 - CHAMPIONS LEAGUE
Paris St G 1-1 Chelsea

Figure 3.5: BBC Sport - Football section

3.2.2 Football statistics websites

There is a large selection of football websites dealing in far more detailed statistics and analysis. These are some examples of websites in this category:

- WhoScored [20]
- Squawka [49]
- Injuries And Suspensions [23]

3.2.2.1 WhoScored

Among all the football stats websites I have analysed, WhoScored is one of the most impressive ones. It has statistics on every possible aspect of a football match, team or player, some of which are less relevant than others (number of throw ins per game is probably not as important as goals scored for example) but it has everything a gambler could want to know before making a bet. It does provide some football news and articles but it is clearly not their main focus and seems a bit amateurish when compared to the bigger news websites. The website is extremely well designed, and its navigation is

intuitive. WhoScored offers statistics and deep analysis on the major European divisions, as well as providing less detailed but still impressive data on over 500 smaller leagues and 15,000 teams. As to the data source, the website is supported by Opta, the largest live sports data company that is used by BBC Sport, Sky Sports and other significant UK sports news providers.

The way "WhoScored" presents information on particular matches, both upcoming and already played, is detailed while being uncluttered, which makes it both very useful and easy to use. This is definitely an aim of this particular project and I will be using a similar format when designing the website.

3.2.2.2 Squawka

Squawka is another website worth looking at.

Upcoming Fixtures				
Show Fixtures and Results in : English Barclays Premier League for : Season 2014/2015				
Displaying 1-30 of 130 1 2 3 4 5 Next » Last »				
Teams	TV Channel	League	Kick Off	
Villa vs Stoke	Premier League	15:00 on 21st February 2015	Set Reminder >	
Chelsea vs Burnley	Premier League	15:00 on 21st February 2015	Set Reminder >	
Palace vs Arsenal	Premier League	15:00 on 21st February 2015	Set Reminder >	
Hull City vs QPR	Premier League	15:00 on 21st February 2015	Set Reminder >	
St'land vs WBA	Premier League	15:00 on 21st February 2015	Set Reminder >	
Swansea vs Man Utd	Premier League	15:00 on 21st February 2015	Set Reminder >	
Man City vs Newcastle	Premier League	17:30 on 21st February 2015	Set Reminder >	
Spurs vs West Ham	Premier League	12:00 on 22nd February 2015	Set Reminder >	

Figure 3.6: Squawka

It is an application for football fans that uses real-time data visualisations to show how a game is progressing. The main idea behind it is to show users the live stats as the game is being played and maybe put a bet on "in-play", for instance it may show that one team is having much more shots on goal than the other so it would make sense to put a bet on them to score the next goal.

From a visual point of view, Squawka has a nicely designed, pleasant interface. However, it is a little bit heavy on the client-side (Javascript), which contributes to sometimes slow performance. Another downside of the website is an extensive amount of adverts that distract the user from the main content.

3.2.3 Bookmakers Websites

With the arrival of the Internet many existing bookmakers opened up web based operations to complement their existing business. Within short period of time online gambling became very popular with punters all around the world. Most online bookmakers contain high quality sports statistics that aim to support users' betting decisions.

Names like Ladbrokes, Bet365, William Hill are among the most popular bookmakers online, but there are literally hundreds of betting websites now.

- Paddy Power [37]
- Ladbrokes [26]
- Bet365 [3]
- William Hill [60]

3.2.3.1 Bet365

According to the Wikipedia [59], "Bet 365 Group Limited, is a United Kingdom based gambling company. Bet365 is one of the world's leading online gambling groups with over 14 million customers in two hundred countries". In my opinion, Bet365 has one of the nicest websites among bookmakers. The website is very well structured, it has intuitive, user-friendly navigation and is easy to use. Bet365 offers a free live streaming service and an impressive coverage of live sports statistics. However, none of the gambling websites I looked at were as intuitive or as thorough with regard to the statistics as Whoscored.com, maybe as they don't really want their customers to win after all.

3.2.4 Communities for sports fans

These websites are specialised social networks for sports fans and punters. So far, websites of this type are not particularly popular, maybe because gambling is more of an individual pursuit and gamblers, while liking to be proved right, do not like to share their winning formula with others or alternatively to show that they were wrong.

Many of the features of the websites in this category, such as experts tips, users' comments, forums, can be very useful from a punter's perspective, especially when dealing with a league you are not overly familiar with. I have analysed three websites in this category.

- OLGB Betting Community - [34]

- Vital Football News and Fans community - [55]
- Punters Lounge - [41]

OLGB is a friendly community for punters with many interesting features and tools. It will be analysed in more detail in a separate subsection below. Vital Football is a "network" website. It runs a separate website for every football club from the Premier League and the Football League in England and the Scottish Premier League with each "club site" having their homepage, own editors and a forum. Punters Lounge is a "betting and poker community". Its most interesting feature is a forum for sports gamblers. The website also offers free betting advice tips, live sports streaming and odds comparison.

3.2.5 OLGB Betting Community

OLGB community market themselves as a website for punters who share their expertise and work together to maximise their betting profit, so it is very relevant to this project. OLGB has a wide variety of features and tools. However, real punters' opinion and tips is the main focus of this website. For example, the user can navigate to an upcoming event page on OLGB website (in the section "free tips") and check how many other users predicted either team to win. Users' choice is usually justified with an comment. For each option OLGB suggests the best odds from one of the most popular bookies. This saves user the trouble of going to a website like OddsChecker[6] to compare the odds. The comments feature is very interesting and it can be applied as an "optional requirement" for this project.

OLBG also runs a virtual betting game (tipster competition) where users can "bet" virtual money on real betting events. The tipsters in the top 100 of the tables for each sport each month receive a real money prize. Users of the website can also see a leaderboard of the most successful tipsters. The leaderboard contains information on the amount of tips made within certain period of time, LSP (Level Strike Profit), ROI (Return on Investment) for each tipster. The virtual betting feature is very interesting and a leaderboard showing how each user is getting on could be a good way of getting people more involved in my application, probably as an optional feature to avoid losing those users that don't want to share their wins and losses.

OLGB also has a betting forum, betting blogs section, betting school and much more. The community website has a connection to several popular bookmaker websites. For example, it offers a tool to help users to compare bookmakers websites. It also promotes free bets and various "bookies" promotions.

The website has also several drawbacks. Firstly, the interface is a little outdated and it takes a while to find your way through the complicated and rather confusing navigation. Secondly, the "Help"

section only answers some basic questions and can be slightly disorientating for a new user. Again, it's statistics are not as detailed as the likes of Whoscored and as it is run as a community, the news and information are nowhere near as good as the specific news websites.

3.2.6 Black-box prediction applications

These are betting applications that are marketed as systems that can predict the outcome of a football match using their own unique statistical models and calculations. Known as black-box systems, these applications avoid sharing the exact logic used inside the system with the user. The punter subscribes to a betting system online and simply follows its suggestions for either the estimated outcome of a football match, or on events within a match. For example it may suggest that a lot of goals are likely to be scored in a particular match or even that a lot of corners are likely and therefore the user should bet on that. These are the example apps in this category:

- Math Betting [32]
- Footbee [12]
- Vitibet [56]
- Forebet.com - Mathematical Prediction [13]

When paying for the subscription for a black-box betting application, the users assume that they are gaining access to a unique system created by team of experienced football experts. Additionally, there is an expectation that the application has complicated statistical models and calculation analysing a large database of football statistics behind each betting tip. The users also hope that they are paying for a *secret* betting system that is known to and is used by only limited number of other punters. Summing up, a black-box system application sounds like an easy way to sustainable profit. Unfortunately, this does not have to be the case.

According to Macos [28], the complexity of a betting system does not necessarily correlate with its profitability. The logic behind a successful betting system can be relatively simple and would still work. Secondly, in order to predict a football match result, it may be enough to analyse only relatively recent events (for example last 6 matches, recent players' performance, etc.), without having to perform computations on large set of historic football data. This is due to the fact that over time many various factors can cause change in team performance. Therefore, analysing data from several months ago would not help the punter to make more precise prediction.

3.2.7 Conclusion

After having analysed the above websites, I came to the following conclusion. A well-chosen combination of several websites would definitely be able to provide enough information to make a thoughtful betting decision.

Many punters have their own football betting system (betting strategy) [28]. Although even the best system cannot guarantee success, it can greatly increase the probability of making a profitable bet. Therefore, before making prediction, a thoughtful punter will conduct a little research for each match. The aim of such research would be to collect relevant information about the teams involved in the game. The type of information needed will depend on the *input variables* of the betting system used by each particular punter. The problem is that many football stats websites overwhelm their users with detailed statistics that is irrelevant for prediction purposes. Hence, punters often have to "hand-pick" the important information from several sources for each match.

The developed application will attempt to put all the relevant statistics in one place and break the information down into logical modules. In addition, the application will enable users to pick the input variables themselves, and assign them a weight of the user's choice, enabling the user to completely ignore data that they don't feel is important and focus instead on what matters when deciding on an outcome to bet on. With that comes the ability to improve their system as they go along, they can see what is or isn't working and hone their system to give them the best possible chance of winning, which is something not provided by the "black box" type applications.

3.3 Requirements Specification

Radice and Phillips [42] define project requirement as follows: "A *requirement* conveys an essential property that the system must or should satisfy." Requirements analysis involves gathering information in order to meet customer needs and defining what the future application is expected to do. This phase of software engineering is especially important in the industrial environment, when developing an application for a customer. In that case, clarifying the requirements in the early stages of the project would help to ensure that both sides understand and agree on the feature set of the future application.

Although it is not very likely that requirements for this project will change during the development process, defining requirements can be very beneficial. The detailed requirements analysis will aid in the understanding of how different parts of the project are expected to interact with each other, as well as how the application will communicate with its users.

For better transparency, project requirements have been split into functional and non-functional requirements. Functional requirements will be further subdivided into mandatory and optional, depending on the degree of constraints.

Before outlining project requirements, it is good to start with some definitions relevant to the project as a whole.

3.3.1 Definitions

Application Football League - in order to reduce unnecessary complexity at this stage of the project, the application will be only supporting one league.

Matches Overview – a list of upcoming and played matches presented on the main page of the website.

Dashboard - an interface available to authorised users. Dashboard is a starting point for users to view, edit and commit saved matches, as well as access other prediction-related content and tools.

Prediction Module – my own term. Each prediction module represents an input system variable in the betting system [27]. The aim of each prediction module is to evaluate and compare in a predefined way blocks of latest football statistics for each of the teams participating in the game (for example, position of each team in the league standings table). The result of this comparison is a module *prediction value* that expresses the probability of either team to win based on one module statistics.

Prediction Settings - a set of weights assigned to prediction values in the betting system in order to forecast the result of the match.

System Default Prediction Settings - the application has a set of "recommended" weights that are used in the prediction calculation by default. **User Default Prediction Settings** - each user of the application can override the system default prediction settings and save their own set of weights. From the moment those weights are in the database, they will be applied by default in the prediction calculation for each match saved by the user.

Match Specific Prediction Settings - each user of the application can also save a set of prediction settings applying to only one match.

Match result - "Home Win" in case of the win of the home team, "Away Win" in case of the win of the away team, "Draw" for the draw.

3.3.2 Functional Requirements

Functional requirements describe the behaviour of the application in terms of its functionality. These are the "must have" functions of the application addressing the business targets that application must satisfy. Good functional requirements must be complete, coherent and unambiguous. In order to add structure to the design and development process, the project was logically divided into high-level features of the future application. The *mandatory* functional requirements are grouped by the functionality related to these features.

3.3.2.1 Authentication and User Profile

These are the requirements for the basic functionality of the web application, such as account registration, login, logout and account management. The requirements relating to the user profile page will be also listed in this subsection.

The application will allow users to register and create a new account with the application.

- User will be able to register using a standard web form.
- For the registration purposes user will provide a valid email address and a password.
- User will confirm a password in a separate input field.
- On completing the registration form, the application will send the user an email containing a confirmation url.
- On successful confirmation of an email address using the above confirmation url, user will be successfully registered.
- In case of any technical problems with the initial confirmation email, the application will generate a new confirmation url and send it to users on request.

The application will allow users to sign into their accounts using a standard web form.

- User will provide email address and a password associated with it.
- When signing in, user will provide valid credentials, otherwise an application will show a validation error.

Account management

- The application will enable users to manage their accounts by changing personal information

related to it (for example, location, favourite football team) using a web form.

- Users will be able to change the email address associated with their account.
- Users will be able to change their passwords at any time.
- The application will provide users with a way to recover their lost passwords.

User profile

- The application will have a special user profile page containing all the essential information about the current user.
- Users will also be able to view profile pages of other users of the application.

3.3.2.2 Matches Overview

Below can be found requirements related to the matches overview.

- On the main page of the application, user will be presented with a list of upcoming matches for the current season in the league.
- User will also be able to view a list of matches already played in the current season and switch between upcoming and played matches using navigation tabs.
- Each of the entries in the match list will contain the most basic information about the match, e.g. names of the teams participating in the event, kick-off time and date, full-time score (only for the played matches).
- For each of the unplayed matches, user should be able to navigate to the match page and see more details about the match.
- From the main page user will be able to save any unplayed match to the dashboard for a later review.
- For each of the played matches user will be able to navigate to the match page and see more details about the played match.

3.3.2.3 Prediction

In this part of the report, the Prediction feature of the application will be explained and the related requirements will be listed.

The outcome of a football match will be calculated after evaluating several factors that can influence the result. An example of such a factor could be the previous match result, the position in the league, the team's performance at home or away, a recent change in team management, individual player's performance, as well as injuries and suspensions of team players. When considered as a part of a betting system, a factor is basically an *input system variable* [28]. As already mentioned above in the "Definitions" 3.3.1, each of those input variables will have a "prediction module" representing it in the application and the main outcome of each prediction module will be a "prediction value", percentage that tells the user which team is more likely to win the match.

Finally, the betting system will assign a weight, the value of which will depend on user settings, to each of the prediction values. The weights or prediction settings determine the relative importance of each factor. By applying the weights, the user indicates that some factors are more important to the outcome of this particular game than the others. The final result will be calculated as a weighted average.

As it can be seen from the above explanation, the list of factors that can be considered in the application can be quite long. To simplify the development process, it was decided to limit this list to only three factors. Factors listed below were chosen based on the analysis of the data obtained from the target audience questionnaire 3.1. "League position" and "Form" were two top answers when answering the question

- League position of each of the two teams
- Form of each of the two teams
- Home/Away performance of each of the two teams

During the implementation phase these factors will be transformed into prediction modules and integrated into the application. On the top of that, an extra prediction module, "User Hunch" will be created. This is a special module that will allow the users to incorporate their intuition into the prediction calculation and thus influence the result of the prediction. "Own Hunch" was also one of the most common responses among the questionnaire participants.

Below can be found the requirements relating to the prediction feature of the application.

- To calculate prediction values for each module, application will use the *system default* settings in absence of either the *user default* prediction settings or the *match specific* settings.
- The application will make use of the *user default* prediction settings (explicitly set by the user using a web form) in absence of the *match specific* settings.

- The application will apply *match specific* settings if the user has set them for this match.
- Match result prediction will be calculated as a weighted average of prediction values using an appropriate set of weights based on the logic outlined above.

3.3.2.4 Upcoming Match View

For each unplayed match, user should be able to view relevant football statistics, change the match result prediction by adjusting the weights and "commit to bet" the match, once they are satisfied with the final output. Below can be found the requirements illustrating this part of the application functionality.

Functionality available for all users.

- Upcoming match view will contain general information about the match in the view "header", e.g. names of the teams participating in the event, kick-off time and date, the result of the last match for each of the teams.
- As well as the match "header", the view will present the user with a list of prediction modules.
- Each prediction module will contain the relevant statistic relating to that module.
- For each prediction module, it will clearly show which team is more likely to win and what the probability of this outcome is if the match prediction was based solely on this module. In other words, each prediction module will have its *prediction value* clearly indicated.
- In case the user has not saved the upcoming match to the dashboard, the view will display only list of prediction modules and associated prediction values. However, it will not be possible to make a match result prediction or commit the match.

Functionality available for the users who saved the match to the dashboard.

- The user will be able to see what weights are being used for each prediction module.
- The application will allow the user to set new match specific prediction weights on this page.
- As well as the modules based on the football statistics, the upcoming match view will contain a special module, "user hunch". Its importance was already outlined in the subsection "Prediction" [3.3.2.3.](#)
- User will be able to commit the match, once satisfied with the result of the prediction.

- Once the match is committed, the prediction cannot be changed.

3.3.2.5 Played Match View

After the game has been played, the user will be able to navigate to the played match page. The page will contain brief match statistics as well as the summary of the performance of users who placed the bet. This view should be more about the retrospective analysis of the users' betting strategies, i.e. the weights used for modules, rather than the detailed statistics from the actual match, i.e. shots on target, possession, etc. Hopefully, the information presented in this view will allow users to compare their results with the fellow punters and encourage them to analyse their own betting strategy and optimise performance.

These are the requirements relating to the played match view.

- Played match view will have a match "header", similar to the header in the upcoming match view.
- User will be able to view a brief summary of the pre-match statistics. This information will give the users an idea on how the two participating teams were performing previously to the played match.
- Users will be able to see an overview of the website population performance, more specifically performance of users, who committed a bet for this match.
- Users will be able to see the visualisation of the website population's choice of prediction weights used for this match, possibly a pie chart.
- The view will also have the visualisation, possibly a bar chart, of the website population's choice of the winner for this match (home team, away team or draw).
- It should be clear from the view what was the result of the bet for an authenticated user. The view will clearly indicate the user's choice and prediction probability. This part of the view will be only available for users who bet on this match and omitted for the rest of the population.

3.3.2.6 Dashboard

Dashboard is a key view of the application.

The idea behind the dashboard in this application is similar to online shopping experience: user saves an item to the shopping basket and can later submit a purchase or cancel it. In our case, user browses

through the list of matches on the main page of the application and saves matches to the dashboard for a later review. The requirements listed below describe the dashboard functionality in more detail.

- User can save a match to the dashboard from the matches overview [3.3.2.2](#).
- User can remove any unplayed or played match from the dashboard.
- Dashboard will have a *dashboard menu* holding the links to various tools and views. Suggested entries of this menu are "Upcoming Matches", "Archive", "Prediction Settings".
- "Upcoming matches" will be a default view of the Dashboard. It will contain all the saved matches that have not been played yet.
- Tab "Archive" will navigate the user to a view containing the saved matches that have already been played.
- "Prediction Settings" tab will open a view with a web form, which can be used to save the user default prediction weights.
- In the list of saved matches (both upcoming and played), it will be clearly indicated (colour coded) whether a saved match was committed and what the user predicted.
- If there are any committed matches in the Archive view, it will be clearly indicated whether the user won or lost a bet.

3.3.2.7 Notifications

The system should be able to notify its users whether they won or lost the bet.

- The end of a match is what will trigger new notification messages in the application. Once a game is finished (the full-time score is available), the application will send a notification to the user.
- The user will be able to view the list of all notifications.
- The user will be able to delete a notification.
- The user will be also able to delete all notifications from the list.

3.3.2.8 Leaderboard

The application will have a Leaderboard, which is a table comparing the current standings of the application users in terms of their betting performance.

- The leaderboard table will display players' usernames, their total win and loss points. The table will be sorted in order of win points.
- Hence, the most successful punters will be on the top of the table.

3.3.2.9 Optional requirements

The functional requirements were split into two main groups: mandatory and optional. This is due to the fact that the application was developed over a relatively short period of time and there was not enough time to implement all of the intended functionality. Mandatory requirements represent the "minimum viable product", application developed with the core features that are sufficient to prove the concept of the future product.

On the other hand, the optional requirements illustrate possible improvements that can be made to the application in the future. Additional research will be needed in order to decide which of the features listed below will be the most useful for the target users. The future scenarios of how the application can be extended in the future will be further discussed in the section "Improvements and Future Work" of the chapter "Conclusion".

- The application should have more prediction modules. Suggested modules are "Recent change of team management", "Injuries and suspensions of players", "Individual players' performance", "Previous game result", etc.
- More football leagues should be supported.
- The user should be able to add/remove any prediction module from the Upcoming Match View. This should help keep the focus on only relevant information from the user's point of view.
- The application should offer a betting performance tracking tool that will record the details of the past bets and also keep track on whether the punter is profitable in the long run.
- The application should offer its users odds comparison functionality. It is a known fact that the odds for the same event may vary considerably between different bookmakers. The application should compare the odds across the most popular bookmakers and suggest the best option with regards to the user's prediction (home win, away win or draw). Clicking on the suggested odds

should take the user to the bookmaker's website or open a bet slip.

- The application should get some features of a sports fans community. Users should be able to leave comments on each match explaining the prediction they made and also follow each other. Followers should see the comments from the followed users in their news feed. The news feed should be accessible from the dashboard.
- The application should implement "Sign In with Facebook" functionality allowing its users to login with a single click.

3.3.3 Non-functional Requirements

Non-functional requirements specify how the system is going to perform.

- **Usability** - The application interface should be easy to understand and learn for a new user. The navigation of the website should be highly intuitive.
- **Responsiveness** - The application should be fully responsive. The websites will be tested for a variety of screen resolutions and the minimum screen resolution will be 640x960 (DVGA - iPhone).
- **Performance** - Optimising performance will be crucial for the application, as there is a direct correlation between the application response time and the user experience. Performance problems will be detected and eliminated as soon as they appear.
- **Cross-browser support** - The application will be supported on a minimum set of web browsers, such as Chrome, Internet Explorer 9+, Safari, Firefox.
- **Maintainability** - The focus should be on delivering clear and maintainable code. The code needs to be easily understandable by other developers. For this purpose the best practices of software development and the used languages will be utilised throughout the implementation phase of the project.
- **Extensibility** – The system will be developed with a large-scale application in mind. It should be easy to apply ongoing changes to the project.

3.4 Overall Architecture

The high level components of this system are quite simple.

Design of an application as a whole, overall design (just boxes and lines) Architectural diagram (overview) (aosabook.org/en/moodle.html -example), quite high level

3.5 Choice of Third Party API

After analysing the functional requirements, it became apparent that this type of application will need the latest football data in order to operate correctly. The easiest way to load data into the application would be to integrate the application with a third party API. The process of finding an appropriate API for the project will be described in this section.

After brief research, one thing became apparent. Live football data is a very desirable product and therefore it is not easy to find free of charge live football data API. The key problem is that the data has to be very recent. Real-time data in particular is very expensive, because of its use by the gambling industry for betting on various markets as the games are going on. It is actually much easier to find free historical football data.

This is a list of API providers that have been researched and evaluated.

- Optasports.com [36].

Opta is the industry leader. It provides a wide range of XML feeds covering many sports. The feeds include fixtures and results, live scores, live player stats and many more. Data provided by Opta is very reliable and is used by top-notch clients, such as BBC Sports, BT Sport, Sky Sports, as well as many betting providers and newspapers.

- Openfooty [35].

Openfooty is an interesting project with very detailed API documentation. However, a quick look at the developer forums shows a stale community and many questions about why no one seems to actually be able to get a developer key. Unfortunately, I also did not manage to obtain a key for this API.

- Football API [11] .

This is a paid API service. The API restricts by IP addresses and limit calls based on the package. On the bright side, it offers the English Premier League endpoints for free (demo use). The API includes endpoints for competitions, teams, standings, live scores, fixtures and commentaries.

- XML Soccer [62].

Another paid API service that offers full access to the Scottish Premier League data for free.

Football API was chosen to be integrated with the application. As already mentioned in the "Definitions" [3.3.1](#), it was decided to support only one league for the time being. Football API offers its users free access to the English Premier League, which seems to be a great choice for our application, as this league's worldwide popularity will hopefully make it easier to find other league-related data and information that might be needed during the development process.

3.6 Project Plan

The project progress timetable is presented in the Gantt diagram below. Two main milestones were set for this project: firstly, to develop the first prototype of the application and secondly, to complete the second prototype by the end of April, 2015 (this includes all the testing and bug fixes).

The first prototype will have implemented most of the high-level features of the application as introduced in the section "Requirements" [3.3](#). The second prototype is a more refined version of the application; it will implement all mandatory requirements listed above and have the visual side of the website polished up.

Chapter 4

Application Design

Before making a start on of the implementation phase, a lot of effort was put into the creation of the application prototype. Prototyping is the process of developing the initial model of the future application in order to determine its correct structure, functionality and the general concept behind it. A prototype is just a model and may differ from the final product.

In order to fully describe a new product, the following aspects of its design should be modelled: product functionality, visual design and navigation. Those aspects are best covered by different techniques: user stories for the application functionality, activity diagrams to describe the user journeys and sketches to capture the visual design. The modelling process will be illustrated in detail in this chapter.

First, the project requirements outlined in the previous chapter were used in order to create a mind map. This is needed for brainstorming and allows the setting out of the initial ideas about what the visual design and the user experience should be. The next step is to produce hand-drawn wireframes that were sketched for each individual page of the application. To complement the wireframes, a set of user journeys were developed to describe the user interaction with the system.

In summary, this chapter describes the process of transforming project requirements into the system design. Several design elements (website structure mind map, a set of wireframes) were produced before the implementation phase. Other elements (user stories, activity diagrams and the database schema) were designed in iterations, alongside with the development of the high-level features introduced in the chapter "Requirements Analysis" [3](#). During the prototyping phase the main focus was on creating a visually appealing, simple and highly usable design.

4.1 Application Structure

The prototyping process started with producing a large mind map of the future application. The mindmapping is a very useful way to brainstorm on the initial ideas, capture and organise them and was an extremely beneficial tool for this project in particular. The final version of the diagram puts together the structure of application pages, navigation scenarios and other ideas relevant to the visual design. A part of the developed mind map can be found in the figure below.

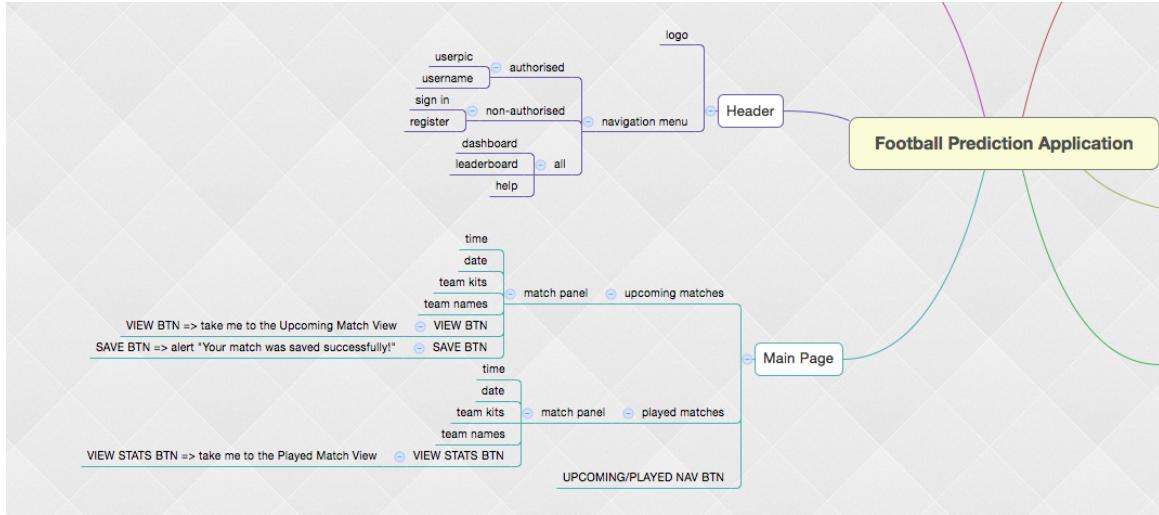


Figure 4.1: Mind map capturing the results of initial brainstorming on the application structure and navigation scenarios.

4.2 Branding and Visual Design

At this stage of the prototyping process it was important to decide on a suitable name for the future application. Below is the list of some names that were considered.

- Too Close To Call
- Sure Thing
- Footy Expert
- Beating the Odds

SureThing was chosen as the project name for being unique, simple and catchy, whilst expressing the essence of the future application. The name evokes optimistic feelings and is quite suitable for a prediction system that is transparent to its users and will increase their chances to win a bet in the long run.

As it can be seen from the long list of mandatory requirements, the project will require a lot of time to be invested in implementing its functionality. Therefore, it was decided to make use of the Twitter Bootstrap framework on the front end [54] [40] in order to reduce the amount of time spent developing the visual side of the application and create simple and consistent interface. As a result, the final design is a mixture of ready-made solutions and my own ideas on how to visualise unique elements of the application layout, such as dashboard side menu, match panels in matches overview, the layout for the prediction modules in upcoming match view, etc.

4.3 Analysis of the competitors experience

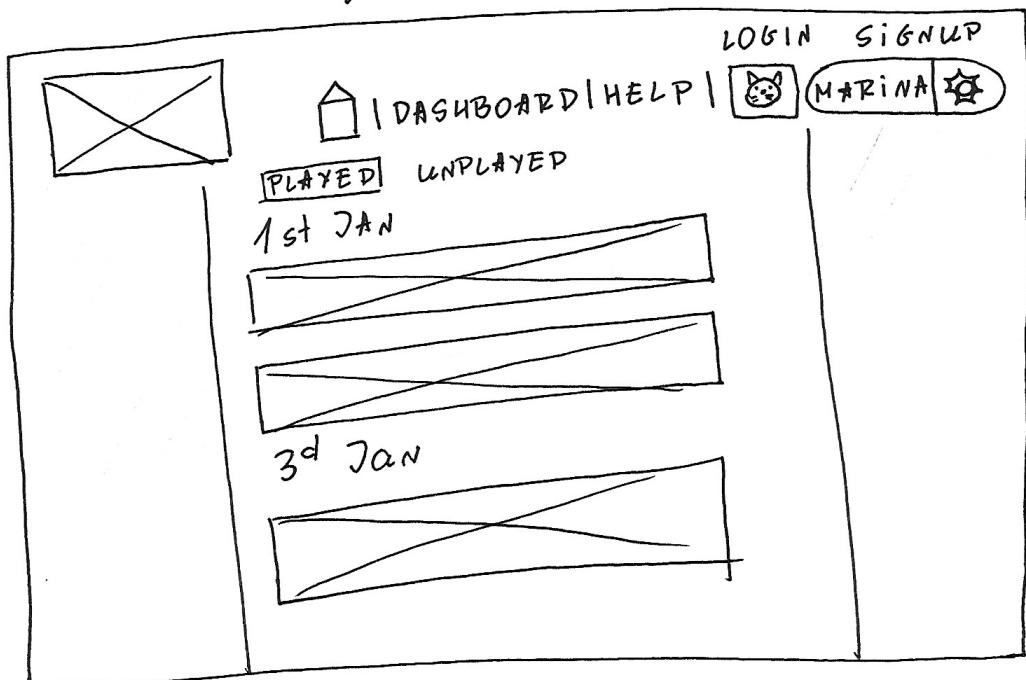
As the next step, I took another look at the existing websites, expecting to get some ideas on how to approach the visual side of the website and to improve its usability. This step is an important stage of an application prototyping process: it allows to learn from the best design practices and possibly avoid potential errors. The usual practice is to concentrate on few websites of the direct competition as a first step. However, it was not possible to identify the direct competitors, as the idea behind the project is quite unique. Therefore, I analysed several football statistics and community websites, namely *WhoScored?* [20], *Goal.com* [16] and *OLGB Betting Community* [34], making a note of how those websites present football statistics to their users, what are the main differences between the presentation of an unplayed and played match and what interesting features each website offers to its audience. This analysis served as a great source of inspiration and the basis for the next step - producing the hand-drawn wireframes.

4.4 Wireframes

In the early stages of prototyping, a piece of paper and a pencil are the first choice of many designers. Sketching has a number of advantages when compared to the use of the graphic design software, such as Fireworks or Photoshop. With the editors it is easy to get distracted by brushing up unnecessary details too early. On the opposite side, sketches allow the quick expression of ideas and offer a lot of flexibility. It is easy to add notes, make small changes or replace an outdated sketch with a fresh one.

In this project, each of the sketches represented a separate “view” of the website. The scale of a “view” might differ. For example, some sketches show a whole page (home page, dashboard page, etc.), others only capture an important part of a page in more detail. Below can be found scans of some of the paper wireframes used for this project.

A0 main page



GAME & DATE DESIGN:

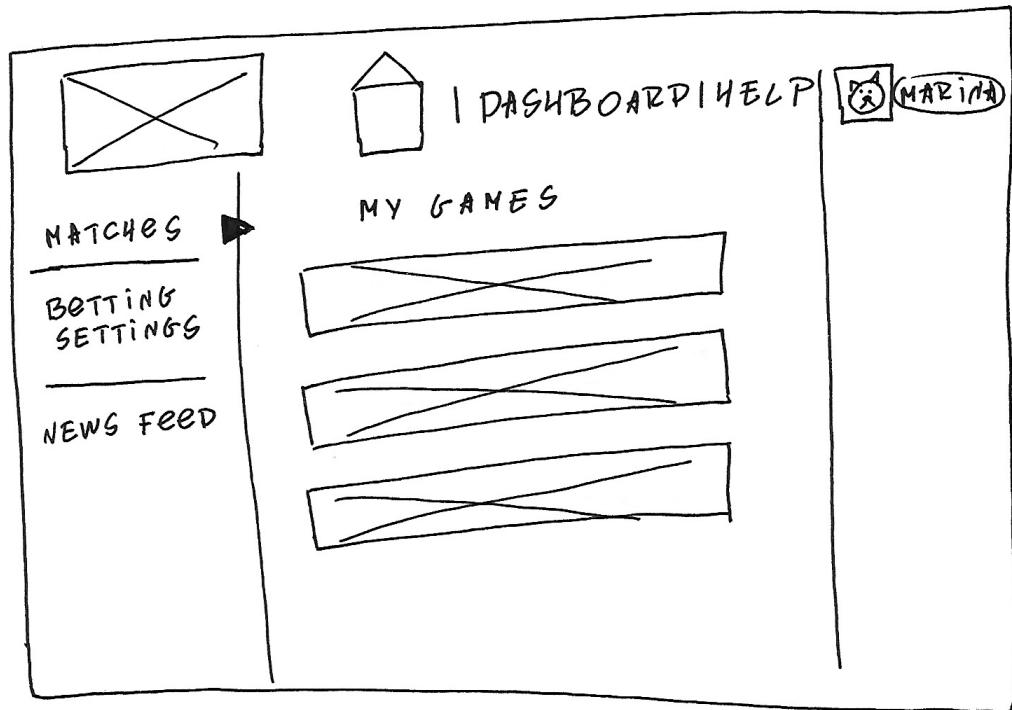
SAT 08 NOV

12:45 vs.
LIVERPOOL
WWLDW CHELSEA
WLWDL

13:45 vs.
SWANSEA
LWLWW WEST HAM
LWDWD

Figure 4.2: A wireframe of the main page of the website.

B0 DASHBOARD

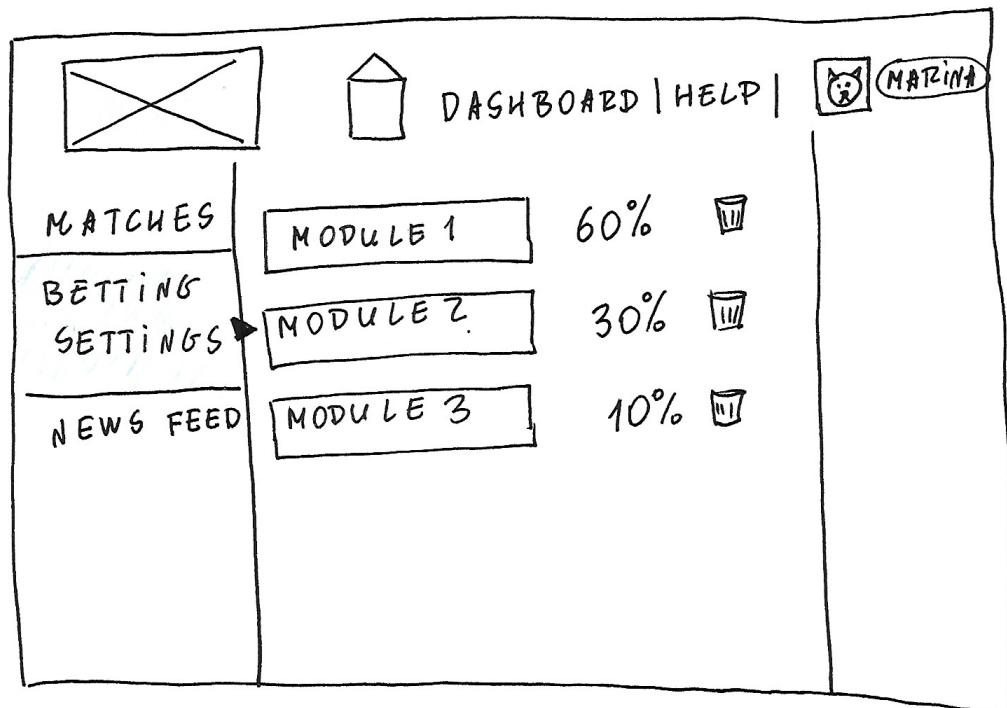


EMPTY DASHBOARD



Figure 4.3: A wireframe of the dashboard of the website, displaying the overview of the saved matches.

B1 DASHBOARD / BETTING SYSTEM



B2 DASHBOARD / NEWS FEED (I'M FOLLOWING)

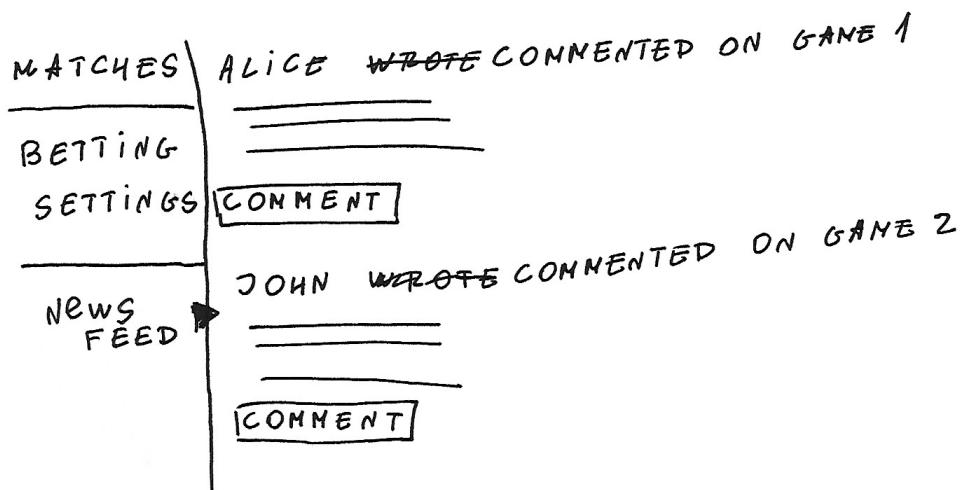


Figure 4.4: A wireframe of the dashboard of the website, displaying the prediction settings form.

CO MAIN PAGE/UPCOMING MATCH VIEW

HEADERS

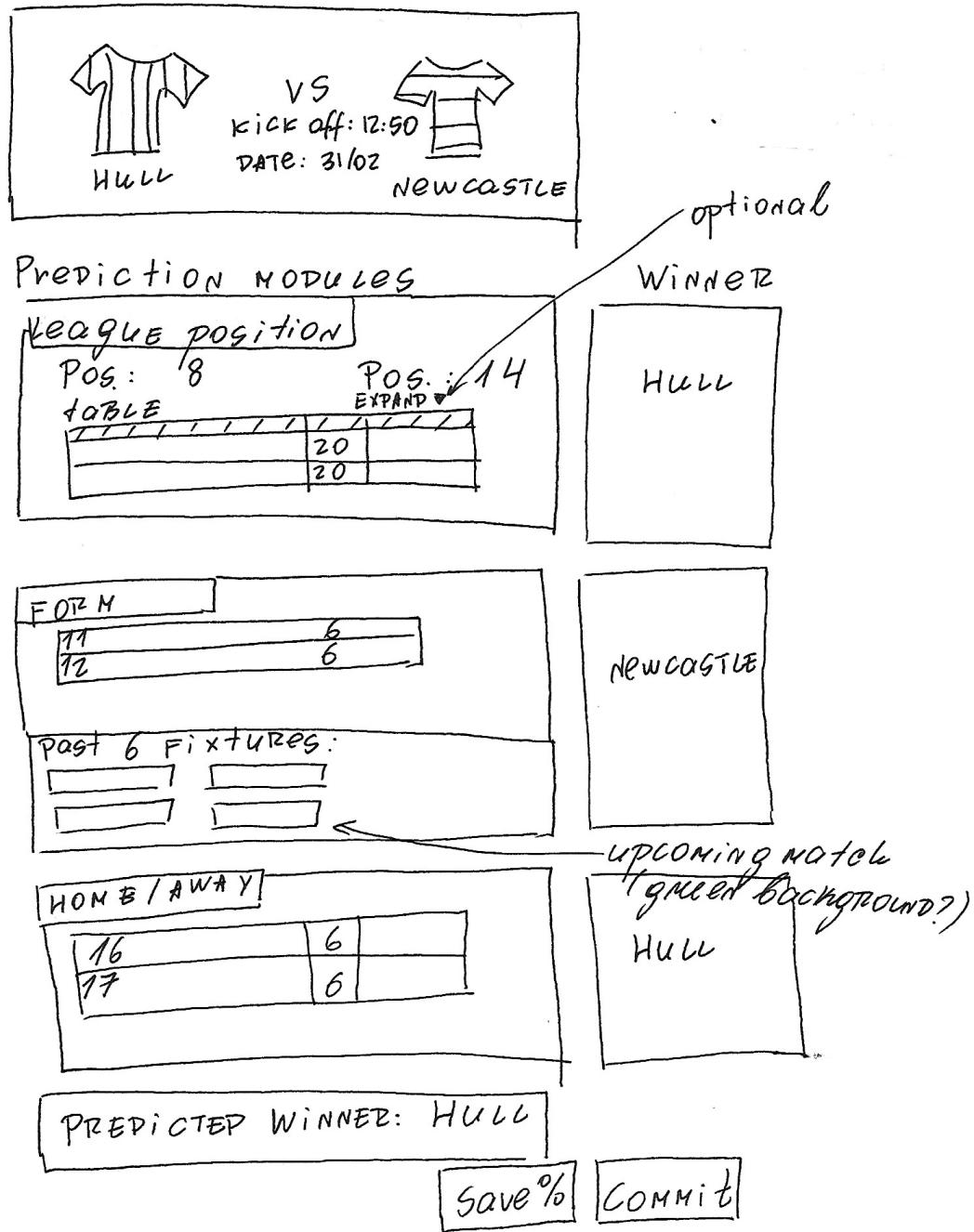


Figure 4.5: A wireframe displaying an example of an upcoming match.

WD MAIN PAGE / PLAYED MATCH VIEW

HEADER

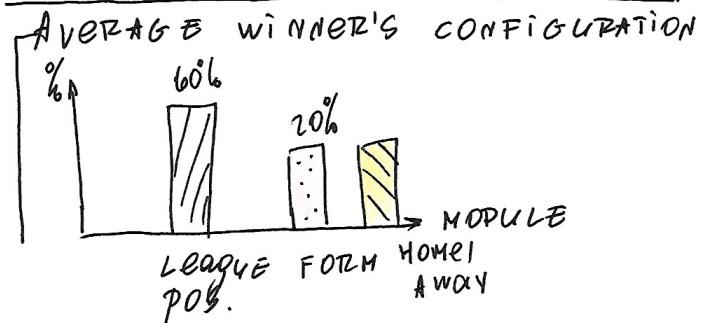
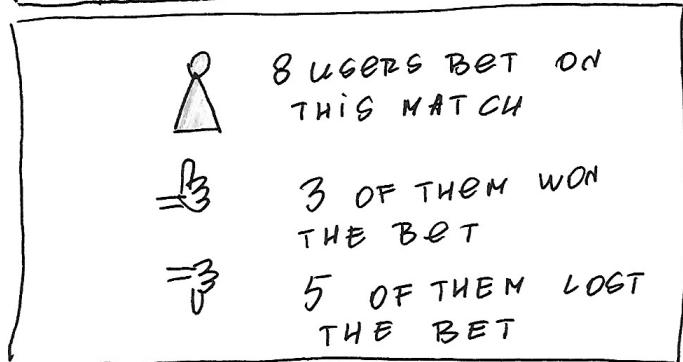
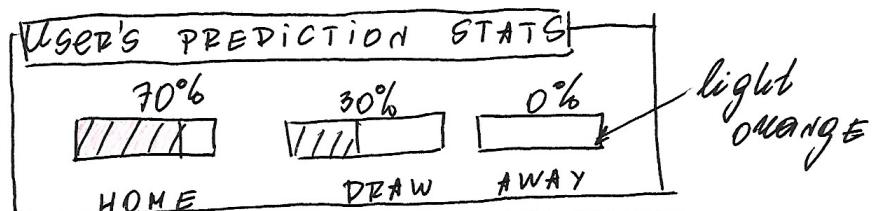
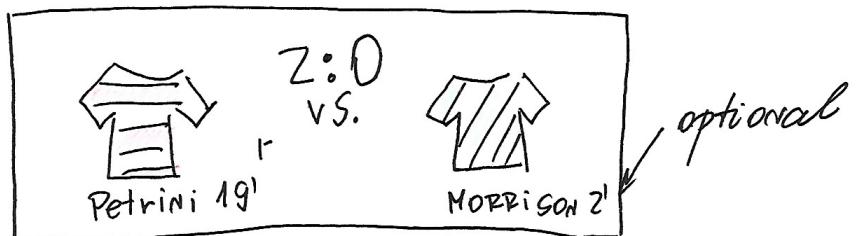


Figure 4.6: A wireframe displaying the stats of a played match.

4.5 User Stories

To describe the application functionality, a set of user stories was produced. User stories serve the similar purpose as the more traditional use cases, but they are not exactly the same. A use case is a graphical description of the interaction between the user and the system. On the other hand, a user story is usually a short sentence that expresses the need for product functionality from a user point of view. User stories are a more lightweight, informal version of the use cases. They are also another way to present the project requirements.

- As an Anonymous User, I would like to register with the application so I can use its full functionality.
- As an Application User, I would like to edit the information associated with my profile.
- As an Application User, I want to be able to browse through the matches list and save a couple of them to my dashboard so I can review them later.
- As an Application User, I would like to make a bet on a particular match.
- As an Application User, I want to see other players' performance so I can compare my performance to others.
- As an Application User, I want to update my prediction settings so I can improve my betting system and a probability of winning.

4.6 Activity Diagrams

In order to illustrate user interaction with the system and to complement the user stories, a number of activity diagrams were created. The diagrams can be very useful in terms of visualising the workflow and linking together individual user stories.

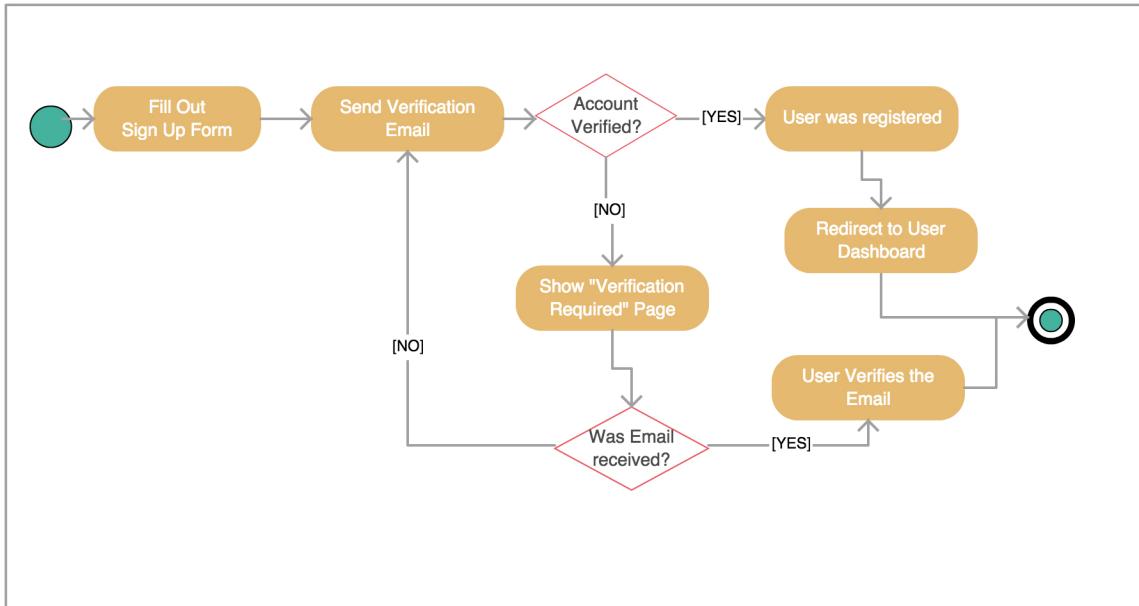


Figure 4.7: User Sign Up activity diagram.

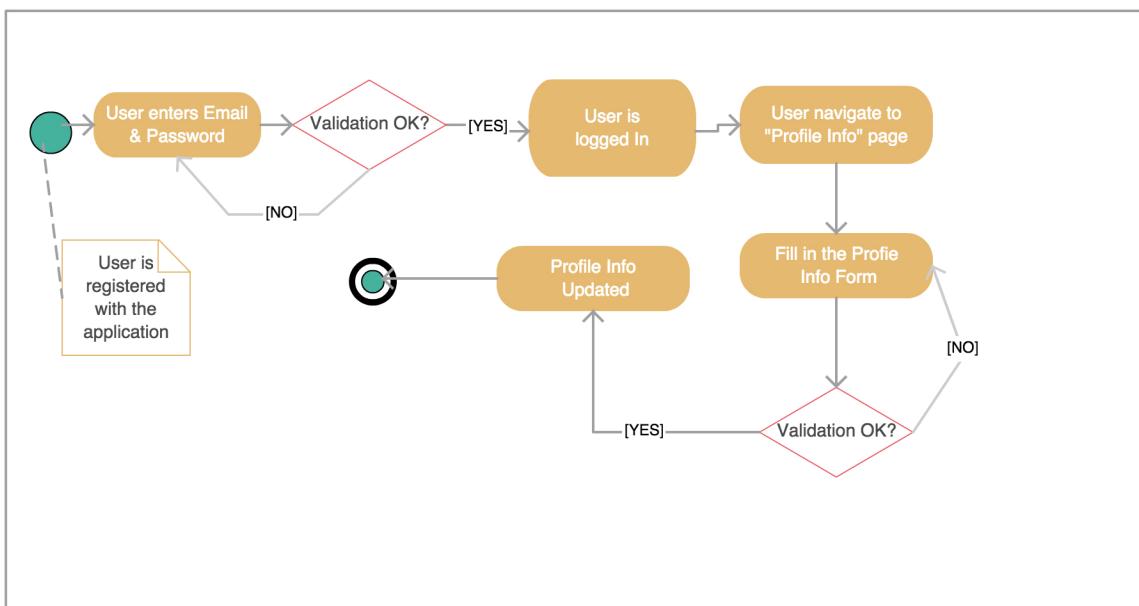


Figure 4.8: Edit User Details activity diagram.

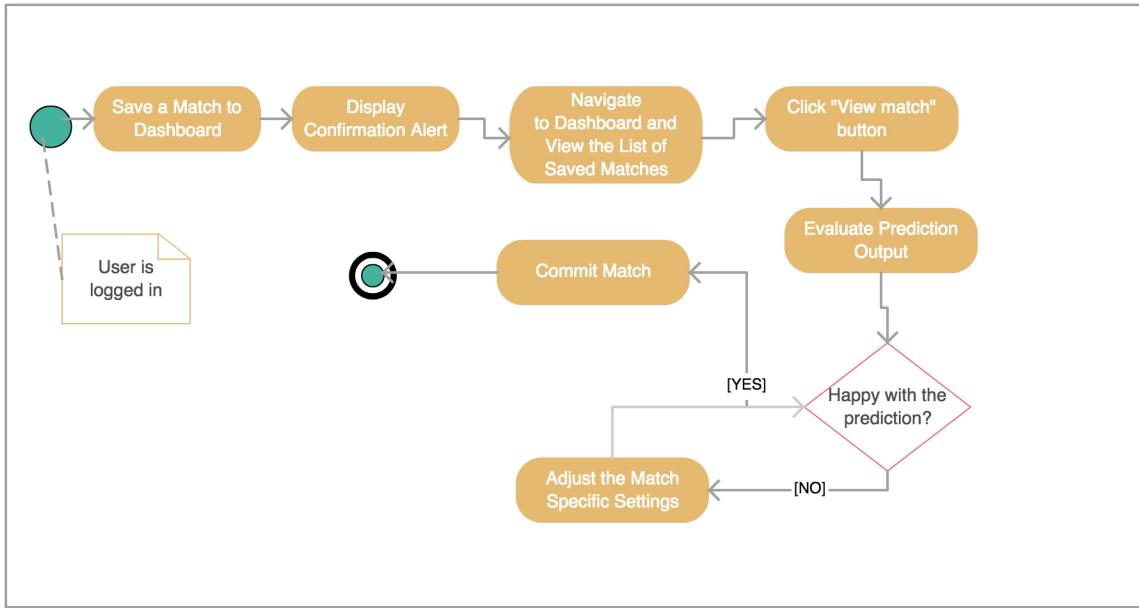


Figure 4.9: Bet on Match activity diagram.

4.7 Database Schema

The application requires a database. The data to be stored there will be represented by a set of database models. The schema for this project was developed in an iterative way along with the development of the high level features of the application - the tables and relationships between them were added gradually. The final, merged version of the diagram is a result of numerous iterations and can be found below.

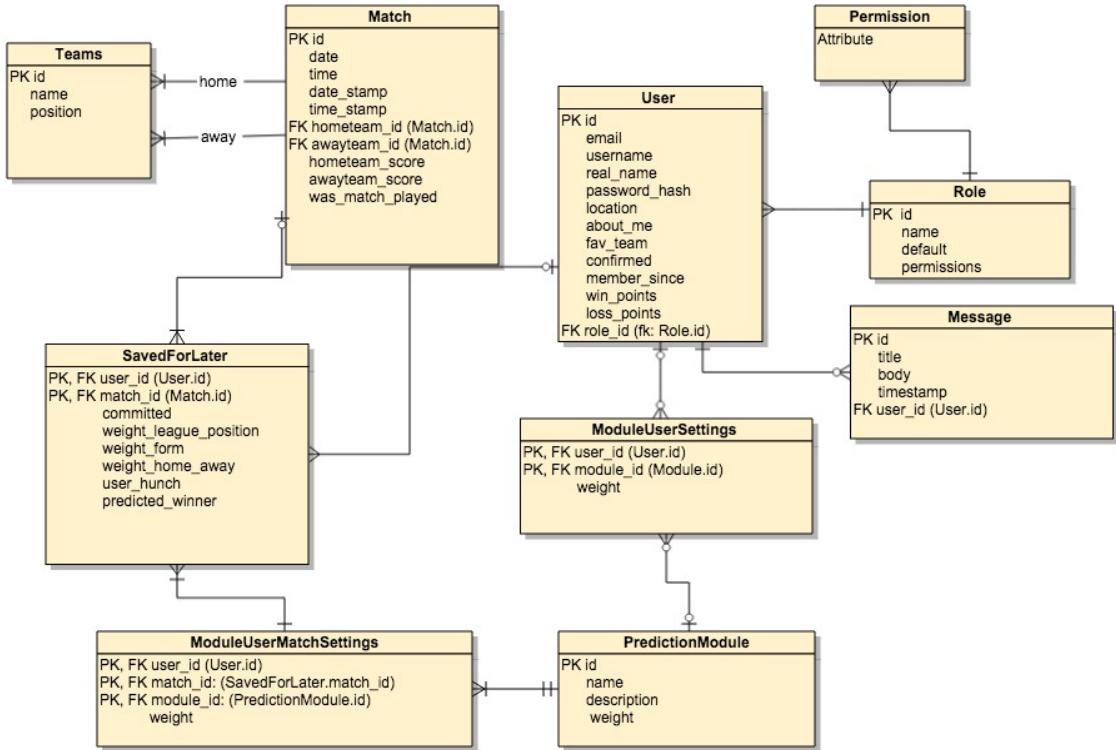


Figure 4.10: A final diagram representing tables in the database.

Chapter 5

Implementation

The aim of this project was to create a working web application that also makes use of the best available development practices, has well-designed architecture and is easy to maintain and extend in the future. The application is a relatively small-scale one, but it was developed with a future large-scale application in mind, that would support a large number of concurrent requests and stay highly responsive. Therefore, great emphasis was put on the scalability and performance.

This chapter examines the process of the application development.

5.1 Choice of Technologies

This section will show the justification for each particular technology the application rely on.

5.1.1 Front End

The markup of the future application will be coded using HTML 5 [57]. The application markup will be built using BEM front end development approach [63]. BEM (short for "Block Element Modifier") is a popular semantic model for markup and a way to organise sections of a website into purposeful blocks and to optimise CSS. The idea behind it is to logically break the HTML down into *independent* blocks, which will allow arbitrary placement of the block anywhere on the page, including nesting the block inside another block. The approach can be very beneficial for large websites, allowing the code to be reused across pages or even projects. However, a small project like the SureThing can also benefit from BEM by making use of independent, context-free CSS that can be easily amended in the future [47].

CSS3 is used to define the visual presentation of the application. In general, CSS has certain limitations of its syntax capabilities. For example, it does not allow the use of variables, macros, mixins (reusable blocks of styles) functions and other features associated with object-oriented development, which inevitably leads to the creation of immensely repetitive stylesheets. In order to overcome these limitations, SASS preprocessor [19] will be used in this project. SASS (short for Syntactically Awesome Stylesheets) is a powerful language that extends CSS with a choice of useful functionality, all in CSS-compatible syntax. Use of SASS would allow to make CSS code more efficient and easily maintainable.

In addition to that, SureThing will make use of a popular CSS framework, Bootstrap 3 [54]. Bootstrap provides a number of ready solutions for designing the layout of the future application. Therefore, the overall architecture of the markup will be defined by identifying BEM blocks and elements. This would bring structure into the code across all front end technologies used during the development process. BEM blocks and elements will be complemented with appropriate Bootstrap classes in order to speed up the development process and make the application fully responsive.

JavaScript, specifically JQuery library [44], will be utilised to add animations and improve the overall user experience.

In order to handle time-consuming and repetitive tasks on the front end side, the application will utilise the task-based command-line tool Grunt. This software comes with a variety of plugins serving different purposes. For this project will be used *grunt-sass* to compile SASS stylesheets into CSS complemented with *grunt-watch* to allow continuous development, *grunt-css* plugin to combine all the external CSS files into one and *grunt-uglify* plugin in order to reduce the size of JavaScript files and speed up loading of the web page in a browser. This is a screenshot of grunt output for this project running in terminal window.

```

localhost: ~/PycharmProjects/surething
→ grunt watch
Running "watch" task
Waiting...
>> File "app/static/scss/_base.scss" changed.
Running "sass:dist" (sass) task
File "app/static/css/main.css" created.

Running "sass:dev" (sass) task
File "app/static/css/main.css" created.

Running "concat:target" (concat) task
File app/static/cssall/main.css created.

Done, without errors.

Execution Time (2015-02-24 10:53:34 UTC)
Loading tasks 750ms [██████████] 41%
sass:dist | Elem 676ms Net [██████████] 39% Edits Console
sass:dev lead...</h3> 370ms [██████████] 21%
Total 1.8s class="home" id="surething" gram_dict="true" cz-shortcut-listen="true"
<!-- navbar -->

```

Figure 5.1: A screenshot from the terminal output running Grunt.

In addition, RequireJS [53], a powerful asynchronous script loader will be used for effective management of JavaScript dependencies. It can load modules in asynchronous manner if desired and thus improve overall website performance.

5.1.2 Back End

The application requires the latest football data to be able to make an accurate prediction. Live data has to be frequently loaded into the system and processed in an appropriate way. Therefore, there would be a need for at least one separate module dealing with a third party football data API and containing business logic to manipulate the received data. The API wrapper is expected to be integrated into the web application, but separated from the presentation, it also has to be relatively easy to execute as a standalone module, encouraging a nicely decoupled design.

Based on the above assumptions, Python was chosen as the primary back end language. The language is known for being well suited for data manipulation and analysis, and is therefore used in scientific computing and other highly quantitative domains, for example physics and finance. First of all, Python is a very effective language for data processing. When processing a large amount of data, memory use management becomes very important. Python provides optimised syntax for processing a sequence, such as generator expressions. Generators enable iteration over a sequence without placing a new list, set, or dictionary into the system memory [39]. Secondly, it is known for being easy to learn and use. This will hopefully provide an increase to the speed of development, leaving plenty of time for identifying the bottlenecks and optimising the code.

The back end of the web application will be built using Python web framework Flask [45]. It is a lightweight framework (the official name is "Python microframework") with a great choice of third-party libraries (e.g. Flask-SQLAlchemy or Flask-Login) that can extend the feature set of the framework core in various ways. Flask application is minimalist to begin with, but it can grow with the project needs. For the purpose of this project this is an advantage compared to the full-featured frameworks like Django that have a lot of functionality already built-in to the basic installation. In addition, the availability of developer-friendly documentation and an easy learning curve makes Flask a quick way to get a simple, Python-powered web site up and running. Therefore, Flask appears to be a great choice for a small project like SureThing.

SQLAlchemy was chosen as database solution for this project [1]. This is a powerful database framework that supports several databases back ends and offers the high-level Object Relational Mapper (for short, ORM). Using ORM provides a great level of abstraction when working with databases. For example, SQLAlchemy uses classes that map to each table in a database. This means that the records interaction can be kept the same regardless of the underlying database system. This offers

a lot of flexibility and allows the use of different database systems for development and production environment. According to Grindberg [17], "Flask-Migrate extension, based on a migration framework Alembic and written by a lead developer of SQLAlchemy, provides a powerful solution to handle database alterations and make database schema updates easily manageable"

5.2 Application Architecture

Application architecture is the foundation of good quality software. The architecture of SureThing was dictated for the most part by the used framework Flask, which uses a variation of MVC for Python called "MTC" (Model-Template-Controller). Florestan [9] in his blog post describes this pattern in the following way:

"The template contains HTML content and presentation logic. It is written in a templating language ... It gets data from the view and outputs a web page. The view (also sometimes called "controller"), written in Python, is just glue code. It uses the web framework to put everything together. The model layer is essentially a persistence layer: its most important dependency is SQLAlchemy. The model knows how to save the data, constituting the most reusable code in the entire project."

However, on top of the standard MVC architecture, SureThing requires few extra components to manage the loading of the data from an external source. Hence, this is the final layout of the application architecture:

- **Model Layer.** Contains Python classes that represent database models and related logic. According to Florestan [9], model layer "represents the essence of [the] system without the details of a user interface."
- **View Layer.** Holds association between URL rules and view functions that are defined with a help of a module-level decorator `route()`. Below can be found an excerpt from the application code, a header of a view function that represents the dashboard page. When a browser requests `/dashboard` URL, the associated view function is called and the return value is sent back to the browser.

```

@main.route('/dashboard')
@login_required
def dashboard():
    saved_matches = current_user.list_matches()

    upcoming_matches = [s for s in saved_matches if not s.was_played]

    return render_template('main/dashboard.html',
                           saved_matches=upcoming_matches,
                           user=current_user,
                           title='Dashboard')

```

Figure 5.2: Dashboard view function.

- **Template Layer.** This is the mediator layer between an HTTP request and the application logic. It consists of a number of Jinja2 templates holding only presentation logic.
- **External Services Layer** [9]. Contains API wrapper class that accesses and manipulates live football data. In general, the functionality of the application can be further extended in many ways. In the future, Football-API might not be the only external source of data. The project might make use of another third-party API or even use a web scraping technique to extract data from other websites . Eventually, all additional modules related to the interaction with external sources of data will become part of this layer.
- **Threading.** The application requests live football data frequently. Loading the data is a costly I/O operation that may become a bottleneck unless performed asynchronously. The threading component of the application defines a class *DataUpdateThread* that takes care of writing the data to the server every 100 seconds. This task is performed in a separate thread.

During the development process, a lot of effort was put into keeping the Template Layer as thin as possible in order to reduce the loading time in the browser and improve the overall performance of the application.

5.3 Patterns And Conventions

Flask offers an excellent extendable core of functionality; its API is also very minimalistic and easy to understand. One of the main advantages of this framework is that it gives developer a lot of freedom to decide how to structure the application. As Wright [61] puts it: ” without patterns or conventions your applications will lose architectural integrity and be difficult to understand by others”. In this section a number of various patterns, conventions and tools used during the implementation phase will be described and explained.

5.3.1 Application Factory

The use of a factory pattern is crucial to a Flask application. For example, SureThing app defines various configurations to be used in different environments (development, testing, production). However, because the application instance is created in the global scope, there is no way to apply those configurations. The problem is that "by the time the script is running, the application instance has already been created, so it is already too late to make configuration changes" [17]. To get around this problem the creation of the application was moved into a separate function, `create_app()`. The configuration name is passed into the function as a parameter. This solution also allows us to create multiple instances of the application and make testing of various configurations easier [46].

```
def create_app(config_name):
    app = Flask(__name__)

    #loading configurations into the app
    app.config.from_object(config[config_name])
```

Figure 5.3: Function responsible for creating the application instance.

5.3.2 Blueprints

Blueprints are related to the View Layer introduced in the section Application Architecture 5.2. A large application is divided into smaller parts and each part is implemented with help of a blueprint. This concept helps to develop a *modular* web application. SureThing was divided into two parts: *main* and *auth*. *auth* holds the endpoints associated with the authentication and user profile related tasks, for example `login()`, `edit_profile()`. On the other hand *main* is in charge of the rest of the application. Notice how these different blueprints are registered on the application instance inside the `create_app()` function:

```
#attach routes and custom error pages here
from main import main as main_blueprint
app.register_blueprint(main_blueprint)

from .auth import auth as auth_blueprint
app.register_blueprint(auth_blueprint, url_prefix='/auth')
```

Figure 5.4: Blueprints registration.

5.3.3 Database Migrations

The database scheme for this project was designed in an iterative way. Models and the relationships between them were added gradually as the application expanded. Therefore, it was crucial to find a tool that allows for effortless updating of the database. To manage frequent database updates, the Alembic database migration tool that was developed specifically by Mike Bayer, the author of SQLAlchemy,

was used. The tool can be added to Flask as an external plugin, Flask-migrate. After installation and initial configuration of the plugin, it allows to migrate the database with two simple commands to be subsequently run in the terminal: *db upgrade* and *db migrate*. Alembic makes migration easier and prevents the developer from the necessity to delete and recreate the database each time there is a need for migration.

5.3.4 Exceptions

In general, it is considered a good practice to take advantage of the standard libraries of a programming language, in our case Python. First of all, it allows the developer to save time implementing a piece of functionality from scratch. Secondly, it makes it easier for other developers to read and maintain the code. Exceptions are built into Python at the language level. Using them will lead to cleaner code and will not have any impact on the performance. ”In a way, try blocks are like transactions. [The] catch has to leave ... program in a consistent state, no matter what happens in the try. For this reason it is good practice to start with a try- catch- finally statement when you are writing code that could throw exceptions ” Martin [31]. SureThing will make use of Exceptions in order to identify and manage failures when making an API call over HTTP. The FootballAPIWrapper class has a private method that calls the API and collects the data in a JSON format from the remote server: `_call_api(action=None, **kwargs)`. The method takes into account the possibility of errors occurring during code execution. When the program calls the API, either JSON data is returned or an Exception is thrown.

As it can be seen from the method definition above, one of the required parameters is *action* that is set to None, unless the value is passed in during the method call. Action is a string that needs to be added to the base url in order to indicate the set of data that is being accessed. Specifying the action is required by the API and the possible values of the parameters (actions) are: competition, standings, today, fixtures, commentaries. For example action *today* will return the matches scheduled today. The *_all_api* method will raise an Exception, if the action is not being supplied. Various exceptions are thrown when the program is attempting to connect to the remote server [5]. Python library *requests* is used to take care of these types of errors [43]. If the domain name does not resolve, the HTTP request will fail before we establish connection. In that case, the program will throw a *requests.exceptions.ConnectionError*. If the remote server is not functioning or the request is structured incorrectly, the server will respond with an bad response status code and the *_call_api* method will raise a *requests.exceptions.HTTPError*.

5.4 Other Implementation Processes

In this section a number of other implementation processes and tools will be discussed.

5.4.1 PEP8

5.4.2 PyLint

5.4.3 Version Control

Git version control system was used throughout the development process. Git is known for being a very useful tool for collaboration across teams of developers. However, it also has many benefits for a solo developer. For example, it helps to track changes and restore previous versions of a project, as well as view the code at any point in the past. The project codebase was uploaded to GitHub that is ”a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features... [It also] provides web-based graphical interface” [58]. For this project the GitHub issue tracker was used as a ”to-do list” to keep the record of tasks (”issues” in GitHub terminology) that needed to be completed in each agile iteration. Custom labels were used to distinguish different types of issues in the GitHub issue tracker, for example, ”performance”, ”design”, ”bug”, ”optional”, etc.

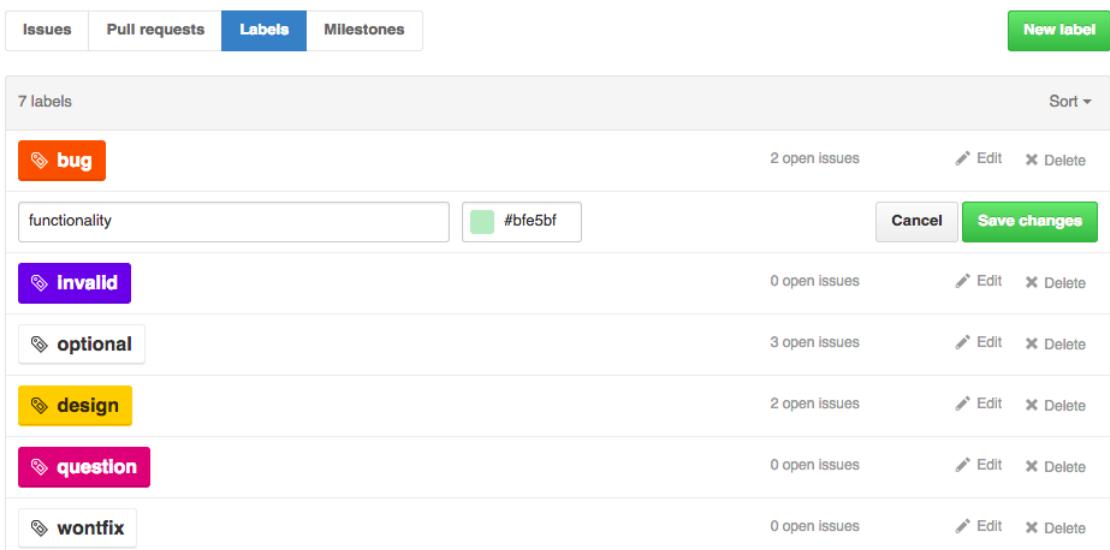


Figure 5.5: GitHub allows developers to add custom labels.

GitHub also allows the users to filter out the issues of a similar type, based on the assigned label.

	Author	Labels	Milestones	Assignee	Sort
<input type="checkbox"/> ① 5 Open ✓ 0 Closed					
<input type="checkbox"/> ④ Forgotten password feature functionality	#27 opened 11 minutes ago by marinamarina	User profile			0
<input type="checkbox"/> ④ More logic behind modules functionality	#26 opened 11 minutes ago by marinamarina				0
<input type="checkbox"/> ④ My profile functionality	#23 opened 12 minutes ago by marinamarina				0
<input type="checkbox"/> ④ Tipsters leaderboard functionality	#22 opened 13 minutes ago by marinamarina				0
<input type="checkbox"/> ④ View match played functionality	#21 opened 13 minutes ago by marinamarina				0

Figure 5.6: GitHub. Issues related to the functionality of the application.

Critically speaking, the option to assign issues various level of importance and order the issues based on their priority would have been a very useful feature to have.

5.4.4 Own Validation in Forms

Flask-WTF is a Flask extension that offers integration with WTForms and it was used to handle forms in this project. In order to make sure the application is secure, the validation has to be implemented preferably on the server side or on both the client- and server-side of the application. WTForms has many built-in validators that can simplify the developer's life. For example **DataRequired** makes the input field mandatory, **Email** checks that the provided input is a valid email address, **EqualTo** helps to ensure that the passwords in the fields "Password" and "Confirm Password" supplied during the user registration are identical. However, sometimes the built-in functionality does not cover all the application needs. In that case, there is an option to create a custom validator that is basically a Python function returning another function (a validator) that throws an exception every time the user violates the prescribed validation rule. Custom validators can be imported into the module describing forms and used in the same way as a built-in validator would be used. I have separated the validators out into a separate module. The set of custom validators can be further extended, however, there is just one at the moment: *validator_user_already_registered()*.

```

# Custom validators
def validator_user_already_registered():
    """custom validator used in the registration form"""

    def _user_already_registered(form, field):
        kwargs = {field.id: field.data}

        if User.query.filter_by(**kwargs).first():
            raise ValidationError(message='User with this ' + str(field.id) + ' is already registered!')

    return _user_already_registered

```

Figure 5.7: Validator function that checks if user with this username has already been registered with the SureThing application

In the example above the validator function checks if a user with provided username is already in the database. If the user is found, the `ValidationError` exception is thrown and the new user is prevented from submitting the form.

5.4.5 Custom Macros

Jinja2 is a default template engine that comes in one package with Flask. It is also one of the most widely used template engines for Python. In order to add some extra presentation logic to our application, custom macros can be used. Jinja2 macro is simply a template function that can be used within HTML in order to avoid developers writing repetitive code. For this project I found macros extremely useful. Custom macros were separated out into a separate template file `_macros.html`.

One of the uses of this was rendering form fields. Each form field related macro contained a piece of HTML code specifically designed for forms in this application, as well as the logic for displaying error messages. Among these types of macros are `render_field(field)`, `render_checkbox(field)`, `render_submit_field(field)`. Some of those macros needed to be adjusted to enable their usage in a specific view. These are the examples of such "adjusted" macros: `render_submit_field_match_preview(field)`, `render_embedded_field(field)`. For example, `render_field` takes care of rendering any standard input field across the application, whereas `render_embedded_field(field)` manages rendering an input field embedded within a prediction module on the Upcoming Match View.

Another macro, `teamkitimage(match, home=1)` renders an image representing a football club. Based on the provided arguments, the function displays an image of a home or away team kit for a specific club.

5.4.6 Integration with third-party API

Many production Python web applications rely on external application programming interfaces (APIs). API can be also referred to as "third party services" or "external platform" [30]. SureThing requires

constant access to current football data. After choosing an appropriate API, it has to be integrated into the application.

There is a variety of tools available for developers for accessing web APIs. These three options were considered when choosing an appropriate tool:

- Helper library (such as Runscope or Apiary) Using a helper library has an overhead of learning how to use another piece of software.
- urllib2, standard Python module *urllib2* module offers very simple implementation and provides most of the required HTTP capabilities, but the API is thoroughly broken and features critical for performance are missing, for example connection re-using/pooling.
- urllib3
- requests, another Python library for handling HTTP requests. It offers a lot of control over the HTTP calls through the use of its powerful features.

After some experiments with other urllib2, urllib3 and requests, *requests* was chosen as the library for this project.

All interaction with the Football-API, including processing the received data, was separated out into a module *football_api_wrapper.py* or just "wrapper" for short. This module contains only one class, FootballAPIWrapper. Fields of the class accommodate the key elements of the interaction with the API that will be re-used in different methods of the wrapper, for example: base url, path to the data directory.

```
def __init__(self):  
    self._premier_league_id = '1204'  
    self._base_url = 'http://football-api.com/api/?Action='  
    self._basedir = os.path.dirname(__file__)  
    self._datadir = os.path.abspath(os.path.join(self._basedir, '..', 'data'))  
    self._proxy_on = False
```

Figure 5.8: Football API wrapper, fields

To get the response from Football-API takes about 11s, therefore it is necessary to move the API calls into a task queue so that they do not block the HTTP request-response cycle for the rest of the web application.

5.4.7 Visual Effects

A number of visual effects were implemented with the help of jQuery, AJAX and the Websockets (Flask extension Flask-SocketsIO) to improve the user experience.

Alerts

A good example is an alert dismissal. SureThing generates many alerts that give the user feedback with regards to the action they have just taken. User has to dismiss the alert manually each time. To avoid this extra thing for a user to do, the alerts are removed from the page using jQuery fadeOut() and then slideUp() animation methods that fades the pop-up and then removes it with a sliding motion after 200ms of its appearance on the page. For compactness the screenshot shows the page opened in a browser simulation of an iPhone screen size. Responsive web design with regards to this project will be examined in more detail in the following subsection.

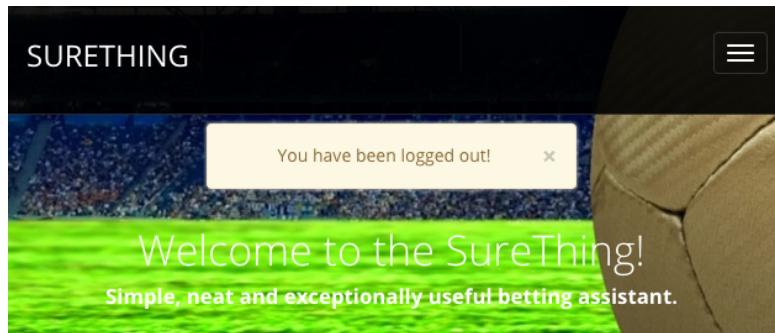


Figure 5.9: An example of an alert on the page appearing after user has logged out

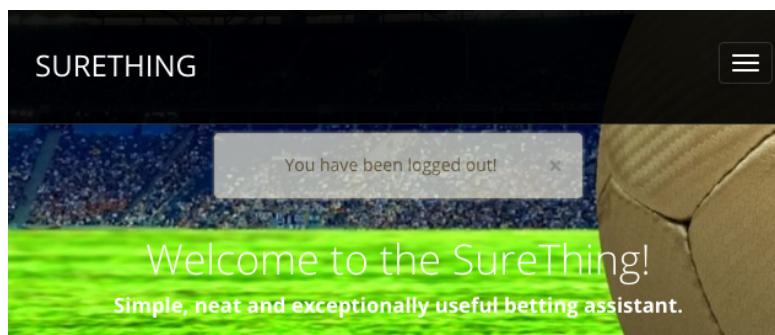


Figure 5.10: The same alert fading out.

New message notifications

Another example is a new message notification. Once the user receives a new in-app message notifying them about the results of their bets, the envelope-shaped icon on the top navigation menu that represents the in-app Inbox turns orange. When the user navigates to the Inbox and reads or deletes

all the new messages, the icon immediately turns grey, indicating that there are no more unread messages in the box.

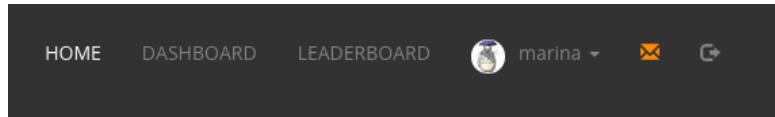


Figure 5.11: You have new messages. Desktop View.

This is what the same top navigation menu looks like for mobile users.

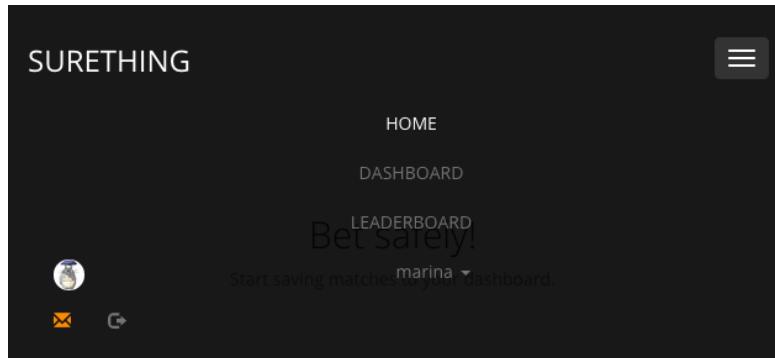


Figure 5.12: You have new messages. Mobile View.

The user has opened the last unread message and the message icon has turned grey. Notice, how the top navigation menu partially covers the message view. The menu rolls back into a compact mode once the user clicks on the menu icon in the top right corner.

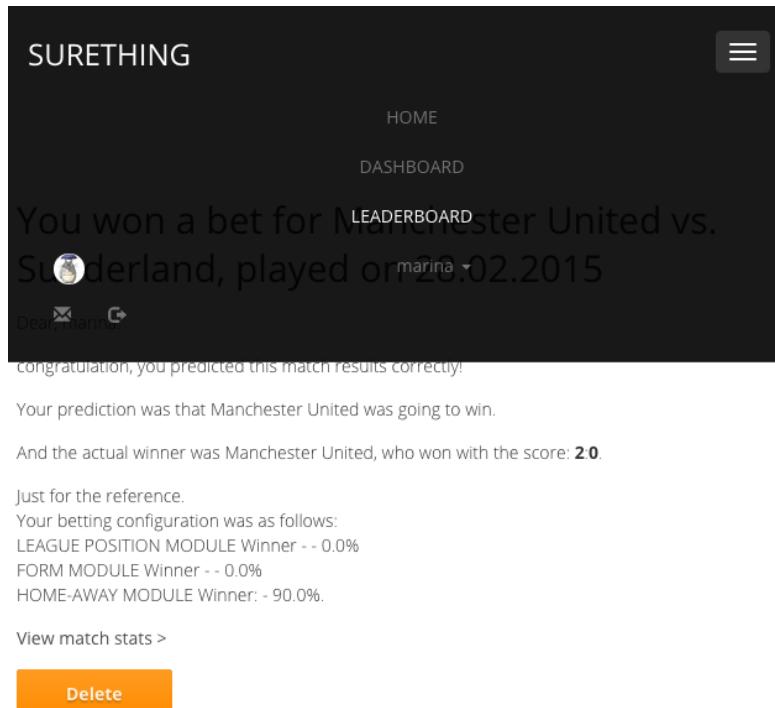


Figure 5.13: The user just read the last new message, the icon turns grey.

5.4.8 Responsive Design

According to the popular portal Statista, "As of 2013, worldwide mobile phone internet user penetration was 73.4 percent. In 2017, figures suggest that more than 90 percent of internet users will access online content through their phones" [22]. There is no doubt that developers need to adapt to the increasing combinations of screen resolution and browsers used by people to access information online. The solution to the expanding variability of the web is to develop a layout that can adapt to any viewport. This approach is known as *responsive web design*. The term was first used by Ethan Marcotte. He combined three already known techniques (flexible grid layout, flexible images, and media and media queries) into a unified approach [14].

Based on the above, it should be assumed that the majority of users will access the website through a device that is not a desktop computer. We need a good fluid grid to build a responsive web application. To handle the responsiveness and to increase development speed SureThing utilises Bootstrap framework Twitter [54]. that offers an responsive fluid grid system that can adjust to the variety of devices or screen sizes. The framework has also predefined classes that can be used to change the page layout options, for example to specify how many columns in the grid system an element will occupy or to set the breakpoints at which the columns stacked on small devices will become horizontal on medium/large devices.

SureThing is a fully responsive web application, and the minimum screen resolution supported is 640 x 960 pixels (an example device using this resolution would be the iPhone 4). These are the screenshots of some of the application pages tested with this resolution:

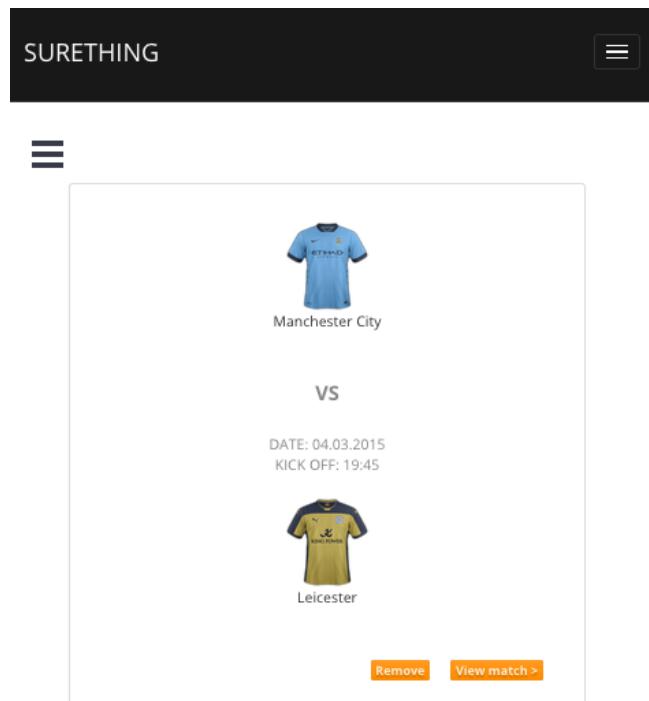


Figure 5.14: Dashboard view, DVGA screen size.

Prediction Modules

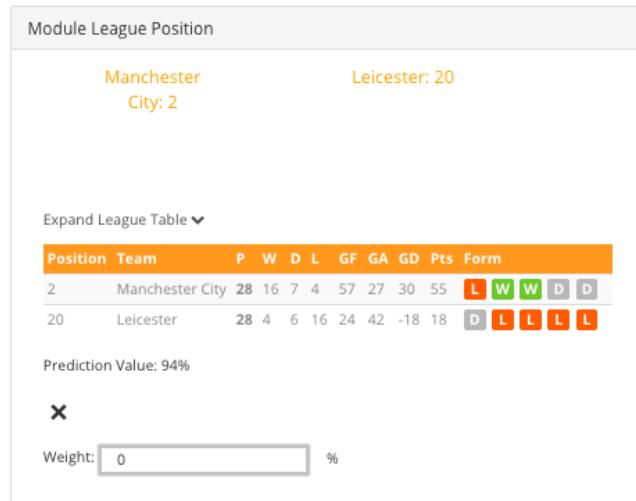


Figure 5.15: Module League Position in the Upcoming Match View, DVGA screen size.

5.5 Features Implementation

In this section the technical details of the project implementation will be described. Each subsection is bound to a high level feature of the application, as introduced in the chapter "Requirements Analysis" 3. The "user journey", or the possible user interactions with the system, will be demonstrated for each view.

5.5.1 Authentication and User Profile

Authentication

The application requires authentication functionality. In order to simplify the development process, a useful Flask extension, Flask-Login, was utilised to handle the common tasks of logging in and out, as well as for new users registration. For new users, the application offers a registration form. On form submission, SureThing sends the user an email with a verification token, expecting them to confirm the email address and complete the registration. It should be noted that both username and email address provided during the registration should be unique, otherwise the application throws a validation error. The screenshot below is an example of what happens when a new user tries to register using an email address that is already in the database:

Register

The screenshot shows a registration form titled "Create your SureThing Account". The "Email" field contains "shchukina.marina@gmail.com", which is highlighted in yellow with a red border, indicating it is a required field. Below the field, a red error message reads "User with this email is already registered!". The "Username" field contains "marina", and the "Password" and "Confirm Password" fields are empty. At the bottom right is an orange "Register" button.

Figure 5.16: SureThing offers registration form for the new users. However, the email address is expected to be unique.

Once a user has created a SureThing account, they can login using a standard login form. An email address and a password are both required fields. A validation error will be thrown if the provided email or password are invalid, or the user with the provided email address was not found in the database.

Gravatar is an abbreviation for "Globally recognized avatar" and it is one of the most popular avatar services. User can register with the service (<http://gravatar.com>) and upload an image that can be used as avatars across many popular websites, such as GitHub (<http://www.github.com>), Stackoverflow (<http://www.stackoverflow.com>) and WordPress (<http://www.wordpress.com>). Gravatars are integrated into the SureThing, and a newly registered user does not have to upload an avatar image to be used in our application. The app will access the avatar associated with the user's email address and pull it from the Gravatar servers. An authorised user will see the icon with their avatar picture in the top right corner of the application navigation menu, as can be seen in the screenshot below:

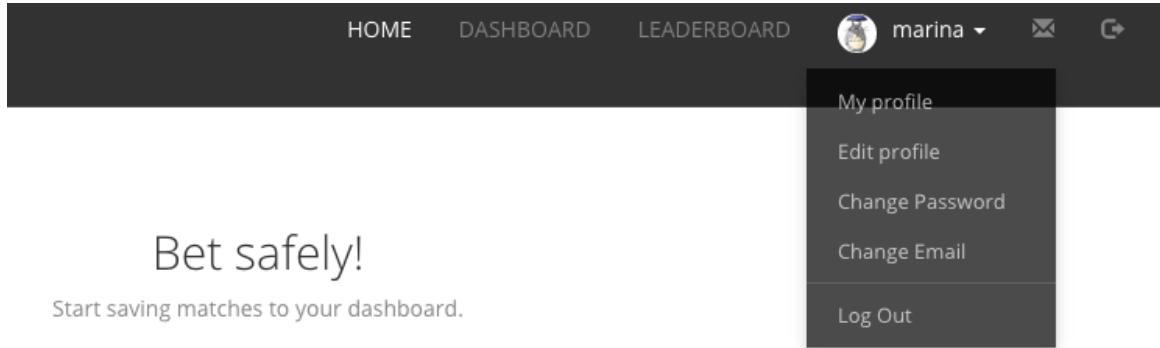


Figure 5.17: User settings and the profile page can be accessed by clicking on the avatar icon located on the application navigation menu panel.

In case the user is not registered with the Gravatar, SureThing will generate a dummy avatar to be used in the application.

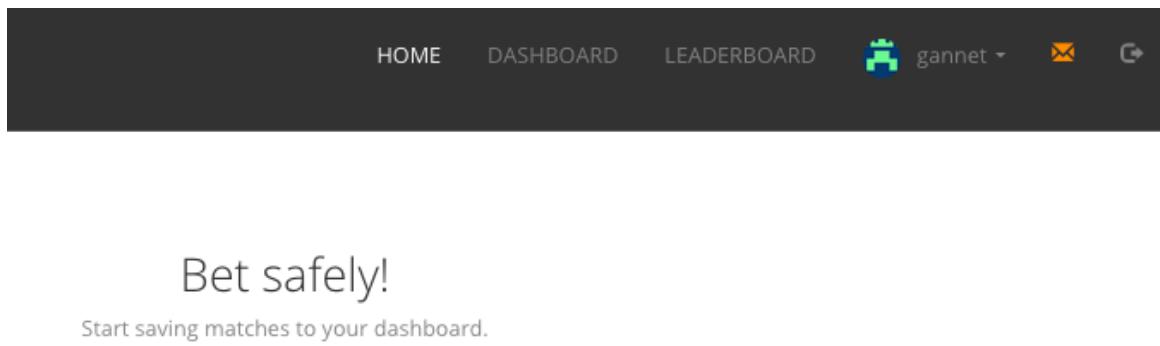


Figure 5.18: An example of a dummy avatar for user **gannet**, who is not registered with Gravatar services.

User Profile

The options available in the dropdown menu, as displayed in the figure 5.17, enable the users to edit their profiles and change their password or email address.

Change password

The screenshot shows a 'Password Change' form. It contains three input fields: 'Old Password *' (with a placeholder 'Old Password'), 'Password *' (with a placeholder 'New Password'), and 'Re-enter Password *' (with a placeholder 'Re-enter New Password'). Below the fields is an orange 'Submit' button.

Figure 5.19: Users can change their passwords for security reasons.

Tell us about yourself

The screenshot shows an 'Edit your profile' form. It includes four input fields: 'Real name *' (with a placeholder 'Real Name'), 'Location *' (with a placeholder 'Location'), 'Fav team *' (with a placeholder 'Fav Team'), and 'About me *' (with a placeholder 'About Me'). Below the fields is an orange 'Submit' button.

Figure 5.20: After registration, users can share more information about themselves by editing their profile.

Clicking on the dropdown option "Profile" will take the user to a separate page containing all the information about the user, for example: profile information, preferences, section "about me", recent won bets including the prediction weights used for those bets, etc.

marina

Edit profile



Profile

Member Since 12/21/2014

Last seen 3 months ago

Full name Marina Shchukina

Location: Old Aberdeen

Preferences

Fav team Chelsea

Activity

Games bet on 87

Bets won 66

Bets lost 21

Followers XXX

Social Media

About marina

I am an RGU student

Recent bets I won

2:1

Swansea vs. Manchester United
Played: 21.02.2015
At: 15:00

My prediction was:
Manchester United

Weights used:
League position: 0%
Form: 0%
Home/Away: 90%

2:2

Tottenham vs. West Ham
Played: 22.02.2015
At: 12:00

My prediction was:
Tottenham

Weights used:
League position: 0%
Form: 0%
Home/Away: 90%

2:0

Manchester United vs. Sunderland
Played: 28.02.2015
At: 15:00

My prediction was:
Manchester United

Weights used:
League position: 0%
Form: 0%
Home/Away: 90%

marina's comments

To be added...

Figure 5.21: User profile page.

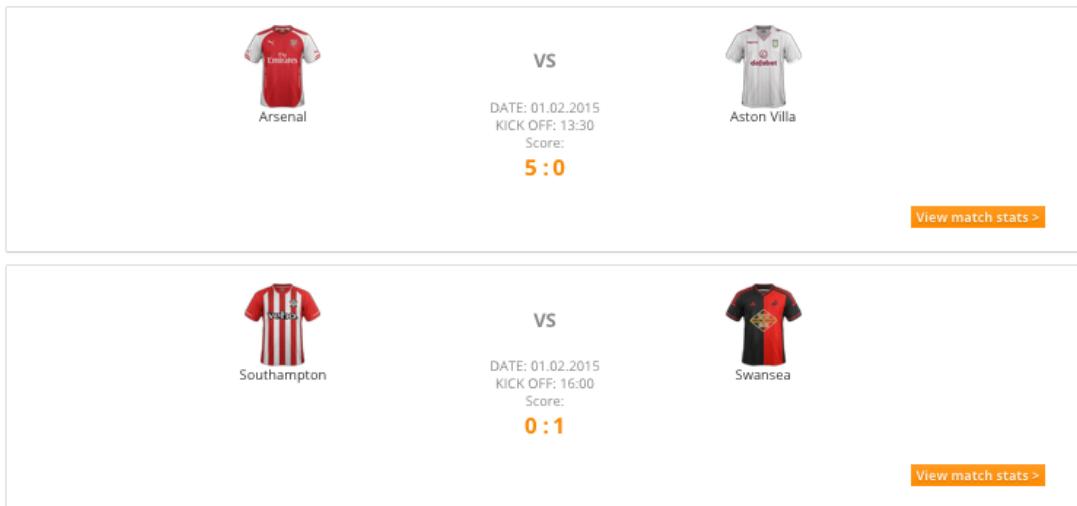
User Journey

From the User Profile page, the user can navigate to the "Edit Profile" form by clicking on the light blue "Edit Profile" button located in the top left corner of the profile page.

5.5.2 Matches Overview

Matches Overview is a view displayed on the main page of the application. It contains lists of upcoming and played matches and the user can toggle between these two lists using navigation buttons. In both lists, the matches are grouped by dates as follows:

February 1st 2015



January 31st 2015

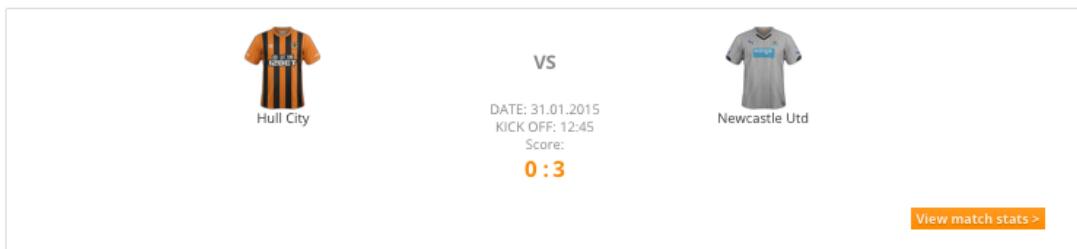


Figure 5.22: Matches in the overview are grouped by dates sorted in ascending order.

Each panel representing a match contains the most basic information about the event, such as the names of participating teams, the date and the kick-off time. The screenshot below is an example of a panel displaying an unplayed match. The navigation buttons can be seen just above the first match in the list. If the match is already saved to the Dashboard by the authenticated user, it is indicated by a floppy disk icon on the right hand side of the panel.

Premier League Fixtures & Results

[Upcoming fixtures](#) [Previous Results](#)

March 4th 2015

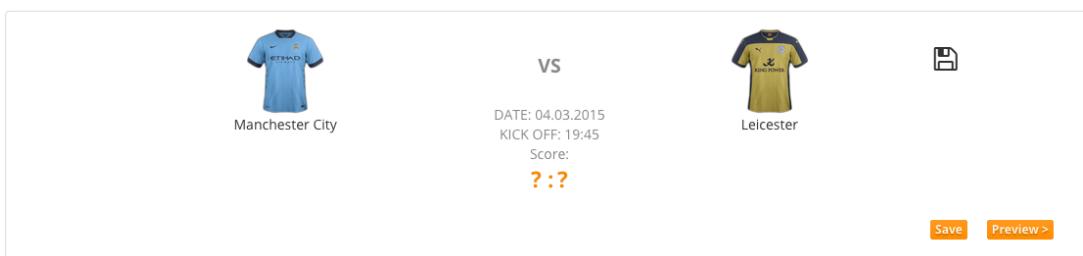


Figure 5.23: Example of an unplayed match displayed in the overview.

Below can be found a screenshot of a played match panel. Notice that the final score of the match is

displayed and instead of two buttons "Save" and "Preview" there is only one button - "View match stats".

February 1st 2015

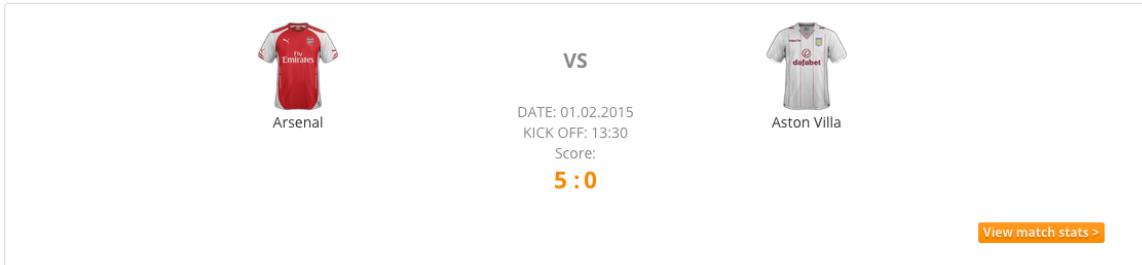


Figure 5.24: Example of a played match displayed in the overview.

If the match is being played at that very moment, it still belongs to the list of "unplayed" matches and is displayed with a small badge "LIVE", indicating a live event. A match is considered as "played" as soon as the full time score is available. Hence, it is possible to make "bets" until the moment the match is considered "played".

February 8th 2015

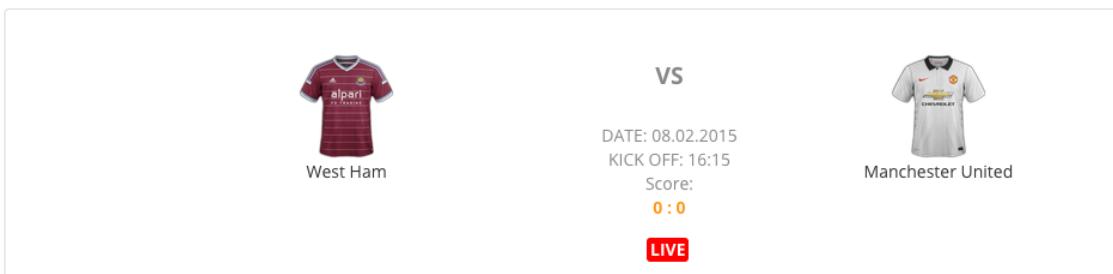


Figure 5.25: Example of a live match displayed in the overview.

User Journey

Just above the list, simple navigation buttons allowing the user to switch between the lists of unplayed and played matches is found. On the right hand side of each list item there is a "Preview" button for an upcoming match and a "View match stats" button for a played match. By clicking those buttons the user can navigate to views with more detailed information about the match (*Upcoming Match View* or *Played Match View*). User can save the match to the dashboard by clicking the "Save" button. This action can be carried out only for an unplayed match.

5.5.3 Upcoming Match View

Implementation of this view was one of the most complex development tasks of the whole project. This is the essence of SureThing - the view allowing the user to predict match results.

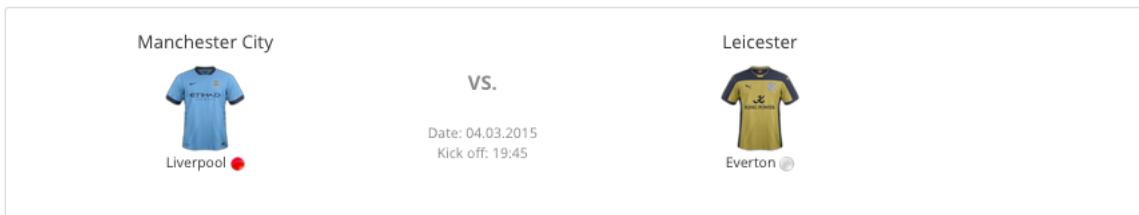
An authenticated SureThing user can navigate to this view either by clicking a "Preview" button on an upcoming match panel in the *Matches Overview* or by clicking the same button on a saved match panel in the user *Dashboard* (if the match has already been saved by the user). Depending on the user route to this view, the Upcoming Match Preview will be displayed differently.

Read-only mode

If the user is coming to the Upcoming Match Preview from the *main page*, the view will display the match header (containing general information about the teams, last played game, match kick-off time and date, etc.) and a list of prediction modules with **prediction values** calculated based on the relevant piece of statistics for each of the teams. These are the prediction modules available in this view:

1. Module League Position
2. Module Form
3. Module Home/Away

The "Save" button can be found at the very bottom of the view.



Prediction Modules

Module League Position

Manchester City: 2		Leicester: 20		Prediction Value: 94%						
Position	Team	P	W	D	L	GF	GA	GD	Pts	Form
2	Manchester City	28	16	7	4	57	27	30	55	L W W D D
20	Leicester	28	4	6	16	24	42	-18	18	D L L L L

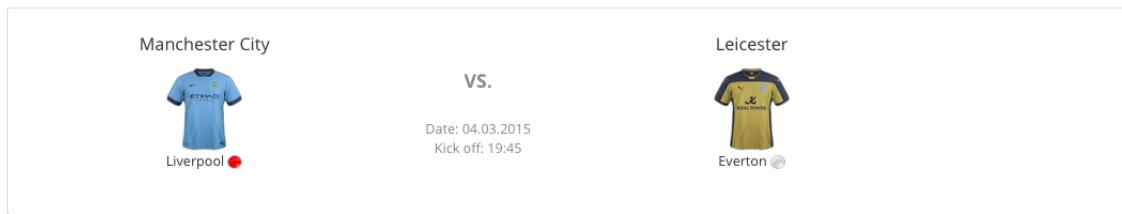
Expand League Table ▾

Figure 5.26: Upcoming Match View in the "read-only" mode: match header and the first prediction module, Module League Position.

This information should be sufficient for the user to decide whether it is worth saving the match to the dashboard for a later revision. An unauthorised user would still be able to see the same content, but the "Save" button will be disabled. If the user navigates to the Upcoming Match View from the main page, they can see the view in the **read-only mode**. It is also important to note that this is one of the few views that are available for an unauthorised user.

Prediction mode

If the user has already saved the match to the dashboard and navigates to the Upcoming Match View by clicking on a saved match panel, the view will enable the prediction feature. This time the view is displayed in the **prediction mode**. In each of the prediction modules, the user will be able to see an embedded input field with weight percentages inside the field. Below can be found a screenshot of the same part of the view as the one above, displayed in prediction mode.



Prediction Modules

Module League Position																																												
Manchester City: 2						Leicester: 20																																						
Prediction Value: 94% X																																												
Weight: <input type="text" value="0"/> %																																												
Expand League Table ▾ <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Position</th> <th>Team</th> <th>P</th> <th>W</th> <th>D</th> <th>L</th> <th>GF</th> <th>GA</th> <th>GD</th> <th>Pts</th> <th>Form</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Manchester City</td> <td>28</td> <td>16</td> <td>7</td> <td>4</td> <td>57</td> <td>27</td> <td>30</td> <td>55</td> <td>L W W D D</td> </tr> <tr> <td>20</td> <td>Leicester</td> <td>28</td> <td>4</td> <td>6</td> <td>16</td> <td>24</td> <td>42</td> <td>-18</td> <td>18</td> <td>D L L L L</td> </tr> </tbody> </table>												Position	Team	P	W	D	L	GF	GA	GD	Pts	Form	2	Manchester City	28	16	7	4	57	27	30	55	L W W D D	20	Leicester	28	4	6	16	24	42	-18	18	D L
Position	Team	P	W	D	L	GF	GA	GD	Pts	Form																																		
2	Manchester City	28	16	7	4	57	27	30	55	L W W D D																																		
20	Leicester	28	4	6	16	24	42	-18	18	D L																																		

Figure 5.27: Upcoming Match View in the "prediction" mode: match header and the first prediction module, Module League Position.

The values displayed inside the fields will be used for calculating the overall match result prediction. If this is the first time the user previews the match, the prediction weights will be either the **system default** (in case the user has not set the prediction settings in the dashboard yet) or the **user default** prediction settings. If the user has already visited this page before and set the match specific settings, the values displayed inside the input fields will be the **match specific** ones. Any module can be eliminated from the prediction by setting its weight to 0%. The total sum of prediction weights must equal 100%.

These are the prediction modules available to the user in the prediction mode:

1. Module League Position
2. Module Form
3. Module Home/Away
4. User Hunch

Notice the extra module is available in this view - User Hunch. This module is very important to the prediction process. The user can personalise the prediction by choosing Home, Away or Draw value in the User Hunch module panel. The module will be explained in more detail in the following subsection, "Prediction" [5.5.4](#)

User hunch

Hometeam Draw Awayteam

Prediction Value: 0.0%

Weight: %

*Please, trust your hunch and choose your value. You do not have to set any value for this module. If you would like the prediction for the match to be 'Draw', please set user hunch value to 'Draw' and set the prediction percentage to 100 and the other modules' prediction percentage to 0.

Figure 5.28: Module User Hunch.

The overall prediction is displayed in a separate panel underneath the modules.

Predicted Match Winner

Predicted result for this match is: Draw

Based on your match specific prediction settings, total probability is: 0.0%

*Please, be advised that probability of one of the teams to win can be maximum 100% (The higher the value, the higher is the probability). If the total probability is equal to 0%, it means that the probability of either of teams to win is equal to 0%, therefore the prediction is **Draw**.

Update Commit

Figure 5.29: Match result prediction.

User journey

In the **read-only** mode, the user can save the match to the dashboard by clicking the "Save" button.

In the **prediction** mode the user can update the prediction settings by overriding the current values in each of the input fields and pressing the button "Update" that is located at the bottom of the view. The overall prediction output will change every time the user updates the prediction settings or changes the hunch value. Once satisfied with the prediction outcome, the user can commit the match by pressing the "Commit" button that is located next to "Update". Before the end of the match, the user can still navigate to the committed match view and see all the details, including the prediction values, used weights and the final prediction. However, this time "Update" and "Commit" buttons will be disabled.

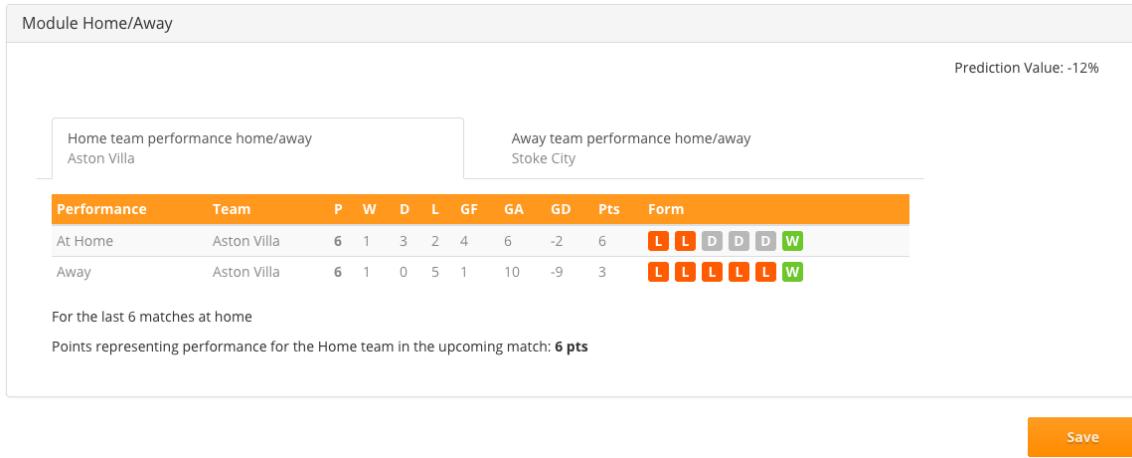


Figure 5.30: An example of a prediction module in the Match Preview, user navigated from the main page.

If the user comes to the upcoming match preview from the *dashboard*, they will be able to see more information related to the actual result prediction and betting. First of all, in each prediction module they will see input fields for setting match specific prediction weights. Secondly, they will see a user hunch module. Finally, at the very bottom of the overview they will see the calculated prediction result and two buttons - one to save the match specific settings and another to commit the bet.

5.5.4 Prediction

After getting familiar with the *Upcoming Match View*, the next logical step is to move on to the Prediction feature of the application. This will aid in understanding how to use the *Upcoming Match View* and what kind of information this view offers to an intended user. The implementation of the Prediction feature was already briefly outlined in the chapter "Requirements Analysis" 3, subsection "Prediction". This part of the report will illustrate the practical side of this feature and explain the several levels of prediction settings used in the application, the way the prediction values are calculated and the logic behind the prediction of the match result.

Three levels of prediction weights

SureThing has three levels of prediction settings or weights. Firstly, the *system default* prediction settings - a set of weights "recommended" to new users by the system. Once the user is registered with the application, they can set their own set of weights that will override the system default settings - *user default* prediction settings. This can be done through a "Prediction Settings" form, as explained in the subsection "Dashboard" 5.5.6. From the moment these weights are saved in the database, they will apply to every newly saved match. The application also enables setting *match specific* prediction

settings that will only apply to one match. The user can set match specific weights through the Upcoming Match view page in prediction mode [5.5.3](#).

The prediction of match result is expressed as a percentage and is calculated as a weighted average of a set of prediction values with different levels of relevance.

Prediction Modules

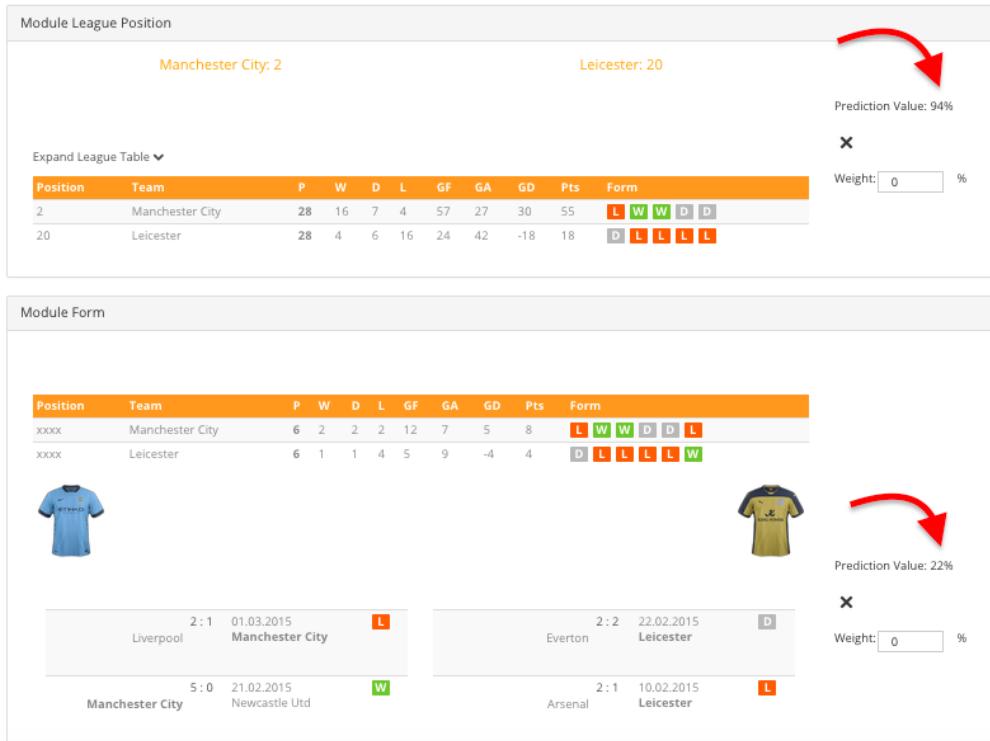


Figure 5.31: Prediction values can be found on the panels for each module.

How the prediction values are calculated

There are quite simple and logical equations behind each of the prediction values. Note that the `prediction_league_position()`, `prediction_form()` and `prediction_homeaway()` functions that hold the calculation code are implemented as properties of a class representing the database model *Match* and therefore can be easily accessed from any part of the code.

The League Position module compares positions in the league table for each of the teams. To calculate the prediction value behind League Position module we first "invert" the league position of each team by subtracting it from the total number of positions in the league, which is 20 for the English Premier League. The maximum "inverted" value that may be achieved by each team is 19 (as the top league position is 1). Next, we find the difference between the two obtained values and divide it by the maximum "inverted" value, which is again 19. Below can be found the equation and the code implementing it.

$$prediction_value = \frac{(20 - hometeam_position) - (20 - awayteam_position)}{19} \quad (5.1)$$

```

@property
def prediction_league_position(self):
    """calculate the winner for the league position prediction module
    ((20-homeposition)-(20-awayposition))/19
    if prediction value is positive, it increases the probability of hometeam to win
    if it is negative, it increases the probability of awayteam to win
    """
    hometeam_diff = 20 - int(self.hometeam.position)
    awayteam_diff = 20 - int(self.awayteam.position)
    prediction_value = (hometeam_diff-awayteam_diff) * 100 / 19

    return prediction_value

```

Figure 5.32: The calculation behind the prediction value for the League Position module.

The Form module takes into consideration a team's performance for the last 6 games. To calculate the prediction value for the Form module, we need to consider the difference between points achieved by the teams in the last 6 games (this value is taken directly from the league standings table). Obtained difference is then divided by 18, the maximum number of points a team can achieve in 6 games.

$$prediction_value = \frac{(hometeam_points) - (awayteam_points)}{18} \quad (5.2)$$

```

@property
def prediction_form(self):
    """calculate the winner for the form prediction module
    (hometeam points - awayteam points)/18
    18 is the maximum amount of points a team can achieve
    if prediction value is positive, it increases the probability of hometeam to win
    if it is negative, it increases the probability of awayteam to win
    """
    hometeam_pts = self.hometeam.form_last_6 pts
    awayteam_pts = self.awayteam.form_last_6 pts
    prediction_value = (hometeam_pts - awayteam_pts) * 100 / 18

    return prediction_value

```

Figure 5.33: The calculation behind the prediction value for the Form module.

For the Home/Away module we consider the last 6 matches that the home team played *at home* and that the away team played *away*. The equation behind the prediction value for the module Home/Away is similar to the one above. However, this time we compare points representing *home* performance for the home team and *away* performance for the away team.

$$prediction_value = \frac{(hometeam_home_points) - (awayteam_away_points)}{18} \quad (5.3)$$

```

@property
def prediction_homeaway(self):
    """calculate the winner for the home away module
    (hometeam at home points - awayteam away points)/18
    18 is the maximum amount of points a team can achieve
    if prediction value is positive, it increases the probability of hometeam to win
    if it is negative, it increases the probability of awayteam to win
    """
    # hometeam's performance at home (last 6 matches)
    hometeam_home_pts = self.hometeam.form_home_away.home pts

    # awayteam's performance away (last 6 matches)
    awayteam_away_pts = self.awayteam.form_home_away.away pts

    prediction_value = (hometeam_home_pts - awayteam_away_pts) * 100 / 18

    return prediction_value

```

Figure 5.34: The calculation behind the prediction value for the Home/Away module.

A simple rule of thumb works for all modules: a positive prediction value represents higher chances for the home team to win. The higher the value the higher is the probability of this to happen. On the other hand, a negative value means that the away team is more likely to win the match.

How the is the match result predicted

As already mentioned above, the overall match result is calculated as a weighted average of the prediction values. The application has only to decide which set of weights to use, depending on whether the user provided match specific or user specific settings. Below is an example of how this calculation is performed.

	Prediction values	User weights	Prediction result
Module league position	78%	20%	15.6%
Module form	44%	20%	8.8%
Module home/away	-14%	10%	-1.4%
Module user hunch	100%	50%	50%
Match result prediction			73.0%

Figure 5.35: The table illustrates the calculation behind the prediction of the match result.

As it can be seen in the table above, the League Position and Form modules' prediction values indicate that the home team has a strong chance to win the match, as the values are positive (78% and 44%). However, the negative value of the module Home/Away implies that there is 14% probability that the away team will be the winner. It is also obvious that the user is favouring the home team, as the user hunch prediction value is positive and equals to 100%. These values are averaged by the user default weights as follows:

$$match_result_prediction = \frac{78\% * 20\% + 44\% * 20\% - 14\% * 10\% + 100\% * 50\%}{100\%} \quad (5.4)$$

The overall result is **78%**. The high positive value of the obtained value means that the home team is very likely to win the match.

5.5.5 Played Match View

The user can navigate to this view by clicking the button "View match stats" on a panel representing a match that has already been played. Unlike the *Upcoming Match View*, the *Played Match View* looks the same to the users coming to this page both from their dashboards and from the matches overview page.

The view contains the already familiar match header, prediction statistics and a personalised feedback for the authenticated user.

[Home](#) / [Dashboard](#) / [Archived Matches](#) / Stoke City VS Manchester City

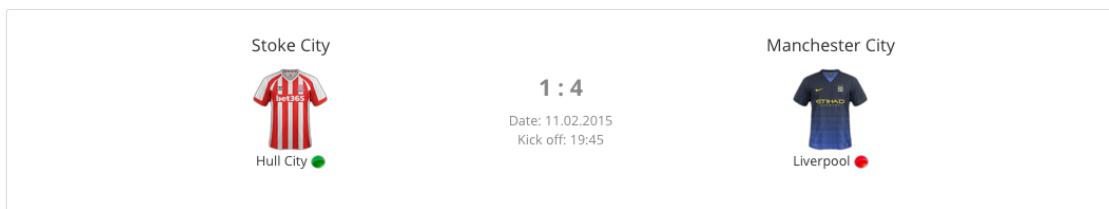


Figure 5.36: Played Match View, match header.

The prediction statistics block contains information of the betting performance across the SureThing users population with regards to this match. The stats contain the information on the number of users who saved the match to their dashboards, made a bet on the match and if they won or lost the bet.

Users' Prediction Stats

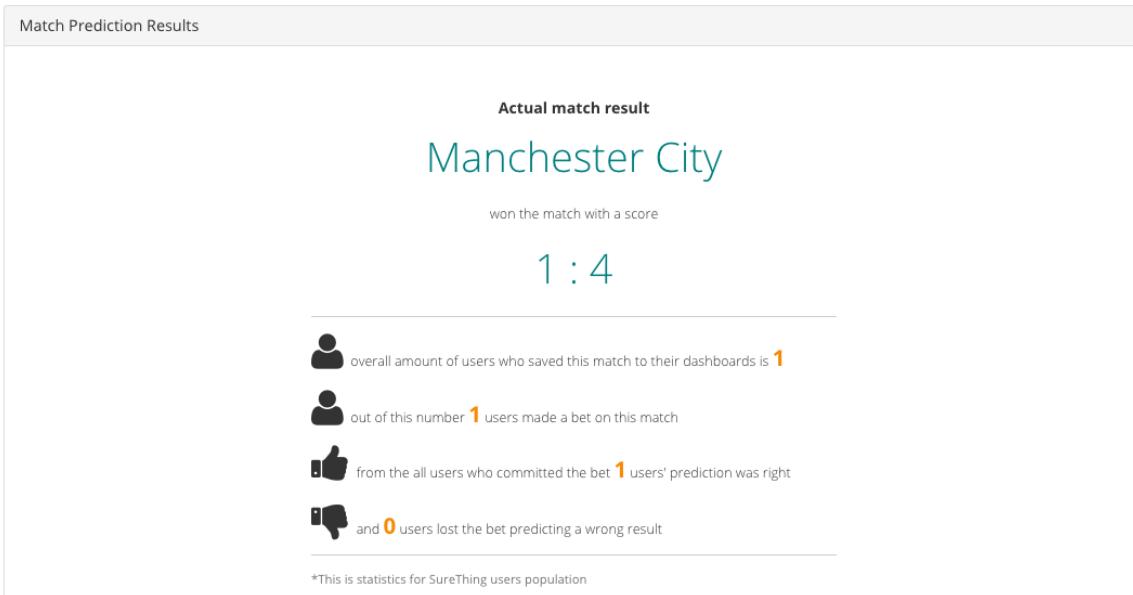


Figure 5.37: Played Match View, users' prediction statistics.

The view also offers a bar chart illustrating a breakdown of user preferences, namely how many users bet on the "home", "draw", or "away" result.

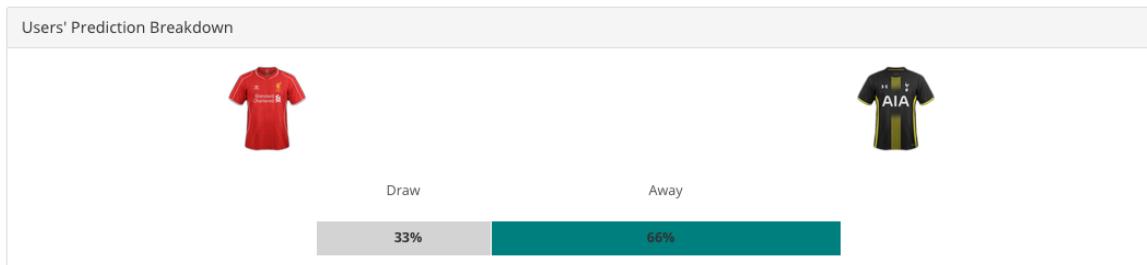


Figure 5.38: Prediction breakdown.

An authenticated user can also view basic feedback indicating whether this particular user won or lost the bet.



Figure 5.39: Played Match View, feedback provided for authenticated users.

User journey

The view is static and does not offer any additional interactions.

5.5.6 Dashboard

The SureThing dashboard is a centralised user space that provides convenient shortcuts to all the important prediction-related pages and tools. The dashboard can be used to store and view matches, change default prediction weights and make bets. It can be navigated to by clicking on the item "Dashboard" in the navigation menu of the website, located at the top of the page. On navigating to the dashboard, the user can see a list of saved matches ordered by date and a *dashboard menu* icon in the top left corner of the view. This is a screenshot of a typical dashboard view (only one match is saved):

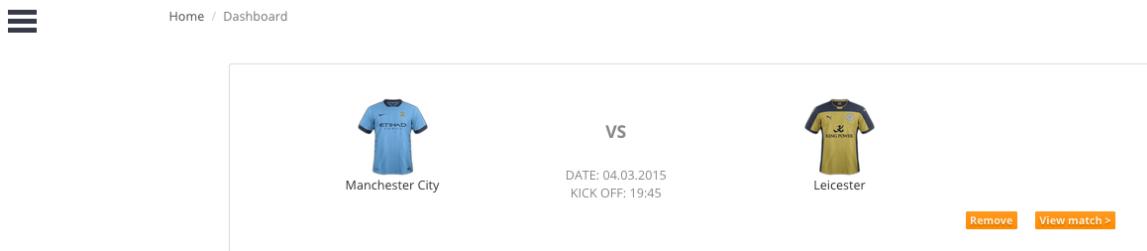


Figure 5.40: Dashboard view.

Matches that already have been committed by the user have a grey background and the predicted winner is highlighted. If the match has already been played, the result of the bet, either "Win" or "Loss", also appears on the match panel.

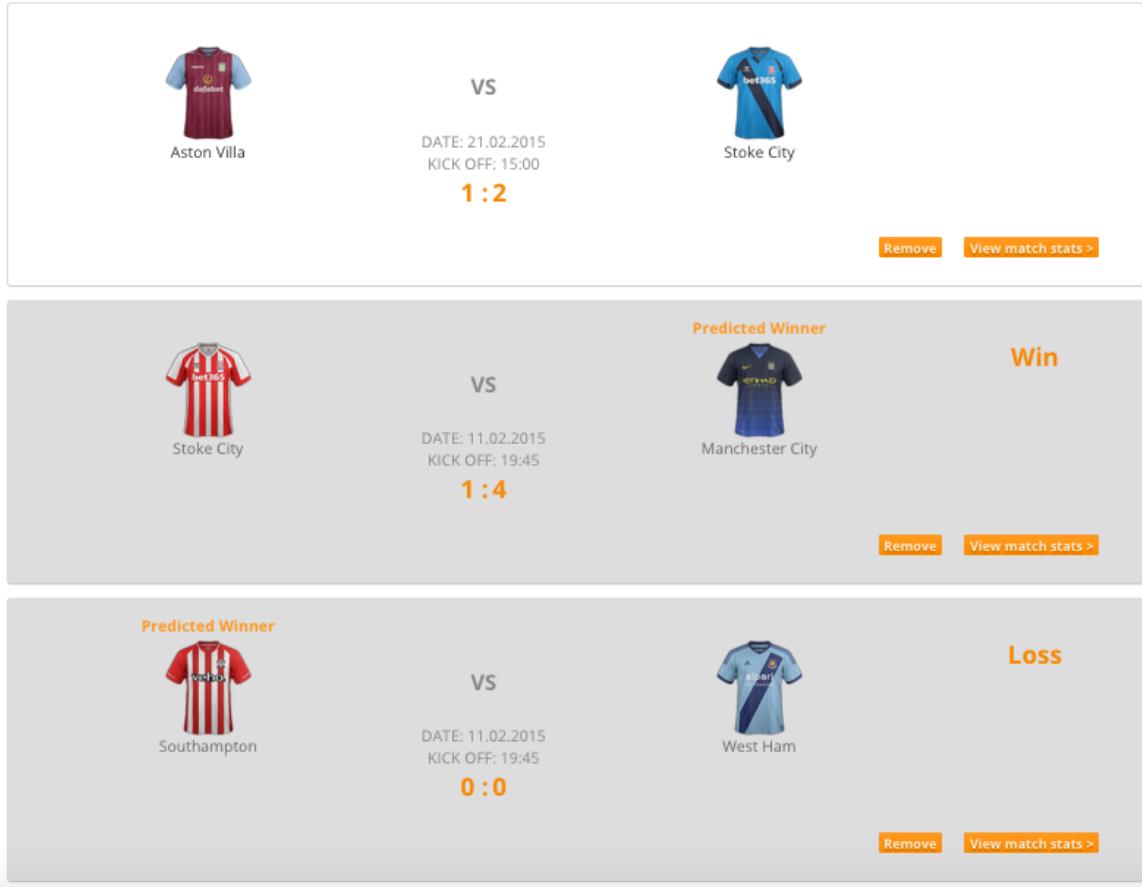


Figure 5.41: An example of a Dashboard View with matches that have been committed and played.

If the user does not have any matches saved, the dashboard looks as follows:



Figure 5.42: Dashboard with no saved matches.

User journey

In the top left corner, the user can see a small menu icon representing the dashboard menu. Clicking the button opens up a menu containing three items:

- Upcoming Matches. Displays all saved matches that have not been played yet.
- Archived Matches. Displays all saved matches that have already been played.
- Prediction Settings. A form that allows the user to set default prediction weights.

A panel containing the dashboard menu slides in and covers part of the page. It can be easily dismissed by choosing an item from the menu or clicking the "close" icon.

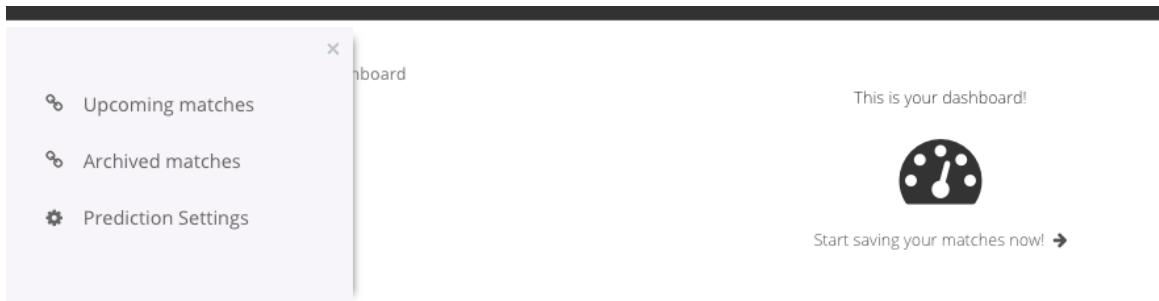


Figure 5.43: Dashboard menu.

If there are no saved matches in the dashboard, the user can navigate to the main pages containing upcoming events by clicking the link "Start saving your matches now ↗".

5.5.7 Notifications

This feature represents the one-way communication flow from SureThing to the application user. Every time a match previously committed by the user is finished (from the technical point of view, the match has changed its status to "played"), the application sends users messages notifying them whether their prediction guess was successful. Such messages contain detailed information about the user prediction, as well as a link to the relevant *Played Match View*.

Home / Messages / Message

You lost a bet for Everton vs. Leicester, played on 22.02.2015

Dear, gannet!

unfortunately, you did not predict this match result correctly!

Your prediction was that Everton was going to win.

And the result of the match was Draw, with the score: **2:2**

Just for the reference.

Your betting configuration was as follows:

LEAGUE POSITION MODULE Winner -- 50.0%

FORM MODULE Winner -- 20.0%

HOME-AWAY MODULE Winner: - 20.0%.

[View match stats >](#)

[Delete](#)

Figure 5.44: An example of a message sent to the user.

User journey

An authenticated user can navigate to the notifications inbox by clicking on the envelope-shaped icon located on the right-hand side of the navigation menu. The orange colour of the icon in the screenshot below indicates that the inbox contains unread messages, otherwise the icon color is grey.

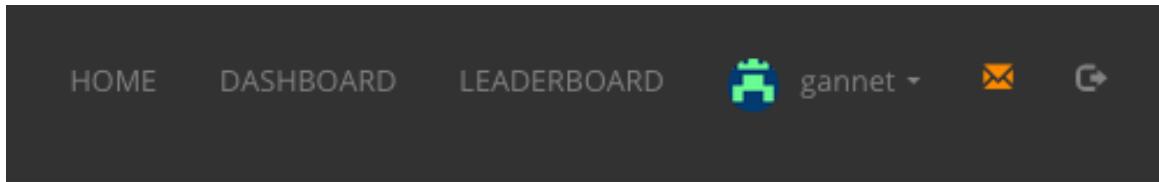


Figure 5.45: Navigation menu panel with an inbox icon.

On clicking the icon, the user is taken to the notifications inbox. Unread messages are displayed in a bold font.

A screenshot of the user inbox interface. At the top left, it shows 'Home / Messages'. Below that is a list of four notifications, each in a separate box. The first notification says 'You won a bet for Southampton vs. Liverpool, played on 22.02.2015' and has a timestamp 'February 24th 2015 10:58' with an orange 'View' button. The second notification says 'You lost a bet for Everton vs. Leicester, played on 22.02.2015' with a timestamp 'February 22nd 2015 16:03' and an orange 'View' button. The third notification says 'You lost a bet for Arsenal vs. Aston Villa, played on 01.02.2015' with a timestamp 'February 1st 2015 16:02' and an orange 'View' button. The fourth notification says 'You lost a bet for Sunderland vs. Burnley, played on 31.01.2015' with a timestamp 'January 31st 2015 21:33' and an orange 'View' button. At the bottom left, there is an orange 'Delete All' button.

Figure 5.46: User inbox.

5.5.8 Leaderboard

The leaderboard is a view containing a table capturing betting performance across the population of the website. This page can be navigated to by clicking the "Leaderboard" entry in the navigation menu of the application. Each line of the Leaderboard table contains the most basic information about application users: usernames, location and their favourite team as well as the betting statistics: games committed, won and lost. The table is ordered by the amount of win points for each user. Thus, the most successful predictors are located at the top of the table.



Home / Leaderboard

Position	User	Location	P	W	L	Fav team
1	marina	Old Aberdeen	87	66	21	Chelsea
2	gannet	Aberdeen	59	47	12	Man City
3	alisa	Aberdeen	50	42	8	None
4	john	Aberdeen	23	12	11	None

Figure 5.47: Leaderboard.

User Journey

Each username in the table is also a link. Clicking on the username of a user will take us to this user's profile page, letting an authenticated user view the profiles of fellow users.

5.6 Application Performance

Performance is a very important aspect of a web application. DEF

response time

sockets, threads multithreading how I fixed performance on Match.update_all_matches

5.7 Deploying the Application

Cloud Deployment is the most recent trend in application hosting. The formal name of this technology is Platform as a Service (PaaS). In the PaaS model, a service provider offers a fully managed platform in which applications can run.

Chapter 6

Testing

Testing was an important part of the project. This chapter describes the testing carried out and the points raised by it.

6.1 Unit Testing

Unit testing is the practice of testing different elements of the software using a test program to provide inputs to each element and evaluate the outputs. Known as white-box testing (meaning that the internal structure of the software is known to the tester), the tests are usually written before or at the same time as the tested components. According to Miguel Grindberg [17], "There are two very good reasons for writing unit tests. When implementing new functionality, unit tests are used to confirm that the new code is working in the expected way. ... A second, more important reason is that each time the application is modified, all the unit tests built around it can be executed to ensure that there are no regressions in the existing code; in other words, that the new changes did not affect the way the older code works." In a way, tests provide a safety net, meaning you can refactor the code at any time without the risk of breaking the original functionality.

Due to the time constraints, the focus was on testing only most critical parts of the project functionality. For this project it was especially important to ensure good code coverage for the business logic behind the model layer (prediction feature) and the external service layer of the application (api wrapper). This is explained in the chapter "Implementation" 5, section "Application Architecture". SureThing has a suite of 30 unit tests that can be run anytime to validate this part of the application functionality. Tests in this project are performed using the Python *unittest* library.

```

localhost: ~/PycharmProjects/surething
→ python manage.py test
....
----WRITING MATCHES----
My url http://football-api.com/api/?Action=fixtures&comp_id=1204&APIKey=2890be06-81bd-b6d7-1dcb4b5983a0&from_date=01.08.2014&to_date=31.05.2015

----WRITING STANDINGS----
My url http://football-api.com/api/?Action=standings&comp_id=1204&APIKey=2890be06-81bd-b6d7-1dcb4b5983a0
league table data updated!
.....
.
.
.
Ran 31 tests in 91.390s
OK

```

Figure 6.1: A screenshot of the terminal output showing the successfully passed test suite of the SureThing.

6.2 Continuous Integration with Travis CI

As the application grows, it may begin to take too long to run the unit tests. Therefore, it is worth automating this process by setting up a "Continuous Integration" or CI server. Travis CI was chosen for this task because it is easy to set up and available for free as a part of the GitHub Student Developer Pack. The service takes care of the unit testing allowing the developer to focus purely on the development process. Travis builds are triggered automatically when developer checks in the project code into the GitHub repository. Intergrating Travis CI was just a matter of creation a configuration file *travis.yml*.

```

1 language: python
2 python:
3   - "2.7"
4
5 # command to install dependencies
6 install: "pip install -r requirements.txt"
7
8
9 # command to run tests
10 script: python manage.py test

```

Figure 6.2: Travis CI configuration file.

A Travis status icon indicating whether the tests passed or failed was embedded into the README file. This is a convenient feature that helped to keep an eye on the build status from the GitHub repository.

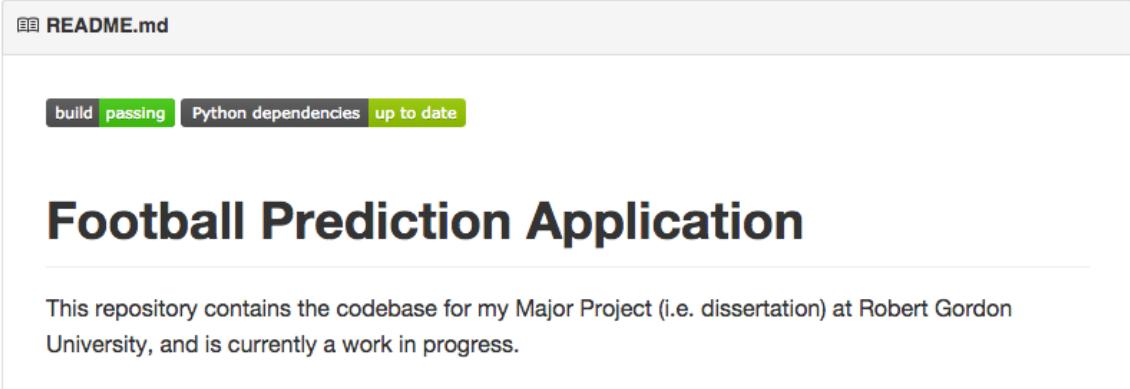


Figure 6.3: An extract from README on GitHub. Travis status icon indicates that the last build passed.

6.3 Use case Testing

”System Testing will test the system as a whole. This will be run by the developer taking into account the users requirements. Test cases will be created from the requirements with the inputs and expected outputs noted before the testing starts. Some of the test cases may satisfy more than one of the requirements. The tests will be carried out via the black box testing technique.” REWRITE IT

6.3.1 Test 1 – Use Case 1 Testing

6.4 User Acceptance Testing

During the final stage of the project, the completed application was given to a selection of target users to perform User Acceptance Testing. This type of testing helps to verify that the final product is easy to understand and use and that it in general works well for the users. User Acceptance Testing or UAT is a black-box technique, meaning that the testers are not aware of the internal workings of the code. It is important that the end users are asked to perform tasks close to how the application would be used in real life.

A suite of test cases highlighting the key functionality of the designed software must be created to perform UAT. In order to test SureThing, the following set of cases were designed and presented to the end users (UAT testers), alongside a questionnaire that can be found in the Appendix B.1.

1. Create a new account
2. Login into the newly created account

3. Look through the list of matches and choose the ones you are the most interested in, then save them to dashboard
4. Set your own prediction weights
5. Commit to betting on a match

The raw data collected from the user acceptance testing can be found in the Appendix [B.2](#).

A number of potential end users were asked to test the application using the above set of steps. It is possible to say that the testing was successful and the application was well accepted by all the respondents. However, the aim of the UAT is to identify and resolve drawbacks in the tested software. The performed testing helped to find some issues with regards to the key functionality.

According to the feedback, the most difficult steps to carry out were "saving matches to the dashboard," "committing a match" and "setting own prediction weights". First of all, it looks like the application needs a better way to communicate, why a user needs to save a match to the dashboard before committing to bet. Secondly, SureThing offers two ways to preview an unplayed match and the user needs to "open" a saved match *from the dashboard* in order to see all the prediction features, be able to adjust the prediction outcome and commit the match. This is a known issue and a suggested resolution would be to allow the user to see a match in the full "prediction mode" as soon as the match has been saved by the user.

Most of the users found that it easy to set up a new account (the average score was 8.7 out 10), they found it moderately easy (the average score was 6.8 out 10) to find and set the prediction settings and a large majority agreed that the website offers enough statistics to support their betting decision (6 out of 7 testers). An interesting point is that some users mentioned that it was not quite clear why they need to set a default set of prediction settings. The application clearly needs a tutorial or a FAQ page, that would explain the purpose of the different levels of settings.

When asked about the "worst aspects of the application", testers again mentioned the lack of an explanation of how prediction settings work and lack of different football leagues. The fact that only one league is supported at the moment is due to the fact that to have more would cost more money. If the application was to be expanded, this would be the one of the first extensions to be made. As to the "best aspects", many of the users praised the application for its neat design and simple layout. Most of them also mentioned that the way the stats are presented is very effective and relevant to the prediction feature.

In general, the User Acceptance Testing was very beneficial and provided a valuable feedback from the potential users. The first outcome is that the application needs a help page or a tutorial. The concept

of the dashboard also needs to be revisited and possibly redesigned, as not all of the testers were clear of its purpose. Some of the answers were very inspirational in terms of the future development. For example, one of the respondents would like to see an option to compete against a team of friends and be able to see a separate leaderboard.

Due to the time constraints, the resolution of the findings is beyond the scope of this project but the testing certainly proved it's worth and for future development of the project this information is extremely useful.

Chapter 7

Conclusion

This final section shows what the main outcomes of the project were and the conclusions that can be drawn from them.

7.1 Evaluation

The main aim of the project was to create an application for predicting the outcome of football matches and to assist its users on creating their own individual betting system and ultimately make a more informed bet. This was certainly achieved. The 3 main problems facing a user that wants to bet on football were addressed also. Firstly, the application does indeed collect and display all the relevant statistics that a football punter would want to know before making a bet and it does this in a way that is both clear and concise. Secondly, it allows the user to create their own prediction formula, which makes it unique amongst similar football websites which usually are either statistics only or "black box" (unmodifiable) prediction systems. Thirdly, it monitors the performance of the user which is a vital tool for improving betting decisions (or for showing a user it might be time to give up gambling altogether).

Going into more detail with regard to the functional requirements of the application that were outlined initially, it can be seen that again most of these were achieved successfully. From creating, signing in to and managing user accounts, to how thorough the available data is as well as how clearly it is presented, all the way through to what was probably the most important aspect, the prediction feature itself, were all very successfully implemented. Of all the initial functional requirements stated at the beginning of the project, the user leaderboard was the only one not fully realised due to time constraints.

The non-functional requirements laid out at the beginning were all met and exceeded expectations.

Usability - The application is extremely user friendly, it is uncluttered and navigation is highly intuitive. The initial, admittedly small number, of test users praised the design and overall ease of use of the application.

Responsiveness - It is a fully responsive website, tested successfully on a variety of screen resolutions down to a minimum of 640x960 (DVGA).

Performance - With the correct choice of architecture used and by keeping the code streamlined, the performance of the application was optimised as much as possible. Only by spending money on a faster API could it be improved, which is something to be considered if ever taking the application to market.

Cross-browser support - The website was tested and works on all the usual web browsers including Chrome, Firefox, IE9 and Safari.

Maintainability - The code written for the website is clear and understandable. There is no doubt that other developers would find it easy to see what has been done and why it was done a certain way.

Extensibility - It became clear early on that the application has the potential to be of a much larger scale. With this in mind, the project was developed to make it a simple matter of adding new and advanced features to it without having to rewrite large parts of the code.

User acceptance testing proved invaluable. Not only for getting fresh eyes on the project that might catch something missed due to over-familiarity of the project, but also for building confidence that this was a worthwhile endeavour.

Overall, the project has been highly successful, with a very well designed web application as a result.

7.2 Personal Statement

I started researching this project in September 2014 and soon after began designing and implementing the application. It would have been nice to be able to focus on this alone but instead I had to fit working on this around studies, exams and courseworks while trying my best not to ignore my husband and 1 year old daughter. Thankfully, by starting early I gave myself the best chance of producing a worthwhile project whilst still being able to complete my studies.

With regard to the development of my skills, I am most happy with how well I learned to use the Python programming language. I started with zero knowledge or experience of using it and now I feel I would be able to use it at a professional level. The most surprising thing about the project for me was how useful Test Driven Development was. With the sheer amount of complicated business logic

running in the background, good test coverage was essential for a smooth developmental process and although I knew it would be useful, just how useful it turned out to be was an important realisation for me.

With more time I believe this application could be taken much further, it has surprised me with the scope and scale that could be achieved with the proper development. As it stands, it is a well built, useful application and I am immensely proud of my work.

7.3 Improvements and Future Work

At the moment, the developed application is a prototype suggesting what the system is capable of, rather than a fully-functional and thoroughly tested application and there are many ways in which SureThing could be developed in the future.

The most logical improvement would be to support more football leagues. There is nothing in the code of the application that limits it to predicting the results for only the English Premier League and not for any other league in the world. The EPL was chosen because the API for it was free and to have more leagues supported would mean paying money to an API provider. As it is possible to bet on almost every match from almost every league in the world, it makes sense that SureThing would have the option of predicting the outcome of any and all of these matches. In fact, for it to be commercially viable then it must have this option as gamblers would want the widest possible choice of matches to bet on.

There are smaller improvements brought to light by the User Acceptance Testing that could also be made, a tutorial or help menu would be useful for those not overly familiar with football statistics or sports betting in general, and maybe a rethink of the dashboard feature.

There would also need to be a decision made about the future purpose of the application, namely whether to link it more closely to the gambling industry or to keep it separate and even turn it into an alternative to gambling. It would be possible to affiliate the application to a specific online bookmakers, and to offer links and odds from that particular website in return for a fee. This is done by many different statistics and prediction websites. However, if the focus was to be on our own users' needs then it would be a very useful feature to be able to compare all the odds from all the online bookies and recommend the best odds available. As the main aim of the project was to help users with their gambling decisions, this would be the logical way forward.

It is also possible to develop the project as a self contained game and act as a substitute to the gambling experience. By improving and expanding the community aspects of the application, this

becomes a distinct possibility. By adding more social network features such as the option to follow other users or commenting on games, as well as improving the users performance league so friends can play against each other

For example, it can remain a self-contained game and become a replacement to the real-world gambling experience. Another option is to link the application with the actual bookmakers and to turn it into a professional gambler's tool used that would provide all the relevant statistics and odds comparison.

The application can be further developed in many ways. When putting together the project requirements, a number of optional requirements were outlined. The optional requirements are basically the ways how the application can be further developed. More tests (check with potential users) would be required

Although, one of the key features of the application is to try not to overwhelm the user with statistics, as opposed to many football stats websites, this view would need a little bit more additional information to complete the big picture.

Appendix A

Target Audience Questionnaire

The questionnaire presented below was answered by 9 respondents with strong interest in football betting.

A.1 Questions

This questionnaire intends to collect information about the way football punters make their betting decisions (decide which team to bet on).

The questionnaire should only take around five minutes to fill in and your answers will be used to aid the development of a web application simulating the football betting experience. The future application will provide its users with all the necessary football statistics (without going too much into detail) and allow them to participate in the prediction process by making their own prediction formula.

Question 1

How many times a week do you bet on football or other sporting events?

- Less than once a week
- 1 time a week
- 2-3 times a week
- More than 3 times a week

Question 2

How many sources of information (websites, newspapers, your favourite mobile app) do you look into before making your placing your bet?

For example, if you use 2 different websites (such as BBC News and Whoscored.com), the answer is 2.

- None
- 2-3 sources
- More than 3 sources

Question 3

Please specify the sources of information that you use to support your decision.

- TV programmes
- Sports info websites (e.g., BBC News, The Guardian)
- Football info websites (e.g., whoscored.com, squawka.com)
- Gambling websites (e.g., Bet365, Ladbrokes)
- Betting communities or forums (e.g., OLGB.com)
- Social Networks (e.g., betting tips from football experts on Twitter)
- Printed media (newspapers)
- Friend's advice
- Mobile apps
- Specify your own:

Question 4

What main factors do you take into account before placing a bet?

Choose more than one or add your own

- Form
- League position
- Result of the previous match for each team

- Home/Away performance so far in the season
- Change of a team manager
- Injuries/susensions of players
- Weather
- Own hunch
- Specify your own:

Question 5

Do you record your betting performance?

- Yes
- No

Question 6

What statement describes you best?

- I only use one bookmaker to place my bets
- I compare the odds several bookmakers and choose the best bookmaker for each match

Question 7

Would you find a web application allowing you to participate in the prediction of a match result by making up you own prediction formula useful?

- Yes
- No
- Not sure

A.2 Answers

Respondent	Question 1	Question 2		Question 3
		2-3 sources		
1	1 time a week	2-3 sources		Sports info websites (e.g., BBC News, The Guardian), Football info websites (e.g., whoscored.com, squawka.com)
2	Less than once a week	2-3 sources		TV programmes, Printed media (newspapers)
3	2-3 times a week	2-3 sources		TV programmes, Printed media (newspapers), Friends advice
4	More than 3 times a week	More than 3 sources		TV programmes, Sports info websites (e.g., BBC News, The Guardian), Gambling websites (e.g., Bet365, Ladbrokes), Betting communities or forums (e.g., OLCGB.com), Social Networks (e.g., betting tips from football experts on Twitter), Mobile apps
5	More than 3 times a week	2-3 sources		TV programmes, Gambling websites (e.g., Bet365, Ladbrokes), Printed media (newspapers)
6	Less than once a week	2-3 sources		Football info websites (e.g., whoscored.com, squawka.com)
7	More than 3 times a week	2-3 sources		Football info websites (e.g., whoscored.com, squawka.com), Gambling websites (e.g., Bet365, Ladbrokes), Mobile apps
8	Less than once a week	2-3 sources		TV programmes, Sports info websites (e.g., BBC News, The Guardian), Printed media (newspapers)
9	2-3 times a week	2-3 sources		Sports info websites (e.g., BBC News, The Guardian)

Table A.1: Table illustrating answers to questions 1-3 in the Target Audience Questionnaire

Respondent	Question 4	Question 5	Question 6	Question 7
1	Form, League position, Result of the previous match for each team, Own hunch	No	I only one bookmaker to place my bets	Yes
2	Form, League position, Injuries/susensions of players	No	I only use one bookmaker to place my bets	Not sure
3	Form, League position, Home/Away performance so far in the season, Own hunch	No	I compare the odds several bookmakers and choose the best bookmaker for each match	Yes
4	Form, League position, Result of the previous match for each team, Home/Away performance so far in the season, Own hunch	No	I only use one bookmaker to place my bets	Yes
5	Form, League position, Home/Away performance so far in the season, Change of a team manager	No	I compare the odds several bookmakers and choose the best bookmaker for each match	Yes
6	Form, Own hunch	No	I only use one bookmaker to place my bets	No
7	Form, Home/Away performance so far in the season, Injuries/susensions of players	No	I only use one bookmaker to place my bets	Yes
8	Form, League position, Result of the previous match for each team, Injuries/susensions of players, Own hunch	No	I only use one bookmaker to place my bets	Yes
9	Form, Own hunch, Whether or not the odds appear to offer good value	No	I only use one bookmaker to place my bets	Yes

Table A.2: Table illustrating answers to questions 4-7 in the Target Audience Questionnaire

Appendix B

User Acceptance Questionnaire

The questionnaire presented below was answered by 7 potential end users.

B.1 Questions

This questionnaire intends to verify that the developed website is easy to understand and use for the potential users.

Please, complete the following steps using the application:

1. Create a new account
2. Login
3. Look through the list of matches and choose the ones you are the most interested in, then save them to dashboard
4. Set your own prediction weights
5. Commit to betting on a match

On completing the steps, please answer the following questions:

Question 1

Which step did you find the most difficult to carry out and why?

Please, provide your own answer.

Question 2

How easy was it to set up a new account?

Please, rate the following question on a scale of 1 (Very easy) to 10: (Very confusing).

Question 3

Was it easy to find and set the prediction settings?

Please, rate the following question on a scale of 1 (Very easy) to 10: (Very confusing).

Question 4

How would you rate the overall user friendliness of the application?

Please, rate the following question on a scale of 1 (Very easy) to 10: (Very confusing).

Question 5

Did you find there was enough statistics to aid your decision making, and if not, what would you want to see?

Please, provide your own answer.

Question 6

What did you find was the best aspect of the application? E.g., design, navigation

Please, provide your own answer.

Question 7

What did you find was the worst aspect of the application?

Please, provide your own answer.

Question 8

What would you like to see in the application?

Please, provide your own answer.

B.2 Answers

Respondent	Question 1	Question 2	Question 3	Question 4
1	Probably the changing the prediction settings but even that wasn't hard really	More leagues	10	8
2	Saving matches to the dashboard, wasn't immediately apparent why I had to do that	A way to communicate with friends using the same app, maybe a separate league table for friends than just for all users	9	7
3	Committing the match, it wasn't immediately apparent that I can only commit a match previously saved in the dashboard	Maybe some news feeds	7	8
4	Changing the prediction settings, it could have been clearer what I was supposed to do	A help menu or FAQ would be useful	7	6
5	It was all fairly easy	More leagues, comparison betting odds from different bookies	10	10
6	Saving matches to the dashboard, wasn't sure why I had to do that	More social interaction, link to Facebook	8	7
7	Setting the prediction weights, wasn't sure of what to set them to so I juts left them at their default settings	A guide to how to use the app would have been nice	10	7

Table B.1: Table illustrating answers to questions 1-4 in the User Acceptance Questionnaire

Respondent	Question 5	Question 6	Question 7	Question 8
1	8	The layout was easy on the eye and simple enough to find your way about	Not enough football leagues represented	Yes
2	5	The design was simple but nice to look at	The dashboard seems a bit pointless	Yes
3	8	The prediction feature was great, especially because I could change it myself	The design was a little boring	Yes
4	5	The way the stats and matches were presented, very clear	The way to change the prediction settings could have been explained better	Yes
5	10	It was very self explanatory how it all worked together	Lack of football leagues to use	No, betting odds
6	7	I liked the design	The dashboard wasn't very useful	Yes
7	5	Navigating from page to page was easy	I was confused about the prediction settings	Yes

Table B.2: Table illustrating answers to questions 5-7 in the User Acceptance Questionnaire

Appendix C

Installation Instructions

The code can be checked out using git by executing the following command in the terminal:

See the following command :

```
1 $ git clone git@github.com:marinamarina/sure-thing.git
```

Installation instructions are found at the following url:

<https://www.github.com:marinamarina/sure-thing/blob/master/README.md>.

If any issues arise regarding installation of any part of the system, do not hesitate to contact me at
1014481@rgu.ac.uk

Appendix D

Credits

Some the application's code was taken from the GitHub repository accompanying the book by Miguel Grindberg "Flask Web Development" [17]. Below are listed the components of the system that rely on this code to some extent:

- user authentication, account verification
- basic test suite
- user authentication test suite

Some of the handy decorator functions used in the Python code were taken from Flask documentation [45], section "View Decorators".

Appendix E

Project Management

The project was not developed following a single software development methodology. Different techniques from various methodologies were used for the project. Some of the planning was made beforehand in the traditional, Waterfall approach fashion, such as researching the project background, defining application requirements, sketching the application wireframes, etc. On the other hand, some of the design techniques were adopted from the Agile methodology, namely user journeys.

The project implementation phase was carried out in an iterative manner. For defining the project requirements, the application functionality was divided into clearly defined units (so called "high-level features" introduced in the chapter "Requirements Analysis" 3), each associated with an implementation iteration. Excel sheets were used for defining the set of tasks for each iteration. GitHub issue tracker was also used as a supporting productivity tool for this project. Code related tasks were recorded as "issues" for the project GitHub repository. In general, the GitHub issue tracker was a very efficient tool for keeping the project organised.

Appendix F

Presentation Slides

The following is a weekly summary of the work carried during the development of this body of work. It covers tasks that were completed, tutorials that were worked through, articles that were read and reviews of discussions / meetings held with the project supervisor and other third parties.

Bibliography

- ¹ Michael Bayer and various contributors. Sqlalchemy orm. [Online]. Available from: http://docs.sqlalchemy.org/en/rel_0_9/orm/, 2014.
- ² BBC Sport. BBC Sport. <http://www.bbc.co.uk/sport>, 2015. Available online. [Accessed 10-11-2014].
- ³ Bet365. Bet365. <http://www.bet365.com>, 2015. Available online. [Accessed 12-11-2014].
- ⁴ Open Web Community Development Bocoup. The javascript task runner. [Online]. Available from: <http://gruntjs.com/>, 2015.
- ⁵ John Boxall. A python guide to handling http request failures. [Online]. Available from: <http://www.mobify.com/blog/http-requests-are-hard/>, 2014.
- ⁶ Odds Checker. Your bets bet. [Online]. Available from: <http://www.oddschecker.com/>, 2015.
- ⁷ Various contributors. Jinja2, full featured template engine for python. [Online]. Available from: <http://jinja.pocoo.org/>, 2015. [Accessed 14-Oct-2014].
- ⁸ Douglas Crockford. Json (javascript object notation). [Online]. Available from: <http://martinfowler.com/articles/continuousIntegration.html#PracticesOfContinuousIntegration>, 2015.
- ⁹ Nando Florestan. Large web apps in python: A good architecture. [Online]. Available from: http://nando.oui.com.br/2014/04/01/large_apps_with_sqlalchemy_architecture.html, 2015. [Accessed 5-Mar-2015].
- ¹⁰ Football 365. Football 365. <http://www.football365.com>, 2015. Available online. [Accessed 10-11-2014].
- ¹¹ Football-API. Football-API. <http://www.football-api.com>, 2015. Available online. [Accessed 1-10-2014].
- ¹² Footbee. Footbee. <http://footbe.net/en/>, 2015. Available online. [Accessed 14-10-2014].

- ¹³ Forebet.com - Mathematical Prediction. Forebet.com - Mathematical Prediction. <http://www.forebet.com/en/>, 2015. Available online. [Accessed 14-10-2014].
- ¹⁴ Ben Frain. *Responsive web design with HTML5 and CSS3*. Packt Publishing Ltd, 2012.
- ¹⁵ Gambling sites. Gambling sites. <http://www.gamblingsites.com/history/microgaming/>, 2015. [Accessed 13-Apr-2015].
- ¹⁶ Goal.com. Goal.com. <http://www.goal.com/>, 2015. Available online. [Accessed 10-11-2014].
- ¹⁷ Miguel Grindberg. *Flask Web Development. Developing Web Applications with Python*. O'Reilly Media, 2014.
- ¹⁸ Guardian Sport. Guardian Sport. <http://www.theguardian.com/uk/sport>, 2015. Available online. [Accessed 10-11-2014].
- ¹⁹ Chris Eppstein Hampton Catlin, Natalie Weizenbaum and various contributors. Sass preprocessor. [Online]. Available from: <http://sass-lang.com/>, 2015.
- ²⁰ ho Scored? Who Scored? <http://www.whoscored.com>, 2015. Available online. [Accessed 12-11-2014].
- ²¹ Automattic Inc. A globally recognized avatar. [Online]. Available from: <https://en.gravatar.com/>, 2015.
- ²² Statista Inc. The statistics portal. [Online]. Available from: <http://www.statista.com>, 2015.
- ²³ Injuries And Suspensions. Injuries And Suspensions. <http://injuriesandsuspensions.com/>, 2015. Available online. [Accessed 12-11-2014].
- ²⁴ ECMA International. EcmaScript language specification, 2011.
- ²⁵ Frank Keogh and Gary Rose. Football betting - the global gambling industry worth billions, 2013. [Accessed 13-Apr-2015].
- ²⁶ Ladbrokes. Ladbrokes. <http://betting.ladbrokes.com/en/football-betting>, 2015. Available online. [Accessed 12-11-2014].
- ²⁷ Jim Macos. Parameters and system variables in a football betting system. <http://www.betstories.com/2013/12/06/football-betting-system-variables-parameters/>, 2013. Available online. [Accessed 13-10-2014].
- ²⁸ Jim Macos. Simple betting System. <http://www.betstories.com/2014/09/24/simple-betting-system/>, 2014. Available online. [Accessed 13-10-2014].
- ²⁹ Joseph Anthony Maguire. Gambling and sports, 2015. [Accessed 13-Apr-2015].

- ³⁰ Matt Makai. Api integration. [Online]. Available from: <http://www.fullstackpython.com/api-integration.html>, 2015.
- ³¹ Robert C Martin. *The Robert C. Martin Clean Code Collection (Collection)*. Prentice Hall, 2011. [Accessed 17-Mar-2015].
- ³² Math Betting. Math Betting. <https://apps.betfair.com/trading/mathbetting/>, 2015. Available online. [Accessed 14-10-2014].
- ³³ Microgaming. Microgaming, our story. <http://www.microgaming.co.uk/our-story>, 2015. [Accessed 13-Apr-2015].
- ³⁴ OLGB Betting Community. OLGB Betting Community. <http://www.olgb.com>, 2015. Available online. [Accessed 14-10-2014].
- ³⁵ Open Footy. Open Footy. <http://www.footytube.com/openfooty/service.php>, 2015. Available online. [Accessed 1-10-2014].
- ³⁶ Opta Sports. Opta Sports. <http://www.optasports.com>, 2015. Available online. [Accessed 1-10-2014].
- ³⁷ Paddy Power. Paddy Power. <http://www.paddypower.com/bet>, 2015. Available online. [Accessed 12-11-2014].
- ³⁸ Harry JW Percival. *Test-Driven Development with Python.* " O'Reilly Media, Inc.", 2014. [Accessed 16-Mar-2015].
- ³⁹ Dusty Phillips. *Python 3. Object-Oriented Programming*. Packt Publishing, 2010.
- ⁴⁰ Sergey Pozhilov. Progressus template. [Online]. Available from: <http://www.gettemplate.com/demo/progressus/>, 2015.
- ⁴¹ Punters Lounge. Punters Lounge. <http://www.punterslounge.com/>, 2015. Available online. [Accessed 1-10-2014].
- ⁴² Ronald A Radice and Richard W Phillips. *Software engineering: an industrial approach*. Prentice-Hall, Inc., 1998. [Accessed 8-Mar-2015].
- ⁴³ Kenneth Reitz. Requests. [Online]. Available from: <http://docs.python-requests.org/en/latest/api/>, 2015.
- ⁴⁴ John Resig. jquery. [Online]. Available from: <http://jquery.com/>, 2015. [Accessed 4-Jan-2015].
- ⁴⁵ Armin Ronacher and various contributors. Flask micro framework. [Online]. Available from: <http://flask.pocoo.org/>, 2015.

- ⁴⁶ Armin Ronacher and various contributors. Flask micro framework - application factories. [Online]. Available from: <http://flask.pocoo.org/docs/0.10/patterns/appfactories/>, 2015.
- ⁴⁷ Maxim Shirshin. Scaling down the bem methodology for small projects. <http://www.smashingmagazine.com/2014/07/17/bem-methodology-for-small-projects/>, 2014.
- ⁴⁸ Sky Sports. Sky Sports. <http://www.skysports.com/>, 2015. Available online. [Accessed 10-11-2014].
- ⁴⁹ Squawka. Squawka. <http://www.squawka.com>, 2015. Available online. [Accessed 12-11-2014].
- ⁵⁰ Max Surguy. Bootsnipp. design elements, playground and code snippets for bootstrap html/css/js framework. [Online]. Available from: <http://bootsnipp.com/>, 2015.
- ⁵¹ The History of Online Gambling. The History of Online Gambling. <http://www.onlinegambling.com/online-gambling-history.htm>, 2015. [Accessed 12-Apr-2015].
- ⁵² The Times. The Times. <http://www.thetimes.co.uk>, 2015. Available online. [Accessed 10-11-2014].
- ⁵³ tobeadded. Requirejs, javascript file and module loader. [Online]. Available from: <http://requirejs.org/>, 2015. [Accessed 4-Jan-2015].
- ⁵⁴ Twitter. Twitter bootstrap css3 framework. [Online]. Available from: <http://getbootstrap.com>, 2015.
- ⁵⁵ Vital Football News and Fans community. Vital Football News and Fans community. <http://www.vitalfootball.co.uk>, 2015. Available online. [Accessed 14-10-2014].
- ⁵⁶ Vitibet. Vitibet. <http://www.vitibet.com/>, 2015. Available online. [Accessed 14-10-2014].
- ⁵⁷ W3C & WHATWG. Html (hypertext markup language). <http://www.w3.org/html/>, 2014.
- ⁵⁸ Wikipedia. Github — Wikipedia, the free encyclopedia. [Online]. Available from: <http://en.wikipedia.org/wiki/GitHub>, 2015. [Accessed 9-Mar-2015].
- ⁵⁹ Wikipedia. Bet365 — Wikipedia, the free encyclopedia. [Online]. Available from: <http://en.wikipedia.org/wiki/Bet365>, 2015. [Accessed 22-Feb-2015].
- ⁶⁰ WilliamHill. WilliamHill. <http://sports.williamhill.com/bet/en-gb/betting/y/5/Football.html>, 2015.
- ⁶¹ Matt Wright. How i structure my flask applications. [Online]. Available from: <http://mattupstate.com/python/2013/06/26/how-i-structure-my-flask-applications.html#s21>, 2013. [Accessed 12-Mar-2015].

⁶² XML Soccer. XML Soccer. <http://www.xmlsoccer.com/>, 2015. Available online. [Accessed 1-10-2014].

⁶³ Yandex. Bem methodology. <http://https://bem.info/>, 2005.