COM814: Project 2015 - 2016

Dissertation

School of Computing & Information Engineering

Predicting The Outcome of Football Matches

Supervisor: Dr Patrick Corr
Second Marker: Dr Giuseppe Trombino

3 October 2016

Ulster University

Plagiarism Statement

I declare that this is all my own work and does not contain unreferenced material copied from any other source. I have read the University's policy on plagiarism and understand the definition of plagiarism. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions in accordance with the policies and procedures of the University. A mark of zero may be awarded and the reason for that mark will be recorded on my file.

Acknowledgements

I would like to thank my project supervisor Dr Patrick Corr for his guidance and feedback given throughout the project. He was always readily available to ensure the project progressed smoothly and deal with any queries that arose. Also my second marker Giuseppe Trombino provided valuable feedback throughout that improved the quality of the project.

Finally, thanks to my family who have provided me with continuous support and assistance and to anyone else that contributed directly or indirectly to the project, in any way.

Table of Contents

Abstract

The task of this project is to develop software that will predict the outcome of Football matches based on previous events. Whilst it is acknowledged that this task would not be completed to 100% accuracy, the aim was that the software predicted the outcome of games with a relatively high degree of accuracy. The system was implemented predicting games from the English Premier League as well as the German Bundesliga.

The solution to this task is based upon Machine Learning, in particular the K-Nearest Neighbour algorithm. Research was conducted on similar solutions and various Machine Learning algorithms that had not been availed of. This enabled the selection of the most appropriate algorithm.

This system was developed as an Android mobile app. A business analysis was undertaken to determine what platform to develop for. Development guidelines and best practices were applied and relevant stakeholders were engaged in order to influence design and system functionality.

Although not required by the project additional functionality was added by altering the implemented algorithm to show a measure of confidence in predictions generated.

Students and fellow professionals can use the project to better understand Machine Learning or build upon it to yield even more accurate predictions. Whilst this project applies to predicting the outcome of Football games, the possible uses of the K Nearest Neighbor algorithm is endless, for example predicting weather forecasts and clinical outcomes.

Keywords: **Football, Android, Machine Learning, K-Nearest Neighbour, Prediction**

Introduction

1.1 Introduction

The development of a mobile world has enabled product and service innovation that was previously unimaginable. All areas of society are increasingly opting for technological solutions to improve products and services, as well as making tasks and processes more efficient. Mobile devices are increasingly becoming embedded in our day to day activities, especially with the rise of The Internet Of Things. Online banking, shopping and even dating our now at our finger tips using mobile devices.

Another key industry to benefit from the digital world is Sports. Technological advances have brought about live sporting coverage on the go, wearable apps to support referees, greater sporting content and the ability to place bets on Sporting events online.

With the rise of online sports betting, bookmakers have been benefiting from record profits (Financial Times, 2016), with online sports betting accounts seen as a window to temptation. A major driver of the bookmakers profitability is their ability to capitalise on the uncertainty around the outcome of sporting events. With Football being the most popular sport on earth and increasingly difficult to predict, average individuals are losing money to bookmakers on a regular basis. For example, the latest English Premier League champions Leicester began the 2015/16 season with 5000/1 odds of becoming champions. This gives an idea as to how unpredictable the sport can be.

Part of this problem is that individuals attempting to predict the outcome of games are influenced by emotion or personal knowledge. With data analytics increasingly being adopted in sports, Machine Learning has increasingly been utilised to predict the outcome of games. This utilises pattern recognition and enables predictions to be made on previous events. It has proven to be of benefit in a variety of areas of society for example predicting financial market movements and medical outcomes.

This provided a great platform to develop a system that predicts the outcome of Football games with a high degree of accuracy. Many systems claim to do so but there is little mention of how they do or how accurate their predictions are.

However often these predictions are not based on data of previous events, failing to avail of powerful prediction mechanisms.

To ensure the solution system operated as desired and as accurately as possible relevant research was undertaken and target end users were consulted throughout the entire development lifecycle. This is discussed in further chapters.

1.2 Problem Statement

Predicting the score of Football matches is a difficult task. Many outlets claim to have the ability to do so, however it is unclear how reliable they are or how they make predictions. A system is required to predict the probability of all possible outcomes of Football matches with a high degree of accuracy based on the outcome of previous events.

1.3 Aim

To develop an Android mobile app that predicts the Win/Lose/Draw probabilities for Football matches to a high degree of accuracy. The system should use data related to previous games and provide predictions based on this, rather than factors that would traditionally be considered, such as form and team selection.

1.4 Objectives

In order to successfully achieve the aim outlined, a set of objectives were decided upon:

- Conduct research of the problem and demonstrate what benefits this system offers
- Conduct research of similar solutions available, critically analysing them
- Liaise with potential end users to gather functional and non functional requirements, as well as HCI preferences.
- Design a user interface that enables end users to quickly access content they are interested in viewing
- Implement an appropriate machine learning algorithm / statistical model to predict the outcome of Football games
- Test the accuracy of predictions against real life match outcomes

- Administer testing with potential end users who have shown interest in the system

1.5 Development Requirements

The application was developed using existing hardware and software that is readily available at little or no cost in order to complete the aim and objectives as described above. The following were required to develop the proposed solution:

- A laptop capable of running Android app development software
- Android Studio the official Android IDE to build apps for devices on Android Operating System. Also contains a range of emulators for testing apps
- Server space, 000WebHost which offered a control panel including MySQL databases and phpMyAdmin to facilitate a database containing previous results of Football matches and related data. The control panel also offered a file manager for storing PHP scripts.
- Text Edit for writing PHP scripts
- Programming knowledge: Java, Android Development, PHP, MySQL

1.6 Dissertation Outline

Chapter two provides a review of available literature highlighting the reasons why predicting the outcome of Football matches is a problematic task. It also provides a review of websites and mobile apps that are currently available addressing the problem, identifying the pros and cons of each. Additionally details of an analysis of the project risks and a business case that were undertaken before development are provided.

Chapter three covers how requirements were gathered from end users. It documents user stories and outlines both functional and non functional requirements to be incorporated in the proposed system.

Chapter four presents an introduction to the website design. It discusses the guidelines and best practices that were consulted, as well as a design template for the software. It also contains an app navigation diagram, as well as the systems architectural design, such as the client server interaction and database design.

Chapter five contains information regarding the development methodology used and rationale for the selection. It then follows on to review the implementation process including the platforms used and the steps taken to complete development of the system.

Chapter six discusses the overall testing strategy, as well as documentation of all test results gathered.

Chapter seven is the conclusion of the dissertation. The final product is critically analysed, addressing if the project met it's objectives and proposing a list of further improvements that could be made in future releases.

Analysis

2.1 Introduction

This stage is the beginning of the Software Development Lifecycle and helps identify the overall direction the project will take. This chapter provides the reader with available literature relating to the problem, an analysis of available solutions and an introduction to some popular machine learning algorithms that currently have not been utilised. A thorough analysis of the projects risks is considered as well as a business case which served as justification to pursue the project.

2.2 Literature Review Of Problem

Predicting the outcome of Football matches has proven difficult with millions of people losing money on bets to bookmakers across the globe on a daily basis. A variety of factors influence the outcome of Football matches, for example Football has become a business and thus increasingly competitive due to the financial rewards on offer (Dobson and Goddard, 2001). Unexpected events such as injuries to key players can impact the outcome (Constantinou et al, 2012) as well as pitch conditions, for example there is a link showing more goals per game are scored on pitches in good condition compared to those in poor condition (BBC, 2009). Football teams now compete in a multitude of competitions, thus rotating their squad and as a result player's fitness and fatigue can vary. Constantinou et al, 2012 identified fatigue as a factor which will impact the outcome of a match. Other factors, which make the outcome of football matches difficult to predict include varying form (Heuer and Rubner, 2009) and playing at home can be of advantage to a team, compared to an away fixture (Palomino et al 2001). Motivation for matches of varying importance will vary (Goddard, 2006), human errors such as referee mistakes have an impact and luck is an important factor in influencing the outcome of matches. All these factors consequently make predicting the outcome of a football match a difficult task, whilst allegiance to one's favourite team can adversely impact decision making when making predictions (Erceg and Galic, 2014). Technologist (2016) summarised the difficulty in predicting the outcome of Football matches citing:

"Soccer isn't an exact science. There's always an element of the unknown"

2.3 Comparison of Similar Solutions

As part of a comprehensive analysis, that would aid development of the best possible solution, research was undertaken of similar systems available, analysing their pros and cons.

2.3.1 Beat The Bookies Football Tips Website

This website was found via a Google search. It has been in operation since 2014 and has a very popular social media following, however no indication of the accuracy of predictions is included but there is a confidence scale of predictions provided.

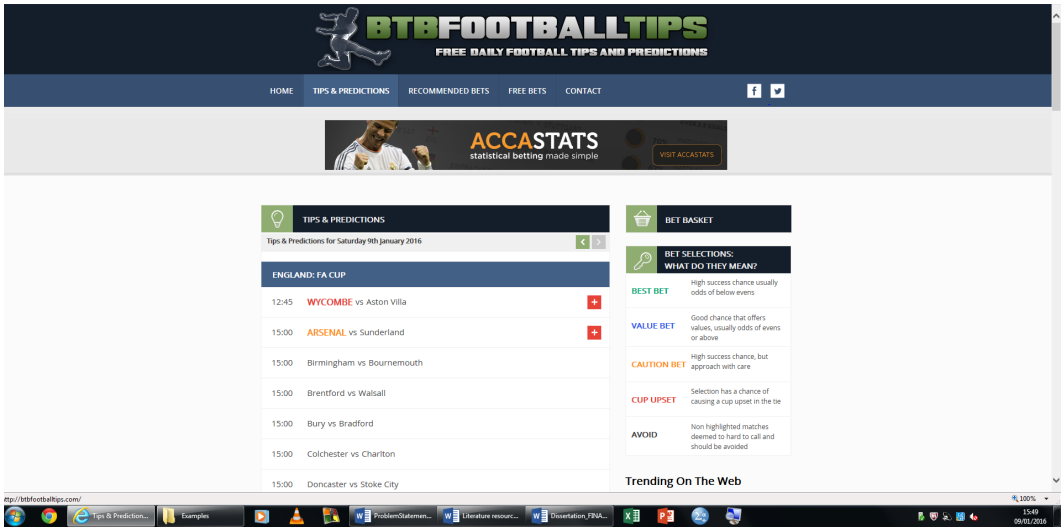


Figure 1: Screenshot of the Beat The Bookies website

Pros	Cons
Website is easy to navigate	No information as to how predictions are picked ie statistically or an expert's opinion
Recommendations are meaningful. Eg degree of certainty of bets is colour coordinated and a guide is provided	No information regarding previous accuracy
Wide range of competitions covered	An additional service, claiming to be more reliable and based upon statistical data is available with a subscription fee. However no information is provided if the data is up to date or what calculations have been performed on it
Reliable/web pages loaded fast	

Updated frequently	
Top picks section is a useful function for users	
Free of charge	

Table 1: Analysis of pros and cons of Beat The Bookies Football Tips

2.3.2 Football Foresight App

This app was found on the Apple App Store as it was the top search result for "football tips". It has a 4.5 / 5 star rating and was released in 2012. Exclusively available on iOS this system claims to be the top Football prediction app on the market.

Figure 2: Screenshot of the Football Foresight App

Pros	Cons
Website is easy to navigate	No information as to how predictions are picked ie statistically or an expert's opinion
High standard user interface	No information regarding previous accuracy
Informative eg displays when last updated	Does not include draws as a possibility in prediction
Predictions displayed as a percentage, which is helpful to users	Only domestic predictions free. In app purchases required to see foreign league predictions
Updated frequently	
Top picks section is a useful function for users	

Table 2 : Analysis of pros and cons of Football Foresight App

2.3.3 Win Draw Win Website

Also found from a Google search, Win Draw Win is a UK based company who offer Football predictions based on opinion.

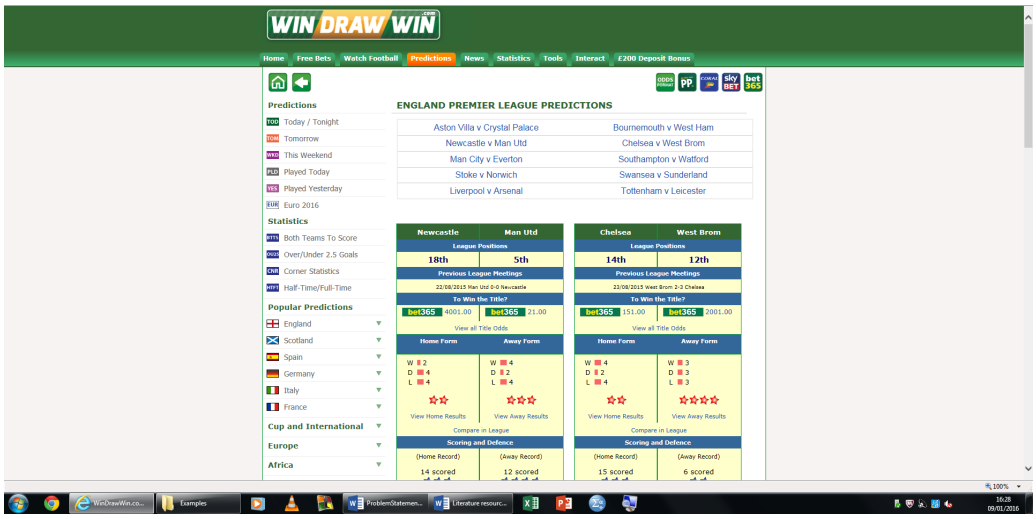


Figure 3: Screenshot of the Win Draw Win Website

Pros	Cons
Website is easy to navigate	Predictions are made based on opinion. No mention of how qualified/level of knowledge predictor has
Information provided as to how predictions are made	No information regarding previous accuracy
Provides further rationale for selections	Very basic interface. Not greatly attractive to user
Wide variety of prediction markets, other than just Win/Draw/Lose market	Contains a lot of needless information and excessive bookmaker advertising
Wide range of competitions covered	
Updated frequently	

Table 3 : Analysis of pros and cons of Football Foresight App

2.4 Alternative Solutions

Upon researching the issue at hand, a variety of alternative solutions were discovered that are not currently utilised by any software systems. These are highlighted within this section.

2.4.1 Probabilistic Models

Constantinou et al (2012) identified Poisson Distribution as a prominent approach to Football prediction through identifying the number of goals likely to be scored by

each team. To implement this probability distribution a large data set of previous results is gathered relating to the teams involved. Average goals scored per game and conceded per game are calculated for each team and compared to the league average for both prediction indicators. This generates attacking strength and defensive strength multipliers, which combined give the predicted score of any given Football match.

2.4.3 Machine Learning

A rising amount of research has been conducted aiming to predict the outcome of future events based on data from previous events. This includes financial markets prediction, medical prediction such as the outcome of Tuberculosis patients and the outcome of sporting events. Utilising Machine Learning techniques can assist this. Machine Learning, a subfield of Artificial Intelligence is the study of computer algorithms for learning to carry out tasks, in this case predicting the likelihood of the outcome of a future event (Football match). There are many machine learning methods available and an analysis of some of the most prominent methods was undertaken.

Bayesian Techniques

Bayesian algorithms are probabilistic models which offer a degree of flexibility to take into account alternative factors. The most popular are Naïve Bayes and Bayesian Network. These methods involve using predictive indicators and altering them through further research and experimentation allowing for a degree of flexibility. Joseph et al (2006) implemented Bayesian algorithms to predict the outcome of Tottenham matches over a season to a poor degree of success however Goddard (2005) found that there was little difference in prediction accuracy between Poisson Distribution and Bayesian Networks.

K Nearest Neighbour

Peterson (2009) suggests K Nearest Neighbour is one of the first Machine Learning models one should inspect when probability parameters are difficult to quantify, which can be the case with Football. It is a simple algorithm, for example if one wanted to investigate the probability of a Football match's most likely outcome they would make use of a series of data points. Figure 4 helps best illustrate the K Nearest

Neighbour approach, whereby there are 3 possible outcomes (W/L/D) and k is equal to 5. K can equal any number but it's function is to define how many nearest neighbours the algorithm is to search the distance for. The clear circle indicates the 5 (K) nearest neighbours to our query point (X), illustrating that 3 out of 5 are wins thus a 3:5 or 60% probability of the outcome being a win. As one of the five nearest points is a draw the probability of a draw is subsequently 1:5 or 20%.The distance of other data points from our query points can be measured using various distance metrics, such as Euclidean and Manhattan. The x and y axis represent prediction indicators (for example difference in average points per game and difference in league position) and can incorporate more than two.

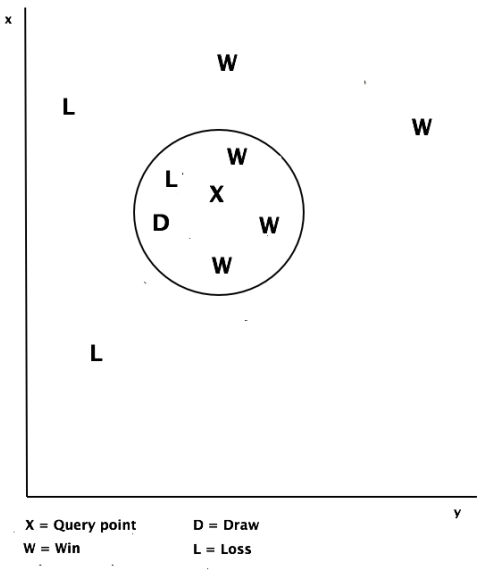


Figure 4: Diagram explaining K's Nearest Neighbour

2.5 Findings

Below is a summary of the key findings upon analysing similar systems:

- Given the vast amount of information they contain, some systems are difficult to navigate such as the Win Draw Win website
- It can be difficult to find an individual prediction that a user is particularly interested in
- A high proportion of similar systems appear cluttered due to too much information/advertising

- The higher quality systems such as Football Foresight offer win / lose probabilities. However they exclude the probability of the match ending in a draw. Beat The Bookies website also does not account for the possibility of the match ending in a draw.
- The vast majority of systems are based on opinion, with no mention of previous success rate
- The systems examined had varying accuracy
- Not all systems on the app market are free. Alternatively some such as Football Foresight and Win Draw Win offer premium features at a charge
- Browsing fixtures/predictions by league is a convenient way to find information the user wants easily as utilised by the Beat the Bookies site and the Football Foresight app
- Updating the system frequently is crucial
- Most popular European leagues are featured within systems. Beat The Bookies website provided this free, whereas Football Foresight charged for additional league coverage
- None of the systems reviewed provided a log in function. The systems functionality does not depend on this
- All systems that were researched were reliable, for example web pages loaded promptly and the Football Foresight did not crash or contain any obvious bugs upon thorough use.

Through clear identification of the problem and a comparative analysis of available applications it is evident that many systems in existence claim to solve the problem at hand, however none effectively address it. For example it is clear that an overwhelming majority of predictions that systems are providing are opinion based. Many others do not even mention bringing in to doubt how reliable they are.

2.6 Recommendations

Following the competitive analysis and findings in 2.5 the following points should be taken into account in the development of the proposed solution:

- The system should provide all content free of charge.
- The system should adopt a minimalistic design, avoiding a cluttered appearance that was found in many similar solutions.
- The system should make predictions availing of machine learning, given current solutions have failed to.
- Predictions generated should be based on relevant and up to date data
- After undertaking a comparative analysis of existing solutions, it was decided that as they all offer English Premier League predictions the system should also offer Premier League match predictions at the least.
- The system should offer predictions for all possible outcomes ie Home win, draw and away win.
- The system should be easy for end users to navigate.
- Speedy access to predictions would be advantageous to users.
- Information as to how predictions are generated should be available for users to review

2.7 Project Risk Analysis

As part of a comprehensive analysis phase an assessment of the physical, technical and project management risks associated with this project were explored:

2.7.1 Physical Risks

This is a standard project with no significant physical risks overall. The Health and Safety risk assessment form has been submitted.

2.7.2 Technical Risks

The Technical were assessed, as shown in Table 4. There were no intolerable risks which needed to be handled.

Risks	P	L	T	Preventative / Contingency Actions
Data loss	2	4	8	Data to be backed up on a regular basis to USB drive and external hard drive. Online cloud storage was also availed of for backups.
Computer Failure	1	3	3	Access to University of Ulster computer's arranged on multiple campuses.
Complexity of coding skills required	2	4	8	Additional effort invested in learning Java outside of university. Also invested additional study time in

				learning PHP.
Virus infecting system	1	2	2	Anti virus software purchased.
Software fails to meet end user needs	2	4	8	Agile methodology to be adopted which encourages continuous engagement with stakeholders.
Unable to implement a strategy for predicting scores	1	5	5	Various machine learning methods of score prediction researched such as Poisson Distribution and K-nearest Neighbour.

Table 4: Technical Risks Table of project

P = Probability on a scale of 1- 5: (Unlikely to likely)

L = Loss on a scale of 1 – 5: (Low to high)

T=Threat on a scale of 1 – 25 (Negligible to intolerable)

2.7.3 Project Management Risks

An Agile software development approach was used for the development of the project. A prototype was produced first and new, more complex, versions were subsequently developed. The system was tested as work was completed

The Project Management Risks were assessed, as shown in Table 5. There were no intolerable risks which needed to be handled.

Risks	P	L	T	Preventative / Contingency Actions
Not delivering the project on time	1	4	4	As part of an agile methodology sprints were undertaken and continually reprioritised. A target completion date was set before project deadline to allow for unscheduled delays.
Not delivering a project within financial budget	1	3	3	Use of free development kits
Procrastination	4	2	8	Adopted Agile Methodology that broke work into smaller iterations and created a timeline of completion dates.
Delivering a low quality project	2	4	8	Extensive stakeholder engagement, critical assessment of work and regular engagement with project supervisor

Table 5: Project Management Risks Table

P = Probability on a scale of 1- 5: (Unlikely to likely)

L = Loss on a scale of 1 – 5: (Low to high)

T=Threat on a scale of 1 – 25 (Negligible to intolerable)

2.8 Business Case Analysis

A business case analysis was undertaken to identify and quantify the perceived benefits of undertaking this project. The costs versus opoortunities of this project are outlined:

2.8.1 Costs

- Opportunity Cost ie wages foregone to complete project
- Software Cost = £0 (Free development kits available)
- Database Cost = £0 (MySQL database available from control panel on free hosting site)
- Data Costs = £0 (Data can be sourced for free from football-data.co.uk)
- Training Costs = £0

2.8.2 Benefits

- Will assist individuals in making more accurate football betting selections in bookmakers. Sports betting is a lucrative market, with Paddy Power recording record annual pre tax profits in 2014 of €167 million (RTE, 2015), to the detriment of consumer's. This project will help consumers make better and more informed selections, resulting in increased financial gain. The project may also be of interest to fantasy football players. Fantasy Football also has a large audience with over 3 million players registered with the Premier League's Fantasy League (Fantasy Premier League, 2015).
- In 2015 it was recorded that there were more than 1.4 billion Android devices worldwide(Android Central, 2015).Therefore depending upon the platform it is implemented on there is a huge reachable audience
- Mobile users are spending an average time on mobile apps of 2 hours and 42 minutes per day, which is more than that spent mobile web browsing
- The app industry has proven itself to be flourishing with Google Play paying out $3 billion to it's community of Android users in 2014 (The Guardian, 2015). The app economy is predicted to be worth £31 billion to the UK economy by 2025 (The Guardian, 2014).
- This project has potential to be monetised through subscription charges
- This project has scope to be monetised via advertisements. This is a viable solution with mobile ad market spending recorded at $18 billion in 2014 and forecasted to rise to $42 billion by 2017 (Tech Crunch, 2014). The BBC (2011) cites that 73% of app users have downloaded apps with ads in them. Such is the financial feasibility of mobile app advertising, Facebook reported 30% of

its advertising revenue in the first quarter of 2013 came from mobile (BBC, 2013).

- This project has an infinite app life, whilst professional football exists, thus maximising market potential.

It was concluded that the project benefits greatly outweighed the costs and the project was pursued.

2.9 Proposed Solution

After analysing similar systems project stakeholders were met to consider appropriate solutions to the task. This process began in December 2015 to ensure a thorough understanding of the strategies that were to be adopted in proceeding with development.

It was proposed that the solution software would implement machine learning to make predictions. The rationale for this was that it had not been used before in commercial prediction systems and that end users were interested in a system based upon up to date data, rather than opinions and guesses that rival software systems currently offer. Research also demonstrated that machine learning has been successful in prediction and pattern recognition in a variety of other fields. For example Amazon Machine Learning have used techniques to forecast demand and detect fraud. Wise.io have implemented machine learning to detect political party affiliation, predict whether an email is spam and IBM Watson have used it to implement medical diagnostics (Peddibhotla, 2015).

After considering prominent Machine Learning techniques such as Bayesian methods, the K Nearest Neighbour algorithm was selected for this project. There were a variety of compelling arguments which led to this decision. Firstly it is a very simple algorithm to implement. Also it has previously yielded encouraging results (Yezus, 2014) compared to Joseph et al (2006). It was noted with end users that this can be a less efficient algorithm, however it was an agreed trade off for accurate results, which end users identified was their main priority. The K Nearest Neighbour approach also provides flexibility and scope for altering and improving the algorithm. For example, any number of prediction indicators can be used and the indicators can be altered for trial and error purposes to find the best indicators to use.

For the purpose of this project three prediction indicators were selected. The diagram in Figure 4 shows the algorithm conceptually with two prediction indicators, however three prediction indicators were selected for additional accuracy

The values that were selected to be used in this system were:

- Difference in average points per game
- Difference in average goals scored per game
- Difference in average goals conceded per game

Via stakeholder engagement it was identified that form, defensive strength and attacking strength were three of the most important factors people considered important when predicting the outcome of Football matches. The above prediction indicators therefore were indicators of these factors and easily quantifiable.

2.10 Summary

It was important to extensively understand the problem before developing software as a solution and the knowledge gained from this analysis phase allowed for the software produced to effectively address the issue. Key issues highlighted as a result of the analysis phase were that there is a wealth of systems which claim to address the problem, however none do so effectively.

Requirements Analysis

3.1 Introductions

System requirements are an essential stage of the software development lifecycle, as they illustrate clearly what is expected of the final product. This chapter details the requirement gathering process, as well as the key findings. As a result user stories were created that allowed the functional and non functional requirements of the proposed system to be outlined at the end of the chapter.

3.2 Pre-requirements

Upon assessing the business case presented in the previous chapter, it was decided that the solution would be implemented as an Android mobile app. Alternative approaches considered included a website, i0S mobile app and a stand alone desktop application. The business analysis highlighted growth of smart phone app usage at the expense of web browsing, as well as the higher volume of Android devices currently being used, therefore an Android app was developed.

3.3 Requirement gathering

Requirements gathering for this project entailed extensive stakeholder engagement. End users who expressed an interest in using the proposed product were approached and collaborated with to gather functional and non functional requirements. The first end user engagement took place in December 2015, which consisted of discussions of what they would like the product to do. This enabled goals and objectives to be established early and progress was documented. This process required extensive discussions and a focus on the requirements of stakeholders, rather than what the chosen development environment happens to do best. Also in any project it is rare to gather all requirements immediately, thus engagement was revisited frequently and reassessed.

3.4 User Stories

As a result of end user engagement user stories were produced as part of the agile methodology adopted.

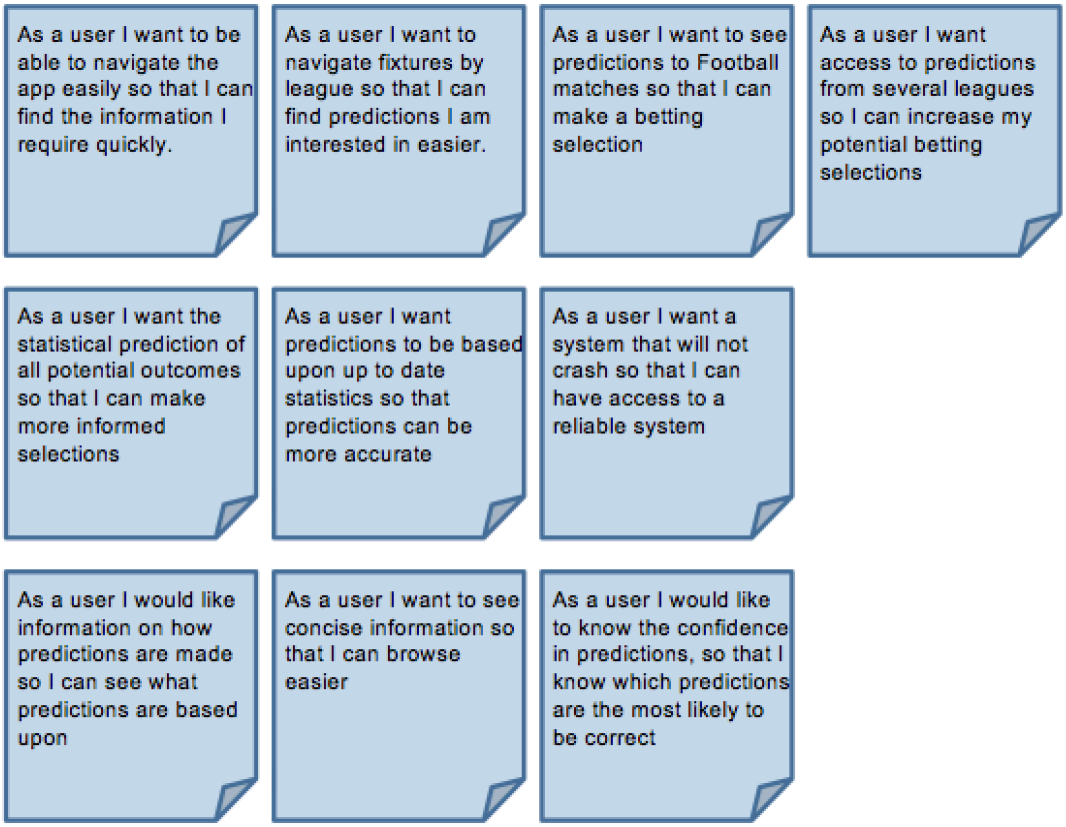


Figure 5: User stories from user engagement

3.5 Functional Requirements

Functional requirements are a description of the facility or features required. They deal with what the system should do or provide for users(Sqa, 2007). The following functional requirements were established after discussion groups between the developer and end users.

1. The app must access a database of football statistics
2. The app must avail of a machine learning algorithm
3. The app must calculate and display the possibility of all outcomes of a football match (ie Home Win, Draw, Away Win)
4. The app must contain information on how predictions are generated
5. The interface should be easily navigated
6. The app should show predictions for football matches from more than one league
7. The app must use up to date data
8. The app must have a modern and attractive, yet simple interface
9. Predictions should be generated within 3 seconds

10. Predictions should have an average game week accuracy of 60%

3.6 Non Functional Requirements

Non functional requirements describe a system rather than the individual functions it will contain. The following non functional requirements are incorporated in the design of the system, in order to deliver a high quality solution:

11. Robustness: The app shouldn't crash on users or be entirely affected by a software bug
12. Extendibility: The app should have scope for future enhancements / features
13. Compatibility: Should be usable on a number of devices, for example phones and tablets
14. Efficiency: Code and algorithms used should be efficient to ensure high performance of the system ie predictions are loaded quickly. The solution must collect and store data efficiently also
15. Consistency: There should be a consistent interface theme throughout in order to deliver a high standard interface
16. Suitability: The app should be suitable for use for a varying degree of abilities. Thus it must be easy to understand for users.
17. Correctness: Numbers provided in the app should be meaningful to users via guidance
18. Usability: Users of the app should be able to easily obtain the information they require
19. Availability: The app should be operational/available for use when required, for example throughout the Football season

3.7 Summary

This chapter describes the requirement gathering process undertaken by the developer and the stakeholders that were engaged. The requirements were split into functional and non functional and were incorporated into the design and implementation of the final solution. The requirements were also driven by the analysis phase of the project and all requirements were validated by end users.

Design

4.1 Introduction

This chapter presents an outline of the design process undertaken before the implementation phase of the project. Specifically it offers an insight of the design guidelines and best practices that influenced the user interface, as well as initial mock ups. As part of best practice the architectural design was also considered throughly before implementation and is summarised in this chapter.

4.2 Mobile App Development Guidelines

From an early stage in development it was essential to appreciate that there is a range of devices the app could be viewed on, including mobile phones and tablets that offer different screen sizes, capabilities and processing power.

Similarly, it is important to consider that the deployed app would be used by a wide audience and to ensure an optimal experince for end users mobile development guidelines were considered carefully throughout the design and implementation phases. The guidelines that were referred to are summarised below.

4.2.1 W3C Mobile Accessibility Initiative

The World Wide Web Consortium (W3C) is an international community were Member organizations, full-time staff, and the public work together to develop Web standards.

They set out the following guidelines which aids the development of online content for all, including end users with disabilities (W3C, 2011). Throughout the design phase, developing for a broad spectrum of users was of high importance, as the importance of equal opportunities for all of society was recognised.

"The internet is a critical public resource important to everyone, and everyone is entitled to access to it" (United Nations, 2016)

Perceivable

- Provide **text alternatives** for non-text content.
- Provide **captions and other alternatives** for multimedia.
- Create content that can be **presented in different ways**, including by assistive technologies, without losing meaning.
- Make it easier for users to **see and hear content**.

Operable

- Make all functionality available from a **keyboard**.
- Give users **enough time** to read and use content.
- Do not use content that causes **seizures**.
- Help users **navigate and find content**.

Understandable

- Make text **readable and understandable**.
- Make content appear and operate in **predictable** ways.
- Help users **avoid and correct mistakes**.

Robust

- Maximize **compatibility** with current and future user tools.

Figure 6: W3C Initiative Content Accessibility Guidelines

Upon considering the above guidelines it was decided error handling and informative messages would be included in the solutions design. Also text was considered carefully and end users were consulted to ensure it was easily readable. The design strived to make content predictable and overall the system was designed to minimise potential mistakes made by end users, as no user input is required.

4.2.2 BBC Mobile Accessibility Guidelines

The BBC Standards and Guidelines for Mobile Accessibility are a set of technology agnostic best practices for mobile web content, hybrid and native apps (BBC,2013). The standards and guidelines are organised into 11 topics as illustrated, with relevant sections being taken into account such as design and images.

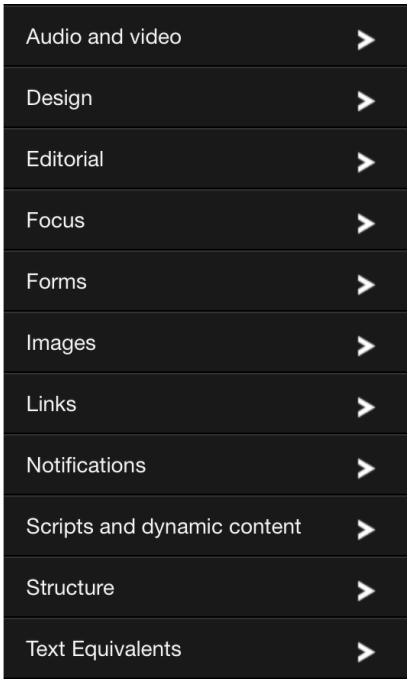


Figure 7 BBC Accessibility Guidelines Topics Considered

4.2.3 AQuA Best Practice Guidelines

The App Quality Alliance Best Practice Guidelines were consulted throughout development and found to be quite similar to the previous two guidelines explored. The App Quality Alliance are a non profit group headed by members and knowledge contributors, working with industry to improve the quality of mobile apps (AQuA, 2013). The document covers extensively best practices in Software Engineering for mobile devices and acknowledges the need for apps to behave in a predictable manner.

4.2.4 Findings From Guidelines

From consulting the identified development guidelines it was acknowledged a variety of factors result in Software Engineering for mobile apps differing from "traditional" Software Engineering. These factors were taken into account throughout design:

- Mobile apps are used on the move in a variety of environments. Users are continuously distracted by environmental influences such as noise and light,using when walking. This differs greatly from a desktop
- Input is different. Mobile apps have no keyboards / mouse etc. Often mobile devices avail of touch input. Therefore there are no hover events, there is

less precision than a mouse and gestures such as swiping need to be incoporated.

- Battery power is considerably less on mobiles and tablets than a laptop or desktop therefore code must be efficient
- Mobile devices such as phones and tablets have less processing power than desktops and laptops
- To be accepted in their respective stores, mobile apps must conform to development guidelines
- Wireless networked connections present security issues
- A quality app should have an equally high standard of interface on different screen sizes
- Most mobile cellular links have much less mb/s than a home link. Mobile Apps needs to be designed not to consume as much or users will be put off. Therefore consider the volume of images in the app, the file extension they are saved as and avoid video content wherever possible

4.2.5 Recommendations From Guidelines Taken Into Consideration

After considering the various design guidelines and best practices consulted, as well as the findings in section 4.2.4 the following design aspects were to be included:

- Minimise the amount of information on each page compared to a desktop/laptop (W3C, 2015)
- Use effective contrasting, as mobile apps are more likely to be used in varied environments such as outdoors were strong lighting sources are more likely such as glare from the sun
- Attempt to support both screen orientations, for example not everyone can rotate their device screen. Some users may have their device mounted in a fixed orientation, such as on the arm of a power wheelchair (W3C, 2015)
- Placing buttons where they are easy to touch. This is not just to make it easy for end users to perform actions but it can also benefit end users with diasbilities, for example someone who only has one hand available. Consideration was also taken that an easy-to-use button placement for some

users might cause difficulties for others e.g. left- vs. right-handed use (W3C, 2015)

- Elements that trigger action will be sufficiently distinct (W3C, 2015)
- Adequate instructions must be provided (BBC, 2013)
- Clear error messages should be provided (BBC, 2013)
- The user interface should be consistent and understandable throughout, e.g. displaying a common series of actions, action sequences, terms, layouts, soft button definitions and sounds that are clear and understandable (AQuA, 2013)
- The application should display graphics, text and other UI elements without noticeable distortion, blurring or pixilation (AQuA, 2013)
- Menu items like Help and About should be presented on the main menu or other easily-found screen of the application (AQuA, 2013)
- Minimising text input helps prevent mistakes and possible barriers to activating and using content. Typing on mobile devices can be difficult and make take longer for screen reader users. When direct input is not required provide options for the user to achieve the same result (BBC, 2013)

4.3 System Navigation

As identified in the analysis phase and after consulting development guidelines and best practices, development of a system which is easily navigated is crucial, as it allows users to access the content they wish quickly. If the app is not easy to navigate then it is likely users will be less willing to use it as end users want information as quickly as possible. The aim was therefore to allow the user to access content they are interested in by visiting minimal pages within the app. As demonstrated in Figure 8 users can view match predictions after 3 button touches.

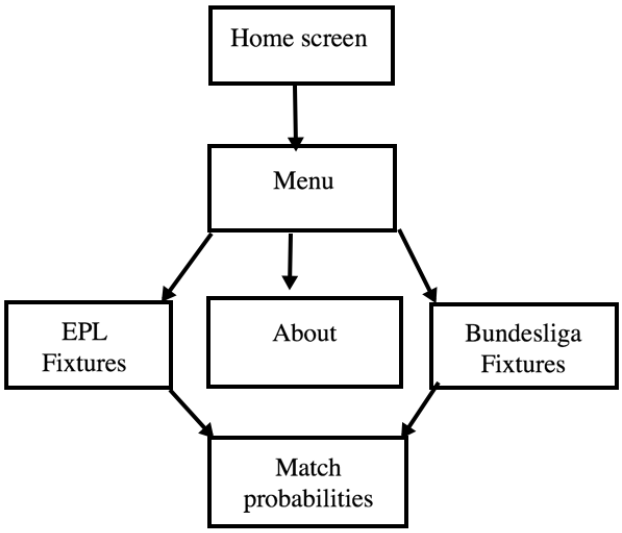


Figure 8 Organisation structure of app

As illustrated above users access an initial home screen which displays an app logo and a button to accress the app menu. The user is taken to a new activity which offers them the ability to view English or German League fixtures or an about page which contains information on prediction generations and a disclaimer. Upon choising a desired fixture, the probabilities of each outcome of the chosen fixture is displayed on a new screen. The decision to group predictions by League was influenced by the comparative analysis of similar systems, as it was found this worked well in thr Football Foresight App and Beat The Bookies Tipster Website. Generating all results together was considered, however given the user guidelines of limiting the amount of text displayed on a page that was disregarded.

4.4 Graphical User Interface Design

The graphical user interface (GUI) was designed with the aim of being as simple to use as possible and consistent in terms of design to make it easier to use and look professional. This was assisted by consulting Shneidermans 8 Golden Rules of Interface Design (Shneiderman, 1998).

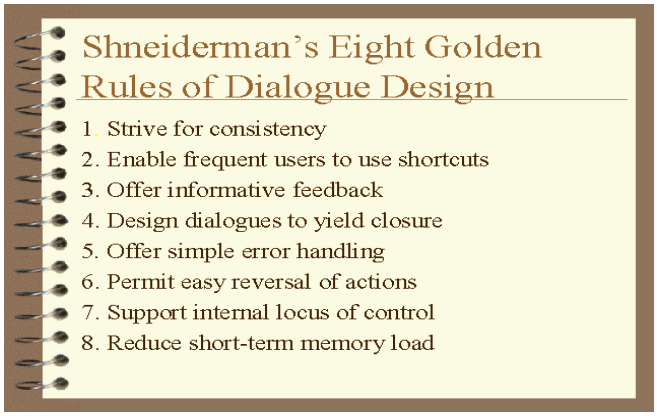


Figure 9 Shneidermans Golden Rules

Table 6 summarises how the design process adhered to Shneiderman's principles of design.

Guideline	Action
Strive for consistency	The entire app follows a green colour scheme with a contrasting white font. All league pages follow the same layout, as do pages displaying the most likely outcome of matches. It is anticipated that a user will be able to use one part of the system extremely quickly and transfer that knowledge to the whole system.
Enable frequent users to shortcuts	The organization structure eradicates the need for shortcuts
Offer informative feedback	Significant operations requiring feedback are minimised. The system will always communicate where the user is located and messages are output when predictions are loading. Error messages are also used when appropriate
Design dialogs to yield closure	Sequences of actions are organised into groups ie Home Screen, Choose a league to view predictions, Choose a match to view it's prediction
Prevent errors	The system was designed so that the only input from users is navigational, therefore the likelihood of user errors is minimised
Permit easy reversal of actions	The system is designed so that if an error is made in navigation it can be rectified by simply pressing the back button
Support internal locus of control	The system is designed to make users the initiators of actions rather than the responders
Reduce short term memory load	Users are not required to remember information from other screens

Table 6 : Actions taken to comply with Shneiderman's Interface Principles.

4.4.1 Interface Templates

After considering design guidelines and the overall structure of the app, design templates were created. This gave an reperesentation of the interface design and allowed for consideration of different colour schemes and layouts. This process was also useful in ensuring that the interface design was consistent throughout as required. The layout design for each screen within the app is outlined:

Home Screen

Upon loading the app this is the first screen that users will see. It consists of an app logo and a button that when pressed takes the user to the menu screen. Originally it was planned that this screen would consist only of a logo and a button that could be pressed to display a hamburger / slider menu, as shown in Appendix C. Upon further consideration and engagement with stakeholders the hamburger style menu was omitted from design. Research found slider menus to offer low discoverability and to be less efficient. Also upon conducting market research of other prominent apps Facebook have recently ceased using it as a primary navigation tool.

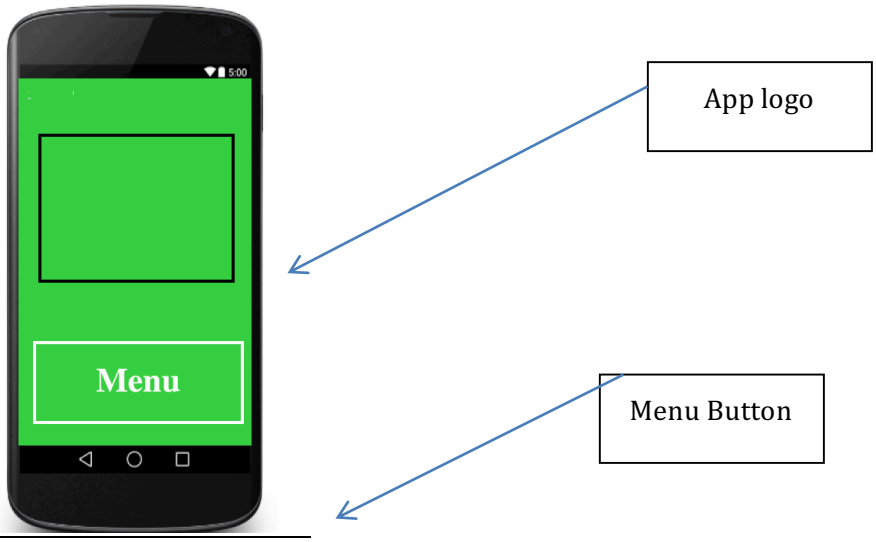


Figure 10 Home Screen design

Menu Screen

This is the second screen users wll be presented with. It contained three pressable buttons that enable the user to access the key functions of the app. The buttons were designed to be large enough to minimise the margin for error when making a

selection, as there is less precision when using a touch screen than typical input means such as a mouse.

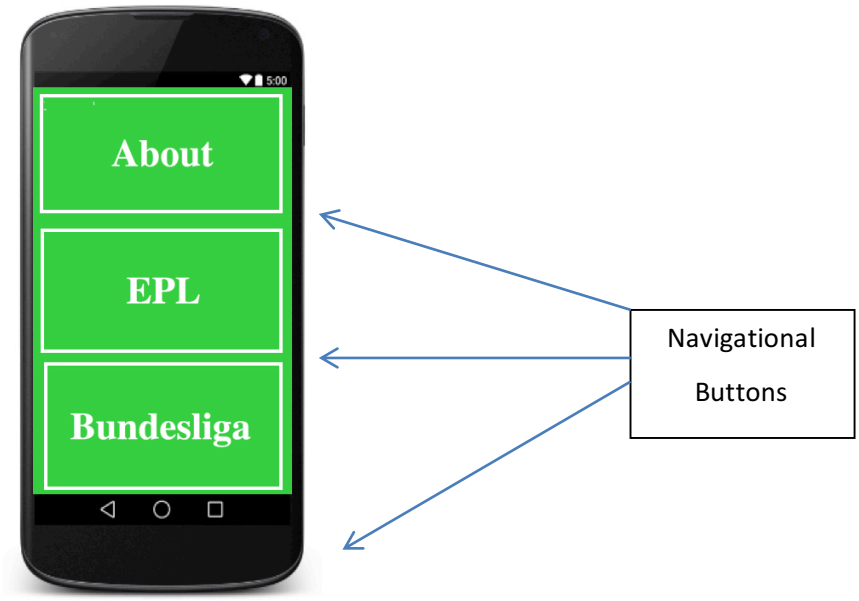


Figure 11 Menu Screen Design

About Screen

As per the functional requirements and analysis stage it was decided that this system would contain information about how predictions were made, as well as a disclaimer as part of ethical requirements. Originally it was decided they would be two separate pages within the app however upon further consideration it was found both sections were concise and it would be better suited to combine them, allowing users to access information quicker. This decision was validated by end users and incorporated into design.

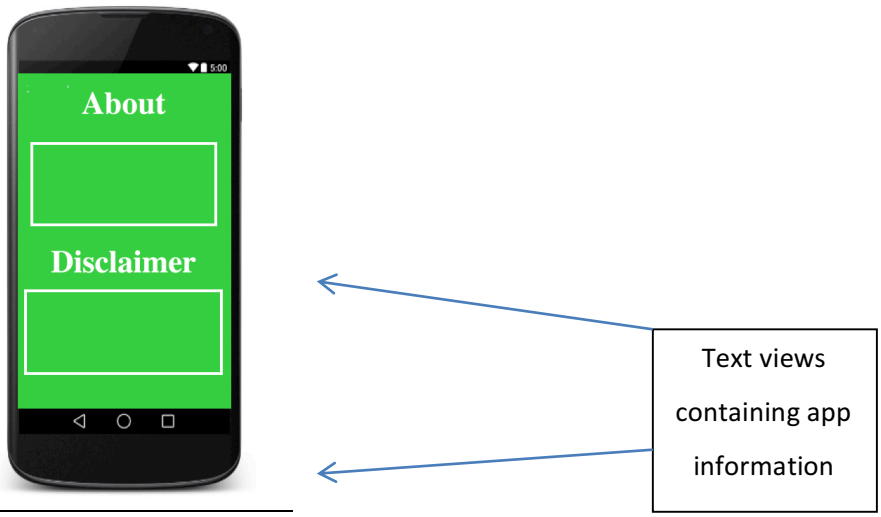


Figure 12 About screen design

Fixtures Screen

Upon making a menu selection of Bundesliga or English Premier League, the available fixtures to select are presented to the user from each league respectively. All fixtures are clickable and take users to the next screen which shows probabilities of matches of interest.

Figure 13 Fixtures Screen Design

Probability Screen

After pressing a fixture on the previous screen, the results screen displays the probabilities of each possible outcome of a game. This screen was kept as simple as possible to allow users to easily obtain the information they seek. The back button can then be pressed on the users device to select other fixtures. The layout of this screen serves to highlight the most important information and ensure it stands out, as it was noted users want to spend as little time as possible browsing information.

Figure 14 Probablity Screen Design

As shown in Appendix C the very first concept design for this app consisted of a darker green. As part of an agile approach to development this was constantly reconsidered. After discussion with end users it was agreed that green was still a valid colour theme, however it was decided a different shade of green was deemed more suitable. Hexadecimal colour code #83F52C was selected, as well as white for text/buttons. This change was made as it provided a more vibrant colour scheme, made the app more modern, whilst still reflecting the Football theme of the app. As per Schneidermans Golden Rules a consistent layout and colour scheme was adopted. For example, the Premier League and Bundesliga Fixture pages had identical layouts.

4.5 Architectural Design

The system was designed using a multitude of programming languages and tools that interact with each other to provide the overall functionality of the product. How the various technologies and tools interact with each other is explored in this section of the chapter

4.5.1 Client / Server Relationship

A server is an instance of a computer program that accepts and responds to requests made by another program, known as a client (Computer Hope, 2016). It can also provide computation for clients, such as algorithms as well as well as managing databases and file applications.

In terms of the client server model the client is a user interacting with the server via another program. In this system the client is an app user, however clients can use other means to interact with a server such as a standalone desktop application.

Given that the app depends on the server to perform tasks, it is crucial that the server is reliable.

This system accesses data held in a database on an online server availing of a client server relationship. This data is converted into a format readable by the Android application by displaying the data as JSON objects. A PHP script takes data from the server and displays it in JSON. The PHP script acts as a link between the app and the database.

Figure 15 System architecture

4.5.2 Programming Languages Used : Java, PHP, SQL, XML

Structured Query Language (SQL) is a standard language for accessing and manipulating databases. It enables a broad range of tasks to be carried out such as creating a new database, creating new tables within a database, inserting records, deleting records, updating records, retrieving data and executing queries against a database.

Hypertext Preprocessor (PHP) is an open source HTML embedded server side scripting language which is used to develop dynamic and interactive web applications and also used a general-purpose programming language. A lot of the syntax is borrowed from other programming languages such as C and Java. As it is a server side scripting language, all of the code in PHP is processed on the web server rather than the clients app (W3Schools, 2014). PHP was chosen to be used as research was undertaken and many sources led to the conclusion it was easy to learn, contained familiar syntax, is free of cost, provides efficient performance and also offers a helpful community for troubleshooting when problems arise.

Java is a programming language, originally released in 1995. It is Object Oriented and therefore everything is an object. It offers a range of benefits to the developer such as high performance and multithreading. Large parts of Android development are written in Java and it's API's are designed to be called primarily from Java. Java is the official programming language of Android, however it is possible to develop Android in C but it is not encouraged by Google (Android Authority, 2016).

Extensible Markup Language (XML) was used to create a user interface for the Android app. It is a software- and hardware-independent tool for storing and transporting data. Android user interface elements are declared within XML files. The advantage to this is that it enables you to better separate the presentation of your application from the code that controls its behavior. Your UI descriptions are external to your application code, which means that you can modify or adapt it without having to modify your source code and recompile. For example, you can create XML layouts for different screen orientations, different device screen sizes, and different languages (Android Developers, 2016).

```
LinearLayout
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="16dp"
    android:orientation="vertical"
    tools:context="com.example.marcmcnally.predictiondemo.menu"
    android:background="#33cc33">


    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_weight="0.33">

        <Button
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:text="About"
            android:id="@+id/button2"
            android:layout_centerVertical="true"
            android:layout_alignParentStart="true"
            android:background="@drawable/outlinebutton"
            android:textStyle="bold"
            android:textSize="50dp"
            android:textColor="#ffffff" />
    </RelativeLayout>
```

Figure16 User interface elements declared within an XML file

4.5.3 Database Design

In order to perform the K Nearest Neighbour algorithm and predict the outcome of Football matches based on previous events a database was required to store previous results and related information.

The database consisted of 2 entities:

- Premier League Results
- Bundesliga Results

Each entity within the database had the same structure, consisting of 4 attributes:

- Difference in average points per game

- Difference in average goals scored per game
- Difference in average goals conceded per game
- Outcome of game

The 3 prediction indicators used were all of Float data type. This data type was selected as it would allow for greater accuracy than an integer when generating predictions and also enabled predictions to be displayed to two decimal points, representing the percentage possibility. Outcome is a text value and contains one of the three possibilities of a Football match:

- W representing a home win
- D representing a draw
- L representing an away win

Within this system the database is used only for storage. It is not a relational database, as there is no link relationships/links between the two tables. There was also no need for a primary key as no record is required to be uniquely identified and data did not need to be normalised as there were no dominating prediction indicators.

4.6 Summary

This chapter discusses the various design guidelines and best practices used prior to implementation, as well as the specific design choices made as a result. An overview of the system navigation was provided, as well as user interface designs and the rationale behind decisions taken. System architecture is covered too, mentioning the programming languages used and the chapter concludes with the database design.

Implementation

5.1 Introduction

This chapter provides an overview of the technologies and platforms used to develop the prediction app. It notes other alternatives that were considered, as well as the development framework followed. The rest of the chapter looks at how the key functionalities of the system were implemented and the choices which influenced them.

5.2 Implementation Platform

As mentioned in the previous chapter this app was implemented using Java, PHP and MySQL. To do so a range of platforms were considered carefully. The platforms used are outlined.

5.2.1 Android Studio

Although Eclipse was also considered to develop the Android app, Android Studio was chosen as it is now the official IDE for Android, offers advanced code completion options and greater user interface customisation. Development experience had already been gained using Android Studio, therefore there was less of a learning curve than other platforms, allowing more time to be invested in development.

5.2.2 Text Edit

Used for creating PHP scripts. Whilst other text editors were considered, such as Espresso, Text Edit was pre installed on the laptop used for development free of charge and deemed fit for purpose.

5.2.3 000WebHost

000WebHost was used for server space for this project. This was chosen ahead of other server implementations as it was convenient and met the requirements of the project. Upon researching other sites offering server space 000WebHost was chosen for a variety of reasons:

- Free of cost
- 99% Uptime Guarantee (This instilled confidence as reliability was a key issue as per user requirements)

- Full PHP and MySQL support
- 1500MB disk space
- Positive feedback from other users

As illustrated the acquired server space met all the requirements of the project and as it provided access to two MySQL databases which allowed the projects databases to be implemented on 000WebHost. Research was undertaken and it was noted that there were alternative database possibilities to MySQL but it was selected due to previous experience and that there were no elaborate requirements from the database other than simply for storage. PHPMyAdmin which was provided by the web host enabled the creation of database tables using a Graphical User Interface. It also facilitated further moderation of the database tables, including importing CSV files into results tables.

5.3 Development Methodology

Prior to commencing development, it was essential to select a development framework. Research was undertaken and despite considering adopting the traditional Waterfall method, an Agile approach to Software Development was selected. Agile is a methodology that promotes continuous iteration of development and testing throughout the software development life cycle of the project. Both development and testing activities are concurrent unlike the Waterfall model (Guru99, 2016).

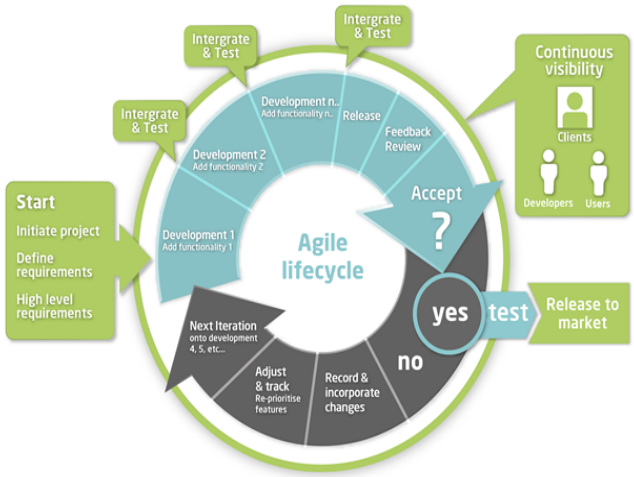


Figure 17 Agile Overview

Despite Agile being a largely team oriented principle, it was chosen for a variety of reasons:

- Agile is better suited to projects with changing requirements. Needs and requirements of this project evolved as the project progressed, therefore Agile was selected as it offered greater flexibility.
- It allowed end users to see a partially working solution from an earlier stage in development
- There is less focus on doucumentation and a greater emphasis on working software
- Gives flexibility to developers to prioritise tasks, if agreed upon with stakeholders
- Higher quality work is produced as features are tested as they are produced. Bugs and defects are identified early and can be rectified at an early stage
- Agile often delivers greater satisfaction to clients as they are involved continuously throughout, therefore they are frequently able to critically assess work undertaken
- Agile techniques virtually eliminate the chance of absolute project failure as you always have a working project from the first sprint (Dummies, 2016)

Figure 18 Agile Manifesto

Through using an Agile approach to Software Development a minimum viable product was identified. This was presented to project stakeholders in June 2016 and then changes were made after user consultation and extra features were added in accordance with the functional requirements section. As mentioned development was undertaken in sprints / iterations. Project stakeholders were engaged

throughout the development process and had a chance to offer feedback after each sprint. A project backlog which detailled sprints to be undertaken and their priorities was constantly reevaluated with stakeholders, therefore project stakeholders were critical to the progress and quality of development.

5.4 Sprints

The remainder of the chapter discusses the key iterations that were completed and the functions added to the project chronologically. After each sprint was completed project stakeholders were presented the work undertaken. These meetings influenced the order of development iterations.

5.4.1 Database Implementation

As identified in the analysis chapter a database of ten years of results for the English Premier League was required. For each match the prediction indicators before the game was played and outcome needed to be entered for all games played. In order to do so this required that a league table was caclulated and updated as matches were played. It was decided that the first five games each team played per season would be ommitted from the database records as they would not provide meaningul data. It was also decided that the last five games per team of a season would be ommitted from database records as motivation at this stage of a season can vary per team and thus the data may be misleading.

To implement this task a league table was calculated by the developer in Microsoft Excel and the prediction indicators were calculated by hand also and entered into the database using SQL statements.

	DiffAveragePointsPerGame	DiffAverageGoalsScoredPerGame	DiffAverageGoalsConcededPerGame	outcome
	0.4	0.6	0	D
	0.6	2.4	0.6	w
	-0.6	-0.2	0.4	w
	-0.4	-0.2	0.4	L
	0.2	-0.8	-1	D
	-0.4	0	0	w
	1.2	1.2	-1.6	w
	-1	-0.6	0	D
	0.4	0.4	0.6	w
	0.4	-0.2	-1.2	w

Figure 19 Database containing data manually entered

This was an extremely time consuming task. When presented to stakeholders concerns were raised that this method may not be accurate and also did not meet

the requirement of the system being extendable. After discussions with stakeholders it was agreed that a Java program was to be created that would read in a text file of previous results and output the database records to a text file that could be imported into a database. This would enable the app to be easily updated in future, meeting system requirements already outlined.

5.4.2 Algorithm Implementation

After stakeholder engagement identified the need for a program to automatically generate database records it was decided that the original database could be used as test data. Implementing the K Nearest Neighbour took the highest priority and was the next key iteration. The rationale was that it would bring the mobile app closer to the minimum viable product and it was the core function of the app.

This feature is implemented in the server side of the system, via PHPand and the Android app is mainly used for display purposes. The justification for performing the algorithm sever side was that the K-Nearest Neighbour algorithm has to traverse through a large volume of database records and perform computation. If the algorithm was performed in the app a large volume of database records would have to go to the app and this would be a very slow process / waste of data. Therefore the decision was taken that the algorithm would run server side and that the app would only receive prediction values.

To generate a prediction for a given fixture the values for the three prediction indicators are sent from the app. This is illustrated in Figure 20 and shows the difference in average points per game equals 1.3, difference in average goals per game equals 0 and difference in average goals conceded equals -0.5.

```
Toast.makeText(getApplicationContext(), "Wait....", Toast.LENGTH_LONG).show();
if (arg2 == 0) {
    teamplay = "Arsenal vs Stroke City";
    new DownloadJson().execute("1.3", "0", "-0.5");//Download Json is an Async task
```

Figure 20 Parameters sent from app

The parameters are sent to the server and the following code on the php script gets them.

```
//takes parameters sent by app
$diff_avg_pt=$_GET['diff_avg_pt'];
$diff_goal=$_GET['diff_goal'];
$goal_conceded=$_GET['goal_conceded'];
$team=$_GET['team'];
```

Figure 21 The server getting parameters

A query is then run to get all records in the database table, which contains all results of league matches for a decade.

After this a loop is run for all records and the difference between each parameter and every database record is squared. This helps calculate the Euclidian Distance for every database record from our query point / parameters.

```
//knn implementation is below
//take square of difference of input and database values
for ($p=0;$p<$i;$p++)
{
    $d1=($diff_avg_pt-$data[$p][0])*($diff_avg_pt-$data[$p][0]);
    $d2=($diff_goal-$data[$p][1])*($diff_goal-$data[$p][1]);
    $d3=($goal_conceded-$data[$p][2])*($goal_conceded-$data[$p][2]);;
    $d4=$d1+$d2+$d3;
    $kn[$p]->difference=$d4;
    $kn[$p]->index=$p;

}
```

Figure 22 Calculating Distances of Database Records

For each row of the database:

- d1=square(Difference in average points per game(sent from app)-Difference in average points per game(each row of database))
- d2=square(Difference in average goals scored per game(sent from app)-Difference in average goals scored per game(each row of database))
- d3=square(Difference in average goals conceded per game(sent from app)-Difference in average goals conceded per game(each row of database))
- d4=d1+d2+d3.

D4 tells the system how close each database record is to our query point. A knn class file also exists and contains an index attribute and a difference attribute. The index of each record and distance from the query point for each record is stored in an array of knn objects.

The array of knn objects is then sorted in ascending order on the basis of distances from the query point. A loop is then ran to the value of k. In this case k is 100. The loop then inspects the fourth column of each record (i.e win / lose / draw) and counts the number of occurrences of each possibility within the first 100 objects. By

adding the number of occurrences and dividing by K (100) the probability of each possible outcome is calculated. This probability is returned from the php page in JSON format. This process is repeated for each match request and the outcomes are displayed in the result activity. Appendix D demonstrates an overview of the algorithm implementation.

This iteration was then shown to project stakeholders. It was identified that the processing of results was quite inefficient as the distances for so many database records were calculated. Also, the sorting method used was bubble sort, as it was easily implemented, however it was acknowledged this is a relatively inefficient sorting algorithm.This caused results to take approximately 6 – 7 seconds to display which led to reservations amongs the relevant stakeholders. This was a trade off for accuracy and control over speed, as accuracy was the main priority of the project. However, algorithm improvements and changes were placed on the project backlog and constantly assessed in order to provide the best end user experience.

5.4.3 Automated Database Generation

As mentioned in 5.4.1 it was agreed with stakeholders that a stand alone Java program to generate a text file capable of being imported into a database was necessary as it made the system more extendable. This sprint was undertaken immediately after the algorithm implementation as it was deemed a key priority within the project.

This program contained three classes:

- Results
- FileData
- FootballLeagueData

The project gets input from a csv file containing all the results in a seasons and stores them in an array list. Then for each team it makes another array list storing information related to the league table such as team name and games played etc.

The program itereates through each row of results then gets the home team and the away team and inserts it into the array list containing league table information. If any team is found again then it’s record is just updated by adding goals scored, conceded, points and number of matches plated.

The main method (contained in FootballLeagueData class) runs and it calls getDataFromFile() method. This method will read the input text file line by line and gets the data and converts it into an instance of the FileData class. This is then inserted into an ArrayList.

The main method then calls the makeResult() method. This method will make "resultList" which contains team's calculated data.

It then prepares an output csv file that will contain the data required to be imported into a database.

Then a for loop (i=0;i<ls.size();i++), iterates through each index of "ls" (the ArrayList containing match results) and gets the home team and away team.

When "home_team" and "away_team" are not in the "resultList" (checked by loop), then they are inserted for the first time with fixed values for example:

```
hteam.setName(home_team);
hteam.setScored(ls.get(i).getFTHG());
hteam.setConceded(ls.get(i).getFTAG());
hteam.setPlayed(1);
// setting points
if(Objects.equals(ls.get(i).getFTR(),"H"))
    hteam.setPoints(3);
else if(Objects.equals(ls.get(i).getFTR(),"D"))
    hteam.setPoints(1);

resultList.add(hteam);
```

Figure 23 Inserting team into resultList

When a team already exist a loop will get the location and then update the relevant data as shown below:

```
else
{
    int index = 0;
    int awayteam_index = 0;
    for(int j=0;j<resultList.size();j++) // finds where the team name occurs in result list
    {
        if(Objects.equals(resultList.get(j).getName(),home_team)) {
            index = j;
            break;
        }
    }
    for(int a=0;a<resultList.size();a++)
    {
        if(Objects.equals(resultList.get(a).getName(),away_team))
        {
            awayteam_index = a;
            break;
```

Figure 24 Finding the location of a team in resultList

By doing so we get the location of the home team and the away team in “resultList” and then calculate the differences in the relevant values and write it to the output csv file. Then it updates the relevant data by overwriting the previous data.

After doing so, the league table is then updated as shown below:

```
    //update league table values
    resultList.get(index).setScored(resultList.get(index).getScored()+ls.get(i).getFTHG());
    resultList.get(index).setConceded(resultList.get(index).getConceded()+ls.get(i).getFTAG());
    resultList.get(index).setPlayed(resultList.get(index).getPlayed()+1);
    if(Objects.equals(ls.get(i).getFTR(),"H")) {

        resultList.get(index).setPoints(resultList.get(index).getPoints() + 3);

    }
    else if(Objects.equals(ls.get(i).getFTR(),"D"))
        resultList.get(index).setPoints(resultList.get(index).getPoints() + 1);
}
```

Figure 25 Updating league table values

Similar code then updates the away teams record in the league table. The output of the program is displayed as follows:

```
0.00,0.00,0.00,H
0.00,0.00,0.00,A
0.00,0.00,0.00,D
0.00,0.00,0.00,H
0.00,0.00,0.00,H
0.00,0.00,0.00,A
0.00,0.00,0.00,A
0.00,0.00,0.00,A
0.00,0.00,0.00,H
3.00,1.00,-3.00,D
0.00,1.00,1.00,D
0.00,1.00,0.00,D
0.00,-1.00,2.00,H
-2.00,0.00,1.00,A
-3.00,-4.00,2.00,A
2.00,1.00,-2.00,D
0.00,-3.00,0.00,A
0.00,0.00,3.00,A
0.00,-0.50,0.50,H
-1.00,0.00,-1.00,A
0.00,2.00,0.00,H
```

Figure 26 csv output

The csv file as shown above is then imported into the MySQL database. The IDE console also shows data for each team so that values can be entered into the app to be sent to the server to generate predictions (Figure 27).

```
ootBallLeagueData (3)
/Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin/java ...
Team                Played  Scored  Conceded    Points  Av Points   Aver scored Aver goals conceded
Bayern Munich        34      80        17        88      2.59        2.35        0.50
Hamburg              34      40        46        41      1.21        1.18        1.35
Augsburg             34      42        52        38      1.12        1.24        1.53
Hertha               34      42        42        50      1.47        1.24        1.24
Darmstadt            34      38        53        38      1.12        1.12        1.56
Hannover             34      31        62        25      0.74        0.91        1.82
Dortmund             34      82        34        78      2.29        2.41        1.00
M'gladbach           34      67        50        55      1.62        1.97        1.47
Leverkusen           34      56        40        60      1.76        1.65        1.18
Hoffenheim           34      39        54        37      1.09        1.15        1.59
Mainz                34      46        42        50      1.47        1.35        1.24
Ingolstadt           34      33        42        40      1.18        0.97        1.24
Werder Bremen        34      50        65        38      1.12        1.47        1.91
Schalke 04           34      51        49        52      1.53        1.50        1.44
Stuttgart            34      50        75        33      0.97        1.47        2.21
FC Koln              34      38        42        43      1.26        1.12        1.24
Wolfsburg            34      47        49        45      1.32        1.38        1.44
Ein Frankfurt        34      34        52        36      1.06        1.00        1.53
```

Figure 27 Output League Table in IDE Console

5.4.4 Error Handling

As identified in the design chapter of the dissertation, giving informative feedback to the user was a priority. However the system was designed to minimise the opportunity for user input, reducing the possibility of errors. One potential error that became apparent during development was trying to access predictions offline, as after the prediction indicator values are sent from the app, all work is done server side requiring an internet connection. It was therefore decided that when a user attempts to view prediction results offline that a message would be displayed to them that they are required to connect to the internet. This was implemented easily within the app using the ConnectivityManager class. As illustrated in Figure 29 if there is no internet connection detected then the user is informed that they need to connect to the internet to access predictions.

```
//display a message if no internet connection
ConnectivityManager cm = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
if (cm.getActiveNetworkInfo() == null)
    Toast.makeText(getApplicationContext(), "Please connect to Internet to see predictions", Toast.LENGTH_LONG).show();
```

Figure 28 Use of the ConnectivityManager Class

As part of an agile approach to development this feature was tested as it was implemented.

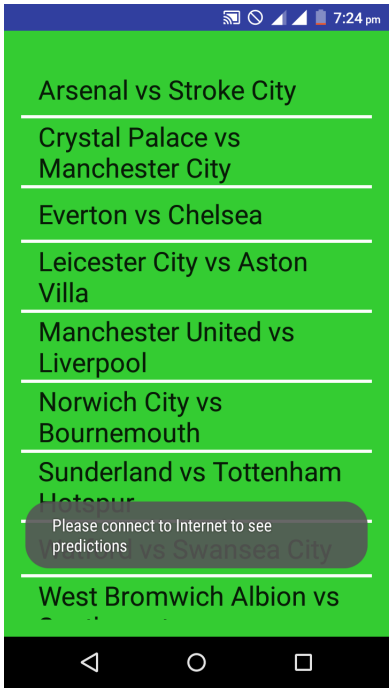


Figure 29 Testing the use of the ConnectivityManager

5.4.5 Algorithm Efficiency

Alternative implementations of the K Nearest Neighbour algorithms were considered to speed the user waiting time for predictions. For example, it was discussed with stakeholders that instead of calculating distances for data points that only the database records within a certain range of our query point were looked at. This would speed up the algorithm dramatically however it presented the problem that there may not be enough records within a given range. Also the traditional K Nearest Neighbour algorithm was preferred due to the control it offered, that it offered results to 2 decimal places and that the initial testing of prediction accuracy provided encouraging results (70% accuracy over game week 5 of a Premier League game week).

Given, that the current implementation was the preferred algorithm it was decided that changes to it were required. As discussed, the calculation of distances of database records was necessary so attention was then turned to the sorting algorithm of results. Research was undertaken of sorting algorithms that may improve performance which led to heap sort being trialled, instead of bubble sort. This reduced the loading time of predictions to under two seconds, which was deemed an acceptable waiting time for users and met the initial functional requirement outlined. Heap sort was adopted after research was undertaken that

resulted in it and quick sort being considered. However heap sort was selected for its superior worst case scenario performance, as well as it's consistency.

5.4.6 Confidence Feature

Upon presenting a working prototype of the app to project stakeholders, it was requested that a feature to show the confidence of predictions generated was added to the system. As an Agile approach to development was adopted this was considered and judged to be feasible. To do so the algorithm displayed the distance of the hundredth (K) nearest neighbour in the data set. The rationale behind implementing in this manner was that if the nearest neighbours were a further distance from a query point they are less likely to be accurate than a hundred nearest neighbours that were closer to a query point. A message is also displayed to the user providing guidance that the highest confidence predictions are those closest to zero. As discussed in the following chapter, the accuracy of the feature has provided encouraging results.

5.47 User Interface Alteration

Whilst the implementation stage was coming to an end, two issues remained that provided cause for concern. The main issue was that the app was not scaling appropriately to devices of different sizes (Shown in Figure 30) This issue was caused largely by hard coding values in the layout design, such as button sizes, images sizes and list view item sizes to make the interface appear similar to the template outlined in chapter four.

Figure 30 Images not scaling to larger devices

Also whilst it was felt that the design of the app was acceptable, much could be improved as it looked slightly outdated. Buttons used were custom designed, ignoring native buttons. Also on further consideration the initial main screen was considered an extra button press for users accessing the information they required.

As a result there was a complete overhaul of the system design. Firstly, the initial greeting screen and the menu screen were combined and the buttons were replaced with Android native buttons, to adhere to Android design guidelines..

Next, the list view was replaced with cards as they are easily scalable to different sized devices and gave a consistent and more professional look throughout. They give a native feel to the app and also are easily customised, for example if it was wished to display team badges as well.

Also, the background colour of screens other than the home screen were reverted to white, giving a less overwhelming appearance to the app.

To ensure all information was alligned correctly, improving the design and scaling of the results activity research led to conclude that table layout best suited the needs of the project and alligned information correctly. This was deemed to be more visually pleasing for users.

Appendices E and F show the difference in design of the app before and after these alterations.

5.5 Summary

This chapter detailled the main implementation steps taken to produce the final version of the app. In addition to this, any issues encountered or changes made are also included, as well as the reasoning behind any decisions made.

Testing

6.1 Introduction

Software testing is a phase of the Software Development Lifecycle which aims to detect bugs that impact software's operation and performance. It is the process of evaluating a system to ensure it meets the the business needs / requirements that influenced it's design and development (ISTQB, 2016).

This chapter describes the quality assurance process undertaken and documents the results obtained, as well as any amendments made as a result.

6.2 Algorithm Testing

This phase of testing was carried out at an early stage in the project when the feature was added. After adding the K Nearest Neighbour algorithm to the system it was tested to ensure it worked. However, to ensure high quality software was deployed it was also decided to check that the algorithm was functioning correctly. To do so dummy data was inserted into the database. This entailed entering one hundred home win records into the database and checking that the system predicted a 100% chance of there being a home win. This algorithm functioned as expected and the feature was then demonstrated to project stakeholders.

6.3 Database Accuracy Verification

This phase of testing was undertaken at an early stage, due to adopting an agile approach to development, however it was crucial to ensure that the data which the K Nearest Neighbor algorithm was being performed on was accurate. In order to verify the program worked correctly the values produced in the console were compared to end of season league tables to ensure they matched. The following output is based upon the 2015/ 2016 Bundesliga season.

```
Run  FootBallLeagueData (3)
/Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin/java ...
Team              Played  Scored  Conceded   Points  Av Points   Aver scored Aver goals conceded
Bayern Munich       34      80        17       88      2.59        2.35        0.50
Hamburg             34      40        46       41      1.21        1.18        1.35
Augsburg            34      42        52       38      1.12        1.24        1.53
Hertha              34      42        42       50      1.47        1.24        1.24
Darmstadt           34      38        53       38      1.12        1.12        1.56
Hannover            34      31        62       25      0.74        0.91        1.82
Dortmund            34      82        34       78      2.29        2.41        1.00
M'gladbach          34      67        50       55      1.62        1.97        1.47
Leverkusen          34      56        40       60      1.76        1.65        1.18
Hoffenheim          34      39        54       37      1.09        1.15        1.59
Mainz               34      46        42       50      1.47        1.35        1.24
Ingolstadt          34      33        42       40      1.18        0.97        1.24
Werder Bremen       34      50        65       38      1.12        1.47        1.91
Schalke 04          34      51        49       52      1.53        1.50        1.44
Stuttgart           34      50        75       33      0.97        1.47        2.21
FC Koln             34      38        42       43      1.26        1.12        1.24
Wolfsburg           34      47        49       45      1.32        1.38        1.44
Ein Frankfurt       34      34        52       36      1.06        1.00        1.53
```

Figure 31 Console output for Bundesliga 2015/2016 season

This data matches real life data as demonstrated in Figure 32.

2015–16 Bundesliga
Standings

#	Team	GP	W	D	L	GF	GA	GD	PTS
1	Bayern Munich	34	28	4	2	80	17	63	**88**
2	Dortmund	34	24	6	4	82	34	48	**78**
3	Bayer	34	18	6	10	56	40	16	**60**
4	Mönchengladbach	34	17	4	13	67	50	17	**55**
5	Schalke 04	34	15	7	12	51	49	2	**52**
6	Mainz 05	34	14	8	12	46	42	4	**50**
7	Hertha BSC	34	14	8	12	42	42	0	**50**
8	Wolfsburg	34	12	9	13	47	49	-2	**45**
9	1. FC Köln	34	10	13	11	38	42	-4	**43**
10	Hamburger SV	34	11	8	15	40	46	-6	**41**
11	Ingolstadt	34	10	10	14	33	42	-9	**40**
12	FC Augsburg	34	9	11	14	42	52	-10	**38**
13	Werder	34	10	8	16	50	65	-15	**38**
14	Darmstadt 98	34	9	11	14	38	53	-15	**38**
15	Hoffenheim	34	9	10	15	39	54	-15	**37**
16	Eintracht	34	9	9	16	34	52	-18	**36**
17	VfB Stuttgart	34	9	6	19	50	75	-25	**33**
18	Hannover 96	34	7	4	23	31	62	-31	**25**

Figure 32 Bundesliga 2015/16 table

However, it was also important to ensure the output csv file containing data to be imported into the databse was correct. In order to do so, some manual tracing through the season was calculated by hand to verify the figures within the output csv file. Also, by checking the last week of fixtures within a season results were verified.

6.4 In House Testing

This phase of software testing was carried out by the developer, after implementation as a checklist of all features that the app incorporated from testing buttons to high level features. The purpose was to ensure all components functioned as expected before releasing to users for acceptance testing. The checks carried out are documented:

Feature	Functions correctly
About button directs user to about screen	✓
EPL button directs user to EPL predictions screen	✓
Bundesliga button directs user to Bundesliga predictions	✓
All EPL fixtures generate the corresponding predictions upon a button press	✓
All Bundesliga fixtures generate the corresponding predictions upon a button press	✓
Predictions are large enough to be read by a user	✓
Retains data upon shaking	✓
Generates output message to connect to internet when offline	✓
App scales to larger devices such as tablets	✓
App scales to smaller phone devices	✓
Back button on each activity functions correctly	✓
Upon updating fixtures and their parameters to be sent to the app that new predictions are displayed	✓
App logo displays appropriately on different screen sizes	✓

Table 7 in house tests carried out

6.5 Acceptance Testing

After the system had corrected all known defects it was decided to issue the solution to users for acceptance testing. The goal of acceptance testing is to establish

confidence in the system and gain feedback from end users (ISTQB, 2016). This included a variety of responses from preferences to further identification of bugs. The Prediction App was acceptance tested by project stakeholders in September 2016. Table 8 documents any issues raised during acceptance testing and the developer's response to them.

User Feedback	Developer's response
On the results activity "Probability" was misspelt	Spelling mistake corrected
There was no consistent structure to the ordering of predictions	Predictions ordered by date, then alphabetically by home team
The results activity contained too much white space	Unchanged. The results page will contain only a small amount of information, as loading all prediction probabilities at once severely impacted performance. Also best practice guidelines advise minimal text on screens
Confidence value was vague	Output to the user that values closest to zero were most confident was added to the results screen
The Bundesliga and Premier League matches displayed were played on different game weeks	Unchanged. The purpose of this version of the app was only to demonstrate game week 5 predictions for each league.
When trying to access the results activity offline nothing happened	A toast message was already included in the app to inform users they must connect to the internet to view predictions. When making final changes to the app this code was erroneously deleted and subsequently included again
Dates of games played should be included	Dates of games added.
Results activity would not load in landscape mode. However successfully loaded when switched back to portrait	Unable to replicate fault.

Table 8 User Feedback

Following this process, a decision was taken to issue the app to a larger volume of users to enhance the testing phase. A focus group of four football enthusiasts were given access to "The Prediction App" and issued a questionnaire that contained questions relating to the app and the ease of which they were able to complete pre defined scenarios. The aim was to further verify whether requirements had been

achieved. Users were then asked to rate the following statements on a scale of one to five, in accordance with the following criteria:

1. Disagree strongly
2. Disagree
3. Neither agree nor disagree
4. Agree
5. Strongly agree

User feedback can be viewed in table 9, documenting the average user rating. As demonstrated, the feedback was very positive from four typical end users.

Task / Statement	Average user rating
I found it easy to navigate to the about section	5
I was able to quickly access the content I wished to	5
I feel suitably informed as to how predictions are made	4.5
I feel the user interface is modern and attractive	4.5
I was able to find Premier League predictions easily	5
I was able to find Bundesliga predictions easily	5
I was able to find predictions for Arsenal versus Stoke easily	5
Output was meaningful to me on the prediction page	4.5

Table 9 Focus group testing results

6.6 Testing prediction accuracy against real life outcomes

In order to test the effectiveness of the app it was necessary to test predictions against real life outcomes. The following tables demonstrate English Premier League and Bundesliga Predictions, Confidence values (Most confident values are those closest to zero) and actual outcomes for game weeks 5,6 and 7. It should be noted that to accurately assess the effectiveness of the solution predictions should be tested over the entire course of a season, however due to time restrictions this was not possible.

Fixture	Prediction / Confidence	Actual Outcome
Arsenal V Stoke	Home win / 0.05	Home win
Crystal Palce V Man City	Away win/ 0.73	Away win
Everton V Chelsea	Home win / 0.24	Home win
Leicester V Aston Villa	Home win / 0.28	Home win
Man Utd V Liverpool	Home win / 0.32	Home win
Norwich V Bournemouth	Home win / 0.21	Home win
Sunderland V Tottenham	Home win / 1.55	Away win
Watford V Swansea	Away win / 1.06	Home win
WBA V Southampton	Away win / 0.25	Draw
West Ham V Newcastle	Home win / 1.32	Home win

Table 10 Game week 5 accuracy testing for Premier League

Fixture	Prediction / Confidence	Actual Outcome
Aston Villa V West Brom	Home win / 0.12	Away win
Bournemouth V Sunderland	Home win/ 0.04	Home win
Chelsea V Arsenal	Away win / 0.53	Home win
Man City V West Ham	Home win / 0.5	Away win
Tottenham V Crystal Palace	Away win / 0.56	Home win
Newcastle V Watford	Home win / 1.11	Away win
Stoke V Leciester	Away win / 1.34	Draw
Swansea V Everton	Home win / 0.04	Draw
Liverpool V Norwich	Draw / 0.94	Draw
Southampton V Man Utd	Home win / 0.55	Away win

Table 11 Game week 6 accuracy testing for Premier League

Fixture	Prediction / Confidence	Actual Outcome
Leicester V Arsenal	Home win / 0.19	Away win
Liverpool V Aston Villa	Home win/ 0.27	Home win
Man Utd V Sunderland	Home win / 0.11	Home win
Newcastle V Chelsea	Away win / 0.12	Draw
Southampton V Swansea	Home win / 0.48	Home win
Stoke V Bournemouth	Away win / 0.48	Home win
Tottenham V Man City	Away win / 0.33	Home win
Watford V Crystal Palace	Home win / 0.15	Away win
West Brom V Everton	Away win / 0.16	Away win

West Ham V Norwich	Home win / 0.36	Draw

Table 12 Game week 7 accuracy testing for Premier League

Fixture	Prediction/Confidence	Actual Outcome
Mainz V Hoffenheim	Home win / 0.87	Home win
Darmstadt V Bayern Munich	Away win / 1.97	Away win
FC Koln V M'gladbach	Home win / 1.27	Home win
Hamburg V Frankfurt	Away win / 0.42	Draw
Werder Bremen V Ingolstadt	Home win / 0.4	Away win
Wolfsburg V Hertha Berlin	Home win / 0.06	Home win
Augsburg V Hannover	Home win / 0.99	Home win
Dortmund V Leverkusen	Home win / 2.67	Home win
Stuttgart V Schalke	Away win / 1.05	Away win

Table 13 Game week 5 accuracy testing for Bundesliga

Fixture	Prediction	Actual Outcome
Bayern Munich V Wolfsburg	Home win / 1.2	Home win
Darmstadt V Werder Bremen	Home win / 0.13	Home win
Hertha Berlin V FC Koln	Home win / 0.9	Home win
Ingolstadt V Hamburg	Home win / 1.33	Away win
Hannover V Stuttgart	Home win / 0.28	Away win
Hoffenheim V Dortmund	Away win / 2.7	Draw
Leverkusen V Mainz	Away win / 0.11	Home win
M'gladbach V Augsburg	Away win / 0.23	Home win
Schalke V Frankfurt	Home win / 0.73	Home win

Table 14 Game week 6 accuracy testing for Bundesliga

Fixture	Prediction	Actual Outcome
FC Koln V Ingolstadt	Home win / 0.57	Draw
Augsburg V Hoffenheim	Home win / 0.18	Away win
Hamburg V Schalke	Home win / 0.58	Away win
Mainz V Bayern Munich	Away win / 1.83	Away win
Stuttgart V M'gladbach	Home win / 1.4	Away win
Werder Bremen V	Home win / 0.61	Away win

Leverkusen		
Wolfsburg V Hannover	Home win / 0.17	Draw
Borussia Dortmund V Darmstadt	Home win / 1.73	Draw
Frankfurt V Hertha	Home win / 0.17	Draw

Table 15 Game week 7 accuracy testing for Bundesliga

The following findings were made from analysis of accuracy testing of results:

- The first game week of testing the accuracy of English Premier League predictions delivered 70% accuracy
- The first game week of testing the accuracy of Bundesliga predictions delivered 78% accuracy
- The second game week of testing the accuracy of English Premier League predictions delivered 20% accuracy
- The second game week of testing the accuracy of Bundesliga predictions delivered 44% accuracy
- The third game week of testing the accuracy of English Premier League predictions delivered 40% accuracy
- The third game week of testing the accuracy of Bundesliga predictions delivered 11% accuracy
- The overall accuracy of the testing period was 42%, with the first week of predictions in each league offering high accuracy rates then dipping in the following weeks predictions
- Of the five predictions made with the strongest confidence value (value closest to zero) over a three week period, 4 out of 5 predictions were correct, offering an optimistic outlook on the confidence feature
- Of the 26 games in the testing sample that ended in a home win, 19 (73%) were correctly predicted. Of the games tested that resulted in an away win, 5 out of 19 were predicted correctly (26%). Of the 12 games that ended in a draw only 1 was predicted correctly. These figures show that the system is most accurate when forecasting home wins, followed by away wins and fairly unreliable at predicting draws.

6.7 Summary

This chapter describes the internal and external testing processes undertaken to verify and improve the quality of software produce. Also, it documents the success of the app in forecasting football results so far.

7.1 Introduction

Following the successful release of the Prediction app, this chapter provides an overview of the work undertaken from the initial problem solving stage of the project through to its implementation and testing.

7.2 Project Summary

Initial background research was conducted investigating the difficulty of predicting the outcome of football games. A broad range of factors were identified via literature review, investigation and a previous interest in the sport. Initial research identified the impact of the identified problem on society, with consumers losing sums of money to bookmakers regularly attempting to predict the outcome of matches.

A range of existing solutions to the problem were examined, across a variety of platforms which helped provide recomendations for the system at an early stage. Alternative prediction solutions that had not been availed of were also investigated such as Machine Learning. A project risk analysis was then documented, followed by a business case analysis which served as justification for pursuing the project, outlining the perceived benefits versus the costs. Once this criteria was satisfied the proposed solution to the problem analysed in Chapter 2 was presented. This included the selection of the Machine Learning algorithm that was used, as well as justification for it's selection over other techniques.

As a result of research carried out in the analysis phase and frequent end user consultation a set of requirements was produced. This was grouped into functional and non functional requirements. Given the project was developed adhering to Agile principles, requirements were constantly reassessed and added to as the project progressed.

The app design was influenced by the WRC Mobile Acessibility Initiative, BBC Mobile Accessibility Guidelines and AQuA Best Practice Guidelines. Combined, they placed a great emphasis on the usability of the app, focusing on elements which may not of been considered otherwise. Shneiderman's Golden Rules were also adhered to when designing the user interface, whilst considering the end user at every stage. This led to the design of app screen templates, which were validated by the user and aided the implementation of the user interface. The navigation of the app was also considered at this stage and validated by end users to ensure the content users wished to find was easily accessible. Before proceeding with implementation the architectural design was carefully considered. This included database design and the client server relationship that was adopted and identifying all programming languages to be used, which helped identify any gaps in knowledge that needed to be addressed.

The project was implemented adhering to Agile Development principles, continuously assessing the priorities of sprints and consulting end users. This led to the project containing the main functionality at a very early stage. After each sprint the features added were tested and presented to project stakeholders to ensure the project was progressing as expected. New features such as the confidence of predictions were added at the request of stakeholders.

Testing was conducted by the developer to ensure all elements of the overall final version functioned as anticipated and performance was optimal. The project was then distributed to end users for acceptance testing and the feedback, along with relevant amendments were documented. Also the accuracy of predictions generated were tested against actual outcomes.

7.3 Results Assessment

Through collaboration a list of functional and non functional requirements were outlined in Chapter 4. Upon completion of development and testing both in house and externally it is fair to say that most of the requirements of the project were met.

The in house testing indicated that a robust solution was developed that demonstrated a high level of performance and was aesthetically pleasing on a variety of devices.

External acceptance testing yielded encouraging feedback that indicated an easy to use system was implemented, that met initial requirements outlined by project stakeholders, as well as features requested throughout development. Also any issues that were brought to light during testing were ammended.

Whilst the first game week tested against acutal outcomes was highly accurate, subsequent testing yielded predictions of varying success. As previously mentioned, a full season of testing would be required to give a more effective overview of the system's accuracy rate however the test period undertaken signals significant room for improvement with the accuracy of predictions generated. It was noted at the beginnning of the project this task would be a great challenge, as no one has yet implemented a system to successfully predict the outcome of Football matches. However with the flexible algorithm in place and the ability to extend the app to take into account different/more prediction indicators it is possible to improve on existing results. However, dissapontingly the overall accuracy rating of the test period did not match the 60% target set out in the functional requirements.

7.4 Objectives and Achievements

The key objectives outlined at the beginning of the project have been met throughout the Software Development Lifecycle of this project:

- Research was conducted analysing the problem at hand, analysing a need for a system providing Football predictions based upon up to date data.
- End users were consulted to help outline a list of functional and non functional requirements from the system. This was also significantly aided by the competitive analysis of similar solutions available
- A simple and consistent GUI was implemented that allowed users swift access to information they desired
- The K Nearest Neighbour algorithm was successfully incorporated within the system
- The predictions of the app for two leagues were tested against real life outcomes
- The testing process included feedback from end users who indicated the app successfully met their expectations

7.5 Recommendations

In the requirements analysis it was identified that this system should be expandable. A list of potential recomendations for development and future expansion are as follows:

- Adding predictions for more leagues, such as the French League, Spanish, English Championship and American MLS League
- Adding further predictions, other than the outcome of games, such as correct score, number of cards in a game and number of corners per game
- Predictions could be generated for other sports, such as Rugby, Basketball and various other sports
- When using the app, the user could select the value of k in the K Nearest Neighbour algorithm
- A feature could be implemented to generate the top 5 most likely predictions each week on a Top Tips screen displayed to the user
- A feature to share predictions with social media accounts or via email

7.6 Summary

The project aimed at developing a solution that could accurately predict the outcome of Football games using previous results. Research was undertaken that led to Machine Learning being utilised to achieve this task. A consistent, appealing and suitable design was implemented based on research of similar systems, collaboration with end users and best practice guidelines. The system was then tested in house and also by end users on a range of Android devices, receiving encouraging feedback. The provision of such a tool assists users to make predictions regarding the outcome of Football matches, whilst responsibly indicating the system's confidence in the values it is generating. It also informs users how predictions are made and provides a disclaimer notice.

The project enabled valuable experience to be gained managing and delivering complex projects. It also provided experience of using agile as a development approach and adopting a flexible approach to meet the evolving needs of clients, as well as increasing knowledge of Machine Learning that is being increasingly utilised

in various technologies. Also crucial lessions were learnt from development, such as the importance of testing and the need to consider performance and code efficiency. Overall the project achieved mixed success. Despite initial prediction testing indicating underwhelming success there are still many positives to be drawn from the project. As mentioned, there is scope for further adjustments to predictions, which will enable the system to deliver higher accuracy rates but it was envisaged that a higher rate of accuracy would have been achieved by the first release of “The Prediction App”.

References

Android Authority. (2016). *I want to develop Android Apps - What languages should I learn?*. [online] Available at: http://www.androidauthority.com/want-develop-android-apps-languages-learn-391008/ [Accessed 17 Jul. 2016].

Android Central. (2015). *Google says there are now 1.4 billion active Android devices worldwide*. [online] Available at: http://www.androidcentral.com/google-says-there-are-now-14-billion-active-android-devices-worldwide [Accessed 1 Jul. 2016].

AQuA, (2013). *Best Practice Guidelines*. [online] App Quality Alliance, pp.1 - 30. Available at: http://www.appqualityalliance.org/AQuA-best-practice-guidelines [Accessed 2 Jul. 2016].

Android Developers. (2016). *Layouts | Android Developers*. [online] Available at: https://developer.android.com/guide/topics/ui/declaring-layout.html [Accessed 17 Jul. 2016].

BBC. (2009). *BBC NEWS | Programmes | More Or Less | The professor's Premiership probabilities*. [online] Available at: http://news.bbc.co.uk/1/hi/programmes/more_or_less/8062277.stm [Accessed 30 Jun. 2016].

BBC. (2013). *BBC Mobile Accessibility Prototype : About*. [online] Available at: http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile/about [Accessed 12 Jul. 2016].

Computer Hope. (2016). *What is server?*. [online] Available at: http://www.computerhope.com/jargon/s/server.htm [Accessed 14 Jul. 2016].

Constantinou, A., Fenton, N. and Neil, M. (2012). pi-football: A Bayesian network model for forecasting Association Football match outcomes. *Knowledge-Based Systems*, 36, pp.322-339.

Constantinou, A. and Fenton, N. (2012). Solving the Problem of Inadequate Scoring Rules for Assessing Probabilistic Football Forecast Models. *Journal of Quantitative Analysis in Sports*, 8(1).

Dobson, S. and Goddard, J. (2001). *The economics of football*. New York: Cambridge University Press.

Dummies. (2016). *Ten Benefits of Agile Project Management - For Dummies*. [online] Available at: http://www.dummies.com/how-to/content/ten-benefits-of-agile-project-management.html [Accessed 27 Jul. 2016].

Erceg, N. and Galić, Z. (2014). Overconfidence bias and conjunction fallacy in predicting outcomes of football matches. *Journal of Economic Psychology*, 42, pp.52-62.

Financial Times. (2016). *Paddy Power Betfair increases profits despite costly Cheltenham — FT.com*. [online] Available at: http://www.ft.com/cms/s/0/05dcce14-11c4-11e6-bb40-c30e3bfcf63b.html#axzz4CECS5gZR [Accessed 21 Jun. 2016].

Flurry. (2014). Apps Solidify Leadership Six Years into the Mobile Revolution. [ONLINE] Available at: http://flurrymobile.tumblr.com/post/115191864580/apps-solidify-leadership-six-years-into-the-mobile. [Accessed 17 January 16].

Goddard, J. (2005). Regression models for forecasting goals and match results in association football. *International Journal of Forecasting*, 21(2), pp.331-340.

Goddard, J. (2006). Who wins the football?. *Significance*, 3(1), pp.16-19.

Guru99. (2016). [online] Available at: http://www.guru99.com/agile-scrum-extreme-testing.html [Accessed 27 Jul. 2016].

Heuer, A. and Rubner, O. (2009). Fitness, chance, and myths: an objective view on soccer results. *The European Physical Journal B*, 67(3), pp.445-458.

ISTQB. (2016). *What is Software Testing?*. [online] Available at: http://istqbexamcertification.com/what-is-a-software-testing/ [Accessed 17 Sep. 2016].

Joseph, A., Fenton, N. and Neil, M. (2006). Predicting football results using Bayesian nets and other machine learning techniques. *Knowledge-Based Systems*, 19(7), pp.544-553

Palomino, F., Rigotti, L. and Rustichini, A. (1998). *Skill, strategy and passion*. Tilburg: Tilburg University.

Peddibhotla, G. (2015). *50 Useful Machine Learning & Prediction APIs*. [online] Kdnuggets.com. Available at: http://www.kdnuggets.com/2015/12/machine-learning-data-science-apis.html [Accessed 27 Jul. 2016].

Peterson, L. (2009). K-nearest neighbor. *Scholarpedia*, 4(2), p.1883.

RTE (2015). Paddy Power reports record pre-tax profits for last year. [ONLINE] Available at: http://www.rte.ie/news/business/2015/0303/684065-paddy-power-results/. [Accessed 17 January 16].

Shneiderman, B. (1998). *Designing the user interface*. Reading, Mass: Addison Wesley Longman.

Sqa. (2007). *Functional and Non-Functional Requirements*. [online] Available at: http://www.sqa.org.uk/e-learning/SDM03CD/page_02.htm [Accessed 4 Jul. 2016].

Technologist. (2016). *Artificial intelligence predicts the winners of Euro 2016 games*. [online] Available at: http://www.technologist.eu/artificial-intelligence-predicts-the-winners-of-euro-2016-games/ [Accessed 22 Jun. 2016].

Tech Crunch (2014). Mobile Ad Market Spending To Hit $18BN In 2014, Rising To ~$42BN By 2017. [ONLINE] Available at: http://techcrunch.com/2014/01/21/mobile-ad-market-forecast-to-2017/. [Accessed 17 January 16].

The Guardian (2015). Android beats iOS for app downloads, but revenues are still a different story. [ONLINE] Available at: http://www.theguardian.com/technology/2015/jan/28/android-ios-app-downloads-revenues-app-annie-google-play-app-store. [Accessed 17 January 16].

The Guardian. (2014). UK app developers predicted to add £4bn to economy this year. [ONLINE] Available at: http://www.theguardian.com/technology/2014/jun/26/uk-apps-economy-worth-four-billion-pounds. [Accessed 17 January 16].

W3C Initiative. (2011). *WCAG 2 at a Glance*. Retrieved May 4, 2015, from W3C Initiative: http://www.w3.org/WAI/WCAG20/glance/

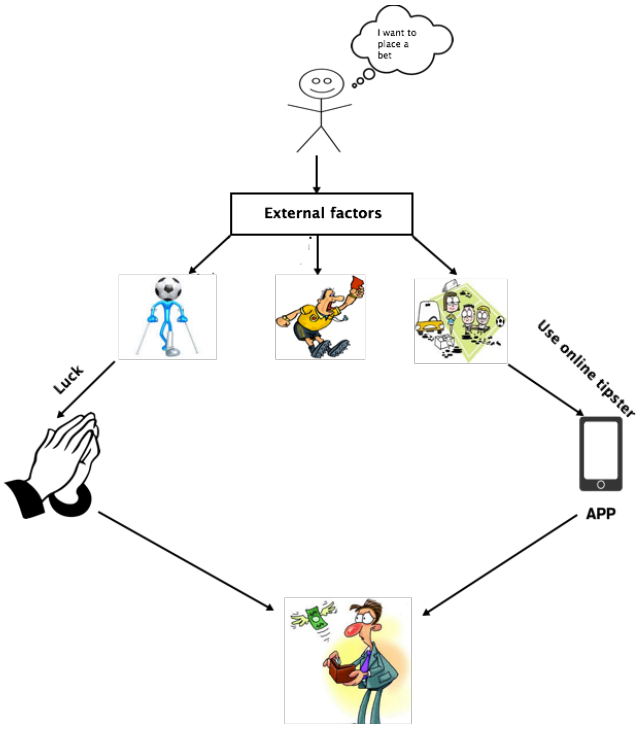
W3C. (2015). *Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile*. [online] Available at: http://www.w3.org/TR/mobile-accessibility-mapping/ [Accessed 10 Jul. 2016].

W3schools. (2014). *PHP Introduction*. [online] Available at: http://www.w3schools.in/php/intro/ [Accessed 17 Jul. 2016].

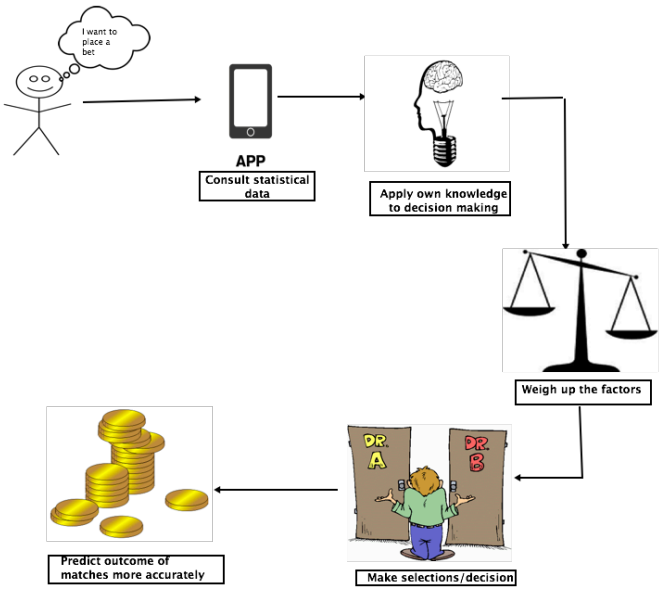
Yezus, A. (2014). *Predicting outcome of soccer matches using machine learning*. [online] Available at: http://www.math.spbu.ru/SD_AIS/documents/2014-12-341/2014-12-tw-15.pdf [Accessed 20 Jan. 2016].

Appendices

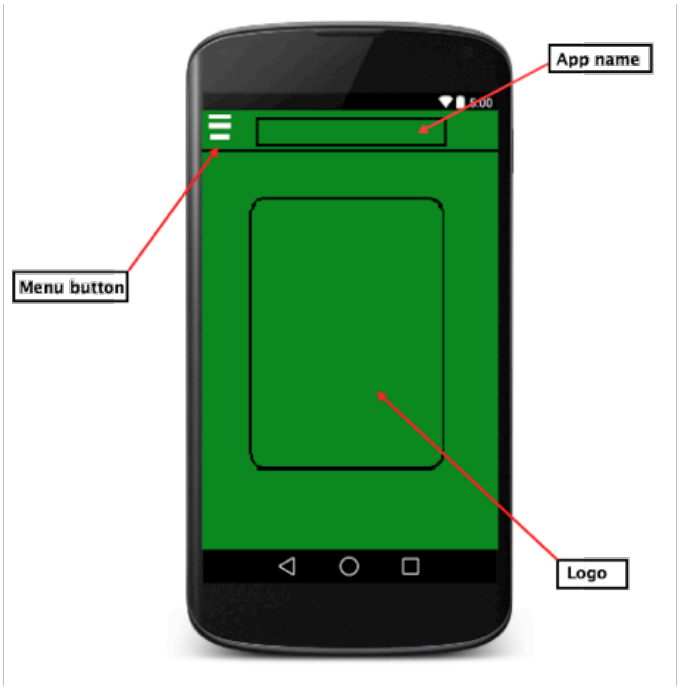
Appendix A – Rich Picture Diagram of Problem

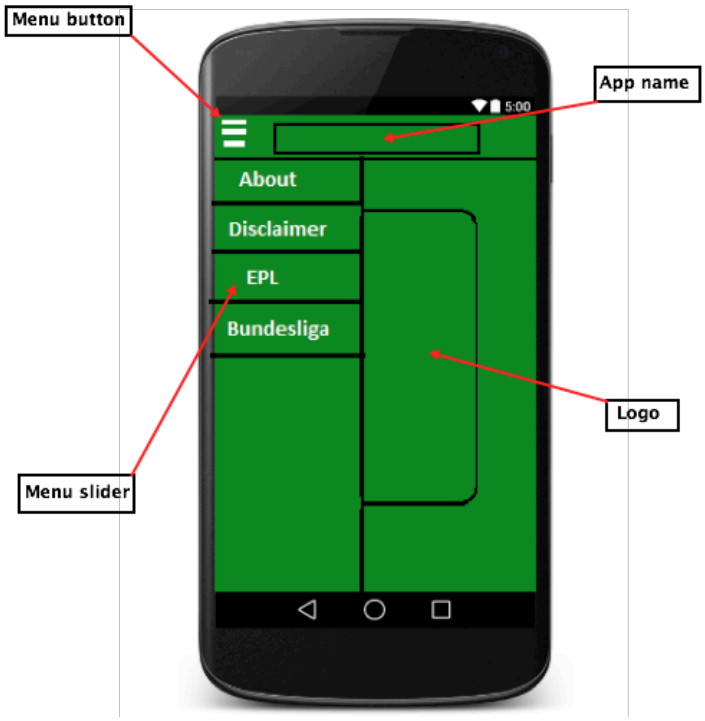


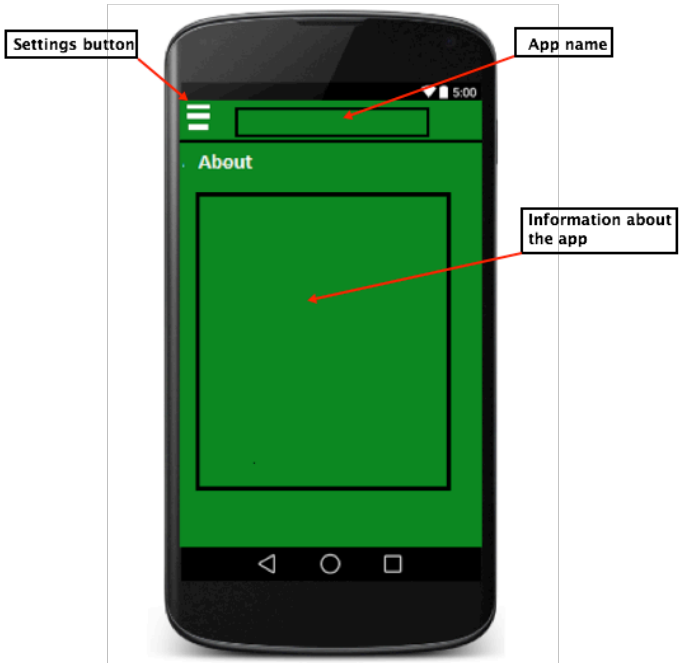
Appendix B – Rich Picture Diagram of Solution

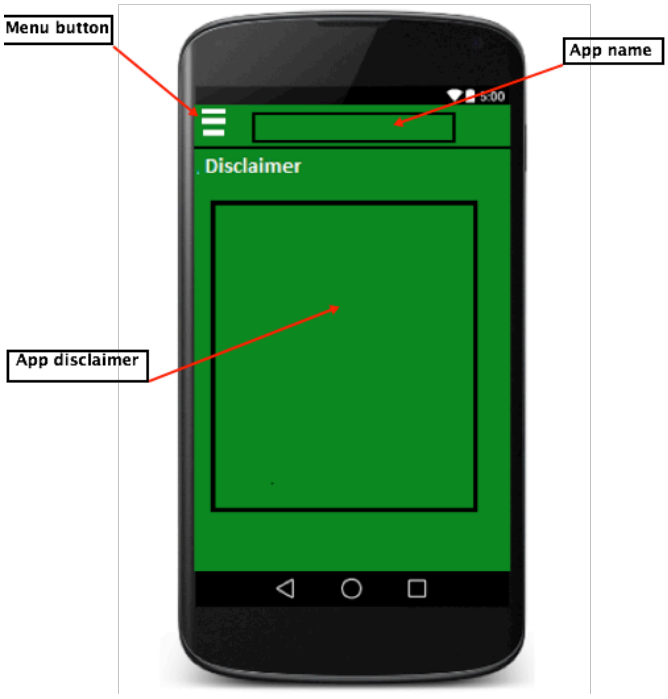


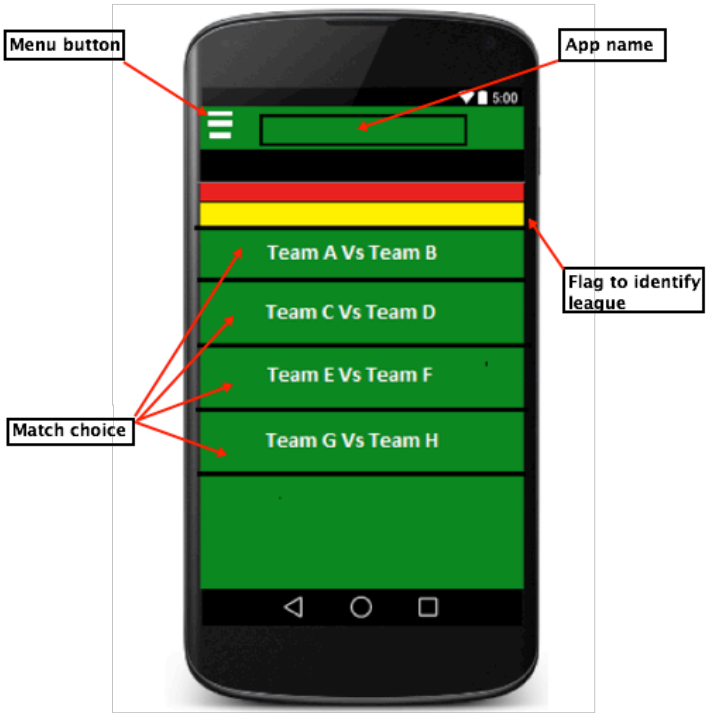
Appendix C Original Home Screen Design

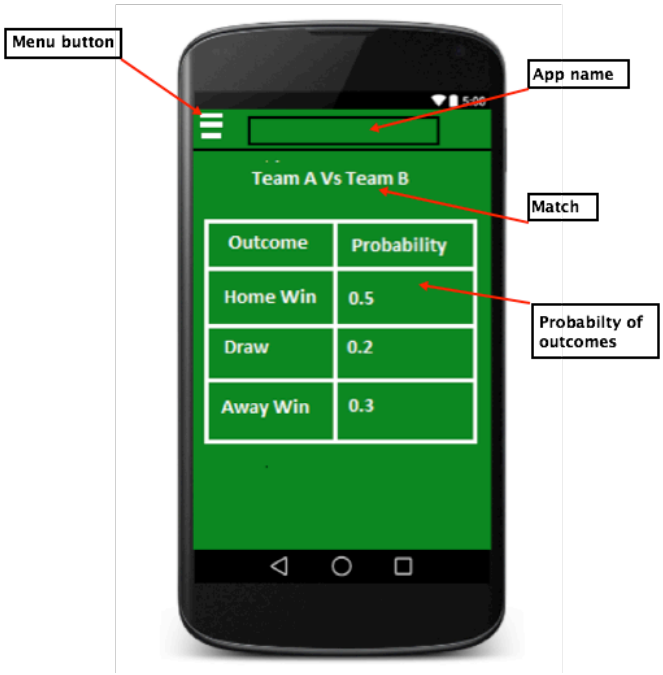




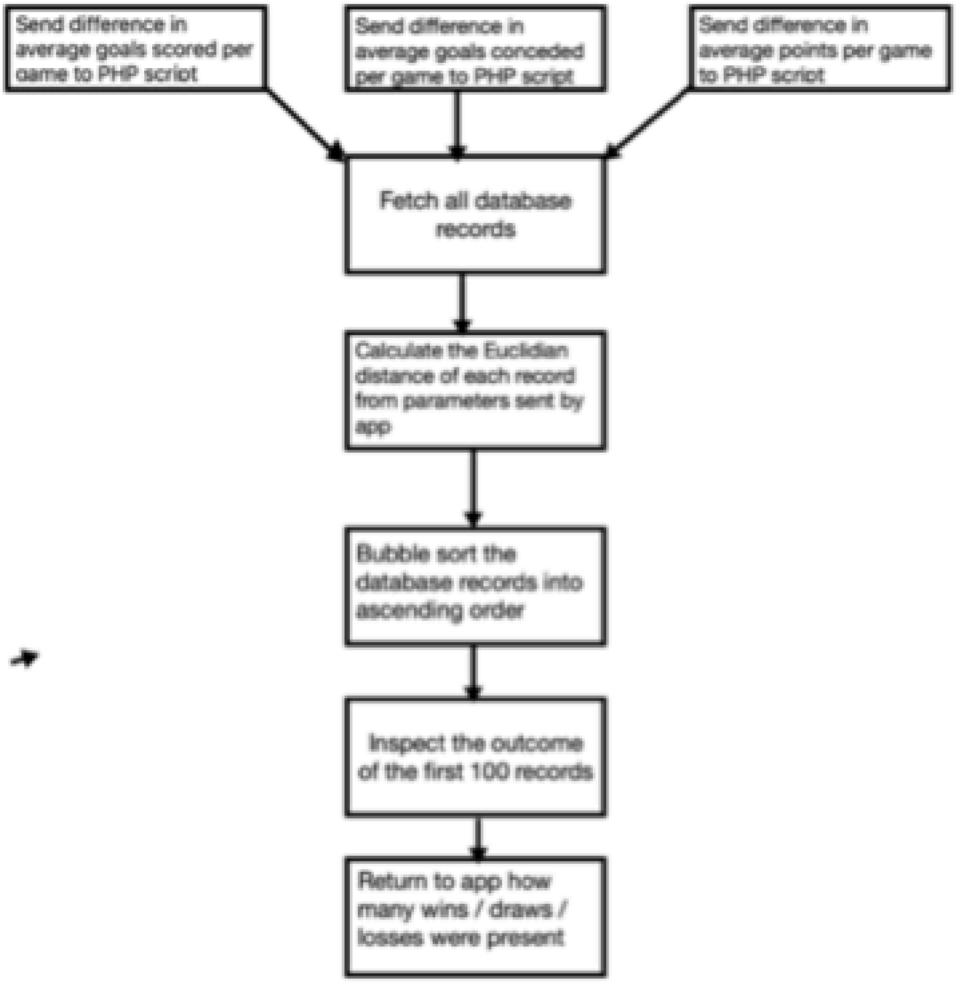

Settings button
App name
5:00
About
Information about the app


Menu button
App name
5:00
Disclaimer
App disclaimer

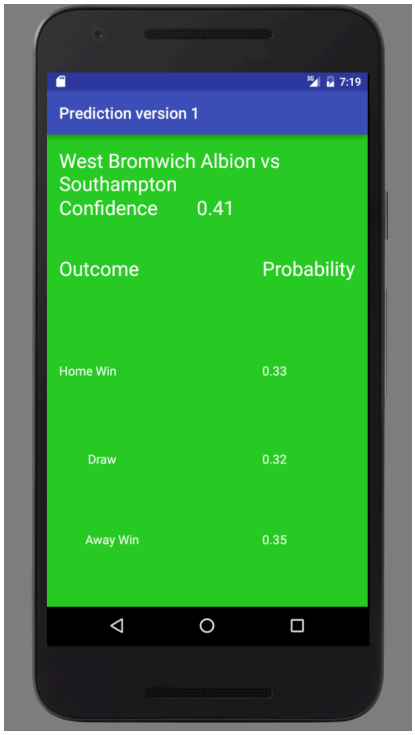

Menu button
App name
5:00
Team A Vs Team B
Team C Vs Team D
Team E Vs Team F
Team G Vs Team H
Flag to identify league
Match choice


Menu button
App name
5:00
Team A Vs Team B
Match
Outcome
Probability
Home Win
0.5
Draw
0.2
Away Win
0.3
Probabilty of outcomes

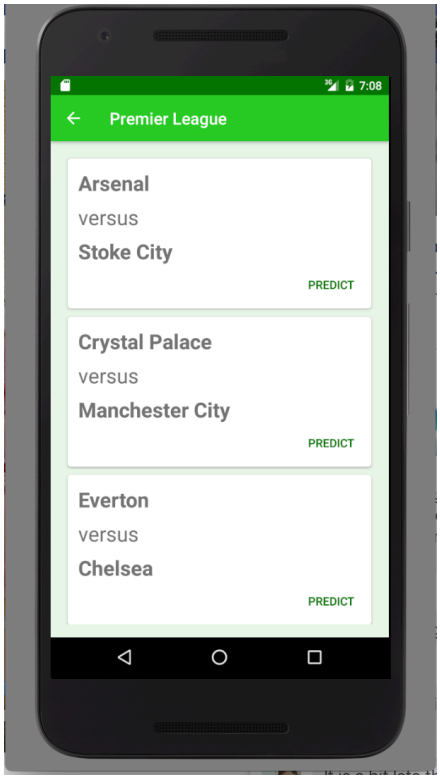
Appendix D Algorithm Implementation

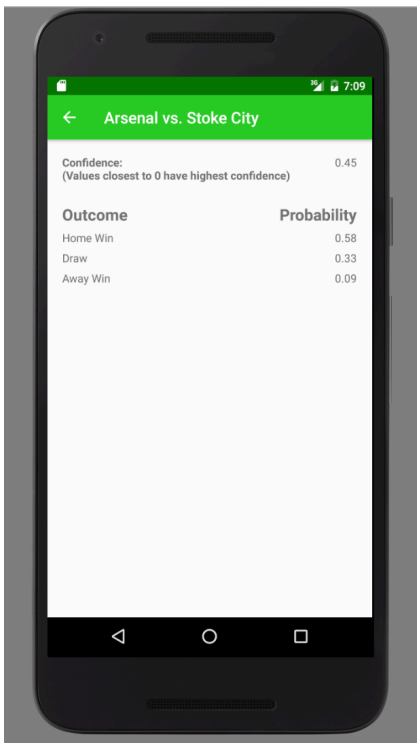


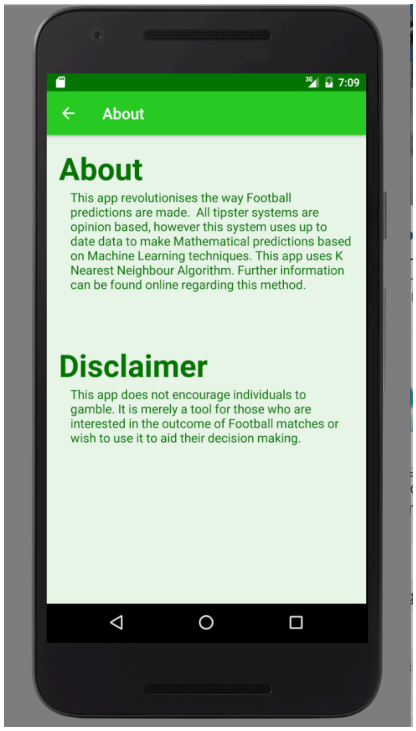
Appendix E Original Implementation GUI



Appendix F Final GUI


7:08
Premier League
Arsenal
versus
Stoke City
PREDICT
Crystal Palace
versus
Manchester City
PREDICT
Everton
versus
Chelsea
PREDICT


7:09
Arsenal vs. Stoke City
Confidence:
(Values closest to 0 have highest confidence)
0.45
Outcome
Probability
Home Win
0.58
Draw
0.33
Away Win
0.09


7:09
About
About
This app revolutionises the way Football predictions are made. All tipster systems are opinion based, however this system uses up to date data to make Mathematical predictions based on Machine Learning techniques. This app uses K Nearest Neighbour Algorithm. Further information can be found online regarding this method.
Disclaimer
This app does not encourage individuals to gamble. It is merely a tool for those who are interested in the outcome of Football matches or wish to use it to aid their decision making.