



COMP4801 Final Year Project [2023/24] Interim Report



KickInsights: Football Application with Machine Learning

Supervisor	Dr. Chim T W	
Student	Fung Hon Yin	3035784497

Date of submission: 2024-01-21

Abstract

In the modern world of technology, there is an ever-rising demand from football fans' to conveniently access comprehensive football data. However, the mobile applications on the market suffer from common downsides including the inconsistency of data, a lack of functions for insights extraction, and the absence of an objective match-winner prediction system. In this project, a brand-new mobile app – KickInsights, will be developed to elevate fans' football experience by tackling the inefficiencies of current alternatives. Upon completion of development, the app is likely to bring comprehensive football data from online databases onto mobile devices and provide tools for statistical comparison, together with a data-driven match prediction system powered by machine learning algorithms.

This report first covers the objectives, scope of data, and deliverables of the project, then explains the frontend development methodologies, in which Figma and React Native have been used for prototyping and application development respectively. For backend development, the selected data benchmarks, sources, cloud storage, and machine learning components to be used will be explained. Also, the challenges faced during the development and their mitigations will be discussed. At present, the project progress is on schedule with the delivery of three frontend screens, supported by a backend database. Development for data collection and wrangling has also been completed. The next steps will be finalizing the frontend development and integrating the machine learning models into the app.

Acknowledgment

I would like to express my sincere and heartfelt gratitude to Dr. TW Chim for supervising this final year project. His continuous support and guidance are crucial for this project.

Moreover, I would like to thank my CAES 9542 lecturer Mr Gagandeep Singh for his help in the project plan and progress reports.

Table of Contents

	Page
1. Introduction	1
1.1 Background	1
1.2 Motivation	1
1.3 Existing Football Applications	4
1.4 Proposed Solution	4
2. Project Objective and Scope	5
2.1 Objective	5
2.2 Scope of Data	5
2.3 Core Functionalities	5
2.3.1 League Table	5
2.3.2 Club Information	6
2.3.3 Club Comparison	8
2.3.4 Player Information	10
2.3.5 Player Comparison	12
2.3.6 Matches Tab	14
2.3.7 Match Centre	15
2.3.8 Machine Learning Prediction (Community Models)	15
2.3.9 Creating Machine Learning Models	17
2.3.10 Machine Learning Prediction (Saved Models)	18
3. Methodology	19
3.1 Frontend Development	19
3.1.1 Prototype	19
3.1.2 UI and UX	20
3.1.3 Application development	21
3.2 Backend Development	21
3.2.1 Data Collection	21

3.2.2 Data Wrangling and Standard	22
3.2.3 Application development	22
3.3 Difficulties and Mitigations	23
4. Project Budget and Schedule	25
4.1 Budget	25
4.2 Schedule	25
5. Conclusion for preliminary work	27
6. References	28
7. Appendices	30

List of Figures

	Page
<i>Figure 1. Club and Player information page from the LiveScore app</i>	1
<i>Figure 2. Club data from the FotMob app</i>	2
<i>Figure 3. Player data from the GOAL app</i>	2
<i>Figure 4. The sequence of actions to locate players' statistics on a web browser</i>	2
<i>Figure 5. Predictions and match result suggestions offered by the LiveScore app</i>	3
<i>Figure 6. Match insights from the FotMob app</i>	3
<i>Figure 7. Prediction polls from the OneFootball and the GOAL app</i>	3
<i>Figure 8. Table Tab</i>	6
<i>Figure 9. Home table</i>	6
<i>Figure 10. Away table</i>	6
<i>Figure 11. Refreshing the league table</i>	6
<i>Figure 12. Clubs Tab</i>	7
<i>Figure 13. Club Info</i>	7
<i>Figure 14. A postponed game</i>	7
<i>Figure 15. Club Stats</i>	7
<i>Figure 16. Shooting aspect in Club Stats</i>	7
<i>Figure 17. Possession aspect in Club Stats</i>	7
<i>Figure 18. Club Players</i>	8
<i>Figure 19. Players information</i>	8
<i>Figure 20. Steps to compare two teams in Club Comparison</i>	9
<i>Figure 21. Comparing a new aspect and a new club in Club Comparison</i>	9
<i>Figure 22. Players Tab</i>	10
<i>Figure 23. General information and statistics of a player</i>	10
<i>Figure 24. General information and statistics of a goalkeeper</i>	11
<i>Figure 25. Locating a player with the search and filter function</i>	11
<i>Figure 26. Locating a player with the club and position filter</i>	12
<i>Figure 27. Comparison interface</i>	13
<i>Figure 28. Adding a player for comparison</i>	13
<i>Figure 29. "Player Added" alert</i>	13
<i>Figure 30. "Compare list is full" alert</i>	13
<i>Figure 31. Comparing an aspect in Player Comparison</i>	14
<i>Figure 32. Matches Tab</i>	15

<i>Figure 33. “Stats” and “Squad” page in Match Centre</i>	<i>15</i>
<i>Figure 34. “Community” page</i>	<i>16</i>
<i>Figure 35. ML Model sorting and filtering</i>	<i>16</i>
<i>Figure 36. Data, Training, and Evaluation Tabs</i>	<i>16</i>
<i>Figure 37. Prediction results</i>	<i>16</i>
<i>Figure 38. Data, Training, and Evaluation Tabs when creating ML models</i>	<i>17</i>
<i>Figure 39. “Saved” page</i>	<i>18</i>
<i>Figure 40. ML model information page</i>	<i>18</i>
<i>Figure 41. Prototype of KickInsights created with Figma</i>	<i>20</i>
<i>Figure 42. Temporary panel for data fetching</i>	<i>23</i>
<i>Figure 43. Loading screen</i>	<i>23</i>

List of Tables

	Page
<i>Table 1. Comparison among popular football apps and KickInsights</i>	4
<i>Table 2. Comparison of clicks required to locate a club's statistic among popular football apps and KickInsights</i>	8
<i>Table 3. Comparison of the average clicks required to compare club statistics among popular football apps and KickInsights</i>	10
<i>Table 4. Comparison of clicks required to locate a player's statistic among popular football apps and KickInsights</i>	12
<i>Table 5. Comparison of the average clicks required to compare player statistics among popular football apps and KickInsights</i>	14
<i>Table 6. User Interfaces of KickInsights on various devices</i>	24
<i>Table 7. Estimated Budget for the project</i>	25
<i>Table 8. Project Schedule Timetable</i>	26

List of Abbreviations

Abbreviation	Definition
API	Application Programming Interface
BaaS	Backend-as-a-Service
GUI	Graphical User Interface
IDE	Integrated Development Environment
iOS	iPhone Operating System
JSON	JavaScript Object Notation
ML	Machine Learning
SVM	Support Vector Machine
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience

1. Introduction

1.1 Background

In the ever-evolving world of football, where there is enormous passion from millions of fans across the globe, there is a growing demand for convenient yet comprehensive access to football data. Supporters crave detailed statistics and insights to keep up with their favourite clubs and players [1]. In particular, 46% of Generation Z's audience regularly use apps while watching sports, highlighting the rise of digital connectivity with sports content [2].

Hence, under today's technological revolution, a variety of mobile apps with popular options including Livescore, OneFootball, and FotMob, have been developed as virtual tools for football fans [3].

1.2 Motivation

Despite the availability of football applications on mobile platforms, most of the existing options share the following downsides.

Firstly, a lack of functions for insights extraction from static data. These apps only provide pages loaded with fixed information with no tools for evaluation. For example, the LiveScore app provides a club or player information page listing their statistics, as shown in Figure 1. However, users cannot take advantage of the data provided to perform further evaluation, including seasonal comparison between clubs, or cross-club player comparison. Without functions for further data exploration, under-utilization of the data has likely resulted.

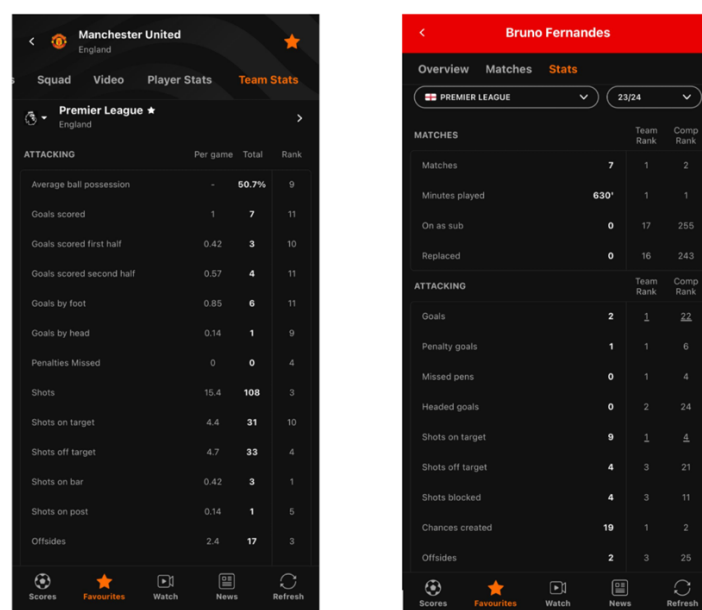


Figure 1. Screenshots from the LiveScore app, showing the Club and Player information page

Secondly, inconsistency and low comprehensiveness of the data provided. For instance, Figure 2 shows that the FotMob app only scratches the surface and offers general club information like points and goal types, while Figure 3 shows that the GOAL app offers only a few columns of player data represented by symbols without clear notation. This leads to a poor overall experience as users are unsure whether the data they are looking for is available in the app.

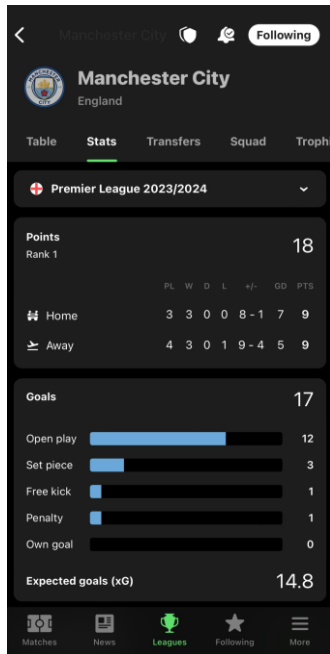


Figure 2. Screenshot from the FotMob app, showing the Club data

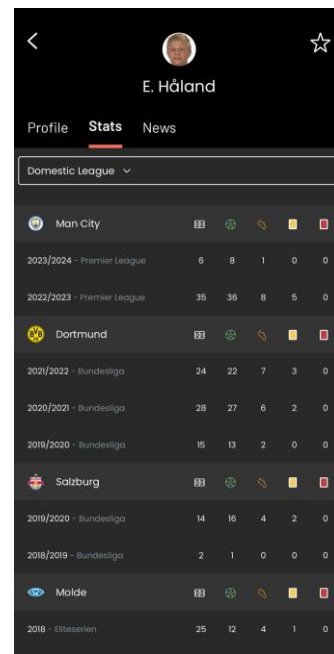


Figure 3. Screenshot from the GOAL app, showing the Player data

For users who desire detailed in-game statistics, like a player's long pass completion rate in the current season, or a goalkeeper's launch rate during goal kicks, it leaves them no choice but to visit large-scale online databases, which is time-consuming and difficult to navigate on a mobile browser, as shown in Figure 4 using iOS's Safari as an example.

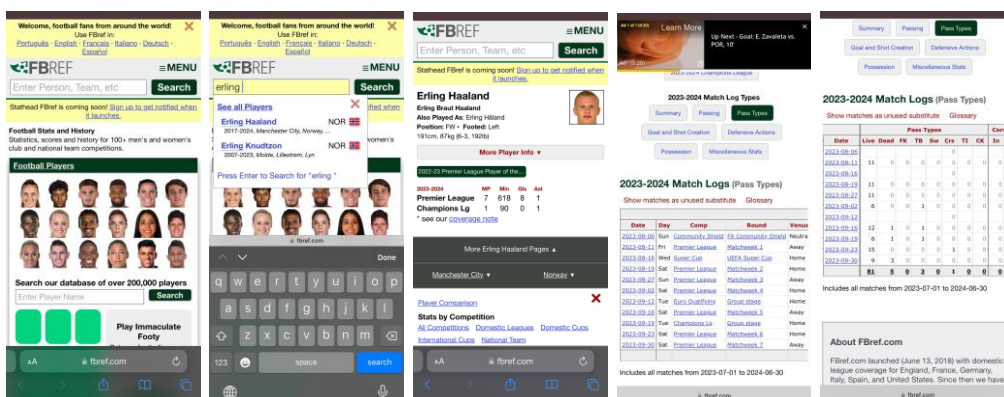


Figure 4. Screenshots from iOS's Safari, showing the sequence of actions to locate players' statistics on a web browser

Thirdly, the absence of an objective match-winner prediction system. Although some apps with prediction features, like LiveScore, offer expert tips and suggestions as shown in Figure 5, they are subjective and prone to errors.

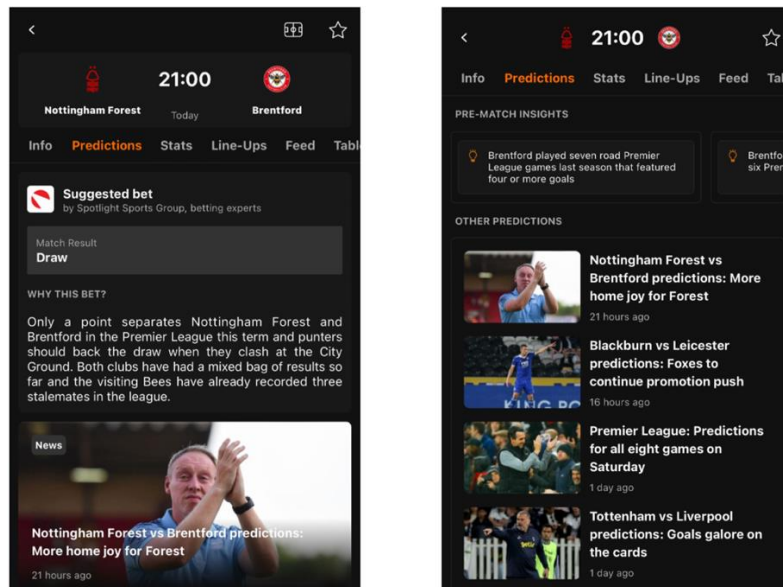


Figure 5. Screenshots from the LiveScore app, showing the predictions and match result suggestions offered by the app

Some apps, like FotMob, only provide unorganized selective insights with a low reference value to users, as shown in Figure 6, while the OneFootball and the GOAL app create polls as an indicator of sentiment, as shown in Figure 7. However, polls may be subject to bias [4]. One possible reason may be due to the different amount of supporters for each club.

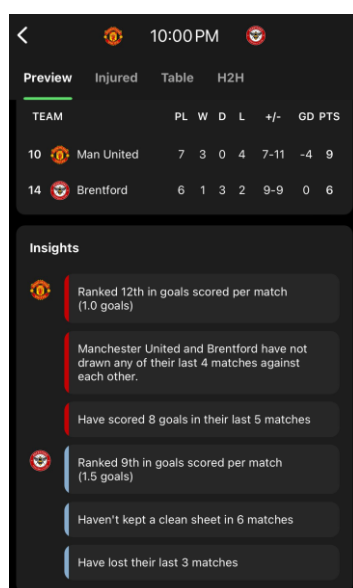


Figure 6. Screenshot from the FotMob app, showing the Match insights

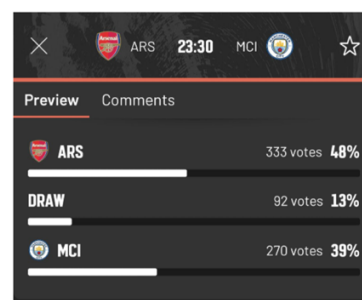
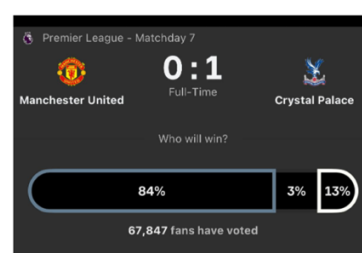


Figure 7. Screenshots from the OneFootball and the GOAL app, showing the prediction polls

1.3 Existing Football Applications

As shown in Table 1, even the best alternatives for obtaining club and player data, LiveScore and OneFootball, only partially provide the data. Moreover, not one of these apps provides an interface for direct club or player comparison, implying that in order to compare these statistics, users would have to navigate between pages to make comparisons in a one-by-one manner. Lastly, match prediction by these apps (if offered) is not statistical-based in general.



Features / Functions	AiScore	FotMob	GOAL	LiveScore	One-Football	KickInsights
In-depth team data [1]	✗	✗	✗	✓	✓	✓
In-depth player data [2][3]	✓	✓	✗	✓	✓	✓
Team/Player Comparison	✗	✗	✗	✗	✗	✓
Match prediction (opinion based)	✓	✗	✓	✓	✓	✓
Match prediction (statistical based)	✗	✓	✗	✓	✗	✓

✓ - Provided , ✓ - Partially Provided , ✗ - Not Provided

[1] See Appendix A

[2][3] See Appendix B, C

Table 1. Comparison among popular football apps and KickInsights (as of 2023-09-13)

1.4 Proposed Solution

The goal of this project is to develop KickInsights, a brand-new mobile app that brings comprehensive football data onto mobile devices. Football enthusiasts will unlikely be required to access in-depth statistics in online databases only via personal computers, or by going through time-consuming and repetitive navigation on mobile web browsers.

In addition to offering the data, KickInsights will also allow users to extract insights from them via interfaces for statistical comparison, and even further utilize them with a data-driven match prediction system powered by ML algorithms.

In the preceding Section 2, the objectives, scope, and design for core functionalities of the app will be discussed, followed by methodology for frontend and backend development in Section 3. Finally, the project budget and schedule will be explained in Section 4.

2. Project Objective and Scope

2.1 Objective

This project aims to develop KickInsights to solve the inefficiencies of current mobile application alternatives, and to integrate ML components with the app to enhance the utilization of the football data provided. To achieve the outcome, the scope of data will be discussed in Section 2.2 and the core functionalities of the app will be elaborated upon in Section 2.3. The list is not exhaustive as extra features may be incorporated into the app depending on the future progress of the project.

2.2 Scope of Data

This project focuses on the Premier League for three reasons. Firstly, it is the most popular football league in the world followed by an estimated 3.2 billion global fanbase [5]. Secondly, it has extensive and reliable data across multiple online databases readily available for scraping. Thirdly, it has been consistently generating high-quality match content for the past decades, assuring that the data possess significant reference value representing football as a whole.

2.3 Core Functionalities

Visuals for Sections 2.3.1 to 2.3.5 are captured from the native environment (iOS). For Sections 2.3.6 to 2.3.10, graphical designs from the app prototype are adopted. Data for the native app and prototype is as of January 11, 2024 and September 16, 2023 respectively.

2.3.1 League Table

The live Premier League table of the current season is provided in the “Table” tab in the menu bar, as shown in Figure 8, so that users can easily access it. Each row contains a club’s ranking, name, matches played (PL), wins(W), draws (D), losses(L), goal difference (GD), and points (Pts). To evaluate the home and away performance of each club, users can select respective tables on the navigation bar on top of the screen, as shown in Figure 9 and 10. Moreover, when a club is clicked, users will be redirected to the information page of that club (Section 2.3.2).

#	CLUB	PL	W	D	L	GD	Pts
1	Liverpool	20	13	6	1	25	45
2	Aston Villa	20	13	3	4	16	42
3	Man City	19	12	4	3	24	40
4	Arsenal	20	12	4	4	17	40
5	Tottenham	20	12	3	5	13	39
6	West Ham	20	10	4	6	3	34
7	Brighton	20	8	7	5	5	31
8	Man Utd	20	10	1	9	-5	31
9	Newcastle	20	9	2	9	10	29
10	Chelsea	20	8	4	8	3	28
11	Wolves	20	8	4	8	-1	28
12	Bournemouth	19	7	4	8	-7	25
13	Fulham	20	7	3	10	-7	24
14	Crystal Palace	20	5	6	9	-7	21

Figure 8. Table Tab

#	CLUB	PL	W	D	L	GD	Pts
1	Aston Villa	10	9	1	0	21	28
2	Liverpool	10	8	2	0	18	26
3	Newcastle	10	8	0	2	16	24
4	Arsenal	10	7	2	1	12	23
5	Man City	9	6	3	0	15	21
6	Tottenham	10	7	0	3	7	21
7	Brighton	10	5	4	1	8	19
8	Fulham	10	6	0	4	7	18
9	West Ham	10	5	3	2	7	18
10	Wolves	10	5	3	2	2	18
11	Man Utd	10	6	0	4	-3	18
12	Chelsea	10	4	3	3	3	15
13	Brentford	10	3	3	4	-1	12
14	Forest	10	3	3	4	-1	12

Figure 9. Home table

#	CLUB	PL	W	D	L	GD	Pts
1	Man City	10	6	1	3	9	19
2	Liverpool	10	5	4	1	7	19
3	Tottenham	10	5	3	2	6	18
4	Arsenal	10	5	2	3	5	17
5	Everton	10	5	1	4	-3	16
6	West Ham	10	5	1	4	-4	16
7	Aston Villa	10	4	2	4	-5	14
8	Chelsea	10	4	1	5	0	13
9	Man Utd	10	4	1	5	-2	13
10	Bournemouth	10	4	1	5	-6	13
11	Brighton	10	3	3	4	-3	12
12	Crystal Palace	10	3	3	4	-4	12
13	Wolves	10	3	1	6	-3	10
14	Burnley	10	2	2	6	-7	8

Figure 10. Away table

To refresh and update the table, users can pull it downwards, as shown in Figure 11.

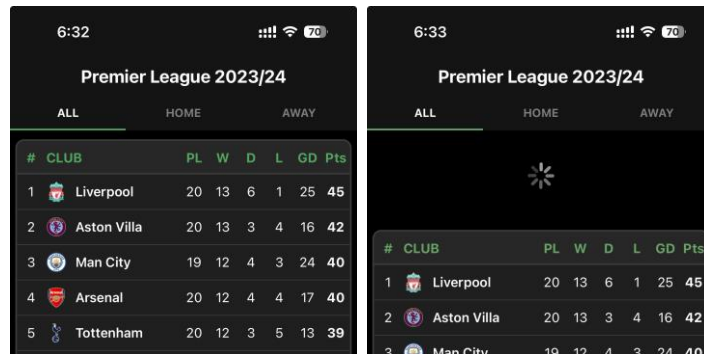


Figure 11. Refreshing the league table

2.3.2 Club Information

In the “Clubs” tab, there is a list containing all 20 clubs in the current Premier League season, as shown in Figure 12. Users can click into a club for its information page, which includes the overview, kits, the last and the next 3 games of the club, as shown in Figure 13. Figure 14 illustrates that if a game is postponed or rescheduled, it will be reflected on the page.



Figure 12. Clubs Tab

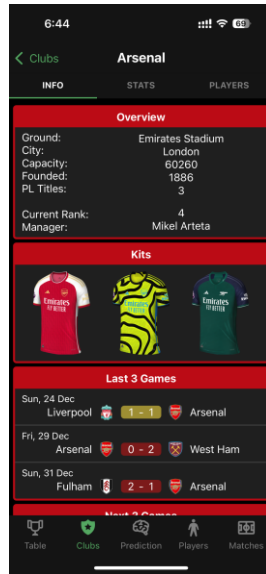


Figure 13. Club Info



Figure 14. A postponed game

Moreover, the “Stats” page provides in-depth club statistics, which includes a wide range of seasonal data segmented into General, Shooting, Passing, Pass Types, Goal and Shot Creation, Defensive Actions, Possession, Goalkeeping, and Miscellaneous data, as shown in Figure 15. Altogether, the page contains 117 rows of data, the most comprehensive among existing mobile apps, yet remains organized and thus easy to navigate. Figure 16 and 17 showcase data included in the page’s “Shooting” and “Possession” aspects.

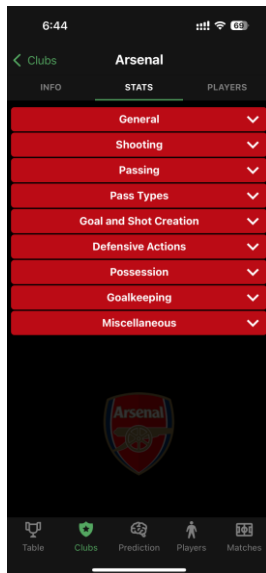


Figure 15. Club Stats

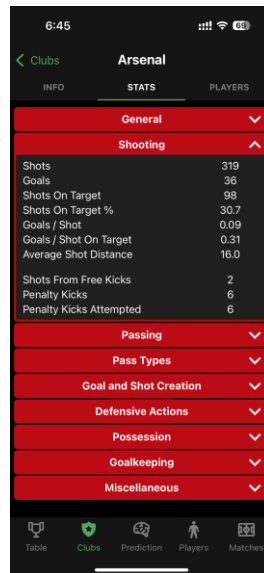


Figure 16-17. Shooting and Possession aspect in Club Stats



The “Players” page stores players of the club sorted according to their playing positions, as shown in Figure 18. Users can view the general information, like nationality, age, and market value, of a player by tapping on the player's box, as shown in Figure 19.



Figure 18. Club Players

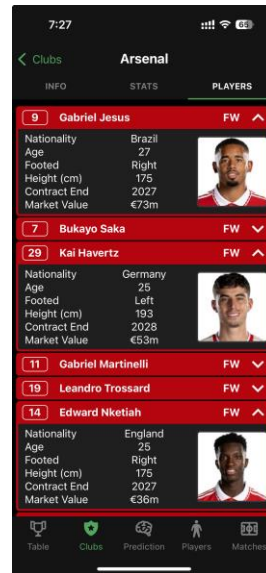
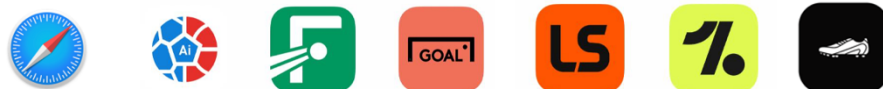


Figure 19. Players information

To conclude, the “Clubs” tab and information page provide simple navigation to access comprehensive club statistics. Table 2 compares the number of clicks required to locate a club’s statistics, in which KickInsights equalled the best-performing alternative, FotMob.



Clicks required to locate a club's statistics	Browser	AiScore	FotMob	GOAL	LiveScore	One-Football	KickInsights
Minimum	6	NA	4	NA	6	6	4
Average (Approximate)	7	NA	5	NA	7	7	5

Table 2. Comparison of clicks required to locate a club’s statistic among popular football apps* and KickInsights

* AiScore and GOAL do not provide club statistics.

2.3.3 Club Comparison

Next to the club list is the page for club statistics comparison. Users can pick two teams and directly compare their seasonal data in a side-by-side manner, as shown in Figure 20.

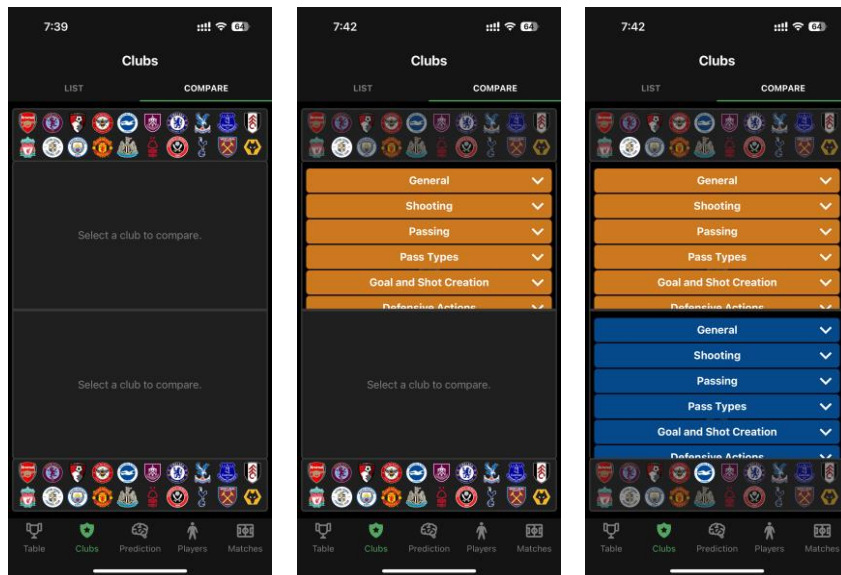


Figure 20. Steps to compare two teams in Club Comparison

To compare an aspect between two clubs, users can expand the respective sections. With two more clicks, a new aspect or a new club can be compared, as shown in Figure 21.

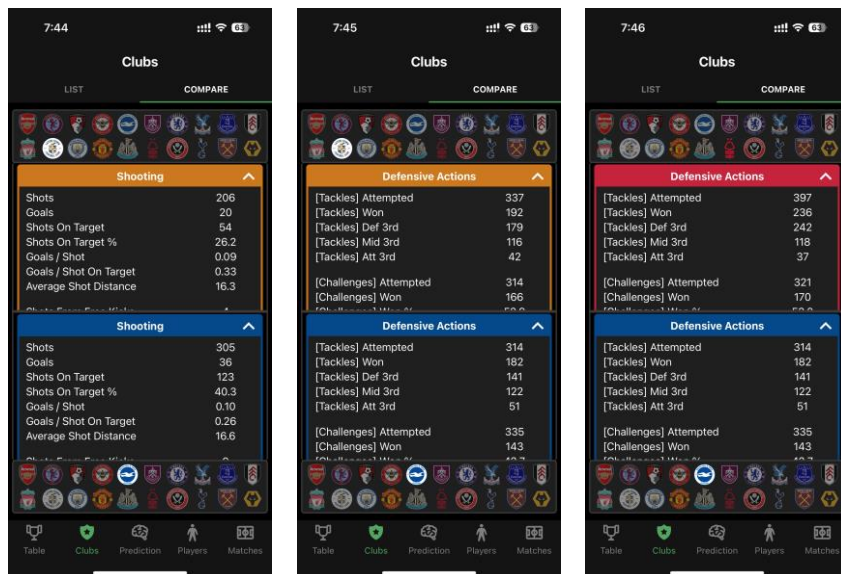


Figure 21. Comparing a new aspect and a new club in Club Comparison

This single-screen interface not only supports comparing multiple attributes on the same screen, but also requires only two more clicks to compare new statistics, avoiding the need to switch between tabs in each new comparison, as in other apps and the mobile browser. Therefore, KickInsights significantly reduces the effort needed to compare seasonal statistics between two clubs, as shown in Table 3.



Average clicks required to compare club statistics	Browser	AiScore	FotMob	GOAL	LiveScore	One-Football	KickInsights
First comparison	14	NA	10	NA	14	14	6
Second comparison	28	NA	20	NA	28	28	8
Third comparison	42	NA	30	NA	42	42	10

Table 3. Comparison of the average clicks required to compare club statistics among popular football apps* and KickInsights

* AiScore and GOAL do not provide club statistics.

2.3.4 Player Information

In the “Players” tab, there is a list containing all 571 Premier League players. For each player, the club, number, and position will be displayed, as shown in Figure 22. By simply tapping on the player's row in the list, users can access the player information page, which contains both the general information and in-depth statistics of a player, as shown in Figure 23.



Figure 22. Players Tab

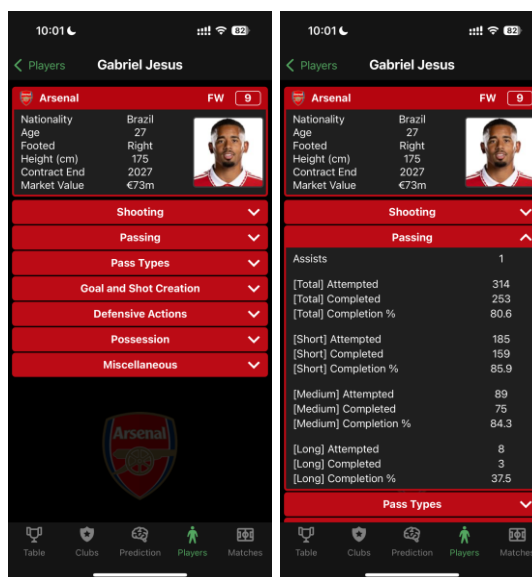


Figure 23. General information and statistics of a player

Figure 24 shows that for goalkeepers, the information page contains specific statistics like goalkeeping and advanced goalkeeping on top of the standard statistics for on-field players.

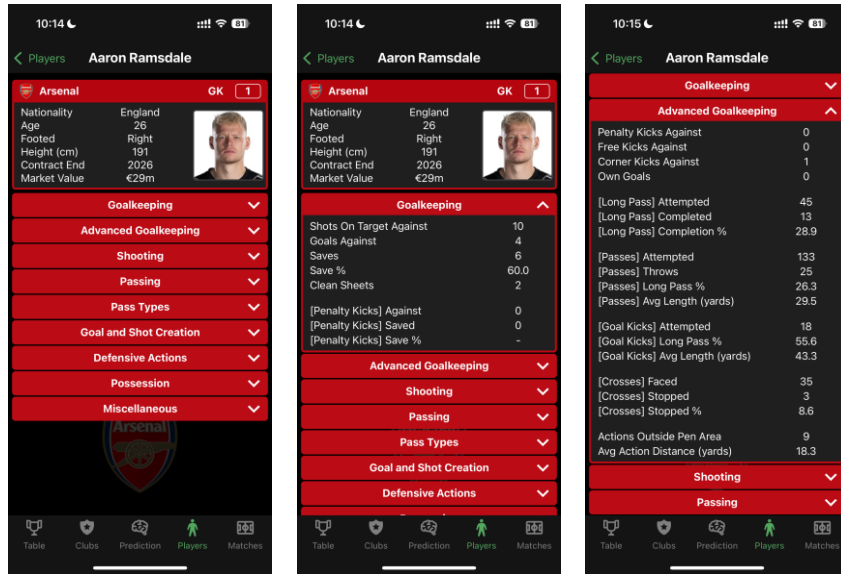


Figure 24. General information and statistics of a goalkeeper

To facilitate navigation, the “Players” tab provides the search and filter functions, which are both absent in existing apps. The functions can be used in any combination. For example, all defenders (DF) whose name includes “tom” can be selected, as shown in Figure 25.

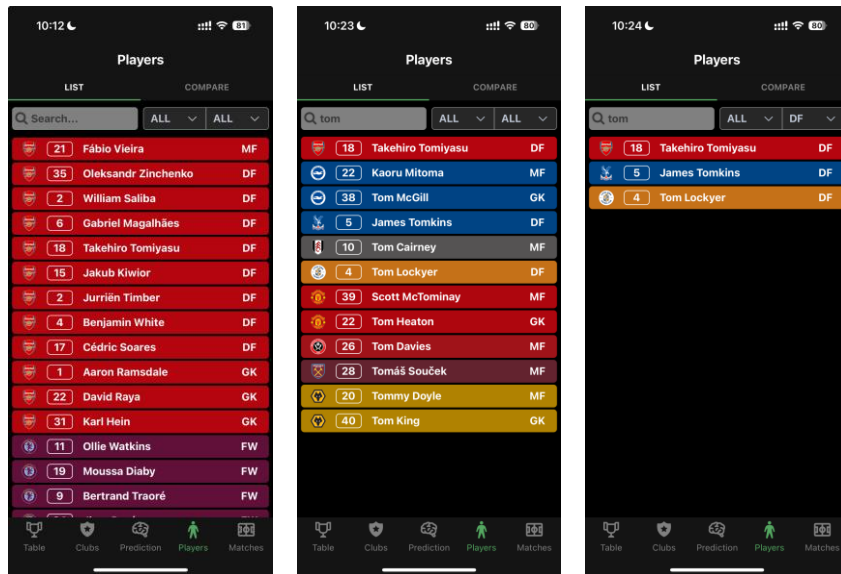


Figure 25. Locating a player with the search and filter function

Figure 26 illustrates that the club and position filter allow users to locate any player in just four clicks, in which two are used for filtering the club and two are used for filtering the position.

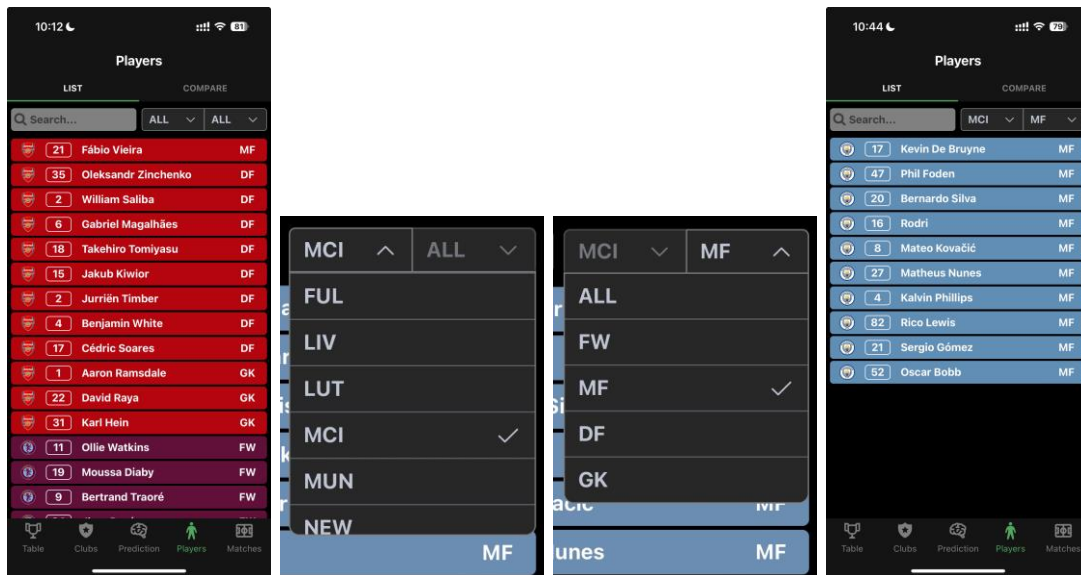


Figure 26. Locating a player with the club and position filter

Therefore, the “Players” tab offers flexibility and ease in locating the information pages of all players, which contain comprehensive player statistics. As shown in Table 4, KickInsights beats other apps in terms of clicks required to locate a player’s statistics.



Clicks required to locate a player's statistics	Browser	AiScore	FotMob	GOAL	LiveScore	One-Football	KickInsights
Minimum	6	6	7	NA	7	6	3
Average (Approximate)	12	8	9	NA	9	9	7

Table 4. Comparison of clicks required to locate a player’s statistic among popular football apps* and KickInsights

* GOAL does not provide player statistics.

Moreover, since there are two transfer windows every season when players join new clubs, users may not be aware of the latest club a transferred player belongs to. While it will be difficult to locate the player in other apps without explicitly looking for transfer news on the internet, users of KickInsights can easily locate the player with the search function.

2.3.5 Player Comparison

Similar to the club comparison function, an interface will also be provided in the “Players” tab for direct comparison of player statistics, as shown in Figure 27. Thus, cross-club and cross-positional comparisons can be made on a single page. Comparison of players in the same club

will also be supported. To add a player for comparison, long-press the player in the player list, as shown in Figure 28.

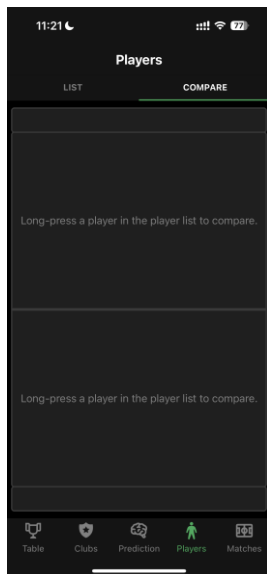


Figure 27. Comparison interface

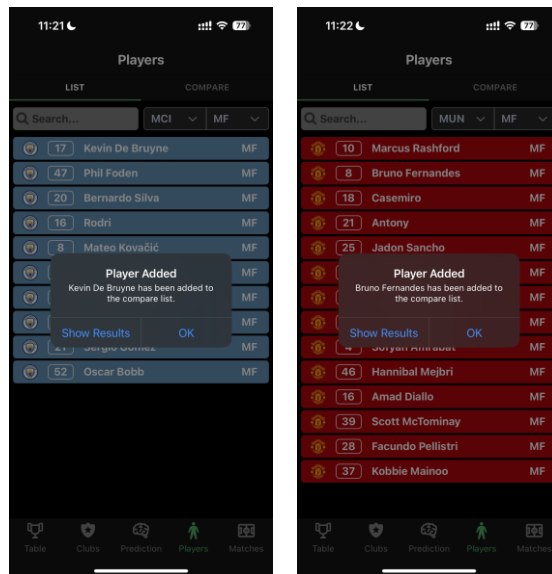


Figure 28. Adding a player for comparison

As shown in Figure 29, besides manually navigating to the comparison interface, users can also view the comparison results by clicking the alert. Figure 30 illustrates that if there are already two players being compared, the alert will require the user to replace one of them.

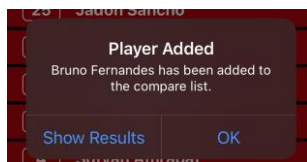


Figure 29. "Player Added" alert

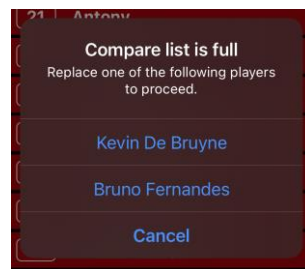


Figure 30. "Compare list is full" alert

Figure 31 shows the comparison interface after two players are added. Note that a frontend design consistent with the club comparison screen is implemented. To compare an aspect between two players, the respective sections can be expanded with two clicks.

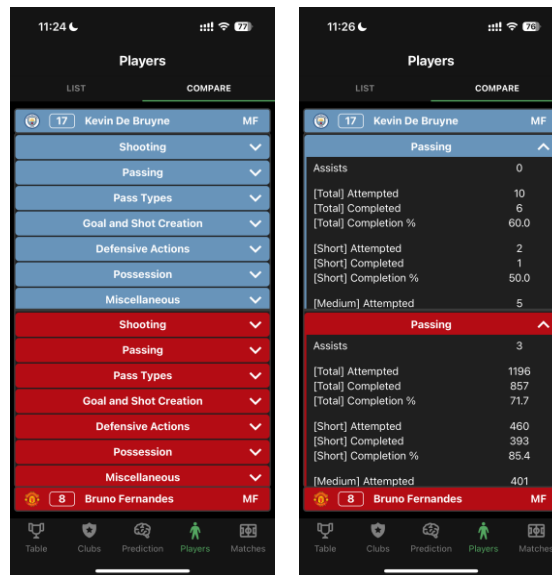


Figure 31. Comparing an aspect in Player Comparison

Similar to the club comparison function, this interface supports comparing multiple attributes on the same screen and avoids the need to switch between tabs. Therefore, once again, KickInsights greatly reduces the effort needed to compare seasonal statistics between two players, as shown in Table 5.



Average clicks required to compare player statistics	Browser	AiScore	FotMob	GOAL	LiveScore	One-Football	KickInsights
First comparison	24	16	18	NA	18	18	14
Second comparison	48	32	36	NA	36	36	16
Third comparison	72	48	52	NA	52	52	18

Table 5. Comparison of the average clicks required to compare player statistics among popular football apps* and KickInsights

* GOAL does not provide player statistics.

2.3.6 Matches Tab

All matches in the current matchweek will be displayed in real time. For example, assume that the match between Newcastle and Brentford is ongoing in the 69th minute, the live scoreline (1-0) will be displayed, as shown in Figure 32. Other matches in the past will be labelled as full-time (FT), while upcoming matches will be shown with their starting time. With the navigation bar on top of the screen, users will be able to view previous results and future schedules for other matchweeks.

2.3.7 Match Centre

For each match, users will be able to access the match centre by clicking the match in the “Matches” tab. As shown in Figure 33, the match centre will contain two pages. The “Stats” page contains in-game statistics for that match, including the goalscorers and head-to-head indicators like possession, passing accuracy, etc., while the “Squad” page contains the formation, first team, and substitute players used by both sides in the match.

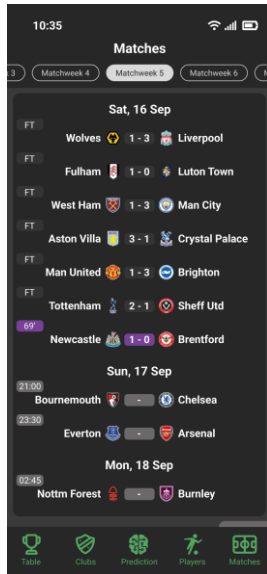


Figure 32. Matches Tab

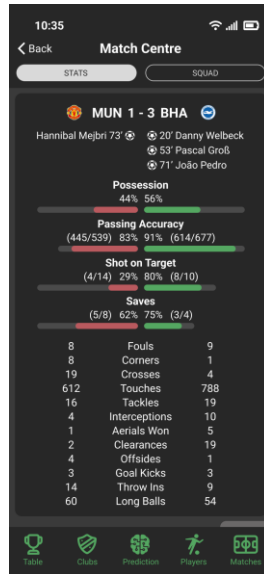
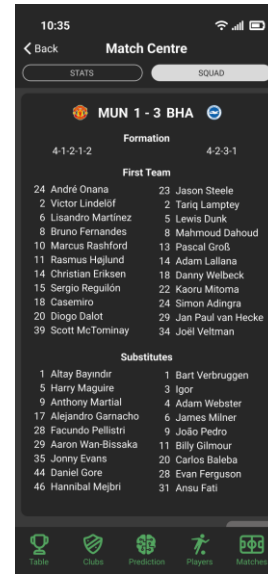


Figure 33. “Stats” and “Squad” page in Match Centre



2.3.8 Machine Learning Prediction (Community Models)

The “Prediction” tab, being the default app page, will contain the unique interactive ML elements. Figure 34 shows the “Community” page, which will include different ML predictive models created by app users (see Section 2.3.9). For each model, its name, algorithms selected, and attributes used will be displayed. The accuracy or F1 score attained by the model will also be shown. Users will be able to leave their feedback for each model by clicking the like or dislike button.

Moreover, the page will support sorting and filtering, as shown in Figure 35. The models will be sorted by upload date by default, while users will also be able to perform sorting by model accuracy, F1 score, or the “Likes”. In addition, there will be two algorithm filters that can be applied to filter models trained by the algorithms selected.

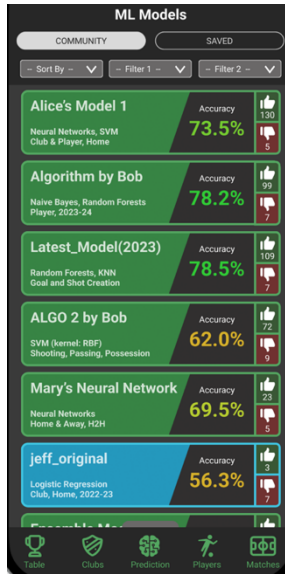


Figure 34. "Community" page

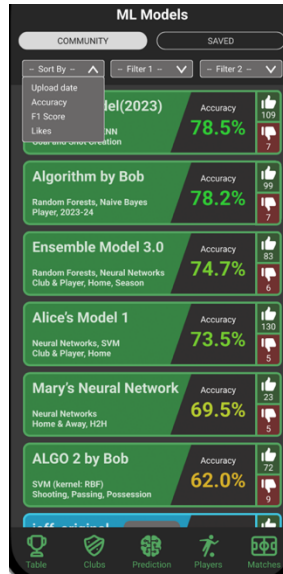
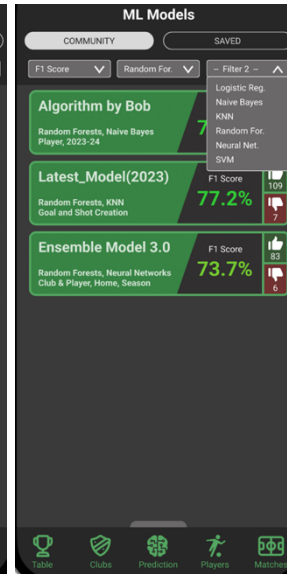


Figure 35. ML Model sorting and filtering



When a model is selected, information about it will be displayed in the "Data", "Training" and "Evaluation" tabs, as shown in Figure 36. In addition, prediction results from the model will be displayed, as shown in Figure 37.



Figure 36. Data, Training, and Evaluation Tabs

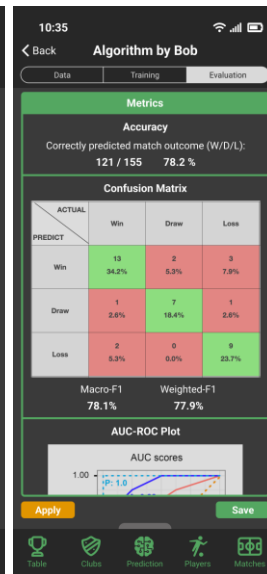
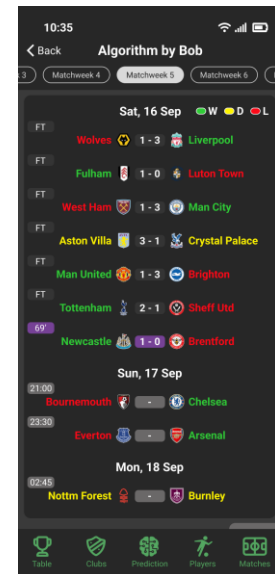


Figure 37. Prediction results



The "Data" tab will list all of the data selected for the model, including the seasons used, data type, and in-game statistics separated into different areas. In each area, users will be able to either select all the data available, include only specific ones, or not choose any for training the model.

The “Training” tab will show the algorithms used for training the model and their respective specifications. For example, users will be able to select the number of trees or neurons used, and the kernel for the SVM. Moreover, the tab will include weights that can be assigned for different attributes for model optimization.

The “Evaluation” tab will summarize the model performance in different evaluation metrics, including confusion matrix with F1 score, AUC-ROC plot, and categorical cross-entropy loss. It will allow users to comprehensively assess the performance of models in the community.

Users will also be able to apply the model and obtain match prediction results by clicking the “Apply” button. Then, for every match, the predicted winning team will be shown in green, the losing team in red, and yellow for both teams if a draw is predicted.

2.3.9 Creating Machine Learning Models

To enhance user experience, a consistent interface with the community ML models will be adapted to implement the “Data”, “Training”, and “Evaluation” tabs, as shown in Figure 38.

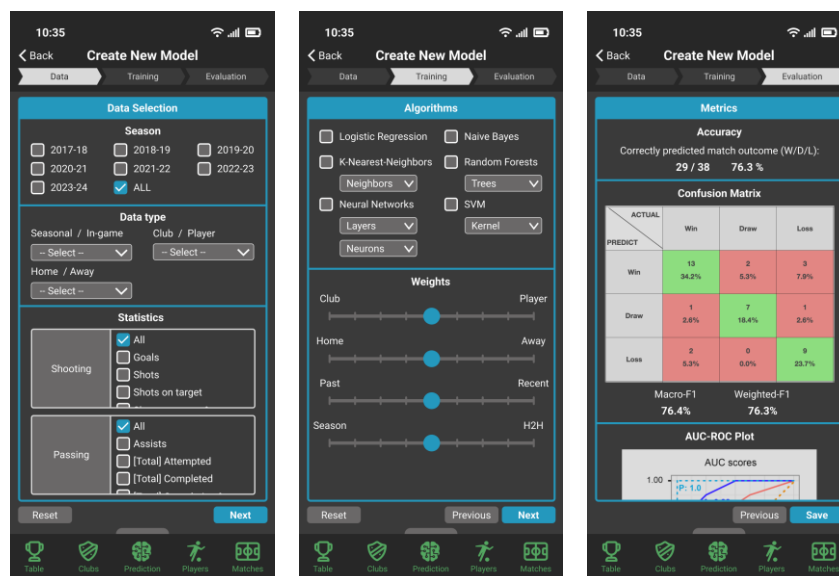


Figure 38. Data, Training, and Evaluation Tabs when creating ML models

To create a model, users should first select the data timeframe, appropriate data types, and statistics used for training the model. They should then select one or more algorithms for the model, specifications for each algorithm used, and weights for different attributes to further calibrate the model. Finally, evaluation results will be presented after the model is trained. The same metrics with the community models will be used.

2.3.10 Machine Learning Prediction (Saved Models)

On the “Saved” page shown in Figure 39, there will be two types of models, including the ones saved from the “Community” page and the ones created by the user, which will be colored green and blue respectively. Sorting and filtering functions for these saved models will also be provided.

When the user selects a model created by himself or herself, the model information page will be displayed, as shown in Figure 40. In addition to applying the model to view match prediction results, users will be able to update it by modifying or adding extra parameters. Moreover, the model can be renamed, published to the community, unpublished, and deleted.

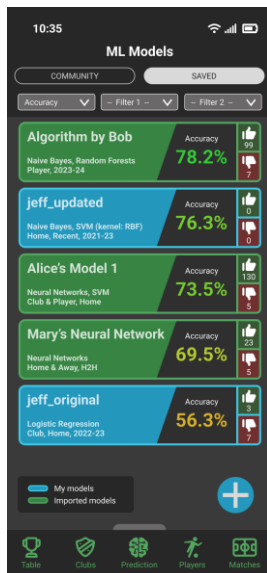


Figure 39. “Saved” page



Figure 40. ML model information page

3. Methodology

This chapter provides a detailed discussion of the implementation of the project. To start with, Section 3.1 covers the frontend development tools that are used for prototype design, UI and UX, and application development. Section 3.2 explores the platforms and methods that are used in backend development in aspects of data collection, wrangling, standard, and application development. Finally, Section 3.3 reviews the difficulties faced in the development process and explains the solution adopted to mitigate the impact.

3.1 Frontend development

3.1.1 Prototype

Figure 41 shows the complete prototype of the app created with Figma, a web-based design tool. With the support for interactive flows which allows designers to explore and test user interactions with the interface [6], it improves the efficiency of the design process. The GUI designs are then used as a reference for initial frontend development and may be modified and optimized in the latter part of the project.

The narrated prototype demonstration can be accessed via the following link.

<https://youtu.be/ymzIQAWuqTk>



Figure 41. Prototype of KickInsights created with Figma

3.1.2 UI and UX

To streamline UI and UX within the app, the React Navigation library is used because it allows developers to easily implement navigation systems like stacks, drawers, and tabs in their mobile apps [7]. The code is highly customizable and can be implemented in JavaScript, which is the main programming language for app development.

Within the library, the Bottom Tab Navigators and Top Tab Navigators are used for traversal between the five main screens and their sub-screens respectively, while the Stack Navigator is used to display detailed club and player statistics. These navigators provide an overall

convenient and consistent user interface, with smooth loading screens and transition animations to sustain users' immersiveness in the app.

3.1.3 Application development

React Native is used with Visual Studio Code as the IDE to first develop code in JavaScript, and then convert the code into native languages. It provides high code reusability because only a single codebase is required for both iOS and Android development. Moreover, development time can be shortened as the open-source framework offers abundant pre-developed components with relevant community support [8].

Expo is used to facilitate the development process. It contains a well-established set of tools and services oriented around React Native for app developers to write and build mobile apps with ease [9]. It also supports hot reloading on emulators and mobile devices, allowing developers to see the results in real time after changes in the React Native code [8].

3.2 Backend development

3.2.1 Data Collection

The two main data sources required for the project is firstly, general club and player information used for tables and lists in the main tabs, and secondly, in-depth seasonal and in-game statistics used for the "stats" pages, match centres, and ML prediction models.

General information is pipelined from RapidAPI, the largest API platform in the world which allows integration of APIs into different applications [10]. It reduces the time and effort needed compared to individual searches for data sources. Additionally, the platform provides code snippets in JavaScript with detailed documentation to facilitate the integration process.

FootApi is selected among various APIs to be the data source for general information since it offers real-time football data like league table standings, fixtures, match detail, etc., with latency within 505 milliseconds and a rate limit of 6 requests per second, which is sufficient for the project scope. Moreover, data is outputted in readily available JSON format.

On the other hand, in-depth statistics are scraped from FBREF, one of the best football data providers constantly collecting team and player statistics, covering the sports comprehensively

in a data-driven approach [11]. Its online database is set as the data endpoint for web-scraping, in which Axios, a JavaScript library, is used due to its robust error-handling mechanism.

3.2.2 Data Wrangling and Standard

After obtaining data from FootApi and FBREF, structuring and cleaning are performed to remove redundant attributes and maintain consistency. For example, players' full names and playing positions are standardized to the FBREF version. Furthermore, Transfermarkt is adopted as the standard for club and player information, as it possesses an extensive database of club and player profiles widely used as the industry benchmark [12].

Additional attributes like the player image URLs, linked to the cloud storage explained in Section 3.2.3, are created in the database for frontend display. Player names with removed diacritics are also used to facilitate the searching function introduced in Section 2.3.4.

3.2.3 Application development

The application backend and database are constructed with Firebase, Google's BaaS which provides robust infrastructure for data storage. The distributed architecture not only provides load balancing for app usage but also syncs data in real-time across data centres, preventing bottlenecks at any specific server.

The wrangled football data is stored in the Google Cloud Firestore Database while player images are stored in the Firebase Storage, as they support offline access of app resources loaded the previous access. They also offer great scalability and data handling capabilities under a cost-effective pricing scheme based on actual usage [13].

A temporary panel, as shown in Figure 42, has been set up to run scripts that fetch data from the API and web endpoints, and then store them into the Firestore Database. When the app is launched, it automatically requests the latest data from the Firestore Database with the loading screen as shown in Figure 43.

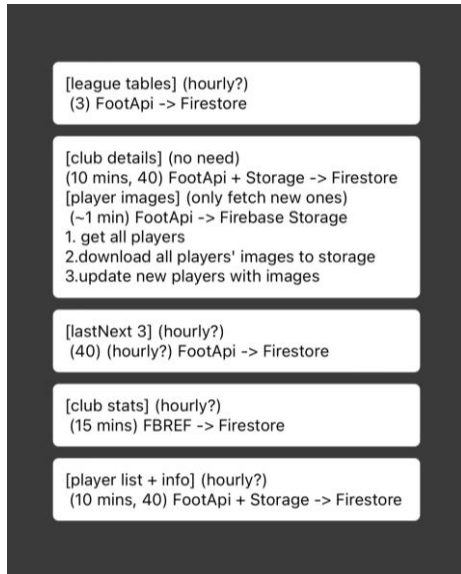


Figure 42. Temporary panel for data fetching



Figure 43. Loading screen

To automate the fetching process, the scripts used in the panel will be migrated to Google's Cloud Functions after the optimal update frequency is calculated based on the fetch time, FBREF and FootApi's fetch limit, and Firestore's usage quota, etc.

Finally, Google's Cloud Run API will be used to serve app requests to build, train, and evaluate the ML models because it is fully compatible with the Node.js framework and the Firestore Database, allowing developers to deploy code using a simple command-line interface [14]. For developing the ML components, Python libraries like scikit-learn and TensorFlow will be used as they cater to diverse models from traditional SVMs and clusterings to advanced artificial neural networks. Further implementations for ML will be covered in the final report.

3.3 Difficulties and Mitigations

One of the major challenges faced in the app development process is the compatibility of frontend interfaces with mobile devices. In recent years, devices from various manufacturers have come in a variety of shapes and sizes, making it difficult to apply a single layout onto different physical screens without distortion or other undesired visual effects. On the other hand, it is not time and resource-efficient to customize an interface for each device.

To mitigate the impact, frontend development and testing have been conducted on the iPhone SE (3rd generation), iPhone XR, and iPhone 14 Pro. These devices were selected because of their distinct screen sizes, diagonally measured from 4.7-inch to 6.1-inch [15]. The wide range

increases the probability that the app will be compatible with most devices. Moreover, the three iPhones were selected because the project prioritizes launching the app on iOS.

Table 6 shows the variation in UIs among the three devices. The size of the devices is proportional.

Device \ Tab	Table	Clubs	Players
iPhone SE (3rd generation)			
iPhone XR			
iPhone 14 Pro			

Table 6. User Interfaces of KickInsights on various devices

4. Project Budget and Schedule

This chapter first introduces the budget for the project in Section 4.1. Then, the project schedule will be explained in Section 4.2. Progress of the deliverables and immediate next steps will also be reported.

4.1 Budget

The estimated budget for the project is listed in Table 7. Three services have been purchased including the FootApi Pro Plan, Firebase BaaS, and Cloud Run. For Firebase BaaS, the price is approximated based on the expected usage of Google Authentication, Cloud Functions, Cloud Storage, and Cloud Firestore. Billing metrics include the number of upload and download operations, document reads and writes, etc. Additional services for ML may be deployed in the future.

Service	Provider	Price (HKD)
FootApi Pro Plan	RapidAPI	\$398.05
Firebase BaaS*	Google	\$50.00
Cloud Run*	Google	\$150.00
Estimated total expenditure		\$598.05

* The service provides a free tier but subsequent usage will be charged.

Table 7. Estimated Budget for the project

4.2 Schedule

The schedule of the project is outlined in Table 8. The preliminary results are as follows. Most of the frontend screens and components have been developed. The remaining “Machine Learning” and “Matches” tabs will be completed in January. For backend development, the data collection and wrangling processes have been established. They will be automated with Google Cloud Functions in February. For Machine Learning, research on models and evaluation metrics has been conducted. The models will be integrated with the app and optimized throughout semester two. Overall, the project deliverables are completed on schedule.

Time Period	Tasks / Deliverables	Status
September 2023	<ul style="list-style-type: none"> • Detailed project plan • Project web page • UI / UX design • App prototype 	Completed
October 2023	<ul style="list-style-type: none"> • Frontend development: Table tab • Backend and database design • Data collection and wrangling 	Completed
November 2023 - December 2023	<ul style="list-style-type: none"> • Frontend development: Clubs, Players tab • Backend and database development • Research on machine learning models 	Completed
January 2024	<ul style="list-style-type: none"> • First presentation • Preliminary implementation • Detailed interim report • Frontend development: Machine Learning, Matches tab • Integration of machine learning models 	In-progress
February 2024	<ul style="list-style-type: none"> • Implementation of Cloud Functions • Implementation of User Account System • Optimization of machine learning models 	TBC
March 2024	<ul style="list-style-type: none"> • Application testing • Codebase review • Improvement in UI / UX • Optimization of machine learning models 	TBC
April 2024	<ul style="list-style-type: none"> • Final presentation • Final tested implementation • Final report 	TBC
May 2024	<ul style="list-style-type: none"> • Project exhibition 	TBC

Table 8. Project Schedule Timetable

5. Conclusion for preliminary work

The project aims to develop KickInsights, an app that tackles the inefficiencies of current market alternatives by bringing comprehensive football data onto mobile devices, offering statistical comparison interfaces for users to evaluate the data and extract insights, and finally, implementing a data-driven match prediction system powered by ML algorithms.

The report first introduces the project's background and motivation to highlight its significance, then covers the objectives, data scope, and design for the core functionalities of the app. Methodologies for frontend and backend development are also discussed, followed by the challenges faced and their mitigations. Finally, the project budget and schedule are reported.

The overall progress is on track with the project schedule. Three out of five frontend screens have been delivered with a structured backend database, supported by data collection and wrangling processes. Hence, the immediate next steps will be completing the frontend development, in order to carry on with the ML integration task. Further results will be updated and discussed in the final report.

6. References

- [1] Data Sports Group, “Keeping up with the game: Football data feeds for fans and teams,” Medium, https://medium.com/@marketing_25315/keeping-up-with-the-game-football-data-feeds-for-fans-and-teams-45881110f9db (accessed Nov. 17, 2023).
- [2] Sportico, “Fan demand drives the change in sports media consumption,” Sportico.com, <https://www.sportico.com/business/sports/2022/fan-demand-drives-the-change-in-sports-media-consumption-1234669318/> (accessed Nov. 17, 2023).
- [3] Jsant, “Top 10 football apps for you to have on your phone!,” Real Valladolid Academy, <https://realvalladolidacademy.com/en/the-10-best-football-apps-for-you-to-have-on-your-mobile/> (accessed Nov. 18, 2023).
- [4] L. Chung, “The 7 types of sampling and response bias to avoid in customer surveys,” Delighted, <https://delighted.com/blog/avoid-7-types-sampling-response-survey-bias> (accessed Nov. 18, 2023).
- [5] H. Jain, “Top 10 most watched football leagues in the world,” SportingWiki, <https://www.sportingwiki.com/top-10/top-10-most-watched-football-leagues/> (accessed Nov. 18, 2023).
- [6] F. Flinn, “Figma prototype: What is it and why use it for design?,” Figma Prototype: What is it and why use it for design? | The Design Project, <https://designproject.io/blog/figma-prototype> (accessed Nov. 18, 2023).
- [7] H. Hariyani, “What’s new in react navigation 6,” Medium, <https://medium.com/simform-engineering/whats-new-in-react-navigation-6-8161eefacc8c> (accessed Nov. 20, 2023).
- [8] H. Agarwal, “Advantages and disadvantages of using react native in 2023,” Techexactly, <https://techexactly.com/blogs/advantages-and-disadvantages-of-using-react-native> (accessed Nov. 20, 2023).
- [9] A. Ravichandran, “Building react native apps - expo or not?,” Medium, <https://adhithiravi.medium.com/building-react-native-apps-expo-or-not-d49770d1f5b8> (accessed Nov. 20, 2023).
- [10] K. Haewon, “RapidAPI review: What is rapidapi and how to use it,” Apidog Blog, <https://apidog.com/blog/what-is-rapidapi-and-how-to-use-it/> (accessed Nov. 21, 2023).
- [11] Toby, “Best sites for Free Football Statistics: Top Soccer Stats websites,” Punter2Pro, <https://punter2pro.com/free-football-statistics-soccer-stats/#:~:text=Fbref%20is%20a%20website%20that,match%20breakdowns%20to%20p-layer%20transfers.> (accessed Nov. 20, 2023).
- [12] C. Wheatley, “Secrets behind transfermarkt and how clubs and players use the platform,” Football London, <https://www.football.london/premier-league/secrets-behind-transfermarkt-how-football-21956019> (accessed Nov. 20, 2023).

- [13] Pathik, “7 reasons to choose google cloud firestore as your database solution,” BlueWhaleApps, <https://bluewhaleapps.com/blog/7-reasons-to-choose-google-cloud-firestore-as-your-database-solution> (accessed Nov. 22, 2023).
- [14] K. Wisniowski, “Cloud run vs app engine vs cloud function (pros and cons),” Cloud Infrastructure Services, <https://cloudinfrastructureservices.co.uk/cloud-run-vs-app-engine-vs-cloud-function/> (accessed Nov. 23, 2023).
- [15] Apple, “Compare iPhone models,” Apple, <https://www.apple.com/ph/iphone/compare/?modelList=iphone-SE-3rd-gen%2Ciphone-14-pro> (accessed Nov. 23, 2023).

7. Appendices

Appendix A

Definition of in-depth team data:

Contains at least one datum from each of the following areas.

General:

Record (W-D-L), Goals For, Goals Against, Clean Sheets

Shooting:

Goals, Shots, Shots on target, Shots on target %, Goals/Shot, Goals/Shot on target, Average Shot Distance, Free kicks, Penalty kicks, Penalty kicks attempted

Passing:

Assists, [Total] Attempted, [Total] Completed, [Total] Completion %, [Short] Attempted, [Short] Completed, [Short] Completion %, [Medium] Attempted, [Medium] Completed, [Medium] Completion %, [Long] Attempted, [Long] Completed, [Long] Completion %

Pass Types:

Live-ball passes, Dead-ball passes, Free Kicks, Through Balls, Switches, Crosses, Throw-Ins, Corner Kicks, Offsides, Passes blocked

Goal and Shot Creation:

Shot Creating Actions: [SCA] Live-Ball passes, [SCA] Dead-Ball passes, [SCA] Take-Ons, [SCA] Shots, [SCA] Fouls, [SCA] Defensive actions

Goal Creating Actions: [GCA] Live-Ball passes, [GCA] Dead-Ball passes, [GCA] Take-Ons, [GCA] Shots, [GCA] Fouls, [GCA] Defensive actions

Defensive Actions:

Tackles, Tackles Won, Tackles in Def 3rd, Tackles in Mid 3rd, Tackles in Att 3rd, Challenges, Challenges Won, Blocks, Shots blocked, Passes blocked, Interceptions, Clearances, Errors

Possession:

Possession, Touches, Touches in Def penalty area, Touches in Def 3rd, Touches in Mid 3rd, Touches in Att 3rd, Touches in Att penalty area, Take-Ons, Take-Ons Success, Take-Ons Tackled, Carries, Total Distance (yards), Progressive Distance (yards), Progressive Carries, Carries into Att 3rd, Carries into Att penalty area, Miscontrols, Dispossessed, Passes Received, Progressive Passes Received

Goalkeeping:

Shot on Target Against, Goals Against, Saves, Save %, Clean Sheets, Long Pass Attempted, Long Pass Completed, Completion %, Passes Attempted, Throws Attempted, Long Pass %, Average Pass Length (yards), Goal Kicks Attempted, Long Goal Kick %, Average GK Length (yards), Crosses Faces, Crosses Stopped, Actions outside penalty area, Average action distance (yards)

Miscellaneous:

Yellow Cards, Red Cards, Second Yellow Cards, Fouls Committed, Fouls Drawn, Penalty Kicks Won, Penalty Kicks Conceded, Own Goals, Ball Recoveries, Aerial Duels Won, Aerial Duels Lost, Aerial Duels Won %

Appendix B

Definition of in-depth on-field player data (Defender, Midfielder, Forward):
Contains at least one datum from each of the following areas.

General:

Nationality, Age, Appearances, Goals, Assists, Tackles

Shooting:

Goals, Shots, Shots on target, Shots on target %, Goals/Shot, Goals/Shot on target, Average Shot Distance, Free kicks, Penalty kicks, Penalty kicks attempted

Passing:

Assists, [Total] Attempted, [Total] Completed, [Total] Completion %, [Short] Attempted, [Short] Completed, [Short] Completion %, [Medium] Attempted, [Medium] Completed, [Medium] Completion %, [Long] Attempted, [Long] Completed, [Long] Completion %

Pass Types:

Live-ball passes, Dead-ball passes, Free Kicks, Through Balls, Switches, Crosses, Throw-Ins, Corner Kicks, Offsides, Passes blocked

Goal and Shot Creation:

Shot Creating Actions: [SCA] Live-Ball passes, [SCA] Dead-Ball passes, [SCA] Take-Ons, [SCA] Shots, [SCA] Fouls, [SCA] Defensive actions
Goal Creating Actions: [GCA] Live-Ball passes, [GCA] Dead-Ball passes, [GCA] Take-Ons, [GCA] Shots, [GCA] Fouls, [GCA] Defensive actions

Defensive Actions:

Tackles, Tackles Won, Tackles in Def 3rd, Tackles in Mid 3rd, Tackles in Att 3rd, Challenges, Challenges Won, Blocks, Shots blocked, Passes blocked, Interceptions, Clearances, Errors

Possession:

Possession, Touches, Touches in Def penalty area, Touches in Def 3rd, Touches in Mid 3rd, Touches in Att 3rd, Touches in Att penalty area, Take-Ons, Take-Ons Success, Take-Ons Tackled, Carries, Total Distance (yards), Progressive Distance (yards), Progressive Carries, Carries into Att 3rd, Carries into Att penalty area, Miscontrols, Dispossessed, Passes Received, Progressive Passes Received

Miscellaneous:

Yellow Cards, Red Cards, Second Yellow Cards, Fouls Committed, Fouls Drawn, Penalty Kicks Won, Penalty Kicks Conceded, Own Goals, Ball Recoveries, Aerial Duels Won, Aerial Duels Lost, Aerial Duels Won %

Appendix C

Definition of in-depth player data (Goalkeeper):

Contains at least one datum from each of the following areas.

General:

Nationality, Age, Appearances, Goals, Assists, Tackles

Goalkeeping:

Shot on Target Against, Goals Against, Saves, Save %, Clean Sheets, Long Pass Attempted, Long Pass Completed, Completion %, Passes Attempted, Throws Attempted, Long Pass %, Average Pass Length (yards), Goal Kicks Attempted, Long Goal Kick %, Average GK Length (yards), Crosses Faces, Crosses Stopped, Actions outside penalty area, Average action distance (yards)

Miscellaneous:

Yellow Cards, Red Cards, Second Yellow Cards, Fouls Committed, Fouls Drawn, Penalty Kicks Won, Penalty Kicks Conceded, Own Goals, Ball Recoveries, Aerial Duels Won, Aerial Duels Lost, Aerial Duels Won %