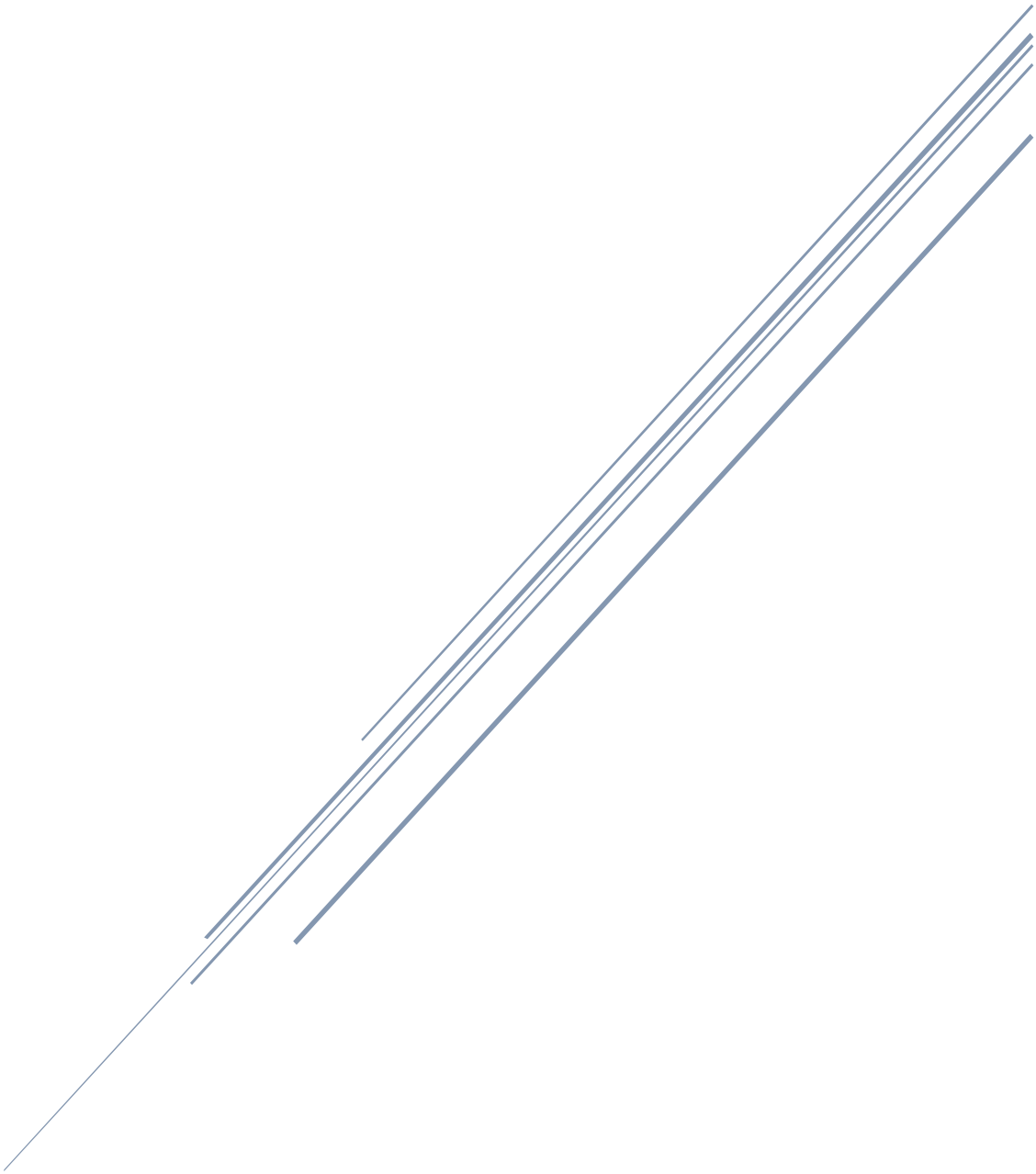
SMARTSCORE FUNCTIONAL SPECIFICATION

4th Year Project Software Development



Student: Aaron Doyle (C00272515)
Supervisor: Paul Barry
Date: 7/10/24

Table of Contents

Introduction

The aim for this specification is to outline some of the functionalities of my project. My project SmartScore is an AI focused fantasy football web application. The main aim for this application is to aid fantasy football managers on their weekly decisions using AI powered by statistics to give them recommendations for the upcoming week. Some of the features that will aid users on their decisions will be recommended transfers, player comparisons, team selection information. This specification aims to explain these features and describe the technologies to be used and the target audiences for the application.

Project Overview

The aim of this project is for the user to have a more informed idea of what they can do every game week in fantasy football and to allow for a new way to play fantasy football using statistics powered AI rather than the traditional scoring methods allowing for more fun and creative ways to play.

Some of the main features that I am looking to add to the application include:

- AI recommended transfers
- AI team selections
- Custom league rules
- Predicted points per week
- In-depth player comparisons

These features will allow users to make more educated decisions on their teams of a weekly basis.

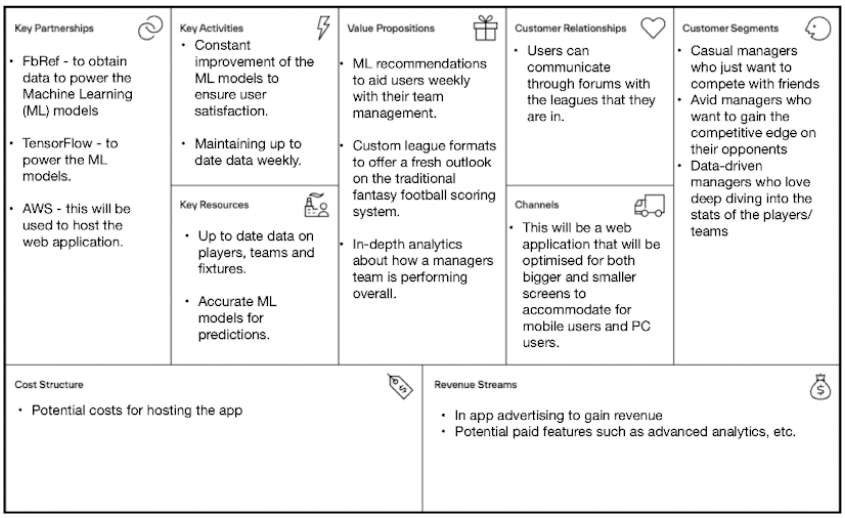


Fig 1. Business Model Canvas

Target Audience

For this application there are two primary types of user that SmartScore is targeted towards:

1. Casual fantasy football managers - SmartScore appeals to the casual manager as a big problem can be that they can feel out of their depth if they are in a league with friends who take it very seriously which leads to them abandoning the app after the first few game weeks due to feeling like they can't keep up

2. Avid fantasy football managers – SmartScore also aims to appeal to the managers who take fantasy football extremely seriously as the features provided in the app can give them the edge that they need to win in their competitive leagues with other knowledgeable managers.

Target Platforms

The target platform for this application will be a web based application. A web app will be used for SmartScore as it means that everyone can access the app without any downloads. This will also be optimised for both bigger and smaller screens so that it can be used on mobile when on the go or on a laptop/PC when at home.

Challenges To Overcome

One of the biggest obstacles that I will have to overcome is obtaining the data and structuring it correctly for the machine learning (ML) models.

I will have to make a data scraper using Python to go through the FBref website to get the data on players and teams so that I can use this to build out my models. Once this data is obtained it will have to be cleansed as it will not be in the correct structure for a ML model to use in its training.

Another big obstacle that I will face will be drawing users away from the Fantasy Premier League (FPL) app.

The FPL app is by far the biggest fantasy football application in the world with over 10 million players in the 2024/2025 season alone. I aim to do this by promoting the fact that SmartScore will be the only fantasy football application out there that combines both the game of fantasy football and AI recommendations for managers to help them with their team. Another alluring factor to SmartScore is the custom league rules feature which no other fantasy football application is using, this allows the users to have a fresh league structure rather than being forced to use the classic fantasy football scoring system which leads to some fun new ways to play the game.

Context Diagram

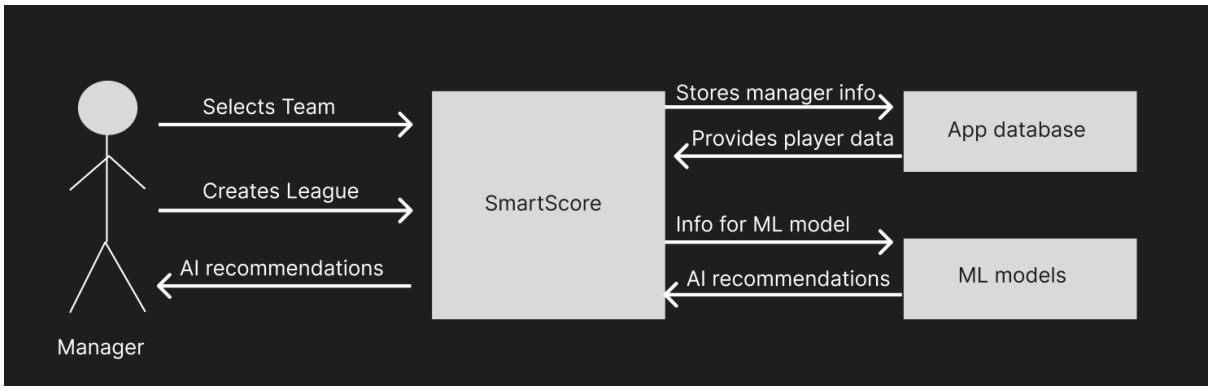


Fig 2. Context Diagram

This context diagram provides a basic overview of how the application interacts with the manager and the backend of the app. This shows some basic functionalities data flow between the different entities using the application.

Problem Statement

A big problem in fantasy football nowadays is that to be able to compete at a high level with friends or those online, you have to put a lot of time and effort into the game. For the casual player this idea isn't appealing as they mightn't have the time or effort to dedicate so much time to researching their team and transfer choices. For these casual players this can take a lot of fun out of the game and lead to managers giving up after only a few game weeks as they have fallen far behind their peers early.

This is where SmartScore wants to level the playing field. SmartScore aims to make a more level playing experience for the casual player by giving AI based recommendations in regards to team selection, transfers and player selections. By giving these to the average manager it allows them to keep up in their leagues with friends and enjoy the fantasy football experience for longer.

These features also appeal to the diehard managers as these recommendations can help them gain the competitive edge over their peers and help them win their leagues. By getting these recommendations it could be the difference of winning their league or losing it.

Overall SmartScore aims to provide a platform where friends can have a competitive and enjoyable league regardless of football knowledge or previous experience of playing other fantasy football games.

Use Case Diagram

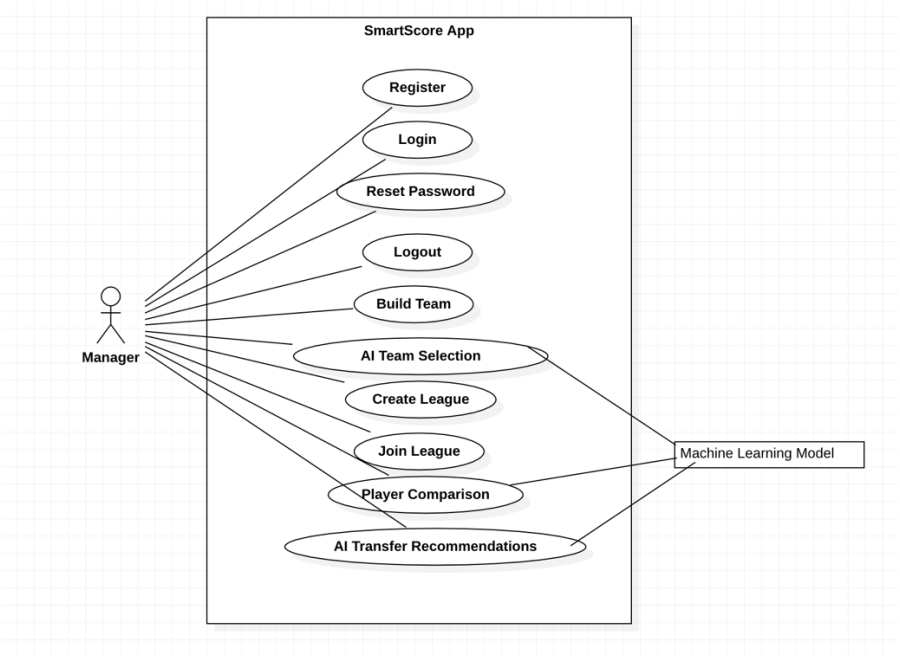


Fig 3. Use Case Diagram

Brief Use Cases

Use Case Name: Register
Actor: Manager
Description: This use case begins when the manager loads into the web app. They then proceed to select a username, email and password and the system stores this into the database.

Use Case Name: Login
Actor: Manager
Description: This use case begins when the manager loads into the web app. They then proceed to enter their selected username and password and the system verifies the login is correct and logs the manager into the app.

Use Case Name: Reset Password
Actor: Manager
Description: This use case begins when the manager loads into the web app. They then select the forgot password button on the login screen. They then enter their email and they can enter the new password they wish to use. The system then updates the managers information in the database.

Use Case Name: Logout
Actor: Manager

Description: This use case begins when the manager logs into the web app. They then select the logout button on the screen. The app then logs the user out of the site.

Use Case Name: Build Team
Actor: Manager
Description: This use case begins when the manager logs into the web app. They then can enter a league code or create a league and it will bring them to the create a team screen, here they can select their team based off of the custom rules requirements that the league has.

Use Case Name: AI Team Selection
Actor: Manager, ML model
Description: This use case begins when the manager views their team, they can then click the recommendations button. This calls the ML model which analyses their starting 11 and bench players and recommends the best squad possible for the upcoming week.

Use Case Name: Create League
Actor: Manager
Description: This use case begins when the manager loads into the web app. They then select the create league button. This then brings them to the league setup screen where they can select custom league rules and get given a unique league code to give to friends so that they can join

Use Case Name: Join League
Actor: Manager
Description: This use case begins when the manager loads into the web app. They then select the join league button, this is where they will be prompted to enter the league code to join their desired league.

Use Case Name: Player Comparison
Actor: Manager, Machine Learning Model
Description: This use case begins when the manager selects the player comparison button on their squad. They then enter both players that they want to select. These two selections are given to the model which determines which player is a better fit for the upcoming game week and tells the user who to select.

Use Case Name: AI Transfer Recommendations
Actor: Manager, Machine Learning Model
Description: This use case begins when the manager loads into their current team. They can then select a player and click the transfer recommendation button. This is passed to the ML model so that it can determine the best player to transfer in for the upcoming game week.

Detailed Use Cases

Use Case Name: Register
Actors: Manager
Brief Description: This happens when a manager loads the app up and selects register
Main Success Scenario: 1. Manager selects register 2. Manager enters username, email & password 3. System validates the inputs 4. System stores them in the database and logs them in with the new information Alternatives: 3a. The details are in an invalid format 1. The manager is brought back to the register page 2. There is an error message displayed detailing the correct format 3b. The details are already in use 1. The manager is brought back to the register page 2. There is an error message displayed detailing that these details are already in use.

Use Case Name: Login
Actors: Manager
Brief Description: This happens when a manager loads the app up and selects login
Main Success Scenario: 1. Manager selects login 2. Manager enters username & password 3. System validates the inputs 4. System logs them in and brings them to the home page Alternatives: 3a. The details are incorrect. 1. The manager is brought back to the login page 2. There is an error message displayed saying that the details entered weren't correct

Use Case Name: Reset Password
Actors: Manager
Brief Description: This happens when a manager wishes to reset their password
Main Success Scenario: 1. Manager selects reset password 2. Manager enters their email 3. System requests a new password 4. Manager enters a new password 5. System updates password and logs the user in to the home page Alternatives: 3a. The same password is entered again. 1. The manager is brought back to the reset password page 2. There is an error message displayed saying the same password cant be used again

Use Case Name: Logout
Actors: Manager
Brief Description: This happens when a manager loads the app up and selects logout
Main Success Scenario: 1. Manager loads their home screen 2. Manager selects logout 3. Manager is successfully returned to the main home page

Use Case Name: Build Team
Actors: Manager
Brief Description: This happens when a manager loads the app up and wants to create their team
Main Success Scenario: 1. Manager logs into the app 2. Manager enters their league code or creates a league 3. Manager is then brought to the team selection screen 4. Manager can select their team based off of the custom league rules Alternatives: 2a. Manager enters the wrong league code 1. The manager is brought back to the enter code screen 2. There is an error message telling the manager the code is incorrect

Use Case Name: AI Team Selection
Actors: Manager, Machine Learning Model
Brief Description: This happens when a manager loads the app up and wants to change their team
Main Success Scenario: 1. Manager logs into the app 2. Manager builds their desired team 3. Manager is then brought to the team selection screen 4. They can select the AI Selection button and the AI will recommend who to start and bench from their team for the upcoming week

Use Case Name: Create League
Actors: Manager
Brief Description: This happens when a manager loads the app up and wants to create their own league
Main Success Scenario: 1. Manager logs into the app 2. Manager selects the create league button 3. Manager is then prompted to select their league rules 4. Manager is then given their league code 5. Manager is brought to their team selection screen

Use Case Name: Join League
Actors: Manager
Brief Description: This happens when a manager loads the app up and wants to join a league using a league code
Main Success Scenario: 1. Manager logs into the app 2. Manager selects the join league button 3. Manager is then prompted to enter their code 4. Manager then enters their code 5. Manager is brought to their team selection screen for this league

Use Case Name: Player Comparison
Actors: Manager, Machine Learning Model
Brief Description: This happens when a manager views their team and wants to compare two players
Main Success Scenario: 1. Manager logs into the app 2. Manager views their team 3. Manager then selects a player in their team 4. Manager then selects the player comparison button where they are prompted to select another player 5. The machine learning model then compares them showing detailed stats for both players showing the better option

Use Case Name: AI Transfer Recommendations
Actors: Manager, Machine Learning Model
Brief Description: This happens when a manager loads the app up and wants to edit their team
Main Success Scenario: 1. Manager logs into the app 2. Manager selects on their team 3. Manager selects the transfer recommendations button 4. The AI analyses the team and displays the recommendations based off of the league rules, fixtures and players

Model

All of the use cases display the signs of the FURPS+ model which I will highlight more below

Functionality

- Managers must be able to create league
- Managers must be able to create their teams
- Managers must be able to create an account
- Managers must be able to login and logout to and from their account
- Managers must be able to get AI based recommendations for transfers and teams
- Managers must be able to use a unique code to join a league

Usability

- There must be a simple to use frontend with good UX and UI design
- The web app must load quickly to ensure good UX
- The design must be optimised cater to mobile and pc sized screens
- The machine learning models must have fast response times
- All features must integrate seamlessly with each other to ensure good UX

Reliability

- Ensure the site is up and running as much as possible especially at weekends when games are on
- Real-time synchronization of league data, player statistics, and points updates

Performance

- AI recommendations processed in real-time without causing delays in team management
- The leagues working the same for small groups and large groups with thousands of users in them
- Fast loading times for more heavy duty features such as the ones that involve the machine learning models

Supportability

- A nicely abstracted codebase to allow for scaling of the project in the future
- Clear tests for the codebase to decrease the likelihood of bugs
- Monitoring tools to track performance of the site
- Well commented code for future developers to be able join in on the codebase
- Good error logging in the codebase for ease of bug finding

Progress So Far

Data Scraping

I have built a data scraper to obtain my data from the FBREF website, this website is one of the leading providers for in depth statistics for football and especially the Premier League which is the league that I will be focusing on for my application. This took a lot of time to get working as I had a lot of trouble extracting certain data that I need to build my data. Eventually after searching through the HTML file I discovered that the table I need was commented out and this was why my code couldn't find the table when I initially was searching for it. I got around this by making a small script that will search the html file and delete the comment line and this allows the table to be picked up correctly.

Data Cleansing

After I figured out the data scraper and had my data in a CSV file, my first step was to cleanse the data. The model that I wanted to build for this alpha was whether or not a player would score over 15 goals in a season or not so my first step was to focus on the attackers. I did this by searching for the players who were forwards in their position column and moving these to a new CSV just for attackers. I then realised I didn't have a predictive value for this as it is current season data so I divided the players goals by games played and multiplied this by 38 to get a prediction for the amount of goals every player would score for the season. This isn't a perfect solution but I cross checked with previous seasons data and the amount of players hitting the 15 goals mark was very similar to my prediction so I decided to proceed with it.

Model Building

For my alpha model I decided to use a decision tree as this model would be providing a binary result as the player either will or won't hit the 15 goal a season mark. This ended up working well as the data was numerical which made it easier to train the model on. The three statistics that I ended up using to train this model were Goals per game, Expected goals per game and non-penalty expected goals per game. I chose these as using per game statistics gives a better idea of how a player is performing overall and it gives a bit more information than just goals as a player could score 5 in one game which can skew predictive stats a bit. This model that I built has a 95% accuracy and can predict whether or not a player will score over or under 15 goals a season based off of the statistics that I listed above.