

# Requirements Document: Fantasy Football

## Project Title

EPL Football Manager App

## 1. General Understanding

Purpose: The Football Manager App will assist users in managing a fantasy football team by integrating real-time player data from the English Premier League (EPL) API. It will enable users to select players, track performances, and make strategic decisions to optimise their team’s performance.

Problem Statement: Managing a fantasy football team requires timely and accurate player data. Manual data collection can be inefficient and error-prone. This app automates data retrieval, analysis, and team management using the EPL API.

## Goals:

- Build a Python-based application with a graphical user interface (GUI) using customTKinter.

- Integrate data retrieval from the EPL API for real-time player statistics.

- Provide user-friendly tools to analyze and manage a fantasy football team.

## 2. Stakeholder Requirements

## Stakeholders:

- Fantasy football players: Require an app to select players, monitor performances, and make strategic changes.

- General football fans: May use the app for tracking live statistics and standings.

Stakeholder Expectations:

- Reliable integration with the EPL API for real-time data.

- Intuitive interface for team management.

- Comprehensive analytics for decision-making.

Stakeholder Validation: Stakeholders will review the app’s prototype and provide feedback on the interface and functionality.

## 3. Functional Requirements

### Team Management:

- Users can create a fantasy football team by selecting players from the EPL API.

- Users can view and edit their team lineup.

## Player Data and Analytics:

- Retrieve and display real-time player statistics (e.g., goals, assists, clean sheets).

- Provide analytics on player performance trends and value-for-money ratios.

## Match Tracking:

- Display upcoming fixtures and results for selected players.

- Provide notifications for key events (e.g., goals, injuries, suspensions).

## League Standings:

- Display current EPL standings and team statistics retrieved from the API.

## Save and Load User Data:

- Allow users to save their fantasy team configuration locally.

- Load saved team data on app startup.

## 4. Non-Functional Requirements

Performance: The app should retrieve and display data from the EPL API within 2 seconds.

Usability: Ensure the interface is simple and intuitive, even for users unfamiliar with fantasy football.

Reliability: Handle API errors gracefully, providing fallback messages or retry options.

Security: Ensure user data, including team configurations, is securely stored.

Portability: The app must run on Windows, macOS, and Linux.

## 5. Data Requirements

Player Data from EPL API: Fields: Player name, position, team, goals, assists, yellow/red cards, clean sheets, current value, form.

Team Data from EPL API: Fields: Team name, position in the table, points, goal difference.

Local User Data: Fields: User-selected players, team name, saved configuration.

## 6. System Requirements

### Hardware Requirements:

- Minimum: 2GB RAM, 1GHz CPU, 500MB storage.

- Recommended: 4GB RAM, 2GHz CPU, 1GB storage.

### Software Requirements:

- Python 3.10 or later.

### - Libraries:

- customTKinter for the GUI.

- Requests for API calls.

- JSON for data handling.

- Matplotlib for visualizing player performance trends.

## 7. Constraints

Time Constraints: The app must be completed within 10 weeks.

Budget Constraints: Development must use open-source tools and free API tiers.

Technical Limitations: Limited to the data fields and refresh rates provided by the EPL API.

## 8. Validation and Review

### Validation:

- Stakeholders will test the app’s functionality against requirements during weekly reviews.

- Functional and non-functional requirements will be verified using unit and integration tests.

### Review:

- Requirements will be revisited every two weeks to ensure alignment with the project’s goals and stakeholders' feedback.

## Final Check

- Requirements are documented in a clear and structured format.

- All functional and non-functional requirements are testable.

- Stakeholders have approved the final requirements document.