Software Requirements Specification

for

FIFA World Cup Analytics

Using API

Prepared by

Group Name: 285 and team

Varun Utukuri	**SE22UCSE285**	**se22ucse285@mahindrauniversity.edu.in**
Saranya Reddy	**SE22UCSE160**	**se22ucse160@mahindrauniversity.edu.in**
Meghana Kaldindi	**SE22UCSE166**	**se22ucse166@mahindrauniversity.edu.in**
Kaushik Reddy	**SE22UCSE041**	**se22ucse041@mahindrauniversity.edu.in**
Rishi Raju	**SE22UCSE220**	**se22ucse220@mahindrauniversity.edu.in**
Rithvik Maragoni	**SE22UCSE224**	**se22ucse224@mahindrauniversity.edu.in**
Suchit Kunathi	**SE22UCSE145**	**se22ucse145@mahindrauniversity.edu.in**
Saketh Boya	**SE22UCSE238**	**se22ucse238@mahindrauniversity.edu.in**

Instructor: VIJAY RAO

Course: SOFTWARE ENGINEERING

Teaching Assistant: MURALI KRISHNA

Date: 10-03-25

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being	00/00/00

1. Introduction

The **FIFA World Cup Analysis** project aims to provide insightful analytics and predictions based on historical and real-time match data.

1.1. Document Purpose

This document provides a comprehensive Software Requirements Specification (SRS) for the FIFA World Cup Analysis mobile application. The application will leverage publicly available APIs to fetch FIFA World Cup data and present interactive visual analytics for users. This document outlines the functional, non-functional, and design requirements necessary for development.

1.2. Product Scope

The FIFA World Cup Analysis application aims to provide real-time and historical data, match insights, team comparisons, and player performance evaluations. It will feature:

- API integration for real-time and historical FIFA World Cup data.
- Data visualization using interactive graphs and charts.
- A user-friendly UI/UX for seamless interaction.

1.3. Intended Audience and Document Overview

This document is intended for:

- Developers (Frontend & Backend)
- Project Managers
- Data Analysts
- Testers
- End Users
- Professors

The document includes detailed functional and non-functional requirements, constraints, use case models, and design guidelines.

1.4. Definitions, Acronyms and Abbreviations

- API: Application Programming Interface
- UI/UX: User Interface/User Experience
- SRS: Software Requirements Specification
- FIFA: Fédération Internationale de Football Association

1.5. Document Conventions

- Font: Arial, size 11.
- Section titles are bold and numbered.
- Diagrams follow UML notation where applicable.

1.6. References and Acknowledgments

- football-data.org for API
- Existing open source projects

2. Overall Description

2.1. Product Overview

The FIFA World Cup Analysis app will fetch and process FIFA World Cup data, displaying insights using interactive visualization techniques. The system consists of:

- A backend for API integration and data processing.
- A frontend for user interaction and data presentation.
- A database for storing processed data.

2.2. Product Functionality-

- Fetch real-time and historical FIFA World Cup data via API.
- Process and structure data for meaningful insights.
- Display match statistics using interactive charts and graphs.
- Provide team and player performance comparisons.
- Ensure smooth user interaction via an intuitive UI.

2.3. Design and Implementation Constraints

- Use of publicly available FIFA APIs.
- Compliance with API rate limits and data policies.
- Backend developed using Python and FastAPI.
- Frontend developed using React Native.
- Data storage using PostgreSQL.
- Deployment on AWS Cloud.

2.4. Assumptions and Dependencies

- Availability of reliable FIFA APIs.

- Public APIs providing updated FIFA data.
- Scalability considerations based on AWS EC2 configurations.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

- A responsive UI built using React Native that allows the user to interact with the website and view filtered analytics based on their choice.
- Navigation via menus and interactive dashboards accompanied by interactive carrousels and accordions.
- Real time graphs and pie charts showing player and team data. View winner predictions and team performance in various aspects.

3.1.2. Hardware Interfaces

- Mobile devices (Android & iOS).
- Internet connectivity for real-time data fetching.

3.1.3. Software Interfaces

- **FIFA World Cup API** and **Opta Sports API** Fetches real-time and historical match statistics, team performance, and player data.
- **Football-Data.org** – Provides fixtures, results, and league standings.

3.2. Functional Requirements

3.2.1 User Registration and Authentication

- The system must allow users to sign up using email, social media, or OAuth providers.
- Users should be able to log in securely using credentials.

3.2.2 Fetching and Storing Match Data

- The system should fetch match data from external APIs such as FIFA API or Football-Data.org.
- Match data should include **team details, player statistics, match scores, and standings.**
- Data should be updated in real-time and stored in a database.

3.2.3. Viewing Match Statistics

- Users should be able to view detailed match statistics such as:
  - Goals scored, possession percentage, shots on target, passes completed.
  - Player-specific data like goals, assists, and performance ratings.
- Statistics should be displayed using **interactive charts and tables.**

3.2.4. Filtering and Sorting

- Users should be able to filter statistics based on:
  - Teams, players, tournaments, and match dates.
- Sorting should be possible by key metrics (e.g., top goal scorers, most assists).

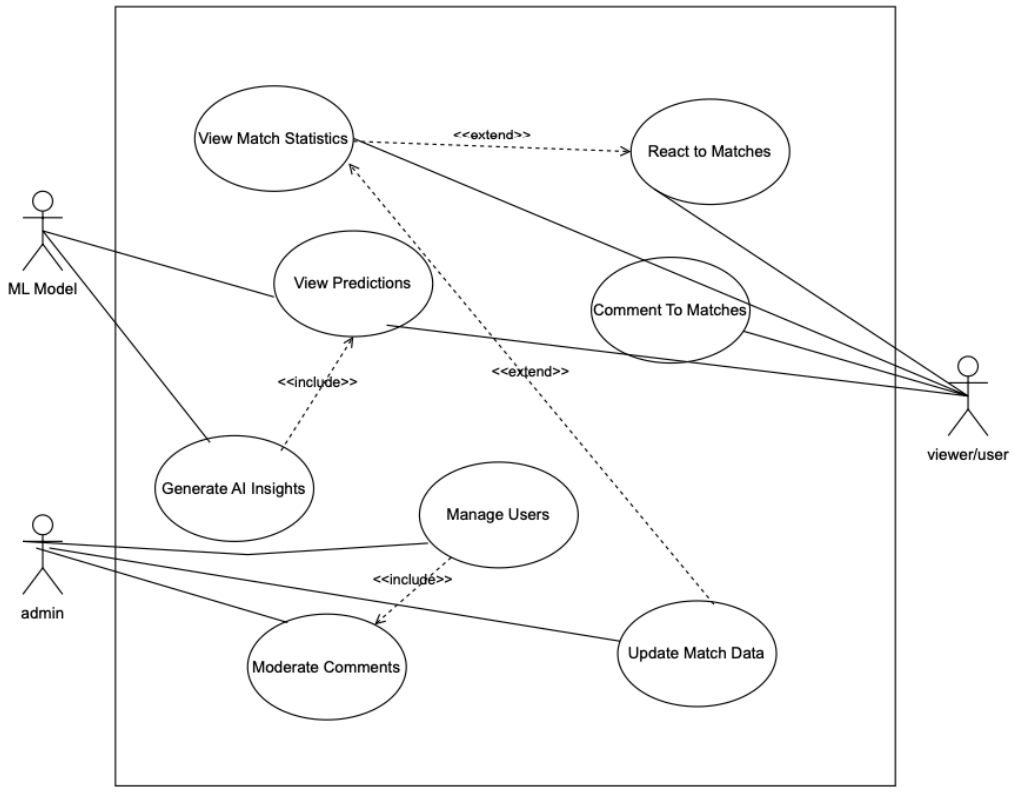
3.2.5. Filtering and Sorting

- Users should be able to filter statistics based on:
  - Teams, players, tournaments, and match dates.
- Sorting should be possible by key metrics (e.g., top goal scorers, most assists).

3.2.6 Generate AI-Based Insights

- The ML Model should analyze historical match data and predict match outcomes.
- Insights should include:
  - Win probability for each team.
  - Key players likely to influence the match.
  - Expected goals based on team form.
- Predictions should be visually represented (e.g., probability charts).

3.3.Use Case Model



3.1.4. Use Case Specifications

Author – Varun Utukuri

Purpose - To allow users to view detailed statistics of matches, including team performance, player stats, and match results.

Requirements Traceability –

- The system should retrieve match statistics from an API.
- The user should be able to interact with statistical visualizations.

Priority - High

Preconditions -

- The system must be connected to the internet.
- The user must be logged in (if required).
- The database must contain relevant match data.

Post conditions -

- The user successfully views the match statistics.
- The system updates the statistics if new data is available.

Actors –

- Viewer/User

Extends –

- React to Matches (U2)
- Comment to Matches (U3)

Flow of Events

1**. Basic Flow -**

- The user navigates to the "Match Statistics" section.
- The system fetches match statistics from the API.
- The statistics are displayed using interactive graphs and tables.

2. **Alternative Flow -**

- If API data is not available, cached data is displayed.
- If the user selects a specific team, only their match statistics are shown.

3. **Exceptions -**

- If the API is down, an error message is displayed.
- If the user has no internet connection, a "No Data Available" message appears.

Notes/Issues -

- Ensure efficient data fetching and caching to reduce API calls.
- Provide an option for users to filter statistics by date or team.

4. Other Non-functional Requirements

4.1. Performance Requirements

- The system should support **concurrent users** without significant performance degradation.
- Data retrieval from external APIs (e.g., FIFA API) should should happen almost instantly.
- Page load times should not take long, even under high traffic.

4.2. Scalability Requirements

- The system should be **horizontally scalable**, allowing new servers to be added dynamically to handle increased user loads.
- The application should be deployable on **cloud platforms** like AWS
- A **load balancing mechanism** should be implemented to distribute traffic efficiently.

4.3. Availability and Reliability

- The system should have **high uptime**, ensuring uninterrupted access to match data and analytics.
- A **failover mechanism** should be in place to switch to backup servers in case of system failure.
- The system should perform automatic backupsto prevent data loss.

4.4. Security Requirements

- **User authentication** should use **OAuth 2.0, JWT, or Multi-Factor Authentication (MFA)**.
- User passwords should be **encrypted using SHA-256 or bcrypt hashing**.

4.5. Usability and User Experience (UX)

- The system should have **a responsive design** that works across desktops, tablets, and smartphones.
- **Dark mode** should be available for enhanced user experience.

4.6. Interoperability

- The system should be able to integrate with **third-party data providers** like OpenFootball API, FIFA API, and Football-Data.org.
- The system should support exporting match data in **CSV, JSON, and XML formats**.

Appendix A – Data Dictionary

Variable	Description	Type
Match ID	Unique identifier for each match	Integer
Player ID	Unique identifier for each match	Integer
Team ID	Unique identifier for each match	Integer
Match Statistics	Data related to match performance	JSON

Appendix B - Group Log

- **Week 1-2:** Initial meetings & planning.
- **Week 2-5:** API integration & data structuring.
- **Week 5-8:** Software Requirements Specification.
- **Week 8-12:** Frontend & UI/UX development.
- **Week 12-14:** Testing and final deployment