Software Design Specifications

FIFA World Cup Analytics v1.0

Prepared by:

Varun Utukuri

Project Lead

Document Information

Title: FIFA World Cup Analytics Design Document

Project Manager: Varun Utukuri

Document Version No: 1.0

Document Version Date: 10-03-25

Prepared By: Group 285

Preparation Date: 10-03-25

1. **INTRODUCTION**

1.1 PURPOSE

The purpose of this Software Design Specification is to define the software design architecture for the FIFA World Cup Analytics mobile application. This document serves developers, testers, project managers, and stakeholders to understand how the system will be built based on the SRS.

1.2 SCOPE

This document applies to the design of a mobile application that provides real-time and historical analytics of FIFA World Cup matches using external APIs. It includes data visualizations, AI predictions, and user-interactive features.

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

API - Application Programming Interface UI/UX - User Interface/User Experience JWT - JSON Web Token OAuth - Open Authorization

1.4 REFERENCES

Football-data.org API Documentation SRS Document dated 10-03-25

2. **USE CASE VIEW**

2.1 USE CASE Name: View Match Statistics Brief Description: Allows users to view interactive statistics including player data, team performance, and match results. Usage Steps:

- User logs in
- Navigates to match statistics section
- Filters data by team/player/date
- Views statistics via interactive graphs and tables

3. **DESIGN OVERVIEW**

3.1 DESIGN GOALS AND CONSTRAINTS

- Ensure real-time data fetching
- Use FastAPI for backend, React Native for frontend

- Use PostgreSQL for persistent storage
- Deploy on AWS Cloud infrastructure
- Ensure compliance with API rate limits

3.2 DESIGN ASSUMPTIONS

- Reliable and stable internet connection for users
- Continued availability of public FIFA APIs
- AWS cloud infrastructure is properly provisioned

3.3 SIGNIFICANT DESIGN PACKAGES

- Frontend Package: React Native UI
- Backend Package: FastAPI Services
- Database Package: PostgreSQL Schema
- ML Module: AI-based prediction engine

3.4 DEPENDENT EXTERNAL INTERFACES

External Module	Interface Name	Description
Football-Data.org	MatchesAPI	Used to fetch match stats
FIFA API	LiveStatsAPI	Used to fetch live player & team performance

3.5 IMPLEMENTED APPLICATION EXTERNAL INTERFACES

Interface Name	Module Implementing	Description
MatchesAPI	Backend Service	Calls to fetch and process match data
UserAuthAPI	Authentication Module	Handles OAuth and JWT-based login

4. LOGICAL VIEW

4.1 DESIGN MODEL

- Class: MatchService
    - Attributes: matchId, team1, team2, stats
    - Methods: fetchData(), processData()
- Class: User
    - Attributes: userId, email, preferences
    - Methods: login(), register(), filterStats()
- Class: MLModel
    - Attributes: modelParams, prediction
    - Methods: predictOutcome(), visualize()

4.2 USE CASE REALIZATION

Use Case: View Match Statistics

- Sequence:
    - User logs in -> Backend authenticates
    - Frontend requests match stats -> Backend fetches from API -> Backend sends data to Frontend -> UI displays graphs

5. DATA VIEW

5.1 DOMAIN MODEL

- Entity: Match
    - Fields: ID, teams, score, stats, date
- Entity: Player
    - Fields: ID, name, team, performance

5.2 DATA MODEL (PERSISTENT DATA VIEW)

5.2.1 DATA DICTIONARY

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

6. **EXCEPTION HANDLING**

- API Down: Show cached data or error message
- No Internet: Display "No Data Available"
- Invalid Input: Alert user and validate forms

7. **CONFIGURABLE PARAMETERS** | Configuration Parameter Name | Definition and Usage | Dynamic? | |----------------------------|-----------------------|----------| | api_refresh_rate | Controls API data fetch interval | Yes | | prediction_threshold | ML model prediction confidence level | Yes |

8. **QUALITY OF SERVICE**

8.1 AVAILABILITY

- Deploy using AWS with auto-scaling and failover mechanisms
- Maintain 99.9% uptime with backups

8.2 SECURITY AND AUTHORIZATION

- OAuth 2.0 and JWT-based authentication
- Encrypted passwords using SHA-256
- MFA supported

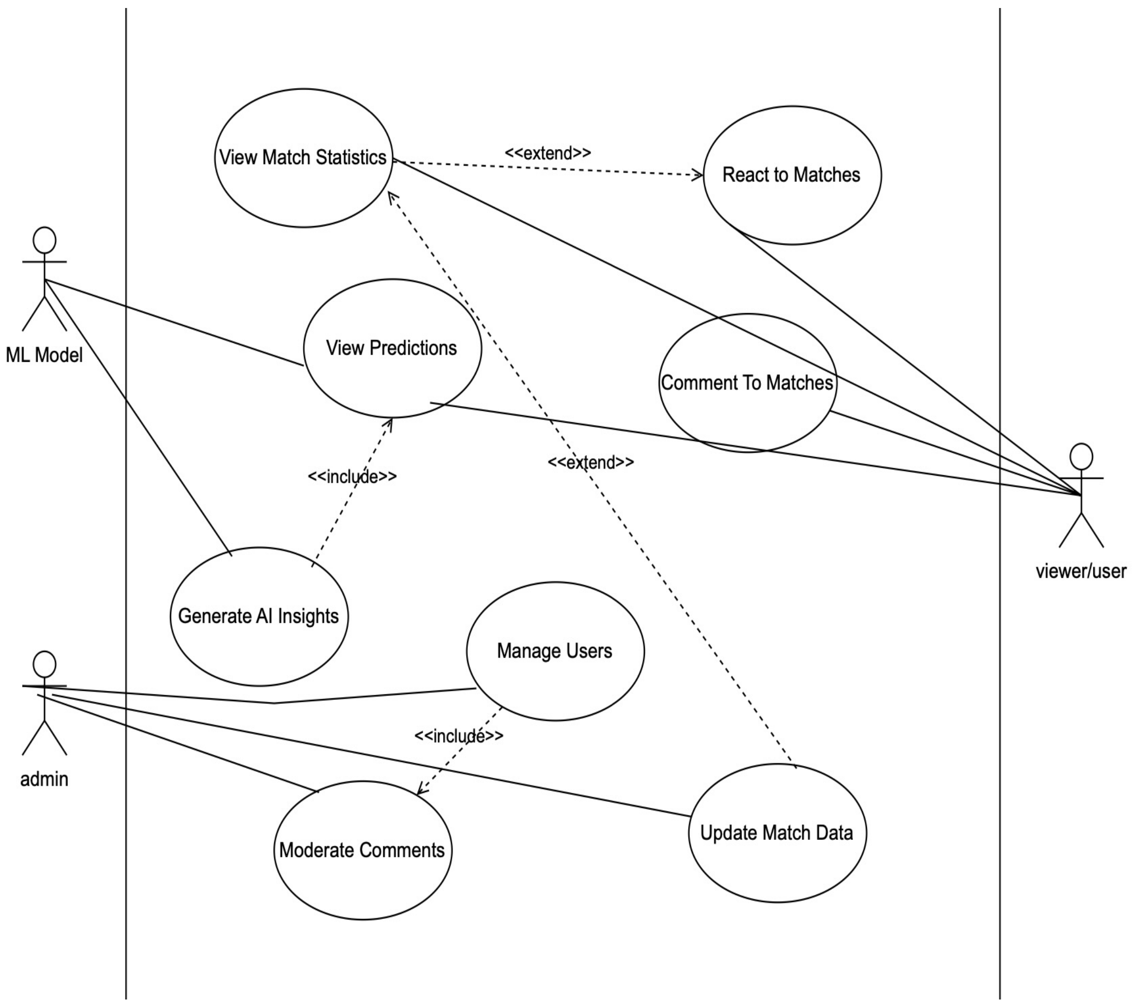
8.3 LOAD AND PERFORMANCE IMPLICATIONS

- Designed to handle thousands of concurrent users
- Uses caching and rate-limiting for API calls

8.4 MONITORING AND CONTROL

- CloudWatch integration for monitoring
- Health check endpoints
- Log aggregation and alerting

USE CASE DIAGRAM:


View Match Statistics
<<extend>>
React to Matches
ML Model
View Predictions
Comment To Matches
<<include>>
<<extend>>
viewer/user
Generate AI Insights
Manage Users
admin
<<include>>
Update Match Data
Moderate Comments