Technical Report - **Product specification**

Game Sense

Course:	IES - Introdução à Engenharia de Software
Date:	Aveiro, 07/10/2024
Students:	113968: Guilherme Rosa 113920: João Roldão 113765: Rui Machado 114588: Henrique Teixeira
Project abstract:	GameSense is a football analytics app that provides real time analysis of simulated football games. It is also capable of simulating games based on past data.

Table of contents:

1 Introduction

2 Product concept
- Vision statement
- Personas
- Main scenarios

3 Architecture notebook
- Key requirements and constrains
- Architetural view
- Module interactions

4 Information perspetive

5 References and resources

Fig. 2: Base Architecture

1 Introduction

GameSense is a football analytics app designed for enthusiasts and professionals seeking deeper insights into the beautiful game. The app provides real-time analysis of team, players performances and statistics, allowing you to easily compare teams and players. Additionally, it can predict game outcomes based on real past statistics.

2 Product concept

Vision statement

Concept

Our system will be used for real-time analysis of simulated football games. It captures detailed statistics of the game, including team performance, individual player metrics, and additional stats. The system generates dynamic insights, creates leaderboards and rankings based on data, allowing users to compare teams and players effectively.

Inspiration

While there are similar products like *Flash Score* and *Zero Zero* that already provide detailed statistics and diverse graphs for every team, player, and managers across all recorded seasons, our system is different. It offers simulation and prediction capabilities that set it apart. Users can simulate matches using historical data, analyze team form, past performances and player's stats to predict outcomes and odds. This helps users explore different match scenarios and prepare more effectively for upcoming games.

Motivation

The concept was developed because we wanted a tool that could help us with our football bets by compiling historical data and identifying the best betting opportunities. We aimed to create a solution that not only provides detailed statistics, but also helps users to simulate and predict match outcomes, enhancing their betting strategies.

Fig. 2: Base Architecture

Use Case Diagram

Fig. 1: Use Cases Diagram

Personas and Scenarios

A **Persona** is a fictional character created to represent potential users and their interactions with a system. It embodies a specific set of traits and motivations, helping humanize and define the context of how an app will be used. Personas help illustrate how different types of users might engage with the system, making future use cases more relatable and easier to understand.

A **Scenario** is a narrative that outlines the interactions between personas (or other actors) and the system. It describes the flow of events in a particular use case, illustrating either how the system is currently used or detailing the required functionality for a new or updated system. Scenarios provide insight into the use case, offering a clearer understanding of the challenges and requirements involved.

Fig. 2: Base Architecture

Persona #1: Fan

Name: António Geirinhas

Age: 34

Location: Lisbon, Portugal

Occupation: Accountant

Description: António is a passionate Sporting CP fan who loves his team. He has loved the sport since he was a child and rarely misses a match of his favorite club, whether in the stadium or watching from home. He actively checks out football statistics and match analysis.

Motivation:

- Wants to stay updated with real-time stats during matches to feel more connected with the game and to better understand how the game is going.
- Enjoys predicting outcomes of matches for fun and sometimes he places bets.
- Appreciates detailed player comparisons and tactical breakdowns with statistics to back up his opinions during arguments with friends.

Persona #2: Coach

Name: Jorge Vitória

Age: 48

Location: Porto, Portugal

Occupation: Head Coach of Leixões Sport Club

Description: Jorge is an experienced football coach who has managed teams in various levels of football, from youth teams to professional clubs. He has a masters degree in statistics, showing how focused he is in extracting maximum performance from his players through data analysis. Jorge stays ahead of other teams by analyzing opponents' tactics and adjusting his strategies accordingly.

Motivation:

- Wants real-time game statistics to make informed decisions during matches, especially for substitutions and tactical changes.
- Enjoys a good post-match analysis to identify areas for improvement and prepare for upcoming opponents.

Fig. 2: Base Architecture

Persona #3: Player

Name: Ricardo Monteiro

Age: 26

Position: Central Midfielder

Club: *Sporting Clube de Braga*

Description: Ricardo is a talented midfielder who prides himself on his passing accuracy and vision on the pitch. He is highly disciplined and a perfectionist in all things in life, making him look for ways to improve his game all the time, whether through personal data analysis or feedback from coaches.

Motivation:

- Wants data insights to track his own performances and benchmark against his past games and league averages.
- Enjoys seeing tactical breakdowns to understand how his positioning impacts team dynamics.

Persona #4: Scout

Name: Marta Lopes

Age: 40

Location: Birmingham, England

Occupation: Head of Scouting of *Wolverhampton Wanderers Football Club*

Description: Marta is an experienced scout who travels across Europe to evaluate up-and-coming talent for her club. She relies heavily on data analysis to pick potential recruits before seeing them live in action. Marta is always looking for ways to gain a competitive edge in identifying talent, including emerging trends in performance analytics.

Motivation:

- Needs performance simulations and historical data to assess if a player is what she needs to fit her team's tactics.
- Analyzes young players, how their stats have been developing, their injury record and which tactics they have played under.
- Looks to back her insights with stats when presenting to the club's manager and board a newly recommended player for acquisition.

Fig. 2: Base Architecture

Scenario #1: António Geirinhas (Fan)

Keeping up with a Game

António is excited about an upcoming match between Sporting CP and their rivals but, unfortunately, he can't watch it. During the game, he opens the GameSense app to keep informed with in-game events (goals, yellow cards, substitutions, etc.) and check for statistics to get a better sense for how the game his going for Sporting, the team he is supporting.

Scenario #2: Jorge Vitória (Coach)

Preparing the tactical plan for a game

Preparing for a match, Jorge uses GameSense to analyze his opponent's recent performances. He simulates different matches to identify potential weaknesses in the opposing team's tactics and vulnerabilities within his own squad. Then, he can help his players exploit the opponent's weaknesses and strengthen their own skills for the game.

Scenario #3: Ricardo Monteiro (Player)

Review player performance

Ricardo wants to improve as a player, and for that, he logs into GameSense to review his personal statistics from his recent matches. Using this information Ricardo knows what to train to improve his game. Ricardo can also check for opponents statistics and try to find out any possible weaknesses that he can exploit.

Scenario #4: Marta Lopes (Scout)

Seeking a new talent

Marta is looking out for a promising player to her team's squad. Using GameSense, she filters players on key performance metrics like passing accuracy, expected goals and adaptability to different tactics. She discovers a young talent whose stats have been consistently improving.

Fig. 2: Base Architecture

Product requirements (User stories)

Live Statistics (Epic #1)

- **User Story #3**
  As António
  I want to check the game result in live
  So that I can follow a game if I am not able to watch it.
- **User Story #4**
  As Jorge
  I want to check my teams game stats in live
  So that I can adapt my tactics in a game to give my team a better chance of winning.
- **User Story #10**
  As Jorge
  I want to see what players of other teams are injured
  So that I can prepare matches in a better way.

Historical Statistics (Epic #2)

- **User Story #1**
  As Ricardo
  I want to check my overall personal statistics
  So that I can improve as a professional.
- **User Story #2**
  As António
  I want track my favorite team's standing in the championship
  So that I stay updated on their progress.
- **User Story #7**
  As Marta
  I want to see every game and player statistics in detail
  So that I can write my reports.
- **User Story #8**
  As Marta
  I want find the player with the most goals
  So that I can convince the team's board to contract him.
- **User Story #9**
  As António
  I want to see the result's history of my team's next opponent
  So that I know what to expect.

Fig. 2: Base Architecture

- **User Story #11**
  As Ricardo
  I want recommendations for specific skills or areas to focus on
  So that I can improve and make a bigger impact on the pitch.
- **User Story #12**
  As Marta
  I want to evaluate the injury history and risk factors of a player
  So that I can make safer recommendations for player signings.

Game Prediction (Epic #3)

- **User Story #6**
  As Jorge
  I want to simulate a game against an opponent
  So that I can see what is a probable outcome and try to beat the odds using statistical data of the simulated game

Social Media Interaction (Epic #4)

- **User Story #5**
  As António
  I want to easily share key stats and insights from games to my social media page
  So that i can have informed discussions with my followers

Fig. 2: Base Architecture

3 Architecture notebook

Key requirements and constraints

Non-functional requirements and constraints

As a real-time football analysis application the primary focus of GameSense is to enable a system that has a low latency data flow for real time match events and a persistent and scalable database that has historical and statistical data. With this, the user can interact with live match updates and see players and teams stats. Another important requirement of our application is to have a user-friendly interface that prioritizes ease of use. This is to allow the user to access any part of the application without much trouble.

Performance and Responsiveness - As a real-time analysis application it is important to have low latency, for example, when a goal is scored the user should receive that information in under 3 seconds.

Scalability - GameSense needs to be able to keep working even if the traffic of users is bigger than usual. The application needs to be able to hold up to 500 to 1000 users and not have noticeable performance degradation.

Data Consistency and Integrity - Being a football analysis app we need to make sure that the data and statistics we are being shown to the user are updated and accurate.

Functional requirements and constraints

The main functionalities of GameSense revolve around providing historical and statistical data of football teams, players and matches, following a match live and game simulations. To ensure user engagement, the app will have push integration and third-party integrations so the users can share and comment about a result or statistical information provided by our application.

Push Notifications System - This push notification system will be able to alert the user of events that are happening in the match he is interested in, or where a team/player that the user is following is a part of.

Game Simulation - The game simulation is going to be fully deterministic and base itself only on historical and statistical data from past performances (ex.: team form, home/away performances, player and team stats).

Fig. 2: Base Architecture

Architectural view

For an easier understanding of how GameSense is going to work we divided the architecture into 4 parts, those being frontend, backend, data sources and message queues.

Data Sources: The GameSense app will use a Python-based API that simulates live football matches. This API will generate match events and statistics every minute, providing real-time data that feeds directly into the system. For a better Game Generation experience the API will get from the Backend information that will help keep the data consistent and plausible.

Message Queues: The Message Queue will be handled by Kafka, which will enable asynchronous communication between the Data Sources and the Backend.

Backend: The backend is going to be responsible for data processing, storage and external integrations. The GameSense app will have 3 databases, a Redis database that will serve as cache for live games, and 2 MySQL databases, one that will handle the persistent data, such as historical match data and player statistics, for example; and the other one is gonna have the User's, their preferences and favourites and Simulated Games. We will also have an integration with the OpenAI API and a python library, MPLSoccer. The OpenAI API will provide AI driven insights of the statistical data we provide. The MPLSoccer library will help us generate football-specific visualizations, like heatmaps.

Frontend: Our main UI is going to use React web app, along with the VITE PWA plugin. As for the communications with the Backend we will use endpoints to communicate with the Redis Cache and show the user the information of the live game and for the other features it will be via a REST APIs.

Base Architecture

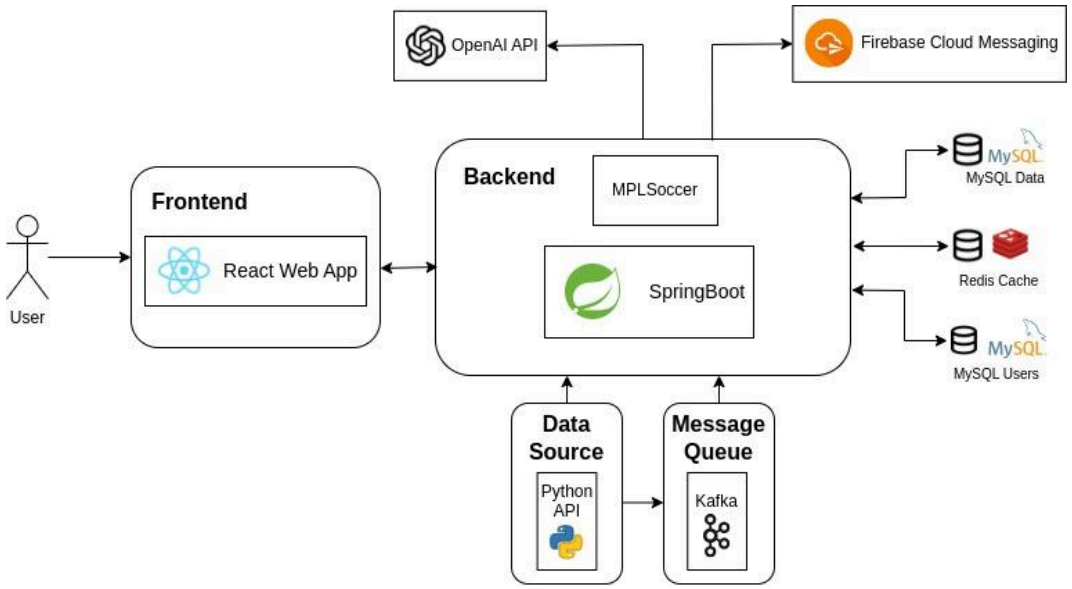

Fig. 2: Base Architecture

Architecture with Microservices

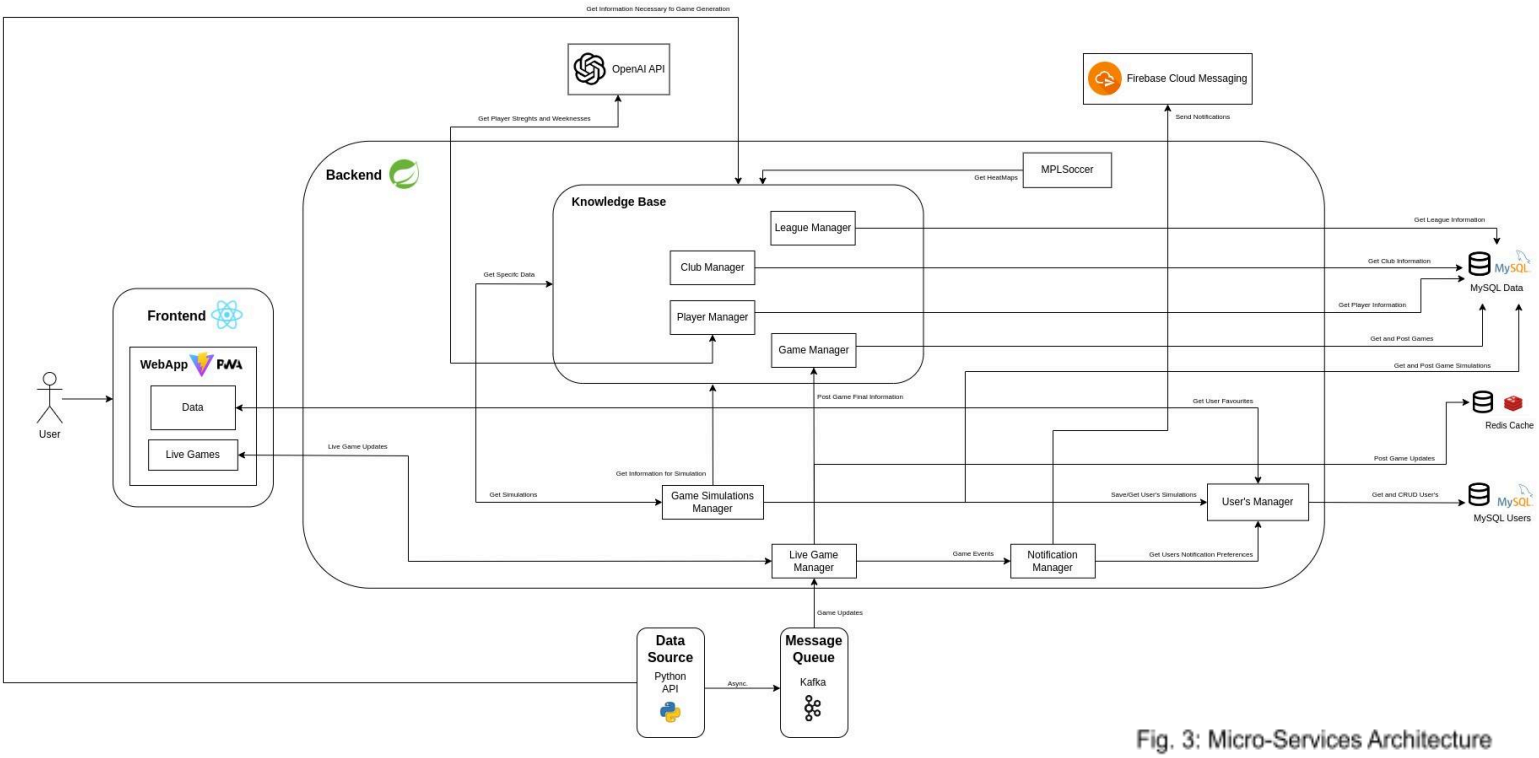

Fig. 3: Micro-Services Architecture

Module interactions

Data Flow for Live Game Updates

Live Game Updates are generated by the Python API (Data Source) which pushes them to the Message Queue. The Backend will receive these updates from the Message Queue and store them in the Redis Cache. When the Frontend asks for updates the Backend retrieves the latest information from the Cache and sends them to the Frontend for display.

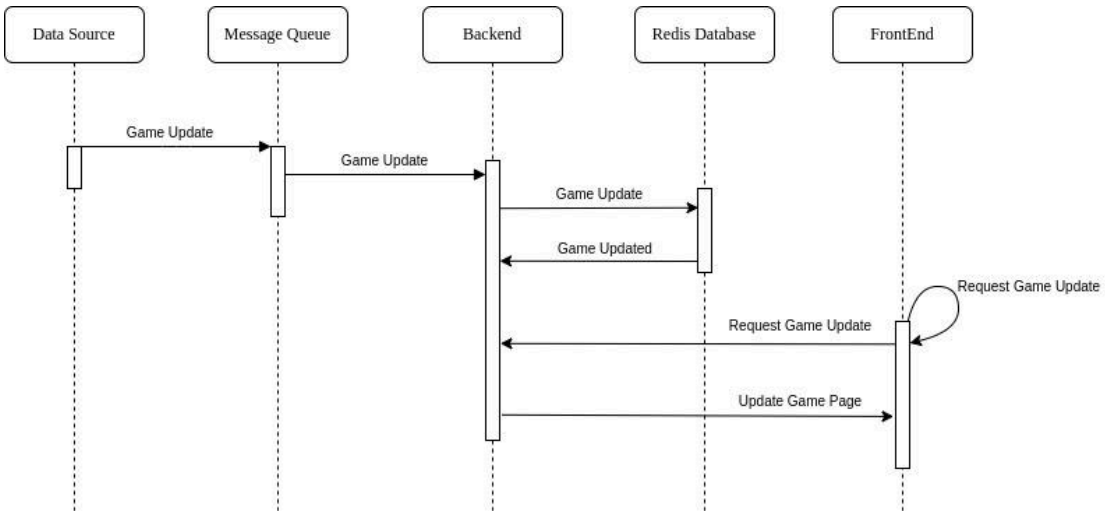

Fig. 4: Live Game Updates Workflow

Fig. 2: Base Architecture

Player & Club Statistics

The Frontend will request for a Specific form of Stats or Data from the Backend which is gonna get them from the mySQL Data Database and display them on the Frontend. If a Graphic/HeatMap is necessary the Backend will request one to the MLPSoccer Library for one to display in the FrontEnd as well.

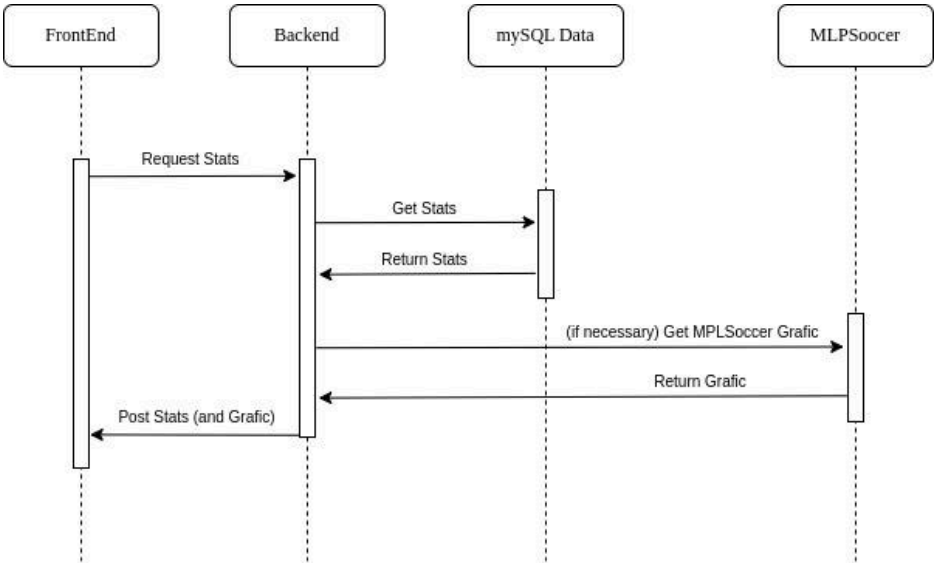

Fig. 5: Get Club and Players Stats Workflow

Game Simulations

The User initiates a Simulation from Frontend, which will be passed to the Backend where it will retrieve the latest statistical information from the Clubs and Players involved and do the Simulation. After running the Simulation it is saved in the mySQL Data Database for future analysis, it is also in the User's Database that the User did that simulation. The results are returned to Frontend for display.

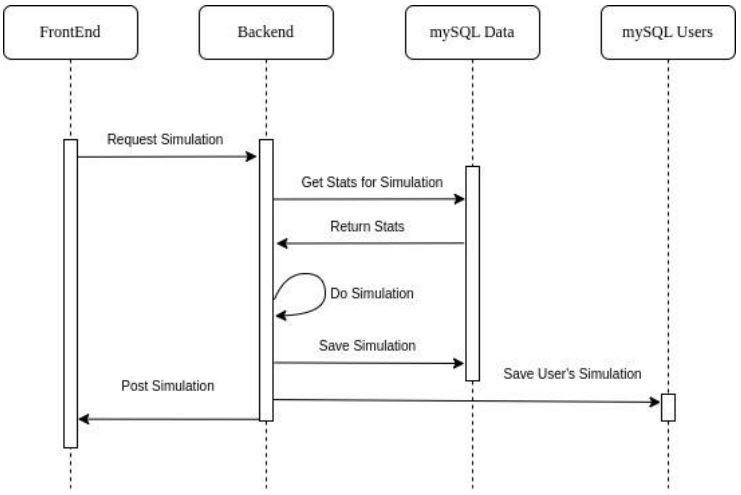

Fig. 6: Game Simulation Workflow

Fig. 2: Base Architecture

Post-Game Data Storage

After the Game Ends the DataSource will send that information which is going to be passed by the Message Queue to the Backend, with that information it does a last update of the Redis Cache and saves all the Game final Stats and Result and puts them in the mySQL Data Database.

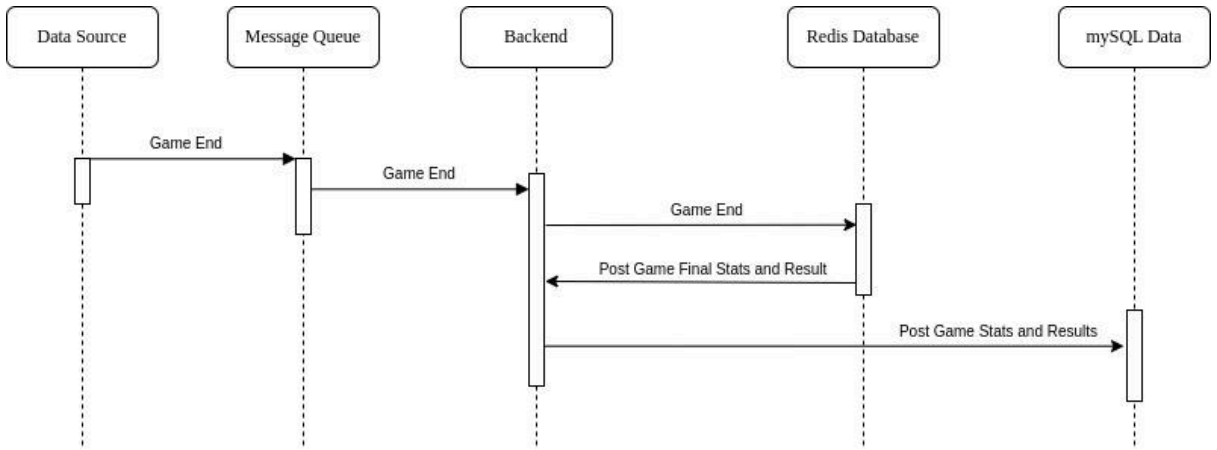

Fig. 7: Live Game Saving Workflow

Push Notifications Mechanism

When the Message Queue receives a Game Event (for example: goals, red cards, substitutions) it will send that information to the Backend which is prepared to trigger notifications using Firebase Cloud Messaging (FCM). The User will receive a notification if the Game Event is something he is subscribed for.

Integration with Internet-Based External Services

The Backend will have external services with the OpenAI API and the MPLSoccer library and when their services are required it will send them the necessary information and receive the results to display on the Frontend. In the case of the OpenAI API it will receive Player statistical data and give out his strengths and weaknesses. For the MPLSoccer library it will help create heatmaps for teams during a game.

User Manager

The User Manager will keep record of user preferences, favourite clubs, leagues and players, and past game simulations. GameSense will also have two types of accounts, being that the Premium one will have access to more Statistical Data and Insights and Game Simulations.

Fig. 2: Base Architecture

4 Information perspective

Domain Concepts

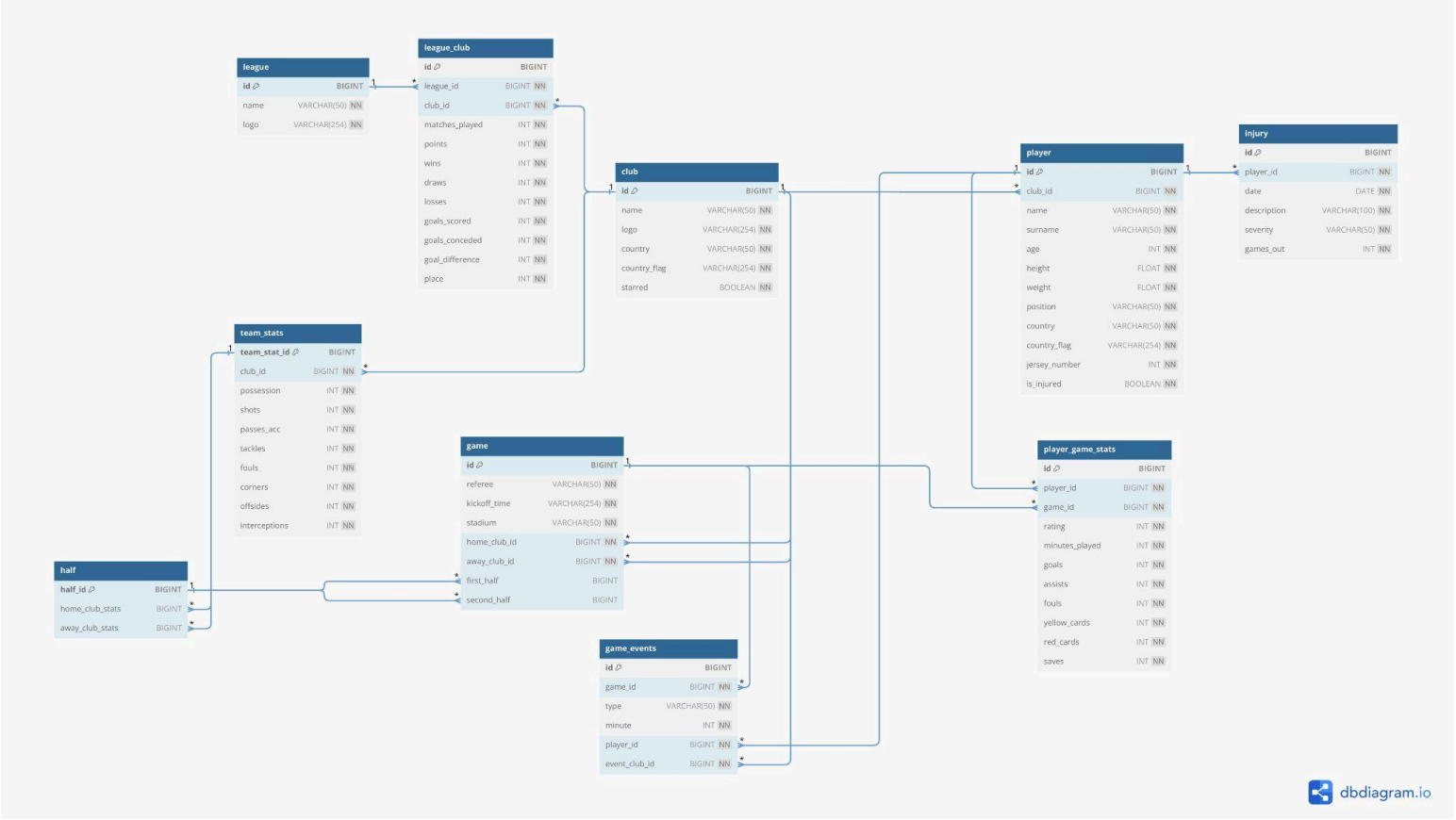

Fig. 8: Persistent Database Diagram

League

Represents football leagues that contain multiple clubs.

League_Club

Serves as a connection between the League and Club entities. It has information about how many games were played, won, drawn, lost, current league standing, etc..

Club

Represents clubs that are playing in leagues.

Player

Represents Players that are associated with a specific club.

Injury

Tracks Injury History of all players.

Game

Represents a Game between two Clubs, it also contains information related to two halves.

Half

A Game has two halves associated with it, this entity saves those halves. Each one of those

Fig. 2: Base Architecture

halves is responsible for storing the stats for that half.

Team_Stats
Each Club has Stats for each Half of a Game, and this entity saves those stats.

Game_Events
Represents major Events that happen in a specific game, associating with the Club and Player involved.

Player_Game_Stats
Each Player has particular stats in a game so this table saves those stats.

Data Flows

Game Data
When a Game is inserted in the database it automatically updates all members involved. The League_Club is updated with the result, Team_Stats and Player_Game_Stats will also have new entries for future detailed analysis and if a Player gets injured it will also update the Injury and Player tables respectively.

Fig. 2: Base Architecture

5 References and resources

Key Components

The GameSense application relies on various technologies to ensure its functionality. Below are the primary technologies used:

Backend

- SpringBoot
- Python
- Kafka
- MySQL
- Redis

Frontend

- React.js
- Vite
- TailwindCSS
- React Auth Kit
- DaisyUI
- HeadlessUI

External Integrations

- OpenAI API
- Firebase Cloud Messaging

Data Visualization

- MPLSoccer Library

Code Version Control

- GitHub

Resources and References

Libraries and Tools Documentation

- HeadlessUI Documentation: https://headlessui.com/
- TailwindCSS Documentation: https://tailwindcss.com/
- Daisy UI Documentation: https://daisyui.com/
- React Auth Kit Documentation: https://authkit.arkadip.dev/
- MySQL Documentation: https://dev.mysql.com/doc
- React.js Documentation: https://reactjs.org
- Redis Official Guide: https://redis.io
- Apache Kafka Overview: https://kafka.apache.org
- Theoric Contents of IES

Fig. 2: Base Architecture

Inspiration and Similar Projects

- FlashScore
- ZeroZero
- FotMob

Fig. 2: Base Architecture