

Projet de Fin d'Etudes DUT

Département : Informatique

Option : Génie informatique

Titre du projet :

KickInSights : Outil d'Analyse et de Suivi des Statistiques des Joueurs pour l'Optimisation des Performances

Préparé par :

Alala Mohamed

Elmaaloumi Hamza

Encadré par

Pr. Samia Nasiri

Soutenu le 11 juin devant la Jury :

Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce rapport. Tout d'abord, je remercie Madame Nasiri pour sa précieuse guidance et ses conseils avisés tout au long de ce projet.

Table de matière

Remerciements	i
Table de matière	ii
Liste des figures.....	v
Introduction	1
Chapitre 1 : Contexte Général du Projet.....	2
1. Présentation du Projet	2
2. Problématiques.....	2
3. Objectifs du projet.....	3
4. Conclusion	4
Chapitre 2 : Analyse et conception.....	5
I. Démarche suivre	5
1. Cycle de vie du projet	5
2. Langage UML.....	7
3. Architecture en couches.....	7
II. Etudes et spécification des besoins.....	9
1. La navigation dans l'application.....	9
2. Les acteurs	10
III. Diagramme de cas d'utilisation.....	11
1. Visiteur	11
2. Directeur	11
3. Administrateur	13
IV. Message émis/reçu.....	14
1. Authentification	14
1.1 Diagramme de séquence.....	15
2. Scraping joueurs	16
IV. Diagramme de classe	19
Chapitre 3 : Réalisations et interfaces	21
I. Outils et technologies utilisées	21
1. HTML	21
2. Type Script	21
3. React	22
4. Python.....	22

5. Django	23
6. Selenium	23
6. Github	23
II. Processus ETL (Extraction, Transformation, Chargement)	24
1. Web Scraping	24
2. Nettoyage et Fusion	25
3. Chargement dans la base de données.....	26
III. Présentation de l'application	26
1. Interface d'authentification.....	26
2. Interface manager	27
2.1 Section des joueurs.....	28
2.1.1 Section des matches d'un joueur.....	29
2.1.2 Section des statistiques d'un joueur	30
2.2 Section des matches.....	31
2.2.1 Matches récents	32
2.2.2 Prochains matchs.....	35
2.3 Section du lineup	38
3. Interface admin	39
3.1 Managers	40
3.1.1 Ajouter un manager.....	41
3.1.2 Modifier un manager.....	42
3.1.3 Supprimer un manager	43
3.2 Utilisateurs	43
3.2.1 Supprimer un utilisateur.....	43
3.3 Leagues.....	44
3.4 Equipes	44
3.4.1 Equipes nationales.....	45
3.4.2 Clubs	45
3.5 Joueurs.....	46
3.5 Scraper.....	47
4. Conclusion	47
IV. Module de Recommandation de Joueurs	48
1. La création des caractéristiques des joueurs	49
2. Pondération personnalisée des caractéristiques	49
3. Cosine Similarity	49
4. Introduction à la représentation vectorielle	51

5. Interface de recommandation	52
6. Conclusion	53
Chapitre 4 : Conclusion du projet.....	54
I. Améliorations techniques	54
II. Améliorations personnelles et professionnelles	54
Webographie	56

Liste des figures

Figure 1 : Cycle de vie d'un projet web	5
Figure 2 : Cycle de vie en V.....	6
Figure 3 : La hiérarchie des diagrammes UML 2.0 sous forme d'un diagramme de classes.	7
Figure 4 : Application model	9
Figure 5 : Navigation dans l' application	9
Figure 6 : Acteurs du système	10
Figure 7 : Diagramme de cas d'utilisation Visiteur.....	11
Figure 8 : Diagramme de cas d'utilisation directeur	12
Figure 9 : Diagramme de cas d' utilisation Administrateur.....	13
Figure 10 : Diagramme de séquence – Authentification	16
Figure 11 : Diagramme de séquence – Scraping joueurs.....	18
Figure 12 : Diagramme de classe.....	19
Figure 13 : Logo de HTML5.....	21
Figure 14 : Type Script Logo	21
Figure 15 : React Logo	22
Figure 16 : Logo de Python.....	22
Figure 17 : Logo de Django.....	23
Figure 18 : Logo de Selenium	23
Figure 19 : Logo de Github	24
Figure 20 : Processus ETL	24
Figure 21 : SofaScore logo	25
Figure 22 : Interface authentication.....	27
Figure 23 : Interface manager	28
Figure 24 : Interface Joueurs.....	29
Figure 25 : Interface des matches joués par un joueur	29
Figure 26 : Exemple des statistiques d' un joueur	31
Figure 27 : Interface des prochains matchs	32
Figure 28 : Interface des matches recents	32
Figure 29 : Exemple des statistiques d'equipe par match	34

Figure 30 : Exemple des statistiques d'un adversaire	37
Figure 31 : Interface de composition	39
Figure 32 : Les possibles schémas tactiques	39
Figure 33 : Interface admin	40
Figure 34 : Interface des managers.....	41
Figure 35 : L'ajout d' un manager.....	42
Figure 36 : Modification d' un manager	42
Figure 37 : Suppression d' un manager	43
Figure 38 : Interface Users	43
Figure 39 : Supprimer un utilisateur	44
Figure 40 : Interface des leagues	44
Figure 41 : Interface des equipes nationaux	45
Figure 42 : Interface des Clubs	46
Figure 43 : Interface des joueurs.....	47
Figure 44 : Interface des boutons de scraping.....	47
Figure 45 : pondérations personnalisées pour chaque position.....	49
Figure 46 : Simple représentation des évaluations de livres	50
Figure 47 : Visualisation des deux vecteurs d'exemple précédent.....	51
Figure 48 : Représentation des statistiques de joueur sous forme de vecteur	52
Figure 49 : Interface de recommandation	53

Liste des tableaux

Tableau 1 : Description de cas d'utilisation - Authentification	15
Tableau 2 : Description de cas d'utilisation - Scraping joueurs	17

Introduction

Dans le cadre de mon projet de fin d'études, j'ai conçu et développé une **plateforme web innovante** dédiée au suivi et à l'analyse des performances des joueurs de football au Maroc. Ce rapport retrace les différentes étapes de ce projet, depuis l'analyse des besoins jusqu'à la conception technique, en passant par les défis rencontrés et les solutions apportées.

Le premier chapitre présente le **contexte général du projet**, incluant les enjeux du football marocain, les problématiques liées à la gestion manuelle des statistiques des joueurs, ainsi que les objectifs visés par cette plateforme. Il détaille également la méthodologie adoptée et le planning suivi pour mener à bien ce travail.

Le deuxième chapitre se concentre sur **l'analyse et la conception du système**. Il expose les besoins fonctionnels et non fonctionnels, identifie les acteurs clés (managers, entraîneurs, analystes), et présente les différents **diagrammes UML** (cas d'utilisation, séquences, classes, etc.) qui ont guidé le développement.

Le troisième chapitre aborde **les réalisations techniques**, décrivant l'architecture du système, les technologies utilisées (web scraping, base de données, frameworks web), ainsi que les fonctionnalités principales : collecte automatisée des données, visualisation des statistiques, outils de comparaison et suggestions de compositions d'équipe.

Enfin, le quatrième chapitre conclut ce rapport en résumant les **acquis du projet**, son impact potentiel sur le milieu footballistique marocain, et les compétences techniques et managériales développées tout au long de ce travail.

Cette plateforme vise à **révolutionner la gestion des performances sportives** en offrant aux professionnels du football un outil fiable, intuitif et basé sur la data, contribuant ainsi à une prise de décision plus éclairée et performante.

Chapitre 1 : Contexte Général du Projet

1. Présentation du Projet

Le football est un sport très populaire au Maroc, avec une forte présence de clubs professionnels et amateurs. Les joueurs, qu'ils évoluent dans les championnats locaux ou qu'ils aspirent à une carrière internationale. Cependant, le suivi de ces performances repose encore largement sur des méthodes traditionnelles, souvent manuelles et peu structurées, ce qui limite l'efficacité des décisions prises par les managers et les coachs.

Face à ce constat, ce projet ambitionne de développer une plateforme web innovante qui permettra aux managers de suivre les statistiques des joueurs marocains de manière efficace et centralisée. Grâce à l'intégration de technologies de web scraping, la plateforme collectera automatiquement des données sur les joueurs et les matchs en temps réel, offrant ainsi une vision claire et précise de leurs performances.

Les managers bénéficieront d'un accès simplifié aux statistiques détaillées des joueurs, qu'il s'agisse de leurs performances en match, de leurs progressions au fil du temps ou de leurs points forts et axes d'amélioration. Ils pourront également consulter les notes des matchs passés et utiliser ces données pour élaborer des stratégies optimales. De plus, la plateforme offrira un outil de création de compositions d'équipe basé sur des critères de performances, facilitant ainsi la gestion et l'optimisation des effectifs.

2. Problématiques

Le projet répond à plusieurs problématiques majeures :

- **Manque de données centralisées** : Les managers ont du mal à obtenir des statistiques fiables sur les joueurs, car les informations sont souvent dispersées entre plusieurs sources. Cette fragmentation rend l'analyse globale difficile et entraîne un manque de visibilité sur les performances réelles des joueurs.
- **Suivi manuel inefficace** : L'analyse des performances repose souvent sur des observations subjectives plutôt que sur des données chiffrées. Cette approche limite la précision des évaluations et empêche une prise de décision éclairée. De plus, l'absence d'un outil automatisé entraîne une perte de temps considérable pour les entraîneurs et les managers qui doivent compiler eux-mêmes les données.

- **Accès limité aux données en temps réel** : Les managers n'ont pas toujours accès aux dernières mises à jour concernant les performances des joueurs, ce qui peut impacter les décisions en cours de saison.
- **Difficulté à comparer les joueurs** : Sans indicateurs normalisés, il est compliqué de comparer objectivement plusieurs joueurs évoluant à des postes similaires.
- **Difficultés d'organisation** : La constitution d'équipes optimales sans accès à des données précises est un défi pour les coachs et managers. Le manque de statistiques actualisées complique la sélection des joueurs en fonction de leurs performances récentes, ce qui peut nuire aux résultats et aux stratégies mises en place.

Pour répondre à ces défis, il est nécessaire de développer une solution technologique qui centralise et automatise la gestion et l'analyse des performances des joueurs, tout en offrant une interface conviviale et accessible aux managers et coachs. L'objectif de ce projet est de concevoir une plateforme web qui répond à ces besoins en simplifiant le suivi des joueurs, en réduisant les approximations dans les évaluations et en améliorant la prise de décision grâce à des données fiables et mises à jour en temps réel.

3. Objectifs du projet

Les objectifs principaux du projet sont les suivants :

- **Automatiser la collecte des données** sur les joueurs et les matchs grâce au web scraping, permettant ainsi de réduire le temps passé à rechercher et compiler manuellement les informations.
- **Fournir une interface intuitive et interactive** permettant aux managers d'accéder rapidement aux statistiques des joueurs et des équipes, avec des tableaux de bord personnalisables et des filtres avancés.
- **Faciliter l'analyse des performances** en intégrant des indicateurs clés tels que les statistiques de match, les moyennes de performance et les comparaisons entre joueurs.
- **Améliorer la prise de décision** en offrant des outils d'aide à la constitution d'équipes, en suggérant les meilleures compositions basées sur les performances récentes et les caractéristiques des adversaires.
- **Créer une base de données historique** permettant aux managers de suivre l'évolution des joueurs sur plusieurs saisons et d'identifier les talents prometteurs.

4. Conclusion

La première phase du projet était de comprendre l'objectif du sujet et de dégager la problématique et les objectifs visés. Dans ce qui suit, nous allons nous intéresser à l'analyse et la conception et à l'élaboration des différents diagrammes UML.

Chapitre 2 : Analyse et conception

I. Démarche suivre

1. Cycle de vie du projet

« **Cycle de vie d'un logiciel** » (en anglais *software lifecycle*), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la **validation** du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la **vérification** du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

L'origine de ce découpage est qu'au cours des cinquante dernières années, l'amélioration des conditions de vie des populations a été un des principaux axes de travail de nombreux programmes de développement. Des investissements considérables ont été consentis tant au niveau humain que financier. Toutefois, force est de constater que les résultats n'ont pas toujours été à la hauteur des attentes. La réponse aux besoins fondamentaux des communautés a parfois été insatisfaisante, partielle ou ponctuelle. Les projets ont souvent coûté et duré plus que prévu et leurs effets, négatifs dans certains cas, n'ont pas toujours été anticipés.

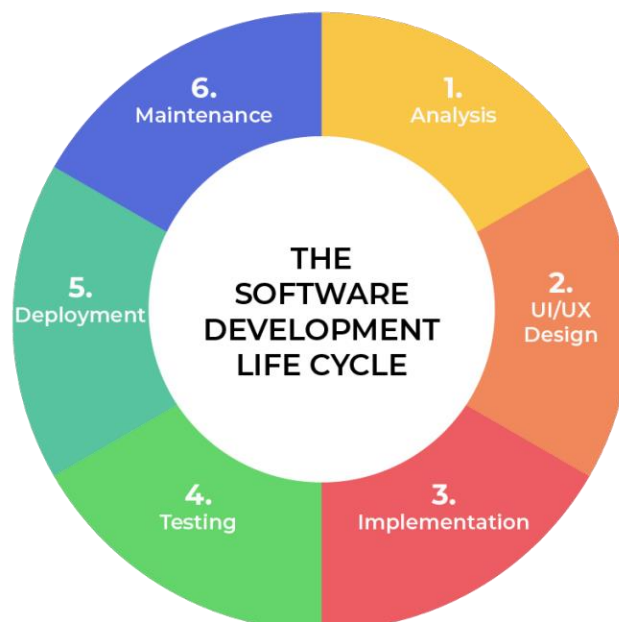


Figure 1 : Cycle de vie d'un projet web

Afin de contrôler les risques et de mener à bon terme le projet vue sa complexité j'ai opté pour le modèle de cycle de vie en V.

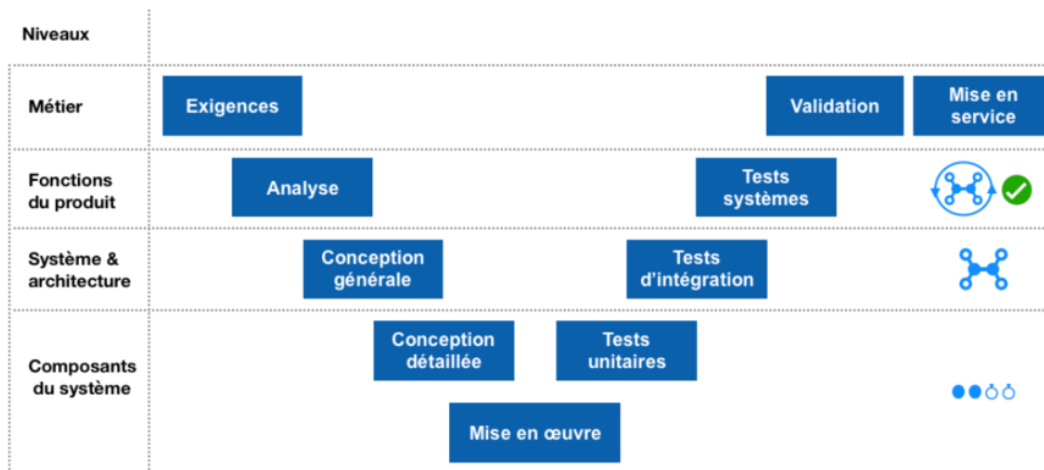


Figure 2 : Cycle de vie en V

- Exigences : les exigences font l'objet d'une expression des besoins. Le cas échéant, une étude de faisabilité peut être conduite avant d'engager les travaux.
- Analyse : il s'agit, à partir de l'expression de besoin, d'établir le cahier des charges fonctionnel ou les spécifications fonctionnelles.
- Conception générale, aussi appelé conception architecturale ou conception préliminaire: il s'agit de concevoir le système qui doit répondre aux exigences et de définir son architecture, et en particulier les différents composants nécessaires.
- Conception détaillée: il s'agit de concevoir chaque composant, et la manière dont ils contribuent à la réponse aux besoins.
- Mise en œuvre: il s'agit de réaliser chaque composant nécessaire. Pour les composants et systèmes logiciels, l'activité est essentiellement le codage.
- Test unitaire: il s'agit de vérifier le bon fonctionnement et la conformité de chaque composant à sa conception détaillée.
- Intégration et test d'intégration: il s'agit d'assembler le système à partir de tous ses composants, et de vérifier que le système dans son ensemble fonctionne conformément à sa conception générale.
- Test système (anciennement « tests fonctionnels ») : vérification que le système est conforme aux exigences.

- Test d'acceptation (également appelés « recette » dans le contexte de la sous-traitance) : validation du système par rapport à sa conformité aux besoins exprimés.

2. Langage UML

UML (*Unified Modeling Language*, que l'on peut traduire par « *langage de modélisation unifié* ») est un langage de modélisation graphique et textuel destiné à comprendre et à décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

Ce langage est né de la fusion de plusieurs méthodes existantes auparavant, et est devenu désormais la référence en termes de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

Le méta modèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relie.

UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues. Une vue est constituée d'un ou plusieurs diagrammes.

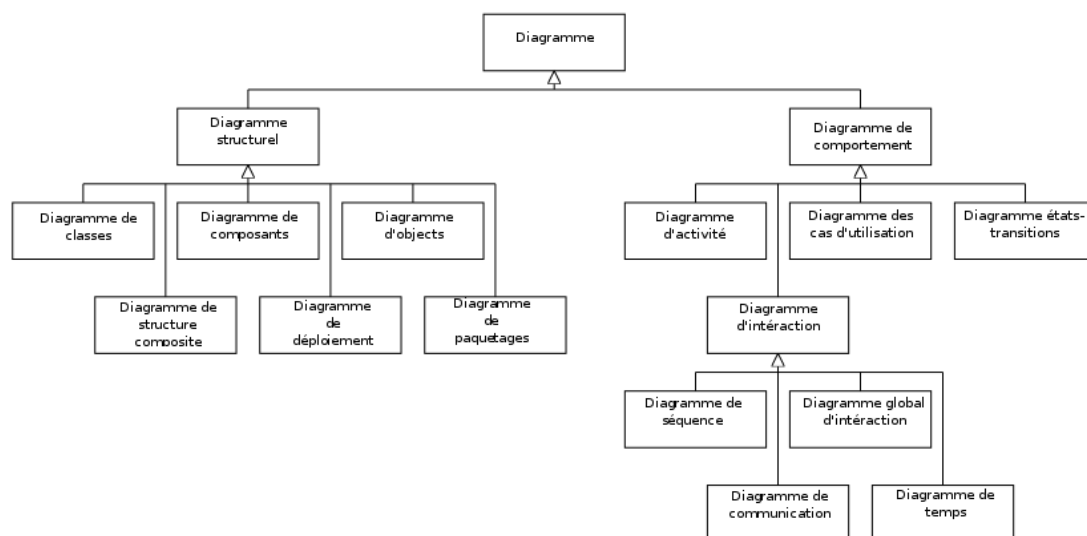


Figure 3 : La hiérarchie des diagrammes UML 2.0 sous forme d'un diagramme de classes.

3. Architecture en couches

Architecture en couches est un modèle de conception utilisé dans le développement logiciel pour organiser le code et structurer les applications en couches distinctes, chacune étant responsable d'un aspect spécifique de l'application. Cette approche favorise la modularité, la

séparation des préoccupations et la maintenabilité, facilitant ainsi le développement, les tests et la gestion des applications. Voici un aperçu de l'architecture en couches et de ses avantages :

Qu'est-ce que l'architecture en couches ?

L'architecture en couches structure une application en plusieurs niveaux, chacun ayant un rôle différent. Ces couches interagissent de manière hiérarchique (du haut vers le bas), où les couches supérieures dépendent des couches inférieures, mais pas l'inverse.

Une configuration courante comprend les couches suivantes :

1. Presentation Layer (Frontend/UI)

- Gère les interactions entre les utilisateurs et le logiciel.
- Se concentre sur l'interface utilisateur (UI) et l'expérience utilisateur (UX). Les développeurs de cette couche travaillent à rendre les applications visuellement attrayantes et faciles à utiliser.
- **Technologies** : React, Angular, Vue.js, Kotlin (Android).

2. Business Logic Layer (Domain Layer)

- Cœur de l'application, où les règles métier et la logique applicative sont implémentées.
- Traite les données, applique les règles métier et contrôle les transactions.
- **Technologies** : Java, Python, C#, Node.js, divers frameworks backend.

3. Data Access Layer (Persistence Layer)

- Gère le stockage et la récupération des données depuis des bases de données ou d'autres systèmes de stockage persistants.
- Effectue les opérations CRUD (Create, Read, Update, Delete) et interagit avec les sources de données (SQL, NoSQL).
- **Technologies** : SQL, MongoDB, ORM (ex. Hibernate, Entity Framework).

4. Data Layer

- La base de données elle-même, où les données sont stockées sous forme structurée ou semi-structurée.
- **Technologies** : MySQL, PostgreSQL, MongoDB, Oracle, Redis.

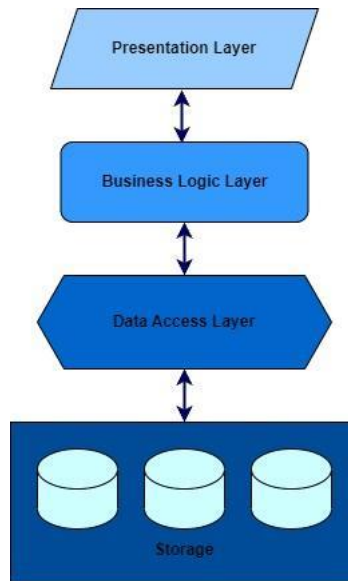


Figure 4 : Application model

II. Etudes et spécification des besoins

1. La navigation dans l'application

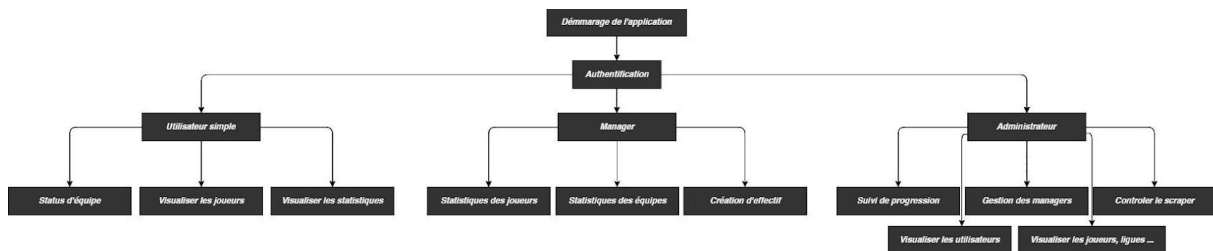


Figure 5 : Navigation dans l' application

Démarrage de l'application

- **Étape d'authentification :**
 - Obligatoire pour les utilisateurs disposant d'un compte.
 - Non requise pour un accès limité (visiteur).

Profil : Utilisateur Simple

- Consultation de la composition et des informations générales de l'équipe.
- Liste complète et fiches descriptives des joueurs.

- Données agrégées (performances des joueurs, résultats d'équipes, historiques de matches).

Profil : Manager

- Statistiques des joueurs (performances individuelles, tendances).
- Statistiques des équipes (analyse comparative, indicateurs clés).
- Gestion tactique (composition des équipes, stratégies).
- Outils de planification.

4. Profil : Administrateur

- Tableau de bord global (activités utilisateurs, données en temps réel).
- Visualisation des comptes et des interactions.
- Création/modification/suppression des comptes managers.
- Contrôle de l'outil de collecte automatique de données.
- Supervision des sources (joueurs, ligues, compétitions).

2. Les acteurs

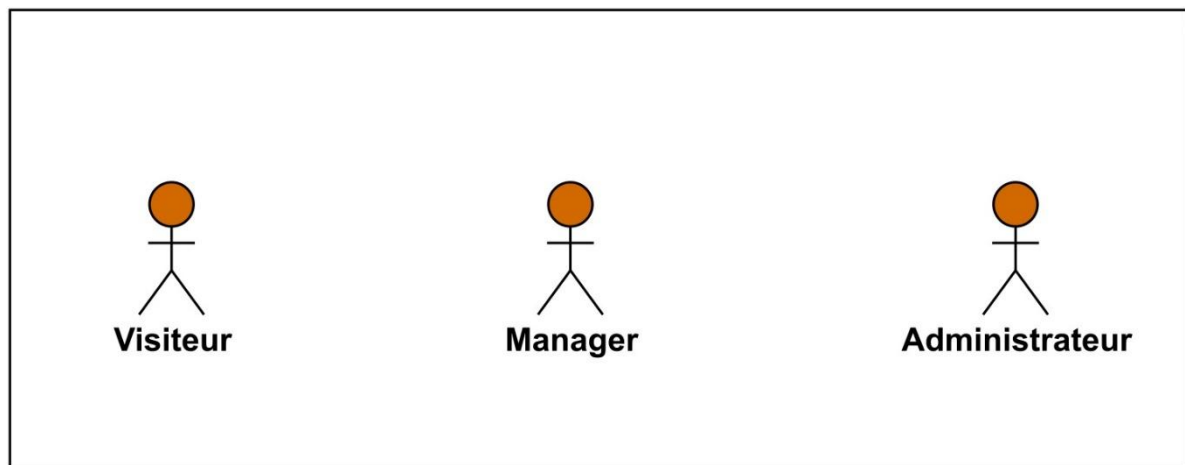


Figure 6 : Acteurs du système

III. Diagramme de cas d'utilisation

1. Visiteur

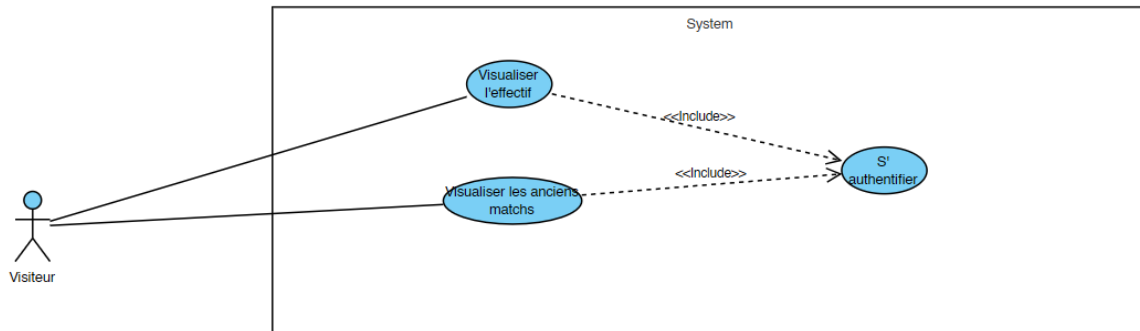


Figure 7 : Diagramme de cas d'utilisation Visiteur

Le diagramme ci-dessus illustre les interactions possibles pour un simple **visiteur** du système. Un visiteur est un utilisateur non authentifié qui peut accéder à certaines fonctionnalités du système sous conditions.

Deux cas d'utilisation principaux lui sont proposés :

- **Visualiser l'effectif** : le visiteur peut accéder à la liste des joueurs, leurs postes, ainsi que d'autres informations générales sur l'équipe.
- **Visualiser les anciens matchs** : cette option permet au visiteur de consulter les résultats et les détails des matchs précédents.

2. Directeur

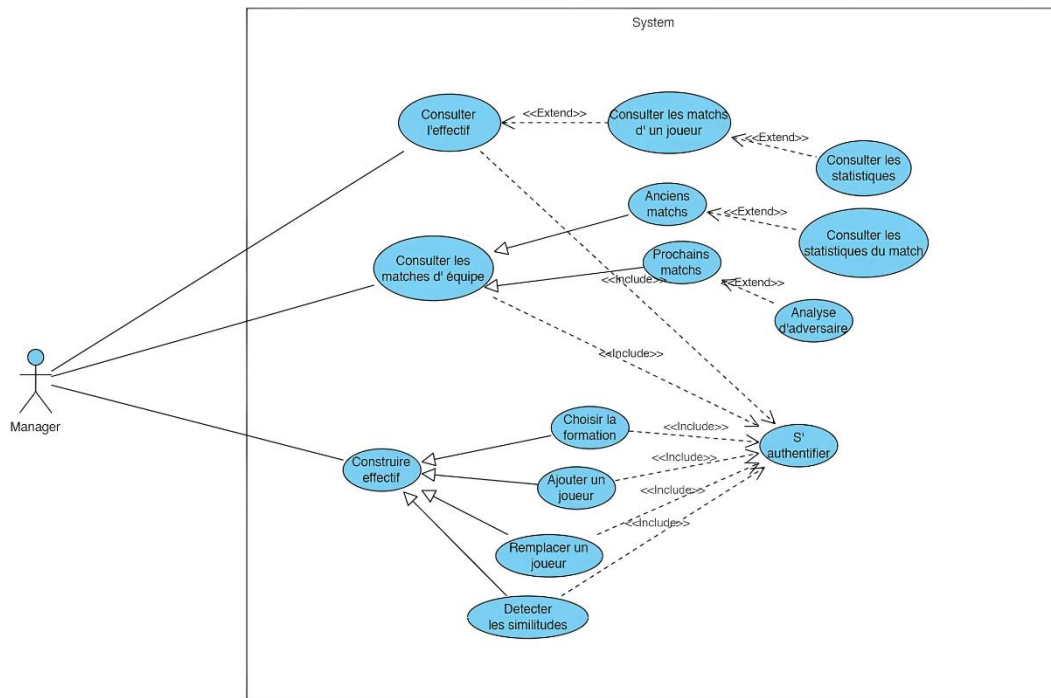


Figure 8 : Diagramme de cas d'utilisation directeur

En tant que Manager, j'ai accès à une interface riche qui me permet de consulter et de gérer de nombreuses informations liées à l'équipe. Le diagramme ci-dessus illustre les différents cas d'utilisation qui me sont proposés dans le système.

Avant d'accéder à ces fonctionnalités, je dois obligatoirement **m'authentifier**, ce qui garantit la sécurité et la traçabilité des actions effectuées.

Une fois connecté, je peux :

Consulter les données :

- **L'effectif de l'équipe** pour avoir une vue d'ensemble sur les joueurs disponibles.
- **Les matches de l'équipe**, qu'ils soient anciens ou à venir.
 - En consultant ces matches, je peux aussi accéder aux **statistiques détaillées**, aussi bien globales que par joueur.
 - Il est également possible d'effectuer une **analyse de l'adversaire** en préparation d'un prochain match.

Gérer l'effectif :

- Je peux **construire un effectif** adapté en ajoutant ou remplaçant des joueurs.

- Une fonctionnalité permet aussi de **détecter les similitudes** entre les joueurs afin de faciliter les décisions de sélection.
- Enfin, je choisis la **formation tactique** qui sera utilisée lors du match.

Grâce à cet ensemble de fonctionnalités, je suis en mesure de préparer efficacement les matchs, d'analyser les performances et d'adapter mes choix en fonction des besoins de l'équipe.

3. Administrateur

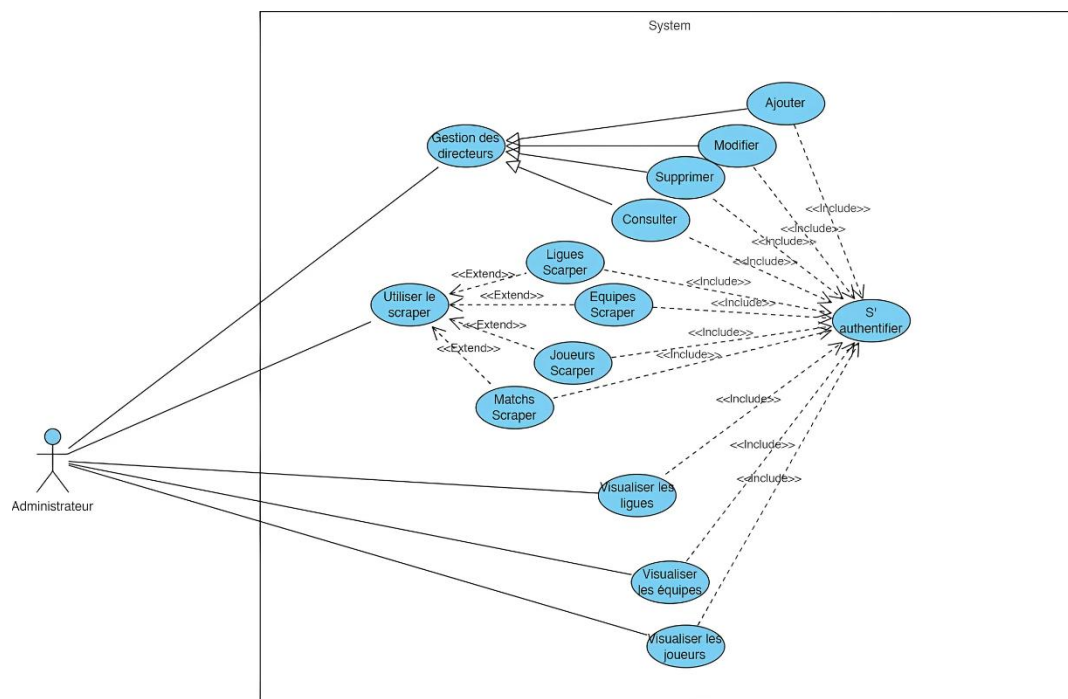


Figure 9 : Diagramme de cas d'utilisation Administrateur

Le diagramme de cas d'utilisation ci-dessus modélise les différentes interactions entre l'acteur principal du système, à savoir **l'administrateur**, et les fonctionnalités offertes par l'application. Ce diagramme a pour objectif de représenter, de manière abstraite et fonctionnelle, les différents services accessibles ainsi que les dépendances entre eux.

1. **S'authentifier** : Ce cas d'utilisation constitue une **prérequis systématique** pour l'accès aux autres fonctionnalités.
2. **Gestion des directeurs** : Ce cas d'utilisation regroupe, via une relation de généralisation, les opérations CRUD suivantes :
 - **Ajouter un directeur**
 - **Modifier un directeur**

- **Supprimer un directeur**
- **Consulter les directeurs**

Ces actions permettent à l'administrateur de gérer efficacement les entités représentant les directeurs au sein du système.

3. **Utiliser le scraper** : Ce cas d'utilisation désigne l'action de lancer un processus de collecte automatique de données à partir de sources externes.

- **Scraper les ligues**
- **Scraper les équipes**
- **Scraper les joueurs**
- **Scraper les matchs**

Cette modélisation traduit la possibilité d'adapter dynamiquement le comportement du cas d'utilisation principal selon les besoins.

4. **Visualisation des données** : Le système offre à l'administrateur la possibilité de visualiser les informations collectées et structurées :

- **Visualiser les ligues**
- **Visualiser les équipes**
- **Visualiser les joueurs**

Ces cas d'utilisation sont également conditionnés par l'authentification de l'utilisateur.

IV. Message émis/reçu

1. Authentification

Ce cas d'utilisation permet aux différents acteurs d'accéder aux services de l'application de gestion de parc informatique selon leurs rôles.

L'authentification permet à un utilisateur d'accéder à l'application, elle se fait par la saisie du login et de mot de passe, si les informations saisies sont identiques aux celles enregistrées dans la base de données, le système affiche la page d'accueil, sinon le système recharge une autre fois la page de l'authentification en affichant un message d'erreur.

Acteur	Utilisateur (visiteur, manager, administrateur)
--------	---

Description	Ce cas d'utilisation décrit le processus d'authentification d'un utilisateur via un formulaire de connexion. Le système vérifie les identifiants fournis pour accorder ou refuser l'accès à l'application.
Action de départ	Soumission du formulaire de connexion avec le nom d'utilisateur et le mot de passe.
Scénario nominal	<ul style="list-style-type: none"> - L'utilisateur saisit ses identifiants et soumet le formulaire. - Le contrôleur vérifie que les champs sont remplis. - Le service d'authentification valide les identifiants. - Si les identifiants sont valides, l'utilisateur est redirigé vers le tableau de bord correspondant à son rôle.
Scénario alternatif	<ul style="list-style-type: none"> - Si des champs sont vides, un message d'erreur est affiché. - Si les identifiants sont incorrects, une erreur d'authentification est retournée à l'utilisateur.

Tableau 1 : Description de cas d'utilisation - Authentification

1.1 Diagramme de séquence

Ce diagramme de séquence illustre le processus d'authentification :

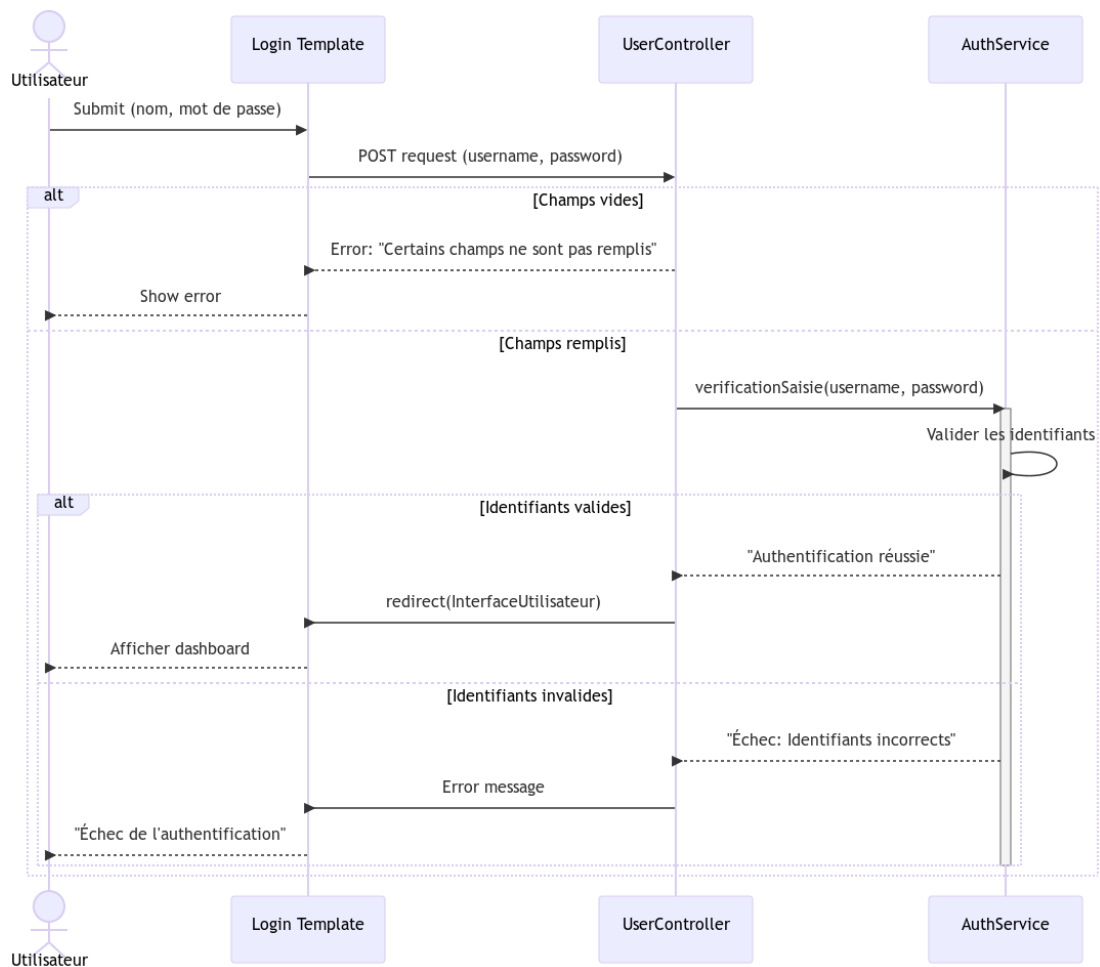


Figure 10 : Diagramme de séquence – Authentification

2. Scraping joueurs

Ce diagramme illustre le processus de scraping des données de joueurs depuis le site **Sofascore** via une requête HTTP initiée par le **Client**.

Acteur	Client (utilisateur de l'application)
Description	Le client envoie une requête pour récupérer la liste des joueurs depuis une source externe (Sofascore).
Action de départ	Requête HTTP : GET /scraping_players envoyée au contrôleur.
Scénario nominal	<ul style="list-style-type: none"> - Le contrôleur appelle le service de scraping. - Le service envoie une requête HTTP à Sofascore. - La réponse HTML est analysée. - Les détails de chaque joueur sont extraits (nom, poste, équipe).

	<ul style="list-style-type: none"> - Les données sont mises en mémoire. - Le service renvoie la liste des joueurs au contrôleur. - Une réponse 200 OK contenant les données est renvoyée au client.
Scénario alternatif	<ul style="list-style-type: none"> - Erreur réseau : Échec de connexion à Sofascore → message d'erreur renvoyé au client. - Erreur de parsing : Échec de l'analyse HTML → message d'erreur renvoyé au client.

Tableau 2 : Description de cas d'utilisation - Scraping joueurs

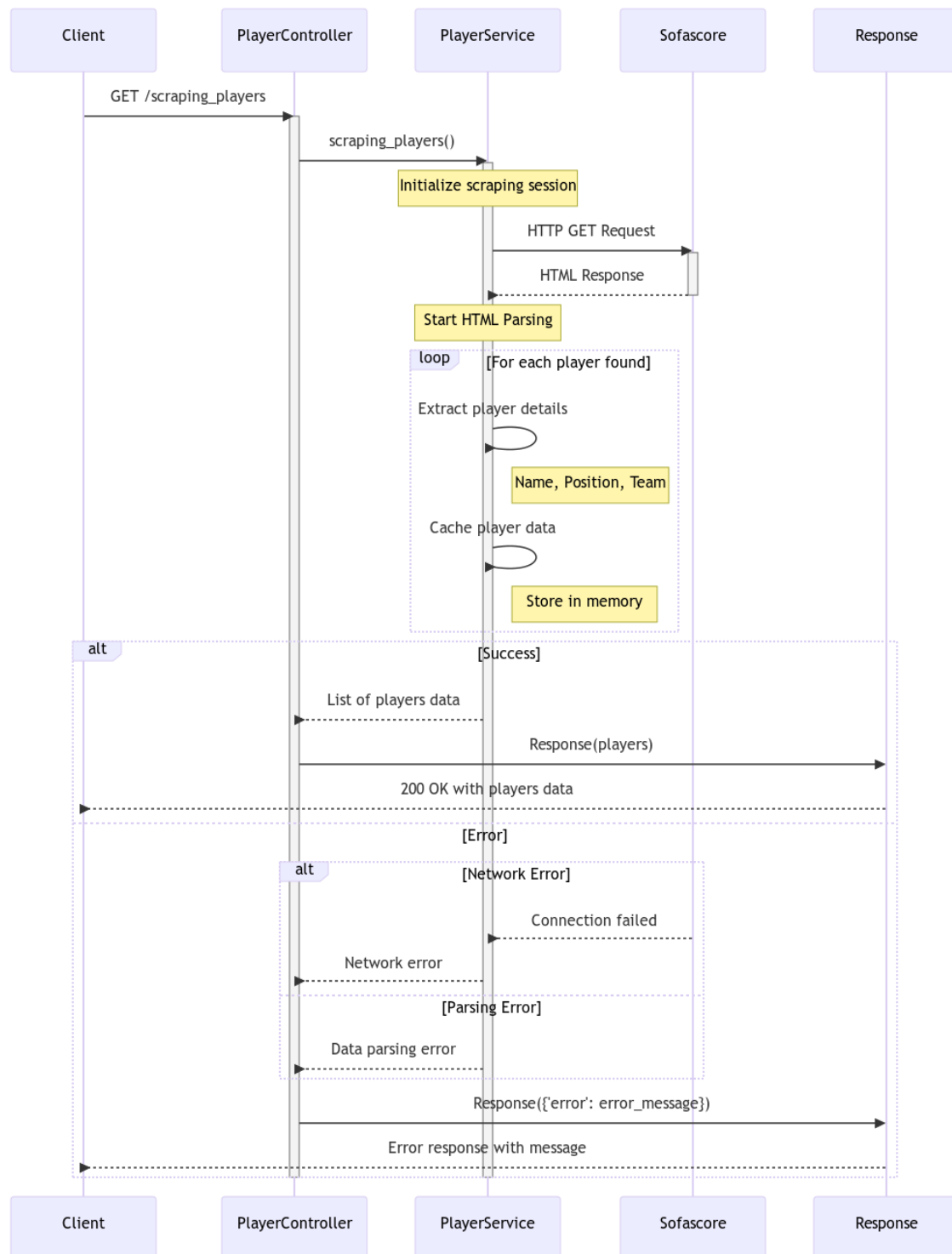


Figure 11 : Diagramme de séquence – Scraping joueurs

IV. Diagramme de classe

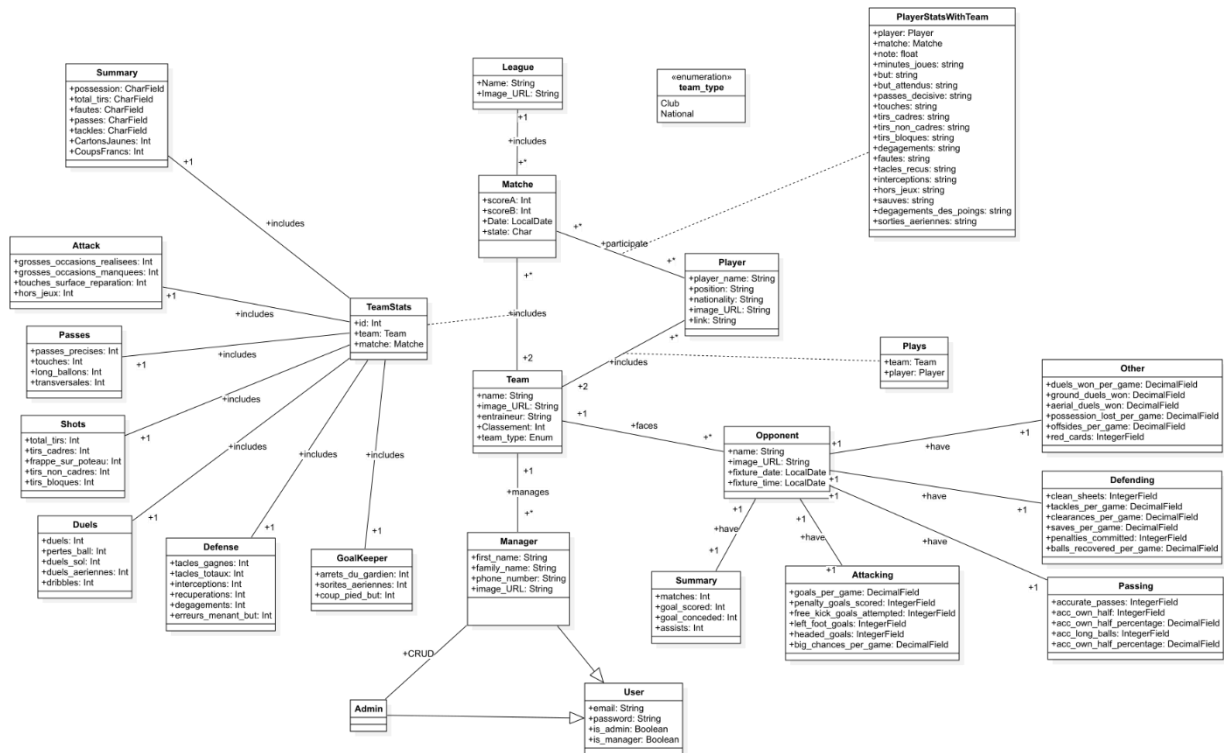


Figure 12 : Diagramme de classe

Le diagramme de classes ci-après modélise les principales entités intervenant dans le système de gestion d'un championnat sportif. Il permet de visualiser la structure des données ainsi que les relations entre les différentes classes du modèle objet.

- **Matche** : cette classe représente une rencontre sportive entre deux équipes. Elle contient les attributs scoreA et scoreB pour enregistrer les scores respectifs des équipes participantes, un attribut Date pour la date du match, ainsi qu'un attribut state pour indiquer l'état actuel du match (en cours, terminé, annulé, etc.).
- **League** : cette classe correspond à une ligue sportive, identifiée par un nom (Name) et une image représentative (Image_URL). Une ligue peut contenir plusieurs équipes, comme l'indique la relation de composition entre League et Team.
- **Team** : cette classe modélise une équipe sportive avec plusieurs attributs : name (nom de l'équipe), image_URL (logo ou image associée), entraîneur (nom de l'entraîneur), Classement (position au classement), ainsi qu'un attribut team_type permettant de spécifier le type d'équipe via une énumération.
- **team_type** : il s'agit d'une énumération représentant les différents types possibles d'équipes (par exemple : club, sélection nationale, etc.).

- **Club** : cette classe semble représenter un type particulier d'équipe, ce qui suggère une relation d'héritage ou de spécialisation avec la classe Team.

Le diagramme illustre également les cardinalités des associations entre les classes, renforçant ainsi la compréhension des contraintes du modèle (par exemple : une ligue contient une ou plusieurs équipes).

Chapitre 3 : Réalisations et interfaces

I. Outils et technologies utilisées

1. HTML



Figure 13 : Logo de HTML5

HTML signifie « HyperText Markup Language » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. D'autres technologies sont utilisées avec HTML pour décrire la présentation d'une page (CSS) et/ou ses fonctionnalités interactives (JavaScript). L'« hypertexte » désigne les liens qui relient les pages web entre elles, que ce soit au sein d'un même site web ou entre différents sites web. Les liens sont un aspect fondamental du Web. Ce sont eux qui forment cette « toile » (ce mot est traduit par web en anglais). En téléchargeant du contenu sur l'Internet et en le reliant à des pages créées par d'autres personnes, vous devenez un participant actif du World Wide Web.

2. Type Script



Figure 14 : Type Script Logo

Type Script est un langage de programmation développé par Microsoft en 2012. Son ambition principale est **d'améliorer la productivité de développement d'applications complexes.**

C'est un **langage open source**, développé comme un sur-ensemble de Javascript. Ce qu'il faut comprendre, c'est que **tout code valide en Javascript l'est également en Type Script**.

3. React

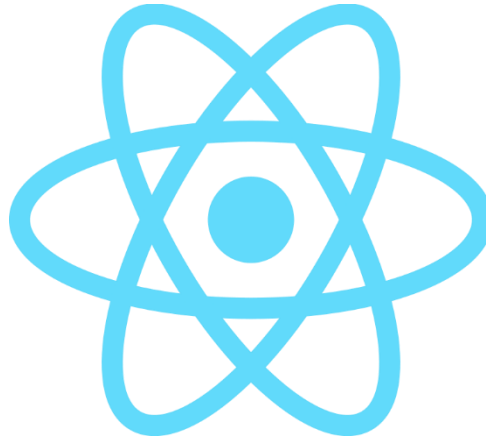


Figure 15 : React Logo

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre. Elle est maintenue par Meta (anciennement Facebook) ainsi que par une communauté de développeurs individuels et d'entreprises depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

4. Python



Figure 16 : Logo de Python

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

5. Django



Figure 17 : Logo de Django

Django est un **framework web open-source** basé sur **Python**, conçu pour faciliter le développement rapide d'applications web sécurisées et maintenables. Suivant le paradigme **MVT (Modèle-Vue-Template)**, il intègre des fonctionnalités prêtes à l'emploi telles qu'un **ORM (Object-Relational Mapping)** pour la gestion des bases de données, un système d'authentification robuste, un panneau d'administration automatique et une gestion simplifiée des formulaires.

6. Selenium



Figure 18 : Logo de Selenium

Selenium est un framework open-source initialement développé pour exécuter des tests automatisés sur différents navigateurs (Chrome, Internet Explorer, Firefox, Safari,...). Cette librairie est également utilisée pour effectuer du web scraping, car **elle permet de naviguer entre des pages internet et d'interagir avec les éléments de la page comme un utilisateur réel**. Ainsi, il devient possible de scraper des sites web dynamiques, c'est-à-dire, des sites qui retournent un certain résultat à l'utilisateur en fonction de ses actions.

6. Github



Figure 19 : Logo de Github

GIT HUB est un service web utilisant le logiciel de gestion de versions Git. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet.

II. Processus ETL (Extraction, Transformation, Chargement)

ETL est l'acronyme de Extract, Transform, Load (extraction transformation et chargement). Il s'agit d'un processus utilisé pour collecter des données à partir de différentes sources, les modifier et les nettoyer, puis les charger dans un système ou une base de données cible. Chaque composant de l'ETL joue un rôle essentiel dans l'intégration précise et fiable des données.

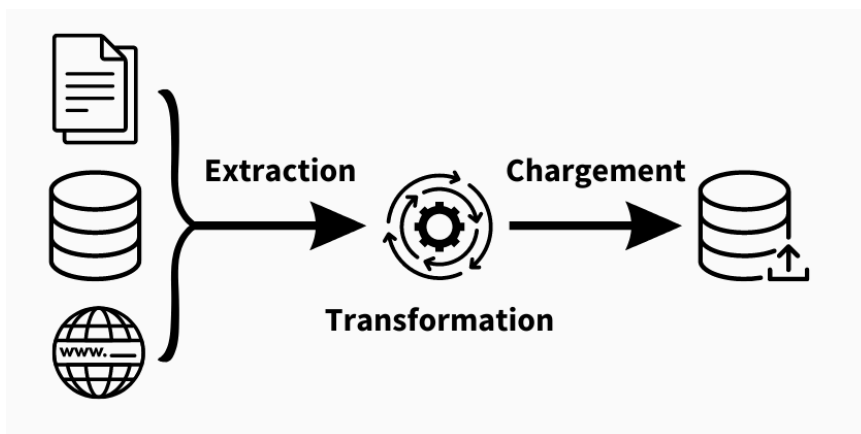


Figure 20 : Processus ETL

1. Web Scraping

Web scraping is a data extraction technique that allows us to get information from websites automatically. Writing code or using specialist tools to browse online pages, download HTML content, and extract pertinent data for analysis or integration into applications is required. Web

scraping allows us to efficiently capture massive amounts of data while also automating the data collection process.

Le web scraping a été utilisé dans le cadre de ce projet de stage afin d'acquérir des données d'événements agrégées depuis Sofa Score. Pour réaliser le scraping web, les méthodes suivantes ont été employées :



Figure 21 : SofaScore logo

La première étape du web scraping consiste à identifier le site web à partir duquel les données seront extraites. Dans ce projet, SofaScore a été choisi comme source principale de données. Ce site fournit des informations détaillées sur les événements footballistiques, notamment des statistiques complètes sur les joueurs.

Pour accéder aux pages web et extraire les données, le langage de programmation Python a été utilisé en combinaison avec la bibliothèque Selenium et l'outil ChromeDriver. Cette configuration permet d'automatiser la navigation dans le navigateur, de charger dynamiquement les pages et d'interagir avec les éléments du DOM.

Une fois les pages chargées, Selenium a permis de parcourir la structure HTML, d'identifier les éléments pertinents (tels que les tableaux, balises div, classes CSS, etc.) et d'extraire les informations nécessaires comme les noms des joueurs, leurs statistiques et autres données liées aux performances. Dans certains cas spécifiques, des expressions régulières ont été utilisées pour affiner l'extraction des données lorsque la structure HTML était complexe ou variable.

2. Nettoyage et Fusion

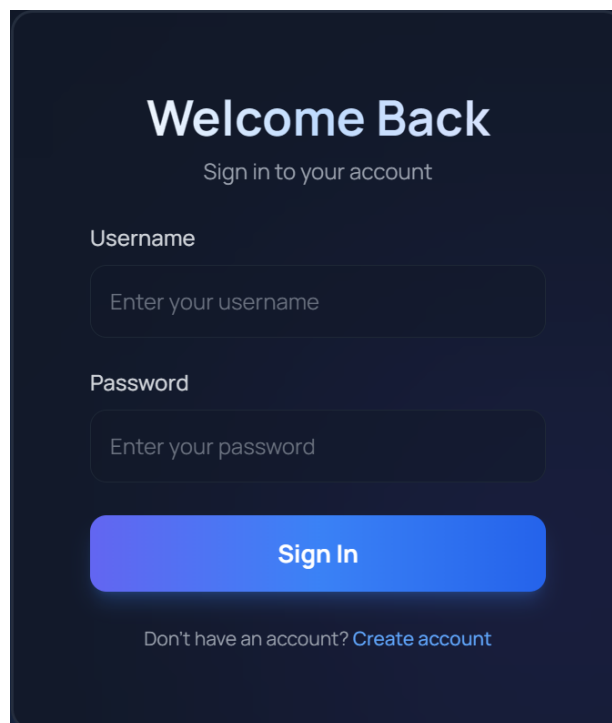
Après l'extraction des données brutes, certaines transformations ont été appliquées afin de nettoyer et de formater correctement les informations collectées. Cela a impliqué la suppression de caractères inutiles, la gestion des valeurs manquantes ou incohérentes, ainsi que la conversion des types de données lorsque cela était nécessaire. Les données ainsi transformées ont été préparées pour une analyse plus approfondie et pour leur intégration dans le pipeline de données du projet.

3. Chargement dans la base de données

Une fois les données extraites et transformées, l'étape suivante consistait à les charger dans une base de données structurée. Pour cela, PostgreSQL a été utilisé comme système de gestion de base de données relationnelle. L'intégration des données dans la base a été facilitée par le framework Django, notamment grâce à l'utilisation des serializers. Ces derniers ont permis de convertir les objets Python (issus du traitement des données) en représentations compatibles avec les modèles de la base de données. Chaque entité extraite (joueurs, équipes, statistiques, etc.) a été mappée sur une table spécifique, assurant ainsi une organisation cohérente, une normalisation des données et une facilité d'accès pour les étapes ultérieures du projet.

III. Présentation de l'application

1. Interface d'authentification



Welcome Back

Sign in to your account

Username

Password

Sign In

Don't have an account? [Create account](#)

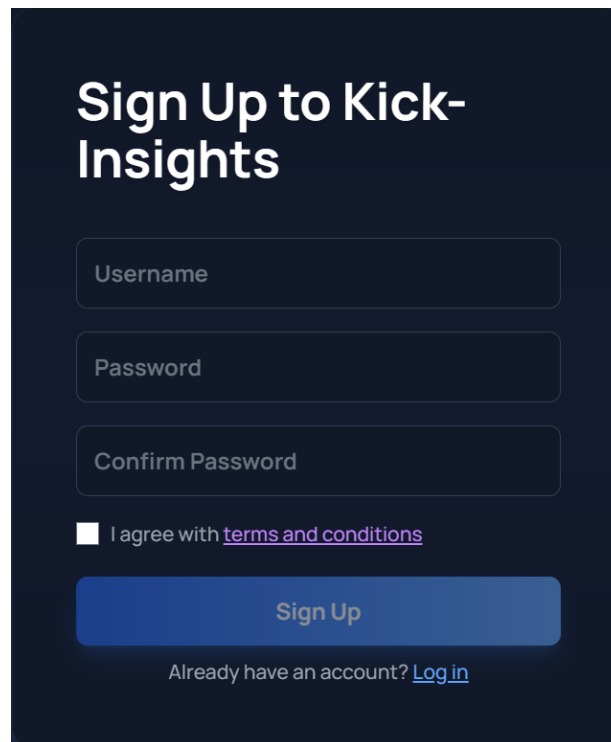
The image shows a dark-themed sign-up form for 'Kick-Insights'. At the top, the title 'Sign Up to Kick-Insights' is displayed in white. Below the title are three input fields: 'Username', 'Password', and 'Confirm Password', each with a light blue border. Under the 'Confirm Password' field is a checkbox with the text 'I agree with [terms and conditions](#)'. At the bottom of the form is a large blue button labeled 'Sign Up'. Below the button, there is a link that says 'Already have an account? [Log in](#)'.

Figure 22 : Interface authentication

Après l'authentification, et si les informations entrées sont valides, le système affiche la page D'accueil, cette page est affichée selon le rôle de l'utilisateur.

2. Interface manager

L'interface manager est conçue pour offrir une navigation simple et intuitive, permettant aux managers de gérer efficacement leurs effectifs et les joueurs associées. Voici les éléments de navigation disponible pour les managers:

- Squad : Gestion globale de l'équipe (joueurs, statistiques).
- Matches : Calendrier des rencontres et résultats.
- Lineup : Composition d'équipe pour les matchs.

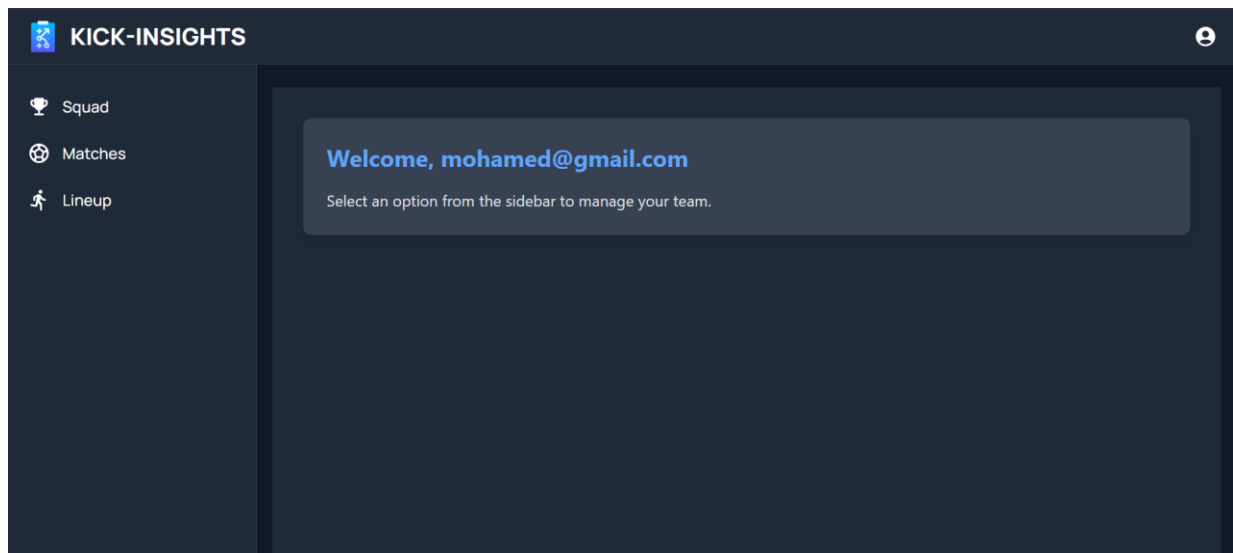


Figure 23 : Interface manager

2.1 Section des joueurs

La section **Squad** offre aux utilisateurs une vue d'ensemble structurée et intuitive de l'effectif de l'équipe nationale. Cette section est divisée en catégories selon les postes occupés par les joueurs. Elle permet de :

- **Consulter les Joueurs par Poste** : Les joueurs sont répartis en groupes distincts (Gardiens, Défenseurs, etc.), ce qui facilite leur identification. Chaque carte de joueur affiche la photo du joueur, son nom, et le club auquel il appartient.
- **Visualiser les Informations des Clubs** : Pour chaque joueur, le nom du club est accompagné de son logo, permettant un repérage visuel rapide des affiliations professionnelles.
- **Accéder aux Statistiques des Joueurs** : En cliquant sur un joueur, les utilisateurs peuvent consulter ses statistiques détaillées, telles que le nombre de matchs joués, les performances récentes, ou encore les notes attribuées lors des dernières rencontres.
- **Naviguer dans une Interface Claire et Organisée** : L'affichage par cartes dans un design sombre et épuré permet une navigation fluide et agréable. Les différentes rubriques (Squad, Matches, Lineup) dans le menu latéral assurent un accès rapide aux autres fonctionnalités de la plateforme.

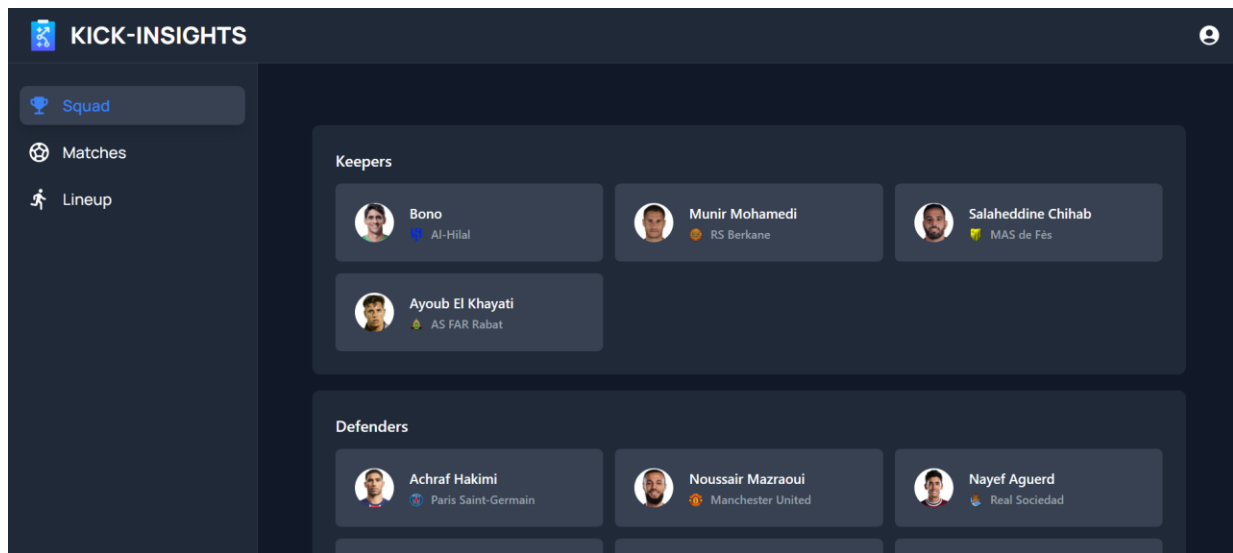


Figure 24 : Interface Joueurs

2.1.1 Section des matches d'un joueur

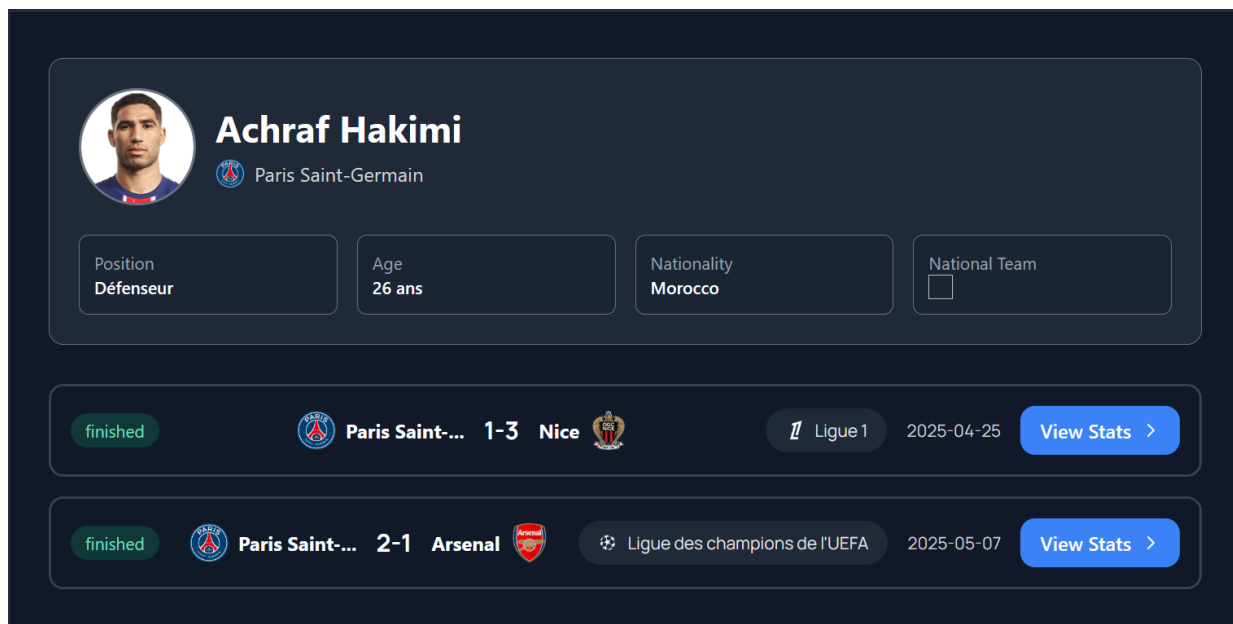


Figure 25 : Interface des matches joués par un joueur

Lorsqu'un utilisateur clique sur un joueur depuis la section **Squad**, il accède à une fiche détaillée qui regroupe l'ensemble des informations personnelles et sportives du joueur sélectionné. Cette interface permet de :

- **Consulter le Profil Complet du Joueur** : La fiche affiche la photo du joueur, son nom complet, le club actuel ainsi que des informations essentielles telles que la position sur le terrain, l'âge, la nationalité, et l'appartenance à l'équipe nationale si disponible.

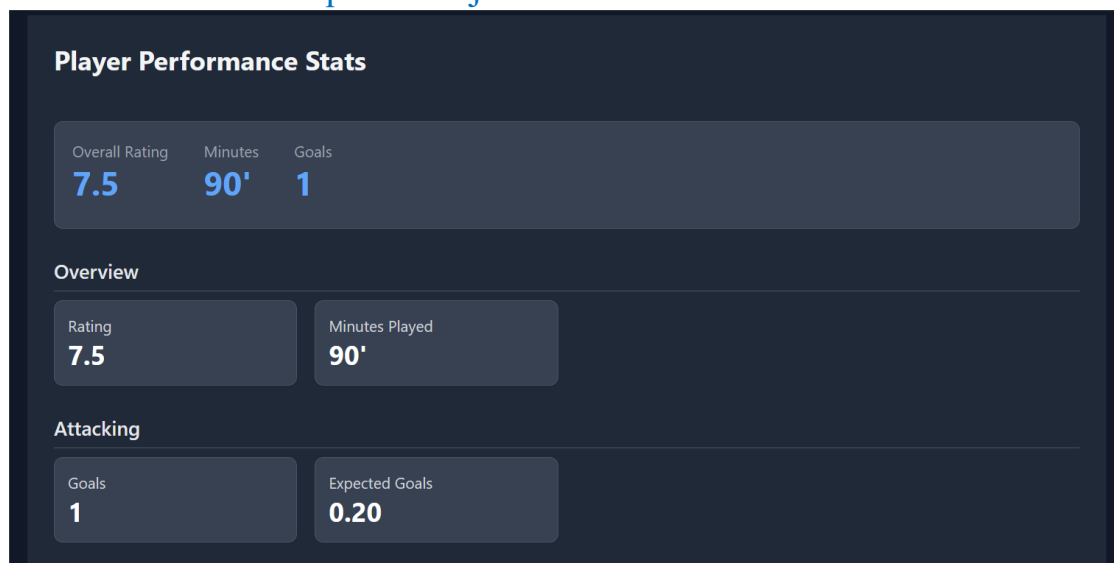
• **Visualiser l’Historique des Matches** : Les matchs récents auxquels le joueur a participé sont listés sous forme de cartes. Chaque carte indique :

- Le score final du match,
- Le nom des deux équipes,
- La compétition concernée (ex. Ligue 1, Ligue des champions de l’UEFA),
- La date de la rencontre,
- Un bouton **"View Stats"** permettant d’accéder aux statistiques détaillées de la performance du joueur durant ce match.

• **Suivre les Performances Individuelles** : En cliquant sur **"View Stats"**, l’utilisateur accède aux indicateurs précis comme les minutes jouées, les passes réussies, les duels gagnés, les tirs, etc., permettant une évaluation objective des performances.

Cette vue offre ainsi une immersion complète dans le parcours sportif du joueur, tant au niveau de ses données personnelles que de ses performances match par match.

2.1.2 Section des statistiques d’un joueur



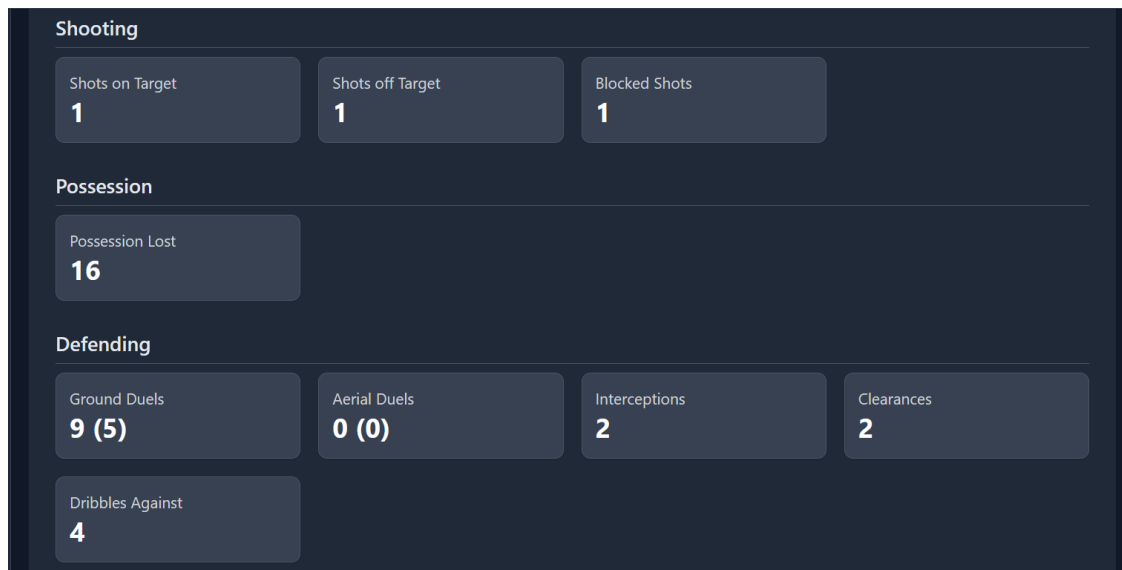


Figure 26 : Exemple des statistiques d'un joueur

Lorsqu'un utilisateur clique sur un joueur dans l'application, une fiche détaillée s'affiche, présentant ses informations personnelles (nom, club, poste, âge, nationalité) ainsi que l'historique de ses matchs. Chaque match affiche un bouton « View Stats » permettant de consulter les **statistiques individuelles du joueur pour ce match**.

2.2 Section des matches

La section dédiée aux **matches** est organisée en deux parties distinctes pour une meilleure lisibilité :

- **Recent Matches** (*Matches récents*) : cette partie affiche les derniers matchs disputés par le joueur. Chaque match est présenté avec :
 - Le nom des deux équipes avec leurs logos,
 - Le score final,
 - La date du match,
 - La compétition concernée (ex. : Ligue 1, Ligue des Champions),
 - Un bouton « **View Stats** » permettant de consulter les statistiques détaillées du joueur pour ce match.
- **Next Matches** (*Prochains matchs*) : cette section liste les rencontres à venir du joueur ou de son équipe. Les informations affichées incluent :
 - Les équipes qui vont s'affronter,
 - La date prévue du match,
 - Le nom de la compétition.

Cette séparation permet à l'utilisateur de suivre à la fois les performances passées du joueur ainsi que les événements à venir, offrant une vision complète de son calendrier sportif.

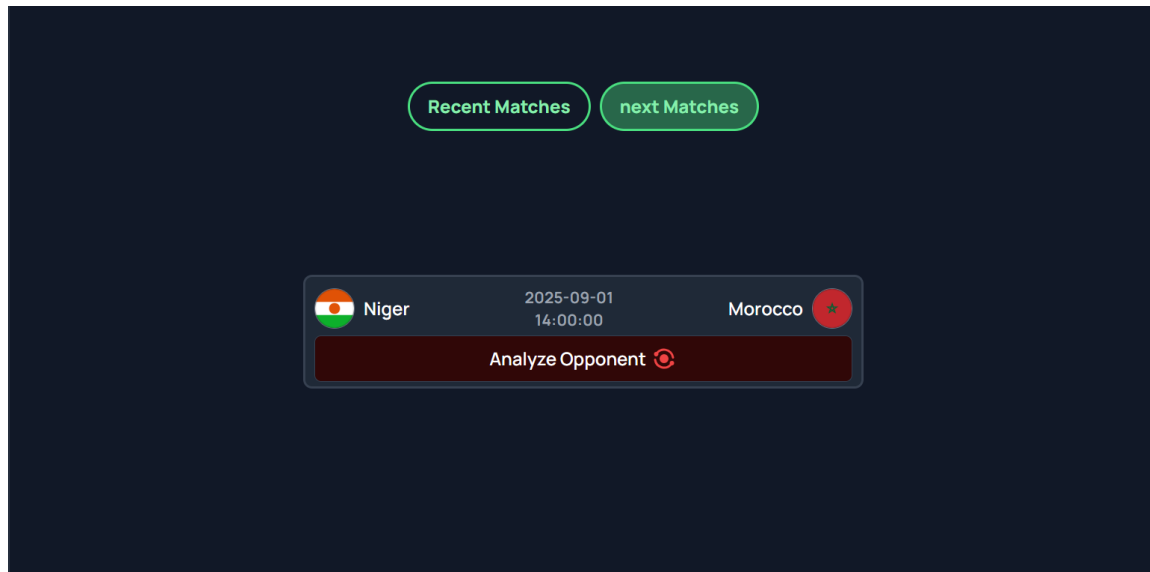


Figure 27 : Interface des prochains matchs

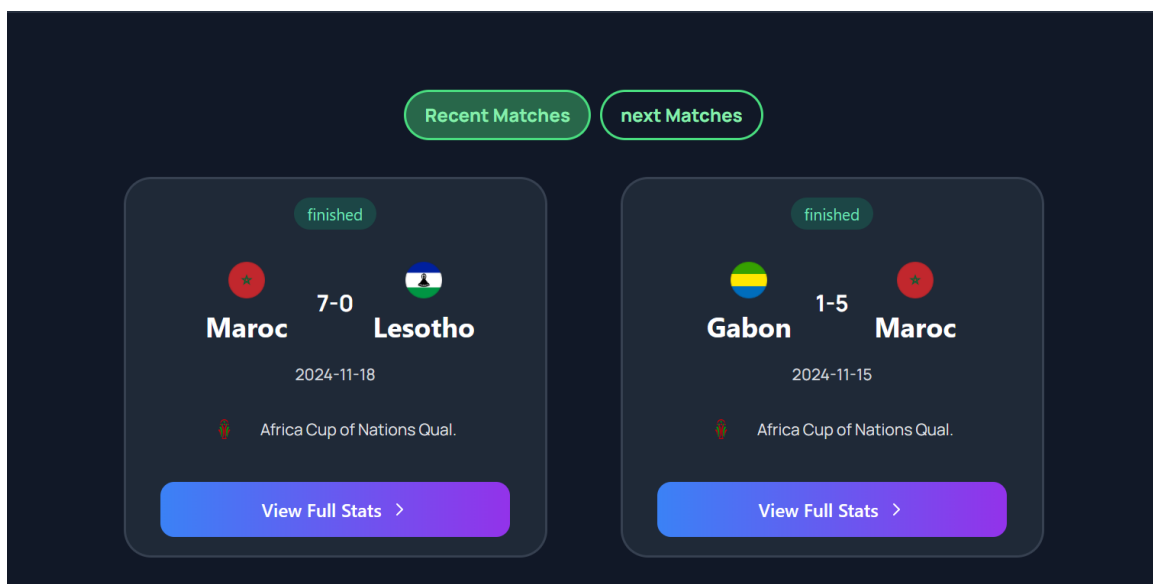


Figure 28 : Interface des matches recents

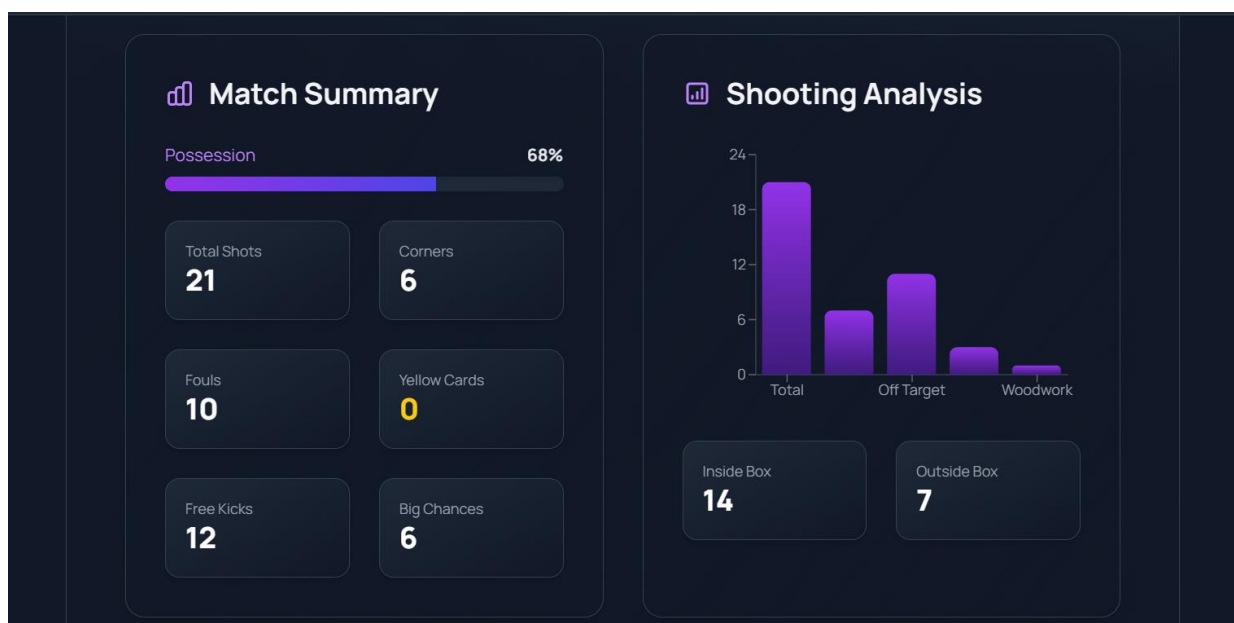
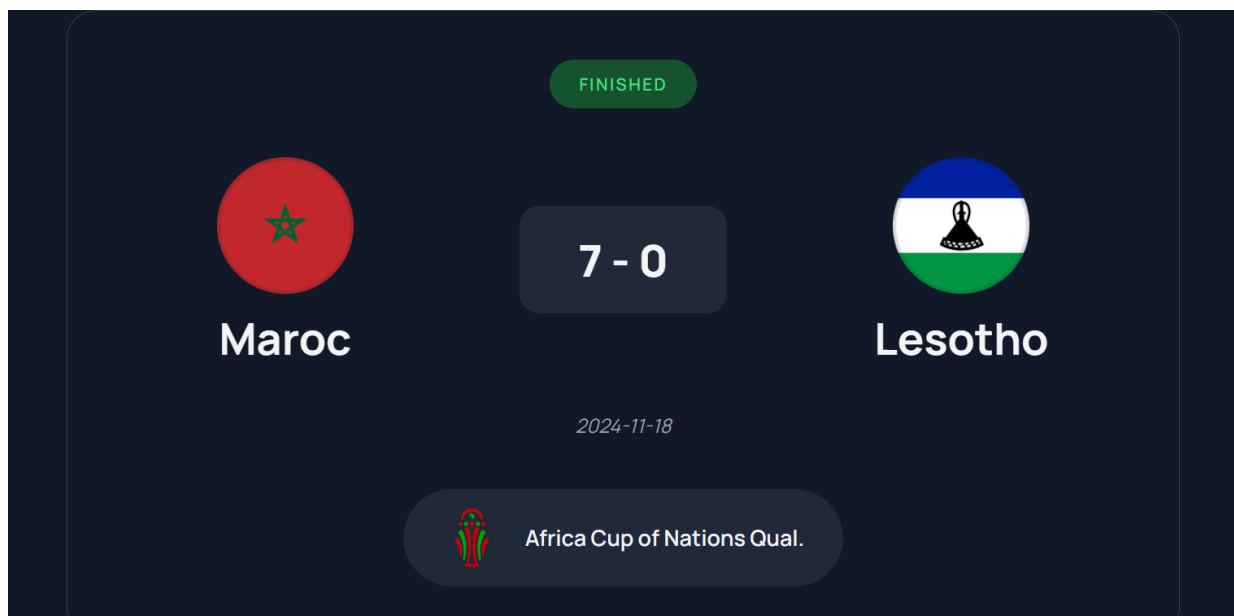
2.2.1 Matchs récents

Sur l'interface utilisateur, chaque carte de match dispose d'un bouton intitulé **"Voir les stats"**. En cliquant sur ce bouton, l'utilisateur accède aux **statistiques détaillées du match sélectionné**. Ces statistiques peuvent inclure divers éléments comme :

- La possession de balle,
- Le nombre de tirs (cadrés et non cadrés),

- Les fautes commises,
- Les corners,
- Les cartons jaunes et rouges,
- Les performances individuelles des joueurs, etc.

Cette fonctionnalité permet à l'utilisateur de consulter des informations plus approfondies sur le déroulement du match, au-delà du simple score final. L'interface assure une transition fluide vers la section des statistiques, offrant une expérience utilisateur intuitive et interactive.



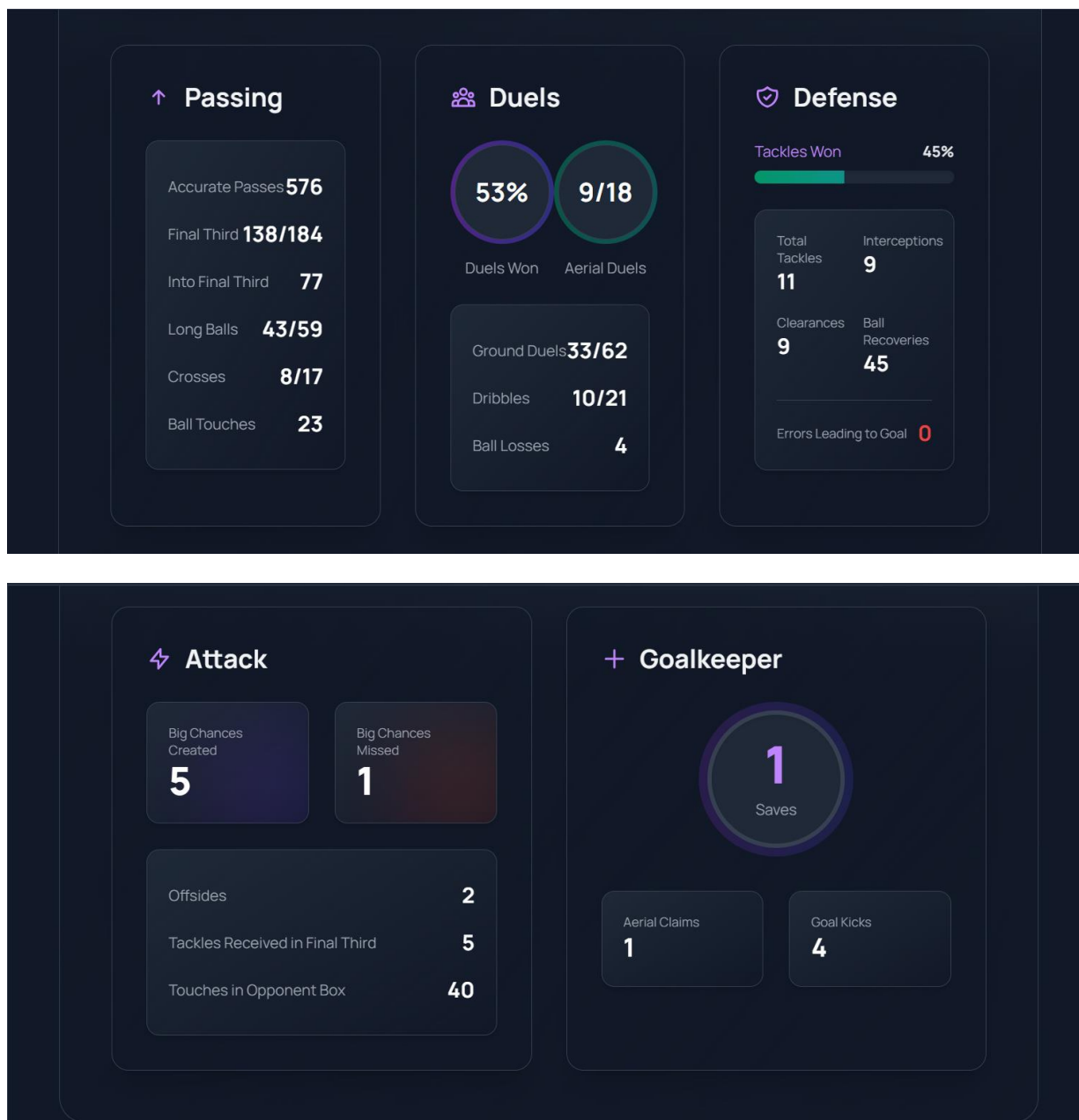


Figure 29 : Exemple des statistiques d'equipe par match

Cette page vise à offrir aux **managers, entraîneurs et analystes techniques** un accès direct à des **données précises et segmentées**, facilitant l'évaluation des performances collectives et individuelles. Elle permet ainsi de :

- Mieux comprendre le déroulement tactique du match,
- Identifier les points forts et les faiblesses de l'équipe,
- Préparer les séances d'entraînement en fonction des performances observées,
- Prendre des décisions plus éclairées pour les matchs à venir.

Grâce à cette approche analytique, la plateforme contribue à une **meilleure prise de décision stratégique** et à l'optimisation des performances sportives.

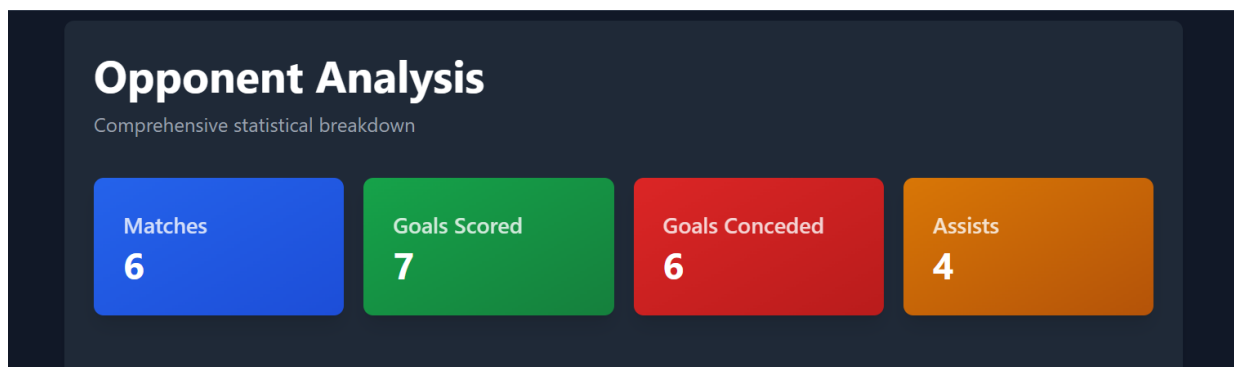
2.2.2 Prochains matchs

La section "**Prochains Matches**" est conçue pour offrir aux **managers et membres du staff technique** une **vue anticipée sur les rencontres à venir**. Elle permet de consulter rapidement la **date**, les **équipes adverses**, et le **contexte du match** (type de compétition, lieu, etc.).

Cette page joue un rôle clé dans la **préparation stratégique**, en fournissant aux utilisateurs les informations nécessaires pour :

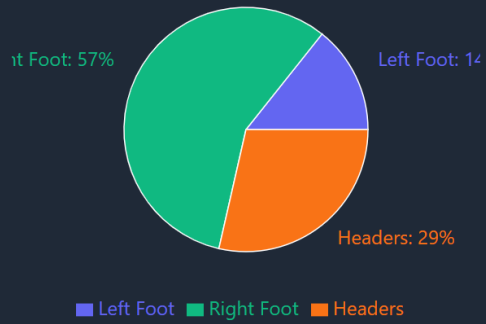
- Planifier les entraînements en fonction des caractéristiques des adversaires,
- Étudier l'historique des confrontations précédentes,
- Anticiper les enjeux tactiques du match,
- Gérer l'effectif (repos, rotations, préparation physique).

Elle sert donc de **point de départ pour l'analyse pré-match**, en complément des statistiques historiques et des performances passées, accessibles via les autres sections de la plateforme.

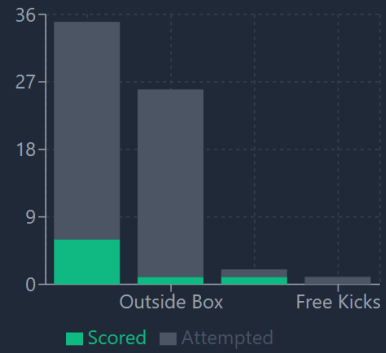


Attacking Performance

Goal Distribution



Shot Conversion Rate



Goals per Game

1.20

Big Chances per Game

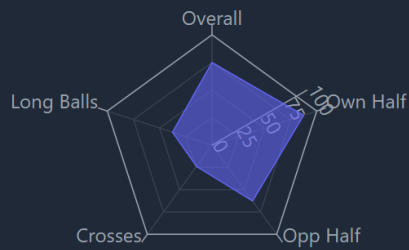
1.30

Shots on Target per Game

2.30

Passing & Possession

Passing Accuracy



Possession Distribution





Figure 30 : Exemple des statistiques d'un adversaire

La plateforme intègre une fonctionnalité essentielle pour la **préparation des matchs à venir** : l'**analyse de l'adversaire**. Cette section est spécialement conçue pour fournir aux **managers, entraîneurs et analystes** une vue détaillée des performances récentes de l'équipe adverse. Elle est divisée en cinq catégories complémentaires, permettant une lecture stratégique efficace :

1. **Opponent Analysis** : Présente une vue d'ensemble de l'équipe adverse, incluant ses résultats récents, formations utilisées, et style de jeu général.
2. **Attacking Performance** : Analyse les forces offensives de l'adversaire, telles que les zones d'attaque privilégiées, le nombre de buts marqués, les tirs cadrés et les actions décisives.
3. **Passing & Possession** : Fournit des statistiques sur la capacité de l'adversaire à garder le ballon, la précision des passes, le rythme de jeu et la construction offensive.

4. **Defensive Performance** : Évalue le comportement défensif de l'équipe adverse (interceptions, duels défensifs, organisation défensive, etc.).
5. **Discipline & Other Stats** : Affiche des données sur le fair-play et la discipline (cartons, fautes, comportement sous pression), ainsi que d'autres statistiques utiles.

Cette analyse approfondie permet au staff technique de **mieux anticiper les comportements tactiques de l'adversaire**, de préparer des plans de jeu adaptés, et d'identifier les faiblesses exploitables. Elle constitue un **outil indispensable dans la phase de préparation d'avant-match**, renforçant ainsi la prise de décision stratégique.

2.3 Section du lineup

La troisième section de la plateforme offre aux **managers et entraîneurs** la possibilité de **créer, modifier ou visualiser la composition de l'équipe** (lineup) pour un match donné. Cette fonctionnalité est conçue pour simuler l'organisation tactique sur le terrain en positionnant les joueurs selon leurs rôles et les schémas de jeu choisis.

Elle permet notamment de :

- Définir la **formation tactique** (4-3-3, 4-4-2, 3-5-2, etc.),
- Positionner chaque joueur à son **poste spécifique** (gardien, défenseur, milieu, attaquant),
- Effectuer des **ajustements en temps réel** pour tester différentes options stratégiques.

Grâce à cette interface interactive, le staff technique peut construire une approche plus structurée et flexible, facilitant la prise de décisions avant et pendant les matchs.

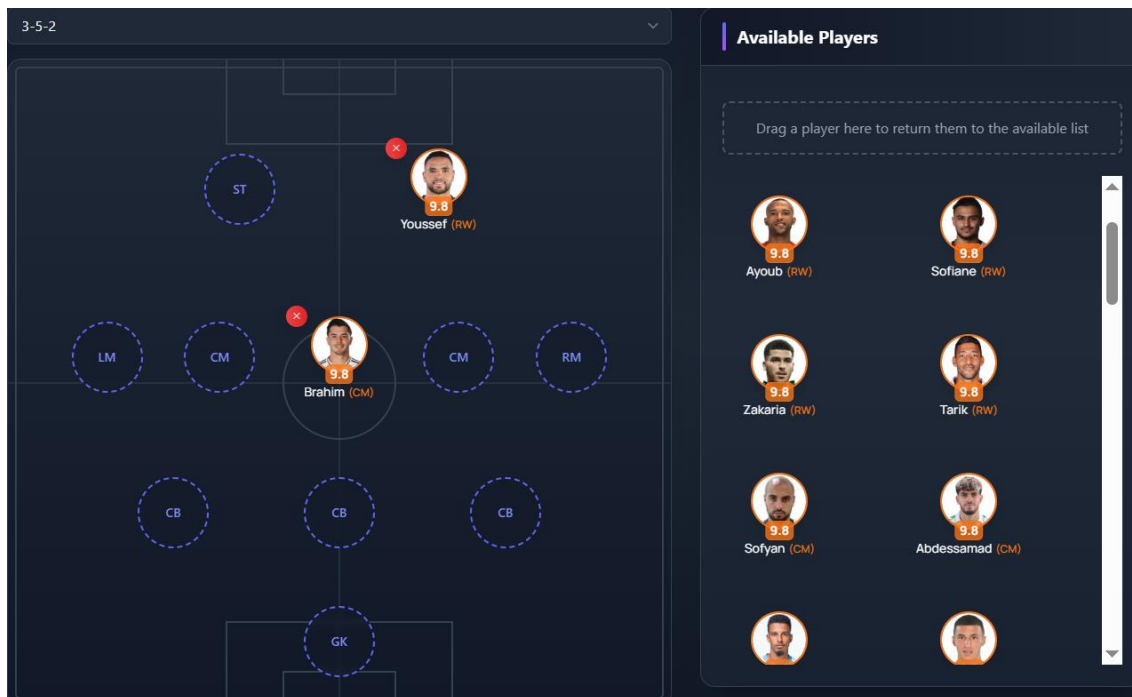


Figure 31 : Interface de composition

La section dédiée à la **composition d'équipe** permet au manager de **concevoir l'alignement tactique de ses joueurs** pour un match spécifique. Cette interface intuitive offre la possibilité de sélectionner les titulaires et les remplaçants, de positionner chaque joueur sur le terrain et de **choisir une formation adaptée à la stratégie souhaitée**.

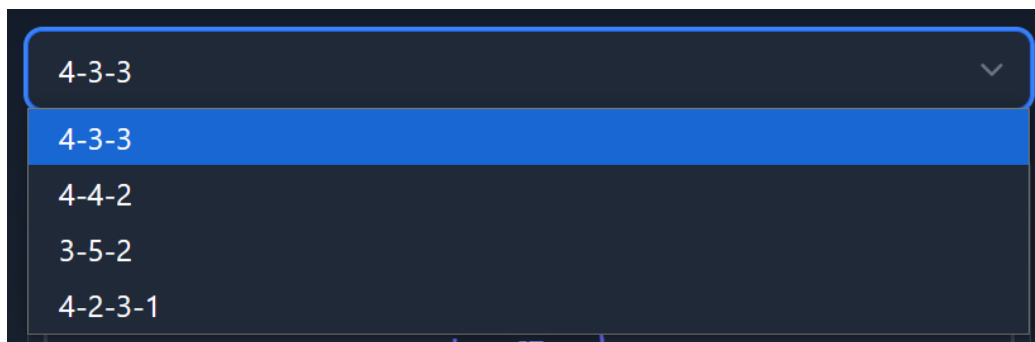


Figure 32 : Les possibles schémas tactiques

Cette flexibilité tactique est essentielle pour répondre aux exigences de chaque rencontre et renforcer la **prise de décision stratégique** en amont du match.

3. Interface admin

L'interface administrateur de la plateforme **Kick-Insights** est pensée pour offrir une **gestion complète, fluide et intuitive** des différents modules du système. Le **menu latéral** regroupe les

principales sections : **Managers, Users, Leagues, Teams, Players** ainsi qu'un outil de **Scrapping** pour la collecte automatisée des données.

Le tableau de bord central fournit des **indicateurs clés** en temps réel : nombre total de managers, joueurs, équipes, ligues et utilisateurs. Une section d'historique affiche les **actions récentes** comme les inscriptions ou ajouts de managers, avec détails précis sur la date, l'heure et l'utilisateur concerné.

Un graphique de type **diagramme circulaire** illustre la **répartition des joueurs** par catégorie ou statistique, offrant une vue globale rapide. Cette interface permet à l'administrateur de **piloter la plateforme de manière efficace** tout en assurant un suivi précis des activités et une visibilité complète sur l'écosystème de l'application.

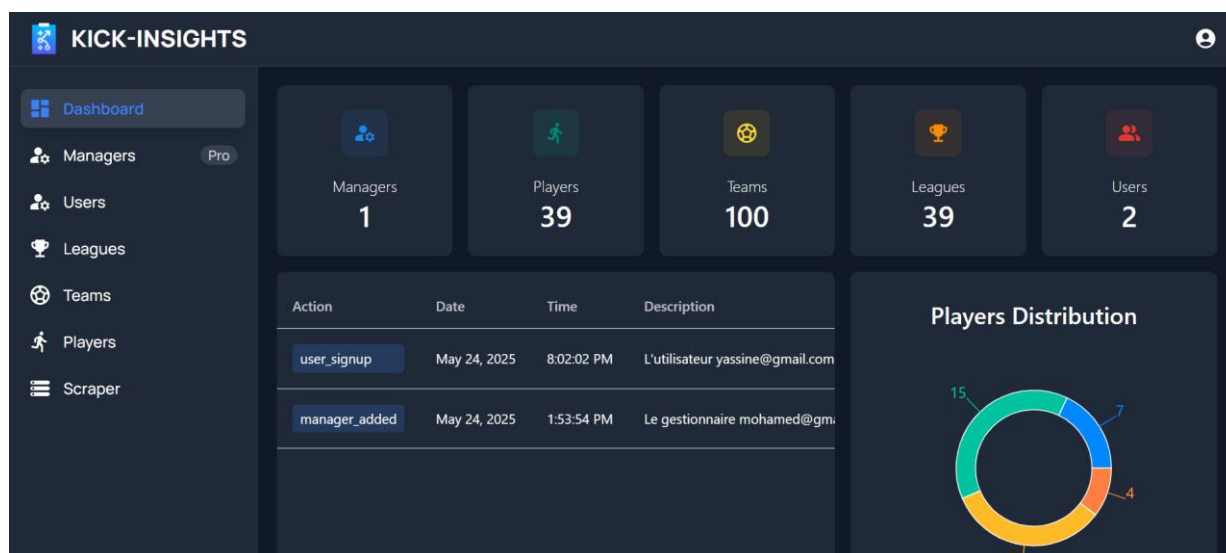


Figure 33 : Interface admin

3.1 Managers

L'une des interfaces développées dans le cadre de ce projet concerne la gestion des managers. Elle se présente sous forme d'un tableau interactif affichant une liste des managers enregistrés dans le système. Chaque ligne du tableau regroupe les informations clés d'un manager, telles que la photo de profil, le nom complet, le numéro de téléphone, la nationalité et la date de naissance. Une colonne dédiée aux actions permet d'effectuer des opérations de gestion supplémentaires (modification, suppression, etc.). L'interface intègre également un champ de recherche permettant de filtrer rapidement les résultats, ainsi qu'un bouton « Add Manager » facilitant l'ajout de nouveaux profils.



<input type="text" value="Search"/>					Add Manager
IMAGE	MANAGER NAME	PHONE NUMBER	NATIONALITY	BIRTHDAY	ACTIONS
	Alala Mohamed	+212151666677	Morocco	2010-01-12	...
	Elmaaloumi Hamza	+212222666666	Australia	2008-10-22	...
Showing 1-2 of 2					< 1 >

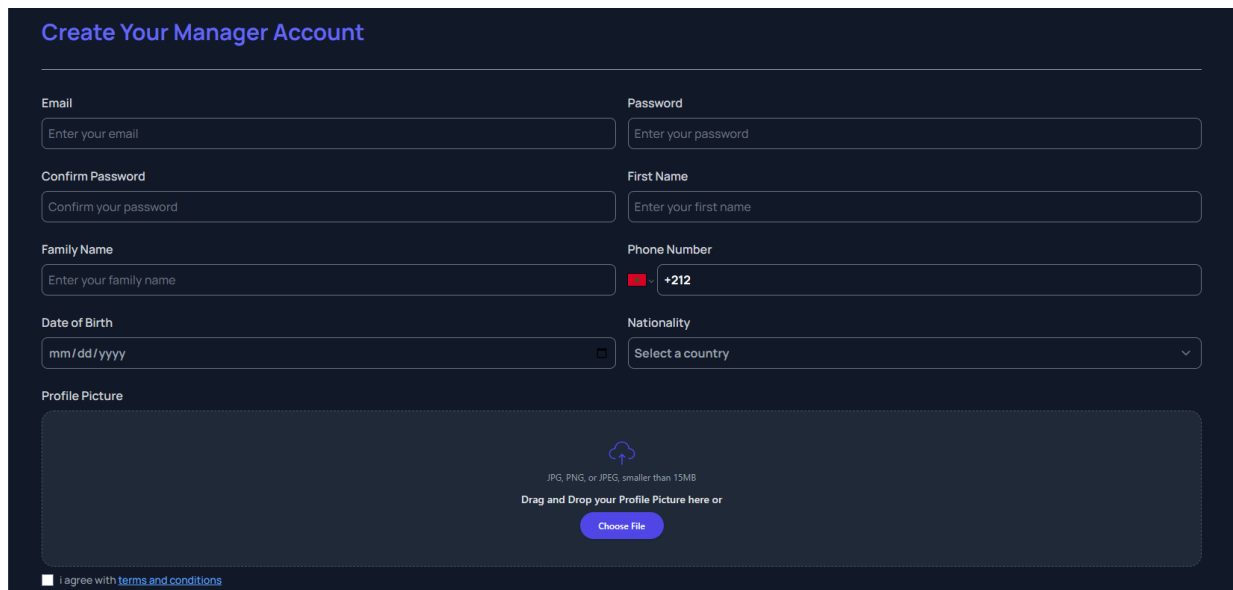
Figure 34 : Interface des managers

3.1.1 Ajouter un manager

L'administrateur peut ajouter un nouveau manager en cliquant sur le bouton « **Add Manager** ». Cette action déclenche l'ouverture d'un formulaire dans lequel les informations suivantes doivent être saisies :

- **Email**
- **Mot de passe**
- **Prénom (First name)**
- **Nom de famille (Family name)**
- **Numéro de téléphone**
- **Date de naissance (Date of birth)**
- **Nationalité**
- **Image de profil**

Une fois le formulaire complété et validé, le nouveau manager est automatiquement ajouté à la liste affichée dans le tableau de gestion.



Create Your Manager Account

Email
Enter your email

Password
Enter your password

Confirm Password
Confirm your password

First Name
Enter your first name

Family Name
Enter your family name

Phone Number
+212

Date of Birth
mm/dd/yyyy

Nationality
Select a country

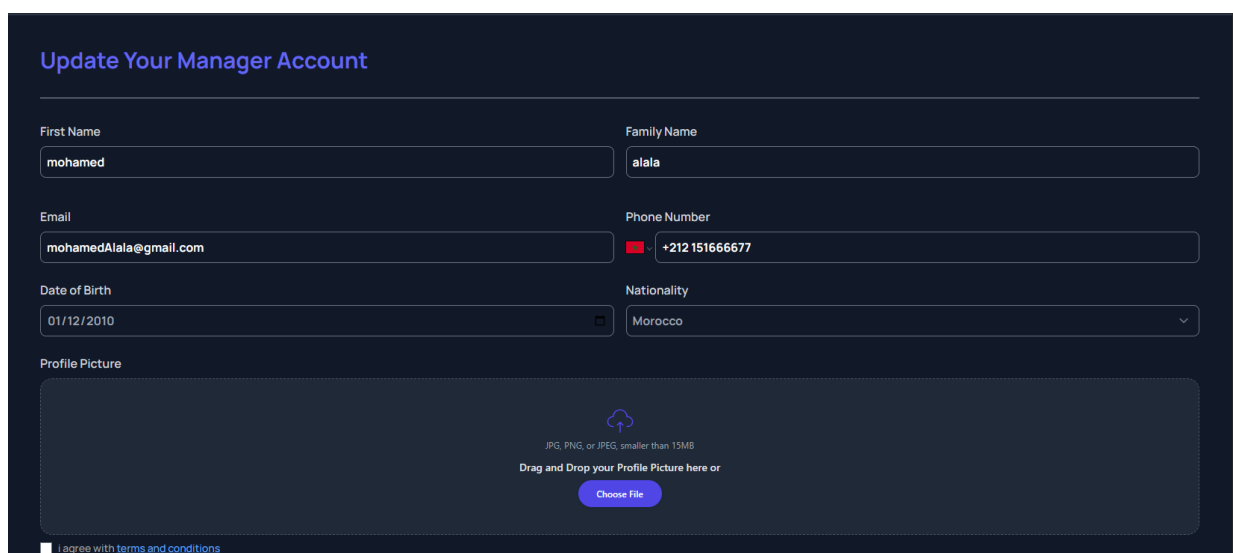
Profile Picture
JPG, PNG, or JPEG, smaller than 15MB
Drag and Drop your Profile Picture here or
Choose File

☐ I agree with [terms and conditions](#)

Figure 35 : L'ajout d' un manager

3.1.2 Modifier un manager

L'administrateur peut modifier un manager existant en cliquant sur l'option « **Modifier** » dans le menu déroulant des actions associé à un manager spécifique. Cette action ouvre un formulaire de modification pré-rempli avec les informations actuelles du manager, telles que l'email, le prénom, le nom de famille, le numéro de téléphone, la date de naissance, la nationalité et l'image de profil. L'administrateur peut ainsi mettre à jour les données souhaitées de manière simple et rapide avant de valider les modifications.



Update Your Manager Account

First Name
mohamed

Family Name
alala

Email
mohamedAlala@gmail.com

Phone Number
+212 151666677

Date of Birth
01/12/2010

Nationality
Morocco

Profile Picture
JPG, PNG, or JPEG, smaller than 15MB
Drag and Drop your Profile Picture here or
Choose File

☐ I agree with [terms and conditions](#)

Figure 36 : Modification d' un manager

3.1.3 Supprimer un manager

Pour supprimer un manager, l'administrateur peut cliquer sur l'option « **Supprimer** » dans le menu déroulant des actions associé à un manager spécifique. Une fenêtre de confirmation s'affiche afin de s'assurer que l'administrateur souhaite réellement procéder à la suppression. En confirmant cette action, le manager est supprimé définitivement de la base de données et retiré de la liste affichée dans l'interface.

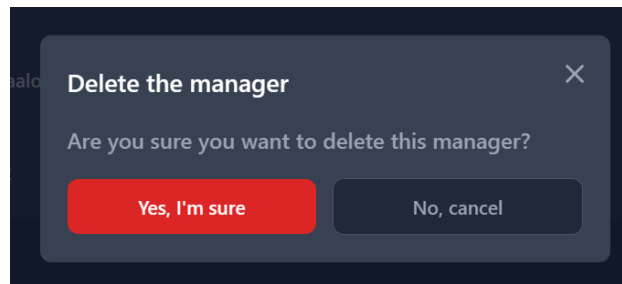


Figure 37 : Suppression d'un manager

3.2 Utilisateurs

L'interface de gestion des utilisateurs permet à l'administrateur de visualiser la liste des utilisateurs normaux enregistrés dans le système. Chaque utilisateur est représenté dans une ligne du tableau avec son **adresse email** et un **bouton de suppression** associé.

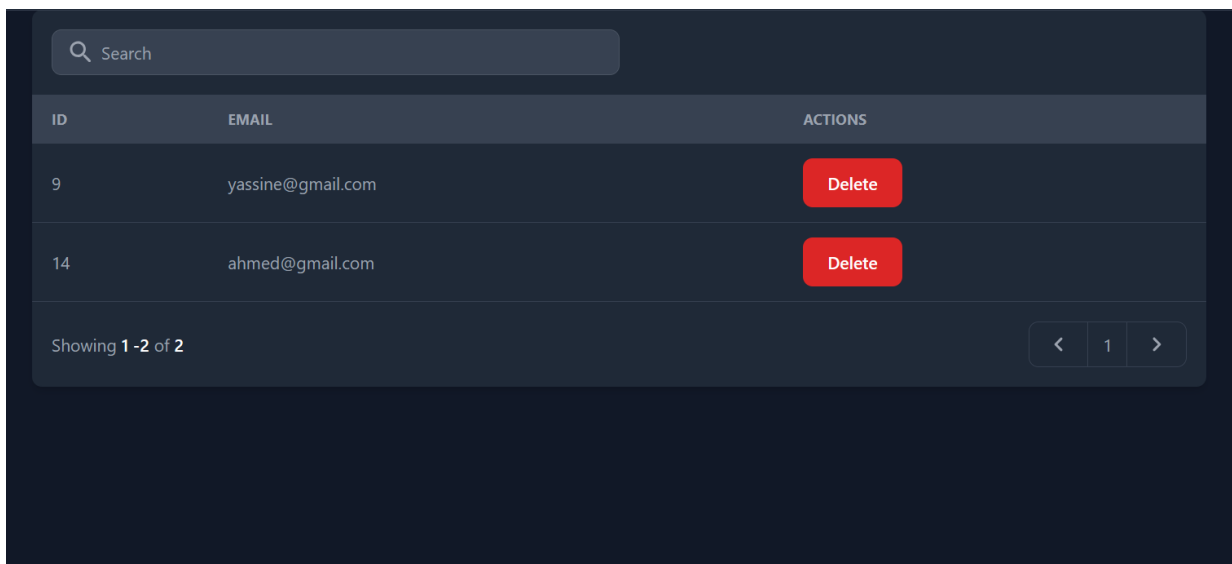


Figure 38 : Interface Users

3.2.1 Supprimer un utilisateur

Pour supprimer un utilisateur, l'administrateur peut cliquer sur le bouton « **Supprimer** » situé dans la colonne correspondante à l'utilisateur concerné. Une fenêtre de confirmation s'affiche alors afin de s'assurer que l'administrateur souhaite réellement procéder à la suppression. En

confirmant cette action, l'utilisateur est supprimé définitivement de la base de données et retiré de la liste affichée dans l'interface.

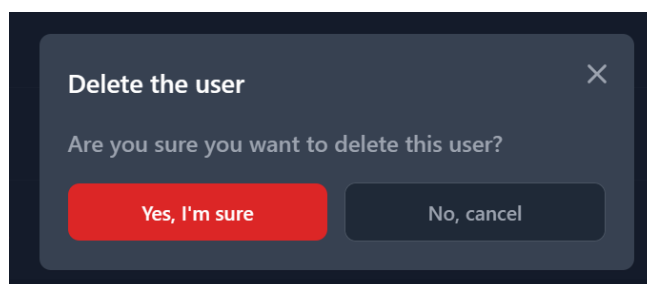


Figure 39 : Supprimer un utilisateur

3.3 Leagues

L'interface affichée ci-dessous permet à l'administrateur de **visualiser l'ensemble des ligues de football récupérées automatiquement via une opération de scraping**. Chaque ligne du tableau représente une ligue, affichant son **logo**, son **nom**. L'interface est conçue de manière claire et intuitive, facilitant la lecture rapide des informations extraites. Cette interface constitue un outil essentiel pour l'administration et la gestion structurée des données collectées automatiquement.

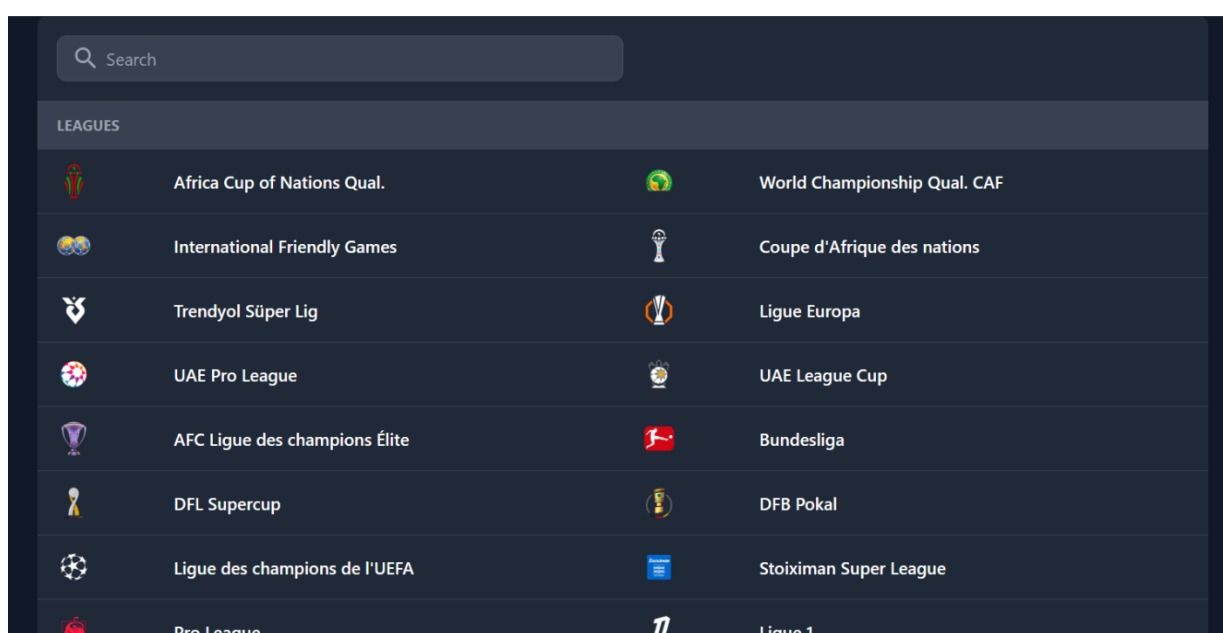


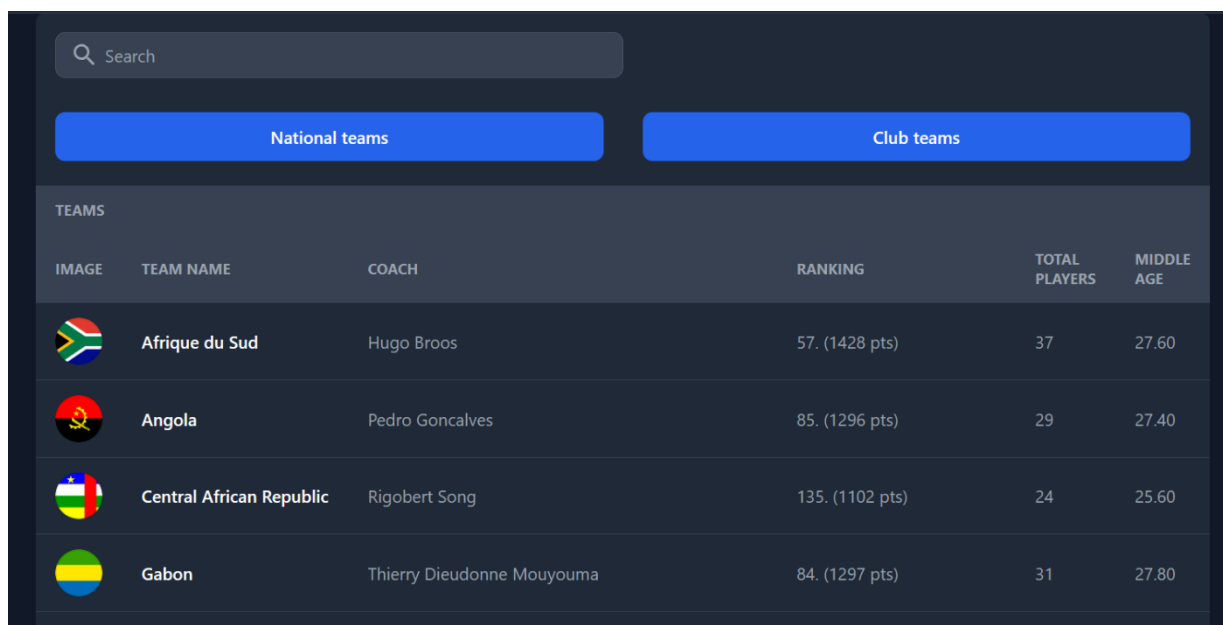
Figure 40 : Interface des leagues

3.4 Equipes

L'interface dédiée à la gestion des équipes permet à l'administrateur de **visualiser l'ensemble des équipes de football extraites automatiquement via le processus de scraping**.

3.4.1 Equipes nationales

La seconde partie de l'interface est consacrée aux équipes nationales. Toutes les équipes nationales scrappées sont présentées dans cette section distincte, facilitant ainsi l'analyse et la consultation spécifique des données concernant les sélections nationales.



The screenshot displays a web interface for national teams. At the top, there is a search bar and two tabs: 'National teams' (selected) and 'Club teams'. Below the tabs is a table titled 'TEAMS' with the following columns: IMAGE, TEAM NAME, COACH, RANKING, TOTAL PLAYERS, and MIDDLE AGE. The table lists four teams: Afrique du Sud, Angola, Central African Republic, and Gabon.





IMAGE	TEAM NAME	COACH	RANKING	TOTAL PLAYERS	MIDDLE AGE
	Afrique du Sud	Hugo Broos	57. (1428 pts)	37	27.60
	Angola	Pedro Goncalves	85. (1296 pts)	29	27.40
	Central African Republic	Rigobert Song	135. (1102 pts)	24	25.60
	Gabon	Thierry Dieudonne Mouyouma	84. (1297 pts)	31	27.80

Figure 41 : Interface des équipes nationales

3.4.2 Clubs

L'interface est divisée en une section dédiée aux clubs, où sont affichées toutes les équipes de clubs scrappées. Cette séparation permet une meilleure organisation des données et une lecture plus claire des informations relatives aux équipes de clubs.

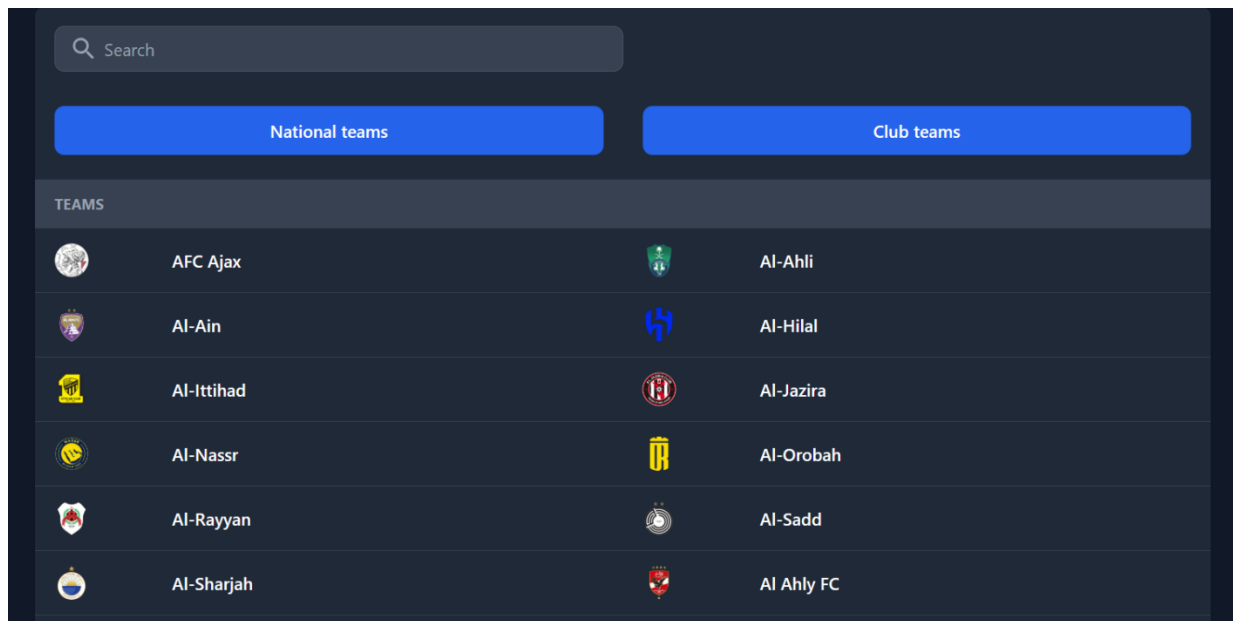


Figure 42 : Interface des Clubs

3.5 Joueurs

L'interface dédiée à la gestion des joueurs permet à l'administrateur de **visualiser l'ensemble des joueurs marocains extraits automatiquement via le processus de scraping.**

<input type="text" value="Search"/> <input type="button" value="Filter"/>					
IMAGE	PLAYER NAME	AGE	POSITION	CLUB	CLUB NAME
	Youssef En-Nesyri	27	Ailier		Fenerbahçe
	Soufiane Rahimi	28	Ailier		Al-Ain
	Amine Adli	24	Ailier		Bayer 04 Leverkusen
	Ayoub El Kaabi	31	Ailier		Olympiacos FC
	Sofiane Boufal	31	Ailier		Royale Union Saint-Gilloise
	Zakaria Aboukhlal	25	Ailier		Toulouse

	Tarik Tissoudali	31	Ailier		Khor Fakkan
	Brahim Díaz	25	Milieu de terrain		Real Madrid
	Abdessamad Ezzalzouli	23	Milieu de terrain		Real Betis
	Azzedine Ounahi	24	Milieu de terrain		Panathinaikos

Showing 1 -10 of 36

< 1 2 3 4 >

Figure 43 : Interface des joueurs

3.5 Scraper

Une autre section de l'interface comprend des boutons dédiés permettant de lancer manuellement le scraping des données. Chaque bouton correspond à une partie spécifique (clubs ou équipes nationales), offrant ainsi un contrôle précis et indépendant sur le déclenchement du processus de récupération des informations.

Scraping the Leagues

- ✓ Make sure your connection is stable before proceeding.
- ✓ The scraping process may take a few minutes.
- ✓ Ensure there is no ongoing heavy network usage.
- ✓ Close other unnecessary applications or browser tabs to speed up the process.
- ✓ The server may be temporarily unavailable at times, so please try again later if needed.

☐ I have reviewed and ensured all the requirements are met before proceeding.

Scrape the Leagues

Scraping the Teams

- ✓ Make sure your connection is stable before proceeding.
- ✓ The scraping process may take a few minutes.
- ✓ Ensure there is no ongoing heavy network usage.
- ✓ Close other unnecessary applications or browser tabs to speed up the process.
- ✓ The server may be temporarily unavailable at times, so please try again later if needed.

☐ I have reviewed and ensured all the requirements are met before proceeding.

Scrape the Teams

Figure 44 : Interface des boutons de scraping

4. Conclusion

Le développement de cette plateforme d'analyse des performances footballistiques a été structuré de manière à centraliser les données des joueurs et à en faciliter l'exploitation par les différents intervenants. Chaque étape du projet – du web scraping à l'affichage des informations dans une interface intuitive – a été pensée pour offrir une solution fonctionnelle, claire et accessible. Les rôles distincts attribués aux administrateurs et aux entraîneurs permettent une gestion efficace des données, tout en garantissant une vision personnalisée et pertinente pour

chaque utilisateur. Grâce à ce système, les décisions sportives peuvent désormais s'appuyer sur des informations fiables et actualisées.

IV. Module de Recommandation de Joueurs

Étudier les similitudes entre les joueurs lors d'un processus de recrutement est une étape cruciale, car cela permet aux équipes d'identifier des remplaçants ou des cibles potentielles possédant des caractéristiques, compétences et styles de jeu similaires au joueur à remplacer. En tenant compte de la similarité entre joueurs, les clubs peuvent assurer une transition plus fluide et maintenir une cohérence dans le style de jeu collectif.

Il existe plusieurs formules et techniques pour évaluer la similarité entre joueurs. L'approche la plus courante repose sur l'analyse statistique, où les performances des joueurs — telles que le nombre de buts marqués, les passes décisives, la précision des passes ou encore les actions défensives — sont comparées. Ces données permettent de calculer des scores de similarité selon différents indicateurs.

De plus, des algorithmes d'apprentissage automatique peuvent être utilisés pour analyser de grandes quantités de données sur les joueurs et détecter des schémas indiquant une ressemblance. Ces algorithmes prennent en compte des facteurs comme le poste de jeu, les compétences techniques, l'âge, le temps de jeu ou encore la valeur marchande. Toutefois, dans notre cas, nous ne disposons que de données descriptives basées sur les performances, le temps de jeu, le poste, l'équipe, l'âge, le salaire et la valeur marchande. Des éléments tels que la conscience tactique ne peuvent pas encore être quantifiés par les données, et les attributs physiques nécessitent des données de suivi (tracking data) via des outils de détection d'objets ou des dispositifs portables comme le GPS, auxquels nous n'avons pas accès.

L'objectif était donc de construire un système capable de générer une liste restreinte de joueurs, représentant les profils les plus similaires à un joueur sortant ou ciblé. Concrètement, le système de recommandation de joueurs repose sur les étapes suivantes :

- Sélection du joueur que l'on souhaite remplacer (par exemple un joueur sur le départ).
- Le système génère une liste classée des meilleurs candidats susceptibles de le remplacer.

1. La création des caractéristiques des joueurs

Dans le processus de création des caractéristiques des joueurs (*player features*), plusieurs métriques ont été utilisées afin de représenter au mieux la performance d'un joueur sur le terrain. Ces métriques incluent notamment les tirs, les passes décisives, les dribbles réussis, les centres, les entrées dans la surface de réparation, le total de passes, les passes courtes (< 32 m), les passes longues (≥ 32 m), les passes dans le tiers offensif, ainsi que les actions défensives réalisées dans le tiers défensif, médian et offensif.

Ces indicateurs permettent de capturer différents aspects du jeu d'un joueur, qu'il s'agisse de ses aptitudes offensives, de sa capacité à créer des occasions, de sa qualité de passe ou encore de ses interventions défensives. En exploitant ces données, le système de recommandation peut évaluer et comparer les performances individuelles afin d'identifier les joueurs dont le profil correspond le mieux aux besoins de remplacement ou de recrutement au sein d'une équipe.

2. Pondération personnalisée des caractéristiques

Au lieu de s'appuyer sur une seule métrique prédictive, nous construisons un score global en utilisant une pondération personnalisée, permettant à l'utilisateur de définir l'importance relative de chaque métrique. Par exemple, pour un ailier à vocation offensive, les buts et les passes décisives peuvent être considérés comme plus importants que les actions défensives.

Chaque cible prédite est normalisée selon la moyenne, puis multipliée par un poids défini par l'utilisateur (entre 0 et 1). Le score final, compris entre 0 et 1, est obtenu en faisant la somme des scores pondérés, divisée par la somme des pondérations. En utilisant des pondérations correspondant aux points forts principaux de la position attaquant, il est ainsi possible de générer une liste restreinte de 10 joueurs correspondant aux critères spécifiés.

```
POSITION_WEIGHTS_ARRAY = {  
  "ST": [10, 1, 20, 8, 6, 3, 2, 6, 2, 2, 9, 6, 3, 5, 2, 2, 1, 1, 1, 1, 1, 1, 1, 8, 4, 1, 1, 1, 1],  
  "RW": [9, 2, 20, 7, 7, 4, 3, 9, 6, 3, 7, 5, 3, 10, 4, 2, 1, 1, 2, 2, 2, 2, 7, 3, 1, 1, 1, 1],  
  "LW": [9, 2, 20, 7, 7, 4, 3, 9, 6, 3, 7, 5, 3, 10, 4, 2, 1, 1, 2, 1, 2, 2, 2, 7, 3, 1, 1, 1, 1],  
  "CM": [8, 3, 4, 4, 8, 7, 9, 8, 3, 5, 3, 2, 1, 4, 6, 4, 3, 2, 3, 3, 5, 4, 2, 5, 2, 1, 1, 1, 1],  
  "CDM": [7, 3, 3, 3, 7, 5, 7, 5, 3, 4, 2, 1, 1, 3, 7, 6, 5, 5, 5, 4, 10, 9, 4, 4, 2, 1, 1, 1, 1],  
  "RB": [6, 4, 2, 2, 5, 6, 6, 4, 7, 5, 3, 2, 2, 4, 8, 7, 6, 5, 4, 6, 9, 9, 5, 3, 2, 1, 1, 1, 1],  
  "LB": [6, 4, 2, 2, 5, 6, 6, 4, 7, 5, 3, 2, 2, 4, 8, 7, 6, 5, 4, 6, 9, 9, 5, 3, 2, 1, 1, 1, 1],  
  "CB": [6, 4, 1, 1, 2, 5, 4, 3, 2, 3, 1, 1, 1, 1, 7, 9, 5, 5, 5, 10, 9, 8, 4, 1, 1, 1, 1, 1],  
  "GK": [5, 1, 1, 1, 1, 2, 3, 2, 1, 1, 1, 1, 1, 1, 2, 3, 2, 1, 1, 7, 3, 2, 2, 1, 1, 10, 9, 8, 7]  
}
```

Figure 45 : pondérations personnalisées pour chaque position

3. Cosine Similarity

La similarité cosinus est une méthode mathématique permettant de mesurer à quel point deux ensembles d'informations sont similaires. En termes simples, elle nous aide à comprendre la

relation entre deux éléments en examinant la « direction » qu'ils prennent, plutôt qu'en comparant simplement leurs valeurs individuelles.

Imaginez que vous êtes passionné de lecture et que vous avez noté trois livres : « *Le Mystère Lunaire* », « *Les Secrets de l'Océan* » et « *Le Vol du Phénix* », sur une échelle de 1 à 5. Votre ami a également noté ces mêmes livres selon la même échelle :

Reviewer	The Lunar Mystery	Secrets of the Ocean	Flight of the Phoenix
You	5	3	4
Your Friend	4	2	4

Figure 46 : Simple représentation des évaluations de livres

Vos évaluations respectives peuvent être représentées sous forme de listes ou, en termes mathématiques, de « vecteurs », soit $[5, 3, 4]$ pour vous et $[4, 2, 4]$ pour votre ami.

Avez-vous des évaluations similaires ? On peut regarder ces listes et dire de manière qualitative « oui, elles sont assez proches », ou bien utiliser la similarité cosinus pour obtenir une mesure quantitative ! Nous reviendrons à cet exemple, mais il est important de savoir que la similarité cosinus est un concept aux applications très vastes, notamment dans les moteurs de recherche, le traitement du langage naturel et les systèmes de recommandation.

La similarité cosinus permet de comprendre les relations entre des données sans se laisser submerger par les détails précis que représente chaque point. Elle offre aussi la possibilité de comparer rapidement des informations contenant des dizaines, des centaines, voire des milliers d'éléments.

La similarité cosinus quantifie la similarité entre deux vecteurs en mesurant le cosinus de l'angle entre eux. Cela est particulièrement utile en analyse de texte, où les textes sont convertis en vecteurs. Chaque dimension du vecteur représente un mot du document, et sa valeur indique la fréquence ou l'importance de ce mot.

Lors du calcul de la similarité cosinus, on commence par trouver le **produit scalaire** des deux vecteurs. Ce produit mesure dans quelle direction les vecteurs sont alignés. Ensuite, on calcule les **normes** (ou longueurs) de chaque vecteur. La similarité cosinus est le produit scalaire divisé par le produit des normes des deux vecteurs.

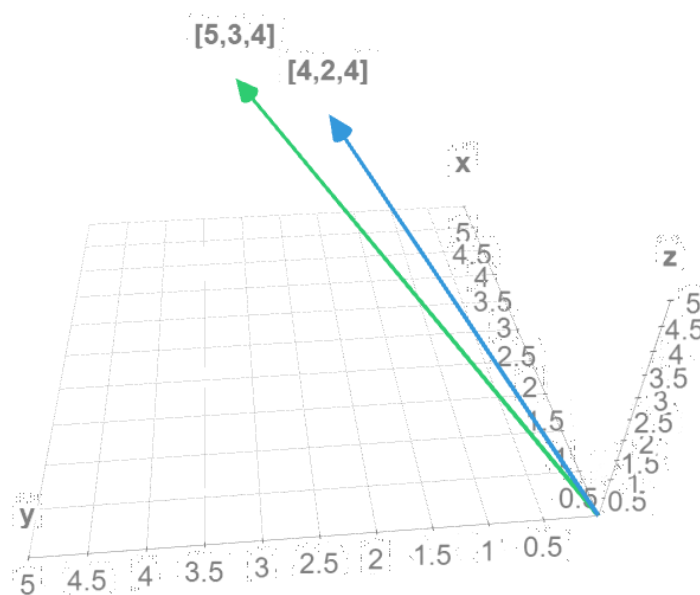


Figure 47 : Visualisation des deux vecteurs d'exemple précédent


Vos évaluations : $[5, 3, 4]$, Les évaluations de votre ami : $[4, 2, 4]$

En utilisant la similarité cosinus, nous pouvons quantifier à quel point ces vecteurs sont similaires. La similarité cosinus renverra une valeur comprise entre -1 et 1 ; une valeur proche de 1 indique une plus grande similarité. Dans notre exemple, le calcul de la similarité cosinus nous donne une valeur de 0.9899, ce qui suggère que vous et votre ami avez des goûts très similaires en matière de livres.

4. Introduction à la représentation vectorielle

Dans ce projet, chaque joueur de football est représenté par un vecteur numérique basé sur ses performances — telles que les buts, les passes décisives, les passes clés, les duels gagnés, les dribbles réussis, et d'autres statistiques pertinentes, généralement normalisées sur 90 minutes. Par exemple, un attaquant comme En-nesiry Youssef peut être représenté par un vecteur reflétant son style de jeu et son impact statistique.

Cette transformation permet d'analyser et de comparer les joueurs à l'aide d'opérations mathématiques dans un espace de grande dimension. Une fois tous les joueurs convertis en vecteurs, il devient possible d'évaluer la similarité entre deux joueurs en mesurant la proximité de leurs vecteurs.



$$= [0.13, 1.54, \dots, 0.86]$$

Figure 48 : Représentation des statistiques de joueur sous forme de vecteur

Par la suite, le processus est répété pour l'ensemble des joueurs, générant des vecteurs pour chacun d'eux à partir de leurs propres métriques. Ces vecteurs sont ensuite utilisés pour entraîner le modèle Word2Vec. Cela permet d'identifier les vecteurs similaires à un vecteur cible, et ainsi de découvrir des joueurs présentant des similitudes avec un joueur donné.

5. Interface de recommandation

Lorsque le manager clique sur une position dans le schéma tactique (par exemple, un poste dans une formation "4-3-3"), une fenêtre contextuelle s'ouvre intitulée "**Position similarities**". Cette fenêtre affiche une liste de joueurs recommandés en fonction de leur **proximité de poste ou de style de jeu** avec le poste sélectionné.

Fonctionnalités visibles :

- **Affichage des joueurs similaires** sous forme de liste.
- Chaque joueur est accompagné :
 - D'une **photo de profil**.
 - De son **nom complet**.
 - D'un **score de similarité** numérique (allant de 0.0000 à 1.0000), indiquant à quel point le joueur est adapté ou proche du poste sélectionné.
- Les joueurs sont **classés par score décroissant**, du plus similaire au moins similaire.

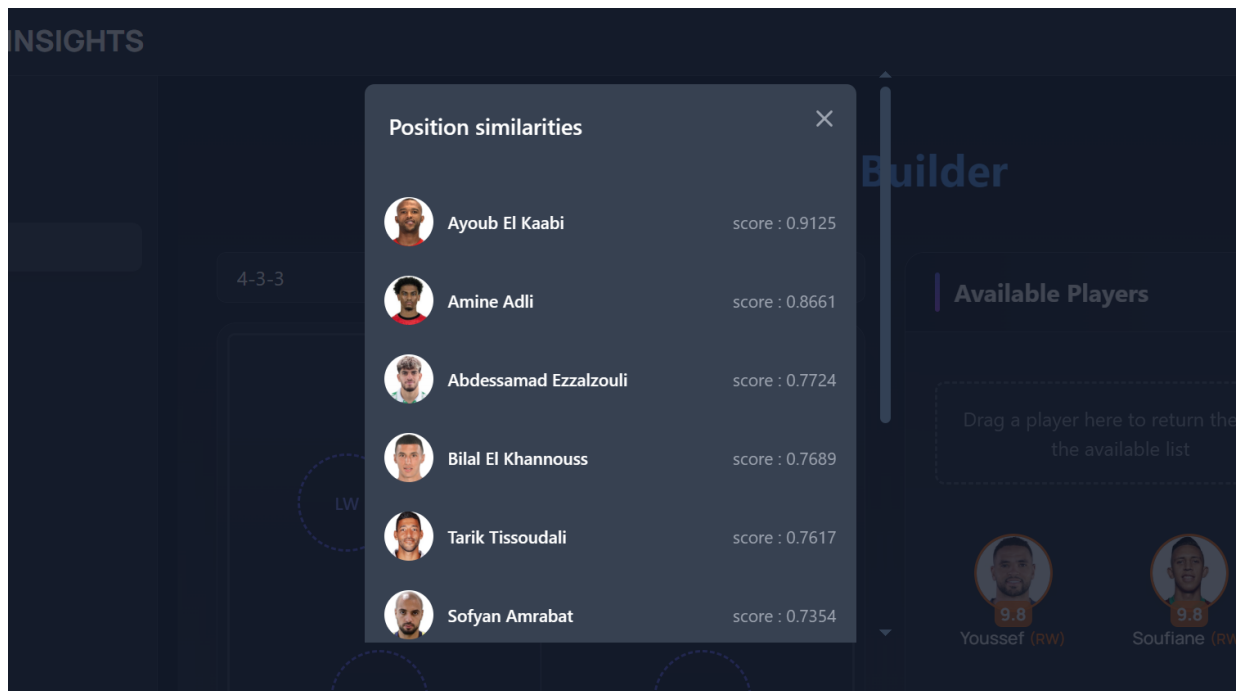


Figure 49 : Interface de recommandation

6. Conclusion

Ce chapitre a présenté le module de recommandation de joueurs, un élément clé de l'application permettant d'assister les entraîneurs dans leurs choix tactiques. À travers la création et la pondération des caractéristiques des joueurs, ce module permet d'établir des profils personnalisés adaptés aux besoins spécifiques d'un match ou d'un poste. L'usage de la similarité cosinus et de la représentation vectorielle des données rend possible une comparaison fine entre profils, assurant ainsi des recommandations pertinentes. Enfin, l'interface dédiée offre une expérience utilisateur fluide et accessible, facilitant l'exploitation de ces recommandations dans un cadre réel. Ce module illustre parfaitement l'apport de l'intelligence computationnelle dans l'analyse et la gestion des effectifs sportifs.

Chapitre 4 : Conclusion du projet

I. Améliorations techniques

- **Compétences en Développement Front-End** : L'utilisation de frameworks modernes comme React a permis d'améliorer considérablement mes compétences en développement front-end. La création de composants réutilisables, la gestion de l'état de l'application et l'intégration de bibliothèques tierces ont été des aspects cruciaux de ce projet.
- **Gestion des Formulaires et Validation** : J'ai développé une expertise dans la gestion des formulaires complexes, y compris la validation en temps réel et la gestion des erreurs. L'utilisation de hooks comme useForm a facilité la manipulation des données du formulaire et a permis une expérience utilisateur plus fluide.
- **Web Scraping** : J'ai développé des compétences en extraction automatisée de données à l'aide de bibliothèques comme Selenium. Cela m'a permis d'intégrer des données externes pertinentes dans l'application et de mieux comprendre les enjeux liés à la collecte de contenu en ligne.
- **Multilinguisme et Localisation** : L'intégration de la bibliothèque de traduction react-i18next a permis de rendre l'application accessible à un public multilingue. Cela m'a aidé à comprendre l'importance de la localisation dans le développement d'applications globales.
- **Expérience Utilisateur et Design UI** : En travaillant sur l'interface utilisateur, j'ai amélioré mes compétences en design UI, en m'assurant que l'application est à la fois esthétique et fonctionnelle. L'importance de l'accessibilité et de la convivialité a été prise en compte tout au long du projet.

II. Améliorations personnelles et professionnelles

- **Gestion de Projet** : La planification et la mise en œuvre de ce projet m'ont permis d'affiner mes compétences en gestion de projet. J'ai appris à définir des objectifs clairs, à gérer les délais et à coordonner les différentes étapes du développement.
- **Résolution de Problèmes** : J'ai rencontré et surmonté divers défis techniques tout au long du projet. Cela m'a aidé à développer une approche méthodique pour la résolution de problèmes, en utilisant des outils de debug et en recherchant des solutions innovantes.

- **Apprentissage Continu** : Le projet m'a poussé à rester à jour avec les technologies modernes et les meilleures pratiques du développement web. Cet engagement envers l'apprentissage continu est essentiel pour rester compétitif dans le domaine technologique en constante évolution.

Webographie

- [1] Sofascore, « Résultats et statistiques sportives », <https://www.sofascore.com/>.
- [2] W3Schools, « Tutoriels web et documentation HTML/CSS/JavaScript », <https://www.w3schools.com/>.

