

**BTS SERVICES INFORMATIQUES AUX ORGANISATIONS**

**Option SLAM — Épreuve E6**

Session 2026

---

**Situation n° 2/2 :**

**Application mobile pour le club de football  
Montpellier Arceaux**

---

Flutter/Dart

API REST Symfony 8 · JWT · Firebase Cloud Messaging

Android & iOS

**EPSI Montpellier  
Yannis EI HAJAOUI**

# Table des matières

---

## Table des matières

|                                                             |    |
|-------------------------------------------------------------|----|
| Table des matières .....                                    | 2  |
| Le contexte.....                                            | 3  |
| Le client.....                                              | 5  |
| Le réseau et l'infrastructure.....                          | 6  |
| Architecture technique retenue .....                        | 6  |
| Sécurité du réseau .....                                    | 6  |
| Les fonctionnalités.....                                    | 7  |
| Application web Symfony .....                               | 7  |
| La démarche projet.....                                     | 13 |
| La base de données et les diagrammes UML.....               | 22 |
| Schéma Base de données .....                                | 22 |
| Diagramme de classes (UML) .....                            | 23 |
| Quelques fonctionnalités expliquées .....                   | 24 |
| 1. Authentification et gestion des rôles .....              | 24 |
| 2. Gestion des joueurs et conformité RGPD .....             | 25 |
| 3. Système de convocations avec courriels automatiques..... | 26 |
| 4. Planning FullCalendar et gestion des séances.....        | 26 |
| 5. Gestion du versioning et suivi du code source .....      | 27 |
| Axes d'amélioration .....                                   | 28 |
| Conclusion .....                                            | 30 |

## Le contexte

---

Le club de football Montpellier Arceaux est une association sportive loi 1901 fondée en 1990, affiliée à la Fédération Française de Football (FFF) et à la Ligue Occitanie de Football. Implanté dans le quartier des Arceaux à Montpellier (34), le club accueille chaque saison plus de 300 licenciés répartis sur 13 catégories d'âge : U6, U7, U8, U9, U10, U11, U12, U13, U15, U17, U18, Sénior Féminine et Seniors. Véritable acteur du football amateur régional, Montpellier Arceaux est reconnu pour son encadrement structuré, la qualité de ses entraîneurs dont plusieurs sont titulaires de diplômes UEFA ainsi que pour son engagement fort dans la formation des jeunes joueurs.

En tant qu'association loi 1901, le club fonctionne sur un modèle où bénévolat et engagement associatif coexistent afin d'assurer le bon déroulement des activités sportives et administratives. La gouvernance est assurée par un bureau composé d'un président, d'un trésorier et d'un secrétaire, chacun responsable d'un périmètre précis, allant de la coordination générale au suivi financier, en passant par l'organisation des événements. Ce bureau est épaulé par cinq entraîneurs diplômés, chacun en charge d'une équipe, ainsi que par des bénévoles qui assurent des missions essentielles comme l'accueil lors des matchs à domicile, la gestion du matériel sportif, l'organisation des déplacements et la communication avec les familles.

Chaque saison sportive, le club doit traiter et mettre à jour un volume important d'informations administratives et sportives. Cela comprend notamment les inscriptions et les renouvellements de licences FFF pour chaque joueur, le suivi des cotisations dont les montants varient selon les catégories d'âge, l'affectation des joueurs dans les différentes équipes, la planification hebdomadaire des séances d'entraînement à raison de deux à trois par semaine, la gestion des convocations pour les matchs de championnat, ainsi que le suivi des feuilles de présence aux entraînements et aux matchs. À cela s'ajoutent la communication des résultats sportifs, la diffusion des actualités du club et la gestion des données sensibles dans le respect du Règlement Général sur la Protection des Données, notamment pour les joueurs mineurs.

À ce jour, l'ensemble de ces processus repose sur des outils disparates et non centralisés. Les informations sont réparties entre des documents papier archivés dans des classeurs au siège du club, des tableurs Excel partagés par messagerie électronique, des communications informelles via des groupes privés sur les réseaux sociaux et des notes manuscrites tenues par les entraîneurs lors des entraînements.

Cette organisation, bien qu'elle reste fonctionnelle dans un contexte de petite structure, montre aujourd'hui ses limites et freine le développement du club. La dispersion des données sur différents supports entraîne des incohérences fréquentes, comme des licences mal renouvelées ou des cotisations non correctement suivies, et ralentit considérablement les tâches administratives du bureau. Les entraîneurs ne disposent pas d'un accès immédiat aux informations de leurs joueurs, notamment lors des déplacements. De leur côté, les joueurs et leurs familles ne bénéficient d'aucun espace numérique centralisé pour consulter les convocations, les présences ou le calendrier des matchs. Enfin, l'absence de traçabilité complète constitue un risque sur le plan réglementaire, en particulier concernant le consentement parental pour les joueurs mineurs.

Face à ces constats, la direction du club Montpellier Arceaux a décidé de moderniser son système d'information en faisant appel à une équipe de développement. Le projet consiste à concevoir et à développer deux solutions numériques complémentaires et interconnectées. La première est une application web complète, développée avec le framework PHP Symfony 8, destinée à l'administration du club, incluant le bureau, les entraîneurs ainsi qu'un espace joueur accessible depuis un navigateur web.

La seconde est une application mobile développée avec Flutter et Dart, permettant aux joueurs, aux parents et aux entraîneurs d'accéder facilement aux informations essentielles depuis leur smartphone.

Ces deux solutions seront reliées à une API REST sécurisée par authentification JWT, garantissant la cohérence et la centralisation de l'ensemble des données au sein d'une base de données relationnelle unique. La mise en place de cette solution numérique permettra ainsi de moderniser les processus du club, de fiabiliser les données, d'améliorer la communication entre les différents acteurs et d'assurer une conformité complète avec les exigences du RGPD.

## Le client

---

Le club de football Montpellier Arceaux, association sportive affiliée à la Fédération Française de Football, souhaite moderniser son système de gestion afin de s'adapter à l'augmentation constante du nombre de licenciés et à la complexité croissante de ses activités administratives et sportives. En effet, au fil des saisons, la gestion des informations liées aux joueurs, aux équipes, aux entraînements, aux convocations et aux résultats est devenue de plus en plus difficile à organiser et à maintenir à jour.

L'utilisation d'outils dispersés, tels que des documents papier, des tableurs et des échanges via messageries, entraîne aujourd'hui un manque de fiabilité des données, une perte de temps importante et un risque accru d'erreurs. Cette situation limite la réactivité du club, complique la communication entre les différents acteurs et nuit à l'efficacité globale de l'organisation.

Dans ce contexte, le club exprime le besoin de mettre en place une solution numérique centralisée afin de regrouper l'ensemble des informations en un seul système, de faciliter leur accès et d'améliorer leur gestion au quotidien. L'objectif est également de structurer les processus internes afin de réduire les erreurs, d'optimiser le suivi des licenciés et de garantir une meilleure organisation des activités sportives.

Le client souhaite ainsi disposer d'une application web dédiée à l'administration du club, afin de permettre au bureau et aux entraîneurs de gérer efficacement les joueurs, les licences, les équipes, les entraînements, les convocations, les résultats sportifs et les actualités. Cette application devra également intégrer un tableau de bord synthétique afin d'offrir une vision globale et rapide des informations essentielles.

En parallèle, le développement d'une application mobile est attendu afin de permettre aux joueurs, aux parents et aux entraîneurs d'accéder facilement aux informations importantes, de consulter les convocations, les calendriers et les résultats, et de renforcer la communication au sein du club.

Enfin, le client insiste sur la nécessité de concevoir une solution évolutive, intuitive et performante, afin de s'adapter aux besoins futurs du club, tout en garantissant un haut niveau de sécurité et de confidentialité des données, notamment dans le respect des réglementations en vigueur.

## Le réseau et l'infrastructure

Le club Montpellier Arceaux dispose d'une infrastructure informatique légère, adaptée à la taille d'une association sportive amateur. Le bureau administratif est équipé de postes de travail fixes et portables utilisés principalement pour la gestion administrative. Les entraîneurs utilisent des smartphones et tablettes personnels pour leurs communications. Aucun serveur local n'est actuellement en place au sein du club.

### Architecture technique retenue

Pour ce projet, nous avons opté pour une architecture moderne basée sur Docker et des conteneurs, déployée sur un serveur distant accessible depuis Internet. Cette approche garantit la portabilité, la maintenabilité et la scalabilité de la solution.

| Composant            | Technologie retenue                              |
|----------------------|--------------------------------------------------|
| Serveur web          | Nginx (reverse proxy) dans conteneur Docker      |
| Backend              | PHP 8.4 — Framework Symfony 8 (conteneur Docker) |
| Base de données      | MySQL 8 (conteneur Docker dédié)                 |
| Authentification API | JWT (LexikJWTAuthenticationBundle)               |
| Application mobile   | Flutter / Dart — API REST Symfony                |
| Hébergement          | VPS Linux (Ubuntu 24.04)                         |
| Versionning          | Git — GitHub (repository privé)                  |
| Environnement dev    | Docker Desktop — .env.local                      |
| Emails (dev)         | Mailtrap (SMTP de test)                          |
| Emails (prod)        | Resend (3000 emails/mois gratuit)                |

### Sécurité du réseau

La sécurité de l'application est assurée à plusieurs niveaux. Les communications entre les clients (navigateurs web, application mobile) et le serveur sont chiffrés via HTTPS/TLS. L'authentification est gérée par des tokens JWT pour l'API et par des sessions sécurisées avec protection CSRF pour l'interface web. Un système de rate limiting est mis en place pour prévenir les attaques par force brute sur le formulaire de connexion.

La hiérarchie des rôles utilisateurs garantit un cloisonnement strict des accès : ROLE\_ADMIN (bureau), ROLE\_COACH (entraîneurs), ROLE\_TREASURER (trésorier), ROLE\_SECRETARY (secrétaire), ROLE\_PLAYER (joueurs), ROLE\_PARENT (parents). Chaque rôle hérite des permissions du rôle inférieur selon la chaîne définie dans la configuration Symfony.

## Les fonctionnalités

---

### Application mobile flutter

L'application mobile constitue l'interface principale du système pour les utilisateurs du club. Développée avec le Framework Flutter et le langage Dart, elle permet d'accéder facilement et rapidement à l'ensemble des informations sportives et administratives depuis un smartphone.

Elle repose sur une architecture basée sur une API REST sécurisée développée avec Symfony, ce qui permet de centraliser les données, de garantir leur cohérence et d'assurer une communication en temps réel entre les différents utilisateurs.

L'application offre une interface moderne, fluide et intuitive, adaptée aux usages mobiles. Elle a été conçue pour répondre aux besoins des différents profils utilisateurs, notamment les joueurs et les entraîneurs, en proposant des fonctionnalités spécifiques à chacun.

L'organisation de l'application repose ainsi sur plusieurs espaces accessibles en fonction du rôle de l'utilisateur connecté, permettant de différencier les fonctionnalités et les niveaux d'accès. Cette structuration garantit une utilisation claire, sécurisée et optimisée, tout en améliorant significativement l'expérience utilisateur.

---

### L'espace joueur

Accessible aux utilisateurs authentifiés disposant des rôles `ROLE_PLAYER` et `ROLE_PARENT`, l'espace joueur de l'application mobile Flutter a été conçu afin d'offrir un accès simple, rapide et centralisé à l'ensemble des informations liées à la vie sportive du joueur.

L'application propose un tableau de bord permettant de visualiser immédiatement les informations essentielles, notamment le prochain match ainsi que la prochaine séance d'entraînement. Cette vue synthétique permet au joueur d'anticiper rapidement ses échéances sportives.

L'utilisateur peut consulter ses convocations aux matchs à travers une liste détaillée indiquant la date, l'adversaire ainsi que son statut (titulaire ou remplaçant). Cette fonctionnalité facilite le suivi des sélections et améliore la communication avec l'entraîneur.

Un calendrier intégré permet de visualiser l'ensemble des matchs et des séances d'entraînement sur une vue mensuelle, offrant ainsi une vision globale et claire du planning sportif.

L'espace joueur comprend également une fiche personnelle regroupant les statistiques individuelles telles que le nombre de buts, les passes décisives, le temps de jeu ainsi que les éventuelles blessures en cours. Cette section permet au joueur de suivre ses performances tout au long de la saison.

Un module dédié aux présences permet de consulter l'historique de participation aux entraînements, favorisant un meilleur suivi de l'assiduité.

Par ailleurs, l'application permet d'accéder aux évaluations hebdomadaires réalisées par l'entraîneur, offrant un retour régulier sur les performances et les axes d'amélioration du joueur.

Enfin, un système de notifications push informe le joueur en temps réel des événements importants, tels qu'une nouvelle convocation, l'annulation d'un entraînement ou la réception d'un message, renforçant ainsi la réactivité et la communication au sein du club.

---

## **L'espace entraîneur**

Accessible aux entraîneurs authentifiés disposant du rôle `ROLE_COACH`, l'espace coach de l'application mobile Flutter constitue un véritable outil de gestion sportive, permettant de centraliser les informations essentielles et de piloter efficacement l'équipe au quotidien.

L'application propose un tableau de bord offrant une vue synthétique de l'effectif, notamment les joueurs disponibles, blessés ou absents en vue du prochain match. Cette vue permet à l'entraîneur de prendre rapidement des décisions adaptées à la situation de son équipe.

Un module de gestion des convocations permet de sélectionner les joueurs pour chaque match en définissant leur participation. Cette fonctionnalité simplifie l'organisation des rencontres et améliore la communication avec les joueurs.

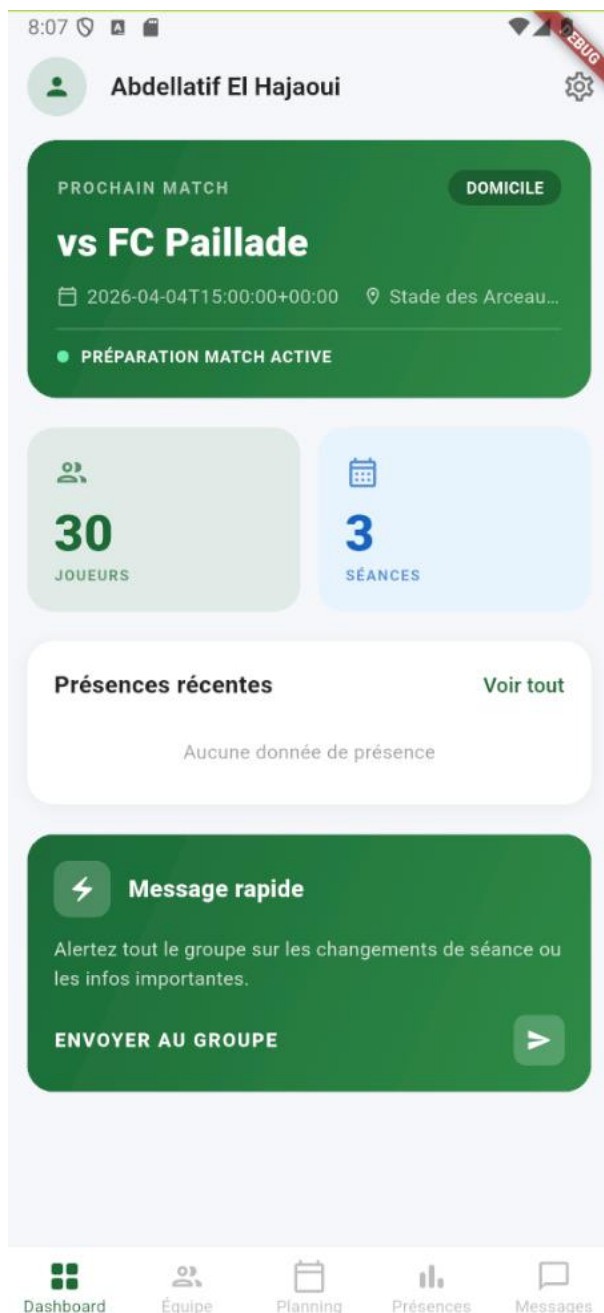
L'entraîneur dispose également d'un outil de composition tactique permettant de positionner les joueurs sur le terrain. Cette fonctionnalité s'appuie sur les composants existants tels que `TacticalBoard` et `SavedLineup`, facilitant la préparation stratégique des matchs.

Après chaque rencontre, l'application permet de saisir les statistiques individuelles des joueurs, notamment les buts, les passes décisives ainsi que les notes attribuées. Cela permet de suivre les performances et d'analyser les résultats de manière précise.

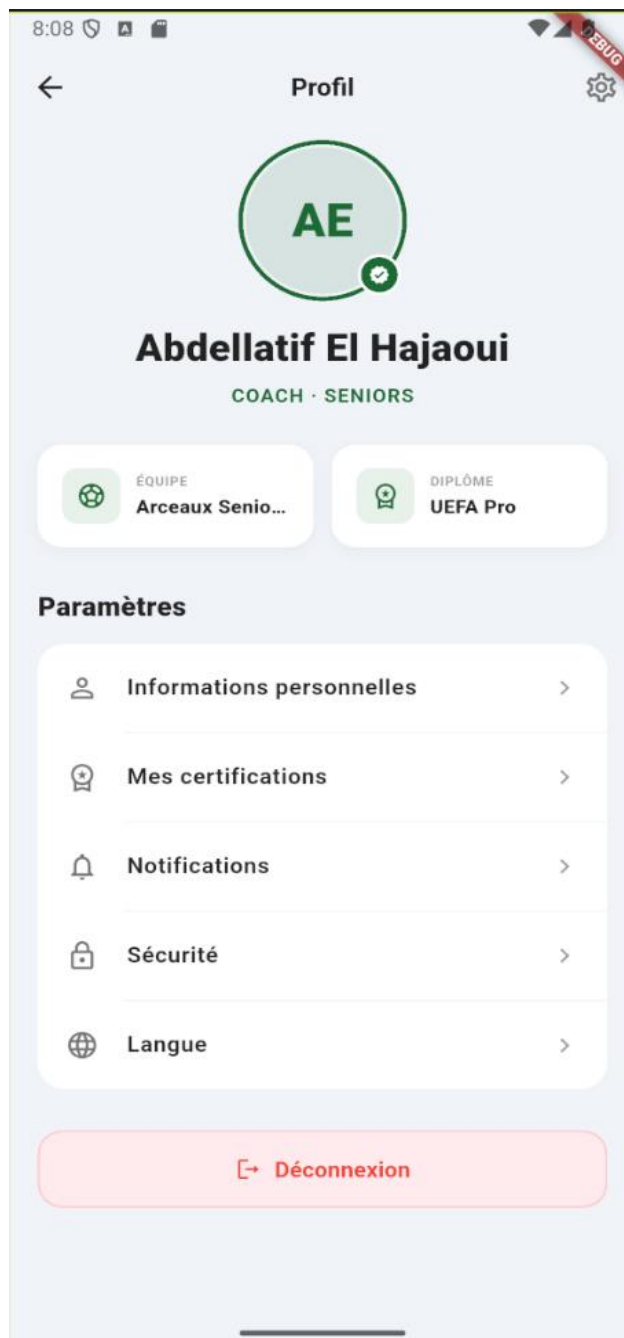
Un module de gestion des présences permet de marquer les joueurs présents ou absents lors des entraînements, offrant ainsi un suivi rigoureux de l'implication des joueurs.

Par ailleurs, l'entraîneur peut ajouter des notes privées sur les joueurs via un système dédié, lui permettant de suivre leur évolution et de préparer les séances d'entraînement de manière personnalisée.

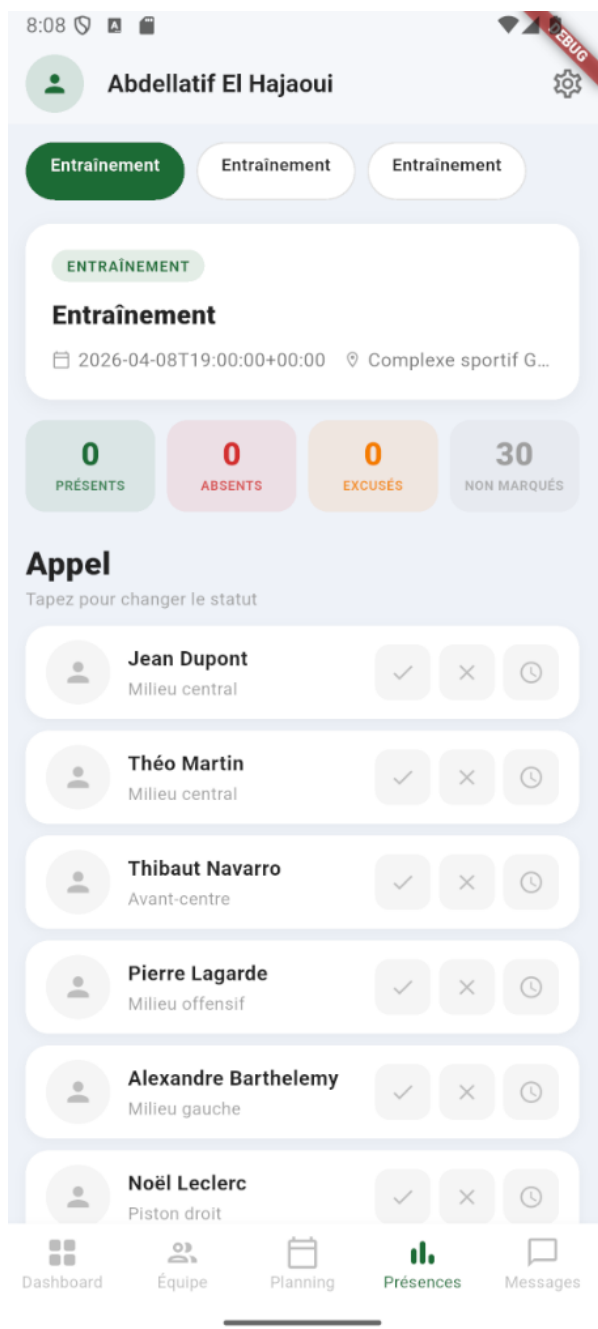
Enfin, l'ensemble de cet espace a été conçu pour une utilisation mobile, avec une interface fluide et intuitive, permettant aux entraîneurs d'accéder rapidement aux informations et de gérer leur équipe en toute situation.



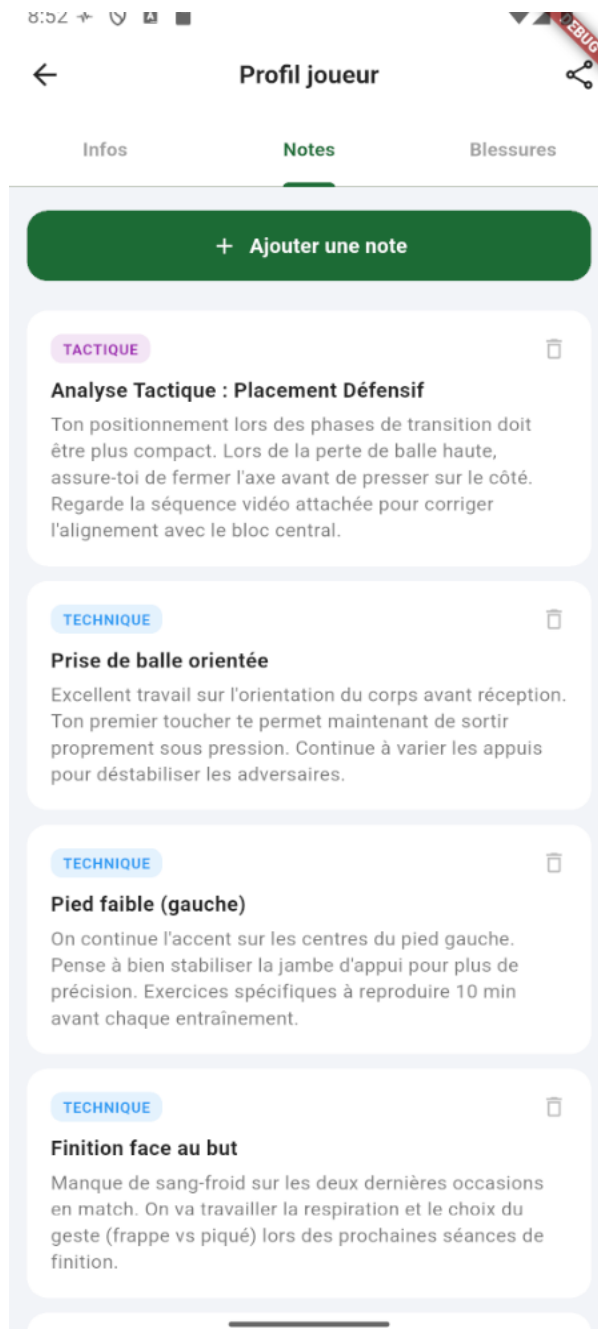
Page tableau de bord coach



page profil coach



Page présence coach



page notes coach

## **Fonctionnalités communes**

Certaines fonctionnalités sont accessibles à tous les utilisateurs de l'application mobile, qu'ils soient joueurs ou entraîneurs, afin de favoriser la communication et l'accès aux informations essentielles du club.

### **Messagerie interne**

L'application intègre une messagerie interne permettant aux joueurs et aux entraîneurs d'échanger directement. Chaque utilisateur peut envoyer et recevoir des messages, créer des conversations individuelles ou en groupe et suivre l'historique des échanges. Cette fonctionnalité améliore la communication rapide entre les acteurs du club, tout en conservant la sécurité et la confidentialité des échanges.

### **Actualités du club**

Les utilisateurs peuvent consulter les dernières actualités publiées par le club via un fil d'informations centralisé. Les articles disponibles sont issus de la base existante (entité Article) et peuvent inclure les résultats des matchs, les annonces importantes ou les événements à venir. Cette fonctionnalité permet aux joueurs et aux entraîneurs de rester informés en temps réel de la vie du club.

### **Événements**

Une section dédiée aux événements (entité Event) offre la possibilité de visualiser tous les matchs, tournois et autres rendez-vous importants de la saison. Chaque événement présente les informations essentielles telles que la date, l'heure, le lieu et les participants. Les utilisateurs peuvent également être notifiés des modifications ou annulations.

### **Blessures**

L'application permet de signaler et de consulter les blessures des joueurs (entité Injury). Les entraîneurs peuvent suivre l'état de santé de chaque joueur et adapter les entraînements en conséquence, tandis que les joueurs peuvent déclarer une blessure et consulter leurs antécédents médicaux. Cette fonctionnalité contribue à la sécurité des joueurs et à une meilleure organisation des séances.

---

## **Le back-office d'administration**

Réservé aux membres du bureau et aux responsables financiers (ROLE\_ADMIN, ROLE\_TREASURER), le back-office centralise toutes les opérations administratives et sportives du club. Il permet de gérer les joueurs, les équipes, les entraîneurs et les partenaires, avec des fonctionnalités complètes pour créer, modifier, supprimer ou anonymiser des données conformément au RGPD. La liste des joueurs peut être filtrée et exportée au format CSV, avec des alertes visuelles pour les mineurs sans consentement parental.

La planification des activités se fait via un calendrier interactif où les séances d'entraînement et les matchs peuvent être déplacés par glisser-déposer. Les convocations peuvent être créées et envoyées automatiquement par courriel, avec suivi des statuts des joueurs. Le suivi financier est intégré pour gérer les cotisations et les licences, avec notifications pour les paiements en attente.

Le back-office inclut également un module de gestion de contenu pour publier, modifier ou supprimer les articles d'actualité diffusés sur le site public. Enfin, un système avancé de gestion des rôles et permissions permet de contrôler l'accès aux différentes fonctionnalités, et offre la possibilité de simuler un accès en tant qu'entraîneur pour tester l'expérience utilisateur.

## La démarche projet

---

Pour la réalisation de ce projet, nous avons adopté une démarche structurée et méthodique, organisée en plusieurs phases complémentaires. Cette approche nous a permis d'assurer la cohérence, la qualité, la sécurité et la fiabilité de la solution tout en respectant les délais impartis. Chaque étape a été pensée pour répondre aux besoins spécifiques du club Montpellier Arceaux, tout en garantissant la maintenabilité et l'évolutivité du système.

### Phase 1 — Analyse des besoins et compréhension du contexte

La première étape du projet a consisté en une analyse approfondie des besoins du club. Cette phase a été réalisée en étroite collaboration avec les membres du bureau et les entraîneurs, afin de comprendre les pratiques existantes, identifier les processus métier et détecter les points de friction.

Nous avons organisé plusieurs sessions d'observation et d'entretiens pour recueillir les informations nécessaires : la gestion des joueurs, des licences, des équipes, des entraînements, des convocations, des résultats sportifs et des cotisations. L'objectif était de comprendre non seulement le fonctionnement actuel, mais aussi les contraintes réglementaires, notamment la conformité avec le RGPD pour le traitement des données des joueurs mineurs.

À l'issue de cette phase, nous avons produit un cahier des charges fonctionnel détaillant l'ensemble des besoins, les priorités et les contraintes. Ce document servait de référence pour toutes les étapes suivantes et comprenait une cartographie des acteurs, des processus et des flux d'information ainsi qu'une définition des indicateurs de succès pour le projet.

### Phase 2 — Conception de l'architecture et modélisation

Une fois les besoins validés, nous avons entamé la conception de l'architecture technique et fonctionnelle. Cette phase a consisté à définir la structure globale de la solution, incluant la base de données, l'API, l'organisation des rôles utilisateurs et la hiérarchie des accès.

La base de données relationnelle a été modélisée avec Doctrine ORM, et comprenait 19 entités représentant les objets métier principaux : utilisateurs, joueurs, équipes, entraîneurs, saisons, matchs, séances, convocations, présences, cotisations, articles, partenaires, et autres entités nécessaires au suivi complet de l'activité du club.

Des diagrammes UML ont été réalisés pour formaliser la solution :

- Les diagrammes de classes ont permis de représenter les entités et leurs relations.
- Les diagrammes de cas d'utilisation ont identifié les interactions possibles selon les rôles.

- Les diagrammes de séquence ont détaillé le déroulement des principales opérations métier, comme la création d'un match, l'envoi des convocations, ou la mise à jour des présences.

Parallèlement, nous avons défini l'architecture API pour la communication entre le backend Symfony et l'application mobile Flutter. L'authentification et la sécurisation des données ont été anticipées avec JWT et des permissions fines par rôle.

### **Phase 3 — Design, maquettes et expérience utilisateur**

La phase de design a consisté à élaborer des maquettes graphiques pour les différents espaces de l'application mobile ainsi que pour le back-office administratif.

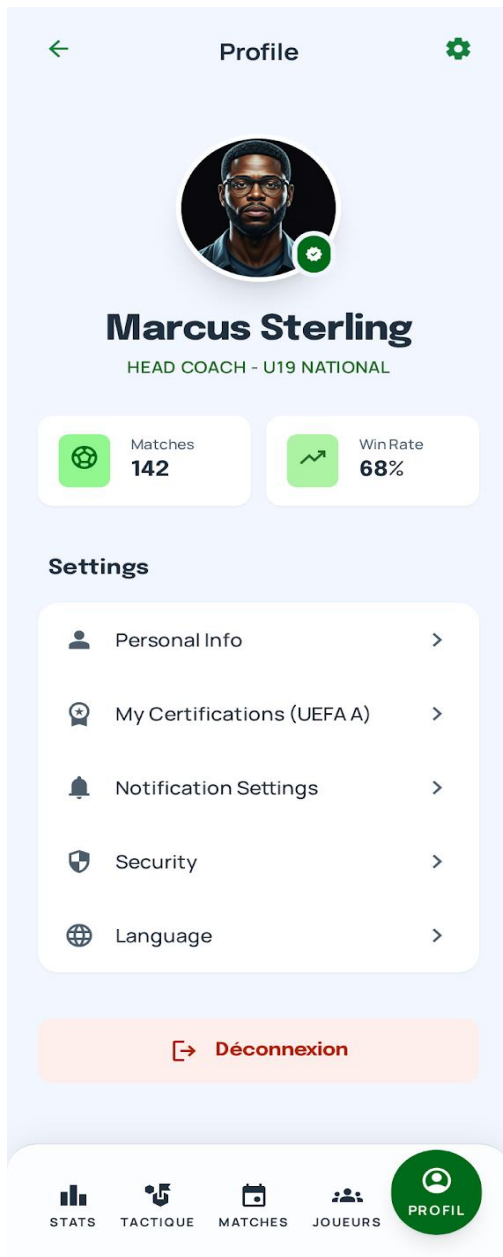
Nous avons défini un design system cohérent, reposant sur :

- Les couleurs du club, principalement le vert (#2E7D32), pour assurer l'identité visuelle
- La police DM Sans, pour la lisibilité et la modernité
- Une approche inspirée du Soft UI, afin de proposer des interfaces épurées, modernes et intuitives adaptées à un usage mobile

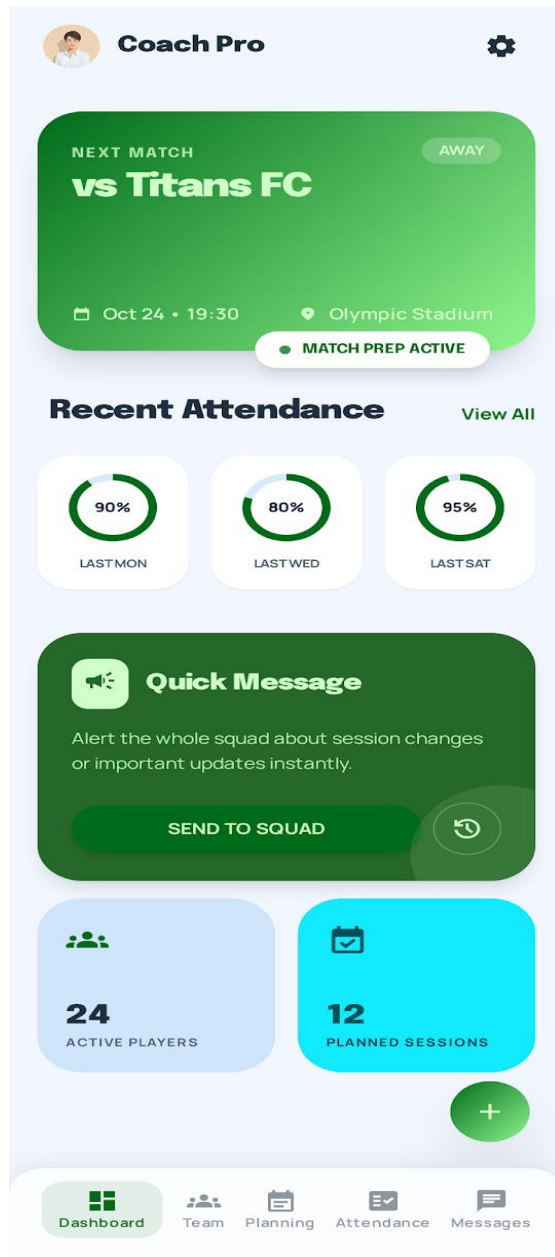
Les maquettes ont servi à :

- Tester et valider les parcours utilisateurs sur mobile
- Identifier les interactions critiques et les points d'optimisation
- Définir la hiérarchie des informations et la navigation entre les différents écrans

Cette phase a été particulièrement importante pour concevoir une expérience utilisateur fluide et adaptée aux contraintes des smartphones, notamment en termes de navigation, d'ergonomie et de rapidité d'accès aux informations.



Maquette profil coach



maquette tableau de bord coach

**Coach Pro**

## Training Session Oct 24

U-19 Elite Squad • Evening Drill

TEAM READINESS

**82%**

18/22 Players  
4 remaining

### Squad Roster

LIVE UPDATE

**Sarah Jenkins**  
Midfielder • #8

☒ ☐ 

**Marcus V.**  
Striker • #10

☒ ☒ 

**Leo G.**  
Defender • #4

☒ ☐ ☒

DM **David Miller**  
Goalkeeper • #1

☒ ☐ 

**Save & Notify Admin** ➤

Session data will be synced to the master cloud.

 Dashboard

 Team

 Planning

 Attendance

 Messages

**CoachConnect**

## Notes Joueurs

Observations privées et suivi technique

🔍 Rechercher un joueur...

**K. MBAPPÉ**

**A. GRIEZMANN**

**O. DEMBÉLÉ**

**T. HERNANDEZ**

### Historique des Notes

K. MBAPPÉ

12 OCT 2023 • ENTRAÎNEMENT

**Explosivité en transition**

★

Excellente forme lors des exercices de contre-attaque. Doit travailler sur le repli défensif après perte de balle haute. Mentalité de leader confirmée.

08 OCT 2023 • MATCH J-1

**Récupération musculaire**

Légère fatigue signalée au quadriceps gauche. Séance adaptée avec le kiné. Sera disponible pour 60 minutes demain.

05 OCT 2023 • TACTIQUE

**Placement 4-3-3**

Intégration fluide dans le nouveau système. Meilleure communication avec le milieu relayeur lors des phases de possession.

Ajouter une note rapide... 

 STATS

 TACTIQUE

 MATCHES

 JOUEURS

 NOTES

## Phase 4 — Développement et mise en œuvre technique

Le développement a été mené en suivant les bonnes pratiques de Symfony 8, notamment en respectant les principes de clean code et de modularité, afin de garantir une architecture robuste et maintenable.

En parallèle, une application mobile a été développée avec Flutter en utilisant le langage Dart, permettant de proposer une interface moderne et adaptée aux usages mobiles. Une attention particulière a été portée à la structuration du code, à la gestion des états ainsi qu'à l'optimisation des performances, afin d'assurer une application fluide, réactive et parfaitement connectée au backend via l'API REST.

Les principaux éléments techniques mis en œuvre sont les suivants :

- L'utilisation des attributs #[Route] et de l'injection de dépendances pour garantir la maintenabilité.
- La gestion avancée des formulaires avec validation côté serveur.
- L'utilisation de fixtures réalistes pour remplir la base de données de test et simuler les scénarios réels du club.
- La conteneurisation complète de l'environnement de développement avec Docker et Docker Compose, garantissant la reproductibilité et la portabilité.

Le développement a également couvert la mise en place de l'API REST sécurisée, nécessaire pour connecter le backend Symfony et l'application mobile Flutter. Chaque Endpoint a été conçu pour respecter les règles de sécurité, l'intégrité des données et la performance.

```

22 static const _green = ■Color(0xFF1A6B35);
23 static const _bg = □Color(0xFFEDF1F7);
24
25 // — Critères temps réel —————
26 bool get _hasMinLength ⇒ _passwordController.text.length ≥ 8;
27 bool get _hasUpperAndDigit ⇒
28     _passwordController.text.contains(RegExp(r'[A-Z]')) &&
29     _passwordController.text.contains(RegExp(r'[0-9]'));
30 bool get _hasSpecial ⇒
31     _passwordController.text.contains(RegExp(r'[@#$%^&*()\-_+=,.<.>?/;:{}\[\]|~]'));
32 bool get _passwordsMatch ⇒
33     _confirmController.text.isNotEmpty &&
34     _passwordController.text == _confirmController.text;
35
36 bool get _canSubmit ⇒
37     _hasMinLength && _hasUpperAndDigit && _hasSpecial && _passwordsMatch;
38
39 @override
40 void initState() {
41     super.initState();
42     // Rerender à chaque frappe pour mettre à jour les critères
43     _passwordController.addListener(() ⇒ setState(() {}));
44     _confirmController.addListener(() ⇒ setState(() {}));
45 }
46

```

*Code pour la réinitialisation du mot de passe*

```
60 routes: [  
61   // — Public  
62   GoRoute(  
63     path: '/',  
64     builder: (_, _) => const SplashScreen(),  
65   ), // GoRoute  
66   GoRoute(  
67     path: '/login',  
68     builder: (_, _) => const LoginScreen(),  
69   ), // GoRoute  
70  
71   // — Shell avec bottom navigation bar  
72   ShellRoute(  
73     builder: (context, state, child) => MainScaffold(  
74       location: state.matchedLocation,  
75       child: child,  
76     ), // MainScaffold  
77     routes: [  
78       GoRoute(  
79         path: '/home',  
80         builder: (_, _) => const HomeScreen(),  
81       ), // GoRoute  
82       GoRoute(  
83         path: '/sessions',  
84         builder: (_, _) => const SessionScreen(),  
85       ), // GoRoute  
86       GoRoute(  
87         path: '/summonings',  
88         builder: (_, _) => const SummoningScreen(),  
89       ), // GoRoute  
90       GoRoute(  
91         path: '/articles',  
92         builder: (_, _) => const ArticleScreen(),  
93       ), // GoRoute
```

*Route pour la navigation sur l'application*

```

124 // — Auth —————
125
126 /// Retourne true si le login réussit. Le token est sauvegardé.
127 Future<bool> login(String email, String password) async {
128     final res = await http.post(
129         Uri.parse('${baseUrl}/api/login_check'),
130         headers: {'Content-Type': 'application/json'},
131         body: jsonEncode({'username': email, 'password': password}),
132     );
133     if (res.statusCode == 200) {
134         await saveToken((jsonDecode(res.body) as Map)['token'] as String);
135         return true;
136     }
137     if (res.statusCode == 401) return false;
138     throw Exception('Erreur serveur (${res.statusCode})');
139 }
140
141 Future<void> logout() => deleteToken();
142
143 Future<void> forgotPassword(String email) =>
144     _post('/api/forgot-password', {'email': email}, auth: false);
145
146 Future<void> resetPassword(String token, String password) =>
147     _post('/api/reset-password', {'token': token, 'password': password}, auth: false);
148
149 /// Profil utilisateur connecté (id, email, roles[]).
150 Future<Map<String, dynamic>> getMe() async =>
151     ((await _get('/api/me')) ?? {}) as Map<String, dynamic>;
152

```

Code de l'api\_service pour l'authentification

```

// — Token —————
Ctrl+L for Agent
Future<String?> getToken() => _storage.read(key: _tokenKey);
Future<void> saveToken(String token) => _storage.write(key: _tokenKey, value: token);
Future<void> deleteToken() => _storage.delete(key: _tokenKey);
Future<bool> hasToken() async => (await getToken()) != null;

/// Décode le payload JWT (base64url) et retourne les rôles.
/// Aucun appel réseau nécessaire.
Future<List<String>> getRolesFromToken() async {
    final token = await getToken();
    if (token == null) return [];
    try {
        final parts = token.split('.');
        if (parts.length != 3) return [];
        // base64url → base64 standard (padding)
        var payload = parts[1].replaceAll('-', '+').replaceAll('_', '/');
        switch (payload.length % 4) {
            case 2: payload += '='; break;
            case 3: payload += '='; break;
        }
        final decoded = jsonDecode(utf8.decode(base64Decode(payload)))
            as Map<String, dynamic>;
        final roles = decoded['roles'] as List?;
        return roles?.cast<String>() ?? [];
    } catch (_) {
        return [];
    }
}

```

Code de l'api\_service pour le token

## Phase 5 — Tests, recettage et qualité

Une fois le développement achevé, une phase exhaustive de tests a été réalisée afin de valider la solution dans son ensemble.

Des tests unitaires et fonctionnels automatisés ont été mis en place à l'aide de PHPUnit afin de vérifier le comportement des différents composants du backend et de détecter les anomalies à un stade précoce.

Un plan de recettage manuel a également été défini, comprenant 87 cas de test couvrant l'ensemble des fonctionnalités du système : authentification, gestion des joueurs, des équipes, des séances, des matchs, des convocations, suivi des présences, gestion des cotisations, envoi de courriels, sécurité et conformité au RGPD.

Par ailleurs, des tests ont été réalisés sur l'application mobile Flutter afin de valider le bon fonctionnement des interfaces, la navigation entre les écrans ainsi que la communication avec l'API REST.

Cette démarche globale a permis de garantir une solution fiable, robuste et conforme aux attentes du club, tout en anticipant les éventuels problèmes liés aux échanges de données et aux interactions utilisateurs.

## Phase 6 — Suivi, maintenance et amélioration continue

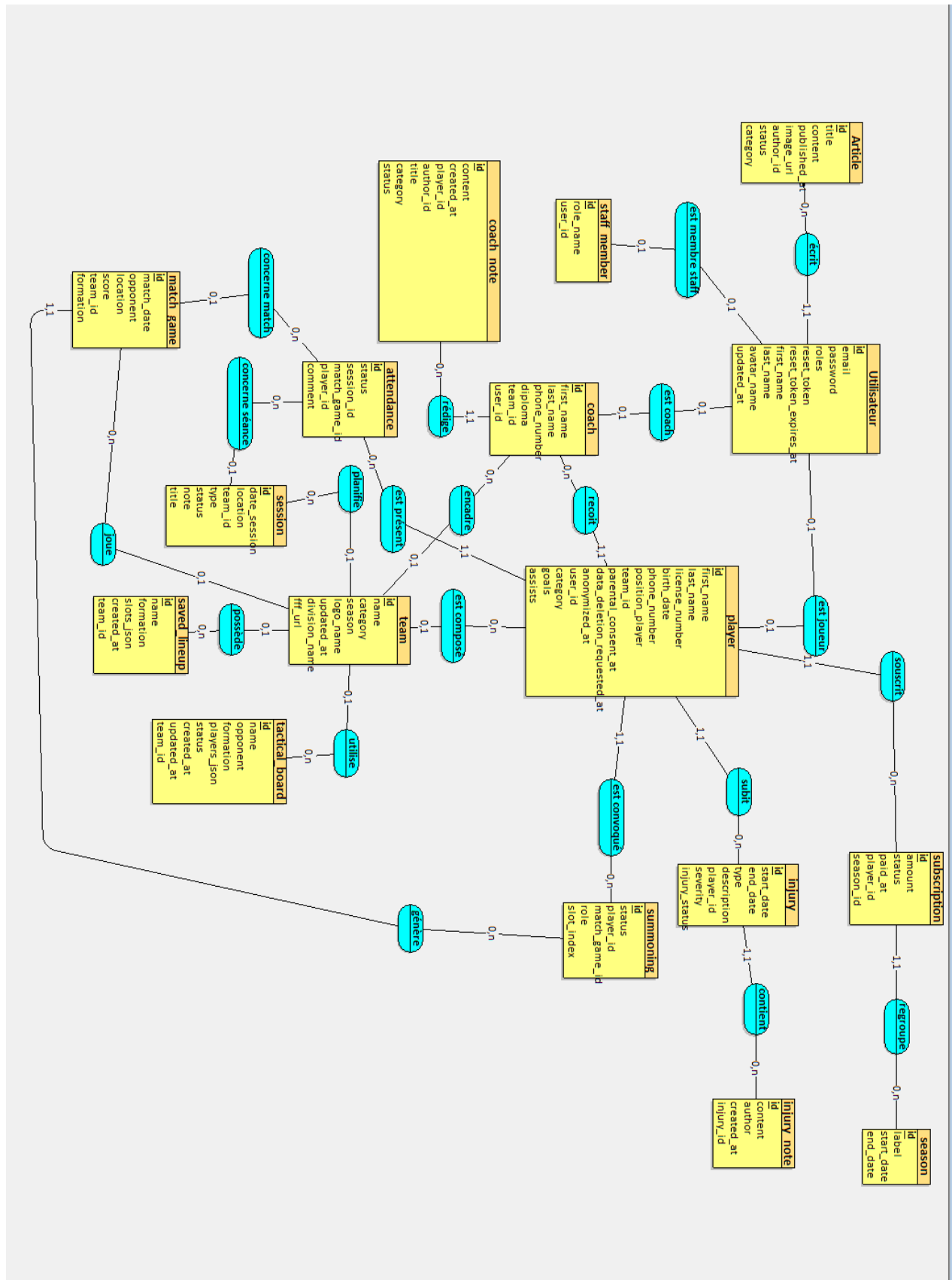
Le projet inclut également une phase de suivi post-déploiement, comprenant la documentation technique et utilisateur, la formation des administrateurs et entraîneurs, ainsi que la mise en place d'indicateurs de performance afin de suivre l'adoption et l'utilisation des solutions mises en place.

Une attention particulière est portée au suivi de l'application mobile Flutter, notamment en analysant les retours des utilisateurs afin d'améliorer l'expérience utilisateur et d'optimiser l'ergonomie de l'application.

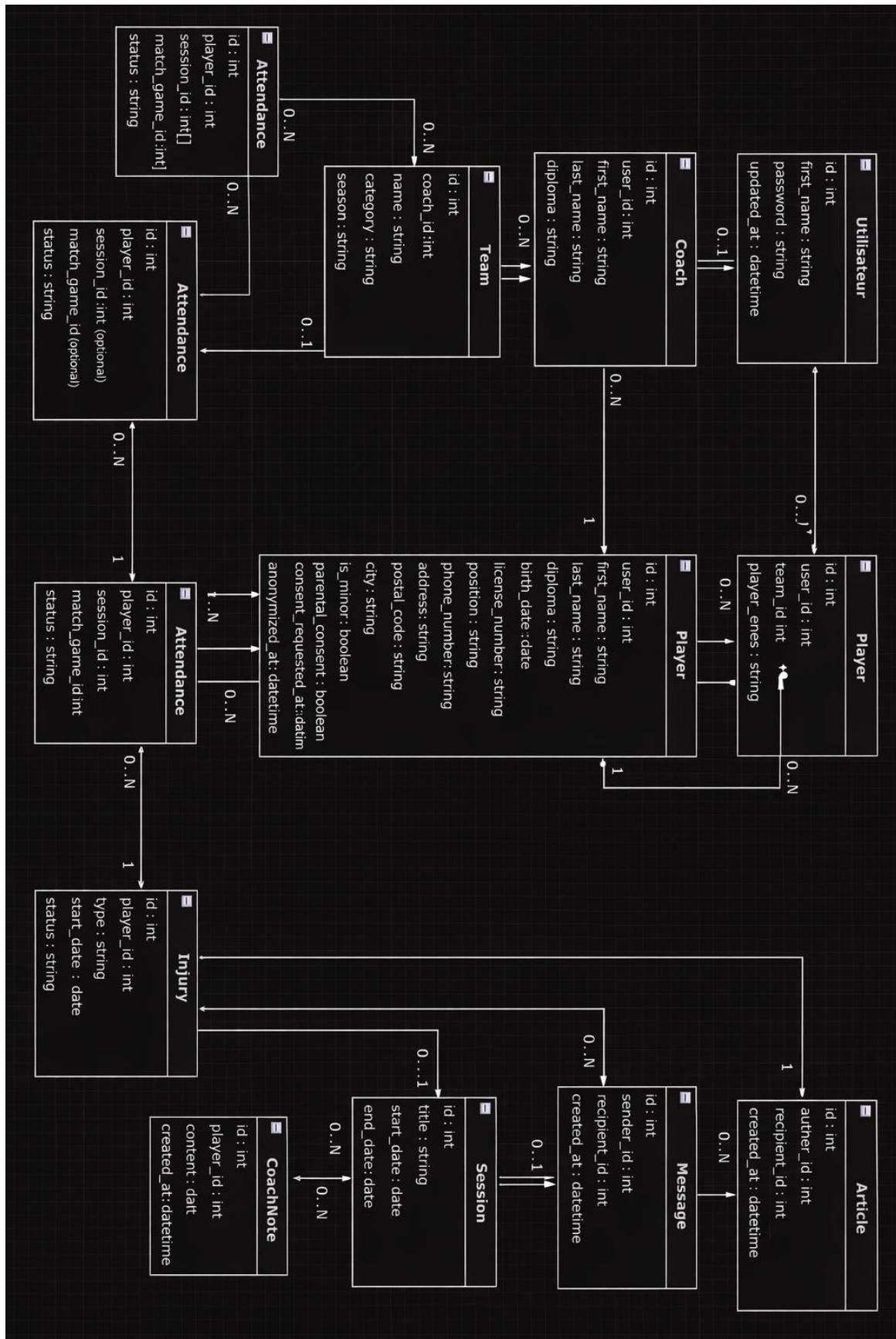
Nous avons prévu un mécanisme de maintenance évolutive, permettant d'ajouter de nouvelles fonctionnalités, de corriger d'éventuelles anomalies et d'optimiser les performances du système en continu. Cette démarche concerne à la fois le backend Symfony et l'application mobile, garantissant la pérennité de la solution et sa capacité à évoluer en fonction des besoins futurs du club.

## La base de données et les diagrammes UML

## Schéma Base de données



## Diagramme de classes (UML)



# Quelques fonctionnalités expliquées

---

## 1. Authentification et gestion des rôles

### Libellé de la fonctionnalité

Système d'authentification sécurisé avec hiérarchie de rôles et accès via application mobile.

### Explication de la fonctionnalité

Dans le cadre du Club des Arceaux, l'authentification des utilisateurs repose sur le composant Security de Symfony, garantissant un haut niveau de sécurité. Les utilisateurs s'authentifient via une API REST sécurisée, permettant une connexion depuis l'application mobile Flutter.

Lors de la connexion, les identifiants sont envoyés à l'API, qui valide les informations et génère un token d'authentification JWT. Ce token est ensuite utilisé par l'application mobile pour accéder de manière sécurisée aux différentes ressources du système.

Une fois authentifié, l'utilisateur accède à un espace spécifique au sein de l'application mobile en fonction de son rôle. Cette séparation constitue un élément central de l'architecture, permettant d'adapter les fonctionnalités et l'interface aux différents profils.

Les joueurs et parents accèdent à leur espace personnel, leur permettant de consulter leurs convocations, leur calendrier et leurs informations sportives. Les entraîneurs disposent d'un espace dédié à la gestion de leur équipe, incluant les convocations, les présences et les statistiques. Les administrateurs, quant à eux, accèdent au back-office web pour gérer l'ensemble du système.

Afin de sécuriser davantage le système, un mécanisme de limitation des tentatives de connexion (rate limiting) est mis en place, permettant de bloquer temporairement un compte après plusieurs échecs consécutifs et de limiter les attaques par force brute.

La gestion des autorisations repose sur une hiérarchie de rôles définie de manière centralisée dans la configuration de sécurité. Cette hiérarchie permet à certains rôles d'hériter automatiquement des droits d'autres rôles, simplifiant la gestion des accès et garantissant la cohérence du système.

La hiérarchie des rôles est définie dans security.yaml :

```
ROLE_ADMIN → [ROLE_COACH, ROLE_TREASURER, ROLE_ALLOWED_TO_SWITCH]
ROLE_TREASURER → [ROLE_SECRETARY]
ROLE_PRESIDENT → [ROLE_ADMIN]
```

Ce mécanisme de hiérarchie évite la duplication des accès et garantit qu'un administrateur hérite automatiquement de tous les droits des rôles inférieurs.

## 2. Gestion des joueurs et conformité RGPD

### Libellé de la fonctionnalité

Gestion complète du cycle de vie des joueurs avec respect du RGPD pour les mineurs.

### Explication de la fonctionnalité

La gestion des joueurs constitue une fonctionnalité essentielle du système d'information du Club des Arceaux. Elle est assurée côté backend avec Symfony et repose sur une base de données centralisée regroupant l'ensemble des informations liées aux joueurs.

Chaque joueur dispose d'une fiche individuelle contenant ses données personnelles, son numéro de licence, son équipe, son poste, ainsi que son historique de participation aux entraînements et aux matchs.

Ces données sont accessibles de manière sécurisée via l'API REST et sont exploitées dans l'application mobile Flutter. Les joueurs et leurs parents peuvent consulter leurs informations personnelles et leur suivi sportif, tandis que les entraîneurs peuvent accéder aux données nécessaires à la gestion de leur équipe.

Une attention particulière est portée à la gestion des données personnelles, notamment pour les joueurs mineurs. Afin de respecter les exigences du RGPD, plusieurs mécanismes ont été mis en place, tels que la gestion du consentement parental, le suivi des demandes de suppression et la mise en œuvre de procédures d'anonymisation.

Lorsqu'une suppression de données est demandée, une procédure d'anonymisation est appliquée. Celle-ci consiste à remplacer les données personnelles identifiables par des valeurs neutres, tout en conservant les données sportives nécessaires au suivi historique du club. Cette approche permet de concilier les obligations légales avec les besoins fonctionnels.

Par ailleurs, un système d'alerte permet d'identifier les joueurs mineurs ne disposant pas de consentement parental. Cette information est visible dans les interfaces de gestion, notamment côté administrateur, afin de faciliter le suivi et la mise en conformité.

### 3. Système de convocations avec courriels automatiques

#### Libellé de la fonctionnalité

Création de convocations pour un match avec envoi automatique de notifications et courriels aux joueurs.

#### Explication de la fonctionnalité

Le module de convocation permet aux entraîneurs d'organiser efficacement la participation des joueurs aux matchs. Depuis l'application mobile Flutter, les entraîneurs peuvent sélectionner les joueurs convoqués, définir leur statut (titulaire ou remplaçant) et ajouter des consignes spécifiques.

Une fois la convocation validée, les données sont enregistrées dans le backend Symfony via l'API REST, garantissant la centralisation et la cohérence des informations. Un processus automatisé est alors déclenché afin d'informer les joueurs concernés.

Ce système repose sur deux canaux complémentaires : l'envoi de notifications push directement sur l'application mobile Flutter, ainsi que l'envoi de courriels personnalisés.

Les notifications push permettent d'informer instantanément les joueurs d'une nouvelle convocation, améliorant ainsi la réactivité et la communication au sein du club.

En complément, les courriels assurent une trace écrite et consultable des informations transmises.

Le système d'envoi de courriels repose sur un service dédié, conçu de manière découplée afin de respecter les bonnes pratiques de développement.

En environnement de développement, les courriels sont interceptés pour permettre leur vérification sans envoi réel, tandis qu'en production, un service externe garantit leur bonne délivrabilité.

Chaque notification contient les informations essentielles relatives au match : adversaire, date, heure, lieu de rendez-vous ainsi que les consignes transmises par l'entraîneur.

### 4. Planning FullCalendar et gestion des séances

#### Libellé de la fonctionnalité

Calendrier mobile interactif permettant la visualisation et la gestion des séances d'entraînement et des matchs.

#### Explication de la fonctionnalité

Dans l'application mobile développée avec Flutter, le Club des Arceaux dispose d'un calendrier interactif conçu spécifiquement pour une utilisation mobile, offrant une visualisation claire et intuitive des activités sportives.

Les événements, tels que les entraînements et les matchs, sont affichés sous forme de calendrier mensuel ou hebdomadaire. Un système de code couleur permet de distinguer facilement les différents types d'activités, améliorant ainsi la lisibilité et la compréhension du planning.

Les utilisateurs peuvent interagir avec le calendrier en fonction de leur rôle. Les entraîneurs, notamment, ont la possibilité de modifier une séance directement depuis leur smartphone. L'interface mobile propose des interactions adaptées, comme le déplacement d'un événement via des gestes tactiles (drag & drop) ou la modification via un formulaire rapide.

Chaque modification effectuée déclenche une mise à jour en temps réel via l'API REST sécurisée. Les données sont envoyées au backend développé avec Symfony, garantissant la cohérence et la synchronisation des informations entre tous les utilisateurs de l'application.

Le calendrier est alimenté dynamiquement par les données issues de l'API, sous format JSON, incluant les informations essentielles telles que le titre de l'événement, les dates, les horaires, le type d'activité et les éventuelles actions disponibles (édition, suppression).

Ce module permet ainsi une gestion centralisée, fluide et accessible du planning sportif directement depuis un appareil mobile, tout en améliorant la réactivité, la communication et l'organisation globale du club.

## 5. Gestion du versioning et suivi du code source

### Libellé de la fonctionnalité

Mise en place d'un système de gestion de versions assurant la collaboration, la traçabilité et la qualité du code de l'application mobile.

### Explication de la fonctionnalité

Dans le cadre du développement de l'application mobile du Club des Arceaux réalisée avec Flutter, un système de gestion de versions complet a été mis en place afin de garantir un suivi rigoureux du code source et de faciliter le travail collaboratif.

Le projet repose sur l'utilisation de Git comme outil de versionning, associé à GitHub pour l'hébergement du dépôt distant. Le code de l'application mobile est isolé dans un dépôt dédié, permettant une gestion indépendante de son cycle de développement tout en restant connecté au backend via l'API.

L'organisation du projet suit une approche inspirée de GitFlow. Une branche principale *main* est utilisée pour les versions stables de l'application, tandis qu'une branche *develop* centralise les évolutions en cours. Chaque nouvelle fonctionnalité ou correction est développée dans une branche spécifique (ex : *feature/calendar*, *fix/login*), facilitant la lisibilité du projet et la maintenance du code.

Afin d'assurer une bonne compréhension de l'historique, le projet adopte la convention *Conventional Commits*. Chaque modification est accompagnée d'un message structuré avec des préfixes explicites tels que *feat*, *fix*, *refactor* ou *docs*. Cette méthodologie permet de suivre précisément l'évolution de l'application et de générer un historique clair et exploitable.

Le processus de validation du code repose sur l'utilisation de *Pull Requests*. Chaque contribution est revue avant son intégration dans la branche *develop*, garantissant ainsi la qualité, la cohérence et la stabilité du code.

Cette pratique favorise également le partage de connaissances et la détection précoce des anomalies.

Enfin, une configuration rigoureuse du fichier *.gitignore* est mise en place afin d'exclure les fichiers sensibles ou inutiles du dépôt, tels que les configurations locales, les fichiers temporaires ou les dépendances générées. Cette démarche garantit la sécurité des données et maintient un dépôt propre et optimisé.

Ce dispositif de versioning constitue un élément clé du projet, assurant une organisation efficace du développement, une meilleure collaboration entre les intervenants et une qualité globale élevée de l'application mobile.

## Axes d'amélioration

---

### Libellé de la fonctionnalité

Identification des évolutions fonctionnelles et techniques envisagées pour améliorer l'application mobile et le système d'information du Club des Arceaux.

### Explication de la fonctionnalité

À l'issue des phases de développement et de validation, plusieurs axes d'amélioration ont été identifiés afin de faire évoluer l'application mobile développée avec Flutter vers une solution encore plus complète, performante et adaptée aux besoins du club.

Un premier axe concerne la gestion des compétitions et tournois. Actuellement, l'application permet de gérer les matchs et les convocations, mais ne propose pas une gestion globale des compétitions. Une évolution consisterait à intégrer un module dédié permettant de créer des tournois, d'inscrire des équipes et de suivre les différentes phases (poules, élimination directe). Cela offrirait une vision centralisée de l'ensemble des engagements sportifs directement depuis l'application mobile.

Un second axe d'amélioration porte sur les statistiques sportives avancées. Bien que certaines données soient déjà disponibles, il serait pertinent d'ajouter des statistiques individuelles détaillées pour chaque joueur (buts, passes décisives, cartons, temps de jeu). L'intégration de graphiques interactifs via des bibliothèques Flutter permettrait de visualiser l'évolution des performances dans le temps et d'apporter une aide à la décision pour les entraîneurs.

Un troisième axe concerne les notifications en temps réel. Une intégration complète avec Firebase Cloud Messaging permettrait d'envoyer des notifications push instantanées lors d'événements importants (convocations, changements de planning, résultats). Cette amélioration renforcerait la réactivité et l'engagement des utilisateurs de l'application.

Par ailleurs, la mise en place d'un module de messagerie en temps réel constitue une évolution stratégique. L'objectif est de permettre des échanges directs entre joueurs, entraîneurs et membres du club sans dépendre d'outils externes. Une solution basée sur Symphony Mercure permettrait de gérer des communications instantanées, offrant une expérience fluide et moderne sur mobile.

Enfin, la gestion des blessures, désormais intégrée dans l'application, peut encore être enrichie. Ce module permet déjà de suivre l'état de santé des joueurs (type de blessure, durée d'indisponibilité, évolution). Des améliorations futures pourraient inclure la mise en place d'alertes automatiques pour signaler les joueurs indisponibles, ainsi que l'ajout d'indicateurs statistiques permettant d'anticiper les risques de blessure et d'optimiser la gestion physique des effectifs.

L'ensemble de ces évolutions s'inscrit dans une démarche d'amélioration continue, visant à proposer une application mobile toujours plus intuitive, complète et adaptée aux enjeux du club.

## Conclusion

---

Dans le cadre de ce projet, une collaboration étroite a été menée avec le club des Arceaux afin d'identifier précisément ses besoins et de proposer une solution numérique adaptée à son organisation et à ses enjeux. Cette démarche centrée utilisateur a permis de concevoir une application cohérente, répondant à des problématiques concrètes de gestion sportive, administrative et de communication interne.

La solution développée repose sur une architecture moderne et robuste, combinant une application web conçue avec le Framework Symfony et une application mobile développée avec Flutter. Ce choix technologique permet de couvrir l'ensemble des usages, aussi bien sur poste de travail que sur mobile, tout en garantissant performance, maintenabilité et évolutivité.

L'application web propose un environnement complet structuré autour de plusieurs espaces utilisateurs distincts, tandis que l'application mobile permet un accès rapide, intuitif et en mobilité aux informations essentielles du club. Cette complémentarité entre web et mobile constitue un véritable point fort du projet, en répondant aux usages réels des différents acteurs.

Une attention particulière a été portée aux aspects de sécurité et de conformité, avec la mise en place d'une authentification sécurisée, d'une gestion hiérarchisée des rôles, d'une protection contre les attaques et d'une prise en compte des exigences du RGPD, notamment pour les joueurs mineurs.

Sur le plan technique, l'architecture repose sur une API REST sécurisée par JWT, une conteneurisation avec Docker et une séparation claire entre le backend et les interfaces clientes, garantissant une solution fiable, maintenable et évolutive.

Les perspectives d'amélioration identifiées témoignent de la capacité du projet à évoluer vers une plateforme toujours plus complète, notamment avec l'ajout de nouvelles fonctionnalités comme la gestion des compétitions ou l'amélioration des interactions en temps réel.

Ce projet s'inscrit pleinement dans le cadre de ma formation en BTS SIO option SLAM au sein de EPSI, qui m'a permis d'acquérir des compétences solides en développement d'applications, en gestion de projet et en conception de systèmes d'information.

La pédagogie mise en place à l'EPSI, basée sur des projets concrets et professionnalisants, m'a permis de travailler dans des conditions proches du monde professionnel. Les cours spécialisés en développement web, en bases de données et en architecture applicative m'ont donné les connaissances nécessaires pour concevoir une application complète répondant à un besoin réel.

L'accompagnement par les intervenants et les coaches a également joué un rôle essentiel dans la réussite de ce projet. Leur suivi régulier m'a permis de structurer ma démarche, de faire des choix techniques pertinents et d'améliorer la qualité globale de mon travail. Cet encadrement m'a aidé à adopter une méthodologie rigoureuse, notamment dans l'analyse des besoins, la conception de l'architecture et la gestion du développement.

Sur la partie backend avec Symfony, j'ai pu renforcer mes compétences en développement structuré, en gestion des données et en sécurisation d'une API. Cette expérience m'a permis de mieux comprendre les enjeux liés à la logique métier et à la conception d'un système fiable.

Sur la partie mobile avec Flutter, j'ai découvert une approche centrée sur l'expérience utilisateur et la mobilité. J'ai appris à concevoir des interfaces modernes, à gérer les interactions utilisateurs et à connecter une application mobile à une API REST. Cela m'a permis de développer une vision plus globale du développement applicatif.

Ce projet m'a ainsi permis de prendre conscience de l'importance de la complémentarité entre les technologies web et mobile, mais aussi du rôle essentiel de l'analyse des besoins et de la communication avec le client dans la réussite d'un projet informatique.

En conclusion, cette réalisation constitue une expérience professionnalisante complète, qui m'a permis de consolider mes compétences techniques, de gagner en autonomie et de me préparer efficacement aux exigences du monde professionnel.