

## CONCEPTION ET DÉVELOPPEMENT D'UNE APPLICATION DE GESTION DE CLUBS SPORTIFS

< Insérez ici votre illustration >

(non obligatoire)

Thèse de Bachelor présentée par

**Capt Thibault**

Pour l'obtention du titre Bachelor of Science HES-SO en

**Informatique et systèmes de communication avec orientation en  
Informatique Logicielle**

**Mars, 2024**

Professeur-e HES responsable

**Eggenberg Niklaus**

Mandant

**Gaëtan Fetter  
Chênois Volleyball**

Légende et source de l'illustration de couverture :

## TABLE DES MATIÈRES

<b>Remerciements (style « Titre 1 »)</b>	<b>vi</b>
<b>Énoncé du sujet (style « Titre 1 »)</b>	<b>vii</b>
<b>Résumé (style « Titre 1 »)</b>	<b>viii</b>
<b>Liste des acronymes (style « Titre 1 »)</b>	<b>ix</b>
<b>Liste des illustrations</b>	<b>x</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Liste des annexes</b>	<b>xii</b>
<b>Introduction</b>	<b>1</b>
<b>Chapitre 1 : Élaboration et définition du cahier des charges</b>	<b>4</b>
<b>1.1 But du projet</b>	<b>4</b>
<b>1.2 Exigences</b>	<b>5</b>
a) Fonctionnalités principales	5
b) Spécifications techniques	6
c) Fonctionnalités secondaires	6
<b>1.3 Livrable</b>	<b>7</b>
<b>Chapitre 2 : Étude des solutions existantes</b>	<b>8</b>
<b>3.1 Analyse comparative</b>	<b>8</b>
a) SportEasy	8
b) TeamSnap	9
c) Spond	10
d) Heja	11
e) TeamLinkt	12
f) MonClubSportif	13
<b>3.2 Manques ou limitations</b>	<b>14</b>
a) Absence de centralisation totale	14
b) Manque de personnalisation	15
c) Interfaces datées ou peu intuitives	15
d) Abonnements payants et fonctionnalités limitées	15
e) Faible intégration de la sécurité et de la vie privée	15
f) Faibles fonctionnalités autour des parents	16
<b>3.3 Inspirations</b>	<b>16</b>
a) Centralisation des fonctionnalités (SportEasy, TeamSnap)	16
b) Gestion des présences et notifications (MonClubSportif, Spond)	17
c) Communication supervisée (MonClubSportif)	17
d) Statistiques et suivi des performances (SportEasy)	18
e) Design épuré et « mobile-friendly » (Heja, TeamLinkt)	18
<b>Chapitre 3 : Exploration des différentes technologies à disposition</b>	<b>19</b>
<b>3.1 Frontend</b>	<b>19</b>
<b>3.2 Backend</b>	<b>24</b>
a) Python	24
b) Node.js	24

c)	C# .....	25
d)	Java Spring Boot.....	25
<b>3.3</b>	<b>Base de données .....</b>	<b>26</b>
<b>3.4</b>	<b>Choix et justifications .....</b>	<b>27</b>
<b>Chapitre 4 : Prototype .....</b>		<b>28</b>
<b>3.5</b>	<b>Architecture globale .....</b>	<b>28</b>
<b>3.6</b>	<b>Explication des processus clés.....</b>	<b>30</b>
3.6.1	Frontend.....	30
3.6.2	Backend.....	32
<b>Conclusion .....</b>		<b>34</b>
<b>Annexes (style « Titre 1 ») .....</b>		<b>35</b>
<b>Références documentaires (style « Titre 1 »).....</b>		<b>39</b>

Dédicace à personne j'me suis fait tout seul.

## REMERCIEMENTS (STYLE « TITRE 1 »)

*< Formulez ici vos remerciements aux personnes qui vous ont aidé dans la réalisation de votre travail. Les remerciements sont rédigés dans le style « Corps de texte » >.*

*NB : les Titres « Titre 1 non numérotés » sont basés sur le titre 1 mais sont en dehors de la numérotation des chapitres. Ils apparaissent néanmoins dans la table des matières.*

## ÉNONCÉ DU SUJET (STYLE « TITRE 1 »)

< Insérez ici la page d'énoncé complété et signé  
par l'enseignant-e responsable (cf. feuille de style fournie) >  
(obligatoire)

*Attention : Tout l'énoncé doit tenir sur une seule page*

## **RÉSUMÉ (STYLE « TITRE 1 »)**

< Insérez ici la page de résumé de votre  
travail rédigée (cf. feuille de style fournie) >  
(obligatoire)

*Attention : Tout le résumé doit tenir sur une seule page*



# **LISTE DES ACRONYMES (STYLE « TITRE 1 »)**

## LISTE DES ILLUSTRATIONS

Illustration 1: diagramme machin .....	<b>Erreur ! Signet non défini.</b>
Illustration 2: schéma bidule .....	<b>Erreur ! Signet non défini.</b>
Illustration 3: Alice, Micro-ordinateur MATRA. ....	<b>Erreur ! Signet non défini.</b>
Illustration 4: diagramme Truc. ....	<b>Erreur ! Signet non défini.</b>
Illustration 5: diagramme Trucmuche. ....	<b>Erreur ! Signet non défini.</b>

*NB: Les listes sont à actualiser au moyen du menu contextuel « Mettre à jour l'index ». Attention, à ne faire qu'au dernier moment pour prendre en compte les numéros de page finaux. Les légendes peuvent être modifiées pour avoir des légendes simplifiées.*

## Références des URL

URL01	<a href="http://ce-site.ch/bla/bli/blo/blou.html">ce-site.ch/bla/bli/blo/blou.html</a>
URL03	<a href="http://ce-site.ch/blou/bli/bla.html">ce-site.ch/blou/bli/bla.html</a>
URL04	<a href="https://commons.wikimedia.org/w/index.php?curid=906980">https://commons.wikimedia.org/w/index.php?curid=906980</a>
URL06	<a href="http://ce-site.ch/monrapportdestage.pdf">ce-site.ch/monrapportdestage.pdf</a>

## LISTE DES TABLEAUX

Tableau 1: Lot de données n°1. ....	Erreur ! Signet non défini.
Tableau 2: Lot de données n°2. ....	Erreur ! Signet non défini.

*N.B. Si vous avez peu de tableaux, vous pouvez les intégrer à la table des illustrations.*

## Références des URL

URL02	ce-site.ch/bli/bla/blo/blou
URL05	ce-site.ch/publications/documents/ rapports/rapportsdestage/ monrapportdestage.pdf

LISTE DES ANNEXES

Annexe 1 .....36

Annexe 2 .....37

Annexe 3 .....38

## INTRODUCTION

Il est parfois compliqué de gérer un club sportif. Coordonner les équipes, suivre les présences, communiquer avec les joueurs et leurs familles, planifier les événements sportifs et extra sportifs... Il y a souvent une multitude d'outils utilisés. Chaque tâche à son logiciel dédié, ce qui complique encore plus la gestion au lieu de la simplifier.

Prenons par exemple, une application de messagerie. Elle est souvent détournée pour gérer la planification des événements et les communiquer avec les joueurs, les obligeants à retranscrire manuellement les informations dans leur propre calendrier, afin d'éviter tout oublis. Il faut ajouter à cela une application distincte pour la comptabilité et la gestion des cotisations. Avec un tel fonctionnement, les informations se retrouvent dispersées, multipliant les risques d'erreurs : doublons dans le suivi des présences, oublis, incohérences... Bref, un système loin d'être optimal.

Cette application se donne pour mission de simplifier et centraliser la gestion de club sportif. Conçue comme un outil tout-en-un, elle permet au directeur sportif de superviser ses équipes, d'avoir un suivi des cotisations et de coordonner efficacement les événements liés au club. Aux entraîneurs, d'avoir une solution intuitive pour créer des événements et de gérer la présence de ses joueurs.

Une des raisons primaires qui m'a aidé à choisir ce sujet est que, moi-même, j'ai dû utiliser une de ces applications lorsque je jouais au football. Lorsque je faisais partie d'une équipe, j'ai été confronté aux limites des outils de gestions sportifs. Ces applications proposaient des abonnements aux clubs. Ceux-ci ne voulaient pas utiliser leur budget pour gérer les plateformes et préféraient utiliser des outils moins adéquats. Nous devions jongler entre plusieurs plateformes pour communiquer et répondre aux présences à l'entraînement ou aux matchs. Cette configuration était loin d'être fluide. Certains joueurs oublièrent

fréquemment de répondre aux convocations, ce qui pouvait engendrer des retards et un manque d'organisation au sein de l'équipe.

Je me souviens que notre coach devait souvent rappeler aux joueurs d'aller répondre sur l'application. J'aimerais vraiment essayer d'apporter une approche plus naturelle où tout est centralisé sur une application pour éviter cette perte de temps. Mon ambition est de rendre la gestion des clubs sportifs non seulement plus facile, mais également plus agréable et naturelle, en optimisant les processus et en facilitant les échanges entre les différents acteurs.

Ce rapport constitue une étape préalable à la réalisation de mon travail de Bachelor en Informatique et Systèmes de Communications à L'Haute Ecole du Paysage, d'Ingénierie et d'Architecture de Genève, incluant la conception détaillée du cahier des charges, une étude des solutions déjà existantes, l'exploration des différentes technologies à disposition, ainsi qu'un prototype réalisé avec les technologies choisies.

Mandaté par Chênois Volleyball, ce projet a pour but de concevoir une application adaptée aux besoins de club. L'outil sera retreint aux exigences et au fonctionnement de Chênois Volleyball, en répondant aux besoins des membres administratifs du club, des entraîneurs et des joueurs (parents pour tous les joueurs mineurs). Elle met en avant les différents cas d'utilisation : gestion des présences, communication, organisation des évènements, partage de documents et suivi des performances des joueurs.

Grâce à l'utilisation de cette application, Chênois pourra nettement améliorer la gestion, la communication et l'organisation. Un gain de temps sera considérablement visible, car il n'y aura plus besoin de courir derrière les joueurs ou à corriger des erreurs de coordination grâce à des fonctionnalités intuitives et automatisées. Grâce au suivi des présences et aux évaluations, le club offrira une transparence totale vis-à-vis des entraîneurs, des joueurs et des parents. Cela va encourager l'implication des joueurs au sein du club, grâce à une gestion simplifiée en réduisant les outils et en améliorant la communication.

Enfin, l'outil est pensé pour évoluer avec les besoins du club, garantissant ses performances à moyen et long terme.

Ce rapport va permettre la réalisation de ce projet lors du projet de Bachelor. Il va couvrir les points suivants :

- Chapitre 1 : Élaboration et définition du cahier des charges
- Chapitre 2 : Étude des solutions existantes
- Chapitre 3 : Exploration des différentes technologies à disposition
- Chapitre 4 : Prototype

# CHAPITRE 1 : ÉLABORATION ET DÉFINITION DU CAHIER DES CHARGES

La grande partie de ce projet de semestre est d'élaborer et définir un cahier des charges. Je suis donc aller à la rencontre de Gaëtan Fetter, directeur sportif de Chênois Volleyball. Il avait, au préalable, réalisé une ébauche des fonctionnalités à implémenter qui se trouvera **en annexe**. Nous avons discuté sur ce document afin de compléter un cahier des charges robuste, qui servira pour le développement de l'application web lors du projet de Bachelor.

## 1.1 BUT DU PROJET

L'objectif principal de ce projet est de réaliser une application web permettant une gestion des activités et des joueurs au sein du club de Chênois Volleyball.

Voici les problématiques et enjeux à répondre lors de l'élaboration de ce cahier des charges :

- Améliorer la gestion des présences des joueurs.
- Avoir un système de suivi des absences et des retards.
- Faciliter l'organisation des compétitions, des entraînements, des matchs, et tous types d'évènements extra-sportif.
- Mettre en place un système de notifications pour les rappels d'évènements.
- Permettre le partage de documents entre les entraîneurs et le directeur sportif.
- Intégrer un système d'évaluation des joueurs afin d'assurer un suivi de leur progression.



## 1.2 EXIGENCES

### a) FONCTIONNALITÉS PRINCIPALES

1. Gestion des présences
  - a. Enregistrement des absences et retards (présent par défaut).
  - b. Justificatif si le joueur est en retard ou absent.
2. Calendrier des évènements
  - a. Calendrier commun pour tous les évènements d'équipe.
  - b. Convocation avec notification par email pour les joueurs et parents si l'enfant est mineur.
3. Directeur sportif
  - a. Gestion des joueurs (assurance mutuelle, type de licence, numéro AVS).
  - b. Gestion des équipes (assigner l'entraîneurs et les joueurs).
  - c. Accès aux calendriers des équipes.
  - d. Accès aux logs de l'application.
4. Entraîneur
  - a. Création d'évènements au sein de son équipe.
  - b. Préparer les convocations aux évènements.
5. Joueurs / Parent
  - a. Répondre aux convocations.
  - b. Gérer son profil (envoi d'une notification au directeur sportif qu'il peut approuver ou non).

**b) SPÉCIFICATIONS TECHNIQUES**

1. Authentification de base.
  - a. **Login** : Email et mot de passe.
  - b. **Rôles** : Directeur sportif, Entraîneur, Joueur / Parent.
2. Création de compte via lien par email (le club dispose de l'email des joueurs / parents).
3. Interface utilisateur simple et adaptée au web et mobile.

**c) FONCTIONNALITÉS SECONDAIRES**

1. Gestion des présences détaillé
  - a. Rapport de suivi des retards et des absences.
  - b. Utilisation de la banque de données jeunesse et sport pour enregistrer les présences aux entraînements via leur API.
2. Notifications et rappels
  - a. Email envoyé aux joueurs pour les convocations.
3. Partage de documents
  - a. Espace de stockage pour documents et vidéos.
4. Directeur sportif
  - a. Importer les joueurs via l'API VolleyManager (qui possède déjà toutes les données sur les joueurs).
  - b. Gestion des cotisations.

## 1.3 LIVRABLE

Les solutions finales de ce projet vont inclure :

- Une application web fonctionnelle permettant la gestion des présences, des évènements et des équipes.
- Un espace de stockage et de partage de document – **si implémenté**.
- Un système de gestion des joueurs du club.
- Une documentation détaillée sur les fonctionnalités et l'utilisation de l'application.

## CHAPITRE 2 : ÉTUDE DES SOLUTIONS EXISTANTES

Dans ce chapitre, nous allons étudier les différentes solutions existantes. Lors des recherches, 6 services ont été retenus et une analyse de leurs fonctionnalités a été réalisée.

SportEasy, TeamSnap, Spond, Heja, TeamLinkt et MonClubSportif.

### 3.1 ANALYSE COMPARATIVE

#### a) **SPORTEASY**

SportEasy est une application multiplateforme conçue pour répondre aux besoins des clubs sportifs en offrant un accès personnalisé pour chaque membre. Elle propose une gestion complète du club, notamment en permettant de centraliser les informations essentielles telles que les coordonnées des membres, les numéros de maillot, les allergies, ou encore les informations liées aux parents des joueurs. Cette fonctionnalité ressemble à un tableur Excel où l'on peut importer directement les données, ce qui facilite l'organisation. De plus, SportEasy propose un module dédié à la gestion des cotisations, ce qui est essentiel pour assurer le suivi financier du club.

Grâce aux outils pratiques et intuitifs, l'application facilite la gestion d'équipe. Les entraîneurs peuvent planifier différents types d'événements, qu'il s'agisse de matchs, d'entraînements, de repas, ou encore de tournois, le tout accessible via un calendrier intégré. SportEasy permet également de créer des compositions d'équipes et de gérer des assignations spécifiques, telles que le lavage des maillots, l'organisation de pique-niques, ou encore la mise en place de covoiturages. Ces fonctionnalités contribuent à renforcer l'organisation et à alléger la charge administrative des responsables. Toutefois, certaines de ces options avancées, comme l'assignation des tâches ou l'accès à des statistiques complètes, sont réservées aux utilisateurs abonnés à la version payante de l'application.

L'application met également un point d'honneur au suivi des performances sportives grâce à des outils dédiés, des championnats et des statistiques détaillées. Ces données incluent des notes individuelles, des résultats en direct, et même la désignation du meilleur joueur de la rencontre. Enfin, SportEasy favorise la communication, en proposant des fonctionnalités pour faciliter les échanges au sein des équipes et entre les membres du club.

Un autre aspect intéressant de SportEasy est l'absence de publicités classiques, ce qui garantit une expérience utilisateur fluide et sans distractions inutiles. Cependant, pour accéder pleinement à toutes les fonctionnalités avancées, un abonnement payant est requis, ce qui peut représenter un frein pour certains clubs disposant d'un budget limité.

## **b) TEAMSNAP**

TeamSnap est une application tout-en-un qui se positionne comme un outil de gestion complet pour les clubs sportifs. L'un de ses points forts réside dans son module de Program Management, qui centralise toutes les informations essentielles du club au sein d'un hub sécurisé. Ce système garantit la protection des données sensibles tout en permettant d'inviter facilement les membres du personnel à rejoindre la plateforme pour collaborer.

L'application propose également des outils utiles pour la gestion des inscriptions. En effet, la gestion des cotisations est simplifiée grâce à la configuration des plans de paiements flexibles. Les clubs peuvent également bénéficier d'outils spécifiques pour l'organisation des saisons sportives, ce qui facilite la planification à long terme.

TeamSnap se distingue aussi par son « website builder », permettant à chaque club de créer un site internet personnalisé. Cette fonctionnalité donne aux clubs une visibilité professionnelle et la possibilité de communiquer efficacement avec leurs membres et leur communauté. De plus, l'application prend en charge l'organisation des tournois, ce qui en fait un outil adapté aux clubs participant régulièrement à des compétitions. TeamSnap va même plus loin en offrant des solutions pour gérer les sponsorships, permettant aux clubs d'attirer et de gérer des partenariats financiers.

La version mobile de TeamSnap inclut des fonctionnalités intéressantes, telles que la communication de bout en bout entre les membres, ainsi qu'un planificateur intégré pour organiser les événements et assigner les équipes. Toutefois, malgré ces atouts, l'interface de l'application mobile est souvent critiquée pour son design peu attrayant et sa finition insuffisante.

En somme, TeamSnap offre des fonctionnalités très variées et adaptées à la gestion d'un club sportif, mais son design d'application mobile qui doit être amélioré, peut être un frein pour les utilisateurs cherchant une expérience plus moderne et intuitive.

### **c) SPOND**

Spond est une application polyvalente qui facilite la gestion des clubs sportifs en offrant des outils intuitifs pour la configuration et l'administration des groupes. Les responsables de club peuvent facilement organiser leurs équipes, ajouter de nouveaux membres, notamment grâce à une fonctionnalité d'importation de données depuis des fichiers Excel, et centraliser la communication. Ces options permettent de simplifier la gestion administrative et de maintenir une organisation efficace.

L'un des points forts de Spond réside dans son système de notifications accessible. Même si un utilisateur ne possède pas l'application mobile, il peut tout de même recevoir des informations importantes par SMS ou email. Cela garantit que tous les membres restent connectés et informés, peu importe leur préférence de support. De plus, l'outil permet d'envoyer des rappels personnalisés, idéal pour confirmer les présences, rappeler les échéances ou notifier des modifications d'événements.

En termes d'expérience utilisateur, le design de Spond est simple mais fonctionnel, répondant aux besoins essentiels sans être particulièrement innovant ou sophistiqué. Cela rend l'application accessible et facile à utiliser, même pour ceux moins familiers avec les outils numériques. Cependant, un inconvénient majeur est la lenteur du site web, qui peut

rendre certaines opérations plus fastidieuses, notamment lorsqu'il y a une charge importante de données ou une utilisation intense.

En conclusion, Spond propose une solution pratique et accessible pour la gestion des équipes et des communications. Malgré ses limites concernant la rapidité de son site, ses outils de notification multiplateforme et son interface simple en font une option solide pour les clubs sportifs recherchant une gestion centralisée et efficace.

#### **d) HEJA**

Heja est une application moderne et intuitive dédiée à la gestion des clubs sportifs, offrant une gamme de fonctionnalités essentielles pour simplifier l'organisation et améliorer la communication entre les membres. L'un de ses principaux avantages est son scheduler, qui permet une planification efficace des événements tels que les entraînements, les matchs, et autres activités liées à l'équipe.

En termes de gestion d'équipe, Heja propose des outils pratiques pour organiser les effectifs, attribuer des rôles, et s'assurer que chaque membre dispose des informations nécessaires pour participer pleinement à la vie du club. Une des fonctionnalités phares de l'application est son système de communication intégré, qui permet aux membres de rester connectés directement dans l'application. Notamment, la possibilité d'effectuer des appels au sein même de l'app est un atout qui la distingue de nombreuses autres solutions, rendant la coordination encore plus fluide et rapide.

Sur le plan esthétique, Heja bénéficie d'un design mobile attractif et bien pensé, offrant une interface moderne, ergonomique et agréable à utiliser. Cela contribue à une expérience utilisateur optimale, en particulier pour les membres qui préfèrent accéder aux fonctionnalités via leur smartphone.

En résumé, Heja est une application élégante et performante, particulièrement adaptée aux clubs sportifs cherchant une solution intuitive pour gérer leurs équipes et

maintenir une communication efficace. Sa capacité à intégrer des appels directement dans l'application et son design soigné en font un choix attrayant, même si l'analyse détaillée de ses autres aspects, comme les limitations éventuelles, pourrait être approfondie.

#### **e) TEAMLINKT**

TeamLinkt est une plateforme complète et flexible dédiée à la gestion des équipes sportives, offrant des outils pour faciliter l'organisation et améliorer l'expérience des clubs et des joueurs. L'application propose un scheduler efficace, permettant de planifier les entraînements, matchs, événements, et autres activités d'équipe de manière structurée.

En termes de gestion d'équipe, TeamLinkt inclut une gamme de fonctionnalités adaptées, telles que l'attribution de rôles et la gestion des inscriptions. De plus, la plateforme se distingue en intégrant la gestion des inscriptions par saison, ce qui rend les processus administratifs plus simples et accessibles.

La fonctionnalité de gestion des tâches permet d'assigner et de suivre des responsabilités, améliorant ainsi la coordination entre les membres d'une équipe. De plus, l'application propose une team app dédiée, offrant un accès centralisé à toutes les informations importantes pour les joueurs et entraîneurs.

Un autre atout majeur de TeamLinkt sont les statistiques. Elles fournissent des données sur les scores, classements et résultats, donnant une vue d'ensemble des performances de l'équipe et permettant aux entraîneurs et joueurs de suivre leur progression.

TeamLinkt permet également aux clubs de personnaliser un site internet selon leurs besoins. Ce site peut inclure des informations sur les équipes, les horaires, les résultats et même des fonctionnalités de collecte de fonds, offrant ainsi une vitrine professionnelle pour le club tout en centralisant les communications avec les membres et les supporters.



En ce qui concerne l'interface, TeamLinkt opte pour une vue type dashboard, épurée et bien organisée, ce qui facilite la navigation et l'accès rapide aux différentes fonctionnalités. Ce design clair et intuitif est particulièrement apprécié pour sa simplicité et son efficacité.

En résumé, TeamLinkt est une solution polyvalente et conviviale, idéale pour les clubs sportifs cherchant à centraliser la gestion de leurs équipes et à simplifier les processus administratifs. Son interface épurée et ses options avancées, telles que la collecte de fonds et la gestion des statistiques, en font un choix intéressant pour de nombreux utilisateurs.

#### **f) MONCLUBSPORTIF**

MonClubSportif se positionne comme une solution complète et adaptée pour la gestion d'équipes sportives et culturelles. Contrairement à certaines plateformes exclusivement sportives, cette application s'adresse également à des publics diversifiés tels que les milieux scolaires, les associations sportives, ainsi que les entraîneurs individuels, ce qui en fait un outil polyvalent. Elle prend également en compte les parents, leur permettant de suivre l'activités des jeunes joueurs.

L'un des points forts de MonClubSportif réside dans sa vue d'ensemble intuitive, qui centralise efficacement les événements, communications, et calendriers. Cela permet une gestion simplifiée et une visibilité accrue pour tous les membres du club.

L'application met l'accent sur une communication sécurisée et saine, garantissant un cadre respectueux et adapté pour les échanges entre membres. Par exemple, la messagerie instantanée empêche les entraîneurs de communiquer seul à seul avec les jeunes joueurs, ce qui renforce la sécurité et la transparence. De plus, un droit de regard sur les discussions est mis en place, permettant aux responsables et aux parents de superviser les échanges au sein du club.

La plateforme intègre également un système de confirmation des présences, où les joueurs (ou leurs parents) peuvent signaler leur statut pour chaque événement (présent, retard, incertain, absent). Cette fonctionnalité favorise une meilleure organisation et une planification optimisée des entraînements et compétitions.

Par ailleurs, MonClubSportif offre un emplacement de stockage pour les documents et vidéos, permettant aux entraîneurs et membres de partager facilement des fichiers liés aux entraînements, stratégies ou informations administratives.

Cependant, un point faible notable est le design de l'application, qui est un peu "vieillot". Bien que fonctionnel, il pourrait bénéficier d'une modernisation pour être plus attractif et intuitif.

En résumé, MonClubSportif est une plateforme solide et polyvalente, parfaitement adaptée aux clubs qui recherchent une solution sécurisée, centrée sur les valeurs de transparence et de communication saine. Son orientation vers les parents et sa capacité à répondre aux besoins variés des milieux sportifs et culturels en font un outil pratique, malgré un design qui pourrait être amélioré.

## 3.2 MANQUES OU LIMITATIONS

### a) ABSENCE DE CENTRALISATION TOTALE

La majorité des applications ne couvrent pas tous les aspects d'un club sportif. En effet, certaines sont fortes pour la gestion des équipes et des présences (ex. : Spond, Heja), tandis que d'autres se concentrent sur les paiements et la planification (ex. : TeamSnap, SportEasy). Très peu offrent une solution **complète** regroupant communication, gestion des cotisations, planification, partage de documents, et statistiques dans une seule interface.

**b) MANQUE DE PERSONNALISATION**

Les clubs sportifs souhaitent un outil qui reflète leur identité (logo, couleurs, site web personnalisé, etc.). Bien que certains outils comme TeamLinkt et SportEasy permettent de personnaliser un site internet, cette fonctionnalité reste limitée et peu développée (design figé, options restreintes).

**c) INTERFACES DATÉES OU PEU INTUITIVES**

Plusieurs outils comme **MonClubSportif** ou **Spond** ont des designs d'applications jugés peu modernes ou simplistes. Cela peut décourager les utilisateurs habitués à des interfaces plus intuitives et attrayantes.

**d) ABONNEMENTS PAYANTS ET FONCTIONNALITÉS LIMITÉES**

La plupart des plateformes proposent des fonctionnalités basiques gratuites, mais les clubs doivent souscrire à des abonnements coûteux pour accéder à des options avancées comme :

- Gestion des cotisations (**SportEasy**).
- Gestion des tâches ou outils statiques (**TeamSnap**, **TeamLinkt**)
- Absence de publicité.

**SportEasy** et **MonClubSportif** imposent souvent une version premium pour des fonctionnalités cruciales comme le suivi des statistiques ou la communication avancée.

**e) FAIBLE INTÉGRATION DE LA SÉCURITÉ ET DE LA VIE PRIVÉE**

Hormis **MonClubSportif**, peu d'outils se préoccupent de la sécurité des communications entre joueurs et entraîneurs, ce qui est pourtant essentiel, surtout pour les équipes de jeunes joueurs. Une communication supervisée ou la possibilité d'impliquer les parents est souvent absente dans ces outils.

**Heja** et **TeamSnap** permettent la communication, mais sans contrôle parental intégré, ce qui peut poser un problème pour les équipes avec des joueurs mineurs.

## Mauvaise gestion des absences et des présences

Les solutions existantes permettent d'enregistrer les présences, mais elles ne vont pas plus loin :

- Pas de système avancé pour la justification des retards directement dans l'application
- Pas de système avancé pour les absences longue durée (ex. : blessure)

### f) FAIBLES FONCTIONNALITÉS AUTOUR DES PARENTS

Bien que **MonClubSportif** mette l'accent sur les parents (droit de regard, notifications), d'autres plateformes comme **Heja**, **TeamLinkt** ou **Spond** n'accordent pas autant d'attention à leur implication. Pourtant, les parents pourraient avoir un rôle important, notamment dans la validation des absences, en recevant les notifications ou en suivant les performances de leurs enfants si ceux-ci sont mineurs.

## 3.3 INSPIRATIONS

### a) CENTRALISATION DES FONCTIONNALITÉS (SPORTEASY, TEAMSNAP)

Ce qui m'inspire :

- La capacité de **SportEasy** à gérer des équipes, planifier des événements et suivre les cotisations.
- La fonction de **TeamSnap** qui centralise la gestion d'un club dans un hub unique, simplifiant l'accès aux informations.

Ce que je pourrais en faire :

- Créer une plateforme où tout est regroupé dans une interface ergonomique.

#### **b) GESTION DES PRÉSENCES ET NOTIFICATIONS (MONCLUBSPORTIF, SPOND)**

Ce qui m'inspire :

- La gestion des présences dans **MonClubSportif**, qui permet de marquer un joueur comme présent, absent ou en retard.
- Le système de notifications automatisées par SMS ou e-mail de **Spond** pour prévenir les membres qui ne possède pas l'application.

Ce que je pourrais en faire :

- Ajouter des notifications intelligentes pour les absences non justifiées.
- Inclure un suivi d'assiduité dans le temps pour aider les entraîneurs à analyser la participation des joueurs.

#### **c) COMMUNICATION SUPERVISÉE (MONCLUBSPORTIF)**

Ce qui m'inspire :

- La messagerie sécurisée de **MonClubSportif**, où les communications entre entraîneurs et jeunes sont supervisées pour garantir un environnement sain.

Ce que je pourrais en faire :

- Proposer un système similaire avec un rôle pour les parents, qui communiqueront eux-mêmes avec l'entraîneur si besoin.

**d) STATISTIQUES ET SUIVI DES PERFORMANCES (SPORTEASY)**

Ce qui m’inspire :

- Les statistiques détaillées de **SportEasy** sur les performances individuelles (notes, meilleur joueur du match) et d’équipe.

Ce que je pourrais en faire :

- Étendre ces statistiques en proposant des indicateurs d’assiduité.

**e) DESIGN ÉPURÉ ET « MOBILE-FRIENDLY » (HEJA, TEAMLINKT)**

Ce qui m’inspire :

- Le design moderne et attrayant de l’application mobile de **Heja**.
- L’interface claire et intuitive de **TeamLinkt**, notamment son tableau de bord.

Ce que je pourrais en faire :

- Proposer un design qui s’adapte aux besoins des différents profils d’utilisateurs (directeur sportif, entraîneurs, joueurs / parents) et qui reste agréable à utiliser sur mobile et desktop.

## CHAPITRE 3 : EXPLORATION DES DIFFÉRENTES TECHNOLOGIES À DISPOSITION

Dans un premier temps, le plus simple est de se recentrer sur une application web afin d'assurer la compatibilité avec tout type d'appareil électronique. J'aime aussi beaucoup coder pour faire du web, donc ce n'est que bénéfique !

### 3.1 FRONTEND

De nos jours, les bibliothèques frontend les plus utilisées sont React, Vue.js et Angular. Ils sont tous basés sur JavaScript. Pour ma part, je préfère travailler avec Type Script qui vise à améliorer JavaScript en ajoutant une syntaxe stricte à celui-ci. Toutes les trois technologies prennent en charge Type Script. De plus, Angular et Vue.js sont des frameworks, tandis que React est une bibliothèque. Ce qui veut dire que React n'est pas une solution tout-en-un comparé à Angular et Vue.js. Je vais donc plutôt comparer Angular et Vue.js avec Next.js qui est une solution basée sur React. Comme ça, la comparaison est plus pertinente, car ce sont 3 environnements distincts.

Pour mes recherches, j'ai lu beaucoup d'articles sur ces sujets, et j'ai aussi demandé à une IA de me comparer ces frameworks. Celle-ci me les a comparés sur 5 points clés.

## 1. Définition et philosophie

Technologie	Type	Philosophie
<b>Angular</b>	Framework complet	Très structuré, basé sur TypeScript, conçu pour les grandes applications scalables.
<b>Vue.js</b>	Framework progressif	Flexible, facile à apprendre, peut être utilisé comme une simple librairie ou un framework complet.
<b>Next.js</b>	Framework basé sur React	Pensé pour la performance et le SEO, facilite le rendu côté serveur (SSR) et la génération statique (SSG).

## 2. Structure et complexité

Critère	Angular	Vue.js	Next.js
<b>Courbe d'apprentissage</b>	Compiqué au début	Facile à prendre en main	Modérée (React requis)
<b>Architecture imposée</b>	Oui	Non (Liberté totale)	Oui (Convention over Configuration)
<b>Langage principal</b>	Type Script	JavaScript (ou Type Script)	JavaScript / Type Script

Remarque :

- Angular est **très structuré** et peut sembler lourd au départ.
- Vue est **flexible et intuitif**.



- Next.js s'appuie sur React, donc il nécessite de connaître d'abord React.

### 3. Rendu et performance

Critère	Angular	Vue.js	Next.js
Rendu côté client (CSR)	✓	✓	✓
Rendu côté serveur (SSR)	✓ Mais pas nativement	✓ Mais pas nativement	✓
Static site Generation (SSG)	✗	✓ Mais pas nativement	✓
Performance	Le plus lourd	Léger et rapide	Le plus rapide

Next.js est le plus performant grâce à SSR et SSG, ce qui est un gros plus pour le SEO et les temps de chargement.

### 4. Cas d'usage idéaux

Besoin	Angular	Vue.js	Next.js
SaaS	✓	✖	✖
Petits projets	✗	✓	✓
Blog, e-commerces, ...	✗	✖	✓
Applications réactives en temps réel	✓	✓	✓

## 5. Communauté et écosystème

Critère	Angular	Vue.js	Next.js
<b>Popularité</b>	En baisse	Force croissance	Très populaire
<b>Écosystème</b>	Complet	Modulaire	Riche (Librairies supplémentaires)
<b>Soutien</b>	Google	Evan You et communauté	Vercel

Pour ma part, je pense qu'Angular a eu un coup de mou, mais est de retour depuis la v17, avec l'introduction des signals<sup>1</sup>.

### Conclusion : lequel choisir ?

- **Angular** → Si tu veux une architecture robuste, un framework complet et TypeScript imposé (idéal pour les grandes entreprises et les applications complexes).
- **Vue.js** → Si tu cherches simplicité, flexibilité et facilité d'apprentissage (parfait pour les startups et les projets évolutifs).
- **Next.js** → Si tu veux le meilleur SEO, des performances optimales et un bon équilibre entre structure et flexibilité (idéal pour les sites web dynamiques et les applications modernes).

---

<sup>1</sup> <https://angular.dev/guide/signals>

## **Ma réflexion après m’être renseigné sur ces 3 possibilités**

Le projet que j’ai à réaliser est quand même assez conséquent, ce n’est pas qu’un simple CRUD à réaliser. Je décide donc déjà de supprimer Vue.js, car il devient complexe sur les gros projets, car il n’a pas directement une structure imposée. Le code pourra donc très vite devenir difficile à maintenir. Nativement, il n’y a de bibliothèques intégrées nativement comparées à Angular, par exemple, qui intègre RxJS pour gérer les observables. Ces bibliothèques tierces à ajouter sont aussi moins puissantes comparé à la concurrence.

Next.js, donc React, est très bon pour ce qui est des performances et du SEO. Pour ma part, je n’ai pas du tout besoin de SEO, car l’application n’a pas besoin d’être référencée sur les moteurs de recherche.

Ce que j’aime bien avec Angular, c’est que c’est une solution tout-en-un. Il est très bien structuré pour les gros projets, qui ont besoin d’une architecture robuste. La courbe d’apprentissage n’est pas vraiment un problème pour moi, car je connais déjà bien Angular. J’ai réalisé quelques applications avec ce framework et j’ai pu aussi en faire en entreprise.

## 3.2 BACKEND

### a) PYTHON

#### Points positifs :

- **Simplicité et Lisibilité** : Python est réputé pour sa syntaxe claire et concise, facilitant le développement rapide et la maintenance du code.
- **Polyvalence** : Excellent pour le traitement de données, l'apprentissage automatique et le développement web avec des frameworks comme Django ou Flask.

#### Points Négatifs :

- **Performance** : Il peut être moins performant que les langages compilés pour certaines tâches intensives.
- **Gestion de la Concurrency** : La gestion des opérations parallèles peut être complexe en raison du GIL (Global Interpreter Lock).

### b) NODE.JS

#### Points positifs :

- **Performance Asynchrone** : Excellente gestion des opérations d'entrée/sortie asynchrones, idéal pour les applications à forte charge.
- **Écosystème NPM** : Vaste répertoire de packages facilitant le développement et l'intégration avec diverses bases de données.

#### Points Négatifs :

- **Gestion CPU Intensive** : Moins adapté pour les tâches nécessitant beaucoup de calculs.

**c) C#**

**Points Positifs :**

- **Performances Élevées** : Langage compilé offrant d'excellentes performances, particulièrement avec .NET Core.
- **Typage Fort** : Réduit les erreurs à l'exécution et améliore la maintenabilité du code.

**Points Négatifs :**

- **Dépendance à l'Écosystème Microsoft** : Peut limiter la flexibilité dans certains environnements.
- **Courbe d'Apprentissage** : Plus complexe à apprendre comparés à Python ou JavaScript.

**d) JAVA SPRING BOOT**

**Points Positifs :**

- **Robustesse et Scalabilité** : Excellent pour les grosses applications.
- **Auto-configuration** : Réduit considérablement le temps de configuration et de déploiement.
- **Intégration Facile** : Supporte nativement de nombreuses bases de données et offre une excellente intégration via Spring Data JPA.

**Points Négatifs :**

- **Complexité** : Peut être perçu comme complexe pour les petits projets ou les débutants.
- **Taille de l'Application** : Les applications Spring Boot peuvent être plus volumineuses en raison des dépendances incluses.

### 3.3 BASE DE DONNÉES

Il y a deux types de base de données : le relationnelle (SQL) et le non-relationnelle (NoSQL). Le SQL stocke les données sous forme de tables, les valeurs peuvent être reliées grâce à des clés primaires et étrangères. Tandis que le NoSQL les stocke sous tous types d'objets (documents, graphes, colonnes, clés-valeurs). Le NoSQL est plutôt utilisé pour stocker une très grande quantité de données "big data". Ce n'est pas vraiment ce qui nous intéresse, donc je vais seulement m'intéresser aux solutions SQL.

Je vais comparer MySQL (MariaDB) et PostgreSQL. Ce sont deux solutions open source qui permette de faire de l'SQL. Les fonctionnalités de ces deux bases de données diffèrent en termes de fonctionnalités, de performances et de cas d'utilisation. PostgreSQL est un système de gestion de base de données objet, ce qui veut dire qu'il est accessible par les langages orientés-objets comme s'il s'agissait des objets de ce langage<sup>2</sup>. Il gère bien les données grâce à des fonctionnalités comme le JSON et le XML. Ce que j'aime beaucoup aussi, c'est qu'il permet d'ajouter des types de données personnalisés.

MariaDB est un fork open source de MySQL. C'est un système de gestion de données léger et rapide. Contrairement à PostgreSQL, il ne gère pas les mêmes capacités d'extension et de gestion des données personnalisés. Ce qui fait sa force, ce sont ses performances sur les requêtes simples et est bien adapté pour les applications à fort trafic.

---

<sup>2</sup> [https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_de\\_gestion\\_de\\_base\\_de\\_donn%C3%A9es\\_relationnel-objet](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es_relationnel-objet)

## 3.4 CHOIX ET JUSTIFICATIONS

Pour le frontend, je choisis donc de partir sur Angular, car il gère mieux les gros projets et que j'ai plus de connaissance en ce framework. Je pense aussi que les signals introduits dans la v17 d'Angular va permettre à Angular de redevenir le numéro 1 !

Pour l'application de mon envergure, un backend Python n'est pas adapté. Il est plus fait pour la gestion de données en générale. En d'autres termes, il y a mieux. Mon choix se porte surtout en C# et Java Spring Boot. Ce sont deux langages que j'apprécie et que j'ai déjà pu tester. Je pense que je vais choisir Java Spring Boot, car Java est un langage que j'ai beaucoup utilisé ces deux dernières années et j'ai déjà pu pratiquer Java Spring Boot en stage en entreprise.

Mon choix du système de base de données se porte sur PostgreSQL. Bien que les deux systèmes soient puissants et adaptés à mon cas d'utilisation, j'ai préféré choisir PostgreSQL principalement pour sa flexibilité et ses fonctionnalités avancées, telle que la gestion des données complexes (JSON, XML). Même si ma décision ne repose pas sur un besoin spécifique mais plutôt sur un choix personnel, PostgreSQL semble un choix bien adapté à mon projet.

## CHAPITRE 4 : PROTOTYPE

Comme le but de mon projet de semestre est d'élaborer un cahier des charges et de choisir les technologies que je vais utiliser, j'ai décidé d'élaborer un prototype pour tester Angular, Java Spring Boot et PostgreSQL.

### 3.5 ARCHITECTURE GLOBALE

La base de données a été configurée dans un container Docker pour simplifier le déploiement et l'environnement de base de données. Docker permet de garantir une consistance entre les environnements de développement, tout en offrant une flexibilité dans la gestion des configurations.

Le backend a été développé avec Spring Boot, un framework Java qui facilite le développement rapide d'application web. Spring Boot offre un ensemble d'outils et de configurations qui permettent de se concentrer davantage sur la logique métier. Avec Spring Data JPA, il est facile d'intégrer PostgreSQL.

Les communications entre le backend Spring Boot et mon frontend Angular se fait via HTTP RESTful, en utilisant la librairie RxJS fournit avec Angular. Elle gère les requêtes asynchrones et le backend va récupérer les données depuis la base PostgreSQL.

Le but à terme est de pouvoir containerisé chaque partie de l'application via Docker. Pour ce prototype, le frontend ne l'est pas encore. Pour l'instant, j'ai un container pour la base de données et un autre pour le backend. Une documentation API sera aussi ajoutée par la suite.



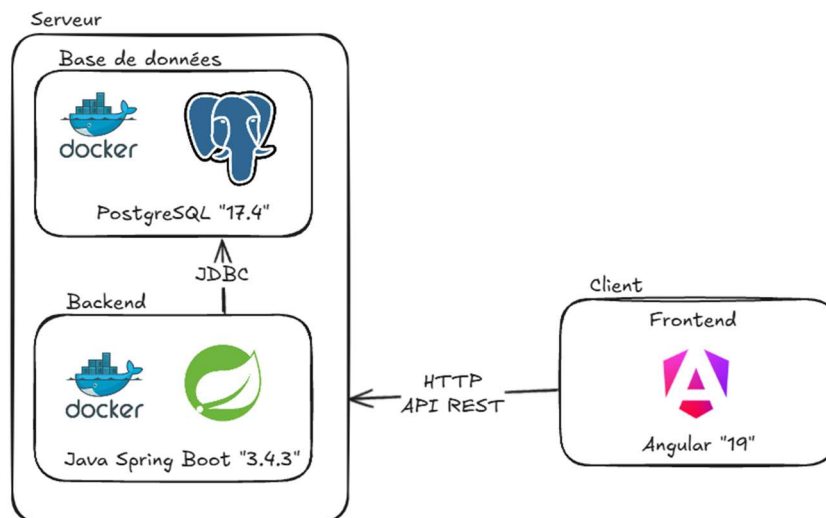


Figure 1, schéma du fonctionnement du prototype

Nous pouvons donc constater que la base de données ainsi que le backend est conteneurisé. Ce n'est pas encore le cas du frontend, mais c'est prévu pour la partie Bachelor. Il faudra aussi sécuriser les secrets via un fichier d'environnement, et même, si possible, via vault de Kubernetes qui est un gestionnaire de secret.

Voici à quoi ressemblera l'architecture finale :

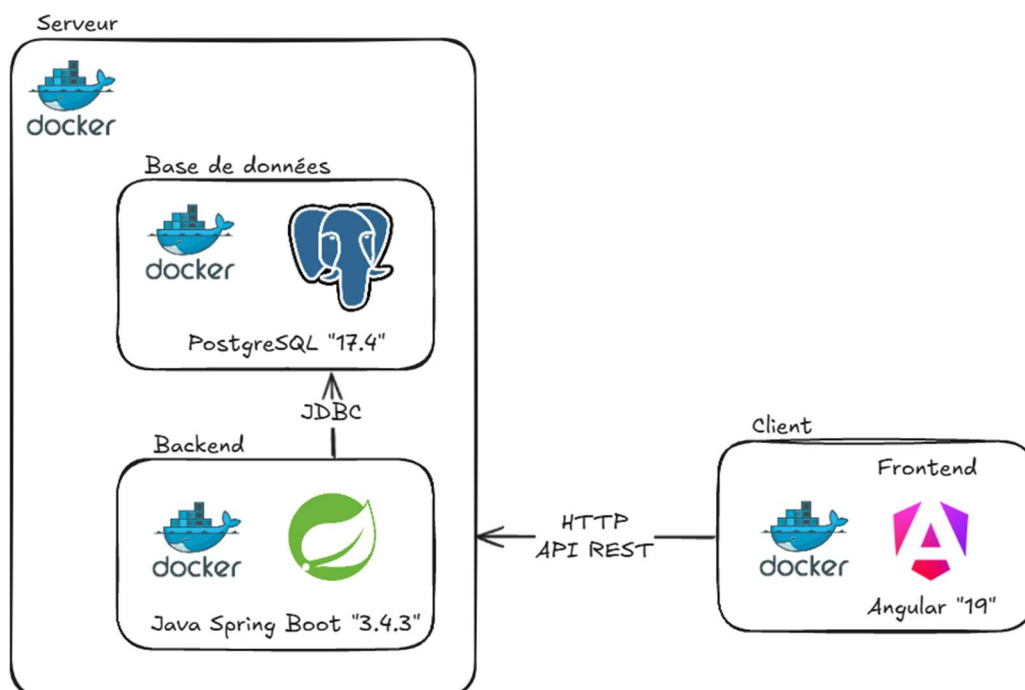


Figure 2, Schéma final de l'application

## 3.6 EXPLICATION DES PROCESSUS CLÉS

### a) FRONTEND

Pour le frontend, j’ai donc utilisé Angular 19 comme Framework. J’ai couplé Angular à quelques librairies pour les icones, j’ai utilisé FontAwesome. Pour le CSS, j’ai utilisé TailwindCSS et Daisyui (un plugin tailwindcss qui crée des composants réutilisable). NgRX SignalStore comme gestionnaire d’état de l’application et Eslint comme linter pour utiliser les bonnes pratiques de code d’Angular.

Au niveau de l’architecture, j’ai séparé le code en 3 parties distincts :

- **Core/**
  - Cette partie contient tout ce qui touche à l’infrastructure de mon application. Il y a donc les modèles de données, les ports et les adapters.
  - Un port est une classe abstraite, générique, qui contient les fonctions utilisées par un service. L’adapter va permettre d’implémenter ce port comme il le souhaite.
  - Prenons l’exemple de la TODO list. Ma classe abstraite (port) est “TodoService”. Je peux donc l’implémenter de deux manières différentes, une classe pour stocker dans le local storage “LocalStorageTodoService” et une autre avec des requêtes HTTP “HttpTodoService”. Dans mon code, je n’utilise que mon port (TodoService) et dans ma config, je choisis quelle implémentation je vais utiliser.

- **Shared/**

- Contient tout ce qui pourrait être réutilisé dans mon projet, cela pourrait être des composants, des directives, des pipes, des services, ...

- **Features/**

- Features contient tous les écrans de mon application. Chaque sous-dossier contenu dans ce répertoire, est un écran visible dans mon application. Si j'ai besoin de créer des composants spécifiques à un écran, je crée sous dossier dans ce dossier.

- Par exemple :

features/todo-list/components/...

Depuis Angular 14, le fonctionnement “standalone” est utilisée. Chaque librairie utilisée est maintenant déclarée directement dans le composant et plus dans un module qui est ensuite importé dans un ou plusieurs composants.

Comme énoncé plus haut, j'ai utilisé un gestionnaire d'état, NgRX SignalStore. C'est un gestionnaire d'état qui se base sur la nouveauté d'Angular 17, les signaux. C'est un wrapper autour d'une valeur qui notifie le système de détection de changement d'angular quand une valeur change (afin de rafraîchir l'affichage). Je peux donc gérer les états de mon application. Ce projet reste assez simple, je n'ai donc qu'un état pour gérer mes TODOs, mais on peut imaginer un état d'application (authentification) et plusieurs états pour les parties de l'application. Mon gestionnaire d'état des TODOs contient donc, une liste de TODOs et un booléen “loading”. Elle permet de notifier quand une requête est effectuée pour, par exemple, afficher un spinner.

**b)        BACKEND**

J'ai utilisé Spring initializr pour générer mon projet. J'ai rajouté certaines dépendances pour le bon fonctionnement du backend.

Dépendances	Informations
<b>Spring data JPA</b>	Faciliter les accès aux données dans Spring.
<b>Spring Boot Starter Security</b>	Sécuriser l'accès aux endpoints.
<b>Spring Boot Starter Validation</b>	Valider les données (exemple : un mot de passe doit contenir minimum 8 caractères)
<b>Spring Boot Starter Web</b>	Fournit toutes les fonctionnalités pour créer une APIRest
<b>Spring Boot Docker Compose</b>	Containeriser le backend
<b>Postgresql</b>	Drivers pour accéder à la base de données Postgres
<b>Lombok</b>	Réduire le code redondant (Getter, Setter, ...)

L'architecture utilisée est « en couche ». Elle permet de structurer le projet en plusieurs couches logiques.

- **Controller** – gère les requêtes HTTP et les réponses.
- **Service** – Contient la logique métier.
- **Repository** – Interagit avec la base de données.
- **Model** – Contient les classes représentant les données.

Cette structure sera aussi utilisée lors du travail de Bachelor. Elle évoluera afin de gérer les erreurs correctement avec un Handler.

## CONCLUSION

Ce projet avait pour objectif de concevoir un cahier des charges détaillé pour la réalisation d'une application de gestion des clubs sportifs, plus particulièrement Chênois Volleyball, pour centraliser les outils et améliorer l'expérience utilisateur.

Dans un premier temps, l'élaboration du cahier des charges avec l'aide de Gaëtan a permis de définir plus précisément les besoins fonctionnels et technique de l'application. Cette étape a été essentielle pour structurer le projet, en détaillant les fonctionnalités clés, les utilisateurs cibles, et les exigences en matière de sécurité et ergonomie.

Ensuite, l'étude des solutions existantes a permis d'analyser les plateformes déjà présentes sur le marché. Cette comparaison a permis de mettre en évidence leurs avantages, mais aussi leurs limites, notamment en termes d'ergonomie, d'accessibilité et de fonctionnalités. J'ai pu aussi m'inspirer des avantages de chacun, afin d'y adapter à mes besoins plus tard.

L'exploration des technologies existantes a permis d'explorer beaucoup de possibilités, toutes presque aussi bonne les unes que les autres. Ce choix a été surtout porté par des préférences personnelles.

Enfin, la conception d'un prototype a permis une première prise en main des technologies choisies précédemment. Elle a aussi permis de se familiariser avec différentes bibliothèques et dépendances ajoutées afin d'en apprendre davantage.

Ce document servira de base solide pour le développement futur lors de la partie travail de Bachelor. Le prochain objectif est de développer l'application, de sécuriser tous les accès, d'effectuer les tests nécessaires et à préparer le déploiement. Ce projet représente donc une avancée significative dans l'optimisation de la gestion avec une approche moderne et centralisée qui répond aux besoins du club.

## **ANNEXES (STYLE « TITRE 1 »)**

*Imprimer idéalement cette page sur une page de couleur.*

*Chaque annexe doit commencer sur une nouvelle page et doit être numérotée : Annexe 1 puis Annexe 2, etc.*

## **ANNEXE 1**



## **ANNEXE 2**

## **ANNEXE 3**

## RÉFÉRENCES DOCUMENTAIRES (STYLE « TITRE 1 »)

*Sites Web consultés – Code repris d’ailleurs – Notices techniques – Articles de presse – Ouvrage imprimés – Ouvrages électroniques – Chapitre dans un ouvrage imprimé – Rapports imprimés – Travaux universitaires – Articles de revues imprimés – Articles de périodiques électroniques – Communication dans un congrès. Pour chacun de ces types de document, les mise en forme sont dans le document « Méthode de citation et de rédaction d’une bibliographie ».*

*Afin de gagner du temps, pensez à utiliser le logiciel de gestion bibliographique Zotero pour la mise en forme et l’édition automatique de vos références à la norme ISO690.*