

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



Projet de semestre présentée par

Capt Thibault

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

Mars, 2025

Professeur-e HES responsable

Eggenberg Niklaus

Mandant

Chênois Volley

Légende et source de l'illustration de couverture :

Illustration 1, Divers athlètes en actions

Source : URL01

TABLE DES MATIÈRES

Remerciements	vi
Résumé	vii
Liste des acronymes	viii
Liste des illustrations	x
Liste des tableaux	xi
Introduction	13
Chapitre 1 : Élaboration et définition du cahier des charges	16
1.1 But du projet	16
1.2 Exigences	17
a) Fonctionnalités principales	17
b) Spécifications techniques	18
c) Fonctionnalités secondaires	18
1.3 Livrable	19
Chapitre 2 : Étude de marché	20
2.1 Analyse comparative	20
a) SportEasy	20
b) TeamSnap	21
c) Spond	22
d) Heja	23
e) TeamLinkt	24
f) MonClubSportif	25
2.2 Manques ou limitations	27
a) Absence de centralisation totale	27
b) Manque de personnalisation	27
c) Interfaces datées ou peu intuitives	27
d) Abonnements payants et fonctionnalités limitées	27
e) Faible intégration de la sécurité et de la vie privée	28
f) Faibles fonctionnalités autour des parents	28
2.3 Inspirations	29
a) Centralisation des fonctionnalités (SportEasy, TeamSnap)	29
b) Gestion des présences et notifications (MonClubSportif, Spond)	29
c) Communication supervisée (MonClubSportif)	30
d) Statistiques et suivi des performances (SportEasy)	30
e) Design épuré et « mobile-friendly » (Heja, TeamLinkt)	30
Chapitre 3 : Exploration des différentes technologies à disposition	31
3.1 Frontend	31
a) Définition et philosophie	32
b) Structure et complexité	32
c) Rendu et performance	33
d) Cas d'usage idéaux	33
e) Communauté et écosystème	34
3.2 Backend	35

a)	Python.....	35
b)	Node.js.....	35
c)	C#.....	36
d)	Java Spring Boot.....	36
3.3	Base de données	37
3.4	Choix et justifications	38
Chapitre 4 : Prototype		39
4.1	Architecture globale	39
4.2	Architecture détaillée.....	41
a)	Frontend	41
b)	Backend	43
4.3	Réalisation.....	44
Conclusion		45
Annexes		46
Références documentaires		49

À ma famille, mes amis et mes proches,
Votre soutien inestimable m'a permis de mener à bien ce projet.

Je vous dédicace ce travail en signe de reconnaissance.

Merci.

REMERCIEMENTS

Je tiens tout d’abord à exprimer ma gratitude envers toutes les personnes qui m’ont soutenu et accompagné tout au long de ce projet.

Tout d’abord, je remercie chaleureusement ma famille pour leur soutien inconditionnel et leur encouragement constant.

Je souhaite également remercier mes amis pour leurs précieux conseils et leur soutien moral pendant les moments plus difficiles.

Un grand merci à mon enseignant M. Eggenberg, qui m’a guidé avec bienveillance et professionnalisme. Ses remarques et conseils avisés m’ont été d’une très grande aide.

RÉSUMÉ

Ce mémoire présente la conception et la réalisation d'un cahier des charges pour le développement d'une application web de gestion de club sportif, avec un premier cas d'utilisation pour Chênois Volley. L'objectif est de centraliser les outils de gestion des présences, des événements, des cotisations et des communications afin d'améliorer l'organisation et la coordination au sein du club. Dans un premier temps, une étude des solutions existantes sur le marché a été réalisée. Elle a permis d'identifier leurs points forts ainsi que leurs limitations. Les principales lacunes relevées sont un manque de centralisation, des fonctionnalités limitées sans abonnement et une personnalisation insuffisante pour correspondre aux besoins des clubs. Ensuite, une exploration des technologies a permis de choisir Angular pour le frontend, grâce à sa structure robuste, Java Spring Boot pour le backend, un très bon outil en raison de ses bonnes performances et de sa scalabilité, et PostgreSQL pour la base de données, en raison de sa flexibilité dans la gestion des données. Enfin, un prototype a été développé pour tester l'architecture globale, avec une base de données et une API REST, conteneurisées avec Docker, permettant la communication avec le frontend. Il s'agit d'une « to-do list », avec la possibilité d'agir sur chaque tâche afin de lire, créer, supprimer et barrer les tâches réalisées. Il a permis de valider les technologies choisies précédemment et d'assurer leur intégration. Ce travail pose les bases du projet de Bachelor, qui consistera à finaliser l'application en ajoutant des fonctionnalités avancées et en optimisant la sécurité. L'outil vise à simplifier la gestion des clubs sportifs en offrant une plateforme complète, performante et intuitive.



Candidat-e :

Professeur-e(s) responsable(s) :

CAPT THIBAUT

EGGENBERG NIKLAUS

Filière d'études : ISC

En collaboration avec : Chênois Volley

Travail de semestre soumis à une convention de stage en entreprise : non

Travail soumis à un contrat de confidentialité : non

LISTE DES ACRONYMES

1. **CSR** - Client-Side Rendering
2. **SSR** – Server-Side Rendering
3. **SSG** – Static Site Generation
4. **NPM** - Node Package Manager
5. **RxJS** – Reactive Extensions for JavaScript
6. **API** – Application Programming Interface
7. **JPA** – Java Persistence API
8. **SQL** – Structured Query Language
9. **NoSQL** – Not Only SQL
10. **JSON** – JavaScript Object Notation
11. **XML** – eXtensible Markup Language
12. **AVS** – Assurance Vieillesse et Survivants
13. **CRUD** – Create, Read, Update, Delete
14. **NgRX** – Angular Reactive Extensions
15. **GIL** – Global Interpreter Lock
16. **SEO** – Search Engine Optimization
17. **C#** – C-Sharp (Langage de programmation)
18. **JPA** – Java Persistence API
19. **IDE** – Integrated Development Environment
20. **SSL** – Secure Sockets Layer

- 21. **Docker** – Plateforme de conteneurisation
- 22. **Kubernetes** – Système d'orchestration de conteneurs
- 23. **SMTP** – Simple Mail Transfer Protocol
- 24. **TLS** – Transport Layer Security

LISTE DES ILLUSTRATIONS

Illustrations

Illustration 1, Divers athlètes en actions	ii
Illustration 2, Schéma d'application du prototype réalisé	40
Illustration 3, Schéma final de l'application	40
Illustration 4, Accueil sur la "to-do list"	44
Illustration 5, « To-do list » avec plusieurs tâches cochées ou non	44

Références des URL

URL01	Illustration, Pinterest [en ligne]. Disponible à l'adresse : https://it.pinterest.com/pin/8444318046209688/ [consulté le 28 mars 2025].
-------	--

LISTE DES TABLEAUX

Tableaux

Tableau 1, Comparaison les principaux frameworks frontend.....	32
Tableau 2, Comparaison des courbes d'apprentissage des frameworks frontend.....	32
Tableau 3, Comparaison des rendus et performances des frameworks frontend.....	33
Tableau 4, Comparaison des cas d'usage idéaux des frameworks frontend	33
Tableau 5, Comparaison de la communauté et de l'écosystème des frameworks frontend.	34
Tableau 6, Présentation des dépendances utilisées dans le prototype.....	43

Références des URL (des données des tableaux)

URL02	CHAHINE, Hugo, 2024. ReactJS vs Angular vs VueJS : Que choisir en 2024 ? <i>Ambient IT</i> [en ligne]. 13 octobre 2024. Disponible à l'adresse : https://www.ambient-it.net/reactjs-vs-angular-vs-vuejs/ [consulté le 5 décembre 2024].
URL03	ÉQUIPE DYMA, 2024. Angular vs Vue.js vs React en 2024. <i>Blog de Dyma</i> [en ligne]. 5 septembre 2024. Disponible à l'adresse : https://dyma.fr/blog/quel-framework-choisir-angular-vue-js-ou-react/ [consulté le 5 décembre 2024].
URL04	MOHIT, Joshi, 2023. Angular vs React vs Vue: Core Differences. <i>BrowserStack</i> [en ligne]. 11 mai 2023. Disponible à l'adresse : https://browserstack.wpengine.com/guide/angular-vs-react-vs-vue/ [consulté le 5 décembre 2024].
URL05	TANAY KUMAR, Deo, 2023. Angular vs React vs Vue: Key Differences. <i>LambdaTest</i> [en ligne]. 4 octobre 2023. Disponible à l'adresse : https://www.lambdatest.com/blog/angular-vs-react-vs-vue/ [consulté le 5 décembre 2024].

URL06 THIBAUT (CTO), 2023. Vue JS, Angular ou react : quelle technologie front-end choisir ? *Access it* [en ligne]. 13 avril 2023. Disponible à l'adresse : <https://www.access-it.fr/actualite/vuejs-angular-ou-react/> [consulté le 5 décembre 2024].

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 extrasportifs... Il y a souvent une multitude d'outils utilisés. Chaque tâche a 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 obligeant à retranscrire manuellement les informations dans leur propre calendrier, afin d'éviter tout oubli. 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 a 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 et aux entraîneurs, d'avoir une solution intuitive pour créer des événements et 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.

Pour mener à bien ce projet, j'ai adopté une méthodologie rigoureuse basée sur un suivi structuré. J'ai tenu un journal de bord, disponible en annexe, et organisé des rendez-vous hebdomadaires avec mon enseignant afin d'assurer une progression constante. Mon travail a débuté par une étude de marché afin d'identifier les solutions existantes. Ensuite, je me suis renseigné sur les technologies disponibles qui pourraient répondre aux besoins du projet. Pour valider mes choix technologiques, j'ai réalisé un prototype permettant d'évaluer la faisabilité et l'efficacité des outils sélectionnés. Enfin, j'ai élaboré un cahier des charges robuste afin de garantir une vision claire et détaillée du projet.

Ce rapport constitue une étape préalable à la réalisation de mon travail de Bachelor en Informatique et Systèmes de Communications à La Haute École 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 Volley, ce projet a pour but de concevoir une application adaptée aux besoins de club. L'outil sera restreint aux exigences et au fonctionnement de Chênois, 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 événements, 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 à l'égard 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 servira de base pour la réalisation 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 allé à la rencontre de Gaëtan Fetter, directeur sportif de Chênois Volley. Il avait, au préalable, réalisé une ébauche des fonctionnalités à implémenter. Celle-ci est disponible 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 Volley.

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'événements extrasportifs.
- Mettre en place un système de notifications pour les rappels d'événements.
- 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 événements
 - a. Calendrier commun pour tous les événements 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'entraîneur et les joueurs).
 - c. Accès aux calendriers des équipes.
 - d. Accès aux logs de l'application.
4. Entraîneur
 - a. Création d'événements au sein de son équipe.
 - b. Préparer les convocations aux événements.
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. **Connexion** : Email et mot de passe.
 - b. **Rôles** : Directeur sportif, Entraîneur, Joueur / Parent.
2. Création de comptes 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 détaillée
 - a. Rapport de suivi des retards et des absences.
 - b. Exporter au format CSV les données des joueurs présents à l'entraînement pour pouvoir, par la suite, les importer dans la Banque de Données Nationale pour le Sport (BDNS).
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 informations des joueurs via l'API VolleyManager¹.
 - b. Gestion des cotisations.

¹ Outil permettant de gérer les informations des joueurs de volleyball.

1.3 LIVRABLE

Les solutions finales de ce projet vont inclure :

- Une application web fonctionnelle permettant la gestion des présences, des événements 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 DE MARCHÉ

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

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

2.1 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 grâce à 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 gestion de programme, 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 paiement 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 partenariats, permettant aux clubs d'attirer et de gérer des partenariats financiers.

La version mobile de TeamSnap inclut des fonctionnalités intéressantes, tel 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 outil de planification, qui permet une planification efficace des événements tels que les entraînements, les matchs, et autres activités liées à l'équipe.

En matière 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'application 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 téléphone.

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 outil de planification 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 un "team app", offrant un accès centralisé à toutes les informations importantes pour les joueurs et entraîneurs.

Un autre atout majeur de TeamLinkt est 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 tableau de bord, épuré et bien organisé, 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'activité 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.

2.2 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). 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 jouer 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.

2.3 INSPIRATIONS

a) CENTRALISATION DES FONCTIONNALITÉS (SPORT EASY, TEAM SNAP)

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 (MON CLUB SPORTIF, 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 email 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.

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 la version mobile et la version bureau.

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.

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 TypeScript qui vise à améliorer JavaScript en ajoutant une syntaxe stricte à celui-ci. Toutes les trois technologies prennent en charge TypeScript. 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 comparée à Angular et Vue.js. Je vais donc plutôt comparer Angular et Vue.js avec Next.js qui sont une solution basée sur React. Comme ça, la comparaison est plus pertinente, car ce sont trois environnements distincts.

Pour mes recherches, j'ai lu beaucoup d'articles sur ces sujets. J'ai aussi demandé l'avis d'une IA pour m'éclairer sur ce sujet. En combinant les deux, voici ce que j'ai retenu.

a) DÉFINITION ET PHILOSOPHIE

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

Tableau 1, Comparaison les principaux frameworks frontend

Source : Données tirées des sources URL02 et URL03. Tableau réalisé par Capt Thibault.

b) STRUCTURE ET COMPLEXITÉ



Critère	Angular	Vue.js	Next.js
Courbe d'apprentissage	Complicé au début	Facile à prendre en main	Modérée (React requis)
Architecture imposée		 (Liberté totale)	 (Convention Over Configuration)
Langage principal	Type Script	JavaScript (Ou Type Script)	JavaScript Et Type Script

Tableau 2, Comparaison des courbes d'apprentissage des frameworks frontend

Source : Données tirées des sources URL03 et URL04. Tableau réalisé par Capt Thibault.

c) RENDU ET PERFORMANCE

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

Tableau 3, Comparaison des rendus et performances des frameworks frontend

Source : Données tirées des sources URL04 et URL05. Tableau réalisé par Capt Thibault.

d) CAS D'USAGE IDÉAUX

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

Tableau 4, Comparaison des cas d'usage idéaux des frameworks frontend

Source : Données tirées des sources URL05 et URL06. Tableau réalisé par Capt Thibault.

e) COMMUNAUTÉ ET ÉCOSYSTÈME

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

*Tableau 5, Comparaison de la communauté et de l'écosystème des frameworks frontend**Source : Données tirées des sources URL02 et URL06. Tableau réalisé par Capt Thibault.*

Le projet que j'ai à réaliser est quand même assez conséquent, ce n'est pas qu'un simple CRUD² 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 pas de librairies intégrées comparées à Angular, par exemple, qui intègre RxJS³ pour gérer les observables. Ces librairies tierces à ajouter sont aussi moins puissantes, comparées à la concurrence.

Next.js, basé sur React, obtient de très bonnes performances et possède un très bon référencement SEO. Ce n'est pas une priorité dans mon projet, car l'application n'a pas besoin d'être référencée sur les moteurs de recherche.

Une des principales forces d'Angular réside dans son écosystème complet pour le développement d'applications web. 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.

² Create, Read, Update, Delete.

³ Librairie JavaScript permettant la programmation réactive et la gestion des flux de données asynchrones dans Angular.

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.⁴

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.

⁴ Global Interpreter Lock

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.
- **Open source** : Depuis 2014, .NET est passé open source. Ce qui veut dire qu'il ne faut plus acheter de licence pour commercialiser les applications .NET.

Points négatifs :

- **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.
- **Autoconfiguration** : 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.
- JRE est maintenant payant : Il faut acheter une licence Oracle pour utiliser Java Runtime Environment dans les fonctions destinées à être commercialisées

3.3 BASE DE DONNÉES

Il y a deux types de bases de données : le relationnel (SQL) et le non-relationnel (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 du 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-objet comme s'il s'agissait des objets de ce langage. Il gère bien les données grâce à des fonctionnalités comme le JSON et le XML. Un autre atout majeur de PostgreSQL, 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 personnalisées. Ce qui fait sa force, ce sont ses performances sur les requêtes simples et sont bien adaptés pour les applications à fort trafic.

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 signaux introduits dans la v17 d'Angular vont permettre à Angular de briller à nouveau.

Pour l'application de mon envergure, un backend Python n'est pas adapté. Il est plus fait pour la gestion de données en général. 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, telles 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.

Je vais aussi utiliser Docker afin de conteneuriser l'application pour assurer une meilleure portabilité et faciliter le déploiement. Docker permettra d'uniformiser l'environnement de développement et d'éviter tous problèmes liés aux différentes configurations. Il y aura un conteneur pour chaque partie : Le frontend, le backend et la base de données.

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. Le code source du projet est disponible sur le git du projet de semestre, dans le dossier *todo-workspace*⁵.

4.1 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 font via HTTP RESTful, en utilisant la librairie RxJS fournie 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 containeriser 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.

⁵ Disponible à la source : <https://gitedu.hesge.ch/niklaus.eggenber/capt-24-25>

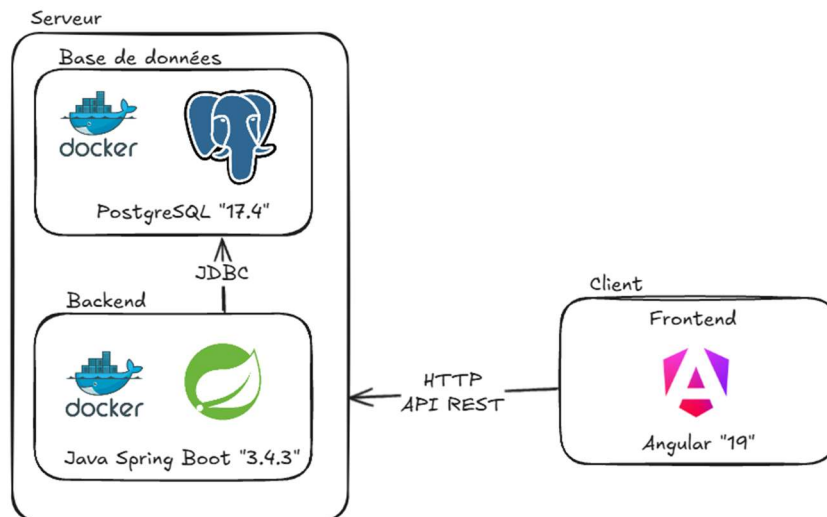


Illustration 2, Schéma d'application du prototype réalisé

Source : Réalisé par moi-même

Nous pouvons donc constater que la base de données ainsi que le backend est conteneurisée. 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.

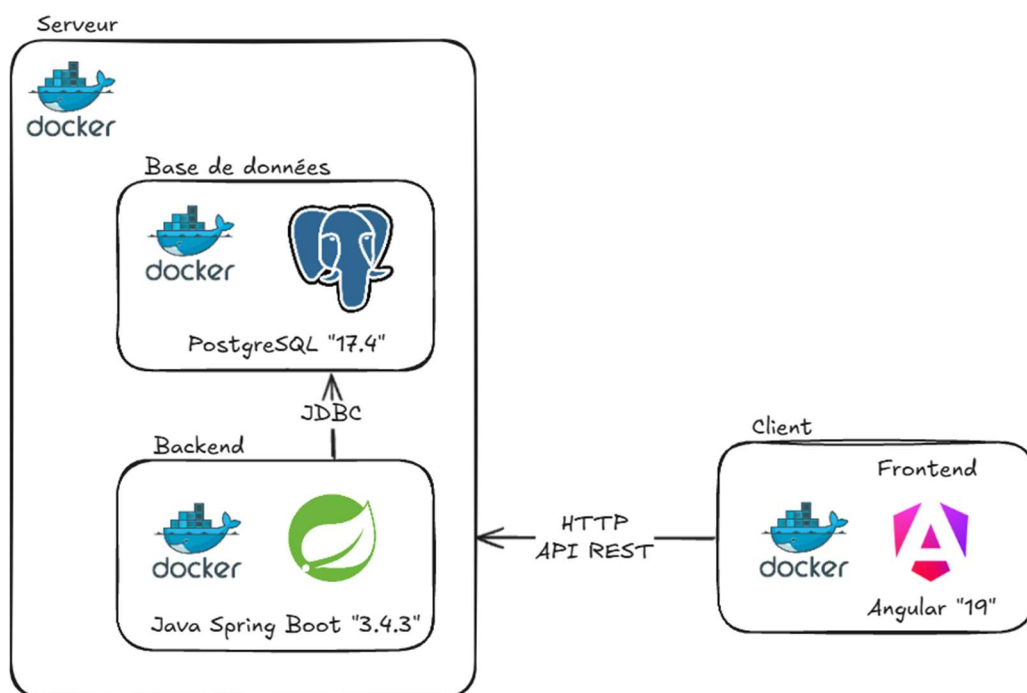


Illustration 3, Schéma final de l'application

Source : Réalisé par moi-même

4.2 ARCHITECTURE DÉTAILLÉE

a) FRONTEND

Pour le frontend, j'ai donc utilisé Angular 19 comme Framework. J'ai couplé Angular à quelques librairies pour les icônes, j'ai utilisé FontAwesome. Pour le CSS, j'ai utilisé TailwindCSS et Daisyui (un plug-in tailwindcss qui crée des composants réutilisables). NgRX SignalStore⁶ 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 trois parties distinctes :

- **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 adapter.
 - 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 « to-do list ». Ma classe abstraite (port) est "*ToDoService*". Je peux donc l'implémenter de deux manières différentes, une classe pour stocker dans le stockage local au navigateur "*LocalStorageToDoService*" et une autre avec des requêtes HTTP "*HttpToDoService*". Dans mon code, je n'utilise que mon port (*ToDoService*) et dans ma configuration (*app.config.ts*), je choisis quelle implémentation je vais utiliser.

⁶ Gestionnaire d'état basé sur la nouvelle API des signaux, permettant une gestion réactive et optimisée des données.

- **Shared/**
 - Contiens 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 utilisé. 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 une enveloppe autour d'une variable qui notifie le système de détection de changement d'Angular quand une valeur vient à changer (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 tâches, mais on peut imaginer un état d'application (authentification) et plusieurs états pour les parties de l'application. Mon gestionnaire d'état des tâches contient donc, une liste de to-do 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
Spring data JPA	Faciliter les accès aux données dans Spring.
Spring Boot Starter Security	Sécuriser l'accès aux endpoints.
Spring Boot Starter Validation	Valider les données (exemple : un mot de passe doit contenir minimum huit caractères)
Spring Boot Starter Web	Fournir toutes les fonctionnalités pour créer une APIRest
Spring Boot Docker Compose	Containeriser le backend
Postgresql	Drivers pour accéder à la base de données Postgres
Lombok	Réduire le code redondant (Getter, Setter ...)
Springdoc-openapi	Génère automatiquement une documentation API

Tableau 6, Présentation des dépendances utilisées dans le prototype

Source : Tableau réalisé par Capt Thibault

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** – contiennent 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.

4.3 RÉALISATION

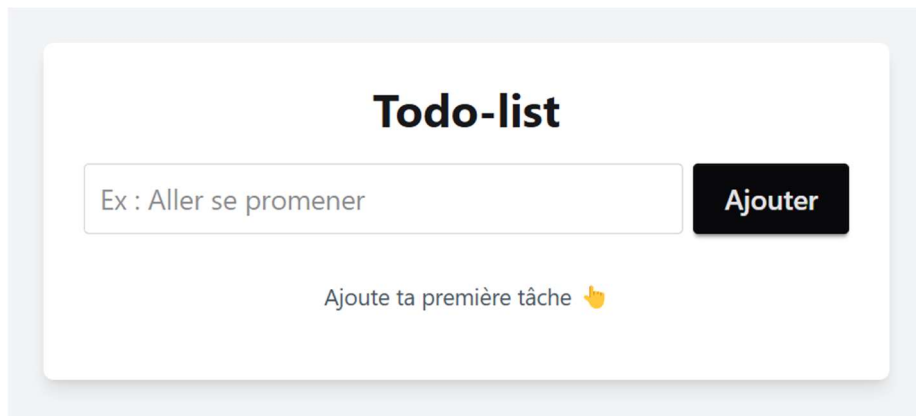


Illustration 4, Accueil sur la "to-do list"

Source : Réalisé par Capt Thibault

Voici la page d'accueil de la to-do list. Un message s'affiche lorsqu'aucune tâche n'est encore ajoutée. Il suffit donc d'écrire une tâche dans le champ disponible et d'appuyer sur ajouter. Par défaut, dans la liste, il n'est pas cocher, mais en cliquant sur la case à cocher, il est possible de le barrer.

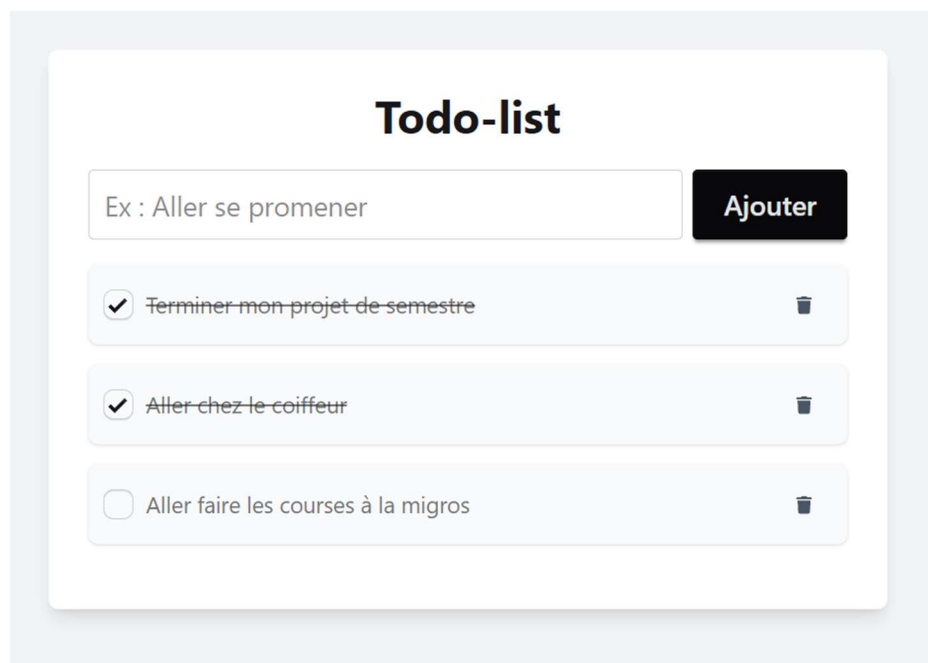


Illustration 5, « To-do list » avec plusieurs tâches cochées ou non

Source : Réalisé par Capt Thibault

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 Volley, 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 techniques 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 bonnes 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 librairies et dépendances ajoutées afin d'élargir mes connaissances sur le sujet.

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 de préparer le déploiement. Ce projet représente une avancée significative dans l'optimisation de la gestion avec une approche moderne et centralisée qui répond aux besoins du club.

ANNEXES

Annexe 1 : Ébauche du cahier des charges de Gaëtan Fetter

Annexe 2 : Journal de bord

ANNEXE 1

ANNEXE 2

RÉFÉRENCES DOCUMENTAIRES

1. AMPHIBEE. L'app n°1 pour gérer une équipe ou un club de sport amateur. [en ligne]. Disponible à l'adresse : <https://www.sporteasy.net/fr/> [consulté le 5 décembre 2024].
2. APPMASTER, 2023. Travailler avec des bases de données en Go | AppMaster. *Travailler avec des bases de données en Go* [en ligne]. 23 juin 2023. Disponible à l'adresse : <https://appmaster.io/fr/blog/bases-de-donnees-en-go> [consulté le 5 décembre 2024].
3. APPMASTER, 2024. Pourquoi Golang est un choix de premier ordre pour le développement backend | AppMaster. *Pourquoi Golang est un choix de premier ordre pour le développement backend* [en ligne]. 16 août 2024. Disponible à l'adresse : <https://appmaster.io/fr/blog/golang-pour-le-backend> [consulté le 5 décembre 2024].
4. ARSENAULT, Cody, 2023. The Pros and Cons of 8 Popular Databases. *KeyCDN* [en ligne]. 3 mars 2023. Disponible à l'adresse : <https://www.keycdn.com/blog/popular-databases> [consulté le 5 décembre 2024].
5. CHAHINE, Hugo, 2024. ReactJS vs Angular vs VueJS : Que choisir en 2024 ? *Ambient IT* [en ligne]. 13 octobre 2024. Disponible à l'adresse : <https://www.ambient-it.net/reactjs-vs-angular-vs-vuejs/> [consulté le 5 décembre 2024].
6. COACHA. Coacha - « A good coach can change a game; A great coach can change a life ». *Coacha* [en ligne]. Disponible à l'adresse : <https://my.coacha.app> [consulté le 5 décembre 2024].

7. ÉQUIPE DYMA, 2024. Angular vs Vue.js vs React en 2024. *Blog de Dyma* [en ligne]. 5 septembre 2024. Disponible à l'adresse : <https://dyma.fr/blog/quel-framework-choisir-angular-vue-js-ou-react/> [consulté le 5 décembre 2024].
8. FARZANE, 2023. Which programming language is best for back-end? | Winatalent. *WINaTALENT Blog* [en ligne]. 27 octobre 2023. Disponible à l'adresse : <https://winatalent.com/blog/best-programming-language-for-back-end/> [consulté le 5 décembre 2024].
9. GOOGLE. *Angular* [en ligne]. Disponible à l'adresse : <https://angular.dev/> [consulté le 7 mars 2025].
10. HEJA, 2024. Heja | Sports team management and communication app. *Heja* [en ligne]. 2024. Disponible à l'adresse : <https://heja.io/> [consulté le 5 décembre 2024].
11. JAVA, 2023. Spring Framework vs. Spring Boot - avantages et inconvénients. *Spring Framework vs. Spring Boot – avantages et inconvénients* [en ligne]. 20 août 2023. Disponible à l'adresse : <https://vmsoftwarehouse.fr/spring-framework-vs-spring-boot-avantages-et-inconvenients> [consulté le 5 décembre 2024].
12. L'ÉQUIPE ÉDITORALE IONOS, 2023. MySQL vs MongoDB : comparatif des systèmes de base de données. *IONOS Digital Guide* [en ligne]. 18 janvier 2023. Disponible à l'adresse : <https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/mysql-vs-mongodb/> [consulté le 5 décembre 2024].
13. MICROSOFT. Qu'est-ce que Java Spring Boot ? – Présentation de Spring Boot | Microsoft Azure. *Qu'est-ce que Spring Boot ?* [en ligne]. Disponible à l'adresse : <https://azure.microsoft.com/fr-fr/resources/cloud-computing-dictionary/what-is-java-spring-boot> [consulté le 5 décembre 2024].

14. MOHIT, Joshi, 2023. Angular vs React vs Vue: Core Differences. *BrowserStack* [en ligne]. 11 mai 2023. Disponible à l'adresse : <https://browserstack.wpengine.com/guide/angular-vs-react-vs-vue/> [consulté le 5 décembre 2024].
15. MONCLUBSPORTIF. MonClubSportif - Logiciel de gestion d'équipes sportives et culturelles. *MonClubSportif* [en ligne]. Disponible à l'adresse : <https://monclubsportif.com/> [consulté le 5 décembre 2024].
16. MYDIGITALSCHOOL. Frameworks de développement web, avantages et inconvénients - Actualité Bordeaux - MyDigitalSchool. *Frameworks de développement web, avantages et inconvénients* [en ligne]. Disponible à l'adresse : <https://www.mydigitalschool.com/actualite-bordeaux/frameworks-de-developpement-web-avantages-et-inconvenients> [consulté le 5 décembre 2024].
17. OFFICE FÉDÉRAL DU SPORT OFSPO, 2022a. Guide d'intégration de l'API «standard» de la BDNS pour les partenaires de l'OFSPO. . p. 3.
18. OFFICE FÉDÉRAL DU SPORT OFSPO, 2022b. Importation du contrôle des présences (CdP). . p. 1.
19. OFFICE FÉDÉRAL DU SPORT OFSPO, 2024. Banque de données nationale pour le sport (BDNS). *Banque de données nationale pour le sport (BDNS)* [en ligne]. 21 novembre 2024. Disponible à l'adresse : <https://www.jugendundsport.ch/fr/banque-de-donnees-nationale-pour-le-sport-bdns> [consulté le 5 décembre 2024].
20. PRUD'HOMME, Frédéric, 2024. Angular vs React : quel est le meilleur framework ? *Qim info* [en ligne]. 18 janvier 2024. Disponible à l'adresse : <https://www.qiminfo.ch/angular-vs-react/> [consulté le 5 décembre 2024].

21. R, Nirina, 2022. Go : tout savoir sur ce langage de programmation. *LEBIGDATA.FR* [en ligne]. 30 décembre 2022. Disponible à l'adresse : <https://www.lebigdata.fr/go-langage-de-programmation> [consulté le 5 décembre 2024].
22. RACHOWICZ, Justyna, 2024. When, How, And Why Use Node.js as Your Backend? *When, How, And Why Use Node.js as Your Backend?* [en ligne]. 26 novembre 2024. Disponible à l'adresse : <https://www.netguru.com/blog/node-js-backend> [consulté le 5 décembre 2024].
23. SCOTT, Pete, 2024. What is PostgreSQL? Everything You Need to Know. *Percona Database Performance Blog* [en ligne]. 2 février 2024. Disponible à l'adresse : <https://www.percona.com/blog/what-is-postgresql-used-for/> [consulté le 5 décembre 2024].
24. SOLUTIONS, Mobio, 2024. Angular for Enterprises: Building Scalable and Secure Web Applications. *Medium* [en ligne]. 22 avril 2024. Disponible à l'adresse : <https://mobiosolutions.medium.com/angular-for-enterprises-building-scalable-and-secure-web-applications-452b7c90e582> [consulté le 5 décembre 2024].
25. SPOND, 2024. Spond- Meilleure application gratuite de gestion d'équipe sportive. *Spond* [en ligne]. 2024. Disponible à l'adresse : <https://www.spond.com/fr/> [consulté le 5 décembre 2024].
26. SPORT DES ADULTES SUISSE, 2022. Documentation technique concernant l'interface standardisée (API). . p. 10.
27. SportEasy - L'app n°1 pour gérer une équipe ou un club de sport amateur, 2024 *SportEasy* [en ligne]. Disponible à l'adresse : <https://www.sporteasy.net/fr/> [consulté le 5 décembre 2024].

28. STAVER, Anastasia, 2021. Pros and Cons of Using Spring Boot. *Bamboo Agile / Custom Software Development Company* [en ligne]. 4 juillet 2021. Disponible à l'adresse : <https://bambooagile.eu/insights/pros-and-cons-of-using-spring-boot> [consulté le 5 décembre 2024].
29. Système de gestion de base de données relationnel-objet, 2024 *Wikipédia* [en ligne]. Disponible à l'adresse : https://fr.wikipedia.org/w/index.php?title=Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es_relationnel-objet&oldid=217863341 [consulté le 21 mars 2025]. Page Version ID: 217863341
30. TANAY KUMAR, Deo, 2023. Angular vs React vs Vue: Key Differences. *LambdaTest* [en ligne]. 4 octobre 2023. Disponible à l'adresse : <https://www.lambdatest.com/blog/angular-vs-react-vs-vue/> [consulté le 5 décembre 2024].
31. TEAMLINKT, 2024. TeamLinkt | All-in-one sports management platform. *TeamLinkt* [en ligne]. 2024. Disponible à l'adresse : <https://teamlinkt.com/> [consulté le 5 décembre 2024].
32. TEAMSNAPE, 2024. TeamSnap - Youth Sports Team, Club & League Management App. *TeamSnap* [en ligne]. 2024. Disponible à l'adresse : <https://www.teamsnap.com/> [consulté le 5 décembre 2024].
33. The Good and the Bad of C# Programming, *AltexSoft* [en ligne]. Disponible à l'adresse : <https://www.altexsoft.com/blog/c-sharp-pros-and-cons/> [consulté le 5 décembre 2024].
34. THIBAUT (CTO), 2023. Vue JS, Angular ou react : quelle technologie front-end choisir ? *Access it* [en ligne]. 13 avril 2023. Disponible à l'adresse : <https://www.access-it.fr/actualite/vuejs-angular-ou-react/> [consulté le 5 décembre 2024].

35. TINTO, Thomas, 2021. Avantages et inconvénients du Golang (Go) | Software Developer India. *Avantages et inconvénients du Golang (Go)* [en ligne]. 30 juin 2021. Disponible à l'adresse : <https://www.software-developer-india.com/fr/avantages-et-inconvenients-du-golang-go/> [consulté le 5 décembre 2024].
36. WER AWLED, Belhedi, 2023. (3) Comparaison entre l'utilisation d'Angular avec Spring Boot et NestJS : Avantages et inconvénients | LinkedIn. *Comparaison entre l'utilisation d'Angular avec Spring Boot et NestJS : Avantages et inconvénients* [en ligne]. 9 juillet 2023. Disponible à l'adresse : <https://www.linkedin.com/pulse/comparaison-entre-lutilisation-dangular-avec-spring-et-awled-belhedi/> [consulté le 5 décembre 2024].
37. ЭЛЛЕОНОРА КЕРРИ, 2023. Coffee break #75. Advantages and disadvantages of using Spring Boot. Functions for Strings in Java. *JavaRush* [en ligne]. 8 août 2023. Disponible à l'adresse : <https://javarush.com/en/groups/posts/en.3380.coffee-break-75-advantages-and-disadvantages-of-using-spring-boot-functions-for-strings-in-> [consulté le 5 décembre 2024].
38. Illustration, *Pinterest* [en ligne]. Disponible à l'adresse : <https://it.pinterest.com/pin/8444318046209688/> [consulté le 28 mars 2025].