



# **H.C.C**

## **Handball Club de Comines**

| <b>Version du document</b> | <b>Rédigé par</b> | <b>Fonction</b>                  | <b>Approuvé par</b> | <b>Date de modification</b> |
|----------------------------|-------------------|----------------------------------|---------------------|-----------------------------|
| <b>V.1</b>                 | <b>Luc KARNE</b>  | Responsable informatique du club |                     | 22/02/2024                  |

# 1 - CONTEXTE

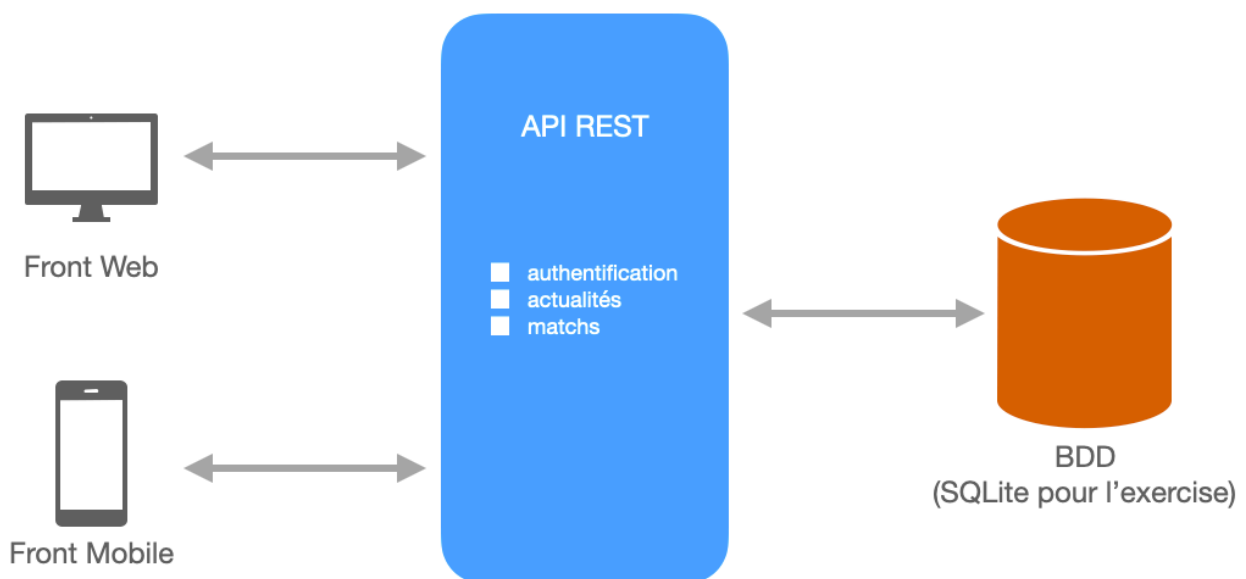
Le H.C.C, club de handball de la ville de Comines, était un petit club local avec seulement quelques adhérents. Les informations étaient échangées de manière informelle, par le biais d'appels téléphoniques et de messages textes. Mais depuis, le club a connu un développement rapide, avec l'arrivée de nouveaux joueurs et l'expansion de son champ d'action.

Avec cette croissance, le club a besoin d'un système plus organisé pour gérer ses adhérents et ses événements, tels que les matchs hebdomadaires. Il souhaite donc mettre en place un site internet et une application mobile pour promouvoir ses activités, avec un fil d'actualités et un espace membre pour les adhérents.

## 2 - BESOINS

Le H.C.C a pris la décision de mettre en place une API REST afin de fournir des informations en temps réel sur les matchs et les événements, ainsi que des mises à jour régulières sur les actualités du club. Public et adhérents, pourront ainsi s'informer sur le club à travers son futur site internet et son application mobile.

## 3 - EXIGENCES TECHNIQUES



L'API sera développée en utilisant NestJS et une base de données SQLite en local pour faciliter les tests et les démos. Les interactions avec la base de données se feront à l'aide de TypeORM.

L'API doit prendre en compte 3 aspects fonctionnels : gestion de l'**authentification**, gestion des **actualités**, gestion des **matchs**.

Concernant l'authentification, l'API doit être en mesure de fournir des tokens **jwt** aux utilisateurs. C'est grâce à ces tokens que l'accès aux différentes parties des ressources est possible (notion de scope et de rôles).

## 4 - SPÉCIFICATIONS FONCTIONNELLES ET RÈGLES DE GESTION

### 4.1 Authentification

- Un adhérent peut avoir soit le rôle « coach », « contributeur », « joueur »
- Un utilisateur peut se créer un compte adhérent, son compte sera validé par le club, son compte aura soit le rôle « coach », soit « contributeur », soit « joueur »
- Les adhérents du club peuvent s'authentifier avec un email et un mot de passe. Il reçoit un token JWT pour accéder aux fonctionnalités réservées aux adhérents

### 4.2 Gestion des adhérents

- Seuls les adhérents ayant le rôle « joueur » peuvent s'inscrire aux matchs
- Les adhérents peuvent s'inscrire à plusieurs matchs
- Les adhérents peuvent se désinscrire des matchs auxquels il se sont inscrits.
- L'API doit être capable de retourner les infos de l'ensemble des adhérents (nom, prénom, date d'inscription, matchs auxquels ils participent)
- L'API doit être capable de retourner les infos d'un adhérent particulier (nom, prénom, date d'inscription, matchs auxquels il participe)

### 4.3 Gestion des actualités

- Seul les adhérents ayant le rôle « contributeur » peuvent publier des actualités.
- L'API doit être capable de retourner l'ensemble des actualités (toutes les infos + auteur).
- L'API doit être capable de retourner une actualité particulière (toutes les infos + auteur).

### 4.4 Gestion des matchs

- Seul un adhérent « coach » peut saisir des matchs
- Seul un adhérent « coach » peut modifier les matchs (ex: scores)
- Un match doit aussi contenir un adversaire et un score final
- Il ne peut pas y avoir un match le même jour
- L'API doit être capable de retourner les infos de l'ensemble des matchs + liste des participants
- L'API doit être capable de retourner les infos d'un match particulier + liste des participants