

Toulouse Ynov Campus

Florent Detres

API de Gestion des salles de sports et utilisateurs

Une solution backend robuste pour la gestion des gyms, abonnements et statistiques

Véritable écosystème

Introduction

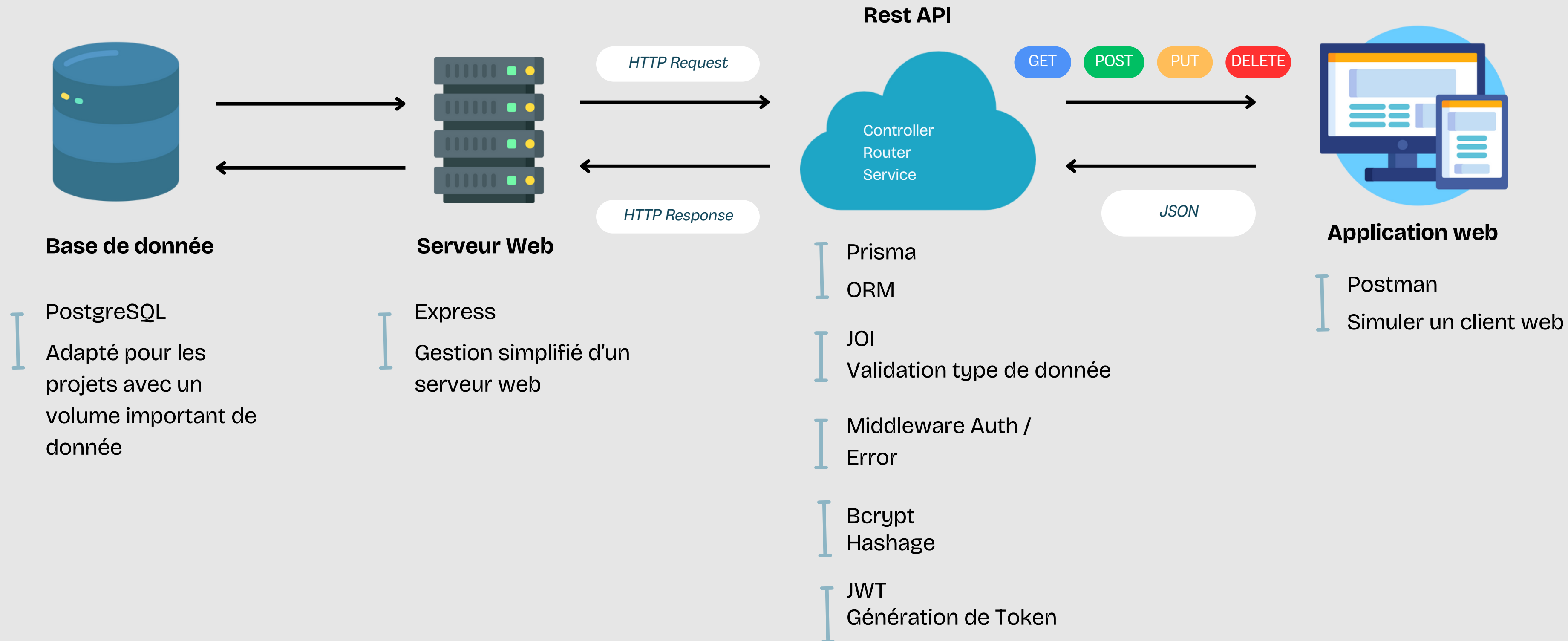
Pour les entreprises



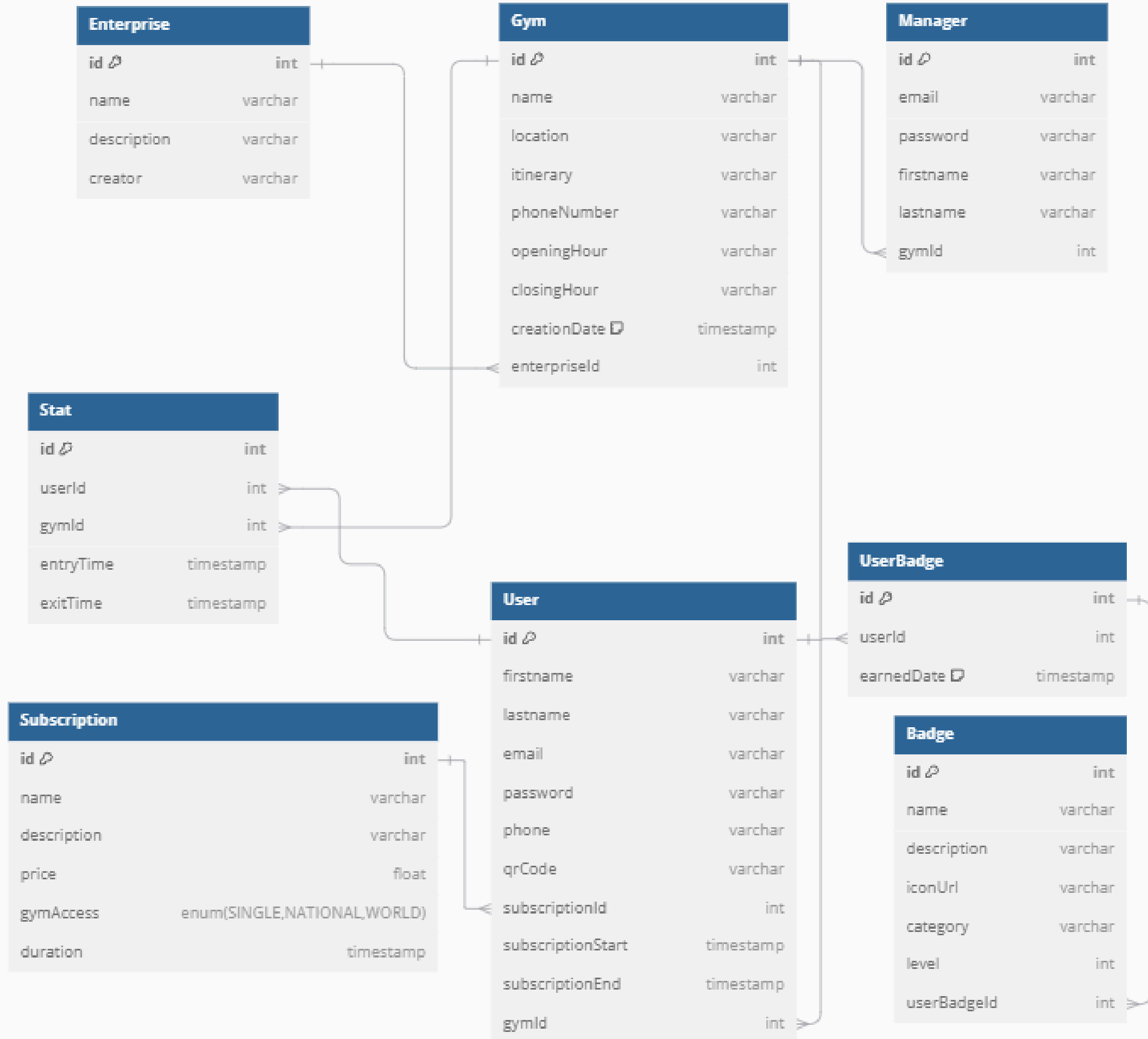
Pour les utilisateurs



Architecture de l'API



MLD



Fonctionnalités principales

```
model Enterprise {
  id          Int    @id @default(autoincrement())
  name        String
  description  String
  creator      String

  gym Gym[]
}

You, last week | 1 author (You)
model Gym {
  id          Int    @id @default(autoincrement())
  name        String
  location     String
  itinerary    String?
  phoneNumber  String?
  openingHour  String?
  closingHour  String?
  creationDate DateTime @default(now())
  enterpriseId Int

  manager  Manager[]
  user     User[]
  stats    Stat[]
  enterprise Enterprise @relation(fields: [enterpriseId], references: [id])
}
```

Gestion des Gyms et Entreprises

Gestion des entreprises:

- Liste des gyms associés
- Calculs automatisés pour suivre les performances des entreprises (ex. revenus générés, nombre de gyms).
- Statistiques globales et par entité (utilisateurs, abonnements, gyms).

Création et gestion des gyms :

- Nom, localisation, horaires d'ouverture
- Association avec des entreprises
- Activation/désactivation des entreprises et utilisateurs pour gérer l'accès.
- Filtrage par distance
- Triage par nom d'entreprise

Fonctionnalités principales

```
model Manager {  
  id      Int    @id @default(autoincrement())  
  email    String  
  password String  
  firstname String  
  lastname String  
  
  gymId    Int?  
  gym Gym? @relation(fields: [gymId], references: [id])  
}
```

Gestion des managers

- Création et gestion des managers associés à des gyms spécifiques
- Accès sécurisé pour les managers pour administrer leurs gyms
- Association d'un manager à un ou plusieurs gyms



Fonctionnalités principales

```
model User {  
  id          Int          @id @default(autoincrement())  
  firstname   String  
  lastname    String  
  email       String       @unique  
  password    String  
  phone       String  
  qrCode      String?  
  
  subscriptionId Int?  
  subscription   Subscription? @relation(fields: [subscriptionId], references: [id])  
  subscriptionStart DateTime?  
  subscriptionEnd DateTime?  
  
  stats      Stat[]  
  userBadge  UserBadge[]  
  
  gymId      Int?  
  gym        Gym? @relation(fields: [gymId], references: [id])  
}
```

Gestion des utilisateurs

- Inscription, connexion sécurisée avec JWT
- Mise à jour des profils et mots de passe
- Attribution de badges et suivi via QR codes



Fonctionnalités principales

```
model Subscription {
  id          Int          @id @default(autoincrement())
  name        String
  description String
  price       Float
  gymAccess   GymAccess
  duration    DateTime
  users       User[]
}
```


```
enum GymAccess {
  SINGLE
  NATIONAL
  WORLD
}
```

Abonnements

Gestion des abonnements : différents plans selon l'accès (gym unique ou multi-gym).

Association des abonnements aux utilisateurs.

Pricing Table Layout




FREE

Ut enim ad minim veniam, quis nostrud exercitation

\$ **0** / month

- ☒ Laboris nisi ut aliquip exi
- ☐ Dolor sit amet
- ☐ Ut enim ad minim
- ☐ Aliquip ex ea
- ☐ Commodo consequat

ORDER NOW




BASIC

Ut enim ad minim veniam, quis nostrud exercitation

\$ **11** / month

- ☒ Nisi ut aliquip exi
- ☒ Dolor sit amet
- ☐ Enim ad minim
- ☐ Ullamco laboris nisi ut
- ☐ Do eiusmod tempor

ORDER NOW




STANDARD

Laboris nisi ut aliquip ex ea commodo consequat.

\$ **22** / month

- ☒ Ullamco laboris nisi ut
- ☒ Dolor sit amet
- ☒ Ut enim ad minim
- ☐ Laboris nisi ut aliquip exi
- ☐ Aliquip ex ea

ORDER NOW




PREMIUM

Sed do eiusmod tempor incididunt ut labore

\$ **55** / month

- ☒ Do eiusmod tempor
- ☒ Aliquip ex ea
- ☒ Ut enim ad minim
- ☒ Ullamco laboris nisi ut
- ☐ Commodo consequat

ORDER NOW



BUSINESS

Lorem ipsum dolor sit amet, consectetur.

\$ **199** / month

- ☒ Commodo consequat
- ☒ Ut enim ad minim
- ☒ Dolor sit amet
- ☒ Ullamco laboris nisi ut
- ☒ Nisi ut aliquip exi

ORDER NOW

Ut enim ad minim veniam, quis nostrud exercitation

Fonctionnalités principales

```
model Badge {
  id          Int          @id @default(autoincrement())
  name        String
  description String
  iconUrl     String
  category    String
  level       Int

  userBadgeId Int
  userBadge UserBadge @relation(fields: [userBadgeId], references: [id])
}
```

You, last week | 1 author (You)

```
model UserBadge {
  id          Int          @id @default(autoincrement())
  userId      Int
  earnedDate  DateTime @default(now())
  user        User         @relation(fields: [userId], references: [id])
  badge       Badge[]
}
```

Badge

- Création, modification, et suppression des badges
- Récompense des utilisateurs avec des badges pour augmenter leur engagement
- Catégorisation des badges (ex. niveau 1, niveau 2, niveau 3) et suivi des niveaux
- Suivi des badges associés aux utilisateurs pour encourager leur fidélité

Fonctionnalités principales

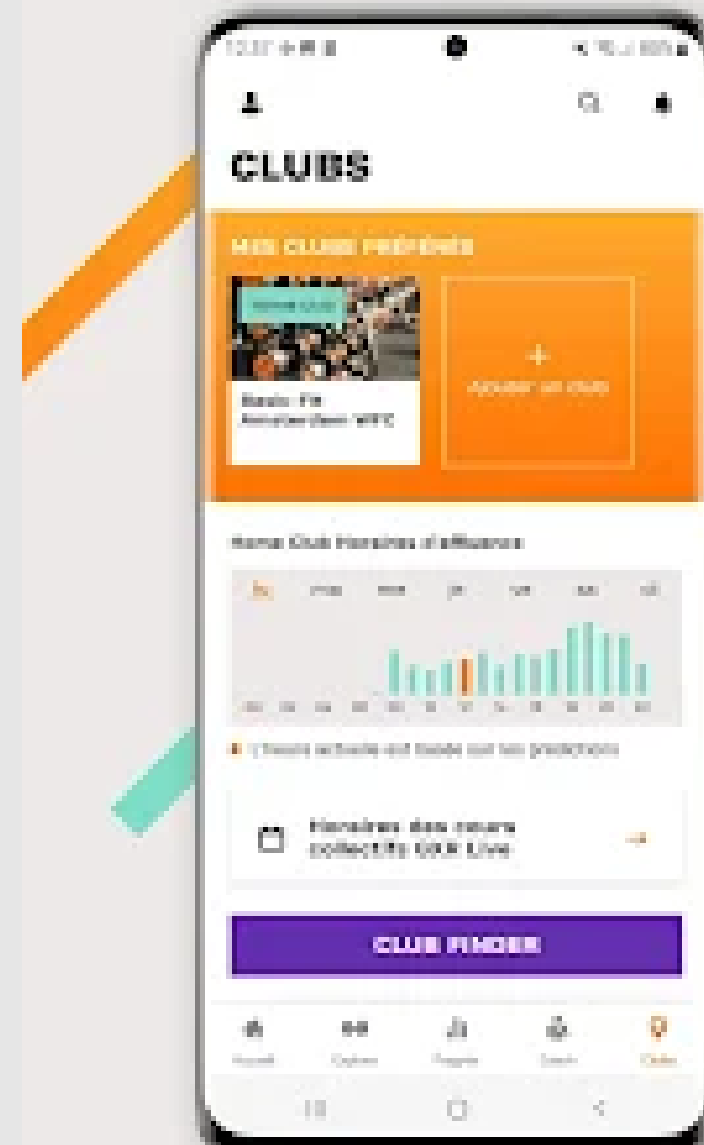
Suivi et Statistiques

Suivi des entrées/sorties des utilisateurs

Génération de statistiques :

- Analyse des abonnements actifs
- Intégration de QR codes pour le suivi des entrées en gym.
- Génération de statistiques sur la fréquentation des gyms.

```
model Stat {  
  id          Int          @id @default(autoincrement())  
  userId      Int  
  gymId       Int  
  entryTime   DateTime  
  exitTime    DateTime?  
  
  user User @relation(fields: [userId], references: [id])  
  gym  Gym @relation(fields: [gymId], references: [id])  
  
  @@index([userId])  
  @@index([gymId])  
}
```



Developpement API

Merci pour votre attention !

Avez-vous des questions ?