

Projet SAE S6

Suivi d'entraînements sportifs et coaching

Le but de ce projet est de réaliser une application web de gestion d'un club de sport orienté Coaching. Ce club bénéficiera d'un logiciel destiné au personnel et d'une application dédiée au sportif. Ce système sera développé avec les outils logiciels utilisés lors du POC.

Vision globale du travail à réaliser

✂ Un Back-office (avec Symfony et EasyAdmin)

Le back-office permettra aux **responsables et coachs** de gérer les utilisateurs, les séances d'entraînement et les exercices via une interface d'administration basée sur **EasyAdmin**. Il inclura une **IHM personnalisée** pour la planification des séances et le suivi des fréquentations, ainsi qu'un tableau de bord permettant aux responsables d'analyser les statistiques et de générer les fiches de paie des coachs.

✂ Une API REST (avec Symfony)

L'API REST servira d'interface entre le back-office et le front-office, exposant des endpoints sécurisés pour la **gestion des utilisateurs, des séances et des réservations**. Un **webservice complexe** sera développé pour fournir des recommandations de séances aux sportifs en fonction de leur historique et de leurs performances, nécessitant un traitement spécifique des données.

✂ Un Front-Office (avec Angular)

L'application Angular offrira aux **internauts et sportifs** une interface fluide et ergonomique pour consulter les coachs, s'inscrire, gérer leurs réservations et suivre leur progression. Les sportifs pourront **accéder à leur historique d'entraînement, voir leurs statistiques personnelles et recevoir des recommandations de séances adaptées** pour optimiser leur progression.

Cas d'utilisation : Définition des acteurs et de leurs fonctionnalités

✂ Acteurs du système

1. Utilisateur anonyme (de l'application Angular)

- Peut consulter les informations publiques (présentation des coachs, des types de séances, planning des créneaux ouverts).
- Peut s'inscrire pour devenir "sportif".

2. Sportif (utilisateur connecté sur l'application Angular)

- Peut consulter ses séances planifiées.
- Peut voir les exercices de ses séances passées et à venir.
- Peut consulter l'historique de ses entraînements validés.
- Peut s'inscrire à une séance libre si un coach a un créneau disponible.
- Peut annuler une séance réservée sous certaines conditions (délai, nombre d'annulations max, etc.).

3. Coach (utilisateur du back-office Symfony avec accès restreint)

- Peut consulter ses séances à venir.
- Peut **créer/modifier des séances** en respectant les règles du système.
- Peut associer 1 à 3 sportifs à une séance.
- Peut enregistrer et valider la réalisation d'une séance.
- Peut gérer la liste des exercices associés aux séances.

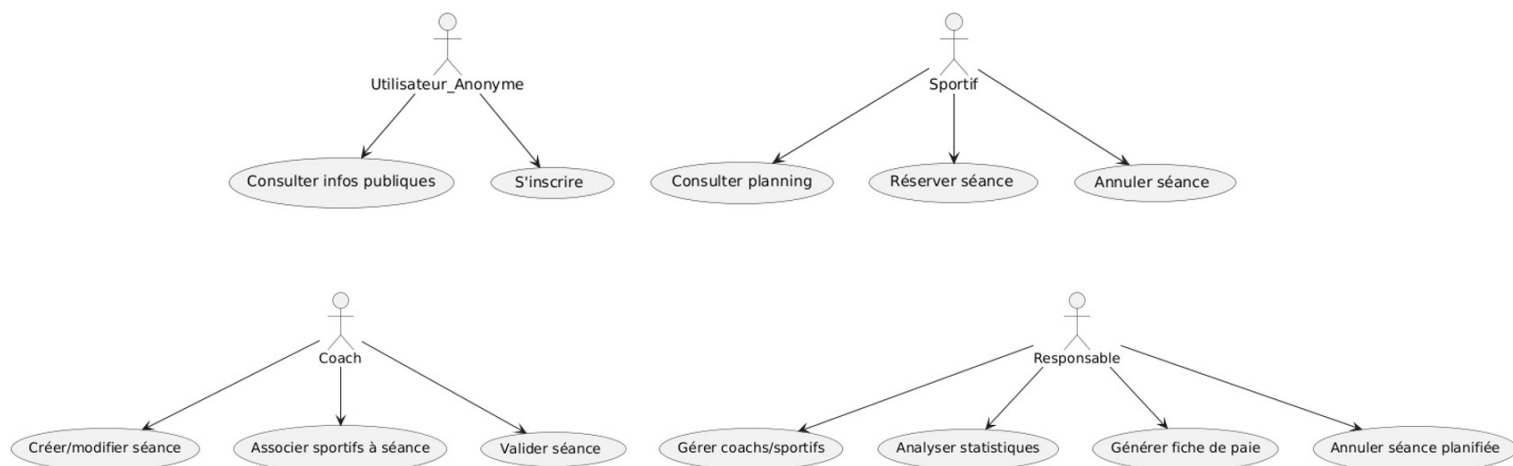
4. Responsable de salle (utilisateur du back-office Symfony avec accès complet)

- Peut tout faire qu'un coach peut faire.
- Peut gérer la liste des coaches et des sportifs.
- Peut **analyser les statistiques** (fréquentation, taux de remplissage des séances, popularité des thèmes d'entraînement).
- Peut **gérer les paiements des coaches** (avec accès aux séances validées pour générer la paie).
- Peut **annuler une séance planifiée**.

📌 Cas d'utilisation principaux

Voici les **cas d'utilisation** les plus importants :

- **Un internaute consulte les informations publiques sur l'application.**
- **Un internaute s'inscrit en tant que sportif.**
- **Un sportif consulte son planning d'entraînement et les exercices associés.**
- **Un sportif réserve une séance disponible.**
- **Un coach planifie une séance d'entraînement avec des sportifs.**
- **Un coach valide une séance terminée et enregistre la présence des sportifs.**
- **Un responsable visualise les statistiques de fréquentation des coaches et des séances.**
- **Un responsable génère la fiche de paiement des coaches en fonction des séances réalisées.**



Définition des entités et tables principales

Vous vous **inspirerez** de ce diagramme de classes pour créer vos entités. Vous choisirez les détails d'implémentation (type d'attribut, valeur par défaut, etc.) selon ce qui vous paraît le plus cohérent pour le projet. Vous prévoyez également des **fixtures** pour « peupler » suffisamment votre base de données.

1. Utilisateur (hérité par Sportif et Coach)

- id (UUID)
- nom
- prénom
- email
- mot_de_passe
- rôle (coach, sportif, responsable)

2. Sportif (hérite d'Utilisateur)

- date_inscription
- niveau_sportif (débutant, intermédiaire, avancé)

3. Coach (hérite d'Utilisateur)

- spécialités (liste des thèmes d'entraînement proposés. Cf. Séances)
- tarif_horaire

4. Séance

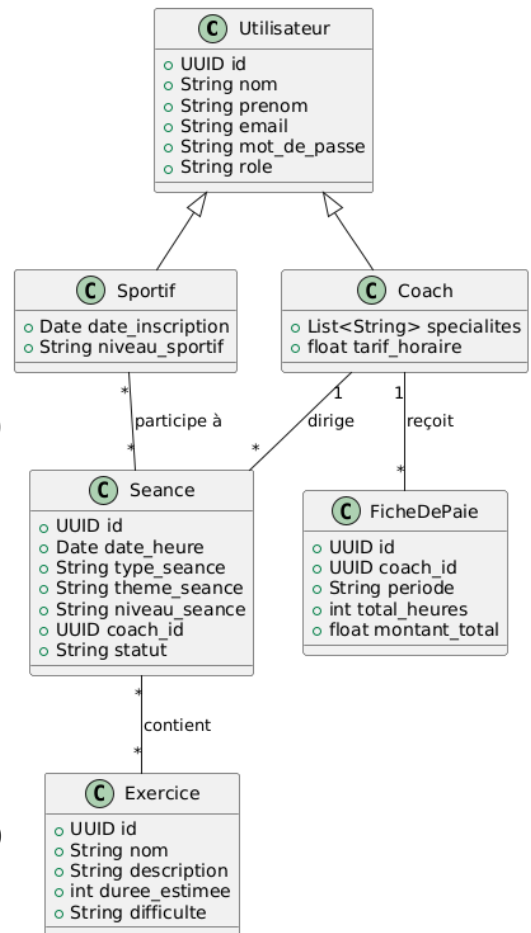
- id
- date_heure
- type_seance (solo, duo, trio)
- theme_seance (fitness, cardio, muscu, crossfit, ...)
- coach_id
- sportifs (relation ManyToMany avec Sportif, max 3)
- exercices (relation ManyToMany avec Exercice)
- statut (prévue, validée, annulée)
- niveau_sceance (débutant, intermédiaire, avancé)

5. Exercice

- id
- nom
- description
- duree_estimee
- difficulte (facile, moyen, difficile)

6. Fiche de paie

- id
- coach_id
- période (mois ou semaine)
- total_heures
- montant_total



Répartition des fonctionnalités par composant logiciel

- **Back-office Symfony + EasyAdmin**
 - **Gestion des coachs, sportifs et séances** (CRUD via EasyAdmin).
 - **Gestion des exercices** (CRUD via EasyAdmin).
 - **Planification des séances par les coachs** (formulaire personnalisé avec contraintes métier).
 - **Validation des séances réalisées** (interface spécifique pour les coachs).
 - **Statistiques sur la fréquentation et les réservations** (tableau de bord pour le responsable).
 - **Génération des fiches de paie** (module dédié pour le responsable).
 - **API REST (Symfony)**
 - **CRUD standard pour alimenter Angular** (séances, coachs, exercices, sportifs).
 - **Webservice complexe : Gestion avancée des séances**
 - Recherche des créneaux disponibles
 - Association des sportifs à une séance en fonction des règles métier
 - Calcul des statistiques de fréquentation
 - **Application Angular (Front-end pour sportifs et internautes)**
 - **Affichage des coachs et des types de séances** (partie publique).
 - **Consultation et réservation des séances disponibles.**
 - **Affichage du planning personnel et des séances passées/validées.**
 - **Affichage des détails des exercices liés aux séances.**
 - **Authentification multi-niveau**
 - **Connexion des utilisateurs sur l'application (JWT ou autre méthode sécurisée).**
 - **Gestion des rôles et permissions sur Symfony (Coach, Sportif, Responsable).**
-

Règles métier et validations spécifiques

Gestion des séances (création, modification, annulation)

- Un coach **ne peut pas programmer** deux séances en même temps.
- Un sportif **ne peut pas être inscrit** à deux séances à la même heure.
- Une séance **ne peut être annulée** que :
 - Par un sportif **jusqu'à 24h avant** (sinon elle est comptabilisée comme une absence).
 - Par un coach **seulement en cas de force majeure** (avec validation du responsable).
- Une séance validée **ne peut plus être modifiée ou annulée**.

Gestion des réservations des sportifs

- Un sportif peut réserver **un maximum de 3 séances à l'avance**.
- Il **ne peut pas réserver** une séance qu'il a déjà annulée deux fois.
- Il peut réserver une séance **uniquement avec une séance correspondante à son niveau** (ex : débutant ne peut pas réserver une séance avancée).

Statistiques et suivi des performances

- Le responsable peut voir :
 - **Le taux d'occupation des séances** par coach et par créneau horaire.
 - **Le taux d'absentéisme des sportifs**.
 - **Le classement des séances les plus populaires**.
 - Les sportifs peuvent voir :
 - Leur **progression personnelle** (nombre de séances suivies, types d'exercices réalisés).
 - Leur **niveau sportif évolutif** (débutant -> intermédiaire -> avancé).
-

Webservice complexe à développer spécifiquement

Bilan d'entraînement personnalisé

Objectif

Permettre à un sportif de voir **un récapitulatif de ses séances et exercices favoris** sur une période sélectionnée (hebdo, mensuel, trimestriel, annuel ou personnalisée).

Entrées

- `id_sportif` (identifiant du sportif)
- `date_min` (début de la période)
- `date_max` (fin de la période)

Sorties attendues

- **Nombre total de séances réalisées** sur la période.
- **Répartition par type de séance** (solo, duo, trio).
- **Top 3 des exercices les plus pratiqués.**
- **Durée totale passée en entraînement.**

Intégration dans l'application

- Onglet "Mon Bilan" dans l'espace personnel du sportif.
- Sélecteurs de période prédéfinis :
 - Boutons rapides : "Semaine", "Mois", "Trimestre", "Année".
 - Mode avancé : Sélecteurs de date de début et date de fin pour une période personnalisée.
- Affichage des résultats sous forme de :
 - Chiffres clés (ex : "12 séances", "5h30 d'entraînement", "+15% vs période précédente").
 - Graphique circulaire pour la répartition des types de séances.
 - Liste des exercices les plus pratiqués avec leur fréquence.

Travail à effectuer

En suivant la logique du POC (**Back Office avec Symfony et EasyAdmin, Front Office Angular qui interagit avec une API de Symfony**) et avec les connaissances acquises en Symfony et Angular au cours de la scolarité, développer une **application web** qui respecte le cahier des charges donné ci-dessus.

Une fois les fonctionnalités minimales en place, vous pouvez **enrichir votre application** de fonctionnalités complémentaires (qui vous donneront l'occasion d'enrichir vos compétences et ne manqueront pas d'impressionner le jury lors de l'évaluation finale).

Vous êtes libres d'utiliser des **outils complémentaires (GIT, ...)** pour votre projet mais vous devrez être autonomes sur ces outils.

Évaluation

Votre application sera évaluée en **20 min maximum** par votre enseignant, selon le **cas de démonstration** que vous aurez préparé, lors du dernier créneau de cours de la SAE en fin de **semaine 10**. La note de cette **recette sera collective** au groupe.

Une note **individuelle** sera de plus établie selon le **QCM "papier"** que vous passerez :

- **Le jeudi 06/03 à 12h00 en salle B007/B008**

(il portera sur tous les points de l'application... restez donc bien informé du travail de toute l'équipe).

Mission : Possible

« Votre mission, à supposer que vous l'acceptiez, consiste à développer cette application. Comme toujours, si vous ou l'un de vos agents étaient capturés ou tués, le département Informatique nierait avoir eu connaissance de vos agissements.

Ce fichier s'autodétruit dans cinq secondes. Bonne chance Geek. »
