

Projet Corpo Padel

Quentin Horgues (*qhorgues*)
Anthony Gimenez (*Githonil*)
Lucas Jusserand
Aelig Hamadou (*aeliggg*)
Mohamed Hachelef (*Aetis404*)
Alexis Guillotin (*leBourreau*)

M. Andrey Pivodori

1. Introduction.....	3
2. Architecture.....	3
a. Schéma d'architecture.....	3
b. Choix technologiques.....	3
c. Modèle de données.....	3
3. Fonctionnalités implémentées.....	4
a. Gestion des équipes.....	4
b. Planning des événements.....	4
c. Liste des matchs à venir.....	5
d. Résultats et classements.....	6
e. Administration.....	6
f. Profil utilisateur.....	6
4. Sécurité.....	6
a. Mesures implémentées.....	6
b. Tests de sécurité.....	7
5. Tests.....	7
a. Stratégie de test.....	7
b. Couverture.....	7
c. Résultats.....	7
6. Difficultés rencontrées.....	7
7. Améliorations possibles.....	8
8. Conclusion.....	8

1. Introduction

Ce projet avait pour objectif de concevoir et développer une application web de gestion de tournois corporatifs de padel, tout en mettant l'accent sur la qualité logicielle, la sécurité et les tests. L'application permet de gérer les équipes, les matchs, les événements ainsi que les résultats, à travers une interface moderne et intuitive adaptée aux différents rôles utilisateurs.

Le développement s'est appuyé sur une architecture backend et frontend distincte, construite à partir d'une base fournie par l'enseignant. En complément de l'implémentation fonctionnelle, le projet incluait la rédaction de deux documents techniques essentiels : un cahier des spécifications et un cahier de tests, visant à formaliser les besoins et à garantir la fiabilité de l'application.

2. Architecture

a. Schéma d'architecture

L'architecture de l'application repose sur une séparation claire entre le frontend, le backend et la base de données.

Le frontend, développé avec SvelteKit, communique avec le backend via une API REST. Le backend, implémenté en Python avec FastAPI, gère la logique métier, la sécurité et l'accès aux données.

Les données sont stockées dans une base de données relationnelle accessible via SQLAlchemy.

Cette architecture permet une bonne maintenabilité, une évolutivité facilitée et une séparation claire des responsabilités.

b. Choix technologiques

Pour ce qui est des technologies utilisées dans le projet, nous avons Python pour le backend où nous utilisons les librairies Pydantic, FastAPI, SQLAlchemy et Pytest.

Pour le front, nous utilisons SvelteKit, qui nous permet de construire une interface moderne et réactive. Nous avons fait le choix de TypeScript pour bénéficier du typage statique et réduire les erreurs lors du développement. Pour le styling, nous nous sommes tournés vers TailwindCSS.

c. Modèle de données

Le modèle de données est conçu autour des principales entités métier de l'application : utilisateurs, entreprises, joueurs, équipes, poules, matchs et événements.

Les relations entre ces entités permettent de garantir la cohérence des données, notamment les contraintes assurant qu'un joueur n'appartient qu'à une seule équipe par saison et qu'une équipe ne participe qu'à un match par événement.

L'utilisation d'un modèle relationnel facilite la validation des règles métier et l'exploitation des données pour les classements et statistiques.

3. Fonctionnalités implémentées

L'interface utilisateur a été développée pour offrir une expérience fluide et intuitive aux utilisateurs. Voici les principales fonctionnalités mises en place :

a. Gestion des équipes

La page dédiée aux équipes affiche l'ensemble des équipes inscrites dans le tournoi. Un système de recherche permet de filtrer rapidement les équipes par nom d'entreprise. Chaque équipe est composée de deux joueurs issus de la même entreprise et peut être affectée à une poule.

Les administrateurs disposent d'un bouton "Ajouter une équipe" qui ouvre un formulaire. Ce formulaire permet de sélectionner l'entreprise puis deux joueurs appartenant à cette même entreprise pour constituer une nouvelle équipe. Le système vérifie automatiquement qu'un joueur ne peut être que dans une seule équipe par saison. Les équipes peuvent également être modifiées (changement des joueurs uniquement si aucun match n'a été joué) ou supprimées (possible uniquement si aucun match n'a été joué). Le formulaire inclut une validation côté client pour s'assurer que tous les champs sont correctement remplis et que les règles métier sont respectées.

b. Planning des événements

Le calendrier constitue le cœur de l'application. Il présente une vue mensuelle interactive où chaque jour peut contenir un ou plusieurs événements. Les jours contenant des événements sont visuellement distingués avec une couleur de fond bleue, un point indicateur, et un compteur affichant le nombre d'événements prévus ce jour-là.

Un système de filtrage permet aux utilisateurs de basculer entre l'affichage de tous les événements et uniquement leurs propres matchs via une case à cocher "Voir uniquement mes événements". Cette fonctionnalité facilite la navigation pour les joueurs qui souhaitent consulter rapidement leur planning personnel. Par défaut, les joueurs voient uniquement les événements où ils sont impliqués, avec la possibilité de décocher le filtre pour voir l'ensemble du calendrier.

En cliquant sur un jour contenant des événements, une page s'ouvre affichant le détail de tous les événements de cette journée. Pour chaque événement, on peut voir l'horaire et la liste complète des matchs programmés. Pour chaque match, les informations suivantes sont

affichées : le numéro de terrain (de 1 à 10), les deux équipes qui s'affrontent identifiées par leur entreprise, le statut du match avec un badge coloré (À venir en bleu, Terminé en vert, Annulé en rouge), et si le match est terminé on peut observer le score.

Les administrateurs ont la possibilité de créer de nouveaux événements via un bouton dédié. Le formulaire de création permet de définir la date, l'heure au format et le nombre de matchs simultanés. Pour chaque match, l'administrateur sélectionne le numéro de terrain (de 1 à 10) ainsi que les deux équipes participantes parmi la liste des équipes déjà existantes.

Le système intègre plusieurs validations strictes pour garantir la cohérence des données comme l'impossibilité de créer un événement dans le passé, vérification du format de l'heure (HH:MM entre 00:00 et 23:59), unicité des terrains (deux matchs d'un même événement ne peuvent pas se dérouler sur le même terrain), et unicité des équipes (une équipe ne peut jouer qu'un seul match par événement).

La modification d'événements existants est également possible pour les administrateurs via un bouton "Modifier" qui pré-remplit le formulaire avec les données actuelles. La suppression n'est autorisée que si tous les matchs de l'événement ont encore le statut "A_VENIR", évitant ainsi de perdre l'historique des matchs déjà joués. Une confirmation est demandée avant toute suppression.

c. Liste des matchs à venir

La page "Matches" affiche la liste des matchs programmés dans les 30 prochains jours sous forme de liste chronologique. Chaque carte de match présente la date et l'heure, le numéro de piste et les noms des entreprises qui s'affrontent, le prénom et nom des quatre joueurs concernés sous chaque équipe, un statut et le nom de la poule.

Le système de filtrage s'adapte automatiquement au rôle de l'utilisateur. Pour les joueurs, la liste affiche par défaut uniquement les matchs de leurs équipes, avec un "Voir tous les matchs" permettant de désactiver ce filtre et d'accéder à l'ensemble du calendrier. Les administrateurs bénéficient de filtres avancés dans une barre dédiée : un filtre par entreprise, un filtre par poule, et un filtre par statut. Ces filtres peuvent être combinés.

Les administrateurs disposent de boutons d'action sur chaque match : "Modifier" et "Supprimer". L'ajout d'un nouveau match se fait via "Ajouter un match". Le formulaire d'ajout contient les champs suivants : date, heure, piste, équipe 1 et équipe 2 (sélection parmi toutes les équipes existantes), et statut avec "A_VENIR" par défaut.

En mode modification, la date et l'heure deviennent non modifiables (champs grisés avec texte explicatif "La date/heure ne peut pas être modifiée"). Seuls la piste, les équipes et le statut peuvent être changés. Le champ statut propose trois options : À venir, Terminé (Un match ne peut être marqué comme terminé que si sa date est passée), et Annulé.

Le système vérifie que la date est aujourd'hui ou dans le futur, que les deux équipes sont différentes, et qu'aucun autre match n'utilise la même piste au même créneau. La suppression n'est autorisée que pour les matchs "À venir" avec confirmation systématique.

Lorsqu'un administrateur crée un nouveau match, le système crée automatiquement un événement contenant ce match, assurant la cohérence des données.

d. Résultats et classements

La page "Résultats" permet aux joueurs de consulter leurs résultats personnels et le classement général des entreprises. Pour les utilisateurs non connectés, un message invite à se connecter.

Les résultats personnels sont affichés sous forme de cartes chronologiques. Chaque match terminé présente la date, les adversaires avec leur entreprise, le score, et le résultat (Victoire en vert ou Défaite en rouge).

Le classement général est présenté dans une colonne à droite. Les informations incluent les matchs joués, les victoires et défaites, le total de points, et les sets gagnés/perdus. Le système applique 3 points par victoire et 0 pour une défaite. L'interface présente les deux sections côte à côte sur grand écran.

e. Administration

La page d'administration est réservée aux administrateurs avec redirection automatique pour les autres rôles. Elle comporte la gestion des joueurs et la gestion des équipes & poules.

La gestion des joueurs présente un tableau avec validation en temps réel lors de l'ajout ou modification. Les règles incluent : nom et prénom de 2-50 caractères, entreprise sélectionnée, numéro de licence au format LXXXXXX, et email valide et unique. Un système permet d'ajouter dynamiquement de nouvelles entreprises. La suppression n'est possible que si le joueur n'est dans aucune équipe.

La section équipes et poules accessible via "Équipes & Poules" comporte deux onglets. L'onglet équipes affiche toutes les équipes avec leur entreprise, poule et joueurs. Le système vérifie que les joueurs sont différents et qu'un joueur n'est que dans une seule équipe. L'onglet poules permet de créer des poules avec nom et description. La suppression d'une poule n'est possible que si aucune équipe ne l'utilise. L'interface utilise un code couleur cohérent (bleu/violet/jaune/rouge).

f. Profil utilisateur

La page de profil affiche les informations personnelles : photo, nom, prénom, date de naissance, email, entreprise, numéro de licence et rôle. Un bouton "Éditer" permet d'accéder à la modification.

La page d'édition propose un formulaire pour modifier l'URL de la photo, le prénom, le nom, la date de naissance et l'email. L'entreprise et le numéro de licence sont en lecture seule. Deux boutons "Sauvegarder" et "Annuler" permettent de valider ou abandonner les modifications. Un message de confirmation s'affiche après sauvegarde réussie.

4. Sécurité

a. Mesures implémentées

Pour ce qui est des mesures de sécurité implémentées notamment sur l'authentification d'un utilisateur nous avons mis en place plusieurs règles. Tout d'abord les mots de passes de chaque utilisateur doivent respecter certaines contraintes comme : être compris entre 12 et 60 caractères, contenir au moins 1 majuscule, 1 minuscule, 1 chiffre, 1 caractère spécial et ne pas contenir d'espace. De plus nous avons un système d'essai max (5 essais) après quoi la tentative de connexion sera bloquée pendant 30 minutes.

b. Tests de sécurité

Lors de la mise en place des tests nous avons visé à tester toutes les sécurités mise en place afin de vérifier leur bon fonctionnement, ainsi nous avons un test de création d'utilisateur avec un mauvais mot de passe pour chaque contraintes ainsi qu'un test de connexion raté pour tester la limite d'essai.

5. Tests

a. Stratégie de test

Pour les tests il y a deux type de test qui sont mis en pratique, les tests unitaire avec pytest (pour le back), une petite partie d'automatisation de test fonctionnelle avec cypress et surtout environ 80 test fonctionnelle réparti dans plusieurs catégorie pour les tests fonctionnel qui sont consigné dans le cahier de recette (a ouvrir sous **libre Office calc**) et qui est basé sur le cahier de spécifications

b. Couverture

Pour la partie back-end couvre environ 90 % du code et pour la partie fonctionnelle, il y a environ un taux de couverture du cahier des charges de 60%

c. Résultats

Pytest est entièrement validé par contre au niveau de la partie fonctionnel il manque plusieurs feature secondaire et une feature critique qui rend l'application inutilisable en l'état. En ce qui concerne Cypress avec la majeure partie des tests sont validés, certaine non, dû à certaine raison expliquer dans le cahier de recette.

6. Difficultés rencontrées

Le principal problème rencontré a été un manque de coordination et de communication au sein du groupe.

La répartition des tâches définie au départ s'est révélée inadaptée aux besoins du projet.

En particulier, le cahier des spécifications a été confié uniquement à deux personnes, alors qu'il aurait nécessité une élaboration collective impliquant l'ensemble du groupe. De la même manière, le cahier des recettes n'a été confié qu'à une seule personne, sans véritable suivi collectif.

En conséquence, le cahier des spécifications n'a été finalisé qu'à un stade très avancé du développement, quasiment à son achèvement. Le cahier des recettes, quant à lui, n'a été achevé qu'après la fin du développement. Cela a conduit l'équipe à progresser sans cadre clairement défini pendant une grande partie du projet, compliquant l'organisation du travail et ne correspondant pas à une méthodologie de gestion de projet classique.

Sur le plan technique, nous avons rencontré quelques défis lors du développement du frontend. La gestion de la réactivité de Svelte, notamment lors du filtrage des événements et de la mise à jour du calendrier après l'ajout ou la modification d'événements, a nécessité plusieurs itérations avant de trouver la bonne approche.

7. Améliorations possibles

Plusieurs améliorations pourraient être apportées à l'application.

Il serait possible d'augmenter la couverture des tests fonctionnels afin de couvrir l'ensemble des fonctionnalités prévues dans le cahier des charges.

Des améliorations ergonomiques pourraient également être envisagées sur certaines pages du frontend, notamment sur la gestion du calendrier et des filtres.

Enfin, une meilleure organisation du travail et une rédaction plus précoce des documents de spécifications permettraient d'améliorer l'efficacité globale du projet.

8. Conclusion

Ce projet a permis de développer une application web complète dédiée à la gestion de tournois corporatifs de padel, en s'appuyant sur une architecture moderne et des choix technologiques adaptés. Les fonctionnalités mises en place répondent aux besoins des utilisateurs, qu'il s'agisse de la gestion des équipes, des matchs, des événements ou des résultats, tout en proposant une interface claire et intuitive.

Un travail important a également été réalisé autour de la qualité de l'application, notamment à travers la mise en place de tests unitaires et de tests fonctionnels, formalisés dans un cahier de recettes, afin de garantir la fiabilité, la cohérence des données et la robustesse globale du système.

Malgré certaines difficultés organisationnelles rencontrées durant le projet, cette expérience a été enrichissante sur les plans technique et méthodologique, et a permis de renforcer nos compétences en développement d'applications web dans un contexte proche des contraintes professionnelles.