

# Projet IFT-2935

**Louis-Pierre Bastien**

886038

**Félix Cormier**

20212545

**Emeric Laberge**

20220275

Dans le cadre du cours

IFT 2935



Département d'informatique et de recherche opérationnelle

Université de Montréal

Canada

10 avril 2024

# Introduction

Organiser des événements sportifs amateurs ou professionnels est généralement une tâche ardue. Il en va souvent de même lorsque vient le temps de participer à de tels événements. Que ce soit un tournoi, une partie ou simplement une pratique, il semble nécessaire d'opter pour une approche concertée et unifiée dans l'organisation des événements sportifs, autant pour les participants que les organisateurs.

Les organisateurs devraient pouvoir accéder aux événements qu'il organisent et gérer qui peut participer, la liste des parties ayant lieu pendant ces événements, la liste des employés participant à l'événement ou tout autre considération nécessaire à ce que l'événement ait lieu.

Les utilisateurs voulant faire du sport devraient pouvoir n'avoir qu'un minimum d'obstacles qui les séparent de leur objectif de participer à des événements sportifs, de s'inscrire à des équipes sportives, de gérer facilement les équipes dans lesquelles ils sont inscrits, d'avoir accès à des informations utiles concernant leurs équipes, leurs événements ou leurs parties.

Les employés pour des événements devraient aussi pouvoir suivre les événements pour lesquels ils travaillent à des fins d'organisation.

## Description du projet

La solution que nous avons privilégiée pour arriver à ces fins est un frontend en **React** avec TypeScript. Nous utilisons également MUI, TailwindCSS, Jotai (pour le state management), react-hot-toast et plus encore afin d'offrir un interface des plus intuitifs et réactifs qui soit. Pour ce qui est du backend, nous avons opté pour le langage **Rust**. C'est un langage plus que rapide, «**blazingly fast**<sup>1</sup>» même, et qui contient une panoplie de fonctionnalités relatives à la sécurité et à l'intégrité de la mémoire, ce qui le rend une option incontournable pour la partie serveur d'une base de données. Nous avons utilisé le crate Tiberius pour se connecter à la base de donnée MSSQL.

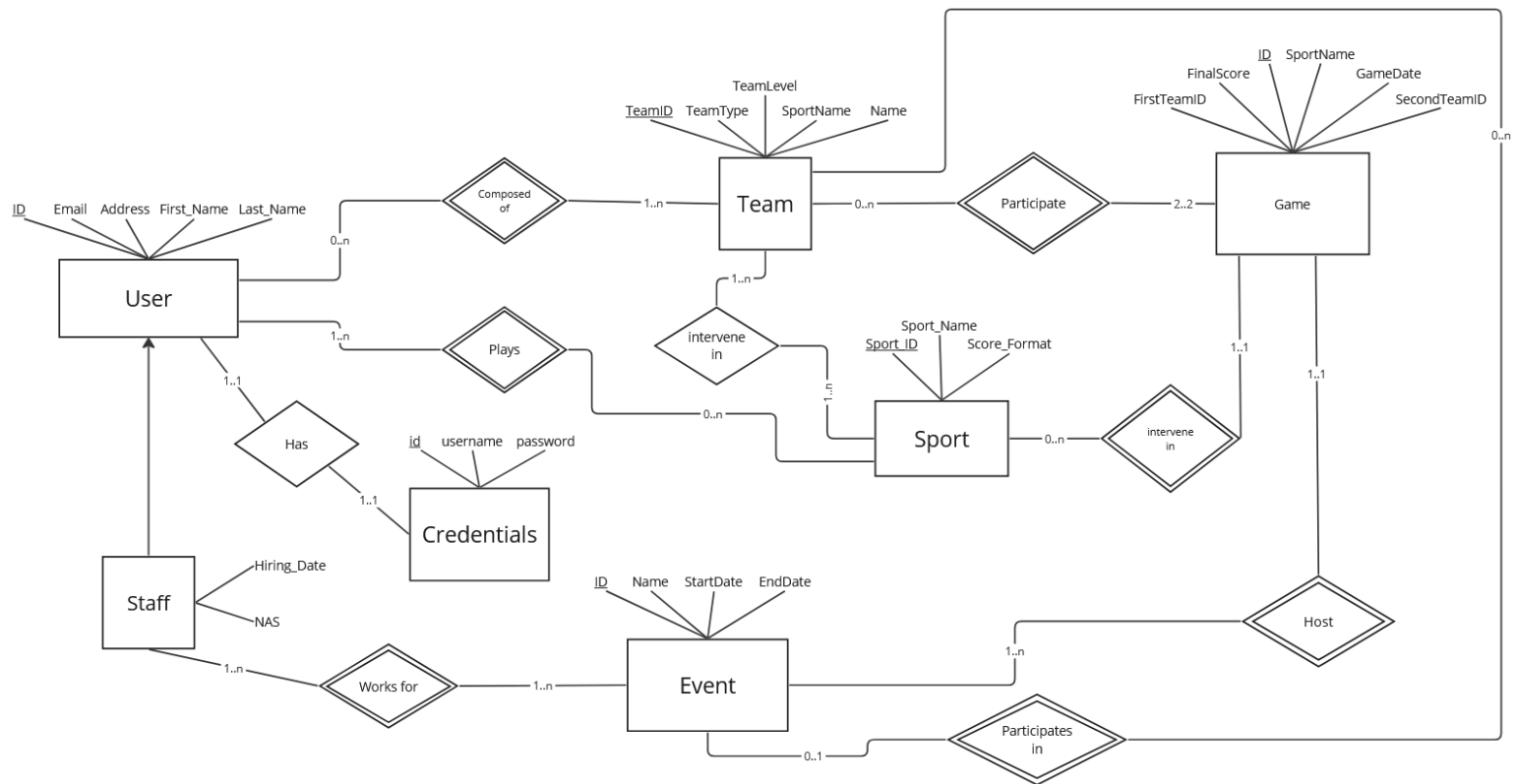
Le présent document fait état des éléments suivants relatifs à l'application :

- 1) La représentation des entités et associations pertinentes à la structure de la base de données relationnelle
- 2) Le schéma relationnel associé à la base de données
- 3) La normalisation de la base de données dans les trois formes normales (1FN, 2FN, 3FN)
- 4) La définition de la base de données (LDD de SQL Server)
- 5) La manipulation des données dans la base de données (LMD)
- 6) La liste des requêtes et l'application en SQL
- 7) Le fonctionnement de l'application avec illustrations

---

1. Selon ThePrimeagen

# Représentation Entité-Association



# Schéma Relationnel

Users(ID, Email, Address, FirstName, LastName)

Players(#UserID, #TeamID)

Staff(#UserID, NAS, HiringDate)

Teams(ID, Name, #TeamLevel, #TeamType, #SportName)

TeamLevel(TeamLevel)

TeamType(Type)

Sports(Name, minJoueurs, maxJoueurs, scoreFormat)

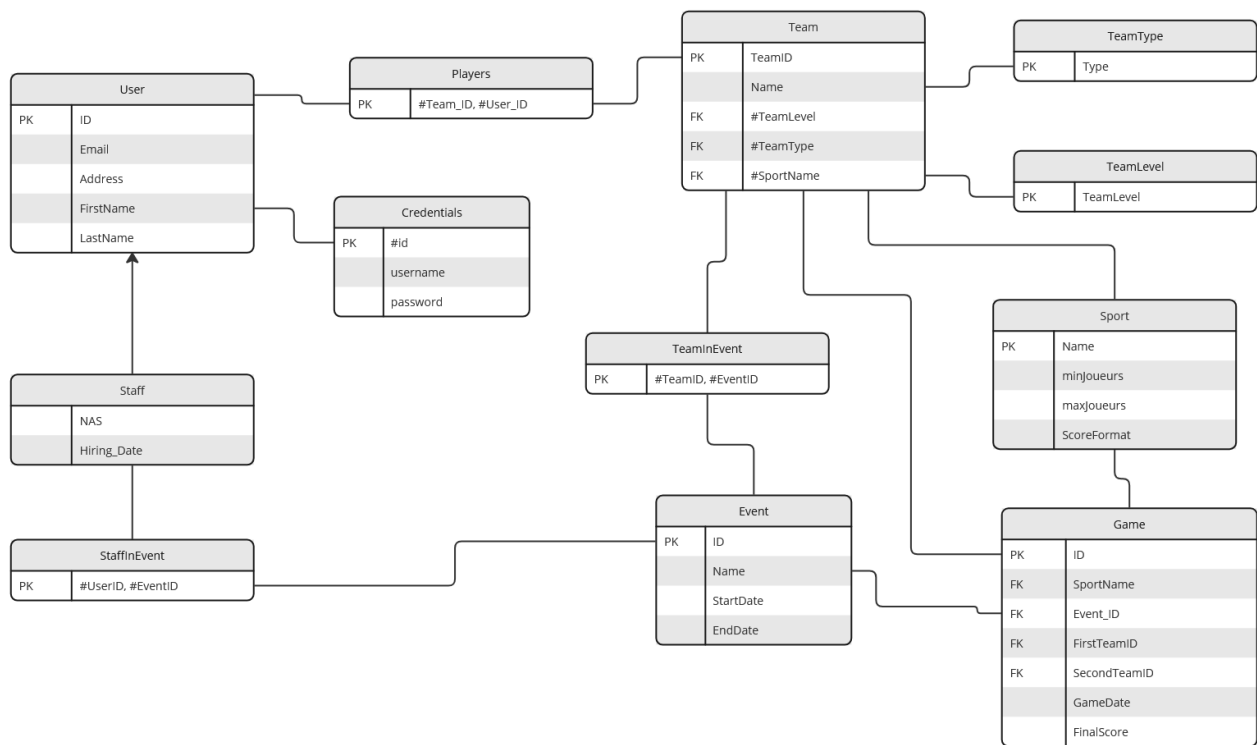
Events(ID, Name, StartDate, EndDate)

TeamInEvent(#EventID, #TeamID)

StaffInEvent(#UserID, #EventID)

Games(ID, #SportName, #EventID, #FirstTeamID, #SecondTeamID, GameDate, FinalScore)

Credentials(#id, username, password)



## Explications du modèle

Avant toute chose, nous devons mentionner que certaines différences mineures peuvent être présentes entre le schéma Entité-Association (E-A) et le modèle relationnel explicité ci-haut. Cela est dû au fait que le modèle relationnel présenté est une représentation des relations qui sont présentement implémentées en SQL, alors que le schéma E-A est notre vision concernant l'objectif à atteindre.

Notre implémentation fait qu'un User devient un Player quand il fait partie d'une équipe et à cette Équipe est attribué un Sport, ce qui permet à un même utilisateur de pratiquer plusieurs sports et même d'être dans plusieurs équipes du même sport.

De manière similaire, une équipe se voit attribuer un niveau (ou calibre) et un type. Ces trois derniers éléments forment des tables afin qu'il soit possible de contrôler ce qui constitue un sport, un niveau et un type d'équipe.

Un Event peut abriter plusieurs parties qui lui sont associées comme dans le cas d'un tournoi, dans chacune de ces parties, 2 équipes d'un même Sport s'affrontent à une date déterminée après laquelle on sauvegarde le FinalScore.

Parallèlement, un User peut aussi être membre du personnel employé, auquel cas il se verra attribuer quelques attributs supplémentaires par son inscription dans la table Staff. Il s'agit d'une instance de polymorphisme puisque Staff garde la même clef primaire que User. Cet employé peut ensuite être associé à des événements au travers de la table StaffInEvent.

La table Credentials est séparée de la table Users pour s'assurer de l'intégrité des données qui s'y trouvent. Une autre approche que nous aurions pu adopter pour un résultat similaire serait d'inclure Credentials dans la table User en lui ajoutant simplement le username et password, en restreignant l'accès à cette table pour l'utilisateur et en créant une vue spécifique pour accéder aux données.

Ce schéma relationnel découle essentiellement du schéma Entité-Association duquel nous avons vérifié la normalisation des tables, ce que nous verrons plus en détail à la section suivante.

## Normalisation

Il est important de noter que toutes les tables de notre application sont d'abord en première forme normale (1FN) car tous les attributs sont atomiques. Ensuite, toutes nos tables sont aussi en deuxième forme normale (2FN) puisque de toutes les tables ayant des clefs composées, donc Players, TeamInEvent et StaffInEvent n'ont aucun attribut qui dépend seulement d'une partie de la clef primaire de sa table. Et Finalement, toutes nos tables sont aussi en troisième forme normale (3FN) car aucun attribut non-clef n'est dépendant d'autres attributs non-clefs. Notez également que nous n'avons pas eu à utiliser de technique de normalisation puisque nos tables respectaient déjà les trois formes lorsque nous nous sommes penché sur la question.

## LDD, LMD et requêtes

Les fichiers contenant la définition des tables et l'insertion initiale de données ainsi que les procédures, triggers et curseurs, et les requêtes de l'application sont joints sous la forme des fichiers LDD\_LMD.sql, T-SQL.sql et requetes.sql respectivement. Notez cependant que certains éléments de T-SQL présents dans le fichier .sql n'ont pas été en mesure d'être implémentés dans l'application par manque d'effectif<sup>2</sup>. Il en va de même pour certaines des requêtes contenues dans le fichier requetes.sql.

---

2. Malheureusement, nous avons subi le départ tardif d'un membre de notre équipe initiale. Conjugué à l'objectif ambitieux que nous nous étions fixé, la perte d'un coéquipier si tard dans le processus a limité significativement notre capacité à atteindre notre objectif

## Résultat

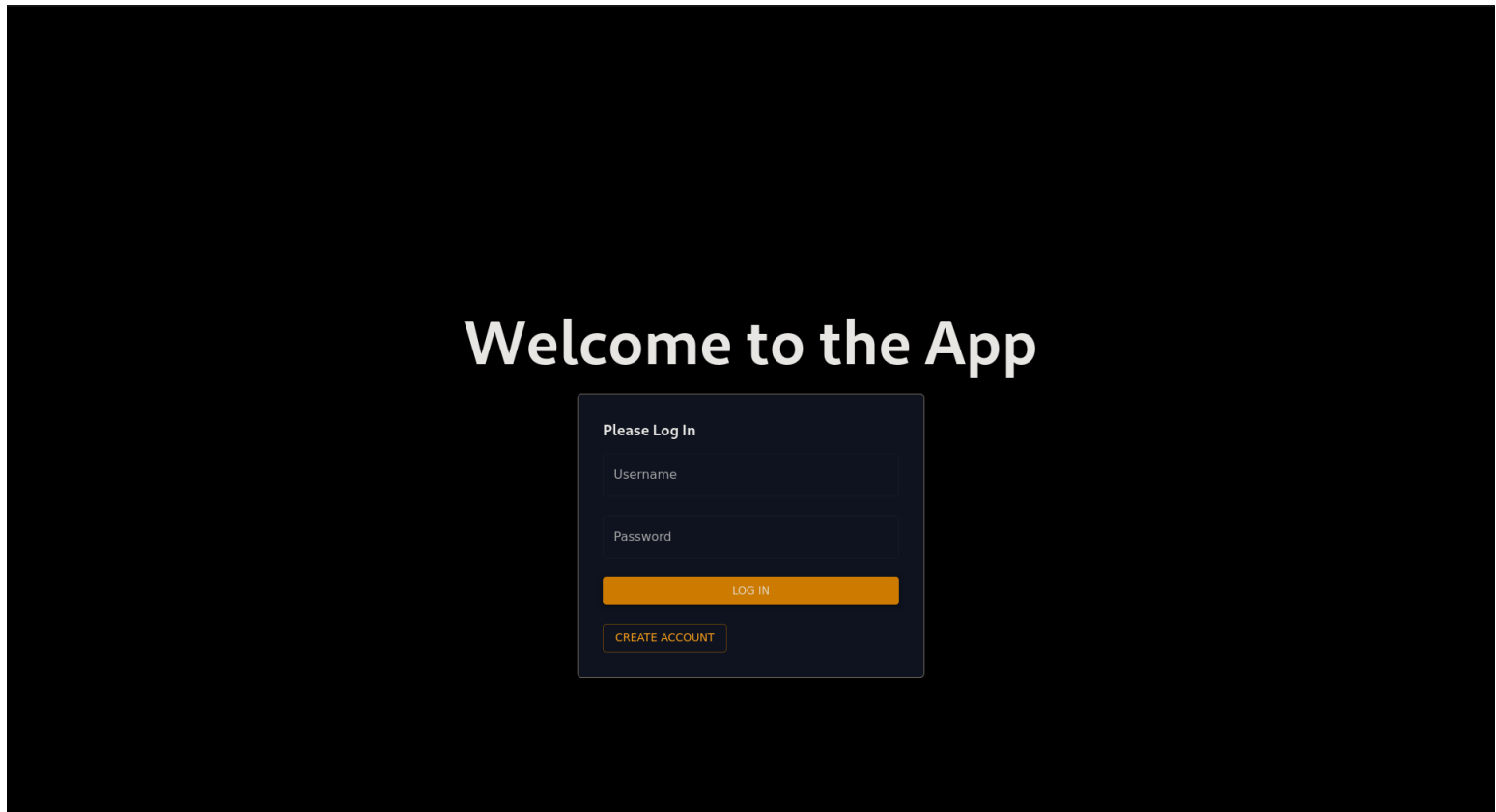
Au final, bien que nous sommes une équipe réduite, nous sommes arrivés à créer un squelette de produit ayant un gros potentiel.

Plusieurs éléments construits dans l'application sont réutilisables pour implémenter les requêtes et éléments de T-SQL présents dans les fichiers, mais non-déjà implémentés dans l'application moyennant des modifications marginales. On pense notamment au mécanisme d'affichage des tables dans l'application qui peut servir presque directement à afficher le résultat de la vue pour obtenir les équipes ayant joué le plus de parties ou celle pour obtenir les événements en cours les plus populaires.

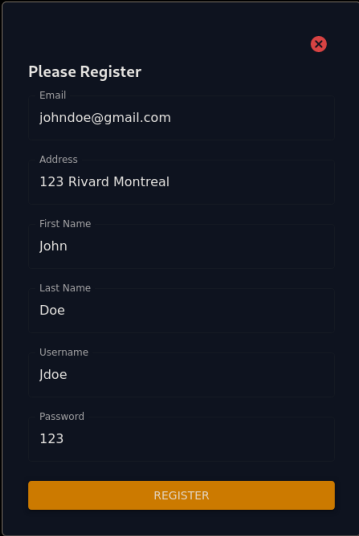
À l'inverse, certaines fonctionnalités implémentées plus tôt dans l'application bénéficieraient autant au niveau de l'efficacité que de la lisibilité et maintenabilité du code à utiliser certaines fonction ou procédures créées plus tard dans le processus de création de l'application et qui, en conséquence, n'ont pas pu être implémentées à temps.

Il en reste que l'application est en soit facile d'utilisation, a une apparence soignée et que les fonctionnalités pleinement implémentées sont extrêmement rapides.

## Visite de l'application



**FIGURE 1** – Écran d'accueil/Login



**Please Register**

Email  
johndoe@gmail.com

Address  
123 Rivard Montreal

First Name  
John

Last Name  
Doe

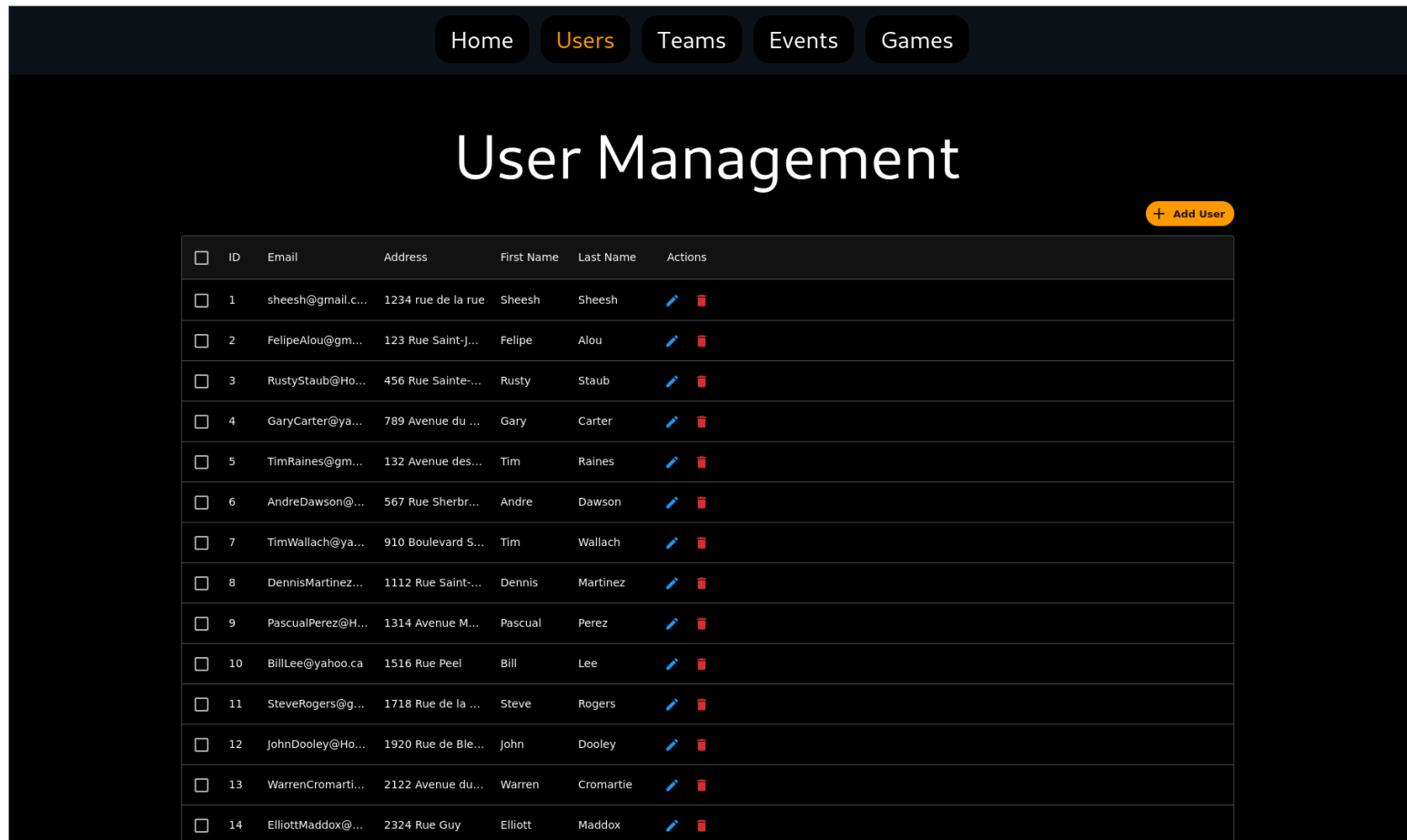
Username  
jdoe

Password  
123

REGISTER

**FIGURE 2** – Création d'un nouvel utilisateur





**FIGURE 3** – Écran qui affiche les utilisateur de la Base de Donnée.  
On peut y modifier les informations des utilisateurs et en ajouter/supprimer des nouveaux à l'aide des boutons correspondants. Il n'est pas destiné à être accessible à tous les utilisateurs.

HomeUsersTeamsEventsGames

View My TeamsView All Teams

Team Management

+ Add Team

| <input type="checkbox"/> | ID | Team Name         | Level        | Type      | Sport      | Actions                             |
|--------------------------|----|-------------------|--------------|-----------|------------|-------------------------------------|
| <input type="checkbox"/> | 1  | Les Tigres        | Junior       | Mixed     | Soccer     | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 2  | The Sluggers      | Competitive  | Masculine | Baseball   | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 3  | Les Hard Hitters  | Competitive  | Masculine | Baseball   | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 4  | The Kicks         | Junior       | Mixed     | Soccer     | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 5  | The Beach Girls   | Recreational | Feminine  | Volleyball | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 6  | VolleyGirls       | Recreational | Feminine  | Volleyball | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 7  | The Blundettos    | Competitive  | Mixed     | Baseball   | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 8  | Les instincteurs  | Competitive  | Mixed     | Baseball   | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 9  | The Thunder Be... | Competitive  | Mixed     | Baseball   | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 10 | The Swamp Cats    | Competitive  | Mixed     | Baseball   | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 11 | Nebula Knights    | Recreational | Mixed     | Football   | <a href="#">✎</a> <a href="#">✖</a> |
| <input type="checkbox"/> | 12 | Phoenix Pirates   | Recreational | Mixed     | Football   | <a href="#">✎</a> <a href="#">✖</a> |

FIGURE 4 – Écran de gestion et d’affichage des différentes équipes

[Home](#) [Users](#) [Teams](#) [Events](#) [Games](#)

View My Events ☐

View All Events ☒

Event Management

Add Event

| <input type="checkbox"/> | ID | Name              | Start Date | End Date   | Actions |
|--------------------------|----|-------------------|------------|------------|---------|
| <input type="checkbox"/> | 1  | Tournois Bas...   | 2024-08-01 | 2024-08-02 |         |
| <input type="checkbox"/> | 2  | Soccer Mixte ...  | 2024-06-02 | 2024-06-04 |         |
| <input type="checkbox"/> | 3  | Beach Volley...   | 2024-07-20 | 2024-07-22 |         |
| <input type="checkbox"/> | 4  | Quebec City ...   | 2024-07-01 | 2024-07-04 |         |
| <input type="checkbox"/> | 5  | Gatineau Voll...  | 2024-05-17 | 2024-05-19 |         |
| <input type="checkbox"/> | 6  | Montreal Foo...   | 2024-11-22 | 2024-11-24 |         |
| <input type="checkbox"/> | 7  | Sherbrooke S...   | 2024-08-15 | 2024-08-18 |         |
| <input type="checkbox"/> | 8  | Trois-Rivières... | 2024-06-07 | 2024-06-09 |         |
| <input type="checkbox"/> | 9  | Laval Volleyb...  | 2024-03-15 | 2024-03-17 |         |

Rows per page:

100

1-9 of 9

FIGURE 5 – Écran de gestion et d’affichage des différents Événements

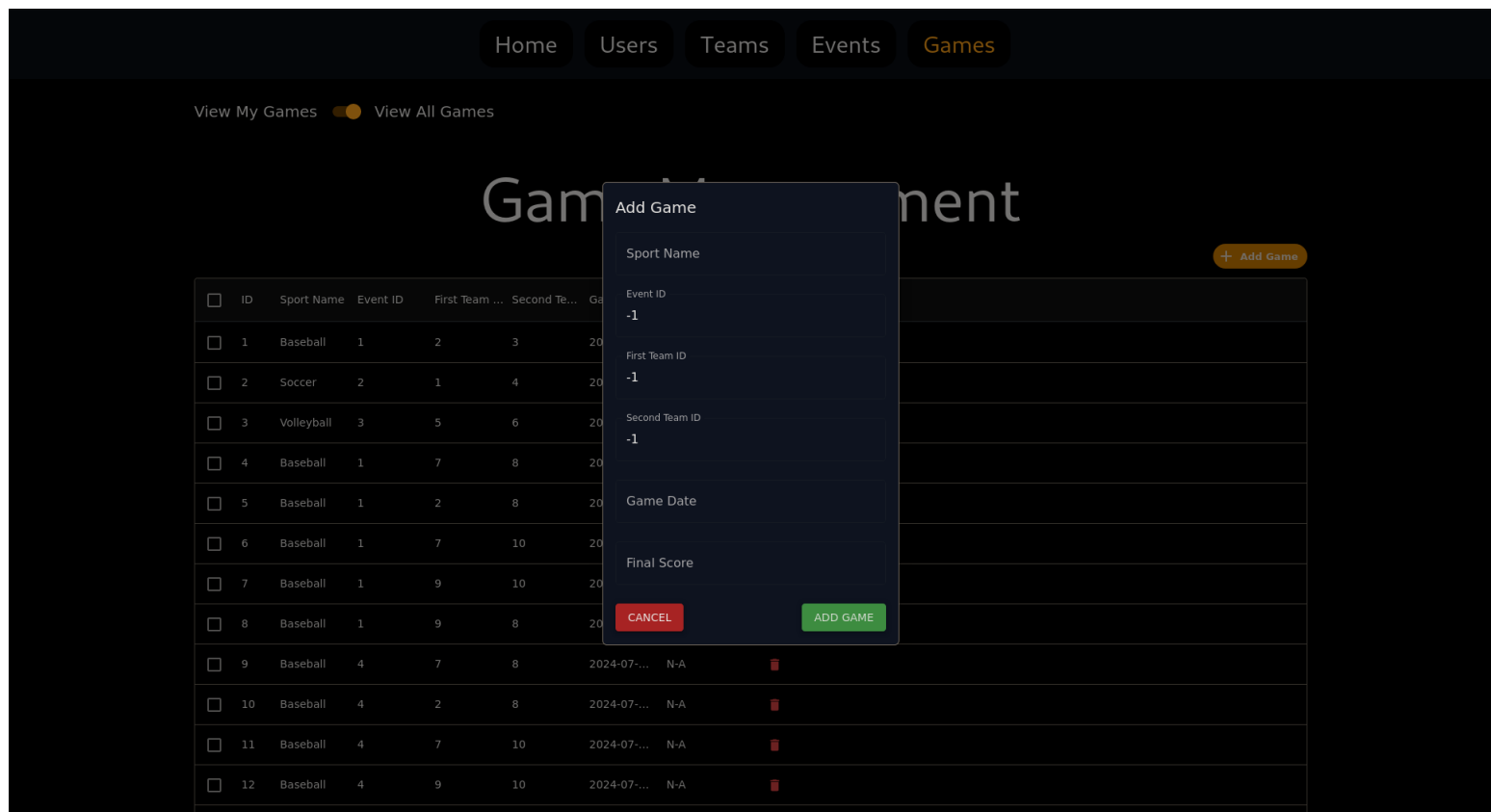


FIGURE 6 – Écran de gestion et d’affichage des parties. Ajout d’une partie en cours.

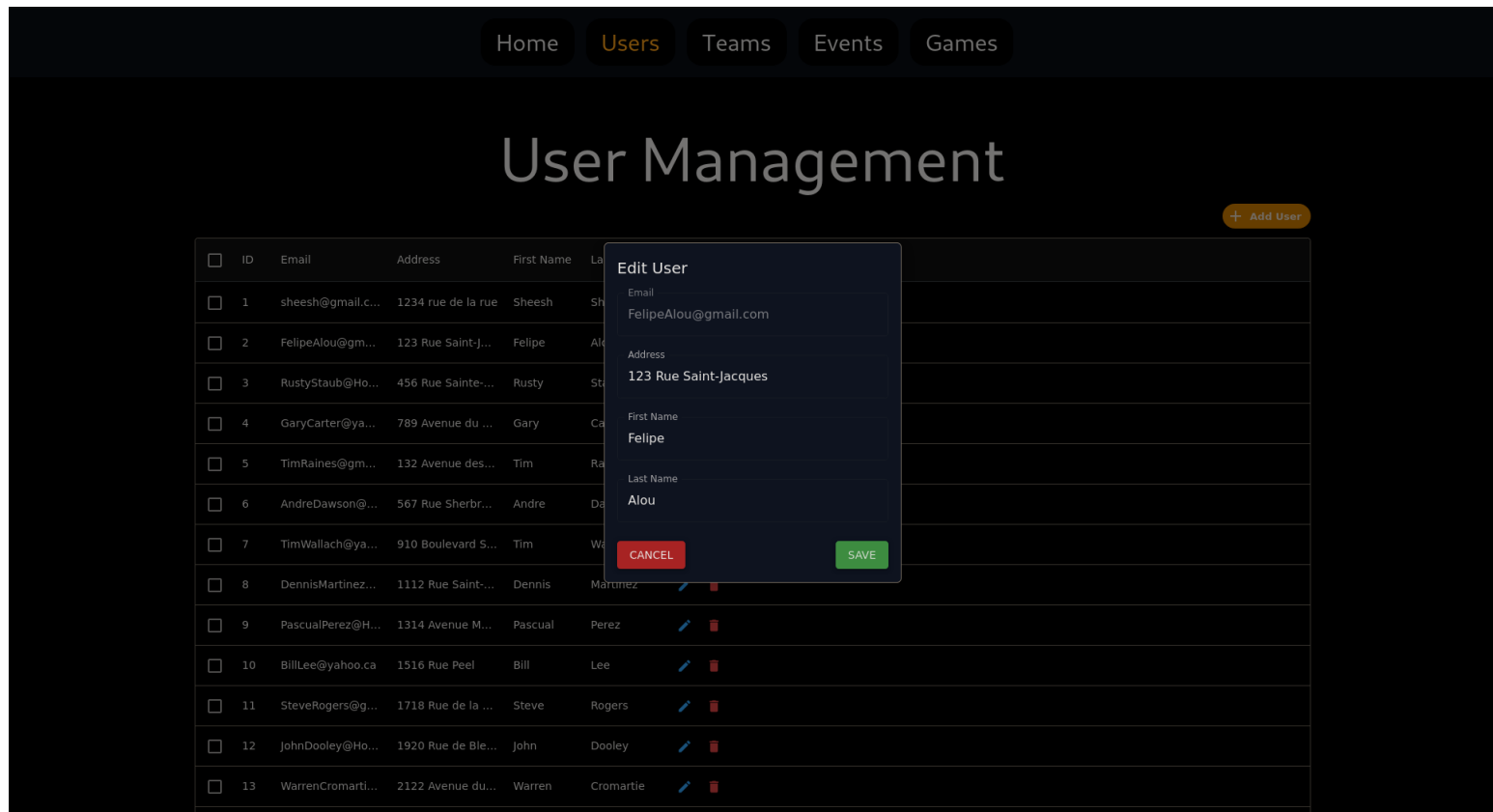


FIGURE 7 – Modification des données d'un utilisateur