

|  |
| --- |
| Application Joutes |

Table des matières

[1 Analyse préliminaire 3](#_Toc9341226)

[1.1 Introduction 3](#_Toc9341227)

[1.2 Organisation 3](#_Toc9341228)

[1.3 Objectifs et fonctionnalités 3](#_Toc9341229)

[1.4 Méthode utilisée 4](#_Toc9341230)

[1.5 Planification initiale 5](#_Toc9341231)

[1.6 Analyse / Conception 6](#_Toc9341232)

[1.7 Concept 6](#_Toc9341233)

[1.8 Stratégie de test 11](#_Toc9341234)

[1.9 Risques techniques 11](#_Toc9341235)

[1.10 Dossier de conception 12](#_Toc9341236)

[1.10.1 Maquettes : Wireframe 12](#_Toc9341237)

[1.10.2 Use Cases & Scénarios 18](#_Toc9341238)

[1.10.3 Base de données 20](#_Toc9341239)

[1.10.4 (Particularité 1) Routes 21](#_Toc9341240)

[1.10.5 (Particularité 2) Connexions utilisateurs avec SAML 21](#_Toc9341241)

[2 Réalisation 21](#_Toc9341242)

[2.1 Dossier de réalisation 21](#_Toc9341243)

[2.2 Description des tests effectués 21](#_Toc9341244)

[2.3 Erreurs restantes 21](#_Toc9341245)

[2.4 Liste des documents fournis 21](#_Toc9341246)

[3 Conclusions 21](#_Toc9341247)

[4 Annexes 22](#_Toc9341248)

[4.1 Résumé du rapport du TPI / version succincte de la documentation 22](#_Toc9341249)

[4.2 Sources – Bibliographie 22](#_Toc9341250)

[4.3 Journal de bord 22](#_Toc9341251)

[4.4 Manuel d'Installation 22](#_Toc9341252)

[4.5 Manuel d'Utilisation 22](#_Toc9341253)

[4.6 Archives du projet 22](#_Toc9341254)

# Analyse préliminaire

## Introduction

Dans le cadre ma formation, je vais effectuer un travail pratique individuel. Pour m’y préparer, j’ai effectué un pré-TPI d’env. 50h.

M. Carrel, professeur au CPNV, est mon client pour ce projet. Mon travail consiste à rajouter des fonctionnalités à l’application web de gestion des Joutes du CPNV. Cette application est un site web, écrit en PHP, avec le Framework Laravel. D’autres langages sont également utilisé, comme l’HTML, CSS, JavaScript, Node.JS, SAML, etc.

Cette application a été développée par les techniciens ES de l’école. L’application est en constante évolution. Mon projet est disponible sur une branche du dépôt GitHub (CPNV-ES/Joutes), qui pourra être fusionne à la branche principale, si mes fonctionnalités sont finies et mon code de bonne qualité. Ma branche se base sur la branche master, version du 8 Février 2019 (commit « 914baf1 »).

Mon TPI sera effectué sur une durée d’environ 90 heures.

## Organisation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Statut** | **Nom** | **Prénom** | **Adrese Email** | **Numéro de tél.** |
| Elève 1 | Germann | Niels | [Niels.germann@cpnv.ch](mailto:Niels.germann@cpnv.ch) | 079/947.14.70 |
| Expert 1 | Roy | Alain | [Alain.roy@vd.oai.ch](mailto:Alain.roy@vd.oai.ch) | 079/444.01.54 |
| Expert 2 | Bertino | Yves | [yves@bertino.ch](mailto:yves@bertino.ch) | 076 540 61 52 |
| Chef de projet | Favre | Raphaël | [Raphael.favre@cpnv.ch](mailto:Raphael.favre@cpnv.ch) | 076 427 93 59 |

## Objectifs et fonctionnalités

Tout d’abord, mes objectifs personnels sont (de continuer) d’apprendre le Framework Laravel, en particulier certains aspects, comme les routes, la création de formulaires avec Blade et la gestion de notifications.

Je souhaite également m’améliorer sur Eloquent, librairie intégrée à Laravel, permettant de créer des requêtes SQL en orienté-objet.

Je désirerais également améliorer ma gestion/compréhension des objets dans Laravel. J’avais eu quelques problèmes pour récupérer des informations contenues dans des objets lors de mon pré-TPI.

Concernant mon projet, ma première fonctionnalité consiste à créer un classement général par tournoi, qui prendrait en compte le nombre de matchs joué par une équipe lors du calcul du classement. Cette fonctionnalité doit prendre en compte les particularités possibles de chaque phase de pool.

Ensuite, ma deuxième fonctionnalité est de créer un historique des tournois par sport.

Depuis cette page, un administrateur pourra dupliquer un tournoi dans un autre (paramètres généraux, pools, etc.)

Ensuite, ma troisième fonctionnalité est de créer un palmarès individuel, qui permet à chaque participant de consulter tous ses résultats dans tous les tournois auxquels il a participé au fil des années. Il pourra également voir les matchs joués par son équipe dans chaque tournoi.

Pour finir, mes objectifs généraux sont d’avoir une utilisation correcte du Framework Laravel (code propre), des commentaires dans mon code, des tests de qualités ainsi qu’un manuel de mise en service de bonne qualité.

## Méthode utilisée

Pour ce qui est de ma planification détaillée, j’utilise Trello, un site web permettant la gestion de projet en ligne, en utilisant une méthode de travail en sprint (un sprint par semaine), comme discuté avec mon chef de projet.

J’utilise également Trello comme plateforme de test ; C’est ici qu’avec mon chef de projet nous allons vérifier et valider chaque point de chaque tableau (fonctionnalité ou élément de la documentation) à la fin de chaque sprint.

Mon tableau Trello est disponible en ligne en [**cliquant ici**](https://trello.com/invite/b/O0EBgruw/77069c6db0b533f4f3898967980ab79c/tpi-niels)

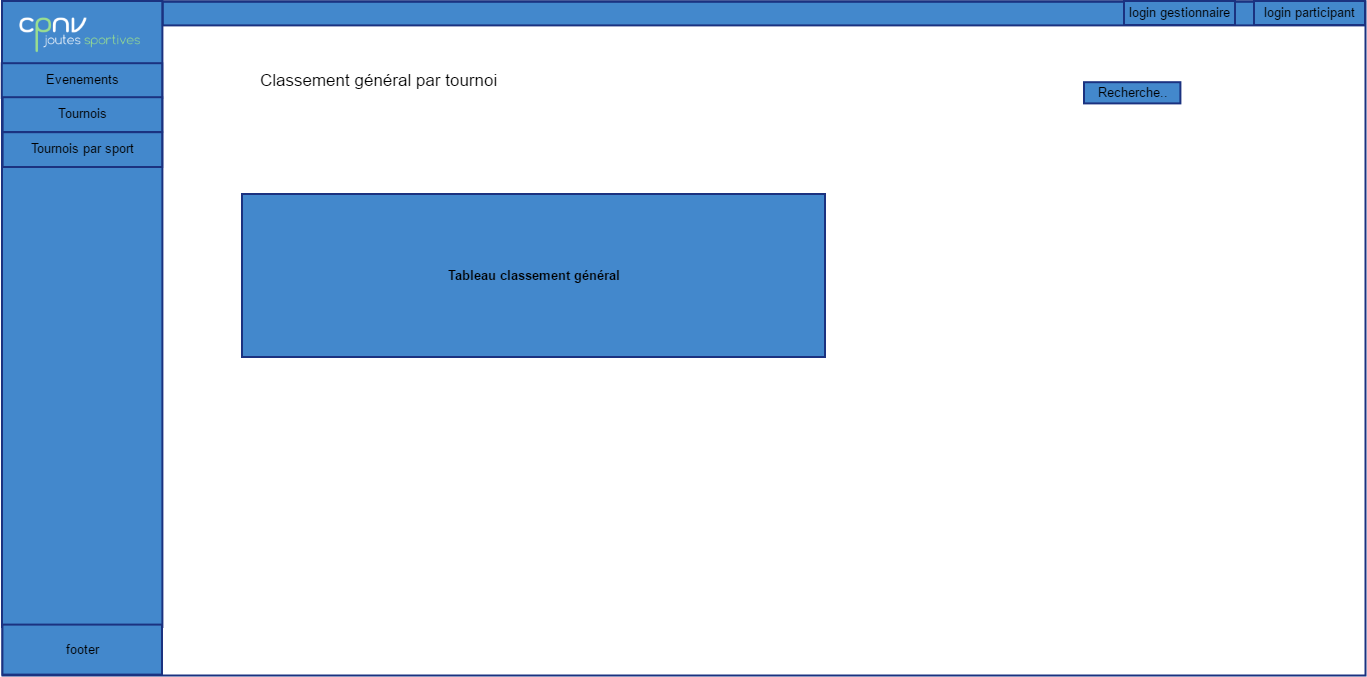
(<https://trello.com/invite/b/O0EBgruw/77069c6db0b533f4f3898967980ab79c/tpi-niels>)

## Planification initiale

## Analyse / Conception

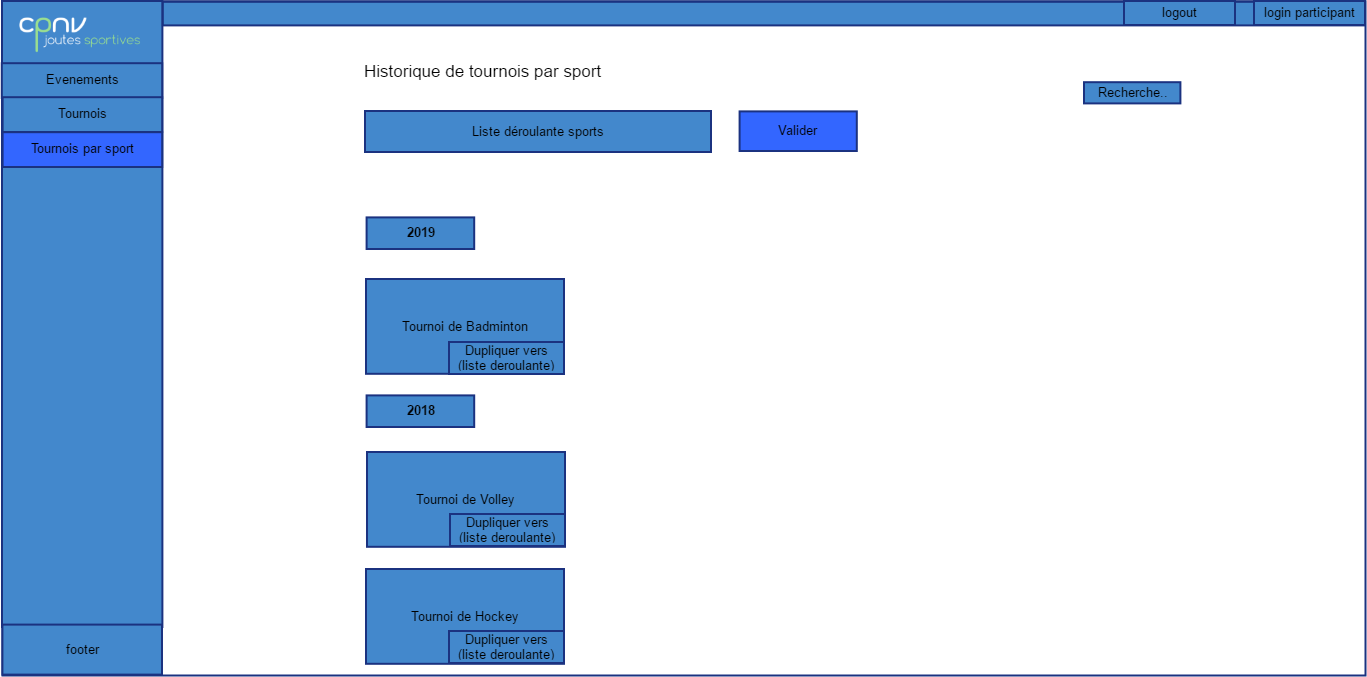
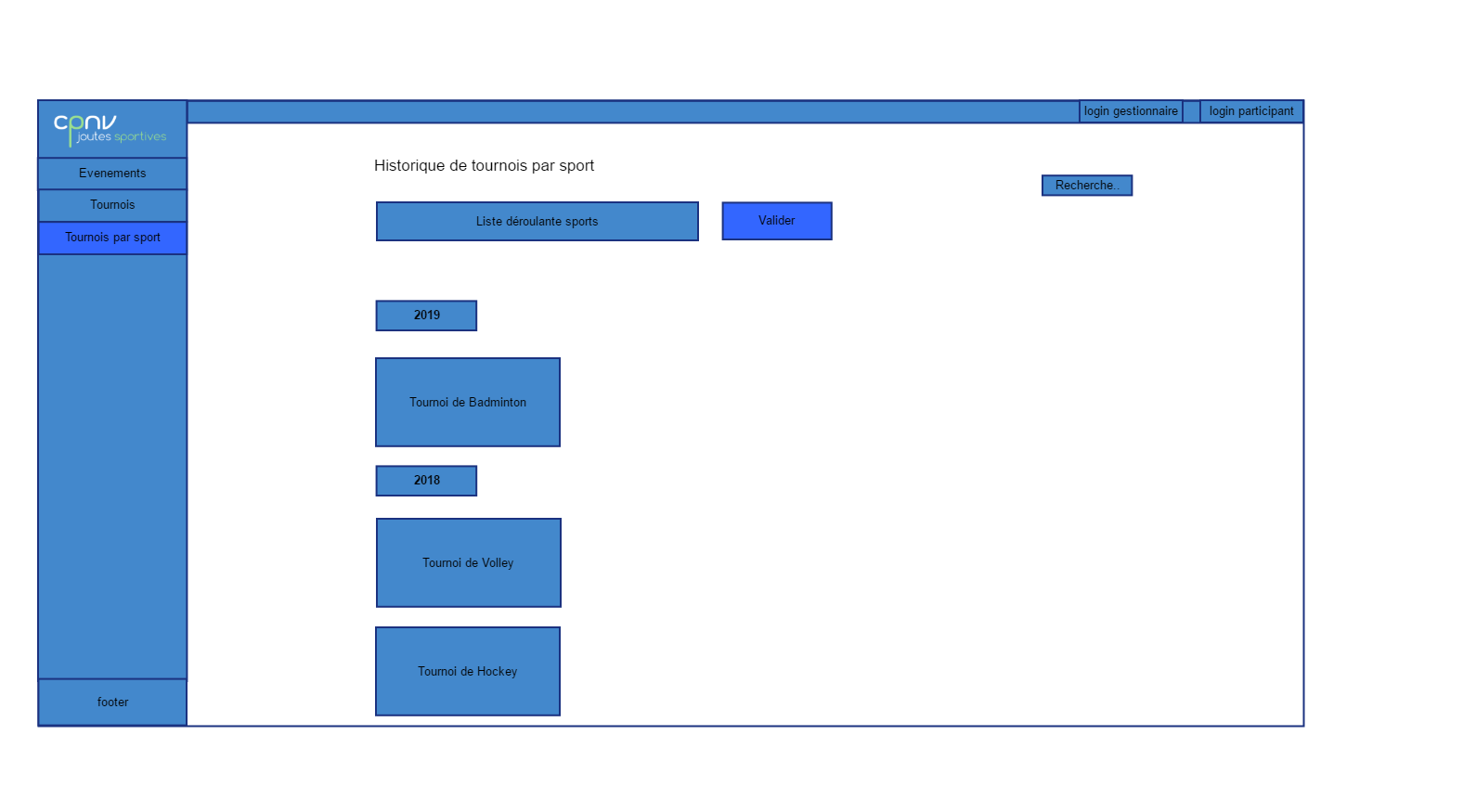
## Concept – Zoning

Je me suis basé sur les pages existantes du projet pour l’alignement de mes blocs, en particulier sur les tableaux de la page des tournois.

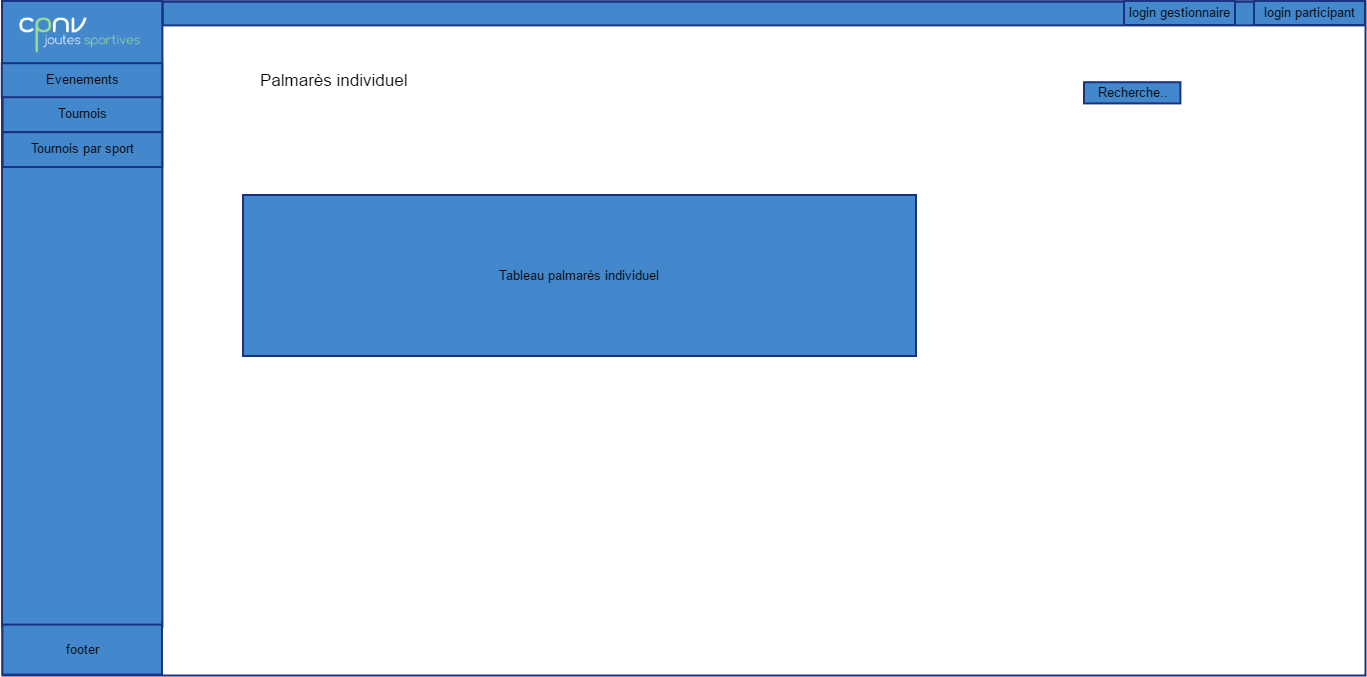


Fonctionnalité 1 : Page classement général - zoning

Fonctionnalité 2 : Page d’historique de tournois par sport - vue participant/visiteur - zoning

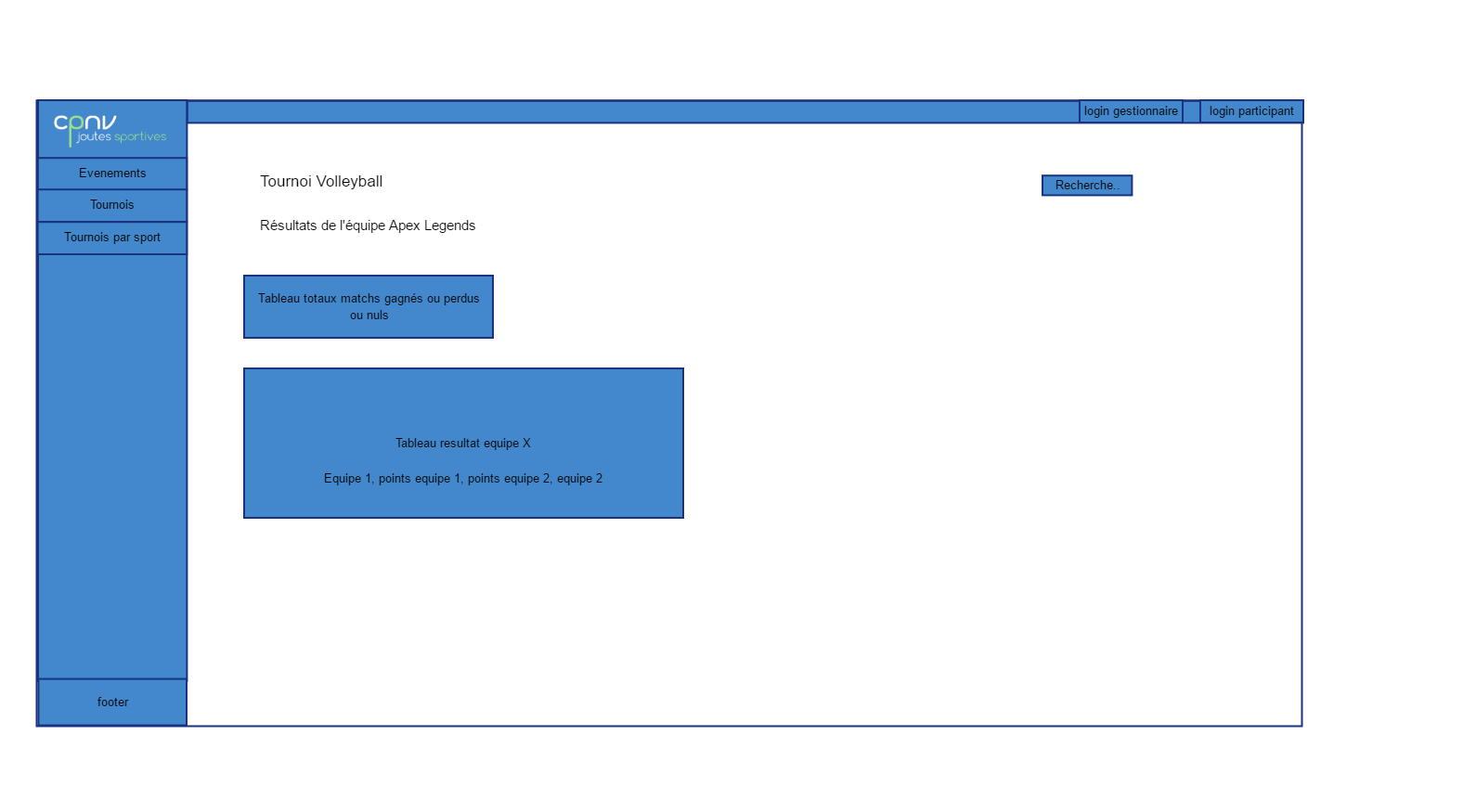


Fonctionnalité 2 : Page d’historique de tournois par sport - vue admin - zoning



Fonctionnalité 3 : Page palmarès individuel - zoning

Fonctionnalité 3 : Page du score de l’équipe d’un participant à un tournoi - zoning



## Stratégie de test

Pour vérifier mes fonctionnalités, je vais tout d’abord faire des tests unitaires, puis des tests d’intégration.

Mes tests unitaires permettront de tester chaque fonctionnalité. Les tests d’intégration permettront de vérifier que toutes les fonctionnalités fonctionnent ensemble, comme serait utilisé le site en condition réelle.

Je vais finir par effectuer des tests fonctionnels ainsi que des tests de performances.

## Risques techniques

Pour commencer, je suis relativement peu à l’aise avec mon projet, même si j’ai réalisé un pré-TPI. En particulier car le langage de programmation est différent, car je vais devoir utilisé Laravel.

Lors de mon Pré-TPI, je n’avais pas bien configuré les routes du projet. Il va falloir que je me renseigne pour les configurer correctement, en particulier en regardant la documentation officielle de Laravel, sur leur site web.

M. Carrel ma également dit que je devrais utiliser un « helper » pour créer les formulaires, chose que je n’ai pas utilisé lors de mon Pré-TPI. Je vais également m’instruire sur le site officiel de Laravel, dans l’aide de Blade et des helpers.

Même si les fonctionnalités que je dois implémenter on l’air compliquées, le fait que je n’ai pas de modifications à faire sur la base de données me fait dire que je devrais avoir le temps de finir correctement mon projet.

La première fonctionnalité est, je pense, celle qui va me prendre le plus de temps, car le classement général doit prendre en compte toutes les situations particulières (pools différentes).

## Historique des modifications

Dernièrement, j’ai eu un meeting avec mon client, M. Carrel. Il m’a montré en particulier un graphique, avec joueurs, pools et classement, qui va beaucoup m’aider pour créer un classement général efficace. J’avais également mal compris la fonctionnalité 3, qui sera en partie basée sur la fonctionnalité 1. M. Carrel m’a également aidé pour cela.

Je n’avais pas compris au début que la page de ma fonctionnalité 3 soit liée à la fonctionnalité 1. Je n’avais pas compris que le palmarès individuel affichait la plupart des informations du classement général (Classement, score, etc.), pour un utilisateur en particulier, bien entendu. J’ai donc dû modifier mes maquettes, en semaine 3.

## Dossier de conception

Pour mener ce projet à bien, je vais utiliser un ordinateur Dell OptiPlex 7050, qui appartient au CPNV et qui tournera sur Windows 10 Edu x64 v1709 (16299.1087)

|  |  |
| --- | --- |
| **Logiciel** | **Version** |
| PHPStorm (IDE) | 2018.3.6 |
| Vagrant | 2.2.4 |
| VirtualBox | 6.0.6 |
| Laravel Homestead | 7.2.1 |
| Git | 2.21 |
| Firefox | 66.0.5 x64 |
| Pencil | 3.0.4 |
| MySQL Workbench | 8.0.15 |
| MS Office - Word | 2016 |
| MS Office – Excel | 2016 |
| MS Office - Project | 2016 |
| PHP | 7.3.5 |

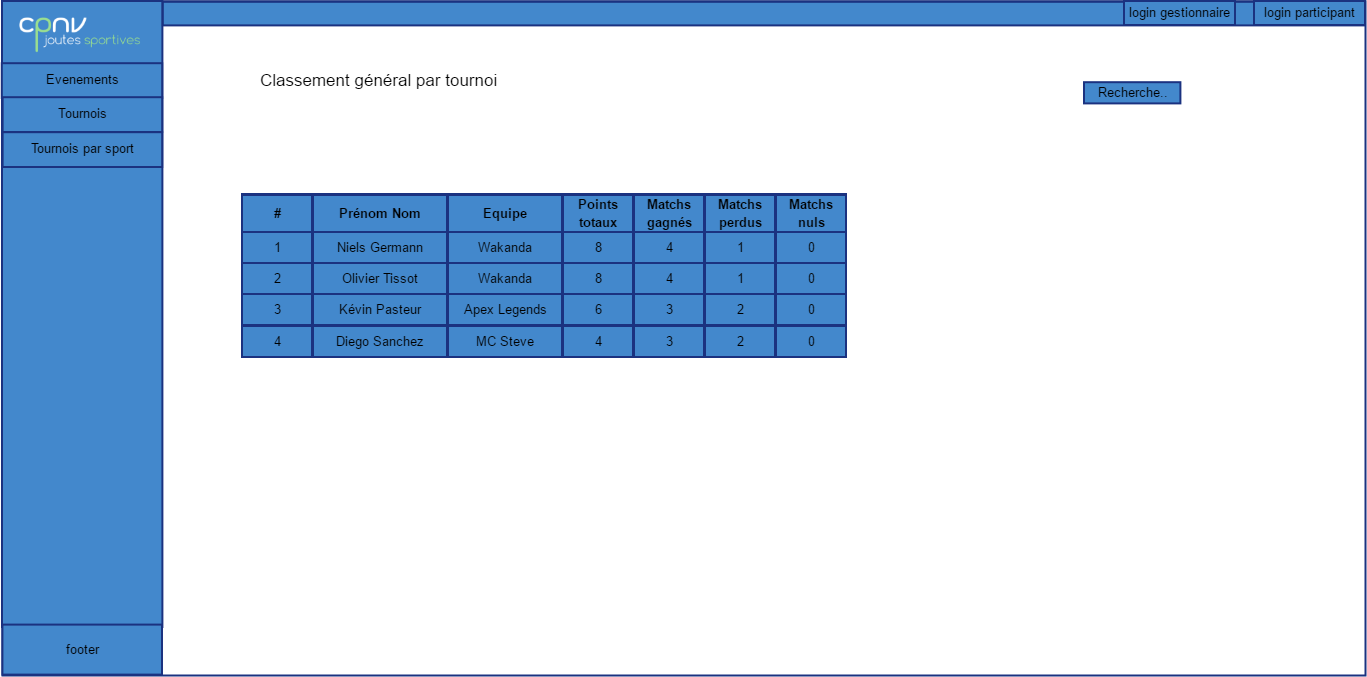
Laravel Homestead est une « box » pour Vagrant (template). Celui-ci permet de créer et configurer des environnements de développement virtuel.

En pratique, Laravel Homestead tourne sur une machine virtuel Ubuntu 18.04, sans interface graphique. Grâce à Homestead, je n’ai pas à installer PHP, un serveur web, une base de données, et d’autres logiciels intégrés. Le serveur web intégré est Nginx. J’utiliserai également le serveur MySQL intégré

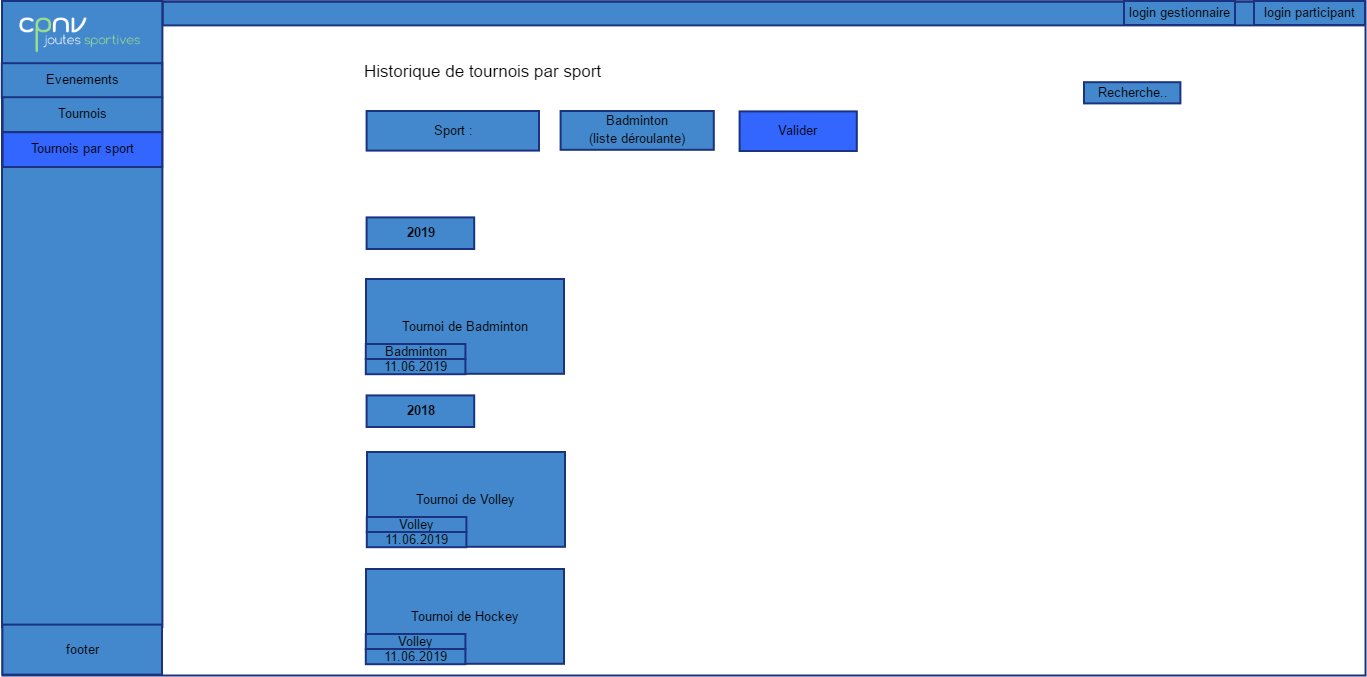
(Liste des logiciels inclus : <https://laravel.com/docs/5.8/homestead#introduction>)

### Maquettes : Wireframe

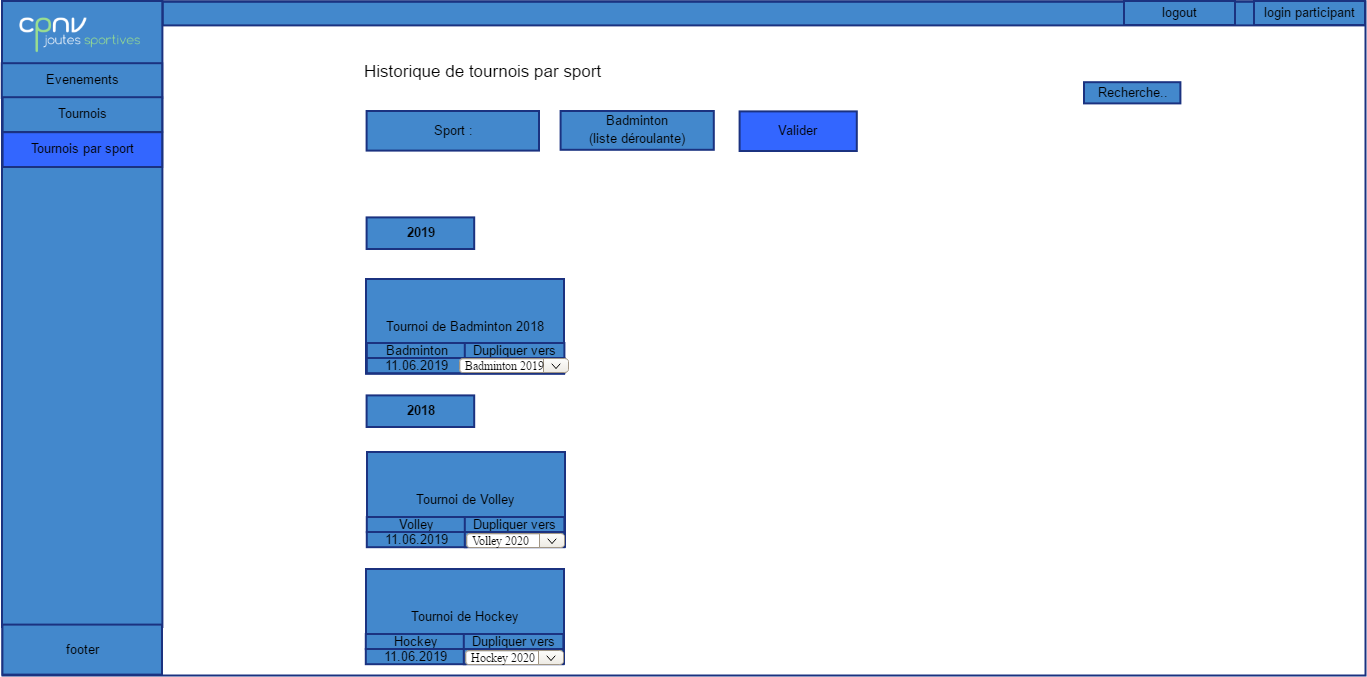
Les zonings sont disponibles au point **2.1**.

Fonctionnalité 1 : Page classement général - wireframe

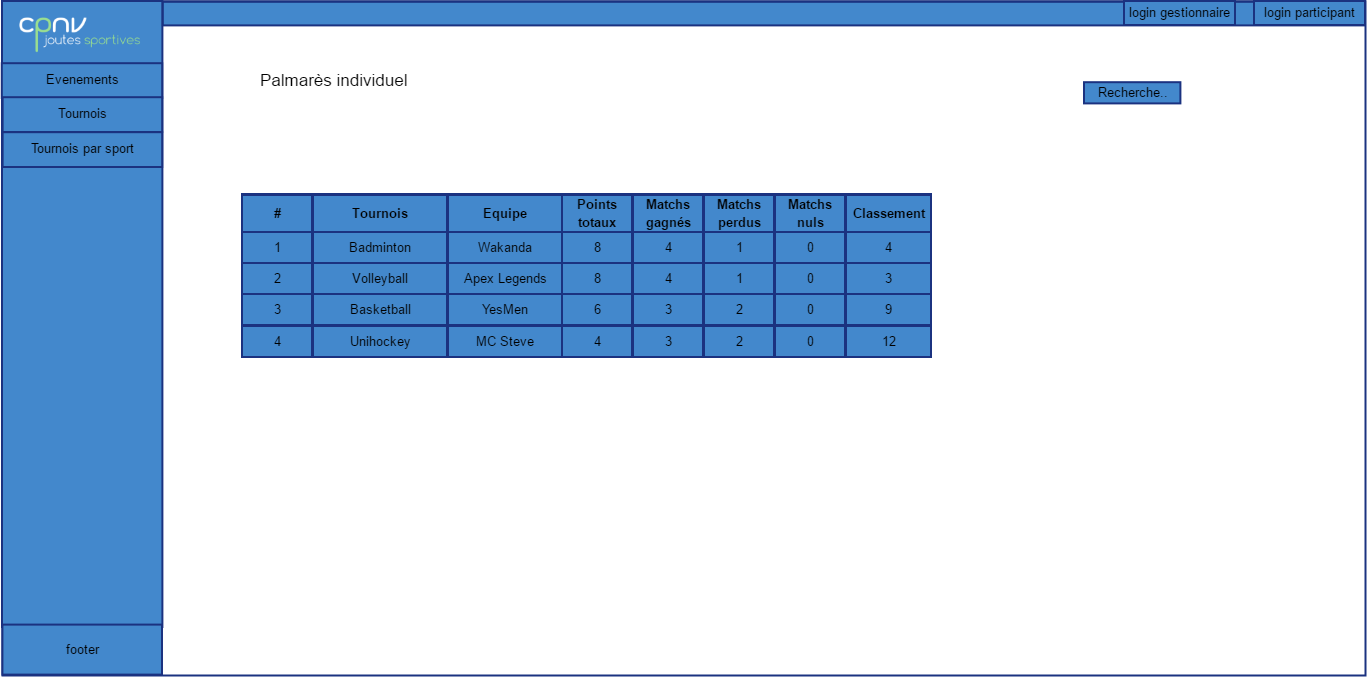
Fonctionnalité 2 : Page d’historique de tournois par sport - vue participant/visiteur – wireframe



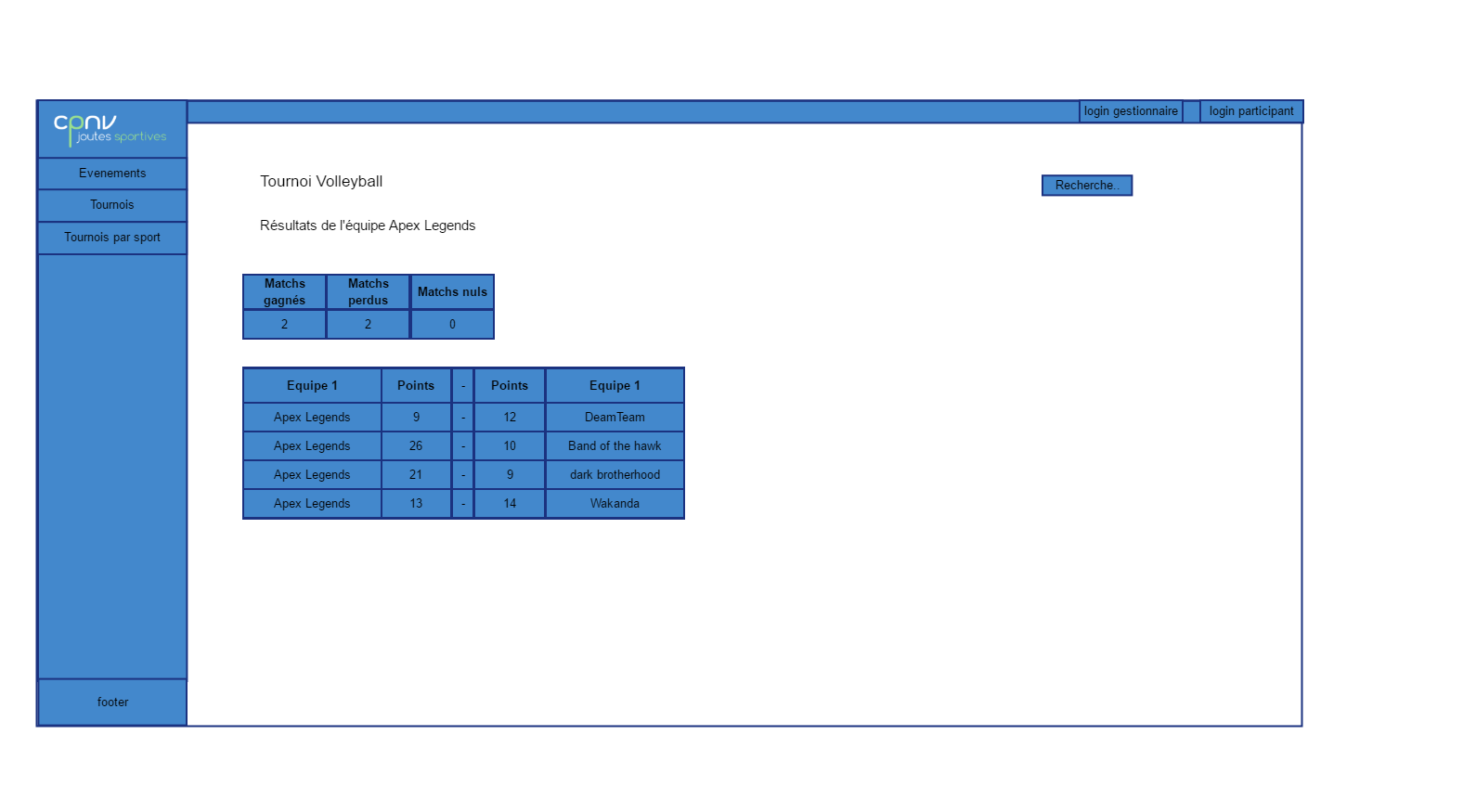
Fonctionnalité 2 : Page d’historique de tournois par sport - vue admin (duplication) – wireframe



Fonctionnalité 3 : Page palmarès individuel – wireframe



Fonctionnalité 3 : Page du score de l’équipe d’un participant à un tournoi - wireframe



### Use Cases & Scénarios

**Use Case fonctionnalité 2**

|  |  |  |
| --- | --- | --- |
| En tant que visiteur/participant | Je veux m’informer sur les tournois ayant eu lieu. | Afin d’être au courant des tournois qui ont eu lieu chaque année pour ce sport |

**Scénario fonctionnalité 2** : Page d’historique de tournois par sport

|  |  |
| --- | --- |
| **Action utilisateur** | **Réponse du site** |
| L’utilisateur est connecté. Il clique sur le bouton « Tournois par sport » dans le menu latéral à gauche. | Il est redirigé sur la page d’historique de tournois par sport |
| Il choisis un sport, badminton, dans la liste déroulante et appuie sur valider | La liste des tournois de badminton s’affiche en dessous, triés par année.  (Il peut accéder à la fonctionnalité 1, le classement général, en cliquant sur un tournoi) |

**Scénario fonctionnalité 2** : Page d’historique de tournois par sport – (**duplication**)

|  |  |
| --- | --- |
| **Action utilisateur** | **Réponse du site** |
| L’utilisateur est connecté. Il clique sur le bouton « Tournois par sport » dans le menu latéral. | Il est redirigé sur la page d’historique de tournois par sport |
| Il choisis un sport, badminton, dans la liste déroulante et appuie sur valider | La liste des tournois de badminton s’affiche en dessous, triés par année. |
| Il clique sur le bouton de duplication, à droite d’un des tournois. Le premier tournoi de badminton | Un liste déroulante et un bouton s’affiche, à droite du bouton de duplication |
| Il choisit un tournoi dans la liste déroulante. Un tournoi récemment créer, qui n’a pas eu lieu. Un tournoi de badminton. Il appuie sur le bouton de validation | Une notification l’avertit que les paramètres de ce tournoi ont été dupliqué dans celui sélectionné dans la liste déroulante. |

**Use Case fonctionnalité 1**

|  |  |  |
| --- | --- | --- |
| En tant que visiteur/participant | Je veux visualiser les résultats de chaque tournoi | Afin de m’informer sur le score de chaque participant à ce tournoi |

**Scénario fonctionnalité 1** : Classement général d’un tournoi

|  |  |
| --- | --- |
| **Action utilisateur** | **Réponse du site** |
| L’utilisateur est connecté. Il clique sur le bouton « Tournois par sport » dans le menu latéral. | Il est redirigé sur la page d’historique de tournois par sport |
| Il choisis un sport, badminton, dans la liste déroulante et appuie sur valider | La liste des tournois de badminton s’affichent en dessous, triés par année. |
| Il clique sur un tournoi, badminton | Il est redirigé vers la page du classement général |
| Il lit le tableau, et appuie sur « matchs gagnés » | Le site lui trie le tableau par matchs gagnés |

**Use Case fonctionnalité 3**

|  |  |  |
| --- | --- | --- |
| En tant que participant | Je veux accéder à mon palmarès | Afin de pouvoir visualiser mes scores pour chaque tournoi que j’ai effectué |

**Fonctionnalité 3 :** Page de palmarès individuel

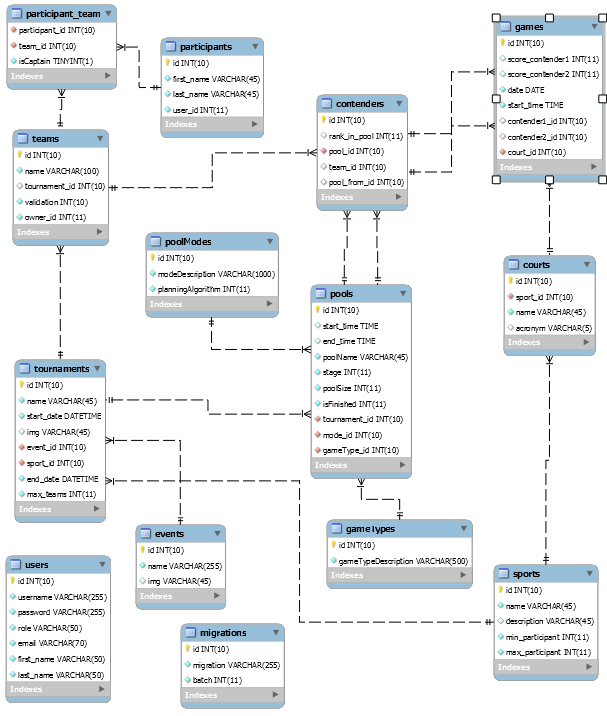
|  |  |
| --- | --- |
| **Action utilisateur** | **Réponse du site** |
| L’utilisateur est connecté. Il clique sur le bouton « Palmarès individuel » dans le menu latéral. | Il est redirigé sur son palmarès individuel. Un tableau avec les points marqués par les équipes dans lequel il a été inscrit (et les tournois auxquels il a participé) s’affiche. |

**Fonctionnalité 3 :** Page des matchs de son équipe dans ce tournoi

|  |  |
| --- | --- |
| **Action utilisateur** | **Réponse du site** |
| L’utilisateur est connecté. Il clique sur le bouton « Palmarès individuel » dans le menu latéral. | Il est redirigé sur son palmarès individuel. Un tableau avec les points marqués par les équipes dans lequel il a été inscrit (et les tournois auxquels il a participé) s’affiche. |
| Il clique sur la première ligne du tableau, sur la première équipe | Il est redirigé sur la page des matchs de son équipe.  Un tableau affiche la liste des matchs effectués par son équipe, avec équipe adverse et les points. |

### Base de données - Modèle

Voici le modèle de la base de données de mon projet.



### Base de données - Explications

Tout d’abord, la table « participants » contient les participants inscrits, qui se sont connectés par l’intranet du CPNV, en utilisant SAML. Il y a également une table users, qui contient des comptes locaux (admin, writer, …).

Chaque participant peut être inscrit à une ou plusieurs équipes (une équipe n’existe que dans un seul tournoi).

La table « teams » stock les informations relatives à chaque équipe. Il faut savoir qu’une équipe ne peut être que dans un seul tournoi. Puisqu’il y a possiblement plusieurs participants dans plusieurs équipes, il y a une table intermédiaire « participant\_team » entre les deux.

Chaque tournoi peut avoir aucune, une ou plusieurs équipes inscrites.

La table « tournaments » (tournoi) contient les infos principales des tournois, comme son nom et sa date de commencement et de fin. Chaque tournoi fait partie d’un évènement, qui contient simplement un nom est une image.

Chaque tournoi est également relié à un sport, qui lui est relié à un court (terrain).

Chaque tournoi est également relié à des pools, qui ont un mode spécifique (table poolModes, exemple : Matchs simples, Aller-retour, Elimination directe, etc.) et un type spécifique (table gametypes, qui ne contient qu’une description).

La table « contenders » est une table intermédiaire entre « team » et « pools ». Elle a également 2 relations avec la table « pools » ; Le champ pool\_id qui permet simplement de rattacher un pool à plusieurs « contenders », ainsi que le champ pool\_from\_id, qui permet de savoir de quel pool vient ce « contender », cette team en quelque sorte. C’est également grâce à ce champ que les tournois peuvent avoir des pools créés sans avoir d’équipe inscrites.

La table « games » contient les matchs qui ont été joués. Elle contient donc les points de chaque « contenders » d’un match.

La relation entre « contenders » et « games » est assez spéciale ; le champ unique de contenders possède deux clés étrangères dans la table « games » ; « contender1\_id » ainsi que « contender2\_id ». La table « games » possède également 2 champs pour le score, 1 pour chaque contender.

### Base de données - Modifications

La seule modification que j’ai effectuée sur la base de données a été de rajouter un champ « bestFinalRank » sur la table « pools ». Ce champ contient la place qu’aura l’équipe N°1, la gagnante, de cette pool. La place des autres équipes de la pool est également calculé avec ce champ, simplement en y incrémentant la valeur de la place de l’équipe dans la poule.

### 

### Particularité 1 : Routes

Une des particularités de Laravel sont les routes. Elles permettent de spécifier pour chaque chemin (url) qui peut y accéder (via des groupes/middlewares), avec quelle méthode (POST, GET, …), sur quelle vue (index, update, show, …) (optionnelle) et en appelant une fonction en particulier d’un contrôleur.

Grâce à ces routes, par défaut, il est impossible de « poster » des données sur les pages où cela n’est pas spécifié. Cela permet de bloquer certaines attaques, augmente la sécurité, et empêche le serveur de traiter des données qui ne devrait pas être envoyé.

Il est également possible d’ajouter un préfixe (ex. admin) à l’url. Il est possible de le spécifier dans un groupe, de sorte à ce que toutes les pages accessibles de ce groupe aient ce préfixe.

### Particularités 2 : Algorithme du classement général

# Réalisation

## Dossier de réalisation

## Description des tests effectués

## Erreurs restantes

## Liste des documents fournis

# Conclusions

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Sources – Bibliographie

## Journal de bord

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet