

TPI – Application Joutes sportives CPNV



Niels GERMANN

Chemin de Criblet 5
1867, Ollon VD
Niels.germann@cpnv.ch
SI-C4b

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Organisation	3
1.3	Objectifs et fonctionnalités	3
1.4	Méthode utilisée	4
1.5	Planification initiale	5
1.6	Analyse / Conception	6
1.7	Concept – Zoning	6
1.8	Stratégie de test	11
1.9	Risques techniques	11
1.10	Historique des modifications	11
1.11	Dossier de conception	12
1.11.1	Maquettes : Wireframe	12
1.11.2	Use Cases & Scénarios	18
1.11.3	Base de données - Modèle	21
1.11.4	Base de données - Explications	22
1.11.5	Base de données - Modifications	22
1.11.6	Particularité 1 : Routes	23
1.11.7	Particularités 2 : Algorithme du classement général	23
2	Réalisation	25
2.1	Dossier de réalisation	25
2.2	Description des tests effectués	26
2.3	Erreurs restantes	28
2.4	Liste des documents fournis	28
3	Historique	28
4	Conclusions	29
5	Annexes	29
5.1	Résumé du rapport du TPI / version succincte de la documentation	29
5.2	Sources – Bibliographie	29
5.3	Glossaire	29
5.4	Journal de bord	30
5.5	Manuel d'Installation	31
5.6	Manuel d'Utilisation	31
5.7	Archives du projet	31

1 Analyse préliminaire

1.1 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 fusionné 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.

1.2 Organisation

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

1.3 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.

1.4 Méthode utilisée

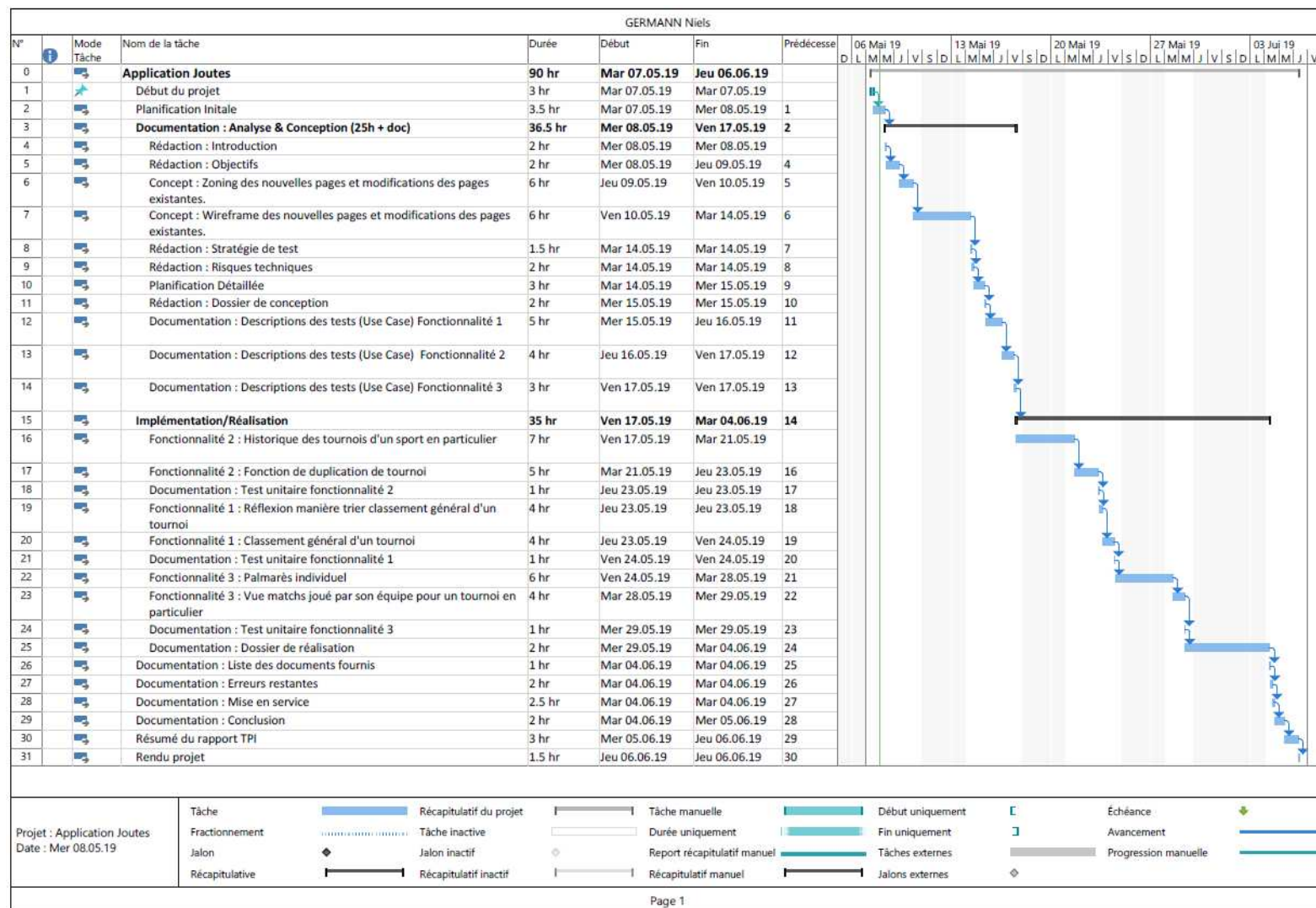
Pour ce qui est de ma planification détaillée, j'utilise Trello, un site web permettant la gestion de projet en ligne, en utilisant une méthode de travail en sprint (un sprint par semaine), comme discuté avec mon chef de projet.

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>)

1.5 Planification initiale



1.6 Analyse / Conception

1.7 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.

cpnv
joutes sportives

login gestionnaire login participant

Evenements

Tournois

Tournois par sport

Classement général par tournoi

Recherche...

Tableau classement général

footer

Fonctionnalité 1 : Page classement général - zoning

cpnv
joutes sportives

Evenements

Tournois

Tournois par sport

footer

login gestionnaire

login participant

Historique de tournois par sport

Recherche...

Liste déroulante sports

Valider

2019

Tournoi de Badminton

2018

Tournoi de Volley

Tournoi de Hockey

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

cpnv
jeux sportives

Evenements

Tournois

Tournois par sport

logout

login participant

Historique de tournois par sport

Liste déroulante sports

Recherche...

Valider

2019

Tournoi de Badminton

Dupliquer vers
(liste déroulante)

2018

Tournoi de Volley

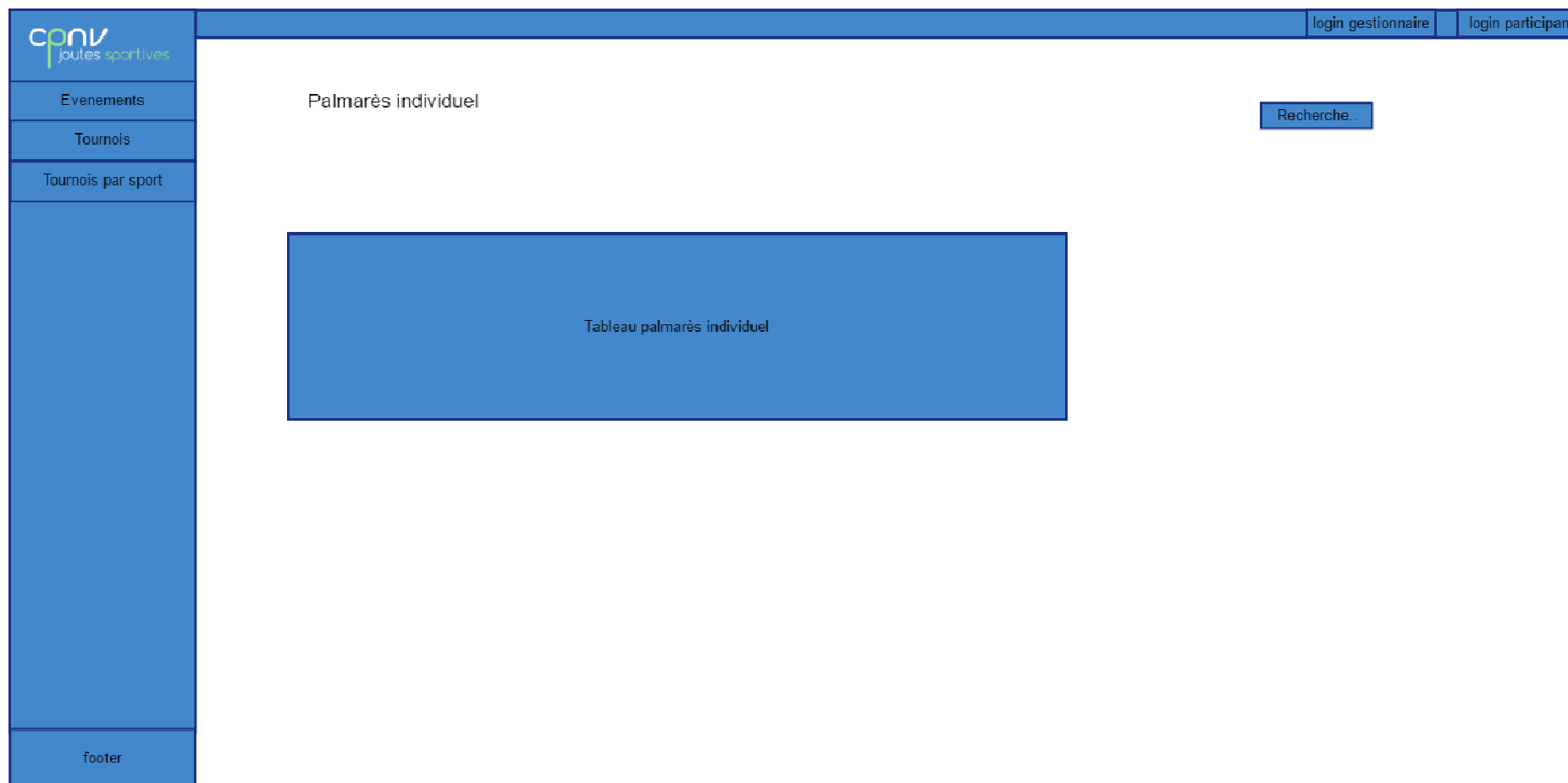
Dupliquer vers
(liste déroulante)

Tournoi de Hockey

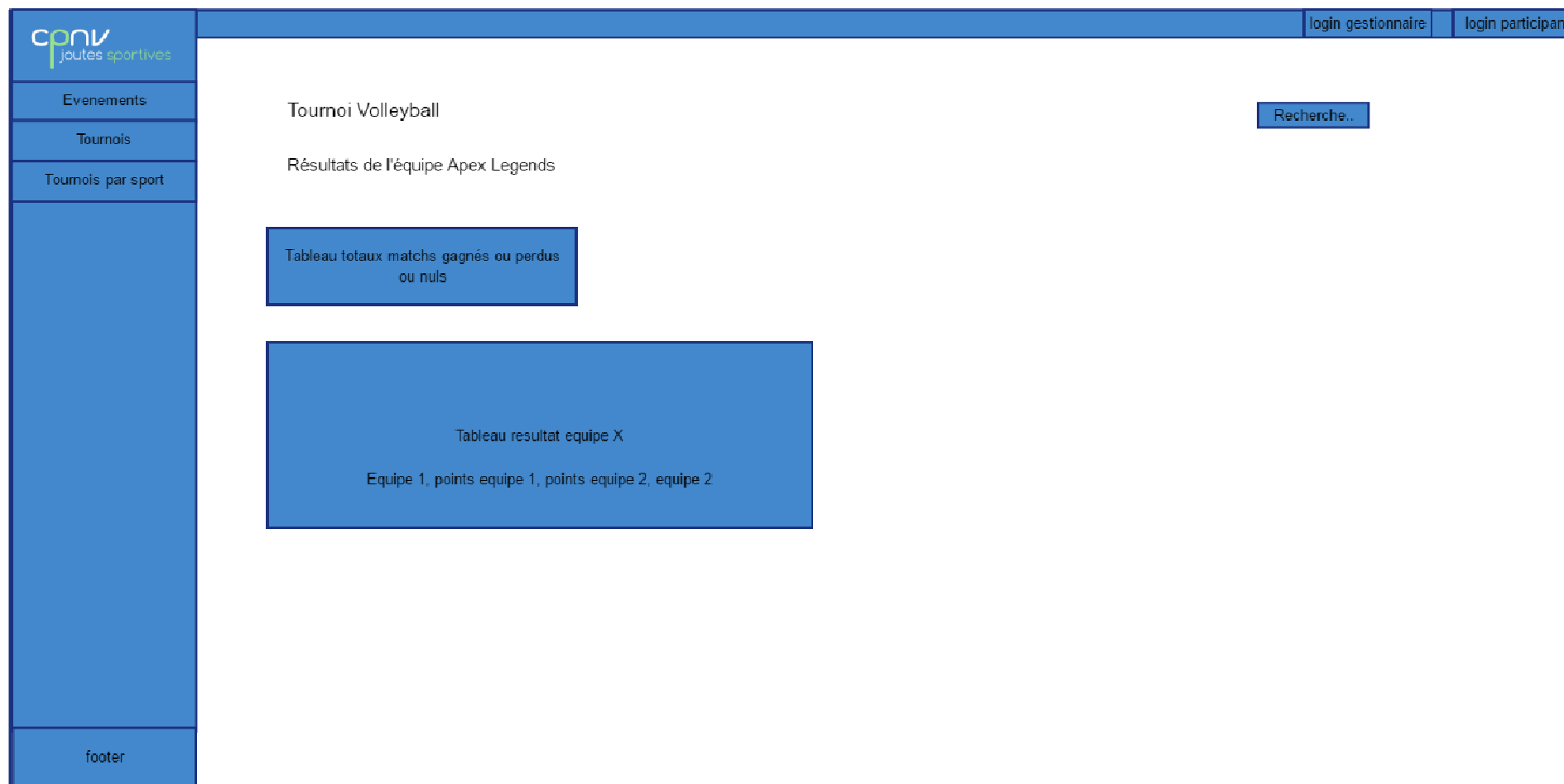
Dupliquer vers
(liste déroulante)

Footer

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

1.8 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.

1.9 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 utiliser 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 m'a 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 ont 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).

1.10 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.

1.11 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>)


1.11.1 Maquettes : Wireframe

Les zonings sont disponibles au point 2.1.


Fonctionnalité 1 : Page classement général - wireframe

	login gestionnaire	login participant																																			
Evenements	Classement général par tournoi																																				
Tournois	<input type="text" value="Recherche.."/>																																				
Tournois par sport																																					
	<table><thead><tr><th>#</th><th>Prénom Nom</th><th>Equipe</th><th>Points totaux</th><th>Matchs gagnés</th><th>Matchs perdus</th><th>Matchs nuls</th></tr></thead><tbody><tr><td>1</td><td>Niels Germann</td><td>Wakanda</td><td>8</td><td>4</td><td>1</td><td>0</td></tr><tr><td>2</td><td>Olivier Tissot</td><td>Wakanda</td><td>8</td><td>4</td><td>1</td><td>0</td></tr><tr><td>3</td><td>Kévin Pasteur</td><td>Apex Legends</td><td>6</td><td>3</td><td>2</td><td>0</td></tr><tr><td>4</td><td>Diego Sanchez</td><td>MC Steve</td><td>4</td><td>3</td><td>2</td><td>0</td></tr></tbody></table>		#	Prénom Nom	Equipe	Points totaux	Matchs gagnés	Matchs perdus	Matchs nuls	1	Niels Germann	Wakanda	8	4	1	0	2	Olivier Tissot	Wakanda	8	4	1	0	3	Kévin Pasteur	Apex Legends	6	3	2	0	4	Diego Sanchez	MC Steve	4	3	2	0
#	Prénom Nom	Equipe	Points totaux	Matchs gagnés	Matchs perdus	Matchs nuls																															
1	Niels Germann	Wakanda	8	4	1	0																															
2	Olivier Tissot	Wakanda	8	4	1	0																															
3	Kévin Pasteur	Apex Legends	6	3	2	0																															
4	Diego Sanchez	MC Steve	4	3	2	0																															
footer																																					

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

 Evenements Tournois Tournois par sport	Historique de tournois par sport		login gestionnaire	login participant
			Recherche...	
	Sport :	Badminton (liste déroulante)	Valider	
footer	2019			
	Tournoi de Badminton			
	Badminton 11.06.2019			
	2018			
	Tournoi de Volley			
	Volley 11.06.2019			
	Tournoi de Hockey			
	Volley 11.06.2019			

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

 Evenements Tournois Tournois par sport	logout			login participant		
	Historique de tournois par sport					
	Recherche...					
	Sport :		Badminton (liste déroulante)		Valider	
	2019					
	Tournoi de Badminton 2018					
	Badminton		Dupliquer vers			
	11.06.2019		Badminton 2019		v	
	2018					
	Tournoi de Volley					
	Volley		Dupliquer vers			
	11.06.2019		Volley 2020		v	
	Tournoi de Hockey					
	Hockey		Dupliquer vers			
11.06.2019		Hockey 2020		v		
footer						

Fonctionnalité 3 : Page palmarès individuel – wireframe

#	Tournois	Equipe	Points totaux	Matchs gagnés	Matchs perdus	Matchs nuls	Classement
1	Badminton	Wakanda	8	4	1	0	4
2	Volleyball	Apex Legends	8	4	1	0	3
3	Basketball	YesMen	6	3	2	0	9
4	Unihockey	MC Steve	4	3	2	0	12

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

<div> <div>cpnv</div> <div>joutes sportives</div> </div>		login gestionnaire	login participant																									
Evenements																												
Tournois																												
Tournois par sport																												
	<div> <div>Tournoi Volleyball</div> <div>Résultats de l'équipe Apex Legends</div> <div> <div> <div>Matches gagnés</div> <div>2</div> </div> <div> <div>Matches perdus</div> <div>2</div> </div> <div> <div>Matches nuls</div> <div>0</div> </div> </div> <div> <table> <tr> <th>Equipe 1</th><th>Points</th><th>-</th><th>Points</th><th>Equipe 1</th></tr> <tr> <td>Apex Legends</td><td>9</td><td>-</td><td>12</td><td>DeamTeam</td></tr> <tr> <td>Apex Legends</td><td>26</td><td>-</td><td>10</td><td>Band of the hawk</td></tr> <tr> <td>Apex Legends</td><td>21</td><td>-</td><td>9</td><td>dark brotherhood</td></tr> <tr> <td>Apex Legends</td><td>13</td><td>-</td><td>14</td><td>Wakanda</td></tr> </table> </div> </div>			Equipe 1	Points	-	Points	Equipe 1	Apex Legends	9	-	12	DeamTeam	Apex Legends	26	-	10	Band of the hawk	Apex Legends	21	-	9	dark brotherhood	Apex Legends	13	-	14	Wakanda
Equipe 1	Points	-	Points	Equipe 1																								
Apex Legends	9	-	12	DeamTeam																								
Apex Legends	26	-	10	Band of the hawk																								
Apex Legends	21	-	9	dark brotherhood																								
Apex Legends	13	-	14	Wakanda																								
footer																												

1.11.2 Use Cases & Scénarios

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	Conditions particulières	Réponse du site
L'utilisateur est connecté. Il clique sur le bouton « Classement général » dans le menu latéral.		Il est redirigé sur la page d'historique de tournois par sport
Il choisit 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	Si le tournoi n'est pas fini, il y aura un simple message d'info sur la page, sans tableau.	Il est redirigé vers la page du classement général

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	Conditions particulières	Réponse du site
L'utilisateur est connecté. Il clique sur le bouton « Classement général » dans le menu latéral à gauche.		Il est redirigé sur la page d'historique de tournois par sport
Il choisit 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)

Use Case fonctionnalité 2 - Admin

En tant qu'Administrateur	Je veux dupliquer la formule d'un tournoi	Afin de gagner du temps lors de la création de tournoi et d'événements.
----------------------------------	--------------------------------------------------	--------------------------------------------------------------------------------

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

Action utilisateur	Conditions particulières	Réponse du site
L'administrateur est connecté. Il clique sur le bouton « Classement général » dans le menu latéral.		Il est redirigé sur la page d'historique de tournois par sport
Il choisit 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 choisit un tournoi à dupliquer, badminton, et parcourt la liste déroulante dans le coin inférieur droit du tournoi. Il sélectionne son tournoi sur lequel dupliquer celui de Badminton. Il choisit le tournoi de badminton 2020.		Une notification l'avertit que les paramètres de ce tournoi ont été dupliqués dans celui sélectionné dans la liste déroulante.

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

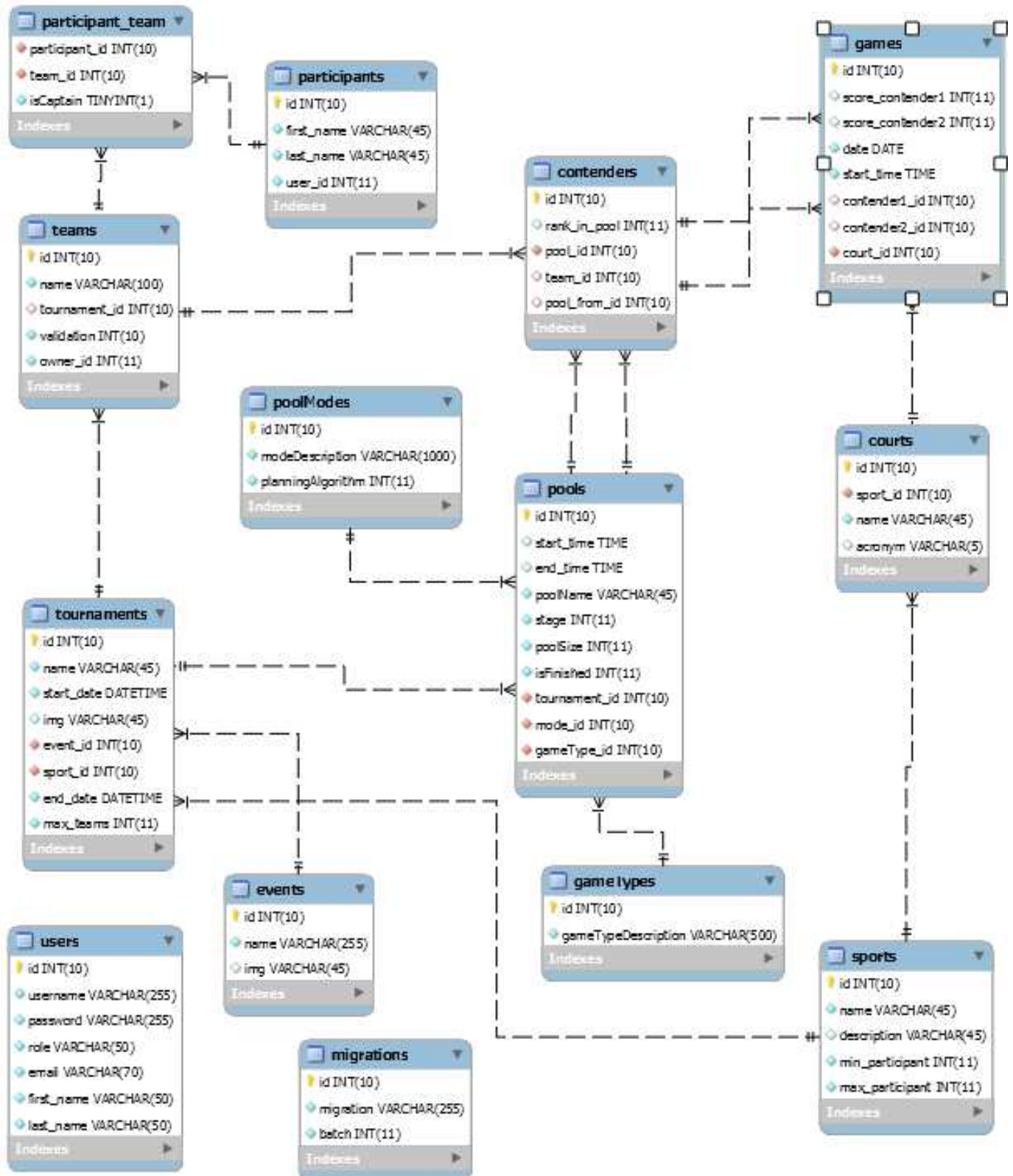
Action utilisateur	Conditions particulières	Réponse du site
L'utilisateur est connecté. Il clique sur le bouton « Palmarès individuel » dans le menu latéral.		Il est redirigé sur son palmarès individuel. Un tableau avec les points marqués par les équipes dans lequel il a été inscrit (et les tournois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

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

Action utilisateur	Conditions particulières	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.

1.11.3 Base de données - Modèle

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



1.11.4 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 et 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 : Matches simples, Aller-retour, Elimination directe, etc.) et un type spécifique (table gametypes, qui ne contient qu'une description).

Une image est disponible au point 1.11.7, ou également en annexe, qui montre les pools d'un tournoi de badminton, avec leurs équipes et leur relation avec un pool de l'étape d'après (permettant d'établir un classement général).

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 matches 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.

1.11.5 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 du pool est également calculée avec ce champ, simplement en y incrémentant la valeur de la place de l'équipe dans la poule, multipliée par le nombre de pools dans l'étape en cours.

1.11.6 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.

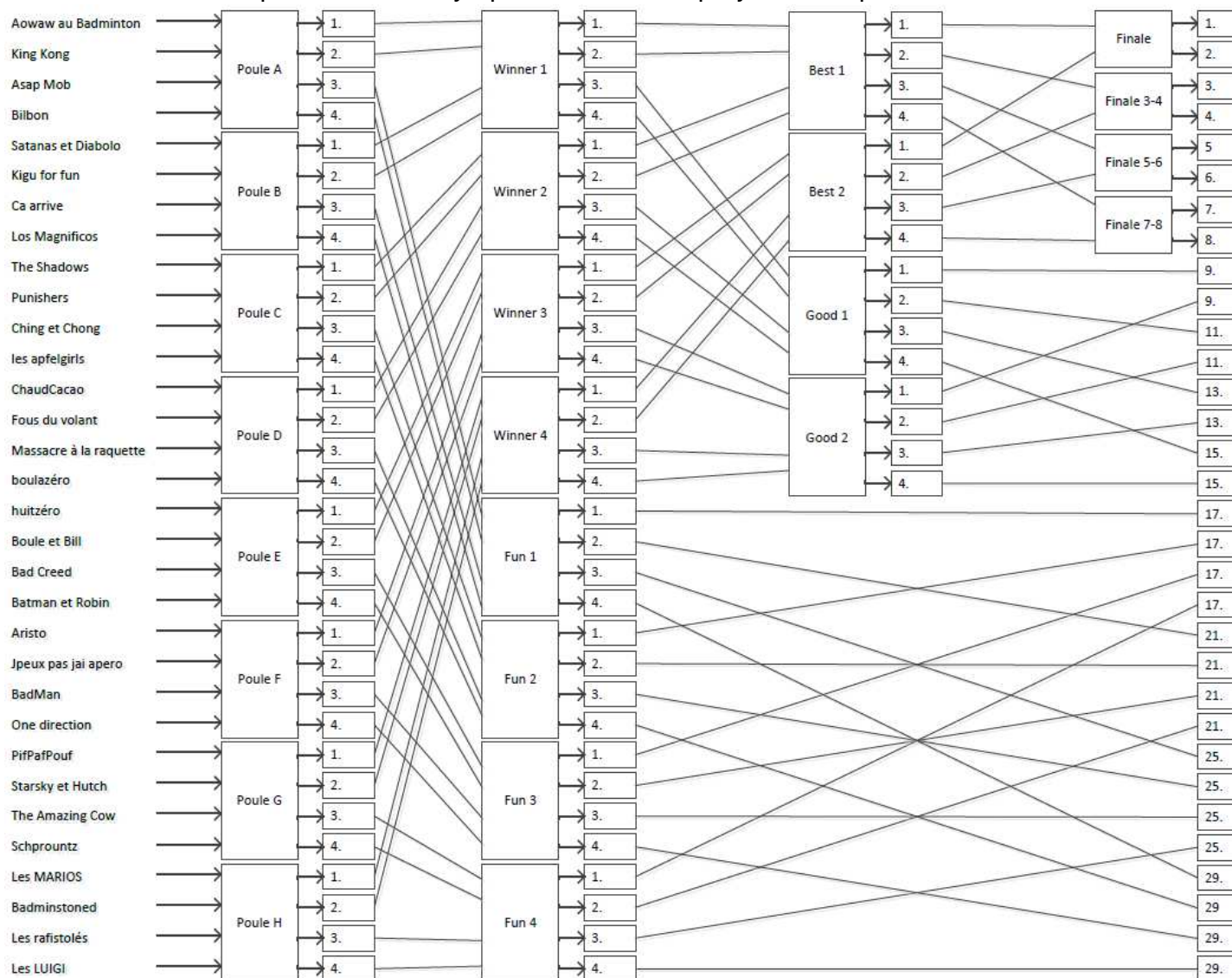
J'utilise la ligne « route ::resource », comme utilisé pour les autres pages, qui permet d'autoriser l'accès à plusieurs vue, avec plusieurs méthodes.

1.11.7 Particularités 2 : Algorithme du classement général

Pour créer le classement général d'un tournoi, je commence par récupérer toutes les pools du tournoi (grâce à la clé étrangère tournament_id) qui ont leur champ « bestFinalRank » non null. Ce champ que j'ai créé permet de savoir qu'elle place aura la meilleure équipe de la pool. Du coup, ce champ est uniquement spécifié dans les pools qui n'ont pas de suivantes (Par exemple pour un tournoi de badminton ; Les perdants du deuxième tour (les pools « Fun »), les perdants du troisième tour (les pools « Best »), et les pools de finales.). La place des équipes suivantes de chaque pool est simplement la valeur de « bestFinalRank » + leur position dans la poule multipliée par le nombre de pool dans l'étape actuelle. Le classement général est uniquement disponible lorsque les pools d'un tournoi sont toutes finies.

J'utilise finalement la fonction array_multisort pour trier le tableau par rang. Je ne peux pas le faire lors de ma requête, car le champ « bestFinalRank » spécifie uniquement le rang du premier de la poule, et les autres sont rang sont insérés par moi par la suite. J'avais commencé par également utilisé la fonction « rankings », disponible dans le modèle « pools » pour avoir le total des matchs gagnés/perdus/nuls ainsi que le score. Malheureusement, cette fonction ne prend qu'un seul enregistrement (pool), et donc ne me donne pas le score total, mais uniquement du dernier pool joué. S'il me reste assez de temps, j'essaierai de rajouter une fonction permettant de faire le total des scores pour toutes les pools joués dans un tournoi par l'équipe.

Je me suis basé sur ce document, qui m'a été envoyé par mon chef de projet, mais qui vient de mon client, M. Carrel.



2 Réalisation

2.1 Dossier de réalisation

Pour mener mon projet à bien, j'ai utilisé Git et Github, qui me permettent de versionner mon projet et mon code.

J'ai gardé la même architecture que celle du projet de base, fait par les techniciens ES du CPNV.

Le dossier « docs/2019 – TPI Niels » contient les différents documents de mon TPI ; Ma documentation, mon journal de travail, mes maquettes et quelques annexes.

Les autres dossiers existaient déjà dans le projet de base.

Le dossier « database » contient des seeders, qui permettent de générer des données pour la base de données, en utilisant php artisan (outil en ligne de commande de Laravel).

Ce dossier contient également des fichiers .sql qui sont utilisés par ces seeder, et qui peuvent, pour certains être exécuté tel quel sur la base de données.

Le dossier public contient les feuilles de style CSS, des scripts Javascript ainsi que des images, uploadées lors de la création de tournoi par exemple.

Le dossier ressources contient quant à lui les vues « Blade » ainsi que scripts Javascript.

Le dossier « App/http/Controllers » contient les différents contrôleurs du projet.

Les modèles Eloquent se trouvent dans le dossier « App »

Le fichier permettant de configurer les routes se trouve dans « routes/web.php »

Les autres dossiers sont pour la plupart des dossiers utilisés par Laravel lui-même, et n'ont pas été modifiés par moi.

Logiciels que j'ai utilisés pour ce projet :

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

2.2 Description des tests effectués

Scénario fonctionnalité 1 : Classement général d'un tournoi

Action utilisateur	Conditions particulières	Réponse du site	Test
L'utilisateur est connecté. Il clique sur le bouton « Classement général » dans le menu latéral.		Il est redirigé sur la page d'historique de tournois par sport, qui permet d'accéder au classement d'un tournoi	OK
Il choisit un sport, badminton, dans la liste déroulante et appuie sur valider		La liste des tournois de badminton s'affichent en dessous, triés par année.	OK
Il clique sur un tournoi, badminton de 2019 par ex	Si le tournoi n'est pas fini, il y aura un simple message d'info sur la page, sans tableau.	Il est redirigé vers la page du classement général	OK, conditions particulières aussi

Scénario fonctionnalité 2 : Page d'historique de tournois par sport

Action utilisateur	Réponse du site	Test
L'utilisateur est connecté. Il clique sur le bouton « Classement général » dans le menu latéral à gauche.	Il est redirigé sur la page d'historique de tournois par sport	OK
Il choisit 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)	OK

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

Action utilisateur	Réponse du site	Test
L'administrateur est connecté. Il clique sur le bouton « Classement général » dans le menu latéral.	Il est redirigé sur la page d'historique de tournois par sport	OK
Il choisit un sport, badminton, dans la liste déroulante et appuie sur valider	La liste des tournois de badminton s'affiche en dessous, triés par année.	OK
Il choisit un tournoi à dupliquer, badminton 2019 par ex, et parcourt la liste déroulante dans le coin inférieur droit du tournoi. Il sélectionne un sport ainsi qu'un événement sur lequel dupliquer ce tournoi.	Une notification l'avertit que les paramètres de ce tournoi (ainsi que les pools) ont été dupliqués dans un nouveau créé dans l'événement choisi, dans le sport choisi.	Erreurs. Fonctionnalité par terminée. Vue en revanche finie.

Fonctionnalité 3 : Page de palmarès individuel

Action utilisateur	Réponse du site	Test
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.	OK, erreurs restantes (login participant hardcodé)

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

Action utilisateur	Réponse du site	Test
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.	OK, erreurs restantes (login participant hardcodé)
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.	Redirection ne marche pas. Page équipe non terminée.

2.3 Erreurs restantes

Pour la fonctionnalité 2, je n'ai malheureusement pas eu le temps de faire la fonction qui permet de dupliquer un tournoi. En revanche, la vue, ainsi que les valeurs dans le formulaire de duplication, sont bonnes. La redirection ainsi que la requête pour dupliquer le tournoi ne sont eux pas implémenté.

Pour la fonctionnalité 3, je n'ai malheureusement pas eu le temps de faire en sorte que la connexion se fasse, via SAML ou via un login simple créé pour l'occasion. Lors de l'affichage de la page, le Palmarès individuel affiche celui du participant ayant l'id n°1. Cette valeur est hardcodée, dans le contrôleur de la page. La configuration de la route pour cette page est également fausse ; il est possible d'accéder à cette page en étant un visiteur, alors que cela ne devrait pas être possible.

Pour la fonctionnalité 3, il y a également la vue permettant de voir le palmarès individuel **par équipe** qui n'est pas disponible. Il y a une erreur lors de la redirection. La vue ainsi que la requête ne sont également pas implémentée.

2.4 Liste des documents fournis

- Documentation (Rapport TPI) (ce document)
- Journal de travail (ci-dessous et en annexe)
- Maquettes du site
- Planification initiale
- Code (Github et sur CD)

3 Historique

Semaine	Éléments modifiés
2	Modifications de mes maquettes, car j'avais mal compris les données que je devais afficher pour mon classement général. Je l'ai remarqué avec mon chef de projet et mon client, M. Carrel.
4	J'ai remarqué des erreurs dans les « seeders » fourni avec le projet des Joutes ; Pour le tournoi de badminton, en finale, une équipe s'affronte elle-même. Les finalistes viennent des deux pools « good » au lieu des pools « best »

4 Conclusions

Tout d'abord, ce projet m'a beaucoup aidé dans la compréhension de la logique de l'orienté-objet, et sur la structure du code de Laravel. Je me suis également amélioré dans la compréhension de base de données relationnels complexes, grâce à la base de données du projet, qui compte 15 tables. Je me suis également amélioré sur la gestion de projet, en particulier sur Trello et mon journal de travail

J'ai eu plusieurs difficultés lors du déroulement de mon TPI, en particulier sur la méthode permettant de faire un classement général optimal. J'ai eu plusieurs difficultés avec Eloquent, qui permet de faire des requêtes SQL en orienté-objet.

J'ai également perdu du temps sur l'algorithme de classement général car ma base de données n'était pas optimale pour faire mes tests ; Les seeders fournis ne créent pas de tournoi finis, et il y avait des erreurs dans les pools de finales ainsi que les pools « good ». Grâce à M. Carrel, j'ai mieux compris la logique derrière l'algorithme de classement général que je devais mettre en place, ainsi que sur les seeders.

J'ai également eu des problèmes avec les formulaires Laravel, pour qui leur bouton d'envoi ne fonctionne pas dans certains cas.

Finalement, ce projet aura été stressant mais très intéressant. En fin de 2^{ème} année, je trouvais l'orienté-objet compliqué pour pas grand-chose, et peu efficace. Maintenant, avec plus de recul, je trouve l'orienté-objet bien mieux pour de grands projets, bien que plus difficile à appréhender.

5 Annexes

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

5.2 Sources – Bibliographie

<https://laravel.com/docs/5.8>

<https://github.com/CPNV-ES/Joutes>

<https://github.com/CPNV-ES/Joutes/wiki>

<https://openclassrooms.com/fr/courses/3613341-decouvrez-le-framework-php-laravel>

<https://openclassrooms.com/forum/categorie/php>

<https://stackoverflow.com/>

<https://www.w3schools.com/>

5.3 Glossaire

Terme	Définition
Laravel	Framework PHP open-source utilisant une structure MVC. Il est en parti basé sur le framework Symfony
Laravel Homestead	Box vagrant permettant de mettre en place rapidement et « simplement » un environnement de développement Laravel virtuel. Utilise Ubuntu, MySQL, nginx, ...
Seeder	Fichier PHP, utilisé par Laravel, permettant de générer simplement et rapidement des données pour une base de

	données. On peut faire appel à des fichiers SQL et faire des calculs en PHP, par exemple.
SAML	Permet d'échanger des données sensibles entre serveurs. Utilisé ici pour se connecter sur l'application des Joutes avec un login de l'intranet du CPNV, sans que le projet aie accès à la base de données directement.
Blade	C'est un « moteur de template » qui permet de facilement mettre en place un système de vue, avec layout/footer/menu latéral séparés.
Eloquent	C'est un mapping objet-relationnel (ORM). Il fait l'interface entre la base de donnée et le code en lui-même. Il permet de faire des requêtes sur la base de données, en orienté-objet.
QueryBuilder	C'est également un ORM comme Eloquent, sauf que lui n'est pas orienté-objet.
Vagrant	Logiciel utilisant un hyperviseur (ex. VMWare ou VirtualBox) qui permet de créer des boxs. Ces boxs permettent de transférer facilement des machines virtuelles avec des configurations spécifiques. Il peut être considéré comme un « wrapper » (patron de conception)
MySQL	C'est un système de gestion de bases de données relationnelles. Les données de l'application sont stockées sur un serveur MySQL, dans ma VM Laravel Homestead

5.4 Journal de bord

Disponible en annexe

5.5 Manuel d'Installation

Tout d'abord, pour mettre en place l'environnement de développement, il faut commencer par suivre la marche à suivre disponible sur le Github du projet.

<https://github.com/CPNV-ES/Joutes/blob/master/readme.md>

J'utilise personnellement Laravel Homestead, machine virtuelle qui permet d'après mon expérience de mettre en place plus facilement l'environnement de développement.

<https://laravel.com/docs/5.8/homestead>

Après avoir généré la base de données avec les commandes « php artisan migrate » et après avoir rempli les tables avec un seeder, comme décrit dans la marche à suivre, (ex : php artisan db :seed --class= BadmintonTournamentSeeder) il faut exécuter sur la base de données mon script, disponible en annexe et dans « /database/sqlFiles/tpi_niels_script_modif_bdd », qui rajoute simplement un champ « bestFinalRank » sur chaque pool, qui peut être null.

Ensuite, le script ajoute également des valeurs à ce bestFinalRank. Ces valeurs dépendent de la formule du tournoi. Les valeurs que j'ai rentrées correspondent aux valeurs pour un tournoi de badminton « simple », comme dans le seeder « BadmintonTournamentSeeder ».

Si vous utilisez un autre type de tournoi, il faudra alors modifier ce script pour rajouter une valeur à « bestFinalRank » aux pools « finales » sans suite. La valeur dépendra du nombre d'étape et de pool, et du type de tournoi (élimination, matchs simples, etc.). Le plus simple reste de s'inspirer du graphique plus-haut montrant les pools d'un tournoi de badminton.

N'oubliez pas de configurer le fichier Homestead.yaml si vous utilisez Laravel Homestead. C'est dans ce fichier que vous devez rentrer le chemin du dossier contenant le code du dépôt, ainsi que le chemin d'une clé privée RSA.

Vous pouvez utiliser PuttyGEN pour créer votre clé RSA, mais copier votre clé en version OpenSSH, au lieu de l'enregistrer avec le bouton. N'oubliez pas de choisir le nom de votre base de données dans le fichier homestead.yaml.

Ensuite, vous pouvez accéder au site en allant sur un navigateur sur l'adresse IP de la machine virtuelle, ou en utilisant une url, définie dans votre fichiers hosts, comme dit dans la marche à suivre de Laravel Homestead.

5.6 Manuel d'Utilisation

Disponible en annexe

5.7 Archives du projet

Mon projet est disponible sur Github, à l'adresse :

<https://github.com/CPNV-ES/Joutes/tree/Niels-TPI>