



Dossier de projet professionnel

Titre professionnel développeur web
et web mobile (DWWM)

réalisé par
Sofiane MSATFA

2021-2022

Table des matières

1	Introduction	3
1.1	Cahier des charges	4
1.1.1	Objectifs	4
1.1.2	Contraintes techniques	5
2	Mise en place du projet	6
3	Récupération des données	8
3.1	Configuration	9
	Bibliographie	10

1 Introduction

Dans le cadre de ma formation développeur web et web mobile, j'ai eu la chance d'assister Mr. Vincent VAURETTE qui est en charge de la cellule de transition numérique de la Fédération Française de Volley-Ball (FFVB).

Le but de son travail est de développer des outils qui facilitent la gestion de tout ce qui est en lien avec le monde du volley-ball, et de redonner un coup de jeune à l'intranet de la fédération.

Durant trois mois, j'ai eu la responsabilité de la transition numérique de l'interface de désignation des arbitres.



La Fédération Française de Volley-Ball est une association créée en 1936 qui constitue l'instance dirigeante du volley-ball en France. Plusieurs centaines de matchs se déroulent par week-end, et la désignation des arbitres les concernant est entrêmement chronophage.

Le site actuel de la fédération date de 2009 et manque d'ergonomie. Celui-ci demande plusieurs click pour effectuer une seule désignation et ne permet pas d'afficher toutes les informations nécessaires pour celles-ci. Les gestionnaires sont alors contraints de passer par un tableur externe pour procéder aux désignations, puis ensuite de rentrer manuellement celles-ci pour chaque match.

L'objectif principal qui m'a été confié a été la création d'un outil permettant d'automatiser ces désignations. Celui-ci devait pouvoir associer un ou plusieurs arbitres habilités et disponibles à chaque match d'une période donnée, tout en laissant le choix de leur modification.

1.1 Cahier des charges

1.1.1 Objectifs

Quels sont les objectifs demandés ?

- L'objectif principal est l'automatisation des désignations des arbitres pour une période donnée. Il faut que cet algorithme puisse proposer un arbitre habilité et disponible pour chaque match de la période voulue.
- Un autre objectif essentiel est la mise en place d'une matrice des distances entre chaque arbitre et chaque club, qui pourra servir à faciliter la désignation des arbitres pour les gestionnaires.
- Enfin, il faut également améliorer l'interface graphique et ajouter quelques fonctionnalités.

Quelles sont les fonctionnalités à ajouter à l'interface ?

- Il faut que l'interface affiche tous les matchs à venir de la saison sous forme de tableau, qu'on pourra également trier par poule si voulu. Pour le confort des utilisateurs, Le tri par poule doit se faire sous forme de SELECT avec la possibilité de rechercher une valeur dans celui-ci.
- Le tri des matchs par colonne doit être possible et une pagination doit être mise en place.
- (Optionnel) Une barre de recherche doit également être disponible pour trouver une donnée précise dans ce tableau.
- Il faut pouvoir désigner les arbitres directement depuis cette interface, en affichant un SELECT avec une liste de tous les arbitres disponibles. Cette liste d'arbitres doit afficher plusieurs informations essentielles les concernant : le numéro de licence, le niveau, le grade, et la distance les séparant du gymnase du match. Les arbitres doivent également être triés par état de disponibilité pour le match sélectionné, avec un code couleur différent pour chacun.
- La désignation doit se faire de façon automatique au changement du SELECT pour éviter de perdre les sélections en cas d'oubli.
- L'interface doit également permettre de lancer l'algorithme d'automatisation des désignations en permettant de choisir une période voulue et de lancer les désignations via un bouton.

Précisions concernant l'automatisation des désignations ?

- Il faut pouvoir distinguer les matchs pour lesquelles deux arbitres sont nécessaires, et proposer le bon nombre d'arbitres à chaque fois.
- Il ne faut proposer que des arbitres qui sont habilités à arbitrer, et disponibles le jour du match : c'est à dire qui n'ont pas précisé d'indisponibilité, et qui ne sont pas déjà désignés sur un autre match.
- Prendre en compte les exceptions : les arbitres ne peuvent pas arbitrer un match joué par leur club, et les matchs du club à qui ils ont donné leurs points d'arbitrage.
- La distance séparant les arbitres du gymnase doit aussi être prise en compte, mais être flexible pour éviter les désignations répétitives des mêmes arbitres pour les mêmes gymnases.
- L'algorithme doit proposer des arbitres pour chaque match de la période, mais également permettre le changement manuel de ceux-ci directement sur l'interface. De plus, la validation des désignations automatiques, contrairement aux désignations manuelles, devra se faire grâce à un bouton de confirmation.
- Certains matchs sont de type 'tournois' et se déroulent donc au même endroit : l'algorithme doit proposer le même arbitre pour chaque paquet de trois matchs de la même poule du même tournois.

1.1.2 Contraintes techniques

Les technologies utilisées sont :

- Bootstrap 4.6 (une autre version peut cependant être utilisée)
- Javascript / JQuery 3.6
- PHP 8 (aucun framework)
- MySQL 8.0

2 Mise en place du projet

Afin de répondre à ces objectifs, il m'a fallu mettre en place un plan d'action en prenant compte les technologies utilisées pour ce projet.

Mon travail devait pouvoir être maintenu facilement par mon tuteur, et pour ce faire la documentation et l'organisation des fichiers étaient primordiales.

Une documentation complète a été réalisée en fin de projet pour apporter une précision supplémentaire aux commentaires écrits avec le standard d'écriture PHP Docs, permettant ainsi de faciliter la compréhension et la reprise du projet par mon tuteur.

J'ai décidé de travailler en Programmation Orientée Objet (POO) pour faciliter la réutilisation des composants.

J'ai donc pensé à organiser mes fichiers par entité qui compose ce projet : chaque classe concernant une entité se trouve dans un dossier différent des autres et permet ainsi de facilement s'y retrouver dans les fichiers.

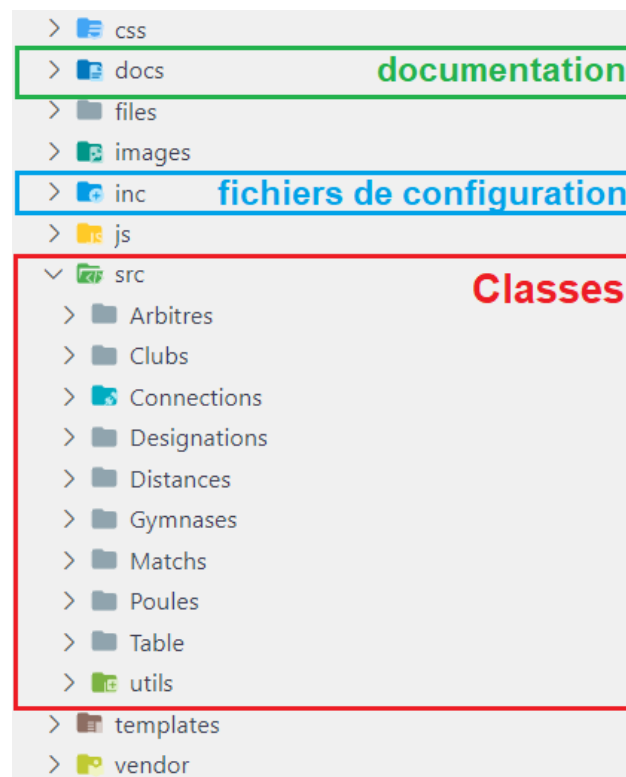


FIGURE 1 – Arborescence des dossiers du projet

Pour effectuer cette séparation des fichiers sans compliquer l'utilisation des classes, j'ai adopté le concept de namespace et utilisé le gestionnaire de dépendances Composer pour ce projet.

Composer est un gestionnaire de dépendances pour PHP. Il permet de réduire l'importation de fichiers en automatisant cette tâche.

➔ Je commencerai par expliquer les méthodes utilisées pour récupérer les données nécessaires à ce projet, puis j'exposerai les modifications apportées à l'interface graphique et enfin je parlerai en profondeur de l'algorithme d'automatisation des désignations.

3 Récupération des données

Par manque d'accès à la base de données fédérale, la première étape consistait à récupérer les données nécessaires à la mise en place des outils.

Après une analyse complète des informations nécessaires au projet, j'en suis arrivé au modèle de données UML suivant :

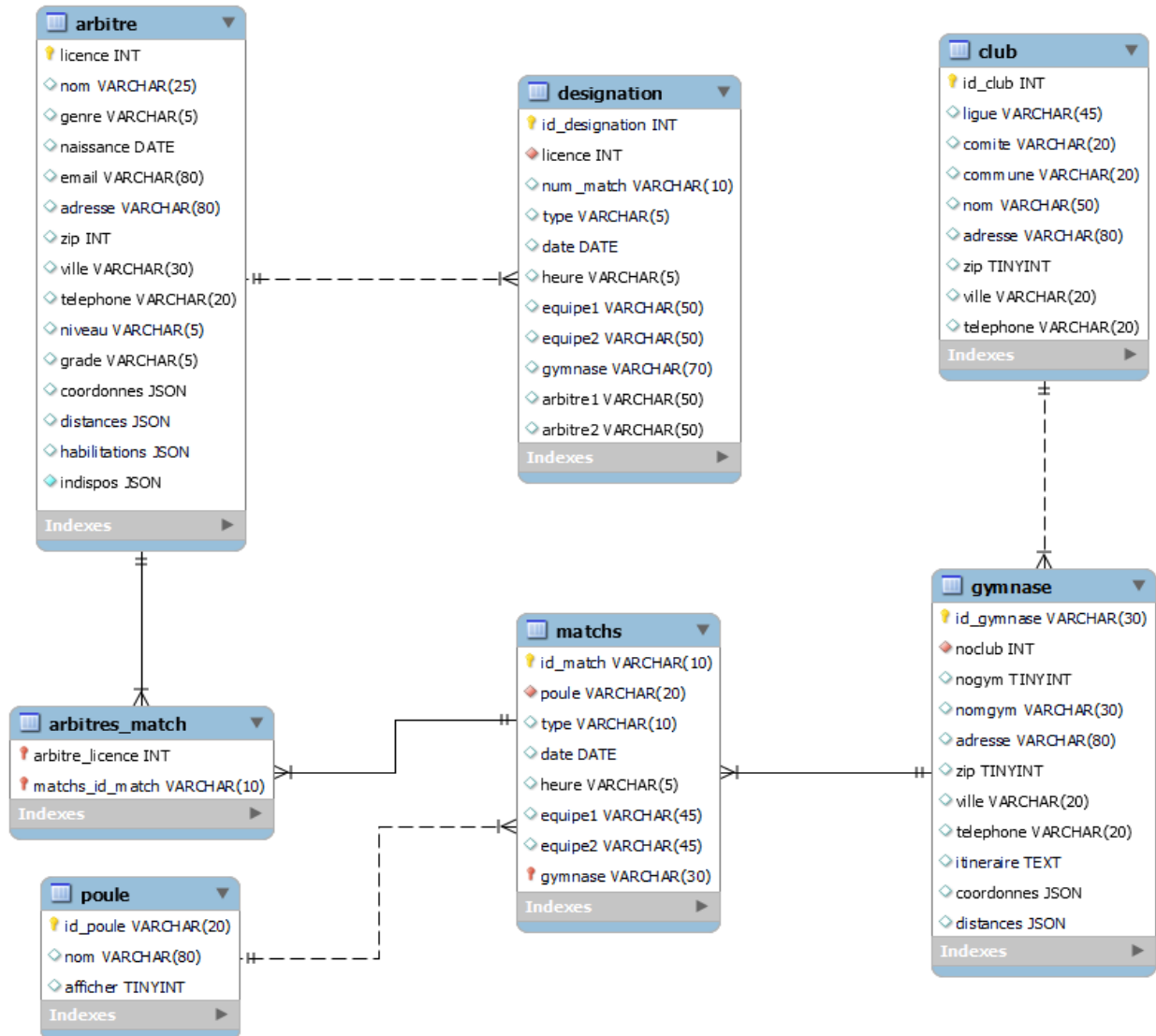


FIGURE 2 – Modèle de données UML effectué avec MySQL Workbench

Certaines données étaient disponibles au format csv (Excel), et d'autres demandaient à être récupérées sur l'intranet actuel de la fédération. Il m'a donc fallu extraire ces dernières via Web Scrapping pour ensuite les insérer dans ma base de données.

3.1 Configuration

Dans un premier temps, afin de faciliter la configuration de l'accès à la base de données lors de la mise en production, j'ai mis en place un fichier de configuration général `globals.php`. Ce fichier contenait un panel de variables de configuration différentes en fonction de l'environnement (développement ou production).

Bibliographie