**Projet Ranlod-Gorras**

**Introduction**

Le projet Ranlod-Gorras consiste à développer un site web pour la gestion d’un tournoi de tennis, celui-ci au travers de nouvelle technologies tel que l’html5, le CSS3 et de l’AJAX.

Il y a deux parties très distincte dans le site, une partie cliente et une administratrice ;

- le client pourra voir la liste des joueurs disponible, il pourra aussi s’abonner à des parties en cours et voir le score en direct, mais aussi voir un classement des joueurs dans le tableau du tournoi, une fois le match terminé le score des sets des deux joueurs sera affiché sur le tableau du tournoi.

-Tandis que dans la partie administratrice, l’administrateur devra se loguer pour accéder aux commandes du site web (non accessibles aux clients), il pourra ajouter et supprimer des joueurs, créer les parties du premier tour avec les 8 joueurs qu’il aura choisis, les tours suivants se créeront automatiquement. Pour le deuxième ou troisième tour du tournoi, l’administrateur ne pourra que modifier le terrain sur lequel les joueurs de tennis vont se disputer un match !

Nous avons essayé de faire le site web tant que pour le client que pour l’administrateur de manière la plus ergonomique possible et avec un style relativement plaisant.

**Les technologies utilisées**

Au niveau du back-end nous avons utilisé le langage PHP, un serveur MySQL avec PhpMyAdmin. Dans le front-end comme demandé nous avons utilisé de l’HTML5, CSS3 et de l’Ajax. Mais aussi du jQuery pour certain affichage et le Framework Bootstrap pour avoir un web design responsive, j’entends par cela qu’il soit compatible avec la plus part des gammes d’appareil clientes tel que les moniteurs d’ordinateur, les Smartphones ou encore les tablettes.

L’utilisation de bootstrap permet aussi d’avoir une interface graphique plus simple à modifier, mais propose aussi un large choix d’option de style pour développer son site.

Dans l’énoncé il nous était demandé d’utiliser de l’Ajax pour réactualiser le score des parties dans lesquels l’utilisateur s’était abonné, mais nous avons voulu pousser plus loin l’utilisation avec le jQuery pour l’affichage de certaines informations.

Comme c’était la première fois pour nous de faire un développement web, il nous a fallu un petit temps avant de comprendre le fonctionnement du jQuery avec ses spécificités propre à lui-même. Après quelque essais-erreurs petit à petit nous avons remarqué qu’utiliser le jQuery est relativement puissant, comme par exemple créer des objets json pour les envoyer d’un endroit à l’autre.

**Partie cliente**

**Le tournoi :**

Le client pourra s’il le désire afficher le tournois, il y a plusieurs informations nous avons les quarts de finale, la demi finale, la finale et le gagnant. Le nom de chaque joueur de tennis inscrit à une partie sera affiché sur le tableau, pendant ce temps il y aura le déroulement de la partie, le score des sets sera affiché seulement si la partie est terminée, sinon il n’y a que le nom des joueurs.

Par exemple ici lors du troisième tour avec Djokovic contre Nishikori, cette partie sera accessible dans le live, elle sera affiché dans le tableau du tournoi pendant que la partie est en cours, mais les points des sets seront affiché que lorsque la partie sera terminée.



**La liste des joueurs**

Le client aura accès à la liste des joueurs inscrit dans le tournoi, celle-ci répertorie leurs noms et leurs nationalités.



**Les matches en direct « Live »**

Le joueur aura le choix de s’abonner à plusieurs parties en live, chaque X secondes, le score et les sets de chaque partie se mettent à jour automatiquement. Ainsi lorsque l’administrateur mettra à jour les points cela sera automatiquement répercuté sur le choix des parties aux quel chaque client c’est abonné.



**Partie d’administration**

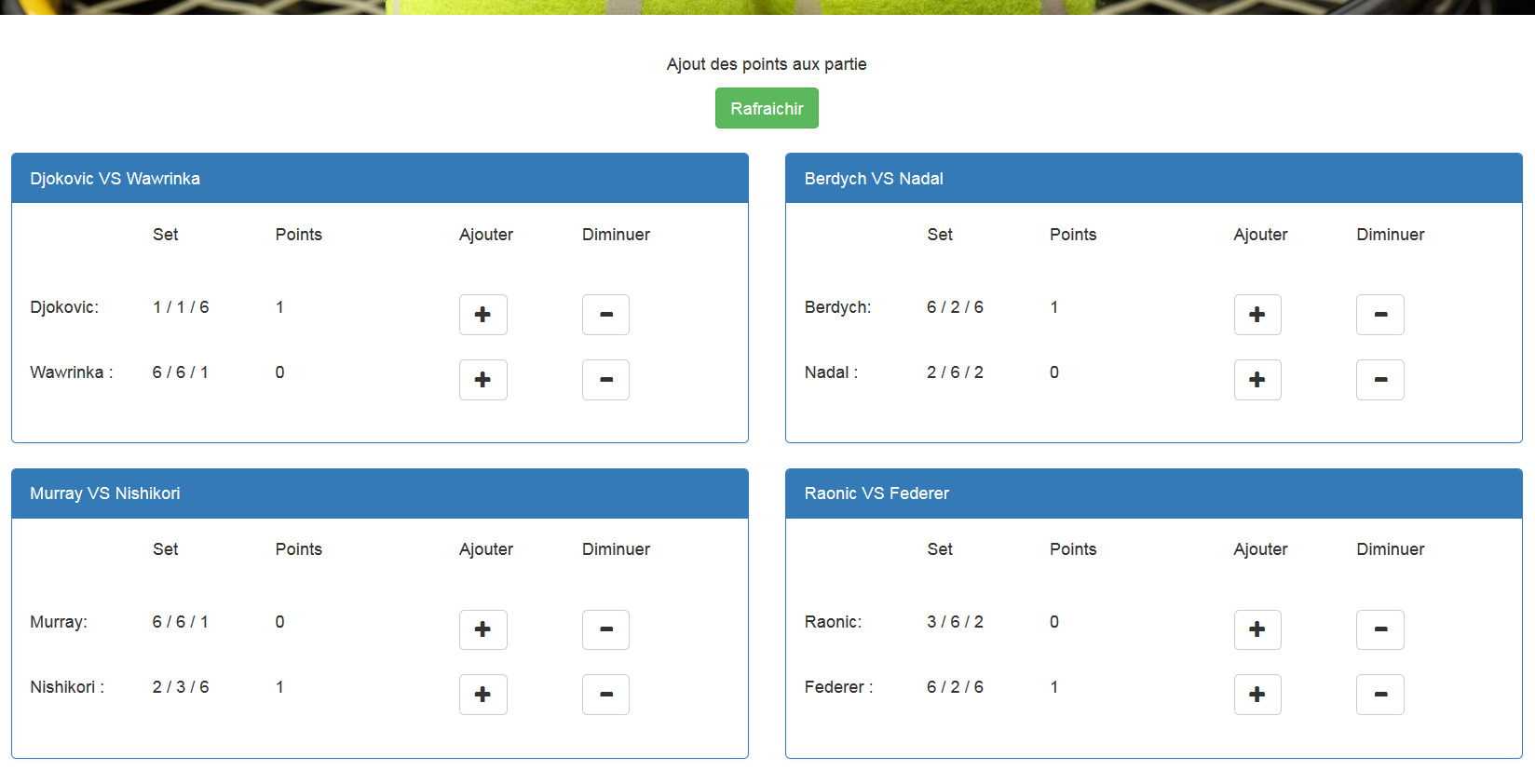
**La création de partie**

Le début du tournoi va commencer par la création des 4 parties avec 8 joueurs, il faut sélectionner le joueur 1, le joueur 2 et le terrain sur lequel ils vont jouer. Une fois que vous avez validé votre première partie, la liste des joueurs suivant diminuent en retirant avec du JavaScript les joueurs déjà sélectionné dans les autres liste déroulantes. Une fois les 4 parties créée on va pouvoir commencer le match avec chaque partie (à voir plus loin avec la gestion du terrain lors des autres tours).

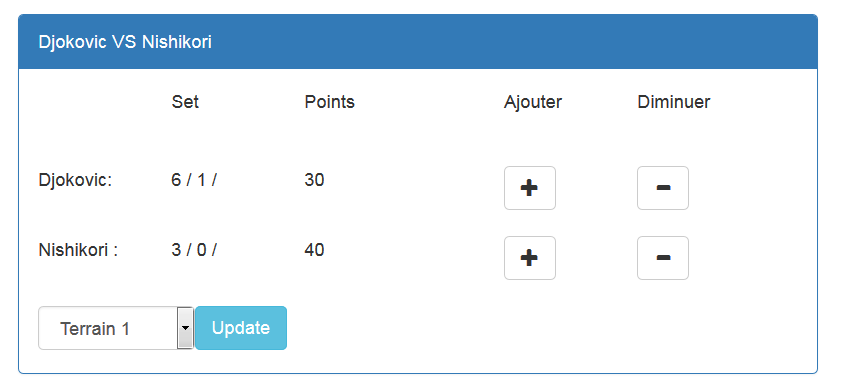
****

**L’ajout de point d’une partie**

L’administrateur a une interface ergonomique et simple pour rajouter les points aux parties, si par exemple nous sommes au premier tour, il y aura 4 panels ou l’ont peur rajouter les points sans devoir changer de page, ainsi l’administrateur pourra facilement gérer plusieurs partie à la fois ! Si jamais l’administrateur se trompe il pourra toujours rectifier le point qu’il a ajouté en trop, la gestion des sets se font automatiquement.

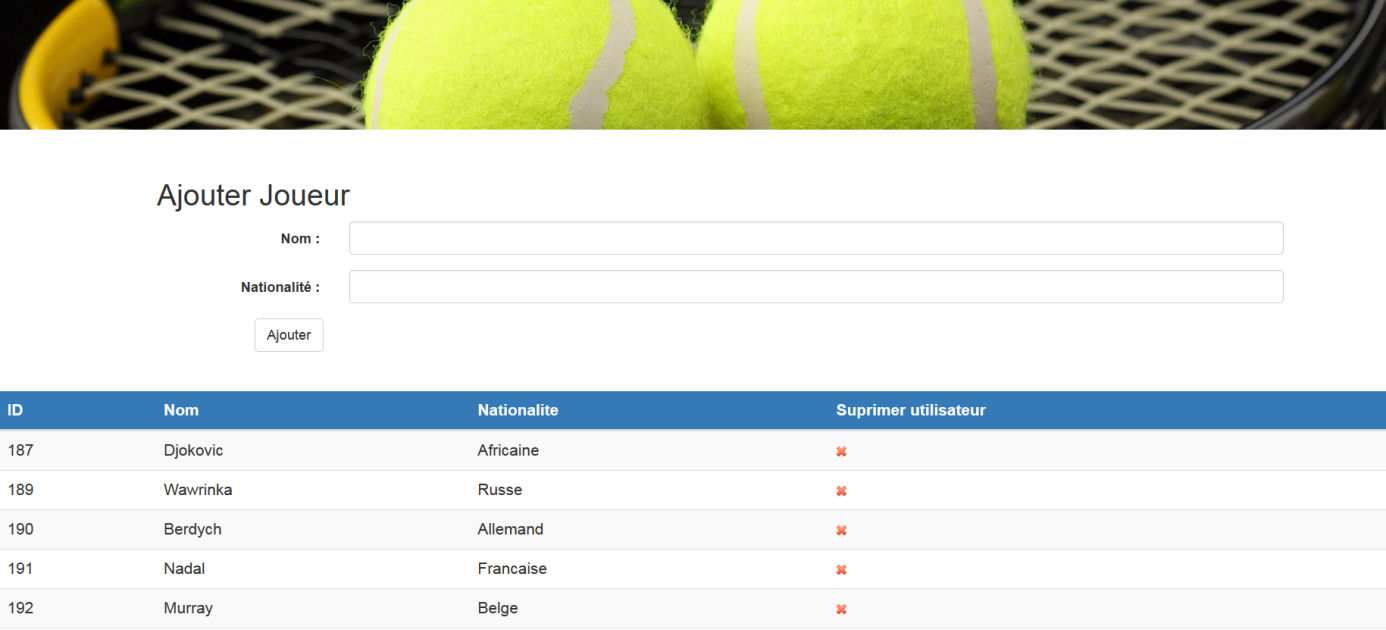


Lorsque la partie sera terminée le gagnant de la partie aura sur son score un 1 pour indiquer que c’est lui le gagnant, ici la vue avec une seule partie, on peut voir les boutons d’addition et soustraction des points, ainsi que la modification du terrain pour la finale et demi finale.



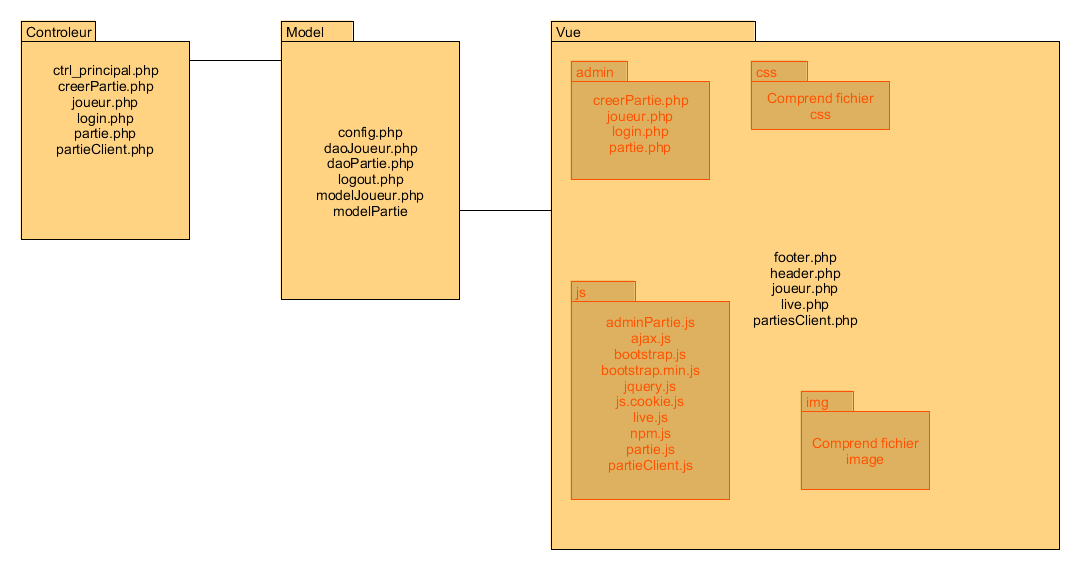
**L’administration des joueurs**

L’administrateur peut facilement ajouter ou supprimer un joueur, nous avons utilisé ici du jQuery pour que lorsque l’administrateur clique sur ajouter un joueur, celui-ci est automatiquement ajouté en fin de page, sans recharger la page. Il ne faut pas oublier que nous ne pouvons pas supprimer les joueurs qui participe biensur à une partie.



**L’architecture de nos fichiers (MVC)**

Nous avons partagé notre projet en plusieurs parties ; le contrôleur, le modèle et la vue. Lorsque nous arrivons à l’index du site celui-ci nous renvoie vers le contrôleur principal qui va décider ce qu’on désire faire. A partir de là des contrôleurs plus petit tel que joueur ou encore partie se charge de communiquer avec le modèle, nous avons notre couche DAO dans le modèle qui communique avec le serveur MySQL. Les fichiers de la section administration sont dans un dossier spécifique, ainsi que les fichiers jQuery, Ajax et JavaScript.



**MCD de la base de donnée**

