Pisano Vincent

Vasseur William

**Rapport technique :**

**Présentation du projet :**

Le contexte de l’application demandée dans ce projet concerne l’arbitrage de compétition de golf amateur. Celle-ci a pour but de créer un fichier PDF de cadence de jeu à destination des arbitres qui arbitrent les compétitions amateures. De nos jours l’arbitre principal d’une compétition reçoit de la part de la FFGolf un fichier Excel contenant la liste de tous les participant à la compétition. Ensuite, à partir de ce fichier l’arbitre créé à la main la cadence de jeu ce qui est très contraignant. En effet il doit lui-même créer les équipes de 2 ou 3 joueurs en regardant la liste des joueurs dans le fichier Excel. De plus il doit remplir l’heure de départ de la compétition ainsi que le temps estimé pour faire chacun des 18 trous dans le terrain de golf. Cela est très fatigant et peut être très facilement fait par une application. C’est pour cela que ce projet nous a été confié.

**Réflexion et Analyse du projet :**

Après avoir longuement pris connaissance du sujet et y avoir réfléchit et débattus en équipe nous avons séparé le projet en 2 grandes parties. La première grande partie consiste à récupérer du fichier Excel envoyer par la FFGolf les données pertinentes et de les conserver dans le format JSON. Les données que nous avons retenues sont la date de la compétition, le nom de la compétition, le nom et prénom de tous les participants ainsi que leurs niveaux. La seconde grande partie consiste à créer le tableau de cadence de jeu à partir des données présentes dans le JSON de la première grande partie ainsi que d’informations rentrées dans un formulaire.

**Présentation UML :**

Diagramme de cas d’utilisation

Une image contenant texte, carte

Description générée automatiquement

Voici notre schéma de cas d’utilisation. Celui-ci présente l’ensemble des utilisations possibles de l’application attendues dans la forme la plus parfaite. Notre équipe étant composée que de 2 membres seulement nous avons dès le début décidé de se concentrer sur la fonction la plus importante qui est la demande de cadence de jeu.

Diagramme de classe :

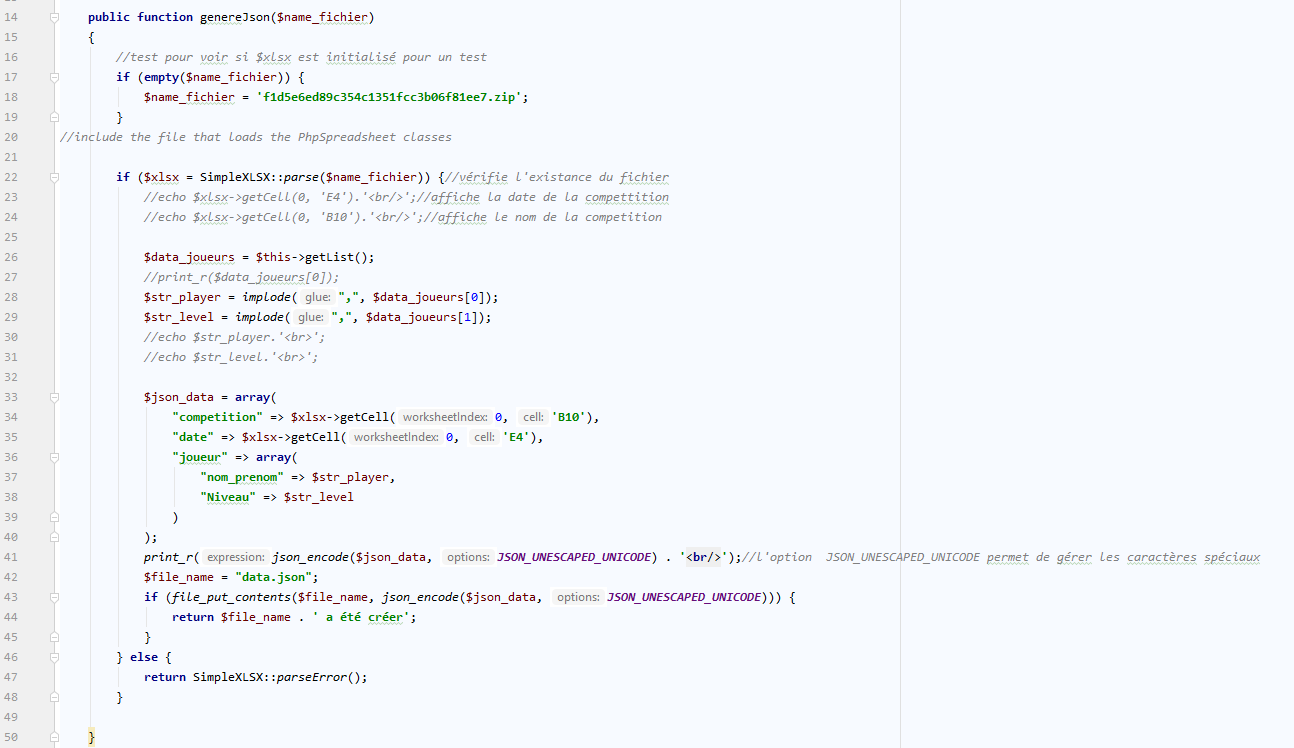
Une image contenant texte

Description générée automatiquement

Voici notre diagramme de classe qui nous a servi d’ébauche à la structure initiale de notre projet.

**Réalisation :**

La première grande réalisation que nous avons effectué dans ce projet fut l’extraction des données du fichier Excel sous le format JSON. Voici-ci-dessous la classe qui réalise cette fonction essentielle pour pouvoir mener à bien la fonction primordiale du projet : Demande de cadence de jeu.

****

Dans cet extrait de code nous pouvons voir la méthode « genereJson » qui permet de générer un fichier JSON à l’aide de la bibliothèque Simplexlsx.

Celle-ci est une bibliothèque avec de nombreuses et diverses méthodes mais la raison pour laquelle je l’ai choisie parmi toutes celles existantes est tout d’abord le fait qu’elle fonctionne sans extension, mais aussi qu’elle est facile à comprendre et intuitive à utiliser. Le GitHub où on a notamment trouvé cette bibliothèque est celui-ci : <https://github.com/shuchkin/simplexlsx> .

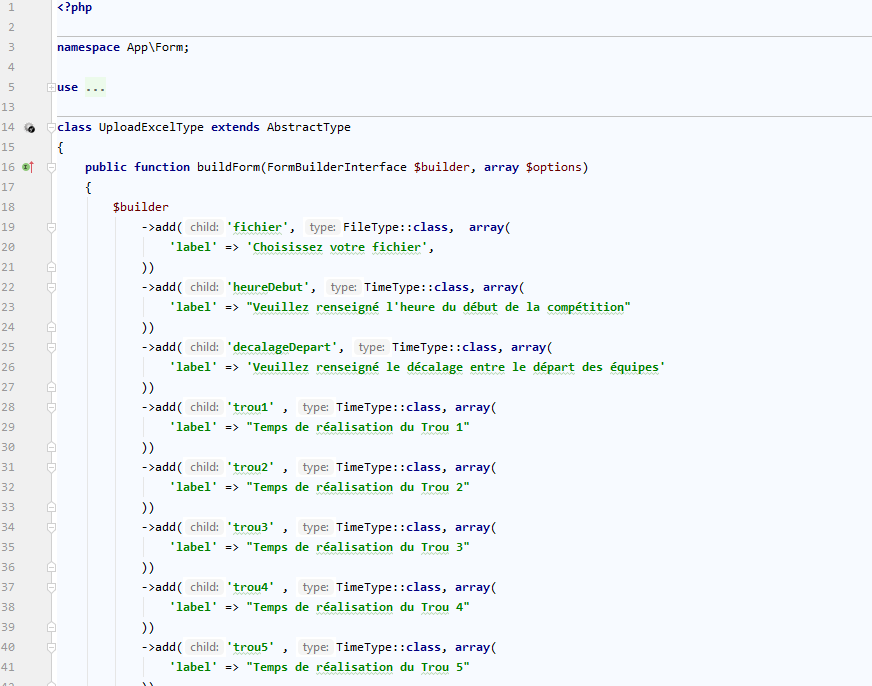
Il est très fourni en explication pour l’installer avec « composer » par exemple. De plus le développeur a mis de très nombreux exemples d’utilisation de sa bibliothèque comme : « comment sélectionner une feuille dans un xlsx », « récupérer le contenu d’une cellule ». Il a aussi mis à disposition des codes d’erreurs que sa bibliothèque retourne. Cela nous a été très utile pour comprendre qu’elle fût nos maladresses quant à l’utilisation de cette bibliothèque.

Intéressons-nous maintenant au code ci-dessus. On commence tout d’abord à la ligne 17 pour vérifier que le fichier Excel existe. S’il existe alors on récupère les données des joueurs comme leur nom et prénom ainsi que leurs Rep qui correspond à leur niveau grâce à la méthode « getList » à la ligne 26. Cette méthode nous retournera un tableau bidimensionnel.

C’est pour cela qu’aux lignes 28 et 29, on convertit en string le tableau bidimensionnel car, il faut insérer du texte dans le fichier JSON. A la ligne 33 on a créé une variable du nom de $json\_data qui stocke toutes les données qui par la suite seront insérées dans le JSON. Dedans est présent en première position le nom de la compétition puis la date de la compétition. Il y a ensuite le nom et prénom de tous les participants et enfin leur niveau.

Pour finir à la ligne 43 ont créé le fichier avec comme nom définit une ligne avant data.json avec comme contenu la variable $json\_date expliquée quelques lignes au-dessus. A cet instant-là, on a rencontré un problème de caractères spéciaux non pris en compte telle que les accents. Pour résoudre ce problème on a ajouté à la méthode « json\_encode » l’option « JSON\_UNESCAPEd\_UNICODE » qui a permis la résolution de ce problème.

Une fois que l’extraction est terminée, nous partons sur le formulaire à remplir afin de renseigner de nombreuses informations telle que le fichier Excel, l’heure du début de la compétition ainsi que la durée de chaque trou.



Ce formulaire a été créé par la commande symfony : php bin/console make :form. Nous ajoutons dans la variable $builder les différentes données que l’arbitre devra rentrer. On ajoute un FileType pour que l’arbitre puisse rentrer le fichier Excel, deux TimeType pour rentrer l’heure de départ de la compétition et le décalage entre le départ des équipes et un SubmitType pour valider les informations rentrées.

Une fois le formulaire fait nous nous sommes orientés vers la phase finale du projet qui est la génération du Tableau de cadence de jeu en PDF. Celui-ci est réalisé grâce à la classe ci-dessous.



Dans le code ci-dessus, à la ligne 105, nous déclarons la variable $tab qui contient le contenu html que nous allons par la suite convertir en un fichier pdf.

On crée une instance vers la classe HTML2PDF en déclarant la variable $pdf qui nous permettra par la suite d’appeler toutes les méthodes présente dans cette bibliothèque. On appelle la méthode writeHTML() avec comme paramètre $tab ce qui permet de préparer avant la génération du pdf le html. Pour terminer, à la ligne 118, la méthode output() avec comme paramètre « cadence\_de\_jeu.pdf » permet de générer le pdf avec les données saisies ci-dessus.

Pour des raisons de sécurité ainsi que le respect du RGPD (Règlement Général sur la Protection des Données) nous avons fait en sorte qu’aucune données ne soient conservées dans notre application. C’est pour cela que les lignes 120 à 127 supprimes toutes les données conservées.

Pour créer les équipes qui participerons à la compétition, nous avons créé et nous employons la classe Team.php :



Voici une des fonctions primordiales de cette classe. Dans la méthode equipeJauneTroisJoueur(), on parcourt l’ensemble des joueur qui ont le niveau jaune et nous réalisons les équipes de trois joueur jaune que nous stockons dans le tableau $team\_3\_jaunes. Pour finir la méthode, nous retournons ce tableau pour qu’il soit utilisable. Il existe aussi trois autre méthode similaire qui permettent de créé des équipes de trois joueur rouges, de 2 joueur jaunes et de 2 joueur rouges.

La structure de la base de données utiliser pour ce projet est :



Tests unitaires :



Ce test unitaire test la méthode genereJson lorsque le code ne parvient pas a trouver le fichier Excel.

A la ligne 12, on crée une instance de la classe ExtractionJson. La variable $EJson est donc un objet. La variable $name\_fichier contient le nom du fichier Excel. La variable $reponseAttendu contient la réponse que le code de la méthode du genereJson est censé nous retourner.

**Bibliothèques :**

* Simplexlsx : C’est une bibliothèque avec de nombreuses et diverses méthodes mais la raison pour laquelle nous l’avons choisie parmi toutes celles existantes est tout d’abord le fait qu’elle fonctionne sans extension, mais aussi qu’elle est facile à comprendre et intuitive à utiliser. Le GitHub où on a notamment trouvé cette bibliothèque est celui-ci : <https://github.com/shuchkin/simplexlsx> . Il est très fourni en explications pour l’installer avec « composer » par exemple. De plus le développeur a mis de très nombreux exemples d’utilisation de sa bibliothèque comme : « comment sélectionner une feuille dans un xlsx », « récupérer le contenu d’une cellule ». Il a aussi mis à disposition des codes d’erreurs que sa bibliothèque retourne. Cela nous a été très utile pour comprendre qu’elle fût nos maladresses quant à l’utilisation de cette bibliothèque.
* HTML2PDF : Il s’agit d’une bibliothèque permettant de convertir du HTML en PDF. Celle-ci est compatible avec PHP des versions 5.4 à 7.2. Afin d’utiliser cette bibliothèque il faut il faut avoir un HTML valide puis générer le document PDF à partir de cela. Pour la partie PDF, la bibliothèque utilise TCPDF qui est une classe PHP répandue qui permet de créer des fichier PDF. Le gitHub de celle-ci est celui-ci : <https://github.com/spipu/html2pdf> . Il est à jour, avec un Readme détaillé qui explicite avec quelle version de PHP elle fonctionne. En plus de cela de nombreux exemples sont présent sur ce git.

**Conclusion :**

Notre groupe durant ce projet a réalisé et achevé la fonction principale du projet qui est la demande de cadence de jeu. La page initiale du projet est le formulaire à remplir. Une fois tous les champs remplis, on arrive à l’écran d’affichage du tableau en web. Pour finir, le pdf est généré si on clique sur le bouton généré en bas du tableau web. Cette partie est opérationnelle mais sans couverture visuelle. Nous nous sommes concentrés sur le respect des consignes données ainsi que sur un rendu correct. Le projet Pragolf n’est pas fini, il reste de nombreuse fonctionnalité à ajouter comme une page d’authentification, du CSS afin de rendre cela plus intuitif et beau visuellement pour les futurs utilisateurs. Cela fut un projet intéressant et instructif qui a permis de développer la communication et le travail d’équipe.