Pisano Vincent

Vasseur William

SIO22

Compte rendue Phase 2 :

Diagramme de cas d’utilisation :

Une image contenant texte, carte

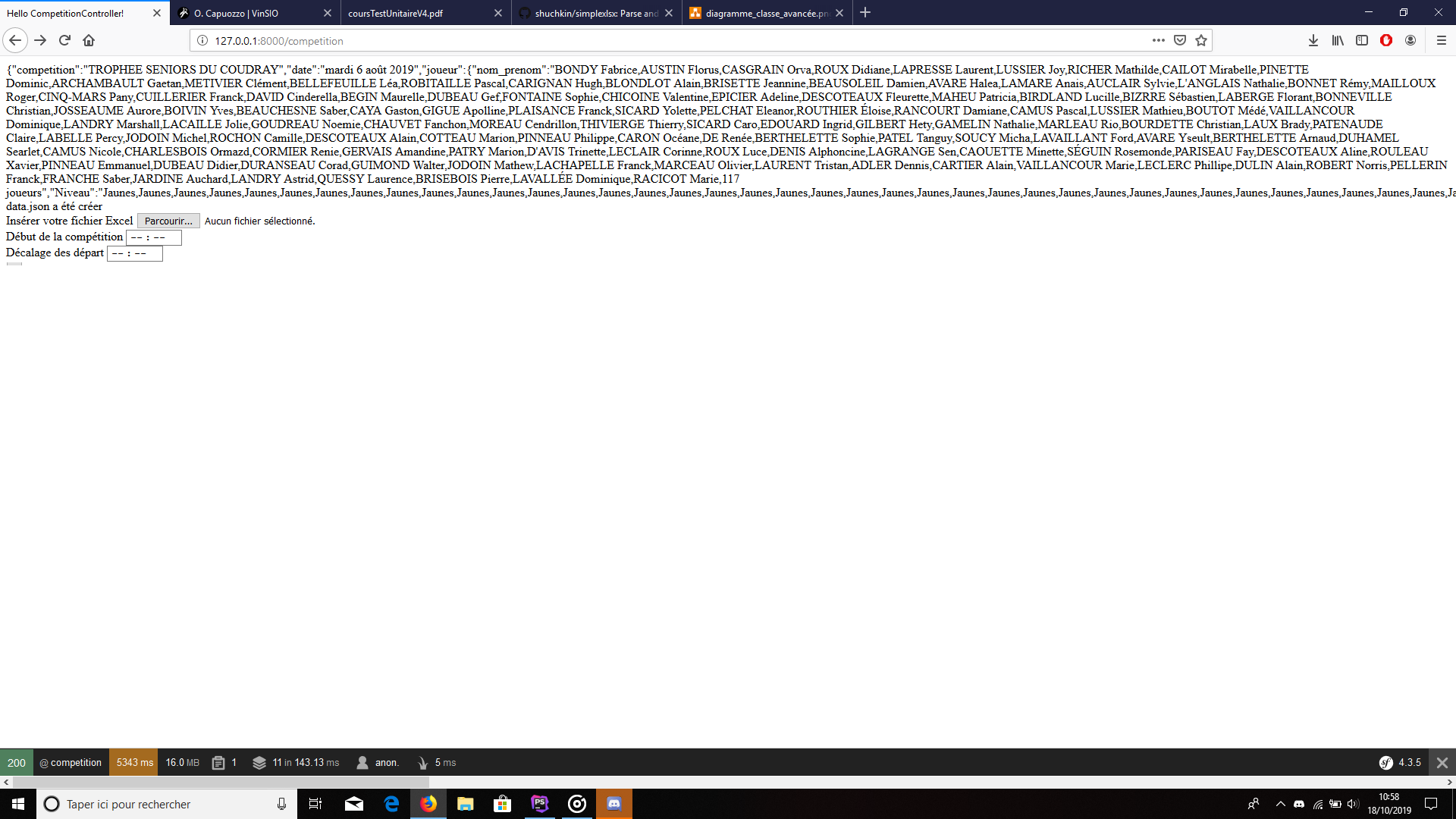
Description générée automatiquement

Diagramme de classe :

Une image contenant texte

Description générée automatiquement

Prototype fonctionnel :



Voici le rendu visuel de notre prototype. La partie fonctionnelle que notre groupe à couvert est celle de la conversion des données présente dans le fichier Excel en un fichier JSON. Tout cela en faisant attention de ne récupérer que les données essentielles pour réaliser la demande de cadence de jeu.

La fonctionnalité de cadence de jeu est donc à moitié fini. Il suffit donc de générer le PDF à partir des données présentent dans le fichier JSON que nous avons créé.

Sur l’image ci-dessus nous voyons présent entre les deux crochets le contenu du fichier JSON. Nous pouvons aussi apercevoir la phrase « data.json a été créée », qui nous indique que le fichier à bel et bien été créé avec les données au-dessus de cette phrase.

En dessous de cette phrase se trouve un formulaire qui permettra une fois fonctionnel de choisir le fichier Excel à transmettre pour une demande de cadence de jeu, d’indiquer l’heure du début de la compétition ainsi que le décalage entre les départs des équipes.

A l’heure actuelle, on peut choisir le fichier Excel, donner l’heure de départ de la compétition et l’intervalle de décalage entre les compétiteurs. Cependant il faut encore transmettre ces données au serveur ainsi qu’uploader le fichier Excel.

Présentation de code :



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 maintenant au code ci-dessus. On commence tout d’abord à la ligne 11 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 13. Cette méthode nous retournera un tableau bidimensionnel.

C’est pour cela qu’aux lignes 15 et 16, on convertit en string le tableau bidimensionnel car, il faut insérer du texte dans le fichier JSON. A la ligne 18 on a créé une variable du nom de $json\_data qui stocke toutes les données qui par la suite seront inséré 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 27 on crée 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.



Sur cette image nous voyons la méthode « getList » qui retourne le tableau bidimensionnel avec dedans le nom et prénom de tous les participants ainsi que leur niveau respectif.

Pour trouver les noms et prénoms ainsi que leur niveau, on parcourt toute la colonne ‘B’ du Excel où sont présents les noms et prénoms. Si la valeur de la cellule est ‘Nom et Prénom’ alors on sait qu’à partir de la case suivante il y a un nom et prénom d’un joueur.

Donc à partir de ce moment on stocke son nom présent à la ligne par exemple i de la colonne ‘B’. Pour trouver son niveau il suffit d’aller récupérer le contenu de la cellule à la ligne i et la colonne ‘I’ qui correspond à la colonne Rep qui en golf représente le niveau d’un joueur.

Dès que l’on détecte que le contenu de la case est vide c’est qu’on a fini de lire le premier tableau et qu’il faut passer au suivant. Une fois toutes les données extraites du Excel et qu’elles sont stockées dans le tableau $list la méthode se termine en retournant ce tableau.

Tests unitaires :



Ce test unitaire test la méthode genereJson lorsque le code ne parvient pas a trouvé le fichier excel.

A la ligne 12, on créer 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 retourné.