**Projet : La gestion des joueurs et de leurs statistiques**

1. **Structure du projet foot :**

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure 1 : Structure du projet foot

**Assets :**

Ce dossier contient tous les assets nécessaires pour le front d’une application Symfony. Les assets sont principalement des fichiers .css et .js.

**Bin :**

Ce dossier contient les fichiers de commandes permettant d’effectuer des actions sur un projet Symfony. Par exemple, de vider le cache Symfony ou mettre à jour la base de données.

**Config :**

Toute la configuration des packages, services et routes se fera dans ce dossier. Les fichiers de configuration sont par défaut en YAML, même s’il est tout à fait possible d’utiliser PHP ou XML.

**Migrations :**

Ce dossier contient les migrations de votre projet générées à chaque changement que vous effectuerez sur votre base de données à l'aide de l'ORM Doctrine.

**Public :**

Par défaut, il ne contient que le contrôleur frontal de votre application, le fichier dont la responsabilité est de recevoir toutes les requêtes des utilisateurs.

**Src :**

Contenant les fichiers sources de l’application (contrôleurs, entités, reposotry...).

* **Controller** : Ce dossier contient vos contrôleurs qui se chargent de rediriger vers les Manager / Service / Repository.
* **Entity** : Dans ce dossier nous allons définir la structure de votre base de données au travers de classes. Chaque Entity représente généralement une table en Base de données.
* **Repository** : Un Repository est toujours rattaché à une Entity, il nous permet de créer nos fonctions qui iront requêter la table de notre Entity.
* **Form** : Dans ce dossier nous allons créer le formulaire associe à chaque Entity.
* **DataFixtures** : Dans ce dossier nous allons trouver un ensemble de données qui permet d’avoir un environnement de développement proche d’un environnement de production avec des fausses données.

**Templates :**

Ce dossier contient en d’autre terme nos views. Symfony utilise le moteur de Template Twig par défaut.

**Tests :**

Contenant les fichiers permettant de tester l’application : les tests unitaires **PHPUnit** :

La commande pour lancer nos tests : php bin/phpunit

**Translations :**

L’internationalisation des applications est très importante aujourd’hui. Il est donc nécessaire de mettre en place un système de traduction dès le début du projet.

**Var :** Utilise par Symfony pendant l’exécution, contenant les données de cache, le log et les sessions.

**Vendor :** Ce dossier contient votre chargeur de dépendances (ou "autoloader") et l'ensemble des dépendances de votre projet PHP installées à l'aide de Composer.

1. **Étude fonctionnelle et analyse des besoins :**
2. **Identifications des acteurs :**

**L’administrateur  (Le dirigeant, Le gestionnaire) :**

L’administrateurpeut assurer les taches suivantes :

* S’authentifier
* Gérer les équipes
* Gestion des joueurs
* Gérer les statistiques des joueurs
* Gérer les saisons

1. **Besoins fonctionnelles :**

**L’administrateur  (Le dirigeant, Le gestionnaire) :**

* S’authentifier :

Le système affiche l’interface d’authentification : Dans ce cas, il faut saisir les coordonnées suivantes (identifiant et Mot de passe) puis cliquer sur le bouton « SE CONNECTER ».

* Gérer les équipes :

L'administrateur a le droit de créer un compte d’une équipe, consulter la liste des équipes et consulter les détails d’une équipe.

* Gestion des joueurs

L'administrateur à la possibilité d'ajouter un joueur, consulter la liste des joueurs.

* Gérer les statistiques des joueurs

L'administrateur a le droit d'ajouter la statistique d’un joueur et consulter les statistiques de chaque joueur pour son équipe pour une saison donnée.

* Gérer les saisons

L'administrateur a le droit d'ajouter une saison et consulter la liste des saisons.

1. **Besoins non Fonctionnelles :**

Performance :

* Temps de réponse : le chargement de l’application, ouverture d’écran et des délais de rafraîchissement, etc.
* En temps de traitement : fonctions, calculs, importations/exportations de données
* L’interrogation de données et Rapports : temps de chargement initial et des chargements

Sécurité :

* Le mot de passe doit être complexe (contient des lettres minuscules, majuscules, chiffres et caractères spéciaux), déconnexion après temps morts d’inactivité.

La maintenance :

* le code de l’application soit bien organisé et commenté pour une maintenance plus simple.

La convivialité :

* l'application doit être pratique. En effet, les interfaces doivent être ergonomiques, simples et adéquates à l’utilisateur.

1. **Langage de modélisation adopté :**

Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML) :

Un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.  
UML nous fournit donc des diagrammes pour représenter l’application à développer : son fonctionnement, sa mise ne route, les actions susceptibles d’être effectuées par l’application.

1. **Etude conceptuelle :**
2. **Diagramme de cas d’utilisation de l’administrateur :**

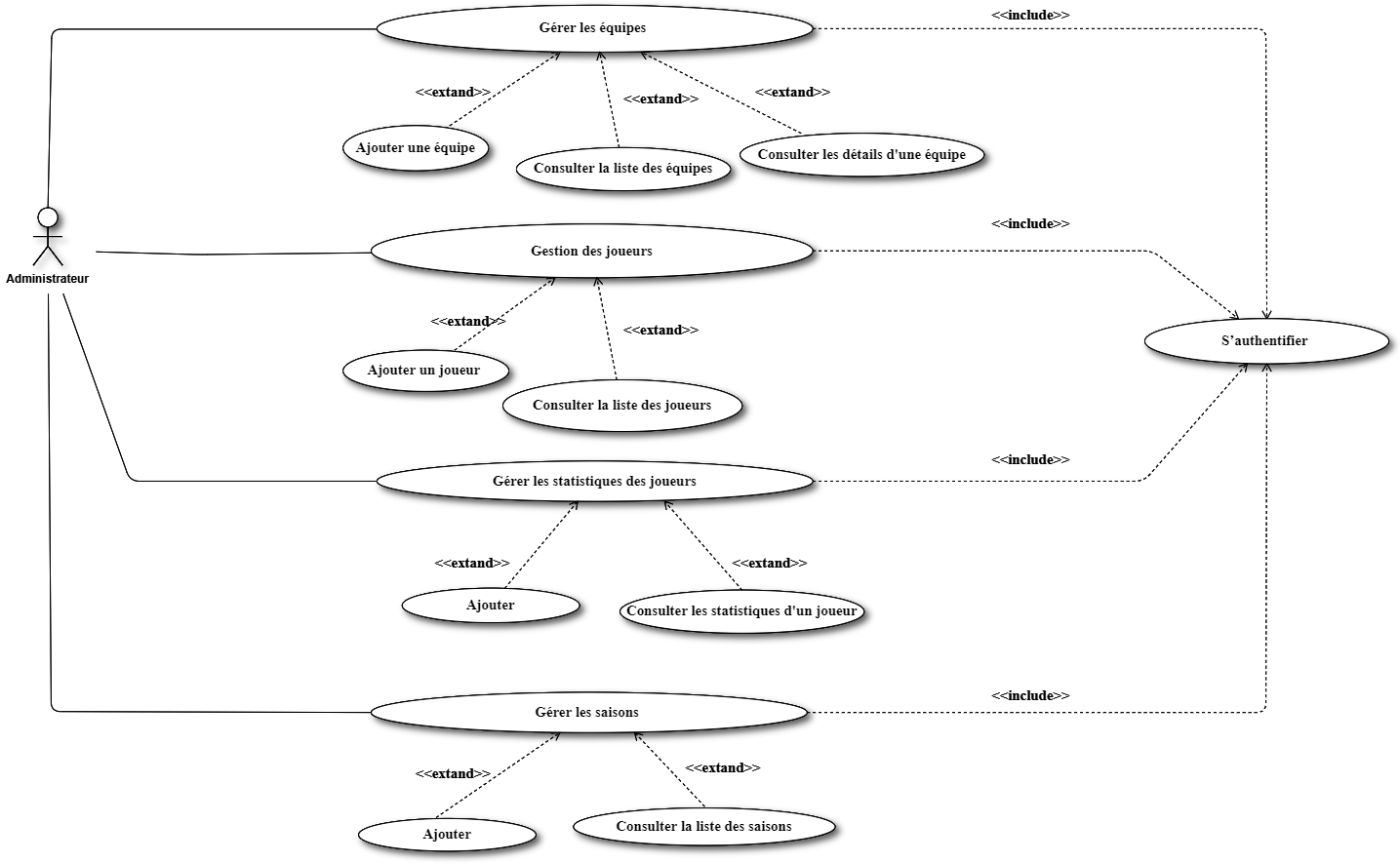


Figure 2 : Diagramme de cas d'utilisation de l'administrateur

1. **Diagramme de classe de l’administrateur :**

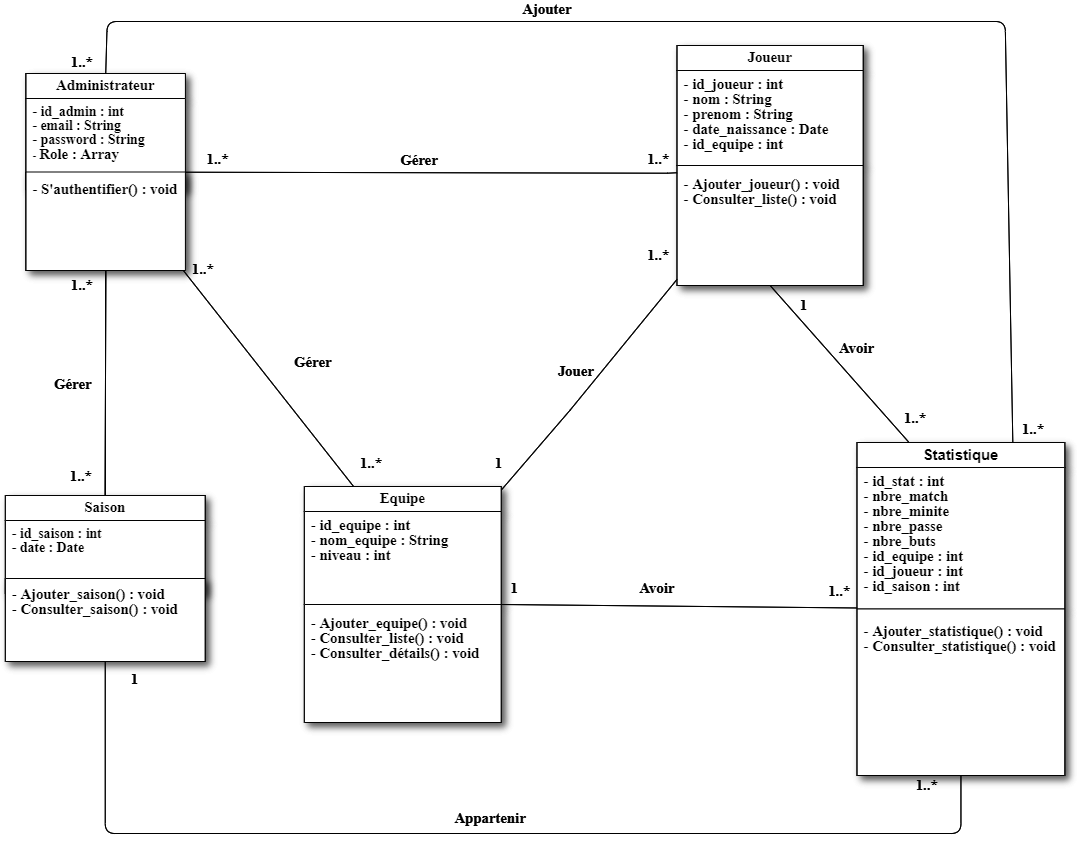


Figure 3 : Le diagramme de classe

1. **Réalisation**
2. **Choix des langages de développement et de SGBD :**

* **Langage de développement** :

 **Symfony (version 6.0)** : Un ensemble de composants PHP ainsi qu'un [framework](https://fr.wikipedia.org/wiki/Framework" \o "Framework) [MVC](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur) [libre](https://fr.wikipedia.org/wiki/Logiciel_libre) écrit en [PHP](https://fr.wikipedia.org/wiki/PHP).

** Bootstrap 5** : Une [collection d'outils](https://fr.wikipedia.org/wiki/Framework) utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de [sites](https://fr.wikipedia.org/wiki/Site_web) et d'[applications web](https://fr.wikipedia.org/wiki/Application_web). C'est un ensemble qui contient des codes [HTML](https://fr.wikipedia.org/wiki/HTML) et [CSS](https://fr.wikipedia.org/wiki/Cascading_Style_Sheet), des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions [JavaScript](https://fr.wikipedia.org/wiki/JavaScript) en option.

* **Base des données** **: Mysql**

 Un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle.

1. **Fixtures et tests techniques /fonctionnels :**

**JoueurControllerTest :**

Code :

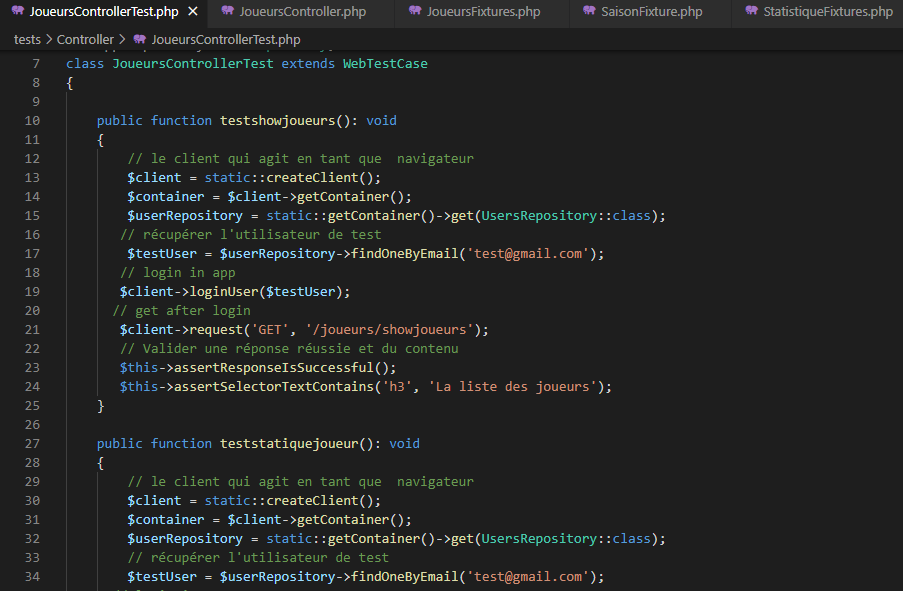


Figure 4 : Le contrôleur de test d'un joueur

Cmd :

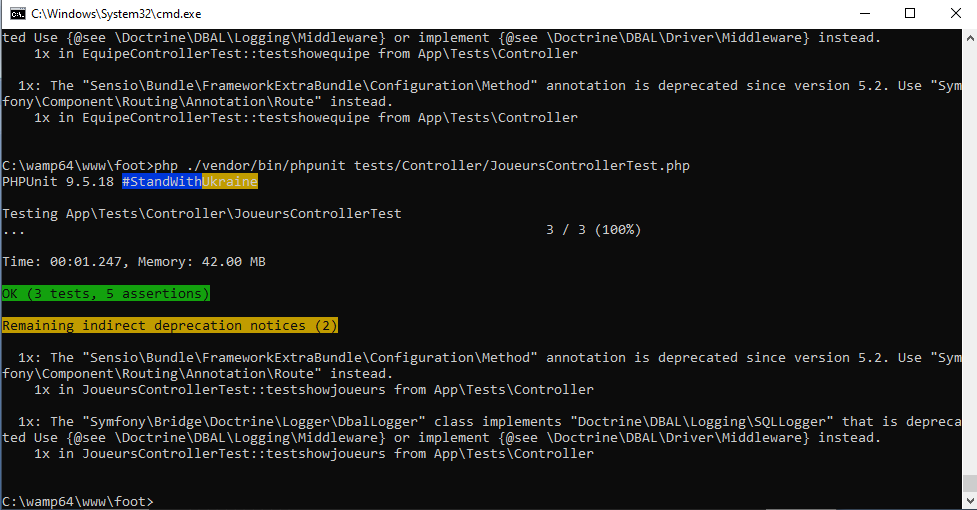


Figure 5 : Vérification de fonctionnalité de test

Si on va traiter des données incorrectes pour vérifier le test :

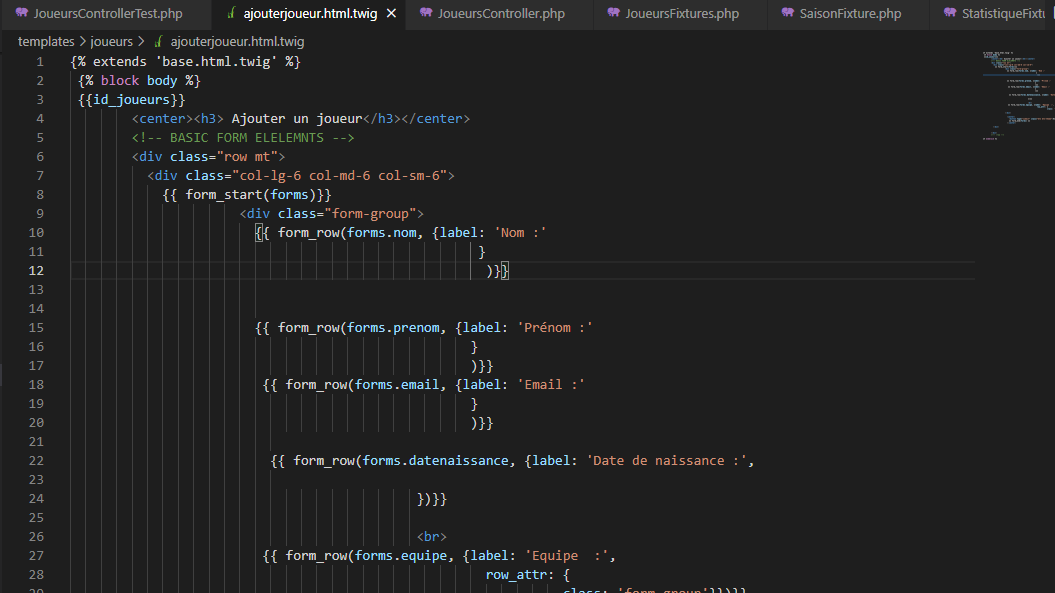


Figure 6 : Insertion des données incorrectes

L’exécution :

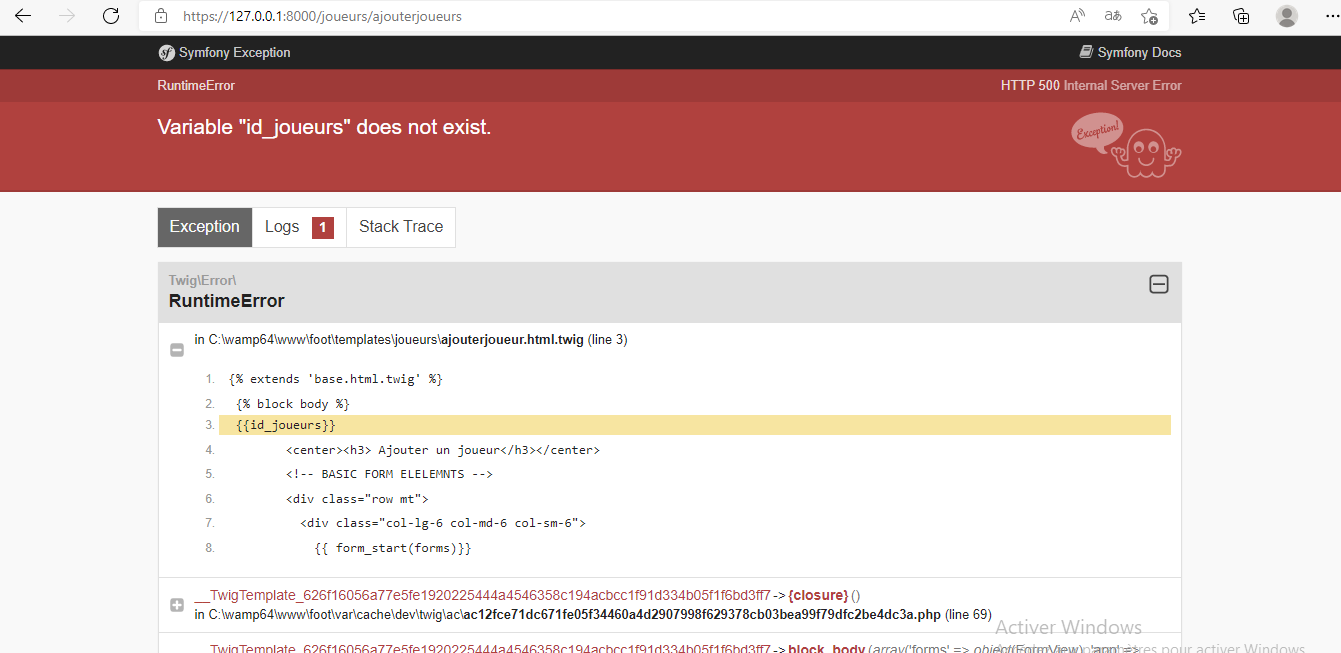


Figure 7 : L’exécution

Avec cmd :

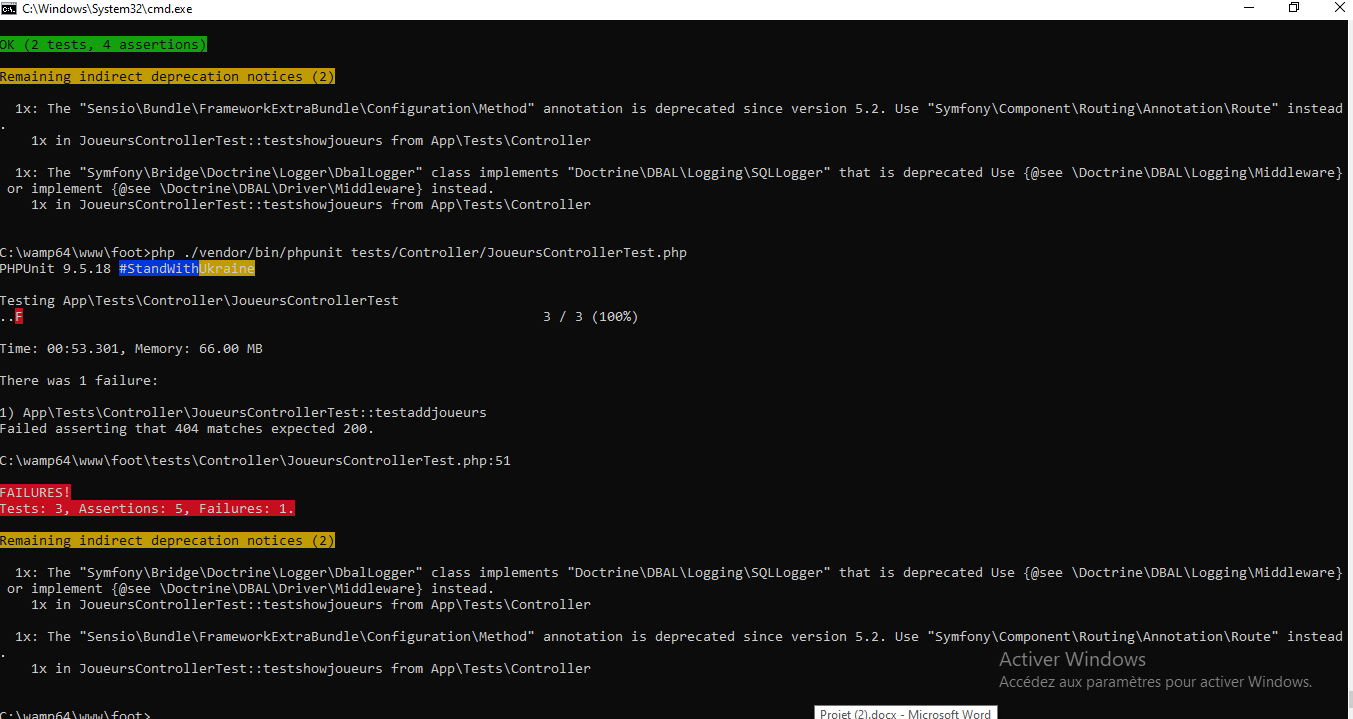


Figure 8 : Vérification de fonctionnalité de test

Fixtures :

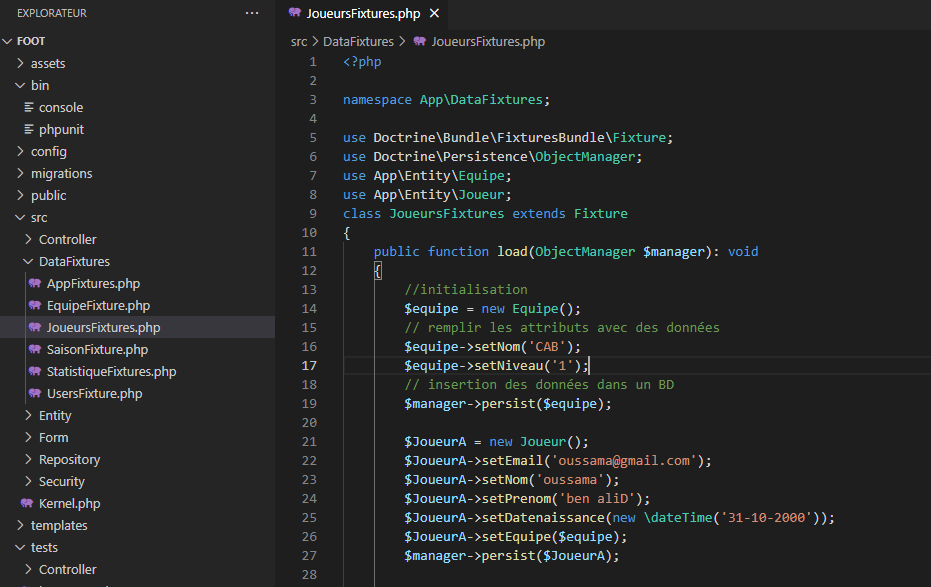


Figure 9 : Fixtures d'entité joueur

Base de donné :

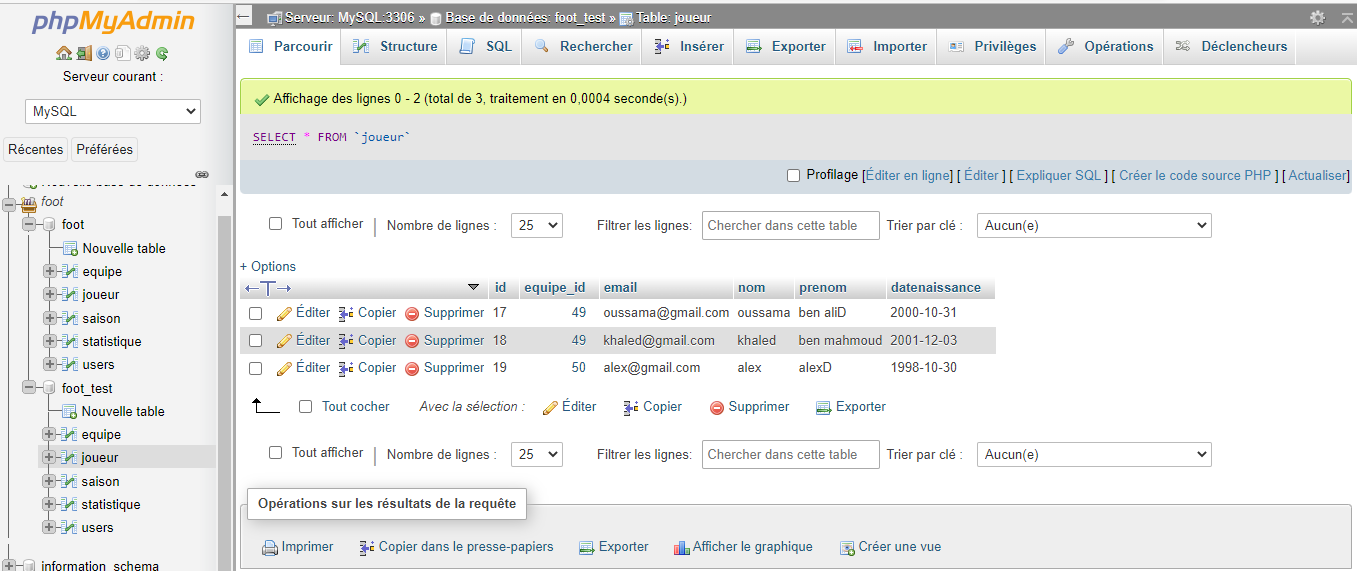


Figure 10 : Table "joueur" de base de données foot\_test

**EquipeControllerTest :**

Code :

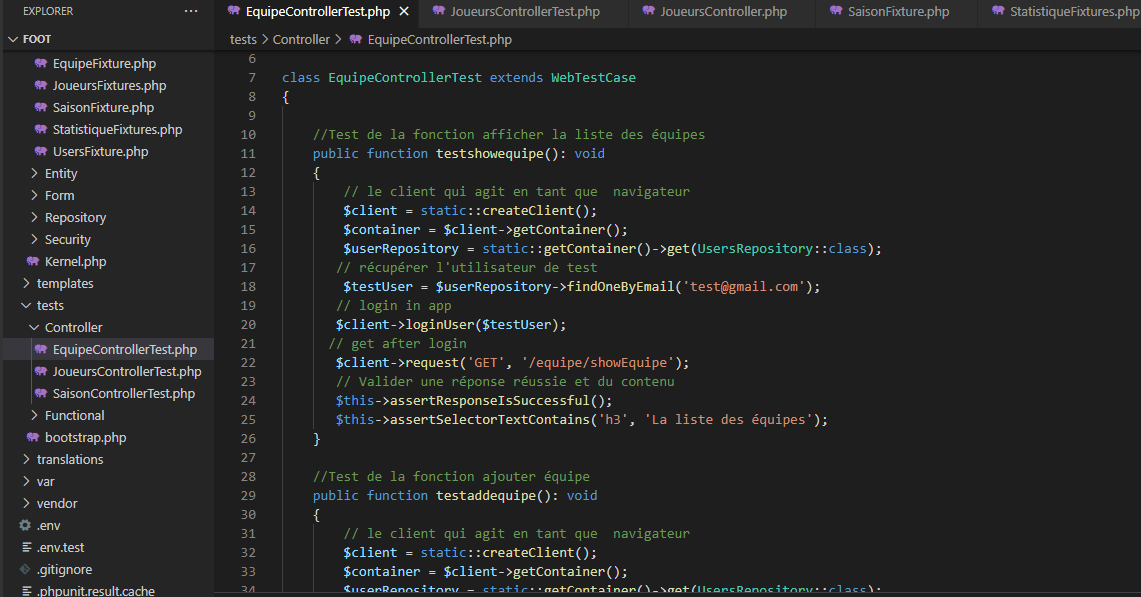


Figure 11 : Le contrôleur de test d'une équipe

Cmd :

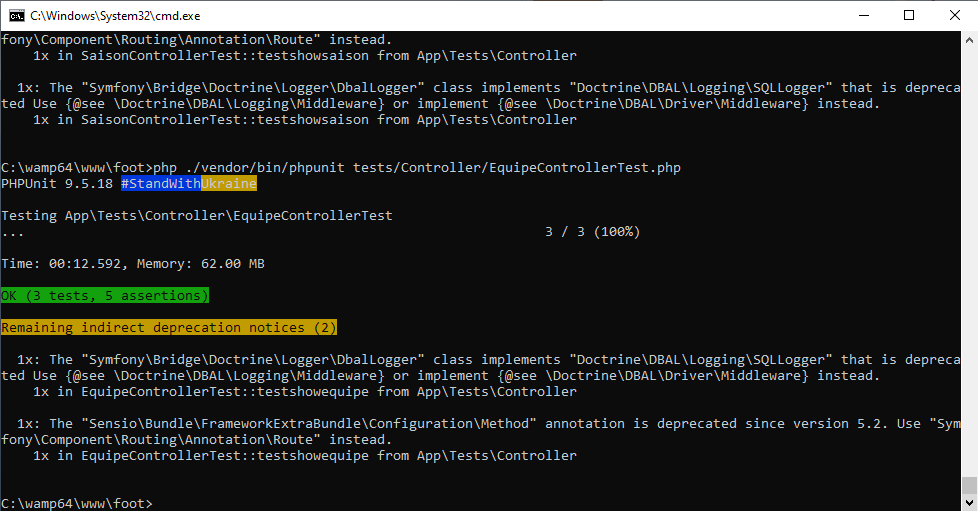


Figure 12 : Vérification de fonctionnalité de test

Données incorrectes:

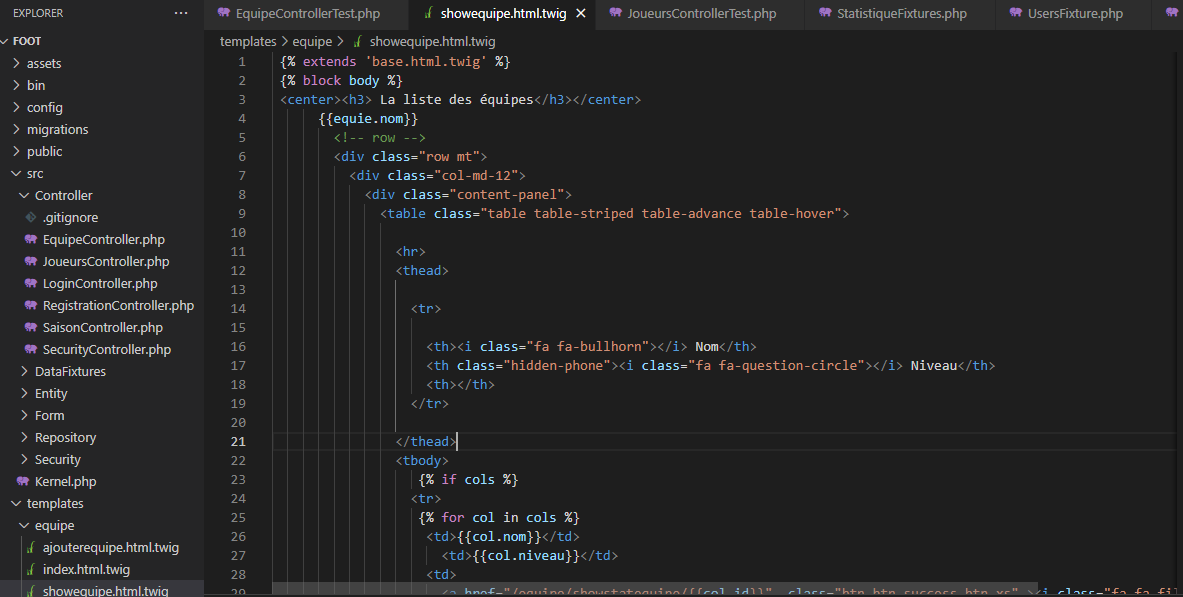


Figure 13 : Insertion des données incorrectes

L’exécution :

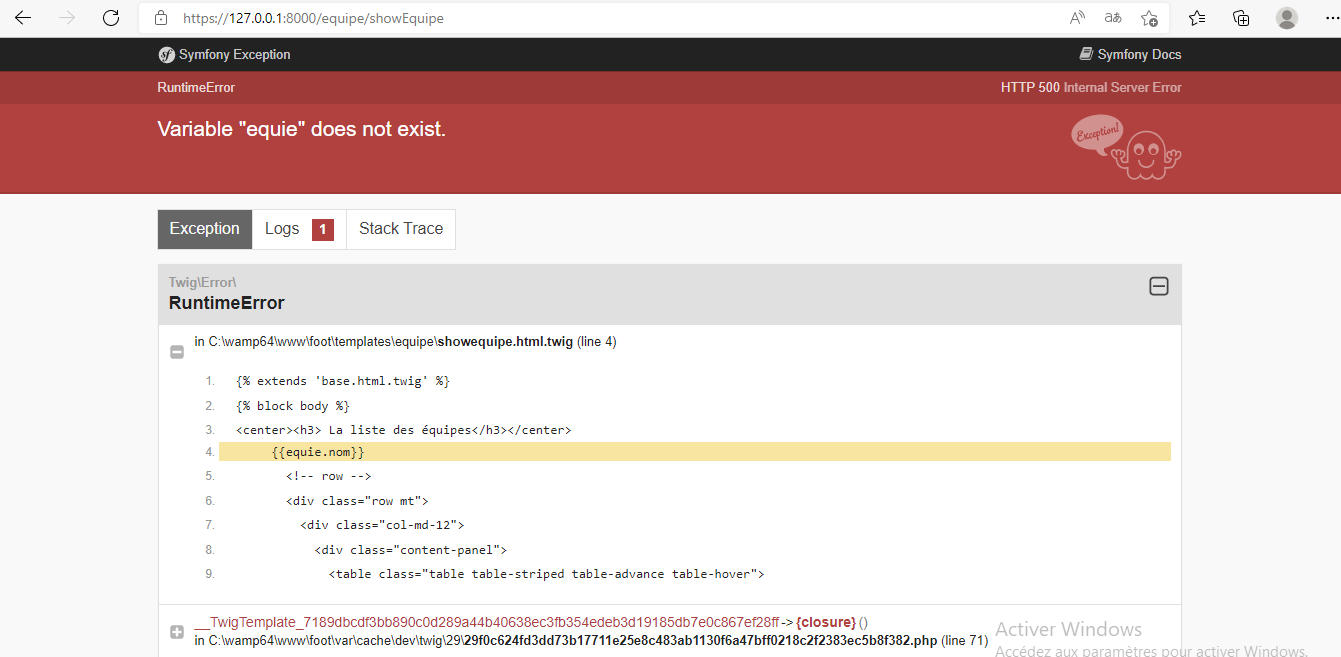


Figure 14 : L’exécution

Cmd :

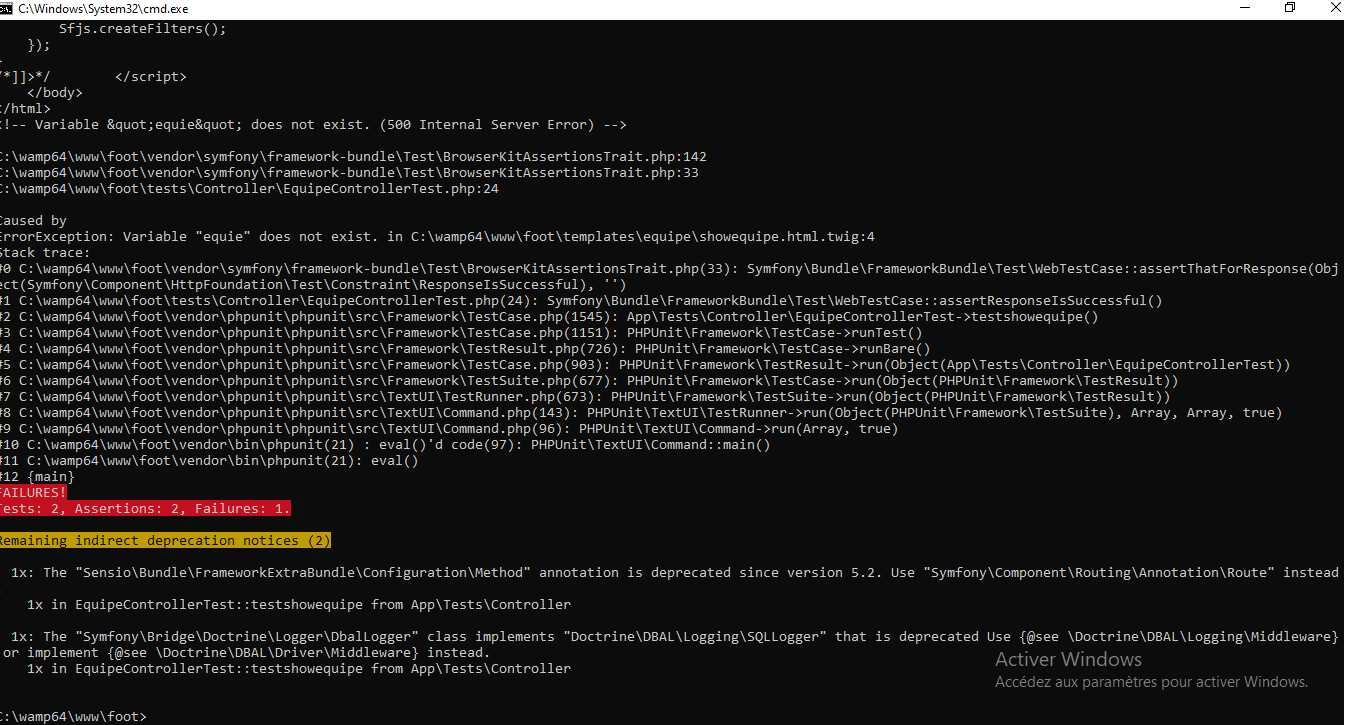


Figure 15 : Vérification de fonctionnalité de test

Fixtures :

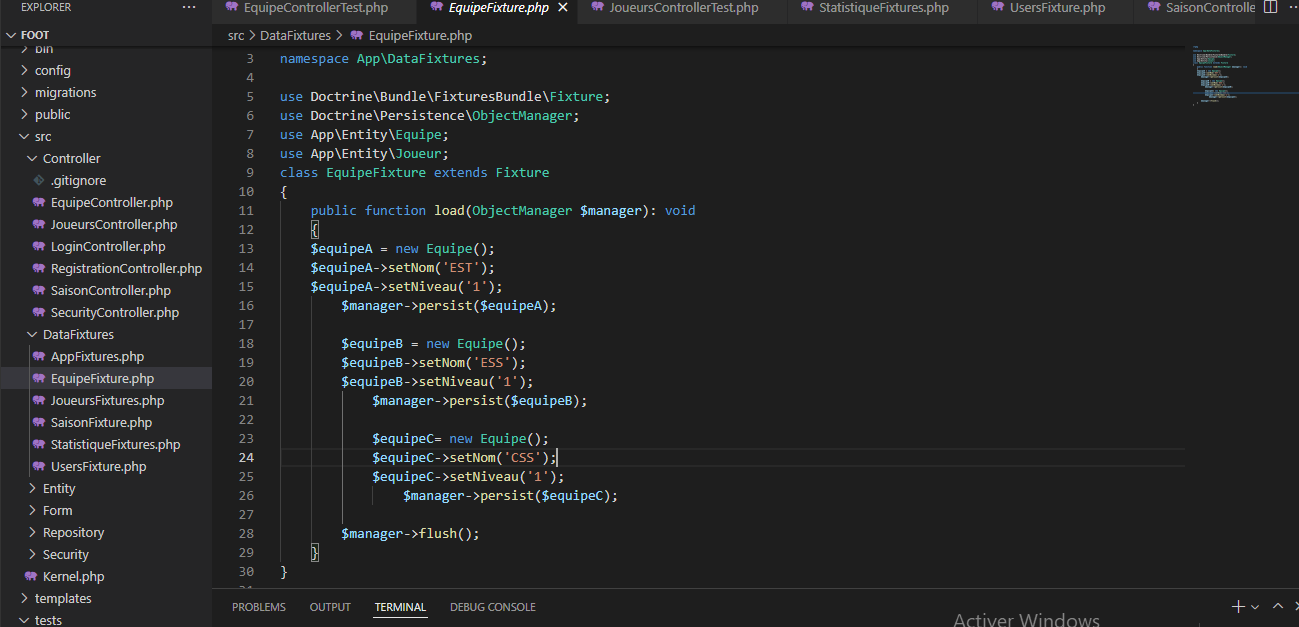


Figure 16 : Fixtures d'entité équipe

Base :

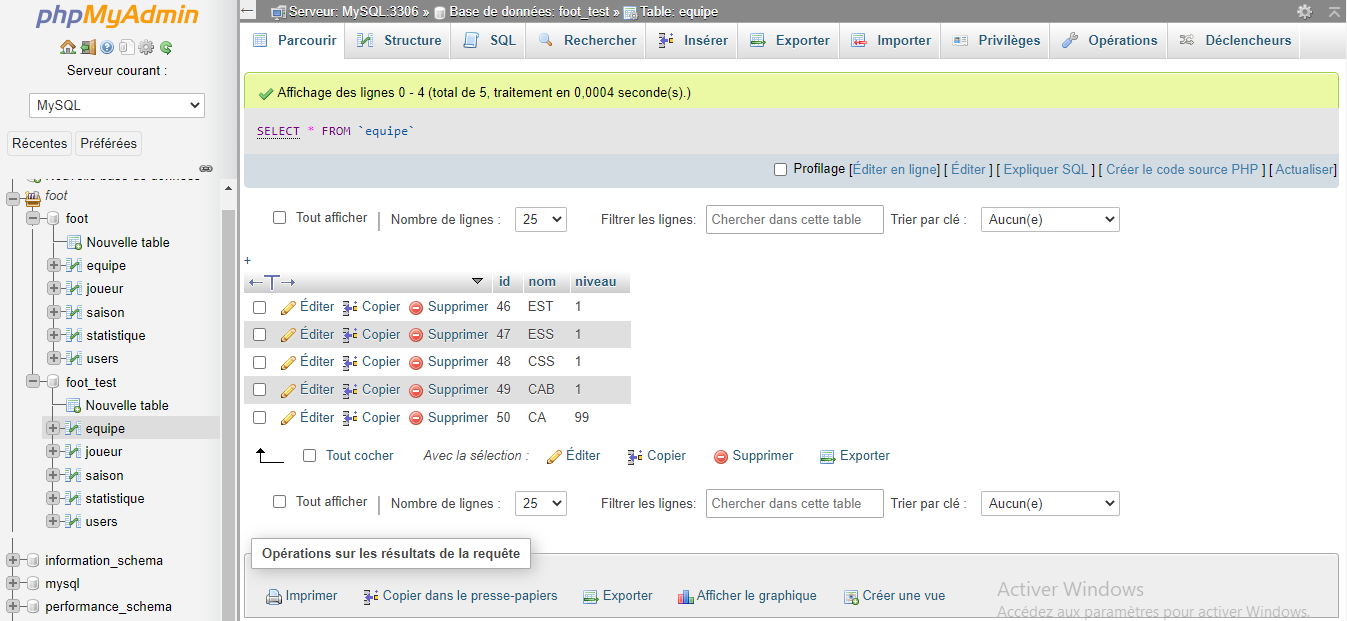


Figure 17 : Table "Equipe" de base de données foot\_test

**SaisonControllerTest :**

Code :

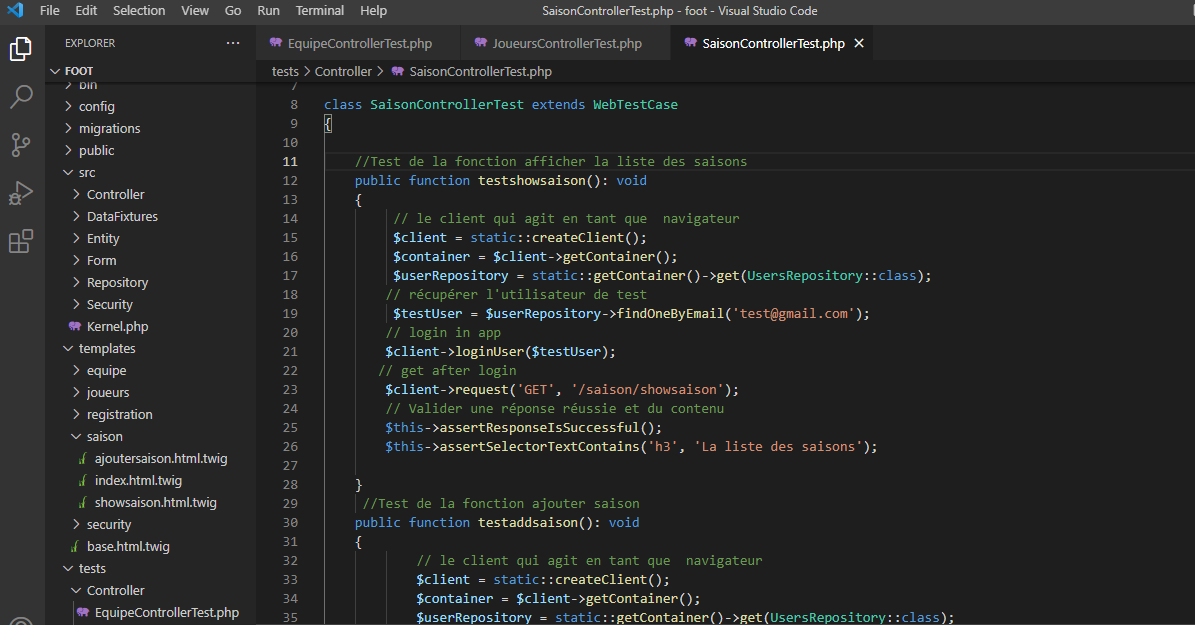


Figure 18 : Le contrôleur de test d'une saison

Cmd :

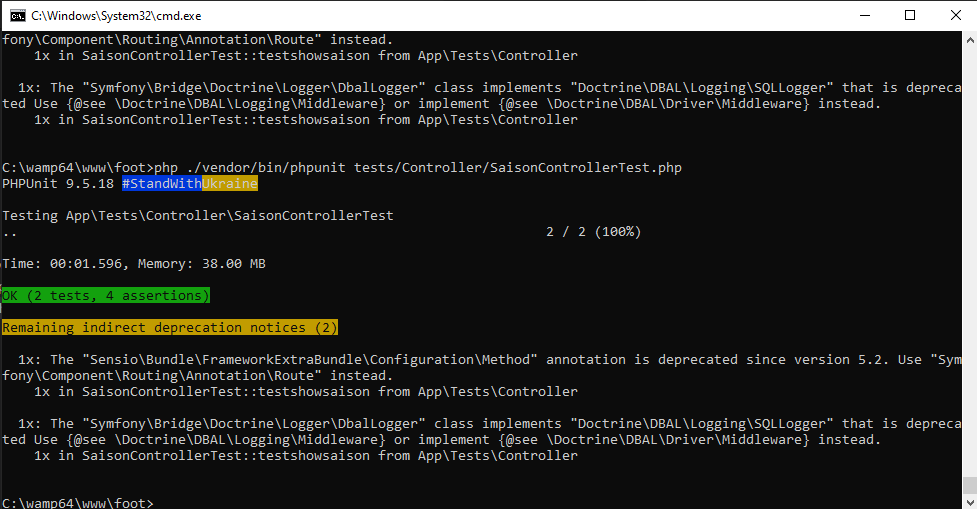


Figure 19 : Vérification de fonctionnalité de test

Données incorrectes:

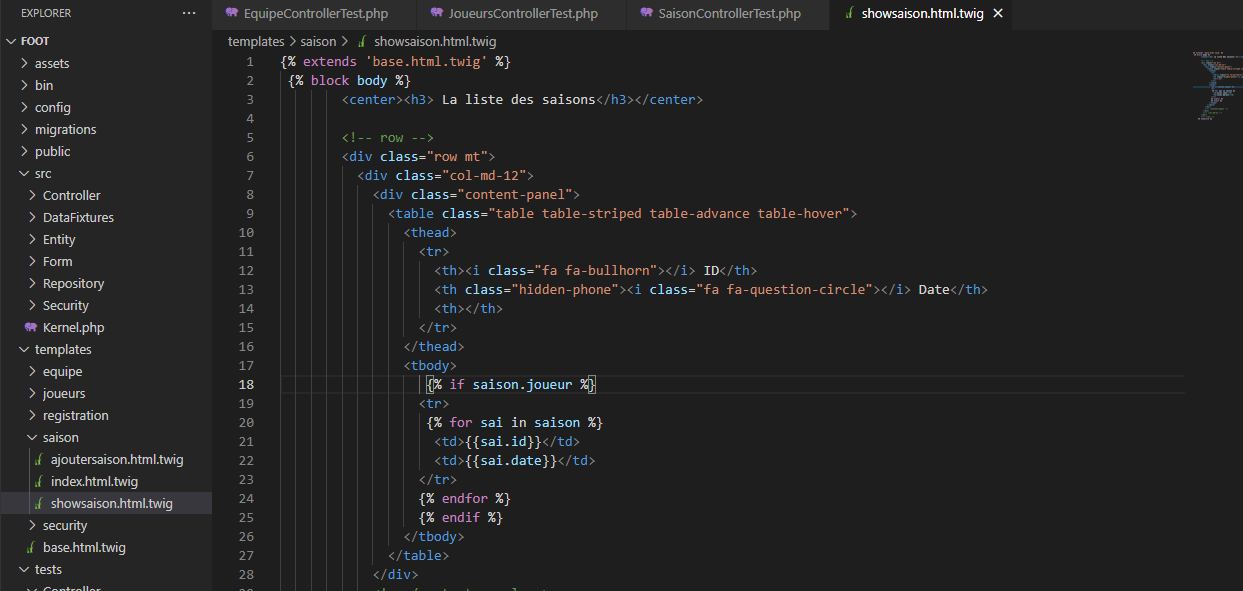


Figure 20 : Insertion des données incorrectes

L’exécution :

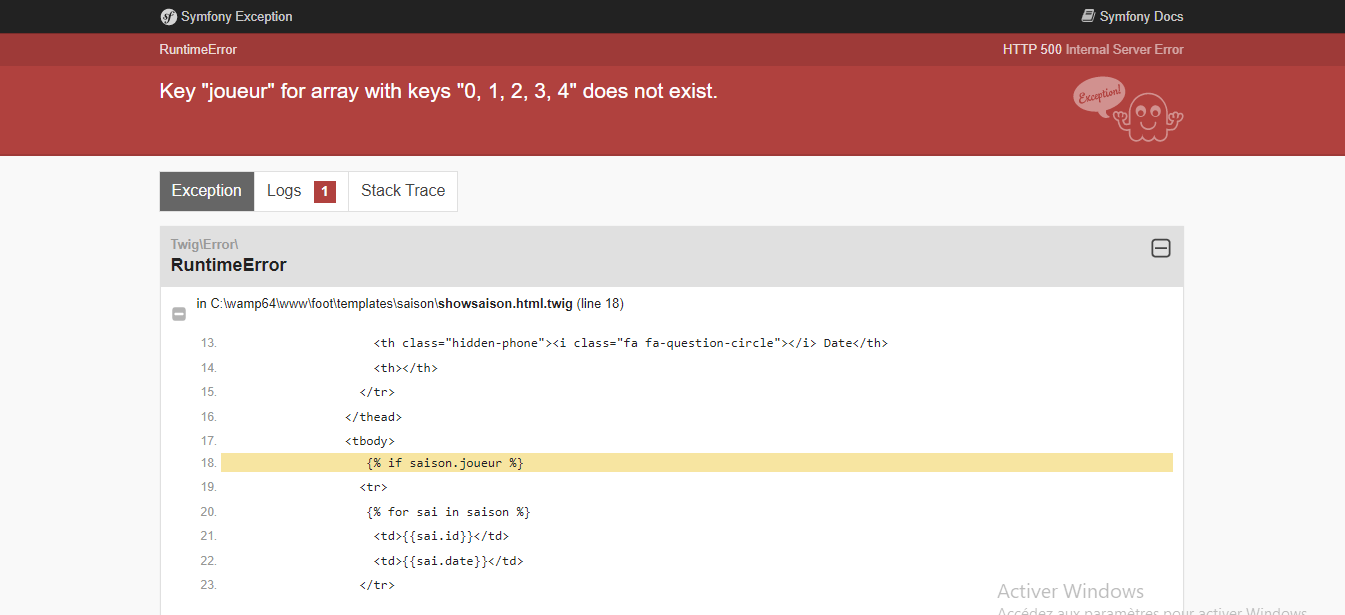


Figure 21 : L’exécution

Cmd :

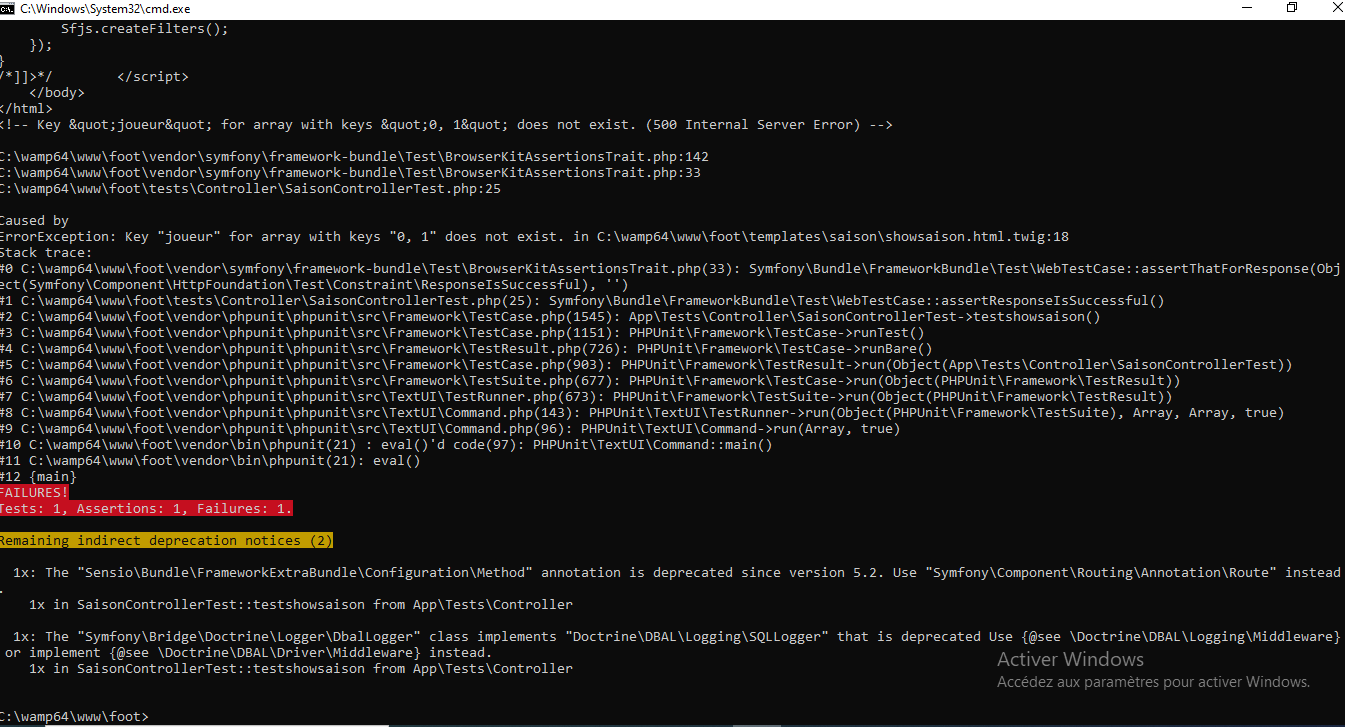


Figure 22 : Vérification de fonctionnalité de test

Fixtures :

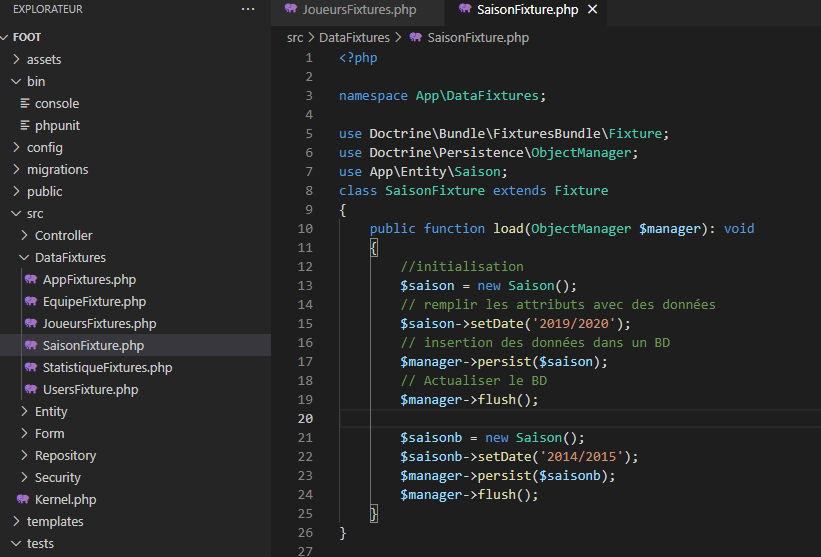


Figure 23 : Fixtures d'entité Saison

Base de données :

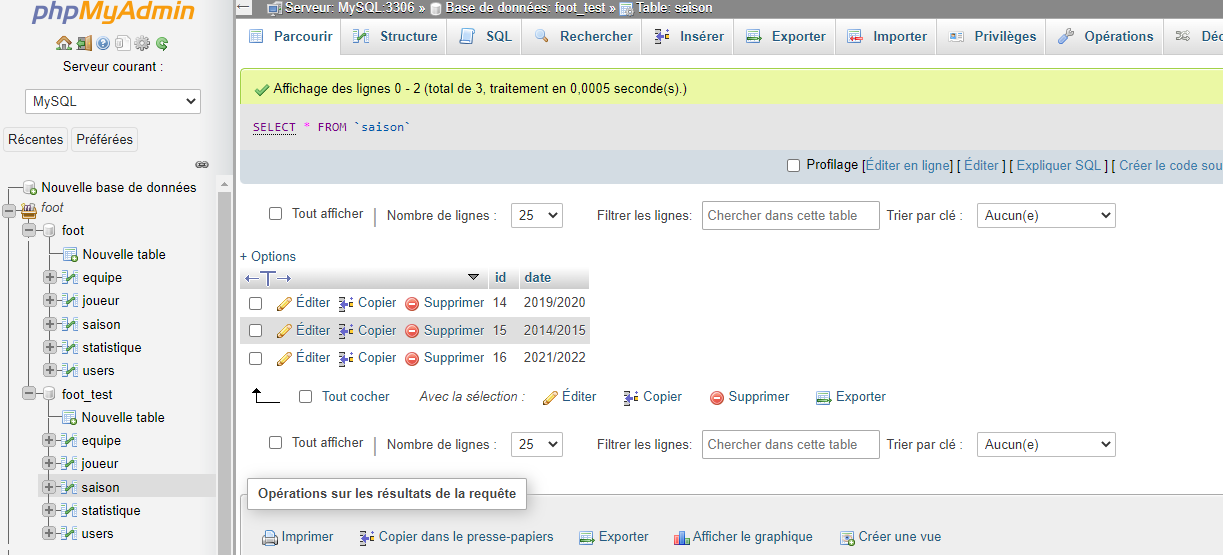


Figure 24 : Table "Saison" de base de données foot\_test

Fixtures des statistiques d'un joueur :

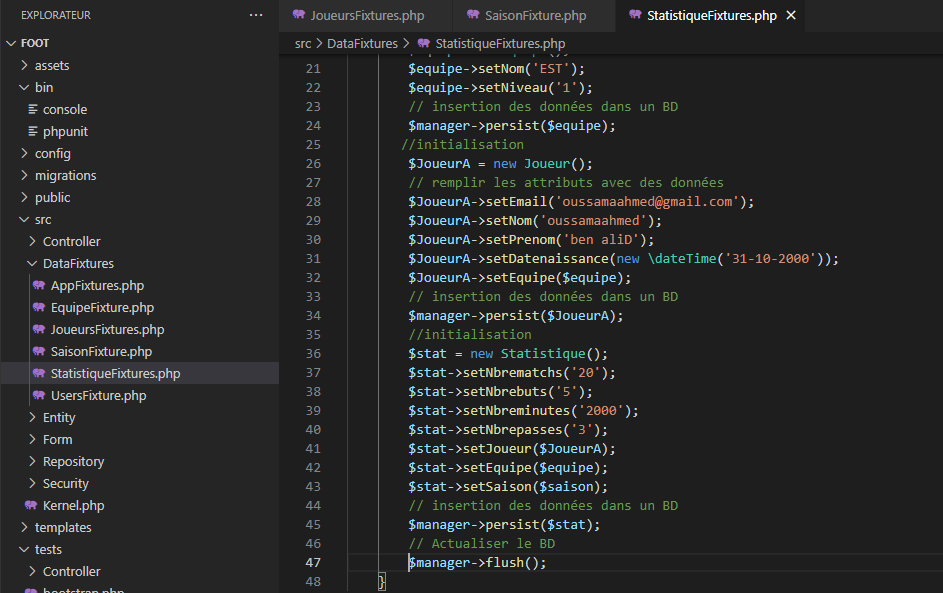


Figure 25 : Fixtures d'entité Statistique

Base de données :

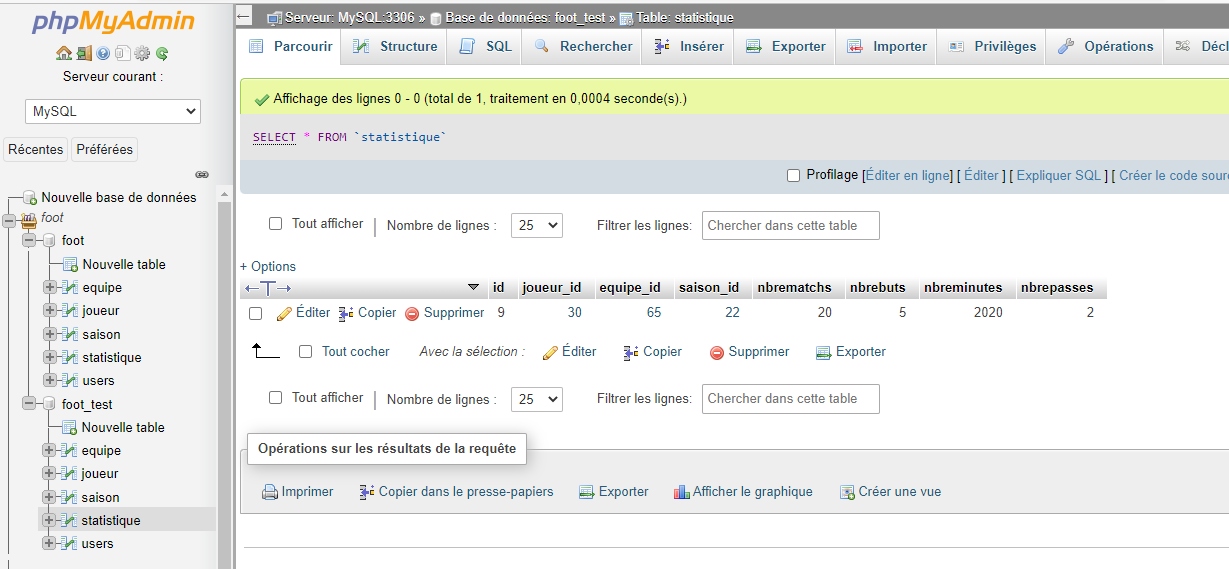


Figure 26 : Table "Statistique" de base de données foot\_test

Fixtures de l’entité users :

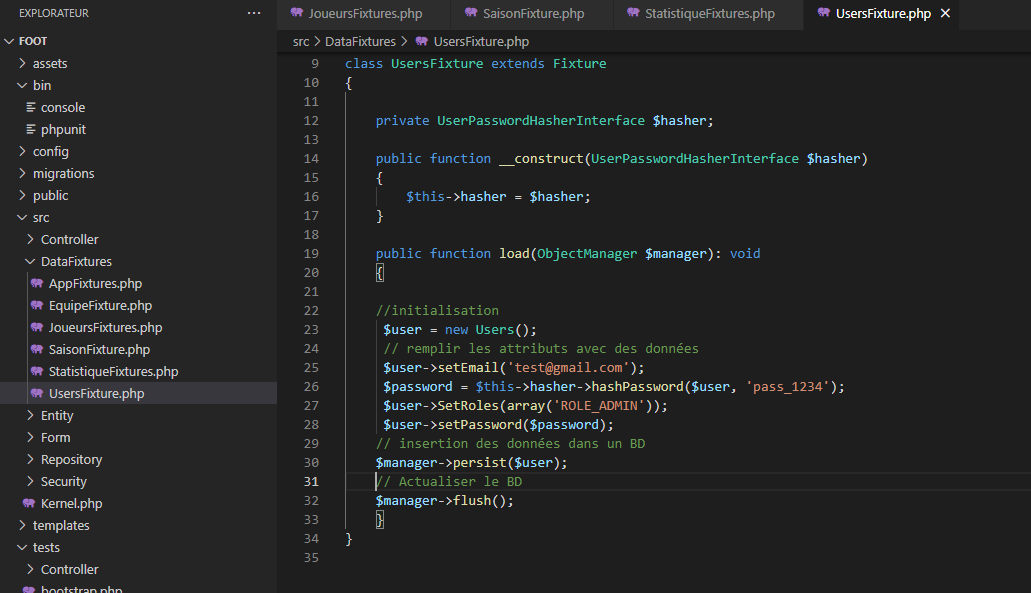


Figure 27 : Fixtures d'entité Users

Base de données :

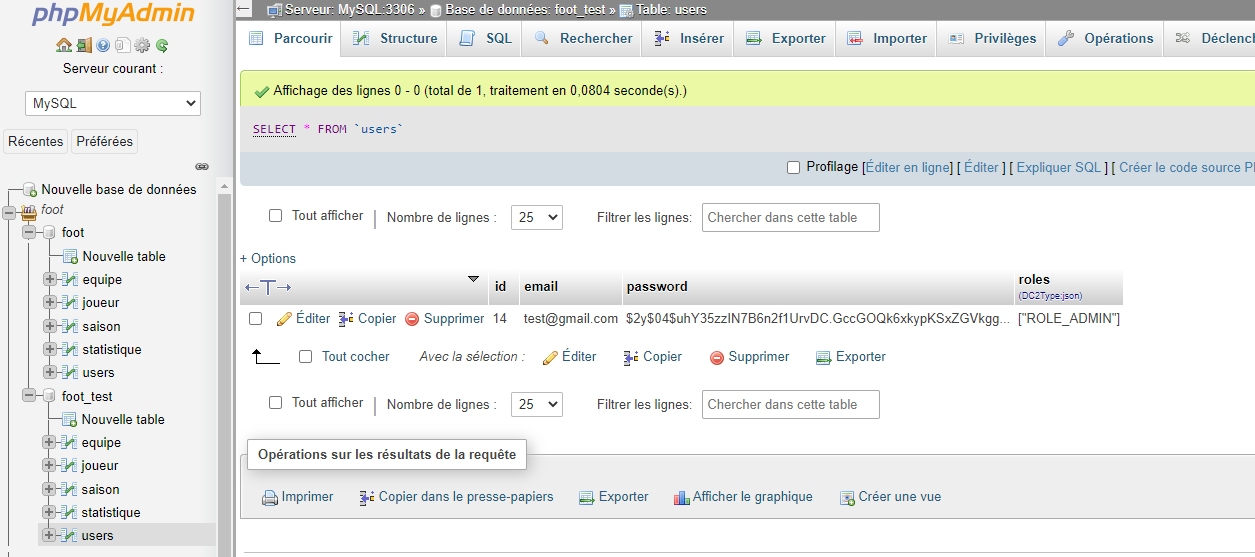


Figure 28 : Table "Users" de base de données foot\_test

**Etape 2 : l’évolution du projet**

1. **Présentation fonctionnelle**
2. **Identifications des acteurs :**

**L’administrateur  (Le dirigeant, Le gestionnaire) :**

L’administrateurpeut assurer les taches suivantes :

* S’authentifier
* Gérer les équipes
* Gestion des comptes des joueurs
* Gérer les statistiques des joueurs
* Gérer les saisons
* Gestion des comptes des gestionnaires des équipes

**Joueur :**

Le joueur peut assurer les taches suivantes :

* S’authentifier
* Gérer les statistiques
* Gérer le profil

1. **Besoins fonctionnelles :**

**L’administrateur  (Le dirigeant, Le gestionnaire) :**

* S’authentifier :

Le système affiche l’interface d’authentification : Dans ce cas, il faut saisir les coordonnées suivantes (identifiant et Mot de passe) puis cliquer sur le bouton « SE CONNECTER ».

* Gérer les équipes :

L'administrateur a le droit de créer un compte d’une équipe, consulter la liste des équipes et consulter les détails d’une équipe.

* Gestion des comptes des joueurs

L'administrateur à la possibilité de créer un compte d’un joueur, consulter la liste des joueurs.

* Gérer les statistiques des joueurs

L'administrateur a le droit d'ajouter la statistique d’un joueur et consulter les statistiques de chaque joueur par équipe pour une saison.

* Gérer les saisons

L'administrateur a le droit d'ajouter une saison et consulter la liste des saisons.

* Gestion des comptes des gestionnaires des équipes

L'administrateur a la possibilité d'ajouter des comptes des gestionnaires des équipes, consulter la liste.

**Le joueur :**

* S’authentifier :

Le système affiche l’interface d’authentification : Dans ce cas, il faut saisir les coordonnées suivantes (identifiant et Mot de passe) puis cliquer sur le bouton « SE CONNECTER ».

* Gérer les statistiques :

Le joueur a le droit d' renseigner et consulter leur statistique.

* Gérer le profil :

Le joueur a le droit de consulter son profil et de modifier leurs coordonnées.

1. **Etude conceptuelle :**
2. **Diagramme de cas d’utilisation de l’administrateur :**

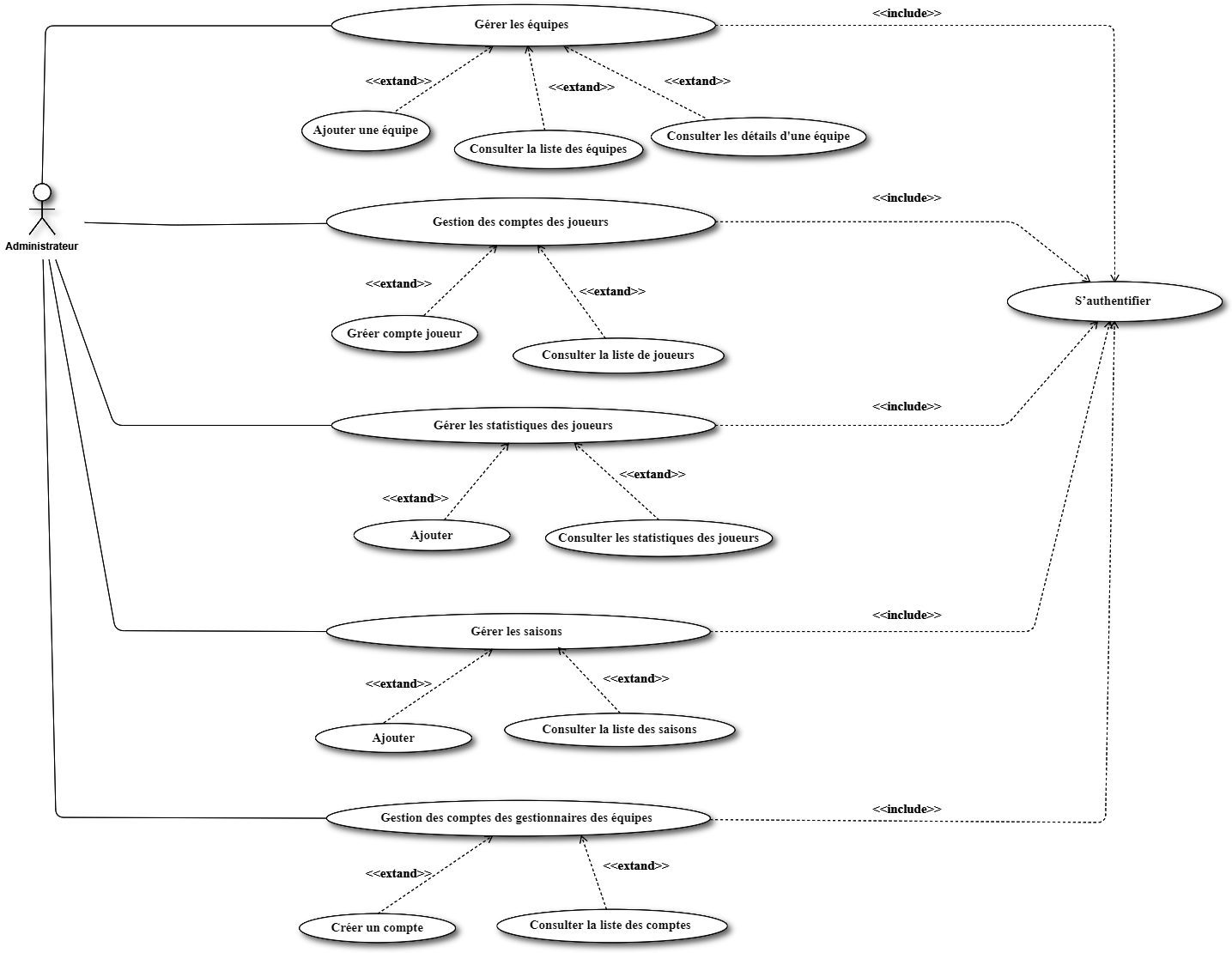


Figure 29 : Diagramme de cas d'utilisation de l'administrateur

1. **Diagramme de cas d’utilisation du joueur :**

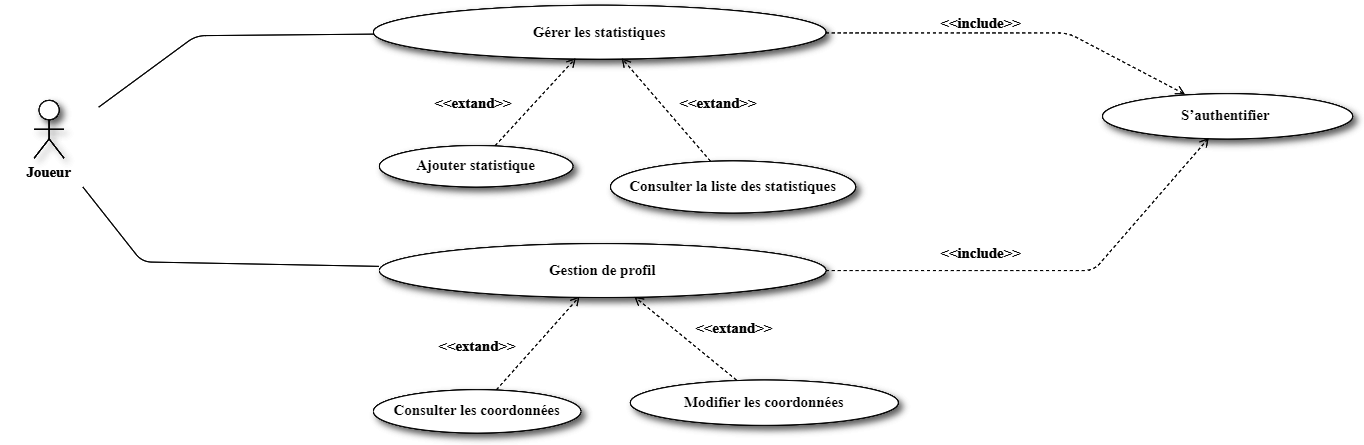


Figure 30 : Diagramme de cas d'utilisation du joueur

1. **Diagramme de classe :**

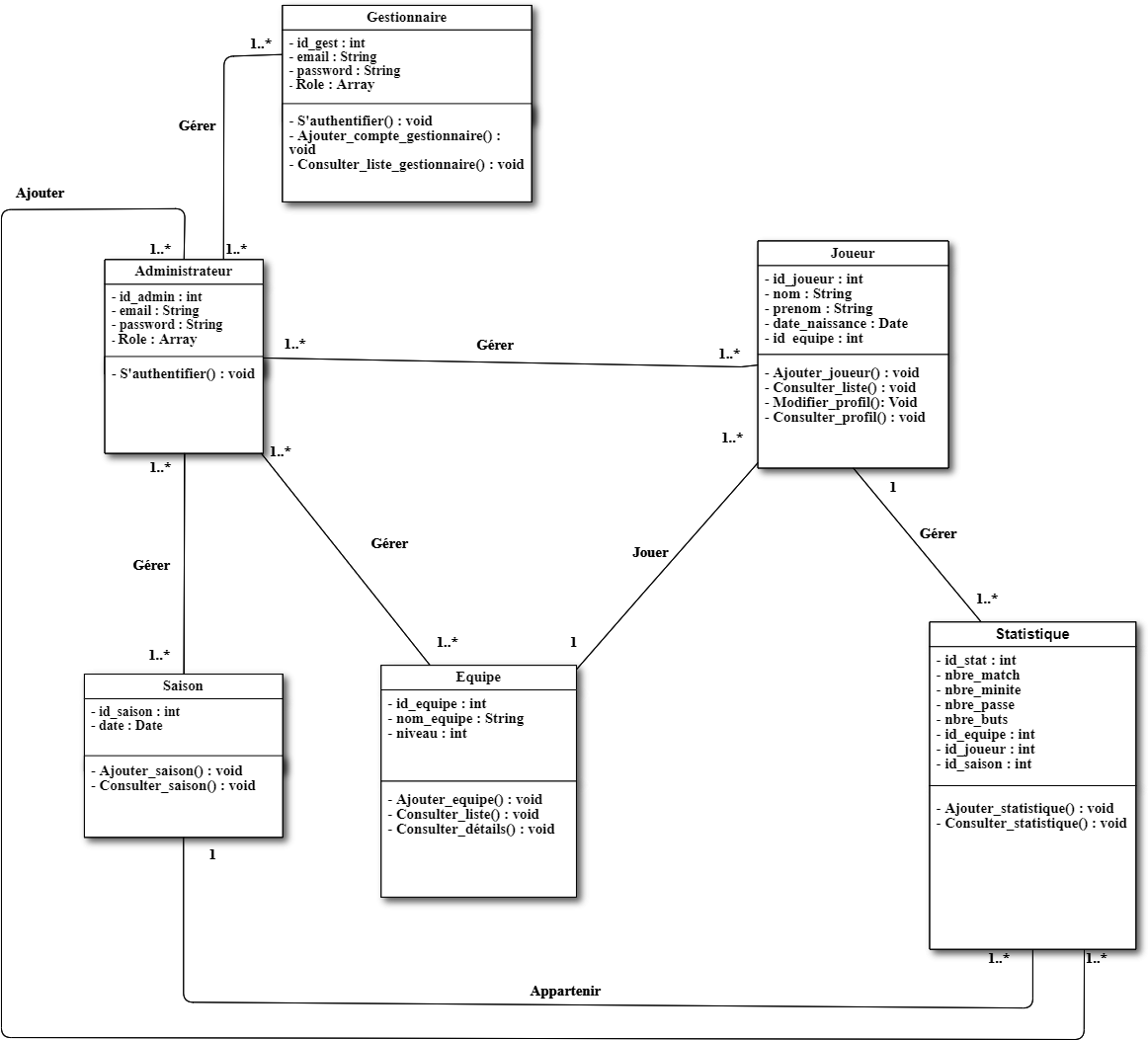


Figure 31 : Diagramme de classe

1. **Une architecture technique :**

**L’architecture logicielle :**

Dans cette application nous expliquons le choix de l’architecture logicielle du notre travail qui se base sur le modelé MVC.

Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

* Un modèle (Model) : Contient les données à afficher.
* Une vue (View) : Contient la présentation de l'interface graphique.
* Un contrôleur (Controller) : Contient la logique concernant les actions effectuées par l'utilisant.

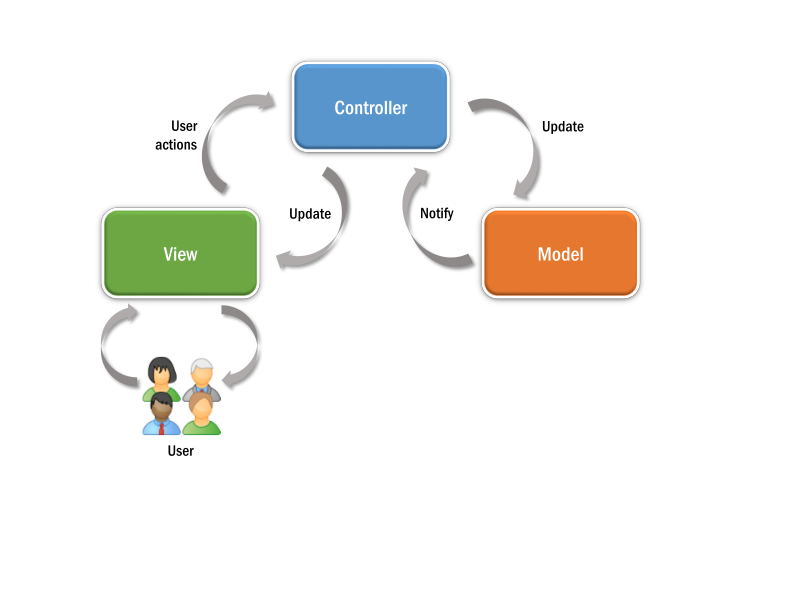


Figure 3 : L'architecture MVC

1. **Planning de l’évolution du service :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fonctionnalité** | **Acteur** | **Description** | **Priorité** | **Temps (jour)** |
| Gestion des comptes des gestionnaires des équipes | Administrateur | En tant qu'un administrateur, je souhaite ajouter des comptes des gestionnaires des équipes et consulter la liste des comptes. | 1 | 3 |
| Gestion des comptes des joueurs | Administrateur | En tant qu'un administrateur, je souhaite ajouter des comptes des joueurs et consulter la liste. | 1 | 3 |
| S’authentifier | Joueur | En tant qu'un Joueur, je souhaite s’authentifier | 1 | 2 |
| Gérer les statistiques | Joueur | En tant qu'un joueur, je souhaite des renseigner et consulter mon statistique | 1 | 3 |
| Gestion de profil | Joueur | En tant qu'un joueur, je souhaite gérer mes données personnelles (Modifier et consulter) | 2 | 2 |