# **Documentation Technique pour EnregDetailsTest.php**

## **Aperçu Général**

Ce fichier contient une classe de test unitaire EnregDetailsTest qui utilise PHPUnit pour tester la fonctionnalité du script EnregDetails.php. La classe teste principalement les validations d'entrée, les erreurs de base de données et les comportements attendus lors de l'enregistrement des détails d'un utilisateur (probablement dans un contexte sportif).

## **Structure de la Classe**

* Classe: EnregDetailsTest (extension de PHPUnit\Framework\TestCase)
* Attributs:
  + $mockPDO: Mock de la classe PDO
  + $mockStatement: Mock de la classe PDOStatement

## **Configuration des Tests**

La méthode setUp() est exécutée avant chaque test et:

* Définit la constante PHPUNIT\_TEST
* Réinitialise le tableau $\_SESSION
* Crée des mocks pour PDO et PDOStatement
* Configure les comportements par défaut des mocks
* Assigne le mock PDO à la variable globale $bdd

## **Méthodes Utilitaires**

* setPostData(array $overrides = []): Initialise le tableau $\_POST avec des valeurs par défaut et permet de les surcharger via un tableau

## **Cas de Tests**

### **1. testFormatNumeroLicenceInvalide()**

* Objectif: Vérifier la validation du format du numéro de licence
* Scénario: Soumet un numéro de licence trop court (moins de 6 caractères)
* Assertion: Vérifie que le message d'erreur contient "Erreur : Le numéro de licence"

### **2. testFormatCodePostalInvalide()**

* Objectif: Vérifier la validation du code postal
* Scénario: Soumet un code postal de seulement 3 chiffres
* Assertion: Vérifie que le message d'erreur contient "Erreur : Le code postal"

### **3. testNumeroTelephoneInvalide()**

* Objectif: Vérifier la validation du numéro de téléphone
* Scénario: Soumet un numéro de téléphone qui ne commence pas par 0
* Assertion: Vérifie que le message d'erreur contient "Erreur : Le numéro de téléphone"

### **4. testErreurPDOPrepare()**

* Objectif: Tester le comportement en cas d'erreur PDO lors de la préparation d'une requête
* Scénario: Configure le mock PDO pour lancer une exception PDOException
* Assertion: Vérifie que le message d'erreur contient "Erreur"

### **5. testDoubleInsertionMemeNumeroLicence()**

* Objectif: Vérifier le comportement lors de l'insertion d'un numéro de licence déjà existant
* Scénario: Configure le mock pour indiquer qu'un numéro de licence existe déjà
* Assertion: Vérifie que le message d'erreur contient "existe déjà"

### **6. testValidationDesDonnees()**

* Objectif: Tester la validation des champs obligatoires
* Scénario: Soumet un formulaire avec un numéro de licence vide
* Assertion: Vérifie que le message d'erreur contient "Erreur : Tous les champs obligatoires"

### **7. testRedirectionApresInsertion()**

* Objectif: Vérifier qu'une redirection se produit après une insertion réussie
* Scénario: Soumet un formulaire valide
* Assertion: Vérifie que la sortie contient "REDIRECT:"

### **8. testReglesMetier()**

* Objectif: Tester le comportement en cas d'échec d'insertion
* Scénario: Configure le mock PDO pour que la méthode exec retourne false
* Assertion: Vérifie que le message d'erreur contient "Erreur"

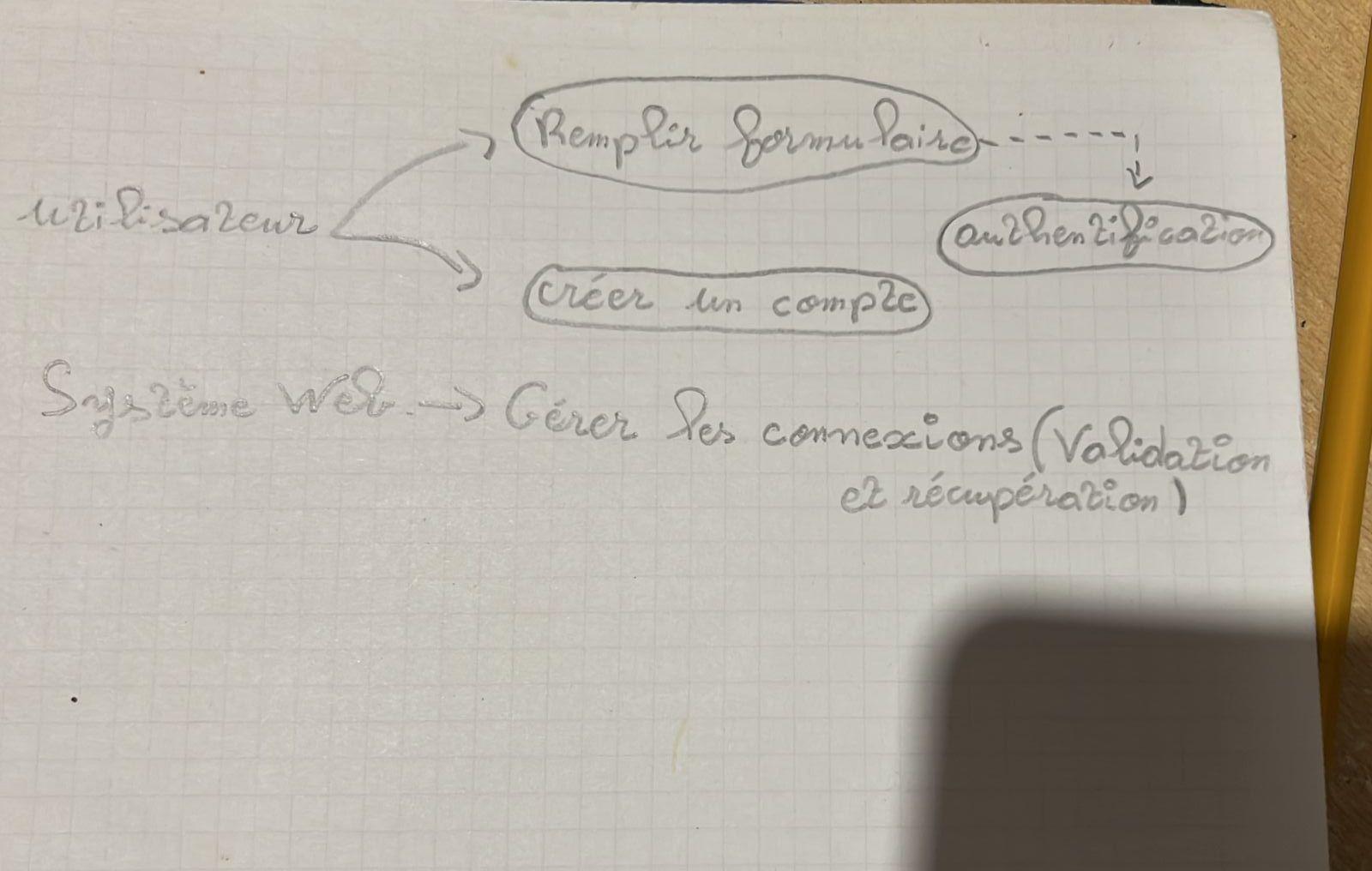
## **Dépendances**

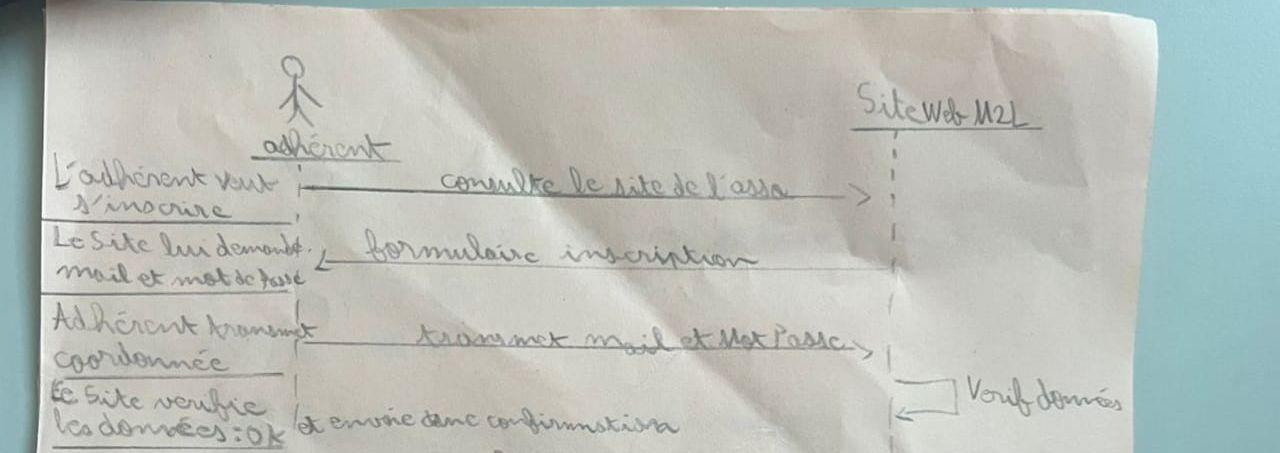
* PHPUnit
* PDO et PDOStatement pour les mock d'accès à la base de données
* Le script EnregDetails.php situé dans le répertoire ../HTML/

## **Notes Techniques**

* Utilise les mocks pour simuler l'interaction avec la base de données sans connexion réelle
* Capture la sortie du script via les fonctions ob\_start() et ob\_get\_clean()
* Les tests sont centrés sur la validation des entrées et la gestion des erreurs

Vous pouvez ajouter des titres (Format > Styles de paragraphe) qui apparaîtront dans votre table des matières.

Présentation des diagrammes



Présentation de la maquette de la base donnée :

