Évaluation 04 –

Développement d'un composant d'accès aux données

Dans le cadre du projet "zumba", cette évaluation va vous permettre de développer le composant d'accès aux données pour la table "adherents". Ce composant sera utilisé aussi bien sur le site internet pour enregistrer les pré-inscriptions que sur le site intranet pour gérer les adhérents. Vous respecterez toutes les bonnes pratiques du métier:
- programmation en code objet
- programmation défensive et validation des données
- protection contre les injections SQL
- protection contre les failles cross-site scripting (XSS)
- documentation du code
- respect des normes PSR-1 et PSR-2
- utilisation du design pattern CRUD

1 – Création de la table adhérent

Déterminer la structure de la table et fournir un script SQL permettant de créer la table create-adherent.sql

La fiche d'inscription contient les données suivantes :
- nom et prénom
- adresse, code postal, ville
- téléphone
- mail
- photo

Les mots de passe devront contenir au moins 8 caractères : au moins une majuscule, une minuscule, un chiffre et un caractère spécial. Le mot de passe devra être crypté.

2 – Sécurité

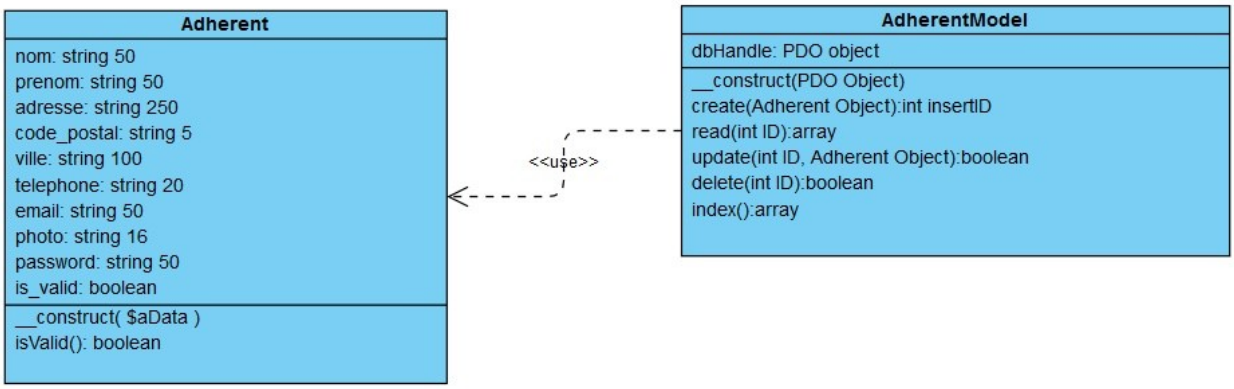
Consulter les documentations suivantes pour connaitre les recommendations de sécurité concernant les injections SQL et failles XSS.
https://fr.wikipedia.org/wiki/Cross-site_scripting
https://www.owasp.org/index.php/PHP_Top_5

3 – Formulaire HTML

Créer un formulaire basique qui vous permettra de tester vos composants. La mise en forme du formulaire et le contrôle des données en javascript ne font pas partie de cet exercice, car les contrôles sont réalisés côté back-end.

4 – Classe adhérent

Créer une classe adhérent disposant d'un constructeur permettant d'initialiser une instance en controlant et validant chacune des données fournies (failles XSS comprises). Le constructeur recevra en paramètre un tableau associatif avec comme clé les noms de champs (tableau $_POST provenant du formulaire).



Écrire un programme de test de la classe adhérent mettant en évidence des cas de classe valide et des cas de classes invalides testant toutes les erreurs possibles. Les cas valides serviront à tester le point 6.
Attention à ne pas oublier de vérifier la validité de la photo fournie (vérification du type jpg ou png et du format du fichier, vérification de la taille du fichier). Si vous disposez de temps à la fin de l'évaluation, veillez à redimensionner l'image en 500 pixels de large dans sa plus grande dimension.

5 – Création de la classe permettant la mise à jour de la BDD

Créer un composant CRUD permettant de mettre à jour la table adhérent en passant en paramètre un objet de la classe adhérent créé plus haut. Cet objet (AdherentModel.php) vous permettra de protéger votre application contre les injections SQL.

Vous vous inspirerez de la classe déjà crée dans le projet prodpoo que vous avez déjà étudié en cours (script model-typprod.php). Cet exemple est beaucoup plus simple que celui déjà étudié.

Écrire un programme de test mettant en oeuvre toutes les fonctionnalités de votre classe.

6- interface / protection des données personnelles

En cas de faille du site internet, afin d'exposer un minimum d'informations des adhérents, vous ne conserverez que les donnnées pas encore transmises au site intranet.

Pour cela vous développerez sur le site internet une interface disposant de 3 fonctions (index, delete, photo). L'interface vérifiera que l'accès est bien réalisé par le site intranet en vérifiant l'adresse IP du client, pour toutes les autres adresses IP vous retournerez le message "Hello world".

Vous écrirez le programme d'interface interface-adherent.php et un programme "client" test-interface.php

Les URL utilisées seront:

-> /interface-adherent.php?action=index

Cette fonction retournera un fichier JSON contenant toutes les données de pré-inscription enregistrées dans la table. Un champ 'id' permettra de spécifier une fiche particulière pour les 2 fonctions suivantes.

-> /interface-adherent.php?action=photo&id=1

Transfèrera la photo enregistrée pour une fiche spécifique

-> /interface-adherent.php?action=delete&id=1

Effacera une fiche spécifique et la photo associée.