Semaine 16 : Intégrer la POO dans une application MVC

Objectif :

Jusqu'à aujourd'hui vous avez pratiqué la POO au sein de micro-exercices qui vous ont permis d'apprendre les bases de ce paradigme de programmation. Le problème est que cela ne vous apprend pas à répondre à la demande d'un client grâce à la programmation orientée objet. De même, vous ne vous représentez peut-être pas encore en quoi la POO est d'une grande utilité lorsqu'on travail en groupe sur un projet. Cette semaine est là pour résoudre le problème. Car vous allez passer votre projet fil rouge du stade d'application procédurale au stade d'application objet organisée avec le pattern MVC.

Compétences acquises :

- Connaître le concept général de POO
- Savoir déclarer une classe
- Gérer la visibilité de ses attributs et méthodes
- Comprendre la portée des éléments
- Utiliser l'héritage
- Hydrater ses objets
- Typer ses arguments
- Réaliser une micro-application web en orientée objet.
- Organiser son projet selon le pattern MVC

Ressources et exercices :

Les ressources de la semaine dernière sont toujours valables. Celle spécifique sur le MVC vous aidera sûrement cette semaine :
- https://openclassrooms.com/en/courses/4670706-adoptez-une-architecture-mvc-en-php/4670713-pourquoi-faire-un-code-professionnel

Pour modéliser vos applications orientées objet, vous pouvez vous référer spécifiquement à cette partie du cours :
- https://openclassrooms.com/en/courses/1665806-programmez-en-oriente-objet-en-php/1667684-uml-presentation-1-2

Pour comprendre le principe des transactions SQL (attention il faudra encore voir comment les appliquer avec PDO) :
- https://openclassrooms.com/en/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1970063-transactions

Projet à rendre : Projet fil rouge

Maintenant que votre application est en ligne, de nombreux bugs et demandes d'amélioration ont été remontés à votre product owner. La squad projet s'est déjà attelée à leur résolution mais le travail est plus compliqué que prévu. Il est difficile d'intervenir dans le code mais aussi de comprendre l'organisation de l'application et l'implémentation des fonctionnalités.

Une réunion a donc été organisée avec l'équipe projet par le scrum master et vous avez décidé de faire évoluer l'application vers une application objet organisée avec le pattern MVC.

Pas de nouvelles fonctionnalités mais votre code doit être écrit en orienté objet partout où cela est possible et vous refactorez l'application entre model view et controller. Vous vous concentrez sur les opérations du CRUD, en plus de la connexion utilisateur :

- Lire les comptes
- Lire un seul compte
- Créer un compte
- Mettre à jour un compte (débit/crédit)
- Supprimer un compte

En ce qui concerne les requêtes SQL, vous essayer de mettre en place les transactions à minima sur l'opération de débit et de crédit.

Cela signifie que vous réalisez aussi un diagramme de classes et un UseCase. Afin de gérer également l'avancée du code au mieux vous continuer d'utiliser le Kanban précédemment réalisé et que bien entendu vous avez toujours maintenant à jour ;-)

Le rendu se fera via Teams avant dimanche soir minuit dans le dossier rendu vous déposerez un fichier txt à votre nom avec le lien vers votre repository GitHub qui contiendra vos exercices.

Vous mettez également votre projet en ligne chez un hébergeur.

Pour aller plus loin :

- Intégrez une nouvelle table dans votre application qui aura pour objectif de stocker les types de comptes bancaires autorisés et utilisez cette table pour tout ce qui concerne les types de comptes.

- Parmi les utilisateur intégrez le rôle conseiller. En effet tous vos utilisateurs sont pour l'instant des utilisateurs lambda mais normalement l'application serait également accessible aux banquiers. Ces derniers ont une liste de client dont ils peuvent accéder aux comptes et aux informations.

- Proposez à l'utilisateur connecté de pouvoir modifier les informations de son profil via un formulaire. Certaines informations comme son age ne sont pas modifiables mais l'adresse par exemple peut faire l'objet d'un changement.