

INFO732 - Application pour les étudiants

Maverick PERROLLAZ - Romain SOUCHON - Gabriel FORRAY - Zied GOBJI -
Quentin PASCAL

Problème :

Création d'une application logicielle permettant le partage de biens et de services, destinée à l'usage des étudiants d'un campus.

Lien du GitHub : https://github.com/M4verickFr/INFO732_TheStudentCorner

Objectif du projet :

Création de l'application TheStudentCorner, son objectif est de permettre l'échange de services / de biens entre étudiants.

Chaque étudiant indique ce qu'il recherche / ce qu'il sait faire. Il y aura la possibilité de faire une proposition à un autre étudiant, cet autre étudiant pourra accepter / modifier / supprimer la demande.

Quand un échange est terminé, ils pourront se laisser un avis, afin d'augmenter l'attrait de leurs profils.

Conception :

Avant de coder notre projet, nous sommes passés par une phase de conception durant laquelle nous avons établi des user stories et avons partagé nos idées afin de proposer une structure claire de l'application. Nous avons utilisé le service LucidChart qui permet de réaliser des diagrammes de cas d'utilisations, de séquences et de classes. Afin de faciliter l'interaction entre les utilisateurs, il faut qu'ils soient alertés quand quelqu'un interagit avec eux. C'est pourquoi nous avons choisi d'implémenter un patron observateur qui aura pour but de notifier les utilisateurs des actions les concernant.

Implémentation :

Pour l'implémentation, nous avons choisi de développer un site via un serveur Web utilisant PHP ainsi qu'une base de données MySQL. Pour gagner du temps sur le développement, nous avons utilisé le modèle architectural MVC qui permet de séparer notre application en trois composants : le modèle, la vue et le contrôleur.

Durant le développement, il nous a fallu un peu de temps afin de tous maîtriser le modèle MVC. Mais cela nous a permis de mettre en place les premières fonctionnalités de notre application, telle que la connexion, la création de compte, ou encore la création de proposition entre utilisateurs.

Une partie de l'équipe s'est occupée de créer des données afin de pouvoir réaliser des tests au sein de l'application ainsi qu'une première démonstration.

Implémentation Patron Observateur :

Nous avons choisi d'implémenter le Patron Observateur, qui permet d'éviter la dépendance inter-objets. Lorsqu'il y a une action sur une classe Observable, les classes Observer en sont informées et peuvent effectuer une action. Ainsi, nous avons conçu un observable nommé **Mailer**, celui-ci permet de prévenir l'administrateur lors de certaines actions sur le site et un observer **onUserAdded** qui réagit lorsqu'un nouvel utilisateur s'enregistre sur notre application. Nous n'avons pas développé la fonction **sendMail** par manque de temps, mais le principe du Patron Observateur est fonctionnel. Vous pouvez le retrouver l'observateur dans le fichier "mailer.php" et la classe observable dans le fichier "controller/UserController.php".

Répartition des tâches :

Prénom	Diagramme des cas d'utilisation	Diagramme de séquences	Diagramme de classe	Script SQL	Implémentation PHP
Maverick					
Gabriel					
Romain					
Zied					
Quentin					