P\_Web2 – Rapport de travail – Librairie en ligne



# Introduction

Le but du projet est de créer un site web de librairie en ligne par groupe de 4. Du point de vue de client, le site web doit contenir une page d’accueil expliquant à quoi sert le site ainsi que les cinq derniers livres ajoutés. Les livres sont ajoutés par un utilisateur depuis le site, qui sera ensuite sauvegardé dans la base de données. Pour avoir accès à cette fonctionnalité, l’utilisateur doit se connecter via un système d’authentification sécurisé face aux injections SQL. L’utilisateur **connecté** pourra donner son avis sur les livres qu’il a lu en leur donnant une note entre 1 et 5 ainsi qu’un accès illimité aux liens dans les catégories, où les utilisateurs non-connectés n’auront pas accès à la totalité de la page catégorie. Les utilisateurs admin du site auront accès à tout, en plus d’un moyen de modifier et supprimer les livres ajoutés par les utilisateurs connectés.

# Analyse

## Réalisation et mise en page HTML

Avant de commencer la mise en page HTML, il a été décidé de se focaliser en premier lieu sur le design du site grâce à Figma et la mise en place de la base de données avec Looping en alternance. Une fois cela fait, il a été décidé de partir sur l’HTML. Une conversion en php a été faite par la suite.

## CSS

Nous avons au début décidé de créer un css commun, mais c’était très compliqué à mettre en place au début, nous avons donc décidé pendant le projet de nous faire chacun un css pour les pages que nous utilisions. Avant les vacances de noël, Leonar a réussi à rassembler les css, ce qui a optimisé grandement le code.

## Base de données à réaliser

### MCD

Pour créer le MCD, nous avons tenu compte des trois formes normales pour créer une base de données solide et saine avec le moins de redondance possible pour la suite du travail. Par exemple Dans cette base de données, un utilisateur peut noter et partager (ajouter) un livre à la base de données. Pour l’ajouter, il faut que l’utilisateur fournisse le titre, un résumé, le nombre de pages, l’auteur, l’éditeur, l’année d’édition et la couverture du livre sous forme de lien. Chaque livre appartient à une catégorie qui correspond au genre de livre (policier, SF). Il est ensuite possible pour un utilisateur de noter tous les livres ajoutés. La base de données prendra tous les avis des utilisateurs et en fera une moyenne à afficher sur le site, si possible avec un système d’étoiles. (voir image de la page suivante). La table writer a finalement été supprimée pour être incorporée dans book.

Une image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquement



### MLD

Généré grâce au MCD préparé ci-dessus. La connexion rate devient la table associative "t\_rate" en prenant les clés étrangères de "t\_user" et de "t\_book". Les 3 autres associations disparaissent grâce aux cardinalités 1,1 de share, write et belong.

### MPD

## Structure du code

Il a été décidé de mettre toutes les fonctions dans un seul fichier qui sera ensuite lié à tous les autres fichiers grâce à la classe database. Nous nous sommes basés sur la structure vue dans l’exercice du module 151 qui nous a accompagné pendant tout ce temps. Cela nous a permis de prendre un bon départ en commençant assez vite à appliquer les fonctionnalités que nous n’avions pas forcément eu le temps de voir comme l’authentification par exemple.

### Schéma UML

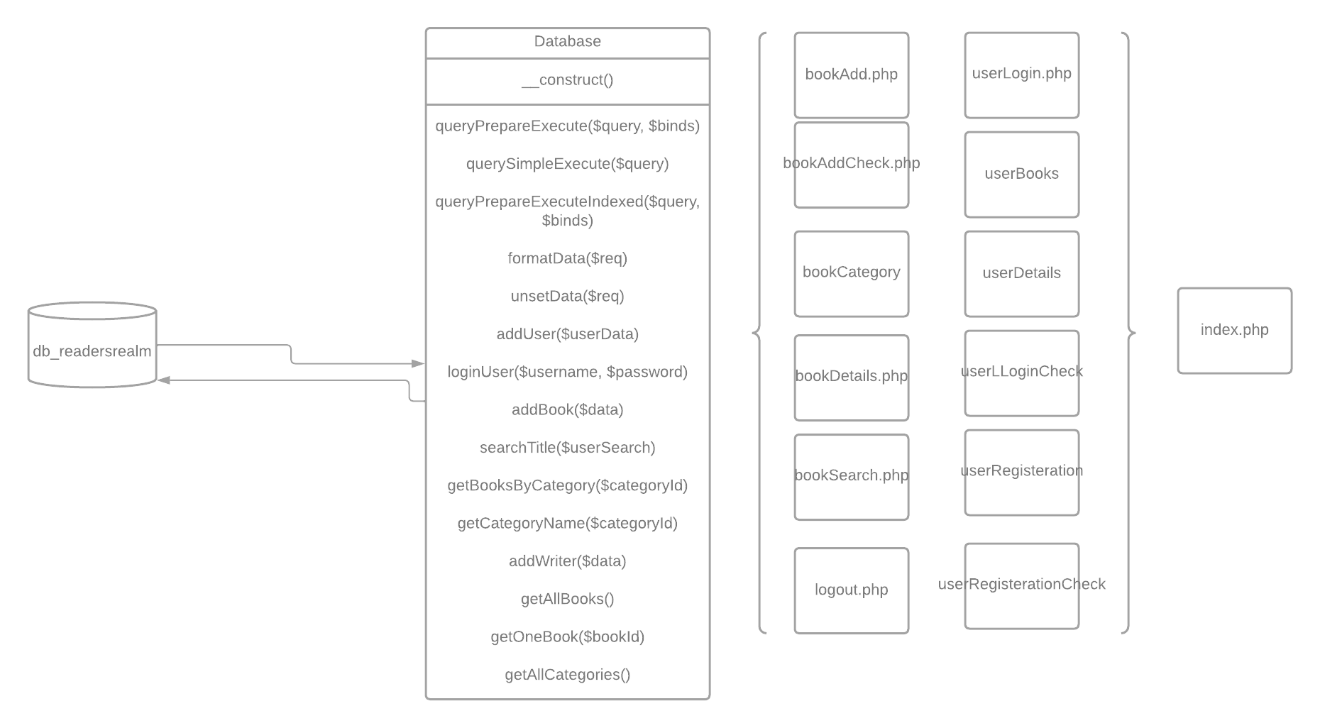




Figure - Schéma UML expliquant la structure du code

# Réalisation

## Mise en place

Séparation en 2 groupes de 2. Deux pour la création du MCD et du MLD ainsi que la rédaction du rapport, et les deux autres sur Figma pour créer la maquette du site internet. Chaque développeur s’est vu affecter une page à réaliser en php ou HTML avec un css attitré à chaque élève, qui si le temps ne manque pas, seront réunis par la suite.

Chaque élève a commencé sur une page différente. Un a commencé la page template sur laquelle toutes les autres s’appuieront. Cette page HTML contient aussi un css général pour toutes les pages qui ont des choses en commun, comme la navigation, le fond ou encore le footer. Un autre se met sur le formulaire de création de livre tandis qu’un autre crée la page de recherche de livre pendant que le dernier crée la page de détails des livres.

## Mesures de sécurité

Barre de recherche immunisée aux injections SQL grâce à la fonction « queryPrepareExecute » qui a été créée dans la classe database.

## Calcul de l’appréciation moyenne d’un ouvrage

Le programme prend les notes de chaque utilisateur pour un ouvrage et en fait la moyenne. Nous n’avons pas eu le temps de l’implémenter.

## Manuel d’utilisation

Un compte admin et lambda sont déjà ajoutés

#### Compte admin

ID : admin 123admin

#### Compte lambda

ID : usertest 1234test

### Accéder à la base de données

Nous utilisons uWamp pour se connecter à la db. L’identifiant et mot de passe de phpmyadmin sont tous les 2 root. (Sans le point).

### Créer un compte lambda

Créer un compte sur le site en allant dans l’onglet connexion puis s’inscrire, vous serez renvoyé à l’accueil, où il faudra recliquer sur connexion pour vous connecter. Une fois cela fait, vous aurez accès aux services du site

### Créer un compte admin

Même opération que pour le compte normal, mais il faut aller dans la base de données et tourner la valeur isAdmin à 1 pour être un utilisateur admin. Cet utilisateur pourra modifier et supprimer un livre même si ce n’est pas lui qui l’a créé

### Se connecter à un compte lambda

Pour se connecter à un compte normal. Il faut faire exactement pour le compte admin, sauf qu’aucune action n’est prise dans la base de données.

### Ajouter un ouvrage

Le formulaire est à remplir où on doit donner le titre, la catégorie, le nombre de pages, un extrait en lien pdf, le synopsis, le nom et prénom de l’auteur, le nom de l’éditeur, l’année d’édition et la couverture de l’ouvrage. Une fois envoyé, le livre sera enregistré dans la base de données et sera ajouté à la page de d’accueil de l’utilisateur qui a rajouté le livre.

### Modifier un ouvrage

Sur la page de l’utilisateur, il voit ses livres. S’il clique sur modifier, il voit le formulaire ci-dessus mais préremplies avec les informations existantes que l’utilisateur peut modifier à sa guise par la suite avant de les enregistrer.

### Supprimer un ouvrage

Sur la page de l’utilisateur, l’utilisateur a le pouvoir de supprimer les livres qu’il a ajouté

### Ajouter une appréciation

### Rechercher un ouvrage

Taper dans la barre de recherche le nom de l’ouvrage et appuyer sur entrée. L’ouvrage en question sera affiché s’il existe déjà dans la base de données, sinon un message d’erreur s’affichera. On pourra voir la moyenne des notes données par les utilisateurs.

### Lancer nos tests unitaires

Avoir installé bash et composer (composer require –dev phpunit/phpunit ^10), aller dans le dossier existant test dans le terminal bash, puis taper $vendor/bin/phpunit tests

# Test

Liste des tests réalisés :

* Création d’un compte lambda.
* Création d’un compte via la base de données pour un compte administrateur
* Suppression d’un compte
* Création d’un livre test
* Modification du livre test du dessus
* Suppression d’un livre test

## Explication pour chaque test

Pour la création d’un compte, il y’a eu quelques problèmes qui ont été réglés. Le test consistait à ajouter l’adresse mail, le pseudo et le mot de passe. Le mot de passe était caché mais on peut l’afficher en cliquant sur l’œil situé dans le champ. Mais en cliquant sur enregistrer, un message d’erreur disait qu’une certaine valeur ne pouvait pas être null. C’était juste un problème d’inattention dans le code qui fut vite corrigée.

Pour la création d’un compte admin, on a testé en créant le compte directement dans la db, mais en faisant cela les mots de passe étaient hachés avec le mot de passe décidé, ce qui rendait impossible de connaître le mot de passe attendu par la base de données. Pour contourner le problème, on a décidé de créer un compte normal et de lui accorder les droits grâce à la valeur isAdmin depuis la db.

Pour la suppression d’un compte, tenter de supprimer ne fonctionnait pas, car il fallait déclarer une valeur dans le script javascript sur lequel nous nous sommes inspirés (script du fil rouge).

Pour la création d’un livre test, il y’a eu quelques difficultés qui nous a un peu ralenti; comme nous avions décidé de retirer fk\_writer dans le livre pour juste créer une valeur bookWriter afin de faciliter le code il fallait réécrire une petite partie du code mais dans plusieurs endroits sans oublier changer cela dans la base de données où il fallut rajouter des entrées.

Nous avons aussi créé des tests unitaires afin de s’assurer que l’application fonctionne comme prévu. C’était délicat à mettre en place mais nous avons fini par réussir à y arriver en créant un test qui regarde si au moins une entrée est dans la base de données.

# Conclusion

## Générale

Le projet a été difficile à mettre en place pour plusieurs raisons, comme par exemple le fait que nous travaillions à quatre sur un même projet sur GitHub. Nous avions déjà fait des projets mais en équipes plus réduites et pas sur GitHub. La prise en main n’a pas été simple et n’est pas encore totalement acquise. Il y’a aussi eu des problèmes avec la base de données à cause de l’import de phpMyAdmin qui rajoutait des lignes de codes indésirables dans le SQL qui faisait bugger notre script. Cela a créé des conflits sur github et des tensions dans le groupe où chacun faisait à peu près ce qu’il voulait sans prendre en compte ou même avertir les camarades, ce qui a irrémédiablement causé des conflits sur les commits github, ou encore la recréation d’une base de données au moins cinq fois, ce qui nous a fait perdre de précieuses heures. Nous avons néanmoins réussi à nous organiser vers la fin du projet où nous nous sommes vraiment mis à avancer.

## Personnelle

### Harun

Je suis plutôt content de résultat de ce projet. Malgré ne pas avoir beaucoup contribué au côté php autant que je voulais, cela m'a permis de revoir les notions que je n'arrivais pas à bien comprendre. Dans le groupe, il y avait pas mal de tensions à la fois, mais jamais des disputes entre les membres. On a su surmonter et régler les problèmes.

### Leonar

Le sujet était bon et nous a motivé dès le début, mais peut-être un peu trop car tout le monde voulait mettre du sien le plus rapidement possible, et cela a évidemment créé des conflits car nous voulions travailler sur les mêmes éléments une fois figma et les schémas terminés. Nous ne nous sommes pas assez concertés entre nous pour savoir qui fait quoi.

### Théo

D’abord j’avais un élan motivation, puis quand nous avons commencé à devoir coder,j’ai été frustré de ne pas avancer car nous n’arrêtions pas de se marcher dessus. Vers la fin du projet, il y’a eu un élan de motivation qui nous a permis à tous les 4 de nous reprendre et c’est à ce moment-là que nous avons réussi à nous répartir efficacement les tâches et à avancer.

### Yann

Le sujet était réaliste et facile à se représenter et j’ai d’abord été motivé avant d’être totalement l’inverse dû à notre manque de communication écrit plus haut, en ne faisant quasiment qu’alimenter le rapport avant de récupérer cette motivation vers la fin du projet.

## Constructive

#### Planification

Nous avons fait une planification rudimentaire entre nous et sur github dans les issues. Nous avions planifié de finir le figma en 2 leçons (1 semaine), mais nous en avons pris 4, car nous nous sommes ralentis au niveau du design pour avoir quelque chose qui nous plaisait à tous. Pendant que 2 s’occupaient du design, 1 s’occupait exclusivement des schémas et de la base de données tandis que le dernier faisait la navette entre les schémas et le rapport. Par la suite nous nous sommes moins concertés et n’avons pas

Nous aurions dû mieux communiquer sur ce que l’on faisait avec nos camarades, de notifier pour chaque changement pour éviter de mauvaises surprises à nos camarades

# Webographie

[StackOverflow](https://stackoverflow.com/questions/16403221/creating-a-php-search-bar)

[Site web php](https://www.php.net/)

[Enseignement section](https://enseignement.section-inf.ch/moduleICT/151/Index.html)

[openclassrooms](https://openclassrooms.com/fr/courses/4087056-testez-unitairement-votre-application-php-symfony/7828665-faites-vos-premiers-pas-avec-phpunit-et-les-tests-unitaires)

[github](https://github.com)