



RAPPORT PROJET - APPLICATION WEB JEE - LA LIBRAIRIE OHARA

JOUSSELIN HUGO

LAMY MARTIN

PAESA THÉO

ROUJAC SARAH

Année universitaire d'ING2 2022/2023

23 décembre 2022



Table des matières

1	Introduction	2
2	Démarche de la conception	2
3	Fonctionnalités & Technicité :	3
4	Problèmes rencontrés	4
5	Pistes d'améliorations	4

1 Introduction

Comme nous l'avons déjà réalisé maintes fois au cours de notre scolarité, ce projet consistait à la création d'un site web. A la différence que cette fois-ci, ce n'était non pas un site, mais une application web en utilisant l'architecture JEE. Nous avons donc été amené à utiliser les Frameworks que nous avons étudié en cours tel que : SpringBoot et Hibernate.

Après une première réunion sur les goûts de chaque membre du groupe, nous avons décidé que notre application serait basée sur l'univers des mangas. Au vu des différentes fonctionnalités attendues, nous sommes partis sur une application web d'une librairie de mangas : la librairie Ohara.

2 Démarche de la conception

Notre application devait contenir les différentes couches :

- la couche persistance (entities),
- la couche accès aux données (DAO),
- la couche service (métier),
- la couche Web.

Comme bonne pratique, nous avons commencé par la conception de la BDD (Base De Données), en utilisant les relations **OneToOne** & **ManyToOne**.

Une fois la conception de la BDD validée, nous sommes passés à la création du projet, en le découpant en plusieurs phases.

Tout d'abord, nous avons mis en place **la couche persistance**, soit la configuration et la création de toutes les entités de l'application.

Puis, nous avons commencé à réfléchir comment implémenter, dans **la couche DAO**, **les méthodes CRUD** (create, read, update, delete) puis par la suite, nous avons codé nos propres méthodes pour manipuler les données.

Ensuite, dans **la couche métier/service**, nous avons implémenté la **logique métier**, c'est-à-dire la partie fonctionnelle de notre application.

Enfin, nous avons utilisé le **framework Spring MVC** pour implémenter **la couche web**.

3 Fonctionnalités & Technicité :

Nous avons fait en sorte que notre application contienne plusieurs pages et que la navigation entre les pages soit simple et bien orientée.

L'application permet :

- d'insérer une entité en BDD, c'est à dire d'ajouter des mangas à notre catalogue.
- de retrouver une entité en BDD, c'est-à-dire de rechercher des mangas dans notre catalogue.
- de lister des entités en BDD, c'est-à-dire de visualiser tous nos mangas disponibles.
- de mettre à jour une entité en BDD, c'est-à-dire de modifier les informations de nos mangas du catalogue.
- de supprimer une entité en BDD, c'est-à-dire de supprimer des mangas de notre catalogue.

Nous avons également implémenté des rôles afin que des utilisateurs différents puissent effectuer des actions différentes :

- Des utilisateurs vont simplement pouvoir acheter des mangas ;
- Des modérateurs vont pouvoir gérer le catalogue du site et les stocks ;
- Des administrateurs vont pouvoir gérer les modérateurs du site.

Les modérateurs et les administrateurs peuvent également acheter des mangas.

Les contrôleurs utilisent les méthodes **HTTP : GET et POST**.

Les contrôleurs reçoivent des infos dynamiques dans l'url (ex : /posts/654324) et dans le body pour les requêtes POST.

L'application utilise le design pattern DAO pour chaque accès à la BDD.

Chaque vue manipule des données transmises par son contrôleur.

Le code source est livré dans un repo Github.

Le repo contient un war (ou jar de l'application).

Nous avons des commits assez régulier seulement tous postés à partir d'un seul ordinateur puisque nous travaillions sur un seul ordinateur au début pour que tout le monde comprenne bien.

4 Problèmes rencontrés

Nous avons rencontré quelques problèmes :

- Lors de nos premières requêtes, il a été compliqué de comprendre comment marchaient les `@GetMapping` et les `@PostMapping`. Nous avons finalement réussi après plusieurs jours de recherches.
- Pour ajouter des produits dans le panier. Au début, on ne pouvait pas ajouter plus d'un produit dans notre panier. Nous avons oublié d'appeler la méthode sauvegarder dans la méthode du contrôleur. Nous avons mis beaucoup de temps pour nous en rendre compte.
- Nous avons tenté au début d'utiliser thymeleaf mais il est impossible de le faire fonctionner à cause de différences dans les versions. Nous avons donc du utiliser des JSP.

5 Pistes d'améliorations

Nous pourrions opter pour un meilleur CSS via par exemple un front-end utilisant le framework Bootstrap. Nous pourrions également rajouter les couvertures de nos mangas pour rendre nos produits plus vendeurs voir encore mieux, permettre au client de lire les premières pages de ces derniers.