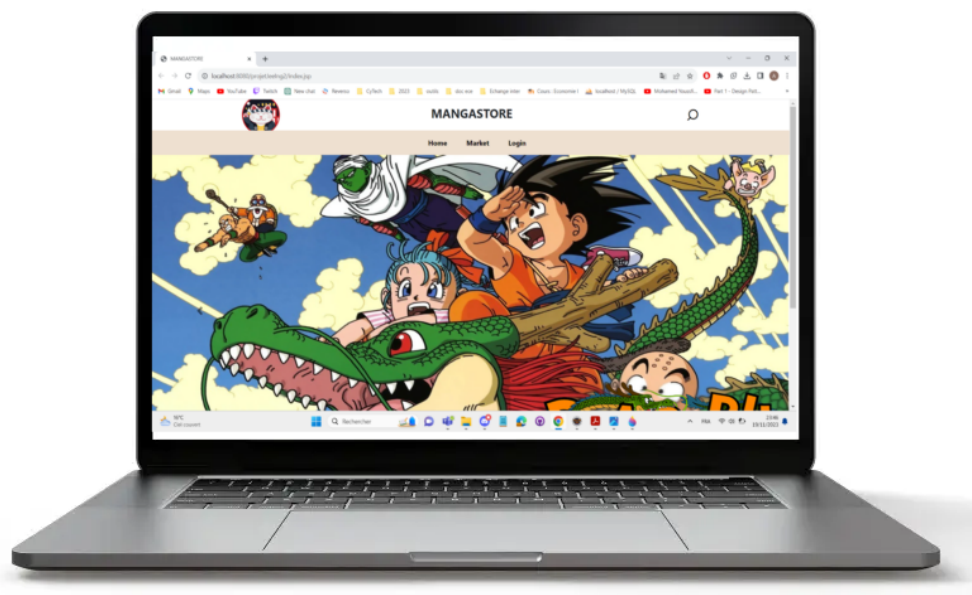


HADDACHE Noaman
CAI Christophe
LEMBA Mohamed
OUAALI Abd-Ennour

ING2 GSI groupe 1
2023-2024

Rapport

Projet JEE



Sommaire

Introduction :	3
Organisation :	4
Diagramme :	5
Structure de la base de données & MCD :	6
Problèmes et solutions :	7
Limites & axes d'amélioration :	9
Conclusion	10

Introduction :

Dans le cadre du module Développement Distribué Java EE, nous avons eu à réaliser un projet permettant de manipuler les outils acquis durant ce cours. En effet, ce projet a permis l'utilisation de l'outil tomcat, des fichiers servlet, jsp et des classes java. Il a également permis l'utilisation de l'architecture MVC ainsi que hibernate pour la partie base de données. L'idée était donc de réaliser un site de commerce en ligne qui met à profit ces outils. Notre groupe compte quatre membres: CAI Christophe, LEMBA Mohamed, HADDACHE Noaman et OUAALI Abd-Ennour. Nous avons décidé de réaliser un site de e-commerce spécialisé dans la vente de manga, ce thème nous regroupant car faisant tout parti de l'association "Animatsuri" de l'école. Mais ce choix s'est surtout fait dans un but de clarifier la compréhension de tous et de ne pas avoir à gérer trop de sous catégorie, ce qui n'est pas le but recherché pour ce projet.

Ce rapport a pour but de mieux comprendre le déroulement, fonctionnement et les moyens mis en place pour la réalisation du projet. Pour cela, nous verrons dans un premier temps le cahier des charges du site web pour mieux comprendre nos choix de structuration que nous verrons par l'intermédiaire d'un diagramme de classes ainsi que le MCD. Par la suite, nous ferons la lumière sur les problèmes rencontrés et les solutions proposées. Puis nous verrons les limites de ces outils ainsi que les axes d'améliorations. Finalement, nous terminerons l'utilisation du framework Spring Boot sur notre projet et l'optimisation qui permet d'atteindre.

Organisation :

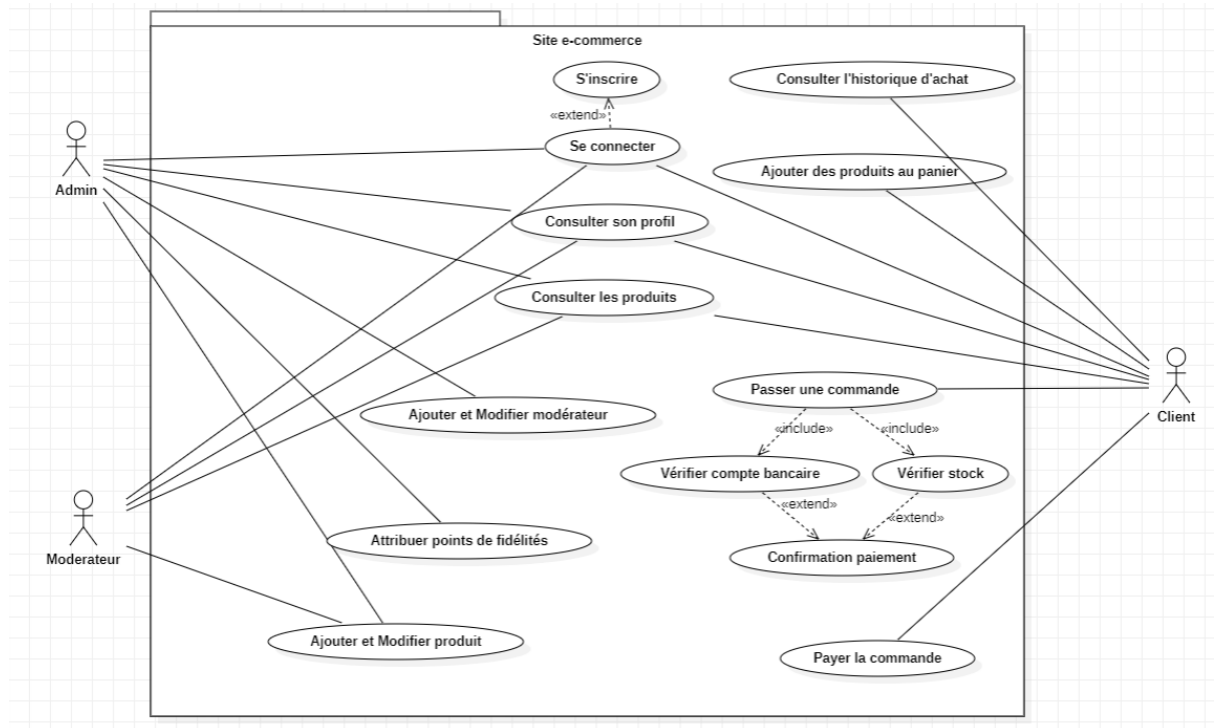
Un tel projet requiert une organisation rigoureuse, ce qui implique la mise en place de moyens appropriés. C'est pourquoi l'utilisation d'un référentiel Git accessible et partagé par l'équipe a permis un partage efficace de tous nos fichiers. L'autre moyen mis en place pour faciliter la communication de l'avancement du projet est l'utilisation de l'application discord qui nous a permis d'organiser des réunions mais également de partager les problèmes rencontrés.

La répartition des tâches s'est faite sur la base des diagrammes de classes mais pas seulement. En effet, un tel site web nécessite l'utilisation de nombreux nouveaux outils comme l'architecture MVC ou encore les bases de données. Les différentes parties du travail respectent une certaine chronologie et se divisait de la manière suivante:

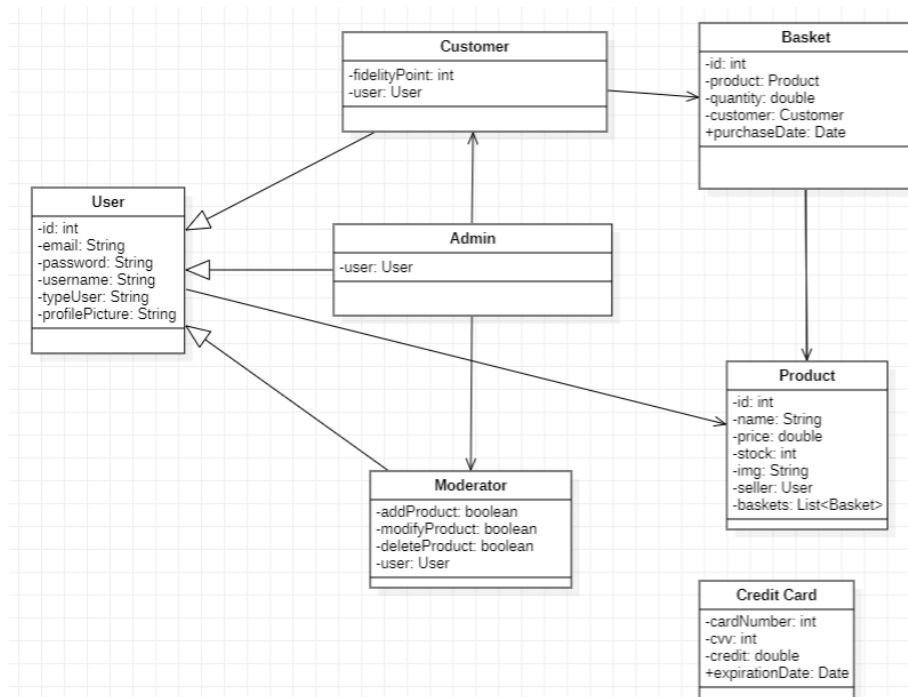
- ❖ Mise en place du projet, avec la compréhension et l'utilisation d'hibernate notamment qui a nécessité un certain temps. Nous avons finalement décidé de créer un projet "Maven" en utilisant l'application Eclipse. La configuration du fichier HibernateUtil, des dépendances pom.xml, etc.
- ❖ Ensuite vient la création de la base de données puis sa connexion avec le projet à travers Hibernate.
- ❖ La partie Model du projet, c'est la partie dans laquelle nous avons créé les entités de chaque table de la base. Une classe Main permet la création des tables et des attributs dans la base de données et également des lignes pour permettre de la tester.
- ❖ La partie Dao du projet, c'est la partie dans laquelle nous avons créé les classes avec le suffix Dao. Ces classes contiennent toutes les méthodes faisant appel à la base de données, elles contiennent donc les requêtes SQL. Cela permet entre autres de centraliser les requêtes et ne plus avoir à faire des requêtes SQL dans les autres parties.
- ❖ La partie Vue du projet, c'est la partie dans laquelle nous avons créé tous les fichiers .jsp, soit les fichiers permettant l'affichage des pages web de notre site. Le code principalement utilisé est donc du HTML mais également du CSS pour la gestion graphique.
- ❖ La partie Control du projet, c'est la partie dans laquelle nous avons implémenté les Servlets. Ces derniers permettent de gérer les réponses des formulaires, les redirections de pages et l'utilisation des méthodes précédemment instanciées dans la partie Dao.
- ❖ Enfin, la partie de la gestion des différents problèmes rencontrés, cela se fait au fur et à mesure de l'évolution du projet, elle est donc à prendre en dehors de cette chronologie car la résolution de problème engendre souvent des changements dans les parties précédentes.

Diagramme :

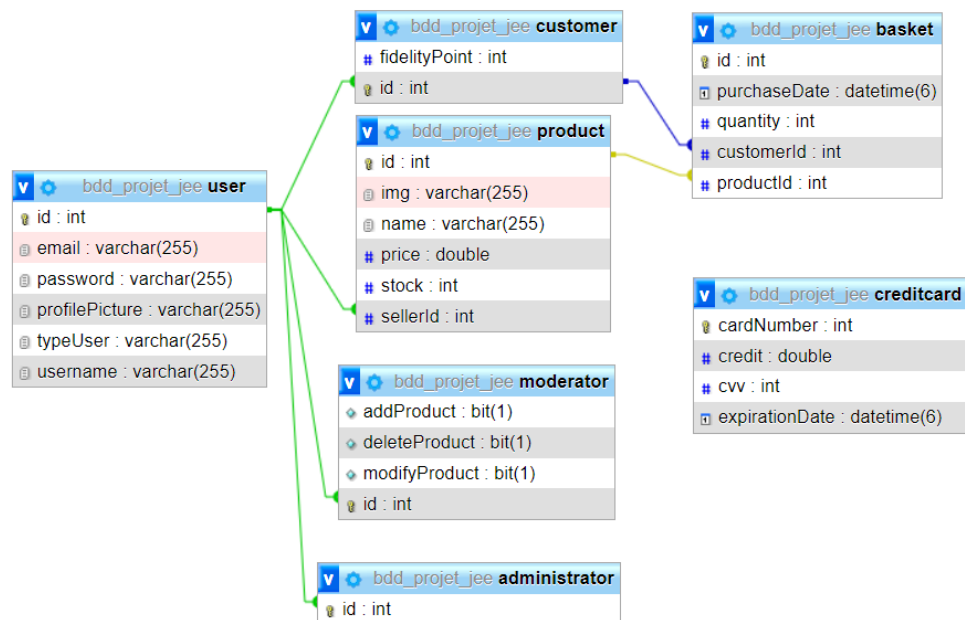
Cas d'utilisation



Classes



Structure de la base de données & MCD :



Problèmes et solutions :

Lors du projet, nous avons dû faire face à de nombreux problèmes empêchant l'avancement optimal du site. Ces problèmes permettent toutefois de mieux comprendre la manière dont nous avons travaillé et le déroulement du projet.

1. Problème de Configuration du Projet avec la Base de Données (BDD):

- **Problème:** Lors de la configuration initiale, la création d'un projet lié à la base de données était impossible, et l'ajout d'éléments était entravé.

- **Solution:** Nous avons initialement tenté de migrer vers IntelliJ, mais étant plus familiers avec Eclipse, nous avons opté pour un projet Maven. La configuration manuelle des fichiers tels que pom.xml et hibernateUtil.java a résolu ce problème.

2. Gestion des Données Variables dans les Bases de Données:

- **Problème:** Les variations dans les données de la base de données entre les utilisateurs pouvaient entraîner des problèmes. En effet, il fallait avoir les mêmes données dans la base pour pouvoir avoir le même référentiel en cas de problème. Un membre du groupe peut par exemple ne pas être au courant d'une modification de la base qui entraîne un dysfonctionnement dû à un code basé sur la nouvelle architecture.

- **Solution:** Nous avons déplacé la création de tables et l'insertion de données correspondantes dans la classe principale (main) afin de garantir une initialisation cohérente des bases de données pour tous les utilisateurs. Ainsi dès qu'une modification était faite dans la base de données, nous avons juste à relancer le main en supprimant les anciennes tables pour être à jour.

3. Problèmes dans la Couche DAO (Data Access Object):

- **Problème:** Des difficultés ont émergé dans la partie DAO, notamment lors de l'implémentation des méthodes liées aux requêtes SQL.

- **Solutions:**

- Eclipse demandait de sélectionner une classe ("SELECT *, 0 AS clazz_ FROM User") qui n'existait pas. Nous avons résolu ce problème en révisant la syntaxe des requêtes.

- Des erreurs ont surgi en raison de tables qui avaient été modifiées dans notre architecture. Nous avons adapté les requêtes pour refléter ces changements.

4. Gestion du Hachage des Mots de Passe:

- **Problème:** Pour des raisons de sécurité, nous avons dû hacher les mots de passe dans la base de données, mais la fonction de comparaison renvoyait toujours "false" car le mot de passe était haché 2 fois.

- **Solution:** Nous avons hacher le mot de passe avant de créer l'utilisateur, ce qui a résolu le problème de comparaison des mots de passe de manière sécurisée, assurant ainsi une gestion appropriée de la sécurité des mots de passe stockés.

5. Gestion de la sauvegarde des images:

- **Problème:** Pour mieux personnaliser les produits, on a voulu implémenter l'ajout d'images dans la base de données. Cependant, on a eu beaucoup de mal à récupérer le fichier donné et à le sauvegarder dans les dossiers.

- **Solution:** On a réussi à trouver un type (Part) pour pouvoir récupérer l'image du formulaire et File pour sauvegarder le fichier dans le dossier.

Ces solutions ont permis de surmonter efficacement les obstacles rencontrés tout au long du développement du projet, assurant une base solide pour le site de commerce en ligne en Java. En effet, il est important et nécessaire de tester toutes les fonctionnalités au fil de l'implémentation et corriger les erreurs éventuelles pour éviter de mauvaises surprises à la fin qui nécessiteraient une rétrospective laborieuse.

Limites & axes d'amélioration :

Notre projet comprend certes quelques limites. Tout d'abord, concernant la carte de crédit, nous ne pouvons pas faire une vérification bancaire car il faudrait pour ça utiliser de vraie carte de crédit. En effet, les banques effectuent une vérification et une confirmation de commande pour l'utilisateur. Ensuite, la confirmation est transmise au site web pour permettre de valider la commande. Nous avons quant à nous fait une vérification des coordonnées bancaires en vérifiant si la carte est dans notre base de données et si les données telles que si la date de péremption est dépassée ou non.

Une autre limite est la suppression des données utilisateur lorsqu'on transforme un client en modérateur et inversement. En effet, ce changement créer un nouveau type, on supprime donc le client de la table et ajoutons une ligne dans la table modérateur ce qui a pour conséquence une perte des données utilisateur de base.

Notre projet présente des axes d'améliorations, nous en avons conscience mais de part notre contrainte de temps il n'était pas envisageable de les ajouter. D'autant plus que ces améliorations concernent essentiellement l'esthétique et l'ergonomie du site. Ainsi, l'essentiel est que toutes les fonctionnalités constituant les cahier des charges et bien plus encore sont disponibles.

Concernant l'ergonomie, nous pourrions par exemple améliorer notre site en ajoutant une pagination des produits car ils sont présents sur une seule page et non classés par catégorie par exemple. De plus, ces produits pourraient être triés et filtrés selon des critères pour encore plus améliorer l'expérience utilisateur.

Concernant les fonctionnalités, nous pourrions par exemple ajouter des pages d'historique pour consulter nos actions déjà menées et revenir sur une commande ou une consultation que nous avons déjà fait. De plus, nous pourrions ajouter une page spécifique pour le panier de commandes pour améliorer la lisibilité.

Conclusion

Ce projet nous a apporté beaucoup sur différents aspects. Au niveau technique comme la programmation, la mise en place des nouveaux outils, la compréhension du fonctionnement de ces derniers, la découverte de nouvelles manières de réfléchir en fonction des problèmes rencontrés. Mais également sur l'organisation, le travail en équipe et toutes les compétences humaines que cela demande.

Tout cela participe au fait que ce projet nous a appris beaucoup de choses qui n'auraient pas été possible d'apprendre en suivant juste des cours de manière conventionnelle. Nous pensons notamment à l'utilisation concrète de Tomcat, Hibernate, les bases de données ainsi que l'architecture web MVC avec l'utilisation des différents types de fichier (servlets, classes java, jsp et tous les langages de programmation web nécessaire). La création de ce site web nous a permis de toucher à tous les aspects de la programmation et non seulement la compréhension de langage informatique. Nous avons dû mettre en relation les différentes catégories de fichiers, trouver des moyens pour résoudre les problèmes rencontrés avec notamment une grande partie du temps dédié à la recherche de solution et la compréhension, l'utilisation d'outil non introduit en cours mais indispensable dans le monde professionnel comme Github. Tous ces éléments nous permettent de mieux nous préparer à ce qui nous attend dans le monde du travail.

Nous tenons à remercier succinctement Monsieur Mohamed Haddache pour l'initiation au java EE et tous les outils qui en découlent ainsi que ce projet qui nous permet de mieux les appréhender. Nous tenons à remercier Monsieur Haddache également pour l'aide et la patience qu'il a eu envers nous.