

Rapport de projet

The logo for Java EE. It features a blue coffee cup with three orange flames rising from it. To the right of the cup, the word "Java" is written in blue, and "EE" is written in orange. A small "TM" trademark symbol is to the upper right of the "EE".

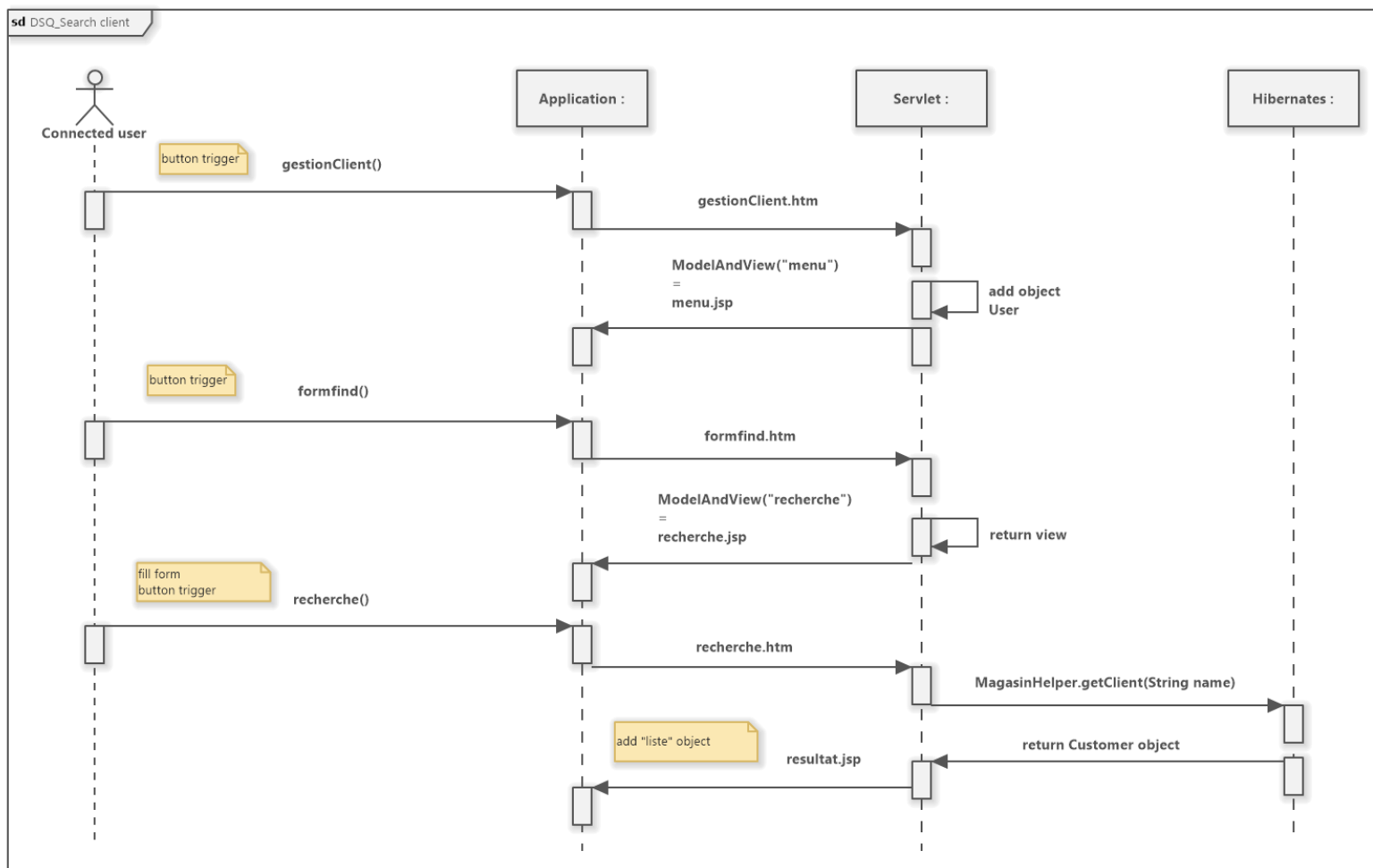
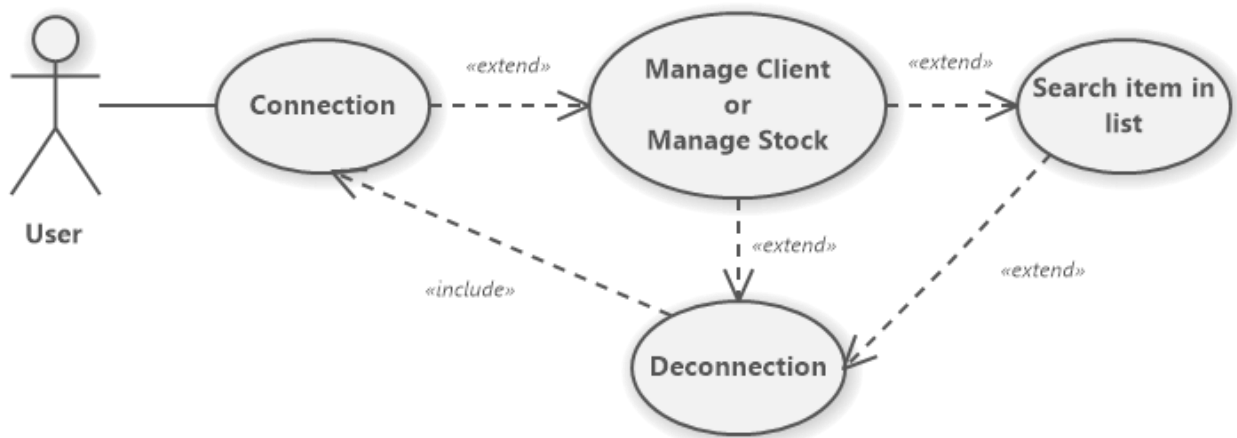
Analyse du besoin :

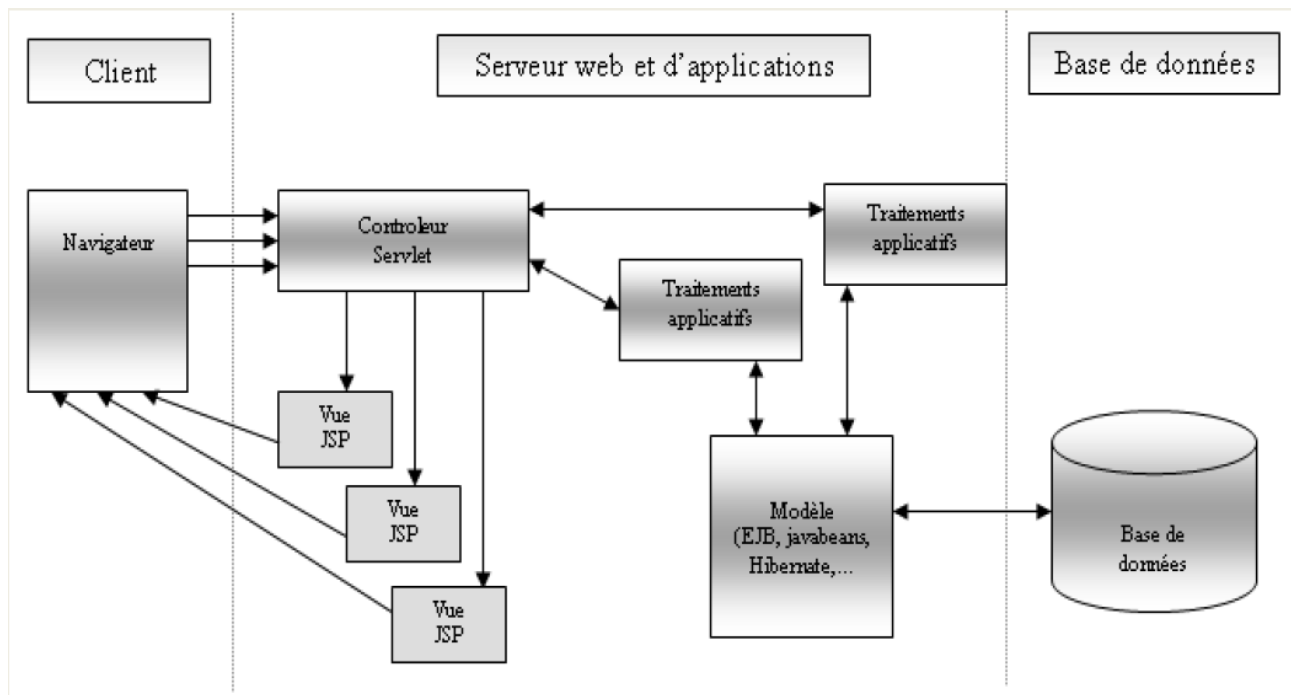
Notre client attend une application web basée sur des technologies JEE (Nous avons choisis Hibernates v4.3.1 & Spring v3.2.7).

Cette application doit permettre à ses utilisateurs des connexions différenciées et protégées grâce à une authentification sécurisée de type formulaire, elle donnera la possibilité aux utilisateurs de gérer les ventes, le stock, les références produits et leurs clients. Sur toutes ces données elle permet les opérations CRUD (Create, Read, Update, Delete), et la recherche d'un client selon son nom et ses achats.

Les accès à chacune de ces fonctionnalités se fera grâce à un menu.

Chaque vues indiquera le login de l'utilisateur connecté et un bouton de déconnexion qui le renverra à la page d'authentification.



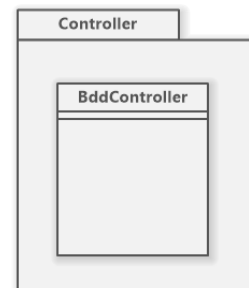


Contrôleur Servlet :

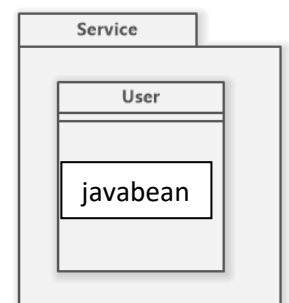
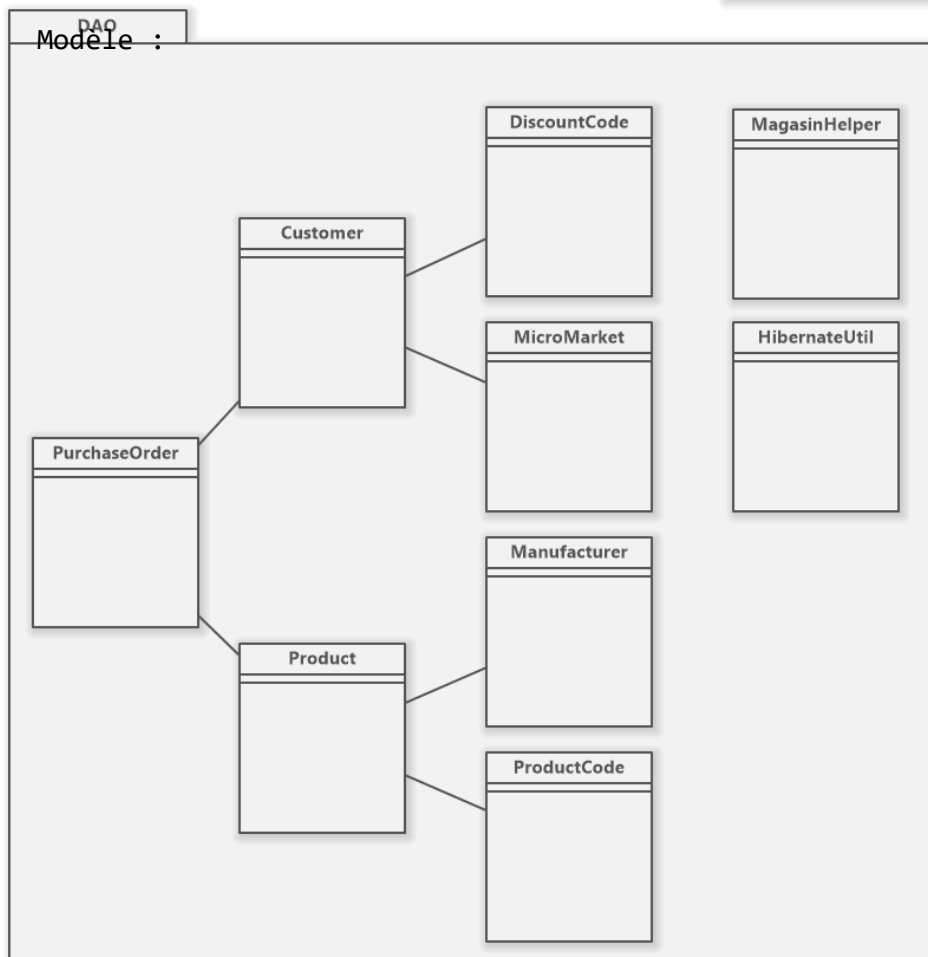
- BddController.java

Base de données :

- Derby sample app



Modèle :



Gestion de projet :

Suite à des problèmes de configurations des environnements de développement pour JEE chez Joris et en raison de notre nombre réduit dans cette équipe (duo). Nous avons fait le choix de travailler en peer-programming, afin de mettre en commun nos compétences et d'échanger plus efficacement sur les choix techniques et stratégiques de notre application.

Grâce à Discord, nous avons organisé nos sessions de travail (en dehors du temps prévu par les cours) sur deux après-midis. Le 19 mai et le 3 juin. La première session a consisté en une nouvelle analyse du besoin afin de fournir l'interface la plus adaptée au cahier des charges. Nous en avons profité pour parfaire notre maîtrise de la configuration du serveur Glassfish et de la base de données Derby. Au bout de 4 heures nous avons réussi à réaménager les requêtes Hibernates et à sécuriser notre application.

Le 3 juin, lors de notre dernière session nous avons terminé de créer la partie applicative sur la gestion des stocks, notamment la partie sur les produits. Comme nous avons déjà bien avancé dans les parties précédentes, il s'agissait d'une grande partie de répétition. Nous avons en outre pris le temps de faire un effort sur la présentation graphique avec une petite pointe de css.

Face aux difficultés rencontrées, nous avons su faire preuve de patience et le fait de travailler en duo a été très épanouissant et très motivant, l'un tirant l'autre vers le haut dans les phases difficiles. Cette organisation du mode projet à deux était une phase plutôt nouvelle pour tous les deux, nous étions plutôt habitués à des groupes de 3 ou 4 personnes. Cela a permis d'économiser beaucoup de temps en communication et en organisation.

Déjà fort d'une synergie sur un autre projet en binôme, nous avons mieux réparti notre travail, tant dans les recherches que dans la compréhension de l'applicatif. Nous avons pu mettre en commun des méthodes de debug.

En ce qui concerne nos choix dans l'application : Nous avons pris le parti de modifier en partie les méthodes de MagasinHelper en passant directement des objets pour la modification. C'est à dire qu'au lieu de passer en paramètre tous les champs dans le contrôleur, nous avons choisi de récupérer l'objet tel qu'il est en base, de modifier l'intégralité des champs (ceux non modifiés conservent donc leurs valeurs), puis d'insérer l'objet directement. Nous avons également appliqué cette décision à l'insertion d'un client et d'un produit.

Pour ce qui est de la modification d'un Produit, en raison d'une impossibilité de caster directement le résultat de la requête Hibernates, nous avons choisi de réaliser en deux étapes la récupération de données et la récupération des données d'un Object en Product. Cela afin de pouvoir profiter des mêmes avantages de tout ce qui avait été installé avant sur la gestion d'un Product plutôt que d'un Object qui est moins malléable.

En ce qui concerne le mapping Hibernates : Product.hbm.xml. Nous avons réalisé des modifications à la main du mapping puisque certaines propriétés étaient renvoyées comme Big Decimal par notre BDD. Les propriétés purchaseCost ainsi que Markup sont donc cast en float lors des requêtes d'un Produit et ont été remappés en conséquence.

Pour ce qui est de la modification de certains champs. Nous avons fait le choix

de limiter certaines opérations à un compte administrateur (groupe "admin" plutôt que "user") qui possède des droits différents. Par exemple, la modification du numéro d'un client n'est possible que par un administrateur. D'autres choix de ce genre sont réalisés comme pour un Produit il n'est pas possible d'indiquer la disponibilité, cela est calculé par le contrôleur et devrait également l'être par une procédure stockée. Par ailleurs, nous avons modifié le Service "User" afin d'implémenter la vérification du statut d'administrateur.

Afin de fournir aux utilisateurs un moyen simple de vérifier l'état de leurs sessions et de se déconnecter. Nous avons utilisé un fichier common.jsp pour afficher un menu sur l'intégralité des pages. La config a été réalisée avec une balise jsp config et include-prepare pour l'inclure dans toute notre application.

Le temps nous manquant, nous avons dû faire le choix également de gérer des erreurs de formulaire avec des instructions try / catch dans le MultiActionController BddController.java. En cas d'erreur nous renvoyons vers une jsp Erreur qui indique la nature de l'erreur. Cela permet de ne pas couper l'utilisateur de l'interface en cas de levée d'une exception mais en contrepartie, il doit reprendre son formulaire en cas de mauvaise utilisation de ce dernier.

Nous avons pris la liberté de refaire une touche de web-design avec un peu de css et quelques couleurs pour égayer et rendre plus agréable à l'oeil (inspiré du bash Ubuntu) ainsi qu'à l'utilisation notre interface.

La réalisation de projet nous a permis de mieux appréhender la configuration d'une application JEE avec Spring. De profiter de puissants outils comme l'ORM Hibernate et aussi de profiter de nos compétences Java que l'on ne cesse de mettre en pratique depuis le début de l'année. Ce projet LDLX conclut parfaitement le suivi des évolutions de la création d'un projet JEE de MVC 1 et jdbc jusqu'à la concrétisation d'une application moderne. Surmonter l'effort et la frustration de la mise en place d'un nouvel IDE parfois un peu capricieux (nous changeons trop souvent de machines cette année) nous a aussi permis de profiter de la libération après un déploiement réussi suite à un clean & build. Bien que sceptique et sans attentes particulières sur ce cours dans un premier temps, il est certain que nous avons évolué de manière significative comparé aux projets réalisés les années précédentes dans le Web (surtout Joris qui n'en avait jamais fait). Ce projet nous aura amené à tous d'eux beaucoup de confiance en notre capacité à appréhender de nouveaux environnements de développement tout en sortant de notre zone de confort.