

Rapport de Projet

Cyril LE ROY
Antoine-Ali ZARROUK
Benjamin ROBERT
Julien NORMAND
Jordan NOURRY

12 novembre 2013

Table des matières

I	Présentation	4
1	Introduction	4
2	Rôles des membres du groupe	4
3	Rappel de la demande	4
II	Gestion de projet	5
4	Organigramme des tâches	5
5	Planning prévisionnel	5
6	Planning réel	5
III	Développement	6
7	Justification des choix techniques	6
7.1	Netbeans	6
7.2	Git	6
7.3	Bootstrap	6
8	Procédures de tests	6
9	Conception UML	7
9.1	Diagramme de cas d'utilisation	7
9.2	Diagramme d'activité	8
9.3	Diagrammes de séquences	9
10	Points forts et points faibles de notre implémentation	12
10.1	Points forts	12
10.2	Points faibles	12
IV	Conclusion et perspective	13
11	Recette	13
12	Analyse des écarts	13
13	Évolutions proposées	13
13.1	Dynamiser la recherche	13
13.2	Tenir l'utilisateur informé	13
13.3	Clarifier l'administration	13
13.4	Dynamiser le suivi des commandes	13

14 Bilans personnels et de groupe	13
14.1 Cyril LE ROY	13
14.2 Antoine-Ali ZARROUK	14
14.3 Benjamin ROBERT	14
14.4 Julien NORMAND	14
14.5 Jordan NOURRY	14
14.6 Bilan de groupe	15

Première partie

Présentation

1 Introduction

Le projet BooXtore consiste à réaliser une maquette fonctionnelle d'un site de vente en ligne de livres. Le délai imposé pour l'analyse et le développement de l'application est de 5 jours, du 4 au 12 Novembre 2013.

2 Rôles des membres du groupe

La répartition de rôles est la suivante :

- **LE ROY Cyril** : Chef de projet, développeur
- **ZARROUK Antoine-Ali** : Développeur et Responsable de la logique métier
- **NORMAND Julien** : Développeur et Responsable de la couche de persistance
- **ROBERT Benjamin** : Développeur et Responsable de la couche de présentation Web
- **NOURRY Jordan** : Développeur et Responsable de la couche de présentation Web

3 Rappel de la demande

Dans le cadre d'un appel d'offre, est demandée la réalisation d'un prototype de site de vente en ligne. Ce site dédie ses ventes aux livres, pouvant donc s'assimiler à une librairie en ligne.

Deux types d'utilisateurs sont alors définis :

- Les clients, navigateurs et / ou acheteurs
- Les libraires, administrateurs et gestionnaires des ouvrages

Deuxième partie

Gestion de projet

4 Organigramme des tâches

5 Planning prévisionnel

6 Planning réel

Troisième partie

Développement

7 Justification des choix techniques

7.1 Netbeans

Netbeans est un IDE. Un IDE est un Environnement de Développement Intégré. Il fournit tous les outils nécessaires pour implémenter notre projet.

Nous avons choisi Netbeans pour sa bonne intégration avec Glassfish : le serveur est directement contrôlable et débuggable depuis Netbeans.

7.2 Git

Git est un logiciel de gestion de versions. Il nous permet de gérer différentes versions concurrentes de notre code.

Git nous a permis de travailler ensemble sur ce projet.

7.3 Bootstrap

Bootstrap est un framework CSS. Il s'agit d'un ensemble de règles définissant l'apparence de notre page web.

Utiliser Bootstrap nous permet de nous concentrer sur la structure de nos pages, plutôt que de créer nous-même nos styles alors que notre temps est déjà limité.

8 Procédures de tests

9 Conception UML

9.1 Diagramme de cas d'utilisation

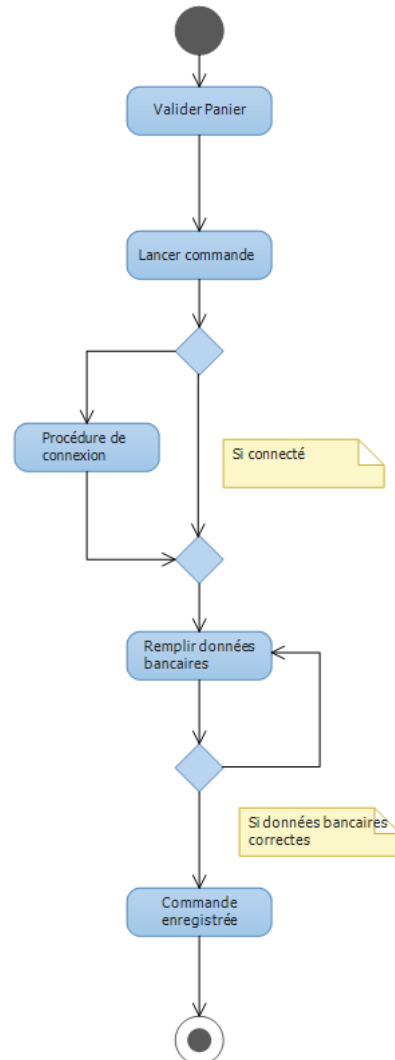


Le diagramme des cas d'utilisation a été créé pour nous fournir une représentation visuelle des relations entre les différentes fonctionnalités de l'application ; ainsi que de nous fournir un récapitulatif de ces dernières.

Nous voyons ici que l'utilisateur lambda a accès à la consultation du catalogue, aux fonctions de recherche, à l'inscription, à la gestion de son panier (comprenant la modification de la quantité des articles contenus, l'ajout / la suppression d'articles)

L'utilisateur peut également décider de se connecter / s'inscrire pour devenir client, ce qui lui donnera accès aux fonctions de création de commande, de modification des informations personnelles ainsi qu'à celles d'affichage de l'historique des commandes

9.2 Diagramme d'activité



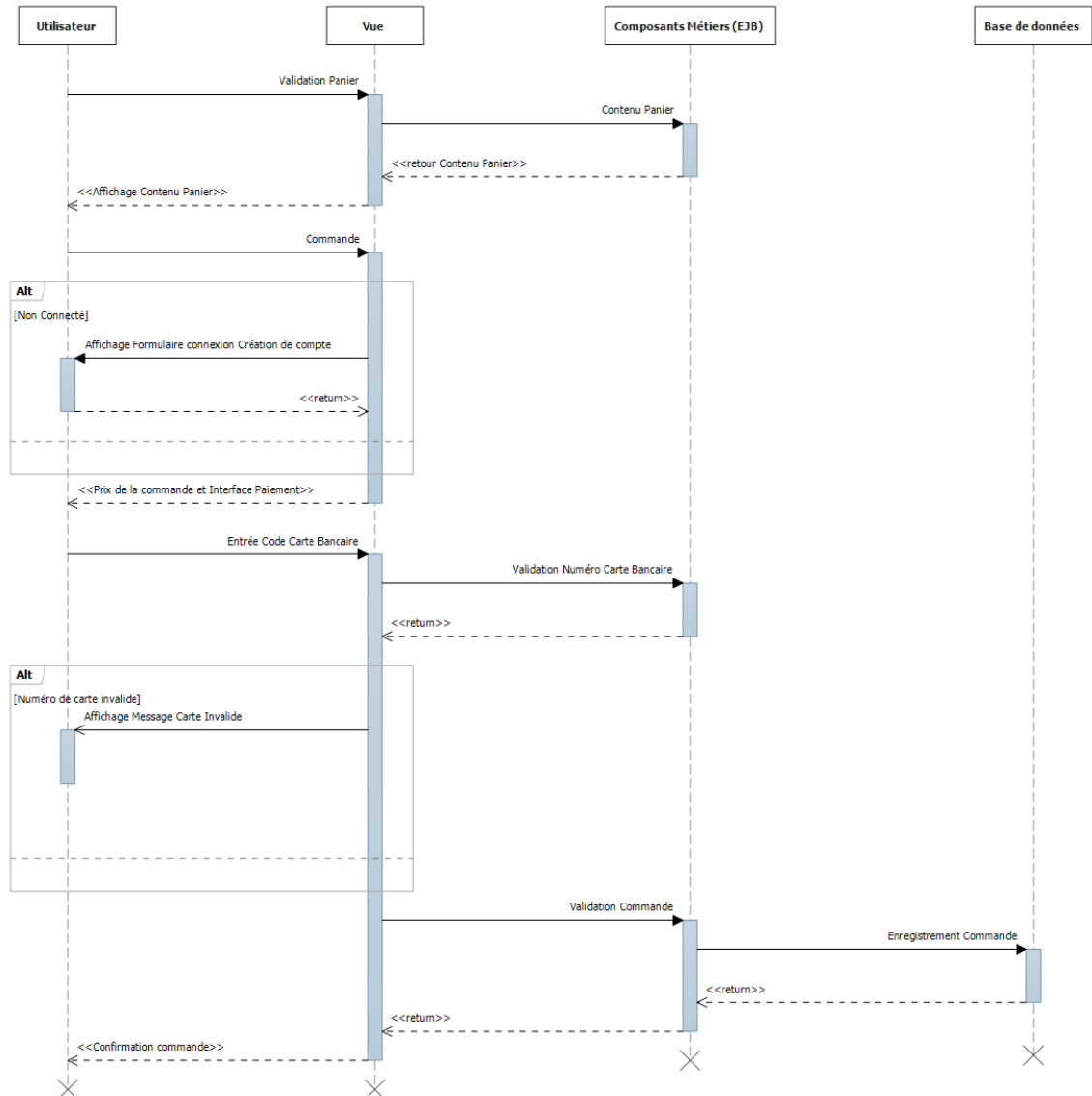
Le diagramme d'activité récapitule l'intégralité des actions à effectuer par l'utilisateur afin de créer une nouvelle commande.

Une fois le panier rempli, l'utilisateur va demander à le valider. Cette validation va lancer la création de la commande. Si l'utilisateur n'est pas connecté, l'application va demander à l'utilisateur de se connecter.

Si l'utilisateur était connecté, alors on passe à la phase de remplissage des données bancaires. Une fois saisies, les données sont vérifiées. Le client devra ressaisir des coordonnées bancaire jusqu'à ce qu'elles soient valides.

Une fois cette vérification effectuée, la commande est enregistrée.

9.3 Diagrammes de séquences



Ce diagramme de séquence décrit les interactions entre les différentes couches de l'application nécessaires pour la création d'une commande.

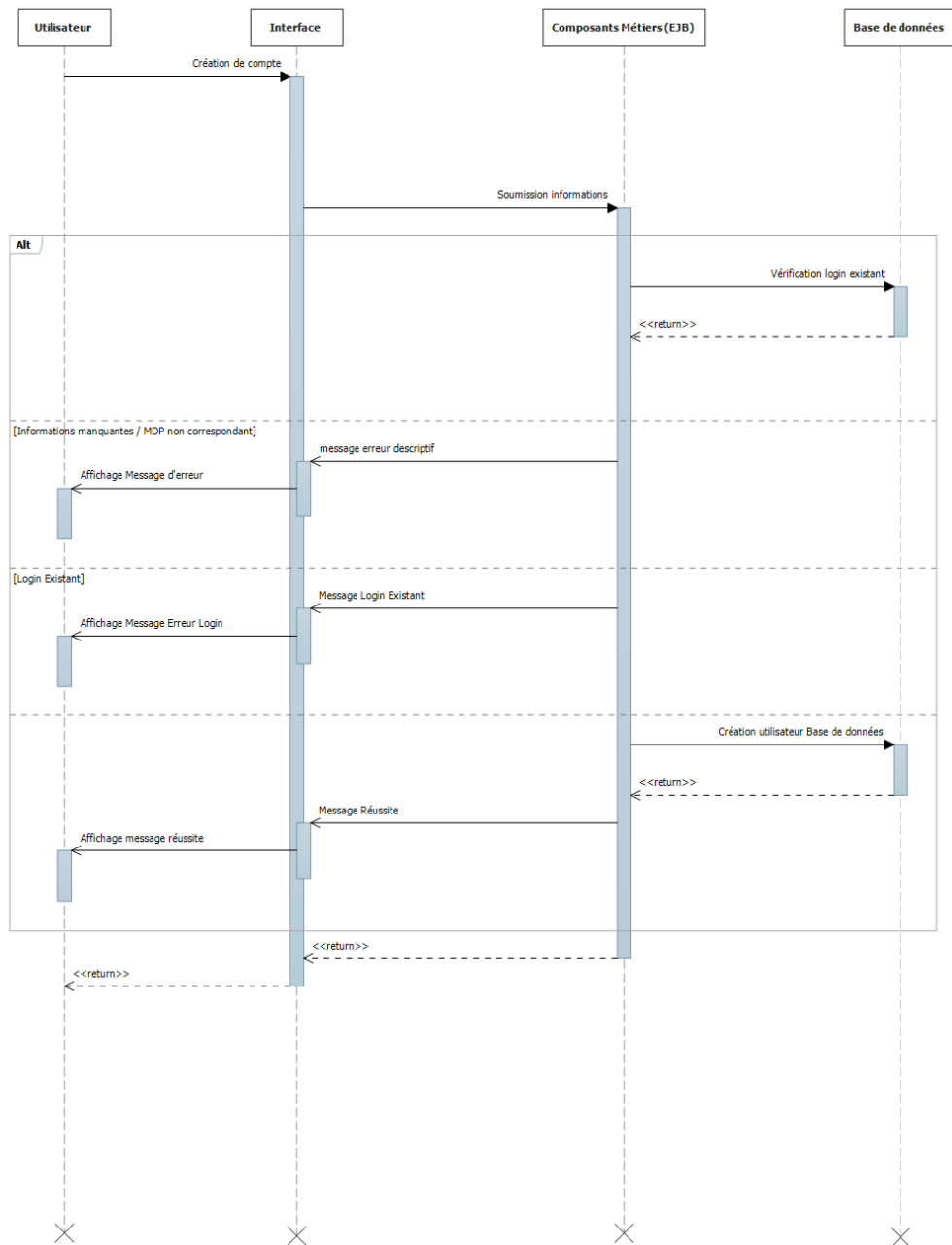
Une fois la validation du panier effectuée par le client, le signal de validation est envoyé à la vue. Vue qui demande le contenu du panier au composant métier dédié.

Le client valide alors sa commande. S'il n'est pas connecté, la Vue lui affiche alors le formulaire de connexion / inscription. Une fois la connexion faite la Vue demande à l'utilisateur de rentrer ses informations bancaires. Si le numéro entré est invalide, on affiche un message d'erreur à l'utilisateur et on redemande la saisie.

Si le numéro est validé, on envoie un message de validation de la commande au composant métier qui va enregistrer la commande en base de données.

Une fois la commande enregistrée, la base renverra une confirmation au compo-

sant métier qui renverra lui même une confirmation à la vue qui informera l'utilisateur de la réussite de l'opération.



Ce diagramme de séquence décrit les interactions entre les différentes couches de l'application nécessaires pour la création d'un compte utilisateur.

Une fois que l'utilisateur a inscrit ses données dans le formulaire d'inscription, l'interface soumet les informations au composant métier qui va vérifier en base si le login existe déjà.

S'il existe, le composant métier renverra un message à la vue qui affichera un message d'erreur à l'utilisateur qui devra choisir un login différent.

Si une autre erreur est détectée, l'interface l'affichera de la même manière.

Si les informations sont correctes, le composant métier créera l'utilisateur en base et en informera la vue qui à son tour en informera l'utilisateur.

10 Points forts et points faibles de notre implémentation

10.1 Points forts

- **L'ergonomie.** Le site a été conçu avec Bootstrap. Il s'agit d'un framework CSS : il va gérer l'apparence de nos pages à notre place. Ceci nous a permis de nous concentrer sur la structure et l'ergonomie. Par exemple :
 - La barre de navigation. Celle-ci permet - comme son nom l'indique - de naviguer dans notre site. Elle fournit des informations utiles à l'utilisateur, comme le contenu de son panier. On peut également effectuer des recherches directement à partir de celle-ci.
 - Le site est *responsive*. Cela signifie que l'affichage s'adapte à la taille de l'écran de l'utilisateur. Ainsi, que celui-ci utilise un smartphone, une tablette, un laptop, ou même un écran plus large, le contenu sera toujours disposé de façon optimale pour lui.
- **La partie client s'exécute rapidement.** Couplé à une bonne ergonomie, cela offre à nos utilisateurs une expérience très fluide.

10.2 Points faibles

- Le catalogue n'affiche pas de résultats par défaut. Autrement dit, il faut effectuer une recherche pour afficher des livres.
- La page de gestion n'est pas paginée, comme peut l'être le catalogue par exemple. Dans le cas d'un nombre important de livres, l'affichage peut être ralenti.

Quatrième partie

Conclusion et perspective

11 Recette

12 Analyse des écarts

13 Évolutions proposées

13.1 Dynamiser la recherche

La recherche est actuellement tout ce qu'il y a de plus statique. Simple, classique, efficace. Mais il est possible d'améliorer et de fluidifier l'expérience utilisateur.

- Utiliser des suggestions de recherches. Comme son nom l'indique, cela consiste à suggérer des résultats à l'utilisateur, lorsqu'il tape sa recherche. Un tel système, bien implémenté, permet à l'utilisateur de trouver avant même d'avoir fini de taper.
- Recharger dynamiquement la page. Actuellement, la page est rechargée dans son intégralité lorsqu'une recherche est effectuée. Il est possible de ne recharger qu'une partie de la page (ici, la partie affichant les résultats de recherche), pour au final une expérience plus fluide.

13.2 Tenir l'utilisateur informé

Il pourrait être possible de tenir l'utilisateur informé des événements importants de notre librairie. On peut imaginer un système d'abonnement via newsletter ou flux RSS. Un abonnement aux dernières sorties d'un genre ou d'un auteur particulier permettrait de tenir le lecteur informé des sujets qui l'intéressent.

13.3 Clarifier l'administration

L'interface d'administration du libraire doit présenter beaucoup d'informations. Implémenter un système similaire à un tableur, où le libraire est en mesure de choisir quelles informations afficher, et les trier plus librement serait une amélioration pour la lisibilité.

13.4 Dynamiser le suivi des commandes

Il serait possible d'intégrer un indicateur à la barre de navigation, qui afficherait en temps réel les informations concernant la dernière commande de l'utilisateur.

14 Bilans personnels et de groupe

14.1 Cyril LE ROY

Ce projet m'a appris deux choses. La première, je n'aime pas Java EE. Quand on est habitué à développer des sites Web du même genre que BooXtore de manière simple et rapide, il est difficile de passer à la plateforme Java, à GlassFish

et ses maudites erreurs de déploiement qui font perdre énormément de temps, aux chargements interminables et j'en passe.

Cependant, le fait de travailler en architecture MVC et d'être supporté par le langage Java (qui reste un langage très puissant et utile pour un développeur grâce à ses librairies riches) est très intéressant car cela se rapproche de nombreux Frameworks PHP et facilite donc de nombreuses étapes de développement.

Évidemment, il est frustrant de passer beaucoup plus de temps à l'apprentissage et donc au développement sur des fonctionnalités qui semblent simples.

Pour conclure, je dirais que je viens tout juste de me rendre compte que depuis deux ans je me plains du manque de temps sur les projets et de ma frustration éternelle sur le fait de ne jamais pouvoir présenter un résultat digne de ce nom.

14.2 Antoine-Ali ZARROUK

Je trouve que Java EE est une technologie intéressante, cependant, je trouve la technologie JSF/JSP assez limitante par rapport à un framework graphique type Vaadin.

Bootstrap a été une bonne surprise de par sa flexibilité et sa facilité d'utilisation.

Cependant, diviser le temps par deux pour un tel projet suppose que tout le groupe soit au top concernant la technologie

14.3 Benjamin ROBERT

Acclimatation obligatoire pour un environnement tel que le Java EE, où l'architecture d'une application légère peut prendre des allures d'application lourde seulement avec un langage.

Néanmoins, pas d'insatisfaction à la pensée de réaliser un site d'e-commerce pour la mise en application de ce langage. De bonnes surprises quant aux différents outils utilisés, tels que BootStrap ou Git. Une expérience trop courte par rapport aux cahier des charges qui peut rapidement être conséquent sur les petites fonctionnalités.

14.4 Julien NORMAND

Pour ce premier projet à l'EXIA.CESI, ça été une grande découverte. Dans un premier temps, il m'a fallu prendre connaissance du fonctionnement d'un projet, les documents que nous sommes amenés à rendre ou à faire. Et la principale chose, comment était partagé le projet.

Concernant le projet en lui même, il était intéressant car on voyait plus en détails comment fonctionnait le JSF dans du Java EE. Vu mon niveau moyen en JAVA, je ne pensais pas parvenir à comprendre ce que je devais faire concernant ma partie.

14.5 Jordan NOURRY

Dans les grandes lignes :

- Je trouve Java EE encore plus désagréable à utiliser que Java SE.
- Bootstrap est vraiment un beau framework. Le prototypage y est très rapide, et il est simple de récupérer les sources pour les recompiler et avoir des CSS qui possèdent une vraie personnalité - tout en se basant sur une architecture très solide.

- C'est la première fois que j'utilise un logiciel de gestion de versions en conditions réelles. Ça aide énormément, d'autant plus lors du travail en groupe.

14.6 Bilan de groupe

Ce projet a été un vrai challenge. Outre le fait de la complexité du projet pour à peine une semaine de développement, la conduite du projet aura été semée d'embûches.

Tout d'abord, nous avons perdu un peu de temps en voulant utiliser un Framework de conception d'interfaces Web pour application Java appelé Vaadin. Au départ, les développeurs semblaient enthousiastes à l'idée d'utiliser ce Framework proposé par l'un des développeurs. Mais au vu de ces différentes utilisations de ce Framework, il semblait évident qu'il n'était pas du tout adapté à la conception de site Web tel que BooXtore. J'ai donc dû prendre la décision de rester sur JSF par sécurité, même si cela nous obligeait à reprendre une partie du travail.

La deuxième embûche est arrivée au milieu du développement. Nos entités avaient été générées grâce à la base de données conçue précédemment mais des erreurs bloquantes insolubles nous empêchaient toute avancée. La décision a donc été prise de laisser tomber la base de données et de recréer nous-mêmes nos entités depuis le début, et ce, deux jours avant la fin de la phase de développement. Cette décision était certes un pari, et un pari gagnant car nous avons vite pu faire de grands progrès. Cependant, le goût amer du temps perdu sur un projet court a été assez démoralisant.

Pour conclure, je ferais part de ma déception envers mon propre travail de chef de projet. J'étais vraiment motivé à conduire ce groupe de projet avec une bonne méthode de développement mais ce n'est que rendu à la fin que je me rends compte d'avoir peut être mal utilisé les ressources du projet et le temps alloué.