

Rapport de Projet

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

12 novembre 2013

Table des matières

I	Présentation	2
1	Introduction	2
2	Rôles des membres du groupe	2
3	Rappel de la demande	2
II	Gestion de projet	3
4	Organigramme des tâches	3
5	Planning prévisionnel	3
6	Planning réel	3
III	Développement	4
7	Justification des choix techniques	4
7.1	Bootstrap	4
7.2	Git	4
8	Procédures de tests	4
9	Conception UML	5
9.1	Diagramme d'activité	5
9.2	Diagramme d'activité	6
9.3	Diagrammes de séquences	7
10	Points forts et points faibles de notre implémentation	9
10.1	Points forts	9
10.2	Point faible	9
IV	Conclusion et perspective	10
11	Recette	10
12	Analyse des écarts	10
13	Évolutions proposées	10
14	Bilans personnels et de groupe	10

Première partie

Présentation

1 Introduction

Le projet BooXstore 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 6 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
- Robert Benjamin : Développeur
- Normand Julien : Développeur
- Nourry Jordan : Développeur

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 Bootstrap

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

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.

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.

8 Procédures de tests

9 Conception UML

9.1 Diagramme d'activité

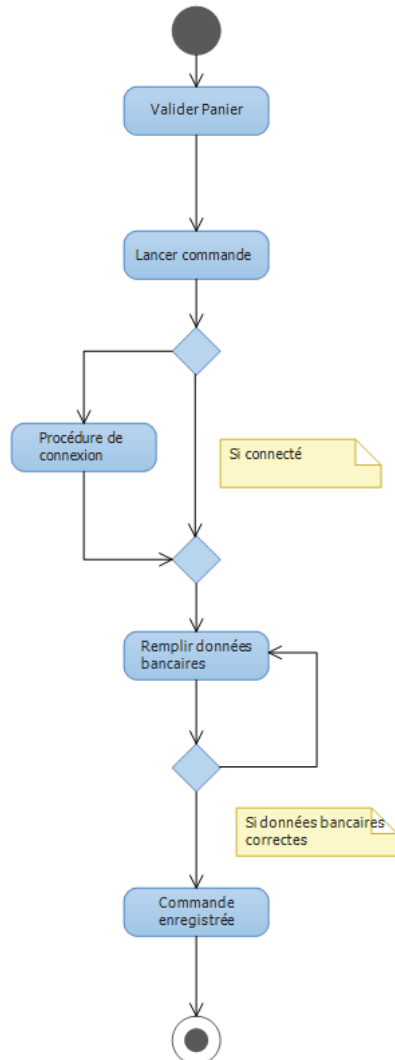


Le diagramme des cas d'utilisation a été créé pour nous fournir une représentation visuelle des relations entre les différentes fonctionnalités de l'application. Ainsi que de nous fournir un récapitulatif de ces dernières.

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

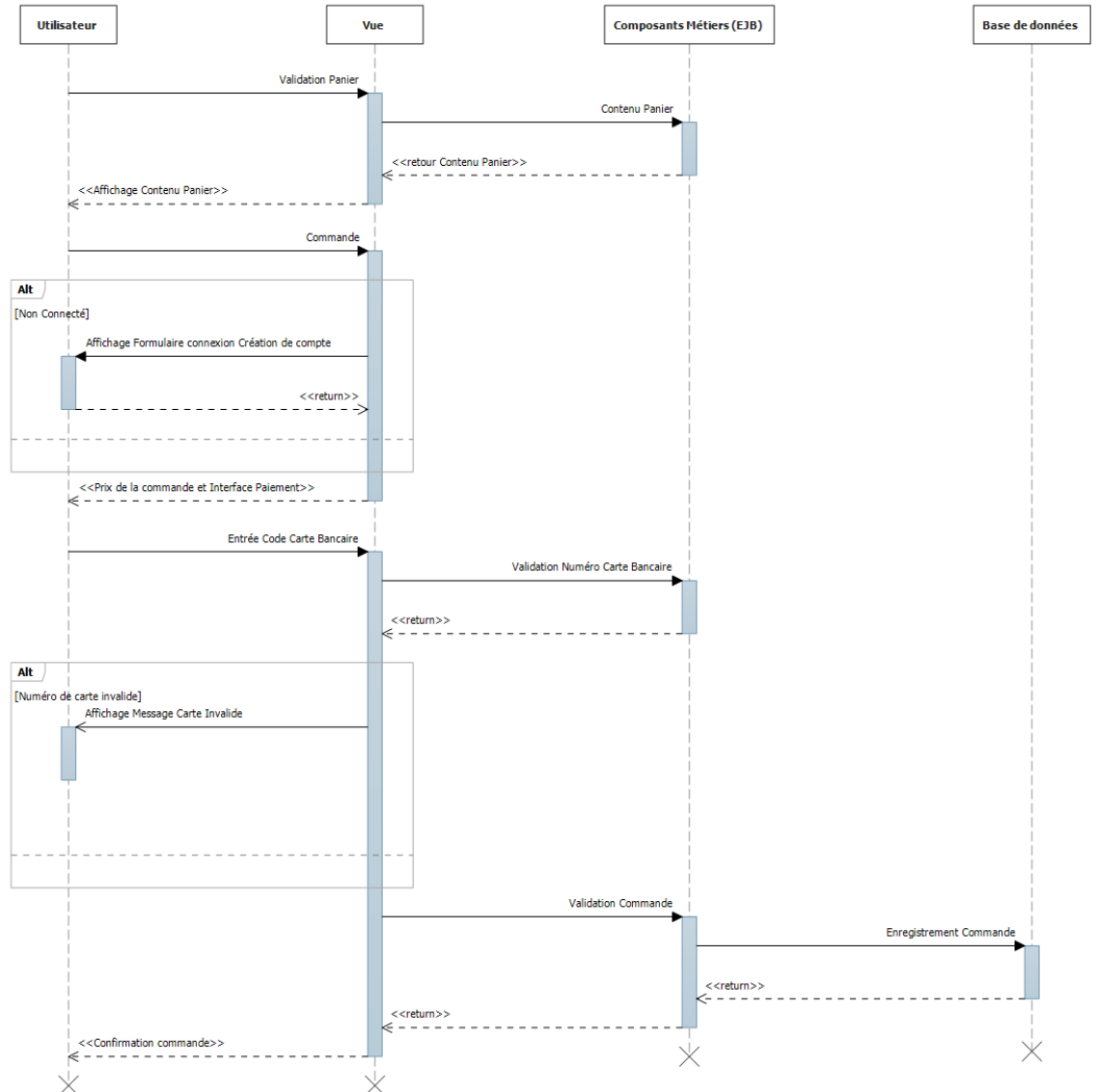
9.2 Diagramme d'activité



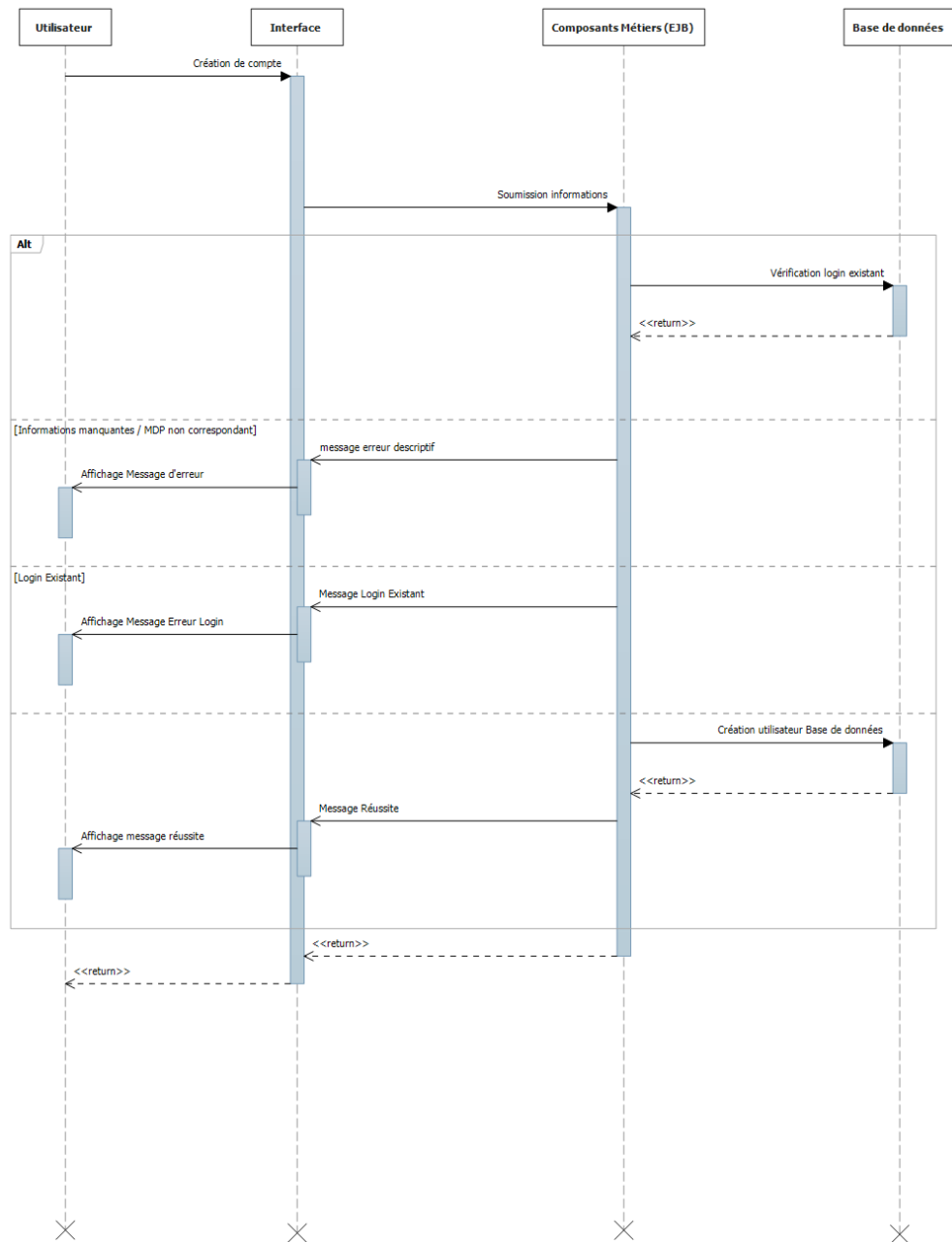
Le diagramme d'activité récapitule l'intégralité des actions à effectuer par l'utilisateur afin de créer une nouvelle commande.

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 validée, 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. La vue demande le contenu du panier au composant métier dédié. Le client valide alors sa commande. Si 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 valide, 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 composant 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. Si 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

- **La navbar** : Elle permet une utilisation rapide de la recherche de livre au lieu de voyager au travers des catégories afin d'avoir le livre qui nous convient le plus. La connexion/inscription de l'utilisateur est aussi présente sur celle-ci. Et pour conclure sur cette navbar, le panier affiche le pris et le nombre d'article de la commande en cours de l'utilisateur. Il lui reste ensuite qu'à cliquer dessus pour valider sa commande et passer au paiement.
- **Bootstrap** : Les pages créées avec ce style, permet d'avoir des pages intuitives pour les utilisateurs du store. L'esthétique/L'intuitivité du site est également important pour que l'utilisateur trouve ces repères rapidement et puisse ainsi revenir s'il trouve que le site lui plait.
- Nos pages du store s'adapte également au fenêtre qu'on lui impose si elle est petite la navbar s'adapte à celle-ci et devient un bouton ou l'on clique dessus pour afficher le reste du menu.

10.2 Point faible

- Le temps de chargement de la page de gestions des libraires est trop longue à s'ouvrir/charger vu le nombre de données qui doit être afficher sur celle-ci. Cette page (management.xhtml) peut-être optimisée pour devenir plusieurs pages au lieu de faire des onglets. En gros, un onglet devient une page.

Quatrième partie

Conclusion et perspective

11 Recette

12 Analyse des écarts

13 Évolutions proposées

14 Bilans personnels et de groupe

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. Evidemment, 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.

Antoine-Ali ZARROUK

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 cahiers des charges qui peut rapidement être conséquent sur les petites fonctionnalités.

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 amené 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.

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.

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.