

Rapport de Projet

0.4

!!

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

12 novembre 2013

Table des matières

I Présentation	3
1 Introduction	3
2 Rôles des membres du groupe	3
3 Rappel de la demande	3
II Gestion de projet	4
4 Organigramme des tâches	4
5 Planning prévisionnel	4
6 Planning réel	4
III Développement	5
7 Justification des choix techniques	5
7.1 Bootstrap	5
7.2 Git	5
8 Procédures de tests	5
9 Conception UML	7
9.1 Diagramme d'activité	7
9.2 Diagramme d'activité	8
9.3 Diagrammes de séquences	9
9.4 Diagramme de composants	11
9.5 Diagramme de classe	11
10 Points forts et points faibles de notre implémentation	13
10.1 Points forts	13
10.2 Point faible	13
IV Conclusion et perspective	14
11 Recette	14
12 Analyse des écarts	16
13 Évolutions proposées	16
13.1 Dynamiser la recherche	16
13.2 Tenir l'utilisateur informé	16
13.3 Clarifier l'administration	16

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

Test unitaire	Résultat
navbar.xhtml	
Affiche un aperçu du panier	Vert
Lien vers connexion / bouton déconnexion	Vert
Recherche rapide	Vert
index.xhtml	
Affiche le top 10 des ventes	Rouge
Lien vers le catalogue	Vert
catalog.xhtml	
Effectue une recherche	Vert
Effectue une recherche par catégorie	Vert
Affiche listing des livres d'une catégorie	Vert
product.xhtml	
Affiche les informations détaillées d'un livre	Vert
authentication.xhtml	
Connexion utilisateur	Vert
Inscription utilisateur	Vert
Vérification login existant (inscription)	Vert
account.xhtml	
Modifie l'adresse mail (adresse valide)	Vert
Modifie le mot de passe (vérification mot de passe)	Vert
Modifie l'adresse	Vert
Affichage historique des commandes	Vert
cart.xhtml	
Supprime un produit	Vert
Incrémente un produit	Vert
Décrémente un produit	Vert
Vider le panier	Vert
Valider le panier	Vert
checkout.xhtml	
Affiche le résumé de la commande	Vert
Vérifie le numéro de carte	Vert
Enregistre la commande	Rouge

9 Conception UML

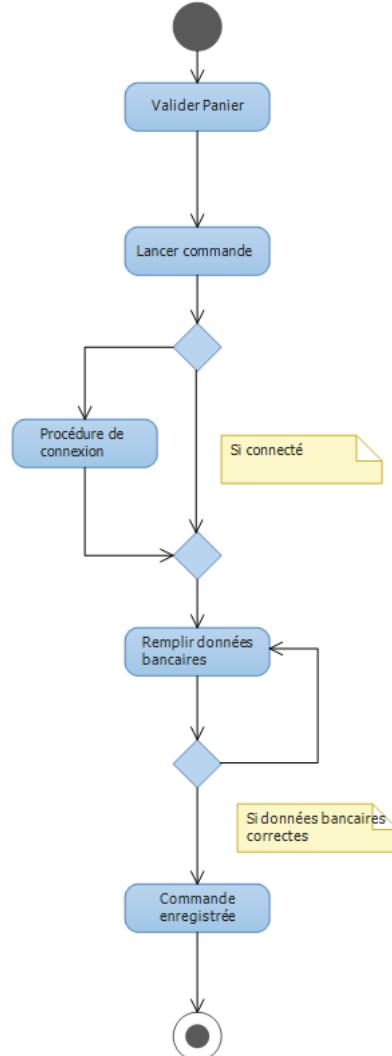
9.1 Diagramme d'activité



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 / l'as 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.

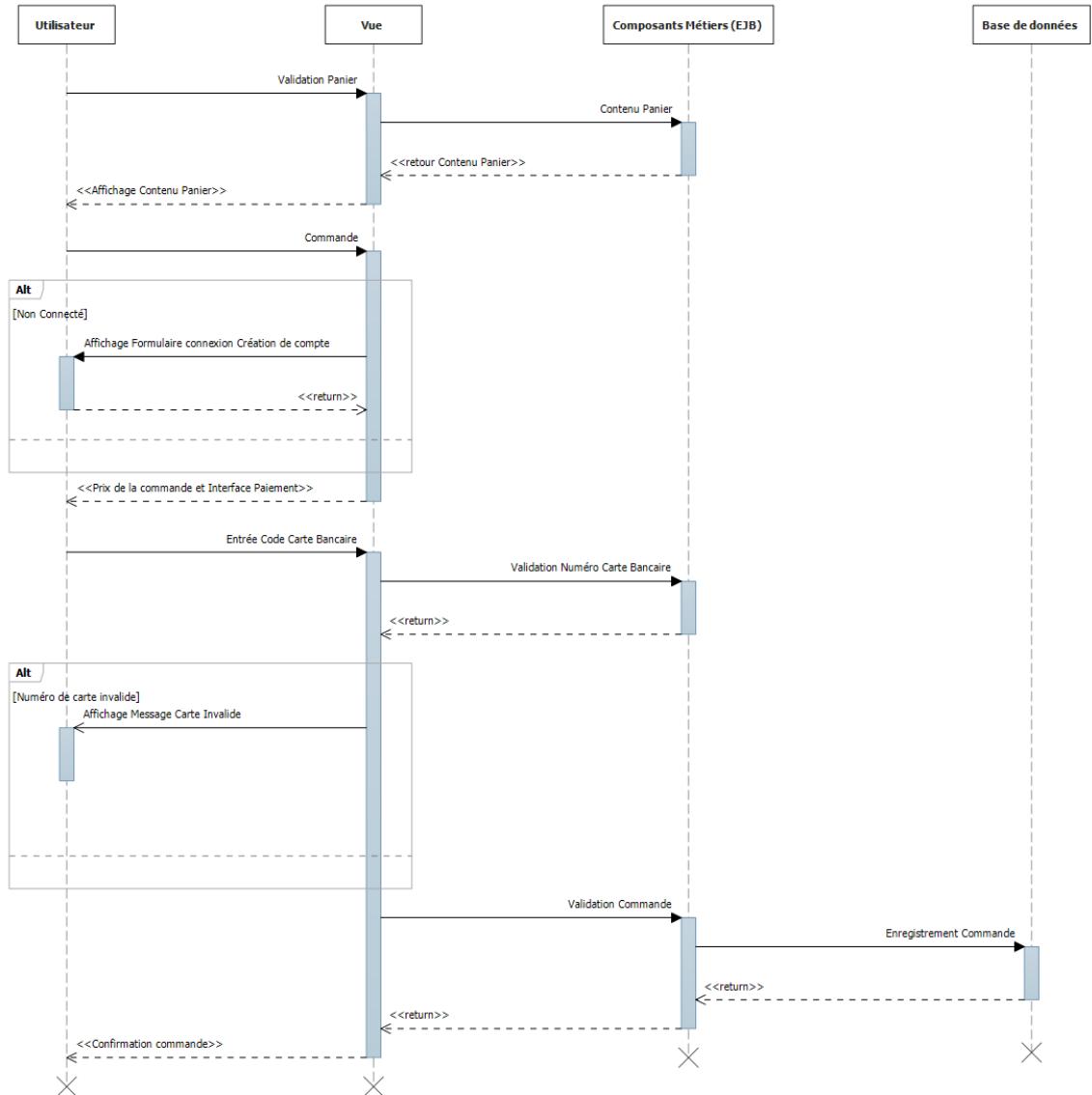
9.2 Diagramme d'activité



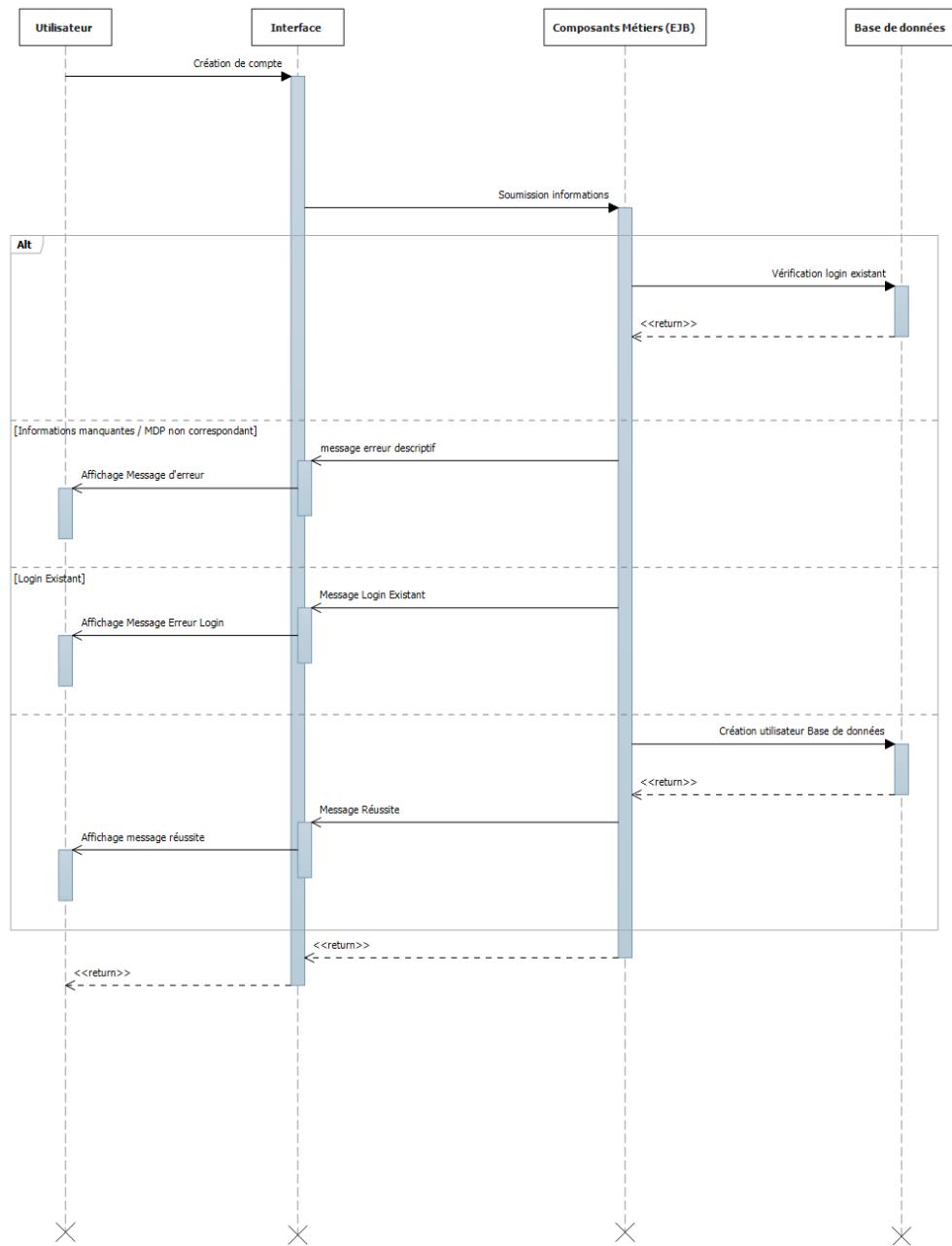
Le diagramme d'activité récapitule l'intégralités des actions à effectuer par l'utilisateur afin de créer une nouvelle commande.

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 resaisir des coordonnées bancaire jusqu'à ce qu'elles soient valides. Une fois cette vérification effectuée, la commande est enregistrée.

9.3 Diagrammes de séquences

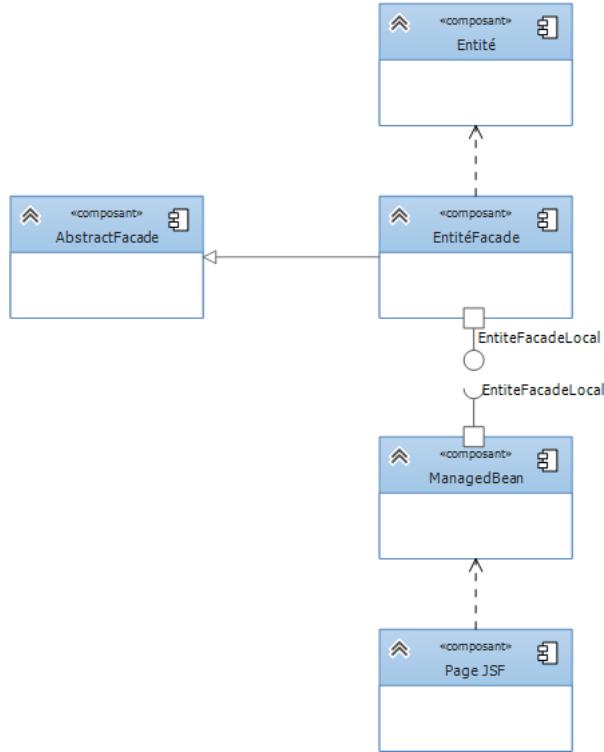


Ce diagramme de séquence décrit les interactions entre les différentes couches de l'application nécessaires pour la création d'une commande. Une fois la validation du panier effectuée par le client, le signal de validation est envoyé à la vue. Vue qui demande le contenu du panier au composant métier dédié. Le client valide alors sa commande. Si il n'est pas connecté la Vue lui affiche alors le formulaire de connexion/inscription. Une fois la connexion faite la vu 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 composant métier qui renverra lui-même une confirmation à la vue qui informera l'utilisateur de la réussite de l'opération.



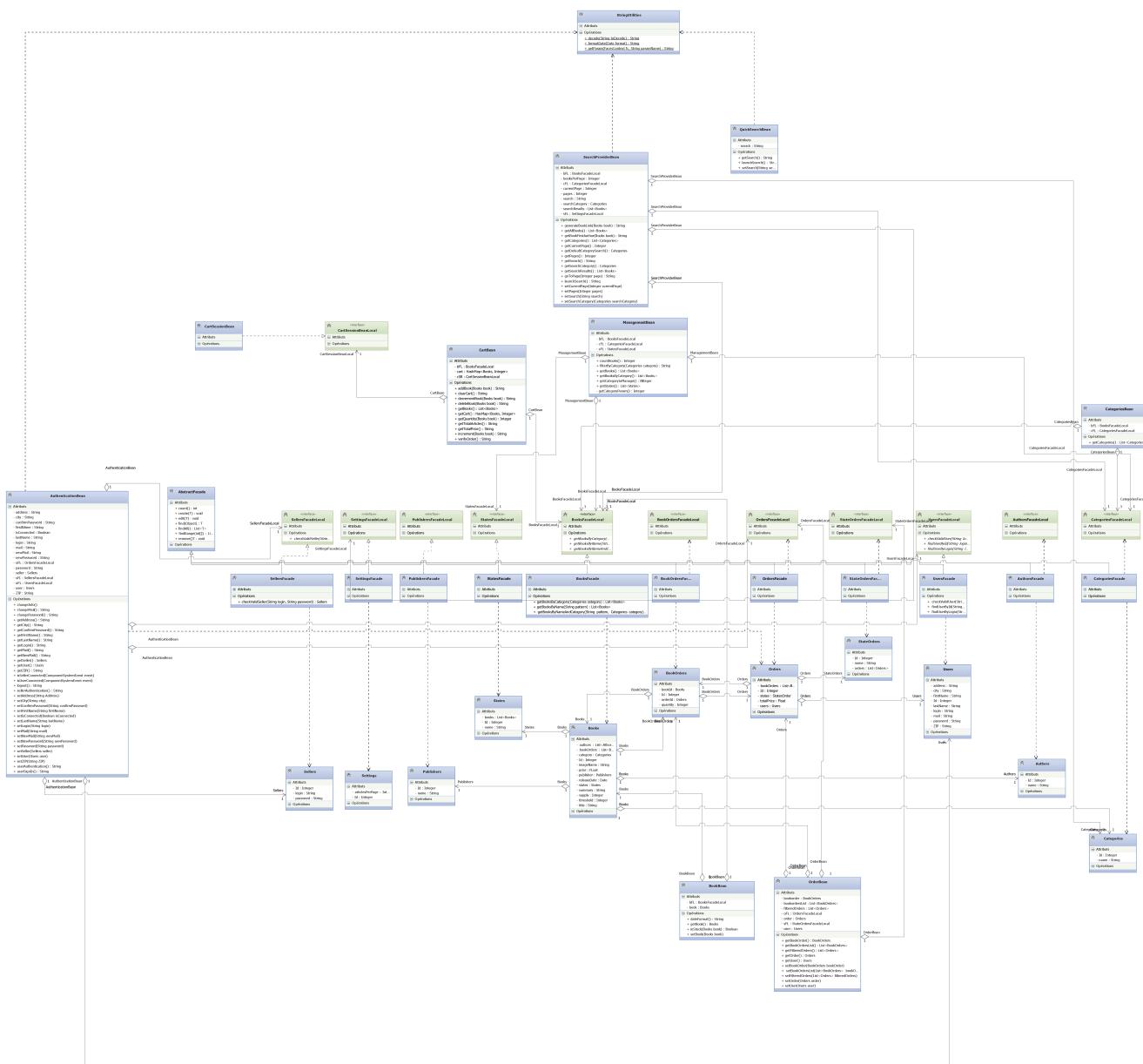
Ce diagramme de séquence décrit les interactions entre les différentes couches de l'application nécessaires pour la création d'un compte utilisateur. Une fois que l'utilisateur a inscrit ses données dans le formulaire d'inscription, l'interface soumet les informations au composant métier qui va vérifier 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.

9.4 Diagramme de composants



Le diagramme de composants a été écrit pour pouvoir généraliser le diagramme de classe qui, Java EE oblige, est très imposant. Ce diagramme décrit donc que les SessionBeans sont nommés selon le nom de l'entité qu'ils sont censés gérer. Qu'ils implémentent une interface locale et héritent de la classe abstraite AbstractFacade qui gère les opérations CRUD communes. Les ManagedBeans intancient les SessionBeans correspondants aux interfaces locales et permettent aux pages JSF d'accéder aux informations stockées en base de données.

9.5 Diagramme de classe



Au vu de la taille du diagramme, il se peut que celui-ci ne soit pas lisible sur une feuille A4, voici un lien pour le télécharger à part : <http://goo.gl/D9HGij>

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 payement.
- **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 plaît.
- 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

Nom page	Etape		Fonctionnalité	
Client				
Accueil (index.xhtml)	Design	JSF	Fonctionnel	Affiche l'accueil du site avec le top 10 des ventes
Catalogue livre par catégorie (catalog.xhtml)	Design	JSF	Fonctionnel	Ajout livre au panier / recherche par catégories / affichage par pages
Fiche détaillée livre (product.xhtml)	Design	JSF	Fonctionnel	Affiche la fiche détaillée d'un livre
Connexion/Inscription Client (authentication.xhtml)	Design	JSF	Fonctionnel	Connexion client / Création compte client
Gestion compte client (account.xhtml)	Design	JSF	Fonctionnel	Modification des informations du compte, visualisation historique commande
Gestion panier client (cart.xhtml)	Design	JSF	Fonctionnel	Permet de gérer le panier
Résumé commande + Validation (checkout.xhtml)	Design	JSF	Fonctionnel	Visualisation du résumé de la commande, saisie des infos bancaires + validation
Module navbar.xhtml	Design	JSF	Fonctionnel	Permet d'effectuer une recherche rapide, de se connecter, déconnecter, afficher panier
Librairie				
Connexion librairie (authentication_admin.xhtml)	Design	JSF	Fonctionnel	Connexion admin
Accueil administration (management.xhtml)	Design	JSF	Fonctionnel	Choix module à gérer
Gestion livre (management.xhtml)	Design	JSF	Fonctionnel	CRUD livre / Modification état livre (nouveauté-en stock-à venir-en réappro)
Gestion catalogue (management.xhtml)	Design	JSF	Fonctionnel	CRUD catalogue / Association livre - catégorie
Gestion suivi commandes (management.xhtml)	Design	JSF	Fonctionnel	Modification état commande (validée-en préparation-annulée-envoyée)
Gestion stock (management.xhtml)	Design	JSF	Fonctionnel	Définir stock pour X livre
Gestion stock critique (management.xhtml)	Design	JSF	Fonctionnel	Modification quantité en stock des article < seuil réappro

Non fonctionnel	Erreur bloquante
Semi-fonctionnel	Comportement différent
Fonctionnel	Fonctionnel

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 implanté, 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 tableauur, où le libraire est en mesure de choisir quelles informations afficher, et les trier plus librement serait une amélioration pour la lisibilité.

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 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. Boostrap à 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

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.

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 le 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'embuches. 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 embuche 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.