



Université Mohammed V de Rabat
École Nationale Supérieure d'Informatique et
d'Analyse des Systèmes



RAPPORT DU PROJET JAVA EE

RÉALISATION D'UNE APPLICATION WEB DE GESTION DES VENTES

Réalisé par :

MOUFID Ibtissam
MOUJIB Sara
NASRALLAH Meriem
WARDI Hafssa
ZEROUAL Najlae

Encadré par :

Pr. M. ElHamlaoui

Année Universitaire : 2019-2020

REMERCIEMENTS

Au terme de ce travail, nous tenons à exprimer nos vifs respects et notre profonde gratitude à tous ceux qui ont contribué à l'élaboration de ce projet dans les meilleures conditions.

De surcroît, nous remercions M. EL HAMPLAOUI MAHMOUD, encadrant du projet JEE, pour nous avoir assuré la possibilité d'évoluer en mettant en application les acquis de cette période dans les conditions les plus favorables.

Finalement, nous remercions tout le corps professoral et administratif de l'école nationale d'informatique et d'analyse des systèmes pour la formation qu'ils nous ont prodigué tout au long de ce parcours. Et nous témoignons par le présent travail notre vive reconnaissance à toute personne ayant aidé de près ou de loin à la réalisation de ce projet.

RESUME

Le présent document synthétise le travail réalisé dans le cadre du projet JEE à l'Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes. Ce projet a pour mission de concevoir et de développer une application web se basant sur les technologies JavaEE.

La gestion des ventes est une discipline commerciale qui se concentre sur l'application pratique des techniques de vente et la gestion des opérations de vente d'une entreprise.

En effet , il se voit judicieux que ce système soit implémenté.

Pour ce faire, nous avons travaillé sur la conception et la réalisation d'une plateforme permettant de visualiser l'ensemble des achats effectués par les clients. A travers laquelle nous aurons des informations qui peuvent nous servir dans plusieurs processus dans le domaine du commerce.

ABSTRACT

This document summarizes the work carried out within the framework of the JEE project at the National School of Computer Science and Systems Analysis. This project's mission is to design and develop a web application based on JavaEE technologies.

Sales management is a business discipline that focuses on the practical application of sales techniques and the management of a company's sales operations.

Indeed, it makes sense that this system be implemented.

To do this, we worked on the design and implementation of a platform to view all of the purchases made by customers. Through which we will have information which can serve us in several processes in the field of trade.

Liste des abréviations

- **UML** : Unified Modeling Language
- **JSF** : JavaServer Faces
- **MVC** : Modèle-vue-contrôleur
- **DAO** : Data Access Object
- **XML** : Extensible Markup Language
- **HTTP** : Hypertext Transfer Protocol
- **JEE** : Java Enterprise Edition
- **HTML** : HyperText Markup Language
- **CSS** : Cascading Style Sheets
- **AJAX** : JavaScript and XML

Table des figures

2.1	Diagramme des cas d'utilisation de l'admin	15
2.2	Diagramme des cas d'utilisation de l'utilisateur	16
2.3	Diagramme de Séquence de gestion des clients	17
2.4	Diagramme de Séquence de gestion des commandes	18
2.5	Diagramme de Séquence de gestion des utilisateurs	19
2.6	Diagramme de classe	20
3.1	Schéma de l'architecture	25
3.2	Schéma du Pattern MVC	26
3.3	Ajout commande	27
3.4	Liste des utilisateurs	28
3.5	Ajout article	29
3.6	Accueil	30
3.7	Liste des articles	31
3.8	Ajout client	31
3.9	Liste des clients	32
3.10	Ajout utilisateur	32
3.11	Liste des commandes	33

Table des matières

Liste des abréviations	4
INTRODUCTION	8
1 Contexte général du projet	9
1.1 Introduction	10
1.2 Description du projet	10
1.2.1 Contexte général du projet	10
1.2.2 Problématique	10
1.2.3 Objectifs	10
1.3 Conclusion	11
2 ANALYSE ET CONCEPTION	12
2.1 Introduction	13
2.2 Spécification des besoins	13
2.2.1 Besoins fonctionnels	13
2.2.2 Besoins non fonctionnels	13
2.3 Analyse des besoins	14
2.3.1 Identification des acteurs	14
2.3.2 Diagramme de cas d'utilisation	14
2.4 Conception de données	16
2.4.1 Diagramme de séquence	16
2.4.2 Diagrammes de classes	19
2.5 Conclusion	19
3 REALISATION	21
3.1 Introduction	22
3.2 Environnement de développement	22
3.2.1 Outils utilisés	22
3.2.2 Technologies et langages utilisées	23
3.2.3 La démarche de développement	24
3.3 Implémentation des interfaces	27
CONCLUSION	34

INTRODUCTION

De petite ou de grande envergure, une société est amenée à effectuer des ventes pour pouvoir survivre. La rentabilité est sa principale mission. Afin d'assurer l'atteinte de cet objectif, l'établissement ne doit en aucun cas négliger l'aspect commercial et ici vient le rôle d'informatique pour automatiser et faciliter les choses. Certes notre pays dispose de plusieurs petites coopératives, ces dernières ont besoin d'avoir des applications web et mobiles pour augmenter leurs ventes et présenter leurs produits aux clients aussi pour dépasser l'utilisation des carnets de notes ou les équipements ultra chers du passé, d'ici vient notre idée d'application, notre objectif est de créer une application Java EE qui facilite aux petites coopératives le processus de vente et leurs permet de synchroniser sans effort les opérations et l'offre commerciale de toutes leurs activités.

Partant de cet objectif , le projet a été réalisé en trois grandes étapes colligées dans le présent rapport :

- En premier lieu , une présentation du contexte général ainsi que l'identification de la problématique et la définition des objectifs du projet.
- Nous abordons par la suite l'analyse et la spécification des besoins, la présentation de l'architecture respectée lors de l'implémentation de notre plateforme, ainsi que la conception du projet, traduite par la réalisation d'un ensemble de diagrammes.
- Finalement, le dernier chapitre concerne les outils utilisés lors de la réalisation de ce projet, ainsi que la présentation des différentes interfaces implémentées et leur fonctionnement.

Chapitre 1

Contexte général du projet

1.1 Introduction

Dans ce chapitre, nous allons situer le projet dans son contexte général, ceci en présentant sa problématique ainsi que les besoins et les objectifs déterminés pour la résoudre.

1.2 Description du projet

1.2.1 Contexte général du projet

Notre application jouera le rôle d'un ERP qui va permettre de gérer le processus de ventes.

A travers cette plateforme, la coopérative pourra contrôler l'ensemble du processus de vente, elle pourra avoir les statistiques de vente, visualiser qui sont ses clients et leur nombre ainsi les commandes qu'ils ont effectuées.

1.2.2 Problématique

La plupart des gens préfèrent consommer des produits naturels qui connaissent leur origine, ce qui explique la demande croissante de biens proposés par un groupe de coopératives, ces dernières se retrouvent face au défi de gérer les biens qui ont été vendus. Par conséquent, nous avons décidé de créer une application qui soulagerait les dirigeants, de ces petites entreprises, de ces problèmes.

1.2.3 Objectifs

La complexité de ces processus nous a incité à penser élaborer une solution convenable afin de résoudre les problèmes mentionnés ci-dessus, ceci en réalisant une application permettant de gérer le processus de ventes d'une coopérative.

Afin de répondre à cet objectif, ce projet repose sur la conception et la réalisation d'une plateforme de gestion de vente en ligne.

- Une partie dédiée au admin, où il a tous les privilèges d'ajout, de modification et de suppression des utilisateurs, des articles, des clients et des commandes.

- L'autre partie est dédiée pour des utilisateurs qui sont des employés de cette coopérative, où ils pourront :
 - Ajouter, modifier et supprimer une commande
 - Ajouter, modifier et supprimer un client
 - Ajouter, modifier et supprimer un article

1.3 Conclusion

Finalement, une coopérative intéressée par ces services de cette application pourrait éviter les procédures classiques et réduire le temps habituel à ces dernières en utilisant cette plateforme.

Chapitre 2

ANALYSE ET CONCEPTION

2.1 Introduction

Dans ce chapitre, nous allons aborder la phase de conception, ceci en clarifiant en premier lieu la vue globale décrite précédemment, en analysant les besoins du système et en décrivant l'architecture générale que nous allons suivre dans la partie réalisation de notre projet. Puis, dans un deuxième lieu nous allons détailler notre choix conceptuel à travers plusieurs diagrammes.

2.2 Spécification des besoins

2.2.1 Besoins fonctionnels

- Visualiser les statistiques des ventes
- Gérer les différents articles (ajout, modification et suppression)
- Gérer les utilisateurs
- Gérer les clients
- Gérer les différentes commandes effectuées par les clients

2.2.2 Besoins non fonctionnels

- **L'extensibilité :**

à tout moment on peut ajouter ou modifier des nouvelles fonctionnalités.

- **La sécurité :**

les informations ne devront pas être accessibles depuis l'extérieur.

- **La performance :**

l'application doit réagir dans un délai précis.

- **La convivialité :**

l'application doit être simple et facile à utiliser.

2.3 Analyse des besoins

Afin de modéliser cette étape, nous avons adopté le langage UML . Ceci permet ainsi : Un gain de précision, un gage de stabilité et l'utilisation d'outils. UML est un support de communication performant : Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse lui font un langage universel.

On a choisi alors de travailler avec UML puisqu'il exprime mieux la vue statique et dynamique des applications web, il est nécessaire alors de faire une analyse très approfondie pour pouvoir dégager les nécessités de développement ainsi que quelques scénarios d'exécution.

2.3.1 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes qui interagissent directement avec le système étudié.

Un acteur est identifié dans notre application :

- **Les employés de ventes** : Utilisateurs et personnes disposant d'un compte et supervisant le processus de vente.

2.3.2 Diagramme de cas d'utilisation

Un cas d'utilisation est une manière spécifique d'utiliser un système. C'est l'image d'une fonctionnalité du système, d'éclenchée en réponse à la stimulation d'un acteur externe. Les cas d'utilisation apportent une solution aux problèmes de la détermination et de la compréhension des besoins.

Le diagramme ci-dessous décrit l'utilisation que permet l'application aux acteurs décrits précédemment :

Diagramme de cas d'utilisation Admin

Un admin disposant d'un compte sur l'application peut donc :

- Ajouter, modifier et supprimer un article.
- Ajouter, modifier et supprimer une commande.
- Ajouter, modifier ou supprimer un client.
- Ajouter, modifier ou supprimer un utilisateur.

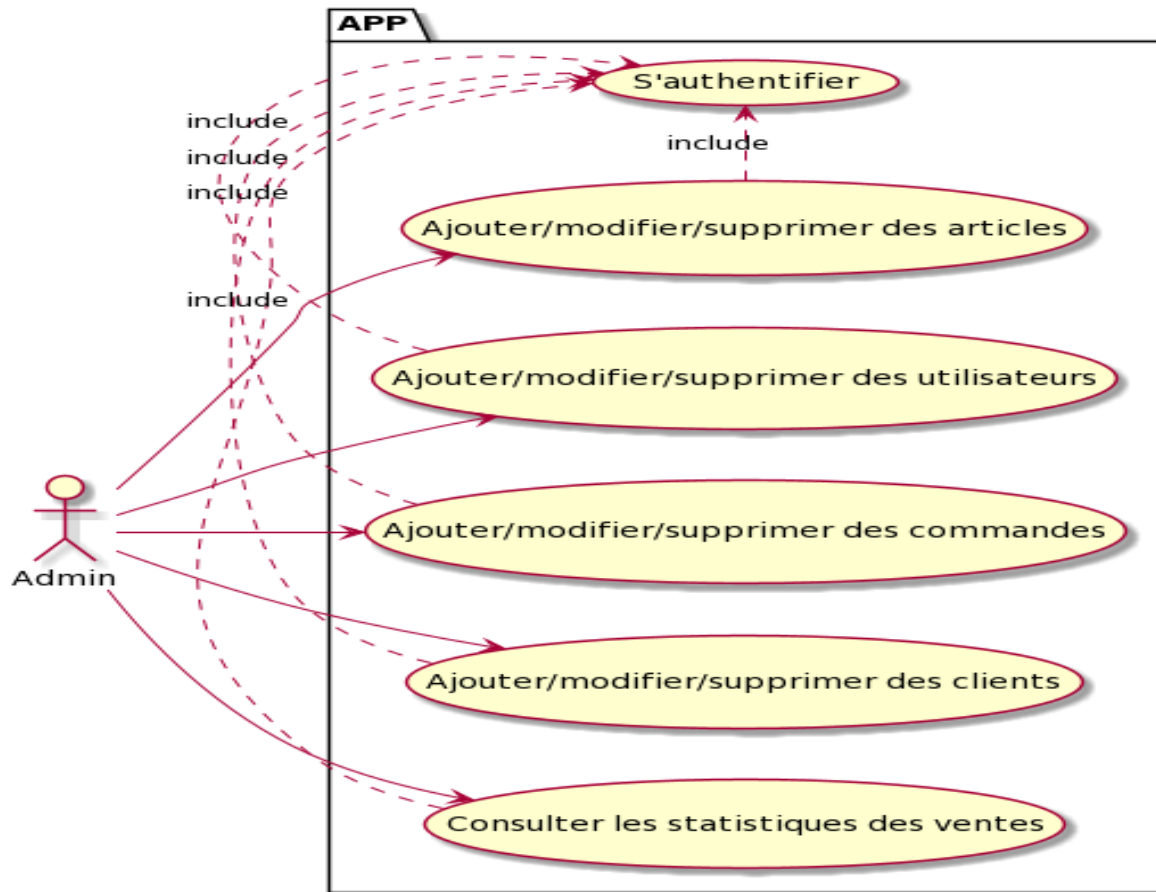


FIGURE 2.1 – Diagramme des cas d'utilisation de l'admin

Diagramme de cas d'utilisation Utilisateur

Un utilisateur disposant d'un compte utilisateur sur l'application peut donc :

- Ajouter, modifier et supprimer un article.
- Ajouter, modifier et supprimer une commande.
- Ajouter, modifier ou supprimer un client.

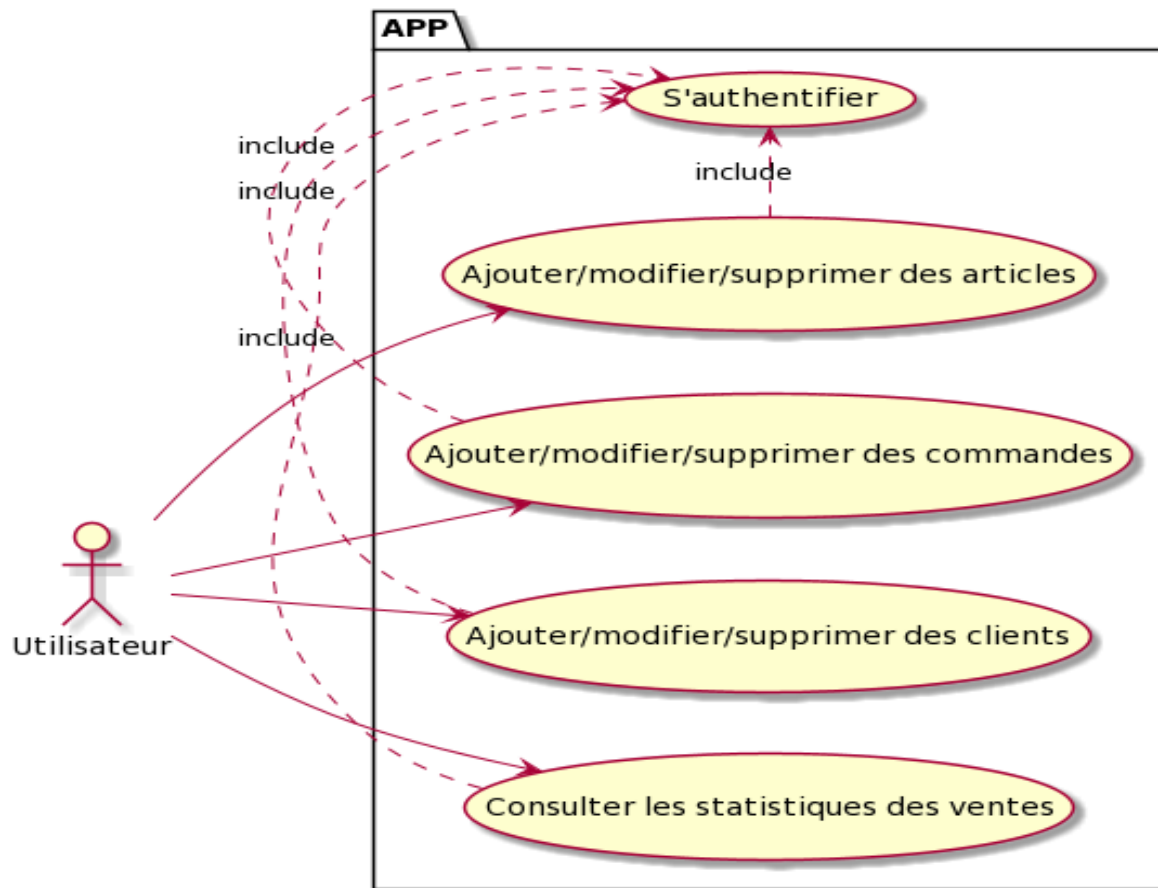


FIGURE 2.2 – Diagramme des cas d'utilisation de l'utilisateur

2.4 Conception de données

2.4.1 Diagramme de séquence

Les diagrammes de séquences :

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

L'admin crée les comptes des utilisateurs, les utilisateurs de l'application sont des employés dans la coopérative cible, ils peuvent réaliser beaucoup de tâches, ces dernières sont exprimées et détaillées dans les diagrammes de séquences suivants :

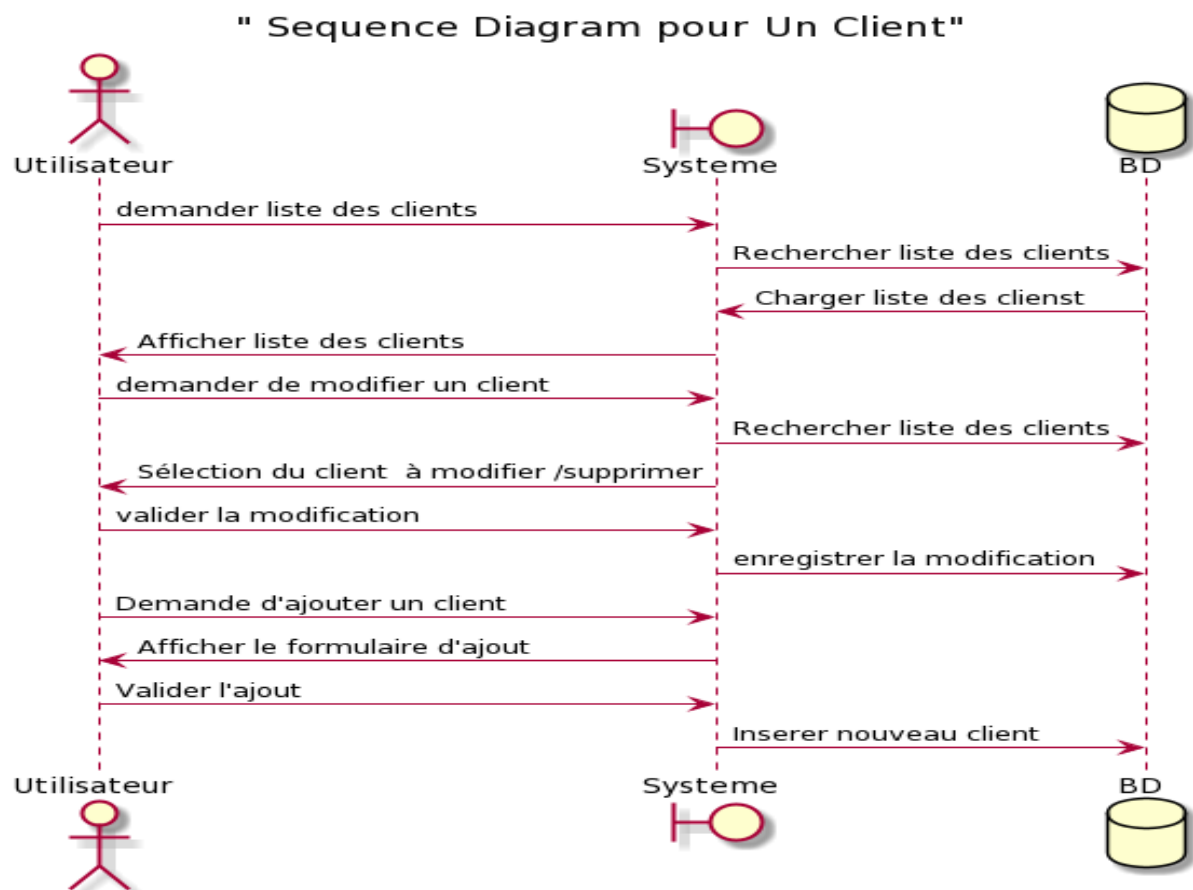


FIGURE 2.3 – Diagramme de Séquence de gestion des clients

" Sequence Diagram pour Une commande"

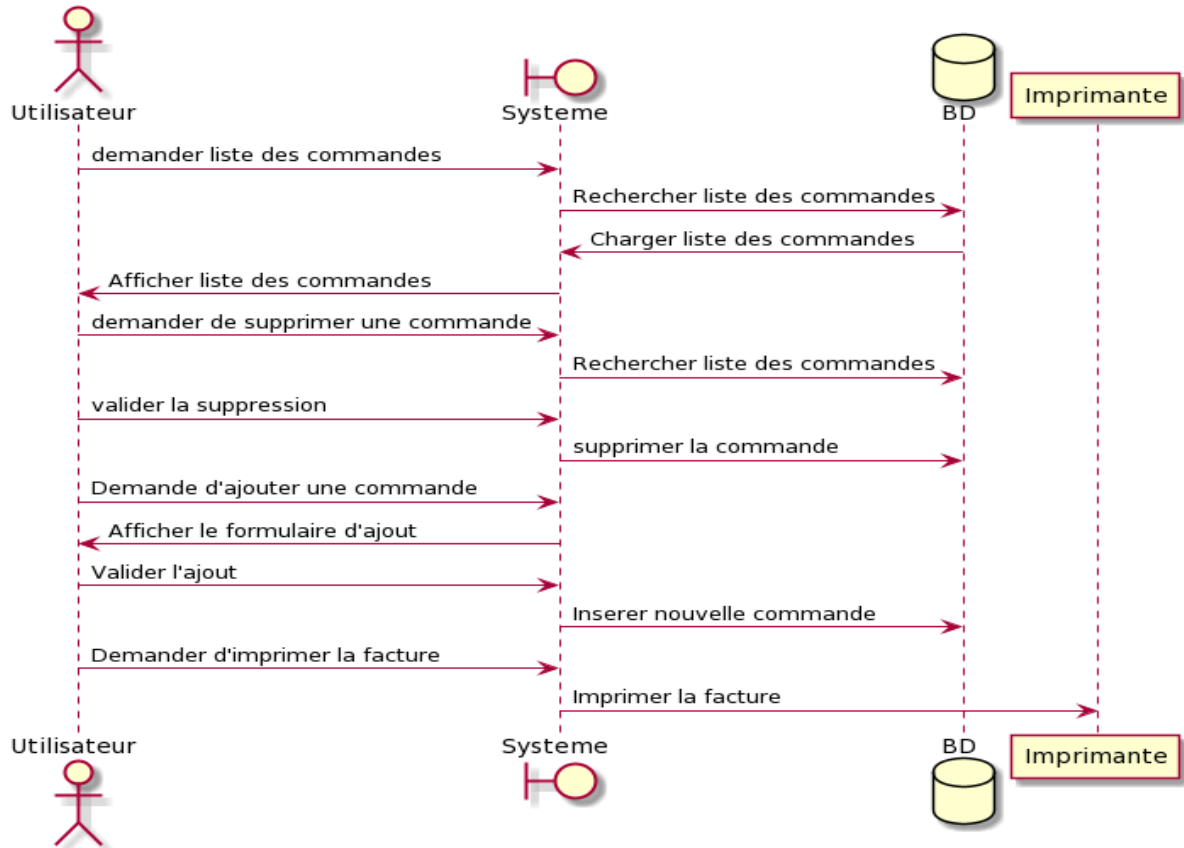


FIGURE 2.4 – Diagramme de Séquence de gestion des commandes

" Sequence Diagram pour Un admin "

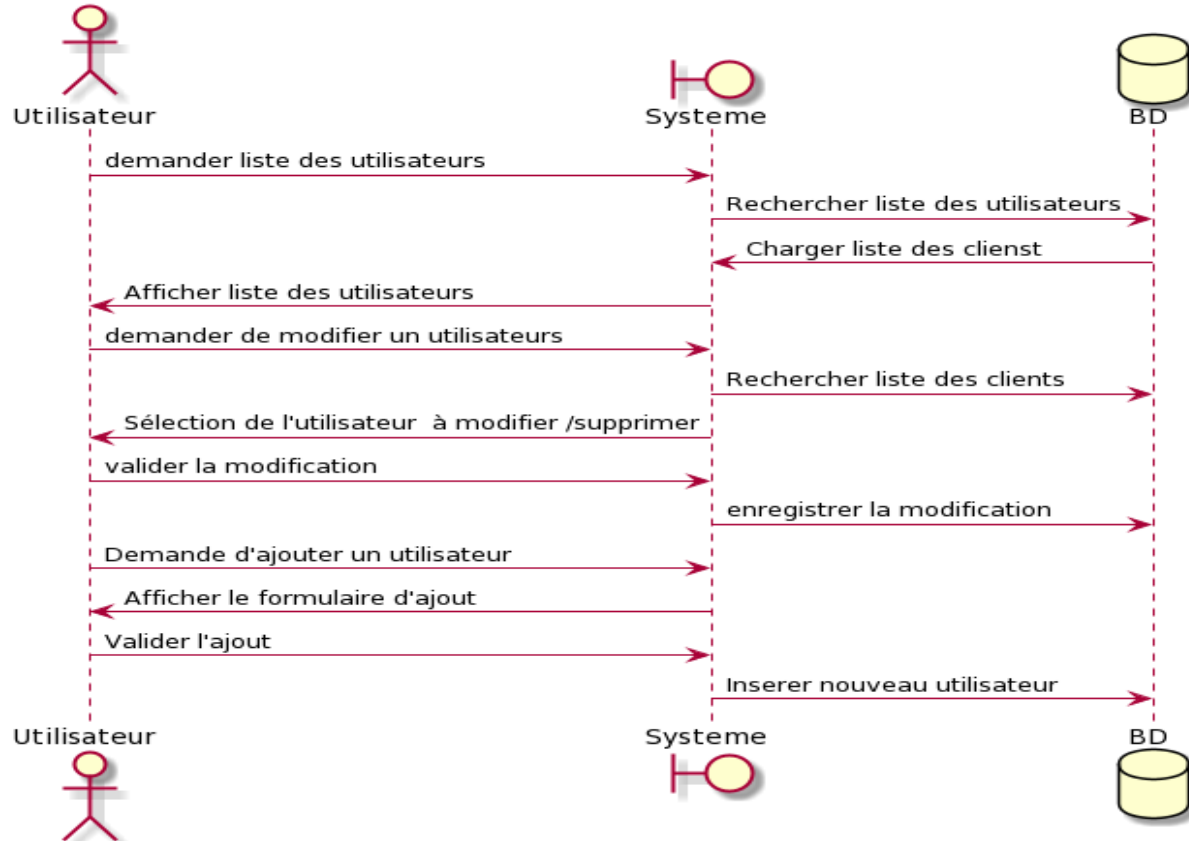


FIGURE 2.5 – Diagramme de Séquence de gestion des utilisateurs

2.4.2 Diagrammes de classes

Le digramme de classe représente l'un des plus importants diagrammes de la modélisation UML lors de la phase de la conception, il représente les différentes classes métiers, leurs attributs et les relations entre elles.

2.5 Conclusion

Dans ce chapitre, nous avons effectués l'analyse et la conception de notre projet à partir des différents diagrammes de modélisation UML. Quant au chapitre suivant, el est consacré à la réalisation de notre application.

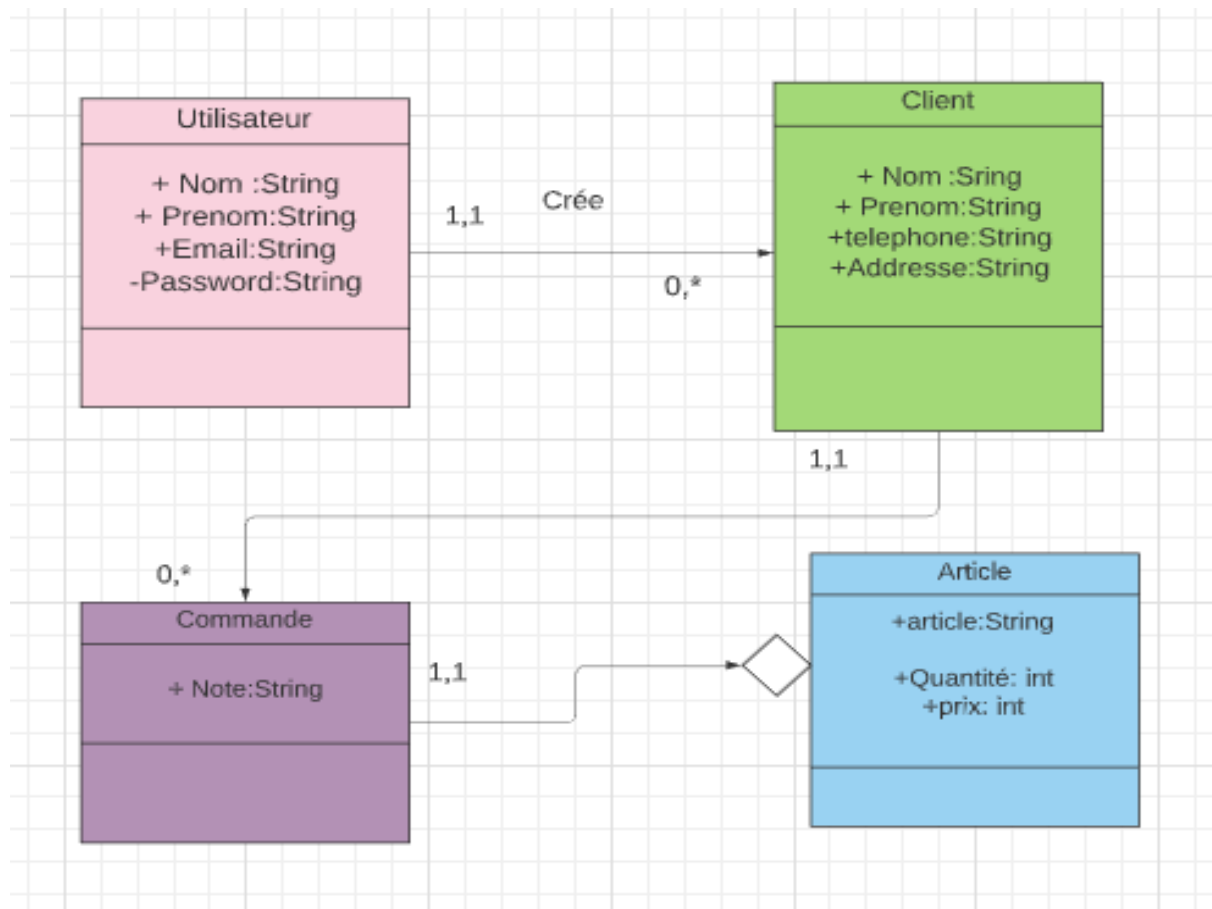


FIGURE 2.6 – Diagramme de classe

Chapitre 3

REALISATION

3.1 Introduction

Dans ce chapitre, nous allons expliquer les outils de développement utilisés pour élaborer notre application, ensuite montrer les interfaces finales de notre application.

3.2 Environnement de développement

3.2.1 Outils utilisés

Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

Apache Tomcat

Apache Tomcat est un conteneur web libre de servlets et JSP. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

phpMyAdmin

phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL.

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder commodément une base

de données sous forme de fichier .sql et d'y transférer ses données, même sans connaître SQL.

3.2.2 Technologies et langages utilisées

Java EE

Java EE est une collection de technologies et d'API pour la plate-forme Java conçue pour prendre en charge les applications "d'entreprise" qui peuvent généralement être classées comme des applications à grande échelle, distribuées, transactionnelles et hautement disponibles, conçues pour prendre en charge les exigences commerciales critiques.

La technologie JSF

Java Server Faces (JSF) est une technologie dont le but est de proposer un framework qui facilite et standardise le développement d'applications web avec Java. Son développement a tenu compte des différentes expériences acquises lors de l'utilisation des technologies standard pour le développement d'applications web (servlet, JSP, JSTL) et de différents frameworks (Struts, ...).

Hibernate

Hibernate est un framework open source gérant la persistance des objets en base de données relationnelle.

Hibernate est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement Java EE complet : WebSphere, JBoss Application Server et Oracle WebLogic Server.

HTML

Le langage HTML est un langage permettant de définir la structure logique d'un document à l'aide de différents éléments de base (titres, paragraphes, liens vers d'autres documents, ...).

HTML n'est pas un langage de traitement de texte en ce sens qu'il ne permet pas de décrire les détails de mise en forme des divers éléments composant un document. La mise en forme dépend du navigateur et des options choisies par le lecteur. Le résultat de la visualisation peut donc varier mais la structuration logique est préservée dans tous les cas.

CSS

Les CSS, Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle.

JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs² avec l'utilisation (par exemple) de Node.JS.

C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototype qui permet d'en créer des objets héritiers personnalisés.

Bootstrap

Bootstrap est une collection d'outils utiles 'à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres 'éléments interactifs, ainsi que des extensions JavaScript en option.

3.2.3 La démarche de développement

Architecture technique de notre application

La classe service représente la couche métier de notre application, chaque entité a sa propre classe métier qui permet de faire les différents traitements liés à cette entité.

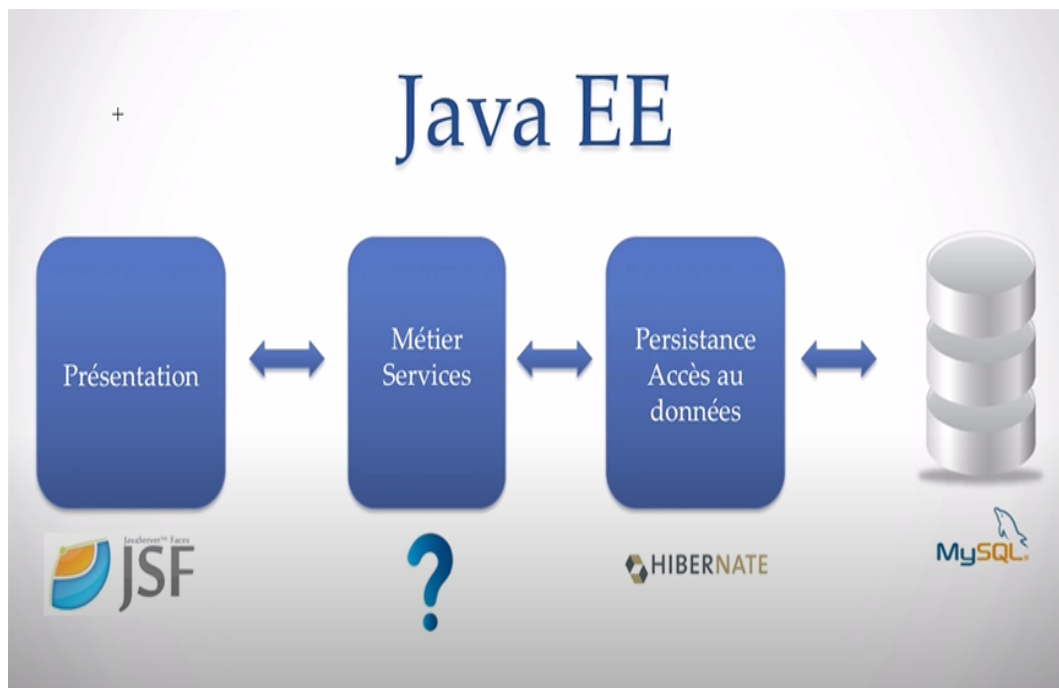


FIGURE 3.1 – Schéma de l'architecture

Patron MVC

MVC est un patron de conception (design pattern en anglais) très répandu pour réaliser des sites web. Ce patron de conception est une solution éprouvée et reconnue permettant de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données.

Rôles des composants

- La partie Modèle d'une architecture MVC encapsule la logique métier ainsi que l'accès aux données. Il peut s'agir d'un ensemble de fonctions (Modèle procédural) ou de classes (Modèle orienté objet).
- La partie Vue s'occupe des interactions avec l'utilisateur : présentation, saisie et validation des données.
- La partie Contrôleur gère la dynamique de l'application. Elle fait le lien entre l'utilisateur et le reste de l'application.

Ajax

Ajax est une architecture informatique qui permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et

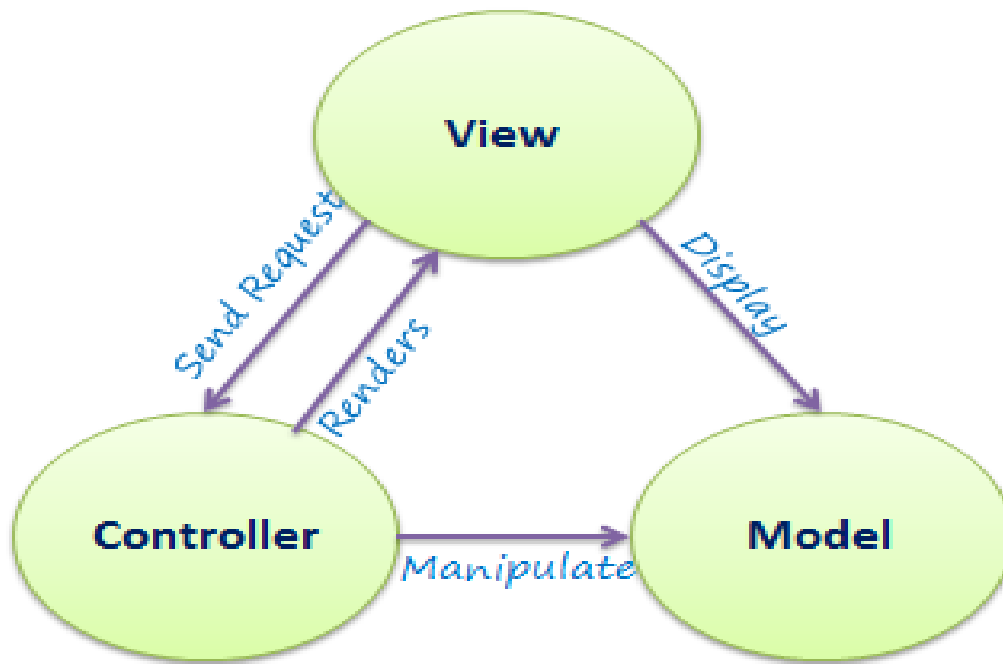


FIGURE 3.2 – Schéma du Pattern MVC

2005. Ajax est l'acronyme d'asynchronous JavaScript and XML : JavaScript et XML asynchrones.

Ajax combine JavaScript et DOM, qui permettent de modifier l'information présentée dans le navigateur en respectant sa structure, les API Fetch et XMLHttpRequest, qui servent au dialogue asynchrone avec le serveur Web ; ainsi qu'un format de données (XML ou JSON), afin d'améliorer maniabilité et confort d'utilisation des applications internet riches. XML, cité dans l'acronyme, était historiquement le moyen privilégié pour structurer les informations transmises entre serveur Web et navigateur, de nos jours le JSON tend à le remplacer pour cet usage.

Modèle DAO

L'utilisation de DAO permet de s'abstraire de la façon dont les données sont stockées au niveau des objets métier. Ainsi, le changement du mode de stockage ne remet pas en cause le reste de l'application. En effet, seules ces classes dites "techniques" seront à modifier (et donc à re-tester). Cette souplesse implique cependant un coût additionnel, dû à une plus grande complexité de mise en œuvre.

Design Factory Singleton

En génie logiciel, le singleton est un patron de conception (design pattern) dont l'objectif est de restreindre l'instanciation d'une classe à un seul objet (ou

bien à quelques objets seulement). Il est utilisé lorsqu'on a besoin exactement d'un objet pour coordonner des opérations dans un système. Le modèle est parfois utilisé pour son efficacité, lorsque le système est plus rapide ou occupe moins de mémoire avec peu d'objets qu'avec beaucoup d'objets similaires.

3.3 Implémentation des interfaces

Réalisation des interfaces

The screenshot shows a web application interface for adding a new order. On the left is a dark sidebar with navigation links: 'Tableau de Bord', 'Articles', 'Commandes' (highlighted in red), 'Clients', and 'Utilisateurs'. The main content area has a header 'Commandes' with a home icon. Below the header is the title 'Ajouter une nouvelle commande'. The form contains four fields: 'Client *' with a dropdown menu showing '--Choisir--' and a link 'nouveau client'; 'Article *' with a dropdown menu showing '--Choisir--' and a link 'nouveau article'; 'Notes' with a text input field containing the placeholder 'entrer des notes'; and 'Quantité *' with a text input field containing the placeholder 'Entrer la quantité' and a note 'Entrer que des nombres'. At the bottom of the form are two buttons: 'Ajouter' (green) and 'Annuler' (white). A Windows watermark 'Activer Windows' is visible in the bottom right corner.

FIGURE 3.3 – Ajout commande

Tableau de Bord

Articles

Commandes

Clients

Utilisateurs

Articles

Liste des Articles

Nouveau Utilisateur

Code	Nom	Prenom	Email	Modifier	Supprimer
3	wardi	hafssa	hafssa@gmail.com		
2	Zeroual	Najjae	najjae@gmail.com		
4	maujib	sara	sara@gmail.com		
7	admin	admin	attariuas@gmail.com		

FIGURE 3.4 – Liste des utilisateurs

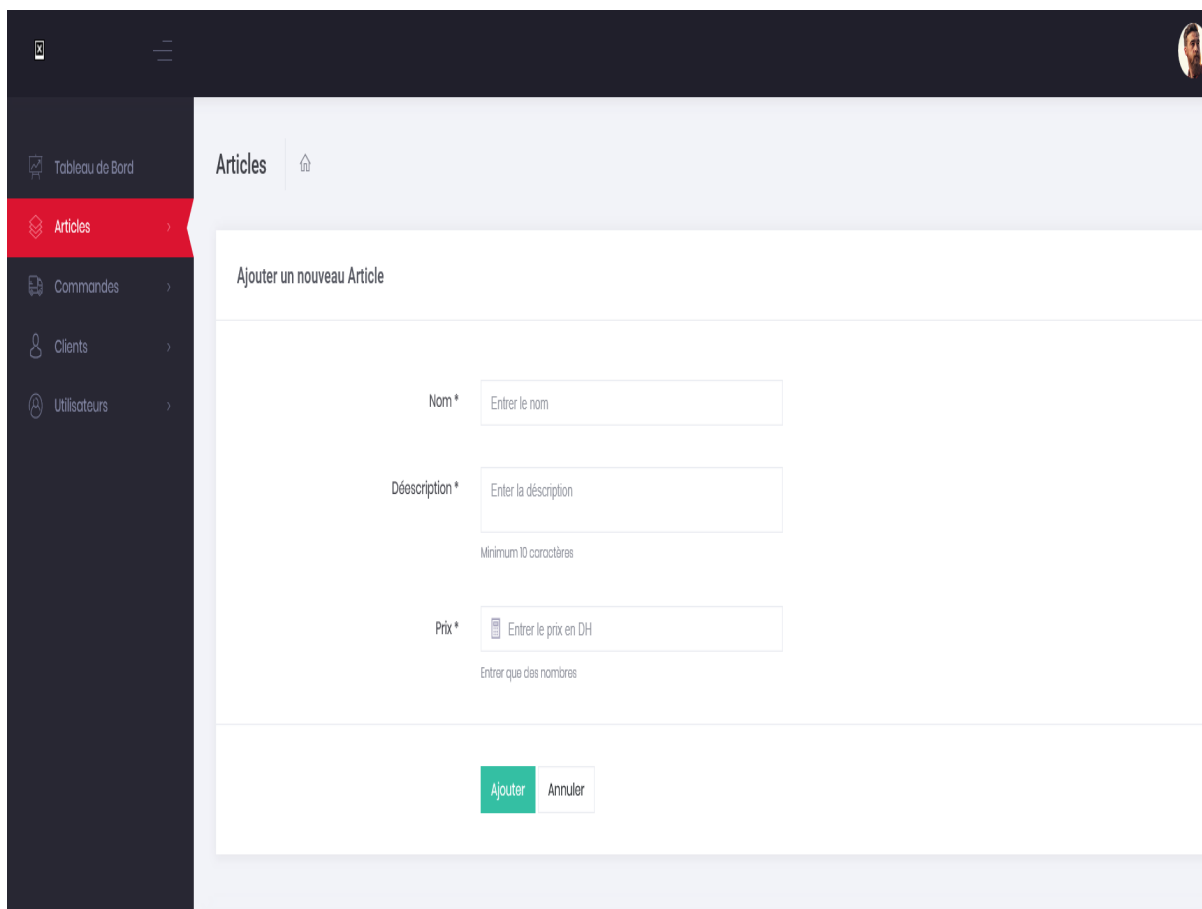


Tableau de Bord

Articles

Commandes

Clients

Utilisateurs

Articles

Ajouter un nouveau Article

Nom * Entrez le nom

Déscription * Entrez la description

Minimum 10 caractères

Prix * Entrez le prix en DH

Entrez que des nombres

Ajouter Annuler

FIGURE 3.5 – Ajout article

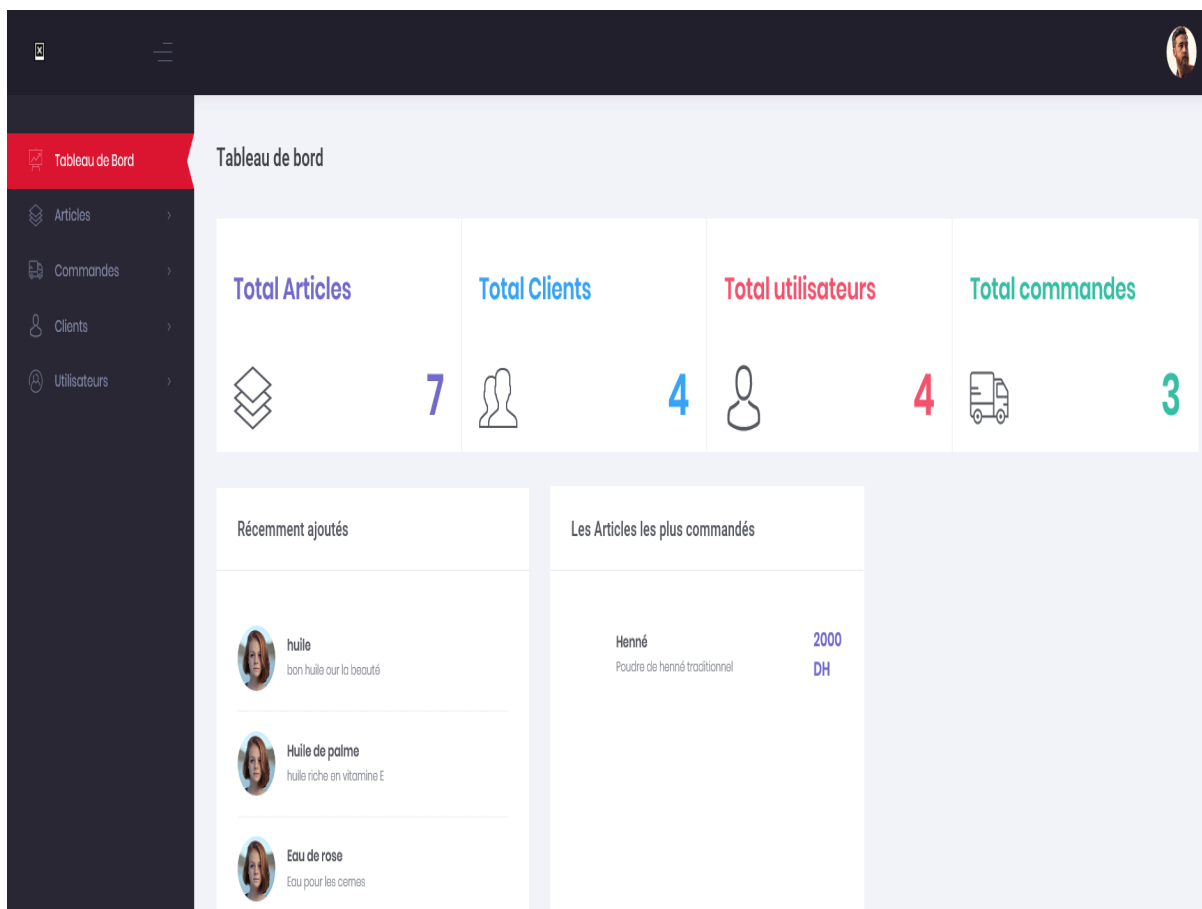


FIGURE 3.6 – Accueil

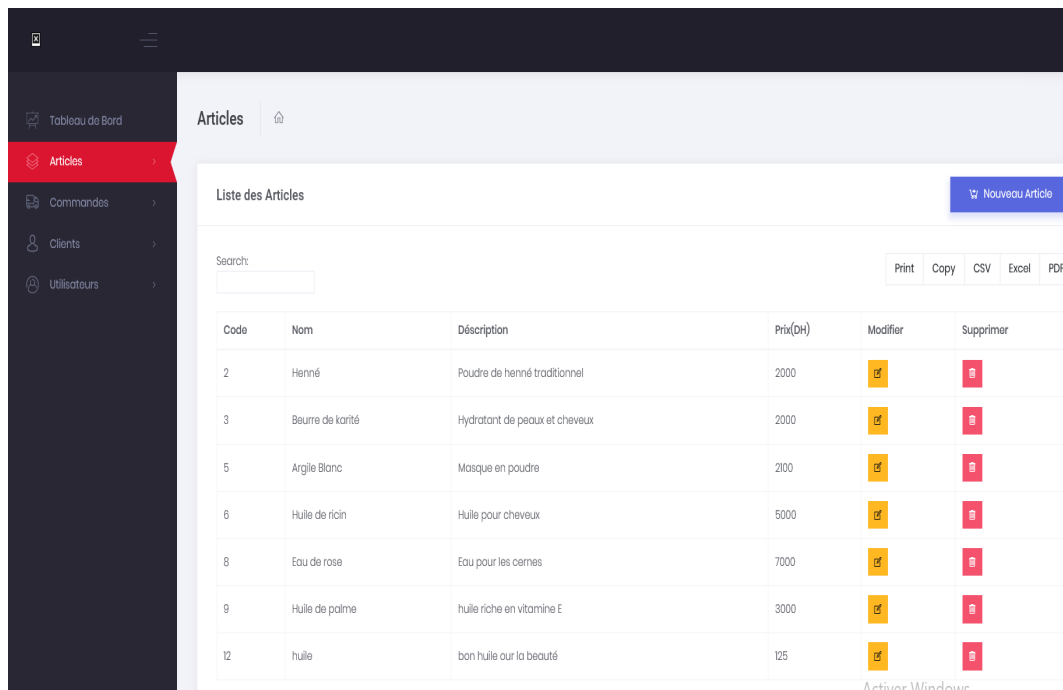


FIGURE 3.7 – Liste des articles

Clients

Ajouter un nouveau Client

Nom *

Prenom *

Téléphone *
Entrer que des nombres

Adresse *

[Ajouter](#) [Annuler](#)

FIGURE 3.8 – Ajout client

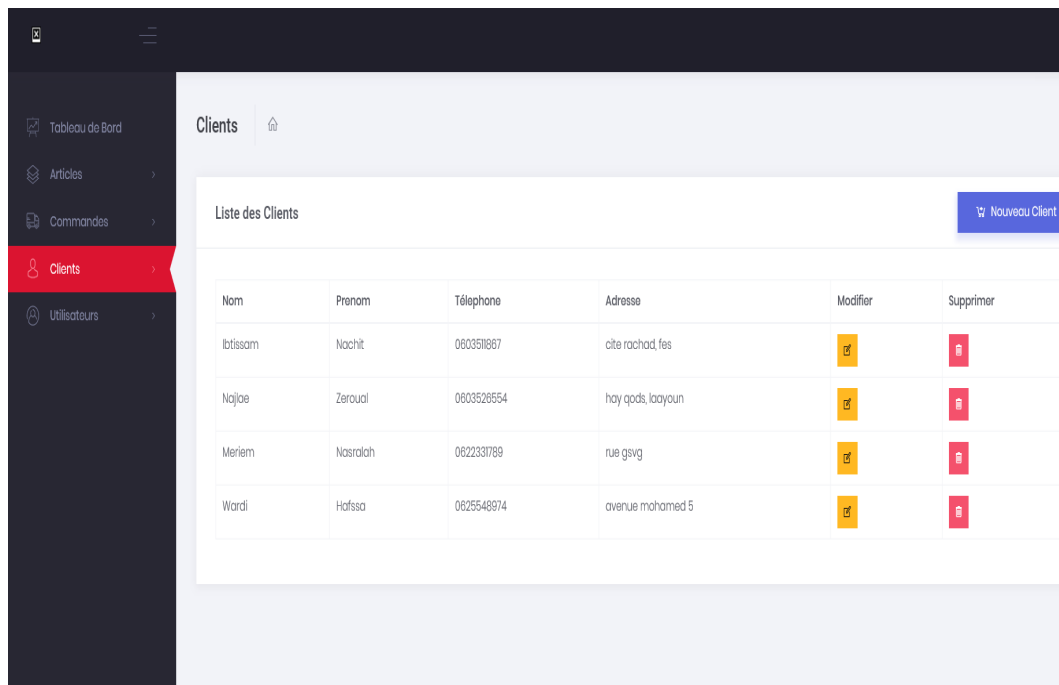


FIGURE 3.9 – Liste des clients

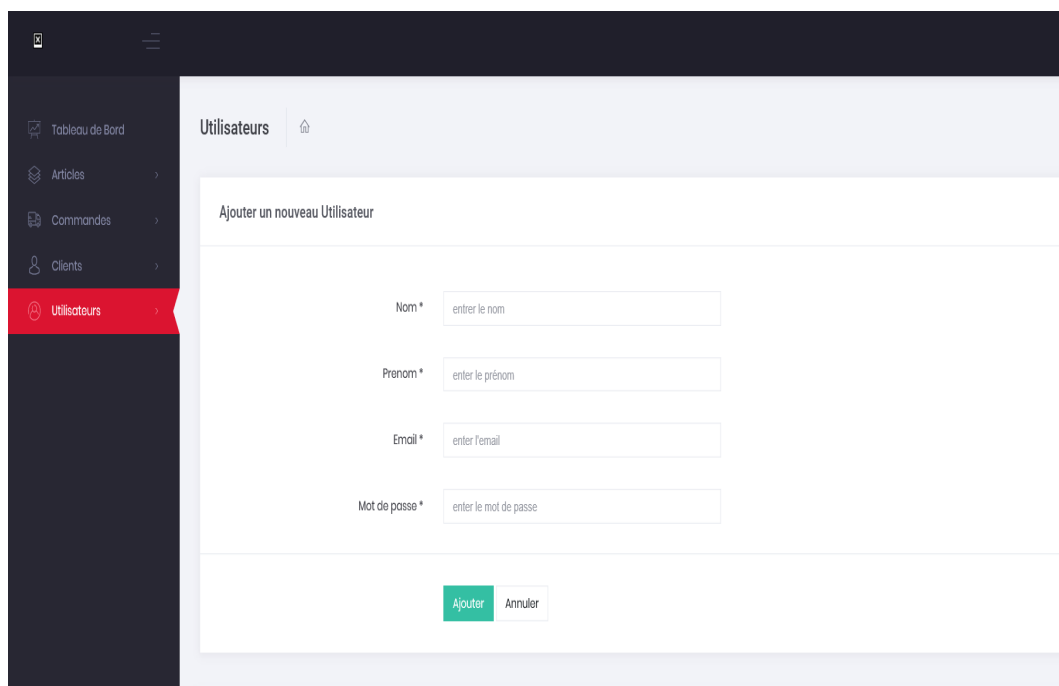


FIGURE 3.10 – Ajout utilisateur

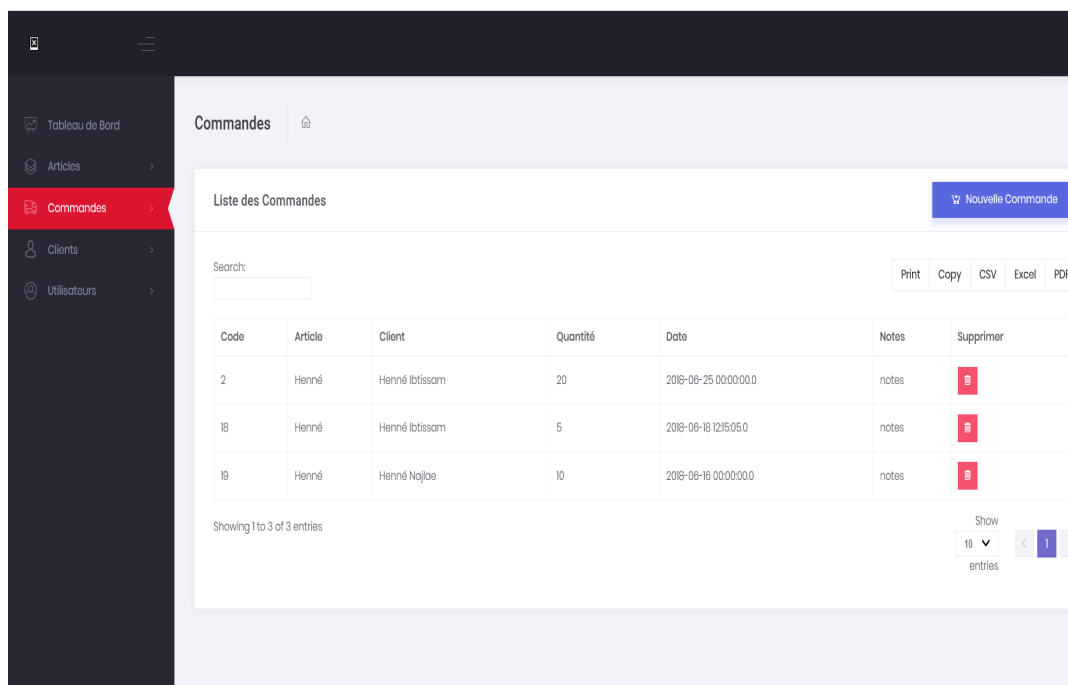


FIGURE 3.11 – Liste des commandes

CONCLUSION

Ce projet consistait à développer une application web de gestion des ventes d'une coopérative. Tout au long de ce rapport, nous avons présenté la démarche que nous avons suivie pour mettre au point cette application. Afin de réaliser notre application et atteindre les objectifs que nous avons visés, le choix s'est porté sur la méthode UML pour modéliser l'analyse et la conception de notre application, et tenter de répondre aux besoins exprimés précédemment. Ce qui importait aussi est l'étape de la réalisation du projet, qui s'est faite grâce à la plateforme Java EE en utilisant la démarche MVC.

Le projet a été une expérience très enrichissante et bénéfique pour nous. Il nous a permis d'approfondir nos connaissances, d'augmenter nos compétences et de découvrir des langages divers qui s'avèrent très utiles pour notre cursus en tant qu'ingénieurs en génie logiciel. Ce projet nous a permis aussi de faire face à plusieurs contraintes et de développer les techniques de recherche d'information. De plus, nous avons pu acquérir l'esprit du travail de groupe est ainsi q'appréhender l'aspect humain, écouter les autres, défendre nos idées, se partager le travail et prendre des initiatives.

Webographie

Documentation de phpMyAdmin : <https://fr.wikipedia.org/wiki/PhpMyAdmin>.

Documentation de CSS : https://en.wikipedia.org/wiki/Cascading_Style_Sheets

Documentation Java EE : <https://docs.oracle.com/javaee/7/tutorial/>

Documentation de HTML : <https://en.wikipedia.org/wiki/HTML>

Documentation de Ajax [https://fr.wikipedia.org/wiki/Ajax_\(informatique\)](https://fr.wikipedia.org/wiki/Ajax_(informatique))

Documentation de Hibernate : <https://fr.wikipedia.org/wiki/Hibernate>

Documentation de Design Factory Singleton : [https://fr.wikipedia.org/wiki/Singleton_\(patron_de_conception\)](https://fr.wikipedia.org/wiki/Singleton_(patron_de_conception))

Cours magistral et tps de M. ElHamlaoui

Lien Github du projet : <https://github.com/ibtissammoufid/ProjetJavaEE>

Lien Balsamiq des maquettes : <https://balsamiq.cloud/sb4e75d/p8uq1dx/rE760>