

République Algérienne Démocratique et Populaire Ministère de  
l'Enseignement Supérieur et de la Recherche  
Université Abderrahmane Mira de Béjaïa  
Faculté des Sciences Exactes  
Département Informatique

## Mémoire de fin d'étude

En vue de l'obtention du Diplôme de Master Professionnel en  
Informatique

Option:

**GENIE LOGICIEL**



**Thème :**

**Conception et réalisation d'une application web  
pour la gestion et le suivi des conteneurs de l'EPB**

**Réalisé par**

HARKOUK Warda  
OURABAH Melissa

**Soutenu le 9 septembre 2020 devant le jury composé de**

<b>Encadrant</b>	<b>Mr ATMANI Mouloud</b>	<b>M.C.B</b>	<b>Université de Béjaïa</b>
<b>Président</b>	<b>Mr MOKTEFI Mohand</b>	<b>M.C.B</b>	<b>Université de Béjaïa</b>
<b>Examineur</b>	<b>Mr DJEBARI Nabil</b>	<b>M.A.A</b>	<b>Université de Béjaïa</b>
<b>Invité</b>	<b>Mr BEDAUCHE Belal</b>	<b>Ingénieur</b>	<b>EPB</b>

**Année universitaire 2019/2020**

# Remerciements

*Nous tenons tout d'abord à remercier **DIEU** le tout puissant de nous avoir donné la santé et la volonté afin d'élaborer ce projet.*

*Ensuite nous remercions :*

*Nos encadreurs : Monsieur ATMANI Mouloud au niveau de l'université Abderrahmane Mira, Monsieur BEDAOUCHE Belal au niveau de l'entreprise d'accueil EPB pour leurs encadrements qui nous a été d'une grande utilité et leurs présences à tout moment de la réalisation de ce modeste projet, sans lesquels il n'aurait jamais vu le jour, ainsi que les membres du jury : Monsieur DJEBARI Nabil et Monsieur MOKTEFI Mohand, pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.*

*Nous remercions également tous les professeurs qui ont contribué de près ou de loin à notre formation universitaire, qui nous a aidé à mener à terme notre projet.*

# *Dédicace*

*C'est avec un immense plaisir que je dédie ce travail*

*À mes chers parents en témoignage et en gratitude de leurs dévouements, de leur soutien  
permanent durant*

*toutes mes années d'études (que dieu les gardes).*

*À mes grands-parents pour leurs présences, soutiens et leurs tendresses.*

*À mes deux frères Mohmed et Mahrez.*

*À ma tante et mes oncles.*

*À mes cousins (e) et à tous mes amis.*

*À toute la famille OURABAH et BELHOCINE.*

**OURABAH Melissa**

## *Dédicace*

*C'est avec un immense plaisir que je dédie ce modeste travail à mon  
cher et tendre père **HARKOUK Abdelaziz**,  
je ne le remercierai jamais assez pour son soutien moral et matériel,  
je lui exprime toute ma tendresse et ma gratitude.  
Que Dieu le protège et le préserve pour nous.*

*À la mémoire de la plus tendre, de celle que tous les mots du monde ne pourront décrire,  
ma chère maman Ghania.*

*À ma belle-mère Fadila.*

*À mes précieuses sœurs, les plus belles et les plus adorables que j'aime tant Yasmine,  
Meriem et Nesrine.*

*À mon petit cœur, mon petit frère Zakari.*

**HARKOUK Warda**

# TABLE DES MATIÈRES

<b>Table des Matières</b>	<b>i</b>
<b>Liste des figures</b>	<b>vi</b>
<b>Liste des tableaux</b>	<b>vii</b>
<b>Liste des abréviations</b>	<b>viii</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Présentation de l'organisme d'accueil &amp; méthodologie de développement</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Présentation de l'organisme d'accueil . . . . .	3
1.2.1 Historique de l'EPB . . . . .	3
1.2.2 Position géographique de l'EPB . . . . .	4
1.2.3 Missions de l'EPB . . . . .	4
1.2.4 Activités de l'EPB . . . . .	5
1.3 Structures de l'EPB . . . . .	5
1.3.1 La direction de la zone logistique extra-portuaire (DZLE) . . . . .	6
1.3.2 Le département de la zone logistique extra-portuaire de TIXTER . . . . .	7
1.4 Vocabulaire spécifique à l'EPB . . . . .	7
1.5 Étude de l'existant . . . . .	8
1.6 Problématique . . . . .	10

1.7	Objectifs et solutions . . . . .	11
1.8	Démarche de développement . . . . .	11
1.8.1	Processus unifié (Unified Process) . . . . .	12
1.8.2	XP (eXtreme Programming) . . . . .	13
1.8.3	UML (Unified Modeling Language) . . . . .	14
1.9	Conclusion . . . . .	15
<b>2</b>	<b>Spécification des besoins</b>	<b>16</b>
2.1	Introduction . . . . .	16
2.2	Besoins de l'entreprise d'accueil . . . . .	16
2.2.1	Besoins fonctionnels . . . . .	16
2.2.2	Besoins non fonctionnels . . . . .	17
2.3	Identification des acteurs . . . . .	18
2.4	Diagramme de contexte dynamique . . . . .	19
2.4.1	Identification des messages échangés . . . . .	19
2.5	Identification des cas d'utilisation . . . . .	21
2.5.1	Diagramme de cas d'utilisation . . . . .	22
2.5.2	Description textuelle des cas d'utilisation . . . . .	24
2.6	Conclusion . . . . .	29
<b>3</b>	<b>Analyse et conception</b>	<b>30</b>
3.1	Introduction . . . . .	30
3.2	Diagramme de séquence système . . . . .	30
3.2.1	Diagramme de séquence système « Authentification » . . . . .	31
3.2.2	Diagramme de séquence système de cas d'utilisation « Ajouter Conteneur » . . . . .	31
3.2.3	Diagramme de séquence système de cas d'utilisation « Ajouter Marchandise » . . . . .	32
3.2.4	Diagramme de séquence système de cas d'utilisation « Ajouter Article BL » . . . . .	34
3.2.5	Diagramme de séquence système de cas d'utilisation « Modifier Article BL » . . . . .	35

3.2.6	Diagramme de séquence système de cas d'utilisation « Supprimer Article BL » . . . . .	36
3.3	Diagramme de classes . . . . .	37
3.4	Modèle relationnel . . . . .	39
3.5	Dictionnaire de données . . . . .	42
3.6	Conclusion . . . . .	45
<b>4</b>	<b>Réalisation</b>	<b>46</b>
4.1	Introduction . . . . .	46
4.2	Application web . . . . .	46
4.3	Classement des langages de programmation et Frameworks . . . . .	47
4.4	Framework Spring . . . . .	48
4.4.1	Généralités . . . . .	48
4.4.2	Définition . . . . .	51
4.4.3	JPA /Hibernate . . . . .	52
4.4.4	Modules du framework Spring . . . . .	52
4.4.5	Architecture du Framework Spring MVC . . . . .	53
4.4.6	Fonctionnement du Framework Spring MVC . . . . .	55
4.5	Langages et outils de développement . . . . .	56
4.6	Notre application de gestion des conteneurs de l'EPB . . . . .	59
4.6.1	Interface d'authentification . . . . .	59
4.6.2	Interface Article BL . . . . .	60
4.6.3	Interface Ajouter Article BL . . . . .	61
4.6.4	Interface Modifier Article BL . . . . .	61
4.6.5	Interface Archive . . . . .	62
4.6.6	Interfaces Statistiques . . . . .	63
4.6.7	Interface Utilisateurs . . . . .	65
4.6.8	Modifier Profil Utilisateur . . . . .	65
4.7	Tests . . . . .	67
4.8	Déploiement . . . . .	67
4.8.1	Création du fichier War du projet Spring MVC . . . . .	67
4.8.2	Déploiement du fichier WAR à l'aide d'Apache Tomcat Manager . . . . .	68

4.9 Conclusion . . . . .	69
<b>Conclusion générale</b>	<b>71</b>
<b>Bibliographie</b>	<b>73</b>



## LISTE DES FIGURES

1.1	Organigramme général de l'Entreprise Portuaire de Bejaia [source EPB]. . . . .	6
1.2	Organigramme de la DZLE [source EPB]. . . . .	7
1.3	Déroulement du processus Up [2]. . . . .	13
2.1	Diagramme de contexte dynamique. . . . .	19
2.2	Diagramme de cas d'utilisation globale du système. . . . .	23
3.1	Diagramme de séquence système de cas d'utilisation « Authentification ». . . . .	31
3.2	Diagramme de séquence système de cas d'utilisation « Ajouter Conteneur ». . . . .	32
3.3	Diagramme de séquence système de cas d'utilisation « Ajouter Marchandise ». . . . .	33
3.4	Diagramme de séquence système de cas d'utilisation « Ajouter Article BL ». . . . .	34
3.5	Diagramme de séquence système de cas d'utilisation « Modifier Article BL ». . . . .	35
3.6	Diagramme de séquence système de cas d'utilisation « Supprimer Article BL ». . . . .	36
3.7	Diagramme de classes. . . . .	38
3.8	Relation un à plusieurs. . . . .	39
3.9	Application de la première règle. . . . .	40
3.10	Relation plusieurs à plusieurs. . . . .	40
3.11	Application de la deuxième règle. . . . .	41
4.1	Classement des langages de programmation [15]. . . . .	47
4.2	Exemple de couplage fort [29]. . . . .	49

---

4.3	Exemple da couplage faible [29]. . . . .	49
4.4	Architecture du framework Spring MVC [29]. . . . .	54
4.5	Fonctionnement du framework Spring MVC [23]. . . . .	55
4.6	Logo de l'application. . . . .	58
4.7	Interface d'authentification. . . . .	59
4.8	Interface Article BL. . . . .	60
4.9	Interface Ajouter Article BL. . . . .	61
4.10	Interface Modifier Article BL. . . . .	62
4.11	Interface Archive. . . . .	62
4.12	Interface Détails Archive. . . . .	63
4.13	Statistique nombre de conteneurs visités. . . . .	64
4.14	Statistique nombre de TC's dépassant les 81jours. . . . .	64
4.15	Interface Utilisateurs. . . . .	65
4.16	Interface Modifier Profil Utilisateur. . . . .	66
4.17	Internationalisation de l'application. . . . .	66
4.18	Exécution des commandes Maven sous STS. . . . .	68
4.19	Gestionnaire d'application web Tomcat. . . . .	69

## LISTE DES TABLEAUX

2.1	Messages échangés entre le système et ses acteurs. . . . .	20
2.2	Identification des cas d'utilisation. . . . .	22
2.3	Description du cas d'utilisation « Authentification ». . . . .	24
2.4	Description du cas d'utilisation « Ajouter Article BL». . . . .	25
2.5	Description du cas d'utilisation « Modifier Article BL ». . . . .	26
2.6	Description du cas d'utilisation « Supprimer Article BL ». . . . .	27
2.7	Description du cas d'utilisation « Ajouter conteneur ». . . . .	27
2.8	Description du cas d'utilisation « Ajouter marchandise ». . . . .	28
3.1	Dictionnaire de données. . . . .	44

## LISTE DES ABRÉVIATIONS

- AOP** : Aspect Oriented Programming JavaEE
- API** : Application Programming Interface
- BMT** : Bejaia méditerranéen terminale
- BL** : Bill of Loding
- CSS** : Cascading Style Sheets
- DAO** : Data Access Objet
- DI** : Injection de Dépendance
- DZLE** : Direction de la zone logistique extra-portuaire
- EJB** : Enterprise Java Beans
- EPB** : Entreprise Portuaire de Bejaia
- HTML** : Hypertext Markup Language
- HTTP** : Hypertext Transfer Protocol
- IEEE** : Institute of Electrical and Electronics Engineers
- IoC** : Inversion of Contrôle
- JAR** : Java Archive
- JDBC** : Java Database Connectivity
- JSP** : JavaServer Pages
- MVC** : Modèle Vue Contrôleur
- MySQL** : My Structured Query Language
- ORM** : Object Relational Mapping
- Pdf** : Portable document format

**SE** : Système d'Exploitation

**SGBD** : Système de Gestion de Base de Données

**STS** : Spring tool suite **TC** : Tank container

**UI** : User Interface

**UML** : Unified Modeling Language

**UP** : Processus Unifié

**WAR** : Web application Archive

**WWW** : World wide web

**XML** : eXtensible Markup Language

**XP** : Extrême Programming

## INTRODUCTION GÉNÉRALE

La technologie est aujourd'hui un élément fondamental pour toute entreprise. L'évolution dans le domaine de l'informatique offre des opportunités pour améliorer les processus de flux des travaux, elle permet de faire circuler l'information dans toute l'entreprise grâce aux logiciels et devient donc un outil de gestion et de pilotage de l'entreprise.

L'arrivée de l'internet a révolutionné les entreprises, notamment avec l'apparition du web, travaillant à relier des données considérables, permettant une vaste communication, et vu les difficultés en matière de diffusion d'informations sur les structures d'entreprises ou autres structures, le www (world wide web) permet l'accès à ce qu'on appelle des sites web (statiques ou dynamique) destinés à promouvoir et faire connaître ces structures mais aussi à assurer leur gestion.

L'entreprise portuaire de Bejaia (EPB) fait partie intégrante des entreprises que l'informatique pourra beaucoup aider. En effet, à l'issue de nos observations lors de notre stage qui s'y est bien déroulé avant l'arrivée de la pandémie COVID-19, nous avons constaté que le département de la zone logistique de Tixter de la direction fonctionnelle de l'EPB connaît assez de difficultés liées à la gestion manuelle des documents entraînant le gaspillage du temps lors du traitement des opérations, ce qui peut entraîner des erreurs de saisie.

Pour prendre en compte ces difficultés, nous proposons de mettre en place une application de gestion et de suivi des conteneurs dès leur arrivée au port de Bejaia jusqu'à leurs transferts vers le port sec de Bordj Bou Arreridj Tixter, qui permettrait de pallier les insuffisances et d'améliorer le fonctionnement du département en question. L'objectif

de notre projet est de concevoir et réaliser un système d'information fiable et de qualité concrétisé sous forme d'une application web dynamique.

Pour implémenter notre proposition, nous avons utilisé le framework Spring mvc de java qui a permis la réalisation de notre application. Cette dernière offre à l'employé de l'EPB un moyen pour mieux gérer les conteneurs, leurs transferts vers Tixter, leurs suivis ainsi que la possibilité de consulter et mettre à jour les différentes informations. En outre, l'extrait des états tels que le bon de sortie pour le transfert est aussi possible via notre application.

Notre mémoire est structuré en 4 chapitres :

Le premier chapitre est consacré à la présentation de l'entreprise EPB, en particulier, le département de la zone logistique de Tixter de la direction fonctionnelle où nous avons effectué notre stage, suivie de la problématique ainsi que la solution proposée et la démarche de développement suivie lors de la conception de notre application.

Le second chapitre détermine les acteurs qui interagissent avec le système à développer, les besoins de l'entreprise d'accueil ainsi que les cas d'utilisation correspondant à chaque acteur.

Dans le troisième chapitre, l'analyse et la conception du système sera détaillée en élaborant les diagrammes de séquence système ainsi que le diagramme de classe.

Le quatrième et dernier chapitre sera consacré à la réalisation de notre application. Nous présenterons les différents outils et langages utilisés ainsi que l'architecture du framework Spring mvc , quelques interfaces et le déploiement de l'application.

Enfin, le mémoire s'achève par une conclusion générale qui résume l'essentiel de notre travail ainsi que quelques perspectives.

## CHAPITRE 1

# PRÉSENTATION DE L'ORGANISME D'ACCUEIL & MÉTHODOLOGIE DE DÉVELOPPEMENT

### **1.1 Introduction**

Dans ce chapitre, nous allons présenter l'organisme d'accueil qui est l'Entreprise Portuaire de Bejaia (EPB) son historique, ses missions et activités, puis on s'intéressera au département zone logistique extra-portuaire. Par la suite nous allons cerner la problématique et proposer une solution ainsi que des objectifs assignés dans le but d'assurer une meilleure gestion, enfin nous terminerons par le processus à suivre ainsi que le langage de modélisation adéquat.

### **1.2 Présentation de l'organisme d'accueil**

Afin d'avoir une idée générale de l'organisme d'accueil, nous décrirons brièvement son évolution à travers le temps, sa localisation géographique, ses missions ainsi que ses activités.

#### **1.2.1 Historique de l'EPB**

Présentant des sites de mouillage naturels, Bejaia a toujours attiré les navires qui y trouvaient dans la baie un refuge sûr, la réalisation du port dans la composante ac-



tuelle débuta en 1834, elle fut achevée en 1987. C'est en 1960 qu'a été chargé le premier pétrolier d'Algérie, et ce depuis le port de Bejaia. Le port de Bejaia aujourd'hui est réputé mixte ; hydrocarbures et marchandises générales y sont traités. L'aménagement moderne des superstructures, le développement des infrastructures, l'utilisation de moyens de maintenance et de techniques adaptés à l'évolution de la technologie des navires et enfin ses outils de gestion modernes, ont fait évoluer le port de Bejaia depuis le milieu des années 1990.

En 2014, un port sec a été construit dans la ville de Tixter, à l'est de la wilaya de Bordj Bou Arreridj permettant de transférer directement les cargaisons vers les haut-plateaux, Cette zone extra-portuaire affectée au port de Bejaia permettra de le désengorger. La zone aura une superficie de 20 hectares et sera reliée par chemin de fer au Port de Bejaia, via Bordj Bou Arreridj.

Le port sec a une capacité de 500 000 conteneurs par an et de 20 millions de tonnes de fret non-conteneurisés.

### **1.2.2 Position géographique de l'EPB**

Le port de Bejaia joue un rôle très important dans les transactions internationales vu sa place et sa position géographique. Il est délimité par :

- Au nord par la route nationale N°9.
- Au sud par les jetées de fermeture et du large sur une largeur de 2 750m.
- A l'est par la jetée Est.
- A l'ouest par la zone industrielle de Bejaia.

### **1.2.3 Missions de l'EPB**

L'entreprise Portuaire de Bejaia a pour missions :

- La gestion, l'exploitation et le développement du domaine portuaire sont les charges essentielles de la gestion de l'EPB, c'est dans le but de promouvoir les échanges extérieurs du pays. Elle se doit d'assumer la police et la sécurité au sein du pays.
- Elle est chargée des travaux d'entretien, d'aménagement, de renouvellement et de création d'infrastructures.

- L'EPB assure également des prestations à caractère commercial, à savoir ; le remorquage, la manutention et l'acconage.

### 1.2.4 Activités de l'EPB

Les principales activités de l'entreprise sont :

- L'exploitation de l'outillage et des installations portuaires.
- L'exécution des travaux d'entretien, d'aménagement et de renouvellement de la super structure portuaire.
- L'exercice du monopole des opérations d'acconage et de manutention portuaire.
- L'exercice du monopole des opérations de remorquage, de pilotage et d'amarrage.
- La police et la sécurité portuaire dans la limite géographique du domaine public portuaire.

## 1.3 Structures de l'EPB

L'EPB est organisée selon des directions opérationnelles et fonctionnelles :

### — Directions opérationnelles

Il s'agit des structures qui prennent en charge les activités sur le terrain et qui ont une relation directe avec les clients.

### — Directions Fonctionnelles

Il s'agit des structures de soutien aux structures opérationnelles.

La Figure 1.1 représente l'organigramme récapitulant l'ensemble des directions de l'EPB :

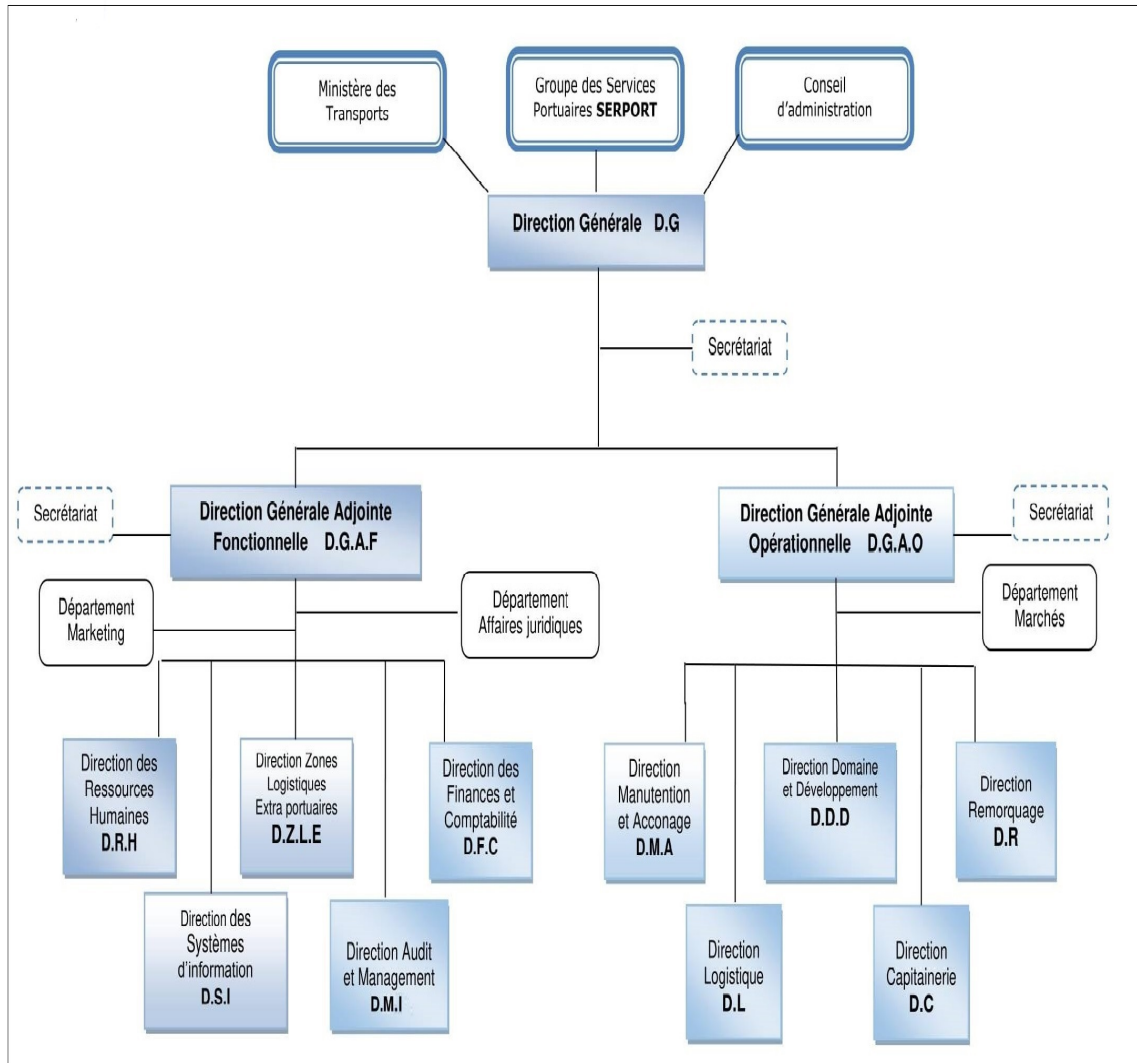


FIGURE 1.1 – Organigramme général de l'Entreprise Portuaire de Bejaia [source EPB].

### 1.3.1 La direction de la zone logistique extra-portuaire (DZLE)

La DZLE se constitue d'une cellule et de deux départements :

- La cellule « Chargé des opérations au port » : se charge du suivi des transferts des conteneurs (*TC's*) vers Tixter ainsi que son suivi administratif.
- Département de la Zone Logistique de Tixter.
- Département de la Zone Logistique d'Ighil Ouberouak..

La Figure 1.2 représente l'organigramme de la direction de la zone logistique extra-portuaire.

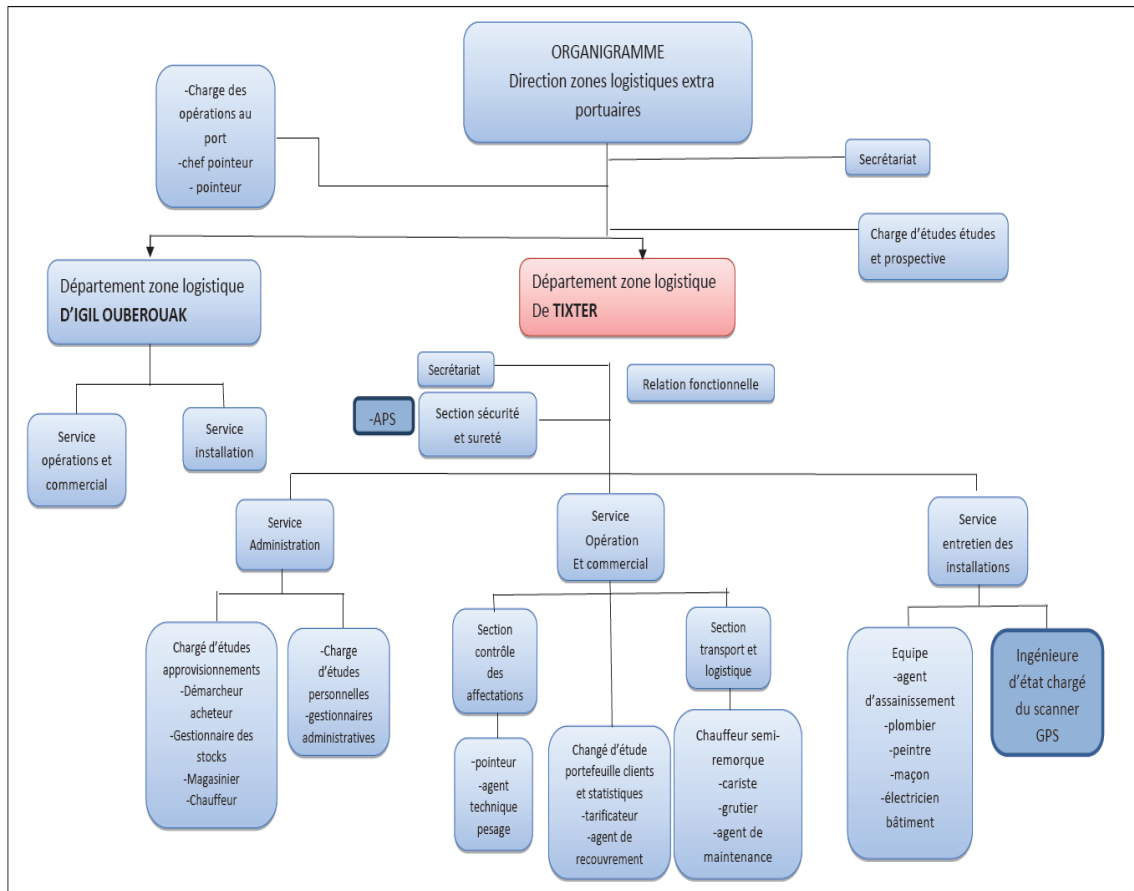


FIGURE 1.2 – Organigramme de la DZLE [source EPB].

### 1.3.2 Le département de la zone logistique extra-portuaire de TIXTER

Le département de la Zone Logistique de TIXTER comporte 03 services, à savoir :

- Le service installations.
- Le service opérations et commercial.
- Le service administration et finances.

## 1.4 Vocabulaire spécifique à l'EPB

Afin de mieux comprendre les sections qui suivent, il est nécessaire de comprendre quelques notions relatives à la gestion des conteneurs, parmi eux nous citons [source EPB] :

**Consignataire :**

Le Consignataire de Navire, également appelé consignataire maritime, agent de consignation ou agent maritime, a pour mission de prendre en charge les navires dans un port et d'organiser leur escale, ainsi que le transport des marchandises.

**Transitaire :**

Un transitaire est une personne ou une entreprise mandatée par l'expéditeur ou le destinataire d'une marchandise qui doit subir plusieurs transports successifs. Sa mission est d'organiser la liaison entre les différents transporteurs et d'assurer ainsi la continuité du transport. (C'est le représentant du client dans le territoire portuaire).

**Réceptionnaire :**

Le réceptionnaire tient un rôle important dans le circuit des produits. Il réceptionne la marchandise finale et contrôle l'état des produits livrés.

**Manifeste :**

État récapitulatif des connaissances relatifs aux marchandises chargées à bord du navire. Il mentionne les éléments d'identification des marchandises tels que les marques et numéros de colis, leur nature, leur poids. (Chaque navire présente un seul manifeste).

**Article BL :**

Le connaissement maritime est le "ticket de transport" de votre marchandise, depuis son origine jusqu'à sa destination. Il est délivré par un transporteur à un expéditeur. Il détaille la méthode et le trajet d'une expédition. Il fonctionne comme un contrat pour le mouvement des marchandises avec les termes souvent décrits au verso du B/L. Il sert également de reçu pour la cargaison et peut éventuellement servir de preuve de la propriété des marchandises.

## 1.5 Étude de l'existant

Un navire dit « Porte-conteneurs » arrivant au port de Bejaia transporte des conteneurs appartenant à plusieurs clients, ceci est mentionné en détail dans un document nommé le « Manifeste ». Ce dernier comporte à son tour un ensemble de Bill of lading (BL), connaissement matérialisant le contrat de transport maritime conclu entre le chargeur et le transporteur. Il est également un titre représentatif des marchandises transportées dans le conteneur.

Le travail de la direction zone logistique extra-portuaire s'effectue en deux parties :

- Suivis du transfert des conteneurs vers TIXTER (au niveau de la cellule chargée des opérations au port de Bejaia).
- Suivis des opérations (au niveau du service opération et commercial du département zone logistique de TIXTER).

#### **A. Suivis du transfert des conteneurs vers TIXTER**

La cellule chargée des opérations au port de Bejaia se charge d'effectuer les opérations suivantes :

1. Saisir l'état des demandes d'éclatement du manifeste sur excel est la première tâche effectuée par le chargé d'étude de la direction et ceci avant l'arrivée du navire, cela sert à préparer le terrain. Ce document est rempli à partir des BL des clients (le manifeste n'étant pas encore arrivé).
2. La cellule chargée des opérations au port transmet le fichier excel établi précédemment vers l'entreprise Bejaia méditerranéen terminale (BMT) qui se charge d'effectuer le tri et de déposer les conteneurs destinés à TIXTER dans le poste n°18.
3. Réception du manifeste et compléter les informations manquantes dans l'état des demandes d'éclatement du manifeste tel que le numéro de gros et de l'article BL.
4. Une fois le tri effectué, un bon de sortie va être établi par un agent appelé pointeur pour expédier les conteneurs.
5. A partir des bons de sorties un document appelé « programme de transfert des conteneurs » va être établi qui contient une partie des données saisies dans l'état des demandes d'éclatement du manifeste et d'autres informations supplémentaires tel que la nature de la marchandise.

#### **B. Suivis des opérations**

Les opérations sur les conteneurs s'effectuent au niveau de la zone logistique de TIXTER par le service « opération et commercial », et pour chacune d'elles un « Bon de commande » est exigé. Parmi ces opérations nous citons :

##### **Opération de dédouanement :**

Inclut les opérations suivantes :

- Levage pour visite.
- Levage pour visite partielle.
- Levage pour visite intégrale.
- Levage pour scanner.
- Le scanning.

**Opération de levage pour empotage/dépotage :**

Consiste en deux opérations :

- Empotage : charger les conteneurs dans les camions.
- Dépotage : décharger la marchandise du conteneur et la charger par la suite dans le camion du client.

**Opération de restitution du vide :**

La compagnie maritime livre pour la cellule chargée des opérations au port une liste des conteneurs à restituer selon la stratégie FIFO (first in first out) qui veut dire le premier conteneur arrivé sera le premier restitué.

Il existe deux sortes de restitution du vide :

1. Restitution : Le client rend le conteneur après l'avoir déchargé vers TIXTER.
2. Rapprochement : Ramener le vide de TIXTER vers l'EPB.

Le tarificateur garde trace de toutes les opérations effectuées sur chaque conteneur dans un document appelé « Fiche de suivi des opérations » en mentionnant la date pour la facturation.

## **1.6 Problématique**

Pour déterminer l'état actuel de la gestion des conteneurs de l'entreprise portuaire de Bejaia et de proposer des améliorations à la direction de la zone logistique extra-portuaire (DZLE), nous avons mené une étude de l'existant. A partir des entretiens avec le personnel de la direction, des anomalies nous ont été signalées d'une part et détectées d'autre part. Le problème majeur qui se pose est l'établissement manuel de tous les documents servant à la gestion et au transfert des conteneurs dès leur arrivée au port jusqu'à

leurs livraisons au client, d'où le risque d'erreurs. En outre, d'autres éléments viennent se greffer à cette situation telle que :

- L'absence d'un système informatique pour la gestion des conteneurs et leurs transports ainsi qu'une base de données.
- Toutes les informations sont saisies et stockées dans des fichiers Excel et Word.
- Le cumul de documents ainsi que le risque de leurs pertes.
- Une charge de travail pour les agents du DZLE.
- Perte de temps et redondance des données lors de leurs saisies par les agents.
- L'absence de statistiques servant à effectuer des rapports mensuels et annuels.
- Problème dans la collecte des données.

## **1.7 Objectifs et solutions**

Le but de notre projet est la réalisation d'une application web pour la gestion et le suivi des conteneurs, dès leur arrivée au port de Bejaia jusqu'à leurs transferts vers le port sec de Bordj Bou Arreridj TIXTER, qui va permettre de pallier les insuffisances citées ci-dessus et d'accomplir certains objectifs qui seront utiles à l'entreprise. Parmi ceux-ci, on trouve :

- Alléger la charge du travail qui se fait actuellement de façon manuelle et d'intégrer certaines fonctionnalités manquantes.
- Réduire le taux d'utilisation du papier et cela en automatisant toutes les fonctions nécessaires à la gestion des conteneurs.
- Réduire le temps de traitement des données.
- Sécuriser la sauvegarde des données.
- Faciliter la recherche et l'accès aux informations à travers une application informatique conviviale et facile à manipuler.

## **1.8 Démarche de développement**

Afin de réaliser notre application, nous avons choisi le Processus Unifié (UP) combiné avec la méthode agile Extrême Programming (XP) qui définit un certain nombre de bonnes pratiques permettant de développer un logiciel dans des conditions optimales en



plaçant le client au cœur du processus de développement. En effet, la méthode du processus unifié s'appuie principalement sur la modélisation et les diagrammes UML d'où notre choix pour ces derniers.

### **1.8.1 Processus unifié (Unified Process)**

Le processus unifié est un processus de développement logiciel construit sur UML. Il est itératif et incrémental, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. Il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise. Les caractéristiques essentielles de ce processus sont [1] :

#### **- Processus guidé par les cas d'utilisation**

Le système à construire se définit d'abord avec les utilisateurs. Les cas d'utilisation permettent d'exprimer les interactions du système avec ces derniers, donc de capturer les besoins [2].

#### **- Processus centré sur l'architecture**

Dès le début des travaux d'analyse et de conception, les auteurs d'UP mettent en avant la préoccupation de l'architecture du système. Il est important de définir le plus tôt possible, l'architecture type qui sera retenue pour le développement, l'implémentation et ensuite le déploiement du système. Le vecteur des cas d'utilisation peut aussi être utilisé pour la description de l'architecture [2].

#### **- Processus orienté par la réduction des risques**

Les risques sont présents à tous les stades de développement d'un système. Il est important de bien évaluer les risques des développements afin d'aider à la bonne prise de décision. Le fait que le processus UP est itératif, cela contribue à la diminution des risques au fur et à mesure du déroulement des itérations successives [2].

### - Processus itératif et incrémental

Ce type de démarche étant relativement connu dans l'approche objet, il paraît naturel qu'UP préconise l'utilisation du principe de développement par itérations successives. Le projet est découpé en itérations ou étapes de courte durée qui permettent de mieux suivre l'avancement globale. A la fin de chaque itération une partie exécutable du système finale est produite, de façon incrémentale [2].

La Figure 1.3 qui suit illustre le déroulement du processus Up.

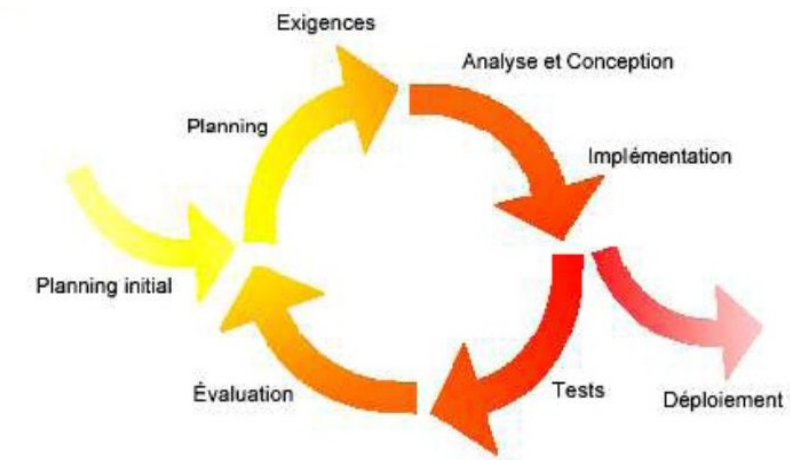


FIGURE 1.3 – Déroulement du processus Up [2].

### 1.8.2 XP (eXtreme Programming)

La méthodologie eXtreme Programming ou XP est une méthode de développement logiciel qui applique à l'extrême les principes du développement agile, c'est-à-dire se concentrer sur les besoins du clients, mettre en place un développement itératif et l'intégration continue de ces derniers. L'équipe projet et ses relations avec le client sont au cœur de XP. La raison de la création de cette méthode est, autrement dit, réduire les coûts qu'engendre les changements dans les projets.

L'Extreme Programming préconise également le travail en binôme des développeurs et avec une petite équipe, facilitant ainsi la production d'un code simple, facilement lisible et maintenable.

La méthode XP repose sur quatre valeurs [3] :

#### **La communication :**

- XP favorise le contact humain, la communication directe, plutôt que les échanges de

courriers électroniques ou de documents formels.

- Les développeurs travaillent directement avec la maîtrise d'ouvrage afin d'éviter des situations de blocage et la découverte tardive de problèmes.

**Le feedback (retour d'information) :**

- Obtenir des retours aussi rapides que réguliers de la part du client, pour s'améliorer et améliorer le produit.

- Le feed-back est une deuxième valeur essentielle dans XP, c'est grâce au feed-back que le processus industriel du développement peut évoluer.

- Par les tests unitaires, le développeur a un retour immédiat sur son système et plus précisément sur les modifications qu'il vient d'apporter.

**Le courage :**

- Il faut du courage pour se borner à réaliser des choses simples, se focaliser uniquement sur les besoins du moment. Il en faut aussi pour accepter de jeter une partie du code qui est devenue inutile, ou réécrire une partie de code jugée trop complexe [4].

**Le respect :**

- Les relations entre l'équipe de développement et le client doivent être basés sur le respect mutuel [5].

### **1.8.3 UML (Unified Modeling Language)**

Le Langage de Modélisation Unifié, se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. UML s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel [6].

## 1.9 Conclusion

Dans ce chapitre nous avons présenté l'entreprise portuaire de Bejaia (EPB), puis on s'est focalisé sur le département zone logistique extra-portuaire au sein duquel on a effectué notre stage. Suite à l'étude de l'existant on a pu cerner la problématique et proposer une solution ainsi que des objectifs à atteindre dans le but d'assurer une meilleure gestion, enfin nous avons terminé par le processus à suivre ainsi que le langage de modélisation adéquat. Pour mieux assimiler le contexte du système il est essentiel de passer par la spécification des besoins qui sera détaillée dans le prochain chapitre.

## 2.1 Introduction

Le développement de n'importe quelle application informatique nécessite de connaître les besoins techniques, ergonomiques et fonctionnels du client. C'est pourquoi nous avons consacré ce chapitre à la spécification des besoins fonctionnels et non fonctionnels de notre application. Nous identifierons par la suite les acteurs qui interagissent avec le système à travers un diagramme de contexte et nous poursuivrons avec l'identification des messages échangés ainsi que les cas d'utilisation que nous avons jugés les plus importants. Pour finir ce chapitre, nous allons modéliser le tout à travers un diagramme de cas d'utilisation global.

## 2.2 Besoins de l'entreprise d'accueil

Avant de passer à la phase pratique et de réalisation de notre application, la bonne pratique nécessite de définir un cahier de charges détaillé sur le projet. Dans cette section, on se limitera à la spécification des besoins de l'entreprise.

### 2.2.1 Besoins fonctionnels

Les besoins fonctionnels représentent les fonctionnalités que le système doit exécuter, il ne devient opérationnel que s'il les satisfait [7]. Les besoins fonctionnels principaux

sont résumés ci-dessous :

- Automatiser le processus de gestion des conteneurs.
- L'authentification : Une fois accédé à l'application, l'utilisateur doit introduire son login et mot de passe pour s'authentifier et avoir ensuite accès à d'autres fonctionnalités.
- Générer des statistiques et rapports de fin d'années pour faciliter la tâche aux employés. Entre autres, ces fonctionnalités permettent d'avoir une vue globale sur le processus de gestion des conteneurs.
- Gérer les Articles BL's des clients, les manifestes et les sauvegarder dans la base de données.
- Gérer les transferts (expédition et rapprochement) et les opérations effectuées sur les conteneurs afin de garder traces des dates de ces derniers.
- Gérer les tables de base : transitaires, consignataires, conteneurs, size, types d'opération, type de marchandise, chauffeurs, moyens de transport, utilisateurs, navires, opération et réceptionnaires.
- Mettre en place un mécanisme permettant d'exporter des documents sous format PDF ou Excel.
- Consulter et rechercher à tout moment les informations afin de mettre à jour les données.

### 2.2.2 Besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou de type de conception [8]. La liste suivante représente les besoins non-fonctionnels de notre application :

- Contraintes d'implémentation (langage de programmation, type de SGBD, SE).
- Performance : l'application doit assurer un temps de réponse minime, tout en répondant aux exigences de l'utilisateur.
- Sécurité : les données recueillies par l'application ne doivent être accessibles que par le personnel autorisé.
- Modularité du code : le code doit être lisible afin de simplifier la maintenance.
- Ergonomie : l'interface doit être simple à utiliser, les informations doivent être

lisibles et ce avec un minimum d'effort.

## 2.3 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [7]. En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. D'autre part, le même acteur peut jouer alternativement plusieurs rôles et le même rôle peut être joué par plusieurs acteurs [8]. Dans notre cas, nous avons identifié trois acteurs :

**Employé de l'EPB** : cet acteur pourra s'authentifier et effectuer les tâches suivantes :

- Gestion des BL's.
- Gestion des manifestes.
- Gestion des tables de bases (conteneurs, consignataire, transitaire, réceptionnaire, navires, types des opérations, chauffeurs et moyens de transport).
- Gestion des expéditions (transfert des conteneurs de l'EPB vers Tixter).
- Consulter les statistiques.
- Gérer son compte utilisateur.

**Employé de Tixter** : cet acteur pourra s'authentifier et aura accès aux fonctionnalités suivantes :

- Gestion des rapprochements (transfert des conteneurs vides de Tixter vers l'EPB).
- Consulter les statistiques.
- Gestion des opérations effectuées sur les conteneurs tel que le scanner, la pesé, la visite...
- Gérer son compte utilisateur.

**Administrateur** : se charge de la gestion des comptes des utilisateurs (activer / désactiver les comptes et réinitialiser les mots de passe) et de la création des comptes pour les employés.

## 2.4 Diagramme de contexte dynamique

Le diagramme de contexte dynamique permet de positionner le système dans son environnement selon le point de vue des communications. Il précise les échanges d'informations qui sont réalisées entre le système et les éléments matériels extérieurs au système. Le système est donc décrit physiquement et logiquement [9].

La Figure 2.1 ci-dessous représente le diagramme de contexte qui permet de présenter et de visualiser tous les messages échangés entre les acteurs du système identifiés auparavant.

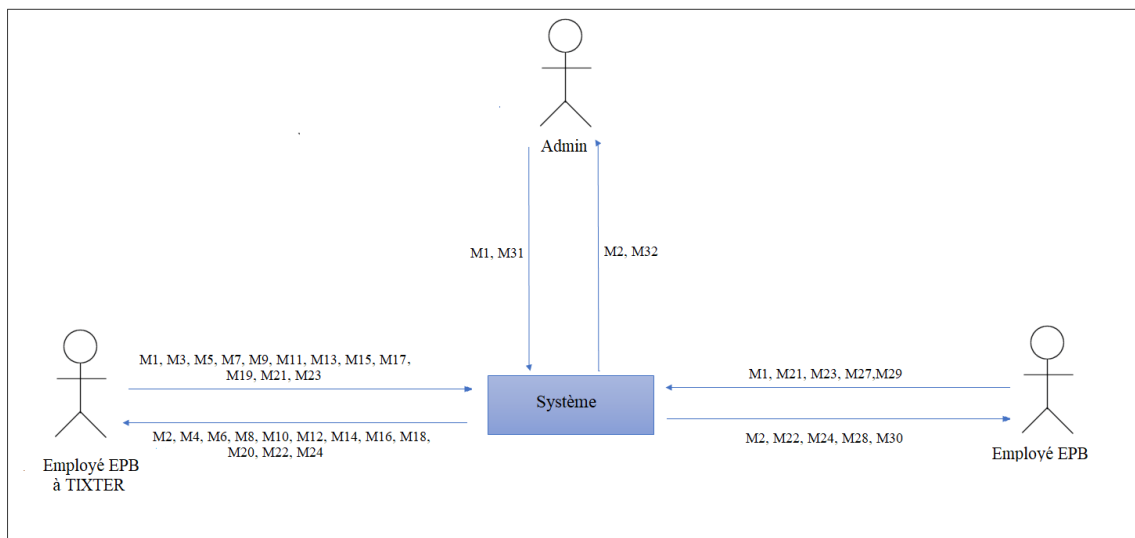


FIGURE 2.1 – Diagramme de contexte dynamique.

### 2.4.1 Identification des messages échangés

Un message est une information envoyée par un acteur et provoquant le déclenchement de certaines actions associées à ce dernier, en terme général c'est un moyen de communication entre les différents acteurs du système [7].

Le Tableau 2.1 ci-dessous représente les différents messages échangés entre les acteurs et le système :



## CHAPITRE 2. SPÉCIFICATION DES BESOINS

Numéro	Message entrant	Numéro	Message sortant
M1	Demande d'authentification	M2	Affichage de l'interface d'accueil
M3	Gestion Article BL	M4	Affichage de l'interface de gestion des Articles BL's
M5	Gestion manifeste	M6	Affichage de l'interface de gestion des manifestes.
M7	Gestion marchandise	M8	Affichage de l'interface de gestion de marchandise
M9	Gestion expédition (vers Tixter)	M10	Affichage de l'interface de gestion des expéditions
M11	Gestion consignataire	M12	Affichage de l'interface de gestion des consignataires
M13	Gestion navire	M14	Affichage de l'interface de gestion des navires
M15	Gestion transitaire	M16	Affichage de l'interface de gestion des transitaires
M17	Gestion conteneur	M18	Affichage de l'interface de gestion des conteneurs
M19	Gestion type marchandise	M20	Affichage de l'interface de gestion des types de marchandise
M21	Gestion chauffeur	M22	Affichage de l'interface de gestion des chauffeurs
M23	Gestion moyen transport	M24	Affichage de l'interface de gestion des moyens de transport
M25	Gestion rapprochement (vers EPB)	M26	Affichage de l'interface de gestion des rapprochements
M27	Gestion opération	M28	Affichage de l'interface de gestion des opérations
M29	Gestion types opération	M30	Affichage de l'interface de gestion de types des opérations
M31	Gestion comptes	M32	Affichage de l'interface de gestion des comptes
M33	Consulter statistiques	M34	Affichage de l'interface

TABLE 2.1 – Messages échangés entre le système et ses acteurs.

## 2.5 Identification des cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

Il modélise un service rendu par le système, exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné [7].

Lors de notre étude, nous avons pu identifier les actions importantes pour chaque acteur qui sont illustrées dans le Tableau 2.2 ci-dessous :

Numéro	Cas d'utilisation	Acteur
1	Gestion Article BL	Ajouter
		Supprimer
		Modifier
		Rechercher
		Exporter(xl, pdf)
2	Gestion manifeste	Ajouter
		Supprimer
		Modifier
		Rechercher
		Exporter(xl, pdf)
3	Gestion marchandise	Ajouter
		Supprimer
		Modifier
		Rechercher
		Exporter(xl, pdf)
4	Gestion expédition (vers Tix-ter)	Ajouter
		Supprimer
		Modifier
		Rechercher
		Exporter(xl, pdf)
5	Gestion des tables de bases	Ajouter
		Supprimer
		Modifier

Employé EPB

TABLE 2.2 – suite du tableau précédent

		Rechercher	
		Exporter(xl, pdf)	
6	Consulter statistiques		Employé Tixter & EPB
7	Gestion opération	Ajouter	Employé Tixter
		Supprimer	
		Modifier	
		Rechercher	
		Exporter(xl, pdf)	
8	Gestion types opération	Ajouter	
		Supprimer	
		Modifier	
		Rechercher	
		Exporter(xl, pdf)	
9	Gestion rapprochement(vers EPB)	Ajouter	
		Supprimer	
		Modifier	
		Rechercher	
		Exporter(xl, pdf)	
10	Gestion comptes	Ajouter	Administrateur
		Valider	
		Modifier	

TABLE 2.2 – Identification des cas d'utilisation.

### 2.5.1 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation servent à décrire un ensemble d'actions (cas d'utilisation) que certains systèmes devraient ou peuvent exécuter en collaboration avec un ou plusieurs utilisateurs externes du système (acteurs). Chaque cas d'utilisation doit fournir des résultats observables et utiles aux acteurs ou autres parties prenantes du système [10]. La Figure 2.2 représente le diagramme de cas d'utilisation global qui décrit l'ensemble des cas d'utilisation associés aux différents acteurs du système :

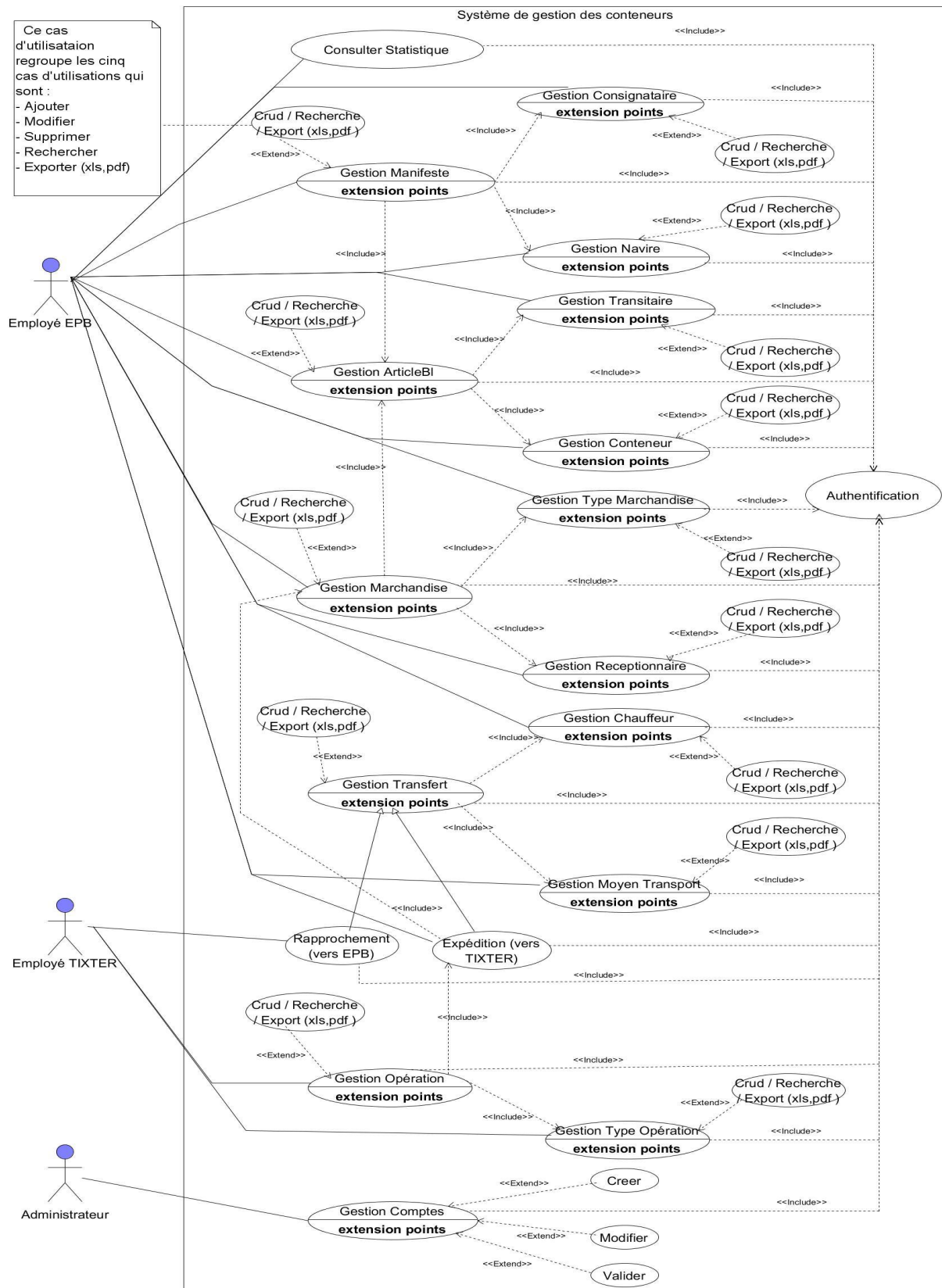


FIGURE 2.2 – Diagramme de cas d'utilisation globale du système.

## 2.5.2 Description textuelle des cas d'utilisation

La description d'un cas d'utilisation permet de clarifier le déroulement de la fonctionnalité. Nous allons décrire quelques cas d'utilisation à travers les tableaux qui suivent.

### Le cas d'utilisation «Authentification»

Sommaire d'identification »	
<b>Titre</b>	Authentification.
<b>Résumé</b>	Ce cas d'utilisation permet à l'utilisateur d'accéder à son compte, et au système de gérer les droits d'accès et d'affichage.
<b>Acteur</b>	Tous les acteurs.
Description des scénarios	
<b>Pré-conditions</b>	Démarrage de l'application.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur introduit son login et mot de passe.</li> <li>2. Le système vérifie l'existence du compte.</li> <li>3. Le système affiche l'interface principale selon le statut de l'utilisateur.</li> </ol>
<b>Enchaînements d'erreurs</b>	Le système affiche un message d'erreur si le compte n'existe pas ou le mot de passe/login est incorrect et réaffiche le formulaire d'authentification.
<b>Post-conditions</b>	L'utilisateur accède à la partie du système qui lui est réservées.

TABLE 2.3 – Description du cas d'utilisation « Authentification ».

---

**Le cas d'utilisation « Ajouter Article BL »**

<b>Sommaire d'identification »</b>	
<b>Titre</b>	Ajouter Article BL.
<b>Résumé</b>	Ce cas d'utilisation permet d'ajouter un nouveau Article BL.
<b>Acteur</b>	L'employé de l'EPB.
<b>Description des scénarios</b>	
<b>Pré-conditions</b>	L'employé authentifié.  Le transitaire et les conteneurs existent déjà dans la base de données.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système vérifie le type d'utilisateur connecté.</li> <li>2. L'employé demande d'accéder à l'interface d'ajout d'un Article BL BL.</li> <li>3. Le système affiche l'interface correspondante.</li> <li>4. L'employé saisi les informations dans le formulaire d'ajout et cherche les conteneurs.</li> </ol>
<b>Enchaînements d'erreurs</b>	Le système afficher un message d'erreur si le transitaire, les conteneur et les champs obligatoires n'ont pas été sélectionnées dans le formulaire d'ajout.
<b>Post-conditions</b>	Le système enregistre l'Article BL dans la base de données.

TABLE 2.4 – Description du cas d'utilisation « Ajouter Article BL ».

---

**Le cas d'utilisation « Modifier Article BL »**

<b>Sommaire d'identification »</b>	
<b>Titre</b>	Modifier Article BL.

TABLE 2.5 – suite du tableau précédent

<b>Résumé</b>	Ce cas d'utilisation permet de modifier un Article BL.
<b>Acteur</b>	L'employé de l'EPB.
<b>Description des scénarios</b>	
<b>Pré-conditions</b>	L'employé doit être authentifié. Le transitaire et les conteneurs existent déjà dans la base de données..
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système vérifie le type d'utilisateur connecté.</li> <li>2. L'employé clique sur le bouton modifier de l'Article BL BL correspondant dans l'interface d'affichage des Articles BL's.</li> <li>3. Le système affiche le formulaire de modification contenant les informations de l'Article BL.</li> <li>4. L'employé saisi les informations dans le formulaire d'ajout et cherche les conteneurs.</li> <li>5. L'employé modifie les informations et les enregistre.</li> </ol>
<b>Enchaînements d'erreurs</b>	Le système afficher un message d'erreur si le transitaire, conteneur et les champs obligatoires n'ont pas été sélectionnés.
<b>Post-conditions</b>	Le système met à jour la base de données.

TABLE 2.5 – Description du cas d'utilisation « Modifier Article BL ».

**Le cas d'utilisation « Supprimer Article BL »**

<b>Sommaire d'identification »</b>	
<b>Titre</b>	Supprimer Article BL.
<b>Résumé</b>	Ce cas d'utilisation permet de supprimer un Article BL.
<b>Acteur</b>	L'employé de l'EPB.
<b>Description des scénarios</b>	

<b>Pré-conditions</b>	L'employé doit être authentifié. .
<b>Scénario nominal</b>	1. Le système vérifie le type d'utilisateur connecté. 2. L'employé clique sur le bouton supprimer de l'Article BL correspondant. 3. Le système affiche la boîte de confirmation.
	4. L'employé confirme la suppression.
<b>Enchaînements d'erreurs</b>	Aucune.
<b>Post-conditions</b>	Le système met à jour la base de données.

TABLE 2.6 – Description du cas d'utilisation « Supprimer Article BL ».

**Le cas d'utilisation « Ajouter conteneur »**

<b>Sommaire d'identification »</b>	
<b>Titre</b>	Ajouter conteneur.
<b>Résumé</b>	Ce cas d'utilisation permet d'ajouter un nouveau conteneur.
<b>Acteur</b>	L'employé de l'EPB.
<b>Description des scénarios</b>	
<b>Pré-conditions</b>	L'employé doit être authentifié. .
<b>Scénario nominal</b>	1. Le système vérifie le type d'utilisateur connecté. 2. L'employé saisie les informations du conteneur. 3. Le système enregistre le nouveau conteneur.
<b>Enchaînements d'erreurs</b>	Le système affiche un message d'erreur si le type, la taille n'ont pas été renseigné.
<b>Post-conditions</b>	Le système enregistre le conteneur.

TABLE 2.7 – Description du cas d'utilisation « Ajouter conteneur ».



**Le cas d'utilisation « Ajouter marchandise »**

<b>Sommaire d'identification »</b>	
<b>Titre</b>	Ajouter marchandise.
<b>Résumé</b>	Ce cas d'utilisation permet d'ajouter les marchandises aux conteneurs..
<b>Acteur</b>	L'employé de l'EPB.
<b>Description des scénarios</b>	
<b>Pré-conditions</b>	L'employé authentifié.  Le réceptionnaire et le type de marchandise existent déjà dans la base de données.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système vérifie le type d'utilisateur connecté.</li> <li>2. L'employé demande d'accéder à l'interface d'ajout d'une marchandise.</li> <li>3. Le système affiche l'interface correspondante.</li> <li>4. L'employé saisie les informations de chaque marchandise et clique sur le bouton 'Ajouter' pour l'insérer dans la base de données.</li> <li>5. Le clique sur 'Valider la marchandise' permet de confirmer les choix.</li> </ol>
<b>Enchaînements d'erreurs</b>	Afficher un message d'erreur si les champs d'information obligatoires n'ont pas été renseignés.
<b>Post-conditions</b>	Le système enregistre la marchandise dans la base de données.

TABLE 2.8 – Description du cas d'utilisation « Ajouter marchandise ».

## 2.6 Conclusion

Ce chapitre nous a permis de livrer une spécification complète des besoins de notre application en montrant le principe de fonctionnement et les différentes interactions entre les acteurs et le système, aussi nous avons présenté le diagramme de contexte ainsi que le diagramme de cas d'utilisation ce qui nous permet d'appréhender plus efficacement le chapitre suivant qui est l'analyse et conception.

### 3.1 Introduction

Après l'étape de spécification des besoins, nous entamons maintenant les phases d'analyse et de conception. Afin d'analyser la dynamique des besoins capturés nous allons commencer par illustrer les différents diagrammes de séquences système représentant les interactions chronologiques entre les acteurs et le système, par la suite nous détaillerons la conception de notre application au moyen du diagramme de classes et du modèle relationnel et nous terminerons par le dictionnaire de données.

### 3.2 Diagramme de séquence système

Le diagramme de séquence est un diagramme d'interaction UML. Il représente des échanges de messages entre éléments au cours d'une interaction, Il comprend un groupe d'objets, représentés par des lignes de vie. L'intérêt d'un diagramme de séquence système est de décrire textuellement de façon détaillée chaque cas d'utilisation [7]. Afin de mieux comprendre le fonctionnement de notre système nous allons dans ce qui suit détailler les cas d'utilisation donnés dans le chapitre précédent par des diagrammes de séquences système.

### 3.2.1 Diagramme de séquence système « Authentification »

À chaque authentification, l'utilisateur doit saisir son login et mot de passe, d'où l'usage de l'opérateur « loop ». L'utilisateur valide sa demande de connexion, deux cas peuvent se présenter : si les champs sont vides, le système affiche un message d'erreur 'Veillez renseigner tous les champs', si les champs sont bien remplis, le système vérifie l'existence de ce compte, deux autres cas peuvent se présenter : si le compte n'existe pas affichage d'un message d'erreur 'Ce compte n'existe pas' sinon affichage de l'interface correspondante au statut de l'utilisateur.

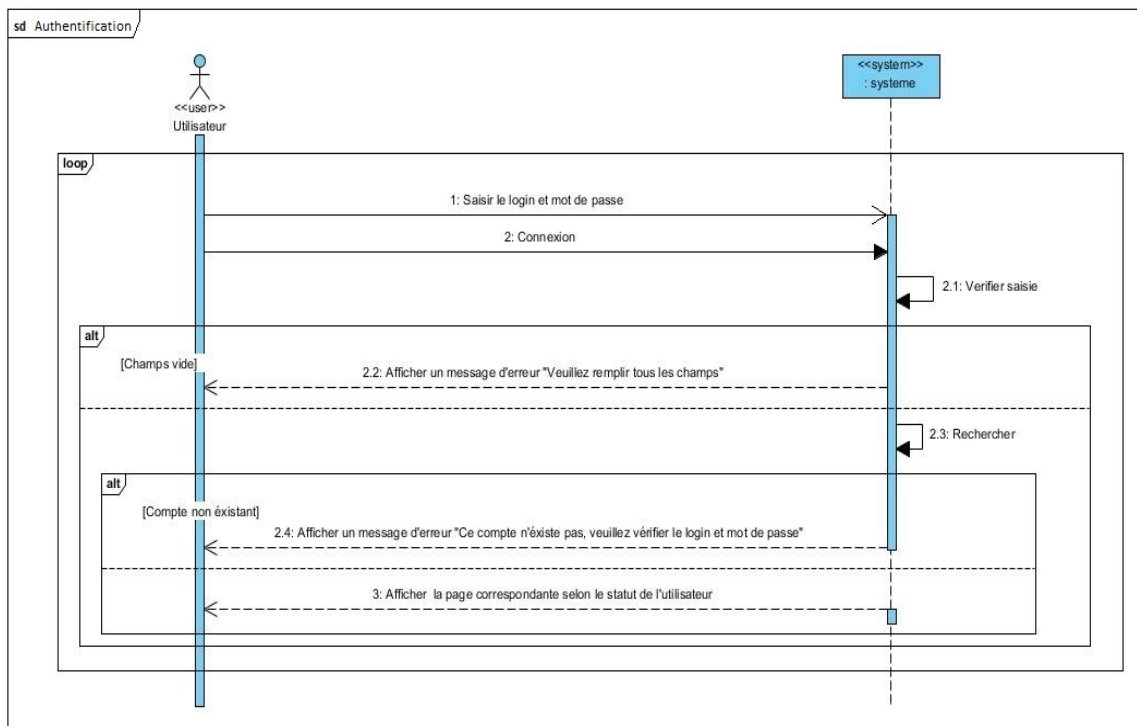


FIGURE 3.1 – Diagramme de séquence système de cas d'utilisation « Authentification ».

### 3.2.2 Diagramme de séquence système de cas d'utilisation « Ajouter Conteneur »

Après s'être authentifié, l'employé de l'EPB accède à l'interface d'ajout du conteneur, remplit les champs du formulaire et valide. Deux cas peuvent se présenter : si les champs obligatoires du formulaire ne sont pas renseignés, le système affiche le message d'erreur « Veuillez renseigner les champs obligatoires » sinon le système vérifie l'existence du conteneur, si ce dernier existe déjà dans la base de données, affichage du message

d'erreur « Ce conteneur existe déjà » sinon le conteneur est ajouté à la base de données et l'utilisateur est redirigé vers l'interface d'affichage de la liste des conteneurs.

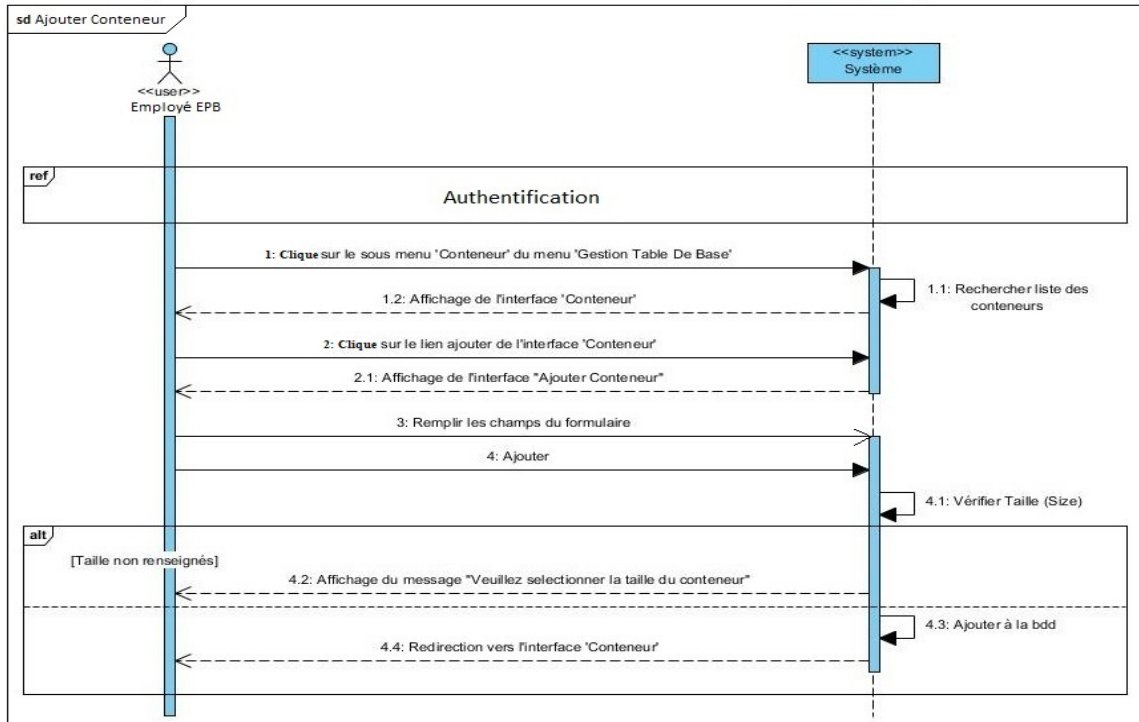


FIGURE 3.2 – Diagramme de séquence système de cas d'utilisation « Ajouter Conteneur ».

### 3.2.3 Diagramme de séquence système de cas d'utilisation « Ajouter Marchandise »

Après s'être authentifié l'employé de l'EPB accède à l'interface d'affichage des conteneurs dont la marchandise n'a pas encore été renseignée, sélectionne un conteneur et clique sur le bouton 'ajouter marchandise', affichage du formulaire d'ajout de la marchandise (plusieurs marchandises peuvent être ajoutées au conteneur d'où l'utilisation de la boucle 'Loop'), l'utilisateur renseigne les champs et valide. Deux cas peuvent se présenter : si les champs obligatoires n'ont pas été renseignés, affichage du message d'erreur 'Veuillez renseigner les champs obligatoires' sinon la marchandise est ajoutée au conteneur et sauvegardée dans la base de données. Deux options peuvent se présenter après l'ajout d'une marchandise (d'où l'utilisation de 'opt') la suppression et la modification de cette dernière. Une fois toute la marchandise renseignée et vérifiée, l'utilisateur clique sur le bouton 'Valider' pour éliminer le conteneur de la liste des marchandises non

renseignées.

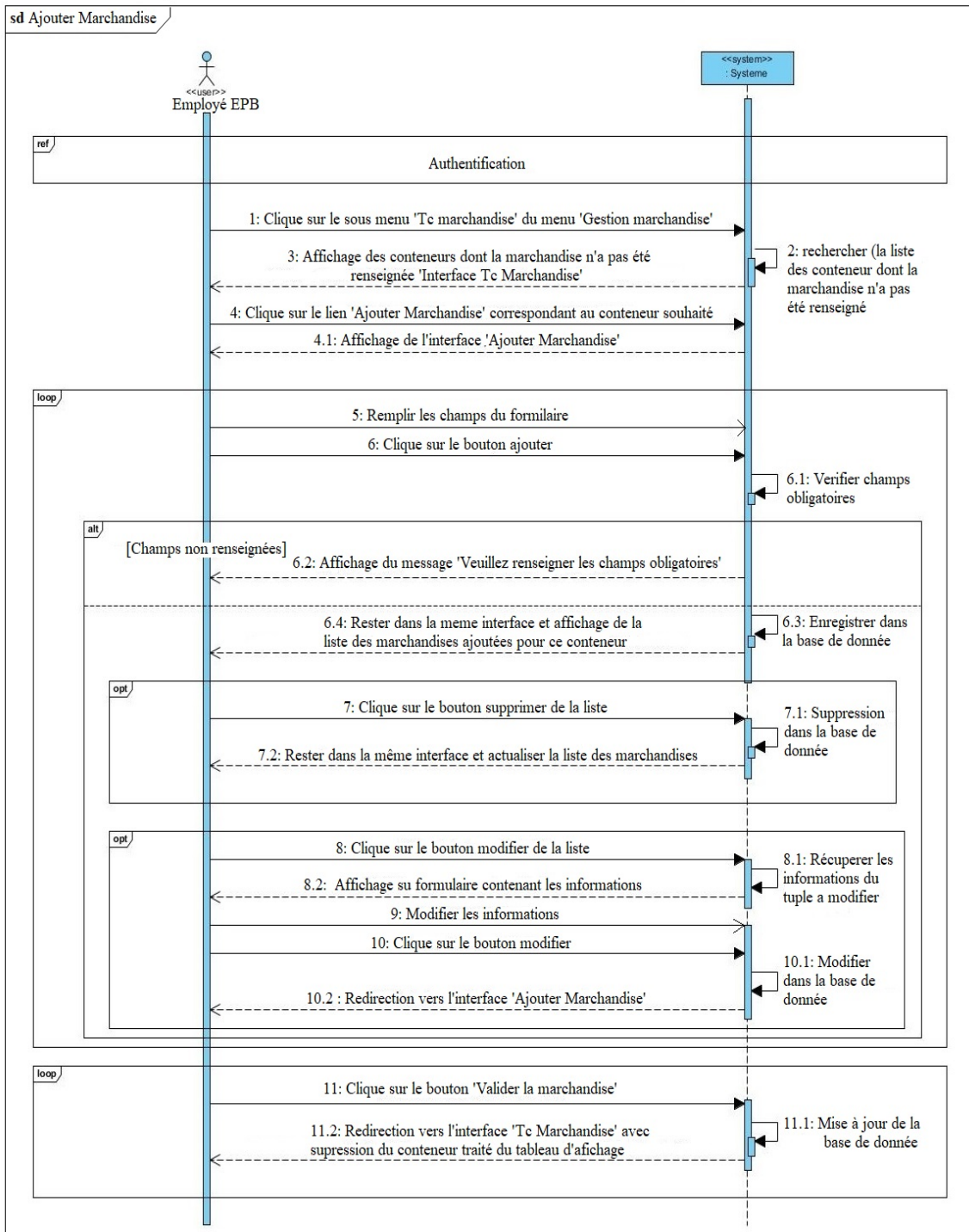


FIGURE 3.3 – Diagramme de séquence système de cas d'utilisation « Ajouter Marchandise ».

### 3.2.4 Diagramme de séquence système de cas d'utilisation « Ajouter Article BL »

Après s'être authentifié, l'employé de l'EPB accède à l'interface d'ajout d'un Article BL, remplit les champs du formulaire, sélectionne un ou plusieurs conteneurs (d'où l'utilisation de 'loop'). Deux cas peuvent se présenter : si le champs 'Transitaire' du formulaire n'a pas été renseigné, le message d'erreur 'Veuillez sélectionner un transitaire' sera affiché, sinon le système recherche le conteneur sélectionné : si ce dernier existe déjà dans la liste, affichage du message d'erreur 'Ce conteneur a déjà été sélectionné ' sinon le système ajoute le conteneur à la liste. Enfin, l'utilisateur clique sur le bouton 'Enregistrer' et le système sauvegarde l'Article BL dans la base de données.

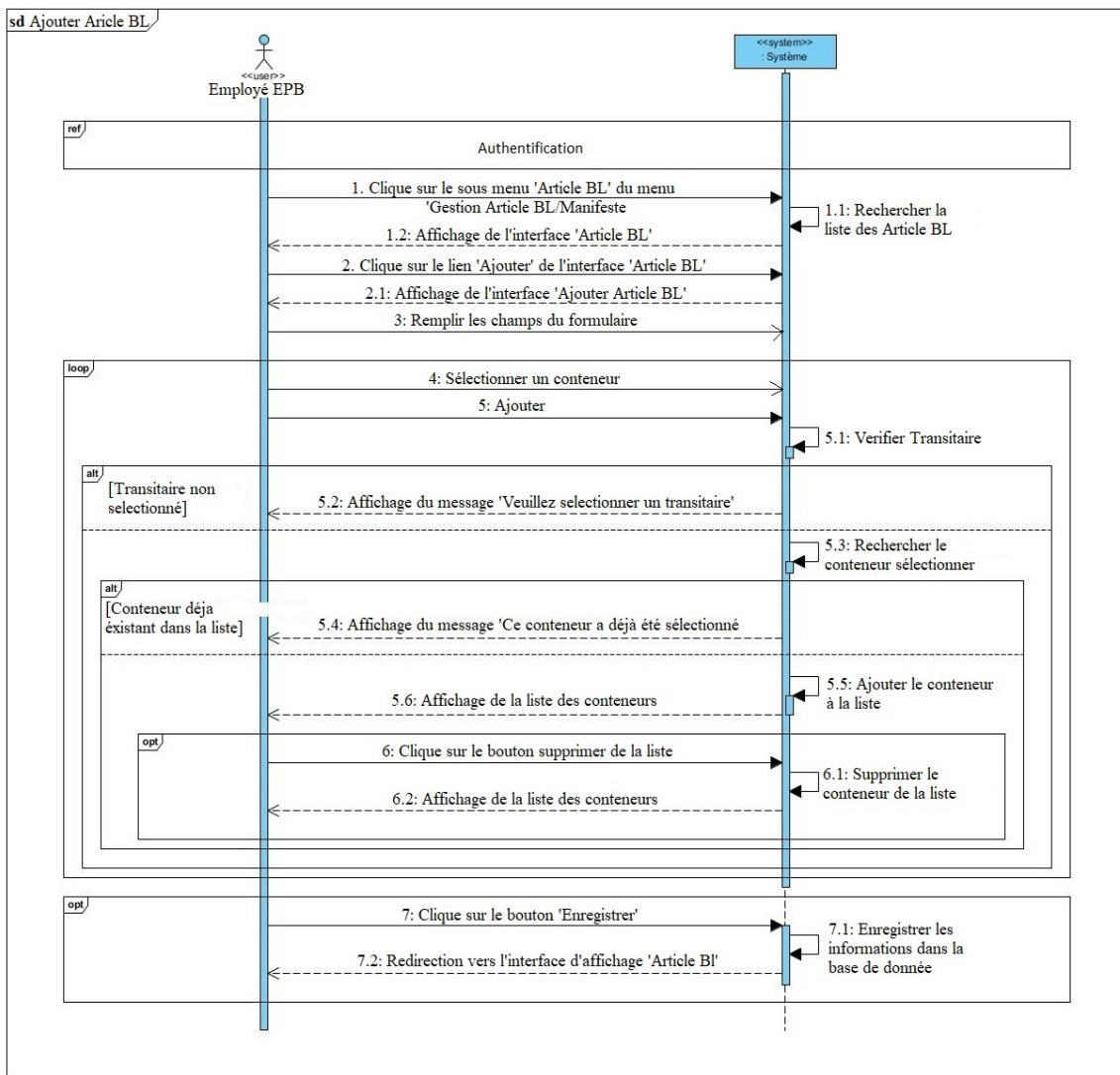


FIGURE 3.4 – Diagramme de séquence système de cas d'utilisation « Ajouter Article BL ».

### 3.2.5 Diagramme de séquence système de cas d'utilisation « Modifier Article BL »

Après s'être authentifié, l'employé de l'EPB clique sur le bouton 'Modifier' correspondant à l'Article BL qu'il souhaite modifier. Le système affiche le formulaire de modification contenant les informations de cet Article BL. L'employé modifie les champs et valide. La base de données est alors mise à jour et l'utilisateur est redirigé vers l'interface d'affichage des Articles BL's.

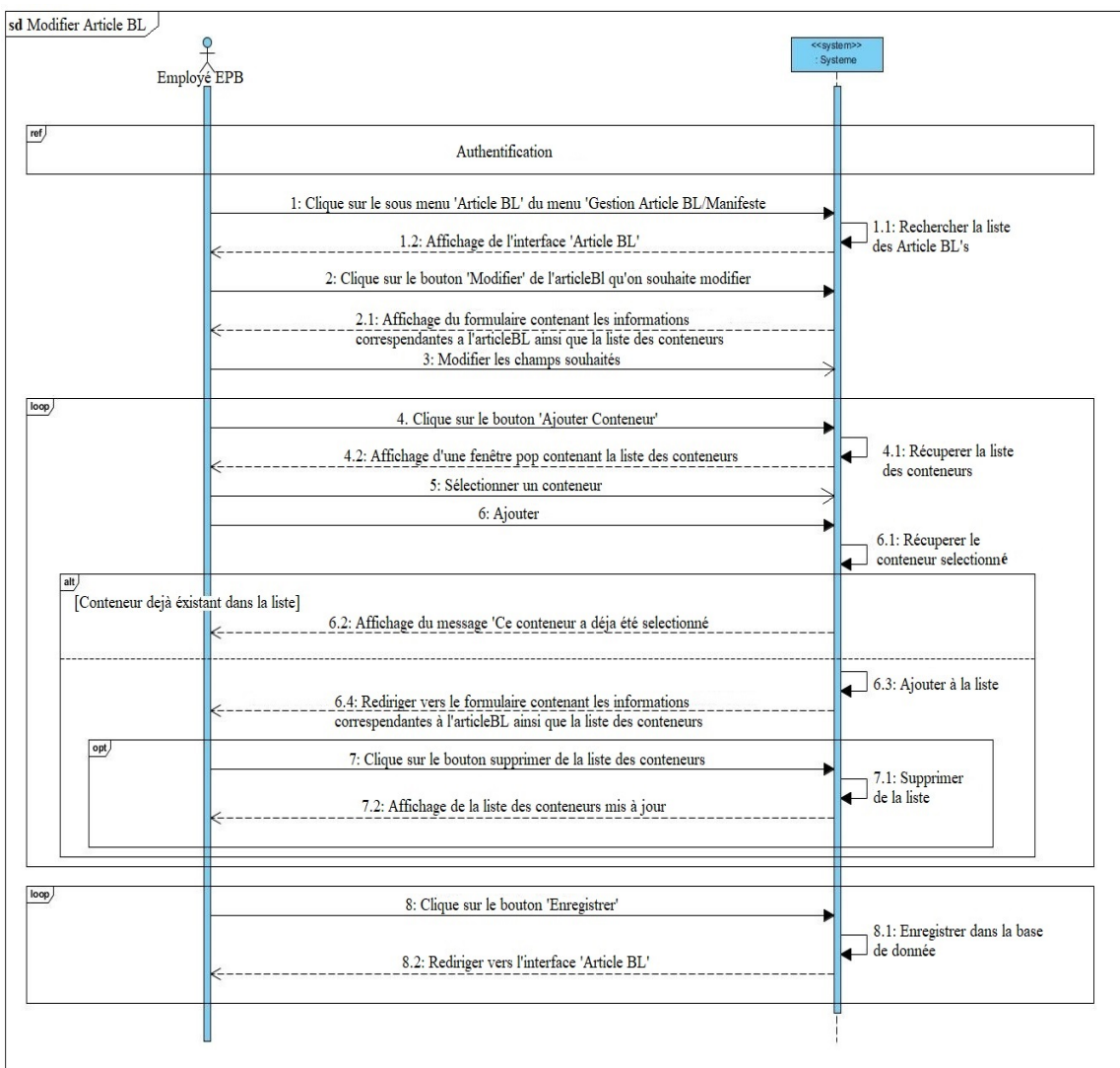


FIGURE 3.5 – Diagramme de séquence système de cas d'utilisation « Modifier Article BL ».



### 3.2.6 Diagramme de séquence système de cas d'utilisation « Supprimer Article BL »

Après s'être authentifié, l'employé de l'EPB clique sur le bouton 'Supprimer' correspondant à l'Article BL qu'il souhaite supprimer. Une boîte de dialogue va être affichée, l'utilisateur peut confirmer ou annuler la suppression de l'Article BL et il sera redirigé vers l'interface d'affichage des Articles BL's.

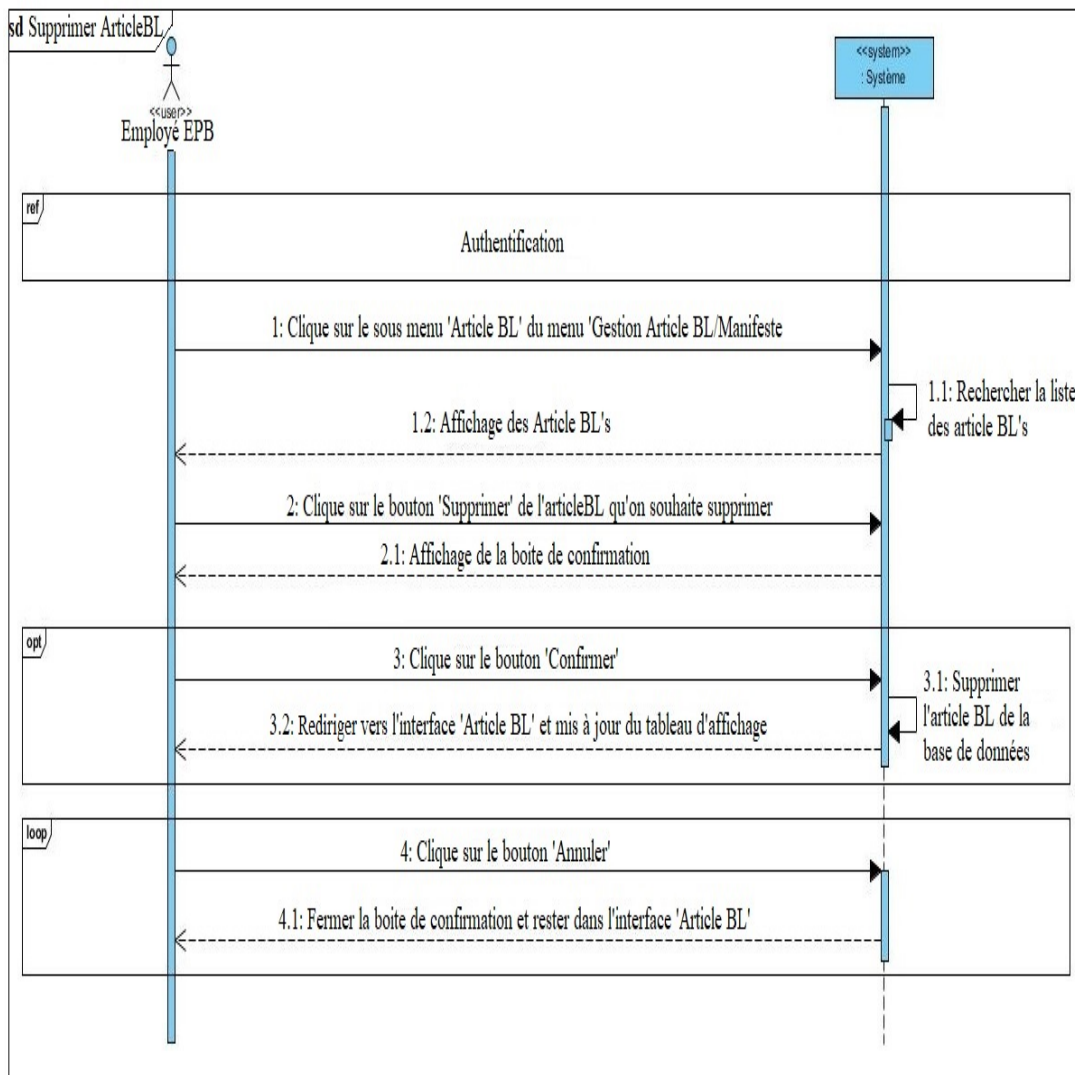


FIGURE 3.6 – Diagramme de séquence système de cas d'utilisation « Supprimer Article BL ».

### 3.3 Diagramme de classes

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objet de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet [7].

Ce type de diagramme représente les classes intervenantes dans le système, chaque classe se décrit par des données dites attributs et de traitements dont elle est responsable vis-à-vis d'elle-même et dans d'autres classes, ainsi que les associations entre elles. La Figure 3.7 suivante montre le diagramme de classe final validé :

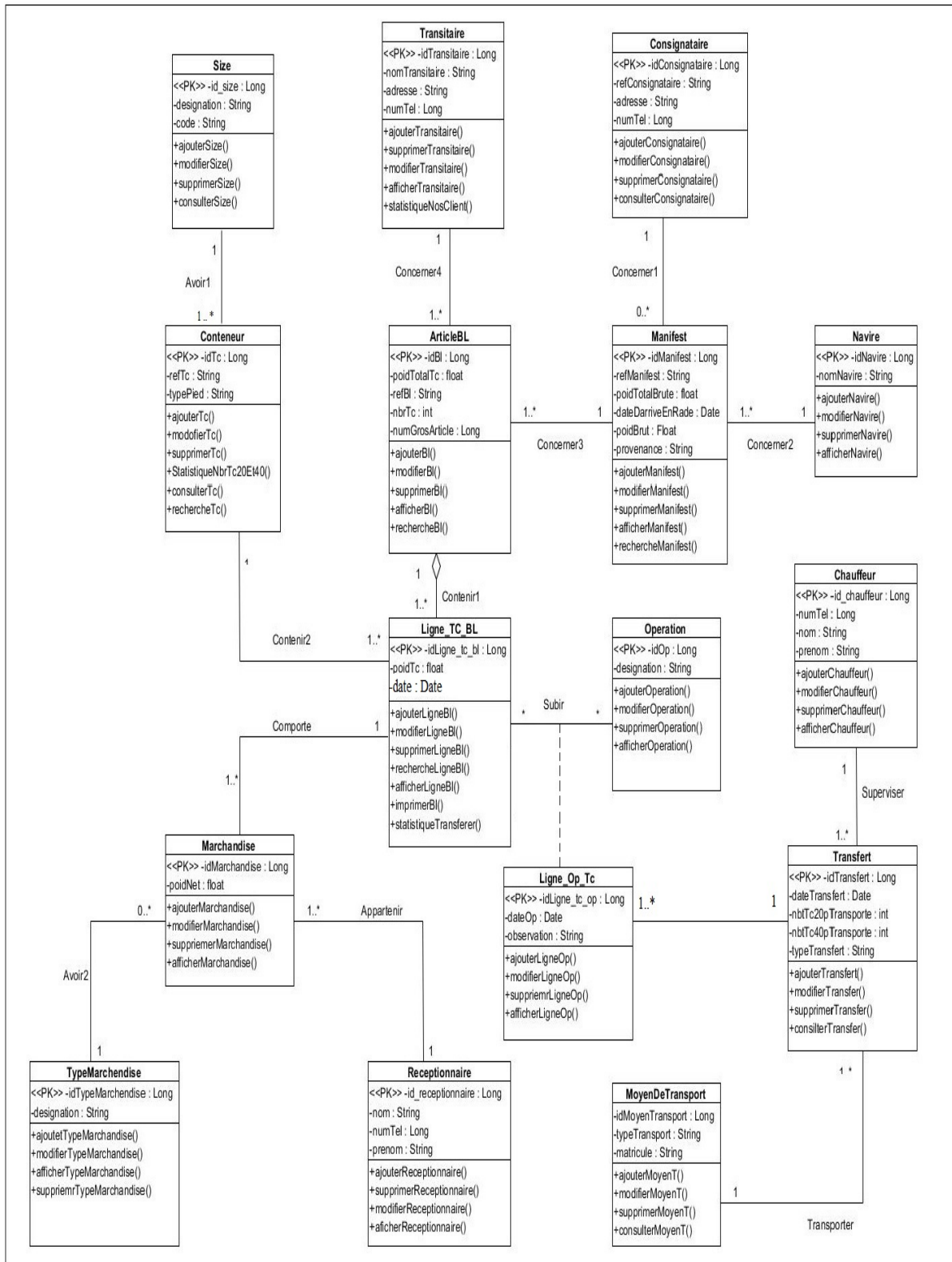


FIGURE 3.7 – Diagramme de classes.

### 3.4 Modèle relationnel

Le modèle relationnel est basé sur une organisation des données sous forme de tables. La manipulation des données se fait selon le concept mathématique de relation de la théorie des ensembles, c'est-à-dire l'algèbre relationnelle. Elle est constituée d'un ensemble d'opérations formelles sur les relations. Les opérations relationnelles permettent de créer une nouvelle relation (table) à partir d'opérations élémentaires sur d'autres tables (par exemple l'union, l'intersection, ou encore la différence) [12].

Afin de transformer le diagramme de classe en modèle relationnel, il est nécessaire d'appliquer les règles suivantes :

#### 1. Première règle :

Présence de la cardinalité (\*..1) d'un côté de l'association

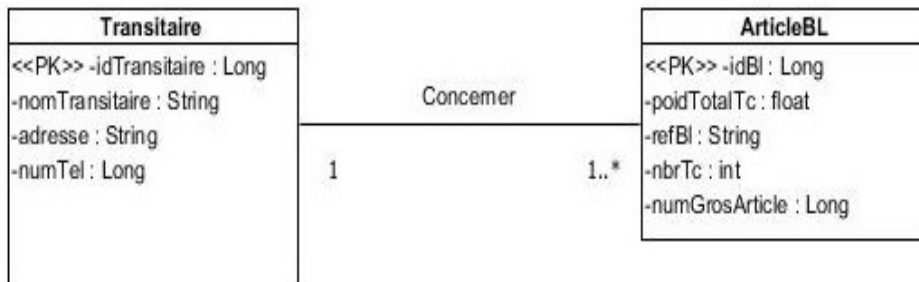


FIGURE 3.8 – Relation un à plusieurs.

- Chaque classe se transforme en une table.
- Chaque attribut de classe se transforme en un champ de table.
- L'identifiant de la classe qui est associé à la cardinalité (1) (ex : Transitaire) devient la clé étrangère de l'autre classe (ex : Article BL).

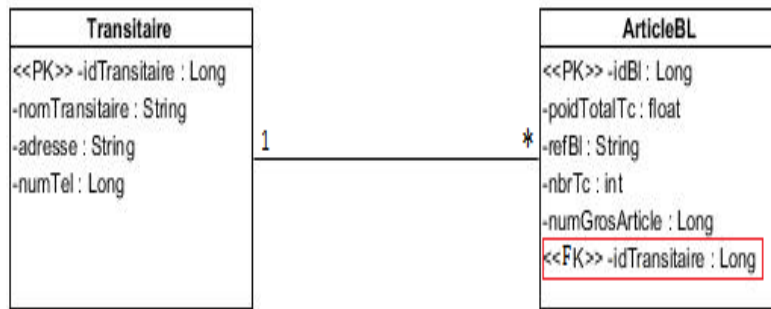


FIGURE 3.9 – Application de la première règle.

## 2. Deuxième règle :

Présence de la cardinalité (\*..\*) des deux côtés de l'association

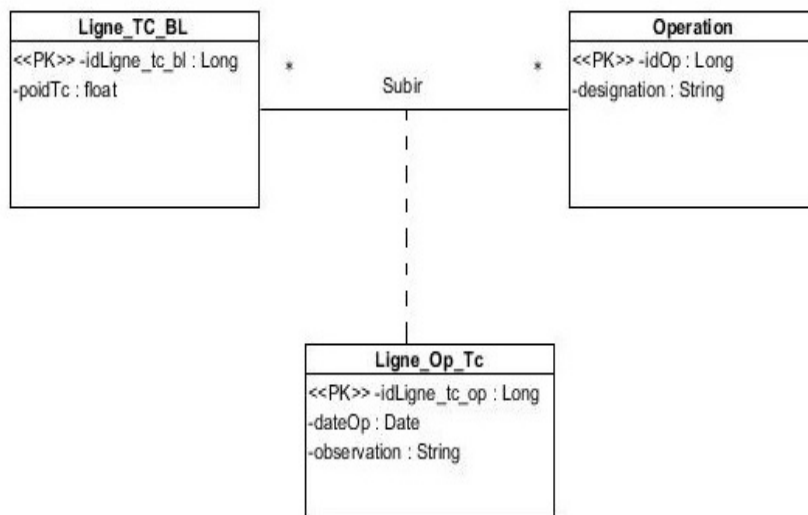


FIGURE 3.10 – Relation plusieurs à plusieurs.

- Chaque classe se transforme en une table..
- Chaque attribut de classe se transforme en un champ de table.
- L'association se transforme en une table. Cette table a comme champs l'identifiant de chacune des deux classes, plus d'éventuels autres attributs..

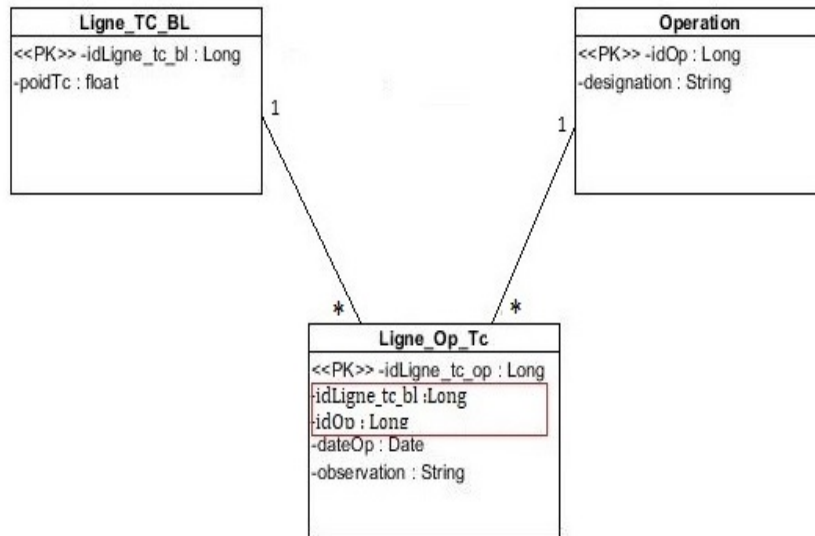


FIGURE 3.11 – Application de la deuxième règle.

### 3. Troisième règle :

Présence d'une généralisation

#### Méthode 1 (push up) :

- Créer une table avec tous les attributs des classes.
- Ajouter un attribut pour distinguer les types des objets.

#### Méthode 2 (push down) :

- Créer une table pour chaque sous type, chaque table se compose des attributs génériques et d'attributs spécifiques.

#### Méthode 3 :

- Créer une table par classe et des associations.

Dans notre cas, seules les deux premières règles ont été utilisées, car le diagramme de classes est dépourvu de généralités.

En suivant les règles de passage du diagramme de classes au modèle relationnel cités précédemment, on obtient le modèle relationnel suivant :

**Size** (idSize, code, designation)

**Transitaire** (idTransitaire, nomTransitaire, numTel, adresse)

**Receptionnaire** (idReceptionnaire, nom, prénom, numTel, adresse)

**Consignataire** (idConsignataire, refConsignataire, numTel, adresse)

**Navire** (**idNavire**, nomNavire)

**Chauffeur** (**idChauffeur**, nom, prénom, numTel, adresse)

**Conteneur** (**idTc**, #idSize , refTc, typePied)

**Article BL** (**idBl**, #idTransitaire , #idManifest , refBl, nbrTc, numGrosArticle)

**Manifest** (**idManifest**, #idNavire , #idConsignataire, refManifest, poidTotalBrut, dateDarriveEnRade, provenance)

**Ligne\_TC\_BL** (**idLigne-tc-bl**, #idTc , #idBl , poidTc)

**Marchandise** (**idMarchandise**, #idLigne-tc-bl , #idReceptionnaire , #idTypeMarchandise , poidNet)

**TypeMarchandise** (**idMarchandise**, designation)

**Operation** (**idOp**, designation)

**Ligne\_TC\_OP** (**idLigne\_tc\_op**, #idLigne-tc-bl , #idOp , #idTransfert , dateOp, observation)

**Transfert** (**idTransfert**, #idMoyenTransport, #idChauffeur , dateTransfert, nbtTc20, nbtTc40,typeTran

**MoyenDeTransport** (**idMoyenTransport**, matricule, typeTransport)

**Utilisateur** (**idUtilisateur**,#idRole, mail, motDePasse, nom, prenom, photo)

**role** (**idRole**, roleName)

### 3.5 Dictionnaire de données

Pendant la phase de conception, les données recueillies et spécifiées sont inscrites dans un dictionnaire de données qui est un document qui permet de recenser, de classer et de trier toutes les informations (les données) collectées lors des entretiens ou de l'étude des documents. Il constitue la référence de toutes les études effectuées par la suite [13].

Le Tableau 3.1 suivant illustre le dictionnaire de données de notre application :

Table	Attribut	Description de l'attribut	Type
<b>conteneur</b>	idTc	Identificateur	int(11)
	refTc	Référence du conteneur	varchar
	typePied	Type du conteneur	varchar
	idSize	Taille du conteneur	int(11)

TABLE 3.1 – suite du tableau précédent

<b>size</b>	idSize	Identificateur	int(11)
	code	Code	varchar
	designation	Désignation	varchar
<b>articlebl</b>	idBl	Identificateur	int(11)
	datePrevisionnel	Date prévisionnelle	varchar
	nbTc	Nombre de conteneurs	int
	poidTolatTc	Poid total des conteneurs	decimal
	refBl	Référence de l'articleBL	varchar
	idManifest	Clé étrangère	int
	idTransitaire	Clé étrangère	int
<b>transitaire</b>	idTransitaire	Identificateur	int(11)
	nom	Nom	varchar
	prenom	Prénom	varchar
	adresse	Adresse	varchar
	numTel	Numéro de téléphone	int(11)
<b>ligne_tc_bl</b>	idLigne_tc_bl	Identificateur	int(11)
	poidTc	Poids du conteneur	decimal
	etatMarchandise	Etat de la marchandise	Booléen
	etatRestitution	Etat de restitution	Booléen
	etatTransfert	Etat du transfert	Booléen
	idBl	Clé étrangère	int(11)
	idTc	Clé étrangère	int(11)
<b>manifeste</b>	idManifest	Identificateur	int(11)
	refManifest	Référence du manifeste	varchar
	dateDarriveEnRade	Date d'arrivé en rade	varchar
	poidBrut	Poids brut	decimal
	provenance	Provenance	varchar
	idConsignataire	Clé étrangère	int(11)
	idNavire	Clé étrangère	int(11)
<b>consignataire</b>	idConsignataire	Identificateur	int(11)
	refConsignataire	Référence du consignataire	varchar
	adresse	Adresse	varchar
	numTel	Numéro de téléphone	int(11)
<b>navire</b>	idNavire	Identificateur	int(11)
	nomNavire	Nom du navire	varchar
<b>marchandise</b>	idMarchandise	Identificateur	int(11)
	poidMarchandise	Poids de la marchandise	decimal
	idLigne_tc_bl	Clé étrangère	int(11)
	idReceptionnaire	Clé étrangère	int(11)
	idTypeMarchandise	Clé étrangère	int(11)
<b>typemarchandise</b>	idTypeMarchandise	Identificateur	int(11)



TABLE 3.1 – suite du tableau précédent

	designation	Designation	varchar
<b>receptionnaire</b>	idReceptionnaire	Identificateur	int(11)
	nom	Nom	varchar
	prenom	Prénom	varchar
	adresse	Adresse	varchar
	numTel	Numéro de téléphone	int(11)
<b>transfert</b>	idTransfert	Identificateur	int(11)
	dateTransfert	Date du transfert	varchar
	nbrTc20	Nombre de conteneur 20 pieds	int(11)
	nbrTc40	Nombre de conteneur 40 pieds	int(11)
	typeTransfert	Type de transfert	varchar
	idChauffeur	Clé étrangère	int(11)
	idMoyenTransport	Clé étrangère	int(11)
<b>chauffeur</b>	idChauffeur	Identificateur	int(11)
	nom	Nom	varchar
	prenom	Prénom	varchar
	adresse	Adresse	varchar
	numTel	Numéro de téléphone	int(11)
<b>moyentransport</b>	idMoyenTransport	Identificateur	int(11)
	typeTransport	Type transport	varchar
	matricule	Matricule	varchar
<b>operation</b>	idOp	Identificateur	int(11)
	designation	Désignation	varchar
<b>ligne_tc_op</b>	idLigne_tc_op	Identificateur	int(11)
	dateOperation	Date de l'opération	varchar
	observation	Observation	varchar
	idLigne_tc_bl	Clé étrangère	int(11)
	idOp	Clé étrangère	int(11)
	idTransfert	Clé étrangère	int(11)
<b>utilisateur</b>	idUtilisateur	Identificateur	int(11)
	email	Email	varchar
	motDePasse	Mot De Passe	varchar
	nom	Nom	varchar
	prenom	Prénom	varchar
	actived	Etat du compte	boolean
<b>role</b>	idRole	Identificateur	int(11)
	RoleName	Nom du role	varchar

TABLE 3.1 – Dictionnaire de données.

## 3.6 Conclusion

Ce chapitre a été consacré à la conception de l'application web pour la gestion de conteneurs. En premier lieu nous avons détaillé l'aspect dynamique du système via des diagrammes de séquences système puis l'aspect statique via un diagramme de classe qui nous a permis d'aboutir au modèle relationnel en suivant les règles de passage d'un diagramme de classe au modèle relationnel afin d'extraire les tables à implémenter dans la base de données. En fin de ce chapitre, nous avons jugé très utile de dresser le dictionnaire de données qui nous facilitera par la suite la création des tables, la codification des attributs avec les détails nécessaires de ces derniers.

Désormais, la phase de conception est terminée, nous pouvons s'étaler alors sur l'activité finale de notre projet qui est la phase de réalisation.

## 4.1 Introduction

Après avoir abordé la partie analyse et conception appropriée à notre projet, nous allons entamer la dernière phase de notre projet qui est la réalisation de notre application. Dans ce chapitre, nous présenterons l'environnement et les outils de développement dont le framework Spring. Nous passerons par la suite à la présentation des différentes interfaces de notre application ainsi que les étapes que nous avons suivi pour son déploiement.

## 4.2 Application web

Une application web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Une application web manipule des fichiers qui sont écrits en langage HTML (les pages web). Ces pages sont stockées sur un serveur et sont exécutées par le navigateur web du client.

Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide de son navigateur favori. La tendance actuelle est d'offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur les ordinateurs. Les technologies utilisées pour développer les applications web sont les mêmes que celles

employées dans la création des sites internet [14].

### 4.3 Classement des langages de programmation et Frameworks

L'institute of Electrical and Electronics Engineers (IEEE), la plus grande association mondiale de professionnels techniques, s'est érigée en baromètre des langages de programmation en livrant au deuxième semestre de chaque année son classement des meilleurs langages selon divers critères : popularité générale, langages en forte croissance, les meilleurs langages pour le développement de sites et applications web. Notons que les données collectées pour le classement de l'IEEE proviennent de huit sources différentes dont : Google Search, GitHub et Stack Overflow...

La sélection initiale comprenait près de 300 langages de programmation, mais le classement final n'en a retenu que 52.

La Figure 4.1 suivante donne le top 10 des langages les plus populaires pour le développement de sites et applications web en 2019 [16] :

Rank	Language	Type	Score
1	Python	🌐 🗨️ ⚙️	100.0
2	Java	🌐 📱 🗨️	96.3
3	JavaScript	🌐	79.4
4	C#	🌐 📱 🗨️ ⚙️	74.5
5	Go	🌐 🗨️	68.0
6	HTML,CSS	🌐	66.8
7	PHP	🌐	65.1
8	Dart	🌐 📱	57.4
9	Rust	🌐 🗨️ ⚙️	55.5
10	Scala	🌐 📱 🗨️	55.3

FIGURE 4.1 – Classement des langages de programmation [15].

Stack Overflow a interrogé sa communauté de développeurs sur les frameworks, les bibliothèques et les autres outils qu'ils utilisent le plus. En 2019, les 10 frameworks de développement web les plus populaires sont [16] :

1. jQuery : 48,3 %,
2. Angular/Angular.js : 32,4 %,
3. React.js : 32,3 %,
4. ASP.NET : 27,2 %,
5. Express : 19,5 %,
6. Spring : 17,2 %,
7. Vue.JS : 15,5 %,
8. Django : 12,3 %,
9. Flask : 11,7 %,
10. Drupal : 3.5 %.

## 4.4 Framework Spring

Afin d'avoir une idée générale sur le framework Spring, nous allons dans ce qui suit définir quelques concepts importants nécessaires à la compréhension de ce dernier.

### 4.4.1 Généralités

#### **Enterprise Java Beans (EJB)**

Les EJB offraient des services, tels que la gestion des transactions et la sécurité, qui étaient nécessaires pour développer des applications d'entreprise robustes et sécurisées et de permettre aux développeurs de faciliter la création d'applications distribuées pour les entreprises [18]. Cependant, le développement d'une application d'entreprise avec EJB n'a pas été facile, car le développeur devait effectuer diverses tâches qui entraînent la complexité de la fourniture de code pour les EJB c'est pour cette raison que le framework Spring est apparu [17].

### Principe du couplage fort / couplage faible

Un couplage fort signifie que les classes et les objets dépendent les uns des autres. En général, le couplage fort n'est pas bon car il réduit la flexibilité et la réutilisation du code, tandis que le couplage faible signifie la réduction des dépendances d'une classe qui utilise directement les différentes classes.

Un objet fortement couplé est un objet qui a besoin de connaître les autres objets et généralement très dépendant des autres.

La modification d'un objet dans une application fortement couplée nécessite souvent de modifier d'autres objets [19].

#### Exemple de couplage fort :

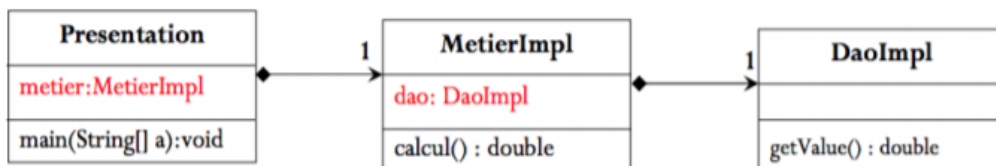


FIGURE 4.2 – Exemple de couplage fort [29].

Le couplage faible permet de réduire les interdépendances entre les composants d'un système dans le but de réduire le risque que les changements dans un composant nécessitent des changements dans tout autre composant.

Le couplage faible est un concept destiné à augmenter la flexibilité du système, à le rendre plus maintenable et à rendre l'ensemble du framework plus stable.

#### Exemple de couplage faible :

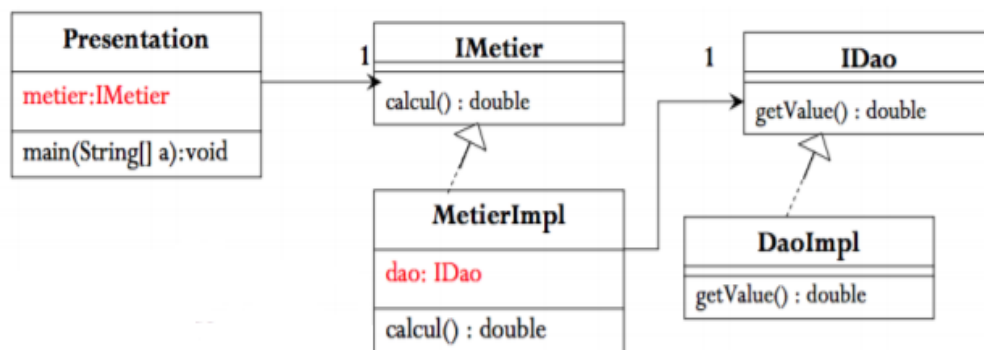


FIGURE 4.3 – Exemple de couplage faible [29].

### **IoC (Inversion de contrôle)**

L'IoC est un principe général de conception pour la programmation dans lequel l'application ne prend pas le contrôle du flux d'une manière séquentielle, elle le donne à un contrôleur externe qui pilote le flux. Cela donne de la flexibilité à l'application [20].

L'IoC est très complet et complexe. Son fonctionnement repose sur 3 éléments clés :

- La configuration par des fichiers XML pour lier les implémentations à leurs interfaces
- Les classes sont des beans et possèdent donc des getters et setters pour les champs à injecter
- L'injection est effectuée via un fichier de configuration (par défaut c'est ApplicationContext.xml) de Spring

Sans IoC, la création d'objets se ferait par l'opérateur new et cette création d'objets par instantiation augmente les dépendances (couplage fort) entre classes [21].

### **DI (Injection de Dépendance)**

L'injection de dépendances (Dependency Injection) est un mécanisme qui implémente le principe de l'inversion de contrôle (IoC). Cela consiste à créer dynamiquement (injecter) les dépendances entre les différentes classes en s'appuyant sur une description (fichier de configuration ou métadonnées) ou de manière programmatique. Ainsi les dépendances entre composants logiciels ne sont plus exprimées dans le code de manière statique mais déterminées dynamiquement à l'exécution [21].

### **DispatcherServlet**

La servlet DispatcherServlet gère toutes les requêtes et réponses HTTP de toutes les applications développées avec Spring MVC. Elle est en effet la tour de contrôle de Spring MVC, précisément c'est le Contrôleur dans le Design Pattern MVC. Chaque requête Web traitée par Spring MVC passe par cette servlet et fournit un point d'entrée unique dans toute application [21].

### **ORM (Mapping Object Relational)**

Un ORM (Object Relational Mapping) est un outil qui permet de passer d'un modèle logique objet vers un modèle physique relationnel. Il en existe plusieurs comme Hi-

bername, MyBatis mais le plus utilisé est Hibernate qui est souvent utilisé avec JPA (Java Persistence API). Il fait donc le lien entre la base de données et les objets Java [22].

### **Le mode transactionnel**

Si une application utilise une ou plusieurs bases de données, on a nécessairement besoin de déclarer des transactions. On peut utiliser les fonctionnalités des drivers JDBC mais dès que l'on doit gérer des services transactionnels qui s'imbriquent, ça commence à se compliquer et le code des services commence à être pas mal pollué par la gestion des transactions.

Avec Spring, la gestion des transactions est gérée par le conteneur, ce qui réduit et simplifie la tâche de développeur. Spring utilise la programmation par aspect (AOP) pour gérer son mode transactionnel et il permet de définir le gestionnaire de transactions (Hibernate, ...) [23].

#### **4.4.2 Définition**

Spring est un projet OpenSource pour l'environnement Java, développé par Rod Johnson et son utilité première était de remplacer les EJBs. Son infrastructure est similaire à un serveur d'application JavaEE mais l'avantage de Spring est que les classes n'ont pas besoin d'implémenter une interface pour être prises en charge. C'est en ce sens que Spring est qualifié de conteneur léger [23]. Il est modulaire et cela lui confère une certaine facilité pour la conception et la réalisation d'applications et permet d'accélérer le développement d'applications Java EE. Aujourd'hui il intègre d'autres implémentations et d'autres technologies comme ".NET" ou Python [21].

Les fonctionnalités offertes par Spring sont :

- La gestion transactionnelle
- Framework MVC
- Framework AOP (Programmation Orientée Aspect)
- Utilisation de l'IoC

Spring est devenu un Framework incontournable lorsqu'il s'agit de développer une application Java. JPA quant à lui est une API permettant d'interagir avec une base de données relationnelle via un ORM. Ce duo de Spring Data JPA constitue aujourd'hui le



socle de référence des applications Java utilisant une base de données relationnelle [24].

### 4.4.3 JPA /Hibernate

Face à la popularité croissante des ORM, la communauté Java a décidé d'éditer un certain nombre de règles généralistes regroupées au sein du standard Java Persistence API ou JPA. JPA est donc une interface définissant un certain nombre de mots-clés (annotations) et de normes à respecter dont le but est d'aider les développeurs à organiser les données relationnelles au sein d'une application et vise à améliorer considérablement la mise en œuvre des couches d'accès aux données. Les différentes solutions/implémentations se sont adaptées à ce nouveau standard, pour permettre une meilleure adoption de leurs frameworks [25].

En ce qui concerne Hibernate, il s'agit d'une implémentation du JPA et c'est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application ainsi que la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant.

Les bases de données supportées sont les principales du marché : DB2, Oracle, MySQL, PostgreSQL, Sybase, SQL Server, Sap DB, Interbase... [26].

Hibernate/JPA permet de sauvegarder rapidement un objet java dans une base de données. Il permet de s'affranchir des requêtes SQL écrite à la main et le plus souvent difficile à maintenir dans le temps.

Le mapping objet relationnel utilisé dans notre application est basé sur JPA /Hibernate.

### 4.4.4 Modules du framework Spring

Le framework Spring peut être considéré comme un ensemble de sous-frameworks, également appelés couches ou modules reposants tous sur le module Spring Core [27] :

1. Spring Core : le noyau, qui contient à la fois un ensemble de classes utilisées par toutes les briques du framework et le conteneur léger.

2. Spring Context : ce module étend Spring Core. Il fournit une sorte de base de données d'objets, permet de charger des ressources (telles que des fichiers de configuration) et supporte l'internationalisation (I18N), EJB.
3. Spring AOP : permet d'intégrer de la programmation orientée aspect.
4. Spring DAO : ce module permet d'abstraire les accès à la base de données avec notamment une implémentation pour JDBC, d'éliminer le code redondant. Il fournit en outre une gestion des transactions.
5. Spring ORM : cette partie permet d'intégrer des frameworks de mapping Object/Relationnel tel que Hibernate, JDO ou iBatis avec Spring. La quantité de code économisé par ce package peut-être très impressionnante.
6. Spring Web : Ensemble d'utilitaires pour les applications web. Il contient notamment Spring Web MVC, qui a une architecture de type MVC (Model-View-Controller) et ses composants servent pour développer des applications Web flexibles et faiblement couplées. Le modèle MVC permet de séparer les différentes parties d'une application web à savoir la gestion de requêtes d'entrée envoyées par le client, la logique métier et logique UI (affichage résultats en réponses aux requêtes) tout en assurant un couplage moins fort (Lazy) entre les différentes classes de l'application.
  - La partie Modèle encapsule les données de l'application ; ces données sont en général définies et traitées par des simples classes java.
  - La partie Vue est responsable du rendu des données du modèle et, en général, elle génère une sortie HTML que le navigateur du client peut interpréter.
  - La partie Contrôleur est responsable du traitement des demandes des utilisateurs et de la construction d'un modèle de vue approprié pour le rendu du résultat [28].

Ces modules peuvent être utilisés séparément lors de la création d'une application Web et peuvent également être regroupés pour fournir de meilleures fonctionnalités dans une application Web et faciliter sa maintenance et son extension.

#### 4.4.5 Architecture du Framework Spring MVC

Les applications web basées sur spring reposent sur trois couches principales :

- **La couche DAO** : s’occupe de l’accès aux données, le plus souvent des données persistantes au sein d’un SGBD en utilisant un ORM tel que Hibernate.
- **La couche Métier** : implémente les traitements «métier» de l’application. Cette couche est indépendante de toute forme d’interface avec l’utilisateur. C’est généralement la couche la plus stable de l’architecture.
- **La couche Web** : représente la partie contrôle qui orchestre les accès à l’application ainsi que les interfaces (graphiques souvent) qui permet à l’utilisateur de piloter l’application et d’en recevoir des résultats [29].

L’architecture du framework Spring est illustrée dans la Figure 4.4 suivante :

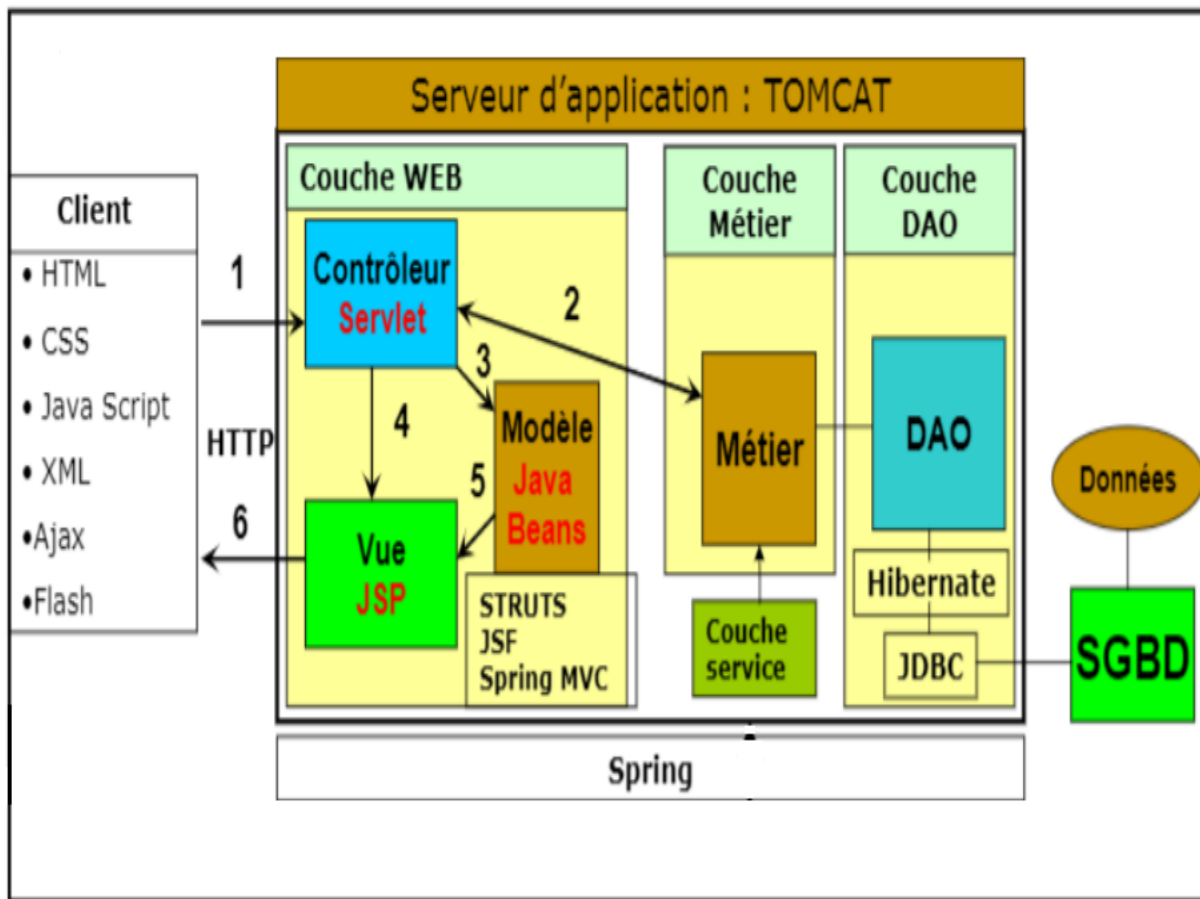


FIGURE 4.4 – Architecture du framework Spring MVC [29].

#### 4.4.6 Fonctionnement du Framework Spring MVC

Le framework Spring MVC se base sur le principe décrit par le schéma de la Figure 4.5 ci-dessous :

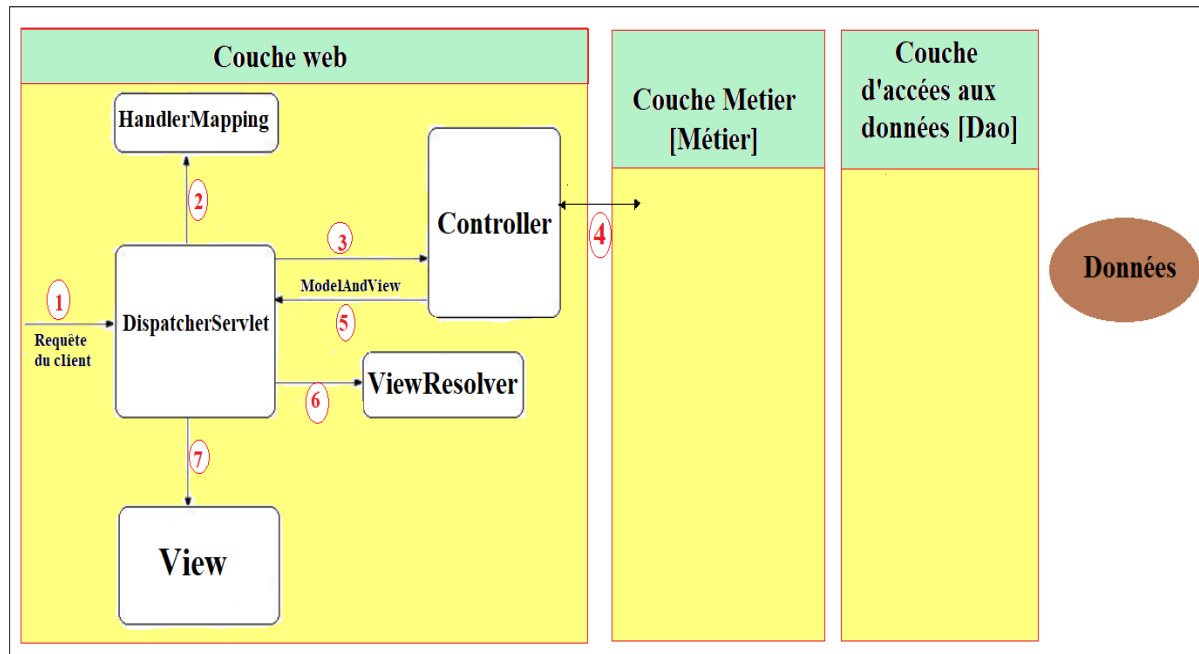


FIGURE 4.5 – Fonctionnement du framework Spring MVC [23].

Les étapes sont :

1. Le DispatcherServlet reçoit une requête envoyée par le client.
2. Le DispatcherServlet consulte son Handler Mapping pour connaître le contrôleur dont le nom de bean (contenant le mapping du contrôleur) correspond à la requête envoyée.
3. Le DispatcherServlet dispatche la requête pour le contrôleur identifié.
4. Le contrôleur traite la requête de l'utilisateur, pour ce faire il peut avoir besoin de l'aide de la couche métier, une réponse est ainsi renvoyée au contrôleur.
5. Le contrôleur retourne au DispatcherServlet un objet de type ModelAndView possédant comme paramètres au minimum le nom logique de la vue à renvoyer.
6. Le DispatcherServlet consulte son View Resolver lui permettant de trouver la vue dont le nom logique correspond à la vue renvoyée par le contrôleur.
7. Le DispatcherServlet communique la requête à la vue associée, ainsi la vue en question va être affichée au client [23].

## 4.5 Langages et outils de développement

Une grande variété de langages de programmation web existe sur le net. Le choix peut se porter également sur des frameworks de développements web qui regroupent des fonctionnalités très intéressantes. Dans le cadre de notre projet, nous avons fait le choix des langages et des outils ci-dessous :

### **Visual Paradigm :**

Visual Paradigm for UML est comme son nom le laisse supposer, un logiciel permettant aux programmeurs de mettre en place des diagrammes UML. Disposant d'un outil créant des rapports personnalisables aux formats PDF, Word ou HTML afin de les partager et les publier sur Internet, cette application est compatible avec de nombreuses applications, standards et environnements. Ainsi, vous pourrez générer notamment des diagrammes de séquences ou de cas d'utilisation et ainsi produire du code source dans de nombreux langages comme le Java ou encore le C++, ou bien faire l'inverse, générer des diagrammes à partir de code déjà existant [30].

### **Tomcat :**

Apache Tomcat est un conteneur libre de servlets et JSP Java EE. Issu du projet Jakarta, Tomcat est un projet principal de la fondation Apache. Il implémente les spécifications des servlets et des JSP du Java CommunityProcess1. Il est paramétrable par des fichiers XML et de propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP [31].

### **Bootstrap :**

Bootstrap est une collection d'outils utile 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 [32].

### **Wampserver :**

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données [33].

**MySQL (My Structured Query Language) :**

MySQL est un système de gestion de bases de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il permet de traiter et de maintenir de grosses bases de données avec une grande fiabilité [34].

**Maven :**

Maven est un outil permettant d'automatiser la gestion de projets Java.

Il offre entre autres les fonctionnalités suivantes :

- Compilation et déploiement des applications Java (JAR, WAR)
- Gestion des librairies requises par l'application
- Exécution des tests unitaires
- Génération des documentations du projet (site web, pdf, Latex)
- Intégration dans différents IDE (Eclipse, JBulder) [35].

**Flickr :**

Flickr, de l'anglais to flick through, feuilleter, est un site web de partage de photographies et de vidéos gratuit, il offre une Interface de programmation (API) permettant à des programmeurs indépendants de développer des services, dans le cas de notre application cette API nous a permis de sauvegarder les photos de profils des utilisateurs dans le serveur flickr avec 1 To de mémoire à notre disposition [36].

**Git :**

Git est de loin le système de contrôle de version le plus largement utilisé aujourd'hui. Git est un projet open source avancé, qui est activement maintenu. À l'origine, il a été développé en 2005 par Linus Torvalds, le créateur bien connu du noyau du système d'exploitation Linux. De plus en plus de projets de développement reposent sur Git pour le contrôle de version, y compris des projets commerciaux et en open source. les serveurs de Git sont : Github, GitLab , Bitbucket [37].

**JavaScript :**

Langage de développement web objet, utilisé dans la conception de pages Web. JavaScript créé en 1995 par Brendan Eich pour la société Netscape et fut intégré pour la première fois dans le navigateur Netscape Navigator 2.0. Le code JavaScript peut être directement intégré dans une page web et est exécuté par le navigateur du côté client [38].

**HTML (Hypertext Markup Language) :**

L'HTML est un langage informatique largement utilisé dans la création des pages web.

Son acronyme signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage.

Ce n'est pas à proprement parlé un langage de programmation, mais plutôt un langage qui permet de mettre en forme du contenu. Les balises permettent de mettre en forme le texte et de placer des éléments interactifs, tel des liens, des images ou bien encore des animations [39].

### **CSS :**

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML [39].

### **Ajax :**

Asynchrones JavaScript And XML (AJAX) : désigne un nouveau type de conception de pages Web permettant l'actualisation de certaines données d'une page sans procéder au rechargement total de cette page. Cette méthode de conception repose sur la combinaison de technologies déjà existantes : HTML/CSS, JavaScript/DOM, XML et les requêtes HTTP [40].

### **Illustrator :**

Adobe Illustrator est un logiciel de dessin faisant partie de la suite Adobe. Créé en 1987, Illustrator est devenu l'un des outils informatiques les plus utilisés des graphistes, illustrateurs ou encore typographes. Sa particularité réside dans le fait qu'il permet de créer des images vectorielles [41].

Cet outil nous a permis de créer notre logo illustré dans la Figure 4.6 suivante :



FIGURE 4.6 – Logo de l'application.

## 4.6 Notre application de gestion des conteneurs de l'EPB

L'interface graphique est le moyen d'échange entre l'humain et la machine, il est important qu'elle soit bien conçue, ergonomique et intuitive afin que l'utilisateur la comprenne facilement et lui rend la tâche facile. Dans ce qui suit une présentation des captures d'écrans des interfaces que nous avons jugés les plus importantes.

### 4.6.1 Interface d'authentification

Cette interface s'affiche au lancement de notre application, le formulaire d'authentification nous demandera d'introduire le nom d'utilisateur et le mot de passe. Cette étape met en valeur l'aspect sécurité de l'application qui est fourni par Spring Security qui est un cadre d'authentification et de contrôle d'accès puissant. C'est la norme de facto pour sécuriser les applications Spring. Il se concentre sur la fourniture à la fois d'authentification et d'autorisation aux applications Java [42], cette interface est illustrée la Figure 4.7 suivante :

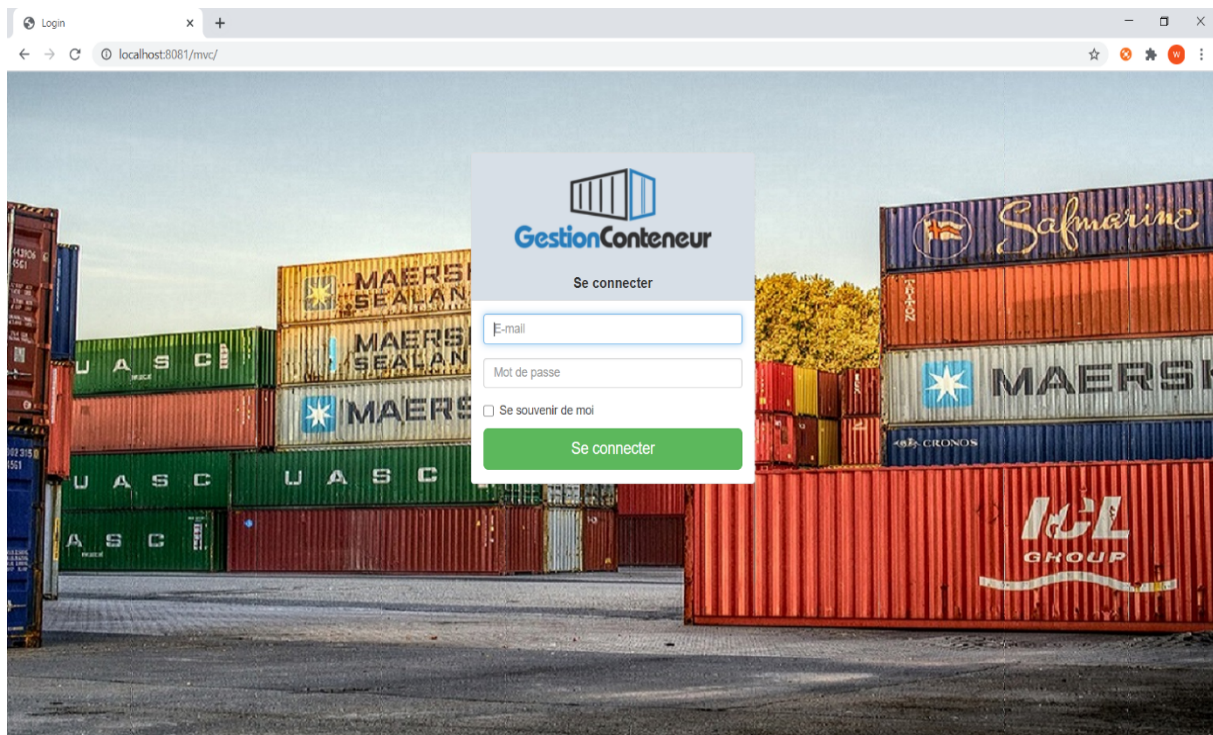


FIGURE 4.7 – Interface d'authentification.



## 4.6.2 Interface Article BL

Cette interface affiche la liste des Articles BL's qui existe dans la base de données avec comme option la possibilité d'ajouter, modifier ou de supprimer un Article BL (le clic sur ce bouton nous affiche une boite de confirmation avec la possibilité de confirmer ou d'annuler la suppression), ainsi que le bouton détails qui permet d'afficher plus d'informations sur l'Article BL correspondant dans un modal (fenêtre pop), ceci est illustré dans la Figure 4.8 suivante :

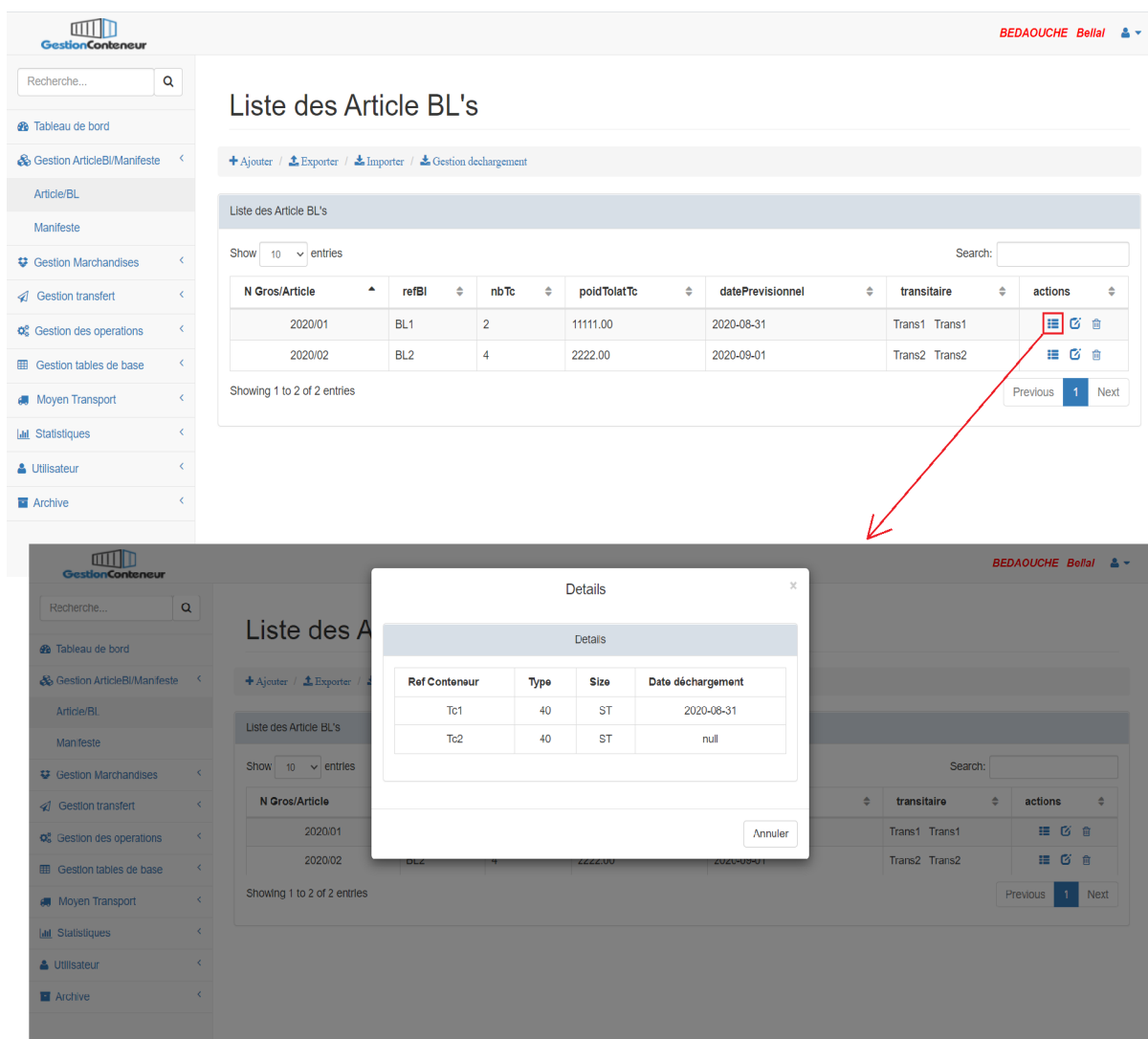


FIGURE 4.8 – Interface Article BL.

### 4.6.3 Interface Ajouter Article BL

L'interface suivante permet à l'agent de l'EPB d'ajouter un nouvel Article BL, il lui suffira de remplir le formulaire et de rechercher les conteneurs qu'il souhaite attribuer au BL dans le panel 'Recherche' cela en choisissant le conteneur dans la liste déroulante (cette opération se fera grâce à une fonction Ajax, qui permet de récupérer le conteneur choisi et de l'afficher dans le tableau en dessous sans avoir à recharger la page et de perdre les données saisies dans le formulaire). Le bouton ajouter conteneur est un raccourci qui permet d'ajouter un conteneur s'il ne figure pas dans la liste déroulante. Cette interface est illustrée dans la Figure 4.9 suivante :

The screenshot shows the 'Nouvelle ArticleBI' form in the 'GestionConteneur' application. The sidebar on the left contains navigation items: 'Tableau de bord', 'Gestion ArticleBI/Manifeste' (with sub-items 'Article/BL' and 'Manifeste'), 'Gestion Marchandises', 'Gestion transfert', 'Gestion des operations', 'Gestion tables de base', 'Moyen Transport', 'Statistiques', 'Utilisateur', and 'Archive'. The main form has the following fields:

- Référence BI :** Input field with placeholder 'Référence BI'.
- N° Article :** Input field with placeholder 'N Article'.
- Transitaire :** Dropdown menu with placeholder 'Veuillez sélectionner un transitaire'.
- Poid total Tc :** Input field with placeholder 'Poid total Tc'.
- Date Prévisionnel :** Input field with placeholder 'jj/mm/aaaa' and a calendar icon.

Below the form are two buttons: 'Enregistrer' (blue) and 'Annuler' (red). The 'Recherche' section includes a dropdown menu for 'Conteneur' with placeholder 'Veuillez sélectionner un conteneur', a search icon, and a '+ Ajouter d'un conteneur' button. Below this is a table with the following structure:

Référence conteneur	Type	Size	Action

FIGURE 4.9 – Interface Ajouter Article BL.

### 4.6.4 Interface Modifier Article BL

Cette interface permet de modifier un Article BL en affichant ses informations dans le formulaire de modification comme illustré dans la Figure 4.10 suivante :

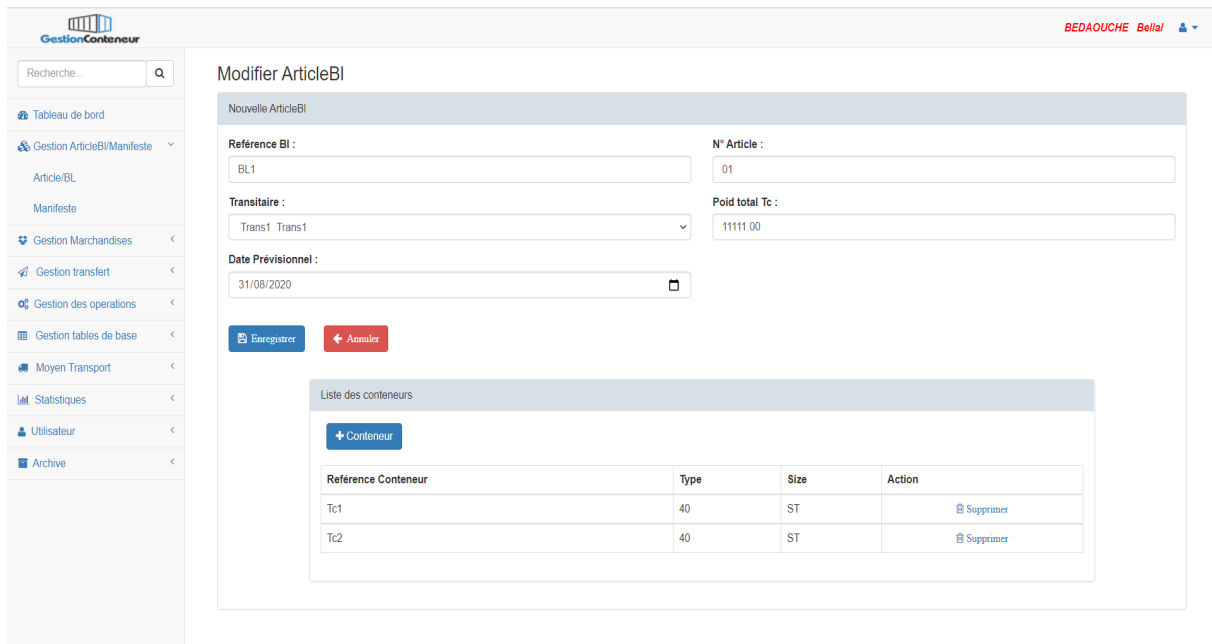


FIGURE 4.10 – Interface Modifier Article BL.

### 4.6.5 Interface Archive

Cette interface affiche la liste des manifestes traités ce qui nous permet de désencombrer nos interfaces et de les aérer afin que l'utilisateur ait une vision actuelle des données et une autre sur les données précédentes. Elle est représentée dans la Figure 4.11 suivante :

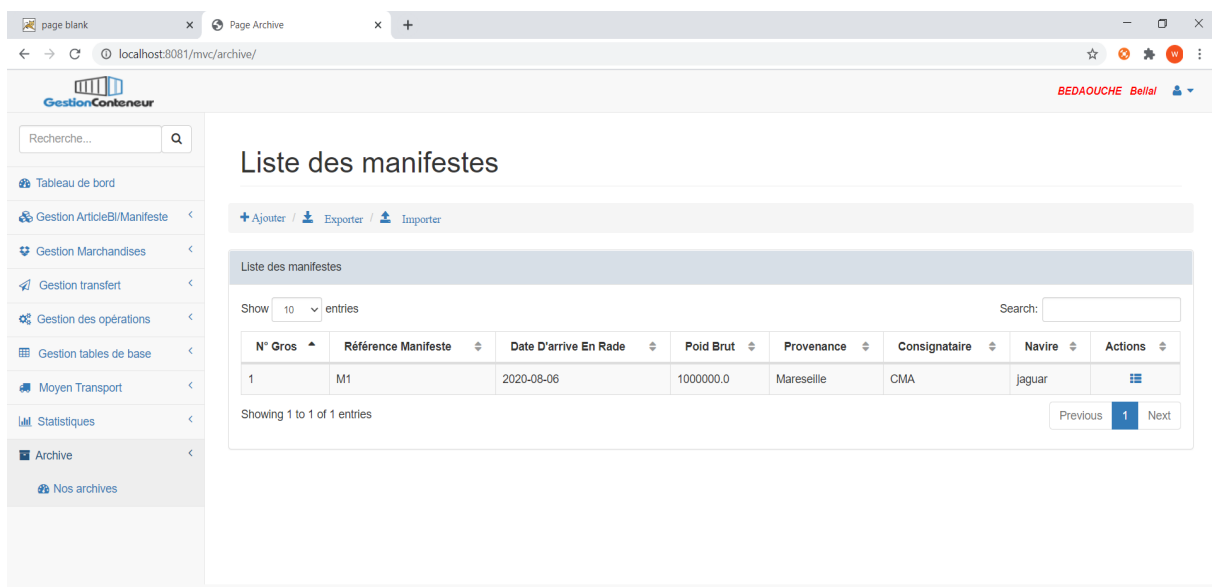


FIGURE 4.11 – Interface Archive.

Le clic sur le bouton détails nous permet d'afficher l'interface suivante :

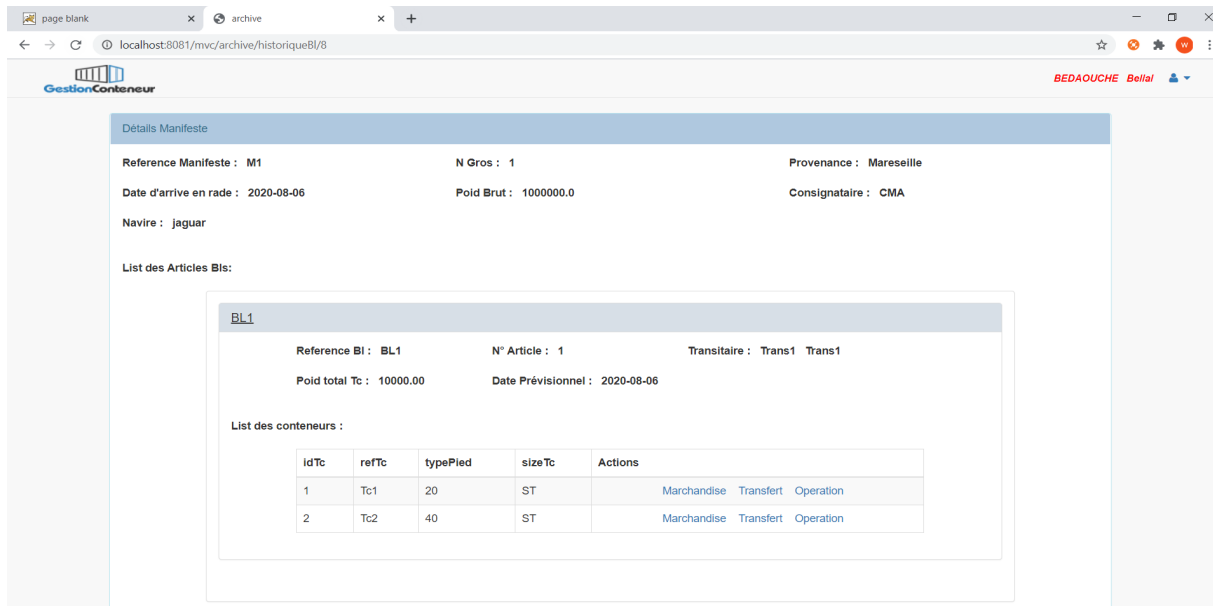


FIGURE 4.12 – Interface Détails Archive.

Cette interface affiche les informations sur le manifeste en question, ses Articles BL's, et pour chaque Article BL les informations le concernant ainsi que la liste de ses conteneurs et pour chaque conteneur la possibilité de visualiser des détails sur son transfert, ses marchandises ainsi que les opérations effectuer.

#### 4.6.6 Interfaces Statistiques

Cette interface affiche des statistiques concernant le nombre de conteneurs visités durant une période précise par l'employé de l'EPB (pour cela il doit renseigner les deux dates et cliquer sur le bouton envoyer) ainsi que le taux de conteneurs visités (le calcul de cette valeur se fait selon une formule délivrée par l'organisme d'accueil), comme illustré dans la Figure 4.13 suivante :

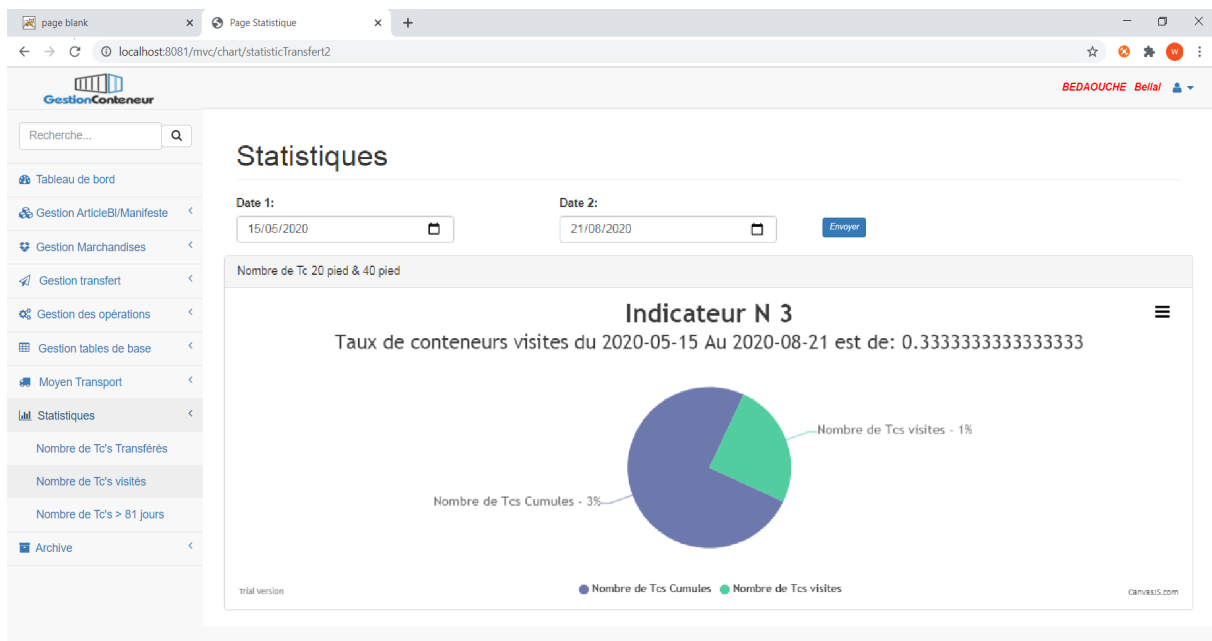


FIGURE 4.13 – Statistique nombre de conteneurs visités.

Cette interface permet de calculer « le nombre de TC’s dépassant les 81 jours de séjour » et « le taux de conteneurs visités » entre les deux dates saisies par l’employé de l’EPB. Ce calcul se fait selon les formules délivrées par l’organisme d’accueil. Les deux liens permettent d’afficher les détails sur les conteneurs concernés, comme illustré dans la Figure 4.14 suivante :

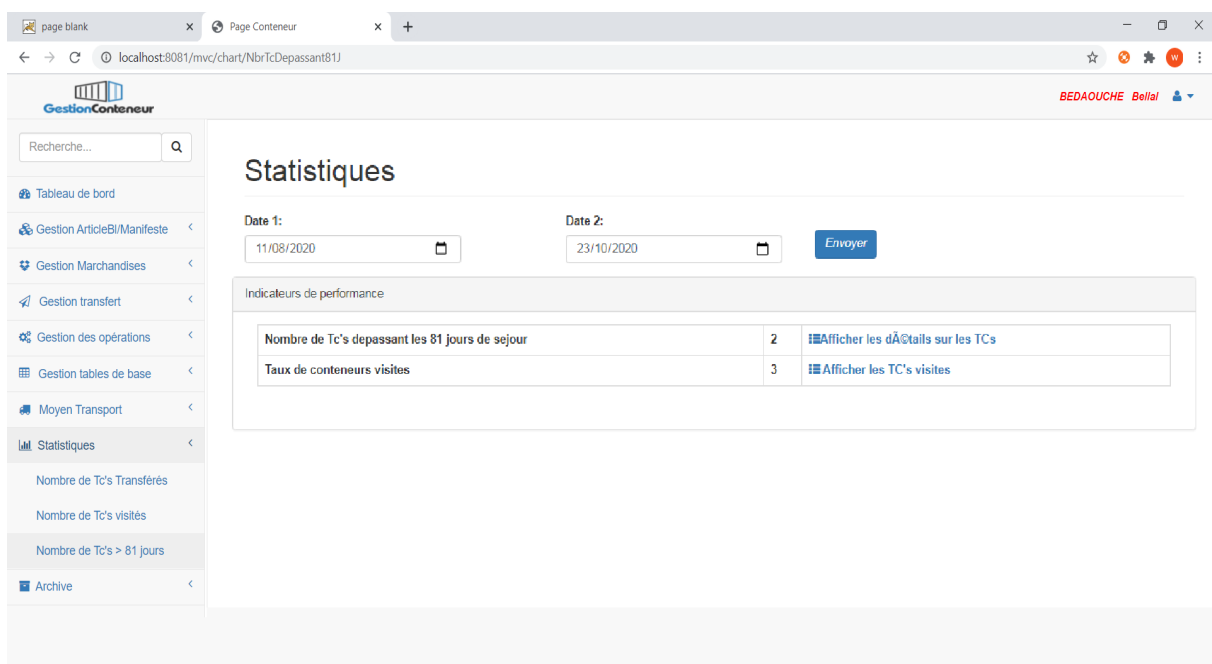


FIGURE 4.14 – Statistique nombre de TC’s dépassant les 81 jours.

### 4.6.7 Interface Utilisateurs

Cette interface permet à l'administrateur de l'application de visualiser les comptes utilisateurs, avec comme options : Activer le compte, Désactiver le compte et Réinitialiser le mot de passe. Ceci est illustré dans la Figure 4.15 qui suit :

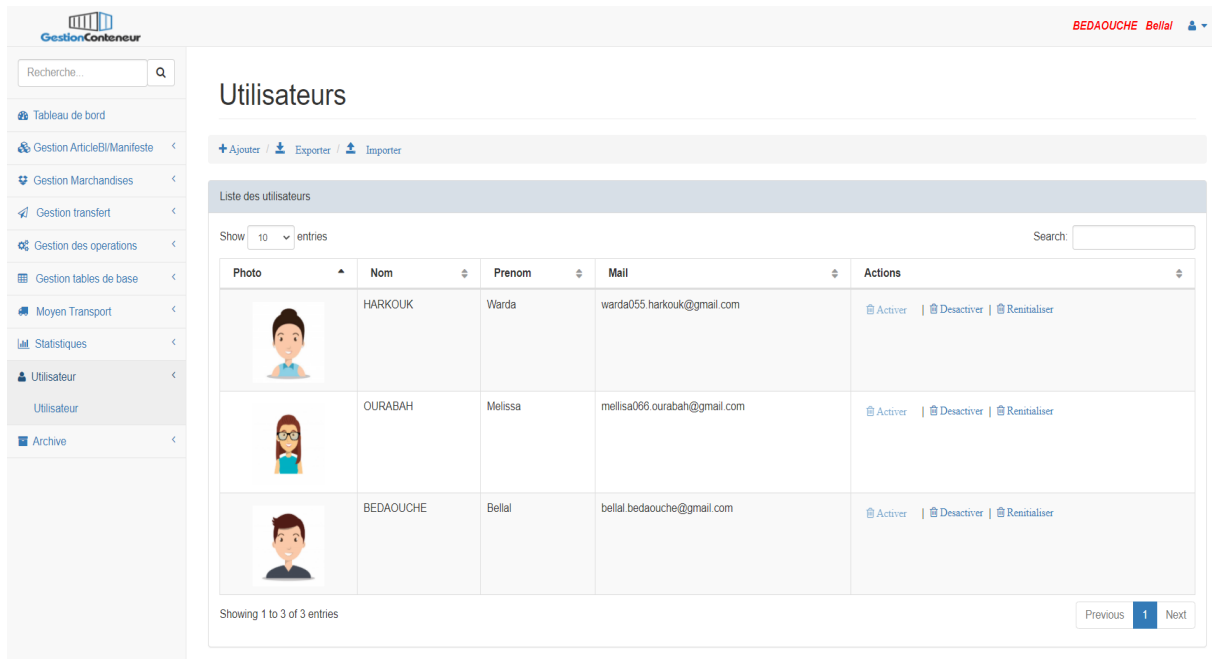


FIGURE 4.15 – Interface Utilisateurs.

### 4.6.8 Modifier Profil Utilisateur

Les utilisateurs de l'application ont la possibilité de modifier leurs profils en appuyant sur le menu "Profil utilisateur" situé dans la barre de navigation. Ceci est illustré dans la figure 4.16 qui suit :

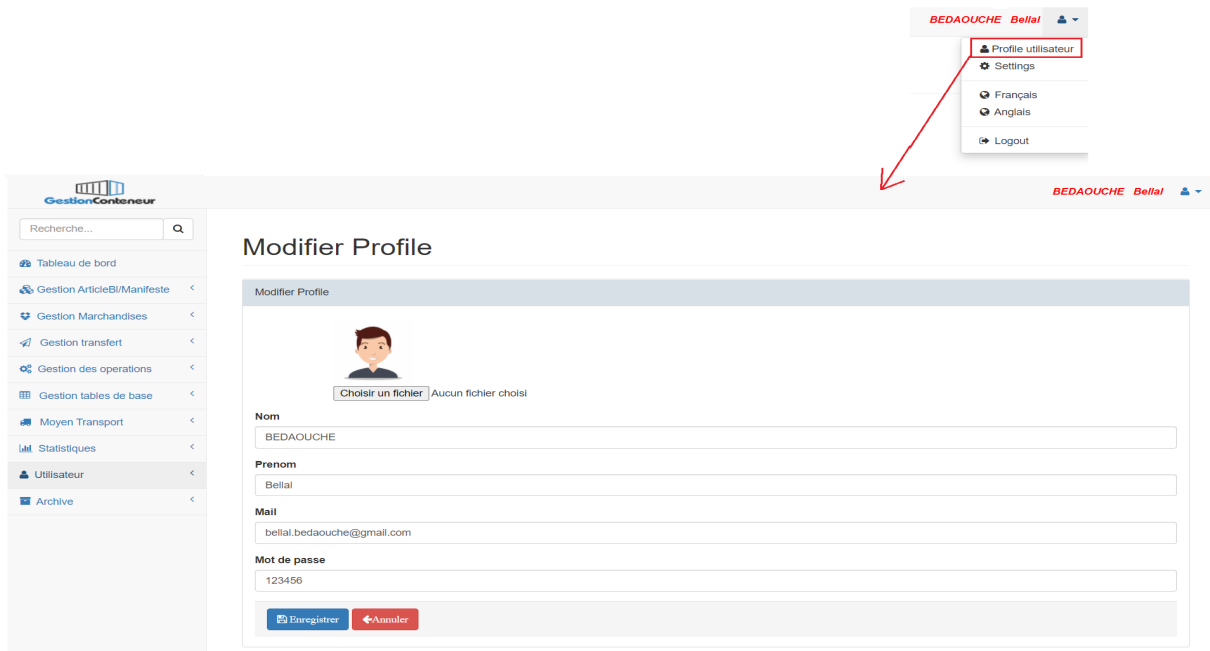


FIGURE 4.16 – Interface Modifier Profil Utilisateur.

Parmi les fonctionnalités de notre application l'internationalisation (i18n) qui permet de basculer de la langue française (langue par défaut) vers la langue anglaise, cette fonctionnalité est disponible dans le menu illustré dans la Figure 4.17 suivante :

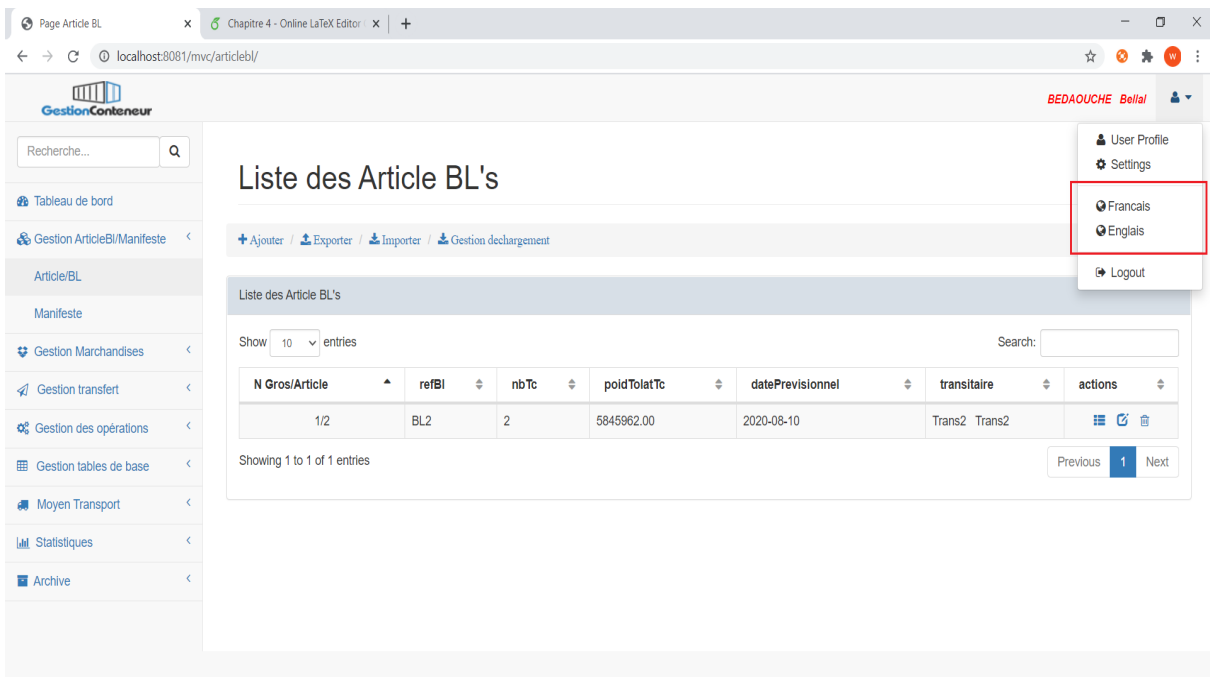


FIGURE 4.17 – Internationalisation de l'application.

## 4.7 Tests

Nous voilà arrivés à la phase des tests qui consiste à identifier un nombre maximal de comportements problématiques du logiciel. Après avoir effectué plusieurs démonstrations auprès du personnel du département de la zone logistique de Tixter, des problèmes ont été détectés. Grâce à ces démonstrations, nous avons pu relever quelques bugs que nous avons pu corriger par la suite, ce qui nous a permis ainsi d'augmenter et d'améliorer la qualité de notre logiciel.

## 4.8 Déploiement

Après avoir développé notre application web en local, elle nécessite un serveur HTTP pour l'héberger dans notre cas on a utilisé le serveur Apache Tomcat précédemment défini. Pour cela, on va devoir réaliser les tâches suivantes :

### 4.8.1 Création du fichier War du projet Spring MVC

Maven nous offre un ensemble de commandes nous permettant de compiler, tester et déployer notre application. Pour ce faire, Nous devons d'abord exporter notre projet dans un fichier war suivant ces commandes :

- Mvn compile : sert à compiler toutes les classes qui se trouvent dans le projet.
- Mvn test : exécute tous les tests unitaires qui existent dans le projet.
- Mvn package : permet de générer le package de l'application WAR qui contient l'application qui sera enregistré dans le fichier « target » de notre projet.

Ces commandes peuvent être exécuter soit en ligne de commande soit sous l'IDE STS(Spring Tool Suite) comme suit :

Clic droit sur le nom du projet -> Run As -> Maven build ... la fenêtre suivante s'affiche :



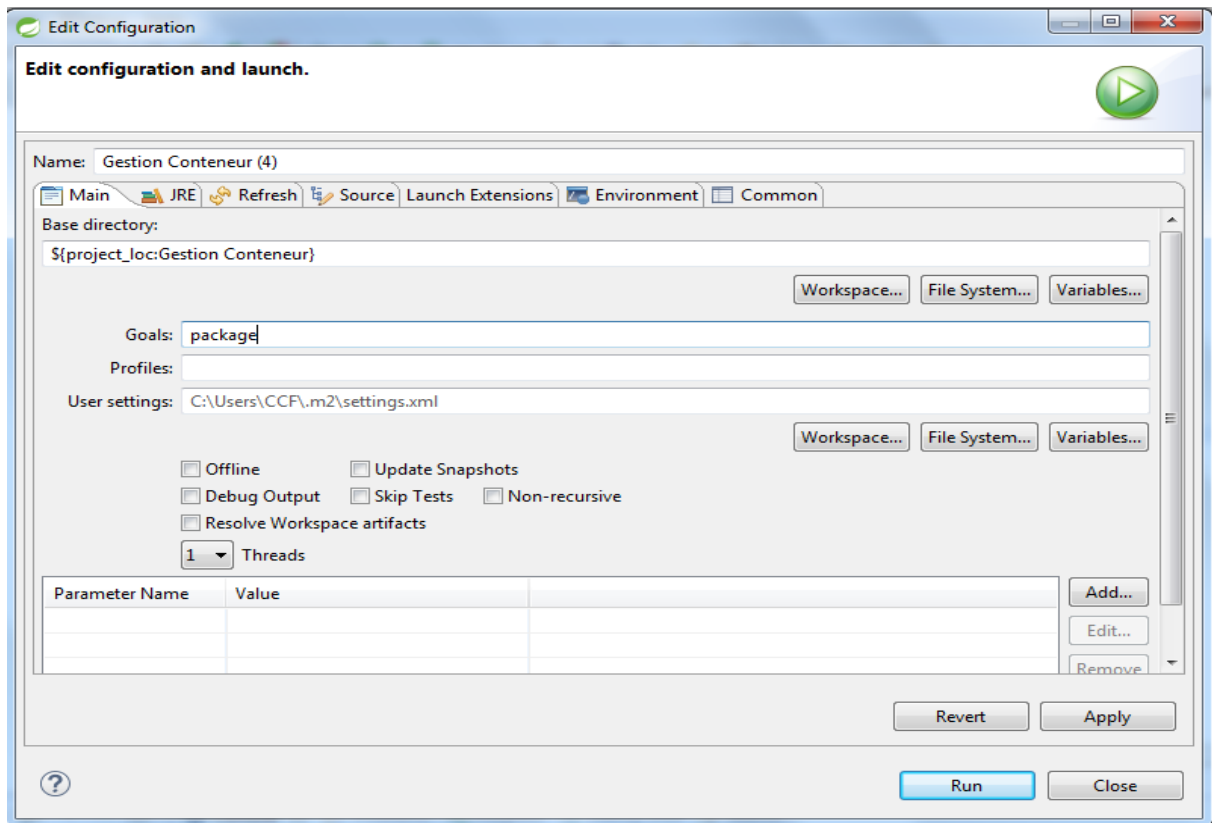


FIGURE 4.18 – Exécution des commandes Maven sous STS.

## 4.8.2 Déploiement du fichier WAR à l'aide d'Apache Tomcat Manager

Le déploiement de l'application se fait suivant ces étapes :

- 1- Après avoir générer le fichier war, démarrer Tomcat et accéder à son interface de configuration via un navigateur en tapant l'URL : localhost :8080.
- 2- Cliquer sur le bouton ManagerApp et introduire le nom d'utilisateur et mot de passe.
- 3- Après s'être authentifier affichage de la console d'administration, comme le montre la figure 19 suivante :



Gestionnaire d'applications WEB Tomcat

Message:

**Gestionnaire**

Lister les applications Aide HTML Gestionnaire Aide Gestionnaire Etat du serveur

Chemin	Version	Nom d'affichage	Fonctionnelle	Sessions	Commandes
/	None specified	Welcome to Tomcat	true	0	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis > 30 minutes
Ma_banque-0.0.1-SNAPSHOT	None specified		false	0	Démarrer Arrêter Recharger Retirer
/docs	None specified	Tomcat Documentation	true	0	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis > 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis > 30 minutes
gestion-conteneur 2	None specified		false	0	Démarrer Arrêter Recharger Retirer
/host-manager	None specified	Tomcat Host Manager Application	true	0	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis > 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis > 30 minutes
imvc-1.0.0-BUILD-SNAPSHOT	None specified		false	0	Démarrer Arrêter Recharger Retirer

**Deployer**

Emplacement du répertoire ou fichier WAR de déploiement sur le serveur

Chemin de contexte (requis)

Version (for parallel deployment)

URL du fichier XML de configuration

URL vers WAR ou répertoire:

**Fichier WAR à déployer**

Choisir le fichier WAR à téléverser  Aucun fichier choisi **1**

FIGURE 4.19 – Gestionnaire d'application web Tomcat.

4- Dans la partie « Fichier war à déployer » (encadré et numéroté 1 dans la figure) choisir le fichier war précédemment créé et cliquer sur le bouton « Déployer », ceci permet d'ajouter notre application à la liste des applications du serveur Tomcat (encadré et numéroté 2 dans la figure).

5- Le clic sur le bouton démarrer permet de lancer l'application indépendamment de STS.

## 4.9 Conclusion

Dans ce chapitre réservé à la partie développement d'une application de gestion des conteneurs de l'entreprise EPB, nous avons commencé par la présentation des langages et des outils de développement web que nous avons utilisé. Nous avons présenté, également, le framework Spring, très utilisé actuellement par les grandes entreprises de développement web dans le monde. Par ailleurs, nous avons présenté quelques interfaces de notre application pour montrer l'implémentation de quelques cas d'utilisation cités dans le chapitre 2. Il est à noter que l'entreprise d'accueil nous a donnée quelques

recommandations que nous avons prises en considération et qui ont largement amélioré l'application.

## CONCLUSION GÉNÉRALE ET PERSPECTIVES

Dans ce mémoire, nous avons présenté le département de la zone logistique extra-portuaire (DZLE) de l'EPB sous ses différents aspects organisationnels et fonctionnels afin de dégager les besoins fonctionnels pour une meilleure conception.

L'étude menée pendant notre stage au sein de cette direction nous a permis de recenser les lacunes, et de déduire les besoins concrets pour définir les objectifs à atteindre afin de répondre aux besoins. Le chef de service du bureau nous a exprimé son souhait d'automatiser tous les documents servant à la gestion et le suivi des conteneurs dès leurs arrivés au port de Bejaia jusqu'à leurs transferts vers le port sec de Bordj Bou Arreridj Tixter ceci en développant une application web.

La solution proposée répond aux critères de gestion et facilite toutes sortes d'interaction avec l'utilisateur en terme de saisie, de mise à jour mais aussi de génération des états imprimables tels que les bons de sortie, liste des opérations. . .

Pour la conception de notre application, nous avons utilisé le processus de développement UP /XP et le langage UML pour la réalisation des différents diagrammes couvrant toutes les phases du processus de développement, offrant ainsi une base solide sur laquelle nous nous sommes appuyés afin de passer à la partie réalisation du projet. Pour la mise en œuvre de l'application, nous avons eu recours à la programmation avec le Framwork Spring mvc de java et le SGBD MySQL.

Après la réalisation de l'application, nous avons effectué plusieurs tests, ces tests nous ont permis de souligner quelques manques concernant notre application, mais ces tests nous ont aussi permis d'ouvrir le champ des possibilités concernant nos perspectives

par rapport à l'amélioration de l'application, comme ajouter un système d'entreposage.

Enfin ce stage, ainsi que le travail fournit, nous a permis de nous épanouir, mais surtout d'avoir un aperçu sur le monde du travail et développer nos compétences en tant que développeur mais aussi notre capacité à travailler en équipe.

## BIBLIOGRAPHIE

- [1] Processus de développement UP, <http://files.gl3.webnode.fr/200000067-4b6db4c670/Chapitre1-ACOO2.pdf>, Consulté le 03/02/2020.
- [2] Unified process, <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/>, Consulté le 29/01/2020.
- [3] Méthodes agiles (RAD, XP), <https://www.commentcamarche.net/contents/477-methodesagiles-rad-xp>, 2017, consulté le 05/03/2020.
- [4] Régis Medina Dominic Williams Jean-Louis Bénard, Laurent Bossavit. Gestion de projet eXtreme Programming. Groupe Eyrolles, 2005. ISBN 2-212-11561-X.
- [5] Extreme programming (XP), <http://agileutile.fr/methodes/extreme-programming-xp/>, consulté le 03/05/2019.
- [6] GRADY Booch et al. Le guide de l'utilisateur UML, Eyrolles, Paris, 2003.
- [7] Pascal Roques, UML 2 : Modéliser une application web, Les Cahiers du Programmeur, Eyrolles ,4e édition, Paris, 2008.
- [8] Mounir OUNAS et Lyes GROUAHAN, Conception et réalisation d'une application web pour la gestion du service médecine du travail, Mémoire de Master professionnel, Administration et Sécurité des réseaux, Béjaia : Université Abderrahmane Mira de Bejaia, 2017-2018, 61p.
- [9] Pascal Roques, UML 2 : Modéliser une application web, Les Cahiers du Programmeur, Eyrolles ,4e édition, Paris, 2008.
- [10] Kirill Fakhroutdinov (2016), UML Use Case Diagrams, <https://www.uml-diagrams.org/use-case-diagrams.html>, Consulté le 20/05/2020

- [11] Transformation du diagramme de classe en modèle relationnel, <http://www.ismaik.rnu.tn/useruploads/files/Transformation%20du%20diagramme%20de%20classe%20en%20mod%C3%A8le%20relationnel.pdf>, Consulté le 23/04/2020.
- [12] Introduction au modèle relationnel, <https://web.maths.unsw.edu.au/lafaye/CCM/relation/relintro.htm>, consulté le 25/04/2020.
- [13] Kass Kasongo Ndjoko, Conception d'un système d'information pour le contrôle des structures des écoles, [https://www.memoireonline.com/03/15/8981/m\\_Conception-d-un-systeme-d-information-pour-le-contrle-des-structures-des-ecoles-11.html](https://www.memoireonline.com/03/15/8981/m_Conception-d-un-systeme-d-information-pour-le-contrle-des-structures-des-ecoles-11.html), 2012, Consulté le 26/07/2020.
- [14] Application web, sur le site Idéematic, <https://www.ideematic.com/dictionnaire-digital/application-web/>, Consulter le 25/07/2020.
- [15] Guilloux, Michael, Meilleurs langages en 2019 selon l'IEEE, <https://programmation.developpez.com/actu/276292/Meilleurs-langages-en-2019-selon-l-IEEE-Python-leader-pour-la-troisieme-annee-consecutive-il-s-impose-dans-tous-les-domaines-dans-lesquels-il-est-utilise-du-developpement-web-a-l-embarque/>, Le 7 septembre 2019, Consulté le 26/07/2020.
- [16] Développement Full Stack : Les langages informatiques les plus populaires en 2019, <https://www.wis-ecoles.com/langages-informatiques-populaires-2019/>, Consulté le 02/08/2020.
- [17] Introduction à Spring Framework, <https://www.geeksforgeeks.org/introduction-to-spring-framework/>, Consulter le 25/07/2020.
- [18] DOUDOUX, Jean-Michel, Les EJB (Entreprise Java Bean), <https://www.jmdoudoux.fr/java/dej/chap-ejb.htm>, Consulté le 20/07/2020.
- [19] Différence entre couplage faible et couplage fort en Java, <https://waytolearnx.com/2020/03/difference-entre-couplage-faible-et-couplage-fort-en-java.html>, 2018, Consulté le 01/08/2020.
- [20] SHANKAR, Ramya, Les frameworks Java. <https://hackr.io/blog/java-frameworks>, 05 août 2020, Consulté le 10/08/2020.
- [21] Spring : Généralités, sur le site ocamil, <http://www.ocamil.com/index.php/spring/generalites>, 13 décembre 2017, Consulté le 25/06/2020.

- [22] MEAUDRE, Adrien, Java et Spring - Concevoir, construire et développer une application, sur le site eni, (ISBN : 978-2-7460-5514-8), <https://www.editions-eni.fr/open/mediabook.aspx?idR=a06b06f9bc29affaf3e45642b0a4a390>, Consulté le 05/06/2020.
- [23] MARTIN, Eric. Spring framework, sur le site SUPINFO, <https://www.supinfo.com/articles/single/362-spring-framework>, 23/09/2015, Consulté le 02/08/2020.
- [24] Maîtriser Spring et JPA avec Hibernate, sur le site DNG Consulting, <https://www.dng-consulting.com/spring-hibernate/>, Consulté le 03/08/2020.
- [25] Thibaut, Différence entre Hibernate et JPA, sur le site w3blog, <https://w3blog.fr/2016/01/15/difference-entre-hibernate-et-jpa/>, 15 janvier 2016, Consulté le 30/07/2020.
- [26] DOUDOUX, Jean-Michel, Hibernate, <https://www.jmdoudoux.fr/java/dej/chap-hibernate.htm>, 2019, Consulté le 21/07/2020.
- [27] Hostettler, Steve, Spring : théorie & pratique. <https://ze-key.developpez.com/articles/spring/>, Consulté le 20/05/2020.
- [28] Spring : Spring MVC, <http://www.ocamil.com/index.php/spring/spring-mvc>, 13 décembre 2017, Consulté le 29/06/2020.
- [29] YOUSFI, Mohmed, Architecture J2EE, sur le site EDOC, <https://edoc.pub/jee-spring-mvc-pdf-free.html>, Consulté le 20/07/2020.
- [30] Publié par La Rédaction, Visual Paradigm for UML : Un logiciel de conception UML très complet, <http://www.clubic.com/telecharger-fiche384046-visual-paradigm-for-uml.html>, consulté le 17/08/2020.
- [31] TOMCAT : Serveur d'applications Java, <https://artduweb.com/tutoriels/tomcat>, 15/04/2016, Consulté le 15/06/2020.
- [32] Bootstrap (framework). [https://fr.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework)), 24 juillet 2020. Consulté le 15/05/2018.
- [33] Romain Bourdon, WAMPSEVER, plate-forme de développement Web sous Windows, <http://www.wampserver.com/>, 15/05/2018, Consulté le 12/08/2020.



- [34] Système de gestion de base de données, [https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_de\\_gestion\\_de\\_base\\_de\\_donn%C3%A9es](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es), 25 juillet 2020, Consulté le 05/02/2020.
- [35] Apache Maven Project, [http://www-igm.univ-mlv.fr/dr/XPOSE2003/site\\_maven/:text=A%20quoi%20sert%20Maven%20%3F,applications%20Java%20\(JAR%2C%20WAR\)](http://www-igm.univ-mlv.fr/dr/XPOSE2003/site_maven/:text=A%20quoi%20sert%20Maven%20%3F,applications%20Java%20(JAR%2C%20WAR)). Consulté le 01/02/2020.
- [36] <https://audreytips.com/glossaire-web/flickr/>, Consulté le 01/02/2020.
- [37] Qu'est-ce que Git?, <https://www.atlassian.com/fr/git/tutorials/what-is-git>, Consulté le 04/02/2020.
- [38] Jonathan Fine, Core JavaScript Documentation, 14 novembre 2017, Consulté le 21/07/2020.
- [39] HTML, <http://glossaire.infowebmaster.fr/html/>, Consulté le 10/02/2020.
- [40] Michel Plasse, Développez en Ajax, édition EYROLLES, 2017.
- [41] Illustrator, <https://edutechwiki.unige.ch/fr/Illustrator>, 25 août 2016, Consulté le 10/02/2020.
- [42] Sécurité du printemps. <https://spring.io/projects/spring-security>. Consulté le 11/02/2020.

## Résumé

---

*Ce travail est réalisé en vue de l'obtention du diplôme master professionnel en génie logiciel, il consiste à la conception et la réalisation d'une application web pour la gestion et le suivi des conteneurs, dès leur arrivée au port de Bejaia jusqu'à leur transfert vers le port sec de Bordj Bou Arreridj Tixter. notre stage a été effectué au sein de l'entreprise portuaire de Bejaia (EPB), précisément au département de la zone logistique de Tixter où ces employés représentent les utilisateurs principaux de notre application. Cette dernière permet de pallier les insuffisances et d'améliorer le fonctionnement du département en question. Ce travail a été réalisé en utilisant le langage de modélisation Unified Modeling Language (UML) qui permet de schématiser la solution ainsi que le langage de programmation java avec Spring comme Framework et MySQL comme Système de Gestion de base de données (SGBD).*

**Mots clés :** UML, JAVA, MySQL, SGBD, Le framework Spring.

## Abstract

---

*This work is carried out with a view to obtaining the professional master's degree in software engineering, it consists in the design and production of a web application for the management and monitoring of containers, from their arrival at the port of Bejaia until their transfer to the dry port of Bordj Bou Arreridj Tixter. our internship was carried out at the Port Company of Bejaia (EPB), precisely in the logistics area department of Tixter where these employees represent the main users of our application. The latter makes it possible to remedy the shortcomings and improve the functioning of the department in question. This work was carried out using the Unified Modeling Language (UML) modeling language which allows to schematize the solution as well as the java programming language with Spring as Framework and MySQL as Database Management System (DBMS).*

**Key words :** UML, JAVA, MySQL, SGBD, The Spring framework.