

ANNÉE UNIVERSITAIRE 2015-2016

UNIVERSITE SIDI MOHAMED BEN ABDELLAH

---

FACULTÉ DES SCIENCES ET TECHNIQUES FÈS

DÉPARTEMENT D'INFORMATIQUE



## PROJET DE FIN D'ETUDES

MASTER SCIENCES ET TECHNIQUES

SYSTÈMES INTELLIGENTS & RÉSEAUX

---

Conception et réalisation du module Camion du système d'information metier  
refonte de marsa maroc

---



LIEU DE STAGE : CGI CASABLANCA

RÉALISÉ PAR : OUBELQUE KHALID

SOUTENU LE : 14 / 06 / 2016

ENCADRÉ PAR : ABDERRAHIM BENABBOU & OMAR CHAMIMI

DEVANT LE JURY COMPOSÉ DE :

MR AZEDDINE ZAHI

MR MED CHAOUKI ABOUNAIMA

MR RACHID BEN ABBOU

MR ABDERRAHIM BENABBOU



# Dédicace

*Je ne saurai trouver les expressions les plus adéquates pour décrire mes sentiments de reconnaissance, de respect et de gratitude envers ceux à qui je dédie ce modeste travail.*

*Je dédie ce présent projet de fin d'études...*

*A vous très chers parents :*

*En témoignage de ma profonde gratitude pour tous les efforts et sacrifices que vous avez fait pour m'accorder une si bonne formation et si bonne culture non seulement durant mon cursus mais aussi depuis ma naissance. Je prie dieu de vous procurer la santé, le bonheur et la vie que vous méritez. Je vous aime.*

*A mes chers Frères*

*Pour tous les moments de joie et de complicité qu'on a partagés ensemble.*

*Pour une enfance qui dure toute la vie.*

*Vous saurez tracer votre chemin, je n'en doute pas.*

*Je vous aime.*

# Remerciements

*Au terme de mon projet de fin d'étude, Je tiens à exprimer mes profondes gratitudee et mes sincères remerciements envers tous ceux qui m'ont aidés et participé au bon déroulement de ce projet.*

*Je remercie M **Abderrahim BENABBOU**, Enseignant Responsable de mon projet pour l'expérience et l'aide qu'il m'a apporté lors de la réalisation de mon projet, pour son soutien, ses encouragements et surtout pour son orientation et son engagement.*

*Je tiens à remercier Mon tuteur de stage à CGI M<sup>r</sup> **Omar CHAMIMI** (Chef de projet Marsa maroc) pour tout le temps qu'il m'a consacré, ses directives précieuses, et pour la qualité de son suivi durant toute la période du stage.*

*Mes vifs remerciements s'adressent également aux membres du jury qui ont accepté d'évaluer ce travail.*

*Je tiens également à adresser mes plus sincères remerciements à l'ensemble du corps enseignant de la faculté des sciences technique FES, pour avoir porté un vif intérêt à notre formation, et pour avoir accordé le plus clair de leur temps, leur attention et leur énergie, et ce, dans un cadre généreusement agréable.*

*Enfin, je ne peux aussi oublier les membres de l'équipe Marsa Maroc qui ont largement contribué à l'ambiance conviviale durant ce stage .*

*Que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent l'expression de mes remerciements les plus chaleureux.*

*Que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent l'expression de mes remerciements les plus chaleureux.*

# Résumé

Le projet s'inscrit dans le cadre de la refonte du SI du Client Marsa Maroc. Ce système a une grande valeur pour Marsa Maroc et pour le pays lui-même, dans la mesure où ce projet sera déployé dans tous les terminaux du pays et favoriserait l'extension de l'activité de Marsa Maroc à d'autres terminaux à l'étranger.

Ce stage consiste à participer dans la réalisation d'une application de gestion portuaire qui s'inscrit dans le cadre de la refonte du système d'information de l'un des clients de CGI spécialisé dans l'exploitation des ports au Maroc.

Le déroulement du travail réalisé est le suivant :

- Recenser les besoins fonctionnels et non fonctionnels du projet.
- L'étude technique et la conception détaillée de l'application.
- Réalisation.

Pour mener à bien le développement de ce projet, la méthodologie agile semble plus adéquate à notre contexte, et plus précisément la méthode Agile combine avec Peritus (Morning meeting), avec le langage UML comme langage de modélisation de ce système.

Relativement au volet technique, il y avait eu recours aux nouvelles technologies et aux bonnes pratiques de la programmation, depuis Maven 3.2 qui nous définit la structure du projet, aux Framework & technologies : Spring, Hibernate/JPA, Flex, Cairngorm et MVC et JavaEE ... etc

Ma mission consiste à développer une partie de l'application pour la gestion du Cycle Camion sur deux parties : une partie back office développée avec la technologie javaEE et ses différents Framework : Hibernante/JPA, Spring et une partie front office avec Flex communiquant à l'aide des Web Services (GraniteDS) avec la partie back office.

# Abstract

The project is part of the rebuild of the Customer Marsa Maroc. This system has great value for Marsa Maroc and for the country itself, since this project will be deployed in all the terminals of the country and promote the extension of Marsa Maroc activity to other terminals in abroad.

This internship is to participate in the realization of a port management application that is part of the redesign of the information system of one of CGI's clients specializing in the operation of ports in Morocco.

The progress of the work is:

- Identify the functional and non functional project.
- The technical study and detailed design of the application.
- Realization.

To complete the development of this project, agile methodology seems most appropriate to our context, Peritus (Morning meeting), and the UML as modeling language of the system.

As the technical side, I used some new technologies and good programming practices, since Maven 3.2 defines us the project structure, the Framework & technologies: Spring, Hibernate/JPA, Flex, Cairngorm and JavaEE.

In a first time, we have begun by the design and development of the back office part using the javaEE technology and its different Frameworks: Hibernate/JPA, Spring. In a second part we are going to develop the front office section using Flex language (Adobe) in order to target a broad customer base and deploy our application.

The realization of this project has appealed to an iterative approach and incremental with a delivery of the back office part.

# Table des abréviations

Abréviation	Description
DAO	Data Access Object
DTO	Data Transfert Object
IOC	Inversion Of Control
JavaEE	Java Entreprise Edition
JDBC	Java Database Connectivity
JPA	Java Persistence API
MVC	Model View Controller
MXML	Macromedia Flex Markup Language
SFD	Spécifications Fonctionnelles Détaillées
ORM	Object Relational Mapping
SFG	Spécifications Fonctionnelles Generale
SVN	Subversion
UML	Unified Modeling Language
XML	eXtensible Markup Language
SSII	Société de Services en Ingénierie Informatique
API	Application Programming Interface

# Liste des Figures

FIGURE 1 : DOMAINE D'EXPERTISE CGI .....	13
FIGURE 2: REFERENCES INTERNATIONALES CGI MAROC.....	14
FIGURE 3: REFERENCES LOCALES.....	15
FIGURE 4: DIAGRAMME D'ACTIVITE DE VISITE MARITIME .....	18
FIGURE 5: OPERATIONS DANS UN TERMINAL A CONTENEUR.....	19
FIGURE 6: PROCESSUS IMPORT .....	20
FIGURE 7: PROCESSUS EXPORT .....	20
FIGURE 8 : DIFFERENTS CYCLE DE L'APPLICATION.....	23
FIGURE 9 : SCHEMA EXPLICATIF DU CYCLE CAMION .....	24
FIGURE 10 : PROCESSUS METIER DU CYCLE CAMION .....	25
FIGURE 11 : AVANTAGES ET INCONVENIENTS ENTRE LES CYCLES DE GESTION DE PROJETS .....	28
FIGURE 12 : DIAGRAMME DE GANT .....	29
FIGURE 13 : DIFFERENTS ACTEURS DANS UN PORT .....	33
FIGURE 14 : TERMINAL MOBILE WINDOWS CE .....	33
FIGURE 15 : DIAGRAMME DE CAS D'UTILISATION GLOBALE DE L'APPLICATION .....	35
FIGURE 16: DIAGRAMME D'ACTIVITE ORC.....	37
FIGURE 17: DIAGRAMME D'ACTIVITE RDV .....	38
FIGURE 18: DIAGRAMME D'ACTIVITE VC .....	38
FIGURE 19: DIAGRAMME D'ACTIVITE PC .....	39
FIGURE 20: DIAGRAMME DE PACKAGE DE CLASSE .....	39
FIGURE 21 : DIAGRAMME DE CLASSE D'ANALYSE GLOBAL DE L'APPLICATION .....	40
FIGURE 22 : DIAGRAMME DE CLASSE D'ANALYSE GLOBAL DE L'APPLICATION .....	41
FIGURE 23 : ARCHITECTURE GENERALE DE L'APPLICATION .....	43
FIGURE 24 : ARCHITECTURE GENERALE DE L'APPLICATION .....	44
FIGURE 25 : FONCTIONNEMENT FLEX .....	47
FIGURE 26 : L'ARCHITECTURE CAIRNGORMS MVC.....	50
FIGURE 27 : L'ARCHITECTURE MVC .....	51



# Table des matières

RESUME.....	4
ABSTRACT.....	5
INTRODUCTION GENERALE .....	11
PARTIE 1 : CONTEXTE GENERALE DU PROJET .....	12
CHAPITRE 1 : ORGANISE D'ACCUEIL .....	13
1. INTRODUCTION .....	13
2. PRESENTATION DE CGI .....	13
2.1. CGI DANS SA SITUATION INTERNATIONALE .....	13
2.2. CGI DANS SA SITUATION EUROPEENNE .....	14
2.3. CGI DANS SA SITUATION FRANÇAISE.....	14
2.4. L'ENTREPRISE PAR RAPPORT AU GROUPE.....	14
3. PRESENTATION DE CGI MAROC .....	15
4. PRESENTATION DU CLIENT MARSA MAROC.....	15
5. CONCLUSION.....	16
CHAPITRE 2 : CAHIER DE CHARGES .....	17
1. INTRODUCTION .....	17
2. PRESENTATION GENERALE DU PROJET .....	17
2.1. BESOINS GENERAUX DU CLIENT.....	17
2.1.1. LA GESTION DES VISITES MARITIMES ET CARGAISON .....	18
2.1.2. LA GESTION AUTOMATISEE DES TERMINAUX A CONTENEUR : .....	19
2.2. LES BESOINS FONCTIONNELS.....	20
2.2.1. GESTION DES VISITES MARITIMES ET CARGAISON.....	21
2.2.2. GESTION DES QUAIS .....	21
2.2.3. GESTION AUTOMATISEE DES TERMINAUX A CONTENEURS .....	21
2.2.4. CYCLE NAVIRE .....	21
2.3. BESOINS NON FONCTIONNELS.....	21
2.4. PERIMETRE DU PROJET .....	23
3. MODULE CYCLE CAMION .....	23
3.1. PRESENTATION DU PROJET.....	23
3.2. MISSIONS DU PROJET .....	24
3.2.1. VOLUMETRIE.....	25
3.2.2. TRAÇABILITE .....	25
3.3. LIVRABLES .....	25
4. CONCLUSION.....	26
CHAPITRE 3 : DEMARCHE DE PROJET : .....	27
1. INTRODUCTION .....	27
2. GESTION DE PROJET AGILE .....	27
3. AGILITE PRATIQUE .....	27
4. DIFFERENTES METHODES POUR DIFFERENTS CONTEXTES .....	28
5. PLANNING PREVISIONNEL ET REEL DU PROJET.....	28
6. EQUIPE DE REALISATION DU PROJET .....	29

7.	CONCLUSION.....	30
<b>PARTIE 2 : MISE EN OEUVRE DU PROJET .....</b>		<b>31</b>
<b>CHAPITRE 4 : ANALYSE.....</b>		<b>32</b>
1.	UML.....	32
1.1.	CHOIX D'UML .....	32
1.2.	LES BESOINS .....	32
1.3.	LA MAITRISE D'ŒUVRE.....	32
1.4.	IDENTIFICATION DES ACTEURS .....	32
1.5.	LE DIAGRAMME DE CAS D'UTILISATION (DCU).....	34
1.5.1.	DIAGRAMME D'ENCHAÎNEMENT DES ÉCRANS: ORC .....	36
1.5.2.	DIAGRAMME D'ENCHAÎNEMENT DES ÉCRANS: RDV .....	37
1.5.3.	DIAGRAMME D'ENCHAÎNEMENT DES ÉCRANS: VC.....	38
1.5.4.	DIAGRAMME D'ENCHAÎNEMENT DES ÉCRANS: PC.....	39
1.6.	LE DIAGRAMME DE CLASSE D'ANALYSE (DC) .....	39
1.6.1.	LES PACKAGES DU DIAGRAMME DE CLASSE .....	39
1.6.2.	DIAGRAMME DE CLASSE .....	40
1.6.3.	GENERATION DES TABLES DE LA BASE DE DONNEES .....	42
<b>CHAPITRE 5 : ETUDE TECHNIQUE DU PROJET .....</b>		<b>43</b>
1.	INTRODUCTION .....	43
2.	L'ARCHITECTURE GENERALE DE L'APPLICATION .....	43
2.1.	VUE D'ENSEMBLE DES DIFFERENTES COUCHES DE L'ARCHITECTURE .....	43
3.	L'ARCHITECTURE DETAILLEE DE L'APPLICATION.....	44
3.1.	LE PATTERN DTO.....	45
3.2.	DTO VS BEAN DU DOMAINE .....	45
3.3.	FRAMEWORK MAPSTRUCT .....	46
3.4.	FONCTIONNEMENT DE FLEX.....	46
3.5.	LA TECHNOLOGIE JEE.....	47
4.	TECHNOLOGIES & FRAMEWORK DE DEVELOPPEMENT .....	47
4.1.	JAVA.....	48
4.2.	L'ACTIONSCRIPT .....	48
4.3.	MXML.....	48
4.4.	XML .....	48
4.5.	HIBERNATE /JPA .....	48
4.6.	GRANITEDS .....	49
4.7.	SPRING IOC.....	49
4.8.	FLEX.....	49
4.9.	CAIRNGORM MVC.....	49
4.10.	MVC .....	50
4.11.	UML.....	51
5.	CONCLUSION.....	51
<b>CHAPITRE 6 : REALISATION .....</b>		<b>52</b>
1.	ENVIRONNEMENT DE DEVELOPPEMENT .....	52
1.	➤ ECLIPSE .....	52
2.	➤ APACHE TOMCAT.....	52
3.	➤ SQL SERVER.....	53
4.	➤ ADOBE FLASH BUILDER .....	53
5.	➤ MAVEN.....	54

6. ➤ SVN .....	54
2. REALISATION .....	55
2.1. FRUIT DU TRAVAIL .....	55
1. ECRAN CREER ENTETE ORDRE DE RESERVATION CONTENEURS «ENTREE PLEIN « .....	55
2. POP-UP CREER LIGNE RESERVATION CONTENEURS «ENTREE PLEIN » .....	56
3. AUTO COMPLETION CODE ISO .....	57
4. POP-UP LIST DES NAVIRES .....	57
5. POP-UP LIST VISITES MARITIMES .....	58
6. POP-UP LIST DES CLIENTS .....	58
7. POP-UP MODIFIER LIGNE RESERVATION CONTENEURS «ENTREE PLEIN » .....	60
8. SUPPRIMER LIGNE RESERVATION CONTENEURS «ENTREE PLEIN » .....	60
9. ECRAN CONSULTER ORDRE DE RESERVATION CONTENEURS «ENTREE PLEIN » .....	61
2.2. TEST UNITAIRE .....	62
2.2.1. TYPES DE TESTS UNITAIRES .....	62
3. CONCLUSION .....	63
CONCLUSION GENERALE .....	64
REFERENCES .....	65
ANNEXES .....	66
ANNEXE A : OUTILLAGES DETAILLES .....	66
FRAMEWORK SPRING .....	66
FRAMEWORK HIBERNATE .....	66
DEFINITION : .....	66
ARCHITECTURE : .....	67

# Introduction Générale

Le secteur portuaire contribue fortement à l'économie nationale du pays, et ce du fait que la quasi-totalité des échanges commerciaux du Maroc se font par voie maritime. Effectivement, les échanges maritimes s'inscrivent dans les méthodes de transport principales pour le commerce international, ce secteur très évolutif, connaît des changements perpétuels et continus.

Avec la mondialisation, le commerce mondial se développe et s'accroît en expansion et par conséquent, il était impératif pour le Maroc d'être compétitif dans ce secteur et de s'ouvrir sur de nouvelles opportunités.

Ainsi, les organismes spécialisés dans l'exploitation des ports au Maroc, conscients de l'importance majeure que détient leur commerce dans le développement économique du pays, s'adressent à des boîtes de conseil spécialisées afin de bénéficier de leurs services et ainsi de maintenir en éveil leur pointe technologique dans la gestion portuaire.

C'est dans ce sens, et dans le cadre de ma formation, que j'ai effectué mon stage PFE à CGI Maroc, leader mondial des services en technologie de l'information (TI) et en gestion des processus d'affaires.

Le contexte de ce projet consistant à développer des modules de gestion portuaires qui s'inscrivent dans le projet de refonte du système d'information d'une grande enseigne client de CGI Maroc qu'est MARSAMAROC.

Le présent rapport décline les différentes phases que j'ai suivies pour la réalisation du projet ainsi que les moyens auxquels j'ai eu recours pour atteindre les objectifs fixés. Le présent rapport est divisé en 2 parties :

- **1ere partie : CONTEXTE GENERALE DU PROJET**

- ❖ Dans le premier, je présenterai l'organisme d'accueil CGI et le client Marsa .
- ❖ La présentation de l'étude fonctionnelle du projet fera l'objet du deuxième chapitre.
- ❖ Le troisième chapitre concernera la démarche de projet en se basant sur les méthodes agiles et les morning meeting « peritus ».

- **2eme partie : MISE EN OEUVRE DU PROJET**

- ❖ Le quatrième chapitre, sera concerné par l'analyse et conception
- ❖ Le dernier chapitre abordera la réalisation des modules entrant dans le cadre de mon projet, ainsi que les tests unitaires. Des informations indispensables à la bonne compréhension du projet sont présentes en annexe.

# PARTIE 1 : CONTEXTE GENERALE DU PROJET

Cette partie introduit le contexte général dans lequel le projet s'intègre, en présentant à la fois son thème principal, la problématique générale et les objectifs du projet, ainsi que la conduite du projet.

# CHAPITRE 1 : Organise d'accueil

## 1. Introduction

Dans ce chapitre, je vais présenter dans la première partie : l'organisme d'accueil où j'ai effectué mon stage de projet de fin d'études, puis une présentation du client à qui est destiné ce projet et dans la deuxième partie l'étude générale du projet en décrivant la conduite et le pilotage du projet, les différentes étapes et la gestion des risques de ce dernier et sa qualité. Il présente aussi le découpage du projet ainsi que le planning de son déroulement au sein de CGI.

## 2. Présentation de CGI

Nous commencerons par donner une présentation sur la CGI, puis de CGI Maroc, où j'ai effectué mon stage, suivi d'une présentation sur le client Marsa Maroc à qui l'application de gestion des escales est destinée.

### 2.1. CGI dans sa situation internationale

Fondée il y a 37 ans, aujourd'hui, CGI compte 68000 professionnels présents dans plus de 400 bureaux répartis dans plus de 40 pays, un revenu annualisé de 10 Milliards de dollars.

Depuis 1986, CGI a adopté une stratégie de croissance axée sur l'acquisition d'entreprises offrant des services d'impartition (BST, AMS, Stanley Inc., Logica...). Après l'acquisition de la SSII européenne Logica en fin mai 2012, CGI devient un nouveau leader (5ème mondial) en services informatiques et d'entreprise. CGI est au service de plus de 10000 entreprises et organismes à travers le monde. Voici une figure illustrant quelques-uns d'entre eux suivant les différents domaines d'expertise.



Figure 1 : Domaine d'expertise CGI

## 2.2. CGI dans sa situation européenne

CGI fait son entrée en Europe après l'acquisition de Logica en 2012 pour 2,1 milliards d'euros. Cette opération permettra au groupe d'intégrer le club des dix plus grandes SSII mondiales avec un chiffre d'affaires de 10,4 milliards de dollars canadiens soit (8,2 milliards d'euros).

## 2.3. CGI dans sa situation française

C'est grâce à Logica que CGI a pu s'introduire en France. CGI France qui compte plus de 3000 collaborateurs, elle devient la 6ème plus importante SSII en France après cette acquisition avec un chiffre d'affaire de 1 milliard d'euros.

## 2.4. L'entreprise par rapport au groupe

CGI Maroc est une société reconnue pour son activité d'offshoring. C'est donc une société non résidente par opposition à CGI Canada dite Onshore qui est résidente.

Ainsi, les clients de CGI Maroc sont d'ordre local mais aussi non local à même que CGI Maroc offre actuellement des services à 50 clients dont 35 formés de grands noms français.

CGI Maroc est répartie sur deux grandes villes, 65% sur Casablanca la métropole où s'est déroulé mon stage, et 35% sur Rabat la capitale. Elle compte plus de 700 collaborateurs, offrant des services complets en IT, dont les services-conseils en management ainsi que des services d'intégration de systèmes en gestion des processus d'affaires et d'impartition (externalisation).



Figure 2: Références Internationales CGI Maroc

Figure 3: Références locales



### 3. Présentation de CGI MAROC

CGI Maroc est répartie sur deux grandes villes, 65% sur Casablanca la métropole et 35% sur Rabat la capitale et c'est à Casablanca où s'est déroulé mon stage.

CGI Maroc compte plus de 700 collaborateurs, offrant des services complets en IT, dont les services-conseils en management ainsi que des services d'intégration de systèmes, en gestion des processus d'affaires et d'impartition (externalisation).

CGI est le premier fournisseur francophone de services IT au Maroc, ce qui lui permet de collaborer étroitement avec plusieurs clients français et de leur offrir un savoir-faire sectoriel tant sur le plan des technologies que des affaires.

CGI Maroc offre actuellement des services à 50 clients dont 35 formés de grands noms français.

CGI au Maroc possède une double certification : ISO 9001 et ISO 27001 et des services liés aux technologies SAP, Java JEE, Mainframe, Oracle et Microsoft, ainsi que des systèmes de gestion de la relation client (CRM) et des RH.

### 4. Présentation du client Marsa Maroc

Marsa Maroc, société d'exploitation des ports, créée conformément au décret n°284844 du premier janvier 2007 pris pour application de la loi 15-02 promulguée par le dahir 184194 du 05 janvier 2007 portant création de la SODEP au nom commercial Marsa Maroc.



La société d'exploitation des ports est une société anonyme à conseil de surveillance et directoire, et à capital public. Son métier est l'exploitation de terminaux portuaires, à travers l'exercice des services aux navires et des services aux marchandises.

Elle intervient comme exploitant commercial dans les ports de Nador, Tanger, Mohammedia, Casablanca, JorfLasfar, Safi, Agadir, Laâyoune et Dakhla Elle est chargée d'assurer pour le compte de l'état selon l'article 4 de la loi 684 portant créations de la SODEP les opérations essentielles suivantes :

- Le pilotage, le remorquage, le lamage et le ravitaillement des navires
- L'aconage, la manutention et le magasinage des marchandises
- Le ravitaillement en eau douce des usagers et la gestion du réseau électrique
- La maintenance des terres pleines, des voiries, des voies ferrées, des réseaux d'assainissement ainsi que la maintenance des voies terrestres
- La construction et la maintenance des magasins et des bâtiments
- La gestion des gares maritimes et leurs annexes
- La location des défenses de quai
- Le transbordement des marchandises de navire à navire
- Les locations diverses de matériel, les locaux, etc.
- Chargement et déchargement des wagons et camions.

## 5. Conclusion

A travers ce chapitre, on a présenté le groupe CGI avec le client Marsa Maroc, son historique et son savoir-faire et leurs gestions dans les entreprises selon leurs natures.

# CHAPITRE 2 : Cahier de charges

## 1. Introduction

Ce chapitre comporte une présentation du besoin, description sur le projet et le cahier des charges. Il inclut la planification employée dans le but de se mettre dans le contexte du projet pour le bon déroulement de sa réalisation.

## 2. Présentation Générale du projet

### 2.1. Besoins Généraux du client

Le projet de refonte du système d'information métier de Marsa Maroc s'inscrit donc dans la continuité de ces actions d'améliorations. Il fait suite à une étude du SI réalisé en 2011 et qui a permis de définir une macro trajectoire de cette refonte.

La refonte du système d'information métier vise à disposer d'une offre SI Métier maîtrisée et à forte valeur ajoutée à même de :

- Assurer la pérennité de l'exploitation portuaire en s'affranchissant des offres tierces,
- Favoriser l'extension de l'activité de Marsa Maroc à d'autres terminaux à l'étranger,
- Piloter de façon adaptative le développement et l'évolution du futur SI métier de l'entreprise.

Marsa Maroc a défini 2 volets majeurs de fonctionnalités incluses dans le périmètre de la refonte de son système d'information métier.

Chaque volet sera sous-traité à un prestataire de services de développement spécifique :

- volet 1 : refonte de la gestion des escales et du système automatisé de la gestion du terminal à conteneur
- volet 2 : refonte du TOS polyvalent et du système de la gestion commerciale et de la facturation.

**Le présent PROJET\_a pour objet la réalisation du volet 1.**

Si on peut résumer tous le projet on peut le résumer comme suivant :

### 2.1.1. La gestion des visites maritimes et cargaison

Qui se décompose en :

- ❖ La gestion des visites maritimes,
- ❖ La gestion des cargaisons
- ❖ La gestion des quais

Le diagramme ci-après présente les points clés du processus :

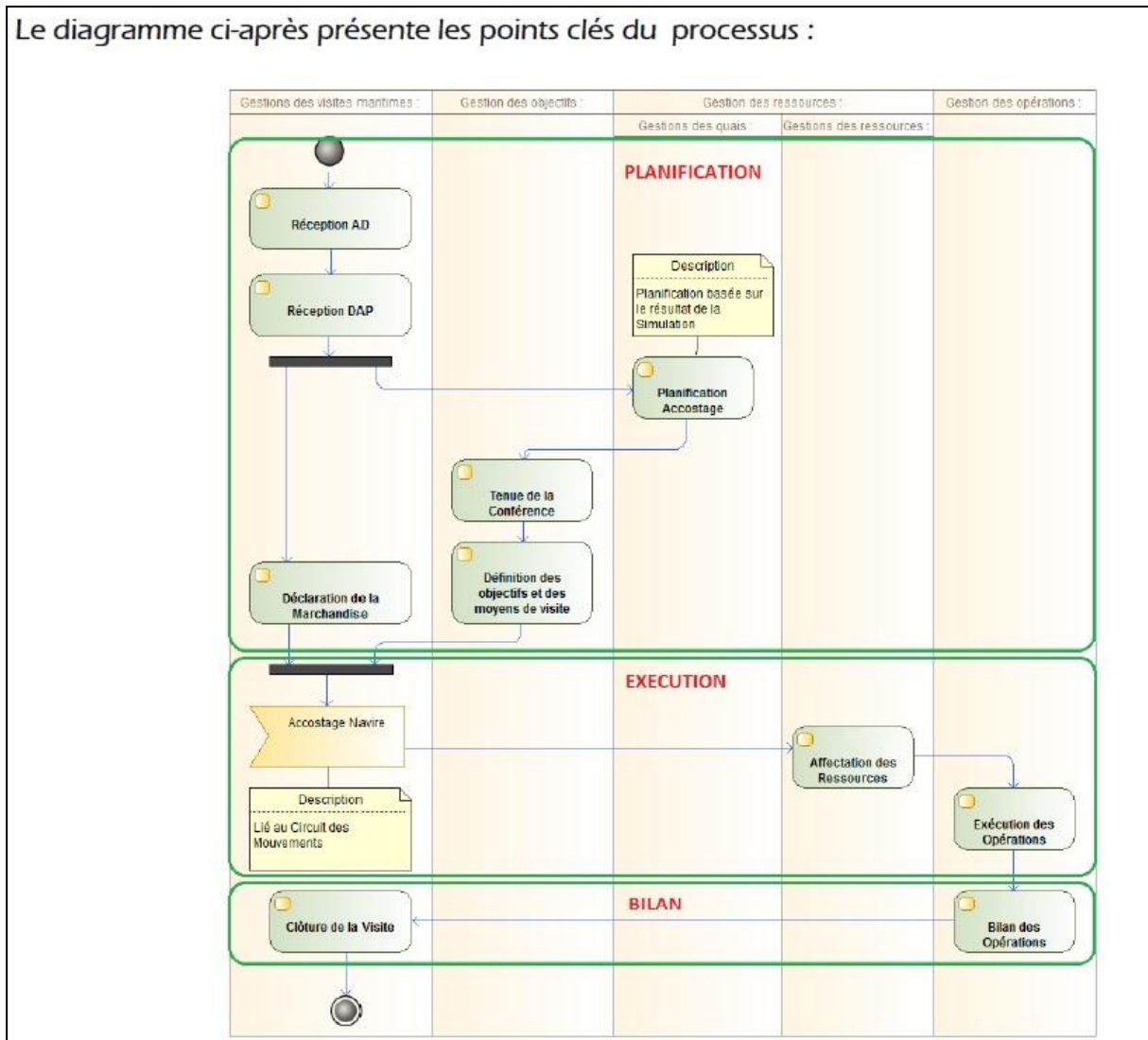


Figure 4: Diagramme d'activité de visite maritime

Le projet de refonte du SI Marsa Maroc couvre toutes les opérations afférentes à la visite maritime depuis l'annonce de son arrivée par le consignataire jusqu'à ce qu'il quitte le terminal.

Avant l'arrivée du navire (quelques jours à quelques semaines), l'agent maritime envoie un **Avis D'arrivée(AD)** pour annoncer l'arrivée de son navire renseignant **la date estimée d'arrivée(ETA)**, le type des opérations qu'il traitera(Import, Export, Transbordement, Cabotage), les provenances, destinations et éventuellement les Co chargeurs(Consignataires en Pool) qui l'accompagneront.

Quelques jours avant l'arrivée de son navire, le consignataire confirmera l'arrivée par l'envoi d'une **Demande d'Attribution de Poste (DAP)** renseignant des informations complémentaires sur le navire (Tirant d'eau, TJB,..) et la cargaison (récapitulatif).

Une **LMD (Liste des Marchandises Dangereuses)** ou **LMS (Liste des Marchandises Spéciales)** accompagnera la DAP décrivant les marchandises dangereuses ou spéciales (Défense nationale) et les équipements les transportant.

Après acceptation de la DAP par Marsa Maroc et notification du consignataire et Co-chargeurs, ces derniers envoient leur manifeste.

Le manifeste englobe tout le détail nécessaire sur la cargaison transportée par le navire (Provenance et destination, Equipements et nature, tonnage, type de marchandise, classe IMDG marchandise, type d'opération).

Les intrants précédents s'accumulant à des normes théoriques de traitement des navires (nombre de boîtes/main/heure, Tonnage/type marchandise/main/shift) devront permettre de faire une simulation d'accostage des navires afin d'avoir une situation réelle et de même faire une projection future de la situation du quai.

### 2.1.2. La gestion automatisée des terminaux à conteneur :

Qui se décompose en :

- Cycle navire
- La gestion des terres d'entreposage ou Cycle Parc
- La gestion des opérations spéciales
- Le pointage

Le TOS automatise l'ensemble des opérations effectuées dans un terminal à conteneur : Entrées/sortie, quai et zones de stockage. Les processus d'import et d'export sont la base des opérations dans le terminal à conteneur

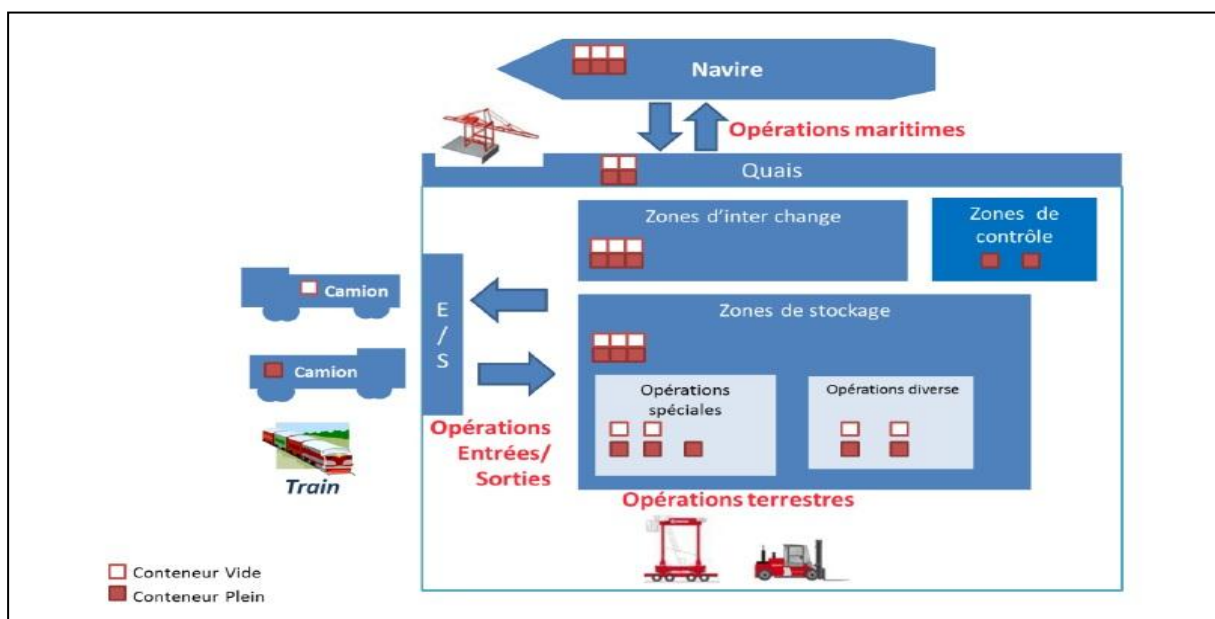


Figure 5: Opérations dans un terminal à conteneur

### 2.1.2.1. Processus d'import et d'export

Les processus d'import et d'export sont la base des opérations dans le terminal à conteneur. Ils déterminent le déroulement des opérations.

- Processus Import:
  - 1- Arrivée du navire au quai (cycle navire)
  - 2- Déchargement des conteneurs à importer (cycle navire)
  - 3- Stockage sur le terminal (cycle parc)
  - 4- Livraison de conteneurs (visite terrestre)

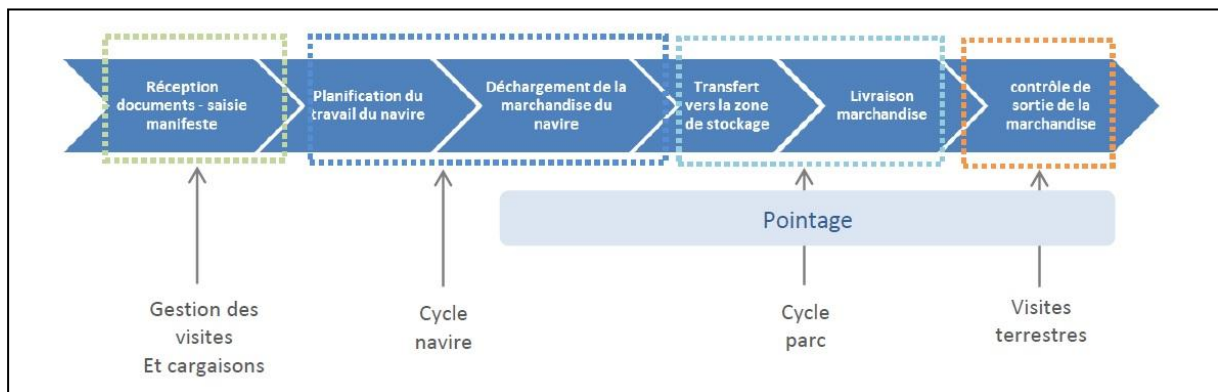


Figure 6: Processus import

- Processus Export :
  - 1- Arrivé du camion/train venant livrer le conteneur à exporter (Visite terrestre)
  - 2- Stockage sur le terminal (Cycle parc)
  - 3- Chargement du conteneur sur le navire (Cycle Navire).

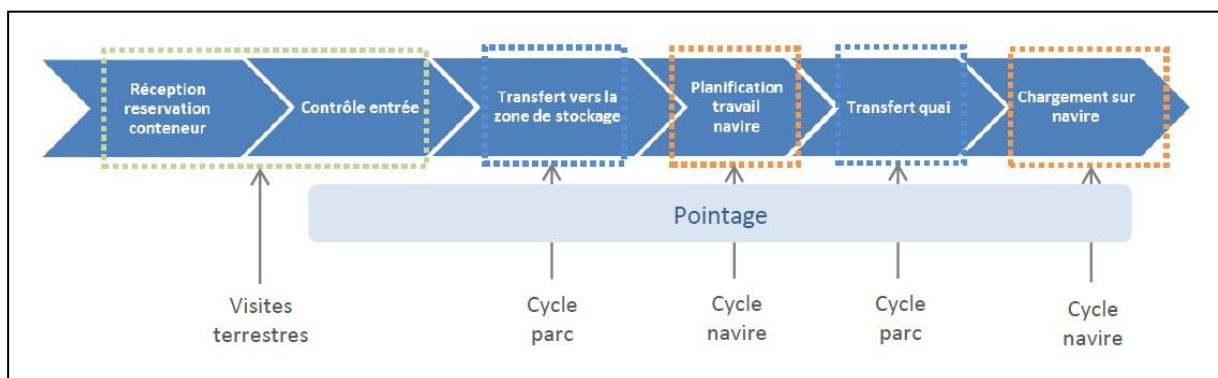


Figure 7: Processus Export

## 2.2. Les besoins fonctionnels

Le futur système doit permettre à l'utilisateur de :

### 2.2.1. Gestion des visites maritimes et cargaison

La gestion des visites maritimes et cargaison est un module transversal aux modules de gestion des terminaux de Marsa Maroc. Elle couvre en particulier :

- ❖ La gestion des réservations des visites navires en prévision de leurs arrivées
- ❖ La gestion des arrivées navires
- ❖ La gestion des mouvements navires
- ❖ La gestion des cargaisons qui comprend :
  - Déclaration des conteneurs
  - Déclaration des remorques
  - Déclaration des marchandises
  - Clôture du manifeste

### 2.2.2. Gestion des Quais

Il s'agit de la gestion des affectations des ressources (quais, postes, mains) aux visites planifiées pour une période donnée (contexte de simulation) et comprend :

1. Définir un contexte de simulation
2. Définir les hypothèses pour chaque visite
3. Simuler un scénario d'accostage
4. Valider un scénario d'accostage
- 5.

### 2.2.3. Gestion automatisée des terminaux à conteneurs

La gestion automatisée des terminaux à conteneurs couvre :

1. La gestion du parc : planification des opérations, gestion de la manutention parc, suivi des réalisations.
2. La gestion des ressources

### 2.2.4. Cycle navire

Planification des opérations de chargement/déchargement, gestion des opérations de chargement/déchargement, bilan d'exécution

1. La gestion des visites terrestres (camion et train) :
2. Gestion des réservations
3. Gestion des opérations spéciales
4. Gestion des pointages conteneurs

## 2.3. Besoins non fonctionnels

Exigence	Critère	Description
<b>Maintenabilité</b>	Lisibilité	Le code et la documentation doivent être compréhensibles par simple lecture, notamment en respectant les règles de gestion documentaire et les normes de développement
	Modularité	La décomposition fonctionnelle de l'application est réalisée par Marsa Maroc.
	Cohérence	Chaque version applicative doit disposer de la documentation appropriée et maintenue à jour.
	Traçabilité	Toute modification apportée au code ou à la documentation doit être tracée et expliquée.
	Adaptabilité	L'ajout, modification, correction ou suppression de fonctionnalités doivent être facilement réalisables.
<b>Exploitabilité</b>	Efficacité	Les impacts en termes de performance doivent être prises-en compte lors des développements. La consommation des ressources (CPU, mémoire, etc.) doit être minimisée.
	Portabilité	Le changement d'environnement d'exécution doit pouvoir se faire facilement, notamment pour intégrer les besoins d'exécution sur les terminaux mobiles.
<b>Capacité fonctionnelle et convivialité</b>	Aptitude	Les composants développés doivent respecter les spécifications fournies par Marsa Maroc.
	Complétude	Les composants développés doivent couvrir tout le périmètre des spécifications sans omission.
	Ergonomie	Le système doit être facilement utilisable et disposer d'interfaces conviviales, notamment par le respect des règles d'ergonomie de Marsa Maroc
	Multilinguisme	Le système doit permettre de gérer le multilinguisme (français-anglais)
<b>Sécurité</b>	Intégrité du système	Les composants du système ne doivent subir aucune altération ou perte. Les dispositions nécessaires à leur protection doivent être mises en œuvre. Les sauvegardes de l'environnement de développement de Marsa Maroc est sous la responsabilité de cette dernière.
	Intégrité des données	Les données ne doivent subir aucune altération ou perte lors de leur manipulation.
	Propriété intellectuelle	Les installations de l'application réalisée doivent être sécurisées de façon à interdire son utilisation à l'extérieur de Marsa Maroc.
	<b>Auditabilité</b>	Le système doit pouvoir restituer l'ensemble des opérations ayant eu lieu et leur auteur

## 2.4. Périmètre du projet

Mon projet de fin d'études s'articule autour de l'étude, l'analyse et le développement de modules applicatifs figurant dans le cycle 3 et spécialement dans le cycle Camion, ci-dessus un schéma représentative des cycles appartenant à mon projet qui concerne la refonte du SI métier de Marsa Maroc .

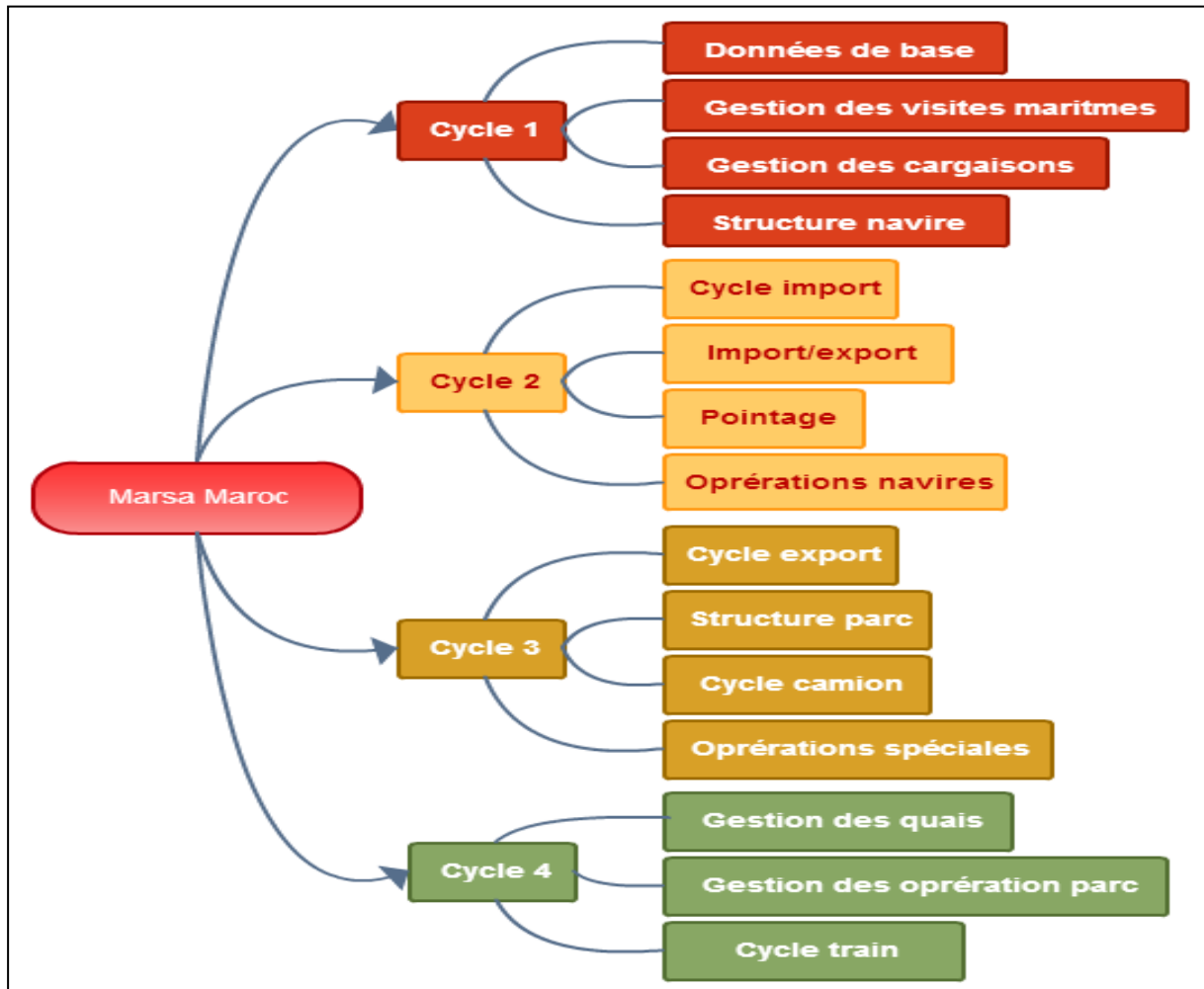


Figure 8 : Différents cycle de l'application

## 3. Module Cycle Camion

### 3.1. Présentation du projet

Le cycle Camion est défini comme étant l'ensemble des faits générés depuis l'annonce de l'arrivée du camion au terminal jusqu'à sa sortie physique de ce dernier à savoir :

- Un Ordre de service de livraison ou d'évacuation (ordre de réservation ORC) du conteneur (Booking) émane du client et enregistré par le terminal avant l'arrivée réelle du camion



- Une fois le camion arrivé, un contrôle administratif est opéré à la porte du terminal
- Un contrôle physique est opéré avant que l'on dirige le camion à la zone d'inter change
- Des instructions de manipulation du Conteneur sont générées et exécutées par l'engin de manutention
- La sortie du camion est réalisée

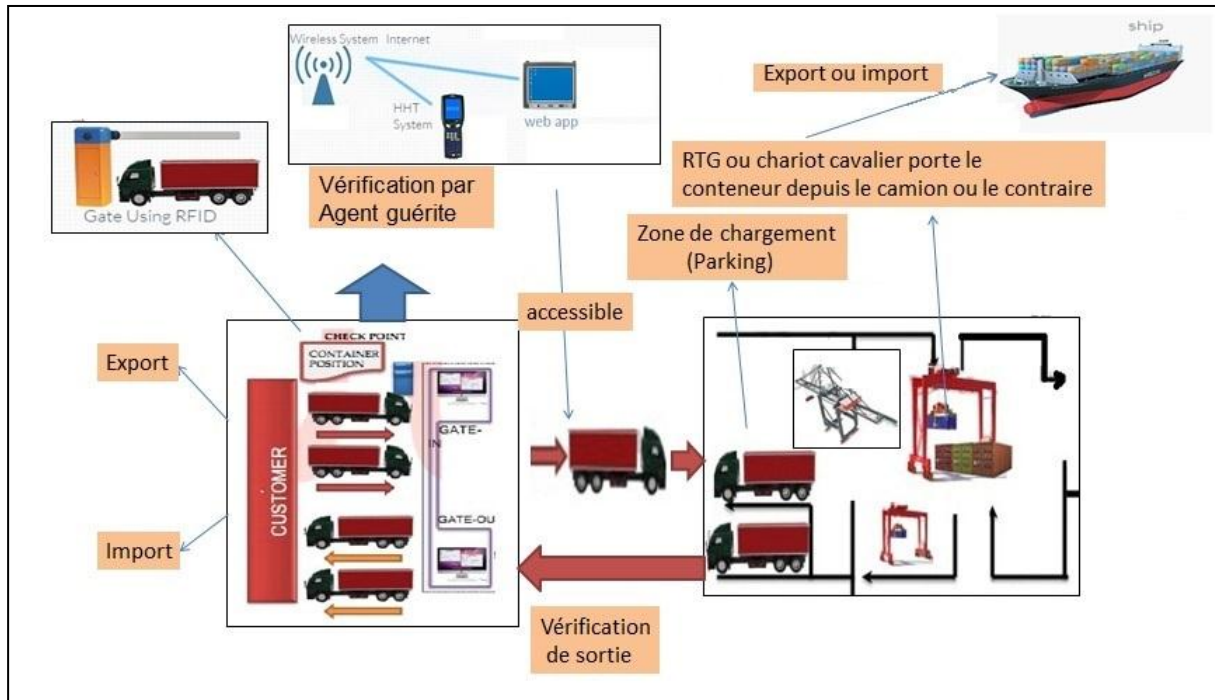


Figure 9 : Schéma explicatif du cycle camion

### 3.2. Missions du projet

Le but de cette partie de mon projet de fin d'étude est de traiter les quatre sous modules fonctionnels suivant :

- Gestion des ordres de réservation Conteneurs
- Gestion des rendez-vous camions
- Gestion des visites camion
- Pointage camion

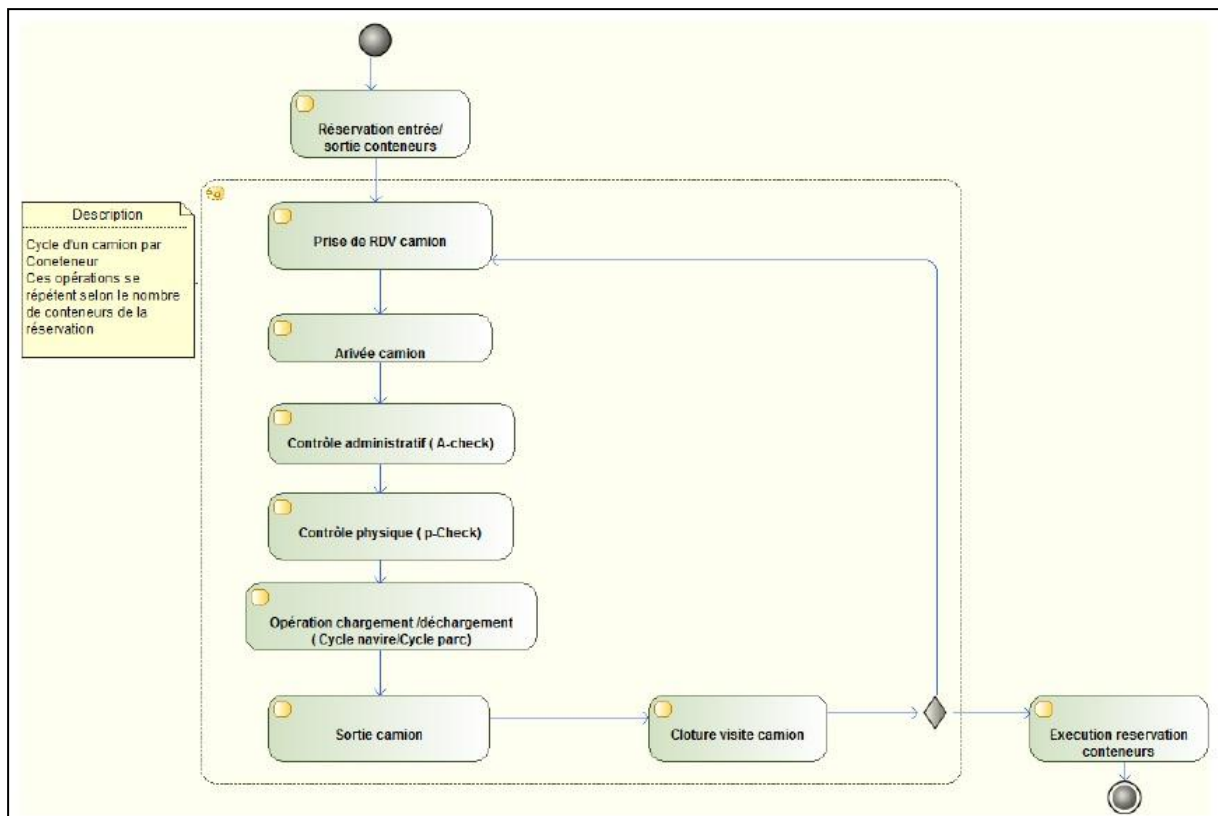


Figure 10 : Processus métier du cycle camion

### 3.2.1. Volumétrie

Ce module est utilisé par une vingtaine d'agents. Le nombre moyen estimé des visites camion par jour est entre 500 et 1000.

### 3.2.2. Traçabilité

Dans toutes les opérations d'ajout et de modification des données, les données à tracer sont les suivantes :

- Date de création
- Date de modification
- Agent ayant effectué la création
- Dernier agent ayant effectué la modification

### 3.3. Livrables

<b>Phase</b>	<b>Livrable</b>	<b>Responsable</b>
<b>Etude des besoins</b>	<i>Cahier des charges</i>	Développeur du projet
<b>Analyse et Conception</b>	<i>Diagrammes UML</i>	<i>Développeur du projet</i>
<b>Développement</b>	<i>Application web</i>	Développeur du projet

<b>Testes et recette</b>	<i>Maintenance du code</i>	Développeur du projet
<b>Documentation</b>	<i>Rapport du projet</i>	Développeur du projet

## 4. Conclusion

A la fin de ce chapitre, on a étudié le besoin du client en présentant l'ensemble des fonctionnalités du futur système, puis on a présente les différents cycles de l'application.

# CHAPITRE 3 : Démarche de projet :

## 1. Introduction

Le diagramme de Gantt repose sur une idée simple. Il s'agit simplement de faire figurer sur un document les différentes phases qui se succèdent dans la réalisation d'un projet. Le diagramme va permettre de voir quand commence une tâche, et quand elle se termine et quelle est sa durée.

## 2. Gestion de projet Agile

Les méthodes agiles visent à fluidifier la relation entre utilisateurs et informaticiens. Elles ont souvent été mises en oeuvre de façon dogmatique. Désormais, la réussite passe par certaines précautions et compromis. Lorsqu'il s'agit de développer une application, de paramétrer un progiciel ou de monter un site web un tant soit peu complexe, les interactions entre informaticiens et utilisateurs sont souvent difficiles à organiser.

La démarche traditionnelle consiste à demander aux utilisateurs de réaliser un cahier des charges qui permet aux développeurs de se mettre au travail. Ces développeurs livrent le résultat plusieurs mois plus tard et sont alors confrontés à la réaction d'utilisateurs estimant avoir été mal compris ou dont les besoins ont évolué. C'est pour combattre cette relation rigide que sont nées les méthodes dites agiles, comme Scrum (la plus connue), EXtreme Programming ou Unified Process. Elles préconisent une relation permanente avec les utilisateurs, une livraison régulière de versions de l'application, ainsi qu'un mode de travail horizontal, avec une responsabilisation de chaque développeur.

Mais ces méthodes ont souvent été mises en oeuvre de façon dogmatique, au point d'effrayer aussi bien les utilisateurs que les développeurs. Désormais, la réussite passe par certaines précautions et compromis.

## 3. Agilité pratique

- Selon Jean-Pierre Vickoff dans « Puma essential » :  
*« L'Agilité est la dynamique de l'organisation réactive et apprenante. »*

- Selon Véronique Messenger-Rota dans « Gestion de projet Agile » :  
*« Une méthode Agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif, avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins clients. »*

## 4. Différentes méthodes pour différents contextes

Une fois exposées les généralités des méthodes Agiles, il existe une douzaine de méthodes pour faire de l'Agilité. Chacune est née à partir des échecs de réalisation de projets. En 2011, un des signataires du manifeste publie un livre blanc sur ces différentes méthodes : <http://www.agilebok.org>. Dans la même lignée que le célèbre PMBOK7 édité par PMI, ce recueil définit les bonnes pratiques de ces différentes approches.

En se basant sur ce référentiel et ces différentes sources et bibliographies sur les méthodes Agiles, on a dressé dans le tableau suivant leurs types et champs d'action :

Méthode	Description	En savoir plus
<b>Scrum</b>	Terme anglais désignant la mêlée au rugby, Scrum casse la rigidité des méthodes de gestion de projets traditionnelles en adoptant une démarche itérative et adaptative.	<a href="http://www.timwi.com">www.timwi.com</a>
<b>Extreme Programming (ou XP)</b>	XP reprend la démarche itérative de Scrum mais prône des « valeurs » (respect, courage, simplicité...) et vise des petites équipes et des besoins très évolutifs. Cette méthode se démarque aussi en abordant les aspects test et intégration.	<a href="http://lagace.developpez.com">http://lagace.developpez.com</a>
<b>Unified Process (UP)</b>	UP est aussi une méthode itérative et incrémentale mais elle cible plus particulièrement la réutilisation de composants.	<a href="ftp://ftp-developpez.com/sabricole/UP.pdf">ftp://ftp-developpez.com/sabricole/UP.pdf</a>

*Figure 11 : Avantages et Inconvénients entre les cycles de gestion de projets*

Dans notre projet on s'est basé sur plusieurs méthodes mais pas une particulièrement, mais on a fait la bonne pratique à savoir la gestion de Backlog , morning meeting ...etc .

## 5. Planning prévisionnel et réel du projet

Ci-dessus une illustration de diagramme de Gantt :

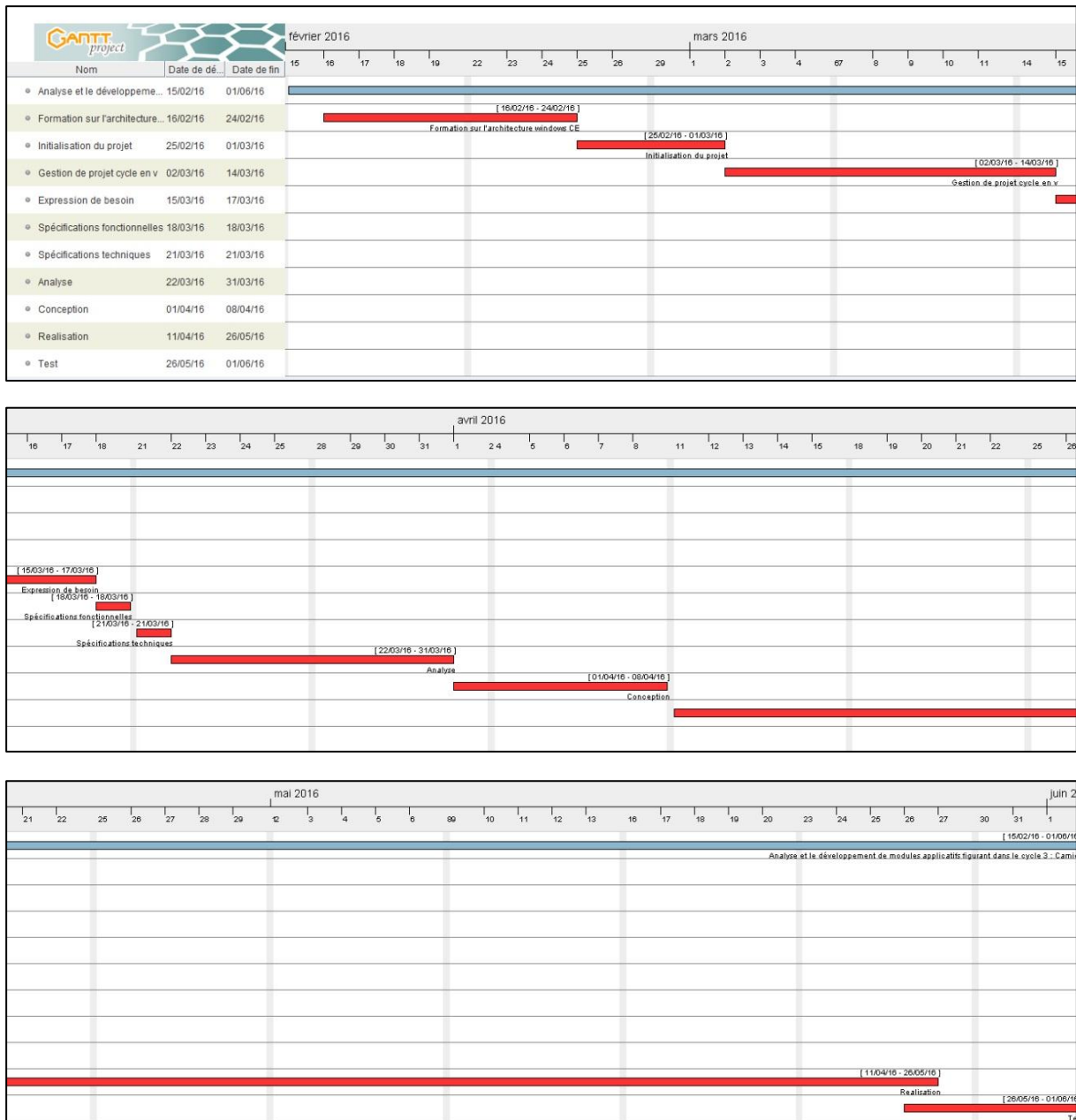


Figure 12 : Diagramme de Gant

## 6. Equipe de réalisation du projet

Nom et Prénom	Fonction/Rôle dans le projet
Oubelque Khalid	Stagiaire / Développeur
Chamimi Omar	Chef du projet

### Développeur :

- élaboration du dossier de gestion de projet

- réalisation de la spécification détaillée
- codage de l'application
- effectuation des tests unitaires

#### **Chef du projet :**

- valide le dossier des spécifications fonctionnelles
- valide le codage
- présentation des besoins fonctionnels du projet
- valide les livrables
- contrôle le respect des demandes

## **7. Conclusion**

Dans ce chapitre on a organisé notre projet on utilisant des points importantes dans l'agilité à savoir les 'Morning meeting', gestion de Backlog ...etc , et en faisant un planning bien détaillé.

## Partie 2 : MISE EN OEUVRE DU PROJET

Dans cette partie, on va mettre en place notre projet en passant de quatre étapes majeures : analyse, conception et réalisation et test unitaires.

Dans la phase d'analyse on modéliser les besoins du client, par la suite on va montrer comment on va réaliser ces besoins dans la phase de conception et à la fin on va présenter les outils de réalisation du projet qui appartiennent au monde JAVAEE et ADOBE.



# CHAPITRE 4 : ANALYSE

## 1. UML

### 1.1. Choix d'UML



UML (Unified Modeling Language, que l'on peut traduire par langage de modélisation unifié) est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

### 1.2. Les besoins

Concevoir une application est un processus qui peut être relativement complexe. En effet, lors de l'élaboration d'un projet, de nombreuses données, concept, processus métiers, acteurs et autres se mélangent et peuvent très vite donner le tournis. Afin d'y voir plus clair, on peut regrouper ces différents termes par familles et les organiser de façon à former un ensemble cohérent, fonctionnel et conforme aux parties en présence.

### 1.3. La maîtrise d'œuvre

Pour les personnes qui conçoivent et réalisent un logiciel, maîtriser ce qu'il faut faire, ce que l'on a déjà fait et ce qu'il reste à faire est primordial. Cela regroupe la conception, la communication ainsi que la planification. La modélisation UML fournit un formalisme relativement simple à appréhender mais qui s'avère très puissant à l'utilisation. Il sert notamment de support à la réflexion en permettant de modéliser les interactions entre les différents utilisateurs et la solution logicielle. La plupart des documentations techniques et fonctionnelles peuvent ainsi voir leur lisibilité grandement améliorée grâce aux différents types de diagrammes supportés par UML.

### 1.4. Identification des acteurs

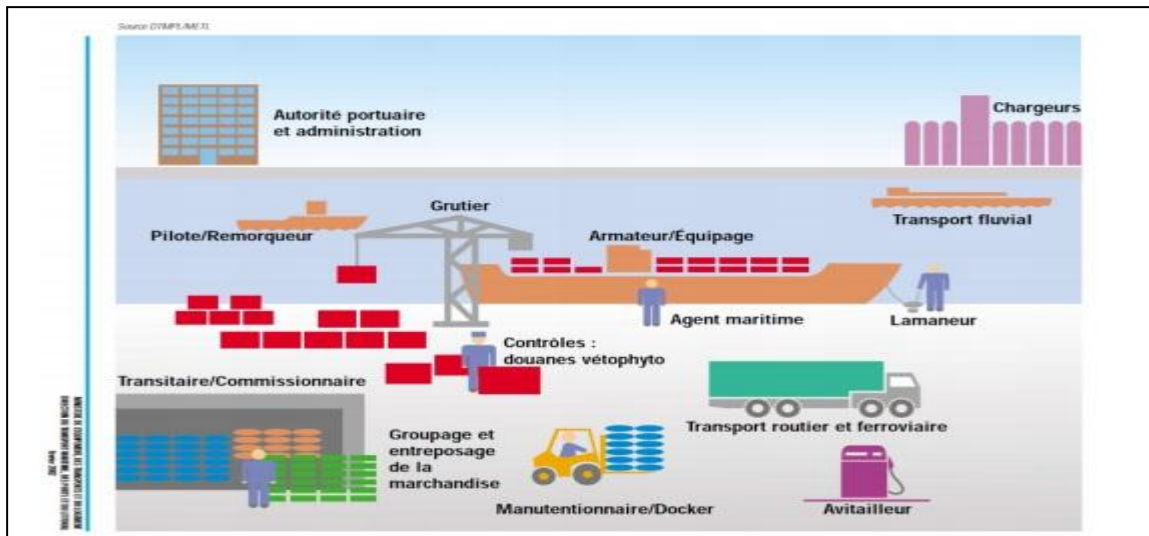


Figure 13 : Différents acteurs dans un port

Nous avons identifié principalement ces utilisateurs suivant leurs accès au système.




- ❖ **Client RIA** (Le terme « RIA » désigne les applications riches qui s'exécutent au sein de navigateurs Internet, tout comme les applications Web classiques.)
- ❖ **Client RDA** (Le terme « RDA » désigne les applications riches qui s'exécutent directement sur le poste client des utilisateurs, comme les applications bureautiques classiques. Elles ne fonctionnent donc pas au sein de navigateurs Internet comme les RIA.)
- ❖ **Client Mobile** (Les terminaux mobiles de Marsa Maroc sont dotés du système Windows CE (6.0). Pour des raisons de compatibilité, nous avons choisis C#/.NET comme plateforme de développement de tous les modules applicatifs destinés aux terminaux mobiles.)



Figure 14 : Terminal mobile Windows CE

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données. Dans le cas de notre application web les acteurs principaux sont :

### Direct :

TYPE D'ACTEUR	DESCRIPTION FONCTIONNELLE
 Agent de marsa maroc	Ce type d'acteur correspond à l'agent de marsa et aura les fonctionnalités suivantes : <ul style="list-style-type: none"><li>• Effectue les ordres de réservation des conteneurs</li><li>• La gestion des visites camion</li></ul>
 Responsable RDV camion	Ce type d'acteur aura les fonctionnalités suivantes : <ul style="list-style-type: none"><li>• Supervision et validation de toutes les prises de RDV Camion</li></ul>
 Pointeur	Ce type d'acteur aura les fonctionnalités suivantes : <ul style="list-style-type: none"><li>• Pointage des conteneurs lors de la visite camion</li><li>• Pointage des plaques camion lors de la visite</li></ul>

### Indirect :

- **Transporteur** : est un client premièrement, c'est une entreprise de transport logistique dans le client qui veut transporter ca marchandises (conteneurs) que ça soit Export ou bien import en passant par Marsa maroc , se sert de ces entreprise de transport pour faire le travail .
- **Agent vérificateur ou agent guérite** : responsable de l'A-check et de contrôle administratif , est opéré à la porte du terminal
- **L'agent maritime** : appelé aussi consignataire, est mandaté par l'armateur du navire afin de le représenter dans un port lors d'une escale.  
L'agent maritime assure tous les besoins du navire (paquebot passagers, navire de commerce, bâtiment militaire) à son arrivée, durant son séjour au port et lors de son départ.

## 1.5. Le Diagramme de Cas d'Utilisation (DCU)

Dans les figures qui suivent, nous présenteront les cas d'utilisation qui mettent en évidence les principales fonctionnalités de chaque acteur dans le système ainsi que un tableau descriptif de chaque cas dont il y a un scénario, alors que chaque scénario est modélisé par un diagramme de scénario séquence.

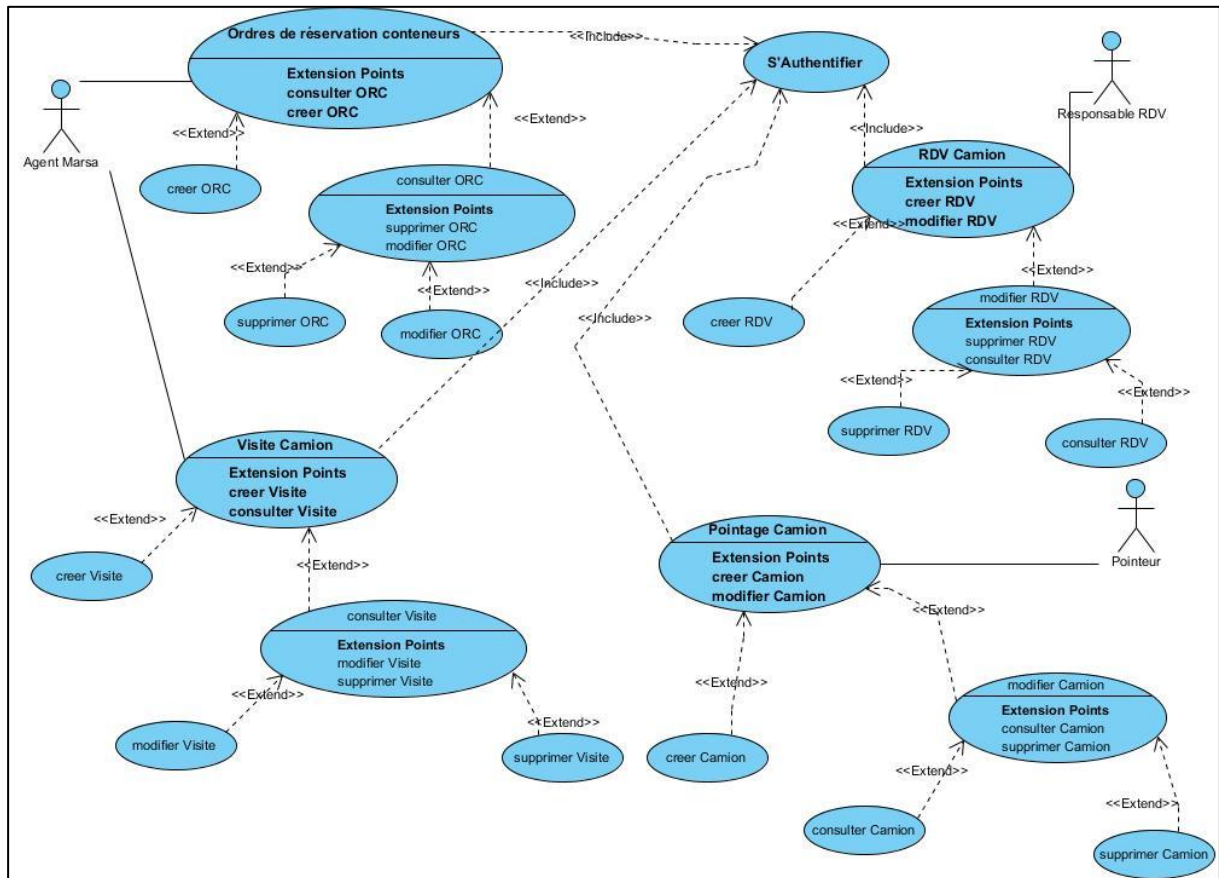


Figure 15 : Diagramme de cas d'utilisation globale de l'application

### Ordres de réservation conteneurs

<b>Cas d'utilisation :</b>	<b>ORC</b>
<b>Titre</b>	Gestion des Ordres de réservation conteneurs
<b>But</b>	gère l'ordre de réservation des conteneurs
<b>Acteur principal</b>	Agent de marsa
<b>Pré condition</b>	Agent marsa maroc doit s'authentifier
<b>Description</b>	Avant l'arrivée réelle du camion, un ordre de réservation de livraison ou d'enlèvement du conteneur du terminal est enregistré. Cet ordre émane du client. Il peut concerner un ou plusieurs conteneurs pleins.

### Rendez-vous Camion

<b>Cas d'utilisation :</b>	<b>RDV</b>
<b>Titre</b>	Gestion RDV Camion
<b>But</b>	gère les RDV camions

<b>Acteur principal</b>	Responsable RDV
<b>Pré condition</b>	Gestionnaire de prise de RDV camion doit s'authentifier
<b>Description</b>	Ce module gère les RDV camions créés pour livrer ou récupérer un conteneur du terminal

### Visite Camion

<b>Cas d'utilisation :</b>	Visite camion
<b>Titre</b>	Gestion visites camions
<b>But</b>	gère les visites camions au terminal
<b>Acteur principal</b>	Agent de marsa maroc
<b>Pré condition</b>	Agent doit s'authentifier
<b>Description</b>	Une fois le camion transportant le conteneur vide arrive à la porte du terminal, l'agent vérificateur procède aux vérifications administratives pour autoriser l'entrée du camion. Il vérifie si chaque conteneur a un ordre de réservation enregistré ou actif sur la base du document présenté par le chauffeur du camion.

### Pointage Camion

<b>Cas d'utilisation :</b>	Pointage camion
<b>Titre</b>	Gestion Pointage camion
<b>But</b>	gère le pointage physique des camions
<b>Acteur principal</b>	Pointeur
<b>Pré condition</b>	Pointeur doit s'authentifier
<b>Description</b>	Ce module gère le pointage physique des camions.

#### 1.5.1. Diagramme d'enchaînement des écrans: ORC

Avant l'arrivée réelle du camion, un ordre de réservation de livraison ou d'enlèvement du conteneur du terminal est enregistré. Cet ordre émane du client. Il peut concerner un ou plusieurs conteneurs pleins.

Un ordre de réservation a les statuts suivants :

- Enregistré : à la création,
- Activé : à l'exécution de la première visite camion,

- Exécuté : à l'exécution de toutes les visites camion,
- annulé

Ce module fonctionnel gère l'ordre de réservation des conteneurs, 5 cas d'utilisations font partie de ce module fonctionnel :

- Créer ordre de réservation conteneurs
- Modifier ordre de réservation conteneurs
- Consulter ordre de réservation conteneurs
- Supprimer ordre de réservation conteneurs

Dans tout ce qui suit, nous utiliserons le terme **ORC** pour **Ordre de Réservation des Conteneurs**.

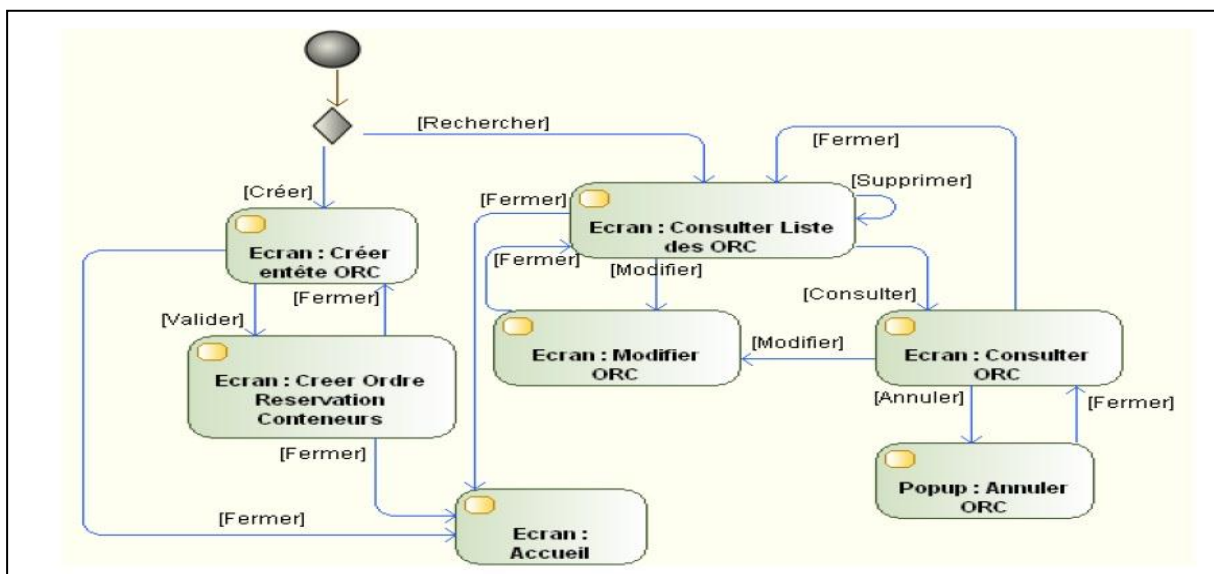


Figure 16: Diagramme d'activité ORC

### 1.5.2. Diagramme d'enchainement des écrans: RDV

Ce module gère les RDV camions créés pour livrer ou récupérer un conteneur du terminal. 5 cas d'utilisations font partie de ce module fonctionnel :

- Créer Rendez-vous camion
- Modifier Rendez-vous camion
- Consulter Rendez-vous camion
- Supprimer Rendez-vous camion

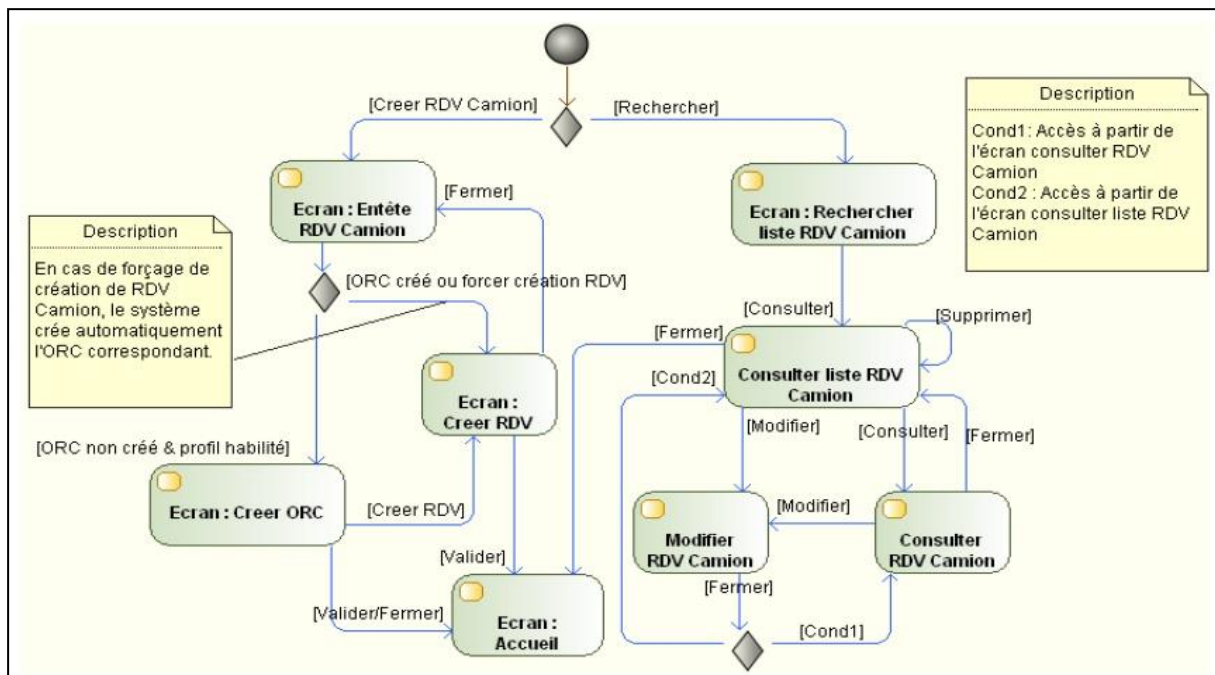


Figure 17: Diagramme d'activité RDV

### 1.5.3. Diagramme d'enchaînement des écrans: VC

Ce module gère les visites camions au terminal.

Une fois le camion transportant le conteneur vide arrive à la porte du terminal, l'agent vérificateur procède aux vérifications administratives pour autoriser l'entrée du camion. Il vérifie si chaque conteneur a un ordre de réservation enregistré ou actif sur la base du document présenté par le chauffeur du camion.

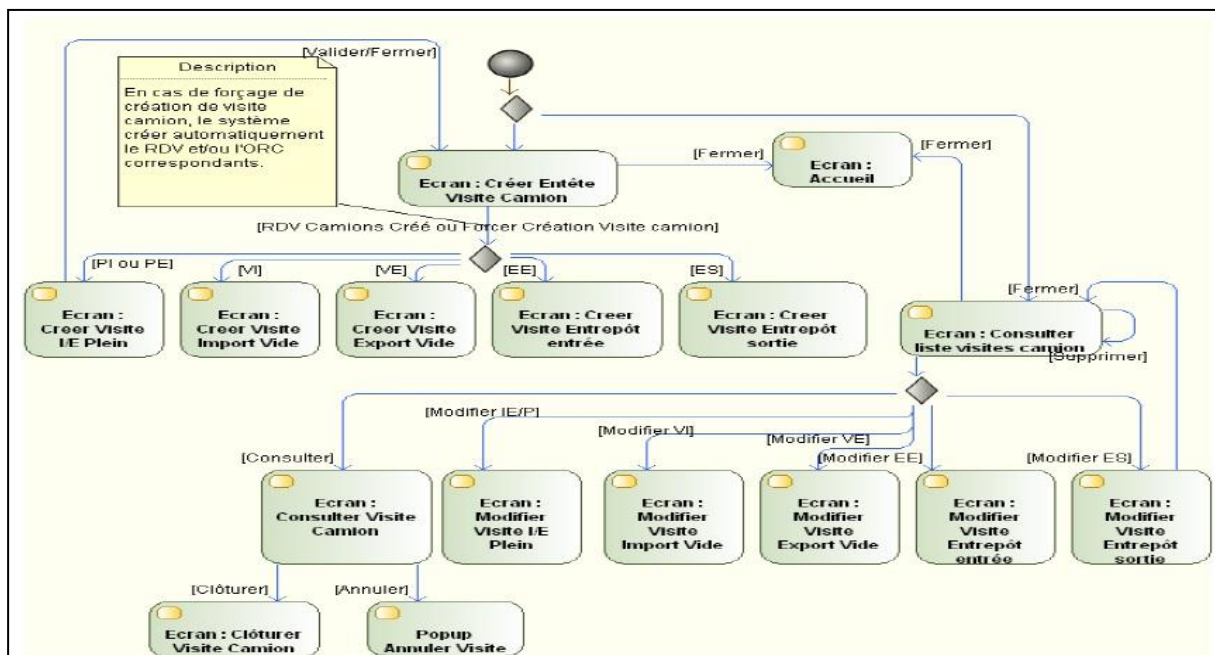


Figure 18: Diagramme d'activité VC



### 1.5.4. Diagramme d'enchaînement des écrans: PC

Ce module gère le pointage physique des camions. 5 cas d'utilisation font partie de ce module fonctionnel :

- Créer pointage camion
- Modifier pointage camion
- Consulter pointage camion
- Supprimer pointage camion

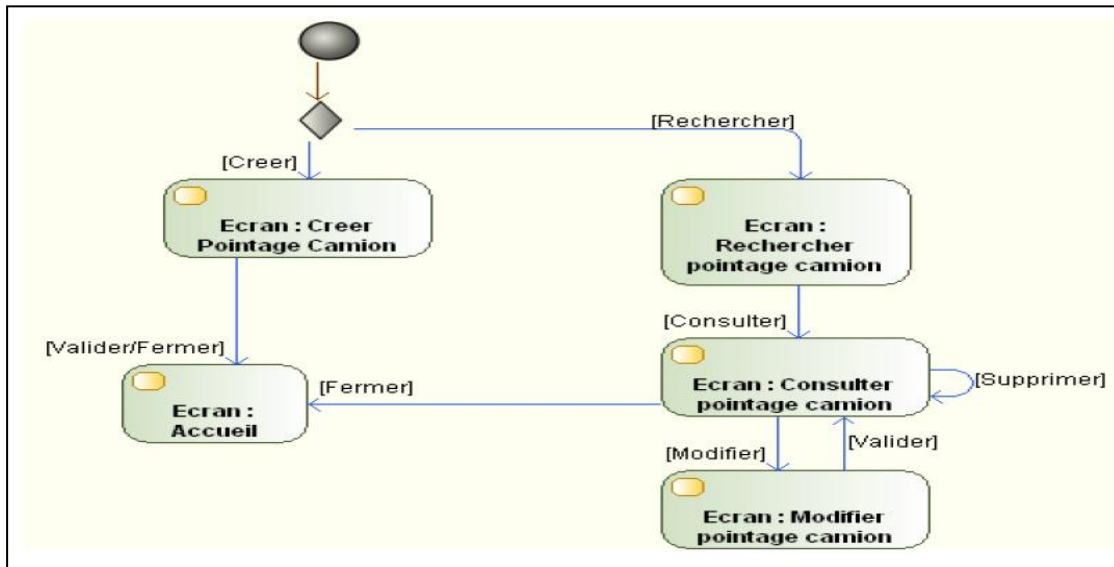


Figure 19: Diagramme d'activité PC

## 1.6. Le Diagramme de Classe D'Analyse (DC)

### 1.6.1. Les packages du diagramme de classe

Un package est un regroupement logique de classes ayant une forte cohésion interne et un faible couplage externe. Le couplage s'exprime structurellement par des liens entre les classes (associations, agrégations, compositions ou généralisation), et dynamiquement par des interactions qui se produisent entre les instances de classes. Nous avons donc créé quatre packages selon la nature des objets : Ordres de réservation conteneurs, Visites camions, pointage camion et RDV Camion (Figure 21).

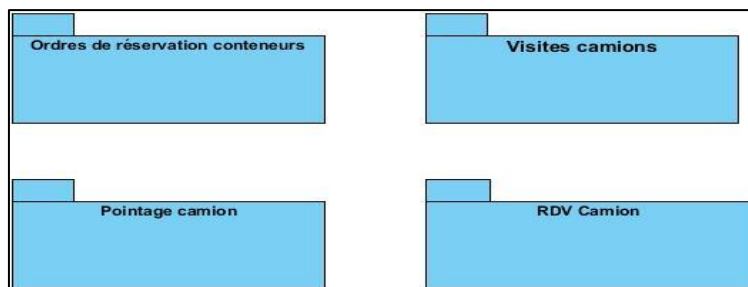


Figure 20: Diagramme de package de classe



## 1.6.2. Diagramme de classe

Dans ce qui suit on va modéliser le diagramme de classe de notre application :

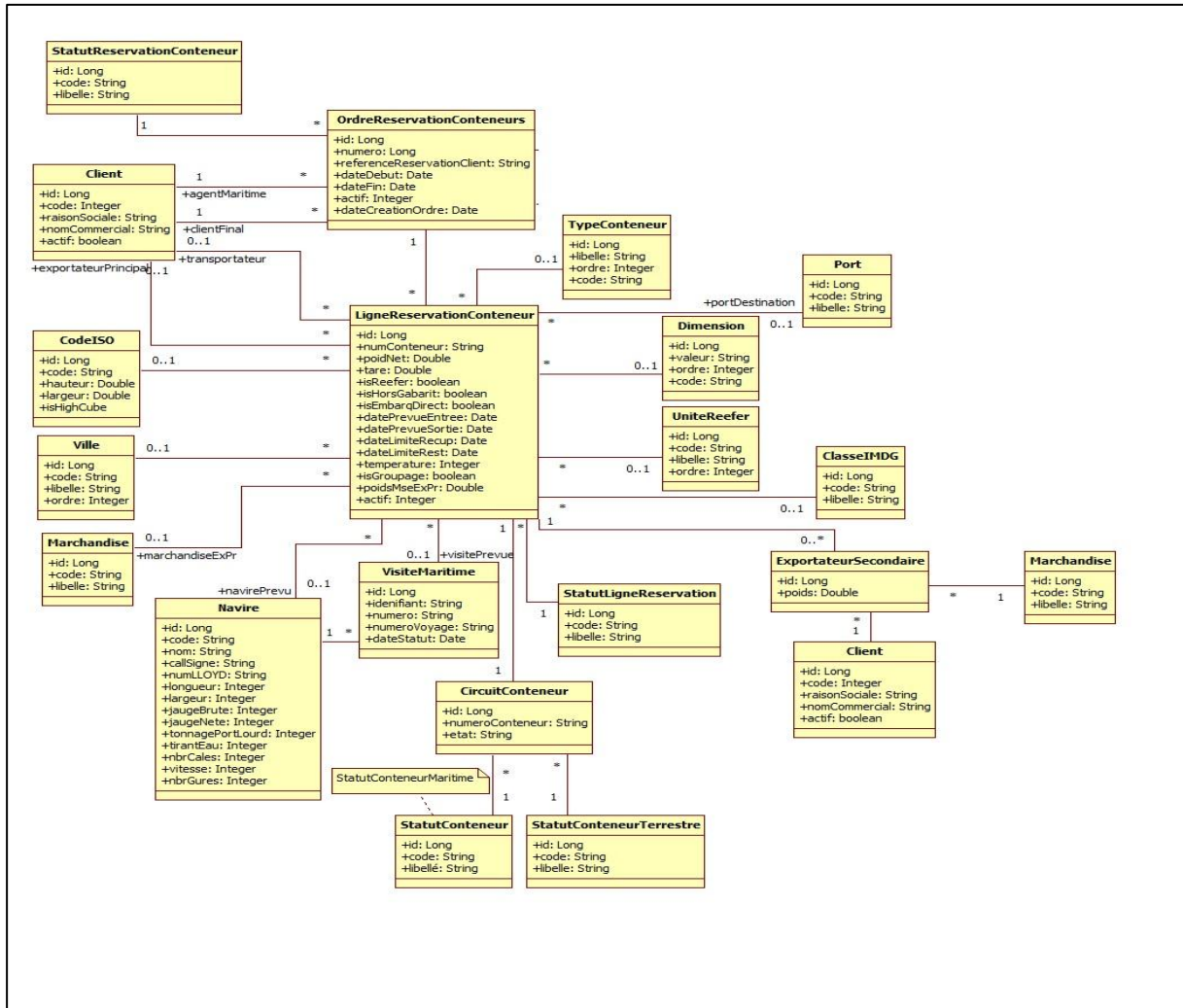


Figure 21 : Diagramme de classe d'analyse global de l'application

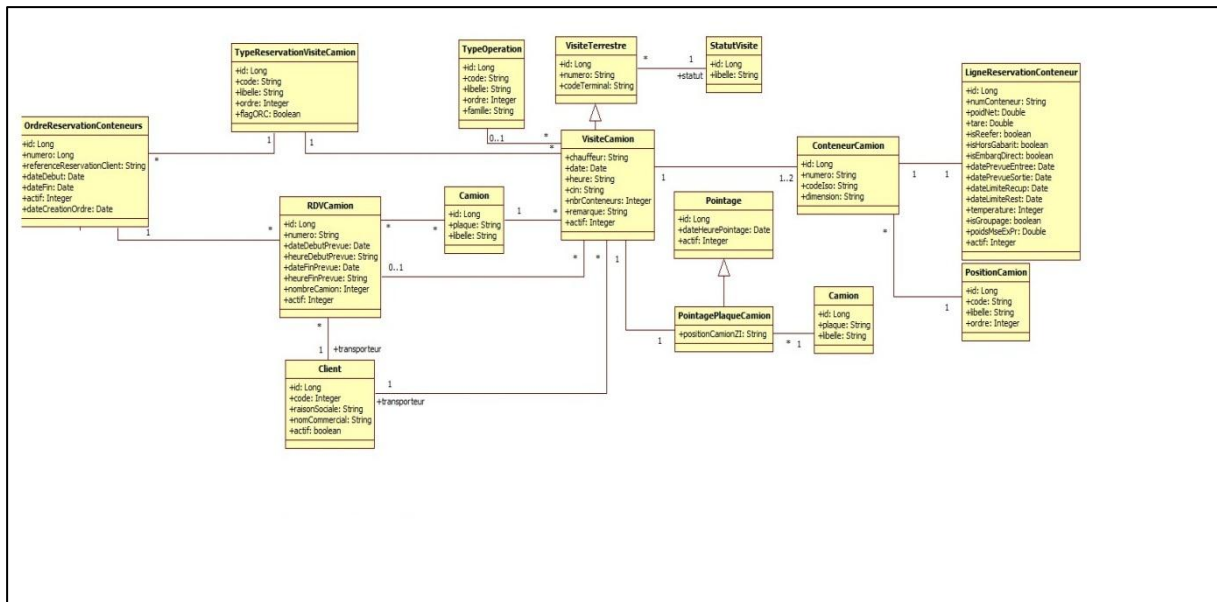


Figure 22 : Diagramme de classe d'analyse global de l'application

La figure 21,22 illustre les principales classes de notre application :

**Classe Client** : est la classe qui contient toutes les informations sur les utilisateurs de notre application à savoir : Transporteur, Agent maritime, responsable et client final ...

**Classe ORC** : cette classe qui contient toutes les informations sur un ordre de réservation conteneur à savoir son numéro sa date prévue ...

**Classe Lignes réservations conteneurs** : fait référence à un conteneur contient toutes les informations sur un conteneur son type (frigo , Dry ...etc ), cette classe est associée avec la classe ORC par une association many to one vu que ORC peut contenir plusieurs conteneurs.

**Classe Navire** : identifie un navire par son code.

**Classe RDVCamion** : contient toutes les informations sur un rendez-vous camion de notre application

**Classe Visite Camion** : contient toutes les informations sur une visite camion

**Classe Pointage plaque camion** : cette classe contient toutes les informations sur un pointage camion

**Classe Type réservation visite camion** : cette classe contient l'objet de la visite camion à savoir :

- 1 **Entrée Plein** : le camion amène un conteneur plein au Terminal
- 2 **Sortie Plein** : le Terminal livre au camion un conteneur plein
- 3 **Entrée Vide** : le camion amène un conteneur vide au terminal pour être exporté
- 4 **Sortie Vide** : le terminal livre au camion un conteneur vide
- 5 **Entrée Dépôt** : le camion amène un conteneur vide pour entreposage momentané
- 6 **Sortie Dépôt** : le terminal livre au camion un conteneur vide qui était en entreposage

**Classe Type conteneur :** cette classe contient toutes les informations sur le type du conteneur à savoir :

- 1 **Dry :** Il s'agit du conteneur le plus courant, capable de transporter la plupart des marchandises « sèches », boîtes, cartons, caisses, sacherie, balles, palettes, futs, etc.
- 2 **Reefer :** Ce type de conteneur est utilisé pour le transport sous température dirigée. Avec isolation thermique, il est équipé d'un moteur électrique .  
...etc



**Classe Marchandise :** cette classe contient tous les codes des marchandises.

### 1.6.3. Génération des tables de la base de données

Dans ce qui suit on a dégagé les tables de notre base de données qu'on aura besoin pour le développement de notre application.

# CHAPITRE 5 : Etude technique du projet

## 1. Introduction

Le choix de l'architecture technique est un facteur déterminant dans la performance de l'application, il dépend de plusieurs exigences techniques que je vais étudier dans ce qui suit.

La structuration d'une application permet de présenter cette dernière en des couches logiques : la logique de présentation, la logique métier et la logique de manipulation de données. Ainsi, une application structurée en couches est facilement maintenable dans la mesure où un nouvel intervenant est capable, à partir de la description de l'architecture, de comprendre le rôle de chaque composant développé et s'insérer rapidement dans cette logique.

## 2. L'architecture générale de l'application

### 2.1. Vue d'ensemble des différentes couches de l'architecture

Les schéma ci-dessous décrit l'ensemble des composants constituant l'architecture applicative de l'application.

L'architecture générale choisie pour le développement de ce projet est une architecture multicouche. Ce Modèle préconise le découpage d'une application en plusieurs couches logiques spécialisées :

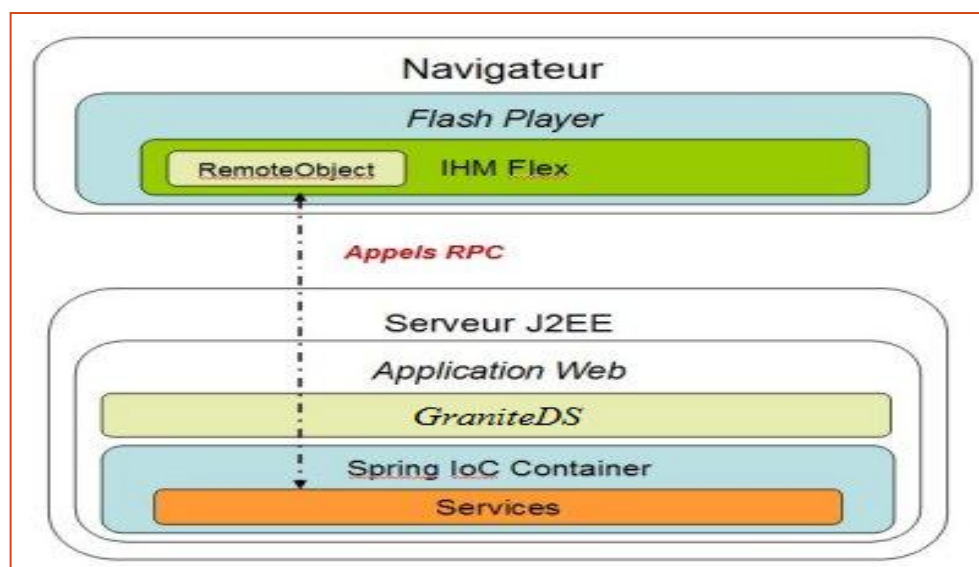


Figure 23 : architecture générale de l'application

Dans cette architecture, le client **Flex réalise des appels de service s'exécutant au sein du serveur d'application JEE**. Pour cela, on utilise les appels de type RPC (Remote Procedure Call), et plus spécifiquement le composant Remote Object de Flex, permettant d'invoquer des services Java, sans avoir à se soucier de la sérialisation/désérialisation des objets échangés entre Flex et java. **Granite DS** est une brique open-source, fournit par Adobe, supportant les appels RPC via RemoteObject. En pratique, il s'agit d'une servlet responsable, entre autres, de :

- Localiser et invoquer les services Java.
- Désérialiser les données Flex reçues en instances d'objets Java et vice-versa.

GraniteDS s'interpose donc entre le client Flex et les services Java à la manière d'un proxy. Dans notre architecture, Spring est utilisé comme container IoC. Spring gèrera donc le cycle de vie de nos services. GraniteDS devra s'adresser à Spring afin de localiser un service à invoquer.

L'intérêt d'un modèle d'architecture en couches réside dans l'indépendance des couches entre elles. Cela permet de disposer d'un système évolutif (changement de base de données ou d'outil de mapping objet / relationnel sans impacter les autres couches par exemple), et de pouvoir développer et tester séparément les couches.

Cela repose toutefois sur l'utilisation de Design Pattern éprouvé dans la structuration des couches et la communication inter-couches :

- MVC pour la Couche présentation
- Business delegate.
- DAO (Data Access Object) pour la couche accès aux données.

### 3. L'architecture détaillée de l'application

L'application est décomposée en couche applicatives, chaque couche à un rôle prédéfini dans l'architecture globale.

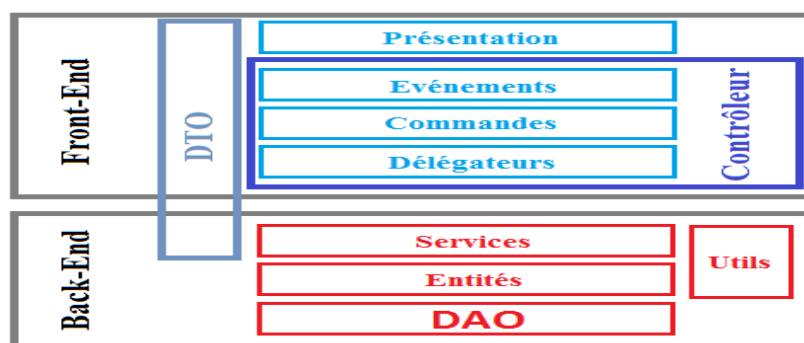


Figure 24 : architecture générale de l'application

Les blocs en bleu représentent les couches du côté Client « front-end » et les blocs en rouge représentent les couches du côté serveur « back-end ».

- ✓ **La couche DTO** : cette couche contient les objets qui transportent les données, que ce soit du côté Flex vers Java (le cas d'une recherche) ou bien du Java vers Flex (le cas de chargement des données, retours des contrôles métier...), et permet de regrouper les données à fin de les transférer d'un seul coup dans le réseau. Le contenu de ces objets doit être conforme aux contenus des écrans de l'application, ceci évite de faire des traitements supplémentaires pour la préparation des données à afficher au niveau de la couche Vues.

### 3.1. Le pattern DTO

Dans une architecture multi-couches sans état, il est relativement fréquent de s'échanger des graphes d'objets en paramètre de méthode. Aujourd'hui le pattern DTO (Data Transfert Object) est la technique la plus utilisée pour assurer le découplage entre la couche de présentation et les objets métier stockés sur le serveur.

Les DTO sont des structures de données qui véhiculent l'état (les propriétés de cet objet à un instant T) des objets serveur sur le client et permettent également de remonter des informations à persister.

Un DTO n'est rien de plus qu'une classe exposant des propriétés, mais pas de méthodes. Du point de vue design pur, le pattern DTO est une solution quasi parfaite car il permet d'isoler le modèle du domaine de la présentation, permettant à la fois un couplage lâche et un transfert de données optimisé.

La couche de présentation et la couche de service peuvent partager des données au travers de contrats (les DTO) et non plus d'entités métier. Un contrat de données est essentiellement une représentation neutre des données échangé entre composant.

Le contrat de données décrit les données qu'un composant reçoit, mais ce n'est pas une classe spécifique du système, comme une entité. Dans sa mise en œuvre, le contrat de données est une classe, mais il ressemble plus à une classe d'assistance spécialement créé pour un service particulier.

L'adoption de contrats de données ajoute beaucoup de souplesse à la couche de service, et par conséquent également à l'application entière.

### 3.2. DTO vs Bean du Domaine

Ci-dessous les avantages et inconvénient du modèle de DTO para rapport à l'utilisation des beans du domaine.

#### Avantage des DTO

- **Amélioration des performances. En transmettant plus de données dans un seul appel distant**, l'application peut réduire le nombre d'appels distants. Les appels distants ralentissent considérablement les applications. La réduction du nombre d'appels est l'un des meilleurs moyens d'améliorer les performances.
- **Transparence des mécanismes internes.** Le fait d'émettre et de recevoir davantage de données dans un seul appel masque plus efficacement les mécanismes internes d'une application distante derrière une interface à granularité grossière.
- **Découverte d'objets métier.** Dans certains cas, le fait de définir un DTO aide à découvrir des objets métier pertinents. Lorsque l'on crée des classes personnalisées servant de DTO, on remarque souvent

des regroupements d'éléments qui sont présentés à un utilisateur ou à un autre système en tant qu'ensemble cohérent d'informations.

- Aptitude au test. L'encapsulation de tous les paramètres dans un objet sérialisable peut favoriser l'aptitude au test. Par exemple, on peut lire les DTO à partir d'un fichier XML et appeler des fonctions distantes pour le tester.

### **Inconvénients des DTO**

- Risque de multiplication du nombre de classes. Si vous choisissez d'utiliser des DTO fortement typés, vous devrez peut-être créer un (ou deux, en tenant compte de la valeur de retour) DTO pour chaque méthode distante. Même dans une interface à granularité grossière, cela peut induire un grand nombre de classes. Il peut devenir difficile d'en coder et gérer une telle quantité. La génération de code automatique peut palier en partie à ce problème.
- Calculs supplémentaires. Le fait de convertir un format de données sur le serveur en flux d'octets transportable sur le réseau, puis à nouveau en format objet dans l'application cliente, peut induire une surcharge assez importante. En général, on agrège des données provenant de sources diverses dans un seul DTO sur le serveur. Pour améliorer l'efficacité des appels distants sur le réseau, il faut faire quelques calculs supplémentaires d'un côté ou de l'autre pour agréger et sérialiser les informations via des outils comme Dozer.
- Codage supplémentaire. Le DTO demande d'instancier un nouvel objet et d'appeler des méthodes d'obtention et de réglage pour chaque paramètre. L'écriture de ce code peut être fastidieuse, même si des outils comme Dozer simplifient cette tâche.

### **3.3. Framework MapStruct**



C'est framework qui permet de générer les classes DTO a partir des entités .

### **3.4. Fonctionnement de FLEX**

Flex est une solution de développement créée par Macromedia en 2004 puis reprise par Adobe en 2006, permettant de créer et de déployer des applications Internet riches(RIA) multiplate-formes grâce à la technologie Flash et particulièrement son lecteur. Son modèle de programmation fait appel MXML (basé sur XML).et ActionScript 3.0, reposant sur ECMAScript.

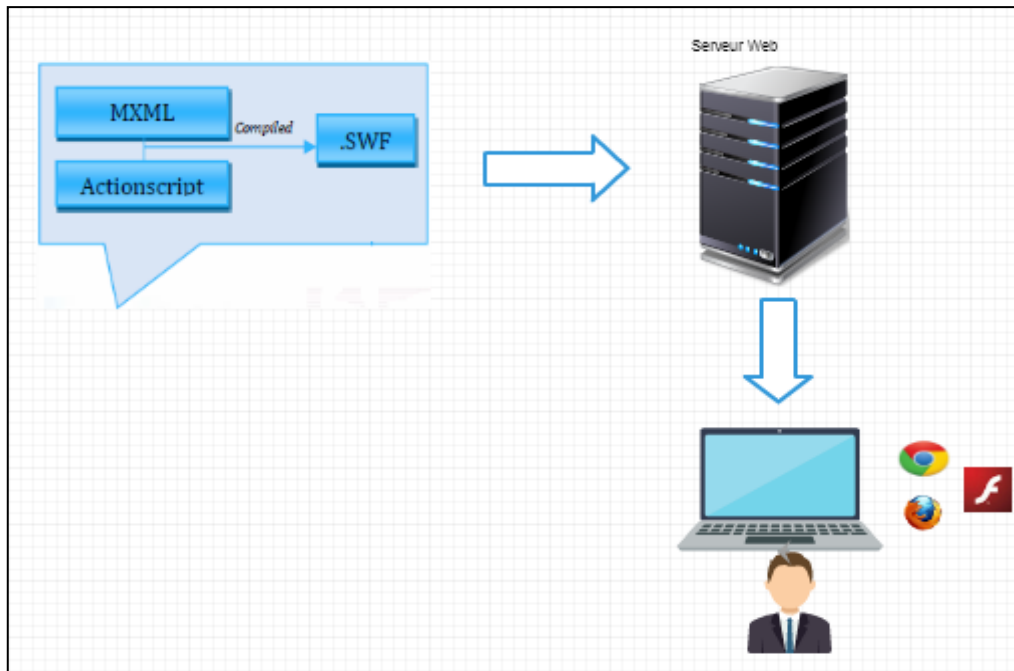


Figure 25 : Fonctionnement flex

### 3.5. La technologie JEE

L'élaboration du portail s'appuie sur la plateforme JEE (Java Enterprise Edition) qui est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants.

JEE définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Ce choix est justifié par plusieurs facteurs à savoir :

- La maturité et la richesse de cette technologie ;
- La possibilité de la réutilisation des différents composants qui en font partie ;
- La séparation forte qu'offre la plupart des framework relevant de cette architecture
- La maintenabilité des applications.

## 4. Technologies & Framework de développement



## 4.1. Java

Java est un langage développé par Sun Microsystems depuis la fin des années 1980. La première version de Java a été mise au point en 1991. Java a connu depuis un essor considérable, notamment dans le domaine des applications distribuées via Internet, grâce au soutien de Netscape depuis 1995.

L'aspect purement Objet de Java permet une meilleure répartition du travail entre les programmeurs qui peuvent développer ou utiliser différents objets sans se préoccuper des traitements réalisés à l'intérieur. De plus, il est plus facile d'adapter et de faire évoluer les spécificités d'un objet à partir du moment où, vu de l'extérieur, cet objet réalise les mêmes actions. Ainsi, les objets mis en oeuvre en Java sont réutilisables à souhait. Pour résumer, Java est portable, sûr, orienté objet et indépendant de toute plateforme.

## 4.2. L'Actionscript

ActionScript est le langage de programmation orientée objet (POO) basé sur ECMAScript utilisé au sein des applications Adobe Flash et Adobe Flex.

ActionScript est une version sophistiquée du langage de script, introduit dans Flash ActionScript 3. Ce langage permet d'ajouter de l'interactivité aux animations Flash, en répondant aux actions de l'utilisateur tout comme c'est le cas dans la vie réelle, les actions sont déclenchées par des événements.

## 4.3. MXML

Le MXML (Macromedia flex Markup Language) est avant tout un langage de balises basé sur XML. Son rôle principal est de permettre l'agencement des composants au sein d'une interface graphique.

## 4.4. XML

XML (Extensible Markup Language, « langage de balisage extensible») est un langage informatique de balisage générique qui dérive du SGML. Cette syntaxe est dite extensible car elle permet de définir différents espaces de noms, c'est à dire des langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, RSS... Cette syntaxe est reconnaissable par son usage des chevrons (< >) encadrant les balises. L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche...) entre systèmes d'informations hétérogènes (interopérabilité).

## 4.5. Hibernate /Jpa

Travailler dans les deux univers qui sont, l'orienté objet et le relationnel peut être lourd et consommateur en temps. Hibernate est un outil de mapping objet/relationnel pour le monde Java. Il s'occupe du transfert des classes Java dans les tables de la base de données (et des types de données Java dans les types de données SQL). Il permet aussi de requêter les données et propose des moyens de les récupérer. Il peut donc réduire de manière significative le temps de développement qui aurait été autrement perdu dans une manipulation manuelle des données via SQL et JDBC.

#### 4.6. GraniteDS

Plateforme open source d'intégration Flex / Java pour la réalisation d'applications RIA. Il offre une passerelle back-end pour invoquer les services Java et désérialiser les données Flex reçues en instances d'objets Java et vice-versa.

#### 4.7. Spring IOC

Permet de faire L'injection de dépendances (Dependency Injection) est un mécanisme qui permet d'implémenter le principe de l'inversion de contrôle. Il 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.

#### 4.8. Flex

Flex est un framework Open Source gratuit qui permet de créer des applications web ultra-interactives et expressives se déployant de manière identique sur la plupart des navigateurs, postes de travail et systèmes d'exploitation.

Flex offre un modèle de programmation évolué qui repose sur des langages standard et gère les modèles de conception courants. MXML, langage déclaratif basé sur XML, sert à décrire l'agencement et le comportement de l'interface utilisateur tandis que le langage de programmation orienté objet ActionScript 3.0 est employé pour la création de fonctions de traitement côté client.

#### 4.9. Cairngorm MVC

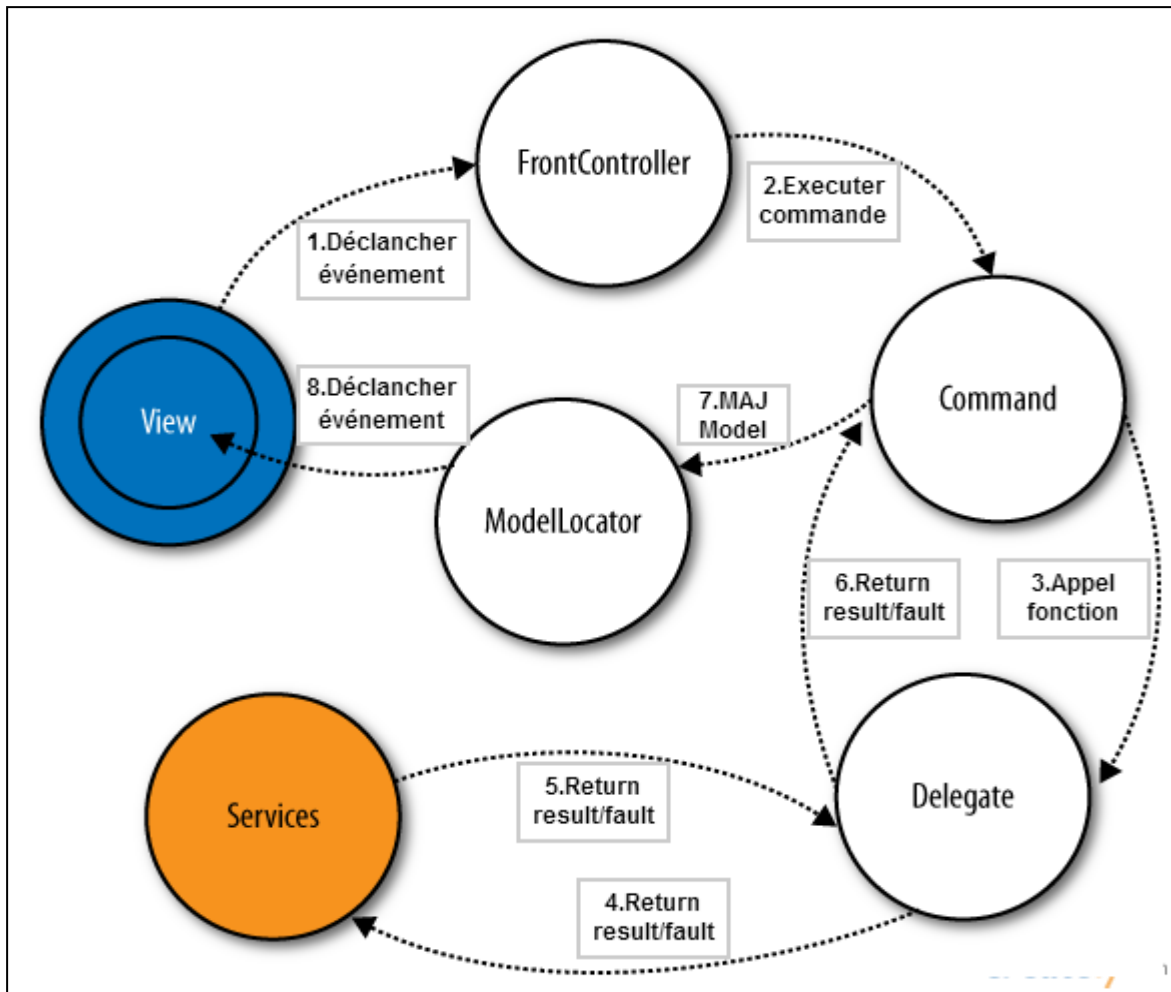


Figure 26 : L'architecture Cairngorms MVC

Cairngorm est le framework officiel MVC de Adobe permettant une séparation des traitements de l'affichage. Le schéma ci-dessus (repris du site Adobe) décrit l'interaction entre les différents composants formant l'architecture MVC Cairngorm

#### 4.10. MVC

Le Model-View-Controller (MVC) est un modèle de conception logicielle très répandu et fort utile. Créé dans les années 80 par Xerox PARC pour Smalltalk-80, il est aujourd'hui fortement recommandé dans l'univers J2EE. Néanmoins il faut retenir que c'est un modèle de conception, et il est donc indépendant du langage de programmation.

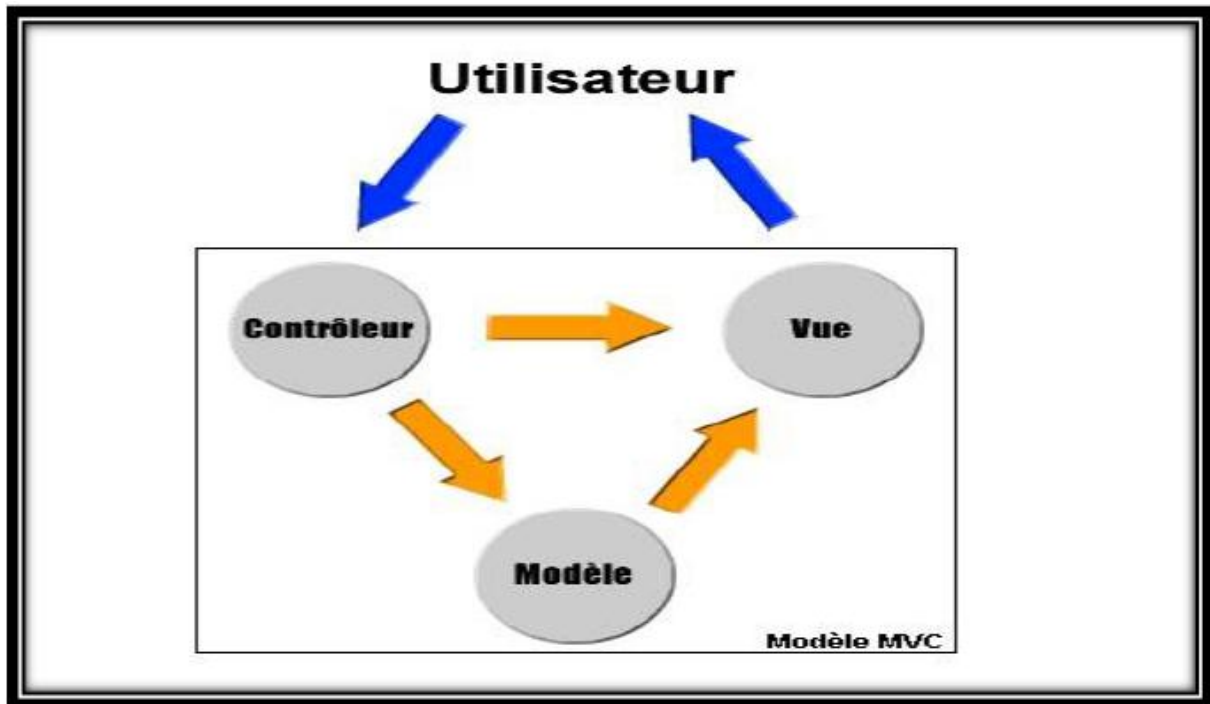


Figure 27 : L'architecture MVC

#### 4.11. UML

Visual Paradigm for UML est, comme son nom la 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.

## 5. Conclusion

L'étude technique du projet a présenté l'architecture logicielle et les exigences techniques de l'application, la technologie utilisée et le rapport détaillé des frameworks techniques choisis. Dans le chapitre qui suit, nous présentons la réalisation et la mise en œuvre de l'application .

# CHAPITRE 6 : Réalisation

## 1. Environnement de développement

### 1. ➤ Eclipse



Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

Plusieurs logiciels commerciaux sont basés sur ce logiciel libre, comme par exemple *IBM Lotus Notes 8*, *IBM Symphony* ou *WebSphere Studio Application Developer*.

### 2. ➤ Apache tomcat



**Apache Tomcat** est un conteneur de servlet J2EE. Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des servlets et des JSP de Sun

Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP.

### 3. ➤ SQL SERVER



*Microsoft SQL Server* est un système de gestion de base de données (abrégé en SGBD) incorporant entre autres un SGBDR (SGBD relationnel) développé et commercialisé par la société Microsoft. Il ne fonctionne que sous les OS Windows.

En fait MS SQL Server est une suite composée de cinq services principaux :

- Le moteur relationnel (OLTP) appelé **SQL Server** ;
- Le moteur décisionnel (OLAP) appelé **SSAS** (*SQL Server Analysis Services*) incluant un moteur de stockage pour les cubes, des algorithmes de forage (*data mining*) et différents outils de BI (Business Intelligence) ;
- Un ETL (Extract Transform and Load) appelé **SSIS** (*SQL Server Integration Services*) destiné à la mise en place de logiques de flux de données, notamment pour alimenter des entrepôts de données (data warehouse) ;
- Un outil de génération d'état appelé **SSRS** (*SQL Server Reporting Services*) permettant de produire des rapports sous différentes formes et exploitant les ressources du moteur décisionnel (bases "reportServer...") à la fois pour y stocker les rapports mais aussi y cacher les données de ces derniers afin de faire du "warmup" ;
- Un système de planification de travaux et de gestion d'alerte appelé **Agent SQL** qui utilise lui aussi les services du moteur SQL (base msdb).

### 4. ➤ Adobe Flash builder



Adobe Flash Builder 4.6 est un environnement de développement de jeux et d'applications basé sur le langage ActionScript et le framework Flex Open Source. Flash Builder Premium inclut des outils de test professionnels : profileurs, système de contrôle du réseau, prise en charge de tests unitaires, etc.

## 5. ➤ Maven



Maven est un outil permettant d'automatiser la gestion de projets Java. Il offre entre autres les fonctionnalités suivantes [MAVEN] :

- Compilation et déploiement des applications Java (JAR, WAR) ;
- Gestion des librairies requises par l'application;
- Intégration dans différents IDE (Eclipse, JBuilder).

Maven utilise un paradigme connu sous le nom de POM (Project Object Model) afin de décrire un projet logiciel, ses dépendances avec des modules externes et l'ordre à suivre pour sa production.

Chaque projet ou sous-projet est configuré par un POM (pom.xml) le décrivant en détail. On y trouve de nombreuses informations concernant le projet (Version, Gestion des configurations, Gestion des dépendances, La structure du projet, Les ressources de l'application...).

## 6. ➤ SVN



SVN est un système permettant de travailler à plusieurs sur une ou plusieurs ressources. Il gère les numéros de version automatiquement à chaque modification faite sur une ressource. Il permet notamment de signaler et résoudre les conflits lorsque deux modifications sont faites sur un même fichier en même temps [SVN].

L'outil SVN s'avère être d'un grand intérêt pour les développements en équipe:

- Permet de centraliser les sources dans un dépôt accessible via le réseau;
- Permet de garder un historique horodaté des modifications ;
- Facilite la collaboration entre développeurs ;
- Gère le versionning

## 2. Réalisation

### 2.1. Fruit du travail

Dans cette partie on va présenter quelques captures d'écran sur le projet :

#### 1. Ecran Créer Entête Ordre de Réservation Conteneurs «entrée plein »

The screenshot shows a web browser window with the URL '128.129.187.37:8092/tos-module-gtc-web/welcome'. The page header includes the 'Marsa Maroc' logo and a navigation menu with items like 'Structure Navire', 'Pointage conteneur', 'Ordre de réservation', 'RDV camion', 'Pointage Camion Vide', and 'Visite camion'. The main content area is titled 'Créer Ordre de réservation conteneurs' and contains a form for creating a container reservation. The form fields are: 'Agent maritime' (000058), 'Réf réservation client' (ref1), 'Date de début' (07/05/2016), 'Date de fin' (13/05/2016), and 'Type' (Entrée plein). A search icon and the text 'TRANSPORTS MAROCAINS' are also visible.



La figure ci-dessus présente l'espace de création d'un ordre de réservation des conteneurs qui contient cinq champs :

- Loupe dans en sélectionne l'agent maritime
- Reference de réservation
- Date de debut
- Date de fin
- Type de réservation : réservation : entrée plein, sortie plein, entrée vide, sortie vide, entrée dépôt et sortie dépôt.

En appuyant sur le bouton « suivant », l'écran détail de la création de l'ORC est affiché. Les champs varient en fonction du type de la réservation

Structure Navire Pointage conteneur **Ordre de réservation** RDV camion Pointage Camion Vide Visite camion

Créer Ordre de Réservation Conteneurs Entrée Plein

**Ordre de réservation conteneurs**

Numéro ORC :  
Agent maritime : 000058 - TRANSPORTS MAROCAINS Type : Entrée plein  
Réf réservation client : ref1  
Date de début : 07/05/2016 Date de fin : 13/05/2016

Liste des conteneurs :

N° Conteneur	Code ISO	Dimension	Type	Poids Global	Port destination	Transporteur	Groupage		

Les informations liées à l'ORC sont récupérées à partir de l'écran «Créer ORC» relatif à la création de l'entête de la réservation. L'utilisateur devra par la suite renseigner les informations liées aux conteneurs ou ce qu'on appelle ligne de réservation conteneur en cliquant sur le bouton '+' qui ouvre une fenêtre **pop-up**

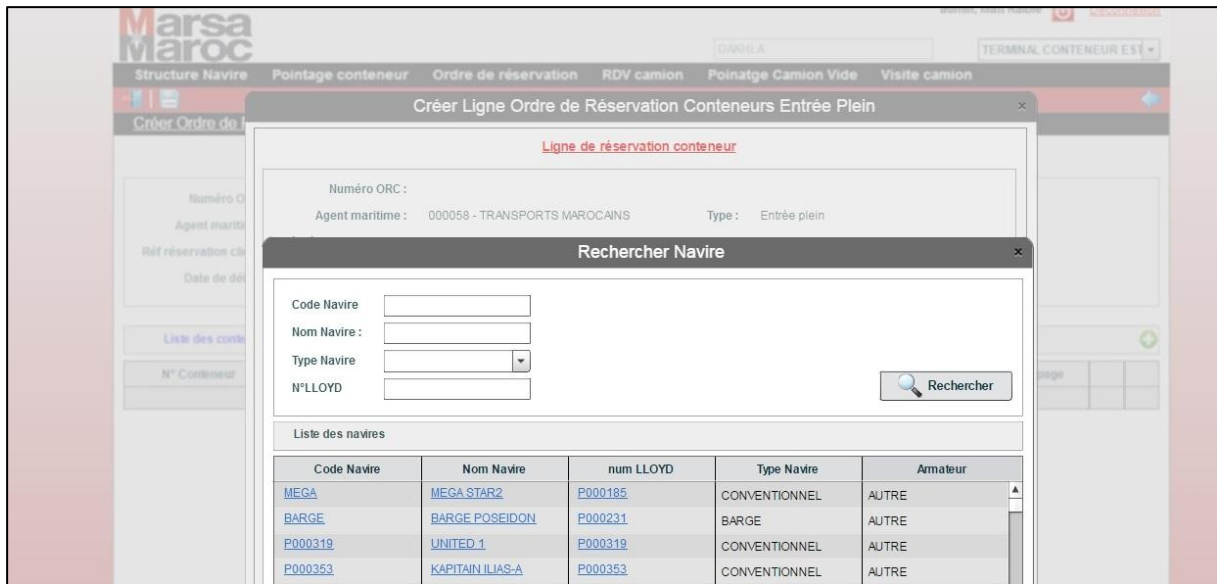
## 2. Pop-up Créer Ligne Réservation Conteneurs «entrée plein »

Après en remplie les informations comme code iso qui sont enregistrer dans la base données suivant le code on définie le type et la dimension du conteneur

### 3. Auto Complétion Code Iso

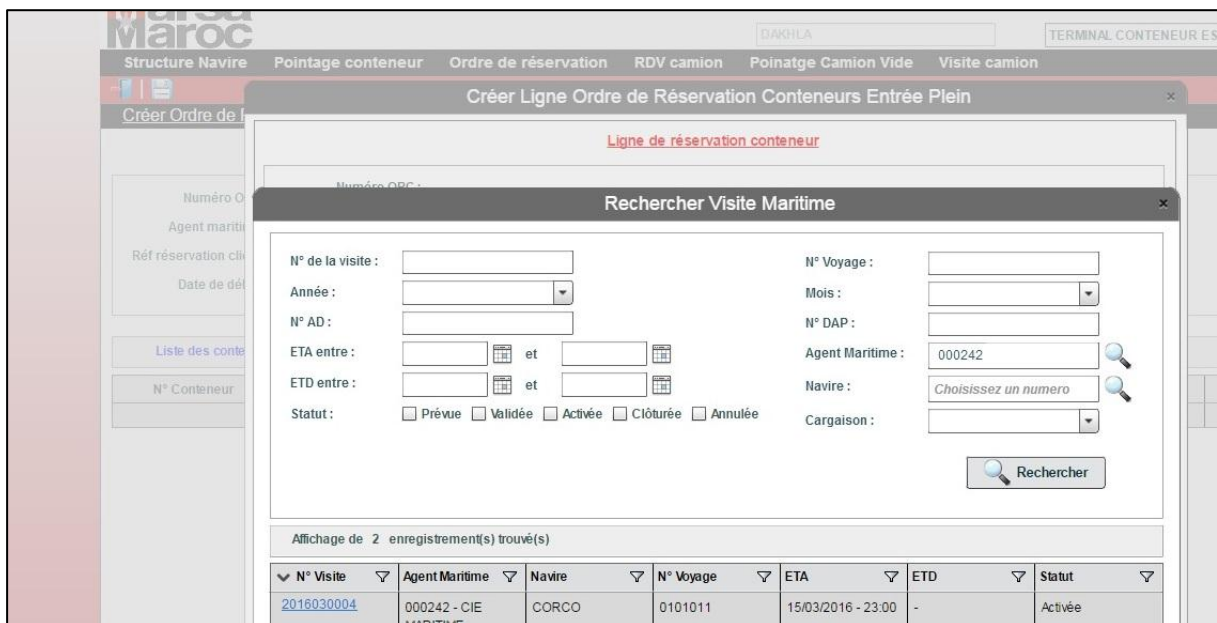
La même chose pour navire lors du clique sur la loupe une popup et affiche dans en charge une liste des navires dans la bdd suivant des critère nom , type de navire , et ceci bien sûr et fait par un service envoyer lors de l'évènement click sur le bouton rechercher .

### 4. Pop-up List des Navires



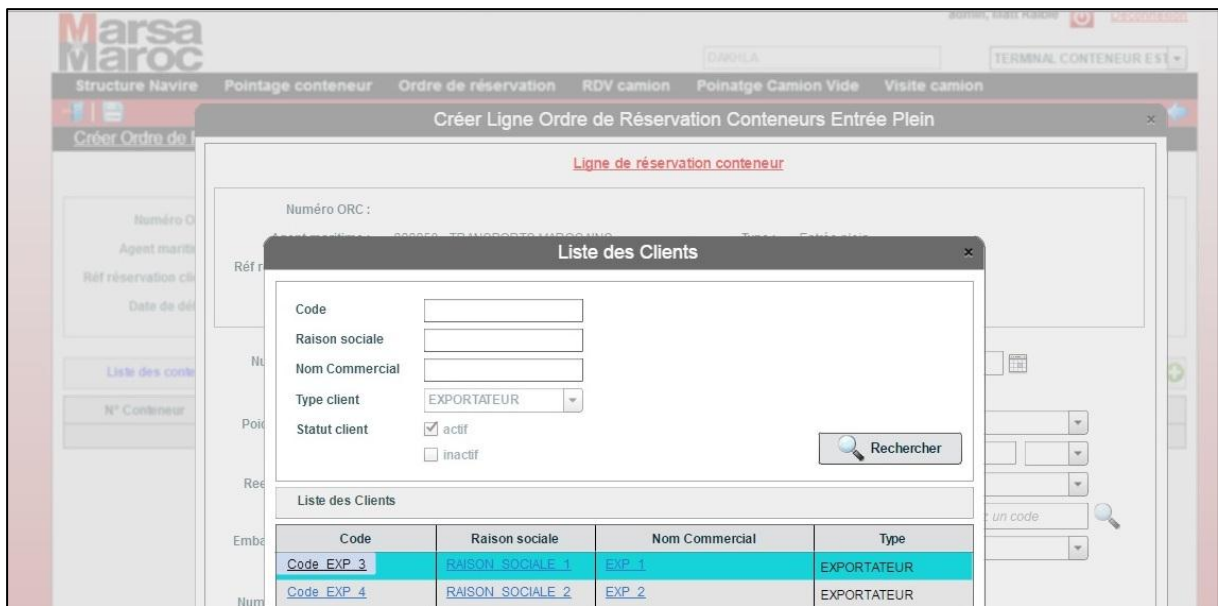
La même chose pour visite maritime

## 5. Pop-up List Visites Maritimes

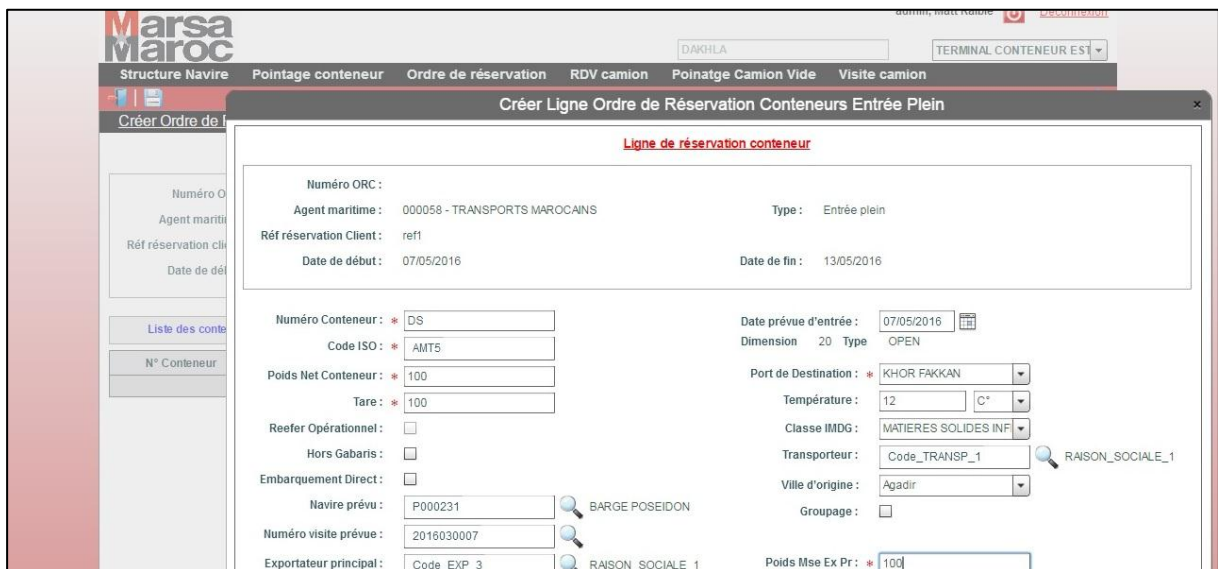


## 6. Pop-up List des Clients

Affichage du popup List client



Finalement en remplie tous les champs



On clique sur le bouton ajouté et le conteneur et bien ajouter



## 7. Pop-up Modifier Ligne Réservation Conteneurs «entrée plein »

Tester la modification d'un conteneur en cliquant sur le bouton modifier.

The screenshot shows a web application interface for modifying a container reservation line. The page title is 'Modifier Ligne Ordre de Réservation Conteneurs Entrée Plein'. The form contains the following fields:

- Numéro ORC: (empty)
- Agent maritime: 000058 - TRANSPORTS MAROCAINS
- Type: Entrée plein
- Réf réservation Client: ref1
- Date de début: 07/05/2016
- Date de fin: 13/05/2016
- Conteneur N°: CONT1
- Code ISO: AMT5
- Date prévue d'entrée: 07/05/2016
- Dimension: 20
- Type: OPEN
- Poids Net Conteneur: 100
- Tare: 100
- Port de Destination: KHOR FAKKAN
- Température: 12 C°
- Classe IMDG: MATIERES SOLIDES INF
- Transporteur: Code\_TRANSP\_1
- Reefer Opérationnel:
- Hors Gabaris:
- Embarquement Direct:
- Navire prévu: P000231 (BARGE POSEIDON)
- Ville d'origine: Agadir
- Groupage:
- Numéro visite prévue: 2016030007
- Exportateur principal: Code EXP\_3 (RAISON SOCIALE 1)
- Poids Mse Ex Pr: 100

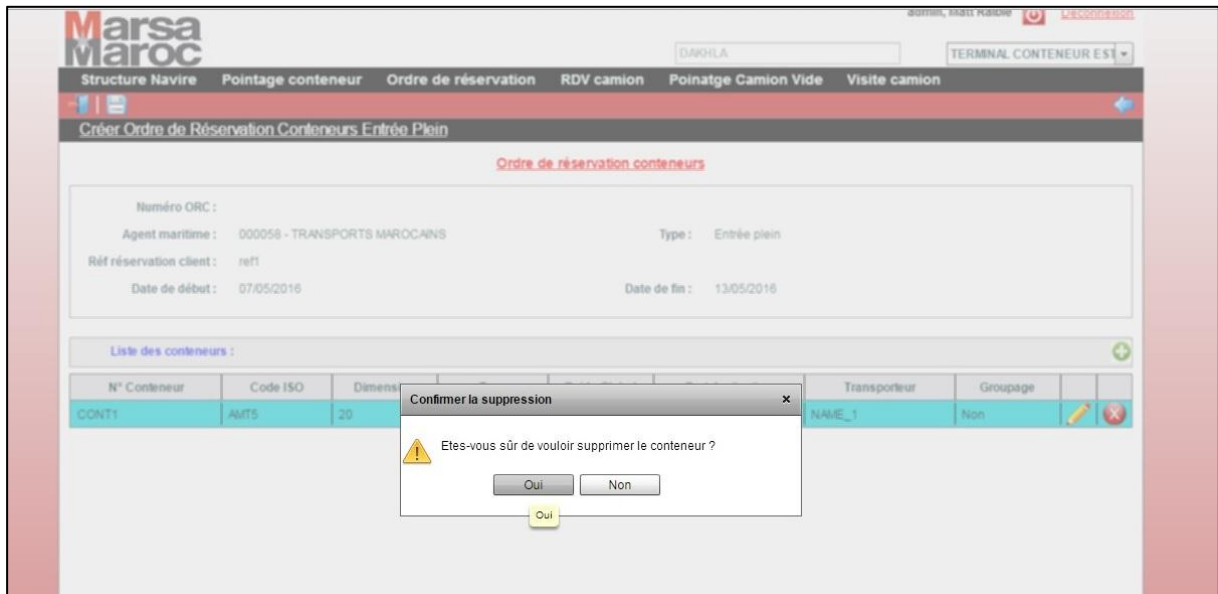
Modification avec succès

The screenshot shows the 'Ordre de réservation conteneurs' page. It displays a table with the following data:

N° Conteneur	Code ISO	Dimension	Type	Poids Global	Port destination	Transporteur	Groupage	
CONT1	AMT5	20	OPEN	100	KHOR FAKKAN	NAME_1	Non	 

## 8. Supprimer Ligne Réservation Conteneurs «entrée plein »

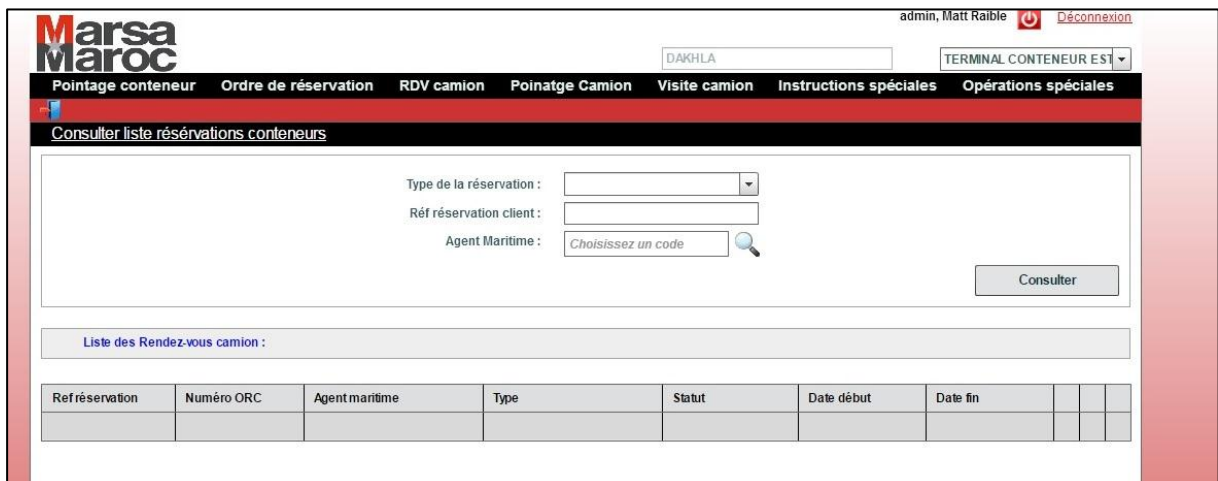
Tester la suppression d'un conteneur :



Suppression avec succès :



## 9. Ecran Consulter Ordre de Réservation Conteneurs «entrée plein »



## 2.2. Test unitaire

En programmation informatique, le **test unitaire** (ou « **T.U.** », ou « **U.T.** » en anglais) est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »).

### 2.2.1. Types de tests unitaires

Un cas de test vérifie un chemin possible emprunté à l'exécution du code.

Un cas de test peut vérifier un des types de test suivant :

- Test passant (TPA)
- Test non passant (TNP)
- Test aux limites (TAL)
- Test de non régression (TNR)

#### Les tests passants (TPA)

Valident les règles de gestion.

⇒ Vérifier une à une toutes les nouvelles règles de gestion des programmes.

Exemple : contrôle la saisie d'une hauteur de camion : de 2 à 10 mètres.

Le test normal consiste alors à tester avec véhicule Camion hauteur entre 2 et 10 -> pas de blocage

#### Les tests non passants (TNP)

Mettent en évidence les erreurs devant être interceptées par le programme.

⇒ Chaque message d'erreur doit être testé.



*Exemple : Teste Camion avec hauteur  $\geq 10$  et hauteur  $\leq 2$   
Le message d'erreur est retourné.*

### **Les tests aux limites (TAL)**

Vérifient le comportement du programme lors d'une utilisation limite.

- ⇒ Envisager tous les cas extrêmes pouvant faire « planter » le programme : valeurs limites, fichiers inexistant, vides ou volumineux, zones non renseignées, format de saisie incorrect...

*Exemple : tester avec Camion de hauteur  $< 0$  ou contenant des caractères.*

### **Les tests de non régression (TNR)**

Vérifient le bon fonctionnement des anciennes règles de gestion (iso fonctionnalité).

- ⇒ Comparer les résultats des tests avant et après modification.

NB : dérouler les TU des évolutions précédentes

*Exemple : tester avant et après modification avec véhicule Voiture de hauteur  $< 2$   
Résultats des traitements identiques.*

## **3. Conclusion**

Dans ce chapitre, on a analysé les besoins de client, ces besoins ont été modélisés dans des diagrammes UML, ce dernier est un langage unifié et n'est pas une méthode d'analyse et conception. Dans ce qui suit on a présenté les diagrammes utiles pour l'analyse à savoir le fameux diagramme des cas d'utilisation plus les diagrammes d'enchaînement des écrans lié à chaque cas d'utilisation et le diagramme de classe (d'analyse). La seconde partie du chapitre, on a attaqué la conception, c'est-à-dire comment on va implémenter notre application suivant une architecture trois tiers (couche présentation, couche métier, couche accès de données). La spécification de cette architecture c'est qu'elle marche bien avec le design pattern MVC. Ce dernier est un modèle de conception modélisé par un diagramme de package qui se compose de trois parties (package model, package vue, et package contrôleur).



# CONCLUSION GENERALE

Le projet qui est autour de « la mise en place d'un système refonte pour La gestion des visites maritimes et cargaison et La gestion automatisée des terminaux à conteneur.

Ainsi, les organismes spécialisés dans l'exploitation des ports au Maroc, conscients de l'importance majeure que détient leur commerce dans le développement économique du pays, s'adressent à des boîtes de conseil spécialisées afin de bénéficier de leurs services et ainsi de maintenir en éveil leur pointe technologique dans la gestion portuaire.

Ma mission principale dans ce projet était de faire une étude d'un module de ce projet (Cycle Camion) qui va permettre toute la gestion des entrées et sorties des camion au terminal ainsi la qualité de gestion, des ressources l'autre partie de ma mission était d'analyser les besoins du client en rédigeant un document où on va spécifier les besoins fonctionnels ainsi que les besoins techniques.

Ce stage a été rentable pour moi tant sur le plan professionnel que personnel. Mes compétences techniques, analytiques et conceptuelles ont été bien développées à travers l'utilisation de plusieurs nouvelles technologies à savoir Adobe (langage actionScript) ainsi l'architecture qui est puissante, aussi bien analyser le processus fonctionnel et métier, ainsi de bien estimer son reste à faire .bref c'était une opportunité pour voir en gros le monde Java et adobe.

Dans le cadre professionnel, ce stage a été une occasion pour améliorer l'écoute active, collaboration en groupe, participation dans des réunions ainsi que l'amélioration d'esprit d'équipe.

# REFERENCES

- <http://docs.spring.io/autorepo/docs/spring/3.2.x/spring.../mvc.html>
- <http://www.developpez.com/>  
<http://stackoverflow.com/questions/7025425/spring-mvc-tutorial-from-the-scratch>
- <https://forums.adobe.com/community/flex>
- [www.adobe.com/devnet/flex.html](http://www.adobe.com/devnet/flex.html)
- <https://www.youtube.com/watch?v=WxHLwot7GbE>
- [www.adobe.com/devnet/flex/articles/introducing\\_cairngorm.html](http://www.adobe.com/devnet/flex/articles/introducing_cairngorm.html)
- <https://sourceforge.net/adobe/cairngorm/.../GettingStartedWithCair...>
- [umf.asso.fr](http://umf.asso.fr) > [UMF](#) > [Les métiers](#)

# Annexes

## ANNEXE A : OUTILLAGES DETAILLES

### Framework Spring



Spring est un Framework Open Source rendant l'utilisation de JEE à la fois plus simple et plus productive. Spring s'appuie sur des concepts modernes, tels que l'inversion de contrôle ou la programmation orientée aspect, afin d'améliorer l'architecture des applications Java/J2EE en les rendant tout à la fois plus souples, plus agiles et plus facilement testables.

S'intégrant avec les grands Framework Open Source tels que Struts ou Hibernate, ainsi qu'avec les standards JEE, Spring propose un modèle d'application cohérent, complet et simple d'emploi. Recommandé par de nombreux architectes et développeurs expérimentés, Spring commence à

se diffuser au sein des SSII et des entreprises marocaines. Une bonne connaissance de ce produit est donc essentielle dans le monde très concurrentiel de l'informatique d'entreprise d'aujourd'hui.

### Framework Hibernate

#### Définition :

Hibernate est un ORM ou Framework de persistance libre (open source) gérant la persistance des objets Java/J2EE en base de données relationnelle. Il peut être utilisé dans la cadre d'un développement client lourd, que dans un environnement léger de type Apache Tomcat ou dans un environnement JEE complet.

Depuis la version 3.6 (sortie en 2010), Hibernate a implémenté le standard de persistance EJB 3.0 **Java Persistence API « JPA »**.

JPA est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java.

La Java Persistence API est à l'origine issue du travail du groupe d'experts JSR 220.

La persistance dans ce contexte recouvre 3 zones :

- l'API elle-même, définie dans le paquetage javax.persistence ;
- le langage Java PersistenceQuery (JPQL) ;
- l'objet/les métadonnées relationnelles.

Le Framework Hibernate se compose de 3 modules, le module qui m'intéresse est celui qui intègre JPA à savoir *Hibernate Entity Manager*. Ce module implémente quelques spécifications JPA, à savoir :

- Les interfaces de programmation ;
- Les règles de cycle de vie des objets persistants ;

- Les fonctionnalités d'interrogation.

## Architecture :

L'architecture du noyau de Hibernate simplifie le fait d'intégrer JPA ou d'autre gestionnaire de persistance (ex. JDBC, JNDI, ...) :

