

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane MIRA de Bejaia



Faculté des Sciences Exactes
Département d'Informatique

Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master II Professionnel en
Informatique

Option : Administration et Sécurité des Réseaux Informatiques

Thème :

Conception et réalisation d'une application web

Cas : GESTION DU PARC DU GROUPE IFRI

Réalisé par :

M^{elle} BOUZIDI Ibtissam

M^{elle} SLIMANI Karima

Soutenu devant le jury composé de :

Président : *M^r* TARI Abdelkamel

Encadreur : *M^r* FARAH Zoubeyr

Examineur : *M^r* SEBAA Abderrazak

Examineur : *M^r* MIR Foudil

Université de Béjaïa

Université de Béjaïa

Université de Béjaïa

Université de Béjaïa

Juin 2013

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous tenons à remercier notre encadreur Mr FARH ainsi que Mr SABAA pour l'orientation, la confiance et la patience sans lesquels ce travail n'aurait pas pu être mené au bon port.

Nos remerciements s'étendent également à Mr MEZIANI Nabil de l'entreprise d'Ifri pour ses bonnes explications qui nous ont éclairé le chemin de la recherche et sa collaboration avec nous dans l'accomplissement de ce modeste travail.

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Nos vifs remerciements vont également aux membres du jury, à leurs têtes, Mr TARI, pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

On remercie également tous ceux qui se sont montrés compréhensifs à l'égard de notre objectif.

Dédicaces

Louanges à ALLAH qui nous a guidé sur le droit chemin tout au long du travail et nous a espéré les bons pas et les juste reflexes. Je dédie ce mémoire

A la personne la plus chère à mon cœur, ma Mère, qui a toujours cru en moi et encouragé.

A la mémoire de mon cher père symbole de sacrifice et de tendresse, qui a été mon ombre durant toutes mes années d'études, qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger.

A mon cher frère...

A tous ceux qui me sont chères.

Karima

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire « El hamdou lillah »

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, à ma Mère. A mon Père qui s'est sacrifié pour mon bonheur et ma réussite.

A la mémoire de mes grands parents paternels. A mes chers grands parents maternels.

A mon adorable sœur, à mon frère. A toutes mes tantes et oncles, cousins et cousines. A toute ma grande famille.

A mes chères amies.

A tous ceux qui me sont chères.

A tous ceux qui m'aiment. A tous ceux que j'aime.

Ibtissam

Table des matières

Table de matière	IV
Liste des figures	VIII
Liste des tableaux	X
Liste des abréviations	XI
Introduction générale	1
<i>Partie 1 : Généralités sur les services web et étude de l'existant</i>	
Chapitre I : Généralités sur les services web	
I.1. Introduction	3
I.2. Définition des services web	3
I.3. Historique des services web	4
I.4. La différence entre un site web et une application web	4
I.5. Architecture des services web	5
I.5.1. Le protocole SOAP	5
I.5.2. Le langage WSDL	6
I.5.3. L'annuaire UDDI	7
I.5.4. Le langage XML	8
I.5.4.1. Mise en page de XML	8
I.5.4.2. Structure des documents XML	8
I.5.4.3. Décodage d'un document XML	9
I.5.4.4. Les avantages de XML	9
I.6. fonctionnement des services web	9
I.6.1. Description en couches des services web	10
I.7. Les modèles de services web.....	10
I.7.1. Les services web sémantiques	10
I.7.2. Composition des services web	11
I.8. Buts des services web.....	11
I.9. Avantages des services web	11
I.10. Inconvénients des services web	12
I.11. La sécurité des services web	12
I.11.1. Les modèles de sécurité des services web.....	12
I.12. Conclusion.....	13
Chapitre II : Etude de l'existant	
II.1. Introduction.....	15
II.2. Présentation générale de l'entreprise d'accueil	15

II.2.1. Aperçu historique.....	15
II.2.2. Définition de la SARL IBRAHIM & fils	16
II.2.2.1. Structure de l'entreprise	16
II.2.2.1.1. Direction générale	18
II.2.2.1.2. Secrétariat	18
II.2.2.1.3. Service informatique	18
II.2.2.1.4. Service qualité.....	18
II.2.2.1.5. Service sécurité	18
II.2.2.1.6. Direction technique.....	19
II.2.2.1.7. Direction financière et comptabilité.....	19
II.2.2.1.8. Direction des achats	19
II.2.2.1.9. Direction commerciale	19
II.2.2.1.10. Direction des ressources humaines.....	19
II.2.2.1.11. Service production.....	20
II.2.2.1.12. Direction des moyens généraux	20
II.2.2.1.13. Direction logistique.....	20
II.3. Récolte des données.....	20
II.3.1. Données récoltées auprès du service commercial	20
II.3.2. Données récoltées auprès du service production.....	22
II.3.3. Données récoltées auprès du service parc	24
II.4. Conclusion	26

Partie 2 : Conception et mise en œuvre

Chapitre III : Démarche adoptée

III.1. Introduction.....	28
III.2. Présentation du processus du développement logiciel.....	28
III.2.1. Le processus unifié.....	28
III.2.2. Le processus 2TUP.....	29
III.2.3. Un processus de modélisation avec UML.....	31
III.3. Conclusion.....	32

Chapitre IV : Etude préliminaire

IV.1. Introduction.....	34
IV.2. Recueil des besoins fonctionnels.....	34
IV.3. Description du contexte.....	34
IV.3.1. Identification des acteurs.....	34
IV.3.2. Identification des messages	35
IV.3.3. Modélisation du contexte.....	35
IV.4. Conclusion	36

Chapitre V : Capture des besoins fonctionnels

V.1. Introduction.....	38
V.2. Identification des cas d'utilisation.....	38
V.3. Description des cas d'utilisation	41
V.3.1. Package « Gestion camion ».....	41
V.3.2. Package « Gestion parc ».....	44
V.4. Identification des classes candidates	47
V.5. Conclusion.....	48

Chapitre VI : Capture des besoins techniques

VI.1. Introduction.....	50
VI.2. Capture des spécifications techniques.....	50
VI.2.1. Architecture 3-tiers.....	51
VI.2.2. Architecture de l'application web.....	52
VI.3. Capture des spécifications logicielles	53
VI.3.1. Les exploitants du système.....	54
VI.3.2. Les cas d'utilisation techniques.....	54
VI.3.3. Description sommaire des cas d'utilisation techniques.....	54
VI.4. Conclusion	57

Chapitre VII : Analyse

VII.1. Introduction.....	59
VII.2. Découpage des classes en catégories	59
VII.3. Dépendances entre catégories	60
VII.4. Développement du modèle statique.....	60
VII.4.1. Identification des classes, associations et attributs.....	61
VII.4.2. Conception des modules.....	62
VII.5. Développement du modèle dynamique.....	63
VII.6. Conclusion	67

Chapitre VIII : Conception

VIII.1. Introduction.....	69
VIII.2. Conception générique.....	69
VIII.2.1. Développement du modèle logique de conception	69
VIII.2.2. Définition du Framework technique	69
VIII.2.3. Description des couches.....	68
VIII.3. Conception préliminaire	71
VIII.3.1. Développement du modèle de déploiement.....	72
VIII.3.2. Développement du modèle d'exploitation.....	72
VIII.4. Conception détaillée	73
VIII.4.1. Concevoir les classes.....	73

VIII.4.2. Concevoir les opérations.....	75
VIII.4.3. Passage au modèle relationnel.....	75
VIII.5. Conclusion.....	76

Chapitre IX : Réalisation du prototype

IX.1. Introduction.....	78
IX.2. Architecture de l'application	78
IX.3. Présentation des outils de développement	80
IX.3.1. Présentation Java.....	80
IX.3.2. Présentation de la plateforme de développement J2EE	80
IX.3.2.1. Présentation JSP.....	80
IX.3.2.2. Présentation Servlet	80
IX.3.2.3. Présentation JavaScript.....	80
IX.3.4. Présentation GlassFish	81
IX.3.5. Description de MySql	81
IX.4. Sécurité du système	81
IX.4.1. Principales menaces de la sécurité des systèmes d'information.....	81
IX.4.2. Solutions élaborées pour la sécurité conçus.....	82
IX.5. Présentation du prototype réalisé	82
IX.6. Conclusion.....	85

Conclusion générale	87
Bibliographie	89
Webographie	89
Annexe	91

Liste des figures

Figure 1 : Architecture des services web	3
Figure 2 : Structure d'un message SOAP.....	3
Figure 3 : Structure d'un document WSDL.....	4
Figure 4 : Schéma général de l'annuaire UDDI	4
Figure 5 : Organigramme de la SARL IBRAHIM & fils.....	17
Figure 6 : Chariot élévateur.....	23
Figure 7 : Palette en bois.....	23
Figure 8 : Palette en plastique.....	23
Figure 9 : Intercalaires en carton.....	24
Figure 10 : Le système d'information soumis à deux types de contraintes.....	30
Figure 11 : Le processus de développement en Y.....	30
Figure 12 : Diagramme du contexte dynamique.....	36
Figure 13 : Les deux packages fonctionnels du système.....	39
Figure 14 : Diagramme de cas d'utilisation du package « Gestion parc »	39
Figure 15 : Diagramme de cas d'utilisation du package « Gestion camion »	40
Figure 16 : Diagramme de cas d'utilisation général	40
Figure 17 : Diagramme d'activité du cas d'utilisation « Gérer escale »	42
Figure 18 : Diagramme d'activité du cas d'utilisation « Consulter escale ».....	43
Figure 19 : Diagramme d'activité du cas d'utilisation « Ajouter camion ».....	44
Figure 20 : Diagramme d'activité du cas d'utilisation « Chercher camion »	45
Figure 21 : Diagramme d'activité du cas d'utilisation « Enregistrer entrée/sortie ».....	46
Figure 22 : Diagramme d'activité du cas d'utilisation « Etat du parc ».....	47
Figure 23 : Diagramme de classes participantes du package « Gestion camion ».....	47
Figure 24 : Diagramme de classes participantes du package « Gestion parc ».....	48
Figure 25 : Réseau globale du groupe Ifri.....	51
Figure 26 : La répartition des différentes couches de l'architecture 3-tiers	52
Figure 27 : Schéma des composants matériels du système	53
Figure 28 : Modèle de spécification logicielle	54
Figure 29 : Diagramme d'activité du cas d'utilisation « manipuler des objets ».....	55
Figure 30 : Diagramme d'activité du cas d'utilisation « Gérer la sécurité ».....	55
Figure 31 : Diagramme d'activité de « Utiliser l'aide »	56
Figure 32 : Diagramme d'activité de « Gérer les erreurs ».....	56
Figure 33 : Représentation graphique d'une catégorie	59
Figure 34 : Schéma présentatif des catégories de classes candidatent.....	60
Figure 35 : Diagramme de package d'analyse	60
Figure 36 : Diagramme de classes associées au module « Escale ».....	62
Figure 37 : Diagramme de classes associées au module « Opération »	62
Figure 38 : Diagramme de séquence du CU « S'authentifier ».....	63
Figure 39 : Diagramme de séquence du CU « Consulter escale ».....	64

Figure 40 : Diagramme de séquence du CU « Gérer escale ».....	65
Figure 41 : Diagramme de séquence du CU «Ajouter camion ».....	66
Figure 42 : Diagramme de séquence du CU « Enregistrer Entrée/Sortie ».....	67
Figure 43 : Diagramme de séquence du CU « Consulter l'état du parc ».....	67
Figure 44 : Organisation du modèle logique de conception	70
Figure 45 : Diagramme de déploiement	72
Figure 46 : Modèle d'exploitation.....	72
Figure 47 : Diagramme de classes général	73
Figure 48 : Architecture de base d'une application web avec la technologie JSF	79
Figure 49 : Fonctionnement d'une page JSF.....	79
Figure 50 : Page d'accueil de notre application	83
Figure 51 : Page d'authentification de notre application.....	83
Figure 52 : Page d'administration de notre application.....	84
Figure 53 : Page de gestion de la rubrique « Transition »	84
Figure 54 : Page d'ajout d'une escale.....	85

Liste des tableaux

Tableau 1 : Structure du bon de commande.....	21
Tableau 2 : Structure de la facture.....	22
Tableau 3 : Structure de la facture de consignation.....	22
Tableau 4 : Structure Capacité des palettes.....	24
Tableau 5 : Ordre de mission.....	25
Tableau 6 : Les acteurs de notre système.....	35
Tableau 7 : Description du cas d'utilisation « Gérer escale ».....	41
Tableau 8 : Description du cas d'utilisation « Consulter escale ».....	42
Tableau 9 : Description du cas d'utilisation « Gérer camion ».....	43
Tableau 10 : Description du cas d'utilisation « Chercher camion ».....	44
Tableau 11 : Description du cas d'utilisation « Enregistrer une entrée/sortie ».....	45
Tableau 12 : Description du cas d'utilisation « Consulter état du parc ».....	46
Tableau 13 : Inventaire du matériel informatique de l'entreprise.....	51
Tableau 14 : Classes d'objets et leurs attributs.....	62
Tableau 15 : Rôles de la couche présentation.....	70
Tableau 16 : Rôles de la couche application	70
Tableau 17 : Rôles de la couche service	71
Tableau 18 : Rôles de la couche DAO.....	71
Tableau 19 : Liste des classes	74
Tableau 20 : Liste des opérations.....	75

Liste des abréviations

Symbole	Designation
<i>CORBA</i>	Common Object Request Broker Architecture
<i>CSS</i>	Cascading StyleSheet
<i>DAML</i>	DARPA Agent Markup Language
<i>DCOM</i>	Distributed Component Object Model
<i>DHTML</i>	Dynamic HyperText Markup Language
<i>DTD</i>	Document Type Definition
<i>EDI</i>	Electronic Data Interchange
<i>GPRS</i>	General Packet Radio Service
<i>HTML</i>	HyperText Mark-Up Language
<i>HTTP</i>	HyperText Transfer Protocol
<i>J2EE</i>	Java 2 Enterprise Edition
<i>JSF</i>	Java Server Faces
<i>LAN</i>	Local Area Network
<i>REST</i>	REpresentational State Transfer
<i>RMI</i>	Revenu Minimum d'Insertion
<i>RPC</i>	Remote Procedure Call
<i>RUP</i>	Rational Unified Process
<i>RSS</i>	Really Simple Syndication
<i>SAML</i>	Security assertion markup language
<i>SARL</i>	Société à responsabilité limitée
<i>SGML</i>	Standard Generalized Markup Language
<i>SOAP</i>	Simple object Access Protocol
<i>SVG</i>	Scalable Vector Graphics
<i>TCP</i>	Transmission Control Protocol
<i>TTC</i>	Toronto Transit Commission
<i>TVA</i>	Taxe sur la valeur ajoutée
<i>UDDI</i>	Universal Description, Discovery and Integration
<i>UML</i>	Unified Modeling Language
<i>URI</i>	Uniform Resource Identifier
<i>URL</i>	Uniform Resource Locator
<i>VPN</i>	Virtual Private Network
<i>W3C</i>	World Wide Web Consortium
<i>WSDL</i>	Web Services Description Language
<i>XACML</i>	eXtensible Access Control Markup Language
<i>XHTML</i>	Extensible HyperText Markup Language
<i>XML</i>	eXtensible Markup Language
<i>XP</i>	eXtreme Programming
<i>XSL</i>	eXtensible StyleSheet Language
<i>XSLT</i>	eXtensible StyleSheet Language Transformation

Introduction générale

La pérennité de toute entreprise passe, entre autre, par une disponibilité permanente de son système d'information. L'information nécessaire au bon fonctionnement de l'entreprise englobe aussi bien les données stratégiques que les données de tous les jours.

La continuité de l'activité de l'entreprise appelle celle de son système d'information. Les responsables d'entreprises d'aujourd'hui savent que la gestion efficace et l'exploitation d'information sont la clé du succès des affaires et sont un moyen indispensable pour créer un avantage concurrentiel.

En ce qui concerne la SARL Ifri, et plus exactement le département transport, les gérants de l'entreprise ont fait appel à leur expérience et ont opté pour une stratégie bien précise pour accroître la rapidité système de transport.

Notre travail porte sur la conception et le développement d'une application web, qui en outre se voulait porteuse de solutions et de facilité dans la gestion du parc au sein du groupe Ifri.

Afin de répondre aux objectifs attendus par les clients, notre travail sera décomposé en deux parties :

Partie 1 : Généralités sur les services web et l'étude de l'existant

Cette partie comporte deux chapitres :

Chapitre I : Généralités sur les services web

Chapitre II : Etude de l'existant

Partie 2 : Conception et mise en œuvre

Cette partie comporte sept chapitres :

Chapitre III : Démarche adopté

Chapitre IV : Etude préliminaire

Chapitre V : Capture des besoins fonctionnels

Chapitre VI : Capture des besoins techniques

Chapitre VII : Analyse

Chapitre VIII : Conception

Chapitre IX : Réalisation du prototype

Chapitre I : Généralités sur les services web

I.1. Introduction

L'accès aux systèmes d'information est de plus en plus important, il doit se faire dans les meilleures conditions et cela tout en s'appuyant sur les technologies Internet. Ainsi, la distribution de l'information entre clients, partenaires et leurs différentes plates formes doit être la plus rapide possible, cela est fait au moyen des services web.

Nous allons faire dans ce chapitre un point sur les services web. On commencera à donner quelques notions de base tout en faisant l'accent sur la différence entre les applications et les sites web. Ensuite, on entamera la phase d'analyse des services web en donnant leur architecture, leur fonctionnement et leurs différents types. Par la suite, une section est mise en place pour donner le but de l'utilisation des services web tout en séparant les avantages et les inconvénients. Enfin, on discutera la sécurité de ces services.

I.2. Définition des services web

Plusieurs définitions de service web ont été mises en avant :

Citation : W3C

« Un service Web est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet. »

Citation : Dico de net

« Une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquels elles reposent. »

Citation : Wikipédia

« Un service Web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel. »

On peut dire alors qu'un service web est un mécanisme de communication entre applications distantes pouvant être de différentes plates-formes et de différents langages de programmation et cela via Internet.

I.3. Historique des services web [FAUD, 08]

Comme les organisations souhaitaient pouvoir communiquer, échanger des informations et accéder à des ressources partagées le plus simplement possible. Alors ici leur système informatique n'est non plus isolé avec les autres.

En 1975, l'EDI qui est un format standard permettait à des personnes d'envoyer des messages via câbles à des interlocuteurs de différents emplacements. Malgré que cette technique avait l'avantage en termes de gain de temps et diminution de taux d'erreurs lors de la saisie, mais elle reste comme même incomplète, et cela à cause de la difficulté de la mise en place de son système et la complexité des techniques employées.

Vers la fin des années 80, les besoins augmentaient et la nécessité de procéder à un véritable traitement distributif apparaît. Plusieurs technologies telles que RMI, DCOM et CORBA sont adoptés mais elles ont généralement échoué en raison de la diversité des plates-formes utilisées par les différentes entités, en ajoutant à ça les problèmes d'adaptation à Internet. Ces technologies utilisées par les applications réparties impliquent un couplage fort entre les objets.

Avec les années 90, et avec l'utilisation des technologies standards du web telles http et XML, les services web réussissent à donner des solutions permettant néanmoins des couplages moins forts, de faciliter le développement des applications réparties sur internet et cela favorise l'utilisation des services web.

I.4. La différence entre un site web et une application web [BERT]

Une application web est une application construite à base des technologies du web. Elle est très différente des sites web, car, elle n'a pas une finalité de communication, ni bouquet de service, elle vise à être l'outil de travail des entités métiers d'un organisme.

Exigences premières :

Le caractère intuitif d'un site web est l'exigence première, la lisibilité et la simplicité de l'interface sont privilégiées pour créer un bon site web.

Par contre, une application exige l'efficacité. L'efficacité ici désigne le temps et les efforts que demande une tâche donnée.

Processus du travail :

Les processus que l'on rencontre dans un site web sont longs et complexes mais divisé en plusieurs étapes plus simples. Au contrario à une application web, les processus du travail sont plus complexes et plus nombreux.

Objectifs :

Un site web a souvent un objectif de séduction à la différence d'une application web qui vise le confort de l'utilisateur.

« La séduction est inutile mais l'esthétique est important »

I.5. Architecture de service web [W1]

Comme on a dit précédemment, les services web reposent sur les principes du Web pour les interactions entre machines. Ces technologies utilisées par les services web sont HTTP, SOAP, WSDL, UDDI et XML-RPC.

Voici l'architecture d'un service web :

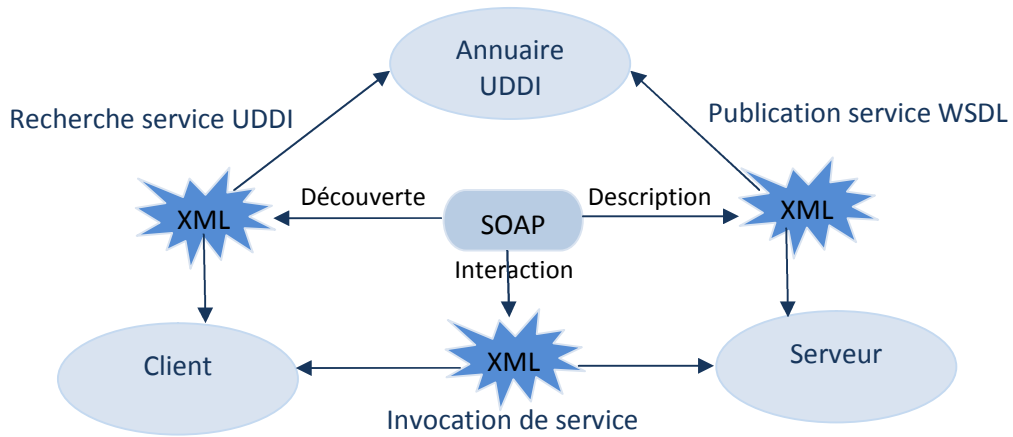


Figure 1 : Architecture des services web

I.5.1. Le protocole SOAP [W2]

SOAP (*Simple object Access Protocol*) est un protocole standard de communication. C'est l'épine dorsale du système d'interopérabilité. SOAP est un protocole décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes. Il circule sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance.

La grammaire de SOAP est assez simple à comprendre et la structure d'un message SOAP est la suivante :

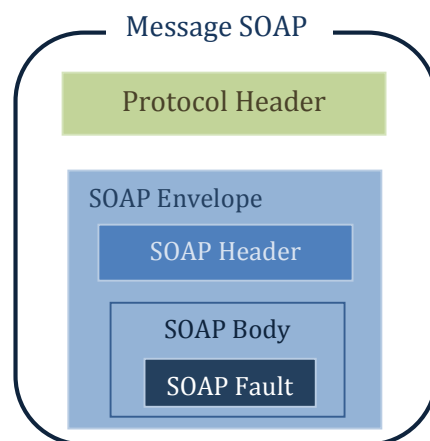


Figure 2 : Structure d'un message SOAP

- SOAP Envelope : contient la spécification des espaces de désignation et du codage de données.

- SOAP Header : c'est une partie facultative qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisée. L'entête est utile surtout, quand le message doit être traité par plusieurs intermédiaires.
- SOAP Body : est un *container* pour les informations mandataires à l'intention du récepteur du message, il contient les méthodes et les paramètres qui seront exécutés par le destinataire final.
- SOAP Fault : est un élément facultatif défini dans le corps SOAP et qui est utilisé pour reporter les erreurs.

I.5.2. Le langage WSDL [W3]

WSDL (*Web Services Description Language*) est un langage de description standard. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser le service Web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées.

Le schéma suivant décrit la structure du document WSDL :

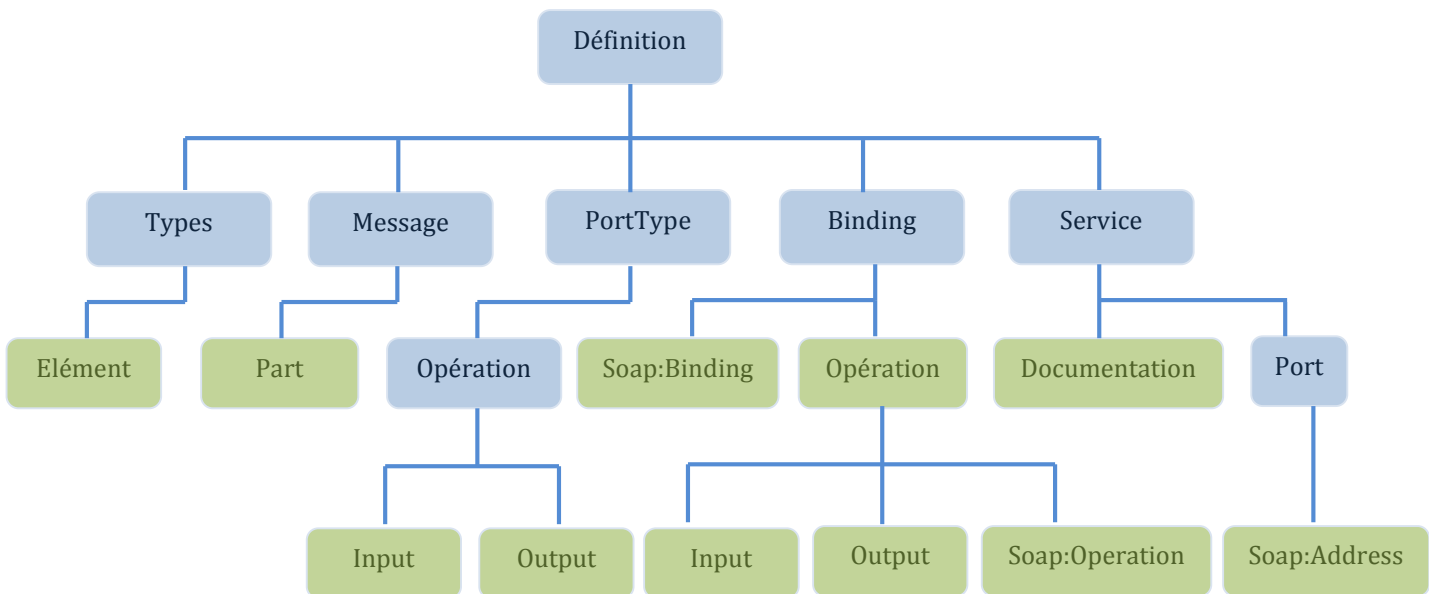


Figure 3 : Structure d'un document WSDL

Alors, les sept éléments du WSDL sont les suivants :

- Types : Contient la définition de types de données utilisés pour décrire les messages échangés.
- Messages : représente une définition abstraite des données en cours de transmission.
- PortTypes : décrit un ensemble d'opérations.

- Binding : spécifie une liaison entre un PortType et un protocole concret (SOAP, HTTP...).
- Service : indique les adresses de port de chaque liaison.
- Port : représente un point d'accès de services défini par une adresse réseau et une liaison.
- Opération : c'est la description d'une action exposée dans le port.

Le document est divisé en deux parties, chaque partie contient des:

- Définitions abstraites : composé des éléments orientés vers clients pour le service physique.
- Descriptions concrètes : composé des éléments qui sont orientés vers la description des capacités du service web.

I.5.3. L'annuaire UDDI [W4]

UDDI (*Universal Description, Discovery and Integration*) est un standard pour la publication et la découverte des services Web. UDDI se concentre sur le processus de découverte de SOA, il utilise des technologies standards telles que XML, SOAP et WSDL afin de simplifier la collaboration entre partenaires dans le cadre des échanges commerciaux.

Les informations qu'il contient peuvent être séparées en trois types :

- les pages blanches : comprennent la liste des entreprises ainsi que les informations associées à celles ci.
- les pages jaunes : identifient les différents services web de chacune des entreprises.
- les pages vertes : donnent les informations techniques des services web fournis.

Le processus de publication et de découverte des services web peut être illustré dans le schéma suivant :

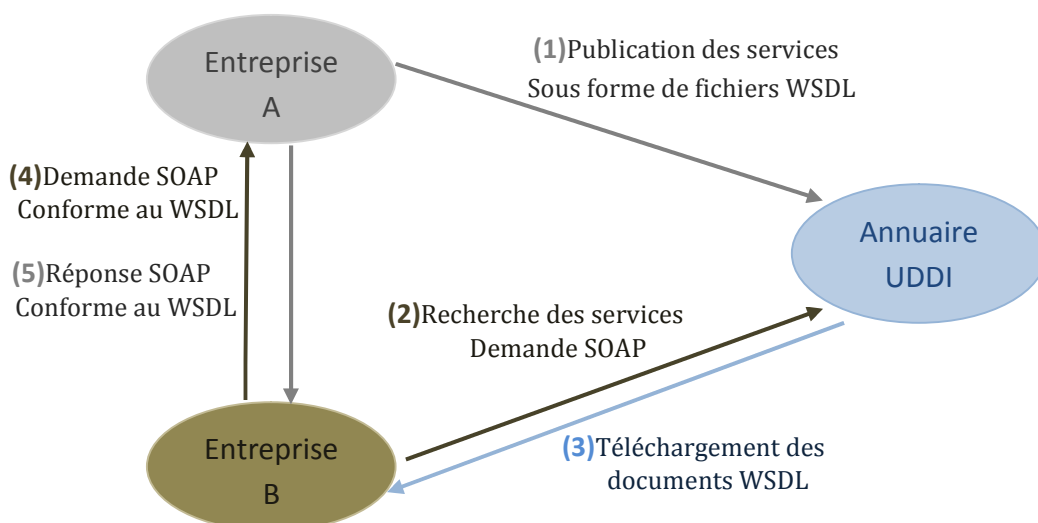


Figure 4 : Schéma général de l'annuaire UDDI

I.5.4. Le langage XML [GAST, 05]

L'Extensible Markup Language (langage de balisage extensible) est un langage informatique de balisage générique qui dérive du SGML.

SGML est un langage de balisage, employé dans les industries de la documentation et de l'édition.

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, SVG... Elle est reconnaissable par son usage des chevrons (< >) encadrant les balises.

XML a été mis au point par le XML Working Group sous l'égide du World Wide Web Consortium (W3C) dès 1996. Depuis le 10 février 1998, XML est un sous ensemble de SGML (Standard Generalized Markup Language), défini par le standard ISO8879 en 1986, utilisé dans le milieu de la Gestion Electronique Documentaire (GED). XML reprend la majeure partie des fonctionnalités de SGML, il s'agit donc d'une simplification de SGML afin de le rendre utilisable sur le web.

L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche...) entre systèmes d'informations hétérogènes (interopérabilité).

I.5.4.1. Mise en page de XML

La mise en page des données est assurée par un langage de mise en page tiers. Il existe trois solutions pour mettre en forme un document XML :

- ✓ **CSS (Cascading StyleSheet)** : La solution la plus utilisée actuellement, étant donné qu'il s'agit d'un standard qui a déjà fait ses preuves avec HTML.
- ✓ **XSL (eXtensible StyleSheet Language)** : Un langage de feuilles de style extensible développé spécialement pour XML. Toutefois, ce nouveau langage n'est pas reconnu pour l'instant comme un standard officiel.
- ✓ **XSLT (eXtensible StyleSheet Language Transformation)** : Permettant de transformer un document XML en document HTML accompagné de feuilles de style.

I.5.4.2. Structure des documents XML

XML fournit un moyen de vérifier la syntaxe d'un document grâce aux DTD (Document Type Definition). Il s'agit d'un fichier décrivant la structure des documents y faisant référence grâce à un langage adapté. Ainsi un document XML doit suivre scrupuleusement les conventions de notation XML et peut éventuellement faire référence à une DTD décrivant l'imbrication des éléments possibles.

- ✦ **Document XML bien formé** : Un document XML est dit bien formé s'il respecte les règles suivantes :

- ◆ Prologue : Le prologue du document, indiquant quelles conventions respecte le document, est obligatoire. Ces conventions disent quelle version de XML est utilisée, ainsi que le jeu de caractères (en anglais encoding) est utilisé dans le document.
- ◆ Imbrication des balises : L'imbrication des éléments suit une forme particulière : elle est arborescente. En effet, XML impose deux règles qui font que tout document peut être représenté sous la forme d'un arbre.
 - Il existe un seul attribut de plus haut niveau.
 - Il est interdit de faire se chevaucher des balises.
- ◆ **Document XML valide** : Un document XML est valide s'il est bien formé et si toutes ses balises appartiennent à une DTD.

I.5.4.3. Décodage d'un document XML

XML permet donc de définir un format d'échange selon les besoins de l'utilisateur et offre des mécanismes pour vérifier la validité du document produit. Il est donc essentiel pour le receveur d'un document XML de pouvoir extraire les données du document. Cette opération est possible à l'aide d'un outil appelé analyseur (parseur).

Le parseur permet d'une part d'extraire les données d'un document XML (on parle d'analyse du document ou de parsing) ainsi que de vérifier éventuellement la validité du document.

I.5.4.4. Les avantages de XML

Voici les principaux atouts de XML :

- ◆ La lisibilité : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML.
- ◆ Auto descriptif et extensible
- ◆ Une structure arborescente : permettant de modéliser la majorité des problèmes informatiques.
- ◆ Universalité et portabilité : les différents jeux de caractères sont pris en compte.
- ◆ Déplorable : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme http.
- ◆ Intégrabilité : un document XML est utilisable par toute application pourvue d'un parseur (c'est-à-dire un logiciel permettant d'analyser un code XML).
- ◆ Extensibilité : un document XML doit pouvoir être utilisable dans tous les domaines d'applications
- ◆ XML est particulièrement adapté à l'échange de données et de documents.

I.6. Fonctionnement des services web [W5]

Le fonctionnement des services web s'articule autour de trois acteurs principaux qui sont les suivants :

- Service provider service : Le fournisseur de service met en application le service Web et le rend disponible sur Internet.
- Service requester programme client : C'est n'importe quel consommateur du service Web. Le demandeur utilise un service Web existant en ouvrant une connexion réseau et en envoyant une demande en XML (REST, XML-RPC, SOAP).
- Annuaire service registry : Le registre de service est un annuaire de services. Le registre fournit un endroit central où les programmeurs peuvent publier de nouveaux services ou en trouver. Les interactions entre ces trois acteurs suivent plusieurs étapes :
 - ✦ La publication du service : le fournisseur diffuse les descriptions de ses services Web dans l'annuaire.
 - ✦ La recherche du service : le client cherche un service particulier, il s'adresse à un annuaire qui va lui fournir les descriptions et les URL des services demandés afin de lui permettre de les invoquer.
 - ✦ L'invocation du service : une fois que le client récupère l'URL et la description du service, il les utilise pour l'invoquer auprès du fournisseur de services.

I.6. 1. Description en couche des services Web

Cette structure est formée de quatre couches majeures :

Découverte de services	UDDI
Description de services	WSDL
Communication	SOAP
Transport	HTTP

- Le transport de messages XML-RPC ou SOAP est assuré par le standard HTTP.
- SOAP ou XML-RPC prévoit la couche de communication basée sur XML pour accéder à des services Web.
- La description d'un service Web se fait en utilisant le langage WSDL. WSDL expose l'interface du service.
- La publication et la découverte des services Web sont assurées par le biais du référentiel UDDI. Un référentiel UDDI est un catalogue de services Web.

I.7. Les modèles de service web [W6]

I.7.1. Les Services Web Sémantiques

Le mécanisme de découverte offert par l'UDDI est limité, on ne peut pas faire des recherches selon les fonctionnalités et les caractéristiques des services, cela est dû au manque d'informations sémantiques dans le fichier WSDL. Pour surmonter ce problème une nouvelle génération de service dite services web sémantiques a été proposée.

Les services web sémantiques visent à faire une combinaison entre le web sémantique et la technologie des services web, en développant des descriptions sémantiques des services web.

Dans cette optique, plusieurs solutions ont été proposées entre autres l'ontologie DAML-S. L'ontologie DAML-S, est constituée de quatre éléments principaux qui sont:

- ✦ La classe ServiceProfile : Qui décrit le service en fonction de ce qu'il fait, elle englobe une description du service (nom,...) et de son fournisseur (nom, adresse physique,...), une description du comportement fonctionnel du service (les entrées/sorties) et une description des attributs fonctionnels du service.
- ✦ La classe ServiceModel : Explique comment le service fonctionne.
- ✦ La classe ServiceGrounding : Montre comment les entrées/sorties d'un service doivent être réalisées concrètement comme messages.
- ✦ Les ressources : Les ressources nécessaires pour l'exécution d'un service sont définies dans une ontologie.

I. 7.2. Composition de services web

La composition de services web est considérée comme une révolution qui permet la distribution, sur le web, non seulement des données et des documents mais aussi des applications. Un service web est une interface qui décrit une collection d'opérations qui sont accessibles à travers des messages XML.

La composition de service web ne consiste pas à composer les services appartenant à un seul partenaire mais plutôt d'aller plus loin et chercher les services des autres partenaires. Ce nouveau paradigme se focalise sur la composition des services complexes à partir des services élémentaires se trouvant sur Internet. Ainsi, les services Web composés constituent différents composants mis en application à différents endroits. Ces services s'exécutent dans différents contextes, cependant ils ont besoin de communiquer pour avoir le comportement désiré. En effet, La composition peut être statique ; cela veut dire que les services interagissent entre eux d'une manière prédéfinie. Elle peut être aussi dynamique; c'est à dire que les services sont découverts les uns des autres.

I.8. But des services web

Comme le but principal des services Web est de permettre à des applications de différentes plates-formes ou compilées avec différents langages, de communiquer ensemble, un langage commun doit être utilisé pour assurer l'interopérabilité : XML est le meilleur candidat pour cela, étant donné sa diffusion et le grand nombre de bibliothèques qui permettent de le manipuler.

I.9. Avantage des services web

- ✦ Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- ✦ Les services Web utilisent des standards et protocoles ouverts.
- ✦ Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.

- ✦ Basés sur le protocole HTTP, les services Web peuvent fonctionner au travers de nombreux pare-feu sans nécessiter des changements sur les règles de filtrage.
- ✦ Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants.

I.10. Inconvénient des services web

- ✦ Les normes de services Web dans certains domaines sont actuellement récentes.
- ✦ Les services Web souffrent de performances faibles comparée à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.
- ✦ Par l'utilisation du protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place au travers des pare-feu.

I.11. La sécurité des services web [W7]

Les différents aspects de la sécurité web sont les suivants :

- ✓ **Confidentialité** : La communication est privée, assurée par l'utilisation d'une connexion SSL pour le transfert des requêtes SOAP entre le client, d'éventuels intermédiaires et le serveur final.
- ✓ **Intégrité** : La requête n'est pas modifiée, elle est assurée soit par l'utilisation d'une connexion SSL ou par l'encodage de la requête à l'aide d'un certificat X.509.
- ✓ **Authentification des utilisateurs** : La première étape de la sécurisation d'une application web est l'authentification des utilisateurs qui la manipulent. En effet, contrairement à un site web, généralement ouvert à tous, les applications web nécessitent généralement que l'identité de l'utilisateur soit connue, ne serait-ce que pour permettre la personnalisation des services qui lui sont offerts.

Pour cela, il est nécessaire de mettre en place un mécanisme de login/password ou d'un certificat X.509 dans l'entête de la requête, permettant ensuite de manipuler les informations propres à l'utilisateur, ses droits, éventuellement ses groupes. Cette authentification se fait à partir d'informations stockées dans une base de données, un annuaire.

- ✓ **Autorisations** : Qui consistent à vérifier que la personne connectée a bien les permissions d'effectuer une action donnée.

I.11.1. Les modèles de sécurité des services web

Les principaux standards associés à la sécurité des services web sont :

- ✓ **Signature XML (XML Signature)** : Pour la représentation des signatures digitales dans un document XML.
- ✓ **Chiffrement XML (XML Encryption)** : Pour les mécanismes d'encryptage et de décryptage des documents XML.
- ✓ **SAML-XML (Security Assertions Markup Language)**: Spécifie une procédure d'authentification.

- ✓ **XACML (Extensible Access Control Markup Language):** Spécifie les stratégies de contrôle d'accès à travers le réseau.
- ✓ **XKMS (XML Key Management Specification):** Décrit l'infrastructure à clé publique.

I.12. Conclusion

Les services web sont des applications accessibles sur Internet réalisant chacune une tâche spécifique. Un grand avantages des services Web c'est qu'il se repose sur les standards de l'internet et notamment le protocole http.

Les services web facilitent non seulement les échanges entre les applications de l'entreprise mais surtout permettent une ouverture vers les autres entreprises.

Chapitre II : Etude de l'existant

II. 1. Introduction

Cette phase constitue un pont de passage obligatoire entre les concepteurs et les utilisateurs, elle a comme objectif de bien comprendre la dynamique et la structure de l'organisation.

A ce niveau, il faut mettre les points sur les îles et cela pour vérifier que les clients, les utilisateurs et les développeurs ont une vision commune et exacte de l'organisation.

Cette étude permet donc d'apporter un jugement correct, de collecter les informations qui semblent nécessaires et relever les dysfonctionnements du système afin de proposer des solutions et concevoir un bon produit.

II. 2. Présentation générale de l'entreprise d'accueil

Pour bien faire connaître l'entreprise, nous commençant tout d'abord de donner l'historique de la SARL IBRAHIM & fils - ifri -, ensuite nous présenterons la structure et les différentes activités de l'entreprise d'accueil et enfin quelques diagnostics sur la position de la SARL IBRAHIM & fils.

II. 2.1. Aperçu historique

La SARL IBRAHIM & fils « Ifri » est une société à caractère industriel spécialisée dans la production des eaux minérales et boissons diverses. Cette société est sise à la zone industrielle dite « Ahrik » dans la commune d'Ighzer Amokran - Daira Ifri – Ouzellaguen, wilaya de Bejaïa.

Son expérience dans le domaine agro-alimentaire remonte à 1986, ayant pour activité la production d'eaux gazeuses et sirops.

Dix ans plus tard, Le premier atelier d'embouteillage d'eau minérale en bouteilles en polyéthylène téréphtalate (PET) d'Ifri ouvre ses portes. A ce niveau, la société hérite d'un statut juridique de SNC puis de SARL.

A cette date, plus de 05 millions de bouteilles d'eau minérale sont commercialisées à l'échelle nationale. En 1999, le chiffre atteint 48 millions de bouteilles. Elles franchiront le cap des millions de litres en 2013.

Suite aux demandes des clients, leurs besoins et leurs envies, la SARL IBRAHIM & fils « Ifri » décide d'élargir en permanence sa gamme de produit tout en accroissant sa capacité de production.

Enfin, une amélioration qualitative et quantitative de la production de la société permettra d'élargir son champ d'action, de fidéliser ses clients et de s'introduire au marché de l'exportation.

Après avoir pris sa place dans le marché nationale la SARL s'est ouverte sur le marché international. Aujourd'hui, ses produits sont exportés vers plusieurs pays à savoir : France, Angleterre, Espagne, Italie, Allemagne, Belgique, Luxembourg, Soudan, Mali, Niger, Emirats arabes unis.

Par ailleurs, plusieurs contrats en phase d'être finalisés avec les partenaires étrangers qui sont soucieux d'acquérir les produits de la SARL IBRAHIM et fils « IFRI », celle-ci se fait l'objectif primaire qui est l'élargissement de son réseau d'exportation vers encore d'autres pays.

II. 2.2. Définition de la SARL IBRAHIM & fils

II. 2.2.1. Structure de l'entreprise

Pour gérer une entreprise, il faut l'organiser et diviser les tâches, répartir les responsabilités et coordonner le travail. La structure d'une organisation doit permettre de bien cerner les tâches, de faciliter le travail de chaque acteur et d'établir les différentes relations entre chaque service.

Voici l'organigramme générale de l'entreprise d'accueil :

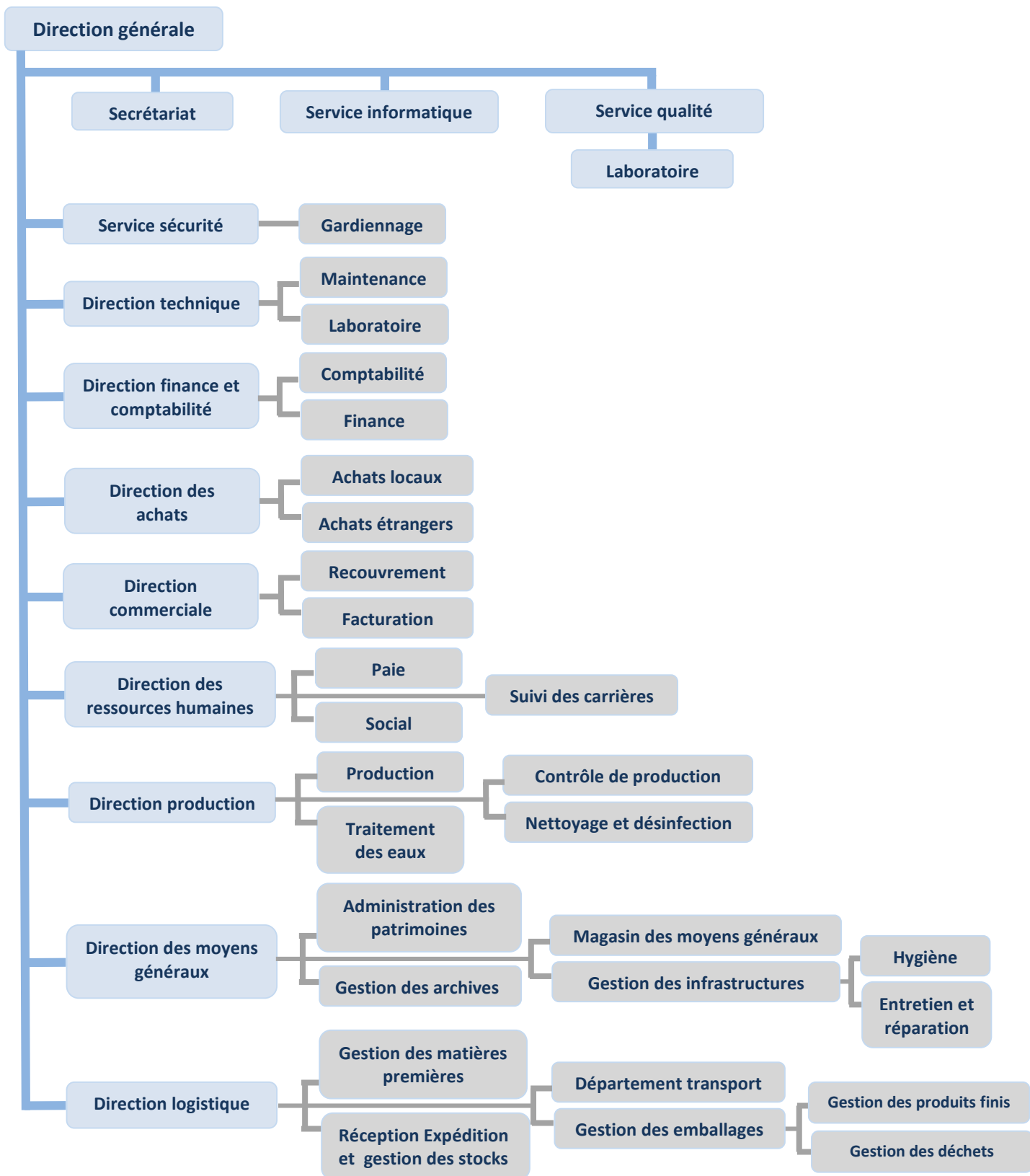


Figure 5 : Organigramme de la SARL IBRAHIM & fils

II. 2.2.1.1. Direction générale

Dirigé par un directeur général qui assure et applique les décisions prises lors des différents conseils d'administration. A l'instar de tout autre centre de décision, la direction générale d'ifri est le poumon de l'ensemble de la société ou tout se coordonne et ce décide pour tout ce qui a trait au quotidien et à la politique de gestion de l'entreprise.

II. 2.2.1.2. Secrétariat

Par sa nature d'organe d'accueil, il est chargé de :

- ✓ L'enregistrement, classement des courriers et fax, et dispatching aux divers services.
- ✓ Saisie sur micro-ordinateur.
- ✓ Orientation et réception des clients et fournisseurs par voies de communication soit verbal, écrite ou téléphonique.
- ✓ Organisation des différents rendez-vous.

II. 2.2.1.3. Service informatique

Le service informatique est indispensable non seulement pour administrer le réseau de la société et maintenir le parc informatique mais aussi pour assurer les objectifs suivants :

- ✓ Améliorer la qualité de service.
- ✓ Mettre en œuvre des systèmes d'information modernes.
- ✓ Requalifier les compétences techniques.
- ✓ Redéployer l'organisation interne.
- ✓ Mettre en œuvre des systèmes avec un fort retour sur investissement.

II. 2.2.1.4. Service qualité

Son rôle principal est :

- ✓ La mise en place des procédures de travail de chaque structure.
- ✓ La mise en place des procédures de travail de chaque structure.
- ✓ Assurer l'établissement, la mise en œuvre et l'entretien des processus nécessaires au système de la qualité.
- ✓ Représenter l'organigramme auprès des parties externes relatif au système de management de qualité.

II. 2.2.1.5. Service sécurité

Ce service est chargé de :

- ✓ Réaliser les préventions générales et intervenir en cas d'accidents.
- ✓ Effectuer la surveillance interne et externe de l'entreprise.
- ✓ Assurer le gardiennage et la surveillance de l'usine.
- ✓ Consigner les anomalies constatées sur un registre prévu à cet effet.
- ✓ Procéder, parfois, aux heures d'entrée et de sorties.
- ✓ Assurer la prévention des autorités quand ça nécessite.

II. 2.2.1.6. Direction technique

Doté de tous les moyens d'intervention et des deux sections : maintenance et le laboratoire d'analyse et préparation des sirops, elle a pour rôle la maintenance des équipements de production en :

- ✓ Veillant au bon fonctionnement des équipements de production.
- ✓ Réglant des machines.
- ✓ Assurant la maintenance et l'entretien des machines et tous les véhicules.

II. 2.2.1.7. Direction financière et comptabilité

L'activité de l'entreprise met en œuvre de nombreux acteurs internes et externes. Cette activité se traduit par des relations croisées, des flux financiers parfois complexes. En plus, des experts de comptabilité sont indispensables et c'est pour :

- ✓ Assurer la conformité des opérations comptables.
- ✓ Etablir la situation financière.
- ✓ Planifier les financements et les investissements.
- ✓ Gérer les recettes et dépenses.

II. 2.2.1.8. Direction des achats

Cette direction est muni de la section achats locaux et la section achats étrangers, où il prend en charge la gestion des achats et assure le suivi des commandes jusqu'à leurs satisfaction en assurant les délais comptables avec l'urgence des besoins et à moindre coût.

II. 2.2.1.9. Direction commerciale

Subdivisé en deux sections à savoir section facturation et section recouvrement, cette direction s'occupe de :

- ✓ Recevoir les bons des commandes des clients.
- ✓ Etablir les factures pro-formats et les ordres de versement pour les clients.
- ✓ Etablir et viser les facturations et les bons de livraison.
- ✓ Répondre à toute demande de la clientèle sur les plans de la qualité et de la quantité.
- ✓ Rapprocher le plus possible le produit du consommateur (Marketing).
- ✓ être à la disposition du consommateur pour toute réclamation ou suggestion.
- ✓ On y trouve la section vente qui s'occupe de toutes les ventes.

II. 2.2.1.10. Direction des ressources humaines

Elle met en œuvre des systèmes de gestion intégrés à la stratégie de l'entreprise qui traduisent une adéquation entre les impératifs économiques et les attentes du personnel. Elle est subdivisée en 3 grandes parties : paie, personnel, moyens généraux.

Ses missions et responsabilités :

- ✓ Mise en œuvre, réalisation et gestion des moyens humains et matériels de manière rationnelle et efficace.

- ✓ Assurer la souscrite des personnes, des moyens matériels, des équipements, des documents, des archives et du transport de fonds.
- ✓ Coordonner, assister le service et les directions en veillant au respect des procédures en vigueur.

II. 2.2.1.11. Service production

Elle est responsable du personnel et des trois ateliers de production, son rôle est :

- ✓ La gestion du carnet de bord de la production.
- ✓ Le contrôle et le suivi des statistiques de production.
- ✓ La production de l'équivalent en quantités demandées par le service commercial et en normes exigées par les laboratoires internes.

II. 2.2.1.12. Direction des moyens généraux

Son rôle principal est :

- ✓ Administration des patrimoines.
- ✓ Gestion des archives.
- ✓ Gestion des infrastructures.

II. 2.2.1.13. Direction logistique

Elle comporte trois services qui sont :

- ✓ Réception expéditions et gestion des stocks.
- ✓ Gestion des matières premières.
- ✓ Gestion des emballages.

Les principales activités de ces services sont :

- ✓ Coordonner les activités des magasiniers.
- ✓ Veiller à la bonne tenue des stocks.
- ✓ Contrôler les différents documents relatifs aux entrées et sorties de marchandise dans les divers magasins.

Département transport : C'est là où on a effectué la plus grande partie de notre stage.

Il a pour rôle :

- ✓ La coordination entre le service commercial et le service parc.
- ✓ La gestion des camions.
- ✓ La gestion des chauffeurs.

II. 3. Récolte des données

II. 3.1. Données récoltées auprès du service commercial

En ce qui concerne la récolte des données, et pour identifier et analyser les différents documents intervenants à partir de la demande des clients jusqu'à la livraison de la marchandise on a eu recours au service commercial.

Pour voir comment fonctionne les opérations de demande et livraison des clients, et le mode d'établissement des factures ainsi l'encaissement des paiements. Les données qu'on a récoltées dans ce service sont :

- ✓ Bon de commande : Le rôle du bon de commande est primordial. Il constitue une sorte de "contrat" vis-à-vis du fournisseur. En cas de contestation, ce dernier peut s'y référer.

	Informations
En-tête	<ul style="list-style-type: none"> ✚ Les coordonnées du client (Adresse, numéro de téléphone/fax). ✚ Registre de commerce du client. ✚ Identification fiscale. ✚ La date d'envoi du bon de commande. ✚ Les informations du fournisseur.
Corps	<ul style="list-style-type: none"> ✚ Lieu de livraison. ✚ Délais de livraison. ✚ Quantité de la marchandise demandée par le client. ✚ Unité de mesure, désignation, prix unitaire et prix total. ✚ Règlement (mode de paiement).
Pied de page	<ul style="list-style-type: none"> ✚ Nom, prénom et fonction du client. ✚ Gryfe du client.

Tableau 1 : Structure du bon de commande

Après l'analyse du bon de commande (par le chargé de la clientèle), ce dernier va être dirigé vers le service de recouvrement afin de vérifier si le client aura la possibilité de couvrir les frais de sa demande. Si c'est le cas, le bon commande sera confirmé et renvoyé au service facturation.

- ✓ Facture : Une facture est un document qui atteste de l'achat ou de la vente de biens ou services.

	Informations
En-tête	<ul style="list-style-type: none"> ✚ Les informations de l'entreprise (Ifri). ✚ Numéro de facture. ✚ Date de facture. ✚ Numéro du bon de commande. ✚ Délai de règlement. ✚ Lieu de livraison. ✚ Les informations du client (Nom, prénom, registre de commerce, identification fiscale et adresse).
Corps	<ul style="list-style-type: none"> ✚ Quantité de la marchandise demandée par le client. ✚ Unité de mesure, désignation, quantité, montant(H.T), TVA, montant(TTC).
Pied de page	<ul style="list-style-type: none"> ✚ Total produit fini, montant remise, total

	transport, TVA, timbre fiscale, montant total et net à payer. + Visa du facturier.
--	---------------------------------------------------------------------------------------

Tableau 2 : Structure de la facture

- ✓ Facture de consignation : La consignation est une somme que le fournisseur facture au client en contrepartie de l'emballage qui lui a emprunté et s'engage à rendre au client quand ce dernier lui rend l'emballage consigné.

	Informations
En-tête	+ Les informations de l'entreprise (Ifri). + Numéro de facture de consignation. + Date de facture. + Dépôt de réception. + Les informations du client (Nom, prénom, registre de commerce, identification fiscale et adresse).
Corps	+ Désignation de l'emballage (palettes, intercalaires et casiers). + Unité de mesure, quantité, montant(H.T), remise, TVA, montant(TTC).
Pied de page	+ Total consignation, montant total et net à payer. + Nom, prénom et matricule du transporteur. + Visa du facturier. + Visa du magasinier.

Tableau 3 : Structure de la facture de consignation

- ✓ Facture de déconsignation : Pour la déconsignation, Deux situations peuvent se présenter dans le cas de la déconsignation d'emballages, la première est une déconsignation au prix de la consignation et la deuxième est une déconsignation à un prix inférieur au prix de la consignation.
- ✓ Bon de livraison : c'est un document SD utilisé pour le traitement d'une livraison de marchandises ou de services à un client. C'est sur la base de ce bon que sont exécutées les activités SD relatives à la planification des articles, à leur prélèvement, à l'expédition, au transport et à la facturation. Ce bon est également utilisé pour l'exécution des contrôles qualité de livraisons.

II. 3.2. Données récoltées auprès du service production

On a aussi eu recours au service production, cela dans le but d'évaluer la capacité de production de l'entreprise et d'énumérer la gamme de produits. Durant cette visite on a pu prélever:

- ✦ Les responsables principales :
 - ✦ Chef de quai : c'est la personne qui s'en charge du chargement de la marchandise dans le quai de production.
 - ✦ Chef de groupe : c'est la personne qui s'en charge des stationnements des camions dans le quai de production.
- ✦ Le moyen de chargement des camions : Chariots élévateurs.



Figure 6 : Chariot élévateur

- ✦ Les différents types de palettes ainsi que leurs capacités : les palettes sont utilisées pour transporter les produits. Leur poids est de 30 Kg lorsqu'elles sont vides et 1 tonne si elles sont pleines. On trouve des palettes en bois et des palettes en plastique, comme l'illustre les figures suivantes.



Figure 7 : Palette en bois



Figure 8 : Palette en plastique

Intercalaires : sont soit en plastique ou en carton, qui sont mises entre les différentes lignes des fardeaux.



Figure 9 : intercalaires en carton

En présentera dans le tableau suivant la capacité des palettes :

Désignation produit	FDX/PTTE	Intercalaires
Ifruit 0,20	310	
Ifruit 0,33	290	5 intercalaires
Ifruit 1L	168	6 intercalaires
Ifruit 2L	80	
EMG 0,33	264	
EMG 1,25	140	
EM 1,50	112	
EM 0,5	175	
Bouchon sport 0,50	150	
Soda 0,33	256	
Soda 1,25	140	
Soda 0,25	60	
Soda 1L	40	
Soda 2L	100	

Tableau 4 : Capacité des palettes

II. 3.3. Données récoltées auprès du service Parc

Et comme nous nous intéressons au problème de transport, les données les plus intéressantes sont récoltées au service parc. D'abord on a prélevé la composition de la flotte (Véhicules de transport des produits), ainsi leurs caractéristiques comme : la

référence de chaque camion, date de mise en circulation, le tonnage, la marque, le type d'énergie et la consommation par 100Km.

Le groupe Ifri se dispose de 130 camions dont 127 sont des semi-remorques et 3 des dix-tonnes. La capacité des semi-remorques est de 26 tonnes.

La capacité du parc est de 200 camions, dans le cas d'allocation de nouveaux camions.

L'entreprise a un champ d'action qui s'étale sur tout le territoire national. Une escale peut durer jusqu'à 7 jours en fonction de la ville de destination.

Le chef du parc confirme l'ordre de mission de chaque chauffeur qui sera orienté à une escale et un camion précis.

✦ **Ordre de mission** : c'est un document délivré par la Bejaïa Logistique.

	Informations
En-tête	<ul style="list-style-type: none"> ✦ Les informations de la Bejaia logistique. ✦ Numéro séquentiel du chauffeur. ✦ Date du document.
Corps	<ul style="list-style-type: none"> ✦ Référence du document. ✦ Référence du véhicule. ✦ Identifiant, nom, prénom et fonction du chauffeur. ✦ Objet de la mission (type de la marchandise transportée). ✦ Ville départ, ville destination. ✦ Nom, prénom et numéro de téléphone du client. ✦ Itinéraire. ✦ Date départ/arrivée, heure départ/arrivée, Km départ/arrivé. ✦ Moyen de transport. ✦ Immatriculation.
Pied de page	<ul style="list-style-type: none"> ✦ Gryfe du chef de parc. ✦ Visa de a sécurité.

Tableau 5 : Ordre de mission

L'activité de la logistique du transport s'étale tout au long d'un processus qu'on le présentera par la suite, dont les deux premières étapes sont assurées par le service commercial.

- ✦ Demande d'une marchandise par un client : Le chargé de la clientèle reçoit les demandes des clients et les dirige au service recouvrement.
- ✦ Traitement et validation des commandes : Le service recouvrement traite et vérifie les commandes. Après la validation, les demandes son transmise au service facturation afin d'établir les factures correspondantes.
- ✦ Chargement de la marchandise : A ce niveau là, la marchandise est chargée dans un camion.
- ✦ Attribution d'un chauffeur et un camion à une escale : En fonction du moyen du transport, le chef du parc affecte un chauffeur à l'escale associée.

- ✦ Enregistrement des entrées/sortie des camions : A l'arrivé/sortie d'un camion, ce dernier est enregistré grâce à des terminales spécialisé.
- ✦ Suivi des escales : Un suivi en temps réel des escales est assuré grâce à l'outil GPRS.

La gestion des escales est assurée par une application disposant d'une simple interface. Cette application est accessible que par les administrateurs système.

Notre solution vise à remettre en forme l'application ainsi de la rendre accessible par les autres employés de l'entreprise.

II.4. Conclusion

Grace à cette étude, on a bien réussi à nous familiariser avec l'environnement autour du quelle nous allons développer notre plate-forme.

Dans ce qui suit, nous allons donner en détail la démarche suivie pour le développement et la réalisation de notre plate-forme.

Chapitre III : Démarche adoptée et sa mise en œuvre

III. 1. Introduction

La conception d'un logiciel exige le passage par une méthode de développement qui s'appuie sur un langage de modélisation. Une méthode de développement est tout simplement une succession d'étapes permettant de maîtriser le déroulement du projet en garantissant l'aboutissement aux résultats finaux attendus. Pour faire, chaque étape du développement logiciel un modèle du système est généré afin d'en tirer certains détails du monde réel.

Pour concevoir notre système, nous avons opté pour l'approche objet avec le processus 2TUP comme méthode de développement et UML comme langage de modélisation.

III. 2. Présentation du processus du développement logiciel

Devant le nombre de méthodes disponibles, le choix parmi elles devient difficile, beaucoup de questions peuvent se poser à un chef de projet lors d'un démarrage de projet :

« Comment vais-je organiser les équipes de développement ! Quelles tâches attribuer à qui ? Quel temps faudrait-il pour livrer le produit ? Comment faire participer le client au développement afin de capter les besoins de celui-ci ?! ... »

Parmi les différentes méthodes du développement logiciel, citant : RUP, XP, 2TUP,... Notre choix s'est porté vers la méthode **2TUP**.

Au contraire du RUP axé processus qui est très coûteux à personnaliser et de l'eXtreme Programming qui est assez flou dans la mise en œuvre, le 2TUP permet de paralléliser et de dissocier les besoins fonctionnels des besoins techniques et donc de mieux répartir les contraintes initiales en amont du projet.

La réalisation de notre projet se base sur un processus de développement bien défini qui nous accompagne depuis la détermination des besoins fonctionnels attendus jusqu'à la conception et le codage final. Qui se base lui-même sur le **Processus Unifié**.

III. 2.1. Le Processus Unifié [W8]

Un processus unifié est un processus construit sur UML (Unified Modeling Language). Les processus unifiés sont le résultat de l'unification, non pas des processus, mais plus exactement les meilleures pratiques du développement objet.

Le Processus Unifié est :

- **Itérative et incrémental** : Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et développer une solution effective de façon incrémentale et cela par des itérations multiples.
- **Conduit par les cas d'utilisation** : Le processus est orienté par les besoins utilisateurs.
- **Piloté par les risques** : Afin de minimiser les risques d'échec du projet on doit définir une priorité pour chaque fonctionnalité.
- **Centré sur l'architecture** : les modèles définit tout au long du processus de développement vont contribuer à établir une architecture cohérente et solide.

Le Processus Unifié comprend 4 phases, sont les suivantes :

a. Phase de démarrage :

- Analyse des fonctionnalités.
- Evaluer les risques.
- Critères d'évaluation : Concurrence, Première validation des besoins, Evaluation des coûts, risques, ...

b. Phase d'élaboration :

- Planifier les activités nécessaires et les ressources requises.
- Concevoir l'architecture.
- Critères d'évaluation : Stabilité du produit et de la conception, Résolution des problèmes critiques, Evaluation des coûts et la validation du produit.

c. Phase de construction :

- Construire le produit comme une série d'itérations incrémentales.
- Critères d'évaluation : Stabilité et maturité des réalisations, Capacité de mettre en œuvre la transition, Coûts acceptables.

d. Phase de transition :

- Fournir le produit aux utilisateurs finaux.
- Critères d'évaluation : Validation des besoins.

Chaque phase de ce processus est composée elle-même en itérations limitées dans le temps. A la fin de chacune d'elles un système testé, intégré et exécutable est donné lieu.

Ces activités de développement sont définies par 6 disciplines fondamentales, pouvant se dérouler à travers plusieurs phases qui sont : la capture des besoins, la modélisation métier, l'analyse et la conception, l'implémentation, le test et le déploiement.

III. 2.2. Le processus 2TUP [PERI, 00]

Le processus 2TUP (Two Track Unified Process) est un processus unifié. Il gère la complexité technologique en donnant part à la technologie dans son processus de développement. «2 Track» signifie littéralement que le processus suit deux chemins. Il

s'agit des chemins fonctionnels et d'architecture technique, qui correspondent aux deux axes de changement imposés au système d'information.

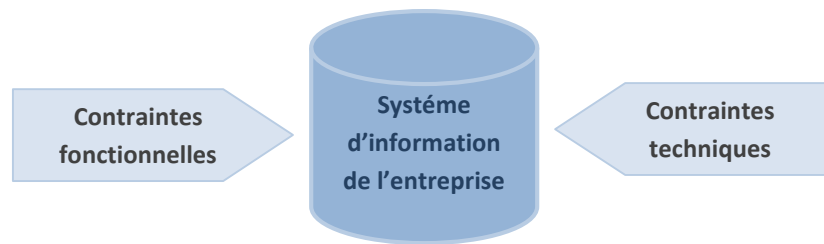


Figure 10 : Le système d'information soumis à deux types de contraintes

Le 2TUP propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle. La figure suivante détaille les étapes de développement des trois branches du processus 2TUP.

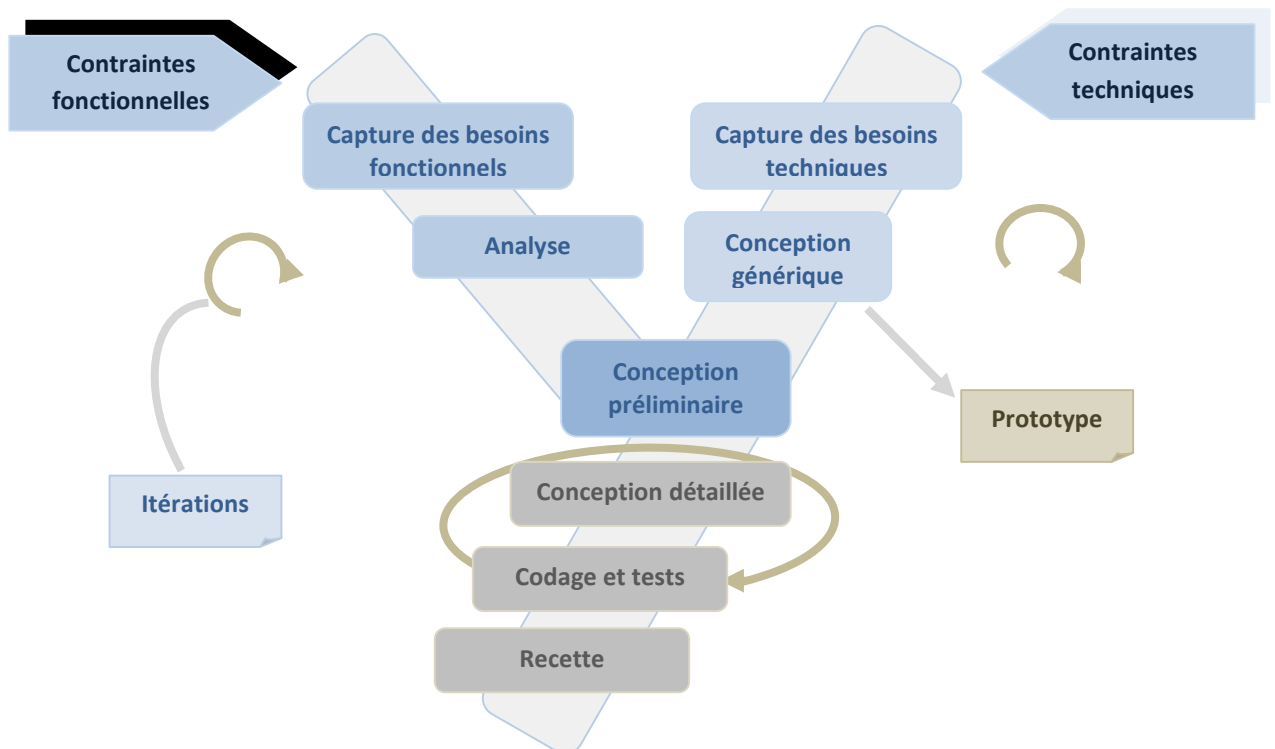


Figure 11 : Le processus de développement en Y

➤ **Branche fonctionnelle (celle du gauche) :** capitalise la connaissance du *métier* de l'entreprise. Elle constitue généralement un investissement pour le moyen et le long terme.

- Capture des besoins fonctionnels : produit le modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs.
- L'analyse : consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

- **Branche architecture technique (celle du droit) :** capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme.
 - La capture des besoins techniques : recense toutes les contraintes sur les choix de dimensionnant et la conception du système. Les outils et le matériel sélectionné ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique).
 - La conception générique : définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle a pour objectif de d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype.

- **Branche conception (celle au milieu) :** à l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches.
 - La conception préliminaire : représente une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.
 - La conception détaillée : étudie ensuite comment réaliser chaque composant.
 - L'étape de codage : produit ses composants et teste au fur et à mesure les unités de code réalisées.
 - L'étape de recette : consiste enfin à valider les fonctionnalités du système développé.

III. 2.3. Un processus de modélisation avec UML [ROQ, 07]

Le processus 2TUP s'appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent de part leur facilité et clarté, de bien modéliser le système à chaque étape.

UML (*Unified Modeling language*) 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.

Voici une présentation rapide des différents diagrammes UML qui vont être utilisés tout au long du projet :

- **Le diagramme des cas d'utilisation :** Représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est normalement utilisé lors des étapes de capture des besoins fonctionnels et techniques.

- **Le diagramme d'activités :** Représente les règles d'enchaînement des activités et d'actions dans le système. Il peut être assimilé comme un algorithme mais schématisé.
- **Le diagramme de packages :** Dans le langage UML, ce diagramme modélise des catégories cohérentes entre elles, pour un souci de partage des rôles. Correspond à l'étape de modélisation des différents scénarios d'un cas d'utilisation.
- **Le diagramme de classes :** Sûrement l'un des diagrammes les plus importants dans un développement orienté objet, sur la branche fonctionnelle, ce diagramme est prévu pour développer la structure des entités manipulées par les utilisateurs. En conception le diagramme de classes représente la structure d'un code orienté objet.
- **Le diagramme de séquence :** Représente les échanges de message entre objet. Dans le cadre d'un fonctionnement particulier du système.
- **Le diagramme de déploiement :** Montre la disposition physique des matériels qui composent le système et la répartition des composants sur ses matériels.

III. 3. Conclusion

La meilleure façon d'aborder un problème, c'est de s'y attaquer de front. C'est ce que propose le 2TUP en faisant une place à part entière à la technologie dans le processus de développement.

Il est alors possible de présenter rapidement aux utilisateurs des éléments de validation. De plus, l'itératif permet une gestion efficace des risques, en abordant dès les premières itérations, les points difficiles. Par exemple, les premières itérations de la phase technique aborderont les aspects sécurité et transaction.

Après avoir défini les différents processus de notre projet, nous entamons le chapitre de l'étude préliminaire ou seront décrits les différents besoins fonctionnels de notre application.

Chapitre IV: Etude préliminaire

IV. 1. Introduction

L'étude préliminaire nous donne la possibilité de développer la première étape de la branche gauche du fameux cycle en Y. Cette étape décrit les différentes fonctionnalités du système et particulièrement la façon de les utiliser.

La technique mise au point comporte deux objectifs principaux :

- ✓ Dialoguer avec le client sur son expérience préliminaire de besoins grâce à une description fonctionnelle qu'il comprend facilement.
- ✓ Préparer la modélisation orientée objet en aidant à travers les classes principales du futur modèle statique.

IV. 2. Recueil des besoins fonctionnels

Il s'agit des besoins spécifiant un comportement d'entrée /sortie du système. Les besoins fonctionnels qu'on a déduit à partir de notre étude sont les suivants :

- Gérer les camions (Ajout, Modification, Suppression).
- Gérer les escales (Ajout, Modification).
- Garder trace des camions entrants et sortants.
- Consulter l'état du parc.
- Recherche d'un camion, d'un chauffeur ou d'une escale (par critère).

IV. 3. Description du contexte

Le recueil effectué lors de l'étude préliminaire, nous a permis de repérer les besoins et de modéliser le contexte du système à réaliser. Cette étape consiste en trois activités successives :

- Identification des acteurs.
- Identification des messages.
- Réalisation des diagrammes de contexte.

IV. 3.1. 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. 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. [ROQ,07]

Nous allons présenter dans le tableau ci-dessous les acteurs de notre système et leurs principaux rôles :

Acteur	Rôle
Chef du parc	<ul style="list-style-type: none"> ▪ Il gère les camions ainsi que leurs escales. ▪ Il enregistre les entrées/sorties des camions. ▪ Affectation des chauffeurs aux escales associées.
Visiteur	<ul style="list-style-type: none"> ▪ Consulter l'état du parc. ▪ Consulter les différentes escales (camions) ainsi que les informations correspondantes.

Tableau 6 : Les acteurs de notre système

IV. 3.2. Identification des messages

Un message est un élément de communication unidirectionnel entre objets qui déclenche une activité dans l'objet destinataire. La réception d'un message provoque un événement dans l'objet récepteur. [ROQ, 07]

Notre système émet les messages suivants :

1. La fiche descriptive d'un camion.
2. La fiche descriptive d'une escale.
3. La liste des camions.
4. La liste des escales.
5. La liste des chauffeurs.
6. La fiche descriptive d'un chauffeur.

Notre système reçoit les messages suivants

7. Création, modification et suppression d'un camion.
8. Création, modification d'une escale.
9. Enregistrement d'une entrée/sortie.
10. Lancement d'une recherche sur un camion, un chauffeur ou une escale.
11. Consultation des escales, des camions et des chauffeurs.

IV. 3.3. Modélisation du contexte

Le diagramme de contexte dynamique permet de positionner le système dans son environnement selon le point de vue des communications. Il reprend les éléments du

contexte statique et précise les échanges d'informations qui sont réalisés entre le système et les éléments matériels extérieurs au système. Le système est donc décrit physiquement et logiquement. [W9]

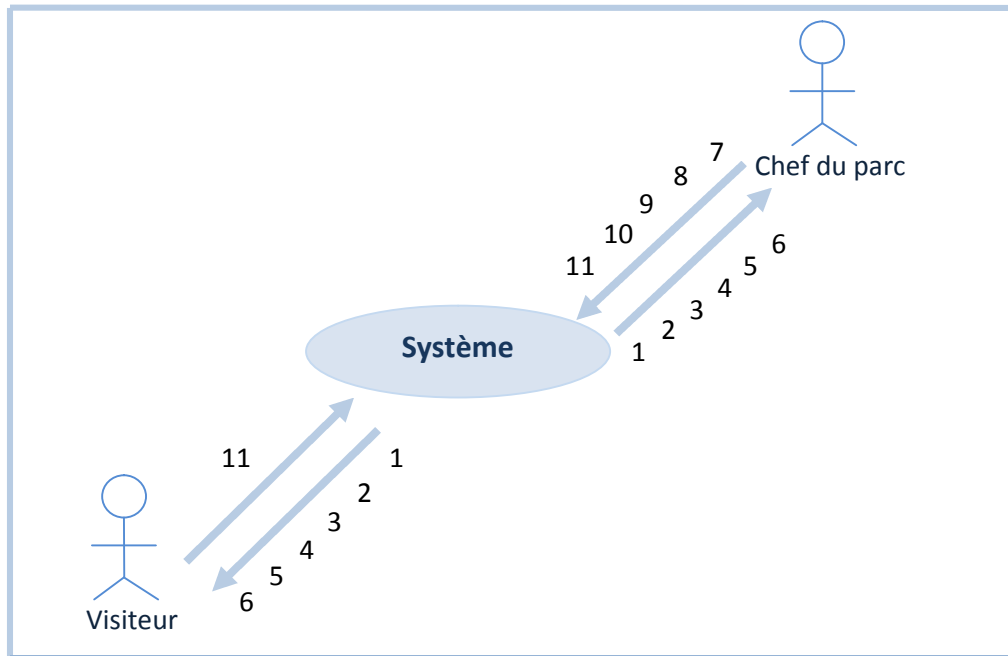


Figure 12 : Diagramme de contexte dynamique

IV. 4. Conclusion

Dans ce chapitre, nous avons établi un recueil initial des besoins fonctionnels. Par la suite, nous avons modélisé le contexte général du système, considéré comme étant une boîte noire, en identifiant les différents acteurs qui interagissent directement avec le système étudié ainsi que les messages échangés.

Chapitre V: Capture des besoins fonctionnels

V. 1. Introduction

Cette phase représente un point de vue « fonctionnel » de l'architecture système. Par le biais des cas d'utilisation, nous serons en contact permanent avec les acteurs du système en vue de définir les limites de celui-ci, et ainsi éviter de trop s'éloigner des besoins réels de l'utilisateur final.

Les points traités dans cette partie sont :

- Identification des cas d'utilisation du système par ses acteurs.
- Description des cas d'utilisation.
- Organisation des cas d'utilisation.
- Identification des classes candidates du modèle d'analyse.

V. 2. Identification des cas d'utilisation

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné. [ROQ, 07]

Le repérage des cas d'utilisation est l'étape qui nous permettra de préciser l'étude du contexte fonctionnel du système, en décrivant les différentes façons d'utiliser le futur système par les acteurs.

A ce niveau un découpage du système en catégories est nécessaire pour décomposer l'activité métier en 2 paquetages :

- Le paquetage « Gestion camion » : Regroupe les cas d'utilisation ayant accès à toute les informations relatives au moyen de transport.
- Le paquetage « Gestion parc » : Encapsule les opérations relatives au parc.

Ces paquetages représentent les besoins d'un métier du système communautaire sous forme d'un ensemble de cas d'utilisation constituants avec les acteurs fonctionnels, un diagramme de cas d'utilisation.

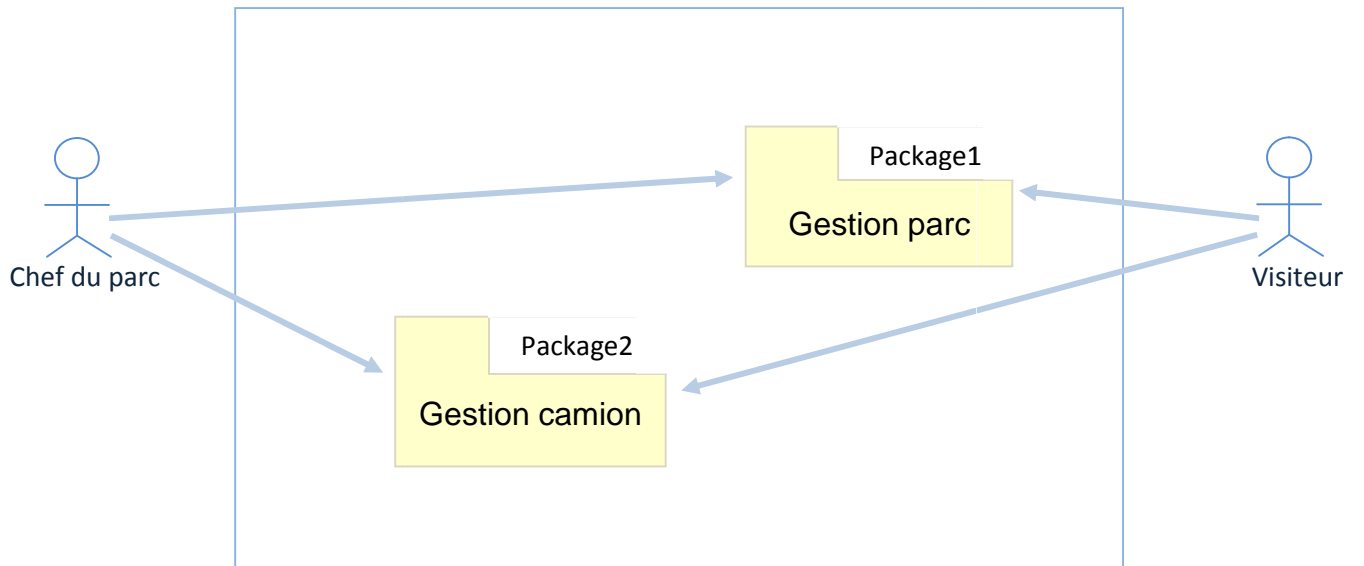


Figure 13 : Les deux packages fonctionnels du système

V.2.1. Diagramme de cas d'utilisation du package « Gestion parc »

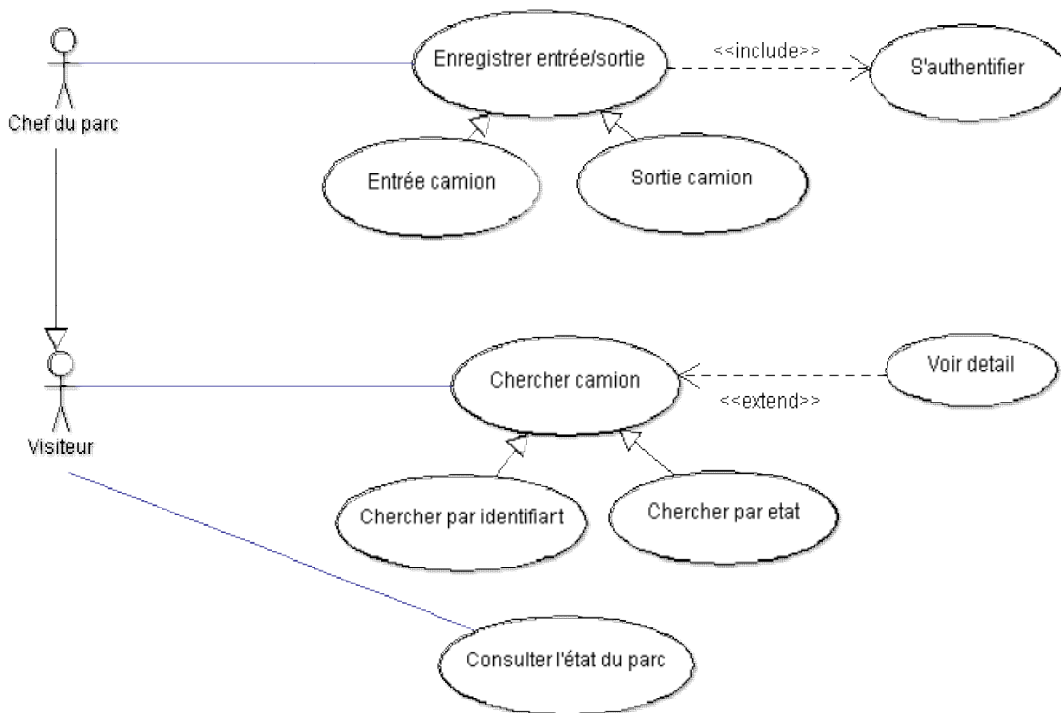


Figure 14 : Diagramme de cas d'utilisation du package « Gestion parc »

V.2.2. Diagramme de cas d'utilisation du package « Gestion camion »

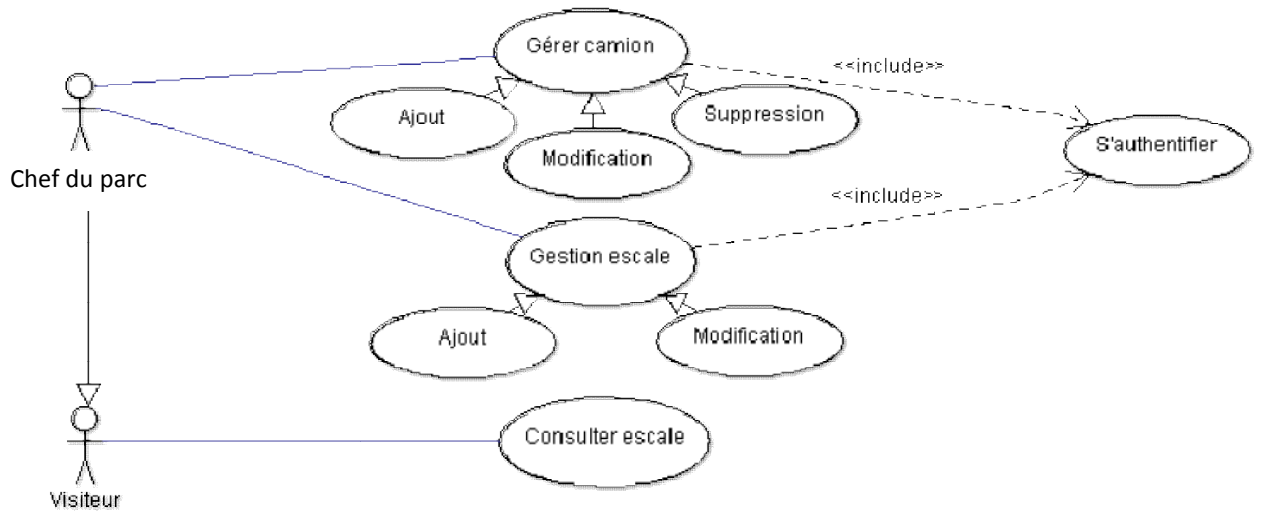


Figure 15 : Diagramme de cas d'utilisation du package « Gestion camion »

V.2.3. Diagramme de cas d'utilisation général

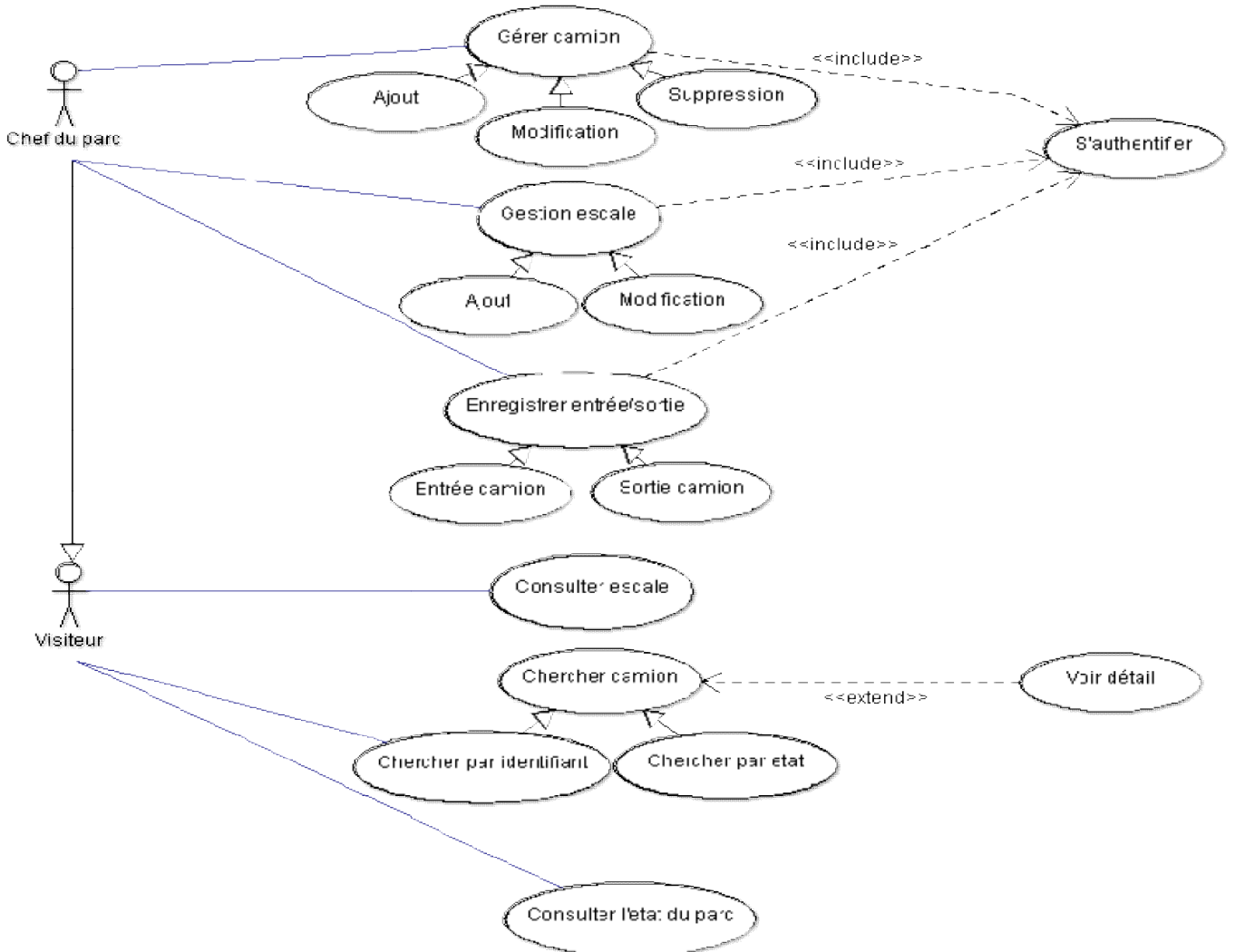


Figure 16 : Diagramme de cas d'utilisation général

V. 3. Description des cas d'utilisation

Nous proposons dans ce qui suit une description textuelle des principaux cas d'utilisation complétée par les diagrammes d'activités.

Un diagramme d'activités est un cas particulier d'une machine d'état, qui est utilisé pour les processus de modèle impliquant une ou plusieurs classificateurs. Son objectif principal est la séquence et conditions pour les actions qui sont prises, plutôt que sur les éléments qui mettent en œuvre les actions. [DESM, O6]

V.3.1. Package « Gestion camion »

Cas d'utilisation « gérer escale »

<p><u>Identification</u> Titre : Gérer escale. But : Gérer les escales (ajout et modification). Acteur : Chef du parc.</p>
<p><u>Séquencement</u></p> <p>Pré-conditions : Le Chef du parc doit s'authentifier.</p> <p>Enchaînement nominal :</p> <ul style="list-style-type: none"> ▪ Le Chef du parc sélectionne la rubrique escale. ▪ Saisie des informations de la nouvelle escale en cas d'ajout. ▪ Dans le cas de modification le Chef du parc choisit l'escale à modifier. ▪ Après les deux étapes précédentes, un contrôle est déclenché pour vérifier la cohérence des données saisies. ▪ Enregistrement des changements. <p>Post-conditions : Mise à jour de la base de données.</p>

Tableau 7 : Description du cas d'utilisation « Gérer escale »

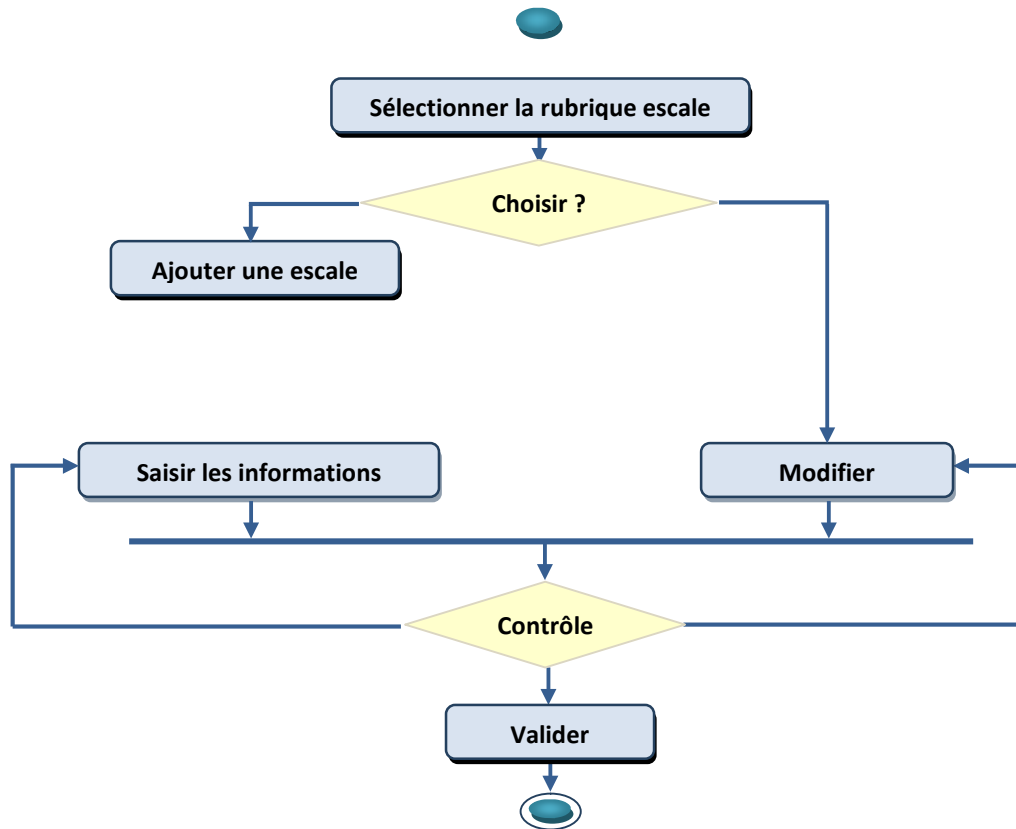


Figure 17 : Diagramme d'activité du cas d'utilisation
 « Gérer escale »

Cas d'utilisation « Consulter escale »

<p><u>Identification</u> Titre : Consulter escale. But : Consulter les informations concernant une escale ou un camion. Acteur : Visiteur, Chef du parc.</p>
<p><u>Séquencement</u> Pré-conditions. Enchaînement nominal :</p> <ul style="list-style-type: none"> ▪ L'utilisateur sélectionne la rubrique consultation et choisit d'afficher la liste des escales. ▪ La liste des escales s'affiche. ▪ L'utilisateur sélectionne l'escale dont il veut consulter les informations. ▪ Les informations concernant l'escale s'affichent. <p>Exception :</p> <ul style="list-style-type: none"> ▪ Le nombre des escales affichés est très important alors l'utilisateur filtre la liste par date. <p>Post-conditions.</p>

Tableau 8 : Description du cas d'utilisation « Consulter escale »

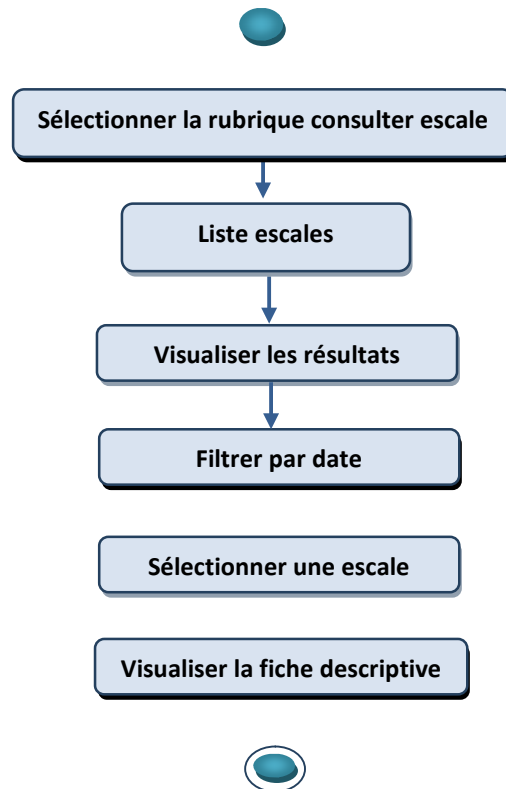


Figure 18 : Diagramme d’activité du cas d’utilisation « Consulter escale »

Cas d’utilisation « Gérer un camion »

<p><u>Identification</u> Titre : Gérer un camion. But : Gérer un camion. Acteur : Chef du parc.</p>
<p><u>Séquencement</u> Pré-conditions : Le Chef du parc doit s’authentifier. Enchaînement nominal :</p> <ul style="list-style-type: none"> ▪ Le Chef du parc sélectionne la rubrique camion. ▪ Le Chef du parc saisie les différentes informations sur le nouveau camion en cas d’ajout. ▪ Dans le cas de modification le Chef du parc choisit le camion à modifier. ▪ Le Chef du parc choisit le camion à supprimer en cas de suppression. ▪ Validation des changements. <p>Post-conditions : Mise à jour de la base de données.</p>

Tableau 9 : Description du cas d’utilisation « Gérer camion »

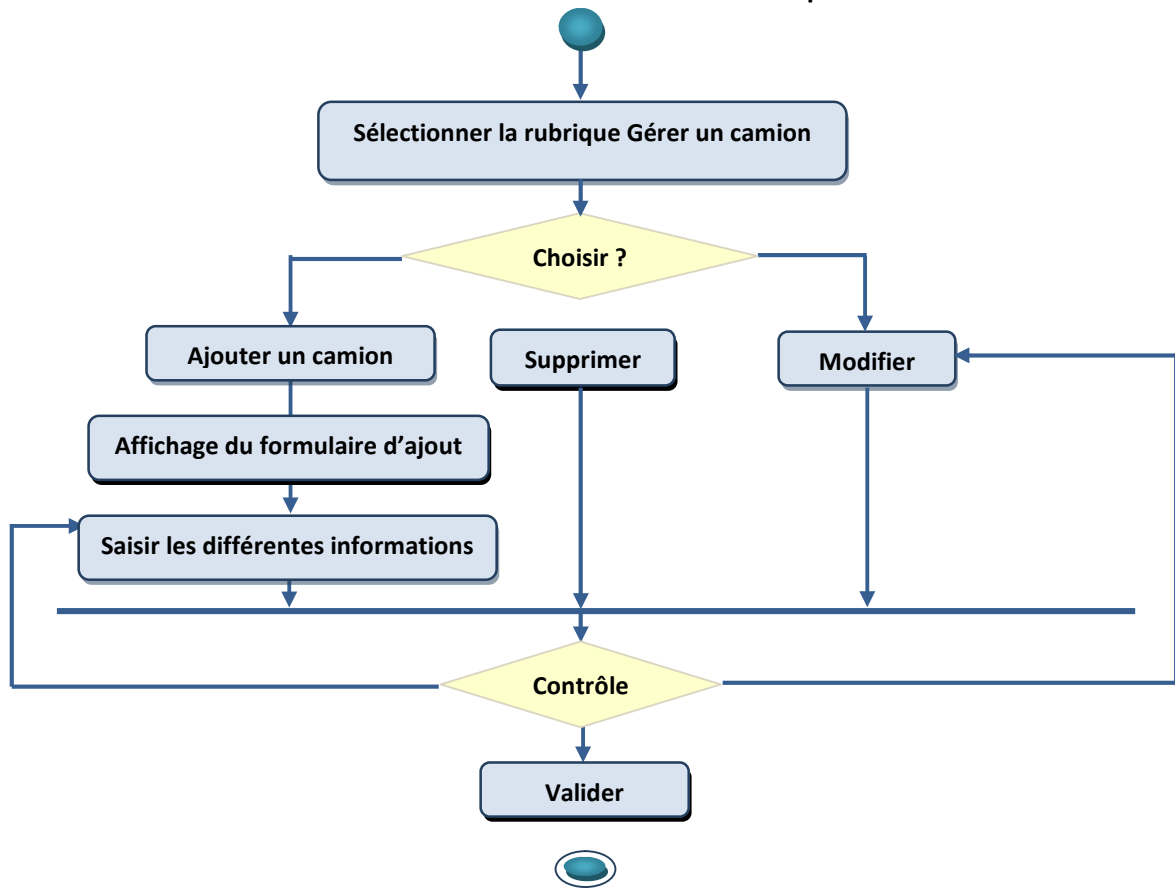


Figure 19 : Diagramme d'activité du cas d'utilisation « Ajouter camion »

V. 3.2. Package « Gestion parc »

Cas d'utilisation « Chercher camion »

<p><u>Identification</u> Titre : Chercher un camion. But : Chercher un camion à partir du numéro d'identification. Acteur : Consultant, Chef du parc.</p>
<p><u>Séquencement</u> Pré-conditions : Néant Enchaînement nominal :</p> <ul style="list-style-type: none"> ▪ L'utilisateur lance une recherche à partir du mot-clé « numéro d'identification ». ▪ Le système affiche une page de résultats. ▪ L'utilisateur sélectionne un camion, alors le système lui présente sa fiche descriptive détaillée. <p>Post-conditions : L'utilisateur a trouvé le camion qu'il cherchait.</p>

Tableau 10 : Description du cas d'utilisation « Chercher camion »

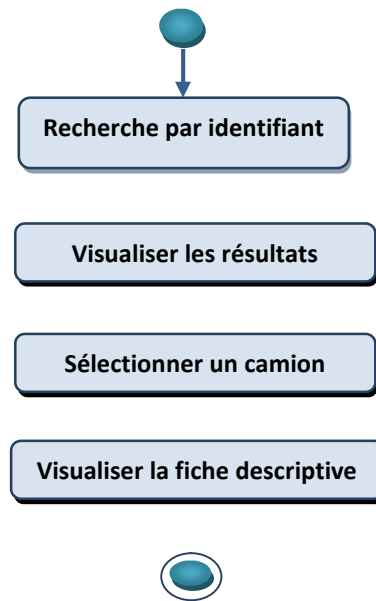


Figure 20 : Diagramme d'activité du cas d'utilisation « Chercher camion »

Cas d'utilisation « Enregistrer une entrée/sortie »

<p><u>Identification</u> Titre : Enregistrer une entrée/sortie. But : Enregistrer une transition de livraison ou de restitution. Acteur : Chef du parc.</p>
<p><u>Séquencement</u> Pré-conditions : Le Chef du parc doit s'authentifier. Enchaînement nominal :</p> <ul style="list-style-type: none"> ▪ Le Chef du parc sélectionne la rubrique transition. ▪ Le système affiche une page d'ajout d'une transition. ▪ Le Chef du parc sélectionne le type de transition puis saisie ses différentes informations. ▪ Le système enregistre la transition. <p>Post-conditions : Mise à jour de la base de données.</p>

Tableau 11 : Description du cas d'utilisation « Enregistrer une entrée/sortie »

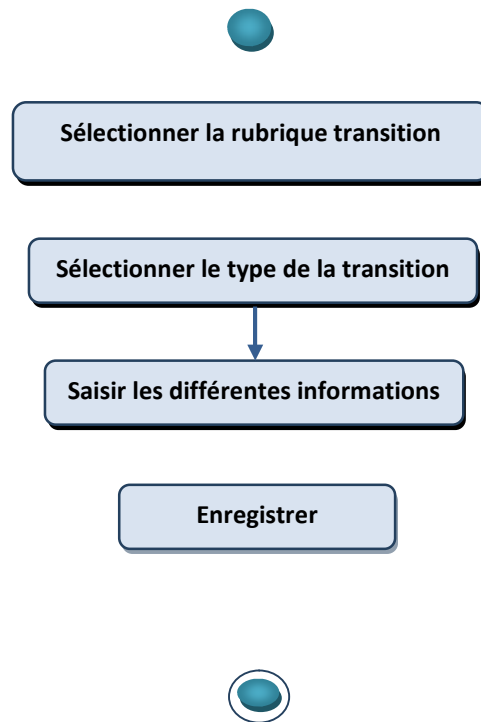


Figure 21 : Diagramme d'activité du cas d'utilisation « Enregistrer entrées/ sorties »

Cas d'utilisation « Consulter état du parc »

<p><u>Identification</u> Titre : Consulter l'état du parc. But : Consulter l'état du parc en vue d'afficher la liste des camions dans le parc. Acteur : Consultant, Chef du parc.</p>
<p><u>Séquencement</u> Pré-conditions. Enchaînement nominal :</p> <ul style="list-style-type: none"> ▪ L'utilisateur sélectionne la rubrique état du parc. ▪ Le système affiche la liste des camions existants dans le parc. <p>Post-conditions : L'utilisateur voulait consulter l'état du parc, alors c'est fait.</p>

Tableau 12 : Description du cas d'utilisation « Consulter état du parc »

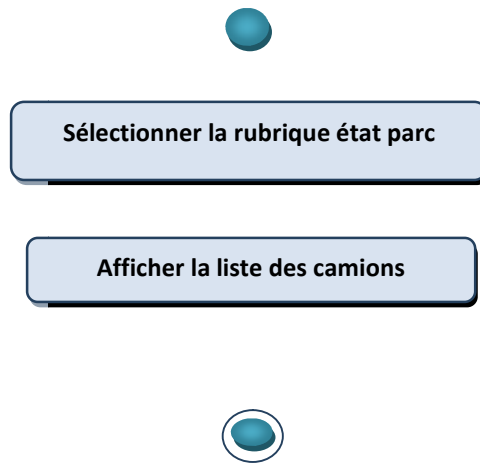


Figure 22 : Diagramme d'activité du cas d'utilisation « Etat du parc »

V. 4. Identification des classes candidates

Cette phase va préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d'analyse.

- On a ajouté une classe personne, qui est une classe mère dont héritent les classes filles : Chauffeur et Client, Afin de remédier au problème de duplication des mêmes attributs des classes filles.

Package « Gestion camion »

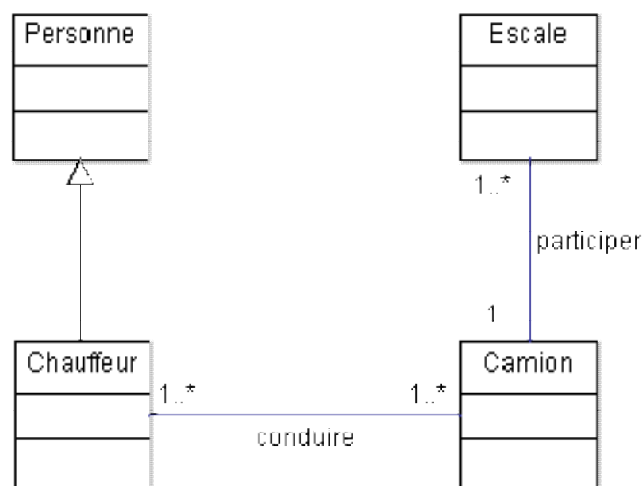


Figure 23 : Diagramme de classes participantes du package « Gestion camion »

Package « Gestion parc »

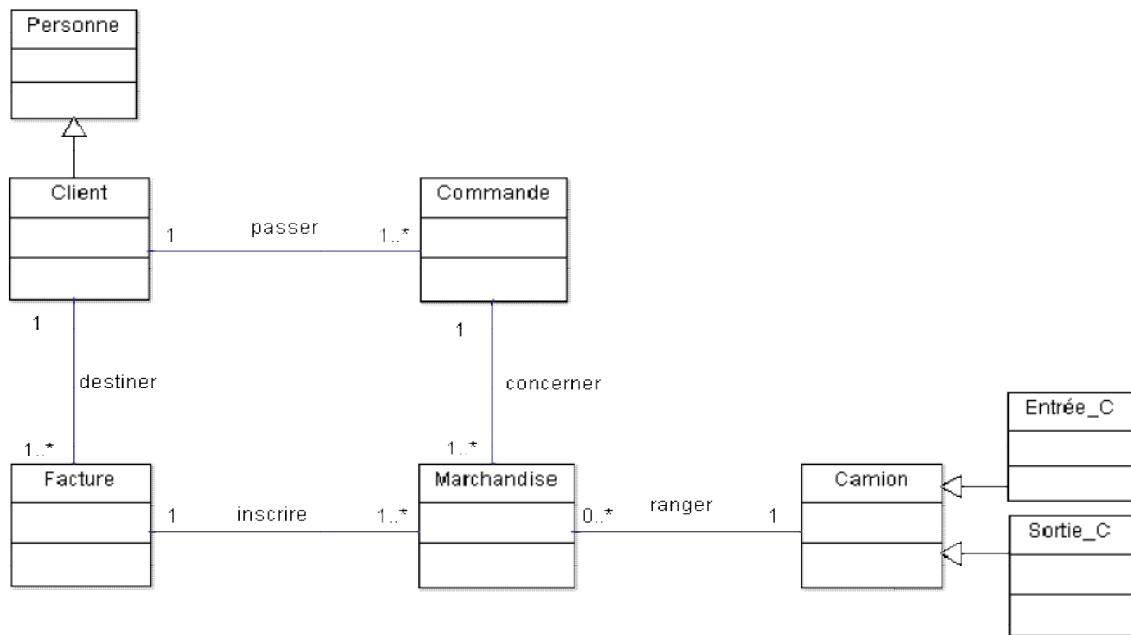


Figure 24 : Diagramme de classes participantes du package « Gestion parc »

V. 5. Conclusion

Les besoins fonctionnels regroupent l'ensemble des services offerts aux différents usagers de notre système. Nous avons décrit de façon détaillée les différents cas d'utilisation ainsi que leurs diagrammes d'activités associés, puis, nous avons identifié les classes candidates. Cette dernière phase nous aidera à la construction du modèle statique d'analyse.

Chapitre VI : Capture des besoins techniques

VI. 1. Introduction

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative et elle se présente comme suit :

- ‡ Capture des spécifications techniques liées à la configuration matérielle.
- ‡ Capture des spécifications logicielles.

VI. 2. Capture des spécifications techniques

Les besoins opérationnels et les choix stratégiques de développement impliquent des contraintes relatives à la configuration du réseau matériel. Elles concernent les performances d'accès aux données, la sécurité du système, l'interopérabilité, l'intégration des applications, la volumétrie et le mode d'utilisation du système.

L'entreprise se dispose d'un grand parc informatique, le tableau ci-dessous représente le matériel existant au sein de l'entreprise d'accueil.

Désignation article	Quantité	Etat
Micro ordinateur & accessoire E4600	1	Bon état
Micro ordinateur HP DX 2400 & accessoire	4	Bon état
Micro portable HP 6730 S core	1	Bon état
Micro-ordinateur HP DC 7900& écran HP LCD &accessoire	6	Bon état
Serveur HP proliant DL 380 G6	2	Bon état
Micro-Portable TOSHIBA	1	Bon état
Multifonction laser canon MF8030Cn	2	Bon état
Imprimante laser monochrome CANON LBP 30/10	5	Bon état
Point d'accès WIFI & antenne	1	Bon état
Disque HP 300 GB 6G 2,5 SAS DP HDD	2	Bon état
Micro-ordinateur Dell VOSTRO 115/230	1	Bon état
Micro-ordinateur HP Compaq 500 MB	1	Bon état
Micro-ordinateur HP Pavillon P2-1121L	2	Bon état
Onduleur APC 500 VA3S	2	Bon état

Fax laser	1	Neuf
Photocopieur numérique, fax, imprimante CANON 2018	1	Bon état
Vidéo projecteur EPSON EB-S7 NTCS	1	Bon état
Onduleur NOVA	6	Bon état

Tableau 13 : Inventaire du matériel informatique de l'entreprise

Le groupe Ifri qui se compose de quatre zones, dispose de plusieurs serveurs installés dans les trois sites : zone I (Ifri), zone III et Général Plast. Les quatre zones sont reliées avec une fibre optique. La sécurité de transfert d'information est assurée grâce au tunnel VPN, comme l'illustre la figure ci-dessous.

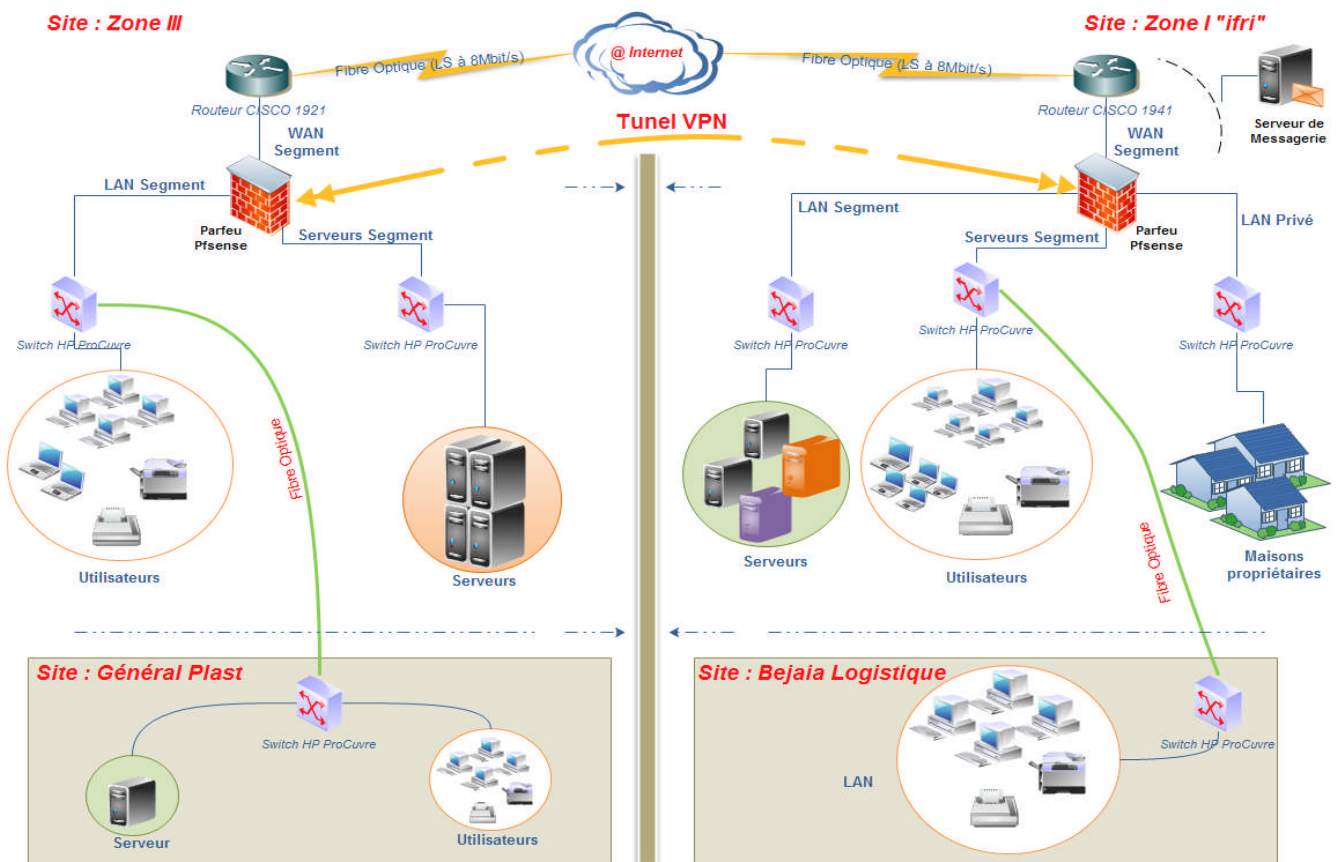


Figure 25 : Réseau globale du groupe Ifri

Les administrateurs de l'entreprise utilisent l'architecture 3-tiers qu'on va la décrire dans la suite.

VI. 2.1. Architecture 3-tiers [GLUC, 12]

L'architecture en niveaux spécifie le nombre de niveaux géographiques et organisationnels où vont se situer les environnements d'exécution du système.

L'architecture à 3-tiers, caractérise les systèmes client/serveur d'où il existe un niveau intermédiaire permettant de fournir des services à un client, c'est-à-dire l'architecture peut être séparée nettement en trois couches logicielles au sein d'une même application.

✦ **Couche présentation (premier niveau)**

Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'interface homme machine. Elle est exploitée par un navigateur web. La couche présentation relaie les requêtes de l'utilisateur à destination de la couche métier, et en retour lui présente les informations renvoyées par les traitements de cette couche.

✦ **Couche métier/business (second niveau)**

Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la «logique», et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation. Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche. La couche métier offre des services applicatifs et métier à la couche présentation. Pour fournir ses services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche inférieure.

✦ **Couche accès aux données (troisième niveau)**

Elle consiste de gérer l'accès aux gisements de données du système. Ces données peuvent être propres au système où gérées par un autre système.

Ce modèle d'architecture à pour objectif d'alléger le poste de travail client et de prendre en compte l'hétérogénéité des plates formes (serveurs, clients, langage, etc.).

Le schéma suivant illustre la répartition des couches dans une architecture trois tiers :



Figure 26 : La répartition des différentes couches de l'architecture 3-tiers

Le style d'architecture 3-tiers facilite la réutilisation au sein d'un système, puisque les composants métier correspondent à des concepts communs à différents métiers de l'entreprise.

VI. 2.2. Architecture de l'application web [GLUC, 12]

Actuellement, la technologie réseaux et les domaines qu'elle inclut prennent de plus en plus une place très importante dans notre vie.

La multiplicité et la variation des architectures d'application web, implique l'existence d'au moins quatre composants d'architecture significatifs :

- ✦ Le navigateur client : c'est un navigateur HTML (Hypertexte Markup Language) standard compatible avec les formulaires et DHTML (Dynamic HTML). Equipé d'une interface utilisateur chargée de la présentation. Son unique fonction est d'accepter et de renvoyer des cookies. L'utilisateur requiert des pages HTML auprès du serveur à travers le navigateur.
- ✦ Le serveur web : c'est le point d'accès principal pour tous les navigateurs clients. C'est un logiciel chargé de transmettre au client en ayant fait la demande via l'URL, les fichiers statiques demandés, présents sur la machine serveur.
- ✦ Le serveur d'applications : c'est le principal exécuteur de la logique «métier» du côté du serveur. Il rend accessible les données de l'application : il doit pouvoir accéder à de nombreuses sources de données. Il prend en charge l'ensemble des fonctionnalités qui permettent à plusieurs clients d'utiliser la même application.
- ✦ Le serveur de données : Fournissant à un serveur d'application les données dont il a besoin. Permet de gérer la persistance des objets métier.

On peut identifier dans les applications web d'aujourd'hui plusieurs patterns architecturaux. Nous avons opté pour le pattern du client web léger.

Le client web léger : Est le pattern architectural le plus classique aujourd'hui, désigne une application accessible via une interface web (en HTML) consultable à l'aide d'un navigateur web, où la totalité du logique métier est traitée du côté du serveur. Pour ces raisons, le navigateur est parfois appelé client universel.



Figure 27 : Schéma des composants matériels du système

VI. 3. Capture des spécifications logicielles

Après avoir traité les spécifications matérielles et d'architecture de notre application, on passera aux fonctionnalités du système technique en procédant à une spécification logicielle, tout en se basant sur les concepts d'exploitant et de cas d'utilisation technique.

VI. 3.1. Les exploitants du système

L'exploitant est un acteur au sens d'UML, si ce n'est qu'il ne bénéficie que des fonctionnalités techniques du système.

Les exploitants de notre système sont :

- ✦ L'utilisateur qui utilise une fonctionnalité de l'application.
- ✦ L'informaticien qui est chargé de déployer et de dépanner le système.

VI. 3.2. Les cas d'utilisation techniques

Un cas d'utilisation technique est destiné à l'exploitant. C'est une séquence d'action produisant une valeur ajoutée opérationnelle ou purement technique.

- ✦ **Manipuler des objets par les utilisateurs** : l'utilisateur va travailler avec des entités sous la forme d'objets, ce qui implique la mise en œuvre des mécanismes de persistance et de gestion de cycle de vie des objets.
- ✦ **Gérer la sécurité** : L'utilisateur doit se connecter et être reconnu du système. L'authentification est le mécanisme qui protège le système des intrusions externes donc les exploitants doivent soumettre à des règles de sécurité qui sont l'authentification et le cryptage.
- ✦ **Utiliser l'aide** : Chaque utilisateur doit disposer d'une aide contextuelle qui l'aide à exploiter le système d'une manière plus efficace.
- ✦ **Gérer les erreurs et les exceptions** : Le système doit être en mesure de générer des traces et des alertes pour faciliter sa maintenance au sein du système.

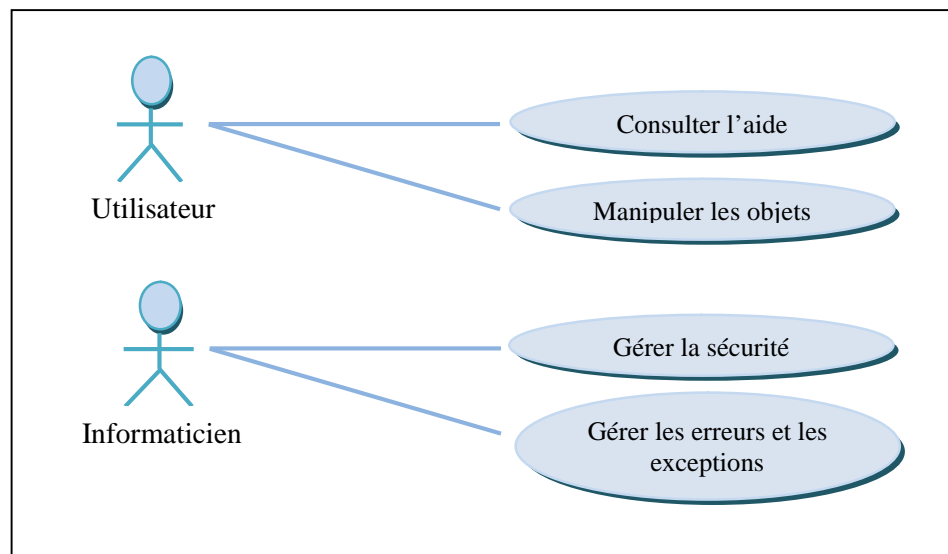


Figure 28 : Modèle de spécification logicielle

VI. 3.3. Description sommaire des cas d'utilisation techniques

VI. 3.3.1. Cas d'utilisation «Manipuler des objets par les utilisateurs»

Titre : manipulation des objets de système.

Exploitant : l'utilisateur.

Intention : l'utilisateur désire agir sur le cycle de vie d'un ou plusieurs objets.

Action : créer, modifier, supprimer un objet.

Exemple : l'utilisateur peut créer, modifier ou supprimer un objet.

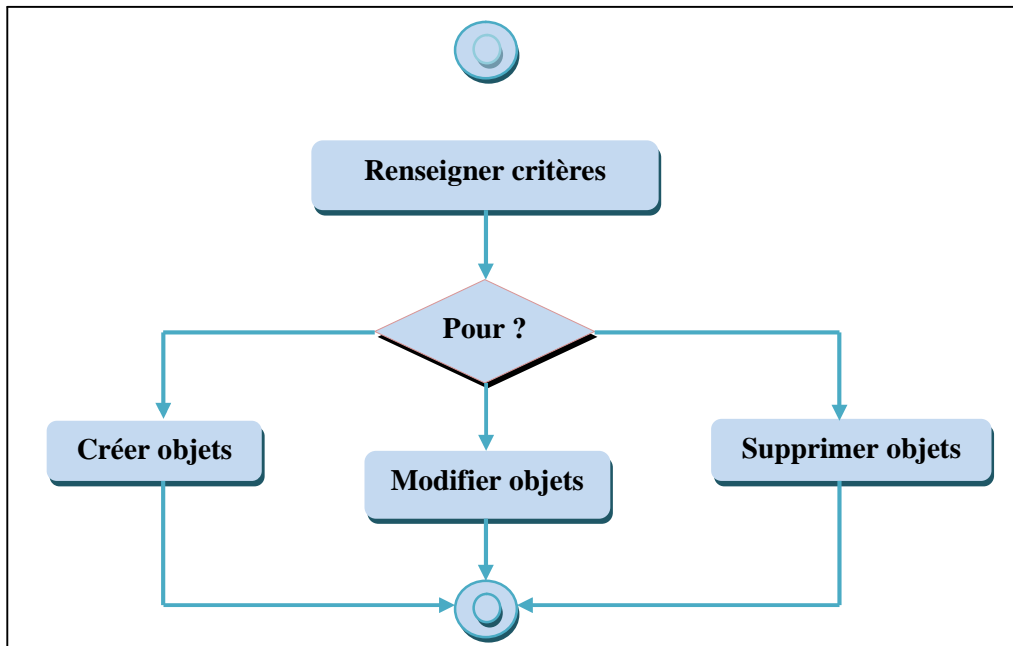


Figure 29 : Diagramme d'activité du cas d'utilisation « manipuler des objets »

VI. 3.3.2. Cas d'utilisation «Gérer la sécurité»

Titre : gérer la sécurité.

Exploitant : Informaticien.

Intention : doter le système d'une sécurité le protégeant des intrusions et des accès malveillants.

Action : écran d'identification, vérification, validation/annulation, connexion.

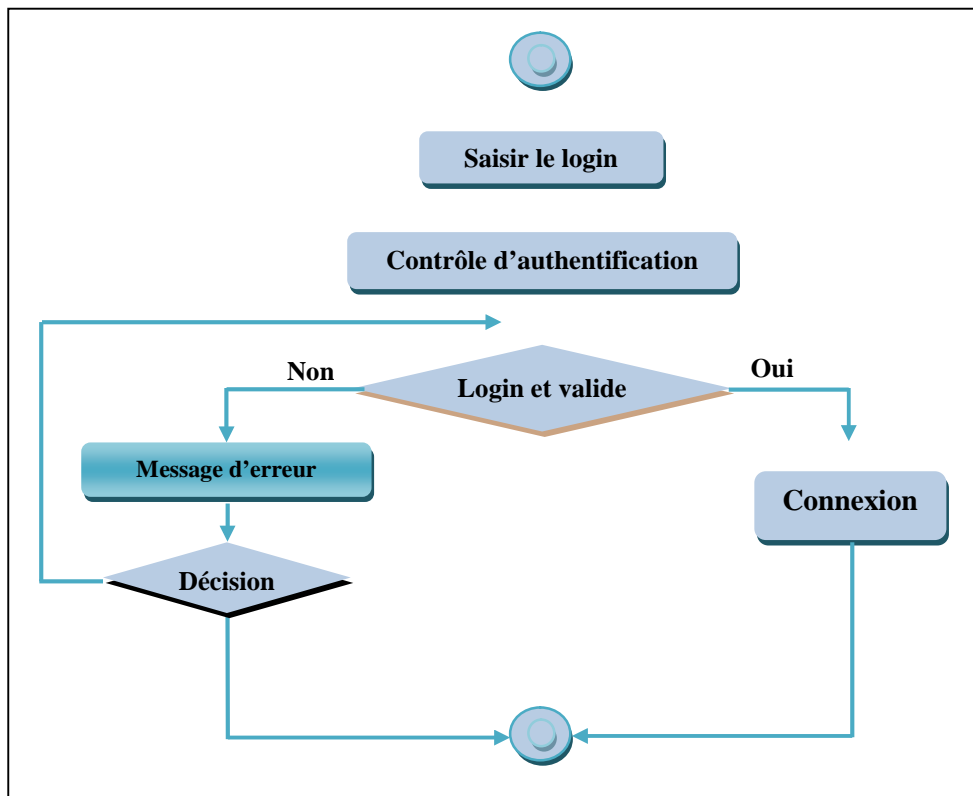


Figure 30 : Diagramme d'activité du cas d'utilisation « Gérer la sécurité »

VI. 3.3.3. Cas d'utilisation «Utiliser l'aide»

Titre : consulter l'aide.

Exploitant : l'utilisateur.

Intention : permettre aux utilisateurs d'utiliser le système d'une manière efficace.

Action : si l'utilisateur trouve des difficultés dans l'exploitation du système il doit trouver à sa disposition une aide pour résoudre le problème.

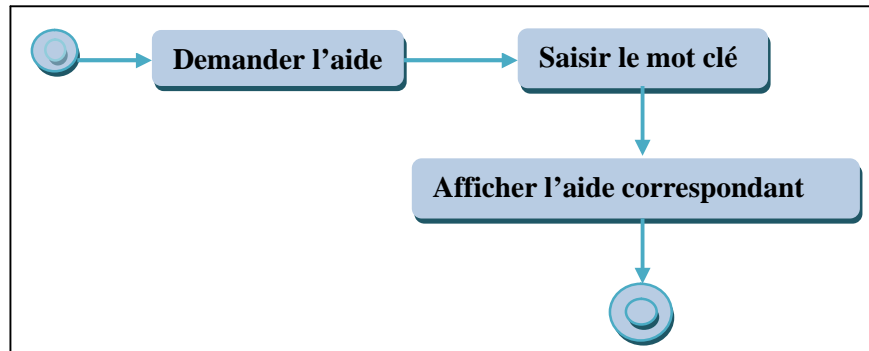


Figure 31 : Diagramme d'activité de « Utiliser l'aide »

VI. 3.3.4. Cas d'utilisation «Gérer les erreurs»

Titre : gestion des erreurs.

Exploitant : l'utilisateur, l'informaticien.

Intention : avoir la possibilité de gérer les erreurs générées par le système pendant son exploitation.

Action : afficher des messages d'erreurs (alertes, traces,...).

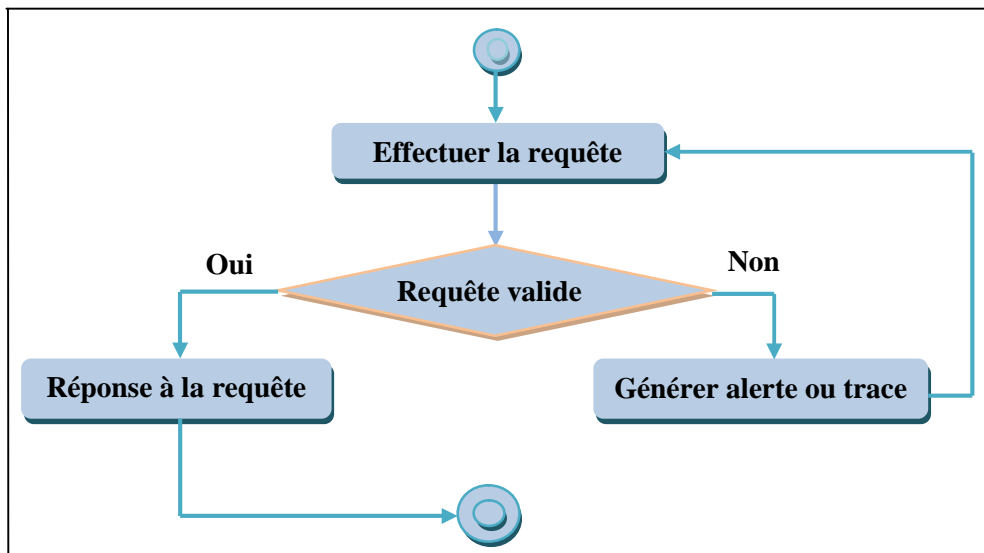


Figure 32 : Diagramme d'activité de « Gérer les erreurs »

VI. 4. Conclusion

Dans ce chapitre nous avons défini l'architecture client/serveur de notre application. Notre choix c'est porté vers cette architecture car elle est particulièrement recommandée pour les réseaux nécessitant un grand niveau de fiabilité.

Le chapitre suivant sera consacré à l'analyse de notre système en se basant sur les principes de l'approche orienté objet.

Chapitre VII :

Analyse

VII. 1. Introduction

L'étape d'analyse consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

Pour passer à l'analyse, il faut se baser sur les principes de l'Approche Orientée Objet, notamment celle de l'encapsulation. À cet effet, il faut passer d'une structuration fonctionnelle via les cas d'utilisations, à une structuration objet via les classes et les catégories.

VII. 2. Découpage des classes en catégories

Cette phase marque le démarrage de l'analyse objet du système à réaliser. Elle utilise la notion de *package* pour définir des catégories de classes d'analyse et découper le modèle UML en blocs logiques les plus indépendants possibles.

Le découpage en catégories constitue la première activité de l'étape d'analyse et elle va s'affiner de manière itérative au cours du développement du projet. Elle se situe sur la branche gauche du cycle en Y et succède à la capture des besoins fonctionnels.

Une catégorie est un regroupement de classes candidates sémantiquement voisines, fortement reliées entre elles plus faiblement reliées aux autres. **[ROQ, 07]**

Une classe est une description abstraite d'un ensemble d'objets qui partagent les mêmes propriétés (attributs et associations) et comportements (opérations et états). **[ROQ, 07]**

Dans le cas où notre système comporte plus que dix classes, il est utile de regrouper celles qui sont fortement couplées en unités plus grandes qui sont les catégories.

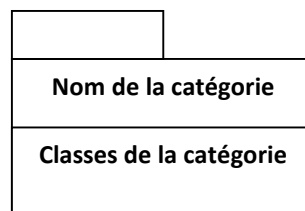


Figure 33 : Représentation graphique d'une catégorie

Le découpage en catégories est représenté dans la figure ci-dessous.

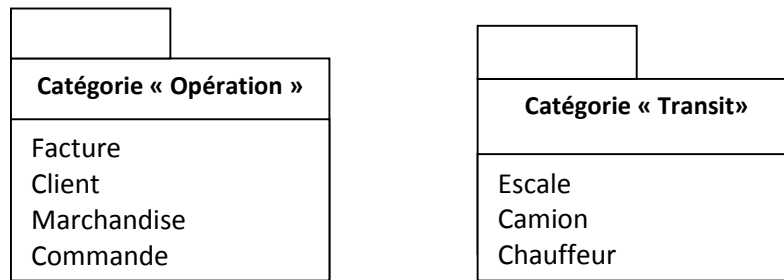


Figure 34 : Schéma présentatif des catégories de classes candidatent

VII. 3. Dépendances entre catégories

Le schéma suivant illustre l'état préliminaire des dépendances entre les différentes catégories au début de la phase d'analyse.



Figure 35 : Diagramme de package d'analyse

VII. 4. Développement du modèle statique

Les étapes essentielles pour l'élaboration d'un modèle statique sont les suivants :

- Affinement des classes : en éliminant les classes redondantes et en séparant les classes ayant trop de responsabilité.
- Affinement des associations : dans cette étape on élimine les associations redondantes, on définit les multiplicités et on précise les agrégations et les compositions.
- Ajouter des attributs : dans cette étape de multiples questions viennent à la tête comme : de quelle façon sont décrites les classes ? quelles informations caractérisent les états des objets ?
- Ajout des opérations : L'établissement de la liste des opérations est en fait l'aspect du modèle statique le plus imbriqué avec la construction du modèle dynamique (par la description des interactions entre objets).
- Optimisation avec la généralisation : A ce stade il est conseillé de rechercher les classes possédant des caractéristiques communes afin de généraliser (à moins que les classes générales n'aient été directement identifiées).

- Ajout de méta-classes ou classes déléguées : Il est fréquent qu'une analyse des responsabilités associées aux classes fasse apparaître le besoin (voire la nécessité) d'introduire de nouvelles classes.
- Ajout de contraintes : Pour finir le modèle statique, il est conseillé de rajouter des contraintes sur attributs.

VII. 4.1. Identification des classes, associations et attributs

Classe	Attribut	Méthode
Escale	Id_esc Date_dep Date_arriv Heure_dep Heure_arriv Km_dep Km_arriv Ville_dep Ville_dest Moy_trans	Ajouter_escale () Modifier_escale () Afficher_escale ()
Camion	Id_cam Num_imat Type Etat Marque Modèle Capacité	Ajouter_camion () Modifier_camion Supprimer_camion () Afficher_camion ()
Personne	Id_pers Nom Prenom Adresse Adr_mail Num_tel Num_fax	Ajouter_personne () Supprimer_personne () Modifier_personne ()
Chauffeur	Id_chauf Salaire Type_chauf Etat_chauf	Ajouter_chauffeur () Supprimer_chauffeur () Modifier_chauffeur () Afficher_chauffeur ()
Client	Code_client R_S	Ajouter_client () Supprimer_client () Modifier_client () Afficher_client ()
Marchandise	Num_serie Désignation Qte Tonage Unité_mes	Ajouter_marchandise () Modifier_marchandise () Afficher_marchandise ()
Facture	Num_fact Date_fact Num_bon_com Lieu_liv Prix_TTC	Saisir_facture () Afficher_facture ()
Commande	Num_com Lieu_liv Délai_liv Date_com	Ajouter_commande () Afficher_commande ()
Entrée_C	Date_entr	Entrée_camion ()

	Heure_entr	
Sortie_C	Date_sort Heure_sort	Sortie_camion ()

Tableau 14 : Classes d'objets et leurs attributs

VII. 4.2. Conception des modules

Module 1 « Transit »

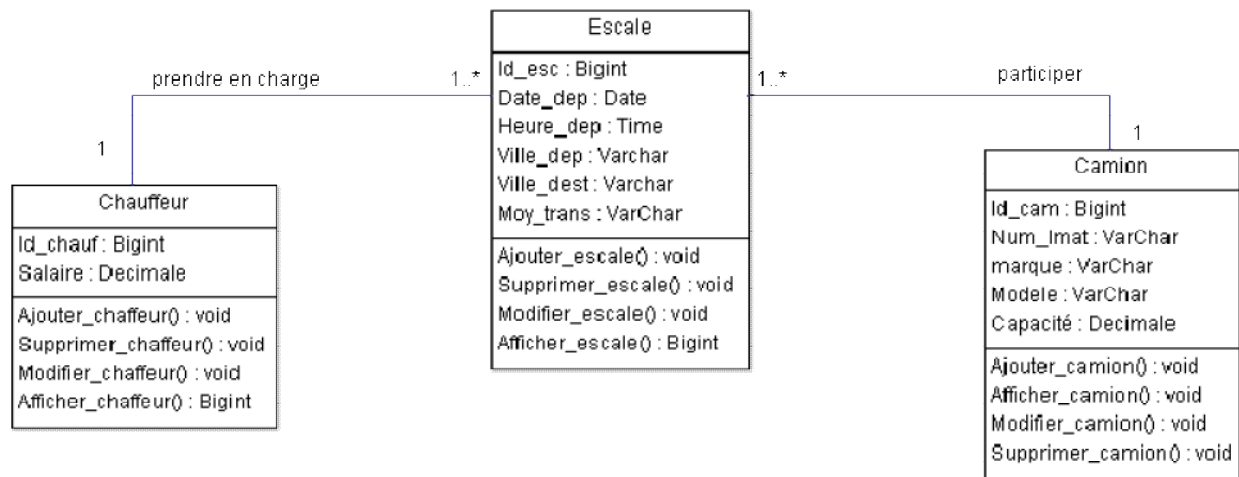


Figure 36 : Diagramme de classes associées au module « Escale »

Module 2 « Opération »

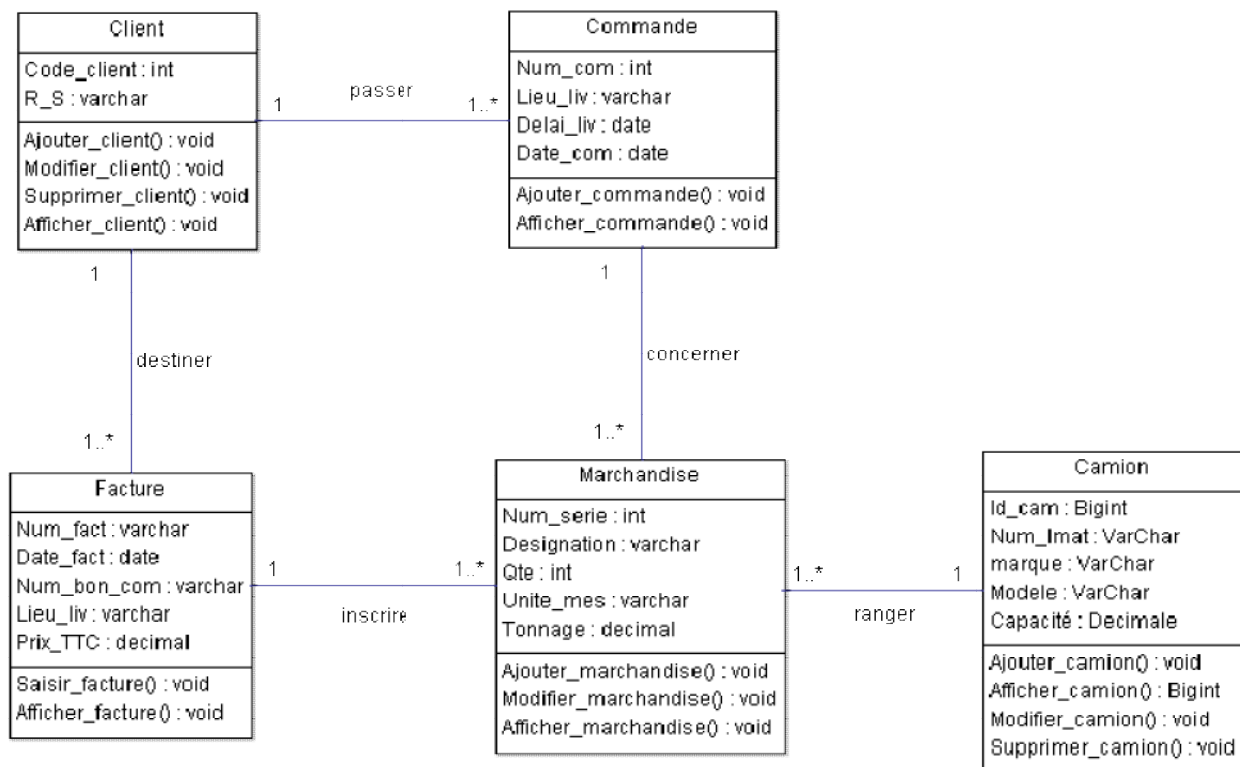


Figure 37 : Diagramme de classes associées au module « Opération »

VII. 5. Développement du modèle dynamique

Le développement du modèle dynamique constitue la troisième activité de l'étape d'analyse. Il s'agit d'une activité itérative, fortement couplée avec *la modélisation statique* et il est conseillé de construire le modèle dynamique de manière pratiquement parallèle avec le modèle statique.

Identification des scénarios

Un scénario est une suite spécifique d'interactions entre les acteurs et le système à l'étude. On peut dire que c'est une « instance » du cas d'utilisation, un chemin particulier dans sa combinatoire. [ROQ, 07]

Les échanges de messages entre objet peuvent être présentés en UML avec deux sortes de diagrammes :

- **Diagramme de séquence** : Montre la séquence verticale des messages passés entre objets au sein d'une interaction. [ROQ, 07]
- **Diagramme de collaboration** : Il permet de représenter le contexte d'une interaction et montrer les relations entre les objets jouant les différents rôles.

Nous allons présenter dans ce qui suit les diagrammes de séquence de notre système afin de donner une formalisation des cas d'utilisation vus précédemment.

Cas d'utilisation « S'authentifier »

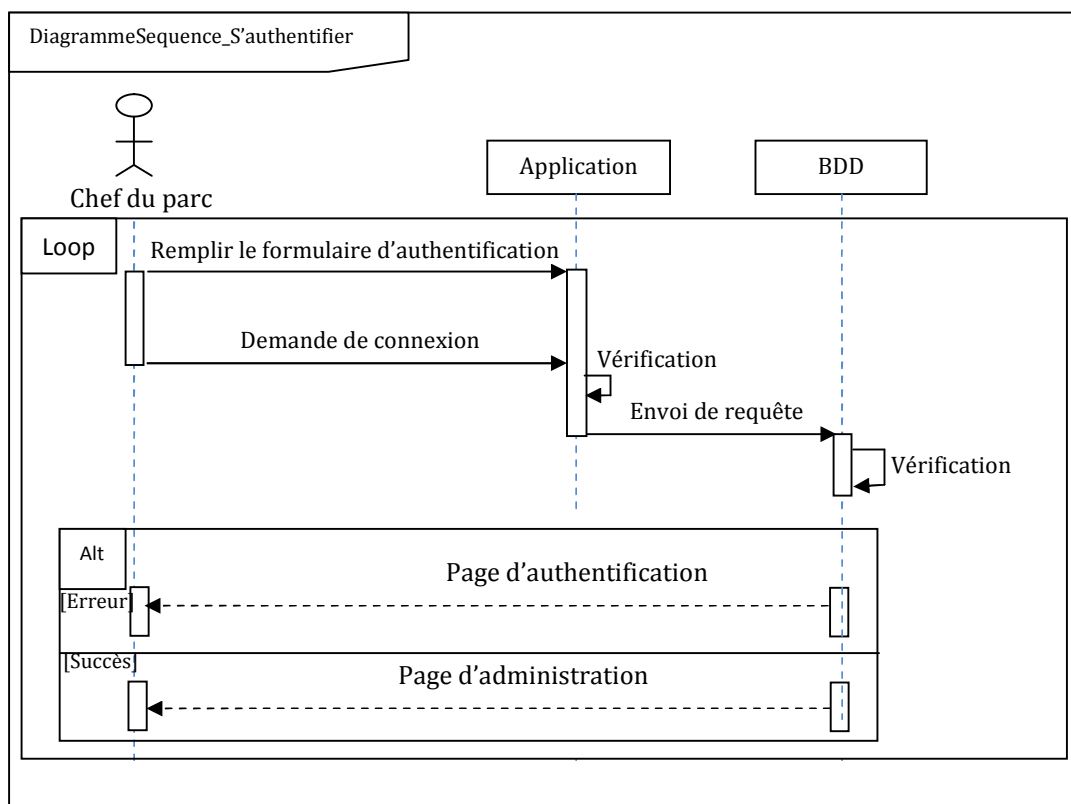


Figure 38 : Diagramme de séquence du CU « S'authentifier »

Cas d'utilisation « Consulter escale »

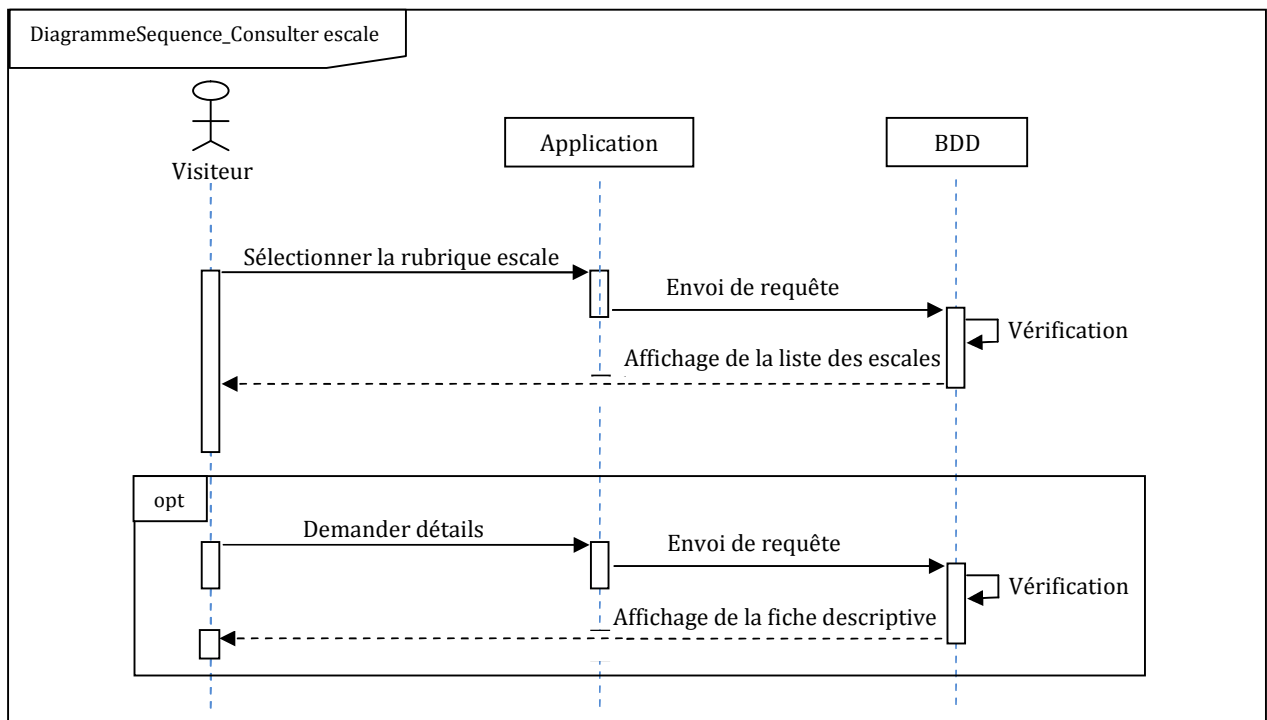


Figure 39 : Diagramme de séquence du CU « Consulter escale »

Cas d'utilisation « Gérer escale »

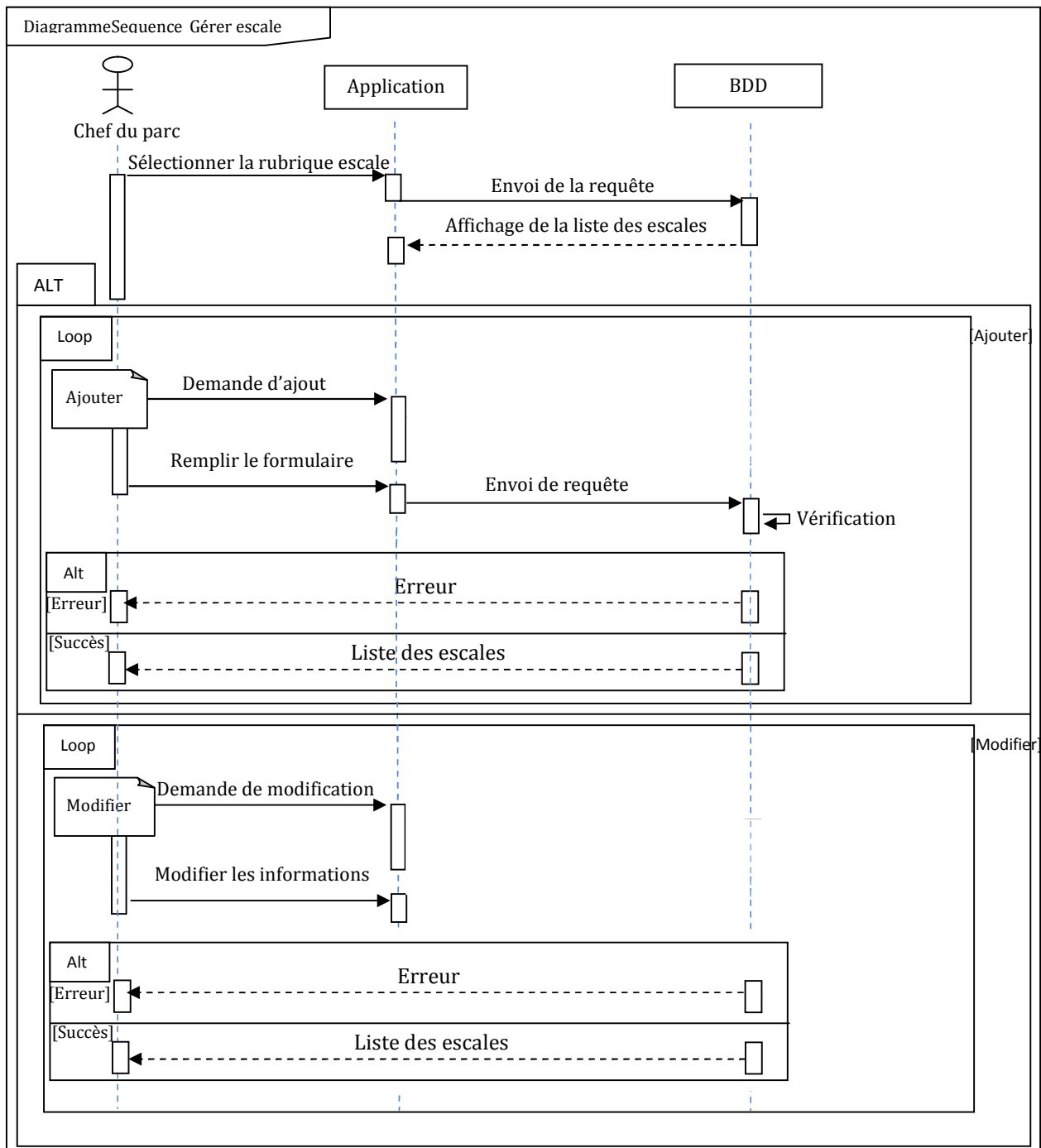


Figure 40 : Diagramme de séquence du CU « Gérer escale »

Cas d'utilisation « Ajouter camion »

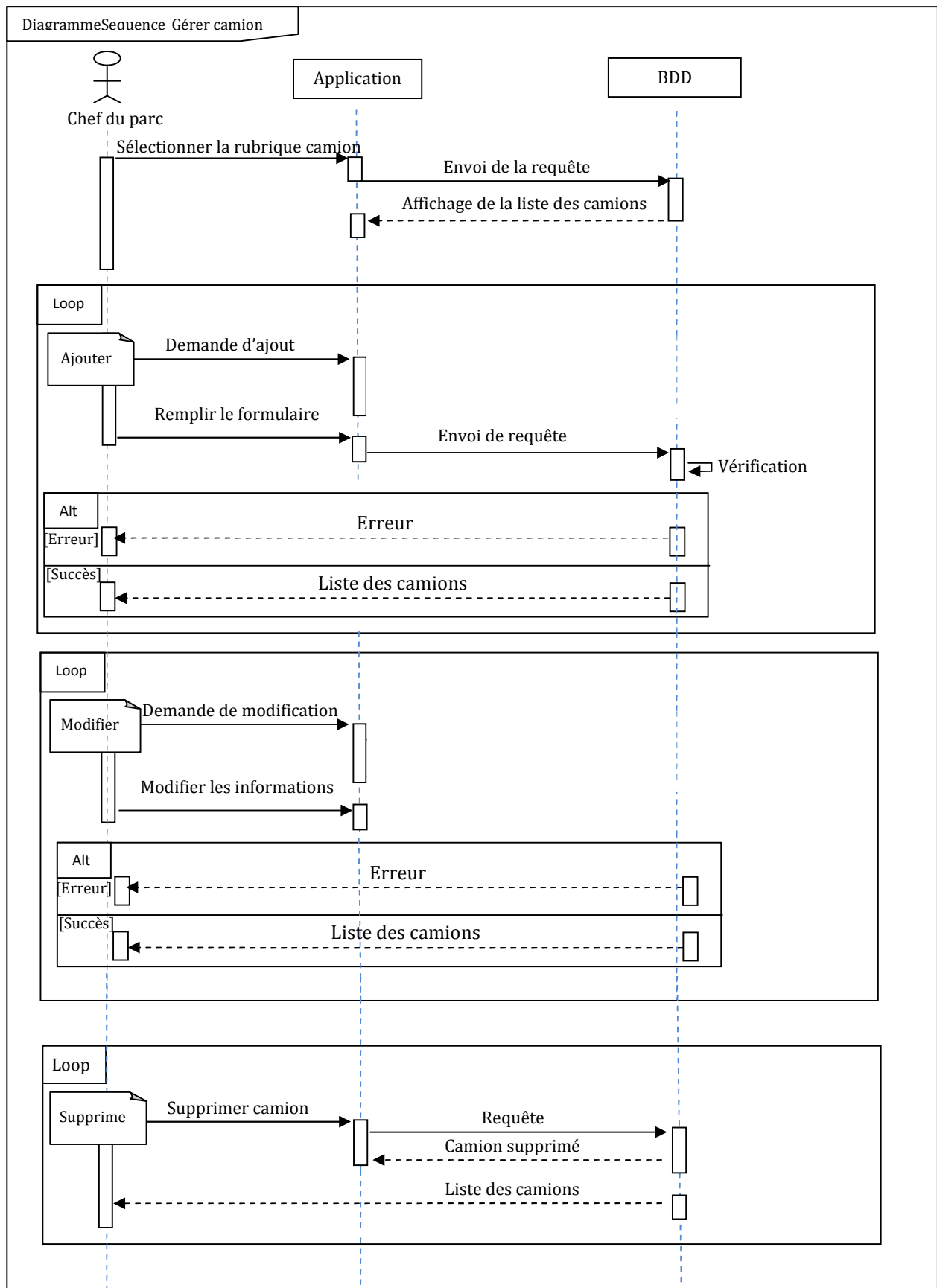


Figure 41 : Diagramme de séquence du CU «Ajouter camion »

Cas d'utilisation « Enregistrer entrée/sortie »

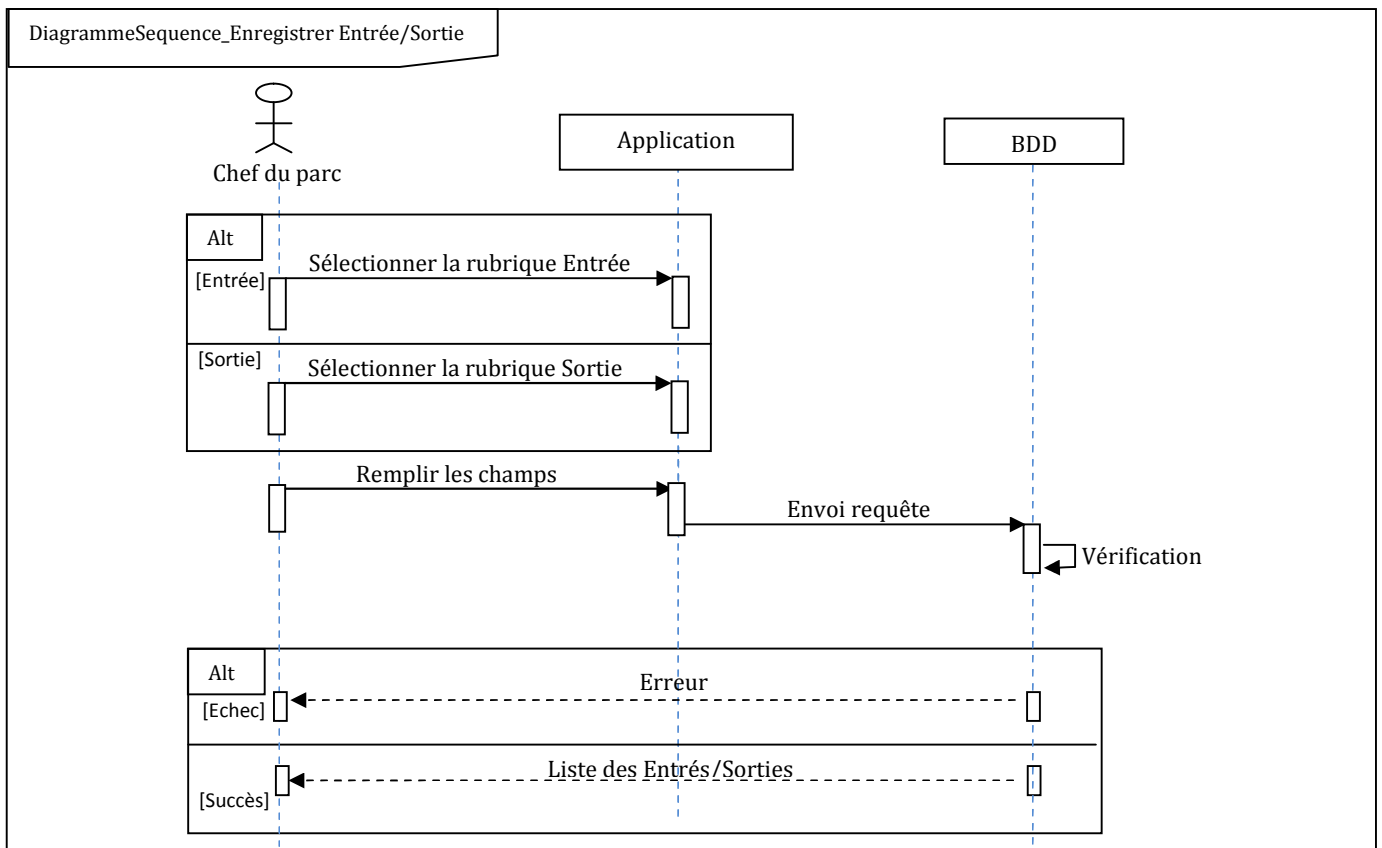


Figure 42 : Diagramme de séquence du CU « Enregistrer Entrée/Sortie »

Cas d'utilisation « Consulter l'état du parc »

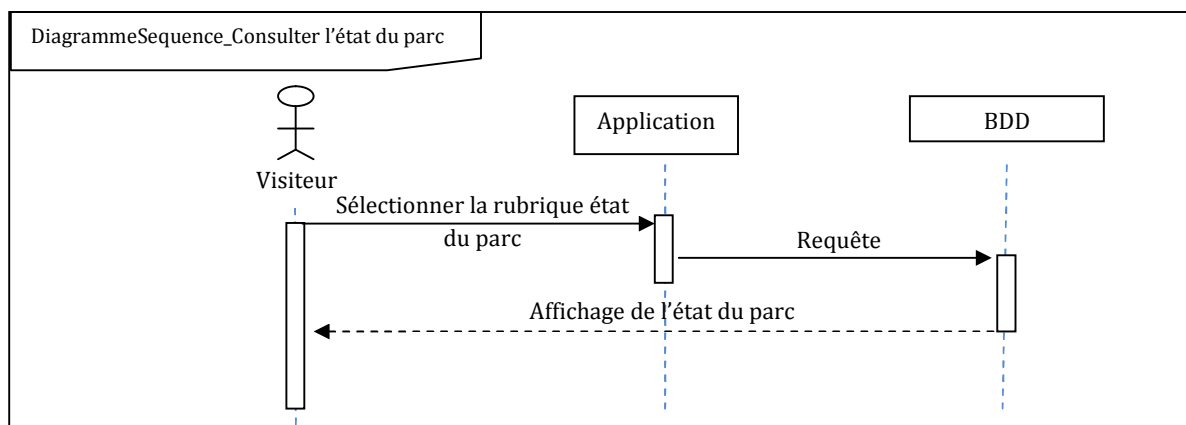


Figure 43 : Diagramme de séquence du CU « Consulter l'état du parc »

VII. 6. Conclusion

Dans ce chapitre nous avons abordé la partie analyse en découpant le système en catégories de classes, nous avons défini le modèle statique qui définit le squelette du système ainsi que le modèle dynamique qui nous présente les enchaînements réalisés par l'utilisateur. Le chapitre suivant sera consacré pour la conception de notre application.

Chapitre VIII : Conception

VIII. 1. Introduction

Après la phase d'analyse, vient l'étape où des décisions doivent être prises afin d'apporter une solution au problème de départ. D'abord à un niveau général dite générique qui définit les composants nécessaires à la construction de l'architecture technique puis à des niveaux de détail de plus en plus fins qui est la conception des objets dite préliminaire.

VIII. 2. Conception générique

La conception générique est une activité de la branche technique (droite) du processus en «Y» qui est entièrement indépendante des aspects fonctionnels. Elle s'appuie sur les couches logicielles déjà identifiées lors de la capture des besoins techniques.

En effet, elle consiste à développer la solution qui répond aux spécifications techniques en construisant les classes techniques qu'on regroupera dans des Frameworks techniques et en organisant ces Frameworks dans un modèle logique de conception technique.

VIII.2.1. Développement du modèle logique de conception

En analyse, nous avons fait un découpage en catégorie des classes, ici les classes techniques seront regroupées non pas dans des catégories mais dans des Frameworks techniques qu'on va organiser dans le modèle logique de conception technique.

A chaque couche logicielle définie dans l'étape de la capture des besoins techniques correspond un Framework technique, l'organisation de ses derniers permet d'élaborer le modèle logique de conception qui est schématisé dans la figure ci-dessous.

VIII.2.2. Définition du Framework technique

Un Framework technique est un réseau de classes qui collaborent à la réalisation d'une responsabilité qui dépasse celle de chacune des classes qui y participent. Il ne concerne que les responsabilités de la branche droite (technique) du processus.

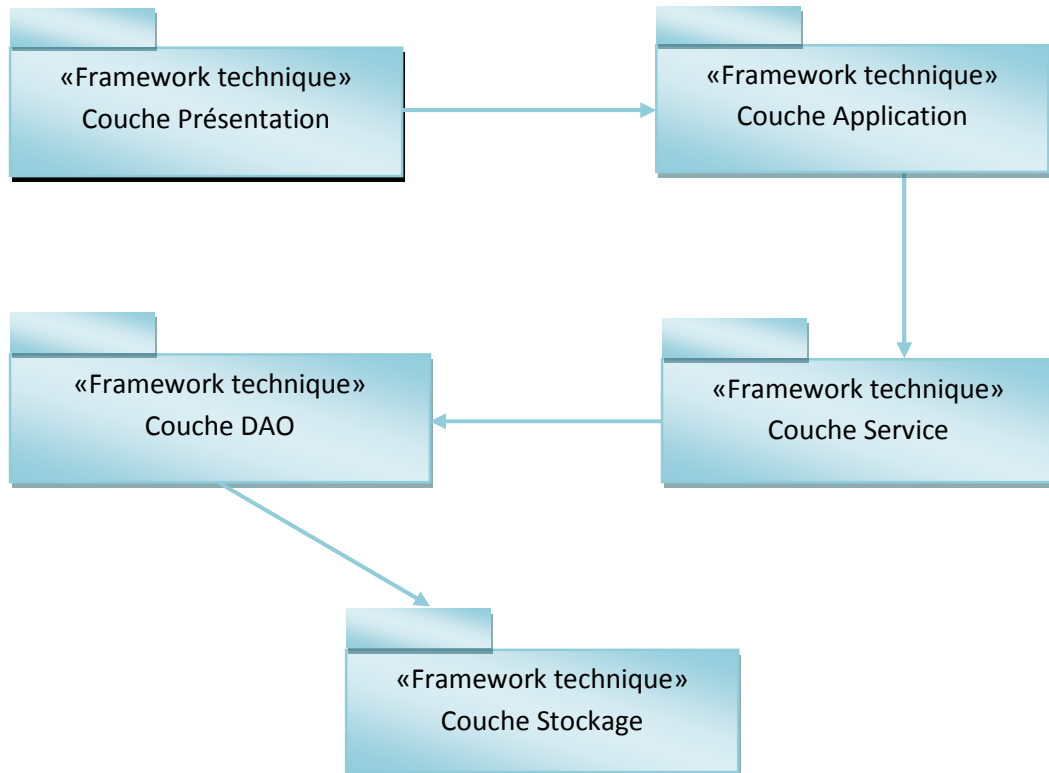


Figure 44 : Organisation du modèle logique de conception

VIII. 2.3. Description des couches

Couche présentation : Cette couche contient les composants qui doivent interagir avec l'utilisateur de l'application, comme les pages Web et les formulaires.

Rôles	Affichage des pages Web
	Gestion de l'interaction avec l'utilisateur
	Validation syntaxique

Tableau 15 : Rôles de la couche présentation

Couche application : Son principal but est de fournir des services spécifiques à la couche Présentation. Ces services correspondent aux règles du métier définies lors de la phase d'analyse. Elle prend en charge les aspects de contrôle des cas d'utilisation. La communication avec la couche supérieure se fait à travers les managed-beans.

Rôles	Contrôle des cas d'utilisation
	Gestion de la sécurité

Tableau 16 : Rôles de la couche application

Couche Service : La couche service reçoit les requêtes de la couche application et traite le logique métier contenu dans ces requêtes. C'est un package qui comporte les classes chargées, d'une part, de garantir la validation sémantique de l'information métier, et d'autre part, de gérer l'interaction avec la base de données. La communication avec la couche Application se fait à travers les interfaces services.

Le Tableau suivant résume les rôles de la couche Service :

Rôles	Comportement métier
	Validation sémantique

Tableau 17 : Rôles de la couche service

Couche DAO : Cette couche est certainement l'une des plus importantes, c'est ici où se trouve les fonctionnalités de base qui permettent de créer, rechercher, modifier et supprimer les entités métier dans le respect des propriétés transactionnelles. C'est également dans cette couche que les mécanismes de conversion objet/relationnel peuvent en partie prendre place.

Le tableau suivant résume les rôles de la couche DAO :

Rôles	Services basiques de Création / Lecture / Mise à jour / Suppression
	Conversion Objet/Relationnel

Tableau 18 : Rôles de la couche DAO

Couche Stockage : Cette couche est responsable du stockage physique de données. Elle assure le support transactionnel. En ce qui concerne notre système, cette couche se basera sur le modèle relationnel. La base de données est générée à partir d'un script SQL.

VIII. 3. Conception préliminaire

La conception préliminaire est certainement l'étape la plus délicate du processus 2TUP car elle représente le cœur. C'est là que s'effectue la fusion des étapes fonctionnelles et techniques ce qui implique que plusieurs activités doivent coexister.

Il convient de :

- ✦ Passer de l'analyse objet à la conception.
- ✦ Intégrer les fonctions métier et applicatives du système dans l'architecture technique.
- ✦ Adapter la conception générique aux spécifications fournies par l'analyse.

La conception préliminaire s'appuie sur les points de vue de spécification fonctionnelle et structurelle de l'analyse, mais également sur les Frameworks de la conception technique. Elle se termine lorsque la conception est organisée suivant son déploiement cible, son modèle d'exploitation et son modèle logique.

VIII. 3.1. Développement du modèle de déploiement

Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels. Les ressources matérielles sont représentées sous forme de nœuds. Les nœuds sont connectés entre eux, à l'aide d'un support de communication. Les diagrammes de déploiement peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds.

Le diagramme de notre système est représenté dans le schéma suivant :

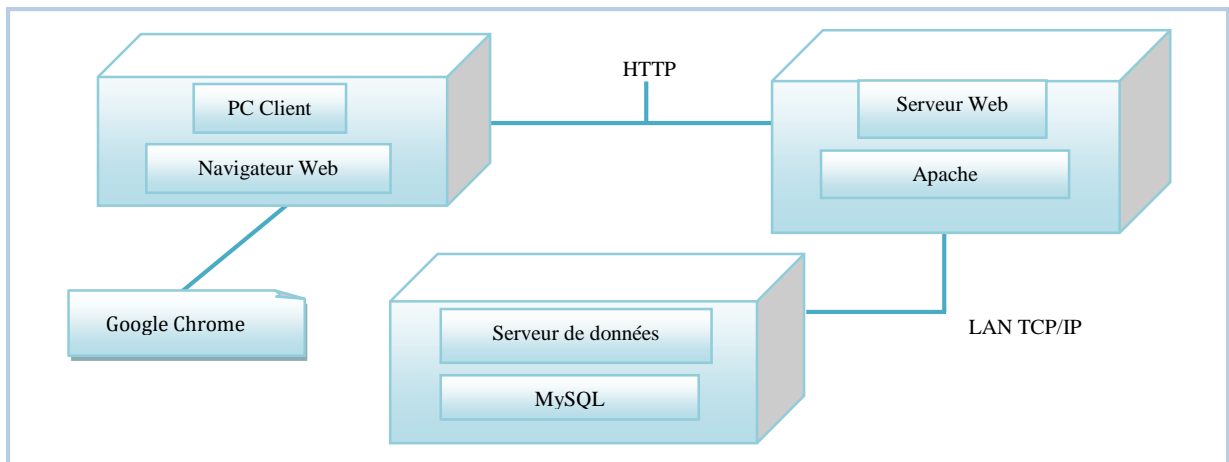


Figure 45 : Diagramme de déploiement

VIII. 3.2. Développement du modèle d'exploitation

Le modèle d'exploitation va définir les applications installées sur les postes de travail, les composants métiers déployés sur les serveurs et les instances de base de données implémentées sur les serveurs également.

Les composants d'exploitation sont :

- ⊕ «File» représente un fichier de code.
- ⊕ «Library» correspond à une librairie statique ou dynamique.
- ⊕ «Table» correspond à une table définie pour une base de données.

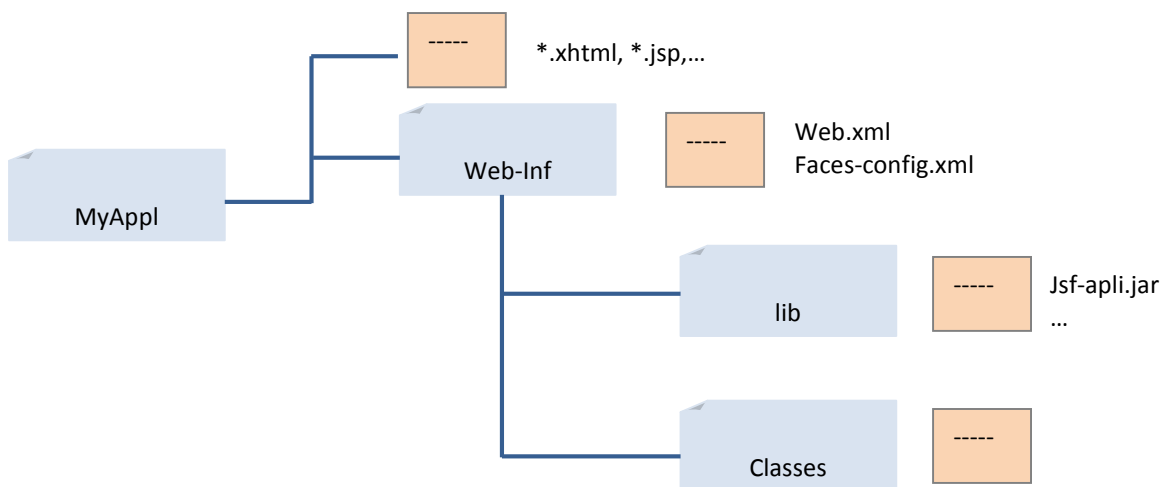


Figure 46 : Modèle d'exploitation

VIII. 4. Conception détaillée

La conception détaillée est la dernière phase de la modélisation avec UML. Après la modélisation des besoins et l'organisation de la structure de la solution, la conception détaillée vient construire et documenter précisément les classes, les interfaces, les tables et les méthodes qui constituent le codage de la solution.

Dans ce chapitre, nous allons traiter :

- ✚ La conception des classes.
- ✚ La conception des associations.
- ✚ La conception des attributs.
- ✚ La conception des opérations.
- ✚ Le passage au modèle relationnel.

VIII. 4.1. Concevoir les classes

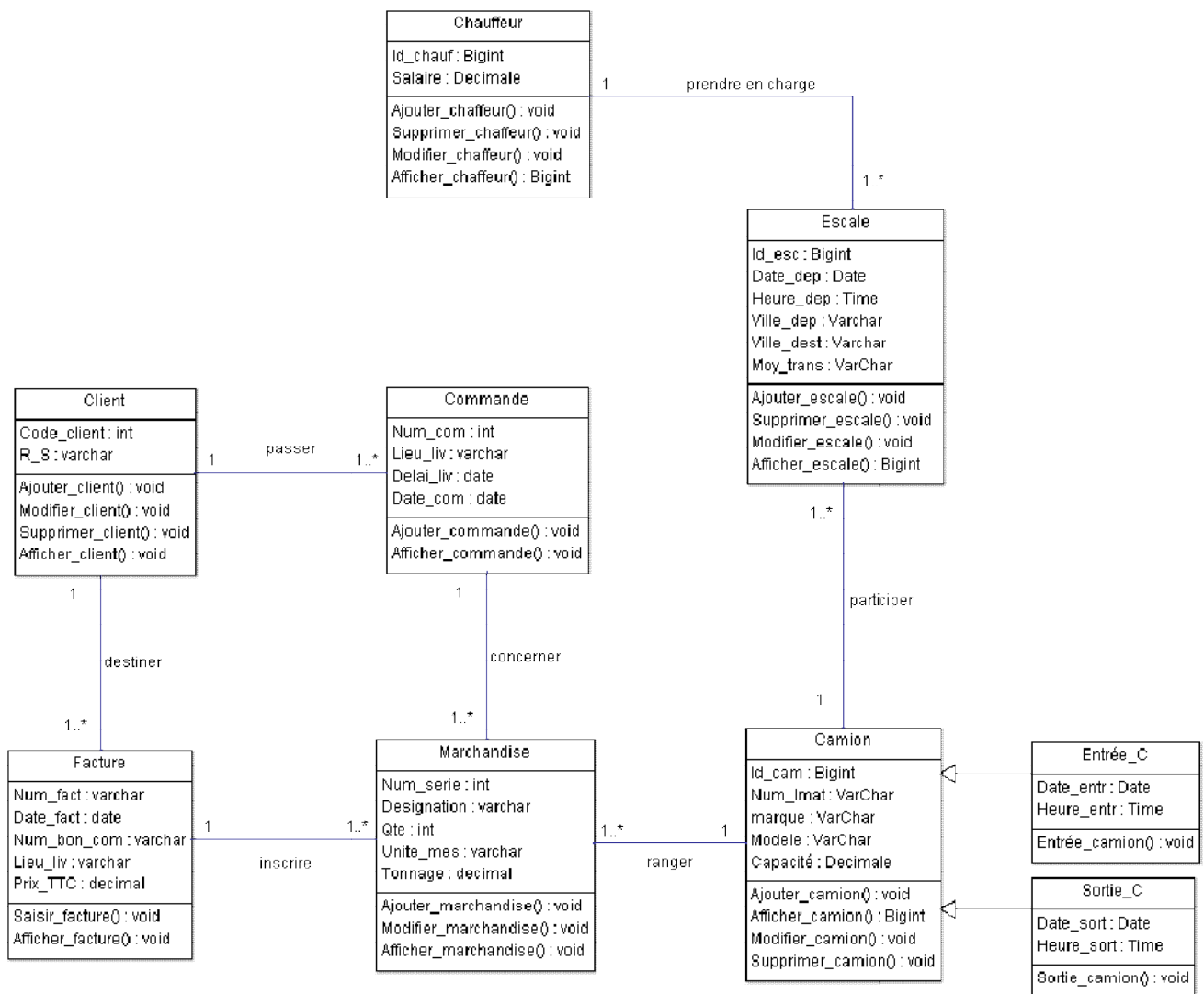


Figure 47 : Diagramme de classes général

Le tableau suivant représente les différentes classes de notre système ainsi que les différents et leurs désignations.

Classe	Attribut	Type	Désignation
Escale	Id_esc	Int(20)	Numéro de l'escale
	Date_dep	Date	Date de l'escale (Départ)
	Heure_dep	Time	Heure de départ
	Date_arriv	Date	Date d'arriver
	Heure_arriv	Time	Heure d'arriver
	Km_dep	Int	Kilométrage départ
	Km_arriv	Int	Kilométrage d'arriver
	Ville_dep	Varchar(30)	Ville de départ
	Ville_dest	Varchar(30)	Ville de destination
	Moy_trans	Varchar(30)	Moyen de transport
Camion	Id_cam	Int(20)	Code camion
	Num_imat	Varchar(30)	Numéro d'immatriculation
	Marque	Varchar(30)	Marque du camion
	Modèle	Varchar(30)	Modèle du camion
	Capacité	Int	Capacité du camion
	Type	Varchar(20)	
	Etat	Varchar(20)	
Personne	Id_pers	Int(20)	Code de la personne
	Nom	Varchar(30)	Nom de la personne
	Prenom	Varchar(30)	Prénom de la personne
	Adresse	Varchar(64)	Adresse de la personne
	Adr_mail	Varchar(100)	Adresse email
	Num_tel	Varchar(50)	Numéro de téléphone
	Num_fax	Varchar(50)	Numéro de fax
Chauffeur	Id_chauf	Varchar(5)	Code de chauffeur
	Salaire	Decimale(6,2)	
	Type_chauf	Varchar(30)	
	Etat_chauf	Varchar(30)	
Client	Code_client	Int(20)	Code de client
	R_S	Varchar(50)	Raison social
Marchandise	Num_serie	Int(20)	Numéro de série
	Designation	Varchar(35)	Désignation de la marchandise
	Qte	Int(4)	Quantité de la marchandise
	Unité_mes	Varchar(10)	Unité de mesure de la marchandise
	Tonage	Decimale(2,2)	Tonnage de la marchandise
Facture	Num_fact	Varchar(20)	Numéro de facture
	Date_fact	Date	Date de la facture
	Num_bon_com	Varchar(20)	Numéro de bon de commande
	Lieu_liv	Varchar(50)	Lieu de livraison
	Prix_TTC	Decimale(15,2)	Montant total
Commande	Num_com	Int(20)	Numéro de la commande
	Lieu_liv	Varchar(50)	Lieu de livraison
	Délai_liv	Date	Délai de livraison
	Date_com	Date	Date de la commande
Entrée_C	Date_entr	Date	Date d'entrer d'un camion
	Heure_entr	Time	Heure d'entrer d'un camion
Sortie_C	Date_entr	Date	Date de sortie d'un camion
	Heure_entr	Time	Heure de sortie d'un camion

Tableau 19 : Liste des classes

VIII. 4.2. Concevoir les opérations

A ce niveau nous allons donner une image assez précise du contenu des méthodes de notre projet, le tableau ci-dessous regroupe la description des différentes méthodes.

Classe d'objets	Méthodes	Description
Escale	Ajouter_escale()	Ajouter une escale
	Modifier_escale()	Modifier une escale
	Afficher_escale()	Afficher une escale
Camion	Ajouter_camion()	Ajouter un camion
	Supprimer_camion	Supprimer un camion
	Afficher_camion()	Afficher un camion
	Modifier_camion()	Modifier un camion
Personne	Ajouter_personne()	Ajouter une personne
	Modifier_personne()	Modifier une personne
	Supprimer_personne()	Supprimer une personne
Chauffeur	Ajouter_chauffeur()	Ajouter un chauffeur
	Modifier_chauffeur()	Modifier un chauffeur
	Supprimer_chauffeur()	Supprimer un chauffeur
	Afficher_chauffeur()	Afficher un chauffeur
Client	Ajouter_client()	Ajouter un client
	Modifier_client()	Modifier un client
	Supprimer_client()	Supprimer un client
	Afficher_client()	Afficher un client
Facture	Saisir_facture()	Saisir une facture
	Afficher_facture()	Afficher une facture
Entrée Sortie	Entrée_camion()	Signaler l'entrée d'un camion
	Sortie_camion()	Signaler la sortie d'un camion

Tableau 20 : Liste des opérations

VIII. 4.3. Passage au modèle relationnel

Après avoir identifié les objets persistances de notre système, nous allons construire à partir de cette description conceptuelle un modèle relationnel.

Le modèle relationnel permet d'organiser des données dans des relations. Une relation est un ensemble de tuples, chaque tuple est défini par un ou plusieurs attributs.

Pour réaliser ce passage de l'UML vers le relationnel, nous avons suivi des règles strictes et précises qui permettent de traduire le contenu conceptuel du diagramme de classes en modèle relationnel. Ces règles sont les suivantes :

- ☞ **Règle 1** : Toute classe d'entité devient une relation.
- ☞ **Règle 2** : Classe d'association binaire de type $X..1 \sim X..*$, la clé primaire de la relation correspondante à la classe d'entité ayant la cardinalité $X..*$, se voit ajoutée comme clé étrangère dans la relation correspondante à l'autre classe d'entité.
- ☞ **Règle 3** : Classe d'association de type $X..* \sim X..*$ donne naissance à une nouvelle relation, chaque attribut de la classe d'association devient un attribut du schéma de la

relation créée. La clé primaire est la concaténation des clés primaires des relations correspondantes aux classes d'entités qui interviennent dans la classe d'association.

∞ **Règle 4** : Classes d'association de type 1..1 ~ 1..1, on ajoute un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un.

En appliquant ces règles de transformation d'un diagramme de classe vers un modèle relationnel, nous avons abouti au schéma suivant :

Personne (**Id_pers**, Nom, Prenom, Adresse, Adr_mail, Num_tel, Num_fax)

Escale (**Id_esc**, Date_dep, Date_arriv, Heure_dep, Heure_arriv, Ville_dep, Ville_dest, Moy_trans, Km_dep, Km_arriv, **Id_cam#**, **Id_chauf#**)

Camion (**Id_cam**, Num_imat, Marque, Modèle, Capacité, Type, Etat)

Chauffeur (**Id_chauf**, Salaire, Type_chauf, Etat_chauf)

Client (**Code_client**, R_S)

Marchandise (**Num_serie**, Designation, Qte, Unité_mes, Tonage, **Id_cam #**, **Num_com#**, **Num_fact#**)

Facture (**Num_fact**, Date_fact, Num_bon_com, Lieu_liv, Prix_TTC, **Code_client#**)

Commande (**Num_com**, Lieu_liv, Délai_liv, Date_com, **Code_client#**)

Entrée (Date_entr, Heure_entr)

Sortie (Date_sort, Heure_sort)

VIII. 5. Conclusion

Dans ce chapitre on a bien suivi la démarche de conception en respectant la modélisation qui a été élaborée dans le chapitre précédant et c'est dans le but de répondre aux objectifs déjà fixés.

Dans ce qui suit, nous entamerons la phase de réalisation dont nous présenterons les choix concernant la technologie utilisée pour la réalisation de notre système.

Chapitre IX : Réalisation du prototype

VIII. 1. Introduction

Dans la réalisation de notre application web nous avons suivi une des architectures web toute en faisant appel à un ensemble d'outils et langages de développement.

Dans ce qui suit, nous décrirons l'architecture de notre service web. Ensuite nous présenteront les outils de développement ainsi que les mécanismes utilisés pour sécuriser le système. Enfin, nous présenterons les différentes fonctionnalités offertes par l'application.

VIII. 2. Architecture de l'application

Face aux différentes technologies des applications web, notre choix c'est apporté vers la technologie JSF qui est un Framework de type MVC destiné aux applications web respectant l'architecture J2EE.

Le premier objectif de JSF, est de procurer un environnement de développement permettant de construire une interface de type web, sans devoir toucher au code HTML et JavaScript. Ceci est réalisé par la mise en place d'un mapping entre l'HTML et les objets concernés. JSF est donc basé sur la notion de composants, comparable à celle de Swing, ou l'état de ces composants est sauvegardé puis restauré au retour de la requête. [W10]

JSF permet :

- ↻ Une séparation nette entre la couche de présentation et les autres couches.
- ↻ Le mapping HTML/Objet.
- ↻ Un modèle riche de composants graphiques réutilisables.
- ↻ Une gestion de l'état de l'interface entre les différentes requêtes.
- ↻ Une liaison simple entre les actions côté client de l'utilisateur et le code Java correspondant côté serveur.
- ↻ La création de composants customs grâce à une API.
- ↻ Le support de différents clients (HTML, WML, XML, ...) grâce à la séparation des problématiques de construction de l'interface et du rendu de cette interface.

Architecture de base d'une application web avec la technologie JSF

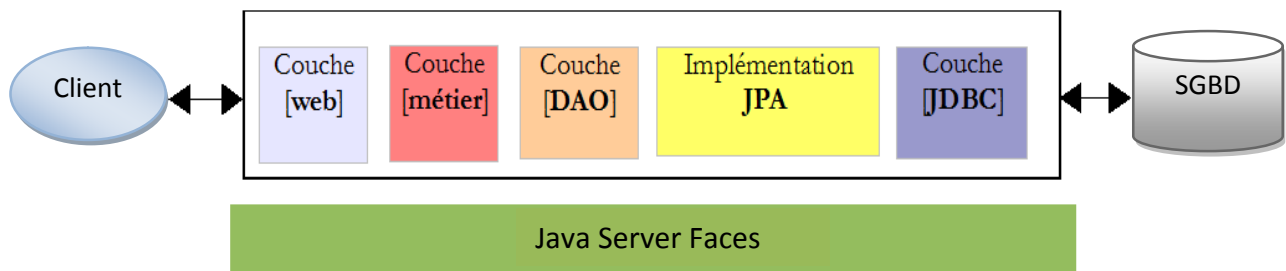


Figure 48 : Architecture de base d'une application web avec la technologie JSF

- ☞ **Couche interface utilisateur (web)** : est la couche qui dialogue avec l'utilisateur, Elle a pour rôle de fournir des données provenant de l'utilisateur à la couche ou bien de présenter à l'utilisateur des données fournies par la couche.
- ☞ **Couche métier** : est la couche qui applique les règles dites métier, la logique spécifique de l'application, sans se préoccuper de savoir d'où viennent les données qu'on lui donne, ni où vont les résultats qu'elle produit.
- ☞ **Couche d'accès aux données** : est la couche qui fournit à la couche des données préenregistrées (fichiers, bases de données, ...) et qui enregistre certains des résultats fournis par la couche.
- ☞ **Implémentation JPA** : est une couche ORM, définie comme une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java.
- ☞ **Couche JDBC** : est la couche standard utilisée en Java pour accéder à des bases de données. Elle isole la couche [dao] du SGBD qui gère la base de données.

La figure suivante représente l'architecture de la technologie JSF.

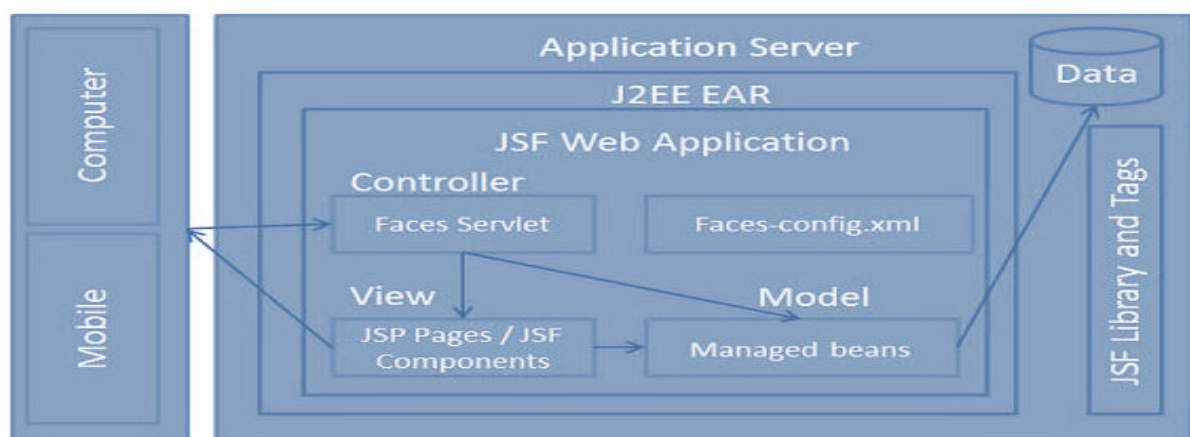


Figure 49 : Fonctionnement d'une page JSF

D'après la figure ci-dessus :

- ☞ Le modèle est représenté par les classes Java, Les Java beans sur le serveur.
- ☞ Les vues sont représentées par des composants Java sur le serveur, transformées en pages HTML sur le client.

☞ Le contrôleur est un servlet qui intercepte les requêtes http liées aux applications JSF et qui organise les traitements JSF.

Afin de réaliser et développer notre application toute en respectant cette architecture, le choix du langage s'est porté vers Java, qui étant orienté objet à la base. Pour plus de confort lors de la programmation il est préférable d'utiliser un environnement de développement intégré ou IDE. Notre choix s'est porté vers Netbeans qui nous permet de concevoir des applications complexes plus rapidement, avec une plus grande assurance de robustesse et de concevoir des applications qui résisteront à l'épreuve du temps.

VIII. 3. Présentation des outils de développement

VIII. 3.1. Présentation Java [BARM, 07]

Apparu fin 1995 début 1996 et développé par Sun Microsystems Java s'est très rapidement taillé une place importante en particulier dans le domaine de l'internet et des applications client-serveur.

Les objectifs de java sont d'être multiplateformes et d'assurer la sécurité aussi bien pendant le développement que pendant l'utilisation d'un programme java. Son point fort qui le démarque des autres est sa portabilité due (en théorie) à ses bibliothèques de classes indépendances de la plateforme ce qui est le point essentiel de la programmation sur internet ou plusieurs machines dissemblables sont interconnectées.

VIII. 3.2. Présentation de la plateforme de développement J2EE

Java EE » signifie *Java Enterprise Edition*, et était anciennement raccourci en « J2EE ». Il fait quant à lui référence à une extension de la plate-forme standard. Autrement dit, la plateforme Java EE est construite sur le langage Java et la plate-forme Java SE, et elle y ajoute un grand nombre de bibliothèques remplissant tout un tas de fonctionnalités que la plate-forme standard ne remplit pas d'origine. L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications. [W13]

VIII. 3.2.1. Présentation JSP

Une JSP est un fichier contenant du code HTML et des fragments de code Java exécutés sur le moteur de Servlet. Les pages JSP sont converties en Servlet par le moteur de Servlet lors du premier appel à la JSP. [BARM, 07]

VIII. 3.2.2. Présentation Servlet

Les Servlet forment l'un des composants JEE les plus utilisés. Elles permettent de gérer des requêtes http et de fournir au client une réponse http et forment ainsi la base de la programmation Web JEE. [W12]

VIII. 3.2.3. Présentation JavaScript

JavaScript est un langage de programmation orienté objet de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur.

VIII. 3.4. Présentation GlassFish

GlassFish est le nom du serveur d'applications Open Source Java EE 5 et désormais Java EE 6. Sa partie *TopLink persistence* provient d'Oracle. C'est la réponse aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération. [W14]

VIII. 3.5. Description de MySQL

C'est un Système de Gestion de Bases de Données Relationnelles (abrégé SGBDR). C'est-à-dire un logiciel qui permet de gérer des bases de données, et donc de gérer de grosses quantités d'informations.

WampServer [W15]

C'est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.

VIII. 4. Sécurité du système

La sécurité du système informatique est devenue l'un des éléments clés de la continuité des systèmes d'information de l'entreprise, quelle que soient son activité, sa taille ou sa répartition géographique.

Assurer la sécurité veut dire assuré :

- œ L'authentification : vise à garantir l'identité de la personne.
- œ L'intégrité : protéger l'information contre toute altération ou destruction.
- œ La confidentialité : garder l'information visible seulement par les personnes autorisées.
- œ La disponibilité : garantir le bon fonctionnement du système d'information au moment son utilisation.

VIII. 4.1. Principales menaces de la sécurité des systèmes d'information

Les attaques touchent généralement les trois composantes suivantes d'un système : la couche réseau, en charge de connecter le système au réseau, le système d'exploitation, en charge d'offrir un noyau de fonction au système, et la couche applicative, en charge d'offrir des services spécifiques.

Détruire, altérer, accéder à des données sensibles dans le but de les modifier ou de nuire au bon fonctionnement des réseaux : les motivations sont diverses et fonction de la nature des informations recherchées et de l'organisme visé.

Quelles formes prennent les attaques ? De qui émanent-elles ? Quelle est leur finalité ?

Les plus importantes des menaces sont les suivantes :

- **Menaces physiques :** Actes de délinquance (vols, détérioration) et accidents naturels sont des menaces physiques qui visent directement le matériel. Souvent ignorés, les évènements naturels, accidentels ou malveillants.
- **Menaces internes :** Bien moins identifiées, les menaces internes sont plus souvent liées à la négligence et à l'ignorance du personnel de l'entreprise. En général ces menaces correspondent à un usage personnel du matériel informatique de l'entreprise avec d'une part le risque d'infection par virus et d'autre part un risque pénal par le téléchargement de programmes ou de fichiers pirates qui sont incompatibles avec les applications professionnelles ou utilisés en dehors du cadre légal. Les cas d'espionnage industriel sont plus rares mais restent néanmoins un danger bien réel qui pèse sur les entreprises œuvrant sur des marchés stratégiques.

La sécurité des systèmes d'information doit passer alors par la mise en place d'une politique efficace de protection visant à empêcher toute possibilité d'action malveillante.

VIII. 4.2. Solutions élaborées pour la sécurité conçus

Après l'identification des différentes menaces aux quelles pourrait être confronté le système d'information, nous identifions dans ce qui suit les différents niveaux de sécurité :

- **Niveau base de données :** Consiste à gérer les privilèges des différents utilisateurs afin de protéger les informations confidentielles.
- **Niveau application :**
 - Contrôle des champs de saisie pour diminuer les fautes commises lors de la saisie.
 - Gestion des privilèges au niveau de l'application.
- **Niveau système d'exploitation :**
 - Installation des logiciels anti-virus.
 - Utilisation des Firewalls.
 - Utilisation des systèmes de détection d'intrusion.

VIII. 5. Présentation du prototype réalisé

La réalisation vient couronner les phases précédentes, donnant une forme concrète à la conception.

Afin de présenter le prototype réalisé, on utilise des prises d'écran pour illustrer les principales fonctionnalités du système.



Figure 50 : Page d'accueil de notre application

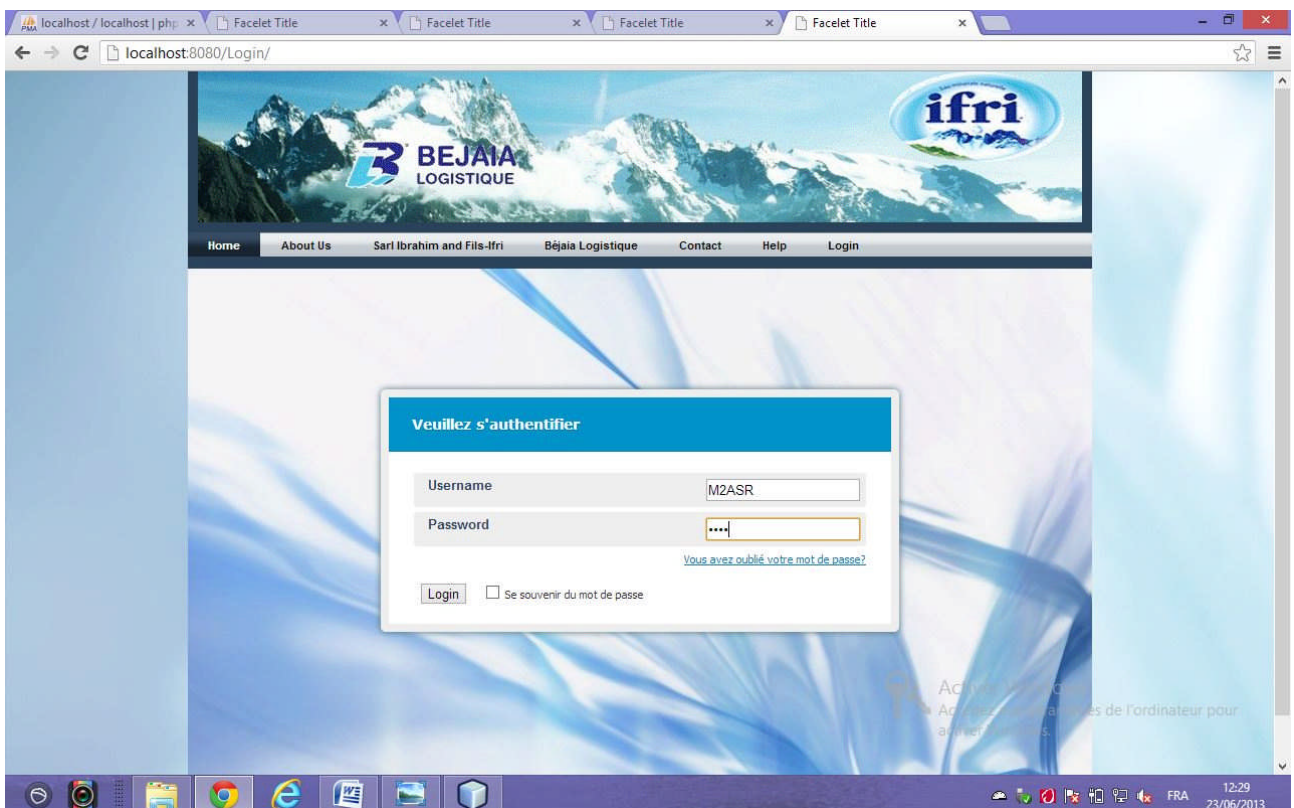


Figure 51 : Page d'authentification de notre application



Figure 52 : Page d'administration de notre application

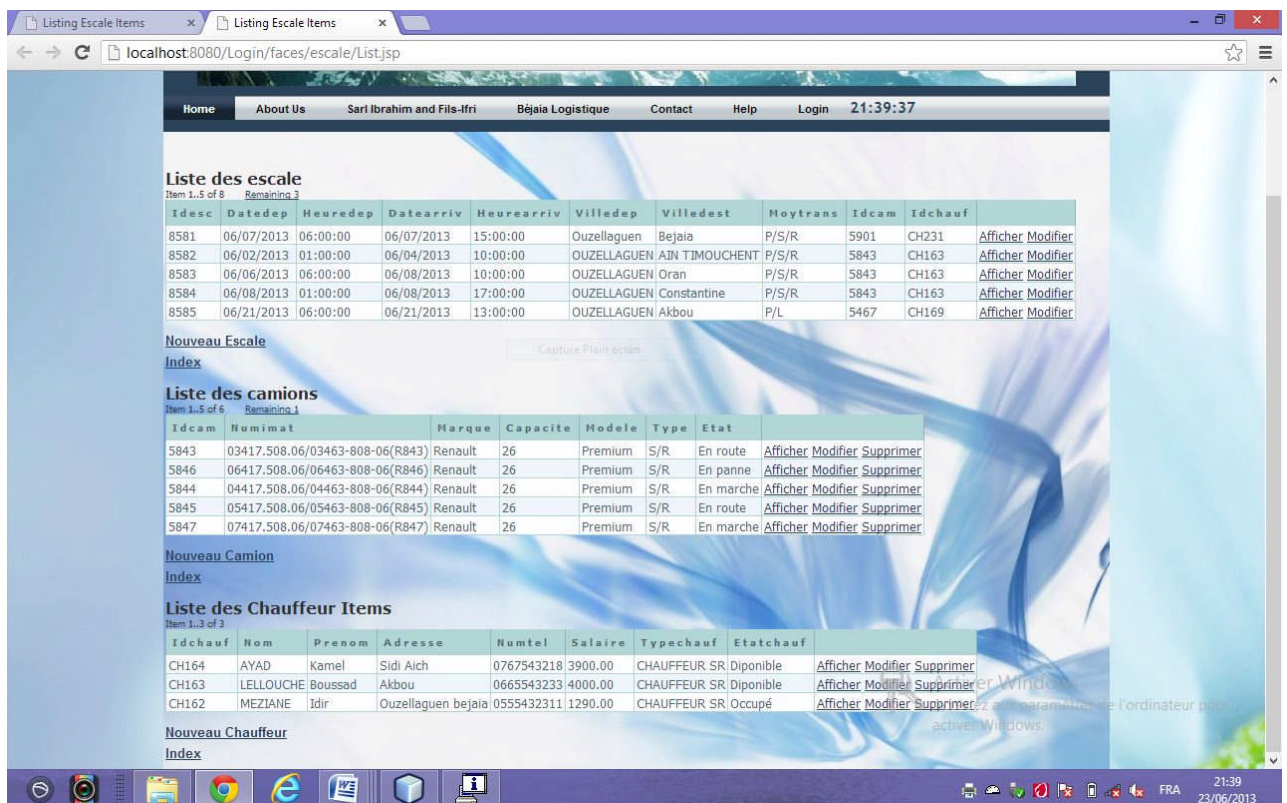


Figure 53 : Page de gestion de la rubrique « Transition »

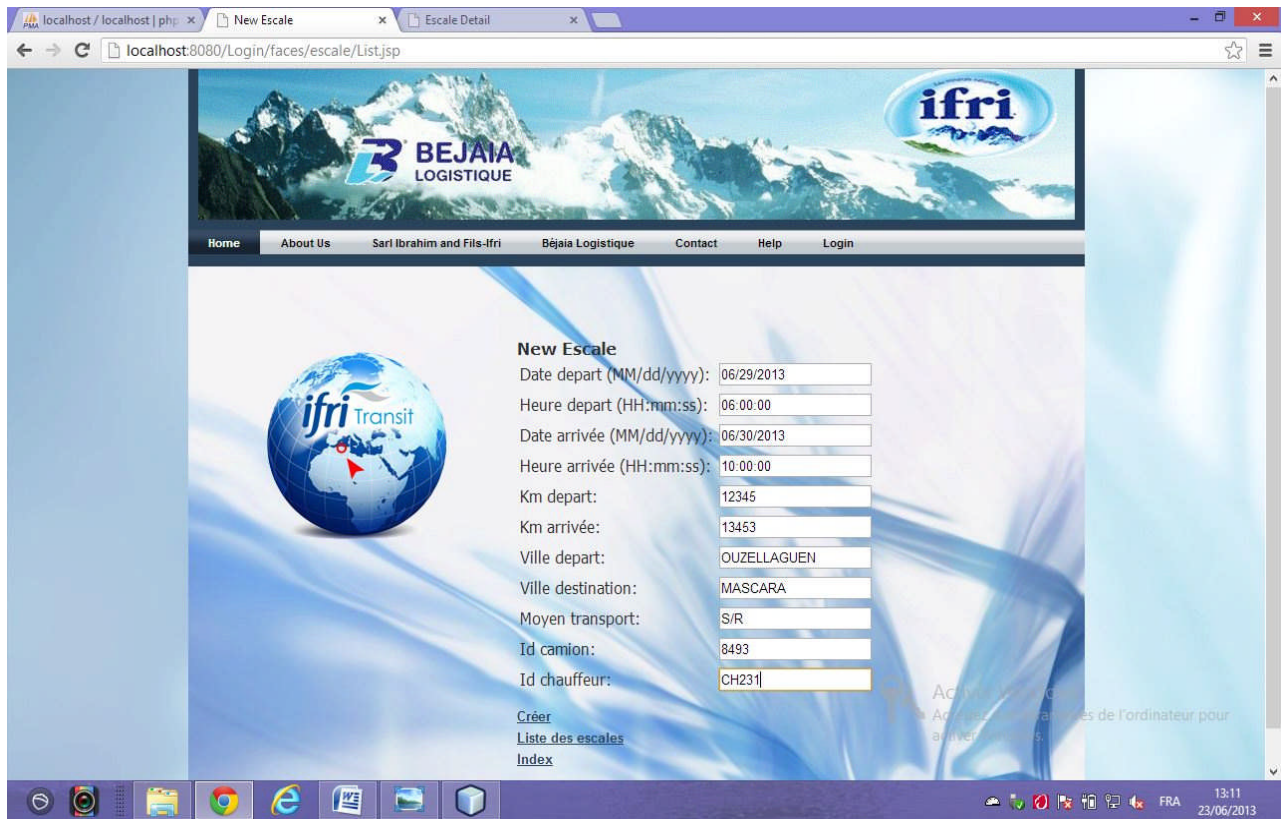


Figure 54 : Page d'ajout d'une escale

VIII. 6. Conclusion

On a pu présenter dans ce chapitre les différents outils qui nous ont permis la réalisation de notre application et les différentes fonctionnalités sous forme de fenêtres.

La réalisation de ce travail nous a permis de bien maîtriser des technologies et des outils de développement tel que Java et JSF.

Conclusion et perspectives

Ce projet représente l'aboutissement de un mois de stage au sein de la SARL IBRAHIM & FILS-IFRI de Bejaia qui nous a permis de mettre en pratique les connaissances acquises durant le cycle de notre formation, de se familiariser avec un environnement dynamique et d'avoir une idée plus profonde et plus pratique sur l'importance des systèmes d'information dans les entreprise.

La première étape de ce projet nous a permis en premier lieu d'élargir nos connaissances dans le domaine des applications web, pour ensuite étudier l'existant au niveau du groupe ifri, par le balayage des différents postes de travail, de l'étude approfondie des procédures et documents employés dans les différents services, afin de déceler les anomalies.

Ensuite, nous avons entamé la conception en utilisant un processus de développement unifié qu'est-le processus « 2TUP » qui est construit sur le langage de modélisation UML, ce qui nous a permis d'avoir une approche complète du développement logiciel.

En ce qui concerne la deuxième étape de réalisation, nous avons utilisé le SGBD MySQL et l'environnement de développement très performant Netbeans pour réaliser une application basée sur les spécifications J2EE.

Cette expérience professionnelle, nous a permis de faire face à la réalité de l'entreprise, apprendre à communiquer avec les utilisateurs et capturer leurs besoins, et orienté nos recherches.

Au terme de notre travail, nous espérons avoir mis en place un système d'information qui offre une solution qui répond aux besoins des utilisateurs.

Néanmoins, plusieurs questions se posent et à concrétiser comme perspectives futures, à savoir :

- ✦ L'intégration de l'application dans le système d'information de la SARL IBRAHIM & FILS-IFRI.
- ✦ Enrichissement de l'outil avec des fonctionnalités spécifiques à l'activité de l'entreprise.

Bibliographie:

[FAUD,08] M. Dumas, M.-C. Intergiciel et Construction d'Applications Réparties Fauvet, 12 juin 2008

[BERT] Patrice Bertrand, White Paper Conception d'applications web:efficacité et utilisabilité, version 1.2

[GAST,05] C.Lablanche, F.Seine et S.Gastaud, Les Web Services, Rapport de TE, 2005

[PERI,00] Eric PAPET, Plan Assurance Qualité, 2UP_Archi_DEV1, 12/12/2000

[ROQ,07] Pascal ROQUES, Modéliser une application web avec UML2, 2007

[DESM,06] Cyrille Desmoulins, UML Diagrammes d' 'activité, 2006

[BARM,07] Mickaël BARON Java pour le développement d'applications Web : Java EE, 2007

[ROB, 04] Robert Michel DI SCALA, L'essentiel de l'Informatique et de la programmation, 2004

[GLUC, 12] Olivier GLUCK, Architecture et communications Client/Serveur, 2012

Webographie:

[w1] <http://www.siteduzero.com/informatique/tutoriels/les-services-web/architecture-d-un-service-web>

[w2] <http://www.siteduzero.com/informatique/tutoriels/les-services-web/le-protocole-de-communication-soap>

[w3] <http://www.siteduzero.com/informatique/tutoriels/les-services-web/le-langage-de-description-wsdl>

[w4] <http://www.siteduzero.com/informatique/tutoriels/les-services-web/l-annuaire-des-services-uddi>

[w5] http://cours-info.iut-bm.univ-fcomte.fr/upload/supports/LP/ProgMultiTiers/2_20_presentations_generales_des_web_Services.pdf

[w6] <http://ceur-ws.org/Vol-547/77.pdf>

[w7] fcomte.fr/upload/supports/LP/ProgMultiTiers/2_20_Presentations_generales_des_web_Services.pdf

[w8] http://lgl.isnetne.ch/methodologie-2005/chap_06/chapitre6.pdf

[w9] http://www.info.arqendra.net/download.php?filename=Files%2F_UML_cours.pdf

[w10] http://gardeux-vincent.eu/Documents/ProjetJEE/HPV_JSF_Castor/jsf.html#jsf_intro

[w11] <http://fr.wikipedia.org/wiki/JavaScript>

[w12] <http://www.mistra.fr/tutoriel-jee-introduction/tutoriel-jee-composants.html>

[w13] <http://www.siteduzero.com/informatique/tutoriels/creez-votre-application-web-avec-java-ee/qu-est-ce-que-java-ee>

[w14] <http://fr.wikipedia.org/wiki/GlassFish>

[w15] <http://fr.wikipedia.org/wiki/WampServer>

Annexe :

SARL BEJAIA LOGISTIQUE
 Z.I AHRIK IGHZER AMOKRANE IFRI OUZELLAGUEN 06010 - BEJAIA -
 Tel : 034 35 12 66 Fax: 034 35 17 00

ORDRE DE MISSIONN° **10618**Du : **02/06/13**

Réf. Doc. **10407**
 Vehicule : **5843**
 Chauffeur **CH163**

Nom : **LELLOUCHE** Prénom : **BOUSSAD**
 Fonction : **CHAUFFEUR SR**

Objet de la mission : **Transport de DIVERS PRODUIT IFRI**Ville départ : **OUZELLAGUEN** Ville destination : **AIN TIMOUCHENT**Client : **SNC ALLOUL ASSOCIE BOISSONS**

Itinéraire :

Date départ : **02/06/2013** Heure départ : **00:00** Km départ : **551 784**Date arrivée : **04/06/2013** Heure arrivée : **10:00** Km arrivé : **553 044**Moyen de transport : **P/S/R**Immatriculation : **03417.508.06 / 03463-808-06(R843)**Contacte du client : **TEL: 0555-61-76-91**En cas d'urgence contacter : **CHEF DE GROUPE:0770744347 MAINTENANCE:0770119874**Le chef de parc**M RABHI**visa sécurité

2-6-13 P 6h30

5-6-13 7h30

9 4980

9

« Ordre de mission délivrée par la Bejaia Logistique »


Sarl IBRAHIM & FILS - ifri -

Production d' eaux Minérales & Boissons Diverses
 Sarl au Capital de : 1283 000 000,00 DA
 Siège Social : Z.I. AHRIK IGHZER AMOKRANE (w) BEJAIA (Algerie)
 Tél: 034-35-12-66 / Fax : 034-35-12-32 / 034 35 17 59
 Cpte C.P.A : 132 400 2173111-13 R.I.B : 004 00132 4002173111 13
 Cpte B.A.D.R. : 367 004245 90 300 0 00 R.I.B : 003 00367 004245300043
 Code Fiscal : 099806018281598 - Article d'imposition N° : 08360848615
 RC N° : 98 B 0182815 N.I.S : 0 995 0636 00283 31

FACTURE N° : F1306597	Doit : C01650 ZEKRI MOHAMMED FAOUZI
Date : 06/05/13	RC: 09 A 0357422-25/02
N° Bon de commande :	IF: 197825010884324
Délai de règlement (Jrs):	N.I.S :
Lieu de livraison :	N° Article : 25100255011
	Activité :
	12 RUE SEDIRA LA MTAICHE LOCAL N°2 CONSTANT CONSTANT AIN S'MAR CON

Désignation	UM	Quantité	Px unitaire	Remise	Montant (H.T)	TVA	Montant (TTC)
EAU MINERALE 1.50 L	Far	1	82,80		82,80	17%	98,88

ARRETEE LA PRESENTE FACTURE A LA SOMME DE :

Quatre-vingt seize Dinar (s), quatre-vingt huit Centime (s)

VISA DU FACTURIER

TOTAL PRODUIT FINI HT	82,80
MONTANT REMISE	0,00
TOTAL TRANSPORT HT	0,00
TVA 17%	14,08
TIMBRE FISCAL	0,00
MONTANT TOTAL T.TC	96,88
NET A PAYER	96,88

NB : Tout produit discounté ou promotionnel, n'est ni retourné ni échangé

« Facture délivrée par le service commercial »


Sarl IBRAHIM & FILS - ifri -

Production d' eaux Minérales & Boissons Diverses
 Sarl au Capital de : 1293 000 000,00 DA
 Siège Social : Z.I. AHRİK IGHZER AMOKRANE (w) BEJAIA (Algerie)
 Tél : 034-35-12-66 /Fax : 034-35-12-32 /034 35 17 59
 Cpte C.P.A : 132 400 2173111-13
 Cpte B.A.D.R. : 367 004245 90 300 0 00
 Code Fiscal : 099806018261598- Article d'imposition N° : 06360646615
 RC N° : 98 B 182615

FACTURE DE CONSIGNATION N° FC311517 Date : 10/06/13 Dépot de réception : EMB.PET	RC: 11 A 1849533-23/01 IF: 196323020000750 N° Article : 23020036645 Doit : C01700 MAZARI RABAH ZONE URBAINE LOT 7 DIV 2 LOCAL 3 EL BONI ANNABA EL BONI
-------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Désignation	UM	Quantité	Px unitaire	Montant (H.T)	Remise	TVA	Montant (TTC)
		0,00					

ARRETEE LA PRESENTE FACTURE A LA SOMME DE:

0,00

TRANSPORTEUR: **MENZOU SAMIR**
 MATRICULE: **811**

TOTAL CONSIGNATION	0,00
MONTANT TOTAL T.TC	0,00
NET A PAYER	0,00

VISA DU FACTURIER	VISA DU MAGASINIER

POIDS BRUT :

KGS

« Facture de consignation délivrée par le service commercial »

– Introduction Générale –

– Chapitre I –
Généralités sur les services
web

–Chapitre II –

Etude de l'existant

– Chapitre III –

Démarche adoptée

–Chapitre IV –

Etude préliminaire

–Chapitre V –

Capture des besoins
fonctionnels

–Chapitre VI –
Capture des besoins
techniques

–Chapitre VII –

Analyse

–Chapitre VIII –

Conception

– Chapitre IX –

Réalisation du prototype

– Conclusion et perspectives –

– Bibliographie et
Webographie –

Résumé :

L'objectif initial d'un système d'information est de permettre à plusieurs utilisateurs d'accéder aux mêmes informations. La technologie du web en pleine expansion depuis quelques années répond parfaitement à cette exigence. Il serait donc intéressant de disposer d'un outil qui regroupe différentes fonctionnalités, ce qui nous amène naturellement vers le concept d'application web qui peut parfaitement incarner cet outil.

Dans ce contexte, il est assigné de concevoir et de réaliser une application web basée sur la technologie J2EE, permettant la planification et la gestion du parc d'IFRI, allant le chargement des marchandises jusqu'à leur livraison aux clients. La réalisation d'un tel système nécessite, avant tout, d'appréhender et d'adopter les briques théoriques et conceptuelle servant à définir le cadre de travail. C'est ce que nous avons élaboré en nous appuyant sur la méthodologie itérative du 2TUP qui répond le plus à notre besoin.

Mots-clés : 2TUP, UML, Application web, JSF, Java, J2EE, Glassfish, MySQL, Architecture client serveur à trois tiers.

Abstract :

The initial objective of an information system is to allow multiple users to access the same information. Web technology booming in recent years meets this requirement perfectly. It would be interesting to have a tool that combines different features, which naturally leads us to the concept of web application that can perfectly embody this tool.

In this context, it is assigned to design and implement a web application based on J2EE technology, enabling the planning and management of the IFRI park, from the loading of the goods until they are delivered to customers. The realization of such a system requires, above all, to understand and adopt the theoretical and conceptual bricks used to define the working environment. This is what we developed building on the iterative methodology 2TUP that best meets our needs.

Keywords: 2TUP, UML, Web application, JSF, Java, J2EE, Glassfish, MySQL, Three-tier server-client architecture.