

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMERY de Tizi-Ouzou

Faculté de Génie électrique et d'informatique

Département informatique



Mémoire

de fin d'études

En vue de l'obtention du diplôme Master en informatique

Option : Conduite des Projets Informatique(CPI).

THEME

**Etude comparatif des Framework Java
(Application sur la gestion des Achat Locaux)**

Cas : ENIEM Oued Aissi

Réalisé par :

M^{elle}: SADOON Lydia

M^{elle} : HAMAZ Rachida

Dirigé par :

M^r: DIB Ahmed

Encadré par:

M^r: KABI Zidane

PROMOTION 2014-2015

Remerciements

Nous remercions d'abord le bon dieu qui nous a aidés et qui nous a donné le courage et la volonté pour réaliser ce modeste travail.

Nous tenons à exprimer nos sincères remerciements à notre promoteur Mr DIB Ahmed pour nous avoir fait confiance et pour nous avoir encouragés tout au long de ce projet.

Nous tenons à présenter nos remerciements à M^{er} KABI Zidane de nous avoir proposés un thème aussi intéressant, de nous avoir guidé tout au long de ce projet.

Nos profondes gratitudee et nos plus vifs remerciements à tous ceux qui ont voulu accepter de juger notre modeste travail.

Nous réservons ici une place particulière pour remercier vivement nos familles pour leur soutien continu. Et à tous les étudiants du département informatique, qui nous ont aidés et encouragés à la réalisation de ce modeste travail. En bref, merci à tous et à toutes.

Dédicaces

Je dédie ce modeste travail :

A mes chers parents qui ont toujours étaient présents pour moi, et à qui je souhaite une bonne santé et une longue vie pleine de bonheur Nchallah.

A mon future mari que j'aime énormément Hassan a toute la famille Ibersiene .

A mes chers frères que j'aime énormément : Madjid, Aziz et à mes cousins *que on était présent a cote de moi durant toute mes études .*

Et à mes adorables sœurs etcousines Djamila, Farida *et sa petit fille Amilia*, Dalila et sa petite fille Élina *et son nouveau née Ali*.

A mes meilleurs amis : Nouria, Roza, ma binôme *lydia* et sa *famille*

A toute ma famille hamaz .

Je remercie tous ceux et celles qui m'ont aidé à réaliser ce travail.

Rachida

Dédicaces

Je dédis ce modeste travail :

A mon très cher mari Farid que dieu me le garde

A mes très chers parents qui ont toujours été là pour moi,
Pour leurs attentions, sacrifice et soutien tout au long de
mes études, que Dieu les garde.

A ma très cher belle mère « Smina »

A la mémoire de mes chers grands parents

A la mémoire de mon beau père « Amar »

A ma grand-mère « FETTA »

A mes très chers frères « RAMDANE et MOKRANE »

A mes très chers sœurs « FAZIA, FETTA et KAHINA »
et leurs époux

A mes très chers beaux frères

A mes très chères belles sœurs

A mes neveux « DJAFER, MALIK, MAHREZ,
BOUKHALFA et MASSINISSA »

A mes nièces « KANZA, ALICIA, SARAH, CHANAZE
et KAMY

A toute ma famille « SADOUN »

A toute ma belle famille « SLAMANI »

A toutes mes copines en particulier « sassi »

A toutes mes amis (e)

A toute la section informatique 2014/2015

Lydia

Liste des figures

Figure I.1: l'architecture de J2EE	5
Figure I.2: Architecture M.V.C.	8
Figure II.3: Organigramme général de l'ENIEM.....	17
Figure II.4: Organigramme du l'Unité de Froid	18
Figure II.5 : Organigramme de services d'achat.....	18
Figure III.6 : Architecture de Hibernate	29
Figure III.7 : Architecture plus détaillé de Hibernate	30
Figure IV.8: Méthodologie de modélisation de notre projet	46
Figure I V.9 : Diagramme de contexte de notre système.....	47
Figure IV.10 : Diagramme du CU des fonctions liées à « Gestion des comptes ».....	50
Figure IV.11 : Diagramme du CU des fonctions liées à « Authentification »	50
Figure IV.12 : Diagramme du CU des fonctions liées à la " gestion des article".....	51
Figure IV.13 : Diagramme du CU des fonctions liées au « fournisseur »	51
Figure IV.14 : Description graphique du CU Gestion d'un « DAI »	52
Figure IV.15 : Diagramme du CU des fonctions liées à « Gestion des Factures »	52
Figure IV.16 : Diagramme du CU des fonctions liées à « Gestion des BC».....	53
Figure IV.17 : Diagramme du CU des fonctions liées à « Gestion des BR»	53
Figure IV.18 : Diagramme de séquence du cas d'utilisation « Authentification »	57
Figure IV.19 : Diagramme de séquence du cas d'utilisation « Ajout d'un compte»	58
Figure IV.20 : Diagramme de séquence du cas d'utilisation « Modification d'un Article»..	59
Figure IV 21 : Diagramme de séquence du cas d'utilisation « suppression d'un fournisseur» ..	60
Figure IV.22 : Diagramme d'activité du cas d'utilisation « Authentification».....	61
Figure IV.23 : Diagramme d'activité du cas d'utilisation « Ajout d'un compte»	62
Figure IV.24 : Diagramme d'activité du cas d'utilisation « Modification d'un Article»	63
Figure IV.25 : Diagramme d'activité du cas d'utilisation « Suppression d'un fournisseur»..	64

Figure IV.26 : Diagramme de classe globale.....	66
Figure V.27 : L'interface de Hibernate.	68
Figure V.28 : L'interface EasyPHP.....	69
Figure V.29 : L'interface PhpMyAdmin.....	69
Figure V.30 : L'interface de Netbeans.....	70
Figure V.31 : L'interface de JEE.....	71
Figure V.32 : Page d'authentification au système.	79
Figure V.33 : page d'accueil de l'application	79
Figure V.34 : Page de Suppression d'une demande d'achat interne (DAI).	80
Figure V.35 : Page Ajout d'un Article.....	80
Figure V.36 : Page de modification d'une réception.	81
Figure V.37 : Page de l'impression d'une commande.	81
Figure V.38 : Page de consultation des Fournisseur.	82
Figure V.39 : Page d' Ajout d'un utilisateur.....	82

Liste des tableaux

Tableau III.1 : Comparaison entre les différentes Framework.....	41
Tableau IV.2 : Tableau des cas d'utilisation.....	48
Tableau IV.3 : Spécification des scénarios des cas d'utilisation.....	49

Sommaire

Introduction générale.....

Chapitre I: Technologie JEE &WEB

I.	Introduction	1
II.	Le web	1
	II.1.Historique du web	1
	II.2.Définition du web:	1
	II.3.Avantage du web:	2
	II.4.Les langage de programmation utilisé pour le web :	2
	1. Au niveau du client :	2
	2. Au niveau du serveur	3
III.	JEE (Java 2 Entreprise Edition) :	4
	III.1.Historique du JEE :	4
	III.2.Définition du JEE :	4
	III.3.Fonctionnement Interne :	5
	III.4.L'architecture du JEE :	5
	III.5 .La plateforme J2EE :	6
	III.5.1.Les API de J2EE :	6
	A) Les composants :	6
	B) Les services :	6
	III.6. L'architecture MVC :	7
	III .6.1. Modèle : des traitements et des données :	8
	III.6.2.Vue : des pages JSP (Java Server Page):	8
	III.6.3.Contrôleur : des servlets :	9
	III .7. L'environnement d'exécution des applications J2EE :	9
	III.8.Avantages du JEE :	9
IV.	Conclusion :	10

Chapitre II: Présentation de l'organisme d'accueil :

I.	Introduction :	11
II.	Présentation de l'organisme d'accueil ENIEM :	11
	1. Historique de L'ENIEM :	11
	2. Situation géographique :	11
	3. Missions et objectifs de L'ENIEM	12
	3.1.Mission :	12

Sommaire

3.2. Objet :	12
4. Organisation général :	13
4.1. Les directions :	13
4.1.1. Direction générale :	13
4.1.2 Direction planification et contrôle de gestion :	13
4.1.3. Direction développement et partenariat :	13
4.1.4. Direction des finances et comptabilité :	14
4.1.5. Direction du marketing et la communication :	14
4.1.6. Direction industrielle :	14
4.1.7. Direction des ressources humaines :	14
4.2. Les unités :	15
4.2.1. Unité froid :	15
4.2.2. Unité cuisson :	15
4.2.3. Unité climatisation :	16
4.2.4. Unité prestation techniques (UPT) :	16
4.2.5. Unité commerciale :	16
5. L'organigramme de l'ENIEM	17
6. L'organigramme général de l'unité froid :	18
7. L'organigramme de service d'achat :	18
III. Etude de l'existence :	19
III.1. Quelques définitions :	19
III.2. Etudes des postes de travail :	19
III.3. Etudes des documents :	20
III.3.1 Liste des documents :	20
III.3.2 Fiches d'analyses des documents :	21
III.4. Organisation du travail :	24
III.4.1 Le diagramme de flux généré :	24
III.4.1.1 Les flux d'informations :	24
IV. Conclusion :	25

Sommaire

Chapitre III : Etude des Framework

I.	Introduction :	26
II.	Notion d'un Framework :	26
III.	Définition D'un Framework :	27
IV.	Objectif d'un Framework :	27
V.	Avantage des Framework :	28
VI.	Les Framework java :	28
VI.1.	Définition :	28
VI.2.	Les différents Framework java:	28
VI.2.1.	Hibernate :	28
VI.2.1.1.	présentation de Hibernate :	28
VI.2.1.2.	Cas d'utilisation :	31
VI.2.1.3.	Les Avantage de Framework Hibernate :	31
VI.2.2.	Java server faces(JSF) :	32
VI.2.2.1.	Présentation de JSF :	32
VI.2.2.2.	Cas d'utilisation :	33
VI.2.2.3.	Les Avantages de JSF :	33
VI.2.2.4.	Les Inconvénients de JSF:	33
VI.2.3.	Prime Faces :	34
VI.2.3.1.	Présentation de Prime Faces:	34
VI.2.3.2.	Cas d'utilisation de Prime Faces :	34
VI.2.3.3.	Les Avantages de Prime Faces :	34
VI.2.3.4.	Les inconvénients de Prime Faces :	35
VI.2.4.	Springs :	35
VI.2.4.1.	Présentation de Springs :	35
VI.2.4.2.	Cas d'utilisation de Springs :	35
VI.2.4.3.	Les Avantages de Springs :	36
VI.2.4.4.	Les inconvénients de Springs :	36
VI.2.5.	Qt Jambi :	36
VI.2.5.1.	Présentation de Qt Jambi :	36
VI.2.5.2.	Cas d'utilisation de Qt Jambi :	37

Sommaire

VI.2.5.3. Les Avantages de Qt Jambi :	37
VI.2.3. Sofi :	38
VI.2.3.1.Présentation de Sofi :	38
VI.2.3.2.Cas d'utilisation de Sofi :	39
VI.2.3.3. Les Avantages de Sofi :	39
VII. La comparaison entre les différents Framework :	41
VIII. Choix du Framework pour j2ee/java :	42
IX. Conclusion :	43

Chapitre IV : Analyse et Conception

I. Introduction :	44
II. Présentation de l'UML :	44
II.1. Définition :	44
II.2. Modélisation de l'UML :	44
Les diagrammes d'UML :	45
✓ Diagramme statique (Structurels):	45
✓ Diagramme dynamique (Comportementaux):	45
IV. Analyse et conception :	45
IV.1 Analyse :	45
IV.1.1 Spécification des besoins :	46
IV.1.2 Définition des besoins :	46
IV.1.2.1 Identification des utilisateurs du nouveau système :	46
A) Acteurs :	46
B) Le diagramme de contexte de notre application :	47
IV.1.2.2 Identification des cas d'utilisation :	48
IV.1.3 Diagramme des cas d'utilisation :	50
IV.1.4 Spécification de quelques cas d'utilisation :	54
IV.1.5 Diagramme de séquence :	56

Sommaire

IV.1.6 Diagrammes d'activités :	61
IV.1.7: Définition des classes	65
IV.1.8 Diagramme de classe global :	66
V. Conclusion :	67

Chapitre V : Réalisation

I. Introduction :	68
II. Les outils de développement	68
❖ Hibernate Framework	68
❖ EasyPHP :	68
❖ phpMyAdmin :	69
❖ Serveur Apache Tomcat :	70
❖ Netbeans :	70
III. Les langages de programmation :	71
❖ Java :	71
❖ Java EE :	71
❖ XHTML :	72
IV. Conception de la base de données :	72
V. Présentation de quelques interfaces de l'application :	79
❖ La page d'authentification au système	79
❖ Page d'accueil de l'application :	79
❖ Page de suppression d'une demande d'achat interne (DAI) :	80
❖ Page Ajout d'un Article :	80
❖ Page de modification d'une réception :	80
❖ Page de l'impression d'une commande :	81
❖ Page de consultation des Fournisseur :	82
❖ Page d'Ajout d'un utilisateur :	82
VI. Conclusion :	83
Conclusion générale	

Introduction Générale

Dans tous les domaines notamment la gestion, l'être humain s'est mobilisé afin d'exploiter le temps comme il se doit.

Le bon fonctionnement d'une entreprise se base, essentiellement sur une bonne organisation de ses services, éventuellement la gestion des achats locaux qui est l'élément crucial.

Dans notre cas, étant affectés à l'Entreprise Nationale des Industries en Electroménager (ENIEM) afin de développer une application pour garantir le bon fonctionnement du service gestion des achats.

Dans notre travail consiste à l'information de la gestion des achats de l'entreprise ENIEM et que le projet porte sur le développement d'une application Web qui répondra aux contraintes de fiabilité, efficacité et surtout de disponibilité.

Notre travail se décompose en 5 chapitres qui sont :

- Chapitre I : Technologie JEE & WEB.
- Chapitre II : Présentation de l'organisme d'accueil.
- Chapitre III : Etude des Framework.
- Chapitre IV : Analyse et conception.
- Chapitre V: Réalisation.

I. Introduction :

Depuis la fin du vingtième siècle on a remarqué que le monde a évolué rapidement dans les technologies de l'information en général et de l'informatique en particulier, cette évolution a touché tous les domaines de notre vie quotidienne et les effets provoquent un changement radical dans notre façon de communiquer.

Dans ce chapitre on va étudier les deux technologies le WEB et JEE.

II. Le WEB :

II.1.Historique du web: [1]

Dans les années 90, un nouveau service de l'Internet est apparu : le *World Wide Web*, la toile d'araignée mondiale, encore désignée par l'acronyme *WWW* ou le diminutif *Web*. C'est ce service qui assure un certain succès à l'Internet. L'idée est de lire des hyperdocuments à l'aide d'un navigateur.

Un hyperdocument est un document électronique contenant des images, du son, du texte, parfois des petits morceaux de programme, mais surtout des liens vers d'autres hyperdocuments : des liens hypertextes. Ces liens apparaissent dans un style qui les distingue, et une simple action de la souris sur un lien suffit à ouvrir le document lié. Les documents peuvent se trouver sur n'importe quelle machine (serveur) de l'Internet à des endroits parfois très éloignés et c'est ce qui donne l'impression à l'utilisateur de naviguer sur le réseau.

Le navigateur est l'outil qui permet de lire les hyper documents. On l'appelle aussi browser et les deux plus connus aujourd'hui sont *Microsoft Internet Explorer* (MSIE) et *Netscape*. Au début conçu pour ne lire que les hyper documents, le navigateur intègre aujourd'hui tous les services de l'Internet (*e-mail, ftp,...*)

Le navigateur désigne par une adresse URL (*Uniform Resource Locator*), les adresses complètes de l'Internet. C'est une adresse qui contient à la fois le nom d'une machine mais aussi le nom du service demandé, le nom d'un document,...

Un autre standard incontournable de l'Internet est HTML (*HyperText Markup Language*). C'est le langage qui permet d'écrire des hyper documents de façon descriptive à l'aide de marqueurs.

II.2.Définition du web:

Le World Wide Web, communément appelé le Web, parfois la Toile, littéralement la « toile d'araignée mondiale », est un système hypertexte public fonctionnant sur Internet et qui permet de consulter, avec un navigateur, des pages mises en ligne dans des sites. L'image de la toile vient des hyperliens qui lient les pages Web entre elles.

II.3.Avantage du web:[2]

- La simplicité et la rapidité d'implémentation et d'utilisation
- La gratuité des outils ou des coûts d'utilisation modique
- La grande interactivité
- Les possibilités de personnalisation des outils
- Possibilités de sauvegarder des documents de différents formats, d'annoter des documents, de laisser des marques (pour un accès ultérieur), d'avoir un historique de navigation...
- Accès direct aux serveurs www.
- Visualisation de documents HTML.
- Restitution de différents formats de fichiers sons, vidéo, graphiques...

II.4.Les langage de programmation utilisé pour le web : [3]

Il existe différents langages web, ils sont classifiés en deux grandes catégories :

1. Au niveau du client :

- HTML (Hyper Text Markup Language): C'est un langage qui permet de créer des pages web utilisant une structure formée avec des balises permettant la mise en forme du texte, enfin nécessite un navigateur web pour visualisation. Chaque page doit être créée au préalable et doit être manuellement mise à jour.
- XML (Extended Markup Language) : C'est Un langage aussi simple et élégant pour la création des documents auto descriptifs. Généralement, les pages web ne seront pas basées sur ce langage, mais celui-ci sera généré dynamiquement par un des langages de programmation web, et s'occupe de la standardisation du langage HTML et des feuilles de style entre autre.
- Applet : C'est une petite application, s'exécute par un navigateur, envoyée par un serveur web, et presque souvent s'écrit en java.
- Java Script : c'est un langage de programmation qui permet d'adopter des améliorations au langage HTML, ne nécessite pas de compilateur, il donne au client la possibilité d'exécuter des commandes, aussi le java script est dépendant du navigateur appelant la page web dans laquelle le script est incorporé donc dans un document HTML.

2. Au niveau du serveur :

- CGI (Common Gateway Interface) : En français veut dire « Interface de passerelle commune ». C'est une interface de programmation. Son utilité est de faire communiquer un serveur web avec un programme permettant d'effectuer des traitements de données, généralement utilisée par les serveurs HTTP.
- ASP (Active Server Pages) : Littéralement « fichier repérable par l'extension .asp ». C'est une technologie mise au point par Microsoft en 1996. il est sensé de développer des applications Web interactives.
- API (Application Programmable Interface) : C'est un ensemble de fonctions permettant d'accéder aux services d'une application. il existe un langage de programmation comme intermédiaire.
- PHP (HyperText Preprocessor) : Est un langage de programmation orienté objet, utilisé généralement pour créer des pages web (ou site) dynamiques en gérant une base de données.
- Servlets : Sont des applications java d'ailleurs leurs noms viennent d'une analogie possible à faire avec les applets. Elles permettent de gérer des requêtes http et de fournir aux clients une réponse http dynamique.
- Java : Le langage choisi pour le développement de l'application et le langage JAVA qui repend au critère de portabilité maximale. En effet, ce langage, développé par « Sun Microsystems Inc. », est disponible pour les principales plates formes du marché, qu'il s'agit de l'Unix, Windows, ou autres et est totalement gratuit. Java possède de nombreuses caractéristiques (orienté objet, fiable, multithread, rapide,

III. J2EE (Java 2 Enterprise Edition) :

III.1.Historique du JEE: [4]

L'environnement Java pour l'entreprise a commencé à émerger assez rapidement après les débuts de Java au milieu des années 90. A son origine, Java était destiné aux environnements contraints (par exemple des petits équipements électroniques). Il a en fait percé dans l'environnement du Web, notamment dans les navigateurs pour le support d'interfaces graphiques riches (notion d'appliquette Java, en anglais *applet*). Les premières déclinaisons de Java dans l'environnement des serveurs sont apparues en 1997 avec les *servlets*, dont l'objectif est la construction programmatique de pages Web, puis avec les *Enterprise Java Beans* dont l'objectif est le support de code métier nécessitant un contexte d'exécution transactionnel.

Après ces premiers pas et un relatif succès de ces technologies, Sun a structuré l'offre technique autour des serveurs d'application Java à travers le standard J2EE. L'objectif de ce dernier est de fédérer dans un cadre cohérent toutes les technologies nécessaires à la mise en oeuvre des applications de l'entreprise (applications orientées «serveur»). La première version des spécification de J2EE est publiée en 1999. Suivront alors la version 1.3 de J2EE en 2001, puis la version 1.4 en 2003 incluant un support complet des standards XML et le support des services Web.

Java est désormais bien installé dans l'écosystème des applications d'entreprise et est devenu le principal concurrent de l'environnement *.NET* de Microsoft dont le chapitre 6 donne un aperçu. Le langage lui-même a acquis de la pérennité puisqu'on compte désormais plus de 4 millions de développeurs Java dans le monde.

III.2.Définition du JEE : [5]

Java 2 entreprise Edition (J2EE) est une spécification pour le langage de programmation Java de Sun destinée aux applications d'entreprise. J2EE offre une plate-forme de développement et déploiement en langage Java pour les applications distribuées à plusieurs niveaux.

On parle généralement de « plate-forme J2EE » pour désigner l'ensemble constitué des services API (Application Programmable Interface) offerts et de l'infrastructure d'exécution.

Dans la mesure où J2EE s'appuie entièrement sur Java, il bénéficie des avantages de ce langage, en particulier une bonne portabilité et une maintenance du code. De plus, l'architecture J2EE repose sur des composants distincts, interchangeables et distribués, ce qui signifie notamment qu'un système reposant sur J2EE peut posséder des mécanismes de haute-disponibilité, afin de garantir une bonne qualité de service et que la maintenance des applications est facilitée.

III.3.Fonctionnement Interne :

Le langage Java, sur lequel les bibliothèques J2EE sont utilisées, met à disposition un compilateur et une machine virtuelle (JVM-Java Virtual Machine) qui se charge de créer un environnement standard pour le lancement de l'application sur tout type de système opérationnel. Le compilateur compile le code source et produit le bytecode, soit un code intermédiaire qui sera en suite exécuté par la machine virtuelle Java. Chaque système opérationnel majeur possède une JVM expressément codée.

III.4.L'architecture du JEE :

J2EE ajoute de nombreuses couches de niveau entreprise au-dessus de la plate-forme J2SE-Java Standard Edition. Chaque couche est conçue pour supporter une différente technologie de développement.

- **Technologie Web application :** Technologies liées à la production des interfaces web dynamique, par exemple JSP (Java server Pages) et Servlet.
- **Technologie entreprise application :** technologies plus directement liées à la logique de business : EJB (Entreprise Java Bean), JavaMail, JMS(Java Message Service), JTA (Java Transaction), etc.
- **Technologie Web Services :** technologies utiles au développement des applications adhérentes au paradigme SOA (Service Oriented Architecture) : web services, JAX-WS (java API for XML-based web services), JAX-RPC (java Api for XML-Based Rpc)
- **Technologie management and Security :** technologie liées à la gestion de la technologie entreprise afin de réaliser l'accès et l'échange d'information entre machines et services distribués : JAAS (Java Authentification and Autorisation Service), JCA (Java Connector Architecture)

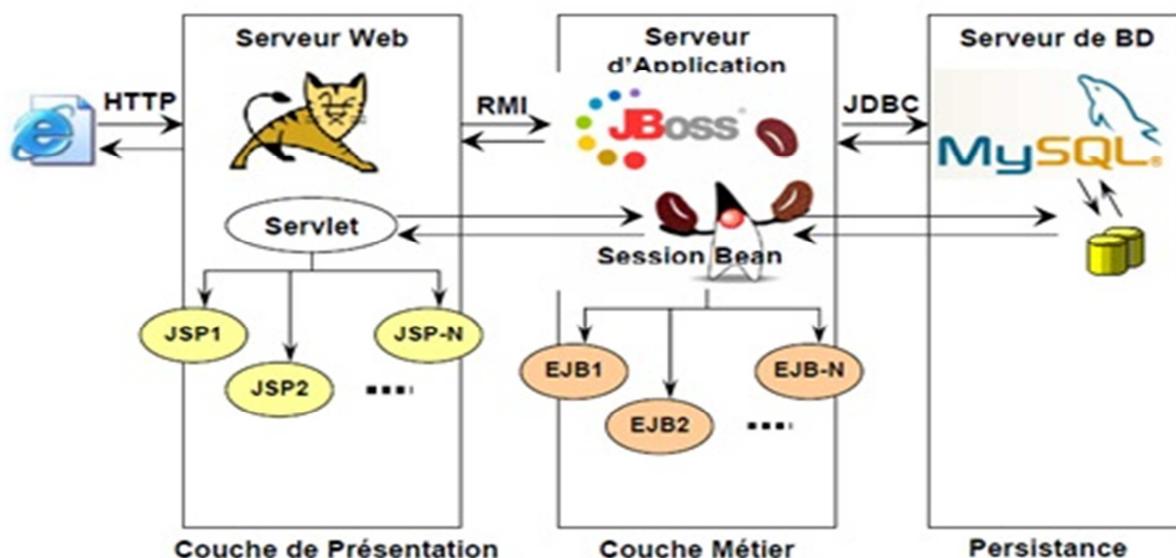


Figure I.1: l'architecture de J2EE.

Nous allons maintenant définir les couches de cette architecture :

- **Couche présentation** : Contient les vues que l'utilisateur va interroger soit pour afficher des données ou envoyer des données.
- **Couche Métier** : La couche la plus lourde normalement, car elle contient les classes de traitement de données (la logique de l'application).
- **Couche Persistance (ou accès aux données)** : La couche basse, c'est elle qui va stocker nos données (texte, XML, base de données, Excel).

Pour expliquer l'utilisation de ces technologies on peut imaginer que les technologies entreprise sont utilisées pour gérer l'accès aux données (généralement une ou plusieurs data base), les technologies web application sont utilisées pour montrer les données aux utilisateurs génériques. Dans un contexte Business to Business, les technologies **Web service** seront utilisées pour échanger les informations avec les partenaires commerciales et les technologies de gestion gèrent tous les processus informationnels assurant la sécurité des transactions.

III.5 .La plateforme J2EE :

En fait, J2EE est une norme qui va spécifier à la fois l'infrastructure de gestion de vos applications et les API des services utilisées pour concevoir ces applications. La plateforme J2EE est essentiellement un environnement fournissant :

- Une infrastructure d'exécution pour faire tourner vos applications.
- Un ensemble de services accessibles via l'API J2EE pour vous aider à concevoir vos applications.

III.5.1.Les API de J2EE : [6]

Les API de J2EE peuvent se répartir en catégories :

- A) **Les composants** : On distingue habituellement deux familles de composants :
- Les composants web : [Servlets](#) et [JSP](#) (Java Server Pages). Il s'agit de la partie chargée

de l'interface avec l'utilisateur (on parle de *logique de présentation*).

- Les composants métier : EJB (Enterprise Java Beans). Il s'agit de composants spécifiques chargés des traitements des données propres à un secteur d'activité (on parle de *logique métier* ou de *logique applicative*) et de l'interfaçage avec les bases de données.
- B) **Les services**, pouvant être classés par catégories :
- Les **services d'infrastructures** : il en existe un grand nombre, définis ci-dessous :
 - **JDBC** (*Java DataBase Connectivity*) est une API d'accès aux bases de données relationnelles.

- **JNDI** (*Java Naming and Directory Interface*) est une API d'accès aux services de nommage et aux annuaires d'entreprises tels que DNS, NIS, LDAP, etc.
- **JTA/JTS** (*Java Transaction API/Java Transaction Services*) est un API définissant des interfaces standard avec un gestionnaire de transactions.
- **JCA** (*J2EE Connector Architecture*) est une API de connexion au système d'information de l'entreprise, notamment aux systèmes dits «Legacy» tels que les [ERP](#).
- **JMX** (*Java Management Extension*) fournit des extensions permettant de développer des applications web de supervision d'applications.
- Les **services de communication** :
 - **JAAS** (*Java Authentication and Authorization Service*) est une API de gestion de l'authentification et des droits d'accès.
 - **JavaMail** est une API permettant l'envoi de courrier électronique.
 - **JMS** (*Java Message Service*) fournit des fonctionnalités de communication asynchrone

III.6. L'architecture MVC: [7]

Un modèle de conception, en anglais design pattern (ou encore patron de conception) est une simple bonne pratique, qui répond à un problème de conception d'une application.

C'est en quelque sorte une ligne de conduite qui permet de décrire les grandes lignes d'une solution. De tels modèles sont issus de l'expérience des concepteurs et développeurs d'applications : c'est en effet uniquement après une certaine période d'utilisation que peuvent être mises en évidence des pratiques plus efficaces que d'autres, qui sont alors structurées en modèles et considérées comme standard.

Java EE permet plus ou moins de coder son application comme on le souhaite, en d'autres termes, on peut coder n'importe comment, et dans Java EE, il y'a « Entreprise », et le développement en entreprise implique entre autres :

- Travailler à plusieurs contributeurs sur un même projet ou une même application (travail en équipe).
- Evoluer une application que l'on n'a pas créée soi-même.
- Maintenir et corriger une application que l'on n'a pas créée soi-même.

Pour toutes ces raisons, il est nécessaire d'adopter une architecture plus ou moins standard, que tout développeur peut reconnaître.

Il a été très vite remarqué qu'un modèle permettait de répondre à ces besoins, et qu'il appliquait particulièrement bien à la conception d'application Java EE : le modèle MVC

Modèle – Vue – Contrôleur). Il découpe littéralement l'application en couches distinctes, et de ce fait impacte très fortement l'organisation du code. Voici ce qu'impose MVC :

- Tout ce qui concerne le traitement, le stockage et la mise à jour des données de l'application doit être contenu dans la couche nommée « Modèle » (le M de MVC).
- Tout ce qui concerne l'interaction avec l'utilisateur et la présentation des données (mise en forme, affichage) doit être contenu dans la couche nommée « Vue » (le V de MVC).
- Tout ce qui concerne le contrôle des actions de l'utilisateur et des données doit être contenu dans couche nommée « Contrôle » (le C de MVC).

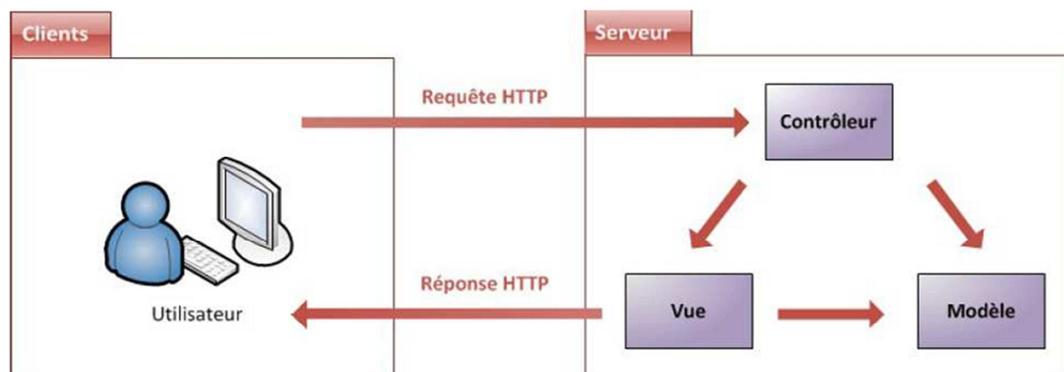


Figure I.2: Architecture M.V.C.

III.6.1. Modèle : des traitements et des données :

Dans le modèle, on trouve à la fois les données et les traitements à appliquer à ces données. Ce bloc contient donc des objets Java d'une part, qui peuvent contenir des attributs (données) et des méthodes (traitements) qui leur sont propres, et un système capable de stocker des données d'autre part. La complexité du code dépendra bien évidemment de la complexité des traitements à effectuer par l'application.

III.6.2.Vue : des pages JSP (Java Server Page):

Une page JSP est destinée à la vue. Elle est exécutée côté serveur et permet la génération de pages web dynamiques. Cette technologie permet de séparer la présentation sous forme de code HTML (CSS, JavaScript, XML, etc.) et les traitements sous formes de classes Java définissant un Bean ou une servlet. Ceci est d'autant plus facile que les JSP définissent une syntaxe particulière permettant d'appeler un Bean et d'insérer un traitement dans la page HTML dynamiquement.

III.6.3. Contrôleur : des servlets

Un servlet est un objet qui permet d'intercepter les requêtes faites par un client, et qui peut personnaliser une réponse en conséquence. Il fournit pour cela des méthodes permettant de scruter une requête http. Cet objet n'agit jamais directement sur les données, il faut le voir comme un simple aiguilleur il intercepte une requête issue d'un client, appelle éventuellement des traitements effectués par le modèle, et ordonne en retour à la vue d'afficher le résultat au client.

III.7. L'environnement d'exécution des applications J2EE : [8]

J2EE propose des spécifications pour une infrastructure dans laquelle s'exécutent les composants. Ces spécifications décrivent les rôles de chaque élément et précisent un ensemble d'interfaces pour permettre à chacun de ces éléments de communiquer.

Ceci permet de séparer les applications et l'environnement dans lequel elles s'exécutent. Les spécifications précisent à l'aide des API un certain nombre de fonctionnalités que doit implémenter l'environnement d'exécution. Ces fonctionnalités sont de bas niveau ce qui permet aux développeurs de se concentrer sur la logique métier.

Pour exécuter ces composants de natures différentes, J2EE définit des conteneurs pour chacun d'eux. Il définit pour chaque composant des interfaces qui leur permettront de dialoguer avec les composants lors de leur exécution. Les conteneurs permettent aux applications d'accéder aux ressources et aux services en utilisant les API.

Les appels aux composants se font par des clients en passant par les conteneurs. Les clients n'accèdent pas directement aux composants mais sollicitent le conteneur pour les utiliser.

III.8. Avantages du JEE :

L'utilisation de J2EE pour développer et exécuter une application représente plusieurs avantages :

- ✓ Une architecture d'application basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement.
- ✓ La possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA...
- ✓ La possibilité de choisir les outils de développement et les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.
- ✓ La plateforme J2EE est qu'elle est très vaste,
- ✓ Propose de très nombreux outils qui adressent toutes les problématiques de l'informatique d'entreprise : exposer des fonctionnalités avancées sur un site

web, accès à des bases de données, communications distantes entre applications Java (EJB ou web services) ou non Java (web services) etc...

IV. Conclusion :

Dans ce chapitre nous avons présenté une vue globale de la technologie WEB et J2EE, ce qui nous aidera dans le développement de notre application. Le chapitre suivant sera consacré à la présentation de l'organisme d'accueil.

I. Introduction :

La présentation de l'organisme d'accueil est une étape importante de l'analyse qui nous permet de prendre connaissance du domaine dans lequel l'organisme souhaite améliorer son fonctionnement. Nous allons commencer par la présentation d'une vue globale sur l'organisme qui nous a accueilli (l'Entreprise Nationale des Industries de l'Electroménager), ainsi la présentation de notre champ d'étude.

II. Présentation de l'organisme d'accueil ENIEM : [9]

1. Historique de L'ENIEM :

L'ENIEM est une entreprise de droit Algérien consiste le 2 Janvier 1983, mais qui existe depuis 1974 sous tutelle de l'entreprise. Son social se situe à Tizi-Ouzou.

Elle est issus de la restriction de l'ex SONELEC (Société Nationale de Fabrication et de Montage Matériel Electrique) suite au décret présidentiel N° 8 3 janvier 1983.

L'ENIEM est passé d'une entreprise publique économique à une société par actions en octobre 1989, avec un capital social de 40.000.000 DA.

Aujourd'hui, son capital social est passé à 2.947.500.000 DA détenu en totalité par le HOLDING HEELIT (Holding Electronique, Informatique et télécommunication).



2. Situation géographique :

L'entreprise ENIEM (Entreprise Nationale des Industries de l'Electroménager) se trouve au sein de la zone industrielle AISSAT-IDIR OUED-AISSI à 10Km de TIZI-OUZOU,

elle s'étale sur une surface totale de 55 Hectares, sa direction générale se trouve au chef lieu de TIZI-OUZOU à proximité de la gare ferroviaire.

3. Missions et objectifs de L'ENIEM :

3.1. Mission :

L'ENIEM est spécialisée dans la fabrication et l'assemblage, elle assure également la commercialisation et le service après vente de ses appareils.

Dans le cadre du développement économique et social, l'ENIEM assure les fonctions suivantes :

La production, le montage, l'assemblage, la commercialisation, comme elle assure aussi le service après vente de ses appareils et la recherche dans les différentes branches de l'électroménager notamment :

- ✓ Les équipements ménagers domestiques.
- ✓ Les équipements industriels.
- ✓ Le petit appareil ménager.

Elle assure également la production :

- ✓ Des appareils réfrigérateurs et congélateurs des différentes capacités (160L à 520L).
- ✓ Des cuisiniers à gaz 4 et 5 feux, dont la production atteint 150 000 appareils par ans.
- ✓ Des climatiseurs types fenêtres et Split système (1CV à 2,5 CV).

Sa finalité est la satisfaction de la demande nationale et l'exportation de sa production.

3.2. Objet :

ENIEM est leader de l'électroménager en Algérie. Elle possède des capacités de production et une expérience de 30 ans dans la fabrication, le développement et la recherche dans le domaine de différentes branches de l'électroménager. Elle assure également la production des appareils de réfrigération, de cuisson, de climatisation et de lavage avec une intégration nationale importante (70% environ).

L'ENIEM s'engage pour la protection de l'environnement selon la norme ISO 14001 et compte relever un autre défi qui consiste en l'amélioration du design de sa gamme de produits et l'introduction de nouveaux produits.

Ces élans doivent être mis à profit pour lui permettre d'asseoir d'avantage sa suprématie sur le marché national et lui ouvrir la voix pour la conquête de nouveaux marchés à l'international.

- **Parmi le principal objectif de L'ENIEM :**
 - ✚ L'amélioration de la qualité des produits.
 - ✚ L'augmentation des capacités d'études et de développement.
 - ✚ La maîtrise des coûts de production.
 - ✚
 - ✚ L'orientation de volume de production en corrélation avec les variations de la demande.
 - ✚ Place de son produits a l'échelle internationale (exportation).
 - ✚ Epargner au pays la sortie des devises (importation).

4. Organisation général :

4.1. Les directions :

4.1.1. Direction générale :

La direction générale est l'unique entité qui est responsable de la stratégie et du Développement de l'entreprise. Elle exerce son autorité hiérarchique et fonctionnelle sur l'ensemble des directions et unités.

4.1.2 Direction planification et contrôle de gestion :

La direction assure le contrôle de gestion. De l'audit finance ainsi que le budget de L'entreprise. Elle assure également les tâches suivantes :

- La réalisation et la présentation de tous les travaux permettant de produire une information complète et cohérente des activités de l'entreprise.
- Préparation, établissement et le suivi du budget de l'entreprise.
- Planification et organisation du programme annuel d'audit finance ainsi sa réalisation.
- Le contrôle de rapport d'activités.
- L'exploitation et l'interprétation des résultats des audits finances, et faire les
- recommandations nécessaires.

4.1.3. Direction développement et partenariat :

Cette direction assure l'étude et le développement du produit fini ainsi que des actions de partenariat et de sous-traitance.

- Elle définit et supervise les actions de développement des produits existants.

- Suit avec la direction industrielle les actions de développement des processus de fabrication et de modernisation de l'outil de production.
- Participe à la définition de l'organisation de la production dans l'objectif de flexibilité, de réduction des coûts de fabrication.

4.1.4. Direction des finances et comptabilité :

Ses missions principales sont :

- Assure la comptabilisation dans les délais de toutes les opérations permettant l'établissement du bilan, du compte de résultat et des déclarations fiscales de l'entreprise.
- Analyse les équilibres financiers de l'entreprise.
- Définit la politique bancaire et l'orientation budgétaire.
- Gère la trésorerie (recette et dépenses) et contrôle les rédactions fiscales périodiques.
- Rédige, vérifie et approuve les dispositions décrites relatives au fonctionnement efficace de son activité.

4.1.5. Direction du marketing et la communication :

Met en œuvre la stratégie concurrentielle de l'entreprise. Chargée de la réparation et l'organisation des foires et salons nationaux.

4.1.6. Direction industrielle :

La direction industrielle est chargée de développer et de mettre en place les moyens et l'organisation industrielle nécessaire à la réalisation de la production en agissant sur les approvisionnements, les moyens et les techniques de production.

- Définit les programmes de production en fonction de la demande commerciale et des capacités installés avec le souci de la rentabilité.
- Améliore la gestion de production en relation avec la structure informatique(GPAO).
- Organise et anime l'industrialisation de produits nouveaux.
- Définit une politique d'amélioration de la maintenance des équipements de production et en assurer le suivi.

4.1.7. Direction des ressources humaines :

En cohérence avec la politique de l'entreprise, la fonction ressources humaines accroit la mobilisation et la valorisation du personnel dans ses actions aux services du client.

- Elle pilote le recrutement, l'accueil, l'information et gère le plan de carrière du personnel et les pouvoirs publics.

- Définit et exécute les plans de formation des besoins de l'entreprise et suivant les niveaux de qualification du personnel.
- Gère les éventuels conflits collectifs dans le respect de la loi et préparer les décisions de direction.
- Organiser et promouvoir la médecine de travail.

4.2. Les unités :

Plus des directions, L'ENIEM est organisée sous forme d'unités.

4.2.1. Unité froid : spécialisée dans la production et le développement des produits de froid domestique tel que : réfrigérateurs, congélateurs et armoires frigorifique.
Le processus de fabrication est assuré par huit (08) ateliers qui sont :

- ✓ Atelier d'injection plastique ;
- ✓ Atelier presses et soudures ;
- ✓ Atelier de refondée et de mise en longueur ;
- ✓ Atelier de traitement et de revêtement des surfaces ;
- ✓ Atelier de fabrication de pièces métalliques ;
- ✓ Atelier de thermoformage ;
- ✓ Atelier de montage final.

Et un laboratoire central composé de trois (03) sections :

- ✓ Laboratoire de chimie ;
- ✓ Laboratoire de métallurgie ;
- ✓ Laboratoire d'essais produits.

4.2.2. Unité cuisson : Cette unité est spécialisée dans la production des produits du cuisson à gaz, électrique **ou** mixte et tout produit de technologie similaire.

Ces activités sont :

- Transformation de la tôle.
- Traitement et revêtement de surface (émaillage, zingage, chromage).
- Assemblage de la cuisinières.

4.2.3. Unité climatisation : Elle est spécialisée dans la fabrication des produits de climatisation et chauffage.

Ces activités sont :

- Transformation de la tôle.
- Traitement et revêtement de surface (peinture).
- Assemblage de climatiseurs.

4.2.4. Unité prestation techniques (UPT) : Cette unité est chargée de fournir et d'exploiter les moyens techniques communs ainsi que de la gestion de la totalité des infrastructures communes (bâtiments, éclairages etc...). Elle assure également la réalisation de pièces de rechanges pour la production, la conception et la fabrication d'outillage (moules, outils etc...), assure toutes les activités informatiques des unités.

4.2.5. Unité commerciale : Elle est chargée de la commercialisation de produit de l'entreprise, de la gestion du réseau et de service après vente.

Ces activités sont :

- Marketing.
- La vente (à travers ses moyens propres et un réseau d'agent agréé) .
- Services après vente.
- Gestion des stocks des produits finis des produits ENIEM.
- La distribution et l'exportation des produits ENIEM.

5. L'organigramme de l'ENIEM :

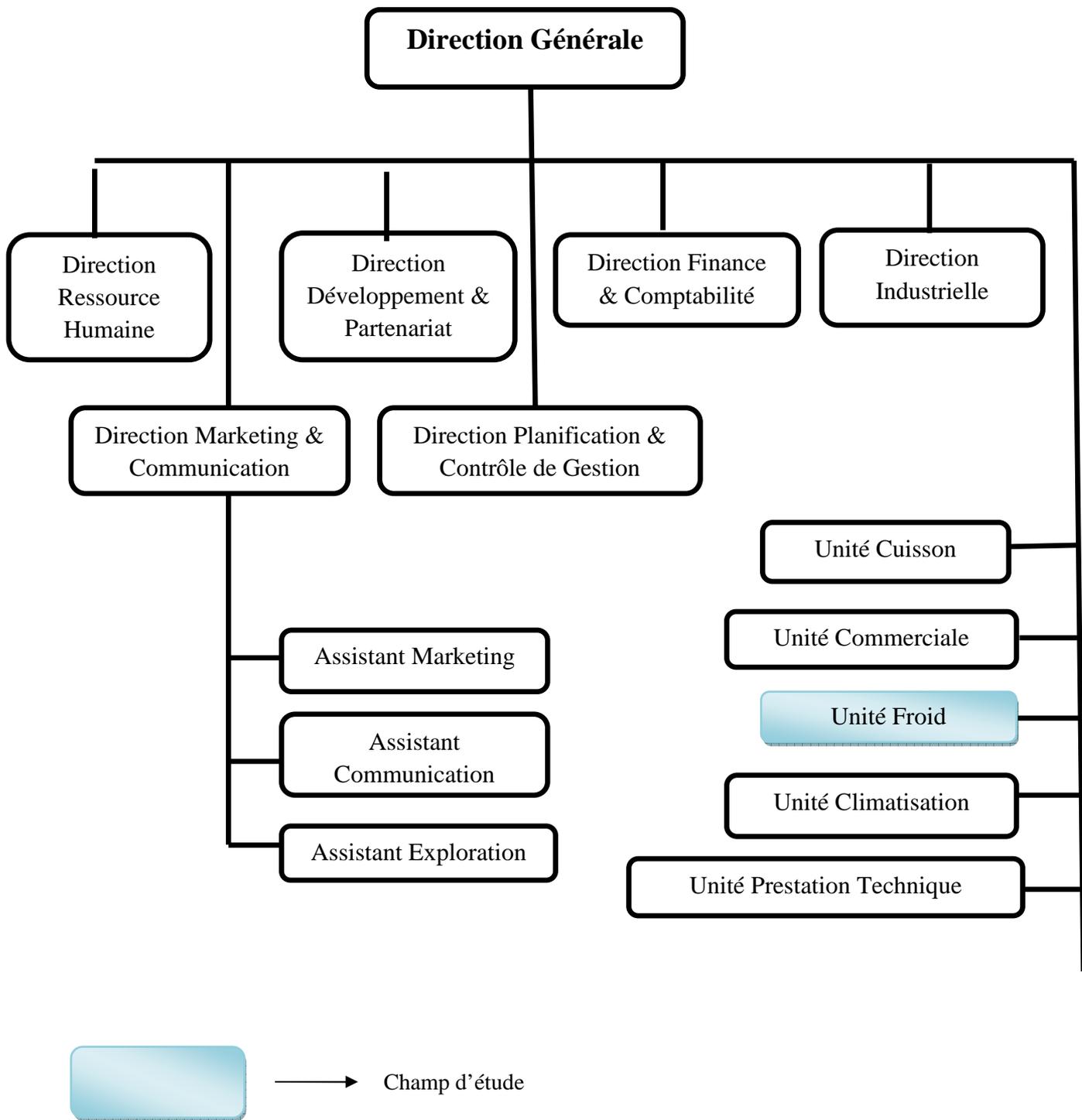


Figure II.3 : Organigramme général de l'ENIEM.

6. L'organigramme général de l'unité froid :

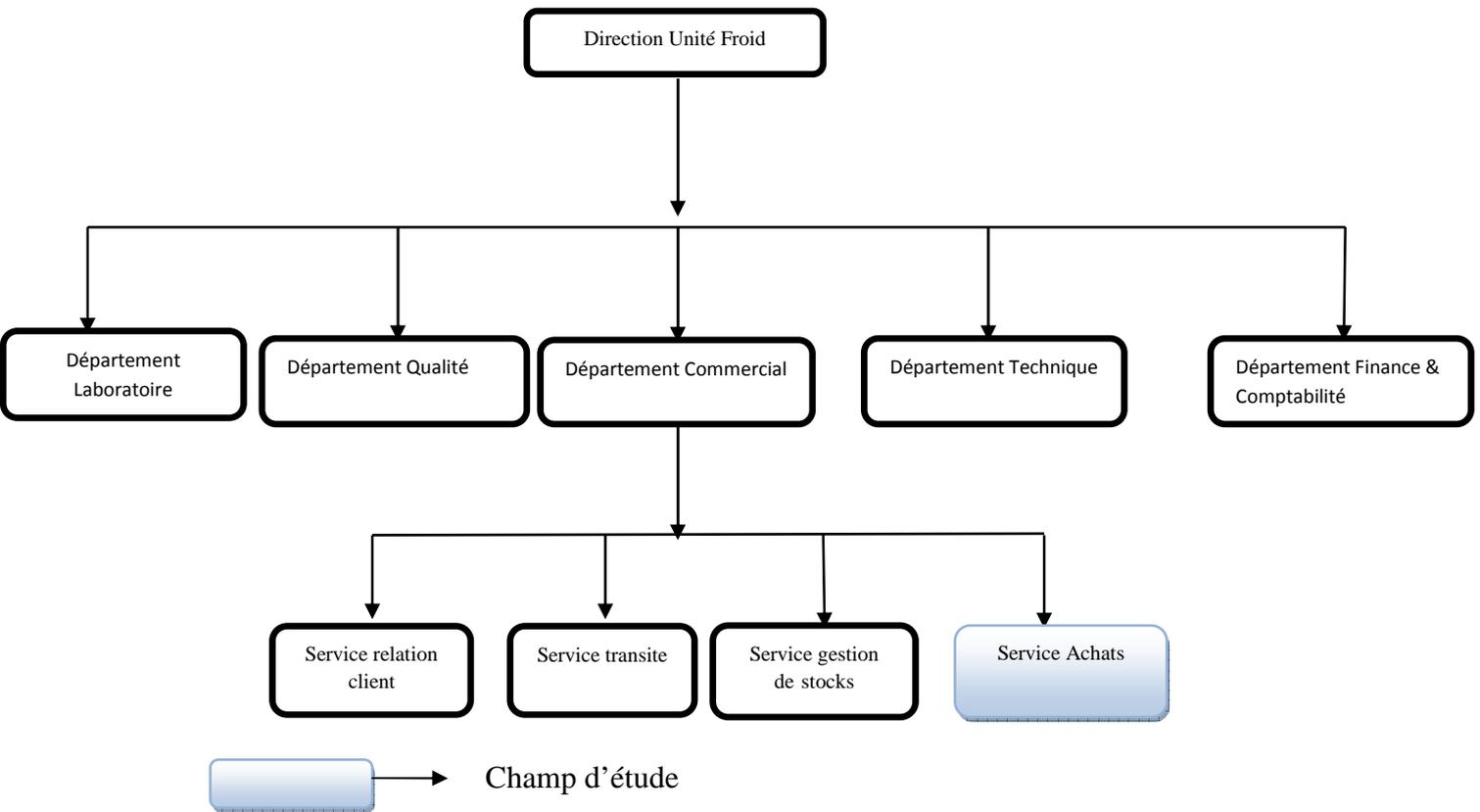


Figure II.4: Organigramme du l'Unité de Froid.

7. L'organigramme de service d'achat :

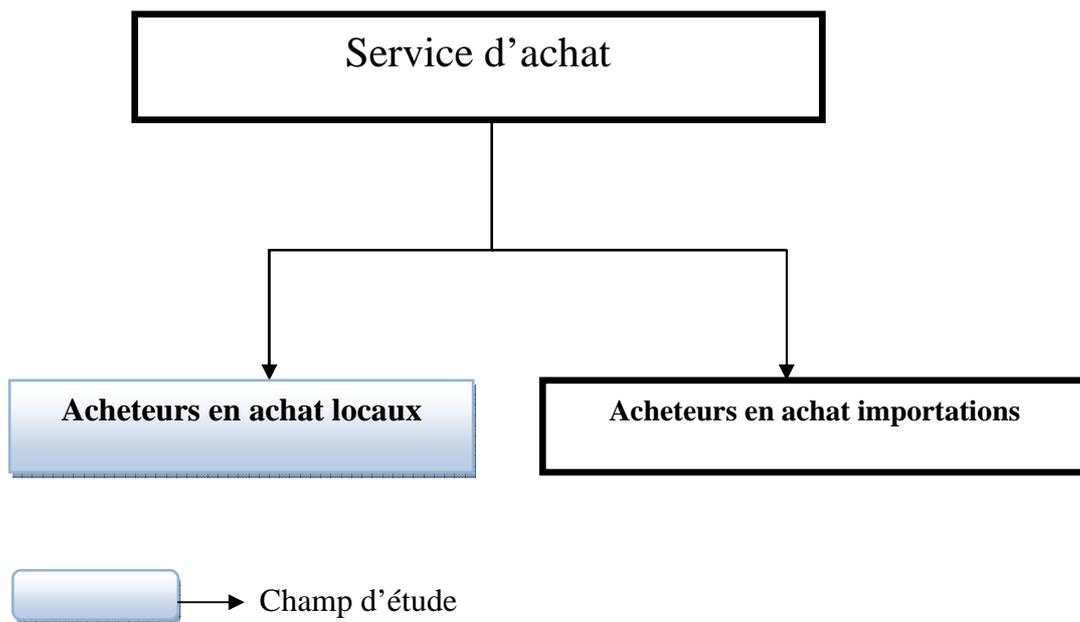


Figure II.5 : Organigramme de services d'achat.

III. Etude de l'existence :

III.1. Quelques définitions :

- ❖ **Demande d'achat interne (DAI) :** C'est une demande d'achat informatisée, établie par le service gestion des stocks (magasin) à la suite d'une naissance d'un besoin. Elle est transmise au service achats après avoir été approuvée et signée par :
 - ✓ **Chef de section magasin**
 - ✓ **Chef de service magasin**
 - ✓ **Chef de département commerciale**
 - ✓ **Directeur de l'unité.**

- ❖ **Demande d'offre (DO) :** Document contenant la liste des produits que l'entreprise souhaite acheter (d'après la DAI) cette demande est envoyée aux fournisseurs.
- ❖ **Les Facture:** Représente les offres des fournisseurs suite à la réception d'une demande d'offre.
- ❖ **Le tableau comparatif des offres (TCO) :** Document établi par les acheteurs après réception des différentes offres des fournisseurs dans le but de sélectionner le meilleur fournisseur après comparaison des offres.
- ❖ **Une commande :** Exprimée par un bon de commande (BC) contenant la liste des produits, les quantités et caractéristique voulues.
- ❖ **Fiche de vérification :** C'est un document contenant les informations du fournisseur retenu.
- ❖ **Bon de réception (BR) :** Document établi par le service magasin contenant la liste des produits reçus.

III.2. Etudes des postes de travail :

Cette étude a pour objectif de recenser les différents postes de travail entrant dans la gestion des achats locaux de l'ENIEM (Unité froid), cela tout en précisant les tâches assurées par chaque poste et les différents documents manipulés (informations, entrées, sorties, ...etc.).

Le service des achats appartenant à L'unité Froid représente le maillon principal dans la chaîne des achats locaux, ce qui mène à nous concentrer sur ce service lors de l'étude de l'existant.

Fiche d'analyse poste N°01
Code:/ Désignation : Acheteurs Rattachement : Service achats Effectif : 02 Responsabilité de poste : -Assure l'achat de matières premières et auxiliaires, composants, pièces de rechange et outillage. -Gestion des dossiers d'achats.
Tâches et responsabilités -Effectuée la prospection du marché. -Lance et réalise des appels d'offres. - Effectuée des études comparatives des différentes offres. -Négocie avec des fournisseurs. -Veille au respect de la réglementation et des procédures en vigueur. -Suit la réalisation et la finalisation des opérations déclenchées. -Assure l'évaluation des fournisseurs . -Constitue les listes des fournisseurs (agrée, qualifié, déqualifié et déclassé) . -Explicite les fiches suiveuses qualité. -Constitué et met a jour les dossiers fournisseurs. -S'assure de la conformité aux procédures du système de management environnemental lors de l'achat des matières premières et composants. -Est chargé de l'acquisition des fiches des données de sécurité. -S'assure de la qualité de l'emballage et de l'étiquetage des contenants des produits chimique avant leurs expédition . -Applique la réglementation relative a l'importation et au transport des matières dangereuses.

III.3. Etudes des documents :

III.3.1 Liste des documents :

Il est nécessaire de connaître les différents documents utilisés dans le système existant, ainsi que toutes les informations nécessaires à l'élaboration du dictionnaire de données qui sert à la conception de la base de données.

Dans le cadre de notre étude, nous avons recensé une liste de onze (6) documents.

N° du document	Code du document	Désignation du document
01	DAI	Demande d'achat interne
02	DO	Demande d'offre
03	/	Facture
04	TCO	Tableau comparatif des offres
05	BC	Bon de commande
06	BR	Bon de réception

III.3.2 Fiches d'analyses des documents :

Fiche d'analyse du document N°01			
Identification			
Code : DAI Désignation : Demande d'achat interne Nature : Interne Source : Service de gestion des stocks Destinataire : services achats Rôle : Exprimer un besoin d'achat Nombre d'exemplaire : 02			
Désignation			
Rubrique	type	Taille	
Observation			
Nom de l'unité émettrice	AN	30	Exp: EDAI
Nom de l'état	A	04	
Nom de demandeur	AN	25	
Date d'établissement de la DAI	D	10	JJ/MM/AAAA
Heure d'édition	AN	05	Exp:10H30
Service émetteur	A	20	
N° de la DAI	N	08	
Date d'émission	D	10	JJ/MM/AAAA
Statut achats	A	01	Exp: F(Fermé)
Code suivi entête DAI	A	01	
Objet de la DAI	A	30	
Numéro séquentiel	N	04	
Désignation article	AN	40	
Quantité demandée	N	10	

Fiche d'analyse du document N°02			
Identification			
Code : DO Désignation : Demande d'offre Nature : Externe Source :Service des achats Destinataire : Fournisseur Rôle : Consultation des fournisseurs pour une demande d'achat Nombre d'exemplaire : 01			
Désignation			
Rubrique	Type	Taille	Observation
Référence client	A	09	Exp: ENIEM UPT
Nom de département	A	10	
Structure émettrice	A	15	
Date d'établissement du document	D	10	JJ/MM/AAAA
Adresse de client	AN	40	
N° de téléphone de l'unité émettrice	N	12	
Nom de fournisseur	AN	30	
N° de fax de fournisseur	N	12	
Objet de document	A	15	Consultation ou Négociation
Désignation de l'article	AN	150	
Unité de mesure de l'article	A	15	
Quantité de l'article	N	10	
Délai de livraison	AN	08	Exp : 10 jours
Nom de chef de services achat	AN	30	

Fiche d'analyse du document N°03			
Identification			
Code : /			
Désignation : Facture			
Nature : Externe			
Source : Fournisseur			
Destinataire : Service achats			
Rôle : Désigner les différents articles, leurs quantités, qualités, et prix proposés.			
Nombre d'exemplaire : 01			
Désignation			
Rubrique	Type	Taille	Observation
Nom de fournisseur	A	40	
Date d'établissement de la facture	D	10	JJ/MM/AAAA
Adresse du fournisseur	AN	30	
N° de téléphone du fournisseur	N	12	
N° de fax du fournisseur	N	12	
N° de registre commercial	AN	10	Exp : 07F9603915
Matricule fiscal du fournisseur	N	15	
Référence client	A	09	
Adresse client	AN	30	
N° de téléphone client	N	12	
N° de fax client	N	12	
N° de la facture pro forma	AN	09	
N° d'ordre article	N	04	
Désignation article	AN	40	
Unité de mesure article	A	12	
Quantité de l'article	N	10	EXP :Kilo, Litre, M²
Prix unitaire de l'article	N	10	
Montant HT de l'article	N	10	
Montant THT de l'article	N	10	
Montant TTC.	N	10	
TVA 17%			

Fiche d'analyse du document N°04			
Identification			
Code : TCO			
Désignation : Tableau comparatif des offres			
Nature : Interne			
Source : Acheteur			
Destinataire : Chef de Service achats			
Rôle : Comparer les factures pro forma puis proposer le fournisseur retenue avec justification			
Nombre d'exemplaire : 01			
Désignation			
Rubrique	Type	Taille	Observation
Code de l'unité émettrice	A	03	
Structure émettrice	A	15	
Centre de frais de l'unité	N	03	Exp:561
N° de document	AN	09	Exp:06/763
Date d'établissement	D	10	JJ/MM/AAAA
Nom de l'acheteur	AN	30	
Fonction de l'acheteur	A	30	
Code article	AN	08	Exp:672204XX
Désignation de l'article	AN	40	
Quantité de l'article	N	10	
Type équipe	A	02	(NL)Normal
Unité de mesure	A	10	
Montant proposé en dinars (01)	N	10	
Montant négocié en dinars (01)	N	10	
Montant proposé en dinars (02)	N	10	
Montant négocié en dinars (02)	N	10	
Montant proposé en dinars (03)	N	10	
Montant négocié en dinars (03)	N	10	
Montant globale	N	10	
Modalité de paiement	A	07	Exp: espèces, chèque
Délais de livraison	AN	10	Exp : 15 jours
Nom de fournisseur proposé	AN	30	
Critère de choix	AN	80	
Montant dinars	N	10	
Nom de fournisseur non retenu et motifs du rejet	AN	80	
Nom de chef de service	AN	30	
Date de vérification du document	D	10	JJ/MM/AAAA

Fiche d'analyse du document N°05			
Identification			
Code : BC			
Désignation : Bon de commande			
Nature : Interne			
Source : Acheteur			
Destinataire : Service achats			
Rôle : Confirmation de l'offre négocie avec le fournisseur			
Nombre d'exemplaire : 04			
Désignation			
Rubrique	Type	Taille	Observation
Date d'établissement du document	D AN	10 40	JJ/MM/AAAA
Nom de l'unité émettrice	AN	60	
Adresse postale de l'unité	N	12	
N° de téléphone de l'unité	N	12	
N° de fax de l'unité	AN	10	Exp : UCL 03/2014
N° de bon de commande	AN	40	
Nom du fournisseur retenu	AN	80	
Adresse de fournisseur retenue	AN	06	Exp :N67525
Désignation de l'article	A	08	
Unité de mesure	N	10	
Quantité de l'article	N	10	
Prix unitaire	N	10	
Montant total d'un article	A	08	
Mode de paiement	/	/	Exp : espèce, chèque
Type d'emballage	AN	08	Non remplie
Délai de livraison	/	/	Exp : 3 jours
Marquage	AN	09	
Montant HT	N	10	
Montant TVA	AN	03	
Taux de la TVA	N	10	
Montant TTC	AN	40	17%
Nom du demandeur	AN	40	
Fonction du demandeur			

Fiche d'analyse du document N°6			
Identification			
Code : BR			
Désignation : Bon de réception			
Nature : Externe			
Source : Service de gestion de stock			
Destinataire : acheteur			
Rôle : Confirme la réception de la marchandise			
Nombre d'exemplaire : 03			
Désignation			
Rubrique	Type	Taille	Observation
Date d'établissement du document	D	10	JJ/MM/AAAA
N° de la commande	AN	07	Exp : 09 /2764
Date de la commande	D	10	JJ/MM/AAAA
Nom du fournisseur	AN	40	
N° de la facture	AN	07	
Date de livraison	D	10	JJ/MM/AAAA
Mode de livraison	N	01	Case à cocher : air, terre, mer
N° d'ordre d'article	N	04	
Code de l'article	AN	08	
Désignation de l'article	AN	40	
Unité de mesure	A	12	
Quantité article livré	N	10	
Quantité article bloqué	N	10	
Quantité article reçu	N	10	=Qté livrée – Qté bloquée
Date de vérification de la marchandise (qualité)	D	10	JJ/MM/AAAA
Date de vérification de la marchandise (quantité)	D	10	JJ/MM/AAAA

III.4. Organisation du travail

III.4.1 Le diagramme de flux généré

- **Objectifs**

Représenter les flux d'informations consiste à analyser les échanges d'informations au sein du système d'information d'une organisation (entreprise, administration ou association) et avec d'autres systèmes d'information.

Cette étude permet de produire un diagramme de flux. Ce diagramme donne une vue d'ensemble de la circulation de l'information entre les acteurs qui participent à la réalisation de l'activité étudiée.

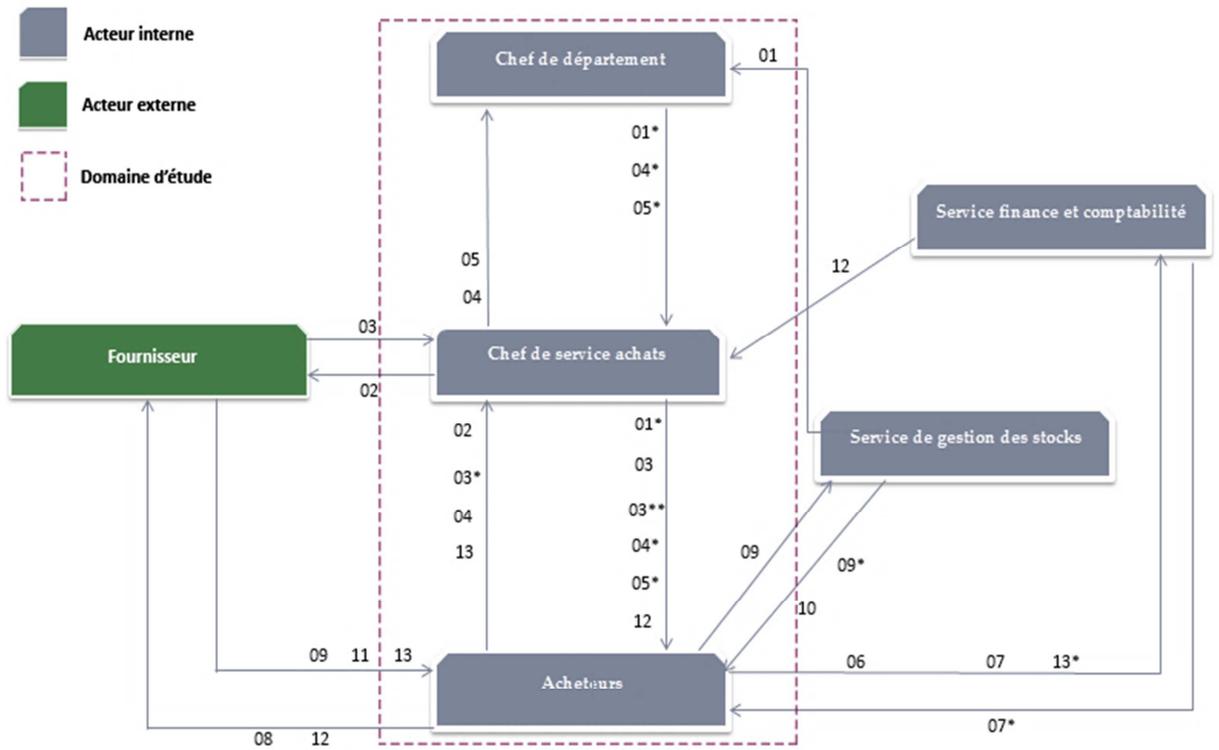
- **Flux**

Un flux désigne un transfert d'information entre deux acteurs du système d'information. Un flux part d'un acteur source pour aboutir à un acteur but, il est représenté par une flèche. On peut identifier différentes catégories de flux (flux physiques, flux financiers par exemple), ces flux doivent être explicités sous la forme de flux d'informations

III.4.1.1 Les flux d'informations :

Les Flux :

N° du flux	Désignation du flux
01	DAI (Demande d'achat interne)
01*	DAI signée
02	Demande d'offres
03	Facture pro forma (Offre)
03*	Facture pro forma (Offre) retenue
03**	Facture pro forma négociée
04	Tableau comparatif des offres (TCO)
04*	TCO contrôlé et signé
05	Parafeur (DAI signée, bon de commande, demande de chèque, facture commerciale)
05*	Parafeur (DAI signée, bon de commande parafé et signé, demande de chèque parafée et signée, facture commerciale parafée et signée)
06	Demande de chèque parafée et signée
07	Bordereau d'envoi (2 copies de bon de commande parafée et signée, 2 copies de factures commerciales parafées et signées)
07*	Bordereau d'envoi signé
08	Bon de commande parafé et signé
09	Bon de livraison
09*	Bon de livraison parafé signé
10	Bon de réception
11	Facture commerciale.
12	Chèque
13	Accusé de réception chèque
13*	Accusé de réception chèque signé



IV. Conclusion :

Dans ce chapitre nous avons décrit une brève présentation de l'organisme d'accueil, afin de comprendre son fonctionnement générale

I. Introduction :

En programmation informatique, un **Framework** est un espace de travail modulaire. C'est un ensemble de bibliothèques, d'outils et de conventions permettant le développement d'applications. Il fournit suffisamment de briques logicielles et impose suffisamment de rigueur pour pouvoir produire une application aboutie et dont la maintenance est aisée. Le principal objectif de cet outil est de proposer une démarche et des ressources pour mieux maîtriser le développement et gagner du temps, et aussi est un ensemble de conventions, de composants et de fonctionnalités utilisables pour développer des applications logicielles structurées, évolutives et maintenables. Ces composants sont organisés pour être utilisés en interaction les uns avec les autres.

Un Framework, qu'il soit java ou autre, rend les développeurs plus rapides, plus stables, plus modulaires et plus évolutifs du fait des conventions à respecter. Il limite les égards de début de développements et réduit ainsi les failles de conception des fondements de son application.

II. Notion d'un Framework : [10]

Au moment d'industrialiser ses développements, toute entreprise cherche le moyen d'avoir la meilleure solution pour gérer son environnement. Cela passe donc bien entendu par le choix d'un IDE, le choix d'un gestionnaire de version, mais surtout par le choix d'un Framework. Celui-ci permet en effet de forcer les développeurs à utiliser des conventions dans leur façon de coder, et donc à n'importe quel développeur de pouvoir prendre en main l'ensemble du projet rapidement par la suite. Il permet aussi de guider les développeurs dans le choix technologique et d'architecture afin de pouvoir ajouter des « briques » au projet facilement par la suite.

Un Framework répond normalement à de nombreuses demandes des développeurs. Il doit par exemple prendre en compte nativement les éléments suivants :

- **Modèle MVC** : modèle vue contrôleur. Il permet de structurer une application en distinguant la partie présentation, la partie base de données et la partie applicative.
- **Template** : gestion des gabarits. Il permet de séparer le code applicatif de la présentation.
- **Cache** : Il permet de stocker les pages afin d'optimiser leur temps de chargement.
- **Gestion des SGBDR** : Il doit pouvoir gérer plusieurs types de base de données. MySQL bien entendu, mais éventuellement Oracle en fonction des besoins.
- **ORM** : Mapping de relation objet. Cela permet de gérer la base de données sous forme d'objets.
- **Conventions** : Oblige les développeurs à utiliser les mêmes conventions de codage afin d'avoir un code uniforme.

III. Définition D'un Framework : [11]

Le mot « Framework » provient de l'anglais « frame » qui veut dire « cadre » en français, et « work » qui signifie « travail ». Littéralement, c'est donc un « cadre de travail ». Vous voilà bien avancés, Concrètement, c'est un ensemble de composants qui sert à créer les fondations, l'architecture et les grandes lignes d'un logiciel. Il existe des centaines de Framework couvrant la plupart des langages de programmation. Ils sont destinés au développement de sites web ou bien à la conception de logiciels.

Un Framework est une boîte à outils conçue par un ou plusieurs développeurs à destination d'autres développeurs. un Framework n'est pas utilisable tel quel. Il n'est pas fait pour être utilisé par les utilisateurs finaux. Le développeur qui se sert d'un framework a encore du boulot à fournir, d'où ce cours !

On trouve différents types de Framework :

1. Framework d'infrastructure système : pour développer des systèmes d'exploitation, des interfaces graphiques, des outils de communication (exemple : Framework .Net, Struts) .
2. Framework d'intégration intergicielle (*middleware*) : pour fédérer des applications hétérogènes. Pour mettre à disposition différentes technologies sous la forme d'une interface unique .
3. Framework d'entreprise : pour développer des applications spécifiques au secteur d'activité de l'entreprise .
4. Framework de gestion de contenu : sont les fondations d'un système de gestion de contenu pour la création, la collecte, le classement, le stockage et la publication de « biens numérisés ».

IV. Objectif d'un Framework :

L'objectif premier d'un Framework est d'améliorer la productivité des développeurs qui l'utilisent. Un Framework offre la possibilité au développeur final d'utiliser tel ou tel composant pour lui faciliter le développement, et lui permet ainsi de se concentrer sur le plus important.

Autrement dit, le Framework s'occupe de la forme et permet au développeur de se concentrer sur le fond. L'avantage premier est donc, le gain en productivité. Mais il en existe bien d'autres ! On peut les classer en plusieurs catégories : le code, le travail et la communauté.

V. Avantages des Framework :

- Le Temps de développement plus court.
- La Sécurité des applications.
- Les Règles de codages stricts.
- Un Investissement pour le futur.
- La portabilité.

VI. Les Framework java :

VI.1. Définition:

En programmation orientée objet, un Framework est typiquement composé de classes mères qui seront dérivées et étendues par héritage en fonction des besoins spécifiques à chaque logiciel qui utilise le Framework. Avec un Framework orienté objets, le programmeur qui utilise le Framework pourra personnaliser les éléments principaux du programme par extension, en utilisant le mécanisme d'héritage : créer des nouvelles classes qui contiennent toutes les fonctionnalités que met en place le Framework, et en plus ses fonctionnalités propres, créées par le programmeur en fonction des besoins spécifiques à son programme. Le mécanisme d'héritage permet également de transformer des fonctionnalités existant dans les classes du Framework

VI.2. Les différents Framework java:

VI.2.1. Hibernate :



VI.2.1.1.présentation de Hibernate : [12]

Hibernate est un Framework en open source, gérant la persistance des objets (qui peuvent être défini par les propriétés, les méthodes ou les évènements qu'il est susceptible de déclencher) dans une base de données relationnelle. La persistance des objets représente la possibilité d'enregistrement de l'état d'un objet, par exemple dans une base de données, afin de pouvoir le recréer plus tard. Quant à la base de données relationnelle, elle contient de nombreuses tables et l'information est organisée par différentes relations entre tables. Pour information, on appelle SGBDR un logiciel mettant en oeuvre une telle base de données.

L'ensemble des données nécessaires au fonctionnement de l'application est sauvegardé dans une base de données. La manipulation des données peut se faire de différentes manières : par l'accès directement à la base en écrivant les requêtes SQL adéquates, utiliser un outil d'ORM (Object Relationnal Mapping) permettant de manipuler facilement les données et d'assurer leur persistance : c'est le cas **d'Hibernate.**

Voici une vue (très) haut niveau de l'architecture d'Hibernate :

Ce diagramme montre Hibernate utilisant une base de données et des données de configuration pour fournir un service de persistance (et des objets persistants) à l'application. L'architecture la plus complète abstrait l'application des APIs JDBC/JTA sous-jacentes et laisse Hibernate s'occuper des détails.

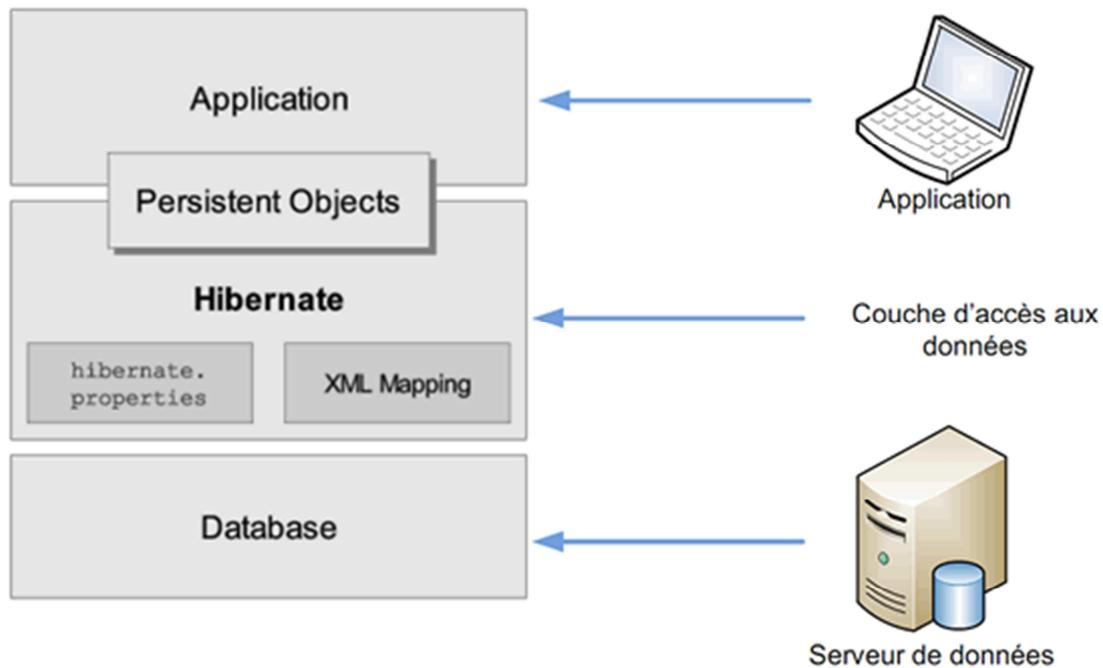


Figure III.6 : Architecture de Hibernate .

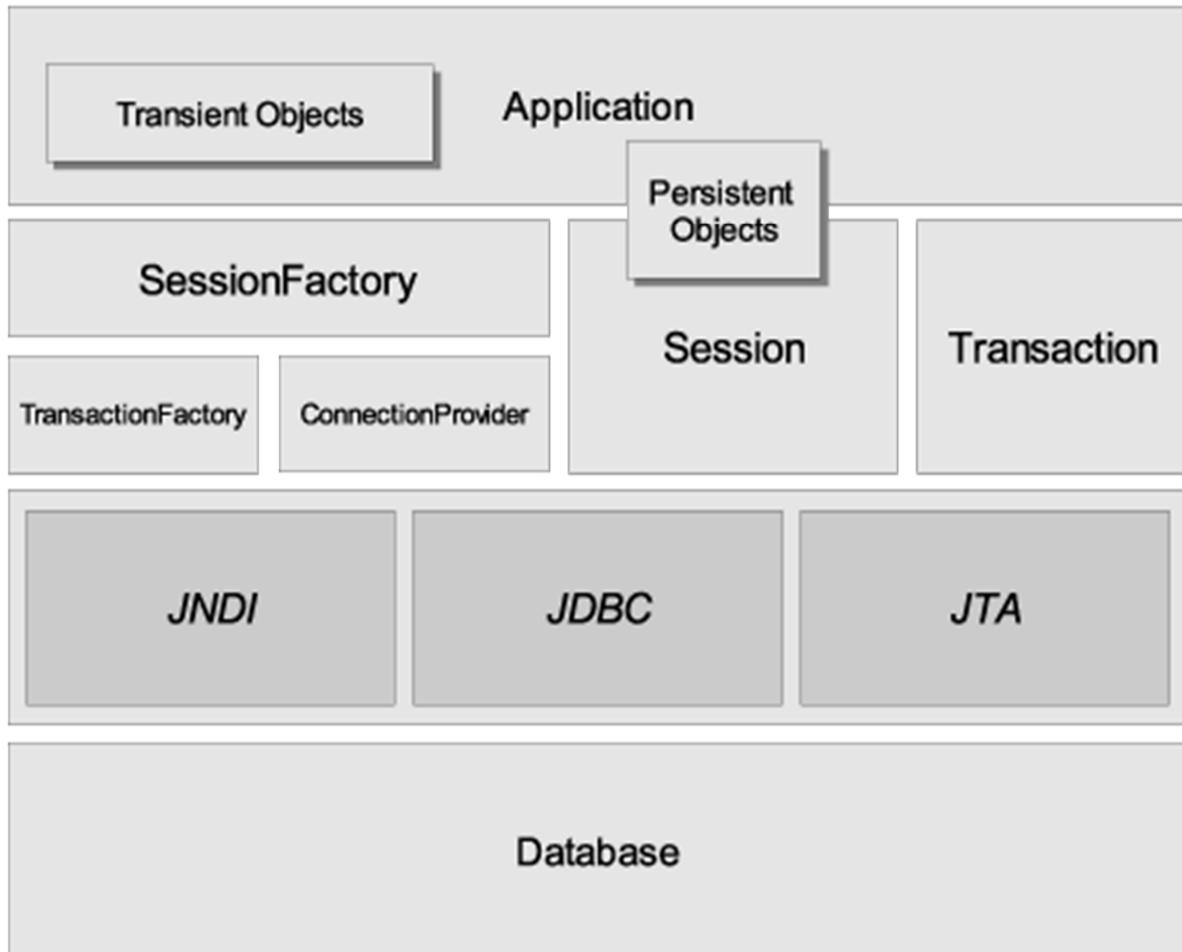


Figure III.7 : Architecture plus détaillé de Hibernate .

Nous allons maintenant définir les objets de diagrammes :

Session Factory :

Un cache immuable (threadsafe) des mappings vers une base de données. Il peut contenir un cache optionnel de données (de second niveau) qui est réutilisable entre les différentes transactions que cela soit au niveau du processus ou au niveau du cluster.

Session :

Un objet mono-threadé, à durée de vie courte, qui représente une conversation entre l'application et l'entrepôt de persistance. Elle encapsule une connexion JDBC. Elle contient un cache (de premier niveau) des objets persistants, ce cache est obligatoire. Il est utilisé lors de la navigation dans le graphe d'objets ou lors de la récupération d'objets par leur identifiant.

Objets et collections persistants

Objets mono-threadés à vie courte contenant l'état de persistance et la fonction métier. Ceux-ci sont en général les objets de type JavaBean (ou POJO) ; la seule particularité est qu'ils sont associés avec une session. Dès que la session est fermée, ils seront détachés et libre d'être utilisés par n'importe laquelle des couches de l'application, c'est à dire de et vers la présentation en tant que Data Transfer Objects (objets de transfert de données).

Objets et collections passagers :

Instances de classes persistantes qui ne sont actuellement pas associées à une session. Elles ont pu être mises en instances par l'application, et ne pas avoir (encore) été persistées ou par une session fermée.

Transaction :

Objet mono-threadé à vie courte utilisé par l'application pour définir une unité de travail atomique et qui abstrait l'application des transactions sous-jacentes qu'elles soient JDBC, JTA ou CORBA. Une session peut fournir plusieurs transactions dans certain cas.

Connection Provider :

Lieu de fabrication de connexions JDBC. Elle abstrait l'application de la source de données ou du manager sous-jacent de pilotes. Elle n'est pas exposée à l'application, mais peut être étendue et implémentée par le développeur.

Transaction Factory :

Fabrique d'instances de Transaction. Non exposée à l'application, mais peut être étendue et implémenté par le développeur.

Pour information, dans une architecture légère, l'application n'utilisera pas les APIs Transaction et Transaction Factory et/ou les APIs Connection Provider pour utiliser JTA ou JDBC.

VI.2.1.2.Cas d'utilisation :

Travailler dans les deux univers que sont l'orienté objet et la base de données relationnelle peut être lourd et consommateur en temps dans le monde de l'entreprise d'aujourd'hui. Hibernate est un outil de mapping objet/relationnel pour le monde Java. Le terme mapping objet/relationnel (ORM) décrit la technique consistant à faire le lien entre la représentation objet des données et sa représentation relationnelle basée sur un schéma SQL dans un projet.

Non seulement, Hibernante s'occupe du transfert des classes Java dans les tables de la base de données (et des types de données Java dans les types de données SQL), mais il permet de requêter les données et propose des moyens de les récupérer. Il peut donc réduire de manière significative le temps de développement qui aurait été autrement perdu dans une manipulation manuelle des données via SQL et JDBC.

VI.2.1.3. Les Avantage de Framework Hibernate :

- Hibernate génère le code nécessaire, ce qui rend l'application plus portable (s'adapte à la base de données).
- La persistance est transparente. Vous pouvez faire de vos classes métiers des classes persistantes sans ajout de code.
- La récupération de données est optimisée. On peut interroger la base de données de plusieurs façon (Requête SQL, langage HQL...).
- Portabilité du code en cas de changement de la base de données.

VI.2.2. Java server faces(JSF) :

VI.2.2.1. présentation de JSF : [13]



JSF, "Java Server Faces" est un Framework de type MVC, destiné aux applications web respectant l'architecture J2EE, ce Framework se repose sur la technologie de Facelets. Facelets est un Template web open source qui permet de générer des vues à partir de code xHTML.

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.

Il existe 2 implémentations de JSF :

- 1.1 associé à J2EE 1.4 : Sun
- 1.2 associée à Java EE 5 : Apache

Le principal atout de JSF est qu'il tient compte des différentes expériences acquises non seulement avec des technologies de type standard comme les Servlets, les JSP,... mais aussi avec les technologies de type framework comme Struts. JSF propose un framework qui puisse être mis en oeuvre par des outils facilitant le développement d'applications web. Rien n'empêche cependant que JSF puisse être utilisé par codage à la volée, néanmoins, l'utilisation d'un outil approprié est vivement recommandé afin d'exploiter rapidement les possibilités de JSF.

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, XML, ...) grâce à la séparation des problématiques de construction de l'interface et du rendu de cette interface.

VI.2.2.2. Cas d'utilisation :

JSF vous évite d'avoir à écrire le code - standard, passe-partout et pénible - responsable du regroupement des saisies utilisateurs (paramètres de requêtes HTTP), de leur conversion & validation, de la mise à jour des données du modèle, de l'invocation d'actions métiers et de la génération de la réponse. Ainsi vous vous retrouvez uniquement avec une page JSP ou une page XHTML (une Facelet) en guise de vue, et un JavaBean en tant que modèle. Cela accélère le développement de manière significative ! Les composants JSF sont utilisés pour lier la vue avec le modèle, et la FacesServlet utilise l'arbre des composants JSF pour effectuer tout le travail.

VI.2.2.3. Les Avantages de JSF :

- Architecture structurée autour des composants. Permet de créer des composants personnalisés.
- Système de navigation statique et dynamique très souple.
- Personnalisation des labels (internationalisation, messages d'erreurs).
- JSF a été créé dans le but de permettre des développements de type RAD (liaison champ/donnée).
- Séparation claire de la couche métier, permettant un prototypage efficace.
- Possibilité d'initialiser certains paramètres dans un fichier de configuration grâce aux managedbeans.
- Framework structuré.
- JSF est une spécification java 2EE.
- génère une forte dynamique et une communauté grandissante.

VI.2.2.4. Les Inconvénients de JSF:

- Framework relativement récent, et donc pas encore mûr.
- Documentation peu riche, du fait de sa maturité.
- Validation coté serveur uniquement.

VI.2.3. Prime Faces :

VII.2.3.1.Présentation de Prime Faces:[14]



Prime Faces est un open source bibliothèque de composants pour Java Server Faces , développé par le Premier Technologie . Il fournit un ensemble de composants plus souvent visuelles (widgets). Ceux-ci peuvent être utilisés par les programmeurs JSF en plus du petit ensemble de composants de base qui sont fournis avec la plate-forme de base de JSF pour composer l' interface utilisateur pour une application Web .

Cette bibliothèque est open source et se compose de trois projets :

- PrimeUI pour les composants graphiques destinés principalement aux interfaces desktop ;
- PrimeMobile pour les composants graphiques destinés aux appareils mobiles (basé sur jQuery Mobile);
- PrimePush pour utiliser du push dans un navigateur en communication client-serveur (basé sur la librairie Atmosphère).

VI.2.3.2.Cas d'utilisation de Prime Faces:

Sur un projet JSF il est très utile, voire indispensable, de sélectionner une bibliothèque de composants graphiques pour gagner en productivité et efficacité. Cela évite de réinventer la roue. Parmi les différentes bibliothèques disponibles, celle qui a particulièrement retenu l'attention des développeurs est PrimeFaces.

PrimeFaces est une suite de composants JSF qui permet de développer des applications Web en Java. La composition des pages se fait en utilisant des balises HTML un peu particulières et des JavaBeans sont utilisés côté serveur pour stocker les données. Le principal avantage est que la mise au point des pages Web peut être faite en modifiant les pages dynamiquement.

VI.2.3.3. Les Avantages de Prime Faces :

- Les composants de Prime faces proposent des fonctionnalités prêtes à l'emploi, nombreux, divers et de qualité.
- Enrichissement des librairies qui ajoute régulièrement de nouvelles améliorations et de nouveaux composants.
- De nombreux composants sont disponibles dans des librairies tierces qui sont par ailleurs généralement facile d'utilisation.
- De nombreux forum et sites de supports disponibles.

- La Simplicité et la performance.
- Facilite son utilisation.
- Communauté forte Commentaires.

VI.2.3.4. Les inconvénients de Prime Faces:

- L'inconvénient principal de **Prime Faces** est qu'il y a beaucoup de XML à manipuler et que certaines erreurs ne se voient qu'en cours d'exécution.
- Nécessite une bonne connaissance de JSF.

VI.2.4. Springs :

VI.2.4.1. Présentation de Springs : [15]

Springs est le principal Framework de développement Java utilisé en entreprise. Ce succès tient au fait qu'il permet de développer facilement et rapidement des applications robustes, légères et bien testées. Apprécié également par les cellules d'architecture et les DSI, Spring permet d'avoir des applications bien découpées en couches, facilement maintenables et évolutives. Cependant, si une configuration Springs simple est à la portée de tous, avoir une architecture Springs correcte, en comprendre le fonctionnement interne et connaître les fonctionnalités avancées du Framework sont des tâches nettement plus ardues. L'objectif de cette formation est de donner les bonnes bases pour comprendre les principaux aspects de Springs, de manière à ce que les participants soient capables de :

- développer une application Spring complète (persistance, métier, validation, transactions, sécurité, Web, Ajax...)
- comprendre le fonctionnement interne de Spring et donc être autonomes lorsqu'un bug ou une situation anormale se produit
- avoir intégré la philosophie de développement Spring, afin de pouvoir facilement prendre en main un sous-projet ou une fonctionnalité Spring qu'ils ne connaissent pas encore

VI.2.4.2. Cas d'utilisation de Springs :

Divers modules Springs Framework viennent en complément du noyau de base pour permettre l'intégration avec les autres bibliothèques et Framework, notamment :

- Accès et intégration des données pour l'intégration de JDBC, des ORM, de la sérialisation des données, de JMS et des transactions.
- AOP et instrumentation pour la programmation par aspects (AspectJ)

- Web pour la réalisation d'applications web avec les technologies Servlet et websocket
- Test pour l'intégration avec des Framework de test comme JUnit ou TestNG.

VI.2.4.3. les Avantages de Springs :

- Le développement de solutions à l'aide de Java nécessite l'usage de Framework applicatif qui nous évite de réinventer la roue. Dans le monde Java, le Framework Springs est considéré comme un standard de fait.
- Plus léger que les implémentations standards JEE (Java Enterprise Edition), Springs reste la préférence tant des architectes techniques que des développeurs.
- La modularité du Framework Springs permet de répondre à énormément de besoins.
- S'intègre avec plusieurs technologies cotées Vue.

VI.2.4.4. Les inconvénients de Springs :

- La configuration est un peu verbeuse.
- La mise en œuvre de Springs n'est pas toujours aisée.

VI.2.5. Qt Jambi :



VI.2.5.1. Présentation de Qt Jambi : [16]

Qt Jambi est un Framework Java permettant de créer des applications riches basées sur la bibliothèque logicielle Qt. C'est la seule version de Qt utilisant un autre langage de programmation que le C++ supporté officiellement par Qt Software. En février 2009, Qt Software a annoncé¹ qu'il arrêterait le développement de ce Framework après la sortie de la version 4.5. Depuis, il est supporté par la communauté.

Même si Qt Jambi utilise Java Native Interface (JNI) pour faire des appels à la bibliothèque Qt C++, rien n'est prévu dans l'interface de programmation pour qu'une application Qt Jambi puisse s'interfacer avec une application Qt C++.

VI.2.5.2.Cas d'utilisation de Qt Jambi :

- Qt-Jambi est un Framework Java permettant de créer des applications riches basées sur la bibliothèque logicielle Qt, reconnue dans le monde C++. C'est la seule version de Qt utilisant un autre langage de programmation que le C++ supportée officiellement.
- Qt Jambi est une technologie prototype qui permet aux développeurs Java d'un cadre de développement de logiciels multiplateforme haute performance. Qt Jambi fait C++ les bibliothèques Qt disponible à la communauté Java. Comparé à d'autres cadres tels que Java Swing et SWT, Qt Jambi a une API intuitive et puissante.
- Qt Jambi fournit le Java User Interface Compiler (JUIC), sa propre version adaptée de l'interface utilisateur du compilateur de Qt. JUIC permet aux développeurs Java de prendre l'avantage de [Qt Designer](#), l'outil de Trolltech pour la conception et la création d'interfaces utilisateur graphiques à partir de composants de Qt, en convertissant [Qt Designer](#) fichiers de définition de l'interface utilisateur de fichiers source Java.

VI.2.5.3. Les Avantages de Qt Jambi :

Ceux-ci et beaucoup d'autres caractéristiques font de Qt Jambi un choix idéal pour les équipes de développement d'applications avancées client riche avec Java. Si Java est la langue de votre choix, ou si vous souhaitez intégrer C++ et Java composants dans un seul projet, Qt Jambi peut offrir les avantages suivants:

✓ **Efficacité accrue de développement**

Qt Jambi augmente l'efficacité de client riche développement Java, et accélère les délais de commercialisation. Grâce à une interface intuitive, facile à apprendre API et les outils de développement intégrés, Qt Jambi permet le développement rapide d'applications riches indigènes, des clients.

- Raccourcir les courbes d'apprentissage, le développement de la vitesse et la facilité d'entretien grâce à l'intuitif API Qt Jambi.
- Offrir plus de fonctionnalités, plus rapide et avec plus de précision que ce qui est possible avec d'autres outils Java.
- Accélérer le développement grâce à des outils intégrés tels que le Qt Designer présentation visuelle constructeur .
- Exploiter tous les avantages de Qt du langage de programmation Java.

✓ Liberté et flexibilité

Qt Jambi fournit aux équipes de développement de la liberté d'intégrer de façon transparente C++ et le code Java et des compétences, travailler avec des outils et des environnements de développement tiers populaires et développer de puissantes applications clientes riches dans la langue de leur choix.

- Combinez C++ et Java code dans un seul projet, et permettre une collaboration plus efficace dans des équipes mixtes
- Rester libre de travailler avec Qt Jambi intérieur IDE populaires comme Eclipse
- Intégration avec les bibliothèques et les outils tiers de premier plan comme OpenGL®

✓ Assurances d'un cadre éprouvé

Eh bien testé et bien documenté, activement maintenu par une équipe de développeurs, et soutenu par un riche écosystème de soutien professionnel, de la formation et des services, Qt Jambi offre toutes les garanties d'un cadre solide, mature.

- S'appuyer sur le bilan de 12 ans de Trolltech de fournir des outils de développement de haute qualité
- Tournez à une équipe de Qt Jambi professionnels soutien ingénieurs disponibles pour vous aider dans votre développement.
- Comptez sur des améliorations régulières et ajouts de fonctionnalités fournis par une équipe de développement dédiée .
- Découvrez les technologies de tiers, services de conseil, de formation et d'expertise dans de Trolltech grand écosystème commerciaux et open source.
- Joignez-vous à Adobe®, Google™ et Skype™, et over 4500 entreprises dans le monde qui comptent sur TrolltechQt et de livrer les principales applications.

VI.2.3. Sofi

VI.2.3.1.Présentation de Sofi : [17]

SOFI (Framework) est un cadre créé pour **simplifier le développement d'applications Web** avec les technologies **Java**. Pour les organisations qui ont choisi de développer en Java, une problématique commune se pose quant au choix du « Framework » de développement puisque, dans le monde du logiciel libre, plusieurs cadres d'applications sont disponibles et complémentaires, mais non intégrés.

SOFI est né de ce besoin d'intégration, c'est-à-dire qu'il est un canevas d'applications qui orchestre plusieurs cadres issus du logiciel libre et offre ainsi une couche de simplification. Avec une conception basée sur les meilleures pratiques du marché et intégrant des composants réutilisables et testés, SOFI permet de développer plus rapidement des applications Web en Java selon une architecture robuste et ouverte.

VI.2.3.2.Cas d'utilisation de Sofi :

SOFI favorise l'utilisation d'outils pour l'automatisation et la gestion des dépendances de bibliothèques pour tous les projets. C'est à l'aide des outils Ant et Ivy que les projets SOFI peuvent simplifier le développement des projets Java.

VI.2.3.3. les Avantages de Sofi :

SOFI inclut l'expertise de nombreux projets Web en Java et favorise l'ajout de nouvelles fonctionnalités via les équipes de développement Java . Une prémisses fort importante à respecter lors d'un changement de SOFI, c'est que l'on doit conserver la rétrocompatibilité avec les versions antérieures. Ainsi, SOFI se voit être une solution très évolutive pour les systèmes, en plus d'offrir un potentiel d'évolution hors pair. SOFI favorise aussi l'intégration de nouvelles technologies Java offertes sur le marché. Ainsi, toute communauté SOFI aura la chance de profiter de ces innovations ou améliorations dans leurs futurs développements. Voici une liste des avantages offerts par SOFI :

- Se doter d'un outil qui répond aux besoins utilisateurs
 - Déjà utilisé par des milliers d'utilisateurs.
 - Offre clé en main, plusieurs fonctionnalités demandées par les utilisateurs.
- Offrir une normalisation aux développeurs et aux utilisateurs
 - Des normes d'interfaces et de programmations sont offertes.
- Offrir plusieurs solutions d'architecture
 - Facilite ainsi le développement d'applications de grande envergure en Java.
 - Permet de développer des systèmes agiles et évolutifs.
 - Permet de profiter d'un maximum de réutilisation pour tout développement.
- Offrir une infrastructure commune pour tous les services techniques normalement requis dans les développements d'applications Web.
 - SOFI centralise l'utilisation de plusieurs services, tels que :
 - Les messages.
 - Les libellés.
 - Les composants de navigation (menu, service, onglet, etc.).
 - Les paramètres systèmes et domaines de valeurs.
 - La sécurité.
- Profiter d'une communauté grandissante
 - Centralisation de l'information offerte sur le Portail SOFI.
 - Forum de discussions.
 - Plusieurs experts et ressources d'expérience.
- Assurer aux programmeurs une utilisation des meilleures technologies offertes sur le marché
 - Déjà basé sur plusieurs cadres d'applications renommés, SOFI offre la possibilité de grandir.

- Permet de réduire les coûts de développement d'une application en Java, et ce, peu importe son envergure.
- Diminuer le temps de programmation d'unités de traitement
 - Permet de réduire le temps ainsi que les coûts de maintenance de l'application
- Ajouter des fonctionnalités d'une grande richesse avec des fonctionnalités Ajax intégrées.
 - Se distingue par sa simplicité de développement et par les nombreux exemples d'utilisation.
- Se conformer aux bonnes pratiques d'un développement par couches basées sur l'architecture orientée services (AOS) .
 - SOFI favorise, par son architecture, le développement d'applications orientées service.

VII. La comparaison entre les différents Framework :

	Date de création	Open source	MVC	usage	Licence	Template	langage	Composants ou requête	Facilite de prise en main /légèreté/simplicité
Qt jambi	1991	oui	Oui	Application	BSD	/	C++/java	composants	Relativement complexe (nécessite une certaine maîtrise de c++)
JSF	2004	oui	Oui	Application web	Logicile libre	Facelete/beans	java	composants	NON. Technologie des servlet a maîtrise
Prime Faces	2008	oui	Oui	Application web	Logicile libre	Fichier XHTML	HTML/css	composants	Relativement complexe (nécessite une certaine maîtrise de JSF)
Hibernate	2006	oui	Oui	<u>persistance & mapping</u>	Logicile libre	Fichier XML	java	composants	NON. Nécessite le temps d'apprentissage
Spring	2004	oui	Oui	d'application d'entreprise	Apache	Jsp,commons ,tiles,velo,city,thymeleaf, more	java	composants	Oui (Nécessite pas le temps d'apprentissage et facile a maîtrise)
Sofi	2003	oui	Oui	Application web	Apache tomcat	/	java	composants	Oui (rapidité de développement)

Tableau III.1 : Comparaison des différentes Framework

VIII .Choix du Framework pour j2ee/java : [18]

Il convient avant tout de définir des critères avant d'élire le **meilleur Framework java**. En voici quelques-uns :

- **Prise en main** : dans tous les Framework, il est nécessaire de passer quelques temps afin de le prendre en main. Cette période peut être plus ou moins longues (de quelques heures à plusieurs semaines).
- **Flexibilité** : le Framework peut imposer des contraintes très fortes pour obliger tous les développeurs à développer de la même façon. Mais cela peut finir par être limitant.
- **Taille des projets** : les petits projets peuvent être développés avec un Framework .
- **Documentation** : une documentation complète est un réel plus !
- **Performance** : certaines Framework sont trop gourmands. Même le moindre « Hello World » peut nécessiter l'appel à plus de 100 fichiers différents.
- **Communauté** : un forum actif sera synonyme de personne prêt à vous aider en cas de problème
- **Évolutivité.**

Tous ces critères peuvent être plus ou moins pondérés en fonction de ce que vous attendez d'un Framework.

Avant de choisir la techno, il faut savoir ce que tu veux faire.

Le framework Hibernate est un L'ensemble des données nécessaires au fonctionnement de l'application sont sauvegardées dans une base de données. La manipulation des données peut se faire de différentes manières : Par l'accès directement à la base en écrivant les requêtes SQL adéquates. Utiliser un outil d'ORM (objectRelationnalMapping) permettant de manipuler facilement les données et d'assurer leur persistance.

Hibernate s'occupe du transfert des classes Java dans les tables de la base de données (et des types de données Java dans les types de données SQL), il permet de faire des requêtes sur les données et propose des moyens de les récupérer. Il peut donc réduire de manière significative le temps de développement qui aurait été dépensé autrement dans une manipulation manuelle des données via SQL et JDBC. Le but d'Hibernate est de libérer le développeur de 95 % des tâches de programmation liées à la persistance des données communes. Hibernate vous aidera à supprimer ou à encapsuler le code SQL spécifique à votre base de données et vous aidera sur la tâche commune qu'est la transformation des données d'une représentation tabulaire à une représentation sous forme de graphe d'objets.

Les raisons de notre choix en insistant sur :

- Réduire le temps de développement de l'application en éliminant une grande partie du code SQL à écrire pour interagir avec la base de données et en encapsulant le code SQL résiduel.
- Les développeurs manipulent les classes dont les données doivent être persistantes comme des classes Java normales.
- Seules une initialisation correcte d'Hibernate doit être effectuée, et quelques règles respectées lors de l'écriture et de la manipulation des classes persistantes.
 - ✓ la complexité d'utilisation,
 - ✓ la documentation,
 - ✓ la communauté,
 - ✓ la pérennité.

Le Framework Hibernate est utilisé pour développer des applications Java plus robustes et plus efficaces, en facilitant la configuration de leurs différents composants et la réalisation des tests, tout en réduisant les efforts de développement.

IX. Conclusion :

Dans ce chapitre, nous avons fait une étude sur les Framework java et nous avons présenté deux types de Framework on a étudié chaque un des Framework de la même façon selon les mêmes critères. Le but de ce chapitre est de choisir un Framework avec lequel on va travailler et qui correspond à notre réalité et c'est Hibernate Framework. Ce chapitre nous a permis de prendre connaissance des différents types de Framework et leurs avantages.

De nos jours, il existe de plus en plus d'applications web et celles-ci sont de plus en plus compliquées. C'est pourquoi afin d'aider leur développement, on utilise des Framework. Ces Framework permettent d'assurer l'évolution et la fiabilité de ces applications web. En s'appuyant sur le modèle MVC ils permettent une meilleure séparation entre les métiers d'informaticiens et ceux d'infographistes.

I. Introduction :

Dans le but d'une meilleure organisation du travail, tout processus de développement de système informatique doit suivre une démarche claire et bien définie.

Dans ce chapitre nous avons mis en évidence les différents acteurs intervenants dans le système, ainsi leurs besoins durant la phase d'analyse ensuite nous arrivons à la conception qui est consacrée essentiellement à la réalisation des résultats obtenus pendant l'analyse . Pour cela nous avons porté notre choix sur le langage UML (UnifiedModelingLanguage) conçu pour la spécification, la visualisation, la documentation et la construction des systèmes à fortes composantes logiciels.

II. Présentation de l'UML : [19]

UML est un langage ou formalisme de modélisation orienté objet qui représente un moyen de spécifier et de représenter les composants d'un système informatique. UML est un standard car à partir de 1997, il est devenu une norme de l'Objet Management Group (OMG). Parmi les objectifs d'UML : être indépendant des langages de programmation et être adapté à toutes les phases du développement.

II.1. Définition :[20]

L'UML est un langage de modélisation graphique fondé sur des concepts orientés objets. Il est conçu pour présenter, spécifier, construire, et documenter un système logiciel .Il ne décrit pas de méthodes ou de processus de développement spécifique dans la mesure où elle ne présente aucune démarche.

II.2. Modélisation de l'UML : [21]

C'est une méthode qui consiste à créer une représentation simplifiée d'un problème : le modèle, grâce à celui-ci il est possible de représenter d'une manière simplifiée un problème, un concept et de le simuler.

La modélisation comporte deux composantes :

- L'analyse, c'est-à-dire l'étude du problème.
- La conception, soit la mise au point d'une solution au problème.

Véhicule une sémantique précise : la vue statique du système ou sa vue dynamique.

Les diagrammes d'UML :

UML est un langage de modélisation avec plusieurs objectifs qui en font un véritable outil de communication:

- Comprendre et décrire les besoins,
- Spécifier (un système),
- Etablir l'architecture logicielle.

Les diagrammes d'UML vont mettre en place deux parties:

- ✓ **Diagramme statique (Structurels):**
 - Diagramme de classe
 - Diagramme de composant
 - Diagramme de déploiement
 - Diagramme de cas d'utilisation
- ✓ **Diagramme dynamique (Comportementaux):**
 - Diagramme d'activité
 - Diagramme de séquence
 - Diagramme d'état de collaboration

Pour la modélisation des besoins, nous utilisons les diagrammes UML suivants :

- Diagramme de cas d'utilisation globale
- Diagramme de séquences
- Diagramme d'activité
- Diagramme de classe

IV. Analyse et conception :**IV.1 Analyse :**

Cette phase s'intéresse à la définition des besoins ainsi qu'au domaine couvert par l'application et cela pour objectif de spécifier de manière claire et détaillé l'application de gestion des achat locaux . Pour cela nous allons adopter la démarche suivante :

- (1) A partir de la définition des besoins nous allons identifier les acteurs et les activités desquels nous déduirons les cas d'utilisation ;
- (2) Ceux-ci nous permettent d'établir un ensemble de scénarios d'utilisation standards à l'aide des diagrammes de séquences ;

(3) Ces diagrammes nous aideront à identifier les classes métier nécessaires qui nous permettent d'aboutir au diagramme de classe Globale ;

La figure suivante illustre notre démarche de modélisation.

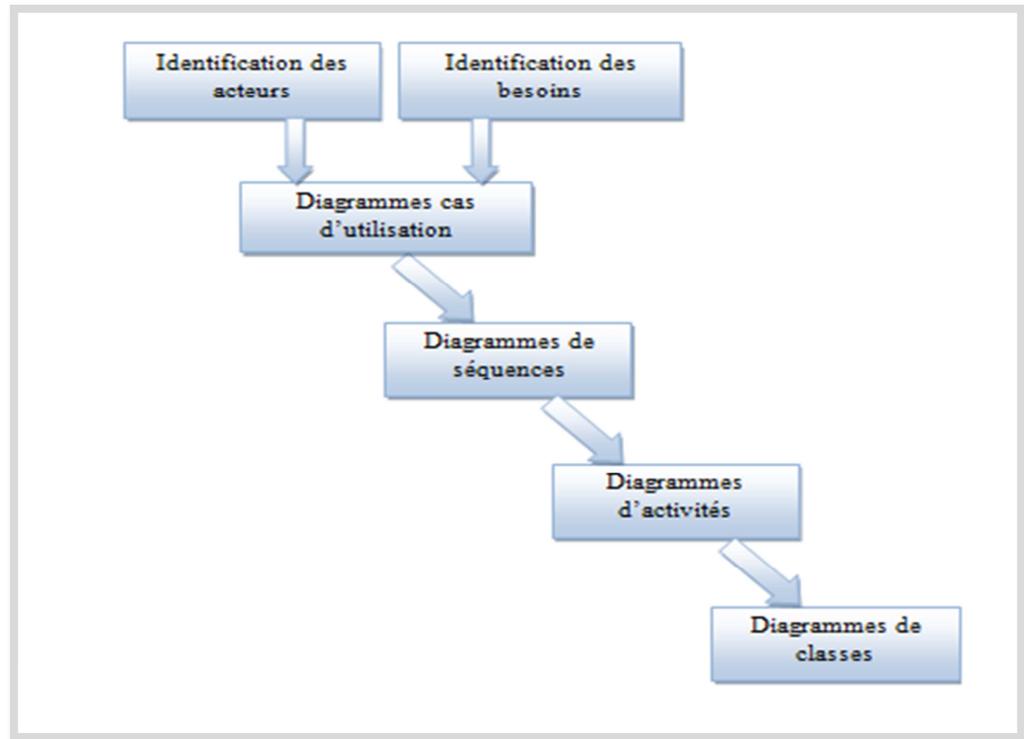


Figure IV.8: Méthodologie de modélisation de notre projet

IV.1.1 Spécification des besoins :

C'est une étape primordiale au début de chaque démarche de développement. Son but est de veiller à développer un logiciel adéquat, sa finalité est la description générale des fonctionnalités du système, en répondant à la question : Quelles sont les fonctions du système?

IV.1.2 Définition des besoins

IV.1.2.1 Identification des utilisateurs du nouveau système :[22]

A) Acteurs

Un acteur : est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.

Dans le cadre de notre travaille nous avons identifié quatre acteurs qui interagissent avec notre application :

- ✚ **Administrateur** : il gère les comptes utilisateur
- ✚ **Gestionnaire du Stock** : il gère et optimise la gestion des stocks (entrées et sorties des marchandises) pour minimiser le niveau de stocks sans risquer la rupture.
- ✚ **Les acheteurs** : L'acheteur est chargé de l'achat des produits nécessaires à l'activité de son entreprise. Il recherche les fournisseurs qui correspondent à ses exigences et négocie avec eux les prix et les délais de livraison. Il occupe un poste clé dans le secteur industriel et dans la grande distribution.

L'identification des acteurs de notre application nous a permis de déduire son diagramme de contexte.

B) Le diagramme de contexte de notre application : [23]

La description des différents acteurs permet de dégager ce qu'on appelle le diagramme de contexte pour le système, il permet de présenter l'utilisation du système par les différents acteurs au vue de la solution adoptée.

Dans la figure ci-dessous, nous avons illustré les différents acteurs qui interagissent dans notre système et ceci à travers un diagramme de contexte.

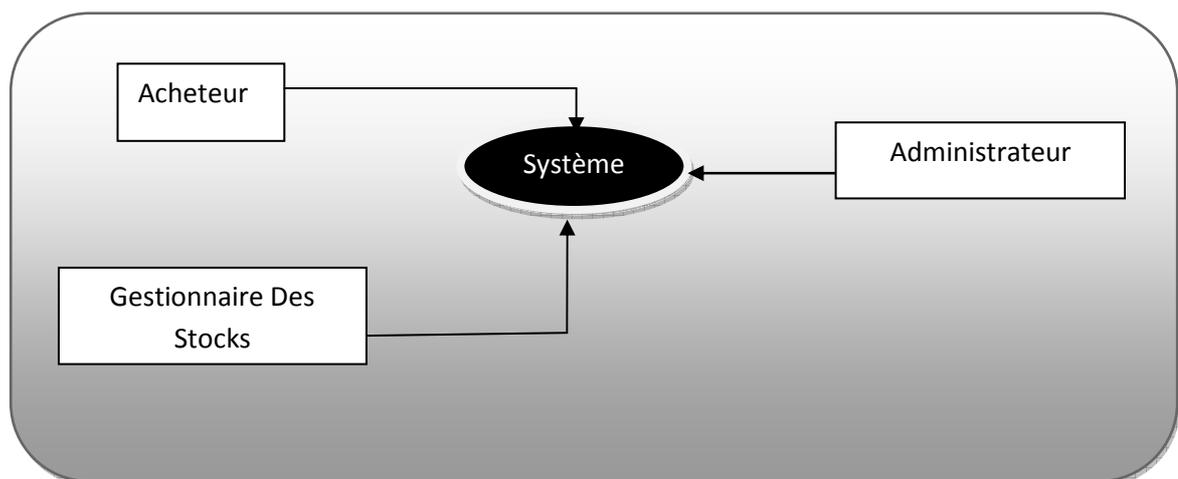


Figure IV.9: Diagramme de contexte de notre système

IV.1.2.2 Identification des cas d'utilisation : [24]

Un cas d'utilisation est une unité cohérente d'une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.

Le tableau suivant résume les cas d'utilisations ou fonctionnalité de notre système :

N°	Acteur	Cas d'utilisation (Taches)
1	Administrateur	T0 : Gestion des comptes
		T1 : Authentification
2	Acheteur	T2: Authentification
		T3: Gestion des fournisseurs
		T4: Gestion des Factures
		T5 : Gestion des commandes
		T6: Gestion des Offres
		T7 : Gestion des Consultation
		T8 : Gestion des articles
		T9 : Gestion des Réceptions
3	Gestionnaire des Stocks	T10: Authentification
		T11 : Gestion des DAI

Tableau IV.2 : Tableau des cas d'utilisation

Spécification des scénarios : Afin d'effectuer une tâche associée à un acteur, celui-ci doit effectuer un certain nombre d'actions. Cette séquence est appelée scénario.

Le tableau suivant rassemble tous les scénarios pouvant être déclenchés par les différents acteurs.

N°	Acteur	Taches	Scénario
1	Administrateur	T0 : Gestion des comptes	S0 : Ajouter un compte S1 : Modifier un compte S2 : Consulter un compte S3 : Désactiver un compte
		T1 : Authentification	S4 : Consulter espace personnel
2	Acheteur	T2 : Authentification	S5 : Consulter espace personnel
		T3 : Gestion des Fournisseurs	S6 : Ajouter un fournisseur S7: Modifier un fournisseur S8 : Consulter un fournisseur S9 : Supprimer un fournisseur
		T4: Gestion des Factures	S10 : Ajouter une Facture S11 : Modifier une Facture S12 : Consulter une Facture S13 : Supprimer une Facture
		T5 : Gestion des commandes	S14 : Créer un bon de commande S15 : Consulter un bon de commande S16 : Modifier un bon de commande S17 : Supprimer un bon de commande S18: Imprimer un bon de commande
		T6 : Gestion des Offres	S19 : Ajouter un Offre S20 : Modifier un Offre S21 : Consulter un Offre S22 : Supprimer un Offre S23: Imprimer un Offre
		T7 : Gestion des Consultation	S24 : Ajouter une Consultation S25 : Modifier une Consultation S26 : Consulter une Consultation S27 : Supprimer une Consultation S28: Imprimer une Consultation
		T8 : Gestion des articles	S29 : Ajouter un article S30 : Consulter un article S31 : Modifier un article S32: Supprimer un article S33 : Supprimer un article
		T9 : Gestion des Réception	S34 : Créer un bon de Réception S35 : Consulter un bon de Réception S36 : Imprimer un bon de Réception S37 : Modifier un bon de commande S38 : Supprimer un bon de commande
3	Gestionnaire du Stock	T10 : Authentification	S39 : Consulter espace personnel
		T11 : Gestion des DAI	S40 : Ajouter une DAI S41: Modifier une DAI S42 : Consulter une DAI S43 : Supprimer une DAI S44 : Imprimer une DAI

Tableau IV.3 : Spécification des scénarios des cas d'utilisation

IV.1.3 Diagramme des cas d'utilisation

A) Description graphique du cas d'utilisation « Gestion des comptes »

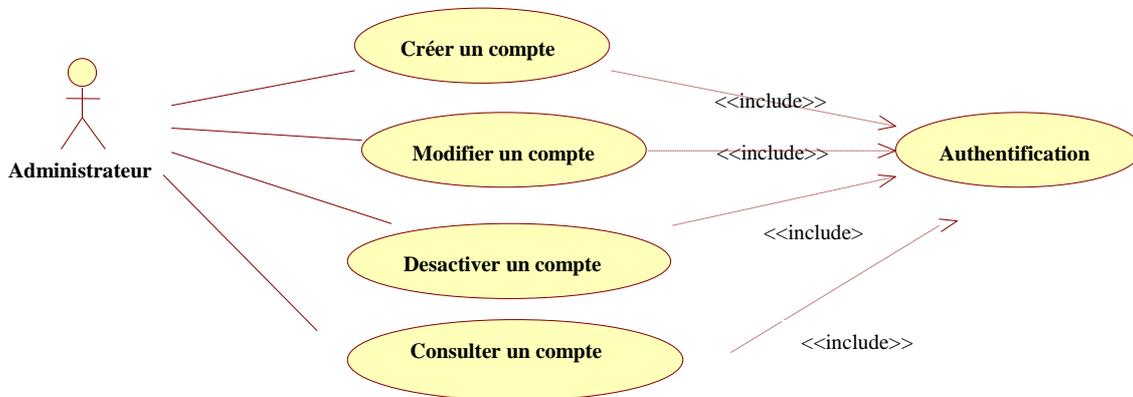


Figure IV.10 : Diagramme du CU des fonctions liées à la « Gestion des comptes »

B) Description graphique du cas d'utilisation « Authentification »

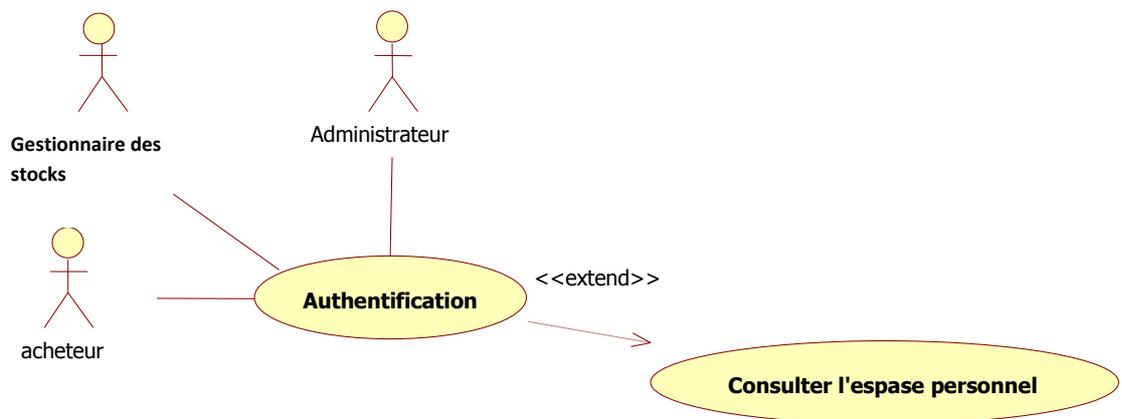


Figure IV.11: Diagramme du CU des fonctions liées à « Authentification »

C) Description graphique des cas d'utilisation Articles :

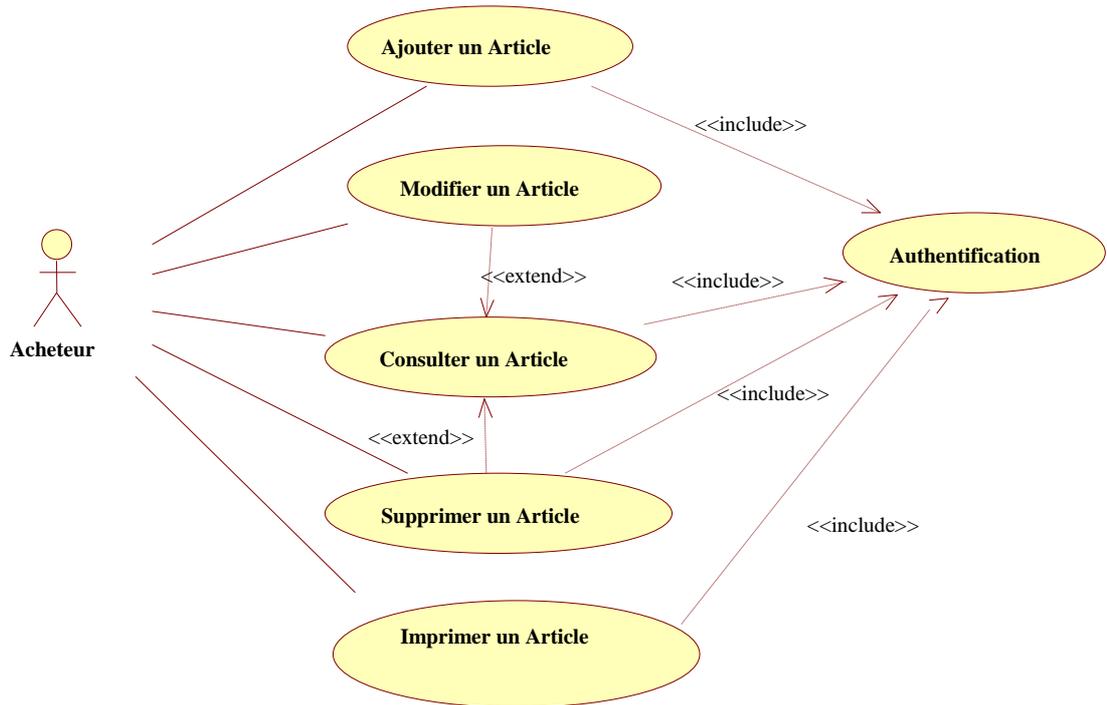


Figure IV.12: Diagramme du CU des fonctions liées à la " Gestion des articles"

D) Description graphique des cas d'utilisation Fournisseur :

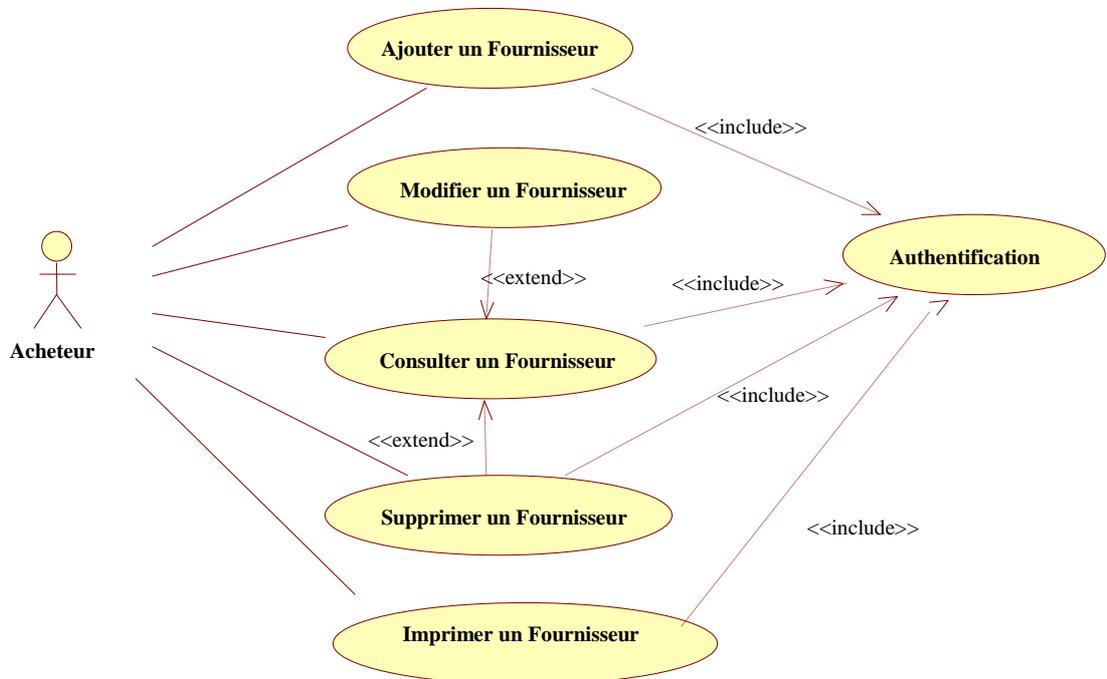


Figure IV.13: Diagramme du CU des fonctions liées au « gestion des fournisseurs »

E) Description graphique des cas d'utilisation gestion des DAI :

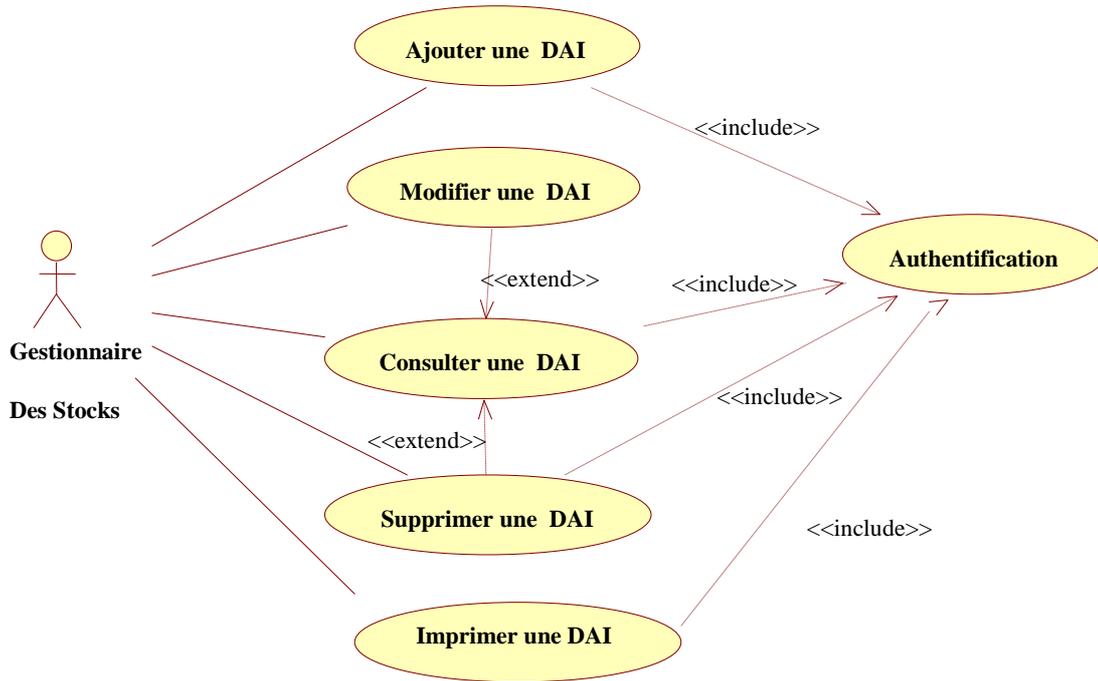


Figure IV.14: Description graphique du CU des fonctions liées à la « Gestion d'une DAI »

F) Description graphique des cas d'utilisation « Gestion des factures »:

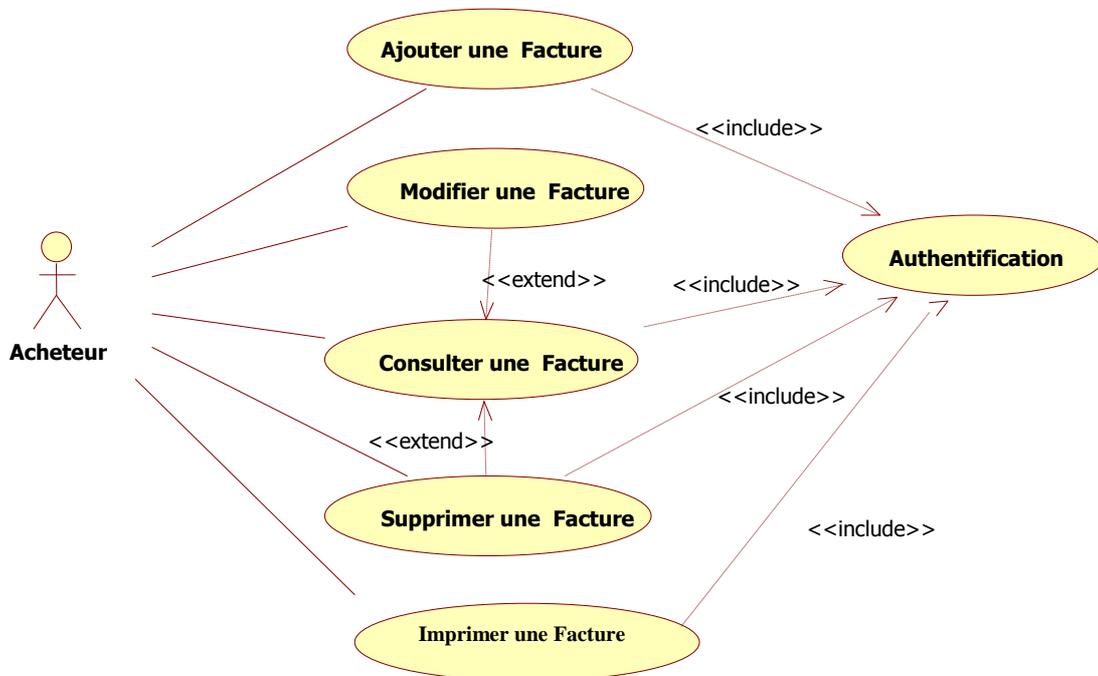


Figure IV.15: Diagramme du CU des fonctions liées à « Gestion des Factures »

G) Description graphique des cas d'utilisation Gestion des commandes :

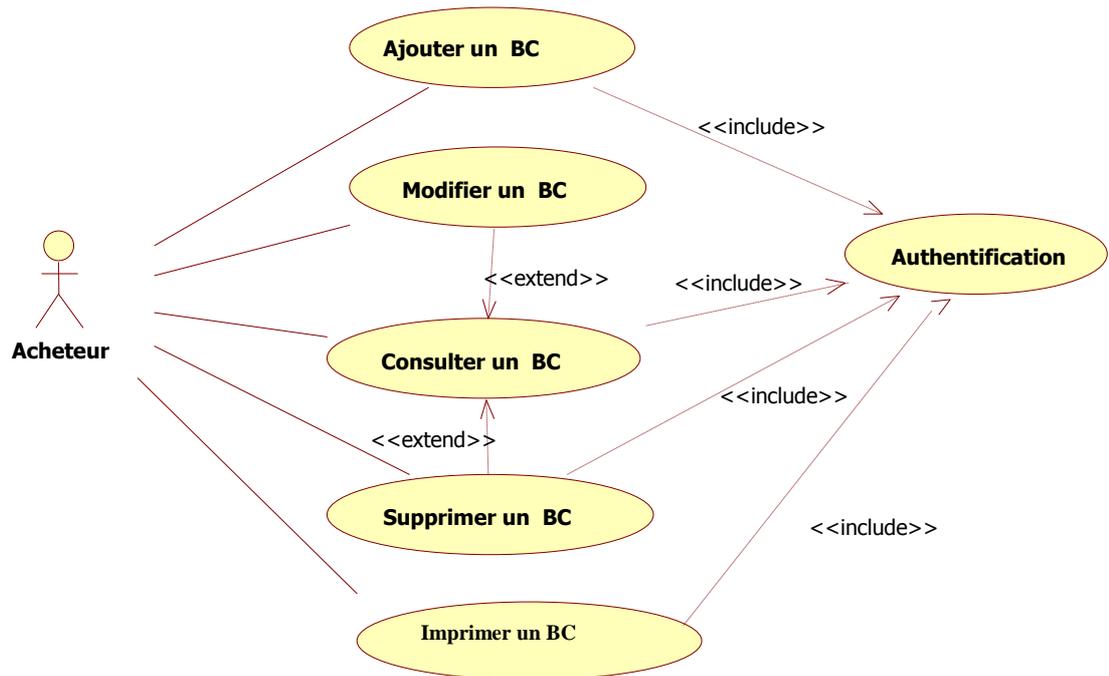


Figure IV.16: Diagramme du CU liées a la « Gestion des BC»

H) Description graphique des cas d'utilisation **Gestion des bons de Réception :**

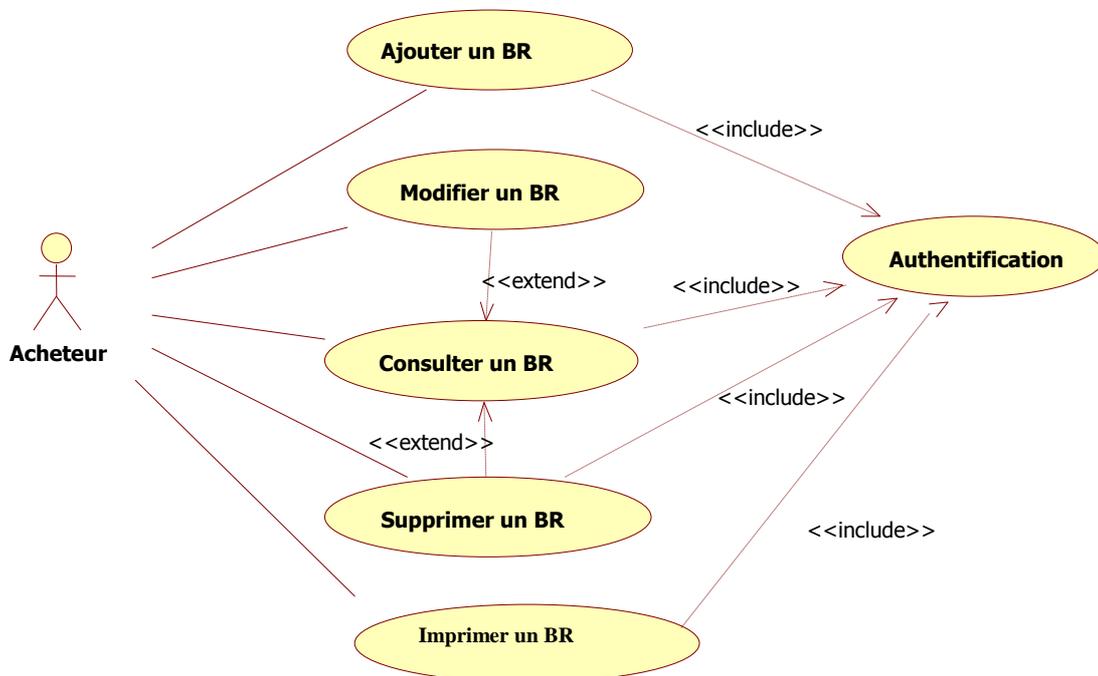


Figure IV.17: Diagramme du CU des fonctions liées à « Gestion des BR»

IV.1.4 Spécification de quelques cas d'utilisation :**❖ Cas d'utilisation « Authentification »**

Use case: Authentification.

Scenarios: S4, S5, S39.

Rôle: Administrateur, Acheteur et Gestionnaire des stocks.

Description :

- 1) L'utilisateur accède à la page d'authentification ;
- 2) Le système construit la page Authentification puis l'affiche ;
- 3) L'utilisateur saisit ses droits d'accès dans le formulaire authentification puis le valide en appuyant sur le bouton connexion ;
- 4) Si les données saisie sont correctes, le système l'envoie vers son espace sinon le renvoie au formulaire d'authentification.

❖ Cas d'utilisation « Ajout d'un compte »

Use case: Ajouter un compte.

Scenarios: S0.

Rôle: Administrateur.

Description :

- 1) L'utilisateur doit être authentifié au système en tant qu'administrateur.
- 2) L'acteur Administrateur clique sur le sous-menu Ajout du menu
Compte qui se trouve dans son espace.
- 3) Le système construit alors le formulaire d'ajout d'un compte correspondant puis l'affiche ;
- 4) L'acteur Administrateur remplit le formulaire puis le valide avec le bouton Enregistrer;
- 5) Le système vérifie alors les données dans la base de données puis construit un message d'erreur et renvoi le formulaire d'Ajout d'un compte si les données ne sont pas en norme ou si le compte à ajouter existe, sinon il met à jour la base de données puis construit une page de confirmation d'Ajout et l'affiche.

➤ **Cas d'utilisation « Modification d'un Article»**

Use case: Modifier un Article.

Scenarios: S31.

Rôle: Acheteur.

Description :

- 1) L'utilisateur doit être authentifié au système en tant que Acheteur.
- 2) L'Acheteur clique sur le sous-menu modifié du menu Article qui se trouve dans son espace .
- 3) le système construit une page intermédiaire pour chercher l'Article à modifier.
- 4) L'utilisateur saisit la désignation de l'article à modifier, puis il valide.
- 5) système vérifie alors si l'article existe dans la base de données ,le système lui construit le formulaire de modification correspondant au précédant choix puis l'affiche.
- 6) L'utilisateur modifie ces champs ensuite il valide sur le Bouton «Modifier».
- 7) Le système affiche le message d'enregistrement.

➤ **Cas d'utilisation « Suppression d'un fournisseur »**

Use case: Supprimer un Fournisseur.

Scenarios: S9.

Rôle: Acheteur

Description :

- 1) L'utilisateur doit être authentifié au système en tant que Acheteur .
- 2) L'acheteur clique sur le sous-menu supprimée du menu Fournisseur qui se trouve dans son espace .
- 3) Le système construit une page intermédiaire pour chercher le fournisseur a supprimée.
- 4) L'utilisateur saisit le nom de fournisseur a supprimée, puis il valide.
- 5) système vérifie alors si fournisseur existe dans la base de données le système le supprime de la base et indique un message de suppression .

❖ Cas d'utilisation « Consultation d'une demande d'achat interne « DAI »

Use case: Consulter une DAI.

Scenarios: S42.

Rôle: Gestionnaire des stocks.

Description :

- 1) L'utilisateur doit être authentifié au système en tant que Gestionnaire des stocks.
- 2) Le Gestionnaire des stocks clique sur le sous-menu Consultée du menu DAI qui se trouve dans son espace.
- 3) le système construit une page intermédiaire pour chercher la DAI a consultée.
- 4) l'utilisateur saisit le numéro de la DAI a consulter, puis il valide.
- 5) système vérifie alors si la DAI existe dans la base de DAI , le système affiche les informations de la DAI.

IV.1.5 Diagramme de séquence

Représente la succession chronologique des opérations réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre.

Nous allons décrire ci-dessous quelques diagrammes de séquences cas utilisation.

A) Diagramme de séquence « Authentification »

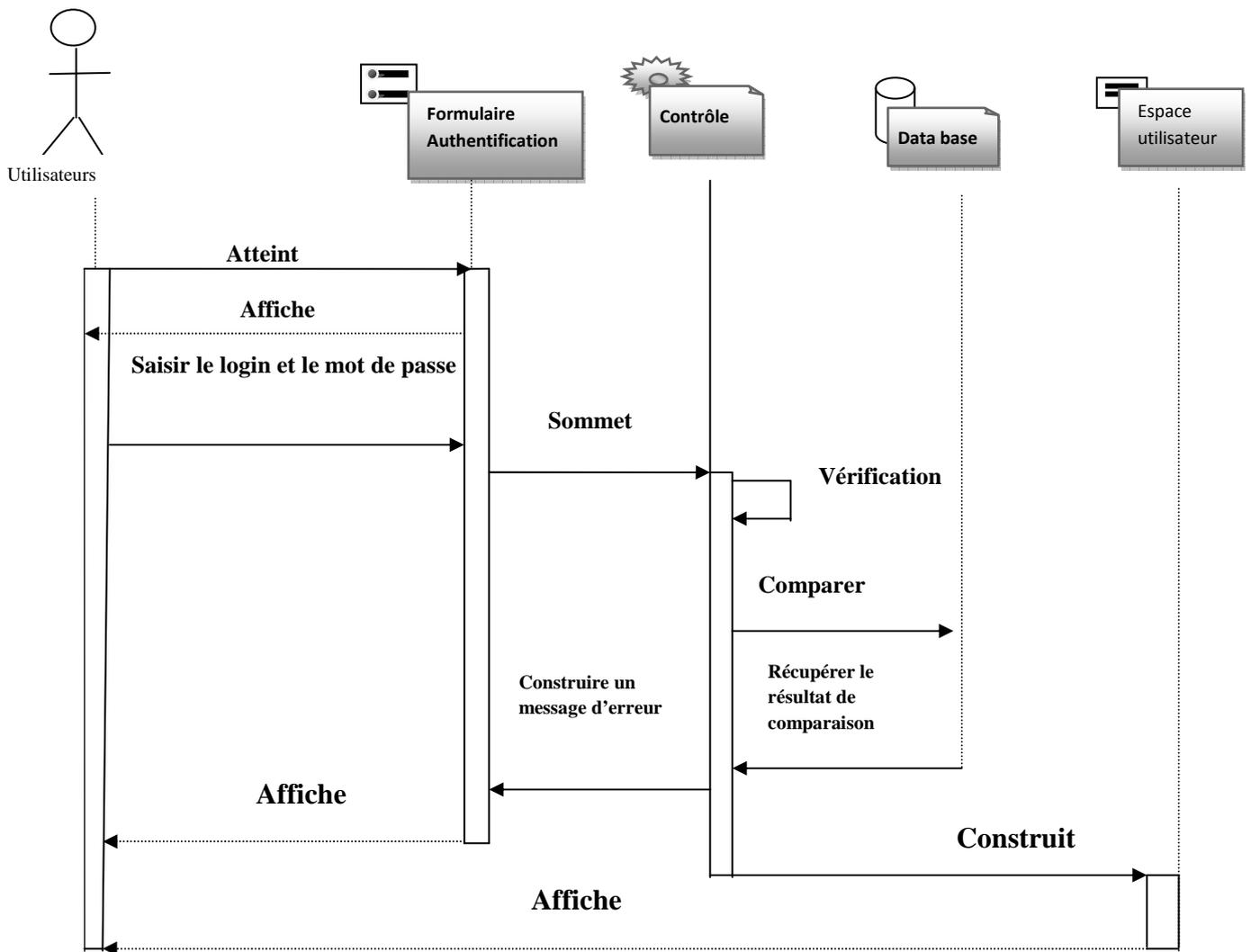


Figure IV.18: Diagramme de séquence du cas d'utilisation « Authentification »

B) Diagramme de séquence « Ajout d'un compte »

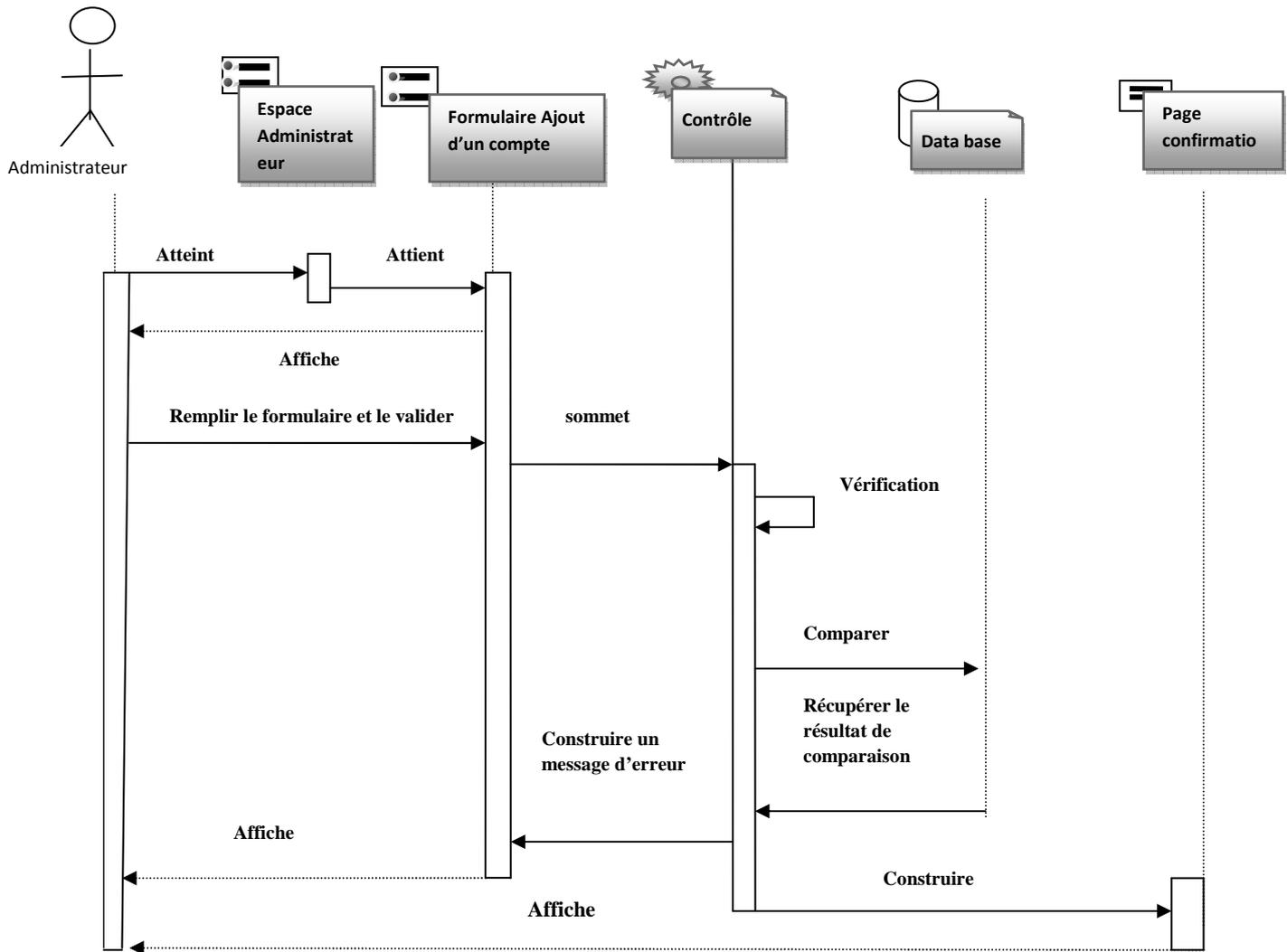


Figure IV.19: Diagramme de séquence du cas d'utilisation « Ajout d'un compte »

C) Diagramme de séquence « Modification d'un Article » :

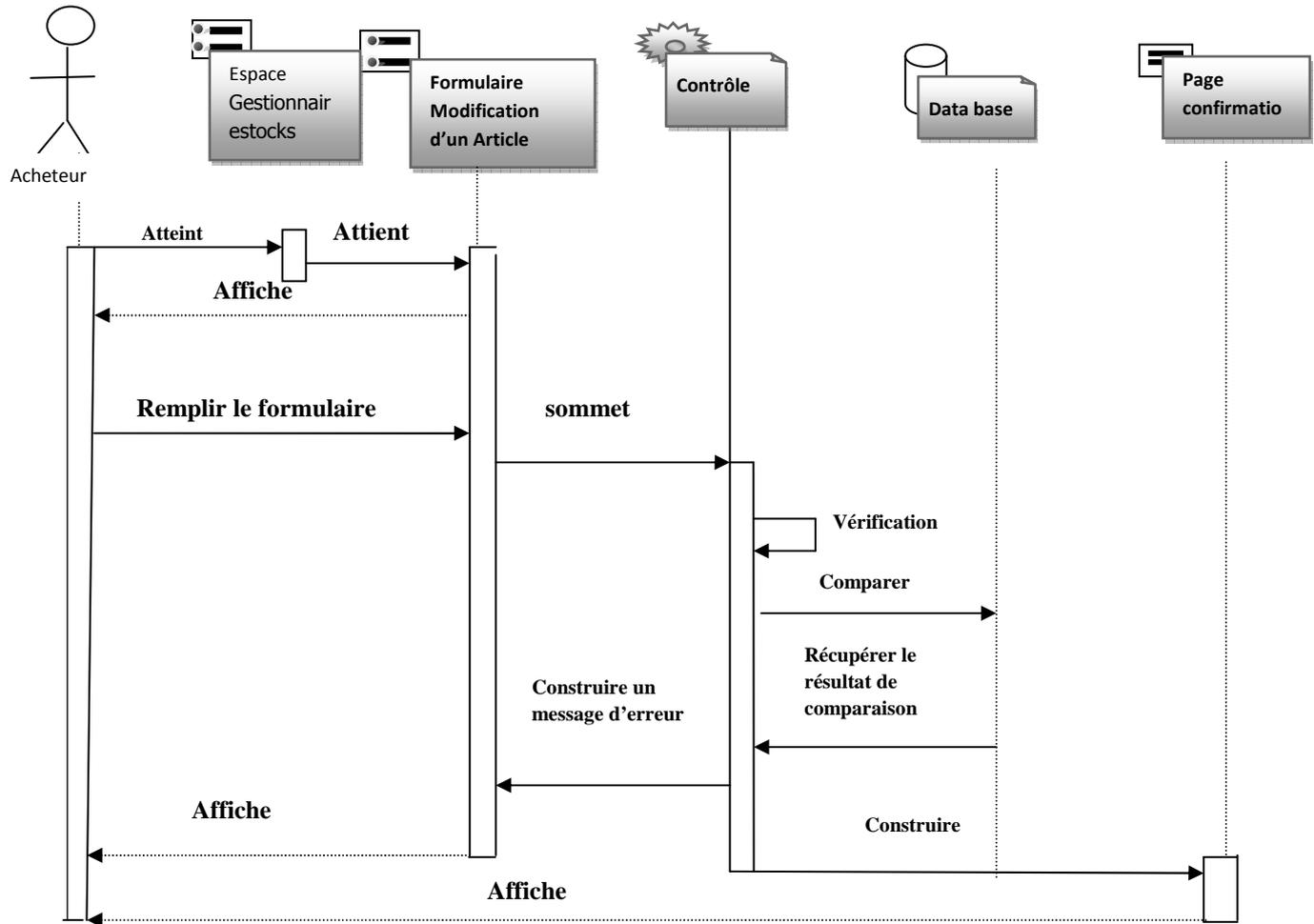


Figure IV.20: Diagramme de séquence du cas d'utilisation « Modification d'un Article »

D) Diagramme de séquence «Suppression d'un fournisseur»

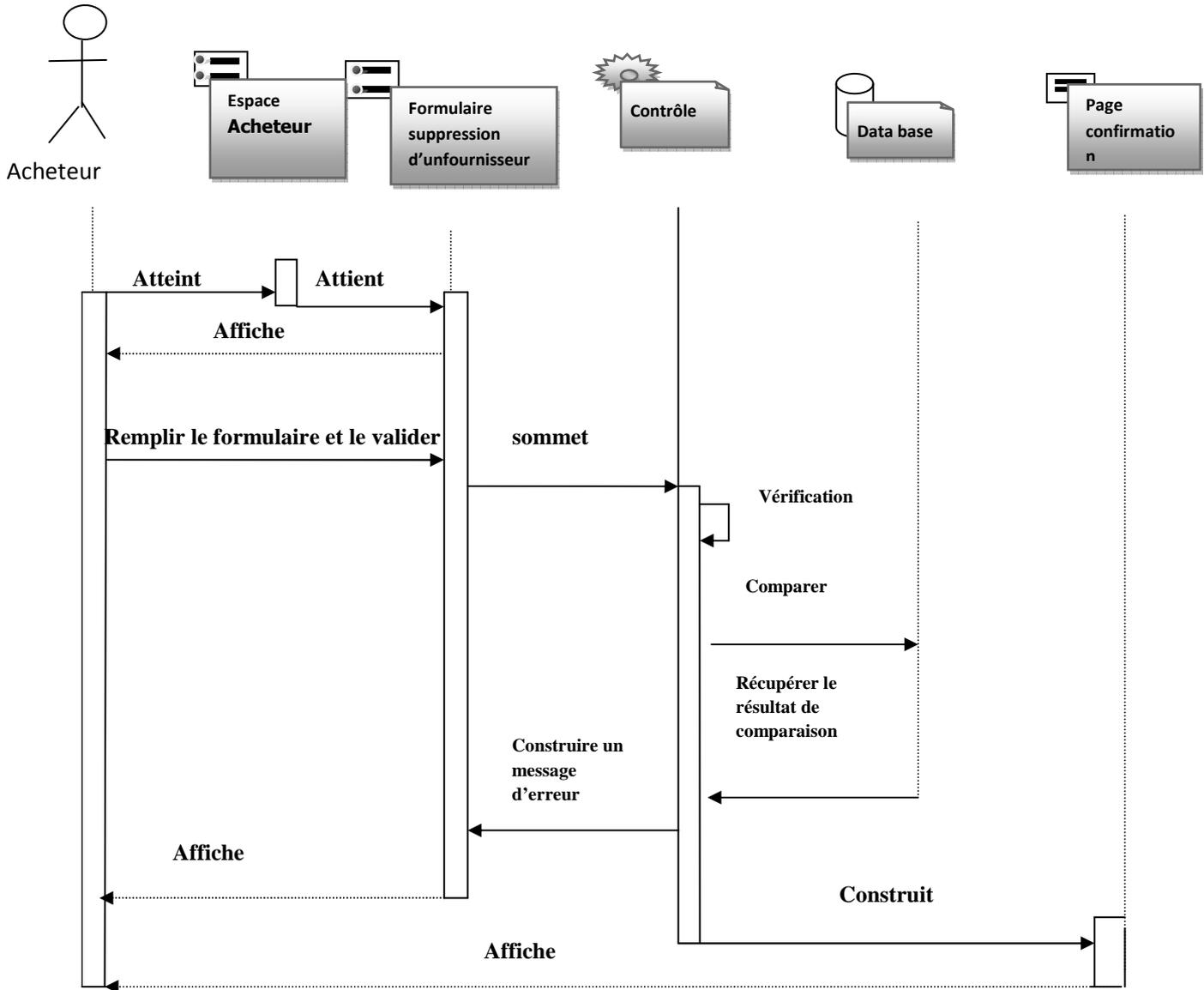


Figure IV.21: Diagramme de séquence du cas d'utilisation « suppression d'un fournisseurs »

IV.1.6 Diagrammes d'activités

Le diagramme d'activités n'est autre que la transcription dans UML de la représentation du processus telle qu'elle a été élaborée lors du travail qui a préparé la modélisation : il montre l'enchaînement des activités qui concourent au processus.

Dans ce qui suit nous allons présenter quelques diagrammes d'activité de l'application.

❖ Diagrammes d'activités « Authentification »

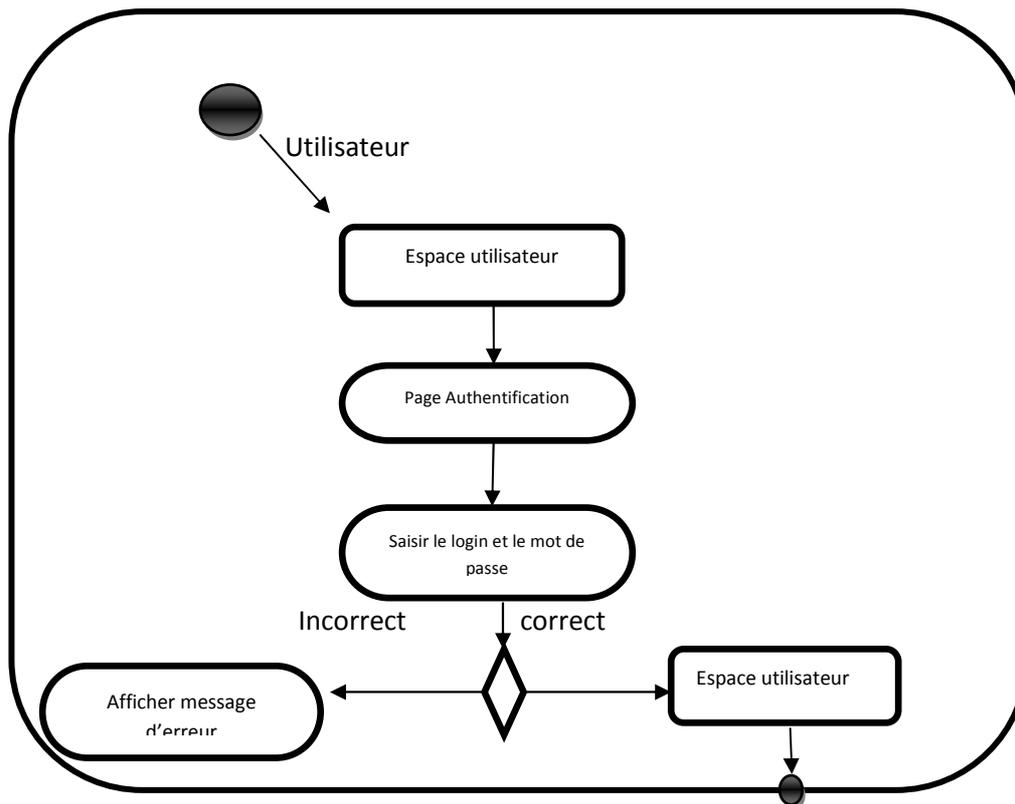


Figure IV.22: Diagramme d'activité du cas d'utilisation « Authentification »

❖ Diagrammes d'activités « Ajout d'un compte »

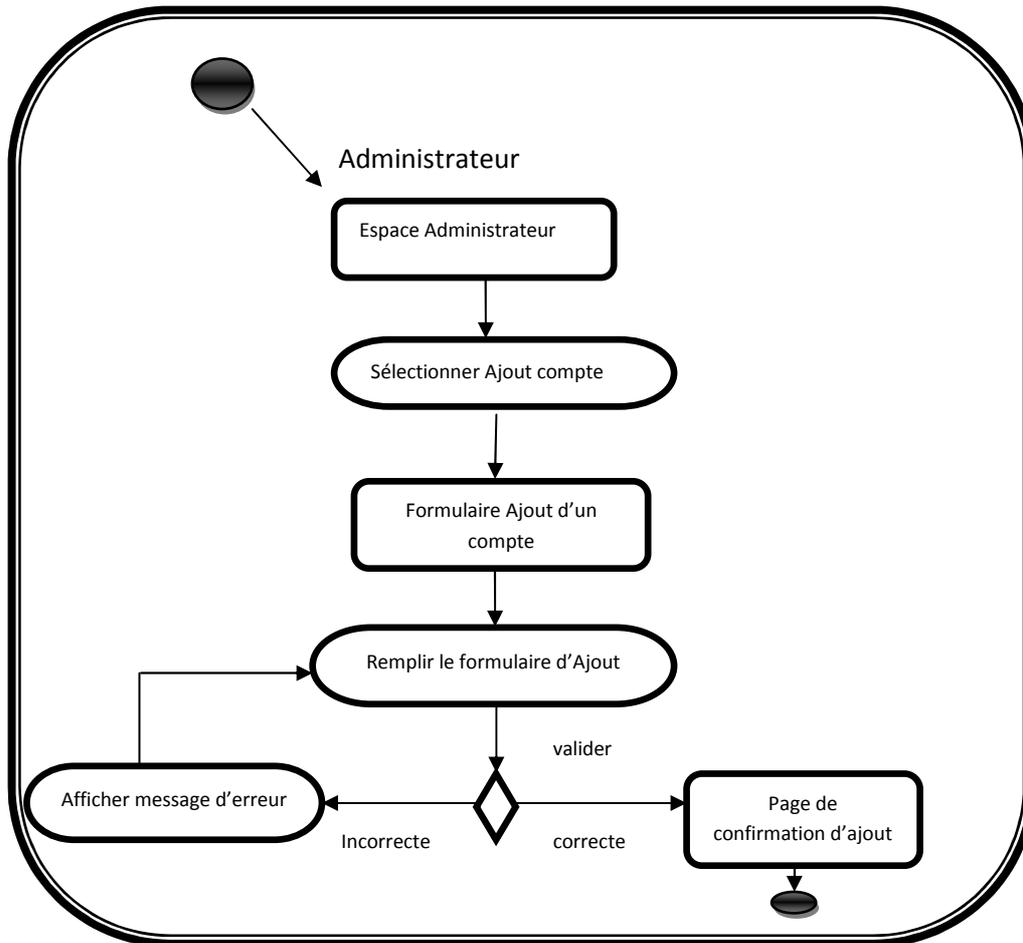


Figure IV.23 : Diagramme d'activité du cas d'utilisation « Ajout d'un compte »

Diagrammes d'activités « Modification d'un Article »

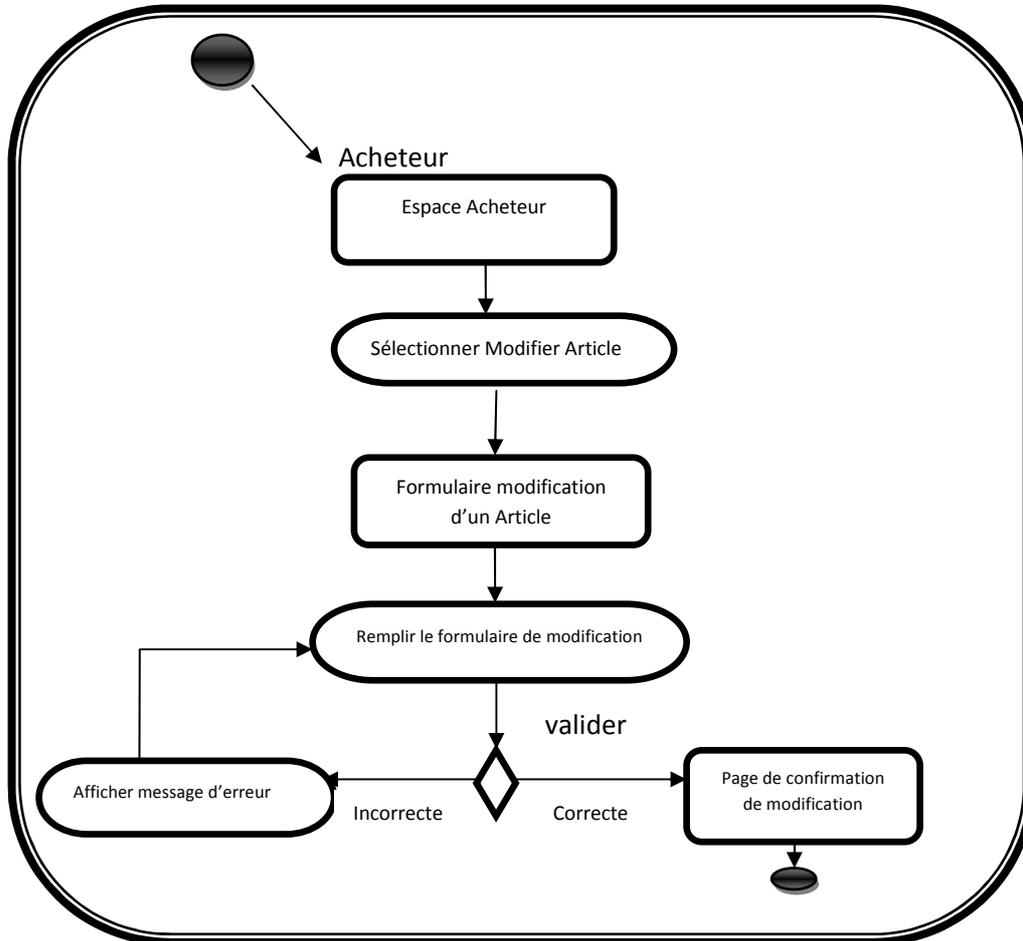


Figure IV.24 : Diagramme d'activité du cas d'utilisation « Modification d'un Article»

❖ Diagrammes d'activités « Suppression d'un fournisseur »

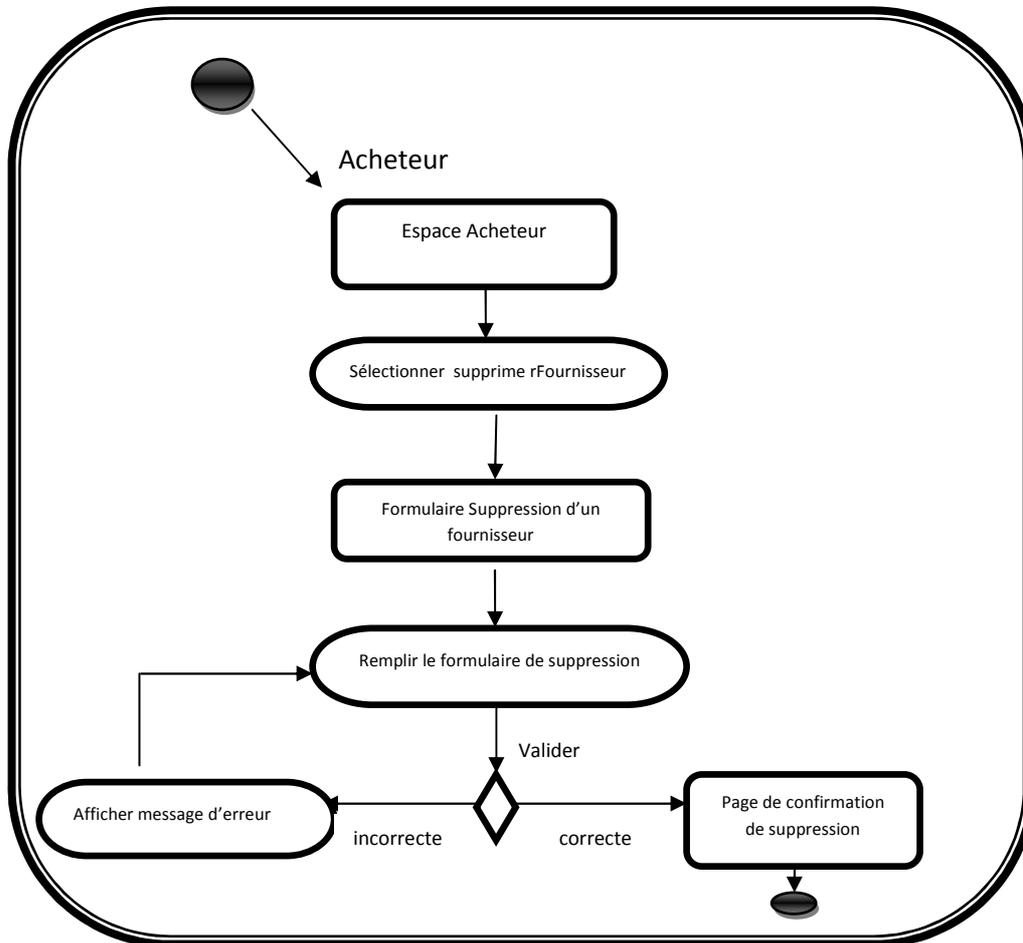


Figure IV.25: Diagramme d'activité du cas d'utilisation « Suppression d'un fournisseur »

IV.1.7 Définition des classes :❖ **Utilisateurs :**

Attributs : id utilisateur, login, mot de passe et profil.

❖ **Articles :**

Attributs : id article, id unité de mesure, désignation, date de création, cump, stock minimum, stock alarme, date de péremption et quantité en stock.

❖ **Bon de Commande :**

Attributs : id commande, id offre, numéro bc, numero dai, nom unite, nom fournisseur, date commande, montant HT, montant TTC, désignation article, quantité demande.

❖ **Consultation :**

Attributs : id consultation, date de consultation, numéro dai, état consultation.

❖ **Dai :**

Attributs : id dai, numéro, nom acheteur, nom unité, date émission, état dai, nombre de l'articles, désignation de l'articles, quantité demandé.

❖ **Offres :**

Attributs : id offre, id consultation, nom fournisseur, désignation article, quantité demande date offre, validité, montant ht, montant ttc, état offre.

❖ **Bon de Réception :**

Attributs : id réception, id facture, date réception, numéro ordre, numero BC, désignation article, nom fournisseur, état réception.

❖ **Unité de mesure :**

Attributs : id unité mesure, désignation, code.

❖ **Fournisseur :**

Attributs : id fournisseur, nom fournisseur, raisons sociale, adresse, téléphone, fax, type fournisseur, date création, domaine d'activité.

❖ **Facture :**

Attributs : id facture, id commande, désignation article, quantité demande, date facture, montant ht, montant ttc, type facture, état facture, mode règlement, montant lettre.

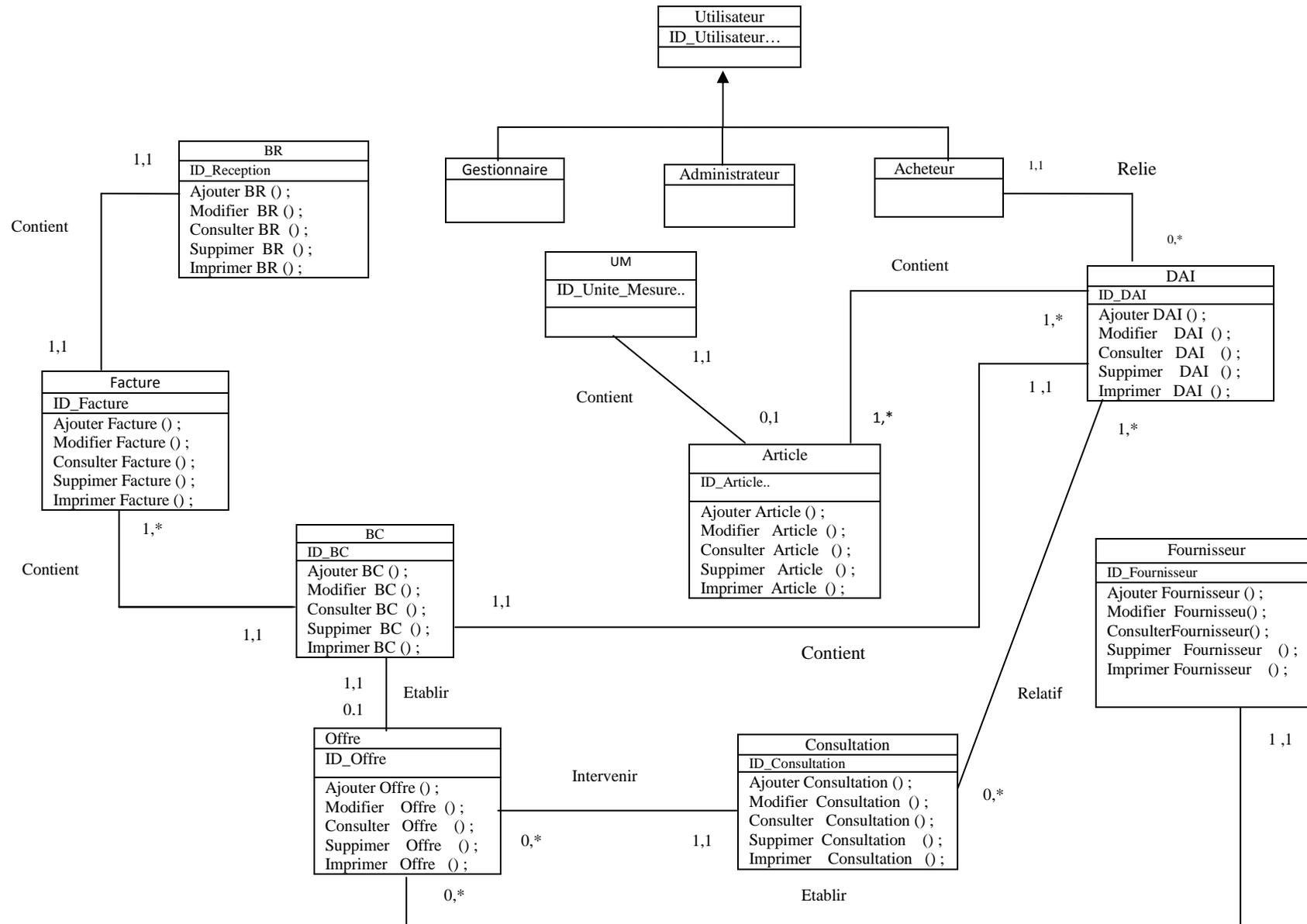


Figure IV.26 : Diagramme de classe globale

V. Conclusion

Afin de modéliser notre application, nous avons introduit l'un des langages de modélisation de l'UML, dans le but de spécifier les cas d'utilisation et concevoir les diagrammes de séquences et ceux de classes et les diagrammes d'activités. Enfin en a cité et défini toutes les tables utilisées dans la BDD.

Le chapitre qui suit sera consacré à l'implémentation de notre application.

I. Introduction :

Après avoir présenté dans le chapitre précédent le processus de conception de notre application, dans ce chapitre nous allons, tout d'abord décrire l'environnement de développement et les langages de programmation de notre application, pour mener à terme la réalisation de notre application ainsi que, quelques interfaces du logiciel.

II. Les outils de développement :

❖ Hibernate Framework :[25]

C'est un logiciel, écrit en java, qui permet de faire le mapping entre Objets Java et objets stockés en base relationnelle en assurant la persistance, et aussi s'occupe du transfert des classes Java dans les tables de la BDD et des types de données dans les types de données SQL.

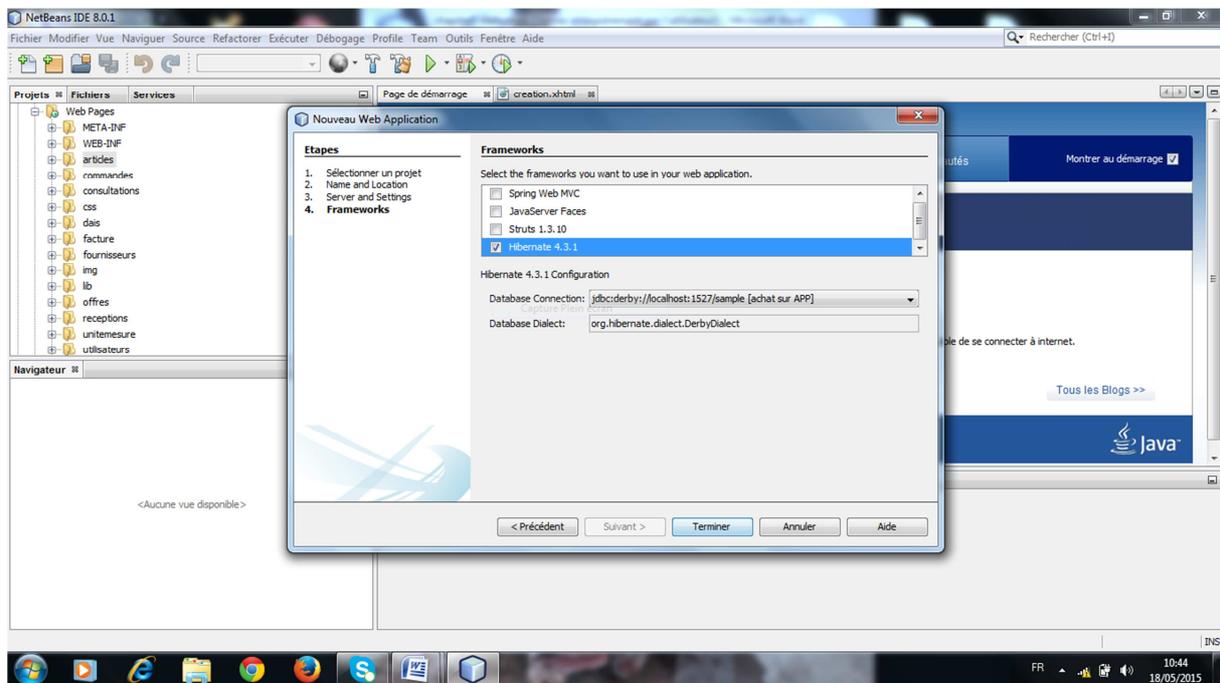


Figure V.27 : L'interface de Hibernate.

❖ EasyPHP : [26]

EasyPHP installe et configure automatiquement un environnement de travail complet qui permet de mettre en œuvre toute la puissance et la souplesse qu'offrent le langage dynamique PHP et son support efficace des bases de données MySQL. EasyPHP regroupe un serveur Apache, un serveur de base de données MySQL, Le langage PHP ainsi que des outils facilitant le développement de sites dynamiques ou d'applications.

La figure suivante montre une copie d'écran de la page d'accueil d'EsayPHP

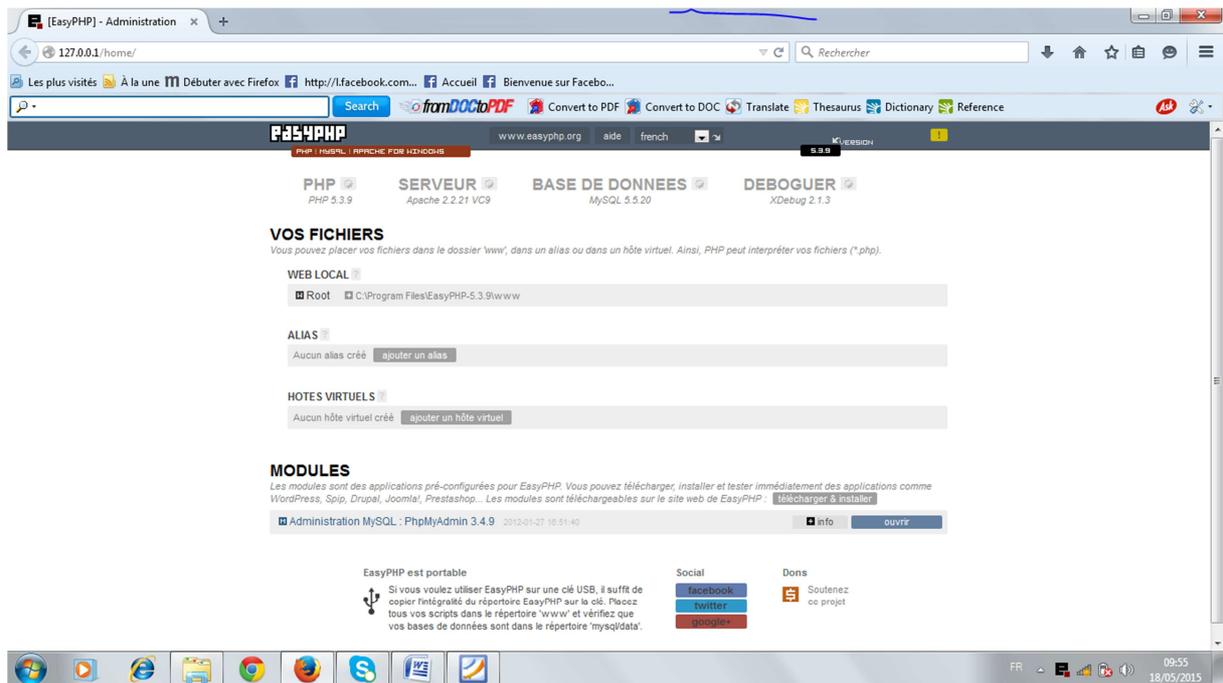


Figure V.28 : L'interface EasyPHP .

❖ PhpMyAdmin :

PhpMyAdmin est une application web permettant de gérer un serveur de bases de données MySQL. Cette interface est écrite en PHP et s'appuie sur le serveur Apache.

La figure suivante montre une copie d'écran de la page d'accueil de PhpMyAdmin.



Figure V.29: L'interface PhpMyAdmin.

❖ Serveur Apache Tomcat:

Apache Tomcat est une implémentation open source d'un conteneur web qui permet donc d'exécuter des applications web reposant sur les technologies servlets et JSP.

Tomcat est diffusé en open source sous une licence Apache. C'est aussi l'implémentation de référence des spécifications servlets jusqu'à la version 2.4 et JSP jusqu'à la version 2.0 implémentées dans les différentes versions de Tomcat.

❖ Netbeans :[27]

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme.

L'IDE Netbeans s'enrichit à l'aide de plugins.

Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web, etc).

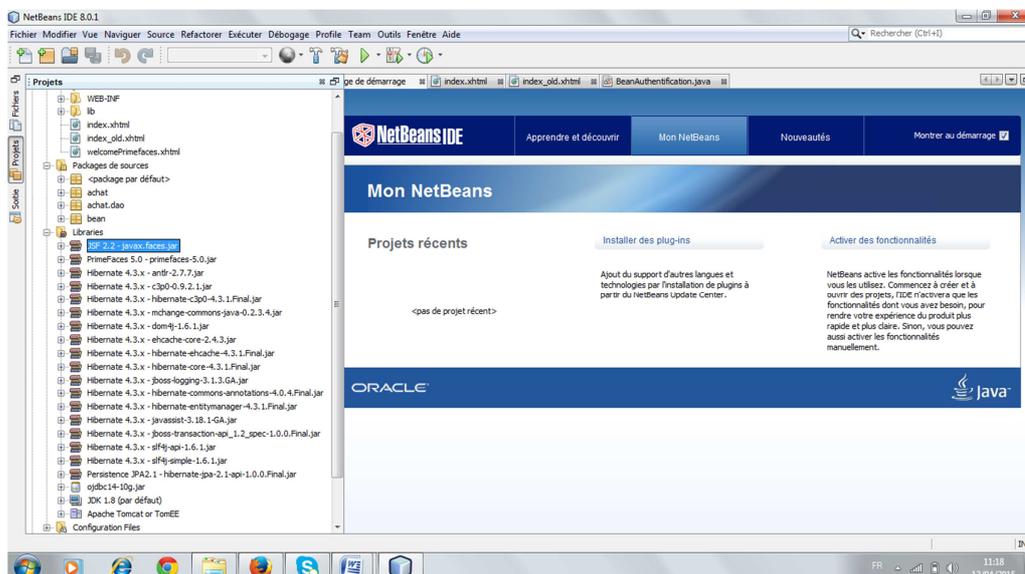


Figure V.30: L'interface de Netbeans .

III. Les langages de programmation :

❖ Java

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems. Mais c'est également un environnement d'exécution.

Java peut être séparée en deux parties. D'une part, le programme écrit en langage Java et d'autre part, une machine virtuelle (JVM) qui va se charger de l'exécution du programme Java.

C'est cette plateforme qui garantit la portabilité de Java. Il suffit qu'un système ait une machine virtuelle Java pour que tout programme écrit en ce langage puisse fonctionner.

❖ Java EE

Java Enterprise Edition, ou **Java EE** (anciennement **J2EE**), est une spécification pour la technique Java de Sun plus particulièrement destinée aux applications d'entreprise. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au *Framework* Java standard (JSE, *Java Standard Edition*) afin de faciliter la création d'applications réparties.

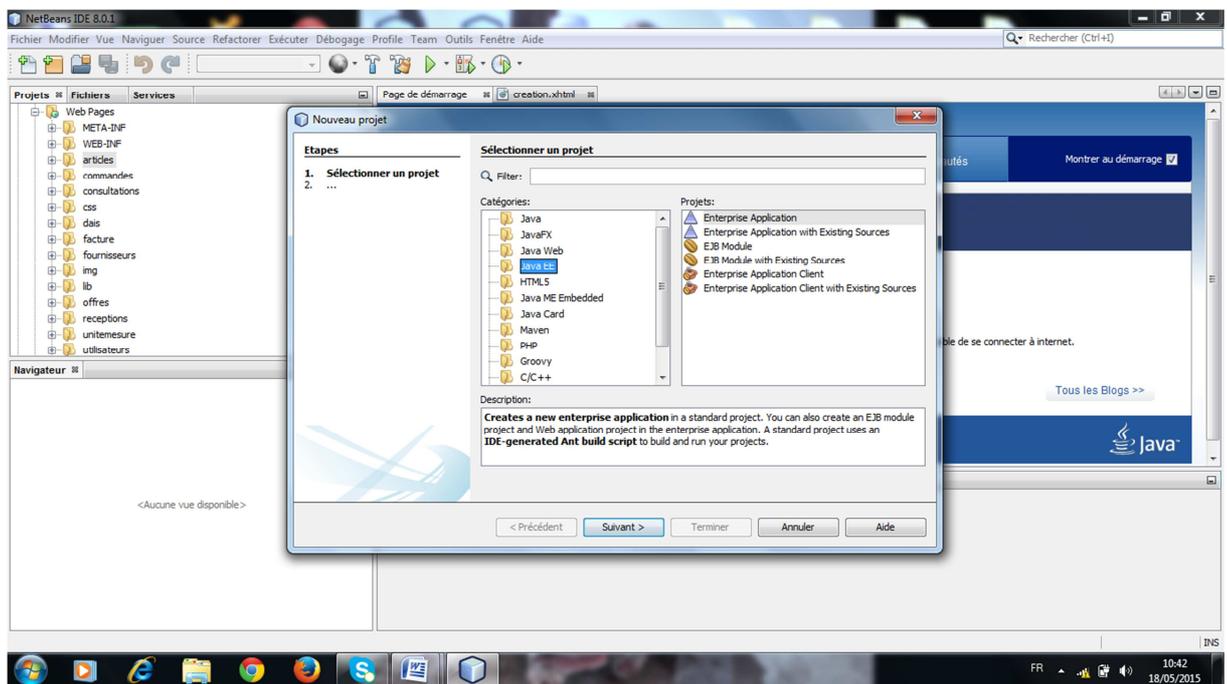


Figure V.31: L'interface de JEE.

❖ **XHTML** :(Extensible HypertextMarkupLanguage)

XHTML est un langage balisé servant à l'écriture de pages du World Wide Web. XHTML est le successeur de HTML (de l'anglais HyperText Markup Language), XHTML respectant la syntaxe définie par XML, plus récente et plus simple que la syntaxe définie par SGML respectée par HTML.

IV. Conception de la base de données :

Dans les étapes précédentes nous avons modélisés notre système avec les différents diagrammes (cas d'utilisation, séquence, activité, classe global), nous intéressons aux données nécessaires pour le fonctionnement de l'application.

Grace aux diagrammes de classes global nous pouvons obtenir la liste des tables suivantes :

❖ **Table Article :**

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_ARTICLE	Identifiant d'un Article	DECIMAL(19,2)	Primaire
ID_UNITE_MESURE	Identifiant d'une unite de mesure	INT (11)	Etrangère
DESIGNATION	Désignation	VARCHAR(50)	
CUMP	Coutd'achat	DECIMAL (8,4)	
DATE_CREATION	Date de création	DATA	
STOCK_MIN	Stock minimum	DECIMAL (22,0)	
STOCK_ALARME	Stock Alarme	DECIMAL (22,0)	
DATE_PEREMPTION	Date de peremption	DATA	
QTE_STOCK	Quantitéen stock	DECIMAL (22,0)	

❖ Table COMMANDE :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_COMMANDE	Identifiant de la commande	DECIMAL (22,0)	Primaire
ID_OFFRE	Identifiant de l'offre	DECIMAL (22,0)	Etrangère
NUMERO_BC	Numéro de bon de commande	DECIMAL (8,4)	
NUMERO_DAI	Numéro de la DAI	VARCHAR(50)	
NOM_UNITE	Non de l'unité	VARCHAR(50)	
NOM_FOURNISSEUR	Non de fournisseur	VARCHAR(50)	
DESIGNATION_ARTICLE	Désignation de l'article	VARCHAR(50)	
QUANTITE_ARTICLE	Quantité de l'article	VARCHAR(50)	
MONTANT_HT	Montant hors taxe	DECIMAL (8,4)	
MONTANT_TTC	Montant Tous taxe compris	DECIMAL (8,4)	
ETAT_COMMANDE	Etat de la commande	VARCHAR2(5)	

❖ Table CONSULTATION :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_CONSULTATION	Identifiant de la consultation	DECIMAL(22,0)	Primaire
DATE_CONSULTATION	Datede la consultation	DATE	
NUMERO_DAI	Numero de la DAI	DECIMAL(22,0)	
ETAT_CONS	Etat de consultation	VARCHAR2(5)	

❖ Table Demande d'achat interne « DAI » :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_DAI	Identifiant de la demande d'achat interne	DECIMAL(22,0)	Primaire
DATE_EMISSION	Date d'émission	DATE	
ETAT_DAI	Etat de la demande d'achat interne	VARCHAR2(30)	
NOM_UNITE	Nom de l'unité	VARCHAR2(30)	
NOM_ACHETEUR	Nom de l'acheteur	VARCHAR2(30)	
NUMERO_DAI	Numéro de la demande interne	DECIMAL(22,0)	
DESIGNATION	Désignation de l'article	VARCHAR2(30)	
QUANTITE_DEMANDE	Quantité demande	DECIMAL(22,0)	

❖ Table DAI CONSULTATION :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_CONSULTATION	Identifiant de la consultation	DECIMAL(22,0)	Primaire
ID_DAI	Identifiant de la demande d'achat interne	DECIMAL(22,0)	Etrangère

❖ Table **DETAIL_COMMANDE** :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_COMMANDE	Identifiant de la commande	DECIMAL(22,0)	Primaire
ID_ARTICLE	Identifiant de l'article	DECIMAL(22,0)	Etrangère
QTE_COMMANDEE	Quantité de la commande	DECIMAL(22,0)	

❖ Table **DETAIL_CONSULTATION** :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_ARTICLE	Identifiant de l'article	DECIMAL(22,0)	Primaire
ID_CONSULTATION	Identifiant de la consultation	DECIMAL(22,0)	Etrangère

❖ Table **DETAIL_DAI** :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_ARTICLE	Identifiant de l'article	DECIMAL(22,0)	Primaire
ID_DAI	Identifiant de la demande d'achat interne	DECIMAL(22,0)	Etrangère

❖ Table **DETAIL_RECEPTION** :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_RECEPTION	Identifiant de la reception	DECIMAL(22,0)	Primaire
ID_ARTICLE	Identifiant de l'article	DECIMAL(22,0)	Etrangère
QTE	Quantité	DECIMAL(22,0)	
UNITE_ACHAT	Unité d'achat	VARCHAR(50)	

❖ Table FACTURE :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_FACTURE	Identifiant de la facture	DECIMAL(22,0)	Primaire
ID_COMMANDE	Identifiant de la commande	DECIMAL(22,0)	Etrangère
DESIGNATION_ARTICLE	Designation de article	VARCHAR(50)	
QUANTITE_DEMANDE	Quantite demande	VARCHAR(50)	
DATE_FACTURE	Date de creation de la facture	DATE	
MONTANT_HT	Montant hors tax	DECIMAL (8,4)	
MONTANT_TTC	Montant tous taxe compris	DECIMAL (8,4)	
TYPE_FACTURE	Type de la facture	VARCHAR2(30)	
ETAT_FACTURE	Etat de la facture	VARCHAR2(5)	
MODE_REGLEMENT	Mode de règlement	VARCHAR2(5)	
MONTANT_LETTRE	Montant en lettre	VARCHAR2(100)	

❖ Table FOURNISSEUR_CONSULTATION :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_CONSULTATION	Identifiant de la consultation	DECIMAL(22,0)	Primaire
ID_FOURNISSEUR	Identifiant de fournisseur	DECIMAL(22,0)	Etrangère

❖ Table FOURNISSEUR :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_FOURNISSEUR	Identifiant de fournisseur	DECIMAL(22,0)	Primaire
NOM_FOUTNISSEUR	Nom de fournisseur	VARCHAR2(50)	
RAISON_SOCIALE	Raison sociale	VARCHAR2(70)	
ADRESSE	Adresse de fournisseur	VARCHAR2(70)	
TELEPHONE	Telephone de fournisseur	VARCHAR2(15)	
FAX	Fax de fournisseur	VARCHAR2(15)	
TYPE_FOURNISSEUR	Type de fournisseur	VARCHAR2(30)	
DATE_CREATION	Date de creation	DATE	
DOMAINE_ACTIVITE	Domaine d'activité	VARCHAR2(70)	

❖ Table OFFRES :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_OFFRE	Identifiantd'offre	DECIMAL(22,0)	Primaire
ID_CONSULTATION	Identifiant de la consultation	DECIMAL(22,0)	Etrangère
NOM_FOURNISSEUR	Nom de fournisseur	VARCHAR2(70)	
DESIGNATION_ARTICLE	Designation de article	VARCHAR2(70)	
QUANTITE_DEMANDE	Quantitedemande	VARCHAR2(70)	
DATE_OFFRE	Date d'offre	DATE	
VALIDITE	Validited'offre	DECIMAL(22,0)	
MONTANT_HT	Montant hors taxe	DECIMAL(8,2)	
MONTANT_TTC	Montanttoustaxecompris	DECIMAL (8,2)	
ETAT_OFFRE	Etatd'offre	VARCHAR2(5)	

❖ Table RECEPTION :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_RECEPTION	Identifiant de la reception	DECIMAL(22,0)	Primaire
ID_FACTURE	Identifiant facture	DECIMAL(22,0)	Etrangère
DATE_RECEPTION	Date de reception	DATE	
NUMERO_ORDRE	Numérod'ordre	VARCHAR2(6)	
ETAT_RECEPTION	Etat de réception	VARCHAR2(5)	
NUMERO_BC	Numero de bon de commande	DECIMAL(22,0)	
DESIGNATION_ARTICLE	Désignation de l'article	VARCHAR2(50)	
NOM_FOURNISSEUR	Nom de fournisseur	VARCHAR2(50)	

❖ Table UNITE_MESURE :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_UNITE_MESURE	Identifiant de l'unité de mesure	DECIMAL(22,0)	Primaire
DESIGNATION	designation	VARCHAR2(70)	
CODE	Code	VARCHAR2(15)	

❖ Table UTILISATEURS :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
ID_utilisateur	Identifiant de l'utilisateur	DECIMAL(22,0)	Primaire
LOGIN	Nom de l'utilisateur	VARCHAR2(8)	
PW	Mot de passe utilisateur	VARCHAR2(8)	
PROFIL	Profil utilisateur	VARCHAR2(45)	

V. Présentation de quelques interfaces de l'application :

❖ La page d'authentification au système :

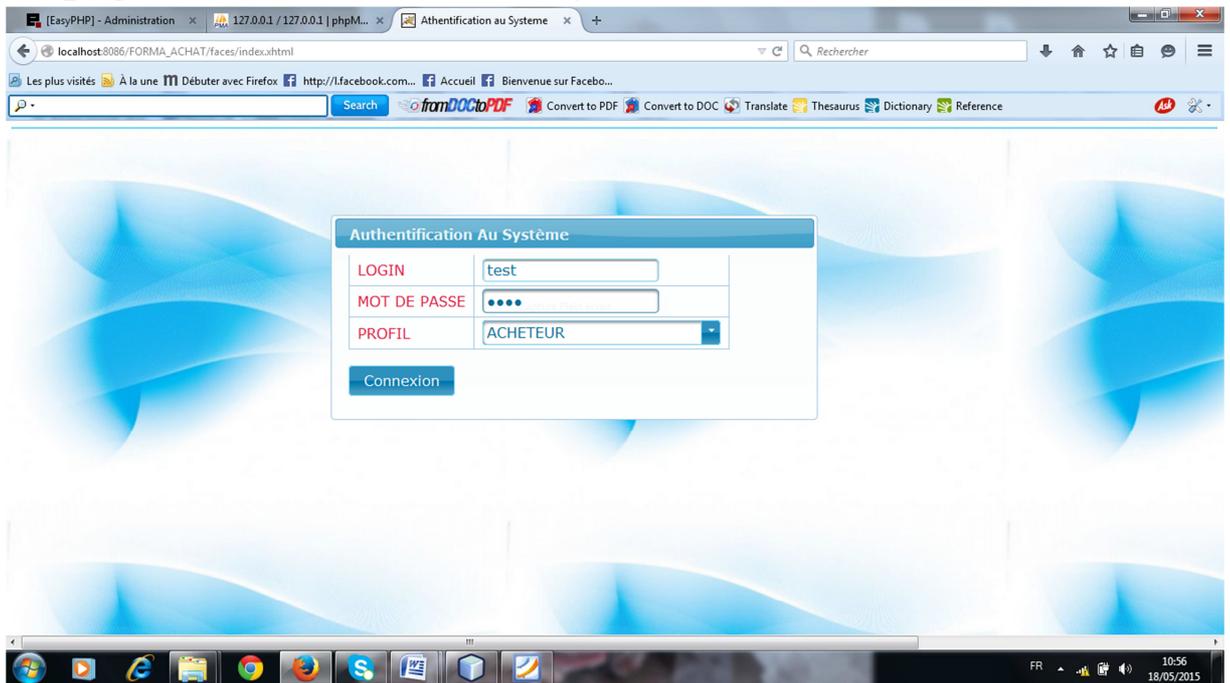


Figure V.32 : Page d'authentification au système.

❖ Page d'accueil de l'application :



Figure V.33 : page d'accueil de l'application.

❖ Page de suppression d'une demande d'achat interne (DAI) :

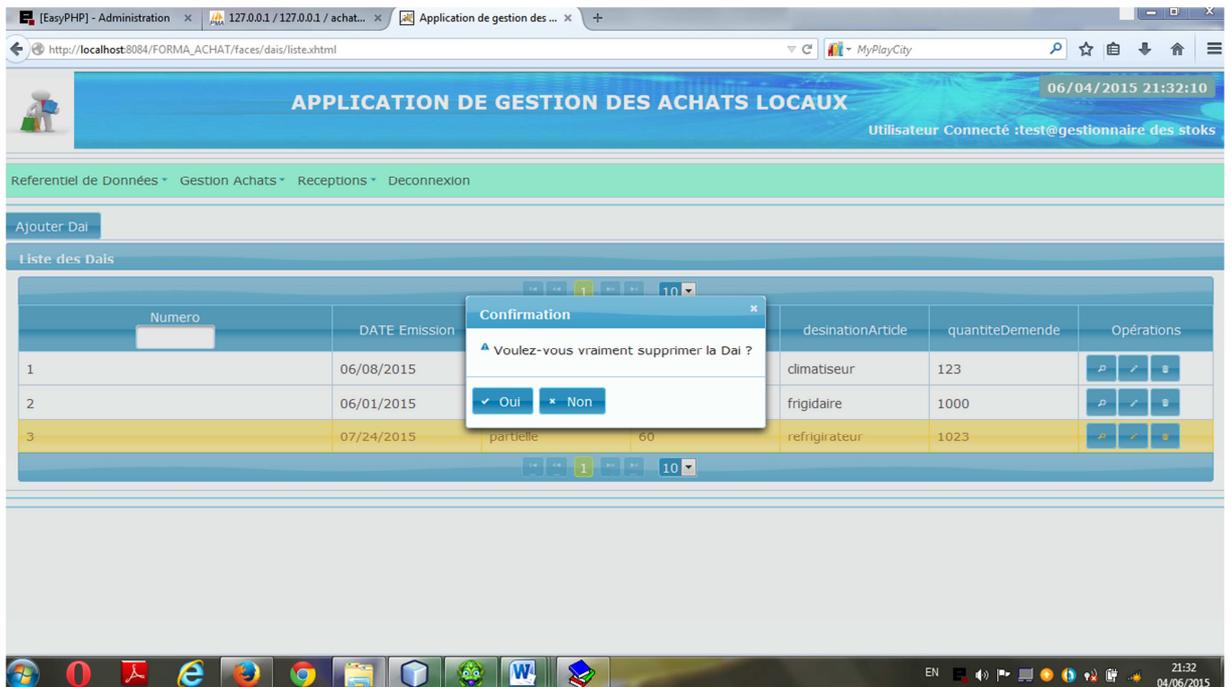


Figure V.34 : Page de Suppression d'une demande d'achat interne (DAI).

❖ Page Ajout d'un Article :



Figure V.35: Page Ajout d'un Article.

❖ Page de modification d'un bon de réception :

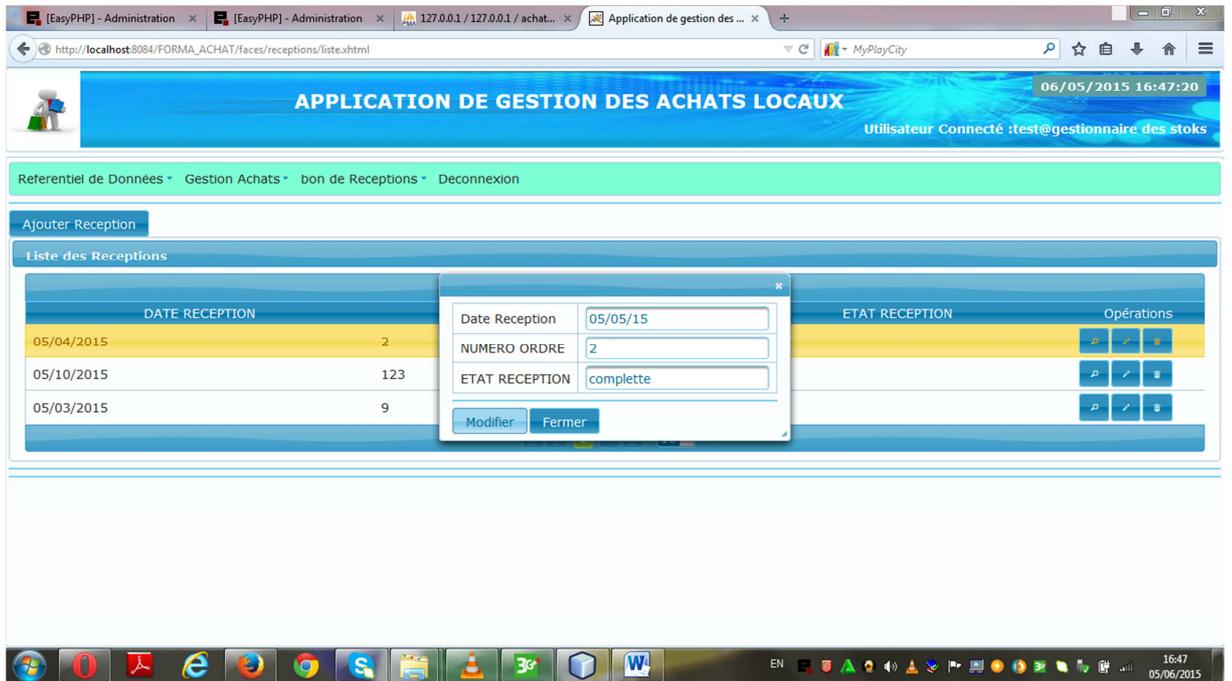


Figure V.36 : Page de modification d'un bon de réception.

❖ Page de l'impression d'un bon de commande :

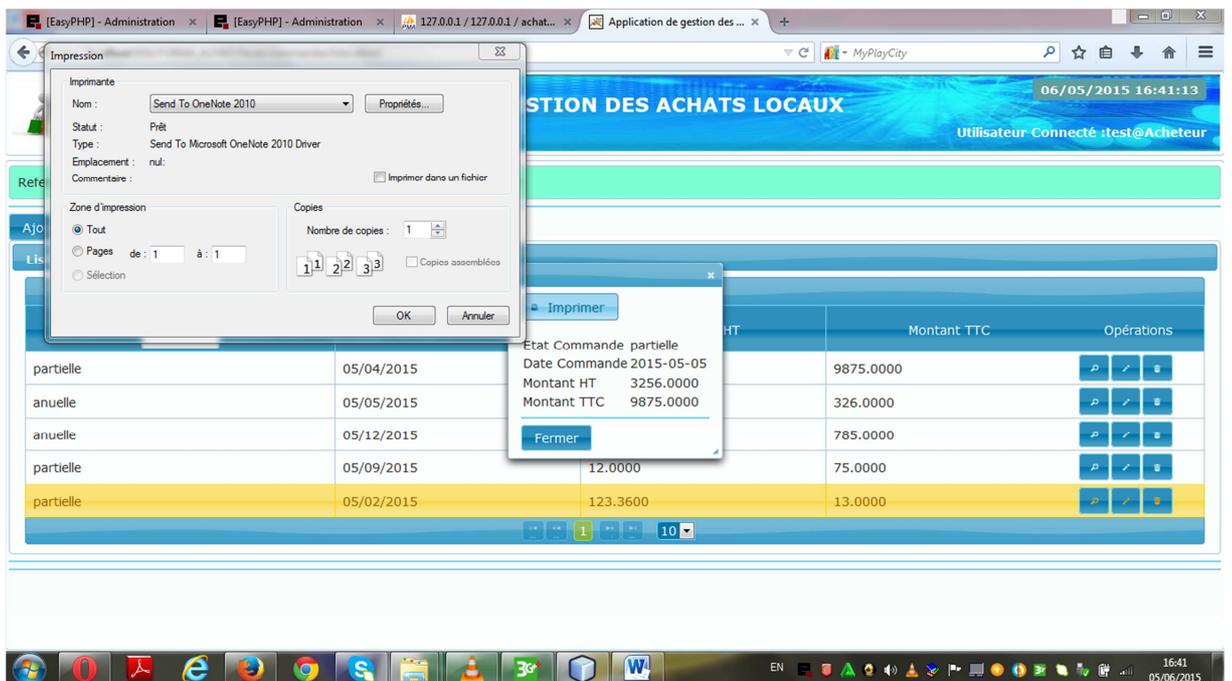


Figure V.37 : Page de l'impression d'un bon de commande.

❖ Page de consultation des Fournisseur :



Figure V.38 : Page de consultation des Fournisseur.

❖ Page d’Ajout d’un utilisateur :

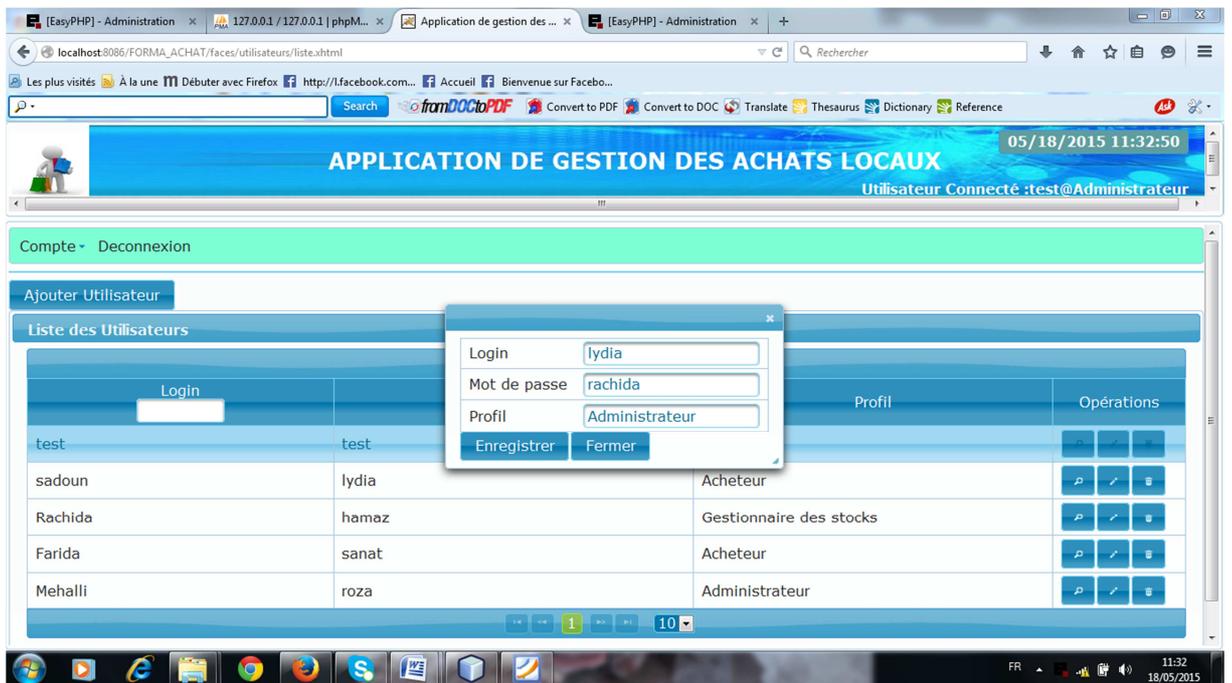


Figure V.39: Page d’Ajout d’un utilisateur.

VI. Conclusion :

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement de développement et la démarche suivie pour la réalisation.

CONCLUSION GÉNÉRALE

Le contact avec le monde du travail, permet de voir l'intérêt et l'apport de l'outil informatique en termes de rapidité d'exécution des tâches administratives récurrentes, qui sont la plupart du temps réalisées manuellement or qu'un ordinateur peut le faire en une fraction de seconde, l'automatisation permet aussi d'offrir un espace de travail convivial.

Le stage que nous avons effectué au sein de l'entreprise nationale des industries de l'électroménager (ENIEM) ainsi que l'étude et le développement de notre projet, nous a permis de mettre en pratique les notions théoriques acquises durant notre cursus d'études universitaires et d'acquérir de nouvelles connaissances (conception avec UML, PHP et JEE) et de mettre entre les mains des utilisateurs du service achat de l'unité froid de l'entreprise ENIEM un outil qui leur facilitera le suivi des achats locaux.

Notre application offre plusieurs services ; et cela en permettant l'accès rapide aux données et ainsi faciliter la collaboration et la communication entre les différents acteurs de l'entreprise.

Enfin, nous souhaitons que ce système réponde efficacement aux besoins de nos champs d'étude et qu'il soit un support de documentation supplémentaire pour les futures stagiaires.

Webliographie /Bibliographie

- [1] : <http://www.grappa.univ-lille3.fr/polys/systreseaux/ch02s06.html>
- [2] : <http://blog.profession-web.ch/index.php/544-avantages-et-inconvenients-du-web-20>
- [3] : Mémoire promotion 2012-2013 réalisé par M^{elle} SEDKAOUI Samia M^{elle}: SADOUN Lydia Dirigé par : M^r HABETE
- [4] : <http://www.lifl.fr/icar/Chapters/J2EE/j2ee.html>
- [5] : <http://ligmembres.imag.fr/plumejeaud/NFE107fichesLecture/J2EE%2520vs%2520NET.df>
- [6] : http://www-igm.univ-mlv.fr/~dr/XPOSE2003/alexandrebole/ejb_2.html
- [7] : Mémoire promotion 2013-2014 réalisé par Mr DAHMANI Toufik & M^{elle} OULDSAID Francesca Dirigé par : M^r HABETE
- [8] : www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm
- [9] : www.eniem.com.dz
- [10] : <http://www.inmediaveritas.com/comparatif-framework/>
- [11] : <http://fr.wikipedia.org/wiki/Framework>
- [12]: <http://www.hibernate.org>
<http://www.alaide.com>
<http://www.developpez.com>
- [13]: jsftutorials.net/

[Présentation sur Developpez.com](http://www.developpez.com)
- [14] : <http://blog.primefaces.org/?p=3510>

<http://primefaces.org/license.html>

<http://jaxenter.com/primefaces-2-2rc1-upgrades-jquery-32479.html>

<https://www.packtpub.com/all?search=primefaces>
- [15] : Mémoire promotion 2013-2014 réalisé par Mr
- [16] : Mémoire promotion 2013-2014 réalisé par
- [17] : Mémoire promotion 2013-2014 réalisé par
- [18] : <http://www.inmediaveritas.com/comparatif-framework/>

[19] : Mémoire promotion 2012-2013 réalisé par M^{elle} SEDKAOUI Samia M^{elle}: SADOUN Lydia Dirigé par : M^r HABETE

[20] : Mémoire promotion 2012-2013 réalisé par M^{elle} SEDKAOUI Samia M^{elle}: SADOUN Lydia Dirigé par : M^r HABETE

[21] : Mémoire promotion 2012-2013 réalisé par M^{elle} SEDKAOUI Samia M^{elle}: SADOUN Lydia Dirigé par : M^r HABETE

[22] : Mémoire promotion 2012-2013 réalisé par M^{elle} AMRANI Katia et M^{elle} TAVARES GOMES Ivandra Marise Dirigé par : M^r Si-Mohammed Malik

[23] : [http : //www-lipn.univ-paris13.fr/ audibert /pages/enseignement/cours.htm](http://www-lipn.univ-paris13.fr/audibert/pages/enseignement/cours.htm)

[24] : [http : //www-lipn.univ-paris13.fr/ audibert/pages/enseignement/cours.htm](http://www-lipn.univ-paris13.fr/audibert/pages/enseignement/cours.htm)

[25] : <http://www.hibernate.org/>

[26] : Mémoire promotion 2012-2013 réalisé par M^{elle} SEDKAOUI Samia M^{elle}: SADOUN Lydia Dirigé par : M^r HABETE

[27] : <http://doc.ubuntu-fr.org/netbeans>