

*République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi Ouzou
Faculté de Génie Electrique et D'informatique
Département D'informatique*

MEMOIRE



DE FIN D'ETUDE

En vue de l'obtention du diplôme de Master en Informatique.

Option : Conduite de Projet Informatique

Thème

*Etude des Framework J2EE
Conception et Réalisation d'une Application
Client /Serveur 3tiers sous Réseaux.
Cas : NAFTAL de Tizi Ouzou*

Dirigé par :

Mr. DIB.A

Réalisé par :

Melle Tiaouinine Zehor

Mr. Salhi Abdelghani

Promotion 2013/2014

Remerciements

Quelle meilleure opportunité que cette thèse qui marque l'achèvement des études universitaires, pour exprimer les plus vifs remerciements à notre promoteur Monsieur DIB pour nous avoir encadrées et guidées tout au long de notre projet.

Nous tenons à remercier chaleureusement le personnel du Service Moyens Généreux de NAFTA.

Nos plus vifs remerciements vont aussi à tous les membres de jury pour avoir accepté d'honorer par leur jugement notre travail.

Sans oublier nos parents qui ont sacrifié jusqu'aujourd'hui, aussi leurs encouragements incessants tout le long de notre parcours.

Ainsi que toute personne qui, de près ou de loin, a contribué à la réalisation de ce projet.

Sommaire

Introduction Générale.....	1
-----------------------------------	----------

Chapitre 01 : Généralité sur les réseaux

I. Introduction :.....	2
II. Les réseaux informatiques :.....	2
1. Définition d'un réseau informatique :.....	2
2. Objectif des réseaux :.....	2
3. Classification des réseaux informatiques :.....	3
4. Architecture des réseaux :.....	4
5. Définition de l'architecture client /serveur :.....	7
5.1. Caractéristiques du modèle client/serveur :.....	8
5.2. Technique du dialogue Client /Serveur :.....	9
5.2.1. Notions de base et fonctionnement de client/serveur :.....	9
5.3. Architecture client/serveur multi niveaux :.....	9
5.3.1 Architecture client/serveur à deux niveaux :.....	9
5.3.2 Architecture trois niveaux :.....	10
5.3.3. Architecture en couche :.....	11
5.4. Les types de serveur :.....	11
5.4.1. Serveur de fichiers :.....	11
5.4.2. Serveur de base de données :.....	11
5.4.3. Serveur de transaction :.....	11
5.4.4 .Serveur d'application Web :.....	12
5.4.5. Serveur groupware :.....	12
5.4.6 Serveur d'application objet :.....	12
5.5. Le fonctionnement du modèle client/serveur :.....	12
III. Conclusion :.....	12

Chapitre 02 : Présentation de l'organisme d'accueil

I. Historique de l'organisme d'accueil :.....	13
II. Problématique :.....	14
III. Présentation du district COM Tizi-Ouzou :.....	14
IV. Mission et activité de l'organisme d'accueil :.....	15
IV.1. Organigramme général de l'organisme d'accueil :.....	16

IV.2. L'organigramme de champs d'étude :	17
V. Description de l'organigramme général :	18
V.1. Présentation des différents départements :	18
V.2. Présentation du Centre Multi (CMP) :	21
V.3. Présentation du centre de stockage et de distribution (CSD) :	22
VI. Situation informatique de l'organisme d'accueil (District COM Tizi-Ouzou) :	22
VII. Etude des postes :	27
1. Introduction :	27
2. Liste des postes :	27
VIII. Etude des documents :	29
1. Introduction :	29
2. Liste des documents :	29

Chapitre 03 : Etude des Framework

I-Introduction :	36
II-les différents types de Framework :	36
II-1- Wave Maker Framework:	37
II-1-1-Introduction:	37
II-1-2-Définition:	37
II-1-3-Evolution (développement):	38
II-1-4-Principe d'utilisation:	39
II-1-5-Cas d'utilisation et exemple d'application :	40
II-1-6-Avantages :	41
II-2-Qt Jambi Framework:	41
II-1-1-Introduction:	41
II-1-2-Définition:	41
II-1-3-Evolution (développement):	42
II-1-4-Principe d'utilisation:	42
II-1-5-Cas d'utilisation et exemple d'application :	43
II-1-6-Avantages :	45
II-3- Sofi Framework:	46
II-1-1-Introduction:	46
II-1-2-Définition:	46
II-1-3-Evolution (développement):	46
II-1-4-Principe d'utilisation:	47
II-1-5-Cas d'utilisation et exemple d'application :	48
II-1-6-Avantages:	49
II-5- Spring Framework:	50
II-1-1-Introduction:	50
II-1-2-Définition:	50

II-1-3-Evolution (développement):.....	51
II-1-4-Principe d'utilisation:.....	51
II-1-5-Cas d'utilisation et exemple d'application :.....	52
II-1-6-Avantages :.....	52
III-La comparaison entre les différents type de Framework :.....	53
III-1-Information sur chaque Framework :.....	53
III-2-comparaison des fonctionnalités :.....	54
IV-Choix du Framework pour j2ee/java :.....	55
V-Conclusion :.....	56

Chapitre 04 : Analyse et Conception

I-Introduction :	57
II. Analyse :.....	57
II.1. Définition des besoins :.....	57
II.2. Identification des acteurs :.....	57
II.3. Identification des espaces :.....	58
II.4. Spécification des tâches :.....	58
II.5. Spécification des scénarios :.....	58
II.5.1. Définition d'un scénario :.....	58
II.6. Diagramme des cas d'utilisation général :.....	61
II.7. Diagrammes de quelques cas d'utilisations :.....	62
II.7.1. Diagramme détaillé du cas d'utilisation «Géré les Bon de Commande » :.....	62
II.7.2. Diagramme détaillé du cas d'utilisation «Géré les Bon d'entré Magasin » :.....	63
II.7.3. Diagramme détaillé du cas d'utilisation «Modifier le compte» :.....	63
II.7.4. Diagramme détaillé du cas d'utilisation «Géré les comptes utilisateur» :.....	64
III. Conception :.....	64
III.1. La démarche de conception de l'application :.....	64
III.2. Elaboration des diagrammes de séquences :.....	65
III.3. Elaboration des diagrammes de classes :.....	65
III.4. Elaboration des diagrammes d'activités :.....	70
III.5. Conception de la base de données :.....	74
IV. Conclusion :.....	79

Chapitre 05 : Réalisation

I. Introduction	80
II. Les outils de développement	80
II.1.Spring Framework :.....	80
II.3.Serveur de base de données PHP My admin	81
II.4.Serveur Apache Tomcat 8.0.8:.....	82
III. Les langages de programmation :	82

III.1.J2EE :	82
III.2.Netbeans :	83
IV. Présentation de quelques interfaces de l'application :	85
IV.1. La page d'accueil :	85
IV.2. La page espace administrateur :	86
IV.3. La page espace chef de section :	87
IV.4. La page espace magasinier :	88
IV.5. La page ajouter un compte utilisateur:	89
IV.6. La page géré les bons de commande :	90
V.	
Conclusion :	91
➤ Conclusion Générale.....	92
➤ Annexe A.....	93
➤ Annexe B.....	104
➤ La Bibliographie.....	112

Introduction Générale

La technologie de l'information aujourd'hui, a envahi tous les domaines de notre vie quotidienne, surtout aux seins des entreprises économiques, qui doivent s'armer de la plus récente des technologies, Car soumise à une rude et virulente concurrence. La gestion et la maîtrise de l'information sont devenues des préoccupations de premier ordre.

La mise en œuvre de cette technologie moderne de traitement de l'information s'avère nécessaire, pour assurer une gestion plus fiable, plus rigoureuse, moins fastidieuse, ainsi pour minimiser le risque d'erreurs et d'assurer la disponibilité de l'information à toute éventuelle demande.

Dans toute organisation, la gestion des ressources humaines est une partie déterminante, sachant que de bons employés et de bons équipements contribuent en grande partie à sa réussite.

Actuellement l'entreprise économique qui veut marquer sa présence sur le marché, doit admettre que l'information n'est pas une fin en soi et qu'elle doit être menée en harmonie avec d'autres actions liées pour l'essentiel à la formation de son potentiel humain.

Le travail qui nous a été confié par le District Commercialisation de Ouad Aissi consiste à mettre en œuvre un système d'information pour la gestion des ressources humaines. Le projet porte sur la mise en œuvre d'un système d'information qui procurera un environnement convivial aux acteurs afin de répondre aux contraintes de fiabilité, efficacité et surtout de disponibilité.

Pour la réalisation de ce système d'information, on a opté pour l'utilisation de l'architecture *Client/serveur 3tiers* sous *Réseaux* qui va nous permettre de réaliser une application déployée sous un réseau LAN.

Notre travail se décompose en 5 chapitres qui sont :

Chapitre 1 : Généralité sur les Réseaux.

Chapitre2 : Présentation de l'organisme d'accueil.

Chapitre3 : Etude des Framework.

Chapitre4 : Analyse et conception.

Chapitre 5: Réalisation.

Et se termine par une conclusion générale et une annexe portant sur le langage UML.

.

I. Introduction :

Un réseau est un ensemble de moyens matériels et logiciels, géographiquement dispersés, destinés à offrir un service comme les réseaux téléphoniques ou à assurer le transport des données ; Les réseaux devront véhiculer toute sorte d'information avec des contraintes temporelles plus au moins fortes avec des volumes extrêmement variés.

II. Les réseaux informatiques : [9] [10] [24] [26]

1. Définition d'un réseau informatique :

C'est une interconnexion d'ordinateurs, terminaux, serveurs, qui permettent un échange d'information entre ces derniers. Cette connexion est réalisée par des câbles électriques, fibres optiques et satellites. Dans un premier temps, ces communications étaient destinées au transport de données informatiques, aujourd'hui les réseaux tendent à intégrer également la parole et la vidéo.

2. Objectif des réseaux :

Les réseaux permettent :

- Le partage de données (fichiers, BDD,...).
- Partage de périphériques.
- Communication entre utilisateurs (messagerie électronique).

3. Classification des réseaux informatiques :

La classification des réseaux se distingue en deux :

a. Selon l'envergure:

• les réseaux personnels ou PAN (Personal Area Network) :

Désignent un réseau restreint d'équipements informatiques habituellement utilisés dans le cadre d'une utilisation personnelle. Les bus utilisés les plus courants sont l'USB, les technologies sans fil telles que Bluetooth ou IR (infra rouge).

• Les réseaux locaux ou LAN (Local Area Network) :

Sont des réseaux informatiques à une échelle géographique relativement restreinte, par exemple une salle informatique, une habitation particulière, un bâtiment ou un site d'entreprise.

Les réseaux métropolitains ou MAN (Metropolitan Area Network) :

Désignent des réseaux étendus d'ordinateurs habituellement utilisés dans les campus ou dans les villes. Ces réseaux utilisent généralement des fibres optiques. Par exemple, une université peut avoir un MAN qui lie plusieurs réseaux locaux.

• Les réseaux étendus ou WAN (Wide Area Network) :

Sont des réseaux informatiques couvrant une grande zone géographique, typiquement à l'échelle d'un pays, d'un continent, voire de la planète entière. Le plus grand WAN est le réseau Internet.

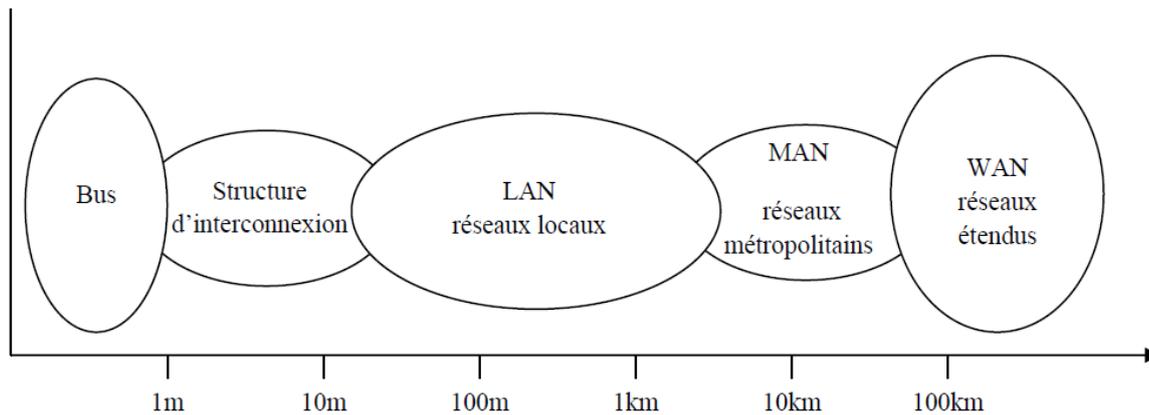


Figure 01 : Classification des réseaux informatiques selon l'envergure

b. Selon la topologie:

On trouve les topologies suivantes :

• **Réseau en étoile :**

Les stations du réseau communiquent bi directionnellement avec le serveur. C'est une liaison point à point entre le serveur et chaque machine. Deux stations peuvent échanger des données à condition d'en passer par le serveur.

• **Réseau en bus :**

Chaque station, serveurs compris, communique en permanence de manière bidirectionnelle sur un support de transmission unique. Les liaisons ne sont pas point à point, c'est un réseau à diffusion. On peut également y pratiquer le multipoint car une machine peut adresser un même paquet de données à plusieurs stations simultanément. Dans cette configuration, il est important de fixer une politique d'accès au bus si on peut éviter que deux stations ne le fassent en même temps.

• **Réseau en anneau :**

Toutes les stations reçoivent toutes les trames d'information et les font passer à leurs voisines. Nous avons ici l'exemple de liaisons point à point monodirectionnelles entre chacune des machines. La circulation de l'information se fait en sens unique.

• **réseau maillé :**

Un réseau maillé est caractérisé par le fait que deux nœuds quelconques soient reliés l'un à l'autre. Ce type de réseau permet plus de souplesse et de fiabilité dans son utilisation, mais il est difficilement envisageable pour un grand nombre de nœuds. Généralement, le maillage n'est pas parfait, on dit que c'est un réseau partiellement maillé. Entre les nœuds. Nous avons ici des liaisons points à point entre machines.

Remarque :

Toute topologie adoptée doit faire au préalable l'objet d'une étude prenant en compte plusieurs facteurs : Nombre de stations à connecter, flux des données, coût, distance entre entités communicantes, évolution possible, résistance aux pannes et lignes de secours, administration ...etc.

4. Architecture des réseaux :

Le transport des données d'une extrémité à l'autre d'un réseau nécessite un support physique de communication. Cependant, pour que ces données arrivent correctement au destinataire, avec la qualité de service exigée, il faut une architecture logicielle. L'ensemble de protocoles nécessaires constitue une architecture. Un protocole est l'ensemble de règles qui définissent les modalités de fonctionnement d'une communication entre deux ordinateurs.

a. l'architecture OSI (Open System Inter connexion) :

Que l'on appelle open system Interconnexion ou Interconnexion des Systèmes ouverts, c'est une architecture qui forme le modèle de référence qui comporte sept couches.



Figure 02: *Les couches du modèle OSI.*

➤ **La couche physique :**

Elle est responsable de la manipulation des détails mécaniques et électriques de la transmission physique d'un flot de bits sur le canal de communication.

➤ **La couche de liaison :**

Sa tâche principale est de détecter et de corriger les erreurs de transmission, l'émetteur traite également les accusés de réception émis par le récepteur.

➤ **La couche réseau :**

Elle s'occupe du routage et du contrôle de congestion. L'information stockée au niveau de cette couche est structurée sous forme de paquets de données.

➤ **La couche transport :**

Elle est responsable du transfert de données entre les entités de sessions. Ce transport devant être transparent, c'est-à-dire indépendant de la succession des caractères et même des éléments binaires transportés.

➤ **La couche session :**

Elle permet à des utilisateurs de différentes machines d'établir des sessions entre eux. Parmi ses services : la gestion du dialogue, le transfert de fichiers, ...

➤ **La couche présentation :**

Elle gère la syntaxe et la sémantique des informations transmises (le codage, la compression, la cryptographie des données, ...).

➤ **La couche application :**

Elle est responsable de l'interaction directe avec les utilisateurs. Elle traite les protocoles de connexion à distance, le courrier électronique ainsi que la définition des terminaux virtuels.

b. L'architecture TCP/IP (Transmission Control Protocol / Internet protocol):

Dans les années 70, DOD (Department Of Defense), devant le foisonnement de machines utilisant des protocoles de communication différents et incompatibles, décide de définir sa propre architecture.

L'architecture TCP/IP est à la source du réseau Internet de nombreux réseaux privés. Cette architecture comme celle d'OSI est répartie en couches.

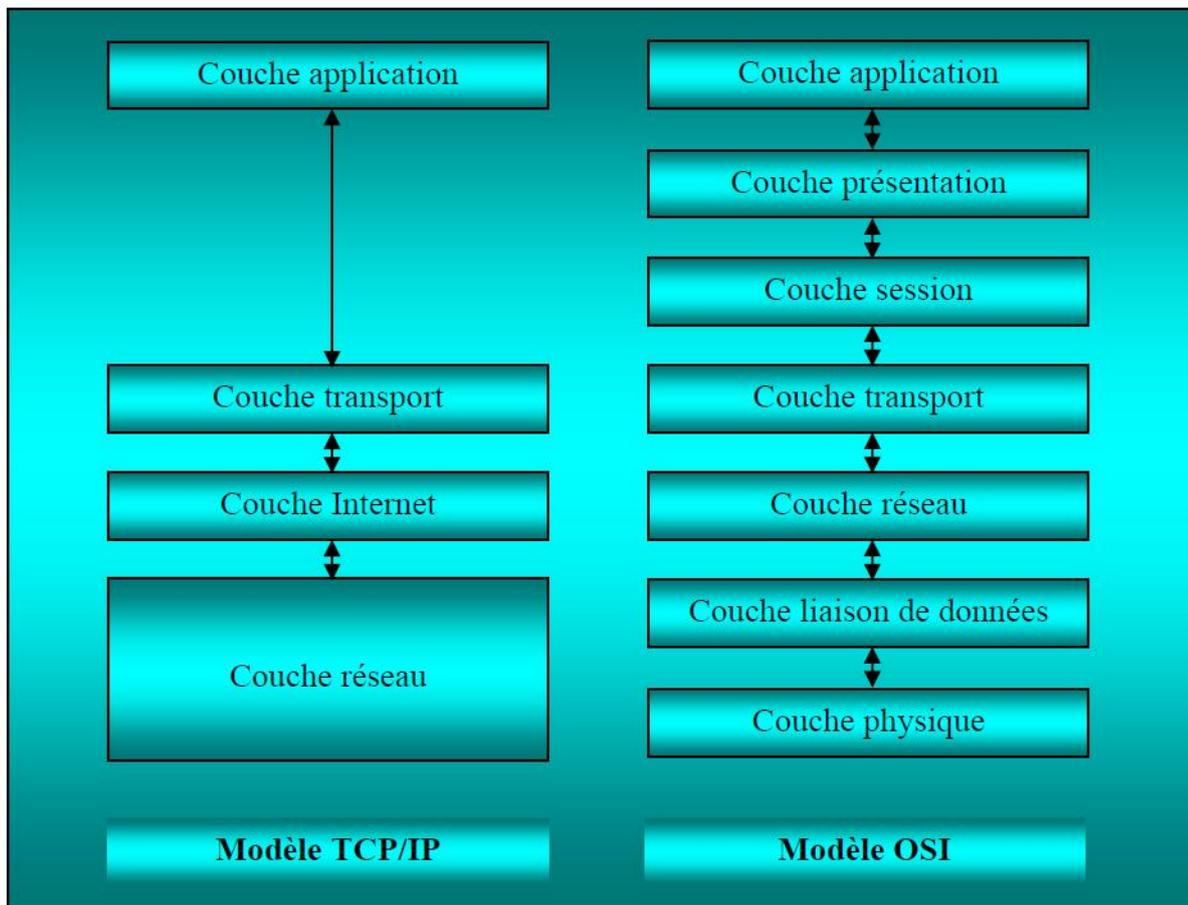


Figure 03: La pile de protocole TCP/IP avec sa correspondance OSI.

➤ **La couche application :**

Elle englobe les applications standards du réseau :

SMTP (Simple Mail Transport Protocol) : permet d'échanger du courrier entre deux serveurs de messagerie.

TELNET : Protocole permettant de se connecter sur une machine distante (serveur) en tant que utilisateur.

FTP (File Transfert Protocol) : Est un protocole permettant d'échanger des fichiers via Internet.

DNS (Domain Name Service) : Service de nom du domaine. Il permet de convertir le nom d'une machine en une adresse réseau (ou IP) et vice versa.

➤ **La couche transport :**

Elle assure l'acheminement des données sur le réseau local et les mécanismes permettant de connaître l'état de la transmission. Les protocoles de cette couche permettent d'envoyer des informations d'une machine à une autre.

TCP : Il assure le contrôle de données, orienté connexion.

UDP (User Datagram Protocol) : Archaïque et non orienté connexion, il n'assure aucun contrôle de transmission de données.

➤ **Couche Internet :**

La couche Internet est chargée de fournir le paquet de données. Elle contient cinq protocoles, les trois premiers sont les plus importants :

IP : gère les destinations des messages, adresse du destinataire.

ARP (Address Resolution Protocol) : gère les adresses des cartes réseau. Chaque carte a sa propre adresse d'identification codée sur 48 bits.

ICMP (Internet Control Message Protocol) : gère les informations relatives aux erreurs de transmission. Il ne corrige pas les erreurs, mais signale aux autres couches que le message contient des erreurs.

RARP (Reverse Address Resolution Protocol) : gère l'adresse IP pour les équipements, qui ne peuvent s'en procurer une, par lecture d'information dans un fichier de configuration. En effet, lorsque le PC démarre, la configuration réseau lit l'adresse IP qu'elle va utiliser. Ceci n'est pas possible dans certains équipements qui ne possèdent pas de disques durs (imprimantes, terminaux).

IGMP (Internet Group Management Protocol) : permet d'envoyer le même message à des machines faisant parti d'un groupe. Ce protocole permet également à ces machines de s'abonner ou de se désabonner d'un groupe. Ceci est utilisé par exemple dans la vidéo conférence à plusieurs machines, envoi de vidéos,...

5. Définition de l'architecture client /serveur :

Client/serveur désigne un mode de communication entre des programmes informatiques :

- a. Un programme tient le rôle de « serveur » : il est conçu de façon à pouvoir répondre à des requêtes extérieures.
- b. Les autres programmes « client » contacte le serveur pour lui soumettre des demandes, et attendre sa réponse.

Un serveur peut fournir un ou plusieurs services, et ceux, a un ou plusieurs client, le programme serveur et /ou les programmes client peuvent se trouver sur la même machine ou sur des machines différentes, reliés entre elles par réseau.

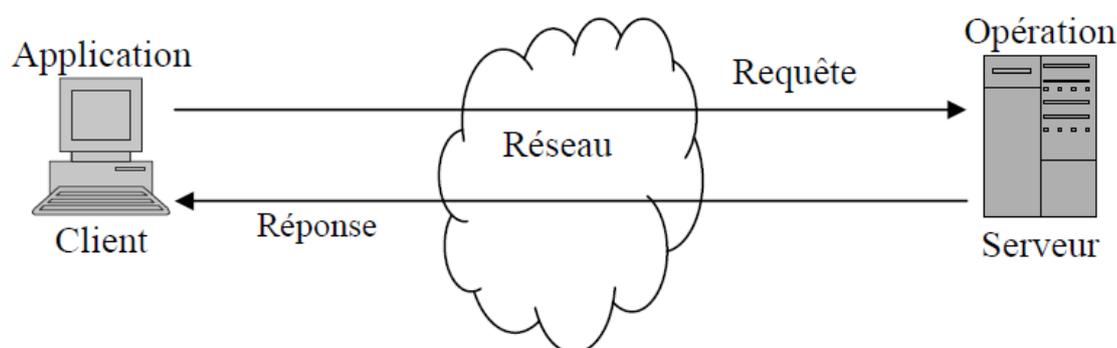


Figure04: Le dialogue client/serveur.

5.1. Caractéristiques du modèle client/serveur :

Bien que le client/serveur soit une forme d'informatique distribuée, il n'est pas supposé d'appeler système client/serveur que ceux qui partagent les caractéristiques suivantes :

- ✓ **Service** : Le modèle Client/serveur est bâti sur une relation entre des processus tournants sur des machines séparées. Le processus serveur est un fournisseur de services, le processus client est un consommateur de services. Le modèle établit ainsi une répartition claire des fonctions à partir de la notion de service.
- ✓ **Partage des ressources** : Un serveur peut traiter plusieurs clients en même temps et contrôler leurs accès aux ressources.
- ✓ **Asymétrie des protocoles** : La relation entre clients et serveur est de type plusieurs vers un. C'est toujours le client qui déclenche le dialogue en demandant un service. Les serveurs attendent passivement les requêtes des clients.
- ✓ **Transparence à la localisation** : Les processus serveur et client peuvent résider sur la même machine ou, par l'intermédiaire d'un réseau, sur deux machines différentes interconnectées. Le logiciel Client/serveur masque aux clients la localisation du serveur en redirigeant les demandes de service si nécessaire. Un programme peut être client, serveur ou les deux.
- ✓ **Assemblage multi vendeur** : Le logiciel Client/serveur idéal est indépendant de la plate-forme matérielle ou du système d'exploitation. On doit toujours pouvoir mélanger et apparier les plates-formes client et serveur.
- ✓ **Échange de messages** : Client et serveur sont des systèmes à liaison épisodique qui interagissent au moyen de messages. Le message est le mécanisme d'émission des demandes de service et des réponses à celles-ci.
- ✓ **Encapsulation des services** : Le serveur est un « spécialiste », un message lui indique quel service est requis. C'est à lui de décider comment rendre ce service. Les serveurs peuvent être mis à niveau sans effet sur les clients tant que l'interface des messages reste la même.
- ✓ **Redimensionnement** : Les systèmes Client/serveur peuvent être redimensionnés horizontalement ou verticalement. Le redimensionnement horizontal correspond à l'ajout ou au retrait de stations clientes, avec un léger impact sur les performances. Le redimensionnement vertical correspond à la migration du serveur vers une machine plus puissante ou consiste à distribuer la charge de traitement sur plusieurs serveurs.
- ✓ **Intégrité** : Le code et les données du serveur sont gérés de façon centralisée, ce qui garantit un moindre coût de maintenance et une meilleure intégrité des données partagées. De l'autre côté, les clients restent individuels et indépendants.

Les caractéristiques ci-dessus facilitent la répartition de l'intelligence sur tout le réseau. Elles constituent une base pour la conception d'applications en réseau à liaisons épisodiques.

5.2. Technique du dialogue Client /Serveur :

5.2.1. Notions de base et fonctionnement de client/serveur :

- **Client (Client) :** Processus demandant l'exécution d'une opération à un autre processus par envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.
- **Serveur (Server) :** Processus accomplissant une opération sur demande d'un client et transmettant la réponse à ce client.
- **Requête (Request) :** Message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte du client.
- **Réponse (Reply) :** Message transmis par le serveur à un client suite à l'exécution d'une opération contenant les paramètres de retour de l'opération.

Un client exécute une application et demande l'exécution d'une opération à un serveur par le biais d'une requête. Il reçoit une réponse, lui indiquant par exemple que l'opération a été bien exécutée. Les appels au service de transport mis en jeu sont au nombre de quatre :

- 1) **Send Request () :** Permet au client d'émettre le message décrivant la requête à une adresse correspondant à la porte d'écoute du serveur.
- 2) **Receive Request () :** Permet au serveur de recevoir la requête sur sa porte d'écoute.
- 3) **SendReply () :** Permet au serveur d'envoyer la réponse sur la porte d'écoute du client.
- 4) **ReceiveReply () :** Permet au client de recevoir la réponse en prévenance du serveur.

Remarque :

Un client ne dialogue qu'avec un serveur à la fois, alors qu'un serveur peut gérer plusieurs connexions simultanément.

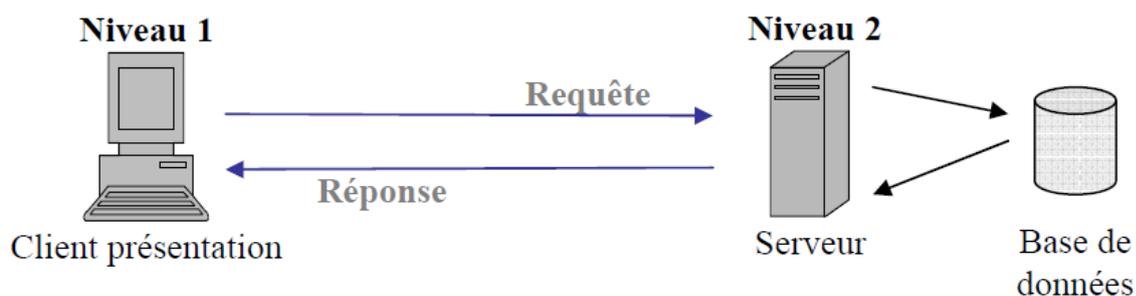
5.3. Architecture client/serveur multi niveaux :

Les applications client/serveur se différencient par la façon dont les fonctions distribuées se partagent entre le client et le serveur. Le modèle oriente serveur place plus de fonctionnalités sur le serveur tandis que le modèle oriente client, le gros de l'application tourne du côté client.

Les spécialistes du client/serveur préfèrent utiliser le terme d'architecture client/serveur à deux niveaux ou à trois niveaux plutôt que ceux d'orienté client ou d'orienté serveur.

5.3.1 Architecture client/serveur à deux niveaux :

L'architecture à deux niveaux (appelé aussi **architecture 2-tiers**) caractérise les systèmes Client/serveur dans lesquels le client demande une ressource et le serveur la lui fournit directement sans faire appel à un autre serveur intermédiaire.



Figure

05: Architecture client/serveur deux niveaux.

La simplicité et le facteur principal qui expliquent la popularité de l'architecture client/serveur à deux niveaux, cette dernière offre une grande souplesse et facilite la création d'outils frontaux permettant aux utilisateurs de créer leur application.

Avec l'arrivée des nouvelles technologies (internet, web) l'architecture client/serveur deux niveaux a évolué vers l'architecture trois niveaux.

5.3.2 Architecture trois niveaux :

L'architecture à trois niveaux (appelé aussi **architecture à 3-tiers**) est généralement partagée entre :

- **Un client** : le demandeur de ressource.
- **Un serveur d'application (middleware)** : qui est chargé de fournir la ressource en faisant appel à un autre serveur.
- **Un serveur secondaire** : fournissant un service au premier serveur (généralement un serveur de base de données).

Les applications de l'architecture client/serveur à trois niveaux sont plus faciles à déployer et à gérer sur le réseau, elles essaient de minimiser les échanges sur le réseau en créant des niveaux de service. L'architecture client serveur à trois niveaux est recommandée pour diverses raisons dont on cite celle de l'Internet

L'architecture à trois niveaux devient de plus en plus à N niveaux, cela est dû au fait que le niveau intermédiaire est implémenté sous forme d'une collection de composants.

Chaque composant se charge d'une fonction relativement petite exécutée suite à une demande d'un client, la demande d'un client peut nécessiter plusieurs composants qui s'entraident pour la satisfaire, d'où le modèle à trois niveaux est en fait à N niveau.

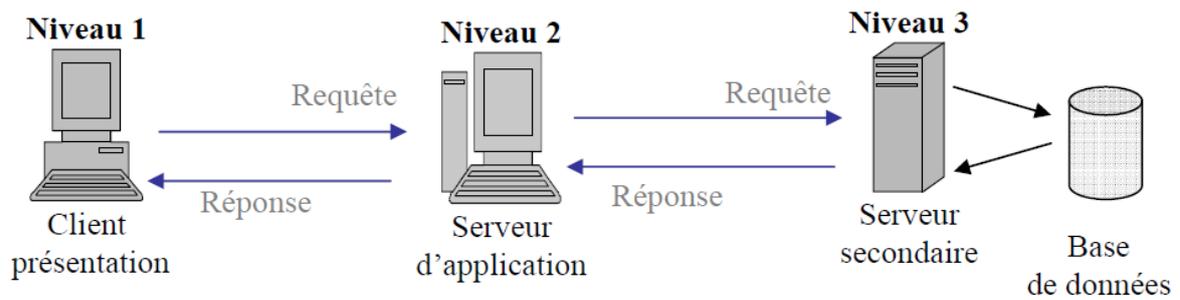


Figure06 : Architecture client/serveur trois niveaux.

5.3.3. Architecture en couche :

Les réseaux d'ordinateurs sont conçus de façon très structurée. Pour réduire la complexité de leur conception, ils sont organisés en séries de couches, chacune étant construite sur la précédente.

Cependant, dans chaque réseau, l'objet de chaque couche de niveau (n) est d'offrir certains services à la couche de niveau (n+1), en leur épargnant les détails de la mise en œuvre de ses services.

5.4. Les types de serveur :

Il existe plusieurs variantes des technologies client/serveur, comme les décrivent les paragraphes suivants :

5.4.1. Serveur de fichiers :

Dans le cas de serveur de fichiers, le client requiert des enregistrements de fichiers en émettant des requêtes au serveur de fichiers. Les serveurs de fichiers sont utiles pour partager des fichiers sur un réseau et ils sont indispensables pour créer des banques de documents, d'images, ...etc. mais l'obtention de l'information nécessite de nombreux échanges de message sur le réseau.

5.4.2. Serveur de base de données :

Dans le cas d'un serveur de base de données, le client émet des requêtes SQL sous forme de message en direction du serveur, le résultat est renvoyé au client. Le serveur utilise sa capacité de traitement pour rechercher les données demandées au lieu de transmettre tous les articles au client et de le laisser en faire la sélection, donc la puissance est répartie et utilisée de façon beaucoup plus efficace.

5.4.3. Serveur de transaction :

Dans ce modèle, le client invoque des procédures distantes résidant sur le serveur qui comporte un moteur de base de données SQL. Ces procédures distantes exécutent un ensemble d'instructions SQL. L'échange sur le réseau consiste en un seul message requête/réponse (contrairement à l'application serveur base de données pour laquelle le message requête/réponse est émis pour chaque instruction SQL). Pour ce type de serveurs l'application Client/serveur nécessite du code source au niveau du serveur.

5.4.4 .Serveur d'application Web :

Le world wide web est la première application Client/Serveur intergalactique, ce modèle consiste en des clients, léger, portable qui communiquent avec de très gros serveurs. Un serveur web renvoie des documents lorsque le client les demandes.

5.4.5. Serveur groupware :

Le groupware s'intéresse à la gestion d'informations semi-structurées telles que le texte, l'image, courrier, messagerie et ordonnancement de tâches.

5.4.6 Serveur d'application objet :

Dans le cas d'un serveur objet, l'application Client/Serveur est écrite sous forme d'un jeu d'objets communicants. Les objets client communiquent avec les objets serveur au moyen d'un courtier d'objets ou ORB (Object Request Broker). Le client invoque une méthode sur un objet distant, l'ORB localise une instance de la classe, appelle la méthode demandée et renvoie les résultats à l'objet Client.

5.5. Le fonctionnement du modèle client/serveur :

La communication entre le client et le serveur est de type transactionnel et coopératif. Le client transmet sa demande et interromp son traitement jusqu'à ce qu'il reçoive la réponse du serveur. Le mode de fonctionnement du client est dit bloquant. Le serveur quant à lui ne fait rien jusqu'à ce qu'il reçoive une requête. On dira que le serveur est en état d'écoute. Lorsque la demande a été traitée par le serveur, celui-ci se remet à l'écoute de nouvelles demandes. Il s'agit du mode de fonctionnement le plus simple entre client et serveur, signalons qu'il existe d'autres mécanismes qui permettant au client de continuer son traitement après l'envoi d'une requête, il s'agit alors d'un fonctionnement non bloquant. Dans ce cas, il n'est pas nécessaire de disposer de mécanismes de synchronisation entre client et serveur. Le client écrit sa requête dans un tampon, une fois la demande de communication au serveur acceptée, il se met en attente de réponse (en mode non bloquant, le client peut continuer son traitement celui-ci ne dépend pas de résultat de la requête). Le serveur lit le contenu du tampon et traite la requête (lui-même ou fait intervenir d'autres serveurs). Une fois le traitement terminé, la réponse écrite dans un tampon accessible par tous les serveurs.

III. Conclusion :

Dans ce chapitre, on a passé en revue quelques généralités sur les réseaux avant de présenter le modèle Client/serveur; modèle de base des systèmes distribués. On déploie aujourd'hui des applications Client/serveur servant des milliers de clients dans l'entreprise.

Ces applications tournant souvent sur de nombreux serveurs et consistent en centaines de composants. Elles sont au cœur fonctionnement de l'entreprise. Les serveurs peuvent accepter des requêtes de l'un des milliers de PC (clients) connectés.



Spa

**BRANCHE COMMERCIALISATION
DISTRICT TIZI OUZOU**



I. Historique de l'organisme d'accueil :

NAFTAL est une entreprise nationale de distribution et de commercialisation des produits pétroliers.

Sur le plan juridique NAFTAL est une société par action (SPA) depuis le 18 avril 1998. Elle est issue de deux restructurations de l'entreprise SONATRACH.

- 1) La première a eu lieu en 1980 par le décret n° 80101 de 06/04/1980 portant création de L'ERDP (Entreprise de Raffinage et de Distribution des produits Pétroliers).
- 2) La deuxième restructuration s'est traduite par la modification du décret n°80101 du 06/04/1980 par le décret 87189 de 25/04/1987 portant création de l'Entreprise Nationale de commercialisation et de distribution des produits pétroliers et dérivés, sous le sigle NAFTAL.

Le 18 avril 1998 NAFTAL est Devenu société par action (SPA), au capital social fixe de 15.650 millions de dinars est détenu entièrement de SONATRACH.

NAFTAL à fonctionné sur le plan régional en unité de distribution (UND) jusqu'au 31 décembre 2000.

Un autre plan de restructuration s'est traduit par la création des divisions, selon ses principaux produits à savoir :

(Décision N° 5540 du 15/12/1990)

- ✚ CLP (Carburant Lubrifiant Pneumatique).
- ✚ GPL (GAZ).
- ✚ AVM (Aviation Marine).
- ✚ Division bitumes.

Pour assurer convenablement ses missions, NAFTAL dispose de moyens matériels, humains et financiers adéquats. Elle applique aussi un système décentralisateur en instaurant des districts dont le district COM de Tizi-Ouzou.

II. Problématique :

La gestion des services moyens généraux et reposé sur trois postes , leur activité est un ensemble de règles qui régit les trois activités suivantes :

- L’approvisionnement du magasin par les fournisseurs.
- La distribution des produits au service qui demande.
- Le suivi des transactions entre les fournisseurs et l’entreprise et les transferts entre magasin au sein de l’entreprise.

Devant la complexité de l’activité de se service dû au sur chargement des dossiers aux quels sont confrontés quotidiennement les responsables et les agents désignés pour avoir l’information a chaque formulation de besoin de celle-ci, le besoin d’un nouveau système simple et efficace s’avère nécessaire pour assurer une gestion plus fiable et plus rigoureuse et cela pour atteindre les objectifs suivants :

- Informatisé tous les tâches du service.
- Géré les déférents documents qui sercule dans le service.
- Garder les archive dans la base de donnée pour éviter la sur charge des documents.
- Eviter les erreurs de calcul.
- Restreindre l’accès aux informations.

C’est dans cette optique que nous essayons à travers notre étude de proposer une solution aux divers problèmes que rencontre le gestionnaire en intégrant l’outil informatique pour assurer une bonne gestion de se service.

Et pour réalisé cette application on va utiliser les outils suivants :

- Netbeans.
- MY SQL serveur.
- Spring Framework qui va nous facilité plusieurs tâche.

III. Présentation du district COM Tizi-ouizou :

Le district COM de Tizi-ouizou se situe dans la zone industrielle AISSAT IDIR à OUED-AISSI (environs 13 kilomètres de Tizi-Ouzou ville).

Il gère deux centres de stockages **CSD** (Centre de Stockage et de Distribution) et **CMP** (Centre Multi Produits) et réseau de seize stations en gestion directe (GD). En générale, il veille sur la gestion, l’organisation, la promotion et le développement de l’activité de distribution des carburants terre, lubrifiants, produits spéciaux et pneumatiques. Il veille aussi sur la commercialisation des produits et la modernisation du réseau des stations services.

Pour ce faire, il analyse sa part dans le marché, il étudie les propositions et met en œuvre toutes actions susceptibles de renforcer sa position par rapport à ses concurrents.

IV. Mission et activité de l'organisme d'accueil :

District COM à pour mission principales :

- ❖ Enfûtage GPL.
- ❖ Formulation des bitumes.
- ❖ Distribution, stockage et commercialisation des carburants, GPL, lubrifiants, bitumes, pneumatique et les produits spéciaux.
- ❖ Transport de produits pétroliers.

IV.1. Organigramme général de l'organisme d'accueil :

DISTRICT COM DE TIZI-OUZOU :

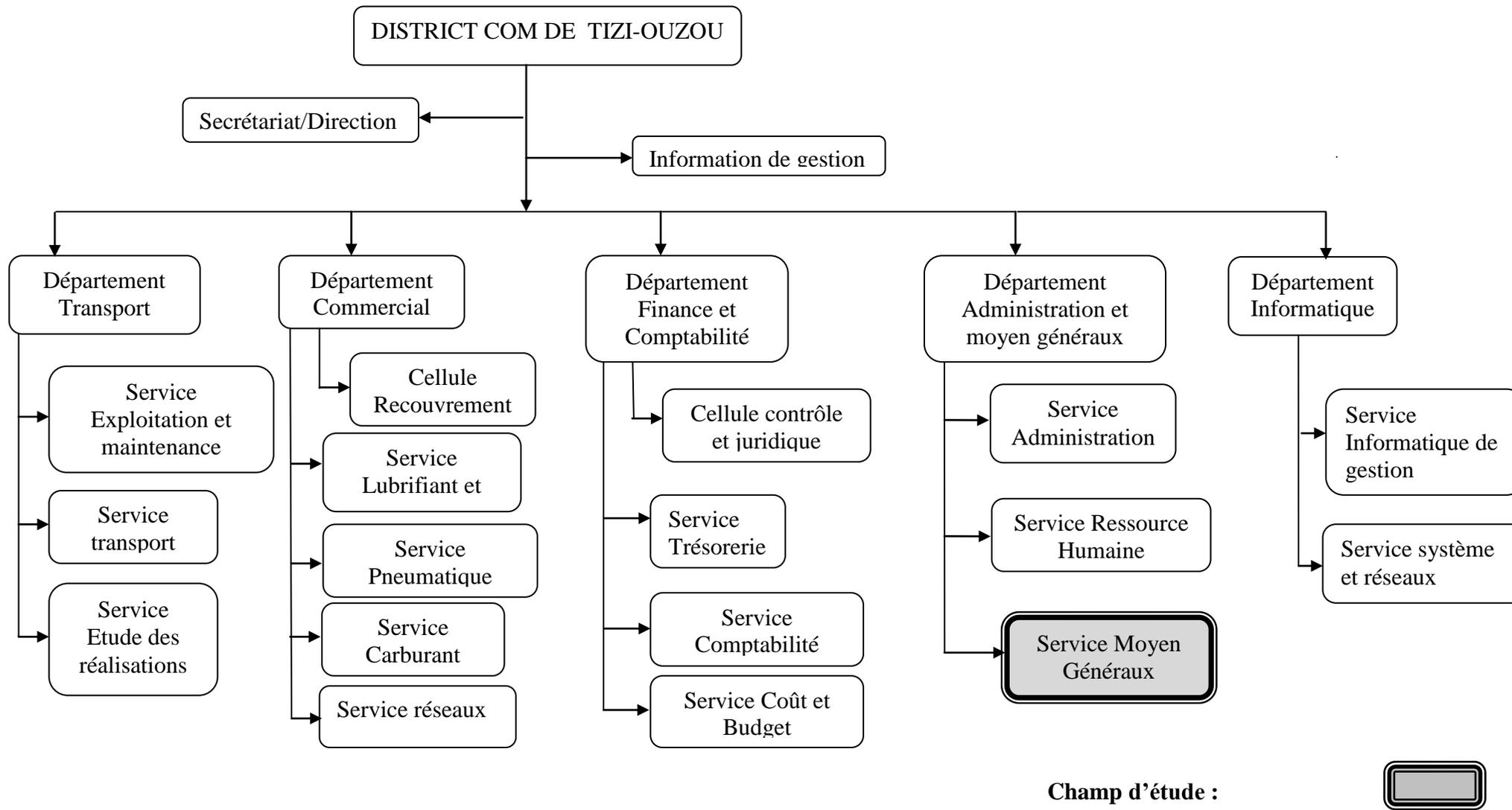


Figure07 : L'organigramme Du DISTRICT COM TIZI-OUZOU

IV.2. L'organigramme de champs d'étude :

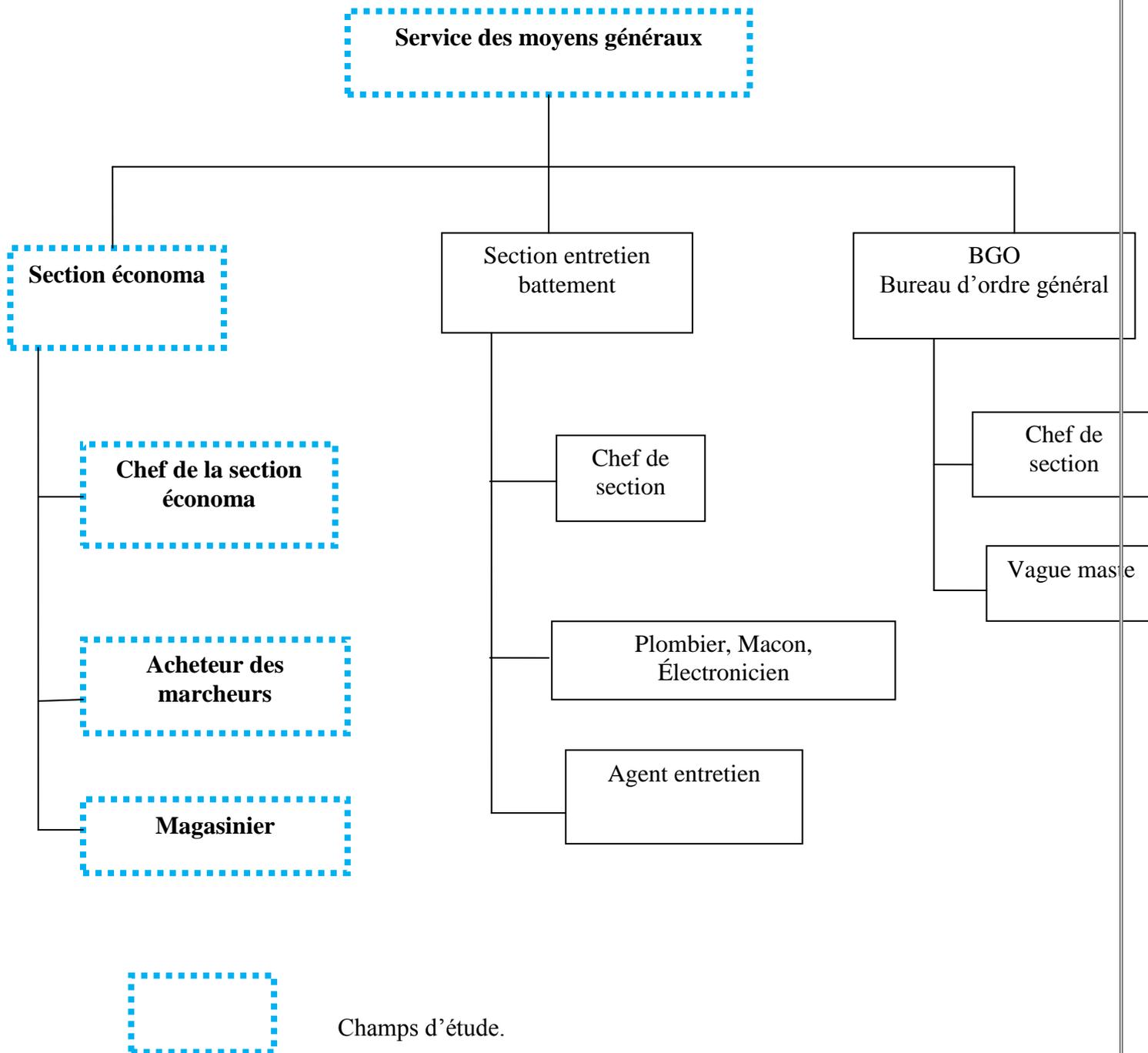


Figure08 : Organigramme de champs d'étude.

V. Description de l'organigramme général :

V.1. Présentation des différents départements :

A. Département commercial :

En terme simple, le département commercial veille à :

- L'application de la politique commerciale de l'entreprise adoptée.
- Les points de vente NAFTAL.

A.1. cellule recouvrement : C'est une cellule rattachée directement au département. Elle a pour fonction principale :

- Elle Suit et contrôle le règlement des clients.
- Elle Procède aussi au recouvrement des créances sur clients.
- Elle établie les reçus d'encaissement et saisis éventuellement les clients retardataires.

Pour ce faire. Elle est en contact permanent avec les autres services.

A.2. Service carburants_: chargé de :

- La commercialisation des carburants.
- Traiter et analyser les données du marché potentiel pour pouvoir établir les plans prévisionnels.
- Gérer et contrôler le mouvement des stocks puis s'assurer de la qualité et de la conformité du produit.
- Enregistre, analyser et suggérer les solutions remédiant aux doléances de la clientèle.

A.3. Service lubrifiants et produits spéciaux : il s'occupe de :

- La présentation te conseils à la clientèle en matière d'utilisations des lubrifiants de NAFTAL
- Suivi et de l'analyse des ventes des produits commercialisés par le centre multi produits.
- Elaboration des plans d'approvisionnements, de distributions et répond aux doléances des clients.

A.4. Service pneumatique : il s'occupe de :

- Suivi et l'analyse des vents pneumatiques.

- Elaboration des plans de ventes.
- Etablissement des programmes et planning d'approvisionnement.

A.5. Service réseau : chargé de :

- Gérer les réseaux des stations service en l'occurrence les gestions directes (GD), les gestions libres (GL), les points de ventes agréés (PVA),...
- Analyser la performance de ses stations.
- Veiller aux respects des normes de gestion.
- Veiller à la présentation de l'image de marque de NAFTAL.

B. Département finances et comptabilité :

Il se consacre principalement à l'interprétation des flux financiers en écritures Comptables puis il est traduit au bilan en fin de chaque période comptable. Il coordonne toutes les activités de comptabilité, de trésorerie budget et de patrimoine. Il comprend :

B.1. Cellule contrôle et juridique : Il veille au :

- Respect des normes de gestion.
- Contrôle des TAC.
- La protection du patrimoine.
- Suivi des contentieux et litiges qui peuvent subvenir au cours de l'exercice de l'activité de NAFTAL.

B.2. Service trésorerie : chargé de :

- Contrôlé les flux recettes et dépenses du district.
- Traiter les dossiers de paiement, d'investissement et autres dépenses.
- Etablir la situation de rapprochement des comptes de trésorerie.
- Contrôler les impayés et effectuer la comptabilisation des comptes et grand livre de trésorerie.

B.3. Service comptabilité générale : Il s'occupe de :

- La traduction des documents aux écritures comptable.
- Suivi de l'état des stocks, l'état des ventes st marges par points de vente ou par client.
- Suivi du mouvement des stocks, leur réception et leur règlement.

- La comptabilisation des charges salariales et de la déclaration fiscale.

B.4. Service coût et budget : chargé de :

- L'élaboration des budgets prévisionnels d'investissement et de fonctionnement du district.
- Consolider l'ensemble des charges nécessaires à la détermination du coût.
- Collecter les informations comptables puis procéder au calcul des charges d'exploitation.

C. Département administration et moyens généraux :

Il a pour mission, d'assurer la gestion des moyens généraux du district, l'administration et la gestion des ressources humaines. Ce département s'occupe essentiellement du facteur humain, son épanouissement formation. Toutefois, Ces tâches sont réparties comme suit :

C.1. Service moyens généraux : il assure :

- Les prestations de service en matière de transport, télécommunication et approvisionnement en fournitures.
- La gestion du stock du magasin et de la réception du courrier, son enregistrement et sa répartition.

C.2. Service administration : chargé de :

- Gérer le personnel en le déclarant aux assurances sociales.
- Elaborer la paie mensuelle.
- Fournir les données nécessaires pour le calcul de la prime de rentabilité collective.

C.3. Service ressources humaines : son rôle est de :

- Suivre l'évolution de l'effectif et du personnel en fonction des besoins du district.
- Elaborer le plan de formation annuel et organiser les départs en retraite.

D. Département transport et technique :

Sa mission consiste en la maintenance du matériel, le plan de distribution ainsi que l'étude et la réalisation des projets. Il est divisé en plusieurs services :

D.1. Service exploitation et maintenance : Il assure :

- Le maintien du matériel des dépôts en veillant à la sécurité préventive et curative.
- La répartition des volucompteurs des GD et l'achat de la pièce de rechange et son emmagasinage.
- L'application des plans de maintenance du réseau.

D.2. Service transport : charger de :

- Elaborer le plan d'approvisionnement et de la distribution par canal, par commune et par segments selon les prévisions du service carburant.
- Suivre les contrats des transporteurs privés et gérer le matériel roulant, les pièces de rechange et le personnel de conduite.

D.3. Service d'études et réalisations : son rôle est :

- Elaborer les études technico-économiques et architecturales.
- Entreprendre toute étude de rénovation ou d'extension des installations.
- Suivre les situations des travaux, les constater sous forme de situations provisoires et les présenter au paiement.

E. département informatique :

Il s'occupe de tout ce qui est informatique, que ce soit software ou hardware, il est chargé aussi de maintenir le système et le réseau, il s'occupe aussi de la récolte d'information et de les mettre à la disposition d'autres services et de la direction générale, il est divisé en deux services :

E.1. service système et réseau : il s'occupe de :

- Système et de tout ce qui est logiciel.
- Les mises à jour.
- Maintenance réseau et matériel.

E.2. service informatique de gestion (ING) : il s'occupe de :

- La récolte d'information.
- Analyse des données (information).
- Mettre les informations à la disposition d'autres services et de la direction générale.

V.2. Présentation du Centre Multi Produit (CMP) :

Le centre multi produit ou (15G) est un lieu de stockage et de vente d'une gamme variée de produits à savoir : les lubrifiants, les pneus, Les acides et l'eau distillée.

Pour l'alimentation du stock, le CMP s'approvisionne de son fournisseur principal qui la raffinerie d'Arzew.

Le CMP situe sur la route nationale N° 12 à 500 m du centre ville.

V.3. Présentation du centre de stockage et de distribution (CSD) :

Le CSD de Tizi-ouzou est implanté sur une surface de dix (10) hectares dans la zone industrielle d'Oued-Aissi, environ 8km à l'est de cette ville, chef lieu de la wilaya. Administrativement, il fait parti de la commune de Tizi-ouzou, il est mit en marche le 02 février 2001.

Il est doté d'installations annexes et générales, d'unités de stockage et de distribution des carburants et d'un hangar de stockage pneumatiques et lubrifiants.

La capacité de stockage du SCD est de 3000 m cubes de carburants et de 500 m cubes et 200 m cubes d'huile usagée.

Le centre reçoit, stocke, vend et distribue les carburants, il s'approvisionne auprès de l'unité de CAROUBIER et d'EL HARRACH.

VI. Situation informatique de l'organisme d'accueil (District COM Tizi-ouzou) :

- **L'aspect Humain :** district COM dispose de :
 - ✓ Deux ingénieurs d'état en informatique dans le service ING (Information de gestion).
 - ✓ Deux ingénieurs d'état en informatique au niveau de service réseau et le CSD.

▣ L'aspect logiciel :

Plusieurs applications ont été développées parmi elles :

- ❖ **SGC :** Système de Gestion des Créances applicable au niveau de la cellule recouvrement.
- ❖ **VENTES LUB :** système Stock Vente Lubrifiants.
- ❖ **CGCA :** système pour le service Comptabilité Générale.
- ❖ **SIG :** Système de Gestion des Investissements.
- ❖ Le logiciel paie.
- ❖ **GRH :** le logiciel Gestion des Ressources Humains.
- ❖ **WINCANAL :** pour le service coûts et budget.
- ❖ **CTM :** le logiciel gérant les flux de trésoreries (Cash Treasury Manager).
- ❖ **NAFT COM :** logiciel de facturation installé au niveau des centres.

 **L'aspect matériel**

Désignation	Service	Nombre
Dell portable Pentium IV -Disque dur 40 GO -RAM 128 MO	Direction Directeur	1
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO		1
Imprimante laser HP 4L.		1
Pentium IV -Disque dur 40 GO -RAM 128 MO	Secrétariat	1
Imprimante laser HP 4L.		1
Pentium III -Disque dur 40 GO -RAM 64 MO		1
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO	Service information de gestion	2
Imprimante Epson LQ 1070.		1
Département finances et comptabilité		
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO	Secrétariat	1
Imprimante laser HP 4L.		1
IBM Pentium III -Disque dur 40 GO -RAM 64 MO		1
Pentium IV (HP Compaq)	Service comptabilité	2

-Disque dur 40 GO -RAM 128 MO Imprimante matricielle 132 colonnes (Epson).	générale	1
AMD Pentium III -Disque dur 20 GO -RAM 64 MO Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO Imprimante laser HP 4L.	Service trésorerie	2 1 1
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO Imprimante laser HP 4L.	Service Coût et budget	2 1
Département commercial		
IBM Pentium III -Disque dur 20 GO -RAM 64 MO	Secrétariat	1
Pentium III -Disque dur 20 GO -RAM 64 MO Imprimante matricielle 132 colonnes (Epson).	Cellule recouvrement	2 2
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO Imprimante matricielle 132 colonnes (Epson).	Service lubrifiant et produit spéciaux	1 1
Pentium IV (HP Compaq) -Disque dur 40 GO	Service	1

-RAM 128 MO Imprimante laser HP 4L.	pneumatique	1
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO Imprimante laser HP 4L.	Service carburant	2 1
Département Transport/Technique		
AMD Pentium III -Disque dur 20 GO -RAM 64 MO	Secrétariat	1
Pentium III -Disque dur 20 GO -RAM 64 MO Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO Imprimante laser HP 4L.	Service Exploitation et maintenance	1 1
Pentium III (Compaq Presario) -Disque dur 20 GO -RAM 64 MO Imprimante matricielle LQ 2080	Service Transport	1 1
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO Imprimante laser HP 4L.	Service Etude des réalisations	1 1

Département Administration et moyen généraux		
IBM Pentium III -Disque dur 20 GO -RAM 64 MO Imprimante laser HP 4L.	Secrétariat	1 1
UNIKA Pentium II Disque dur 10 GO -RAM 32 MO DELL Pentium IV -Disque dur 40 GO -RAM 128 MO Imprimante laser HP 4L. Imprimante matricielle LQ 2080	Service Administration	1 3 1 3
AMD Pentium III -Disque dur 20 GO -RAM 64 MO Imprimante matricielle LQ 2080	Service Ressource Humaine	1 1
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO Imprimante matricielle LQ 2080	Service moyen généraux	1 1
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO AMD Pentium III -Disque dur 20 GO	Centre 15G5 (CSD)	6 2

-RAM 64 MO Imprimante matricielle 132 colonnes (Epson).		2
Pentium IV (HP Compaq) -Disque dur 40 GO -RAM 128 MO Imprimante matricielle 132 colonnes (Epson).	Centre 15C5 (CMP)	3
		3

VII. Etude des postes :

1. Introduction :

L'objectif de cette partie consiste à étudier les différents postes de travail concernés par notre domaine d'étude et de recueillir toutes les informations manipulées et tous les documents échangés entre les postes.

L'analyse des postes est réalisée en spécifiant tout particulièrement :

- Les tâches à accomplir.
- Les documents en entrée.
- Les documents en sortie.
- Les documents interne au poste.
- Les fichiers et registres utilisés.
- Le matériel utilisé par ce poste.

2. Liste des postes :

- a. chef de section économat.
- b. Administrateur.
- c. Magasinier.

➤ Fiche d'étude du poste :

2.1. Premier poste :

- Description : chef de section.
- Service de rattachement : moyens généraux.
- Nombre d'employer : 1.
- Responsabilité : responsable de la section.
- Taches des postes :
 - *Inspection des fournisseurs.
 - *Comparaison entre les trois fournisseurs et choisir le moins cher.
 - *Etablir le bon de commande et le signer.
 - *Envoyer le bon de commande à l'acheteur.
 - *Envoyer le bon de livraison au magasinier.
 - *Comparer le bon de commande et la facture.
 - *Constitution de dossier pour le paiement de la facture.

*Remettre le dossier au département finance pour établir un chèque ou un virement.

Documents entrants	Provenance	Documents sortants	Destination
Facture	Fournisseurs (acheteurs)	Bon de commande	Acheteur
Bon de livraison	Fournisseurs (acheteurs)		
Bon entée	Magasinier	Dossier pour paiement	Département finance

2.2. Deuxième poste :

- désignation : Administrateur.
- nombre d'employer : 1.
- responsabilité : Maintenance de l'application.
- taches de poste :
 - *Géré les comptes utilisateur.
 - *Géré les produits.

2.3. Troisième poste :

- désignation : magasinier.
- Nombre d'employer : 1.
- Responsabilité : gérer le stock.
- taches de poste :
 - *Vérifier la marchandise livrée et la compter selon le bon de livraison.
 - *Remettre la marchandise en stock.
 - *Etablir un bon d'entrée (comme quoi la marchandise est entée).
 - *Envoyer le bon d'entée au chef de section.
 - *signer la demande de fourniture.
 - *établir un bon de sortie.
 - *établir un bon de transfert.

Documents entrants	Provenance	Documents sortants	Destination
Bon de livraison	Chef de section	Bon entée	Chef de section
Demande de fourniture	Agent d'un service quelconque	Bon de sortie	Service quelconque
		Bon de transfert	Chef de section

VIII. Etude des documents :

1. Introduction :

Au cours de l'étape de recherche d'information nous avons recensé les différents documents utilisés entre les divers acteurs, ainsi que nous avons noté l'importance de ces documents car ils nous ont offert la possibilité de relever des données nécessaires à notre étude, d'examiner les relations entre les éventuelles rubriques et les contrôles qui peuvent y être appliqués.

Pour chaque document on précisera ses caractéristiques et on le décrit avec des indications pour la répartition des informations et pour chacune, on désignera la longueur, le type et des observations sur des rubriques si elles existent et sa en utilisant une fiche.

Le type des rubriques peut être :

A : Alphabétique.

N : Numérique.

AN : Alphanumérique.

Pour le type des documents il existe deux types :

- Document interne.

- Document externe.

a- Document interne :

C'est tous les documents utilisés à l'intérieure du champ d'étude.

1-Document interne de position :

Sont des documents utilisés que dans le poste même ou ils sont créés.

2-Document interne de liaison :

Sont des documents qui transitent entre les postes de champ d'étude

b- Document externe :

C'est tous les documents qui proviennent ou destinés à l'extérieur de champ d'étude.

2. Liste des documents :

Fiche d'analyse du document N0 : 01			
Identification : Bon de commande			
Code : 01			
Désignation : Bon de commande			
Rôle : Ce document contient la liste des produits commandés			
Nombre d'exemplaire : 03			
Poste Créateur : Chef de section			
Nature : Externe			
Désignation rubrique	type	taille	observation
Numéro de Bon de Commande	N	(15)	
Date établissement	Date		
Nature de la commande	A	(30)	
Commander à	AN	(70)	
Livrer à	AN	(70)	
Référence Devis	N	(10)	
Date Devis	Date		
Délai de Livraison	AN	(15)	
Quantité commandée	N	(15)	
Prix unitaire	N	(20)	
montant	N	(30)	

Solliciter par	A	(30)	
Approuvé par	A	(30)	
Autoriser par	A	(30)	
Solliciter Titre	A	(30)	
Approuvé Titre	A	(30)	
Autoriser Titre	A	(30)	
Description de l'imputation	A	(30)	

Fiche d'analyse du document N0 : 02

Identification : Bon d'entrée Magasin

Code : 02

Désignation : Bon d'entrée Magasin

Rôle : Ce document contient la liste des produits Entrée en Magasin

Nombre d'exemplaire : 03

Poste Créateur: Magasinier

Nature : interne de liaison

Désignation rubrique	type	taille	observation
Numéro du bon d'entrée magasin	N	(10)	
Unité	AB	(15)	
Magasin	AB	(15)	
Fournisseur	AB	(20)	
Numéro facture	N	(15)	
Numéro commande	N	(15)	
Date de réception	Date		
Monnaie d'origine	A	(15)	
Montant en dinar	N	(30)	
DD+Montant Facture	N	(30)	
Montant total en dinar	N	(30)	
Bon d'entre au magasin	AB	(30)	
Quantité entrée	N	(15)	
Prix unitaire en dinars y compris droit douane	N	(20)	
Montant en Dinars y compris Droit Douane	N	(20)	
Montant total en Dinars y compris droit douane	N	(30)	
Montant en lettre	AB	(40)	
Nom de marcheur	AB	(20)	
Date de signature par marcheur	Date		
Nom de magasinier	AB	(20)	
Date de signature par magasinier	Date		
Nom de responsable fiche de stock	AB	(30)	
Date de signature par responsable fiche de stock	Date		

Fiche d'analyse du document N0 : 03

Identification : Bon de Sortie

Code : 03

Désignation : Bon de Sortie

Rôle : Ce document contient la liste des produits Sortie du Magasin

Nombre d'exemplaire : 04

Poste Créateur: Magasinier

Nature : Externe

Désignation rubrique	type	taille	observation
Numéro bon de sortie magasin	N	(10)	
Bon de sortie magasin	A	(30)	
Date de sortie magasin	Date		
Imputation	A	(10)	
Item	AB	(05)	
Code référence	N	(10)	
Quantité demandées	N	(10)	
Quantité fournées	N	(10)	
Prix unitaire	N	(10)	
montant	N	(10)	
Montant total	N	(20)	
observation	AB	(20)	
Nom de demandeur	AB	(10)	
Date de demande	Date		
Nom de réceptionnaire	AB	(10)	
Date de réception	Date		
Nom de magasin	AB	(10)	
date	Date		
Nom de gérant de stock	AB	(10)	
Date de réception	Date		

Fiche d'analyse du document N0 : 04

Identification : Bon de transfert Magasin

Code : 04

Désignation : Bon de transfert Magasin

Rôle : Ce document contient la liste des produits transféré par le Magasin

Nombre d'exemplaire : 04

Poste Créateur: Magasinier

Nature : Externe

Désignation rubrique	type	taille	observation
Numéro de bon de transfert magasin	N(10)	(10)	
Bon de transfert magasin	AB(30)	(30)	
Unité	AB(20)	(20)	
Magasin	AB(10)	(10)	
Date de transfert	Date		
Adresse de transfert	AN(80)	(80)	
Code Produit	N(15)	(15)	
UM	N(5)	(05)	
Quantité transférée	N(10)	(10)	
Prix unitaire	N(20)	(20)	
Montant	N(15)	(15)	
Montant Total	N(15)	(15)	
Montant en lettre	AB(50)	(50)	
Nom de chef de magasin cédant	AB(10)	(10)	
Date de signature cédant	Date		
Nom de chef de magasin cessionnaire	AB(10)	(10)	
Date de signature cessionnaire	Date		

Fiche d'analyse du document N0 : 05

Identification : Demande de fourniture

Code : 05

Désignation : Demande de fourniture

Rôle : Ce document contient la liste des produits demandé par un autre service

Nombre d'exemplaire : 03

Poste Créateur: un service quelconque

Nature : Externe

Désignation rubrique	type	taille	observation
Numéro de la demande fourniture	N	(15)	
Date d'établissement	Date		
Numéro d'envoi	N	(10)	
Imputation	A	(10)	
Code Produit	N	(10)	
Demande fourniture	AB	(30)	
Numéro référence	AN	(10)	
Quantité demandée	N	(10)	
Quantité accordée	N	(10)	
Observation	AB	(15)	
Date de réception marchandise	Date		
Nom du réceptionnaire	AB	(10)	
Fonction de récepteur	AB	(20)	
Nom demandeur	AB	(10)	
Qualité Demander	AB	(20)	
Nom magasinier	AB	(10)	
Qualité Livrai	AB	(20)	

Fiche d'analyse du document N0 : 06

Identification : Bon de Livraison**Code :** 06**Désignation :** Bon de Livraison**Rôle :** Ce document contient la liste des produits Livrai a l'Enterprise**Nombre d'exemplaire :** 02**Poste Créateur:** Le Fournisseur**Nature :** Externe

Désignation rubrique	type	taille	observation
Numéros de bon de livraison	N(10)	(10)	
Bon de livraison	A(30)	(30)	
Quantité Livrai	N(15)	(15)	
Date de livraison	Date		
Adresse de livraison	AN(60)	(60)	
expéditeur	AB(50)	(50)	
Adresse de l'expéditeur	AN(60)	(60)	

Fiche d'analyse du document N0 : 07

Identification : Facture**Code :** 07**Désignation :** Facture**Rôle :** Ce document contient la liste des produits Livrai et la Montant**Nombre d'exemplaire :** 02**Poste Créateur:** Le Fournisseur**Nature :** Externe

Désignation rubrique	type	taille	observation
Numéro facture	N	(10)	
Désignation de produit	A	(30)	
Quantité Produit	N	(15)	
Numéro bon de commande	Date		
Date facture	N	(10)	
Numéro bon de livraison	N	(10)	
Type de Règlement	A	(10)	
Fournisseur	AN	(60)	
Adresse de fournisseur	AN	(60)	
Montant total	N	(20)	
TVA	N	(20)	
Montant a payer	N	(20)	
Montant en lettre	A	(60)	

Fiche d'analyse du document N0 : 08

Identification : Tableau de Comparaison

Code : 08

Désignation : Tableau de Comparaison

Rôle : Ce document contient la liste des facture préforma

Nombre d'exemplaire : 01

Poste Créateur: Chef de section

Nature : interne de Position

Désignation rubrique	type	taille	observation
Numéro	N	(15)	
Date d'établissement	Date		
Désignation produit	AN	(120)	
Quantité produit	N	(10)	
Nom fournisseur1	A	(25)	
Nom fournisseur2	A	(25)	
Nom fournisseur3	A	(25)	
Prix fournisseur1	N	(15)	
Prix fournisseur2	N	(15)	
Prix fournisseur3	N	(15)	
Montant a payé par naftal	N	(15)	
Nom fournisseur moins distant	N	(15)	
Nom chef service MGX	A	(25)	
Nom chef de dept AMGX	A	(25)	
Nom directrice district	A	(25)	

Fiche d'analyse du document N0 : 09

Identification : Bilan

Code : 09

Désignation : Bilan

Rôle : Ce document contient le Bilan d'un mois

Nombre d'exemplaire : 02

Poste Créateur: Chef de section

Nature : interne de Position

Désignation rubrique	type	taille	observation
Date du Bilan	N	(10)	
Numéro Bilan	Date		
Numéro Compte	N	(15)	
Stock initiale	N	(20)	
Entrée	N	(10)	
Inter unité	N	(10)	
Total entrée	N	(30)	
Consommait	N	(20)	
Cession	N	(10)	
Stock Final	N	(30)	

I-Introduction :

En programmation informatique, un **Framework** est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture). Un Framework se distingue d'une simple bibliothèque logicielle principalement par :

- son caractère générique, faiblement spécialisé, contrairement à certaines bibliothèques ; un Framework peut à ce titre être constitué de plusieurs bibliothèques chacune spécialisée dans un domaine. Un Framework peut néanmoins être spécialisé, sur un langage particulier, une plateforme spécifique, un domaine particulier (Reporting, mapping, etc.).
- le cadre de travail (traduction littérale de Framework) qu'il impose de par sa construction même, guidant l'architecture logicielle voire conduisant le développeur à respecter certains patterns ; les bibliothèques le constituant sont alors organisées selon le même paradigme.

Les Framework sont donc conçus et utilisés pour modeler l'architecture des logiciels applicatifs, des applications web, des middleware et des composants logiciels. Les Framework sont acquis par les ingénieurs, puis incorporés dans des logiciels applicatifs mis sur le marché, ils sont par conséquent rarement achetés et installés séparément par un utilisateur final. Des tentatives de francisation du terme ont été faites. On trouve ainsi parfois les termes cadre d'applications, proposé par l'Office Québec de la langue française, canevas ou cadriceil terme en usage depuis au moins 1997, aussi socle d'applications.

II-les différents types de Framework : [3] [5] [22] [23]

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 inter logicielle (*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*.
 - On va s'intéresser aux Framework d'entreprise car on va faire une application de gestion de stock pour une entreprise.
 - On a sélectionné quelque Framework entreprise : Wavemaker, Sofi, Spring, Qt Jambi, Hibernate, Xyster Framework, Laravel, Dojo.
 - On va prendre en considération les Framework entreprise compatible avec java :

Wavemaker : WaveMaker est une interface graphique et de l'environnement de développement d'applications open-source (ADE).

Sofi : SOFI (Framework) est un cadre créé pour simplifier le développement d'applications Web avec les technologies Java.

Qt jambi : Qt Jambi est un Framework Java permettant de créer des applications riches basées sur la bibliothèque logicielle Qt.

Spring : Spring 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,

II-1- Wave Maker Framework: [6] [7] [8] [22]

II-1-1-Introduction:

Wave Maker est un environnement graphique de développement rapide d'applications Web basées sur des normes. Utilisez-le pour créer des applications telles que la saisie des commandes, gestion de projet, et des formulaires Web.

II-1-2-Définition:

WaveMaker est une interface graphique et de l'environnement de développement d'applications open-source (ADE). Il assure le développement rapide d'applications (RAD), les outils qui sont utilisés pour créer des applications Web, y compris les forums Web et les formulaires de saisie des commandes. Compatible avec Windows, Linux et Mac OS, applications WaveMaker exécutés sur un serveur Java standard.

Composantes essentielles de WaveMaker sont WaveMaker Studio et cadre d'exécution WaveMaker. WaveMaker Studio fonctionne dans un navigateur Web pour faciliter glissé-déposer le développement d'applications Web. Cadre d'exécution WaveMaker est inclus dans l'application Studio.

II-1-3-Evolution (développement):

7 août 2008 :

Mots à la mode 2.0: Qu'est-ce que le Web 2.0? Qu'est-ce que AIR? Qu'est-ce que Ajax?

Le très hyper mais mal défini termes Web 2.0, Rich Internet Application (RIA) et Ajax sont mieux compris lorsqu'ils sont définis ensemble.

12 août 2008 :

Visualisation Web Services: WaveMaker et Xignite

Seth Godin, le gourou du marketing, définit un mashup comme une manière distincte de la diffusion des idées. En particulier, le Web 2.0 mashups permettent aux développeurs de combiner des données intéressantes et visualiser ces données par le biais d'une application web.

21 août 2008 :

WaveMaker en Top 10 Téléchargements Apple

WaveMaker a été choisi comme un Pick-major pour le site de téléchargement d'Apple , et ont fait irruption dans le top 10 des téléchargements d'Apple aujourd'hui, devançant Google Earth pour la dixième fente!

10 décembre 2009 :

WaveMaker 6 fait SaaS simple pour Mac

Présente première plate-forme de développement Open Cloud pour Mac OS X

23 décembre 2009 :

DevBuild, Beta, GA ... WaveMaker 6.1.0 DevBuild 1

Dans les versions récentes WaveMaker a commencé à utiliser le terme «DevBuild».

9 février 2010 :

WaveMaker (ami-395dbb50) Maintenant sur TheCloudMarket.com

WaveMaker est maintenant sur TheCloudMarket.com: [#](http://thecloudmarket.com/image/ami-395dbb50) / détails

20 juillet 2010 :

Review: WaveMaker offre pour le nuage comme PowerBuilder fait pour le client-serveur

29 juillet 2010 :

Une enquête révèle que suite développeur Maj Pour Java

WaveMaker a récemment mené une enquête auprès des membres WaveMaker

communautaires. L'enquête portait sur comment et pourquoi les développeurs migrent des plates-formes de développement de propriété à Java basée sur des normes.

13 octobre 2010 :

Annonçant la DevBuild 6.2.2

La 6.2.2 DevBuild est maintenant disponible. Nous avons amélioré la fonction de thème et des modèles fournis qui s'intègrent à thèmes.

1 octobre 2011 :

Deuxième version bêta - 6.4.2

WaveMaker 6.4.2 Beta est maintenant disponible. membres de la communauté peuvent télécharger à partir de

la page de DevBuild . 6.4.2 est la version bêta finale prévue pour 6.4. La prochaine version sera la version GA.

16 décembre 2011 :

WaveMaker 6.4 GA

20 avril 2012 :

Écran Casts pour WaveMaker 6,5 M1

de nouvelles fonctionnalités dans le WaveMaker 6.5 Milestone presse. Pour aider à vous familiariser avec quelques-unes des nouvelles fonctionnalités,

17 décembre 2012 :

WaveMaker 6.6.0 M1

Nous sommes fiers de lancer WaveMaker 6.6.0 M1. La version M1 inclut des changements majeurs à widgets composites, ce qui rend beaucoup plus facile à construire et partager des composants réutilisables avec d'autres utilisateurs et d'autres projets.

30 janvier 2013 :

WaveMaker 6.6.0 M2

L'équipe de développement est fière WaveMaker annoncer la disponibilité de WaveMaker 6.6.0 M2, la deuxième étape de la libération 6.6.0.

23 juillet 2013 :

WaveMaker mise à jour

Nous avons travaillé dur pour fermer de nombreux problèmes de retard dans Jira. Nous avons créé une branche 6.6.0 M3 dans GitHub (

https://github.com/cloudjee/wavemaker/tree/6.6M3_Release). WaveMaker 6.6.0 M3, la troisième étape de la libération 6.6.0 sortira la semaine prochaine. Le WaveMaker 6.6.0 GA version comprendra l'intégration avec la plate-forme CloudJee (<http://www.cloudjee.com>)

Le 3 décembre 2013 :

WaveMaker 6.7.0 M3

L'équipe de développement WaveMaker annonce la disponibilité de WaveMaker 6.7.0 M3.

II-1-4-Principe d'utilisation:

Téléchargez et installez WaveMaker, comme décrit dans le Guide d'installation WaveMaker 6.4 . Travailler à travers les didacticiels pour obtenir une compréhension de la façon d'utiliser WaveMaker. Les nouveaux utilisateurs devraient commencer par Bonjour Ondulés mondiale. WaveMaker permet aux développeurs web de créer Ajax applications. Le cadre WaveMaker s'intègre Spring , Spring Security (anciennement Acegi) , Dojo Toolkit 1.0, l'authentification, Lightweight Directory Access Protocol (LDAP), Active Directory et Plain Old Java Objects (POJO) et leurs produits comprennent de Visual Ajax Studio applications Internet riches (RIA) le développement et le serveur WaveMaker Rapid Deployment pour l'application Java.

Applications générées par WaveMaker sont accordées sous licence Apache .

Applications WaveMaker sont créés en utilisant le studio WaveMaker, un WYSIWYG studio de développement qui s'exécute dans un navigateur et permet de glisser-déposer le développement d'applications web suivant un modèle-vue-contrôleur architecture.

WaveMaker soutient le développement rapide d'applications (RAD) pour le web, semblable à ce que des produits tels que PowerBuilder et Lotus Notes prévus pour serveur informatique client.

Applications WaveMaker exécutés dans un serveur standard Java basé sur Tomcat , le Toolkit Dojo , Spring et Hibernate . Actuellement, il est pris en charge sur Microsoft Windows, Linux et Macintosh.

Les principales caractéristiques de WaveMaker aide en font un environnement intuitif dans lequel pour construire des applications.

- **WaveMaker Studio fonctionne dans un navigateur.** Applications créées avec WaveMaker sont effectivement en cours d'exécution pendant que vous les construisez, ce qui rend beaucoup plus facile de tester des applications Web et de débogage.
- **WaveMaker construit des applications utilisant des widgets.** L'interface utilisateur d'une application WaveMaker comprend des pages et des widgets qui fournissent des fonctionnalités et le comportement pré-construit. WaveMaker contient une grande variété de widgets configurables tels que des formulaires, des éditeurs et des boutons qui permettent aux utilisateurs d'interagir avec votre application. Pour un aperçu des fonctionnalités du widgets, consultez les sections du widgets sous Applications bâtiment: notions essentielles . Pour obtenir une description de chaque type de widgets, consultez les sections "Widgets référence» sur la page d'accueil de la documentation WaveMaker .
- **WaveMaker objets de données sont des widgets puissants liés à la base de données.** Après avoir défini une nouvelle base de données ou d'importer une base de données existante, WaveMaker crée automatiquement un widgets objet de données pour chaque table dans la base de données. widgets d'objets de données, il est facile de construire des formulaires Web qui créent, lire, mettre à jour et supprimer des enregistrements d'une base de données sans codage.
- **Applications WaveMaker sont événementielles.** Utilisateurs de contrôler ce qui se passe par les actions qu'ils entreprennent. Par exemple, lorsqu'un utilisateur clique sur un bouton, choisit un élément dans un menu ou entrer des données dans un widget d'un éditeur, un ou plusieurs événements sont déclenchés. Ces événements provoquent

le programme pour répondre en accédant à une nouvelle page, insertion d'un enregistrement dans une base de données, et ainsi de suite.

- **Widgets propriétés définissent le comportement.** Vous n'avez plus le développement en définissant des propriétés sur les widgets dans WaveMaker Studio. Faites glisser un widget sur la toile pour le positionner et définir des propriétés sur le widget pour spécifier son comportement.
- **Fonctions Widgets étendre comportement.** Widgets ont également des fonctions qui peuvent être appelées directement en utilisant JavaScript. Ces fonctions permettent aux développeurs avancés d'étendre le comportement d'un widget.

II-1-5-Cas d'utilisation et exemple d'application :

WaveMaker peut être utilisée pour les buts suivants:

- Pour prototype documents et des applications Web sans base de données ou codage de la connaissance
- Pour développer des applications Web pilotées par formulaire, sans codage
- Pour concevoir rapidement des applications Internet riches - même sans connaissances Java
- Pour concevoir l'entreprise évolutive en tant que service (EAA) et Software as a Service (SaaS) les applications

Qui devrait utiliser WaveMaker?

Utilisateurs dans une variété de rôles et à des niveaux allant de large expertise technique peuvent créer des applications avec WaveMaker:

- **Analyste d'affaires** Un analyste d'affaires peut prototyper des applications Web et des exigences en matière de documents sans codage et aucune connaissance des bases de données.
- **développeur de Citizen** Un développeur ayant des connaissances de base des bases de données peut utiliser WaveMaker pour construire des applications Web orientées forme sans codage.
- **Développeur d'affaires** Un développeur en connaissance de Javascript utiliser WaveMaker pour construire des applications Internet riches rapidement et sans aucune connaissance de Java.
- **Développeur** Un expert en informatique ou développeur de logiciels fourni avec la connaissance de Java peuvent utiliser WaveMaker de construire SaaS évolutives et les applications d'entreprise.

II-1-6-Avantages :

WaveMaker est la seule solution pour construire et déployer rapidement des applications Web évolutives qui répondent à l'architecture, de la sécurité et de données des politiques de l'AIC. Visuel Ajax Studio WaveMaker permet de projets délégués DSI aux équipes ministérielles, en

réduisant les coûts de développement tout en éliminant les projets informatiques voyous. Caractéristiques CIO Safe™ de Studio comprennent:

- Réduire les risques et les charges qui résultent de développement extensif de code personnalisé et d'entretien.
- Éliminer verrouillage du fournisseur en utilisant facilement disponibles, open source Ajax widgets et services Web: les clients de graisse plus exclusives, les déploiements de postes de travail coûteux ou mises à jour forcées.
- Connectez développeurs de services gérés qui ont été créés et / ou approuvé par elle.

WaveMaker de Visual Studio génère automatiquement Ajax le cadre de déploiement rapide. Cadre de déploiement rapide de WaveMaker pour Java permet Secure IT, gérer et maintenir des applications web fiables en utilisant la sécurité existant et gestion de l'identité tels que LDAP, SiteMinder et la sécurité de db. De par sa conception, le cadre n'introduit pas de nouveaux modèles de sécurité ou exigences. WaveMaker s'intègre avec, et s'appuie sur les meilleures pratiques qui sont déjà en place tels que les contrôles d'accès basés sur les rôles, en adhérant à la gouvernance de la DSI et les exigences de conformité SOX. Cadre de WaveMaker se branche sur le logiciel de sécurité et de gestion existant qui a déjà été approuvé, acheté et déployé par le CIO et le centre informatique.

Principales caractéristiques du cadre comprennent:

- Aucun serveur de boîte noire: WaveMaker est open source et construit sur des standards ouverts,
- Evolutivité: Le cadre peut héberger des applications qui vont de servir de petites équipes commerciales sur toute la hauteur de grandes entreprises sur des grilles. En fait, la WaveMaker Studio est construit sur le cadre WaveMaker.
- Miser sur l'infrastructure existante: il fonctionne sur le standard industriel de AppServers Apache Tomcat et J2EE (Java EE).

Ensemble, le WaveMaker visuel Ajax Studio et cadre de déploiement rapide (pour Java) comprennent une solution complète, CIO sécurité pour l'entreprise.

II-2-Qt Jambi Framework: [17] [18] [19] [22]

II-2-1-Introduction:

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.

II-2-2-Définition:

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++.

II-2-3-Evolution (développement) :

Qt Software est fondée par Eirik Eng et Haavard Nord en 1994 sous le nom de *Quasar Technologies*. Elle change rapidement de nom pour celui de Trolltech. Les deux fondateurs commencent le développement de la bibliothèque Qt en 1991.

Le 28 janvier 2008, Nokia annonce qu'un accord est conclu pour l'acquisition de Trolltech. Cet achat ne remet pas en cause la politique de double licence de son produit phare, Qt⁴. Une fondation liée à KDE a été créée en 1998 et a pleine autorité pour rendre Qt libre sous une licence de style BSD si la version libre de Qt n'était plus développée, si sa licence était modifiée ou encore si Qt Software déposait le bilan⁵. L'entreprise change officiellement de nom le 28 septembre 2008 pour devenir Qt Software. Ce changement de nom est également l'occasion pour renommer Qtopia, la version embarquée de Qt, en *Qt Extended* et de publier une version 4.4.3 de Qt ne contenant que les changements de nom dans les divers fichiers sources.

Le 7 mars 2011, on apprend que les droits concernant QT Frameworks vont être transférés de Nokia à Digia, une entreprise finlandaise.

II-2-4-Principe d'utilisation:

Qt-Jambi est une bibliothèque logicielle orientée objet et développée en Java par la société Qt-software. Elle offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. Qt-Jambi est par certains aspects un Framework lorsqu'on l'utilise pour concevoir des interfaces graphiques ou que l'on architecture son application en utilisant les mécanismes des signaux et slots par exemple.

Qt-Jambi permet la portabilité des applications qui n'utilisent que ces composants par simple recompilation du code source. Les environnements supportés sont les Unix (dont Linux) qui utilisent le système graphique X Windows System, Windows et Mac OS X. Le fait d'être une bibliothèque logicielle multiplateforme attire un grand nombre de personnes qui ont donc l'occasion de diffuser leurs programmes sur les principaux OS existants.



Qt Software fournit un ensemble de logiciels pour faciliter le développement d'applications Qt-Jambi :

- Qt-designer est un concepteur d'interface graphique, il enregistre les fichiers .ui

- Qt-démo présente un ensemble de petits programmes (sous forme d'exécutables et de sources) qui montrent les fonctionnalités offertes par la bibliothèque.

Note : il est possible d'utiliser Qt-Jambi en parallèle avec la librairie Swing en utilisant qt-awt-bridge.

II-2-5-Cas d'utilisation et exemple d'application :

Cas d'utilisation :

- 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.

Exemple d'application :

Nous allons jeter un œil de plus près Qt Jambi pour voir comment le Framework Qt est intégré à Java. Nous allons le faire en écrivant un petit exemple d'application. L'application est un jeu appelé *Qtanoid*, inspiré par le classique d'arcade d'un nom similaire, où les joueur marque des points en utilisant une palette de rebondir une balle contre un mur de briques dans une tentative de démolir brique par brique. Nous verrons aussi comment utiliser Qt Designer et JUIC pour créer l'interface utilisateur de l'application.



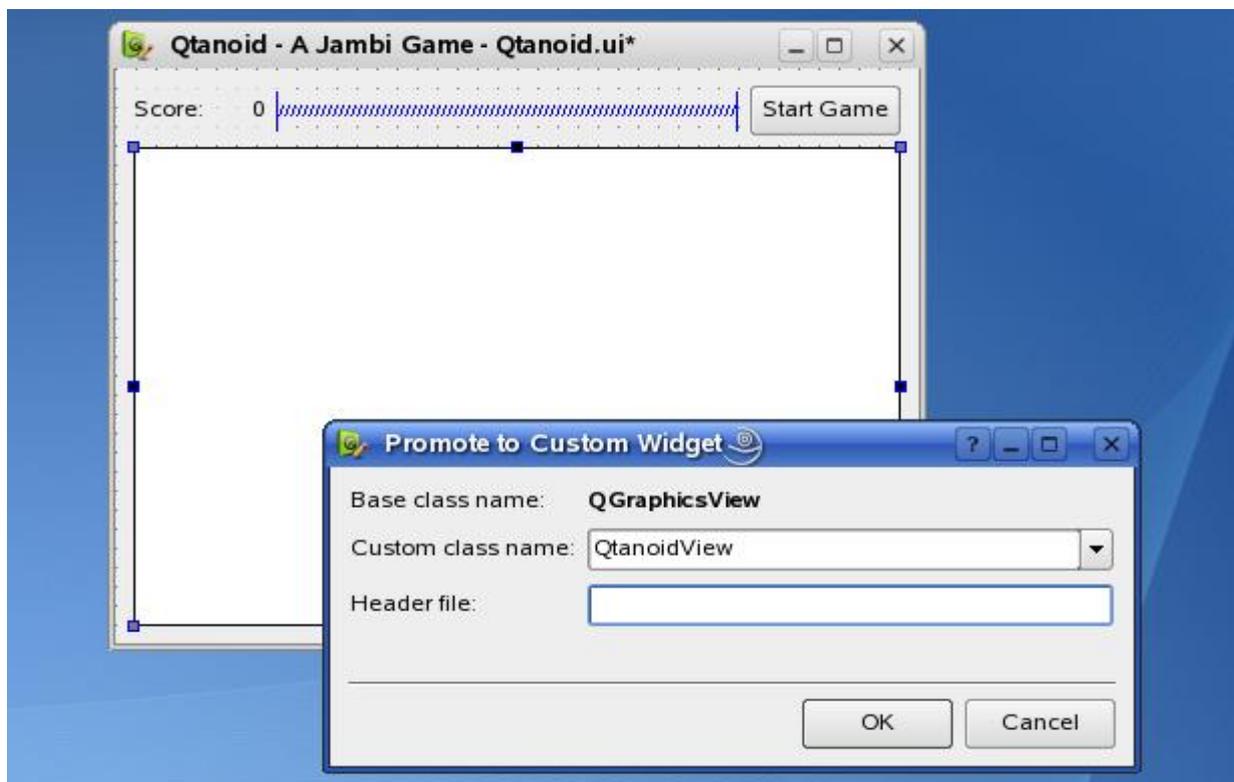
La version actuelle Qt Jambi est le troisième aperçu de la technologie. Il est construit sur Qt 4.2, offrant un accès à de nouvelles fonctionnalités comme Qt Graphics View cadre que nous allons utiliser lors de l'écriture de notre match.

Création d'une interface utilisateur avec Qt Designer

L'interface utilisateur du jeu *Qtanoid* est créé en utilisant Qt Designer , mais il est tout à fait possible d'écrire l'interface utilisateur entière à la main en Java, comme nous pouvons avec C ++. L'interface utilisateur contient trois éléments différents, une étiquette indiquant le score, un bouton pour démarrer le jeu, et un écran où toute l'action se passe.

L'affichage est mis en œuvre par une sous-classe de la QGraphicsView classe. Lors de l'utilisation de Qt Designer pour créer notre interface utilisateur, ce widget personnalisé n'est pas disponible pour nous. Mais il est encore possible d'inclure le widget dans la définition de l'interface utilisateur par "promouvoir" l'un des widgets régulièrement de Qt comme un espace réservé à cet effet sur le formulaire que nous créons dans Qt Designer .

Ainsi, lors de la conception de l'interface utilisateur, nous avons mis un QGraphicsView un widget sur le formulaire, puis nous ouvrons un menu contextuel au-dessus et cliquez sur **Promouvoir Custom Widget**. Dans la boîte de dialogue qui s'ouvre, nous précisons le nom de notre sous-classe: QtanoidView dans ce cas.



Lorsque JUIC est exécuté, il génère du code pour un objet QtanoidView la place de l'espace réservé QGraphicsView widgets. C'est un moyen très pratique de mettre des widgets personnalisés dans les interfaces utilisateur aussi longtemps que vous n'avez pas besoin de les

pré visualiser, signal d'établissement et les connexions à sous, ou de modifier les propriétés de votre coutume widgets. Il vous permet également de concevoir une interface utilisateur qui intègre une composante qui n'a pas encore été écrite.

II-2-6-Avantages :

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 avec
- 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 Trolltech Qt et de livrer les principales applications
- Son principal produit est la bibliothèque multiplate-forme Qt développée en C++ qui est notamment à la base de l'environnement graphique libre KDE. *Trolltech* est également à l'origine d'une version pour systèmes embarqués de Qt : Qt Extended (anciennement Qtopia). Qt est également accessible à partir de Java avec le binding Qt Jambi.

II-3- Sofi Framework: [21] [22] [23]

II-3-1-Introduction:

Fondé en 2003, **SOFI** fait aujourd'hui un pas de géant en devenant logiciel libre. Il se métamorphose en une plateforme de développement offert sous forme de service (PAAS) via les nuages (Cloud Computing). Voici le fruit de tout l'investissement fait par **Nurun** et ses partenaires au cours des dernières années afin de faciliter de développement Web transactionnel. Maintenant basé sur les plateformes **Google Apps** et **JIRA Studio**, voici **SOFI Framework** et son infrastructure (**SOFI Infrastructure**) offert à tous.

II-3-2-Définition:

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.

II-3-3-Evolution (développement):

Voici les faits saillants de l'évolution de SOFI :

- 2003 : Naissance du cadre sous le nom de code : **Extension Framework**
- 2004 : 1er partenaire et baptême officiel au nom de SOFI
- 2005 : Évolution constante (sous forme de versions de maintenance) grâce aux nombreux projets dans les organisations ayant choisi SOFI comme plateforme de développement
- 2006 : Lancement de la communauté SOFI avec le site Internet du Portail SOFI et nomination aux Octas sous la catégorie **Réussite Commerciale**
- 2007 : Lancement de **SOFI 2.0** et ajout d'un forum de discussion sur le Portail SOFI
- 2008 : Après de nombreuses migrations d'applications de SOFI 1.x à SOFI 2.0, une importante version de maintenance est offerte (version 2.0.2), suivie d'une version 2.0.3 qui inclut beaucoup plus de documentation

- 2009 : Nurun introduit officiellement SOFI Framework en tant que logiciel libre
- 2011 : **22 février**, lancement de SOFI logiciel libre avec les produits **SOFI Framework** et **SOFI Infrastructure**, d'un Portail et d'une plateforme de développement offerte sous force de service (PAAS).

II-3-4-Principe d'utilisation:

- Un développement de système Web avec SOFI se base tout d'abord sur une infrastructure technologique. Des applications développées avec SOFI existent présentement en production sur des serveurs Windows, Linux et AIX. Cette possibilité assure aux clients de SOFI une indépendance du fournisseur technologique.
- Le développement est également basé sur la norme Java Enterprise Edition (JEE) de base, c'est-à-dire un serveur d'application tel qu'Oracle Application Server, BEA Weblogic, Websphere, JBoss et autres. Encore une fois, les applications SOFI sont indépendantes de la couche serveur utilisée, permettant ainsi un déploiement sur différentes plateformes.
- Les applications SOFI s'appuient ensuite sur des cadres d'applications techniques matures et populaires. Il s'agit de logiciels libres respectant les meilleures pratiques du marché.
- La couche SOFI apparaît ensuite. L'abstraction des cadres d'applications et l'importante quantité de composants basés directement sur l'infrastructure Java EE (Enterprise Edition) sont uniques et exclusifs à SOFI.
- Côté sécurité, SOFI offre une solution d'authentification unique intégrée avec Microsoft Active Directory, Oracle SSO ou avec une application d'authentification unique maison développée par une organisation.
- Les solutions Web utilisent principalement la couche SOFI afin d'offrir leur services. C'est une solution clé en main, que ce soit pour développer la logique d'affaires ou les composants d'interfaces. À l'aide des composants SOFI, il est possible d'offrir des systèmes de grande qualité. En effet, tous les composants d'interfaces de SOFI respectent le **Web 2.0**.
- Toute l'architecture de SOFI comprend des patrons de conception et intègre les bonnes pratiques du marché qui demeurent toujours au centre des considérations lors du développement. Le but premier de SOFI est de ne pas réinventer la roue.
- En tant que cadre d'applications fonctionnel, SOFI est basé sur plusieurs cadres d'applications provenant en grande majorité de la communauté de logiciels libres Java. Il bénéficie donc de l'expertise de milliers de programmeurs Java à la grandeur de la planète. De plus, SOFI est supporté par tous les serveurs d'applications J2EE. Il est donc simple de déployer sur des plateformes telles qu'**Oracle Applications Serveur, IBM WebSphere, BEA, Tomcat ou JBoss**.

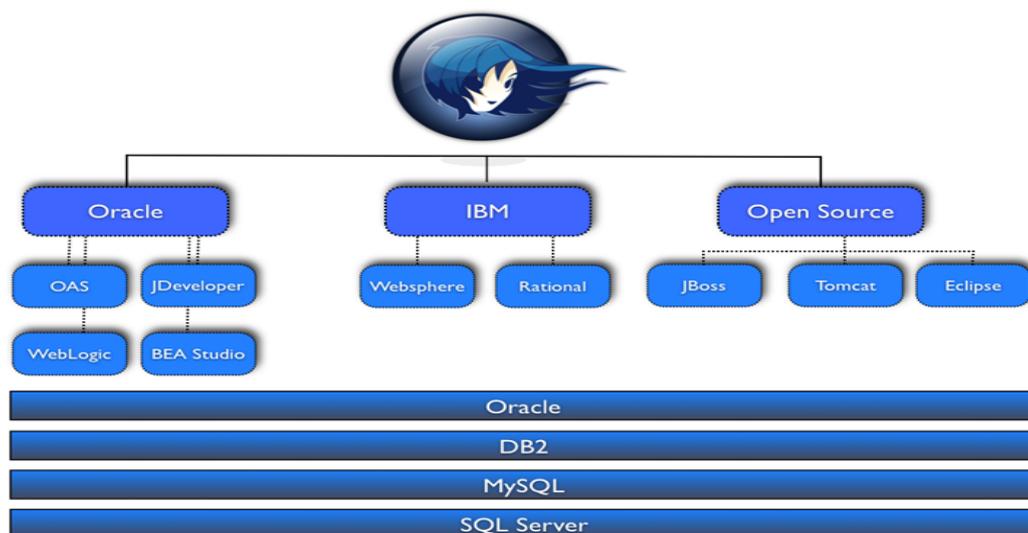


Figure 09 : SOFI, une solution pour toutes les plateformes et outils de développement

II-3-5-Cas d'utilisation et exemple d'application :

Cas d'utilisation :

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.

Exemple d'application :

Exemple d'utilisation appliqué à une liste de valeurs

Il est aussi possible d'utiliser le `<sofi-html:link />` pour rafraîchir des champs suite à une sélection dans une liste de valeur. Dans l'exemple ci-dessous, nous rafraîchissons l'adresse et le courriel de l'utilisateur lors d'une nouvelle sélection.

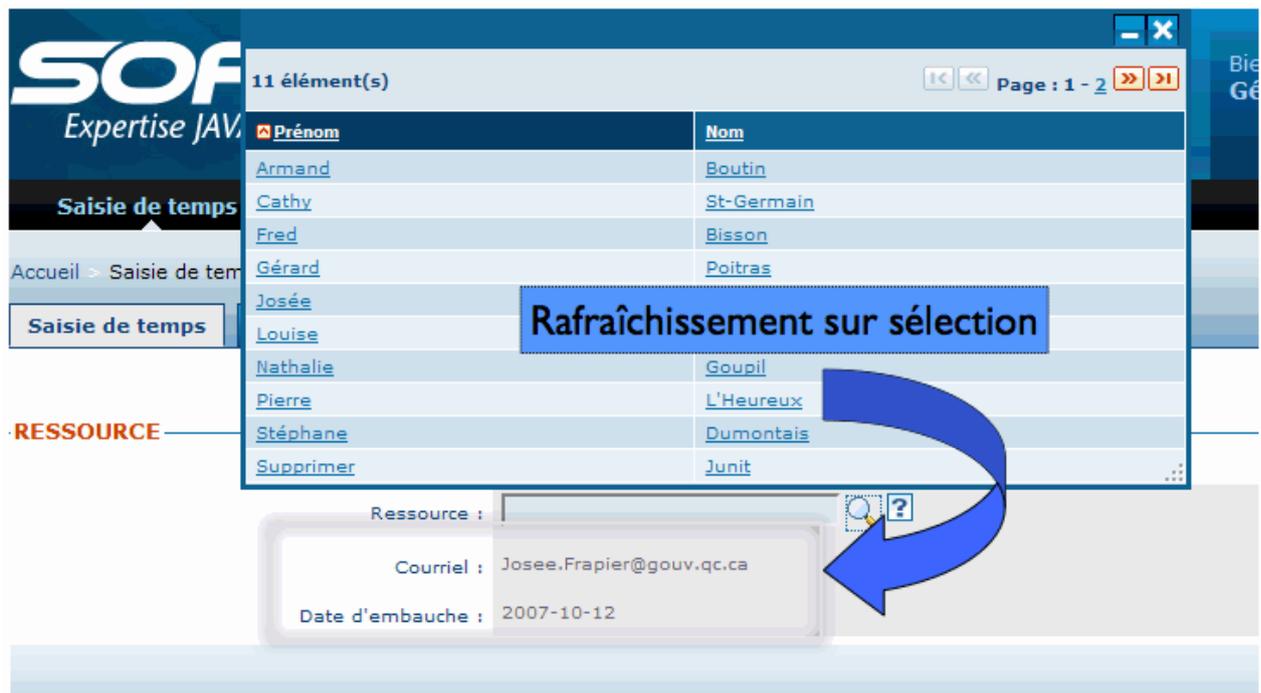


Figure 10: `Rafraichissement sur sélection d'une liste de valeurs

II-3-6-Avantages :

Les avantages de l'utilisation 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, tant fonctionnel qu'organique** 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 journalisation
- la sécurité
- la gestion des tâches asynchrones
- etc.
- 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 la 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
 - Aucune étiquette

II-5- Spring Framework: [1] [2] [4] [5] [14] [15] [16]

II-5-1-Introduction:

Le framework Spring fournit un modèle de programmation et la configuration globale pour les applications d'entreprise modernes basées sur Java - sur n'importe quel type de plate-forme de déploiement. Un élément clé du printemps est un soutien infrastructurel au niveau de l'application: Printemps met l'accent sur la «plomberie» des applications d'entreprise afin que les équipes puissent se concentrer sur la logique métier de niveau de l'application, sans liens inutiles aux environnements de déploiement spécifiques.

Le framework Spring est un framework vraiment très puissant qui peut ajouter beaucoup de fonctionnalités à vos applications Java.

II-5-2-Définition:

Spring 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 Spring simple est à la portée de tous, avoir une architecture Spring 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 Spring, 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.

II-5-3-Evolution (développement):

La communauté Spring Source et leur projet Spring Framework sont encore jeunes (moins de 10 ans aujourd'hui) mais sont déjà adoptés par de nombreuses personnes et par de nombreuses entreprises (tel que Adobe, Google,...).

- **2004** : Création de par Rob Johnson et création de la version 1 de Spring Framework
- **2006** : Version 2 de Spring Framework
- **2008** : Rachat de Spring Source par VMWare et sortie de la version 3 de Spring Framework
- **A l'écriture de ces lignes** : Version 3.2 de Spring Framework
- Spring Roo est un outil de SpringSource créé en 2009 et dont la version actuelle est la version 1.2.2

II-5-4-Principe d'utilisation:

La première chose à savoir est que le cœur de Spring Framework 3 est composé d'un ensemble d'une vingtaine de modules qui sont regroupés en plusieurs grandes parties :

- Spring Core Container : regroupe les modules de base pour mettre en œuvre le conteneur
- AOP and Instrumentation : permet de mettre en œuvre l'AOP
- Data Access/Integration : regroupe les modules d'accès aux données
- Web : regroupe les modules pour le développement d'applications web
- Test : propose des fonctionnalités pour les tests automatisés avec Spring

La méthode recommandée pour commencer à utiliser `spring-framework` de votre projet est un système de gestion de la dépendance - l'extrait ci-dessous peut être copié et collé dans votre construction. Besoin d'aide? Voir nos guides de démarrage sur la construction avec Maven et Gradle .

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>4.0.2.RELEASE</version>
  </dependency>
</dependencies>
```

Spring Framework inclut un certain nombre de modules différents, ici nous montrons `spring-context` qui fournit les fonctionnalités de base. Reportez-vous aux guides de démarrage sur le droit pour d'autres options.

Une fois que vous avez configuré votre compilation avec le `spring-context` dépendance, vous serez en mesure de faire ce qui suit:

```
hello/MessageService.java
```

```
package hello ; public interface MessageService { String getMessage () ; }
```

```
hello/MessagePrinter.java
```

```
package hello ; import org.springframework.beans.factory.annotation.Autowired ; import
org.springframework.stereotype.Component ; @Component public class MessagePrinter { @Autowired private
MessageService service ; public void printMessage () { System . out . println ( this . service . getMessage ()); } }
```

```
hello/Application.java
```

```
package hello ; import org.springframework.context.ApplicationContext ; import
org.springframework.context.annotation.* ; @Configuration @ComponentScan public class Application {
@Bean MessageService mockMessageService () { return new MessageService () { public String getMessage () {
return "Hello World!" ; } } ; } public static void main ( String [] args ) { ApplicationContext context = new
AnnotationConfigApplicationContext ( Application . class ); MessagePrinter printer = context . getBean (
MessagePrinter . class ); printer . printMessage (); } }
```

L'exemple ci-dessus montre le concept de base d'injection de dépendance, la `MessagePrinter` est découplé du `MessageService` mise en œuvre, le câblage de Spring Framework tout ensemble.

II-5-5-Cas d'utilisation et exemple d'application :

Compléments :

Divers modules Spring 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 frameworks de test comme JUnit ou TestNG

II-5-6-Avantages :

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 Spring est considéré comme un standard de fait.

Plus léger que les implémentations standards JEE (Java Enterprise Edition), Spring reste la préférence tant des architectes techniques que des développeurs.

La modularité du Framework Spring permet de répondre à énormément de besoins. Basé sur le pattern IoC, possibilité d'intégrer des services en plug-and-play. S'intègre avec plusieurs technologies cotées Vue.

III-La comparaison entre les différents type de Framework :[22]

III-1-Information sur chaque Framework :

Dans le tableau ci-dessous, nous avons choisi les principaux critères de comparaison entre les différents Framework java:

Projet	langage	Date de début	Version stable courante	Date de publication	Type de Framework	Date de mise a jour	licence
Spring	java	2004	3.1.2	2004	Framework d'application d'entreprise	09/07/2012	Apache
Wavemaker	Java standard	07/09/2008	6.3.3	10/12/2009	Framework d'application web	22/09/2011	Apache
Qt jambi	C++/java	1991	4.4.3	1994	Framework d'application	28/01/2008	BSD
Sofi	java	2003	2.0.3	2004	Framework d'application web	2007	Apache tomcat

Tableau 01 : Comparaison des Framework

III-2-comparaison des fonctionnalités :

Dans le tableau ci-dessous, nous avons choisi les principaux critères de comparaison entre les différentes fonctionnalités des Framework java:

projet	ajax	Mvc framework	Mvc push/pull	Internationalisation	ORM	Testing Framework	DB migration Framework	Security Framework	Template Framework	Caching Framework	Form validation Framework
spring	oui	Oui	Push	Oui	Hebernatismore	Mock objects tests unitaire		Spring sécurité(ex-Acegi)	Jsp,commons,tiles,velo,city,thymeleaf,more	Ehcache more	Commons validation, bean validation
wavemaker	Dojo toolkit	Oui	Push	Dojo toolkit	hibernate	junit	hibernate	Spring sécurité(ex-Acegi) contrôle d'accée a base de role	Dojo toolkit	Dojo toolkit	Regular expression, schema dreven validation
Qt jambi	jQuery	Page oriented	Pull	Oui	Hibernate, Cayenne	Oui		pluggable	Velocity, JSP	Cached templates	Built-in validation
Sofi	jQuery	Oui	Push-pull	Oui	Entity Engine(OOTB ORM)	JUnit	Entity Engine Tools, Data File Tool, CSV Parser, Apache POI	OOTB Security framework	Freemarker(Recommended), Velocity(Support Available), JSP(Support Available)	OOTB Cache Maintenance	Server side validation, Client Side Validation(jQuery)

Tableau 02 : Comparaison des Fonctionnalités des Framework

IV-Choix du Framework pour j2ee : [14] [15] [16] [22]

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

Spring est un framework plus récent, qui n'est pas adapté pour les débutants, sauf à se faire des maux de cranes très souvent. Les 'pros' te le conseilleront sûrement car il utilise des concepts novateurs, et ça fait vendre des formations.

Spring est bon pour des petites applications, mais pour des applications de grande envergures, il est conseillé de travailler directement avec les EJB.

De façon générale, je préconiserai de bien savoir ce que l'on veut pour éviter la chasse au buzzword : un nouveau framework viendra toujours remplacer le plus couru du moment, reste à savoir si on peut vivre dans la durée avec les choix réalisés. Je dis notamment cela par rapport au passage où il est question de "framework de pointe" : c'est un critère bien dur à décrire précisément.

les raisons de notre choix en insistant par exemple sur :

- les outils qui facilitent leurs usages,
- la complexité d'utilisation,
- la documentation,
- la communauté,
- la pérennité.

le Framework Spring 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.

➤ Conteneur léger et inversion de contrôle

La prise en charge des objets par Spring.

La gestion du cycle de vie des objets confiés à Spring (pattern Singleton, Prototype).

La programmation par interface et le couplage faible (pattern Strategy).

L'injection de dépendance.

Présentation de l'injection par setter et par constructeur, quel type d'injection dans quel cas ?

L'accès programmatique au conteneur.

La configuration par annotations.

- .Présentation de Spring Security pour la gestion de l'authentification et des autorisations.

Création, chargement et récupération d'un contexte de sécurité.

Activation d'un filtre.

Création d'un formulaire de login.

Sécurisation des méthodes.

Application d'une stratégie de sécurité par aspects, par annotations.

V-Conclusion :

Dans ce chapitre, nous avons fait une étude sur les Framework java j2ee et nous avons présenté quatre types de Framework et sont (spring ,wavemaker, Qt jambi, sofi), on a étudié chaque un des Framework de la même façon selon les mêmes critères .

Ce chapitre nous a permis de prendre connaissance des différents types de Framework et leurs avantages.

Le but de ce chapitre est de choisir un Framework avec lequel on va travailler et qui correspond à notre réalité et c'est le spring Framework.

Le chapitre qui suit est l'analyse & la conception.

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.

Nous allons entamer ce processus par une analyse qui mettra en évidence les différents acteurs intervenants dans le système ainsi que leurs besoins. La phase de conception va s'appuyer sur les résultats de la phase d'analyse et qui donnera la modélisation des objectifs à atteindre, pour cela nous avons porté notre choix sur le langage UML conçu pour la spécification, la visualisation, la documentation et la construction des systèmes à fortes composantes logiciels.

II. Analyse :

Dans cette partie nous allons spécifier d'une manière bien détaillée et claire l'application des services moyens généraux de centre de distribution carburant d'Oued_Aissi. Pour cela nous allons procéder en premier lieu à la détermination d'une manière générale de ce qui se trouve dans le champ de l'application.

II.1. Définition des besoins :

Notre projet porte sur la mise au point d'une application client/serveur pour l'automatisation des procédures de la gestion du service moyens généraux.

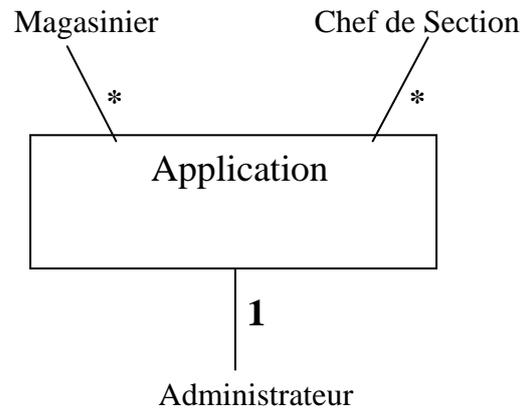
L'application assure un environnement interactif afin de rendre les tâches de gestion accessibles aux différents acteurs et ce via réseau local, afin d'assurer les procédures associées à la maintenance au sein de l'entreprise NAFTAL.

II.2. Identification des acteurs :

Pendant le stage que nous avons effectué au sein de la section économat de CDS de Oued_Aissi, nous avons procédé à l'identification des principaux acteurs qui seront les futurs utilisateurs de système, ces acteurs sont :

- **Le chef de section :** il gère les commandes, les bon de livraisons, les factures, Consultation des bons de d'entrés magasin (BEM) et les bons de transfert magasin (BTM).
- **Le magasinier :** il gère les bons d'entrés magasin (BEM), les bons de sortie magasin (BSM), les bons de transfert magasin (BTM).
- **L'administrateur :** C'est le maître d'ouvrage ; il gère les comptes utilisateurs et les produits.

Diagramme de contexte :



II.3. Identification des espaces :

À chaque utilisateur du système est attribué un espace qui regroupe toutes les tâches qu'il peut effectuer. Après l'analyse de notre système on est arrivé à l'identification de cinq espaces qui sont :

- Espace Chef de section.
- Espace Magasinier.
- Espace Administrateur.

II.4. Spécification des tâches :

Acteur	Tâches
Le chef de section	T1.S'authentifier. T2.Modifier le compte. T3.Comparé les factures préforma. T4.Géré les bon de commande. T5.Géré les bon de livraison. T6.Géré les facture. T7.Consultation du bon d'entré magasin. T8.Consultation du bon de transfert magasin.
Le magasinier	T9.S'authentifier. T10.Modifier le compte. T11.Géré les bon d'entré magasin. T12.Géré les bon de sortie magasin. T13.Géré les bon de transfert magasin. T14.Géré les demande de fourniture.
L'administrateur	T15.S'authentifier. T16.Géré les comptes utilisateurs. T17.Géré les produits.

Tableau4.1 : Spécification des tâches

II.5. Spécification des scénarios :

II.5.1. Définition d'un scénario :

Un scénario est une instance de cas d'utilisation, tout résultat possible de l'exécution d'un cas d'utilisation est appelé scénario. Les scénarios doivent être décrits par les utilisateurs eux même, chacune des tâches effectuées par un ou plusieurs acteurs sera décrite par un ensemble de scénarios.

➤ **Remarque :**

Vu le nombre de cas d'utilisation et de leurs scénarios, nous ne décrivons que quelques scénarios pour quelques cas d'utilisations.

Les scénarios sont représentés dans le tableau suivant :

Acteur	Tâches	Scénarios par Tâche
Le chef de section.	T1.S'authentifier.	S0.Saisir le Login et le mot de passe. S1.Valider la saisie.
	T2.Modifier le compte.	S3.Sélectionné le lien « Modifier Compte ». S4.Supprimé les donnée dans les champs Autorisé. S5.Saisir les nouvelles données. S6.Valider la saisie.
	T3.Comparé les factures préforma.	S7.Sélectionné le lien « Comparé facture préforma ». S8.Saisir les factures préforma dans le tableau. S9.Choisir une facture préforma. S10.Valider le choie.
	T4.Géré les bon de commande.	S11.Sélectionné le lien « Bon de commande ». S12.Sélectionné le lien « Ajouter ». S13.Remplir les champs du bon de commande. S14.Valider la saisie.
	T5.Géré les bon de livraison.	S15.Sélectionné le lien « Bon de livraison ». S16.Sélectionné le lien « Ajouter ». S17.Remplir les champs du bon de livraison. S18.Valider la saisie.
	T6.Géré les facture.	S19.Sélectionné le lien « Facture ». S20.Sélectionné le lien « Ajouter ». S21.Remplir les champs de la facture. S22.Valider la saisie.

	T7.Consultation du bon d'entré magasin.	S23.Sélectionné le lien « Consultation ». S24.Sélectionné le lien « Bon d'entré magasin ». S25.Valider le document.
	T8.Consultation du bon de transfert magasin.	S26.Sélectionné le lien « Consultation ». S27.Sélectionné le lien « Bon de transfert magasin ». S28.Valider le document.
Le magasinier	T9.S'authentifier.	S29.Saisir le Login et le mot de passe. S30.Valider la saisie.
	T10.Modifier le compte.	S31.Sélectionné le lien « Modifier Compte ». S32.Supprimé les donnée dans les champs Autorisé. S33.Saisir les nouvelles données. S34.Valider la saisie.
	T11.Géré les bon d'entré magasin.	S35.Sélectionné le lien « Bon d'entré magasin ». S36.Sélectionné le lien « Ajouter ». S37.Remplir les champs du bon d'entré magasin. S38.Valider la saisie.
	T12.Géré les bon de sortie magasin.	S39.Sélectionné le lien « Bon de sortie magasin ». S40.Sélectionné le lien « Ajouter ». S41.Remplir les champs du bon de sortie magasin. S42.Valider la saisie.
	T13.Géré les bon de transfert magasin.	S43.Sélectionné le lien « Bon de transfert magasin ». S44.Sélectionné le lien « Ajouter ». S45.Remplir les champs du bon de transfert magasin. S46.Valider la saisie.
	T14.Géré les demande de fourniture.	S47.Sélectionné le lien « Bon de la demande de fourniture ». S48.Sélectionné le lien « Ajouter ». S49.Remplir les champs du bon de la demande de fourniture. S50.Valider la saisie.
		T15.S'authentifier.
L'administrateur	T16.Géré les comptes utilisateurs.	S53.Sélectionné le lien «Géré les comptes ». S54.Choisir l'opération a effectué. S55.Entré les donnée. S56.Valider la saisie.

	T17.Géré les produits.	S57.Sélectionné le lien «Géré les produits ». S58.Choisir l'opération a effectué. S59.Entré les donnée. S60.Valider la saisie.
--	------------------------	---

Tableau 4.2: Les scénarios des cas d'utilisation

II.6. Diagramme des cas d'utilisation général :

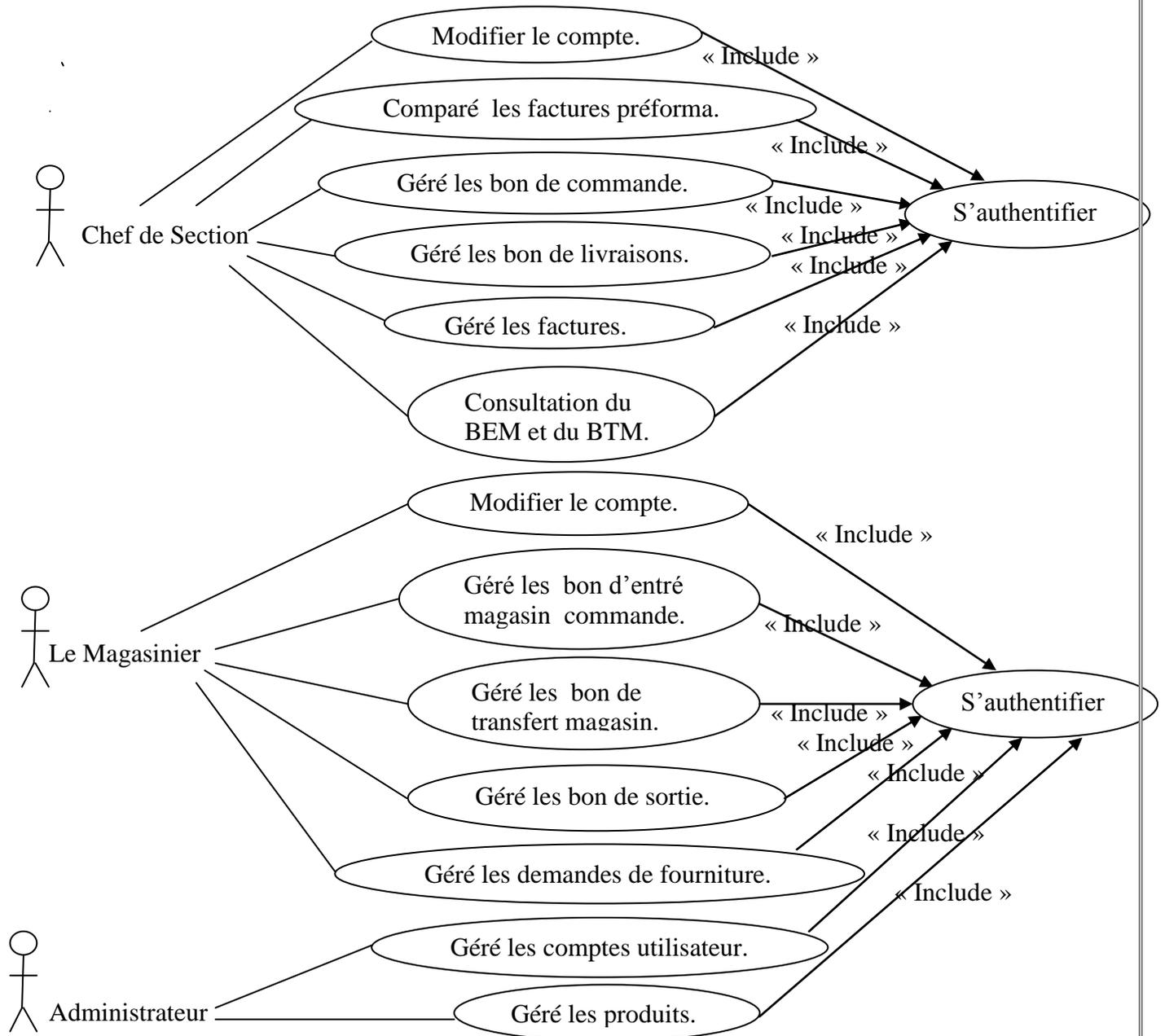


Figure 11 : Diagramme des cas d'utilisation général.

II.7. Diagrammes de quelques cas d'utilisations :

II.7.1. Diagramme détaillé du cas d'utilisation «Géré les Bon de Commande » :

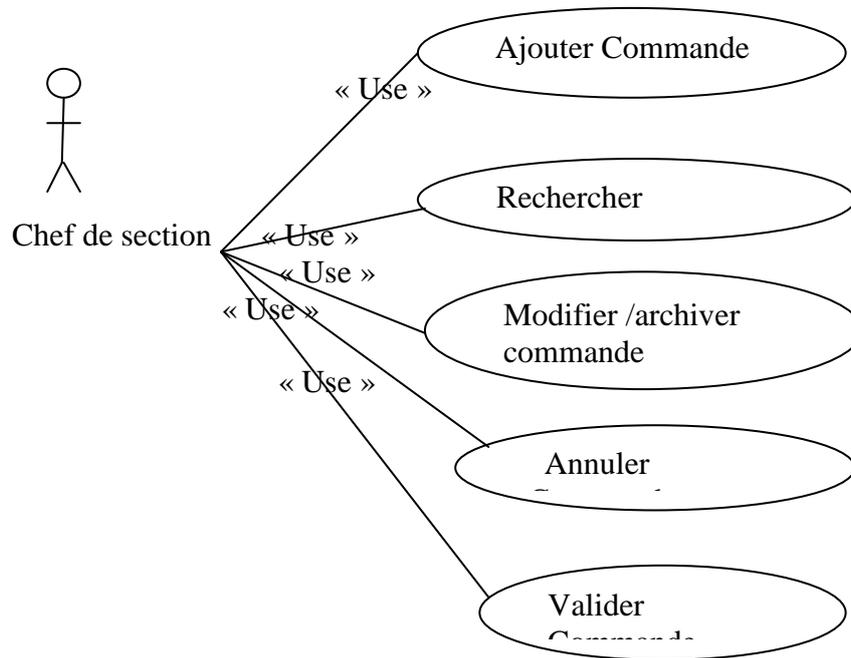


Figure 12 : Diagramme détaillé du cas d'utilisation « Géré les Bon Commande »

III.7.2. Diagramme détaillé du cas d'utilisation «Géré les Bon d'entré Magasin » :

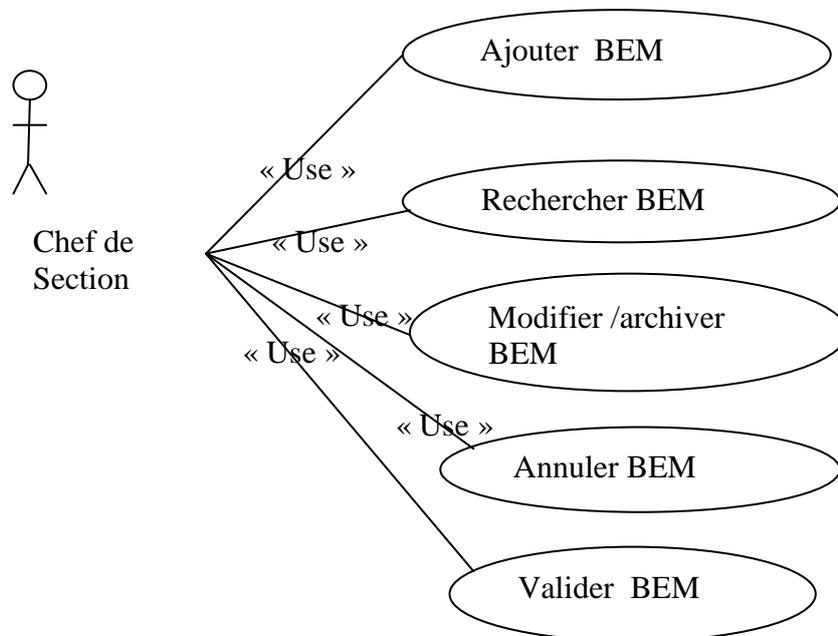


Figure 13: Diagramme détaillé du cas d'utilisation « Géré les Bon d'entré magasin »

III.7.3. Diagramme détaillé du cas d'utilisation «Modifier Profil» :

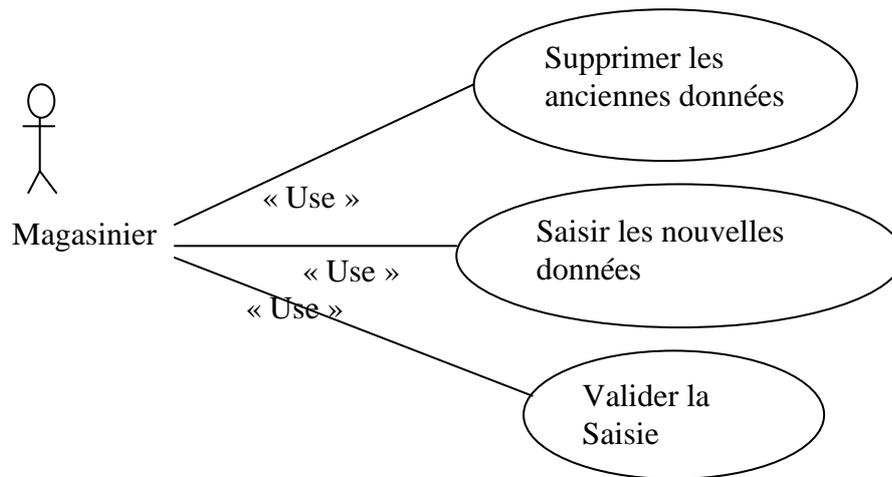


Figure 14: Diagramme détaillé du cas d'utilisation «Modifier le compte

III.7.4. Diagramme détaillé du cas d'utilisation «Géré les comptes utilisateur» :

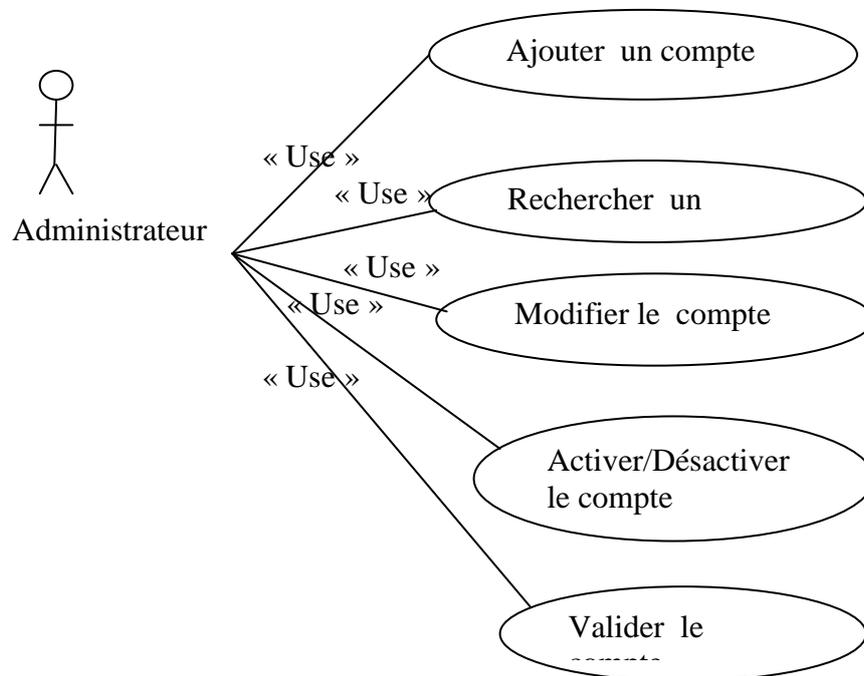


Figure 15 : Diagramme détaillé du cas d'utilisation « Géré les comptes utilisateur »

III. Conception :

III.1. La démarche de conception de l'application

Le processus de conception de notre application repose sur deux niveaux : le niveau applicatif et le niveau donné.

➤ Le niveau applicatif :

S'appuie essentiellement sur quelques diagrammes de l'extension pour les applications client/serveur du langage de modélisation UML. A cet effet, nous avons adopté la démarche suivante :

- Après l'identification des acteurs ainsi que les différentes fonctions du système à concevoir durant la partie d'analyse, nous allons mettre en évidence les cas d'utilisations mis en œuvre par les différents acteurs du système.
- Le diagramme de cas d'utilisation global est élaboré, ensuite des diagrammes de cas d'utilisation plus détaillée seront dégagés.
- A l'aide des diagrammes de séquence on formalise graphiquement le ou les scénarios qui décrivent chaque cas d'utilisation.
- Une identification des classes est fournie par la synthèse des diagrammes de séquences ainsi une liste de toutes les classes manipulées est dressée et le diagramme de classe est élaboré.

➤ Le niveau donné :

Ce niveau concerne l'organisation conceptuelle, logique et physique des données. Les données de l'application sont identifiées durant la phase analyse de notre démarche au niveau applicatif, les classes significatives seront dégagées, à ce stade. La conception de la base de données peut être élaborée.

III.2. Elaboration des diagrammes de séquences :

Le diagramme de séquence représente la succession chronologique des opérations (scénarios) réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à un autre.

III.3. Elaboration de diagrammes de classes :

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage) ou une relation organique (agrégation).

Nous obtenons donc les diagrammes suivants :

Modifier le compte Chef de section :

Diagramme de classe du cas d'utilisation « Accéder a l'espace Chef de section ».

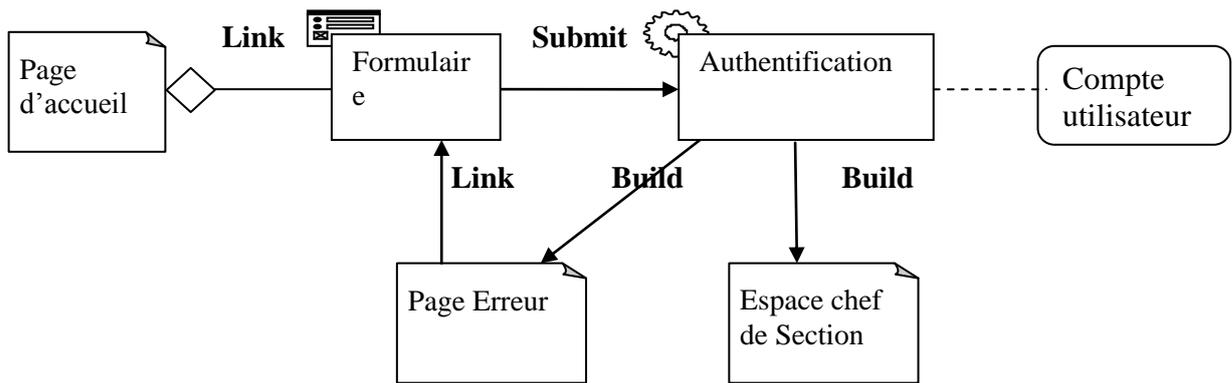


Figure 16 : Diagramme de classe du cas d'utilisation « Accéder a l'espace Chef de section »

Diagramme de classe du cas d'utilisation « Modifier le Compte Chef de section ».

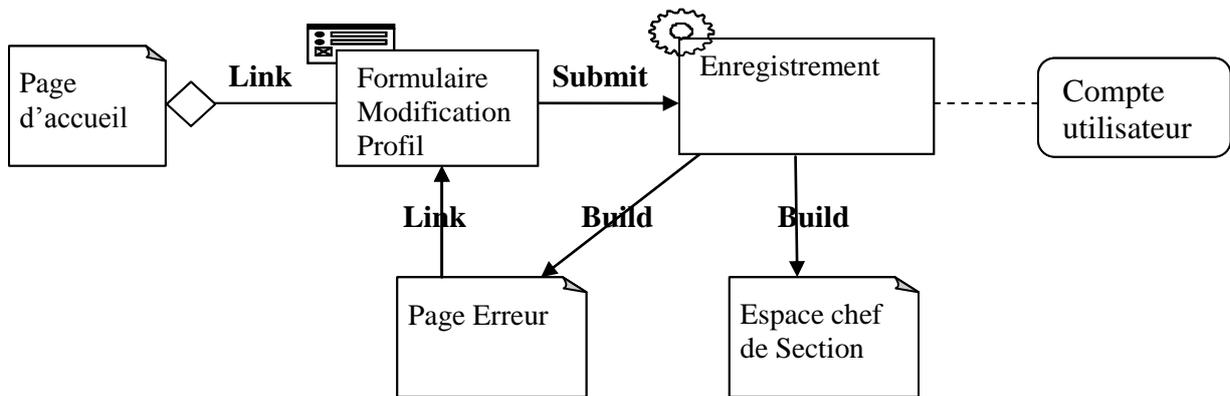


Figure 16 : Diagramme de classe du cas d'utilisation « Modifier le Compte Chef de section »

Diagramme de séquence du cas d'utilisation «Modifier le compte Chef de section »

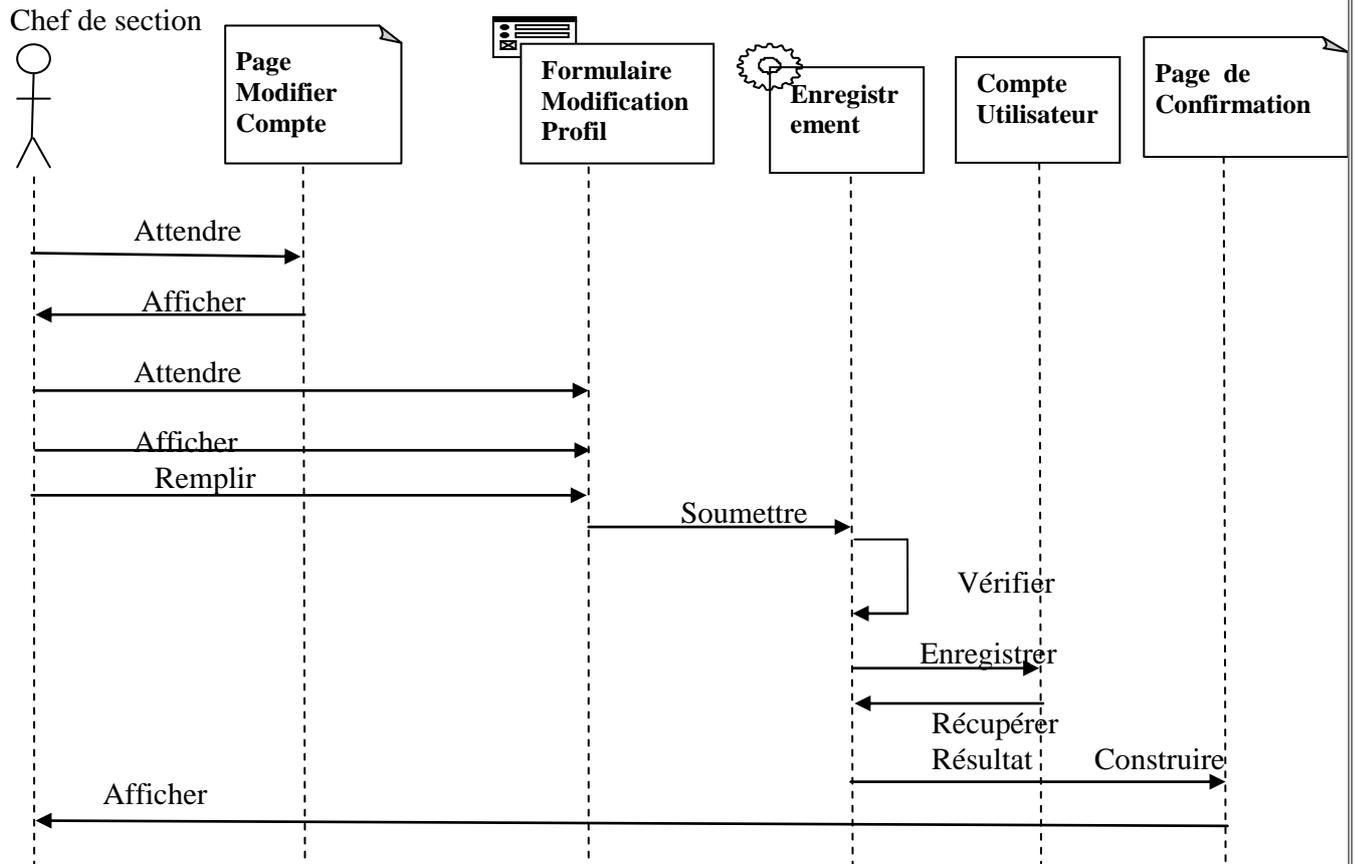


Figure17: Diagramme de séquence du cas d'utilisation «Modifier Profil Chef de section »

Diagramme de classe du cas d'utilisation «accéder à l'espace administrateur».

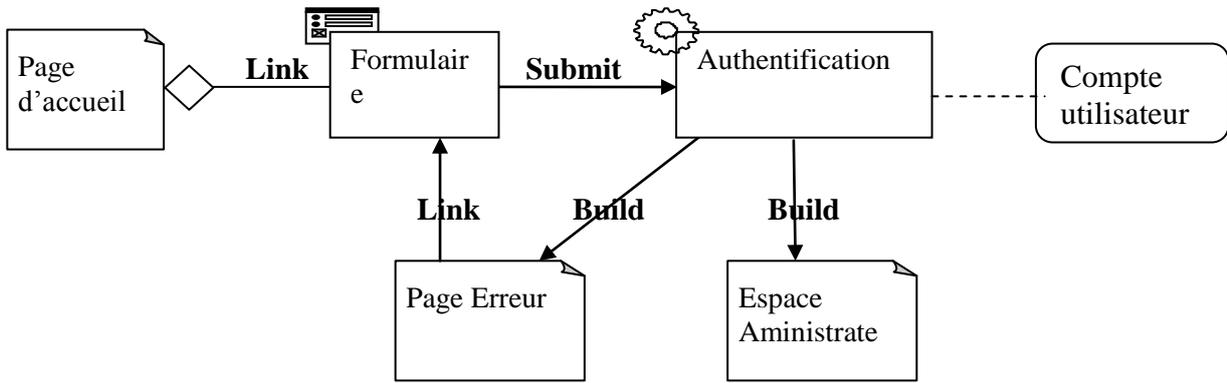


Figure 18 : Diagramme de classe du cas d'utilisation «accéder à l'espace administrateur».

Diagramme de séquence du cas d'utilisation «accéder à l'espace administrateur ».

Administrateur

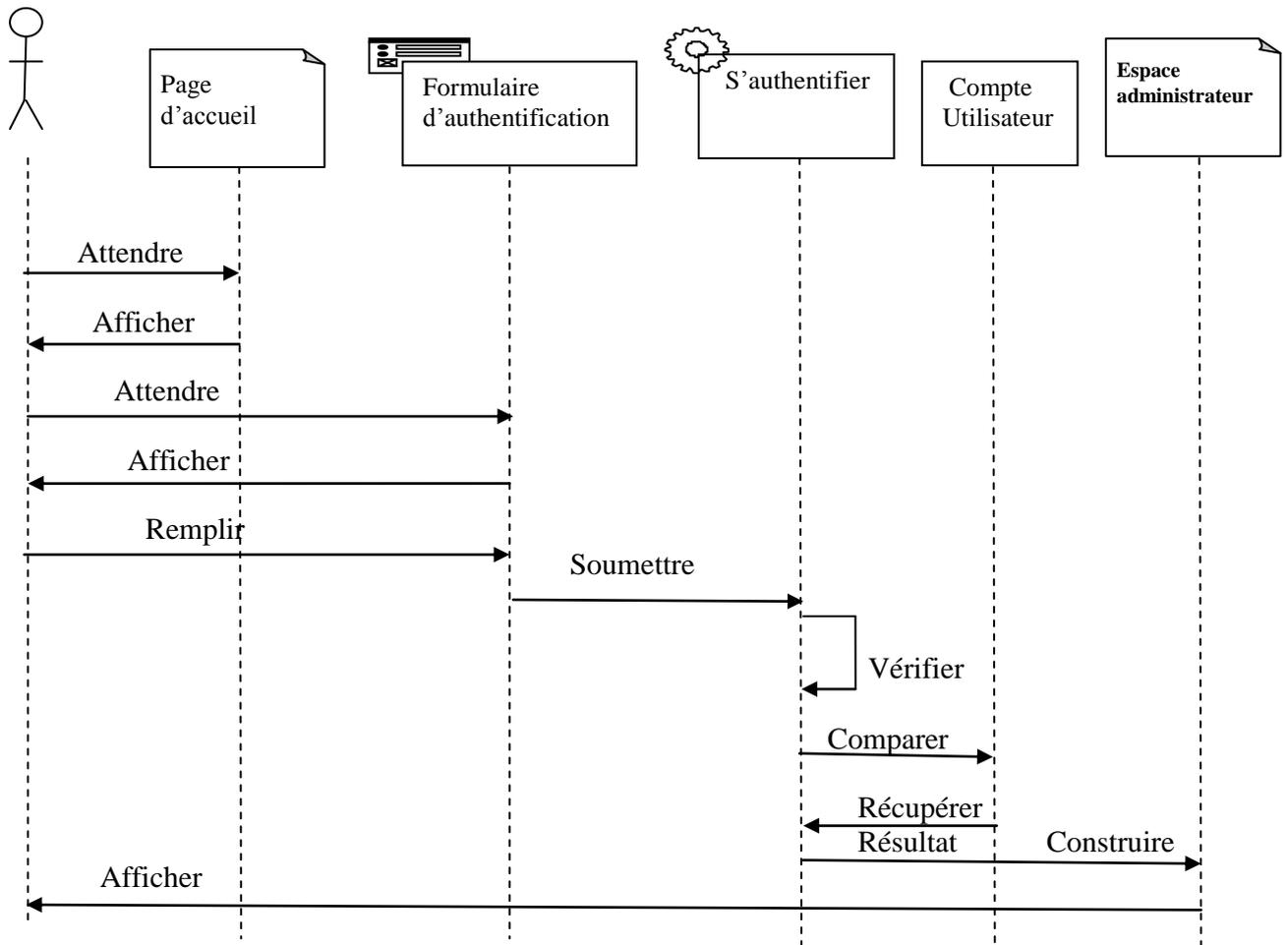


Figure 19: Diagramme de séquence du cas d'utilisation «accéder à l'espace administrateur ».

Diagramme de classe du cas d'utilisation « Ajouter un bon de commande ».

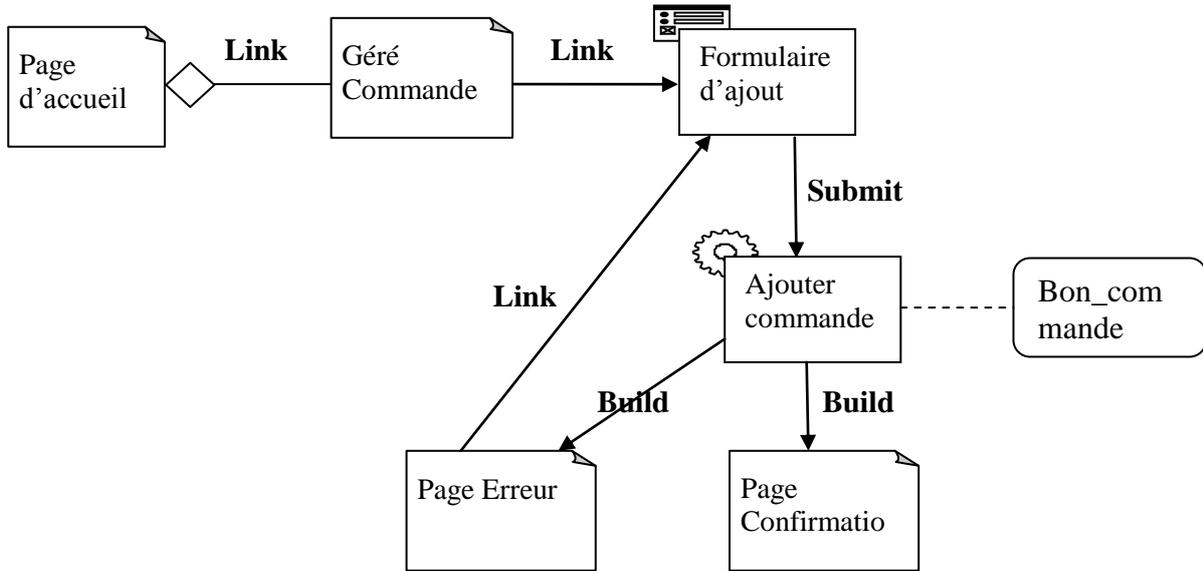


Figure 20 : Diagramme de classe du cas d'utilisation « Ajouter un bon de commande ».

Diagramme de séquence du cas d'utilisation «Géré les bon de commande ».

Chef de section

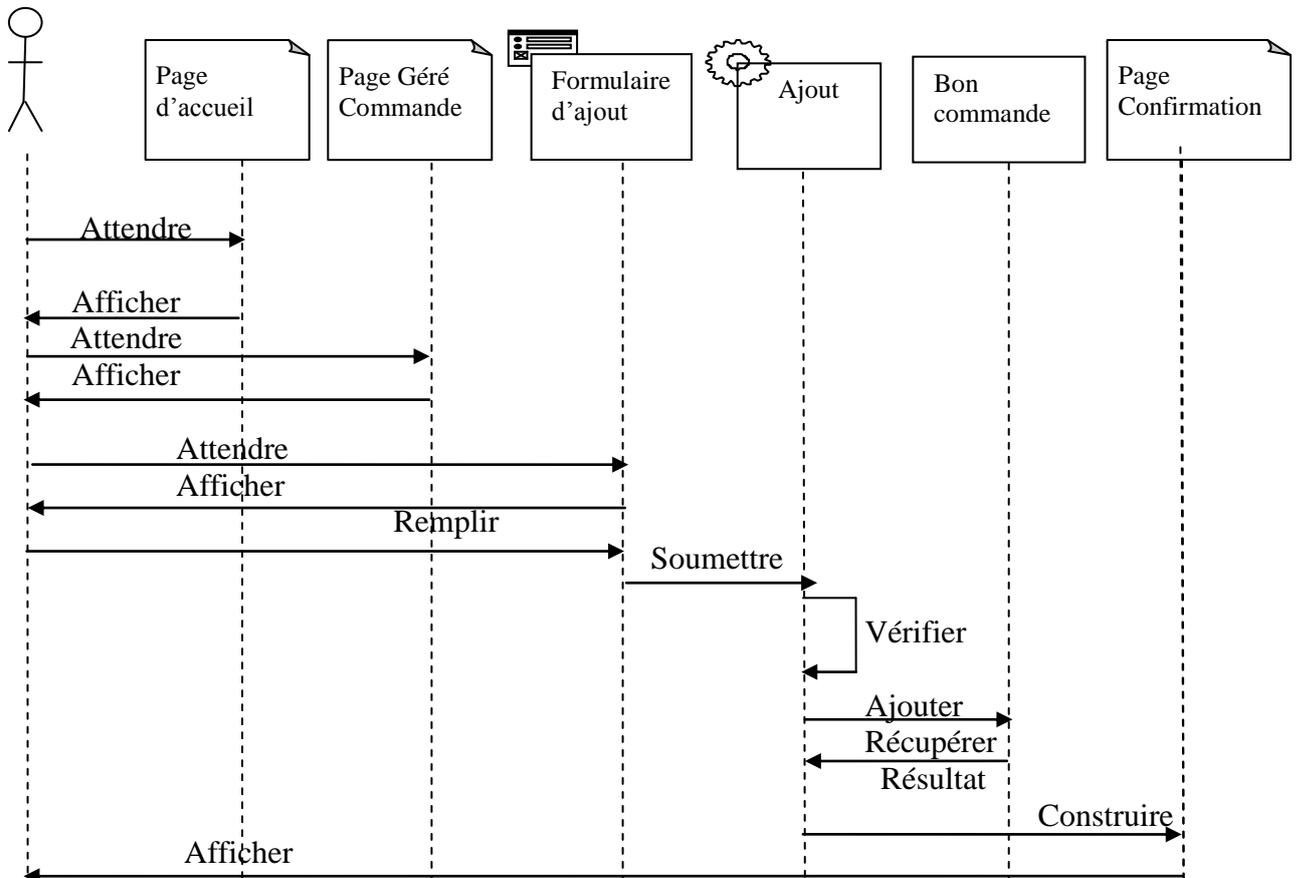


Figure 21 : Diagramme de séquence du cas d'utilisation «Géré les bon de commande ».

Diagramme de classe du cas d'utilisation «Ajouter une Facture».

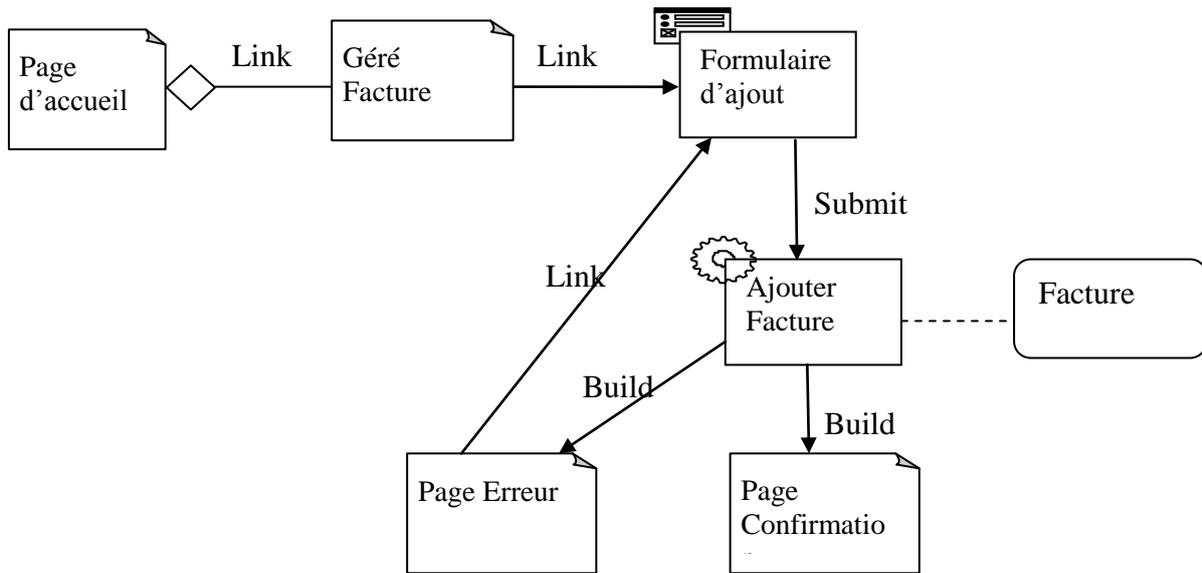


Figure 22 : Diagramme de classe du cas d'utilisation «Géré les Facture».

Diagramme de séquence du cas d'utilisation «Géré les Facture».

Chef de Section

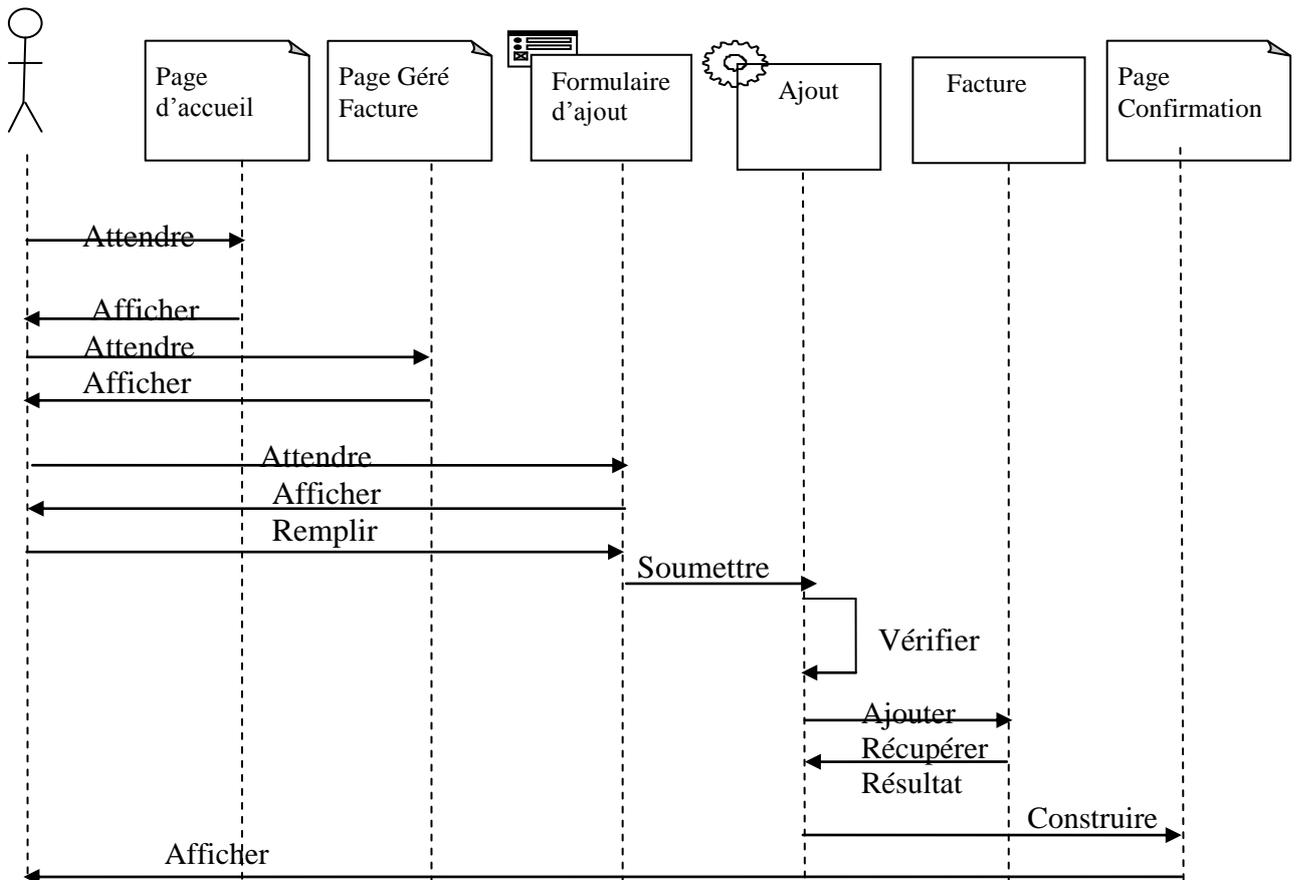


Figure 23 : Diagramme de séquence du cas d'utilisation «Géré les Facture».

III.4. Elaboration des diagrammes d'activités :

Le diagramme d'activité permet de représenter le dynamique du système d'information. Il est considéré comme une variante du diagramme d'état transition. Il est attaché à une classe, processus, acteur ou entité, à un cas d'utilisation ou bien à une opération. C'est un graphe orienté qui décrit un enchaînement traitements.

Le déroulement ainsi présenté est appelé flot de contrôle.

Diagramme d'activité du cas d'utilisation «Ajouter une commande » :

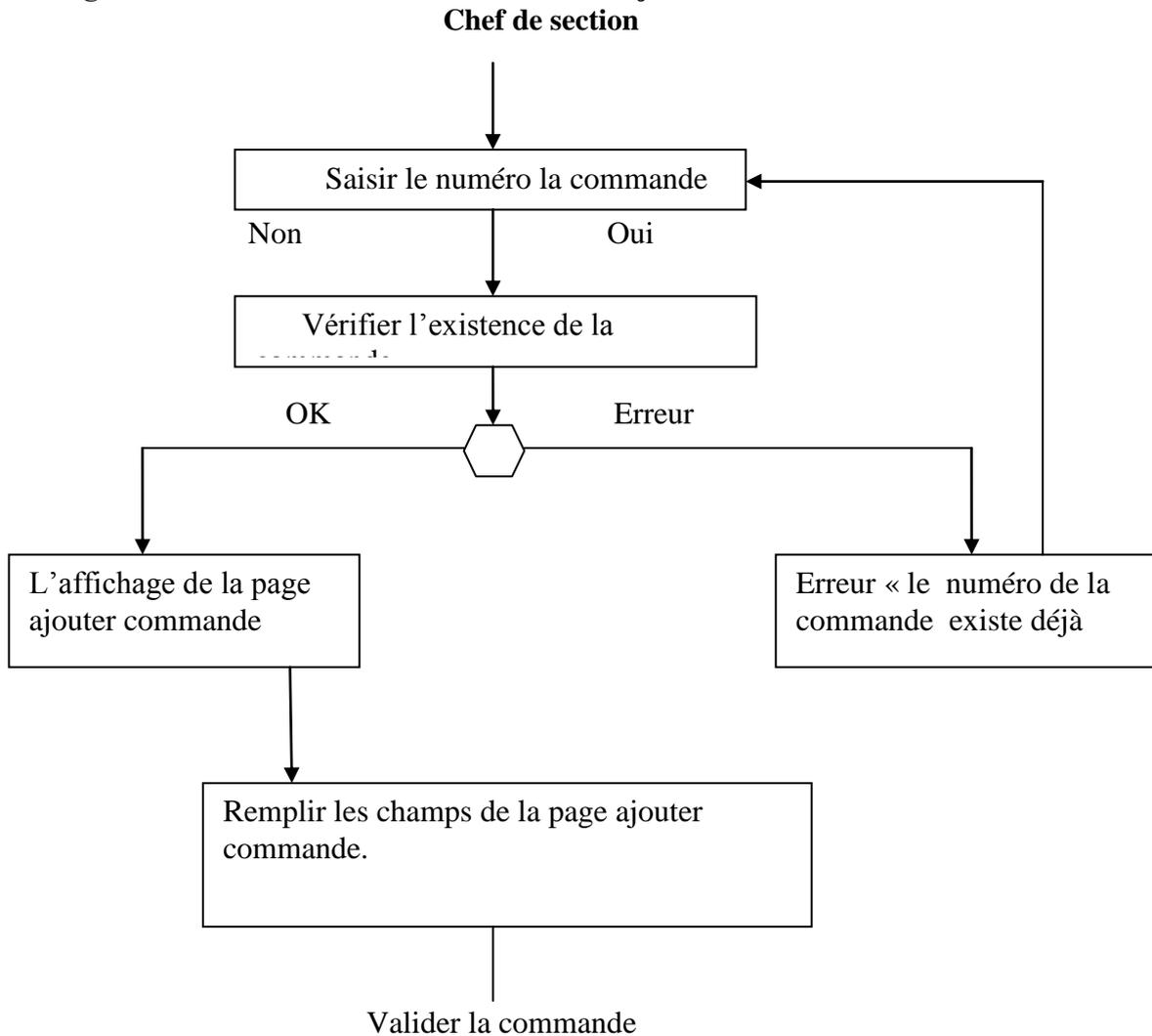


Figure 24 : Diagramme d'activité du cas d'utilisation «Ajouter une commande »

Diagramme d'activité du cas d'utilisation «Ajouter un produit » :
Administrateur

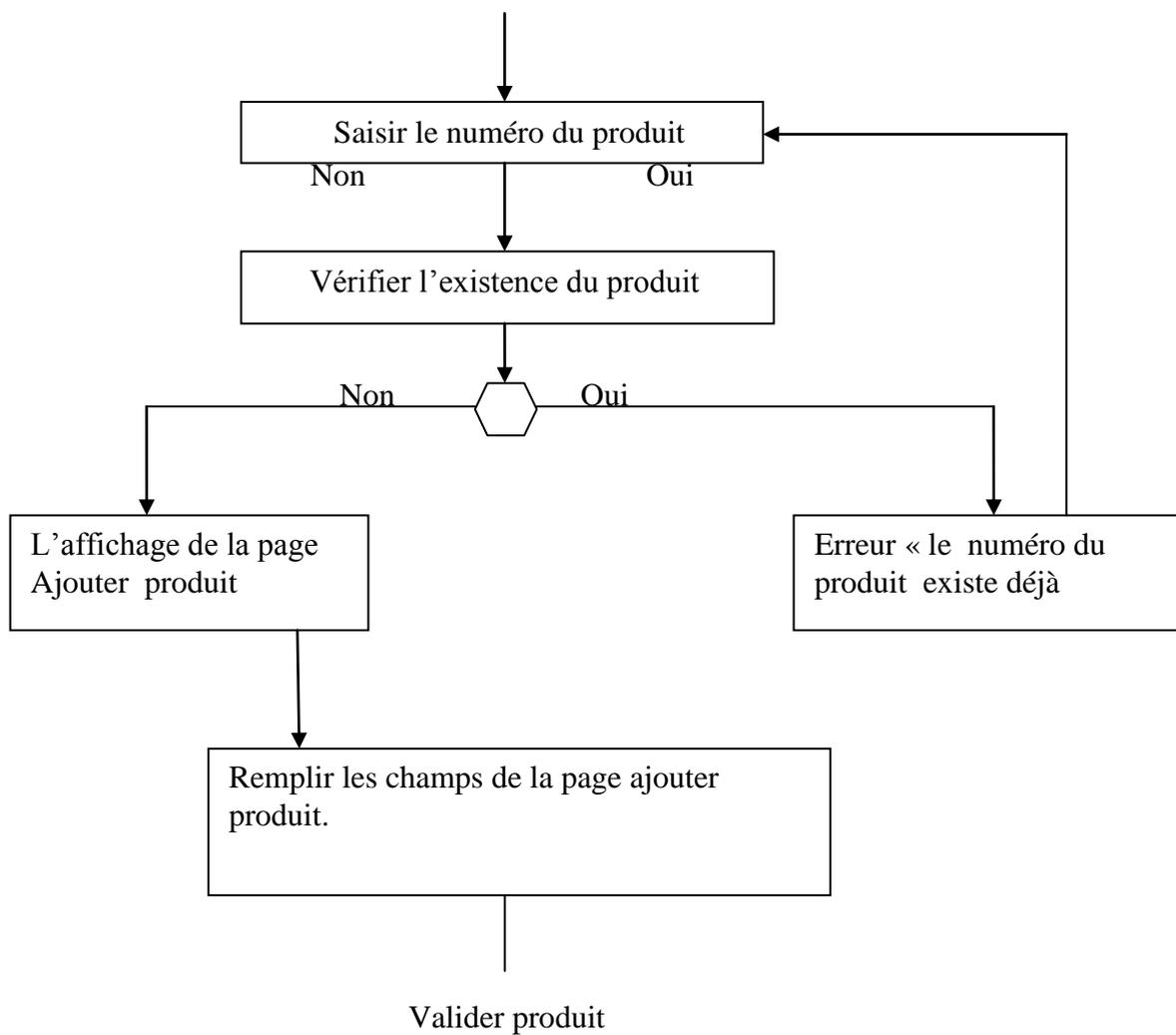


Figure 25 : Diagramme d'activité du cas d'utilisation «Ajouter un produit »

Diagramme d'activité du cas d'utilisation «Ajouter Facture » :

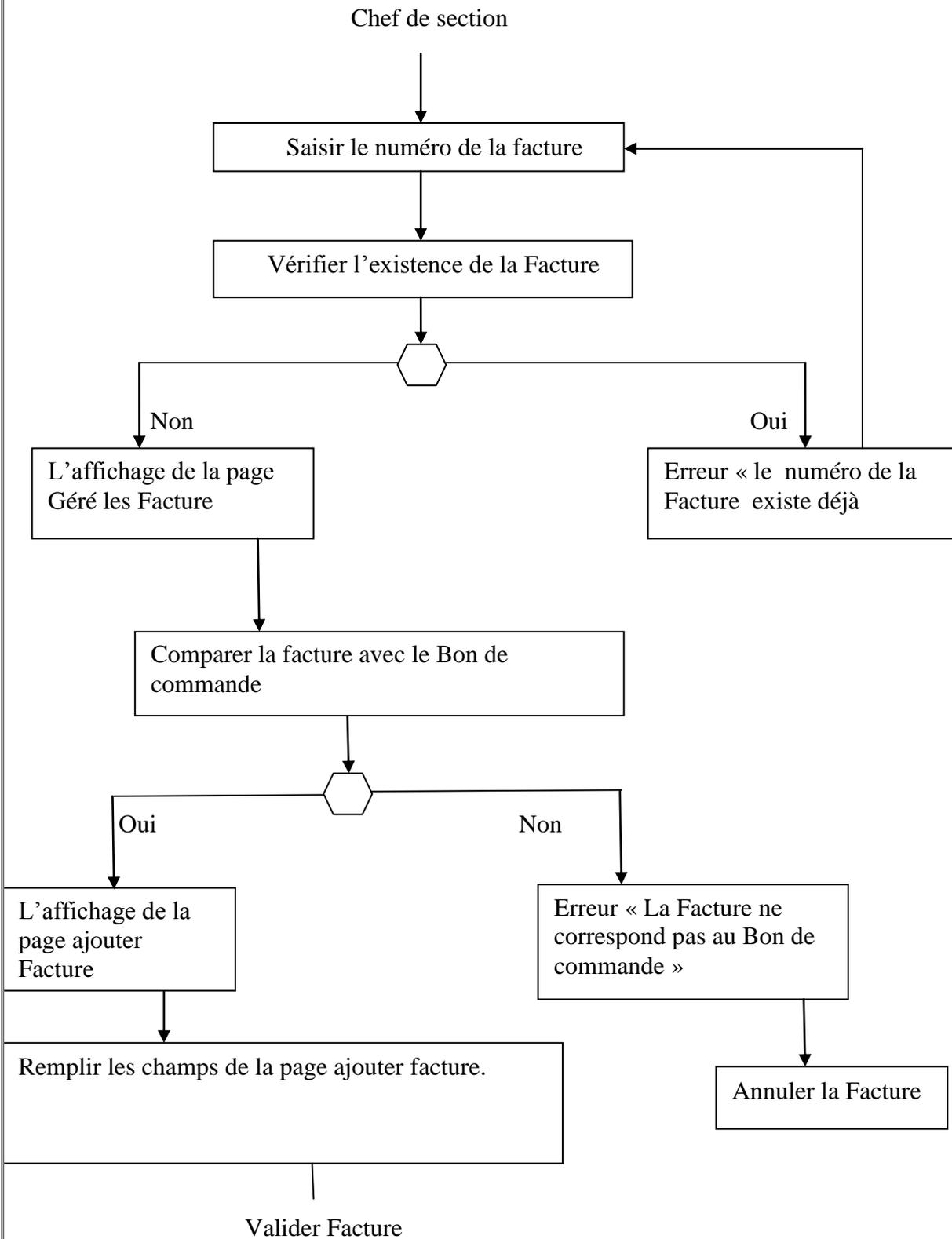


Figure 26 : Diagramme d'activité du cas d'utilisation «Ajouter Facture »

Diagramme de Classe Générale :

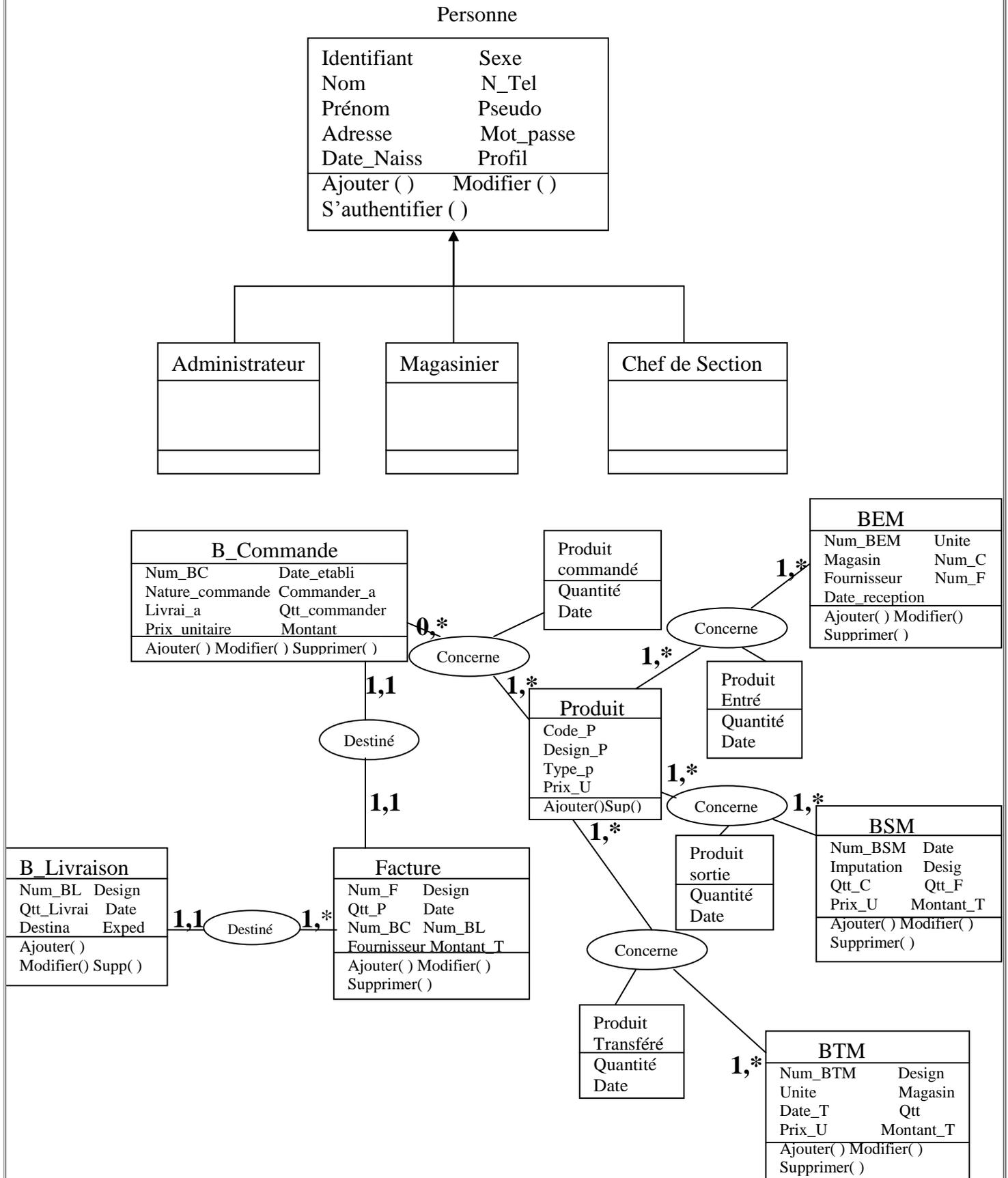


Figure 27 : Diagramme d'activité du cas d'utilisation

III.5. Conception de la base de données :

Dans ce niveau, le travail consiste à traduire le diagramme de classe de façon à optimiser l'implémentation de la base de données pour finir par le modèle physique qui représente l'implémentation des tables de la base de données. Après la traduction des diagrammes de cas d'utilisation en diagramme de séquences puis en diagramme de classes, le diagramme de classes final nous a servi de référence pour l'implémentation de la base de données, et à cette étape chaque classe sera implémentée comme une table avec ces colonnes et ces attributs.

Les tables :

L'application dans tous ces cas d'utilisation manipule des données stockées dans la base de données « gestionstock » et elle contient les tables suivantes :

La table produit :

Nom du Champ	Type de donnée	Description	Clef
Code	N(10)	Code de produit	Primaire
Designation	AB(30)	Désignation de produit	
Type_Produit	AB(10)	Type de produit	
Prix_unitaire	N(10)	Prix unitaire produit	

La Table Compte Utilisateur :

Nom du Champ	Type de donnée	Description	Clef
Numero	N(11)	Identificateur d'utilisateur	Primaire
Nom	AB(30)	Nom d'utilisateur	
Prenom	AB(30)	Prenom d'utilisateur	
Date_de_N	Date	Date de naissance d'utilisateur	
Adresse	AN(120)	Adresse d'utilisateur	
Sexe	AB(15)	Sexe d'utilisateur	
N_Tel	N(15)	N° de Tel d'utilisateur	
Mot_de_Passe	AN(20)	Mot de passe utilisateur	
Pseudo	AN(20)	Pseudo utilisateur	

La Table Compte Administrateur :

Nom du Champ	Type de donnée	Description	Clef
Nom	AB(30)	Nom de l'administrateur	Primaire
Prenom	AB(30)	Prenom de l'administrateur	
Date_de_N	Date	Date de naissance de l'administrateur	
Adresse	AN(120)	Adresse de l'administrateur	
Sexe	AB(15)	Sexe de l'administrateur	
N_Tel	N(15)	N° de Tel dde l'admonistrateur	
Mot_de_Passe	AN(20)	Mot de passe administrateur	
Pseudo	AN(20)	Pseudo administrateur	

La Table Facture :

Nom du Champ	Type de donnée	Description	Clef
Num_facture	N(10)	Numéro facture	primaire
Designation_produit	A(30)	Désignation de produit	
Quantité_Produit	N(15)	Quantité Produit	
Date	Date	Date facture	
Num_B_C	N(10)	Numéro bon de commande	
Num_B_L	N (10)	Numéro bon de livraison	
Reglement	A(10)	Type de Règlement	
Fournisseur	AN(60)	Fournisseur	
Adersse_fournisseur	AN(60)	Adresse de fournisseur	
Mountant_Total	N(20)	Montant total	
TVA	N(20)	TVA	
Mountant_Payer	N(20)	Montant a payer	
Montant_En_Lettre	A(60)	Montant en lettre	

La Table Bon de commande :

Nom du Champ	Type de donnée	Description	Clef
Num_BC	N(15)	Numéro de Bon de Commande	Primaire
Date_etablie	Date	Date etablissement	
Nature_Commande	A(30)	Nature de la comande	
Commander_A	AN(70)	Commander à	
Livrais_A	AN(70)	Livrer à	
Référence_Devis	N(10)	Référence Devis	
Date_Devis	Date	Date Devis	
Délai_Livraison	AN(15)	Délai de Livraison	
Quantite_comande	N(15)	Quantité comandée	
Prix_Unitaire	N(20)	Prix unitaire	
Montant	N(30)	montant	
Solliciter_Par	A(30)	Solliciter par	
Apraue_Par	A(30)	Apprue par	
Autoriser_Par	A(30)	Autoriser par	
Solliciter_Titre	A(30)	Solliciter Titre	
Aprrouve_Titre	A(30)	Approuvé Titre	
Autoriser_Titre	A(30)	Autoriser Titre	
Description_Imputation	A(30)	Description de l'imputation	

La Table Bon de livraison :

Nom du Champ	Type de donnée	Description	Clef
Num_BL	N(10)	Numéros de bon de livraison	Primaire
Designation	A(30)	Bon de livraison	
Quantité_Livrai	N(15)	Quantité Livrai	
Date	Date	Date de livraison	
Adresse	AN(60)	Adresse de livraison	
Expediteur	AB(50)	expéditeur	
Adresse_expediteur	AN(60)	Adresse de l'expéditeur	

Table Bon d'entrée Magasin :

Nom du Champ	Type de donnée	Description	Clef
Num_BEM	N(10)	Numéro du bon d'entrée magasin	primaire
Unite	AB(15)	Unité	
Magasin	AB(15)	Magasin	
Fournisseur	AB(20)	Fournisseur	
Num_Facture	N(15)	Numéro facture	
Num_Commande	N(15)	Numéro commande	
Date_Reception	Date	Date de réception	
Monnaie_Origine	A(15)	Monnaie d'origine	
Montant_Dinare	N(30)	Montant en dinar	
Montant_total_Dinar	N(30)	Montant total en dinar	
DD_Montant_Facture	N(30)	DD+Montant Facture	
Designation	AB(30)	Bon d'entre au magasin	
Quantite_Entrée	N(15)	Quantité entrée	
Prix_Unitaire_Dinar	N(20)	Prix unitaire en Dinars y compris droit douane	
Montant_En_Dinar	N(20)	Montant en Dinars y compris Droit Douane	
Montant_Total_Dinars	N(30)	Montant total en Dinars y compris droit douane	
Montant_en_Lettre	AB(40)	Montant en lettre	
Nom_Demarcheur	AB(20)	Nom de marcheur	
Date_Signature_Marcheur	Date	Date de signature par marcheur	
Nom_Magasinier	AB(20)	Nom de magasinier	
Date_Signature_Magasinier	Date	Date de signature par magasinier	
Nom_Responsable_Fiche_Stock	AB(30)	Nom de responsable fiche de stock	
Date_Signature_Responsable_fiche_Stk	Date	Date de signature par responsable fiche de stock	

Table Bon de Sortie Magasin :

Nom du Champ	Type de donnée	Description	Clef
--------------	----------------	-------------	------

Num_BSM	N(10)	Numéro bon de sortie magasin	Primaire
Designation	A(30)	Bon de sortie magasin	
Date	Date	Date de sortie magasin	
Imputaion	A(10)	imputation	
Item	AB(5)		
Code_refernece	N(10)	Code référence	
Q_D	N(10)	Quantité demandées	
Q_F	N(10)	Quantité fournées	
Prix_Unitaire	N(10)	Prix unitaire	
Montant	N(10)	montant	
Montant_Total	N(20)	Montant total	
Observation	AB(20)	observation	
Nom_Demandeur	AB(10)	Nom de demandeur	
Date_demande	Date	Date de demande	
Nom_Receptionnaire	AB(10)	Nom de réceptionnaire	
Date_Reception	Date	Date de réception	
Nom_Magasin	AB(10)	Nom de magasin	
Date_Sortie_Mag	Date	date	
Nom_Gestion_Stock	AB(10)	Nom de gérant de stock	
Date_Gestion_Stock	Date	Date de réception	

Table Bon de Transfert Magasin :

Nom du Champ	Type de donnée	Description	Clef
Num_BTM	N(10)	Numéro de bon de transfert magasin	Primaire
Designation_Prod	AB(30)	Bon de transfert magasin	
Unite	AB(20)	Unité	
Magasin	AB(10)	Magasin	
Date_Transfert	Date	Date de transfert	
Adresse_Transfert	AN(80)	Adresse de transfert	
Code_Prod	N(15)	Code Produit	
UM	N(5)	UM	
Quantite	N(10)	Quantité transférée	
Prix_Unitaire	N(20)	Prix unitaire	
Montant	N(15)	Montant	
Montant_Total	N(15)	Montant Total	
Montant_En_Lettre	AB(50)	Montant en lettre	
Nom_Chef_Mag_Cedant	AB(10)	Nom de chef de magasin cédant	
Date_Segn_Cedant	Date	Date de signature	
Nom_Chef_Mag_Cess	AB(10)	Nom de chef de magasin cessionnaire	
Date_Segn_Cess	Date	Date de signature	

Table de comparaison des Facture préforma :

Nom du Champ	Type de donnée	Description	Clef
Num_Comparaison	N(15)	Numéro	Primaire
Date_etablie	Date	Date d'établissement	
Désignation	AN(120)	Désignation produit	
Qte	N(10)	Quantité produit	
Nom_fournisseur1	A(25)	Nom fournisseur1	
Nom_fournisseur2	A(25)	Nom fournisseur2	
Nom_fournisseur3	A(25)	Nom fournisseur3	
Prix_fournisseur1	N(15)	Prix fournisseur1	
Prix_fournisseur2	N(15)	Prix fournisseur2	
Prix_fournisseur3	N(15)	Prix fournisseur3	
Montant_payé_naftal	N(15)	Montant a payé par naftal	
Nom_fournisseur_MD	N(15)	Nom fournisseur moins distant	
Nom_chef_service	A(25)	Nom chef service MGX	
Nom_chef_dept	A(25)	Nom chef de dept AMGX	
Nom_directrice_distr	A(25)	Nom directrice district	

Table demande de fourniture :

Nom du Champ	Type de donnée	Description	Clef
Num_DF	N(15)	Numéro de la demande fourniture	Primaire
Date_etabli	Date	Date d'établissement	
Num_Envoie	N(10)	Numéro d'envoi	
Imputation	A(10)	Imputation	
Code_produit	N(10)	Code Produit	
Designation_Prod	AB(30)	Demande fourniture	
Num_Reference	AN(10)	Numéro référence	
Quantite_Demandee	N(10)	Quantité demandée	
Quantite_Accordee	N(10)	Quantité accordée	
Observation	AB(15)	Observation	
Date_Reception	Date	Date de réception marchandise	
Nom_Recepteur	AB(10)	Nom du réceptionnaire	
Fonction_Recepteur	AB(20)	Fonction de récepteur	
Nom_Demandeur	AB(10)	Nom demandeur	
Qualite_Demander	AB(20)	Qualité Demander	
Nom_Magasinier	AB(10)	Nom magasinier	
Qualite_Livraier	AB(20)	Qualité Livrai	

La Table bilan matières et fourniture :

Nom du Champ	Type de donnée	Description	Clef
--------------	----------------	-------------	------

Num_Bilan	N(10)	Numéro Bilan	Primaire
Date_Bilan	Date	Date du Bilan	
Num_Compt	N(15)	Numéro Compte	
Stock_initiale	N(20)	Stock initiale	
Entrée	N(10)	Entrée	
Inter_unité	N(10)	Inter unité	
Total_entrée	N(30)	Total entrée	
Consommat	N(20)	Consommat	
Cesion	N(10)	Cesion	
Stock_Final	N(30)	Stock Final	

IV. 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 :

Cette étape d'implémentation a comme principales activités la traduction des interfaces de la conception en code exécutable.

Dans ce chapitre nous allons, tout d'abord décrire l'environnement de développement et les langages de programmation de notre application. Puis nous nous focaliserons sur la construction des composants spécifiques au web et plus particulièrement les pages web. Et on conclura par la présentation des interfaces essentielles de notre application.

II. Les outils de développement :

II.1.Spring Framework :[15] [16]

Le framework Spring fournit un modèle de programmation et la configuration globale pour les applications d'entreprise modernes basées sur Java - sur n'importe quel type de plate-forme de déploiement. Un élément clé du printemps est un soutien infrastructurel au niveau de l'application: Printemps met l'accent sur la «plomberie» des applications d'entreprise afin que les équipes puissent se concentrer sur la logique métier de niveau de l'application, sans liens inutiles aux environnements de déploiement spécifiques.

Le framework Spring est un framework vraiment très puissant qui peut ajouter beaucoup de fonctionnalités à vos applications Java.

Spring 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 Spring simple est à la portée de tous, avoir une architecture Spring 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 Spring, 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

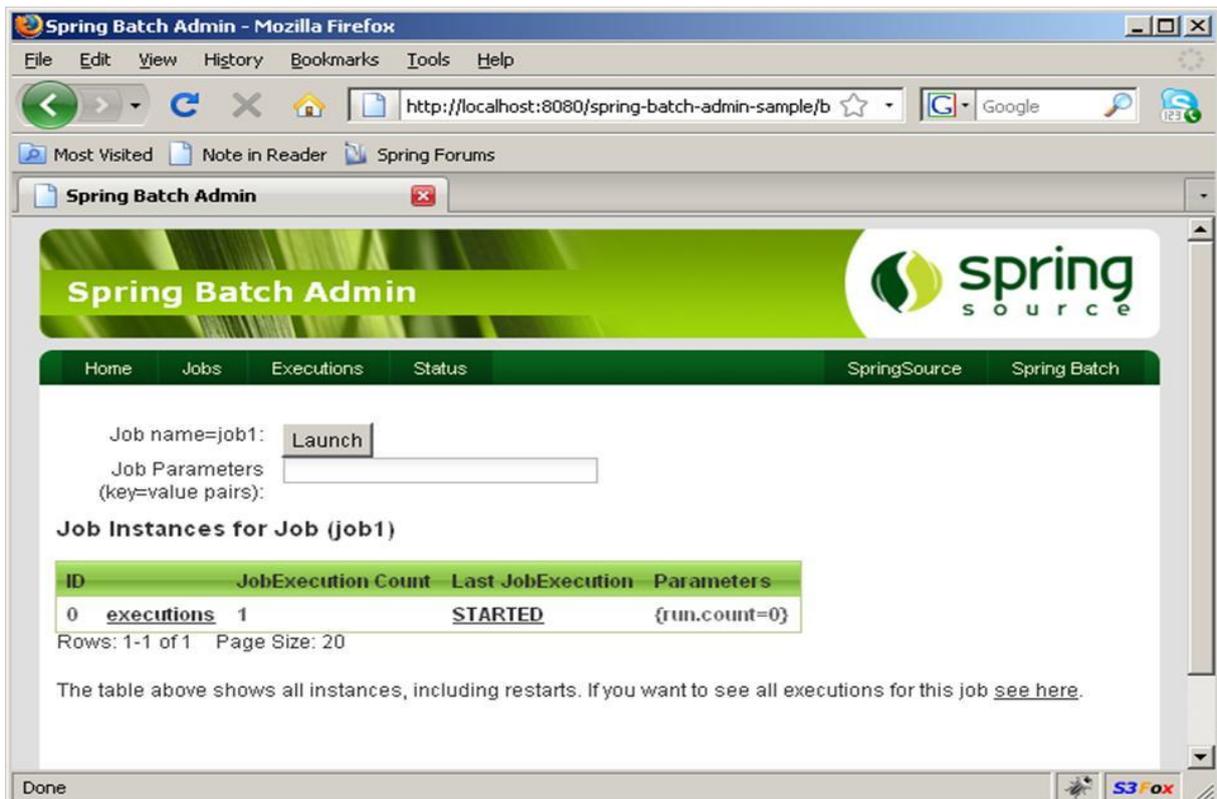


Figure 26 :l'interface de Spring Framework

II.2.Serveur de base de données PHP My admin : [11]

Est une application web qui permet de gérer un serveur de BD MySQL. Dans un environnement multiutilisateur, cette interface écrite en PHP permet également de donner à un utilisateur l'accès à ses propres bases de données. C'est un outil qui fournit une interface facile à utiliser pour la gestion d'une BD. L'utilisation de PHP MyAdmin implique vous êtes l'administrateur de la base de données c'est-à-dire le seul qui puisse changer les données des différentes tables de votre base, mais vous ne pouvez pas effectuer des modifications sur les tables d'un tiers utilisateur.

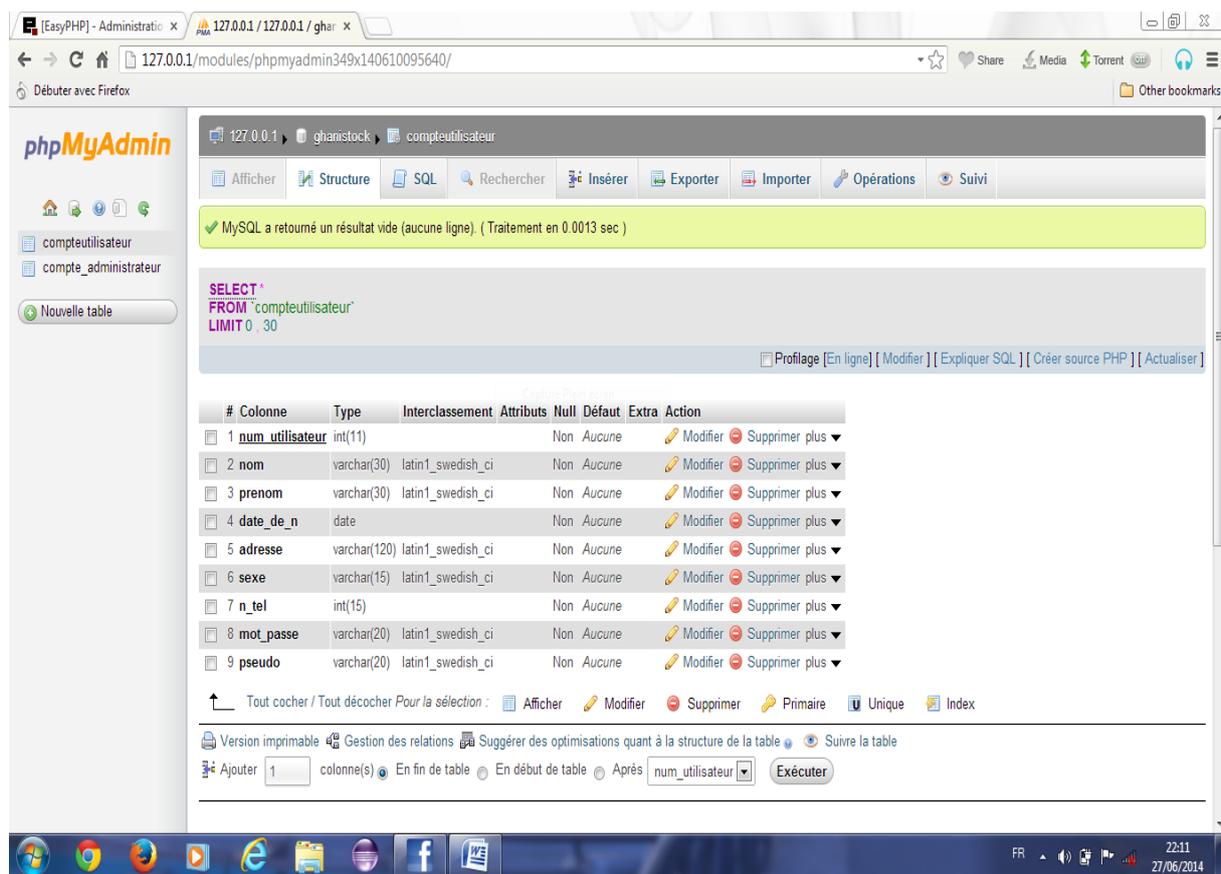


Figure 27 : L'interface de MySQL admin de EasyPHP.

II.3. Serveur Apache Tomcat 8.0.8: [27]

C'est le serveur le plus répandu sur internet, permettant la configuration de l'environnement d'exécution de pages web, il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type l'Unix, mais il a désormais été porté sur de nombreux systèmes d'exploitation dont Microsoft Windows grâce à sa conception modulaire (morceau de code) qui correspond à différents aspects ou fonctions du serveur.

III. Les langages de programmation :

III.1. J2EE: [26]

Sun (le concepteur de Java) a mis en place un ensemble de technologies pour réaliser des applications Web.

Ces technologies sont regroupées sous le nom J2EE (*Java 2 Enterprise Edition*).

Depuis la version 5, le chiffre 2 a disparu (ambiguïté avec la version)

En simplifiant, il est possible de représenter Java EE comme un ensemble de spécifications d'API, une architecture, une méthode de packaging et de déploiement d'applications et la gestion d'applications déployées sur un serveur compatible Java. Apparue à la fin des années 90. Destinée aux gros (très gros) systèmes d'entreprises.

A pour objectif de crédibiliser Java pour les applications d'entreprise.

Java EE ne remplace en aucun cas J2SE mais elle constitue une extension de J2SE.

Elle permet un développement d'applications qui vont s'exécuter sur un serveur d'applications. La plateforme J2SE offre des outils de développement d'applications client/serveur, applications graphiques fenêtrées et Applets.

La plateforme J2SE est composée des éléments suivants :

La machine virtuelle Java (JVM) :

Permet d'exécuter des applications Java (tinterpréteure)

Permet une portabilité entre les architectures (byte code)

La bibliothèque de classes Java:

un ensemble de composants logiciels prêt à l'emploi

Les outils de développement :

javac, java, javadoc, Jconsole...

Pourquoi choisir Java EE :

- la portabilité
- l'indépendance,
- Sécurité
- la multitude de bibliothèques proposées
- fonctionnalités sont développées (robustesse, sûreté,)
- l'accès aux bases de données
- l'envoi de mails
- Les transactions
- la gestion de fichiers
- la gestion d'images
- le téléchargement
- la supervision du Système.

La plateforme Java EE est basée sur des spécifications donc les projets sont portables sur n'importe quel serveur d'applications conforme (Tomcat, JBoss, WebSphere...)

III.2.Netbeans :[26]

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 greffons.

Historique :

En 1997, NetBeans naît de Xelfi, un projet d'étudiants dirigé par la Faculté de mathématiques et de physique de l'Université Charles de Prague. Plus tard, une société se forme autour du projet et édite des versions commerciales de l'EDI NetBeans, jusqu'à ce qu'il soit acheté par Sun en 1999. Sun place le projet sous double licence CDDL et GPL v2 en juin de l'année suivante.

Multilingue :

NetBeans est disponible en 23 langues (version 7.0 sortie en avril 2011), dont la plupart de ces traductions a été réalisée par la communauté des utilisateurs.

Environnement :

Netbeans est un IDE qui supporte une large variété de langages de programmation et d'outils de collaboration.

Environnement de base :

L'environnement de base comprend les fonctions générales suivantes:

- configuration et gestion de l'interface graphique des utilisateurs,
- support de différents langages de programmation,

- traitement du code source (édition, navigation, formatage, inspection..),
- fonctions d'import/export depuis et vers d'autres IDE, tels qu'Eclipse ou JBuilder,
- accès et gestion de bases de données, serveurs Web, ressources partagées,
- gestion de tâches (à faire, suivi ...),
- documentation intégrée.

Principaux langages supportés :

L'éditeur intégré propose des fonctions de complètement, de contrôles syntaxiques et sémantiques, d'avertissements et de conseils, de reprises de code (« refactoring » : renom mage, changement des méthodes, gestion des classes, ...), de sauvegarde et reprise.

Il supporte principalement les langages suivants :

- Java (Java SE, Java ME, Java FX, Java EE), Javadoc ;
- Groovy et Grails ;
- PHP (dont les environnements Zend et Symfony) ;
- JavaScript ;
- C, C++, Fortran. Netbeans ne requiert pas l'utilisation d'un compilateur particulier. À noter le support des bibliothèques Qt. Les plates formes supportées sont Microsoft Windows, Linux, Mac OS, Solaris 10 et OpenSolaris ;
- Python ;
- HTML, XHTML, RHTML ;
- XML ;
- DTD ;
- CSS ;
- JSP, JSF ;
- Ruby (via un greffon depuis la version 7.0 de netBeans) .

Bases de données :

NetBeans comprend un explorateur de bases de données qui supporte toutes les bases relationnelles pour lesquelles un connecteur JDBC existe (selon les versions des gestionnaires de bases de données): JavaDB (Derby) MySQL, PostgreSQL, Oracle, Microsoft SQL, PointBase, jTDS, IBM Redistributable DB2, ...

L'explorateur comprend un éditeur de requêtes, un gestionnaire intégré de bases de données MySQL.

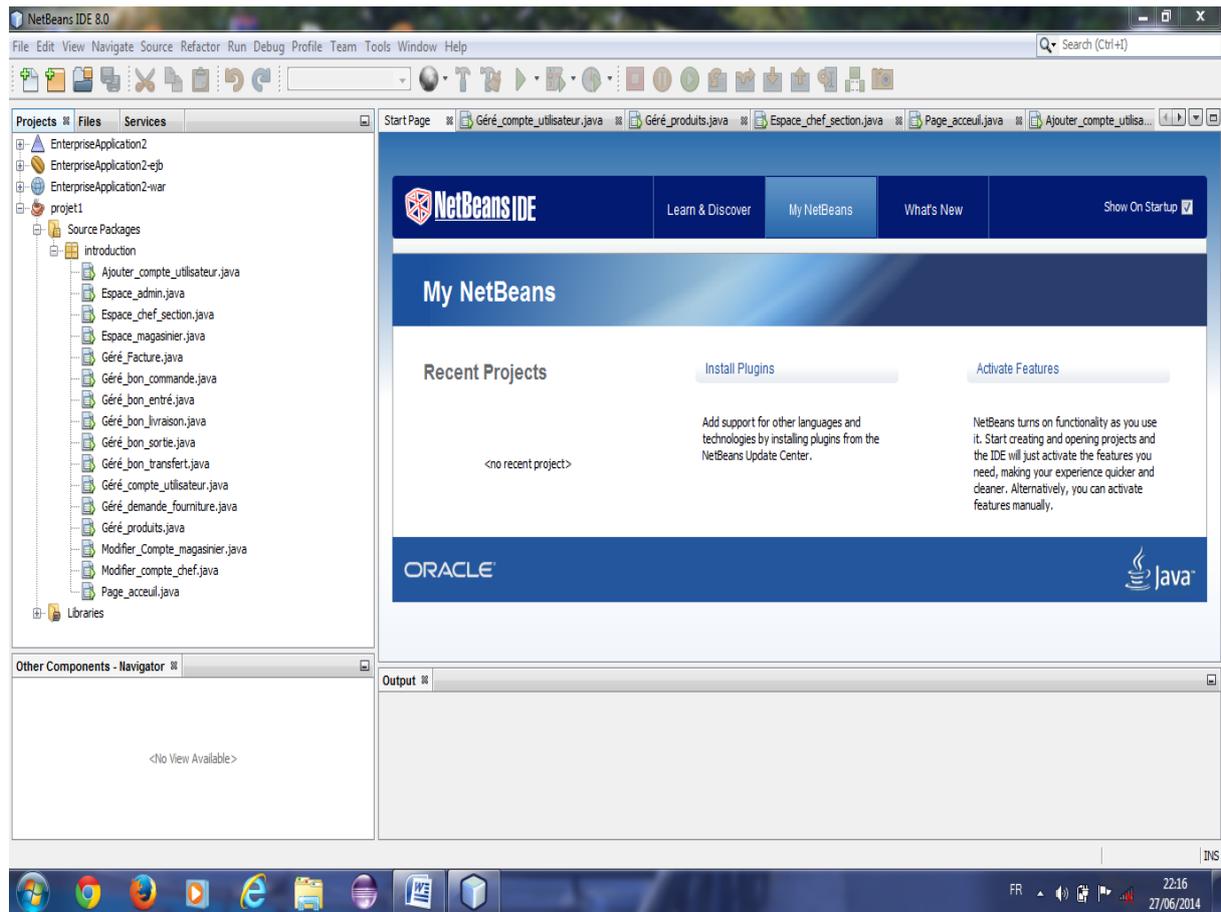


Figure 28 :L'interface de netbeans

IV. Présentation de quelques interfaces de l'application :

IV.1. La page d'accueil :

C'est la premier page visualisé par l'utilisateur, , Cette page leurs permise la possibilité de s'inscrire, s'authentifier autant que utilisateur pour accéder à son propre espace et bénéficier de tous les services proposé ou bien, accédé autant que administrateur pour géré les comptes utilisateur et les produits .

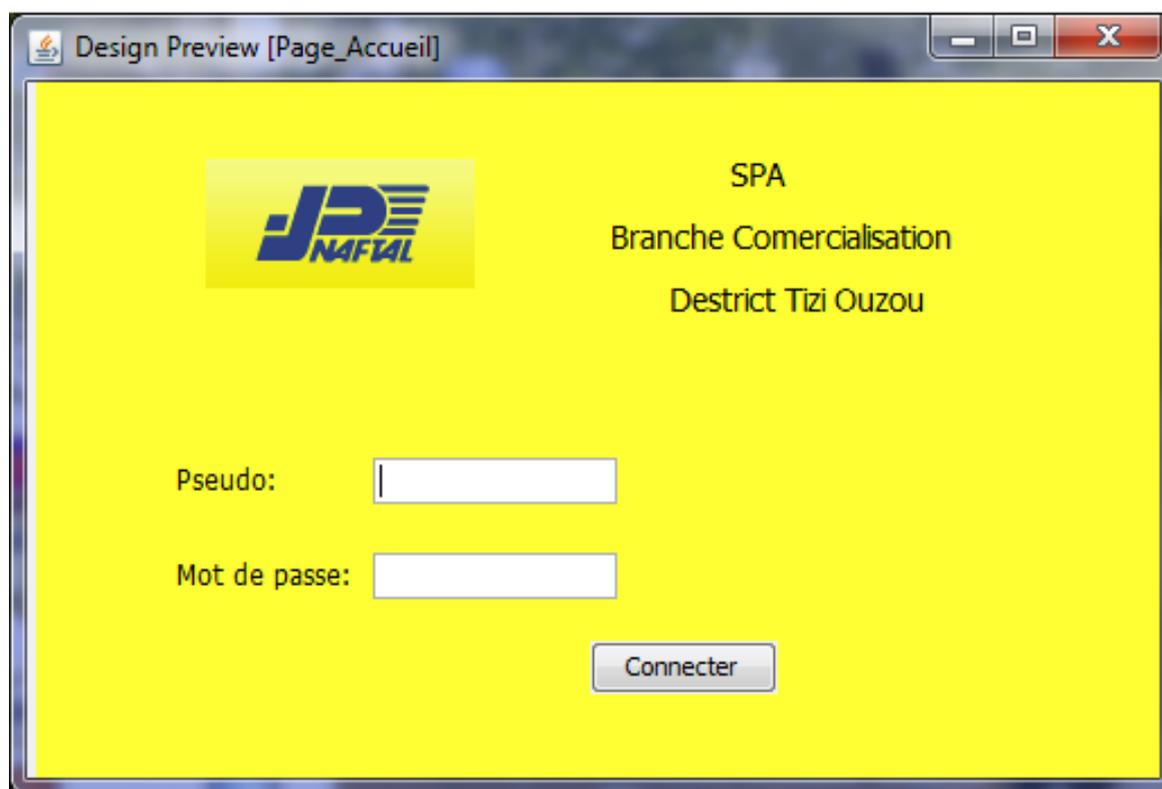


Figure 29 : Page d'accueil de notre application.

IV.2. La page espace administrateur :

L'administrateur peut gérer les compte utilisateur on cliquant sur le lien « géré les comptes utilisateur» et dans cette page il peut :

- Ajouter un compte.
- Recherche un compte.
- Modifier un compte.
- Activer/désactivé un compte.
- Valider un compte.

Et en cliquant sur le lien « géré les produits » il peut :

- Ajouter un produit.
- Recherche un produit.
- Modifier un produit.
- Activer/désactivé un produit.
- Valider un produit.

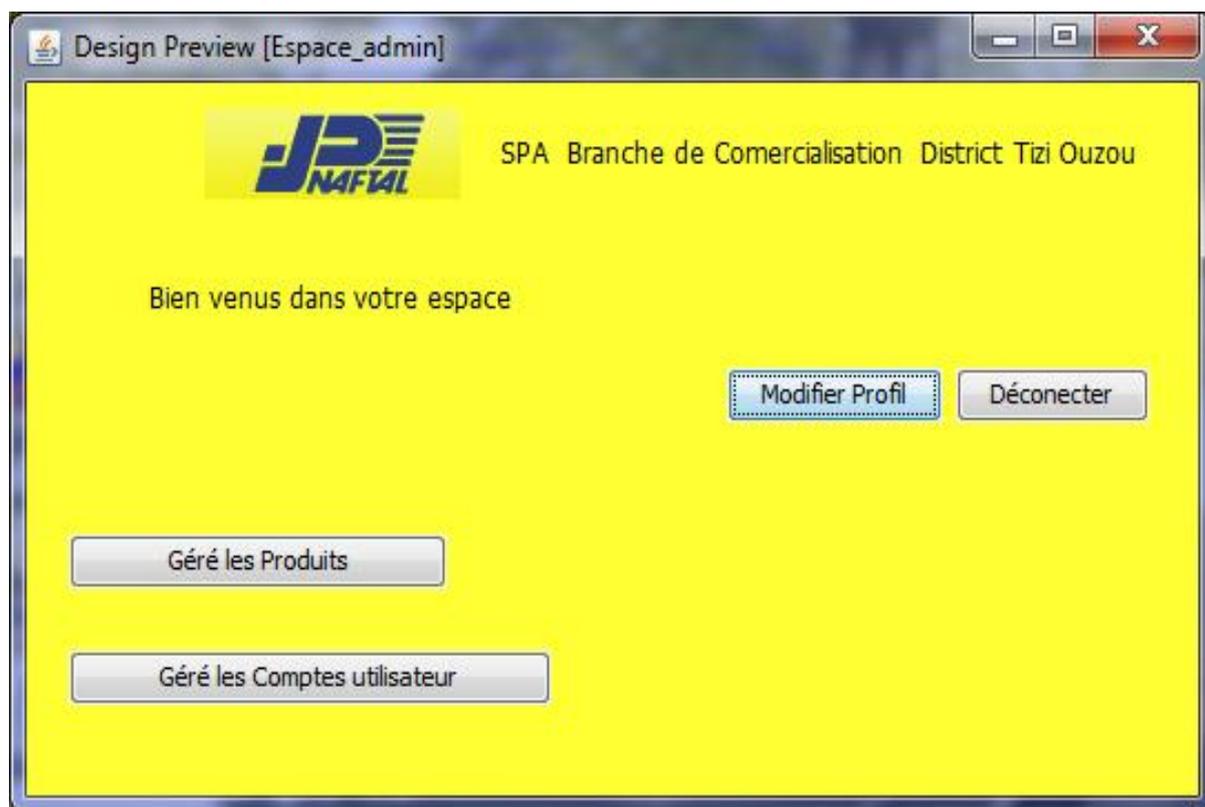


Figure 30 : Page espace administrateur.

IV.3. La page espace chef de section :

Dans cette page le chef de section peut faire les tâches suivantes :

- Modifier Profil.
- Comparé les factures préforma.
- Géré les bon de commande.
- Géré les bon de livraison.
- Géré les facture.
- Consultation du bon d'entré magasin.
- Consultation du bon de transfert magasin



Figure 31: Page espace chef de section.

IV.4. La page espace magasinier :

Dans cette page le magasinier peut faire les tâches suivantes :

- Modifier Profil.
- Géré les bon d'entrée magasin.
- Géré les bon de sortie magasin.
- Géré les bon de transfert magasin.
- Géré les demande de fourniture.



Figure 32 : Page espace magasinier

IV.5. La page ajouter un compte utilisateur:

Pour qu'un utilisateur possède un compte propre sur cette application de l'entreprise, il doit être inscrit par l'administrateur.

The image shows a web browser window with a yellow background. At the top left is the logo for 'JP NAFTAL'. To its right, the text reads 'SPA Branche de Comercialisation Destrict Tizi Ouzou'. Below this, the title 'Fomulaire d'inscription' is centered. The form contains the following fields and controls:

- Nom:
- Prénom:
- Date de Naissance:
- Adresse:
- Sexe:
- Num Telephone:
- Pseudo:
- Mot de passe:

At the bottom of the form are two buttons: 'Valider' and 'Annuler'.

Figure 33: Page espace magasinier

IV.6. La page Modifier Profil :

Dans cette page le chef de section il peut faire les tâches suivant :

- Saisir les nouvelles données.
- Valider la saisie.

The image shows a software window titled 'Formulaire de Modification' for 'SPA Branche Comercialisation District Tizi Ouzou'. The window has a yellow background and a blue header bar with the 'JP NAFTAL' logo. It contains four text input fields for 'Adresse:', 'Num Telephone:', 'Ancien Mot de passe:', and 'Nouveau Mot de passe:'. At the bottom, there are two buttons: 'Valider' and 'Annuler'.

Figure 34 : Page Modifier Profil.

V. Conclusion :

Dans ce chapitre on a présenté en premier lieu l'environnement d'exécution et de programmation de l'application, ensuite on est passé à la présentation de la base de données, et pour terminer on a présenté quelques interfaces de l'application.

Conclusion générale

Aujourd'hui, toute entreprise doit se mettre au diapason de l'évolution de la technologie et cela quelque soit sa vocation et son caractère.

Pour cela, toute organisation qui aspire à l'amélioration de son rendement et de son service doit se doter d'un système d'information automatisé.

Le stage que nous avons réalisé dans le district naftal de Ouad Aissi nous a permis:

- D'acquérir des connaissances sur l'entreprise Naftal, des districts de Ouad Aissi et ses différents services et centres.
- De voir de près la circulation de l'information, les moyens utilisés pour véhiculer cette dernière.
- De découvrir, voir de près les tâches complexes auxquelles le personnel des différents services concernés fait face.
- Approfondir nos connaissances sur les systèmes d'information, La gestion et l'administration des bases de données.
- Acquérir de nouvelles connaissances sur le MySQL (EasyPHP 5.3.9) (pour l'implémentation de la base de données)
- Acquérir des connaissances sur le langage de modélisation UML.
- Acquérir des connaissances plus avancées sur le modèle client/serveur et son implémentation.

Ce stage nous a permis aussi de mettre en application nos connaissances, acquises durant tout le cursus .En plus de l'expérience dans le développement d'application, ce projet à travers ces étapes de mise en œuvre de notamment l'analyse, la conception et la réalisation, nous a conduit à l'utilisation des outils de conception « UML » et de réalisation « produit de MySQL » avec tous ce qu'il offre comme fonction et outils.

Cependant, malgré les diverses fonctionnalités (les différents services) qu'offre notre application quelques améliorations et compléments peuvent être apportés, en guise des perspectives :

Réalisation d'une application qui gère toute les tâches du District de Ouad Aissi.

Nous souhaitons que ce modeste travail constitue un support d'aide aux étudiants à venir et sera de grand intérêt pour le district et que les futurs utilisateurs y trouveront satisfaction

Annexe A

I. Introduction : [12]

L'UML (Unified Modeling Language) est un langage conçu pour représenter, spécifier, construire et documenter les artefacts d'un système à dominante logicielle. Il permet d'écrire avec un langage standardisé les plans d'élaboration et de construction de logiciels. Il prend en compte aussi bien des éléments conceptuels tels que les processus d'entreprises et les fonctions du système, que des éléments concrets tels que les classes écrites dans un langage de programmation, les schémas de bases de données et les composants logiciels réutilisables.

Face à la diversité des méthodes d'analyse et de conception orientée objet, en particulier aux différentes notations des mêmes concepts, UML vient d'unifier les approches d'analyse et de conception objet. Elle représente une synthèse des trois méthodes : OOD (Object Oriented Design) de G.Booch, OOSE (Object Oriented Software Engineering) de I.Jacobson et OMT (Object Modeling Technique) de J.Rumbaugh. Pour résumer, la méthode OOD était particulièrement expressive lors des phases de conception et de construction de projets, OOSE représentait un excellent outil pour les cas d'utilisation en matière de définition des exigences, d'analyse et de conception générale et OMT était plus particulièrement utile à l'analyse et aux systèmes d'information contenant une grande quantité de données.

UML est destiné avant tout aux systèmes à forte composante logicielle. Il a été utilisé avec succès dans des domaines tels que [Booch & all, 00]:

- Les systèmes informatiques d'entreprises,
- Les services bancaires et financiers,
- Les télécommunications,
- Les transports,
- La défense/l'aérospatiale,
- Le commerce de détail,
- L'électronique médicale,
- Les sciences,
- Les services distribués basés sur le Web.

UML est indépendant des processus utilisés, ce qui signifie qu'il n'est pas lié à un cycle de développement particulier. Cependant, pour exploiter UML au maximum, il faut envisager un processus qui soit fondé sur les cas d'utilisation, centré sur l'architecture et itératif/ incrémental.

II. Cycle de développement logiciel [Booch & all, 00] : [13]

Le Rational Unified Process (RUP) est un processus itératif. Il est l'une des approches de cycle de vie particulièrement bien adapté à UML. Il saisit certaines des meilleures pratiques actuelles de développement sous une forme qui s'adapte à un grand nombre de projets et d'organisations.

Phases et itération d'RUP

Rational Unified Process comporte quatre phases :

⇒ **Inception** : *Définir le cadre du projet*

Au cours de cette phase d'inception, on définit le cadre du système et on délimite la partie du projet. Le cadre comprend les critères de réussite, la mise en évidence des risques, les estimations des ressources nécessaires et un plan de phase qui contient un planning des principaux jalons. Généralement un prototype exécutable qui sert de vie du projet et on décide si l'on est en mesure de passer au développement à grande échelle.

⇒ **Elaboration** : *Etablir un plan du projet et une architecture solide*

Les objectifs de cette phase consistent à analyser le domaine du problème, établir une base architecturale solide, développer le plan du projet et éliminer les éléments qui présentent le plus de

risques pour le projet. On prend des décisions relatives à l'architecture, ce qui implique d'avoir une compréhension du système dans son intégrité. Afin de vérifier l'architecture, on implémente un système qui démontre les choix architecturaux et exécute des cas d'utilisation importants.

A l'issue de cette phase, on examine en détail les objectifs et la portée du système, le choix de l'architecture et la résolution des principaux risques, puis on décide de passer ou non à la phase de construction.

⇒ **Construction : Développer le système**

Au cours de cette phase, on développe un produit complet et prêt à transiter vers les utilisateurs, de manière itérative et incrémental. Ce qui implique la description des exigences restantes ainsi que les critères d'acceptation, d'enrichir la conception et de compléter l'implémentation avant de tester le logiciel.

A l'issue de cette phase, on décide si le logiciel, le site et les utilisateurs sont opérationnels.

⇒ **Transition : Fournir le système aux utilisateurs finals**

Au cours de cette phase, on déploie le logiciel pour les utilisateurs. Lorsque les utilisateurs finals disposent du système, souvent des problèmes surgissent. Il est alors nécessaire de développer à nouveau afin d'ajuster le système, de corriger certains problèmes non détectés ou d'achever certaines fonctionnalités qui avaient été remises à plus tard. Cette phase commence généralement par une version initiale du système que l'on remplace par le système définitif.

A l'issue de cette phase, on décide si les objectifs du cycle de vie du projet sont attendus et on détermine si l'on doit commencer un autre cycle de développement.

⇒ **Itérations**

Chaque phase du Rational Unified Process peut être divisée en itérations. Une itération est une boucle de développement complète qui aboutit à la sortie d'une version d'un produit exécutable qui constitue un sous-ensemble du produit final en cours de développement. Cette version est ensuite enrichie d'itération en itération avant d'aboutir au système final. Au cours de la phase d'inception, on se concentre surtout sur les exigences. Lors de la phase d'élaboration, on se concentre plutôt sur l'analyse et la conception. Au moment de la construction, l'implémentation est l'activité principale alors que la transition est centrée sur le déploiement.

III. Notations et Diagrammes UML : [25]

Le langage UML repose sur trois éléments essentiels : *Les briques de base d'UML*, les règles qui déterminent la manière de les assembler et quelques *mécanismes généraux* qui s'appliquent à UML dans son ensemble.

III.1. Notations UML :

III.1.1. Les acteurs :

Les acteurs représentent dans UML les frontières du système. En effet, les acteurs sont des utilisateurs du système qui ont une bonne connaissance des fonctionnalités du système par ce que se sont les utilisateurs eux-mêmes qui les définissent.

L'acteur est représenté dans UML par un petit bonhomme facile à dessiner (figure A-1).

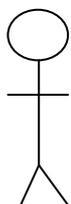


Figure 35. La représentation graphique d'un acteur dans UML

III.1.2. Cas d'utilisation :

Un cas d'utilisation décrit un ensemble de séquences d'actions, y compris des variantes, qu'un système exécute pour produire un résultat tangible pour un acteur. Un cas d'utilisation est représenté par une ellipse.

Caractéristiques d'un cas d'utilisation

Les acteurs :

Un acteur représente un ensemble de rôles joués par les utilisateurs des cas d'utilisation en interaction avec ces cas d'utilisation. En règle générale, un acteur représente un rôle qu'un homme, une machine ou même un autre système joue avec le système.

Flots d'événements :

On peut préciser le comportement d'un cas d'utilisation en décrivant des flots d'événements à l'aide d'un texte suffisamment clair pour être compris facilement, en spécifiant comment et quand le cas d'utilisation commence et se termine, le moment où il interagit avec les acteurs et les objets qui sont échangés.

III.1.3. Note :

Une note est un symbole graphique utilisé pour représenter des contraintes et des commentaires rattachés à un élément ou à un ensemble d'éléments. Elle est représentée graphiquement par un rectangle écorné qui contient un commentaire textuel ou graphique.

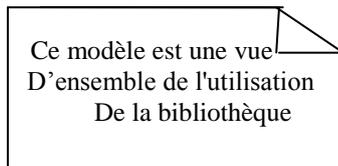


Figure 36. Exemple d'une note

III.1.4. Une contrainte :

C'est une relation sémantique entre les éléments d'une modélisation.

Elle s'écrit entre accolades dans une note. Elle est représentée graphiquement comme suit:

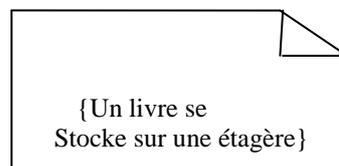


Figure 37. Exemple d'une contrainte

Paquetage :

Un paquetage est un mécanisme d'ordre général qui permet d'organiser des éléments en groupes. On le représente par un dossier étiqueté.

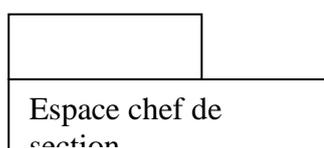


Figure 38: Représentation d'un paquetage en UML

Caractéristiques d'un paquetage

Contenu :

Un paquetage peut posséder d'autres éléments, y compris des classes, des interfaces, des composants, des nœuds, des collaborations, des cas d'utilisation, des diagrammes et même d'autres paquetages.

Visibilité :

Pour contrôler la visibilité des éléments contenus dans un paquetage, on ajoute un symbole devant son nom. Ainsi, les éléments publics sont précédés du symbole +, les éléments protégés sont précédés du symbole # et les éléments privés sont précédés du symbole -. Les éléments protégés sont visibles seulement par les paquetages qui héritent d'un autre paquetage et les éléments privés ne sont pas visibles du tout en dehors de leur paquetage.

Généralisation :

Elle ressemble beaucoup à la généralisation entre classes. Les paquetages spécialisés héritent des éléments publics et protégés du paquetage plus général.

III.1.5. Une classe :

Elle constitue l'élément de base le plus important d'un système orienté objet. Elle correspond à la description d'un ensemble d'objets qui partagent les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique. Elle est symbolisée par un rectangle.

Caractéristiques d'une classe

◆ *Attributs :*

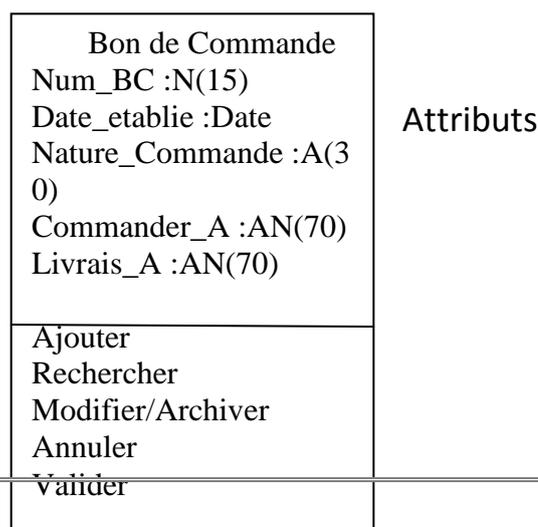
Un attribut est une propriété nommée d'une classe qui décrit un ensemble de valeurs que les instances de cette propriété peuvent prendre. Un attribut représente une propriété de la classe modélisée et cette propriété est commune à l'ensemble des objets de cette classe.

◆ *Opérations :*

Une opération est une implémentation d'un service qui peut être demandé à tous les objets d'une même classe dans le but de déclencher un comportement, ou il correspond à une abstraction de ce que peut réaliser un objet et qui est réalisable par tous les objets de la classe.

◆ *Responsabilités :*

Une responsabilité est un contrat ou une obligation qu'une classe doit respecter. Donc, elles correspondent à l'état et au comportement qui sont similaires à tous les objets d'une classe. Les responsabilités est un exemple de stéréotype prédéfini qu'on peut représenter dans une note.



Operations

Figure 39 : *Exemple d'une classe*

III.1.6. Classe paramétrable :

Les classes paramétrables sont des modèles de classes. Ces dernières doivent être instanciées avec des paramètres pour obtenir une classe concrète. UML représente une classe paramétrable de la façon suivante :

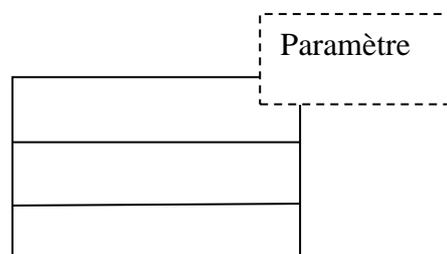


Figure 40 : *La représentation graphique d'une classe paramétrée dans l'UML*

III.1.7. Une interface :

Une interface est un ensemble d'opérations utilisées pour définir un service d'une classe ou d'un composant. Elle est représentée par un cercle, mais elle peut être représentée par une classe stéréotypée en plaçant le mot clé <<interface>> avant le nom de la classe. Dans tous les cas, doit être accompagnée d'une classe qui implémente les opérations définies par l'interface.

Caractéristiques d'une interface

◆ *Opération :*

Les opérations peuvent être décorées avec des propriétés de visibilité, des stéréotypes, des étiquettes et des contraintes.

◆ *Relations :*

Une interface peut participer à des relations de généralisation, d'association et de dépendance.



Figure 41 : *Représentation graphique d'une classe interface en UML [Booch & all, 00]*

III.1.8. Classe d'association :

Elle permet d'ajouter des attributs et des opérations à l'association. Elle est représentée dans l'UML par une classe reliée par un trait en pointillée sur l'association concernée.

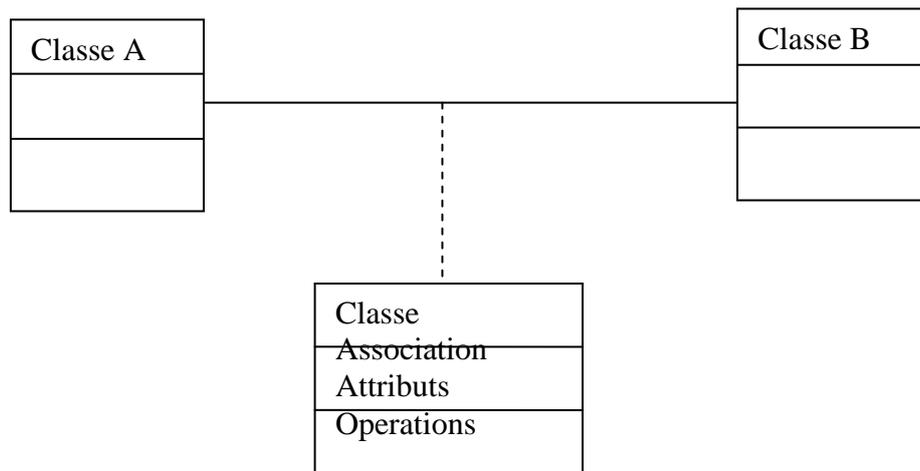


Figure 42 : la représentation graphique d'une classe d'association dans l'UML

III.1.9. Classe abstraite :

Elle sert de spécification et elle n'est pas instanciable. Elle est représentée graphiquement dans l'UML par une classe dont le nom est <<classe abstraite>> et est notée en italique.

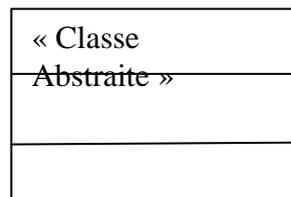


Figure 43 : La représentation de la classe abstraite dans l'UML

III.1.10. Classe active :

Un objet actif est un objet qui possède un processus ou un thread et qui peut déclencher une activité de contrôle. Une classe active est une classe dont les instances sont des objets actifs. Une classe active est représentée par un rectangle aux lignes épaisses. Les processus et les threads sont représentés par des classes stéréotypées. UML définit deux stéréotypes standards qui s'appliquent aux classes actives :

- Process : spécifie un flot qui peut s'exécuter en concurrence avec d'autres processus.
- Thread : spécifie un flot qui peut s'exécuter en concurrence avec d'autres threads à l'intérieur du même processus.

III.1.11. Dépendance :

Une dépendance est une relation d'utilisation qui établit qu'un changement de spécification d'un élément peut en affecter un autre qui l'utilise, mais l'inverse n'est pas nécessairement vrai. Elle sert à démontrer qu'un élément en utilise un autre. Elle est symbolisée par une flèche en pointillés, dirigée vers l'élément à l'égard duquel existe la dépendance.



Figure 44 : Exemple d'une relation de dépendance

III.1.12. Généralisation :

Une généralisation est une relation de spécialisation/généralisation selon laquelle les attributs de l'élément spécialisé (l'enfant) peuvent se substituer aux attributs de l'élément généralisé (le parent). De cette manière, l'enfant partage la structure et le comportement du parent. Elle est représentée par une flèche.

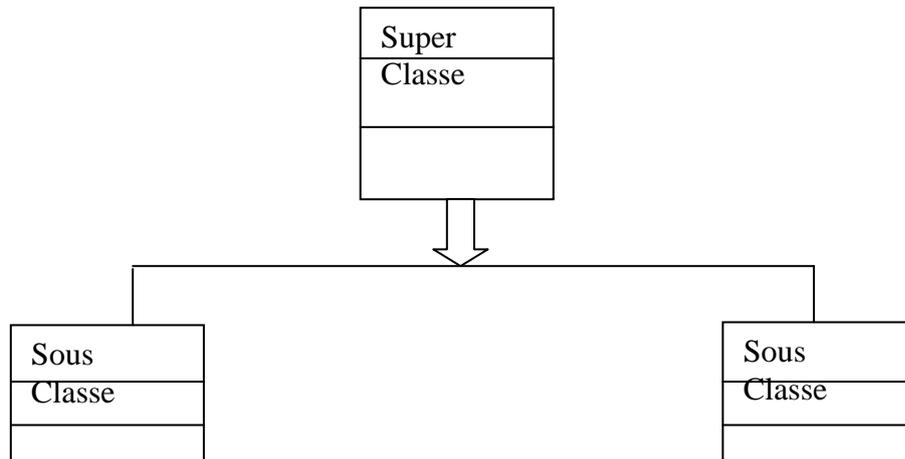


Figure 45: Représentation de la généralisation en UML

III.1.13. Association :

Une association est une relation structurelle qui décrit un ensemble de liens constituant une relation entre différents objets. Elle est représentée par une ligne qui peut être fléchée, elle comprend parfois une étiquette et souvent d'autres décorations, comme la multiplicité et les noms des rôles.



Figure 46 : Exemple d'une association

III.1.14. Agrégation :

Une agrégation est un type particulier d'association, qui représente une relation structurelle entre un tout et ses parties.

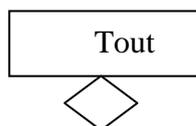


Figure 47: Schéma d'une agrégation en UML

III.1.15. Les restrictions :

Elles consistent à sélectionner un sous-ensemble d'objets qui participent à l'association. La sélection des objets est caractérisée par un attribut que l'on appelle clé. Tous les objets qui vérifient la clé feront partie de l'association. Elle est représentée graphiquement comme suit :

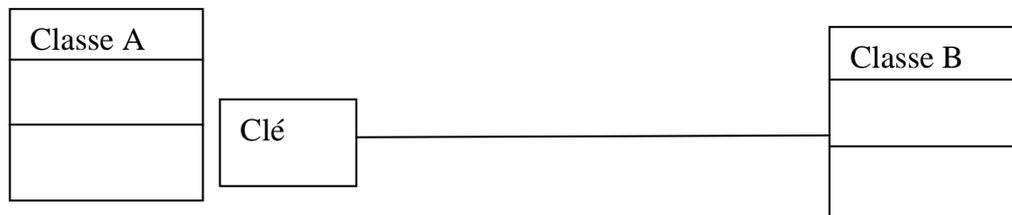


Figure 48: La représentation graphique d'une restriction UML

III.1.16. Les collaborations :

Une collaboration est un ensemble de classes, d'interfaces et d'autres éléments qui travaillent ensemble pour fournir un comportement de coopération plus important que la somme de toutes ces parties. Elle est également la spécification de la manière dont un élément, tel qu'un classificateur (comme une classe, une interface, un composant, un nœud ou un cas d'utilisation) ou une opération est réalisée par un ensemble de classificateurs ou d'associations qui jouent un rôle spécifique et sont utilisées de façon spéciale. Elle est représentée par une éclisse en pointillés.

Caractéristiques d'une collaboration

Structure :

Les collaborations comportent deux aspects : une partie structurelle qui précise les classes, les interfaces et autres éléments qui travaillent ensemble pour réaliser la collaboration désignée et une partie comportementale qui précise la dynamique de l'interaction entre ces éléments.

Comportement :

En règle générale, on représente la partie structurelle d'une collaboration à l'aide d'un diagramme de classe, on en représente la partie comportementale par un diagramme d'interaction. Un diagramme d'interaction précise une interaction qui illustre un comportement composé d'un ensemble donné pour atteindre un objectif spécifique.

III.1.17. Les composants :

Un composant est une partie physique remplaçable d'un système qui fournit la réalisation d'un ensemble d'interfaces. Il est représenté par un rectangle avec des onglets.

Types de composant

- *Composants de déploiement* qui sont nécessaires et suffisants pour un système exécutable, comme les bibliothèques dynamiques (DLL) et les exécutables (EXE) etc.
- *Composants produits* par le développement qui sont essentiellement le résultat et les fichiers de données à partir desquels les composants de déploiement sont créés.

▪ *Les composants d'exécution*, comme par exemple un objet COM+ qui est instancié à partir d'une DLL.

UML définit cinq stéréotypes standards qui s'appliquent aux composants :

Exécutable : précise un composant qui peut être exécuté sur un nœud.

Library : précise une bibliothèque objet statique ou dynamique.

Table : précise un composant qui représente une table de base de données

File : précise un composant qui représente un document contenant un code source ou des données.

Document : précise un composant qui représente un document.

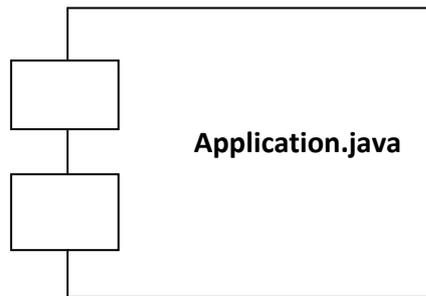


Figure 49. Exemple de composant

III.1.18. Les nœuds :

Un nœud est un élément physique qui existe au moment de l'exécution et représente une ressource de calcul. En règle générale, il a au moins de la mémoire et souvent des capacités de traitement. Un nœud est représenté par un cube.

Caractéristiques d'un nœud

Connexion :

C'est la connexion physique entre les nœuds, une ligne série ou un bus partagé. On peut même utiliser les associations pour modéliser des connexions indirectes, comme par exemple, une liaison satellite entre des processeurs éloignés.

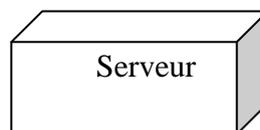


Figure 50. Exemple de serveur

III.1.19. Automate à états finis :

Est un comportement qui précise les séquences d'états d'un élément ou d'une interaction au cours de leur durée de vie, en réponse à des événements, ainsi que leurs réactions à ces événements. Le comportement d'une classe seule ou d'une collaboration de classes peut être précisé par un automate à états finis qui implique un certain nombre d'autres éléments, y compris des états, des transitions, des événements et des activités. Un état est schématisé par :

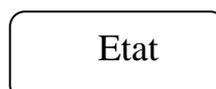


Figure 51 : Représentation d'un état en UML

III.2. Les diagrammes d'UML :

Un diagramme est la représentation graphique d'un ensemble d'éléments qui constituent un système. Les diagrammes servent à visualiser un système sous différentes perspectives et sont donc des projections dans un système.

UML distingue neuf diagrammes :

- Diagramme de classe,
- Diagramme d'objets,
- Diagramme de cas d'utilisation,
- Diagramme de séquences,
- Diagramme de collaborations
- Diagramme d'états-transitions,
- Diagramme d'activités,
- Diagramme de composants,
- Diagramme de déploiement.

III.2.1. Les diagrammes de classes :

Ils représentent un ensemble de classes, d'interfaces et de collaborations, ainsi que leurs relations. Ce sont les diagrammes les plus utilisés dans la modélisation des systèmes orientés objet. Ils présentent la vue de conception statique des systèmes. Par contre, ceux qui comprennent des classes actives présentent la vue de processus statique d'un système.

III.2.2. Les diagrammes d'objets :

Ils représentent un ensemble d'objets avec leurs relations. Ils correspondent aux vues statiques des instances des éléments qui apparaissent dans les diagrammes de classes. Comme les diagrammes de classes, ils présentent la vue de conception ou la vue de processus statique d'un système à partir de cas réels ou de prototypes.

III.2.3. Les diagrammes de cas d'utilisation :

Représentent un ensemble de cas d'utilisation et d'acteurs et leurs relations. Ils présentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système.

III.2.4. Les diagrammes de séquence :

Sont des diagrammes d'interaction qui mettent l'accent sur le classement chronologique des messages.

III.2.5. Les diagrammes de collaboration :

Sont des diagrammes d'interaction qui mettent l'accent sur l'organisation structurelle des objets qui envoient et reçoivent des messages.

Remarque :

Les diagrammes de séquence et les diagrammes de collaboration sont deux types de diagrammes d'interaction. Les diagrammes d'interaction représentent une interaction, c'est-à-dire, un ensemble

d'objets et leurs relations, y compris les messages qu'ils peuvent s'échanger. Ils présentent ainsi une vue dynamique du système. Les diagrammes de séquence et les diagrammes de collaboration sont isomorphes, c'est-à-dire que l'un peut être transformé en l'autre [Booch & all, 00].

III.2.7. Les diagrammes d'activités :

Un diagramme d'activités est un type particulier de diagramme d'états-transitions qui décrit la succession des activités au sein d'un système. Ils présentent la vue dynamique d'un système, ils sont particulièrement importants dans la modélisation de la fonction d'un système et mettent l'accent sur le flot de contrôle entre les objets.

IV. Les mécanismes généraux d'UML [Booch & all, 00] [26]

Une construction est plus simple et plus harmonieuse si elle se conforme à un ensemble de caractéristiques communes. Quatre mécanismes généraux qui s'appliquent au langage, le rendent plus simple à utiliser :

◆ Spécifications :

Derrière chaque élément de sa notation graphique, il existe une spécification qui fournit un énoncé textuel de la syntaxe et de la sémantique de cet élément.

Par exemple, derrière une icône de classe se trouve une spécification qui précise l'ensemble des attributs, des opérations et des comportements mobilisés par cette classe.

De ce fait, les spécifications UML fournissent une base sémantique qui renferme toutes les parties de tous les modèles d'un système. Ces parties étant en relation les unes avec les autres d'une manière cohérente.

◆ Décorations :

La plupart des éléments d'UML possèdent une notation graphique unique et directe qui fournit une représentation visuelle des aspects les plus importants de ces éléments. La notation de ces éléments (une classe par exemple) peut inclure d'autres informations, comme la visibilité de ses attributs et de ses opérations ou encore indiquer si elle est abstraite ou non.

◆ Distinctions communes :

Dans la modélisation des systèmes orientés objet, on opère, le plus souvent, au moins deux distinctions fondamentales.

- Tout d'abord il y a la distinction entre classe et objet. Une classe est une abstraction alors qu'un objet est la manifestation concrète de cette abstraction.
- Ensuite, interface et implémentation sont séparées. Une interface définit un contrat et une implémentation représente la réalisation concrète de ce contrat, représentant fidèlement la sémantique complète de l'interface.

◆ Mécanismes d'extensibilité :

Les mécanismes d'extensibilité d'UML permettent d'augmenter la portée du langage de manière bien précise. Ils incluent les stéréotypes, les étiquettes et les contraintes.

- *Un stéréotype* étend le vocabulaire d'UML, en permettant de créer de nouvelles sortes de briques de base qui dérivent de celles qui existent déjà mais qui sont spécifiques à un problème donné.
- *Une étiquette* étend les propriétés d'une brique de base d'UML, en permettant la création de nouvelles informations dans la spécification de cet élément.
- *Une contrainte* étend la sémantique d'une brique de base d'UML, en permettant l'ajout de nouvelles règles ou la modification de celles qui existent déjà.

On utilise ces mécanismes pour adapter UML aux besoins spécifiques d'un domaine ou d'une culture de développement.

Annexe B

Quelques exemple d'application des Framework et leurs guide d'installation :

Sofi Framework : [21]

Installation de SOFI Framework

Ouvrir un terminal et localisez-vous à l'endroit désiré pour extraire SOFI Framework.

Par exemple, localisez-vous :

- Sous Windows
 - C:\java\sofi

Extraire les sources de SOFI Framework

svn checkout <https://sofiframework.jira.com/svn/Framework/trunk> sofi-framework

Compilation

Par exemple, localisez-vous :

- Sous Windows

Création d'une application depuis le gabarit

Extraire les sources du gabarit

Par exemple, localisez-vous :

- Sous Windows
 - C:\java\sofi\
svn checkout https://sofiframework.jira.com/svn/GABARIT/trunk sofi-gabarit

Exporter les sources afin de créer un nouvel application

Par exemple, localisez-vous :

- Sous Windows
 - C:\java\sofi-gabarit
svn export sofiframework-gabarit-eclipse ../../demo

Générer un fichier de déploiement

Par exemple, localisez-vous :

- Sous Windows
 - C:\java\demo\ant
ant war

Déploiement de l'application

Téléchargement de Tomcat

Télécharger une version propre, si ce n'est déjà fait, du conteneur Tomcat :

<http://tomcat.apache.org/download-60.cgi>. Pour l'exemple ci-dessous décompresser l'archive sous le nom **apache-tomcat**.

Copie d'un serveur pour déploiement de l'application

Par exemple, localisez-vous dans la racine du dossier principal du Tomcat (ex. apache-tomcat), ensuite faire la copie des dossiers et fichiers dans un dossier de votre application.

- Sous Windows
xcopy c:\apache-tomcat c:\java\demo\server /s
- Sous Linux, Mac OS ou Unix
cp -R /apache-tomcat /java/demo/server/

Configuration de Tomcat

Avant de déployer votre application et l'exécuter dans Tomcat, vous devez spécifier quelques paramètres nécessaires à son bon fonctionnement. Par exemple, vous devez spécifier l'URL pour communiquer avec l'infrastructure SOFI.

Ouvrir le fichier **context_tomcat.txt** présent dans /demo/presentation et copier les propriétés suivantes dans le fichier **context.xml** présent dans /demo/server/config Prendre les propriétés suivantes :

```
<Environment
  name="infrastructure.url"
  type="java.lang.String"
  override="false"
  value="http://http://www.sofiframework.org/authentification/distant"
/>
```

```
<Environment
  name="hibernate.show_sql"
  type="java.lang.String"
  override="false"
  value="true" />
```

```
<Environment
  name="hibernate.dialect"
  type="java.lang.String"
  override="false"
  value="org.hibernate.dialect.HSQLDialect" />
```

- C:\java\sofi\sofi-framework\projet\ant
ant publish

Localisation des librairies sur votre poste

Afin de consulter les libraires qui sont publié localement sur votre poste, il suffit d'ouvrir un navigateur de fichiers et vous localiser sur :

Sous Windows :

Toutes les librairies nécessaires au développement de projet SOFI se retrouve dans le dossier cache :
La version de SOFI publié localement sur votre poste se retrouve dans le dossier local. À noter que ces librairies ont priorité sur les libraires en cache reçu du répertoire des libraires.

- Copier dans context.xml tel que cet exemple :

```
<!-- Uncomment this to enable Comet connection tacking (provides
events
on session expiration as well as webapp lifecycle) -->
<!--
<Valve
className="org.apache.catalina.valves.CometConnectionManagerValve"
/>
-->

<Environment
name="infrastructure.url"
type="java.lang.String"
override="false"
value="http://www.sofiframework.org/authentication/distant" />

<Environment
name="hibernate.dialect"
type="java.lang.String"
override="false"
value="org.hibernate.dialect.HSQLDialect" />

<Environment
name="hibernate.show_sql"
type="java.lang.String"
override="false"
value="true" />

</Context>
```

Spring : [15]

Exemple d'application :

Spring Roo est un outil de SpringSource crée en 2009 et dont la version actuelle est la version 1.2.2
Cet outil possède plusieurs particularités :

- Capable de monter rapidement une application Web Java
- Dit « fullstack » qui s'occupe de toutes les couches d'une l'application
- Gère le code rébarbatif

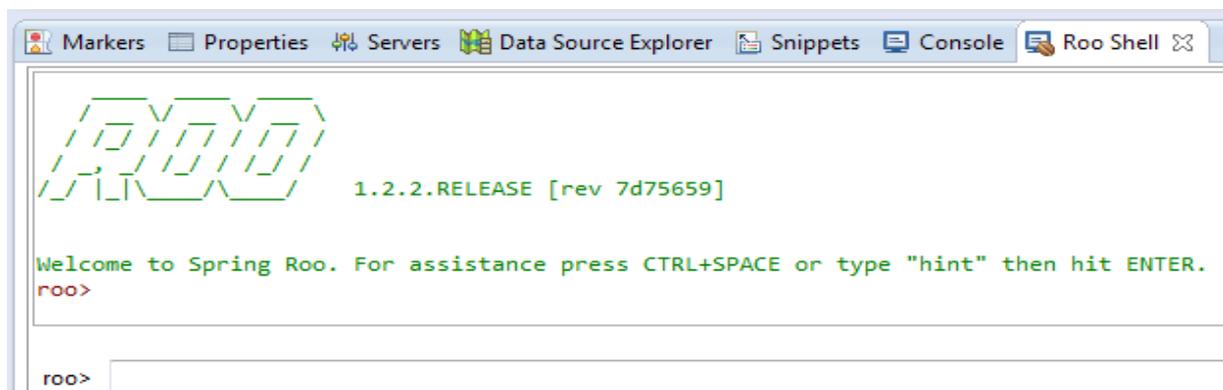
Démonstration :

Comment démarrer avec Spring ROO ?

Pour bien démarrer, il faut avoir le plugin Eclipse STS et créer un nouveau projet Spring ROO :

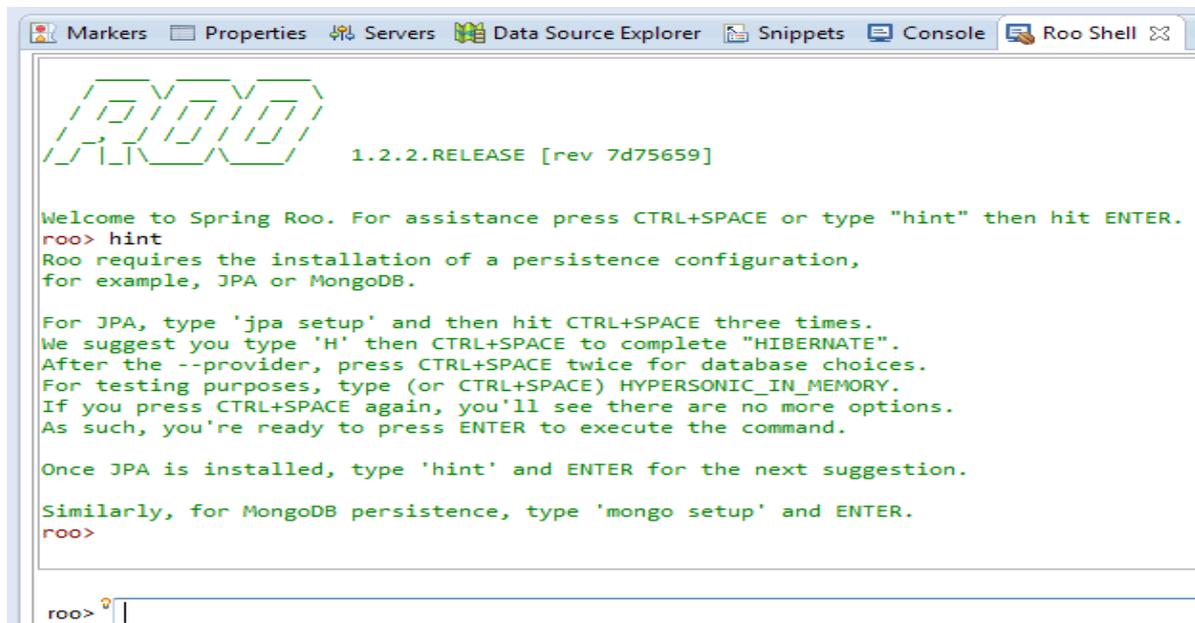
- ▲ Spring
 - Spring Bean Configuration File
 - Spring Bean Definition
 - Spring Project
 - Spring Roo Project
 - Spring Template Project
 - Spring Web Flow Definition File

C'est ensuite que le développement commence.



```
Markers Properties Servers Data Source Explorer Snippets Console Roo Shell
1.2.2.RELEASE [rev 7d75659]
Welcome to Spring Roo. For assistance press CTRL+SPACE or type "hint" then hit ENTER.
roo>
```

A la création du projet, une console s'ouvre. Celle-ci est le centre de Spring ROO.



```
Markers Properties Servers Data Source Explorer Snippets Console Roo Shell
1.2.2.RELEASE [rev 7d75659]
Welcome to Spring Roo. For assistance press CTRL+SPACE or type "hint" then hit ENTER.
roo> hint
Roo requires the installation of a persistence configuration,
for example, JPA or MongoDB.

For JPA, type 'jpa setup' and then hit CTRL+SPACE three times.
We suggest you type 'H' then CTRL+SPACE to complete "HIBERNATE".
After the --provider, press CTRL+SPACE twice for database choices.
For testing purposes, type (or CTRL+SPACE) HYPERSONIC_IN_MEMORY.
If you press CTRL+SPACE again, you'll see there are no more options.
As such, you're ready to press ENTER to execute the command.

Once JPA is installed, type 'hint' and ENTER for the next suggestion.

Similarly, for MongoDB persistence, type 'mongo setup' and ENTER.
roo>
```

Après avoir questionner la console avec la commande "hint", Spring nous propose nous occuper de la persistance en configurant notre base de donnée.

```
Markers Properties Servers Data Source Explorer Snippets Console Roo Shell

/  \  \
//  //  //
/  \  //  \
//  \  \  \  1.2.2.RELEASE [rev 7d75659]

Welcome to Spring Roo. For assistance press CTRL+SPACE or type "hint" then hit ENTER.
roo> hint
Roo requires the installation of a persistence configuration,
for example, JPA or MongoDB.

For JPA, type 'jpa setup' and then hit CTRL+SPACE three times.
We suggest you type 'h' then CTRL+SPACE to complete "HIBERNATE".
After the --provider, press CTRL+SPACE twice for database choices.
For testing purposes, type (or CTRL+SPACE) HYPERSONIC_IN_MEMORY.
If you press CTRL+SPACE again, you'll see there are no more options.
As such, you're ready to press ENTER to execute the command.

Once JPA is installed, type 'hint' and ENTER for the next suggestion.

Similarly, for MongoDB persistence, type 'mongo setup' and ENTER.
roo>
required --database: The database to support; no default value

required --provider: The persistence provider to support; no default value
roo> jpa setup --database POSTGRES --provider HIBERNATE
Created SPRING CONFIG ROOT\database.properties
Please update your database details in src/main/resources/META-INF/spring/database.properties.
Updated SPRING CONFIG ROOT\applicationContext.xml
Created SRC MAIN RESOURCES\META-INF\persistence.xml
Updated ROOT\pom.xml [added dependencies postgresql:postgresql:9.1-901.jdbc3, org.hibernate:hibernate-core:3.6.9.Final, org.hibernate:hibernate-entitymanager:3.6.9.Final,
org.hibernate.javax.persistence:hibernate-jpa-2.0-api:1.0.1.Final, commons-collections:commons-collections:3.2.1, org.hibernate:hibernate-validator:4.2.0.Final,
javax.validation:validation-api:1.0.0.GA, cglib:cglib-nodep:2.2.2, javax.transaction:jta:1.1, org.springframework:spring-jdbc:${spring.version}, org.springframework:spring-orm:${spring.version},
commons-pool:commons-pool:1.5.6, commons-dbcp:commons-dbcp:1.3]
roo>

roo> ?
```

Après plusieurs "Ctrl+Espace" et quelques informations renseignées, ROO nous a maché le travail en préparant la connexion à la base.

You can create entities either via Roo or your IDE.
Using the Roo shell is fast and easy, especially thanks to the CTRL+SPACE completion.

Start by typing 'ent' and then hitting CTRL+SPACE twice.
Enter the --class in the form '~.domain.MyEntityClassName'
In Roo, '~' means the --topLevelPackage you specified via 'create project'.

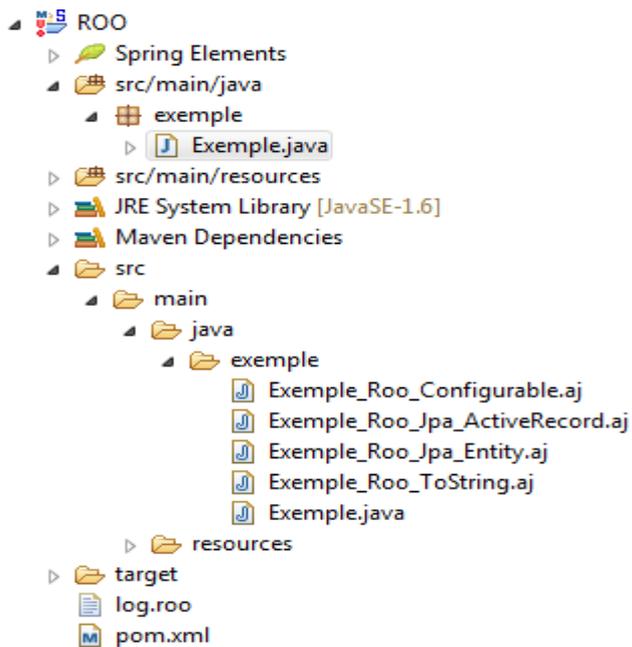
After specify a --class argument, press SPACE then CTRL+SPACE. Note nothing appears.
Because nothing appears, it means you've entered all mandatory arguments.
However, optional arguments do exist for this command (and most others in Roo).
To see the optional arguments, type '--' and then hit CTRL+SPACE. Mostly you won't
need any optional arguments, but let's select the --testAutomatically option
and hit ENTER. You can always use this approach to view optional arguments.

After creating an entity, use 'hint' for the next suggestion.

```
roo>
```

```
roo> |
```

- ROO nous propose maintenant de créer notre première classe, nous allons taper "*entity jpa --class exemple.exemple*" pour créer notre première classe **Exemple** :



Les différents fichiers .aj présent permettent de générer le code rébarbatif. J'ajoute deux propriétés à notre classe Exemple :

```

package exemple;

import org.springframework.roo.addon.javabean.RooJavaBean;

@RooJavaBean
@RooToString
@RooJpaActiveRecord
public class Exemple {

    private String chaine;
    private int nombre;
}

// WARNING: DO NOT EDIT THIS FILE. THIS FILE IS MANAGED BY SPRING ROO.

package exemple;

import exemple.Exemple;

privileged aspect Exemple_Roo_Jpa_Entity {

    declare @type: Exemple: @Entity;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long Exemple.id;

    @Version
    @Column(name = "version")
    private Integer Exemple.version;

    public Long Exemple.getId() {
        return this.id;
    }

    public void Exemple.setId(Long id) {
        this.id = id;
    }

    public Integer Exemple.getVersion() {
        return this.version;
    }

    public void Exemple.setVersion(Integer version) {
        this.version = version;
    }
}

// WARNING: DO NOT EDIT THIS FILE. THIS FILE IS MANAGED BY SPRING ROO.

package exemple;

import exemple.Exemple;

privileged aspect Exemple_Roo_ToString {

    public String Exemple.toString() {
        return ReflectionToStringBuilder.toString(this, ToStringStyle.SHORT_PREFIX_STYLE);
    }
}

```

Les getters et setters sont donc générés automatiquement dans le fichier Exemple_Roo_Jpa_Entity.aj et le toString() dans le fichier Exemple_Roo_ToString.aj .

Voilà donc un bref exemple de l'utilisation de Spring ROO pour créer une application WEB simplement et rapidement.

Qt Jambi : [17]

Installation Sur Windows

- Téléchargez Qt-Jambi sur le lien suivant : ftp://ftp.qtsoftware.com/qtjambi/source/qtjambi-win32-lgpl-4.5.2_01.zip
- Extraire le contenu de l'archive.
- Lancer qtjambi.exe : cela lance la démo de Qt-Jambi et paramètre votre environnement pour utiliser Qt-Jambi.
- Vous pouvez aussi accéder à la documentation de Qt-Jambi à travers l'outil de démo ou bien en lançant le fichier index.html dans doc\html\index.html.
- Vous pouvez lancer Qt-designer, le concepteur d'interface graphique de Qt-Jambi, en double cliquant sur designer.bat.
- Afin d'expliquer à l'environnement de travail comment comprendre et utiliser les méthodes de Qt-Jambi modifiez la variable d'environnement PATH. Sur Windows Vista/Seven, vous devez vous rendre dans Panneau de Configuration puis dans Système. Cliquez ensuite sur Paramètres systèmes avancés puis sur le bouton Variables d'environnement... Sélectionnez ensuite la variable PATH et cliquez sur Modifier... Saisissez ensuite, sans modifier ou supprimer les valeurs présentes, le chemin vers le dossier \bin, précédé d'un point-virgule. Par exemple : C:\Users\test\Desktop\Logiciels\qtjambi-win32-lgpl-4.5.2_01\bin. Il faut ensuite modifier la variable CLASSPATH de la même manière et ajouter cette fois le jar. Par exemple : C:\Qt-Jambi\qtjambi-win32-lgpl-4.5.2_01\qtjambi-4.5.2_01.jar.
- Validez ensuite votre choix.

Intégration dans Eclipse

- => Attention, afin de pouvoir utiliser Eclipse vous aurez besoin d'un JRE.
- Afin d'intégrer Qt-Jambi à Eclipse, il suffit d'ajouter à vos projets en Qt-Jambi une « Referenced Librairies ». Pour cela, faites un clic droit sur votre projet en Qt-Jambi. Dans le menu contextuel choisissez Build Path puis Add External Archives... Sélectionnez ensuite qtjambi.jar (le nom de l'archive qtjambi.jar est souvent suivie de son numéro de version) et validez.

Bibliographie :

- [1] Julien Dubois, Jean-Philippe Retailé, Thierry Templier, Eyrolles, *Spring par la pratique* 2006,517 p.
- [2] Gary Mak, PEARSSON, *Spring* , 2008, 492 p.,
- [3] Ken Chazotte, *Build Your Own Framework with Visual FoxPro*, Hentzenwerke , 2004.
- [4] Nicolas Guelfi, Egidio Astesiano et Gianna Reggio, *Scientific engineering for distributed Java applications*, Springer - 2003,
- [5] Jean-Francois Brassard resolved FRAMEWORK-196 - Problème de rafraîchissement journalier des caches dans une application Spring as 'Fixed'FRAMEWORK-196 Publié par : Jean-Francois Brassard | 2014-02-06
- [6] WaveMaker ressorts pour VMware. WaveMaker. 8 mars 2011.
- [7] Pramati acquiert WaveMaker . Pramati Technologies . 01/05/2013.
- [8] WaveMaker, Visual Studio 4.0 AJAX , com
- [9] Mémoire promotion 2008/2009 réalisé par Melle Kerriche Hajila & Melle Lehad Lynda. Dirigé par M. Daoui.
- [10] holer pflugmann TCP/IP ; Edition micro application février 2001.
- [11] G. A. Leirier et R. Scoll ; Grand livre PHP 4 & MYSQL ; Edition Micro Application ; 2000.

- [12] Merise vers OMT et UML, Josepf Gabay, Edition Interedition 1998
- [13] Le guide de l'utilisateur UML.

Webliographie:

- [14] www.spring.io 15/05/2014
- [15] <https://src.springsource.org/svn/spring-maintenance/tags/release-1-0/changelog.txt> 12/05/2014
- [16] ego.developpez.com/spring 15/05/2014
- [17] ftp://ftp.qtsoftware.com/qtjambi/source/qtjambi-win32-lgpl-4.5.2_01.zip 20/05/2014
- [18] <http://www.siteduzero.com/tutoriel-3-3158-du-qt-en-java-avec-qt-jambi.html> 20/05/5/2014
- [19] http://doc.qt.nokia.com/qtjambi-4.5.2_01/index.html 20/05/2014
- [20] <ftp://ftp.qtsoftware.com/qtjambi/source/> 20/05/2014
- [21] <http://www.sofiframework.org>. 17/05/2014
- [22] <http://www.Wikipédia.com> 10/05/2014
- [23] <http://www.commentcamarche.com> 11/05/2014
- [24] <http://www.commentcamarche.net/contents/internet/tcpip.php3> 04/05/2014
- [25] <http://www.Uml.free.fr> 25/05/2014
- [26] Le site du zero. 10/05/2014
- [27] <http://www.apache.com> 25/05/2014