

Ministère des Enseignements Secondaire et Supérieur
(MESS)

Secrétariat Général

Université Polytechnique de Bobo-Dioulasso (U.P.B.)

Ecole Supérieure d'Informatique (E.S.I)



Cycle des Ingénieurs de Travaux Informatiques (C.I.T.I)
Option : Analyse et Programmation (A.P)

Rapport de fin de cycle

Gestion des stocks des fournitures de bureau du Centre MURAZ

Période du 1^{er} octobre 2013 au 31 décembre 2013

Auteurs : KAFANDO Tégawendé. Beranger et SOME Kpèssiou Thierry

Maître de stage

M. N. Félix KAFANDO
DAF du Centre MURAZ

Superviseur

Dr. OUATTARA/COMPAORE
Anasthasie Joëlle

Enseignante chercheur à l'École
Supérieure d'Informatique

Année académique : 2012-2013

Vu le 31/07/2014

DEDICACE

A nos chers parents.

A tout le personnel du Centre MURAZ.

REMERCIEMENTS

Nos remerciements vont à l'endroit de toute l'équipe pédagogique de l'École Supérieure d'Informatique (ESI) et des intervenants professionnels responsables de la formation d'ingénieurs de travaux informatiques, pour avoir assuré la partie théorique de celle-ci.

Nous remercions également toute l'équipe professionnelle du Centre MURAZ pour l'expérience enrichissante qu'elle nous a fait vivre durant ces trois (03) mois au sein du Centre.

Nous tenons également à remercier les personnes suivantes:

- Pr. Nicolas MEDA, DG du Centre MURAZ
- M. Félix N. KAFANDO, DAF du Centre MURAZ, notre maître de stage;
- Dr Joëlle A. OUATTARA, Enseignante chercheur à l'ESI, notre superviseur;
- Mme Kadiatou BAYALA, la Responsable logistique et du patrimoine du Centre MURAZ
- Dr Sadouanouan MALO, Enseignant chercheur à l'ESI

SOMMAIRE

DEDICACE	I
REMERCIEMENTS.....	II
SOMMAIRE	III
SIGLES, ABREVIATIONS ET ACRONYMES	VII
LISTE DES FIGURES	IX
LISTE DES TABLEAUX.....	X
PREAMBULE	XI
RESUME	XII
INTRODUCTION GENERALE	1
CHAPITRE I: PRESENTATION GENERALE.....	2
I. Le Centre MURAZ	4
I.1. Historique.....	4
I.2. Missions et objectifs	5
a. Missions	5
b. Objectifs.....	5
I.3. Organisation.....	5
II. Présentation du thème	7
II.1. Problématique.....	7
II.2. Résultats attendus	7
III. Approche de résolution	8
III.1. Outils d’analyse et de conception.....	8
III.1.1. Langage de modélisation	8
III.1.2.Processus de développement logiciel.....	8

III.2. Planning prévisionnel.....	9
III.2.1. Détermination de la liste des tâches.....	10
III.2.2. Diagramme de GANTT prévisionnel.....	10
III.2.3. Diagramme de GANTT réel.....	11
III.2.4. Explication des écarts.....	11
IV Acteurs du projet.....	11
IV.1. Groupe de pilotage.....	12
IV.2. Groupe de projet.....	12
IV.3. Groupe des utilisateurs.....	13
CHAPITRE II:ANALYSE.....	14
I. Analyse des besoins.....	16
I.1. Spécifications fonctionnelles.....	16
I.1.1.Interviews des utilisateurs.....	16
I.1.2.Recueil des fonctionnalités du futur système.....	18
I.1.3.Identification des acteurs du système.....	18
I.1.4. Identification des cas d'utilisation.....	18
I.2. Spécifications techniques.....	19
I.2.1. Plateforme de développement.....	19
I.2.2. SGBD.....	20
I.2.3.Serveur d'applications.....	20
I.2.4.Solution d'ORM.....	20
I.2.5.Méthode de calcul du coût de développement.....	21
II. Étude des scenarii.....	22
II.1. Architecture réseau.....	22

II.1.1. Formalisme.....	22
II.1.2. Premier scénario.....	23
II.1.3. Deuxième scénario.....	24
II.2. Estimation des coûts de réalisation.....	25
II.3. Comparaison des scénarii.....	26
II.4. Scénario retenu.....	27
CHAPITRE III :CONCEPTION.....	28
I. Diagramme de cas d'utilisation / Diagramme de séquences.....	30
I.1. Diagramme de cas d'utilisation.....	30
I.1.1. Représentation du diagramme de cas d'utilisation.....	30
I.1.2. Documentation de certains cas d'utilisation.....	31
I.2. Diagramme de séquences.....	34
I.2.1. Cas d'utilisation « S'authentifier».....	34
I.2.2. Cas d'utilisation « Gérer les Articles ».....	35
I.2.3. Cas d'utilisation « Gérer les EB ».....	36
II. Diagramme de classes.....	37
II.1. Représentation du diagramme de classes.....	37
II.2. Description de quelques classes.....	40
II.2.1. Description de la classe « Article ».....	40
II.2.2. Description de la classe « EXPRESSION_BESOINS ».....	40
CHAPITRE IV:REALISATION.....	42
I. Présentation de quelques écrans.....	44
I.1. Connexion.....	44
I.2. Accueil.....	45

I.3. Enregistrement d'un nouvel article.....	46
I.4. Enregistrement d'une nouvelle expression de besoin.....	46
I.5. Enregistrement d'une sortie d'article	47
I.6. Liste des articles.....	48
II. Politique de sécurité.....	49
II.1. Protection contre les virus	49
II.2. Sauvegarde et restauration.....	49
CONCLUSION GENERALE.....	50
Bibliographie et webographie.....	i
Bibliographie:	i
Webographie:.....	i
ANNEXES.....	ii
Annexe 1:Présentation de JSF	ii
Annexe2: Présentation de Hibernate.....	iv
Annexe 3 :Présentation de PostgreSQL.....	vi

SIGLES, ABREVIATIONS ET ACRONYMES

SIGLE	DEFINITION
2TUP	Two Tracks Unified Process
BD	Base de données
CICI	Cycle d'Ingénieurs de Conception Informatique
CITI	Cycle d'Ingénieurs de Travaux Informatiques
COCOMO	Constructive Cost Model
DAF	Directeur des Affaires Financières
EB	Expression de Besoins
ESI	École Supérieure d'Informatique
HM	Homme-Mois
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
JSF	Java Server Faces
JSP	Java Server Pages
KILS	Kilo Ligne d'Instructions dans le programme source
ORM	Object Relational Mapping
RD	Responsable du Département
RLP	Responsable Logistique et du Patrimoine

SGBD	Système de Gestion de Base de Données
SGBDRO	Système de Gestion de Base de Données Relationnelles Objet
SGSF-MURAZ	System de Gestion des Stocks de Fournitures de bureau du centre MURAZ
SQL	Structured Query Language
TIC	Technologies de l'Information et de la Communication
UML	Unified Modeling Language
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

LISTE DES FIGURES

Figure 1.1: Organigramme du Centre MURAZ.....	6
Figure 1.2: Illustration du fonctionnement de 2TUP	9
Figure 1.3 : Liste des tâches.....	10
Figure 1.4 : Diagramme de GANTT prévisionnel.....	10
Figure 1.5 : Diagramme de GANTT réel.....	11
Figure 2.1 : Symboles utilisés pour la représentation de l'architecture réseau.....	22
Figure 2.2 : Schéma architectural réseau du premier scénario	23
Figure 2.3 : Schéma architectural du deuxième scénario	24
Figure 3.2 : Diagramme de séquence du cas d'utilisation «S'authentifier »	34
Figure 3.3 : Diagramme de séquence du cas d'utilisation « Gérer les Articles ».....	35
Figure 3.4 : Diagramme de séquence du cas d'utilisation <<Gérer les EB>>.....	36
Figure 3.5: Diagramme de classes	39
Figure 4.1 : Écran de connexion	44
Figure 4.2 : Écran de connexion après un échec d'authentification	45
Figure 4.3 : Écran d'accueil.....	46
Figure 4.4 : Écran d'enregistrement d'un nouvel article	46
Figure 4.5 : Écran d'enregistrement d'une nouvelle expression de besoin	47
Figure 4.6 : Écran d'une sortie d'article.....	48
Figure 4.7: Écran de la liste des articles enregistrés	48
Figure A1.1: Exemple de hiérarchie de composants JSF	ii
Figure A1.2: Étapes de traitements d'une requête pour l'affichage d'une page web par JSF	iii
Figure A2.1 : Architecture de Hibernate.....	v

LISTE DES TABLEAUX

Tableau 2.1:Liste des cas d'utilisation.....	19
Tableau 2.2:Formules de calcul proposées par COCOMO	21
Tableau 2.3 : Cout de réalisation de l'application	25
Tableau 3.1:Description textuelle du cas d'utilisation «S'authentifier ».....	31
Tableau 3.2:Description textuelle du cas d'utilisation « Gérer les Articles »	32
Tableau 3.3:Description textuelle du cas d'utilisation « Gérer les EB ».....	33
Tableau 3.4:Description de la classe «Article».....	40
Tableau 3.5:Description de la classe « EXPRESSION_BESOINS ».....	40

PREAMBULE

L'Université Polytechnique de Bobo-Dioulasso a été créée le 23 mai 1997. Elle est située à quinze (15) kilomètres à l'Ouest de Bobo et est composée de six (06) établissements dont l'École Supérieure d'Informatique (ESI) où nous suivons notre formation.

L'ESI a l'organisation pédagogique suivante :

- La formation au Cycle des Ingénieurs de Travaux Informatiques (CITI) avec les options : Analyse et Programmation (AP) et Réseau et Maintenance Informatique (RéMI);
- La formation au Cycle des Ingénieurs de Conception en Informatique (CICI).

La formation au cycle des ingénieurs de travaux informatiques option Analyse et Programmation a pour objectif de former des cadres moyens opérationnels et évolutifs qui sont aptes à participer efficacement à la conception, à la réalisation et à la maintenance d'applications informatiques. Cette formation aboutit au diplôme d'ingénieur de travaux informatiques.

Ce diplôme n'est accordé qu'aux étudiants ayant validé trois (03) années d'études, et démontré leurs capacités lors d'un stage de trois (03) mois en entreprise.

C'est dans ce contexte, que nous avons été accueillis au Centre MURAZ pour notre stage de fin de cycle.

RESUME

Dans un univers hautement concurrentiel, la gestion des informations dans l'entreprise moderne est devenue un véritable enjeu stratégique. En effet, afin de créer et/ou de maintenir ses avantages, l'entreprise se doit d'optimiser la gestion des informations qui transitent en son sein.

L'optimisation de cette gestion des informations passe essentiellement par la collecte et la centralisation de celles-ci d'une part, d'autre part, par le traitement et l'analyse des informations collectées, cela, pour prendre les meilleures décisions. De plus, la complexité de cette optimisation varie en fonction de la taille et du type d'informations que traite l'entreprise.

Le travail à effectuer demande un effort d'analyse, de conception et de réalisation d'un logiciel facilitant la gestion des stocks de fournitures de bureau du Centre MURAZ. Le processus de développement utilisé est Two Tracks Unified Process (2TUP), une méthode d'analyse et de conception basée sur le langage de modélisation graphique Unified Modeling Language (UML). La technologie utilisée est Java Server Faces (JSF) [4] [7], un Framework Java Enterprise Edition (JEE) [1][2] [7] qui est couplé avec la librairie de composants graphiques PrimeFaces [5] [6] [9]. L'abstraction de la couche d'accès aux données est assurée par le Framework de mapping objet/relationnel Hibernate [3] [8].

INTRODUCTION GENERALE

L'automatisation des traitements de données constitue aujourd'hui un enjeu stratégique pour les entreprises et les institutions. Elle constitue un avantage concurrentiel important et d'efficacité pour celles qui en ont la maîtrise.

Conscient de ces enjeux, le Centre MURAZ s'est orienté vers l'automatisation de certains modules de gestion dont la gestion des stocks de fournitures de bureau. C'est ainsi que nous avons été accueillis en son sein afin de mener une étude pour l'automatisation de ce dernier. Cette automatisation implique un certain nombre d'opérations plus ou moins complexes. La gestion des stocks étant un module nécessitant une certaine exigence et rigueur, le souci est de réaliser une application performante, conviviale et facile à utiliser.

Ce rapport résume notre travail au sein du Centre MURAZ. Dans la suite, nous présenterons d'abord l'analyse du système après avoir décrit le cadre général du thème. Ensuite, nous étalerons sa conception détaillée. Enfin nous présenterons la phase de réalisation de l'application.

CHAPITRE I : PRESENTATION GENERALE

Introduction

La conduite d'un projet dans une structure nécessite préalablement une bonne connaissance de cette dernière, la maîtrise du thème d'étude ainsi que l'utilisation d'une méthode de travail.

Dans ce chapitre, il sera donc question de projeter une vue générale du contexte de ce projet. De la présentation de la structure d'accueil, nous aboutirons sur la problématique suscitée par ce projet. De là, les résultats attendus seront bien cernés et nous pourrons ainsi adopter une approche de résolution.

I. Le Centre MURAZ

I.1. Historique

Le Centre MURAZ est un Centre d'excellence et d'innovation dans le domaine de la recherche et de la santé. Le Centre MURAZ répond aux besoins en formation et recherche aussi bien en sciences humaines que biomédicales sur le plan national et international grâce à la multidisciplinarité de ses chercheurs.

Créé en 1939, Il se dénommait alors "Service Général Autonome de la Maladie du Sommeil (SGAMS)" et son objectif premier était de faire reculer la trypanosomiase humaine ou maladie du sommeil qui sévissait dans toute l'Afrique Occidentale Française (AOF). Le SGAMS devint en 1945 le Service Général d'Hygiène Mobile et de Prophylaxie qui prendra en 1956 la dénomination de "Centre MURAZ" en hommage à son premier directeur et fondateur, le médecin militaire Français Gaston MURAZ décédé en 1955. Intégré à l'Organisation de Coopération et de Coordination pour la lutte contre les Grandes Endémies (OCCGE) de 1960 à l'an 2000, le Centre MURAZ a connu un rayonnement international mémorable. Il devint, à part entière, en 2001 un institut public burkinabè de recherche pour la santé.

Comme résultats des recherches du Centre MURAZ, on peut citer entre autres :

- le déploiement par l'OMS des programmes de contrôle de la trypanosomiase, de la dracunculose et de l'onchocercose, particulièrement dans les vallées de la Volta à partir des résultats scientifiques probants fournis par le Centre MURAZ et ses partenaires de l'Institut de Recherche pour le Développement (IRD) ;
- l'évaluation des meilleures stratégies de prévention de la transmission mère-enfant du VIH qui fondent aujourd'hui les programmes nationaux de prévention de la transmission mère-enfant du VIH sous l'égide de l'UNICEF ;
- la contribution du Centre MURAZ et de l'Institut de Recherche en Sciences de la Santé (IRSS) pour la validation du premier vaccin préventif efficace contre le paludisme en collaboration avec d'autres institutions africaines.

I.2. Missions et objectifs

a. Missions

Trois missions essentielles sont dévolues au Centre MURAZ. Il s'agit de la recherche dans le domaine de la santé, de la formation des personnels de santé et du renforcement des capacités des compétences nationales (en priorité) de sorte à disposer sur le plan technique d'expert de haut niveau. De plus le centre est sollicité par des organismes tels que l'OMS pour l'animation de séminaires de formation, le contrôle de qualité de laboratoire, etc.

b. Objectifs

Les principaux objectifs du Centre MURAZ sont de :

- Promouvoir et harmoniser les recherches fondamentales, appliquées ou opérationnelles sur les maladies transmissibles ;
- Promouvoir la lutte contre ces maladies transmissibles ou tout problème de santé publique défini par le gouvernement du Burkina Faso et, éventuellement, sur les recommandations des organisations internationales compétentes ;
- Collaborer avec les autres services nationaux pour recueillir et diffuser l'information sanitaire pertinente et la mettre à la disposition des autorités sanitaires nationales ;
- Disposer en permanence d'une expertise technique de haut niveau développée prioritairement à partir des compétences nationales ;
- Participer à la formation des personnels de santé ;
- Participer dans la mesure de ses moyens, à la coordination de la lutte contre les maladies transmissibles relevant de ses domaines de compétence avec les autres pays de la sous-région et tout particulièrement avec les États voisins.

I.3. Organisation

Le Centre MURAZ est constitué d'organes d'administration et de direction qui sont le Conseil d'Administration, la Direction Générale, les cadres de concertation et les organes consultatifs. Son organigramme est représenté par la figure 1.1.

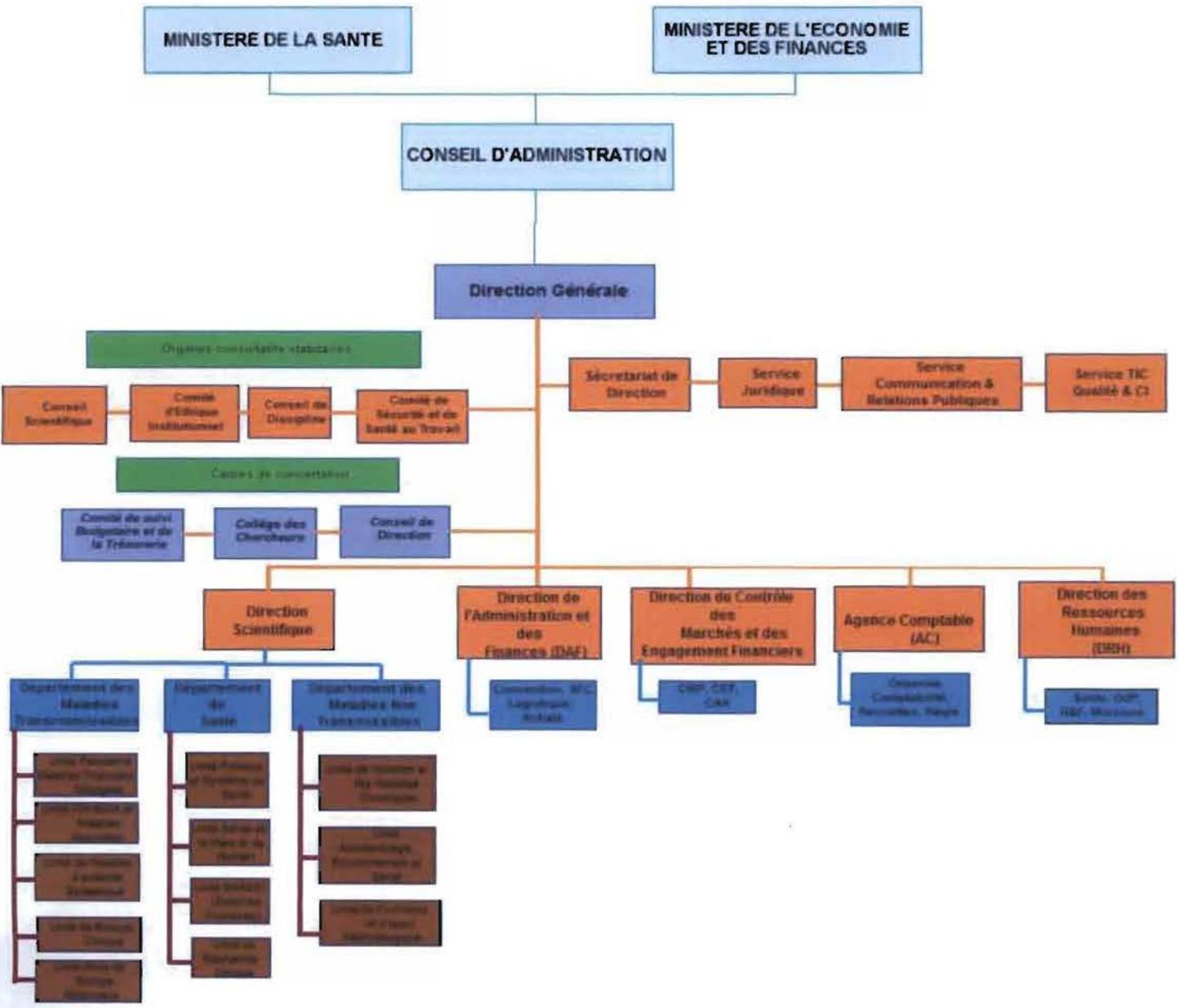


Figure 1.1 : Organigramme du Centre MURAZ

II. Présentation du thème

II.1. Problématique

Présentement la gestion des stocks de fournitures de bureau du Centre MURAZ se fait manuellement. Cette gestion manuelle pose de nombreuses difficultés à savoir :

- Difficulté pour le suivi du stock de fournitures de bureau ;
- La complexité dans l'établissement de l'état du stock tant au niveau de chaque direction qu'au niveau global ;
- La pérennité de l'information traitée, surtout en cas de sinistres ;
- La lenteur dans le traitement de l'information, que ce soit dans la transmission des informations ou pour une recherche dans les données collectées ;
- La difficulté dans l'analyse des données collectées.

Outre ces problèmes, il faut aussi souligner celui de la convivialité du système. En fin il est important de noter la difficulté due au fait que la gestion est multi-stock.

Afin de pallier ces problèmes et optimiser la gestion, le Centre MURAZ nous a confié la mission de développer une solution logicielle simple, conviviale et robuste qui lui permettra de gérer ses stocks de fournitures de bureau.

II.2. Résultats attendus

Face à cette problématique, le système à développer devra être capable de :

- donner l'état du stock d'article à tout moment
- prévenir le risque de rupture de stock
- suivre les mouvements de stocks
- prévoir les approvisionnements

III. Approche de résolution

III.1. Outils d'analyse et de conception

III.1.1. Langage de modélisation

Le développement de logiciels est une entreprise complexe. Notre mémoire à court terme étant fortement limitée, nous éprouvons le besoin naturel de modéliser pour simplifier et augmenter ainsi artificiellement notre capacité intellectuelle à résoudre les problèmes. Cette modélisation nécessite l'utilisation d'un langage permettant la description du système logiciel ainsi que sa compréhension par ses futurs utilisateurs. Pour ce faire, nous choisissons UML comme langage de modélisation de notre système, car il comble une lacune importante des technologies **objet**. Il permet d'exprimer et d'élaborer des modèles objets, indépendamment de tout langage de programmation. De plus, grâce à sa notation graphique, il permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions. Enfin, l'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.

III.1.2. Processus de développement logiciel

Afin d'obtenir un logiciel de qualité, qui répond aux besoins des utilisateurs en respectant les délais et les coûts prévus, nous utilisons le processus de développement Two Tracks Unified Process (2TUP)

2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Le processus s'articule ensuite autour de trois (03) branches essentielles :

- une branche technique ou branche de droite ;
- une branche fonctionnelle ou branche de gauche ;
- une branche de conception ou branche du milieu.

La figure 1.2 illustre le fonctionnement de 2TUP.

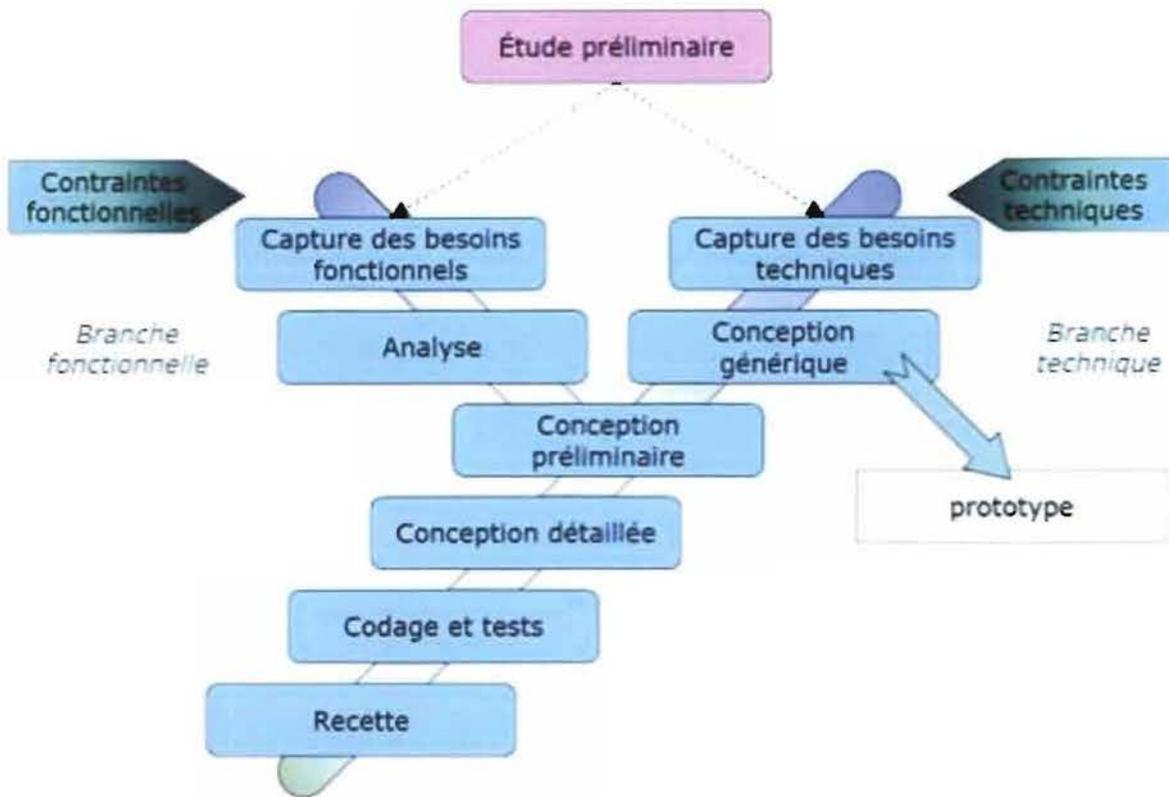


Figure 1.2 : Illustration du fonctionnement de 2TUP

III.2. Planning prévisionnel

La réalisation d'un projet passe par l'établissement et surtout le respect d'un planning prévisionnel bien défini en accord avec le groupe de pilotage. Ce planning doit tenir compte des contraintes liées à l'organisation interne de la structure d'accueil, du temps qui est imparti au groupe de projet et de la méthode d'analyse. Il doit permettre au groupe de projet de suivre l'avancée du projet.

III.2.1. Détermination de la liste des tâches

Afin de mieux gérer la complexité d'un travail, il convient de le diviser en de petites tâches plus faciles à comprendre et à réaliser. La figure 1.3 présente la liste des tâches qui composent notre projet.

Nom	Date de début	Date de fin
• Accueil et installation	01/10/13	02/10/13
• Etude préliminaire	03/10/13	09/10/13
• Capture des besoins	10/10/13	16/10/13
• Validation des besoins	17/10/13	21/10/13
• Etude détaillée	22/10/13	01/11/13
• Validation de l'étude détaillée	04/11/13	05/11/13
• Codage	06/11/13	17/12/13
• Test de l'application	18/12/13	19/12/13
• Rédaction du manuel d'utilisation	20/12/13	25/12/13
• Validation du manuel d'utilisation	26/12/13	26/12/13

Figure 1.3 : Liste des tâches

III.2.2. Diagramme de GANTT prévisionnel

La figure 1.4 représente le diagramme de GANTT prévisionnel.

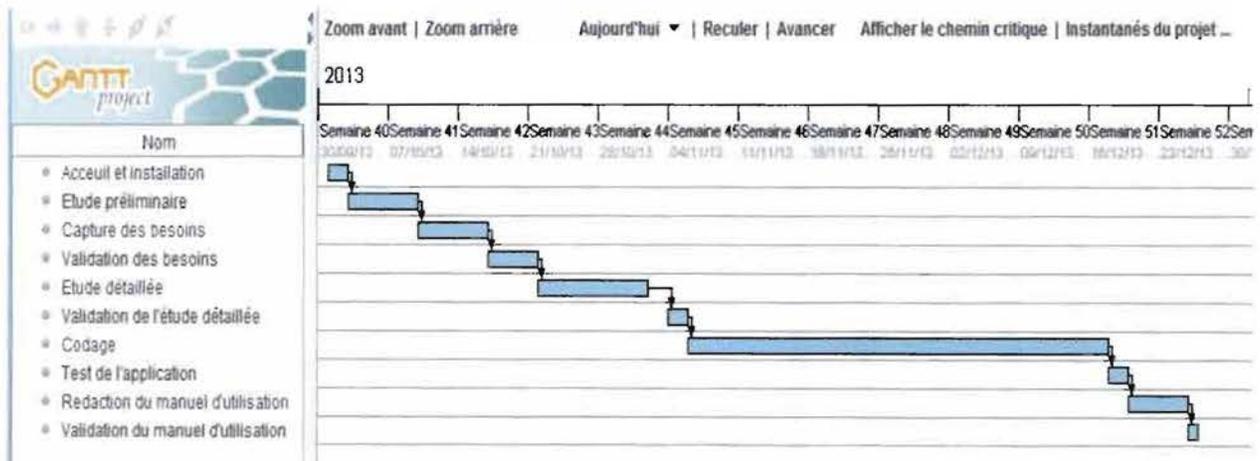


Figure 1.4 : Diagramme de GANTT prévisionnel

III.2.3. Diagramme de GANTT réel

La figure 1.5 représente le diagramme de GANTT du planning réalisé.

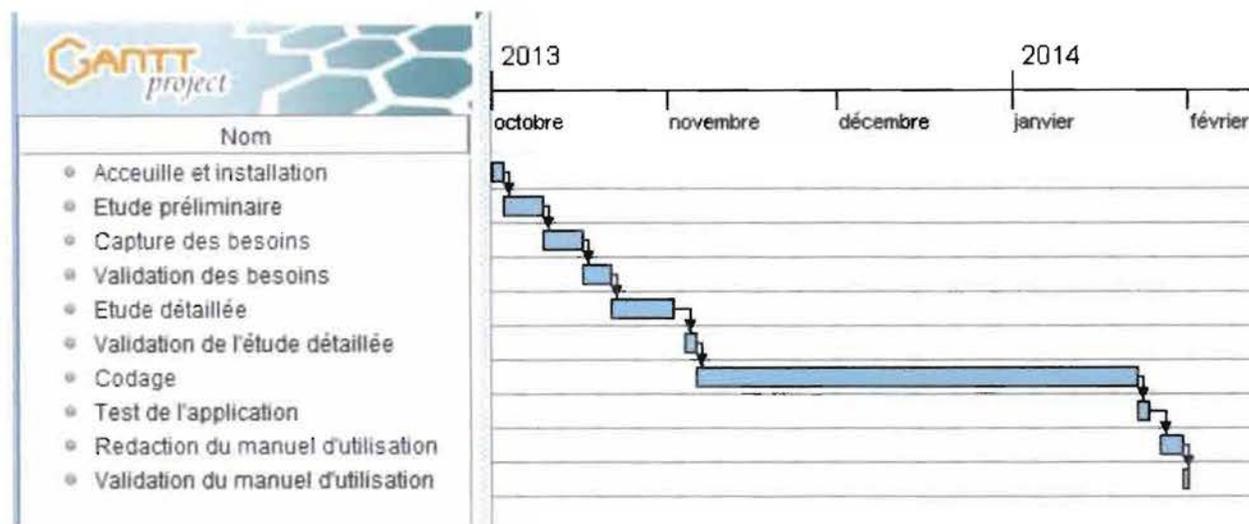


Figure 1.5 : Diagramme de GANTT réel

III.2.4. Explication des écarts

Une des raisons majeures de ces écarts constatés entre le diagramme prévu et celui réalisé est la difficulté de prise en main des outils nécessaires à l'utilisation de la plate-forme JEE [1] [2] [7]. En effet, la compréhension du fonctionnement de PrimeFaces et de Hibernate a pris plus de temps que prévu. De plus, il y a la complexité de communication avec les utilisateurs.

IV Acteurs du projet

Les acteurs d'un projet sont toutes les personnes qui interviennent dans sa gestion. Ils sont divisés en trois (03) groupes qui sont : le groupe de pilotage, le groupe de projet et le groupe des utilisateurs.

IV.1. Groupe de pilotage

Le groupe de pilotage est un groupe d'encadreurs chargé d'arbitrer et de contrôler les décisions à prendre. Il valide les grands choix techniques, fixe les orientations générales et les délais à respecter. Il définit également les moyens à mettre en place pour la réalisation du projet et approuve le plan d'action établi par le groupe de projet.

Il est constitué de :

- **M. N. Félix KAFANDO, DAF du Centre MURAZ ;**
- **Madame Kadiatou BAYALA, responsable logistique et du patrimoine du Centre MURAZ**
- **Dr Joëlle OUATTARA, enseignant chercheur à l'ESI.**

IV.2. Groupe de projet

Le groupe de projet est chargé de l'exécution du projet, c'est-à-dire l'analyse, la conception, la réalisation et le déploiement de l'application. Il établit également les rapports sur l'activité et l'avancement du projet auprès du groupe de pilotage.

Ce groupe est composé de :

- **KAFANDO Tégawendé. Beranger, élève-ingénieur en Analyse et Programmation 3^e année à l'ESI ;**
- **SOME Kpèssiou Thierry, élève-ingénieur en Analyse et Programmation 3^e année à l'ESI.**

IV.3. Groupe des utilisateurs

Il est constitué des utilisateurs potentiels du système qui sera développé. Il joue donc un rôle important dans la capture des besoins du système, et dans la validation des fonctionnalités développées.

Il est composé de :

- **M. N. Félix KAFANDO**, DAF du Centre MURAZ ;
- **Madame Kadiatou BAYALA**, responsable logistique et du patrimoine du Centre MURAZ ;
- **M. OUATTARA Evariste**
- **les responsables de département**
- **les autres agents administratifs du Centre**

Conclusion

Conscient de l'intérêt que peut lui apporter l'automatisation des tâches, le Centre MURAZ nous a soumis pour étude l'automatisation de la gestion des stocks de fournitures de bureau.

Les acteurs de ce projet seront le personnel du Centre MURAZ, nous-mêmes développeurs et notre superviseur. Pour la modélisation du système, nous avons recours au processus de développement 2TUP.

Après la présentation du cadre général du thème dans ce chapitre, nous aborderons dans le suivant l'analyse du système.

CHAPITRE II : ANALYSE

Introduction

La phase d'analyse permet d'établir une liste exhaustive et précise des fonctionnalités attendues en se focalisant sur le métier des utilisateurs. La redondance des informations et la lenteur dans le traitement de celles-ci étant quelques faiblesses du système existant, l'analyse permettra également de définir clairement les objectifs à atteindre (en termes de performance, de maintenance, etc.), afin de pallier ces limites.

Ainsi, dans ce chapitre nous présenterons d'abord l'analyse des besoins tant fonctionnels que techniques, puis nous décrirons les solutions proposées pour atteindre les objectifs fixés.

I. Analyse des besoins

I.1. Spécifications fonctionnelles

I.1.1. Interviews des utilisateurs

Les principales techniques utilisées pour la capture des besoins fonctionnels sont : la lecture de documents décrivant le métier des utilisateurs et les interviews. Pour notre projet, nous utilisons la technique des interviews car elle présente l'avantage de favoriser la communication entre les développeurs et les futurs utilisateurs du système.

Interview du DAF du Centre MURAZ

<p><u>Personne :</u> M. N. Félix KAFANDO</p>	<p><u>Thème :</u> Gestion des stocks de fournitures de bureau du Centre MURAZ</p>	<p>Date d'interview : 04/10/2013 <u>Heure :</u> 16h15 Durée : 30 minutes</p>
<p>Nous souhaiterions avoir une application nous permettant le suivi de nos stocks de fournitures de bureau. A tout moment nous devrions être capables de connaître l'état de nos stocks (quantité et valeur du stock). Chaque début d'année, nous faisons un grand approvisionnement de nos stocks après avoir recensé les besoins des différents départements. Une commission de réception, dont le responsable logistique, se charge de réceptionner les produits, et les enregistrer. Une fois l'enregistrement effectué la DAF et le responsable logistique se chargent de ravitailler les différents départements selon leur besoin. Dès que le besoin se fait sentir au niveau d'un département, l'employé en informe son département qui se charge d'en informer la DAF. Sur accord du DAF le responsable logistique ravitaille le département en enregistrant l'employé ainsi que son département et son unité. Lorsque le stock de sécurité d'un article est atteint, il nous faut lancer une nouvelle commande pour cet article.</p> <p>N.B. le montant global d'une commande est enregistré ainsi que le montant des produits à l'entrée</p>		

Interview du responsable logistique et du patrimoine

<p><u>Personne:</u> Mme Kadiatou BAYALA</p>	<p><u>Thème :</u> Gestion des stocks de fournitures de bureau du Centre MURAZ</p>	<p><u>Date d'interview:</u>9/10/2013 <u>Heure:</u>09h25 <u>Durée:</u>25 minutes</p>
<p>Notre travail dans la gestion de stock de fournitures de bureau consiste à réceptionner les articles lors des approvisionnements puis les dispatcher en même temps au niveau des différentes directions selon ce qu'elles ont demandé. Il n'y a pas de stock de masse à notre niveau (la DAF) pour tout le Centre. Nous n'avons que le stock de la DAF qui constitue en même temps le stock de sécurité du Centre MURAZ. En ce qui concerne le grand approvisionnement, chaque direction remplit une fiche d'expression des besoins que nous transférons au responsable des marchés publics. A la réception des articles il y a des documents comme le bordereau de livraison (BL) qui ne nous sont malheureusement pas fournis. Lorsqu'un agent est dans le besoin et qu'il doit se ravitailler à notre niveau, ce dernier doit remplir une fiche d'expression de besoin que son responsable signe et nous le transmet pour que nous puissions le satisfaire.</p> <p>Nous aimerions pouvoir assurer le suivi total de nos stocks de fournitures de bureau(entrée, sortie, quantité restante, valeur du stock ...); le suivi doit être fait au niveau de chaque direction également et qu'à tout moment nous puissions connaître l'état du stock global et à chaque niveau; aussi nous comptons revoir l'entreposage.</p>		

NB : Nous avons également interviewé le DRH et le magasinier mais vu que les informations recueillies sont incluses dans les interviews ci-dessus, nous n'avons pas jugé nécessaire de les mentionner.

I.1.2. Recueil des fonctionnalités du futur système

Avec le groupe de pilotage, les différentes fonctionnalités qui ont été retenues pour le système à développer sont:

- La gestion des utilisateurs : ce module permettra dans un premier temps la création d'utilisateurs ainsi que la définition de leurs profils. Dans un second temps, il permettra la gestion des mots de passe : leur longueur, leur durée de vie
- Le suivi de tous les stocks assuré par le RLP et contrôlé par le DAF
- Le suivi du stock local au niveau de chaque département assuré par les responsables des départements
- Le contrôle des expressions de besoins assuré par le RLP et validée par le DAF

I.1.3. Identification des acteurs du système

Avant de dresser la liste des acteurs du système, il convient de définir clairement ce que c'est qu'un acteur. Un acteur est un humain ou une machine ne faisant pas partie de la solution à réaliser mais qui participe au fonctionnement général de la solution par une interaction.

Les acteurs susceptibles d'interagir avec le système sont:

- Le DAF : il consulte et valide les expressions de besoins. Il peut consulter les articles et voir l'état des stocks ;
- Le RLP: il gère les articles et les expressions de besoins et peut voir l'état des stocks. C'est lui également qui s'occupe de la répartition des articles à la réception ;
- Le RD : il effectue les sorties d'articles vers les personnes bénéficiaires. Il a la possibilité de voir l'état du stock de son département et peut consulter les articles.
- Agent du Centre :il a la possibilité d'émettre des expressions de besoins et de consulter la liste des articles.

I.1.4. Identification des cas d'utilisation

L'identification des acteurs en interaction avec le système et des différentes fonctionnalités attendues permettent de déterminer les cas d'utilisation présentés dans le tableau 2.1.

Tableau 2.1: Liste des cas d'utilisation

N°	Cas d'utilisation
CU01	S'authentifier
CU02	Consulter les articles
CU03	Gérer les articles
CU04	Exprimer un besoin
CU05	Consulter les expressions de besoins
CU06	Gérer les expressions de besoins
CU07	Effectuer une sortie
CU08	Afficher l'état du stock du département
CU09	Afficher l'état des stocks
CU10	Gérer les comptes utilisateurs
CU11	Gérer les catégories
CU12	Changer mot de passe

I.2. Spécifications techniques

I.2.1. Plateforme de développement

La plateforme de développement utilisée pour la réalisation de *SGSF-MURAZ* est JEE (Java Enterprise Edition dans sa version 7) [1][2] [7] qui est l'extension serveur de la plateforme JSE (Java Standard Edition) de SUN. C'est une plateforme de développement qui permet de développer des applications Web composées de Servlet et JSP. JEE est également une

spécification destinée aux éditeurs de logiciels qui désirent créer des serveurs d'applications compatibles JEE.

Le Framework couplé à JEE est JSF 2.2 [4] [7] dont le but est d'accroître la productivité des développeurs dans le développement d'interfaces utilisateur tout en facilitant leur maintenance. JSF est un Framework d'interfaces utilisateur pour les applications web basées sur les technologies JSP et Servlets. Il implémente MVC 2, une architecture MVC pour les applications Web dans lesquelles les requêtes HTTP sont transmises par le client à un "Contrôleur" servlet unique qui met à jour le «Modèle», puis invoque le cas échéant la « Vue » de rendu.

Pour profiter de toute la puissance de cette plateforme, nous utilisons la librairie de composants graphiques PrimeFaces dans sa version 4.0 [5] [6] [9].

I.2.2. SGBD

La gestion des données du futur système est assurée par le système de gestion de base de données (SGBD) PostgreSQL 9.3[11], largement reconnu pour son comportement stable et pour ses possibilités de programmation étendues, directement dans le moteur de la base de données, via PL/pgSQL.

I.2.3. Serveur d'applications

Le serveur d'application sur lequel sera déployé l'application, une fois achevée, est GlassFish Server Open Source dans sa version 4.0 [2] [10]. C'est un serveur d'applications facile à utiliser, rapide et faisant parti des leaders du marché. Il permet également d'accroître les performances tout en offrant des fonctions de clustering [10] et de disponibilité élevée aux services évolutifs qui sont capables de fonctionner malgré une défaillance matérielle ou logicielle. En Java EE, on parle de clustering de serveurs d'applications pour parler de la mise en relation d'un certain nombre de serveurs entre eux [10]. Une application Java serveur (EJB, application web etc.) sera déployée sur l'ensemble des serveurs en parallèle.

I.2.4. Solution d'ORM

Afin de rendre le futur système portable d'un point de vue SGBD et de permettre l'abstraction de toute sa partie SQL, nous utilisons le Framework de mapping objet-relationnel (en anglais object relational mapping ou ORM) Hibernate3.2.5 [3] [8]. En effet, ce Framework facilite la persistance et la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant plus que ce code est généralement fastidieux et redondant.

1.2.5.Méthode de calcul du coût de développement

La méthode de calcul de coût utilisée pour évaluer l'effort à fournir pour la réalisation de notre projet est COCOMO (acronyme de l'anglais *CO*nstructive *CO*st *MO*del). Elle propose trois (03) formules de calcul en fonction de la complexité de l'application à réaliser : *S* (en anglais *organic*), *P* (en anglais *semi detached*) et *E* (en anglais *embedded*).

Le tableau 2.2 présente les formules proposées par COCOMO en fonction de la complexité de l'application à réaliser.

Tableau 2.2:Formules de calcul proposées par COCOMO

COMPLEXITE	EFFORT (EN HM:HOMME MOIS)	COUT DE DEVELOPPEMENT (EN F CFA)
S	<i>Effort</i> HM= 2, 4 * $KILS^{1,05}$	Coût de développement = HM * Salaire
P	<i>Effort</i> HM= 3 * $KILS^{1,12}$	
E	<i>Effort</i> HM= 3, 6 * $KILS^{1,2}$	

Compte tenu de sa complexité, l'application à réaliser est de type semi-détaché (P). Nous utiliserons donc les méthodes de calcul de ce type pour calculer le coût financier de l'application.

II. Étude des scenarii

II.1. Architecture réseau

II.1.1. Formalisme

Pour la représentation de l'architecture réseau des différents scenarii, nous utiliserons les symboles listés dans la figure 2.1.

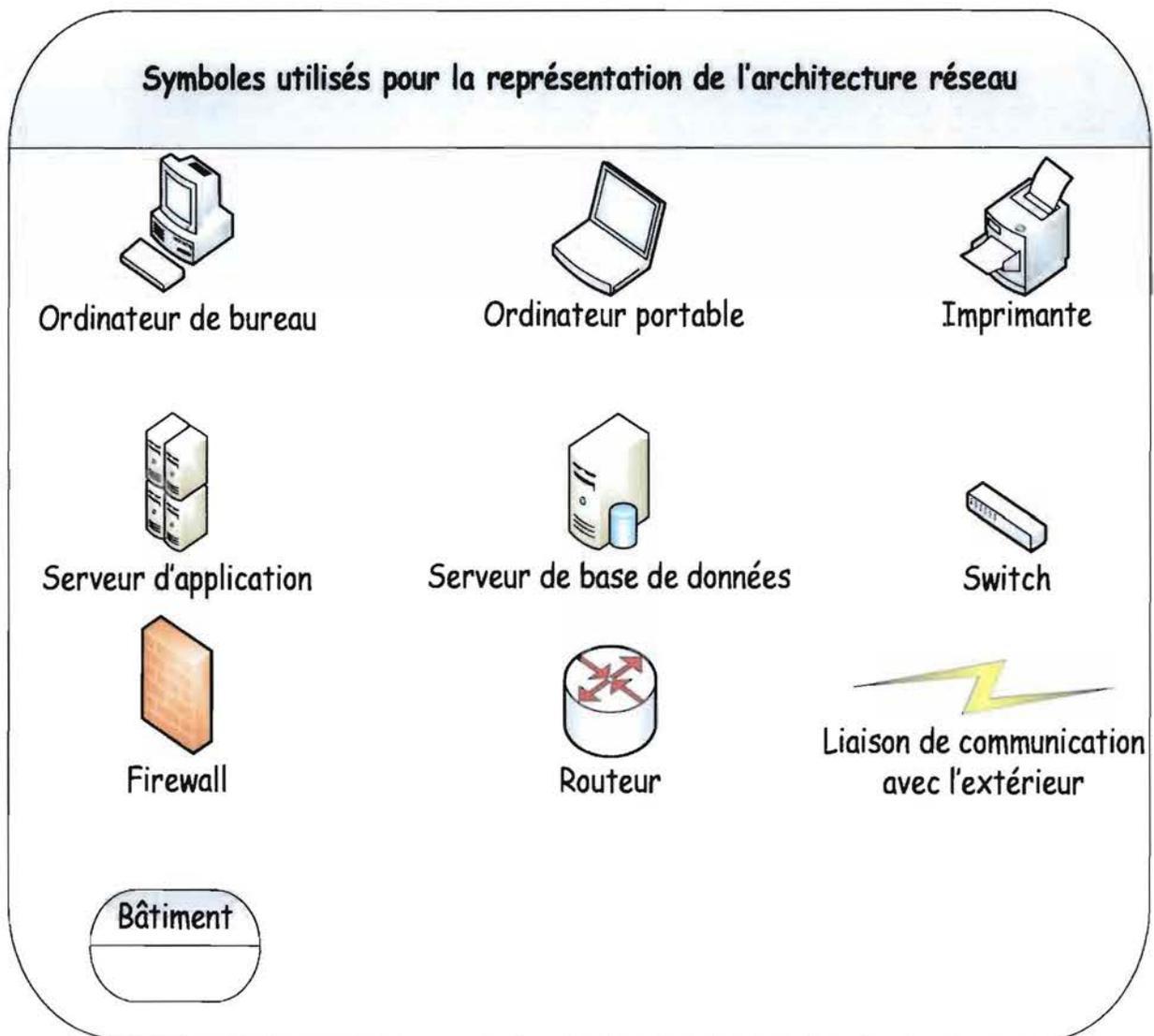


Figure 2.1 : Symboles utilisés pour la représentation de l'architecture réseau

II.1.2. Premier scénario

Ce premier scénario dont l'architecture est présentée dans la figure 2.2 permet de satisfaire les contraintes de fonctionnement du système. Tous les acteurs doivent se rendre dans l'enceinte de l'entreprise, pour accéder à l'application à partir d'un client léger (navigateur web).

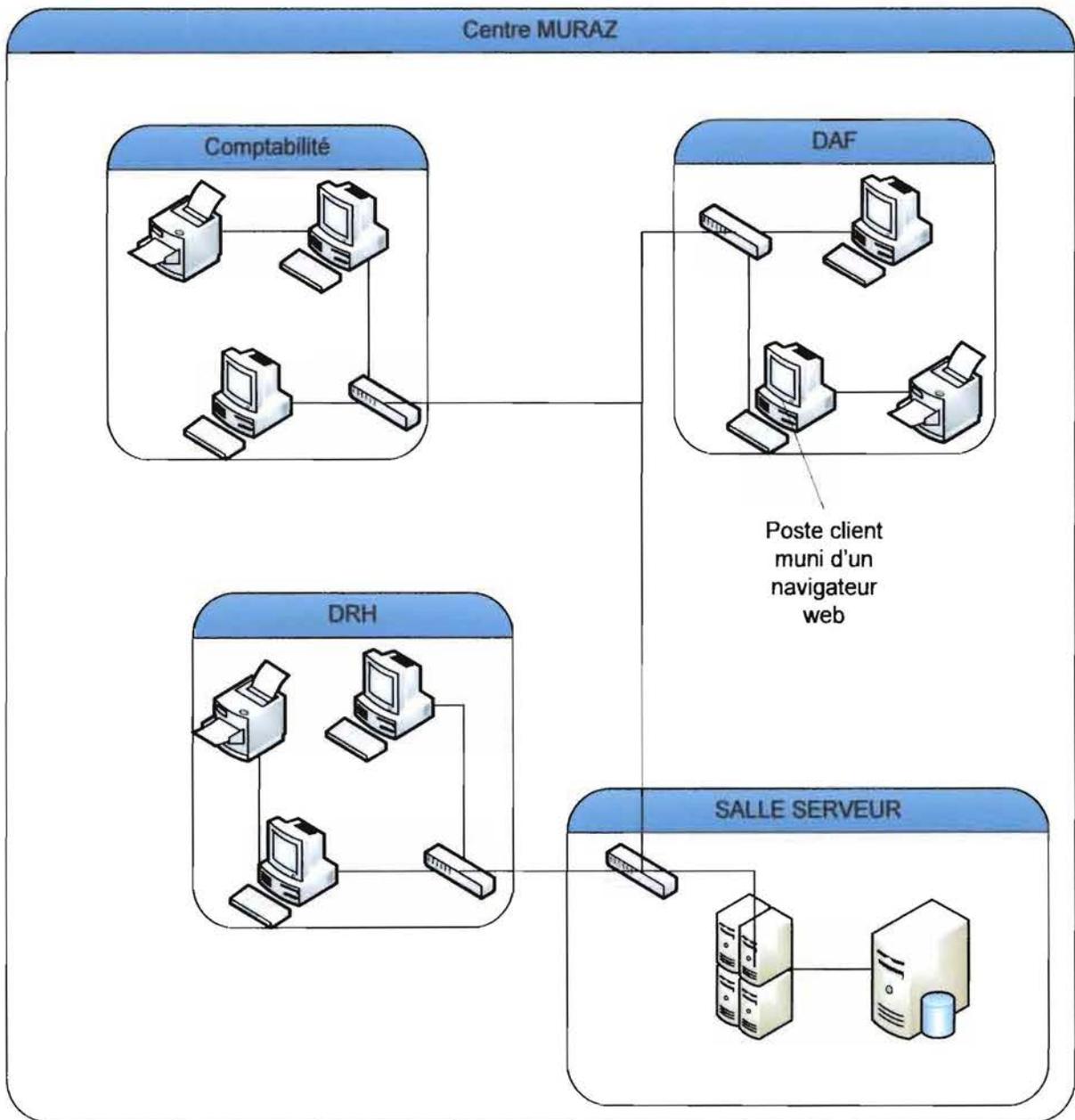


Figure 2.2 : Schéma architectural réseau du premier scénario

II.1.3. Deuxième scénario

Le deuxième scénario ressemble au précédent mais diffère de lui par son ouverture à Internet. Cela permettra aux utilisateurs finaux de se connecter au système et d'accomplir pleinement leurs tâches sans contraintes de localisation. L'architecture de ce scénario est représentée par la figure 2.3.

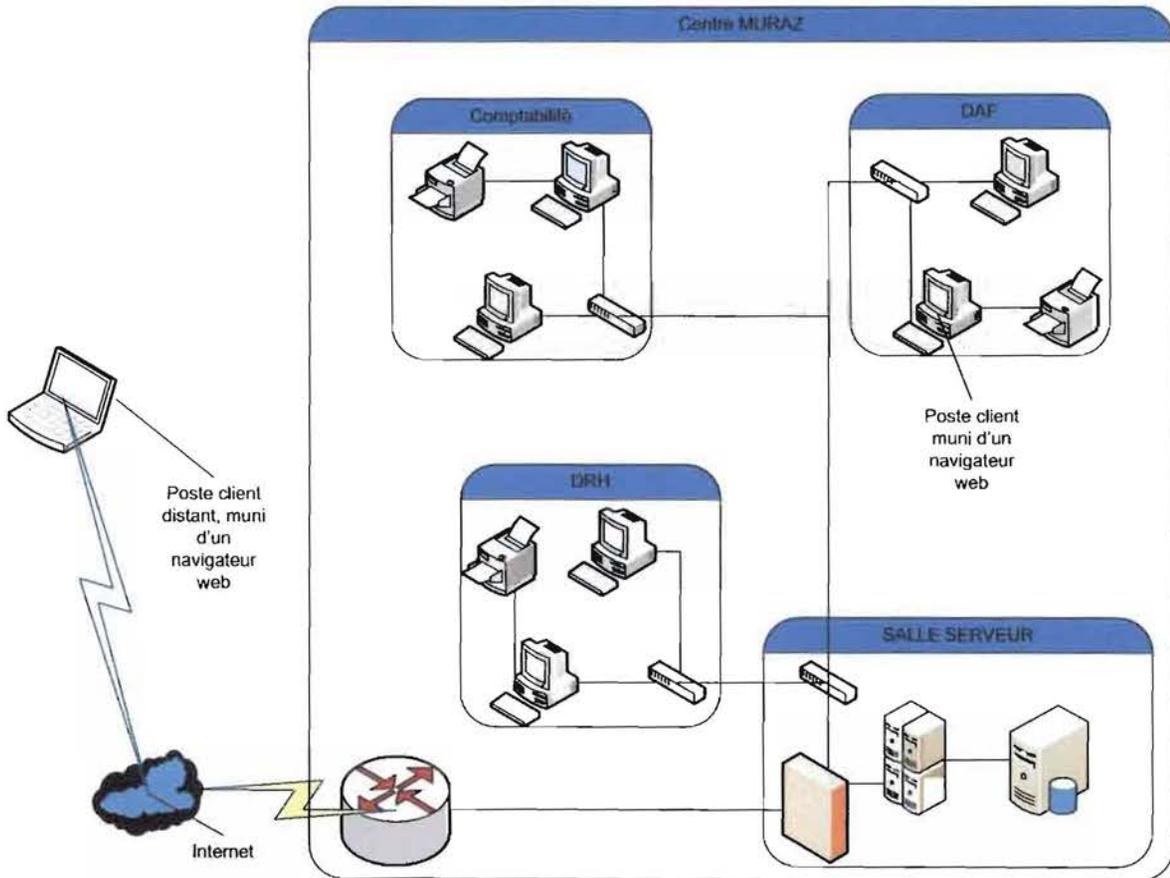


Figure 2.3 : Schéma architectural du deuxième scénario

II.2. Estimation des coûts de réalisation

Les différents coûts de l'application sont regroupés dans le tableau 2.3. Dans ce tableau, le terme « N/A » dans la colonne « Quantité » est mis pour les éléments qui ne sont pas quantifiables. Les éléments en gras concernent uniquement le deuxième scénario. De ce fait le coût de ces éléments sera ôté du premier scénario.

Tableau 2.3 : Cout de réalisation de l'application

Catégorie	Désignation	Quantité	Prix unitaire (en F CFA)	Prix total (en F CFA)
Matériels et logiciels	Ordinateurs postes fixes	85	Déjà existant	0
	Imprimante Laser	16	Déjà existant	0
	Connexion internet	N/A	Déjà existant	0
	Antivirus Kaspersky Antivirus	N/A	Déjà existant	0
	Switch	13	Déjà existant	0
	Serveur (BD, application)	01	Déjà existant	0
	Firewall (scénario 2 uniquement)	01	250 000	250 000
	Routeur(scénario 2 uniquement)	01	250 000	250 000
	GlassFish Server Open Source 4.0	N/A	Gratuit	0
	NetBeans 7.3.1	N/A	Gratuit	0
	PostgreSQL 9.3	N/A	Gratuit	0
	JSF 2.2	N/A	Gratuit	0

	Hibernate 3.2.5	N/A	Gratuit	0
	Coût total des matériels et logiciels			500 000
Coût de développement (CD)	Formule P de COCOMO avec les variantes suivantes : Nombre de lignes de code = 4780, Salaire (S) = 200000FCFA			3 460 274
	$CD = HM * S$ et $HM = 3 * KILS^{1, 12}$			
Formation des utilisateurs	Formation des utilisateurs	85	5 500	467 500
	Coût total de formation des utilisateurs			467 500
Coût de réalisation de l'application avec le premier scénario				3 927 774
Coût de réalisation de l'application avec le deuxième scénario				4 427 774
Estimation du coût du câblage réseau interne				54 287 000

Source service informatique Centre MURAZ/ SDI 2012 pour le coût matériel

NB : Le réseau interne est indispensable au déploiement de notre application. Cependant nous ne prenons pas en compte la mise en réseau dans notre évaluation car ce projet figure en bonne place dans le SDI2012 du Centre MURAZ.

II.3. Comparaison des scénarii

Le premier scénario présente l'avantage d'avoir une mise en œuvre facile. En outre, puisqu'elle est à l'abri d'Internet, l'application est protégée contre les intrusions : elle a donc un niveau de sécurité élevé. Son seul inconvénient est qu'elle a une accessibilité réduite : elle ne permet pas la mobilité des utilisateurs.

Comme pour le premier scénario, la mise en œuvre de l'application ici est facile pour le deuxième. De plus, l'application a une accessibilité accrue : elle permet la mobilité des utilisateurs. Cependant, du fait de son ouverture à Internet, les risques d'intrusion et d'attaques sont élevés : elle a donc une sécurité relativement faible.

II.4. Scénario retenu

Au regard des contraintes et des besoins du système à développer, le scénario qui a été retenu, en accord avec le groupe de pilotage est le premier scénario. En effet, l'accessibilité en tout temps et lieu n'est pas un souci majeur pour les futurs utilisateurs du système. Les deux critères décisifs pour le choix de ce scénario furent les suivants:

- ✓ Le haut niveau de la sécurité dû au fait que l'application n'est pas ouvert à Internet,
- ✓ le coût de réalisation qui est relativement faible par rapport au deuxième scénario.

Conclusion

Les deux (02) parties de ce chapitre ont permis de cerner le cadre de réalisation de ce projet. En effet les interviews des utilisateurs du futur système et la spécification des fonctionnalités attendues ont permis de comprendre les attentes vis-à-vis de ce projet.

La méthode d'analyse et de conception utilisée est 2TUP. Aussi, la plateforme de développement retenue est JEE 7 avec le Framework JSF 2.2 et la librairie de composants PrimeFaces4.0, le serveur d'application qui hébergera *SGSF-MURAZ* est GlassFish Server Open Source 4.0. Enfin, le SGBD utilisé est PostgreSQL 9.3 et la solution d'ORM retenue est Hibernate 3.2.5.

Des deux (02) scénarii présentés, le choix s'est porté sur le premier scénario pour deux raisons :

- L'absence d'ouverture à Internet qui n'est pas handicapant dans la mesure où l'application est uniquement utilisée en interne.
- le coût de réalisation relativement faible (3 735 274 F CFA).

CHAPITRE III : CONCEPTION

Introduction

La phase de conception décrira de manière précise et détaillée, le fonctionnement du système futur, afin d'en faciliter la réalisation. Il sera donc question de présenter le diagramme de cas d'utilisation, la description de quelques cas et le diagramme des classes. La description complète des diagrammes de conception est effectuée dans le document technique.

I. Diagramme de cas d'utilisation / Diagramme de séquences.

I.1. Diagramme de cas d'utilisation

I.1.1. Représentation du diagramme de cas d'utilisation

La figure 3.1 présente le diagramme de cas d'utilisation définissant les exigences fonctionnelles attendues, les acteurs (utilisateurs du système) ainsi que les relations qui unissent acteurs et fonctionnalités.

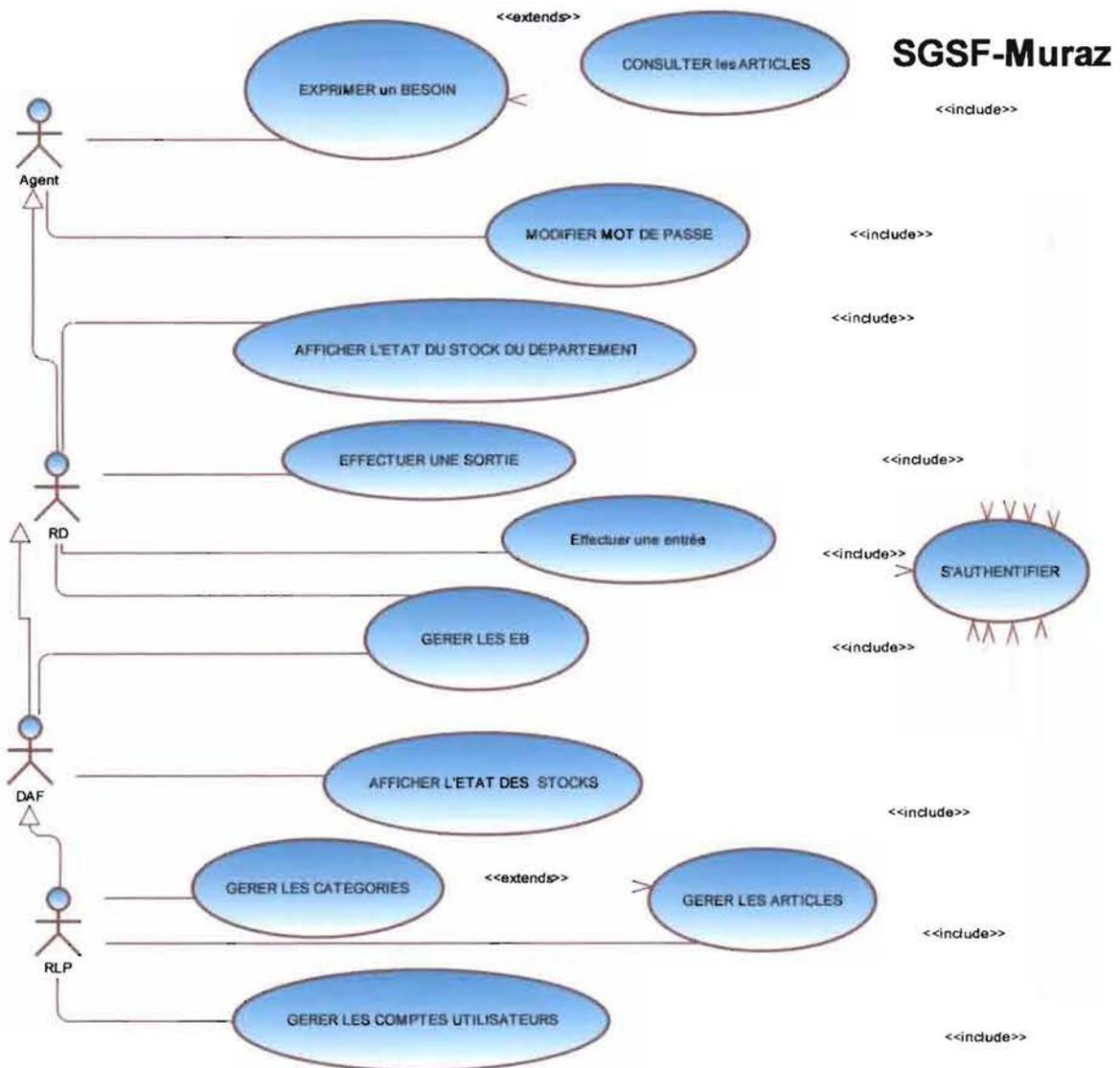


Figure 3.1 : Diagramme de cas d'utilisation de SGSF-MURAZ

1.1.2. Documentation de certains cas d'utilisation

Description textuelle du cas d'utilisation « S'authentifier »

Le tableau 3.1 présente l'enchaînement des opérations pour s'authentifier.

Tableau 3.1:Description textuelle du cas d'utilisation «S'authentifier »

CU01 :“ S'authentifier”	
Objectif : Ce cas d'utilisation permet à un utilisateur de se connecter au système.	Acteurs : DAF, RLP, RD, Agent du Centre Version: 1.0. Date de création: 25/10/2013.
Pré conditions : L'utilisateur n'est pas encore connecté.	
Scénario nominal: <ol style="list-style-type: none"> 1. L'utilisateur demande à accéder au système. 2. Le système demande l'identifiant et le mot de passe. 3. L'utilisateur saisit son identifiant et son mot de passe. 4. Le système vérifie les informations saisies. 5. Le système présente l'interface d'accueil selon les droits de l'utilisateur. 	
Enchaînements d'exception: <p>E1:l'identifiant et/ou le mot de passe sont incorrects : l'enchaînement commence au point 4 du scénario nominal.</p> <p>5. Le système notifie à l'utilisateur que l'identifiant et/ou le mot de passe sont incorrects. L'enchaînement continue au point 2 du scénario nominal.</p> <p>E2:le nombre de tentatives de connexion vaut cinq : l'enchaînement commence au point 4 du scénario nominal.</p> <p>5. Le système notifie à l'utilisateur de réessayer plus tard.</p>	
Post conditions : Un nouvel utilisateur est authentifié sur le système.	

Description textuelle du cas d'utilisation <<Gérer les Articles>>

L'enchaînement des opérations permettant de réaliser le cas d'utilisation « Gérer les Articles » est présenté dans le tableau 3.2.

Tableau 3.2: Description textuelle du cas d'utilisation « Gérer les Articles »

CU03 :“ Gérer les Articles”	
Objectif : Ce cas d'utilisation permet de créer de mettre à jour un article.	Acteurs : RLP. Version: 1.0. Date de création: 25/10/2013.
Pré conditions : L'utilisateur est connecté et accède au menu de gestion des articles.	
Scénario nominal: <ol style="list-style-type: none"> 1. L'utilisateur demande à créer ou à mettre à jour un article. 2. Si la demande est une création, le système lui présente un formulaire vide de création. Sinon l'utilisateur choisit l'article à mettre à jour puis le système lui présente le formulaire de mise à jour muni des informations à modifier. 3. L'utilisateur saisit les informations exigées. 4. Le système vérifie les informations saisies. 5. Le système valide le formulaire et crée ou met à jour l'article. 6. Le système informe l'utilisateur du résultat de sa requête. 	
Enchaînements d'exception: <p>E1: Les informations saisies sont erronées/incomplètes : l'enchaînement commence au point 4 du scénario nominal.</p> <ol style="list-style-type: none"> 5. Le système notifie à l'utilisateur que les informations sont erronées ou incomplètes 6. L'utilisateur corrige/complète les informations. <p>L'enchaînement continue au point 6 du scénario nominal.</p>	
Post conditions : le système contient un nouvel article ou un article est mis à jour.	

Description textuelle du cas d'utilisation « Gérer les EB »

Le tableau 3.3 présente l'enchaînement des opérations pour suivre les Expressions des Besoins.

Tableau 3.3:Description textuelle du cas d'utilisation « Gérer les EB »

CU06 :“ Gérer les EB ”	
Objectif : Ce cas d'utilisation permet de créer ou de mettre à jour une EB.	Acteurs : RD, RLP. Version: 1.0. Date de création: 25/10/2013.
Pré conditions : L'utilisateur est connecté et accède au menu de gestion des EB.	
Scénario nominal: <ol style="list-style-type: none"> 1. L'utilisateur demande à créer ou à mettre à jour une EB. 2. Si la demande est une création, le système lui présente un formulaire vide de création. Sinon l'utilisateur choisit l'EB à mettre à jour puis le système lui présente le formulaire de mise à jour muni des informations à modifier. 3. L'utilisateur saisit les informations exigées. 4. Le système vérifie les informations saisies. 5. Le système valide le formulaire et crée ou met à jour l'EB. 6. Le système informe l'utilisateur du résultat de sa requête. 	
Enchaînements d'exception: <p>E1:Les informations saisies sont erronées ou incomplètes : l'enchaînement commence au point 4du scénario nominal.</p> <ol style="list-style-type: none"> 5. Le système notifie à l'utilisateur que les informations sont erronées ou incomplètes 6. L'utilisateur corrige ou complète les informations. <p>L'enchaînement continue au point 5du scénario nominal.</p>	
Post conditions : le système contient une nouvelle EB ou une EB est mise à jour.	

I.2. Diagramme de séquences

I.2.1. Cas d'utilisation « S'authentifier »

La figure 3.2 présente la séquence des opérations permettant de réaliser le cas d'utilisation « S'authentifier ».

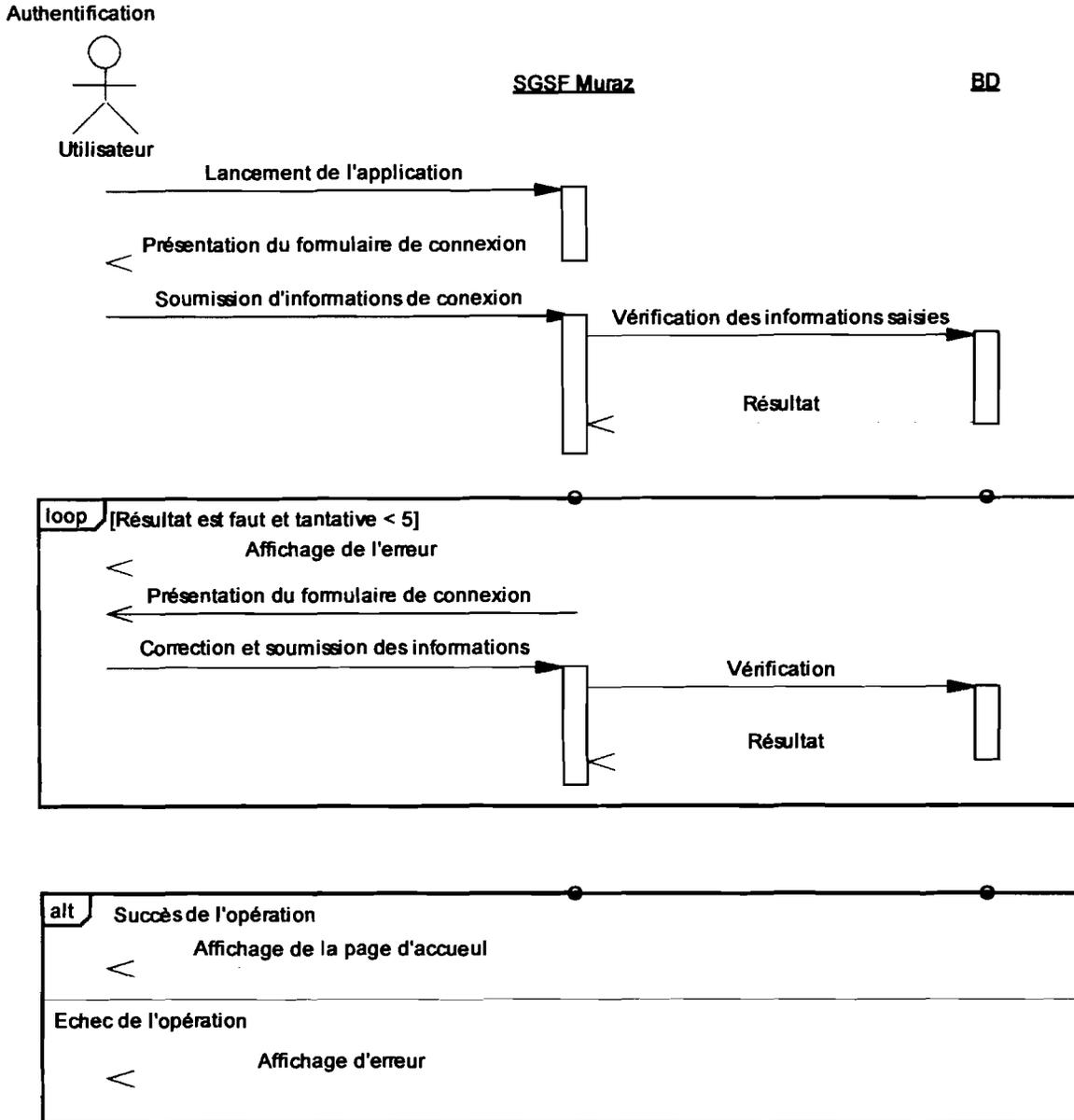


Figure 3.2 : Diagramme de séquence du cas d'utilisation «S'authentifier »

I.2.2. Cas d'utilisation « Gérer les Articles »

La figure 3.3 présente la séquence des opérations permettant de réaliser le cas d'utilisation « Gérer les Articles ».

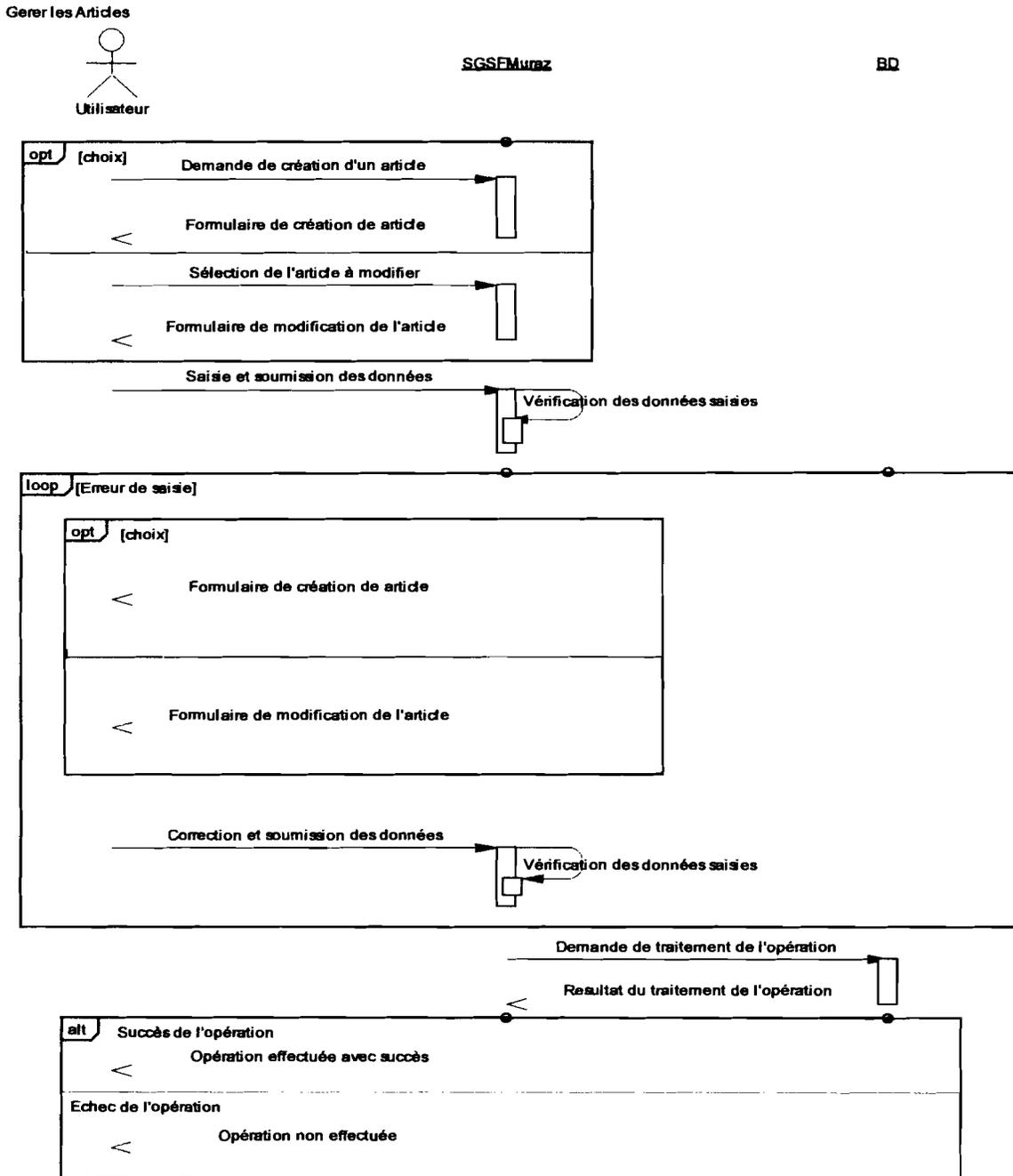


Figure 3.3 : Diagramme de séquence du cas d'utilisation « Gérer les Articles »

1.2.3. Cas d'utilisation « Gérer les EB »

La figure 3.4 présente la séquence des opérations permettant de réaliser le cas d'utilisation « Gérer les EB ».

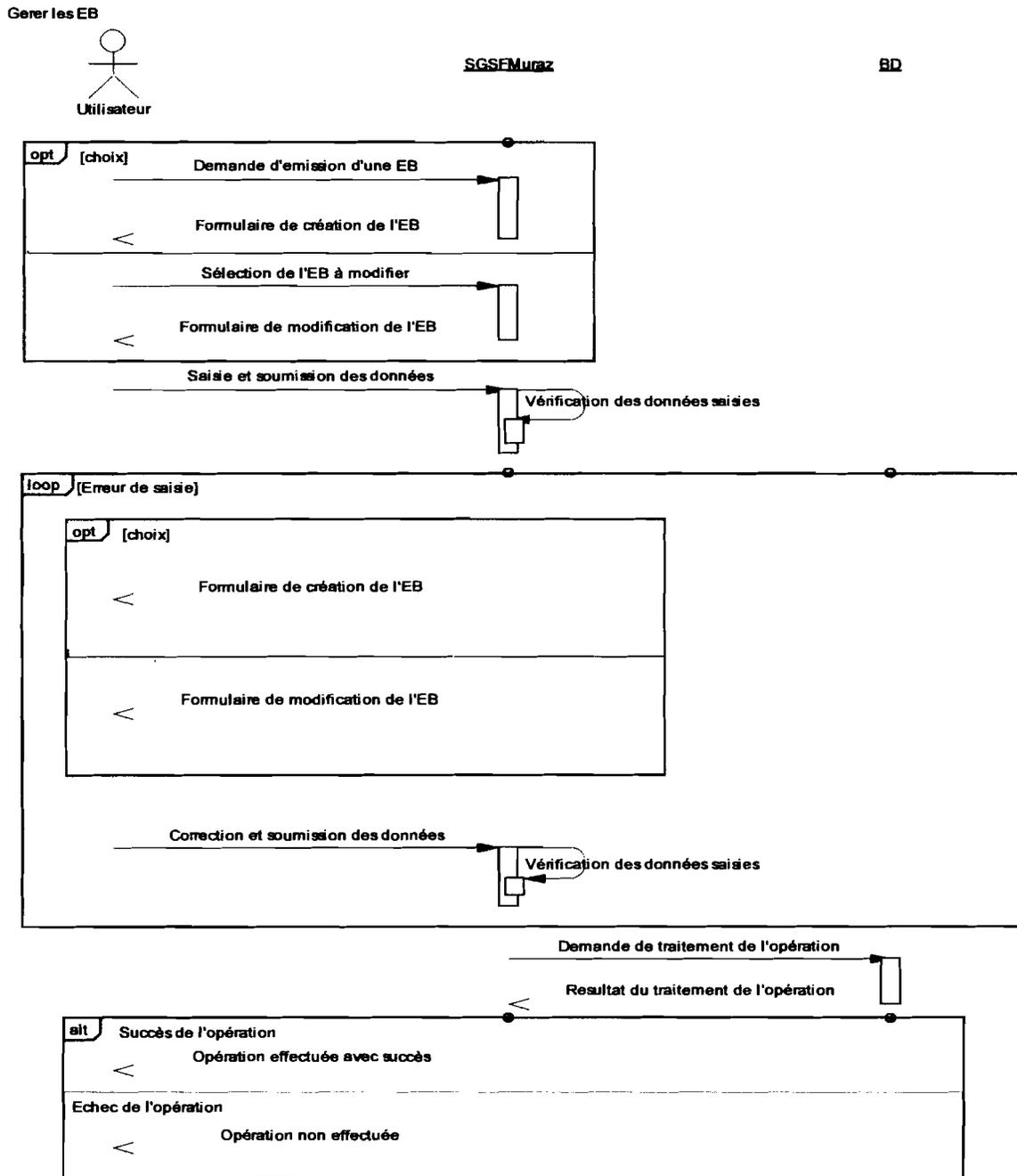


Figure 3.4 : Diagramme de séquence du cas d'utilisation «Gérer les EB»

II. Diagramme de classes

II.1. Représentation du diagramme de classes

La figure 3.5 présente la structure statique du système c'est-à-dire la partie du système toujours présente lors des interactions définies par les cas d'utilisation. Les règles de gestion associées au diagramme sont les suivantes :

Règle 1	Une personne est dans une et une seule unité qui à son tour est dans un et un seul département alors qu'un département contient une ou plusieurs unités où chaque unité contient une ou plusieurs personnes.
Règle 2	Une personne possède plusieurs comptes où chaque compte est associé à un ou plusieurs groupes alors qu'un groupe contient plusieurs comptes appartenant chacun à une et une seule personne.
Règle 3	Une personne peut effectuer une ou plusieurs sorties et une sortie est effectuée par une et une seule personne.
Règle 4	Une personne peut effectuer une ou plusieurs expressions de besoins alors qu'une expression de besoins est effectuée par une et une seule personne.
Règle 5	Une personne à travers son compte peut ouvrir une ou plusieurs sessions où chaque session peut être l'objet d'une ou plusieurs opérations alors qu'une opération est effectuée dans une et une seule session associée à un compte et un seul.
Règle 6	Un article est dans une et une seule catégorie tandis qu'une catégorie contient au moins un article.
Règle 7	Un article peut sortir plusieurs fois et une sortie d'article contient au moins un article.
Règle 8	Un article peut se retrouver dans plusieurs livraisons et chaque livraison contient un article au moins.
Règle 9	Un article est stocké dans au moins un département et un département peut avoir plusieurs articles en stock.

Règle 10	Un article peut se retrouver dans plusieurs expressions de besoins et une expression de besoins contient un article au moins.
Règle 11	Un article peut se retrouver dans plusieurs entrées et chaque entrée contient un article au moins.
Règle 12	Une commande peut être livrée plusieurs fois alors qu'une livraison concerne une commande et une seule.
Règle 13	Une commande peut contenir plusieurs expressions de besoins alors qu'une expression de besoins est liée à une commande au plus.

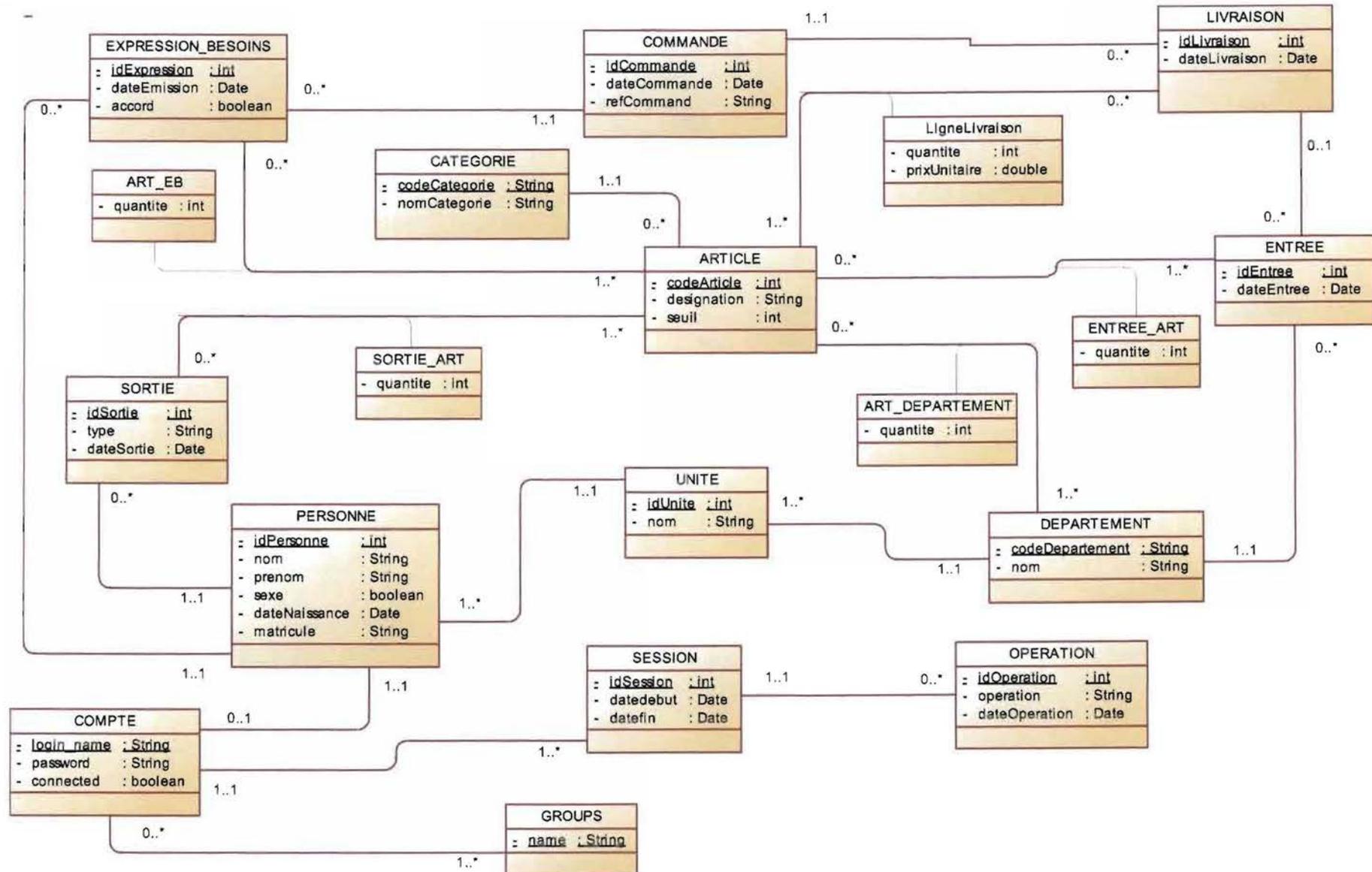


Figure 3.5: Diagramme de classes

II.2. Description de quelques classes

II.2.1. Description de la classe « Article »

Le tableau 3.4 présente l'ensemble des attributs et des méthodes qui forment la classe « Article ».

Tableau 3.4: Description de la classe « Article »

CLASSE : Article		
Attributs		
Code	Type	Description
codeArticle	String	Identifiant de l'Article
Seuil	Int	Quantité maximum déclenchant une alerte

II.2.2. Description de la classe « EXPRESSION_BESOINS »

La description de la classe « EXPRESSION_BESOINS » est résumée dans le tableau 3.5.

Tableau 3.5: Description de la classe « EXPRESSION_BESOINS »

CLASSE:EXPRESSION_BESOINS		
Attributs		
Code	Type	Description
idExpression	Int	Identifiant de l'expression
dateEmission	Date	Date d'émission de l'expression

Conclusion

Dans ce chapitre, nous avons abordé le fonctionnement du système. Le diagramme de cas d'utilisation présente les fonctionnalités principales du système du point de vue extérieur. Le diagramme de séquence est une représentation temporelle des interactions entre les objets. Il complète la description du diagramme de cas d'utilisation. Le diagramme des classes quant à lui est une représentation de la structure interne du logiciel.

Le chapitre suivant présente quelques modules développés ainsi que la politique de sécurité inhérente à *SGSF-MURAZ*.

CHAPITRE IV:REALISATION

Introduction

Après les phases d'analyse et de conception, vient à présent celle de la réalisation.

Dans ce chapitre, nous présenterons d'abord quelques modules du logiciel développé. Ensuite nous exposerons la politique adoptée pour assurer la sécurité du système.

I. Présentation de quelques écrans

Ce chapitre permet d'avoir un aperçu du travail qui a été concrètement réalisé.

I.1. Connexion

La figure 4.1 présente l'interface de connexion à *SGSF-MURAZ*. Un utilisateur voulant accéder à son espace de travail doit impérativement s'authentifier. Pour ce faire, il saisit son identifiant et son mot de passe.



The screenshot shows a web interface for logging into the SGSF-MURAZ system. At the top, there is a blue banner with the text "CONNEXION A SGSF-MURAZ". Below this banner is a white rectangular area containing the login form. The form is titled "Authentification" in blue text. It features two input fields: "Utilisateur : " with the text "Thierry" and "Mot de passe : " with a masked password of seven dots. Below the input fields are two buttons: "annuler" and "connexion".

Figure 4.1 : Écran de connexion

Si les informations fournies par l'utilisateur ne sont pas correctes, l'interface de la figure 4.2 lui sera présentée.



Figure 4.2 : Écran de connexion après un échec d'authentification

Une fois authentifié, l'utilisateur peut accéder à page d'accueil de *SGSF-MURAZ*.

I.2. Accueil

La page d'accueil est la première interface présentée à tout utilisateur authentifié. Elle est constituée essentiellement d'un menu qui effectue la synthèse du métier de l'utilisateur connecté. Ce menu est paramétré en fonction du type d'utilisateur, les administrateurs ayant un contrôle total de l'application. La figure 4.4 illustre l'affichage du menu lorsqu'un administrateur se connecte.



Figure 4.3 : Écran d'accueil

I.3. Enregistrement d'un nouvel article

L'enregistrement d'un nouvel article consiste à remplir les champs correspondants aux propriétés d'un article. L'utilisateur doit impérativement saisir certaines propriétés (*la désignation et son seuil*) mais a la possibilité de sélectionner d'autres (*la catégorie de l'article*). Une fois enregistrées, elles apparaîtront dans la liste des articles à sélectionner. La figure 4.6 donne un aperçu de l'enregistrement d'un nouvel article

Nouvel article

Désignation : * cartable

Seuil : * 20

Catégorie : * classement et archivage + nouvelle catégorie

annuler valider

Figure 4.4 : Écran d'enregistrement d'un nouvel article

I.4. Enregistrement d'une nouvelle expression de besoin

L'interface de la figure 4.7 permet l'enregistrement d'une expression de besoins par les utilisateurs.

Nouvelle expression

les articles sélectionnés		
article	quantite	action
ram A4	3	supprimer
ram A3	2	supprimer

article : *

quantite : *

[annuler](#) [ajouter](#)

[annuler](#) [valider](#)

Figure 4.5 : Écran d'enregistrement d'une nouvelle expression de besoin

I.5. Enregistrement d'une sortie d'article

La figure 4.8 présente l'écran permettant à un responsable de département d'effectuer une sortie d'article.

Nouvelle sortie

les articles sélectionnés		
article	quantite	action
ram A4	2	supprimer

article : *

quantite : *

[annuler](#) [ajouter](#)

Type de sortie : *

Bénéficiaire : *

[annuler](#) [effectuer](#)

Figure 4.6 : Écran d'une sortie d'article**I.6. Liste des articles**

La figure 4.9 présente la liste de tous les articles. Elle donne également à l'utilisateur en question d'effectuer une recherche d'un article selon l'un des critères de recherche tels que le code de l'article, sa désignation, sa catégorie ou son seuil, ou selon tous les critères en saisissant le motif de la recherche dans le champs "Rechercher". Situé en bas et à droite de la figure 4.7

code article	désignation	catégorie	seuil
art-0000	ram A4	papier	30
art-0001	ram A3	papier	20
art-0002	bic shneider	écriture et correction	20
art-0004	gomme	écriture et correction	20
art-0006	ruban adhésif	adhésifs,agrafage et découpe	20
art-0007	ciseaux	adhésifs,agrafage et découpe	5

Figure 4.7 : Écran de la liste des articles enregistrés

II. Politique de sécurité

II.1. Protection contre les virus

Un virus est un programme informatique susceptible de porter préjudice au bon fonctionnement d'un système. Un seul virus sur un ordinateur peut infecter un très grand nombre d'ordinateurs en contact avec lui et causer des dommages importants. *SGSF-MURAZ* gérant des données sensibles, nous ne pouvons laisser la moindre brèche leur permettant d'y accéder. Pour ce faire, nous préconisons:

- L'installation d'anti-virus sur tous les ordinateurs et leur mise à jour régulière;
- L'interdiction de télécharger et/ou installer n'importe quel logiciel sur l'un des ordinateurs.

II.2. Sauvegarde et restauration

De nombreuses études statistiques ont montré que l'une des raisons majeures du dépôt de bilan des entreprises est la perte de données. Il est donc nécessaire d'élaborer une politique de gestion de leur sauvegarde et de leur restauration.

La gestion des sauvegardes et de la restauration des données de notre système est planifiée par des logiciels (Natsop Backup Server 4.21 et Memeo Instant Backup 4.60.0.7923) installés sur le serveur de données.

Conclusion

Ce chapitre a fait étalage du travail concrètement réalisé à travers la présentation de quelques modules de *SGSF-MURAZ*.

CONCLUSION GENERALE

Dans le cadre de notre stage de fin d'étude du Cycle des Ingénieurs de Travaux Informatiques option analyse et programmation, nous avons été accueillis au Centre MURAZ.

Cette structure nous a soumis pour étude et réalisation, la gestion des stocks de fournitures de bureau.

Nous avons utilisé le processus de développement 2TUP pour l'analyse et la conception du futur système. Pour le développement, nos choix se sont portés sur JEE avec comme Framework JSF, sur la librairie de composants graphiques PrimeFaces ainsi que sur le Framework de mapping objet/relationnel Hibernate. Et pour la gestion des données, nous avons retenu le SGBD PostgreSQL.

Ce nouveau système qui est toujours à sa phase de développement permettra au Centre MURAZ d'être plus efficace et surtout d'être plus rapide dans le traitement des informations.

BIBLIOGRAPHIE ET WEBOGRAPHIE

Bibliographie :

- [1] 285022-creez-votre-application-web-avec-java-ee.pdf de Coyote du siteduzero.com
- [2] Java EE 6 et GlassFish 3.pdf de Antonio Goncalves, PEARSON
- [3] Java Persistence et Hibernate.pdf de Anthony Patricio, EYROLLES
- [4] Java Server Faces (JSF) avec Eclipse.pdf de François-Xavier Sennesal, ENI
- [5] primefaces_cookbook.pdf de Mert Çalışkan et Oleg Varaksin
- [6] primefaces_users_guide_4_0_edn2.pdf de Çağatay Çivici

Webographie :

- [7] <http://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm>
- [8] <http://docs.jboss.org/hibernate/orm/3.2.5/manual/en-US/html/>
- [9] <http://www.primefaces.org/>
- [10] <http://humbert-florent.developpez.com/javaee/glassfish/clustering/>
- [11] <http://docs.postgresqlfr.org/>

ANNEXES

Annexe 1 : Présentation de JSF

JSF est un Framework de présentation Web, qui permet la création de vues (pages Web ou parties de pages Web) par assemblage de composants. C'est une norme faisant partie de la plateforme Java EE.

Il existe deux implémentations de la norme JSF :

- l'implémentation de référence de Sun ;
- le projet Apache MyFaces ;

1. Vue JSF = hiérarchie de composants

Une vue JSF est structurée sous la forme d'une hiérarchie de composants JSF.

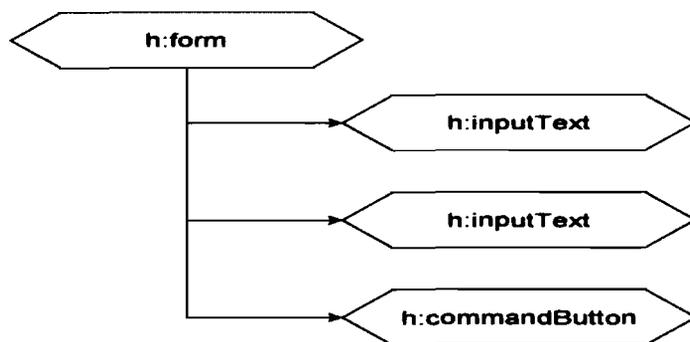


Figure A1.1 : Exemple de hiérarchie de composants JSF

```
<h : form>
<h : inputTextvalue="#{userHome.name}" />
<h : inputTextvalue="#{userHome.firstname}" />
<h : commandButtonvalue="Save" action="#{userHome.save}" />
</h : form>
```

Plusieurs technologies peuvent être utilisées pour la création de vues JSF :

- JSP, dont la nature pas vraiment hiérarchique peut poser problème ;

- Facelets, qui est la technologie préconisée sur la plateforme Seam, et qui utilise des fichiers XHTML, strictement hiérarchiques, pour l'écriture des vues
-

2. Cycle de vie des pages JSF

Le traitement d'une requête pour l'affichage d'une page Web par JSF va être composé des six étapes suivantes :

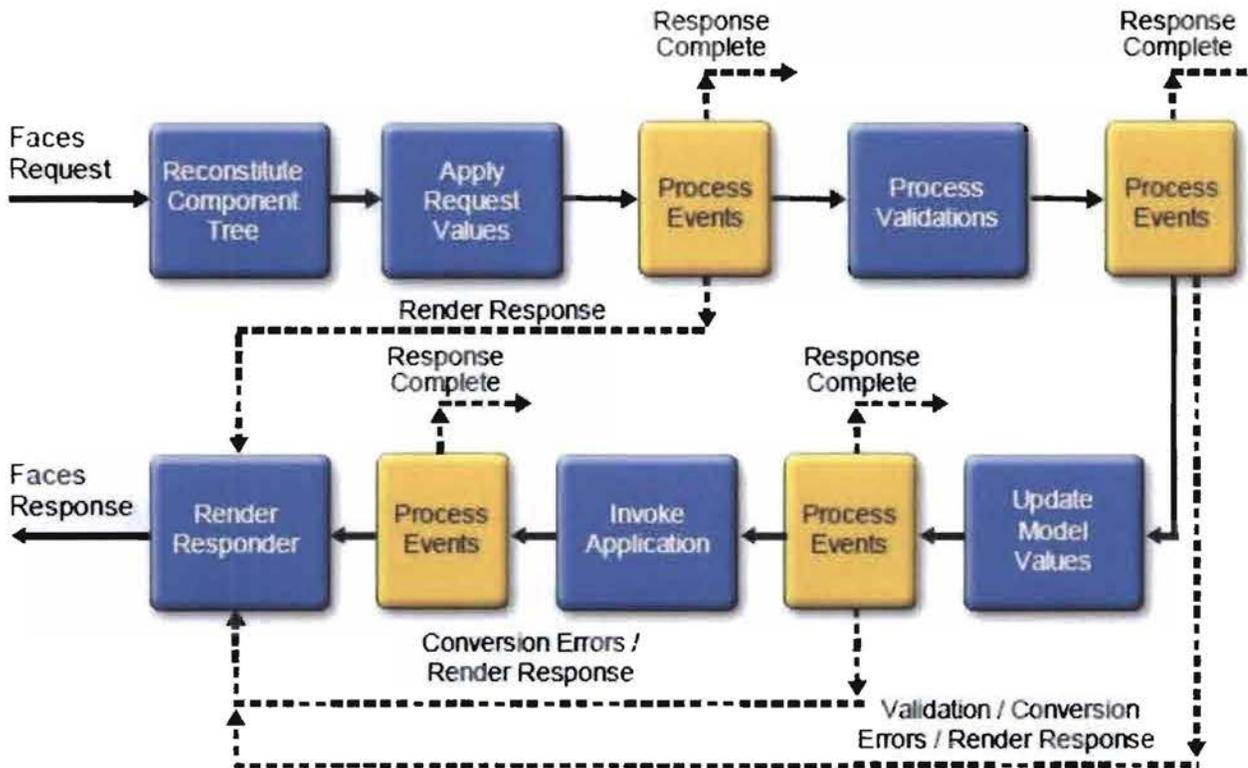


Figure A1.2 : Étapes de traitements d'une requête pour l'affichage d'une page web par JSF

Ces six étapes ne sont toutes exécutées que dans le cas d'une requête correspondant à la soumission d'un formulaire (requête HTTP de type POST).

Dans le cas du premier affichage d'une page (requête HTTP de type GET), la plupart des étapes sont sautées.

Annexe2 : Présentation de Hibernate

- Présentation

Développé par un groupe de développeurs Java dirigés par Gavin King, Hibernate est un Framework Java de persistance. Il permet de faire correspondre des tables de base de données relationnelles avec des objets java simples (POJO ou «Plain Old Java Object»). Une fois la correspondance entre les deux mondes définie, le programme Java peut manipuler toutes les données en utilisant que des JavaBean, masquant alors totalement la base de données sous-jacente et ses spécificités. Le Framework assure le remplissage de ces objets et la mise à jour de la base en se basant sur leur contenu.

Hibernate apporte donc une solution aux problèmes d'adaptation entre le paradigme objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau.

- Mise en œuvre du Framework Hibernate

Hibernate a besoin de plusieurs éléments pour fonctionner :

- ✧ Une classe de type JavaBean qui encapsule les données d'une table donnée nommée « classe de persistance » ;
- ✧ Un fichier de correspondance qui configure la correspondance entre la classe et la table ;
- ✧ Des propriétés de configuration, notamment des informations concernant la connexion à la base de données.

Une fois ces éléments correctement définis, il est possible d'utiliser Hibernate dans le code des traitements à réaliser. L'architecture de Hibernate est donc la suivante :

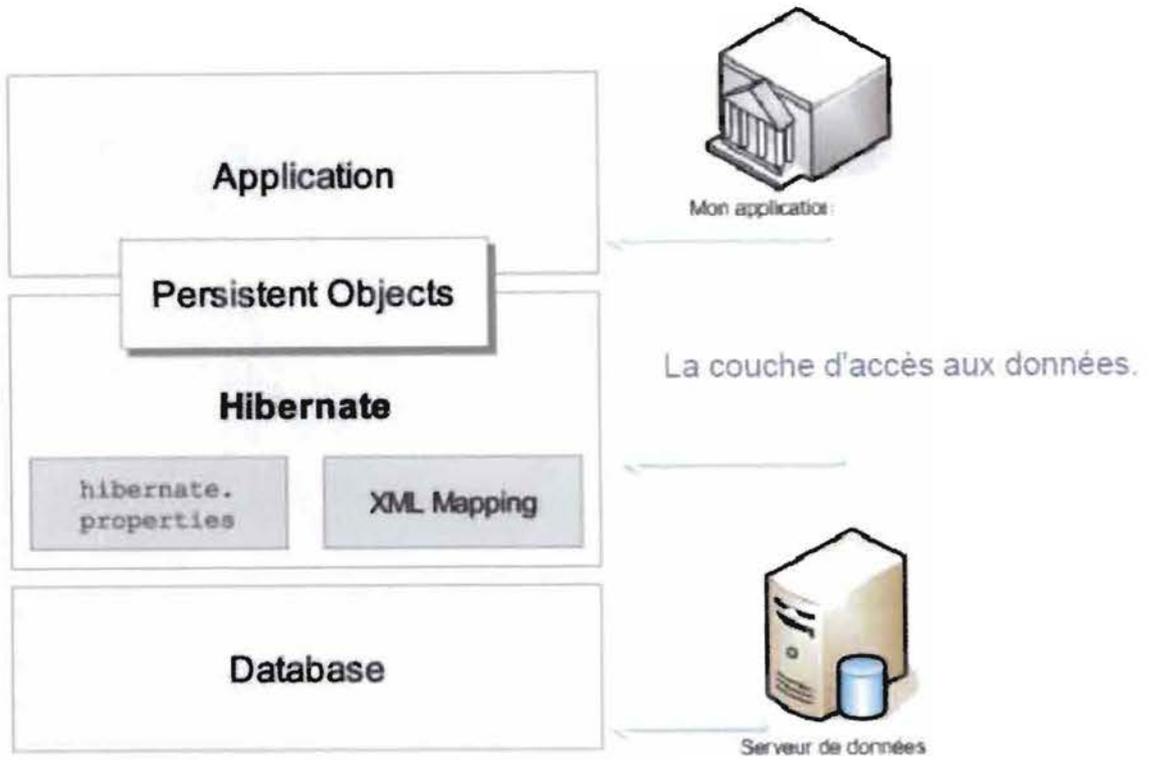


Figure A2.1 : Architecture de Hibernate

Annexe 3 : Présentation de PostgreSQL

- **Présentation**

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD.

Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MySQL et Firebird), ou propriétaires (comme Oracle, Sybase, DB2 et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises.

- **Principales caractéristiques**

PostgreSQL peut stocker plus de types de données que les types traditionnels entiers, caractères, etc. L'utilisateur peut créer des types, des fonctions, utiliser l'héritage de type etc.

PostgreSQL est pratiquement conforme (de plus en plus conforme) aux normes ANSI SQL 89, SQL 92 (SQL 2), SQL 99 (SQL 3), SQL : 2003 et SQL : 2008. Il fonctionne sur diverses plates-formes matérielles et sous différents systèmes d'exploitation.

PostgreSQL fonctionne sur Solaris, SunOS, Mac OS X, HP-UX, AIX, Linux, IRIX, Digital Unix, BSD, NetBSD, FreeBSD, OpenBSD, SCO Unix, NeXTSTEP, UnixWare et toutes sortes d'Unix. Depuis la version 8.0, PostgreSQL fonctionne également nativement sur Windows. Avant la version 8, il fallait un émulateur de type cygwin pour faire fonctionner PostgreSQL sur ce système d'exploitation.

PostgreSQL est largement reconnu pour son comportement stable, proche d'Oracle. Mais aussi pour ses possibilités de programmation étendues, directement dans le moteur de la base de données, via PL/pgSQL. Le traitement interne des données peut aussi être couplé à d'autres modules externes compilés dans d'autres langages.