

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

THEME : « GESTION AUTOMATISEE DES PUBLICATIONS DE L'INSD »

Période du 01 Septembre au 30 Novembre 2013

Auteur : Bene-wende Odilon Isaïe ZOUNGRANA

Maître de stage

M Alex SOME

Informaticien à l'INSD

Pour Dr. Sere Abdoulaye,
le directeur Adjoint **Superviseur**

Dr. Abdoulaye SERE

Enseignant chercheur à l'Unité
de Formation et de Recherche
en Science et Technologie
(UFR/ST)

Année académique : 2012-2013

DEDICACES

Au Créateur.

A mes Parents.

A tous Ceux qui me sont Chers.

REMERCIEMENTS

Au début de ce rapport, il serait trop injuste de nous accorder tous les mérites du travail réalisé. Il y'a des personnes sans qui, les couleurs de ce chef-d'œuvre auraient pu être ternies, et à qui, nous devons une fière chandelle.

Nous tenons donc à remercier dans un premier temps, toute l'équipe pédagogique de *l'Ecole Supérieure d'Informatique* et les intervenants professionnels responsables de la formation des ingénieurs de travaux en informatiques, pour avoir assuré la partie théorique de celle-ci.

Nous remercions également toute l'équipe professionnelle de l'INSD pour l'expérience enrichissante et pleine d'intérêt qu'elle nous a fait vivre durant ces trois mois au sein de l'institut. Nos vifs remerciements vont aussi aux personnes suivantes :

- **Dr SERE Abdoulaye**, enseignant à l'université polytechnique de Bobo-Dioulasso notre superviseur,
- **M. Dienyéle Alexandre SOME**, informaticien à l'INSD, notre maître de stage,

Enfin, au risque de n'oublier personne, nous témoignons notre gratitude à tous ceux qui, de près ou de loin ont apporté leur soutien à notre formation ou à la réussite de ce travail.

SOMMAIRE

DEDICACES	I
REMERCIEMENTS	II
SOMMAIRE	III
SIGLES, ACRONYMES ET ABBREVIATIONS	VI
LISTE DES FIGURES	VII
LISTE DES TABLEAUX	VIII
AVANT PROPOS	IX
INTRODUCTION GENERALE.....	1
Chapitre I: GENERALITES.....	2
INTRODUCTION.....	2
1.1.PRESENTATION DE L'INSD	2
1.1.1 Création	2
1.1.2 Activités.....	3
1.1.3. Organisation.....	4
1.2. PRESENTATION DU THEME.....	6
1.2.1. Description du thème	6
1.2.2. Problématique.....	6
1.2.3. Résultats attendus	6
1.3. LANGAGE, METHODE ET DEMARCHE D'ANALYSE.....	7
1.3.1. Le langage de modélisation	7
1.3.2. Méthode de développement.....	9
1.3.3. Démarche d'analyse.....	12
1.4. ACTEURS DU PROJET ET PLANNING PREVISIONNEL.....	13
1.4.1. Acteurs du projet	13

I.4.2. Planning prévisionnel	14
CONCLUSION	19
CHAPITRE II : ETUDES FONCTIONNELLES ET TECHNIQUES	20
INTRODUCTION.....	20
II.1. ETUDE PRELIMINAIRE	20
II.1.1. Description générale du fonctionnement.....	20
II.1.2. Fonctionnalité de GAPI.....	21
II.2. CAPTURE DES BESOINS	22
II.2.1. Capture des besoins fonctionnels	22
II.2.2. Capture des besoins techniques.....	24
II.3. L'ARCHITECTURE LOGICIELLE DE L'APPLICATION	27
II.4. L'ENVIRONNEMENT DE DEVELOPPEMENT (IDE)	28
II.5. UNE PLATEFORME PHP.....	29
II.6. OUTILS DE MODELISATION.....	30
II.7: ANALYSE DU FUTUR SYSTEME.....	30
II.7.1. Description générale.....	31
II.7.2. Estimation des Besoins.....	32
II.7.3. Coût de développement et de formation.....	33
CONCLUSION	36
Chapitre III: CONCEPTION ET DEVELOPPEMENT	37
INTRODUCTION.....	37
III.1. CONCEPTION	37
III.1.1. Diagrammes de cas d'utilisation	37
III.1.2. Diagramme de séquence	44

III.1.3. Diagramme de classe	47
III.1.4. Diagramme de déploiement	48
III.2. DEVELOPPEMENT DE GAPI	49
III.3. DEPLOIEMENT DE GAPI.....	51
III.3.1. Politique de sécurité	51
III.3.2. Gestion des attaques	51
III.3.3. Gestion des catastrophes	52
III.3.4. Gestion des incidents d'exploitation	53
III.4. POLITIQUE DE SECOURS	53
III.4.1. Plantage du système.....	53
III.4.2. Panne due à un accès au réseau et Autres pannes critiques	53
III.5.TRANSITION	53
III.5.1. Procédure transitoire.....	53
III.5.2. Formation des utilisateurs	54
CONCLUSION.....	54
CONCLUSION GENERALE	55
BIBLIOGRAPHIE ET WEBOGRAPHIE,.....	56
ANNEXES	57
Annexe 1 : Présentation de Zend Framework	57
Annexe 2 : Présentation de MySQL	58

SIGLES, ACRONYMES ET ABBREVIATIONS

<i>Sigle</i>	<i>Définition</i>
ACL	Access Control List
API	Application Programming Interface
AJAX	Asynchronous JAVascript Xml
BD	Base de Données
CICI	Cycle des Ingénieurs de Conception Informatique
CITI	Cycle des Ingénieurs des travaux Informatique
COCOMO	COnstructive COst Model
CU	Cas d'Utilisation
CSS	Cascading Style Sheets
EDI	<i>Integrated Development Environment</i>
GAPI	<i>Gestion Automatisée des Publications de l'INSD</i>
HM	Homme Machine
HOOD	Hierarchical Object Oriented Design
HTML	HyperText Markup Language
INSD	Institut National de la Statistique et de Développement
KLSL	Kilo Lignes Sources de Logiciel
JSP	Java Server Page
J2EE	Java 2 Enterprise Edition
MAJ	Mise à Jour
MVC	Model View Controller
ND	Non Disponible
OMG	Object Management Group
OMT	Object Modeling Technique
OOA	Object Oriented Analysis
OOD	Object Oriented Design
OOSE	Object Oriented Software Engineering
PHP	PHP : Hypertext Preprocessor
PU	Processus Unifié
RUP	Rational Unified Processus
SDK	Software Development Kit
SGBD	Système de Gestion de Base de Données
SQL	Structured Query Language
UML	Unified Modeling Language
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
XP	eXtreme Programming
2TUP	2 Track Unified Processus

LISTE DES FIGURES

Figure 1 : Organigramme de l'INSD.....	5
Figure 2 : Le système d'information soumis à deux natures de contraintes	11
Figure 3 : Le processus de développement en Y.....	11
Figure 4 : Diagramme de Gantt du planning prévisionnel	15
Figure 5: Diagramme de Gantt du planning réel.....	17
Figure 6 : Présentation du matériel	31
Figure 7 : Architecture réseau	32
Figure 8 : Diagramme de cas d'utilisation global	37
Figure 9 : Représentation détaillée du cas d'utilisation « gérer les documents »	38
Figure 10: Représentation détaillée du cas d'utilisation «gérer les utilisateurs»	38
Figure 11 Représentation détaillée du cas d'utilisation «gérer les Clients»	39
Figure 12 : Diagramme de séquence du cas d'utilisation « s'authentifier »	45
Figure 13 : Diagramme de séquence du cas d'utilisation «Gérer documents»	45
Figure 14 : Diagramme de séquence du cas d'utilisation «Enregistrer une commande».....	46
Figure 15 : Diagramme de séquence du cas d'utilisation «Approvisionner stock»	46
Figure 16 : Diagramme de classes.....	48
Figure 17 : Diagramme de déploiement.....	49
Figure 18 : Arborescence du projet zend framework.....	50
Figure 19 Ecran de connexion avec échec d'authentification	52
Figure 20: Page d'accueil.....	53
Figure 21: Ecran de gestion des documents	53

LISTE DES TABLEAUX

Tableau 1 : Etude comparative de quelques processus de développement	9
Tableau 2 : Planning réel.....	15
Tableau 3 : Identification des cas d'utilisations	22
Tableau 4 : Description fonctionnelle du cas d'utilisation « gérer les documents ».....	23
Tableau 5 : Description fonctionnelle du cas d'utilisation « gérer les clients».....	23
Tableau 6 : Description fonctionnelle du cas d'utilisation « gérer les utilisateurs »	24
Tableau 7 : Présentation de quelques technologies web	25
Tableau 8 : Tableau comparatif de frameworks PHP	27
Tableau 9 : Tableau comparatif d'EDI.....	28
Tableau 10 : Tableau comparatif de plateformes PHP.....	29
Tableau 11 : Estimation des besoins	32
Tableau 12 : Coût de développement.....	35
Tableau 13 : Coût de formation des utilisateurs.....	35
Tableau 14 : Coût de réalisation.....	36
Tableau 15 : Description textuelle de quelques cas d'utilisation CU « S'authentifier ».....	40
Tableau 16 : Description textuelle de quelques cas d'utilisation « Gérer un utilisateur »	41
Tableau 17 : Description textuelle de quelques cas d'utilisation « Gérer document »	42
Tableau 18 : Description textuelle de quelques cas d'utilisation « Gérer clients ».....	43
Tableau 19 : Liste des méthodes liées aux différentes classes	49

AVANT PROPOS

L'**Université Polytechnique de Bobo** (UPB), jadis Centre Universitaire de Bobo-Dioulasso (CUPB), a été créé le 23 mai 1997 par décret n°97-254/PRES/PM/MESSRS. Installée à une quinzaine de kilomètre à l'ouest de Bobo-Dioulasso, elle est composée de six (6) établissements :

- ✓ Ecole Supérieure d'Informatique (ESI) ;
- ✓ Institut de Développement Rural (IDR) ;
- ✓ Institut Universitaire de Technologie (IUT) ;
- ✓ Institut des Sciences de la Santé (INSSA) ;
- ✓ Unité de Formation et de Recherche Sciences et Technologies (UFR/S.T)
- ✓ Unité de Formation et de Recherche des Sciences Politiques, juridiques et Economie de Gestion (UFR/S.J.P.E.G)

L'Ecole Supérieure d'Informatique où nous avons suivi notre formation, a une organisation pédagogique qui s'articule autour des deux axes suivants :

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

Les étudiants en fin d'étude du Cycle des Ingénieurs de Travaux en Informatique (CITI), dans le cadre normal de renforcement et de valorisation de leur formation, effectuent un stage en entreprise au cours duquel ils sont confrontés à des problèmes informatiques réels du monde professionnel. Trois mois durant, ils travailleront à trouver des solutions conceptuelles et technologiques qui répondent aux besoins informatiques exprimés par leurs structures d'accueil.

Ce stage s'est déroulé du 01 septembre au 30 novembre 2013

INTRODUCTION GENERALE

La production et la distribution des données statistiques et des publications fiables est un enjeu important du fait de leur portée pour l'ensemble des utilisateurs en général et des dirigeants de l'Etat en particulier. La gestion des publications de l'INSD se trouve toujours à un stade embryonnaire. Ce volet de la production et de la distribution des documents, nécessite une introduction forte des nouvelles technologies afin de répondre efficacement à la forte demande des consommateurs. Avec la mise en œuvre du Schéma Directeur Statistique 2010-2015, des efforts remarquables ont été faits dans la production statistique pour répondre aux besoins de plus en plus importants et complexes des utilisateurs. La prise en compte de l'automatisation de la gestion des publications viendra en appui répondre à cette problématique de la production-distribution des publications. Aussi, l'INSD a-t-il saisi l'opportunité de ce stage pour instruire le thème "Gestion automatisée des publications de l'INSD" afin de pallier aux insuffisances et imperfection du traitement non informatisé de la gestion des documents. Le développement de l'application sur la "Gestion automatisée des publications de l'INSD" sollicité auprès des stagiaires, vient résoudre les difficultés manuelles rencontrées dans les tâches du Service de Management de l'Information Statistique (SMIS) en gestion des publications ou documents produits à l'INSD.

Ce rapport de stage qui est le résumé de notre étude sera organisé en trois chapitres. Un premier chapitre sera consacré aux généralités, un deuxième portera sur les études fonctionnelles et techniques que nous mènerons et le dernier chapitre traitera de la conception dans son intégralité.

INTRODUCTION

Ce chapitre concerne la présentation de notre structure d'accueil et du thème de notre étude. Nous allons également faire cas de notre démarche d'analyse pour mener à bien le projet. Enfin, nous présenterons les différents acteurs du projet et nous donnerons un planning prévisionnel du déroulement des différentes phases de l'analyse

I.1.PRESENTATION DE L'INSD

L'Institut National de la Statistique et de la Démographie est l'organisme officiel de l'Etat responsable de produire, d'analyser et de diffuser les informations statistiques officielles, objectives et de qualité pour le Burkina Faso. Celles-ci enrichissent les connaissances, éclairent les débats et appuient la prise de décision des différents acteurs de la société burkinabè.

I.1.1 Création

Le système statistique national du Burkina Faso doit être placé dans le cadre du développement économique, social et politique du pays. S'il est indéniable qu'on ne peut concevoir un peuple organisé sans statistiques, il convient tout de même de reconnaître que la tentative de créer un système statistique plus ou moins cohérent date de l'ère coloniale.

L'Institut National de la Statistique et de la Démographie a été créé en 1974 comme une direction centrale de l'administration puis transformé en un Établissement Public de l'État à caractère Administratif (EPA), par le Décret N°2000-508/PRES/PM/MEF du 27 Octobre 2000.

En 1974, il succédait alors à la Direction Nationale de la Statistique et de la Mécanographie créée en mars 1966 et qui entra dans la mémoire collective des Voltaïques d'alors, comme étant le service chargé du traitement de la solde des fonctionnaires. Cette direction remplaçait le Service National de la Statistique et des Études Économiques (SNSEE) lui-même créé trois ans auparavant, en février 1963, pour répondre aux nouveaux besoins en données plus

élaborées, apparus à l'aube des indépendances le 5 août 1960. La première unité à vocation statistique a été le "Bureau Statistique" créé en 1958.

I.1.2 Activités

L'INSD est l'organe officiel de l'Etat en matière d'information statistique. Son objectif premier est la collecte, le traitement, l'analyse, la publication et la diffusion de l'information statistique.

A ce titre, il est chargé de :

- promouvoir la recherche, le développement, la coordination et l'efficacité des études à caractère statistique, économique et démographique suivant des principes uniformes, conformément aux directives nationales et aux normes internationales approuvées par le Burkina Faso ;
- préparer, sur les plans technique et méthodologique la collecte des statistiques en assurant leur complémentarité et leur comparabilité ;
- effectuer le traitement, l'analyse et la publication des statistiques officielles de l'Etat, suivant les normes nationales et internationales ;
- élaborer et publier les comptes économiques ;
- préparer et exécuter les recensements généraux de la population et les enquêtes démographiques nationales ;
- préparer et exécuter les études et recherches en matière de population ;
- assurer le secrétariat permanent du Conseil National de Coordination Statistique ;
- participer à la préparation de tout règlement administratif dans le domaine de la statistique ;
- établir les budgets économiques en collaboration avec les directions et départements ministériels concernés ;
- réaliser des prévisions économiques à court, moyen et long termes à l'aide de modèles appropriés ;

- suivre la conjoncture économique nationale et internationale ;
- élaborer un Tableau de Bord Trimestriel de l'Economie ;
- élaborer le Tableau de Bord Social du Burkina Faso ;
- élaborer le Tableau de Bord Agro-pastoral et de l'Environnement ;
- mettre en place des outils de suivi des conditions de vie des ménages ;
- mener des études et offrir des prestations de services ;
- mettre en place une centrale de bilans.

I.1.3. Organisation

L'institut National de la Statistique et de la démographie est un établissement public à caractère administratif placé sous la tutelle technique du ministre chargé de la statistique, et sous la tutelle financière du ministre chargé des finances. Il est dirigé par un Directeur Général et comprend les directions suivantes :

Au niveau central :

- ✓ la direction de la coordination statistique, de la formation et de la recherche(DCFR)
- ✓ la direction de la démographie(DD)
- ✓ la direction des statistiques et des synthèses économiques(DSSE)
- ✓ la direction des statistiques et des conditions de vie des ménages(DSCVM)
- ✓ Direction de l'Informatique et du Management de l'Information Statistique (DIMIS)
- ✓ la direction de l'administration et des finances(DAF)
- ✓ l'agence comptable(AC)

Au niveau régional

- ✓ les directions régionales de l'Institut Nationale de la Statistique et de la Démographie (DR/ISND)

L'organigramme de l'institut est présenté à la figure 1

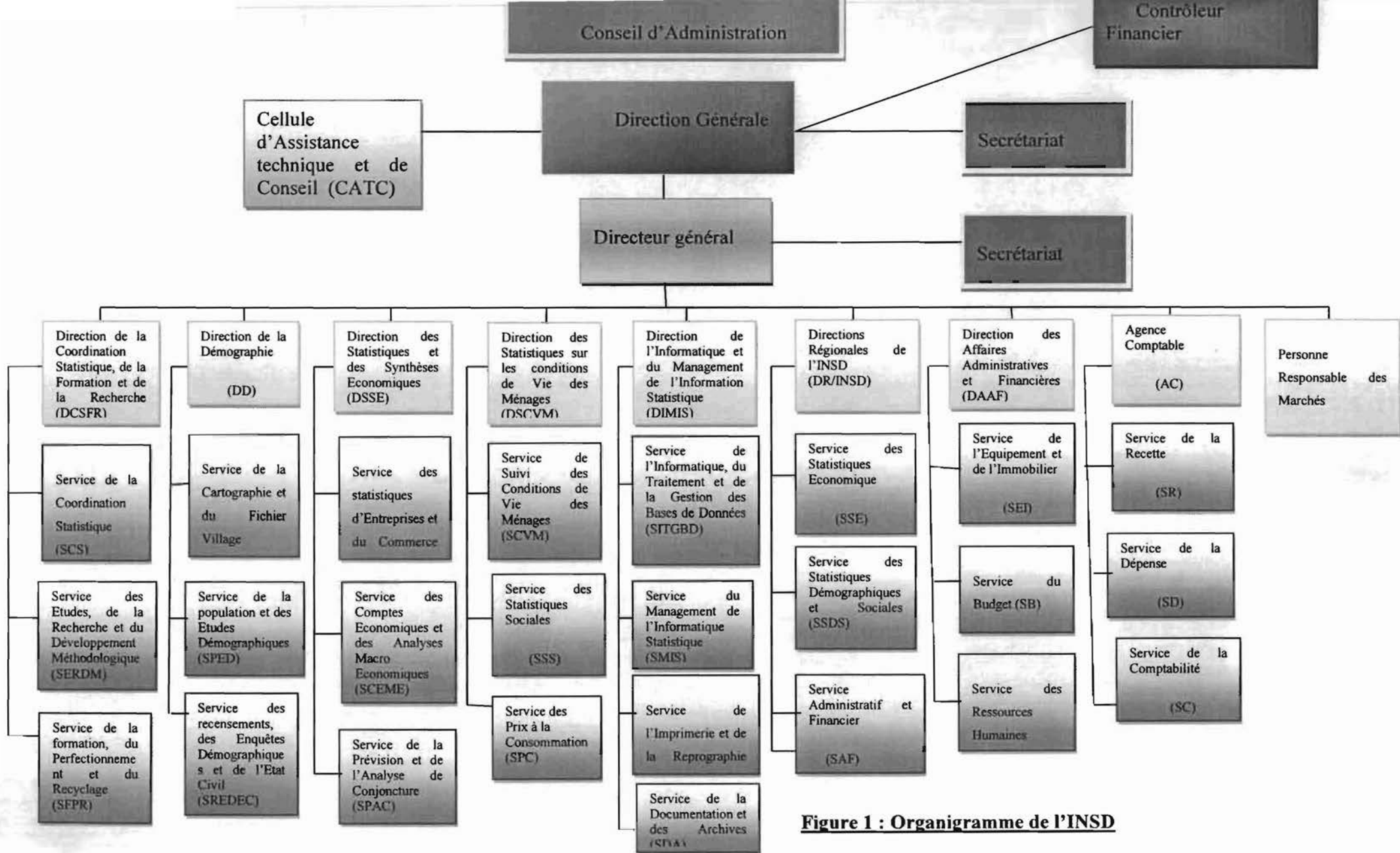


Figure 1 : Organigramme de l'INSD

I.2. PRESENTATION DU THEME

I.2.1. Description du thème

Notre thème «Gestion automatisée des publications de l'INSD» stipule la mise en place d'une application pouvant gérer de façon rationnelle et fiable, le processus d'écoulement des documents produits à l'INSD.

I.2.2. Problématique

L'Institut National de la Statistique et de la Démographie mène des enquêtes suite auxquelles paraissent des documents. Ceux-ci sont mis à la disposition de toute personne désirant les consulter. On peut donc s'en procurer auprès du service marketing de la structure. Le problème qui se pose c'est que la gestion de ces documents au niveau de l'INSD se fait jusque-là manuellement, sans aucun outil informatique, ce qui entraîne des inconstances au niveau des traitements des commandes, des produits en stock, de la distribution, de la traçabilité des documents écoulés, etc.

I.2.3. Résultats attendus

En référence à ce qui a été formulé à la problématique, le projet à terme doit permettre d'avoir une information à jour sur les documents et les clients qui s'y sont intéressés. C'est-à-dire tous ce qui a concerné le document depuis sa production jusqu'à l'utilisateur final.

Il nous revient donc d'effectuer une analyse complète des différentes activités menées par les agents du service marketing concernant la publication des documents et de procéder à l'informatisation des tâches identifiées au cours de cette analyse. La solution devra donc permettre :

- ✓ la gestion des documents
- ✓ la gestion des clients
- ✓ la traçabilité des documents
- ✓ l'état de stock
- ✓ les recettes des documents.

I.3. LANGAGE, METHODE ET DEMARCHE D'ANALYSE

La construction d'un logiciel est complexe parce qu'elle met en œuvre de nombreuses ressources : humaine, matérielles, technologiques. En plus de cela pour des raisons de qualités en termes de robustesse et de précision, il est nécessaire d'utiliser un processus bien défini, un langage de modélisation approuvé et des techniques de modélisation rigoureuses.

I.3.1. Le langage de modélisation

La modélisation est une technique d'ingénierie qui permet de comprendre un système par l'établissement de modèles pour mettre au point une solution à un problème.

Elle nous aide à représenter un système :

- ✓ en précisant sa structure ;
- ✓ en définissant ce qu'il fait ;
- ✓ en déterminant comment il le fait ;
- ✓ en fournissant un canevas qui guide sa construction ;
- ✓ en le documentant.

Pour ce faire nous avons choisi d'utiliser pour ce projet le langage UML (Unified Modeling Language) à cause de sa façon de procéder qui cerne tous les contours de notre projet.

UML est un langage de modélisation fondé sur les concepts orientés objets. Il a été conçu pour la modélisation de tous les phénomènes de l'activité de l'entreprise indépendamment des techniques d'implémentation mises en œuvre par la suite. Il est né de la fusion de trois (03) méthodes de référence : OMT, Booch, OOSE. UML n'impose pas une démarche particulière pour l'analyse d'un système mais préconise d'adopter une démarche ayant les caractéristiques suivantes :

- ✓ itérative et incrémentale ;
- ✓ guidée par le besoin des utilisateurs du système ;
- ✓ centrée sur l'architecture logicielle.

Pour faciliter la vision du non-programmeur et le travail du programmeur et afin d'obtenir une vue d'ensemble du système en un temps beaucoup plus court, UML a initié le concept visuel

fondé sur les diagrammes. UML 2.0 définit treize (13) diagrammes repartis selon les modèles suivants :

➤ les modèles statiques

Ce sont des diagrammes qui permettent de visualiser, spécifier, construire et documenter l'aspect statique ou structurel du système d'information. Les modèles statiques sont entre autre le diagramme de classe, le diagramme d'objets, le diagramme de déploiement, le diagramme de structure composite, le diagramme de composants et le diagramme de paquetages.

➤ les modèles dynamiques

Ils modélisent les aspects dynamiques du système, c'est-à-dire les différents éléments qui sont susceptibles de subir des modifications. Parmi eux, on distingue les diagrammes de cas d'utilisation, les diagrammes d'états - transitions et les diagrammes d'activités.

➤ les modèles d'interactions

Ils représentent les interactions entre le système lui-même et les différents acteurs du système, mais aussi la façon dont les différents objets contenus dans le système communiquent entre eux. Il s'agit des diagrammes de séquence, de communication (collaboration), d'interaction globale et de temps.

Les avantages présentés par UML, nous permettent de faciliter la modélisation des différents aspects de notre projet. En effet UML présente l'avantage d'être le standard de la modélisation objet universellement reconnu. Il est un langage visuel. Sa notation graphique permet d'exprimer visuellement des solutions objets facilitant ainsi la comparaison et l'évaluation de celles-ci. C'est un langage formel et normalisé doté d'un gain de précision et d'un gage de stabilité. Il est aussi un support de communication performant car il cadre l'analyse tout en facilitant la compréhension des représentations abstraites complexes. En outre, UML sert à formaliser tous les documents techniques d'un projet et permet d'affiner les détails de l'analyse au fur et à mesure de l'avancée du projet. Il est possible d'utiliser le même atelier de génie logiciel depuis l'expression des besoins jusqu'à la génération de tout ou d'une partie du code. Enfin, il est indépendant des langages de programmation et des processus de développement.

UML propose un moyen pour représenter diverses projections d'un système: les vues.

Elles sont généralement constituées d'un ou plusieurs diagrammes UML qui sont des représentations graphiques qui s'intéressent à un aspect précis du modèle dont chaque type est

composé d'éléments de modélisation prédéfinis et la combinaison offre une vue complète des aspects fonctionnels, statiques et dynamiques d'un système.

UML est une avancée importante pour le génie logiciel mais ce n'est ni une méthode, ni un processus. Si UML permet de modéliser un système, il ne définit pas le processus d'élaboration des modèles.

Dans quel ordre doit-on utiliser les treize diagrammes ?

A quel moment de la conception d'un système doivent-ils intervenir ?

Seul un processus de développement peut répondre à ces questions !

I.3.2. Méthode de développement

I.3.2.1 Choix de la méthode de développement

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

Comment vais-je organiser les équipes de développement ?

Quelles tâches attribuer à qui ?

Quel temps faudrait-il pour livrer le produit ?

Comment faire participer le client au développement afin de capter les besoins de celui-ci ?

Comment éviter des dérives et de mauvaises estimations qui vont allonger les coûts et le temps de développement ?

Comment vais-je procéder pour que le produit soit évolutif et facilement maintenable ?

Nous présentons ici quelques processus de développement sous forme de comparaison comme le montre le tableau 1.

Tableau 1 : Etude comparative de quelques processus de développement

Processus	Description	Points forts	Points faibles
Cascade	-Propose de dérouler les phases projet de manière séquentielle -Citée pour des raisons historiques	Distingue clairement les phases projet	-Non itératif -Ne propose pas de modèles de documents
RUP	-Promu par Rational. -Le RUP est à la fois une méthodologie et un outil prêt à l'emploi (documents types partagés dans un référentiel Web)	-Itératif -Spécifie le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes...	-Coûteux à personnaliser : batterie de consultants -Très axé processus, au détriment du développement : peu de place pour le code et la technologie

	-Cible des projets de plus de 10 personnes	-Propose des modèles de documents, et des canevas pour des projets types	
XP	-Ensemble de «Bests Practices» de développement (travail en équipes, transfert de compétences...) -Cible des projets de moins de 10 personnes	-Itératif, simple à mettre en œuvre -Fait une large place aux aspects techniques : prototypes, règles de développement, tests -Innovant: programmation en duo, kick-off matinal debout ...	-Ne couvre pas les phases en amont et en aval au développement : capture des besoins, support, maintenance, tests d'intégration... -Elude la phase d'analyse, si bien qu'on peut dépenser son énergie à faire et défaire -Assez flou dans sa mise en œuvre: quels intervenants, quels livrables ?
2TUP	-S'articule autour de l'architecture -Propose un cycle de développement en Y -Cible des projets de toutes tailles	-Itératif -Fait une large place à la technologie et à la gestion du risque -Définit les profils des intervenants, les livrables, les plannings, les prototypes	-Plutôt superficiel sur les phases situées en amont et en aval du développement: capture des besoins, support, maintenance, gestion du changement... -Ne propose pas de documents types

Notre choix s'est porté sur la méthode **2TUP**, du fait de son approche nouvelle, originale. Notre projet est basé sur un processus de développement bien défini qui va de la détermination des besoins fonctionnels attendus du système jusqu'à la conception et le codage final.

1.3.2.2. Le processus de développement 2TUP

« La famille des " Unified Process (UP)" constitue une trame commune pour intégrer les meilleures pratiques de développement. Un processus UP est itératif et incrémental, centré sur l'architecture, conduit par les exigences des utilisateurs, piloté par les risques et orienté composants. Le processus 2TUP signifie « 2 Track Unified Process». C'est un processus UP qui répond aux caractéristiques que nous venons de citer. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. «2 Track» signifie littéralement que le processus suit deux chemins. Il s'agit des chemins «fonctionnels» et «d'architecture technique», qui correspondent aux deux axes de

changement imposés au système informatique. La figure 2 nous donne un aperçu de ses deux contraintes.

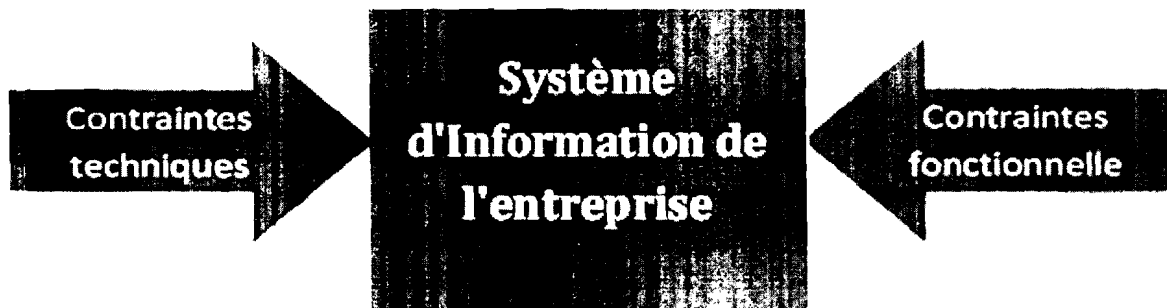


Figure 2 : Le système d'information soumis à deux natures de contraintes

L'axiome fondateur du 2TUP consiste à constater que toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement, suivant un axe fonctionnel et un axe technique. »

À l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des deux branches. Cette fusion conduit à l'obtention d'un processus de développement en forme de Y, comme illustré par la figure 3.

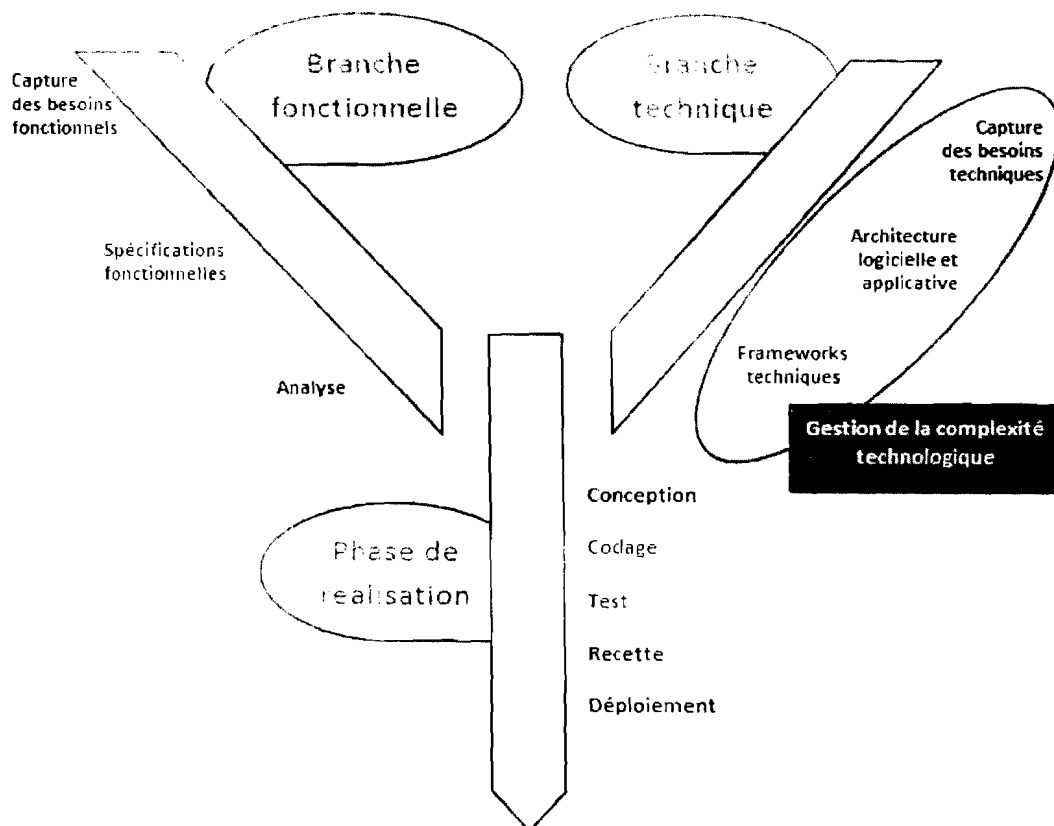


Figure 3 : Le processus de développement en Y

☞ **La branche gauche (fonctionnelle) comporte :**

- ✓ la capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs. De son côté, la maîtrise d'œuvre consolide les spécifications et en vérifie la cohérence et l'exhaustivité de l'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier. Les résultats de l'analyse ne dépendent d'aucune technologie particulière.

☞ **La branche droite (architecture technique) comporte :**

- ✓ la capture des besoins techniques, qui recense toutes les contraintes et les choix dimensionnant la conception de système. Les outils et les matériels sélectionnés ainsi que la prise en compte de contraintes d'intégration avec l'existant conditionnent généralement des prérequis d'architecture technique ;
- ✓ la conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est la moins dépendante possible des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser le même mécanisme pour tout un système.

L'architecture technique construit le squelette du système informatique et écarte la plupart des risques de niveau technique. L'importance de sa réussite est telle qu'il est conseillé de réaliser un prototype pour assurer sa validité.

☞ **La branche du milieu comporte :**

- ✓ la conception préliminaire, qui représente une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer ;
- ✓ la conception détaillée, qui étudie ensuite comment réaliser chaque composant ;
- ✓ l'étape de codage, qui produit ces composants et teste au fur et à mesure les unités de code réalisées ;

L'étape de recette, qui consiste enfin à valider les fonctions du système développé.

1.3.3. Démarche d'analyse

2TUP est organisée suivant les quatre phases suivantes : **initialisation, élaboration, construction et transition.**

➤ La phase d'initialisation :

Elle conduit à définir la «vision» du projet, sa portée, sa faisabilité, son business case, afin de pouvoir décider au mieux de sa poursuite ou de son arrêt.

➤ La phase d'élaboration :

Elle poursuit trois objectifs principaux en parallèle :

- ✓ identifier et décrire la majeure partie des besoins des utilisateurs ;
- ✓ construire (et pas seulement décrire dans un document !) l'architecture de base du système ;
- ✓ lever les risques majeurs du projet.

➤ La phase de construction :

Elle consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base). C'est la phase la plus consommatrice en ressources et en effort.

➤ la phase de transition :

Elle permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux. Les mots-clés sont : conversion des données, formation des utilisateurs, déploiement, béta-tests.

1.4. ACTEURS DU PROJET ET PLANNING PREVISIONNEL

1.4.1. Acteurs du projet

Ce sont toutes les personnes qui interviennent dans l'exécution de ce projet. Ils sont regroupés en groupe de pilotage, groupe de projet et en groupe des utilisateurs.

1.4.1.1. Groupe de pilotage

Le groupe de pilotage prend les décisions relatives aux objectifs recherchés et fixe les orientations générales, les délais à respecter et 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 s'agit notamment de :

- *M. SOME Dienyéle Alexandre* (vice-directeur du service informatique à l'INSD)
- *Dr. SERE Abdoulaye* (enseignant à l'Université de Polytechnique BOBO-DIOULASSO)

1.4.1.2. Groupe de projet

Le groupe de projet est chargé de l'exécution du projet, de la conception et éventuellement la réalisation du projet. Ce groupe est composé de :

- *ZOUNGRANA B Odilon Isaïe* (Elève ingénieur de travaux informatique à l'ESI, Stagiaire à l'INSD)

1.4.1.3. Groupe des utilisateurs

Le groupe des utilisateurs a un rôle consultatif, il est chargé de fournir toutes les informations nécessaires à la bonne conduite du projet. Il intervient également dans la validation des dossiers d'étude produits par le groupe de projet. Ce groupe est constitué des agents du service marketing de l'INSD.

I.4.2. Planning prévisionnel

La réussite d'un projet dépend fortement de la capacité de ses acteurs à prévoir et à planifier. Sur la base des plannings prévisionnels de certains de nos prédécesseurs et de nos estimations, nous avons établi le planning suivant :

On distingue les phases suivantes dans le planning :

- * Phase 1 : Accueil
- * Phase 2 : Analyse
- * Phase 3 : Prise en main des outils de développement
- * Phase 4 : Développement
- * Phase 5 : Rédaction du rapport de stage

Ce planning est traduit par la figure 4 qui nous montre bien la succession des tâches.

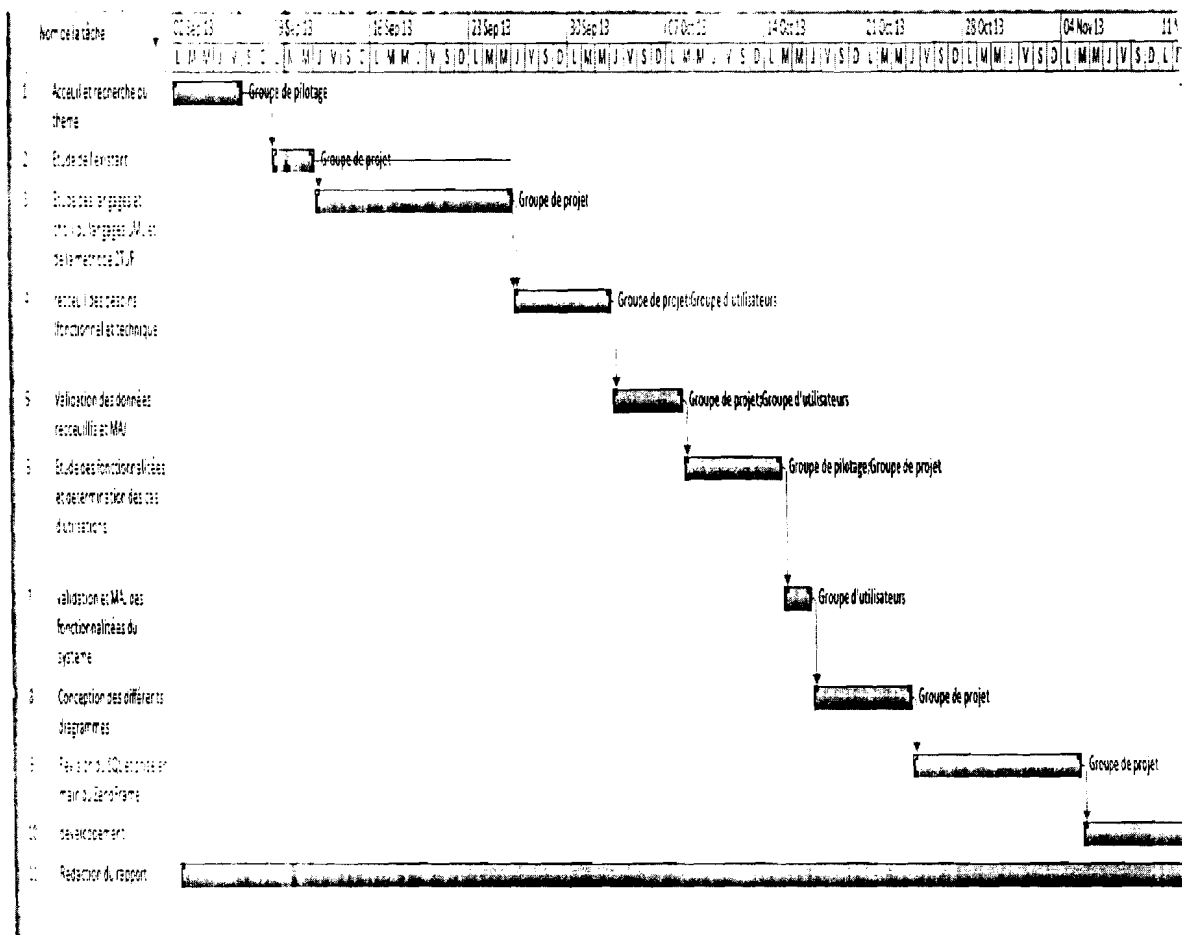


Figure 4 : Diagramme de Gantt du planning prévisionnel

Dans la pratique, les choses ne se sont pas passées comme prévu. D'autres tâches se sont ajoutées ou ont été modifiées en terme de durée ou de succession. Le tableau 2 nous donne les tâches du planning réel.

Tableau 2 : Planning réel

Mois	Jours	N°	Tâches	Dépendance	Phases	
Septembre	02-03	01	Accueil, modalités sur le stage	-	1	
	04	02	Présentation et visite de l'INSD	01	2	5
	05-09	03	Attente de thème, Formation en Outlook	02		
	10	04	Echange avec le service marketing pour dégager le thème	02		
	11-12	05	Interview pour comprendre et élucider le thème			
	13-16	06	Compréhension et acception du thème,			

			resoumis au service marketing		
	17-18	07	Recherche sur le thème	06	
	19-25	08	Etude de l'existant, Etude du langage UML et du processus 2TUP		
	A partir du 19		Début de rédaction du rapport		
Septembre- Octobre	26-02	09	Etude des fonctions, détermination des cas d'utilisations	05	
Octobre	03-09	10	Recueil des besoins (fonctionnel et technique)	06	
	10-16	08	Validation des données recueillies	07	
	17-23	09	Conception des différents diagrammes		
	22-25	10	Elaboration du SGBD		
	28	11	Validation et MAJ des données recueillies		
	29-31	12	Installation d'outils de développement, Zend Framework	08	3
Novembre	01-12	13	Révision du SQL, prise en main de zend framework	09	
	06-18	14	Développement	10	4
	19		Prise en compte des dernières remarques		
	20-29	12	Développement	11	
3 mois	90 jours				

Ce tableau nous a permis d'avoir le diagramme de Gantt de la figure 5:

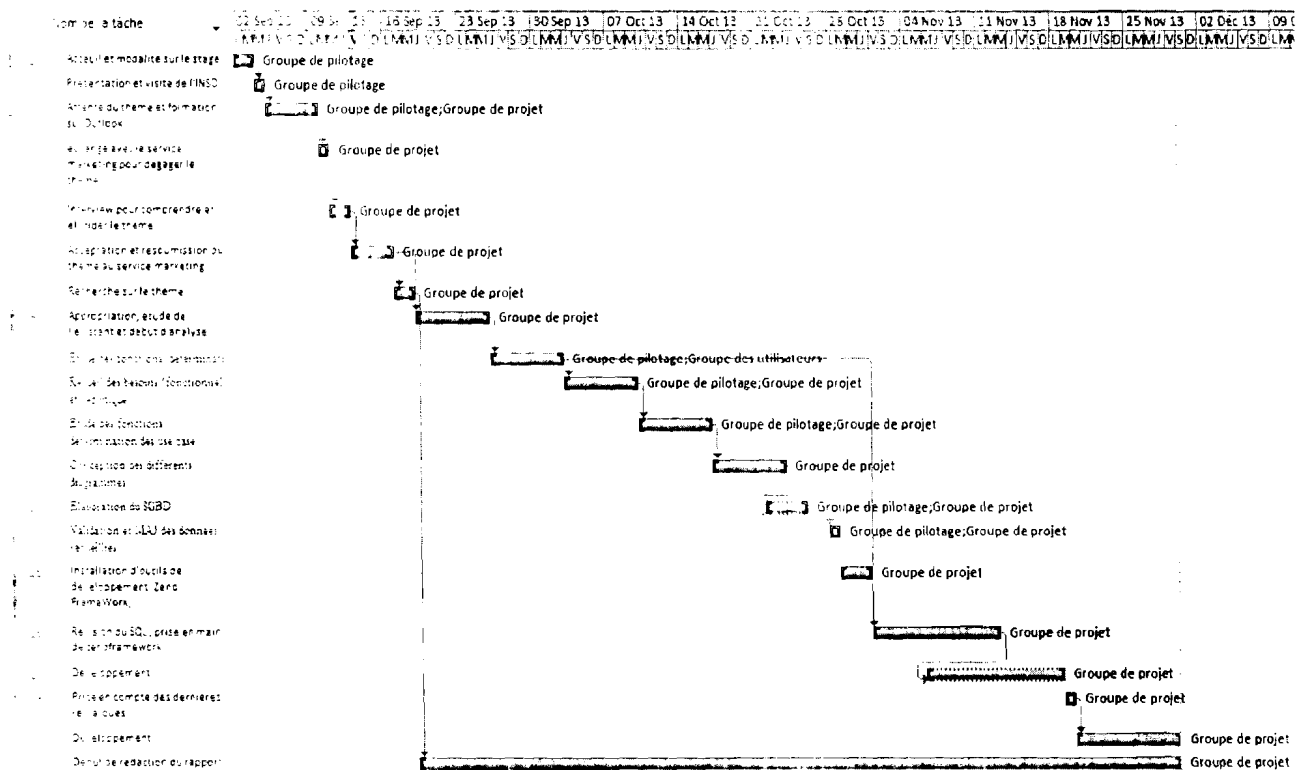


Figure 5: Diagramme de Gantt du planning réel

1.4.2.1. Explication des écarts

Pour un certain nombre de raisons, nous n'avons pas pu respecter le planning prévisionnel

Les figures 6 et 7 nous donnent respectivement la liste des tâches prévisionnelles et celle des tâches réelles

	Nom de la tâche	Durée	Début	Fin	Prédécesseurs
1	Accueil et recherche du thème	5 jours	Lun 02/09/13	Ven 06/09/13	
2	Etude de l'existant	3 jours	Lun 09/09/13	Mer 11/09/13	1
3	Etude des langages et choix du langage UML et de la méthode 2TUP	10 jours	Jeu 12/09/13	Mer 25/09/13	2
4	recueil des besoins (fonctionnel et technique)	5 jours	Jeu 26/09/13	Mer 02/10/13	3;2
5	Validation des données recueillies et MAJ	3 jours	Jeu 03/10/13	Lun 07/10/13	4
6	Etude des fonctionnalités et détermination des cas d'utilisations	5 jours	Mar 08/10/13	Lun 14/10/13	5
7	validation et MAJ des fonctionnalités du système	2 jours	Mar 15/10/13	Mer 16/10/13	6
8	Conception des différents diagrammes	5 jours	Jeu 17/10/13	Mer 23/10/13	7
9	Revision du SQL et prise en main du ZendFrame	8 jours	Jeu 24/10/13	Lun 04/11/13	8
10	developpement	17 jours	Mar 05/11/13	Mer 27/11/13	9
11	Rédaction du rapport	65 jours	Lun 02/09/13	Ven 29/11/13	

Figure 6 : Liste des tâches prévisionnelles

L'obtention du thème ne s'est pas faite dans les premiers jours comme nous l'avions indiqué dans le planning.

La merveilleuse formation que nous avons bénéficié sur Outlook nous a pris quelques jours également

La validation et la mise à jour des données ont mis plus de temps que prévu. Nous avons reporté plusieurs rendez-vous au niveau des agents du service marketing pour des raisons de mission.

Entre autres, nous avons été rappelé au cours du mois de Novembre (du 25/11 /2013 au 07/12/2013) par notre école pour un cours de rattrapage d'une matière dénommée « Programmation Mathématique et Optimisation »

La prise en main des outils de développement (la plateforme zendframework) a pris également plus de temps que prévu. Ce qui conduit au diagramme de gant de la figure 7 :

	Nom de la tâche	Durée	Début	Fin	Prédécesseurs
1	Accueil et modalités sur le stage	2 jours	Lun 02/09/13	Mar 03/09/13	
2	Présentation et visite de l'INSD	1 jour	Mer 04/09/13	Mer 04/09/13	1
3	Attente du thème et formation sur Outlook	3 jours	Jeu 05/09/13	Lun 09/09/13	2
4	échange avec le service marketing pour dégager le thème	1 jour	Mar 10/09/13	Mar 10/09/13	3
5	Interview pour comprendre et élucider le thème	2 jours	Mer 11/09/13	Jeu 12/09/13	4
6	Acceptation et resoumission du thème au service marketing	2 jours	Ven 13/09/13	Lun 16/09/13	5
7	Recherche sur le thème	2 jours	Mar 17/09/13	Mer 18/09/13	6
8	Appropriation, étude de l'existant et début d'analyse	5 jours	Jeu 19/09/13	Mer 25/09/13	6
9	Etude des fonctions déterminantes	5 jours	Jeu 26/09/13	Mar 02/10/13	8
10	Recueil des besoins fonctionnel et technique	5 jours	Jeu 03/10/13	Mar 09/10/13	9
11	Etude des fonctions, détermination des use case	5 jours	Jeu 10/10/13	Mer 16/10/13	10
12	Conception des différents diagrammes	3 jours	Jeu 17/10/13	Lun 21/10/13	11
13	Elaboration du SGBD	4 jours	Mar 22/10/13	Ven 25/10/13	12
14	Validation et MAJ des données recueillies	1 jour	Lun 28/10/13	Lun 28/10/13	12
15	Révision du SQL	3 jours	Mar 29/10/13	Jeu 31/10/13	
16	Installation d'outils de développement, Zend Framework	3 jours	Ven 01/11/13	Mar 05/11/13	9
17	Développement	9 jours	Mer 06/11/13	Lun 18/11/13	16
18	Prise en compte des dernières remarques	1 jour	Mar 19/11/13	Mar 19/11/13	
19	Développement	8 jours	Mer 20/11/13	Ven 29/11/13	18
20	Début de rédaction du rapport	35 jours	Lun 14/10/13	Ven 29/11/13	

Figure 7 : Liste des tâches réelles

CONCLUSION

Cette première étape nous a permis de nous familiariser avec notre structure d'accueil et aussi de bien nous imprégner du thème d'étude. Nous avons pu définir une démarche d'analyse qui va nous permettre de réaliser notre projet. Dans la suite du travail, nous, nous emploierons à suivre les différentes étapes de notre démarche.

INTRODUCTION

Le chapitre précédent fut l'objet entre autre de la description du processus 2TUP. Nous avons vu que ce processus est un processus en forme de Y et est décomposé en 2 branches d'étude : La branche fonctionnelle et la branche technique. Ce chapitre fera l'objet d'une part de l'Etude préliminaire et d'autre part de l'étude de notre système selon ces 2 branches et enfin d'une analyse du futur système pour déterminer le coût de réalisation.

II.1. ETUDE PRELIMINAIRE

II.1.1. Description générale du fonctionnement

➤ Enquête

L'enquête est le point de départ de l'établissement d'un document. Un service technique exprime et matérialise le besoin d'initier une enquête qui va nécessairement aboutir à la production d'un document. Cette enquête peut être un devoir périodique, ou seulement entamé sur demande ou nécessité. En fait, à la fin de l'enquête, les données sont rassemblées, corroborées et analysées. Ainsi donc, elle aboutit à l'élaboration d'un document mis au service de qui le désire.

➤ Document

Il est le résultat d'une enquête dirigée par un service technique, même si des documents paraissent parfois périodiquement, sans enquête préalable. Ces documents sont donc reproduits au niveau de la reprographie, puis transmis au service marketing, qui procède à la classification, au stockage et à la distribution ou écoulement pour la clientèle. Le document est distribué au service public désireux d'utiliser le document à des fins nationales, vendus aux privés et aux particuliers. Toutefois, tout document dont le coût de production est inférieur à 1000 FCFA est donné gratuitement à qui veut.

➤ Clientèle et acquisition

Peut être client, tout organe ou toute personne désirant consulter le document. En s'adressant au service marketing, on peut acquérir le document.

II.1.2. Fonctionnalité de GAPI

La description générale du fonctionnement de l'application nous permet de recenser les fonctions suivantes réparties en modules :

➤ Gérer Document

Ce module permettra les actions qui ont trait direct aux documents : Leur manipulation, leur classification, leur stockage ... Ainsi, le document qui est beaucoup plus manipulé par le service marketing, aura des actions qui le concernent telles, **enregistrer document, voir recettes du document, suivre ses traces, ...**

➤ Gérer Client

Les clients représentent tous ceux qui doivent ou veulent acquérir un document quelconque. Ce module doit donc pouvoir permettre de récapituler les habituels clients, de voir les meilleurs clients, d'ajouter un nouveau client, de supprimer un client.

➤ La gestion Stock

C'est le lieu où sera géré le stock des documents. Ce module a pour principales fonctionnalités, voir état du stock, vérifier les sorties, évaluer recettes totales, les documents en rupture et ceux qui seront disponibles sous peu.

➤ La gestion des paramètres et des utilisateurs

Les accès à l'application doivent être contrôlés d'où la nécessité de ce module. Il doit permettre d'ajouter un utilisateur et de lui attribuer des rôles et lui permettre de gérer son compte. Les paramètres basiques et avancés de l'application pourront aussi être définis.

II.2. CAPTURE DES BESOINS

II.2.1. Capture des besoins fonctionnels

II.2.1.1. Cas d'utilisation

Un cas d'utilisation est un ensemble d'actions susceptibles d'être réalisées par un système et produisant un résultat observable intéressant pour un acteur particulier du système. C'est l'image d'une fonctionnalité du système, déclenchée en réponse à la stimulation d'un acteur.

Il illustre, détecte puis décrit un besoin d'un utilisateur. La totalité des cas d'utilisation constitue l'ensemble des fonctionnalités du système.

Le diagramme des cas d'utilisation représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il sert à structurer les besoins des utilisateurs et les objectifs correspondants du système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée à l'acteur concerné. C'est en regroupant les intentions fonctionnelles des utilisateurs en unités cohérentes qu'on identifie les cas d'utilisation.

II.1.1.2. Identification des cas d'utilisation

Le système étudié présente un très grand nombre de fonctionnalités donc de cas d'utilisation. Ces cas d'utilisation seront repartis en cas d'utilisation de haut niveau et ses cas seront détaillés par la suite pour une plus grande lisibilité. Les cas d'utilisation de haut niveau s'identifient comme étant les différents modules du système. On obtient donc le tableau 3 contenant les différents cas d'utilisation.

Tableau 3 : Identification des cas d'utilisations

Cas d'utilisation	Acteur(s)	Message(s) émis	Message(s) reçu
Gérer les enquêtes	Agents Service Marketing	Saisie/MAJ, consulter	Liste des Enquêtes
Gérer les Services techniques	Agents Service Marketing	Saisie/MAJ, consulter	Liste des Services Techniques
Gérer les Commandes	Agents Service Marketing	Saisie/MAJ, consulter	Liste des Commandes
Gérer les Types de Documents	Agents Service Marketing	Saisie/MAJ, consulter	Liste des types de Documents
Gérer le Stock	Agents Service Marketing	Saisie/MAJ, consulter	Liste des Documents

Gérer la Traçabilité	Agents Service Marketing	Saisie/MAJ, consulter	Coordination Documents/Clients
Gérer les Clients Physiques	Agents Service Marketing	Saisie/MAJ, consulter	Liste des clients Physiques
Gérer les Clients Moraux	Agents Service Marketing	Saisie/MAJ, consulter	Liste des clients Moraux
Gérer les Commandes des Clients	Agents Service Marketing	Saisie/MAJ, consulter	Liste des Commandes
Gérer les Profils	Administrateur	Saisie/MAJ, consulter	Liste des profils
Gérer les Modules	Administrateur	Saisie/MAJ, consulter	Liste des Modules
Gérer les Privilèges	Administrateur	Saisie/MAJ, consulter	Liste Privilèges

II.1.1.3. Description fonctionnelle des cas d'utilisation

➤ Gérer Document

Le tableau 4 donne la description fonctionnelle du cas d'utilisation « gérer les documents »

Tableau 4 : Description fonctionnelle du cas d'utilisation « gérer les documents »

Cas d'utilisation	Description
Gérer les documents	Enregistre les informations sur un document. Est lié à type de document, service technique, enquête, commande
Gérer les Enquêtes	Enregistre les informations sur une Enquêtes. Est lié à Service technique, Document
Gérer les Services technique	Enregistre les informations sur un service technique. Est lié à Document, enquête
Gérer les types de documents	Enregistre les Différents types de documents. Est lié à Document
Gérer les Stocks	Permet de lister les Documents disponibles et leur quantité. Mais en relation Document, Commande
Gérer traçabilité	Met en relation le document et le client qui l'acquiert

➤ Gérer Clients

Le tableau 5 donne la description fonctionnelle du cas d'utilisation « gérer les clients »

Tableau 5 : Description fonctionnelle du cas d'utilisation « gérer les clients »

Gérer les Clients Physiques	Enregistre les informations sur une personne physique. Est en relation avec Client
-----------------------------	--

Gérer les Clients Moraux	Enregistre les informations sur une personne morale. Est en relation avec Client
Gérer les Commandes des Clients	Met en relation les clients et leur commande

➔ Gérer utilisateurs

Le tableau 6 donne la description fonctionnelle du cas d'utilisation « gérer les utilisateurs »

Tableau 6 : Description fonctionnelle du cas d'utilisation « gérer les utilisateurs »

Gérer utilisateur	Enregistre les informations sur un utilisateur. Est en relation avec commande et document
Gérer profil	Enregistre les différents profils des utilisateurs. Est en relation avec utilisateur et privilège
Changer mot de passe	Change le mot de passe d'un utilisateur

II.2.2. Capture des besoins techniques

La capture des besoins techniques est une étape qui traite des contraintes techniques et logicielles. C'est le lieu où l'on fait les choix nécessaires (technologies, outils, matériel) pour un déploiement de son système. Ces choix sont soumis aux contraintes de performance d'accès aux données, de sécurité du système, de l'interopérabilité, de la volumétrie et du mode d'utilisation du système. Pour faire un choix judicieux il est important de faire une étude comparative afin de recenser les inconvénients, les avantages et aussi une évaluation du coût des outils logiciels qui pourraient être utilisés. Ces choix portent principalement sur le système de gestion de base de données, le langage de programmation, l'architecture logicielle de l'application et l'environnement de développement intégré (IDE).

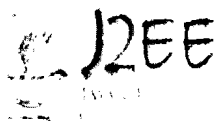

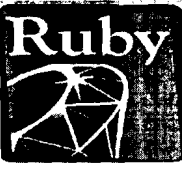
II.2.1.1. Le langage de programmation

Le langage de programmation est la matière première des programmes informatiques. Il existe plusieurs langages qu'on peut classer selon différents critères : concepts utilisés (déclaratifs ou impératifs), génération (L3G, L4G)... Pour les applications d'entreprise, on rencontre le plus souvent la technologie Java ou la technologie .NET et d'autres.

Avec l'accord du groupe de pilotage, il a été décidé de mettre en place un système de type application web. Pour de telles applications, parmi les langages les plus populaires, on peut citer: Java (*JSP+servlets*), *PHP*, *Ruby*. Il n'y a pas de meilleur langage dans l'absolu, tous ces

langages que nous avons cités peuvent permettre d'atteindre les objectifs. Nous présentons succinctement à travers le tableau 7 ces trois langages.

Tableau 7 : Présentation de quelques technologies web

Technologie web	Caractéristiques	Observations	Niveau de connaissance
 Java (JSP+servlets)	<ul style="list-style-type: none"> -compilé et interprété ; -orienté objet et autres paradigmes ; -ramasse-miettes ; -machine virtuelle ; -syntaxe familière (proche de C et C++) ; -multiplateforme. 	<ul style="list-style-type: none"> -beaucoup de concepts à maîtriser ; -très répandu ; -robuste. 	Moyenne
 PHP	<ul style="list-style-type: none"> -interprété ; - orienté objet et autres paradigmes ; -typage dynamique faible ; -système de gestion d'exception ; -multiplateforme. 	<ul style="list-style-type: none"> -très répandu ; -code souvent mal structuré (dû au HTML et à Javascript). 	Bonne
 Ruby	<ul style="list-style-type: none"> -interprété ; - orienté objet et autres paradigmes ; -ramasse-miettes ; -système de gestion d'exception ; -possibilité de modification de classe pendant l'exécution ; -nombres entiers de taille illimitée ; -multiplateforme. 	<ul style="list-style-type: none"> -pas très répandu. 	Non

Nous utiliserons PHP. PHP 5 plus précisément. Ce choix est guidé par nos prétentions d'orientation professionnelle, mais aussi Pour sa souplesse, sa portabilité ainsi que son intégration complète du modèle objet, PHP 5 est le langage qui a retenu l'attention de l'équipe de projet.

Pour la présentation des pages, nous utiliserons l'indispensable XHTML associé à javascript et aux feuilles de style CSS 2.

II.2.1.2. Le framework

a- Qu'est-ce qu'un framework?

« Un framework est un kit 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 étant 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. »¹

b. Pourquoi utiliser un framework ?




- Un framework n'est pas une nécessité absolue, mais néanmoins il est très utile.
- Un framework est un gage de qualité, d'évolutivité et de facilité de maintenance des applications.
- Les applications développées à l'aide de framework sont conformes avec les standards du marché.

¹ Source : <http://fr.wikipedia.org/wiki/Framework>

c. Quel framework allons-nous utiliser ?

Plusieurs frameworks PHP ont été mis au point. Nous avons choisi trois parmi les plus connus pour notre comparatif comme l'illustre le tableau 8.

Tableau 8 : Tableau comparatif de frameworks PHP

Framework PHP	Les plus	Les moins
 CakePHP CakePHP 1.3	<ul style="list-style-type: none"> - Implémentation du pattern MVC ; - Intégration de bibliothèques AJAX (Prototype, script.aculo.us) ; - Licence Open Source. 	<ul style="list-style-type: none"> - Ne supporte pas PDO (PHP Data Object).
 Symfony Symfony 2	<ul style="list-style-type: none"> - Configuration en mode graphique ; - Implémentation du pattern MVC ; - Moteur de templates (twig) ; - Licence Open Source. 	<ul style="list-style-type: none"> - Difficile à prendre en main ; - Pas de composant pour les ACL et l'authentification.
 Zend Framework Zend Framework 1.12	<ul style="list-style-type: none"> - Composants pour les ACL, l'authentification et les sessions ; - Intégration de bibliothèques AJAX (jQuery, Dojo) ; - Implémentation du pattern MVC ; - Licence Open Source. 	<ul style="list-style-type: none"> - Documentation de l'API pas bien détaillée.

Nous utiliserons Zend Framework pour ses composants tels que les ACL, l'authentification et les sessions ; et la bibliothèque AJAX jQuery que nous connaissons bien.

NB : Pour plus d'information sur zend framework, voir annexe 1 à la page 58.

II.3. L'ARCHITECTURE LOGICIELLE DE L'APPLICATION

L'architecture du système est guidée par le framework de développement bien qu'il ne l'impose pas. Zend Framework implémente le pattern MVC (Model-View-Controller).

« Le design pattern Modèle-Vue-Contrôleur (MVC) est un pattern architectural qui sépare les données (le modèle), l'interface homme-machine (la vue) et la logique de contrôle (le contrôleur).

Ce modèle de conception impose donc une séparation en 3 couches :

Le modèle : Il représente les données de l'application. Il définit aussi l'interaction avec la base de données et le traitement de ces données.

La vue : Elle représente l'interface utilisateur, ce avec quoi il interagit. Elle n'effectue aucun traitement, elle se contente simplement d'afficher les données que lui fournit le modèle. Il peut tout à fait y avoir plusieurs vues qui présentent les données d'un même modèle.

Le contrôleur : Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.




La synchronisation entre la vue et le modèle se passe avec le pattern Observer. Il permet de générer des événements lors d'une modification du modèle et d'indiquer à la vue qu'il faut se mettre à jour. »²

II.4. L'ENVIRONNEMENT DE DEVELOPPEMENT (IDE)

Un Environnement de Développement Intégré (EDI) est un programme regroupant un ensemble d'outils pour le développement de logiciels. En règle générale, un EDI regroupe un éditeur de texte, un compilateur, des outils automatiques de fabrication, et souvent un débogueur.

Plusieurs environnements de développement permettent le développement web. Le tableau 9 nous fait une petite comparaison de quelques 'un d'entre eux.

Tableau 9 : Tableau comparatif d'EDI

EDI	Support Zend Framework	Outil de débogage	Outil Tests Unitaires	Licence/prix
 Zend Studio 9.0	Oui	Oui	Oui	Propriétaire/ payant
 Netbeans 7.2	Oui	Oui	Oui	Open source/ gratuit
 Eclipse for PHP Developers 3.0.2	Non	Oui	Oui	Open source/ gratuit

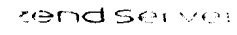
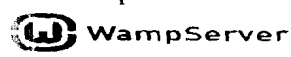

² Source: <http://julien-pauli.developpez.com/tutoriels/php/mvc-controleur/>

Ce comparatif montre qu'Eclipse ne supporte pas Zend Framework, il n'est donc pas adapté. Les deux autres EDI offrent les mêmes fonctionnalités : cependant Netbeans en raison de sa simplicité d'utilisation, de sa convivialité, de sa rapidité et de sa gratuité, notre choix portera donc sur Netbeans 7.2.

II.5. UNE PLATEFORME PHP

Une plateforme PHP est un ensemble de logiciels permettant le développement, les tests et le déploiement d'applications web PHP. C'est un ensemble constitué en général du serveur web Apache, de l'interprète PHP et d'un SGBD (MySQL ou PostgreSQL). C'est cet ensemble qui sera le moteur de notre application ; grâce à lui, on pourra générer les interfaces et exécuter les différents traitements dévolus au système. En un mot, c'est le cerveau du système. Plusieurs plateformes PHP sont disponibles. En voici quelques un sous forme de tableau comparatif dans le tableau 10.

Tableau 10 : Tableau comparatif de plateformes PHP

Plateformes PHP	Composants	Les plus	Les moins
 Zend Server CE 5.6	<ul style="list-style-type: none"> -PHP 5.3; -Apache 2.2; -MySQL 5.1; -phpMyAdmin 3.3; -Zend Framework; -Zend Optimizer+; -Zend Loader; -Zend Debugger; -Zend Cache . 	<ul style="list-style-type: none"> -API de cache de données ; -Zend Framework intégré ; -Accélération de code ; -Interface de débogage ; -Connectivité base de données intégrée (Oracle, PostgreSQL...); -Multiplateforme ; -Gratuit. 	<ul style="list-style-type: none"> -Mises à jour pas très fréquentes ; -Téléchargement d'Apache et MySQL lors de l'installation.
 WampServer 2.2	<ul style="list-style-type: none"> -PHP 5.3; -Apache 2.2; -MySQL 5.5; -phpMyAdmin 3.4; -XDebug 2.1; -sqlbuddy1.3; -webgrind1.0. 	<ul style="list-style-type: none"> -Possibilités d'ajouter des versions d'Apache, PHP, MySQL ; -Large communauté d'utilisateurs ; -Gratuit. 	<ul style="list-style-type: none"> -Plateforme Windows uniquement.
 XAMPP 1.8	<ul style="list-style-type: none"> -PHP 5.4; -Strawberry Perl 5.16; -Apache 2.4; 	<ul style="list-style-type: none"> -Inclut les dernières versions d'Apache, PHP, MySQL ; -Multiplateforme ; 	<ul style="list-style-type: none"> -Installation complexe.

-MySQL 5.5; -phpMyAdmin 3.5; -Tomcat 7.0; -OpenSSL.	-Gratuit.
--	-----------

Pour éviter tout problème lié à un changement possible de système d'exploitation, nous écartons WampServer qui est uni-plateforme. Quant aux deux autres plateformes, Zend Server est plus fourni en fonctionnalité et intègre notre framework de développement, nous opterons donc pour Zend Server CE 5.6.

NB : pour plus d'information sur MySQL, voir annexe 2 à la page 59.

II.6. OUTILS DE MODELISATION

Les outils de modélisation qui ont servi à la réalisation du projet sont les suivants:

- **Visio 2013** pour la conception des différents diagrammes UML et de l'architecture réseau insérés dans le rapport.
- **Gantt Project** pour l'établissement du diagramme de Gantt au niveau des plannings;

II.7: ANALYSE DU FUTUR SYSTEME

Nous présentons ici le scénario retenu pour le futur système.

On utilisera les graphiques suivants pour présenter les architectures réseaux correspondantes:

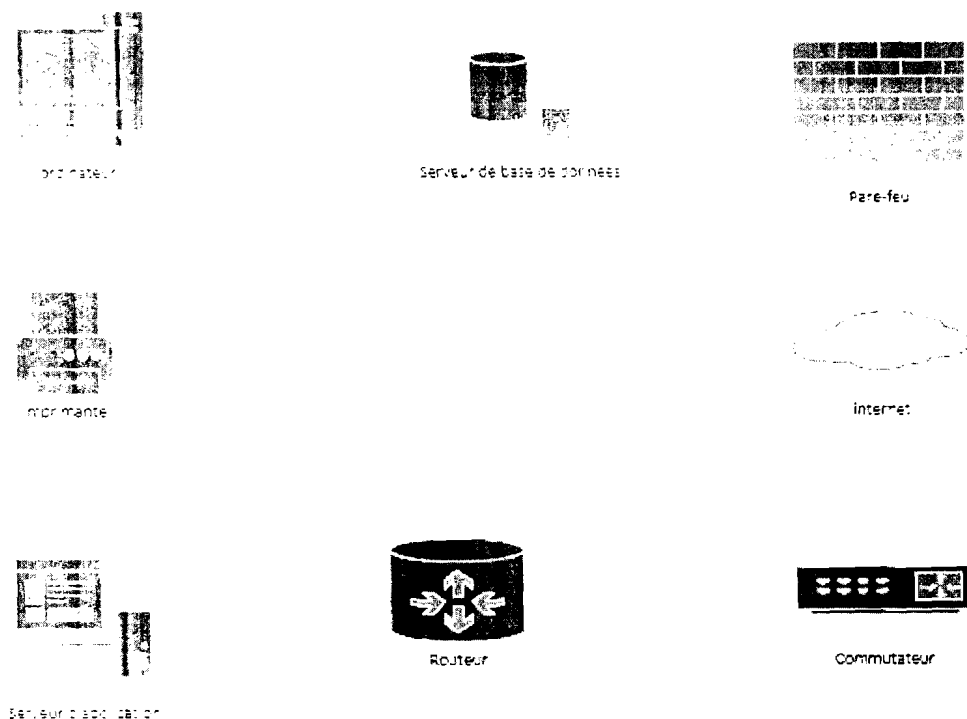


Figure 6 : Présentation du matériel

II.7.1. Description générale

Il consiste en la mise en place d'une application web qui fonctionne selon une architecture 3-tiers où l'on séparera les couches de présentation, traitement et d'accès aux données et respectant le modèle MVC préconisé par Zend Framework.

Cette application sera déployée en interne et utilisera le réseau local de l'INSD, c'est-à-dire que le serveur d'application ainsi que celui de base de données seront installés au sein de l'INSD.

Pour assurer la sécurité du réseau de l'INSD, une zone démilitarisée (DMZ, DeMilitarized Zone) est créée pour les serveurs.

Les utilisateurs accéderont à l'application grâce à un navigateur web au niveau de leur poste client (client léger) pour les différents traitements.

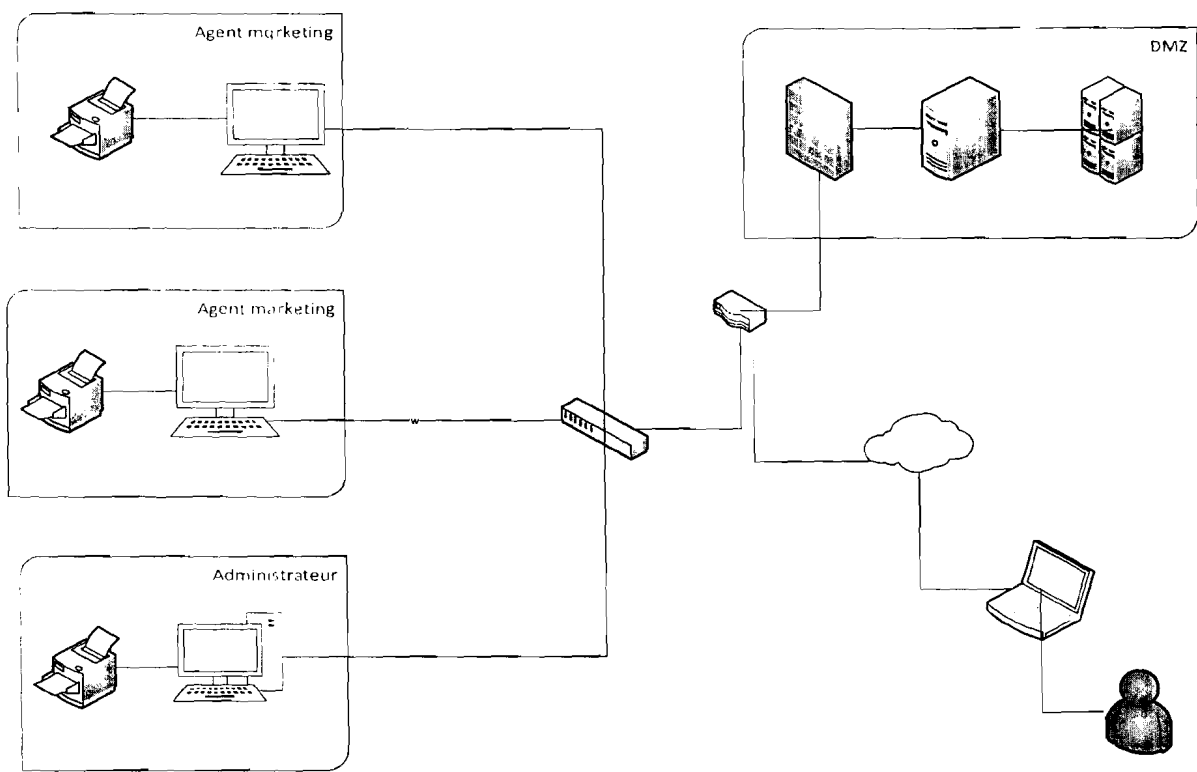


Figure 7 : Architecture réseau

II.7.2. Estimation des Besoins

Ce dont a besoin notre solution pour un fonctionnement optimal, est récapitulé dans le tableau 11.

Tableau 11 : Estimation des besoins

<i>Désignation</i>	<i>Caractéristiques</i>	<i>Prix unitaire</i>	<i>Quantité</i>	<i>Disponibilité</i>
<i>Serveur</i>	<i>Serveur HP ProLiant ML110 G7 E3-1240 3.30GHz 4-core 1P 4GB-U Hot Plug SATA 460W RPS EU</i>	<i>595 395³</i>	<i>1</i>	<i>Existant</i>
<i>Ordinateur</i>	<i>Pentium IV</i>	<i>ND</i>		<i>Existant</i>

³ Tirée sur <http://www.senetic.fr/product/626475-421>

² Tirée sur <http://store.apple.com/fr/product/TR423ZM/A/onduleur-idowell-ibox-ups-550%C2%A0va>

<i>Onduleur</i>	<i>Onduleur iDowell iBox UPS 550 Va</i>	<i>163 717.25²</i>	<i>1</i>	<i>Existant</i>
<i>Planification</i>	<i>ProjectProfessional</i>	<i>Gratuit</i>		
<i>Modélisation</i>	<i>PowerDesigner15</i>	<i>Evaluation</i>	<i>1</i>	<i>Existant</i>
<i>Système d'exploitation (SE)</i>	<i>Debian GNU/Linux 3.1 – Sarge</i>	<i>Gratuit</i>	<i>1</i>	
	<i>SE coté client</i>	<i>Windows 7</i>		<i>Existant</i>
<i>Antivirus</i>	<i>Kaspersky Internet Security 2013</i>			<i>Existant</i>
<i>SGBDR</i>	<i>MySQL 5.5.30 ou ></i>	<i>Gratuit</i>		
<i>IDE</i>	<i>Netbeans</i>	<i>Gratuit</i>		
<i>Serveur d'application</i>	<i>Apache 2.2</i>	<i>Gratuit</i>		
<i>Internet</i>	<i>Configuration professionnel, Accès ADSL débit >= 512kbt/s</i>	<i>ND</i>		<i>Existant</i>

II.7.3. Coût de développement et de formation

II.7.3.1. Développement

On distingue plusieurs méthodes permettant d'estimer le coût de développement d'un logiciel parmi lesquelles nous avons le modèle COCOMO (CONstructive COst MOdel). Cette méthode existe en trois versions : **simple, intermédiaire et détaillée.**

Nous utiliserons le modèle COCOMO simple qui est le mieux documenté, il donne des estimations des coûts en s'appuyant sur la taille (estimée) du logiciel et sur le type de logiciel ou projet à réaliser. Il existe trois (03) types de projets que sont :

➤ les projets de mode organique :

Ces projets sont réalisés par une équipe de taille relativement petite travaillant dans un environnement familier et dans un domaine d'application connu de l'équipe.

➤ les projets de mode semi-détaché :

Ce sont des types de projets qui ne sont pas trop complexes. L'équipe de développement se connaît un peu, et les technologies peuvent être mal connues, mais pas d'une grande difficulté d'appréhension.

➤ les projets de mode embarqué :

Le système à développer est une partie d'un système complexe et les modifications de spécifications destinées à contourner des problèmes logiciels sont en général impossibles.

Les formules permettant de calculer le coût, ou plus exactement l'effort requis pour le développement du logiciel en fonction du type de projet sont les suivantes :

- mode organique : $HM = 2,4 (KLSL)^{1,05}$;
- mode semi-détaché : $HM = 3 (KLSL)^{1,12}$;
- mode embarqué : $HM = 3,6 (KLSL)^{1,20}$.
- HM (signifie Homme-Mois) : représente l'effort requis pour le développement de l'application.
- KLSL (Kilo-Lignes-Sources du logiciel) : correspond à 1/1000 du nombre de lignes de code du logiciel.
- Le modèle COCOMO simple permet également d'estimer le temps nécessaire au développement d'un projet (TDEV). Les équations pour les différents types de projets sont les suivantes :
- Mode organique : $TDEV = 2,5 (HM)^{0,38}$;
- Mode semi-détaché : $TDEV = 2,5 (HM)^{0,35}$;
- Mode embarqué : $TDEV = 2,5 (HM)^{0,32}$.

Le nombre de personnes requis pour réaliser le projet dans cet intervalle de temps est donc :

$$➤ N = HM/TDEV.$$

Le coût total de réalisation est donné par :

$$➤ \text{Coût} = HM * \text{ValeurHM}.$$

Où ValeurHM représente le salaire moyen d'un informaticien. Au Burkina Faso nous estimons ce salaire à 225.000 FCFA.

Le nombre de lignes de code est estimé à 6100.

La méthode COCOMO simple en mode semi-détachée est la méthode retenue pour le calcul des coûts :

$$HM = 3 (KLSL)^{1,12},$$

$$TDEV = 2,5 (HM)^{0,35};$$

$$N = HM/TDEV \text{ et}$$

Coût = HM*ValeurHM ce qui donne.

Tableau 12 : Coût de développement

Désignation	Montant
Kilo-Lignes-Sources (KLSL)	6 100
Homme-Mois (HM)	23,57
Temps de développement	7,5
Nombre de développeur (N)	3
Valeur HM	225 000
Coût de développement	5 303 250

12.3.3. Formation

Afin que les utilisateurs puissent se familiariser avec le logiciel, une formation sera faite à leur égard dont le coût est évalué dans le tableau 13 :

Tableau 13 : Coût de formation des utilisateurs

Heures	Montant de l'heure	Utilisateurs	Montant total
10	15 000	4	150 000

II.7.3 - Coût de réalisation

En résumé, avec la méthode COCOMO, le cout de la réalisation de notre solution avec tous les contours réunis s'élève à 5 453 250 FCFA comme le montre le tableau 14.

Tableau 14 : Coût de réalisation

Désignation	Montant
Besoins	0
Analyse et conception	1 500 000 FCFA
Développement	5 303 250 FCFA
Formation	150 000 FCFA
Total	6 953 250 FCFA

CONCLUSION

A travers les études fonctionnelles et techniques, nous avons pu d'une part, dégager les grandes fonctionnalités et d'autre part l'architecture physique et logicielle de notre futur système. Dans le chapitre suivant, nous aborderons l'étude conceptuelle du système qui consistera à définir la manière dont ces contraintes fonctionnelles et techniques seront implémentées.

INTRODUCTION

Après les études fonctionnelles et techniques que nous avons réalisées, nous allons maintenant passer à la phase de conception. Le processus de conception permet de passer de l'analyse à la fabrication de la solution en langage objet. C'est à cette étape du processus 2TUP que s'effectue la fusion des études fonctionnelles et techniques.

Dans ce chapitre nous présenterons un échantillon des diagrammes qui ont servi à la conception du système que sont les diagrammes de cas d'utilisation, les diagrammes de séquences, les diagrammes de classes et les diagrammes de déploiement.

III.1. CONCEPTION

III.1.1. Diagrammes de cas d'utilisation

Nous présentons dans un premier temps le diagramme de cas d'utilisation global à la figure 10, puis les diagrammes détaillés de certains cas d'utilisation que nous décrivons textuellement.

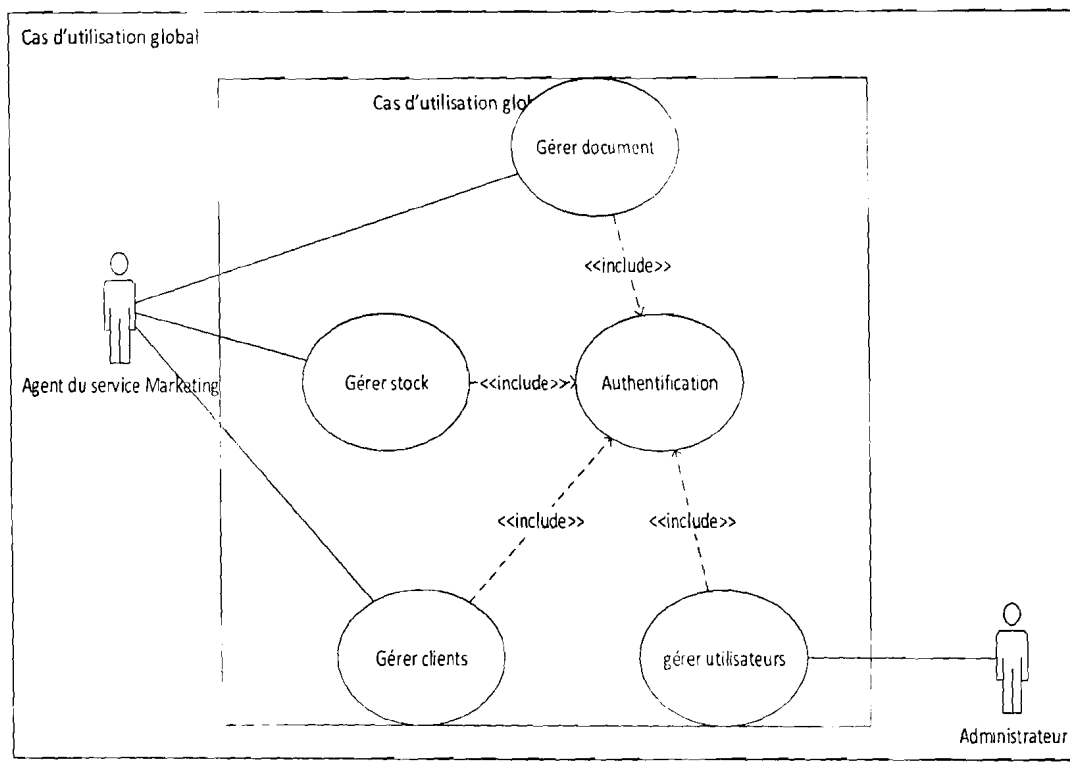


Figure 8 : Diagramme de cas d'utilisation global

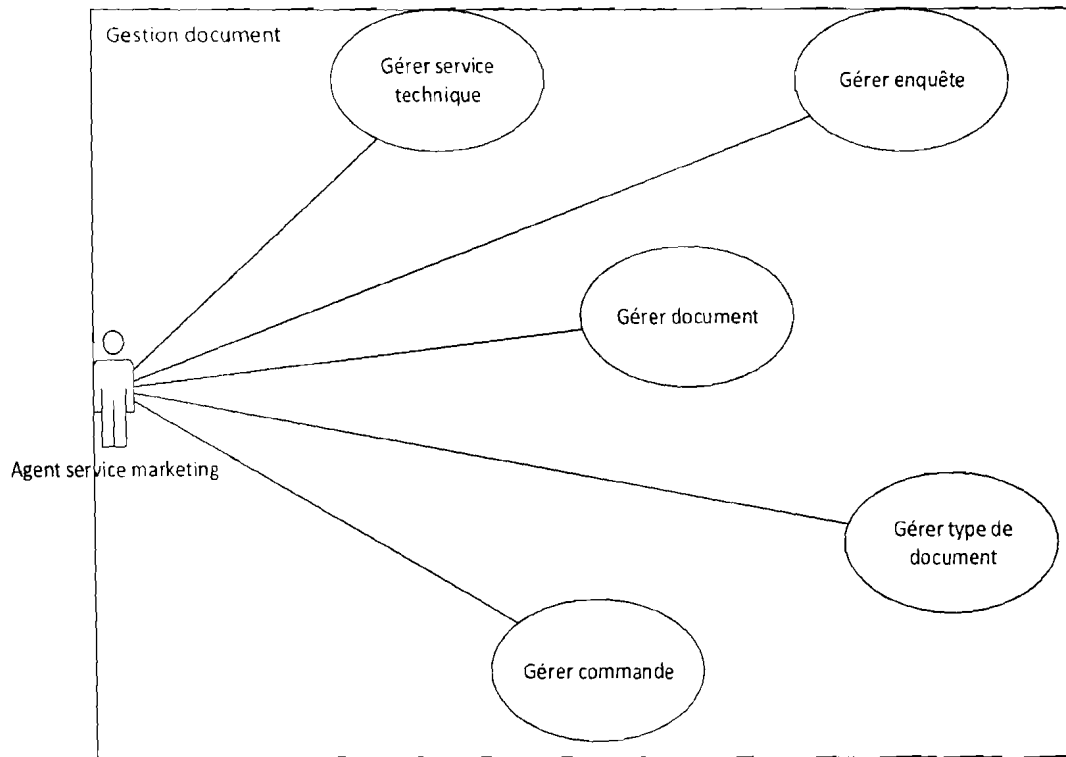


Figure 9 : Représentation détaillée du cas d'utilisation « gérer les documents »

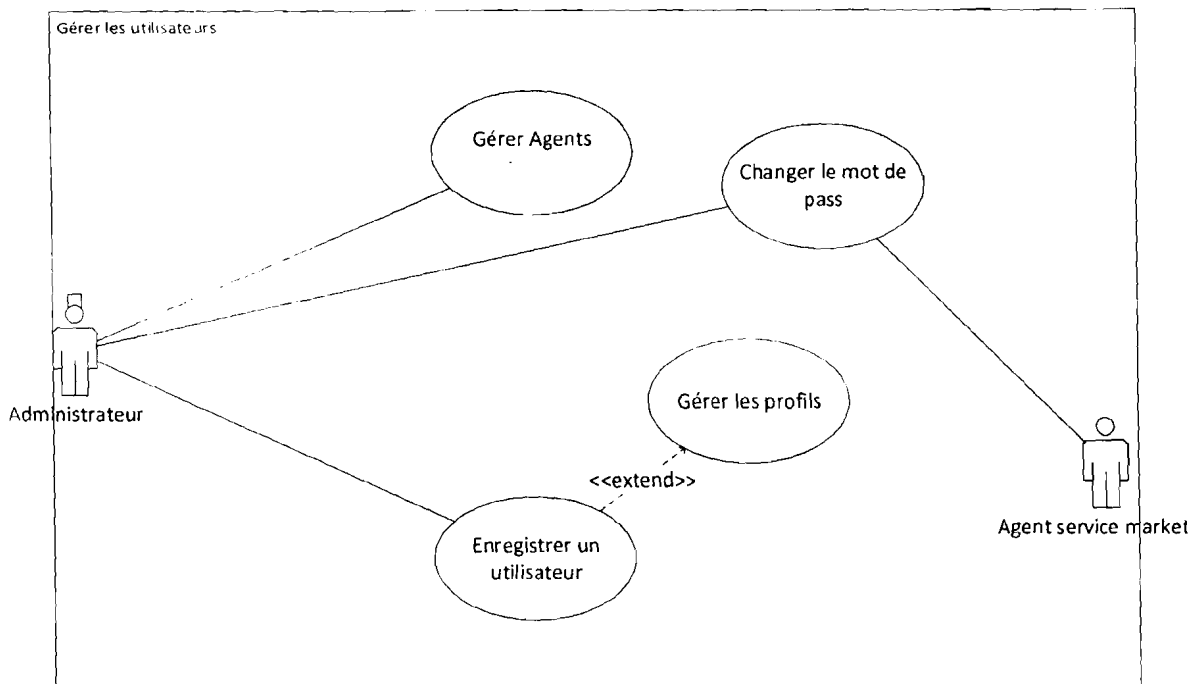


Figure 10: Représentation détaillée du cas d'utilisation «gérer les utilisateurs»

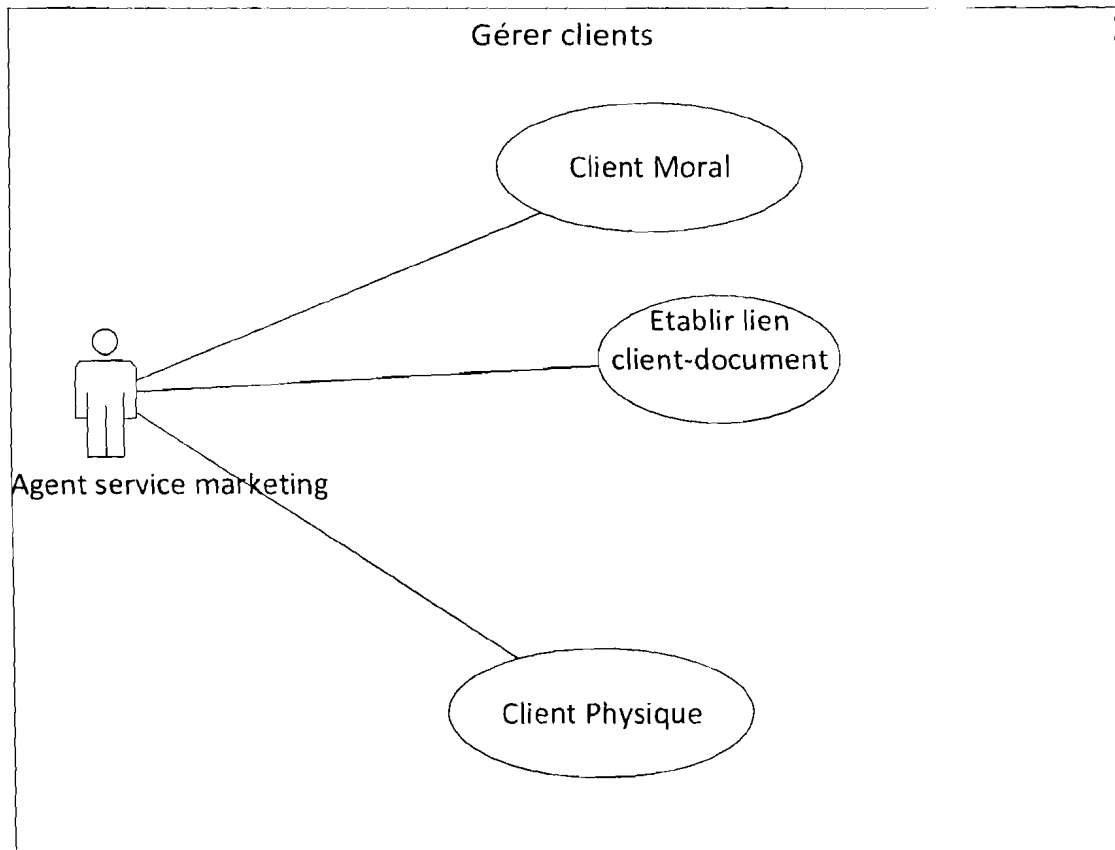


Figure 11 Représentation détaillée du cas d'utilisation «gérer les Clients»

III.1.1.1. Description textuelle de quelques cas d'utilisation

Documentation des cas d'utilisations :

Un scénario est une instance de cas d'utilisation. On distinguera dans la description des cas d'utilisation trois types de scénario à savoir :

*** Scénario nominal ("basic path")**

Il s'agit du chemin d'exécution le plus général du cas d'utilisation. C'est le chemin qui n'implique pas d'erreurs.

*** Scénario alternatif**

Un scénario alternatif est une succession d'une ou plusieurs étapes qui peuvent remplacer une ou plusieurs étapes du scénario nominal. Un scénario alternatif propose une autre possibilité de déroulement du cas d'utilisation que celle prévue dans le scénario nominal.

*** Scénario d'exception**

Une exception est une succession d'une ou plusieurs étapes qui s'ajoutent au scénario nominal en cas d'erreur(s) survenue(s) lors de la réalisation d'une étape du scénario nominal. Ces erreurs sont telles que les post-conditions ne sont pas vérifiées et entraînent l'arrêt du cas d'utilisation.

Du tableau 15 au tableau 18 nous décrivons textuellement certains cas d'utilisations comme « s'authentifier », « gérer les utilisateurs », « gérer les documents » et « gérer les clients ».

Tableau 15 : Description textuelle de quelques cas d'utilisation CU « S'authentifier »

Nom du cas d'utilisation : S'authentifier
Résumé : Ce cas d'utilisation permet à chaque utilisateur de se connecter au système en utilisant son identifiant et son mot de passe
Acteur : Tous les utilisateurs du système
Pré condition : Avoir été enregistré comme utilisateur du système
<p>Scénario nominal :</p> <p>«début»</p> <ol style="list-style-type: none"> 1. l'utilisateur lance l'application ; 2. le système lui affiche la fenêtre de connexion ; 3. l'utilisateur renseigne les informations concernant l'identificateur et le mot de passe ; 4. le système vérifie les informations fournies (E1, E2); 5. le système affiche la page d'accueil de l'utilisateur qui s'est authentifié ; <p>«fin»</p>
<p>Scénario d'exception :</p> <p>E1 : l'identificateur ou le mot de passe est incorrect.</p> <ol style="list-style-type: none"> 1. Le système informe l'utilisateur que les informations sont erronées 2. On repart au point 3 du scénario nominal. <p>E2 : un au moins des champs de saisie n'est pas renseigné.</p> <ol style="list-style-type: none"> 1. Le système informe l'utilisateur que les champs doivent être remplis 2. On repart au point 3 du premier scénario.

Tableau 16 : Description textuelle de quelques cas d'utilisation « Gérer un utilisateur »

<p>Nom du cas d'utilisation : Gérer un utilisateur</p>	
<p>Résumé : Ce cas d'utilisation permet de saisir un utilisateur avec la possibilité de mettre à jour les informations le concernant</p>	
<p>Acteur : Administrateurs du système</p>	
<p>Pré condition : Avoir été enregistré comme utilisateur du système avec un profil « administrateur »</p>	
<p>Scénario nominal :</p> <p>« début »</p> <ol style="list-style-type: none"> 1. L'administrateur lance la fenêtre d'ajout et de mis à jour des utilisateurs; 2. le système lui affiche la fenêtre (A1, A2); 3. L'administrateur renseigne les informations (E1, E2); 4. le système vérifie les informations fournies; 5. L'administrateur confirme la validation ; 6. le système enregistre le nouvel utilisateur; <p>« fin »</p>	
<p>Scénario alternatif :</p> <p>A1 : L'administrateur décide de mettre à jour un utilisateur</p> <ol style="list-style-type: none"> 1. Il sélectionne l'utilisateur en question; 2. Le système lui affiche les anciennes données de l'utilisateur; 3. Il effectue les différentes modifications; 4. Il demande à valider ses modifications; 5. Le système prend en compte les modifications ; 6. Fin du cas d'utilisation. <p>A2 : L'administrateur décide de supprimer un utilisateur</p> <ol style="list-style-type: none"> 1. Il sélectionne l'utilisateur en question; 2. Le système lui affiche une boîte de dialogue pour lui demander de confirmer la suppression de l'utilisateur; 3. Il confirme l'opération ; 4. Le système supprime l'utilisateur ; 5. Fin du cas d'utilisation. 	
<p>Scénario d'exception :</p> <p>E1 : un au moins des champs de saisie obligatoires n'est pas renseigné.</p>	

1. Le système informe l'utilisateur que les champs doivent être remplis
 2. On repart au point 4 du scénario nominal.
- E2 : les données sont erronées.**
1. Le système informe l'utilisateur que les données ne sont pas correctes ;
 2. On repart au point 4 du scénario nominal ;

Tableau 17 : Description textuelle de quelques cas d'utilisation « Gérer document »

Nom du cas d'utilisation : Gérer document	
Résumé : ce cas d'utilisation permet d'avoir des informations complètes et à jour sur un document donné. C'est-à-dire le service technique et l'enquête qui ont aboutis à son élaboration.	
Acteurs : Agent du service marketing	
Pré-condition : Avoir été enregistré comme utilisateur du système	
Scénario nominal : « début » <ol style="list-style-type: none"> 1. L'utilisateur lance la fenêtre de saisie des documents ; 2. Le système lui affiche la fenêtre ; 3. Il renseigne les informations (A1, A2, E2,E2) 4. Il valide la saisie 5. Le système contrôle et rassure que tous les champs ont été bien remplis 6. Le système enregistre les données « fin »	
Scénario alternatif : A1 : L'utilisateur décide de mettre à jour un document <ol style="list-style-type: none"> 1. Il sélectionne le document en question; 2. Le système lui affiche les anciennes données du document; 3. Il effectue les différentes modifications; 4. Il demande à valider ses modifications; 5. Le système prend en compte les modifications ; Fin du cas d'utilisation. A2 : L'utilisateur décide de supprimer un document <ol style="list-style-type: none"> 1. Il sélectionne le document en question; 2. Le système lui affiche une boîte de dialogue pour lui demander de confirmer la suppression du document; 3. Il confirme l'opération ; Le système supprime le document ;	
Scénario d'exception : E1 : un au moins des champs de saisie obligatoires n'est pas renseigné. <ol style="list-style-type: none"> 3. Le système informe l'utilisateur que les champs doivent être remplis 4. On repart au point 3 du scénario nominal. E2 : les données sont erronées. <ol style="list-style-type: none"> 3. Le système informe l'utilisateur que les données ne sont pas correctes ; On repart au point 3 du scénario nominal ;	

Tableau 18 : Description textuelle de quelques cas d'utilisation « Gérer clients »

Nom de cas : Gérer client
Résumé : ce cas d'utilisation permet d'avoir et de mettre à jour les informations sur un client
Acteurs : les agents du service marketing
Pré-condition : Avoir été enregistré comme utilisateur du système
<p>Scénario nominal :</p> <p>« début »</p> <ol style="list-style-type: none"> 1. L'utilisateur lance la fenêtre gérer client; 2. Le système lui affiche la fenêtre ; 3. Il sélectionne ajouter un nouveau client 4. Il renseigne les informations (A1, A2, E2, E2) 5. Il valide la saisie 6. Le système contrôle et rassure que tous les champs ont été bien remplis 7. Le système enregistre les données <p>« fin »</p>
<p>Scenario alternatif :</p> <p>A1 : L'utilisateur décide de mettre à jour un client</p> <ol style="list-style-type: none"> 1. Il sélectionne le client en question; 2. Le système lui affiche les anciennes données du client; 3. Il effectue les différentes modifications; 4. Il demande à valider ses modifications; 5. Le système prend en compte les modifications ; 6. Fin du cas d'utilisation. <p>A2 : L'utilisateur décide de supprimer un client</p> <ol style="list-style-type: none"> 1. Il sélectionne le client en question; 2. Le système lui affiche une boîte de dialogue pour lui demander de confirmer la suppression du client; 3. Il confirme l'opération; 4. Le système supprime le client ;
<p>Scénario d'exception :</p> <p>E1 : un au moins des champs de saisie obligatoires n'est pas renseigné.</p> <ol style="list-style-type: none"> 5. Le système informe l'utilisateur que les champs doivent être remplis 6. On repart au point 4 du scénario nominal. <p>E2 : les données sont erronées.</p> <ol style="list-style-type: none"> 7. Le système informe l'utilisateur que les données ne sont pas correctes ; 8. On repart au point 4 du scénario nominal ;

NB : Ai devant une activité indique l'existence d'un scénario alternatif numéro i et Ei devant une activité indique l'existence d'un scénario exceptionnel numéro i.

La description textuelle de ces cas d'utilisation sera accompagnée par le diagramme de séquence qui montrera de façon séquentielle le flux de message échangé entre l'utilisateur et le système.

III.1.2. Diagramme de séquence

Les diagrammes de séquences permettent de représenter des collaborations entre acteurs et objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. Le temps s'écoule "de haut en bas" de l'axe vertical du diagramme.

Les opérateurs

Ils permettent de faire des traitements particuliers en blocs.

- **L'opérateur «Loop»** est utilisé pour décrire un ensemble d'interactions qui s'exécutent en boucle. En général, une contrainte appelée garde indique le nombre de répétitions (minimum et maximum) ou bien une condition booléenne à respecter. La boucle est répétée au moins minimum fois avant qu'une éventuelle condition de garde booléenne ne soit testée. Tant que la condition est vraie, la boucle continue au plus maximum fois.
- **L'opérateur «ref»:** une référence peut être vue comme un pointeur ou un raccourci vers un autre diagramme de séquence existant. Cela équivaut à copier le contenu du diagramme de séquence pointé en lieu et place de la référence.

A travers les figures 14 à 17 nous présentons les diagrammes de séquences de certains cas d'utilisation tel « s'authentifier », « gérer document », « enregistrer une commande » et « approvisionner stock ».

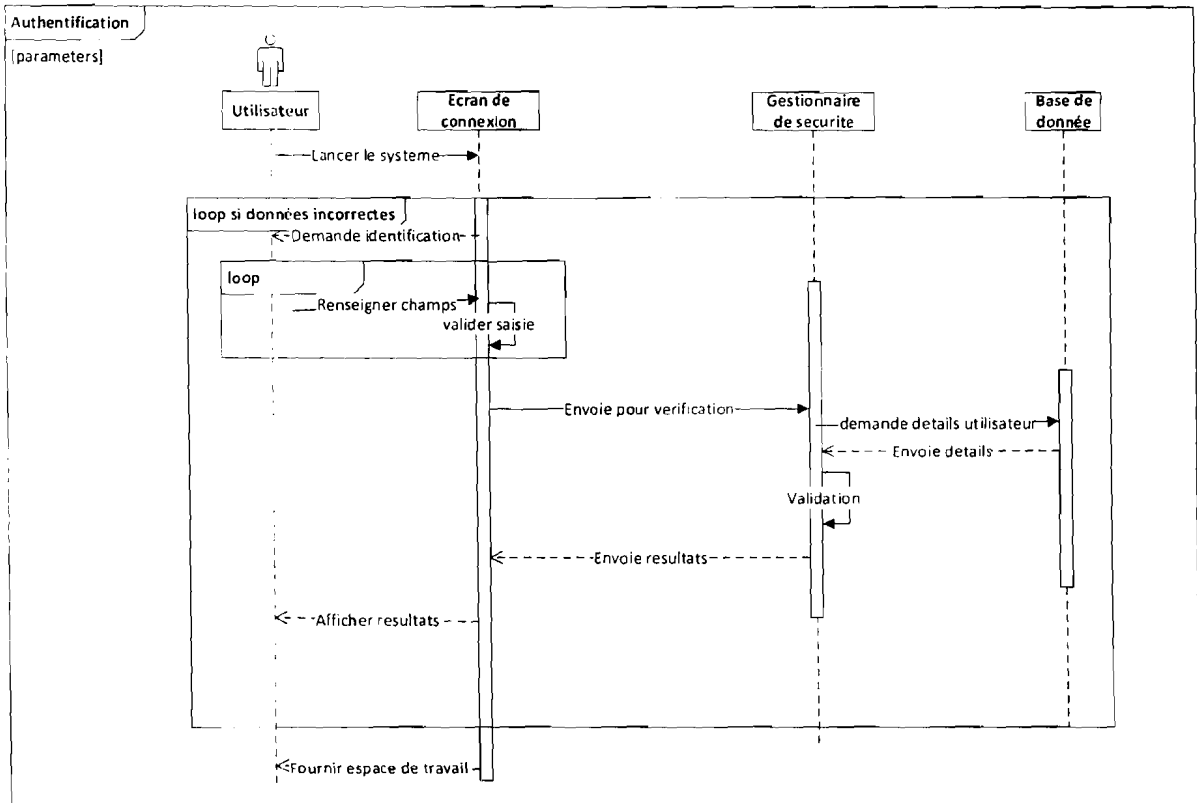


Figure 12 : Diagramme de séquence du cas d'utilisation « s'authentifier »

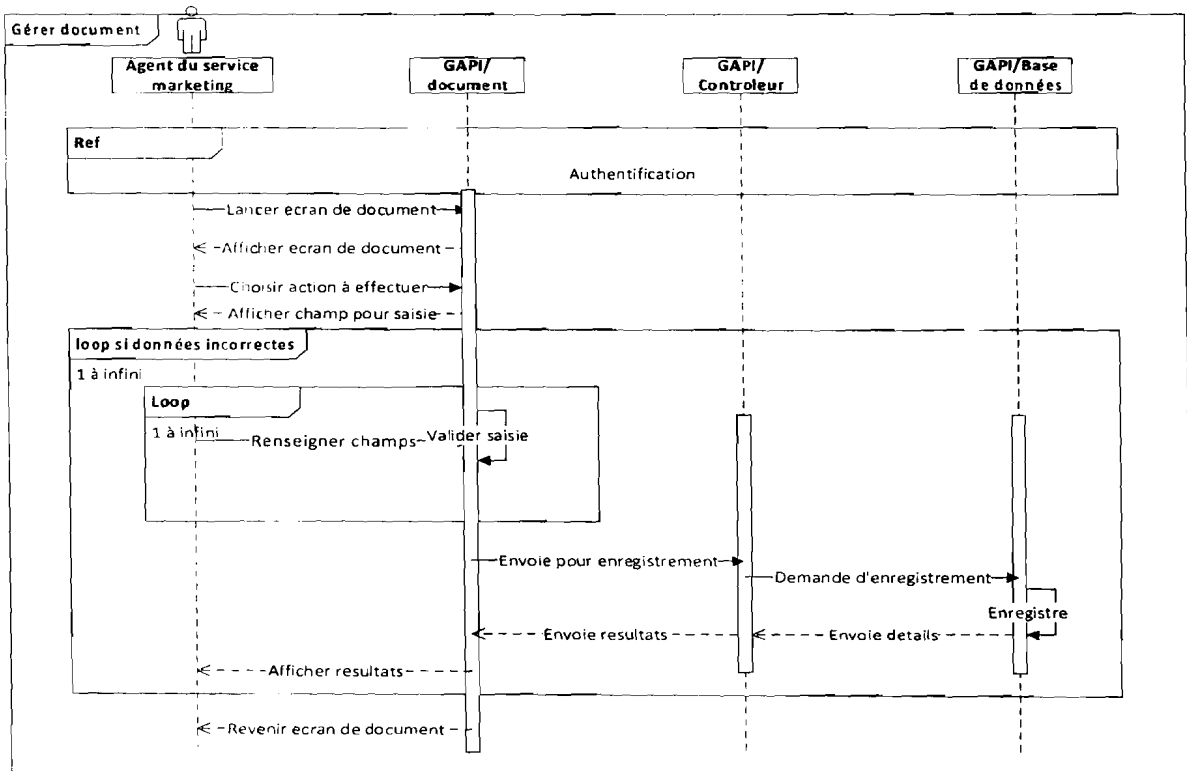


Figure 13 : Diagramme de séquence du cas d'utilisation «Gérer documents»

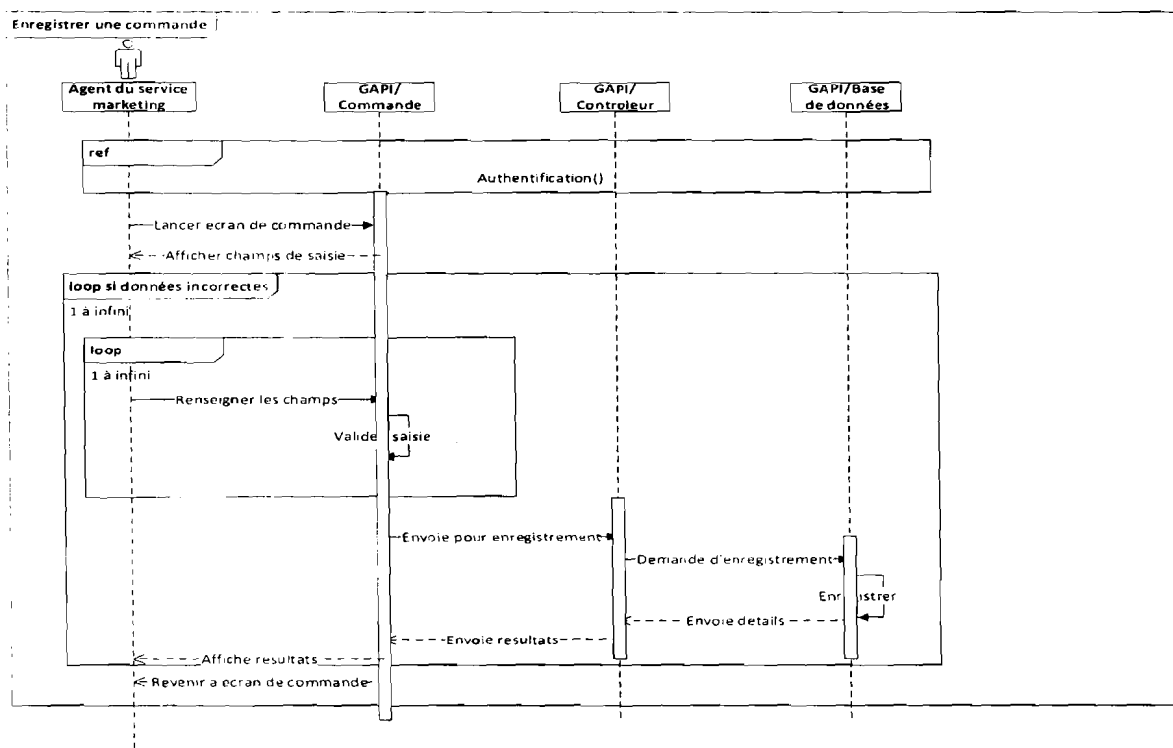


Figure 14 : Diagramme de séquence du cas d'utilisation «Enregistrer une commande»

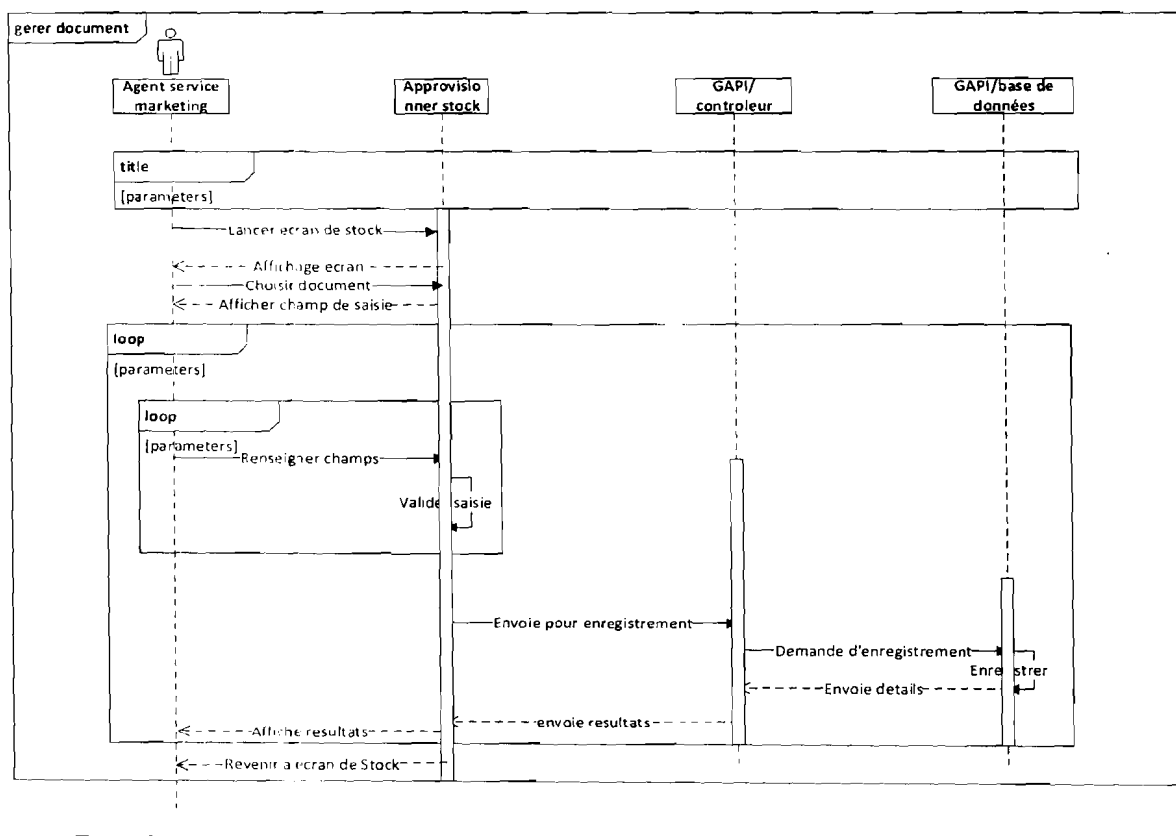


Figure 15 : Diagramme de séquence du cas d'utilisation «Approvisionner stock»

III.1.3. Diagramme de classe

III.3.1.1. Règles de gestion (RG)

Les règles de gestion suivantes, organisées par grand domaine fonctionnel expriment les relations statiques qui seront prises en compte entre les entités interagissant dans la gestion de la publication de l'INSD. Ce sont ces mêmes règles qui régissent le diagramme de classe de la figure 18.

➤ *GESTION DES DOCUMENTS*

- un document est produit par un service technique
- un service technique peut produire un ou plusieurs documents
- un service technique dirige une ou plusieurs enquêtes
- une enquête est dirigée par un service technique
- un document peut avoir plusieurs types
- un type peut concerner plusieurs documents

➤ *GESTION DES UTILISATEURS*

- un utilisateur est une personne
- un utilisateur a un profil
- un profil peut admettre plusieurs utilisateurs
- un utilisateur enregistre un ou plusieurs documents
- un document est enregistré par un utilisateur

➤ *GESTION DES CLIENTS*

- un client est soit moral ou physique
 - un client lance une ou plusieurs commandes
 - une commande concerne un et un seul client
 - une commande concerne un ou plusieurs documents
 - un document peut faire l'objet de plusieurs commandes
- un document peut faire l'objet de plusieurs commandes

III.3.1.2. Schéma du diagramme de classe

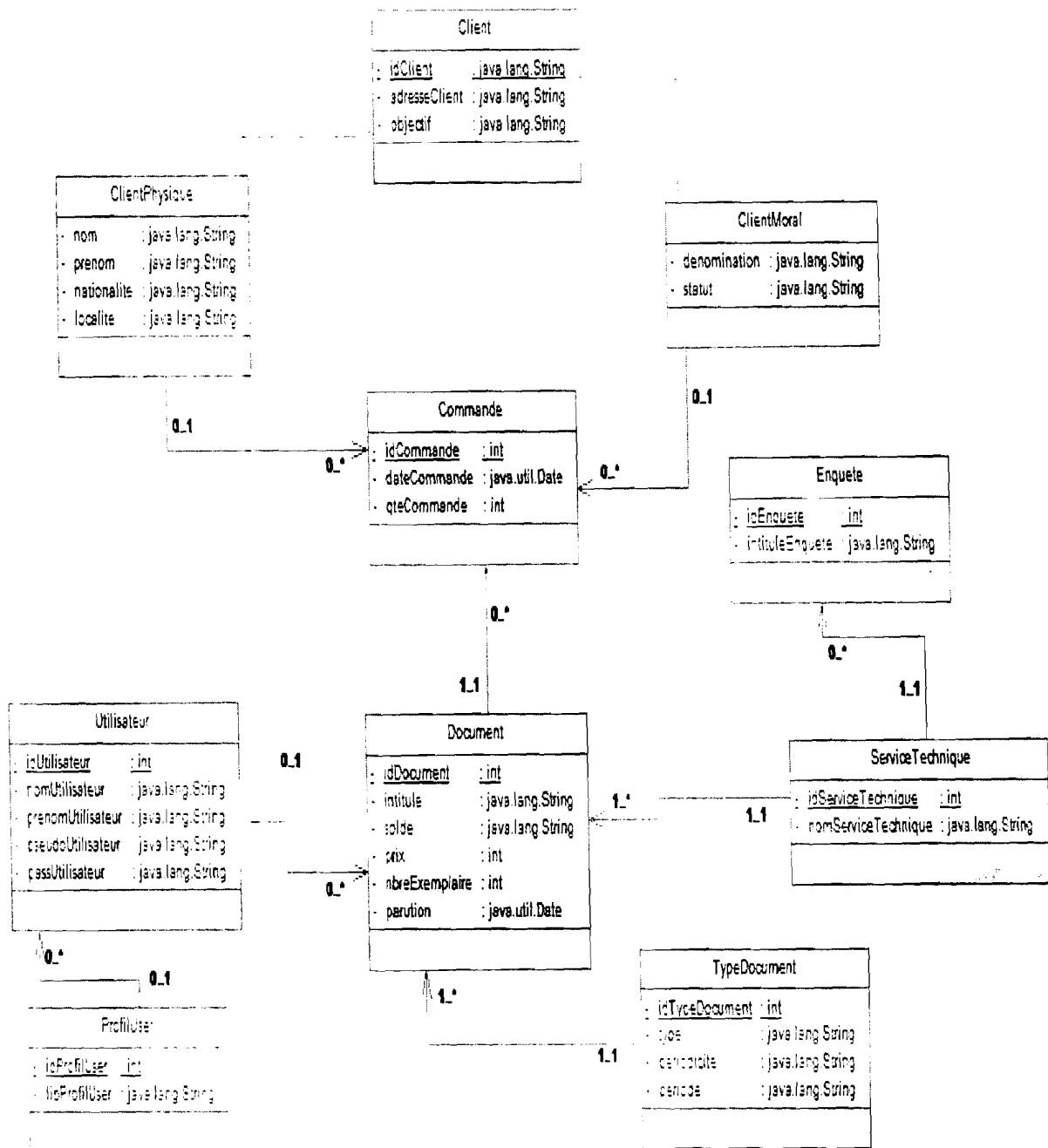


Figure 16 : Diagramme de classes

III.1.4. Diagramme de déploiement

Les diagrammes de déploiement UML2 sont utilisés pour représenter l'architecture physique d'un système en utilisant des nœuds et des connexions entre ces nœuds comme le montre la figure 19.

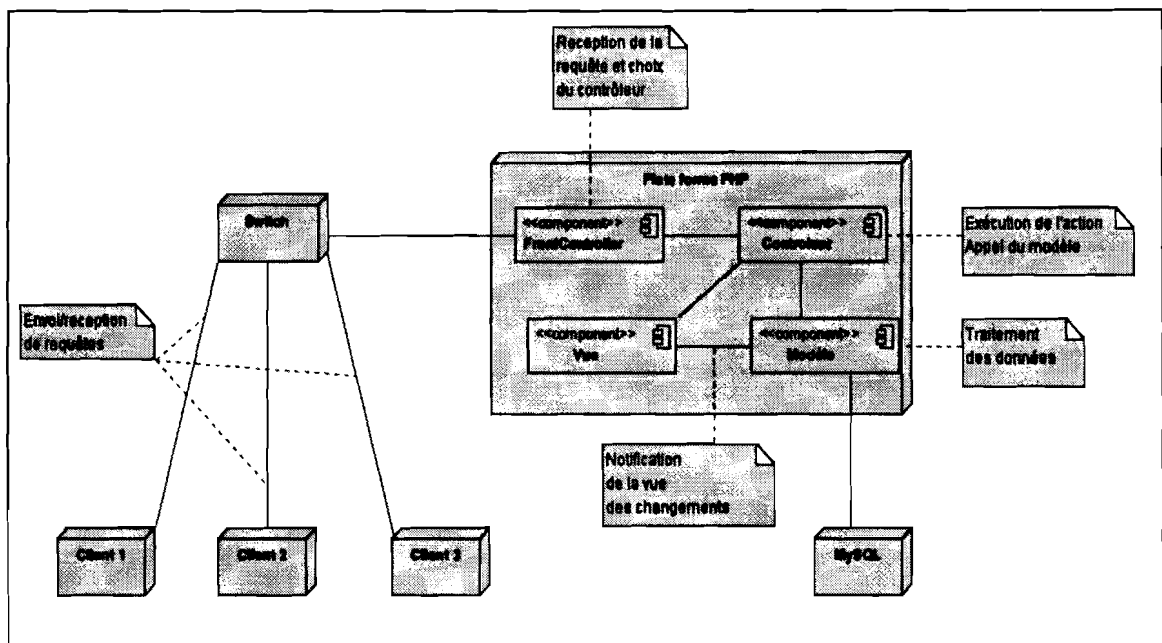


Figure 17 : Diagramme de déploiement

III.2. DEVELOPPEMENT DE GAPI

Voici la liste des différentes méthodes associées aux différentes classes de notre solution.

Tableau 19 : Liste des méthodes liées aux différentes classes

METHODES	VISIBILITE	TYPE	DESCRIPTION
getAll[nom_classe]	Public	List	Sélection de tous les champs d'une classe
getAll[nom_attribut]	Public	List	Sélection de toutes les lignes d'un attribut d'une classe
set[nom_attribut]	Public	Void	Assigne une valeur à un attribut
get[nom_attribut]	Public	Type de l'attribut	Récupère la valeur d'un attribut
del[nom_classe]	Public	String	Supprime une ligne d'une classe
save[nom_classe]	Public	String	Enregistre des données dans une classe
update[nom_classe]	Public	String	Met à jour les données d'une classe

III.2.1 Arborescence du projet zend framework dans NetBeans

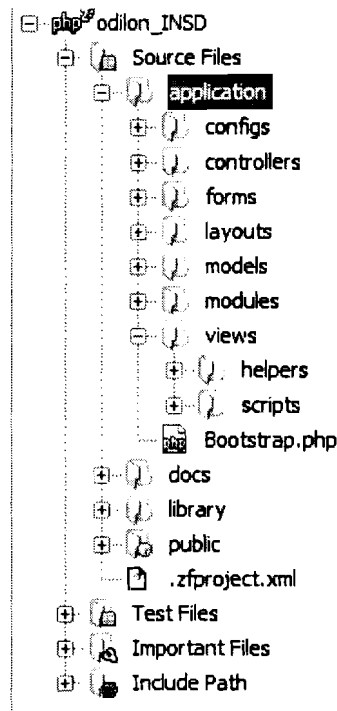


Figure 18 : Arborescence du projet zend framework

Dans un projet NetBeans, le code de l'application se trouve dans le dossier « Sources Files », les fichiers de test dans « Test Files ». Nous nous intéressons au dossier le plus important : « Sources Files », il est organisé en sous-dossiers selon l'arborescence recommandée par zend framework.

➤ application

- ✓ le fichier ***Bootstrap.php*** contient le code de chargement automatique des ressources de l'application au démarrage. Le chargement automatique permet d'initialiser l'application et de charger certains composants du framework ;
- ✓ ***configs*** contient le fichier *application.ini* qui stocke les données de configuration globale de l'application telles que les paramètres de connexion à la base de données, le fuseau horaire, le doctype html, ... ;
- ✓ ***controllers*** contient les fichiers correspondants aux différents contrôleurs ;
- ✓ ***models*** stocke les fichiers correspondants aux modèles ;
- ✓ ***views*** stocke les vues ;

- ✓ *layouts* contient des scripts pour la gestion de l'interface graphique ; ces scripts contiennent le code commun à toutes les pages ;
- ✓ *modules* contient les modules de l'application ; chaque module a ses dossiers *controllers*, *models*, et *views* ; en effet, zend framework considère chaque module comme une application indépendante ; *forms* contient les formulaires utilisés dans le module. Chaque module a aussi son fichier *Bootstrap.php*
 - **library** : ce dossier contient la bibliothèque jQuery et le framework zend lui-même.
 - **Public** :
- ✓ *css* : ce dossier contient les feuilles de style ;
- ✓ *js* : c'est le dossier contenant les scripts javascript ;
- ✓ le fichier *index.php* est le point d'entrée de l'application, toutes les requêtes passent par là ;
- ✓ Le fichier *.htaccess* est utilisé par le serveur web Apache.
 - **Docs** : ce dossier est dédié à la documentation générée ou saisie.

Cette arborescence permet d'assurer une certaine sécurité quant à l'accès aux fichiers. Le serveur Apache est configuré pour n'accéder qu'au dossier *public*.

III.2.2 Quelques écrans de GPF

➤ Connexion

Pour pouvoir accéder à la page d'accueil de l'application, l'utilisateur doit au préalable s'identifier. Cette identification consiste en la saisie de son identifiant et de son mot de passe.

Tant que l'authentification n'est pas réussie, l'utilisateur ne peut pas accéder à son espace de travail. Un message vous avise dans ce cas de l'échec de votre authentification.



Figure 19 Ecran de connexion avec échec d'authentification

Une fois l'authentification réussie, l'utilisateur accède à sa page d'accueil.

➔ La page d'accueil

C'est le cadre de travail quotidien des utilisateurs. Il comprend l'ensemble des tâches qu'un utilisateur peut effectuer dans le processus de la Gestion automatisée des publications de l'INSD. Tous les utilisateurs n'ont pas les mêmes droits. Ainsi un utilisateur administrateur possède les droits de gestion des utilisateurs alors qu'un non-administrateur n'en possède pas. Sur la partie supérieure de la page d'accueil sont affichées les informations concernant l'utilisateur connecté.



Figure 20: Page d'accueil

➤ L'explorateur

L'explorateur permet à l'utilisateur connecté de naviguer à travers les différentes fonctionnalités que lui offre son profil.

La liste des documents est affichée lorsque l'on clique sur le menu « Gestion documents ». Les opérations de tri, d'ajout, de modification, de suppression et d'affichage détaillé des informations d'un document sont offertes.



Tous les documents Ajouter un Document

Afficher: 10 lignes /page Recherche

Numero	Catégorie/Série	Nom du document	solde	Prix	Nbr exples	Date parution	Actions
1	Resultats	Projection demographique	oui	100	1750	2014-05-29	✖ ✖ ✖
2	Resultats	Projection demographique	oui	100	1750	2014-02-15	✖ ✖ ✖
3	Resultats	Rapports d'analyse	oui	200	1750	2014-05-05	✖ ✖ ✖
4	annuaire	statistique	oui	100	1750	2014-07-14	✖ ✖ ✖
5	annuaire	qtatiqtique	oui	500	10000	2014-07-08	✖ ✖ ✖

Affichage : 1 - 5 / 5 ligne(s)

Figure 21: Ecran de gestion des documents

III.3. DEPLOIEMENT DE GAPI

III.3.1. Politique de sécurité

La sécurité d'une application doit être prise en compte à tous les stades de sa vie. Elle commence au moment de la conception, quand on met en place les choix stratégiques; elle est constante, et enfin elle se concrétise par la surveillance des opérations journalières lorsque l'application sera mise en production.

La politique de sécurité a pour but de minimiser les risques de panne, d'éviter que la base de données soit dans un état d'incohérence, d'éviter les accès non autorisés à la base et d'éviter la présence de programmes indésirables dans le réseau. Il s'agit donc de prendre toutes les dispositions utiles afin de réduire au minimum les effets néfastes des pannes matérielles ou logicielles.

III.3.2. Gestion des attaques

L'attaque est le moyen par lequel une entité accède de façon subite et avec l'intention de nuire ou de prendre le contrôle d'un système. Nous avons identifié deux attaques possibles :

➤ **Les virus et autres logiciels malveillants**

Considérés comme le mal le plus répandu de la sécurité de l'information, les virus, dans leur majorité, ont pour but premier l'infection en vue d'une déstabilisation du système informatique auquel ils accèdent. Ce sont des programmes informatiques plus ou moins autonomes dans leur fonctionnement qui se propagent par les supports de stockage.

Dans notre système, la présence d'un virus provoquerait des désagréments énormes du fait de sa capacité à se propager à travers le réseau et donc une infection de tout le système si des mesures adéquates ne sont pas prises. Pour éviter ces désagréments, il sera installé sur chaque poste client, *un antivirus* en vue de permettre un contrôle beaucoup plus rapide des informations que les acteurs du système auront à traiter.

➤ **Les accès non autorisés**

Les accès non autorisés ou accès malveillants représentent des attaques qui touchent à la confidentialité et à la sécurité des données. Les attaques d'accès malveillants prennent diverses formes selon que l'information est stockée sur un support physique (clé USB, disque dur, CD-ROM) ou électronique (réseau). Ces attaques peuvent donc être réalisées grâce à l'accès physique de l'agresseur dans le local où se trouve l'information ; mais aussi grâce à un dispositif informatique permettant d'intercepter l'information en transit sur le réseau.

La définition de profil utilisateur au moyen de l'utilisation de mot de passe et de nom de connexion permettra d'offrir à chaque utilisateur les données et traitements auxquels il a droit. Aussi les informations telles que les mots de passe de connexion ne seront stockées ni en clair ni sous forme décodable dans la base ; ces informations feront l'objet d'un cryptage qui permettra le brouillage des mots de passe.

Pour remédier aux attaques portant atteinte à la sécurité des données, l'INSD a mis en place un contrôle d'identification des personnes qui accèdent au local où se trouve le serveur.

III.3.3. Gestion des catastrophes

Le maillon faible du système est le serveur ; quant il est hors service, c'est tout le système qui ne tourne plus. On se concentre donc sur la sécurité du serveur. Les catastrophes qui constituent une menace pour le serveur sont les incendies, les inondations, la foudre. Pour cela, l'INSD dispose des moyens qui sont déjà mis en place pour lutter contre ces différentes catastrophes, à savoir des extincteurs dans les locaux, un parafoudre. Les risques d'inondation sont réduits, les bureaux des agents du service marketing étant au deuxième niveau

III.3.4. Gestion des incidents d'exploitation

En cas d'incident, l'utilisateur concerné doit faire appel à une personne qualifiée ayant des compétences en informatique pour la résolution du problème. Si l'incident est lié à l'application, il pourra éventuellement se servir des dossiers de programmation pour y parvenir. Au cas où le problème persiste, l'INSD pourra faire recours aux programmeurs.

III.4. POLITIQUE DE SECOURS

III.4.1. Plantage du système

La première mesure à prendre en cas de plantage du système est de le relancer. Si le problème persiste, il faudrait faire intervenir le groupe de projet.

III.4.2. Panne due à un accès au réseau et Autres pannes critiques

En cas de panne due à l'accès au réseau, on pourra faire appel à un technicien réseaux.

Pour toutes autres pannes, on fera intervenir une fois de plus le groupe de projet.

III.5. TRANSITION

III.5.1. Procédure transitoire

Avant la mise en place du système futur, celui-ci sera soumis à deux types de tests afin de valider sa qualité. Ces tests seront effectués par des développeurs. Il s'agit :

- ☉ **d'un test fonctionnel** : il consiste à vérifier que les résultats produits par le système sont ceux attendus ; ce test prendra en compte les scénarii nominaux alternatifs et exceptionnels des différentes fonctionnalités du système.

- **d'un test structurel** : beaucoup plus professionnel, il vise à contrôler le mode et les normes métiers de réalisation des différentes fonctionnalités.

A la suite de ces tests, viendra le déploiement du système. Pour permettre la continuité des services des départements couverts par le système, nous préconisons un fonctionnement en parallèle du nouveau système avec le système actuel pendant un mois. Cette période de couplage des deux systèmes sera mise à profit pour l'identification d'éventuelles discordances ou dysfonctionnements du système mais aussi et surtout d'apporter des corrections et des améliorations afin de fournir un produit qui réponde le mieux aux besoins des utilisateurs. Toutes les opérations de la procédure transitoire feront l'objet d'une itération jusqu'à l'obtention de 90% de la qualité externe.

III.5.2. Formation des utilisateurs

Elle permettra aux utilisateurs de se familiariser avec le nouveau système. Un système informatique n'est efficace que lorsque les différents utilisateurs prennent conscience de certains aspects sécuritaires et normes d'utilisation. Cette prise de conscience passe nécessairement par cette formation. En effet, les utilisateurs doivent être formés à bien utiliser les services du système en évitant les opérations qui pourraient le déstabiliser ou présenter des failles de sécurité et en privilégiant les opérations qui participent le mieux à son maintien et à sa sécurité.

Les différents utilisateurs auront une formation, avant toute exploitation du système, pour se familiariser avec ce nouveau système. Les critiques émises lors de la formation seront un tremplin pour améliorer la qualité du logiciel.

CONCLUSION

À ce niveau, toutes les questions relatives à l'agencement et aux détails de la solution ont été modélisées

A l'issue du codage, l'application sera évaluée via des tests avant d'être déployé et mis à la disposition des utilisateurs.

CONCLUSION GENERALE

Au terme de notre étude et de notre stage, des acquis considérables ont été engrangés. Ainsi, il apparaît clairement que l'informatisation de la gestion des publications de l'INSD sera d'un apport considérable à cette dernière.

De ce fait les documents que produit l'INSD seront suivis depuis leur production jusqu'à leur écoulement. Notre application permettra aux agents du service marketing d'avoir une vue globale à chaque instant de leurs documents à savoir les recettes, les clients qui s'y sont intéressés, le service technique qui a produit le document ainsi que le stock de ce document.

Pour que cette problématique puisse être résolue il a fallu qu'on se dote d'une démarche d'analyse robuste et assez complète pour pouvoir cerner tous les contours de notre projet.

Dans cette démarche le langage de modélisation utilisé est UML dans sa version 2.0 qui se fait compléter par le processus de développement 2TUP.

D'un traitement manuel inadapté, l'INSD passera à un traitement automatique des données et un gain considérable en temps.

Durant ces trois mois de stage nous avons aussi pu capitaliser de nombreuses connaissances en matière de programmation Web, tout en réalisant des acquis certains en matière de professionnalisme.

Il est cependant notable que certaines insuffisances sont à relever et à corriger pour parfaire cette application. Aussi le groupe de projet s'engage-t-il à poursuivre des efforts dans ce sens, tout en étant disposé à recevoir d'éventuelles contributions.

BIBLIOGRAPHIE ET WEBOGRAPHIE,

BIBLIOGRAPHIE :

- ✧ **Julien Pauli et Guillaume Poncon**, Les cahiers du programmeur Zend Framework
Bien développer en PHP
- ✧ **Jean Engels**, PHP5, Paris, Eyrolles, 2008
- ✧ **Christophe Porteneuve**, Bien développé pour le web 2.0, Paris, Eyrolles, 2008

WEBOGRAPHIE :

- ✧ <http://fr.wikipedia.org/wiki/Framework>
- ✧ <http://julien-pauli.developpez.com/tutoriels/php/mvc-controleur/>
- ✧ www.alsacreations.com/tutoriels
- ✧ <http://zend-framework.developpez.com/>
- ✧ <http://store.apple.com/fr/product/TR423ZM/A/onduleur-idowell-ibox-ups-550%C2%A0va>
- ✧ <http://http://www.senetic.fr/product/626475-421>

RAPPORTS CONSULTÉS

- ✧ NANA Séni et KOLOGO Robert: « Gestion du patrimoine foncier : cas des collectivités territoriales » rapport de stage ESI 2010-2011.
- ✧ SAGA Abdoul Karim et KIMA Franc 2011-2012 : « Informatisation de la gestion d'Elite Hôtel » rapport de stage ESI 2011-2012

Annexe 1 : Présentation de Zend Framework

Le Zend Framework est un Framework haut niveau orienté composants, pour PHP5. Il fonctionne à partir de PHP 5.1.4 et apporte un socle de solutions orienté objet pour les applications web, des plus simples au plus complexes.

Zend Framework (ZF) est un Framework open-source destiné aux développements d'applications web et de services web avec PHP5. Le Zend Framework est construit en utilisant 100% de code orienté-objet. La structure des composants du Zend Framework est quelque peu unique ; chaque composant est conçu avec de faibles dépendances envers les autres composants. Cette architecture faiblement couplée permet aux développeurs d'utiliser les composants individuellement.

Les fonctionnalités de Zend Framework

Avec à ce jour une cinquantaine de packages, Zend Framework propose un ensemble de composants, permettant entre autre

- ✓ La connexion et la gestion de bases de données,
- ✓ L'intégration d'un modèle MVC,
- ✓ Un support de l'Ajax au travers de Dojo et d'un composant de création de formulaire,
- ✓ La gestion de l'authentification,
- ✓ La gestion de la session PHP,
- ✓ Et bien plus encore...

Composants de Zend Framework

Bien qu'ils puissent être utilisés individuellement, les composants de la librairie standard de Zend Framework forment un Framework d'application web puissant et extensible quand ils sont combinés. Le ZF offre une robuste et performante implémentation du motif MVC, une

Zend Framework



Développeur	Zend Technologies  
Dernière version	1.11.4 (3 Mars 2011) [+/-]
Écrit en	PHP
Environnement	Multi-plateforme
Langue	Multilingue
Type	Framework
Licence	New BSD license
Site web	framework.zend.com  

modifier 

abstraction de base de données simple d'utilisation, et un composant de formulaire qui implémente un rendu HTML, la validation et le filtrage des données, ainsi les développeurs peuvent consolider toutes ces opérations en utilisant une interface orienté-objet facile d'utilisation. Quel que soit le besoin de votre application, vous avez toutes les chances de trouver un composant de Zend Framework qui peut être utilisé pour réduire drastiquement votre temps de développement avec une base de tests solide

Annexe 2 : Présentation de MySQL

Présentation générale

MySQL est un système de gestion de base de données (SGBD). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde¹, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

Principales caractéristiques

MySQL supporte la norme SQL2 (utilisation des `RIGHT JOIN` et `LEFT JOIN`), ce qui fait de lui un SGBD sûr, la conformité à cette norme garantissant qu'il honorera les requêtes normalisées correspondantes. Cependant, les fonctionnalités des normes SQL les plus récentes ne sont pas toutes implémentées et certaines ne respectent pas la syntaxe recommandée (la concaténation par exemple), empêchant l'interopérabilité des requêtes entre différents SGBD.

Ne supportant (sauf en utilisant des moteurs comme InnoDB) ni transactions ni intégrité automatique des tables, il n'est pas destiné aux transactions financières intensives; cependant, ses performances (parfois meilleures que celles de systèmes concurrents fermés) et son prix d'implantation nettement inférieur le font adopter dans des entreprises ou services ayant besoin d'une base de données simple et peu onéreuse à mettre en œuvre pour des applications non critiques.

