

Ministère des Enseignements Secondaire, Supérieur et de la
Recherche Scientifique (MESSRS)

Université Polytechnique de Bobo-Dioulasso (UPB)

Ecole Supérieure d'Informatique (ESI)

Cycle des Ingénieurs en Conception Informatique (CICI)



SOFTNET - BURKINA

www.softnet.bf

Services et Intégration de Réseaux

Fournisseur de Service Internet

Formation

Mémoire de fin d'étude

Présenté par

ZOUNGRANA OUSMANE

Élève ingénieur en conception informatique

THEME:

***Mise en place d'une application web pour le
suivi des prestations de SOFTNET-BURKINA***

Année académique 2004-2005

Date de soutenance: Janvier 2006

Superviseur

M. Loé SANOU

Maître de stage

M. Edmond BERE

TABLE DES MATIÈRES

LISTE DES FIGURES.....	3-
INTRODUCTION.....	4-
I. LA SOCIÉTÉ SOFTNET-BURKINA.....	5-
I.1. PRÉSENTATION.....	5-
I.2. ORGANISATION.....	5-
II. LE CONTEXTE DU STAGE.....	6-
II.1. PROBLÉMATIQUE.....	6-
II.2. LE PLANING DU STAGE.....	7-
II.3. PRÉSENTATION DES OUTILS UTILISÉS.....	8-
II.3.1. LA TECHNOLOGIE .NET.....	8-
II.3.2. IIS 6.....	10-
II.3.3. SQL Server 2000.....	10-
II.3.4. MICROSOFT Exchange 2003.....	10-
II.3.5. ISA serveur 2004.....	11-
II.4. DÉMARCHE UTILISÉE.....	12-
III. ANALYSE DES BESOINS.....	14-
III.1. UML.....	14-
III.2. LE DIAGRAMME DES CAS D'UTILISATION.....	15-
III.2.1. DÉFINITION.....	15-
III.2.2. FORMALISME.....	15-
III.2.3. REPRÉSENTATION DU DIAGRAMME DES CAS D'UTILISATION.....	16-
III.2.4. DESCRIPTION TEXTUELLE DES CAS D'UTILISATION.....	17-
III.3. LES DIAGRAMMES D'ACTIVITÉ.....	31-
III.3.1. DÉFINITION.....	31-
III.3.2. FORMALISME.....	31-
III.3.3. REPRÉSENTATION DES DIAGRAMMES D'ACTIVITÉ.....	32-
III.4. LE DIAGRAMME DE CLASSES.....	41-
III.4.1. DÉFINITION.....	41-
III.4.2. LES CONCEPTS UTILISÉS.....	41-
III.4.3. REPRÉSENTATION DU DIAGRAMME DE CLASSES.....	42-
IV. ÉTUDE CONCEPTUELLE DE LA BASE DE DONNÉES.....	44-
IV.1. LA MÉTHODE MERISE.....	44-
IV.2. LE MODÈLE CONCEPTUEL DE DONNÉES.....	45-
IV.2.1. DÉFINITION.....	45-
IV.2.2. LES CONCEPTS UTILISÉS.....	45-
IV.3. LE MODÈLE LOGIQUE DES DONNÉES.....	47-
V. LES PROCÉDURES DE SÉCURITÉ.....	49-
V.1. LA MISE EN PLACE D'UN RÉSEAU VPN.....	49-
V.2. LA ROBUSTESSE DE L'APPLICATION.....	50-
V.2.1. LES INJECTIONS DE CODES.....	51-
V.2.2. LES ATTAQUES SUR LES IDENTIFIANTS DE SESSION.....	53-
V.2.3. LA RÉAUTHENTIFICATION DES UTILISATEURS.....	53-
V.2.4. LA PROTECTION DES MOTS DE PASSE ET NOMS D'UTILISATEURS.....	53-
V.2.5. L'OBLIGATION DE LA CONNEXION.....	54-
V.3. LA SÉCURITÉ DU SERVEUR WEB.....	55-
V.4. SAUVEGARDE RÉGULIÈRE DE LA BASE DE DONNÉES.....	55-

REMERCIEMENTS

Ce travail de fin d'étude a été mené grâce à l'aide de plusieurs personnes auxquelles il nous tient à coeur de dire un grand merci.

Il s'agit notamment de:

- 📖 Monsieur ZONGO Koudibi Directeur Général de la société SOFTNET-BURKINA qui a bien voulu nous accepter dans sa structure.
- 📖 Monsieur BERE Edmond notre maître de stage.
- 📖 Tout le personnel de la société SOFTNET-BURKINA pour leur étroite collaboration.
- 📖 Monsieur Loé SANOU notre superviseur.
- 📖 Ma famille et mes amis pour leur soutien.
- 📖 Monsieur TIROGO Hamidou et ses amis pour leurs conseils et encouragements.

Nous remercions également le corps professoral de l'Ecole Supérieure d'Informatique (ESI) pour les connaissances qu'il nous a transmises tout au long de notre formation.

DEDICACES

À toute ma famille qui m'a été toujours la garante d'une existence paisible et surtout à ma chère Maman.

Ousmane

VI. LE TRAVAIL RÉALISÉ.....	- 56 -
VI.1. LE DIAGRAMME D'ENCHAÎNEMENT DES PAGES	- 56 -
VI.2. PRÉSENTATIONS DE QUELQUES PAGES	- 57 -
<i>VI.2.1. LA PAGE DE CONNEXION</i>	<i>- 57 -</i>
<i>VI.2.2. LES PAGES D'ADMINISTRATION.....</i>	<i>- 58 -</i>
<i>VI.2.3. LES PAGES DES PRODUITS.....</i>	<i>- 61 -</i>
<i>VI.2.4. LA PAGE DE VÉRIFICATION DE GARANTIE.....</i>	<i>- 62 -</i>
<i>VI.2.5. LES PAGES D'ÉTABLISSEMENT D'UNE FACTURE PROFORMA.....</i>	<i>- 63 -</i>
<i>VI.2.6. LES PAGES DE LIVRAISON DE MATÉRIEL.....</i>	<i>- 67 -</i>
<i>VI.2.7. LES PAGES DE LA MAINTENANCE DE MATÉRIELS.....</i>	<i>- 68 -</i>
<i>VI.2.8. LES PAGES DE DEMANDE DE MATÉRIEL.....</i>	<i>- 71 -</i>
<i>VI.2.9. LA PAGE DES RAPPORTS HEBDOMADAIRES</i>	<i>- 72 -</i>
CONCLUSION.....	-73-
BIBLIOGRAPHIE	-74-
ANNEXE.....	-75-

LISTE DES FIGURES

Figure I.A : Organigramme de SOFTNET-BURKINA	-5-
Figure II.A: Architecture générale de .net.....	-9-
Figure III.A : Le diagramme des cas d'utilisation.....	-16-
Figure III.B : Diagramme d'activité: Connexion.....	-32-
Figure III.C : Diagramme d'activité: Demande de matériels.....	-32-
Figure III.D : Diagramme d'activité: Validation d'une demande de matériels.....	-33-
Figure III.E: Diagramme d'activité: Etablissement d'un bon de sortie.....	-34-
Figure III.F: Diagramme d'activité: Conception de proformas.....	-35-
Figure III.G: Diagramme d'activité: Enregistrement d'une commande.....	-36-
Figure III.H: Enregistrement d'une livraison (SOFTNET-> Client).....	-37-
Figure III.I: Diagramme d'activité : Vérification de garantie.....	-38-
Figure III.J: Diagramme d'activité : Réception d'un matériel en panne.....	-38-
Figure III.K: Diagramme d'activité: Diagnostique d'un matériel en panne.....	-39-
Figure III.L: Diagramme d'activité: Intervention.....	-39-
Figure III.M: Diagramme d'activité: Restitution d'un matériel reçu pour dépannage au client.....	-40-
Figure III.G: un exemple de diagramme de classe.....	-42-
Figure IV.A: Exemple de MCD.....	-45-
Figure V.A : Une liaison VPN entre la direction technique et le siège.....	-49-
Figure V.B : Une connexion VPN à partir d'un site.....	-50-
Figure VI.A : Diagramme d'enchaînements des pages.....	-56-
Figure VI.B : La page de connexion.....	-57-
Figure VI.C: La page des paramètres.....	-58-
Figure VI.D: Écran de mise à jour du personnel.....	-59-
Figure VI.E : Écran de mise à jour des clients.....	-59-
Figure VI.F: La page des affectations sur site.....	-60-
Figure VI.G: La page d'attribution des responsabilités.....	-60-
Figure VI.H: Le catalogue des produits.....	-61-
Figure VI.I : La page de mise à jour des produits.....	-61-
Figure VI.J: La page de vérification de garantie.....	-62-
Figure VI.K: L'écran des informations sommaires d'une proforma.....	-63-
Figure VI.L: L'écran des produits constitutifs d'une proforma.....	-64-
Figure VI.M: La page d'ajout / suppression d'accessoires à un produit.....	-65-
Figure VI.N : Quelques accessoires liés à un produit.....	-66-
Figure VI.O : Les Taxes et totaux d'une proforma calculés automatiquement.....	-66-
Figure VI.P: L'écran des informations sommaires d'une livraison.....	-67-
Figure VI.Q: La liste des produits d'une livraison.....	-67-
Figure VI.R : La page de réception de matériel en panne.....	-68-
Figure VI.S: La page de diagnostique d'un matériel en panne.....	-69-
Figure VI.T: La page des interventions sur un matériel en panne.....	-70-
Figure VI.U: La liste des demandes de matériel.....	-71-
Figure VI.V: La page d'établissement et de validation d'une demande de matériel.....	-71-
Figure VI.W: la page des rapports hebdomadaires.....	-72-

INTRODUCTION

L'École Supérieure d'Informatique (ESI) forme en cinq (5) ans des ingénieurs en conception informatique. Pour évaluer les connaissances plus ou moins théoriques des enseignements et permettre une bonne intégration des étudiants sur le marché de l'emploi, un stage pratique de fin de cycle est institué.

D'une durée de six (6) mois, ce stage donne l'occasion aux étudiants de réaliser un projet de fin d'étude. Ce projet consiste en la résolution d'un problème scientifique et technique posé par la structure d'accueil de l'étudiant et faisant intervenir tout ou partie de l'ensemble des disciplines relevant des sciences pour l'ingénieur enseignées au cours de la formation.

C'est ainsi que nous avons été accueillis par la société SOFNET-BURKINA pour la mise en place d'une application web pour le suivi de ses prestations.

I. LA SOCIÉTÉ SOFTNET-BURKINA

I.1. PRÉSENTATION

SOFTNET-BURKINA SA est une société anonyme de droit burkinabé inscrite au registre du commerce sous le N° RCCM BFOUA 2003 B1798.

SOFTNET-BURKINA est une société de prestation de services et de vente de matériels informatiques. Créée en 2003, elle est devenue l'une des sociétés informatiques bénéficiant d'une confiance croissante des sociétés et administrations burkinabés. Son objectif principal est de répondre rapidement et de façon efficace aux appels de ses clients face à des besoins informatiques de plus en plus exigeants en leur fournissant des conseils et des services de haut niveau.

Ces principales activités sont :

- 📖 Consulting (Audit, rédaction de cahier des charges, conduite du changement)
- 📖 Intégration (Installation et paramétrage)
- 📖 Formation (Centre de formation technique; partenaire certifié MICROSOFT)
- 📖 Assistance (Centre de maintenance, partenaire certifié HP)
- 📖 Audit et sécurité des systèmes et réseaux

I.2. ORGANISATION

SOFTNET-BURKINA est constituée d'une direction générale, d'une direction technique, d'une direction commerciale, d'un service de comptabilité et d'un secrétariat. Son organigramme se présente comme suit :

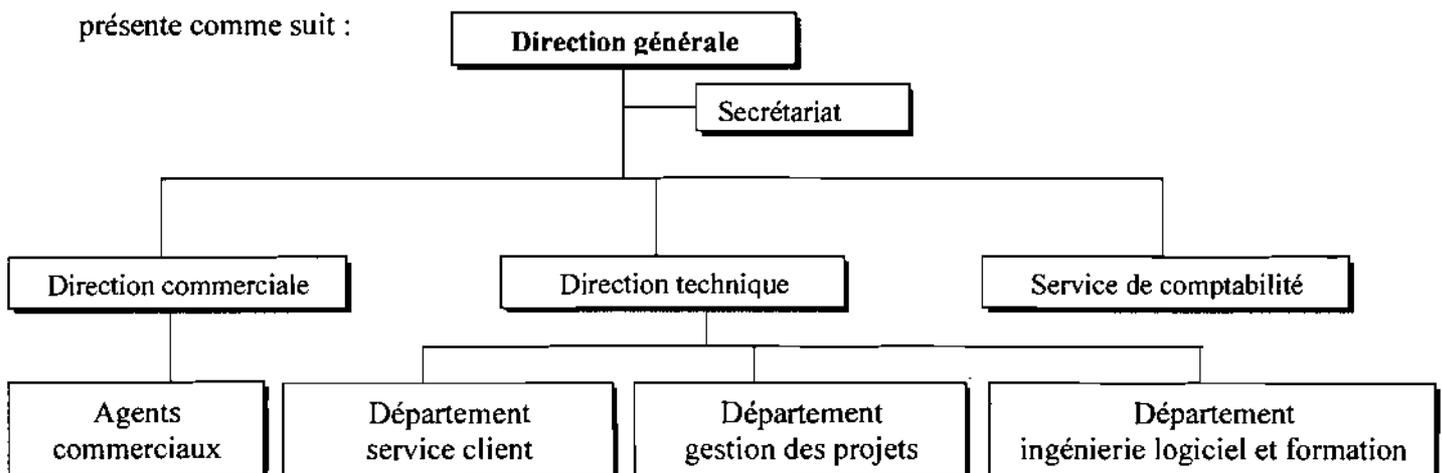


Figure I.A : Organigramme de SOFTNET-BURKINA

II. LE CONTEXTE DU STAGE

II.1. PROBLÉMATIQUE

Il est paradoxal que, tandis que SOFTNET-BURKINA solutionne les problèmes informatiques de ses clients, elle continue de souffrir d'une gestion manuelle de ses prestations. En effet, la gestion actuelle souffre d'un suivi plus ou moins manuel rendant ainsi les décisions difficiles.

Parmi tant d'autres citons quelques difficultés :

- ☐ La société livre des matériels informatiques aux clients avec une certaine période de garantie. Au cours de cette période, SOFTNET-BURKINA doit réparer gratuitement ces matériels s'ils tombent en panne ou les remplacer aux cas où la panne serait grave. Cependant, vu la quantité de matériel livrés, vu le temps de garantie offert (pouvant atteindre 3 ans pour certains produits) et vu le fait que la garantie est gérée par le service commercial, et les réparations assurées par le service technique, la gestion manuelle souffre dans la recherche de matériels sous garantie.
- ☐ Pour certaines commandes de matériels, les clients retiennent sur le montant total une caution de garantie qu'ils payent au terme de la garantie si SOFTNET-BURKINA assure ses engagements pendant la période de garantie (réparer les matériels tombés en panne ou les remplacer). Vu également le nombre de livraisons, la gestion actuelle ne permet pas de détecter et de poursuivre les cautions des garanties ayant pris terme.
- ☐ SOFTNET-BURKINA a des contrats de maintenance de parcs informatiques avec l'administration et certaines sociétés privées ou publiques. Pour honorer en permanence les contrats, des techniciens sont détachés auprès de ces sociétés. Leurs activités sont suivies par la direction technique mais la distance qui sépare ces sites avec la direction technique rend le suivi difficile.

Certains des partenaires de SOFTNET-BURKINA tels HP et Microsoft exigent une gestion informatisée des prestations de leurs partenaires en leur proposant des cahiers de charge informatiques. Pour résoudre ces problèmes liés à la gestion manuelle et répondre au cahier de charge de ses partenaires, l'instance décisionnelle de la société a décidé de procéder à une informatisation de ses différents services. Du service commercial, au service technique en passant

objet et la possibilité qu'ils ont d'utiliser les multiples classes existantes du framework .net. Ceci simplifie grandement le développement d'applications complexes.

La séparation de code et l'interface (HTML)

Les pages ASP.NET se distinguent des pages asp par l'extension .aspx de leurs fichiers. Une des grandes nouveautés d'ASP.NET est la séparation du code de programmation proprement dit et de l'interface utilisateur.

Chaque page ASP.NET est en fait scindée en deux fichiers distincts. Un premier fichier porte l'extension .aspx et ne contient que le code HTML nécessaire à l'affichage des données, ainsi que des références au fichier lui est rattaché (le "code behind"). Le second fichier porte l'extension .aspx.vb ou .aspx.cs selon que l'on développe en VB.NET ou C#. Il contient tout le code de programmation nécessaire à l'exécution de la page.

La compilation des applications

Contrairement à ASP qui était un langage interprété ASP.net est compilé.

A la première demande d'une page par un utilisateur de l'application, le code source en texte (extension .aspx) est d'abord retranscrit en MSIL (Microsoft Intermediate Languages) qui est un code portable par un compilateur appelé CLR (Common Language Runtime). Ce code sera ensuite compilé en code binaire et placé puis exécuté en mémoire. Une fois la page compilée, aux demandes suivantes on ne va plus effectuer ce processus, mais ce sera la page en mémoire qui sera transmise aux visiteurs (sauf si les données de la page changent entre temps).

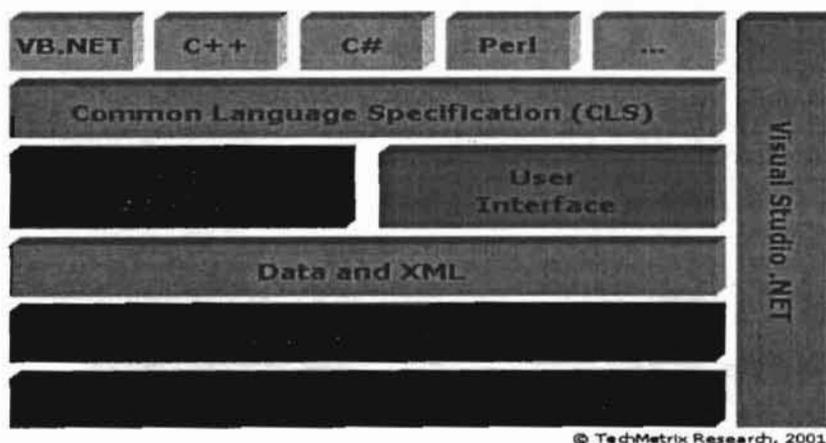


Figure II.A: Architecture générale de .net

III.2.4. DESCRIPTION TEXTUELLE DES CAS D'UTILISATION

La description textuelle permet de décrire en détails les cas d'utilisation en mettant l'accent sur les interactions entre le système et les utilisateurs dans les scénarii.

Un scénario est une instance d'un cas d'utilisation. Dans la description des cas d'utilisation, on distinguera trois types de scénario :

- le scénario nominal qui montre un déroulement normal ;
- le scénario alternatif qui est une variante du scénario nominal ;
- le scénario d'exception qui illustre un déroulement anormal du cas d'utilisation.

Cas d'utilisation : Connexion	
Résumé: Le control d'accès des utilisateurs à l'application.	
Les acteurs : Tout utilisateur du système et le système.	Version : 1.0
Préconditions : Néant	
Scénario nominal	
<u><début></u>	
<ol style="list-style-type: none"> 1- L'utilisateur demande à accéder au système. 2- Le système demande le nom d'utilisateur et le mot de passe de l'utilisateur. 3- L'utilisateur fournit son nom d'utilisateur et son mot de passe (A1). 4- Le système présente les fonctionnalités du système. 	
<u><fin></u>	
Scénario alternatif	
A1 : Nom d'utilisateur et/ou mot de passe incorrect(s).	
Démarre après le point 3 du scénario nominal.	
<ol style="list-style-type: none"> 3.1-Le système informe l'utilisateur que le mot de passe est incorrect. 	
Le système scénario au point 2 du scénario nominal.	

NB: Pour l'ensemble des cas d'utilisation suivants la connexion et une précondition triviale.

II.3. PRÉSENTATION DES OUTILS UTILISÉS

SOFTNET-BURKINA, étant partenaire de Microsoft, a décidé d'utiliser les technologies de Microsoft pour ce projet:

Outil de développement : Visual studio 2003 .net.

Serveur web: IIS 6.

SGBD: SQL serveur 2000.

Serveur de messagerie : Exchange 2003.

Firewall: ISA serveur 2004.

Il est également prévu pour ce projet d'interopérer avec le système de gestion de stock SAGE.

II.3.1. LA TECHNOLOGIE .NET

ASP.net est une nouvelle version d'ASP (Active server pages) de Microsoft. C'est une technologie d'écriture de script de serveur qui peut être utilisée pour créer des logiciels dynamiques et interactifs. Une page ASP.net est une page HTML contenant des scripts de serveur lesquels sont traités par un serveur avant d'être envoyé au navigateur de l'utilisateur.

Beaucoup plus performant qu'ASP, ASP.net est basé sur le framework .net et supporte plusieurs langages (vb.net, C#, etc.). Un module spécifiquement destiné aux applications mobiles existe également.

ASP.NET, tout comme ASP, fonctionne grâce à un ensemble de composants installés sur le serveur web, permettant l'exécution d'applications écrites dans différents langages. ASP .NET fait partie d'une infrastructure plus large, appelée framework.net (littéralement "cadre de travail») qui doit être installée sur la machine où exécutent les programmes .NET. Les principales innovations de .NET sont :

☐ L'introduction de la programmation objet

Plusieurs langages de programmation peuvent être utilisés pour créer des pages ASP.NET. Les plus utilisés sont Visual Basic .NET (dérivés du langage Visual Basic de Microsoft) et le C#, proche du C++. Une des grandes forces de ces langages de programmation est leur orientation

Les avantages de .NET :

- ☞ Le code n'est plus interprété mais compilé, il y a donc un gain de vitesse.
- ☞ Le choix parmi plusieurs langages objets pour construire les pages.
- ☞ Le code ASP.net est séparé du code HTML engendrant une clarté et une efficacité dans le développement.

II.3.2. IIS 6

IIS est le serveur web de Microsoft. La version 6 que nous devons utiliser est intégrée à Windows Server 2003. Il joue un triple rôles au sein de la plate forme .NET : l'émission de contenu web statique, la génération de contenu web dynamique via ASP.net (Active Server Pages) et il sert également de point d'entrée pour les Web Services.

II.3.3. SQL Server 2000

Microsoft SQL Server 2000 Édition Entreprise est une solution complète de base de données et d'analyse conçue pour le développement de solutions «data warehouse », d'applications métier et de commerce électronique. IL est utilisé dans ce projet pour gérer la base de données.

II.3.4. MICROSOFT Exchange 2003

Exchange 2003 est le serveur de messagerie et de collaboration de Microsoft qui permet à une entreprise de communiquer plus efficacement. Outre les fonctionnalités client enrichies fournies par Microsoft Office Outlook 2003, Exchange 2003 offre un accès à la messagerie électronique mobile, à distance et au bureau avec une sécurité et une confidentialité élevées par rapport à Echange 2000. Il assure la collaboration basée sur la messagerie électronique.

Dans ce projet il est utilisé par l'application pour envoyer des mails aux utilisateurs.

II.3.5. ISA serveur 2004

Microsoft Internet Security and Acceleration (ISA) Server 2004 est une solution avancée de pare-feu applicatif, réseau privé virtuel (VPN) et cache Web. ISA Server 2004 offre une protection avancée, une facilité d'utilisation, et un accès rapide et sécurisé pour tous les types de réseaux. Il est particulièrement adapté à la protection des réseaux exécutant des applications Microsoft, tels que Microsoft Outlook Web Access (OWA), Microsoft Internet Information Services (IIS), Office SharePoint Portal Server, le service de routage et d'accès distant (RRAS), le service d'annuaire Active Directory, etc. Il sera utilisé pour garantir la sécurité de l'application.

II.4. DÉMARCHE UTILISÉE

Nous avons opté d'utiliser une démarche de développement par incrément.

Afin de permettre aux utilisateurs de suivre la réalisation du travail, et de prendre en compte très tôt leurs préoccupations, nous avons découpé l'application en modules. Plus le module concerne leur secteur d'activité plus, les utilisateurs maîtrisent les contours et peuvent par conséquent mieux guider la réalisation. C'est ainsi que, dans la décomposition, nous avons tenu compte des domaines d'activité. Les modules que nous avons pu faire ressortir sont :

L'élaboration rapide de factures proformas, le suivi des commandes, le suivi des livraisons, le suivi des garanties, le suivi des activités de maintenance en atelier et sur les sites, l'interopérabilité avec le système de gestion de stock.

Pour chaque module nous faisons:

📖 Une analyse des besoins en procédant comme suit :

📄 Interviews

Les interviews sont des occasions au cours desquels les utilisateurs expriment leurs besoins, les difficultés rencontrées dans la gestion actuelle. Nous profitons de ces occasions pour cerner avec précision les problèmes à résoudre.

📄 Formalisation des besoins

Une fois l'étape des interviews passée, nous procédons, toujours avec les utilisateurs, à la formalisation du système décrit à travers des diagrammes UML (cas d'utilisation, séquence...).

📄 Maquettages ou prototypages rapides

Pour être sûrs d'avoir compris les préoccupations, nous réalisons des maquettes d'écran interactives sous Microsoft Access 2003. Ces maquettes sont conçues rapidement sans code d'exécution et soumis à l'appréciation des utilisateurs.

📄 Validation des prototypes

L'idée de maquettage nous évite de perdre un temps à réaliser un module inadéquat que les utilisateurs rejettent à la validation. Les maquettes conçues sont modifiées si nécessaire jusqu'à la satisfaction des utilisateurs.

Correction des diagrammes.

Si la modification des maquettes touche la description du système faite par les diagrammes établis précédemment, ces diagrammes sont corrigés.

La réalisation du module

Nous basant sur les diagrammes et les maquettes, et en tenant compte des spécificités de l'outil de développement nous procédons à la réalisation du module.

Le test du module

Le module réalisé est testé en présence des utilisateurs pour s'assurer qu'il n'y a pas d'écart entre les fonctionnalités validées au niveau maquette et celles réalisées.

L'intégration du module aux précédents

Le module est ensuite joint aux modules précédents. L'application à réaliser voit ainsi progressivement le jour.

Le test d'intégration

Nous testons à cette étape l'interaction entre le module courant et les modules précédents.

III. ANALYSE DES BESOINS

Pour avoir une même compréhension des besoins que les utilisateurs nous avons modélisé le système avec un formalisme simple qu'on a pris le soin de leur expliquer au préalable les concepts. Le langage de modélisation choisi est UML (Unified Modeling Language). Le choix d'un langage de modélisation objet se justifie aussi par le fait que les langages de programmation de .NET (C#, VB.net ...) sont orientés objet.

III.1. UML

UML est un langage de modélisation orienté objet développé en réponse à l'appel lancé par l'OMG (Object Management Group) dans le but de définir une notation standard pour la modélisation des applications construites à l'aide d'objets. Il est hérité de plusieurs autres langages tels que OMT (Object Modeling Technique), OOSE (Object Oriented Software Engineering) et Booch.

Pour formaliser un système, UML propose neuf (9) diagrammes dont 4 (quatre) pour la description structurelle (le diagramme de classes, le diagramme d'objets, le diagramme de composants et le diagramme de déploiement) et 5 (cinq) pour la description de l'aspect dynamique du système (le diagramme de cas d'utilisation, le diagramme de séquences, le diagramme de collaborations, le diagramme d'états-transitions, le diagramme d'activités)

Cependant, aucune logique de succession dans la réalisation des diagrammes n'a été imposée par le langage. Il appartient aux analystes d'adopter un cycle de vie permettant d'atteindre leurs objectifs.

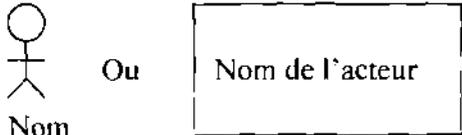
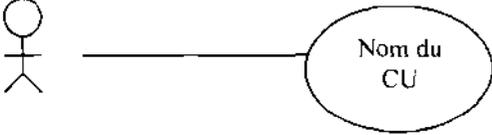
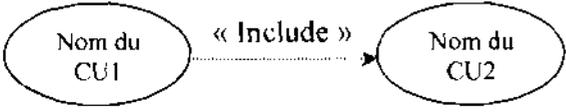
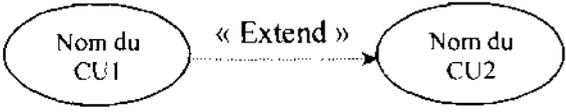
Nous utilisons le diagramme de cas d'utilisation pour l'analyse des besoins. Il permet de modéliser les acteurs et les services attendus du futur système. Sa simplicité graphique permet un dialogue facile avec les utilisateurs. Une description textuelle détaillée explique le déroulement séquentiel de chaque cas d'utilisation puis un diagramme d'activité récapitule les spécifications fonctionnelles de chaque cas d'utilisation. L'aspect structurel du système est ensuite étudié à l'aide d'un diagramme de classe.

III.2. LE DIAGRAMME DES CAS D'UTILISATION

III.2.1. DÉFINITION

Les cas d'utilisation (CU) décrivent le comportement du système du point de vue de l'utilisateur. Ils permettent de définir les limites du système et les relations entre le système et son environnement. Un cas d'utilisation est une manière spécifique d'utiliser le système. C'est l'image d'une fonctionnalité déclenchée en réponse à la stimulation d'un acteur externe. Ils permettent de centrer la construction du système sur les besoins des utilisateurs. Les diagrammes font intervenir les acteurs, les autres systèmes et les cas d'utilisation eux-mêmes.

III.2.2. FORMALISME

Présentation du symbole	Définition du symbole et du type d'information véhiculée
	<p>Un acteur : Il définit un ensemble cohérent de rôles qu'un utilisateur ou une entité peut jouer en interagissant avec le système. Il peut consulter et modifier le système en émettant ou en recevant des messages.</p>
	<p>Les cas d'utilisation (CU) : Un cas d'utilisation est une unité cohérente de fonctionnalités offertes par le système. Un CU modélise un service rendu par le système.</p>
	<p>Association Représente la participation d'un acteur à un CU.</p>
	<p>Relation d'inclusion « include » : Une relation d'inclusion du CU1 vers le CU2 indique qu'une instance du CU1 contient également le comportement spécifié dans le CU2. Ce comportement est inséré à un endroit défini dans le CU1.</p>
	<p>Relation d'extension « Extends » : La relation d'extension d'un CU1 à un CU2 indique qu'une instance du CU2 peut être augmentée (augmentation soumise à des conditions spécifiques) par le comportement défini par le CU1.</p>

III.2.3. REPRÉSENTATION DU DIAGRAMME DES CAS D'UTILISATION

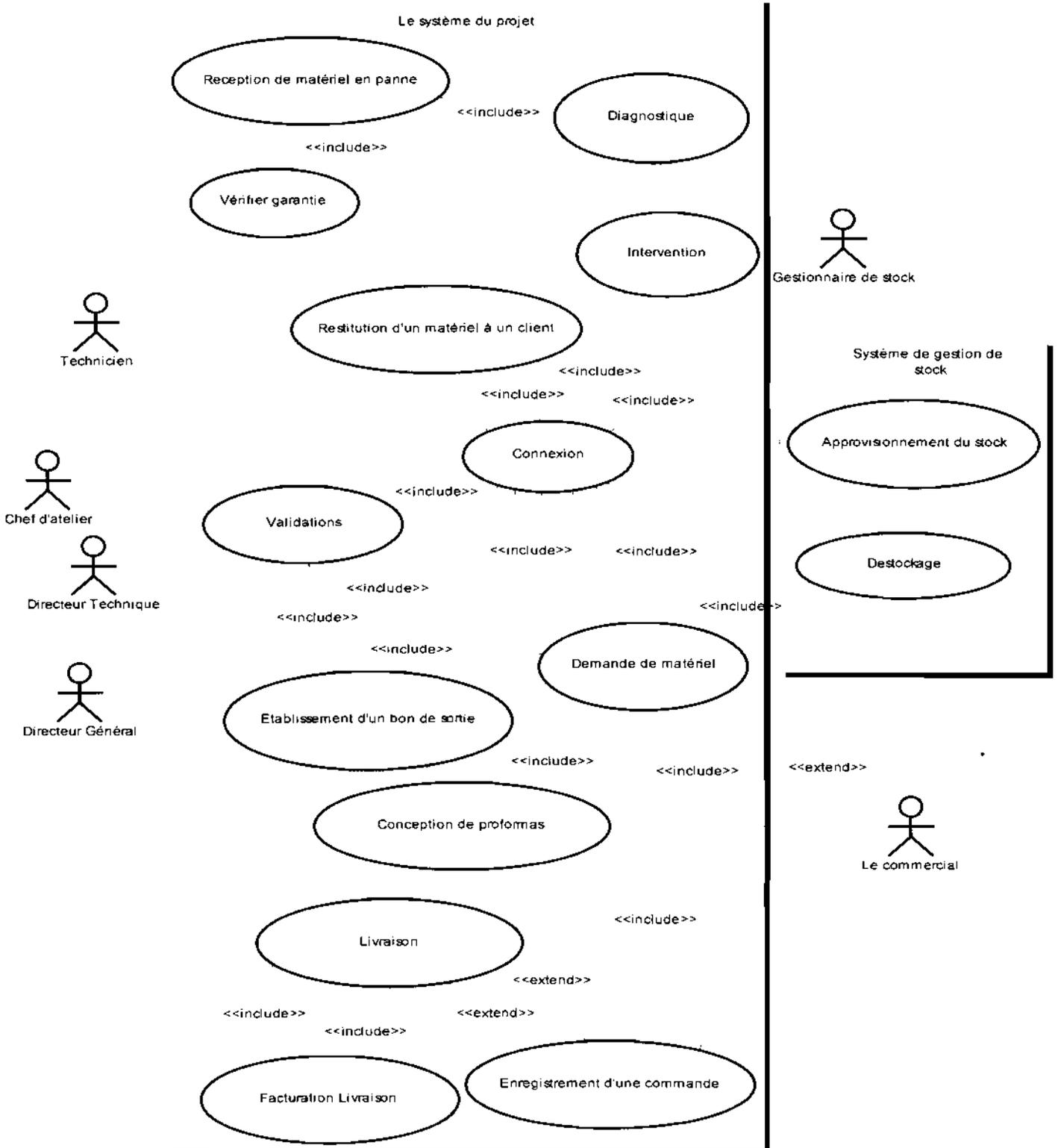


Figure III.A : Le diagramme des cas d'utilisation

III.2.4. DESCRIPTION TEXTUELLE DES CAS D'UTILISATION

La description textuelle permet de décrire en détails les cas d'utilisation en mettant l'accent sur les interactions entre le système et les utilisateurs dans les scénarii.

Un scénario est une instance d'un cas d'utilisation. Dans la description des cas d'utilisation, on distinguera trois types de scénario :

- le scénario nominal qui montre un déroulement normal ;
- le scénario alternatif qui est une variante du scénario nominal ;
- le scénario d'exception qui illustre un déroulement anormal du cas d'utilisation.

Cas d'utilisation : Connexion	
Résumé: Le control d'accès des utilisateurs à l'application.	
Les acteurs : Tout utilisateur du système et le système.	Version : 1.0
<p><u>Préconditions :</u> Néant</p> <p><u>Scénario nominal</u> <u><début></u></p> <ol style="list-style-type: none"> 1- L'utilisateur demande à accéder au système. 2- Le système demande le nom d'utilisateur et le mot de passe de l'utilisateur. 3- L'utilisateur fournit son nom d'utilisateur et son mot de passe (A1). 4- Le système présente les fonctionnalités du système. <p><u><fin></u></p> <p><u>Scénario alternatif</u> A1 : Nom d'utilisateur et/ou mot de passe incorrect(s). Démarre après le point 3 du scénario nominal.</p> <ol style="list-style-type: none"> 3.1-Le système informe l'utilisateur que le mot de passe est incorrect. <p>Le système scénario au point 2 du scénario nominal.</p>	

NB: Pour l'ensemble des cas d'utilisation suivants la connexion et une précondition triviale.

Cas d'utilisation : **Demande de matériels**

Résumé: Permet à un technicien de faire une demande de matériel depuis son site d'affectation

Les acteurs : un technicien et le système.

Version : 1.0

Préconditions:

Le technicien doit avoir les droits de chef de site dans le système.

Scénario nominal

<début>

- 1- Le technicien demande à enregistrer sa demande de matériel.
- 2- Le système demande le type, les caractéristiques du matériel et le service où il est destiné.
- 3- Le technicien fournit les informations demandées.
- 4- Le système enregistre la demande.
- 5- Le système envoie un mail au chef d'atelier pour l'informer de la demande.

<fin>

Cas d'utilisation : **Validation d'une demande de matériels**

Résumé: Permet au chef d'atelier, au directeur technique et au directeur général de valider une demande de matériels.

Les acteurs : le chef d'atelier, le directeur technique le directeur général et le système.

Version : 1.0

Préconditions:

Une demande de matériel non encore validée doit être envoyée par un technicien (point d'inclusion du cas d'utilisation « *Demande de matériels* »)

Scénario nominal

<début>

- 1- Le chef d'atelier valide la demande après s'être rassuré que le besoin est réel (E1).
- 2- Le système enregistre la validation et envoie un mail au directeur technique pour l'informer de la demande de matériel.
- 3- Le directeur technique valide la demande (E2).
- 4- Le système enregistre la validation et envoie un mail au directeur général pour l'informer de la demande de matériel.
- 5- Le directeur général valide la demande (E3).

<fin>

Cas d'utilisation : Validation d'une demande de matériels	
Résumé: Permet au chef d'atelier, au directeur technique et au directeur général de valider une demande de matériels.	
Les acteurs : Le chef d'atelier, le directeur technique, le directeur général et le système.	Version : 1.0
<p><u>Scénarii exceptionnels</u></p> <p>E1 : Le chef d'atelier ne valide pas la demande.</p> <ol style="list-style-type: none"> 1- Le chef d'atelier indique les raisons pour lesquelles la demande n'est pas validée. 2- Le système enregistre les raisons. 3- Le système envoie un mail au technicien ayant fait la demande pour l'informer que la demande n'est pas validée ainsi que les raisons. <p>Le système reprend à la balise <fin> du scénario nominal.</p> <p style="text-align: center;">*****</p> <p>E2 : Le directeur technique ne valide pas la demande.</p> <ol style="list-style-type: none"> 1- Le directeur technique indique les raisons pour lesquelles la demande n'est pas validée. 2- Le système enregistre les raisons. 3- Le système envoie un mail au technicien ayant fait la demande pour l'informer que la demande n'est pas validée ainsi que les raisons. 4- Le système envoie un mail au chef d'atelier pour l'informer que la demande n'est pas validée, ainsi que les raisons. <p>Le système reprend à la balise <fin> du scénario nominal.</p>	

Cas d'utilisation : Validation d'une demande de matériels	
Résumé: Permet au chef d'atelier, au directeur technique et au directeur général de valider une demande de matériels.	
Les acteurs : Le chef d'atelier, le directeur technique, le directeur général et le système.	Version : 1.0
<p>Scénarii exceptionnels (suite)</p> <p>E3 : Le directeur général ne valide pas la demande</p> <ol style="list-style-type: none"> 1- Le directeur général indique les raisons pour lesquelles la demande n'est pas validée. 2- Le système enregistre les raisons. 3- Le système envoie un mail au technicien ayant fait la demande pour l'informer que la demande n'est pas validée, ainsi que les raisons. 4- Le système envoie un mail au chef d'atelier pour l'informer que la demande n'est pas validée ainsi que les raisons. 5- Le système envoie un mail au directeur technique pour l'informer que la demande n'est pas validée ainsi que les raisons. 6- Si un bon de sortie a été établi et que le matériel n'est pas encore livré par le système de gestion de stock, il est alors supprimer. <p>Le scénario nominal reprend à la balise <fin> du scénario nominal.</p>	

Cas d'utilisation : Etablissement d'un bon de sortie

Résumé: L'établissement d'un bon de sortie de matériel.

Les acteurs : le directeur technique ou le directeur général et le système.

Version : 1.0

Préconditions:

Une demande de matériel validée au moins par le directeur technique et non encore livrée doit être enregistrée dans le système (point d'inclusion du cas d'utilisation « *Validation d'une demande de matériels* »).

Scénario nominal

<début>

- 1- Le système présente la liste des demandes de matériel en attente.
- 2- Le directeur technique ou le directeur général sélectionne la demande de matériel à satisfaire.
- 3- Le système présente le niveau de stock pour le matériel demandé et demande de saisir la quantité.
- 4- Le directeur technique ou le directeur général saisit la quantité (A1).
- 5- Le système enregistre le bon de sortie.
- 6- Le système fait enregistrer le bon de sortie par le système de gestion de stock

<fin>

Scénario alternatif

A1 : la quantité saisie est supérieure à quantité en stock.

Démarre après le point 4 du scénario nominal.

- 4.1- Le matériel et la quantité reliquat sont enregistrés comme matériel à commander

Le système reprend au point 5 du scénario nominal.

Cas d'utilisation : Conception de proformas

Résumé: La sélection des produits à proposer à un client et calcul automatique de prix.

Les acteurs : Le directeur commercial, le système.

Version : 1.0

Préconditions:

Le système doit contenir les produits demandés par le client.

Scénario nominal

<début>

- 1- Le commercial demande à enregistrer une proforma.
- 2- Le système demande les informations sur la proforma.
- 3- Le commercial saisit les données sommaires de la proforma (la date, le nom et l'adresse du client, la TVA (Taxe sur la Valeur Ajoutée), le BIC (Bénéfice Industriel et Commercial)).
- 4- Le système attribut un numéro automatique à la nouvelle proforma et présente la liste des produits.
- 5- Le commercial sélectionne les produits demandés par le client (A1).
- 6- Pour chaque produit sélectionné le commercial précise la quantité, le prix d'achat unitaire, la devise du prix d'achat (\$, €...), la marge bénéficiaire souhaité.
- 7- Le système calcule le prix de vente acceptable de chaque produit sélectionné.
- 8- Le commercial, s'inspirant des prix de vente calculés automatiquement, saisit les prix de vente des produits à proposer au client.
- 9- Le système calcule le Total Hors Taxe (Total HT), les taxes (TVA et BIC), le total Toutes Taxes Comprises (total TTC).
- 10- Le commercial imprime la facture proforma pour le client.

<Fin>

Cas d'utilisation : Conception de proformas**Résumé:** La sélection des produits à proposer à un client et calcul automatique de prix.**Les acteurs :** Le directeur commercial, le système.

Version : 1.0

Scénario alternatif

A1 : Des produits accessoires doivent être joints à certains produits commandés par le client.

Démarre après le point 5 du scénario nominal.

5.1- Le commercial sélectionne le produit auquel lier les produits accessoires.

5.2- Le système présente la liste des produits accessoires déjà dans le système.

5.3- Le commercial sélectionne les accessoires.

5.4- Pour chaque produit accessoire sélectionné le commercial précise le prix d'achat unitaire, la devise du prix d'achat (\$, €, ..), la marge bénéficiaire.

Le système reprend au point 7 du scénario nominal.

Cas d'utilisation : **Enregistrement d'une commande (Client->SOFTNET)****Résumé:** L'enregistrement d'une commande adressée par un client dans le système.**Les acteurs :** Le directeur commercial, le système.

Version : 1.0

Préconditions:

Le système doit contenir les produits commandés par le client.

Scénario nominal

<début>

- 1- Le commercial demande à enregistrer une commande.
- 2- Le système demande les informations sur la commande.
- 3- Le commercial saisit les données sommaires de la de la commande (la date, le nom et l'adresse du client, le délai de livraison...).
- 4- Le système présente la liste des proformas adressées au client (A1).
- 5- Le commercial sélectionne le numéro de la proforma concernée par la commande.
- 6- Le système ajoute les produits de la proforma à la commande (A2).
- 7- Le commercial saisit les quantités commandées si elles ne correspondent pas à celles qui figuraient sur la proforma.
- 8- Le système enregistre la commande.

<Fin>

Scénarii alternatifs**A1 :** Le client n'a pas reçu de proforma.

Démarré après le point 4 du scénario nominal.

- 4.1- Le commercial précise que le client n'a pas reçu de proforma.
- 4.2- Le système présente la liste des produits de la base.
- 4.3- Le commercial sélectionne les produits commandés.
- 4.4- Le système ajoute les produits sélectionnés à la commande.

Le système reprend au point 7 du scénario nominal.

Cas d'utilisation : Enregistrement d'une commande (Client->SOFTNET)

Résumé: L'enregistrement d'une commande d'un client.

Les acteurs : Le directeur commercial, le système.

Version : 1.0

Scénarii alternatifs (suite)

A2 : Certains produits de la proforma n'ont pas été commandés.

Démarre après le point 6 du scénario nominal.

6.1- Le commercial supprime les produits qui n'ont pas fait l'objet de commande.

Le système reprend au point 8 du scénario nominal.

Cas d'utilisation : Enregistrement d'une livraison (SOFTNET-> Client)

Résumé: L'enregistrement des produits livrés à un client.

Les acteurs : Le commercial, le gestionnaire de stock, le système du présent projet, le système de gestion de stock.

Version : 1.0

Pré conditions

Une commande non encore livrée doit être enregistrée par le commercial (point d'inclusion du cas d'utilisation «*Enregistrement d'une commande (Client->SOFTNET)* »)

Scénario nominal

<début>

- 1- Le gestionnaire de stock enregistre une sortie de matériel pour la livraison dans le système de gestion de stock. (raison de la sortie = commande, référence de la commande, liste des produits à livrée les quantités et les numéros de série).
- 2- Le commercial saisit les données sommaires de la commande (la date, le nom et l'adresse du client, les références de la commande, le technicien devant faire la livraison).
- 3- Le système sélectionne du système de gestion de stock, les informations sur les produits à livrer (référence, désignation, marque, type, caractéristique, numéro de série) et les présente au commercial.
- 4- Le commercial vérifie les quantités déstockées par rapport à celles commandées (A1).
- 5- Le système enregistre la livraison.
- 6- Le commercial imprime le bordereau de livraison pour le client.

<Fin>

Cas d'utilisation : Enregistrement d'une livraison (SOFTNET-> Client)	
Résumé: L'enregistrement des produits livrés à un client.	
Les acteurs : Le commercial, le gestionnaire de stock, le système du présent projet, le système de gestion de stock.	Version : 1.0
<p><u>Scénario alternatif</u></p> <p>A1 : La quantité commandée est supérieure à celle disponible en stock. Démarre après le point 4 du scénario nominal.</p> <p>4.1- Le système précise que la livraison est partielle et donne le numéro de la livraison (Première, deuxième, ...) ainsi que les reliquats des quantités. Le système reprend au point 5 du système nominal.</p>	
Cas d'utilisation : Facturation d'une livraison (SOFTNET-> Client)	
Résumé: L'établissement du facture à une livraison.	
Les acteurs : Le directeur général et le système.	Version : 1.0
<p><u>Pré conditions</u></p> <p>Une livraison non encore facturée doit être enregistrée dans le système (point d'inclusion du cas d'utilisation « <i>Enregistrement d'une livraison (SOFTNET-> Client)</i> »).</p> <p><u>Scénario nominal</u></p> <p><début></p> <ol style="list-style-type: none"> 1- Le directeur demande à enregistrer une facture. 2- Le système présente la liste des livraisons non encore facturées. 3- Le directeur général sélectionne la livraison concernée. 4- Le système enregistre la facture. 5- Le directeur imprime la facture qui sera adressée au client. <p><Fin></p>	

Cas d'utilisation : Vérification de garantie.

Résumé: Vérification de la garantie d'un matériel.

Les acteurs : Un technicien, le système.

Version : 1.0

Préconditions

Néant

Scénario nominal

<début>

- 1- Un technicien demande à vérifier la garantie d'un matériel.
- 2- Le système demande le numéro de série du matériel.
- 3- Le technicien saisit le numéro de série du matériel.
- 4- Le système vérifie la garantie du matériel (E1).
- 5- Le système donne résultat de la vérification (matériel sous garantie ou pas, la date d'expiration de la garantie, la référence de la commande, caution de garantie).

<Fin>

Scénario exceptionnel

E1 : Matériel non livré par SOFTNET-BURKINA.

Démarre après le point 4 du scénario nominal.

- 4.1 Le système affiche <<numéro de série inconnu>>.

Le système reprend à la balise <Fin> du scénario nominal.

Cas d'utilisation : Réception d'un matériel en panne

Résumé: L'enregistrement d'un matériel en panne apporté par un client pour dépannage.

Les acteurs : Un technicien, le système.

Version : 1.0

Préconditions

Néant

Scénario nominal

<début>

- 1- Le technicien demande à enregistrer une réception de matériel.
- 2- Le système demande les informations sur le matériel.
- 3- Le technicien saisit les informations sur le matériel (type de matériel, numéro de série, le client, la panne signalée par le client).
- 4- Le système vérifie la garantie du matériel (point d'inclusion du cas d'utilisation « *vérification de garantie* »).
- 5- Le système enregistre la réception de matériel.
- 6- Le technicien imprime la décharge de réception du matériel pour le client.

<Fin>

Cas d'utilisation : Diagnostic d'un matériel en panne

Résumé: L'enregistrement d'un diagnostic de matériel en panne.

Les acteurs : Un technicien, le système.

Version : 1.0

Préconditions

Un matériel doit être reçu pour dépannage (point d'inclusion du cas d'utilisation « *Réception d'un matériel en panne* »).

Le matériel ne doit pas être encore restitué au client.

Le technicien voulant faire le diagnostic doit être affecté sur le site où le matériel est reçu.

Scénario nominal

<début>

- 1- Le technicien demande à enregistrer un diagnostic.
- 2- Le système présente la liste de matériels reçus pour dépannage.
- 3- Le technicien choisit le numéro de réception du matériel diagnostiqué.
- 4- Le système demande les détails du diagnostic.
- 5- Le technicien saisit les détails du diagnostic.
- 6- Le système enregistre le diagnostic.

<Fin>

Cas d'utilisation : **Intervention**

Résumé: L'enregistrement des différentes opérations effectuées sur un matériel en vue de le dépanner

Les acteurs : Un technicien, le système.

Version : 1.0

Préconditions

Le matériel en dépannage doit être d'abord diagnostiqué (point d'inclusion du cas d'utilisation «*Diagnostic d'un matériel en panne* »).

Le matériel ne doit pas être encore restitué au client.

Le technicien voulant faire l'intervention doit être affecté sur le site où le matériel est reçu.

Scénario nominal

<début>

- 1- Le technicien demande à enregistrer une intervention.
- 2- Le système présente la liste de matériels reçus et non encore dépannés sur le site ou le technicien est affecté.
- 3- Le technicien sélectionne le numéro de réception du matériel.
- 4- Le système demande les détails de l'intervention.
- 5- Le technicien saisit les détails de l'intervention.
- 6- Le système enregistre l'intervention.

<Fin>

Cas d'utilisation : Restitution d'un matériel reçu pour dépannage au client

Résumé : L'enregistrement de la remise d'un matériel reçu pour dépannage au propriétaire.

Les acteurs : Un technicien, le système.

Version : 1.0

Préconditions

Le technicien voulant restituer le matériel doit être affecté sur le site où le matériel est reçu.

Scénario nominal

<début>

- 1- Le technicien demande à enregistrer une restitution de matériel.
- 2- Le système présente la liste de matériels reçus sur le site ou le technicien est affecté pour dépannage.
- 3- Le technicien choisit le numéro de réception du matériel à restituer (E1).
- 4- Le système demande le nom et l'adresse de la personne à qui le matériel est transmis.
- 5- Le technicien saisit le nom et l'adresse de la personne à qui le matériel est restitué.
- 6- Le système enregistre la restitution.
- 7- Le technicien imprime le bordereau d'intervention pour le client.

<Fin>

Scénario exceptionnel

E1 : Le matériel a déjà été restitué.

- 1- Le système informe le technicien que le matériel a été déjà restitué.

Le scénario reprend à la balise *<fin>* du scénario nominal.

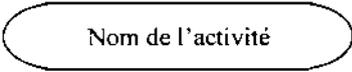
III.3. LES DIAGRAMMES D'ACTIVITÉ

III.3.1. DÉFINITION

Le diagramme d'activités sert à décrire conceptuellement les fonctionnalités d'un cas d'utilisation. On s'intéresse principalement aux activités qui y sont exécutées.

Il permet de bien comprendre les différentes activités réalisées et leurs enchaînements dans le temps.

III.3.2. FORMALISME

Présentation du symbole	Définition du symbole et du type d'information véhiculée
	Une activité.
	Une transition d'une activité à une autre.
 	Une barre de synchronisation Les transitions au départ d'une barre de synchronisation sont déclenchées simultanément. Cependant, la barre de synchronisation ne peut être franchie que lorsque toutes les transitions qui y aboutissent ont été déclenchées.
	Un marqueur d'état initial.
	Un marqueur d'état final.
	Une décision.

III.3.3. REPRÉSENTATION DES DIAGRAMMES D'ACTIVITÉ

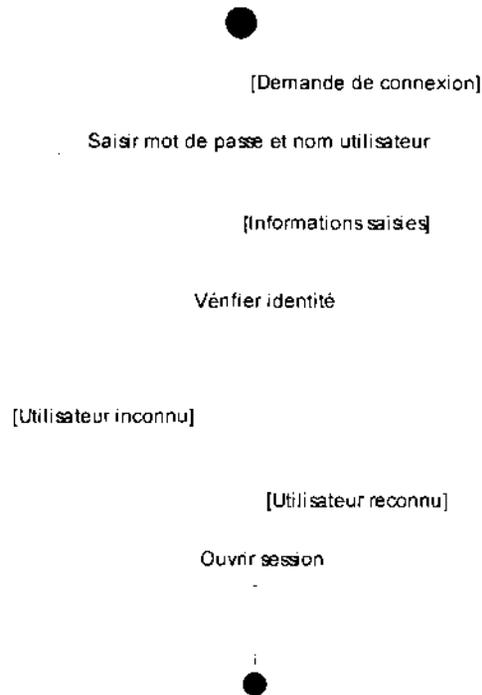


Figure III.B: Diagramme d'activité: Connexion

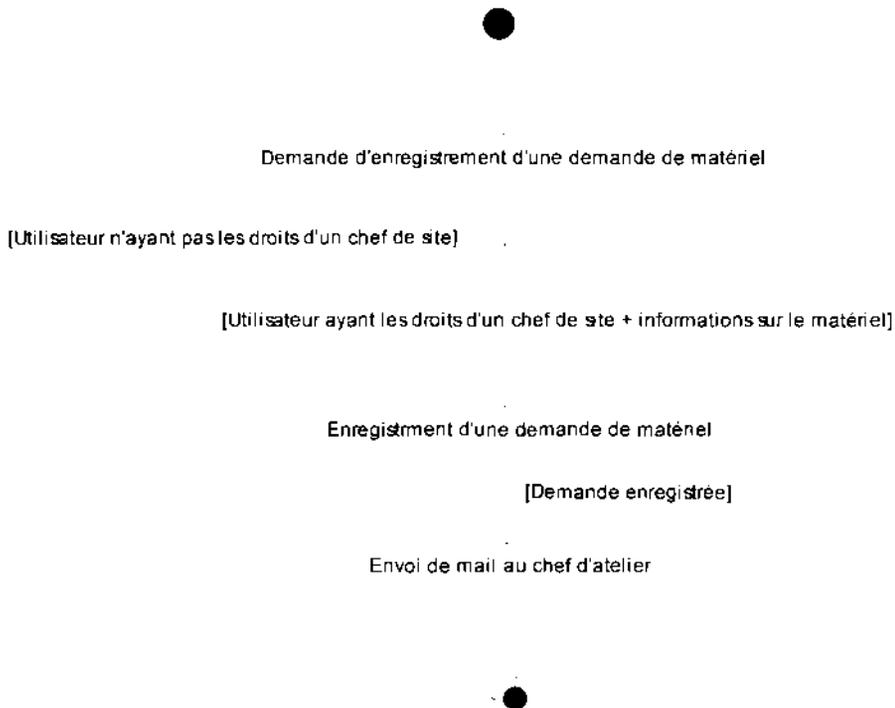


Figure III.C: Diagramme d'activité: Demande de matériels



Figure III.D : Diagramme d'activité : Validation d'une demande de matériels

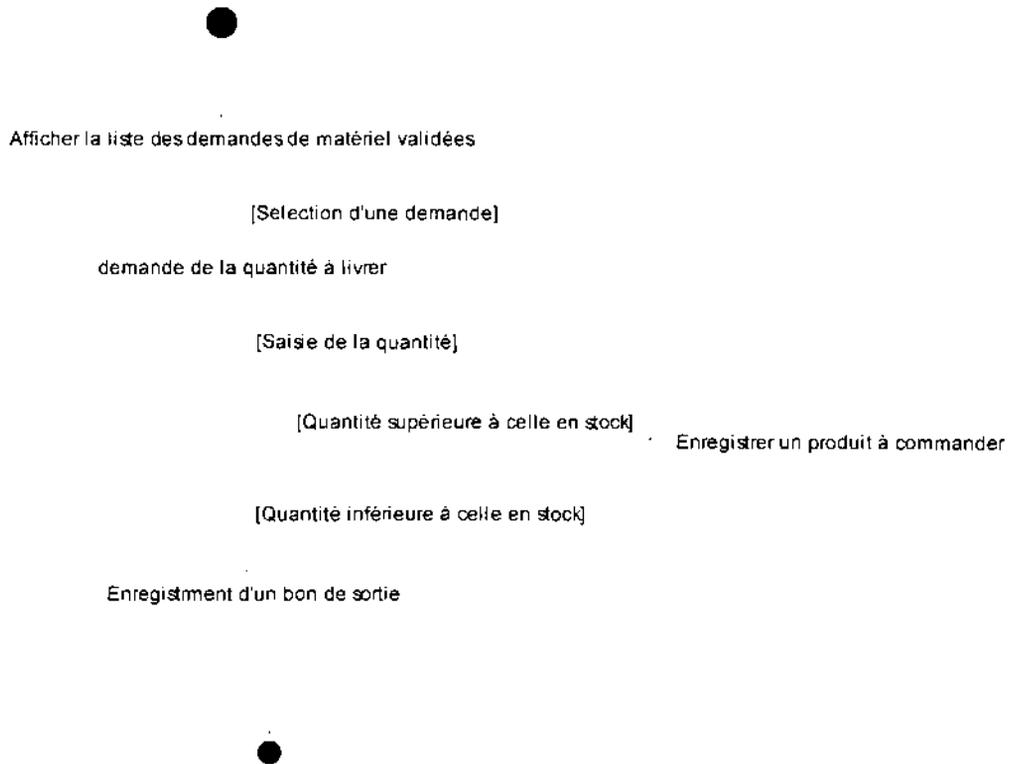


Figure III.E: Diagramme d'activité: Etablissement d'un bon de sortie.

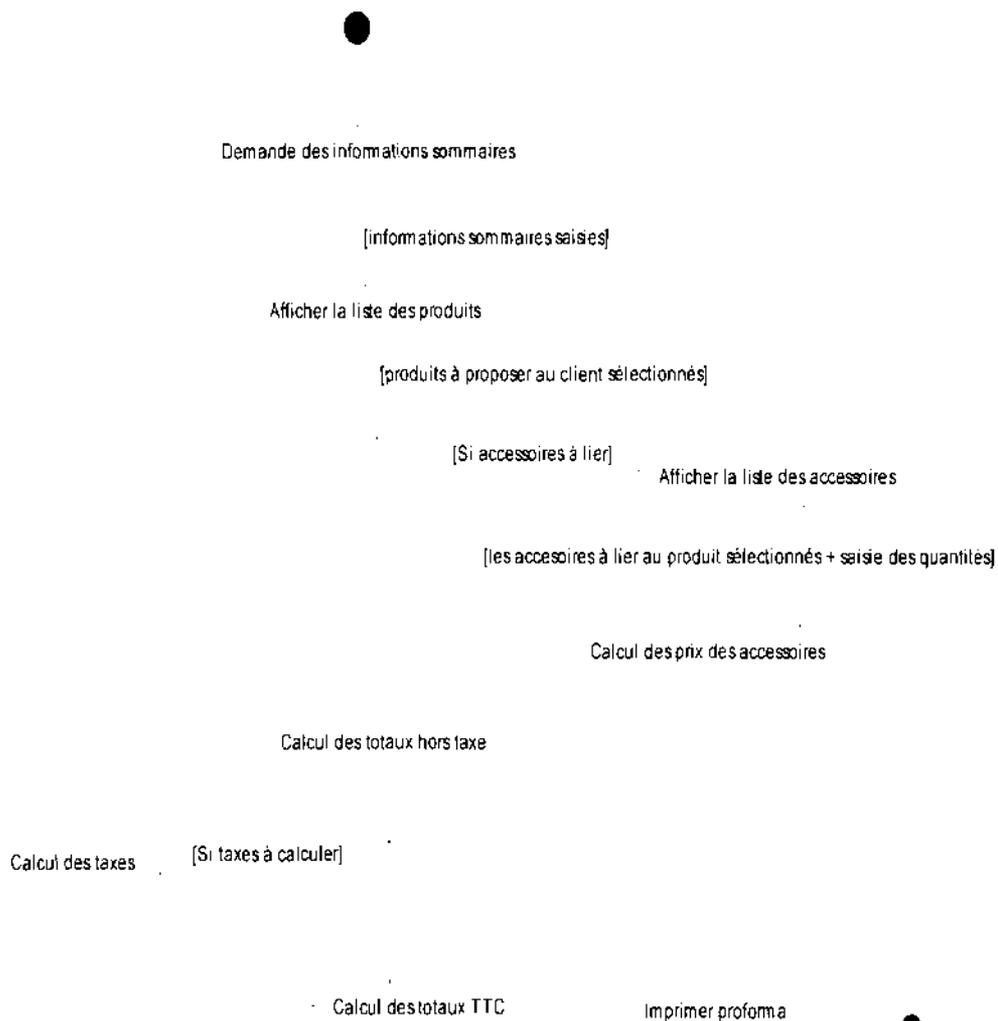


Figure III.F: Diagramme d'activité: Conception de proformas

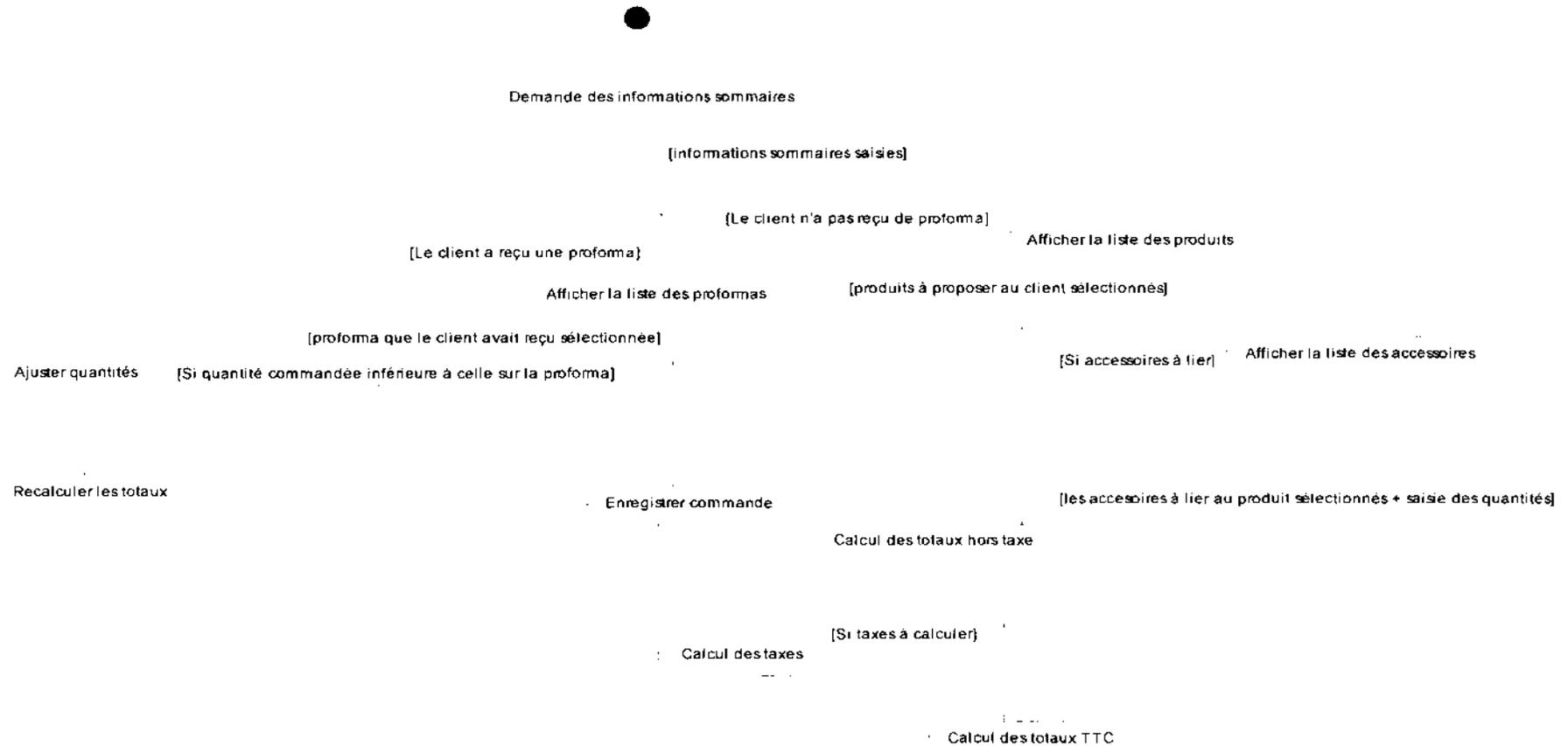


Figure III.G: Diagramme d'activité: Enregistrement d'une commande (Client->SOFTNET)

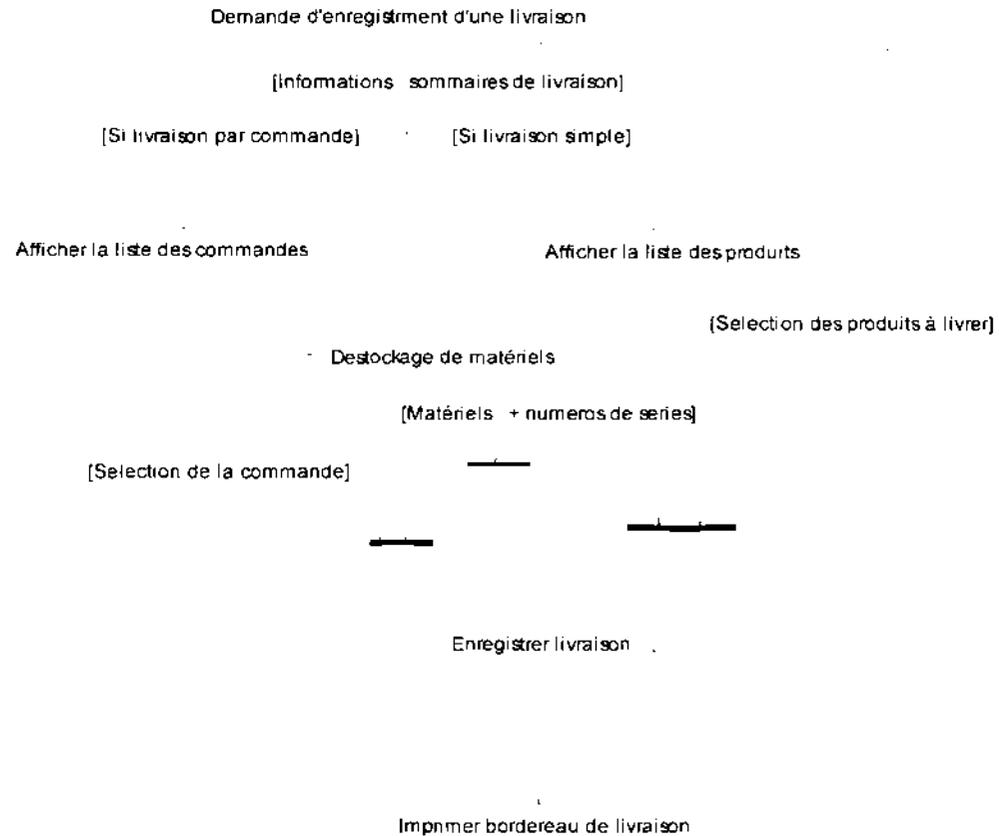


Figure III.H: Enregistrement d'une livraison (SOFTNET-> Client)

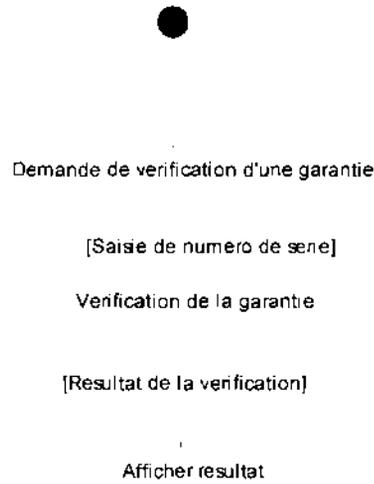


Figure III.I: Diagramme d'activité: Vérification de garantie

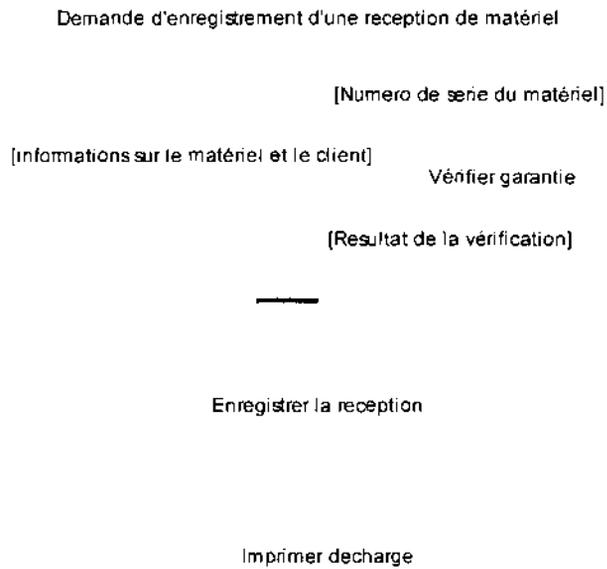


Figure III.J: Diagramme d'activité: Réception d'un matériel en panne

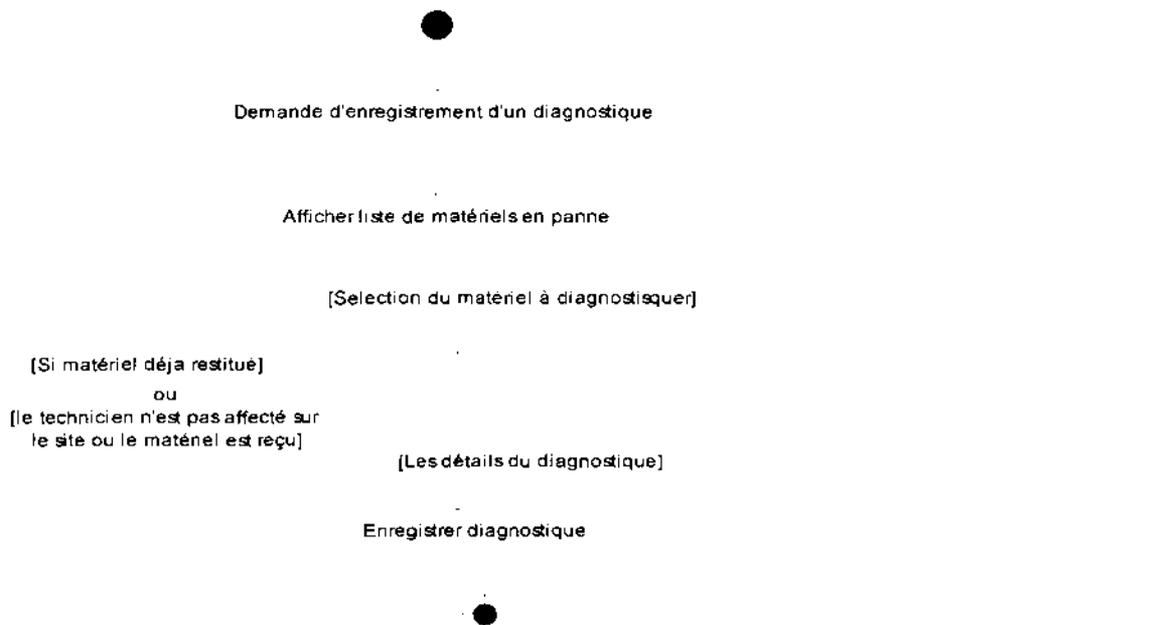


Figure III.K: Diagramme d'activité: Diagnostic d'un matériel en panne

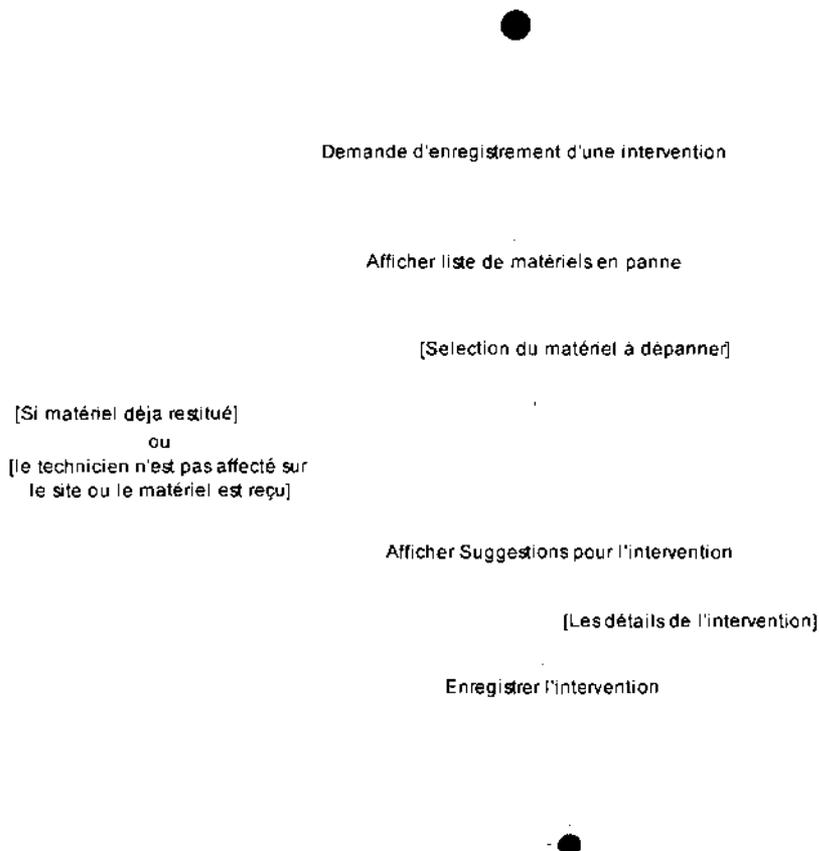


Figure III.L: Diagramme d'activité: Intervention

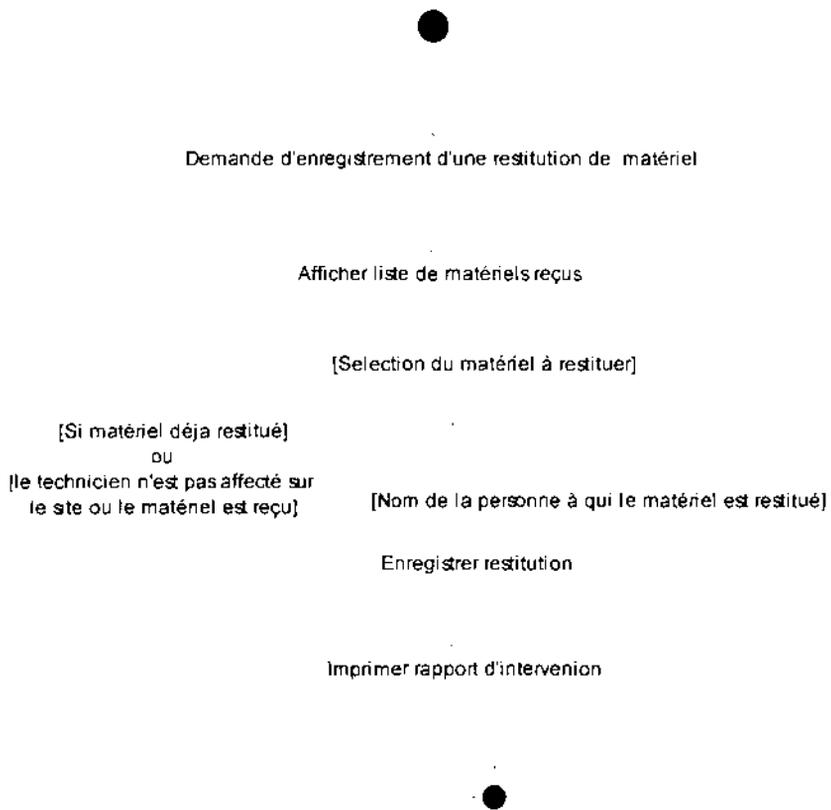


Figure III.M: Diagramme d'activité: Restitution d'un matériel reçu pour dépannage au client

III.4. LE DIAGRAMME DE CLASSES

III.4.1. DÉFINITION

Le diagramme de classes représente la structure statique d'un système. Il permet de modéliser les entités du système d'information.

III.4.2. LES CONCEPTS UTILISÉS

1. *Une classe* : c'est la description d'une famille d'objets ayant la même structure et le même comportement. La classe comporte des attributs et des méthodes ou opérations.

2. *Une opération ou méthode* : c'est une fonctionnalité assurée par la classe.

3. *Un attribut* : c'est une information élémentaire composant une classe. Un attribut peut permettre d'identifier la classe. Il est typé (integer, real, string ...).

4. *Une association* : c'est un lien sémantique entre des classes.

5. *Une multiplicité* : c'est le nombre d'instances d'une classe impliquée dans une association. Elle peut être représentée par un intervalle Min...Max.

6. *Une classe association* : c'est une association porteuse d'attributs.

7. *Une agrégation* : c'est un type particulier d'association. Elle met en évidence une classe agrégat et une classe agrégée. Elle permet de représenter des relations de type maître et esclaves

Exemple

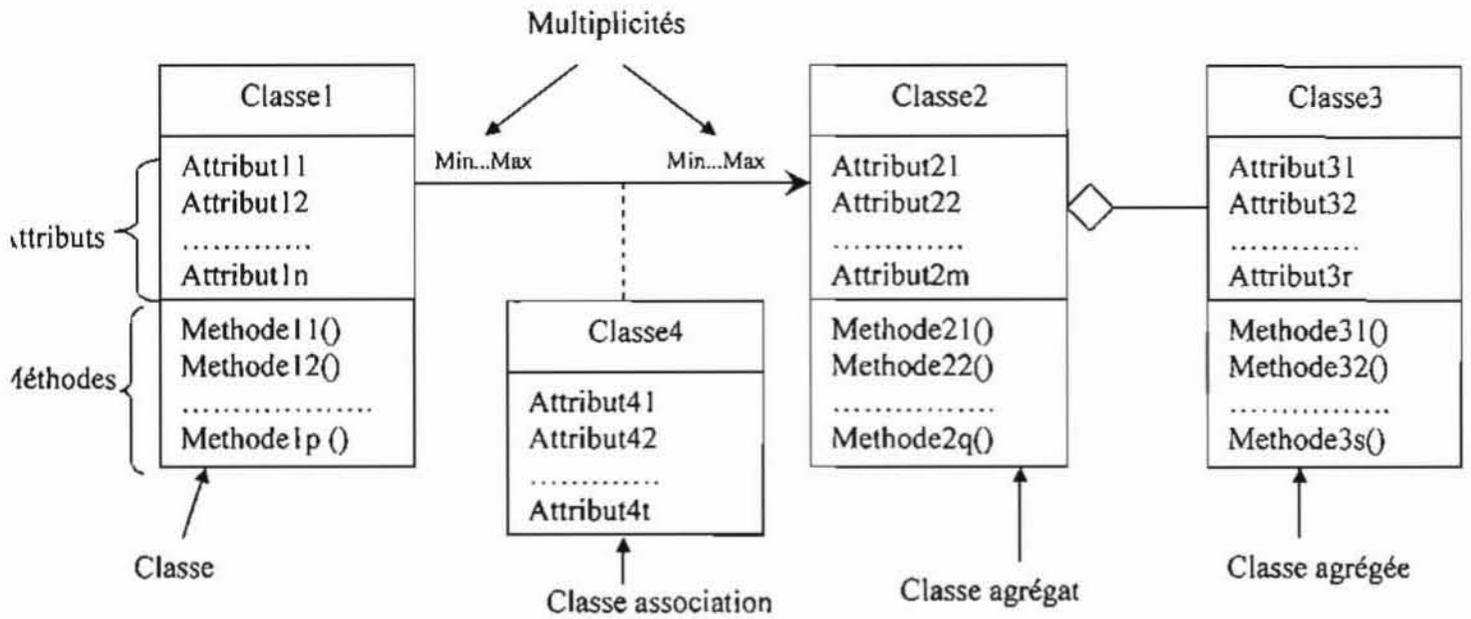


Figure III.G: un exemple de diagramme de classe.

NB : Dans la représentation suivante du diagramme de classe nous faisons abstraction des méthodes.

III.4.3. REPRÉSENTATION DU DIAGRAMME DE CLASSES

IV. ÉTUDE CONCEPTUELLE DE LA BASE DE DONNÉES

La base de données du présent projet est relationnelle. L'étude conceptuelle de sa réalisation est faite grâce au modèle conceptuel de données et du modèle logique de la méthode MERISE.

IV.1. LA MÉTHODE MERISE

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. Elle est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques.

La partie traitement et les fonctionnalités du système étant déjà étudiées plus haut (avec UML), nous utilisons MERISE pour l'analyse des données.

La méthode MERISE prévoit une conception des données par niveaux, et définit pour cela 3 niveaux essentiels:

Le niveau conceptuel : Modèle conceptuel des données

Il décrit l'ensemble des données du système d'information, sans tenir compte de l'implémentation informatique de ces données.

Le niveau logique : Modèle logique des données

Il prend en considération l'implémentation du système d'information par un SGBD (Système de Gestion de Base de Données).

Ce niveau introduit la notion des tables logiques, et constitue donc le premier pas vers les tables des SGBD.

Le niveau physique : Modèle physique des données

Il contient finalement les tables définies à l'aide d'un SGBD spécifique.

IV.2. LE MODÈLE CONCEPTUEL DE DONNÉES

IV.2.1. DÉFINITION

Le Modèle Conceptuel de Données (MCD) est la représentation formelle de l'ensemble des données manipulées par le système d'information.

IV.2.2. LES CONCEPTS UTILISÉS

1. *Une entité* : C'est une représentation d'un objet matériel ou immatériel qui a un intérêt pour le domaine d'étude.
2. *Une propriété* : Elle permet de conceptualiser un type d'information du domaine.
3. *Identifiant* : C'est une propriété correspondant à une seule occurrence de l'entité. Elle sert à référencier chaque occurrence de l'entité de façon unique.
4. *Une relation* : Elle conceptualise les liens sémantiques qui peuvent exister entre les occurrences d'entités.
5. *Une cardinalité* : Les cardinalités d'une relation entre entités indiquent le nombre d'occurrence maximal et minimal de participation des entités dans la relation.

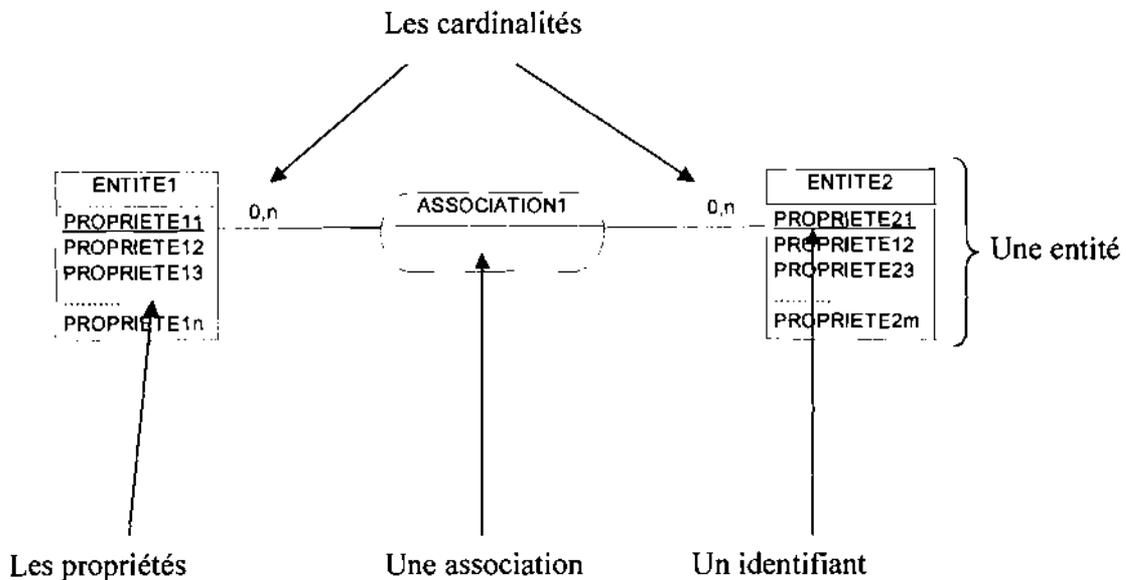


Figure IV.A: Exemple de MCD

IV.3. LE MODÈLE LOGIQUE DES DONNÉES

Le Modèle Logique des Données (MLD) est une transformation du modèle conceptuel des données. Il dépend du type de base de données utilisé.

V. LES PROCÉDURES DE SÉCURITÉ

V.1. LA MISE EN PLACE D'UN RÉSEAU VPN

La direction technique et le siège seront reliés par un VPN (Virtual Private Network) formant ainsi un Intranet. L'avantage d'une telle architecture c'est qu'elle permet d'utiliser Internet pour connecter deux (ou plusieurs) réseaux géographiquement éloignés. Les informations transitant d'un bout à l'autre des réseaux sont cryptées. De ce fait, un tunnel logique et sécurisé se crée entre les deux réseaux donnant ainsi un seul et unique réseau logique. Avec une telle architecture, l'application hébergée au siège peut être accessible depuis la direction technique en utilisant une connexion Internet mais avec un bon niveau de sécurité.

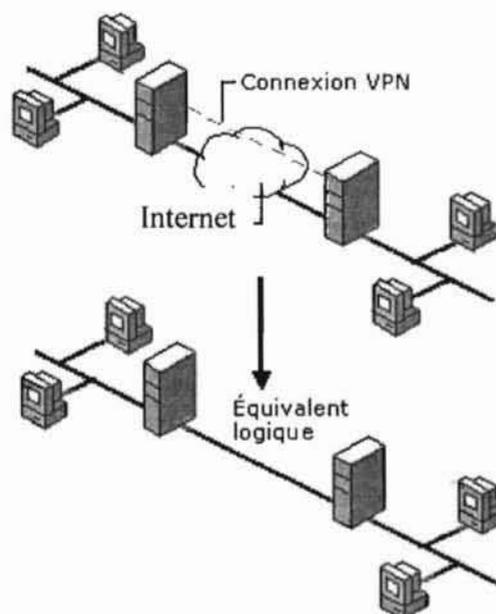


Figure V.A : Une liaison VPN entre la direction technique et le siège.

La même technique d'accès sera déployée entre le siège et les sites de maintenance. Les techniciens pourront également accéder à l'application via un tunnel sécurisé du VPN.

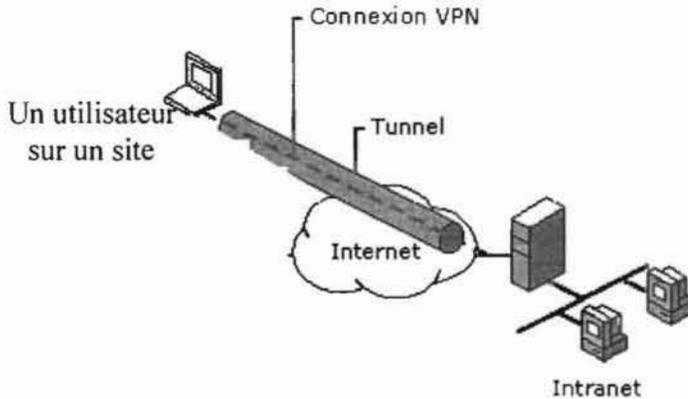


Figure V.B : Une connexion VPN à partir d'un site.

Une telle architecture nous protège des attaques de type «écoute clandestine du réseau».

V.2. LA ROBUSTESSE DE L'APPLICATION

Dans la mise en place d'une application Web, la sécurité doit être un des principaux soucis du développeur. La robustesse d'une application web ne dépend pas seulement du pare-feu utilisé mais aussi des précautions prises au moment du développement. Avec l'architecture VPN, l'application n'est accessible que dans l'Intranet. Ce qui réduit considérablement la surface d'attaque par rapport à une application ouverte à Internet et exposée à tout utilisateur d'Internet. Ce pendant, l'attaque pouvant venir de l'intérieur il est nécessaire de sécuriser l'application elle-même. Nous montrons ici les règles de base d'une application web sécurisée du point de vue du développeur que nous avons implémenté pour garantir la sécurité de l'application. Ces règles consistent à protéger des données en entrée, à limiter les informations en sortie et à se protéger des principales attaques.

V.2.1. LES INJECTIONS DE CODES

L'injection de code se produit quand un attaquant provoque l'exécution de code arbitraire dans le contexte de sécurité de votre application. Le risque augmente si l'application s'exécute à l'aide d'un compte privilégié. Il existe plusieurs types d'attaques par injection de code. Les principaux sont : l'injection SQL et le cross-site-scripting (XSS)

☒ L'INJECTION SQL

Cette attaque vise le code d'accès aux données vulnérables. L'attaquant envoie du code SQL entrant qui modifie la requête attendue ou qui exécute des requêtes totalement nouvelles dans la base de données. Les pages de connexion sont les cibles courantes du fait que le nom d'utilisateur et le mot de passe sont employés pour l'interrogation de la base de données. C'est une conséquence directe d'un mauvais contrôle des données entrées par l'utilisateur. En effet, les caractères « ' » et « ; » peuvent être utilisés pour enchaîner plusieurs requêtes SQL à la suite l'une de l'autre. Elle concerne les requêtes de type " *SELECT * FROM Users WHERE Nom = '' + txtbNom + ''* (txtbNom est une zone de texte donc une entrée de l'utilisateur). Si un utilisateur entre dans le contrôle txtbNom la chaîne suivante :

*'Ousmane'; DELETE * FROM Users WHERE Nom = 'Zoungrana' "* alors il supprimera les utilisateurs s'appelant 'Zoungrana' de la table Users. L'attaquant profite ainsi de la richesse du langage SQL pour détruire des données.

☒ LE CROSS-SITE SCRIPTING (XSS)

Le principe du Cross Site Scripting (ou XSS) est d'attaquer les utilisateurs de l'application plutôt que l'application elle-même. Au lieu de saisir de « bonnes » données sur un formulaire, un attaquant saisit un script malveillant. L'application enregistre ce script dans la base de données puis l'envoie aux utilisateurs comme des données ordinaires. Lorsqu'il est reçu par le navigateur client, ce code est alors exécuté sur le poste de la victime, dans le contexte de l'application, et dans le contexte de sécurité de l'utilisateur courant. Le code malveillant peut avoir plusieurs objectifs néfastes et agir comme un virus.

☞ UNE SOLUTION : LE CONTRÔLE DES DONNÉES ENTRÉES PAR L'UTILISATEUR

Les données entrées doivent être minutieusement analysées et vérifiées avant leur utilisation afin qu'elles ne créent pas de dommages dans l'application. En plus de vérifier si les données sont correctes, il faut aussi vérifier qu'elles ne sont pas nocives. Asp.Net fournit un nombre d'outils qui permettent de vérifier les données :

- ☞ La propriété « *MaxLength* » permet de limiter la taille des données saisies dans les textbox.
- ☞ Le *RequiredFieldValidator* est un composant qui permet de vérifier qu'un champ X a été renseigné (n'est pas vide).
- ☞ Le *CompareValidator* permet de comparer la valeur entrée avec une autre valeur fixée.
- ☞ Le *RangeValidator* permet de vérifier qu'un nombre ou une chaîne de caractères voire une date est entre 2 bornes.
- ☞ Le *RegularExpressionValidator* permet des vérifications très puissantes en utilisant les expressions régulières.
- ☞ Le *CustomValidator* permet de créer notre propre logique de validation.
- ☞ Le *ValidationSummary* est un contrôle qui permet de résumer toutes les erreurs trouvées sur la même page web.

Pour l'ensemble de ces contrôles, il s'agit de validations coté client, si un utilisateur désactive les scripts coté client, les contrôles ne seront plus faits. Nous faisons des contrôles aussi coté serveur.

Les principales mesures sont :

- ☞ Limiter la taille des données en entrée. Par exemple si un nom ne fait que 10 caractères, alors limiter l'entrée à 10 caractères.
- ☞ Interdire les mots clés tels SELECT, INSERT, DELETE, UNION, LIKE... qui sont les primitifs des requêtes SQL.
- ☞ Interdire les caractères < et < qui forment les balises des scripts.

V.2.2. LES ATTAQUES SUR LES IDENTIFIANTS DE SESSION

Le protocole HTTP est un protocole déconnecté. C'est-à-dire qu'entre deux requêtes, la connexion entre le client et le serveur est coupée. Le serveur ne peut donc pas reconnaître un client qui a déjà commencé une transaction dans l'application Web. Pour remédier à cela, on utilise un identifiant de session, échangé à chaque demande de page entre le client et le serveur, que ce soit au niveau du cookie, de l'url ou d'un champ caché de formulaire. Le serveur maintient ainsi un contexte de transaction pour chaque identifiant de session généré.

Une attaque classique consiste à voler la session d'un utilisateur qui vient de s'authentifier sur le système en essayant de deviner la valeur de son identifiant de session. Si la valeur de celui-ci est découverte, un attaquant peut alors se faire passer pour l'utilisateur légitime en injectant l'identifiant récupéré dans sa propre session. Les attaques de type XSS peuvent être utilisées pour lire la variable de session d'un utilisateur et l'envoyer à un attaquant. Ce dernier peut donc se faire passer pour un utilisateur légitime en injectant la variable dans ses requêtes. Lorsque certaines erreurs d'exécution surviennent, les pages d'erreurs affichées sont très riches en informations sur les identifiants de session. Ces informations peuvent être utilisées pour deviner les identifiants de sessions des utilisateurs. Pour palier à ce type d'attaque nous empêchons la remontée des erreurs jusqu'aux utilisateurs en proposant une page d'erreurs personnalisée.

V.2.3. LA RÉAUTHENTIFICATION DES UTILISATEURS

Le manque de réauthentification permet souvent à un attaquant de prendre le contrôle des comptes utilisateurs. Pour palier à ce type d'attaque nous décidons de réauthentifier les utilisateurs connectés toutes les vingt minutes. Ainsi même si un attaquant arrive à prendre le contrôle d'une session son action sera limitée ans le temps.

V.2.4. LA PROTECTION DES MOTS DE PASSE ET NOMS D'UTILISATEURS

Un autre type d'attaque consiste à utiliser une autre application pour accéder à la base de données et récupérer les mots de passe et les noms d'utilisateurs. Une fois qu'il accède à ces informations, l'attaquant peut se faire passer pour un utilisateur légitime de l'application. La solution que nous

avons implémentée pour protéger l'application de ce type d'attaque est le cryptage des mots de passe et noms d'utilisateurs et la sécurisation de la base de données avec un mot de passe.

Visual studio .net 2003 permet de crypter les mots de passe selon deux (2) algorithmes cryptographiques SHA1 et MD5.

MD5 (Message Digest 5) est une fonction de hachage cryptographique qui permet d'obtenir pour chaque message une empreinte numérique (en l'occurrence une séquence de 128 bits) avec une probabilité très forte que, pour deux messages différents, leurs empreintes soient différentes.

MD5 est très populaire, mais n'est plus considéré comme un algorithme sûr.

SHA1 (Secure Hash Algorithm) est une fonction de hachage cryptographique conçue par la National Security Agency des États-Unis (NSA), et publiée par le gouvernement des États-Unis comme un standard fédéral de traitement de l'information (Federal Information Processing Standard du NIST). Elle produit un résultat (appelé « hash » ou condensat) de 160 bits.

Nous utilisons ce dernier pour garantir un minimum de sécurité des mots de passe. Le souci étant de ne pas laisser les mots de passe en claire dans la base.

Exemple de *condensat* :

SHA1 ("zoung&ous") = 9EC1D01CEB1B46C64C003F39A77930E267080560

En modifiant un caractère, la signature change radicalement.

SHA1 ("Zoung&ous ") = 6F84373558F0AB09DF5D2D028C444594431A4498

V.2.5. L'OBLIGATION DE LA CONNEXION

L'étape de la connexion est obligatoire pour tout utilisateur de l'application. Avant d'avoir accès à toute fonctionnalité du système les utilisateurs doivent d'abord s'authentifier à l'aide de leur mot de passe et nom d'utilisateur. Cette obligation de la connexion est prise en charge par .Net mais à travers des cookies pour différencier les utilisateurs déjà connectés à ceux en quête de connexion. Comme les cookies peuvent être désactivés coté client. Il nous appartient d'implémenter un contrôle à la demande de chaque page. En effet quelque soit l'arsenal déployé au niveau de la page de connexion, si les attaquants ont la possibilité d'accéder aux pages protégées rien qu'en saisissant leurs URLs, ce n'est que peine perdue. Avant d'ouvrir toute page, nous vérifions que l'utilisateur est passé par la page de connexion. Si tel n'est pas le cas, il y'a une redirection systématique vers la page de connexion. Il est de même pour le contrôle des

autorisations. Une fois l'utilisateur authentifié, une variable de session est créée pour permettre la vérification des autorisations à la demande de chaque page.

V.3. LA SÉCURITÉ DU SERVEUR WEB

Certaines vulnérabilités, exploitables par les attaquants, de IIS sont de fois découvertes mais aussitôt Microsoft met sur son site des patches de sécurité que l'on peut télécharger pour la mise à jour. Il faut donc se tenir informé pour pouvoir appliquer les dernières mises à jour.

V.4. SAUVEGARDE RÉGULIÈRE DE LA BASE DE DONNEES

La base de données doit être régulièrement sauvegardée pour restaurer le système en cas d'éventuels dommages des données

VI. LE TRAVAIL RÉALISÉ

VI.1. LE DIAGRAMME D'ENCHAÎNEMENT DES PAGES

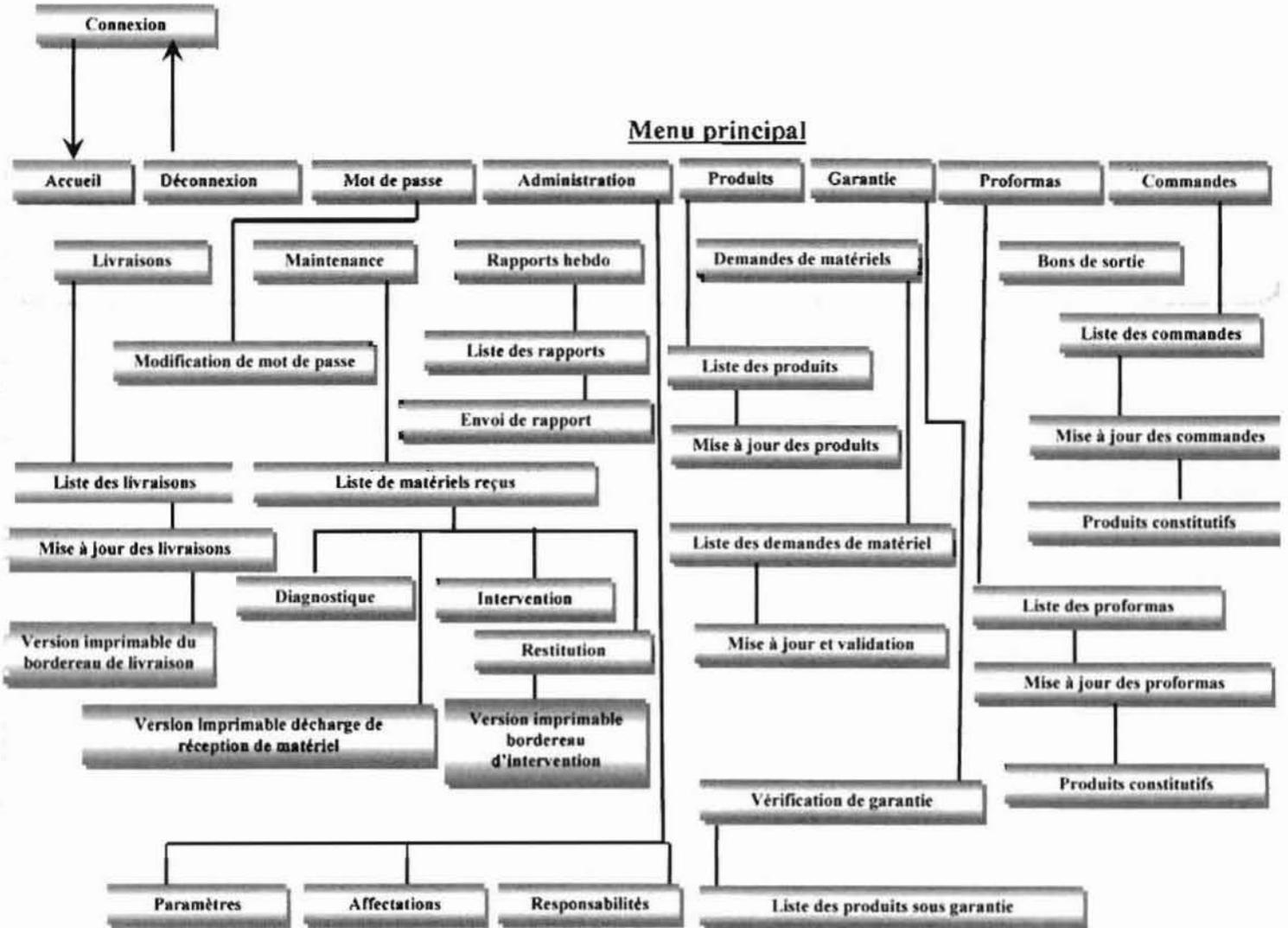
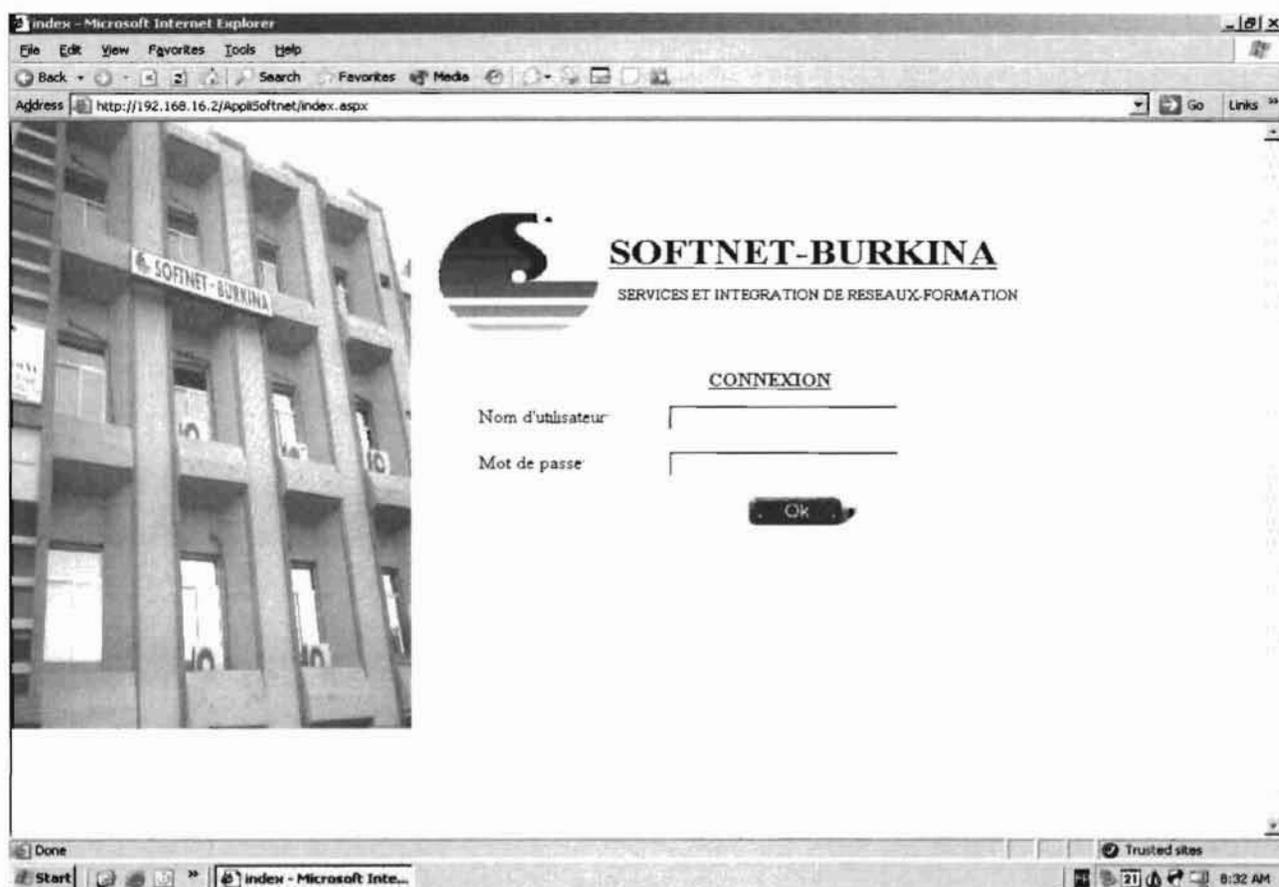


Figure VI.A : Diagramme d'enchaînements des pages

VI.2. PRÉSENTATIONS DE QUELQUES PAGES

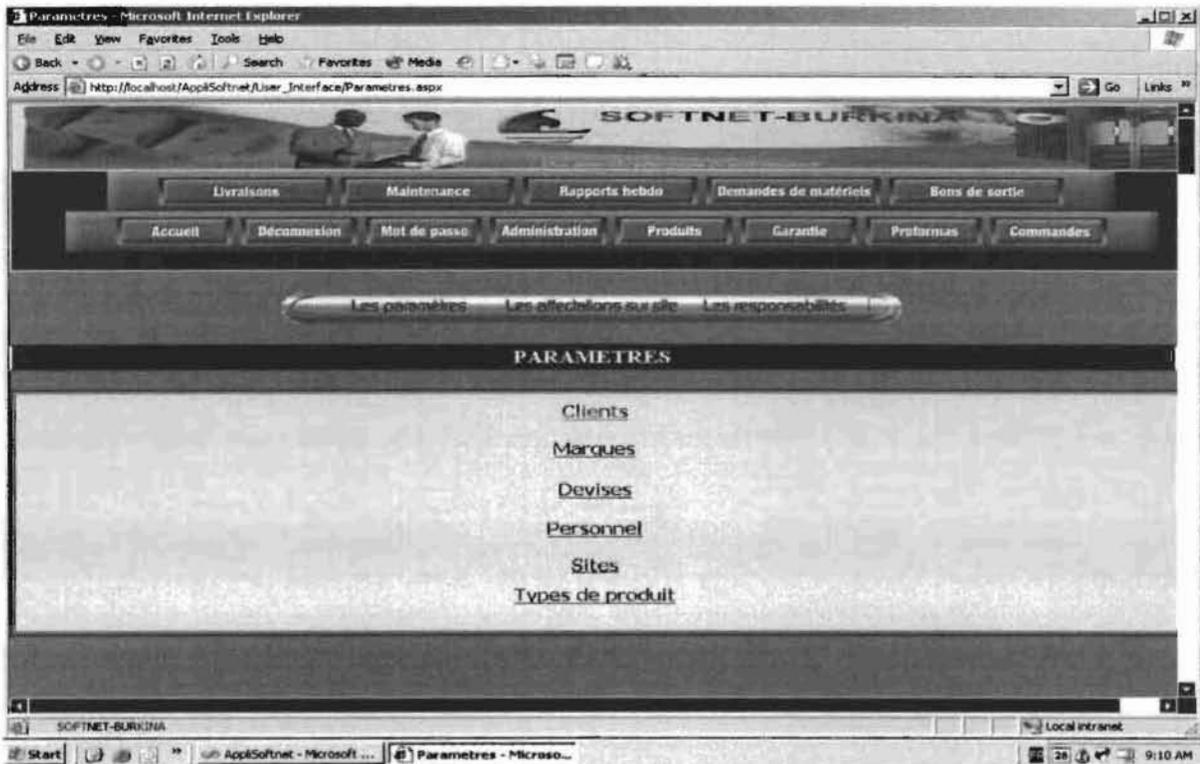
VI.2.1. LA PAGE DE CONNEXION



FigureVI.B: La page de connexion

VI.2.2. LES PAGES D'ADMINISTRATION

La page suivante est la première qui s'affiche lorsque l'administrateur clique sur « Administration » dans le menu. Un sous menu donne accès à la page des paramètres, la page des affectations du personnel sur les sites et la page d'attribution des responsabilités.



FigureVI.C: La page des paramètres

Chacun des liens sur la page des paramètres donne accès à des écrans de mise à jour. Nous présentons plus bas les écrans de mise à jour du personnel et des clients.

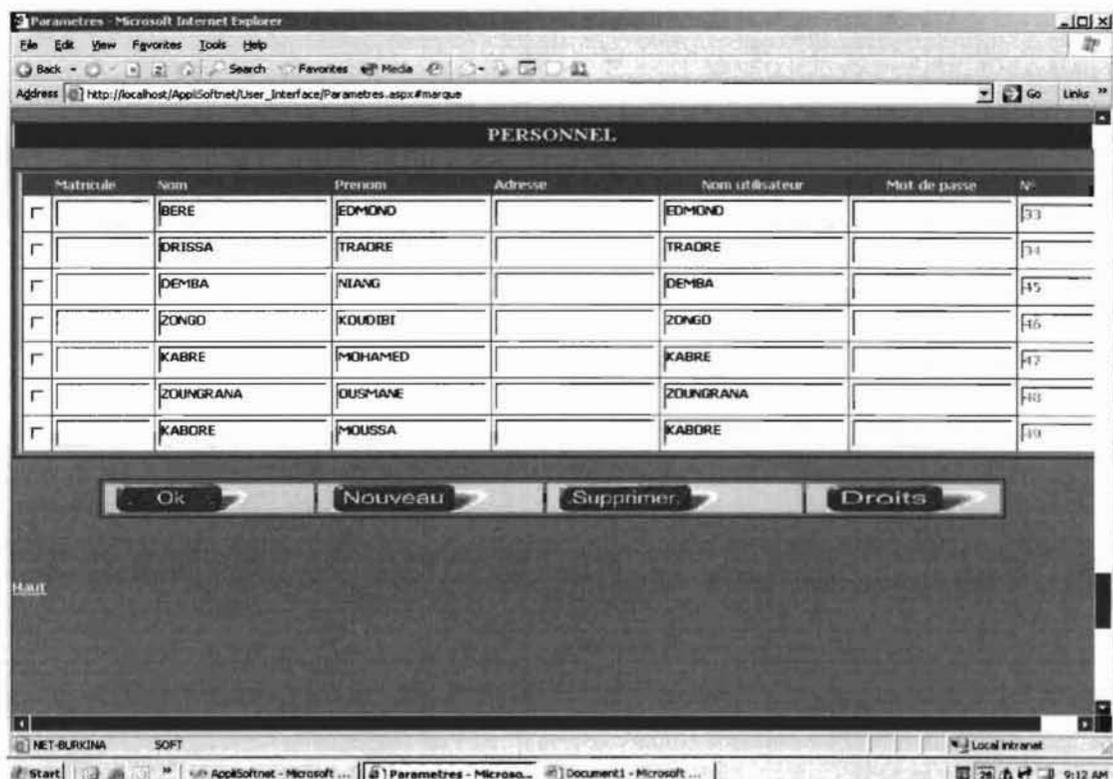


Figure VI.D: Écran de mise à jour du personnel

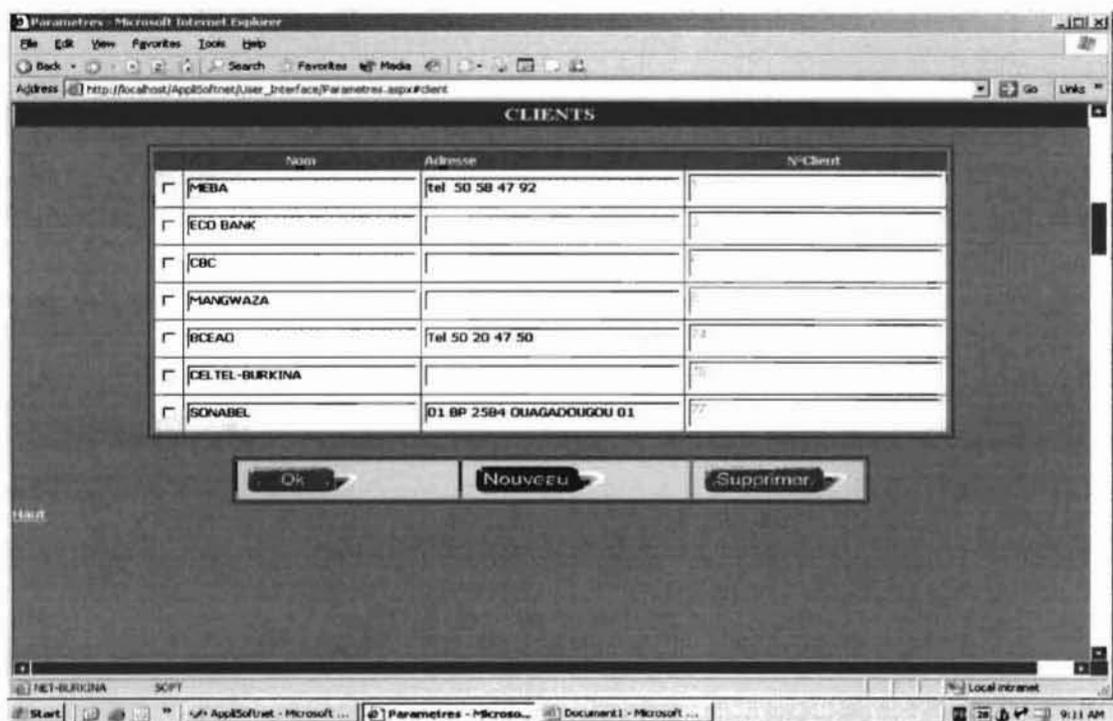


Figure VI.E : Écran de mise à jour des clients

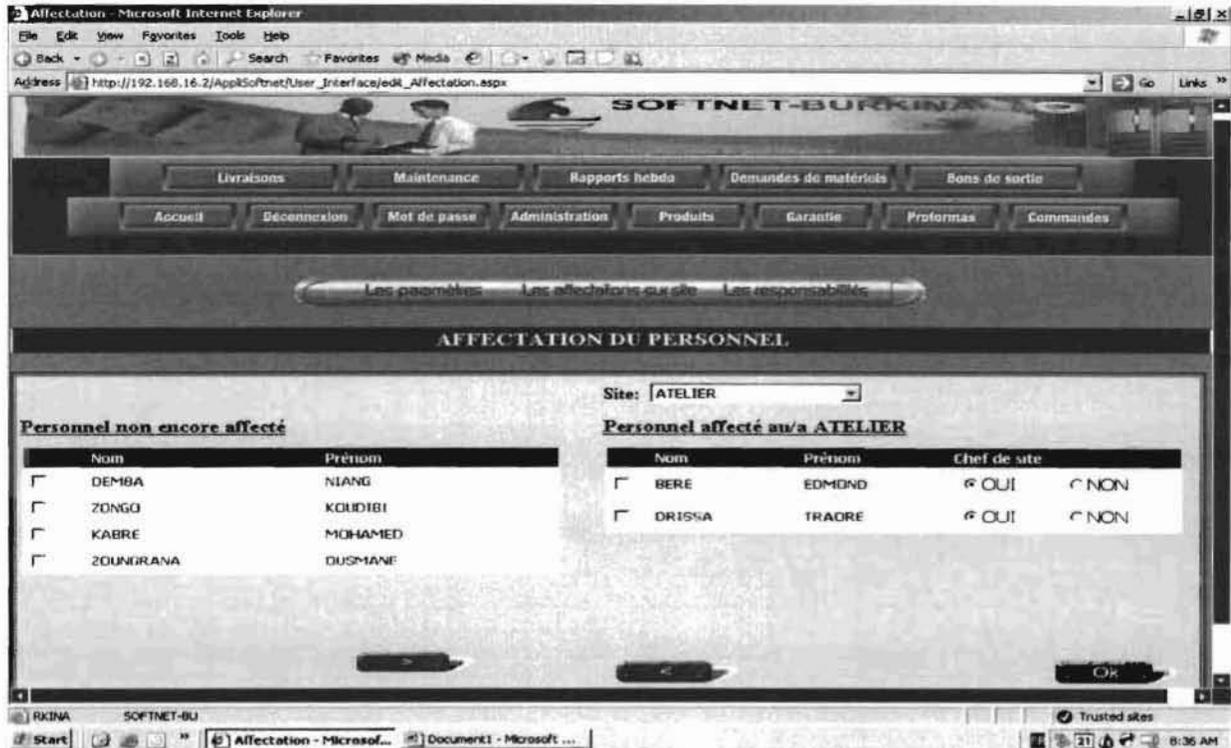


Figure VI.F: La page des affectations sur site

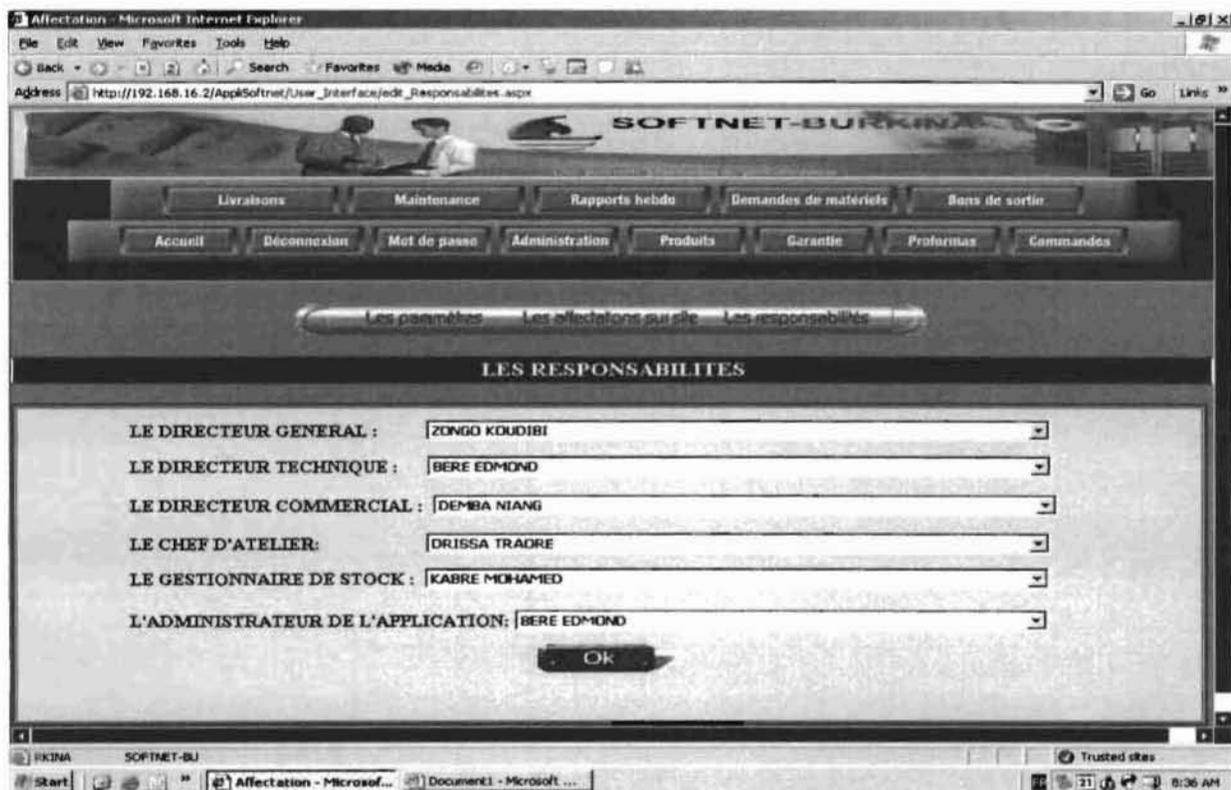


Figure VI.G: La page d'attribution des responsabilités

VI.2.3. LES PAGES DES PRODUITS

Les différents produits sont enregistrés dans la base pour former un catalogue.



Figure VI.H: Le catalogue des produits

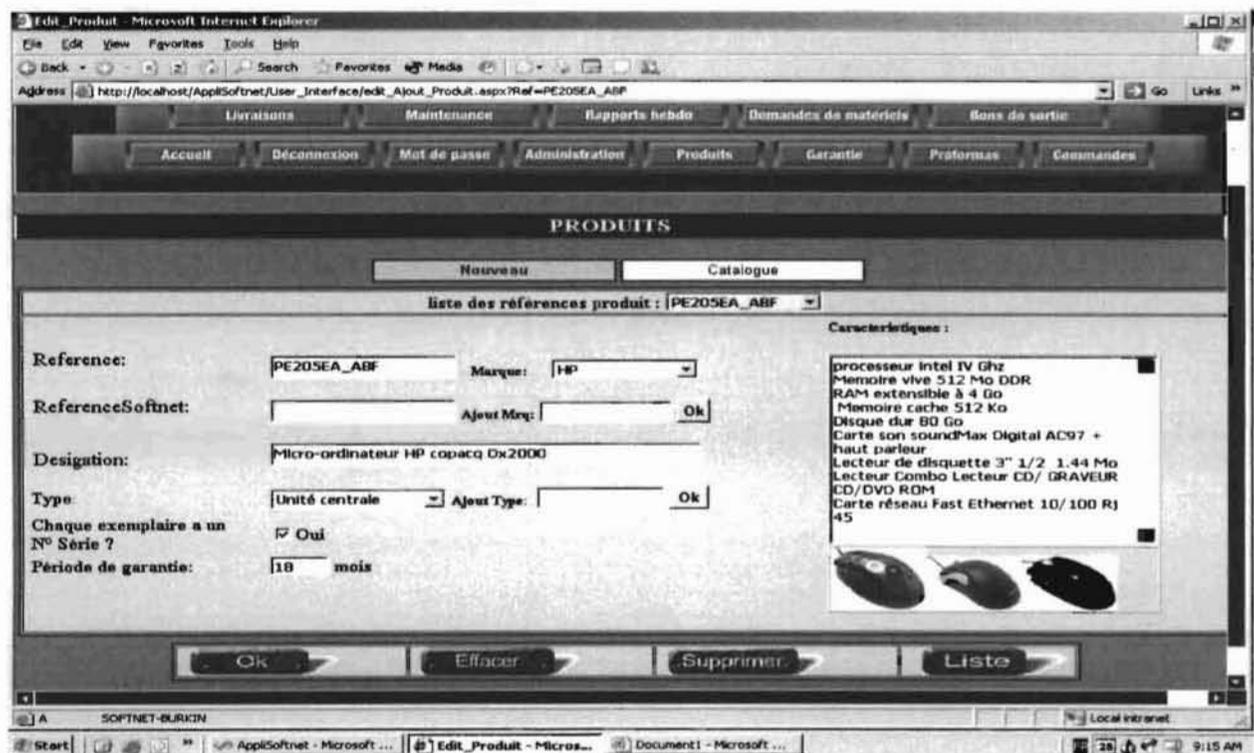


Figure VI.I : La page de mise à jour des produits

VI.2.4. LA PAGE DE VÉRIFICATION DE GARANTIE

La page de vérification de garantie permet d'interroger la base pour déterminer si un produit livré est sous garantie ou pas. L'interrogation se fait sur la base du numéro de série.

The screenshot shows a web browser window titled 'Edit_livraison - Microsoft Internet Explorer'. The address bar contains 'http://192.168.16.2/AppiSoftnet/User_Interface/edit_rechercheGarantie.aspx'. The page content is as follows:

VERIFICATION DE GARANTIE

Buttons: **Vérification de garantie** | **Liste des produits sous garantie**

Form: **Numéro série:** FGDHFHGFHDH

Button: **Ok**

PRODUIT SOUS GARANTIE

Informations complémentaires

N° livraison:	000002/2005
Ref Commande:	CMD1
Client:	CEL TEL-BURKINA
Date de Livraison:	6/11/2005
Livré par:	DRISSA TRAORE
Reçu par:	

The taskbar at the bottom shows the system tray with the time 8:30 AM and the taskbar with open windows for 'Edit_livraison - Micro...' and 'Document1 - Microsoft...'.

Figure VI.J: La page de vérification de garantie

VI.2.5. LES PAGES D'ÉTABLISSEMENT D'UNE FACTURE PROFORMA

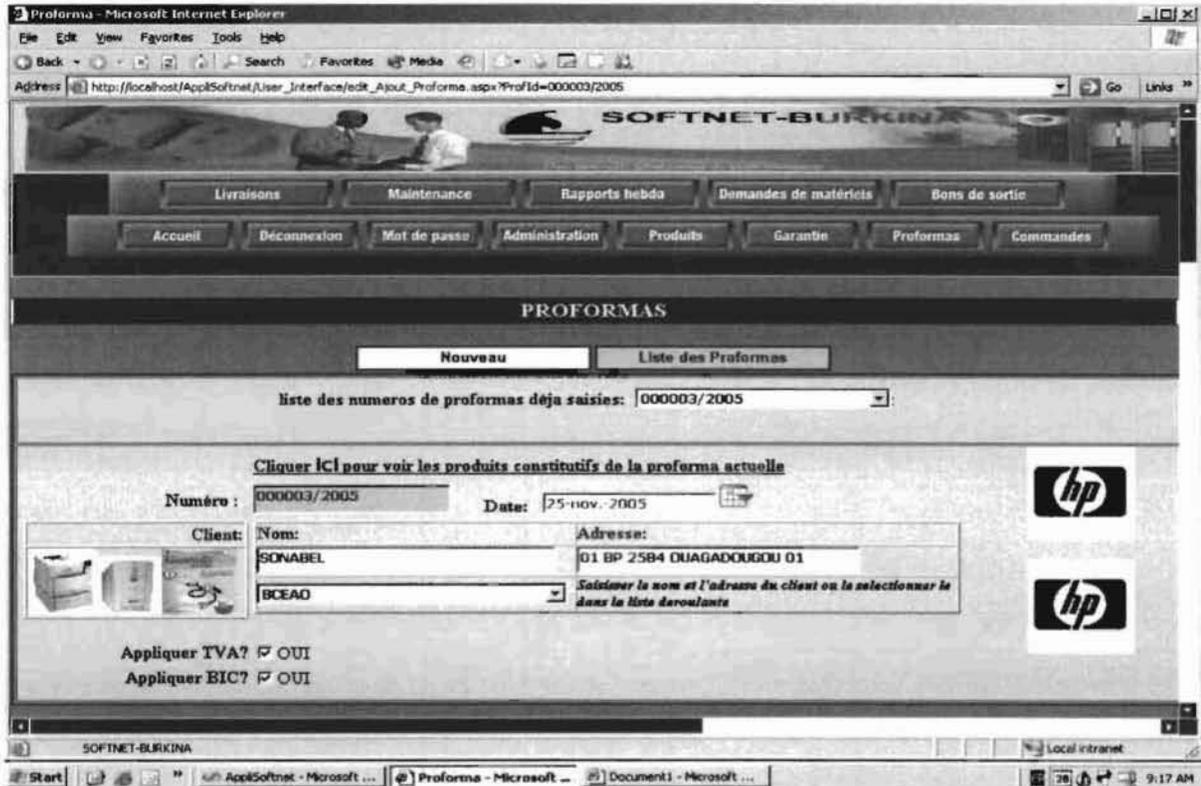


Figure VI.K: L'écran des informations sommaires d'une proforma

Un clic sur le lien « ici » présente l'écran des produits constitutifs de la proforma.

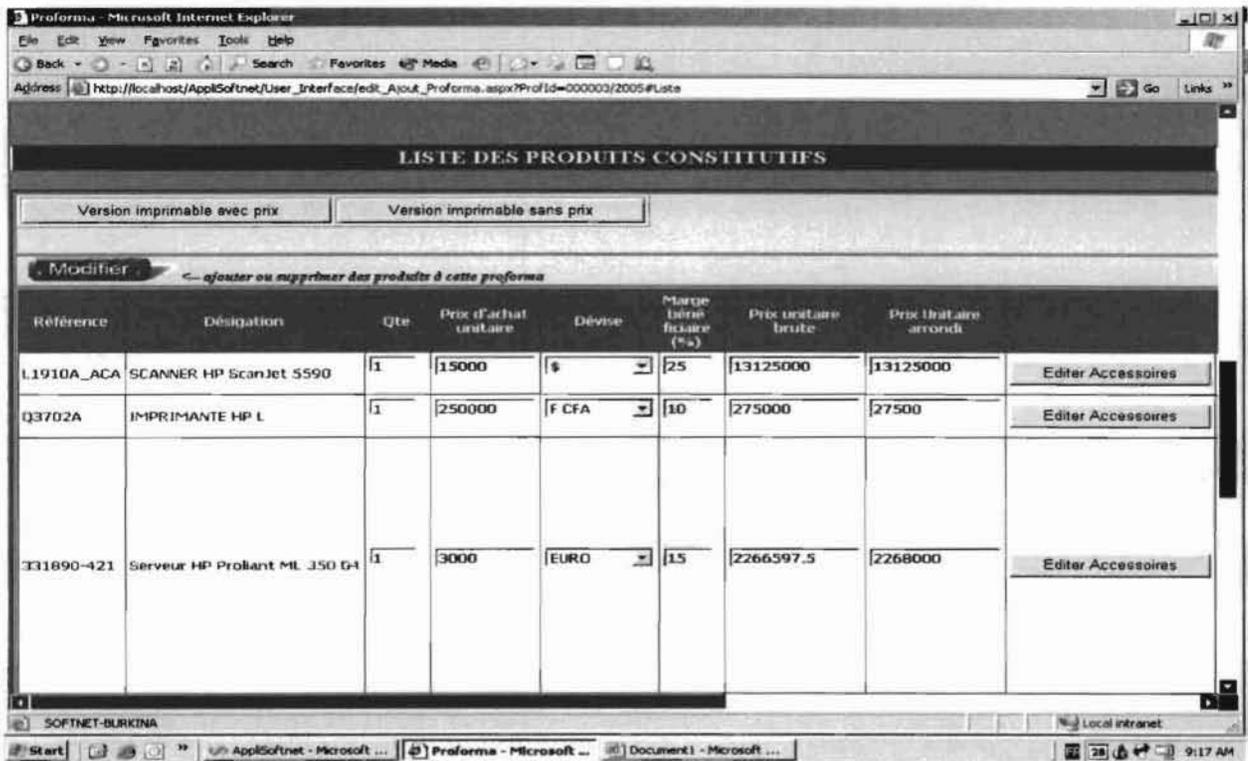


Figure VI.L: L'écran des produits constitutifs d'une proforma

Le bouton « *Éditer accessoires* » présente la page suivante permettant de lier ou de supprimer des accessoires à un produit.

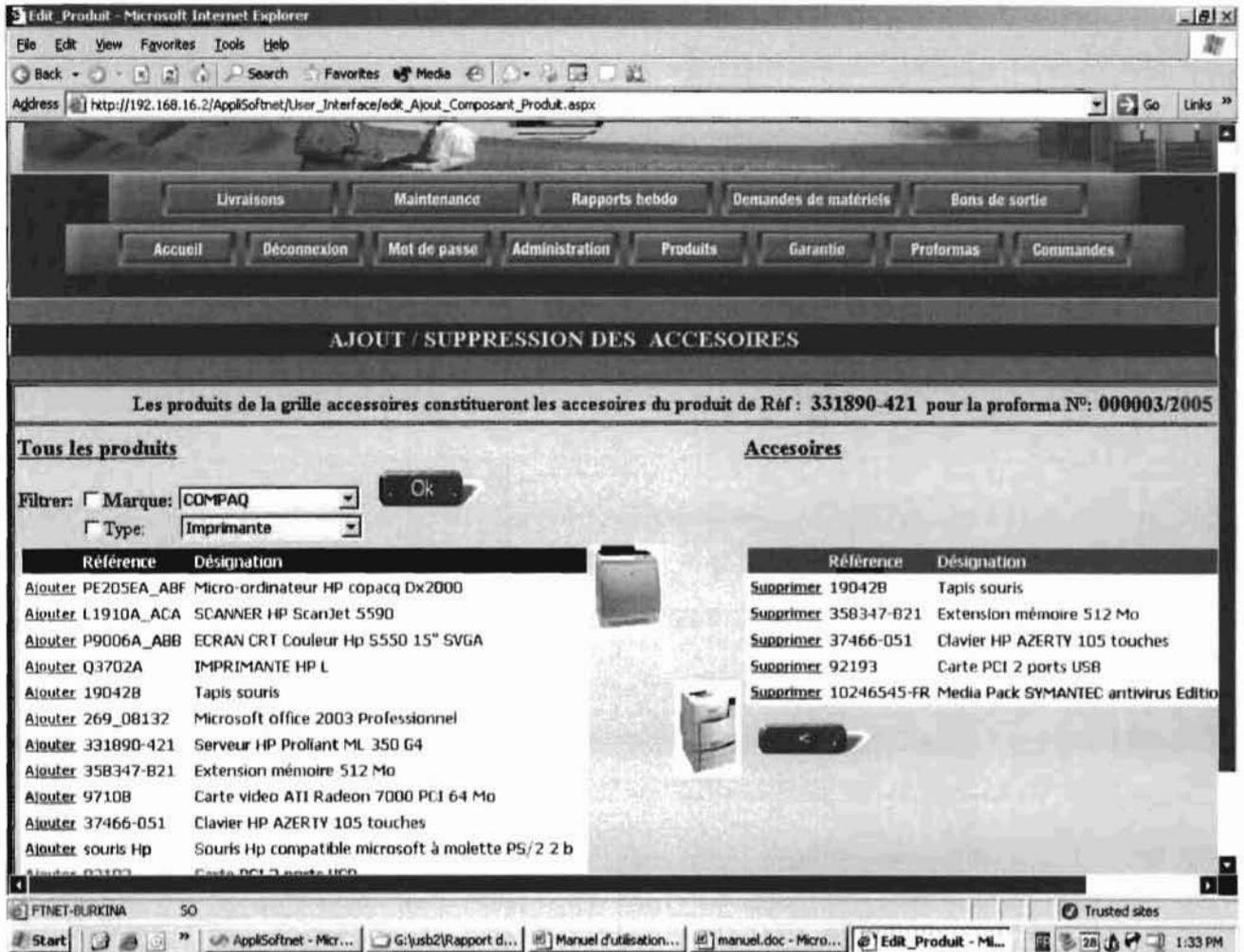


Figure VI.M: La page d'ajout / suppression d'accessoires à un produit

Référence	désignation	Prix d'achat unitaire	Devise	Marge
19042B	Tapis souris	0	\$	0
35B347-B21	Extension mémoire 512 Mo	0	\$	0
37466-051	Clavier HP AZERTY 105 touches	0	\$	0
Q2193	Carte PCI 2 ports USB	0	\$	0
10246545-FR	Media Pack SYMANTEC antivirus Edition Corporate 9,10246545-FR	0	\$	0

Figure VI.N : Quelques accessoires liés à un produit

Référence	Désignation	Qte	Prix d'achat unitaire	Devise	Taux de bénéfice financier (%)	Prix unitaire brute	Prix unitaire arrondi
L1910A_ACA	SCANNER HP ScanJet 5590	1	15000	\$	25	13125000	13125000
Q3702A	IMPRIMANTE HP L	1	250000	F CFA	10	275000	275000
331890-421	Serveur HP Proliant ML 350 G4	1	3000	EURO	15	2266597.5	2268000

Total HT:	15420500
TVA (18%):	2775690
BIC (02%):	363923.8
Total TTC:	18560113.8

Figure VI.O : Les Taxes et totaux d'une proforma calculés automatiquement

VI.2.6. LES PAGES DE LIVRAISON DE MATERIEL



Figure VI.P: L'écran des informations sommaires d'une livraison

Le lien « ici » affiche la liste des produits livrés et leurs numéros de série.

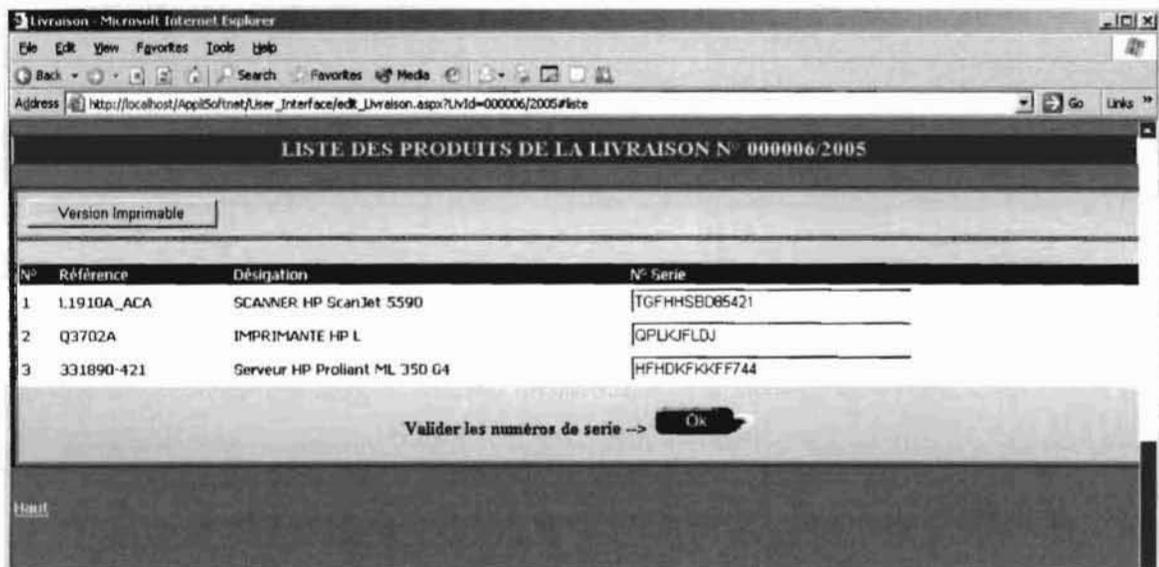


Figure VI.Q: La liste des produits d'une livraison

VI.2.7. LES PAGES DE LA MAINTENANCE DE MATERIELS

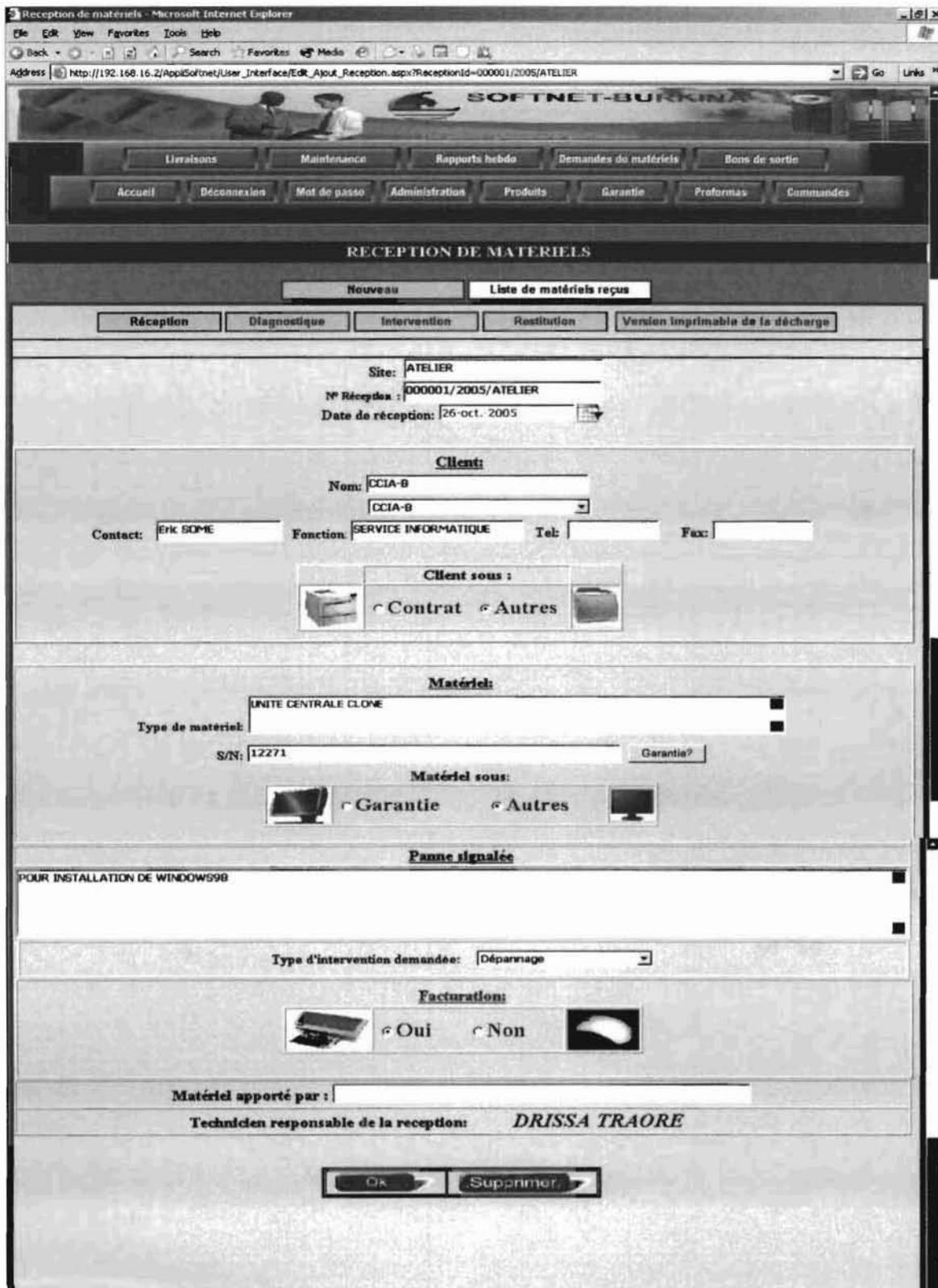


Figure VI.R : La page de réception de matériel en panne

Reception de matériels - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media Go Links

Address http://localhost/AppliSoftnet/User_Interface/Edit_Diagnostique.aspx?ReceptionId=000016/2005/ATELIER

SOFTNET-BURKINA

Livrations Maintenance Rapports hebdo Demandes de matériels Bons de sortie

Accueil Déconnexion Mot de passe Administration Produits Garantie Proformas Commandes

DIAGNOSTIQUE DE MATERIELS

Liste de matériels reçus

Réception Diagnostique Intervention Restitution

Site: ATELIER N° Réception : 000016/2005/ATELIER

Client: CREPA Date de réception : 11/11/2005

Matériels S/N: C2C4391HW3 Type de matériel : LIC COMPAQ DX2000

Diagnostique

SATURATION DU DISQUE DUR



Conclusion:

Commentaire:

Technicien responsable du diagnostic: **DRISSA TRAORE**

OK

Haut

SOFTNET-BURKINA Local intranet

start AppliSoftnet - Microsoft ... Reception de matéri... Document1 - Microsoft ... 28 9:31 AM

Figure VI.S: La page de diagnostic d'un matériel en panne

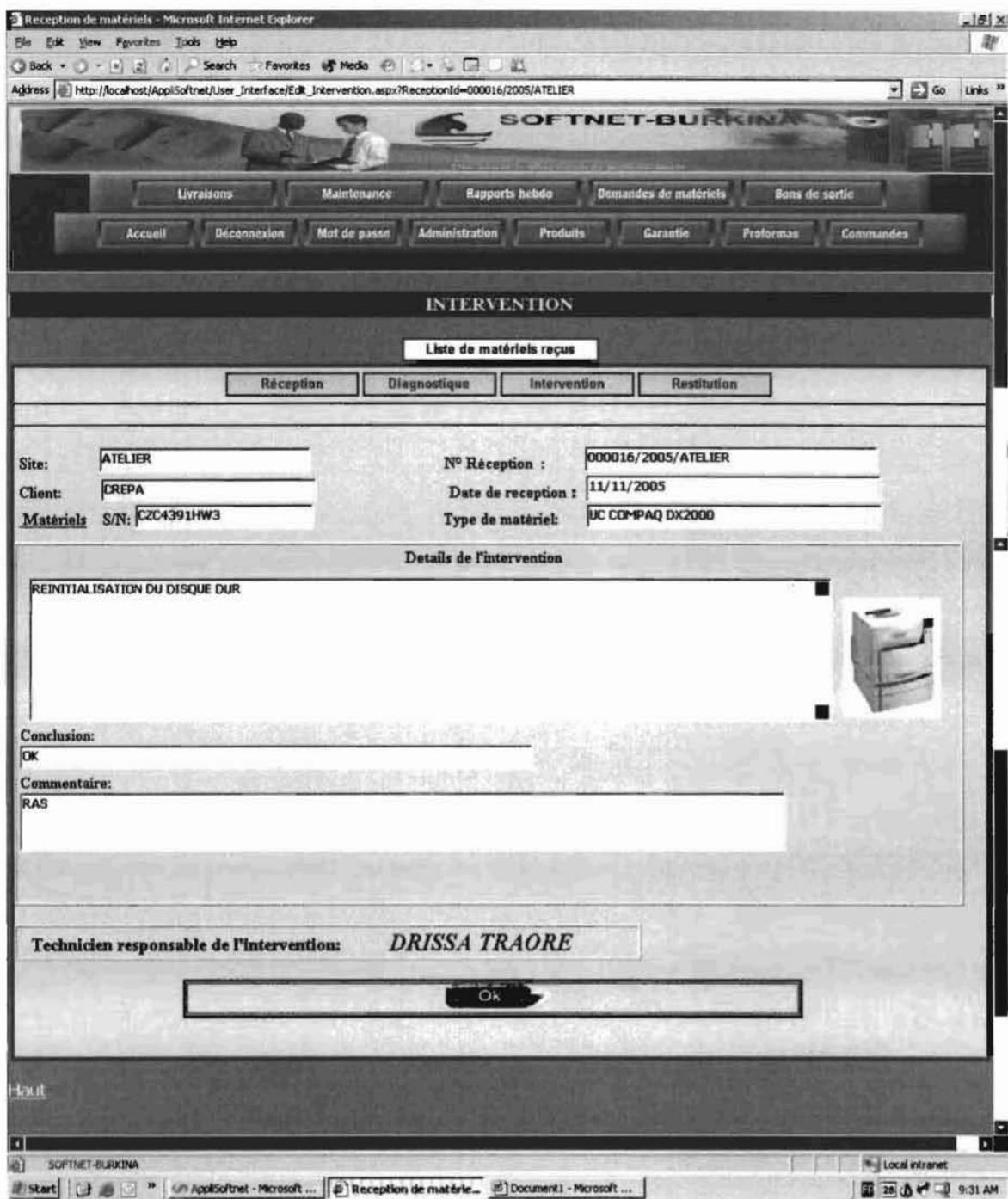


Figure VI.T: La page des interventions sur un matériel en panne

VI.2.8. LES PAGES DE DEMANDE DE MATERIEL

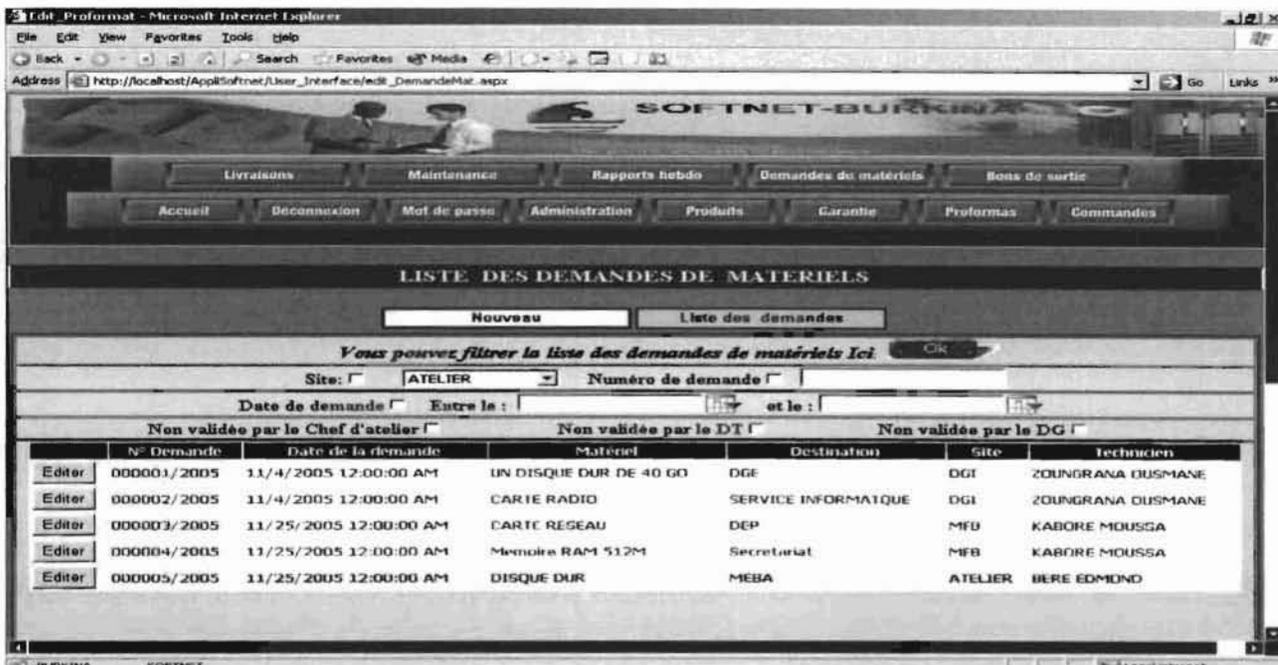


Figure VI.U: La liste des demandes de matériel.

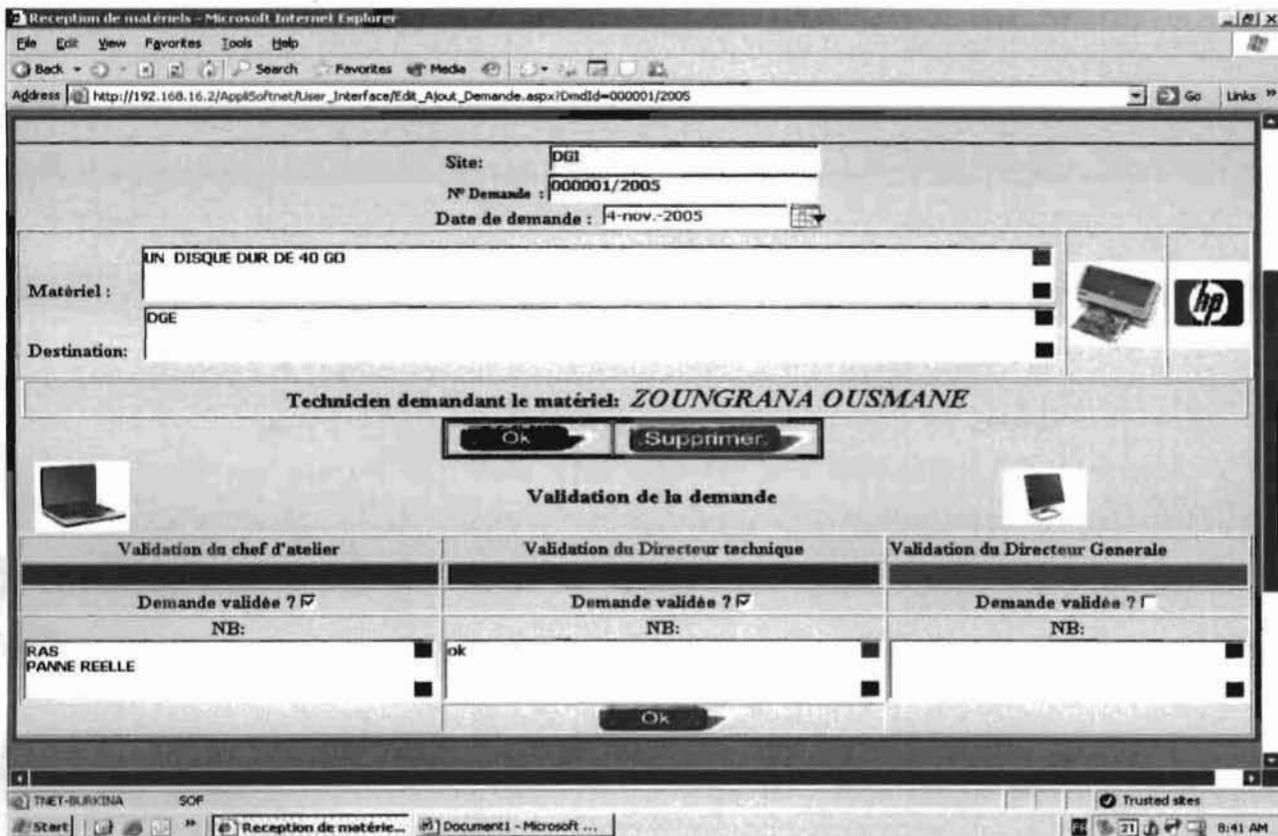


Figure VI.V: La page d'établissement et de validation d'une demande de matériel.

VI.2.9. LA PAGE DES RAPPORTS HEBDOMADAIRES

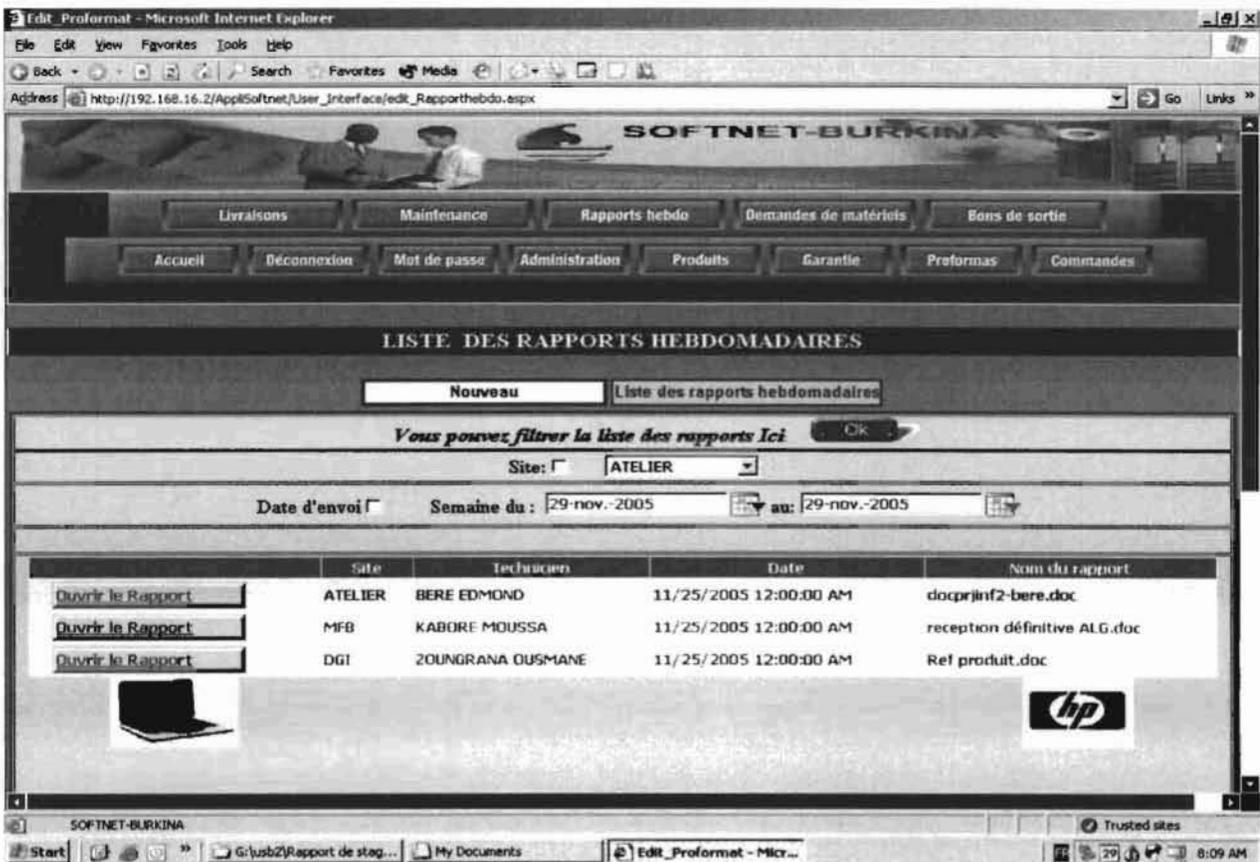


Figure VI.W: la page des rapports hebdomadaires

CONCLUSION

Nous avons conçu et développé une application web pour le suivi des prestations de SOFTNET-BURKINA. L'application simplifie l'activité des services technique et commercial et rend les prises de décision faciles à tous les niveaux.

Sur le plan professionnel, ce projet nous a été une bonne occasion pour développer une telle application et de maîtriser pas mal de technologies. Sur le plan humain, il nous a permis d'approfondir nos connaissances et nous a été une véritable occasion pour voir de près la nature, l'aspect et le déroulement de la vie professionnelle.

Nous avons fait de notre mieux pour laisser au sein de SOFTNET-BURKINA une bonne image de École Supérieure d'Informatique où nous sommes issus et présenter un travail digne des enseignements que nous avons reçus.

L'application n'est pas complètement au point car il reste à interopérer avec le système de gestion de stock qui jusqu'à présent n'est pas encore à notre disposition. De ce fait, le stage prend fin mais le projet continue.

BIBLIOGRAPHIE

Developing Web Applications with Visual Basic .NET and ASP.NET by John Alexander and Billy Hollis.
ASP.NET Bible by Bill Evjen.

SOURCES INTERNET:

<http://www.asp-magazine.com>
<http://www.labo-dotnet.com>
www.microsoft.com/france/asp.net/default.aspx
<http://www.microsoft.com/france/msdn/technos/asp.aspx>
<http://www.aspfr.com>
<http://www.4guysfromrolla.com>
<http://www.asp-php.net>

ANNEXE



SOFTNET - BURKINA

SERVICES ET INTEGRATION DE RESEAUX - FORMATION

1017, Av. Kwamé N'KRUMAH
11 BP : 1018 CMS OUAGA 11
Tél. : (226) 50 30 19 41
Fax. : (226) 50 30 19 42
Email : softnet@fasonet.bf

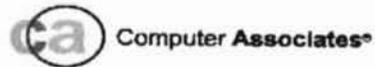
SA ou Capital de : 10.000.000 FCFA
N° de compte 100 923 801 019 ECOBANK
N° de compte 09053 011832 001 BICIA-B
N° de compte 0122501 000 31 BCB
N° de compte 11 3802 0040149 SGBB
Ouagadougou - BURKINA FASO
IFU N° 72308055U
RCCM BF 2003 81798
N° CNSS : 32717 M
Agrément Fournisseur de l'Etat N° 139/2004
Division Fiscol : Kadiogo I
Régime : Réel Normal

DECHARGE DE RECEPTION DE MATERIELS

N° 000001/2005/ATELIER

Site: ATELIER

NON DU CLIENT: CCIA-B		Contact: Eric SOME		Fonction: SERVICE INFORMATIQUE	
Tél: Fax:					
TYPE DE MATERIEL UNITE CENTRALE CLONE			S/N: 12271		
TYPE D'INTERVENTION	Dépannage	X	FACTURATION	OUI	X
	Mise en service			NON	
	Expertise		CLIENT SOUS	Contrat	
	Maintenance			Garantie	
Autres		Autres	X		
PANNE DECLAREE PAR LE CLIENT					
POUR INSTALLATION DE WINDOWS98					
Pour SOFTNET			Pour le client		
Nom: DRISSA TRAORE			Nom:		
Date et signature:			Date et signature:		





SOFTNET - BURKINA

SERVICES ET INTEGRATION DE RESEAUX - FORMATION

1017, Av. Kwamé N'KRUMAH
11 BP : 1018 CMS OUAGA 11
Tél. : (226) 50 30 19 41
Fax. : (226) 50 30 19 42
E-mail : softnet@fasonet.bf

PROFORMA

SA ou Capital de : 10.000.000 FCFA
N° de compte 100 923 801 019 ECOBANK
N° de compte 09053 011832 001 BICIA-B
N° de compte 0122501 000 31 BCB
N° de compte 11 3802 0040149 SGBB
Ouagadougou - BURKINA FASO
IFU N° 72308055U
RCCM BF 2003 81798
N° CNSS : 32717 M
Agrément Fournisseur de l'Etat N° 139/2004
Division Fiscale : Kadiogo I
Régime : Réel Normal

Client: ECO BANK

N°: 000002/2005

Adresse:

Date: 04/11/2005

Cher client,
nous avons bien reçu votre demande de proforma et nous vous en remercions.
Nous vous prions de trouver ci-dessous nos conditions les meilleures.

Reference	Designation	Qte	Prix Unitaire	Total
L1910A_ACA	SCANNER HP ScanJet 5590 scanner à plat couleur vitesse daperçu 7 seconde Résolution de numérisation 2400 pp profondeur 48 bit Chargeur automatique de 50 feuilles Interface USB 2.0	1	172 500	172 500
PE205EA_ABF	Micro-ordinateur HP copacq Dx2000 processeur intel IV Ghz Memoire vive 512 Mo DDR RAM extensible à 4 Go Memoire cache 512 Ko Disque dur 80 Go Carte son soundMax Digital AC97 + haut parleur Lecteur de disquette 3" 1/2 1.44 Mo Lecteur Combo Lecteur CD/ GRAVEUR CD/DVD ROM Carte réseau Fast Ethernet 10/100 Rj 45	2	1 986 000	3 972 000
19042B	Tapis souris			
358347-B21	Extension mémoire 512 Mo			
97108	Carte video ATI Radeon 7000 PCI 64 Mo			
37466-051	Clavier HP AZERTY 105 touches			

Reference	Designation	Qte	Prix Unitaire	Total
Q3702A	IMPRIMANTE HP L Technologie laser couleur vitesse 19 pages / mn en noir , 4 pages/mn couleur resolution 600*600 ppp Memoire 64 Mo extensible à 192 Mo Interface Parallele et USB Bac 125 feuilles Cartouche Tonner Préinstallée Livré avec pilotes	3	89 000	267 000

TOTAL HT:	4 411 500 F CFA
TVA (18%):	794 070 F CFA
BIC (02%):	104 111 F CFA
TOAL TTC :	5 309 681 F CFA



SOFTNET - BURKINA

SERVICES ET INTEGRATION DE RESEAUX - FORMATION

SA au Capital de : 10.000.000 FCFA
 N° de compte 100 923 801 019 ECOBANK
 N° de compte 09053 011832 001 BICIA-B
 N° de compte 0122501 000 31 BCB
 N° de compte 11 3802 0040149 SGBB
 Ouagadougou - BURKINA FASO
 IFU N° 72308055U
 RCCM BF 2003 81798
 N° CNSS : 32717 M
 Agrément Fournisseur de l'Etat N° 139/2004
 Division Fiscol : Kadiogo I
 Régime : Réel Normal

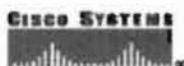
1017, Av. Kwamé N'KRUMAH
 11 BP : 1018 CMS OUAGA 11
 Tél. : (226) 50 30 19 41
 Fax. : (226) 50 30 19 42
 E-mail : softnet@fasonet.bf

Rapport d'intervention

Site: ATELIER

N° 000001/2005/ATELIER

NON DU CLIENT: CCIA-B		Contact: Eric SOME		Fonction: SERVICE INFORMATIQUE	
Tél:		Fax:			
TYPE DE MATERIEL UNITE CENTRALE CLONE				S/N: 12271	
TYPE D'INTERVENTION	Dépannage	X	FACTURATION	OUI	X
	Mise en service			NON	
	Expertise		CLIENT SOUS	Contrat	
	Maintenance			Garantie	
	Autres			Autres	X
PANNE DECLAREE PAR LE CLIENT			DIAGNOSTIQUE		
POUR INSTALLATION DE WINDOWS98			ABSENCE DE SYSTEME D'EXPLOITATION		
DETAILS DE L'INTERVENTION INSTALLATION DE WINDOWS98 ET DE MSOFFICE2000 ENTRETIEN GENERAL				OBSERVATION DU CLIENT	
Conclusion RAS					
Date et heure de debut de l'intervention Le 31/10/2005 à 06:00			Date et heure de fin de l'intervention Le 04/10/2005 à 06:00		
Pour SOFTNET Nom: DRISSA TRAORE Date et signature:			Pour le client Nom: ERIC SOME Date et signature:		



Computer Associates®