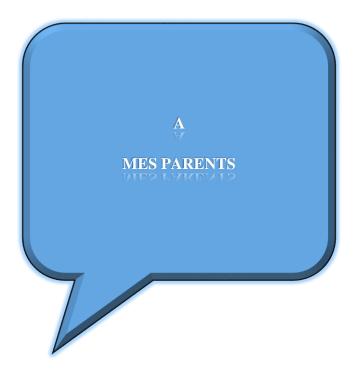
# **Sommaire**

Sommaire	1
Dédicace	Erreur! Signet non défini.i
Remerciements	iii
Liste des figures	iv
Listes des tableaux	v
Liste des abreviations	vi
Résumé	Erreur! Signet non défini.
Abstract	viii
Avant-propos	Erreur! Signet non défini.
INTRODUCTION GENERALE	1
Chapitre 1: PRESENTATION DE L'ENTREPRISE	2
SECTION 1 : ENVIRONNEMENT INTERNE	2
SECTION 2 : DEROULEMENT DU STAGE	3
SECTION 1 : ETUDE DE L'EXISTANT	5
SECTION 2 : CAHIER DE CHARGE	6
CHAPITRE 3 : MODELISATION DETAILLE	
SECTION 1 : PRESENTATION DU LANGAGE DE MO	DDELISATION UTILISE 15
SECTION 2 : MODELISATION DU FUTUR SYSTEMI	E16
CHAPITRE 4: IMPLEMENTATION	31
CONCLUSION GENERALE	37
BIBLIOGRAPHIE	38
Table des matières	39

# **Dédicace**



# **Remerciements**

Nous profitons par le biais de ce rapport pour exprimer nos sincères remerciements à toutes les personnes qui ont contribué de façon morale, physique, et intellectuel à l'élaboration de ce travail. De ce fait notre réflexion va à l'endroit de :

- ➤ Pr. TAVEA Fréderic Marie, promoteur de l'ISSI, pour tous les moyens consentis dans le but d'effectuer mon parcours scolaire dans de meilleures conditions
- ➤ Pr. KENMOGNE Sidonie Beatrice, Directrice de l'ISSI, pour tous les moyens consentis dans le but d'effectuer mon parcours scolaire dans de bonnes conditions
- ➤ M. KAMWA Michael, notre encadreur académique, qui a su m'encadrer et m'orienter vers les meilleures voies en vue de la réalisation du présent document
- ➤ M. DOUMI Serge Bruno pour les divers conseils, encadrement et le suivi qu'il a fait envers ma personne
- M. HOUDJEU Donald, pour m'avoir soutenu et m'aider avec les idées dans le cadre de mon rapport
- ➤ L'Administration de notre Institut sous la direction de Madame Tavea Sidonie pour les conseils, l'appui et l'accompagnement dans notre processus de formation avec un corps enseignant qualifié, professionnel et proactif
- ➤ Les enseignants de ISSI pour leur encouragement et les différents enseignements prodigués durant ces deux années académiques
- ➤ M. WAFO Julio, directeur de Hi-Tech vision, pour avoir eu à m'intégrer au sein de son entreprise et qui a toujours été disponible à mon égard
- ➤ M. FEUSSI Woguem Thierry Darid encadreur professionnel, pour m'avoir donné les différents conseils et m'orienter sur la réalisation et la conception de mon application web
- Le personnel de Hi-Tech vision, pour m'avoir conseillé et aidé à mieux m'y connaitre dans le monde professionnel
- ➤ Mes frères et mes sœurs, pour m'avoir soutenues, ma mère chérie pour les conseils prodigués, Ma grand-mère et mon grand-père chérie pour leurs conseils et tous ces instants passés auprès d'eux
- ➤ Mes camarades de promotion pour leur soutien moral et psychologique ainsi que l'air qui nous animait
- Mes amies pour m'avoir soutenu dans les pires moments, et m'avoir conseillé

A tous ceux qui ont participé de près ou de loin dans l'élaboration de ce travail, nous leur disons également merci.

# Liste des figures

FIGURE 1 : ORGANIGRAMME DE HI-TECH VISION	2
FIGURE 2: DIAGRAMME DE GANTT	12
FIGURE 3 : DIAGRAMME DE CAS D'UTILISATION	18
FIGURE 4 : DIAGRAMME DE CLASSE DE GESTION DES STOCKS	22
FIGURE 5 : MESSAGE SYNCHRONE ET ASYNCHRONE	27
FIGURE 6 : DIAGRAMME DE SEQUENCE CAS D'AUTHENTIFICATION	29
FIGURE 7: DIAGRAMME DE SEQUENCE CAS D'UTILISATION	30
FIGURE 8 : PROCESSUS DE CREATION DE LA BASE DE DONNEE GESTION DE STOCKS	34
FIGURE 9 : INTERFACE DE CONNEXION	34
FIGURE 10 : INTERFACE DU MENU PRINCIPAL	35
FIGURE 11 : INTERFACE D'ENREGISTREMENT D'UN ARTICLE	35

# Listes des tableaux

TABLEAU 1 : CHRONOGRAMME DES ACTIVITES DU STAGE	4
TABLEAU 2 : RESPONSABILITES DES PARTIES ENGAGEES	9
TABLEAU 3 : CONTRAINTE FINANCIERES	10
TABLEAU 4 : CONTRAINTES HUMAINES	11
TABLEAU 5 : CONTRAINTES TEMPORELLES	11
TABLEAU 6 : DICTIONNAIRE DES DONNEES	32

# Liste des abréviations

SIGLES	SIGNIFICATIONS
CSS	Cascading Style Sheet
HML	Hypertext Make up Language
IT	Technologie de l'Information
	Méthode d'Etude et de Réalisation
MERISE	Informatique pour les Systèmes d'Entreprise
MOE	Maitre d'Oeuvre
MOA	Maitre d'Ouvrage
PHP	Php Hypertext Preprocessor
RCCM	Registre du Commerce et du Crédit Mobilier
SQL	Structured Query Language
UML	Unified Modeling Language

# Résumé

Dans le cadre de notre formation en **génie logiciel**, nous avons effectué un stage académique de deux mois au sein de, l'entreprise Hi-Tech vision, entreprise spécialisé dans le développement d'application web, mobile, Windows; dans l'audit, la vente du matériel informatique ainsi que dans la formation du personnel. Tout au long de notre stage au sein de cette structure, nous avons travaillé sur un projet innovateur. Ce dernier est d'une grande envergue et très vaste. L'objectif de ce stage était donc la conception et la réalisation d'une application web de gestion de stock du matériel informatique. Ce projet est effectué après une longue période de recherche sur la gestion de stocks et ses différentes phases, dans le but de préparer pour l'entreprise une solution complète qui répond à ses attentes. Cette application intègre un grand nombre de fonctionnalité entre autre la gestion des clients. Concernant le développement du projet, le choix s'est sur la méthodologie...et la modélisation du système est réalisée grâce au langage UML (Unified Modeling Language). Ce rapport présente les différentes étapes pour aboutir à la réalisation et la mise en place de cette solution.

# **Abstract**

As part of out training in software engineering. We did a two-month academic intership at **HITECH VISION**, a company specialising in the developpement of web, mobile and Windows applications, in the audit, the sale of computer equipements well as the training of personnel. Throughout our stay in this structure, we worked on an innovative project. The latter is farreaching and very extensive. The objective of this intership was therefore **the design and production of a web application for management of stock of computer equipements**. This project is carried out after a long period of research on stock management and its differents phases, white the aim of preparing a complete solution for the company that meets its expectations. This application integrates a large number of features, including customers management. Concerning the choice is fixed on the methodology...and the modeling of the system is carried out thanks to the UML (Unified Modeling Language) language. This report present the differents steps to achieve the realization and implementation of this solution.

# **Avant-propos**

Dans le but d'assurer un développement durable et de fournir aux entreprises une main d'œuvre compétente et compétitive dans divers domaines, le gouvernement camerounais par le biais du ministre de l'enseignement supérieur a permis l'ouverture des instituts privées d'enseignements supérieurs. Donnant ainsi l'opportunité aux institutions privées, de contribuer à l'acquisition d'une formation académique et professionnel en adéquation avec le monde professionnel.

C'est ainsi qu'est créé l'ISSI (Institut Supérieur des Sciences et de l'Industrie) par arrêté N° 14/0420/MINISUP du 9 juillet 2014. Situé dans la région du littoral, département du Wouri et de l'arrondissement de douala 3°.

L'institut supérieur des sciences de l'industrie forme dans les filières suivantes :

# **CYCLE BTS**

#### **♣** Commerce-Vente

- Commerce international
- Marketing commerce vente

#### Gestion

- Assistant manager
- Banque et finance
- Comptabilité et gestion des entreprises
- Gestions des projets
- Gestions des ressources humaines
- Gestions des systèmes d'information
- Micro finance

# Carrière juridique

- Douane et transit
- Gestions fiscales

# Génie informatique

- Génie logiciel
- Maintenance des systèmes informatiques
- E-commerce et marketing numérique

# **♣** Tourisme, hôtellerie, et restauration

- Hôtellerie
- Restauration

# <<Conception et réalisation d'une application web de gestion des stocks du matériel informatique : cas de Hi-Tech Vision>>

- Tourisme et loisirs
- **♣** Science de l'éducation
- **Letude médico-sanitaire** 
  - Science infirmière
  - Opticien lunetier

# INTRODUCTION GENERALE

Aujourd'hui, l'informatique a pris de l'ampleur dans le monde en facilitant la tâche aux entreprises. Il participe aux besoins de l'entreprise tout en évitant la perte des données. Au pare avant, les entreprises utilisaient la méthode traditionnelle mais au fil du temps avec l'avènement des technologies de l'informatique et de la communication elles se sont modernisés pour mieux gérer leurs stocks et la sauvegarde de leurs données. Elles ont su faire profit de cette technologie pour mieux leurs données et s'arrimer à l'ère qui est celui de la digitalisation. Ainsi, nous avons mis sur pieds durant notre période de stage académique une application de gestion de stocks qui nous a permis de remédier aux différends des entreprises tels que : les stocks insuffisants qui peuvent entrainer le mécontentement des clients, le coût de l'entreposage et de l'assurance, les retards dans les délais de préparation et de livraison, la mauvaise gestion des commandes et un espace d'entreposage mal organisé. La gestion des stocks consiste à effectuer le suivi des stocks depuis les fabricants aux entrepôts puis jusqu'au point de vente. Elle a pour objectif de disposer des bons produits au bon endroit et au bon moment. Cela nécessite une visibilité des stocks : savoir quand passer une nouvelle commande, combien commander et où entreposer le stock. Elle s'accompagne inévitablement de la recherche de la minimisation des coûts des produits laquelle dépend d'une bonne politique d'approvisionnement, d'une gestion rationnelle et adéquat des stocks. Cependant, c'est en considérant l'importance de cette discipline qu'est la gestion des stocks que nous avons décidés d'étudier cette notion à travers l'étude de cas d'une structure commerciale grâce au thème : LA GESTION DES STOCKS DU MATERIEL INFORMATIQUE: CAS DE HI-TECH VISION. Dès lors, notre présente réflexion mettra l'accent sur deux parties :

**PARTIE** I : présentation de l'entreprise et études préalables

CHAPITRE 1 : présentation de l'entreprise cas de HI-TECH VISION

**CHAPITRE 2**: études préalables

PARTIE II : modélisation détaillée et implémentation

**CHAPITRE 3**: modélisation détaillée

**CHAPITRE 4**: implémentation

# **Chapitre 1: PRESENTATION DE L'ENTREPRISE**

#### **SECTION 1: ENVIRONNEMENT INTERNE**

## I. ENVIRONNEMENT INTERNE

# 1. HISTORIQUE ET EVOLUTION

# a) Présentation

HI-TECH VISION est une entreprise de conception et mise en place de solution informatique situé à 500m de NDOKOTI face UIES/INSAM dans la ville de douala. C'est une agence digitale spécialisée dans la prestation de services informatiques et solutions numériques. Son but est de réaliser des produits et services informatiques pour apporter de la productivité et de l'efficience dans la réalisation des activités de la clientèle.

#### b) **EVOLUTION**

Depuis **2013 HI-TECH VISION** est dirigé par **M. WAFO JULIO**. Son siège principal est à **NDOGSIMBI**. Depuis sa création, elle a beaucoup évolué. Elle conçoit des applications de développement web à l'intérieur comme l'extérieur du pays. Elle a conçu son tout nouveau logo pour refléter son engagement envers l'innovation et l'excellence en matière de prestation de services informatiques.

#### 2. ORGANISATION ET FONCTIONNEMENT

#### Organisation

L'organisation des entreprises repose sur un ensemble de caractéristiques. Celle-ci se fait par la concurrence, la compétition, la bonne matière et la qualité du produit. L'activité de cette entreprise est repartie au sein d'un ensemble de fonction à savoir : la production et le commerce

#### a. Attribution

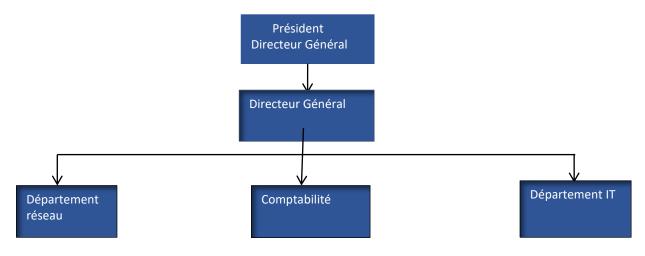


FIGURE 1: ORGANIGRAMME DE HI-TECH VISION

#### • Fonctionnement

# a) Moyens matériels

La structure **HI-TECH VISION** s'est munie d'un ensemble d'équipements et logiciels permettant un suivi agréable au sein de l'entreprise dans le but de faire un travail cohérent et satisfaisant.

#### b) Moyens humains

**HI-TECH VISION** est une entreprise de service du numérique qui se positionne de manière concurrentielle sur le secteur des nouvelles technologies et informatiques sur le territoire national camerounais dans des domaines variés tel que :

- La conception des sites internet ;
- La conception des applications ;
- Réseaux et télécommunications :
- Maintenance informatique et réseaux ;
- Sécurité électronique (vidéo surveillance, contrôle d'accès);
- Sécurité informatique ;

#### **SECTION 2: DEROULEMENT DU STAGE**

#### I. PRESENTATION DU DEPARTEMENT IT

#### 1. LES OBJECTIFS ET SERVICES DU DEPARTEMENT IT

# a) Objectifs du département IT

Le département IT a pour but de réaliser des produits et services informatiques pour apporter de la productivité et de l'efficience dans la réalisation des activités de la clientèle, apportant ainsi l'assurance de la satisfaction clientèle. Il a la ferme volonté de confirmer son rôle d'initiateur de tendance technologique, porté par l'excellence et le d'initiateur de tendance technologique, porté par l'excellence et le d'une grande intégrité et d'un sens aigu de l'innovation.

#### b) Services du département IT

L'entreprise met à la disposition de la clientèle des services divers :

- La conception des sites internet ;
- La conception des applications ;
- Réseaux et télécommunications ;
- Maintenance informatique et réseaux ;
- Sécurité électronique (vidéo surveillance, contrôle d'accès) ;

# <Conception et réalisation d'une application web de gestion des stocks du matériel informatique : cas de Hi-Tech Vision>>

- Sécurité informatique ;
- Formation en informatique;
- Marketing digital;

# II. CHRONOGRAMME DES ACTIVITES

TABLEAU 1: CHRONOGRAMME DES ACTIVITES DU STAGE

PERIODES	SERVICES	ACTIVITES		
Du 1 <sup>er</sup> au 10 juillet	Support	<ul> <li>Découverte de service de l'entreprise et son environnement</li> <li>Prise de connaissance</li> </ul>		
Du 13 au 17 juillet	Support	<ul><li>Maintenance des pc</li><li>Attribution des thèmes de stage</li></ul>		
Du 19 au 22 juillet	Support	<ul> <li>Installation et paramétrage de serveurs XAMPP</li> <li>Installation des applications nécessaires pour le projet</li> </ul>		
Du 25 juillet au 5 août	Analyse	<ul> <li>Collecte des informations</li> <li>Modélisation du projet</li> <li>Analyse des informations collectées</li> </ul>		
Du 6 au 19 août	Développement	<ul> <li>Acquisition des technologies à utiliser</li> <li>Implémentation du projet</li> </ul>		
Du 22 au 31 août	Développement	<ul> <li>Implémentation sur le modèle à utiliser : MERISE</li> <li>Modélisation du projet</li> </ul>		

# **CONCLUSION**

En somme il était question pour nous de présenter l'entreprise où nous avons effectués le stage. Ainsi nous avons montré d'une première part l'environnement interne et d'une seconde part le déroulement du stage. Cependant nous allons parler de **l'étude préalable** au chapitre deux.

# **CHAPITRE 2: ETUDE PREALABLE**

L'étude préalable est une étape qui consiste principalement à recenser l'existant. En d'autre terme les solutions informatiques déjà mises en œuvre dans l'entreprise et à recenser les besoins notamment en terme de fonctionnalités nouvelles. Il peut s'agit d'étudier la pertinence du projet envisagé, sa fiabilité. Dès lors la présente réflexion mettra l'accent sur quatre volets à savoir : l'étude d'opportunité, l'élaboration du cahier de charge, la critique de l'existante et le projet d'automatisation.

# **SECTION 1**: ETUDE DE L'EXISTANT

#### I. DESCRIPTION DE L'EXISTANT

Nous avons constaté que les entreprises n'arrivent pas à gérer leurs stocks à cause d'une mauvaise organisation dans les entrepôts. Une application de gestion des stocks consiste à effectuer le suivi des stocks depuis la fabrication jusqu'au point de vente en passant par les entrepôts. Cependant, après fabrication des produits dans les usines, ils sont transportés dans les magasins où ils seront stockés (rangé et identifié) ensuite elles seront transférées dans les points de vente sous commande. Cela nécessite un suivi à savoir : quand penser à passer une nouvelle commande, combien commander et où entreposer le stock. Elle s'accompagne inévitablement de la recherche de la minimisation des coûts des produits laquelle dépend d'une bonne politique d'approvisionnement, d'une gestion rationnelle et adéquat des stocks. Ce pendant l'application consiste à mieux faciliter la gestion des stocks pour ne plus être en rupture tout en connaissant les produits entrant et sortant ainsi que celle ajouter. Pour créer cette application nous avons procédé de la manière suivante :

- Analyse des besoins ;
- Spécification des besoins ;
- Conception détaillé;
- Implémentation;
- Vérification et validation ;
- Intégration et test ;
- Maintenance;

# II. CRITIQUE DE L'EXISTANT

#### 1. Sur le plan fonctionnel

Durant notre stage au sein de l'entreprise **Hi-Tech vision** nous avons fait un constat de surcharge de travail au sein du personnel car il y'avait peut d'employé et ces employés

n'arrivaient pas tous le même jour ce qui faisait qu'on se retrouvait parfois avec un employé qui effectuait toute les tâches de l'entreprise.

# 2. Sur le plan organisationnel

Pendant notre stage à **Hi-Tech vision**, nous avons constaté que le système organisationnel de l'entreprise n'était pas correct comme toute autre entreprise mais pour se rattraper, ils ont dû travailler en parfaite la collaboration avec les autres employés pour booster l'évolution de l'entreprise.

#### III. SOLUTION RETENUE

Pour pallier aux problèmes rencontrés en entreprise, nous devons :

- Eviter la surcharge au sein du personnel;
- Sécuriser les données ;

#### **SECTION 2**: CAHIER DE CHARGE

#### I. ELABORATION DU CAHIER DE CHARGE

#### 1. **DEFINITION**

Un cahier de charge est un document qui permet de comprendre et expliquer un projet dans l'ensemble, avec toute les contraintes, les besoins et les objectifs. Autrement dit c'est la base de de conception d'un projet technique en indiquant tous les éléments qui devons être pris en compte. Ainsi il jaillit de cette définition les points suivants à développer tels que :

Les fonctionnalités à développer, les contraintes techniques, un état de l'existant, une explication des besoins, le budget, la liste des intervenants et leurs rôles respectifs.

# 2. DESCRIPTION DE LA SITUATION ACTUELLE (ETUDE DE L'EXISTANT)

L'étude de l'existant permet de comprendre la nature du système actuel, de décrire la solution présente du domaine d'étude. Elle a pour but de rechercher les points forts et ceux faibles du système existant. Cependant, l'analyse de l'existant fait état des lieux du système actuel. L'entreprise stockent les produits dans le magasin puis scindent ces produits dans les différents points de vente ensuite le client passe une commande de son produit et le fournisseur enregistre le client ainsi que le produit choisi. Le client (point de vente) fait une demande on vérifie si les éléments présents ainsi que leurs quantités sont vraiment en stock. Si oui le fournisseur effectue une préforma qu'il envoie au client et attend validation de celui-ci. Après validation le fournisseur réalise une facture et soumet au client à la livraison. Puis un bon de livraison lui est remis. Dans le cas où l'article commandé n'est pas en stock, celle-ci transmet un message au client l'informant de la rupture des stocks.

Les problèmes rencontrés :

- Difficulté à retrouver les clients dans le fichier car au fil du temps la liste devient longue;
- Difficulté à sortir l'état des dettes fournisseurs (tache manuelle) ;
- Difficulté à suivre fidèlement nos clients ;
- Difficulté à sortir l'état de nos créances vis-à-vis de nos clients ;

#### 3. DELIMITATION DU PERIMETRE DU PROJET

Le périmètre d'un projet est la phase de préparation du projet au cours de laquelle une liste des produits livrables, des affectations, des dépenses et des délais sont déterminés et enregistrés. L'énoncé du périmètre d'un projet défini les limites du projet, attribut des rôles à chaque membre de l'équipe et spécifie les processus de vérification et d'approbation du travail terminé. Dès lors le périmètre de notre projet se résume à la :

- Gestion des clients et fournisseurs ;
- Gestion des commandes, produits et livraisons ;
- La rédaction du manuel d'utilisation ;
- La gestion des entrées et sorties ;

# 4. DELIMITATION DU SYSTEME D'INFORMATION

Un système d'information se définis comme étant un ensemble de ressource permettant de collecter, d'analyser, de stocker, de traiter, et de communiquer les informations de toute forme dans une organisation. Ici le système d'information soumis à notre étude est celui de la conception et de la réalisation d'une application web de gestion des stocks du matériel informatique dont la responsabilité revient á la structure Hi-Tech vision. Les objectifs de ce système sont : la collecte, l'analyse et le traitement des informations relative á tous les clients et les fournisseurs de l'entreprise.

#### 5. SPECIFICATION FONCTIONNEL DE L'APPLICATION

Les spécifications fonctionnelles désignent l'ensemble des tâches que vas réaliser l'application dans le but de remédier aux problèmes rencontrés. Dans le cadre de notre application, les spécifications fonctionnelles sont les suivants :

• Gestion des clients : elle vise à concevoir une interface utilisateur approprié permettant de gérer les clients (ajouter, modifier, supprimer les informations relatives attribué à un client), ainsi un client doit être caractérisé par : identifiant, nom, prénom, téléphone, ville.

- Gestion des fournisseurs : elle vise à concevoir une interface utilisateur conviviale permettant de gérer les fournisseurs (ajouter, modifier, supprimer un fournisseur) ainsi il est caractérisé par : identifiant, nom, prénom, téléphone, adresse
- Gestion des produits : a permet de concevoir une interface utilisateur ayant pour but de gérer les produits (ajouter, modifier, supprimer un produit) il se caractérise par : identifiant, nom du produit, prix, quantité
- Gestion des commandes: vise à concevoir une interface utilisateur conviviale nécessitant de gérer les commandes (ajouter, modifier, supprimer une commande) il est caractérisé par : identifiant, identifiant du produit, identifiant du fournisseur, quantité, date de la commande
- **Gestion des livraisons** : vise à concevoir une interface utilisateur permettant de gérer les livraisons (ajouter, modifier, supprimer une livraison). Elle est caractérisée par : identifiant livraison, identifiant commande, identifiant fournisseur, quantité, date de la livraison.

#### 6. SPECIFICATION NON FONTIONEL DE L'APPLICATION

Les spécifications non fonctionnelles désignent l'ensemble des besoins qui caractérise l'application les besoins en qualité d'efficacité, du type de matériel et de conception.

Une fois les besoins fonctionnels définies les besoins non fonctionnels doivent être pris en compte tout au long de la phase de développement des applications.

Dans le cadre de notre projet, les besoins non fonctionnels

- L'application doit être multi utilisateur c'est à dire elle doit être utilisée par plusieurs personnes simultanément;
- Elle doit être facile d'utilisation ;
- Fiable;
- Maintenanciable c'est-à-dire elle doit être capable d'être déployé sur plusieurs machines ;
- L'application doit être **ergonomique et conviviale** c'est-à-dire elle doit fournir une interface conviviale aux différents utilisateurs ;
- Intégrité ;

- **Sécurité** l'application doit assurer un minimum de sécurité pour toutes les informations traitées ;
- L'application doit être **latence** c'est-à-dire avoir un délai de réponse raisonnable ;
- Portable c'est-à-dire fonctionnelle sur n'importe quel ordinateur ;

# 7. RESPONSABILITE ET ROLES RESPECTIF DES PARTIES ENGAGEES PAR L'INTERVENTION

Dans cette partie du **cahier de charge** qui est cité et décrit de façon assez claire les parties prenantes lier au projet c'est-à-dire les personnes qui ont un intérêt quelconque dans le résultat du projet ; c'est généralement des membres de l'équipe du projet. Concernant notre projet, les parties prenantes et leurs rôles sont récapitulé dans le tableau suivant :

TABLEAU 2: RESPONSABILITES DES PARTIES ENGAGEES

Noms	Rôles	Description du rôle et niveau	Fonction
		d'intervention	
HI-TECH	Maitre	Il fournit et contrôle la pertinence	Contrôle et
VISION	d'ouvrage	des informations et veille à l'étroit	coordonne
	(MOA)	collaboration avec le fournisseur	l'avancée du
		d'information	projet
Mme	Maitre d'œuvre	Supervise l'évolution du projet	Développeur
DJOUMESSI	(MOE)	(phase de conception et réalisation	
HORLANE-		en particulier)	
JESICA			

# 8. EVALUATION DES CONTRAINTES

Les contraintes liées à un projet sont les limites générales d'un projet notamment les délais, les coûts et les risques. Il est important d'identifier les contraintes d'un projet, car elles ont les répercussions sur les performances de ce dernier. Dans le cadre de notre projet nous avons identifié les contraintes à plusieurs niveaux :

# a. Contrainte financière du projet

Une contrainte financière est un ensemble de pressions socio-économiques exercées sur un agent économique, limitant ainsi ses responsabilités d'investissement ou de dépenses. Cette contrainte est normalement celle qui pose le moins de problèmes au développeur, mais il est important d'en tenir compte par exemple des conséquences financières inhérentes au projet ou

la prestation comme la conception détaillée, l'implémentation ou le codage, la formation du personnel et la maintenance manuelle. Le tableau ci-dessous illustre clairement nos explications :

**TABLEAU 3: CONTRAINTE FINANCIERES** 

Désignation	Quantité	Prix unitaire (FCFA)	Prix total (FCFA)
Recueil de l'existant	1	150.000	150.000
Spécification du besoin	2	270.000	540.000
Conception détaillé	2	350.000	700.000
Test et validation	6	30.000	180.000
Mise en production	1	75.000	75.000
	Total	1	1.645.000

# b. Contraintes techniques

Une contrainte technique est une donnée d'entrée dans la recherche d'une solution technique. Elle va accomplir le développement ou la mise en œuvre d'une solution. Pour cela il est important de rester innovant et avoir une expérience utilisateur parfaite et des fonctionnalités avancées ce qui nécessité d'adopter les derniers pratiques pour l'architecture des applications web. Pour pouvoir réaliser une application web maintenanciable, il faut pourvoir éviter les problèmes suivants :

- La faible testabilité de la logique métier ;
- La paralysie technique ;
- Le choix technique erroné;
- La duplication du code ;

Il est donc vital d'avoir une équipe technique employant le bon motif d'architecture pour chaque projet. En générale, quand on développe une application web, on utilise l'architecture trois tiers (ou trois couches) incluant : l'interface utilisateur, la logique métier, et l'accès aux données.

# c. Contraintes humaines

Cette partie de notre travail consiste à présenter les différentes ressources humaines et leur rôle dans la réalisation de notre projet. Cela sera regroupé dans le tableau suivant :

TABLEAU 4: CONTRAINTES HUMAINES

Désignation	Profil	Compétence	Mission
Un concepteur	BTS génie logiciel	Compétence en	Participer à la conception
		modélisation	détaillée du projet
		avec MERISE	
Les	BTS génie logiciel	Compétence en	Aider dans l'implémentation du
développeurs		HTML5, CSS3,	projet
		JavaScript, PHP	
Les testeurs	BTS génie logiciel	Compétence en	Participer au test de validation du
		test	projet

# d. Contraintes temporelles

Les contraintes temporelles représentent l'ensemble d'évènements ou facteurs naturels pouvant empiéter sur l'avancée du projet. Le tableau suivant représente les contraintes temporelles :

**TABLEAU 5: CONTRAINTES TEMPORELLES** 

TACHES A FAIRE	DATE DEBUT	DATE FIN
Collecte des informations	03 juillet 2023	14 juillet 2023
Analyse des besoins	17 juillet 2023	28 juillet 2023
Spécification	28 juillet 2023	3 août 2023
Conception	04 août 2023	01 septembre 2023
Programmation	04 septembre 2023	02 novembre 2023
Test de validation et d'intégration	02 novembre 2023	15 novembre 2023
Mise en production	16 novembre 2023	20 novembre 2023
Formation	21 novembre 2023	05 décembre 2023

# Conception et réalisation d'une application web de gestion des stocks du matériel informatique : cas de Hi-Tech Vision>>

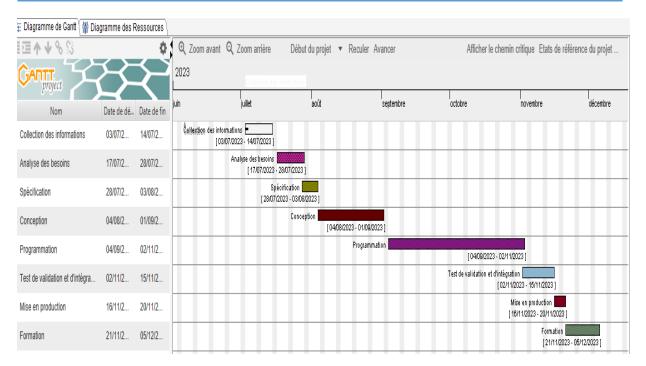


FIGURE 2: DIAGRAMME DE GANTT

Le projet portera sur une durée de 104 jours c'est-à-dire 03 mois 02 semaines 05 jours

# e. Contraintes technologiques

Une contrainte technologique représente un ensemble de pression technologique qui peuvent influencer sur l'implémentation d'un projet informatique. Ici il est question pour nous de présenter toutes les technologies que nous allons utiliser tout au long de la réalisation de notre projet sont :

- **MERISE** : méthode qui nous permet de ressortir :
  - Le modèle conceptuel des données
  - Le modèle logique des données
- UML : langage de modélisation qui nous permettra de ressortir :
  - Le diagramme de cas d'utilisation
  - Le diagramme de classe
  - Le diagramme de séquence
- HTML5 :
- CSS3 : utilisé pour réaliser des mises en forme sur des interfaces
- PHP7.2.4 : utilisé pour gérer les interfaces entre l'interface utilisateur et la base de donnée
- **SQL 5.7.21** : utilisé pour la création, la gestion de la base de données ainsi que pour exécuter les requêtes demandées par l'utilisateur.

# 9. EVALUATION DES MESURES DE SUCCES

L'évaluation des mesures de succès permet de montrer par quel type de tests, par qui, comment et avec quel logiciel sera testé une application. Les tests permettent de valider une application ou un module tout au long de son développement. Ceux-ci sont le plus souvent effectués sur deux niveaux :

- **Niveau structurel** : c'est-à-dire au niveau du code
- Le niveau fonctionnel : représente par des tests portés sur les fonctionnalités de bas niveau comme de haut niveaux

Les tests fonctionnels étant appliqués sur plusieurs niveaux, il est donc préférable de présenter les tests d'une manière générale, suivant la manière dont ils sont effectués. Les types de test permettant de valider la structure d'une application sont :

#### • Les tests unitaires

Ce type de test consiste à tester individuellement les composants de l'application. On pourra ainsi valider la qualité du code (méthode, classe) et les performances d'un module.

# • Les tests d'intégration

Ils sont exécutés pour valider l'intégration des différents modules entre eux et dans leur environnement. Ils permettront de mettre en évidence des problèmes d'interface entre différents programmes.

# • Les tests de performance

Permettent de tester la qualité du code développé. En effet un exemple de test de performance peut consister à vérifier les temps de réponse d'une application, module, méthode, suivant un chargement de données plus ou moins conséquent.

#### • Les tests fonctionnels

Ils sont pour but de vérifier la conformité de l'application développée avec le cahier de charge initial. Ils sont basés sur les spécifications fonctionnelles et techniques.

#### • Les tests de validations

Ici ce test permet de vérifier si toutes les exigences du client décrit dans le document de spécification du logiciel sont respectées. Le test se décompose généralement en plusieurs phases à savoir :

- La validation fonctionnelle : les tests fonctionnels assurent que différents modules ou composants implémentent correctement les exigences du client
- Validation des solutions : ici elle s'assure que les exigences du client sont respectées d'un point de vue du cas d'utilisation

Validation de performance et de robustesse : la validation de performance vérifie la conformité de la solution par rapport à ses exigences de performance, tandis que les tests de robustesse vont essayer de mettre en évidence des éventuels problèmes de stabilité et de fiabilité dans le temps.

L'application de gestion des stocks du matériel informatique sera au préalable testé par le maitre d'œuvre à travers les différents tests cités plus haut.

#### II. PROJET D'AUTOMATISATION

Le projet d'automatisation vise à expliquer comment est-ce qu'on regarde la future application informatique que nous allons mettre sur pieds.

Etant donné que l'entreprise fonctionne manuellement, alors dans notre projet d'automatisation, nous allons devoir mettre sur pieds une base de données digitale et centralisé qui permettra de fonctionner dans un réseau informatique ce qui va garantir le partage et la cohérence des données. A cette base de données nous allons intégrer une application informatique qui va devoir être exploité par différents utilisateurs pour pouvoir exercer leur besoin métier. Avec un tel projet l'entreprise sera de plus en plus efficace car le traitement des activités se fera rapidement, les données seront sécurisées, centralisées donc partageable c'est-à-dire que les utilisateurs pourront accéder aux données aisément.

#### **CONCLUSION**

En somme, il était question pour nous de parler de l'étude préalable basée sur notre projet. Ainsi, pour y parvenir nous avons débuté par une étude de l'existant d'une part et d'autre part du cahier de charge. Mais pour mieux comprendre nous devrons passer par une étape qui est **l'étude détaillé.** 

# **CHAPITRE 3: MODELISATION DETAILLE**

Dans le souci de mieux comprendre et concevoir un système d'information, la modélisation vi le jour. La modélisation détaillée est une technique qui consiste à analyser et à concevoir les informations contenues dans un système afin de représenter ses informations, de structurer le stockage et les traitements informatiques. Elle est la première étape pour définir la structure des données. Elle est un processus de description de la structure, des associations, des relations et des contraintes relatives aux données disponibles. La modélisation sert à établir des normes et à coder des règles de gestion des informations dans l'organisation.

#### SECTION 1: PRESENTATION DU LANGAGE DE MODELISATION UTILISE

#### I. BREVE PRESENTATION DE LA METHODE UML

Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogramme conçu comme méthode normalisée de visualisation dans le domaine du développement logiciel et en conception orientée objet. UML est une synthèse de langage de modélisation objet antérieur comme : BOOCH ; OMT ; OOSE ; issu des travaux de GRADY BOOCH, JAMES RUMBAUGH et IVAR JACOBSON. De nos jours UML est à présent un standard adopté par l'Object Management Group (OMG).

## II. Les principaux diagramme d'UML

# **❖** Diagramme de classes

Étant donné que de nombreux logiciels reposent sur la programmation orientée objet, dans laquelle les développeurs définissent différentes fonctions pouvant être utilisées, les diagrammes de classes sont le type de diagramme UML le plus couramment employé. Les diagrammes de classes exposent la structure statique d'un système, notamment les classes, leurs attributs et leurs comportements, ainsi que les liens entre chacune d'elles.

Une classe est représentée par un rectangle contenant trois compartiments empilés verticalement. Le compartiment supérieur contient le nom de la classe et est indispensable, tandis que les deux compartiments inférieurs fournissent des détails sur les attributs et les opérations ou comportements de la classe.

# **❖** Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation modélisent la manière dont les utilisateurs, sont représentés sous forme de figurines appelées « acteurs », interagissent avec le système. Ce type de diagramme UML est une vue d'ensemble des relations entre les acteurs et les systèmes, ce qui en fait un excellent outil pour présenter votre système à un public non technique.

# **Diagramme de séquence**

Un diagramme de séquence, parfois appelé diagramme d'événements ou scénario d'événements, montre l'ordre dans lequel les objets interagissent. Ils vous permettent ainsi de représenter visuellement des scénarios d'exécution simples. Modèle de diagramme de séquence UML pour système d'achat en ligne Diagramme de séquence UML pour système d'achat en ligne (cliquez sur l'image pour le modifier en ligne)

#### **SECTION 2: MODELISATION DU FUTUR SYSTEME**

#### I. ANALYSE DES BESOINS FONCTIONNELS

#### a) Les acteurs du système

Les acteurs: utilisateurs qui interagissent avec un système. Un acteur peut être une personne, une organisation ou un système externe qui interagit avec votre application ou votre système. Il s'agit nécessairement d'objets externes qui produisent ou consomment des données. C'est pour des acteurs que le système est construit sans acteurs le système n'a pas de raison d'être! Une personne peut être plusieurs acteurs pour le système (un thésard vacataire est à la fois un étudiant et un professeur), il n'est pas toujours nécessaire de créer un acteur pour tous les intervenant extérieurs. On utilise aussi les acteurs pour modéliser des systèmes externes comme une imprimante, un autre logiciel, le système d'exploitation, le système de gestion de fichier, un réveil matin avec un branchement TCP/IP, etc. ...Il faut décrire les acteurs en fonctions des fonctionnalités qu'ils demandent au système et/ou des fonctionnalités que le système leur demande. La création des acteurs se fait avec le client.

# b) Rôles des acteurs du système

UML n'emploie pas le terme d'utilisateur, mais d'acteur. Les acteurs d'un système sont les entités externes à ce système qui interagissent (saisie de données, réception d'information...) avec lui. Les acteurs du système sont : le client il a pour rôle de se connecter, passer la commande, choisir le produit puis envoyer la quantité du produit au système. Le fournisseur quant à lui a pour but de vérifier la disponibilité du produit c'est-à-dire si la quantité du produit commandé par le client est suffisant et est disponible alors le fournisseur valide la commande sinon il retourne un message au client lui demandant d'annuler ou de diminuer le stock. Si le stock est diminué alors il sauvegarde et passe à livraison.

#### II. DIAGRAMME DE CAS D'UTILISATION

# 1) Objectif

Les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système. Ils décrivent les fonctions générales et la portée d'un

système. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique. Il ne faut pas négliger cette première étape pour produire un logiciel conforme aux attentes des utilisateurs.

#### 2) PRESENTATION DES CONCEPTS CLES ET FORMALISME

❖ En langage UML, les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système. Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ils identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisations décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne.

Le diagramme de cas d'utilisation illustre et définit le contexte et les exigence d'un système entier, ou des parties essentielles d'un système. Vous pouvez modéliser un système complexe avec un seul diagramme de cas d'utilisation, ou créer de nombreux diagramme de cas d'utilisation pour modéliser les composants du système.

Les rubriques suivantes décrivent les éléments de modèle dans les diagrammes de cas d'utilisation :

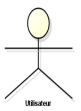
# **❖** Cas d'utilisation

Un cas d'utilisation décrit une fonction qu'un système exécute pour atteindre l'objectif de l'utilisateur. Un cas d'utilisation doit renvoyer un résultat observable qui est utile pour l'utilisateur du système.



#### \* Acteur

Un acteur représente un rôle d'un utilisateur qui interagit avec le système que l'on modélise. L'utilisateur peut être un utilisateur humain, une organisation, une machine ou un autre système externe.



# Sous systèmes

Dans les modèles UML, les sous-systèmes sont un type de composant stéréotypé représentant des unités comportementales indépendantes dans un système. Les sous-systèmes sont utilisés dans les diagramme de classes, de composants et de cas d'utilisation pour représenter des composant de grande taille dans le système à modéliser.

# ❖ Relation dans les diagrammes de cas d'utilisation

En langage UML, une relation est une connexion entre les éléments de modèle. Une relation UML est un type d'élément de modèle qui ajoute une sémantique à un modèle en définissant la structure et le comportement entre les éléments de modèle.

Il existe principalement deux types de relation en UML:

- Les dépendances stéréotypées : qui sont plus explicitées par un stéréotype (les plus utilisés sont l'inclusion et l'extension).

Une dépendance se représente par une flèche avec un trait pointillé.

- Relation d'inclusion;
- Relation d'extension ;
- La généralisation ou la spécialisation

Une généralisation se représente par un trait plein dont la pointe est un triangle fermé désignant le cas le plus générale.

#### 3) Modélisation du diagramme des cas d'utilisation

Les diagrammes de cas d'utilisation modélisent la manière dont les utilisateurs, sont représentés sous forme de figurines appelées « acteurs », interagissant avec le système. Ils décrivent les fonctions générales et la portée d'un système.

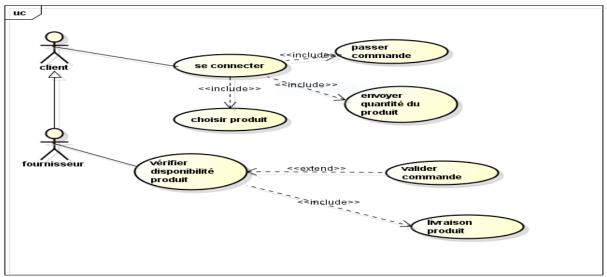


FIGURE 3: DIAGRAMME DE CAS D'UTILISATION

# 4) Description textuelle des cas d'utilisation

Le diagramme de cas d'utilisation illustré ci-dessus représente un système de gestion de stocks du matériel informatique.

Le client se connecte sur le site de l'entreprise, après validation par le fournisseur, le client passe une commande après être authentifié puis choisi son fournisseur et le produit qu'il désire et envoie la quantité du produit au fournisseur. Le fournisseur procède donc au suivi de la commande et passe ensuite à la vérification de la disponibilité de stock du produit valide la commande et livre le produit au client.

#### III. DIAGRAMME DES CLASSES

#### 1. Objectif

Les diagrammes de classes présentent de nombreux avantages pour n'importe quel type d'organisation. Vous pouvez les utiliser pour : Illustrer des modèles de données pour des systèmes d'information, quel que soit leur degré de complexité. Mieux comprendre l'aperçu général des schémas d'une application.

# 2. Présentation des concepts du diagramme des classes

**Diagramme de classe** est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie pratique de l'UML, ne s'intéressant pas aux aspects temporels et dynamiques.

Une classe décrit les responsabilités. Le comportement et le type d'un ensemble d'objet. Les éléments de cet ensemble sont les instances de la classe. Elle est un ensemble de fonction et de données (attribut) qui sont liés par un champ sémantique.

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet. Il modélise les concepts du domaine d'application ainsi les concepts internes crées de toutes les pièces dans le cadre de l'implémentation d'une application. Le diagramme de classe permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation.

Les rubriques suivantes décrivent les éléments de modèle dans les diagrammes de classe :

# **\*** Encapsulation, visibilité, interface

#### ClasseX

-attribut\_1: int
-attribut\_2: string
+set\_attribut\_1(int): void
+get\_attribut\_1(): int

+set\_attribut\_2(string): void +get\_attribut\_2(): string

L'encapsulation est un mécanisme consistant à rassembler les données et les méthodes au sein d'une structure en cachant l'implémentation de l'objet, c'est-à-dire en empêchant l'accès aux données par un autre moyen que les services proposés. Ces services accessibles (offerts) aux utilisateurs de l'objet définissent ce que l'on appelle l'interface de l'objet (sa vue externe).

La visibilité est constituée de :

- **Public ou** +: tout élément qui peut voir le conteneur peut également voir l'élément indiqué.
- **Protected ou # :** seul un élément situé dans le conteneur ou un de ses descendants peut voir l'élément indiqué.
- **Private ou :** seul un élément situé dans le conteneur peut voir l'élément.
- Package ou ~ ou rien : seul un élément déclaré dans le même paquetage peut voir l'élément.

# 3. Classe

Une classe est un concept abstrait représentant des éléments variés comme :

- Des éléments concrets (ex. : des avions),
- Des éléments abstraits (ex. : des commandes de marchandises ou services),

Une classe est la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes. Un objet est une instance d'une classe. C'est une entité discrète dotée d'une identité, d'un état et d'un comportement que l'on peut invoquer. Les objets sont des éléments individuels d'un système en cours d'exécution. Par exemple, si l'on considère qu'Homme (au sens être humain) est un concept abstrait, on peut dire que la personne Marie-Cécile est une instance d'Homme. Si Homme était une classe, Marie-Cécile en serait une instance : un objet.

#### 4. Relation

#### ✓ Association

Une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc qu'il peut y avoir des liens entre des instances des classes associées.

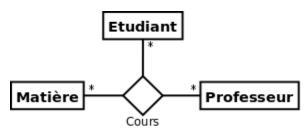
# ✓ Association binaire

#### Exemple:

Une association binaire est matérialisée par un trait plein entre les classes associées. Elle peut être ornée d'un nom, avec éventuellement une précision du sens de lecture (▶ ou ◄).

Quand les deux extrémités de l'association pointent vers la même classe, l'association est dite réflexive.

#### ✓ Association n-aire



Une association n-aire lie plus de deux classes. La section détaille comment interpréter les multiplicités d'une association n-aire. La ligne pointillée d'une classe-association peut être reliée au losange par une ligne discontinue pour représenter une association n-aire dotée d'attributs, d'opérations ou d'associations. On représente une association n-aire par un grand losange avec un chemin partant vers chaque classe participante. Le nom de l'association, le cas échéant, apparaît à proximité du los

# 5. Modélisation du diagramme des classes

Ce diagramme fait partie de la partie pratique d'UML, ne s'intéressant pas aux aspect temporels et dynamique.

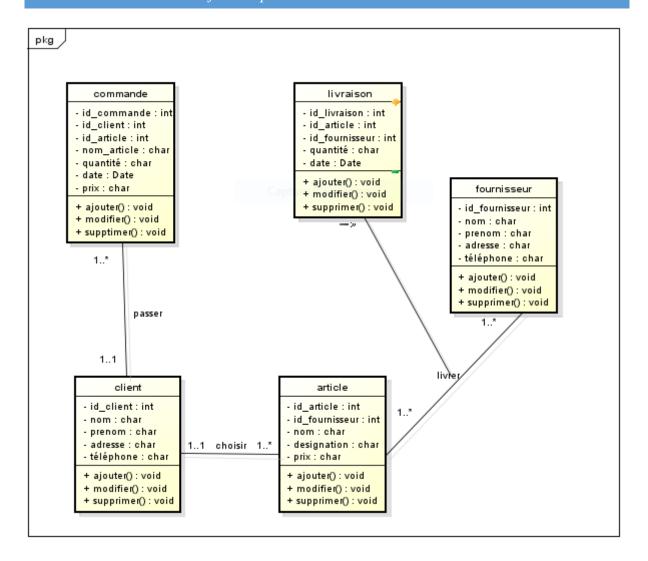


FIGURE 4: DIAGRAMME DE CLASSE DE GESTION DES STOCKS

#### 6. Textuelle des classes du système

Cas de classe du système ci-dessus représente le système de gestion des stocks. Ici le client choisi le produit ou l'article et passe la commande au fournisseur ; donne la quantité du produit puis le fournisseur vérifie la disponibilité des stocks du produit : si c'est disponible il valide sinon il renvoi la commande et envoie un message de produit indisponible. S'il y a erreur sur le nom du produit ou du fournisseur on appuie sur le bouton modifié et on le modifie.

# IV. DIAGRAMME DE SEQUENCE

#### 1. Objectif

L'utilité du diagramme de séquence. Le but est de décrire comment se déroulent les interactions entre les acteurs ou objets. La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets.

# 2. Présentation des concepts du diagramme des séquences

Les diagrammes de séquence sont une solution populaire de modélisation dynamique en langage UML, car ils se concentrent plus précisément sur les *lignes de vie*, les processus et les objets qui vivent simultanément, et les messages qu'ils échangent entre eux pour exercer une fonction avant la fin de la ligne de vie. Un diagramme de séquence est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Les diagrammes de séquence sont parfois appelés diagrammes d'événements ou scénarios d'événements.

Les diagrammes de séquence peuvent constituer des références utiles pour les entreprises et d'autres organisations. On dessine un diagramme de séquence pour :

- Représenter les détails d'un cas d'utilisation UML;
- Modéliser le déroulement logique d'une procédure, fonction ou opération complexe ;
- Voir comment les objets et les composants interagissent entre eux pour effectuer un processus;

# <<Conception et réalisation d'une application web de gestion des stocks du matériel informatique : cas de Hi-Tech Vision>>

Symbole	Nom	Description
	Symbole	Représente une classe ou un objet en
NomObjet : NomClasse	d'objet	langage UML. Le symbole objet montre
		comment un objet va se comporter dans le
		contexte du système. Les attributs de
		classe ne doivent pas être énumérés dans
		cette forme.
	Boîte	Représente le temps nécessaire pour qu'un
	d'activation	objet accomplisse une tâche. Plus la tâche
		nécessite de temps, plus la boîte
		d'activation est longue.
	Symbole	Montre les entités qui interagissent avec le
	d'acteur	système ou qui sont extérieures à lui.
District		
Perkers	Symbole de	Utilisé dans la notation UML 2.0 pour
Package	paquetage	accueillir les éléments interactifs du
Attributes		diagramme. Également connue sous le
		nom de « cadre », cette forme
		rectangulaire est représentée par un petit
		rectangle intérieur qui contient l'intitulé
		du diagramme.
	Symbole de	Représente le passage du temps qui se
	ligne de vie	prolonge vers le bas. Cette ligne verticale
		en pointillés montre les événements
:User		séquentiels affectant un objet au cours du
:		processus schématisé. Les lignes de vie
		peuvent commencer par une forme
1		rectangulaire avec un intitulé ou par un
		symbole d'acteur.

# <<Conception et réalisation d'une application web de gestion des stocks du matériel informatique : cas de Hi-Tech Vision>>

Loop	Symbole de	On utilise ce symbole pour modéliser des
[Condition]	boucle	scénarios ou une situation qui ne se
	optionnelle	produira qu'à certaines conditions.
Alternative	Symbole	Symbolise des choix (qui en général
[Condition]	d'alternatives	s'excluent mutuellement) entre deux
[Else]		séquences de messages ou plus. Pour
		représenter les alternatives, utilisez la
		forme rectangulaire comportant un intitulé
		et une ligne en pointillés à l'intérieur.

Les diagrammes de séquence sont composés des icônes et des éléments suivants :

# **Symboles de messages courants**

	Nom	Description
Symbole		
	Symbole de messages synchrones	Représentés par une ligne pleine terminée par une pointe de flèche pleine. On utilise ce symbole lorsqu'un expéditeur doit attendre une réponse à un message avant de continuer. Le diagramme doit montrer à la fois l'appel et la réponse.

	Symbole	de	Représentés par une ligne
	messages		pleine terminée par une
$\longrightarrow$	asynchrones		pointe de flèche. Les
			messages asynchrones ne
			nécessitent pas de
			réponse avant que
			l'expéditeur ne continue.
			Seul l'appel doit être
			inclus dans le diagramme.
			moras dans le diagramme.
	Symbole	de	Représentés par une ligne en pointillés
<b>&lt;</b> − − −	messages	de	terminée par une tête de flèche.
	retour		
•	asynchrones		
	Symbole	de	Représentés par une ligne en pointillés
- < <create>&gt;</create>	messages	de	terminée par une pointe de flèche. Ces
	création		messages créent de nouveaux objets.
	asynchrones		
	Symbole	de	Représentés par une ligne en pointillés
	messages	de	terminée par une pointe de flèche, ces
/	réponse		messages sont des réponses aux appels.
1	Symbole	de	Représentés par une ligne pleine terminée
	messages	de	par une pointe de flèche pleine, suivie du
	suppression		symbole X. Ces messages détruisent un

# 3. Acteur

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs :

Les objets au cœur d'un système interagissent en s'échangent des messages. Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

#### **Ligne de vie**

Une ligne de vie représente un participant à une interaction (objet ou acteur). Une ligne de vie est une instance, donc il y a nécessairement les *deux points* (:) dans son libellé. Dans le cas d'une collection de participants, un sélecteur permet de choisir un objet parmi n.

#### 4. Messages

Un message définit une communication particulière entre des lignes de vie (objets ou acteurs). Plusieurs types de messages existent, dont les plus courants :

- L'envoi d'un signal (typiquement utilisé pour la gestion éventuelle) ;
- L'invocation d'une opération (appel de méthode) ;
- La création ou la destruction d'un objet ;

La réception des messages provoque une période d'activité (rectangle vertical sur la ligne de vie) marquant le traitement du message (spécification d'exécution dans le cas d'un appel de méthode).

#### Messages synchrones et asynchrones

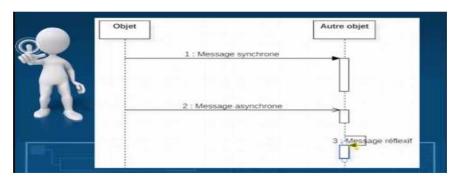


FIGURE 5: MESSAGE SYNCHRONE ET ASYNCHRONE

Un message *synchrone* bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur.

## <<Conception et réalisation d'une application web de gestion des stocks du matériel informatique : cas de Hi-Tech Vision>>

- Si un objet A envoie un message synchrone à un objet B, A reste bloqué tant que B n'a pas terminé ;
- On peut associer aux messages d'appel de méthode un message de retour (en pointillés) marquant la reprise du contrôle par l'objet émetteur du message synchrone ;

**Un message** *asynchrone* n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.

Les messages synchrones correspondent le plus souvent à une opération :

- A l'invocation, le flux contrôle passe de l'émetteur au récepteur ;
- L'émetteur attend la fin de l'exécution, et reprend après le retour ;

Les messages asynchrones correspondent à des signaux dans le diagramme de classes. Les signaux sont des objets dont la classe est stéréotypée << signal >> et dont les attributs (porteurs d'information) correspondent aux paramètres du message.

#### 5. Modélisation du diagramme des séquences

Le diagramme de séquence fait partie des diagrammes comportementaux (dynamique) et plus précisément des diagrammes d'interactions (le diagramme d'objet, le diagramme de composant, le diagramme de structure composite). Il montre l'ordre dans lequel les objets interagissent. Ils vous permettent ainsi de représenter visuellement des scénarios d'exécution simples. c'est le cas de ce diagramme de séquence de gestion des stocks et d'authenfication.

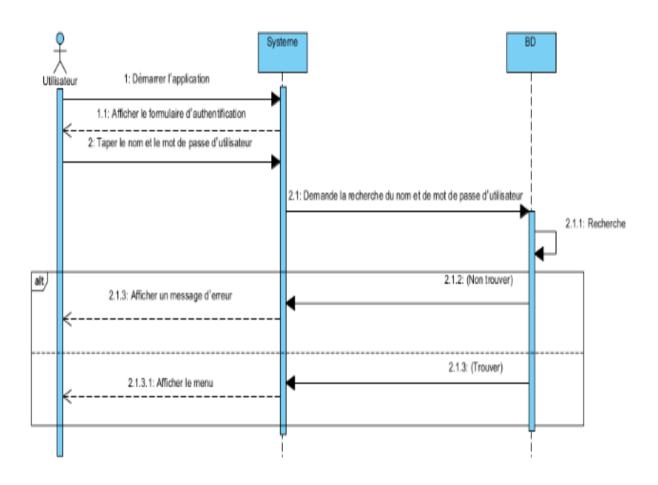
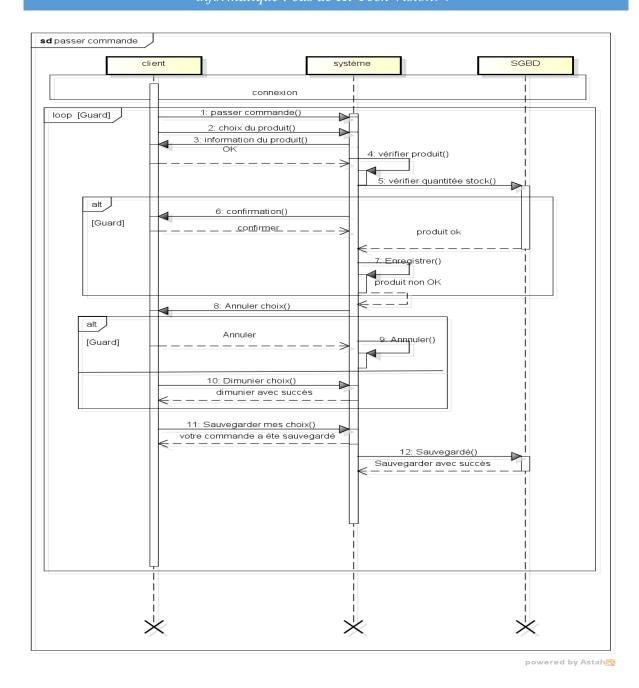


FIGURE 6: DIAGRAMME DE SEQUENCE CAS D'AUTHENTIFICATION



#### FIGURE 7: DIAGRAMME DE SEQUENCE CAS D'UTILISATION

#### **CONCLUSION**

Parvenue au terme de ce chapitre, il était question pour nous de réaliser la modélisation détaillée portant sur notre projet, ainsi pour y parvenir nous avons débuté par une brève présentation de la méthode UML, modélisation du diagramme des cas d'utilisation, modélisation du diagramme de classe ainsi que la modélisation du diagramme de séquence. Cependant, pour y arriver nous devons passer par une dernière étape qui est **l'implémentation**.

#### **CHAPITRE 4: IMPLEMENTATION**

L'implémentation est la réalisation, l'exécution ou la mise en pratique d'un plan, d'une méthode ou bien d'un concept, d'une idée, d'un modèle, d'une spécification, d'une norme ou d'une règle dans un but précis. Dans ce chapitre, il sera question pour nous de présenter quelques SGBD, les serveurs de base de données choisit ainsi que quelques interfaces de notre application. Pour y parvenir, nous allons présenter d'une premier temps l'implémentation de base de données et dans un deuxième temps, l'implémentation de notre application.

#### I. IMPLEMENTATION DE LA BASE DE DONNEES

#### 1. Définition d'une base de données

Dans le domaine de l'informatique, l'un des véritables soucis de l'utilisateur fut du pouvoir stocker massivement les données et de pouvoir en disposer régulièrement afin de les consulter et de les modifier ; ce qui a permis la création des bases de données. **Une base de données informatisée** est un ensemble de structure de données enregistrées sur des supports accessibles par ordinateur.

C'est la pièce centrale d'un système d'information ou d'un système de base de données qui régit la collecte, le stockage, le traitement et l'utilisation de données. Ce dispositif comporte souvent un logiciel moteur, des logiciels applicatifs, et un ensemble de règles relatives à l'accès et l'utilisation des informations.

#### 2. Enjeux des bases de données

Les bases de données ont pris une place importante en informatique et particulièrement dans le domaine de la gestion. L'étude des bases de données a conduit au développement de concepts, méthodes et algorithmes spécifiques, notamment pour gérer les données en mémoire secondaire (disque dur). En effet dès l'origine de la discipline, les informations ont observé que la taille de la RAM ne permettait pas de charger l'ensemble d'une base de données en mémoire. Car le volume ne cesse de s'accroitre sous la poussée des nouvelles technologies.

#### 3. Quelques exemples de SGBD

SGBD (Système de Gestion des Bases de Données) est un logiciel système permettant aux utilisateurs et programmeurs de créer et de gérer des bases de données. Il existe de nombreuses SGBD parmi lesquels nous pouvons citer :

- PostgreSQL
- MYSQL
- Oracle
- **♣** IBM DB 2

- ♣ Microsoft SQL
- Sysbase
- MS Access
- **♣** Informix

#### 4. Pourquoi avoir choisi MYSQL ou SQL SERVER

MYSQL est un système de gestion de base de données relationnelles (souvent appelé par son acronyme : SGBDR). Il fonctionne sous linux et Windows. Les avantages de MYSQL sont nombreux et expliquent sa grande popularité auprès des développeurs : il est totalement open source et gratuit, ses performances sont excellentes et il est en plus multi-utilisateurs et prévu pour fonctionner parfaitement avec PHP, qui est le langage de programmation que nous allons également utiliser pendant la mise en place de notre application, nous pouvons également ajouter à cela qu'il est très facile à mettre en place et à utiliser.

Quand à SQL SERVER est l'outil pour implémenter le mécanisme d'un système de gestion de base de données relationnelle. Il permet aux développeurs de travailler avec des données afin d'offrir une bonne expérience à l'utilisateur. Dans les organisations, SQL Server est le moyen de traitement des données le plus privilégié car il est très capable de traiter une multitude de données.

#### A. Dictionnaire de données

TABLEAU 6: DICTIONNAIRE DES DONNEES

N°	Variable	Signification	Туре
1	Id_client	Numéro	Int
		d'identification	
2	Nom	Nom de l'entreprise	Varchar(255)
3	Prénom	Téléphone du client	Varchar(255)
4	Email	email du client	Varchar(255)
5	Adresse	Adresse du client	Varchar(255)
6	Ville	Ville du client	Varchar(255)
7	Id_article	Numéro	Int
		d'identification	
8	Nom	Identifiant de l'article	Varchar(255)

# <Conception et réalisation d'une application web de gestion des stocks du matériel informatique : cas de Hi-Tech Vision>>

9	Description	Description de l'article	Varchar(255)
10	Prix	Prix de l'article	Float
11	Image	Image de l'article	Varchar(255)
12	Id_commande	Identifiant de la commande	Int
13	Id_client	Identifiant du client	Int
14	Date_commande	Date de la commande	Datetime
15	Montant	Montant de la commande	Float
16	Etat	Etat de la commande	Varchar(255)
17	Id_fournisseur	Identifiant du fournisseur	Int
18	Nom	Nom du fournisseur	Varchar(255)
19	Adresse	Adresse du fournisseur	Varchar(255)
20	Ville	Ville du fournisseur	Varchar(255)
21	Telephone	Telephone du fournisseur	Varchar(255)
22	Email	Email du fournisseur	Varchar(255)
23	Id_livraison	Identifiant de la livraison	Int
24	Id_commande	Identifiant de la commande	Int
25	Date_livraison	Date de la livraison	Datetime
26	Statut	Statut de la livraison	Varchar(255)

#### 1. Processus de création de la base de données

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

---
---
---
---
---
---
---
Structure de la table `articles`
---
---

CREATE TABLE `articles` (
    `id` int(11) NOT NULL,
    `nom` varchar(255) DEFAULT NULL,
    `quantite` float NOT NULL,
    `image` varchar(255) DEFAULT NULL
    `image` varchar(255) DEFAULT NULL
```

FIGURE 8: PROCESSUS DE CREATION DE LA BASE DE DONNEE GESTION DE STOCKS

#### II. IMPLEMENTATION DE NOTRE APPLICATION

a. Interface de connexion



FIGURE 9: INTERFACE DE CONNEXION

#### b. Interface de menu principale



FIGURE 10: INTERFACE DU MENU PRINCIPAL

- c. Interface de gestion des stocks
- **\*** Enregistrer un article

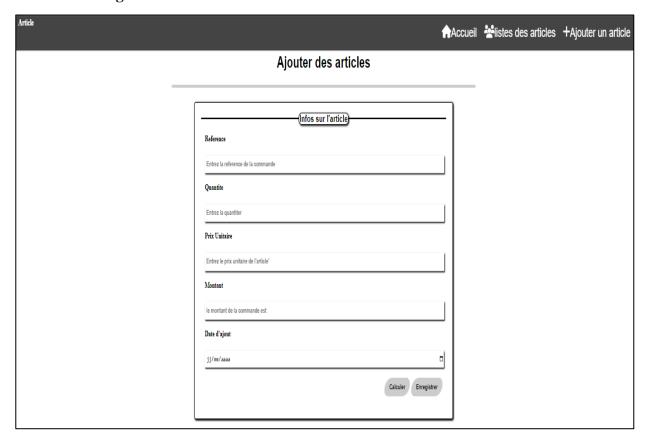


FIGURE 11: INTERFACE D'ENREGISTREMENT D'UN ARTICLE

#### d. Sécurité informatique

La sécurité IT (ou sécurité informatique) est un dispositif vaste et multiforme visant à protéger un réseau informatique et ses données contre toute violation, fuite, publication d'information privées ou attaque. Dans le cadre de notre application, elle sera basée sur la confidentialité des informations et ceci à travers une interface de connexion qui permettra à l'utilisateur de se connecter à l'application ; une interface de récupération de mot de passe oublié qui donnera la possibilité à l'utilisateur du système de récupérer son mot de passe via son adresse mail.

#### **CONCLUSION GENERALE**

Au terme de notre projet de fin d'étude qui consistait à « l'étude, la conception et la réalisation d'une application de gestion de stock du matériel informatique », en passant par les différentes étapes du cycle de développement d'un projet depuis l'étude de l'existant et la spécification des besoins, suivi de la modélisation du système suivant le langage UML, et de l'implémentation pour finir par la mise en exploitation et le déploiement du système. L'expérience vécue durant ce stage était fort profitable, dans la mesure où elle a permis de consolider nos acquis au sein de HI-TECH VISION notamment en matière de développement informatique. On a eu ainsi l'opportunité de travailler dans un environnement de très haut niveau et de développer notre relationnel au sein de l'organisme. L'accueil du personnel de HI-TECH VISION nous a ravi, facilité notre insertion dans cette équipe enthousiaste, travailleuse et motivée ; nous a permis de pourvoir mettre en pratique les connaissances acquises dans divers domaines. Chose que nous pouvons dire bien acquise puis que nous avons muri en capacité pratique et en expérience professionnelle. Nous avons également approfondi notre sens des responsabilités et de savoirfaire, du travail en équipe, du respect de la hiérarchie et de l'éthique professionnel qui constituent un socle dans la relation humaine. Cependant, conscient de l'imperfectibilité de toute œuvre humaine, nous restons ouverts en toute humilité aux éventuelles critiques et suggestions en vue d'amélioration.

#### **BIBLIOGRAPHIE**

### Ouvrages

➤ Collection le major de M. DOUMI SERGE BRUNO

#### **4** Sites internet

- > www.publicimpact.eu
- > www.oracle.com
- > www.wikimemories.net
- > www.africmemoir.com

## Table des matières

Sommaire	1
Dédicace	Erreur! Signet non défini.i
Remerciements	iii
Liste des figures	iv
Listes des tableaux	v
Liste des abréviations	vi
Résumé	Erreur! Signet non défini.
Abstract	viii
Avant-propos	Erreur! Signet non défini.
INTRODUCTION GENERALE	1
Chapitre 1 : PRESENTATION DE L'ENTREPRISE	2
SECTION 1 : ENVIRONNEMENT INTERNE	2
I. ENVIRONNEMENT INTERNE	2
1. HISTORIQUE ET EVOLUTION	2
a) Présentation	2
b) EVOLUTION	2
2. ORGANISATION ET FONCTIONNEMENT	2
a. Attribution	2
a) Moyens matériels	3
b) Moyens humains	3
SECTION 2 : DEROULEMENT DU STAGE	3
I. PRESENTATION DU DEPARTEMENT IT	
1. LES OBJECTIFS ET SERVICES DU DEPARTEM	
a) Objectifs du département IT	
b) Services du département IT	3
II. CHRONOGRAMME DES ACTIVITES	4

CONCLUSION	4
SECTION 1 : ETUDE DE L'EXISTANT	5
I. DESCRIPTION DE L'EXISTANT	5
II. CRITIQUE DE L'EXISTANT	5
1. Sur le plan fonctionnel	5
2. Sur le plan organisationnel	6
III. SOLUTION RETENUE	6
SECTION 2 : CAHIER DE CHARGE	6
I. ELABORATION DU CAHIER DE CHARGE	6
1. DEFINITION	6
2. DESCRIPTION DE LA SITUATION ACTUELLE (ETUDE DE L'EXISTANT).	6
3. DELIMITATION DU PERIMETRE DU PROJET	7
4. DELIMITATION DU SYSTEME D'INFORMATION	7
5. SPECIFICATION FONCTIONNEL DE L'APPLICATION	7
6. SPECIFICATION NON FONTIONEL DE L'APPLICATION	8
7. RESPONSABILITE ET ROLES RESPECTIF DES PARTIES ENGAGEES PAR	
L'INTERVENTION	9
8. EVALUATION DES CONTRAINTES	9
a. Contrainte financière du projet	9
b. Contraintes techniques	10
c. Contraintes humaines	10
d. Contraintes temporelles	11
e. Contraintes technologiques	12
9. EVALUATION DES MESURES DE SUCCES	13
II. PROJET D'AUTOMATISATION	14
CONCLUSION	14
CHAPITRE 3 · MODELISATION DETAILLE	15

SECTION	1 : PRESENTATION DU LANGAGE DE MODELISATION UT	ILISE 15
I. B	REVE PRESENTATION DE LA METHODE UML	15
II.	Les principaux diagramme d'UML	15
SECTION	2 : MODELISATION DU FUTUR SYSTEME	16
<b>I. A</b> ]	NALYSE DES BESOINS FONCTIONNELS	16
a)	Les acteurs du système	16
<b>b</b> )	Rôles des acteurs du système	16
II.	DIAGRAMME DE CAS D'UTILISATION	16
1)	Objectif	16
2)	PRESENTATION DES CONCEPTS CLES ET FORMALISME	17
3)	Modélisation du diagramme des cas d'utilisation	18
4)	Description textuelle des cas d'utilisation	19
III.	DIAGRAMME DES CLASSES	19
1.	Objectif	19
2.	Présentation des concepts du diagramme des classes	19
3.	Classe	20
4.	Relation	21
5.	Modélisation du diagramme des classes	21
6.	Textuelle des classes du système	22
IV.	DIAGRAMME DE SEQUENCE	22
1.	Objectif	22
2.	Présentation des concepts du diagramme des séquences	23
3.	Acteur	26
4.	Messages	27
5.	Modélisation du diagramme des séquences	28
CONCLU	SION	30
CHAPITR	E 4: IMPLEMENTATION	31

# <<Conception et réalisation d'une application web de gestion des stocks du matériel informatique : cas de Hi-Tech Vision>>

I. IMPLEMENTATION DE LA BASE DE DONNEES		31	
	1. D	éfinition d'une base de données	31
	2.	Enjeux des bases de données	31
	3.	Quelques exemples de SGBD	31
	4.	Pourquoi avoir choisi MYSQL ou SQL SERVER	32
		A. Dictionnaire de données	32
	1.	Processus de création de la base de données	33
II.	IM	PLEMENTATION DE NOTRE APPLICATION	34
	a	. Interface de connexion	34
	b	. Interface de menu principale	35
	c.	Interface de gestion des stocks	35
	d	. Sécurité informatique	36
CON	NCLU	SION GENERALE	37
BIB	LIOG	RAPHIE	38
Tabl	le des	matières	39