**Dédicace**

Je dédie ce travail :

* A ma mère **YEBA Nicole Pascaline** qui a toujours cru en moi et qui m’a soutenu quoiqu’il puisse arriver afin de me procurer un soutien moral quant à l’investissement intellectuel fourni par moi-même.
* A mon père **DAH-BONGO Jean Marc** qui a également contribué à ma scolarité et m’a permis d’arriver au niveau où je suis actuellement sans pour autant être proche de moi, néanmoins j’ai grâce à ses conseils pu redoubler d’efforts afin de travailler d’arrache pieds pour enfin concevoir ce travail de recherche.
* A tout le reste de la famille qui a toujours été à mes côtés de loin comme de près me permettant d’avancer avec le soutien de tout un chacun.

Aucun mot, aucune dédicace ne pourrait exprimer mon respect, ma considération ainsi que l’expression de mon profond amour que je ressens pour vous.

Puisse Dieu vous accorder santé, bonheur et longue vie afin que nous pussions un jour combler de joie vos vieux jours.

**Remerciements**

Je tiens tout d’abord à remercier Dieu le tout puissant et miséricordieux, qui m’a donné la force et la patience d’accomplir ce modeste travail.

Ensuite, je remercie **Mr SECK** qui m’a permis de bénéficier de son encadrement ainsi que ses enseignements et conseils me permettant de réaliser ce travail.

Je veux également exprimer par ces quelques lignes ma gratitude envers tous ceux qui, par leurs présences, leurs soutiens, leurs disponibilités et leurs conseils m’ont donné le courage nécessaire d’accomplir ce projet.

Enfin, je ne peux achever mes remerciements sans exprimer ma gratitude à tous les enseignants de l’**Institut** **Supérieur** d’**Informatique** de Dakar, pour leur dévouement et leur assistance tout au long de cette année et des deux précédentes qui définissent mon parcours d’études supérieurs à l’**Institut** **Supérieur** d’**Informatique**.

**Avant-propos**

L’institut supérieur d’informatique est un institut d’enseignement supérieur avec une expérience de plus de 25 ans dans la formation continue de type académique et plus particulièrement dans les domaines informatique et gestion tels que : Génie Logiciel, MIAGE, Maintenance Informatique, Réseaux Informatique, Finance et Comptabilité, BAFA, Assistanat de Direction… Et il délivre les diplômes suivants : Diplôme de Technicien Supérieur (DTS), Brevet de Technicien Supérieur (BTS), Licence et Master Professionnel…

Pour l’obtention de la licence en Génie Logiciel, l’ISI exige aux étudiants la rédaction d’un mémoire de fin de cycle. C’est dans ce cadre que nous avons élaboré ce document qui a pour sujet : « **Étude et mise en place d’une plateforme de vente de denrées alimentaires** »

Cette étude a donc pour objet la mise en place d’une web application qui permettra de gérer les différents problèmes que rencontrent certains clients lorsque ces derniers souhaitent effectuer des courses notamment les variations météorologiques, les difficultés de mobilité, la fatigue des transports et bien d’autres.

Ce document constitue notre premier travail de recherche académique, c’est pourquoi nous sollicitons de la part du jury, beaucoup d’indulgence pour ce qui concerne son évaluation.

**Sommaire**

[I Première partie : cadres théorique et méthodologique x](file:///D:\DEUXIEME%20ANNEE%20ISI\Memoire%20BTS\Memoire%20Licence%202019%20-%20Copie%20Vrai.docx#_Toc23164041)

[I.I Présentation de la structure xi](#_Toc23164042)

[I.I.I Présentation générale xi](#_Toc23169304)

[I.I.II Historique xi](#_Toc23169305)

[I.I.III Mission...……………………………………………………………………………..xi](#_Toc23169306)

[I.II Cadre méthodologique xii](#_Toc23164043)

[I.II.I Méthodes fonctionnelles xii](#_Toc23169308)

[I.II.II Langage de modélisation oriente objet xii](#_Toc23169309)

[I.II.III Comparaison entre l’approche fonctionnelle et l’approche objet xii](#_Toc23169310)

[II Deuxième partie : Méthodologie d’Analyse et de Conception, et choix méthodologique xiv](file:///D:\DEUXIEME%20ANNEE%20ISI\Memoire%20BTS\Memoire%20Licence%202019%20-%20Copie%20Vrai.docx#_Toc23164044)

[II.I Méthodologie d’Analyse et de Conception xv](#_Toc23164045)

[II.I.I Définition des concepts xv](#_Toc23169313)

[A L’Analyse : xv](#_Toc23169314)

[B La Conception : xv](#_Toc23169315)

[C L’Implémentation : xv](#_Toc23169316)

[II.I.II L’importance d’une méthode d’analyse et de conception xv](#_Toc23169317)

[II.I.III Classification des approches ou méthodes d’analyse et de conception xvi](#_Toc23169318)

[D Approches / Méthodes cartésiennes ou fonctionnelles xvi](#_Toc23169319)

[E Approches / Méthodes systémiques xvi](#_Toc23169320)

[F Approches / Méthodes orientées objets xvii](#_Toc23169321)

II.II [Choix méthodologique xviii](#_Toc23164046)

[II.II.I Choix d’une méthodologie d’analyse et de conception xviii](#_Toc23169323)

[II.II.II Présentation d’UML xviii](#_Toc23169324)

[G Diagramme de cas d’utilisation : xx](#_Toc23169325)

[H Diagramme d’activité : xxi](#_Toc23169326)

[I Diagramme de classe : xxi](#_Toc23169327)

[J Diagramme de composants : xxiv](#_Toc23169328)

[K Diagramme de déploiement : xxv](#_Toc23169329)

[III Troisième partie : Expression des besoins xxvi](file:///D:\DEUXIEME%20ANNEE%20ISI\Memoire%20BTS\Memoire%20Licence%202019%20-%20Copie%20Vrai.docx#_Toc23164047)

[III.I Étude du cahier des charges xxvii](#_Toc23164048)

[III.I.I Problématique xxvii](#_Toc23169332)

[III.I.II Expression des besoins xxvii](#_Toc23169333)

[III.I.III Démarche de résolution xxviii](#_Toc23169334)

[III.II Identification des acteurs du système xxviii](#_Toc23164049)

[iv Quatrième partie : Réalisation et Implémentation xxx](file:///D:\DEUXIEME%20ANNEE%20ISI\Memoire%20BTS\Memoire%20Licence%202019%20-%20Copie%20Vrai.docx#_Toc23164050)

[IV.I Analyse et conception xxxi](#_Toc23164051)

[IV.I.I Diagrammes de Cas d’utilisation xxxi](#_Toc23169338)

[L Cas d’utilisation globale du système xxxi](#_Toc23169339)

[M Gestion Utilisateurs xxxii](#_Toc23169340)

[N Gestion Ventes de Produits xxxiii](#_Toc23169341)

[O Gestion Clients xxxiv](#_Toc23169342)

[P Gestion Entrées / Sorties xxxv](#_Toc23169343)

[Q Gestion Fournisseurs xxxvi](#_Toc23169344)

[R Gestion Stock xxxvii](#_Toc23169345)

[IV.I.II Diagrammes d’activité des Cas d’utilisation xxxviii](#_Toc23169346)

[S Diagramme d’activité de la connexion xxxviii](#_Toc23169347)

[T Diagramme d’active de la vente xxxix](#_Toc23169348)

[U Diagramme d’active entrée / sortie xl](#_Toc23169349)

[V Diagramme d’active de la gestion du stock xli](#_Toc23169350)

[IV.I.III Diagramme de classe général xliii](#_Toc23169351)

[IV.II Politique de Sécurité xliii](#_Toc23164052)

[IV.II.I Sécurité au niveau Applicatif xliv](#_Toc23169353)

[W Sécurité de l’application Web xliv](#_Toc23169354)

[X Sécurité au niveau Base de Données xlv](#_Toc23169355)

[Y Sécurité au niveau matériel xlv](#_Toc23169356)

[Conclusion xlvi](#_Toc23164053)

**Glossaire**

**UML**………………………………………………...... (Unified Modeling Language)

**OOSE**……………………………………………… (Object-Oriented Software Engineering)

**OMT**…………………………………………………………. (Object Modeling Technique)

**OMG**…………………………………………………………….... (Object Modeling Group)

**SQL**…………….………………………………………………………. (Structured Query Language)

**CRUD** ………………………………………………………… (Create, Read, Update, Delete)

**Liste des figures**

[Figure 1 Représentation de l’évolution de ‘UML’ xx](#_Toc23152288)

[Figure 2 Représentation d’un acteur du système xxi](#_Toc23152289)

[Figure 3 Représentation d’un cas d’utilisation xxi](#_Toc23152290)

[Figure 4 Les différents nœuds utilisés dans un digramme d’activité xxii](#_Toc23152291)

[Figure 5 Représentation d’une classe en UML xxiii](#_Toc23152292)

[Figure 6 Représentation d’une généralisation entre deux classe en UML xxiii](#_Toc23152293)

[Figure 7 Représentation d’une association entre deux classes en UML xxiv](#_Toc23152294)

[Figure 8 Représentation d’une agrégation en UML xxiv](#_Toc23152295)

[Figure 9 Représentation d’une composition en UML xxv](#_Toc23152296)

[Figure 10 Représentation d’un diagramme de composant en UML xxvi](#_Toc23152297)

[Figure 11 Représentation d’un diagramme de déploiement en UML xxvi](#_Toc23152298)

[Figure 12 Représentation du diagramme de cas d’utilisation global du système xxxii](#_Toc23152299)

[Figure 13 Diagramme de cas d’utilisation de la gestion des utilisateurs xxxiii](#_Toc23152300)

[Figure 14 Diagramme de cas d’utilisation de la gestion de projet xxxiv](#_Toc23152301)

[Figure 15 Diagramme de cas d’utilisation de la gestion des clients xxxv](#_Toc23152302)

[Figure 16 Diagramme de cas d’utilisation de la gestion des produits xxxvi](#_Toc23152303)

[Figure 17 Diagramme de cas d’utilisation de la gestion des vendeurs xxxvii](#_Toc23152304)

[Figure 18 Diagramme de cas d’utilisation de la gestion des commandes xxxviii](#_Toc23152305)

[Figure 19 Diagramme d’activité de la connexion xxxix](#_Toc23152306)

[Figure 20 Diagramme d’activité d’inscription xl](#_Toc23152307)

[Figure 21 Diagramme d’activité de client xli](#_Toc23152308)

[Figure 22 Diagramme d’activité de la gestion de projet xlii](#_Toc23152309)

Introduction Générale :

L’achat et la vente de différentes denrées alimentaires sont des activités qui font partie intégrante de notre vie quotidienne. Chaque jour, il y’a d’une part il y’a de plus en plus de vendeurs qui souhaitent écouler leurs marchandises sur les marchés, et d’autre part des clients qui souhaitent les consommer. Par ailleurs, on observe également l’implantation de divers supermarchés internationaux, offrant aux clients de par la diversité de leurs marchandises un choix plus large dans l’achat des produits, ceci dit à des prix légèrement plus élevés que ceux des marchés nationaux.

Ainsi nous pouvons assister à plusieurs évènements, d’une part au niveau des vendeurs avec la diversité de vendeurs de même produit au niveau des marchés, ce qui peut constituer un élément de concurrence concernant la fidélisation des clients. La venue des supermarchés internationaux proposant un panel de marchandises plus élargit et structuré, poussant certains clients à se ravitailler dans ces derniers au détriment des vendeurs de marchés. D’autre part, nous avons les clients qui sont confrontés aux aléas des conditions météorologiques, la fatigue des transports ou encore les problèmes de mobilité pour les personnes souffrantes

Cette plateforme nous sera très utile car elle permettra aux clients d’effectuer leurs achats directement sur cette plateforme sans avoir à se déplacer. Elle va également proposer un service de livraison aux clients afin de recevoir leurs commandes depuis leur emplacement. Concernant les vendeurs, ces derniers pourront proposer leurs marchandises avec leur prix respectifs facilitant leur commercialisation. Pour les produits de supermarchés, il sera également possible de passer des commandes car tous les produits y seront répertoriés. Cette plateforme sera considérée comme passerelle entre le vendeur qui commercialise et le client qui consomme.

De nos jours on remarque de plus en plus de commerçants se plaindre de la situation des marchés à savoir la rareté des clients, la concurrence interne au niveau des marchés nationaux, mais aussi externe avec les supermarchés internationaux qui font baisser leur chiffre d’affaires. En outre, il existe des clients n’ayant pas le temps de se déplacer pour faire des courses dignes d’approvisionnement, faute de temps ou par simple capacité de mobilité réduite suite à des antécédents de santé ou une inaptitude.

En ce qui concerne notre mémoire l’objectif sera donc ici de recenser les différents produits alimentaires dont les clients sont susceptibles d’avoir besoin, répertorier les potentiels fournisseurs de ces produits, enregistrer ces derniers et leur permettre de vendre leurs produits en ligne.

Du point de vue cadre méthodologique, rien de plus simple à voir dans la société qui nous entoure car force est de constater que la digitalisation a eu un grand impact notamment à la suite des différents désagréments causés par la pandémie et bien d’autres facteurs qui ont poussés les plus petites comme les plus grandes entreprises à avoir recours au digital pour écouler leurs marchandises.

Dans le but de mieux comprendre le fonctionnement de cette plateforme, nous ferons l’analyse des différentes étapes de recensement des différents produits à mettre en vente, des fournisseurs qui auront un espace de vente, et bien sur les clients qui auront libre accès au panel de produits. Pour bien étudier ce sujet, notre plan de travail portera sur quatre parties décomposées comme suit :

**Cadres théorique et méthodologique**

* **Méthodes d’Analyse et de Conception et choix méthodologique**
* **Expressions des besoins**
* **Réalisation et Implémentation**

# Première partie : cadres théorique et méthodologique

## Présentation de la structure

### Présentation générale

La plateforme TOKOSS MARKET est une application web qui voit le jour en 2022. Elle est basée sur le web sous forme de marché virtuel. Elle a pour but de pouvoir vendre différents types de produits et denrées alimentaires. Un panel de produits proposé par différents fournisseurs s’y trouve permettant aux clients d’effectuer leurs achats. Un service de livraison sera garanti afin que le client puisse recevoir sa marchandise sans avoir à se déplacer. Tout fournisseur désireux de vendre ses produits à la possibilité de créer un compte et commencer à déposer ses articles avec les détails nécessaires.

### Historique

TOKOSS MARKET est une plateforme qui fut mise en place par un administrateur qui suit son évolution avec l’intégration des produits se chargeant des tâches d’inscriptions de fournisseurs, suivi des commandes et sécurité des comptes. Avec un départ plutôt timide, c’est avec un optimisme que ce dernier sera dévoué au travail afin que la digitalisation au travers un marché virtuel apporte une solution efficace chez les clients en ce qui concerne leurs besoins d’achats de produits en ligne.

### Mission

La plateforme **TOKOSS MARKET** a pour objectif par son accessibilité le rôle de réunir un maximum de produits alimentaires, le but étant d’offrir une possibilité aux clients de gagner en temps en ce qui concerne toutes leurs courses en passant directement par ce marché virtuel contenant un maximum de denrées alimentaires de première nécessité.

D’autre part offrir à des fournisseurs, une plateforme permettant la vente de leurs produits ce qui augmenterai leurs bénéfices car ils n’auront comme simple effort la publication de leurs marchandises.

## Cadre méthodologique

### Méthodes fonctionnelles

Les méthodes fonctionnelles (également qualifiées de méthodes structurées) mettent en évidence les fonctions à assurer et proposent une approche hiérarchique descendante et modulaire. Le plus haut niveau représente l'ensemble du problème (sous forme d'activité, de données ou de processus, selon la méthode). Chaque niveau est ensuite décomposé en respectant les entrées/sorties du niveau supérieur. La décomposition se poursuit jusqu'à arriver à des composants maîtrisables. L'approche fonctionnelle dissocie le problème de la représentation des données, du problème du traitement de ces données. En résumé, l'architecture du système est dictée par la réponse au problème.

### Langage de modélisation oriente objet

L'approche considère le logiciel comme une collection d'objets dissociés, identifiés et possédant des caractéristiques. Une caractéristique est soit un attribut, soit une entité comportementale de l'objet. La fonctionnalité du logiciel émerge alors de l'interaction entre les différents objets qui le constituent. L'une des particularités de cette approche est qu'elle rapproche les données et leurs traitements associés au sein d'un unique objet.

### Comparaison entre l’approche fonctionnelle et l’approche objet

L'approche structurée privilégie la fonction comme moyen d'organisation du logiciel. En effet, les méthodes d'un objet sont des fonctions. Ce qui différencie sur le fond l'approche objet de l'approche fonctionnelle, c'est que les fonctions obtenues à l'issue de la mise en œuvre de l'une ou l'autre méthode sont distinctes. L'approche objet est une approche orientée données. Dans cette approche, les fonctions se déduisent d'un regroupement de champs de données formant une entité cohérente, logique, tangible et surtout stable quant au problème traité. L'approche structurée classique privilégie une organisation des données postérieures à la découverte des grandes, puis petites fonctions qui les décomposent, l'ensemble constituant les services qui répondent aux besoins.

En approche objet, l'évolution des besoins aura le plus souvent tendance à se présenter comme un changement de l'interaction des objets. S'il faut apporter une modification aux données, seul l'objet incriminé (encapsulant cette donnée) sera modifié. Toutes les fonctions à modifier sont bien identifiées : elles se trouvent dans ce même objet : ce sont ses méthodes.

Dans une approche structurée, l'évolution des besoins entraîne souvent une dégénérescence, ou une profonde remise en question, car la décomposition des unités de traitement (du programme principal aux sous-fonctions) est directement dictée par ces besoins.

D'autre part, une modification des données entraîne généralement une modification d'un nombre important de fonctions éparpillées et difficiles à identifier dans la hiérarchie de cette décomposition. La structure d'un logiciel issue d'une approche structurée est beaucoup moins malléable, adaptable, que celle issue d'une approche objet.

# Deuxième partie : Méthodologie d’Analyse et de Conception, et choix méthodologique

## Méthodologie d’Analyse et de Conception

Une méthode d’analyse et de conception est un procédé qui a pour objectif la formalisation des étapes préliminaires de développement d’un système afin de le rendre plus fidèle aux besoins des utilisateurs. Dans ce chapitre nous aborderons les différentes méthodes d’analyse et de conception afin de mieux organiser notre travail.

### Définition des concepts

#### **L’Analyse :**

L’analyse est la première étape de conception de logiciel. Elle permet d’approcher la solution de façon logique et donc garantie un résultat logiquement correcte. Cette approche peut se faire de deux façons :

* Analytique ou cartésienne : consiste à subdiviser les problèmes à résoudre en problème de moindre complexité. Cette approche a surtout pour avantage la faisabilité universelle.
* Systémique ou global : Cette approche permet de garantir une solution logique aux problèmes globales mais sa mise en œuvre est très complexe.

#### **La Conception :**

La conception est la phase créative d’un projet. Cette phase succède à l’analyse des besoins, elle met en œuvre tout un ensemble d’activités qui à partir d’une démarche d’informatisation d’un processus permet de concevoir et mettre en place un programme informatique jusqu’à sa livraison. Le but premier de la conception est de permettre de créer un système ou un processus répondant à un besoin en tenant compte des contraintes.

#### **L’Implémentation :**

L’implémentation décrit les interactions d’éléments dans les langages de programmation. Elle consiste à la mise en œuvre des programmes dans un langage de programmation conforme aux spécifications définies dans la phase précédente.

### L’importance d’une méthode d’analyse et de conception

Le choix d’une méthode d’analyse et de conception dans la conduite d’un projet est très fondamental car ce dernier garantit la réussite du projet. En effet, l’objectif principale d’un projet informatique est de fournir un logiciel de qualité dans le délai et à moindre coût tout en respectant le besoin du client. Une méthode d’analyse et de conception permet la construction d’une application fonctionnellement et techniquement conforme aux attentes des utilisateurs.

### Classification des approches ou méthodes d’analyse et de conception

Malgré la diversité des méthodes d’analyse et de conception, il est possible d’énumérer quelques-unes.

#### **Approches / Méthodes cartésiennes ou fonctionnelles**

Les méthodes fonctionnelles ou cartésiennes consistent à décomposer hiérarchiquement une application en un ensemble de sous applications. Les fonctions de chacune de ces dernières sont affinées successivement en sous fonctions simples à coder dans un langage de programmation donné. Le plus haut niveau représente l'ensemble du problème. Chaque niveau est ensuite décomposé en respectant les entrées/sorties du niveau supérieur.

Parmi ces méthodes : SADT, Warnier, ...

Les points forts des méthodes fonctionnelles sont les suivants :

* Simplicité du processus de conception préconisé,
* Capacité à répondre rapidement aux besoins ponctuels de leurs utilisateurs.

Les points faibles des méthodes fonctionnelles sont les suivants :

* Difficulté de fixer des limites pour les décompositions hiérarchiques,
* Éventuelle redondance des données

#### **Approches / Méthodes systémiques**

Les méthodes systémiques proposent une double démarche de modélisation : la modélisation des données et celle des traitements. Elle est influencée par les systèmes de gestion de bases de données. Parmi ces méthodes : MERISE, AXIAL...

Les points forts des méthodes systémiques sont les suivants :

* Approche globale qui prend en compte la modélisation des données et des traitements,
* Introduction des niveaux d'abstraction dans le processus de conception (niveau conceptuel, niveau logique et niveau physique),
* Bonne adaptation à la modélisation des données et à la conception des bases de données.

Les points faibles des méthodes systémiques sont les suivants :

* Double démarche de conception : les données et les traitements,
* Pas de fusion possible des deux aspects (données et traitements).

#### **Approches / Méthodes orientées objets**

Les méthodes orientées objets considèrent le logiciel comme une collection d'objets dissociés définis par des propriétés[[1]](#footnote-2). Elles ont pour objectif de modéliser les besoins des utilisateurs. En effet, on commence par l'identification des objets. Ces objets sont regroupés dans des classes selon leurs propriétés. Ensuite ces classes sont à nouveau regroupées en classes plus abstraites appelées modules ou sous-systèmes jusqu'à la modélisation du problème posé.

Parmi ces méthodes : OMT, BOOCH’93, OOSE

Les points forts des méthodes orientées objets sont les suivants :

* Intégrer dans l'objet des données et des traitements,
* Profiter des avantages des concepts objets : phase d'analyse et de conception
* Prendre en compte une plus large gamme d'applications,
* Favoriser la conception et la réutilisation des composants : concevoir dans un but de réutilisation et non pas pour répondre à un besoin ponctuel,
* Améliorer la productivité et la rentabilité en utilisant des bibliothèques de composants réutilisables,
* Simplifier le passage conceptuel/physique,
* Faciliter le prototypage.

## Choix méthodologique

### Choix d’une méthodologie d’analyse et de conception

Notre choix porte sur la méthodologie orientée objet du fait qu’elle permet une modélisation stable par rapport aux entités du monde réel et une réutilisation des éléments d’un projet à un autre. La simplicité du modèle objet nous a aussi attiré car, pour exprimer de manière uniforme l’analyse, la conception et la réalisation d’une application informatique, il ne nécessite que cinq concepts fondateurs que sont : les objets, les messages, les classes, l’héritage, l’encapsulation et le polymorphisme. Sa capacité à regrouper ce qui a été séparé, à construire le complexe à partir de l’élémentaire, et surtout à intégrer statiquement et dynamiquement les constituants d’un système a été aussi déterminant dans notre choix. La modélisation et les formalismes existent depuis toujours (croquis, plans, maquettes…). La modélisation orientée-objet est apparue durant les années 70 et 80 suite à l’émergence de la programmation orientée-objet. En effet, il était indispensable d’associer à ces nouvelles techniques de développement de nouveaux outils de description adaptés. Entre 1989 et 1994, le nombre de méthodes et de langages de modélisation orientés objet est passé de moins de 10 à plus de 50. Parmi cet assortiment d’outils, trois (3) émergeaient plus particulièrement :

* La méthode Booch (du nom de Grady Booch son concepteur) ;
* La méthode OOSE (pour Object-Oriented Software Engineering crée par Ivar Jacobson) ;
* La méthode OMT (pour Object Modeling Technique conçue par James Rumbaugh).

Face à l’évolution convergente de ces méthodes, leurs auteurs (Booch, Rumbaugh et Jacobson) ont décidé de s’associer au sein de l’entreprise Rational pour créer le langage UML (Unified Modeling Language) ; l’objectif de cet outil de modélisation étant d’unir les apports de leurs travaux respectifs. Le formalisme UML est « stabilisé » depuis 1997 (avec la mouture UML 1.1, ce qui n’empêche pas les évolutions puisqu’actuellement la toute récente version.

### Présentation d’UML

UML 2.4 est suivi par l’OMG (Object Modeling Group, autorité de normalisation liée à l’objet).

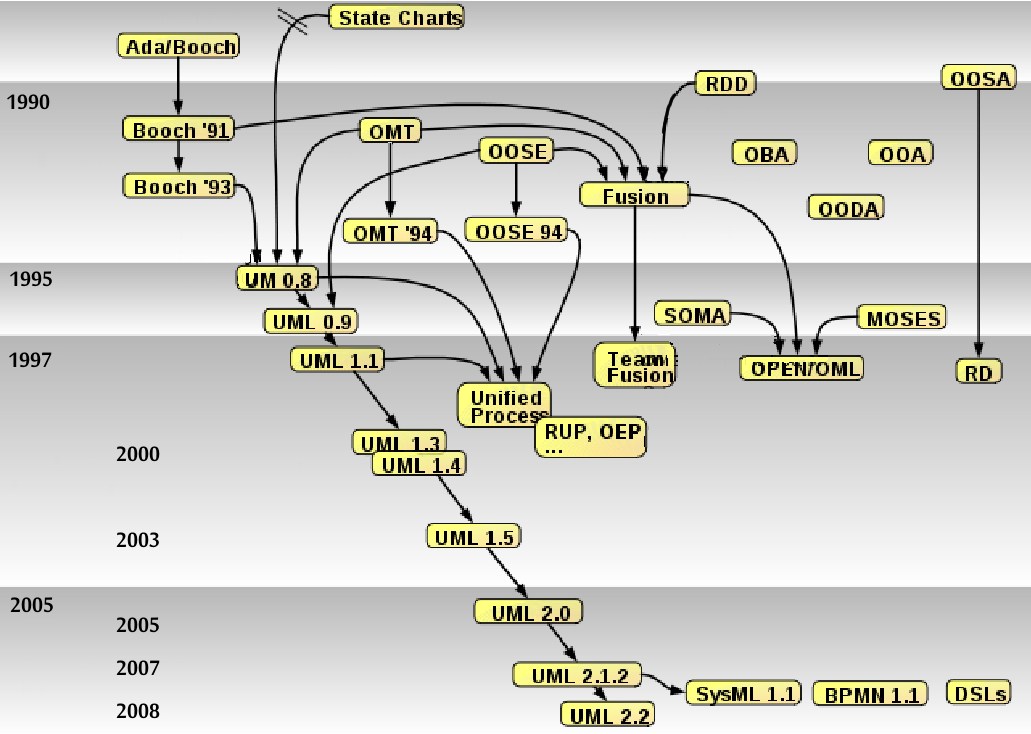


Figure 1 Représentation de l’évolution de ‘UML’

UML ne constitue aucunement une méthode de conception mais bien un langage de modélisation. UML a été conçu selon les objectifs suivants :

* **Objectif longitudinal** : accompagner l’ensemble du cycle de conception ;
* **Objectif vertical** : être adapté à toutes les échelles de modélisation (du plus haut niveau de considération à la prise en compte des détails les plus fins) ;
* **Objectif transversal** : être suffisamment compréhensible et convivial pour être facilement accessible à l’homme et être dans le même temps suffisamment formel et rigoureux pour être adapté au traitement informatique.

Nous allons donner le formalisme de quelques diagrammes que nous allons utiliser dans la suite de notre étude.

#### **Diagramme de cas d’utilisation :**

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer les besoins des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique. Les éléments des diagrammes de cas d'utilisation sont :

* **Acteur** : Les acteurs représentent les éléments externes aux délimitations du système qui sont amenés à interagir avec celui-ci. Ils comprennent aussi bien les acteurs humains (utilisateurs) que les acteurs d’autre type (machine, autre système, tout élément de l’environnement). Un acteur se représente graphiquement sous la forme d’un personnage accompagné de sa désignation.



Figure 2 Représentation d’un acteur du système

* **Cas d'utilisation** : Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service. il se représente graphiquement sous la forme d’une figure ovale complétée de la fonctionnalité représentée.



Figure 3 Représentation d’un cas d’utilisation

* **Association** : Une association est le chemin de communication entre un acteur et un cas d'utilisation.

Les relations admises au sein d’un diagramme de cas d’utilisation sont l’inclusion, l’extension, la généralisation ou spécialisation et le déclenchement.

#### **Diagramme d’activité :**

Les diagrammes d’activités décrivent une phase active du sujet au moyen d’organigrammes regroupant une succession d’étapes organisées séquentiellement. D’un point de vue plus formel, les éléments de vocabulaire dont le modeleur dispose pour réaliser un diagramme d’activité sont les suivants :

Etat initial et état final ;

Etat d’action / état d’activité ;

Branchement conditionnel ;

Synchronisation (fourches et jonctions) ;

Travées (découpes structurantes du diagramme) ;

* Objets (créés ou modifiés par un état).

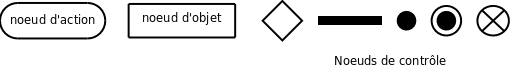


Figure 4 Les différents nœuds utilisés dans un digramme d’activité

**Légende :** De la gauche vers la droite, on trouve : le nœud représentant une action, un nœud objet, un nœud de décision ou de fusion, un nœud d'union, un nœud initial, un nœud final et un nœud final de flot.

#### **Diagramme de classe :**

Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations. Il décrit l’agencement de classes connectées par des relations de différents types et représente les classes intervenant dans le système.

* **Classe :**

Une classe décrit un groupe d’objets ayant la même structure (même ensemble d’attributs) et le même comportement (mêmes opérations). Le formalisme UML décrit une classe en utilisant la symbolique suivante :



Figure 5 Représentation d’une classe en UML

Un attribut représente une propriété de la classe modélisée. Une méthode (opération) est l’implémentation d’un service qui peut être demandé à tous les objets de la classe modélisée.

Le formalisme UML associe aux classes un certain nombre de relations permettant d’expliciter leurs rapports

* **Généralisation / Spécialisation :**

Il s’agit d’une relation entre une classe-mère et une classe-fille qui décrit la relation d’héritage entre la classe-fille et la classe-mère : la sous-classe hérite de l’ensemble de ses propriétés : attributs, méthodes, associations. L’héritage peut être multiple afin de préciser en particulier la nature multiple d’une classe. La représentation graphique est la suivante :

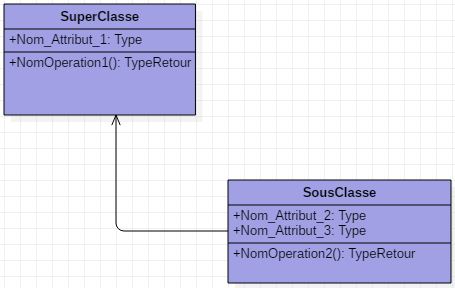


Figure 6 Représentation d’une généralisation entre deux classes en UML

* **Association :**

Cette relation permet de donner du sens à un lien organisationnel entre classes. Une association peut concerner une seule classe (association unaire) ou plusieurs classes (association binaire ou n-aire). La représentation graphique est la suivante :

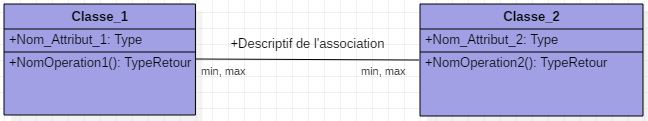


Figure 7 Représentation d’une association entre deux classes en UML

* **Agrégation :**

Une agrégation est une association qui représente une relation d'inclusion structurelle ou comportementale d'un élément dans un ensemble. Graphiquement, on ajoute un losange vide du côté de l'agrégat.

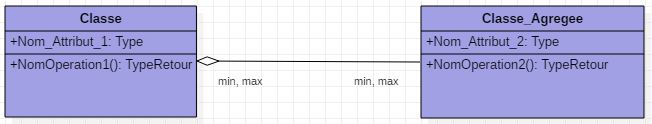


Figure 8 Représentation d’une agrégation en UML

* **Composition :**

La composition décrit une contenance structurelle entre instances. Ainsi, la destruction de l'objet composite implique la destruction de ses composants. Graphiquement, on ajoute un losange plein du côté de l'agrégat.

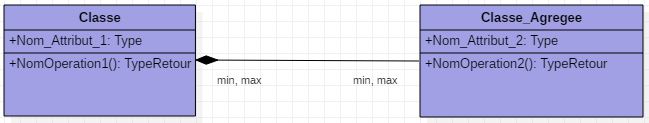


Figure 9 Représentation d’une composition en UML

#### **Diagramme de composants :**

Un composant est un élément logiciel remplaçable et réutilisable qui fournit ou reçoit un service bien précis. Il peut être vu comme une pièce détachée du logiciel. Les plugins, les drivers, les bibliothèques sont des composants.

La notion de composant est proche de celle d’objet, dans le sens de la modularité et de réutilisation avec toutefois une granularité qui peut être différente. Le composant est à l’architecture du logiciel ce que l’objet est à l’architecture du code.

Les composants fournissent des services via des interfaces. Un composant peut être remplacé par n’importe quel autre composant compatible c'est-à-dire ayant les mêmes interfaces. Un composant peut évoluer indépendamment des applications ou des autres composants qui l’utilise à partir du moment où les interfaces sont respectées.

Il existe deux types d’interface :

* Les interfaces requises : Ce sont des interfaces qui fournissent un service au composant et dont il a besoin pour fonctionner.
* Les interfaces fournies : Ce sont des interfaces par lesquels le composant fourni lui-même un service.

Les interfaces requises (représentées par un demi-cercle) et les interfaces fournies (représentées par un cercle) sont raccordées au composant par un trait.



Figure 10 Représentation d’un diagramme de composant en UML

#### **Diagramme de déploiement :**

Le diagramme de déploiement[[2]](#footnote-3) est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Il décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds.

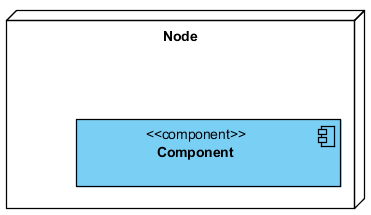


Figure 11 Représentation d’un diagramme de déploiement en UML

# Troisième partie : Expression des besoins

## Étude du cahier des charges

### Problématique

Durant ces derniers mois encore plus qu’avant l’outil informatique a connu un impact considérable notamment dans tout ce qui est commerce en ligne.

En effet, une multitude de site e-commerce a vu le jour si bien qu’il est très difficile d’avoir un site de vente en ligne et atteindre certains objectifs de vent et cela dit pour plusieurs raisons parmi lesquelles on peut citer :

* La pluralité de sites proposant les mêmes services et produits ;
* Le doute au niveau des clients quant à la fiabilité de certains sites ;
* Les problèmes de maintenance et failles de sécurité ;

### Expression des besoins

Le cahier des charges fonctionnel est un document qui permet de formaliser avec précision le besoin[[3]](#footnote-4) du demandeur. Avant d’imposer une solution, il est toujours important vers ce dernier pour aboutir de manière structurée à la solution. En effet, le but d’un projet est de satisfaire le besoin du client, il est donc important pour nous de bien cerner les objectifs à atteindre afin d’éviter toute confusion.

Le chef d’entreprise étant ici représenté comme étant une personne morale nous allons directement proposer des solutions relatives aux besoins que l’on retrouve dans notre projet à savoir :

* L’enregistrement et la connexion des fournisseurs au sein de la plateforme ;
* La gestion des produits en fonction des commandes effectuées par les clients ;
* La gestion des produits en fonction de leur stock ;

### Démarche de résolution

Pour contourner les difficultés ci-dessus citées, améliorer la qualité des données collectées et réduire le temps de traitement, nous avons fait le choix d’introduire l’utilisation des outils informatiques (Tablette, Ordinateur) et des techniques de traitement de l’information et de la communication dans la gestion commerciale en lieu et place du support papier.

Notre travail consistera à simplifier le processus de collecte de données pendant les différentes opérations ou transactions en remplaçant les anciens formulaires papier par des formulaires électroniques qui permettront des informations sur notre serveur en temps réels. Ainsi il sera possible d’héberger les données dans une base de données propre au projet et surtout sécurisée, de gérer ces données et d’en avoir accès à tout instant.

Il est donc attendu du système qui sera mis en place :

* D’une interface de gestion des fournisseurs, qui permettra à tout fournisseur de pouvoir s’enregistrer, se connecter.
* De différentes interfaces d’informations relatives à l’identité ainsi que les activités de la plateforme.
* D’une interface permettant l’ajout des produits proposés par les différents fournisseurs.
* D’une interface contenant la liste de tous les produits disponibles en fonction des catégories auxquelles ils appartiennent
* D’une interface de commande permettant d’avoir des détails sur un produit spécifique
* D’une interface de constituant le panier du client
* D’une interface permettant au client d’effectuer la commande d’un produit en renseignant les détails nécessaires

## Identification des acteurs du système

Vu le cahier de charge que nous avons un peu plus haut, avec de récurrentes discussions tenues avec les futurs utilisateurs de l’application, pour répondre aux exigences de ceux-ci, les principaux acteurs du futur système sont :

* L’administrateur : C’est celui qui se charge de la création de la plateforme et de sa maintenance, il se également de la gestion des fournisseurs et des produits.
* Le fournisseur : C’est celui qui vient proposer un produit ou service à vendre au niveau de la plateforme
* Le client : C’est celui qui vient consulter la plateforme afin d’effectuer les achats qui conviennent à ses besoins

# Quatrième partie : Réalisation et Implémentation

## Analyse et conception

### Diagrammes de Cas d’utilisation

#### **Cas d’utilisation globale du système**

Pour répondre aux diverses exigences du projet il est nécessaire de développer les fonctionnalités suivantes : un module pour la gestion des employés, un module pour la gestion des volontaires, un module pour la gestion des clients et de leurs projets respectifs, un module pour les adoptions.

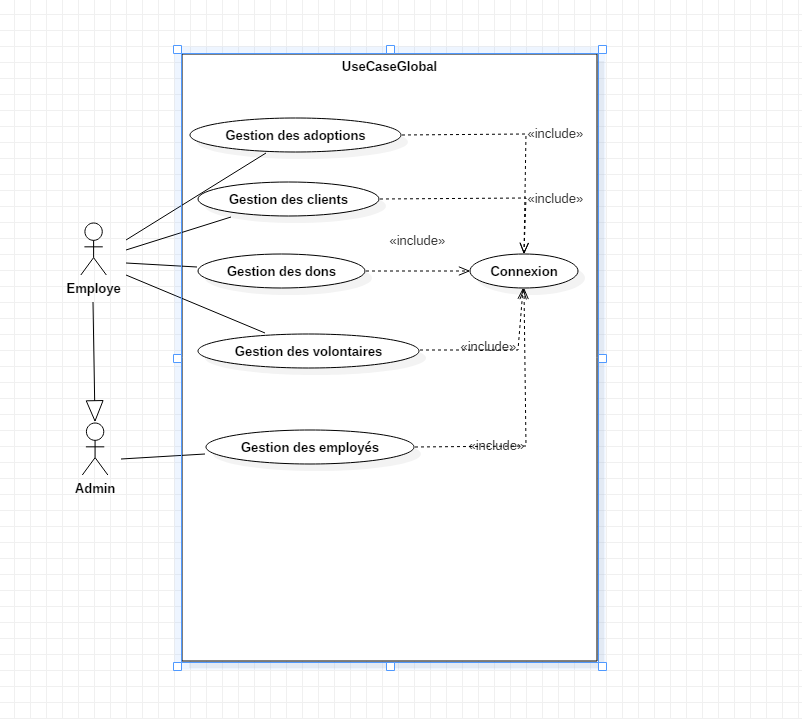


Figure 12 Représentation du diagramme de cas d’utilisation global du système

#### Gestion des employés

Un employé[[4]](#footnote-5) est une entité extérieure au système et devant pouvoir y accéder par un système d’authentification. Tout compte d’utilisateur doit être crée par l’administrateur qui donne le(s) profile(s) adéquat(s) à chaque personne. Certaines informations, comme le mot de passe, sont confidentielles et même ce dernier ne doit pas avoir la main mise dessus. Par exemple, lors de la création d’un compte d’utilisateur, un mot de passe doit être généré de façon aléatoire, crypté avant d’être enregistré dans la base de données puis envoyé en clair à l’utilisateur détenteur par email. Ce module permettra aussi de gérer l’attribution et la révocation de rôle à un utilisateur, l’activation comme l’inactivation d’un compte utilisateur.

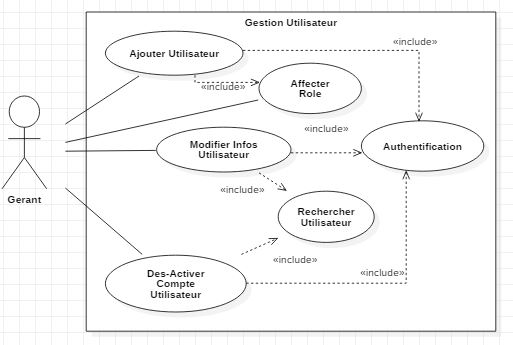


Figure 13 Diagramme de cas d’utilisation de la gestion des utilisateurs

#### Gestion Ventes de Produits

La vente est l’une des phases clés de la gestion de l’entreprise. Ce module permettra donc aux différents employés d’effectuer chaque opération de vente. En choisissant tous les produits concernes, et la quantité respective de chaque produit choisi. L’avantage est que les prix de tous les produits sont renseignes à l’ avance dans la base de donnée, ce qui veut tout simplement dire que l’employé n’aura pas à saisir les prix. L’autre avantage est qu’après chaque vente, le système sera en capacité de générer une facture sur laquelle toutes les informations de la vente y sont inscrites, en précisant la quantité de chaque produit choisi, ainsi que le montant total de la vente. Et finalement le règlement de la facture se fera en espèces au niveau du gérant lui-même.

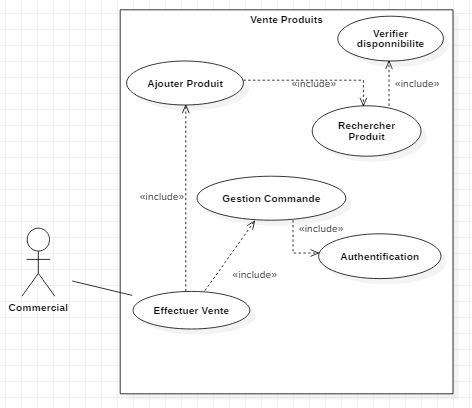


Figure 14 Diagramme de cas d’utilisation de la gestion de vente

#### Gestion Clients

Dans ce diagramme de cas d’utilisation de gestion des clients, l’administrateur et le gérant ont la possibilité d’ajouter un client, de consulter la liste des clients, de mettre à jour les informations relatives à n’importe quel client et le système exige a tout moment à l’utilisateur de se connecter à travers son login et son mot de passe.

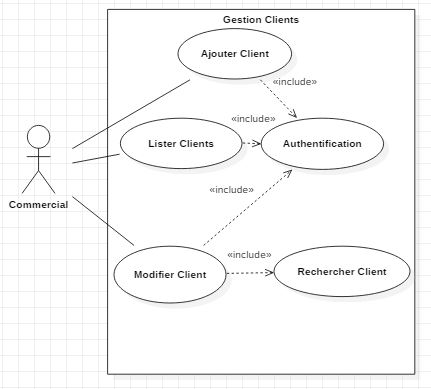


Figure 15 Diagramme de cas d’utilisation de la gestion des clients

#### Gestion Approvisionnement / Commande

Apres la vente, l’approvisionnement est l’une des phases clés de la gestion de l’entreprise, dans le cadre ou l’on ne peut effectuer de ventes sans approvisionnement. L’administrateur étant le seul à avoir accès à cette section, pouvoir donc établir des fiches de commande il fera parvenir aux différents fournisseurs pour des livraisons ultérieures. Dans l’établissement de la commande, l’administrateur n’aura plus à saisir à nouveau le nom des produits qui existent déjà dans la base, il ne fera que les rechercher et en suite les ajoutes a la commande, sinon il leur ajoute dans la base de données. Il a aussi la possibilité de faire toutes les opérations possibles sur une commande, à savoir : l’ajout, la modification, la suppression,…

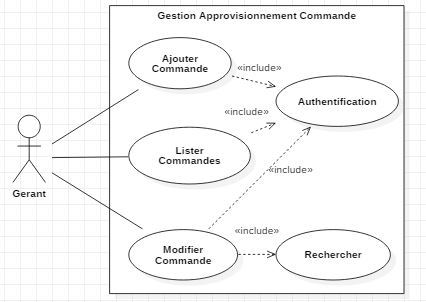


Figure 16 Diagramme de cas d’utilisation de la gestion d’approvisionnement

#### Gestion Fournisseurs

Dans ce scenario, l’administrateur a la possibilité d’ajouter un fournisseur, de consulter la liste des fournisseurs, de mettre à jour les informations relatives à un fournisseur tout en passant aussi par une recherche, ce qui lui permettra même de gagner beaucoup plus de temps et le système exige à tout moment à l’utilisateur de se connecter à travers son login et son mot de passe.

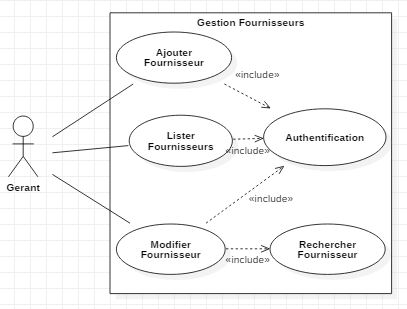


Figure 17 Diagramme de cas d’utilisation de la gestion des fournisseurs

#### Gestion Stock

Dans ce cas, l’administrateur a la possibilité de faire toutes les mises à jour possible[[5]](#footnote-6) concernant n’importe quel produit dans le cas où le système lui donne l’opportunité de mettre à jour un produit tout en le recherchant à travers un champ de saisi, de consulter les produits existant en vérifiant les différentes quantités restantes pour chacun de ces produits et également de vérifier la conformité entre les produits livrés et les produits commandés.

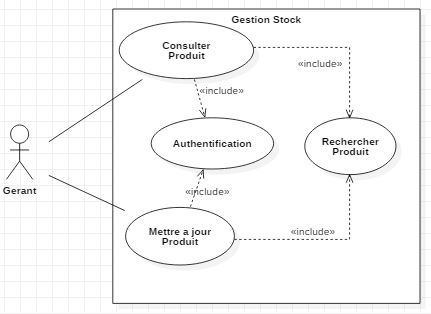


Figure 18 Diagramme de cas d’utilisation de la gestion du stock

### Diagrammes d’activité des Cas d’utilisation

Dans cette partie dédiée aux diagrammes d’activités des cas d’utilisation nous allons surtout mettre l’accent sur les cas qui concernent la vente, l’approvisionnement et la gestion du stock qui sont les parties les plus importantes du système, dans le sens où l’on ne peut parler de vente de produits sans qu’il n’y ait d’approvisionnement, mais aussi, pas de gestion de stock s’il n’y a pas de changements au sein des données figurante dans la base de données c’est-à-dire la vente.

#### Diagramme d’activité de la connexion

La connexion est la phase initiale pour l’utilisation de notre application, elle donne la possibilité aux utilisateurs de saisir leurs identifiants (Login[[6]](#footnote-7) et Mot de Passe[[7]](#footnote-8)), à la fin de la saisie de ces informations, le système sera donc capable de vérifier si les identifiants sont corrects afin de le rediriger sur la page d’accueil ou l’informer que les identifiants sont incorrects.

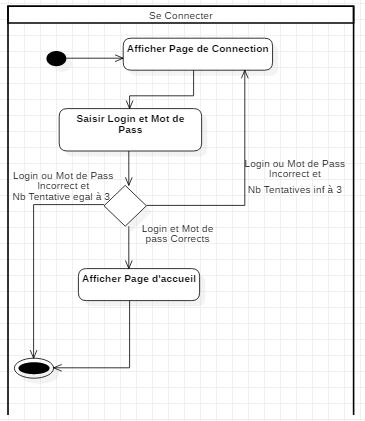


Diagramme d'activité de catégorieFigure 19 Diagramme d’activité de la connexion

#### Diagramme d’active de la vente

L’enregistrement d’une fiche de **vente** se fait après une opération de vente effectuée par le gérant de la boutique ou l’administrateur. Tous les détails de l’opération de vente y sont inscrits, y compris ceux des produits vendus.

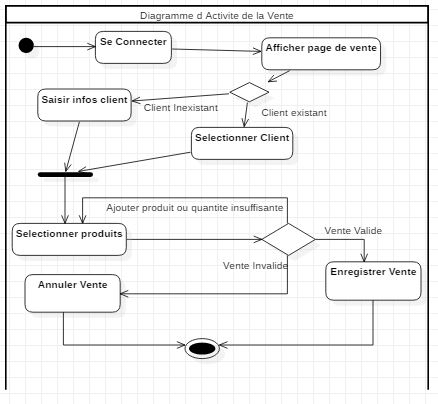


Figure 20 Diagramme d’activité de la vente

#### Diagramme d’active de l’approvisionnement

L’enregistrement d’une fiche d’**approvisionnement** se fait après une ou plusieurs opérations de vente effectuée par le gérant de la boutique ou l’administrateur. Ce qui entraine une rupture de stock, ou encore si l’administrateur veut augmenter la quantité de certains produits ou acheter de nouveaux produits. Tous les détails de la dite commande y seront inscrits, ainsi que ceux du fournisseur concerné.

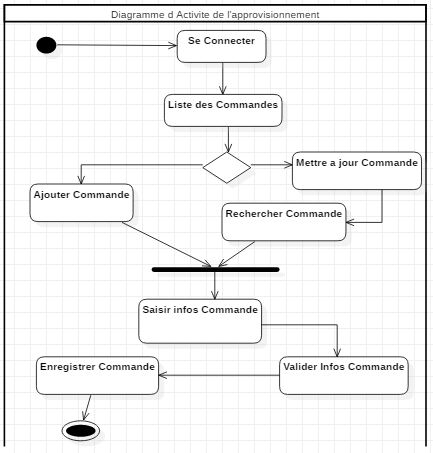


Figure 21 Diagramme d’activité de l’approvisionnement

#### Diagramme d’active de la gestion du stock

La gestion du stock est un scenario effectue par l’administrateur. Scenario lui permettant de vérifier l’état du stock, cette information lui permettra donc de prévoir des fiches de commande qui seront ultérieurement établis.

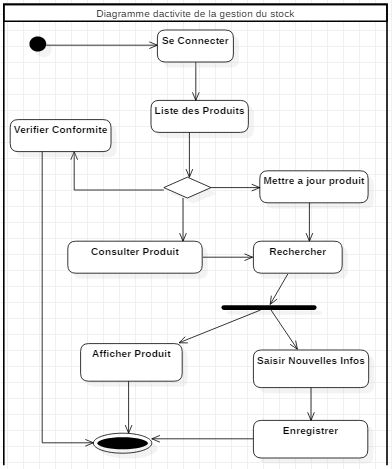


Figure 22 Diagramme d’activité de la gestion du stock

### Diagramme de classe général

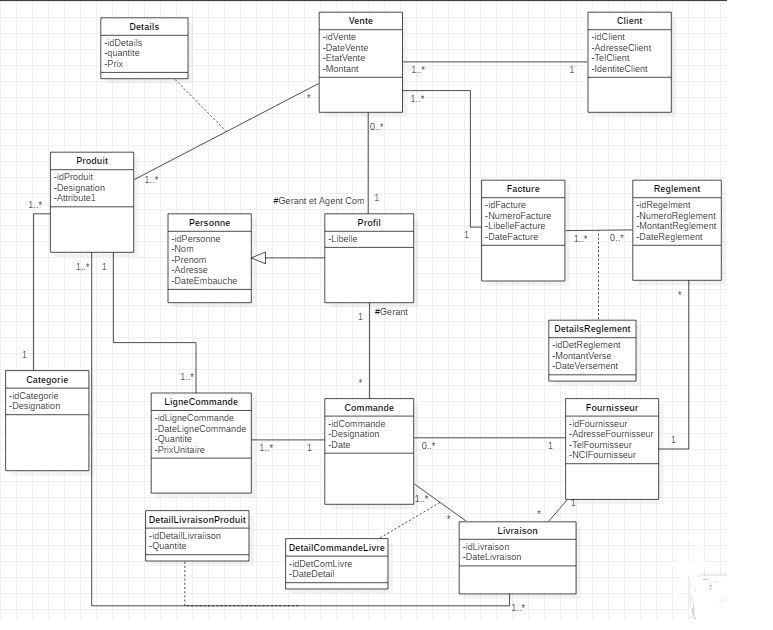


Figure 23 Diagramme de classe general

### Diagramme de déploiement

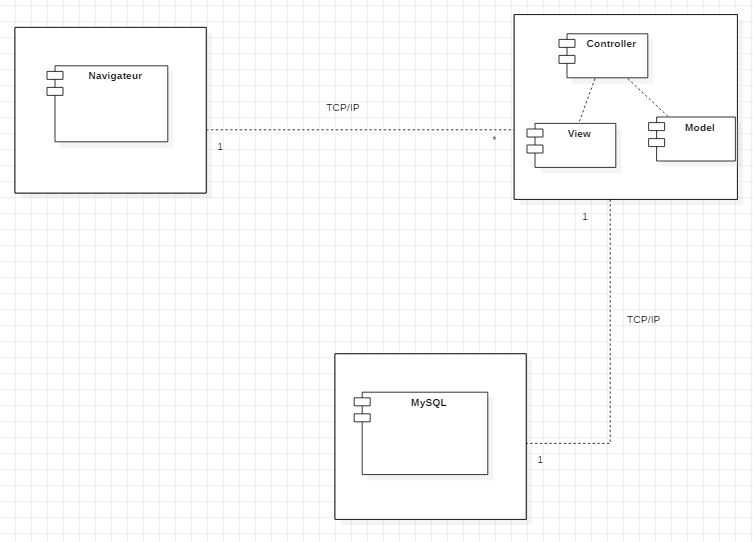


Figure 24 Diagramme de déploiement

## Politique de Sécurité

L'informatique est aujourd'hui un composant critique de l'entreprise. Il est dès lors important de la protéger par des mesures appropriées et de détecter au plus tôt les problèmes de sécurité qui pourraient la menacer. La maîtrise de la sécurité des données et des moyens informatiques de l’entreprise exige la définition d’une politique de sécurité et sa traduction dans des outils informatiques adaptés qui permettent de vérifier sa bonne application. Le système d'information est généralement défini par l'ensemble des données et des ressources matérielles et logicielles de l'entreprise permettant de les stocker ou de les faire circuler. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger. La sécurité des systèmes d'information (SSI) est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité de l'information et du système d'information. La sécurité des systèmes d'information se cantonne généralement à garantir les droits d'accès aux données et aux ressources d'un système, en mettant en place des mécanismes d'authentification et de contrôle. Ces mécanismes permettent d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leur ont été octroyés.

### Sécurité au niveau Applicatif

Il y a un certain nombre de failles très connues et qui pourtant fonctionnent encore aujourd’hui sur de nombreuses applications web. Parmi ces types d’attaque on peut citer la mauvaise gestion de l’authentification et des droits associés, le Cross-Site Scripting (XSS), les injections SQL, l’envoi de fichiers interprétables, l’injection de commandes systèmes, l’exposition d’informations via les messages d’erreur. Pour la sécurité de notre application nous avons suivi quelques règles de bonne pratique du développement en vérifiant toutes les données venant des formulaires, en privilégiant l'utilisation des requêtes POST, en utilisant les requêtes paramétrées, en évitant de réinventer la roue, en sécurisant l'accès aux données.

#### Sécurité de l’application Web

Pour la gestion de la sécurité, des utilisateurs et des rôles, le plugin fournit le script *s2-quickstart* pour générer les classes du domaine, les contrôleurs et les vues associés. L’exécution de la commande *s2-quickstart* ajoute le mapping[[8]](#footnote-9) entre les classes générées et Spring Security dans le fichier Config.groovy. Le plugin nous permet également de générer les CRUD pour les classes User, Role et UserRole (classe générée par le plugin pour attribuer des rôles à un utilisateur donné). L’étape suivante consiste à procéder à la sécurisation des écrans de l’application en définissant des règles de sécurité sur les URLs[[9]](#footnote-10) avec les différents rôles autorisés à y accéder.

#### Sécurité au niveau Base de Données

Dans un système client-serveur, tous les clients accèdent à la base de données, ceci pourrait rendre le serveur vulnérable (plantage de serveur dû aux surcharges des requêtes utilisateurs, ou intrusions de données nocives). Avec une architecture trois tiers, l'accès à la base est rendu possible grâce au serveur d'application, qui est le seul à connaitre la connexion à la base. Ce dernier ne partage aucune information permettant l'accès aux données. De plus, on peut gérer la sécurité au niveau de la base, par exemple en maintenant la liste des utilisateurs et leurs mots de passe, ainsi que leur droit d'accès aux fonctions du système. Dans notre cas, l'accès à la base de données est géré au niveau du serveur d'application. Pour accéder aux fichiers de la base de données nous utilisons L’adresse IP du serveur, le port sur lequel est ouverte la connexion de notre PostgreSQL suivit du nom de la base de données. La connexion d’un utilisateur à la base de données se fait avec un nom d’utilisateur et d’un mot de passe.

#### Sécurité au niveau matériel

Pour assurer la sécurité complète de l’application, la vérification seule du code ne suffit pas. Il faut s’assurer que la base de données de l’application, et toute l’infrastructure soit correctement configurée. Quand on parle de protéger sa machine serveur, on pense souvent aux virus et autres spywares qui viennent polluer le système d'exploitation. Pourtant la protection ne se limite pas au logiciel. Le serveur étant relié au réseau électrique, il est sujet à des variations de tension, des micros coupures, des surtensions ou même une éventuelle panne de courant. Il peut aussi être frappé par la foudre de manière indirecte via la ligne téléphonique ou l'alimentation électrique. Nous proposons l’achat d’un onduleur pour protéger le serveur au niveau électrique. En plus d’avoir un circuit électrique qui empêche toute tension anormalement élevée, les onduleurs peuvent prendre le relais en cas de coupure de l'alimentation électrique. La batterie assure alors une certaine autonomie dépendant de la puissance du PC et de la capacité de l'onduleur. Ils sont aussi capables d'agir sur le système d'exploitation. Via une connexion USB ou série, ils peuvent lancer une application de backup[[10]](#footnote-11) ou un shutdown[[11]](#footnote-12) de la machine. L'autonomie varie très fortement en fonction de la puissance de l'onduleur et des appareils qui y sont connectés.

# Conclusion

Nous voici au terme de notre travail portant sur l'étude et la mise en place d'une application de gestion commerciale, cas de ‘**Abdoul et frères’**. L’introduction de l’utilisation des outils informatiques et de nouvelles techniques de ventes et d’approvisionnement en lieu et support constitue aujourd’hui un véritable moyen pour accroitre la productivité et l’efficacité dans les boutiques grossistes. Ceci a donc pousse les différents services commerciaux à opter pour une informatisation de la procédure de vente et d’approvisionnement de leurs entreprises.

Ce projet de fin de cycle avait pour principal objectif de comprendre d’appliquer et d’expliquer l’ensemble des phases conduisant à la réalisation d’un projet informatique, de l’analyse des besoins du client à la modélisation, du choix technologique au déploiement en passant par le déploiement de la solution.

La réalisation de ce projet fut un important défis pour nous du fait que les technologies utilisées étaient nouvelles, d'où un effort d’auto-formation, de compréhension et d'application du Framework Samane qui est développé par l’un de nos professeurs Mr Ngor SECK.

La solution issue de l’étude est une application web, permettant au gérant et aux agents commerciaux d’effectuer toutes les opérations de vente, d’approvisionnement, de gestion de stock, de commandes, de fournisseurs et de clients avec des interfaces respectives et faciles à utiliser.

Ce travail nous a permis d’améliorer notre connaissance sur de nombreux langages de programmation, de nouvelles technologies et de Framework; certaines bonnes pratiques en génie logiciel.

**Bibliographie**

* Mémoire de Kamile SENE ‘**Etude et mise en place d’un système de surveillance démographique et sanitaire**’. Cas KSHDSS du service de parasitologie et mycologie de l’UCAD.
* Mémoire de Mareme MBoup.
* Mémoire de Mariame NIANG ‘**Mise en place d’un outil de gestion de la relation client {CRM} d’une entreprise commerciale**’. Cas de Casino Sahm.
* Mémoire de Mamadou Alpha DIALLO ‘**Etude et mise en place d’un réseau social**’.

**Webographie**

|  |  |  |
| --- | --- | --- |
| **URL** | **Date de Consultation** | **Heure de Consultation** |
| <https://www.google.com/search?q=historique+du+commerce&oq=historique+du+commerce&aqs=chrome..69i57.6585j0j7&sourceid=chrome&ie=UTF-8> | **18/08/2019** | **12h-45** |
| <https://fr.wikipedia.org/wiki/Histoire_du_commerce> | **18/08/2019** | **12h-48** |
| <http://www.cosmovisions.com/Commerce-Histoire.htm> | **18/08/2019** | **15h-20** |
| <https://www.universalis.fr/classification/economie-et-gestion/histoire-economique/histoire-du-commerce/> | **20/08/2019** | **18h-35** |
| <https://www.etudier.com/dissertations/Histoire-Et-Evolution-Du-Commerce/66334789.html> | **20/08/2019** | **20h-15** |
| https://www.zone-telechargement1.com/autoformations/ | **08/09/2019** | **03h-25** |
| https://astuces-top.blogspot.sn/search/label/R%C3%A9seau%20Informatique?&max-results=5 | **11/09/2019** | **03h-45** |
| https://openclassrooms.com/courses | **11/09/2019** | **04h-00** |
| http://www.commentcamarche.net/contents/ | **13/09/2019** | **17h-35** |
| https://yggtorrent.com/torrents/application/2176-formation | **13/09/2019** | **17h-40** |
| <https://www.google.com/search?q=problematique+des+boutiques+non+informatisees&oq=problematique+des+boutiques+non+informatisees&aqs=chrome..69i57.40935j0j7&sourceid=chrome&ie=UTF-8> | **21/10/2019** | **01-43** |
| <https://www.memoireonline.com/12/06/303/problematique-fraude-fiscale-developpement-republique-democratique-congo.html> | **21/10/2019** | **02h-33** |
| <https://www.dynamique-mag.com/article/informatiser-entreprise-commencer.6892> | **21/10/2019** | **03h-04** |
|  |  |  |

**Table des Matières**

[1 Première partie : cadres théorique et méthodologique x](file:///D:\DEUXIEME%20ANNEE%20ISI\Memoire%20BTS\Memoire%20Licence%202019%20-%20Copie%20Vrai.docx#_Toc23169302)

[Chapitre 1 Présentation de la structure xi](#_Toc23169303)

[Section 1 Présentation générale xi](#_Toc23169304)

[Section 2 Historique xi](#_Toc23169305)

[Section 3 Mission xi](#_Toc23169306)

[Chapitre 2 Cadre méthodologique xii](#_Toc23169307)

[Section 1 Méthodes fonctionnelles xii](#_Toc23169308)

[Section 2 Langage de modélisation oriente objet xii](#_Toc23169309)

[Section 3 Comparaison entre l’approche fonctionnelle et l’approche objet xii](#_Toc23169310)

[2 Deuxième partie : Méthodologie d’Analyse et de Conception, et choix méthodologique xiv](file:///D:\DEUXIEME%20ANNEE%20ISI\Memoire%20BTS\Memoire%20Licence%202019%20-%20Copie%20Vrai.docx#_Toc23169311)

[Chapitre 3 Méthodologie d’Analyse et de Conception xv](#_Toc23169312)

[Section 1 Définition des concepts xv](#_Toc23169313)

[A L’Analyse : xv](#_Toc23169314)

[B La Conception : xv](#_Toc23169315)

[C L’Implémentation : xv](#_Toc23169316)

[Section 2 L’importance d’une méthode d’analyse et de conception xv](#_Toc23169317)

[Section 3 Classification des approches ou méthodes d’analyse et de conception xvi](#_Toc23169318)

[A Approches / Méthodes cartésiennes ou fonctionnelles xvi](#_Toc23169319)

[B Approches / Méthodes systémiques xvi](#_Toc23169320)

[C Approches / Méthodes orientées objets xvii](#_Toc23169321)

[Chapitre 4 Choix méthodologique xviii](#_Toc23169322)

[Section 1 Choix d’une méthodologie d’analyse et de conception xviii](#_Toc23169323)

[Section 2 Présentation d’UML xviii](#_Toc23169324)

[A Diagramme de cas d’utilisation : xx](#_Toc23169325)

[B Diagramme d’activité : xxi](#_Toc23169326)

[C Diagramme de classe : xxi](#_Toc23169327)

[D Diagramme de composants : xxiv](#_Toc23169328)

[E Diagramme de déploiement : xxv](#_Toc23169329)

[3 Troisième partie : Expression des besoins xxvi](file:///D:\DEUXIEME%20ANNEE%20ISI\Memoire%20BTS\Memoire%20Licence%202019%20-%20Copie%20Vrai.docx#_Toc23169330)

[Chapitre 5 Etude du cahier des charges xxvii](#_Toc23169331)

[Section 1 Problématique xxvii](#_Toc23169332)

[Section 2 Expression des besoins xxvii](#_Toc23169333)

[Section 3 Démarche de résolution xxviii](#_Toc23169334)

[Chapitre 6 Identification des acteurs du système xxviii](#_Toc23169335)

[4 Quatrième partie : Réalisation et Implémentation xxx](file:///D:\DEUXIEME%20ANNEE%20ISI\Memoire%20BTS\Memoire%20Licence%202019%20-%20Copie%20Vrai.docx#_Toc23169336)

[Chapitre 7 Analyse et conception xxxi](#_Toc23169337)

[Section 1 Diagrammes de Cas d’utilisation xxxi](#_Toc23169338)

[A Cas d’utilisation globale du système xxxi](#_Toc23169339)

[B Gestion des utilisateurs xxxii](#_Toc23169340)

[C Gestion Ventes de Produits xxxiii](#_Toc23169341)

[D Gestion Clients xxxiv](#_Toc23169342)

[E Gestion Approvisionnement / Commande xxxv](#_Toc23169343)

[F Gestion Fournisseurs xxxvi](#_Toc23169344)

[G Gestion Stock xxxvii](#_Toc23169345)

[Section 2 Diagrammes d’activité des Cas d’utilisation xxxviii](#_Toc23169346)

[A Diagramme d’activité de la connexion xxxviii](#_Toc23169347)

[B Diagramme d’active de la vente xxxix](#_Toc23169348)

[C Diagramme d’active de l’approvisionnement xl](#_Toc23169349)

[D Diagramme d’active de la gestion du stock xli](#_Toc23169350)

[Section 3 Diagramme de classe général xliii](#_Toc23169351)

[Chapitre 8 Politique de Sécurité xliv](#_Toc23169352)

[Section 1 Sécurité au niveau Applicatif xliv](#_Toc23169353)

[A Sécurité de l’application Web xliv](#_Toc23169354)

[B Sécurité au niveau Base de Données xlv](#_Toc23169355)

[C Sécurité au niveau matériel xlv](#_Toc23169356)

[Conclusion xlvii](#_Toc23169357)

**Résumé**

La plupart des boutiques grossistes de nos jours veulent automatiser leurs systèmes de vente et d’approvisionnement. Afin de faciliter ces différentes opérations, la plus part des responsables des différentes boutiques grossistes optent pour l’automatisation de leur système de gestion commerciale.

L’objectif de notre étude est donc de permettre au gérant et aux agents commerciaux de l’entreprise ‘**Abdoul et Frères’** d’effectuer toutes les opérations de vente, d’approvisionnement, de gestion de stock, de commandes avec des interfaces respectives et faciles à utiliser.

A l’issu donc de ce travail, nous avons procédé a l’analyse globale du système que nous avons mis en place en utilisant la méthode UML, et nous avons utilisé le langage de programmation PHP pour mettre en place l’application avec l’accompagnement du Framework Samane.

**Abstract**

Most wholesalers nowadays want to automate their sales and supply systems. In order to facilitate these different operations, most of the managers of the different wholesalers' shops opt for the automation of their commercial management system.

The objective of our study is therefore to enable the manager and sales agents of 'Abdoul et Frères' to carry out all the sales, supply, stock management and ordering operations with respective interfaces Easy to use.

At the end of this work, we proceeded to the global analysis of the system that we implemented using the UML method, and we used the PHP programming language to set up the application with the accompaniment of the Samane Framework.

1. Une propriété est soit un attribut soit une opération [↑](#footnote-ref-2)
2. C’est le fait de transférer une application vers un serveur [↑](#footnote-ref-3)
3. Tout ce que le demandeur souhaite qu’on réalise dans l’application [↑](#footnote-ref-4)
4. C’est un acteur qui interagi directement avec l’application en se connectant en tant que tel [↑](#footnote-ref-5)
5. Ajout, Modification, Suppression et Liste des produits [↑](#footnote-ref-6)
6. Unique a un utilisateur [↑](#footnote-ref-7)
7. Appartient a un seul utilisateur [↑](#footnote-ref-8)
8. La mise en cohérence entre deux types d'informations distincts [↑](#footnote-ref-9)
9. Mots ou suite de mots attachés servant d’adresse pour une page web [↑](#footnote-ref-10)
10. Sauvegarde de données [↑](#footnote-ref-11)
11. Commande permettant l'extinction d’une machine à partir du terminal. [↑](#footnote-ref-12)