**«CONCEPTION ET REALISATION D’UNE APPLICATION WEB DE VENTE DES MATERIELS NUMERIQUES »**

**Bujumbura, Aout 2022**

**Sous l’encadrement de :**

**Msc. UWIZEMANA Providence**

**Par :**

**IRAKOZE Chris Martin**

**Et**

**IZERE Audran**

**UNIVERSITE DU LAC TANGANYIKA**

**MINISTERE DE L’EDUCATION NATIONALE ET DE LA RECHERCHE SCIENTIFIQUE**

**REPUBLIQUE DU BURUNDI**

**FACULTE D’INFORMATIQUE**

**OPTION : GENIE LOGICIEL**

**’’*CONCEPTION ET REALISATION D’UNE APPLICATION WEB DE VENTE DES MATERIELS NUMERIQUES* ’’**

***Cas de Flema Shop***

**C**

Rapport de stage présenté et défendu en vue de l’obtention du Diplôme de Baccalauréat en Informatique

# 

# DEDICACES

- A mes très chers parents,

- A ma regrettée Tante Habonimana Jeanne d’Arc

- A mes frères et sœurs,

- A mes oncles et tantes, Cousins et Cousines

- A toutes mes Connaissances,

- Ainsi qu’à tous ceux qui ont participé à ma scolarité.

**IRAKOZE CHRIS MARTIN**

Je dédie ce travail

- A mes deux chers parents

- A toute ma famille

- A tous mes amis

- Et ainsi qu’à tous ceux qui ont participé dans ma scolarité.

**IZERE Audran**

# REMERCIEMENTS

Tout d'abord, nous sommes très reconnaissants et exprimons également une très grande appréciation à la puissante forteresse notre Dieu par qui ce travail a été rendu possible.

Nos sincères remerciements vont à Madame **UWIZEMANA Providence** qui est notre encadreuse, pour son soutien tout au long de notre Travail. Ses suggestions, ses critiques, son adhésion et ses conseils continus ont fait de notre projet un succès.

Nous remercions tous les enseignants de la faculté d’informatique en particulier ceux de l’option génie logiciel pour leurs efforts visant à mettre à disposition les connaissances requises pendant toute la durée de nos cours.

Nous remercions le supermarché FLEMA SHOP pour ses données, ses informations qui ont aidé à la réalisation de ce travail.

Nous sommes également très redevables à nos amis pour leur soutien et le partage de tout ce qui se développe tout au long de mes études. Enfin, nous remercions tous ceux qui, d'une manière ou d'une autre, ont contribué à la réussite de notre travail. Que Dieu vous bénisse tous !

# LISTES SIGLES ET ABBREVIATIONS

**UML :** Unified Modeling Language

**CAB:** Code à Barre

**OMT:** Object Modeling Technique

**OMG :** Object Modeling Group

**DCU :** Diagram Case Usage

**SQL :** Structure Query Language

**SGBD:** Système de Gestion de base De Données

**PHP:** Hypertext Preprocessor

**MVC:** Modèle Vue Contrôleur

**URL:** Uniform Ressource Locator

**ORM:** Object Relationnal Mapping

**HTTP:** Hyper Text Transfer Protocol

**HTML:** Hyper Text MarkUp Language

**CSS:** Cascading Style Sheet

**ULT :** Université du Lac Tanganyika

**JSON:** Javascript Object Notation

**XML:** Extensible Markup Language

**MSc:**  Master of Science

**OBR:** Office Burundais des Recettes

**UMAMEBU:** Usine de Matériel Métallique au Burundi

# LISTE DES FIGURES

[Figure 1 : Cycle de Vie d’un Projet 4](#_Toc6018)

[Figure 2 : Organigramme Flema Shop 7](#_Toc5796)

[Figure 3 : Représentation d’un Acteur dans un Diagramme 14](#_Toc17690)

[Figure 4 : Représentation d’un Objectif dans un Diagramme 15](#_Toc3160)

[Figure 5 : Représentation association d’inclusion 15](#_Toc31716)

[Figure 6 : Représentation d’une relation d’extension 16](#_Toc28029)

[Figure 7 : Représentation d’une relation de Généralisation 16](#_Toc26522)

[Figure 8 : Représentation diagramme d’activite 17](#_Toc6333)

[Figure 9 : Représentation graphique d'une classe 20](#_Toc23144)

[Figure 10 : Représentation diagramme de séquence 27](#_Toc24760)

[Figure 11 : Exemple d'un diagramme de déploiement 28](#_Toc9211)

[Figure 12 : Diagramme de Cas d’utilisation de notre Système 30](#_Toc15282)

[Figure 13 : Page de Connexion de l’application 38](#_Toc24284)

[Figure 14 : Page d’acceuil de l’application 39](#_Toc7048)

[Figure 15 : Page de Vente du Caissier 39](#_Toc10236)

[Figure 16 : Page de gestion Fournisseurs 40](#_Toc489)

[Figure 17 : Page de gestions des Clients 40](#_Toc29840)

[Figure 18 : Page Gestion des Utilisateurs](#_Toc9410) [41](#_Toc9410)

[Figure 19 : Page Ajout des Produits 41](#_Toc31262)

[Figure 20 : Page Ajout des Categories 42](#_Toc14074)

[Figure 21 : Page Création des Unites de Mesures 42](#_Toc23768)

[Figure 22 : Page approvisionnement de l’article 43](#_Toc10592)

LISTE DES TABLEAUX

**TABLE DES MATIERES**

[DEDICACES I](#_Toc24734)

[REMERCIEMENTS II](#_Toc17050)

[LISTES SIGLES ET ABBREVIATIONS III](#_Toc6351)

[LISTE DES FIGURES IV](#_Toc4133)

[LISTE DES TABLEAUX VI](#_Toc12532)

[CHAPITRE I. INTRODUCTION GENERALE 1](#_Toc7176)

[I.1. Introduction 1](#_Toc3193)

[I.2 Problématique 1](#_Toc6027)

[I.3.Objectifs 2](#_Toc19664)

[I.3.1. Objectif Global 2](#_Toc9859)

[I.3.2. Objectif Spécifiques 2](#_Toc30232)

[I.4. Intérêt du sujet 2](#_Toc21951)

[I.5. Méthodes et Techniques utilisées 3](#_Toc24139)

[I.5.1. Méthodes 3](#_Toc22035)

[I.5.2. Les Techniques utilisés 4](#_Toc25253)

[I.6 Délimitation du sujet 4](#_Toc10398)

[I.6.1. Délimitation dans l’espace 4](#_Toc32087)

[I.6.2. Délimitation dans le temps 4](#_Toc3632)

[I.6.3. Délimitation dans le domaine 4](#_Toc26057)

[I.7. Hypothèse du travail 5](#_Toc23299)

[I.8 Articulation du sujet 5](#_Toc30709)

[CHAPITRE II : PRESENTATION DU MAGASIN FLEMA SHOP 6](#_Toc6439)

[II.1. Présentation générale 6](#_Toc17322)

[II.2. Situation GéographiquePAGEREF \_Toc377298474 \h16 6](#_Toc30537)

[II.3. Organigramme 6](#_Toc27305)

[II.4. Définition et le rôle du code-barres 7](#_Toc11968)

[a. Définition 7](#_Toc27909)

[b. Rôle 7](#_Toc8971)

[II.5. Analyse et Critique de l’existant 9](#_Toc15659)

[II.6. Solutions proposées 9](#_Toc19998)

[CHAP III : LA CONCEPTION DE L’APPLICATION WEB DE VENTE DES MATERIELS NUMERIQUES (UML) 10](#_Toc381)

[III.1 Introduction 10](#_Toc26670)

[III.2 HISTORIQUE 10](#_Toc22368)

[III.3 Choix entre UML et MERISE ? 11](#_Toc28390)

[III.3.1 Présentation des diagrammes UML 11](#_Toc19506)

[III.3.1.1 Les diagrammes de Structure 12](#_Toc3874)

[III.3.1.2 Les diagrammes de comportement 12](#_Toc8840)

[III.3.1.3 Diagramme de cas d’Utilisation 14](#_Toc19876)

[a) Les Acteurs 14](#_Toc28310)

[b) Le Système 15](#_Toc28260)

[c) Les Objectifs (Use Case) 15](#_Toc10937)

[III.3.2 Relations des diagrammes de cas d’utilisation 15](#_Toc22378)

*[Composants de base d’un diagramme d'activités](#_Toc11644)* [17](#_Toc11644)

*[Composants de base d’un diagramme de classes](#_Toc21320)* [19](#_Toc21320)

[III.4 Diagramme de Séquence 26](#_Toc29836)

[III.5. Construction du Diagramme de Séquence 27](#_Toc30906)

[III.6 Diagramme de déploiement 27](#_Toc5234)

[III.7 MODELISATION DU NOUVEAU SYSTEME 28](#_Toc26613)

[CHAPITRE IV : REALISATION DE L’APPLICATION 32](#_Toc23318)

[IV.1 Introduction 32](#_Toc21413)

[IV.2. Choix des outils et technologie utilisés 32](#_Toc25425)

[IV.2.1. Choix du SGBD 32](#_Toc15025)

[IV.2.2. PhpMyAdmin 33](#_Toc2586)

[IV.2.3 Choix du Langage de programmation 33](#_Toc8501)

[IV.2.3.1 Introduction 33](#_Toc31250)

[IV.2.3.3 HTML (Hyper Text Markup Language) 35](#_Toc19039)

[IV.2.3.4 CSS 36](#_Toc26748)

[IV.2.3.5 JavaScript 36](#_Toc16905)

[IV.2.3.6 Choix de l’Editeur (Visual Studio Code) 36](#_Toc14051)

[IV.2.3.7 Navigateur 37](#_Toc15429)

[a) Authentification 38](#_Toc1785)

[b) La page d’accueil 38](#_Toc16959)

[c) La page du Caissier 39](#_Toc31354)

[d) Gestion Client et des Fournisseurs 40](#_Toc7955)

[e) Page de création d’un nouvel utilisateur 41](#_Toc26390)

[e) Page de Création de l’article du magasin 41](#_Toc21655)

[f) Page Création de Catégorie 42](#_Toc18090)

[g) Page de Création de l’ unité de Mesure 42](#_Toc24353)

[h) Page approvisionnement de l’article 43](#_Toc11862)

[CONCLUSION GENERALE ET RECOMMANDATIONS 44](#_Toc2260)

[Conclusion générale 44](#_Toc11060)

[Recommandations 45](#_Toc20473)

[BIBLIOGRAPHIE 45](#_Toc6576)

# 

# CHAPITRE I. INTRODUCTION GENERALE

## I.1. Introduction

Il ne fait désormais aucun doute que la révolution de l'information est la plus importante et la plus novatrice qui ait marqué la vie de l'humanité au cours de ce siècle. En effet, loin d'être une mode passagère ou une tendance passagère, l'informatique vient de nous apporter de nombreux avantages à notre mode de vie. Aucun domaine n'était étranger à cette stratégie qui offre de nombreux services tant pour les entreprises que pour l'administration. Mais au-delà de l'utilisation individuelle de l'ordinateur, c'est surtout la mise en place de la communication entre ordinateurs qui a contribué à révolutionner les méthodes de travail. Ainsi, nous avons assisté à l'émergence de réseaux.

Cette étape apporte de nouveaux outils aux nouveaux utilisateurs et leur permet d'améliorer leur rentabilité et leur productivité. Aujourd'hui, les affaires se font en ligne dans le monde entier, la gestion des institutions se fait grâce à la technologie des réseaux, tous les systèmes de gestion de l'information ont été numérisés. Toutes ces innovations ont pour but de vous simplifier la vie en réalisant beaucoup de choses facilement et en peu de temps. C'est dans ce cadre d'idées que notre projet : "**CONCEPTION ET REALISATION D’UNE APPLICATION DE GESTION DE VENTE DES MATERIELS NUMERIQUES**" qui peut être utilisé par les supermarchés comme mon étude de cas est développé comme travail final pour répondre aux exigences d'obtention d'un baccalauréat en technologie de l'information.

## I.2 Problématique

Une mauvaise gestions des Stocks fait perdre de plusieurs manières , du temps et de l’argent aux flottes, Au vu de l’accroissement du coût des réparations mécaniques, mettre en place en pratiques de gestion des stocks peut éviter, des dépenses additionnelles inutiles

Mais aussi la gestion de vente sur des méthodes classiques comme le papier ou des logiciels inapproprié est un casse-tête. C’est ainsi que l’idée nous est venue de faire une application qui va répondre à son problème d’où le sujet de notre travail de stage. Ce dernier nous parla des issues dans son travail quotidien qui ont finalement fait objet de notre application. Les voici ci-après:

* Les difficultés de savoir l'évolution du stock ;
* Les difficultés de gérer les dépenses faites ;
* Les difficultés de la compilation des tous les rapports de vente  ;
* L'utilisation d'un système non actualisé et non adapté comme Microsoft Access et Excel.
* La sécurité des données

Pour remédier à cela, nous avons estimé nécessaire de concevoir et réaliser une application de gestion de vente afin de répondre à ses problèmes.

## I.3.Objectifs

### I.3.1. Objectif Global

L’objectif principal de notre travail est de proposer une solution aux problématiques de notre sujet. Cela se traduit par la conception et la réalisation d’une application web qui va aider à gérer et vendre les matériels d’un magasin.

### I.3.2. Objectif Spécifiques

Notre travail a pour mission de répondre aux problèmes que font face les commerçants, rapports de vente des articles dont ils disposent. Pour ce, nous aurons à atteindre les objectifs spécifiques suivants :

* Mise en place d’une base des données : cette base de données contiendra tous les articles qui sont disponible dans le magasin
* La confidentialité (sécurité) des données : mettre en place une interface d’authentification pour que chaque utilisateur s’identifie préalablement avant d’utiliser l’application.
* Assurer l'intégrité des données archivées afin de garantir leur lisibilité, la stabilité de l'information conservée et la traçabilité des actions effectuées sur ces données.
* Implémenter un système moderne d’identification unique du produit par **un code-barres**
* Production des statistiques de vente du magasin

## 

## I.4. Intérêt du sujet

Une fois le travail implémenté, il pourra bien être utile pour nous, pour la société et en général à tous les commerçants, mais également pour les étudiants.

* Pour nous : ce travail nous a permis d’améliorer, d’approfondir les connaissances et notions acquises et appliquer ce que nous avons appris en classe. Cela étant, nous avons voulu améliorer la performance dans l’analyse, la conception, le développement et l’implémentation des applications web gestionnaire de contenu ;
* Pour le supermarché Flema Shop et son domaine de Travail : la réalisation de l’application permettra à notre ami de disposer d’un logiciel de gestion et de vente dans ses articles.
* Pour les étudiants : ils pourront s’en servir pour leur travail de recherche ou pour approfondir d’avantage la nôtre étant donné que le domaine de gestion recouvre beaucoup de choses. Ils pourront trouver une copie du code en ligne sur Github sous le lien : [git@github.com:Kaferas/POS.git](../../../C:/Users/Kazokura/Downloads/git@github.com:Kaferas/POS.git) pour plus de collaboration, avantage ajoute du sujet car le projet est OpenSource.

## I.5. Méthodes et Techniques utilisées

### I.5.1. Méthodes

Le mot méthode peut signifier la poursuite ou la recherche d’une voie pour réaliser quelque chose. Pour atteindre notre objectif de recherche, nous avons utilisé UML comme méthodologie de modélisation et la méthode de prototypage comme approche du cycle de développement du logiciel.

La méthodologie de cycle de développement d’un logiciel adoptée est le « Cycle en V » il est représenté schématiquement comme suit :

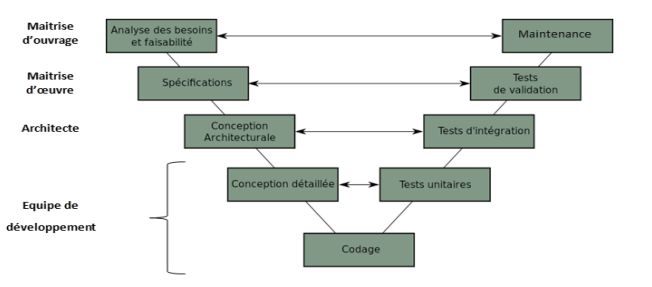


Figure 1: Cycle de Vie d’un Projet

### I.5.2. Les Techniques utilisés

Pour mener à bien notre travail, quelques techniques de collecte et d’analyse des données ont été utilisées. Ce sont les suivantes :

* L’interview : Des interviews avec le personnel du magasin FLEMA Shop; entre autres le coordinateur, le responsable chargé de la gestion de stock et le chargé des ventes ont éclairci la façon dont on a procédé pour avoir l’idée de la structure dont devrait disposer une application de la gestion de stock et de vente ainsi que leurs attentes.
* L’observation : une observation minutieuse a été faite au champ de travail. Elle nous a permis d’observer le processus de vente et du stockage des articles

Nous avons aussi remarqué que le magasin serait prêt à accueillir cette nouvelle technologie d’archivage une fois que l’application serait mise au point.

* La documentation : après une analyse du processus de stockage et de vente on a consulté quelques ouvrages et mémoires, quelques documents, quelques sites web. On s’est donc documenté sur tous les besoins de FLEMA Shop en rapport avec la gestion et la vente pour y parvenir.

## I.6 Délimitation du sujet

Pour tout travail scientifique il faut toujours le délimiter afin d’identifier correctement le travail. Alors ce travail se limite dans le temps, dans l’espace et dans le domaine.

### I.6.1. Délimitation dans l’espace

Notre étude n'a pas couvert tout le magasin de FLEMA SHOP vu le temps imparti. Ainsi donc, nous avons choisi l’agence qui est situé au quartier CARAMA

### I.6.2. Délimitation dans le temps

Dans le temps, notre travail sera basé sur les données de la période allant du mois de mars 2022 jusqu’au jour de test de notre application.

### **I.6.3. Délimitation dans le domaine**

Notre travail se limite à une solution informatique de gestion et de vente des matériels numériques.

## I.7. Hypothèse du travail

Dans le but de trouver une solution aux problématiques que nous avons soulevés, l’hypothèse à confirmer ou à infirmer est que la gestion et la vente des articles informatiques qui existaient déjà est possible à travers une application web. Tout en rendant le personnel de Flema Shop capable à l’utiliser et en améliorant leur condition de travail.

## I.8 Articulation du sujet

Notre travail intitule **« *CONCEPTION ET REALISATION D’UNE APPLICATION WEB DE VENTE DES MATERIELS NUMERIQUES* »**  s’articule de quatre chapitres :

* **Le premier chapitre** parle sur l’introduction générale ;
* **Le second chapitre** est basé sur la description du lieu du travail
* **Le troisième chapitre**concerne la conception de l’application web de vente des matériels numériques
* **Le quatrième chapitre** c’est laréalisation de l’application web de vente des matériels numériques
* En fin, il y a une conclusion générale et des recommandations.[PAGEREF \_Toc377298472 \h16](#_Toc377298472)

# CHAPITRE II : PRESENTATION DU MAGASIN FLEMA SHOP

## **II.1. Présentation générale**

Le magasin Flema Shop est un magasin qui a vu le jour en 2009. Depuis sa création il n’a cessé d’ouvrir de nouvelles franchises et actuellement il s’étant sur toutes les communes de la ville de Bujumbura. Au départ le magasin vendait du matériel de Bureau et quelques années plus tard il a évolué depuis, actuellement il vend également des matériels numériques. En outre, un service de livraison à domicile a été instauré pour ce magasin. Un seul clic et quelques minutes plus tard la commande est livrée.

## II.2. Situation Géographique[PAGEREF \_Toc377298474 \h16](#_Toc377298474)

Le siège social du magasin se situe à la 8ème avenue de la zone Kinama dans la commune Ntahangwa au nord de la province de Bujumbura Mairie. Il poursuit son extension sur toute l’étendue de la mairie de Bujumbura à travers ses trois communes Mukaza, Ntahangwa et Muha. [PAGEREF \_Toc377298475 \h16](#_Toc377298475)

## **II.3. Organigramme**

De nombreux grands magasins utilisent une structure organisationnelle de produit. Ils regroupent leurs départements selon différents types de produits, tels que les articles ménagers, les articles de sport, les vêtements pour femmes et les cosmétiques. Les grands magasins utilisent des structures organisationnelles de produits parce que leurs départements proposent de nombreuses gammes de produits différentes. Les dirigeants préfèrent généralement que les employés se familiarisent avec un groupe de produits pour améliorer le service à la clientèle. Les grands magasins peuvent également utiliser une structure organisationnelle fonctionnelle, surtout s'ils sont plus petits et transportent moins d'articles.

Dans notre cas une structure organisationnelle fonctionnelle serait contrôlée par le directeur du magasin, de l’administrateur, du logisticien et de la comptabilité. L’organigramme de notre magasin est organisé comme suit :

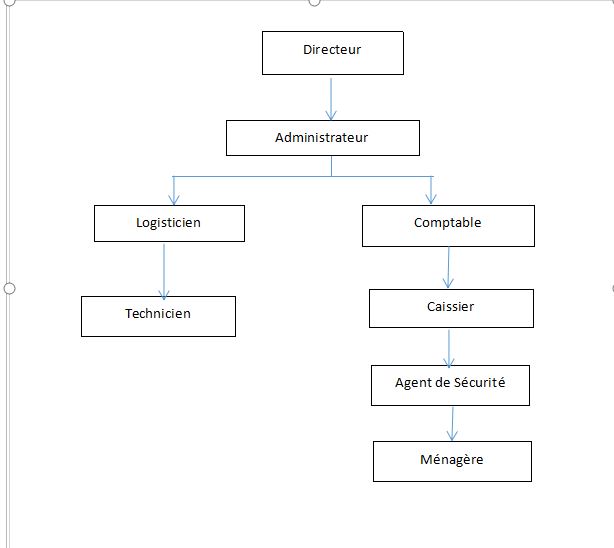


Figure 2 : Organigramme Flema Shop

## [II.4. Définition et le](#_Toc377298478) rôle du code-barres

#### a. Définition

Un code-barres, ou code à barres (CAB), est la représentation d'une donnée numérique ou alphanumérique sous forme d'un symbole constitué de barres et d'espaces dont l'épaisseur varie en fonction du symbole utilisé et des données ainsi codées. Il existe des milliers de codes-barres différents; ceux-ci sont destinés à une lecture automatisée par un capteur électronique, le lecteur barre code. Pour l'impression des codes-barres, les technologies les plus utilisées sont l'impression laser et le transfert thermique.

#### b. Rôle

Les codes-barres se retrouvent dans de nombreuses sphères de l'activité économique, principalement celles qui impliquent la circulation des biens et l'identification. Parmi les principaux usages de cette technologie, mentionnons :

* Suivi des articles dans les magasins à rayons, les grands magasins, les marchés d'alimentation, etc., afin de mieux gérer les inventaires, la facturation et la comptabilité.
* Suivi des envois postaux, comme les colis ou les lettres recommandées
* Suivi des bagages sur les lignes aériennes.
* Identification des patients et des médicaments.
* Indexation de documents.
* Depuis 2005, plusieurs compagnies aériennes utilisent les codes-barres sur les cartes d'embarquement.
* Codage d'hyperliens en code 2D pouvant être lus par un téléphone mobile, ce qui donne accès à une page web.

Après avoir discuté avec notre ami le propriétaire du magasin, nous avons constaté qu’il a le problème de la gestion de vente de ce dernier. En effet, il ne sait pas comment gérer le stock de son magasin mais aussi savoir des articles qu’ils lui restent après la vente.A[PAGEREF \_Toc377298480 \h19](#_Toc377298480)a

Dans le cadre de faciliter la gestion de son magasin, nous avons pensé de mettre sur pied une application qui va résoudre le problème de la gestion du magasin via ces points suivants :

-Mise en place d’une base des données  qui va contenir tous les articles disponible dans le magasin

-La confidentialité (sécurité) des données : mettre en place une interface d’authentification pour que chaque utilisateur s’identifie préalablement avant d’utiliser l’application.

-Assurer l'intégrité des données archivées afin de garantir leur lisibilité, la stabilité de l'information conservée et la traçabilité des actions effectuées sur ces données.

-Implémenter un système moderne d’identification unique du produit par un code-barres

Production des statistiques de vente du magasin

L’utilisation du code-barres qui va se retrouver sur les produits va permettre au personnel du magasin de savoir la quantité d’articles qu’il possède dans le stock mais aussi de savoir combien d’articles vendus et aussi ceux qui sont entrain de dépasser leur période d’utilité. Cela va être possible grâce à la base des données des articles qui sera déjà mis en place.

## II.5. Analyse et Critique de l’existant

Après les entretiens faites auprès du personnel du Magasin, nous avons constaté que le service de gestion et de vente au sein du magasin n’existait pas vraiment de façon systématique et efficace.

En effet avec leur logiciel «Excel», le personnel du magasin stocke régulièrement chaque lot d’articles qu’il vient de recevoir, mais ce lot donc n’est pas classe par ordre d’article mais aussi il est classe en vrac. Sur le long terme, le personnel ne trouvera pas facilement l’article en cas de demande de ce dernier.

En outre, lorsque on fait l’inventaire des articles qui se sont écoules et ceux qui restent dans le stock, cela va constituer un inconvénient parce que on ne pourra pas savoir quel article était sorti et celui qui y reste. Du coût la gestion de vente sera un casse-tête pour le comptable qui va suivre l’évolution commerciale dé chaque article.

Le système de gestion comporte donc pas mal de lacunes sur le point de vue sécurité et gestion du temps de recherche des articles, en plus d’une structure de gestion non arborescente. Notons également qu’il y a encore quelques phases de ce processus qui sont encore analogiques (entre autres les ID).

## II.6. Solutions proposées

La conception et la réalisation d’une application web de gestion et de ventes de matériels informatiques s’avère nécessaire pour un bon fonctionnement du service de gestion. Cette application aura donc pour fonctionnalités principales :

* La gestion électronique des articles informatiques pour la classification, l'indexation, la recherche multicritères, devient facile ;
* L’accès facile et rapide aux articles stockés dans le magasin;
* La sécurité des données ;
* Le gain du temps lors de la recherche d’un article informatique**.**

# **CHAP III : LA CONCEPTION DE L’APPLICATION WEB DE VENTE DES MATERIELS NUMERIQUES (UML)**

## III.1 Introduction

Le langage de modélisation unifié de l’anglais «Unified Modeling Language» (UML) est un langage de modélisation par intermédiaire des diagrammes et des textes qui facilitent le programmeur lors du développement d’une application informatique depuis le besoin originel (analyse) à la phase d’implémentation (codage) et permet aussi à d’autres personnes qui ne sont pas du domaine de pouvoir comprendre le fonctionnement de l’application grâce aux diagrammes conçus dans le but de fournir une méthode normalisée afin de visualiser la conception d'un système.

Grâce au modèle, il est possible de représenter simplement un problème, un concept et le simuler.

La modélisation comporte deux composantes :

• L'analyse, c'est-à-dire l'étude du problème ;

• La conception, soit la mise au point d'une solution au problème.

Le modèle constitue ainsi une représentation possible du système pour un point de vue donné.

Dans notre cas, il a été jugé préférable d’utiliser UML par rapport à MERISE

## III.2 HISTORIQUE

Début des années 1990, plusieurs équipes proposent alors des méthodes qui, pour la plupart, modélisent les mêmes concepts fondamentaux dans différents langages, avec une terminologie, des notations et des définitions différentes.

En 1995, James Rumbaugh, l’auteur de la méthode OMT et Grady Booch, l’auteur de la méthode OOD, publient le document qu’ils ont intitulé « Unified Method V0.8 ». La publication est soumise aux utilisateurs de l’internet pour commentaires .

Il est important de noter que contrairement à ce qui avait été envisagé au départ, le processus de développement a été sorti du champ couvert par le projet de norme. UML est donc une norme du langage de modélisation objet qui a été publiée, dans sa première version, en novembre 1997 par l'OMG , instance de normalisation internationale du domaine de l'objet.

En quelques années, UML s'est imposée comme standard à utiliser en tant que langage de modélisation objet. Aujourd'hui, au milieu de la deuxième décennie des années 2000, nous avons déjà une dizaine d'années de recul sur l'enseignement et la pratique d’ULM en entreprise

## III.3 Choix entre UML et MERISE ?

En ce qui concerne le deuxième critère (précision), **MERISE** est moins préférable. Malgré sa clarté, il manque une précision du fait qu’elle est éloignée du langage donc difficile à implémenter alors qu’**UML** intègre les éléments communs des différents langages, sa volonté est d’être fidèle à la réalisation finale. De plus, la modélisation **UML** permet de vulgariser les aspects liés à la conception et à l’architecture, propres au logiciel, au client. Aussi, elle apporte une compréhension rapide du programme à d’autres développeurs externes en cas de reprise du logiciel et facilite sa maintenance.

### III.3.1 Présentation des diagrammes UML

UML dans sa 2**ème** version propose treize diagrammes qui peuvent être utilisés pour la description d’un système. Ces diagrammes sont regroupés dans deux grands ensembles qui sont :

* **Les diagrammes de Structure :** décrivant les aspects statiques servent à spécifier, visualiser, construire et documenter les aspects statiques d′un système
* **Les diagrammes de comportement :** Les diagrammes de comportement mettent l’accent sur l’aspect dynamique du système ou du processus logiciel. Ces diagrammes présentent la fonctionnalité d’un système et mettent en évidence ce qui est prévu dans le système modélisé.

### III.3.1.1 Les diagrammes de Structure

Ces diagrammes ont comme vocation de représenter l’aspect statique d’un système. Ils permettent d’identifier les objets constituant le programme, leurs attributs, leurs opérations et les méthodes qui leurs sont associés. Ils sont au nombre de six à savoir :

* **Diagramme de Classe** (class diagram) : qui représente la description d′instances des classes, la partie dédiée aux données et celle consacrée au traitement ;
* **Diagramme d’objet**(object diagram) : représentation des instances de classes (objets) utilisées dans le système ; souvent utilisé comme un moyen de revérifier l’exactitude d’un diagramme de classes. Autrement dit, s’il fonctionnera dans la pratique ;
* **Diagramme de composant** (component diagram) : représentation des composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre ([fichiers](https://fr.wikipedia.org/wiki/Fichier_(informatique)" \o "Fichier (informatique)), [bibliothèques](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle" \o "Bibliothèque logicielle), [bases de données](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es" \o "Base de données)…) ;
* **Diagramme de déploiement** (deployment diagram) : qui décrit l′architecture d′un système avec une vue centrée sur la répartition des composants dans la configuration d′exploitation ;
* **Diagramme de Paquetage** (package diagram) : représentation des dépendances entre les paquets (un paquet étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML), c'est-à-dire entre les ensembles de définitions;
* **Diagramme de structure composite** (composite structure diagram) : qui expose la structure interne d’une classe ainsi que les collaborations que cette dernière rend possible.

### **III.3.1.2 Les diagrammes de comportement**

Ces diagrammes ont comme vocation de décrire les aspects dynamiques qui servent à visualiser construire et documenter les aspects dynamiques d′un système, on les appelle des diagrammes de comportement et s′appuient sur sept diagrammes :

* **le Diagramme d′état– transition (state machine diagram) :** qui montre les différents états des objets en réaction aux événements ;
* **le Diagramme d′activités (Activity diagram)**: qui donne une vision des enchaînements des activités propre à une opération ou à un cas d′utilisation. Il permet aussi de représenter les flots de contrôle et les flots de données ;
* **le Diagramme de cas d′utilisation (use-case diagram) :** qui représente la structure des fonctionnalités nécessaires aux utilisateurs du système, ce diagramme est utilisé dans la capture de besoins fonctionnels et techniques ;
* **le Diagramme de séquence (Séquence diagram) : qui** permet de décrire les scénarios des cas d′utilisation en mettant l′accent sur les objets et les messages échangé ;
* **le Diagramme de collaboration (communication)** **(communication diagram)**: qui est une autre représentation des scénarios des cas d′utilisation qui met plus l′accent sur la chronologie des opérations en interaction avec les objets ;
* **Le Diagramme global d′interaction (interaction overview diagram) :** qui fournit une vue générale des interactions décrites dans le diagramme de séquences et des flots de contrôle décrits dans le diagramme d′activités ;
* **Le Diagramme de temps (timing diagram)**: qui permet de représenter les états et les interactions d′objets dans un contexte ou le temps a une forte influence sur le comportement du système à gérer.

Parmi les diagrammes cités ci-haut, nous avons opté à utiliser cinq à savoir :

* Diagramme de cas d’Utilisation
* Diagramme d’activité
* Diagramme de classe
* Diagramme de séquence
* Diagramme de déploiement

### III.3.1.3 Diagramme de cas d’Utilisation

Les **diagrammes de cas d'utilisation** (DCU) sont des **diagrammes** UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les **cas d'utilisation** sont plus appropriés.

Les composants de base des diagrammes de cas d’utilisation sont :

* Les **acteurs** ;
* Le **système**
* Les **objectifs**

#### Les Acteurs

**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.

Du point de vue du système, on peut différencier deux types d’acteurs :

• Primaire : utilise le système ; met en œuvre un ou plusieurs fonctionnalités du système; répond aux différentes questions : à qui va servir le système ? Qui va l’utiliser ? Qui le système doit-il aider ?

• Secondaire : administre le système ; possède un rôle d’administration et de maintenance, paramètre le système et lui fournit toutes les informations nécessaires à son bon fonctionnement pour les acteurs primaires ; répond aux différentes questions : qui gère le système ? Qui l’administre ? Qui le paramètre ?



Figure 3 : Représentation d’un Acteur dans un Diagramme

#### **Le** **Système**

**Le système :** séquence spécifique d'actions et d'interactions entre les acteurs et le système. Un système peut également être appelé scénario. Souvent il est représenté par un rectangle.

#### Les Objectifs (Use Case)

**Les objectifs :** résultat final de la plupart des cas d'utilisation. Un diagramme réussi doit décrire les activités et les variantes utilisées pour atteindre l'objectif. Représenté par une ellipse constituant un moyen de recueillir, de décrire les besoins des acteurs du système, il définit les fonctionnalités utilisables pour un acteur.



Figure 4 : Représentation d’un Objectif dans un Diagramme

## 

### III.3.2 Relations des diagrammes de cas d’utilisation

Il existe 3 relations importantes entre un acteur et un cas d’utilisation :

**L’association d’inclusion (include) :** le cas d’utilisation d’où part la ligne de connexion en pointillés inclut un deuxième cas d’utilisation, pointé par la pointe de flèche.

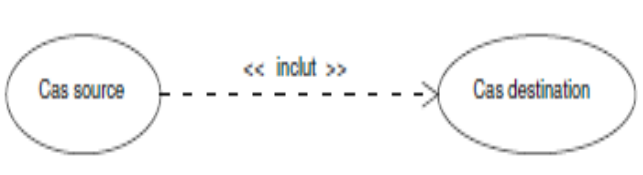


Figure 5 : Représentation association d’inclusion

Dans notre cas le cas source, inclut un cas d’utilisation, appelé cas destination, si les comportements décrits par le cas source contiennent les comportements décrits par le cas destination

1. **Extension d’un cas d’utilisation (lien extend)**

**L’association d’extension :** le cas d’utilisation d’où part la ligne de connexion en pointillés commence peut, sous certaines conditions, étendre le cas d’utilisation pointé par la tête de flèche. Mais ce n’est pas toujours le cas.

* Un cas d’utilisation, appelé cas source, **étend** un cas d’utilisation, appelé cas destination si le comportement décrits par le cas source étend les comportements décrits par le cas destination.

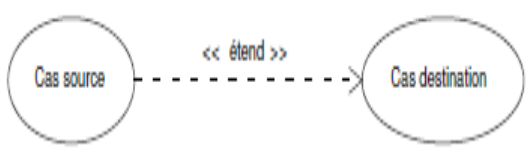


Figure 6 : Représentation d’une relation d’extension

1. **Généralisation entre deux cas d’utilisation (le lien généralisation)**

* Un cas d’utilisation, appelé cas d’utilisation parent, généralise un cas d’utilisation, appelé cas d’utilisation enfant, lorsque les comportements décrits par le cas d’utilisation parent spécialisent les comportements décrits par le cas d’utilisation enfant.
* Une généralisation indique que le cas d’utilisation enfant est une variation du cas d’utilisation parent.
* Les généralisations entre cas d’utilisation sont assez rarement utilisées.

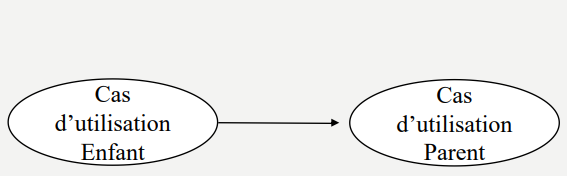


Figure 7 : Représentation d’une relation de Généralisation

**III.3.2.2 Diagramme d’Activité**

Le langage de modélisation unifié regroupe plusieurs sous-catégories de diagrammes, notamment les diagrammes de structure, les diagrammes d'interaction et les diagrammes comportementaux. Les diagrammes d’activités, ainsi que les diagrammes de cas d’utilisation et d'états-transitions sont considérés comme des diagrammes comportementaux, car ils décrivent ce qui doit arriver dans le système modélisé. Dans une entreprise, les diagrammes d’activités aident les différents intervenants – côté commercial et côté développement – à collaborer pour comprendre un même procédé et un même comportement.

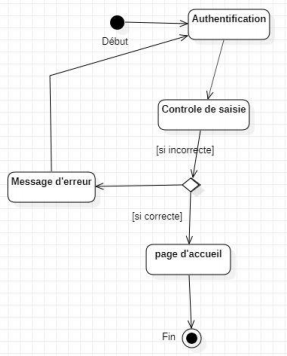


Figure 8 : Représentation diagramme d’activite

Les diagrammes d'activités présentent plusieurs avantages pour les utilisateurs. Songez à créer un diagramme d’activités pour :

* Démontrer la logique d'un algorithme
* Décrire les étapes effectuées dans un cas d'utilisation d'UML
* Illustrer un processus métier ou un flux de travail entre les utilisateurs et le système
* Simplifier et améliorer n'importe quel processus en clarifiant les cas d'utilisation complexes
* Modéliser des éléments de l'architecture de logiciels, tels que la méthode, la fonction et l'utilisation

## ***Composants de base d’un diagramme d'activités***

Voici quelques-uns des composants les plus courants d'un diagramme d'activités

* **Action** : étape dans l'activité où les utilisateurs ou le logiciel exécutent une tâche donnée.
* **Nœud de décision :** embranchement conditionnel dans le flux, qui est représenté par un losange. Il comporte une seule entrée et au moins deux sorties.
* **Flux de contrôle :** autre nom donné aux connecteurs qui illustrent le flux entre les étapes du diagramme.
* **Nœud de départ :** élément symbolisant le début de l'activité, que l'on représente par un cercle noir.
* **Nœud de fin :** élément symbolisant l'étape finale de l'activité, que l'on représente par un cercle noir avec un contour.

**III.3.2.3 Diagramme de Classe**

Le diagramme de classe est l'un des types les plus populaires en langage UML. Très utilisé par les ingénieurs logiciels pour documenter l'architecture des logiciels, les diagrammes de classe sont un type de diagramme de structure, car ils décrivent ce qui doit être présent dans le système modélisé. Le diagramme de classe constitue l’un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d’association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vice versa des autres classes.

Le But est de :

* Mieux comprendre l’aperçu général des schémas d’une application.
* Exprimer visuellement les besoins d'un système et diffuser cette information dans toute l'entreprise.
* Créer des schémas détaillés qui mettent l'accent sur le code spécifique qui doit être programmé et mis en œuvre dans la structure décrite.

## ***Composants de base d’un diagramme de classes***

Le diagramme de classe standard est composé de trois sections :

* **Section supérieure :** contient le nom de la classe. Cette section est toujours nécessaire, que vous parliez du classifieur ou d'un objet.
* **Section intermédiaire :** contient les attributs de la classe. Utilisez-la pour décrire les qualités de la classe. Elle n'est nécessaire que lors de la description d'une instance spécifique d'une classe.
* **Section inférieure :** contient les opérations de la classe (méthodes), affichées sous forme de liste. Chaque opération occupe sa propre ligne. Les opérations décrivent la manière dont une classe interagit avec les données.

Une ébauche de diagramme de classes peut aussi être produite en phase de spécification, après l’analyse du cahier des charges et la production du diagramme de cas d’utilisation afin d’orienter les phases de conception ; on parle alors de diagramme de classes d’analyse. Cette ébauche est par la suite complétée lors de la phase de conception détaillée.

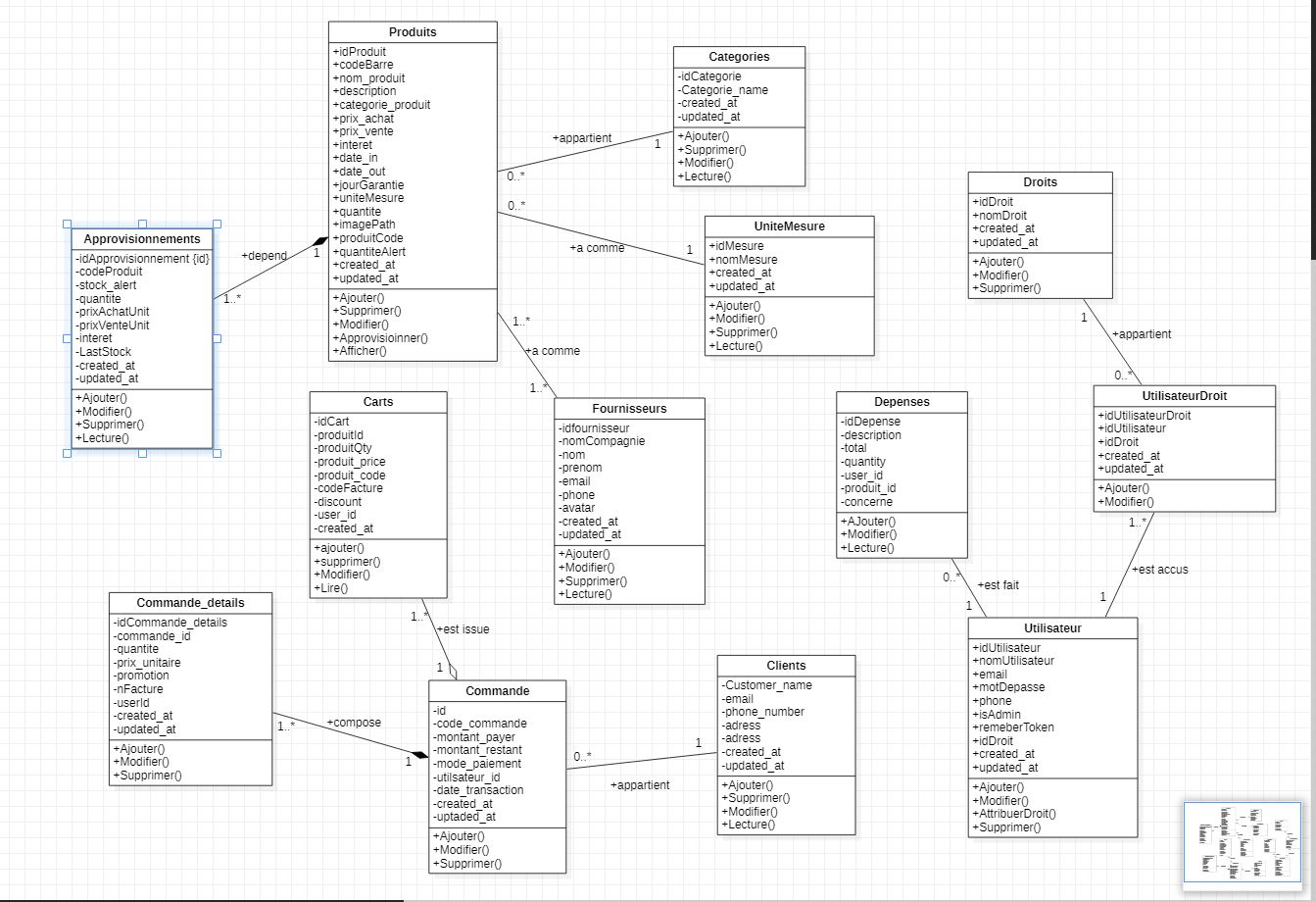


Figure 9 : Représentation graphique d'une classe

|  |  |  |  |
| --- | --- | --- | --- |
| Produit | **Attributs** | **Types et tailles** | **Description de l’attribut** |
| Idproduit | int(11) | Identifiant (matricule) du produit |
| CodeBarre | text | Stockage du codeBarre version visuel |
| Nom\_produit | varchar(255) | Nom du Produit |
| description | text | Mot de passe de l’utilisateur |
| Catégorie\_produit | Int(11) | Clé étrangère |
| Prix\_achat | Int(11) | Prix d’achat article |
| Prix\_vente | Int (11) | Prix de vente article |
| Date\_in | date | Date d’entree article |
| Date\_out | date | Date Expiration Marchandise |
| jourGarantie | int(11) | Nombre jour Garantie Article |
| uniteMesure | int(11) | Unité de Mesure article (Clé étrangère) |
| quantité | Int | Quantité article en stock |
| imagePath | varchar(100) | Image de l’article |
| produitCode | Date | Date de naissance |
| QuantiteAlert | varchar(100) | Quantité qui alertera la fin du stock |
| Created\_at | varchar(100) | Date création article |
| Updated\_at | varchar(100) | Date Modification article |
| UniteMesures | idMesure | Int(11) | Identifiant de unité de Mesure |
| nomMesure | varchar(100) | Nom de l’unité de Mesure |
| Created\_at | Date | Date création article |
| Updated\_at | Date | Date Modification |
| Catégories | idCategorie | Int(11) | Identifiant de l’adresse |
| Categorie\_name | Varchar(100) | Nom de la Catégorie |
| Created\_at | Date | Date Création |
| Updated\_at | Date | Date Modification |
| Fournisseurs | idFournisseur | Int(11) | Identifiant du Fournisseur |
| nomCompagnie | varchar(100) | Nom de la Compagnie |
| Nom | varchar(100) | Nom Fournisseur |
| Prénom | Varchar(100) | Prénom Fournisseur |
| Email | Varchar(100) | Adresse mail Fournisseur |
| Phone | Varchar(50) | Numéro Téléphone |
| Avatar | Varchar(100) | Image Fournisseur |
| Created\_at | Date | Date\_création |
| Updated\_at | Date | Date Modification |
| Approvisionnements | idApprovisonement | Int(11) | Identifiant de l’approvisionnement |
| codeProduit | varchar(100) | CodeBarre du Produit |
| Stock\_alert | varchar(100) | Quantité Min Produit alerte stock |
| quantité | Int(11) | Quantité approvisionne |
| prixAchatUnit | int(11) | Prix d’achat Article |
| prixVenteUnit | Int(11) | Prix de Vente article |
| interet | varchar(100) | Intérêt du produit |
| LastStock | varchar(100) | Stock précédent |
| Created\_at | varchar(100) | Date création article |
| Updated\_at | varchar(100) | Date modification article |
| Panier | idPanier | int(11) | Identifiant Panier |
| produitId | Int(11) | Cle étrangère |
| produitQty | int(11) | Quantite produit demande |
| Produit\_price | Float | Prix du produit |
| Produit\_code | Varchar(100) | Code du produit |
| codeFacture | Varchar(100) | Code du facture généré |
| Discount | Int(11) | Prix promotion |
| User\_id | Int(11) | Utilisateur faisant le panier (Cle étrangère) |
| Created\_at | Date | Date création panier |
| Commande | idCommande | int | Identifiant Commande |
| id | Int(11) | Identifiant commande |
| Code\_commande | varchar(100) | Code de la commande |
| Montant\_payer | Float | Montant payer pour la commande |
| Montant\_restant | Float | Montant restant(Échange) |
| Mode\_paiement | Varchar(100) | Mode Utiliser pour le paiement |
| Utilisateur\_id | Int(11) | Identifiant utilisateur(Clé étrangère) |
| Date\_création | Date | Date enregistrement commande |
| Date\_modification | Date | Date modification commande |
| Commande\_Détails | IdCommande\_détails | int | Identifiant du détails Commande |
| Commande\_id | Int | Identifiant de la commande(Clé étrangère) |
| Quantite | Int(11) | Quantité de l’article acheter |
| Prix\_unitaire | Float | Prix unitaire de l’article |
| Promotion | Float | Prix de l’article de la promotion |
| nFacture | Varchar(100) | Numéro de la facture |
| userId | Int(11) | Utilisateur faisant la commande(Clé étrangère) |
| Clients | idClient | Int(11) | Identifiant Client |
| Customer\_amen | Varchar(100) | Nom du client |
| Email | Varchar(100) | Adresse E-mail Client |
| Phone\_number | Varchar(100) | Numéro telephone Client |
| Created\_at | Date | Date création du client |
| Updated\_at | Date | Date modification client |
| Dépenses | idDepense | Int(11) | Identifiant de la dépensé |
| Description | varchar(100) | Description de la dépensé |
| Total | Float | Montant de la dépense |
| Quantity | Int(11) | Quantité article dépense |
| User\_id | Int(11) | Utilisateur créant la dépense(Clé étrangère) |
| Produit\_id | Int(11) | Identifiant du produit dépense |
| Concerne | varchar(100) | Nom du depenseur |
| situation | varchar(100) | Situation de validation de la déclaration de décès |
| Utilisateur | idUtilisateur | Int(11) | Identifiant Utilisateur |
| nomUtilisateut | Varchar(100) | Le nom de l’utilisateur |
| Email | Varchar(100) | Adresse e-mail de l’utilisateur |
| Mot de passe | varchar(300) | Mot de passe de l’utilisateur |
| Phone | Varchar(100) | Numéro de téléphone Utilisateur |
| isAdmin | Int(11) | Le droit de l’utilisateur (Clé étrangère) |
| rememberToken | Varchar(100) | Token Unique représentant l’utilisateur |
| Created\_at | Date | Date création Utilisateur |
| Updated\_at | Date | Date Modification Utilisateur |
| Droits | IdDroit | Varchar(100) | Numéro matricule de l’officier |
| Nom Droit | Int(11) | Clé étrangère |
| Crevante\_at | Date | Date création Droit |
| Updated\_at | Date | Date Modification Droit |

Tableau Dictionnaire des données

## III.4 Diagramme de Séquence

Les diagrammes de séquence décrivent le déroulement de chaque cas d’utilisation, en montrant la façon dont les diverses entités mises en œuvre dans les cas d’utilisation interagissent et collaborent dans le temps afin de réaliser les fonctionnalités attendues. Le but est de déterminer :

* Les divers entités, appelées objets, mises en jeu dans la réalisation d’une fonctionnalité ;
* Les interactions entre ces divers objets ;
* Le déroulement dans le temps de ces interactions.

Le diagramme de séquence est en réalité un cas particulier de diagramme d’interaction ayant pour but de mettre en avant l’aspect chronologique des interactions décrites. Typiquement, chaque cas d’utilisation déterminé dans le diagramme des cas d’utilisation fait l’objet d’une étude temporelle des interactions en utilisant un diagramme de séquence. Par conséquent, on doit avoir autant de diagrammes de séquence que de cas d’utilisation ; occasionnellement, un cas d’utilisation peut être décrit par plusieurs diagrammes de séquence afin de clarifier l’ensemble.

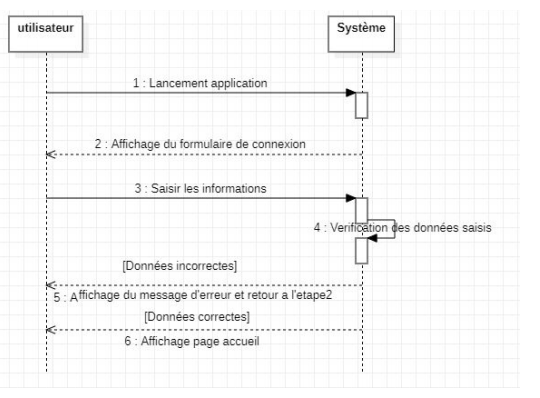


Figure 10 : Représentation diagramme de séquence

## III.5. Construction du Diagramme de Séquence

##### Les objets : Un objet mis en œuvre dans un diagramme de séquence peut symboliser :

* Un acteur, humain ou non-humain ;
* Le système, ou une partie ou composante de celui-ci. Un objet est représenté par un rectangle, ainsi qu’une ligne pointillée verticale portant de ce rectangle et dirigée vers le bas qui représente ainsi la ligne de vie de l’objet. Dans le cas d’un objet symbolisant un acteur, le rectangle peut être remplacé par un personnage.

## III.6 Diagramme de déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. 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.

Ils sont utiles à toutes les phases d’un processus de conception. Ils représentent l’organisation physique du système à déployer. Mais plus souvent encore, ils constituent la photographie de l’organisation existante, c'est-à-dire l’ensemble des contraintes matérielles dans lesquelles il faudra insérer le futur développement. Il doit contenir les détails du système qui sont importants pour la communication avec le client ou l’équipe de développement. Si une fonctionnalité du système n’est pas déterminante, il est inutile de la faire figurer. Elle risque d’encombrer le diagramme et de détourner l’attention des éléments véritablement important.

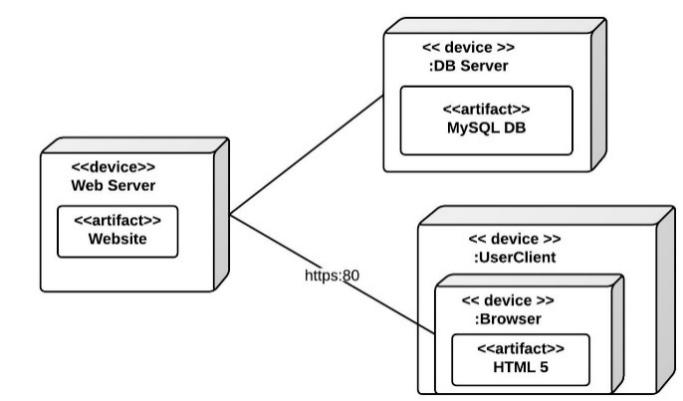


Figure 11 : Exemple d'un diagramme de déploiement

## III.7 MODELISATION DU NOUVEAU SYSTEME

Comme nous l’avons déjà décrit ci haut nous avons opté à utiliser les cinq diagrammes pour des raisons particuliers, bases sur le fait qu’ils sont les plus couramment utilises et les plus faciles à comprendre même pour les non informaticiens, Pour cela on peut classer ces diagrammes dans deux catégories de vues statiques et les vues dynamiques :

* Vues Statiques
  + Diagrammes de cas d’utilisation décrivent le comportement et les fonctions d’un système du point de vue de l’utilisateur ;
  + Diagrammes de classe décrivent la structure statique, les types et les relations des ensembles d’objets
  + Diagrammes de déploiement décrivent la répartition des programmes exécutables sur les différents matériels.
* Vues dynamiques
  + Diagrammes d’activités décrivent le comportement d’une opération (en termes d’actions) ;
  + Diagrammes de Séquence décrivent de manière temporelle les interactions entre objets et acteur
  + Diagrammes de déploiement sont utilisés pour visualiser les processeurs matériels, les nœuds et les dispositifs d'un système, les liens de communication entre eux et le placement des fichiers logiciels sur ce matériel.

**III.6.1 Diagramme de cas d’utilisation**

Le choix de ce dernier repose sur le fait qu’il est le plus facile à comprendre même pour un non professionnel dans le monde de l’informatique et sa facilite de lecture et comprendre ce qui se passe dans le système

*Identification des acteurs*

Un acteur est l’idéalisation d’un rôle joue par une personne externe, un processus ou une chose qui interagit avec le système.

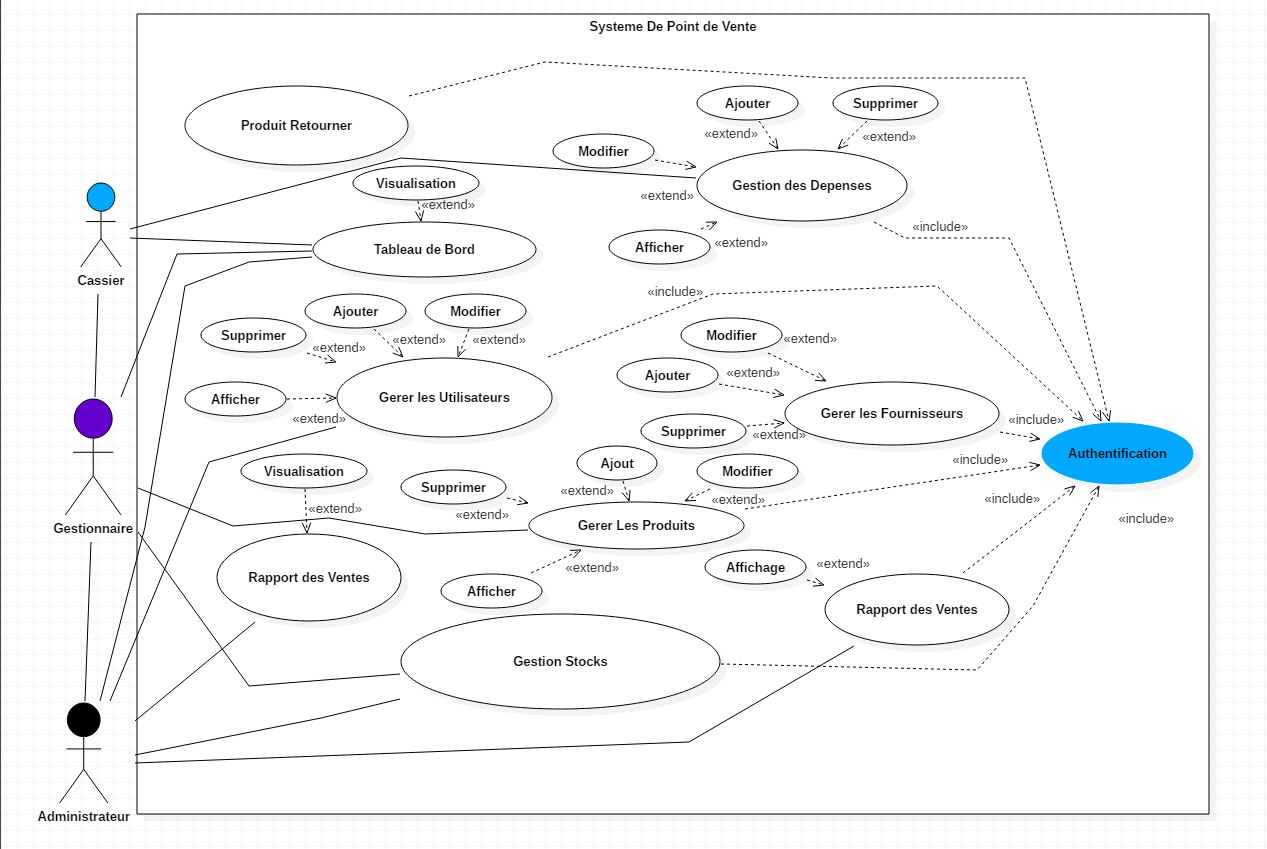


Figure 12 : Diagramme de Cas d’utilisation de notre Système

Dans notre système les différents acteurs y intervenant sont :

|  |  |
| --- | --- |
| **Acteurs** | **Rôles** |
| * Administrateur | -S’authentifier  -Gérer les Produits  -Gérer les Utilisateurs  -Gérer les Commandes  -Visualiser les Rapports  -Gestion Clients et des Fournisseurs |
| * Caissier | * S’authentifier * Vendre * Gérer les dépenses |
| - Gestionnaire Stock | * S’authentifier * Gestion Produits * Gestion Catégories * Gestion Approvisionnement |
|  |  |

**Tableau 2 : Description textuelle de cas d’utilisation**

* **Authentification**
* Le But : Afin d’assurer la sécurité des données et l’application, chaque utilisateur doit s’authentifier pour accéder aux fonctionnalités du système.
* Acteurs : Chacun des acteurs décrits dans le tableau ci haut à un rôle particulier diffèrent de l’autre.
* Condition : Être inscrit parmi l’utilisateur de l’application, lancer l’application.
* **Déroulement Normal**
  + - 1. L’utilisateur saisit son log-in et son mot de passe et valide en appuyant sur le bouton « LOGIN ».
      2. Le système affiche une fenêtre d’accueil suivant le rôle de chacun ainsi que les menus correspondant à ses rôles.

**Déroulement anormal**

1. Si le système perçoit que le nom d’utilisateur et/ou le mot de passe n’a (ont) pas été saisi(s), il se réinitialise automatiquement afin de rediriger vers la page d’authentification.

2. Si le nom d’utilisateur et /ou le mot de passe sont incorrects l’utilisateur reste à la page d’authentification.

# CHAPITRE IV : REALISATION DE L’APPLICATION

## **IV.1 Introduction**

Ce chapitre présente les différents outils utilisés lors de la réalisation de notre application de gestion de vente des matériels électroniques sans oublier l’environnement dans lequel elle a été conçue. Nous allons aussi présenter les différentes fonctionnalités de l’application en illustrant à l’aide des captures d’écran, quelques interfaces décrivant la réalisation du système.

La réalisation nous a exigé d’utiliser UML comme langage de modélisation, Myste comme système de gestion des bases de données relationnelles (SGBDR), Laravel [] comme Framework PHP, Livewire librairie développée pour permettre de construire des interfaces réactives et dynamiques en utilisant Blade et un peu de JavaScript. PHPMyAdmin comme interface de gestion de SGBDR (MYSQL ET MARIADB), Visuel Studio Code comme éditeur de texte et SQL (Structure Query Langage) pour communiquer avec la base des données.

## IV.2. Choix des outils et technologie utilisés

### 

### IV.2.1. Choix du SGBD

Une base de données et un système de gestion de base de données, abrégé en SGBD, forment un système de base de données (Ce dernier terme est toutefois souvent appelé simplement « base de données »). De manière générale, un SGBD est logiciel système servant à stocker, à manipuler ou gérer, et à partager des données dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Notre choix a donc été porté sur le système de gestion de la base de données MySQL. MySQL est un système de données désire après avoir installé et paramètre le système de gestion de base de données correspondant. Les systèmes de gestion de base de données sont fondamentaux dans les bases de données et se démarquent par différents avantages et points forts. Il y a beaucoup de raison à considérer pour préférer MySQL à d’autres systèmes de gestion de base de données :

* Le premier avantage du système de gestion de base de données MySQL est qu’il est multiplate-forme. Les systèmes d’exploitation tels que Windows, Mac, Linux, etc. peuvent utiliser MySQL pour traiter et gérer les données.
* Bien qu’il ne soit pas payant, MySQL dispose d’une assez bonne fonctionnalité mufti-utilisateur. Le système de gestion de base de données SQL peut être utilisé par plusieurs utilisateurs sans rencontrer de problèmes ni de difficultés importantes.
* Vous pouvez utiliser différents types de données dans MySQL, tels que entier, flottant, char, texte, date et horodatage. Avec ces divers types de données, bien sûr, cela peut vous aider et vous faciliter la gestion d’une base de données.
* Bien qu’il ne soit pas payant, le système de gestion de base de données MySQL possède de très bonnes fonctionnalités de sécurité. Avec un système de contrôle d’accès, les administrateurs peuvent déterminer qui a le droit d’obtenir un accès complet à MySQL. Ainsi, l’accès aux données de confidentialité peut être minimisé grâce au contrôle de ce système.

## IV.2.2. PhpMyAdmin

MySQL étant un SGBDR avec beaucoup des fonctionnalités, l’outil phpMyAdmin est une interface qui nous facilite la manipulation graphique de notre base de données notamment l’import et l’export, liaison entre les tables et les contraintes des champs etc...

## IV.2.3 Choix du Langage de programmation

### IV.2.3.1 Introduction

Afin de mettre en place son application ou son site web il faut le/la code du coup un langage de programmation entre en jeu, du coup de nos jours il est rare de trouver un développeur codant son application complet avec un langage Natif, il fait recours au Framework qui est défini comme étant : un modèle de programme fonctionnel qui peut être modifié de manière sélective par l'ajout de code. Et qui utilise des ressources partagées - telles que des bibliothèques, des fichiers d'image et des documents de référence - et les regroupe dans un package. Ce package peut être modifié pour répondre aux besoins spécifiques du projet. Avec un Framework, le développeur peut ajouter ou remplacer des caractéristiques pour donner une nouvelle fonctionnalité à l'application dans la manière la plus simple possible en utilisant le patron de conception MVC.

MVC étant un modèle dans la conception de logiciels. Il met l'accent sur la séparation entre la logique métier et l'affichage du logiciel. Cette « séparation des préoccupations » permet une meilleure répartition du travail et une maintenance améliorée.

Un Framework requiert 3 principaux prérequis afin de faciliter le développement dont :

* **Le système de routage :** les serveurs web et les navigateurs communiquent via le protocole http ; ce qui signifie que les serveurs attendent les requêtes HTTP du navigateur via les liens URL, puis renvoient des informations dans les réponses.
* **ORM (Object Relationnel Mapping) :** est un ensemble des classes permettant de manipuler les tables d’une base de données relationnelle comme s’il s’agissait d’objets, il a comme avantage de ne plus se soucier du système de base de données utilise, c’est l’ORM qui a la charge de transformer les requêtes pour les rendre compatibles avec la base de données
* **Moteur de Template :** un modèle dans la conception de logiciels. Il met l'accent sur la séparation entre la logique métier et l'affichage du logiciel. Cette « séparation des préoccupations » permet une meilleure répartition du travail et une maintenance améliorée.

**IV.2.3.2 PHP (Laravel)**

PHP est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. Inventé en 1994 par **Rasmus Lerdorf**, Le langage s’intègre dans les pages HTML et il est exécuté cote serveur, il demeure le principal langage utilise sur le Web. Du coup le PHP n’a pas cessé d’évoluer donc il n’a pas été à l’écart de l’évolution des Framework du coup il en dispose une dizaine dont les principaux sont : Laravel, Symfony, Zend, CakePHP, Yii, CodeIgniter …. Dans notre cas nous avons opté pour Laravel qui s’avère être le meilleur parmi la liste présente ci-haut.

### IV.2.3.3 HTML (Hyper Text Markup Language)

HTML est un format qui indique à un ordinateur comment afficher une page Web. Les documents eux-mêmes sont des fichiers de texte brut avec des “balises” spéciales ou des codes qu’un navigateur Web utilise pour interpréter et afficher des informations sur l’écran d’un ordinateur. Dans notre cas présent notre Framework Laravel dispose de ce qu’on a appelé un Moteur de Template (**BLADE**), étant un outil de modèle structurel qui simplifie la syntaxe pour assurer une bonne maintenabilité de son projet web en permettant de dissocier la partie présentation (HTML) de la partie programmation (PHP).

Il présente comme avantage :

- Toutes les vues Blade sont compilées dans du code PHP simple et mises en cache jusqu'à ce qu'elles soient modifiées, ce qui signifie que Blade n'ajoute pratiquement aucune surcharge à votre application.

- l’héritage des modèles et les sections. Étant donné que la plupart des applications Web conservent la même disposition générale sur différentes pages, il est pratique de définir cette disposition comme une seule vue Blade page « maître » dont les autres héritera.

- Possibilité de rendre du JSON : parfois, vous pouvez passer un tableau à votre vue avec l'intention de le rendre en JSON afin d'initialiser une variable JavaScript.

- Structure de Control : En plus de l'héritage des modèles et de l'affichage des données, Blade fournit également des raccourcis pratiques pour les structures de contrôle PHP courantes, telles que les instructions conditionnelles et les boucles. Ces raccourcis fournissent une manière très propre et concise de travailler avec les structures de contrôle PHP, tout en restant familier à leurs homologues PHP.

### IV.2.3.4 CSS

**CSS est** un langage de description permettant de gérer la présentation d’une page Web. Il est la recommandation du (W3C) au même titre que le HTML ou XML, il permet de séparer la structure d’un document HTML et sa présentation. En vous obligeant à structure nos pages de façon précise, le CSS vous amène à organiser notre site, ce qui facilitera grandement ses évolutions futures. Cette organisation facilite également l’accessibilité de notre site, c’est-à-dire sa navigation à travers des moyens un peu particuliers. Quant à lui, il présente aussi plusieurs Framework dont les plus pertinents sont : Bootstrap, Bulma, TailwindCss, Material Framework. Et parmi cette liste la nôtre a été le Bootstrap dont la réputation revient à la tête.

### IV.2.3.5 JavaScript

JavaScript est un langage de programmation orientée objet de script principalement employé dans les pages Web, Influence par le Java ainsi que sa syntaxe similaire a C , étant le troisième couche des technologies Standard du web après le HTML et CSS , de nos jours il a beaucoup évolue du coup qu’il est capable les applications mobile, fonctionner cote serveur , cote client et dans les micro processeurs

### IV.2.3.6 Choix de l’Editeur (Visual Studio Code)

**Visual studio code** ou **VS Code** est un éditeur de code développé par **Microsoft** en 2015. Contrairement à ce à quoi Microsoft a eu l’habitude de nous habituer durant des années, il est l’un de ces premiers produits open source et gratuit, et surtout disponible sur les systèmes d’exploitation Windows, Linux et Mac. **VSCode** est développé avec le Framework **Electron** et conçu principalement pour développer des projets avec **JavaScript**, **Node.js** ou encore **Type Script**.

Décider de changer d’éditeur de code n’est pas toujours simple. On possède souvent une solution à laquelle nous sommes habitués, qui nous convient la majorité du temps et on a peur de se retrouver perdu et de perdre par la même occasion en productivité. Pourtant, Visual Studio Code rassure la majorité des nouveaux utilisateurs dès les premières heures d’utilisation.

Facile à installer, à comprendre, à utiliser et rapide, il dispose d’une interface graphique responsive et customisable via des thèmes déjà installés.

Quel que soit le langage : JavaScript, PHP, JAVA, C#, C++ ou autres, **VS code** permet de développer soit via les fonctionnalités par défaut (pour le HTML, CSS, JavaScript, Type script…) ou en ajoutant des extensions disponibles selon les besoins de chacun.

Avec la fonctionnalité « IntelliSense », **VS Code** assiste les développeurs tout au long de leurs développements afin de simplifier l’édition du code, l’appel des propriétés, ou encore sur l’importation des dépendances. Ce système est disponible par défaut raison pour lesquels il a été notre choix prioritaire.

### IV.2.3.7 Navigateur

Un navigateur web est un logiciel conçu pour consulter et afficher les pages web.

Il est composé de divers modules logiciels permettant de communiquer suivant des standard des réseaux, d’un moteur de rendu des standards du Web, d’une interface utilisateur adaptée au système d’exploitation qui l’accueille.

Nous avons utilisé les navigateurs Mozilla Firefox et Chrome.

**IV.3. Présentation des interfaces de l’application**

Nous procédons donc à la création des interfaces qui permettront aux utilisateurs d’interagir avec le système. Afin de voir de manier visuel la représentation des captures d’écran de l’application réalisée en soit.

#### Authentification

Pour des raisons de sécurité, l’utilisateur doit entrer son identifiant et son mot de passe puis il clique sur le bouton valider. Si le login ou le mot de passe est incorrect, la validité échoue. Si le login et le mot de passe entrés correspondant à celle d’un utilisateur.

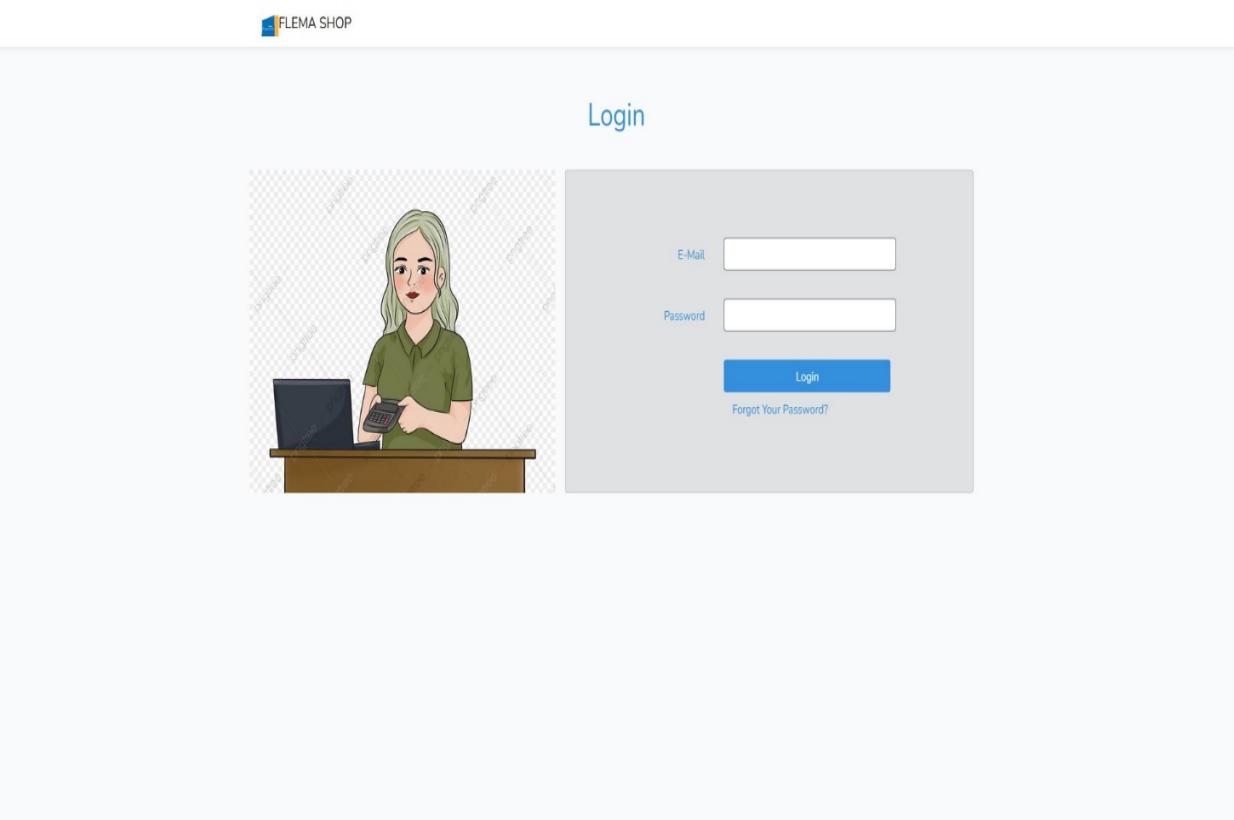


Figure 13: Page de Connexion de l’application

#### La page d’accueil

Après l’authentification de l’utilisateur, une page d’accueil s’ouvre. Celle-ci affiche l’écran de bord de applications qui présente les menus dépendant des droits dont vous possédez ainsi que les cartes au centre montrant le nombre de ventes faites pendant la journée, le nombre globale de clients dont vous disposez et le nombre des articles.

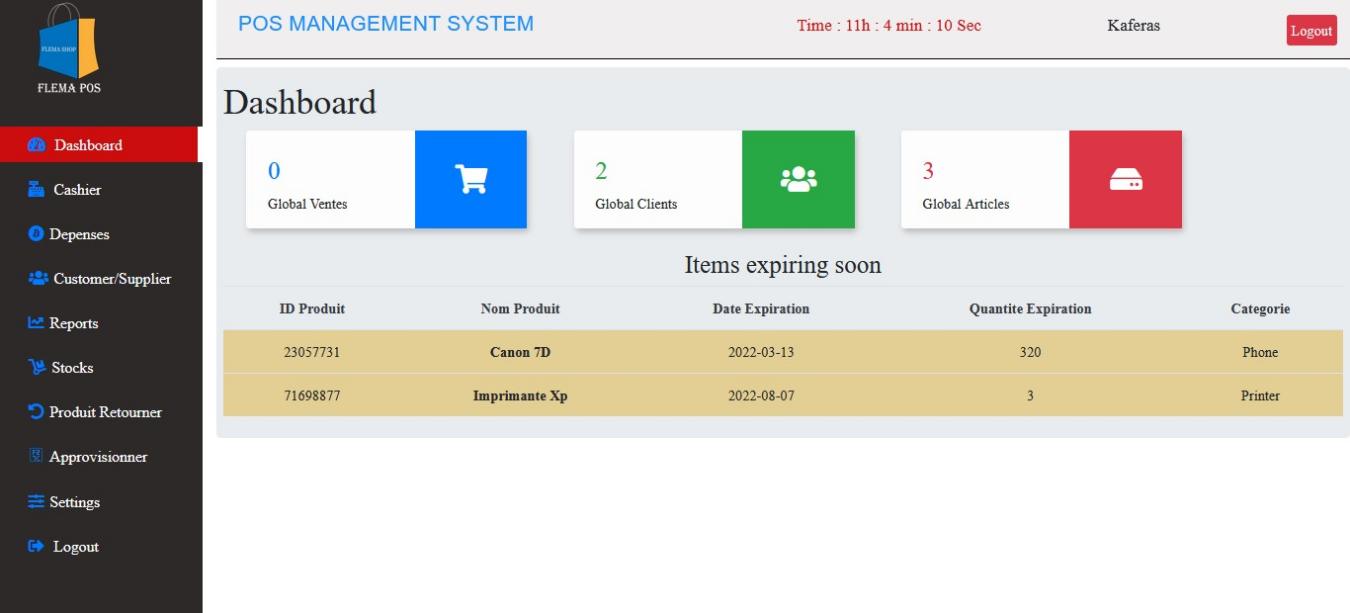
****

Figure 14 : Page d’acceuil de l’application

#### La page du Caissier

Afin de pouvoir faire une commande a un client et la payer il faut qu’il y est une interface qui facilitera la tâche ainsi calculer le nombre de produit dont il veut, le prix total des articles commandes, enlever et ajouter une article, faire une recherche d’une article désirée par code barre ainsi que le moyen de paiement utiliser pour payer la facture

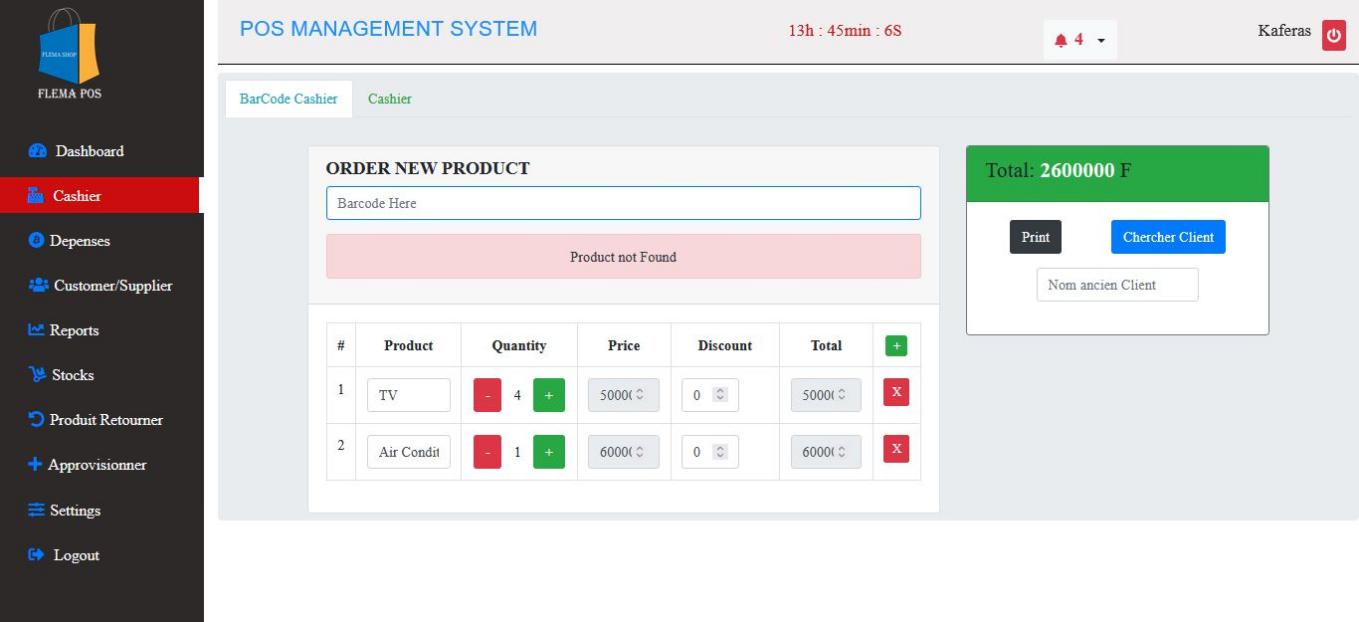


Figure 15 : Page de Vente du Caissier

#### Gestion Client et des Fournisseurs

Le menu clients/fournisseur permet de bien gérer les clients ainsi que les fournisseurs du magasins afin de bien suivre les achats et les marchandises ainsi que les fournisseurs.

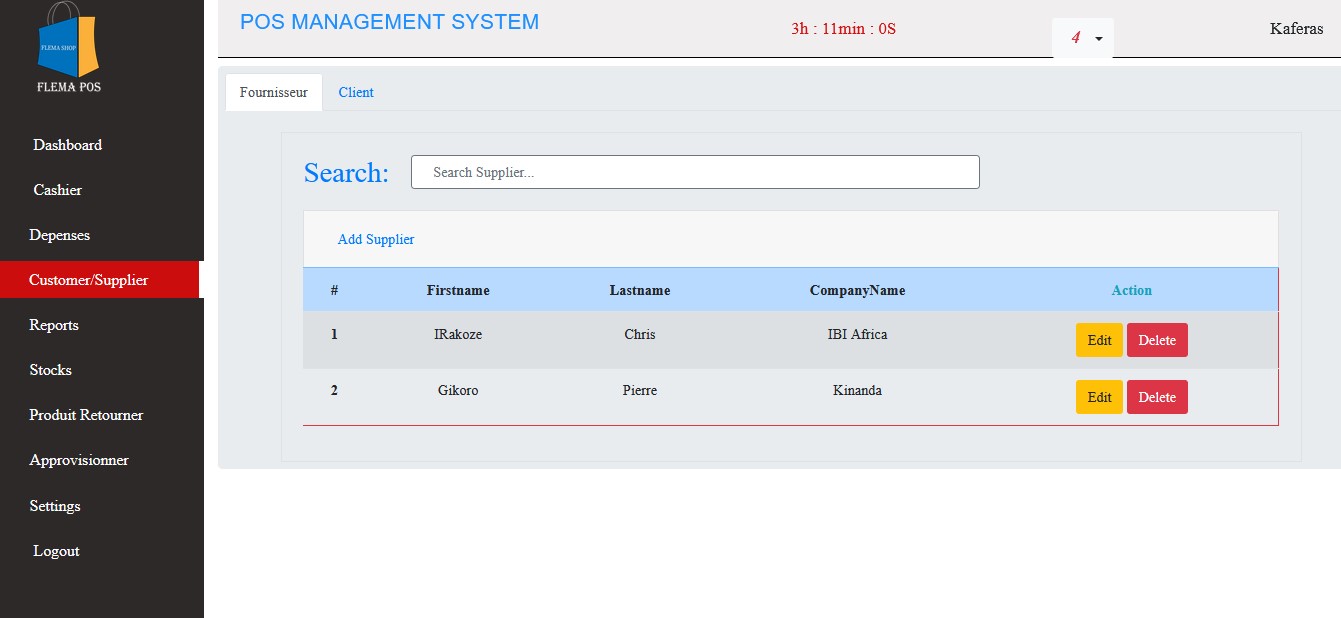


Figure 16 : Page de gestion Fournisseurs

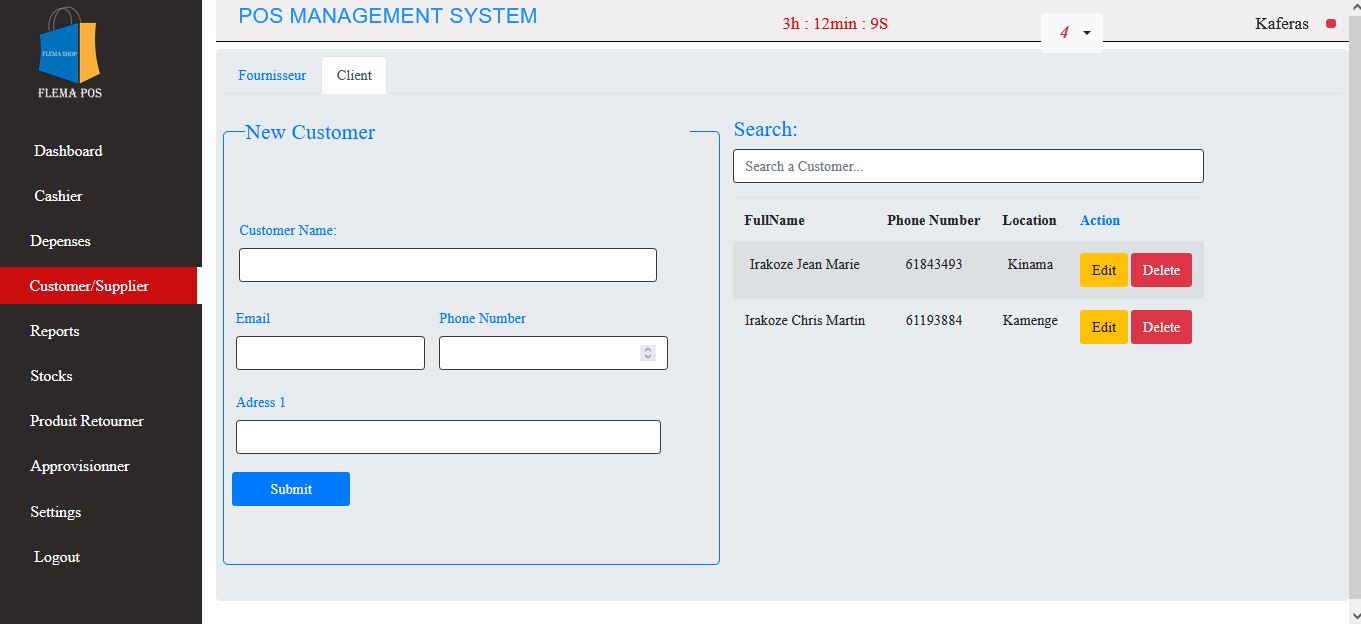


Figure 17 : Page de gestions des Clients

#### Page de création d’un nouvel utilisateur

Cette page permet de créer un nouvel utilisateur, lui donner un nom d’utilisateur et un mot de passe lui permettant de s’authentifier.

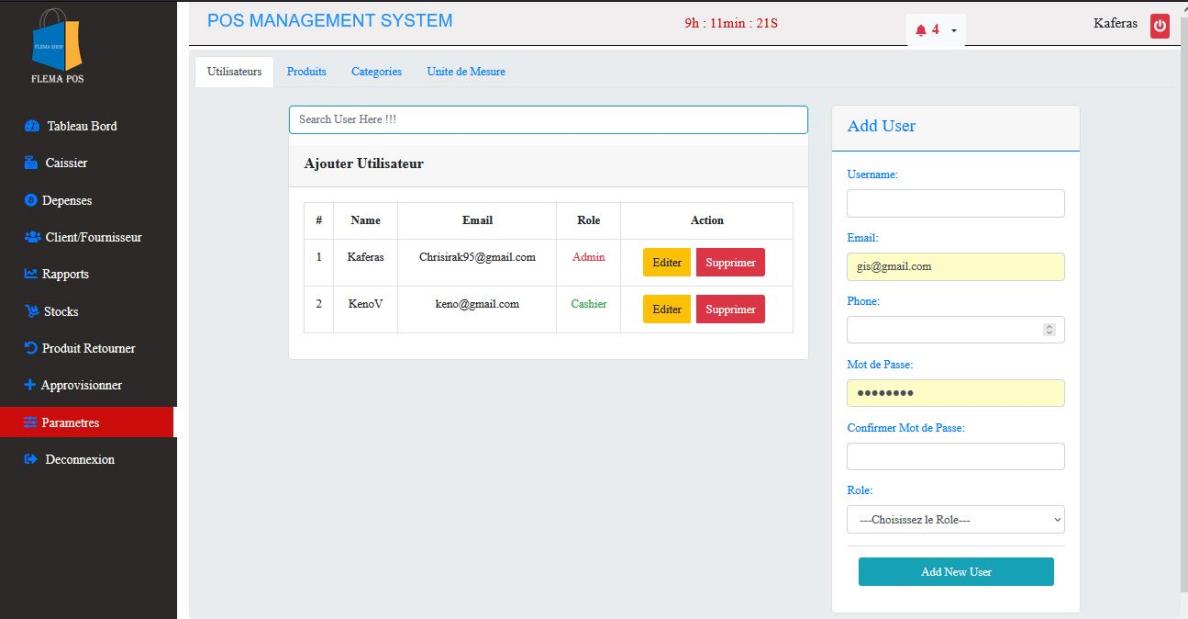


Figure 18 : Page Gestion des Utilisateurs

#### e) Page de Création de l’article du magasin

Afin de pouvoir faire des ventes de différents articles du magasin il faut qu’ils soient enregistrer quelque part dans le système dans ce cas cette page représente l’endroit de l’enregistrement de l’article et son son Code barre

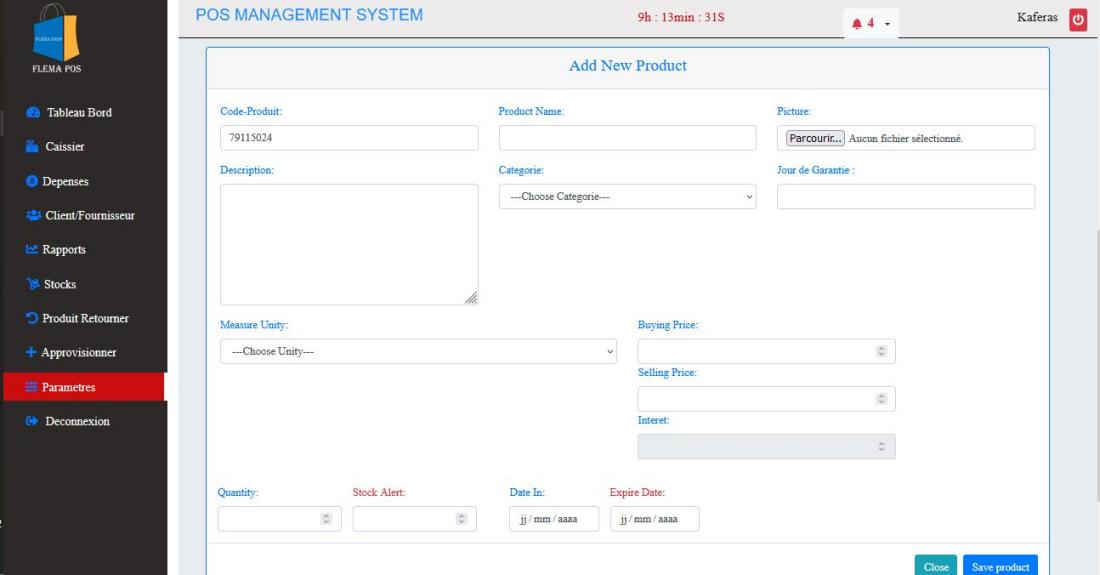


Figure 19 : Page Ajout des Produits

#### Page Création de Catégorie

La création de l’article se fait facilement en les classant selon leur Catégories afin de les ranger bien .. la page suivante montre la création du catégorie

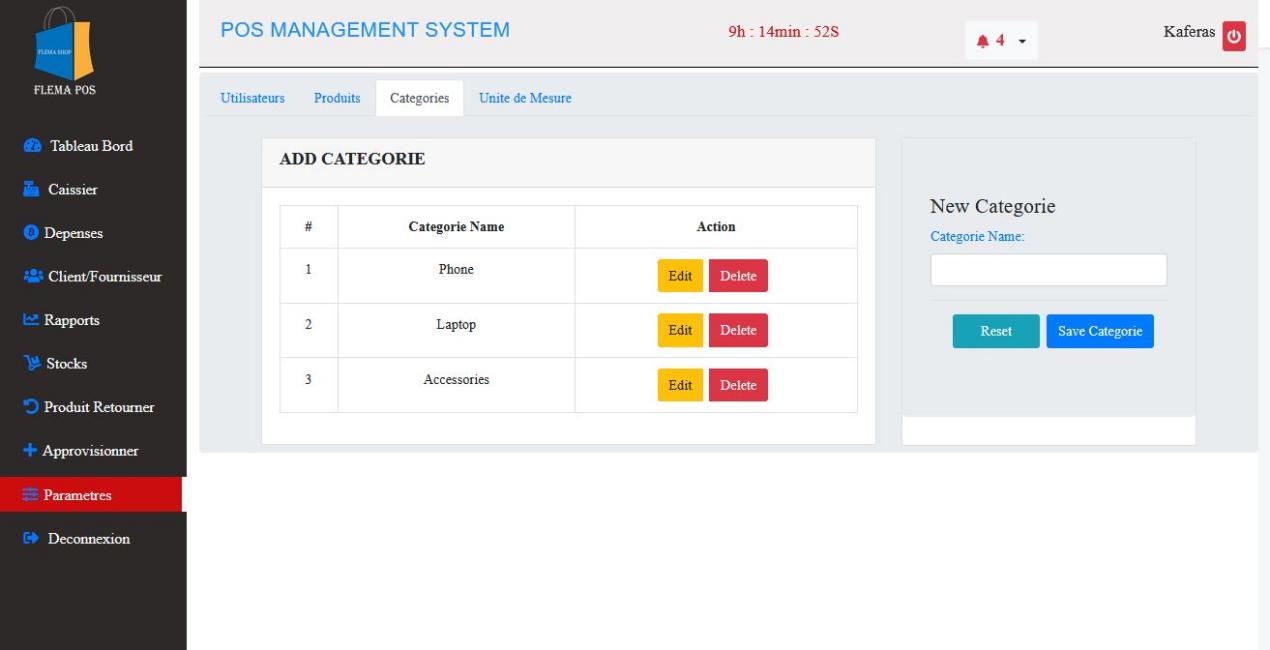


Figure 20 : Page Ajout des Catégories

#### g) Page de Création de l’ unité de Mesure

Les articles sont vendus ou calcules selon une unité de mesure prédéfini dans ce cas il faut le prévoir au préalable afin que lors de l’enregistrement de l’article l’ unité y soit déjà

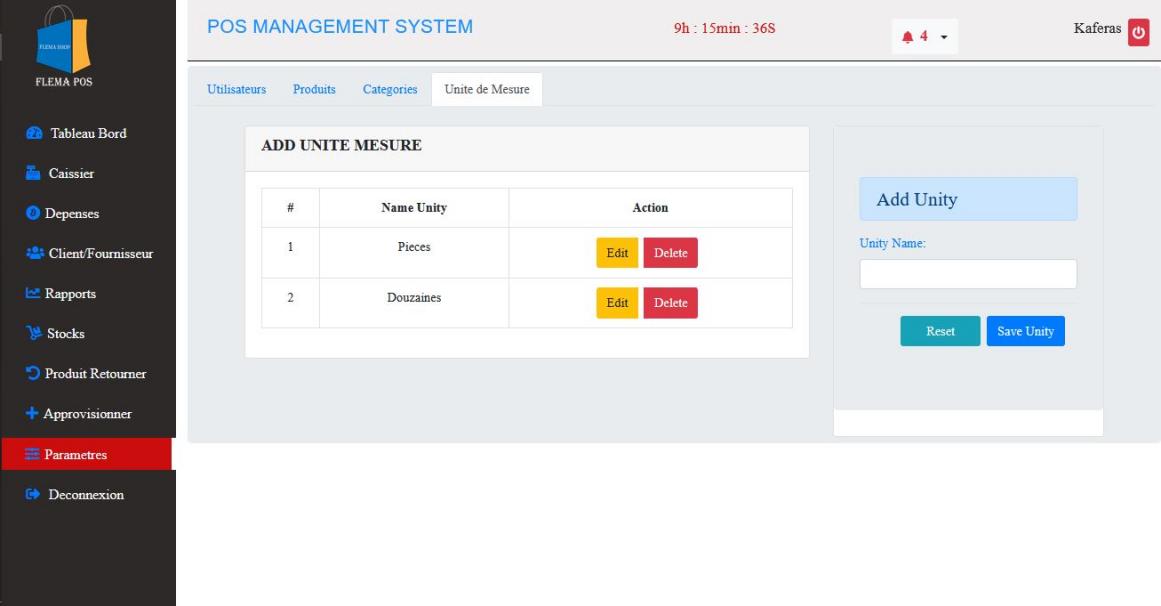


Figure 21 : Page Création des Unités de Mesures

#### h) **Page approvisionnement de l’article**

Le stock écoule pendant une période de vente dans notre il faut que lors de l’achat ou approvisionnement de du nouveau article il y est une page qui nous servira de la fonctionnalité d’ou la page suivante:

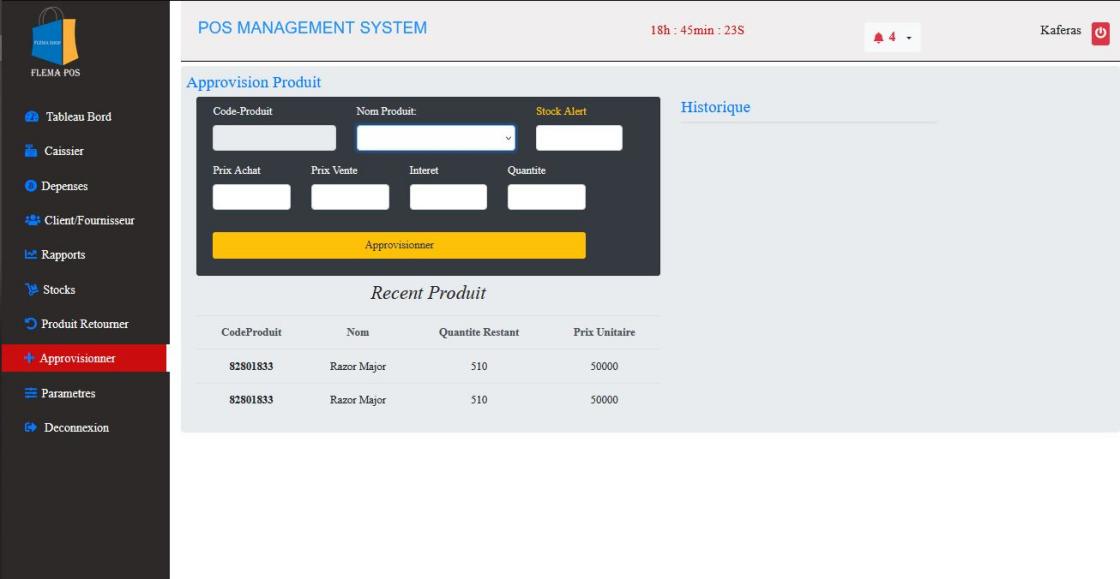


Figure 22 : Page approvisionnement de l’article

# CONCLUSION GENERALE ET RECOMMANDATIONS

# Conclusion générale

Au cours de ce travail, notre préoccupation principale a été de trouver une solution informatique permettant de mieux aider le commerce du magasin Flema Shop mais en général tous les marchands

L’objectif était de concevoir et de réaliser une application web de gestion de vente de matériels électronique  : cas du super marche Flema Shop. Il est grand temps de générer une conclusion et de proposer quelques recommandations.

Pour bien mener ce travail, nous l’avons subdivisé en quatre chapitres :

1. Dans l’introduction, nous avons parlé du contexte général, de l’intérêt du sujet, de sa problématique, de l’objectif global et des objectifs spécifiques, de la méthode et techniques utilisées, de sa délimitation.
2. Dans le second chapitre nous avons fait la présentation générale de l’entreprise, l’analyse et critique de l’existant, les solutions proposées, l’historique .
3. Dans le troisième chapitre nous avons développé la conception et la modélisation de l’application par UML à travers ses différents diagrammes.
4. Le dernier chapitre comporte à son tour, la présentation du résultat final de notre travail de fin de cycle de baccalauréat qui est une application de gestion de vente de matériels électronique  : cas du super marche Flema Shop.

# Recommandations

Avant de clore notre travail, on aimerait bien laisser quelques recommandations à l’endroit des responsables du Magasin Flema Shop :

* A l’endroit des responsables du Magasin , nous leur recommandons de tout faire pour exploiter ce travail afin de rendre le Commerce du Magasin plus sécurisés, plus fiable et accessible en ligne pour un usage facile partout ou l’on est.
* Aux futurs lauréats : Nous leur recommandons de continuer à travailler sur ce projet afin améliorer le travail notamment comme le fait de gérer plusieurs stocks, envoyer les factures a l’OBR, plus de rapport des stock , le Control du stock ainsi que les modules nous n’avons pas touchés et ajouter sur l’une ou l’autre fonctionnalité que notre application n’a pas pu couvrir.

# BIBLIOGRAPHIE

1. Désiré NIYONGABO et Fiston BAYUBAHE, « CONCEPTION ET REALISATION D’UNE APPLICATION WEB DE GESTION DE VENTE DES MATÉRIELS DE CONSTRUCTION : CAS DE L’UMAMEBU », ULT, Novembre 2020
2. Jean Népomuscѐne NTUNZWENIMANA, « CONCEPTION ET REALISATION D’UNE APPLICATION WEB DE GESTION DE VENTE DES PRODUITS D’UNE ENTREPRISE DE PRODUCTION : CAS DE L’ENTREPRISE LIQUIDS, UNIVERSITE DU LAC TANGANYIKA »
3. Yves RUSUKU et Élie BAHEMUKA, « APPLICATION D’AIDE A LA GESTION DE LA PRODUCTION, VENTES, ET STOCK DES PRODUITS BOULANGERS, UNIVERSITE DU LAC TANGANYIKA. »
4. MOUSSZ Ally et NGABO Gaël : « conception et réalisation d’une application de gestion d’attribution des places dans un marché » ( cas de la société burundaise des services s.a) ULT Décembre 2016