**REPUBLIQUE DU SENEGAL**

**UN PEUPLE - UN BUT - UNE FOI**

**MINISTERE DE LA FORMATION PROFESSIONNELLE DE L’APPRENTISSAGE ET DE L’INSERTION**



**CENTRE DE FORMATION PROFESSIONNELLE ET TECHNIQUE**

**SÉNÉGAL / JAPON (CFPT- S/J)**

ROUTE DE ‘AEROPORT SUD FIDAK VDN

BP 8411 Dakar – Yoff

Tél : +221 33 827 6 070 – 33 869 15 42

**SECTION : INFORMATIQUE INDUSTRIELLE**

**(IIR)**

**Spécialité : INFORMATIQUE INDUSTRIELLE ET RÉSEAUX**

**MÉMOIRE DE FIN D'ÉTUDE**

**THÉME :**

**MISE EN PLACE D’UNE APPLICATION POUR LA GESTION D’UN MAGASIN DE PIÉCES DETACHÉES**

**En vue de l’obtention du Brevet de Technicien Supérieur (BTS)**

**Présenté et soutenu par :                                                                                                               Encadré par :**

**Mame Faty KANE Mme Khadija Bâ**

**Mouhamadou Moustapha MBENGUE**

**DEDICACES**

Ce mémoire est dédié à :

Nos mères, qui ont œuvré pour notre réussite, de par leur amour, leur soutien, leurs précieux conseils et leurs sacrifices consentis, reçoivent à travers ce travail, l’expression de nos sentiments et éternelle gratitude.

Nos pères, qui peuvent être fiers et trouver ici le résultat de longues années de sacrifices et de privations.

Nos frères, sœurs et amis qui nous ont aidés.

**REMERCIEMENTS**

Nous rendons grâce à ALLAH le tout puissant d’avoir exaucé nos vœux et prières.

Comment une personne peut-elle dire merci à tout le monde quand il y a autant de gens à remercier ? Évidemment, par ce mémoire, nous voulons dire merci à tous ceux qui ont contribué de près ou de loin à notre réussite.

Nous remercions très sincèrement nos parents qui représentent de puissants modèles à imiter dans nos vies, qui nous ont enseigné amour et bienveillance mais surtout qui nous ont soutenus depuis le début. Merci à vous chers parents.

Nous tenons à saisir cette occasion et adresser nos profonds remerciements à notre encadreur Mme Khadija BA.

Nous remercions tous les enseignants qui nous ont suivis durant notre formation, notamment M. Saliou DIALLO chef de section informatique du CFPT-S/J pour son encadrement. Nous remercions M. GNING, M. TENDENG, M. BA, Mme DIALLO pour leurs accompagnements et leurs valeureux conseils.

Nous remercions toute personne qui a participé de près ou de loin à l’exécution de ce travail.

**Merci à tous**

**TABLE DES MATIÈRES**

[INTRODUCTION GENERALE 6](#_Toc79865718)

[PROBLÉMATIQUE 7](#_Toc79865719)

[PREMIÈRE PARTIE : 8](#_Toc79865720)

[PRÉSENTATION GÉNÉRALE DU PROJET 8](#_Toc79865721)

[I.1. INTRODUCTION 9](#_Toc79865722)

[I.2. CHOIX DU THEME 9](#_Toc79865723)

[I.3. INTERET DE L’ETUDE 9](#_Toc79865724)

[I.4. PROBLEMES 9](#_Toc79865725)

[I.5. OBJECTIFS 10](#_Toc79865726)

[I.6. CAHIER DE CHARGE 10](#_Toc79865727)

[I.6.1. Activités 10](#_Toc79865728)

[I.6.2. Fonctionnement 10](#_Toc79865729)

[II.7. Conclusion 11](#_Toc79865730)

[DEUXIEME PARTIE : 12](#_Toc79865731)

[LE CHOIX DES OUTILS ET DE LA MODELISATION DE LA BASE DE DONNÉES 12](#_Toc79865732)

[II.1. INTRODUCTION 13](#_Toc79865733)

[II.2. SYSTEME D’INFORMATION ET BASE DE DONNÉES 13](#_Toc79865734)

[II.2.1. Système d’information 13](#_Toc79865735)

[II.2.2. Les bases de données 14](#_Toc79865736)

[II.3. PRÉSENTATION DE LA MÉTHODE MERISE 17](#_Toc79865737)

[II.3.1. Approche par niveau 17](#_Toc79865738)

[II.3.2. Approche par étapes 18](#_Toc79865739)

[II.3.3. Caractéristique de MERISE 19](#_Toc79865740)

[II.4. L’ÉTUDE CONCEPTUELLE 19](#_Toc79865741)

[II.4.1. Le modèle conceptuel des données (MCD) 19](#_Toc79865742)

[II.4.2. Le modèle Logique de données (MLD) 22](#_Toc79865743)

[II.4.3. Le modèle Physique de données (MPD) 23](#_Toc79865744)

[II.5. CONCLUSION 24](#_Toc79865745)

[TROISIEME PARTIE : 25](#_Toc79865746)

[LE CHOIX DU LANGAGE ET LA REALISATION 25](#_Toc79865747)

[III.1. INTRODUCTION 26](#_Toc79865748)

[III.2. OUTILS DE DÉVELOPPEMENT 26](#_Toc79865749)

[III.2.1. WampServer 26](#_Toc79865750)

[III.2.2. PowerAMC 29](#_Toc79865751)

[III.2.3. NetBeans 30](#_Toc79865752)

[III.3. LANGAGES UTILISÉES 32](#_Toc79865753)

[III.3.1. Langage de requête SQL 32](#_Toc79865754)

[III.3.2. Langage de programmation Java 32](#_Toc79865755)

[III.4. ORGANIGRAMME D’INTERACTION ET LES DIFFÉRENTES INTERFACES 33](#_Toc79865756)

[III.4.1. Organigramme d’interaction 33](#_Toc79865757)

[III.4.2. Les différentes interfaces 35](#_Toc79865758)

[III.4.2.1. PAGE D’AUTHENTIFICATION OU LOGIN 35](#_Toc79865759)

[III.4.2.2. MENU PRINCIPALE 38](#_Toc79865760)

[III.4.2.3. GESTION DES UTILISATEURS 39](#_Toc79865761)

[III.4.2.4. GESTION DES CLIENTS 40](#_Toc79865762)

[III.4.2.5. GESTION DES FOURNISSEURS 41](#_Toc79865763)

[III.4.2.6. GESTION DES PRODUITS 42](#_Toc79865764)

[III.4.2.7. CACHIER / FACTURE 43](#_Toc79865765)

[III.4.2.8. RECU 47](#_Toc79865766)

[III.5. CONCLUSION 48](#_Toc79865767)

[CONCLUSION GENERALE 49](#_Toc79865768)

[WEBOGRAPHIE 50](#_Toc79865769)

[BIBLIOGRAPHIE 51](#_Toc79865770)

[TABLE DES FIGURES 52](#_Toc79865771)

[TABLE DES TABLEAUX 52](#_Toc79865772)

# INTRODUCTION GENERALE

Avant la naissance de l’ordinateur, on enregistrait toutes les informations manuellement sur des supports en papier, ce qui engendrait beaucoup de problèmes tels que la perte de temps considérable dans la recherche d’informations ou la perte de ces dernières…De ce fait, l’ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l’information. Cette invention a permis d’informatiser les systèmes de données des entreprises, ce qui est la partie essentielle dans leur développement d’aujourd’hui. Cette machine nous facilite aujourd’hui la mémorisation de nos informations et même la sécurité de ces dernières.

C’est surtout la mise en communication des ordinateurs qui a permis de révolutionner les méthodes de travail. Ainsi, on a assisté à l’émergence des réseaux. Ce nouveau progrès offre aux utilisateurs de nouveaux outils de travail et leur permet d’améliorer leur rentabilité et leur productivité.

Cependant, pour une meilleure gestion, il ne suffit pas seulement d’avoir un bon matériel et un taux d’information élevé, mais il s’agit surtout de réaliser un système d’information performant qui gère et qui répond à tous les besoins de gestion.

C’est dans ce cadre d’idées que s’inscrit notre projet de fin d’étude qui consiste à la création d’une application dynamique: la mise en place d’une application pour la gestion d’un magasin de pièces détachées.

A cet effet, nous avons réparti notre mémoire en trois parties :

* Nous commencerons par une étude préalable dans la première partie qui englobe une présentation générale du projet.
* Ensuite nous poursuivrons avec la deuxième partie qui comporte le choix des outils et de la modélisation de la base de données.
* Enfin, nous terminerons par la troisième partie qui illustre le choix du langage ainsi que l’environnement utilisé pour la réalisation de l’application en présentant un aperçu de l’application et quelques fenêtres et organigramme de fonctionnement.

# 

# PROBLÉMATIQUE

Le système de gestion d’achat, de vente et de stock de pièces détachées rencontre certaines difficultés :

- La perte de documents ;

- L’oubli du stock en cours ;

- La redondance dans le saisi des données ;

- Le manque d'informations pouvant contribuer à l'élaboration des rapports au moment opportun ;

- Temps de réponse très long;

- Difficultés dans l'inventaire périodique.

En somme, nous pouvons dire que le problème le plus important est le manque d’automatisation des données des magasins dont seules les procédures manuelles sont utilisées. Les enregistrements des entrées et sorties de stocks sont faits sans recours de logiciel ou d’application de gestion de stock.

Par conséquent, comment palier à tout ceci pour une meilleure gestion de son stock ?

# PREMIÈRE PARTIE :

# PRÉSENTATION GÉNÉRALE DU PROJET

## 

## I.1. INTRODUCTION

Nous présenterons dans cette partie le choix du thème suivi de l’intérêt de l’étude. Ensuite nous poserons les problèmes rencontrés du responsable du magasin, et enfin nous parlerons des objectifs.

## I.2. CHOIX DU THEME

Notre sujet d’étude s’intitule : Mise en place d’une application pour la gestion de stock d’un magasin de pièces détachées.

Ce thème a été choisi dans le but d’essayer de résoudre les problèmes et les difficultés que rencontre quotidiennement le magasinier.

## I.3. INTERET DE L’ETUDE

Pour l’étudiant:

* Mise en pratique des acquis de notre formation ;
* De faire une étude complète d’un problème réel c’est-à-dire de l’étude préalable jusqu’à la mise en exploitation ;
* D’accroitre ses connaissances et son expérience professionnelle,

Pour le magasin :

* De passer d’un processus lent avec beaucoup de risques d’erreurs manuelles vers un processus automatisé rapide et fiable ;
* Bonne gestion du magasin en temps réel ;
* Sécurité et suivie des données à distance ;

## I.4. PROBLEMES

Vu l’importance des tâches qui s’effectuent quotidiennement au niveau du magasin, la gestion de stock manuelle n’est pas fiable. Aux cours des interviews que nous avons réalisés avec le responsable, nous sommes arrivés à confirmer l’existence de ces défaillances qu’on résume comme suit :

* Manque d’informations sur la disponibilité des produits en stock.
* Difficulté d’établir un état des stocks réel dû à la mauvaise gestion des entrées.
* Masse d’informations importantes traitées manuellement.
* Risque de perte de document durant l’archivage.
* La redondance dans le saisi des données.

## I.5. OBJECTIFS

En effet, le magasin Keur Sylla Transport a besoin d'un système plus performant que celui utilisé actuellement.

L'objectif principal de ce travail de recherche est d'informatiser la gestion du magasin Keur Sylla Transport grâce à une application.

C’est la raison qui nous a poussé à choisir le sujet **<<création d’application gestion d’un magasin de pièces détachées>>** pour concevoir un logiciel susceptible de gérer l’achat, la vente et le stock des pièces détachées et aussi de connaitre les fournisseurs et les clients.

## I.6. CAHIER DE CHARGE

Le cahier de charge permet de définir les besoins de l’entreprise et explique en même temps la solution proposée pour illustrer la gestion commerciale. C’est pourquoi il doit être précis.

Une fois les besoins définis dans le cahier de charge, on passe à la phase pratique de réalisation.

### I.6.1. Activités

1. Garage : vente des pièces.
2. Administration : gérer les clients, les fournisseurs, les utilisateurs, la facture et les ventes

3. Dépôt : stockage et ventes des pièces.

### I.6.2. Fonctionnement

L’application doit permettre :

* d’identifier un utilisateur par login et mot de passe enregistrés au préalable par l’administrateur de l’application dans la base de données. Dans ce cas seul l’administrateur ou le directeur a le droit de gérer les comptes des utilisateurs (création, modification, suppression).
* De gérer les factures et les reçus via le cachier qui ne gère que ces derniers.
* De consulter les informations concernant les clients, les produits et les fournisseurs.
* D’ajouter les nouvelles données que ce soit clients, produits ou fournisseurs.
* De modifier les propriétés concernant ces données.
* De supprimer les données.
* De gérer les ventes des produits.
* De gérer les achats auprès des fournisseurs.
* De gérer le stock.
* Imprimer ces reçus (l’état du stock, bon livraison, facture).

## II.7. Conclusion

Dans ce chapitre, nous avons parlé du choix du thème et de l’intérêt de l’étude, des problèmes, des objectifs recherchés ainsi que du cahier de charge que nous utiliserons comme référence pour la mise en place de notre application de gestion de stock ou ses démarches seront détaillées dans le chapitre suivant.

# DEUXIEME PARTIE :

# LE CHOIX DES OUTILS ET DE LA MODELISATION DE LA BASE DE DONNÉES

## II.1. INTRODUCTION

Comme pour tout travail, il est primordial avant de passer à la phase de réalisation, de bien étudier le système en vue de recenser tous les éléments qui pourraient avoir une incidence sur les traitements des données par la suite afin d’optimiser au maximum la conception de la solution à adopter, et c’est justement ce point-là qui fera le sujet de ce chapitre.

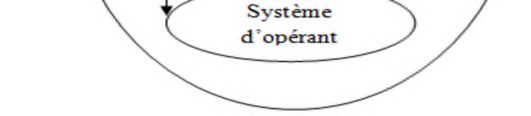
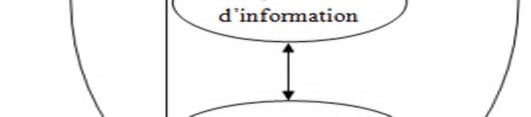
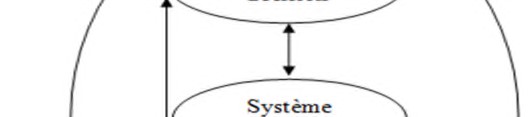
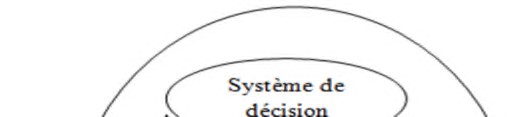
Dans ce chapitre nous proposons d'apporter des précisions quant aux termes et concepts qui seront utilisés dans ce présent travail en les inscrivant dans le contexte de leur utilisation tant au plan général que celui particulier. Pour cela nous allons parler de généralités sur le système d’information et les bases de données, comme nous allons aborder une vue globale sur la méthode Merise.

## II.2. SYSTEME D’INFORMATION ET BASE DE DONNÉES

### II.2.1. Système d’information

Le système d’information (S.I) d’une organisation est l’ensemble des éléments chargés de stocker et de traiter les informations (ordinateurs, postes de travail, règles et méthodes).

L’entreprise peut être décrite comme un système composé de trois sous-systèmes : le système de décision, le système opérant et le système d’information dont sa fonction est de produire et de mémoriser les informations et de les mettre à la disposition du système de pilotage et du système opérant.



**Figure 1 :** **Vision globale d’une entreprise**

### II.2.2. Les bases de données

Il est difficile de donner une définition exacte de la notion de base de données. Si bien que l’on distingue les quelques définitions suivantes :

Une base de données peut être vue comme une collection de données persistantes, opérationnelles, enregistrées en mémoire secondaire (disque dure par exemple). Ces données doivent être cohérentes, non redondantes (ou de redondance minimale), accessibles simultanément par plusieurs programmes (ou utilisateurs). Il faut noter qu’une base de données est indépendante des programmes d’application qui l’utilisent.

En d’autres termes, une base de données est un ensemble de données organisé en vue de son utilisation par des programmes correspondants à des applications distinctes et de manière à faciliter l’évolution indépendante des données et des programmes.

#### II.2.2.1. L’utilité d’une base de données

L’utilité d’une base de données est de regrouper les données communes à une application dont le but est :

* D’éviter les redondances et les incohérences de données.
* D’offrir des langages de haut niveau pour la définition et la manipulation des données.
* De partager les données entre plusieurs utilisateurs.
* De contrôler l’intégrité, la sécurité et la confidentialité des données.
* D’assurer l’indépendance entre les données et les traitements.

#### II.2.2.2. Les modèles de bases de données

* **Le modèle hiérarchique (années 60)**

Dans le premier modèle de SGBD, les données sont classées hiérarchiquement. Ce modèle utilise des pointeurs entre les différents enregistrements, organisés dans une structure arborescente de façon à ce que chaque enregistrement n’ait qu’un seul possesseur.

* **Le modèle réseau (années 70)**

Il lève de nombreuses limites du modèle hiérarchique grâce à la possibilité d’établir des liaisons de type n-n, les liens entre objets pouvant exister sans restriction. Pour retrouver une donnée dans ce modèle, il faut connaitre le chemin d’accès (les liens), ce qui rend les programmes dépendants de la structure de données.

* **Le modèle relationnel**

Les données sont représentées dans des tables, sous forme de n-uplets. C’est le modèle le plus utilisé. Il a donné lieu au langage SQL (Structured Query Language), extension de l’algèbre relationnel, standardisation en 1987. Près de 80% des utilisations en entreprise.

* **Le modèle orienté objet**

Il permet de voir une base de données comme un ensemble de classe d’objets, ayant des liens d’héritage, d’agrégation, de composition, ou de simples associations entre elles.

#### II.2.2.3. Les niveaux d’une architecture de bases de données

Dans une architecture de BDD (base de données), on distingue généralement trois niveaux qui sont :

* **Le niveau externe :** c’est le niveau où les utilisateurs voient les données, il est appelé aussi niveau utilisateur.
* **Le niveau physique :** ce niveau est relatif à la mémoire physique (disque dur par exemple), il s’agit du niveau ou les données sont stockées, appelé aussi niveau interne.
* **Le niveau conceptuel :** c’est le niveau intermédiaire entre les deux précédents, appelé aussi niveau logique.

Il existe plusieurs niveaux externes, chacun spécifique à un utilisateur particulier. Par contre, il existe un seul niveau conceptuel qui donne la représentation abstraite de la totalité de la base de données et un seul niveau interne représentant la totalité de la base de données telle qu’elle est enregistrée en mémoire.



**Figure 2 : Les trois niveaux d’une architecture de base de données**

#### II.2.2.4. Notion d’un système de gestion de base de données (SGBD)

Un système de gestion de base de données est un ensemble de programmes qui permet la gestion et l’accès à une base de données. Un SGBD héberge généralement plusieurs bases de données, qui sont destinées à des logiciels ou des thématiques différentes. Nous distinguons couramment les SGBD classiques, dits SGBD-R ou SGBD relationnels des SGBD-O ou SGBD orientés objet.

#### II.2.2.5. Les objectifs d’un système de gestion de base de données (SGBD)

Les bases de données et les systèmes de gestion de base de données ont été créés pour répondre à un certain nombre de besoins et pour résoudre un certain nombre de problèmes. Ses objectifs sont les suivants :

* Efficacité des accès aux données.
* Administration centralisée des données.
* Non redondance des données.
* Cohérence des données.
* Sécurité des données.

## II.3. PRÉSENTATION DE LA MÉTHODE MERISE

Merise est née vers 1978-1979, à la suite d’une vaste consultation lancée en 1977 par le ministère de l’industrie pour choisir plusieurs sociétés de service et de conseil en informatique et le centre d’étude technique, pour mettre au point une méthode de troisième génération de conception et de réalisation de système d’information.

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d’arriver à concevoir un système d’information. La méthode Merise est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. En effet, l’agencement des données n’a pas à être souvent remanié, tandis que les traitements sont plus fréquemment.

Pour cela, MERISE utilise deux approches qui sont :

* L’approche par niveau
* L’approche par étape

### II.3.1. Approche par niveau

#### II.3.1.1. Niveau conceptuel

A ce niveau le système d’information (SI) est représenté indépendamment de son organisation et les moyens physiques et informatiques. Les règles mises en évidence à ce niveau sont des règles de gestion traduisant les objectifs et les contraintes qui présentent l’entreprise.

Deux modèles sont proposés à ce niveau :

* Le modèle conceptuel des données (MCD).
* Le modèle conceptuel des traitements (MCT).

#### II.3.1.2. Niveau logique

A ce niveau sont faits tous les choix techniques afin de déterminer qui fera quoi, où, quand et comment. Les traitements sont réalisés dans le système d’information en tenant compte des contraintes imposées par ces choix.

Les deux modèles proposés à ce niveau sont :

* Le modèle organisationnel des traitements (MOT).
* Le modèle logique des données (MLD).

#### II.3.1.3. Niveau physique

A ce niveau sont faits les choix techniques permettant de déterminer les moyens informatiques. Les règles misent en évidence sont des règles techniques.

Les deux modèles proposés à ce niveau sont :

* Le modèle physique des données (MPD).
* Le modèle opérationnel des traitements (MOPT).

### II.3.2. Approche par étapes

Merise propose aussi six étapes qui sont :

* **Schéma directeur :** l’objectif de cette étape est de faire le point entre la stratégie de l’entreprise et ses besoins en termes de système d’information.
* **Etude préalable :** cette étape a pour but, de prendre les domaines d’activités domaine par domaine ; et établir d’une manière plus approfondie les projets à mettre en œuvre et leurs interfaces.
* **Etude détaillée :** elle a pour but, de décrire d’une manière détaillée les champs d’études ; et de concevoir un nouveau système en intégrant les nouvelles orientations qui répondent du problématique posé, lors de l’étude préalable.
* **Réalisation :** elle comprend l’écriture des programmes, l’implémentation des bases de données où des tests sont faits pour garantir une meilleure intégration de l’ensemble des transactions.
* **Mise en œuvre :** cette étape constitue la mise à la disposition des utilisateurs, le nouveau système en assurant que toutes les consignes et recommandations seront prises en considérations.
* **Maintenance :** elle consiste à assurer la maintenance de l’application mise en place afin de la faire durer plus longtemps.

### II.3.3. Caractéristique de MERISE

* Merise est une méthode globale qui assure la cohérence entre les différents composants du système et les objectifs de l’entreprise.
* Une distinction nette entre les données et les traitements afin de dégager les aspects statiques (données) et les concepts dynamiques (traitements) du système d’information, car ce n’est qu’après la réalisation, que ces aspects seront approchés.
* Une démarche par niveaux dont l’objectif est la formalisation du futur système.
* Une démarche par étapes, chaque résultat d’une étape achevée sera le point de départ de la suivante

## II.4. L’ÉTUDE CONCEPTUELLE

L’étude conceptuelle consiste à approfondir la solution d’ensemble établit dans l’étude préalable ; elle consiste à concevoir le système par trois niveaux d’abstraction (conceptuel, organisationnel et opérationnel).

|  |  |  |
| --- | --- | --- |
| **Niveau** | **Statique** | **Dynamique** |
| **Niveau conceptuel** | Modèle conceptuel des données (MCD) | Modèle conceptuel des traitements (MCT) |
| **Niveau organisationnel** | Modèle logique des données (MLD) | Modèle organisationnel des traitements (MOT) |
| **Niveau opérationnel** | Modèle physique des données (MPD) | Modèle opérationnel des traitements (MOPT) |

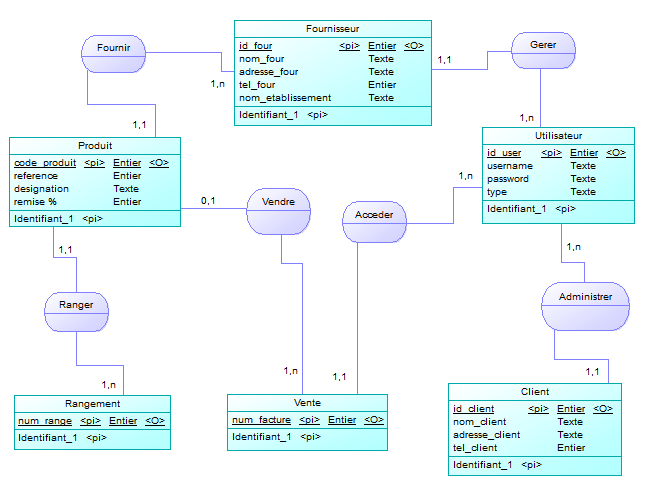
**Tableau 1 : Les niveaux d’abstraction de Merise**

### II.4.1. Le modèle conceptuel des données (MCD)

Chaque méthode de conception s’appuie sur un modèle conceptuel spécifique, pour la méthode MERISE c’est le modèle Entité / Association qui est utilisé, il permet de décrire et de modéliser graphiquement les informations et les objets (appelés entités) manipulés par le système d’information ainsi que les liens entre les entités. En d’autres termes, le modèle conceptuel des données est une représentation statique (donnée) d’un système d’information.

#### II.4.1.1. Le schéma du MCD

Après avoir défini toutes les entités et les associations mises en jeu avec leurs constituants et en appliquant la méthode de MERISE déjà détaillée lors de la partie précédente, on aboutit au Modèle Conceptuel de Données représenté sur le schéma qui suit :



**Figure 3 : Modèle conceptuel de données (MCD)**

#### II.4.1.2. Dictionnaire des données

Le dictionnaire de données présente la définition, le type (texte, numérique…) et le format (nombres de caractères, de décimales…) de l’ensemble des données gérées dans la base de données.

Le dictionnaire des données est un tableau qui regroupe toutes les données du système d’information.

Pour chaque donnée il faut préciser :

• Sa désignation.

• Sa longueur (en caractère).

• Un code (attribué).

• Son type (numérique, alphabétique, alphanumérique).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entité | Code | Désignation | Type | Taille |
| Utilisateur | id\_user | Identifiant de l’utilisateur | INT |  |
| username | Nom de l’utilisateur | VARCHAR | 50 |
| password | Mot de passe de l’utilisateur | VARCHAR | 50 |
| type | Type d’utilisateur | VARCHAR | 50 |
| Produit | code\_produit | Code du produit | INT |  |
| reference | Référence du produit | INT |  |
| designation | Nom du produit | VARCHAR | 50 |
| rangement | Place du produit | VARCHAR | 50 |
| remise % | Remise | INT |  |
| prix | Prix du produit | INT |  |
| stock | Stock du produit | INT |  |
| Client | id\_client | Identifiant du client | INT |  |
| nom\_client | Nom du client | VARCHAR | 100 |
| adresse\_client | Adresse du client | VARCHAR | 100 |
| tel\_client | Téléphone du client | INT |  |
|  | Id\_four | Identifiant du fournisseur | INT |  |
| Fournisseur | nom\_four | Nom du fournisseur | VARCHAR | 100 |
| adresse\_four | Adresse d fournisseur | VARCHAR | 100 |
| tel\_four | Téléphone du fournisseur | INT |  |
| nom\_etablissement | Nom de son établissement | VARCHAR | 100 |
| Vente | num\_facture | Numéro de la facture | INT |  |

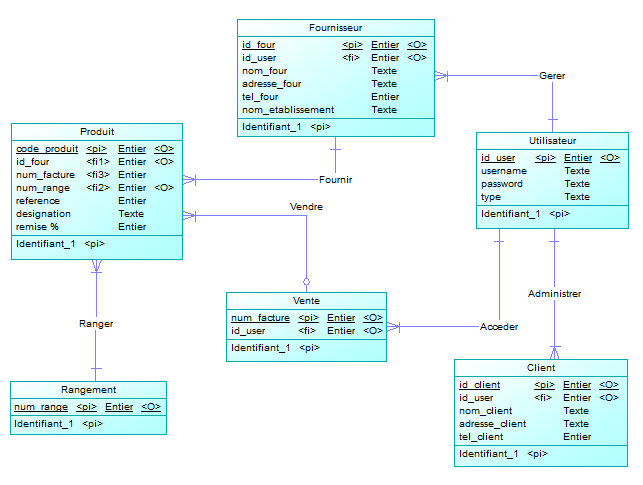
**Tableau 2 : Dictionnaire des données**

### II.4.2. Le modèle Logique de données (MLD)

Le modèle logique de données MLD fournit une description des données tenant compte de moyennes informatiques mises en œuvre. Il complète le MCD en introduisant la notion d’organisation. Il indique donc comment les données seront organisées.

Règles de passage du MCD au MLD :

* Pour les entités : toute entité devient une table, les propriétés de l’entité sont les attributs de la table, l’identifiant de l’entité est la clé primaire de la table.
* Pour les associations : cela dépend des cardinalités. Deux cas sont possibles :
* Association (1,1) – (1,n) : la relation est matérialisée par l’ajout d’une clé étrangère.
* Association (x,n) – (x,n) : la relation donne lieu à la création d’une table.
* Les cardinalités (0,1) – (0,n) se traitent comme les cardinalités (1,1) – (1,n).



**Figure 4 : Modèle logique de données (MLD)**

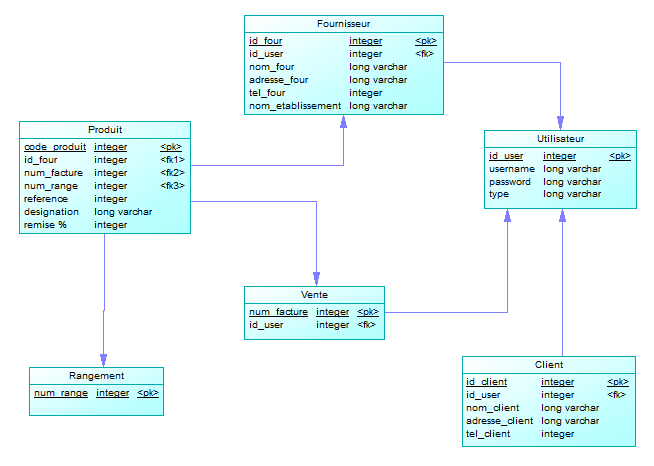
### II.4.3. Le modèle Physique de données (MPD)

**Le modèle physique des données** (MPD) d'une base de données permet d'avoir une représentation graphique de la structure d’une base de données et ainsi de mieux comprendre les relations entre les différentes tables. Il permet d'avoir un point de vue global sur l'ensemble de la base de données.

Règles de passage du MLD au MPD :

Le passage du MLD au MPD exige que les tables, qui jusque-là externe à la base de données, se traduisent en fichiers faisant partie intégrante de la base de données. Ainsi :

* Les tables décrites au niveau du schéma logique deviennent des fichiers de données appelées <<tables>>.
* Les propriétés deviennent des champs de tables.
* Les identifiants deviennent des clés primaires.
* Les clés héritées deviennent des clés étrangères.



**Figure 5 : Modèle physique de données**

## II.5. CONCLUSION

Dans ce chapitre, nous avons présenté trois parties ; la première sur le système d’information et les bases de données, la deuxième sur la méthode Merise, la troisième sur la méthode conceptuelle.

Les bases de données permettent le stockage des informations qui sont structurées dans le but de faciliter l’exploitation des différents programmes. Quant au système de gestion de bases de données (SGBD), il peut être vu comme le logiciel destiné au stockage et à la manipulation de la base de données.

Dans la deuxième partie, nous avons vu le principe de la méthode Merise qui est d'une part une méthode de conception de S.I. et d'autre part une démarche méthodologique de développement.

Dans la troisième partie, l’étude conceptuelle consiste à concevoir le système par trois niveaux d’abstraction (conceptuel, organisationnel et opérationnel).

# 

# TROISIEME PARTIE :

# LE CHOIX DU LANGAGE ET LA REALISATION

## 

## III.1. INTRODUCTION

Après analyse et conception de notre application, nous entamons la phase de réalisation de cette application qui utilise des technologies spécifiques à ces besoins.

Dans ce chapitre nous présenterons en premier lieu l’environnement de développement et les langages utilisés, nous passerons par la suite à la présentation de l’organigramme d’interaction et les différentes interfaces de notre application.

## III.2. OUTILS DE DÉVELOPPEMENT

### III.2.1. WampServer

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.



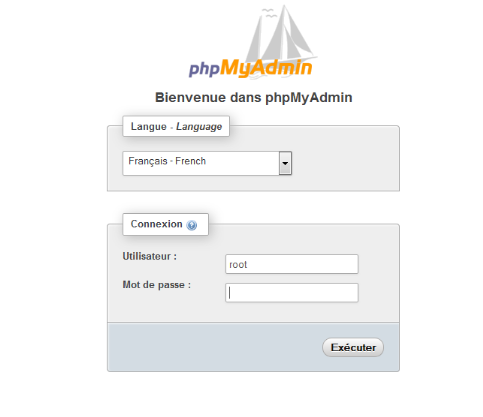
**Figure 6 : Logo WampServer**

#### III.2.1.1. L’interface PHPMyAdmin

PhpMyAdmin est une application web développée en PHP qui permet d’administrer un serveur MySQL (sous réserve de disposer d’un compte utilisateur MySQL ayant les droit suffisants).

PhpMyAdmin permet de :

* Gérer les bases de données de serveur MySQL.
* Gérer les utilisateurs et les droits d’accès à la base de données.
* Gérer les différents objets d’une base de données (table, colonnes, index, etc.).
* Editer et exécuter les requêtes SQL.
* Charger des fichiers textes dans des tables.
* Exporter ou importer des tables.
* Exporter les données des tables dans différents formats (CVS, XML, PDF,…etc.).



**Figure 7 : Interface Connexion PhpMyAdmin**

#### III.2.1.2. Les serveurs Apache/MySQL

##### Apache

* + **Présentation de serveur apache**

Apache est l’un des serveurs les plus répondu sur internet et il basé sur le protocole http.

L’objectif est de fournir, à un maximum de personne et d’organisation, une plateforme solide pour réaliser des tests et des applications de productions, l’un des avantages est son fonctionnement sur de nombreux système d’exploitation dont Microsoft, Linux, etc.

* + **Les fonctionnalités du serveur Apache**

Transformation de l’URL en fichier script.

Détermination du type de données, de la taille des données, des langages, etc.

Vérification d’accès.

Envoi de la réponse au client.

Traitement évolués (accès à des bases de données).

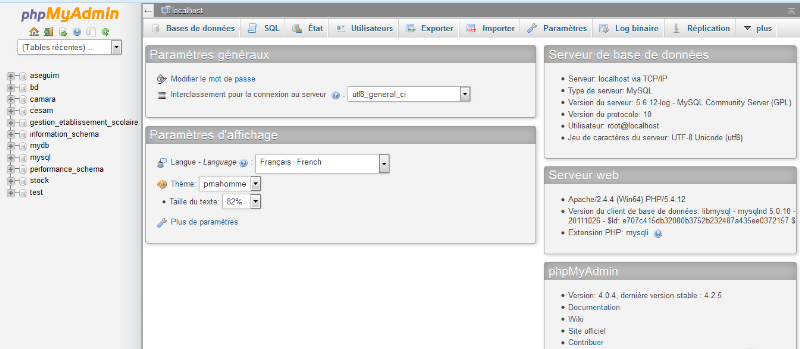
##### MySQL

* + **Présentation du serveur MySQL**

MySQL est un système de gestion de bases de données relationnelles (SBGDR) robuste et rapide. Une base de données permet de manipuler les informations de manière efficace, de les trier, de les lire d’y effectuer des recherches. Le serveur MySQL contrôle l’accès aux données pour s’assurer que plusieurs utilisateurs peuvent se servir simultanément d’une même base de données pour y accéder rapidement. MySQL est donc un serveur multiutilisateur et multithread.

* + **Caractéristique de MySQL**

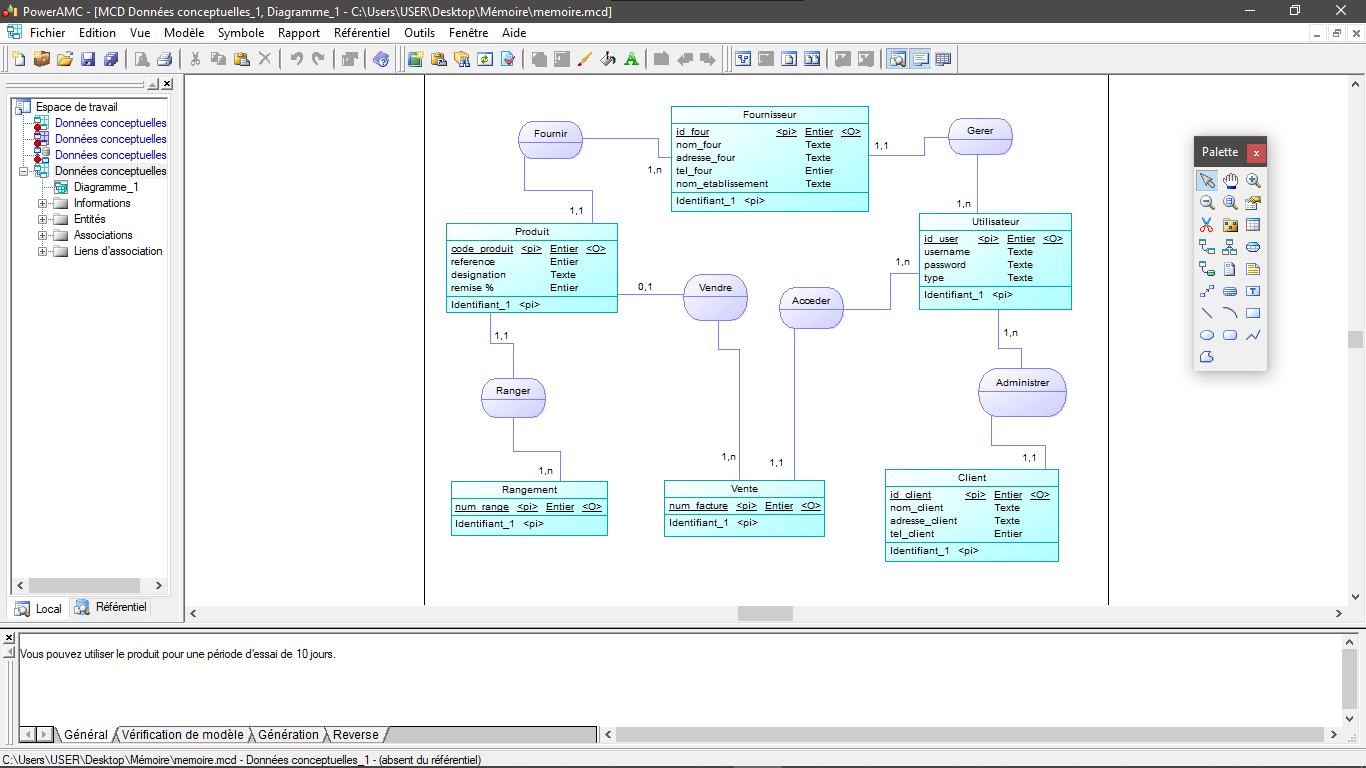
MySQL est un serveur de base de données relationnelle, il a l’avantage d’être orienté vers le service de donnée déjà en place que vers celui de mises à jour fréquentes sécurisées. Il est multithread, c’est-à-dire qu’il gère plusieurs processus en même temps et aussi multiutilisateurs.



#### **Figure 8 : Interface PhpMyAdmin**

### III.2.2. PowerAMC

PowerAMC est une solution de modélisation d’entreprise graphique qui prend en compte les méthodologies et notations de modélisation standard et permet de procéder au reverse engineering ou à la génération de code de façon automatique en utilisant des templates personnalisables. PowerAMC fournit également une solution de référentielle d’entreprise évolutive qui est dotée de fonctionnalités de sécurité et de gestion des versions très robustes qui vous aident à gérer des projets de développement multi-utilisateurs et de fonctionnalités de rapport très puissantes, et qui est en outre largement extensible.



**Figure 9 : Interface PowerAMC**

### III.2.3. NetBeans

NetBeans est un [environnement de développement intégré](https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9) (IDE), placé en [open source](https://fr.wikipedia.org/wiki/Open_source) par [Sun](https://fr.wikipedia.org/wiki/Sun_Microsystems) en [juin 2000](https://fr.wikipedia.org/wiki/Juin_2000) sous licence CDDL et GPLv2 ( [Common Development and Distribution License](https://fr.wikipedia.org/wiki/Common_Development_and_Distribution_License)) pour la création de programmes d'ordinateur dans un certain nombre de langues différentes.

Développement NetBeans se réfère au processus de l'utilisation de NetBeans pour créer, éditer et organiser le code qu’on développe dans un programme informatique.



**Figure 11 : Logo NetBeans**

En plus de Java, NetBeans permet également de supporter différents autres langages, comme  [Python](https://fr.wikipedia.org/wiki/Python_%28langage%29), [C](https://fr.wikipedia.org/wiki/C_%28langage%29), [C++](https://fr.wikipedia.org/wiki/C_plus_plus), [JavaScript](https://fr.wikipedia.org/wiki/JavaScript), [XML](https://fr.wikipedia.org/wiki/Extensible_Markup_Language), [Ruby](https://fr.wikipedia.org/wiki/Ruby), [PHP](https://fr.wikipedia.org/wiki/PHP) et [HTML](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language). Il comprend toutes les caractéristiques d'un IDE moderne ([éditeur en couleur](https://fr.wikipedia.org/wiki/Coloration_syntaxique), projets [multi-langage](https://fr.wikipedia.org/wiki/Multi-langage), [refactoring](https://fr.wikipedia.org/wiki/Refactoring), éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://fr.wikipedia.org/wiki/Linux), [Solaris](https://fr.wikipedia.org/wiki/Solaris_%28informatique%29) (sur [x86](https://fr.wikipedia.org/wiki/X86) et [SPARC](https://fr.wikipedia.org/wiki/Architecture_SPARC)), [Mac OS X](https://fr.wikipedia.org/wiki/Mac_OS_X) ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java).

Dans ce contexte, le développement se réfère à un codage à l'appui, le débogage et la compilation du code dans ces langues. Dans la phase de codage, NetBeans vérifie le code en temps réel pour assurer une syntaxe correcte. NetBeans comprend à un certain nombre d'outils de débogage pour nous aider à isoler et corriger les bugs non liés à la syntaxe. NetBeans peut aussi compiler le code dans un programme de travail.

Pour que notre code soit compilé il faut que le logiciel soit installé avec le kit de développement associé à chaque version (dans notre cas on a pris jdk\_8u40-nb-8\_0\_2Windows-x64 /pour NetBeans IDE 8.0.2).

Un environnement Java Development Kit [JDK](https://fr.wikipedia.org/wiki/JDK) est requis pour les développements en Java. Le Java Développement Kit, communément appelé JDK, est le kit de développement de base que propose gratuitement la société Oracle. Le Kit de développement comprend plusieurs outils, parmi lesquels :

* **Javac** : le compilateur Java, c'est-à-dire qu'il transforme le code source en bytecode, un fichier binaire intermédiaire interprétable par la machine virtuelle sur n'importe quelle plate-forme.
* **Java** : un interpréteur d'applications (machine virtuelle) permettant de lire le byte code.
* **Applet Viewer :** un interpréteur d'applets, a pour but de pouvoir visualiser l'exécution d'un applet (il est aussi possible de la visualiser sur la plupart des navigateurs internet.
* **Jdb :** un débogueur.
* **Javap :** un décompilateur, pour revenir du byte code au code source.
* **Javadoc :** un générateur de documentation, c’est un utilitaire permettant de créer une documentation au format HTML à partir des programmes, grâce aux commentaires /\*\* \*/ prévus à cet effet, et qui sont à incorporer dans le code.
* **Jar :** l'éditeur d'archives Java, c’est un utilitaire permettant de compresser toutes les classes d'un même projet dans une archive (formellement, un fichier zip avec l'extension .jar) pour constituer une bibliothèque de classes que l'on peut ensuite importer comme bibliothèque externe à un autre projet (avec l'option -cp). Il est aussi possible de créer une archive exécutable, c'est à dire définir une classe principale dont la méthode main sera automatiquement exécuté, sans avoir à ouvrir l'archive.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque [Swing (Java)](https://fr.wikipedia.org/wiki/Swing_%28Java%29)). L'[IDE](https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9) NetBeans s'appuie sur cette plateforme.

## III.3. LANGAGES UTILISÉES

### III.3.1. Langage de requête SQL

SQL Signifie « Structured Query Language » c'est-à-dire « Langage d’interrogation structuré ».

En fait SQL est un langage complet de gestion de base de données relationnelle. Il permet de communiquer avec une base de données afin de gérer ou de l’interroger. Il s’agit d’un langage déclaratif à la syntaxe très simple, qui figure parmi les plus utilisés pour l’accès aux bases de données.

SQL permet l’interaction avec le serveur et les informations qu’il héberge en soumettant une commande au SGBD sous la forme d’une requête.



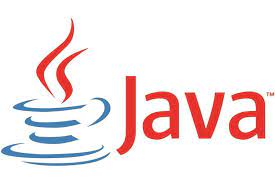
**Figure 12 : Logo SQL**

### III.3.2. Langage de programmation Java

Java est à la fois un langage de programmation et un environnement d’exécution. Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d’exploitation tels qu’Unix, Microsoft Windows, Mac Os ou Linux avec peu ou pas de modifications.

Java permet de développer des applications autonomes mais aussi, et surtout, des applications client-serveur. Coté client, les applets sont à l’origine de la notoriété du langage. C’est surtout coté serveur que Java s’est imposé dans le milieu de l’entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (Java Server Pages).

Les applications Java peuvent être exécutées sur tous les systèmes d’exploitation pour lesquels a été développée une plate-forme Java, donc le nom technique est JRE (Java Runtime Environnement d’exécution Java). Cette dernière est constituée d’une JVM (Java Virtual Machine-Machine Virtuelle Java), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d’une bibliothèque standard à partir de laquelle doivent être développés tous les programmes en Java. C’est la garantie de probabilité qui a fait la réussite de Java dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes.



**Figure 13 : Logo JAVA**

## III.4. ORGANIGRAMME D’INTERACTION ET LES DIFFÉRENTES INTERFACES

### III.4.1. Organigramme d’interaction

L’organigramme d’interaction est la manière dont marche l’application.

NON

OUI

Se déconnecter

Authentification

Saisir le login et le mot de passe

Authentification

Réussi

Accès à la page principale

\_Consultation des informations de l’établissement

\_Accès à la page de gestion des utilisateurs

\_Sélection d’une rubrique

\_Accès à la page de gestion des clients

\_Accès à la page de gestion des fournisseurs

\_Accès à la page de gestion du cahier

\_Accès à la page de gestion des produits

Appliquer sur la tache sélectionnée une des opérations suivantes :

\_Ajouter

\_Modifier

\_Supprimer

\_Actualiser

\_Rechercher

Retour à la page d’authentification

Se déconnecter

NON

Oui

**Figure 14 : Organigramme d’interaction des utilisateurs avec l’application**

### III.4.2. Les différentes interfaces

### III.4.2.1. PAGE D’AUTHENTIFICATION OU LOGIN

La figure représente l’interface d’authentification ou Login, cette fdernière permet à l’utilisateur de s'introduire au menu de l’application. Il devra saisir le nom d’utilisateur ou Username, le mot de passe ou Password.



**Figure 15 : Page d’authentification ou Interface Login**

Le système vérifie le nom d’utilisateur et le mot de passe. Si les informations ne sont pas correctes le système affiche un message d’erreur sinon le système affiche le menu principal.



**Figure 16 : Page d’authentification en « cas d’erreur »**

Au cas ou il y a erreur avec le « **username** » ou le « **mot de passe** » il y a message d’erreur qui s’affiche disant que **login de l’utilisateur ou le mot de passe est incorrect**.



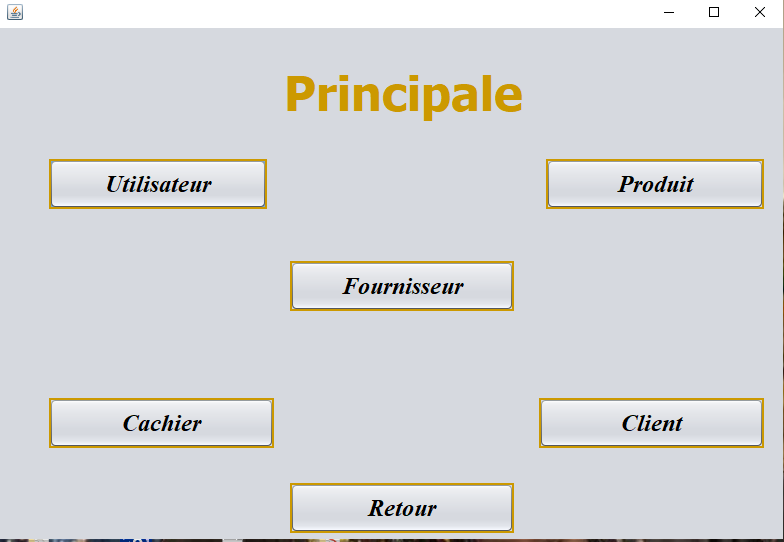
**Figure 17 : Page d’authentification en « cas d’erreur » avec message de modification**

Le message en rouge nous dit exactement ce que l’on doit faire en cas d’erreur.

### III.4.2.2. MENU PRINCIPALE

Après une fois l’utilisateur authentifié, la figure présente permet de voir les informations de l’établissement.

La forme de menu principal contient les différentes parties de l’application permettant d’accéder aux différents formulaires. Elle comprend six interfaces qui sont : Utilisateur, Client, Fournisseur, Produit et Cachier.



**Figure 18 : Interface Principale**

### III.4.2.3. GESTION DES UTILISATEURS

Cette interface permet de gérer les différents utilisateurs qui peuvent accéder à l’application, elle permet aussi la consultation, l’ajout, la modification, la suppression et l’actualisation des informations des comptes utilisateurs.



**Figure 19 : Interface Utilisateur**

### III.4.2.4. GESTION DES CLIENTS

Cette interface permet de gérer les différents clients de l’établissement, elle permet aussi la consultation, l’ajout, la modification, la suppression et l’actualisation des informations des clients.



**Figure 20 : Interface Client**

### III.4.2.5. GESTION DES FOURNISSEURS

Cette interface permet de gérer les différents fournisseurs de l’établissement, elle permet aussi la consultation, l’ajout, la modification, la suppression et l’actualisation des informations des fournisseurs.



**Figure 21 : Interface Fournisseur**

### III.4.2.6. GESTION DES PRODUITS

Cette interface permet de gérer les différents produits de l’établissement, elle permet aussi la consultation, l’ajout, la modification, la suppression et l’actualisation des informations des produits.



**Figure 22 : Interface Produit**

### III.4.2.7. CACHIER / FACTURE

La figure représente l’interface d’entrée et sortie, cette interface permet à l’utilisateur de voir son stock et créer une facture de vente.

Elle est composée de trois parties : le tableau des produits, la partie d’ajout d’une nouvelle vente et la partie facture ou l’on peut avoir accès aux revenues.



**Figure 23: Interface Cachier/Vente**

La figure représente l’interface erreur au cas où l’on oublie un des données on reçoit un message d’erreur qui nous demande d’entrer toutes les données qui est **« SVP entrer vos données ».**



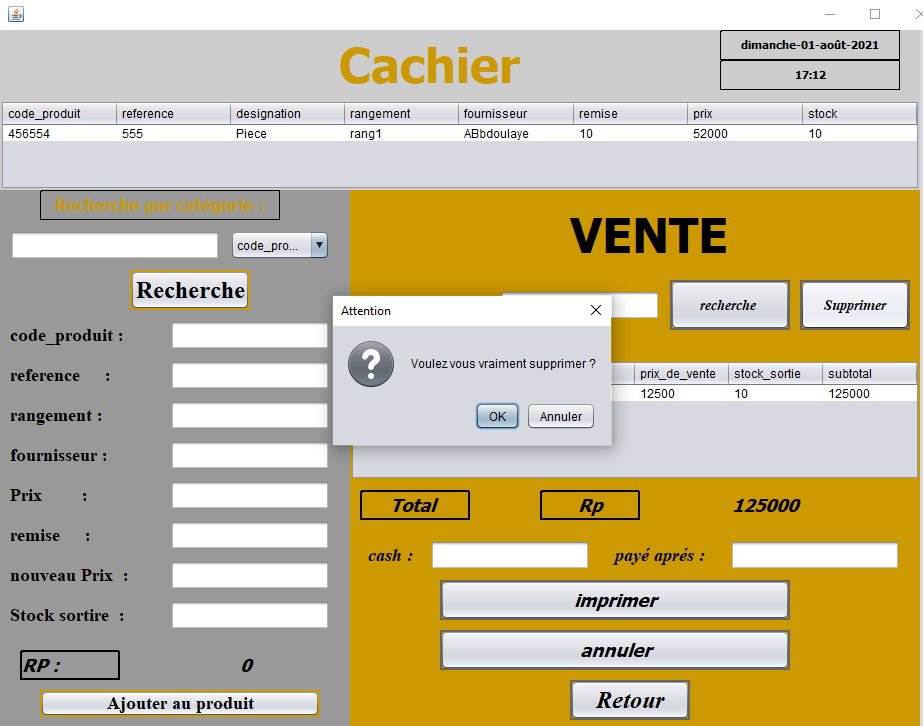
**Figure 24 : Interface Cachier/Vente en « cas d’erreur »**

La figure représente l’interface erreur au cas où l’on oublie d’entrée le numéro de la facture qui est le pilier principal de cette interface. Donc on a un message qui dit **« SVP entrer le numéro de la facture ».**



**Figure 25 : Interface Cachier/Vente en « cas d’oubli du numéro de la facture »**

La figure représente l’interface de suppression d’une nouvelle entrée de vente au cas où on veut supprimer une facture on reçoit un message d’avertissement qui demande **« voulez-vous vraiment supprimer ? »** sois on valide ou on annule.



**Figure 26 : Interface Cachier/Vente en « cas de suppression »**

### III.4.2.8. RECU

La figure représente l’interface Reçu ou après avoir entré le nouveau prix et le stock on peut imprimer notre reçu qui sont les dépenses, les prix d’achats et les bénéfices.



**Figure 27 : Interface Reçu**

### III.5. CONCLUSION

Dans ce chapitre, nous avons réalisé une présentation des outils de développement et les langages utilisés, ainsi que quelques interfaces de notre application et leurs fonctionnements.

# CONCLUSION GENERALE

Au cours de ce mémoire, nous avons présenté les différentes étapes de la conception et la réalisation de l’application pour la gestion du magasin de pièces détachées « Keur Sylla Transport ».

Afin de satisfaire les besoins des utilisateurs et de leur facilité leur stock ou bien de bien gérer leurs clients ainsi que leurs fournisseurs, nous avons commencé la conception en utilisant le formalisme de Merise et la mise en œuvre des bases de données avec le gestionnaires de bases de données MySQL, ensuite l’implémentation des requêtes SQL pour la manipulation des données, puis la création des modèles aves PowerAMC et enfin la concrétisation du logiciel sous l’environnement de programmation NetBeans avec le langage de programmation Java.

Ce projet a fait l’objet d’une expérience intéressante, qui nous a permis d’améliorer nos connaissances et nos compétences dans le domaine de la programmation et de la base de données.

En guise de perspective, ce travail reste prêt pour toute amélioration envisageable comme ajouter les bons de commande, créer un dispositif permettant d’imprimer les reçus et surtout permettre aux utilisateurs de les avoir sur leurs téléphones portables afin de pouvoir suivre et gérer leurs stocks à n’importe quel moment et n’importe où.

# WEBOGRAPHIE

<https://www.wikipedia.org/>

<https://www.google.com/>

<https://netbeans.apache.org/>

<https://openclassrooms.com/fr/courses/6173501-debutez-la-programmation-avec-java/6458461-gerez-les-piles-de-donnees-avec-la-bonne-collection>

<https://dev.mysql.com/doc/refman/8.0/en/>

<https://merise.developpez.com/>

<https://www.youtube.com/watch?v=XFK1fLxVgxU>

<https://www.memoireonline.com/>

# BIBLIOGRAPHIE

**Mémoire par Ben Amara Houa et Ben Yahia Housna :** Conception et Réalisation d'une application pour la gestion des stocks Cas d’étude : les cantines scolaires de l’éducation

**Mémoire par Bordji Fatima Zohra et Bouakkaz Khadîdja :** Gestion D’un Magasin De Pièces De Rechange GMPR (Conception et réalisation d’une application)

**Mémoire par Hamdaoui Abdelilah :** Réalisation d’une application de gestion Commerciale

**Compte Rendu Stage :** Application à l’automatisation de la comptabilité matière à la Division de gestion des services centraux du Ministère des Affaires Etrangères

# TABLE DES FIGURES

# 

[**Figure 1 :** **Vision globale d’une entreprise** 13](file:///C:\Users\USER\Desktop\Mémoire\RÉDACTION%20MÉMOIRE%20%20(1).docx#_Toc79858789)

[**Figure 2 : Les trois niveaux d’une architecture de base de données** 15](file:///C:\Users\USER\Desktop\Mémoire\RÉDACTION%20MÉMOIRE%20%20(1).docx#_Toc79858790)

[**Figure 3 : Modèle conceptuel de données (MCD)** 20](#_Toc79858791)

[**Figure 4 : Modèle logique de données (MLD)** 22](#_Toc79858792)

[**Figure 5 : Modèle physique de données** 23](#_Toc79858793)

[**Figure 6 : Logo WampServer** 26](#_Toc79858794)

[**Figure 7 : Interface Connexion PhpMyAdmin** 27](#_Toc79858795)

[Figure 8 : Interface PhpMyAdmin 29](#_Toc79858796)

[**Figure 9 : Interface PowerAMC** 30](#_Toc79858797)

[**Figure 11 : Logo NetBeans** 30](#_Toc79858798)

[**Figure 12 : Logo SQL** 32](#_Toc79858799)

[**Figure 13 : Logo JAVA** 33](#_Toc79858800)

[**Figure 14 : Organigramme d’interaction des utilisateurs avec l’application** 36](#_Toc79858801)

[**Figure 15 : Page d’authentification ou Interface Login** 36](#_Toc79858802)

[**Figure 16 : Page d’authentification en « cas d’erreur »** 37](#_Toc79858803)

[**Figure 17 : Page d’authentification en « cas d’erreur » avec message de modification** 38](#_Toc79858804)

[**Figure 18 : Interface Principale** 39](#_Toc79858805)

[**Figure 19 : Interface Utilisateur** 40](#_Toc79858806)

[**Figure 20 : Interface Client** 41](#_Toc79858807)

[**Figure 21 : Interface Fournisseur** 42](#_Toc79858808)

[**Figure 22 : Interface Produit** 43](#_Toc79858809)

[**Figure 23: Interface Cachier/Vente** 44](#_Toc79858810)

[**Figure 24 : Interface Cachier/Vente en « cas d’erreur »** 45](#_Toc79858811)

[**Figure 25 : Interface Cachier/Vente en « cas d’oubli du numéro de la facture »** 46](#_Toc79858812)

[**Figure 26 : Interface Cachier/Vente en « cas de suppression »** 47](#_Toc79858813)

[**Figure 27 : Interface Reçu** 48](#_Toc79858814)

# TABLE DES TABLEAUX

[**Tableau 1 : Les niveaux d’abstraction de Merise** 21](#_Toc79169849)

[**Tableau 2 : Dictionnaire des données** 25](#_Toc79169850)