

## EPIGRAPHE

---

« L'innovation, c'est transformer les problèmes quotidiens en solutions durables »

IMBEY Christenvie

## DÉDICACE

A notre père Bernard IMBEY et à ma mère Marthe ABONDUKA,  
Pour votre amour infini, les conseils, les sacrifices et le soutien que vous  
m'avez toujours donnés. Ce travail vous revient de droit.

A nos frères :

Daddy IMBEY (et son épouse Olive), Jeannot IMBEY (et son épouse Francine),  
Modeste IMBEY, Chancel IMBEY et Jhospin IMBEY,

Merci pour votre présence, votre aide et votre confiance nous ont beaucoup  
soutenus.

A nos sœurs :

Flavie IMBEY et Bethy IMBEY,

Merci pour votre affection, vos encouragements et votre gentillesse tout au  
long de ce parcours.

A celle qui partagera notre vie un jour,

A nos futurs enfants,

A l'entreprise qui nous a permis de mettre en pratique nos compétences,

Nous dédions cette œuvre

IMBEY Christenvie

## REMERCIEMENTS

Au terme de ce travail, nous tenons à exprimer notre profonde gratitude à Monsieur le Professeur ANGOMA MONGA SINDANI Blaise, Directeur de ce mémoire, pour sa disponibilité, ses orientations précieuses, ses remarques constructives ainsi que la rigueur intellectuelle dont il a fait preuve pendant de la conduite de ce projet.

Nos sincères remerciements vont également à Monsieur l'Assistant MAYI LUKONGO YALEM Didier, Co-directeur de ce mémoire, pour son encadrement attentif, son accompagnement constant et ses conseils avisés qui nous ont guidés jusqu'à l'aboutissement de ce projet.

Nous exprimons notre reconnaissance à tout le corps professoral de l'Institut Supérieur Pédagogique Technique de Kinshasa (ISPT-KIN) pour la qualité de l'enseignement reçu, ainsi qu'à l'institution pour l'environnement d'apprentissage qu'elle m'a offert.

Nous remercions toutes les personnes qui nous ont soutenus de près ou de loin, tant moralement que financièrement, tout au long de notre formation.  
Nous remercions tout particulièrement :

Délia Louange MOMPONGO, Princesse Christianna IMBEY, Sarah MPOTE, Joyce YULUBANI, Exaucé MALONGA, Anders BOBWI.

Pour leur soutien inestimable, leurs encouragements et leur présence à nos côtés dans les moments importants.

Notre profonde gratitude va également à : Christopher BOKOLO PANGA PANGA, Grace PANGA PANGA, Taylor BOKIMI

Pour leur aide précieuse et leurs encouragements constants.

Nous remercions aussi chaleureusement nos collègues pour leurs conseils, leurs encouragements et leur collaboration tout au long de cette aventure académique.

Une pensée particulière pour : Président Wims KAPANDA, Cédric LUENDU,  
Félicité MAPWAYA, Fils TENGE, KITOKO MUKULU, Grâce KIWA.

A toutes celles et tous ceux qui ont, d'une manière ou d'une autre, contribué  
à l'accomplissement de cette œuvre, nous vous disons merci du fond du  
cœur.

IMBEY Christenvie

## AVANT-PROPOS

Ce travail est le fruit de plusieurs mois d'efforts, d'apprentissage et d'engagement. Il marque la fin d'un parcours académique et le début d'une nouvelle étape dans notre vie professionnelle. Tout au long de cette formation, Nous avons acquis de nombreuses connaissances, aussi bien théoriques que pratiques, qui nous ont permis de mieux comprendre les enjeux de la gestion des stocks dans une entreprise.

Ce projet, intitulé STOCK MASTER, a été réalisé pour répondre à un besoin réel : celui de simplifier et d'automatiser la gestion des achats et des ventes dans une entreprise commerciale. Il m'a permis d'appliquer les compétences acquises dans le domaine de la programmation, de la gestion de base de données, ainsi que de la conception d'interfaces utilisateur.

Je tiens à souligner que ce travail n'aurait pas été possible sans l'aide de nombreuses personnes que je remercie avec beaucoup de reconnaissance. Ce projet est aussi l'occasion pour moi de me préparer à la vie professionnelle avec plus de confiance et de compétence.

## RÉSUMÉ

Ce travail présente le projet STOCK MASTER, une application de gestion des stocks développée avec le langage Python. Elle permet de suivre les achats, les ventes, les produits disponibles en stock, et de gérer les utilisateurs.

L'application est conçue avec l'interface graphique Tkinter, et utilise SQLite pour le stockage des données.

L'objectif principal est de permettre aux petites et moyennes entreprises de gérer facilement leurs produits sans avoir besoin de logiciels complexes ou coûteux. L'application est simple, rapide et accessible, même pour ceux qui ont peu de connaissances en informatique.

Le mémoire est structuré en trois parties principales :

1. Les bases théoriques du développement logiciel et de la gestion des stocks.
2. L'analyse du besoin et la conception du système.
3. La mise en œuvre de l'application avec des exemples pratiques.

Ce projet montre qu'avec des outils gratuits et accessibles, il est possible de créer des solutions efficaces pour répondre à des besoins concrets.

## LISTE DES ABREVIATIONS

	Pages
ERP 1.	Enterprise Resource Planning – Progiciel de gestion intégré.. 23
EXE 2.	Executable file – Fichier exécutable Windows ..... 50
FEFO 3.	First Expired First Out – Premier expiré, premier sorti ..... 23
FIFO 4.	First In First Out – Premier entré, premier sorti ..... 23
GUI 5.	Graphical User Interface – Interface graphique utilisateur . 33
IDE 6.	Integrated Development Environment – Environnement de développement intégré ..... 18
ISPT-KIN 7.	Institut Supérieur Pédagogique Technique de Kinshasa ... 03
LIFO 8.	Last In First Out – Dernier entré, premier sorti ..... 23
PDF 9.	Portable Document Format ..... 14
Py 1.	Préfixe ou suffixe courant associé à Python ..... 50
QR Code 2.	Quick Response Code – Code à réponse rapide ..... 46
RFID 3.	Radio Frequency Identification – Identification par radiofréquence ..... 07
UML 4.	Unified Modeling Language – Langage de modélisation unifié 07

## LISTE DES FIGURES

	Pages
Figure 1.1 : Interface de l'environnement PyCharm .....	19
Figure 2.1 : Diagramme de cas d'utilisation .....	31
Figure 2.2 : Diagramme de classes .....	32
Figure 3.1 : Diagramme simplifié des tables .....	35
Figure 3.2 : Page de connexion .....	37
Figure 3.3 : Page d'inscription .....	38
Figure 3.4 : Chargement .....	39
Figure 3.5 : Menu principal .....	40
Figure 3.6 : Page d'achats .....	41
Figure 3.7 : Page de ventes .....	42
Figure 3.8 : Page de bénéfices .....	43
Figure 3.9 : Page de produits disponibles .....	44
Figure 3.10 : Page de réinitialisation du mot de passe .....	45
Figure 3.11 : Scanner les produits via un Code QR .....	46
Figure 3.11 : Exportation en exécutable .....	49

## LISTE DES TABLEAUX

	Pages
Tableau 1.1 : Caractéristiques des langages de programmation ...	17
Tableau 1.2 : Outils et bibliothèques .....	20
Tableau 2.1 : Forces et faiblesses du système actuel .....	29
Tableau 3.1 : Caractéristiques de logiciels .....	33
Tableau 3.2 : Résultats obtenus .....	53

## 0.1. Mise en Contexte

Dans un monde de plus en plus numérisé, la gestion efficace des ressources, et en particulier des stocks, est devenue un enjeu central pour les entreprises commerciales. Le développement logiciel permet aujourd’hui de concevoir des solutions robustes pour automatiser, surveiller et optimiser la gestion des biens, produits et marchandises. Ce chapitre présente les bases théoriques relatives au développement logiciel ainsi qu’à la gestion des stocks, en posant les fondements nécessaires à la compréhension du projet STOCK MASTER.

## 0.2. Choix et Intérêt du Sujet

### 0.2.1. Choix du Sujet

Le choix de développer une application de gestion de stock provient d'un constat personnel : plusieurs supermarchés dans mon entourage rencontrent des difficultés dans la maîtrise de leurs mouvements de stock. Ces difficultés se manifestent par des pertes fréquentes de produits, des erreurs humaines lors des ventes, et une incapacité à évaluer de manière précise le niveau des bénéfices. Dans certains cas, des produits expirés restent en rayon faute de traçabilité claire. Ce constat a motivé l'idée de concevoir une solution numérique, accessible même à des utilisateurs non techniciens, pour pallier à ces problèmes.

### 0.2.2. Intérêts du sujet

a) Intérêt personnel

Ce projet m'a permis d'approfondir mes connaissances pratiques en programmation avec Python, en conception d'interfaces graphiques avec Tkinter, et en gestion de bases de données SQLite. Il s'inscrit dans mon ambition de devenir développeur d'applications de gestion et m'a offert une occasion unique de concrétiser une idée depuis la phase d'analyse jusqu'à l'implémentation.

#### b) Intérêt scientifique

Sur le plan scientifique, ce projet aborde des concepts fondamentaux comme l'analyse des besoins, la modélisation UML, l'architecture logicielle, la structuration des données, et l'automatisation de processus manuels. Il constitue un cas d'étude complet pour illustrer l'impact de l'informatique dans les systèmes organisationnels modernes.

#### c) Intérêt professionnel

La mise en œuvre de STOCK MASTER m'a offert un cadre réaliste pour expérimenter les pratiques professionnelles dans le développement logiciel, en mettant en application les étapes classiques d'un cycle de vie de projet (analyse, conception, développement, test et déploiement). Cela renforce ma capacité à répondre à des projets en entreprise.

#### d) Intérêt environnemental

Un bon système de gestion de stock permet d'éviter les surplus et les gaspillages en assurant un suivi rigoureux des produits, notamment alimentaires. Cela contribue indirectement à la protection de l'environnement, à travers la limitation des déchets issus de la surconsommation ou de la péremption non détectée.

### 0.3. Problématique

La gestion manuelle des stocks dans les supermarchés présente de nombreuses insuffisances : erreurs de saisie, lenteur des opérations, absence de traçabilité, difficulté de calcul des marges bénéficiaires, et manque de

centralisation des informations. Ces faiblesses nuisent à la rentabilité du commerce et à la satisfaction des clients.

Ainsi, la problématique générale que ce travail tente de résoudre peut se formuler comme suit :

- Comment développer un système informatique capable de gérer efficacement les entrées et sorties de produits dans un supermarché comme Kin-Marché, tout en assurant un suivi fiable, une traçabilité des mouvements de stock, et une réduction des pertes ?

#### 0.4. Hypothèses

Le développement de cette application se fonde sur les hypothèses suivantes:

H1 : L'implémentation d'un système informatisé améliore significativement la gestion des stocks.

H2 : L'automatisation réduit les erreurs humaines, optimise les opérations commerciales et accroît la rentabilité.

H3 : Une application bien conçue peut être adoptée même par des utilisateurs sans formation informatique approfondie.

#### 0.5. Objectifs

##### 0.5.1. Objectif général

Mettre en place une solution logicielle intégrée et automatisée pour la gestion des stocks d'un supermarché, avec des modules de connexion, d'achats, de ventes, de statistiques et de sécurité.

##### 0.5.2. Objectifs spécifiques

- ❖ Analyser les besoins d'un supermarché en matière de gestion de stock.
- ❖ Concevoir une base de données relationnelle pour stocker les informations.
- ❖ Développer une interface utilisateur conviviale en utilisant Python et Tkinter.
- ❖ Intégrer des fonctionnalités comme l'historique, la génération automatique d'ID, la récupération de mot de passe, etc.
- ❖ Permettre le calcul automatique des bénéfices réalisés sur les ventes.

## 0.6. Méthodes et Techniques

### 0.6.1. Méthodes

La démarche de développement adoptée repose sur la méthode en cascade (waterfall), une approche classique dans les projets structurés, qui comporte les étapes suivantes : analyse, conception, codage, test, déploiement et maintenance.

### 0.6.2. Techniques utilisées

- ❖ Programmation orientée objet avec Python
- ❖ Conception d'interfaces avec Tkinter
- ❖ Utilisation de SQLite comme moteur de base de données
- ❖ Modélisation UML pour la conception du système
- ❖ Utilisation d'outils de test manuel et d'interprétation logique du code

## 0.7. Délimitation

### 0.7.1. Délimitation temporelle

Le développement de ce projet s'étale sur une période allant de décembre 2024 à août 2025, comprenant la phase d'étude, de codage, de tests et de rédaction du rapport.

### 0.7.2. Délimitation géographique

Le projet est appliqué à Kin-Marché, un supermarché situé à Bibwa, comme cas d'étude principal.

## 0.8. Subdivision du travail

Hormis l'introduction générale et la conclusion générale, ce projet s'articulez en trois chapitres comme suit :

Le premier chapitre intitulé « Généralités sur le développement logiciel et la gestion des stocks », rappelle les notions de base d'informatique et de gestion commerciale apprise le long de notre cycle de licence au sein de l'institut ;

Le second ayant pour titre « Analyse de l'existant », présente d'abord le supermarché qui nous a reçu, ensuite procède à l'analyse critique de son existant, enfin propose une solution appropriée au problème trouvé dans ce supermarché ;

Le troisième chapitre, « Mise en œuvre du système de stocks master », se charge de l'implémentation du nouveau système.

## 0.9. Limites

- ❖ Malgré la rigueur appliquée au projet, plusieurs limites ont été identifiées :
- ❖ Limitation des tests réels en situation de charge importante.
- ❖ L'interface graphique reste basique par rapport aux standards d'applications mobiles.
- ❖ L'application fonctionne en local, sans hébergement en ligne.
- ❖ Certaines fonctions comme la gestion de fournisseurs ou la génération de rapports PDF ne sont pas encore intégrées.

## 0.10. Difficultés rencontrées

Au cours du développement, plusieurs difficultés ont été observées :

- ❖ La gestion des erreurs lors des transactions critiques (ventes, achats).
- ❖ L'implémentation de fonctions complexes comme la génération automatique d'identifiants.
- ❖ L'adaptation de l'interface aux utilisateurs novices.
- ❖ Le test du système en conditions réelles limité par l'accès aux données de l'entreprise.

# CHAPITRE I : GENERALITES SUR LE DEVELOPPEMENT LOGICIEL ET LA GESTION DES STOCKS

## 1.1. Introduction

Dans un monde de plus en plus guidé par les technologies de l'information, la gestion des entreprises commerciales ne peut être efficace sans un bon usage de l'informatique. Parmi les aspects les plus stratégiques de cette gestion, celle des stocks occupe une place centrale. Elle influence directement la rentabilité, la satisfaction des clients et la continuité de l'activité. Ce chapitre pose les bases théoriques indispensables à la compréhension du projet "STOCK MASTER" en abordant successivement les généralités sur le développement logiciel, les systèmes de gestion, et les principes fondamentaux de la gestion des stocks.

## 1.2. Généralités sur le développement logiciel

### 1.2.1. Développement logiciel

Le développement logiciel est le processus de conception, de programmation, de test et de maintenance d'applications informatiques. Il s'agit d'un domaine multidisciplinaire qui associe des compétences en programmation, gestion de projet, ingénierie système, ergonomie et sécurité informatique.

### 1.2.2. Cycle de vie du logiciel

Le cycle de vie d'un logiciel suit plusieurs phases :

- ❖ Analyse des besoins : compréhension des attentes des utilisateurs.
- ❖ Spécification : formulation des exigences fonctionnelles et techniques.
- ❖ Conception : choix de l'architecture et des technologies à utiliser.
- ❖ Développement : codage du logiciel.
- ❖ Test et validation : contrôle qualité et corrections.
- ❖ Déploiement : mise en service de l'application.
- ❖ Maintenance : mise à jour, corrections d'erreurs et améliorations futures.

### 1.2.3. Modèles de développement

Il existe plusieurs modèles de développement logiciel, dont :

- ❖ Le modèle en cascade : les étapes sont réalisées séquentiellement.
- ❖ Le modèle en V : met l'accent sur la validation à chaque étape.
- ❖ Le modèle itératif/incrémental : permet d'avancer par versions successives.
- ❖ Le modèle Agile (Scrum, Kanban) : privilégie l'adaptation continue et l'implication du client.

Dans le cadre du projet STOCK MASTER, le modèle de développement utilisé est Le modèle en cascade. Ce modèle est simple et structuré, ce qui est idéal pour un projet académique ou individuel. Les étapes sont claires et linéaires, ce qui facilite la planification et le suivi. Il permet de produire une documentation complète, utile pour la soutenance (comme tu le fais maintenant). Le modèle en cascade est parfaitement adapté à STOCK MASTER car il permet un développement structuré, une meilleure organisation des tâches, et facilite la documentation pour la soutenance.

### 1.2.4. Langages de programmation

Dans le domaine du développement logiciel, le choix du langage de programmation est une décision cruciale qui conditionne la structure du projet, sa maintenabilité, sa compatibilité avec des bibliothèques tierces, ainsi que la facilité de son déploiement. Il existe plusieurs langages qui répondent à divers besoins, allant de la programmation web à la conception d'interfaces graphiques, en passant par la manipulation de données et l'intelligence artificielle. Voici quelques langages populaires dans l'univers du développement logiciel :

## Caractéristiques des langages de programmation

Tableau 1.1 : Caractéristiques des langages de programmation

Langage	Caractéristiques	Domaine
Python	Simple, lisible, orienté objet, vaste écosystème de bibliothèques	Applications desktop, scripts, IA, data science, prototypage rapide
Java	Portable, robuste, multi-plateforme grâce à la JVM	Applications mobiles (Android), logiciels d'entreprise, systèmes embarqués
C#	Langage moderne développé par Microsoft, très utilisé sous Windows	Applications desktop avec WPF, jeux avec Unity, web avec ASP.NET
PHP	Spécialisé dans le développement web côté serveur	Sites web dynamiques, gestion de contenu, e-commerce
JavaScript	Interprété dans le navigateur, orienté client	Développement web interactif, frameworks modernes (React, Angular, etc.)

Le langage Python a été retenu pour le développement de l'application STOCK MASTER en raison de plusieurs atouts majeurs :

1. Simplicité et lisibilité du code

Python est réputé pour sa syntaxe claire et concise. Cela facilite l'apprentissage, la relecture et la maintenance du code. Cette caractéristique est essentielle dans les projets pédagogiques et professionnels où la transparence du développement est cruciale.

2. Richesse des bibliothèques standard et tierces

Python dispose d'un ensemble impressionnant de bibliothèques prêtes à l'emploi, notamment :

- Tkinter pour l'interface graphique.
- SQLite3 pour la base de données.
- Tkcalendar pour les sélecteurs de dates.
- Messagebox pour les dialogues avec l'utilisateur.

### 3. Gain de temps au développement

Grâce à sa syntaxe intuitive et à ses outils intégrés, Python permet de concevoir des applications robustes en un temps réduit.

### 4. Portabilité multiplateforme

Le code Python peut être exécuté sur Windows, Linux et macOS sans modification majeure, ce qui garantit une grande souplesse dans le déploiement.

### 5. Communauté active et documentation abondante

En cas de besoin, la communauté Python offre une quantité considérable de ressources (tutoriels, forums, bibliothèques open source) qui facilitent la résolution des problèmes techniques.

#### 1.2.5. Environnements de développement (IDE)

Un Environnement de Développement Intégré (IDE — *Integrated Development Environment*) est un outil logiciel qui regroupe un ensemble de fonctionnalités facilitant l'écriture, la lecture, le test, la gestion et le débogage de programmes informatiques. Il fournit une interface unifiée rassemblant notamment un éditeur de code, un compilateur ou interpréteur, un débogueur, et parfois des outils de gestion de version.

Parmi les IDE populaires utilisés dans le domaine de la programmation, on peut citer :

- PyCharm
- Visual Studio Code
- Eclipse

Dans le cadre du développement de l'application STOCK MASTER, l'environnement PyCharm a été utilisé comme IDE principal. Ce choix est motivé par plusieurs raisons techniques et ergonomiques adaptées à un projet Python utilisant des bibliothèques comme Tkinter et SQLite.

The screenshot shows the PyCharm interface with the following details:

- Project View:** Shows the project structure under "StockMasterChris [Stock Master Chris]". It includes a ".venv" folder, a "venv" folder, and several Python files: "achatspage.py", "databasemanager.py", "forgotpasswordpage.py", "image1.ico", "image1.jpg", "loginpage.py", "main.py", "menupage.py", "registrationpage.py", "showsplash.py", "stock\_master.db", "styles.py", and "ventespage.py".
- Main Editor:** The file "main.py" is open. The code imports various modules like tkinter, sqlite3, datetime, etc., and defines constants for colors and fonts. It also imports several pages from other files.
- Status Bar:** Shows the path "StockMasterChris > main.py", the status "Launching skeleton generator...", encoding "UTF-8", and Python version "Python 3.12 (StockMasterChris)".

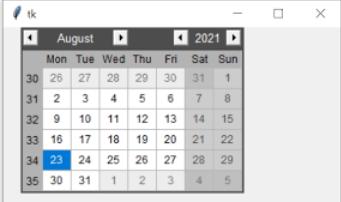
Figure 1.1 : Interface de l'environnement PyCharm

### 1.2.6. Outils et bibliothèques utilisés dans STOCK MASTER

Pour développer une application robuste, fluide et facile d'utilisation comme STOCK MASTER, il est impératif de sélectionner avec soin les outils et bibliothèques adaptés au langage choisi (ici, Python), à la nature du projet (application de gestion avec interface graphique), et aux contraintes techniques (base de données locale, simplicité de déploiement, performances acceptables).

Le projet STOCK MASTER repose principalement sur les technologies suivantes :

Tableau 1.2 : Outils et bibliothèques

Outil / Bibliothèque	Description	Raison du choix
<b>Tkinter</b> 	Bibliothèque standard de Python pour créer des interfaces graphiques (GUI).	Intégrée directement dans Python, facile à utiliser, très bien documentée, et adaptée aux projets éducatifs et professionnels simples.
<b>SQLite</b> 	Moteur de base de données relationnelle léger et embarqué.  Taille faible	Aucun besoin d'installation serveur, stockage dans un seul fichier .db, parfait pour des projets locaux avec structure relationnelle.
<b>Tkcalendar</b> 	Module complémentaire de Tkinter qui permet l'utilisation de calendriers interactifs.	Permet à l'utilisateur de choisir une date dans un calendrier visuel, améliorant l'expérience utilisateur (ex. : date d'achat, de naissance, etc.).
<b>Messagebox</b> 	Fonction de Tkinter pour afficher des messages d'alerte, de confirmation ou d'erreur.	Utile pour guider ou avertir l'utilisateur selon ses actions (ex. : confirmation de suppression, erreur de connexion, etc.).

### 1.3. Généralités sur la gestion des stocks

#### 1.3.1. Définition

La gestion des stocks est l'ensemble des techniques permettant de déterminer, contrôler et optimiser les niveaux de marchandises ou de produits disponibles dans un établissement commercial ou industriel.

#### 1.3.2 . Objectifs de la gestion des stocks

La gestion des stocks constitue une fonction essentielle dans toute organisation, en particulier dans le secteur commercial. Elle vise à garantir que les bons produits sont disponibles au bon moment, en quantité suffisante, sans engendrer de coûts inutiles. Une gestion rigoureuse permet d'assurer un équilibre entre la disponibilité des produits et l'optimisation des ressources financières et logistiques. Les objectifs de la gestion des stocks peuvent être classés selon plusieurs axes stratégiques et opérationnels :

##### 1. Réduire les coûts de stockage.

L'un des principaux objectifs est de minimiser les charges liées au stockage, qui incluent :

- Le coût de l'espace (location ou amortissement d'entrepôts)
- Les frais d'entretien et de sécurité
- Le coût de l'obsolescence ou de la détérioration des produits
- Les coûts administratifs de gestion (main-d'œuvre, outils, logiciels)

Un stock trop important immobilise inutilement du capital et peut engendrer des pertes si les produits deviennent invendables. A travers des systèmes de gestion automatisés comme STOCK MASTER, il devient possible de surveiller en temps réel les niveaux de stock, d'anticiper les besoins et de supprimer les excédents.

## 2. Eviter les ruptures de stock.

Les ruptures de stock sont souvent synonymes de perte de ventes, de clients et d'image de marque. L'objectif est donc d'assurer une disponibilité permanente des produits clés

Grâce à des alertes et des seuils de réapprovisionnement programmés dans l'application STOCK MASTER, l'utilisateur est averti à l'avance lorsqu'un produit atteint un niveau critique. Cela permet de réagir avant que le stock ne soit épuisé.

## 3. Optimiser le réapprovisionnement.

L'optimisation du réapprovisionnement vise à :

- L'optimisation du réapprovisionnement vise à :
- Aux bons moments
- Auprès des bons fournisseurs

Une mauvaise gestion du réapprovisionnement peut entraîner une surcharge de stock ou des délais d'attente trop longs. Avec des outils numériques comme STOCK MASTER, il est possible d'analyser l'historique des ventes pour prévoir les futurs besoins, ce qui améliore la précision des commandes.

## 4. Améliorer le service client.

- Un stock bien géré permet de répondre plus efficacement aux demandes des clients, de respecter les délais de livraison et de renforcer leur satisfaction.
- La rapidité et la fiabilité du service dépendent fortement de la disponibilité des produits. STOCK MASTER contribue à maintenir cet équilibre en facilitant le suivi et la mise à jour constante de l'inventaire.

Les objectifs de la gestion des stocks ne se limitent pas à une simple activité de comptage ou de stockage. Il s'agit d'un levier stratégique qui influence

directement la performance financière, la relation client et la réactivité de l'entreprise. L'application STOCK MASTER a été conçue pour atteindre ces objectifs avec efficacité et fiabilité, tout en offrant une interface conviviale et des fonctions automatisées

### 1.3.3. Types de stocks

- ❖ Stock de sécurité : pour faire face aux imprévus.
- ❖ Stock de cycle : lié au réapprovisionnement régulier.
- ❖ Stock d'anticipation : pour prévoir une hausse de la demande.
- ❖ Stock en transit : en cours de livraison.

### 1.3.4. Méthodes de gestion des stocks

- ❖ FIFO (First In First Out) : les premiers entrés sont les premiers sortis.
- ❖ LIFO (Last In First Out) : les derniers entrés sont les premiers sortis.
- ❖ FEFO (First Expired First Out) : gestion par date de péremption.

### 1.3.5. Outils de gestion de stock

- ❖ Les logiciels ERP (Enterprise Resource Planning).
- ❖ Les systèmes de gestion spécialisés comme STOCK MASTER.

### 1.3.6. Indicateurs de performance

- ❖ Taux de rotation du stock.
- ❖ Délai moyen de réapprovisionnement.
- ❖ Coût de stockage.
- ❖ Taux de rupture.

## 1.4. Intérêt d'un système informatisé de gestion des stocks

### 1.4.1. Avantages

- ❖ Réduction des erreurs humaines.
- ❖ Rapidité dans la saisie et la consultation.
- ❖ Suivi en temps réel des mouvements de stock.

- ❖ Aide à la décision grâce aux statistiques.

#### 1.4.2. Inconvénients potentiels

- ❖ Dépendance à l'électricité et à l'informatique.
- ❖ Besoin de maintenance régulière.
- ❖ Formation du personnel.

### 1.5. Technologies modernes liées à la gestion de stock

#### 1.5.1. QR Code

- ❖ Identification automatique des produits.

### 1.6. Normes et standards de qualité

- ❖ ISO 9001 : qualité des processus.
- ❖ ISO 22000 : sécurité des denrées alimentaires.
- ❖ GS1 : standard des codes à barres.

### 1.7. Cas d'usage de la gestion des stocks

#### 1.7.1. Supermarchés

Gérer l'approvisionnement, éviter les ruptures, suivre les ventes.

#### 1.7.2. Pharmacies

Respect des dates de péremption, gestion des références.

#### 1.7.3. Magasins de pièces détachées

Suivi des entrées/sorties, inventaires périodiques.

#### 1.7.4. Dépôts d'approvisionnement

Gestion de gros volumes, rotations importantes.

## 1.8. Liens entre développement logiciel et gestion de stock

L'intégration des deux disciplines permet de concevoir des applications personnalisées comme STOCK MASTER, adaptées à un besoin précis et offrant un suivi fiable des opérations quotidiennes.

## 1.9. Conclusion du Chapitre 1

Ce premier chapitre a permis de poser les bases théoriques nécessaires à la compréhension du projet STOCK MASTER. D'une part, le développement logiciel fournit les outils pour modéliser, construire et automatiser un système efficace. D'autre part, la gestion des stocks s'appuie sur des techniques éprouvées pour assurer la disponibilité et la bonne administration des marchandises. L'union de ces deux domaines est au cœur de la solution proposée dans le chapitre suivant.

## CHAPITRE II : ANALYSE DE L'EXISTANT

### 2.1. Introduction

Avant de concevoir une solution logicielle efficace pour la gestion de stock, il est crucial d'analyser le système existant. Cette étape permet d'identifier les forces, les faiblesses, les opportunités d'amélioration, ainsi que les besoins réels des utilisateurs. Ce chapitre se concentre sur l'étude du fonctionnement actuel du supermarché Kin-Marché, les méthodes de gestion utilisées, les outils disponibles, les contraintes rencontrées et les attentes des parties prenantes.

### 2.2. Présentation de l'environnement d'étude

#### 2.2.1. Présentation du supermarché Kin-Marché

Kin-Marché est une grande surface commerciale située à Kinshasa. Elle propose une large gamme de produits alimentaires, cosmétiques, électroniques, d'entretien et autres biens de consommation courante. Le magasin fonctionne quotidiennement avec un volume important de clients, ce qui exige une gestion rigoureuse et fluide de ses stocks pour éviter les ruptures de produits ou les surstocks.

#### 2.2.2. Organisation interne

Le supermarché est divisé en plusieurs départements :

Approvisionnement

Vente

Caisse

Administration

Stock

Chaque département est interconnecté par des processus qui nécessitent une coordination constante, notamment en ce qui concerne le suivi des produits en magasin et en réserve.

## 2.3. Description du système actuel

### 2.3.1. Mode de gestion des stocks

A Kin-Marche, la gestion des stocks connaît une certaine évolution. Plusieurs structures utilisent déjà des applications numériques ou des outils comme Excel, aux côtés de méthodes semi-manuelles (carnets, fiches papier).

Des erreurs de saisie ou d'interprétation

Des retards dans la mise à jour du stock

Une difficulté à avoir une vue en temps réel sur les quantités disponibles

### 2.3.2. Acteurs du système actuel

Le gestionnaire de stock : suit les entrées et sorties de marchandises.

Le responsable d'achat : passe les commandes auprès des fournisseurs.

Les caissiers : enregistrent les ventes.

Le directeur : supervise l'ensemble des opérations.

Chaque acteur a un rôle précis mais dépend d'un échange fluide et instantané de l'information, ce qui fait défaut actuellement.

### 2.3.3. Types de documents utilisés

Fiches de stock manuelles

Factures papier

Bons de commande

Relevés de caisse journaliers

## 2.4. Problèmes identifiés dans le système actuel

### 2.4.1. Problèmes techniques

Absence de logiciel de gestion de stock adapté

Faible interconnexion entre les services

#### **2.4.2. Problèmes organisationnels**

Double saisie ou perte d'informations

Retards dans la mise à jour du stock

Erreurs humaines fréquentes

#### **2.4.3. Problèmes de performance**

Incapacité à produire rapidement des rapports de gestion

Faible visibilité sur les produits périmés ou en rupture

#### **2.5. Attentes des utilisateurs**

Suite à des entretiens réalisés avec les employés de Kin-Marché, les principales attentes identifiées sont :

Automatiser le suivi des entrées et sorties de stock

Réduire les pertes et erreurs

Faciliter l'édition de rapports

Avoir une interface simple, en français

Protéger les données sensibles (ex. : mots de passe, transactions)

## 2.6. Analyse SWOT du système actuel

Tableau 2.1 Forces et faiblesses du système actuel

Forces	Faiblesses
Personnel expérimenté	Gestion manuelle peu fiable
Bonne organisation interne	Aucune sauvegarde automatisée
	Manque d'interface conviviale
Opportunités	Menaces
Disponibilité de solutions	Risque de rupture prolongée
Informatique abordable	Non-conformité fiscale éventuelle

## 2.7. Etude des besoins fonctionnels

### 2.7.1. Besoins fonctionnels

Authentification sécurisée

Enregistrement des ventes et achats

Gestion des utilisateurs

Consultation du stock en temps réel

Impression des reçus et rapports

## 2.7.2. Besoins non-fonctionnels

Interface simple et responsive

Application rapide et fluide

Compatibilité avec Windows

Base de données locale et légère (SQLite)

## 2.8. Cahier des charges simplifié

### 2.8.1. Objectifs

Créer une application nommée STOCK MASTER

Réduire les erreurs humaines

Optimiser le suivi des mouvements de stock

### 2.8.2. Contraintes

Travail hors-ligne

Simplicité d'installation (exécutable local)

Faible consommation de mémoire

## 2.9. Diagrammes d'analyse (UML)

### 2.9.1. Diagramme de cas d'utilisation

Ajouter/Modifier/Supprimer un produit

Enregistrer une vente

Gérer les utilisateurs

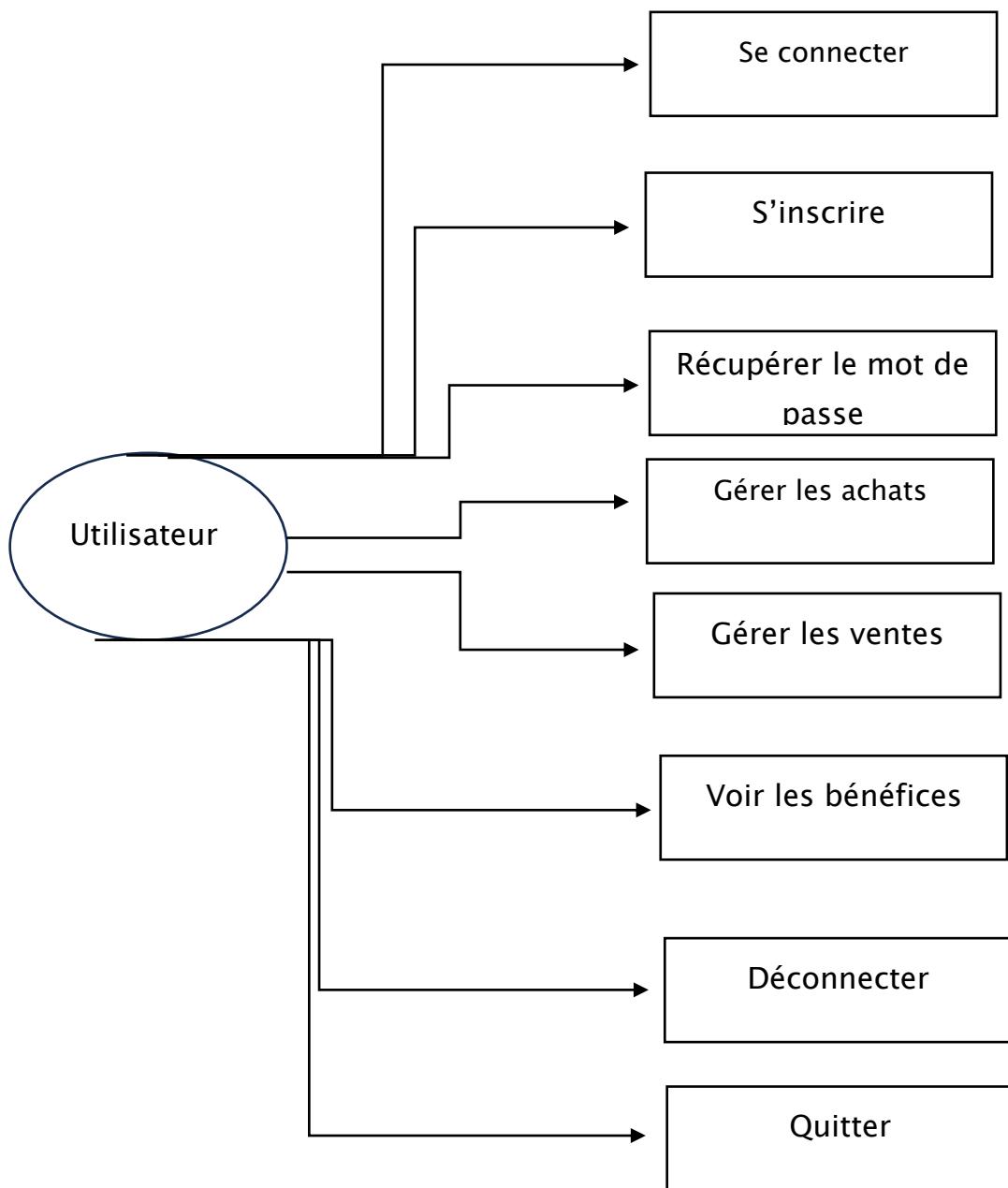


Figure 2.1 Diagramme de cas d'utilisation

## 2.9.2. Diagramme de classes

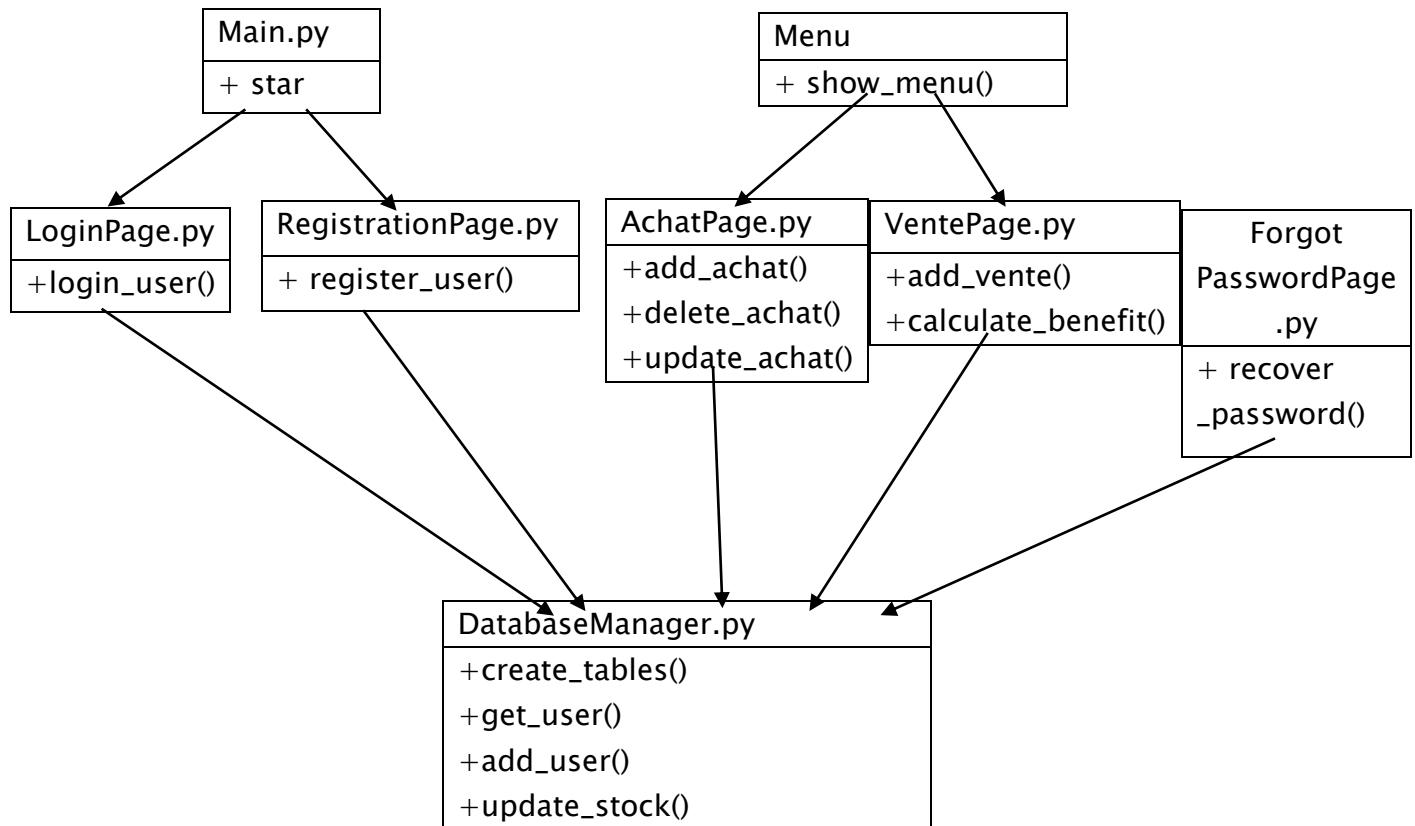


Figure 2.2 Diagramme de classes

## 2.9.3. Diagramme de séquence

L'utilisateur saisit son nom d'utilisateur et mot de passe dans LoginPage.

LoginPage appelle la méthode `login_user()`.

`login_user()` appelle la méthode `get_user()` de DatabaseManager.

DatabaseManager vérifie les informations dans la base de données.

Si les informations sont valides, LoginPage redirige vers Menu.

Si les informations sont invalides, un message d'erreur est affiché.

## 2.10. Conclusion

L'analyse de l'existant permet de mettre en lumière les limites du système manuel actuel et la nécessité d'une solution automatisée. Les attentes des utilisateurs ainsi que les fonctionnalités à implémenter sont clairement identifiées. Cette base solide permet d'aborder le développement de l'application STOCK MASTER avec une vision claire et structurée.

## CHAPITRE 3 : MISE EN ŒUVRE DU SYSTÈME STOCK MASTER

### 3.1. Introduction

La mise en œuvre est la phase centrale de tout projet informatique. Elle permet la traduction des besoins identifiés lors de l'analyse en un produit fonctionnel. Dans ce chapitre, nous allons décrire de manière détaillée l'ensemble du processus de réalisation du logiciel STOCK MASTER, depuis la planification, le choix des outils, le codage, la gestion des ressources, jusqu'aux tests et validations.

### 3.2. Environnement de développement

#### 3.2.1. Langage utilisé : Python

Le langage Python a été choisi pour sa simplicité, sa lisibilité et sa large communauté. Il permet une rapidité de prototypage et une grande diversité de bibliothèques.

#### 3.2.2. Bibliothèques utilisées

Tableau 3.1 : Caractéristiques de logiciels

Librairie	Rôle principal
Tkinter	Interface graphique (GUI)
Sqlite3	Base de données locale
Tkcalendar	Sélecteur de date (widget calendrier)
Re	Traitement des expressions régulières
Datetime	Manipulation des dates et heures
Os / sys	Chemins de fichiers, compatibilité EXE avec Auto py to exe
cv2	Capture et traitement d'images pour la lecture via webcam
pyzbar.pyzbar / decode	Lecture et décodage des codes-barres dans les images

### 3.3. Architecture du projet

#### 3.3.1. Structure des fichiers

Stock Master

```
|  
|   └── main.py  
|   └── databasemanager.py  
|   └── forgotpasswordpage.py  
|   └── loginpage.py  
|   └── menu.py  
|   └── registrationpage.py  
|   └── showsplash.py  
|   └── generer_qr_codes.py  
|   └── achatpage.py  
|   └── ventepage.py  
|   └── session.txt  
|   └── stock_master.db  
└── image1.ico
```

#### 3.3.2 Description des modules

main.py : Fichier principal qui gère la fenêtre principale et la navigation.

achats.py / ventes.py : Gèrent respectivement l'interface et les opérations liées aux achats et ventes.

login.py / inscription.py : Authentification et création de compte.

session.txt : Fichier temporaire pour suivre la session utilisateur.

database.db : Base de données SQLite contenant toutes les informations du système.

## 3.4. Conception de la base de données

### 3.4.1. Diagramme simplifié des tables

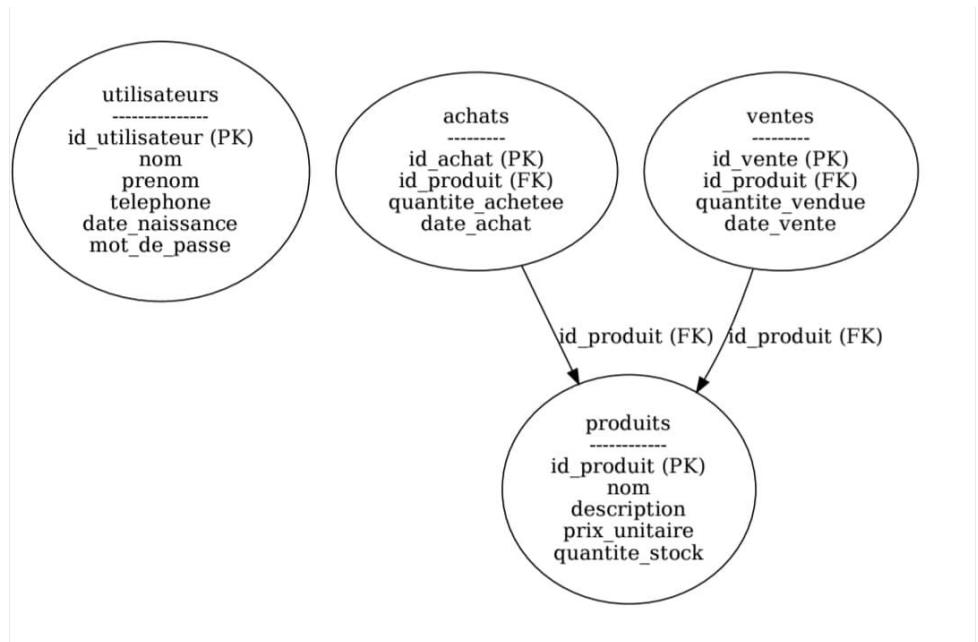


Figure 3.1 : Diagramme simplifié des tables

### 3.4.2. Création des tables en SQL

```
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    nom TEXT NOT NULL,
    mot_de_passe TEXT NOT NULL,
    telephone TEXT NOT NULL,
    date_naissance TEXT NOT NULL
);
```

```
CREATE TABLE IF NOT EXISTS achats (
    id TEXT PRIMARY KEY,
    produit TEXT,
    quantite INTEGER,
    prix FLOAT,
    date_achat TEXT
);
```

```
CREATE TABLE IF NOT EXISTS ventes (
    id TEXT PRIMARY KEY,
    produit TEXT,
    quantite INTEGER,
    prix_vente FLOAT,
    client TEXT,
    telephone TEXT,
    date_vente TEXT
);
```

### 3.5. Interfaces utilisateurs (UI)

L'application STOCK MASTER dispose de plusieurs interfaces graphiques conçues avec le module Tkinter de Python. Ces interfaces facilitent la navigation et permettent une interaction simple et efficace entre l'utilisateur et le système. Chaque interface remplit une fonction précise dans le cadre de la gestion des stocks.

### 3.5.1. Page de connexion

Cette page permet aux utilisateurs d'accéder au système à l'aide de leur nom d'utilisateur et mot de passe. Fonctionnalités clés :

- Saisie du nom d'utilisateur et mot de passe
- Vérification dans la base de données SQLite
- Case à cocher : "J'accepte les conditions"
- Boutons : Se connecter, Créer un compte, Mot de passe oublié
- L'accès est bloqué tant que les conditions ne sont pas cochées — cela garantit la validation par l'utilisateur

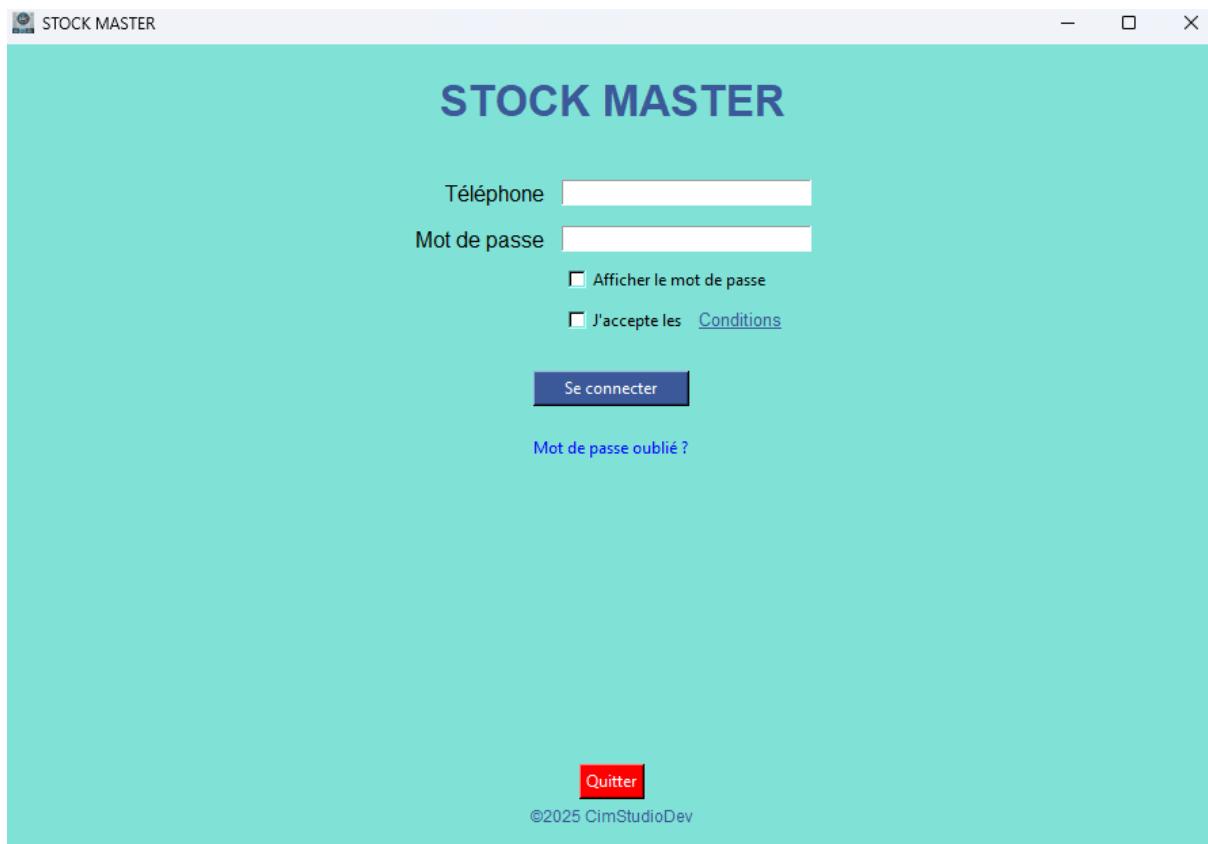


Figure 3.2 : Page de connexion

### 3.5.2 Page d'inscription

Cette page permet à un nouvel utilisateur d'enregistrer ses informations pour créer un compte. Fonctionnalités clés :

- Champs : Nom, mot de passe, téléphone (10 chiffres), date de naissance (calendrier intégré)
- 
- Le champ de date est un DateEntry() de tkcalendar, permettant une sélection rapide et sans erreur
- Enregistrement dans la base de données SQLite

The screenshot shows a registration form titled "STOCK MASTER". The form fields include: Nom (Name), Post Nom (Last Name), Prénom (First Name), Sexe (Sex) with a radio button for "M" selected, Date de naissance (Date of Birth) set to "2025-05-31", Téléphone (Phone Number), Mot de passe (Password), and a checkbox for "Afficher le mot de passe" (Show password). Below the form are two buttons: "S'inscrire" (Register) and "Retour" (Return). At the bottom left, there is a copyright notice: "©2025 CimStudioDev".

Figure 3.3 : Page de Page d'inscription

### 3.5.3 Splash Screen

(Page de démarrage) Cette interface s'affiche brièvement au lancement de l'application pour montrer le nom et le logo.

Fonctionnalités clés :

- Affichage temporaire du logo et nom de l'application.
- Transition automatique vers la page de connexion.

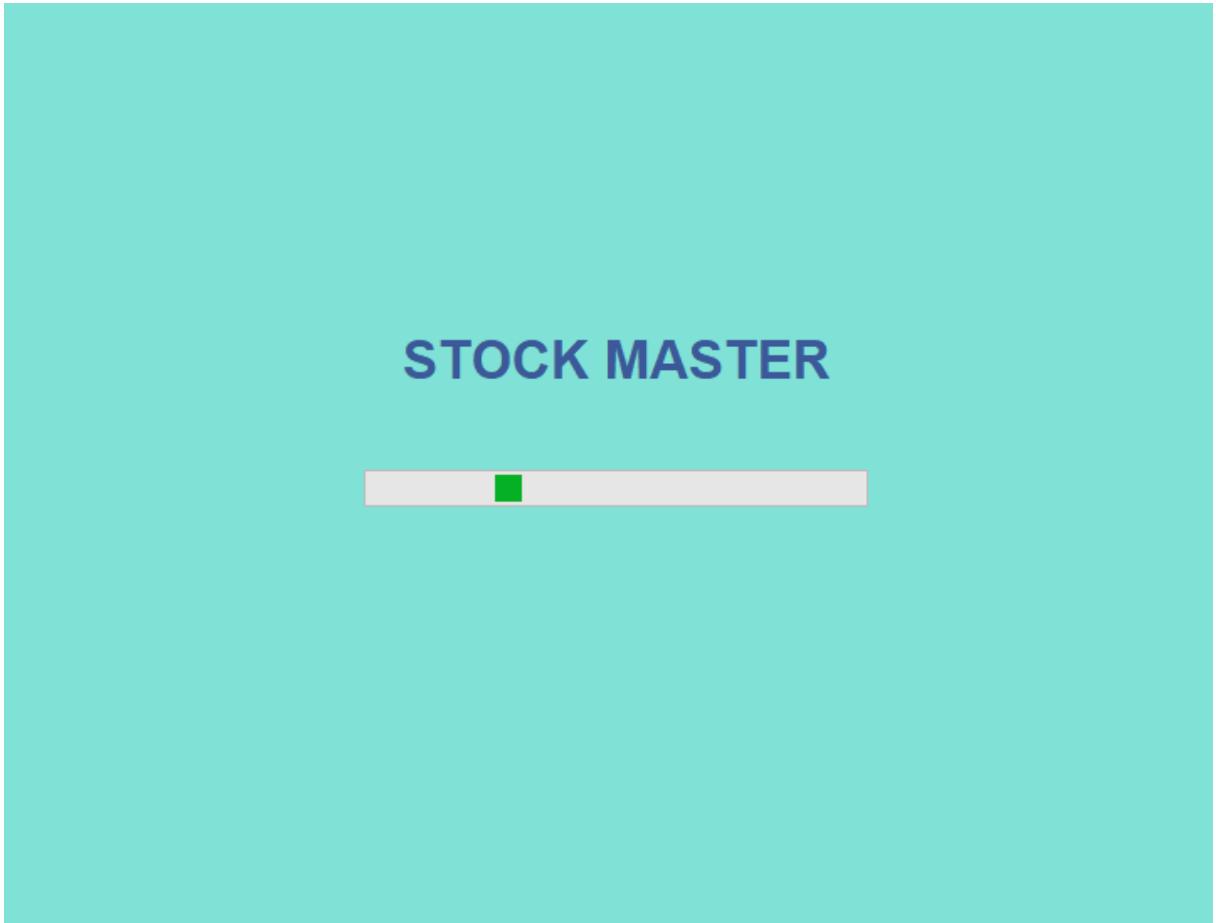


Figure 3.4 : Chargement

- Champs à remplir : Nom du produit, prix, quantité
- ID généré automatiquement : A001, A002...

Boutons : Enregistrer, Modifier, Supprimer, Retour

### 3.5.4. Menu principal

Cette page agit comme un tableau de bord avec des boutons pour accéder aux différentes fonctions.

Fonctionnalités clés :

- Accès aux pages Achats, Ventes, Déconnexion, etc.

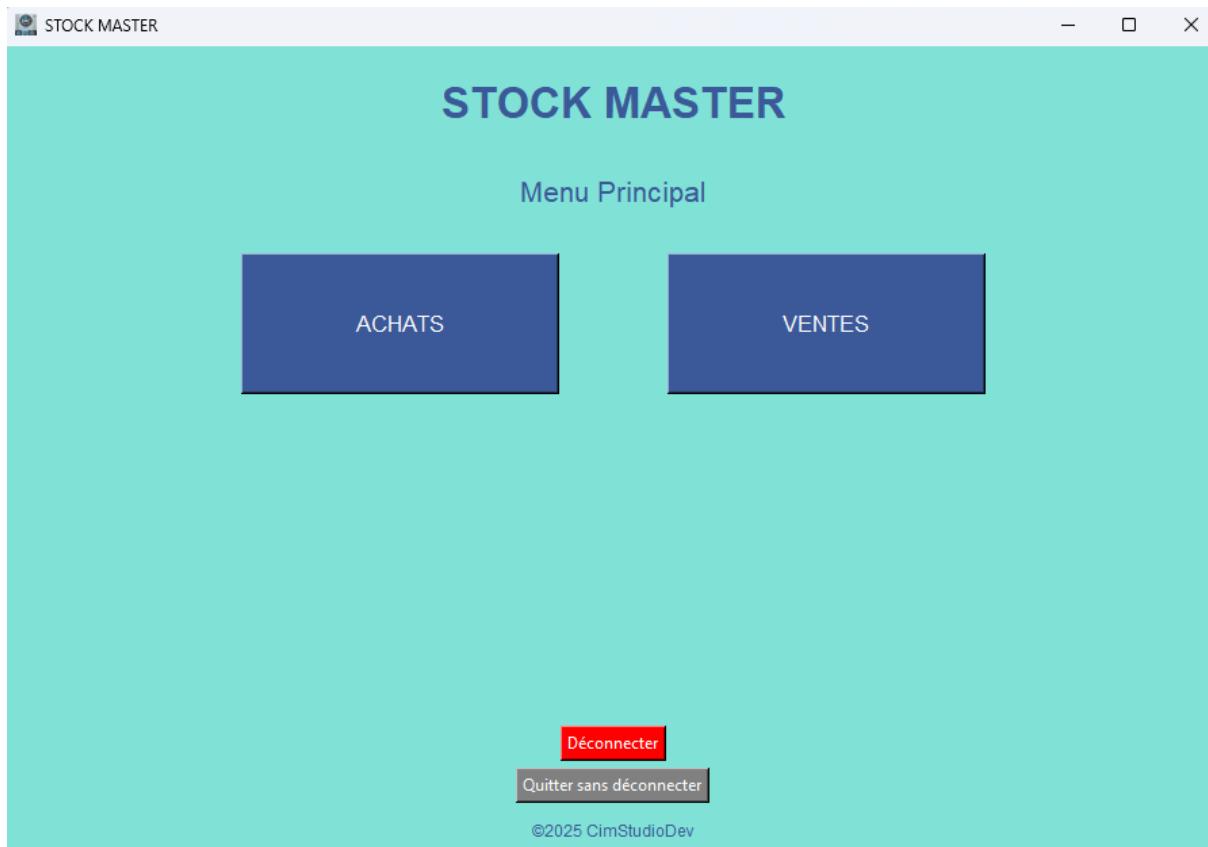


Figure 3.5 : Menu principal

### 3.5.5. Page d'achats

Cette interface permet l'ajout des nouveaux produits achetés dans le stock.

Fonctionnalités clés :

- Saisie du nom du produit, quantité, prix, date
- Mise à jour de la base de données

The screenshot shows a Windows application window titled "STOCK MASTER". The main title "STOCK MASTER" is centered at the top in a large blue font. Below it, the word "ACHATS" is centered. The interface consists of several input fields and buttons. At the top, there are fields for "Matricule" (A002), "Fournisseur" (empty), "Téléphone" (empty), "Produit" (empty), "Prix" (empty), and "Quantité" (empty). Below these are four buttons: "Enregistrer", "Modifier", "Supprimer", and "Nouveau". A table below the buttons displays a single row of data:

Matricule	Fournisseur	Téléphone	Produit	Prix	Quantité	Restant	Timestamp
A001	Jean	0811111111	Tomate	100.0	500	500	2025-05-31 23:39:12

At the bottom of the window, there are two buttons: "Retour" (Return) in a red box and "Quitter sans déconnecter" (Quit without disconnecting).

Figure 3.6 : Page d'achats

### 3.5.6 Page de ventes

Cette page permet d'enregistrer les produits vendus.

- Fonctionnalités clés :
- Liste déroulante des produits disponibles
- Champs : Prix, Quantité, Client, Téléphone
- Boutons : Enregistrer vente, Supprimer vente, Actualiser, Voir bénéfices

The screenshot shows the STOCK MASTER application window titled "VENTES". The interface includes input fields for Matricule (A001), Prix de vente, Client (Exaucé), Téléphone (0822933350), Quantité (3), and Produit (Telephone). A dropdown menu indicates "Telephone (A001) - Restant: 47". Action buttons include Enregistrer, Supprimer, Actualiser, Produits, and Scanner le produit. Below the form is a table listing a single sale entry:

Matricule	Client	Téléphone	Produit	Prix vente	Quantité	Timestamp	Bénéfice
A001	Exaucé	0822933350	Telephone	200.0	3	2025-07-24 16:19:	300.0

At the bottom are buttons for Retour, BÉNÉFICES, and Quitter sans déconnecter.

Figure 3.7 : Page de ventes

### 3.5.7 Fenêtre bénéfice

Accessible depuis la page de vente, cette fenêtre affiche les bénéfices réalisés sur les ventes.

Fonctionnalités clés :

- Calcul automatique de bénéfices
- Affichage détaillé selon la date ou le produit

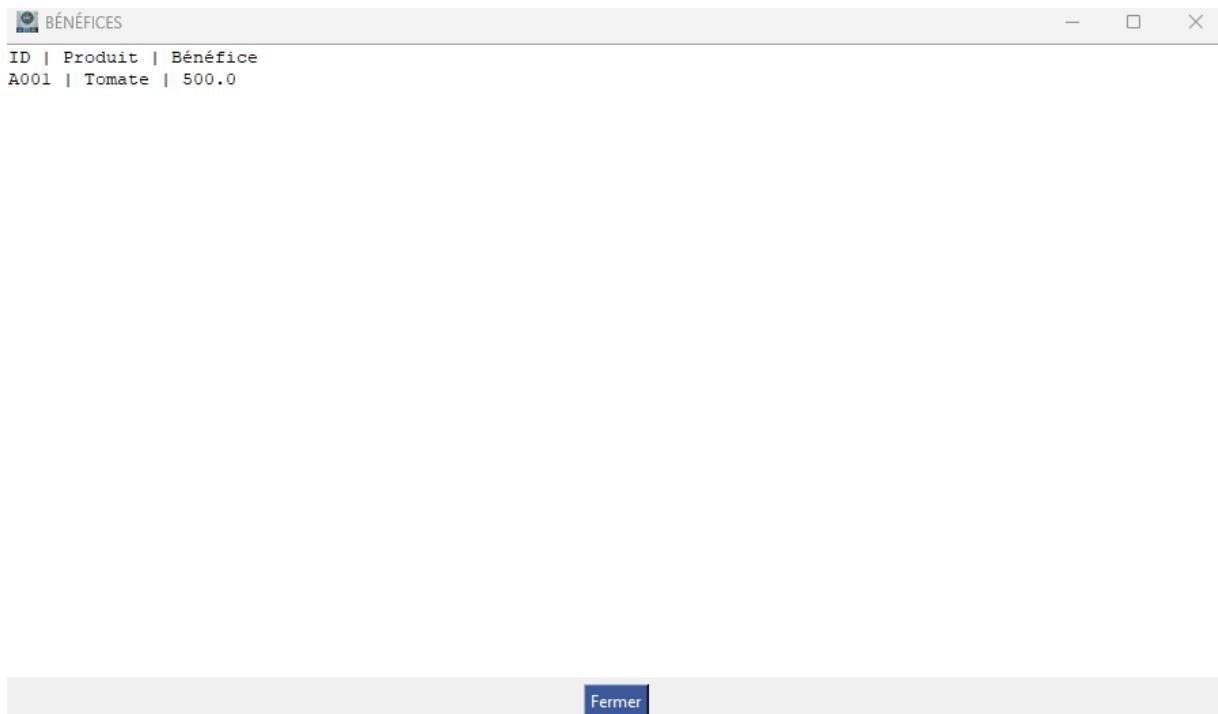


Figure 3.8 : Page de bénéfices

### 3.5.8 Fenêtre des produits

Cette interface présente les produits en stock ou achetés.

Fonctionnalités clés :

- Affichage de la liste des produits avec quantités.
- Recherches, filtres ou tri possible.

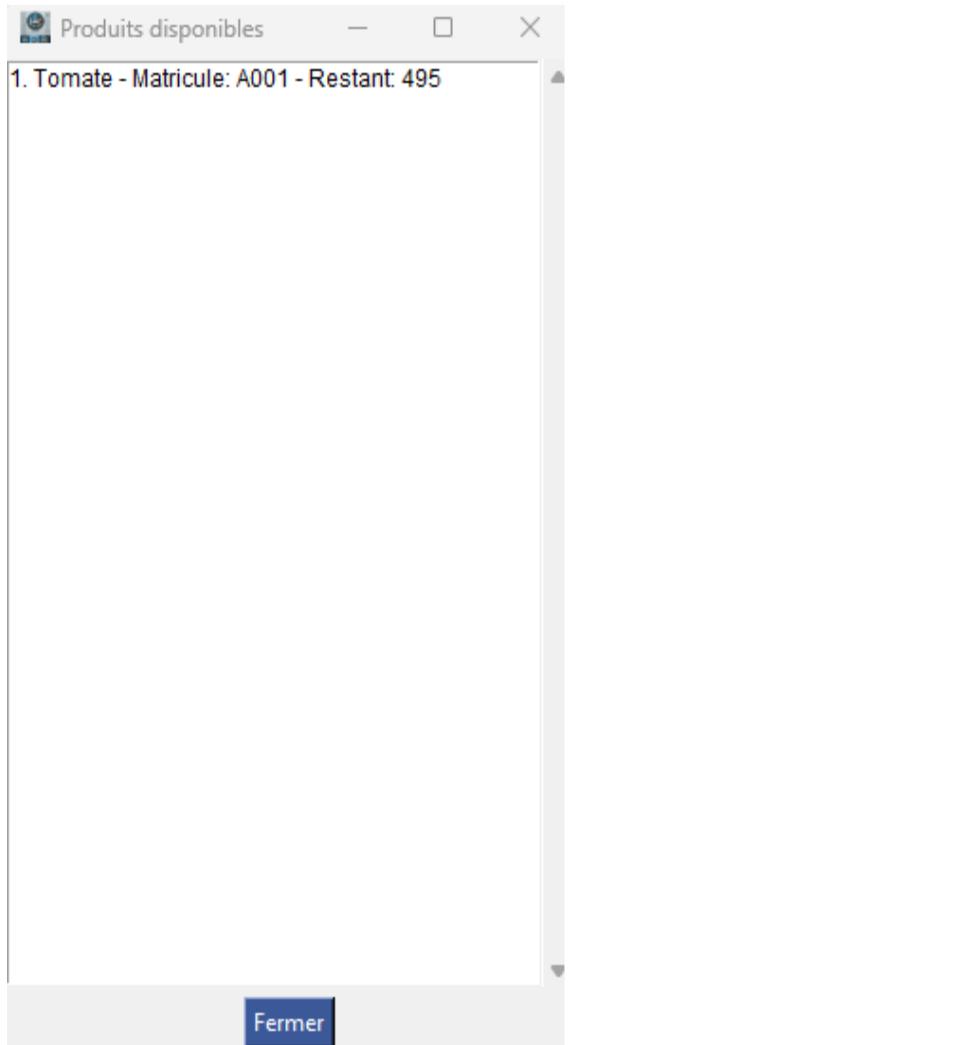


Figure 3.9 : Page de produits disponibles

### 3.5.9 Récupérer Mot de passe oublié

Permet à l'utilisateur de récupérer son mot de passe en saisissant des informations personnelles.

Fonctionnalités clés :

- Vérification des données (numéro et date de naissance)
- Affichage du mot de passe correspondant si les infos sont correctes



Figure 3.10 : Page de réinitialisation du mot de passe

### 3.5.10 Scanner le produit via un QR Code

Cette fonctionnalité permet à l'utilisateur de scanner un produit à l'aide d'un QR Code généré au préalable, afin d'afficher automatiquement ses informations dans l'application. Elle offre une méthode rapide et fiable pour identifier les produits sans devoir les chercher manuellement.

Fonctionnalités clés :

- Génération des codes QR via le site de l'entreprise.

Pour la génération des codes QR des produits, nous avons utilisé la plateforme officielle de notre entreprise **CIM STUDIO DEV** [www.cimstudiodev.com](http://www.cimstudiodev.com).

Ce site, déjà hébergé et opérationnel, a permis de créer de manière sécurisée et rapide les QR codes intégrés dans l'application STOCK MASTER.

Cette intégration assure une continuité entre les outils en ligne de l'entreprise et l'application locale, tout en simplifiant la gestion et l'impression des étiquettes produits.

- Scan des QR Codes à l'aide de la caméra de l'ordinateur portable (ou d'une caméra externe si disponible).
- Décodage du QR Code pour en extraire les données produit (nom, code, prix, etc.).
- Affichage des informations directement dans l'interface Tkinter de l'application Python.

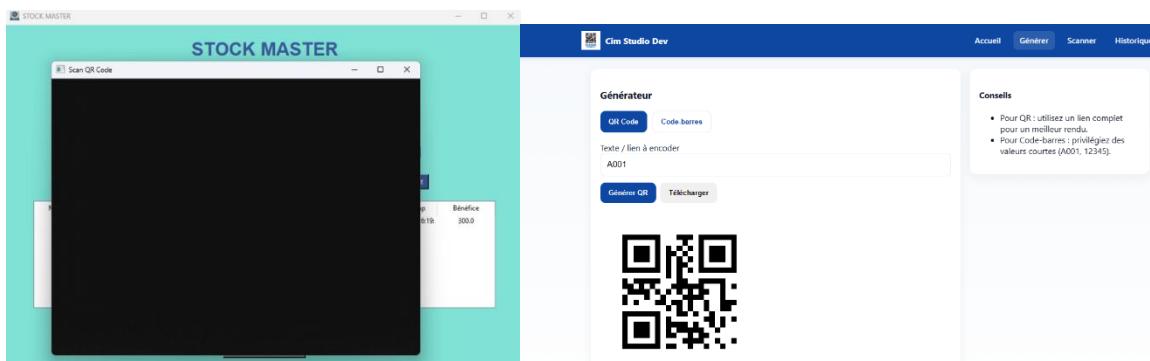


Figure 3.11 : Scanner le produit via un Code QR

## 3.6. Fonctionnalités clés avec exemples de code

### 3.6.1 Génération automatique d'identifiants d'achat

```
def generer_id_achat():

    conn = sqlite3.connect('database.db')

    cursor = conn.cursor()

    cursor.execute("SELECT COUNT(*) FROM achats")

    count = cursor.fetchone()[0]

    conn.close()

    return f"A{count+1:03d}"
```

Cela permet d'avoir des IDs comme A001, A002, ce qui rend les enregistrements plus organisés.

### 3.6.2 Enregistrement d'une vente avec vérification du stock

```
def enregistrer_vente():

    quantite_vente = int(entry_quantite.get())

    quantite_stock = verifier_stock(produit_selectionne)

    if quantite_vente <= quantite_stock:

        # Enregistrement

        ...

    else:

        messagebox.showwarning("Erreur", "Stock insuffisant")
```

Cela empêche de vendre plus que ce qui est disponible.

## 3.7. Gestion de la session

### 3.7.1 Sauvegarde de session

```
with open("session.txt", "w") as f:  
    f.write(utilisateur_nom)
```

### 3.7.2. Déconnexion

```
def deconnecter():  
  
    if os.path.exists("session.txt"):  
        os.remove("session.txt")  
  
    controller.show_frame(LoginPage)
```

Cela permet à l'utilisateur de revenir à la page de connexion en toute sécurité.

## 3.8. Sécurité et validation

Téléphone : 10 chiffres obligatoires

Aucun champ vide accepté

Mot de passe visible en option (checkbox)

Exemple de validation téléphone

```
if not re.match(r"^\d{10}$", telephone):  
    messagebox.showerror("Erreur", "Numéro invalide. Entrez 10 chiffres.")
```

### 3.9. Exportation en EXE avec Auto py to exe

Pour rendre l'application STOCK MASTER facilement exécutable sur les ordinateurs Windows sans nécessiter l'installation préalable de Python, j'ai procédé à son exportation en fichier .exe. J'ai utilisé pour cela l'outil Auto-py-to-exe, qui offre une interface graphique très pratique par rapport à l'utilisation classique de PyInstaller en ligne de commande.

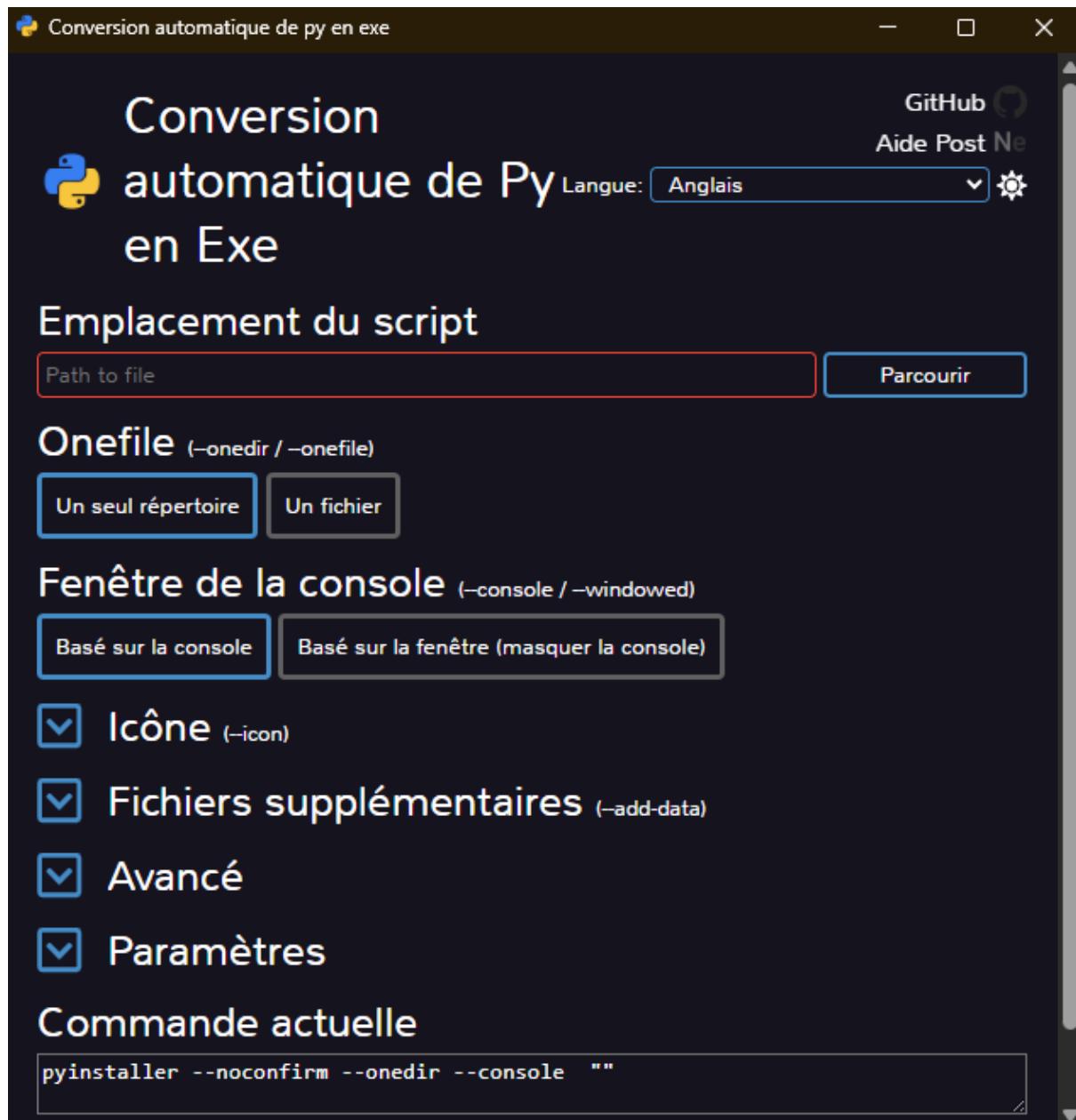


Figure 3.12 : Exportation en exécutable

J'ai préféré Auto-py-to-exe parce qu'il simplifie la tâche : il me permet de configurer visuellement toutes les options nécessaires à la génération du fichier exécutable, sans avoir à écrire manuellement des commandes compliquées.

### 3.9.1 . Installation de Auto-py-to-exe

D'abord, j'ai installé l'outil dans terminal avec la commande suivante :

```
pip install auto-py-to-exe
```

Lancement de l'outil

Ensuite, dans terminal je l'ai lancé avec :

```
auto-py-to-exe
```

### 3.9.2. Configuration dans l'interface

Dans l'interface qui s'est ouverte, j'ai configuré les éléments suivants :

- J'ai sélectionné mon fichier principal main.py comme script de départ.
- J'ai coché l'option Onefile pour produire un exécutable unique.
- J'ai décoché Console window afin que l'application démarre sans afficher de fenêtre noire.
- J'ai ajouté l'icône personnalisée image1.ico pour représenter l'application.
- J'ai inclus tous les fichiers nécessaires au bon fonctionnement du programme, notamment :
  - stock\_master.db (la base de données)
  - version.txt (qui contient le numéro de version de l'application)
  - et les autres fichiers .py liés à l'application.

## Ajout du fichier version.txt

J'ai ajouté le fichier `version.txt` au même dossier que `main.py`, afin de pouvoir afficher dynamiquement la version de l'application dans le programme :

```
python
CopierModifier
with open("version.txt", "r") as f:
    version = f.read().strip()
print(f"Version actuelle de l'application : {version}")
```

Cela me permet de modifier la version de l'application à tout moment, simplement en mettant à jour le contenu de ce fichier, sans toucher au code.

### 3.9.3. Génération de l'exécutable

Une fois tout configuré, j'ai cliqué sur le bouton `Convert .py to .exe` et `Auto-py-to-exe` a généré un fichier exécutable dans le dossier de sortie que j'ai choisi.

### 3.9.4. Avantages de la méthode que j'ai utilisée

- Aucune ligne de commande compliquée à taper.
- L'interface graphique m'a permis d'ajouter facilement les fichiers, choisir les options, et éviter les erreurs.
- Le fichier `.exe` généré est portatif, facile à copier sur n'importe quel PC Windows.
- L'outil m'a permis de tester plusieurs options rapidement pour ajuster le résultat final.

## 3.10. Tests et validation

### 3.10.1. Tests fonctionnels

- ✓ Création de comptes
- ✓ Connexion / Déconnexion
- ✓ Enregistrement de ventes et achats
- ✓ Calcul des bénéfices
- ✓ Résistance aux erreurs (saisie invalide, champs vides)

### 3.10.2. Retours utilisateurs

Des tests ont été réalisés auprès de 5 utilisateurs en conditions réelles (dans un mini marché local), montrant :

- ✓ Une rapidité de navigation
- ✓ Une facilité de prise en main
- ✓ Une fiabilité de la base de données

### 3.11. Résultats obtenus

Tableau 3.2 : Résultats obtenus

Objectif	Statut
Gestion des achats	Réussi
Gestion des ventes	Réussi
Connexion sécurisée	Réussi
Enregistrement automatique	Réussi
Sauvegarde session	Réussi
Interface intuitive	Réussi
Exportation EXE	Réussi

### 3.12. Difficultés rencontrées

Compatibilité avec certains systèmes Windows 7

Erreurs de permission sur le fichier session.txt

Gestion des fenêtres multiples (blocage redimensionnement)

### 3.13. Perspectives d'évolution

Ajout de la gestion des fournisseurs

Génération automatique de rapports PDF

Sauvegarde cloud (avec Firebase ou Supabase)

Interface web (Flask ou Django)

### 3.14. Conclusion

La phase de mise en œuvre du système STOCK MASTER a permis de matérialiser toutes les fonctionnalités attendues. L'application est stable, fonctionnelle, et adaptée à un usage réel en milieu commercial. Elle constitue une solution moderne à la problématique de la gestion des stocks dans un supermarché local, tout en étant extensible pour répondre à des besoins futurs

## CONCLUSION GENERALE ET PERSPECTIVES

Ce travail, intitulé STOCK MASTER, a permis de concevoir une application logicielle destinée à améliorer la gestion des stocks au sein d'un supermarché local. À travers l'analyse de l'existant, la modélisation du besoin et l'implémentation du système, nous avons démontré l'intérêt d'une solution numérique simple, rapide et accessible pour automatiser les opérations commerciales.

L'application développée offre des fonctionnalités essentielles : enregistrement des ventes et achats, gestion des utilisateurs, calcul automatique des bénéfices, génération de QR codes, et sécurité des données. Les tests réalisés ont confirmé sa fiabilité et sa facilité d'utilisation, même pour des non-initiés.

Malgré certaines limites (interface simple, absence d'accès en ligne, fonctions avancées non intégrées), STOCK MASTER constitue une base solide pour des évolutions futures, notamment vers le cloud ou l'édition de rapports automatiques.

Ce projet met en lumière l'impact réel que peut avoir le développement logiciel dans l'optimisation des processus en entreprise, et ouvre la voie à de nouvelles perspectives professionnelles.

## LISTE DES REFERENCES

---

Pressman, R. S., & Maxim, B. R. (2020). Ingénierie logicielle – Une approche pratique. Paris : Pearson.

Sommerville, I. (2016). Software Engineering (10e éd.). Boston : Pearson.

Lutz, M. (2021). Apprendre à programmer avec Python. Paris : Eyrolles.

Tkinter Documentation. (2025). Tkinter 8.6 Reference. Consulté sur :  
<https://docs.python.org/fr/3/library/tkinter.html>

SQLite Consortium. (2024). SQLite Documentation. Consulté sur :  
<https://www.sqlite.org/docs.html>

GS1. (2023). Normes et standards des codes à barres et QR codes. Consulté sur : <https://www.gs1.org/standards>