**Introduction générale**

L

a gestion de stock est un élément essentiel pour toute entreprise visant à maintenir un équilibre entre l'offre et la demande de ses produits.

Un système de gestion de stock permet de suivre les niveaux de stock, de minimiser les coûts et d'optimiser les opérations commerciales.

Ce projet vise à créer une application de gestion de stock en utilisant le langage de programmation Python.

L'objectif principal est de fournir une plateforme où les utilisateurs peuvent effectuer des opérations telles que l'ajout de nouveaux produits, la mise à jour des niveaux de stock, l’achat d’un produit

1. Fonctionnalités clés du Projet :

* Ajout de Produits : Les utilisateurs peuvent ajouter de nouveaux produits à la base de données du système de gestion de stock. Chaque produit sera associé à des détails tels que le nom, la quantité initiale, le prix unitaire, etc.
* Mise à Jour des Niveaux de Stock : Les utilisateurs peuvent mettre à jour les niveaux de stock en fonction des nouvelles entrées, des ventes ou de tout autre ajustement nécessaire.
* Recherche de Produits : Une fonction de recherche permettra aux utilisateurs de trouver rapidement des produits spécifiques dans la base de données, facilitant ainsi la gestion quotidienne.

1. Technologies utilisées :

Ce projet sera développé en utilisant le langage de programmation Python, avec l'utilisation éventuelle de bibliothèques comme Pandas pour la manipulation des données et Flask pour la création d'une interface web conviviale.

En résumé, ce projet de gestion de stock en Python vise à simplifier et à optimiser les processus de gestion de stock, offrant ainsi une solution informatisée efficace pour les entreprises souhaitant maintenir un contrôle précis sur leurs stocks.

**Chapitre 1 : Analyse et conception**

1. Analyse :
2. Les spécifications fonctionnelles

Les spécifications fonctionnelles d'un projet de gestion de stock définissent en détail les fonctionnalités et les capacités que le système doit fournir. Voici quelques spécifications fonctionnelles générales que vous pourriez inclure dans un projet de gestion de stock :

1. Gestion des Produits :

* Ajout de nouveaux produits avec des détails tels que le nom, la description, le prix unitaire, etc.
* Mise à jour des informations des produits.
* Suppression de produits obsolètes ou non pertinents.

1. Gestion des Stocks :

* Suivi des niveaux de stock pour chaque produit.
* Mise à jour automatique des niveaux de stock lors de l'ajout ou de la vente de produits.
* Gestion des alertes pour les niveaux de stock bas.

1. Modification de produit :

* Enregistrement des entrées et sorties de produit.

1. Recherche et Filtrage :

* Recherche de produits par nom, catégorie.

1. Gestion des Utilisateurs :

* Création de comptes utilisateurs avec des niveaux d'accès différents.
* Gestion des autorisations d'accès.

1. Affichage des produits :

* Liste les détails de produit.

1. Gestion d’archive :

* . Possibilité d'archiver des produits qui ne sont plus vendus ou qui sont obsolètes.
* Archivage des mouvements de stock passés pour un suivi historique.

1. Gestion d’inventaire :

* Fonctionnalité permettant la réalisation d'inventaires pour vérifier les niveaux de stock réels par rapport aux enregistrements du système.
* Mise à jour des données en fonction des résultats de l’inventaire.
* Ces spécifications fonctionnelles servent de base pour détailler les fonctionnalités nécessaires à la conception et à la mise en œuvre du système de gestion de stock. Vous pouvez les adapter en fonction des besoins spécifiques de votre projet et de l'entreprise concernée.

2.Les spécifications non fonctionnelles :

Les spécifications non fonctionnelles d'un projet de gestion de stock décrivent les caractéristiques du système qui ne sont pas directement liées à ses fonctionnalités, mais qui sont tout aussi importantes pour assurer sa performance, sa sécurité, son évolutivité, etc. Voici quelques spécifications non fonctionnelles que vous pourriez inclure dans votre projet de gestion de stock en Python :

1. Performance :

* Temps de réponse rapide pour les requêtes utilisateur.
* Capacité à gérer efficacement un grand volume de données.

1. Sécurité :

* L’application doit respecter la confidentialité des données .

1. Fiabilité :

* Disponibilité élevée du système pour assurer un accès constant aux utilisateurs.
* Gestion des erreurs et des pannes de manière à minimiser les perturbations.

1. Évolutivité :

* Capacité du système à s'adapter à une augmentation du nombre d'utilisateurs ou de produits.
* Facilité d'ajout de nouvelles fonctionnalités ou de modifications futures.

1. Interopérabilité :

* Capacité à interagir avec d'autres systèmes ou logiciels existants.
* Conformité aux normes et protocoles de l'industrie.

1. Convivialité de l'Interface Utilisateur :

* Interface utilisateur intuitive pour faciliter l'interaction.
* Prise en charge de différentes tailles d'écrans et dispositifs.

1. Maintenabilité :

* Facilité de maintenance et de gestion du code source.
* Documentation claire pour les développeurs et les administrateurs.

1. Performances en Termes de Chargement :

* Temps de démarrage rapide de l'application.
* Temps nécessaire pour charger les pages et les données.

1. Coûts :

* Coûts de développement, de maintenance et d'exploitation raisonnables.
* Économie d'utilisation des ressources matérielles.
* Ces spécifications non fonctionnelles contribuent à garantir que le système de gestion de stock est robuste, sécurisé, performant et conforme aux normes de qualité. Elles sont tout aussi importantes que les spécifications fonctionnelles pour assurer le succès global du projet.

1. Conception :
2. Unified Modeling Language
3. Définition :

UML est un langage de modélisation visuelle utilisé dans le domaine du génie logiciel pour concevoir et documenter les systèmes logiciels. Il offre un ensemble de notations graphiques standardisées permettant aux développeurs, aux architectes logiciels et aux analystes de visualiser, spécifier, construire et documenter les aspects d'un système logiciel.

1. Choix de la méthodologie de conception et justification :

Dans ce projet, nous avons préfère concevoir avec le langage de modélisation UML vu qu’UML représente un standard pour les langages de modélisation objet. UML permet de modéliser de manière claire et précise la structure et le comportement d’un système indépendamment de toute méthode ou de tout langage de programmation. Il n’impose pas des méthodes particulières et peut donc être intégré a n’importe quel processus de développement du logiciel de manière transparente. UML encourage l’utilisation d’outils et se caractérise par une grande précision.

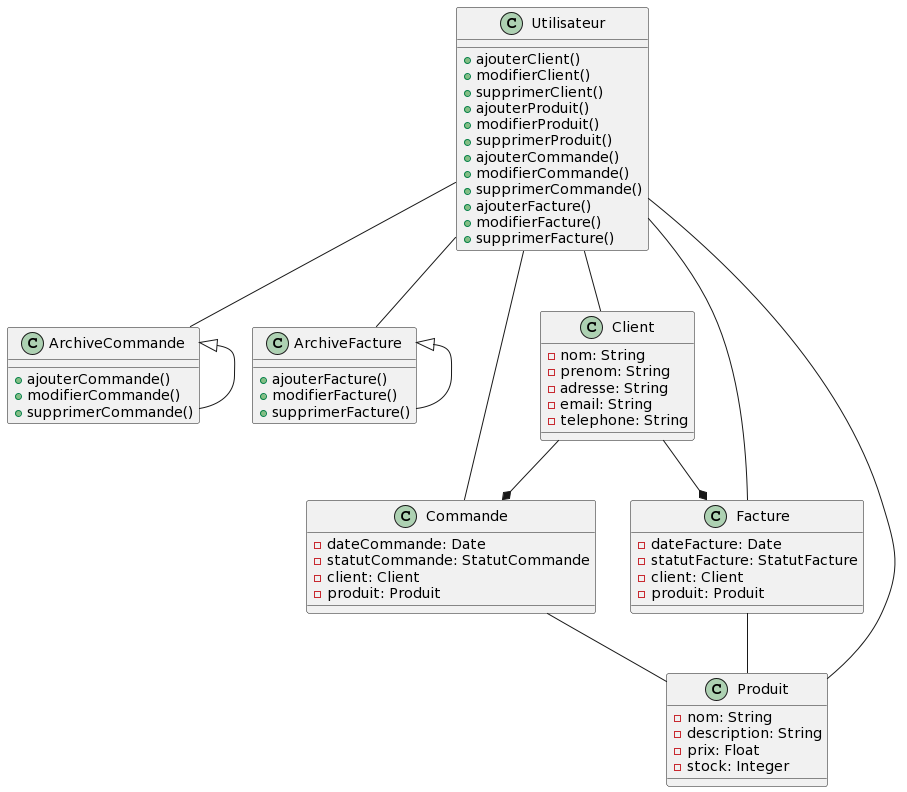


Figure 1 : Diagramme de .

**Chapitre 2 : Réalisation**

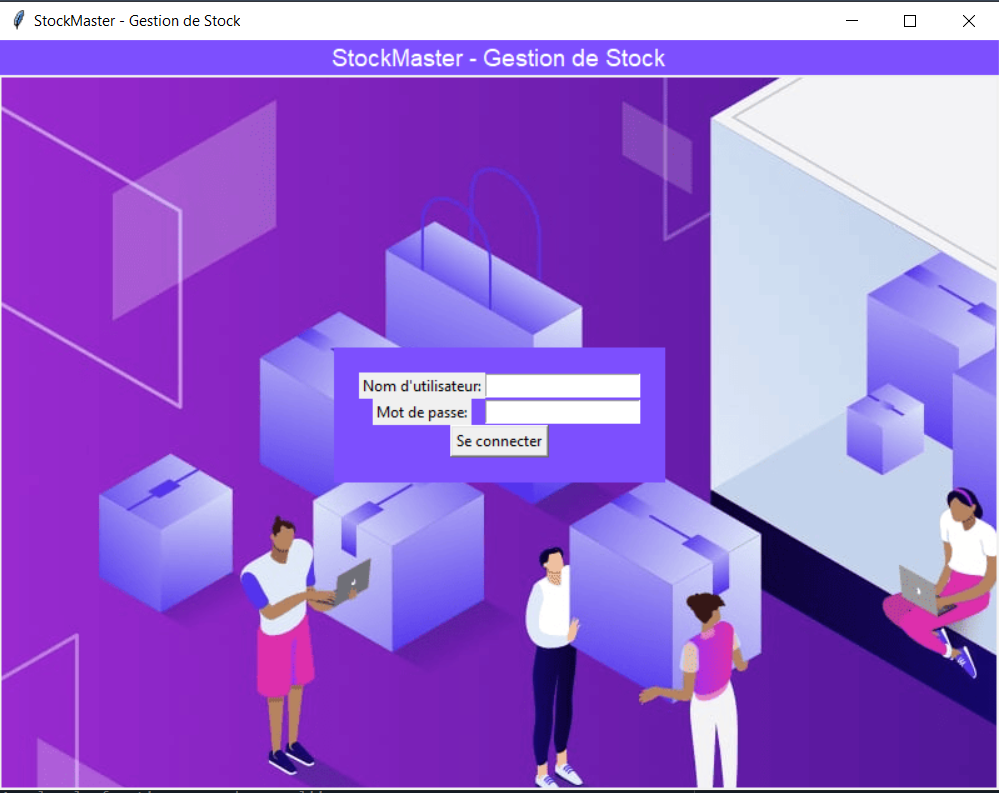
****

Figure 2 :

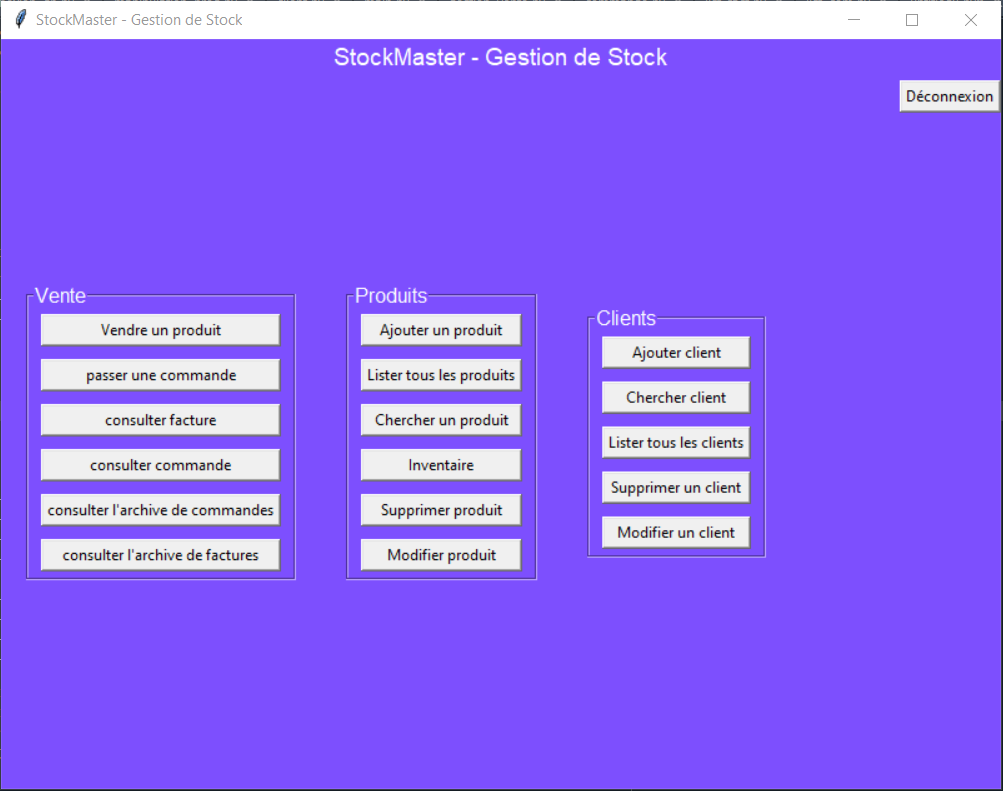
****

Figure 3 :

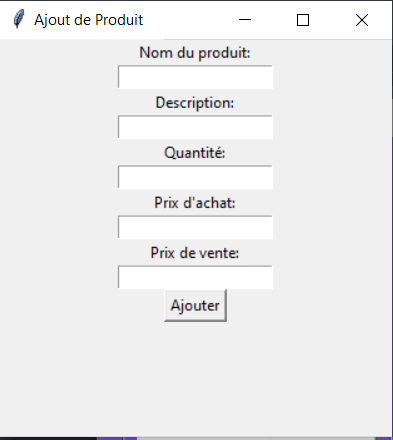
****

Figure 4 :

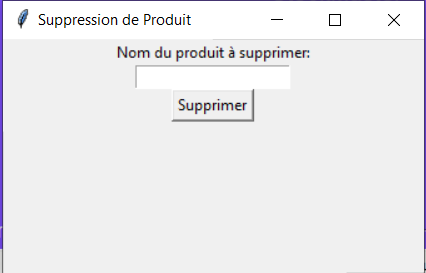
****

Figure5 :

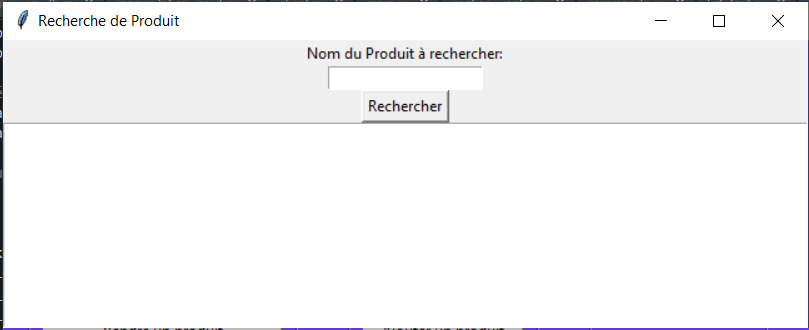
****

Figure 6 :

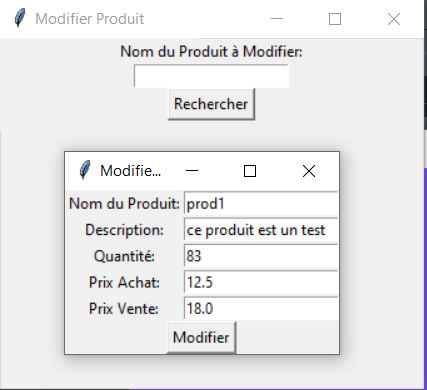
****

Figure 7 :

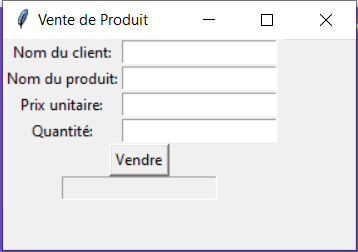
****

Figure 8 :

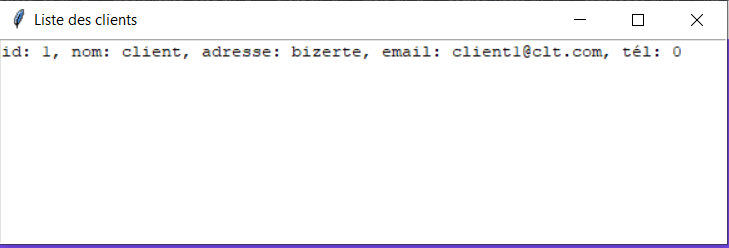
****

Figure 9 :

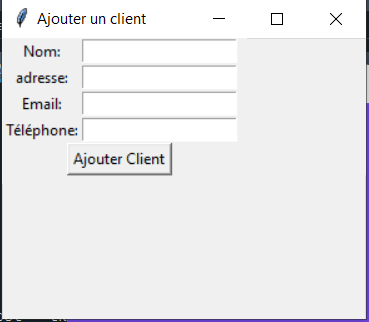
****

Figure 10 :

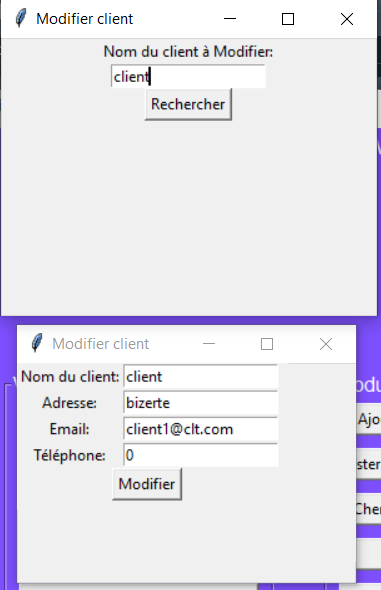
****

Figure 11 :

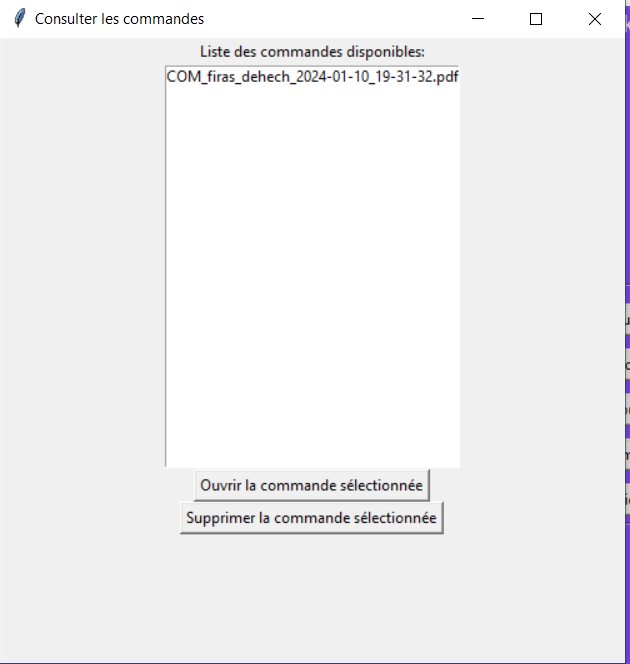
****

Figure 12 :

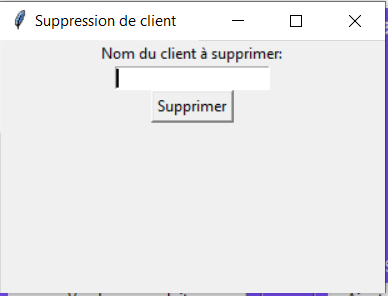
****

Figure 13 :

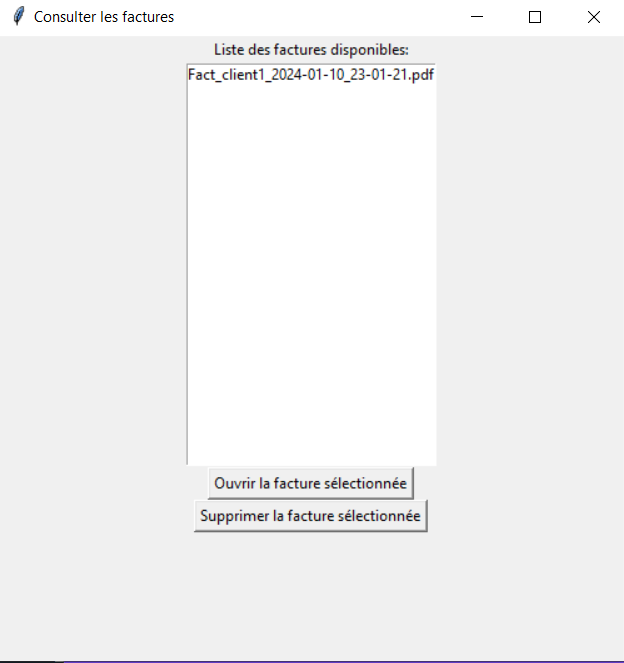
****

Figure 14 :

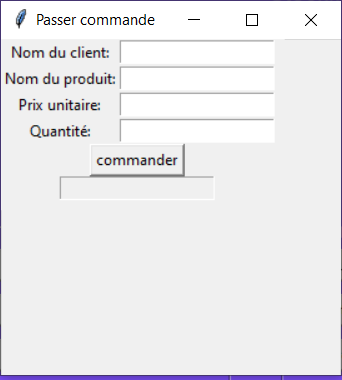
****

Figure 15 :

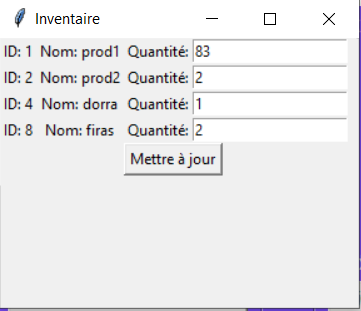
****

Figure 16 :

**Conclusion et perspectives**

1. Conclusion :

En conclusion, le projet de gestion de stock en Python vise à fournir une solution efficace et automatisée pour répondre aux besoins complexes de gestion des stocks des entreprises. L'application proposée offre une gamme complète de fonctionnalités, de la gestion des produits à la génération de rapports, en passant par la traçabilité des mouvements de stock. La conception du système prend en compte des principes tels que la convivialité, la sécurité des données et la modularité pour garantir une expérience utilisateur optimale et une évolutivité future.

1. Perspectives

Bien que le projet réponde aux besoins actuels, plusieurs perspectives d'amélioration et d'expansion peuvent être envisagées pour l'avenir :

* Intégration avec d'autres Systèmes : Explorer des possibilités d'intégration avec d'autres systèmes d'entreprise tels que la comptabilité, la gestion de la chaîne d'approvisionnement.
* Développement d'une Interface Mobile : Création d'une application mobile pour permettre un accès aux fonctionnalités clés depuis des appareils mobiles.
* Intégration de gestion de banque , gestion des engins , gestion de caisse et gestion des fournisseurs

En mettant en œuvre ces perspectives d'amélioration, le projet peut évoluer pour rester pertinent et répondre aux besoins changeants des entreprises tout en restant à la pointe des technologies émergentes.

**Sommaire**

Introduction Général

**Chapitre 1 : Analyse et conception**

1. Analyse
2. Les spécifications fonctionnelles
3. Gestion des Produits
4. Des Stocks
5. Modification de produit
6. Recherche et Filtrage
7. Gestion des Utilisateurs
8. Affichage des produits
9. Gestion d’archive
10. Gestion d’inventaire
11. Les spécifications non fonctionnelles
12. Performance
13. Sécurité
14. Fiabilité
15. Évolutivité
16. Interopérabilité
17. Convivialité de l'Interface Utilisateur
18. Maintenabilité
19. Performances en Termes de Chargement
20. Coûts
21. Conception
22. Unified Modeling Language
23. Définition
24. Choix de la méthodologie de conception et justification

**Chapitre 2 : Réalisation**

Conclusions et perspectives

Liste des figures

Figure 1 :

Figure 2 :

Figure 3 :

Figure 4 :

Figure 5 :

Figure 6 :

Figure 7 :

Figure 8 :

Figure 9 :

Figure 10 :

Figure 11 :

Figure 12 :

Figure 13 :

Figure 14 :

Figure 15 :

Figure 16 :