ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR

Membre de HONORIS UNITED UNIVERSITIES

L'École Marocaine des Sciences de l'Ingénieur

Projet de Fin d'Année

Développement d'un site web de gestion de stock

Réalisé par :
MOSTAFI Saad, NOUA Med Saad

Encadrant :
DAOUDI Imane

Année universitaire : 2024–2025

Dedicace

À ma mère et à mon père, Aucune dédicace ne saurait véritablement traduire tout le respect, l'amour profond et la reconnaissance que je vous porte. Vos sacrifices inestimables pour mon éducation et mon bien-être ont été le socle de mon parcours. Merci pour votre soutien inconditionnel et votre affection constante depuis mon enfance. Puisse ce modeste travail représenter l'accomplissement de vos espoirs, le fruit de vos efforts silencieux et de vos prières répétées. Je sais que je ne pourrai jamais vous rendre tout ce que vous m'avez donné. Que Dieu, le Très-Haut, vous accorde une longue vie, la santé et le bonheur, et qu'Il m'aide à toujours être digne de vous.

À mes frères, Merci pour votre présence fidèle, les instants partagés, les épreuves traversées et les encouragements qui m'ont porté tout au long de ce parcours. Votre soutien a été une force essentielle à chaque étape.

À mes amis, Pour leur amitié sincère, leur écoute et leur présence rassurante dans les moments de doute comme dans les moments de joie.

À mes professeurs, Pour leur encadrement, leur savoir transmis avec passion, et leur engagement dans ma formation.

À tous ceux que j'aime, Merci d'avoir été, chacun à votre manière, une source de motivation et d'inspiration.

Remerciements

Je souhaite exprimer ma profonde gratitude à toutes les personnes qui ont, de près ou de loin, contribué à la réalisation de ce projet de fin d'année.

Avant tout, je rends grâce à Dieu Tout-Puissant, qui m'a accordé la force, la patience et la persévérance nécessaires pour mener à bien ce travail.

Je tiens à remercier chaleureusement mon encadrant pédagogique, [Nom de l'encadrant], pour ses conseils éclairés, sa disponibilité constante et son accompagnement précieux tout au long de ce projet.

Mes remerciements s'adressent également à l'ensemble du corps professoral de l'EMSI, pour la qualité de l'enseignement dispensé et leur engagement tout au long de ma formation.

Je n'oublie pas ma famille, dont le soutien indéfectible, l'amour et la confiance ont été pour moi une source permanente de motivation.

Enfin, je remercie sincèrement mes amis et collègues pour leur aide, leur soutien moral et les échanges enrichissants qui ont jalonné cette belle aventure.

Table des matières

Chapitre 1

Contexte général du projet

1.1 Introduction

Dans un contexte économique en constante évolution, la gestion efficace des stocks constitue un enjeu stratégique pour toute entreprise commerciale. Le suivi rigoureux des produits, des fournisseurs, des mouvements de stock (entrées et sorties), ainsi que la surveillance des niveaux de stock sont essentiels pour éviter les ruptures, limiter les surstocks, et optimiser la chaîne d'approvisionnement.

Ce rapport présente le développement d'une application web dédiée à la gestion de stock, réalisée dans le cadre d'un projet académique. Ce projet s'inscrit dans une démarche pédagogique visant à mettre en pratique les connaissances acquises en analyse et conception logicielle, en particulier à travers l'utilisation des techniques UML (Unified Modeling Language) et du framework Django, reconnu pour sa robustesse et sa modularité dans le développement d'applications web.

L'objectif principal de ce projet est de proposer une solution simple, intuitive et sécurisée permettant :

La gestion des produits et de leurs catégories,

L'enregistrement des fournisseurs,

Le suivi des mouvements de stock (entrées/sorties),

L'affichage d'indicateurs clés via un tableau de bord,

Et la sécurisation de l'accès par un système d'authentification. Ce travail met l'accent sur une approche orientée objet, en s'appuyant sur les diagrammes UML tels que le diagramme de cas d'utilisation, le diagramme de classes, le diagramme de séquence et le diagramme d'activités. Ces outils ont permis de modéliser les exigences fonctionnelles et techniques de l'application, et de structurer efficacement son architecture logicielle.

1.2 Contexte du projet

De nombreuses entreprises, en particulier les petites et moyennes entreprises (PME), font face à des difficultés croissantes dans la gestion efficace de leur inventaire. Ces difficultés peuvent se traduire par plusieurs problèmes opérationnels et stratégiques :

Ruptures de stock fréquentes qui empêchent de satisfaire la demande client à temps,

Surstockage engendrant des coûts de stockage inutiles et un risque de péremption ou d'obsolescence des produits,

Manque de visibilité en temps réel sur les niveaux de stock, les mouvements d'entrées et de sorties, ou encore sur les fournisseurs,

Traçabilité insuffisante, rendant difficile la remontée d'informations précises en cas d'erreurs, de retours produits ou de contrôles qualité.

Dans bien des cas, ces problèmes découlent de l'utilisation de méthodes manuelles ou d'outils non adaptés comme des fichiers Excel ou des enregistrements papier. Ces solutions, bien que simples, deviennent rapidement inefficaces dès que le volume des données augmente ou que l'entreprise souhaite améliorer ses performances logistiques.

C'est dans cette optique qu'émerge le besoin de numériser et automatiser la gestion des stocks à l'aide d'une application logicielle dédiée. Une telle solution permet non seulement de centraliser les informations liées aux produits, fournisseurs et mouvements de stock, mais aussi de fiabiliser les opérations, de réduire les erreurs humaines, et d'optimiser la prise de décision grâce à des tableaux de bord dynamiques.

Le recours à une application web moderne, accessible via un simple navigateur, offre également l'avantage de faciliter le travail collaboratif, notamment lorsque plusieurs utilisateurs (magasiniers, responsables des achats, administrateurs) doivent intervenir sur les mêmes données.

Ainsi, la mise en place d'une application de gestion de stock permet à l'entreprise de franchir un cap vers une gestion plus intelligente, réactive et performante, tout en posant les bases d'une stratégie numérique plus large.

1.3 Périmètre du projet

Le présent projet a pour ambition de proposer une solution logicielle complète et moderne dédiée à la gestion des stocks au sein d'une entreprise commerciale. Il vise à automatiser les tâches récurrentes, à réduire les erreurs humaines, et à centraliser l'ensemble des informations relatives aux produits, aux mouvements de stock, aux fournisseurs et aux catégories dans une interface unique, claire et sécurisée.

En s'appuyant sur les technologies web et les bonnes pratiques de conception orientée objet, l'application permet une supervision rigoureuse, en temps réel, des ressources physiques de l'entreprise. Elle offre aux administrateurs autorisés une interface intuitive et réactive, adaptée aux besoins opérationnels quotidiens.

- Ajout, modification et suppression de (produits,avec des informations détaillées quantité, prix, catégorie, fournisseur)
- Gestion des mouvements de stock,(incluant l'enregistrement des entrées et des sorties de produits, avec traçabilité et historisation)
- Consultation et analyse de l'état du stock disponible (avec affichage des quantités actuelles, alertes sur seuils critiques, et statistiques globales via un tableau de bord)
- Gestion des fournisseurs (permettant d'enregistrer, de consulter et de mettre à jour les informations liées aux partenaires commerciaux)
- Gestion des catégories de produits (pour organiser l'inventaire de manière structurée et faciliter la recherche et la classification)

1.4 Problématique

De nombreuses entreprises rencontrent encore des difficultés dans la gestion quotidienne de leurs stocks, souvent à cause de méthodes peu adaptées comme les fichiers Excel ou les suivis manuels. Cela entraîne des erreurs fréquentes, des ruptures de stock, du surstockage ou un manque de visibilité sur les produits.

Face à ces enjeux, il devient essentiel de disposer d'un outil numérique capable de suivre les produits en temps réel, de gérer les mouvements de stock, et de centraliser les informations sur les fournisseurs et les catégories, tout en restant facile à utiliser et évolutif selon les besoins.

La question à laquelle ce projet cherche à répondre est donc :

Comment concevoir une application simple, fiable et évolutive qui permette à une entreprise de gérer efficacement ses stocks au quotidien ?

1.5 Objectifs du projet

L'objectif principal de ce projet est de développer une solution logicielle permettant d'améliorer la gestion des stocks au sein d'une entreprise commerciale. Cette application doit répondre aux besoins concrets des utilisateurs en offrant un outil à la fois fonctionnel, intuitif et fiable. Elle vise à automatiser certaines tâches, à améliorer la visibilité sur l'état des stocks, et à limiter les erreurs liées aux manipulations manuelles.

Pour cela, les objectifs suivants ont été identifiés :

- Faciliter la gestion des articles en stock(Offrir une interface simple et efficace pour ajouter, modifier, consulter ou supprimer des produits, avec une gestion claire des catégories et des fournisseurs.)
- Réduire les erreurs humaines liées au suivi manuel(Éliminer la dépendance aux documents papier ou aux tableurs, sources fréquentes d'incohérences, en automatisant les opérations critiques comme les mouvements de stock.)
- Mettre à disposition un tableau de bord clair(Fournir une vue synthétique et en temps réel de l'état du stock, du nombre total de produits, des niveaux critiques, et des alertes à surveiller)
- Assurer la traçabilité des opérations(Enregistrer chaque mouvement de stock (entrée ou sortie) avec la date, le produit concerné, la quantité et l'utilisateur ayant effectué l'action, afin de garantir un historique fiable.)

1.6 Benchmark

Des outils comme *Odoo*, *Sage* ou *StockIt* ont été analysés. Bien qu'efficaces, leur complexité ou leur coût les rendent inadaptés aux petites structures. Le projet proposé cherche à fournir une solution légère et personnalisée.

1.7 Méthodologie de gestion de projet

La méthode adoptée pour la réalisation de ce projet s'inspire du cycle en V, un modèle classique en ingénierie logicielle qui repose sur une approche séquentielle, rigoureuse et structurée. Ce cycle permet d'assurer une traçabilité claire entre chaque étape de développement et les tests associés, afin de garantir la qualité du produit final.

Le processus se décompose selon les étapes suivantes :

Spécification des besoins Cette première phase consiste à recueillir et analyser les besoins fonctionnels et non fonctionnels de l'application. Elle permet de définir précisément les fonctionnalités attendues, les contraintes techniques, ainsi que les profils utilisateurs. Le cahier des charges joue ici un rôle central.

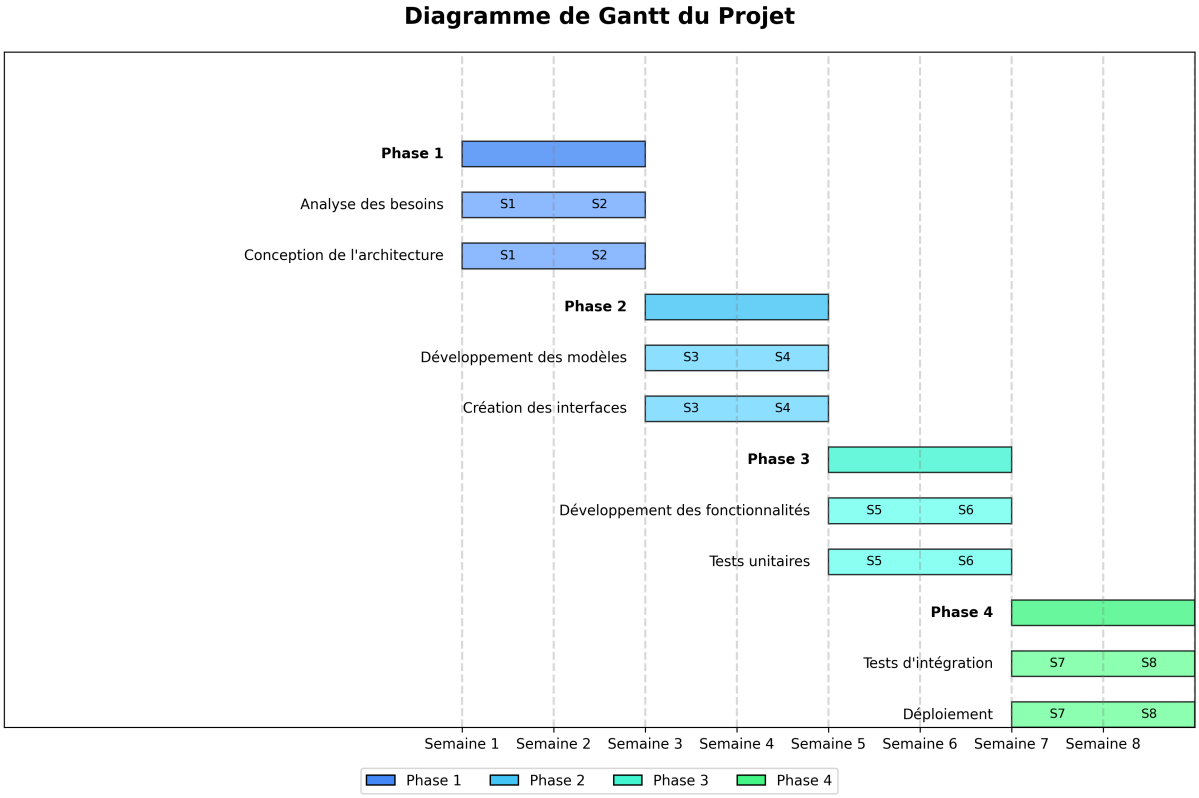
Conception UML Une fois les besoins définis, l'architecture de l'application est conçue à l'aide des diagrammes UML (cas d'utilisation, classes, séquence, activités). Cette phase permet de modéliser la structure du système, les interactions entre les composants, et de poser les bases de l'implémentation.

Réalisation (Développement) C'est la phase de codage proprement dite. À l'aide du framework Django, les modèles de données sont créés, les vues et les interfaces sont développées, et les différentes fonctionnalités sont implémentées. Cette phase respecte les principes du design modulaire et les bonnes pratiques de Django.

Tests unitaires et fonctionnels Des tests sont réalisés pour vérifier la fiabilité de chaque composant (tests unitaires) ainsi que le bon fonctionnement de l'application dans son ensemble (tests fonctionnels). Cette phase vise à identifier d'éventuelles erreurs et à s'assurer que le système répond correctement aux exigences initiales.

Livraison finale Une fois les tests validés, le projet est considéré comme finalisé. L'application peut alors être déployée sur un environnement de production ou hébergée localement pour démonstration. La documentation du projet est fournie, accompagnée d'un guide d'utilisation pour les futurs utilisateurs.

1.8 Diagramme de Gantt



1.9 Conclusion

Ce premier chapitre a permis de poser les bases du projet en présentant le contexte général, la problématique à résoudre, ainsi que les objectifs fonctionnels et la méthodologie utilisée. Il a mis en évidence la nécessité pour les entreprises d'adopter des solutions numériques efficaces pour une gestion de stock moderne, fiable et évolutive.

Le chapitre suivant sera consacré à l'analyse fonctionnelle du système, à travers l'identification des besoins utilisateurs et la définition des cas d'utilisation. Il abordera également la conception du système, en s'appuyant sur les diagrammes UML nécessaires à la modélisation de l'architecture logicielle de l'application.

Chapitre 2

Analyse et conception

2.1 Introduction

La phase d'analyse et de conception constitue une étape cruciale dans le processus de développement logiciel. Elle vise à traduire les besoins exprimés lors de la phase de spécification en modèles formels facilitant la compréhension, la planification et l'implémentation du système.

Cette étape repose sur l'utilisation des diagrammes UML (Unified Modeling Language) afin de modéliser les interactions entre les utilisateurs et le système, la structure des données, ainsi que le comportement des différentes entités. Cette modélisation permet d'anticiper les problématiques techniques, de clarifier les rôles des différents composants, et de garantir la cohérence globale de l'architecture.

2.2 Besoins fonctionnels

Les besoins fonctionnels définissent les actions que le système doit permettre aux utilisateurs d'effectuer. Ils sont directement liés aux fonctionnalités principales de l'application de gestion de stock. Les besoins identifiés sont les suivants :

- Ajouter, modifier, supprimer un produit (L'utilisateur (principalement un administrateur ou un gestionnaire) doit pouvoir gérer les informations des produits : nom, quantité, prix, catégorie, et fournisseur).
- Enregistrer une entrée ou une sortie de stock (Le système doit permettre de tracer les mouvements de stock en enregistrant toute entrée (approvisionnement) ou sortie (vente, transfert) de produits, avec date et quantité).
- Gérer les utilisateurs (Il est nécessaire d'intégrer un système d'authentification afin de gérer l'accès au système. Certains utilisateurs auront des droits restreints, tandis que d'autres (administrateurs) disposeront d'un accès complet aux fonctionnalités).
- Afficher l'état du stock (L'application doit fournir une interface permettant de consulter à tout moment les quantités en stock, de générer des alertes en cas de seuil critique, et de présenter des statistiques via un tableau de bord clair).

2.3 Besoins non fonctionnels

Les besoins non fonctionnels correspondent aux exigences de qualité du système. Bien qu'ils ne décrivent pas directement des fonctionnalités, ils garantissent une utilisation

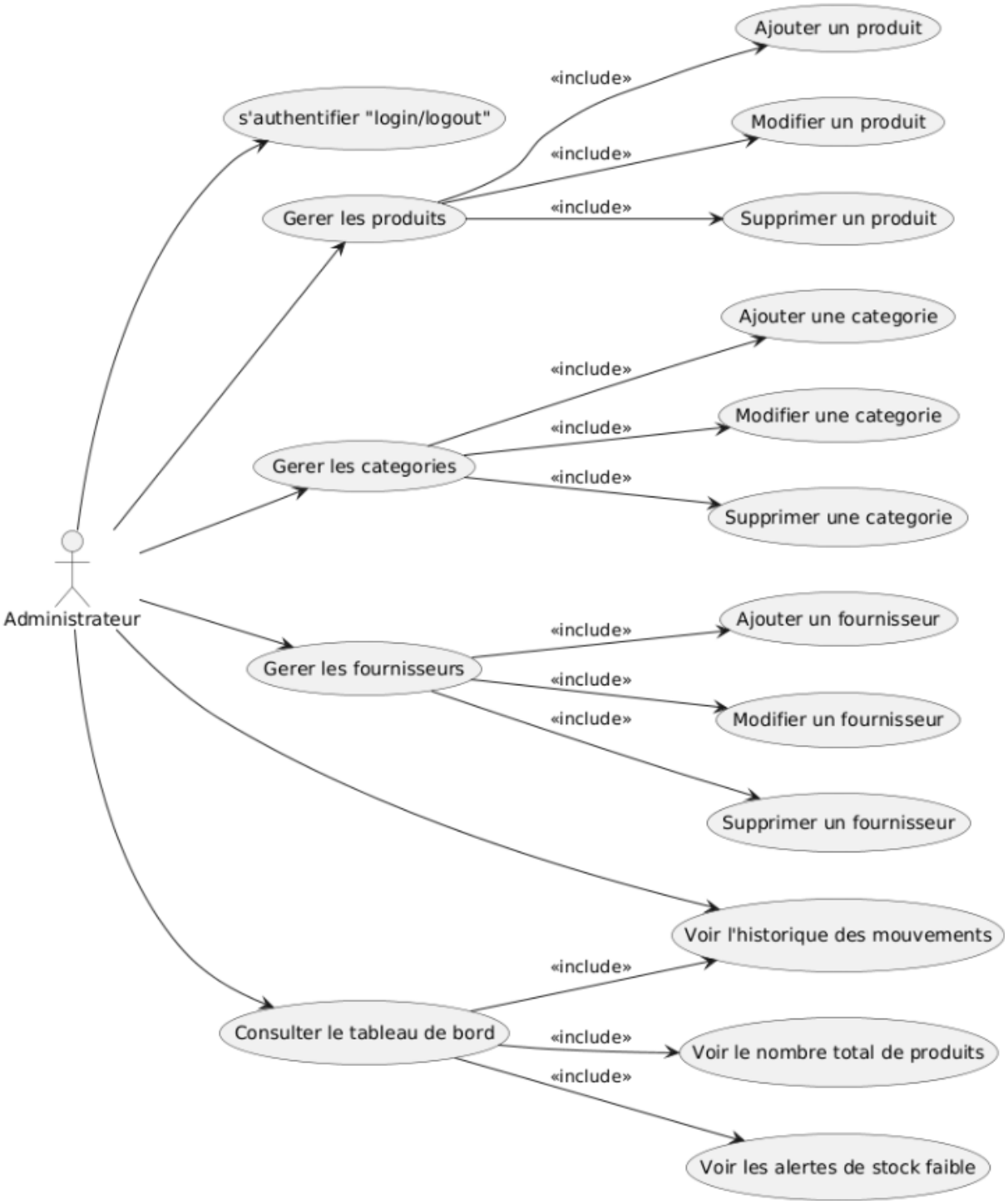
optimale, sécurisée et performante de l'application :

- Interface intuitive (L'interface utilisateur doit être claire, simple à prendre en main, et accessible depuis n'importe quel navigateur. Elle doit utiliser des composants cohérents (formulaires, tableaux, menus) et un design responsive compatible avec différents écrans (PC, tablette...)).
- Temps de réponse inférieur à 2 secondes (L'application doit réagir rapidement aux requêtes des utilisateurs (consultation de produits, enregistrement d'un mouvement, etc.) afin de garantir une bonne expérience utilisateur).
- Sécurité des accès selon les rôles (Le système doit restreindre l'accès à certaines fonctionnalités en fonction du rôle de l'utilisateur (ex. : seuls les administrateurs peuvent ajouter ou supprimer des produits). La gestion des sessions, la protection CSRF et le hashage des mots de passe sont également essentiels).

2.4 Diagramme de cas d'utilisation

Le diagramme montre les interactions principales entre les acteurs (administrateur, magasinier) et les fonctionnalités comme *Gérer article*, *Gérer stock*, etc.

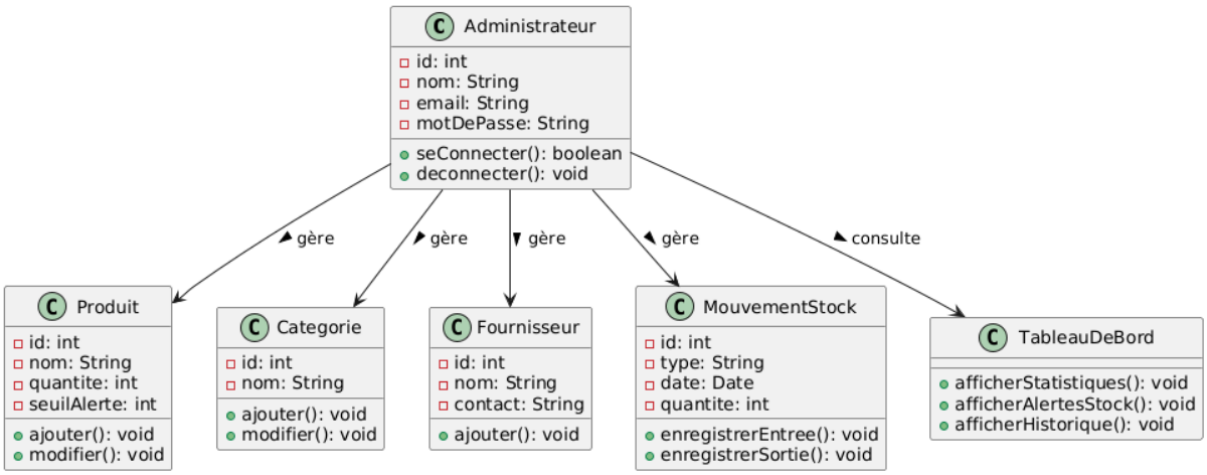
2.4.1 Scénario nominal – Gérer un article



1. L'administrateur s'authentifie.
2. Il sélectionne « Ajouter article ».
3. Il saisit les informations (nom, quantité, prix).
4. Il valide l'enregistrement.
5. L'article est ajouté à la base de données.

2.5 Diagramme de classes

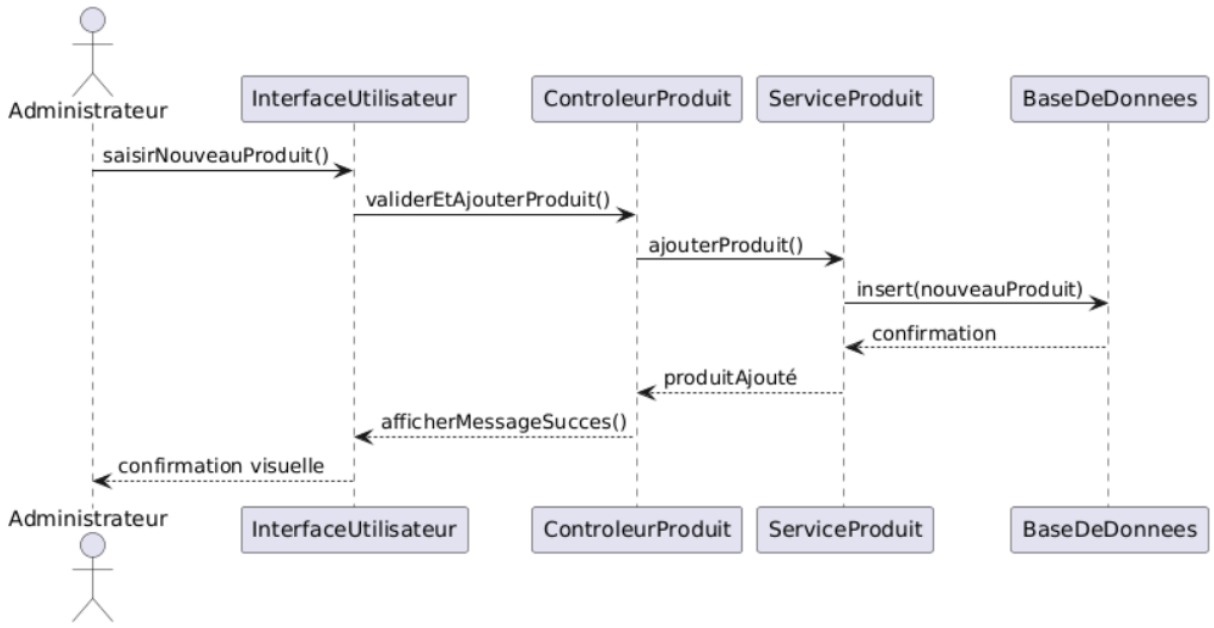
Ce diagramme montre les principales entités : `Article`, `Stock`, `Utilisateur`, `Entrée`, `Sortie` avec leurs attributs et relations (association, héritage).



2.6 Diagrammes de séquence

Les diagrammes illustrent les échanges entre objets (utilisateur, interface, base de données) pour les cas d'utilisation *Gérer stock*, *Ajouter article*, etc.

(Inclure un tableau décrivant pour chaque cas : nom du cas, acteurs impliqués, étapes principales)



2.7 Conclusion

La modélisation UML constitue une étape clé dans la conception d'un système d'information. Elle permet de représenter de manière claire, formelle et structurée l'ensemble des éléments du système : les acteurs, les cas d'utilisation, les entités métier, les interactions, ainsi que les processus dynamiques.

En traduisant les besoins fonctionnels en diagrammes visuels cohérents, UML facilite la compréhension du système par l'ensemble des intervenants du projet (développeurs, encadrants, utilisateurs) et sert de référence solide pour la phase de développement.

Grâce à cette modélisation, il devient possible d'anticiper les problèmes d'organisation du code, de détecter les incohérences ou redondances, et de garantir une architecture logicielle bien structurée. Elle contribue ainsi à réduire les risques d'erreur, à accélérer le développement, et à améliorer la maintenabilité de l'application sur le long terme.

Chapitre 3

Réalisation

3.1 Introduction

Ce chapitre présente la mise en œuvre technique du projet de gestion de stock. À partir des modèles UML définis lors de la phase de conception, les différentes composantes de l'application ont été développées et intégrées selon une approche modulaire. L'objectif est de décrire l'architecture choisie, les outils utilisés, ainsi que les interfaces clés de l'application.

3.2 Architecture de la solution

L'application est construite sur une architecture MVC (Modèle-Vue-Contrôleur), qui favorise la séparation des responsabilités :

Le modèle contient la logique métier et les structures de données (produits, fournisseurs, mouvements...).

La vue représente l'interface utilisateur, développée en HTML/CSS avec Bootstrap.

Le contrôleur correspond aux vues Django (views), qui font le lien entre les données et les pages affichées à l'utilisateur.

Cette organisation permet une meilleure lisibilité du code, une facilité de maintenance et une évolutivité à long terme.

3.3 Outils et technologies utilisés

Le développement de l'application a mobilisé les outils et technologies suivants :

— Langage de programmation : Python Utilisé avec le framework Django pour sa simplicité, sa clarté syntaxique et sa large communauté.

— Base de données : SQLite Choisie pour sa légèreté et sa facilité d'intégration en environnement de développement. Elle peut être migrée vers PostgreSQL si nécessaire.

— Environnement de développement : Visual Studio Code Utilisé pour sa légèreté, ses extensions dédiées à Django et ses outils de débogage intégrés.

— Framework : Django Permet de construire rapidement des applications web robustes, sécurisées et bien structurées.

— Autres :

Bootstrap pour le design responsive des interfaces. Django Admin pour la gestion rapide des données en back-office.

3.4 Interfaces de l'application

Formulaire d'ajout de produit Permet de créer ou modifier un article avec ses informations principales (nom, prix, quantité, catégorie, fournisseur).

Tableau des produits en stock Liste les articles disponibles, leurs quantités, et permet la recherche ou la suppression rapide.

Formulaires de mouvements de stock Permettent d'enregistrer une entrée ou une sortie de stock, avec mise à jour automatique des quantités.

Tableau de bord Donne une vue synthétique du stock global, du nombre total de produits, et des alertes sur les niveaux faibles*L'application propose plusieurs interfaces utilisateur essentielles pour la gestion quotidienne des stocks. Voici un aperçu des principales vues :*

- *Page de connexion Permet aux utilisateurs de s'authentifier avant d'accéder au système.*
- *Formulaire d'ajout de produit Permet de créer ou modifier un article avec ses informations principales (nom, prix, quantité, catégorie, fournisseur).*
- *Tableau des produits en stock Liste les articles disponibles, leurs quantités, et permet la recherche ou la suppression rapide.*
- *Formulaires de mouvements de stock Permettent d'enregistrer une entrée ou une sortie de stock, avec mise à jour automatique des quantités.*
- *Tableau de bord Donne une vue synthétique du stock global, du nombre total de produits, et des alertes sur les niveaux faibles*

3.5 Conclusion

Le développement de l'application s'est appuyé sur une base de conception solide, modélisée en amont à l'aide des outils UML. Grâce à l'utilisation de Django et d'une architecture claire (MVC), l'application est fonctionnelle, stable et conforme aux objectifs définis en phase de spécification.

Elle permet de gérer efficacement les produits, les mouvements de stock, les fournisseurs, et propose une interface fluide et réactive. Cette base technique pourra être enrichie à l'avenir avec des fonctionnalités supplémentaires (gestion multi-utilisateur avancée, génération de rapports, notifications...).

Chapitre 4

Conclusion Générale

Les trois premiers chapitres ont posé les fondations essentielles pour le développement d'une application de gestion de stock moderne, fiable et évolutive. Le premier chapitre a introduit le contexte, la problématique et la méthodologie adoptée, en soulignant l'importance d'une solution numérique adaptée aux besoins des entreprises pour optimiser la gestion des stocks.

Le deuxième chapitre, consacré à l'analyse fonctionnelle et à la conception, a permis de formaliser les besoins utilisateurs à travers une modélisation UML complète. Cette étape a structuré la compréhension du système via les diagrammes de cas d'utilisation, de classes et de séquences, garantissant ainsi une architecture cohérente et facilitant la collaboration entre les différents acteurs du projet.

Enfin, le troisième chapitre a décrit la phase de réalisation technique. S'appuyant sur les modèles définis en amont, le développement avec Django a permis de construire une application modulaire, sécurisée et performante, intégrant les fonctionnalités clés pour gérer les produits, les mouvements de stock et les utilisateurs. L'utilisation d'une architecture MVC claire et d'outils modernes comme Bootstrap assure une interface intuitive et réactive.

Globalement, ce travail structuré et rigoureux, alliant analyse, conception et développement, ouvre la voie à une solution robuste, facilement maintenable et évolutive, capable de répondre efficacement aux défis actuels de gestion de stock en entreprise.