

REPUBLIQUE DU CAMEROUN

REPUBLIC OF CAMERON

PAIX – TRAVAIL – PATRIE

PEACE - WORK - FATHERLAND

MINISTERE DES ENSEIGNEMENTS
SUPERIEURS

MINISTRY OF HIGHER
EDUCATION



PROJET TUTORE

Conception et réalisation d'un logiciel de gestion de stocks des médicaments dans une clinique

Groupe 6

- NGASSAM TCHANANGA Angela Danielle
- NGNIPICBA DJIATOU Carlex

Supervisé par :

M. KEMKENG Aurélien Hermis

Année académique 2025 / 2026

Table des matières

Introduction	3
I. Contexte	3
II. Objectifs	4
III. Utilisateurs cibles	5
IV. Langues ciblées	6
V. Description fonctionnelle	6
VI. Technologies utilisées	11
VII. Structure de la base de données	14
VIII. Planning et réalisation	21
Conclusion	24

Introduction

La gestion efficace des stocks de médicaments constitue un enjeu majeur pour les établissements de santé. Une mauvaise gestion peut entraîner des ruptures de stock critiques, des péremptions coûteuses ou des erreurs de dispensation mettant en danger la santé des patients.

Ce projet vise à concevoir et développer une solution logicielle complète permettant d'automatiser et d'optimiser la gestion des stocks pharmaceutiques au sein d'une clinique. L'application offrira une interface intuitive pour le suivi en temps réel des médicaments, la gestion des approvisionnements, le contrôle des dates de péremption et la génération de rapports d'activité.

Cette solution s'inscrit dans une démarche de modernisation des processus de gestion pharmaceutique et contribuera à améliorer la qualité des soins tout en optimisant les coûts opérationnels de la clinique.

I. Contexte

Les cliniques font face à des défis importants dans la gestion quotidienne de leurs stocks de médicaments. Les méthodes traditionnelles basées sur des registres papier ou des tableurs sont sources d'erreurs, chronophages et ne permettent pas une traçabilité optimale.

Le présent projet répond aux besoins suivants :

- Centralisation des informations sur les médicaments disponibles
- Traçabilité complète des mouvements de stock (entrées, sorties, transferts)
- Alertes automatiques en cas de stock minimum ou de péremption proche
- Génération automatique de rapports pour la prise de décision
- Sécurisation des accès selon les profils utilisateurs
- Conformité avec les exigences réglementaires en matière de gestion pharmaceutique

II. Objectifs

2.1. Objectif général

Développer un système informatisé de gestion de stocks de médicaments permettant d'optimiser les processus d'approvisionnement, de stockage et de dispensation dans une clinique.

2.2. Objectifs spécifiques

- Gestion des médicaments : Créer, modifier et supprimer les fiches produits avec toutes les informations nécessaires (DCI, forme galénique, dosage, prix, etc.)
- Suivi des stocks : Assurer un suivi en temps réel des quantités disponibles avec historique complet des mouvements
- Gestion des approvisionnements : Enregistrer les entrées de stock avec bon de livraison, fournisseur et dates de péremption
- Gestion des sorties : Enregistrer les dispersions avec traçabilité (patient, prescripteur, quantité)
- Système d'alertes : Notifier automatiquement les stocks faibles et les produits proches de la péremption
- Rapports et statistiques : Générer des états de stock, rapports de consommation et analyses de tendances
- Gestion des utilisateurs : Différencier les droits d'accès entre administrateurs et pharmaciens
- Sécurité : Assurer l'authentification, l'autorisation et la traçabilité de toutes les opérations

III. Utilisateurs cibles

3.1. Administrateur

Profil : Responsable informatique, directeur de la clinique ou pharmacien en chef

Responsabilités :

- Gestion complète du système (configuration, paramétrage)
- Création et gestion des comptes utilisateurs
- Attribution des rôles et permissions
- Gestion du catalogue des médicaments (création, modification, suppression)
- Gestion des fournisseurs
- Configuration des seuils d'alerte
- Accès au tableau de bord global avec indicateurs clés de performance
- Génération et consultation de tous les rapports
- Validation des opérations critiques
- Supervision de l'activité de tous les utilisateurs
- Gestion des sauvegardes et de la sécurité du système

3.2. Pharmacien

Profil : Pharmacien d'officine de la clinique

Responsabilités :

- Enregistrement des entrées de stock (réceptions de commandes)
- Enregistrement des sorties de stock (dispensations)
- Consultation du catalogue des médicaments
- Vérification des disponibilités
- Consultation des alertes (stocks faibles, péremptions)
- Génération de rapports de consommation
- Recherche et consultation de l'historique des mouvements

- Inventaire périodique

IV. Langues ciblées

L'application sera développée en français comme langue principale, étant donné le contexte francophone de la clinique.

Éléments concernés :

- Interface utilisateur complète (menus, boutons, formulaires)
- Messages d'erreur et de confirmation
- Rapports générés
- Documentation utilisateur
- Libellés des champs de saisie

Perspectives d'évolution : La possibilité d'ajouter une fonctionnalité multilingue (anglais) pourra être envisagée dans une version ultérieure si nécessaire.

V. Description fonctionnelle

5.1. Module d'authentification et gestion des utilisateurs

Fonctionnalités :

- Connexion sécurisée : Authentification par identifiant et mot de passe
- Gestion des sessions : Maintien de la session utilisateur avec timeout automatique
- Gestion des profils : Création, modification, désactivation des comptes utilisateurs (Administrateur uniquement)
- Attribution des rôles : Définition des permissions selon le profil (Administrateur/Pharmacien)
- Traçabilité : Journalisation de toutes les connexions et actions des utilisateurs

5.2. Module de gestion des médicaments

Fonctionnalités :

- Catalogue des médicaments :
 - Ajout de nouveaux médicaments (Administrateur)
 - Modification des informations (Administrateur)
 - Suppression (Administrateur, avec vérification des dépendances)
 - Recherche avancée (multi-critères : nom, email, D¹CI, catégorie, fournisseur)
 - Consultation des fiches détaillées (tous les utilisateurs)
- Informations enregistrées :
 - Nom commercial
 - Dénomination Commune Internationale (DCI)
 - Forme galénique (comprimé, gélule, sirop, injection, etc.)
 - Dosage
 - Catégorie thérapeutique
 - Fabricant
 - Code-barres
 - Prix d'achat unitaire
 - Prix de vente unitaire
 - Seuil d'alerte (stock minimum)
 - Conditions de stockage
 - Description et indications

¹ **Nom commercial** : choisi par le laboratoire, souvent protégé par une marque.

DCI : universelle, écrite en minuscule, et utilisée pour les médicaments génériques.

Grâce à la DCI, on sait qu'il s'agit toujours de la même molécule, quel que soit le pays ou la marque.

Paracétamol est la DCI.

Ses noms commerciaux varient : **Doliprane®**, **Dafalgan®**, **Efferalgan®**, etc.

5.3. Module de gestion des stocks

Fonctionnalités :

- **Enregistrement des entrées :**
 - Date de réception
 - Numéro de bon de livraison
 - Fournisseur
 - Médicament concerné
 - Quantité reçue
 - Numéro de lot²
 - Date de fabrication
 - Date de péremption
 - Prix d'achat unitaire
- **Enregistrement des sorties :**
 - Date de dispensation
 - Médicament dispensé
 - Quantité
 - Numéro d'ordonnance (si applicable)

² Le numéro de lot sert à identifier une série entière de fabrication d'un médicament, pas chaque boîte individuellement. Toutes les boîtes produites en même temps portent le même numéro, ce qui permet de savoir d'où elles viennent. Grâce à ce numéro, une clinique peut retracer facilement quels médicaments ont été donnés à quels patients. Ainsi, si un lot pose problème, on peut immédiatement retrouver les patients concernés et les prévenir. Lorsqu'un laboratoire fabrique un médicament, il produit une **quantité donnée** (par exemple 100 000 boîtes de paracétamol 500 mg, cette production est regroupée sous un **même numéro de lot**, Toutes les boîtes issues de cette série portent ce numéro, ce qui permet de les identifier comme appartenant à la même fabrication).



- Exemple
 - Une clinique reçoit deux livraisons de **vaccin antigrippal** :
 - Lot **A123** (fabriqué en janvier)
 - Lot **B456** (fabriqué en mars)
 - Si le lot **A123** est rappelé, la clinique sait immédiatement quels patients ont reçu ce lot et peut les informer.

- Patient (nom et/ou identifiant)
 - Prescripteur
 - Pharmacien dispensateur
-
- **Consultation des stocks :**
 - Vue en temps réel des quantités disponibles
 - Historique complet des mouvements
 - Filtrage par médicament, période, type de mouvement
 - État valorisé des stocks

5.4. Module d'alertes et notifications

Fonctionnalités :

- Alertes de stock faible : Notification automatique lorsqu'un médicament atteint le seuil minimum défini
- Alertes de péremption : Notification des produits arrivant à expiration (paramétrable : 3 mois, 6 mois avant)
- Tableau de bord des alertes : Centralisation de toutes les alertes actives
- Accusé de lecture : Traçabilité de la prise en compte des alertes

5.5. Module de gestion des fournisseurs

Fonctionnalités :

- Création de fiches fournisseurs (nom, coordonnées, contact)
- Historique des commandes et livraisons par fournisseur
- Évaluation des fournisseurs (délais, qualité)

5.6. Module de rapports et statistiques

Rapports disponibles :

- État des stocks : Liste complète avec quantités et valeurs
- Rapport de consommation : Analyse par période, par médicament ou par catégorie
- Rapport de péremption : Liste des produits périmés ou à péremption proche
- Rapport de mouvements : Historique détaillé des entrées/sorties
- Rapport de valorisation : Valeur totale du stock
- Rapport d'activité : Statistiques d'utilisation du système
- Tableaux de bord graphiques : Visualisation des tendances de consommation

Formats d'export : PDF, Excel, CSV

5.7. Module inventaire

Fonctionnalités :

- Création de sessions d'inventaire
- Saisie des quantités physiques
- Comparaison avec les quantités théoriques
- Détection et justification des écarts
- Régularisation des stocks après validation
- Historique des inventaires

5.8. Tableau de bord (Dashboard) Administrateur

Indicateurs affichés :

- Nombre total de médicaments en stock
- Valeur totale du stock
- Nombre de médicaments en alerte (stock faible)
- Nombre de médicaments proches de la péremption

- Graphiques de consommation mensuelle
- Top 10 des médicaments les plus dispensés
- Alertes récentes
- Activité récente du système

VI. Technologies utilisées

6.1. Backend (Serveur)

Framework : Django 5.x (Python 3.11+)

Justification : Django est un framework web Python robuste et mature, parfaitement adapté aux applications de gestion. Il offre :

- Un ORM puissant pour la gestion de la base de données
- Un système d'authentification et de permissions intégré
- Une architecture MVC (MVT en Django) claire et maintenable
- Une excellente sécurité par défaut
- Une administration automatique pour le développement

Composants Django utilisés :

- Django REST Framework : pour créer l'API RESTful
- Django CORS Headers : pour gérer les requêtes cross-origin
- Django Filters : pour le filtrage avancé des données
- Celery : pour les tâches asynchrones (alertes, rapports)
- Django Channels (optionnel) : pour les notifications en temps réel

6.2. Frontend (Interface utilisateur)

Framework : React 18+ avec TypeScript

Justification : React est la bibliothèque JavaScript la plus populaire pour créer des interfaces utilisateur modernes et performantes :

- **Flexibilité maximale** : Architecture basée sur des composants réutilisables sans contraintes rigides
- **Courbe d'apprentissage progressive** : Plus facile à maîtriser que les frameworks complets, idéal pour le développement rapide
- **Virtual DOM** : Performances optimales pour les mises à jour d'interface en temps réel
- **Écosystème riche** : Vaste communauté et abondance de bibliothèques tierces de qualité
- **TypeScript intégré** : Typage fort optionnel pour une meilleure maintenabilité du code
- **Popularité** : Framework le plus utilisé au monde (42% des développeurs selon Stack Overflow 2024)

Bibliothèques React :

- **React Router** : Gestion du routage et de la navigation dans l'application
- **Material-UI (MUI) ou Ant Design** : Composants UI prêts à l'emploi avec design moderne
- **React Query ou SWR** : Gestion intelligente des appels API et du cache
- **Zustand ou Redux Toolkit** : Gestion d'état globale simplifiée
- **React Hook Form** : Gestion performante des formulaires avec validation
- **Recharts ou Chart.js** : Visualisation de données et graphiques interactifs
- **React Table** : Tableaux de données avancés avec tri, filtrage et pagination
- **Axios** : Client HTTP pour la communication avec l'API Django

6.3. Base de données

SGBD : PostgreSQL 15+

Justification : PostgreSQL est un système de gestion de base de données relationnelle open-source, puissant et fiable :

- Conformité ACID pour l'intégrité des données
- Performances élevées
- Support des transactions complexes
- Fonctionnalités avancées (JSON, recherche plein texte)
- Excellente intégration avec Django

Alternative : MySQL (selon les contraintes d'infrastructure)

6.4. Outils et environnement

Gestion de version : Git / GitHub

Environnement de développement :

- Python : virtualenv ou Poetry pour la gestion des dépendances
- Node.js et npm : pour les dépendances React

Serveur de développement :

- Backend : Django development server
- Frontend : Serveur de développement intégré à l'outil de construction (par exemple, Vite Dev Server)

Déploiement (production) :

- Serveur web : Nginx ou Apache
- Serveur d'application : Gunicorn (pour Django)
- Reverse proxy : Nginx

Autres outils :

- Postman : pour tester l'API

- pgAdmin : pour administrer PostgreSQL
- Visual Studio Code : éditeur de code recommandé

VII. Structure de la base de données

7.1. Schéma relationnel

Table : User (Utilisateurs)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
username	VARCHAR(150)	Nom d'utilisateur (unique)
email	VARCHAR(254)	Adresse email
password	VARCHAR(128)	Mot de passe hashé
first_name	VARCHAR(150)	Prénom
last_name	VARCHAR(150)	Nom
role	VARCHAR(20)	Rôle (ADMIN, PHARMACIEN)
is_active	BOOLEAN	Compte actif
date_joined	DATETIME	Date de création
last_login	DATETIME	Dernière connexion

Table : Fournisseur (Suppliers)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
nom	VARCHAR(200)	Nom du fournisseur
adresse	TEXT	Adresse complète
telephone	VARCHAR(20)	Numéro de téléphone

email	VARCHAR(254)	Adresse email
contact_nom	VARCHAR(200)	Nom du contact
date_creation	DATETIME	Date d'ajout
is_active	BOOLEAN	Fournisseur actif

Table : CategorieMedicament (Drug Categories)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
nom	VARCHAR(100)	Nom de la catégorie
description	TEXT	Description

Table : Medicament (Drugs)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
nom_commercial	VARCHAR(200)	Nom commercial
dci	VARCHAR(200)	Dénomination Commune Internationale
forme_galenique	VARCHAR(100)	Forme (comprimé, sirop, etc.)
dosage	VARCHAR(100)	Dosage
categorie_id	INTEGER (FK)	Référence à CategorieMedicament
fabricant	VARCHAR(200)	Nom du fabricant
code_barre	VARCHAR(50)	Code-barres (unique)
prix_achat	DECIMAL(10,2)	Prix d'achat unitaire
prix_vente	DECIMAL(10,2)	Prix de vente unitaire
seuil_alerte	INTEGER	Stock minimum avant alerte
conditions_stockage	TEXT	Conditions de conservation
description	TEXT	Description et indications
date_creation	DATETIME	Date d'ajout

is_active	BOOLEAN	Médicament actif
-----------	---------	------------------

Table : Stock (Inventory)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
medicament_id	INTEGER (FK)	Référence à Medicament
quantite_disponible	INTEGER	Quantité totale disponible
derniere_maj	DATETIME	Dernière mise à jour

Table : LotMedicament (Drug Batches)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
medicament_id	INTEGER (FK)	Référence à Medicament
numero_lot	VARCHAR(100)	Numéro de lot
date_fabrication	DATE	Date de fabrication
date_peremption	DATE	Date de péremption
quantite	INTEGER	Quantité dans ce lot
prix_achat_unitaire	DECIMAL(10,2)	Prix d'achat pour ce lot
fournisseur_id	INTEGER (FK)	Référence à Fournisseur
date_reception	DATETIME	Date de réception

Table : MouvementStock (Stock Movements)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique

medicament_id	INTEGER (FK)	Référence à Medicament
type_mouvement	VARCHAR(20)	ENTREE ou SORTIE
quantite	INTEGER	Quantité concernée
date_mouvement	DATETIME	Date et heure du mouvement
utilisateur_id	INTEGER (FK)	Utilisateur ayant effectué l'opération
motif	TEXT	Motif/Observations
reference_externe	VARCHAR(100)	N° bon livraison ou ordonnance
lot_medicament_id	INTEGER (FK)	Référence au lot concerné

Table : Entrée (Stock Entries)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
numero_bon_livraison	VARCHAR(100)	Numéro du bon de livraison
fournisseur_id	INTEGER (FK)	Référence à Fournisseur
date_reception	DATETIME	Date de réception
utilisateur_id	INTEGER (FK)	Utilisateur ayant enregistré
observations	TEXT	Observations

Table : LigneEntree (Entry Lines)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
entree_id	INTEGER (FK)	Référence à Entree
lot_medicament_id	INTEGER (FK)	Référence à LotMedicament
quantite	INTEGER	Quantité reçue
prix_achat_unitaire	DECIMAL(10,2)	Prix d'achat unitaire

Table : Sortie (Stock Outputs)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
numero_ordonnance	VARCHAR(100)	Numéro d'ordonnance
patient_nom	VARCHAR(200)	Nom du patient
patient_id	VARCHAR(50)	Identifiant patient
prescripteur	VARCHAR(200)	Nom du prescripteur
date_dispensation	DATETIME	Date de dispensation
utilisateur_id	INTEGER (FK)	Pharmacien dispensateur
observations	TEXT	Observations

Table : LigneSortie (Output Lines)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
sortie_id	INTEGER (FK)	Référence à Sortie
lot_medicament_id	INTEGER (FK)	Référence à LotMedicament
quantite	INTEGER	Quantité dispensée
prix_vente_unitaire	DECIMAL(10,2)	Prix de vente unitaire

Table : Alerte (Alerts)

Champ	Type	Description

id	INTEGER (PK)	Identifiant unique
medicament_id	INTEGER (FK)	Référence à Medicament
type_alerte	VARCHAR(50)	STOCK_FAIBLE ou PEREMPTION
message	TEXT	Message de l'alerte
date_creation	DATETIME	Date de création de l'alerte
est_lue	BOOLEAN	Alerte lue ou non
date_lecture	DATETIME	Date de lecture
utilisateur_lecture_id	INTEGER (FK)	Utilisateur ayant lu l'alerte

Table : Inventaire (Physical Inventory)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
date_inventaire	DATE	Date de l'inventaire
utilisateur_id	INTEGER (FK)	Utilisateur responsable
statut	VARCHAR(20)	EN_COURS, VALIDE, ANNULE
observations	TEXT	Observations générales
date_creation	DATETIME	Date de création
date_validation	DATETIME	Date de validation

Table : LigneInventaire (Inventory Lines)

Champ	Type	Description
id	INTEGER (PK)	Identifiant unique
inventaire_id	INTEGER (FK)	Référence à Inventaire
medicament_id	INTEGER (FK)	Référence à Medicament
quantite_theorique	INTEGER	Quantité dans le système
quantite_physique	INTEGER	Quantité comptée physiquement

ecart	INTEGER	Différence (calculée)
justification	TEXT	Justification de l'écart

7.2. Diagramme de relations

Entité A	Cardinalités entre A et B	Entité B
User	(1,n)	MouvementStock, Entree, Sortie, Inventaire
Fournisseur	(1,n)	Entree, LotMedicament
CategorieMedicament	(1,n)	Medicament
Medicament	(1,1)	Stock
Stock	(1,n)	LotMedicament, MouvementStock, Alertre, LigneInventaire
Entree	(1,n)	LigneEntree
LigneEntree	(n,1)	LotMedicament
Sortie	(1,n)	LigneSortie
LigneSortie	(n,1)	LotMedicament
Inventaire	(1,n)	LigneInventaire

7.3. Règles de gestion

1. Un utilisateur ne peut avoir qu'un seul rôle (ADMIN ou PHARMACIEN)
2. Un médicament appartient à une seule catégorie
3. Un médicament peut avoir plusieurs lots avec des dates de péremption différentes
4. Chaque mouvement de stock doit être associé à un utilisateur (traçabilité)

5. Les sorties utilisent la méthode FIFO (First In, First Out) pour les lots
6. Une alerte de stock faible est générée automatiquement si quantité_disponible ≤ seuil_alerte
7. Une alerte de péremption est générée automatiquement si date_peremption - date_actuelle ≤ 6 mois
8. Un inventaire ne peut être modifié une fois validé
9. Les prix sont stockés en devise locale (FCFA ou autre selon la clinique)
10. Toutes les dates et heures sont enregistrées avec timezone

VIII. Planning et réalisation

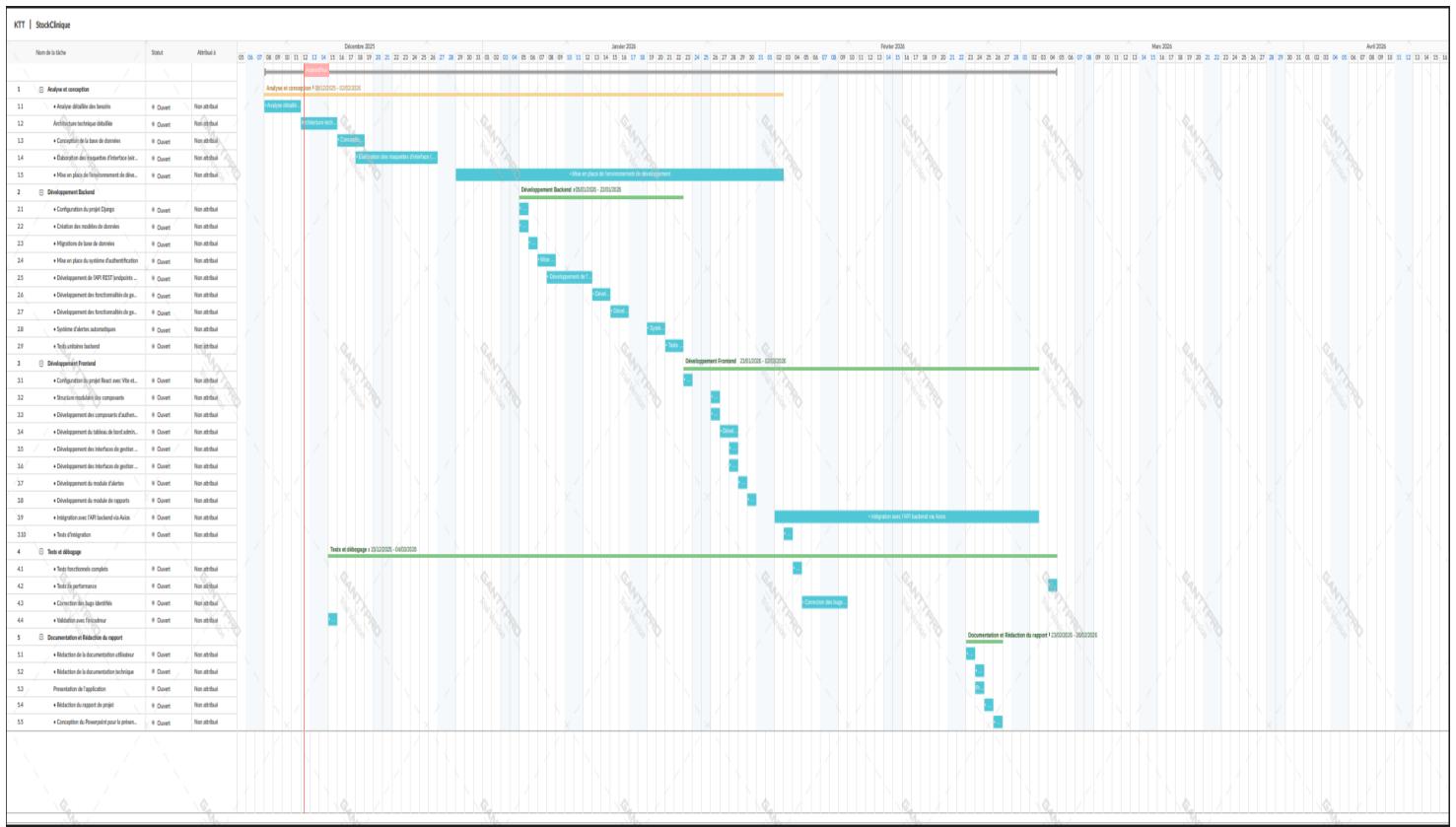
- **Phase 1 : Analyse et conception (2 semaines)**
 - **Livrables :**
 - Cahier des charges validé
 - Diagrammes UML (cas d'utilisation, classes, séquences)
 - Maquettes d'interface
 - Schéma de la base de données
- **Phase 2 : Développement Backend (4 semaines)**
 - **Livrables :**
 - API REST fonctionnelle
 - Documentation de l'API (Swagger/OpenAPI)
 - Base de données opérationnelle
- **Phase 3 : Développement Frontend (4 semaines)**
 - **Livrables :**
 - Application frontend complète
 - Interfaces utilisateur fonctionnelles et responsive
- **Phase 4 : Tests et débogage (2 semaines)**
 - **Livrables :**
 - Application testée et validée
 - Rapport de tests
 - Liste des bugs corrigés

- **Phase 5 : Documentation et Réécriture du rapport (1 semaine)**

- **Livrables :**

- Manuel utilisateur
- Documentation technique
- Guide d'installation et de maintenance
- Rapport de projet final
- Support de présentation

- **Diagramme de Gantt**



- **Durée totale estimée : 14 semaines (3,5 mois)**
- **Jalons importants**
- **J1 (Fin semaine 2) : Validation du cahier des charges**
- **J2 (Fin semaine 6) : API backend opérationnelle**
- **J3 (Fin semaine 10) : Application frontend complète**

- **J4** (Fin semaine 12) : Application testée et validée
- **J5** (Fin semaine 13) : Déploiement en production
- **J6** (Fin semaine 14) : Remise du rapport final

Conclusion

Ce projet de conception et de réalisation d'un logiciel de gestion de stocks de médicaments représente une solution complète et moderne pour optimiser la gestion pharmaceutique d'une clinique. En s'appuyant sur des technologies robustes (Django et Angular) et une architecture bien structurée, cette application répond aux besoins critiques de traçabilité, de sécurité et d'efficacité opérationnelle. L'application offrira aux utilisateurs (administrateurs et pharmaciens) des outils puissants pour : suivre en temps réel les stocks de médicaments, prévenir les ruptures et les péremptions, assurer une traçabilité complète des mouvements, générer des rapports d'aide à la décision, sécuriser l'accès aux données sensibles Le respect du planning proposé permettra de livrer une solution fonctionnelle, testée et documentée dans un délai de 14 semaines. Ce projet constituera non seulement un outil pratique pour la clinique bénéficiaire, mais également un travail académique solide démontrant la maîtrise des technologies modernes de développement web et des bonnes pratiques en gestion de projet informatique. La réalisation de ce projet nécessitera rigueur, organisation et application des connaissances acquises en développement logiciel, en gestion de bases de données et en architecture des systèmes d'information. Il s'inscrit pleinement dans les objectifs de formation en développement d'applications et contribuera à l'amélioration de la qualité des soins dans le secteur de la santé.