

REPUBLIQUE DU CAMEROUN

\*\*\*\*\*

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

\*\*\*\*\*

UNIVERSITE DE YAOUNDE 1

\*\*\*\*\*

UNIVERSITE SAINT JEAN INGENIEUR DU  
CAMEROUN



INSTITUT UNIVERSITAIRE SAINT JEAN  
SAINT JEAN INGENIEUR

REPUBLIC OF CAMEROON

\*\*\*\*\*

MINISTRY OF HIGHER EDUCATION

\*\*\*\*\*

UNIVERSITY OF YAOUNDE 1

\*\*\*\*\*

SAINT JEAN INGENIEUR UNIVERSITY OF  
CAMEROON



---

## CONCEPTION ET IMPLEMENTATION D'UNE MARKET- PLACE POUR MEDICAMENT SUR UNE ARCHITECTURE MICROSERVICE

---

### **Rapport de stage INGENIEUR**

Stage effectué du **02/05/2024** au **26/08/2024** à **SCI 2M**

Rédigé et Présenté par :

**WAFO KAMGUE JERRY ARISTIDE**

Étudiant en **Quatrième Année**

Option : **Informatique et Système d'Information**

Sous la supervision de :

**Mr TIH SAMUEL, DIRECTEUR TECHNIQUE SCI 2M**

**Année Académique 2024-2025**

---

# CONCEPTION ET IMPLEMENTATION D'UNE MARKET- PLACE POUR MEDICAMENT SUR UNE ARCHITECTURE MICROSERVICE

---

## **Rapport de stage INGENIEUR**

Stage effectué du **02/05/2024** au **22/08/2024** à **SCI 2M**

Rédigé et Présenté par :

**WAFO KAMGUE Jerry Aristide**

Étudiant en **Quatrième Année**

Option : **Informatique et Système d'Information**

Sous la supervision de :

**Mr TIH Samuel** , Directeur Technique SCI 2M

**DÉDICACE**

**À MES PARENTS**

## REMERCIEMENTS

**A**u terme de cette expérience enrichissante au sein de SCI 2M, nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à toutes les personnes qui ont participé à la réussite de ce projet. Ces remerciements s'adressent particulièrement :

- À Dieu, par la volonté de qui tout ceci a été possible ;
- À M. BITANG Marcel, directeur de SCI 2M, pour nous avoir permis d'effectuer notre stage dans son entreprise ;
- Au père Jean-Hervé, directeur de Saint Jean Ingénieurs pour la formation que nous avons reçu;
- Au Pr. KOUAMOU EDOUARD Georges pour m'avoir offert l'opportunité d'acquérir de l'expérience et pour tout le soutien qu'il a bien voulu m'accorder ;
- À notre encadreur professionnel, M. TIH Samuel pour sa disponibilité et ses conseils ;
- À mon compagnon DJOMGUEM Herve pour sa bienveillance et l'environnement convivial partagé avec lui ;
- À ma famille et mes amis pour tous les encouragements et le soutien ;
- À tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail ;

## SOMMAIRES

<b>Remerciements.....</b>	<b>i</b>
<b>Liste des Figures .....</b>	<b>iv</b>
<b>Listes des tables .....</b>	<b>vi</b>
<b>Abréviations .....</b>	<b>vii</b>
<b>Avant-propos.....</b>	<b>viii</b>
<b>Introduction générale.....</b>	<b>1</b>
<b>Chapitre 1 Présentation de l'entreprise et déroulement du stage.....</b>	<b>2</b>
1.1 Historique et emplacement .....	3
1.2 Organisation de SCI 2M .....	6
1.3 Déroulement du stage .....	8
<b>Chapitre 2 Présentation du problème.....</b>	<b>11</b>
2.1 Identification et origine du problème .....	12
2.2 Solutions existantes et limites.....	12
<b>Chapitre 3 Solution proposée et Méthodologie.....</b>	<b>15</b>
3.1 Description de la solution .....	16
3.2 Méthodologie et Environnement de travail .....	16
3.3 Analyse, Conception et Implémentation .....	18
<b>Chapitre 4 Implémentation et résultat .....</b>	<b>32</b>
4.1 Outils et choix technologiques .....	33

4.2	Résultat .....	37
<b>Conclusion .....</b>		<b>45</b>
<b>Bibliographie.....</b>		<b>46</b>
<b>Annexes .....</b>		<b>48</b>

## LISTE DES FIGURES

Figure 1.1-1 : Plan de localisation SCI 2M.....	5
Figure 1.2-1 : Schéma organisationnel de SCI 2M.....	7
Figure 3.2-1 : Illustration de la méthode SCRUM.....	17
Figure 3.3-1 : Diagramme de cas d'utilisation point de vue du pharmacien.....	22
Figure 3.3-2 : Diagramme de cas d'utilisation du point de vue du client.....	22
Figure 3.3-3 : Diagramme de classe métier .....	25
Figure 3.3-4 : Architecture logique .....	27
Figure 3.3-5 : Diagramme de composant.....	28
Figure 3.3-6 : Cas d'utilisation « Commander des médicaments » .....	29
Figure 3.3-7 : Cas d'utilisation « Chatter avec un pharmacien ».....	30
Figure 3.3-8: Cas d'utilisation « Proposer des médicaments ».....	31
Figure 4.1-1: Framework Angular.....	33
Figure 4.1-2: Framework Springboot.....	33
Figure 4.1-3 : Bibliothèque Socket.io .....	34
Figure 4.1-4 : SGBD Postgres.....	34
Figure 4.1-5: Plateforme Docker.....	35
Figure 4.1-6 : Plateforme Gitlab .....	35
Figure 4.1-7 : Logiciel Postman.....	36
Figure 4.1-8 : Editeur IntelliJ.....	36
Figure 4.2-1 : Framework OpenApi.....	37
Figure 4.2-2 : Documentation de l'api du microservice Pharmacy .....	38
Figure 4.2-3 : Documentation de l'api du micro-service Pharmacy.....	38
Figure 4.2-4 : Test d'API via Postman.....	39
Figure 4.2-5 : Distributeur RabbitMQ .....	40
Figure 4.2-6 : Distribution d'un message via RabbitMQ .....	40
Figure 4.2-7: Base de données Redis .....	41
Figure 4.2-8 : Interface d'exécution de Redis.....	41

Figure 4.2-9: Extension PostGIS.....	42
Figure 4.2-10 : Modèle de réponse Postgis.....	42
Figure 4.2-11 : Carte représentant l'emplacement des pharmacies.....	43
Figure 4.2-12: Bibliothèque Redis .....	44



## LISTES DES TABLES

Tableau 1.1-1: Fiche d'informations .....	3
Tableau 1.3-1: Tâches effectuées en entreprise.....	9
Tableau 3.3-1: Besoins fonctionnels .....	19
Tableau 3.3-2 : Besoins non-fonctionnels .....	21
Tableau 3.3-3 : Explication du cas d'utilisation "Gérer ma commande" .....	22
Tableau 3.3-4 : Explication du cas d'utilisation "Proposer des médicaments" ...	23

## ABRÉVIATIONS

Abréviations	Significations
<b>SCI 2M</b>	Société Civile Immobilière 2M
<b>SJI</b>	Saint Jean Ingénieurs
<b>IUSJC</b>	Institut Universitaire Saint Jean Ingénieur du Cameroun
<b>ENSPY</b>	École Nationale Polytechnique de Yaoundé
<b>CTI</b>	Commission du Titre d'Ingénieur
<b>IPES</b>	Institut Privée d'Enseignement Supérieur
<b>DDD</b>	Domain Driven Design
<b>IDE</b>	Integrated Development Environment
<b>API</b>	Application Programming Interface
<b>UML</b>	Unified Modeling Language
<b>HTTP</b>	Hyper Text Transfert Protocol
<b>REST</b>	Representational State Transfer
<b>ACID</b>	Atomicity Consistency Isolate Durability
<b>SQL</b>	Structured Query Language
<b>AMQP</b>	Advanced Message Queuing Protocol
<b>SIG</b>	Systèmes d'Information Géographique

## AVANT-PROPOS

Saint Jean Ingénieurs (SJI) est une faculté de l'Institut Universitaire Saint Jean du Cameroun (IUSJC) qui a été fondée en 2016. Son but est de répondre aux besoins de formations techniques professionnelles et d'ingénierie de haut niveau sans devoir quitter le Cameroun.

Elle est placée sous la tutelle académique de l'École Nationale Polytechnique de Yaoundé (ENSPY) et a comme principal partenaire pédagogique l'Université Technologique de Troyes en France qui l'a accompagné dans son processus d'accréditation CTI (accréditation qui permet aux diplômes délivrés d'être reconnus en Europe et en Amérique du Nord). Composée d'un cursus décliné en Français et en Anglais pour le cycle Ingénieur et Licence et d'un Master professionnels en informatique, SJI fait partie du cercle restreint des IPES autorisés au Cameroun à délivrer un diplôme d'ingénieur officiel et reconnu par le MINESUP (arrêté n°22-0005/MINESUP du 12 janvier 2022).

Durant le cursus de chaque étudiant à Saint Jean Ingénieurs, de nombreux stages sont prévus parmi lesquels un stage de trois (03) à quatre (04) mois après le second semestre de la quatrième année. Au cours dudit stage, les étudiants réalisent un projet professionnel qui fait l'objet à son terme d'une présentation devant un jury constitué à cet effet.

Ce rapport est le fruit d'un stage de quatre mois, réalisé au sein de l'entreprise SCI 2M en vue de la validation de la quatrième année d'ingénieur en option Informatique et Système d'Information (ISI).



## INTRODUCTION GÉNÉRALE

Dans le cadre de mon stage de quatre mois au sein de la Société Civile Immobilière 2M (SCI 2M), j'ai eu l'opportunité de contribuer à un projet de grande envergure, centré sur la conception et l'implémentation d'une marketplace de médicaments. Cette mission s'inscrit dans la volonté de l'entreprise d'améliorer l'expérience utilisateur de son application GUI-MBÔ, dédiée à la gestion des actifs immobiliers, tout en y intégrant des services essentiels tels que la gestion des stocks pharmaceutiques et la géolocalisation des pharmacies.

Le choix de développer l'application s'est faite avec des contraintes architecturales pour pouvoir garantir une évolutivité, une flexibilité et une maintenabilité accrues. Mon travail s'est donc concentré sur le développement de plusieurs micro-services.

Ainsi, notre rapport sera divisé en quatre chapitres. Dans le premier chapitre nous présenterons l'entreprise et les tâches effectuées. Dans le deuxième nous détaillerons le problème et étudierons l'existant. Dans le troisième chapitre nous ferons un point sur la méthodologie de travail ainsi que la partie analyse et conception. Et enfin à la quatrième partie nous présenterons le résultat et les technologies utilisées.

# CHAPITRE



## PRÉSENTATION DE L'ENTREPRISE ET DÉROULEMENT DU STAGE

*Dans ce chapitre, nous allons faire une présentation de l'entreprise dans laquelle nous avons effectué notre stage pré-ingénieur : Société Civile Immobilière 2M*

## 1.1 Historique et emplacement

### 1.1.1 Historique

La Société Civile Immobilière 2M est née en 2005 de l'initiative de Mr Marcel BITANG. Ses locaux se trouvent au quartier Bastos, plus précisément au quartier Golf près du palais des Congres, lieu où nous avons effectué notre stage. Elle fait dans la location, ou dans la vente d'appartements et résidences, ainsi que des terrains aménagés.

<b>Initiales</b>	<b>SCI 2M</b>
<b>Nom de l'entreprise</b>	Société Civile Immobilière 2M
<b>Siege</b>	Carrefour Foire et le rond-point Bastos Yaoundé
<b>Statut</b>	Société À Responsabilités Limitées

*Tableau 1.1-1: Fiche d'informations*

### 1.1.2 Valeurs et objectif de SCI 2M

*"Celui qui n'a pas d'objectif ne risque pas de leur atteindre" - Sun Tzu.*

Comme quoi toute entreprise voulant avoir un avenir prospère et dominant doit avoir des valeurs et des objectifs propres à elle. Elle a pour objectif de devenir le leader en services immobilier au Cameroun dans 10 ans à l'aide des outils informatiques.

#### **1.1.2.1 Valeurs**

La réussite d'une entreprise ne relève pas seulement de sa bonne gestion ou de la qualité de ses produits ou prestations, elle passe aussi par quelque chose qui ne s'inscrit ni dans les cahiers des charges, ni dans les tableaux Excel: La culture d'entreprise. SCI 2M se distingue par des valeurs fortes tels que l'Amour, Justice et Intégrité.

#### **1.1.2.2 Objectifs**

SCI 2M est une entreprise œuvrant dans le secteur de l'immobilier. Elle ambitionne d'offrir des services novateurs dans ce secteur d'activité et s'appuie fortement sur les technologies de l'information et de communication. Nous pouvons citer entre autres:

- ❖ Augmenter de façon significative la productivité et les gains des clients en leur fournissant à moindre coût et dans les délais les produits numériques adaptés ;
- ❖ Satisfaction continue des clients afin de développer un relationnel puissant avec sa clientèle ;
- ❖ Promouvoir des logements décents au Cameroun en créant des complexes résidentiels modernes et sécurisés ;
- ❖ Offrir des produits et services innovants, répondant aux besoins de toutes les classes sociales, incluant espaces verts, loisirs, et commerces ;Emplacement

Le siège social et le bureau technique de SCI 2M se situent à Yaoundé au lieu-dit carrefour Foire



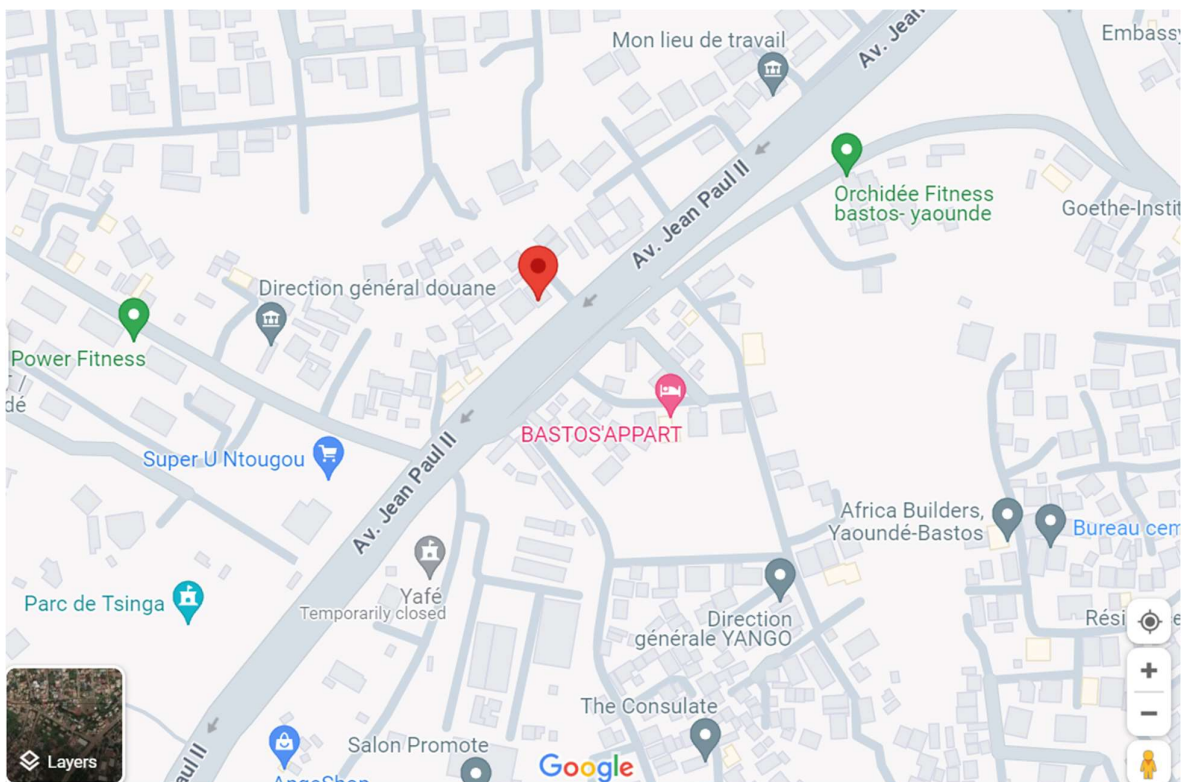


Figure 1.1-1 : Plan de localisation SCI 2M

## 1.2 Organisation de SCI 2M

La structure de l'entreprise est organisée comme suit :

1. **Direction Générale** : à la tête de celle-ci, se trouve un Directeur Général. Il a un rôle majeur dans la gestion de l'entreprise, il doit donc veiller à ce que SCI 2M atteigne tous ses objectifs en restant en accord avec cette valeur ;
2. **La Direction Des Ressources Humaines** : il a pour rôle d'attirer, d'acquérir et conserver les ressources humaines en qualité afin de permettre à l'entreprise de réaliser ses objectifs. Il assure l'évolution et le développement de l'ensemble du personnel et l'administration quotidienne de ressources humaines ;
3. **Direction Commerciale** : à sa tête, se trouve un directeur commercial. Il a pour principale fonction la mise en place d'une politique de vente et de liquidation de stocks tels que les produits finis et les services ;
4. **Direction Technique** : le métier de directeur technique fait partir du domaine de recherche. Si les missions peuvent différer selon l'activité de l'entreprise, le directeur technique garde toujours son rôle essentiel d'interface entre le directeur général et les différents pôles de l'entreprise ;

Ce schéma illustratif permet d'avoir une vue plus détaillée de l'organisation de l'entreprise

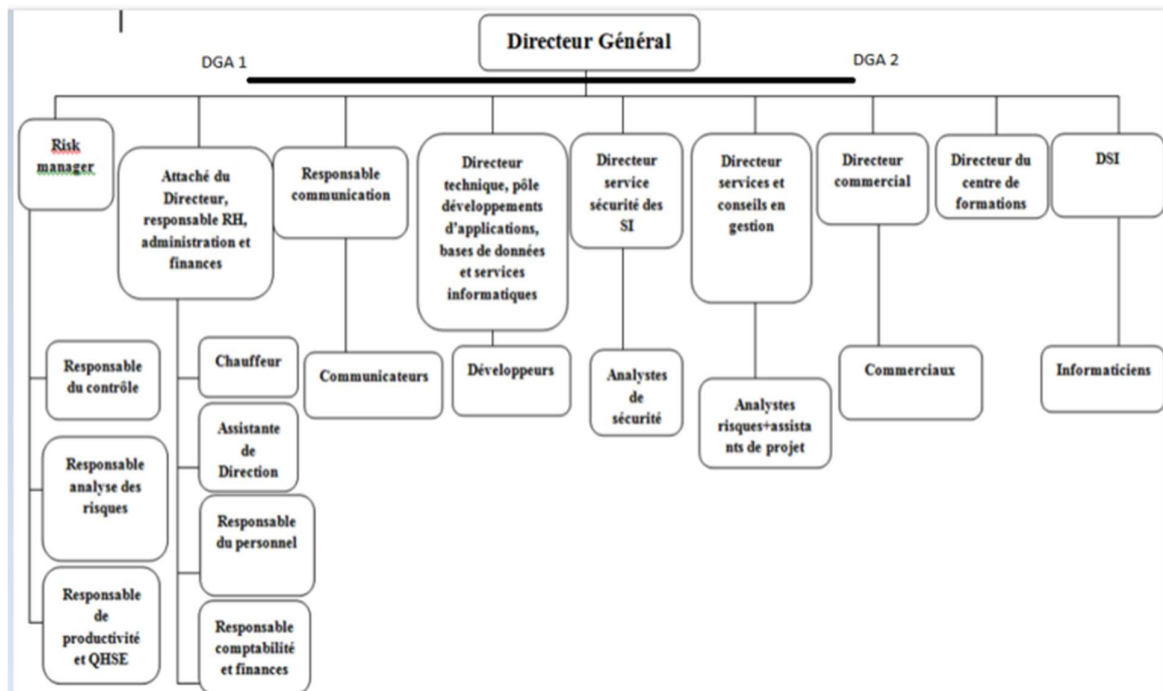


Figure 1.2-1 : Schéma organisationnel de SCI 2M

## 1.3 Déroulement du stage

### 1.3.1 Activités menées

Dans cette partie nous listerons toutes les tâches réalisées durant nos 16 semaines de stage :

Semaines	Tâches effectuées
Semaine 1	Test technique orienté sur la réalisation d'une maquette complète d'une application web responsive sur le starter Angular de l'entreprise.
Semaine 2 et 3	Prise en main de l'application Figma et propositions d'un nouveau design pour l'application GUI-MBÔ en utilisant Figma.
Semaine 4	Prise en main du document relatif aux besoins pour la conception du marketplace et début des séances de brainstorming avec la hiérarchie
Semaine 5	Prise en main du Starter Spring et compréhension globale de son fonctionnement.
Semaine 6	Prise en main des documents existant relatif à l'utilisation des micro-service de l'application GUI-MBÔ.
Semaine 7	Apprentissage du DDD Apprentissage du logiciel draw.io Conception des différents diagrammes
Semaine 8	Apprentissage de RabbitMQ, Redis. Début de la réalisation du micro-service de gestion des médicaments et en java.

	Apprentissage de JUnit et Mockito. Configuration des environnements et profils de tests et de production. Réalisation des test unitaires et des tests d'intégrations.
<b>Semaine 9</b>	
<b>Semaine 10</b>	Début de la réalisation du micro-service de gestion des stocks. Réalisation des tests unitaires et d'intégrations du micro-service de gestion de stock .
<b>Semaine 11</b>	Apprentissage de PostGIS pour la géolocalisation Début de la réalisation du micro-service de géolocalisation en Java. Réalisation des tests unitaire et d'Intégration.
<b>Semaine 12</b>	Apprentissage du Framework OpenAPI Documentation de toutes les Apis réalisées grâce au Framework OpenAPI. Réalisation d'une interface fonctionnelle pour tester notre Backend dans des conditions d'utilisation.
<b>Semaine 13</b>	Implémentation des webSocket (SocketIO) pour le chat et la communication avec le serveur de messagerie. Implémentation du panier avec la bibliothèque Redux pour la gestion du panier.
<b>Semaine 14</b>	Finalisation du projet.

*Tableau 1.3-1: Tâches effectuées en entreprise*

### **1.3.2 Difficultés rencontrées**

Comme difficultés rencontrées on peut citer :

- ❖ La compréhension du domaine métier : il était difficile de comprendre le domaine du marketplace et de l'agencer avec le domaine pharmaceutique ;
- ❖ Les méthodes de conceptions étant différentes il nous a fallu pour chaque méthode un temps d'adaptation afin de comprendre au mieux les concepts utilisés ;
- ❖ La compréhension des programmes déjà existant ;
- ❖ Le problème de version des dépendances sur java ;
- ❖ Les lieux de restaurations presque inexistant dans le secteur ;

### **CONCLUSION**

Cette partie avait pour but de présenter globalement l'entreprise et le déroulement de notre stage. Nous avons aussi énuméré quelques difficultés de notre stage, qui nous ont permis d'améliorer notre façon d'aborder un projet et bien sûr de gagner en compétences. Par la suite, nous présenterons le problème que nous avons tenté de résoudre et détaillerons la démarche mise en place pour y parvenir.

## CHAPITRE

# 2

### PRÉSENTATION DU PROBLÈME

*Dans ce chapitre, nous allons présenter le problème que l'entreprise souhaite résoudre ainsi que son impact, lister les solutions existantes et leurs limites.*

## **2.1 Identification et origine du problème**

L'ajout d'une marketplace pour médicaments découle de :

- ❖ La nécessité de créer une plateforme plus intégrée et complète, capable de gérer non seulement les aspects immobiliers mais aussi les services essentiels au bien-être des résidents, en particulier dans des contextes comme les établissements de santé, les résidences avec services médicaux ;
- ❖ Ce besoin d'intégration est né de la demande croissante des utilisateurs pour des solutions tout-en-un qui simplifient leur gestion quotidienne, en rassemblant divers services sous une même interface ;
- ❖ En offrant cette nouvelle fonctionnalité SCI 2M cherche à se positionner par l'intermédiaire de GUI-MBÔ comme un acteur clé dans l'évolution du marché, où la gestion immobilière ne se limite plus aux biens matériels, mais englobe aussi la gestion des services essentiels aux personnes ;

## **2.2 Solutions existantes et limites**

Face aux problèmes liés aux soucis de la satisfaction du client et au désir de positionnement en tant qu'acteurs clés dans le marché de l'immobilier, nous avons dû étudier quelques solutions existantes et ressortir leurs limites afin de construire une application sur mesure qui réponds aux besoins des clients.

Nous pouvons relever entre autres :

### **2.2.1 Pharmacie Lafayette**

Pharmacie Lafayette est une plateforme en ligne française qui permet aux utilisateurs d'acheter des médicaments et des produits de santé en ligne. Elle



propose une large gamme de produits, y compris des médicaments sur ordonnance, des produits de parapharmacie, et des soins personnels. Elle propose également des options de livraison à domicile. .

Nous pouvons relever comme limite :

- La vérification des ordonnances peut parfois entraîner des délais supplémentaires pour la livraison.

### 2.2.2 CVS Pharmacie

CVS Pharmacie est une chaîne de pharmacies américaine qui propose également une marketplace en ligne où les clients peuvent acheter des médicaments sur ordonnance, des produits de santé et de bien-être, ainsi que des articles de pharmacie. Ils offrent également des services comme la consultation en ligne, la livraison à domicile et des réductions .

Nous avons pu relever comme limite :

- **Complexité de l'interface** : La diversité des produits et services proposés peut parfois rendre la plateforme complexe.

### 2.2.3 HealthKart

HealthKart est une plateforme en ligne indienne spécialisée dans la vente de complément alimentaires, de produits de santé, et de médicaments en ligne. Elle propose : une large gamme de médicament de molécule différente ; Des conseils de santé et des informations sur les produits.

Nous avons pu relever comme limite :

- Moins axé sur les médicaments prescrits, avec un accent sur les suppléments et produits de bien-être.

#### **2.2.4 Medindex**

Medindex est une application mobile camerounaise destinée à faciliter l'accès aux informations sur les médicaments et les services de santé. Elle permet aux utilisateurs de rechercher des médicaments spécifiques, de connaître leurs prix, et de localiser les pharmacies où ils sont disponibles.

Nous avons pu relever comme limite :

- Dépendance aux données fournies : La précision des informations dépend de la mise à jour régulière des données par les pharmacies et les établissements de santé, ce qui peut parfois poser des problèmes.

### **CONCLUSION**

Nous avons décrit brièvement le problème que nous avons constaté, nous avons aussi présenté l'étude de l'existant qui nous a permis de mieux comprendre les défauts de la solution actuelle et de mieux appréhender les défis que notre solution devra relever en prenant en compte tous les besoins de l'entreprise et les ressources mises à notre disposition.

# CHAPITRE 3

## SOLUTION PROPOSÉE ET MÉTHODOLOGIE

***L** s'agit tout au long de ce chapitre de décrire notre démarche de résolution du problème posé, les approches retenues et les technologies implémentés. Nous commençons tout d'abord par présenter le processus logiciel, ensuite l'analyse et enfin la conception du système à mettre sur pied.*

### 3.1 Description de la solution

Notre service de marketplace pharmaceutique intégré à la plateforme GUI-MBÔ est une application web développée pour effectuer des commandes de médicaments dans la pharmacie la plus proche du client 24h/24 ; Elle combine deux parties :

- ❖ **La partie cliente** : Qui permet de commander un médicament ne nécessitant pas d'ordonnance ; D'entrer en contact avec un pharmacien pour un médicament nécessitant une ordonnance , cette partie permet également d'avoir les prix des médicaments en fonction des pharmacies ; Discuter des médicaments en temps réel avec le pharmacien ;
- ❖ **La partie pharmacien** : Qui permet de répondre aux commandes des clients et d'engager des conversations avec ces clients , elle permet également de constituer un panier de médicament et le proposer au client ;

### 3.2 Méthodologie et Environnement de travail

#### 3.2.1 Méthodologie de travail

Pour pouvoir mettre en place notre système, nous avons travaillé en méthode agile Scrum. Cette méthode nous permet de travailler avec les différentes entités (développeurs, designers, experts en accessibilité, chef de projet etc.) qui sont d'une grande importance dans le processus de développement du logiciel.

La période durant laquelle l'équipe conçoit, réalise et teste de nouvelles fonctionnalités s'appelle un « sprint ». Dans ce projet, chaque sprint dure 1 semaine et couvre la majorité des réunions requises en méthode agile Scrum. Nous pouvons citer :

- ❖ **Sprint Planning au début de chaque Sprint** : Cette réunion a pour but de déterminer les tâches que nous allons développer durant le sprint et les classer par ordre de priorité ;
- ❖ **Daily meeting** : C'est la réunion de synchronisation quotidienne, elle se déroule pendant 15 minutes tous les jours durant le Sprint. Elle nous permet de faire le point sur l'avancement des tâches et de résoudre les éventuels problèmes ;
- ❖ **Sprint Review** : Cette réunion marque la fin du sprint. C'est le moment de présenter les différentes réalisations du sprint.
- ❖ **Product Backlog** : Il représente l'ensemble des tâches nécessaires à la réalisation du projet, dans notre cas la liste de tous les composants que nous devons implémenter.
- ❖ **Sprint Backlog** : Il s'agit des tâches qui seront implémentées durant le Sprint.

L'image ci-dessous illustre bien toutes les étapes généralement parcourues durant un Sprint

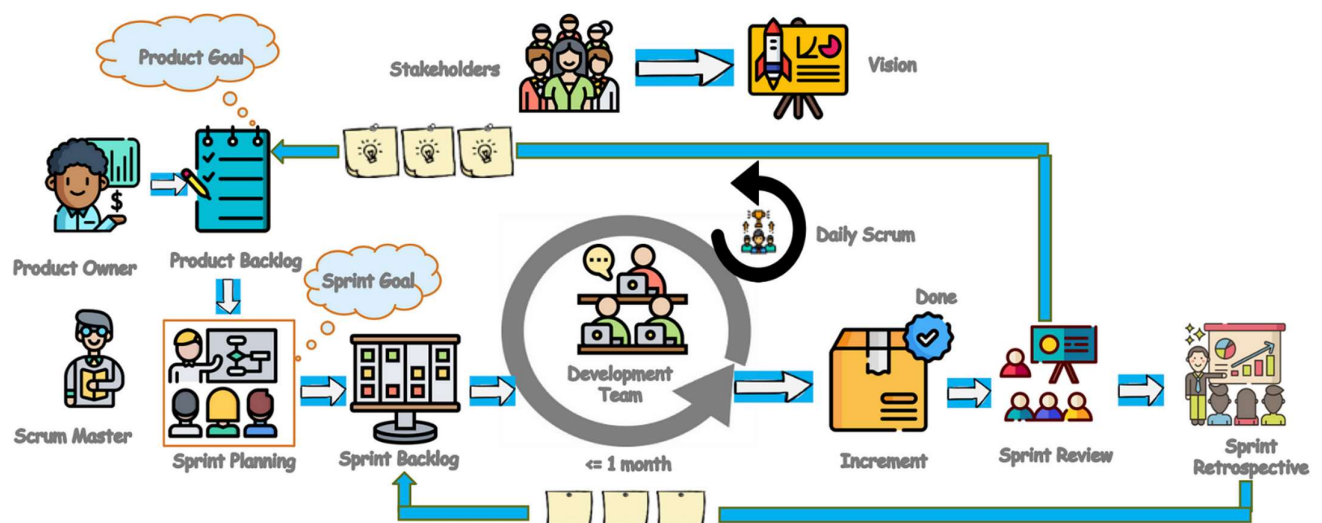


Figure 3.2-1 : Illustration de la méthode SCRUM

### **3.2.2 Environnement de travail**

Pour mettre en place notre application, nous avons utilisé plusieurs environnements pour le développement.

#### **➤ Visual Studio Code**

Visual Studio Code est un éditeur de code source léger et puissant développé par Microsoft. Il est particulièrement apprécié pour sa vitesse, sa flexibilité et son intégration avec des outils de gestion de version comme Git.

#### **➤ IntelliJ Idea**

IntelliJ IDEA est un environnement de développement intégré (IDE) développé par JetBrains, conçu principalement pour le développement en Java. Il facilite le développement de projets complexes et multi-technologies.

#### **➤ Gitlab**

Gitlab est une plateforme de gestion de code source basée sur Git qui offre des fonctionnalités complètes pour le développement collaboratif de logiciels.

#### **➤ Postman**

Postman est un outil de développement utilisé pour tester les API. Il offre une interface conviviale pour créer, envoyer et gérer des requêtes HTTP, permettant aux développeurs de tester les endpoints des API

### **3.3 Analyse, Conception et Implémentation**

Un langage de modélisation est un langage utilisé pour représenter graphiquement et formellement des concepts, des systèmes ou des processus informatiques, facilitant ainsi la conception, la communication et l'analyse. Notre

choix s'est porté sur le langage de modélisation unifié (UML) car il est conçu pour fournir une méthode normalisée pour visualiser la conception d'un système.

### 3.3.1 Analyse

#### 3.3.1.1 Besoins fonctionnels

Utilisateurs	Fonctionnalités
Client	✓ Rechercher un médicament
	✓ Ajouter les médicaments dans le panier
	✓ Mettre un médicament en favoris
	✓ Soumettre une commande
	✓ Engager une discussion avec un pharmacien
	✓ Payer ma commande
Pharmacien	✓ Visualiser les commandes soumises
	✓ Valider les commandes
	✓ Regarder les ordonnances
	✓ Discuter avec un client
	✓ Constituer les médicaments d'un client
	✓ Enregistrer, Modifier et Rechercher un médicament

---

*Tableau 3.3-1: Besoins fonctionnels*

#### 3.3.1.2 Besoins non-fonctionnels

Besoins	Descriptions
Ergonomie	✓ L'application doit être optimisée pour réduire au minimum les délais d'attente pour les utilisateurs lors du chargement des pages et des données.
	✓ Les interfaces seront faites selon la charte graphique GUI-MBÔ ;
	✓ L'application devra être facile à utiliser, avoir des interfaces conviviale
Maintenabilité et évolutivité	✓ L'application doit pouvoir être facilement maintenable, indépendamment du développeur ;
Scalabilité	✓ L'application doit être capable de gérer un grand nombre de connexions simultanées sans compromettre les performances ou la stabilité.
Architecture	✓ L'application doit être en micro-service et doit s'intégrer de manière fluide et efficace dans l'architecture micro-services de l'application GUI-MBÔ
Format d'ordonnance	✓ L'ordonnance peut être envoyé en pdf et en image



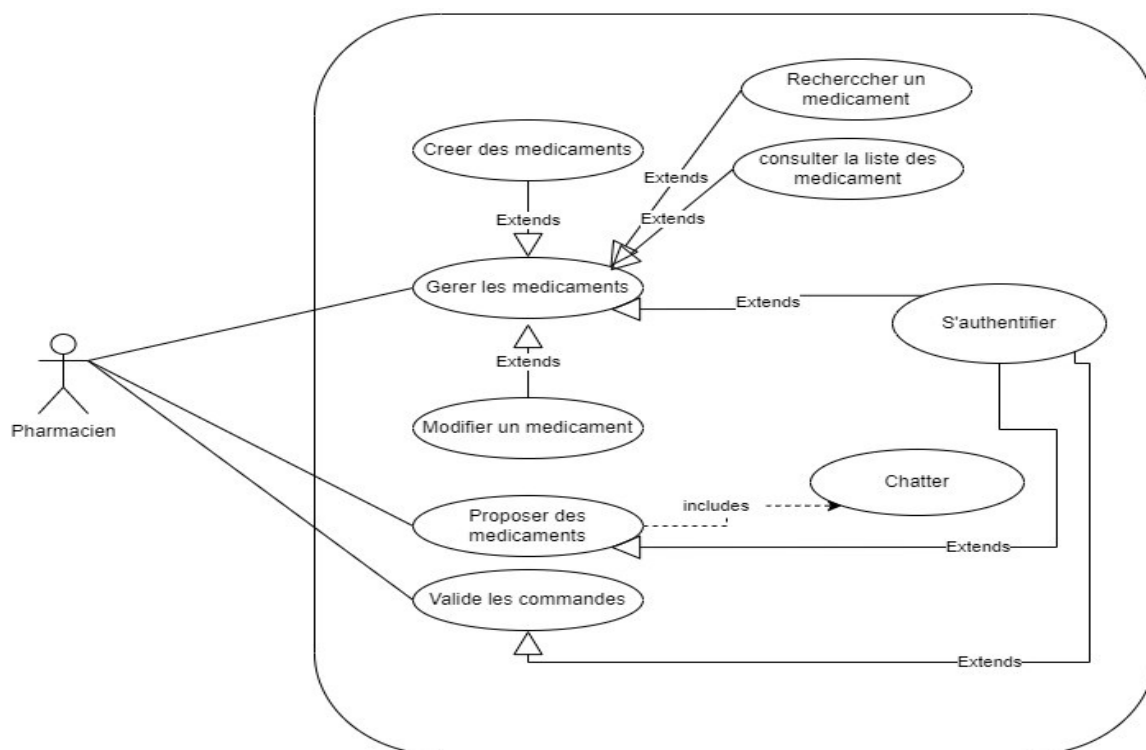
<b>Mode de paiement</b>	✓ Les modes de paiement en ligne devront être uniquement MTN MoMo , Orange Money
<b>Constitution de la commande par le pharmacien</b>	✓ Le client doit voir la constitution en temps réel et il doit pouvoir communiquer avec le pharmacien

Tableau 3.3-2 : Besoins non-fonctionnels

### 3.3.1.3 Diagramme cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme utilisé pour la représentation du comportement fonctionnel d'un système logiciel. Il présente le système du point de vue de l'utilisateur et représente les besoins du client sous forme de cas d'utilisation et d'acteurs. Nous avons les diagrammes suivants:

- *Cas d'utilisation du pharmacien*



- *Cas d'utilisation du client*

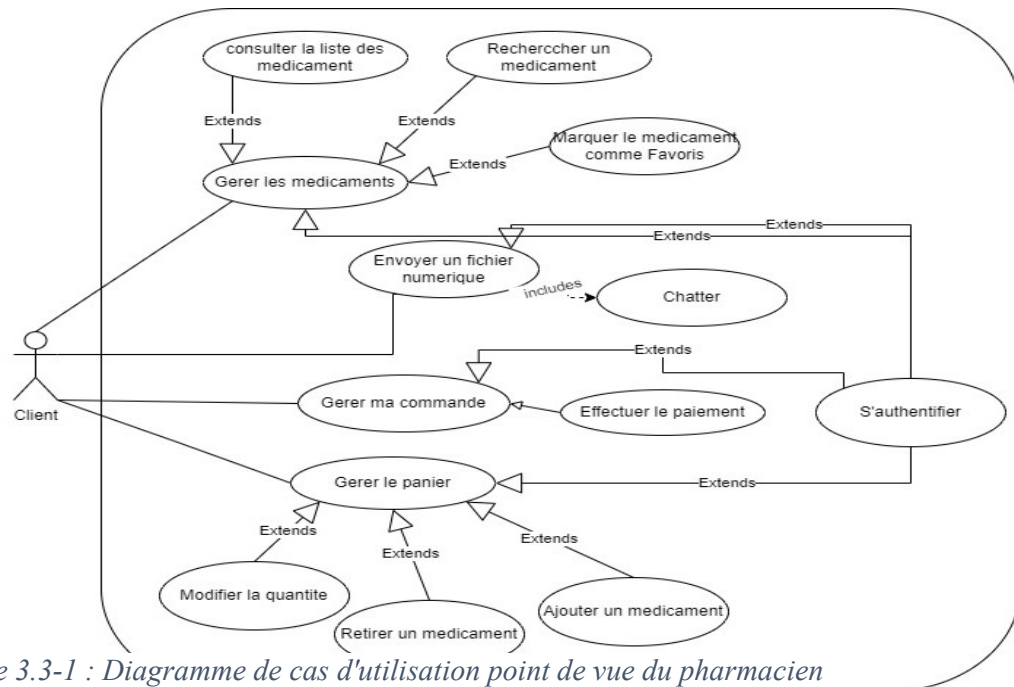


Figure 3.3-1 : Diagramme de cas d'utilisation point de vue du pharmacien

Figure 3.3-2 : Diagramme de cas d'utilisation du point de vue du client

### DESCRIPTION TEXTUELLE DES CAS D'UTILISATIONS :

Nous décrivons ci-dessous les différents cas d'utilisation que nous jugeons pertinents et qui ont besoin d'une description plus avancée. Il s'agit ici de :

- Gérer ma commande
- Proposer les médicaments

Tableau 3.3-3 : Explication du cas d'utilisation "Gérer ma commande"

Cas d'utilisation	Gérer ma commande
<i>Acteur</i>	Acteur

<b>Précondition</b>	Le client doit être authentifié et avoir un panier contenant des articles.
<b>Scénario nominal</b>	<ol style="list-style-type: none"><li>1. Le client sélectionne l'option "Gérer ma commande".</li><li>2. Le système affiche les informations sur les articles du panier.</li><li>3. Le client peut modifier la quantité, ajouter ou retirer des articles.</li><li>4. Le client valide sa commande.</li><li>5. Le système enregistre la commande.</li></ol>
<b>Scenario alternatif</b>	Si les informations fournies sur le médicament sont incomplètes, le système affiche un message d'avertissement et le scénario reprend à l'étape 2.
<b>Postcondition</b>	La proposition est faite avec succès et le système affiche une confirmation.

Tableau 3.3-4 : Explication du cas d'utilisation "Proposer des médicaments"

Proposer des médicaments	
<b>Cas d'utilisation</b>	
<b>Acteur</b>	Pharmacien
<b>Précondition</b>	Le pharmacien doit être authentifié et sélectionné un chat avec un patient(client).

<b><i>Scénario nominal</i></b>	<ol style="list-style-type: none"><li>1. Le pharmacien sélectionne l'option pour proposer un médicament.</li><li>2. Le système lui demande de fournir les informations sur le médicament.</li><li>3. Le pharmacien saisit les informations et valide.</li><li>4. Le système enregistre la proposition de médicament le transfert au client ensuite affiche un message de confirmation.</li></ol>
<b><i>Scenario alternatif</i></b>	Si le panier est vide, le système affiche un message d'erreur et propose au client de retourner à la liste des médicaments.
<b><i>Postcondition</i></b>	La commande est enregistrée avec succès et le client reçoit une confirmation.

#### 3.3.1.4 Diagramme de classe

Cette phase de l'analyse permet de définir les différentes classes que constitue notre système. Une classe est une représentation abstraite d'un objet, elle contient les informations nécessaires à la construction de l'objet. Ce diagramme peut donc être considéré comme le modèle, le moule ou la notice qui va permettre la construction d'un objet

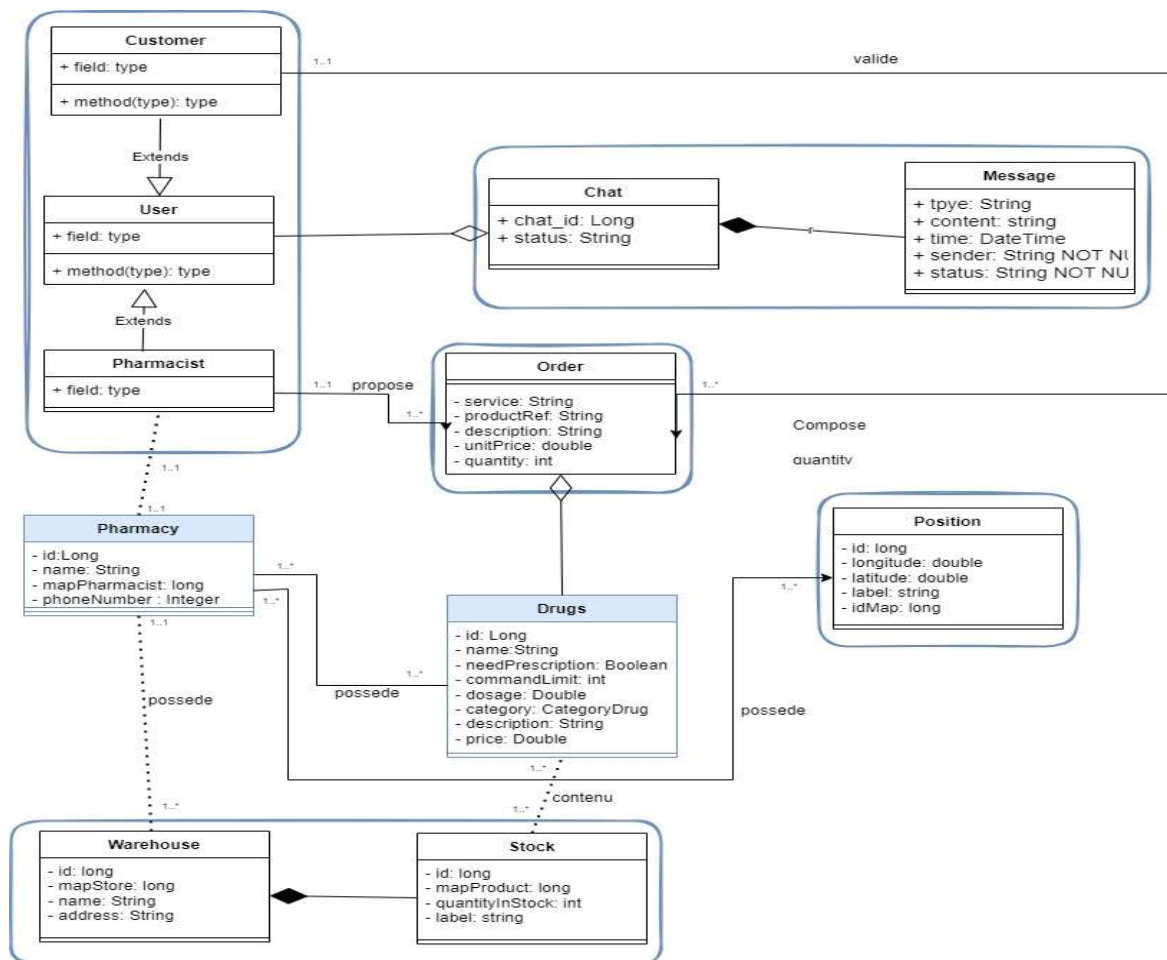


Figure 3.3-3 : Diagramme de classe métier

### 3.3.2 Conception de la solution

### 3.3.2.1 Architecture logique

L'architecture logique du système repose sur une architecture micro-services orchestrée par un API Gateway. Elle est divisée en plusieurs couches et composants interconnectés, chacun ayant un rôle spécifique pour assurer le bon fonctionnement de l'application.

- **Couche de présentation (Présentation layer) :** Cette couche est représentée par une application Web qui interagit avec les utilisateurs

finiaux. Les requêtes de l'utilisateur passent d'abord par l'API Gateway, qui agit comme point d'entrée unique pour les services backend.

➤ **Couche métier (Business layer) :** Cette couche contient les différents micro-services qui implémentent la logique métier de l'application. Chaque micro-service est indépendant et gère une fonctionnalité spécifique, tels que :

- **Geolocation.API :** Gère les fonctionnalités liées à la géolocalisation en utilisant PostGIS pour les données géographiques.
- **Ordering.API :** S'occupe des commandes et des transactions, stockant ses données dans une base de données PostgreSQL.
- **Messaging.API :** Responsable des fonctionnalités de messagerie, utilisant MongoDB pour le stockage.
- **Notification.API :** Gère l'envoi de notifications, avec des interactions via RabbitMQ pour la gestion des messages asynchrones.
- **Pharmacy.API :** S'occupe de la gestion des pharmacies, intégrant PostgreSQL pour le stockage des données et Redis pour la gestion des sessions ou le cache.

Ces micro-services sont encapsulés dans des conteneurs Docker, assurant une gestion isolée et flexible des environnements d'exécution.

➤ **Couche de données (Data access layer) :** Cette couche est constituée des bases de données utilisées par les différents micro-services. Chaque service a accès à sa propre base de données pour le stockage persistant des informations, garantissant une séparation des préoccupations et une scalabilité optimale. Les bases de données incluent PostgreSQL, MongoDB et Redis.

Ainsi, cette architecture micro-services garantit une modularité, une scalabilité et une flexibilité accrue, permettant aux services de communiquer entre eux via des messages asynchrones grâce à RabbitMQ, tout en étant déployés et gérés de manière indépendante avec Docker.

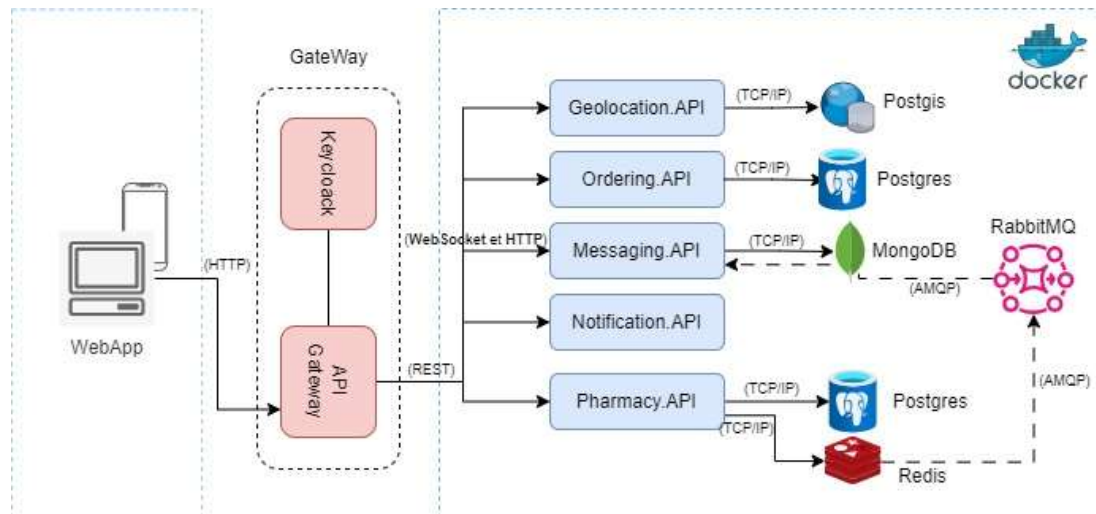


Figure 3.3-4 : Architecture logique

### 3.3.2.2 Diagramme de composant

Le diagramme de composants est une partie essentielle de l'analyse et de la conception d'un système logiciel. Il permet de décrire l'architecture physique du système en mettant en évidence les composants logiciels individuels et leurs relations. Le diagramme de composants offre une vue d'ensemble de la manière dont ces unités interagissent et se connectent entre elles, facilitant ainsi la compréhension de l'architecture globale. Nous avons le diagramme suivant :

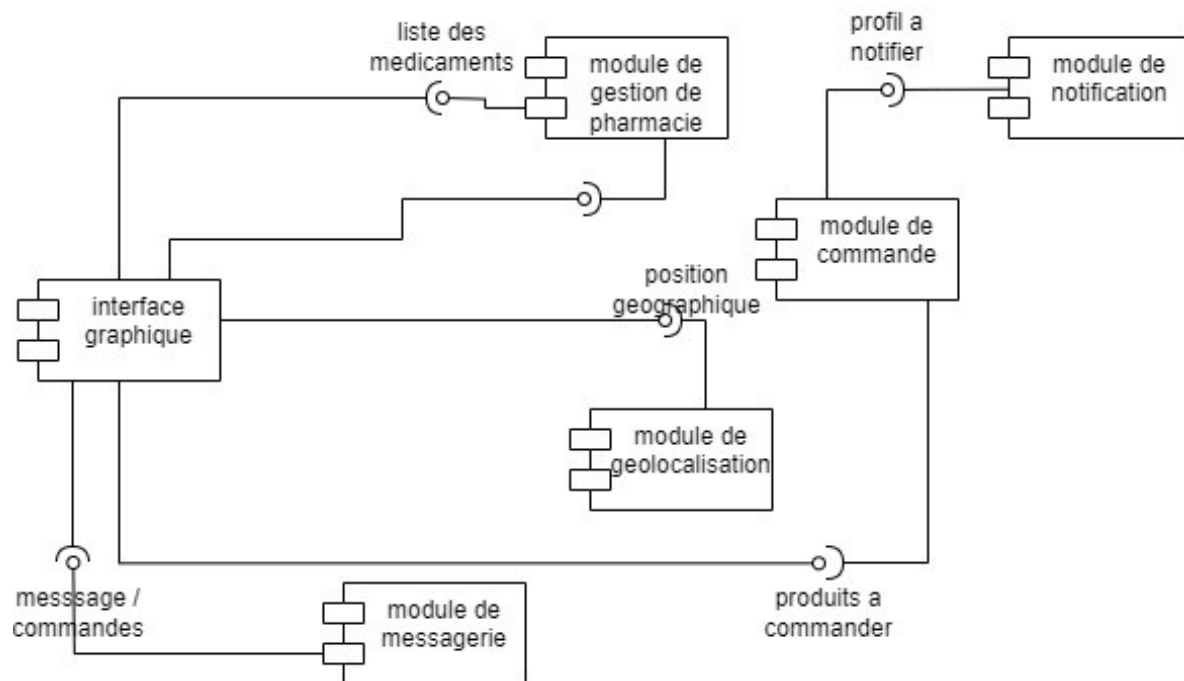


Figure 3.3-5 : Diagramme de composant



### 3.3.2.3 Diagramme de séquence

Le **diagramme de séquence** permet de visualiser l'ordre chronologique des échanges entre les services, mettant en évidence le flux des messages et les étapes de nos processus métier. Voici un exemple de diagramme de séquence illustrant le workflow des interactions entre le client et les services associés dans notre marketplace de médicaments.

- Diagramme de séquence du cas d'utilisation « Commander des médicaments »:

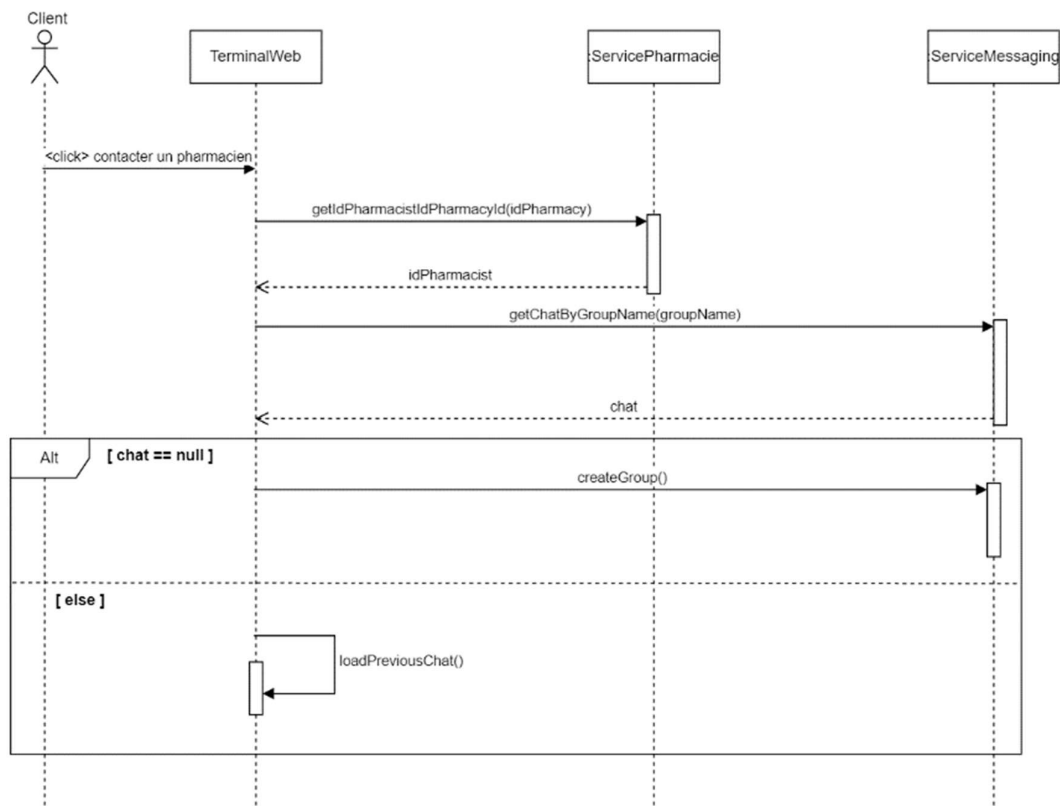


Figure 3.3-6 : Cas d'utilisation « Commander des médicaments »

- Diagramme de séquence du cas d'utilisation « Chatter avec un pharmacien »:

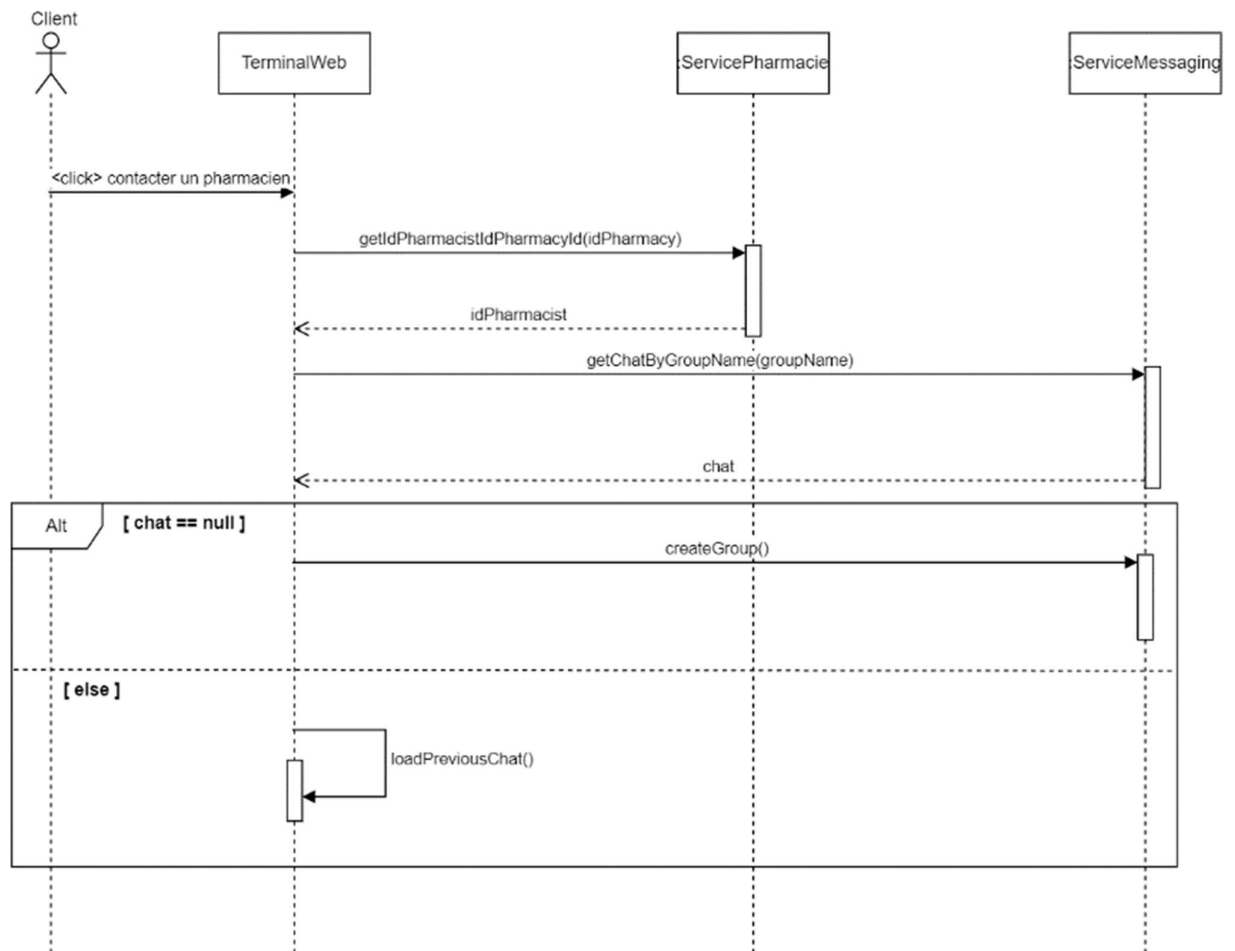


Figure 3.3-7 : Cas d'utilisation « Chatter avec un pharmacien »

- Diagramme de séquence du cas d'utilisation «Proposer des médicaments »:

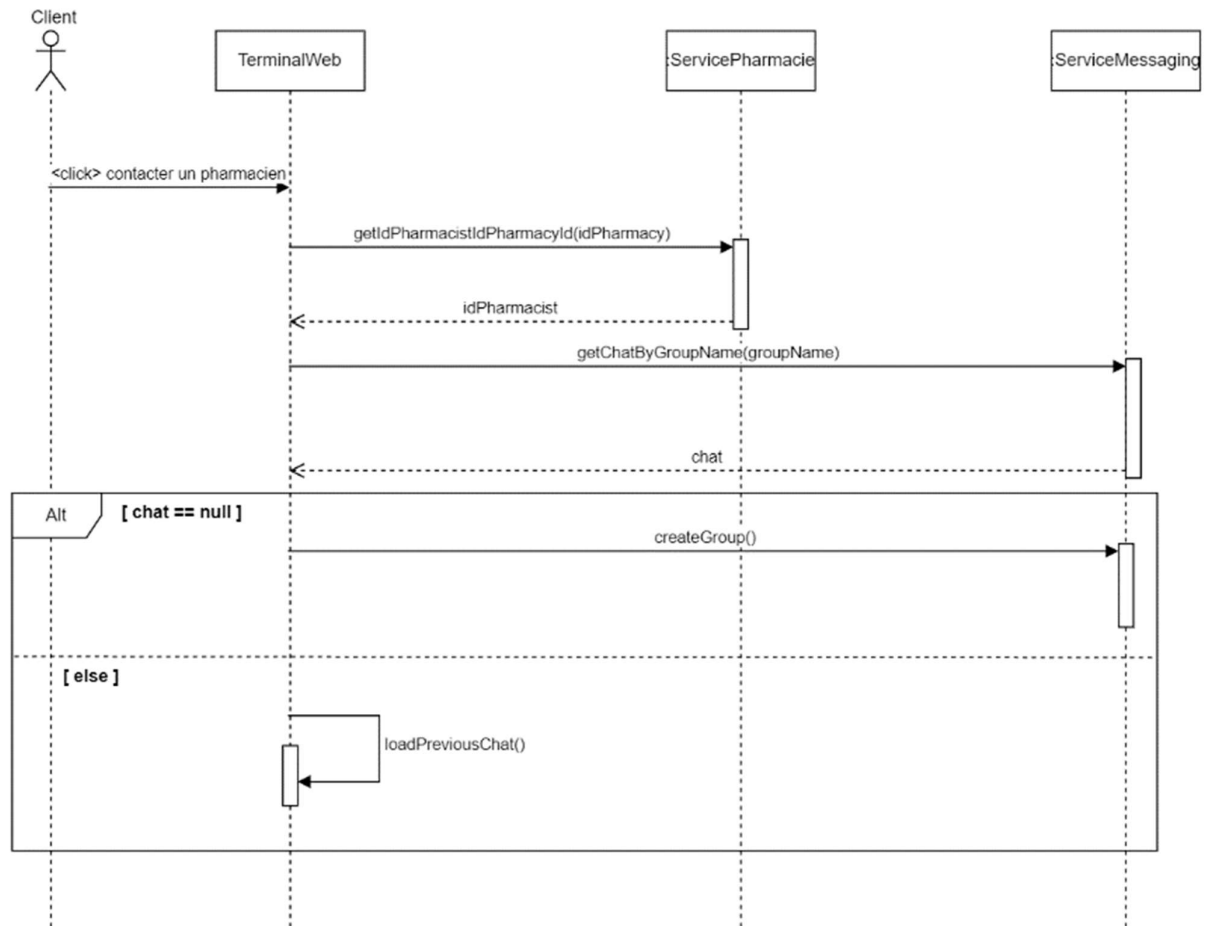


Figure 3.3-8: Cas d'utilisation « Proposer des médicaments »

# CHAPITRE

# 4

## IMPLÉMENTATION ET RÉSULTAT

*Ce chapitre présente les différents outils de travail et les technologies que nous avons utilisés pour la mise en œuvre de la solution décrite dans la méthodologie.*

## 4.1 Outils et choix technologiques

Dans cette partie, nous allons présenter les choix technologiques pour la mise en œuvre du système, et les outils de travail que nous avons utilisés pour le développement.

### 4.1.1 Choix technologiques

Les technologies utilisées ont été choisies en fonction des besoins des développeurs dans les différents services de SCI 2M :

➤ Angular



*Figure 4.1-1: Framework Angular*

Angular est un Framework web client développé par Google, utilisant TypeScript pour créer des applications dynamiques. Il adopte une architecture modulaire basée sur des composants pour le développement d'interfaces interactives. Ses fonctionnalités incluent la liaison bidirectionnelle des données, le routage et la gestion des formulaires.

➤ Springboot



*Figure 4.1-2: Framework Springboot*

Spring Boot est un Framework Java qui simplifie le développement d'applications Spring avec des configurations automatiques. Il permet de créer des applications autonomes, y compris des services web RESTful et des micro-services. En réduisant la configuration manuelle, il accélère le développement backend.

➤ SocketIO



*Figure 4.1-3 : Bibliothèque Socket.io*

Socket.io est une bibliothèque JavaScript qui permet la communication en temps réel entre client et serveur via des WebSocket et autres mécanismes de transport. Elle est utilisée pour des applications interactives comme les chats en temps réel, les jeux en ligne, et les notifications instantanées. Socket.io gère automatiquement la connexion et la reconnexion en cas de perte de connexion.

➤ Postgres



*Figure 4.1-4 : SGBD Postgres*

PostgreSQL est un système de gestion de base de données relationnelle open source connue pour sa robustesse, sa flexibilité et ses capacités avancées. Il prend

en charge les types de données complexes, les transactions ACID, les requêtes SQL avancées et une conformité aux normes relationnelles.

#### 4.1.2 Outils utilisés

➤ Docker



*Figure 4.1-5: Plateforme Docker*

Docker est une plateforme de conteneurisation qui permet de créer, déployer et exécuter des applications dans des conteneurs portables. Les conteneurs encapsulent le code et ses dépendances, assurant une cohérence sur différents environnements. Docker simplifie la gestion des environnements en isolant les applications et en facilitant leur déploiement.

➤ Git et Gitlab



*Figure 4.1-6 : Plateforme Gitlab*

Git est le système de gestion de versions décentralisé le plus populaire, offrant un suivi et une organisation optimisés des projets. GitHub est une plateforme web utilisant Git pour héberger et gérer le développement de logiciels. Il propose des fonctionnalités comme le contrôle d'accès, la collaboration et la gestion des tâches.

➤ Postman



*Figure 4.1-7 : Logiciel Postman*

Postman est un logiciel permettant de développer et de tester des API REST dans les plus fins détails, grâce à une interface ergonomique et intuitive. Il nous permet principalement de tester des APIs en émettant des requêtes HTTP spécifiques (GET, POST, PUT, DELETE etc.) sur les différents endpoints de notre API.

➤ IntelliJ IDEA



*Figure 4.1-8 : Editeur IntelliJ*

C'est un éditeur de code gratuit, défini et optimisé pour la création et le débogage d'applications web, mobile, cloud etc. IntelliJ supporte des langages de programmation tels que Java, JavaScript, type script et bien d'autres



## 4.2 Résultat

Dans cette section il est question de présenter d'une part, les technologies qui sont intervenues et d'autre part au travers d'un scénario de capture d'écran comment fonctionnent nos micro-services et comment est-ce qu'ils répondent aux besoins présentés. Pour cela, nous décrivons principalement 03 scénarios d'utilisation :

- La documentation de notre API Rest ;
- L'utilisation de RabbitMQ pour la publication des messages ;
- L'utilisation de PostGIS pour la création de la carte dans le module de géolocalisation et l'enregistrement de l'emplacement des pharmacies ;
- L'enregistrement des commandes dans Redis ;

### 4.2.1 Documentation de l'API REST du module de Pharmacie

La documentation de l'API REST est réalisée avec OpenAPI. En effet OpenAPI est un langage de spécification pour API http.



*Figure 4.2-1 : Framework OpenApi*

La documentation du micro-service Pharmacie est la suivante :

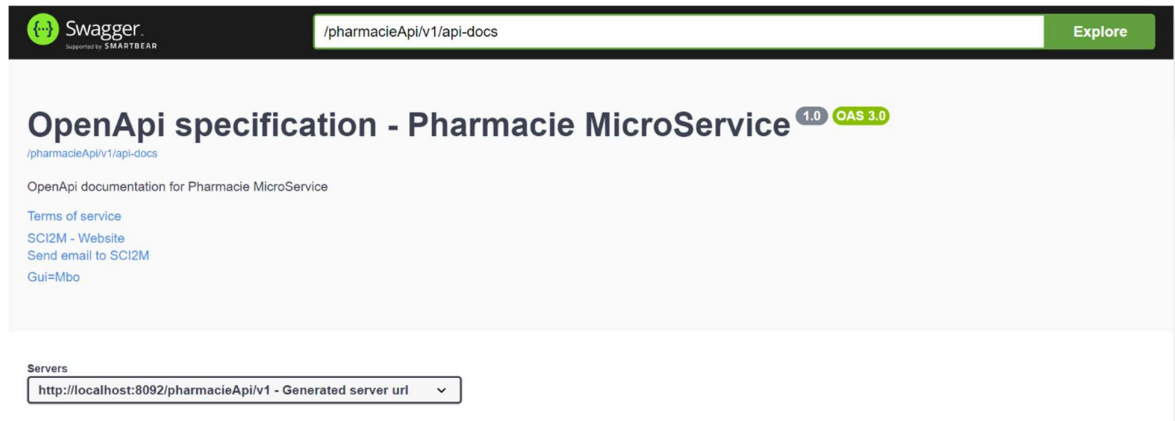


Figure 4.2-2 : Documentation de l'api du microsevice Pharmacy

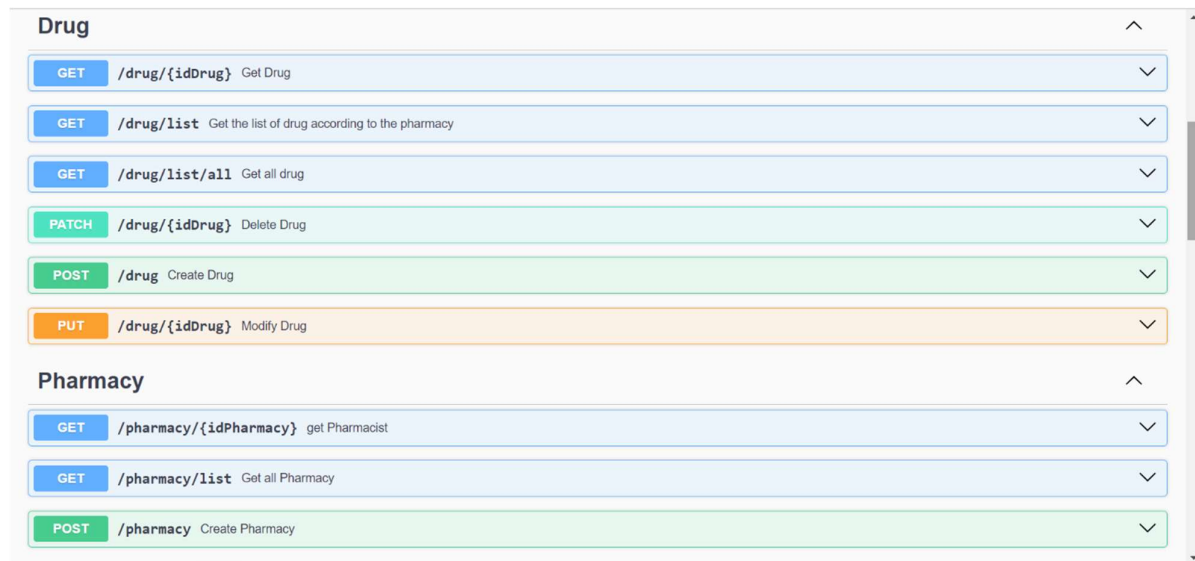


Figure 4.2-3 : Documentation de l'api du micro-service Pharmacy

### 4.2.1 Test d'un endpoint

Notre test d'endpoint s'effectue avec Postman : cet endpoint est celui qui permet au pharmacien de faire une proposition de médicament au client.

Nous entrons la liste des médicaments en paramètres et en variable d'url on passe l' identifiant du client et celui du pharmacien et on reçoit en retour la proposition avec la liste des médicaments et les prix correspondant à la pharmacie où est enregistré le pharmacien.

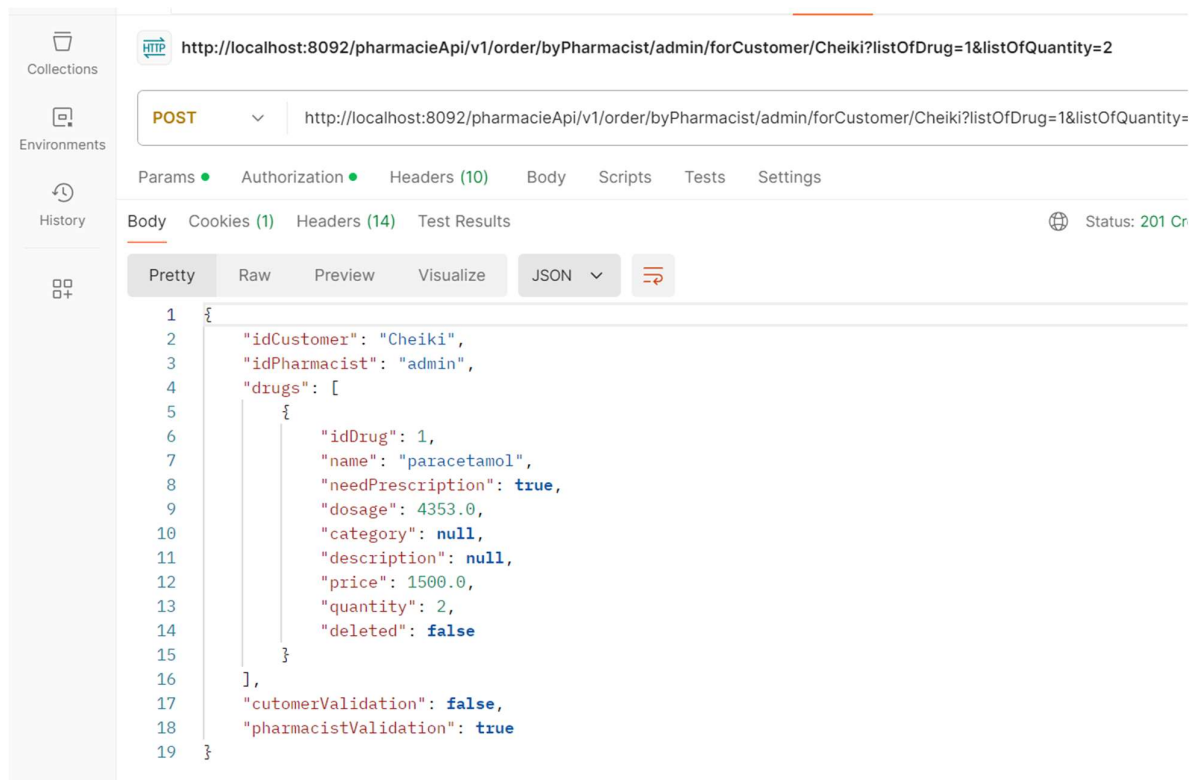


Figure 4.2-4 : Test d'API via Postman

Pour que l'action « **proposer des médicaments** » puisse aboutir il faudrait que plusieurs fonctions s'exécutent correctement à savoir : l'enregistrement de la proposition dans le cache ( Redis ) et l'envoi de la proposition au micro-service de messagerie par l'intermédiaire de Rabbit.

## 4.2.2 Publication des messages avec RabbitMQ



Figure 4.2-5 : Distributeur RabbitMQ

*RabbitMQ est un système open source de gestion de files d'attente de messages qui utilise le protocole AMQP. Il facilite la communication asynchrone en stockant et transmettant les messages de manière fiable, avec des fonctionnalités telles que la persistance, les accusés de réception, et la haute disponibilité.*

Il est essentiel dans notre architecture car il permet une communication asynchrone entre les services (Pharmacie → messagerie ; Commande → Notification). Il facilite la transmission de messages de manière fiable et distribué. Le serveur Rabbit reçoit le message ( la proposition de médicament ) qu'il enverra dans la queue appropriée pour la partager avec le micro-service de messagerie qui lui à son tour permet au client par l'intermédiaire d'un évènement en particulier récupérera la proposition et ce en temps réel.

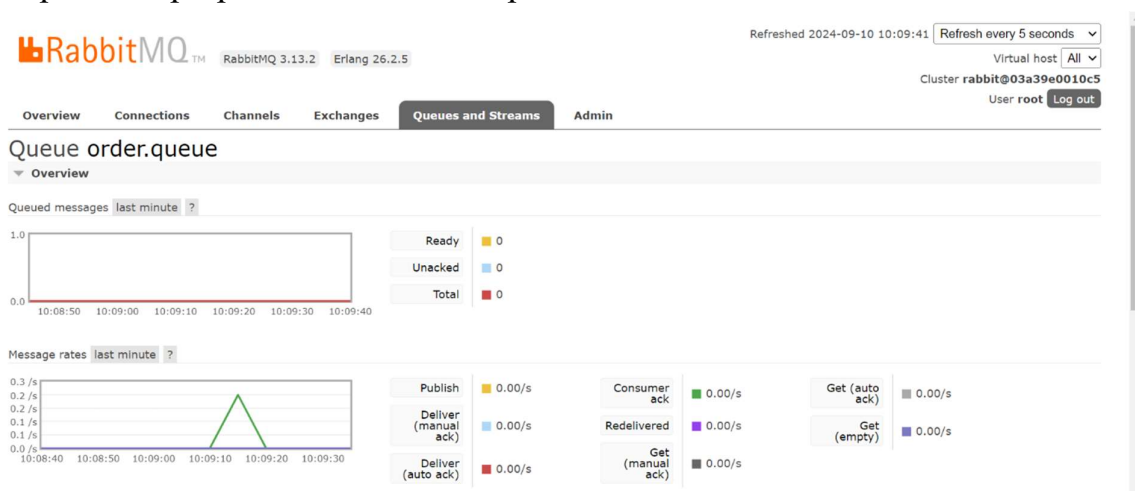


Figure 4.2-6 : Distribution d'un message via RabbitMQ

### 4.2.3 L'enregistrement de la commande dans le cache



Figure 4.2-7: Base de données Redis

*Redis est une base de données en mémoire, clé-valeur, utilisée principalement comme cache ou pour la gestion des sessions. Elle est prisée pour ses performances rapides et sa capacité à gérer des structures de données complexes. Redis sert aussi pour les files d'attente de messages et les opérations de publication/abonnement.*

Dans notre projet il nous permet d'enregistrer les propositions soit du Pharmacien ou les choix du client sur une durée de 24h.

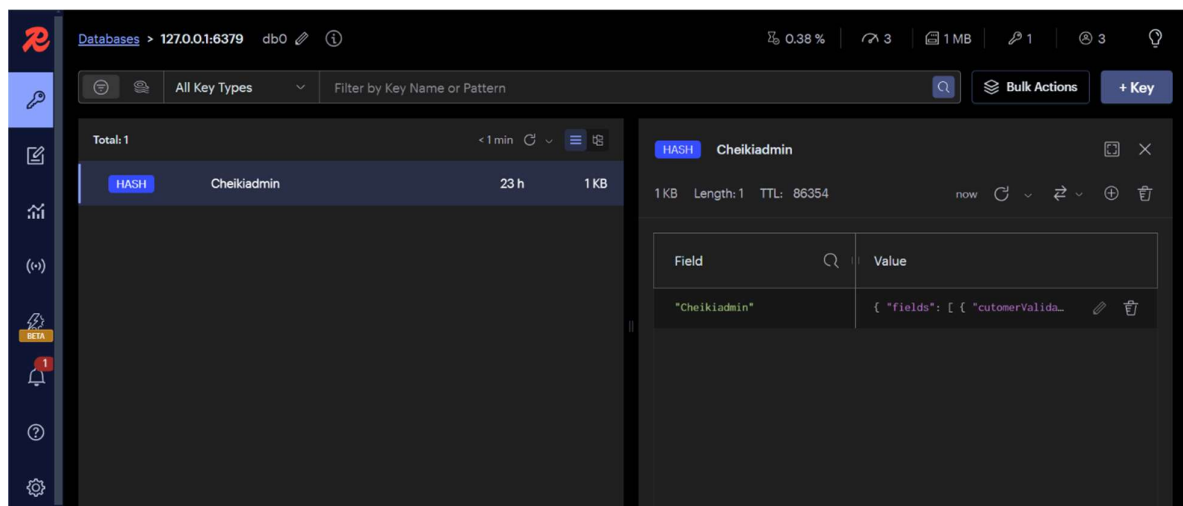


Figure 4.2-8 : Interface d'exécution de Redis

#### 4.2.4 L'enregistrement des pharmacies et le calcul des distances entre elles



Figure 4.2-9: Extension PostGIS

*PostGIS est une extension de PostgreSQL pour la gestion des données géospatiales. Elle permet de stocker, interroger et manipuler des données géographiques avec des fonctionnalités pour le traitement spatial, telles que les calculs de distance et les analyses spatiales. PostGIS est utilisé dans les applications de cartographie et les systèmes d'information géographique (SIG).*

PostGIS nous permet dans notre projet final d'enregistrer l'emplacement exacte des pharmacies et de savoir par rapport à une coordonnées (x ; z) qu'elle

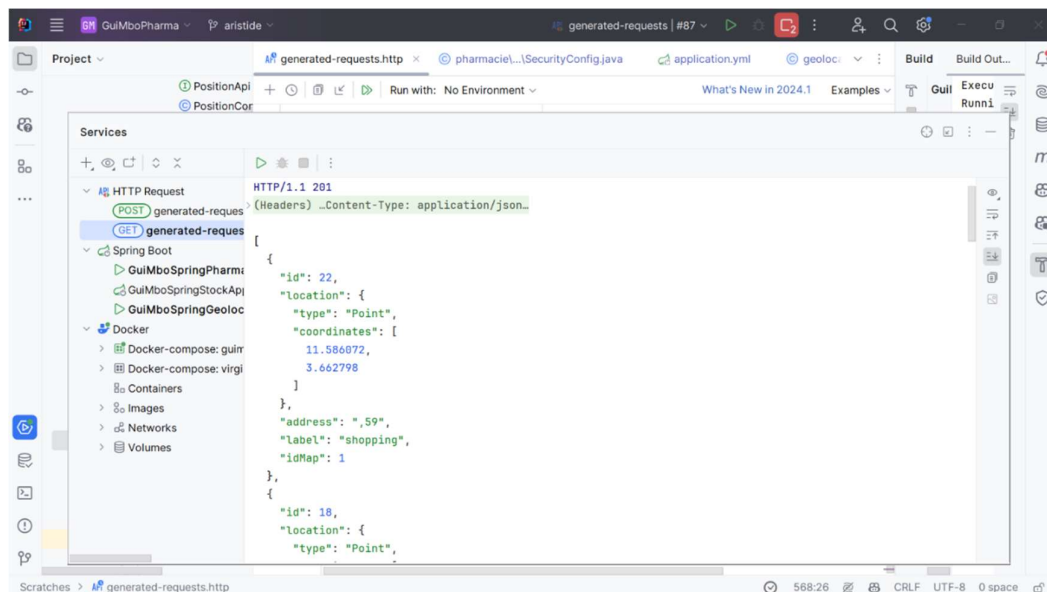


Figure 4.2-10 : Modèle de réponse Postgis

est la pharmacie la plus proche d'elle , de savoir également quelle sont les pharmacies dans un rayon de y km.

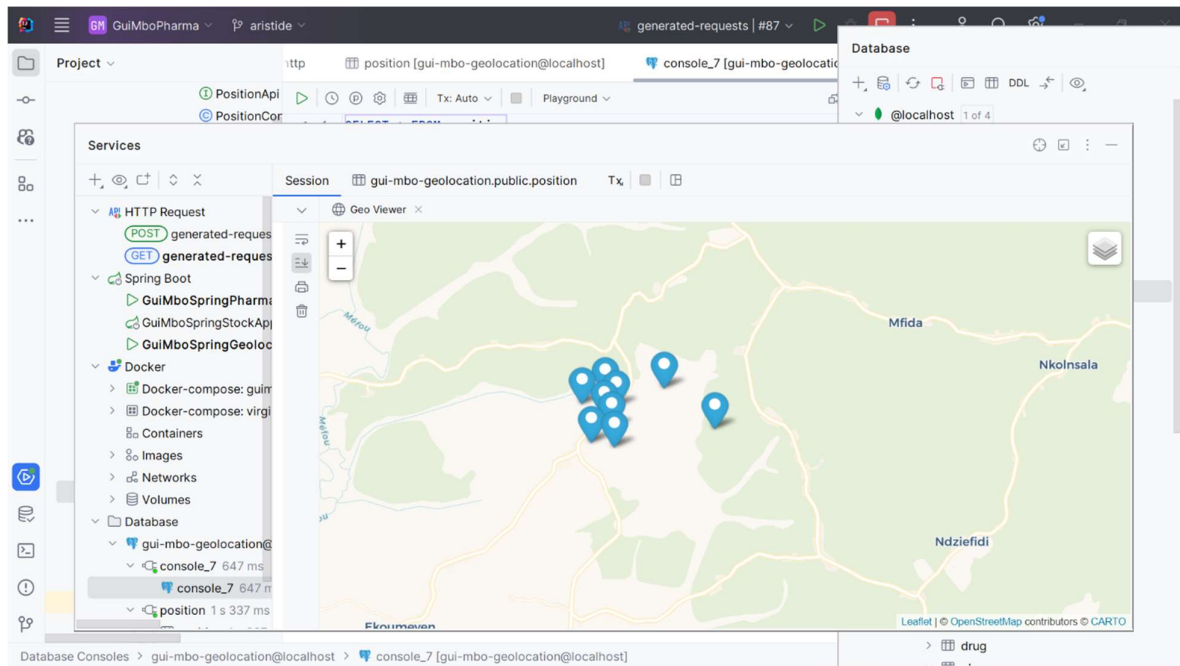


Figure 4.2-11 : Carte représentant l'emplacement des pharmacies

Sur l'image au dessus vous pouvez voir la position géographique de plusieurs pharmacies enregistrées au préalable.

➤ Redux



Figure 4.2-12: Bibliothèque Redux

*Redux est une bibliothèque JavaScript pour la gestion d'état, souvent utilisée avec React. Elle centralise l'état dans un store unique et immuable, facilitant la synchronisation des différentes parties de l'application. Redux suit un modèle de flux de données unidirectionnel pour rendre les mises à jour d'état prévisibles et faciles à suivre.*

Redux dans notre projet nous a permis de sauvegarder de manière locale ( au niveau du client web ) son panier de sélection ( les médicaments qu'il désire finalement acheter )



## CONCLUSION

L'objectif de mon stage au sein de la SCI 2M était de concevoir et d'implémenter une marketplace pharmaceutique intégrée à l'application GUI-MBÔ. Ce projet visait à optimiser les services offerts aux clients, notamment dans la dématérialisation des tâches courantes tout en adoptant une architecture micro-services pour garantir la flexibilité et l'évolutivité du système.

Durant ces 16 semaines de stage, j'ai eu l'opportunité d'aborder différentes phases du projet, allant de la prise en main des besoins fonctionnels à l'implémentation concrète des micro-services. J'ai notamment travaillé sur la refonte des interfaces de GUI-MBÔ, la conception de services comme la gestion de pharmacie, et l'intégration de nouvelles technologies comme PostGIS pour la géolocalisation et RabbitMQ pour la gestion des messages asynchrones. Il m'a également permis de mieux comprendre ce qu'était l'architecture micro-services et les bonnes pratiques pour mettre en place des systèmes complexes. Sur le plan professionnel, j'ai renforcé ma capacité à collaborer en équipe et à proposer des solutions face à des problématiques techniques.

Notre solution permettra aux clients de SCI 2M d'améliorer leurs confort et contribue à l'évolution de SCI 2M. Toutefois il sera nécessaire de toujours revoir le système afin de le mettre à jour régulièrement fonction des nouveaux besoins et de l'évolution des technologies.

## BIBLIOGRAPHIE

### 1. How to implement webSocket

- <https://medium.com/@shubhsalunkhe4199/guide-to-implement-web-sockets-in-angular-d8ce2b01abd4>

### 2. All about PostGIS (Introduction, how to implement geospatial)

- <https://medium.com/symphonyis/introduction-to-postgis-88fad193c684>
- <https://medium.com/@AlexanderObregon/how-to-implement-geospatial-queries-in-spring-boot-with-postgresql-bce52e7ffafa>
- [https://www.youtube.com/watch?v=UuNmwlHDT\\_E&t=886s&pp=ygUGc3ByaW5n](https://www.youtube.com/watch?v=UuNmwlHDT_E&t=886s&pp=ygUGc3ByaW5n)
- <https://www.baeldung.com/java-geospatial-applications>
- <https://www.baeldung.com/hibernate-spatial>

### 3. Swagger like a pro + Documentation

- <https://medium.com/@berktorun.dev/swagger-like-a-pro-with-spring-boot-3-and-java-17-49eed0ce1d2f>
- <https://www.baeldung.com/swagger-2-documentation-for-spring-rest-api>

### 4. Handling Exception and Error

- <https://www.baeldung.com/exception-handling-for-rest-with-spring>
- <https://medium.com/@ovinduuu/validation-and-exception-handling-in-spring-boot-ea226861938c>

### 5. Start with Redis

- <https://medium.com/simform-engineering/spring-boot-caching-with-redis-1a36f719309f>
- <https://www.baeldung.com/spring-data-redis-tutorial>

6. Tutorial for RabbitMQ and AMQP

- <https://www.rabbitmq.com/tutorials/tutorial-one-spring-amqp>
- <https://www.baeldung.com/rabbitmq/>
- <https://www.baeldung.com/spring-amqp/>

7. Tutorial for testing: <https://www.baeldung.com/spring-boot-testing>

8. Database auditing JPA: <https://www.baeldung.com/database-auditing-jpa>

9. All about DDD: <https://medium.com/@karahanzen/all-you-need-to-know-about-domain-driven-design-9d06c5234990>

## TABLE DES MATIERES

<b>Remerciements .....</b>	<b>i</b>
<b>Liste des Figures.....</b>	<b>iv</b>
<b>Listes des tables .....</b>	<b>vi</b>
<b>Abréviations.....</b>	<b>vii</b>
<b>Avant-propos .....</b>	<b>viii</b>
 <b>Introduction générale .....</b>	 <b>1</b>
 <b>Chapitre 1 Présentation de l'entreprise et déroulement du stage .....</b>	 <b>2</b>
1.1 Historique et emplacement.....	3
1.1.1 Historique .....	3
1.1.2 Valeurs et objectif de SCI 2M .....	3
1.1.2.1 Valeurs.....	4
1.1.2.2 Objectifs .....	4
1.1.3 Emplacement .....	4
1.2 Organisation de SCI 2M.....	6
1.3 Déroulement du stage.....	8
1.3.1 Activités menées.....	8
1.3.2 Difficultés rencontrées.....	10
 <b>Chapitre 2 Présentation du problème .....</b>	 <b>11</b>
2.1 Identification et origine du problème .....	12
2.2 Solution existante et limites .....	12
2.2.1 Pharmacie Lafayette .....	12
2.2.2 CVS Pharmacie.....	13
2.2.3 HealthKart .....	13
2.2.4 Medindex .....	14
 <b>Chapitre 3 Solution proposée et Méthodologie .....</b>	 <b>15</b>

3.1	Description de la solution.....	16
3.2	Méthodologie et Environnement de travail.....	16
3.2.1	Méthodologie de travail.....	16
3.2.2	Environnement de travail.....	18
3.3	Analyse, Conception et Implémentation .....	18
3.3.1	Analyse .....	19
3.3.1.1	Besoins fonctionnels .....	19
3.3.1.2	Besoins non-fonctionnels .....	19
3.3.1.3	Diagramme cas d'utilisation .....	21
3.3.1.4	Diagramme de classe.....	24
3.3.2	Conception de la solution .....	25
3.3.2.1	Architecture logique.....	25
3.3.2.2	Diagramme de composant.....	27
3.3.2.3	Diagramme de séquence .....	29
<b>Chapitre 4</b>	<b>Implémentation et résultat.....</b>	<b>32</b>
4.1	Outils et choix technologiques .....	33
4.1.1	Choix technologiques .....	33
4.1.2	Outils utilisés .....	35
4.2	Résultat.....	37
4.2.1	Documentation de l'API REST du module de Pharmacie .....	37
4.2.1	Test d'un endpoints.....	39
4.2.2	Publication des messages avec RabbitMQ .....	40
4.2.3	L'enregistrement de la commande dans le cache .....	41
4.2.4	L'enregistrement des pharmacies et le calcul des distances entre elles....	42
<b>Conclusion</b>	<b>.....</b>	<b>45</b>
<b>Bibliographie</b>	<b>.....</b>	<b>46</b>
<b>Annexes</b>	<b>.....</b>	<b>48</b>