



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Benyoucef BENKHEDDA- Alger1

Faculté des Sciences

Département d'Informatique

Mémoire de Licence

en Informatique

Spécialité :

*Système informatique/Ingénierie des systèmes
d'information et logiciels*

Thème

Conception et réalisation d'un système web de
recherche et géolocalisation des pharmacies de
garde

Encadré par

- Mr. DJAAFRI Ilyes

Réalisé par

- ATBANE Mounir

- BENMEBAREK Rafik

- BERBACHI Djihane Neila Nour El Houda

Remerciements

D'abord, nous adressons nos sincère remerciements à Monsieur "Ilyes Djaafri", qui nous a fait l'honneur de diriger notre travail. C'est avec un grand respect que nous adressons notre remerciement à tous nos enseignants, pour leurs conseils et leurs remarques. Nous tenons aussi à remercier sincèrement tous nos camarades au sein de l'université. Nous ajoutant à nos remerciements, tous qui nous ont encouragés et nous aidés.

Dédicaces

Je dédie ce travail tout d'abord,

A mes parents bien-aimés afin de leur témoigner ma profonde gratitude,

A ma mère qui a été ma source d'inspiration et qui m'a donnée de la force quand j'ai pensé à abandonner,

A mon père qui m'apporte continuellement son soutien moral, spirituel, émotionnel et financier,

A mon grand-père paternel qui a toujours été présent pour moi quand j'ai eu besoin de lui,

A Islam mon petit frère pour être le meilleur des frères,

A toute ma famille,

A mes meilleures amies qui m'ont soutenue tout au long de ces trois dernières années et dont l'aide et la bonne humeur m'ont été très précieuses,

A mon trinôme sans lequel ce travail n'aurait pas vu le jour,

Spéciale dédicace à mon ami Rafik BENMEBAREK qui m'a permis de faire partie de ce projet.

BERBACHI Djihane Neila Nour El Houda

Dédicaces

Je dédie ce mémoire :

A mes professeurs qui m'ont diffusé leur savoir dans la patience et la pédagogie et

A mon frère qui m'encourageait et m'inspirait.

ATBANE Mounir

Dédicaces

Je dédie cet humble ouvrage à mes grands-parents paternels, Kamir et Abderahman paix à leurs âmes,

Mes grands-parents maternels Mohamed et Louisa, que dieu les garde parmi nous,

Mes chers parents, qui ont tant fait pour moi, en faisant tout ce qui est en leur pouvoir pour me soutenir,

A mon trinôme qui ont été des partenaires très actifs et agréables durant la réalisation de ce projet,

BENMEBAREK Rafik

Résumé

Notre projet se concentre sur l'étude, la conception et la réalisation d'un système web, qui permet la géolocalisation et la recherche de pharmacies de garde.

Les objectifs majeurs de ce site sont : la possibilité de rechercher les pharmacies ouvertes les plus proches à partir d'une position donnée ou en géolocalisant l'utilisateur, de chercher une pharmacie par affiliation aux systèmes de sécurité sociale, de faciliter la tâche de recherche de médicaments.

Ce présent document, décrit le projet et son contexte et résume le déroulement de toutes les étapes nécessaires à sa réalisation

Mots clés : internet, Web, base de données, position, géolocalisation, GPS, GSM, GMap, Pharmacie . . .

Abstract

Our project focusses on the study, conception and realization of a web system that allows the geolocation and the research of all-night drugstore.

The main drives of this web site are: the possibility of searching the nearby open drugstore from a given location or by geolocating the user, searching a drugstore by affiliation to social security systems and to ease the research of medicines.

This draft describes the project, its context, and summarizes all the necessary steps leading to its realization.

Key Words: internet, Web, database, Geolocation, position, GPS, GSM, GMap, pharmacy...

Table des matières

Liste des figures	3
Liste des tableaux	4
Liste des abréviations.....	5
Introduction générale	6
Problématique.....	6
Objectif.....	7
1. Etat de l'art.....	8
1.1. Introduction.....	8
1.2. Définitions	8
1.3. Techniques de géolocalisation	9
1.4. Etude de l'existants	14
1.5. Media Queries.....	16
1.6. Architecture Client-Serveur.....	19
1.7. Conclusion	20
2. Analyse des besoins	21
2.1. Introduction	21
2.2. Modélisation	21
2.3. Acteurs et modélisation du contexte	22
2.4. Identification des besoins.....	23
2.5. Conclusion.....	26
3. Conception	28
3.1. Introduction	28
3.2. Diagramme de séquence	28
3.3. Diagramme de classe	40
3.4. Passage au modèle relationnel.....	44
3.5. Conclusion	45
4. Réalisation	46
4.1. Introduction.....	46
4.2. Design Pattern MVC ^[25]	46
4.3. Implémentation et outils utilisés	48
4.4. Frameworks.....	49
4.5. APIs	50

4.6. Outils de développement	51
4.7. Captures d'écran	52
4.8. Conclusion	58
Conclusion générale et perspectives	59
Webographie.....	60
1. Annexe.....	i
1.1. UML	i
1.2. Règles de passage au modèle relationnel ^[33]	iii
1.3. Diagramme de séquence « Authentification »	iv

Liste des figures

FIGURE 1 : SCHEMA DE PRINCIPE DE LA GEOLOCALISATION PAR GPS ^[3]	9
FIGURE 2 : REPERAGE D'UNE POSITION PAR SATELLITES ^[6]	10
FIGURE 3 : SCHEMA REPRESENTANT LE PRINCIPE DU EOTD ^[7]	11
FIGURE 4 : SCHEMA REPRESENTANT LE PRINCIPE DU CELL-ID ^[9]	11
FIGURE 5 : SCHEMA REPRESENTANT LE PRINCIPE DE LA TRIANGULATION ^[10]	12
FIGURE 6 : APPROCHE DE GEOLOCALISATION PAR WI-FI ^[12]	13
FIGURE 7 : EXEMPLE DE GEOLOCALISATION PAR ADRESSE IP ^[14]	13
FIGURE 8 : INTERFACE DE L'APPLICATION PLYCE ^[15]	14
FIGURE 9 : INTERFACE DE L'APPLICATION YASSIR ^[16]	15
FIGURE 10 : INTERFACE DE L'APPLICATION WAZE	15
FIGURE 11 : INTERFACE DE LOCALISATION D'UN PHARMACIEN SUR WINPHARM ^[17]	16
FIGURE 12 : EXEMPLE D'UTILISATION DES MEDIA QUERIES ^[18]	17
FIGURE 13: SCHEMA DESCRIPTIF DE L'ARCHITECTURE CLIENT-SERVEUR	19
FIGURE 14 : DIAGRAMME DE CAS D'UTILISATION DU SYSTEME	25
FIGURE 15 : DIAGRAMME DE SEQUENCE "RECHERCHER PHARMACIE"	31
FIGURE 16 : DIAGRAMME DE SEQUENCE "TRANSMETTRE ORDONNANCE"	33
FIGURE 17 : DIAGRAMME DE SEQUENCE "SOUMETTRE COMMENTAIRE"	35
FIGURE 18 : DIAGRAMME DE SEQUENCE "CREER COMPTE PHARMACIEN"	38
FIGURE 19 : DIAGRAMME DE SEQUENCE "GERER GARDES"	39
FIGURE 20 : DIAGRAMME DE CLASSE PHARMAP	41
FIGURE 21 : MODELE RELATIONNEL PHARMAP	45
FIGURE 22 : FONCTIONNEMENT DU PATTERN MVC ^[26]	47
FIGURE 23:PAGE D'ACCEUIL PHARMAP	52
FIGURE 24: MENU D'OPTIONS	53
FIGURE 25: LISTE DES RESULTATS.....	54
FIGURE 26: INSCRIPTION CLIENT	55
FIGURE 27: FORMULAIRE D'INSCRIPTION POUR UN COMPTE PHARMACIEN	56
FIGURE 28: CONNEXION.....	57
FIGURE 29: COMPTE PHARMACIEN	58
FIGURE 30 : HISTORIQUE SUR L'UML ^[30]	I
FIGURE 31 : LA HIERARCHIE DES DIAGRAMMES UML 2.0 SOUS FORME D'UN DIAGRAMME DE CLASSES ^[32]	II

Liste des tableaux

TABLEAU 1 : TABLEAU DE DESCRIPTION DES CLASSES

41

Liste des abréviations

API	Application Programming Interface
IHM	Interface Homme Machin
GNSS	Global Navigation Satellite System
UMTS	Universal Mobile Telecommunications System
LTE	Long Term Evolution
GPS	Global Positioning System
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
RFID	Radio Frequency Identification
SIG	Système d'Information Géographique
EOTD	Enhanced Observed Time Difference
IANA	Internet Assigned Numbers Authority
MAC	Media Access Control
SSID	Service Set Identification
IP	Internet Protocol
UML	Unified Modeling Language
XML	Extensible Markup Language
PC	Personal Computer
GUI	Graphical User Interface,
HTTP	Hyper Text Transfer Protocol
W3C	World Wide Web Consortium
GMap	Google Maps
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
SCSS	Sassy Cascading Style Sheet
SASS	Syntactically Awesome Stylesheets
PHP	Hypertext Preprocessor
SQL	Structured Query Language

Introduction générale

Depuis quelques années, la géolocalisation s'est démocratisée et est très utilisée notamment pour la navigation routière.

Les technologies dans ce domaine n'ont cessé d'évoluer, surtout avec l'apparition des APIs tels que Google Maps, Bings Maps, OpenStreetMap, W3C et bien d'autres. Leur mise à disposition et l'existence des nombreux outils libres a rendu la géolocalisation plus accessible. Les systèmes développés sont utilisés comme des applications de promotion touristique, de calcul d'itinéraire, de recherche géolocalisée d'activités ou de ressources d'un territoire. Ainsi l'intérêt du grand public envers la géolocalisation s'est agrandi ^[1].

Notre travail consiste à concevoir et réaliser un système web de recherche et géolocalisation des pharmacies de garde, pour un utilisateur équipé d'un Smartphone ou d'un ordinateur, connecté à internet.

Afin de réaliser et atteindre notre objectif nous avons divisé le travail en quatre chapitres comme suit :

- **Chapitre 1** : Etat de l'art
- **Chapitre 2** : Analyse des besoins
- **Chapitre 3** : Conception du système
- **Chapitre 4** : Réalisation du système

Problématique

Pouvoir localiser une pharmacie ouverte à une heure tardive du soir n'est pas toujours une chose évidente, et surtout en cas d'urgence. C'est pour cela que nous voulons réaliser un système web qui permet de chercher et de localiser les pharmacies de garde les plus proches.

Objectif

L'objectif visé par cette étude est de proposer aux utilisateurs une plateforme cartographique en ligne capable de localiser les pharmacies de gardes les plus proches. Cette plate-forme devra aussi leur permettre en outre de les localiser à n'importe quel moment de la journée. Aussi, elle pourra servir de moyen de partage d'avis entre les clients, et permettra de faciliter la recherche de médicaments.

Chapitre 1

1. Etat de l'art

1.1. Introduction

Il est indispensable avant de se lancer dans un projet quelconque de bien étudier et analyser son environnement, dans ce chapitre nous allons parler de la géolocalisation ainsi que des techniques utilisées dans ce cadre.

1.2. Définitions

1.2.1. La géolocalisation

La géolocalisation est le procédé d'identification et de positionnement d'un objet, une personne ou un véhicule sur une carte à l'aide de ses coordonnées géographiques.

Cette opération est réalisée à l'aide d'un terminal capable d'être localisé grâce à un système de positionnement par satellites et un récepteur GPS par exemple, ou par d'autres techniques ; de plus, le terminal est en mesure de publier, en temps réel ou de façon différée, ses coordonnées géographiques latitude/longitude. Les positions enregistrées peuvent être stockées au sein du terminal et être extraites ultérieurement, ou être transmises en temps réel vers une plateforme logicielle de géolocalisation. La transmission en temps réel nécessite un terminal équipé d'un moyen de télécommunication de type GSM/GPRS, radio ou satellite lui permettant d'envoyer les positions à des intervalles réguliers ce qui permet la visualisation de la position du terminal au sein d'une carte sur une plateforme accessible depuis internet ^[2].

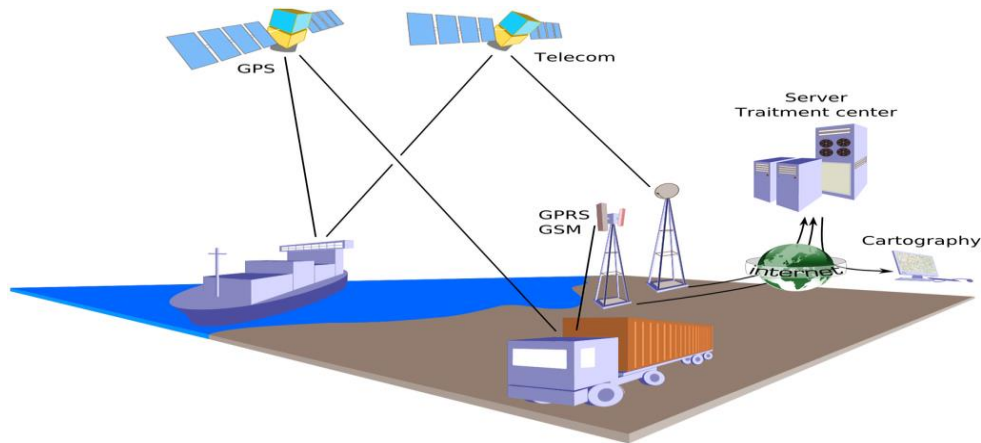


Figure 1 : Schéma de principe de la géolocalisation par GPS [3]

1.2.2. Système d'information géographique

Un système d'Information Géographique (SIG) est un outil informatique permettant de représenter et d'analyser toutes les données qui existent sur terre ainsi que tous les événements qui s'y produisent.

Les SIG offrent toutes les possibilités des bases de données (telles que requêtes et analyses statistiques) et ce, au travers d'une visualisation unique et d'analyse géographique propres aux cartes. Ces capacités spécifiques font du SIG un outil unique, accessible à un public très large et s'adressant à une très grande variété d'applications [4].

1.3. Techniques de géolocalisation

Il existe plusieurs techniques de géolocalisation, chacune se basant sur des technologies spécifiques ayant leurs avantages et leurs inconvénients, nous pouvons en citer :

1.3.1. Géolocalisation par satellite

Le système de localisation par satellite fonctionne selon le principe de triangulation : la position de l'objet observé est calculée grâce aux signaux échangés entre le récepteur dont il est équipé et un ensemble d'au moins 3 satellites faisant partie d'une constellation. La mesure de temps de parcours de ces signaux permet d'identifier la

longitude, la latitude, et l'altitude de l'objet observé, avec une précision allant de 10 à 100 mètres.

Le réseau satellite de positionnement le plus connu est le GPS, bien que l'alternative européenne nommée Galileo soit en cours de déploiement [5].



Figure 2 : Repérage d'une position par satellites [6]

1.3.2. Géolocalisation par géocodeur

Le géocodage consiste à affecter des coordonnées géographiques (longitude/latitude) à une adresse postale. Ce procédé conduit à la mise en place de traitements automatisés de manière ponctuelle ou sur des fichiers d'adresses (individus, entreprises, points d'intérêt, etc.). Les coordonnées géographiques permettent de situer chaque adresse sur une carte numérique via un SIG [34].

1.3.3. Géolocalisation GSM

Il est aussi possible de faire de la Géolocalisation GSM qui utilise les antennes et les technologies des réseaux de téléphonie mobile, conçus pour la voix et le transfert de données, comme le GSM, l'UMTS ou le LTE. On y distingue 3 méthodes de localisation :

A. Différence de temps observé

Aussi appelée différentiel temps ou EOTD (Enhanced Observed Time Difference), le portable envoie un signal aux stations environnantes ; la plus proche lui renvoie le signal. C'est le temps écoulé entre l'émission et la réception du signal (le round-trip

delay time) qui permet à un ordinateur externe de déterminer la localisation du portable.

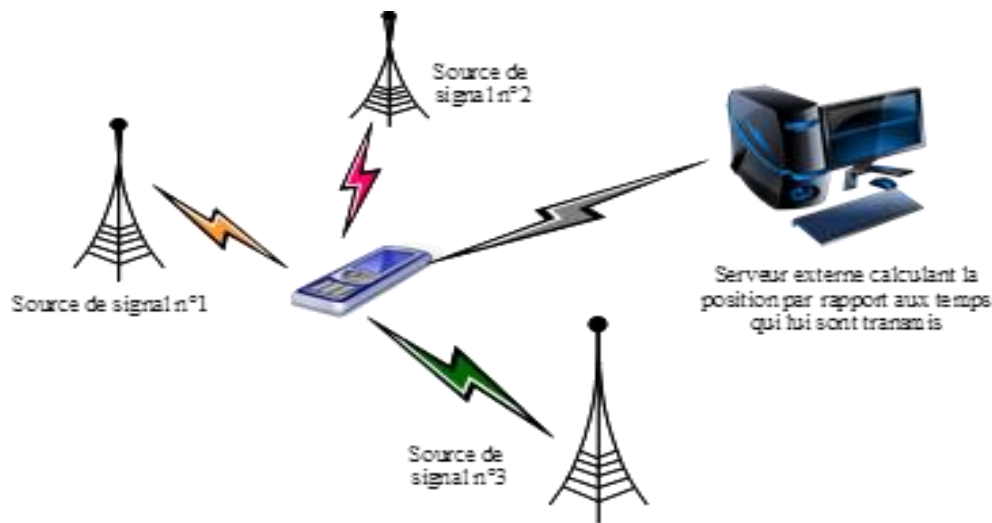


Figure 3 : Schéma représentant le principe du EOTD [7]

B. Cell-ID

Pratique, simple et économique, cette technologie permet de localiser l'utilisateur qui se situe dans une zone couverte par le réseau grâce à l'identification de la cellule de son antenne mobile. Cette technique est très rapide puisque la géolocalisation peut être effectuée en moins de 3 secondes. En revanche, elle est relativement imprécise puisqu'il y a une marge d'erreur de 200 à 300 mètres en ville lorsque le réseau est dense, sinon, la marge est de plus ou moins 10 kilomètres en campagnes ou en zone rurale [8].

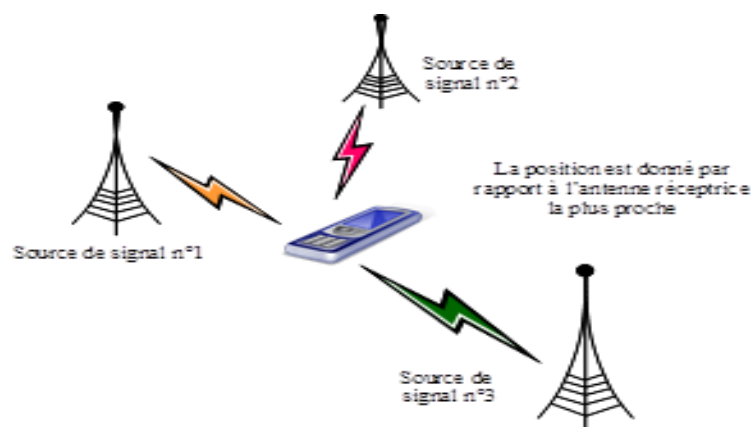


Figure 4 : Schéma représentant le principe du Cell-ID [9]

C. Triangulation

Cette technique de localisation par GSM fait appel à l'antenne hertzienne du téléphone mobile de l'utilisateur. Trois relais émetteurs et récepteurs envoient des informations qui sont ensuite traitées et analysées. Les relais changent en fonction des déplacements du téléphone portable. La triangulation est un peu plus longue, mais beaucoup plus précise et la marge d'erreur est relativement réduite :

- 120 à 130 mètres en ville lorsque le réseau est dense
- 5 kilomètres environ en campagne ou en zone rurale

Néanmoins, cette méthode nécessite l'installation d'un programme sur la carte SIM du portable.

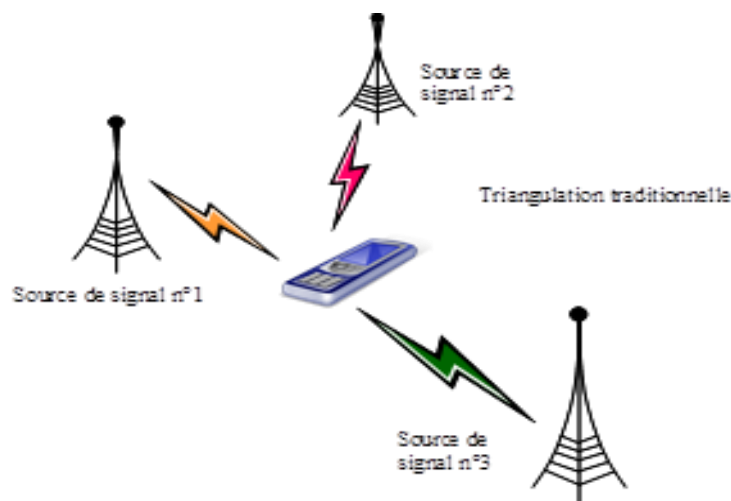


Figure 5 : Schéma représentant le principe de la triangulation ^[10]

1.3.4. Géolocalisation par Wi-Fi

De la même façon qu'un terminal GSM peut se localiser par la méthode du Cell ID sur un réseau mobile GSM, un terminal Wi-Fi peut utiliser la même méthode en se basant sur les identifiants des bornes Wi-Fi (SSID ou adresses MAC) qu'il détecte et en mesurant l'intensité du signal. Il existe des bases de données recensant une multitude d'identifiants de bornes d'accès Wi-Fi ainsi que leur position géographique. L'appareil alors, interrogera une base de données mondiale pour combiner l'empreinte du signal à sa position ^[11].

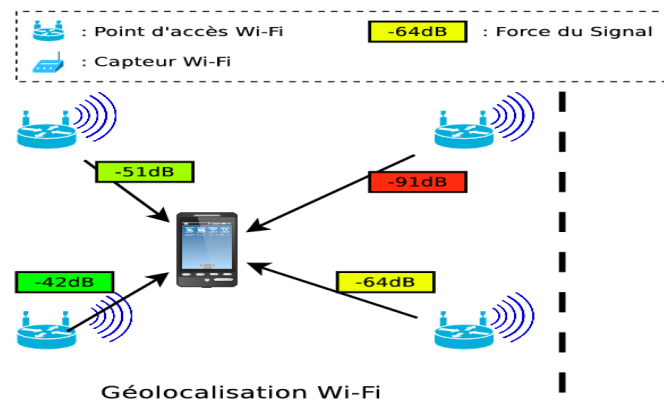


Figure 6 : Approche de géolocalisation par Wi-Fi [12]

1.3.5. Géolocalisation par adresse IP

C'est la méthode qui permet de déterminer la position géographique d'un terminal connecté à internet en se basant sur son adresse IP. Les adresses IP sont gérées par l'IANA, une organisation chargée de découper les blocs d'adresses IP disponibles et de les distribuer de façon très contrôlée aux pays qui en demandent. Toutes ces attributions étant très bien documentées, il est possible de savoir dans quel pays se trouve un terminal connecté à Internet grâce à son adresse IP. Un niveau de précision de l'ordre de la ville est possible.

Le principe est d'identifier à partir de l'adresse IP utilisée par un Internaute pour la comparer en temps réel à un référentiel d'adresses IP comprenant leur localisation. C'est le point d'accès technique de l'internaute qui est géolocalisé [13].

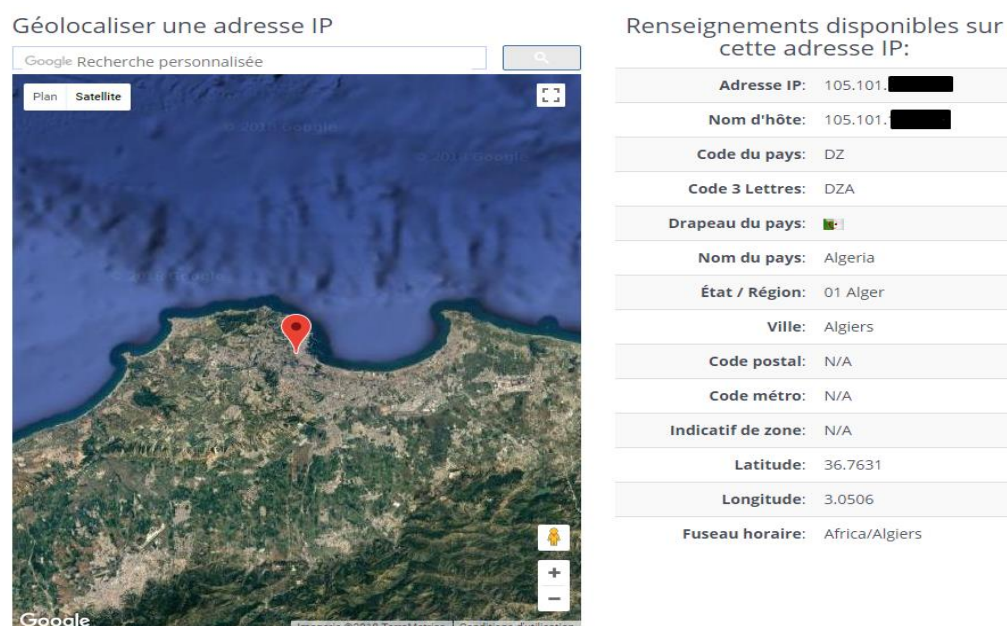


Figure 7 : Exemple de géolocalisation par adresse IP [14]

1.4. Etude de l'existants

Avant de se lancer dans la conception et la réalisation de notre application, il est primordial de d'abord s'informer sur les applications basées sur la géolocalisation. Cette dernière peut être exploitée dans plusieurs domaine tel que : la navigation routière, le tourisme, le commerce etc...

-Plyce : C'est une application faite pour le shopping. Plyce est une application mobile lancée en 2010 par une start-up française portant le même nom. Elle permet à ses utilisateurs de géolocaliser en temps réel les points de ventes à proximité de leur position, qui proposent des réductions et de bons plans. Plyce devient alors un allié de poids pour les personnes désirant faire quelques économies.



Figure 8 : Interface de l'application Plyce [15]

- **YAssir** : YAssir est un service de transport sur smartphone, permettant à ses utilisateurs de choisir un point de départ ainsi qu'une destination, cette requête est ensuite transmise au chauffeur le plus proche, et ce dernier reçoit aussi une indication sur le chemin le moins encombré pour se rendre à la destination souhaitée.

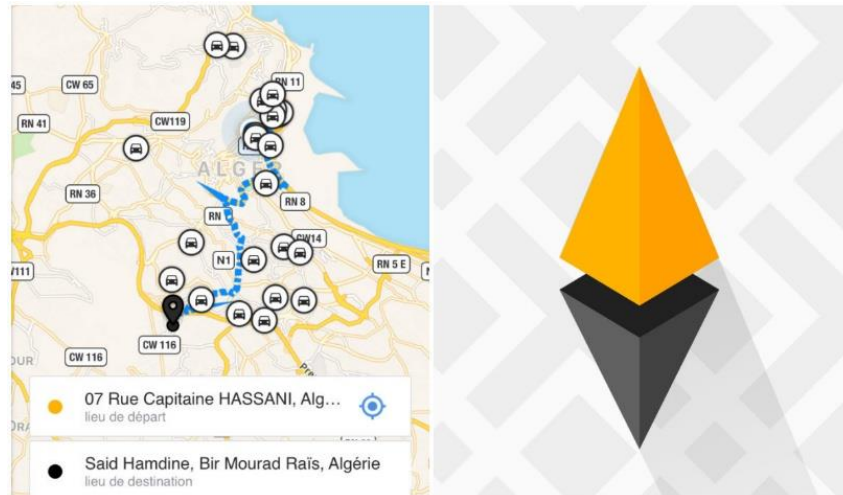


Figure 9 : Interface de l'application Yassir [16]

- **Waze** : Waze est une application de navigation routière, offrant aux conducteurs utilisant cette application d'avoir des informations en temps réel de l'état de du trafic et des routes de la part d'autres utilisateurs. Ce qui leurs permet alors, d'avoir un calcul du meilleur itinéraire pour une consommation en carburant amoindri.

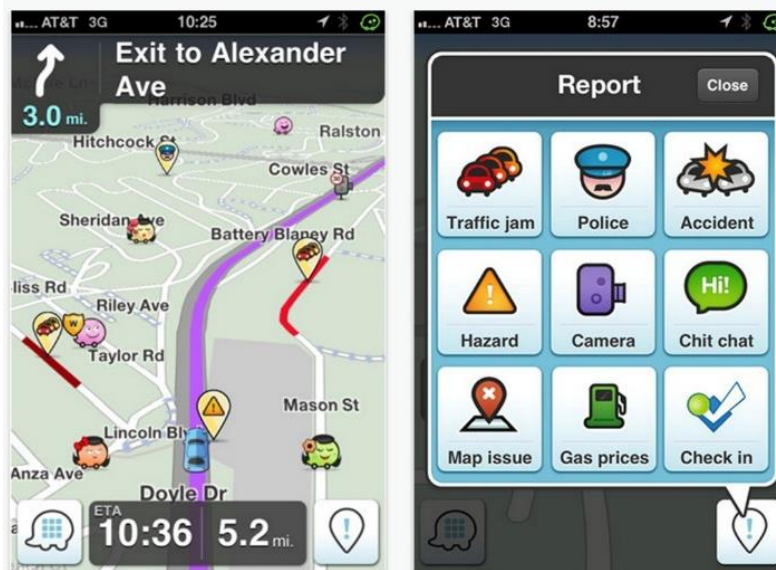


Figure 10 : Interface de l'application Waze

- **Winpharm** : Winpharm est une application créée par l'agence Victura le 1 mai 2016. Elle utilise la géolocalisation pour permettre aux patients de trouver les pharmacies les plus proches se trouvant autour d'eux, qu'elles soient de garde ou ouvertes aux horaires habituels et ce, n'importe où sur le territoire Algérien.

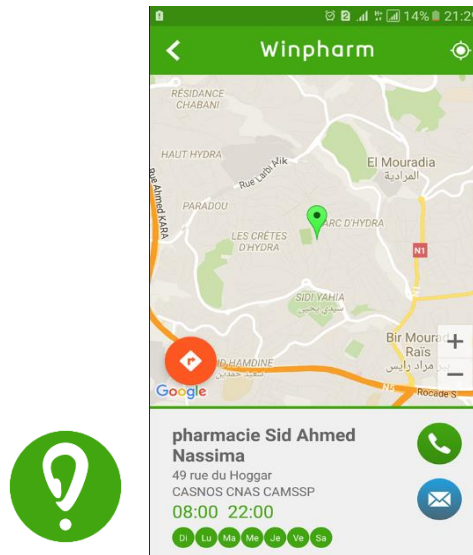


Figure 11 : Interface de localisation d'un pharmacien sur Winpharm^[17]

1.5. Media Queries

Une media query est une fonctionnalité HTML/CSS qui permet au contenu d'une page Web de s'adapter au type de média dans lequel la page est rendue, tel qu'un écran d'ordinateur ou celui d'un téléphone ou d'une tablette. Ceci est considéré comme une technologie de base pour la mise en œuvre de conception Web réactive et a été recommandé pour la mise en œuvre en tant que norme en Juin 2012 avec d'autres fonctionnalités CSS3.

Nous allons appliquer le concept des Media Queries sur notre projet car c'est un site web, et que son utilisation est beaucoup plus orientée pour les utilisateurs mobiles donc il a fallu adapter l'interface et la rendre la plus agréable et adaptée pour ce support, mais ses services restent quand même disponibles sur les ordinateurs.



Figure 12 : Exemple d'utilisation des Media Queries [18]

1.5.1. Comparatif des Responsive Design Frameworks [19]

A. Bootstrap

Framework créé pour les besoins de Twitter en 2011, il est le framework le plus utilisé dans le monde. Celui-ci inclut des éléments de CCS, HTML et Javascript.



Avantages :

- Une très grande communauté, permettant d'obtenir de l'aide ou consulter des documentations très facilement.
- Mit à jour régulièrement et toujours aussi populaire.
- Version téléchargeable modulable afin d'inclure ou non certains paquets, possibilité également depuis le fichier config.
- Supporte SASS et SCSS.
- Flexible.

Inconvénients :

- Plus lourd que les autres framework CSS.
- Demande à surcoucher pour ne pas se retrouver avec un site web des plus banals.

B. Foundation



Framework Open Source géré par ZURB.

Avantages :

- Du CSS par défaut pour ne pas avoir à démarrer de zéro.
- Moins populaire que Bootstrap, ce qui permet de créer un style unique de site web.
- Une grille flexible, permettant d'appliquer plus de personnalisation dans les déclarations de classes.
- Une bonne documentation en ligne, couplée à une aide et des cours proposés de la part du service technique de Foundation.
- Supporte le SCSS.

Inconvénients :

- Ne supporte pas le SASS.
- Demande beaucoup d'efforts pour une customisation du site complète.
- Grille de base peu modulable.

C. Materialize CSS



Framework Open source édité par Google, axé Material design et inspiré de Bootstrap.

Avantages :

- Inclut une série de composants JavaScript permettant de réaliser une multitude d'effets additionnels comme les vagues sur les boutons ou encore des effets parallaxe.
- Code simplifié sur les formulaires et les barres de navigation.
- Facile à prendre en main.
- Supporte le SCSS et le SASS.
- Flexible.
- Très léger.

- De très bonnes documentations.
- Permet le responsive.

Inconvénients :

- Mauvais support du SASS.
- Dans certaines situations la grille a du mal à s'adapter.

1.6. Architecture Client-Serveur

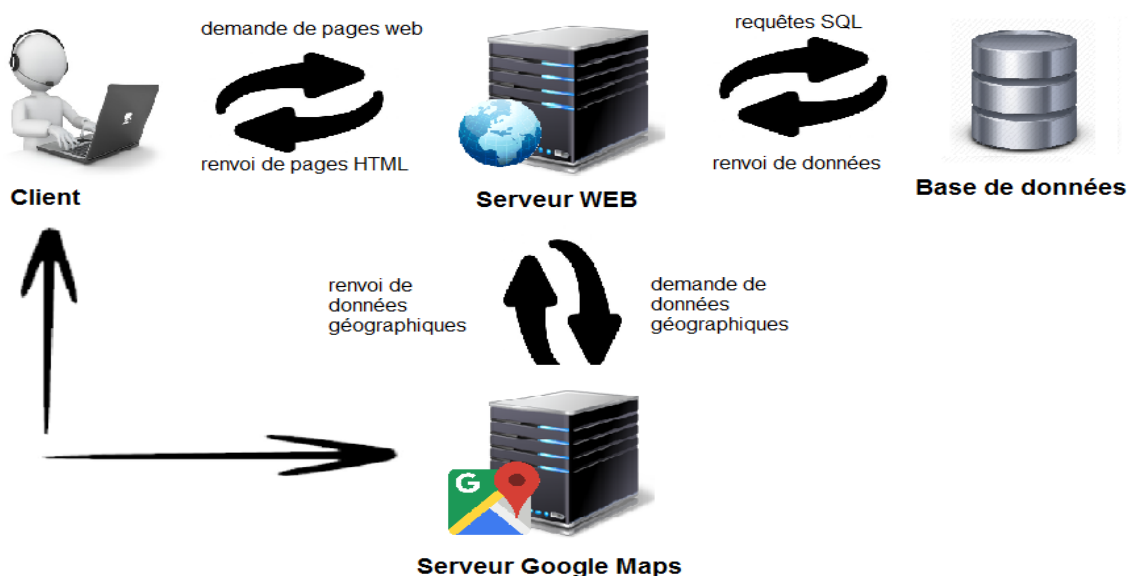


Figure 13: Schéma descriptif de l'architecture Client-Serveur

L'architecture client-serveur est une structure d'application distribuée qui répartit les tâches ou les charges de travail entre les fournisseurs d'une ressource ou d'un service, appelés serveurs, et les demandeurs de services, appelés clients. Les clients et les serveurs communiquent sur un réseau distinct, tout comme ils peuvent résider dans le même système.

Notre système est basé sur une architecture client/serveur, un client effectue une demande de service, dans notre système, le client cherche à localiser une pharmacie donc il initie le contact en lançant une requête de recherche auprès du serveur Google Maps le serveur reçoit la requête du client, effectue des calculs et répond au service demandé. Celui-ci peut répondre à plusieurs clients simultanément.

1.7. Conclusion

Dans ce chapitre nous avons parlé de la géolocalisation et des différentes techniques utilisées dans ce domaine, nous avons aussi présenté une étude des systèmes existants, afin de pouvoir exploiter leurs points forts et éviter leurs points faibles.

Chapitre 2

2. Analyse des besoins

2.1. Introduction

Lors de la réalisation d'une application, l'écriture du code n'est pas la seule activité à effectuer, celle-ci doit impérativement être précédée d'une méthodologie d'analyse et de conception pour formaliser les étapes préliminaires du développement afin de le rendre plus fidèle aux besoins du client, la phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, de sécurité, d'extensibilité... elle consiste aussi en la construction d'un modèle du monde réel (le modèle d'analyse), qui met en évidence les propriétés importantes du problème. C'est une abstraction précise et concise du but de l'application et non de la façon dont elle sera implémentée.

2.2. Modélisation

2.2.1. Modèle

Un modèle est une représentation abstraite et simplifiée d'une entité du monde réel en vue de le décrire, ou de l'expliquer, il permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement de manière significative. Il reflète ce que le concepteur croit important pour la compréhension et la prédiction du phénomène modélisé ^[20].

2.2.2. Pourquoi modéliser ?

Modéliser un système avant sa réalisation permet de mieux comprendre le fonctionnement du système, de maîtriser sa complexité tout en assurant sa cohérence. Il assure aussi une compréhension commune par tous les membres de l'équipe. Un modèle permet de mieux répartir les tâches et d'automatiser certaines d'entre elles. C'est également un facteur de réduction des coûts et des délais, indispensable pour assurer un bon niveau de qualité et une maintenance efficace ^[20].

Pour ce faire, nous avons utilisé le langage UML qui est un langage spécialisé dans la modélisation des architectures logicielle. [Voir Annexe]

2.3. Acteurs et modélisation du contexte

2.3.1. Identifications des acteurs

Qu'est-ce qu'un Acteur ?

Un acteur représente un rôle joué par une entité externe au système modélisé (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec lui, en réponse à une action d'un acteur, le système fournit un service qui correspond à son besoin. Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données ^[21].

Les acteurs qui interagissent avec notre système sont :

A. Visiteur :

Un utilisateur dont les fonctionnalités se limitent à la localisation de la pharmacie ouverte la plus proche de sa position, en le localisant ou en saisissant une position. Il s'agit de n'importe quel utilisateur qui accède au site.

B. Client :

C'est un utilisateur ayant les mêmes fonctionnalités que le visiteur, sauf qu'il a la possibilité d'envoyer des ordonnances aux pharmaciens, et de poster des avis sur ces dernières, mais cela nécessite d'abord une authentification.

C. Pharmacien :

Il s'agit du propriétaire d'une pharmacie, il s'inscrit puis propose son emploi du temps et fait des mises à jour concernant ses horaires et jours de travail et ses gardes.

D. Serveur GoogleMaps :

L'un des acteurs principaux du système, il s'occupe de l'affichage des cartes, la récupération des données et la mise à disposition de ces dernières, du calcul des distances et des itinéraires.

2.4. Identification des besoins

Cette phase consiste à comprendre le contexte du système et de le décrire sans ambiguïté. Il s'agit des diverses fonctionnalités et des acteurs les plus importants.

2.4.1. Diagramme de cas d'utilisation

A. Définition

Les rôles des diagrammes de cas d'utilisation sont de recueillir, d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML pour la conception d'un système.

Un diagramme de cas d'utilisation capture le comportement d'un système, tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Ainsi ces cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique [22].

B. Composition du diagramme de cas d'utilisation ^[23]

Le diagramme de cas se compose de trois éléments principaux :

- **Un Acteur** : c'est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom inscrit dessous.
- **Un cas d'utilisation** : Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service. Il représente par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype.
- **Les relations** : Les relations indiquent que le cas d'utilisation source présente les mêmes conditions d'exécution que le cas issu. Une relation simple entre un acteur et une utilisation est un trait simple.

2.4.2. Besoins fonctionnels

Les besoins fonctionnels expriment une action devant être effectuée par le système en réponse à une demande.



Figure 14 : Diagramme de cas d'utilisation du système

2.4.3. Besoins non fonctionnels

A part les besoins fondamentaux, notre système doit répondre à certains critères qui agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur :

- a. **Fiabilité et disponibilité** : L'application fonctionne de façon cohérente sans erreurs et disponible 24/24, 7/7 sauf période de maintenance.
- b. **Les erreurs** : Les ambiguïtés sont signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre site web.
- c. **Utilisabilité** : L'application est adaptée à l'utilisateur sans qu'il ne fournisse aucun effort (utilisation claire et facile) de point de vue navigation entre les différentes pages, couleurs et mise en textes utilisés.
- d. **Sécurité** : Le système respecte la confidentialité des données personnelles des utilisateurs qui reste l'une des contraintes les plus importantes dans les sites web.
- e. **Aptitude à la maintenance et la réutilisation** : Le système est conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation.
- f. **Compatibilité et portabilité** : Un site web quel que soit son domaine, son éditeur et son langage de programmation ne peut être fiable qu'avec une compatibilité avec tous les navigateurs web et tous les moyens que ce soit PC, IPAD ou Mobiles.
- g. **Déploiement** : Le système du côté client va être utilisé sur un navigateur, sans aucune configuration.
- h. **Matériels** : Disponibilité d'un PC ou d'un Smartphone doté d'un accès internet sont les configurations matérielles minimales nécessaires au fonctionnement de notre système.
- i. **Performance** : Temps de chargement d'une page : Le chargement d'une page Web dans le navigateur ne devrait pas prendre plus de 15 sec en condition normale.

2.5. Conclusion

Dans ce chapitre, nous avons modélisé notre système en identifiant les différents acteurs impliqués et en détaillant les besoins fonctionnels grâce au diagramme des

cas d'utilisation et les besoins non fonctionnels qui sont tous les deux indispensables pour mieux faciliter la réalisation de notre travail.

Chapitre 3

3. Conception

3.1. Introduction

La réalisation d'un produit de qualité nécessite le passage par la phase de conception, celle-ci permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système et son architecture globale à mettre en œuvre. Elle fournit un support à l'implémentation physique du système et doit traduire la solution dans une architecture réalisable en termes de domaines techniques.

Au cours de ce chapitre nous allons détailler notre choix conceptuel à travers différents types de diagrammes.

3.2. Diagramme de séquence

A. Définition

Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont.

Les diagrammes de séquences sont organisés en fonction du temps qui s'écoule au fur et à mesure que nous parcourons la page.

Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence ^[24].

B. Composition d'un diagramme de séquences ^[22]

Ce type des diagrammes est composé par les éléments suivants :

- **Les lignes de vie** : Une ligne verticale qui représente la séquence des événements, produite par un participant, pendant une interaction, alors que le temps progresse en bas de ligne.
Ce participant peut être une instance d'une classe, un composant ou un acteur.
- **Les messages** : deux types de messages dans le diagramme de séquences, le premier est dit message synchrone utilisé pour représenter des appels de fonction ordinaires dans un programme, le deuxième est appelé message asynchrone, étant utilisé pour représenter la communication entre des threads distincts ou la création d'un nouveau thread.
- **Les occurrences d'exécution** : représente la période d'exécution d'une opération.
- **Les commentaires** : Un commentaire peut être joint à tout point sur une ligne de vie.
- **Les itérations** : représente un message de réponse suite à une question de vérification.

3.2.1. Description des cas d'utilisations

A. Description du cas « Rechercher une pharmacie »

Titre : Rechercher une pharmacie

Acteur : Utilisateur

But : Localiser les pharmacies à proximité

Résumé : Ce cas d'utilisation décrit les étapes que l'utilisateur doit suivre afin d'afficher les pharmacies à proximités qui respectent les critères qu'il aura établi.

Description d'enchaînement

Préconditions :

- Avoir une connectivité Internet
- Activer la localisation

Postcondition :

- La localisation des pharmacies, et l'affichage d'une liste de ces dernières ordonnées par proximité par rapport à l'utilisateur

Enchaînement nominal :

1. L'utilisateur spécifie un point de recherche (Sa propre position, ou une adresse)
2. Le système recherche la position donnée
3. Le système affiche une liste de pharmacies ouvertes, ordonnée par distance par rapport à l'utilisateur
4. L'utilisateur sélectionne une pharmacie dans la liste
5. L'utilisateur choisit l'itinéraire menant à la pharmacie (à pied ou en voiture)
6. Le système calcule les meilleurs itinéraires possibles
7. Le système affiche les itinéraires ainsi que les informations disponibles sur la pharmacie

Enchaînement alternatif :

Aucune connexion internet

2.a Impossible d'avoir la position du point de recherche

4.a Aucune pharmacie respectant les critères de recherche n'est ouverte

Diagramme de séquence

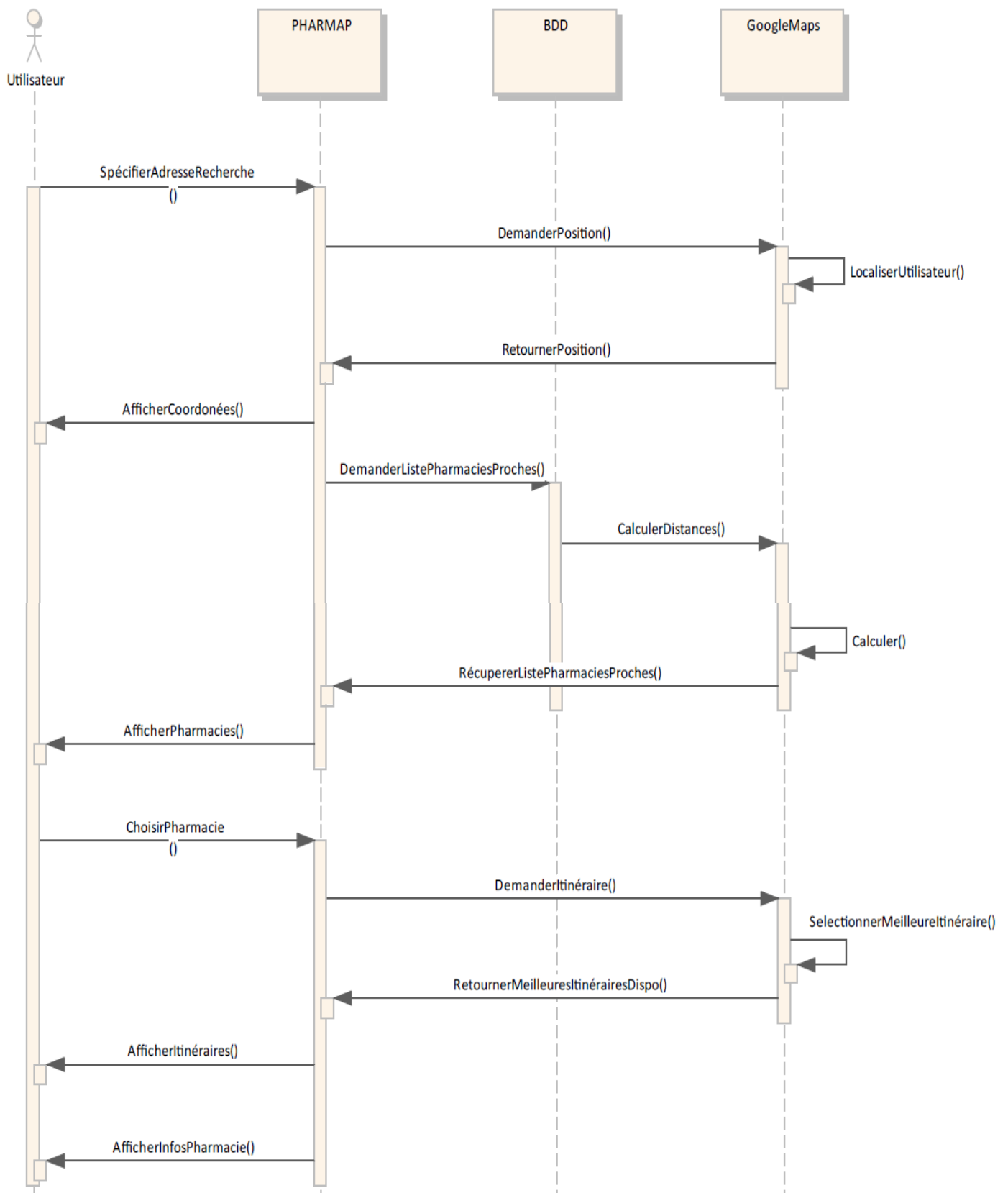


Figure 15 : Diagramme de séquence « Rechercher pharmacie »

B. Description du cas « Transmettre ordonnance à une pharmacie »

Titre : Transmettre une ordonnance à une pharmacie

Acteur : Client

But : Envoyer une ordonnance à une pharmacie afin de vérifier la disponibilité de certains médicaments

Résumé : Après avoir choisi une pharmacie le client peut contacter cette dernière en lui envoyant une ordonnance, et attendre que le pharmacien le contacte pour l'informer des disponibilités des médicaments présents sur l'ordonnance.

Description d'enchaînement

Préconditions :

- Être authentifié
- Avoir choisi une pharmacie destinataire
- Avoir une ordonnance

Postcondition :

- Réception de l'ordonnance de la part du pharmacien

Enchaînement nominal :

1. L'utilisateur introduit une image de son ordonnance dans le système
2. Le système envoie l'ordonnance au pharmacien et le notifie de l'arrivée de celle-ci
3. Le système fait appel au cas d'utilisation « Traiter les ordonnances reçues »

Enchaînement alternatif :

1.a L'utilisateur n'a pas d'ordonnance

2.a Le système fait une erreur lors de l'envoi de l'ordonnance

Diagramme de séquence

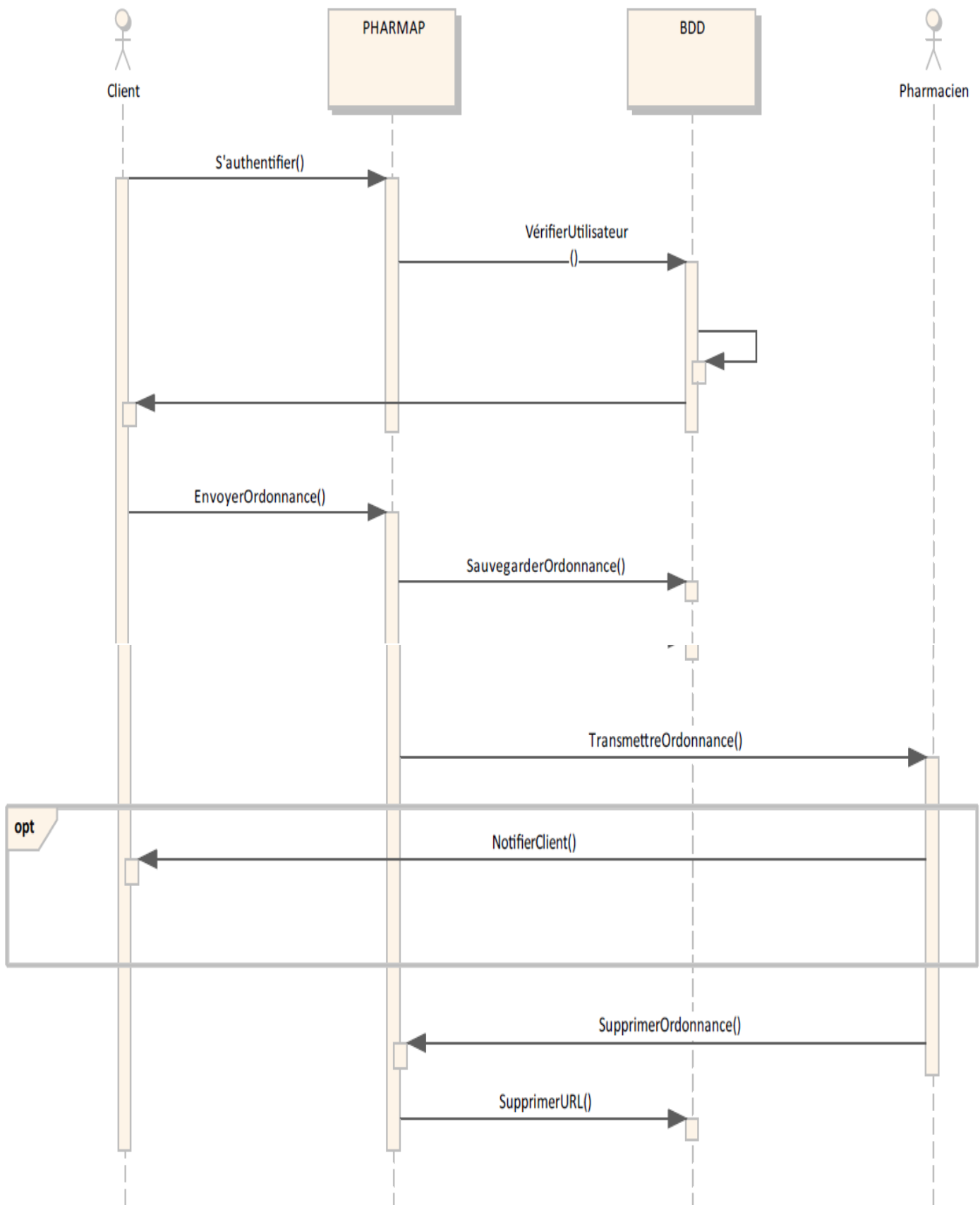


Figure 16 : Diagramme de séquence "Transmettre Ordonnance"

Description du cas « Soumettre commentaire »

Titre : Soumettre un commentaire sur le profil de la pharmacie

Acteur : Client

But : Poster un avis et une note sur la pharmacie

Résumé : Le client peut poster un commentaire à propos de la pharmacie et y donner son avis et une note.

Description d'enchainement

Précondition :

- Être authentifié

Postcondition :

- Publication du commentaire sur le profil du pharmacien

Enchainement nominal :

1. L'utilisateur va sur le profil d'une pharmacie
2. Le système affiche la liste des avis sur la pharmacie
3. L'utilisateur rédige le commentaire à propos de la pharmacie
4. L'utilisateur attribue une note sur 5 à la pharmacie
5. L'utilisateur poste le commentaire sur le profil de la pharmacie
6. Le système actualise les commentaires et la moyenne des avis sur la pharmacie
7. Le système notifie le pharmacien de la publication d'un nouvel avis sur son profil

Enchainement alternatif :

5.a L'envoi du commentaire sur le profil de la pharmacie échoue

7.a Le pharmacien n'est pas notifié de la publication d'un nouvel avis

Diagramme de séquence

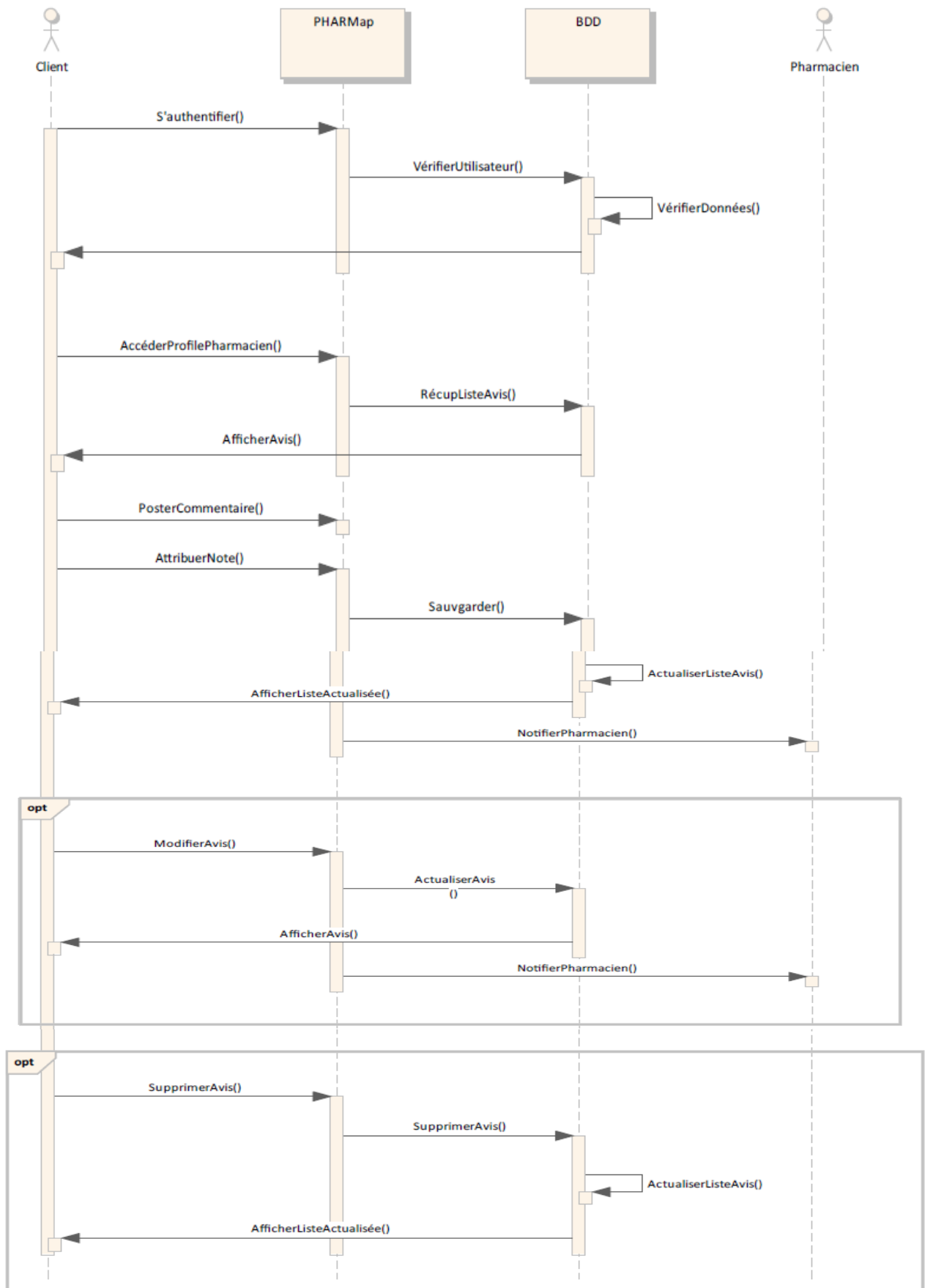


Figure 17 : Diagramme de séquence "Soumettre commentaire"

C. Description du cas « Créer un compte pharmacien »

Titre : Créer un compte pharmacien

Acteur : Pharmacien

But : Enregistrer une pharmacie dans le système

Résumé : Ce cas d'utilisation décrit les étapes que le pharmacien doit suivre afin de créer son compte et d'enregistrer sa pharmacie dans le système.

Description d'enchaînement

Précondition :

- Aucune

Postcondition :

- L'enregistrement de la pharmacie dans le système

Enchaînement nominal :

1. Le Pharmacien introduit ses informations personnelles (Nom, prénom, email...)
2. Le Pharmacien spécifie les informations spécifiques à sa pharmacie (code pharmacie, Assurance sociale, heures de travail etc...)
3. Le Pharmacien spécifie l'emplacement de sa pharmacie sur la carte
4. Le système enregistre les informations fournies par le pharmacien dans sa base de données

Enchaînement alternatif :

Aucune connexion internet

3.a La pharmacie n'est pas trouvable sur la carte

4.a La pharmacie est déjà présente dans le système

Diagramme de séquence

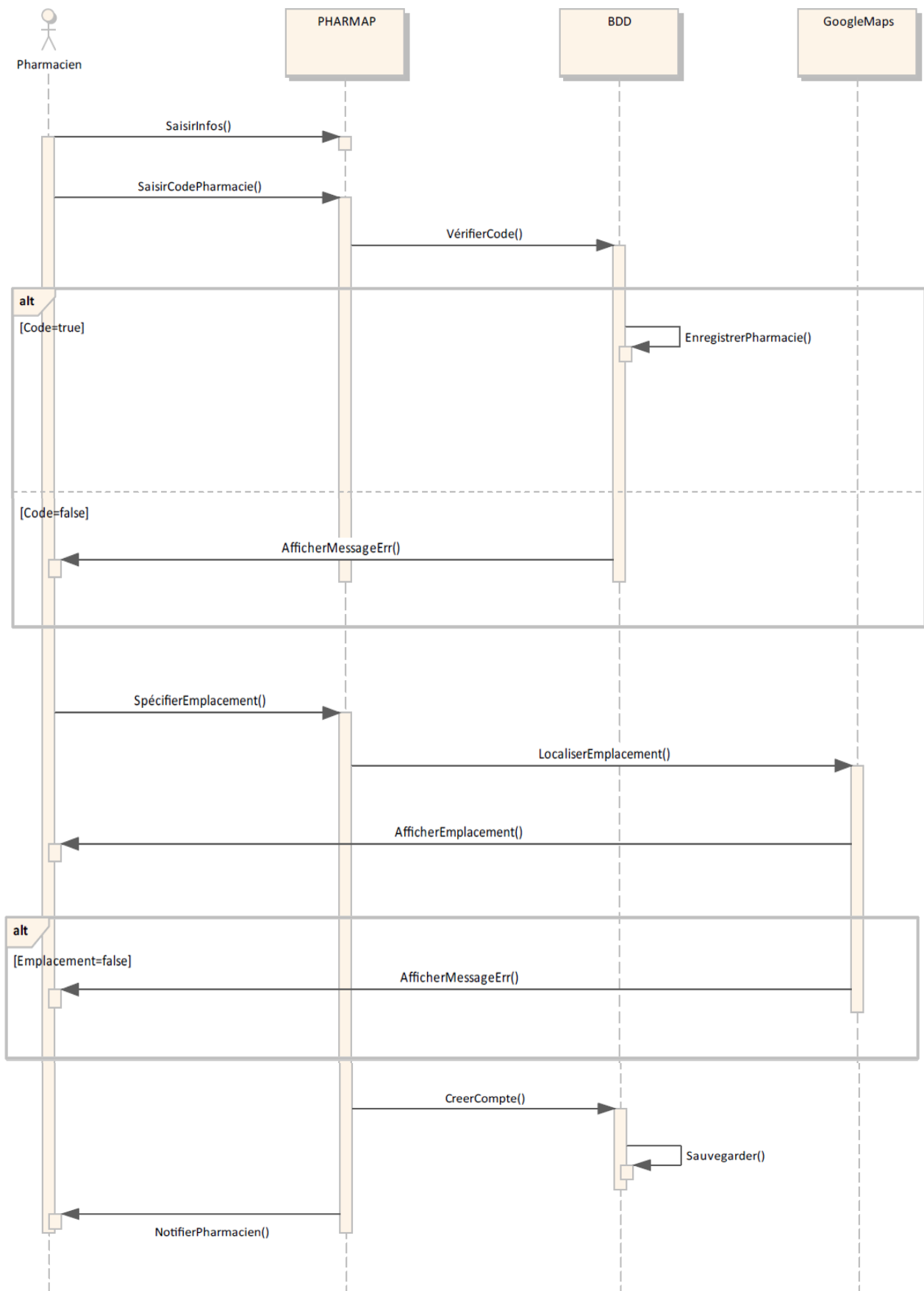


Figure 18 : Diagramme de séquence "Créer compte pharmacien"

D. Description du cas « Gérer les gardes »

Titre : Gérer les gardes

Acteur : Pharmacien

But : Ajouter, modifier, supprimer un jour de garde de la pharmacie

Résumé : Le pharmacien peut spécifier les jours où il sera de garde, et ce cas d'utilisation décrit comment il fait pour gérer ses gardes.

Description d'enchaînement

Précondition :

- Être authentifié

Postcondition :

- La mise à jour de la liste des jours de garde de la pharmacie

Enchaînement nominal :

1. Le pharmacien choisit l'option « Gestion de gardes »
2. Si le pharmacien choisit l'option d'ajout ou de modification, il spécifie les horaires de début et de fin de la garde
3. Le pharmacien choisit le jour où il sera de garde
4. Le pharmacien valide ce qu'il a saisi
5. Le système enregistre les informations de la garde dans sa base de données

Enchaînement alternatif :

- 2.a Le pharmacien choisit de supprimer la garde

Diagramme de séquence

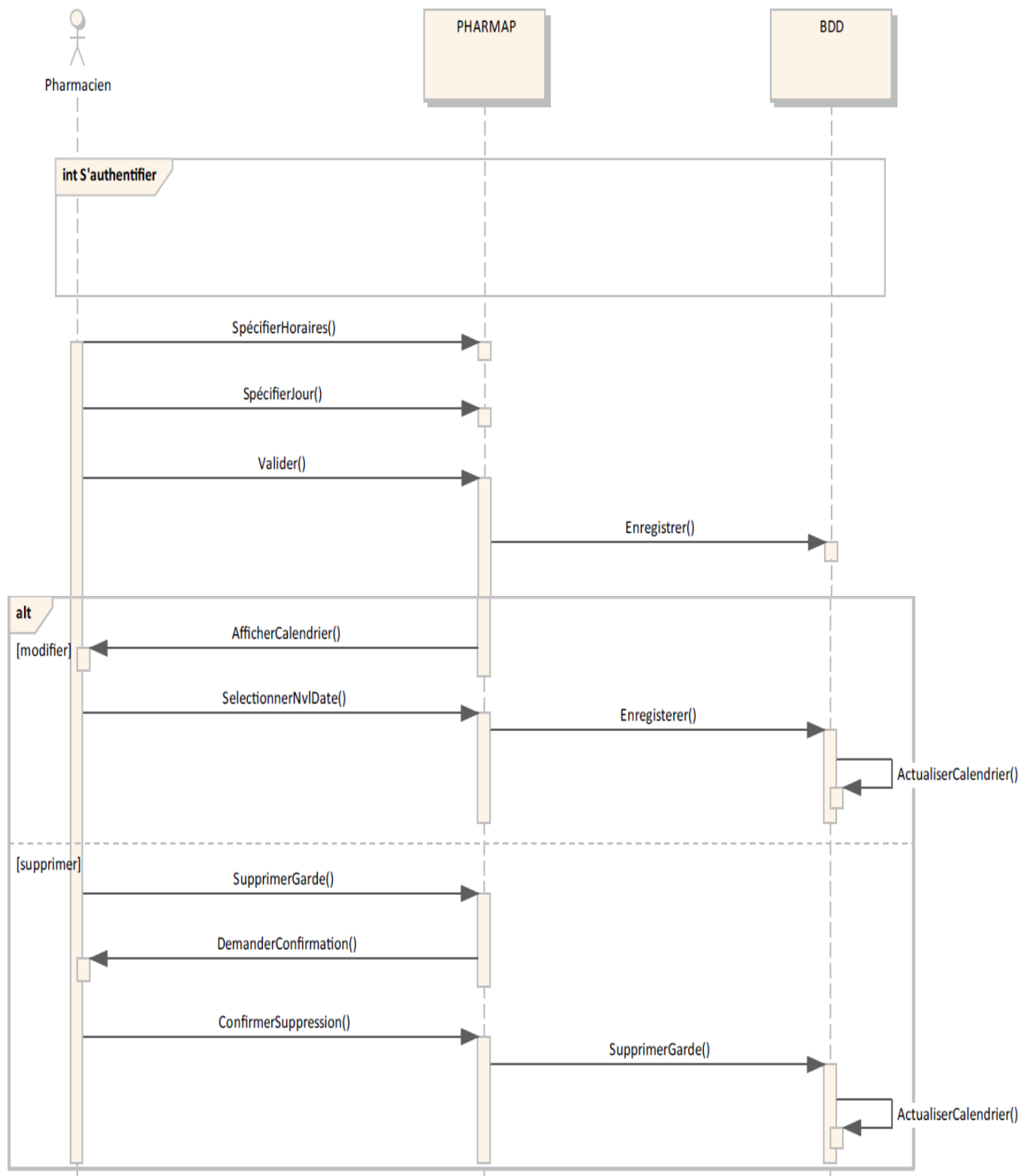


Figure 19 : Diagramme de séquence "Gérer gardes"

3.3. Diagramme de classe

A. Définition

Un diagramme de classes UML décrit les structures d'objets et d'informations utilisées sur notre site web, à la fois en interne et en communication avec ses utilisateurs. Il décrit les informations sans faire référence à une implémentation particulière. Ses classes et relations peuvent être implémentées de nombreuses manières, comme les tables de bases de données, les nœuds XML ou encore les compositions d'objets logiciels [22].

En général un diagramme de classe peut contenir les éléments suivants :

- **Les classes** : une classe représente la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes. Elle est représentée en utilisant un rectangle divisé en trois sections. La section supérieure est le nom de la classe, la section centrale définit les propriétés de la classe alors que la section du bas énumère les méthodes de la classe.
- **Les associations** : une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc que des liens peuvent exister entre des instances des classes associées.
- **Les attributs** : les attributs représentent les données encapsulées dans les objets des classes. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisé. Le nom de l'attribut doit être unique dans la classe [22].

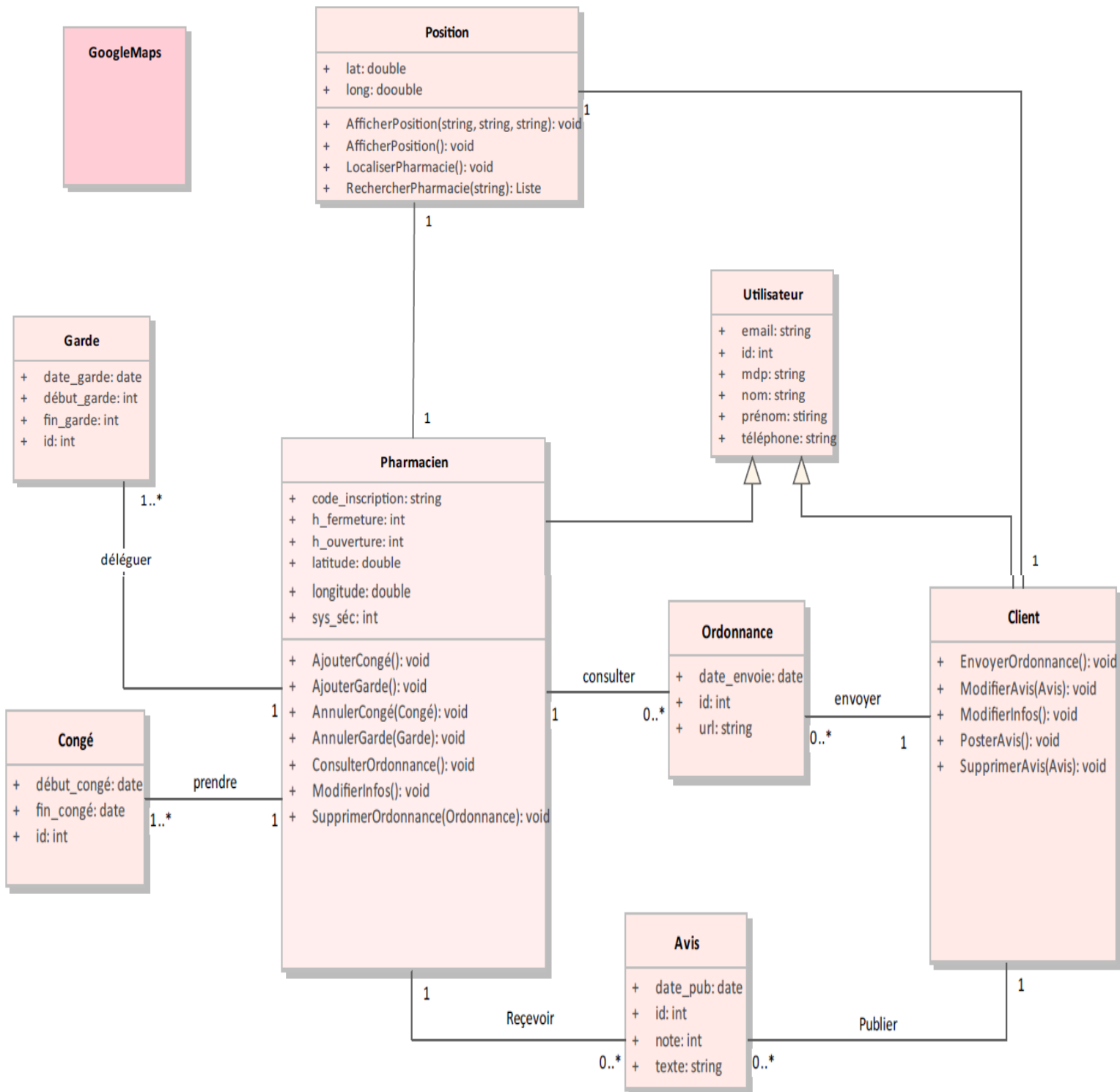


Figure 20 : Diagramme de classe PHARMAP

B. Description des classes

Classe	Attribut	Description	Résumé
Utilisateur	email	Adresse email de l'utilisateur (Pharmacien ou Client)	Classe abstraite représentant les utilisateurs du système, et regroupant les informations générales d'un utilisateur.
	id	Identifiant de l'utilisateur	
	mdp	Mot de passe de l'utilisateur	
	nom	Nom de l'utilisateur	
	prénom	Prénom de l'utilisateur	
	téléphone	Numéro de téléphone de l'utilisateur	
Pharmacien	code_inscription	Code d'inscription du pharmacien	Classe qui stocke les informations concernant l'ouverture et fermeture de la pharmacie et son affiliation aux systèmes de sécurité sociale, elle s'occupe aussi des traitements faits par le pharmacien tel que la gestion des congés, des gardes mais aussi les informations de la pharmacie
	h_fermeture	Heure de fermeture de la pharmacie	
	h_ouverture	Heure d'ouverture de la pharmacie	
	latitude	Latitude de la pharmacie	
	longitude	Longitude de la pharmacie	
	sys_séc	Systèmes de sécurité sociale	
Client	Hérite tous les attributs de la classe utilisateur		Elle représente l'utilisateur basique du système, cette classe s'occupe des traitements fait par un client : publier un avis, le modifier ou le supprimer, et l'envoi d'une ordonnance.

Position	long	Longitude	C'est la classe pivot de notre système, elle s'occupe de tous les traitements en rapport avec les positions des utilisateurs, les calculs de distances et d'itinéraires, et ce, grace à l'API Google Maps avec qui elle est en contact.
	lat	Latitude	
Garde	date_garde	Date de garde	Classe représentant les gardes qu'un pharmacien prévoit de faire.
	debut_garde	Heure du début de la garde	
	fin_garde	Heure de fin de la garde	
	id	Identifiant de la garde	
Congé	début_congé	Date du début de congé	Classe pour stocker les périodes de congés.
	fin_congé	Date de fin de congé	
	id	Identifiant du congé	
Ordonnance	date_envoie	Date d'envoi de l'ordonnance	Classe pour stocker les ordonnances envoyées par les clients aux pharmaciens.
	id	Identifiant de l'ordonnance	
	url	Url de l'ordonnance	
Avis	date_pub	Date de publication du commentaire	Classe pour sauvegarder les notes et commentaires publiés par les clients sur les profils des pharmaciens.
	id	Identifiant du commentaire posté sur le profil d'un pharmacien	
	note	Note attribuée par un client à un pharmacien	
	texte	Contenu du commentaire	

GoogleMaps API	///	///	C'est l'API sur laquelle se base notre application, elle affiche les cartes, et fournit les données spécifiques aux locations demandées.
-------------------	-----	-----	--

Tableau 1 : Tableau de description des classes

3.4. Passage au modèle relationnel

Le modèle relationnel est une manière de modéliser les relations existantes entre plusieurs informations, et de les ordonner entre elles.

Afin de pouvoir implémenter une base de données, il faut pouvoir traduire le modèle conceptuel en modèle logique. Cela signifie qu'il faut pouvoir convertir un modèle UML en modèle relationnel.

Après l'application des règles de passage [voir Annexe] on obtient le schéma relationnel suivant :

Pharmaciens (id, nom, prenom, email, password, numero, tel, assurance, lat, lng, ouverture, fermeture)

Ordonnance (id, #id_pharmacien, #id_client, date, url)

Clients (id, nom, prenom, email, password, tel)

Avis (id, #id_pharmacien, #id_client, date, note, avis)

Gardes (id, #id_pharmacien, garde_date, garde_debut, garde_fin)

Conges (id, #id_pharmacien, cong_debut, cong_fin)

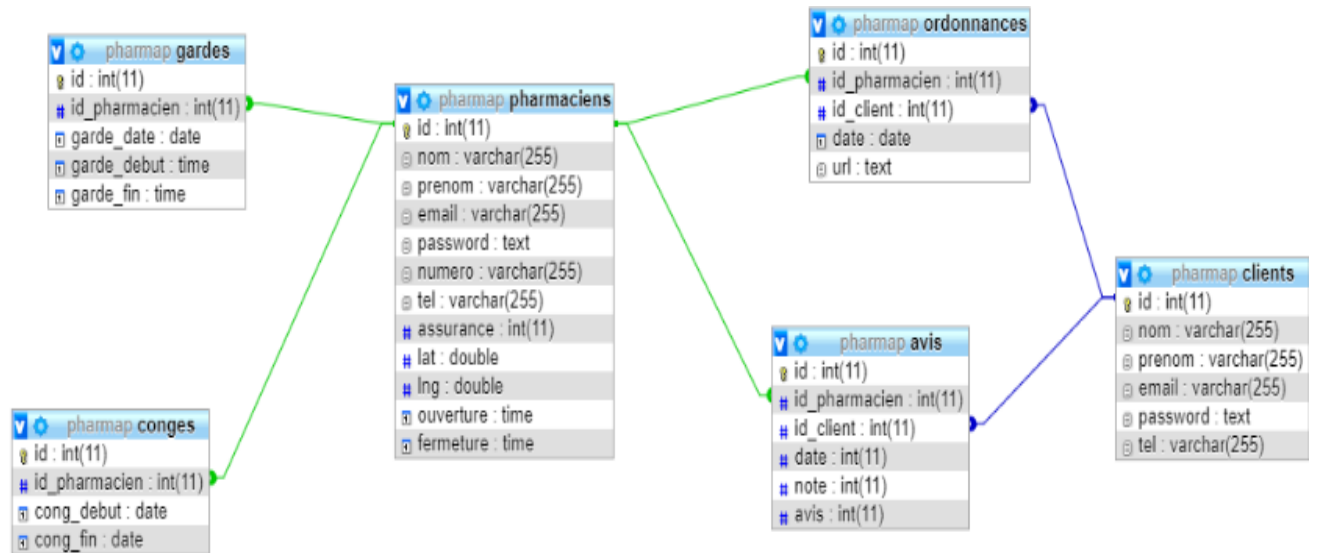


Figure 21 : Modèle relationnel Pharmap

3.5. Conclusion

Dans ce chapitre, nous avons modélisé les besoins de notre système grâce au diagramme de cas d'utilisation, diagramme de séquence et diagramme de classe dans le but de faciliter son implémentation.

Chapitre 4

4. Réalisation

4.1. Introduction

Au cours de ce chapitre nous allons exposer le travail accompli. Tout d'abord, nous allons commencer par la description de deux concepts qui ont été utilisés au cours de la réalisation de ce projet, des langages et autres outils utilisés puis nous clôturerons ce chapitre en exposant quelques captures d'écran de l'interface de notre application.

4.2. Design Pattern MVC ^[25]

Dans le cadre de ce projet en groupe, l'architecture adoptée durant la réalisation de l'application a été l'architecture MVC. C'est un pattern très populaire qui nous a permis d'être efficace et structuré lors du développement de notre projet.

MVC (Model - View - Controller) est un pattern destiné à la conception d'applications GUI. Son principe de base est la séparation des composants suivants :

- **Le modèle** : il conserve toutes les données relatives à l'application (sous quelque forme que ce soit : base de données, fichiers...) et contient la logique métier de l'application.
- **La vue** : elle a pour rôle d'offrir une présentation du modèle (IHM). On peut avoir de nombreuses vues pour un même modèle, chacune présentant les informations de manière différente (on pourrait ainsi imaginer la liste de pharmacies à proximité d'un client présentable à la fois sur l'écran d'un navigateur web, sur un minitel ou sur une imprimante)

- **Le contrôleur** : c'est le composant qui répond aux actions de l'utilisateur. Il traduit les événements de l'IHM en modifications du modèle et définit également la manière dont l'IHM doit réagir face aux interactions de l'utilisateur.

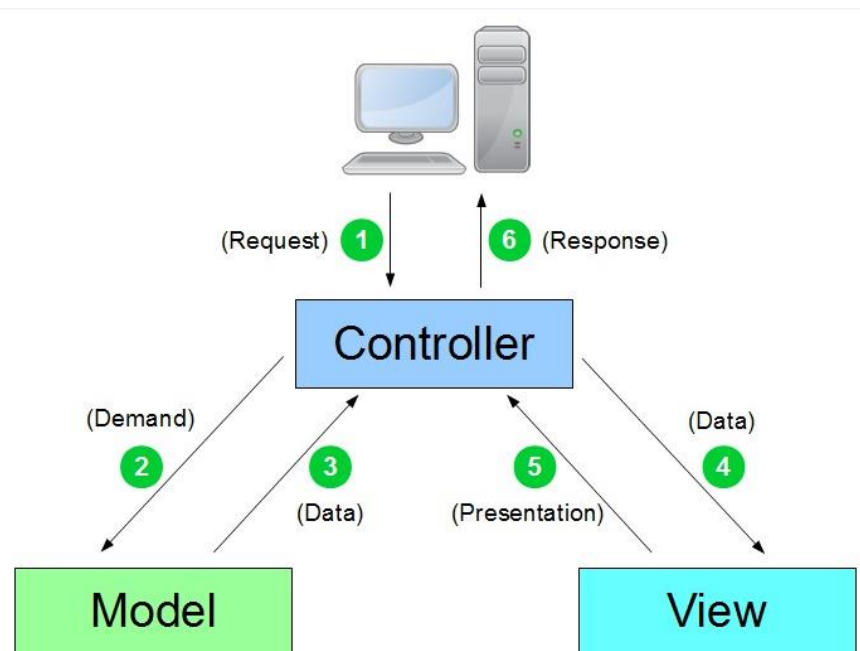


Figure 22 : Fonctionnement du pattern MVC [26]

La compréhension et la maîtrise de l'architecture MVC nécessitent un temps d'apprentissage assez conséquent, mais cet inconvénient est largement contrebalancé par le temps gagné lors des évolutions et de la maintenance.

Toutefois, nous avons aussi pu relever d'autres avantages liés à l'architecture MVC, qui ne sont autre que :

- Les 3 parties de l'application – logique de présentation, logique métier et logique applicative – sont parfaitement indépendantes. Ainsi, la répartition du travail de programmation de chacune de ces parties a pu être distribuée de la manière la plus efficace possible.
- La maintenance de l'application est plus souple. Ce découpage nous a accordé la possibilité de, par exemple de modifier la présentation de l'interface, et ce, sans toucher à la structure du site ni à la logique métier.

4.3. Implémentation et outils utilisés

4.3.1. HTML



HyperText Markup Language généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est le langage principal du web, et le plus populaire, et aussi compatible avec tous les navigateurs web qui existent. Nous l'avons utilisé pour écrire et structuré le contenu de nos vues.

4.3.2. CSS



Cascading Style Sheets, aussi appelées Feuilles de style. Le rôle de ce langage est de gérer l'apparence, la mise en forme des pages web. Le CSS a besoin d'une page web pour fonctionner, c'est pour cela qu'il est très étroitement utilisé avec le HTML5, car ils se complètent mutuellement.

4.3.3. Javascript



Javascript est un langage de programmation web et aussi de scripting. Il est principalement utilisé pour améliorer les pages Web afin de fournir une expérience plus conviviale. Cela inclut la mise à jour dynamique des pages Web, les améliorations de l'interface utilisateur telles que les menus et les boîtes de dialogue.

Mais dans le cadre de notre projet, javascript fût principalement utilisé pour la manipulation de [Google Maps Javascript API](#), via des méthodes que cette dernière fournit, comme l'initialisation de la carte, de la position de départ, du zoom etc...

4.3.4. PHP



PHP est langage impératif orienté objet coté serveur, open source, principalement utilisé pour produire des pages web dynamiques en interprétant des requêtes HTTP venant de la part des utilisateurs du site web (demandes concernant la base de données par exemple), et en gérant le comportement du serveur suivant la requête.

Il existe aussi d'autres langages coté serveur utilisé pour la création d'applications web, tel que Django (Framework Python), ASP.NET, mais nous avons opté pour la version 7.1.16 de PHP, car elle y assure davantage de sécurité, un traitement plus rapide, mais principalement car PHP est supporté par la grande majorité des hébergeurs et serveurs web (Apache, Nginx...).

4.3.5. MariaDB



MariaDB est un Système de Gestion de Bases de Données Relationnelles, qui utilise le langage SQL. C'est un SGBDR très populaire, et sa popularité est due en grande partie au fait qu'il s'agit d'un logiciel Open Source (sous Licence GNU GPL), ce qui signifie que son code source est librement disponible et que quiconque qui en ressent l'envie et/ou le besoin peut modifier MariaDB pour l'améliorer ou l'adapter à ses besoins.

Non seulement le fait qu'il soit gratuit, nous avons opté pour MariaDB car il garantit une facilité d'installation, de hautes performances et une administration simplifiée.

4.4. Frameworks

4.4.1. Bootstrap



Bootstrap est un framework open-source développé par l'équipe Twitter. C'est un outil combinant HTML, CSS, et JavaScript. Bootstrap est une collection gratuite d'outils pour la création de sites Web et d'applications Web.

Il contient des modèles de conception HTML et CSS pour la typographie, les formulaires, les boutons, la navigation et d'autres composants d'interface, ainsi que des extensions JavaScript optionnelles ^[27].

Parmi un grand choix de Responsive Design Frameworks, nous avons choisis Bootstrap car c'est le plus adapté à nos besoins, le plus complet, l'un des plus populaire ce qui garantit la présence d'une communauté très grande en cas de besoin d'aide, et d'une documentation très riche ce qui facilite sa manipulation, et donc de

créer des interfaces adaptées à toutes les tailles d'écrans garantissant une navigation des plus agréables.

4.5. APIs

4.5.1. Google Maps API



C'est une API créée par Google, et en utilisant Javascript, on peut intégrer des cartes sur notre site web, exécuter de diverses requêtes, tel que le calcul d'itinéraire, la recherche d'un point etc...

Afin d'exécuter ces requêtes nous avons eu recours aux services de l'API Google Maps suivants :

- **Service Geocoding** : Le géocodage est le processus qui permet de convertir des adresses que vous pouvez ensuite utiliser pour placer des marqueurs ou positionner la carte. Le géocodage inversé est le processus de conversion de coordonnées géographiques en adresses lisibles. Le géocodeur inversé vous permet également de retrouver l'adresse correspondant à un identifiant de lieu donné ^[28].
- **Service Directions** : C'est le service qui s'occupe du calcul et de l'affichage des itinéraires. Le service Directions peut renvoyer des itinéraires multi-segments passant par une série de points de cheminement. L'itinéraire est affiché sous forme de polyligne qui représente le trajet sur une carte.
- **Service Distance Matrix** : calcule les distances et les durées des trajets entre plusieurs points de départ et destinations avec un mode de transport donné. Ce service ne renvoie pas d'informations détaillées sur l'itinéraire. Pour obtenir les informations d'itinéraire, il faut transmettre le point de départ et la destination souhaités au *Service Directions* ^[29].

Il existe d'autres alternatives à Google Maps API, tel qu'OpenstreetMaps, Bing Maps, Yahoo Maps etc... Mais notre choix s'est porté sur Google Maps de par le fait qu'il soit le plus aboutis et de par sa notoriété, ce qui fait que c'est l'API qui contient le plus de données exploitables.

4.6. Outils de développement

4.6.1. Xampp



Xampp est une plateforme de développement Web permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP ou Perl. Xampp n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs Apache et MariaDB (L'utilisation d'autres SGBD est possible, tel que MySQL dans notre cas), un interpréteur de script (PHP, Perl) ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

4.6.2. Visual Studio Code



Visual Studio Code est un éditeur de code source développé par Microsoft pour Windows, Linux et MacOS. Il inclut la prise en charge du débogage, du contrôle Git intégré (pour le versioning), de la mise en évidence de la syntaxe, de l'achèvement de code intelligent, des extraits et du refactoring de code. Il est également personnalisable, de sorte que les utilisateurs peuvent modifier le thème de l'éditeur, les raccourcis clavier et les préférences. Il est gratuit et open-source, bien que le téléchargement officiel soit sous licence exclusive.

Visual Studio Code combine la simplicité d'un éditeur de code source avec de puissants outils de développement, tels que l'achèvement et le débogage de code IntelliSense. Il facilite l'écriture et le débogage du code et ce sans aucun problème, ce qui nous a accordé plus de temps pour exécuter nos idées.

4.7. Captures d'écran

Nous allons présenter dans cette partie les principales interfaces développées dans notre site web et leurs scénarios applicatifs.

4.7.1. Page d'accueil « la carte »

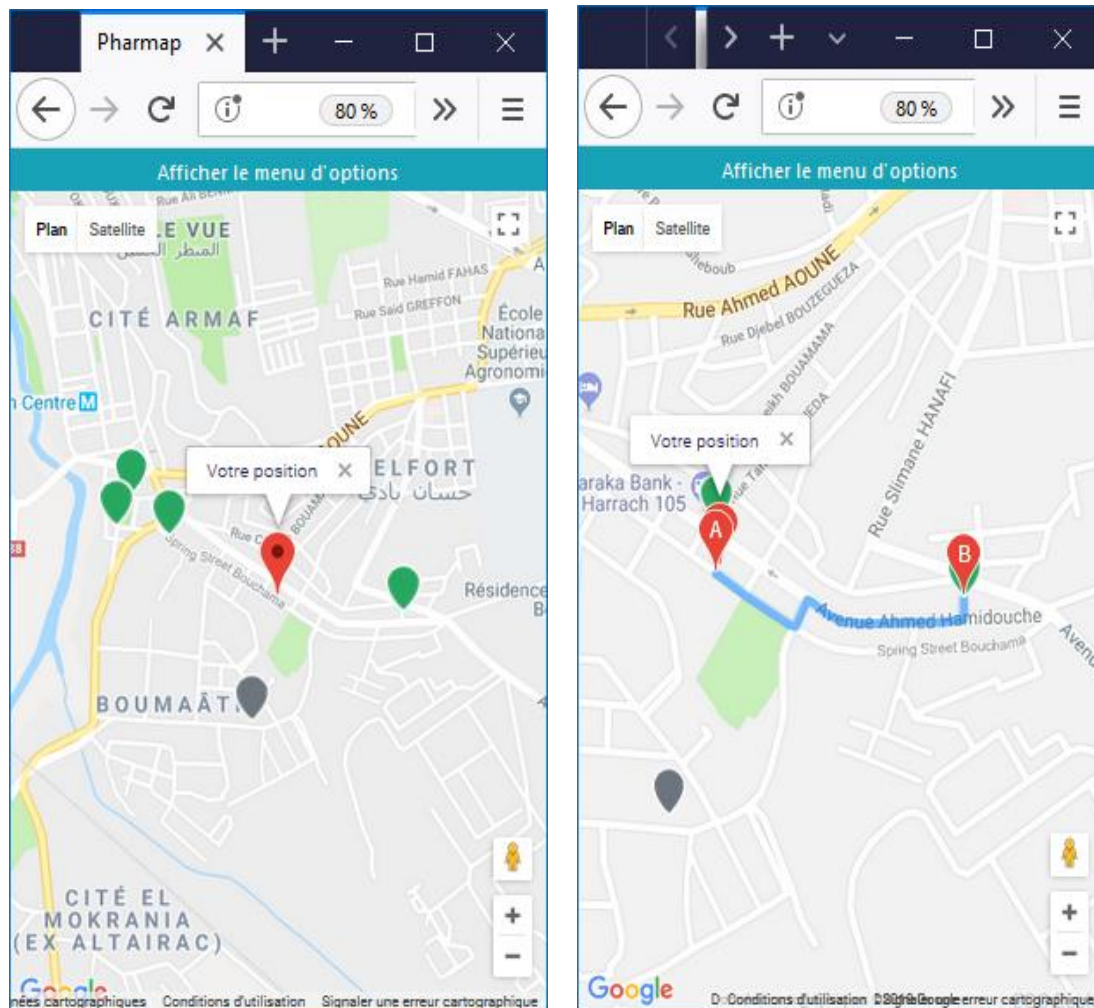


Figure 23:page d'accueil PHARMAP

Dès l'accès au site, la carte s'affiche indiquant :

- **Un marqueur rouge** : Désignant la position de l'utilisateur.
- **Un marqueur vert** : Désignant la position d'une pharmacie ouverte à proximité.
- **Un marqueur gris** : Désignant la position d'une pharmacie fermée à proximité.

4.7.2. Menu d'options



The screenshot shows a mobile application interface for 'Pharmap'. At the top, there's a dark blue header with the app name 'Pharmap' and standard mobile window controls (back, forward, refresh, info, zoom 80%, and a menu icon). Below the header, a teal bar contains the text 'Masquer le menu d'options'. The main content area is light green and titled 'Recherche de pharmacies'. It features a blue link 'liste des resultats'. Below this are four input fields: 'Ville', 'Commune', 'Adresse', and 'Distance de recherche' (set to 10 Km). There are two radio buttons for 'Choix d'itinéraire': 'A pied' (selected) and 'Par vehicule'. A green 'Valider' button is at the bottom of the form. At the very bottom, there are two blue links: 'se connecter en tant que pharmacien' and 'se connecter en tant que client'.

Figure 24: Menu d'options

En cliquant sur le menu d'option, le lien « liste de résultats » nous permet de consulter les informations des pharmacies les plus proches localisées depuis la position par défaut.

On a aussi la possibilité de saisir un point de recherche s'affiche, on introduit la ville, la commune, ou l'adresse, tout en précisant le périmètre de recherche qui est par défaut initialisé à 10 KM, mais aussi de choisir l'itinéraire à pied ou par véhicule puis avec un clic sur le bouton « Valider », une liste de résultats s'affichera.

En bas s'affiche aussi les deux options « se connecter en tant que pharmacien » et « se connecter en tant que client ».

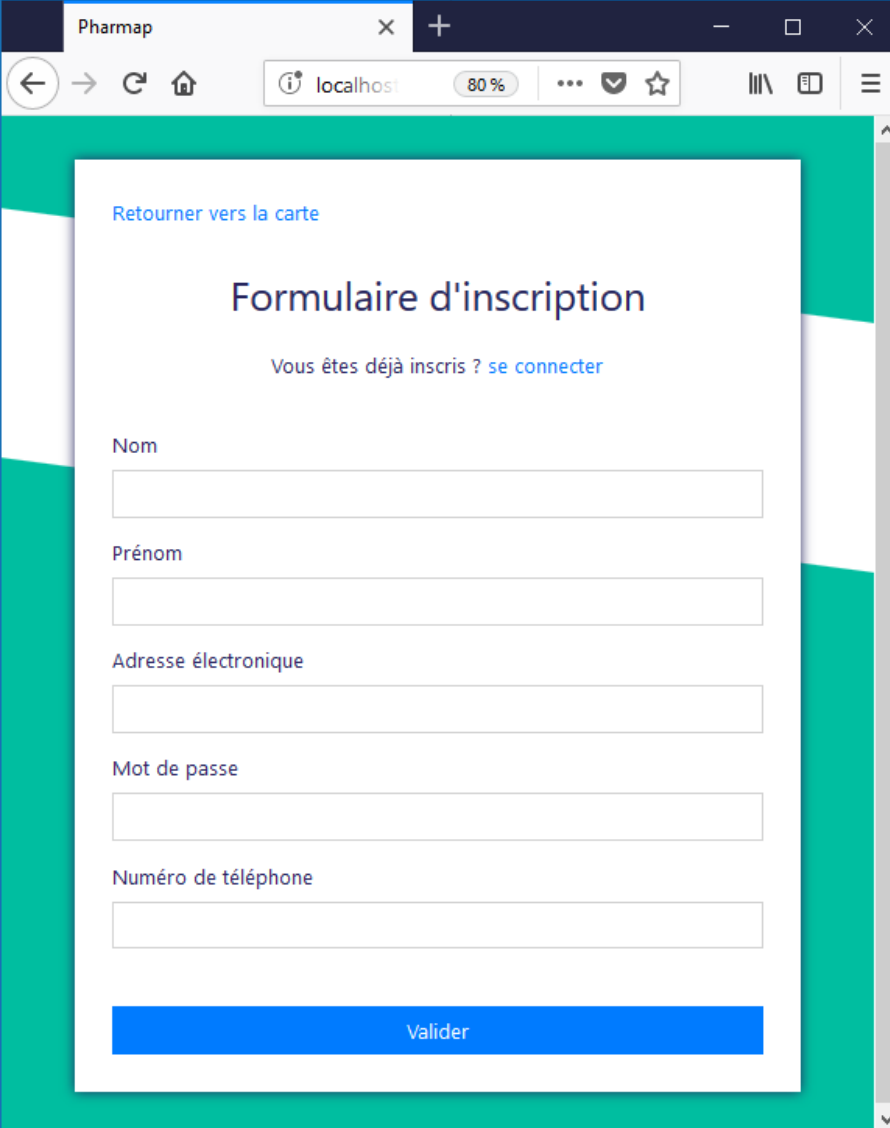
4.7.3. Liste des résultats



Figure 25: Liste des résultats

La liste des résultats s'affiche manuellement en cliquant sur le lien « Liste des résultats » depuis le menu d'option dès l'accès au site, ou après la validation de la recherche manuelle.

4.7.4. Inscription Client



The screenshot shows a web browser window with the title 'Pharmap'. The address bar displays 'localhost' and a loading indicator at '80%'. The page content features a teal header with a link 'Retourner vers la carte'. Below this is a white box containing the title 'Formulaire d'inscription' and a link 'Vous êtes déjà inscrits ? se connecter'. The form includes five input fields: 'Nom', 'Prénom', 'Adresse électronique', 'Mot de passe', and 'Numéro de téléphone'. At the bottom of the form is a blue button labeled 'Valider'.

Figure 26: Inscription Client

Pour la création d'un compte client l'utilisateur doit remplir tous les champs du formulaire d'inscriptions.

4.7.5. Inscription Pharmacien

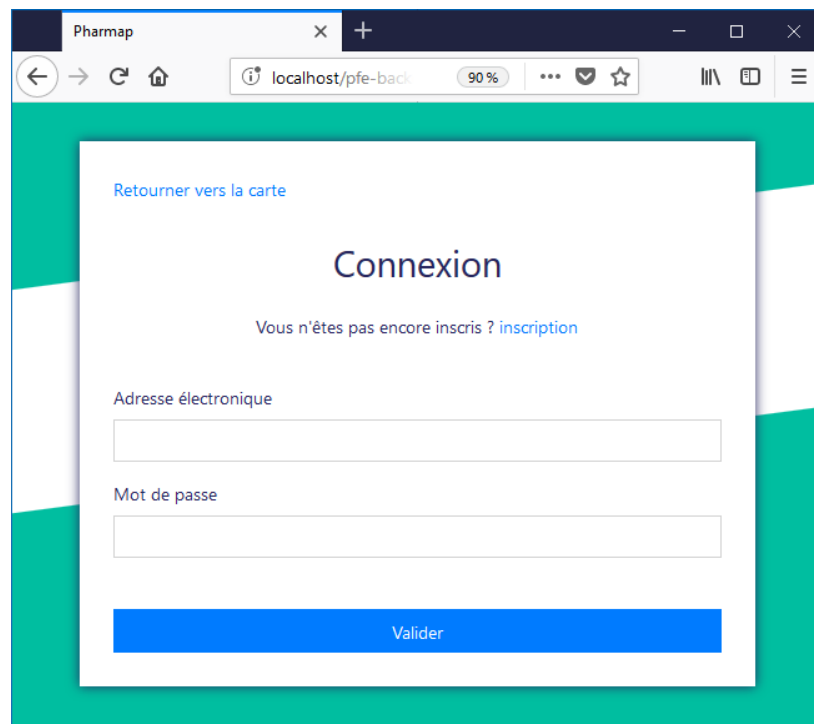
The figure displays four sequential screenshots of a web browser window titled "Pharmap" at the URL "localhost/". The browser's zoom level is set to 80%.

- Top Left Screenshot:** Shows the "Formulaire d'inscription" (Registration Form). It includes a link "Retourner vers la carte" (Return to the map), a question "Vous êtes déjà inscrits ? se connecter" (Are you already registered? log in), and a section titled "Informations du compte" (Account Information) with input fields for "Nom" (Name), "Prénom" (First Name), "Adresse électronique" (Email Address), and "Mot de passe" (Password).
- Top Right Screenshot:** Shows the "Information sur la pharmacie" (Pharmacy Information) section. It contains input fields for "Nom de la pharmacie" (Pharmacy Name), "Numéro d'inscription à l'ordre des pharmaciens" (Registration Number with the Order of Pharmacists), and "Numéro de téléphone de la pharmacie" (Pharmacy Phone Number). Below these are checkboxes for the "Système d'assurance sociale" (Social Security System): "CASNOS", "CNAS", and "CAMSSP".
- Bottom Left Screenshot:** Shows the "Informations de géolocalisation et horaires" (Geolocation and Hours) section. It includes input fields for "Ville" (City), "Commune" (Municipality), and "Adresse" (Address), followed by a blue "Localiser" (Locate) button.
- Bottom Right Screenshot:** Shows the "Localisation de la pharmacie" (Pharmacy Location) section. It features a Google Map of Algeria with a marker at "Alger" (Algiers). Below the map are input fields for "Horaires d'ouverture" (Opening Hours) and "Horaires de fermeture" (Closing Hours), and a blue "Valider" (Validate) button.

Figure 27: Formulaire d'inscription pour un compte pharmacien

Pour la création d'un compte pharmacien l'utilisateur doit introduire des informations personnelles, des informations sur sa pharmacie, et des informations sur la géolocalisation de celle-ci et ses horaires d'ouvertures et fermetures.

4.7.6. « Se connecter en tant que client » et « Se connecter en tant que pharmacien »



The screenshot shows a web browser window with the title 'Pharmap'. The address bar displays 'localhost/pfe-back' with a 90% zoom level. The main content area has a teal background. A white modal box is centered on the screen with the following elements:

- A link at the top left: [Retourner vers la carte](#)
- A title in the center: **Connexion**
- A text prompt: Vous n'êtes pas encore inscrit ? [inscription](#)
- A label 'Adresse électronique' above a text input field.
- A label 'Mot de passe' above a text input field.
- A blue button at the bottom labeled 'Valider'.

Figure 28: Connexion

La connexion à un compte client ou pharmacien se fait après la saisie de l'adresse électronique et du mot de passe associés à un compte préalablement créé.

4.7.7. Connexion à un compte pharmacien

The screenshot shows a web browser window titled 'Pharmap' with the address bar displaying 'localhost/pharmap/Horaires'. The page features a sidebar on the left with four menu items: 'Informations du compte', 'Informations sur les horaires', 'Informations de géolocalisation', and 'Déconnexion' (highlighted in orange). The main content area is divided into two sections. The top section, titled 'Horaires de fermeture', contains two input fields with the values '08:00:00' and '20:00:00', each with a clear button (X). A blue 'Valider' button is positioned below these fields. The bottom section, titled 'Informations sur les Gardes', contains three input fields: 'Date de la garde' with the value '06/06/2018', 'Heure de début' with the value '20:00', and 'Heure de fin' with the value '08:00'. Each of these fields also has a clear button (X).

Figure 29: Compte pharmacien

Après la connexion à un compte pharmacien un menu s'affiche avec tout en haut l'option « Informations du compte » qui permet la modification des informations en relation avec le compte tel que le nom et le prénom, les informations sur la pharmacie mais aussi la modification du mot de passe et la suppression définitive du compte.

L'option « Informations de géolocalisation » quant à elle, permet de modifier l'emplacement de la pharmacie en cas de changement de local.

Quant à l'option « Informations sur les horaires », elle permet la modification des horaires d'ouvertures et de fermetures et la saisie des horaires et dates de gardes et de congé.

4.8. Conclusion

Dans ce dernier chapitre nous avons présenté les différents outils qui nous ont permis de mettre au point notre système mais aussi les différentes interfaces de notre site web afin de faciliter et de mettre au clair les étapes de son utilisation.

Conclusion générale et perspectives

Ce projet de fin d'étude consiste à réaliser un site web dynamique de géolocalisation et de recherche de pharmacies de garde.

Au cours de ce mémoire, nous avons présenté les différentes étapes de conception et de réalisation de notre application.

La réalisation de ce projet nous a permis d'apprendre à nous organiser, d'apprendre à travailler en groupe, d'améliorer nos connaissances et nos compétences dans le domaine de la conception et modélisation nous avons pu apprendre à mieux maîtriser le langage UML mais aussi le domaine de la programmation. Nous avons appris à mieux manipuler les langages PHP, HTML, MYSQL et JavaScript.

En effet, ce travail étant une première expérience, n'est donc pas un modèle unique et parfait, car par faute de temps nous n'avons malheureusement pas pu tout accomplir (Envoi d'ordonnance et d'avis), aussi, Google ont récemment annoncé que l'utilisation de l'API Google Maps allait devenir payante à partir du 11 Juin, L'API devient certes payante, mais une fois dépassé un certain nombre de requête par jour seulement, ce qui fait que l'application ne pourra être opérationnel si le nombre de requête dépasse ce nombre.

Nous restons ouverts à toutes les critiques et nous sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer davantage cette initiative. Etant donné que tout travail informatique abouti a été toujours l'œuvre d'une équipe.

Webographie

1. MILLOGO, F. (2012). Memoire Online - Mise en place d'une application webmapping de géolocalisation des points d'intérêt de la ville de Ouagadougou - Frédéric MILLOGO. [En Ligne] Memoire Online. Disponible sur: <https://www.memoireonline.com/05/13/7195/Mise-en-place-dune-application-webmapping-de-geolocalisation-des-points-dintert-de-la-vill.html> [Consulté le 2 Mar. 2018].
2. Eveil-delaconscience.com. (2016). La géolocalisation dans notre vie quotidienne. [En Ligne] Disponible sur: <http://www.eveil-delaconscience.com/la-geolocalisation-dans-notre-vie-quotidienne> [Consulté le 4 Mar. 2018].
3. Chassaing, É. (2009). Schéma de principe de la géolocalisation par GPS. [Image] Disponible sur: <https://fr.wikipedia.org/wiki/%C3%A9olocalisation#/media/File:Geolocation.png> [Consulté le 4 Mar. 2018].
4. Le Telegramme. (2003). Qu'est ce qu'un système d'information géographique (SIG) ?. [En Ligne] Disponible sur: <http://www.letelegramme.fr/ar/viewarticle1024.php?aaaammjj=20030920&article=6792270&type=ar> [Consulté le 5 Mar. 2018].
5. Géolocalisation véhicules. (S.d.). La géolocalisation par satellite : comment ça marche ? - Géolocalisation véhicules. [En Ligne] Disponible sur: <https://geolocalisation-vehicule.fr/geolocalisation-par-satellite/> [Consulté le 5 Mar. 2018].
6. Géolocalisation satellite. (S.d.). [Image] Disponible sur: <http://anr-prodige.com/uploads/GeolocalisationSatellite.png> [Consulté le 5 Mar. 2018].
7. Géolocalisation EOTD. (S.d.). [Image] Disponible sur: <http://anr-prodige.com/uploads/GeolocalisationEOTD.png> [Consulté le 6 Mar. 2018].
8. Géolocalisation véhicules. (S.d.). Technologies de géolocalisation : la localisation par gsm - Géolocalisation véhicules. [En Ligne] Disponible sur: <https://geolocalisation-vehicule.fr/localisation-par-gsm/#0> [Consulté le 6 Mar. 2018].
9. Géolocalisation Cell-ID. (s.d.). [Image] Disponible sur: <http://anr-prodige.com/uploads/GeolocalisationCellID.png> [Consulté le 6 Mar. 2018].

10. Géolocalisation Triangulation. (S.d.). [Image] Disponible sur: <http://anr-prodige.com/uploads/GeolocalisationTriangulation.png> [Consulté le 6 Mar. 2018].
11. Fr.wikipedia.org. (s.d.). Géolocalisation. [En Ligne] Disponible sur: <https://fr.wikipedia.org/wiki/G%C3%A9olocalisation> [Consulté le 10 Mar. 2018].
12. Géolocalisation Wifi et Traçage Wifi. (S.d.). [Image] Disponible sur: https://connect.ed-diamond.com/var/diamond/storage/images/connect/misc/mischs-008/wifi_privacy-article/wifi_geoloc/436688-1-fre-FR/WiFi_geoloc1.png [Consulté le 12 Mar. 2018].
13. Bluenote-systems.com. (s.d.). Géolocalisation adresse IP, définition et fonctionnement. [Online] Disponible sur: <https://www.bluenote-systems.com/faq-crm-sugarcrm/geolocalisation-adresse-ip-definition.html> [Consulté le 15 Avr. 2018].
14. Capture d'écran. (2018). [Image] Disponible sur: <http://geolocaliser-ip.com/> [Consulté le 16 Mar. 2018].
15. Brunet, P. (2011). App iPhone Plyce. [Image] Disponible sur: <https://www.ithink.fr/startup/plyce-bons-plans-proximite/> [Consulté le 6 Avr. 2018].
16. Interface YAssir. (S.d.). [Image] Disponible sur: <https://s-i.huffpost.com/gen/5585190/original.jpg> [Consulté le 6 Avr. 2018].
17. Winpharm. (2016). [Image] Disponible sur: https://fibladi.com/media/k2/items/src/91_winpharm.png [Consulté le 6 Avr. 2018].
18. Responsive web design. (S.d.). [Image] Disponible sur: <https://classically.me/blogs/playing-media-queries-and-responsive-web-design> [Consulté le 2 Jun. 2018].
19. AROCAS, V. (2016). Comparatif de 10 des frameworks utilisés en Front-end. | SUPINFO, École Supérieure d'Informatique. [En Ligne] Supinfo.com. Disponible sur: <https://www.supinfo.com/articles/single/2694-comparatif-10-frameworks-utilises-front-end> [Consulté le 2 Jun. 2018].
20. AUDIBERT, L. (2013). UML 2 - de l'apprentissage à la pratique. [En Ligne] Developpez.com. Disponible sur: <https://laurent-audibert.developpez.com/Cours-UML/?page=introduction-modelisation-objet> [Consulté le 3 May 2018].
21. Cheheb, S. et Benabdelaziz, K. (2000). Memoire Online - Conception et réalisation d'un site web pour le département d'informatique - Cheheb Sonia Benabdelaziz Katia. [Online]

Memoire Online. Disponible sur: https://www.memoireonline.com/01/13/6844/m_Conception-et-realisation-dun-site-web-pour-le-departement-dinformatique18.html [Consulté le 4 Avr. 2018].

22. RAISSI, A. (2013). Conception et développement d'un site web de e-commerce pour le compte de LSAT_Nokia [En Ligne] Mastère en nouvelles technologies de télécommunication et réseaux. Tunis : Université Virtuelle de Tunis. Disponible sur : <http://docplayer.fr/451277-Memoire-de-mastere-pour-obtenir-le-mastere-en-nouvelles-technologies-de-telecommunication-et-reseaux-theme.html> [Consulté le 2 Avr. 2018].

23. Fr.wikipedia.org. (s.d.). Diagramme des cas d'utilisation. [En Ligne] Disponible sur: https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation [Consulté le 9 May 2018].

24. Docwiki.embarcadero.com. (2013). Définition des diagrammes de séquence UML 1.5 — RAD Studio. [En Ligne] Disponible sur: http://docwiki.embarcadero.com/RADStudio/Tokyo/fr/Définition_des_diagrammes_de_séquence_UML_1.5 [Consulté le 13 Mar. 2018].

25. Tichit, L. (s.d.). Le modèle d'architecture Modèle – Vue – Controleur Application à Qt4. [Ebook] Luminy. Disponible sur: http://www.dil.univ-mrs.fr/~tichit/gl/2011-2012/09_MVC_Qt4.pdf [Consulté le 6 Mai 2018].

26. Soysouvanh, V. (2014). Le MVC. [Image] Disponible sur: <http://www.spotcrea.fr/outil/le-mvc> [Consulté le 6 Mai 2018].

27. Florian Perrier - Développement web Paris. (2015). Facilitez votre conception grâce aux Frameworks CSS - Florian Perrier - Développement web Paris. [En Ligne] Disponible sur: <http://www.florianperrier.com/blog/2015/09/facilitez-votre-conception-grace-aux-frameworks-css/> [Consulté le Avr. 2018].

28. Google Developers. (s.d.). Service Geocoding | Google Maps JavaScript API | Google Developers. [En Ligne] Disponible sur: <https://developers.google.com/maps/documentation/javascript/geocoding?hl=FR> [Consulté le 17 Mar. 2018].

29. Google Developers. (S.d.). Service Distance Matrix | Google Maps JavaScript API | Google Developers. [En Ligne] Disponible sur: <https://developers.google.com/maps/documentation/javascript/distancematrix?hl=fr> [Consulté le 18 Mar. 2018].

30. Scheithauer, A. (2008). Historique d'UML. [Image] Disponible sur: [https://fr.wikipedia.org/wiki/UML_\(informatique\)#/media/File:OO-histoire.jpg](https://fr.wikipedia.org/wiki/UML_(informatique)#/media/File:OO-histoire.jpg) [Consulté le 6 Mai 2018].
31. Formation-uml.fr. (S.d.). Diagrammes structurels - Formation UML. [En Ligne] Disponible sur: <http://www.formation-uml.fr/langage/diagrammes-structurels/> [Consulté le 28 Mar. 2018].
32. Potte, J. (2011). La hiérarchie des diagrammes UML 2.0 sous forme d'un diagramme de classes. [Image] Disponible sur: [https://fr.wikipedia.org/wiki/UML_\(informatique\)#/media/File:Uml_diagram-fr.png](https://fr.wikipedia.org/wiki/UML_(informatique)#/media/File:Uml_diagram-fr.png) [Consulté le 29 Avr. 2018].
33. S.Crozat. (2016). Passage UML Relationnel : classes et associations [En Ligne] Compiègne : Université de Technologie de Compiègne, 19 pages, Disponible sur: <https://stph.scenari-community.org/idl-bd/idl-bd4.pdf> [Consulté le 2 Jun. 2018].
34. Geocaching, F. (n.d.). France Geocaching - Géocodage et conversions GPS. [En Ligne] France-geocaching.fr. Disponible sur: <https://france-geocaching.fr/articles.php?lng=fr&pg=451&mnuid=211&tconfig=0> [Consulté le 6 May 2018].

Annexe A

1. Annexe

1.1. UML

L'OMG (Object Management Group) définit UML comme suit :

« Le langage UML (Unified Modeling Language) est un langage graphique permettant de visualiser, de spécifier, de construire et de documenter les artefacts d'un système à forte composante logicielle. L'UML offre un moyen standard d'écrire les plans d'un système, y compris les concepts tels que les processus métier et les fonctions système, ainsi que des éléments concrets tels que les instructions de langage de programmation, les schémas de base de données et les composants logiciels réutilisables. »

A. Historique

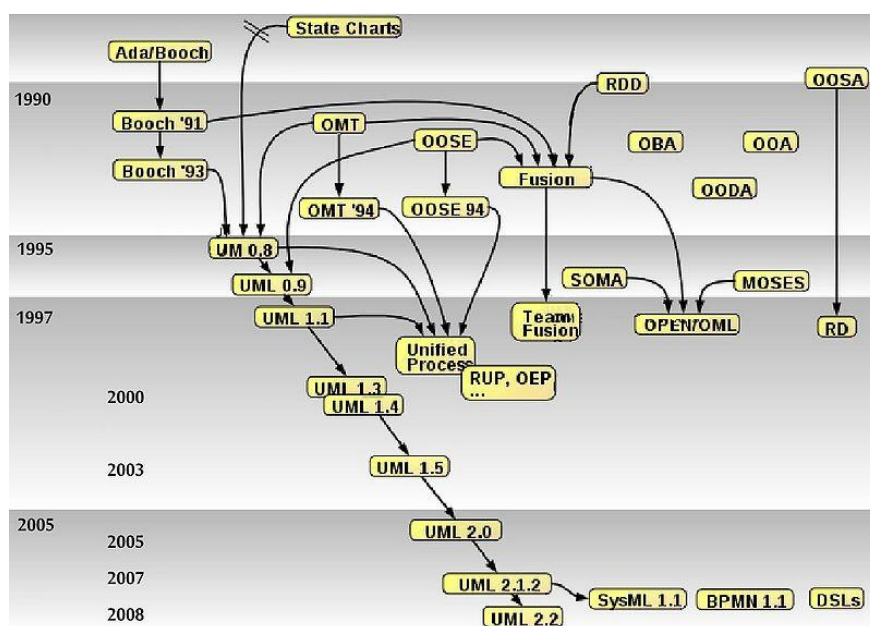


Figure 30 : Historique sur l'UML [30]

B. Diagrammes UML

La modélisation d'un système ne s'appuie pas seulement à comprendre son fonctionnement mais plutôt à comprendre son dynamisme et sa sémantique. C'est dans ce sens qu'une panoplie de diagrammes a été proposée ces diagrammes sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. Il en existe quatorze depuis UML 2.3.

- **Diagrammes structurels** : Ils permettent de présenter une vue des éléments statiques du système : les objets en présence, les classes d'objets, les composants... Il n'est pas ici question de modéliser de quelle manière le système se comporte, mais de quels éléments il est constitué ^[31].
- **Diagrammes de comportement** : Ils focalisent sur la description de la partie dynamique du système à modéliser et des processus et des événements chronologiques.
- **Diagrammes d'interactions** : Ils permettent de modéliser comment les objets communiquent entre eux et tendent à traduire un fonctionnement dans un contexte particulier et l'accent est mis spécifiquement sur la séquence des opérations plutôt que sur les données qui sont véhiculées.

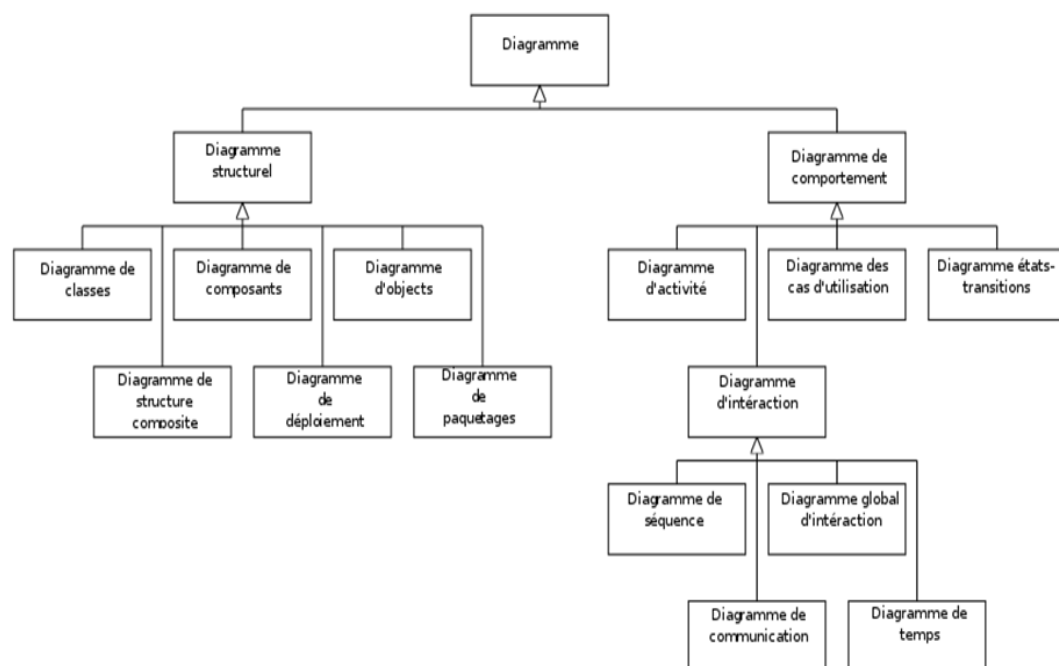


Figure 31 : La hiérarchie des diagrammes UML 2.0 sous forme d'un diagramme de classes ^[32]

Pour faciliter notre tâche nous avons recours au langage de modélisation unifié. Ce langage est né de la fusion de plusieurs méthodes existantes auparavant, car c'est devenu une référence en terme de modélisation objet, à un tel point que sa connaissance devienne indispensable pour un développeur.

1.2. Règles de passage au modèle relationnel ^[33]

Pour faire cela on doit suivre des règles de transformation bien précise

A. Transformation des classes :

Pour chaque classe non abstraite et classe porteuse d'informations permanentes,

- On crée une relation dont le schéma est celui de la classe ;
- La clé primaire de cette relation est un des attributs de la classe.

B. Transformation des attributs :

Pour chaque attribut d'une classe on crée un attribut correspondant dans notre schéma.

C. Transformation des méthodes :

On ne représente pas les méthodes dans le modèle relationnel, ils seront calculés dynamiquement soit par des procédures internes à la base de données, soit par des procédures au niveau applicatif.

D. Transformation des associations :

Dans notre diagramme on a des associations de type 1..N (1..*) et de type 1..1.

Pour chaque association binaire de type 1..N :

- On ajoute à la relation côté N une clé étrangère vers la relation côté 1.

Si la cardinalité est exactement 1 (1..1) côté 1, alors on ajoutera une contrainte de non nullité sur la clé étrangère,

La solution la plus simple et la plus générale pour transformer une association 1..1 consiste à traiter cette association 1..1 comme une association 1..N, puis à ajouter une contrainte UNIQUE sur la clé étrangère pour limiter la cardinalité maximale à 1.

1.3. Diagramme de séquence « Authentication »

