

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Béjaia

Faculté des Sciences Exactes

Département d'Informatique



Mémoire de Fin de Cycle

En Vue de l'Obtention du Diplôme de Master Professionnel en Informatique
Option: Génie Logiciel

Thème

Conception et Réalisation d'une Application
Mobile Dédiée Aux Corps Médicaux

Réalisé par:

LAHDIR Lydia

AKKOUCHE Seghira

Devant le jury composé de:

Président: Mme ADEL Karima

Examineur: Mme KHOULALENE Nadjat

Examinatrice: Mlle AIT HATRIT Fatima

Encadreur: Mme BRAHAMI née EL BOUHISSI Houda

Mr BEN BRAHIM Seifeddine

Promotion 2018

Remerciements

En préambule à ce mémoire nous remercions **ALLAH** qui nous a donné la force et le savoir pour réaliser ce travail.

Aucune œuvre humaine ne peut se réaliser sans la contribution d'autrui.

Ce projet est le résultat d'un effort constant, cet effort n'aurait pas pu aboutir sans la contribution d'un nombre de personnes.
Ainsi se présente l'occasion de les remercier.

Un grand merci à toutes nos familles surtout nos parents pour leur encouragement et leur suivi avec patience de déroulement de notre projet.

Nous tenons à remercier également notre promotrice
Mme EL BOUHISSI HOUDA, et notre promoteur
Mr BEN BRAHIM SEIFEDDINE
d'avoir accepté de nous guider tout au long de ce travail.

Nous tenons à remercier également **Mr CHRISTIAN PERSON**,
Mme ALLAM DIANA, **Mr LAHDIR MUSTAPHA**
Et toute l'équipe d'UMALIS.

Nous tenons aussi à remercier tous les membres de jury pour nous avoir honorés en acceptant d'évaluer notre travail.

Enfin, Nous tenons à exprimer nos vifs remerciement à tous ceux qui ont collaboré avec leurs idées, leurs expériences et ont participé de près ou de loin à la réalisation de ce travail.

Dédicaces

J'ai le plaisir de dédier ce modeste travail :

A mes très chers parents en témoignage de ma grande affection et ma gratitude pour tous les sacrifices qu'ils ont consentie pour notre réussite et leurs soutiens indéfectibles pour l'obtention du diplôme de master.

Que dieu les gardes et leur procure la santé et le bonheur.

A ma grand-mère,

A mes sœurs : «Feriel, Sabrina et Sonia»,

A mon frère « Abderrahmane »,

A Mon oncle «Mustapha »,

A Mon binôme « Seghira »,

A Ma chère copine « Khelfaoui Lydia »,

A mes tantes, mes oncles, cousins et cousines,

A toute ma famille ,

A tous mes amis (es),

A tout le personnel du département Informatique,

Et tous les personnes que j'aime et je n'ai pas cité leur noms.

Lydia

J'ai le plaisir de dédier ce modeste travail :

A mes très chers parents en témoignage de ma grande affection et ma gratitude pour tous les sacrifices qu'ils ont consentie pour notre réussite et leurs soutiens indéfectibles pour l'obtention du diplôme de master.
Que dieu les gardes et leur procure la santé et le bonheur.

A mes frères : «Idir, Massinissa et Yanis»,

A Mon binôme «Lydia»,

A mes tantes, mes oncles, cousins et cousines,

A toute ma famille,

A tous mes amis (es),

A tout le personnel du département Informatique,

Et tous les personnes que j'aime et je n'ai pas cité leur noms.

Seghira

Liste des abréviations

API: Application Programming Interfaces.

BTS: Base Transceiver Station.

GPS: Global Positioning System.

GMS : Global System for Mobile.

HTML : Hypertext Markup Language.

iOS : Internetworking Operating System.

IP : Internet Protocol.

JSON : JavaScript Object Notation.

JS : JavaScript .

NoSQL : Not Only SQL.

OHA : Open Handset Alliance .

PC : Personal Computer. .

PDA : Personal Digital Assistant .

SQL : Structured Query Language .

SDK : Software Devloppement Kit.

SA: Societé Anonyme .

SOAP : Simple Object Access Protocol.

UML : Unified Modeling Language .

UMTS: Universal Mobile Télécommunication System .

UP : Unified Process .

UDDI: Universal Description Discovery and Integration.

WSDL: Web Services Description Language.

Liste des figures

2.1	Les versions d'ANDROID	7
2.2	Organigramme du cycle de vie d'une Application ANDROID [10]	8
3.1	Les diagrammes d'UML	13
3.2	Le schéma synthétique de UP	15
3.3	Diagramme de Gantt	16
3.4	Le diagramme général de cas d'utilisation	18
3.5	Diagramme de séquence de cas d'utilisation « créer un compte »	27
3.6	Diagramme de séquence de cas d'utilisation « s'authentifier »	28
3.7	Diagramme de séquence de cas d'utilisation « modifier le profil »	29
3.8	Diagramme de séquence de cas d'utilisation « Activer GPS »	30
3.9	Diagramme de séquence de cas d'utilisation « géolocaliser local »	31
3.10	Diagramme de séquence de cas d'utilisation « rechercher un local »	32
3.11	Diagramme de séquence de cas d'utilisation « prendre un rendez-vous »	33
3.12	Diagramme de séquence de cas d'utilisation « gérer rendez-vous »	34
3.13	Le diagramme d'activité du cas d'utilisation « S'authentifier »	35
3.14	Le diagramme d'activité du cas d'utilisation « Géolocaliser un local »	36
3.15	Diagramme de classe de conception	38
3.16	Diagramme de déploiement de l'application	39
3.17	Exemple du format orienté documents	42
4.1	Interface Bienvenue	49
4.2	Interface d'accueil	49
4.3	Interface liste des hôpitaux	50
4.4	Détails hôpitaux	50
4.5	liste pharmacies	51
4.6	détails pharmacie	51
4.7	interface médecins	52
4.8	interface médecins	52
4.9	Prendre un rendez-vous	53
4.10	Interface d'authentification médecin	53
4.11	profile-médecin	54
4.12	inscription médecin	54
4.13	Test authentification	55
4.14	Interface d'accueil	57

Liste des tableaux

3.1	identification des acteurs du système	17
3.2	Description textuelle de cas d'utilisation « créer un compte »	19
3.3	Description textuelle de cas d'utilisation « Authentification »	20
3.4	Description textuelle de cas d'utilisation « modifier le profil »	21
3.5	Description textuelle de cas d'utilisation « Activer GPS »	22
3.6	Description textuelle de cas d'utilisation « Géolocaliser un local »	23
3.7	Description textuelle de cas d'utilisation « Rechercher un local »	24
3.8	Description textuelle de cas d'utilisation « Prendre un rendez-vous »	25
3.9	Description textuelle de cas d'utilisation « Gérer rendez-vous »	26
3.10	Dictionnaire de données	37
3.11	Comparaison entre SGBD Relationnels NoSQL [20]	40

Table des matières

Liste des abréviations	I
Liste des figures	II
Table des matières	IV
1 <i>Introduction et motivation</i>	1
1.1 Contexte et problématique	1
1.2 Présentation de l'organisme d'accueil	2
1.3 Objectifs	2
1.4 Méthodologie de travail	3
1.5 Organisation du mémoire	3
2 <i>La technologie du mobile</i>	4
2.1 Introduction	4
2.2 L'Informatique mobile	4
2.2.1 Applications mobiles	4
2.2.2 Domaines d'application mobiles	5
2.2.3 Avantages d'une application mobile	5
2.3 Système Android	6
2.3.1 Historique	6
2.3.2 Versions d'Android	6
2.3.3 Cycle de vie d'une application Android	7
2.3.4 Avantages d'Android	9
2.4 La géolocalisation :	9
2.4.1 Les techniques de géolocalisation	9
2.4.2 Usages pratiques de la géolocalisation par GPS	10
2.5 Conclusion	11
3 <i>Analyse des besoins et conception</i>	12
3.1 Introduction	12
3.2 Conduite du projet	12
3.2.1 Choix de la méthode de conception	12
3.2.2 Le processus unifié	14
3.2.3 Gestion de temps	15
3.3 Spécification des besoins	16
3.3.1 Identification des acteurs du système	16
3.3.2 Identification des besoins	17

3.4	Analyse des besoins	18
3.4.1	Le diagramme général de cas d'utilisation	18
3.4.2	Description textuelle des cas d'utilisation	19
3.5	Conception du projet	27
3.5.1	Diagrammes de séquences	27
3.5.2	Le diagramme d'activité	35
3.5.3	Elaboration du diagramme de classe	37
3.5.4	le diagramme de déploiement	38
3.6	Création de la base de données	39
3.7	Conclusion	42
4	<i>Expérimentation</i>	43
4.1	Introduction	43
4.2	Environnement de développement	43
4.2.1	Environnement matériel	43
4.2.2	Environnement logiciels	44
4.3	Langages de développement	47
4.4	Les services web utilisés	47
4.4.1	Google Places API Web Service	48
4.4.2	Google Places Details API	48
4.4.3	Google Maps Directions API	48
4.5	Description des interfaces réalisées	49
4.5.1	Interface Bienvenue	49
4.5.2	Interface accueil	49
4.5.3	Interface hôpital	50
4.5.4	Interface détail-hôpital	50
4.5.5	Interface pharmacie	51
4.5.6	Interface détail-pharmacie	51
4.5.7	Interface médecins	52
4.5.8	Interface détails médecin	52
4.5.9	Interface prendre rendez-vous	53
4.5.10	Interface d'authentification médecin	53
4.5.11	Interface profile médecin	54
4.5.12	Interface créer un compte	54
4.6	Conclusion	56
5	<i>Conclusion générale</i>	58
5.1	Introduction	58
5.2	Travail réalisé	58
5.3	Apport du mémoire	58
5.4	Problèmes rencontrés	59
5.5	Perspectives et travaux futurs	59

Chapitre 1

Introduction et motivation

1.1 Contexte et problématique

Ces dernières années, la téléphonie mobile a été sans doute le secteur le plus dynamique, le plus rentable et le plus innovant de toute l'Industrie des Télécommunications.

D'un simple téléphone portable pour émettre des appels à un téléphone évolué doté de capacités proches d'un véritable ordinateur appelé Smartphone dont la principale utilité est de pouvoir communiquer à distance, tout en pouvant se déplacer.

Aujourd'hui ces périphériques ont atteint une puissance de calcul, une taille mémoire ainsi qu'un débit nécessaire pour faire tourner des applications aussi diverses que variées qui vont de d'une application mobile simple jusqu'aux applications de navigation GPS.

Etant donné la forte croissance du marché du mobile et des applications mobiles, le développement d'applications mobiles intéresse énormément d'utilisateurs et il est reconnu dans la plupart des domaines y compris les domaines de la santé. En effet, les logiciels et les applications mobiles dans le domaine de la santé connaissent actuellement un essor important. Leur utilisation se démultiplie et ils peuvent être très variés.

En particulier, à la ville de Bejaia, selon le meilleur de nos connaissances et à nos jours, nous avons remarqué un manque d'outils efficace pour la recherche des corps médicaux en particulier les pharmacies pour l'achat de médicaments et ceci pour les personnes dépourvus de véhicules de transport.

Durant le stage mené au sein de l'entreprise UMALISTunisia que nous allons présenter dans le paragraphe suivant, il nous a été demandé de faire la conception et le développement d'une application mobile qui permet de trouver des corps médicaux en cas de besoin (pharmacies, hôpitaux et médecins privés) et prise de rendez-vous chez un médecin.

L'origine de ce sujet était une simple idée pour fournir des informations concises et pertinentes sur les corps médicaux dans la wilaya de Bejaia facilement accessible en cas de besoin. Au fur et à mesure cette idée a évolué pour concevoir un système de recherche des corps médicaux et prise de rendez-vous chez un médecin.

1.2 Présentation de l'organisme d'accueil

UMALIS Tunisia est une société de développement d'applications informatiques. Elle réunit dans ses locaux de jeunes professionnels dont les activités sont principalement axées sur la réalisation de sites web et d'applications mobiles.

UMALIS Tunisia est une filiale de Umalis Group (affilié à Paris) qui est un groupe de sociétés de portage salarial en pleine expansion, et qui porte la forme juridique d'une société anonyme(SA).Ce groupe a été créé en 2008 et il est composé de 6 filiales: UmalisInternational,UmalisResearch, Umalis Fit, Umalis Vip, Sysalia et UmalisTunisia.

Le groupe accompagne des experts en informatique souhaitant développer une activité professionnelle sécurisée en toute indépendance en bénéficiant des avantages du statut de salarié. Ce groupe est organisé en réseau autour d'une plateforme informatique destinée à simplifier et rationaliser la gestion du portage salarial des consultants.

Parmi les activités de UMALIS Tunisia :

- Le consulting et l'adaptation des solutions pour satisfaire les besoins de communication du groupe UMALIS et de ses différentes filiales.
- Le développement des applications web et mobiles pour les différents projets du groupe et assurer la maintenance logicielle.
- La publication des informations pertinentes du groupe et ses filiales sur les réseaux sociaux ainsi que sur les différents sites d'actualités.
- L'amélioration du référencement des sites web du groupe et ses filiales sur les moteurs de recherche.

1.3 Objectifs

Notre mémoire de fin d'étude est intitulé « Conception et réalisation d'une application mobile dédiée aux corps médicaux ».

Notre travail consiste à explorer les solutions existantes relatives à la recherche des corps médicaux et concevoir et à développer une application mobile sous Android s'adaptant à nos besoins.

Cette application permet de faciliter la recherche des corps médicaux (cabinets médicaux, hôpitaux et pharmaciens) et de pouvoir les localiser sur la carte ainsi que la prise de rendez-vous chez un médecin au besoin.

En résumé notre application contient deux volets. Un volet « *utilisateur* » qui est un espace client permettant aux clients de faire une recherche des corps médicaux.

Un autre volet « *médecin* » qui est un espace réservé aux médecins pour gérer leurs rendez-vous, en acceptant ou déclinant une demande de rendez-vous.

L'outil que nous proposons, avec la version actuelle s'avère une solution simple, innovante et à la fois efficace.

Ce type d'application métier s'avère très utile non seulement afin de subvenir aux besoins des personnes et de leur faciliter la recherche des corps médicaux mais il peut aussi représenter un réel avantage pour les patients. Il s'agit de rapprocher le médecin et faciliter la prise des rendez-vous sans aucun effort ou déplacement.

1.4 Méthodologie de travail

Dans ces grandes lignes, la démarche adoptée pour notre travail est guidée par de nombreuses questions issues des préoccupations de la communauté des utilisateurs des applications dédiées à la santé.

Etant donné que la recherche des corps médicaux et la prise de rendez-vous est un processus parfois difficile, long et fastidieux, nous nous sommes forcés tout au long de ce mémoire de prendre suffisamment de recul afin de proposer un cadre méthodologique relativement global et suffisamment complet afin de mieux aider et mieux guider les utilisateurs dans leur processus.

Notre démarche de travail repose plus précisément sur les étapes suivantes :

- Étape de recherche et d'analyse : qui établit un état de l'art des différentes solutions proposées dans le cadre de la recherche des corps médicaux.
- Étape d'identification du problème et la solution : qui permet de définir la problématique et la solution proposée en vigueur.
- Étape d'implémentation et d'expérimentation du système proposé : qui met en évidence le système proposé, son fonctionnement et son intérêt, accompagnée d'études de cas pour la validation.

1.5 Organisation du mémoire

Le reste du mémoire est organisé comme suit :

- **Chapitre 2:** Ce chapitre sera réservé à la présentation des notions fondamentales utiles pour la mise en œuvre du mémoire.
- **Chapitre 3:** Le troisième chapitre présente la spécification des besoins qui nous permet de décrire sans ambiguïté l'application à développer avec une modélisation formelle à travers des diagrammes UML et un processus de développement adéquat.
- **Chapitre 4:** Ce chapitre est dédié à la réalisation, où nous présentons l'environnement et les outils de développement utilisés pour la réalisation et la mise en œuvre de notre application avec des copies d'écran du système proposé.
- **Chapitre 5:** Enfin, nous concluons avec le chapitre 5 et nous présentons quelques perspectives en vue d'améliorer le système proposé.

Chapitre 2

La technologie du mobile

2.1 Introduction

Les technologies de l'information et de la communication évoluent à une vitesse vertigineuse, c'est à cet effet qu'un centre d'innovation technologique a été inauguré offrant ainsi des vitesses supérieures à la téléphonie mobile.

Dans ce chapitre, nous allons commencer par introduire l'informatique mobile, ses types ainsi que ses dispositifs appropriés.

Nous passons par la suite au système d'exploitation Android qui constitue le noyau de notre application, ensuite nous verrons les techniques de géolocalisation et d'orientation nécessaires pour la réalisation de notre travail.

2.2 L'Informatique mobile

L'informatique mobile traite l'ensemble des solutions informatiques développées sur des plateformes comme les téléphones portables, les Smartphones, les netbooks, les tablettes, etc. L'informatique mobile permet à un utilisateur de conserver une partie de ces outils numériques tout en lui apportant de nouveaux services. Elle lui permet d'interagir avec son environnement via de nombreux canaux comme les technologies sans-fils, etc[1].

2.2.1 Applications mobiles

Une application mobile est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation d'un smartphone ou d'une tablette.

Les applications mobiles sont adaptées aux différents environnements techniques des smartphones et à leurs contraintes et possibilités ergonomiques (comme l'écran tactile). Elles permettent généralement un accès plus confortable et plus efficace à des sites ou services accessibles par ailleurs en versions mobiles ou web.

Nous distinguons différents types d'applications mobiles : les applications web, les applications hybrides et les applications natives[2].

- Applications web mobiles (webapps)

En opposition à une application native, une WebApps est une application mobile développée avec les outils de développement web actuel : HTML5, CSS3 et JavaScript. C'est une application qui une fois développée est accessible et exécutable sur tous les smartphones via leur navigateur web.

L'avantage de ces applications, c'est le gain de temps et d'argent réalisé grâce à leur développement unique et leur déploiement multiplateformes. Dans un cas, vous développez une seule application alors que dans l'autre, vous développez trois applications (pour Android, IOS et Windows Phone [3]).

- **Applications natives**

Une application native est une application développée spécifiquement pour une seule plateforme, grâce aux outils conçus pour celle-ci. Elle est développée avec un langage spécifique à son système d'exploitation et est distribuée uniquement par l'intermédiaire de son store (AppStore IOS, PlayStore pour Android, ...). Pour développer une application pour deux plateformes différentes, il vaut mieux de développer deux applications distinctes[3].

- **Applications hybrides**

Les applications hybrides sont des applications qui combinent les éléments d'une WebApps et les éléments d'une application native. Elles reposent essentiellement sur la solution Cordova/PhoneGap, cette solution sert de passerelle entre le langage web et le natif. Cette solution nous permet d'utiliser un seul et même outil pour le développement et les langages issus du développement Web pour tous les mobiles (IOS, Android et Windows Phone)[3].

2.2.2 Domaines d'application mobiles

Avec les possibilités matérielles incorporées aux terminaux (camera, GPS, gyroscope, tactile, écran large), les applications Smartphones et Tablettes peuvent intégrer des fonctionnalités spécifiques et dédiées pour les utilisateurs comme [4] :

- Les jeux mobiles
- Les automatismes industriels
- Le GPS et les services permettant la localisation
- Les opérations bancaires
- Les suivis des commandes, l'achat de billets
- Des applications médicales mobiles
- L'écoute et la visualisation des multimédia
- La consultation d'Internet

2.2.3 Avantages d'une application mobile

- Une parfaite ergonomie est assurée pour les applications mobiles en comparaison aux sites mobiles cela encourage les utilisateurs à demeurer fidèles aux applications. En effet, le développement d'application mobile tient compte la taille du smartphone, le temps de chargement et autres paramètres.

- Les applications mobiles favorisent l'intégration des options de téléphone et ainsi, l'expérience utilisateur devient plus développée.
- Pas besoin d'avoir accès à l'internet pour que l'application fonctionne. Facile à trouver sur les stores par rapport aux sites mobiles, les applications mobiles ont connu ainsi un usage plus répandu auprès des jeunes surtout qu'elles notifient sur les événements en cours [5].

2.3 Système Android

Android est le système d'exploitation mobile créé par Google. Il équipe la majorité des téléphones portables du moment (smartphones). Son principal concurrent est Apple avec l'iPhone. Android est un système vous permettant de personnaliser votre téléphone, télécharger des applications (navigateur Internet, GPS, Facebook...). Android équipe également les tablettes tactiles [6].

2.3.1 Historique

En juillet 2005, Google a acquis Android, Inc., une petite startup qui développait des applications pour téléphones mobiles. C'est à ce moment-là que des rumeurs sur l'entrée de Google dans le secteur du mobile ont commencé. Mais personne n'était sûr, dans quels marchés ils allaient se positionner.

Après ce rachat, à Google, une équipe dirigée par Andy Rubin, un ancien d'Android Inc, a commencé à travailler sur un système d'exploitation pour appareil mobile basé sur linux. Durant 2 ans, avant que l'OHA soit créé officiellement, un certain nombre de rumeurs ont circulé au sujet de Google. Il a été dit que Google développait des applications mobiles de son moteur de recherche, qu'ils développaient un nouveau téléphone mobile, etc.

En 2007, le 5 novembre, l'OHA a été officiellement annoncée, ainsi que son but : développer des standards open source pour appareil mobile.

Le premier standard annoncé a été Android, une plateforme pour appareils mobiles basée sur un kernel linux 2.6. En septembre 2008, la première version stable du SDK a vu le jour, actuellement, nous sommes à la version 1.2 [7].

2.3.2 Versions d'Android

Dans l'ensemble, les différentes versions d'Android ont toutes des noms de desserts (en anglais), qui sont sculptés et affichés devant le siège social de Google (mountain view) et ça depuis la sortie de la version 1.5 et suivent une logique alphabétique (de A à Z) [8].



FIG. 2.1 – Les versions d'ANDROID
[9]

2.3.3 Cycle de vie d'une application Androïd

Pour des raisons de priorité d'activités, le système Android peut tuer une activité quand il a besoin de ressources. Pour cette raison, aucune activité ne peut penser pouvoir vivre jusqu'au bout de son traitement. Une activité possède quatre états [10]:

- **«Active»** : l'activité est lancée par l'utilisateur, elle s'exécute au premier plan ;
- **«Pause»** : l'activité est lancée par l'utilisateur, elle s'exécute et est visible, mais elle n'est plus au premier plan. Une notification ou une autre activité lui a volé le focus et une partie du premier plan ;
- **«Stoppée»** : l'activité a été lancée par l'utilisateur, mais n'est plus au premier plan et est invisible. L'activité ne peut interagir avec l'utilisateur qu'avec une notification ;
- **«Morte»** : l'activité n'est pas lancée

L'organigramme suivant (FIG.2.2)indique le cycle de vie d'une activité et les méthodes appelées lors des changements d'état .

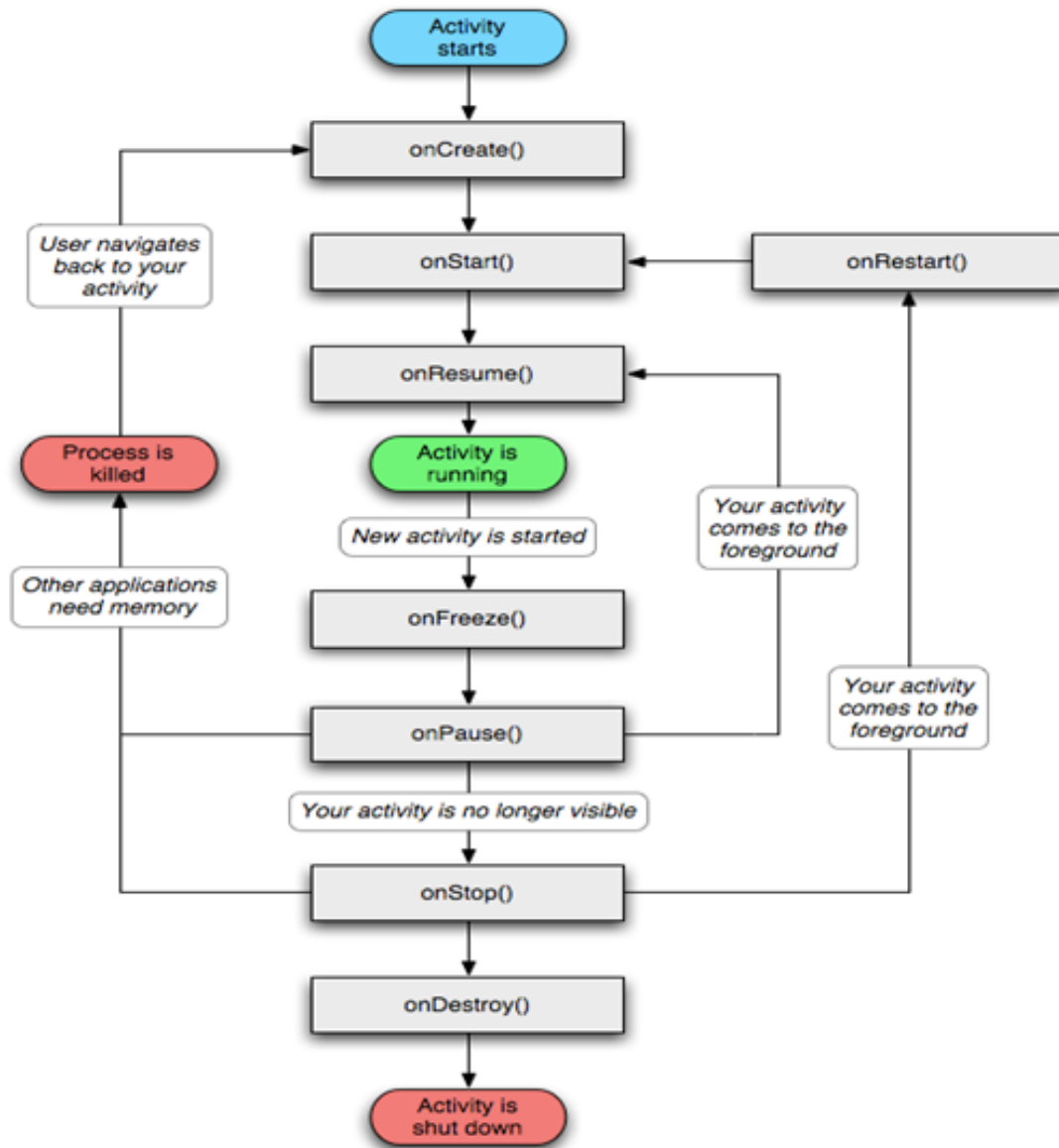


FIG. 2.2 – Organigramme du cycle de vie d'une Application ANDROID [10]

Les différentes étapes de l'organigramme précédent sont :

- Démarrage de l'Activité: la méthode `onCreate` est appelée, Pendant l'utilisation d'une activité, l'utilisateur presse la touche Accueil, ou bien l'application téléphone, qualifiée comme prioritaire et qui interrompt son fonctionnement par un appel téléphonique entrant.
- Arrêt de l'Activité (appel de `onStop`), le développeur détermine l'impact sur l'interface utilisateur, par exemple la mise en pause d'une animation puisque l'activité n'est plus visible.
- Une fois l'appel téléphonique terminé, le système réveille l'activité précédemment mise en pause (appel de `onRestart`, `onStart`).
- L'activité reste trop longtemps en pause, le système a besoin de mémoire, il détruit l'activité (appel de `onDestroy`).

OnPause et onResume rajoutent un état à l'activité, puisqu'ils interviennent dans le cas d'activité partiellement visible, mais qui n'ont pas le focus. La méthode onPause implique également que la vie de cette application n'est plus une priorité pour le système. Donc si celui-ci a besoin de mémoire, l'activité peut être fermée. Ainsi, il est préférable, lorsque l'on utilise cette méthode, de sauvegarder l'état de l'activité dans le cas où l'utilisateur souhaiterait y revenir avec la touche Accueil[11].

2.3.4 Avantages d'Android

- **Open source:**

Android utilise des bibliothèques open source puissantes, comme par exemple SQLite ou hyper file SQL pour les bases de données et Open GL pour la gestion d'images 2D et 3D (pour faire des jeux)[12].

- **Facile à développer :**

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès [12].

- **Facile à vendre :**

Le Play Store (anciennement Android Market) est une plateforme immense et très visitée, c'est donc une mine d'opportunités pour qui conque veut diffuser une application dessus [12].

- **Flexible :**

Le système est extrêmement portable, il s'adapte à beaucoup des différentes structures . Les smartphones, les tablettes, la présence ou l'absence de clavier ou de trackball, différents processeurs [12].

- **Complémentaire :**

L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leur développement. Ces applications se trouvent essentiellement sur internet et leur principe est que vous pouvez combiner plusieurs composants totalement différents pour obtenir un résultat surpuissant [12].

2.4 La géolocalisation :

La géolocalisation peut être définie comme étant un procédé permettant de positionner un objet (une personne, un véhicule, ect.) sur un plan ou une carte à l'aide de ses coordonnées[13].

2.4.1 Les techniques de géolocalisation

Cinq techniques permettent de géolocaliser ou de positionner des personnes et des objets sont basées sur les réseaux mobiles GSM et UMTS (2G, 3G), les réseaux fixes Wifi, le réseau

internet et le système GPS.

- **Le positionnement par GSM ou par UMTS**

Lorsqu'un utilisateur s'identifie à un réseau mobile, son numéro *IMEI* (*International Mobile Equipment Identify*) obtenu à partir de la carte SIM, (code unique composé de 15 chiffres), est transmis au système de gestion de l'opérateur afin de pouvoir l'autoriser à utiliser les différentes options disponibles ; un réseau GSM est découpé en de nombreuses cellules avec une BTS (*Base Transceiver Station*) qui prend en charge les communications radios des téléphones mobiles[14].

- **L'identification par cellule**

La première méthode de localisation d'un mobile va s'effectuer à travers l'adresse de la BTS sur laquelle il est connecté (CellID, identification de cellule). Très simple la précision de positionnement du mobile va varier de 250 mètres en zone urbaine à 10 kilomètres en zone rurale[14].

- **L'identification par triangulation**

La deuxième méthode établit un rapport de distance d'un mobile entre trois stations BTS. par la valeur de la puissance du signal radio reçu du mobile;, la précision va dépendre de la densité des antennes[14].

- **L'identification par temps**

La troisième méthode liée à l'émission d'un signal du mobile mesure le temps écoulé entre l'émission et la réception du signal[14].

- **Par adresse IP**

le mobile s'identifie sur le réseau avec une adresse IP unique fixe ou temporaire. La précision du positionnement est la village ou le quartier[14].

- **Par Wi-Fi**

La géolocalisation d'un appareil mobile donne une précision de 50 à 100 mètres. Pour un positionnement à 5 mètres, une triangulation entre différentes antennes Wi-Fi avec analyse de la puissance du signal radio est effectuée [14].

- **Par GPS**

Le positionnement par GPS s'effectue par la réception de signaux provenant de plusieurs satellites se trouvant en orbite par trilatération (technique proche de la triangulation)[14].

2.4.2 Usages pratiques de la géolocalisation par GPS

De nombreux professionnels utilisent déjà la géolocalisation dans le cadre de leurs activités. Ce système possède différentes fonctions applicables dans une multitude de secteurs. Son utilisation est un facteur d'optimisation dans certains domaines cruciaux de l'entreprise [15].

- **Lutte contre le vol :**

Un système GPS permet de repérer le véhicule ou l'article volé en temps réel n'importe où dans le monde grâce à un simple coup de téléphone et facilite l'intervention de la police. De plus, le récepteur peut être équipé d'un coupe-circuit qui éteint le moteur à distance [15] .

- **Guidage assisté :**

Grâce à la géolocalisation, la tâche de tous les travailleurs de la route, chauffeurs, livreurs, commerciaux, ambulanciers... ainsi que celle des marins est facilitée par le système de guidage GPS qui leur indique clairement l'itinéraire à suivre engendrant un gain de temps et donc d'argent [15].

- **Surveillance des employés :**

Placé dans un véhicule professionnel, le traceur GPS permet à l'employeur de vérifier en temps réel la position de l'employé durant ses heures de travail. L'appareil de géolocalisation peut être muni d'un micro ou d'un détecteur de survitesse [15].

2.5 Conclusion

Tout au long de ce chapitre, nous avons introduit l'informatique mobile dans son contexte général, ainsi que le système d'exploitation Android qui est le noyau de notre application. Nous avons ensuite énuméré les techniques utilisées pour réaliser ce mémoire, pour garantir une meilleure compréhension du sujet et les fonctionnalités du système utiliser.

Dans le chapitre suivant, nous allons détailler la conception de notre application en expliquant d'une manière claire et précise la démarche utilisée.

Chapitre 3

Analyse des besoins et conception

3.1 Introduction

La conception est une étape fondamentale dans le cycle de vie d'une application informatique. En effet c'est d'elle que dépendent la qualité et la cohérence du produit réalisé au développement. Des méthodes de génie logiciel ont alors été développées afin de guider le concepteur dans sa tâche.

Pour mener à bien ce travail, il est nécessaire de définir une méthodologie de travail. Dans cette partie nous présentons les choix conceptuels en termes de méthode de conception et le cycle de vie de logiciel.

3.2 Conduite du projet

3.2.1 Choix de la méthode de conception

Le plus grand avantage d'une méthode orientée objet est qu'elle permet de structurer un système sans se focaliser sur l'analyse uniquement sur les données ou uniquement sur les traitements mais sur les deux à la fois. Une telle approche a pour but de modéliser les propriétés statiques et dynamiques de l'environnement du système. Elle met en correspondance le problème et la solution, en préservant la structure et le comportement du système analysé.

Ceci, nous a conduit à adopter l'approche orientée objet pour modéliser notre système en se basant sur les diagrammes UML [17].

Les Vues UML

UML fournit un moyen astucieux permettant de présenter diverses projections d'une même représentation grâce aux vues. Une vue est constituée d'un ou plusieurs diagrammes. On distingue deux types de vues [17]: (FIG.3.1)

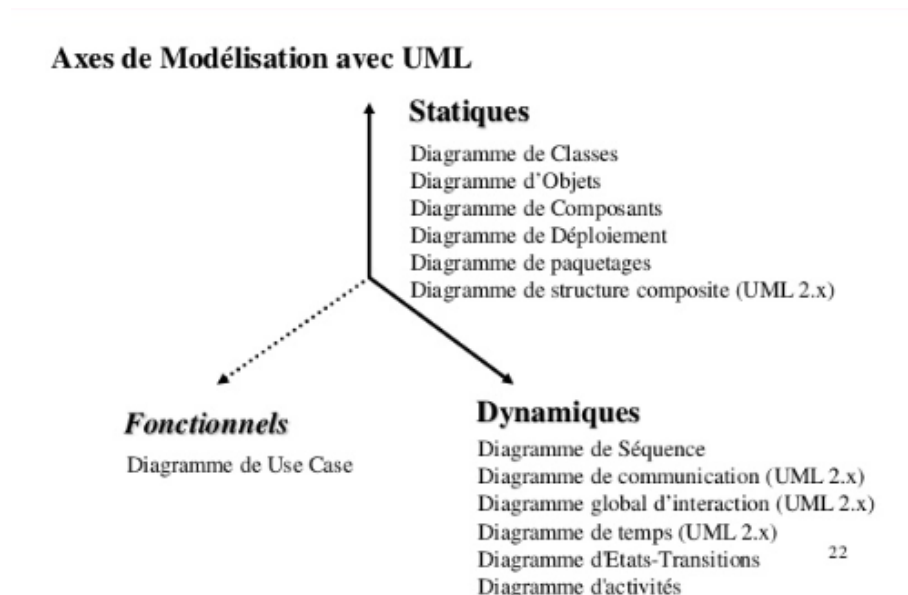


FIG. 3.1 – *Les diagrammes d'UML*
[17]

a) Les vues statiques

Elles permettent de représenter le système physiquement. On trouve alors les diagrammes suivants :

- **Le Diagramme de classes** : il représente la structure statique en terme de classes et de relations entre elle, il représente aussi un ensemble d'interface et de paquetages ainsi que leur relations.
- **Le Diagramme d'objets** ou le diagramme d'instances représente une instance possible du diagramme de classes.
- **Le Diagramme de composant** : représente les morceaux d'applications packagés sous la forme de composants disposants d'interfaces. Il permet de décrire ces composants qui sont : le sous-système, le module, le programme et le sous-programme, le processus et la tâche.
- **Le Diagramme de déploiement** : complémentaire du diagramme de composants, il décrit la répartition physique des instances de composants, de processus et d'objets d'une application distribuée.

b) Les vues dynamiques

Les cinq diagrammes comportementaux (ou dynamiques) représentent des vues dynamiques du système :

- **Le Diagramme de cas d'utilisation** : sont des vues qui décrivent les interactions entre les différents acteurs externes (utilisateurs du cas) et les fonctionnalités du système. La description de l'interaction est réalisée suivant le point de vue de l'utilisateur. Leur but est d'identifier les acteurs du domaine, leurs responsabilités respectives et de décrire leurs besoins.
- **Le Diagramme de collaboration** : il décrit l'interaction modélisée par les échanges de messages entre objets ou entre acteurs et objets.

- **Le Diagramme de séquence** : il diffère légèrement du diagramme de collaboration par l'ajout d'une dimension temporelle en précisant la chronologie des échanges de messages entre les objets.
- **Le Diagramme d'états transitions** : il décrit l'ensemble des états des objets du système et les transitions qui déclenchent le passage d'un état donné vers un autre état.
- **Le Diagramme d'activités** : est une variante des diagrammes d'états-transitions. Il décrit l'ensemble des activités effectuées par les acteurs du système en les décomposant en sous-activités et en spécifiant les contraintes relatives à l'enchaînement de ces dernières.

3.2.2 Le processus unifié

Nous avons choisi d'adopter le processus unifié comme étant la méthode de l'étude conceptuelle. En effet, cette méthode englobe l'ensemble des activités exigées par un projet logiciel à travers un ensemble de principes génériques, adaptables en fonction des spécificités des projets.

Donc c'est un Framework de processus générique pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de projets.

Ce processus présente plusieurs avantages, que nous citons :

- Limiter les coûts, en termes de risques, aux strictes dépenses liées à une seule itération.
- Limiter les risques de retard de mise en place du produit développé.
- Accélérer le rythme de l'ensemble du développement.
- Prendre en compte le fait que les besoins des utilisateurs et les exigences correspondantes ne puissent être intégralement définies à l'avance.

En fait, le Processus Unifié (UP) est un processus de développement logiciel « **itératif et incrémental, centré sur l'architecture, guidé par les cas d'utilisation et piloté par les risques** » :

- **Itératif et incrémental** : le projet est découpé en itérations ou étapes de courte durée qui permettent de mieux suivre l'avancement global. A la fin de chaque itération une partie exécutable du système finale est produite de façon incrémentale (par ajout).
- **Centré sur l'architecture** : tout système complexe doit être décomposé en partie modulaire afin d'en faciliter la maintenance et l'évolution. Cette architecture (Fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML, et pas seulement documentée en texte.
- **Piloté par les risques** : les risques majeurs du projet doivent être identifiés au plutôt mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre d'itération.
- **Guidé par les cas d'utilisation** : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et classés par priorité.

La gestion d'un tel processus est organisée selon les quatre phases suivantes :
Initialisation, élaboration, construction et transition comme le montre la figure :

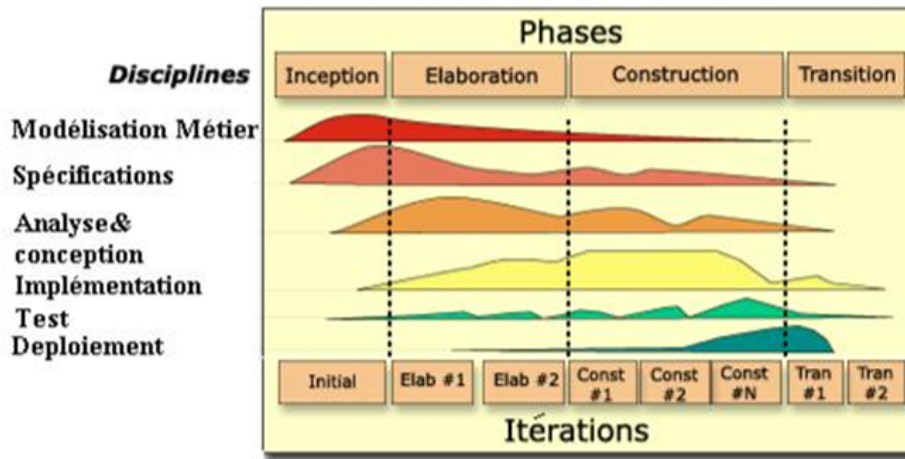


FIG. 3.2 – *Le schéma synthétique de UP*
[18]

La phase d'initialisation conduit à définir la « vision » du projet, sa portée, sa faisabilité, son « business case », pour décider au mieux de sa poursuite ou de son arrêt.

La phase d'élaboration poursuit trois objectifs principaux en parallèle :

- Identifier et décrire la majeure partie des besoins utilisateurs ;
- Construire (et pas seulement décrire dans un document) l'architecture de base du système ;
- Lever les risques majeurs du projet.

La phase de construction consiste surtout à concevoir et à implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base). C'est la phase la plus consommatrice en ressources et en efforts.

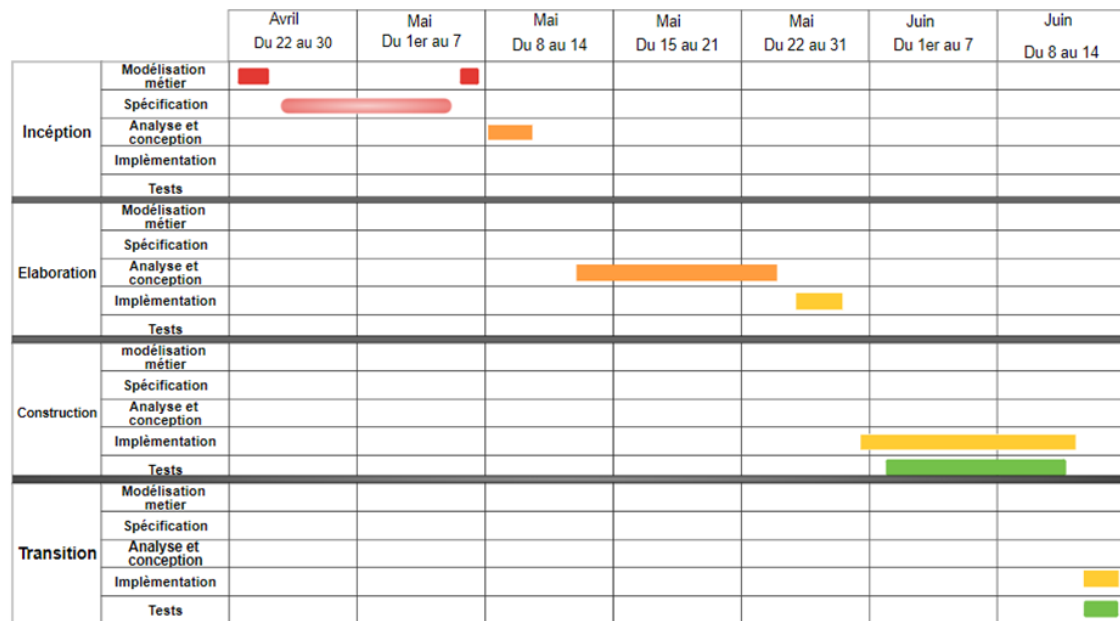
Enfin, **la phase de transition** permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux. Les mots-clés en sont : conversion des données, formation des utilisateurs, déploiement, bêta-tests.

Les activités de développement sont définies par cinq disciplines fondamentales qui décrivent la capture des exigences, l'analyse et la conception, l'implémentation, le test et le déploiement. La modélisation métiers est une discipline en amont, optionnelle et transverse aux projets. PU doit donc être compris comme une trame commune des meilleures pratiques de développement et non comme l'ultime tentative d'élaborer un processus universel[18].

3.2.3 Gestion de temps

Nous avons utilisé le diagramme de Gantt qui est un outil standard de gestion de projet utilisé pour planifier efficacement les tâches et suivre ensuite leurs progressions. Il offre une représentation graphique du projet en planifiant les tâches individuelles sur un calendrier. Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin[20].

La figure ci-dessous (FIG.3.3) illustre la planification des tâches



FIG. 3.3 – *Diagramme de Gantt*


3.3 Spécification des besoins

3.3.1 Identification des acteurs du système

Un acteur représente une personne, un matériel ou un logiciel qui interagit directement avec le système en question. Un acteur peut consulter et/ou modifier directement l'état du système en émettant ou recevant des messages susceptibles d'être porteurs de données.

Les acteurs qui interagissent avec notre système sont les suivants :

Acteur	Cas d'utilisation
	<ul style="list-style-type: none"> – Inscription – Authentification – Gérer les rendez-vous – Modifier le profil

	<ul style="list-style-type: none"> – Géolocaliser les pharmacies, les hôpitaux, les médecins privée et afficher leurs informations – Prendre un rendez-vous chez un médecin privé
---	---

TAB. 3.1 – *identification des acteurs du système*

3.3.2 Identification des besoins

• Les besoins fonctionnels

Dans cette partie nous détaillons l'ensemble des fonctionnalités que l'application doit offrir aux utilisateurs. En effet, le système à réaliser doit répondre aux besoins fonctionnels suivants :

- a. Authentification du médecin : possède un login et un mot de passe spécifique qui lui permet de vérifier son identité, afin d'autoriser l'accès de cette entité à des ressources en toute sécurité.
- b. Inscription médecin en cas ou il ne possède pas un compte.
- c. Gestion de son compte (modification de ses informations).
- d. des rendez-vous Le médecin peut consulter la liste des rendez-vous et répondre aux patient par téléphone à leur demande.
- e. localiser des pharmacies et hôpitaux sur la carte par apport à la position d'utilisateur.
- f. afficher les informations de pharmacie ou hôpital.
- g. Localiser un médecin.

• Les besoins non fonctionnels

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système.

Les principaux besoins non fonctionnels de notre application se résument dans les points suivants :

- a. Le code doit être clair pour permettre des futures évolutions ou améliorations ;
- b. L'ergonomie : l'application doit offrir une interface conviviale et facile à utiliser ;
- c. La sécurité : l'application doit respecter la confidentialité des données ;
- d. Garantir l'intégrité et la cohérence des données à chaque mise à jour et à chaque Insertion [20].

3.4 Analyse des besoins

3.4.1 Le diagramme général de cas d'utilisation

Dans cette partie, nous traduisons les besoins exprimés auparavant en un diagramme UML, ce diagramme est le diagramme de cas d'utilisation general suivant:



FIG. 3.4 – *Le diagramme général de cas d'utilisation*

3.4.2 Description textuelle des cas d'utilisation

Le tableau suivant présente les différentes interactions entre les acteurs pour le cas d'utilisation « créer un compte »

Sommaire d'identification	
Cas d'utilisation	<ul style="list-style-type: none"> • créer un compte
Acteur	<ul style="list-style-type: none"> • Medecin
Objectif	<ul style="list-style-type: none"> • Créer un compte avec lequel le médecin accède aux fonctionnalités dont il a le droit
Description des enchainements	
Pre-condition	—
Post-condition	<ul style="list-style-type: none"> • Compte créé
Scénario nominal	
<ul style="list-style-type: none"> • Le médecin demande l'inscription • Le système affiche le formulaire d'inscription • L'utilisateur remplit le formulaire et valide • Le système vérifie la validité des informations [A1] • Le système envoie une demande de vérification d'existence du médecin dans la BDD [A2] • Le système crée un compte pour le médecin 	
Enchainement alternatif [A1]	
<ul style="list-style-type: none"> • Si l'information n'est pas, le système affiche un message d'erreur et réaffiche l'interface d'inscription. 	
Enchainement alternatif [A2]	
<ul style="list-style-type: none"> • Si le médecin existe déjà, le système affiche un message d'erreur et réaffiche l'interface d'authentification. 	

TAB. 3.2 – *Description textuelle de cas d'utilisation « créer un compte »*

Le tableau suivant présente les différentes interactions entre les acteurs pour le cas d'utilisation « Authentification »

Sommaire d'identification	
Cas d'utilisation	<ul style="list-style-type: none"> • Authentification
Acteur	<ul style="list-style-type: none"> • Médecin
Objectif	<ul style="list-style-type: none"> • Vérifier que le médecin a bien le droit d'accès à l'application
Description des enchainements	
Pre-condition	<ul style="list-style-type: none"> • Le médecin doit avoir un compte
Post-condition	<ul style="list-style-type: none"> • Accès à la page d'accueil du médecin
Scénario nominal	
<ul style="list-style-type: none"> • Le médecin demande l'accès à l'application • Le système affiche l'interface authentification • Le médecin introduit son identifiant et le mot de passe • Le système vérifie les champs [A1] • Le système envoi une demande de vérification d'existence du médecin dans la BDD [A2] • Le système donne l'accès à l'interface correspondante 	
Enchainement alternatif [A1]	
<ul style="list-style-type: none"> • Si l'information n'est pas valide, le système affiche un message d'erreur et réaffiche l'interface d'authentification. 	
Enchainement alternatif [A2]	
<ul style="list-style-type: none"> • Si le médecin n' existe pas, le système affiche un message d'erreur et réaffiche l'interface d'authentification. 	

TAB. 3.3 – *Description textuelle de cas d'utilisation « Authentification »*

Le tableau suivant présente les différentes interactions entre les acteurs pour le cas d'utilisation « Modifier le profil »

Sommaire d'identification	
Cas d'utilisation	<ul style="list-style-type: none"> • modifier le profil
Acteur	<ul style="list-style-type: none"> • Médecin
Objectif	<ul style="list-style-type: none"> • Mise à jour du profil
Description des enchainements	
Pre-condition	<ul style="list-style-type: none"> • Le médecin doit s'authentifier et avoir un compte
Post-condition	<ul style="list-style-type: none"> • Modifier les informations du médecin
Scénario nominal	
<ul style="list-style-type: none"> • Le médecin demande l'accès à l'application (le médecin s'authentifie) • Le médecin demande le formulaire de modification • Le système affiche le formulaire de modification • Le médecin introduit les nouvelles informations et valide • Le système envoie une demande de modification du médecin dans la BDD [A1] • Le système crée un compte pour le médecin 	
Enchainement alternatif [A1]	
<ul style="list-style-type: none"> • Si l'information n'est pas, le système affiche un message d'erreur et réaffiche l'interface d'inscription. 	

TAB. 3.4 – *Description textuelle de cas d'utilisation « modifier le profil »*

- Cas d'utilisation « géolocaliser un local »

Le tableau suivant présente les différentes interactions entre les acteurs pour le cas d'utilisation « Activer GPS »

Sommaire d'identification	
Cas d'utilisation	• Activer GPS
Acteur	• Utilisateur
Objectif	• Afficher la position de l'utilisateur
Description des enchainements	
Pre-condition	–
Post-condition	• Afficher ma position
Scénario nominal	
<ul style="list-style-type: none"> • L'utilisateur demande sa position • Le système vérifie l'état de GPS [A1] • Le système affiche la position de l'utilisateur sur la carte 	
Enchainement alternatif [A1]	
<ul style="list-style-type: none"> • Si le GPS n'est pas activé le système demande une connexion GPS puis il affiche la position du local sur la carte d'inscription. 	

TAB. 3.5 – *Description textuelle de cas d'utilisation « Activer GPS »*

Le tableau suivant présente les différentes interactions entre les acteurs pour le cas d'utilisation « Géolocaliser un local »

Sommaire d'identification	
Cas d'utilisation	<ul style="list-style-type: none"> • Géolocaliser un local
Acteur	<ul style="list-style-type: none"> • Utilisateur
Objectif	<ul style="list-style-type: none"> • Afficher la position d'un local
Sommaire d'identification	
Pre-condition	<ul style="list-style-type: none"> • Afficher ma position
Post-condition	<ul style="list-style-type: none"> • Afficher la position du local
Scénario nominal	
<ul style="list-style-type: none"> • L'utilisateur demande de Géolocaliser un local • Le système affiche la position de l'utilisateur sur la carte • Le système fait appel au service google places • Le service retourne la position du local • L'utilisateur sélectionne un local • Le système affiche le nom et l'adresse du local 	

TAB. 3.6 – *Description textuelle de cas d'utilisation « Géolocaliser un local »*

Le tableau suivant présente les différentes interactions entre les acteurs pour le cas d'utilisation «Rechercher un local»

Sommaire d'identification	
Cas d'utilisation	<ul style="list-style-type: none"> • Rechercher un local
Acteur	<ul style="list-style-type: none"> • Utilisateur
Objectif	<ul style="list-style-type: none"> • Recherche d'un local
Description des enchainements	
Pre-condition	<ul style="list-style-type: none"> • Afficher ma position
Post-condition	<ul style="list-style-type: none"> • Afficher la position du local en recherche
Scénario nominal	
<ul style="list-style-type: none"> • L'utilisateur saisit une recherche • Le système retourne la position de l'utilisateur • Le système fait appel au service google places [A1] • Le système affiche la position du local en recherche sur la carte 	
Enchainement alternatif [A1]	
<ul style="list-style-type: none"> • Si le local en recherche n'existe pas le système envoi un message d'erreur d'inscription. 	

TAB. 3.7 – *Description textuelle de cas d'utilisation « Rechercher un local »*

Le tableau suivant présente les différentes interactions entre les acteurs pour le cas d'utilisation «Prendre un rendez-vous»

Sommaire d'identification	
Cas d'utilisation	<ul style="list-style-type: none"> • Prendre un rendez-vous
Acteur	<ul style="list-style-type: none"> • Utilisateur
Objectif	<ul style="list-style-type: none"> • Prendre un rendez-vous avec un médecin
Pre-condition	—
Post-condition	<ul style="list-style-type: none"> • Ajouter a la liste des rendez-vous
Scénario nominal	
<ul style="list-style-type: none"> • L'utilisateur demande l'interface prendre rendez-vous • Le système affiche le formulaire de prise de rendez-vous • L'utilisateur introduit les informations et valide • Le système vérifie les champs [A1] • Le système ajoute le rendez-vous a la liste des rendez-vous du médecin 	
Enchainement alternatif [A1]	
<ul style="list-style-type: none"> • Si un champ d'information n'est pas valide, le système affiche un message d'erreur et réaffiche le formulaire de prise de rendez-vous 	

TAB. 3.8 – *Description textuelle de cas d'utilisation « Prendre un rendez-vous »*

Le tableau suivant présente les différentes interactions entre les acteurs pour le cas d'utilisation «Gérer rendez-vous»

Sommaire d'identification	
Cas d'utilisation	<ul style="list-style-type: none"> • Gérer un rendez-vous
Acteur	<ul style="list-style-type: none"> • Médecin
Objectif	<ul style="list-style-type: none"> • Accepter ou refuser un rendez-vous
Pre-condition	<ul style="list-style-type: none"> • Le médecin doit s'authentifier
Post-condition	<ul style="list-style-type: none"> • Rendez-vous confirmé ou décliné
Scénario nominal	
<ul style="list-style-type: none"> • Le médecin demande l'accès à l'application (le médecin s'authentifie) • Le médecin demande la liste des rendez-vous • Le système affiche la liste des rendez-vous • Le médecin vérifie les horaires [A1] 	
Enchaînement alternatif [A1]	
<ul style="list-style-type: none"> • Si les horaires sont disponibles le médecin confirme le rendez-vous sinon il l'annule 	

TAB. 3.9 – *Description textuelle de cas d'utilisation « Gérer rendez-vous »*

3.5 Conception du projet

3.5.1 Diagrammes de séquences

- La figure suivante montre le diagramme de séquence relatif au cas d'utilisation « créer un compte »

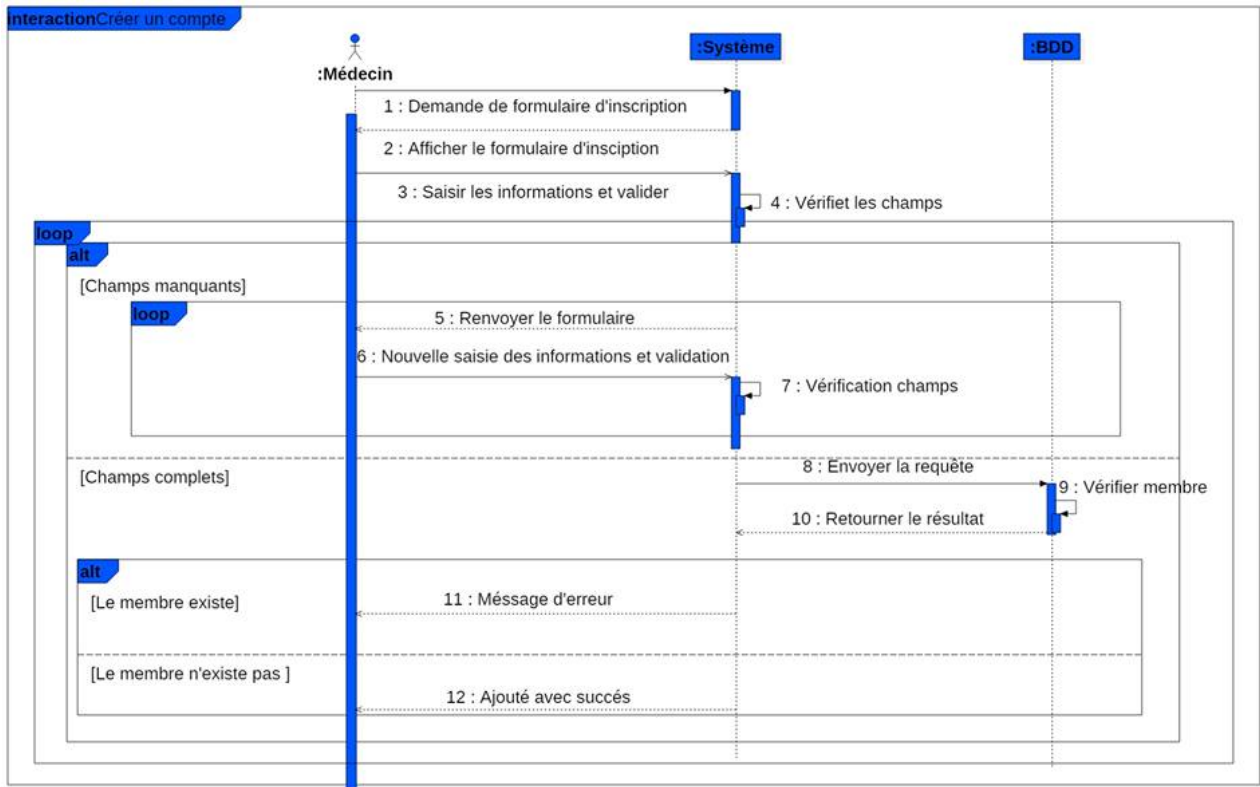


FIG. 3.5 – Diagramme de séquence de cas d'utilisation « créer un compte »

- La figure suivante montre le diagramme de séquence relatif au cas d'utilisation « s'authentifier »

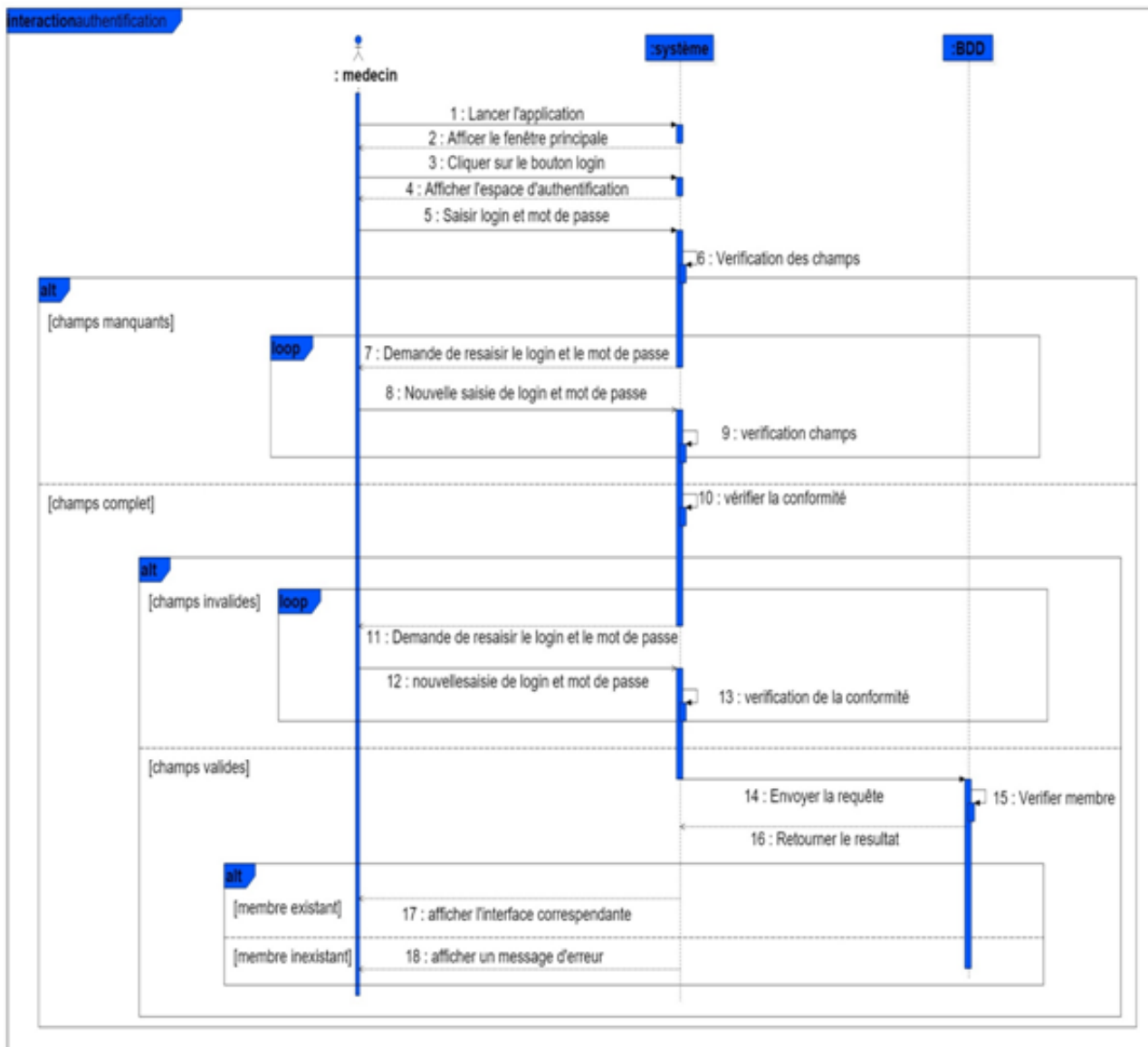


FIG. 3.6 – *Diagramme de séquence de cas d'utilisation « s'authentifier »*

- La figure suivante montre le diagramme de séquence relatif au cas d'utilisation « modifier le profil »

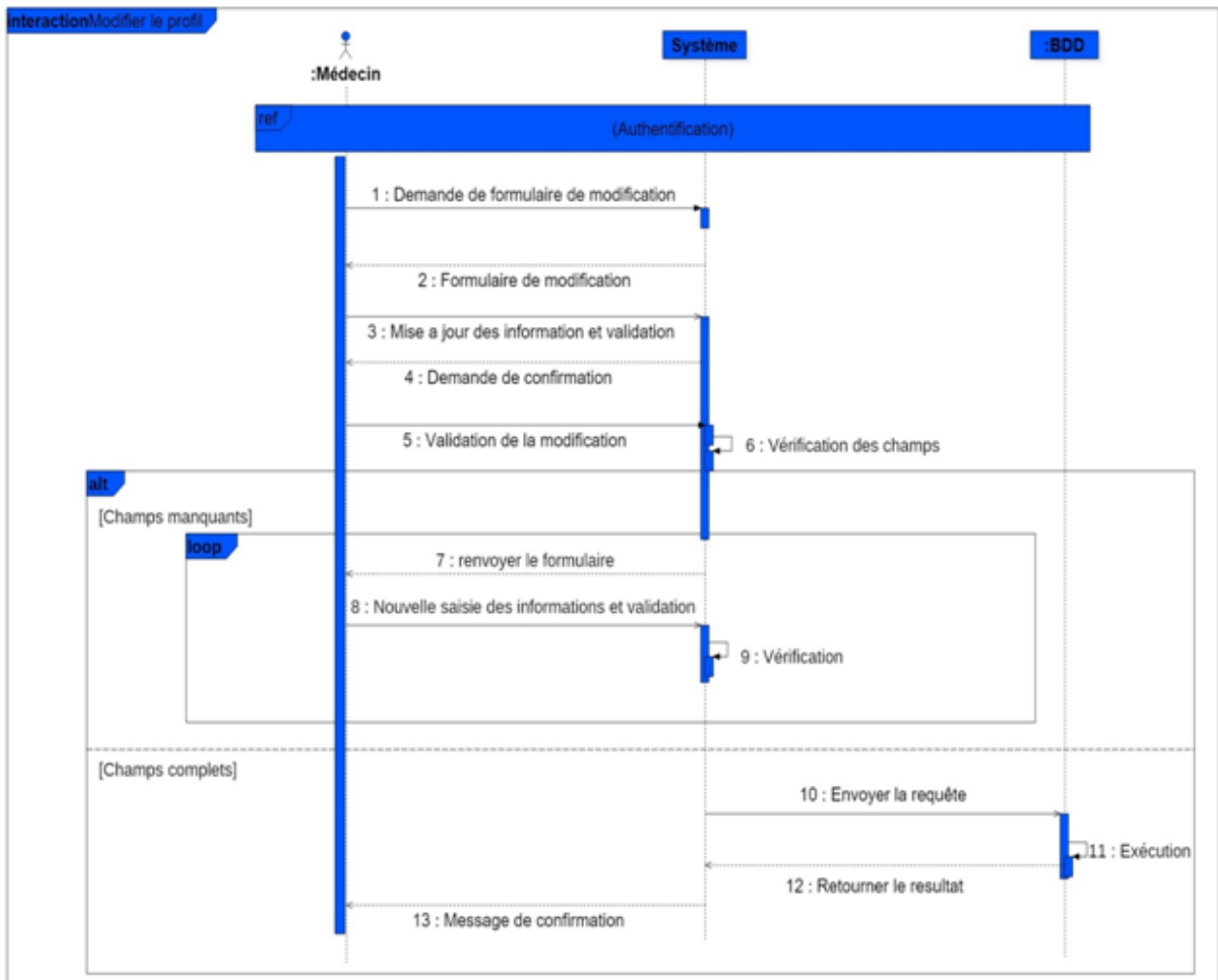


FIG. 3.7 – Diagramme de séquence de cas d'utilisation « modifier le profil »

- La figure suivante montre le diagramme de séquence relatif au cas d'utilisation « Activer GPS »

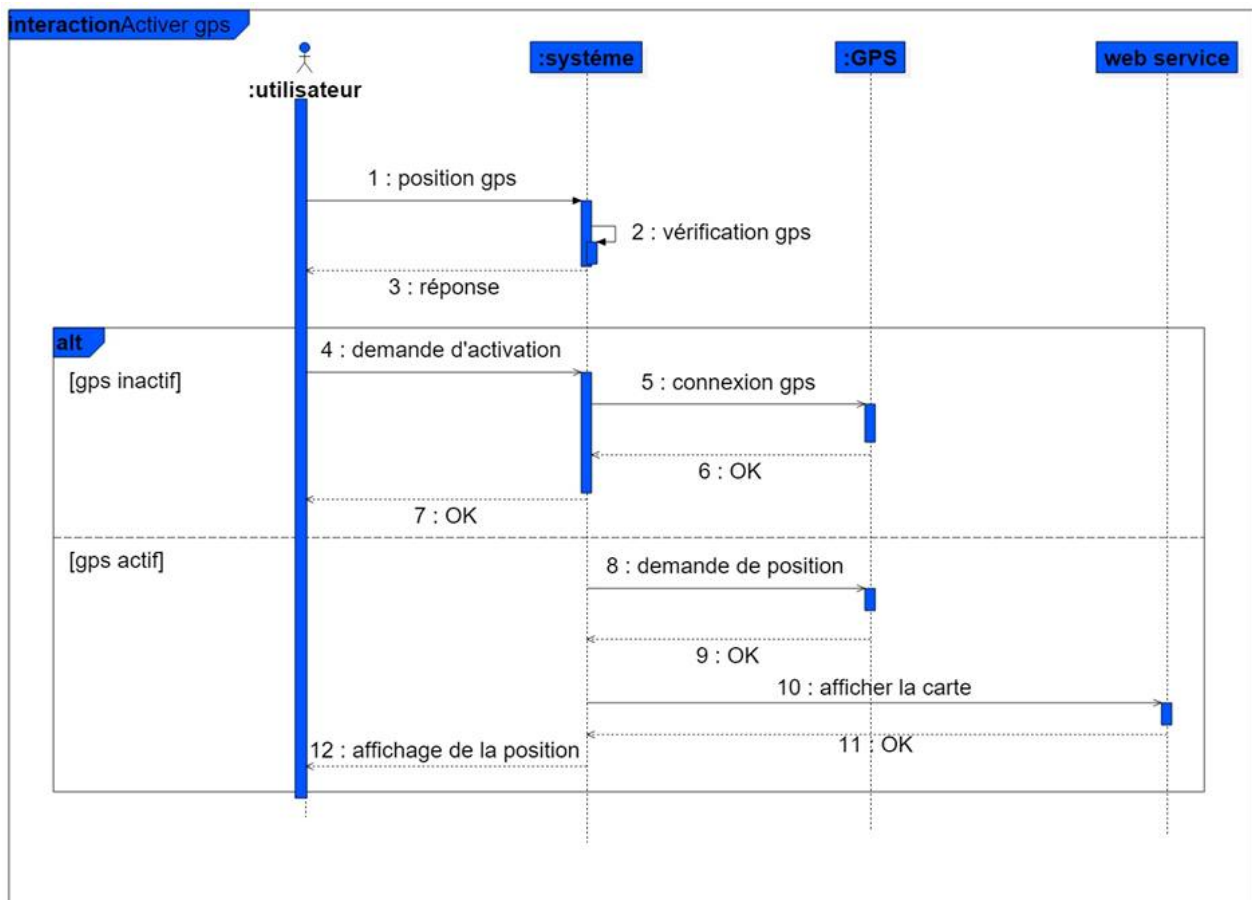


FIG. 3.8 – *Diagramme de séquence de cas d'utilisation « Activer GPS »*

- La figure suivante montre le diagramme de séquence relatif au cas d'utilisation « géolocaliser local »

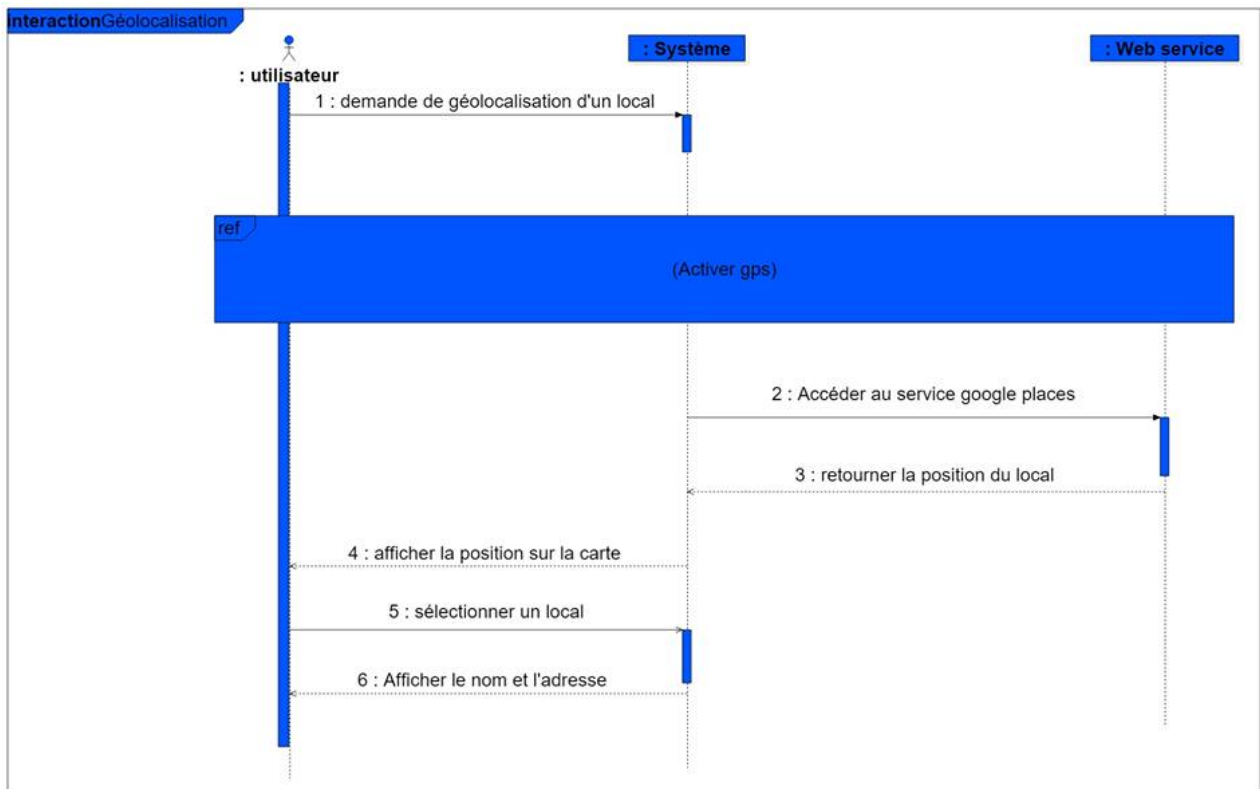


FIG. 3.9 – *Diagramme de séquence de cas d'utilisation « géolocaliser local »*

- La figure suivante montre le diagramme de séquence relatif au cas d'utilisation « rechercher un local »

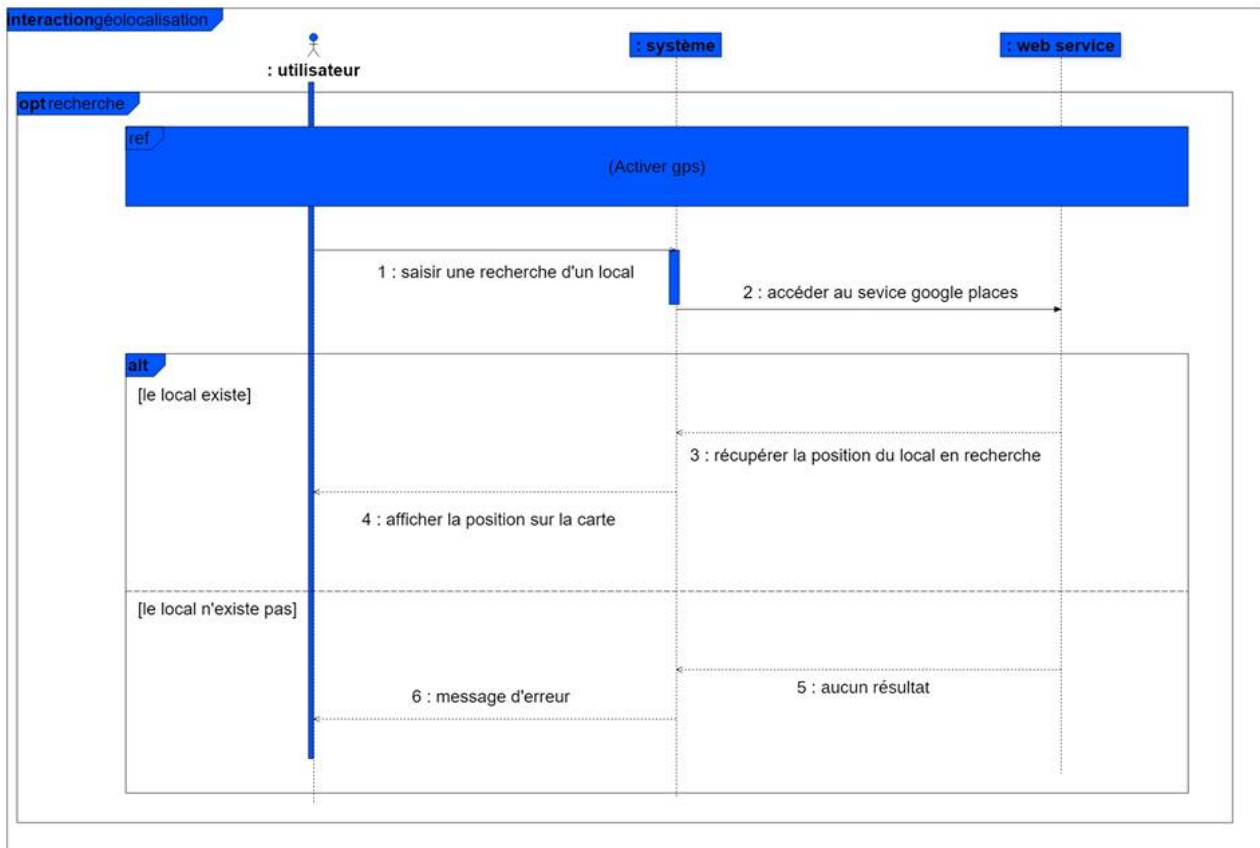


FIG. 3.10 – *Diagramme de séquence de cas d'utilisation « rechercher un local »*

- La figure suivante montre le diagramme de séquence relatif au cas d'utilisation « prendre un rendez-vous »

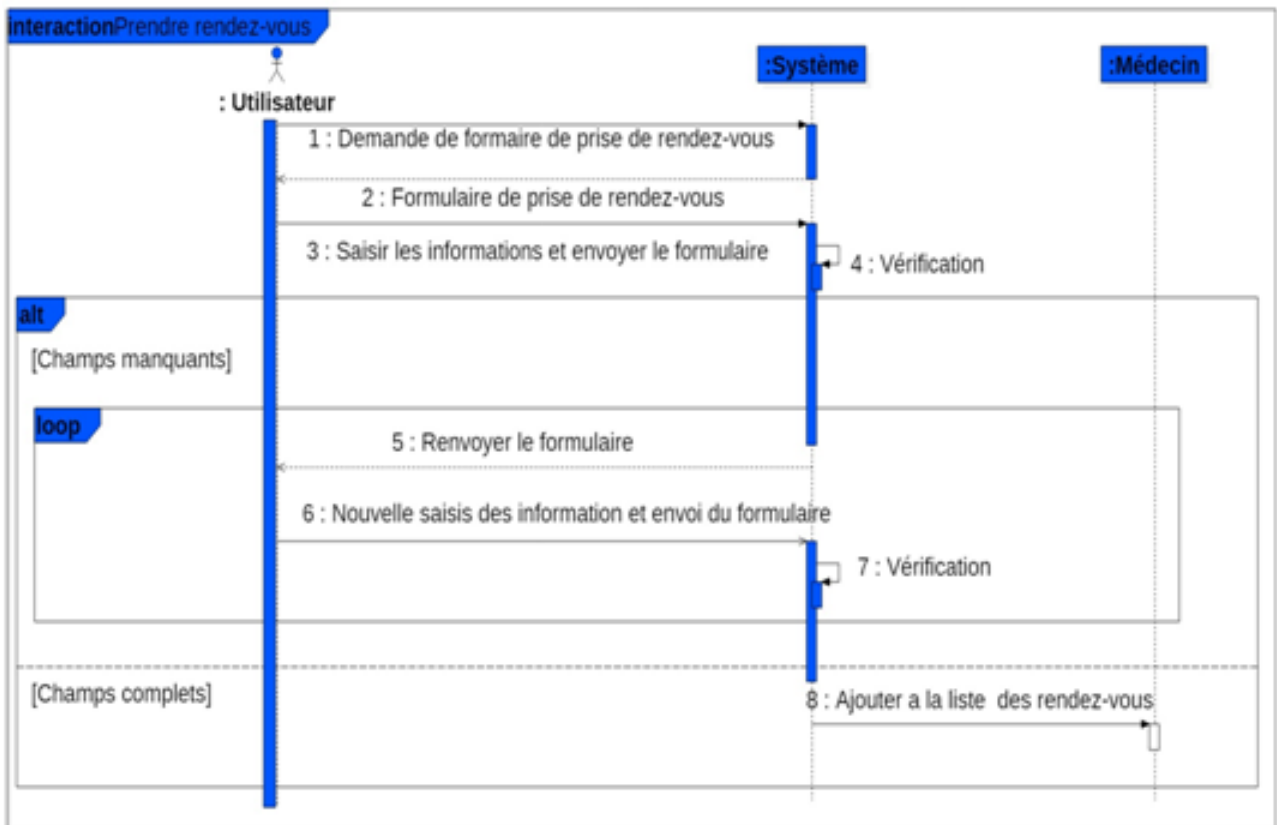


FIG. 3.11 – *Diagramme de séquence de cas d'utilisation « prendre un rendez-vous »*

- La figure suivante montre le diagramme de séquence relatif au cas d'utilisation « gérer rendez-vous »

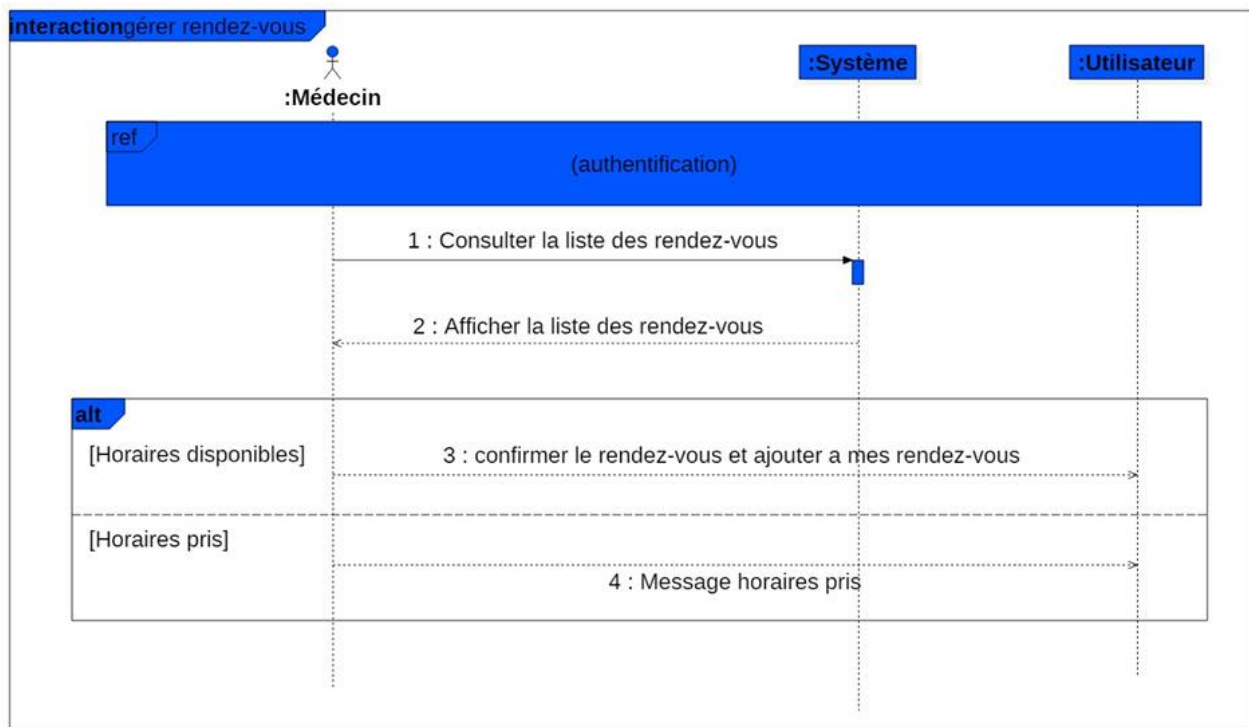


FIG. 3.12 – *Diagramme de séquence de cas d'utilisation « gérer rendez-vous »*

3.5.2 Le diagramme d'activité

- La figure suivante montre le diagramme d'activité du cas d'utilisation « s'authentifier »

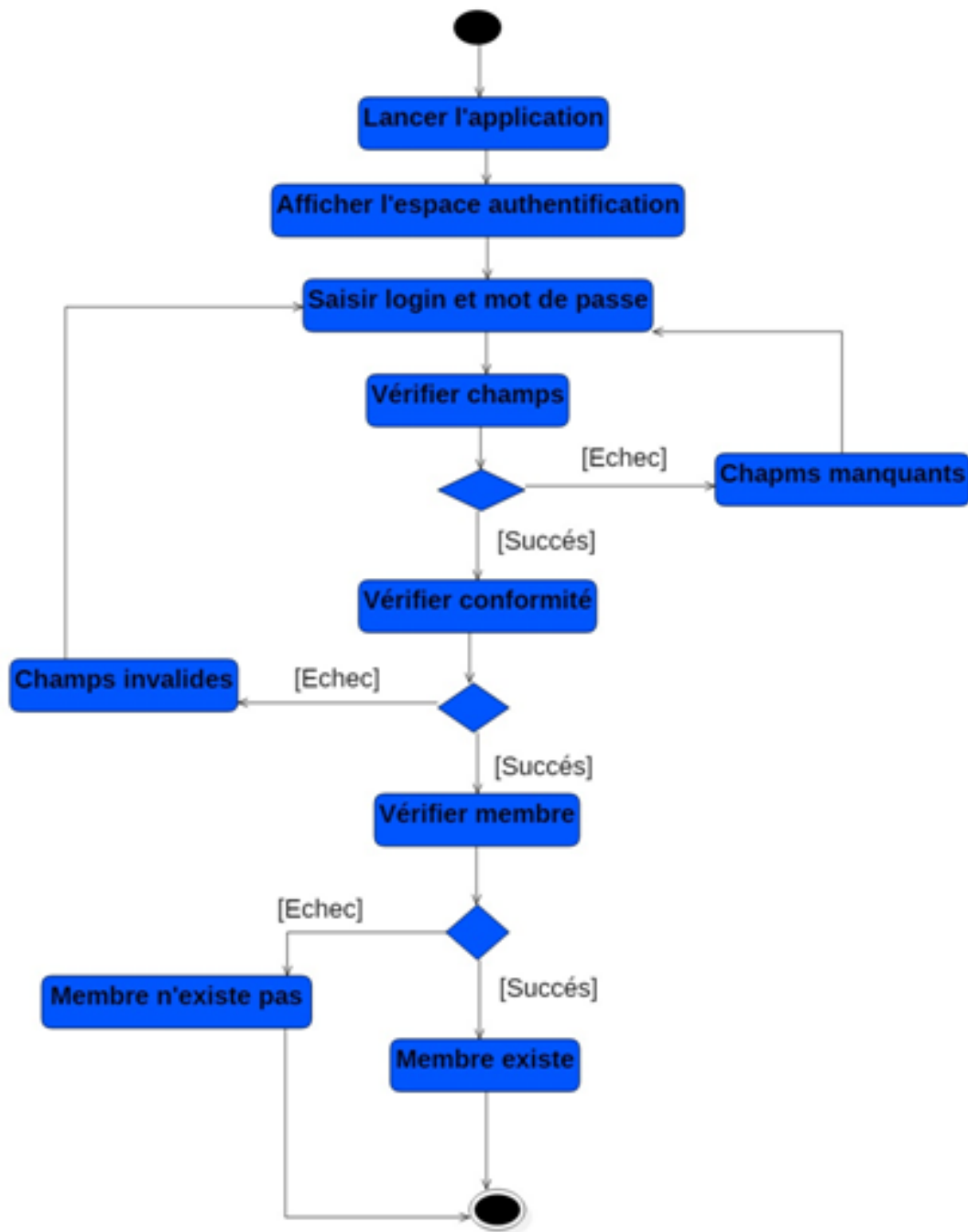


FIG. 3.13 – Le diagramme d'activité du cas d'utilisation « S'authentifier »

- La figure suivante montre le diagramme d'activité du cas d'utilisation « Géolocaliser un local »

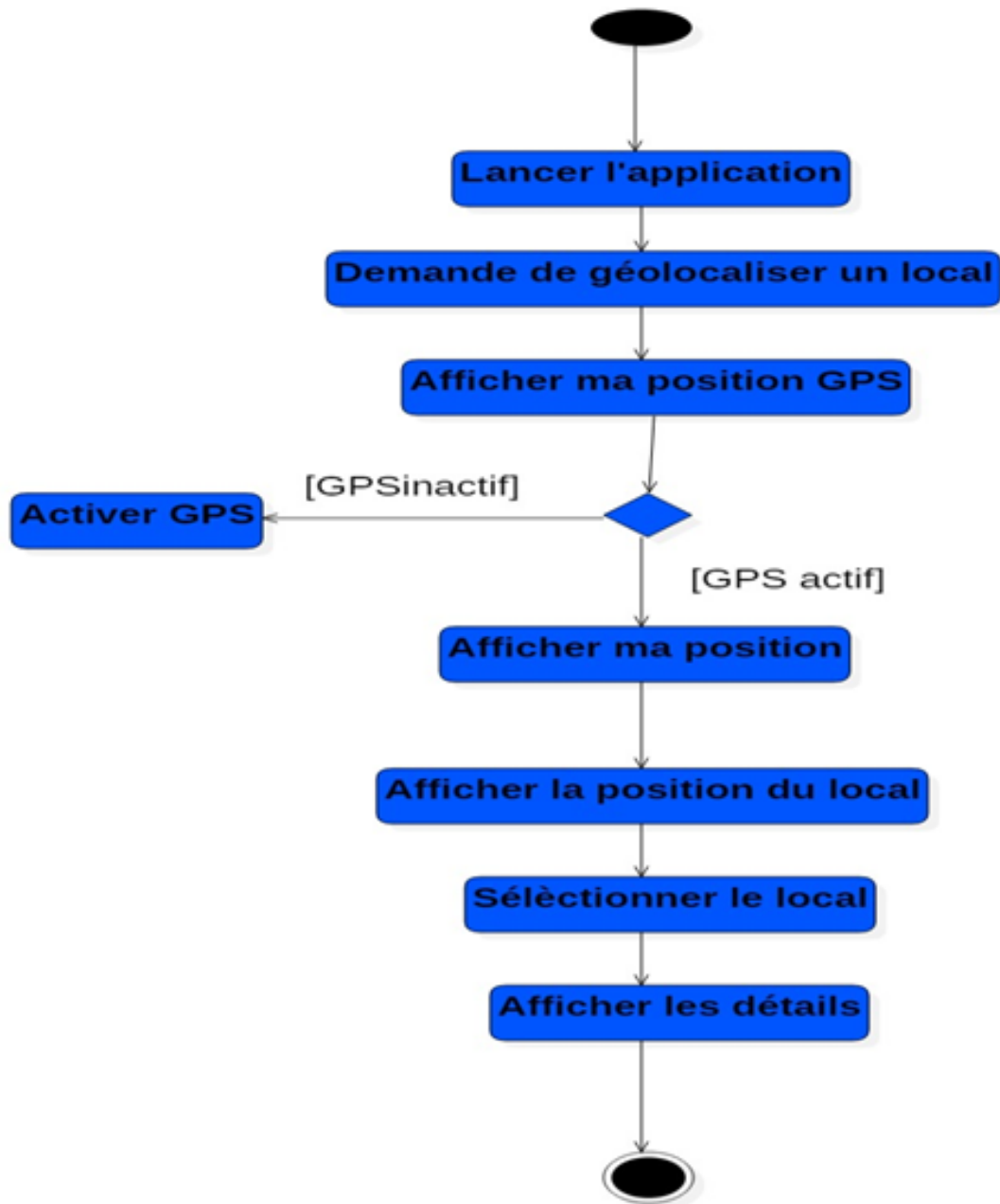


FIG. 3.14 – *Le diagramme d'activité du cas d'utilisation « Géolocaliser un local »*

3.5.3 Elaboration du diagramme de classe

- Le dictionnaire de données

Table	Attribut	Désignation	Type	Taille
Utilisateur	• id_utilisateur	• L'identifiant de l'utilisateur	Varchar	10
	• nom_utilisateur	• Le nom de l'utilisateur	Varchar	20
	• prenom_utilisateur	• Le prénom de l'utilisateur	Varchar	20
	• ntel_utilisateur	• Le numéro de téléphone de l'utilisateur	Integer	10
Local	• id_local	• L'identifiant du local	Varchar	10
	• nom	• Le nom du local	Varchar	20
	• adresse	• L'adresse du local	Varchar	50
	• ntel	• Le numéro de téléphone de local	Integer	10
	• longitude	• La longitude	Double	20
	• latitude	• La latitude	Double	20
Médecin	• Login	• Le login du médecin	Varchar	10
	• mot_de_passe	• Le mot de passe	Varchar	10
	• spécialité	• La spécialité	Varchar	20
Rendez-vous	• id_rdv	• L'identifiant du rendez-vous	Varchar	10
	• etat_rdv	• L'état du rendez-vous	Varchar	10
	• description	• La description du rendez-vous	Varchar	50
	• date	• La date	Date	10

TAB. 3.10 – *Dictionnaire de données*

- La figure suivante montre le diagramme de classe de notre solution informatique

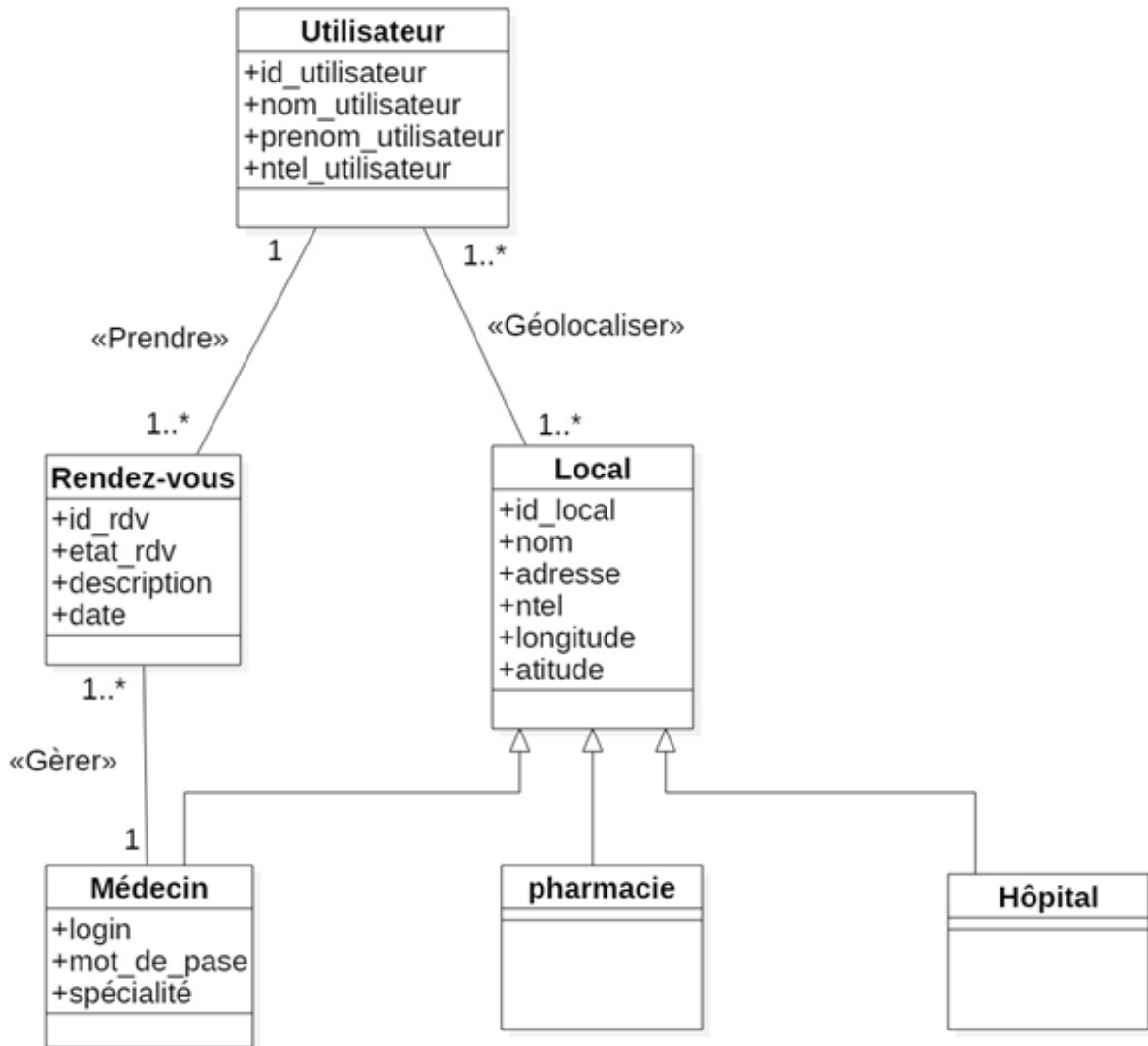


FIG. 3.15 – *Diagramme de classe de conception*

3.5.4 le diagramme de déploiement

Le diagramme de déploiement suivant montre la distribution de différents composants de notre solution informatique (composants, artefacts et base de données) sur les différents terminaux matériels (nœuds sous forme de cubes), les dépendances entre les composants (flèches discontinues) et les relations (connexions) entre les terminaux (droite continue).

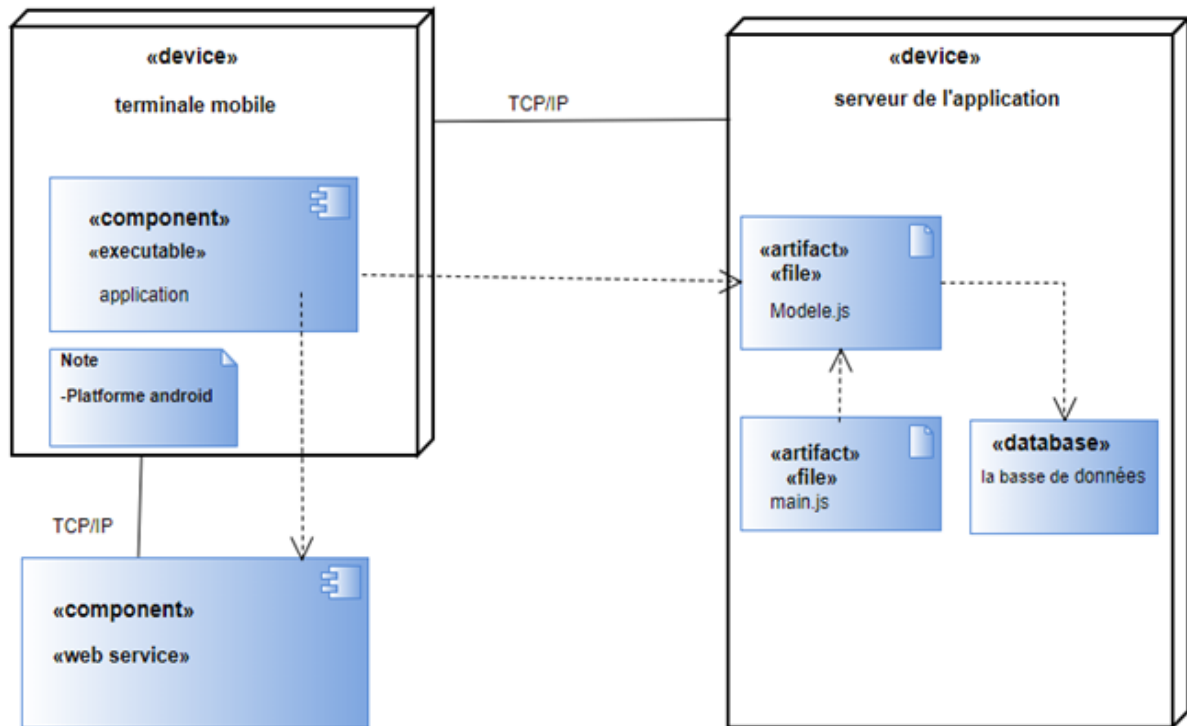


FIG. 3.16 – *Diagramme de déploiement de l'application*

3.6 Création de la base de données

La faiblesse des SGBDR en termes de performance, d'évolutivité et de flexibilité des besoins de traitement à grande échelle des données complexes a conduit à l'apparition des nouvelles tendances des SGBD à savoir les NoSQL (Not only SQL). Ceux-ci offrent en effet de très bonnes performances pour traiter de gros volumes de données faiblement structurées[20].

Le NoSQL, pour "not only SQL" désigne les bases de données qui ne sont pas fondées sur l'architecture classique des bases de données relationnelles. Développé à l'origine pour gérer du Big Data, l'utilisation des bases de données NoSQL a explosé depuis quelques années, elles ont été conçues pour résoudre les problèmes de traitements de données en volume, multi-sources et multi-formats.

Aujourd'hui le NoSQL apporte une nouvelle façon d'appréhender la modélisation des données. Le NoSQL n'est pas là pour remplacer les bases de données relationnelles, il répond à des besoins différents mais les deux approches peuvent cohabiter. Le choix de l'une ou de l'autre sera donc fortement dépendant du contexte et du besoin.

L'intérêt des systèmes de stockage NoSQL réside surtout dans les choix d'architecture logicielle qui ont été pris lors de leurs conceptions. Parmi les raisons principales qui ont mené à la création de ces systèmes, on retrouve surtout deux points principaux :

- La possibilité d'utiliser autre chose qu'un schéma fixe sous forme de tableaux dont toutes les propriétés sont fixées à l'avance .

- La possibilité d’avoir un système facilement distribué sur plusieurs serveurs et avec lequel un besoin supplémentaire en stockage ou en montée en charge se traduit simplement par l’ajout de nouveaux serveurs[21].

Le modèle NoSQL a une caractéristique principale qui consiste à ne plus avoir un schéma commun à un ensemble de données au sein d’une même structure. Chaque donnée a son propre schéma, indépendamment des autres. Par conséquent, la notion de contraintes d’intégrité sur les structures de données n’est plus ou peu assurée par ces systèmes. Contrairement à l’ère relationnelle, connu par le développement et la prospérité d’une poignée de SGBD, l’ère NoSQL, a vu une explosion de nombre de SGDB, car les besoins des entreprises ont réellement multiplié.

Nous présentons dans ce qui suit, une comparaison des deux systèmes de gestion de base de données. Cette comparaison montre que la grande différence entre ces trois familles concerne de manière générale l’évolutivité horizontale non supportée et les requêtes complexes faiblement gérées par les SGBD relationnels. Ainsi, la jointure est partiellement supportée par les NoSQL[20].

	Relationnel	NoSQL
Application	Transaction	Recherche
Schéma	Tables	Clé-valeur, document, colonne, graphe
Évolution horizontale	Non supporté	Supporté
Langage SQL	Supporté	Non supporté
Requête complexe	Faible	Fort
Jointure	Supportée	Supportée partiellement

TAB. 3.11 – /
Comparaison entre SGBD Relationnels NoSQL [20]

Notre application relève des systèmes distribués et gère des données hébergées sur des serveurs et qui changent en temps réel nous avons donc besoin de stocker nos données en temps réel et y accéder aussi rapidement, et les bases de données traditionnelles à savoir relationnelles n'arrivent pas à fournir un temps de réponse assez satisfaisant, c'est la raison pour laquelle, nous avons opté donc pour l'utilisation des bases de données NoSQL.

La mise en œuvre du bases de données NoSQL est possible grâce à la dénormalisation, dans une relation entre deux tables A et B, la dénormalisation qui consiste à dupliquer certaines données de la table B dans la table A afin d'optimiser les requêtes qui pourront se contenter d'interroger la table A sans avoir à faire de jointures entre A et B.

Nous allons présenter dans ce qui suit, les principales familles du concept NoSQL.

- **Famille clé/valeur**

Le but de la famille clé-valeur est l'efficacité et la simplicité. Un système clé-valeur agit comme une énorme table de hachage distribuée sur le réseau. Tout repose sur le couple Clé/Valeur. La clé identifie la donnée de manière unique et permet de la gérer. La valeur contient n'importe quel type de données [21].

- **Famille en colonnes**

Traditionnellement, les données sont représentées en ligne, représentant l'ensemble des attributs.

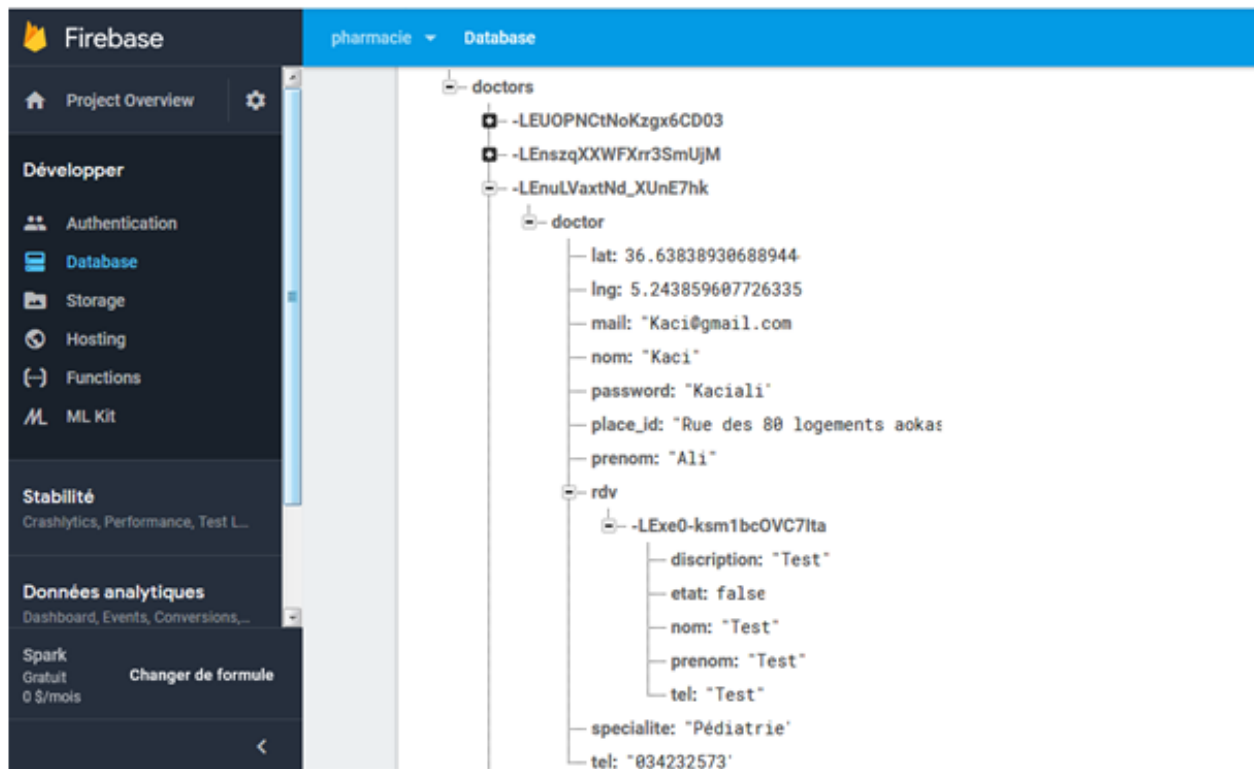
Le stockage orienté colonne change ce paradigme en se focalisant sur chaque attribut et en les distribuant. Il est alors possible de focaliser les requêtes sur une ou plusieurs colonnes, sans avoir à traiter les informations inutiles (les autres colonnes) [21].

- **Familles Orientées graphes**

Dans la base orientée graphe, les données stockées sont : les nœuds, les liens et des propriétés sur ces nœuds et ces liens. Les requêtes que l'on peut exprimer sont basées sur la gestion de chemins [21].

- **Famille Orientée document**

Les bases orientées documents ressemblent sans doute le plus à ce que l'on peut faire dans une base de données classique pour des requêtes complexes. Le but de ce stockage est de manipuler des documents contenant des informations avec une structure complexe (types, listes, imbrications). Il repose sur le principe du clé/valeur, mais avec une extension sur les champs qui composent ce document [19]. Ci-dessous se trouve un exemple de cette famille[21].

FIG. 3.17 – *Exemple du format orienté documents*

3.7 Conclusion

Tout au long de ce chapitre, nous avons détaillé la conception de notre application à travers les différents diagrammes afin que la phase réalisation et la mise en place de l'application soit plus souple et plus aisée.

Dans le chapitre suivant nous entamons la phase finale de notre mémoire qui est la phase d'implémentation où nous présentons l'environnement logiciel et matériel adapté.

Chapitre 4

Expérimentation

4.1 Introduction

Après avoir élaboré la conception de notre application, nous abordons dans ce chapitre le dernier volet de ce mémoire , qui a pour objectif de présenter la phase de réalisation.

Nous menons tout d'abord une étude technique où nous décrivons les ressources logicielles utilisées dans le développement de notre mémoire. Nous présentons en premier lieu notre choix de l'environnement de travail, où nous spécifions l'environnement matériel et logiciel que nous avons utilisé pour réaliser notre application puis nous détaillons l'architecture, aussi nous présentons quelques interfaces de notre application pour illustrer le fonctionnement de quelques activités du système.

4.2 Environnement de développement

4.2.1 Environnement matériel

Pour le développement de notre application nous avons utilisé un PC portable « TOSHIBA C55-B856 » dont la configuration est la suivante :

- Processeur Intel Core i3-3120M avec fréquence 2.5 GHz
- Quantité de mémoire vive 4 Go
- Capacité du disque dur 500 Go
- Type de système système exploitation 64bits

De plus, pour tester notre application, nous avons utilisé un Smartphone Condor Plume P8 dont la configuration est :

- Version android 5.1
- Ram 2 Go
- Processeur Octa Core 1.3 Ghz (64bit)

4.2.2 Environnement logiciels

React

C'est une bibliothèque en langage de programmation Javascript qui permet la création d'applications web. Elle est open source c'est-à-dire à la disposition de tous et enrichie par toutes les contributions des développeurs. React est déjà très connu et utilisé par de nombreux développeurs[22].

React-native

React native est un framework mobile hybride développé par facebook depuis début 2015. Il continue d'évoluer avec le soutien de nombreux contributeurs sur Github. Le but de react native est de pouvoir réutiliser le maximum de code entre les différentes plateformes (iOS et Android)[23].

React Native est une technologie très prometteuse avec un énorme potentiel, qui a déjà été dûment appréciée par des géants comme Airbnb, Baidu, Discovery, Instagram et d'autres. Constamment améliorée et mise à jour par ses fondateurs, elle est à deux pas de devenir une alternative à part entière à Objective C, Swift et Java[24].

Parmi les avantages de react native, nous citons :

- Il permet d'utiliser Javascript

La première de ses particularités est que ce framework permet de développer des applications mobiles natives sous iOS et Android en utilisant le concept et le design de React[25].

- Un gain de temps considérable

Quand un développeur construit une application web, il lui suffit juste de sauvegarder les changements et de recharger le navigateur pour voir les modifications apportées. Pour une application native c'est différent, il doit à chaque fois re-compiler même s'il y a juste un changement de texte ou de pixel.

React Native permet aux développeurs de ne plus recompiler mais de recharger l'application instantanément. Les développeurs gagnent ainsi du temps pour se concentrer sur les détails qui ont un vrai intérêt[25].

- Les performances sont optimisées

Contrairement à d'autres frameworks, React Native permet de créer des applications mobiles natives très performantes. Elles sont fluides et responsives et offrent une très bonne expérience et interface utilisateur (UX et UI). Il fait le pont entre le code Javascript et les composants natifs de l'appareil sans passer par la case navigateur web. Il n'y a donc pas de compromis de performance. Elles ont aussi une bonne réactivité au niveau de l'interface[25].

Node.js

NodeJS c'est une plateforme de développement qui met à disposition plusieurs bibliothèques JavaScript directement sur notre machine. Ce n'est pas un serveur, ni un framework, mais un applicatif permettant d'interagir par le biais du langage JavaScript et ainsi effectuer des actions en fonction des entrées / sorties détectées.

NodeJS repose entièrement sur le moteur V8 pour fonctionner et permet d'avoir des performances de très haut niveau[26].

NPM (Node Packaged Modules)

Le package manager officiel qui simplifie grandement l'installation de modules (une librairie dédié à NodeJS)[27].

Firebase

Firebase est un ensemble de services de haut niveau pour le développement d'applications web ou mobile.

Firebase offre en particulier des services très bien réalisés pour la gestion des utilisateurs, des notifications, du stockage de fichiers, des bases de données[28].

Firebase dispose de nombreux services, nous citons quelques uns:

- Une base de données en temps réel

Ceci est probablement la fonctionnalité la plus spectaculaire de Firebase. Lorsqu'une base de données est mise à jour, tous les utilisateurs connectés reçoivent les nouvelles données en temps réels. Cela signifie que l'application peut rester à jour sans l'interaction de l'utilisateur. Fini les « pull to refresh » pour mettre à jour une tableView !

Si un client change une donnée, tous les autres utilisateurs reçoivent le changement quasi instantanément. Et si un utilisateur perd la connexion, l'application va synchroniser et fusionner les nouvelles données dès que la connexion sera rétablie.

- L'authentification des utilisateurs

Firebase offre une suite complète de méthodes d'authentification (par email, Twitter, Facebook). Elle permet d'avoir un contrôle total sur l'authentification des utilisateurs sans se soucier des règles de sécurité. Avec Firebase, l'authentification est rendue simple et rapide à implémenter.

La stack fournit une suite complète de méthodes d'authentification, permettant aux développeurs d'avoir un contrôle total sur l'authentification des utilisateurs. La mise en place de règles de sécurité est rendue simple et flexible par une approche déclarative très intéressante.

- Push notifications

Un service de notifications est disponible afin d'envoyer des notifications aux utilisateurs ciblés.

- Firebase Analytics

Firebase intègre les outils d'Analytics afin d'obtenir des informations et statistiques sur les utilisateurs de notre application. L'utilisation est très simple via le SDK de Firebase. Nous pouvons par exemple enregistrer l'ensemble des pages qui sont ouvertes dans l'application afin de connaître le comportement de nos utilisateurs et les fonctionnalités les plus utilisées.

- L'hébergement

Firebase permet aussi de stocker des médias (documents textes, musiques, images) de manière sécurisée.

La base de données FirebaseRealtime

C'est une base de données hébergée dans le cloud. Les données sont stockées en JSON et synchronisées en temps réel avec chaque client connecté. Lors de la création des applications multiplates-formes avec nos SDK iOS, Android et JavaScript, tous les clients partagent une instance de base de données en temps réel et reçoivent automatiquement des mises à jour avec les dernières données.

Expo

expo est une chaîne d'outils gratuite et open source construite autour de React Native pour nous aider à construire des projets natifs iOS et Android en utilisant JavaScript et React[29].

Sublime text

Sublime Text est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités.

Depuis la version 2.0, sortie le 26 juin 2012, l'éditeur prend en charge 44 langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares[30].

4.3 Langages de développement

- **Javascript**

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel. JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances npm, avec environs 500 000 paquets en août 2017[31].

- **JSON**

(JavaScript Object Notation - Notation Objet issue de JavaScript) est un format léger d'échange de données. Il est facile à lire ou à écrire pour des humains. Il est aisément analysable ou général par des machines. Il est basé sur un sous-ensemble du langage de programmation JavaScript .

JSON est un format texte complètement indépendant de tout langage, mais les conventions qu'il utilise seront familières à tout programmeur habitué aux langages descendant du C, comme par exemple : C lui-même, C++, C#, Java, JavaScript, Perl, Python et bien d'autres. Ces propriétés font de JSON un langage d'échange de données idéal [32].

4.4 Les services web utilisés

Un Service Web est une fonctionnalité qu'on peut appeler à distance à travers un réseau (comme un intranet d'entreprise ou internet lui-même). Cette fonctionnalité a été réalisée par un tiers langage et qui peut être invoquée à travers des messages XML ou JSON [33].

Nous distinguons deux générations de Services Web :

- Service web 1.0 basés sur le protocole SOAP.
- Service web 2.0 basés sur le style architectural REST.

SOAP est un protocole tandis que REST est un style d'architecture.

La différence majeure entre ces 2 éléments est le degré de liaison entre le client et le serveur. Un client développé avec le protocole SOAP ressemble à un logiciel d'ordinateur, car il est étroitement lié au serveur. Si une modification est effectuée d'un côté ou de l'autre, l'ensemble peut ne plus fonctionner. Il faut effectuer des mises à jour du client s'il y a des changements sur le serveur et vice-versa[33].

Un client de type REST sait utiliser un protocole et des méthodes standardisées. Son application doit rentrer dans ce modèle. On ne crée pas de méthodes supplémentaires, on utilise les méthodes standardisées que l'on développe pour le type de media dont on a besoin. Il y a en conséquence beaucoup moins de couplage entre le client et le serveur : un client peut utiliser un service de type REST sans aucune connaissance de l'API. A l'inverse, un client SOAP doit tout savoir des éléments qu'il va utiliser pendant son interaction avec le serveur, sinon cela ne fonctionnera pas[33].

Par ailleurs, REST possède des caractéristiques spécifiques qui le différencient de SOAP. REST est indépendant d'un protocole. On peut utiliser aussi bien le protocole HTTP que FTP, tant qu'il s'agit d'un protocole possédant un schéma standard pour une URI [33].

Vu la simplicité du style REST, nous avons opté pour l'invocation des services web REST existants avec des messages JSON.

4.4.1 Google Places API Web Service

Permet de rechercher des informations de lieux dans toute une série de catégories, telles que des établissements, des points d'intérêt importants, des points géographiques, et rechercher des lieux par proximité ou via une chaîne de texte. Une requête Place Search (recherche de lieux) renvoie une liste de lieux avec des informations récapitulatives sur chaque lieu ; des informations supplémentaires peuvent être obtenues à l'aide d'une requête de détails de lieux Place Details[34].

4.4.2 Google Places Details API

Une fois un paramètre place id ou référence obtenu à partir d'une recherche de lieux (Place Search) , nous pouvons demander des détails supplémentaires sur un établissement ou un point d'intérêt spécifique en lançant une requête de détails de lieu (Place Détails).

Une requête Place Détails renvoie des informations supplémentaires sur le lieu indiqué, telles que l'adresse complète, le numéro de téléphone, la note et les avis des utilisateurs[35].

4.4.3 Google Maps Directions API

C'est un service qui calcule des itinéraires entre des points géographiques au moyen d'une requête http[36].

4.5 Description des interfaces réalisées

4.5.1 Interface Bienvenue

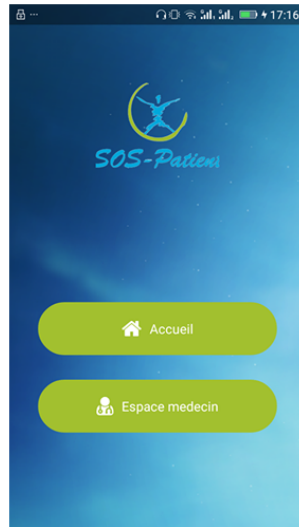


FIG. 4.1 – *Interface Bienvenue*

L'interface bienvenue est la première interface de notre application, dont il y a deux boutons Accueil qui permet d'accéder à l'interface accueil. Espace médecin qui permet d'accéder à l'interface authentification médecin.

4.5.2 Interface accueil

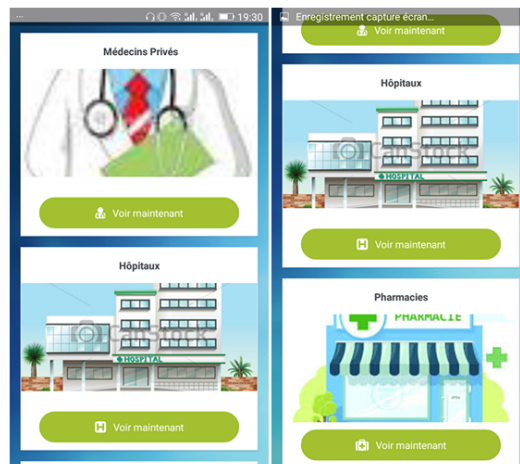


FIG. 4.2 – *Interface d'accueil*

Cette interface nous permet de choisir le corps médical que nous cherchons.

4.5.3 Interface hôpital

Cette interface nous permet de voir la liste des hôpitaux proximité selon la position de l'utilisateur .



FIG. 4.3 – *Interface liste des hôpitaux*

4.5.4 Interface détail-hôpital

Cette interface nous permet de voir les détails d'un hôpital choisis par l'utilisateur (nom, adresse et numéro de téléphone).

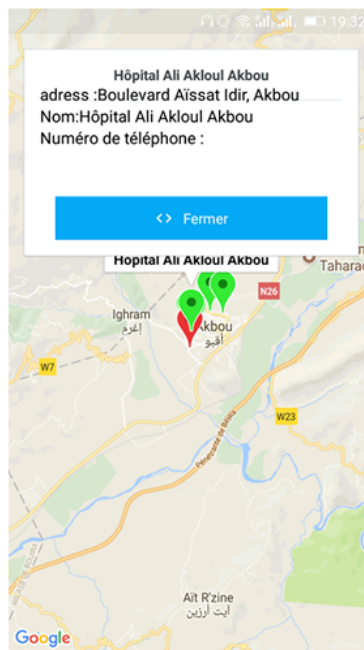


FIG. 4.4 – *Détails hôpitaux*

4.5.5 Interface pharmacie

Cette interface nous permet de voir la liste des pharmacies à proximité selon la position de l'utilisateur .



FIG. 4.5 – *liste pharmacies*

4.5.6 Interface détail-pharmacie

Cette interface nous permet de voir les détails d'une pharmacie choisie par l'utilisateur (nom, adresse et numéro de téléphone).

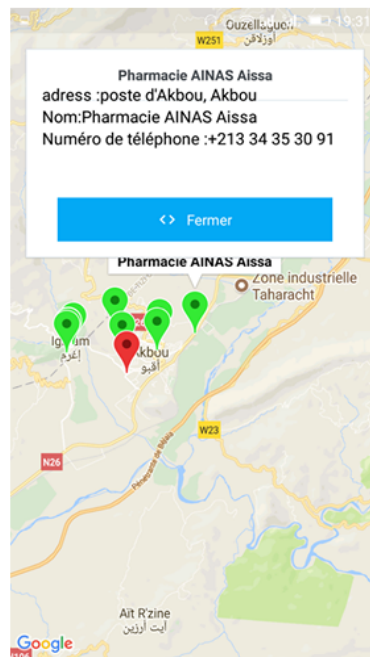


FIG. 4.6 – *détails pharmacie*

4.5.7 Interface médecins



FIG. 4.7 – *interface médecins*

4.5.8 Interface détails médecin

Cette interface affiche les détails d'un médecin après avoir choisit un médecin et cliqué sur le marqueur .

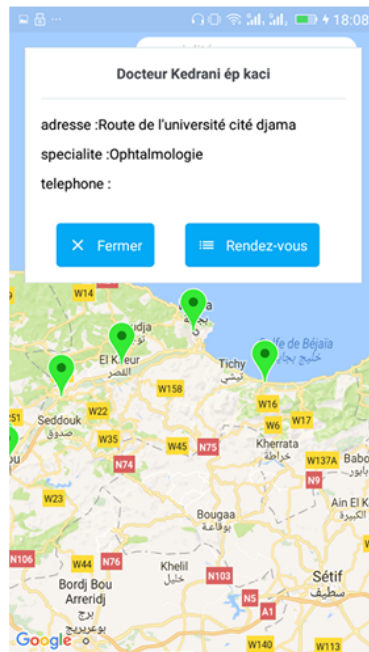


FIG. 4.8 – *interface médecins*

4.5.9 Interface prendre rendez-vous

Cette interface permet à l'utilisateur de prendre un rendez-vous chez un médecin choisi.

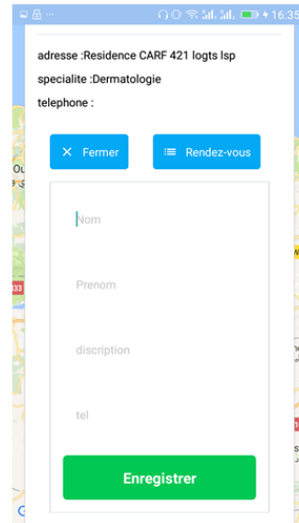


FIG. 4.9 – *Prendre un rendez-vous*

4.5.10 Interface d'authentification médecin

L'interface d'authentification est une des interfaces les plus importantes dans notre application, car chaque médecin doit être enregistré dans notre système pour qu'il puisse consulter ses rendez-vous.

A travers cette interface le médecin donne son email et son mot de passe. Si cette combinaison correspond aux informations qui existent dans la base de données, l'application le redirige vers son profile sinon un message d'erreur apparaît.

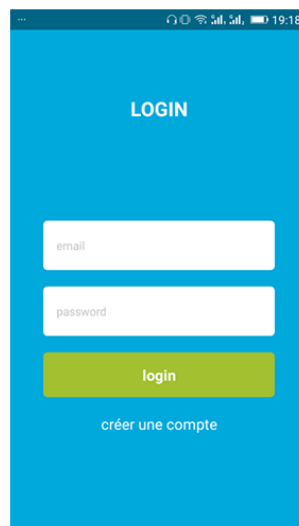


FIG. 4.10 – *Interface d'authentification médecin*

4.5.11 Interface profile médecin

Cette interface affiche l'espace de médecin après avoir saisi son email et mot de passe ou il peut consulter les rendez-vous.

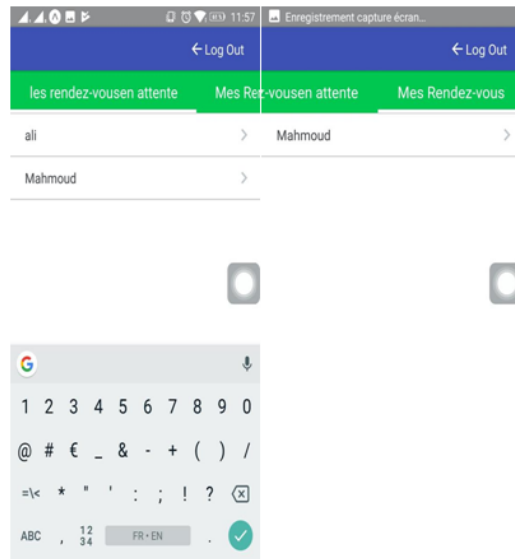


FIG. 4.11 – *profile-médecin*

4.5.12 Interface créer un compte

Cette interface permet au médecin de créer un compte dans notre application. Le médecin saisi le nom, prénom, email, téléphone, mot de passe, spécialité et il fait une recherche de son adresse sur le map . Puis il les enregistre dans la base de données.

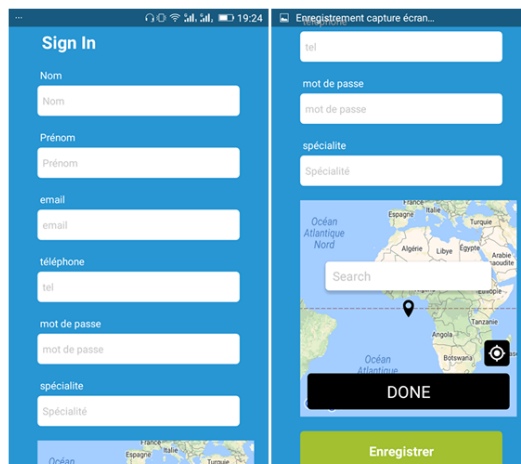


FIG. 4.12 – *inscription médecin*

- Test

L'intérêt de cette partie est la présentation des jeux de tests effectués, pour les cas critiques du système, afin de s'assurer de leur bon fonctionnement.

- Test authentification

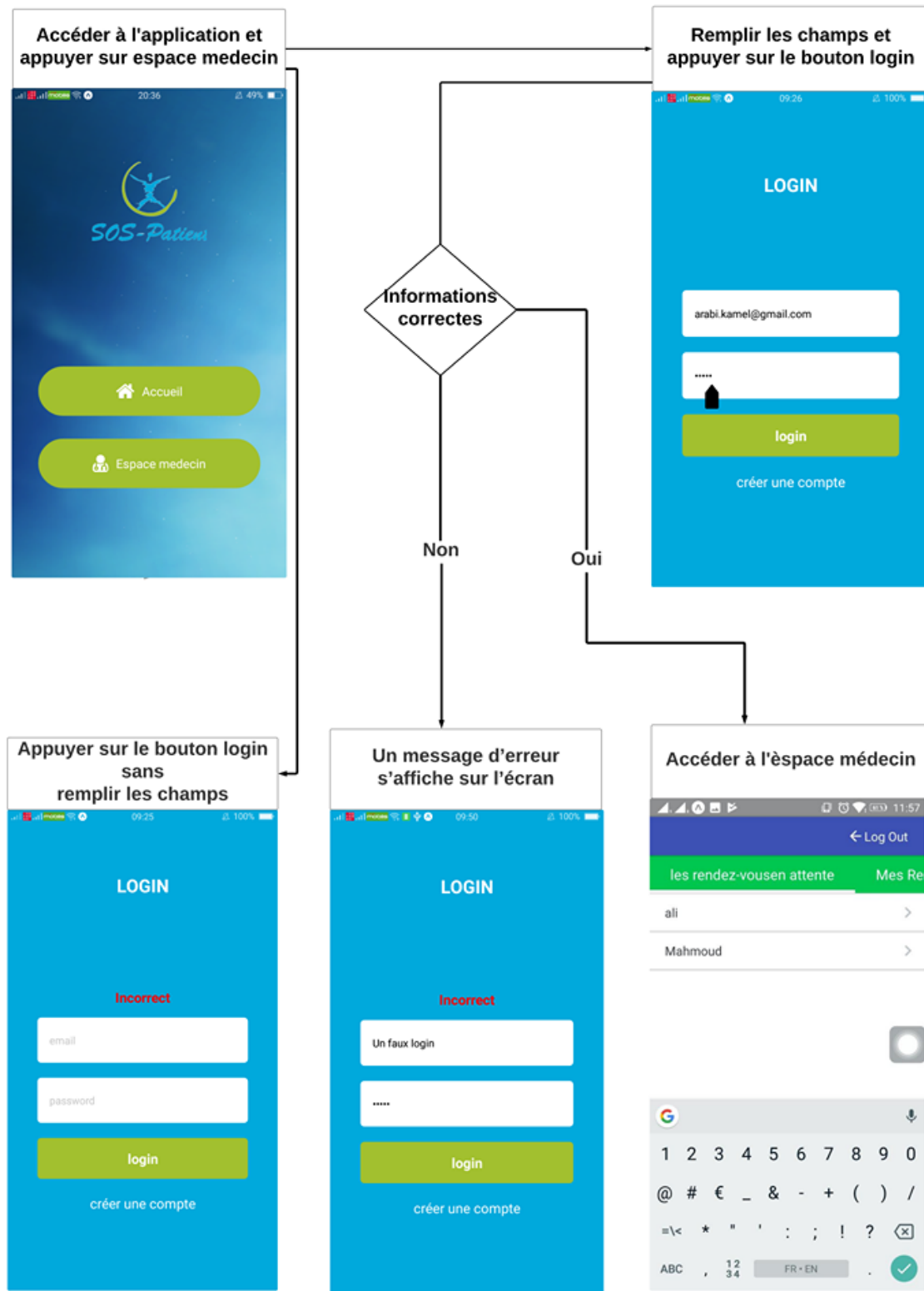


FIG. 4.13 – *Test authentification*

4.6 Conclusion

La phase de réalisation est une étape importante dans le cycle de vie d'une application. Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement, les outils et les langages de développement associés à notre système. En effet, nous avons achevé l'implémentation tout en respectant la conception élaborée.



FIG. 4.14 – *Interface d'accueil*

Chapitre 5

Conclusion générale

5.1 Introduction

Avec l'avancée et l'émergence des technologies mobiles, les développements embarqués sont de plus en plus demandés sur le marché. De nos jours, avoir un dispositif mobile est devenu incontournable pour tout individu, ce qui nous a poussé et motivé à réfléchir sur une application mobile utile pour la communauté.

Notre application "SOS-Patient" est dédiée aux corps médicaux, elle permet de Géolocaliser les pharmacies, les hôpitaux et les médecins privés ainsi qu'une prise de rendez-vous chez un médecin au besoin en utilisant des Smartphones. Cette application permet entre autres aux médecins de gérer leurs rendez-vous.

Notre étude a été menée au niveau d'UMALIS TUNISIA, et le travail accompli au sein de cette entreprise, nous a permis de s'acquitter des notions théoriques acquises durant notre cursus universitaire et de découvrir le milieu professionnel.

5.2 Travail réalisé

L'étude que nous avons réalisée se focalise particulièrement sur l'application de la technologie des services web dans un contexte informatique mobile. Son but est destiné à l'exécution des services web n'importe où et à tout moment via des messages.

Nous avons présenté une conception et implémentation d'une application mobile sous Android à base de services web. Le système que nous avons mis en œuvre est intitulé "SOS-Patient", il permet à un client d'éviter de faire plusieurs recherches sur le web pour satisfaire son besoin.

5.3 Apport du mémoire

Ce travail de conception et de développement nous a été bénéfique sur plusieurs plans et il nous a permis d'améliorer nos connaissances théoriques et pratiques acquises durant notre cursus de master.

Du point de vue technique, ce mémoire nous a permis de s'adapter à l'environnement du développement informatique, de même il nous a permis de maîtriser les méthodes de développement ; le Processus unifié et des technologies de programmation. Il nous a également permis de mettre

en évidence les concepts de services web en particulier la communication et l'échange de données en suivant le principe des services REST.

Au début de notre travail, nous avons pris du recul pour l'étude conceptuelle afin de proposer un cadre méthodologique adéquat et performant.

5.4 Problèmes rencontrés

Les services Web mobiles s'avèrent d'un intérêt croissant. Ils se traduisent par l'extension des services web électroniques aux environnements mobiles.

Cependant, les limites et les caractéristiques d'un environnement mobile ainsi que, les appareils mobiles ont des ressources limitées. Ces derniers relèvent des difficultés en la découverte et l'exécution des services web au sein des réseaux.

Dans une optique d'utiliser des technologies de pointe pour aboutir à une application performante et très efficace, nous avons trouvé des difficultés pour s'acquitter des outils de développement et leur documentation qui est insuffisante, surtout qu'il s'agit de rassembler plusieurs technologies au sein d'une même application. Seulement, cette expérience nous a permis de découvrir beaucoup d'outils et d'environnement de développement.

Aussi, la distance qui nous sépare de l'organisme d'accueil et qui est relativement consistante, représentait une difficulté pour mieux gérer le temps et les délais.

5.5 Perspectives et travaux futurs

Enfin, nous avons atteints les objectifs fixés au début, néanmoins des perspectives pour améliorer notre système, restent toujours envisageables pour l'enrichir avec des fonctionnalités avancées telles que :

Partie technique :

- Géolocalisation d'autres corps médicaux tels que les cliniques privées.
- La recherche de plus court chemin à une pharmacie s'il en existe plusieurs pour l'atteindre.
- La création d'un espace patient afin de gérer ses rendez-vous.
- La Vérification du médecin : sachant que notre application renferme un volet réservé aux médecins, où le médecin peut s'authentifier pour créer son espace et gérer ses rendez-vous. Seulement, tout acteur peut jouer le rôle d'un médecin, donc un outil de vérification de l'identité du médecin est fortement recommandé.

Dans cette perspective, nous pouvons améliorer notre application en ajoutant une image de la carte d'identité du médecin ou sa carte professionnelle lors de la création de l'espace. Les informations capturées seront vérifiées et validées par l'administrateur.

- La planification d'un calendrier préalable pour le médecin.
- Le Suivi des patients à distance.

Références

- [1] <http://www.datafirst.fr/SiteCollectionDocuments/Techno/DATAFIRST-BULLETIN-DE-VEILLE-TECHNOFLASH2-MOBILITE.pdf>, dernière consultation le 18/06/2018.
- [2] <https://www.definitions-marketing.com/definition/applicationmobile/>, dernière consultation le 19/06/2018.
- [3] REMACI Zineb Yasmina et GHITRI Salim "Développement d'une application mobile Le jeu " smile " " mémoire licence année 2013-2014 université Tlemcen .
- [4] Telli Fatiha et Benjeddou Khadija "Une application mobile : Une télécommande Bluetooth d'une souris optique" Année Universitaire : 2012 /2013 université ouagla.
- [5] <https://www.taktilcommunication.com/blog/applications-mobile/sitemobile-vsapplicationmobileavantagesetinconvenients.html>, dernière consultation le 18/06/2018.
- [6] <https://cours-informatique-gratuit.fr/dictionnaire/android/>, dernière consultation le 18/06/2018.
- [7] <http://www-igm.univ-mlv.fr/dr/XPOSE2008/android/index.html#OHA>, dernière consultation le 18/06/2018.
- [8] <http://www-igm.univ-mlv.fr/%7Edr/XPOSE2008/android/index.html>, dernière consultation le 18/06/2018.
- [9] <http://www.gizlogic.fr/android-m-nouvelle-version-os-google/>, dernière consultation le 18/06/2018.
- [10] <https://mathias-seguy.developpez.com/tutoriels/android/comprendre-cyclevie-activite/>, dernière consultation le 18/06/2018.
- [11] www.memoireenligne.com vente flash université de Bejaïa, dernière consultation le 18/06/2018.
- [12] REMACI Zineb Yasmina et GHITRI Salim "Développement d'une application mobile Le jeu " smile " " 2013-2014 université Tlemcen.
- [13] <http://telecom.insa-lyon.fr/sites/default/files/cgt/promo-20092012%20-%20geolocalisation.pdf> , dernière consultation le 18/06/2018.
- [14] <https://sites.google.com/site/wikismartphone/techno/techniques-de-geolocalisation>, dernière consultation le 18/06/2018.

- [15] <https://www.companeo.com/geolocalisation-de-vehicules/guide/geolocalisation-gps#0>, dernière consultation le 18/06/2018.
- [16] *Rawdha MABROUKI " Conception et Développement d'un Module Dynamics Ax de Gestion de Visa" Année universitaire 2015/2016, école supérieure privée des sciences appliquées et de management, Tunisie.*
- [17] https://www.memoireonline.com/06/08/1172/m_application-de-mailing-de-masse7.html , dernière consultation le 18/06/2018.
- [18] *Pascal roques, Groupe Eyrolles 2002 ISBN :2-212-11070-7.*
- [19] https://www.memoireonline.com/06/12/5947/m_Conception-dun-outil-dadministration-reseaux3.html, dernière consultation le 19/06/2018.
- [20] *Lahcène BRAHIMI "Données de tests non fonctionnels de l'ombre à la lumière : une approche multidimensionnelle pour déployer une base de données " THESE pour l'obtention du Grade de DOCTEUR DE L'ÉCOLE NATIONALE SUPÉRIEURE DE MÉCANIQUE ET D'AÉROTECHNIQUE Ecole Doctorale : Sciences et Ingénierie pour l'Information, Mathématiques Soutenue le 03 juillet 2017*
- [21] *RUDI BRUCHEZ, " Les base de données NoSQL et le Big Data comprendre et mettre enœuvre", EYROLLES, 2e édition, 2015.*
- [22] <https://www.ideematic.com/actualites/2017/06/le-developpement-mobile-avec-react-native/>, dernière consultation le 18/06/2018.
- [23] <https://makina-corpus.com/blog/metier/2016/decouverte-de-react-native>, dernière consultation le 18/06/2018.
- [24] <http://www.arca-computing.fr/react-native-une-bonne-alternative-au-developpement-natif/>, dernière consultation le 18/06/2018.
- [25] <https://usts.fr/react-native-avantages-de-framework-concevoir-application-mobile/>, dernière consultation le 18/06/2018.
- [26] <https://www.supinfo.com/articles/single/600-qu-est-que-nodejs>, dernière consultation le 18/06/2018.
- [27] <https://nodejs.developpez.com/tutoriels/javascript/nodejs-debuter-1/>, dernière consultation le 18/06/2018.
- [28] <https://www.axopen.com/firebase/> , dernière consultation le 18/06/2018.
- [29] <https://expo.io/> , dernière consultation le 19/06/2018.

- [30] <https://creationhouseblog.wordpress.com/2017/02/16/sublime-texte-le-html-pour-les-nul/> ,dernière consultation le 19/06/2018.
- [31] <http://dictionnaire.sensagent.leparisien.fr/JavaScript/fr-fr/> ,dernière consultation le 19/06/2018.
- [32] <http://www.json.org/jsonfr.html> ,dernière consultation le 19/06/2018.
- [33] <https://www.journaldunet.fr/web-tech/developpement/1202749-soap-vs-rest-les-principales-differences/> ,dernière consultation le 19/06/2018.
- [34] <https://developers.google.com/places/web-service/search?hl=fr>,dernière consultation le 19/06/2018.
- [35] <https://developers.google.com/places/web-service/details?hl=fr>,dernière consultation le 19/06/2018.
- [36] <https://developers.google.com/maps/documentation/directions/intro?hl=fr>,dernière consultation le 19/06/2018.

Résumé

Aujourd'hui, l'informatique a atteint une prodigieuse évolution technologique dans différents domaines (réseaux informatiques, bases de données, le Web, etc.). Cette évolution est nécessaire pour remédier aux problèmes rencontrés dans la vie actuelle.

L'évolution de la technologie des télécommunications a donné naissance à plusieurs services, notamment le service de la géolocalisation. Ce dernier offre diverses possibilités dans la localisation et le positionnement sur carte des points dans différents domaines.

Notre travail consiste à développer une application fiable et aisée, munie de fonctionnalités nécessaires et adéquates aux besoins des patients. Le système a été modélisé selon le processus unifié en utilisant « UML » et développé sous un Framework « REACT NATIVE » les différents services Google avec une base de données « FIREBASE ».

Mots clés : Application Mobile,Géocalisation,REACT NATIVE and FIREBASE .

Abstract

Today, computing has reached a prodigious technological evolution in different fields (computer networks, databases, the Web, etc.). This evolution is necessary to remedy the problems encountered in today's life.

The evolution of telecommunications technology has given rise to several services, including the geolocation service. The latter offers various possibilities locating and positioning, on a map, points in different areas.

Our project is to develop a reliable and easy-to-use application with the necessary and adequate functionalities that meet the needs of patients.

The system was modeled according to the unified process using «UML» and developed under a "REACT NATIVE" Framework the different google services with a «FIREBASE» database.

Mots clés : Géolocation,MobileApplication,REACT NATIVE and FIREBASE.