

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Bejaia
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de fin de cycle
En vue de l'obtention du diplôme
de Master professionnel en informatique
Spécialité : Génie Logiciel

Thème

Conception et réalisation d'un environnement M-santé pour le quotidien des patients atteints de maladies chroniques

Réalisé par :

Iskandar HOCINI
Meriem IFOURAH

Encadré par :

M^{me} N. YESSAD

Devant le jury composé de :

Examinatrice 1 : M^{me} S. LAHLAH MCB Université de Béjaïa
Examinatrice 2 : M^{me} D. KESSIRA MAA Université de Béjaïa
Encadrante : M^{me} N. YESSAD MCB Université de Béjaïa

Année universitaire : 2021-2022

Remerciements

*Nos remerciements vont à notre encadrante M **NAWAL YESSAD** pour la préciosité de ses conseils, son infinie disponibilité et son orientation qui ont constitué un apport considérable grâce auquel ce travail a pu être mené à bon port.*

Nous tenons à remercier chacun des membres du jury pour l'intérêt qu'ils ont porté à notre projet en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

*Nos remerciements s'étendent à tous nos enseignants et les membres du département d'Informatique de l'université **ABDERRAHMANE MIRA**.*

Ainsi qu'à tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement travail.

Dédicaces

A nos parents, pour leurs sacrifices déployés à notre égard, pour leur patience, leur amour et leur confiance. Qu'ils trouvent dans ce modeste travail, le témoignage de notre profonde affection et de notre attachement indéfectible ; nulle dédicace ne puisse exprimer ce qu'on leur doit.

A nos frères et sœurs et tous nos amis pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.

Table des matières

Table des figures	i
Liste des tableaux	ii
LISTE DES ABREVIATIONS	iii
Introduction Générale	1
1 La Santé Mobile	3
1.1 Introduction	3
1.2 Applications mobiles	3
1.2.1 Historique	3
1.2.2 Définition	3
1.2.3 Objectifs d'une application mobile	4
1.3 Applications santé mobile	4
1.3.1 E-santé	4
1.3.2 M-santé	5
1.4 Systèmes d'exploitation mobiles	6
1.4.1 Définition	6
1.4.2 Type de système d'exploitation mobiles	6
1.5 Etude de la plateforme Android	7
1.5.1 Présentation d'Android	7
1.5.2 Fonctionnalités d'Android	7
1.5.3 Architecture de la plateforme Android	7
1.5.4 Cycle de vie d'une application Android	9
1.6 Conclusion	10
2 Cahier Des Charges	11
2.1 Objectif du chapitre	11
2.2 Étude de l'art	11
2.2.1 Modèles existant	11
2.2.2 APPORT DU PROJET	11
2.3 Perspectives du projet	12
2.4 Design du projet	12
2.5 Membres de l'équipe et les tâches attribués dans ce projet	12
2.6 Présentation du projet	12
2.7 Spécifications non-fonctionnelles	13
2.8 Spécifications fonctionnelles	14
2.8.1 Identification des acteurs	14
2.8.2 Fonctionnalités :	14
2.9 Contraintes Générales	16
2.9.1 Qualité	16
2.9.2 Délais	16
2.9.3 Budget	17
2.10 Contraintes Technologiques	17
2.10.1 Méthode de travail	17
2.10.2 Méthodes et outils de modélisation	17
2.10.3 Environnement et langages de développement	17
2.11 Conclusion	18

3	Conception	19
3.1	Introduction	19
3.2	Méthodologie de conception	19
3.2.1	Définition du processus unifié (UP)	19
3.2.2	Diagramme de contexte	21
3.2.3	Identification des cas d'utilisations.	22
3.2.4	Description textuelle des cas d'utilisation	30
3.2.5	Diagramme de séquence	36
3.2.6	Diagramme de classe	43
3.2.7	Modèle relationnel	45
3.2.8	Dictionnaire de données	47
3.3	Conclusion	48
4	Réalisation de l'environnement	49
4.1	Introduction	49
4.2	Outils de développement	49
4.3	Langage de développement	51
4.4	Présentation des interfaces de notre environnement	52
4.5	Conclusion	61
	Conclusion Générale	62
	Bibliographie	63

Table des figures

1.1	Architecture d'android	9
1.2	Cycle de vie d'une application Android	10
2.1	Triangle de gestion de projet	16
3.1	Diagramme de contexte statique du système à réaliser.	21
3.2	Formalisme de représentation d'un diagramme de cas d'utilisation.	22
3.3	Relation d'inclusion.	22
3.4	Relation d'extension.	23
3.5	Diagramme de cas d'utilisation associé à un visiteur.	24
3.6	Diagramme de cas d'utilisation associé à un Patient.	25
3.7	Diagramme de cas d'utilisation " Suivre le diabète ".	26
3.8	Diagramme de cas d'utilisation associé à un médecin.	27
3.9	Diagramme de cas d'utilisation associé à un administrateur.	28
3.10	Diagramme de cas d'utilisation générale de notre système.	29
3.11	Diagramme de séquence " s'authentifier ".	38
3.12	Diagramme de séquence " suivre un régime alimentaire ".	39
3.13	Diagramme de séquence " se rappeler des médicaments ".	40
3.14	Diagramme de séquence " prendre un rendez-vous ".	41
3.15	Diagramme de séquence " ajouter des medecins ".	42
3.16	Diagramme de séquence "gérer la liste des hôpitaux".	43
3.17	Diagramme de classe.	44
4.1	Interface AVD.	50
4.2	L'interface d'accueil	52
4.3	L'interface d'affichage des médecins	53
4.4	L'interface d'inscription et de connexion	54
4.5	Espace médecin	55
4.6	Menu des maladies.	56
4.7	Programme sportif pour le cholestérol.	57
4.8	Interface de connexion.	58
4.9	Interface d'acceuil.	59
4.10	Espace hôpitaux.	60
4.11	Interface de gestion des hôpitaux.	61

Liste des tableaux

2.1	Membres d'équipe et les tâches attribuées.	13
2.2	Identification des acteurs.	14
2.3	Le temps accordé pour l'accomplissement du projet.	17
3.1	Cycles de vie UP.	20
3.2	Description textuelle du cas d'utilisation s'authentifier.	30
3.3	Description textuelle du cas d'utilisation " consulter le programme régime ".	30
3.4	Description textuelle du cas d'utilisation " remplir le carnet d'auto surveillance glycémique ".	31
3.5	Description textuelle du cas d'utilisation " se rappeler des médicaments ".	31
3.6	Description textuelle du cas d'utilisation " prendre des conseils ".	32
3.7	Description textuelle du cas d'utilisation " calculer la masse corporelle ".	32
3.8	Description textuelle du cas d'utilisation " consulter tableau des catégories de poids ".	33
3.9	Description textuelle du cas d'utilisation " conseil et horaire de mesure de la tension ".	33
3.10	Description textuelle du cas d'utilisation " prendre un rendez-vous ".	34
3.11	Description textuelle du cas d'utilisation " contacter son médecin et contacter son malade ".	34
3.12	Description textuelle du cas d'utilisation " consulter la liste des médecins ".	34
3.13	Description textuelle du cas d'utilisation " ajouter des médecins ".	35
3.14	Description textuelle du cas d'utilisation " gérer les rendez-vous ".	35
3.15	Description textuelle du cas d'utilisation " gérer la liste des hôpitaux ".	35
3.16	Dictionnaire de données.	47

LISTE DES ABREVIATIONS

- **E-santé** : santé électronique.
- **MOS** : Mobile Operating System.
- **DVM** : Dalvik virtuelle machine .
- **JVM** : Java Virtual Machine.
- **ADT** : Android Développement Tools.
- **API** : Application Programing Interface.
- **IOS** : IPhone Operating System.
- **UP** : Unified Process.
- **CU** : Cas d'utilisation.
- **PDA** : Personal Digital Assistant.
- **ASG** : Auto Surveillance Glycimie.
- **IG** : Index Glycimie.
- **HTML** : HyperText Markup Language.
- **XML** : Extensible Markup Language.
- **CSS** : Cascading Style Sheets.
- **SDK** : Software Developement Kit.
- **AVD** : Android Virtuel device.

Introduction Générale

La digitalisation et l'innovation sont au cœur de nos activités quotidiennes. Le secteur de la santé est autant concerné par cette transformation digitale. Dans un environnement dynamique et concurrentiel, les innovations en santé croissent de jour en jour. Les technologies numériques bouleversent les pratiques médicales actuelles. Sous ses différents domaines santé mobile, télémédecine, systèmes d'information en santé, la e-santé est aujourd'hui en plein essor. Il y a convergence de deux phénomènes. D'une part, les professionnels de santé s'emparent des outils et des services numériques. D'autre part, les citoyens-patients cherchent à prendre la main sur leur état de santé comme sur d'autres domaines.

Dans ce contexte, nous sommes appelés à concevoir, développer et mettre en œuvre, un environnement m-santé pour le quotidien des patients atteints de maladies chroniques. Notre environnement est représenté par deux applications. La première est une application mobile permettant d'augmenter le rendement du personnel médical, guider un patient dans 58 wilayas en lui assurant la localisation des différents hôpitaux, ambulances, donneurs de sang ainsi que les cabinets médicaux, faciliter les tâches aux patients ainsi pour les professionnels dans ce domaine en leur résumant l'état du patient pendant une période donnée. Pour ce faire, notre choix s'est porté sur la plateforme mobile Android qui est open source, gratuite et qui englobe une communauté importante par rapport à d'autres plateformes.

La seconde est une application web utilisée par un administrateur afin d'assurer la bonne gestion et la sûreté des informations publiées et présentées sur l'autre côté de l'environnement.

Le présent mémoire est divisé en quatre différents chapitres permettant ainsi la réalisation de cet environnement m-santé. Dans le premier chapitre, nous allons présenter les applications mobiles et les stratégies préconisées pour leur développement. Nous exposerons par la suite les applications mobiles de santé puis nous évoquerons le système d'exploitation mobile « Android », ses caractéristiques et son architecture. Et enfin, nous terminerons avec une conclusion.

Le deuxième chapitre est consacré au cahier des charges qui englobe les différents besoins du projet ainsi que sa perspective et sa présentation. Nous analyserons par la suite les fonctionnalités de l'environnement m-santé à réaliser et les acteurs qui joueront un rôle dans l'outil de gestion à mettre en œuvre.

Le troisième chapitre tournera autour du cadre du projet, de sa méthodologie, son analyse et sa conception, où nous allons décrire notre projet et le formalisme UML. Nous allons aussi recenser les différents cas d'utilisation et établir des diagrammes de séquence pour les cas d'utilisation les plus pertinents. Enfin nous étendrons la représentation de ces diagrammes au niveau conceptuel en construisant un diagramme de classes et le modèle relationnel de notre base de données.

La réalisation fera l'objet d'un quatrième et dernier chapitre dans lequel nous décrivons

les choix technologiques effectués concernant l'environnement de développement de notre environnement et l'architecture matérielle du système, suivi d'une présentation des différentes interfaces de notre environnement pour le suivi du quotidien des patients atteints de maladies chroniques.

Enfin, nous clôturerons ce mémoire par une conclusion générale résumant les points essentiels de notre travail et dégageons quelques perspectives envisagées pour notre application mobile.

Chapitre 1

La Santé Mobile

1.1 Introduction

Depuis son apparition, le mobile a marqué un grand changement dans notre quotidien de telle sorte à nous faciliter beaucoup de tâches. L'avantage le plus remarquable dans ceci est de pouvoir télécharger et installer des nouvelles applications qui nous offrent pleins de services gratuits ou payants dans la majorité des secteurs à savoir l'éducation, sport, industrie et la santé. Le secteur de la santé s'est donc saisi de ces services tant du côté des professionnels de santé, autrefois réservés au milieu médical, devient grâce au numérique accessible au grand public.

Dans ce chapitre nous commencerons par définir l'application mobile et bien précisément l'application mobile de santé, puis nous présenterons le système d'exploitation qui adopte ce genre d'application à savoir Android et nous citerons ses fonctionnalités suivies de son architecture, et enfin nous allons clôturer ce premier chapitre par un résumé de l'ensemble des points cités auparavant.

1.2 Applications mobiles

Une part croissante de la population accède aux informations et aux services de santé par la téléphonie mobile qui offre un large éventail de solutions des SMS jusqu'aux applications complexes sur smartphone.

1.2.1 Historique

La naissance des applications mobiles remonte à la fin du vingtième siècle, leur industrie a commencé quand l'iPhone d'Apple et leur magasin iTunes ont été introduits en 2007. C'était aussi le moment où les utilisateurs ont commencé à utiliser leurs appareils mobiles pour surfer sur internet, Seules quelques personnes regardaient leurs courriers électroniques, mais avec l'iPhone le taux de consommateurs a augmenté et les téléphones mobiles ne sont plus utilisés que pour des appels téléphoniques, néanmoins pour beaucoup plus que ça. Puisque le nombre de nouvelles applications mobiles s'accroissent de jour en jour et ces applications nous offrent plusieurs services tels que la consultation du courrier électronique, la téléphonie-IP etc [1].

1.2.2 Définition

Au moment du lancement de cette nouvelle invention technologique, tout le monde se posait la même question : Qu'est-ce qu'une application mobile ?

Une application mobile est un logiciel applicatif développé pour un appareil électronique mobile, tel qu'un assistant personnel, un téléphone portable, un smartphone, un baladeur numérique, une tablette tactile, ou encore certains ordinateurs fonctionnant avec le système d'exploitation Windows Phone ou Chrome OS. [2]

Une application mobile est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation d'un dispositif mobile (PDA, smartphone, tablette). Les applications mobiles sont distribuées depuis des plateformes de téléchargement tels que l'App Store (plateforme d'Apple), le Google Play (plateforme de Google/Android), ou encore le Windows Phone Store (plateforme de Microsoft) [2][3].

1.2.3 Objectifs d'une application mobile

C'est parce que le mobile simplifie en premier lieu la vie du client, par téléphone, mail, réseaux sociaux. . . ou sur l'application.

Pour une entreprise ou une enseigne, il existe 3 principaux objectifs pour une application mobile à savoir [2] :

- Améliorer le service client ;
- Améliorer les processus de traitement des demandes clients ;
- Amplifier l'intensité relationnelle avec le client.

1.3 Applications santé mobile

Les technologies numériques comme les applications mobile de santé sont en train de devenir une ressource importante pour la fourniture de services de santé et la santé publique et pour cela nous allons approfondir dans les prochaines titres sur les différents domaines de ces technologies.

1.3.1 E-santé

E-santé (en anglais, E-health) désigne le domaine consistant à rassembler toutes les nouvelles technologies pour les services de santé. C'est pourquoi quand on parle de e-santé, on peut aussi parler de santé connectée.

E-santé réunit à la fois les technologies numériques, la médecine, la santé publique et les entreprises. C'est une notion qui est apparue au début des années 2000. Il s'agit d'un secteur qui ne cesse de se développer. D'abord, parce qu'il apporte des réelles solutions aux patients et aux professionnels de santé pour améliorer la qualité des soins. Ensuite, à cause des innovations technologiques qui sont de plus en plus poussées. Enfin, parce que la santé connectée représente un marché fructueux pour les entreprises [4].

— Les avantages de E-santé :

Il y a quelques années, la e-santé était présentée comme réponse à plusieurs problématiques du système de santé [4] :

1. En premier lieu, c'est un moyen de réduire les coûts. Par exemple, la télémédecine peut éviter le transport inutile d'un malade à l'hôpital. La transmission des documents de santé par une application sécurisée évite quant à elle les dépenses liées à l'impression des documents et à leur envoi par la poste ;

2. En deuxième lieu, elle offre une réponse aux problèmes des déserts médicaux. Grâce à des plateformes de téléconsultation, les patients qui n'ont pas accès aux soins en raison de l'absence d'un médecin proche de chez eux peuvent quand même consulter un professionnel. Avec les distributeurs de médicaments, ils peuvent également accéder à leur traitement, et ce même s'ils sont loin d'une pharmacie ;
3. En troisième lieu, les dispositifs de santé connectée renforcent le suivi des maladies chroniques. Certains objets connectés permettent en effet, aux patients d'effectuer eux-mêmes certaines mesures, comme le taux d'insuline ou la tension, consulter, prévenir automatiquement leurs médecins en cas de résultats inquiétants ;
4. Finalement, e-santé est un synonyme de progrès médical. Elle est à l'origine de nombreux appareils et solutions qui facilitent et améliorent le bien-être et la prise en charge des patients par les équipes médicales. Ces dernières peuvent alors diagnostiquer plus rapidement une maladie, réaliser des opérations chirurgicales moins intrusives, etc.

1.3.2 M-santé

La santé mobile est parmi les domaines de la santé connectée, elle recouvre les pratiques médicales et de santé publique reposant sur des dispositifs mobiles, tels que les téléphones portables qui peuvent collecter, stocker, traiter et diffuser des données ou peuvent accomplir des actions spécifiques en fonction des informations reçues. Elles effectuent des mesures en temps réel et permettent de renseigner de nombreux paramètres influant sur la santé : poids, température corporelle, pouls, tension, rythme respiratoire, rythme cardiaque, taux de glycémie, qualité de sommeil etc.

Cela englobe également les applications concernant le mode de vie et le bien-être qui peuvent se connecter à des dispositifs médicaux, ainsi que les systèmes de conseil personnalisés. La première utilité est l'alerte de tout changement d'un des paramètres transmis de manière brutale ou atteignant une valeur critique préalablement fixée incite à une intervention, une surveillance particulière, à des conseils ou à des recommandations [5].

— **Les avantages de m-santé :**

La m-santé a été proposée pour atteindre plusieurs objectifs à savoir [5] :

- La communication sur de longues distances ;
- Des méthodes informatisées de collecte et d'évaluation des données de santé ;
- Un meilleur suivi des patients, par exemple les maladies chroniques ;
- Des meilleures options de traitement dans les zones rurales pour lutter contre les déserts médicaux.

1.4 Systèmes d'exploitation mobiles

Dans le secteur informatique, les systèmes d'exploitation les plus répandus sont Windows, Mac OS pour les ordinateurs d'Apple, Linux et Unix pour les serveurs. Pour les téléphones on trouve Android, iOS.

1.4.1 Définition

Un système d'exploitation mobile ou MOS "Mobile Operating System " en anglais est un système d'exploitation conçu pour fonctionner sur un appareil mobile. Ce type de système d'exploitation se concentre entre autres sur la gestion de la connectivité sans fil et celle des différents types d'interface [6].

1.4.2 Type de système d'exploitation mobiles

Les système d'exploitation les plus répandus sont [6] :

- **Android** : est un système d'exploitation fondé sur un noyau Linux. Disponible via une licence Apache version 2, le système d'exploitation inclut tous les utilitaires requis par un constructeur ou par un opérateur pour mettre en œuvre un téléphone portable. Android a été conçu pour intégrer au mieux des applications existantes de Google comme le service de courrier Gmail, ou celui de cartographie, Google Maps, ou encore Google Agenda, Google Talk, YouTube.
- **iOS ou iPhone OS** : iOS, anciennement iPhone OS, est le système d'exploitation mobile développé par Apple pour l'iPhone. Reconnu pour sa fluidité, son ergonomie et son intuitivité : c'est le système d'exploitation le plus abouti à ce jour. Il dispose du portail App Store, qui avec un catalogue de 500 000 applications.
- **Windows Mobile** : C'est le système d'exploitation conçu par Microsoft. Il est présent sur les téléphones Windows. Les applications pour Windows Mobile ont été conçues à partir de la suite Visual Studio.

1.5 Etude de la plateforme Android

1.5.1 Présentation d'Android

Android est un système d'exploitation pour Smartphones, tablettes tactiles, PDA et terminaux mobiles. C'est un système open source, utilisant le noyau Linux, conçu par Android, une startup rachetée par Google, et annoncé officiellement le 5 novembre 2007. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs, des montres, des autoradios et même des voitures. Il est basé essentiellement sur la simplicité d'utilisation et surtout sur une capacité de personnalisation importante [7].

1.5.2 Fonctionnalités d'Android

Les fonctionnalités proposées par Android diffèrent d'une version à une autre, on peut citer les plus importantes [7] :

- Augmentation de la performance ;
- Fonctionnalité de Hot spot Wifi ;
- Partage de contact sur Bluetooth ;
- Ecran d'accueil personnalisable ;
- Disponibilité des Widgets ;
- La possibilité d'échanger directement des fichiers entre deux téléphones android à proximité ;
- La possibilité de filmer et de prendre des photos en même temps ;
- Mise à jour automatique des applications.

1.5.3 Architecture de la plateforme Android

L'architecture Android contient un nombre important des composants pour prendre en charge tous les besoins des appareils Android. Le logiciel Android contient un noyau linux open source ayant une collection d'un certain nombre de bibliothèques C/C++ qui sont exposées via des services d'infrastructure d'application.

Parmi tous les composants, le noyau linux fournit la fonctionnalité principale des fonctions du système d'exploitation aux smartphones et la machine virtuelle Dalvik (DVM) fournit une plate-forme pour exécuter une application Android.

D'habitude la lecture de cette architecture se fait d'une manière ascendante (de bas en haut)[6] [7] :

1. **Noyau Linux** : Le noyau linux est le cœur de l'architecture Android. Il gère tous les pilotes disponibles tels que les pilotes d'affichage, les pilotes de caméra, les pilotes Bluetooth, les pilotes audio, les pilotes de mémoire, etc. qui sont nécessaires pendant l'exécution.

Le noyau linux fournira une couche d'abstraction entre le matériel de l'appareil et les autres composants de l'architecture Android. Il est responsable de la gestion de la mémoire, de l'alimentation, des périphériques, etc.

Les fonctionnalités du noyau linux :

Parmi les services fournis par ce dernier sont présentés ci-dessous :

- **Sécurité** : Le noyau linux gère la sécurité entre l'application et le système ;

- **Gestion de la mémoire** : il gère efficacement la gestion de la mémoire, offrant ainsi la liberté de développer nos applications ;
 - **Gestion des processus** : il gère bien le processus, alloue des ressources aux processus chaque fois qu'ils en ont besoin ;
 - **Network Stack** : Il gère efficacement la communication réseau ;
 - **Modèle de pilote** : il garantit que l'application fonctionne correctement sur les fabricants des périphériques et matériaux responsables de la création de leurs pilotes dans la version linux.
2. **Platform libraires** : Les bibliothèques de plate-forme comprennent diverses bibliothèques principales C/C++ et des bibliothèques basées sur Java telles que Media, Graphics, Surface Manager, OpenGL, etc. pour fournir un support pour le développement Android.
- La bibliothèque multimédia fournit un support pour lire et enregistrer des formats audio et vidéo ;
 - Gestionnaire de surface chargé de gérer l'accès au sous-système d'affichage ;
 - SGL et OpenGL, deux interfaces de programme d'application (API) multi-langues et multiplateformes, sont utilisées pour l'infographie 2D et 3D ;
 - SQLite fournit un support de base de données et FreeType fournit un support de police ;
 - Web-Kit Ce moteur de navigateur Web open source fournit toutes les fonctionnalités pour afficher le contenu Web et pour simplifier le chargement des pages ;
 - SSL (Secure Sockets Layer) est une technologie de sécurité permettant d'établir un lien crypté entre un serveur Web et un navigateur Web.
3. **Application runtime** : L'environnement d'exécution Android est l'une des parties les plus importantes d'Android. Il contient des composants tels que les bibliothèques principales et la machine virtuelle Dalvik (DVM). Il fournit principalement la base du Framework d'application et alimente notre application à l'aide des bibliothèques principales.

Comme Java Virtual Machine (JVM), Dalvik Virtual Machine (DVM) est une machine virtuelle basée sur un registre et spécialement conçue et optimisée pour Android afin de garantir qu'un appareil peut exécuter plusieurs instances efficacement. Cela dépend de la couche du noyau linux pour le threading et la gestion de la mémoire de bas niveau. Les bibliothèques de base nous permettent d'implémenter des applications Android en utilisant les langages de programmation standard JAVA ou Kot lin.

4. **Application Framework** : Application Framework fournit plusieurs classes importantes qui sont utilisées pour créer une application Android. Elle fournit une abstraction générique pour l'accès au matériel et aide également à gérer l'interface utilisateur avec les ressources d'application. Généralement, elle fournit les services à l'aide desquels nous pouvons créer une classe particulière et rendre cette classe utile pour la création d'application.

Elle comprend différents types de services gestionnaires d'activités, gestionnaire des notifications, système de vue, gestionnaire de package, etc. qui sont utilisés pour le développement de notre application en fonction des prérequis.

5. **Applications** : Les applications sont la couche supérieure de l'architecture Android. Les applications préinstallées telles que les contacts, l'appareil photo, la galerie, etc. et les applications tierces téléchargées à partir du Play Store telles que les applications de chat, les jeux, etc. seront installés sur cette couche uniquement. Ils s'exécutent dans le temps d'exécution d'Android à l'aide des classes et des services fournis par le Framework d'application.

La figure 1.3 nous montre cette architecture [8] :

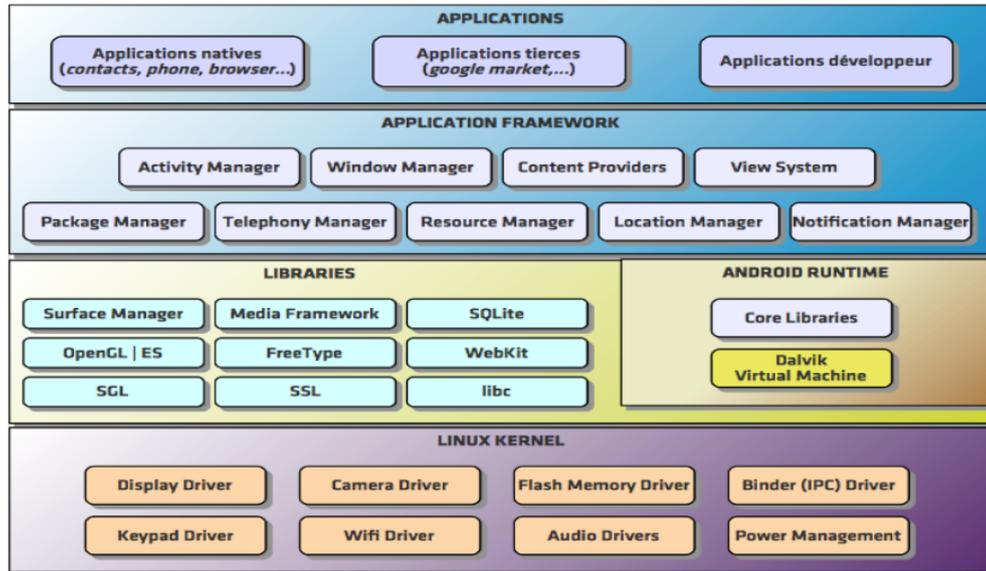


FIGURE 1.1 – Architecture d'android

1.5.4 Cycle de vie d'une application Android

Sous Android, une application est composée d'une ou plusieurs activités. Une activité est la base d'un composant pour la création d'interfaces utilisateur. Afin de faciliter la cinématique (mouvement) de l'application, il est préconisé de n'avoir qu'une interface visuelle par activité. Le fonctionnement du cycle de vie d'une application mobile se fait comme suit :

Au démarrage d'une activité ; la méthode `onCreate` est appelée. Pendant l'utilisation de cette 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.

Une fois l'appel téléphonique terminé, le système réveille l'activité précédemment mise en pause (appel de `onRestart`, `onStart`).

Si 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'activités partiellement visibles, 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 souhaitera y revenir avec la touche Accueil[9].

la figure 1.4 propose une vision d'un cycle de vie d'une application Android [9].

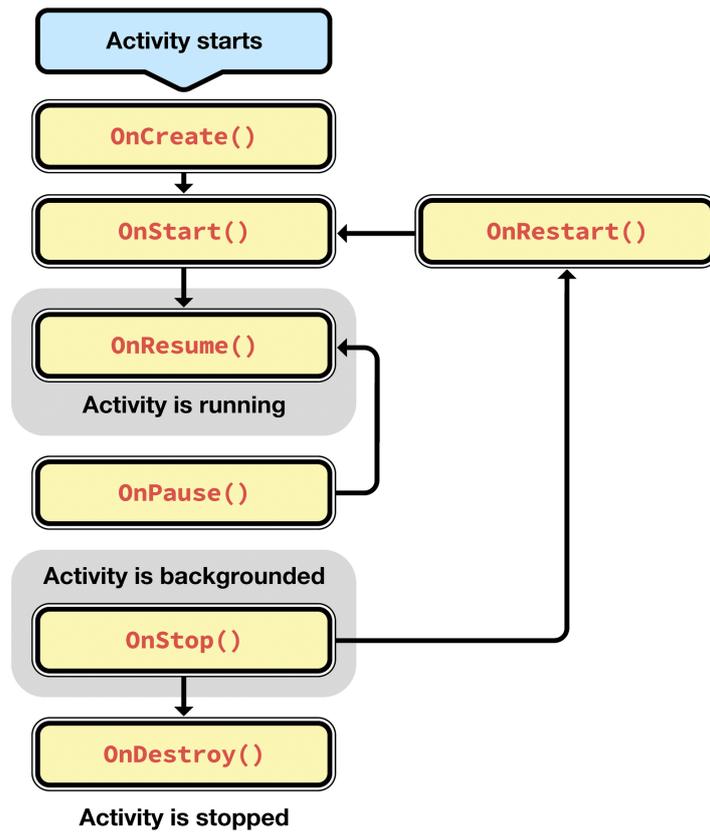


FIGURE 1.2 – Cycle de vie d’une application Android

1.6 Conclusion

Dans ce chapitre nous avons entamé la partie théorique et les différentes définitions des applications mobiles. En outre nous avons clarifié les bienfaits apportés sur le secteur de santé pendant l’intégration des technologies de l’information et de communications. Enfin, nous avons fait une étude du système d’Android qui approuve ce genre d’application tout en présentant un bref historique, les fonctionnalités de ce dernier et son architecture. Ceci nous a permis de bien acquérir les capacités et les notions de base à exploiter durant les chapitres à venir.

Chapitre 2

Cahier Des Charges

2.1 Objectif du chapitre

Le présent chapitre représente un cahier des charges pour la conception et la réalisation d'un environnement m-santé pour le quotidien des patients atteints de maladies chroniques. Il est destiné à servir d'appui aux démarches d'informatisation des différentes fonctionnalités de notre application.

2.2 Étude de l'art

Il existe de nombreuses applications de santé permettent aujourd'hui aux patients de veiller sur leur état de santé général et aux professionnels de santé d'améliorer la qualité de la prise en charge médicale.

2.2.1 Modèles existant

selon notre recherche qui s'est déroulée sur les applications mobile de santé en algérie nous avons trouver deux modèles que nous allons les présenter ci-dessous :

Atibaa El Djazaïr

Atibaa El Djazaïr (Médecins d'Algérie) est une application mobile recensant les médecins selon leurs spécialités ainsi que les établissements de santé. Disponible uniquement en langue arabe.

Annuaire Médical Algérie

Est une application mobile destinée aux algériens et leurs permet de retrouver les hôpitaux et les pharmacies et les laboratoires d'analyses .

2.2.2 APPORT DU PROJET

D'après cette recherche nous avons voulu de vous faire découvrir les avantages de notre application par rapport à ces deux dernières applications présenter ci-avant Notre application est une application mobile destinée aux grands publics soit pour suivre des maladies ou bien pour vérifier seulement la norme de leurs résultats en interprétant ces derniers ainsi qu'un utilisateur pourra profiter des avantages d'informations à savoir des hôpitaux, des cabinets médicaux, ambulances, donneurs de sang et des pharmacies .

D'autre part, nous assurant toute validité et sureté des informations publiées en associant notre application avec une autre application web pour les administrateurs.

2.3 Perspectives du projet

La population est vieillissante et par conséquent les médecins sont amenés à suivre de plus en plus des patients souffrant d'une maladie chronique. Ces patients sont à la fois les premiers concernés par ces soins, mais également la population qui se renseigne le plus sur sa santé. L'inscription dans la durée de la pathologie force à réfléchir à des systèmes facilitants leurs prises en charge au quotidien. L'émergence de la m-santé permettrait potentiellement de déporter la prise en charge de ces patients au domicile grâce à la miniaturisation des moyens de surveillance. Ce type d'applications contribue à améliorer la qualité de vie des patients souffrant des maladies, grâce aux possibilités de surveillance et au suivi qu'elles offrent.

2.4 Design du projet

Le design de l'application est un facteur déterminant pour de nombreux utilisateurs. En effet, lorsque vous téléchargez l'application, la première chose que vous remarquerez sera le design et surtout les couleurs. Le choix de la couleur n'est pas seulement une question d'esthétique ou de tendance du design application mobile. Nous devons nous rappeler que la couleur est aussi une façon de communiquer, de sorte qu'elles seront significatif. Dans ce contexte, nous avons choisis le bleu, le vert et le blanc pour nos applications pour les significations suivantes :

- Bleu : apporte de la sérénité et de la paix à l'utilisateur et Génère un sentiment de sécurité et de confiance aux patients.
- Vert : associée avec la nature et la santé et comme déjà prévue le projet est destinées aux patients et aux médecins.
- Blanc : signifie la sureté et la simplicité, la sureté est réservée pour les informations publiées.

2.5 Membres de l'équipe et les tâches attribués dans ce projet

Pour mener à bien le projet, les membres de l'équipe doivent connaître au tout début chacune de leurs tâches d'une manière claire et explicite pour éviter un éventuel retard de livraison. Pour ce faire, nous nous sommes concertés et nous avons optés pour une équipe composée d'une coordinatrice menée pour des tâches de design et un développeur /testeur.

La table 2.1 détaille des tâches attribués par les membres de l'équipe :

2.6 Présentation du projet

Le projet à réaliser s'intitule « conception et réalisation d'un environnement pour le suivi des maladies chroniques » cet environnement comporte en premier lieu une application mobile permettant aux différents patients de suivre leurs maladies via un smartphone. Cette application mobile sera le carnet de santé de chaque patient, ce dernier lors de son accès à son profil pourra revoir toutes les valeurs prises en avant ce qui facilite les tâches aux médecins afin de mieux suivre l'état de chaque patient. En plus du suivi et de l'enregistrement, cette application réduira une certaine distance entre le médecin et son patient en lui offrant la possibilité de le communiquer en cas d'urgences ou de résultats inquiétants, tous ces services sont autorisés à un utilisateur doté d'un compte néanmoins pour le visiteur il est autorisé à consulter et ajouter à la liste des médecins, hôpitaux, donneurs de sang et ambulances. Et en second lieu une application web permettant à un administrateur de gérer toute action faite par un visiteur ou un utilisateur depuis l'application mobile.

Membres	Tâches
I FOURAH Meriem	<ul style="list-style-type: none"> — Coordinatrice : — Prendre des notes lors des réunions de planification et d'examen du projet. — Définir le plan de réalisation ; — Etablir la liste des risques ; — Définir le plan et le statut du projet. — Design — Etablir des maquettes pour la version web et mobile ; — Choisir les couleurs ;
Hocini Iskandar	<ul style="list-style-type: none"> — Développeur/Testeur : — Code source (Front-end et Back-end) ; — Effectuer les tests et résoudre les bugs ; — Rédiger le manuel d'utilisation.
Tous les membres de l'équipe	Conception.

TABLE 2.1 – Membres d'équipe et les tâches attribuées.

2.7 Spécifications non-fonctionnelles

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 système se résument comme suit :

- Le code doit être clair pour permettre à des futures évolutions ou améliorations ;
- L'ergonomie : l'application offre une interface conviviale et facile à utiliser ;
- La sécurité : l'application doit respecter la confidentialité des données ;
- Garantir l'intégrité et la cohérence des données à chaque mise à jour et à chaque insertion.

2.8 Spécifications fonctionnelles

2.8.1 Identification des acteurs

Avant de citer les spécifications fonctionnelles de notre environnement, nous allons présenter les différents acteurs qui interagissent avec le système.

La table 2.2 Représente les différents acteurs et leurs définitions

acteur	définition
Visiteur	Désigne une personne qui utilise l'application mobile sans avoir un compte avec un accès limité.
Patient	Désigne une personne ayant un compte sur l'application mobile et pouvant ainsi profiter pleinement de l'ensemble des fonctionnalités.
Médecins	désigne une personne ayant un compte sur l'application et profitant des services conçus spécialement pour elle.
Administrateur	désigne une personne qui administre l'application assurant ainsi sa bonne gestion.

TABLE 2.2 – Identification des acteurs.

2.8.2 Fonctionnalités :

Après l'identification des acteurs, nous allons définir les fonctionnalités de chaque acteur défini dans le tableau précédent.

— **Visiteur :**

- Consulter/ajouter des médecins à la liste des médecins : l'application offre un espace qui contient une liste des médecins pour chaque wilaya dont le visiteur pourra ajouter à cette liste comme il pourra la consulter ;
- Consulter/ajouter des donneurs de sang à la liste des donneurs : l'application offre un espace qui contient une liste des donneurs pour chaque wilaya dont le visiteur pourra ajouter à cette liste comme il pourra la consulter ;
- Consulter/ajouter des ambulances à la liste des ambulances : l'application offre un espace qui contient une liste des ambulances pour chaque wilaya dont le visiteur pourra ajouter à cette liste comme il pourra la consulter ;
- Consulter/ajouter des pharmacies à la liste des pharmacies : l'application offre un espace qui contient une liste des pharmacies pour chaque wilaya dont le visiteur pourra ajouter à cette liste comme il pourra la consulter ;
- Consulter/ajouter des hôpitaux à la liste des hôpitaux : l'application offre un espace qui contient une liste des hôpitaux pour chaque wilaya dont le visiteur pourra ajouter à cette liste comme il pourra la consulter ;
- Consulter la liste des numéros d'urgence ;
- Donner son avis : le visiteur pourra donner un avis sur l'application .

— **Patient :**

- Suivre ses maladies : notre application offre le suivi des maladies suivantes :
 - Le diabète :
 - Consulter tableau index glycémique.
 - Consulter un régime alimentaire|un programme sportif.
 - Remplir le carnet d'auto surveillance glycémique et les enregistrer.
 - Rappel des prises des médicaments.
 - Générer une page d'aide.
 - L'obésité :
 - Calcul de la masse corporelle après avoir saisi le poids et la taille.
 - Consulter le tableau des catégories de poids après le calcul de l'IMC
 - Consulter un régime alimentaire|un programme sportif.
 - L'hypertension artérielle
 - Consulter les étapes à suivre pour utiliser l'appareil à la maison
 - Conseil et horaire de mesure de la tension
 - Cholestérol :
 - Comprendre les résultats des analyses.
 - Consulter un régime alimentaire|un programme sportif. .
 - Générer une page d'aide.
 - Crohn :
 - Consulter les symptômes et les complications de cette maladie.
 - Les causes de cette maladie.
 - Consulter un régime alimentaire.
- Prendre un rendez-vous : le patient utilisant cette application il pourra prendre un rendez-vous chez un médecin ;
- Contacter un médecin : l'utilisateur de cette application profite de la rubrique communication avec un médecin ;
- Générer son carnet de santé : toutes les valeurs insérer ou générer par l'application son enregistrés et afficher dans cet espace.

— **Médecin :**

- Contacter un malade : le médecin répond aux patients déjà qui l'ont contactés ;
- Gérer les rendez-vous : les medecins dotés des comptes de l'application pourra modifier et supprimer les listes de leurs rendez-vous ;

— **Administrateur :**

- Gérer la liste des hôpitaux : l'administrateur est le responsable de la validité des informationtions dans la liste des hôpitaux donc il est chargé d'ajouter , modifier , supprimer les informations.
- Gérer la liste des médecins : l'administrateur est le responsable de la validité des informationtions dans la liste des médecins donc il est chargé d'ajouter , modifier , supprimer les informations.
- Gérer la liste des donneurs de sang : l'administrateur est le responsable de la validité des informationtions dans la liste des donneurs de sang donc il est chargé d'ajouter , modifier , supprimer les informations.
- Gérer la liste des pharmacies : l'administrateur est le responsable de la validité des informationtions dans la liste des pharmacies donc il est chargé d'ajouter , modifier

- supprimer les informations.
- Gérer la liste des ambulances : l'administrateur est le responsable de la validité des informations dans la liste des ambulances donc il est chargé d'ajouter modifier supprimer les informations.
- voir et supprimer les avis.

2.9 Contraintes Générales

Chaque projet a des limites et risques, qui doivent être prises en compte et gérées afin d'assurer la réussite finale du projet. Les trois principales contraintes qui doivent être prise en compte sont la durée, la portée et le coût. Elles sont souvent appelées triples contraintes ou triangle de la gestion de projet. Chaque contrainte est liée aux deux autres. Ainsi, l'augmentation de la portée du projet demandera plus de temps et de budget, tandis qu'une accélération du calendrier peut diminuer les coûts, mais également la portée [10].

la figure 2.1 Enrichie le paragraphe précédent et montre l'objectif de ce triangle.

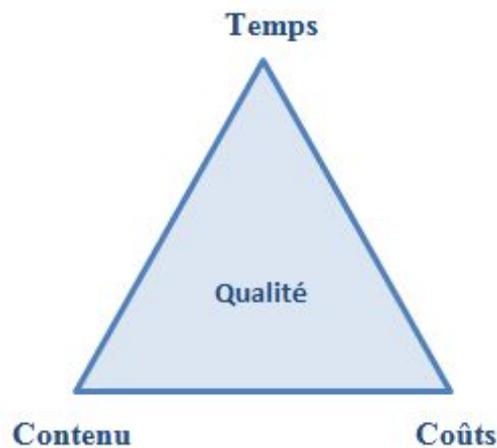


FIGURE 2.1 – Triangle de gestion de projet

2.9.1 Qualité

La qualité est la conformité par rapport aux attentes ou aux exigences définies pour le projet au départ, et le logiciel doit pouvoir être utilisés avec le maximum de confort, de sécurité et d'efficacité par le plus grand nombre d'utilisateurs. Ainsi que le choix des technologies peut affecter la rapidité, l'efficacité, ainsi que sur le résultat donné par le logiciel et donc sur la qualité.

2.9.2 Délais

Le tableau 2.3 nous montre le temps accordé pour l'accomplissement du projet

tâche	Date de début	Date de fin
Spécification des besoins	5/03/2022.	17/03/2022
Conception <ul style="list-style-type: none"> — V 1 — V 2 	<ul style="list-style-type: none"> — 13/03/2022 — 17/03/2022 	<ul style="list-style-type: none"> — 15/03/2022 — 31/03/2022
Design	03/04/2022.	15/04/2022
Développement	6/05/2022.	30/06/2022

TABLE 2.3 – Le temps accordé pour l’accomplissement du projet.

2.9.3 Budget

Un budget fait état des dépenses prévisionnelles planifiées pour ce projet. Donc c’est un outil incontournable pour la boîte de développement qui l’utilisent à des fins de pilotage, de prévision et de contrôle des activités, ainsi pour les clients pour qu’ils sachent s’ils sont prêts à dépenser une telle somme.

2.10 Contraintes Technologiques

2.10.1 Méthode de travail

Le processus unifié est une méthode de développement pour les logiciels orientés objets. C’est une méthode générique, itérative et incrémentale. Il vient compléter la systémique des modèles UML. [11]. Nous définissons dans le prochain chapitre les différents principes de ce processus

2.10.2 Méthodes et outils de modélisation

1. Langage de modélisation :

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML modélise l’ensemble des données et des traitements en élaborant des différents diagrammes [12][13].

2. Les outils de modélisation :

- **Astah** : Anciennement appelé Jude (Java and UML Developers Environment), est un outil de modélisation UML créé par la compagnie japonaise (ChangeVision). Il est très facile et simple à manipuler et offre plusieurs possibilités et formes nécessaire pour la conception des diagrammes UML qui vont être le sujet de ce rapport [14].
- **Visual Paradigme** : Notre choix a porté sur l’outil Visual Paradigm for UML 16.2 pour la production de différents diagrammes. Visual Paradigm est un logiciel de création de diagrammes dans le cadre d’une programmation et possède plusieurs options permettant une large possibilité de modélisation en UML [15].

2.10.3 Environnement et langages de développement

1. Langages de développement :

- **Mobile** :

- Flutter.
- **Web**
- **Front end**
 - Html;
 - CSS;
 - JavaScript ;
 - Bootstrap .
- **Back end**
 - PHP.

2. Environnement de développement

- Sdk Android
- Android Studio
- IntelliJ IDEA
- VS Code

2.11 Conclusion

Dans ce chapitre, nous avons donc élaboré un cahier des charges, qui, selon sa définition technique, représente un contrat qui fournit des informations détaillées sur le projet. Nous avons d'abord défini les acteurs et les fonctionnalités de notre environnement puis nous avons recensé les besoins et les limites de ce système et enfin, nous avons cité brièvement les différents outils et environnements de travail utilisés pour la réalisation de notre environnement.

Chapitre 3

Conception

3.1 Introduction

Avant de passer au développement de notre application, il est indispensable d'identifier la méthode de conduite du projet et les étapes à suivre pour y parvenir.

Dans ce chapitre, nous décrirons la méthode de conception et la présentation des diagrammes des cas d'utilisation pour chaque acteur puis nous passerons à la description textuelle des cas d'utilisation et la présentation des diagrammes de séquence. Enfin la phase de conception s'achèvera par l'élaboration du diagramme de classes conceptuel.

3.2 Méthodologie de conception

3.2.1 Définition du processus unifié (UP)

Le processus unifié (Unified Process) est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus 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 l'entreprise. Nous définissons dans ce qui suit les différents principes de ce processus [11].

1. **Cycle de vie d'un processus unifié :**

La gestion d'un tel processus est organisée suivant les quatre phases : création, élaboration, construction et transition [16] :

Phase de Création	<p>Traduit une idée en vision de produit fini et présente une étude de rentabilité pour la création de ce produit</p> <ul style="list-style-type: none"> — Que va faire le système pour les utilisateurs ? — A quoi peut ressembler l'architecture d'un tel système ? — Quels sont les organisations et les coûts du développement de ce produit ? <p>On fait apparaître les principaux cas d'utilisation. L'architecture est provisoire, identification des risques majeurs et planification de la phase d'élaboration.</p>
Phase d'élaboration	<p>Permet de préciser la plupart des cas d'utilisation et de concevoir l'architecture du système. L'architecture doit être exprimée sous forme de vue de chacun des modèles. Emergence d'une architecture de référence.</p> <p>A l'issue de cette phase, le chef de projet doit être en mesure de prévoir les activités et d'estimer les ressources nécessaires à l'achèvement du projet.</p>
Phase de construction	<p>Moment où l'on construit le produit. L'architecture de référence se métamorphose en produit complet, elle est maintenant stable. Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs on décide de mettre au point pour cette version. Celle-ci doit encore avoir des anomalies qui peuvent être en partie résolus lors de la phase de transition.</p>
Phase de transition	<p>Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts. Cette phase suppose des activités comme la fabrication, la formation des utilisateurs client, la mise en œuvre d'un service assistant et la correction des anomalies constatées. (Ou le report de leur correction est la version suivante).</p>

TABLE 3.1 – Cycles de vie UP.

2. Activités du processus

Pour mener efficacement un tel cycle, les développeurs ont besoin de toutes les représentations du produit logiciel qui se définissent à partir des étapes suivantes [16][17] :

— Expression des besoins :

Comme son nom l'indique, cette étape permet de définir les différents besoins comme :

- Inventorier les besoins principaux et fournir une liste de leurs fonctions.
- Recenser les besoins fonctionnels (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation.
- Appréhender les besoins non fonctionnels (techniques) et livrer une liste des exigences

— Analyse :

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences de l'utilisateur. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structures sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système.

— **Conception :**

La conception permet d'acquérir une compréhension approfondie des contraintes liées aux langages de programmation, à l'utilisation des composants et au système d'exploitation.

Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune et elle décompose le travail d'implémentation en sous-système. Elle constitue un point de départ à l'implémentation.

— **Test :**

Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mène à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

3.2.2 Diagramme de contexte

Le diagramme de contexte permet de définir les limites de l'étude. Il place le diagramme dans son contexte en listant les acteurs ou éléments qui agissent ou interagissent avec lui[18].

La figure 3.1 Représente le diagramme de contexte statique du système à réaliser. Bien qu'il ne fasse pas partie des diagrammes UML, nous avons trouvé utile de le réaliser car il montre le nombre d'instances de chaque acteur (au maximum) connectées au système à un instant donné.

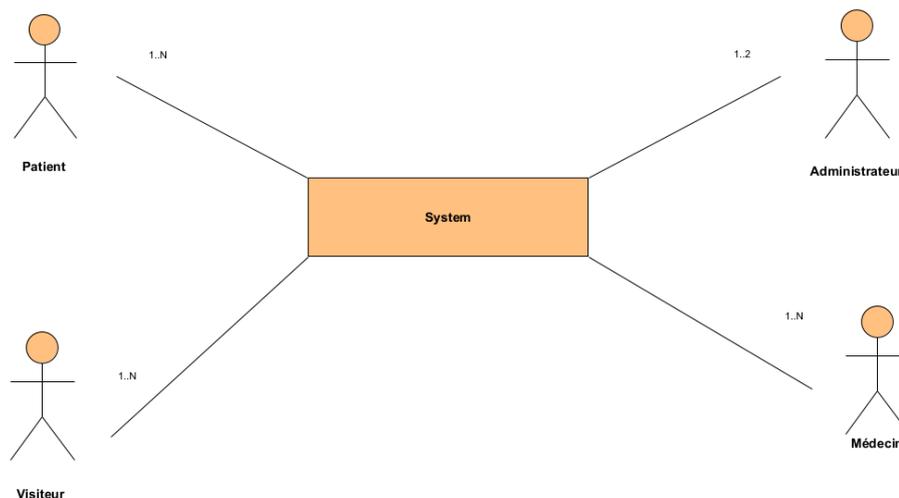


FIGURE 3.1 – Diagramme de contexte statique du système à réaliser.

3.2.3 Identification des cas d'utilisations.

Un cas d'utilisation (use case) représente un ensemble de séquences d'action qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera [19]. On obtient un diagramme de cas d'utilisation en représentant sur un schéma les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs (personnage) [20].

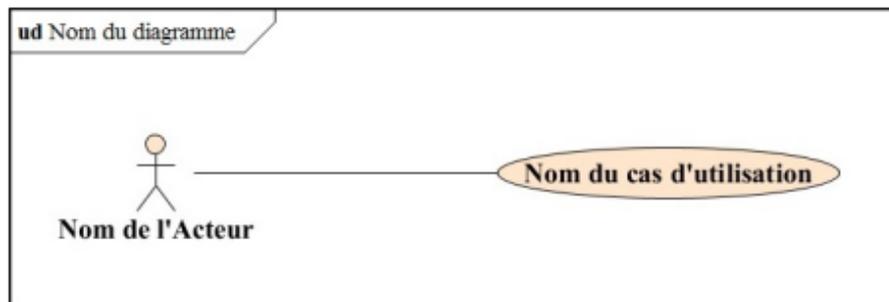


FIGURE 3.2 – Formalisme de représentation d'un diagramme de cas d'utilisation.

Afin de bien organiser les cas d'utilisation, UML définit trois (03) types de relations standardisées entre cas d'utilisation, détaillées ci-après [20] :

— La relation d'inclusion

Le cas de base en incorpore explicitement un autre, à un endroit spécifié dans ces enchaînements. Le cas d'utilisation inclus n'est jamais exécuté seul, mais seulement en tant que partie d'un cas de base plus vaste. On utilise fréquemment cette relation pour éviter de décrire plusieurs fois le même enchaînement, en factorisant le comportement commun dans un cas d'utilisation à part.

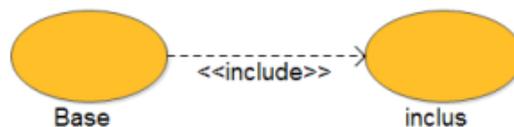


FIGURE 3.3 – Relation d'inclusion.

— La relation d’extension

Le cas de base en incorpore implicitement un autre, à un endroit spécifié indirectement dans celui qui étend. Le cas de base peut fonctionner tout seul, mais il peut également être complété par un autre, sous certaines conditions, et uniquement à certains points particuliers de son flot d’événements appelés points d’extension. On utilise principalement cette relation pour séparer le comportement optionnel (les variantes) du comportement obligatoire.



FIGURE 3.4 – Relation d’extension.

— La relation de généralisation/spécialisation

Les cas d’utilisation peuvent être hiérarchisés par généralisation/spécialisation. Les cas d’utilisation descendants héritent de la sémantique de leur parent. Ils peuvent comprendre des interactions spécifiques supplémentaires, ou modifier les interactions héritées.

Reprenons un à un les quatre acteurs et listons les différentes façons qu’ils ont d’utiliser le futur système.

1. **Le visiteur** : Utilise l'application sans inscription pour consulter et ajouter dans les 58 wilayas des :
 - Médecins à la liste des médecins ;
 - Donneurs de sang à la liste des donneurs de sang ;
 - Hôpitaux à la liste des hôpitaux ;
 - Pharmacies à la liste des pharmacies ;
 - Des ambulances à la liste des ambulances ;
 - Consulter la liste des numéros d'urgence ;
 - Donner un avis.

la figure 3.5 illustre le diagramme de cas d'utilisation associé à un visiteur.

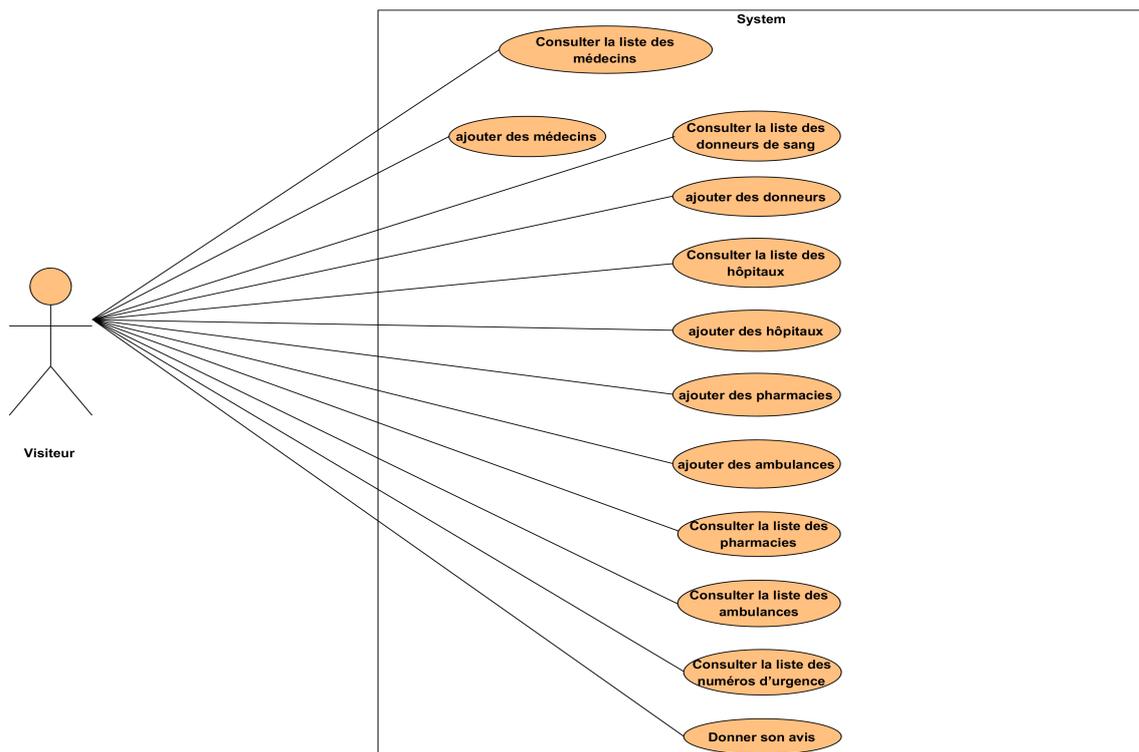


FIGURE 3.5 – Diagramme de cas d'utilisation associé à un visiteur.

2. **Le patient** : Le patient est un visiteur qui hérite toutes ces fonctionnalités même après son authentification, néanmoins, il profite également de certaines fonctionnalités spécifiques à ce dernier, à savoir :
 - Suivre ses maladies ;
 - Prendre un rendez-vous ;
 - Contacter un médecin ;
 - Générer son carnet de santé.

la figure 3.6 Illustré le diagramme de cas d'utilisation associé à un patient.

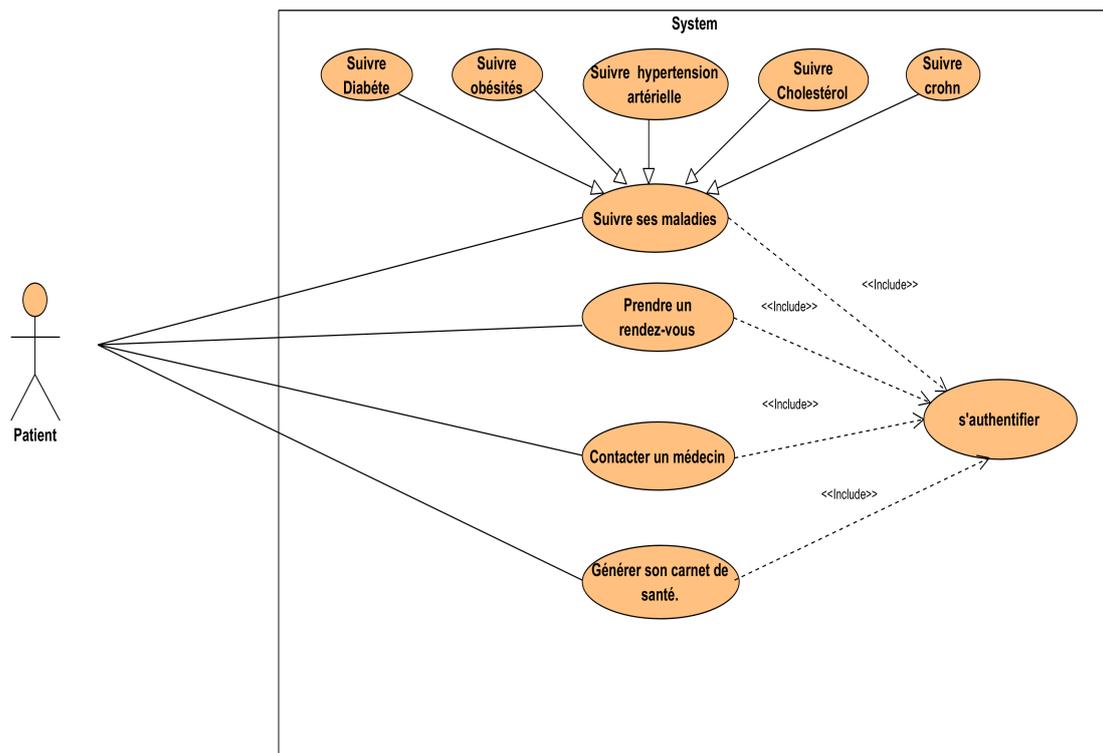


FIGURE 3.6 – Diagramme de cas d'utilisation associé à un Patient.

Détail du cas d'utilisation suivre ses maladies associé à un patient.

Notre application offre la possibilité de suivre plusieurs maladies.

la figure 3.7 Représente le cas d'utilisation " suivre le diabète " qui est un exemple du cas d'utilisation principale " suivre ses maladies "

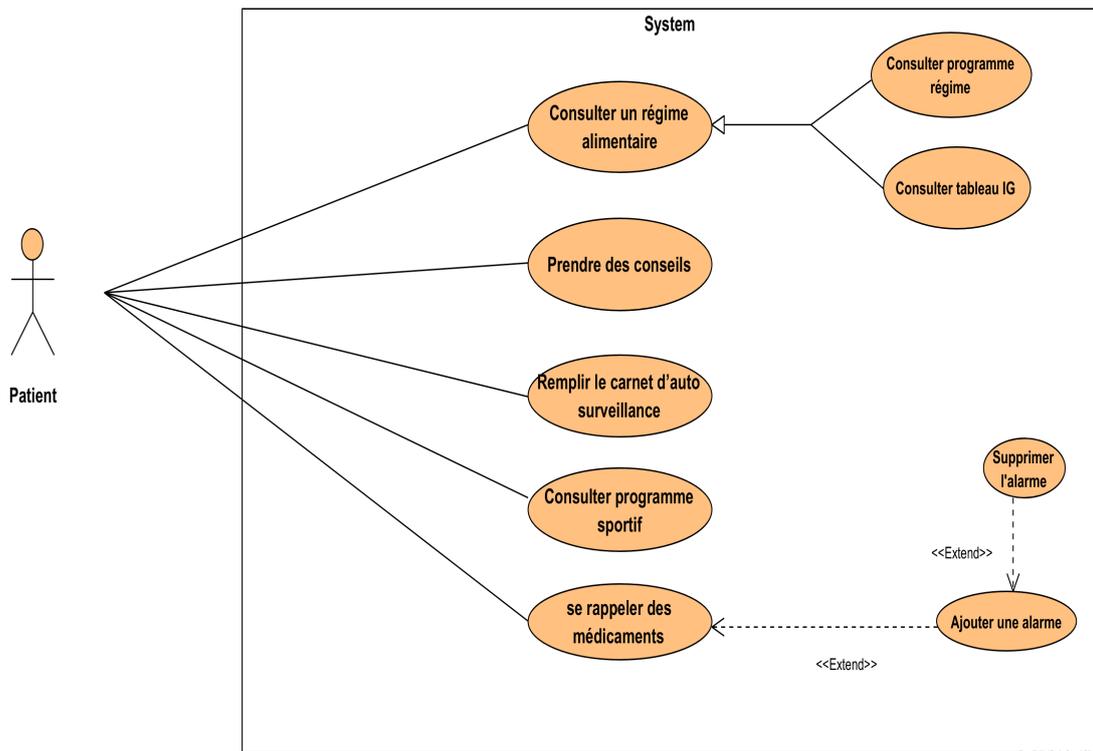


FIGURE 3.7 – Diagramme de cas d'utilisation " Suivre le diabète ".

3. **Médecin** : Pour qu'il puisse profiter de ses fonctionnalités il doit avoir un compte, et après l'authentification il peut :
- Contacter un patient ;
 - Gérer les rendez-vous ;

la figure 3.8 est une représentation d'un cas d'utilisation associé à un médecin .

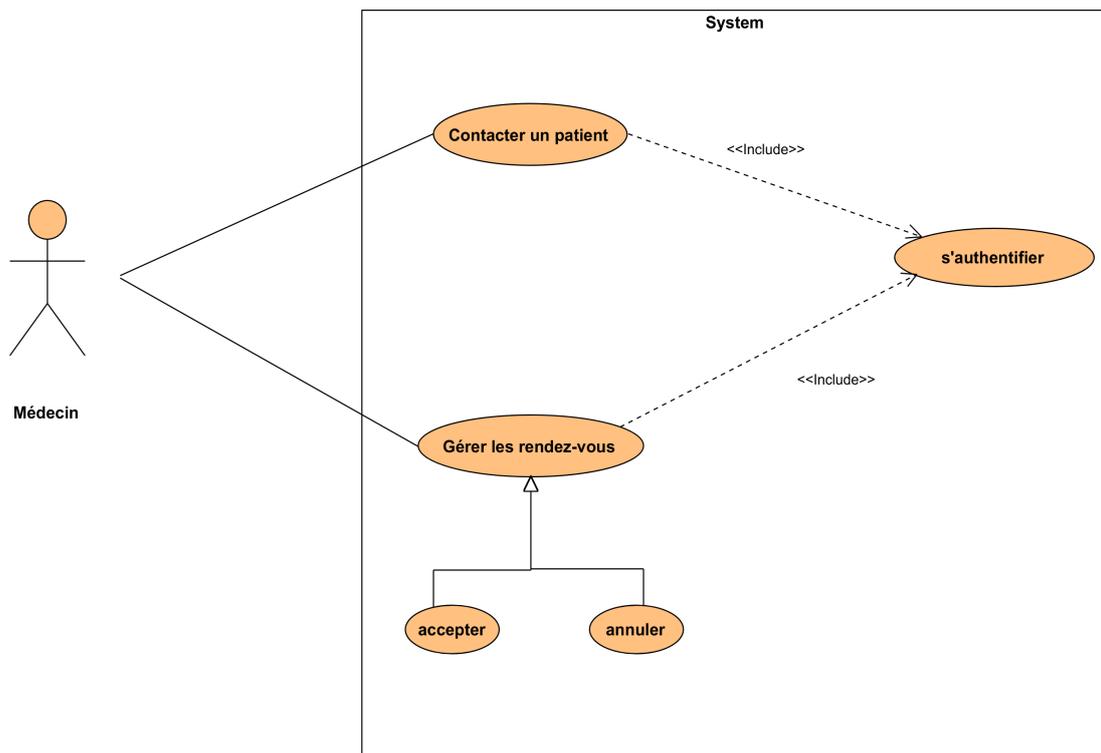


FIGURE 3.8 – Diagramme de cas d'utilisation associé à un médecin.

4. **L'administrateur** : Qui administre, il peut en effet, après s'être authentifié :

- Gérer la liste des hôpitaux : l'administrateur est le responsable de la validité des informations dans la liste des hôpitaux donc il est chargé d'ajouter , modifier , supprimer les informations.
- Gérer la liste des médecins : l'administrateur est le responsable de la validité des informations dans la liste des médecins donc il est chargé d'ajouter , modifier , supprimer les informations.
- Gérer la liste des donneurs de sang : l'administrateur est le responsable de la validité des informations dans la liste des donneurs de sang donc il est chargé d'ajouter , modifier , supprimer les informations.
- Gérer la liste des pharmacies : l'administrateur est le responsable de la validité des informations dans la liste des pharmacies donc il est chargé d'ajouter , modifier , supprimer les informations.
- Gérer la liste des ambulances : l'administrateur est le responsable de la validité des informations dans la liste des ambulances donc il est chargé d'ajouter modifier , supprimer les informations.
- voir et supprimer les avis ;

la figure 3.9 Montre le diagramme de cas d'utilisation associée à un administrateur.

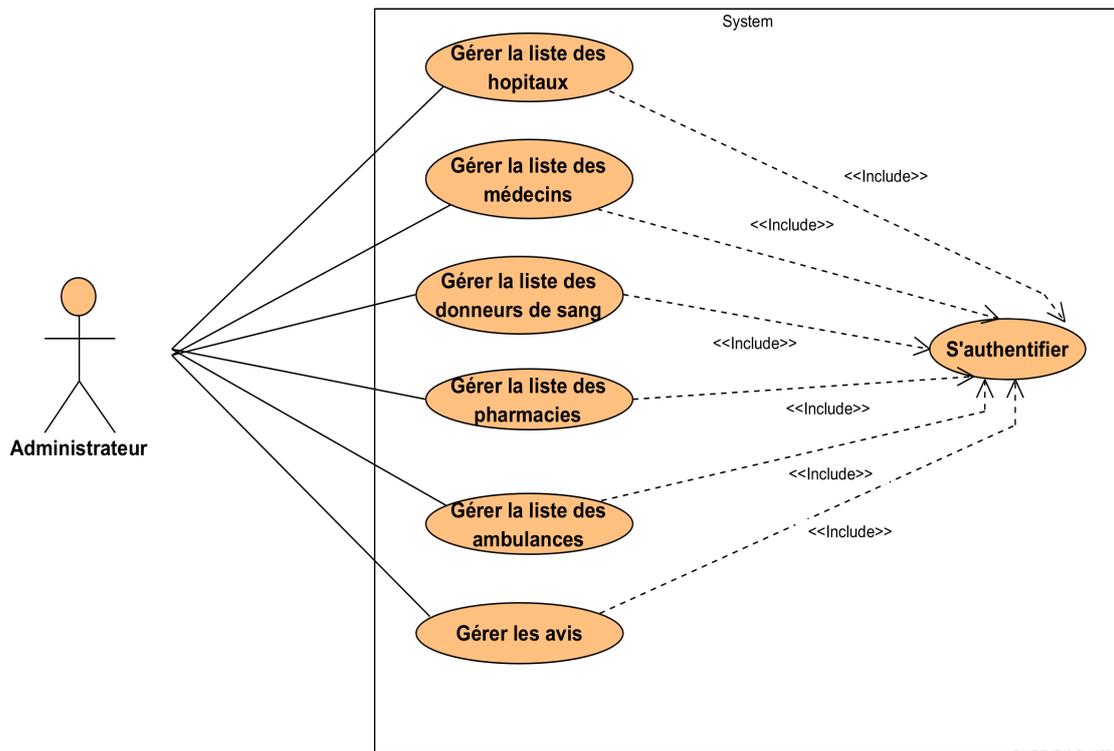


FIGURE 3.9 – Diagramme de cas d'utilisation associé à un administrateur.

5. Diagramme de cas d'utilisation générale de notre système

Nous rassemblons tous les cas d'utilisation précédemment identifiés, dans un diagramme général illustré dans la figure 3.10.

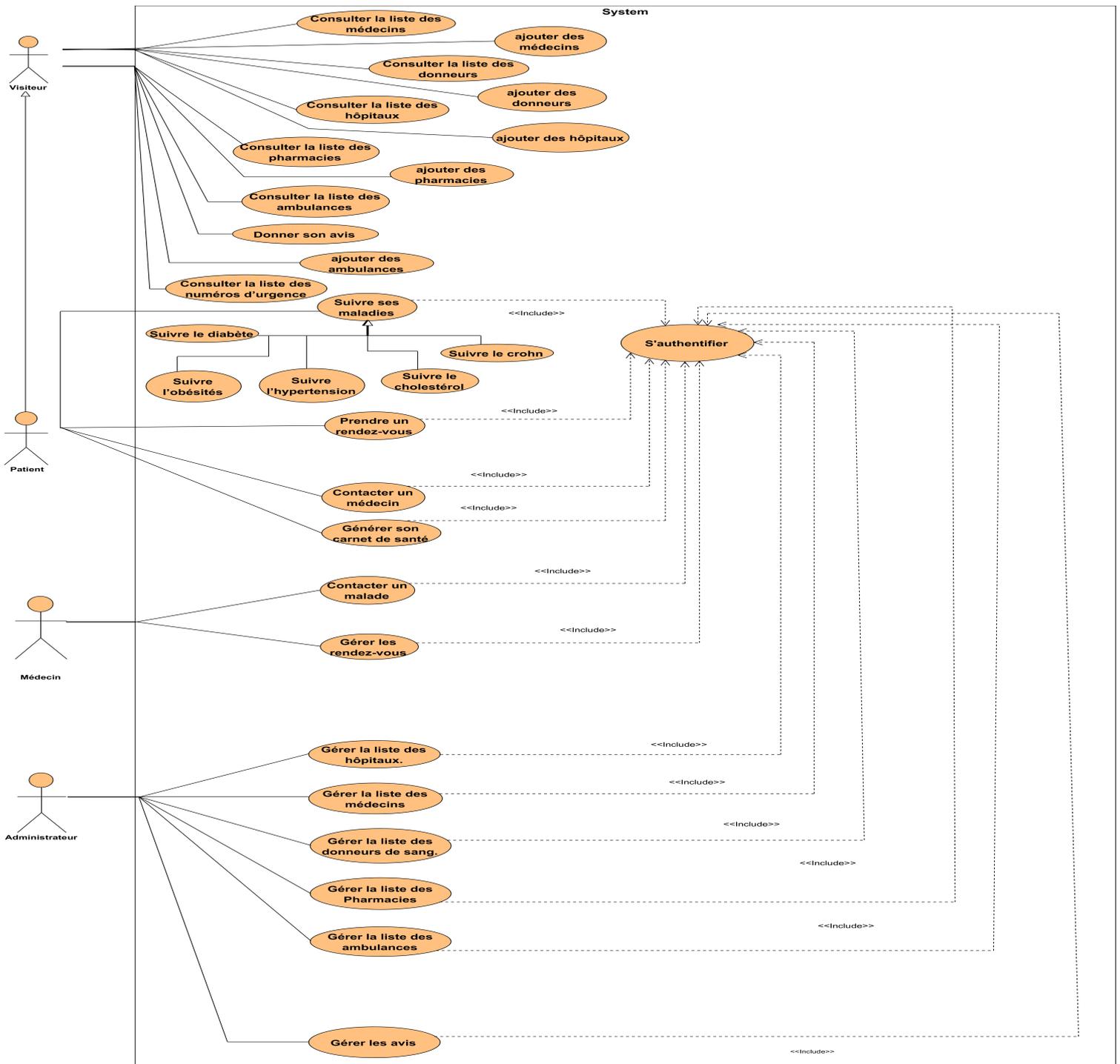


FIGURE 3.10 – Diagramme de cas d'utilisation générale de notre système.

3.2.4 Description textuelle des cas d'utilisation

1. Cas d'utilisation s'authentifier

Description	Vérification d'identité et contrôle d'accès
Pré-condition	Avoir un compte
Acteur	Patient, Médecin, Administrateur
Scénario nominal	(a) L'utilisateur accède à l'application (b) L'utilisateur demande l'interface d'authentification (c) Le système affiche l'interface (d) L'utilisateur remplit le formulaire de connexion (saisie du login et son mot de passe) (e) Le système donne l'accès à l'interface suivante
Scénario alternatif	Si le formulaire est incomplet ou les informations entrantes par l'utilisateur sont incorrecte le système réaffiche l'interface d'accueil

TABLE 3.2 – Description textuelle du cas d'utilisation s'authentifier.

2. Cas d'utilisation consulter le programme régime

Description	Programme régime pour une alimentation bien équilibrée et contrôlée en calories.
Pré-condition	Authentification
Acteur	Patient
Scénarios nominal	(a) Le patient demande d'accéder à l'interface programme régime (b) Le système affiche l'interface
Scénarios alternatif	Aucun.

TABLE 3.3 – Description textuelle du cas d'utilisation " consulter le programme régime ".

3. Cas d'utilisation remplir le carnet d'auto surveillance glycémique

Description	Suivi de la glycémie des diabétiques pour noter et contrôler les différentes prises quotidiennes
Pré-condition	Authentification
Acteur	Patient
Scénarios nominal	<ul style="list-style-type: none"> (a) Le patient accède à l'interface du carnet (b) Le patient demande de remplir le carnet d'ASG (c) Le système affiche le formulaire à remplir (d) Le patient doit lire comment fonctionne le carnet (e) Le patient mesure sa glycémie via un glucomètre avant et après les repas, et note les résultats (f) Le patient note la quantité des médicaments qu'il a pris (insuline et/ou médicament) (g) Le patient doit remplir toutes les informations nécessaires et les valider (h) Le système envoie un message de confirmation
Scénarios alternatif	Aucun.

TABLE 3.4 – Description textuelle du cas d'utilisation " remplir le carnet d'auto surveillance glycémique ".

4. Cas d'utilisation se rappeler des médicaments

Description	Un service qui permet de rappeler les différentes prises quotidiennes (médicament /insuline)
Pré-condition	Authentification
Acteur	Patient
Scénarios nominal	<ul style="list-style-type: none"> (a) Le patient demande l'interface du rappel des médicaments . (b) Le système affiche le formulaire d'ajout . (c) Le patient remplit ce formulaire . (d) Le patient remplit les heures et valide le formulaire . (e) Le système déclenche des notifications si le jour j est venu .
Scénarios alternatif	Aucun.

TABLE 3.5 – Description textuelle du cas d'utilisation " se rappeler des médicaments ".

5. Cas d'utilisation prendre des conseils

Description	Un ensemble d'informations et de conseils pour les patients
Pré-condition	Authentification
Acteur	Patient
Scénarios nominal	(a) Le patient accède à l'interface conseils (b) Le patient lit les différents conseils pour bien se renseigner sur sa maladie
Scénarios alternatif	Aucun.

TABLE 3.6 – Description textuelle du cas d'utilisation " prendre des conseils ".

6. Cas d'utilisation calculer la masse corporelle

Description	Calculer la masse corporelle après avoir entré le poids et la taille
Pré-condition	Authentification
Acteur	Patient
Scénarios nominal	(a) Le patient accède à l'interface de l'obésité (b) Le patient choisit la section du calcul de l'IMC (c) Le système renvoie le formulaire (d) Le patient remplit le formulaire avec son poids et sa taille (e) Le système renvoie le résultat de l'IMC
Scénarios alternatif	Aucun.

TABLE 3.7 – Description textuelle du cas d'utilisation " calculer la masse corporelle ".

7. Cas d'utilisation Consulter tableau des catégories de poids

Description	Patient après le calcul de la masse corporelle peut voir sa catégorie de poids
Pré-condition	Authentification
Acteur	Patient
Scénarios nominal	<ul style="list-style-type: none"> (a) Le patient accède à l'interface de l'obésité (b) Le patient choisit la section du calcul de IMC (c) Le système renvoie le formulaire (d) Le patient remplit le formulaire avec son poids et sa taille (e) Le système renvoie le résultat de IMC (f) Le patient choisit la section de catégorie de poids (g) Le système renvoie l'interface. (h) Le patient lit le tableau des catégories.
Scénarios alternatif	Aucun.

TABLE 3.8 – Description textuelle du cas d'utilisation " consulter tableau des catégories de poids ".

8. Cas d'utilisation conseil et horaire de mesure de la tension

Description	Patient peut consulter les conseils et les horaires de mesure de la tension
Pré-condition	Authentification
Acteur	Patient
Scénarios nominal	<ul style="list-style-type: none"> (a) Le patient accède à l'interface d'hypertension (b) Le patient choisit la section conseils et horaires de mesure de la tension (c) Le système renvoie l'interface (d) Le patient lit les conseils et consulte les horaires.
Scénarios alternatif	Aucun.

TABLE 3.9 – Description textuelle du cas d'utilisation " conseil et horaire de mesure de la tension ".

9. Cas d'utilisation prendre un rendez-vous

Description	Le Patient peut prendre un rendez-vous chez son médecin
Pré-condition	Authentification
Acteur	Patient
Scénarios nominal	(a) Le patient accède à l'interface pour choisir son médecin (b) Le système renvoie l'interface (c) Le patient choisit la section prendre un rendez-vous. (d) Le système renvoie le formulaire (e) Le patient remplit le formulaire (f) Le système renvoie un message de confirmation
Scénarios alternatif	Aucun.

TABLE 3.10 – Description textuelle du cas d'utilisation " prendre un rendez-vous ".

10. Cas d'utilisation contacter son médecin et contacter son patient

Description	Le patient et le médecin peuvent se mettre en contact
Pré-condition	Authentification
Acteur	Patient, Médecin
Scénarios nominal	(a) L'utilisateur accède à l'interface pour choisir son médecin ou son patient (b) Système renvoie l'interface (c) L'utilisateur remplit le formulaire de contact (d) Système renvoie un message de confirmation
Scénarios alternatif	Aucun.

TABLE 3.11 – Description textuelle du cas d'utilisation " contacter son médecin et contacter son malade ".

11. Cas d'utilisation consulter la liste des médecins

Description	Le visiteur peut consulter l'ensemble des informations sur les médecins
Pré-condition	Aucune
Acteur	Visiteur , Patient
Scénarios nominal	(a) L'utilisateur accède à l'interface liste des médecins (b) L'utilisateur choisit la spécialité du médecin (c) L'utilisateur lit les différentes informations sur le médecin
Scénarios alternatif	Aucun.

TABLE 3.12 – Description textuelle du cas d'utilisation " consulter la liste des médecins ".

12. Cas d'utilisation ajouter des médecins à la liste

Description	Le visiteur peut ajouter des médecins
Pré-condition	Aucune
Acteur	Visiteur
Scénarios nominal	(a) Visiteur accède à l'interface liste des médecins (b) Visiteur choisit la section ajouter un médecin (c) Système renvoie le formulaire d'ajout (d) Visiteur remplit le formulaire d'ajout (e) Système renvoie un message de confirmation
Scénarios alternatif	Aucun.

TABLE 3.13 – Description textuelle du cas d'utilisation " ajouter des médecins ".

13. Cas d'utilisation gérer les rendez-vous

Description	Médecin peut gérer tous les rendez-vous demandés par les patients
Pré-condition	Authentification
Acteur	Médecin
Scénarios nominal	(a) Médecin demande l'interface des rendez-vous (b) Système renvoie l'interface (c) Médecin peut accepter ou annuler les rendez-vous des patients
Scénarios alternatif	Aucun.

TABLE 3.14 – Description textuelle du cas d'utilisation " gérer les rendez-vous ".

14. Cas d'utilisation gérer la liste des hôpitaux

Description	Administrateur peut ajouter, supprimer, modifier la liste des hôpitaux
Pré-condition	Authentification
Acteur	Administrateur
Scénarios nominal	(a) Administrateur demande d'accéder à l'interface (b) Système renvoie l'interface (c) Administrateur choisit la section de la liste des hôpitaux (d) Administrateur choisit d'ajouter ou de supprimer ou de modifier un hôpital (e) Système affiche le formulaire (f) Administrateur remplit le formulaire (g) Système affiche un message de confirmation
Scénarios alternatif	Aucun.

TABLE 3.15 – Description textuelle du cas d'utilisation " gérer la liste des hôpitaux ".

3.2.5 Diagramme de séquence

Nous utilisons le terme de diagramme de séquence pour souligner le fait de considérer le système informatique comme une boîte noire. Le comportement du système est décrit vu de l'extérieur, sans préciser comment il le réalisera.

L'objectif des diagrammes de séquence est de représenter les interactions entre les objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés [20].

1. Formalisme et concepts de base

Dans un diagramme de séquence, les objets et les acteurs sont énumérés en colonnes avec leurs lignes de vie verticales indiquant la durée de vie de l'objet [21].

Un objet est une instance de classe rangé horizontalement et représentée par un rectangle.

Une ligne de vie (LifeLine) représente l'ensemble des opérations exécutées par un objet. La notation utilisée pour une ligne de vie est une ligne pointillée verticale partant de l'objet.

Une activation est modélisée par des boîtes rectangulaires sur la ligne de vie. Elles indiquent quand l'objet effectue une action.

Un message est modélisé par une flèche horizontale entre les activations. Il indique les communications entre les objets.

Dans un diagramme de séquence, deux types de messages peuvent être distingués [22] :

- **Message synchrone** : L'émetteur reste en attente d'une réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message .
- **Message asynchrone** : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. Une flèche avec une extrémité qui n'est pas pleine symbolise ce type de message .

La figure 3.9 illustre le formalisme général suivi pour représenter un diagramme de séquence . Pour enrichir ce diagramme, nous ferons apparaître les renvois aux enchaînements d'erreur au moyen de notes.

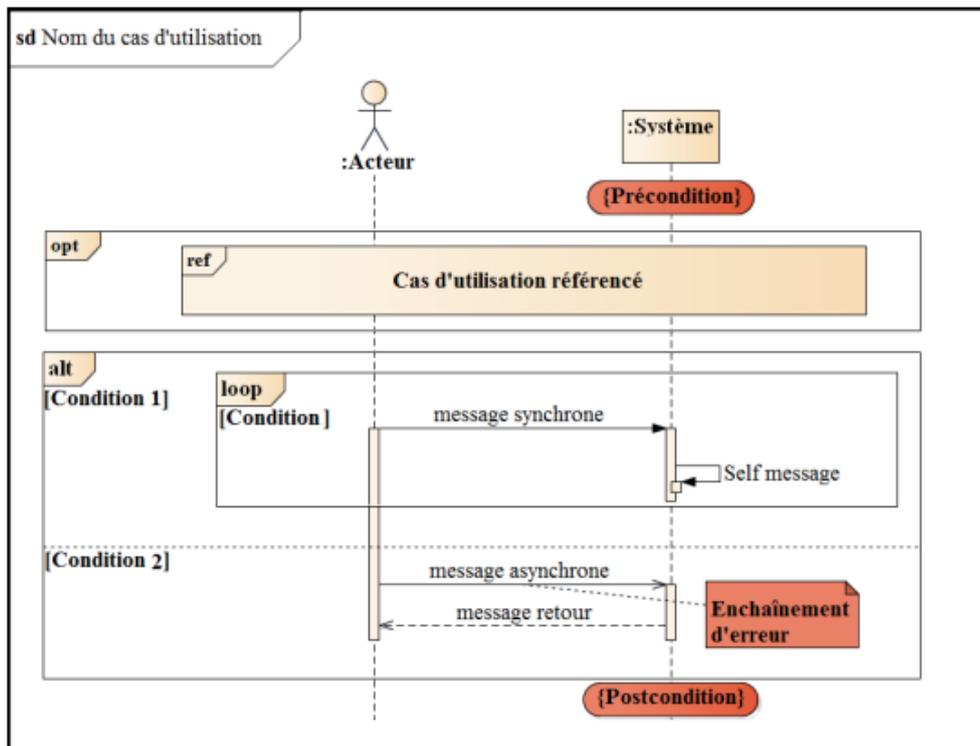


Figure 3.9 :Formalisme général d'un diagramme de séquence.

2. Fragment d'interaction

Dans un diagramme de séquence, il est possible de distinguer des sous-ensembles d'interactions qui constituent des fragments.

Un fragment d'interaction correspond à un ensemble d'interactions auxquelles on applique un opérateur. On le représente globalement, comme un diagramme de séquence, dans un rectangle avec indication dans le coin gauche du nom de l'opérateur [22].

Treize (13) opérateurs ont été définis dans UML : alt, opt, loop, par, strict/weak, break, ignore/consider, critical, negative, assertion et ref. Nous décrivons dans ce qui suit les opérateurs utilisés pour l'analyse de notre projet : alt, opt, loop et ref [22].

- **alt** : L'opérateur alternative ou alt correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il permet aussi d'utiliser les clauses de type sinon et se représente dans un fragment possédant au moins deux parties séparées par des pointillés.
- **opt** : L'opérateur optional ou opt correspond à une instruction de test sans alternative. Il se représente dans un fragment possédant une seule partie.
- **loop** : L'opérateur loop correspond à une instruction de boucle qui permet d'exécuter une séquence d'interactions tant qu'une condition est satisfaite.
- **ref** : L'opérateur ref permet d'appeler une séquence d'interactions décrite par ailleurs, constituant ainsi une sorte de sous-diagramme de séquence.

3. Diagrammes de séquence des cas d'utilisation

Dans ce qui suit, nous représentons le diagramme de séquence système d'un scénario représentatif de chacun des cas d'utilisation décrits précédemment.

Diagramme de séquence " s'authentifier "

La figure 3.11 représente le diagramme de séquence d'authentification , en expliquant le scénario qui se déroule entre l'utilisateur , le système et la base de données

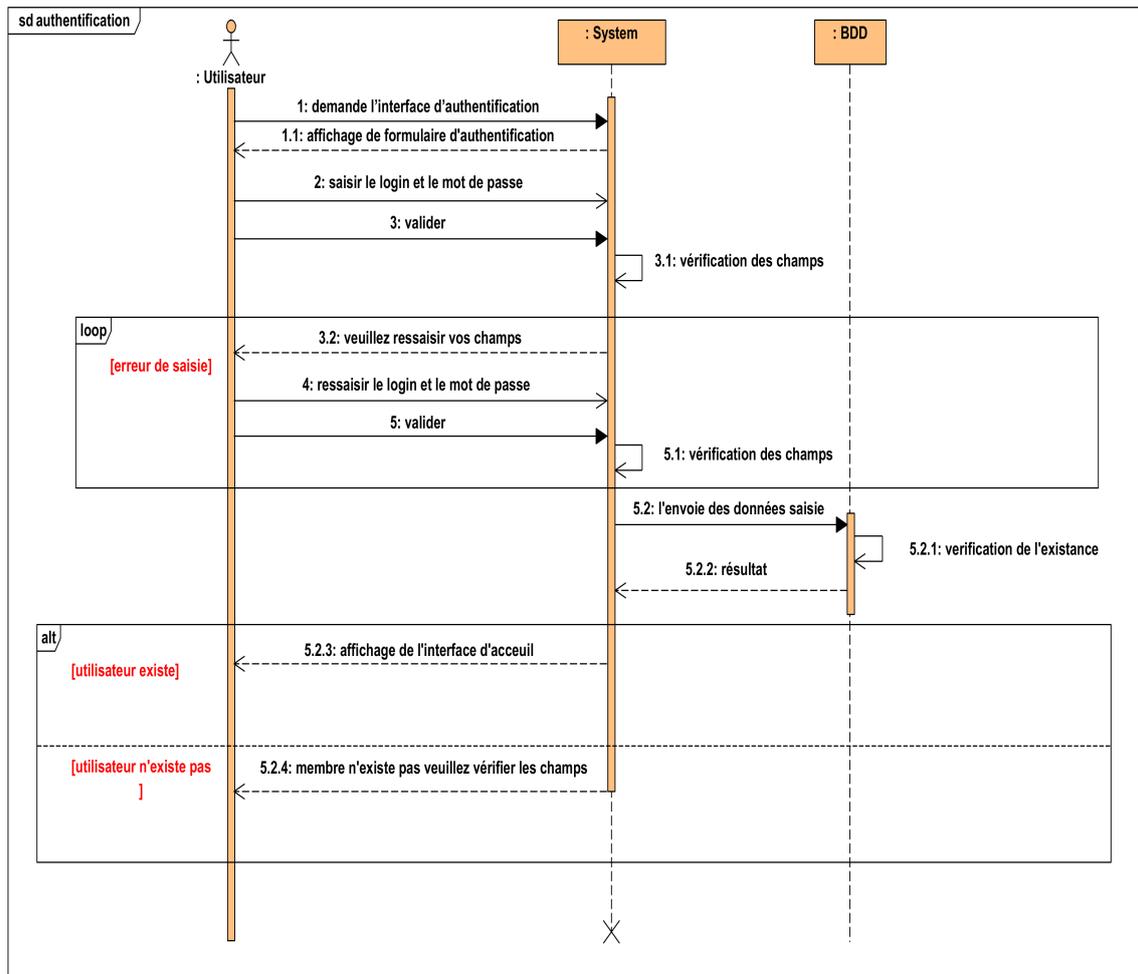


FIGURE 3.11 – Diagramme de séquence " s'authentifier ".

Diagramme de séquence " suivre un régime alimentaire "

La figure 3.12 représente le scénario entre le patient et le système pendant l'accès à l'interface des consultation du régime alimentaire ainsi le tableau d'index glycémie

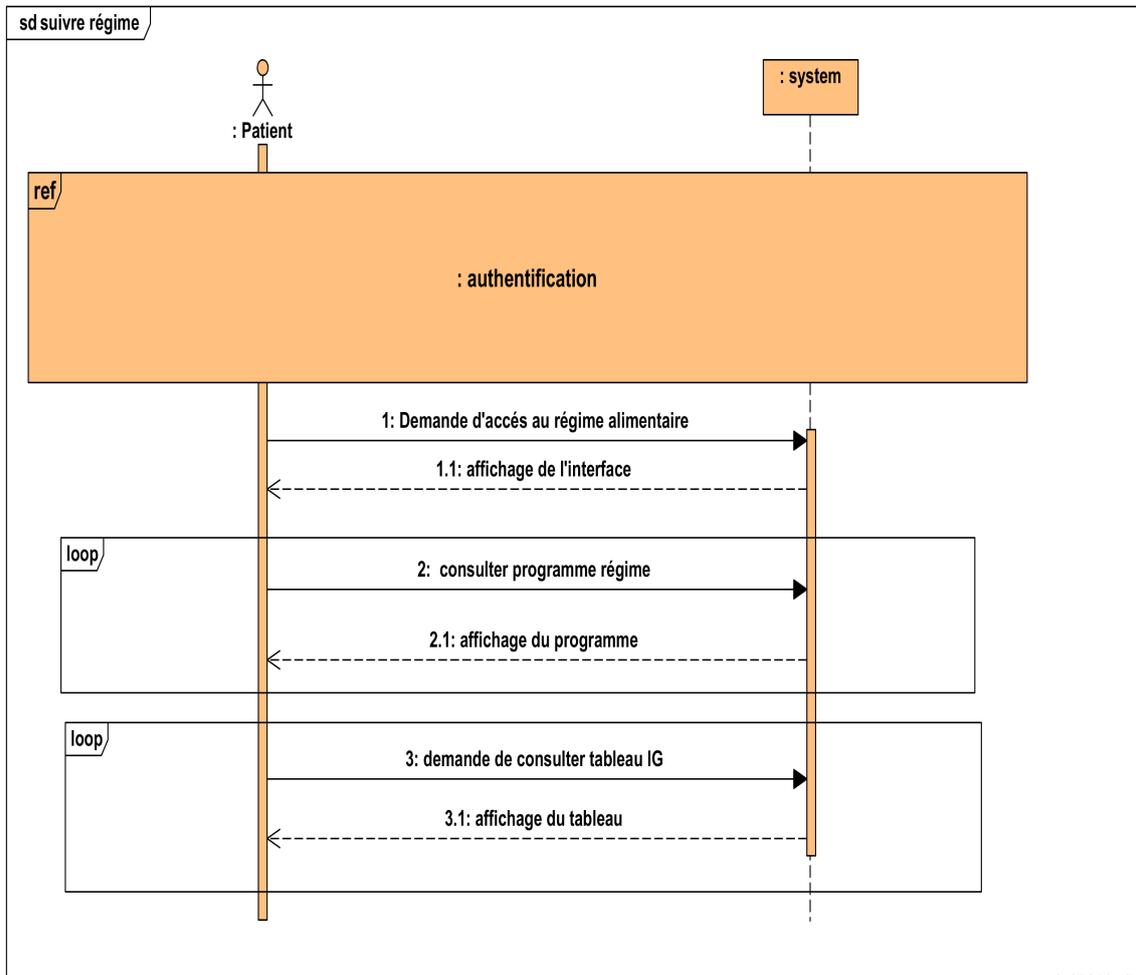


FIGURE 3.12 – Diagramme de séquence " suivre un régime alimentaire ".

Diagramme de séquence " se rappeler des médicaments "

La figure 3.13 démontre le scénario déroulé entre le patient et le système pour déclencher une alarme

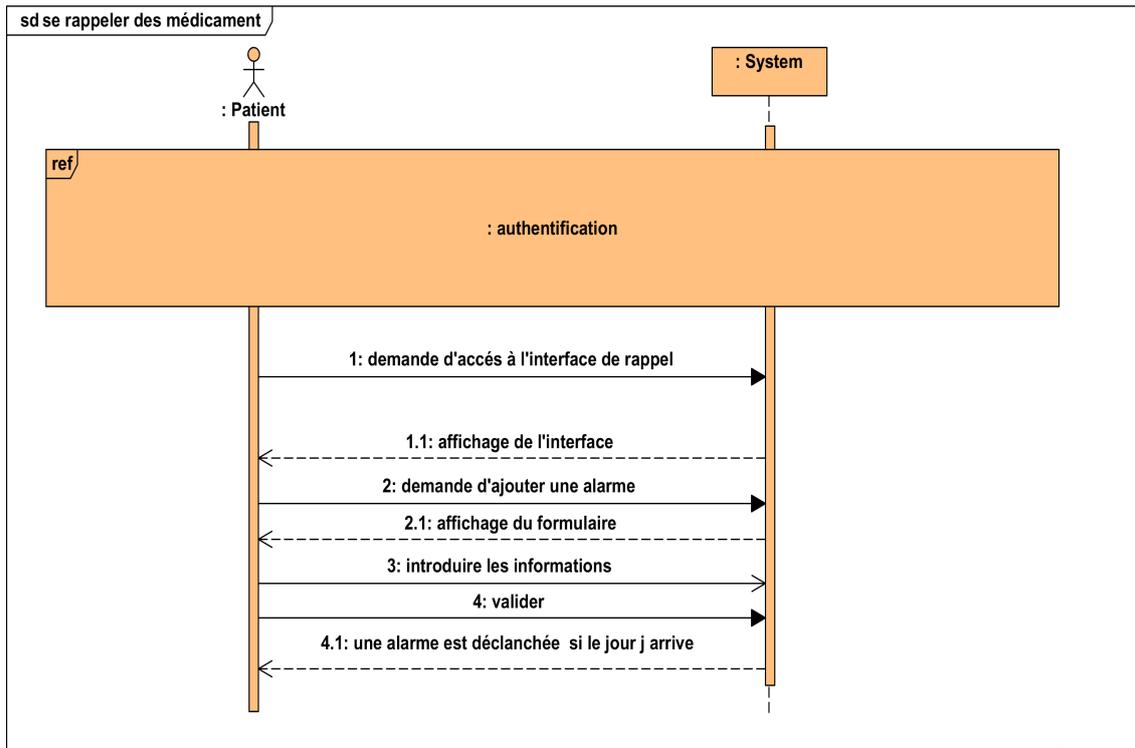


FIGURE 3.13 – Diagramme de séquence " se rappeler des médicaments ".

Diagramme de séquence " prendre un rendez-vous ".

La figure 3.14 représente le diagramme de séquence prendre un rendez-vous par un patient

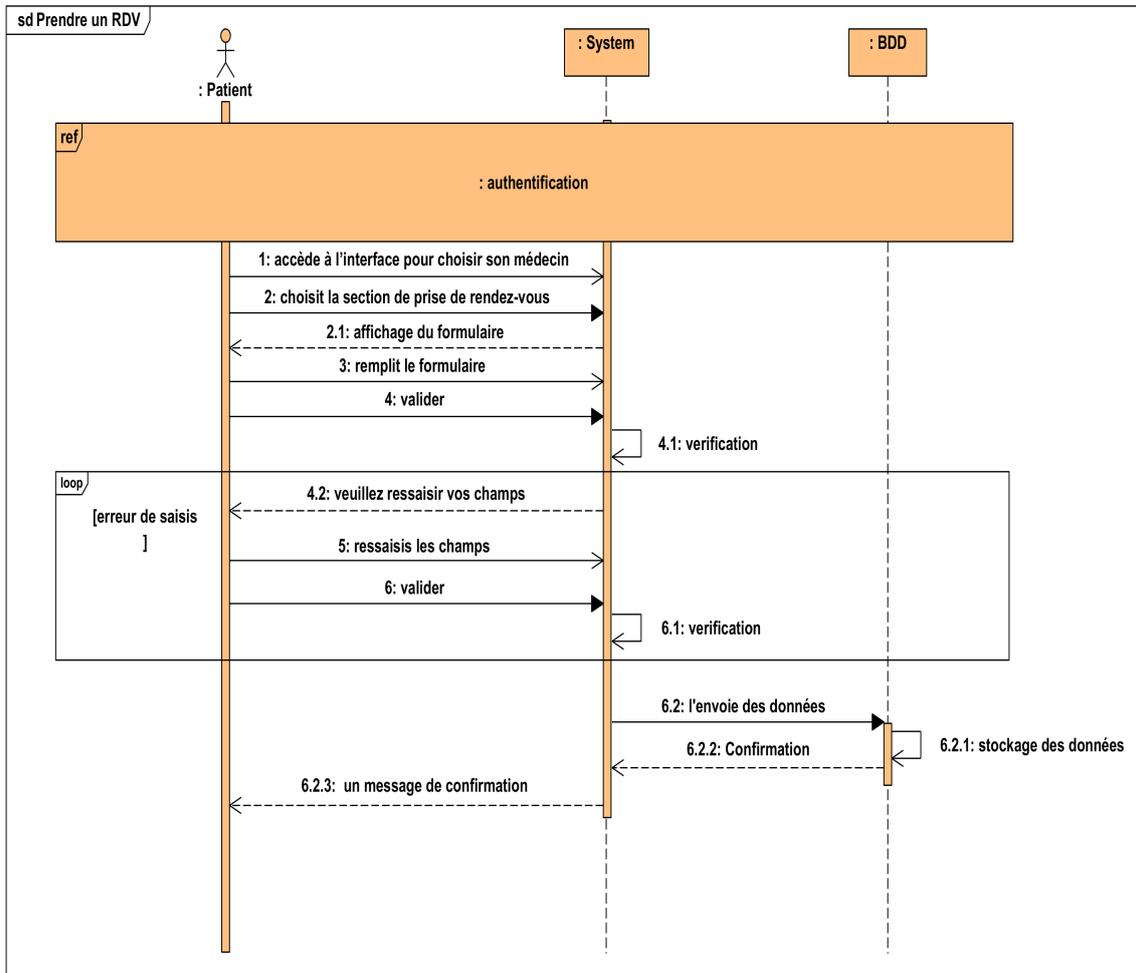


FIGURE 3.14 – Diagramme de séquence " prendre un rendez-vous ".

Diagramme de séquence " ajouter des médecins ".

La figure 3.15 représente le scénario d'un utilisateur (visiteur | patient) qui demande l'ajout d'un médecin au guide des médecins

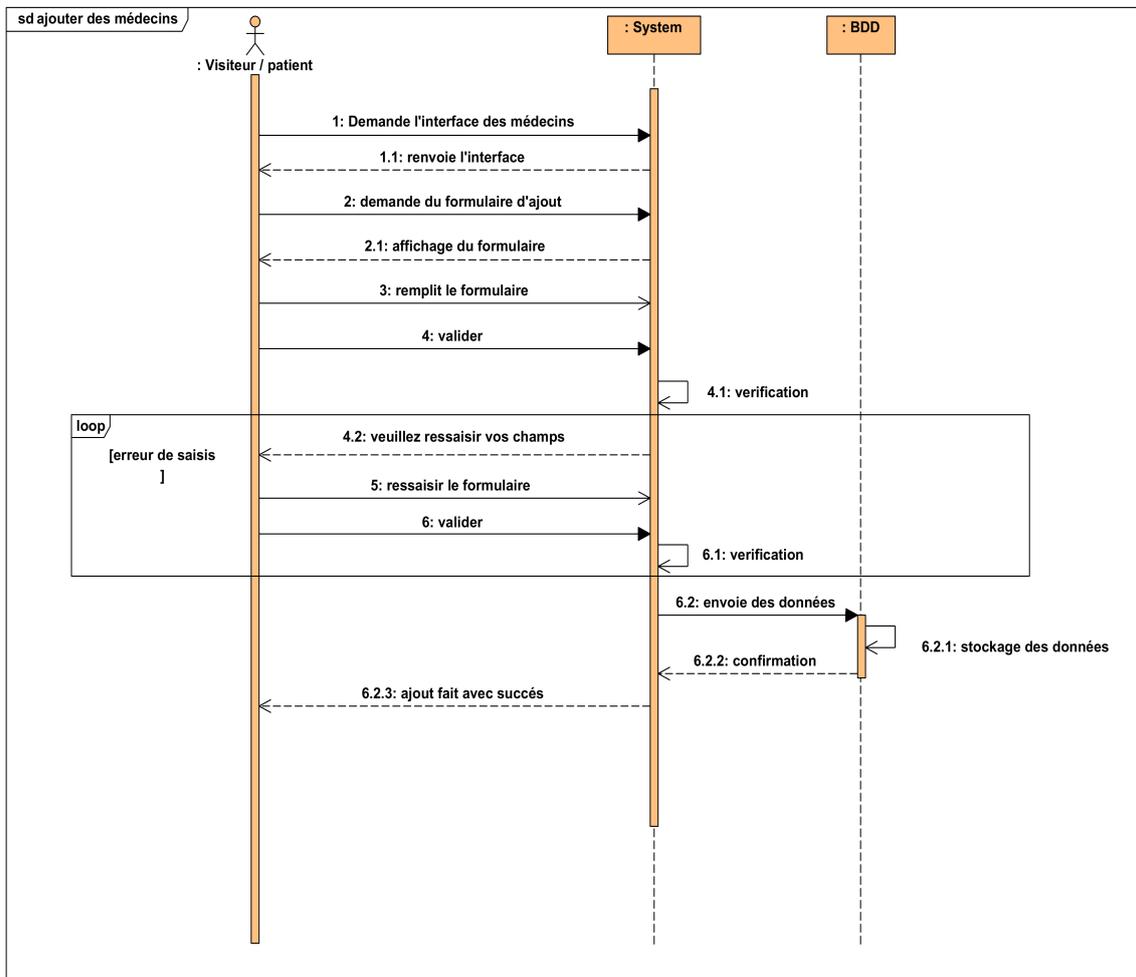


FIGURE 3.15 – Diagramme de séquence " ajouter des médecins ".

Diagramme de séquence gérer la liste des hôpitaux

La figure 3.16 illustre le scénario lorsque un administrateur (ajoute / modifie / supprime) un hopital depuis la liste .

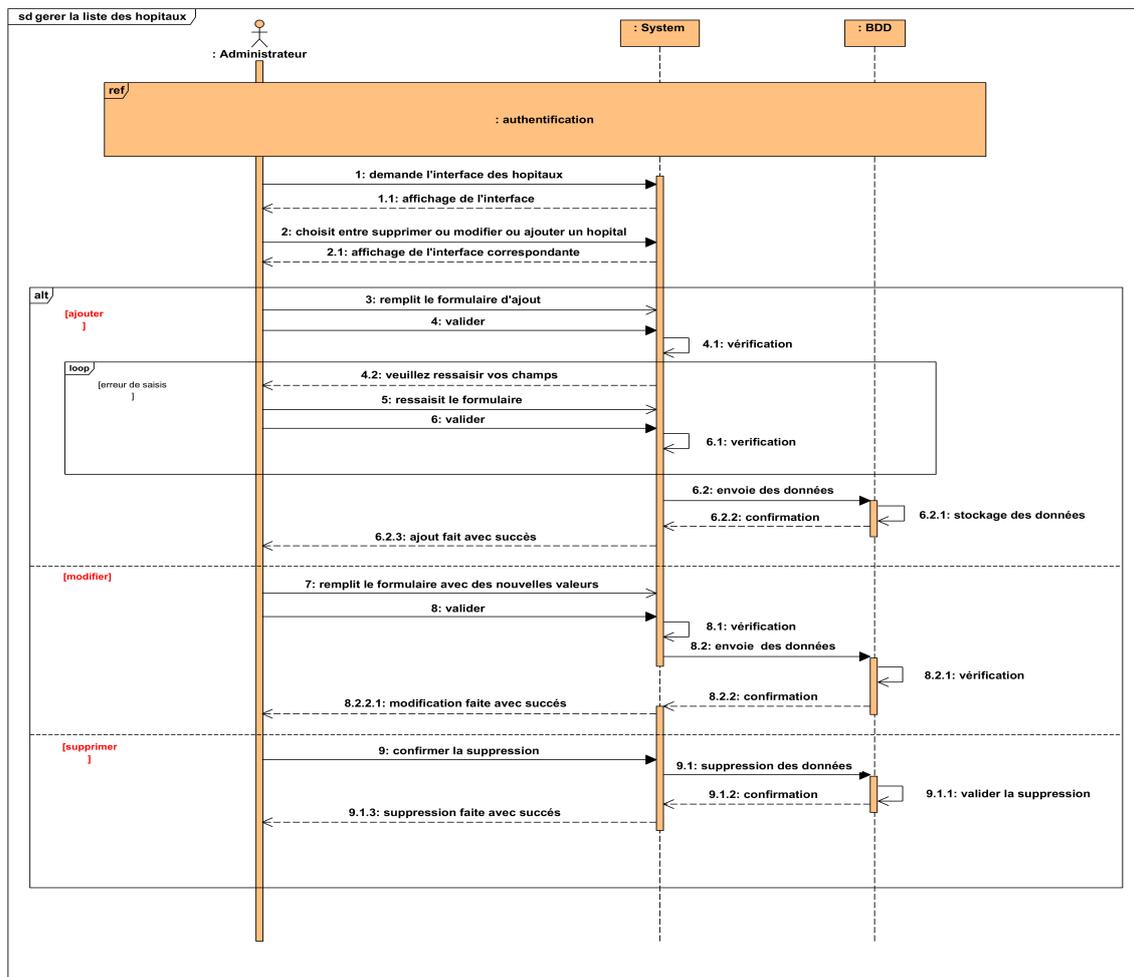


FIGURE 3.16 – Diagramme de séquence "gérer la liste des hôpitaux".

3.2.6 Diagramme de classe

Le diagramme de classes permet de définir quelles seront les composantes du système final. Il représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements, elles sont reliées entre elles par des associations.

La figure 3.17 illustre le diagramme de classes qui correspond à notre environnement.

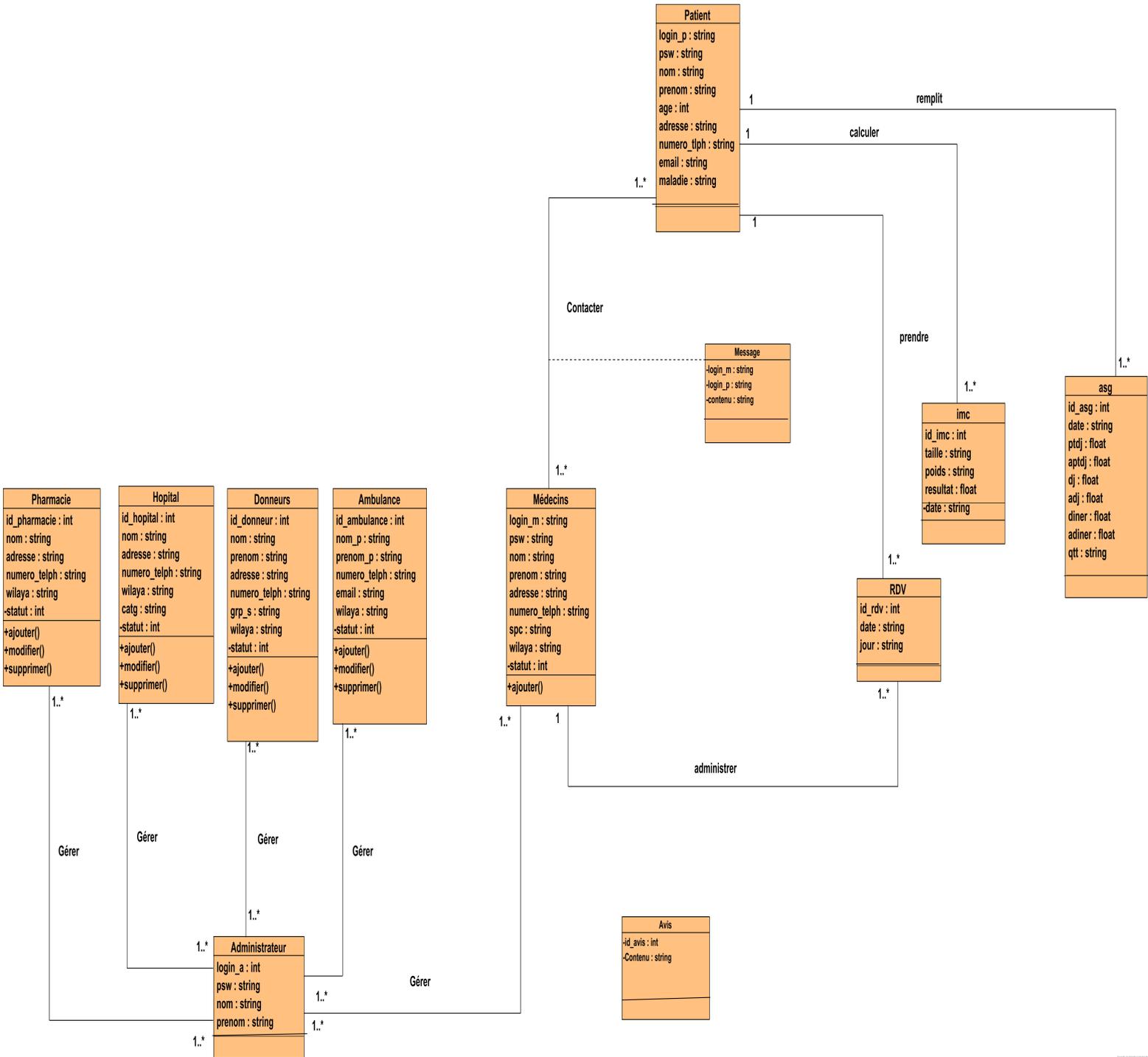


FIGURE 3.17 – Diagramme de classe.

3.2.7 Modèle relationnel

Règles de dérivation du modèle relationnel à partir d'un modèle de classes :

Dans cette section, nous procédons à la construction pas à pas du modèle relationnel implémentant la base de données de l'application à partir du modèle de classes obtenues. Les règles utilisées pour le passage sont [23] :

1. Règle 1 : Transformation des classes

Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant. Si aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire.

2. Règle 2 : Transformation de l'héritage

Trois décompositions sont possibles pour traduire la relation d'héritage en fonction des contraintes existantes :

— **Décomposition par distinction :**

Il faut transformer chaque sous-classe en une relation. La clé primaire de la surclasse, migre dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère.

— **Décomposition descendante (push-down) :**

S'il existe une contrainte de totalité ou de partition sur l'association d'héritage, il est possible de ne pas traduire la relation issue de la surclasse. Il faut alors faire migrer tous ses attributs dans la (les) relation(s) issue(s) de la (des) sous-classe(s).

— **Décomposition ascendante (push-up) :**

Il faut supprimer la (les) relation(s) issue(s) de la (des) sous-classe(s) et faire migrer les attributs dans la relation issue de la surclasse.

3. Règle 3 : Association un-à-plusieurs

Cette règle consiste à ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

4. Règle 4 : Associations plusieurs-à-plusieurs ou classes-associations

L'association devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectées à l'association. Les attributs de l'association doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

En appliquant ces règles à notre modèle, nous obtenons les relations suivantes :

- Administrateur (login-a , psw , nom , prenom)
- Médecins (login-m , psw , nom , prenom , adresse , numero-telph , spc , wilaya , #login-a , statut)
- Patient (login-p ,psw , nom , prenom ,age , adresse ,numero-telph , email , maladie)
- Pharmacie (id-pharmacie , nom , adresse ,numero-telph , wilaya , #login-a , statut)
- Hopital (id-hopital , nom , adresse ,numero-telph , wilaya , catg ,#login-a , statut)
- Donneur (id-donneur , nom ,prenom , adresse ,numero-telph , grp-s, wilaya ,#login-a , statut)
- Ambulance (id-ambulance , nom-p ,prenom-p ,numero-telph , email, wilaya ,#login-a , statut)
- RDV (id-rdv ,#login-m,#login-p, date , jour)
- Avis (id-avis , Contenu)
- Message (login-m ,login-p,Contenu)
- asg (id-asg , jour, ptdj, aptdj, dj, diner, adiner, qtt, #login-p)
- imc (id-imc , taille , poids , resultat , #login-p)

3.2.8 Dictionnaire de données

Attribut	Définir l'attribut	Type	taille
login-a	Nom d'utilisateur d'un administrateur	Chaine de caractères	150
psw	Mot de passe	Chaine de caractères	100
login-m	Nom d'utilisateur d'un médecin	Chaine de caractères	150
spc	Spécialité	Chaine de caractères	100
login-p	Nom d'utilisateur d'un patient	Chaine de caractères	150
id-pharmacie	l'identifiant d'une pharmacie	Entier	150
id-hopital	l'identifiant d'un hopital	Entier	150
id-donneur	l'identifiant d'un donneur	Entier	150
id-ambulance	l'identifiant d'une ambulance	Entier	150
grp-s	Groupe sanguin	Chaine de caractères	150
nom-p	Nom du propriétaire	Chaine de caractères	150
prenom-p	Prénom du propriétaire	Chaine de caractères	150
catg	Catégorie de l'hopital	Chaine de caractères	150
id-rdv	l'identifiant d'un rendez-vous	Entier	150
id-asg	l'identifiant du carnet d'auto surveillance	Entier	150
id-imc	l'identifiant de la masse corporelle	Entier	150
ptdj	petit dejeuner	Entier	150
aptdj	après petit dejeuner	Entier	150
dj	dejeuner	Entier	150
adj	après petit dejeuner	Entier	150
qtt	quantité de médicament	text	

TABLE 3.16 – Dictionnaire de données.

3.3 Conclusion

Ce chapitre clôture la partie conception, il nous a permis de tirer avantage des différents diagrammes UML afin de bien structurer notre application. Dans ce dernier, nous avons décrit de façon détaillée les cas d'utilisation en recensant de façon textuelle toutes les interactions entre les acteurs et le système. Nous avons complété cette description textuelle par une représentation graphique UML : le diagramme de séquence système. Grâce auquel nous sommes parvenus à concevoir le diagramme de classes de conception donnant ainsi une vue plus structurée des éléments qui formeront la base de données liée à notre application.

Chapitre 4

Réalisation de l'environnement

4.1 Introduction

Ce chapitre est consacré à la partie pratique de la réalisation de notre environnement m-santé présenté dans les chapitres précédents. Il comporte une description des outils de développement utilisés ainsi qu'une présentation de nos applications. Le choix de nos outils de développement s'est reposé principalement sur leur gratuité et l'open source. Effectivement, en implémentant une solution qui se base sur des technologies gratuites et open source, nous avons plus de chance d'éviter toutes sortes de problèmes liés aux licences, contrats, etc., réduisant ainsi les coûts.

4.2 Outils de développement

- **Sdk Android** :Le SDK Android (kit de développement logiciel) est un ensemble d'outils de développement utilisés pour développer des applications pour la plateforme Android qui est devenue le plus grand rival d'Apple dans le domaine des smartphones. Le SDK Android inclut les éléments suivants [24] :
 - Bibliothèques requises ;
 - Débogueur ;
 - Un émulateur ;
 - Documentation pertinente pour les interfaces de programme d'application (API) Android ;
 - Exemple de code source ;
 - Tutoriels pour le système d'exploitation Android .
- **Android Studio** :Est l'environnement de développement intégré (IDE) officiel pour le développement d'applications Android. Il est basé sur IntelliJ IDEA, un environnement de développement intégré Java pour les logiciels, intégrant ainsi ses outils d'édition de code et de développement [25].
- **IntelliJ IDEA** : IntelliJ IDEA est un environnement de programmation spécial ou un environnement de développement intégré (IDE) largement destiné à Java. Cet environnement est utilisé notamment pour le développement de programmes. Il est développé par une société appelée JetBrains, qui s'appelaient officiellement IntelliJ. Il est disponible en deux éditions : l'édition communautaire sous licence Apache 2.0 et une édition commerciale connue sous le nom d'Ultimat Edition. Les deux peuvent être utilisés pour créer des logiciels qui peuvent être vendus. Ce qui rend IntelliJ IDEA si différent de ses homologues, c'est sa facilité d'utilisation, sa flexibilité et sa conception solide [26].

- **VS Code** : Visual studio code est un éditeur de code source léger mais puissant qui s'exécute sur bureau et est disponible pour Windows, MacOS et Linux. Il est livré avec un support intégré pour JavaScript, Type Script et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C ++, C, Java, Python, PHP, Go) et des environnements d'exécution (tels que .NET et Unity) [27].
- **AVD** : L'Android Virtual Device, aussi appelé AVD, est un émulateur de terminal sous Android, c'est-à-dire que c'est un logiciel qui se fait passer pour un appareil sous Android sur ordinateur. C'est la raison pour laquelle on n'a pas besoin d'un périphérique sous Android pour développer [28].

La figure 4.1 Montre l'interface de l'émulateur utilisés pendant le développement de notre application mobile.

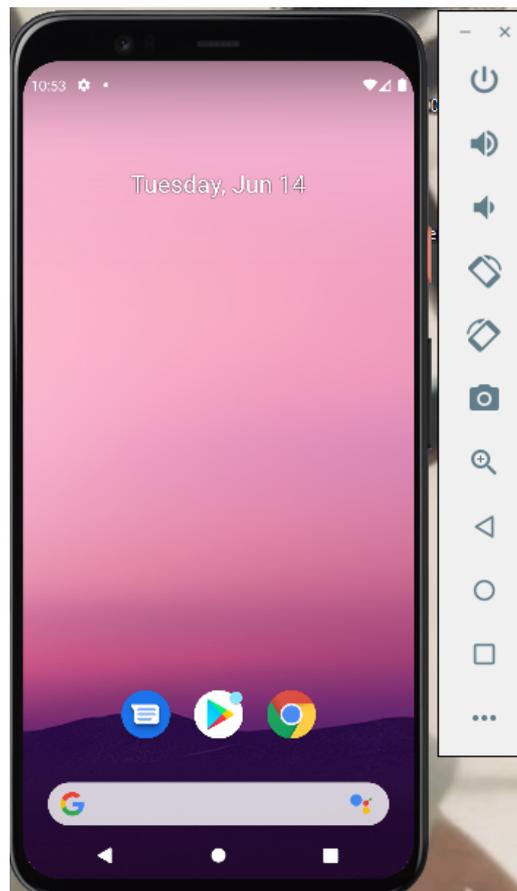


FIGURE 4.1 – Interface AVD.

4.3 Langage de développement

- **Mobile :**

- **Flutter :**

- Fait référence à une boîte à outils d'interface utilisateur open source que les développeurs peuvent utiliser pour les plates-formes mobiles, de bureau et Web via une seule base de code. Les développeurs Flutter peuvent écrire du code une seule fois et l'exécuter sur toutes les plates-formes et tous les appareils [29].

- **Dart :**

- Est un langage de programmation open source, polyvalent et orienté objet avec une syntaxe de style C développé par Google en 2011. Le but de la programmation Dart est de créer une interface utilisateur frontale pour les applications Web et mobiles.

- Il est en cours de développement actif, compilé en code machine natif pour la création d'applications mobiles, inspiré d'autres langages de programmation tels que Java, JavaScript, C et est fortement typé. Étant donné que Dart est un langage compilé, vous ne pouvez pas exécuter votre code directement ; à la place, le compilateur l'analyse et le transfère dans le code machine [29].

- **Web :**

- 1. **Front end :**

- **Html :** HyperText Markup Language est le langage de balisage conçu pour représenter les pages web, il permet grâce à ses différentes balises qui sont le signe inférieur et supérieur d'écrire de l'hypertexte, de structurer sémantiquement une page, de mettre en forme le contenu [30].

- **CSS :** Les feuilles de style en cascade CSS (Cascading Style Sheets), forment un langage informatique décrivant la présentation des documents HTML et XML, ce langage permet la gestion de la mise en forme de la page et donne ainsi à l'utilisateur le contrôle sur le style de sa page web [30].

- **JavaScript :** JavaScript est un langage de programmation de scripts orienté objet introduit dans les pages HTML et principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Il est exécuté sur l'ordinateur de l'internaute par le navigateur contrairement aux langages serveurs s'exécutant sur une machine distante [30].

- **Bootstrap :** Bootstrap est un framework pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et même des plugins jQuery de qualité pour enrichir les pages web [31].

- 2. **Back end :**

- **PHP :** HyperText Preprocessor, est un langage de programmation interprété libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes [32].

4.4 Présentation des interfaces de notre environnement

Dans cette partie du mémoire, nous allons présenter les principales fonctionnalités de notre système qui fera par la description de quelques interfaces.

Notre environnement m-santé réalisé se compose de ces deux applications :

1. Mobile :

— L'interface d'accueil :

La figure 4.2 représente l'interface d'accueil qui sera vu pendant la connexion d'un utilisateur donnée que ça soit dans le but de se connecter ou en tant qu'un visiteur.

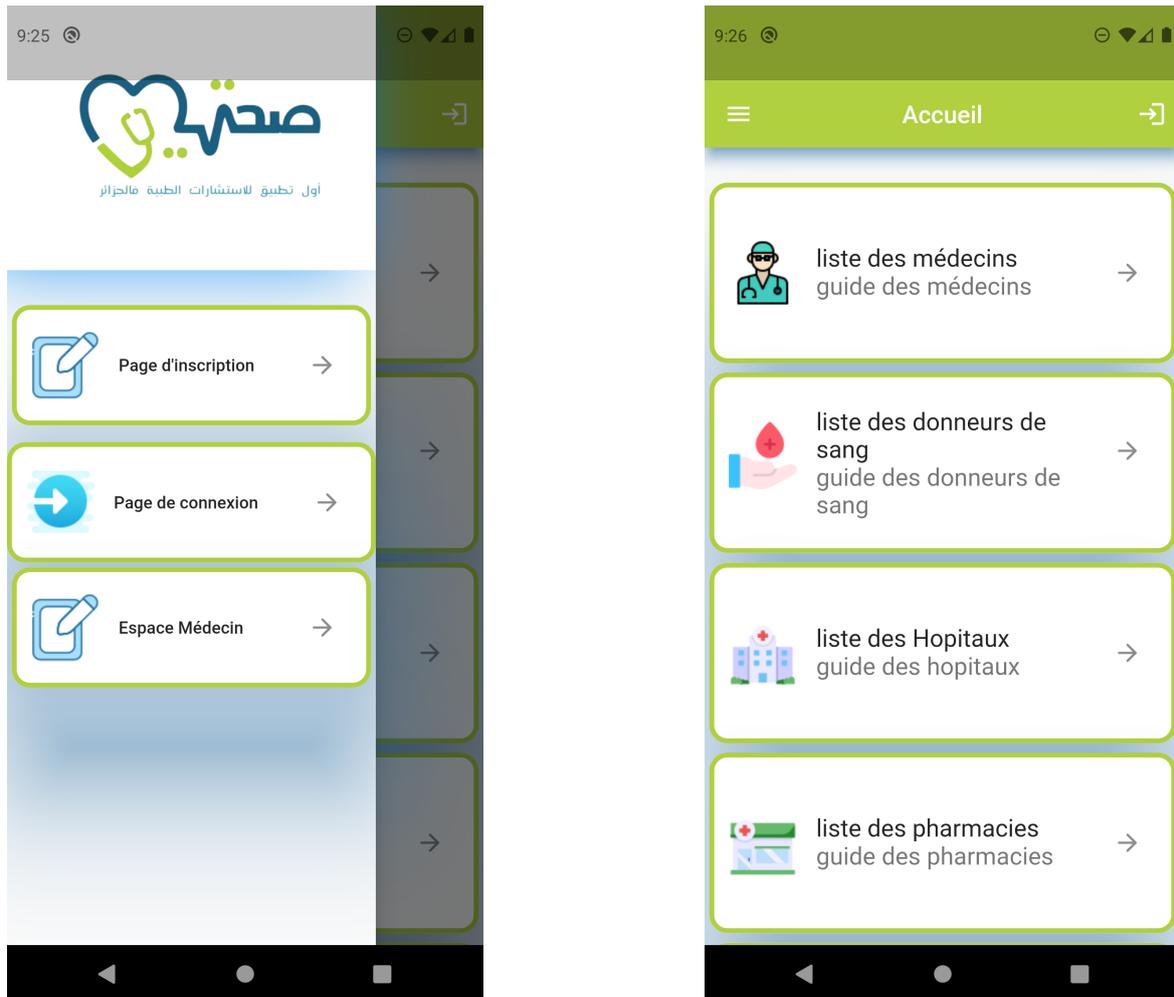


FIGURE 4.2 – L'interface d'accueil

— **L'interface d'affichage des médecins :**

La figure 4.3 illustre l'interface d'affichage des médecins, l'utilisateur après avoir passé par l'interface d'accueil il pourra choisir cette interface afin de rechercher un médecin selon sa spécialité et sa wilaya comme il pourra aussi rajouter à cette liste un médecin donnée en remplissant le formulaire qu'on pourra y accéder en cliquant sur le bouton en haut de la fenêtre.

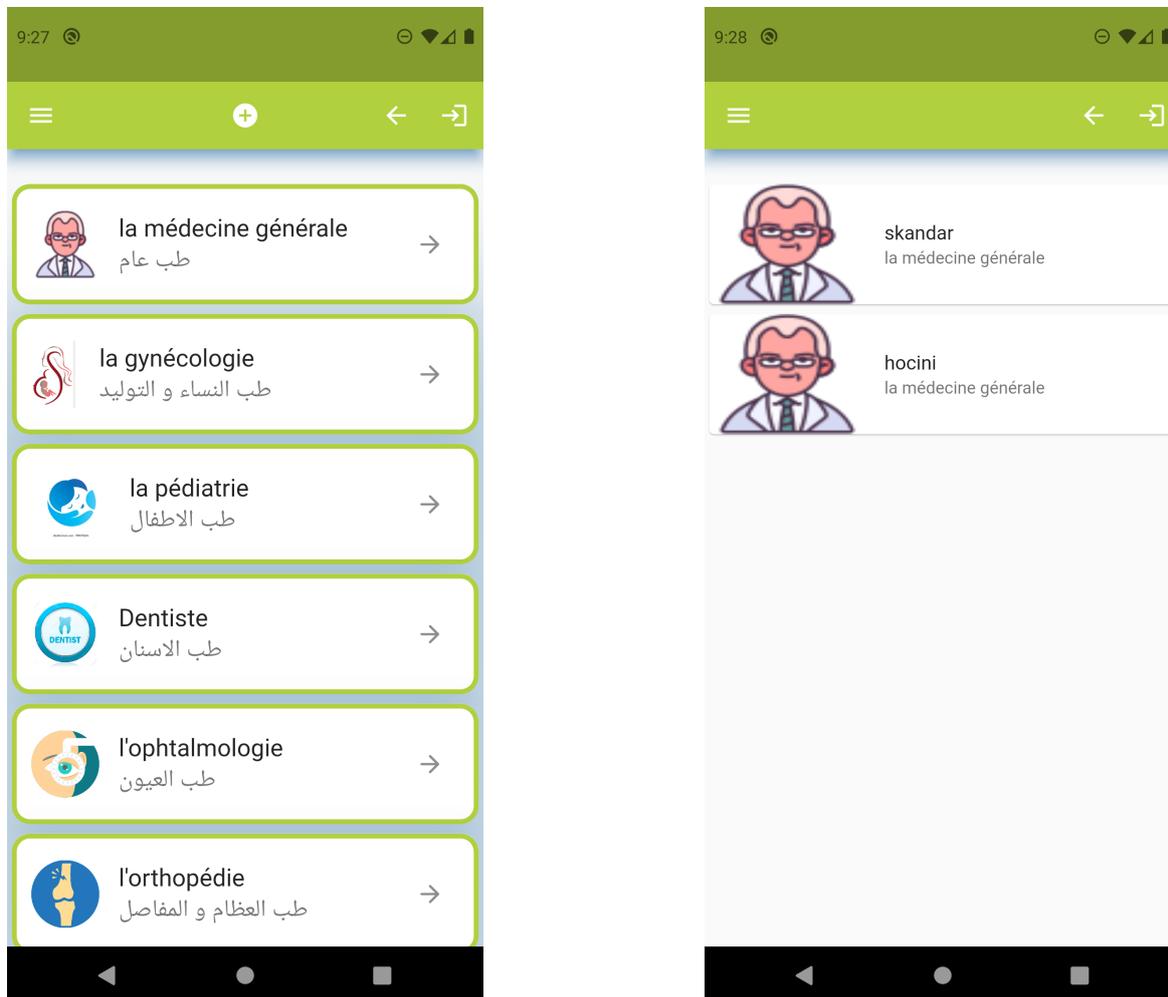


FIGURE 4.3 – L'interface d'affichage des médecins

— **L'interface d'inscription et de connexion :**

La figure 4.4 représente respectivement la page d'inscription qui permettra à un utilisateur d'avoir un compte en précisant des informations personnelles ainsi qu'une page de connexion qui est utilisée par des patients déjà inscrits afin d'y accéder à leurs comptes.

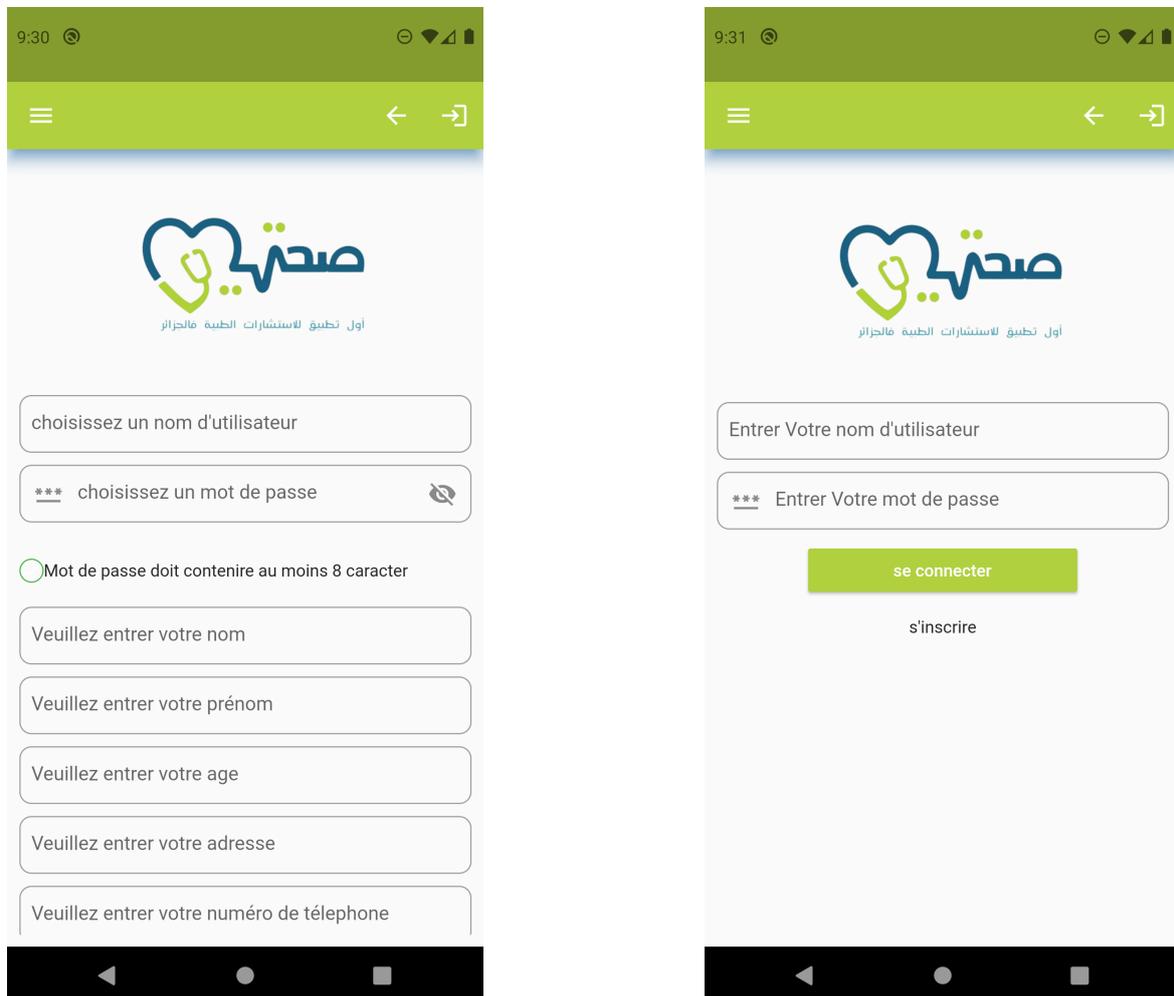


FIGURE 4.4 – L'interface d'inscription et de connexion

— **Espace médecin :**

La figure 4.5 illustre les interfaces affichées lors d'une connexion d'un utilisateur en tant que médecin.

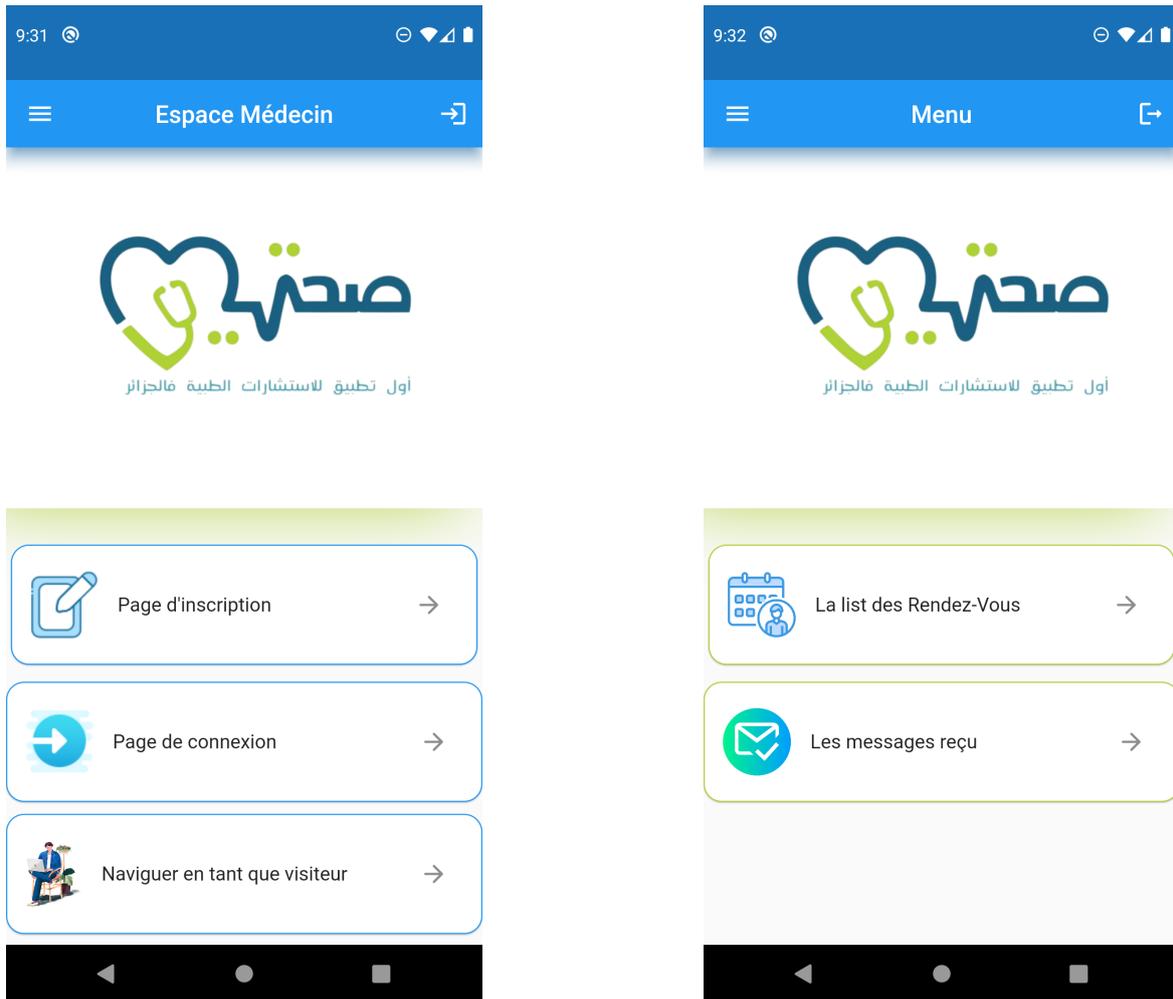


FIGURE 4.5 – Espace médecin

— **Menu des maladies**

La figure 4.6 représente une interface affichée que pour les utilisateurs ayant un compte afin de suivre leurs maladies.

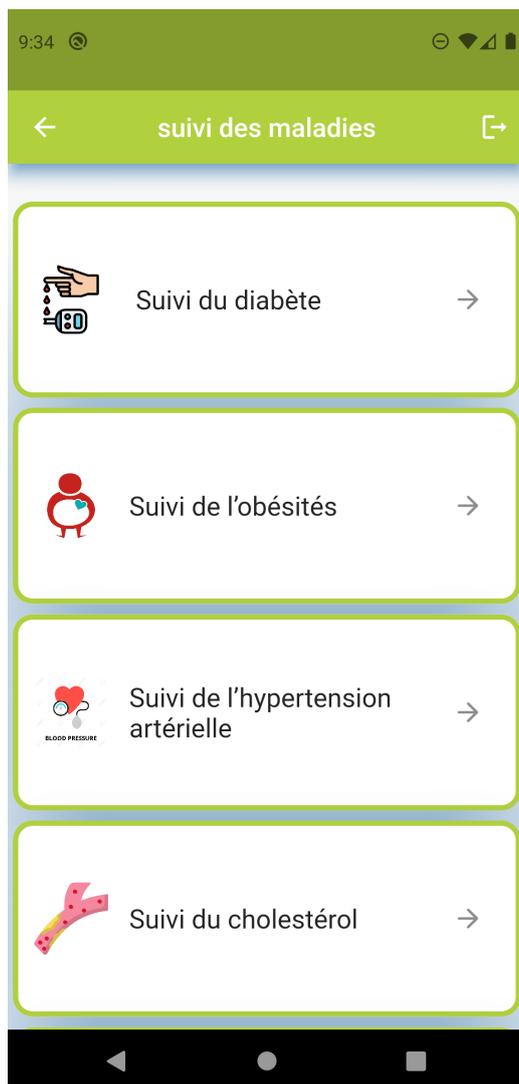


FIGURE 4.6 – Menu des maladies.

— **Programme sportif pour le cholestérol :**

La figure 4.7 représente l'interface programme sportif pour les patients atteints de cholestérol, ce dernier est exemplé par les différentes options qui pourront être suivies dans chaque maladie.

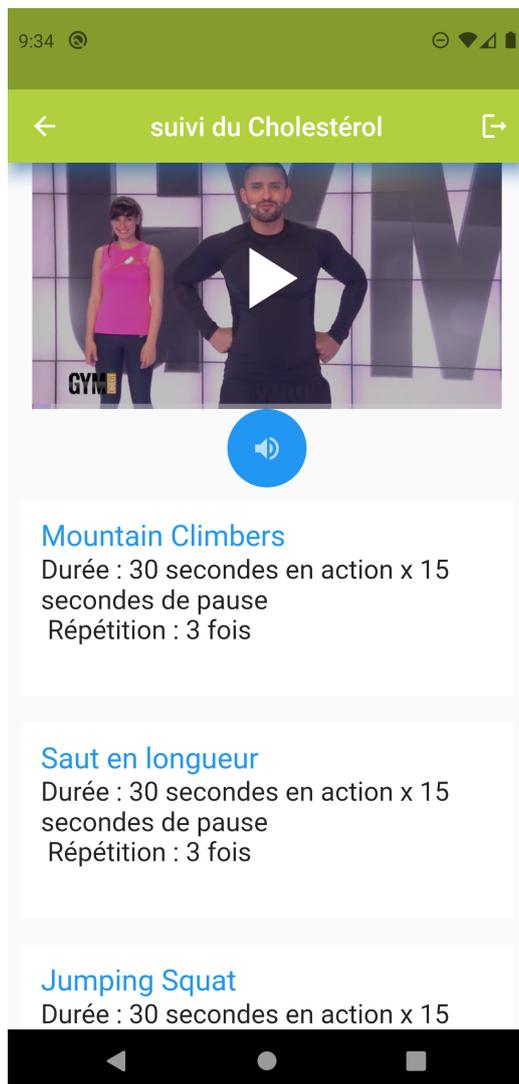


FIGURE 4.7 – Programme sportif pour le cholestérol.

2. web :

— l'interface d'authentification :

C'est la page de connexion d'un administrateur à son compte en introduisant son login et son mot de passe, la figure 4.8 illustre cette page de connexion.

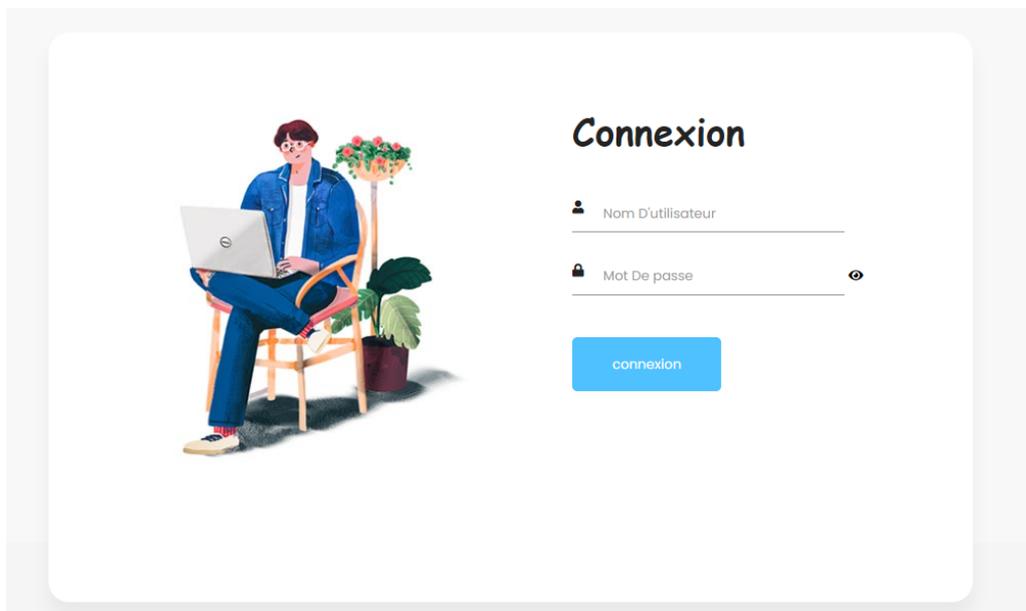


FIGURE 4.8 – Interface de connexion.

— **Interface d'accueil :**

Après l'introduction correcte des informations demandées lors de la connexion ,une page d'accueil sera affichée , la figure 4.9 représente cette interface d'accueil.

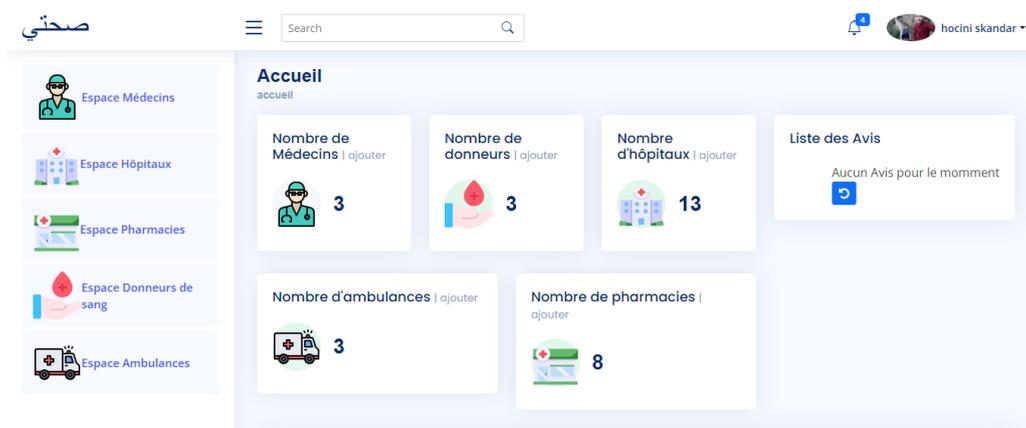


FIGURE 4.9 – Interface d'accueil.

— **Espace hôpitaux :**

L'action "cliquez sur espace hopitaux" déclenche un changement d'interface comme représenté dans la figure 4.10 .

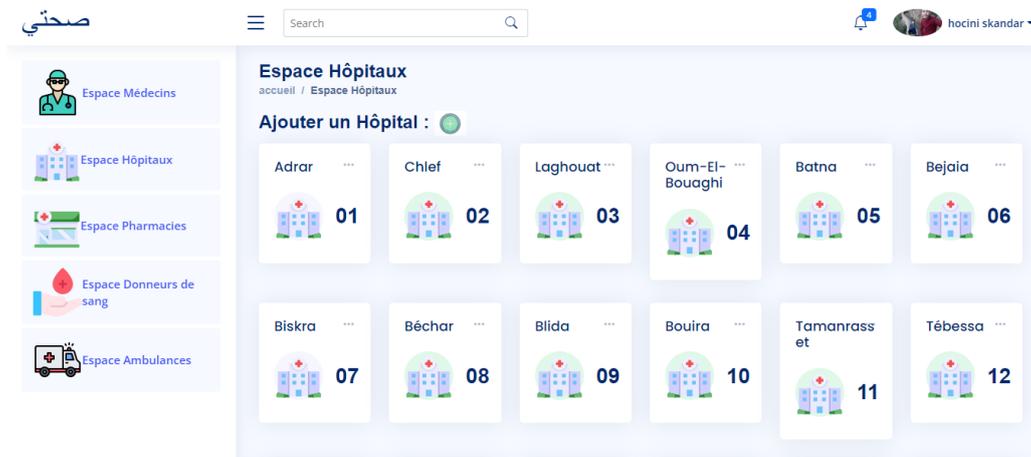


FIGURE 4.10 – Espace hôpitaux.

— **Interface de gestion des hôpitaux :**

Notre application web était conçue spécialement pour la gestion de l'application mobile par un administrateur qui pourra se connecter comme montré dans les deux figures précédentes. Afin de bien expliquer les opérations qu'un administrateur pourra effectuer nous allons vous présenter dans la figure 4.11 l'interface de gestion des hôpitaux.

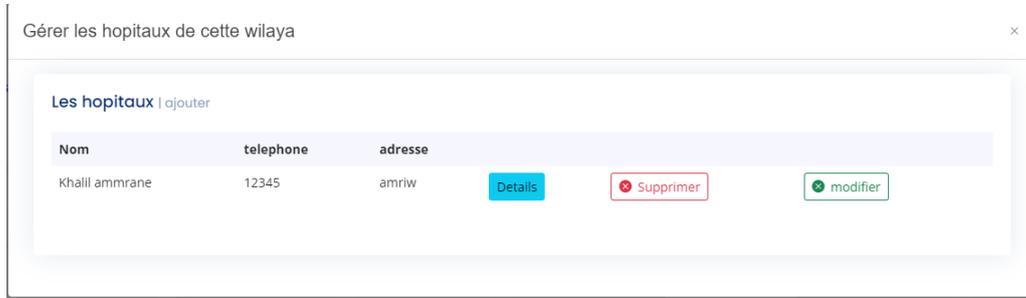


FIGURE 4.11 – Interface de gestion des hôpitaux.

4.5 Conclusion

Dans cette dernière partie de notre travail, nous avons présenté l'environnement de développement et les principaux outils utilisés, ainsi que les Framework les plus importants, qui nous ont permis de réaliser notre environnement en respectant les normes et les standards d'un bon système. Nous avons également présenté notre environnement par quelques interfaces accompagnées d'une brève description.

Conclusion Générale

Les applications de soins de santé mobiles sont intéressantes pour les soins primaires et les cas d'urgence. D'une part dans les pathologies chroniques doit être améliorées, d'autre part les smartphones font partie intégrante de la vie quotidienne d'un grand nombre de patients. Pour cela, L'objectif de notre projet de fin d'études était la conception et la réalisation d'un environnement de suivi des patients atteints des maladies chroniques, dans le but de faciliter le partage des informations différentes entre professionnels de santé et patients de manière sécurisée.

Ce document a débuté par une description générale des applications mobiles. Nous avons présenté en premier lieu les stratégies préconisées pour le développement de ces applications. Nous avons ensuite introduit les principaux systèmes d'exploitation mobiles sur lesquels repose leur fonctionnement. Nous avons aussi fait un tour d'horizon sur les problèmes des patients dans leur quotidien et rendant plus souple et plus efficace le suivi de leur maladie. Nous avons ensuite présenté les différentes étapes de la conception en commençant par l'analyse des besoins et la spécification des exigences, puis nous avons modélisé les fonctionnalités identifiées à l'issue de cette spécification en utilisant le formalisme UML afin de satisfaire les besoins des utilisateurs finaux de notre système.

Enfin, nous sommes passés à la partie réalisation de notre projet en développant notre environnement constitué d'une application mobile sous la plateforme Android ainsi l'application web en mettant en œuvre notre base de données relationnelle avec le langage de requêtes structurées MySQL. Ce projet a fait l'objet d'une expérience à la fois intéressante et enrichissante, qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine du développement et de la conception des systèmes complexes.

Des perspectives d'amélioration de notre application restent toutefois indispensables. Nous envisageons ainsi d'ajouter de nouvelles fonctionnalités telles que la gestion des rendez-vous d'une façon automatique, le développement d'une messagerie instantanée permettant aux patients d'échanger avec leurs médecins.

Bibliographie

- [1] <https://skylineapps.wordpress.com/2012/01/16/the-birth-of-the-mobile-applications/>.
- [2] <https://pulse.digital/service/application-mobile-multi-plateforme-ios-android/>.
- [3] <https://conceptic.io/application-mobile/>.
- [4] <https://infonet.fr/lexique/definitions/e-sante/>.
- [5] http://medecine.univ-batna2.dz/sites/default/files/medecine/files/benaldjia_interet_tic_en_sante_2021.pdf.
- [6] https://www.editions-ellipses.fr/PDF/9782340025547_extrait.pdf.
- [7] https://www.memoireonline.com/03/12/5548/m_Rapport-de-stage-sur-le-projet-Locate-my-car-google-map-android3.html.
- [8] http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/archi_comp.html.
- [9] Meier, Reto: *Android Application Développement. Etat unis*, wiley publishing . 2009.
- [10] <https://www.wrike.com/fr/project-management-guide/faq/quest-ce-que-les-contraintes-en-gestion-de-projet/>.
- [11] Frédéric, DI GALLO: *méthodologie du system d'information-UML*. 2001.
- [12] <https://www.msn.com/fr-xl/>.
- [13] https://www.memoireonline.com/08/09/2577/m_Conception-et-realisation-dune-application-de-suivi-de-patients-dans-un-etablissement.html.
- [14] http://next.owlapps.net/owlapps_apps/articles?id=1062093.
- [15] <https://www.africmemoire.com/>.
- [16] <https://docplayer.fr/83306743-Memoire-de-fin-de-cycle-theme.html>.
- [17] <https://emailing4you.wordpress.com/2011/05/10/up-unified-process/>.
- [18] <http://tvaira.free.fr/dev/sysml/sysml-contexte.pdf>.
- [19] Roques, Pascal: *UML 2 par la pratique 7ème édition*. Editions Eyrolles, 2009.
- [20] Roques, Pascal: *UML 2 : Modéliser une application web*. Editions Eyrolles, 2008.
- [21] Grimaldi, Michel: *Modélisation UML Diagrammes Structurels*. Thèse de doctorat, 2007.
- [22] Gabay, Joseph et David Gabay: *UML 2 Analyse et conception : Mise en œuvre guidée avec études de cas*. Dunod, 2008.
- [23] Christian, Soutou: *UML 2 pour les bases de données. EYROLLES*.
- [24] <https://www.lojiciels.com/quest-ce-que-la-racine-du-sdk-android/>.
- [25] <https://blogground.ro/fr/reponse-rapide-android-studio-est-il-base-sur-java/>.
- [26] <https://definir-tech.com/idee-intellij-2/>.

- [27] <https://www.freedaz.com/en/p/meilleur-editeur-de-code-88>.
- [28] <https://developer.android.com/studio/run/managing-avds>.
- [29] <https://blog.back4app.com/fr/lapplication-backend-flutter/>.
- [30] Lemainque, Fabrice: *HTML, XHTML, CSS, Scripts - Le Guide Complet. Micro Application*. 2010.
- [31] <https://getbootstrap.com/>.
- [32] <https://sites.google.com/site/portefeuillemalikchetouane/utilisation-du-php/>.

Résumé

En effet la complexité croissante de la médecine occidentale pousse de manière naturelle à la mise en place des systèmes d'information SI étant capables d'aider le praticien dans ses tâches quotidiennes. Afin d'atteindre cet objectif, il nous a été proposé de concevoir et réaliser un environnement m-santé constitué de deux applications complémentaires, la première est une application mobile sous Android assurant le suivi des patients tout en proposant plusieurs options et fonctionnalités comme la prise de rendez-vous en ligne ,la consultation de son carnet de santé, permettant ainsi de le guider dans le terrain en lui offrant toutes les informations nécessaires des différents hôpitaux, cabinets médicaux ainsi que les numéros de téléphones des ambulanciers et des donneurs de sang dans les 58 wilayas , permettant ainsi d'améliorer la collaboration entre les acteurs dans le secteur médical afin d'offrir un meilleur rendement et de meilleures prestations. La deuxième application quant à elle, elle s'agit d'une application web qui a pour but de compléter l'application mobile avec une sûreté et une fiabilité des informations publiées .

Abstract

Indeed, the increasing complexity of Western medicine naturally leads to the establishment of information systems IS capable of helping the practitioner in his daily tasks. In order to achieve this objective, we have been proposed to design and implement an m-health environment consisting of two complementary applications, the first is an Android-based mobile app that tracks patients while offering several options and features, such as online appointment booking, viewing your health book, allowing him to guide him in the field by offering him all the necessary information from the various hospitals, medical offices and the telephone numbers of ambulance attendants and blood donors in the 58 cities , to improve collaboration between players in the medical sector in order to offer better performance and benefits. The second application is a web application that aims to complement the mobile application with a security and reliability of the published information.