

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



Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master Professionnel en Informatique

Option : Administration et Sécurité des Réseaux

Thème

Conception et réalisation d'une application mobile pour le suivi d'un cabinet médical

Réalisé par

M. BARKA Anis

Devant le jury composé de

Président :	M. AMROUN Kamal	Maître de Conférences A	Université A. Mira Béjaïa
Examineur :	M. FARAH Zoubeyr	Maître de Conférences B	Université A. Mira Béjaïa
Examinatrice :	M ^{me} BOUADEM Nassima	Maître Assistante A	Université A. Mira Béjaïa
Encadrant :	M. TARI Abdelkamel	Professeur	Université A. Mira Béjaïa

Remerciements

C'est avec un immense plaisir que je réserve ces quelques lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

Je souhaite adresser, en premier lieu, mes remerciements les plus sincères à mon encadrant M. TARI Abdelkamel pour sa disponibilité, sa patience et son précieux suivi tout au long de la réalisation de ce travail.

Je tiens également à remercier les membres du jury d'avoir consacré une partie de leur temps à la lecture de ce mémoire et pour l'intérêt qu'ils ont porté à ce travail.

Mes remerciements s'étendent à tous mes enseignants du département d'Informatique de l'Université Abderrahmane Mira de Béjaïa.

Je remercie enfin toutes les personnes qui ont contribué de près ou de loin à l'accomplissement de ce travail.

Dédicaces

A mes très chers parents.

M. BARKA Anis

Table des matières

Table des matières	i
Table des figures	iii
Liste des tableaux	iv
Introduction générale	1
1 Généralités sur les applications mobiles	3
1.1 Introduction	3
1.2 Applications mobiles	3
1.2.1 Définition	3
1.2.2 Stratégies de développement	4
1.2.3 Systèmes d'exploitation mobiles	4
1.3 Applications mobiles pour le domaine médical	5
1.3.1 Dossier Médical Personnel	5
1.3.2 Dossier Médical Personnel Informatisé	6
1.3.3 Intérêt des applications mobiles dans le secteur médical	7
1.4 Conclusion	8
2 Analyse et conception	9
2.1 Introduction	9
2.2 Expression initiale des besoins	9
2.3 Processus de développement du système	9
2.4 Modélisation des besoins	10
2.4.1 Exigences fonctionnelles	10
2.4.2 Exigences non fonctionnelles	10
2.4.3 Spécification des exigences d'après les cas d'utilisation	11
2.4.4 Spécification détaillée des exigences	17
2.4.5 Diagrammes de séquence	20
2.5 Conception architecturale	26
2.5.1 Diagramme de classes	26
2.5.2 Passage au modèle relationnel	27
2.6 Conclusion	31

3 Réalisation	32
3.1 Introduction	32
3.2 Environnement de développement	32
3.2.1 Android Studio	32
3.2.2 IntelliJ IDEA	32
3.2.3 Bibliothèques utilisées	33
3.3 Implémentation de la base de données	34
3.4 Présentation de l'application	34
3.4.1 Arborescence des vues de notre application	34
3.4.2 Présentation des interfaces	35
3.4.3 Sécurité	40
3.5 Conclusion	41
Conclusion générale et perspectives	42
Bibliographie	43

Table des figures

2.1	Diagramme de cas d'utilisation associé au patient.	12
2.2	Diagramme de cas d'utilisation associé au médecin.	13
2.3	Diagramme de cas d'utilisation associé à l'assistant médical.	14
2.4	Diagramme de cas d'utilisation associé à l'administrateur.	15
2.5	Diagramme général des cas d'utilisation du système à réaliser.	16
2.6	Diagramme de séquence <i>Authentication</i>	21
2.7	Diagramme de séquence <i>Ajouter un dossier médical/patient</i>	22
2.8	Diagramme de séquence <i>Rechercher un patient/dossier médical</i>	23
2.9	Diagramme de séquence <i>Modifier un dossier médical</i>	24
2.10	Diagramme de séquence <i>Ajouter un rendez-vous au planning</i>	25
2.11	Diagramme de séquence <i>Prendre un rendez-vous</i>	26
2.12	Diagramme de classes de l'application à réaliser.	27
2.13	Règle de transformation des classes.	28
2.14	Règle de transformation de l'héritage (push-down).	30
2.15	Association un-à-plusieurs.	30
3.1	Arborescence des vues de notre application.	35
3.2	Interface menu de connexion.	36
3.3	Interface de connexion du médecin.	36
3.4	Interface menu des services disponibles pour le médecin.	37
3.5	Interface formulaire d'ajout d'un nouveau patient.	37
3.6	Interface des disponibilités du cabinet.	38
3.7	Interface de recherche d'un patient.	39
3.8	Interface de la liste des dossiers médicaux.	39
3.9	Interface du planning des rendez-vous.	40

Liste des tableaux

2.1	Description textuelle du cas d'utilisation <i>Authentification</i>	17
2.2	Description textuelle du cas d'utilisation <i>Ajouter un dossier médical/patient</i>	18
2.3	Description textuelle du cas d'utilisation <i>Rechercher un patient/dossier médical</i>	18
2.4	Description textuelle du cas d'utilisation <i>Modifier un dossier médical</i>	19
2.5	Description textuelle du cas d'utilisation <i>Ajouter un rendez-vous au planning</i>	19
2.6	Description textuelle du cas d'utilisation <i>Prendre un rendez-vous</i>	20

Introduction générale

De nos jours, le téléphone mobile n'est plus seulement utilisé pour les communications et l'échange de messages courts, de nouveaux usages sont apparus tels que la messagerie électronique, la navigation sur Internet, la musique, les vidéos, etc. Grâce à la généralisation des téléphones portables tactiles à écrans larges ainsi qu'au développement des logiciels et des réseaux, les applications mobiles sont capables de satisfaire un large éventail de besoins notamment dans le secteur de la santé.

La démarche médicale est fondée sur l'observation du malade. Les données médicales étaient rassemblées sous forme d'articles médicaux, de registres à visée épidémiologique et administrative. La mémoire du médecin était autrefois suffisante pour enregistrer les données relatives aux patients et servir l'exercice médical. Avec la multiplication des effets de l'environnement, de nos jours, la bonne tenue d'un dossier exige des moyens informatiques. L'automatisation du système d'information consiste à structurer et gérer un ensemble de données dont le but est de les organiser et d'avoir des résultats rapides. Toutefois, elle suscite de nombreuses interrogations de la part des usagers du système et des professionnels de santé, qu'il s'agisse du respect de la déontologie médicale et des droits de la personne, de la transformation des pratiques professionnelles et de la relation de confiance entre le patient et son praticien.

Dans ce contexte, nous sommes appelés à concevoir, développer et mettre en œuvre, une application mobile pour le suivi d'un cabinet médical afin d'améliorer la qualité des soins de santé délivrés aux patients, augmenter le rendement du personnel médical et faciliter les tâches administratives au sein du cabinet. 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.

Le présent mémoire est divisé en trois chapitres décrivant les étapes suivies pour la réalisation d'une application mobile destinée au suivi d'un cabinet médical. Dans le chapitre 1, nous présentons les applications mobiles et les stratégies préconisées pour leur développement. Nous exposons par la suite les avantages qu'ont apportés ces applications au domaine médical en introduisant le Dossier Médical Personnel. Dans le chapitre 2, nous analysons les besoins et identifions les acteurs qui joueront un rôle dans l'outil de gestion à mettre en œuvre. Nous définissons par la suite les fonctionnalités de ces acteurs à travers des diagrammes de cas d'utilisation et de séquence. Nous étendrons enfin la représentation de ces diagrammes au niveau conceptuel en construisant un diagramme de classes. Dans le chapitre 3, nous décrivons les choix technologiques effectués concernant l'environnement de développement de notre application et l'architecture matérielle du système, suivi d'une présentation des différentes interfaces de notre application mobile pour le suivi d'un cabinet médical. Enfin, nous

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

Chapitre 1

Généralités sur les applications mobiles

1.1 Introduction

L'essor et les progrès récents survenus dans les technologies mobiles ont permis la réalisation de multiples appareils mobiles prenant de plus en plus de place sur le marché et dans le paysage numérique où les projets d'applications mobiles sont devenus un moyen essentiel de création de nouveaux services à destination des *mobinautes*. Dans ce chapitre, nous présenterons en premier lieu les applications mobiles et les stratégies préconisées pour leur développement. Nous aborderons par la suite les systèmes d'exploitation mobiles les plus répandus sur le marché. Enfin, nous verrons les avantages qu'ont apportés ces applications au domaine médical et plus particulièrement à la gestion des dossiers patients.

1.2 Applications mobiles

De nos jours, les applications mobiles prennent une place de plus en plus importante dans notre quotidien tant les fonctionnalités qu'elles offrent nous facilitent grandement la vie. Dans cette section nous aborderons les notions d'applications mobiles et des systèmes d'exploitation sur lesquels repose leur fonctionnement.

1.2.1 Définition

Une application mobile est un logiciel applicatif développé pour être installé sur un appareil mobile, généralement un téléphone mobile, un téléphone intelligent ou une tablette numérique. Les applications mobiles sont des programmes relativement légers, autonomes, utilisés pour des services de l'information, des médias sociaux, des jeux, etc. [1].

Avec les possibilités matérielles incorporées aux terminaux mobiles (caméra, GPS, gyroscope, etc.), les applications installées sur ces derniers peuvent intégrer des fonctionnalités spécifiques et dédiées pour les utilisateurs, permettant ainsi d'enrichir leur spectre fonctionnel et imaginer des usages non couverts jusqu'à présent par les systèmes d'information tels que la géolocalisation, le scan de QR codes (*Quick Response codes*), la réalité augmentée, le m-commerce¹, etc.

1. Mobile commerce, regroupe l'ensemble des applications commerciales liées aux terminaux mobiles.

1.2.2 Stratégies de développement

La conception d'applications mobiles peut se faire suivant trois stratégies de développement distinctes. Dans ce qui suit, nous donnons un bref aperçu de chaque stratégie [2].

1.2.2.1 Application native

Une application native est une application mobile spécifiquement développée pour un système d'exploitation mobile. Elle est conçue avec le langage et les outils associés à son système d'exploitation, et installée directement sur le mobile. Cette installation se faisant soit au travers d'un téléchargement via Internet soit par déploiement depuis un ordinateur connecté au mobile. Les tests pour vérifier le comportement de ces applications nécessitent des compétences techniques spécifiques et des appareils très coûteux.

1.2.2.2 Application Web

Une application mobile Web est une application développée en HTML, accessible et exécutable par le biais d'un navigateur Internet pour téléphone mobile. Elle utilise le navigateur du Smartphone et ne nécessite pas forcément de télécharger l'application. Les applications mobiles Web complètent les applications natives qui sont développées spécifiquement pour un système d'exploitation et qui doivent être téléchargées et installées par les *mobinautes*. Elles s'adressent donc à l'ensemble des utilisateurs de mobiles, et non à une population spécifique utilisant une marque bien précise. Toutefois, les applications Web doivent être testées pour chaque navigateur, résolution et taille d'écran, à l'instar de n'importe quel site Web.

1.2.2.3 Application hybride

L'application hybride est une application pour mobiles qui combine des éléments HTML5 sous forme d'application mobile Web et des éléments d'une application native permettant l'utilisation des fonctionnalités natives des Smartphones et d'être distribuée en tant qu'application sur les stores des systèmes mobiles (App Store, Play Store, etc.).

1.2.3 Systèmes d'exploitation mobiles

Tout comme un ordinateur dispose d'un système d'exploitation, les téléphones mobiles se composent également d'une plateforme qui contrôle toutes leurs fonctionnalités. Ceci est connu comme un système d'exploitation mobile. Généralement connu sous le nom d'OS (*Operating System*) mobile, il s'agit d'un système d'exploitation qu'exploite un appareil mobile tel qu'un Smartphone, une tablette tactile, etc. Il contrôle et coordonne toutes les opérations de base du téléphone mobile comme les options d'écran tactile, Bluetooth, Wifi, appareil photo, etc. et assure la liaison entre les ressources matérielles, l'utilisateur et les applications (traitement de texte, jeux vidéo, etc.) [3].

1.2.3.1 iOS

Anciennement connu sous le nom iPhone OS, iOS est le système d'exploitation mobile développé par *Apple* pour l'iPhone, l'iPod et l'iPad. Il est dérivé de OS X dont il partage les fondations (le noyau hybride XNU basé sur le micro-noyau Mach, les services Unix et Cocoa, etc.). iOS est le deuxième système d'exploitation le plus répandu sur le marché [5].

1.2.3.2 Windows Phone

Développé par *Microsoft* pour les Smartphones et Pocket PC, Windows Phone succède à Windows Mobile en proposant en plus des applications basiques comme la messagerie électronique, Internet, Chat et Multimédia, des fonctionnalités média sociaux tels que Facebook et Twitter [3].

1.2.3.3 BlackBerry OS

Système d'exploitation fonctionnant sur les Smartphones BlackBerry, il permet aux développeurs de mettre en place des applications en utilisant les APIs (*Application Programming Interface*) BlackBerry. Toute application doit être signée numériquement par le compte RIM (*Research In Motion*) du développeur [3].

1.2.3.4 Android

Android est un système d'exploitation *open source* développé par *Android Inc.*, une startup rachetée par Google en 2007. Fondé sur un noyau Linux, le système a d'abord été conçu pour les Smartphones et tablettes tactiles, puis s'est diversifié dans les objets connectés et ordinateurs comme les télévisions, les voitures, les ordinateurs et les Smartwatch. Android a été conçu pour intégrer au mieux des applications existantes de Google telles que le service de messagerie électronique Gmail, celui de la cartographie Google Maps, ou encore YouTube, Google Talk et Google Calendar [4]. En 2015, Android est le système d'exploitation le plus utilisé dans le monde avec plus de 80% des parts de marché dans les Smartphones [7].

1.3 Applications mobiles pour le domaine médical

Dans cette section, nous abordons les avantages apportés par les applications mobiles au domaine médical. Pour cela, nous présentons le Dossier Médical Personnel et l'intérêt de son informatisation.

1.3.1 Dossier Médical Personnel

Le Dossier Médical Personnel (DMP) est un dossier qui rassemble les informations médicales relatives à un patient nécessaires à la coordination des soins : prescriptions, comptes rendus d'hospitalisation, résultats d'analyses, mentions d'allergies, différents bilans, etc. [6].

1.3.1.1 Composition du Dossier Médical Personnel

Le Dossier Médical Personnel se compose des parties suivantes [6] :

- **Partie administrative** : contenant des données dites démographiques (identité, âge, adresse, profession, etc.);
- **Partie médicale** : contenant les données recueillies par le personnel médical et leur interprétation ; les diagnostics, les ordonnances, rapports sur examens, prescriptions sur examens, actes pratiqués sur le malade et leurs résultats, etc.
- **Partie instrumentale** : contenant les résultats des analyses, radios et images numériques, etc.

1.3.1.2 Intérêt du Dossier Médical Personnel

Le dossier médical du patient est un outil à usages multiples. Tout d'abord, le dossier médical est un outil de suivi du patient. C'est dans ce dossier que les demandes d'examens et leurs résultats sont colligés et que le médecin exprime ses réflexions, ses interrogations et ses conclusions. Le dossier patient est également un outil de communication. En effet, le travail médical est de plus en plus un travail d'équipe (cabinets de groupe, centres de santé, hôpitaux, etc.). Les informations pertinentes doivent être disponibles à tous les professionnels qui ont traité le malade. Le dossier est un des meilleurs moyens d'assurer la communication de ces informations. De plus, il représente un outil de gestion hospitalière efficace ; connaître les diagnostics, les actes thérapeutiques, le coût entraîné par la population de malades qui fréquente un service, un département ou un hôpital est indispensable à celui qui a la responsabilité de gérer ces structures. Enfin, les dossiers patients hospitaliers peuvent donner des aperçus intéressants sur la santé de la population [6]. En résumé, le dossier patient est d'un grand intérêt, d'une part, pour le patient qui bénéficie d'une meilleure continuité des soins et d'une sécurité accrue grâce à la réduction du risque d'erreurs de médication, et d'autre part pour les professionnels de santé qui profitent d'une réduction des activités administratives et la capacité d'accéder aux dossiers complets à la demande.

1.3.2 Dossier Médical Personnel Informatisé

Le Dossier Médical Personnel Informatisé (DMPI) est tout simplement la version électronique du dossier papier du patient, pouvant être consulté par les professionnels de santé, afin d'effectuer des opérations bien définies nécessaires à la prise en charge du patient.

1.3.2.1 Caractéristique du Dossier Médical Personnel Informatisé

Toute conception d'un dossier personnel doit être guidée par les caractéristiques suivantes [6] :

- Être précis, concis et logique ;
- Garantir une rapidité d'accès selon les besoins ;

- Assurer la sécurité et le respect de la confidentialité ainsi que le secret médical ;
- Désigner les acteurs qui sont amenés à y porter des écritures afin qu'on puisse, le cas échéant, leur demander un complément d'informations.

Le partage du dossier patient entre professionnels de santé ne peut avoir lieu qu'à la condition que soient pleinement respectées les règles déontologiques et la législation en vigueur [6] :

- Aucune information extraite du dossier médical partagé ne doit faire l'objet d'une utilisation à des fins commerciales directes ou indirectes ;
- Lorsque le médecin se sert pour des publications scientifiques de ses observations médicales, il doit faire en sorte que l'identification des patients soit impossible.

1.3.2.2 Intérêt du Dossier Médical Personnel Informatisé

L'informatisation du dossier médical permet d'apporter les avantages suivants [1] :

- Amélioration de la qualité des soins ;
- Amélioration de la lisibilité des informations ;
- Amélioration de la protection et de la confidentialité des données ;
- Partage de compétences (grâce à la télémédecine) ;
- Réduction des frais de transfert des personnes ;
- Réduction des durées et des coûts de séjour en hospitalisation ;
- Réduction des coûts de création et de gestion de l'information médicale.

1.3.3 Intérêt des applications mobiles dans le secteur médical

Étant donné que le personnel médical est appelé à être la majorité de son temps de service en mouvement dans les hôpitaux ou dans les cabinets médicaux, les applications mobiles implémentant un système qui répond aux besoins des professionnels de santé représente le meilleur moyen pour faciliter leurs tâches. Ainsi cela permet aux personnels médicaux d'optimiser leur temps de service et d'améliorer leur rendement.

Pour mener à bien ce projet, nous comptons utiliser les ressources dont les mobiles disposent, afin d'implémenter des fonctionnalités très pratiques pour le domaine médical, par exemple le fait de gérer le planning des rendez-vous efficacement, pouvoir prendre un rendez-vous chez son médecin traitant sans avoir à se déplacer, etc.

1.4 Conclusion

Le monde de la santé se lie de plus en plus au numérique, et depuis quelques années, se développe sur les mobiles. De la e-santé à la m-santé, des outils voient le jour pour les professionnels de santé et les patients. Dans ce chapitre, nous avons présenté en premier lieu les applications mobiles et les principaux systèmes d'exploitation sur lesquels repose leur fonctionnement. Nous avons par la suite décrit les caractéristiques du Dossier Médical Personnel et l'intérêt de son informatisation. Enfin, nous avons exposé les avantages qu'ont apportés les applications mobiles au domaine médical. Le chapitre suivant sera consacré à l'analyse des besoins et à la conception de notre application mobile pour le suivi d'un cabinet médical.

Chapitre 2

Analyse et conception

2.1 Introduction

La première étape de la conception consiste à analyser la situation pour tenir compte des contraintes, des risques et de tout autre élément pertinent afin de développer un système répondant aux besoins du client. Le présent chapitre nous permet d'identifier toutes les fonctionnalités de notre future application pour chaque type d'utilisateurs en recensant les besoins fonctionnels, et permet d'établir la liste des exigences traduites par les besoins non fonctionnels. Ceci se fera par l'identification des acteurs et la définition de tous les besoins qui seront par la suite modélisés par un diagramme de cas d'utilisation pour chaque entité, suivi de leurs diagrammes de séquence et enfin du diagramme de classes.

2.2 Expression initiale des besoins

Notre future application mobile aura comme objectifs d'assurer la bonne gestion du cabinet médical et de faciliter le suivi des patients afin de parvenir à une organisation performante. Cette application devra permettre de gérer facilement l'accès aux dossiers médicaux ainsi que leurs mises à jour afin d'assurer un meilleur suivi de l'état de santé des patients par leur médecin traitant.

2.3 Processus de développement du système

Un processus définit une séquence d'étapes, partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objectif d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles [9].

Le processus que nous avons suivi pour la réalisation de notre projet est le processus 2TUP (*2 Track Unified Process*). Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. *2 Track* signifie littéralement que

le processus suit deux chemins. Il s'agit des chemins *fonctionnels* et d'*architecture technique*, qui correspondent aux deux axes de changement imposés aux systèmes informatiques [10].

2.4 Modélisation des besoins

Dans cette section, nous identifions les fonctionnalités de notre future application mobile. Ces fonctionnalités sont ensuite modélisées par des diagrammes UML appropriés.

2.4.1 Exigences fonctionnelles

Notre application mobile devra regrouper toutes les fonctionnalités nécessaires de gestion, de recherche et de découverte détaillée.

- **Gestion** : l'objectif principal de notre application consiste à administrer les dossiers médicaux et le suivi des patients, en plus de gérer les périodes de disponibilité du cabinet et le réapprovisionnement en matériel médical. En d'autres termes, administrer l'activité du cabinet médical.
- **Recherche** : le médecin aura accès à tous les dossiers médicaux de ses patients. Il aura à sa disposition une méthode de recherche, lui permettant d'atteindre le dossier médical souhaité. Le résultat de la recherche sera disponible sur une page particulière et devra pouvoir être facilement parcourue.
- **Découverte détaillée** : l'utilisateur autorisé pourra voir les détails concernant chaque dossier médical, on y trouvera en particulier les informations civiles, les prescriptions, l'évolution du traitement du patient et les résultats obtenus, etc.

2.4.2 Exigences non fonctionnelles

Nous répartissons les exigences non fonctionnelles en deux catégories ; les exigences de qualité et les exigences de performance.

2.4.2.1 Exigences de qualité

Les exigences de qualité sont résumées par les points décrits ci-après.

- **Ergonomie attractive et efficace**

Le design des interfaces doit permettre une identification immédiate de ses différents éléments pour permettre à l'utilisateur d'accéder de manière intuitive à ce qu'il cherche, dès la première utilisation.

- **Formulaires intelligibles**

Les formulaires ne doivent contenir qu'un minimum de champs à renseigner et devrons donc être particulièrement soignés.

2.4.2.2 Exigences de performance

- Les dossiers médicaux des patients doivent être confidentiels et sécurisés.
- L'application doit supporter une activité intense sans dégradation des performances.

2.4.3 Spécification des exigences d'après les cas d'utilisation

Acteurs et cas d'utilisation sont les concepts UML fondamentaux pour la spécification des exigences. Dans cette section, nous les identifierons à partir de l'expression initiale des besoins de notre étude de cas. Nous structurerons, relierons et classerons ensuite ces cas d'utilisation et élaborerons les représentations graphiques UML associées.

2.4.3.1 Identification des acteurs

Un *acteur* représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [8].

Dans le cas de notre système, nous avons identifié principalement quatre (04) acteurs en interaction avec celui-ci :

- **Le médecin** : désigne la personne habilitée à pratiquer la médecine et chargée d'examiner les patients et de diagnostiquer les maladies, lésions et pathologies afin de les traiter.
- **L'assistant(e) médical(e)** : désigne la personne qui, dans un cabinet médical, effectue des tâches qui s'articulent autour de quatre grands axes ; l'accueil des patients et la gestion des rendez-vous, les tâches administratives, l'assistance des praticiens et les travaux de laboratoire.
- **Le patient** : désigne la personne examinée par le médecin et qui se voit administrer un traitement.
- **L'administrateur** : c'est la personne chargée de la maintenance de l'application et de la gestion des comptes des utilisateurs. Il veille au bon fonctionnement du serveur de données et à sa sécurité.

2.4.3.2 Identification des cas d'utilisation

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 [8].

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

- **Patient**

- Créer un compte utilisateur.
- Consulter son dossier médical.
- Consulter les disponibilités du cabinet.
- Prendre un rendez-vous.

Le cas d'utilisation *authentication* est un cas qui doit être réalisé afin de permettre à chaque acteur d'exécuter ses propres cas d'utilisation. Ce cas d'utilisation est qualifié de « fragment » ; il ne représente pas un objectif à part entière de l'acteur, mais plutôt un objectif de niveau intermédiaire.

La figure 2.1 représente le diagramme de cas d'utilisation associé au patient.

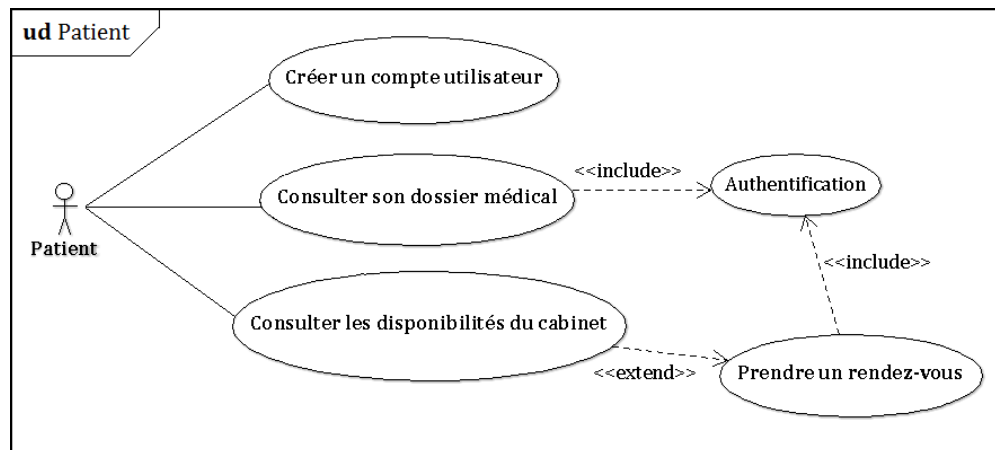


FIGURE 2.1 – Diagramme de cas d'utilisation associé au patient.

- Médecin

- Ajouter un dossier médical/patient.
- Consulter un dossier médical.
- Modifier un dossier médical.
- Ajouter une image médicale.
- Rechercher un patient/dossier médical.
- Consulter le planning des rendez-vous.
- Ajouter une note/mémo partagée.

Ces cas d'utilisation sont représentés sur la figure 2.2.

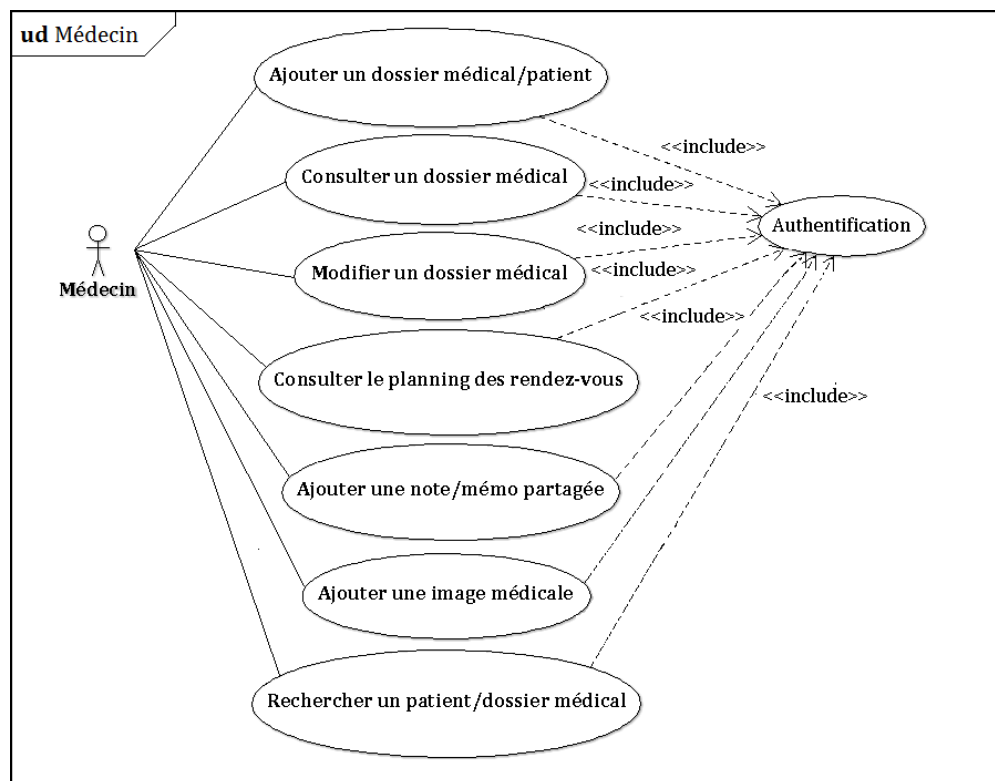


FIGURE 2.2 – Diagramme de cas d'utilisation associé au médecin.

- **Assistant(e) médical(e)**

- Ajouter un rendez-vous au planning.
- Supprimer un rendez-vous du planning.
- Rechercher un rendez-vous.
- Consulter une note/mémo partagée.
- Modifier les disponibilités du cabinet.

La figure 2.3 représente le diagramme de cas d'utilisation associé à l'assistant médical.

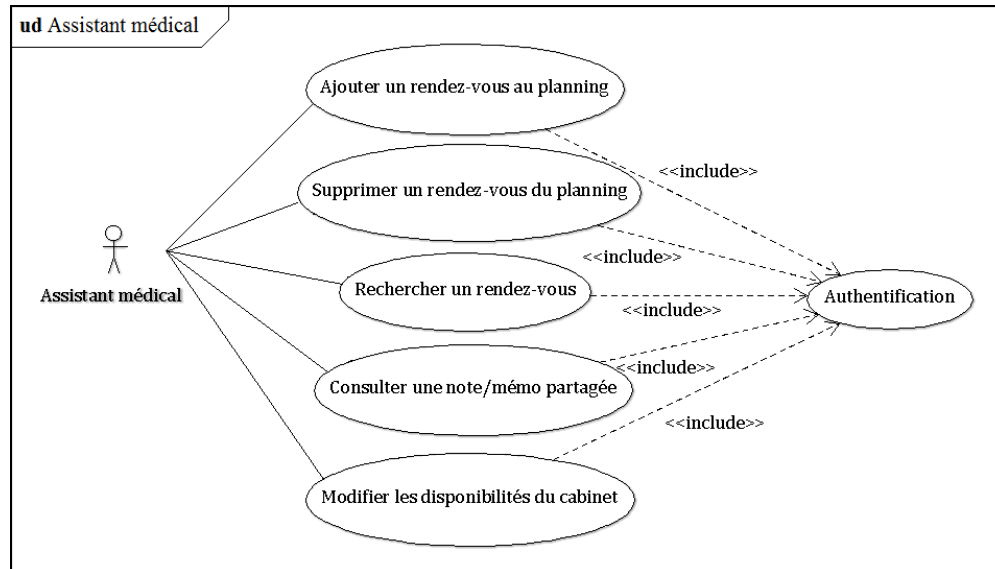


FIGURE 2.3 – Diagramme de cas d'utilisation associé à l'assistant médical.

- **Administrateur**

- Ajouter un compte utilisateur.
- Supprimer un compte utilisateur.
- Récupérer un mot de passe.
- Modifier les privilèges de l'assistant.

Ces cas d'utilisation sont représentés sur la figure 2.4.

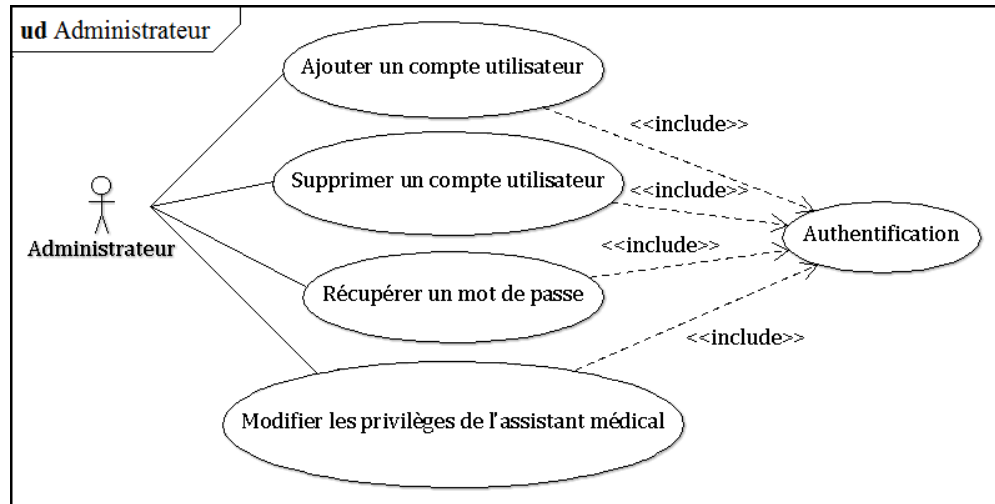


FIGURE 2.4 – Diagramme de cas d'utilisation associé à l'administrateur.

Le médecin pouvant avoir les mêmes cas d'utilisation que l'administrateur, il existe donc une relation de spécialisation entre ces deux acteurs.

Nous rassemblons tous les cas d'utilisation précédemment identifiés, dans un diagramme général comme sur la figure 2.5.

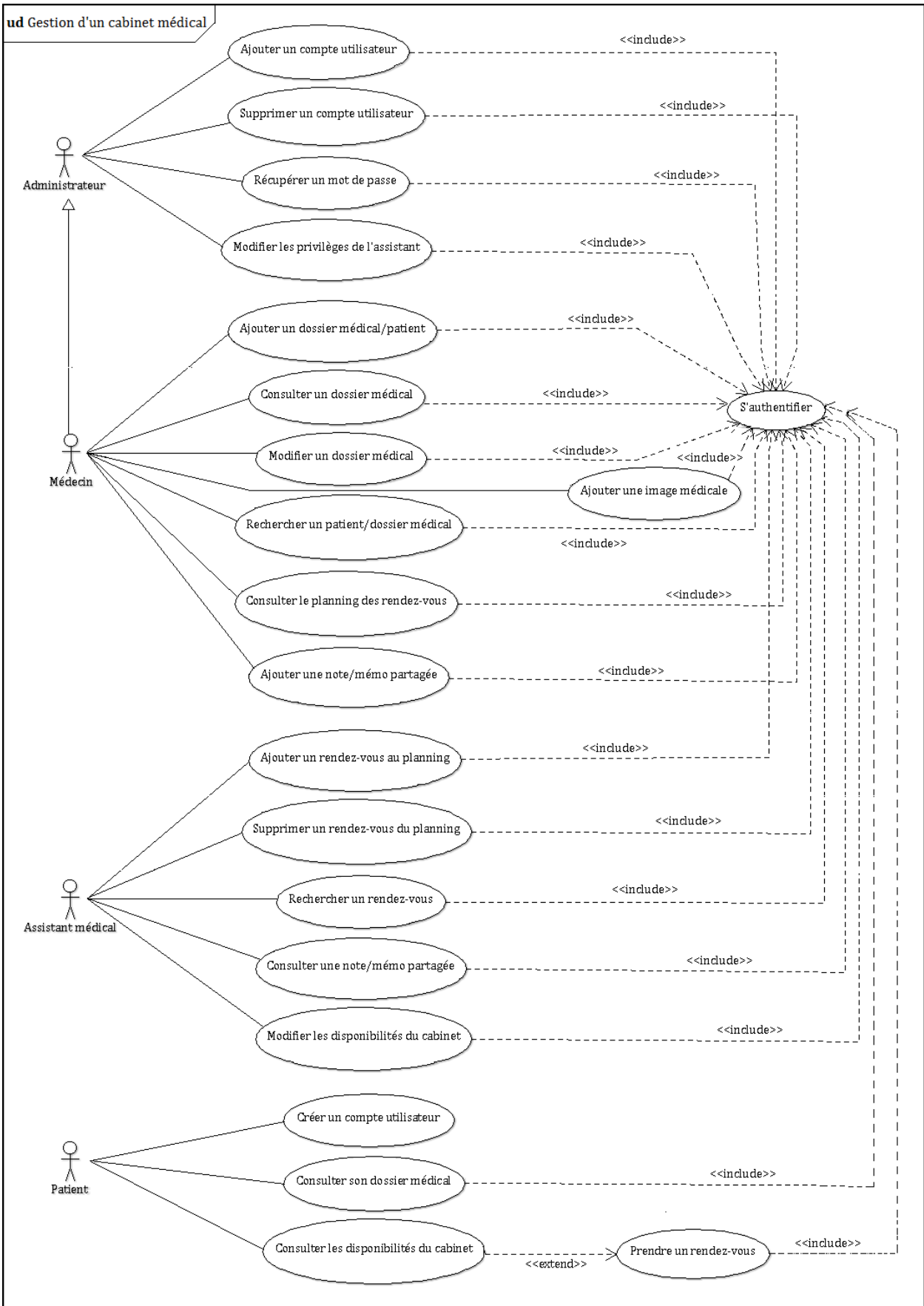


FIGURE 2.5 – Diagramme général des cas d'utilisation du système à réaliser.

2.4.4 Spécification détaillée des exigences

Dans ce qui suit, nous décrirons de façon détaillée certains cas d'utilisation identifiés précédemment en recensant de façon textuelle toutes les interactions entre les acteurs et le système.

2.4.4.1 Cas d'utilisation *Authentification*

Sommaire d'identification							
Titre du cas d'utilisation	Authentification						
Résumé	L'authentification permet d'accéder à des fonctionnalités réservées à un type d'utilisateur donné.						
Acteurs	Patient, Médecin, Assistant médical, Administrateur						
Description des scénarios							
Préconditions	- Application accessible.						
Scénario nominal	<table border="1"> <tbody> <tr> <td>1. L'utilisateur accède à la page d'authentification.</td> <td>2. L'utilisateur choisit sa catégorie.</td> </tr> <tr> <td>3. Le système affiche le formulaire d'authentification.</td> <td>4. L'utilisateur saisit son login et son mot de passe.</td> </tr> <tr> <td>5. Le système vérifie l'existence du compte.</td> <td>6. Le système renvoie l'interface correspondante.</td> </tr> </tbody> </table>	1. L'utilisateur accède à la page d'authentification.	2. L'utilisateur choisit sa catégorie.	3. Le système affiche le formulaire d'authentification.	4. L'utilisateur saisit son login et son mot de passe.	5. Le système vérifie l'existence du compte.	6. Le système renvoie l'interface correspondante.
1. L'utilisateur accède à la page d'authentification.	2. L'utilisateur choisit sa catégorie.						
3. Le système affiche le formulaire d'authentification.	4. L'utilisateur saisit son login et son mot de passe.						
5. Le système vérifie l'existence du compte.	6. Le système renvoie l'interface correspondante.						
Enchainements d'erreur	5a. Aucun compte correspondant au couple login/mot de passe indiqué : le système lève une exception ; le cas d'utilisation se termine en échec.						
Postconditions	L'utilisateur est authentifié et accède aux fonctionnalités qui lui sont dédiées.						

TABLE 2.1 – Description textuelle du cas d'utilisation *Authentification*.

2.4.4.2 Cas d'utilisation *Ajouter un dossier médical/patient*

Sommaire d'identification							
Titre du cas d'utilisation	Ajouter un dossier médical/patient						
Résumé	Le médecin ajoute un nouveau dossier médical relatif à un patient.						
Acteurs	Médecin						
Description des scénarios							
Préconditions	- Application accessible. - Le médecin est authentifié.						
Scénario nominal	<table border="1"> <tr> <td>1. Le médecin demande le formulaire d'ajout.</td> <td>2. Le système renvoie le formulaire.</td> </tr> <tr> <td>3. Le médecin remplit le formulaire d'ajout.</td> <td>4. Le médecin valide et envoie le formulaire.</td> </tr> <tr> <td>5. Le système vérifie que le patient n'existe pas déjà.</td> <td>6. Le système crée un dossier médical et renvoie un message confirmant l'action.</td> </tr> </table>	1. Le médecin demande le formulaire d'ajout.	2. Le système renvoie le formulaire.	3. Le médecin remplit le formulaire d'ajout.	4. Le médecin valide et envoie le formulaire.	5. Le système vérifie que le patient n'existe pas déjà.	6. Le système crée un dossier médical et renvoie un message confirmant l'action.
1. Le médecin demande le formulaire d'ajout.	2. Le système renvoie le formulaire.						
3. Le médecin remplit le formulaire d'ajout.	4. Le médecin valide et envoie le formulaire.						
5. Le système vérifie que le patient n'existe pas déjà.	6. Le système crée un dossier médical et renvoie un message confirmant l'action.						
Enchainements d'erreur	4a. Le médecin annule l'action : échec du cas d'utilisation. 5a. Le nom du patient existe déjà : le cas d'utilisation se termine en échec.						
Postconditions	Un nouveau dossier médical est ajouté à la liste.						

TABLE 2.2 – Description textuelle du cas d'utilisation *Ajouter un dossier médical/patient*.2.4.4.3 Cas d'utilisation *Rechercher un patient/dossier médical*

Sommaire d'identification					
Titre du cas d'utilisation	Rechercher un patient/dossier médical				
Résumé	Ce cas d'utilisation permet au médecin d'effectuer une recherche sur les patients de son cabinet.				
Acteurs	Médecin				
Description des scénarios					
Préconditions	- Application accessible. - Le médecin est authentifié. - Liste des patients non vide.				
Scénario nominal	<table border="1"> <tr> <td>1. Le médecin lance une recherche rapide avec mots-clés.</td> <td>2. Le système affiche une page résultat.</td> </tr> <tr> <td>3. Le médecin sélectionne un patient.</td> <td>4. Le système présente une fiche détaillée du patient sélectionné.</td> </tr> </table>	1. Le médecin lance une recherche rapide avec mots-clés.	2. Le système affiche une page résultat.	3. Le médecin sélectionne un patient.	4. Le système présente une fiche détaillée du patient sélectionné.
1. Le médecin lance une recherche rapide avec mots-clés.	2. Le système affiche une page résultat.				
3. Le médecin sélectionne un patient.	4. Le système présente une fiche détaillée du patient sélectionné.				
Enchainements alternatifs	2a. Plusieurs patients retournés : le médecin peut choisir d'effectuer une recherche avancée en choisissant un critère de recherche (date de la dernière consultation, par exemple).				
Enchainements d'erreur	2a. Patient inexistant : le cas d'utilisation se termine en échec.				
Postconditions	Le médecin accède au dossier du patient souhaité.				

TABLE 2.3 – Description textuelle du cas d'utilisation *Rechercher un patient/dossier médical*.

2.4.4.4 Cas d'utilisation *Modifier un dossier médical*

Sommaire d'identification			
Titre du cas d'utilisation	Modifier un dossier médical		
Résumé	Le médecin met à jour les informations relatives à un patient.		
Acteurs	Médecin		
Description des scénarios			
Préconditions	<ul style="list-style-type: none"> - Application accessible. - Le médecin est authentifié. - Dossier médical existant. 		
Scénario nominal	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> 1. Le médecin demande le formulaire de modification. 3. Le médecin met à jour les informations du patient. 5. Le médecin confirme la modification. </td> <td style="width: 50%; vertical-align: top;"> 2. Le système renvoie le formulaire. 4. Le système demande une validation. 6. Le système affiche un message de confirmation. </td> </tr> </table>	1. Le médecin demande le formulaire de modification. 3. Le médecin met à jour les informations du patient. 5. Le médecin confirme la modification.	2. Le système renvoie le formulaire. 4. Le système demande une validation. 6. Le système affiche un message de confirmation.
1. Le médecin demande le formulaire de modification. 3. Le médecin met à jour les informations du patient. 5. Le médecin confirme la modification.	2. Le système renvoie le formulaire. 4. Le système demande une validation. 6. Le système affiche un message de confirmation.		
Enchaînements d'erreur	5a. Le médecin annule l'action : échec du cas d'utilisation.		
Postconditions	Le dossier médical correspondant à un patient donné possède de nouvelles informations.		

TABLE 2.4 – Description textuelle du cas d'utilisation *Modifier un dossier médical*.2.4.4.5 Cas d'utilisation *Ajouter un rendez-vous au planning*

Sommaire d'identification			
Titre du cas d'utilisation	Ajouter un rendez-vous au planning		
Résumé	L'assistant(e) médical(e) ajoute au planning un rendez-vous pris par un patient.		
Acteurs	Assistant(e) médical(e)		
Description des scénarios			
Préconditions	<ul style="list-style-type: none"> - Application accessible. - L'assistant(e) médical(e) est authentifié(e) - Liste des rendez-vous non pleine. 		
Scénario nominal	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> 1. L'assistant médical demande le formulaire d'ajout. 3. L'assistant médical remplit le formulaire d'ajout. 5. Le système vérifie que l'horaire n'est pas déjà pris. </td> <td style="width: 50%; vertical-align: top;"> 2. Le système renvoie le formulaire. 4. L'assistant médical valide et envoie le formulaire. 6. Le système ajoute le rendez-vous au planning et renvoie un message confirmant l'action. </td> </tr> </table>	1. L'assistant médical demande le formulaire d'ajout. 3. L'assistant médical remplit le formulaire d'ajout. 5. Le système vérifie que l'horaire n'est pas déjà pris.	2. Le système renvoie le formulaire. 4. L'assistant médical valide et envoie le formulaire. 6. Le système ajoute le rendez-vous au planning et renvoie un message confirmant l'action.
1. L'assistant médical demande le formulaire d'ajout. 3. L'assistant médical remplit le formulaire d'ajout. 5. Le système vérifie que l'horaire n'est pas déjà pris.	2. Le système renvoie le formulaire. 4. L'assistant médical valide et envoie le formulaire. 6. Le système ajoute le rendez-vous au planning et renvoie un message confirmant l'action.		
Enchaînements d'erreur	4a. L'assistant médical annule l'action : échec du cas d'utilisation. 5a. L'horaire est déjà pris : le cas d'utilisation se termine en échec.		
Postconditions	Un nouveau rendez-vous est ajouté au planning du médecin.		

TABLE 2.5 – Description textuelle du cas d'utilisation *Ajouter un rendez-vous au planning*.

2.4.4.6 Cas d'utilisation *Prendre un rendez-vous*

Sommaire d'identification			
Titre du cas d'utilisation	Prendre un rendez-vous		
Résumé	Le patient prend un rendez-vous pour une consultation.		
Acteurs	Patient		
Description des scénarios			
Préconditions	<ul style="list-style-type: none"> - Application accessible. - Le patient est authentifié. - Liste des rendez-vous non pleine. 		
Scénario nominal	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> 1. Le patient demande le formulaire de prise de rendez-vous. 3. Le patient remplit le formulaire et choisi un horaire. 5. Le système vérifie que l'horaire n'est pas déjà pris. </td> <td style="width: 50%; vertical-align: top;"> 2. Le système renvoie le formulaire. 4. Le patient valide et envoie le formulaire. 6. Le système ajoute le rendez-vous au planning et renvoie un message confirmant l'action. </td> </tr> </table>	1. Le patient demande le formulaire de prise de rendez-vous. 3. Le patient remplit le formulaire et choisi un horaire. 5. Le système vérifie que l'horaire n'est pas déjà pris.	2. Le système renvoie le formulaire. 4. Le patient valide et envoie le formulaire. 6. Le système ajoute le rendez-vous au planning et renvoie un message confirmant l'action.
1. Le patient demande le formulaire de prise de rendez-vous. 3. Le patient remplit le formulaire et choisi un horaire. 5. Le système vérifie que l'horaire n'est pas déjà pris.	2. Le système renvoie le formulaire. 4. Le patient valide et envoie le formulaire. 6. Le système ajoute le rendez-vous au planning et renvoie un message confirmant l'action.		
Enchainements d'erreur	4a. Le patient annule l'action : échec du cas d'utilisation. 5a. L'horaire est déjà pris : le cas d'utilisation se termine en échec.		
Postconditions	Le patient a pris un rendez-vous chez son médecin traitant.		

TABLE 2.6 – Description textuelle du cas d'utilisation *Prendre un rendez-vous*.

Pour documenter les cas d'utilisation, la description textuelle est indispensable car elle seule permet de communiquer facilement avec les utilisateurs et de s'entendre sur la terminologie employée. En revanche, le texte présente des limites puisqu'il est difficile de montrer comment les enchainements se succèdent. Il est donc recommandé de compléter la description textuelle par un ou plusieurs diagrammes dynamiques UML (diagrammes de séquence, d'activité, d'état-transition ou encore de collaboration) [8].

2.4.5 Diagrammes de séquence

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 [9].

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

2.4.5.1 Cas d'utilisation *Authentification*

Le premier scénario pour l'utilisateur consiste à s'authentifier auprès du système. La chronologie de ce scénario est représentée par la figure 2.6.

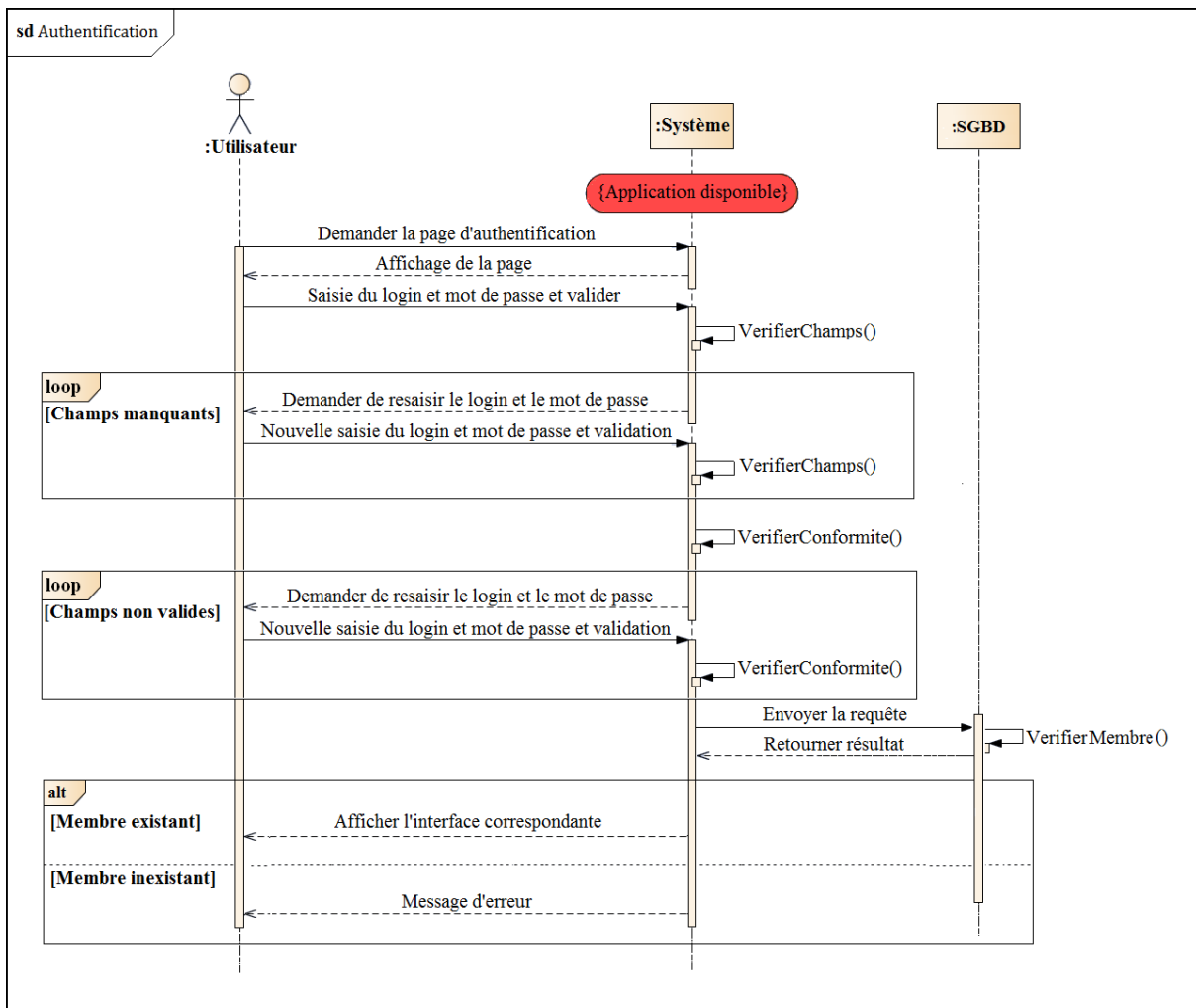


FIGURE 2.6 – Diagramme de séquence *Authentification*.

2.4.5.2 Cas d'utilisation *Ajouter un dossier médical/patient*

Le scénario d'ajout d'un dossier médical (patient) se fait selon la chronologie représentée par la figure 2.7.

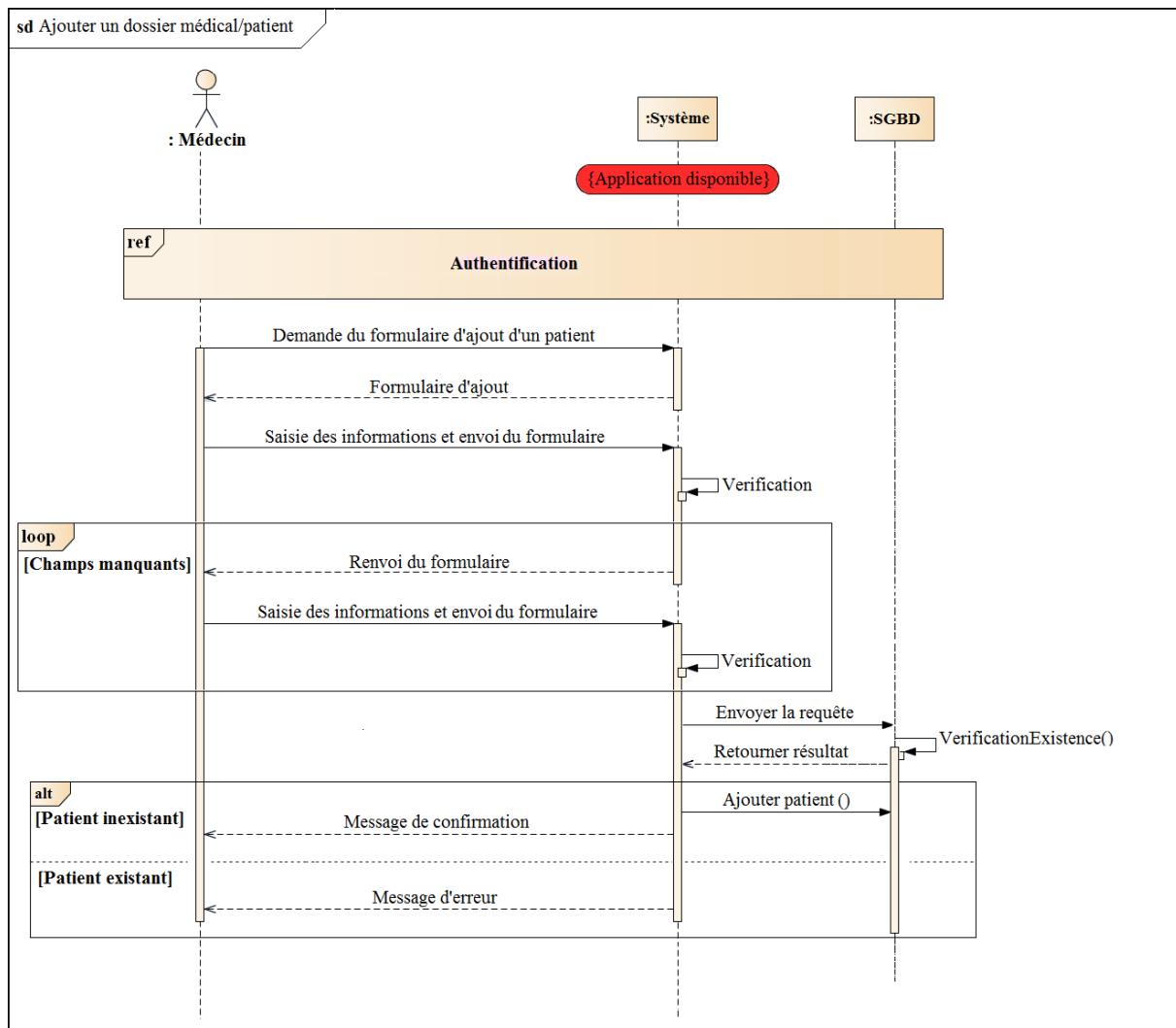


FIGURE 2.7 – Diagramme de séquence *Ajouter un dossier médical/patient*.

2.4.5.3 Cas d'utilisation *Rechercher un patient/dossier médical*

Le médecin a la possibilité d'effectuer deux types de recherche ; simple ou avancée.

- *Recherche simple* : le médecin saisit le nom et prénom du patient à rechercher.
- *Recherche avancée* : le médecin sélectionne un critère (date de la dernière consultation par exemple) puis remplit le champs du formulaire correspondant.

Le scénario de recherche d'un patient (dossier médical) se fait selon la chronologie représentée par la figure 2.8.

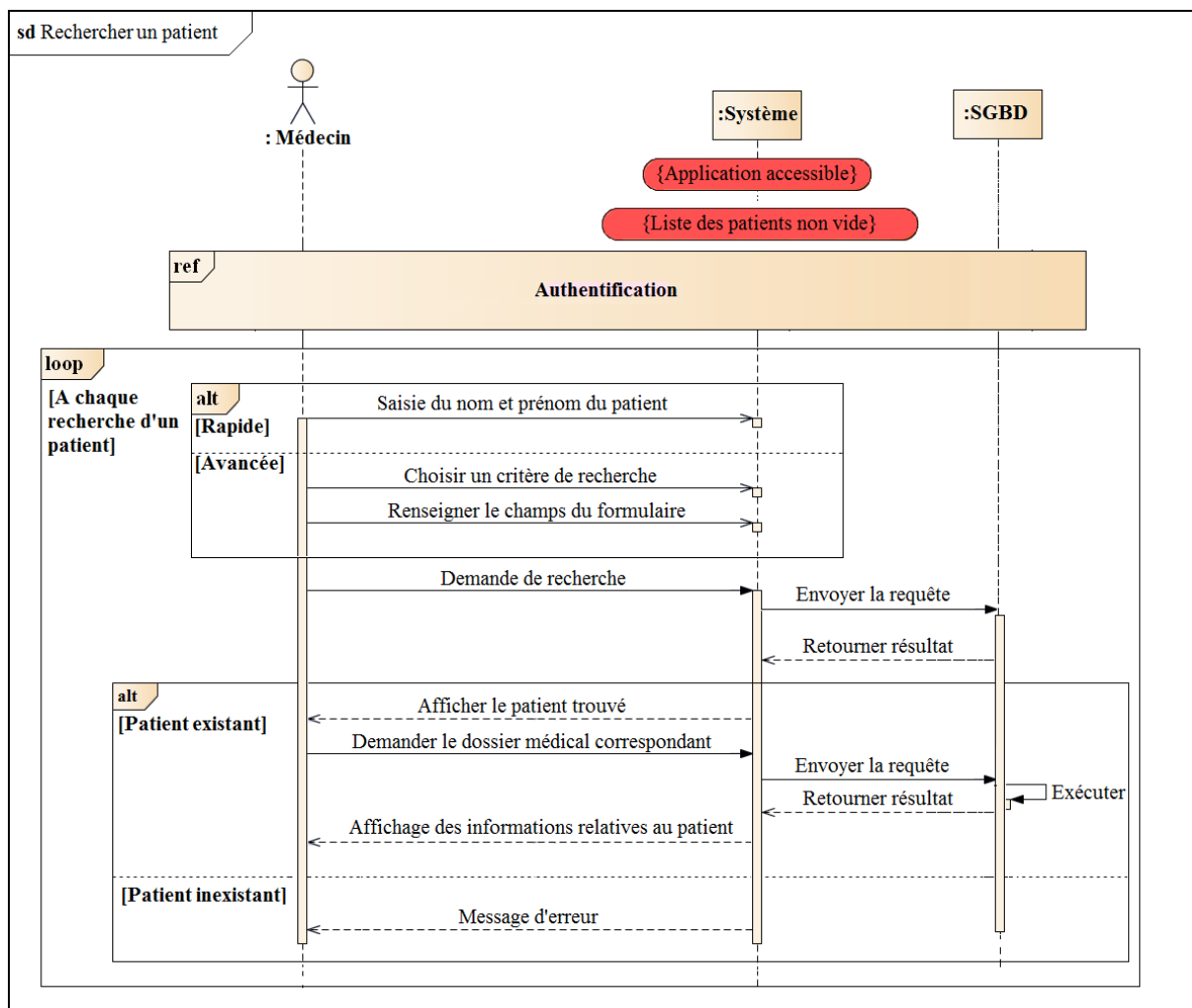


FIGURE 2.8 – Diagramme de séquence *Rechercher un patient/dossier médical*.

2.4.5.4 Cas d'utilisation *Modifier un dossier médical*

La chronologie du scénario relatif à la modification d'un dossier médical est représentée par la figure 2.9.

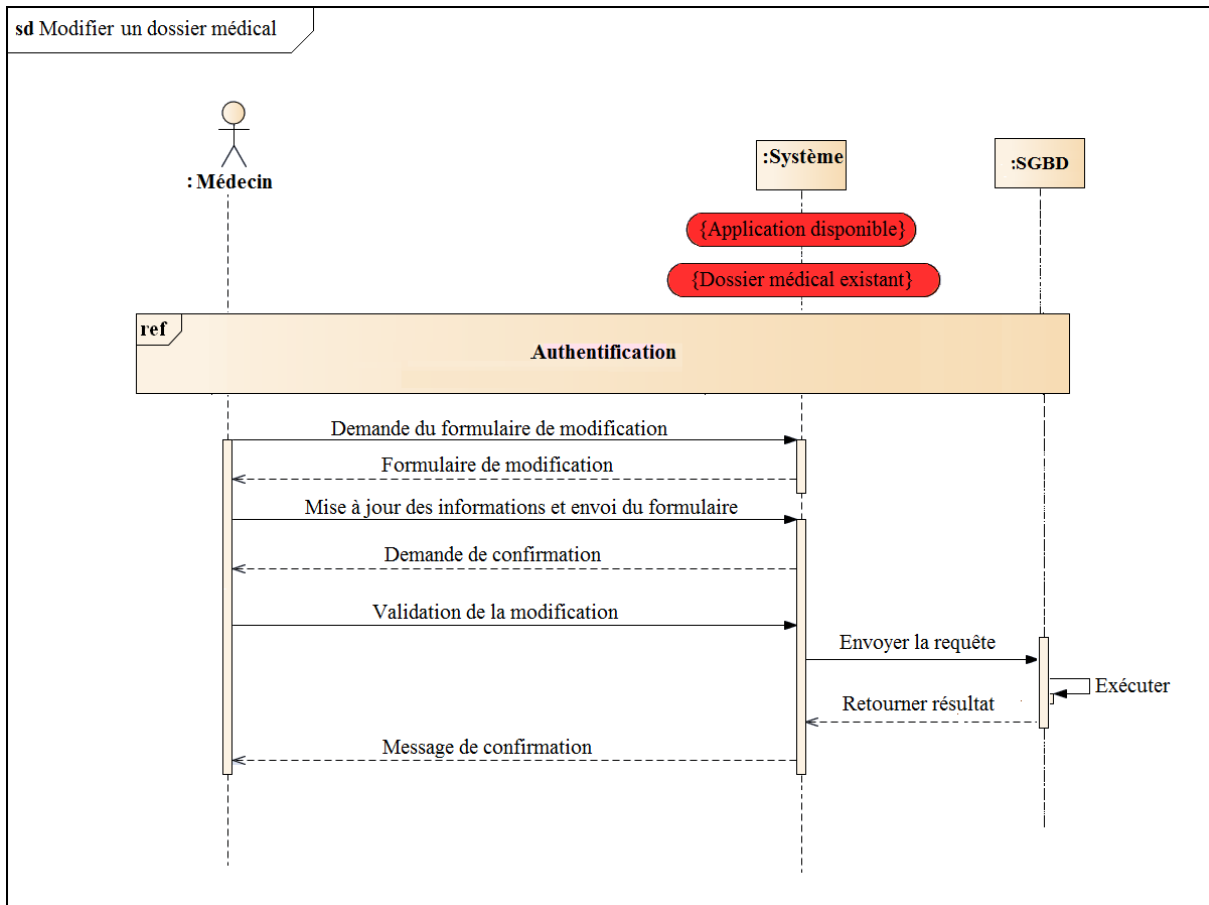


FIGURE 2.9 – Diagramme de séquence *Modifier un dossier médical*.

2.4.5.5 Cas d'utilisation *Ajouter un rendez-vous au planning*

Le scénario d'ajout d'un rendez-vous au planning du médecin se fait selon la chronologie représentée par la figure 2.10.

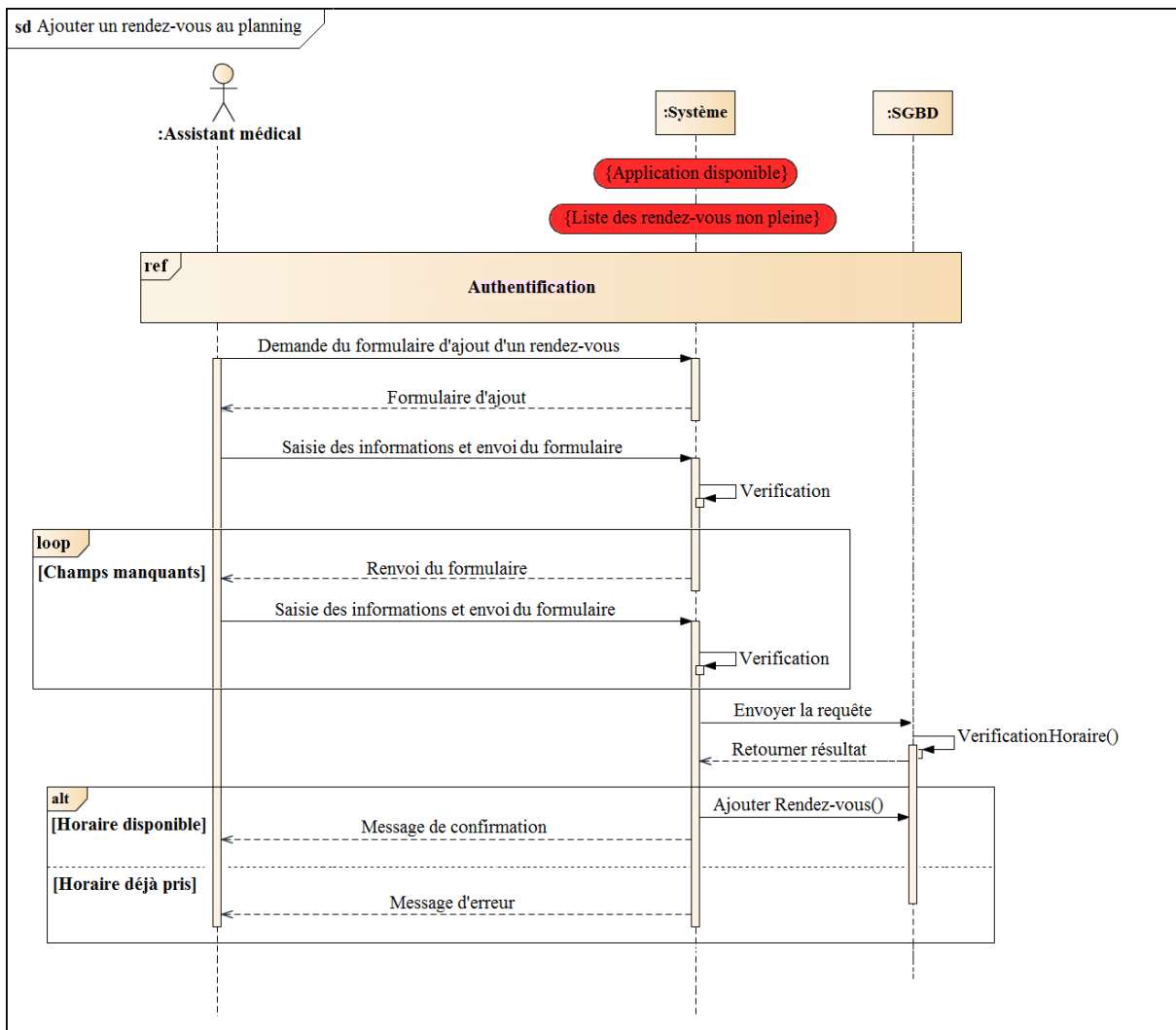


FIGURE 2.10 – Diagramme de séquence *Ajouter un rendez-vous au planning*.

2.4.5.6 Cas d'utilisation *Prendre un rendez-vous*

Le scénario de prise de rendez-vous pour une consultation se fait selon la chronologie représentée par la figure 2.11.

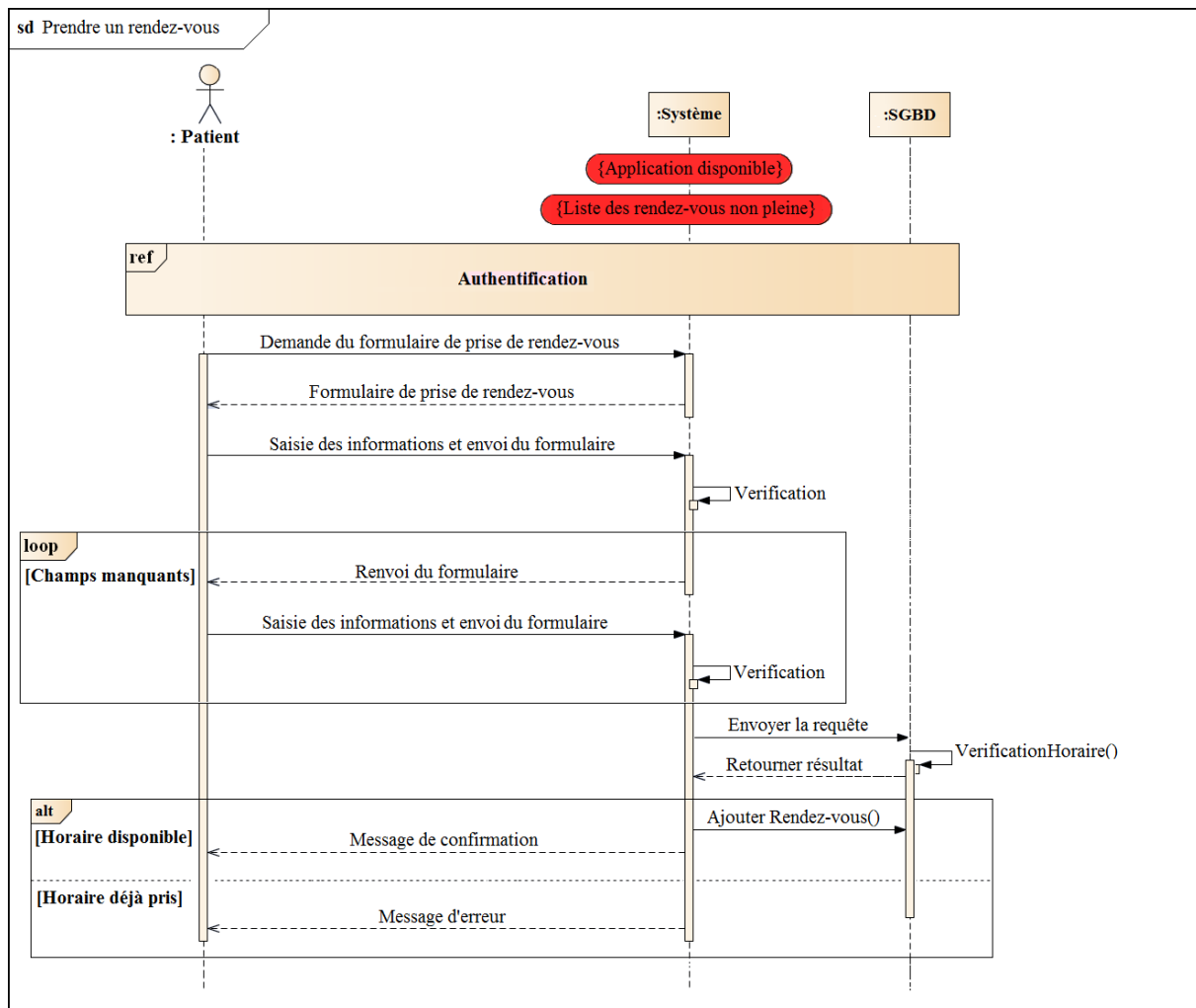


FIGURE 2.11 – Diagramme de séquence *Prendre un rendez-vous*.

2.5 Conception architecturale

2.5.1 Diagramme de classes

Un diagramme de classe se définit comme étant un ensemble de classes contenant des attributs et des opérations, reliées les unes aux autres par des relations et ceci en ayant des conditions de participation (cardinalités) ; il s'agit de la version UML de la base de données [8].

Le diagramme de classes de notre application est représenté par la figure 2.12.

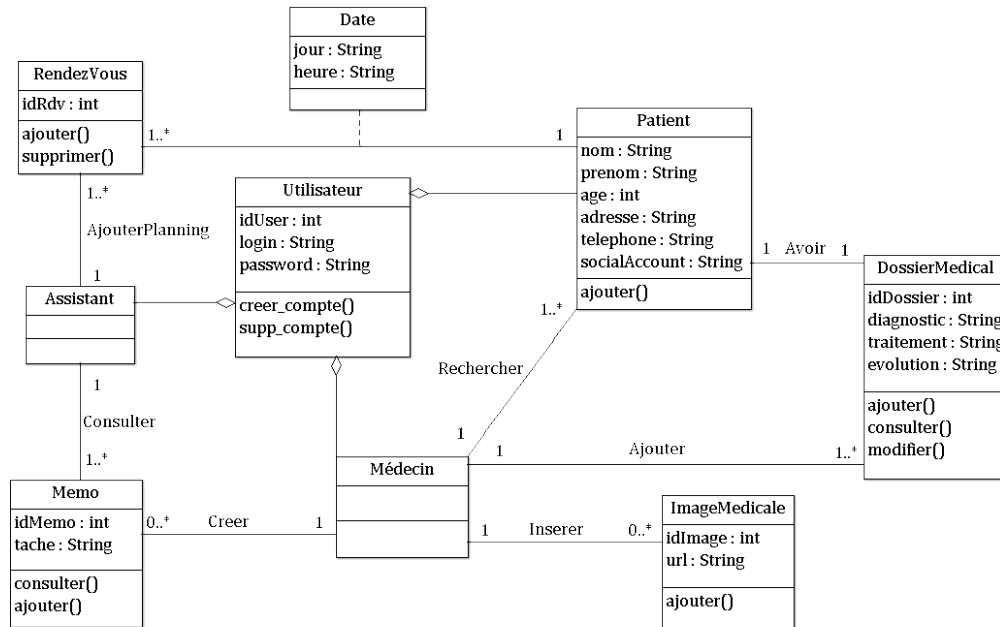


FIGURE 2.12 – Diagramme de classes de l’application à réaliser.

2.5.2 Passage au modèle relationnel

Dans cette section, nous passons au modèle relationnel à partir de la description conceptuelle, afin de pouvoir implémenter notre base de données.

2.5.2.1 Terminologie de l’approche relationnelle

Dans ce qui suit, nous décrirons les différentes terminologies de l’approche relationnelle [9].

- **Domaine**

Il s’agit de l’ensemble des valeurs admises pour un attribut, il établit les valeurs acceptables dans une colonne. Une relation peut faire appel au même domaine pour plusieurs de ses attributs. La notion de domaine est fondamentale en matière de gestion de données. Elle permet d’exprimer des contraintes d’intégrité sémantique très fortes sur les données d’une BD.

- **Classe**

Une classe représente la description abstraite d’un ensemble d’objets possédant les mêmes caractéristiques.

- **Attribut**

Un attribut est un identifiant (un nom) décrivant une information stockée dans une base de données.

- **Clé primaire**

Une clé primaire est un ensemble minimal de colonnes permettant d'identifier de manière unique chaque tuple dans une table.

- **Clé étrangère**

Il s'agit d'une ou plusieurs colonnes dans une table qui a pour but d'assurer une liaison entre deux tables. On y arrive en dupliquant la clé primaire de la deuxième table dans la première. On l'appelle aussi clé externe.

- **Tuple**

Un tuple représente une ligne dans une table.

2.5.2.2 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 notre application à partir du modèle de classes obtenu. Les règles utilisées pour le passage sont décrites ci-après.

- **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.

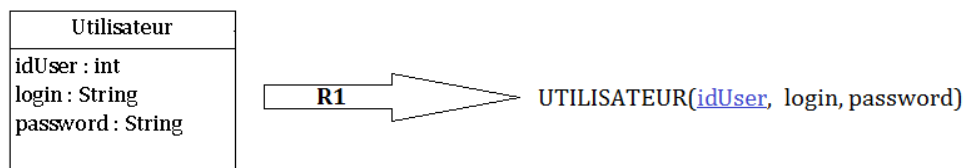


FIGURE 2.13 – Règle de transformation des classes.

En appliquant cette règle à notre modèle, nous obtenons les relations suivantes :

- ★ UTILISATEUR (idUser, login, password)
- ★ PATIENT (nom, prenom, age, adresse, telephone, socialAccount)
- ★ MEDECIN ()
- ★ ASSISTANT ()
- ★ DOSSIERMEDICAL (idDossier, diagnostic, traitement, evolution)
- ★ IMAGEMEDICALE (idImage, url)
- ★ MEMO (idMemo, tache)
- ★ RENDEZVOUS (idRdy)

• 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 sur-classe, 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 sur-classe. 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 sur-classe.

1. Application de la décomposition descendante (push-down)

Aucun utilisateur ne peut être à la fois « Médecin », « Patient » et « Assistant médical ». De plus, il n'existe pas un utilisateur n'étant ni médecin, ni patient, ni assistant. Donc les trois relations héritent le contenu intégral de la relation issue de la super-classe « Utilisateur ».

D'après la règle, il n'est pas nécessaire de faire apparaître la relation « Utilisateur » au niveau logique. On dit que « Utilisateur » est une classe abstraite (il n'existe pas d'instances de cette classe).

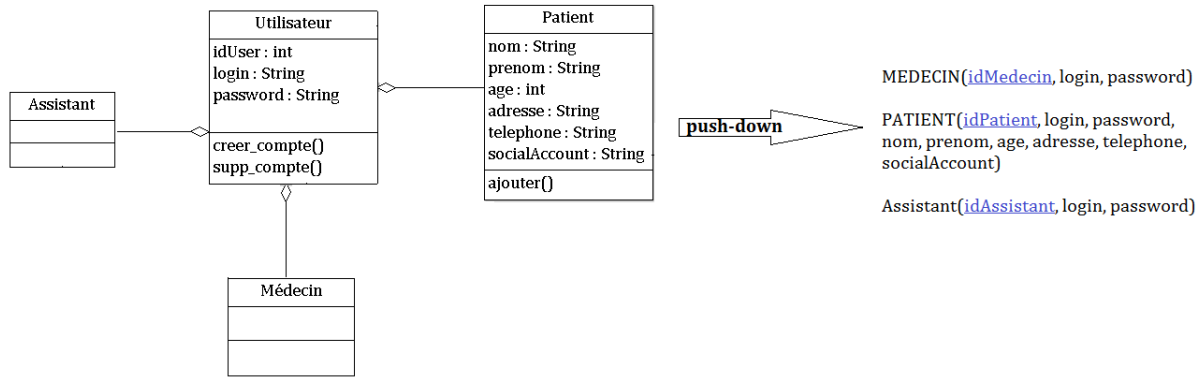


FIGURE 2.14 – Règle de transformation de l’héritage (push-down).

• 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.

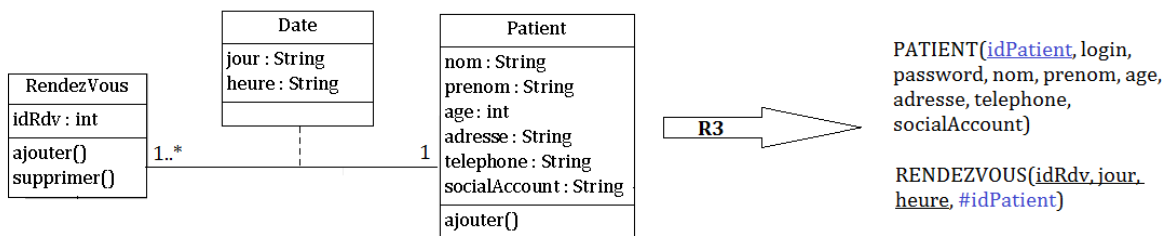


FIGURE 2.15 – Association un-à-plusieurs.

En appliquant cette règle à notre modèle, nous obtenons les relations suivantes :

- ★ PATIENT (idPatient, login, password, nom, prenom, age, adresse, telephone, socialAccount, idMedecin) où **idMedecin** est une clé étrangère qui fait référence au schéma de relation MEDECIN.
- ★ DOSSIERMEDICAL (idDossier, diagnostic, traitement, evolution, idMedecin) où **idMedecin** est une clé étrangère qui fait référence au schéma de relation MEDECIN.
- ★ IMAGEMEDICALE (idImage, url, idMedecin) où **idMedecin** est une clé étrangère qui fait référence au schéma de relation MEDECIN.
- ★ MEMO (idMemo, tache, idMedecin, idAssistant) où **idMedecin** est une clé étrangère qui fait référence au schéma de relation MEDECIN et **idAssistant** une clé étrangère qui fait référence au schéma de relation ASSISTANT.
- ★ RENDEZVOUS (idRdv, jour, heure, idPatient, idAssistant) où **idPatient** est une clé étrangère qui fait référence au schéma de relation PATIENT et **idAssistant** une clé étrangère qui fait référence au schéma de relation ASSISTANT.

2.6 Conclusion

Dans ce chapitre, nous avons décrit de façon détaillée les cas d'utilisation en recensant de manière 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. Par la suite, en définissant les relations entre les entités, nous sommes parvenus à concevoir le diagramme de classes donnant ainsi une vue plus structurée des éléments qui formeront la base de données liée à notre application. Enfin, ce chapitre nous a permis de préparer la phase de réalisation qui concrétisera tout ce qui a été présenté jusque là.

Chapitre 3

Réalisation

3.1 Introduction

Ce chapitre est consacré à la partie pratique de la réalisation de notre application mobile. Il comporte une description des outils de développement utilisés ainsi qu'une présentation de notre application mobile. Le choix de nos outils de développement s'est fondé 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.

3.2 Environnement de développement

Dans cette section, nous définissons l'environnement de développement Android Studio et les différentes APIs qui ont servis au développement de notre application mobile.

3.2.1 Android Studio

Android Studio est un environnement de développement (IDE - *Integrated Development Environment*) pour développer des applications sous la plateforme Android. Il est basé sur IntelliJ IDEA de JetBrains et permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément. En 2014, Android Studio passe de version bêta à une version stable 1.0. L'environnement devient alors conseillé par Google et l'environnement Eclipse se voit délaissé.

3.2.2 IntelliJ IDEA

IntelliJ IDEA est un IDE Java commercial développé par JetBrains¹. Il est fréquemment appelé par le simple nom *IntelliJ* ou *IDEA*. La première version d'IntelliJ IDEA est rendue publique en janvier 2001 et est, à ce moment-là, le seul IDE pour Java disponible disposant de possibilités de navigation avancée dans le code et de factorisation de code. En Décembre 2014, Google a annoncé

1. Entreprise informatique éditant des logiciels pour développeurs de logiciels.

la version 1.0 de Android Studio, un IDE open source pour les applications Android, basé sur l'open source Community Edition de IntelliJ. D'autres environnements de développement se basent sur IntelliJ dont AppCode, Clion, PhpStorm, PyCharm, RubyMine, WebStorm et MPS. Au nombre de ses fonctionnalités, IntelliJ apporte une étroite intégration avec quelques-uns des outils de développement libres les plus répandus tels que Git, CVS, Subversion, Ant et Maven, JUnit et TestNG. Enfin, IntelliJ permet de gérer plusieurs serveurs dont GlassFish, JBoss, Tomcat, Jetty, WebLogic, WebSphere et Geronimo.

3.2.3 Bibliothèques utilisées

Les bibliothèques auxquelles nous avons fait appel dans notre projet, sont toutes sous licence open source et implémentées en Java. Dans ce qui suit, nous énumérons les plus importantes ainsi que leur apport à notre implémentation des fonctionnalités du système.

- **Material design**

Introduit l'année dernière avec l'arrivée d'Android Lollipop, cette bibliothèque englobe à la fois des éléments de design et d'interface de la dernière version d'Android.

- **Google Maps Api**

Service permettant le géo-repérage virtuel grâce au GPS intégré de l'appareil ce qui permet de surveiller à distance la position et le déplacement d'un objet et de prendre des mesures si la position ou le déplacement s'écarte de certaines valeurs fixées d'avance.

- **ZXing API**

Projet multi-format de code-barres 1D/2D de traitement d'images mis en œuvre en Java. Ce projet met l'accent sur l'utilisation de la caméra intégrée sur les téléphones mobiles et de décoder les codes-barres sur l'appareil (QR codes), sans communiquer avec un serveur. En utilisant ce principe, nous avons mis en place un système d'identification. Cela a été réalisé à l'aide d'une génération de QR codes qui sont placés dans des endroits spécifiques, le personnel médical souhaitant se connecter sans introduire ses identifiants, n'a qu'à scanner le code collé sur son bureau, badge ou autres.

- **Twitter API**

Bibliothèque que nous avons utilisé pour charger l'interface de connexion de Twitter afin de permettre une connexion via un compte Twitter.

- **Facebook API**

Bibliothèque que nous avons utilisé pour charger l'interface de connexion de Facebook afin de permettre une connexion via un compte Facebook.

- **Sweet Alert**

Bibliothèque qui nous a permis d'afficher les alertes qui sont échangés entre les acteurs du secteur médical de notre système.

- **Picasso**

Bibliothèque que nous avons utilisé pour charger les différentes pièces-jointes (format images) à partir de notre serveur et les stocker dans un cache pour accélérer l'accès.

3.3 Implémentation de la base de données

Pour la création des tables de notre système, nous avons utilisé le langage SQL (*Structured Query Language*) qui est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles. Outre le langage de manipulation des données, la partie langage de définition des données permet de créer et de modifier l'organisation des données dans la base de données, la partie langage de contrôle de transaction quant à elle permet de commencer et de terminer des transactions, et la partie langage de contrôle des données permet d'autoriser ou d'interdire l'accès à des données à certaines personnes.

3.4 Présentation de l'application

Le volet technique de ce chapitre étant terminé, nous allons désormais consacrer cette partie du chapitre à la présentation des principales interfaces de notre application mobile.

3.4.1 Arborescence des vues de notre application

Pour illustrer le menu qu'offre notre application, la figure 3.1 présente graphiquement l'arborescence de ses fonctionnalités.

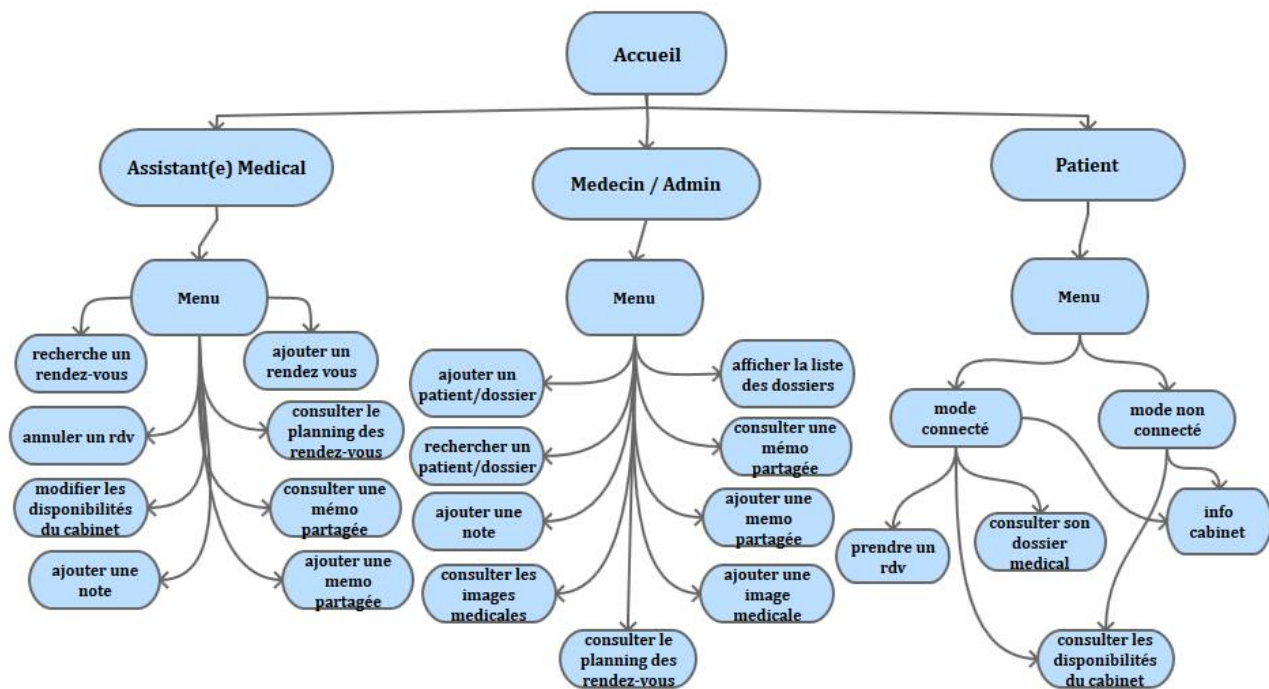


FIGURE 3.1 – Arborescence des vues de notre application.

3.4.2 Présentation des interfaces

Les interfaces de notre application s'adaptent aux dimensions écrans des différents équipements, mais de préférences à des écrans de 5' à 6'.

3.4.2.1 Scénario d'ajout d'un patient

La figure 3.2 représente l'interface du menu de connexion.

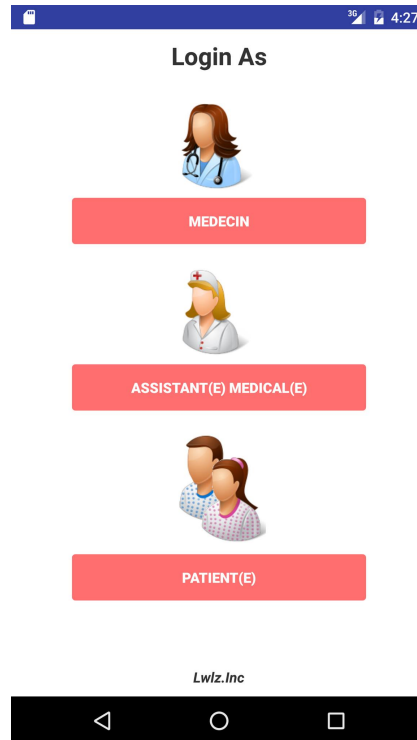


FIGURE 3.2 – Interface menu de connexion.

En cliquant sur la catégorie « Médecin », l'interface de connexion apparaît et deux possibilités se présentent ; une connexion traditionnelle via l'insertion du login et du mot de passe dans les champs requis ou bien le scan d'un QR code (voir figure 3.3).

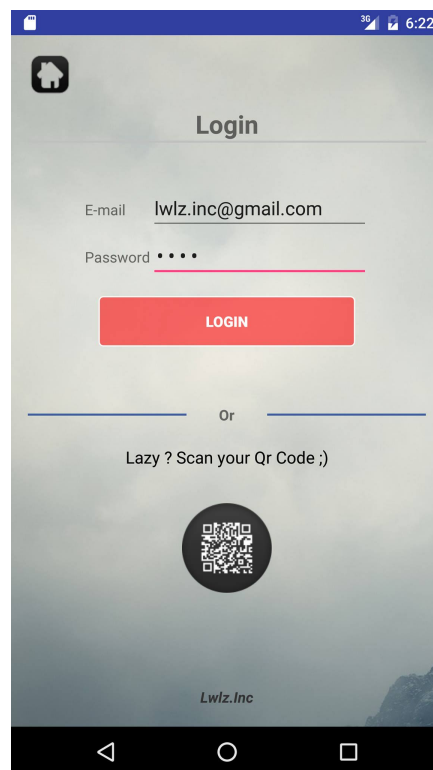


FIGURE 3.3 – Interface de connexion du médecin.

Après authentification, le menu apparaît proposant plusieurs possibilités et services au médecin authentifié comme illustré par la figure 3.4.

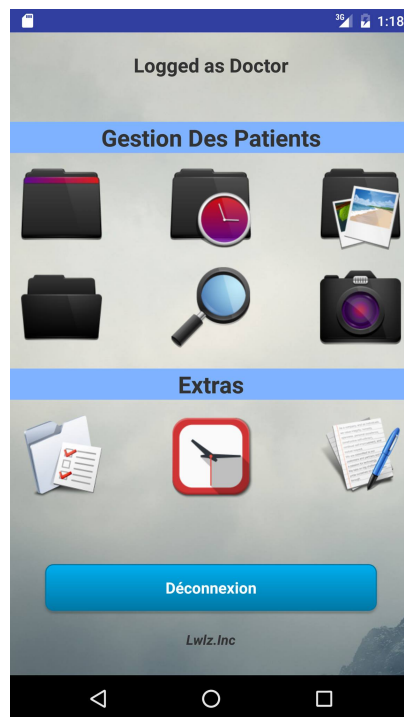


FIGURE 3.4 – Interface menu des services disponibles pour le médecin.

Après avoir cliqué sur la première icône de la deuxième ligne en partant de la gauche, le formulaire d'ajout d'un nouveau patient apparaît (voir figure 3.5).

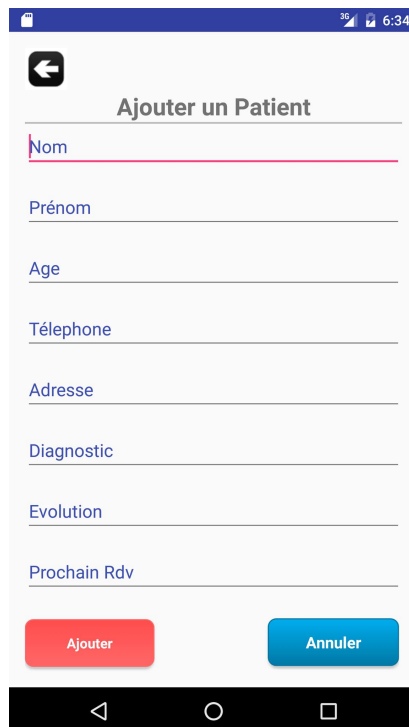


FIGURE 3.5 – Interface formulaire d'ajout d'un nouveau patient.

3.4.2.2 Disponibilités du cabinet médical

Après avoir cliqué sur l'icône de l'horloge en rouge (voir figure 3.4), l'interface des disponibilités du cabinet apparaît, le patient pourra alors consulter les horaires d'ouverture du cabinet comme illustré par la figure 3.6.

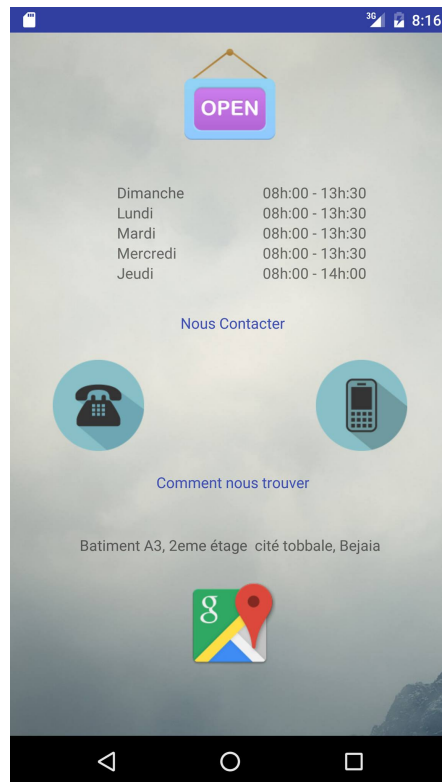


FIGURE 3.6 – Interface des disponibilités du cabinet.

3.4.2.3 Recherche d'un patient

En cliquant sur la loupe de la figure 3.4, nous aurons l'interface de recherche où le médecin pourra rechercher le dossier d'un patient spécifique (voir figure 3.7).

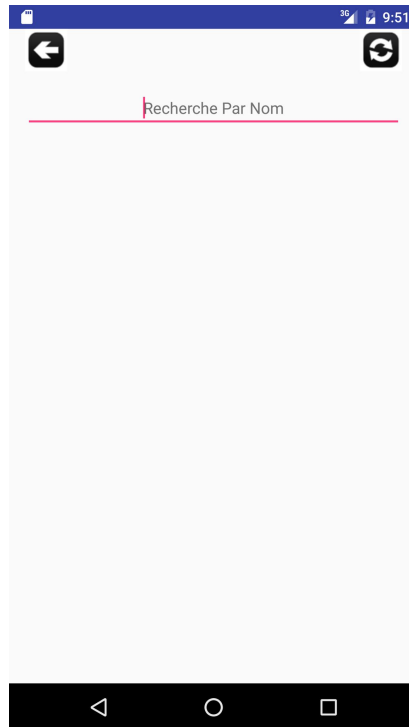


FIGURE 3.7 – Interface de recherche d'un patient.

3.4.2.4 Liste des dossiers médicaux

En cliquant sur la première icône de la première ligne (voir figure 3.4), nous obtiendrons l'interface contenant la liste des dossiers médicaux comme illustré par la figure 3.8.

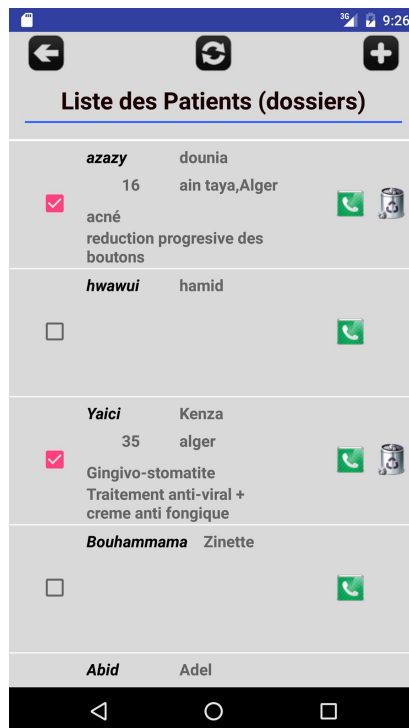


FIGURE 3.8 – Interface de la liste des dossiers médicaux.

3.4.2.5 Planning des rendez-vous

En cliquant sur la deuxième icône le planning des rendez-vous apparait (voir figure 3.9). Le médecin peut alors connaître le nombre de patients qu'il doit voir durant la journée.

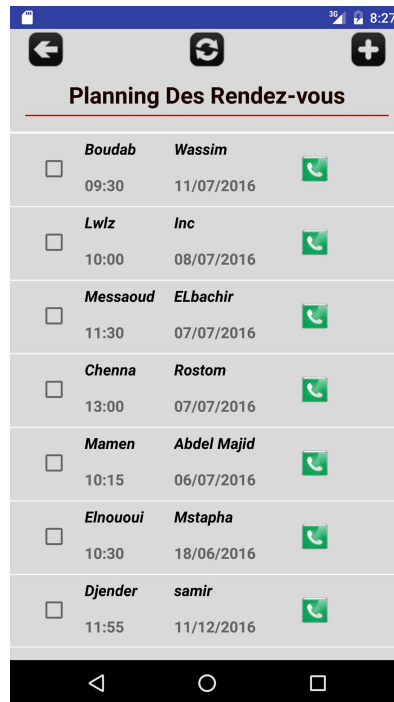


FIGURE 3.9 – Interface du planning des rendez-vous.

3.4.3 Sécurité

Plusieurs précautions ont été prises afin de sécuriser le système implémenté. Notre service de données prend en charge un mécanisme de sécurité très flexible pour restreindre l'accès aux objets stockés dans le serveur, en utilisant un principe d'autorisation et de rôles applicable aux utilisateurs. Une autorisation peut soit accorder ou rejeter une opération pour un actif particulier. Dans le cadre du service de données, l'actif est un objet que l'application peut récupérer, mettre à jour ou supprimer. Les autorisations peuvent être accordées ou rejetées à l'échelle globale, qui s'appliquent à toutes les tables et tous les objets dans la base de données. En outre, chaque table peut avoir sa propre matrice d'autorisation et politique de propriétés, une instruction spéciale précisant au propriétaires s'ils peuvent ou ne peuvent pas récupérer/modifier/supprimer les objets qui leurs appartiennent. Enfin, chaque objet a sa propre liste de contrôle d'accès (ACL - *Access Control List*) qui est une matrice des autorisations pour les opérations applicables spécifiquement à l'objet.

3.4.3.1 Protection contre les injections SQL et les failles XSS

Notre application contient beaucoup de formulaires, d'où la nécessité de faire des vérifications de chaque champ, en éliminant toute sorte de caractères qui peuvent nuire au système, ces vérifications se font avant de soumettre une requête au SGBD.

3.4.3.2 Protection contre les attaques CSRF

Pour sécuriser l'appel direct à notre API qui contient les différentes fonctions de notre système, le nom d'utilisateur et le mot de passe ne suffisent pas pour sécuriser ces appels, il faut alors utiliser un troisième argument qui est l'API key (token), qui se génère et s'affecte pour une session unique, cette mesure de sécurité, non seulement nous aide à sécuriser les appels mais aussi éliminé les attaques de type CSRF (Cross-Site Request Forgery).

3.5 Conclusion

Dans cette dernière partie de notre travail, nous avons présenté l'environnement de développement et les principaux outils utilisés, ainsi que les frameworks les plus importants, qui nous ont permis la réalisation de notre application en respectant les normes et les standards d'un bon système. Nous avons présenté également, notre application mobile à travers une arborescence des vues de cette dernière ainsi qu'un scénario d'exécution d'un ajout de patient.

Conclusion générale et perspectives

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. Puis, nous avons décrit les avantages qu'ont apportés ces applications au domaine médical notamment dans la gestion des dossiers patient. Dans la seconde partie de ce mémoire, nous avons recensé les principales fonctionnalités de l'application à réaliser. Nous avons ensuite présenté les différentes étapes de la conception à commencer 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 application mobile sous la plateforme Android et en mettant en œuvre notre base de données relationnelle avec le langage de requête structurée SQL. 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 de systèmes complexes.

Des perspectives d'amélioration de notre application restent toutefois indispensables. Nous envisageons ainsi d'ajouter de nouvelles fonctionnalités telles que la gestion de la comptabilité et de la caisse nationale d'assurance maladie au sein du cabinet médical.

Bibliographie

- [1] POIRIER F., *Étude sur les besoins de compétences dans le développement d'applications mobiles*, TechnoCompétences, Montréal, 2013.
- [2] GUILLET O., *Les différents types d'applications mobiles : natives, Web apps, hybrides, Flash*, <http://olivierguillet.com/2012/02/les-differents-types-dapplications-mobiles-natives-web-apps-hybrides-flash/>, Consulté le 20 Janvier 2016.
- [3] ROUINI H., *Introduction aux systèmes d'exploitation mobiles*, 2013.
- [4] <https://fr.wikipedia.org/wiki/Android>, Consulté le 25 Janvier 2016.
- [5] [https://fr.wikipedia.org/wiki/IOS_\(Apple\)](https://fr.wikipedia.org/wiki/IOS_(Apple)), Consulté le 25 Janvier 2016.
- [6] MEZIANE A., *Le dossier médical partagé*, Bulletin d'information trimestriel du CERIST, pp. 10-19, 2012.
- [7] <http://www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.htm>, Consulté le 02 Février 2016.
- [8] ROQUES P., *UML 2 par la pratique*, EYROLLES, 7^e édition, 2009.
- [9] ROQUES P., *UML 2 Modéliser une Application Web*, EYROLLES, 4^e édition, 2008.
- [10] ROQUES P. & VALLÉE F., *UML 2 en action : de l'analyse des besoins à la conception*, EYROLLES, 2011.

Résumé

Ce mémoire de fin de cycle présente la conception et la réalisation d'une application mobile pour le suivi d'un cabinet médical de dermatologie. Le système de santé Algérien se modernise de jour en jour et se retrouve dans la nécessité d'informatiser l'information médicale. En effet la complexité croissante de la médecine occidentale pousse de manière naturelle à la mise en place de systèmes d'information étant capable d'aider le praticien dans ses tâches quotidiennes. Afin d'atteindre cet objectif, il nous a été proposé de concevoir et réaliser une application mobile sous Android assurant la gestion et le suivi des patients tout en proposant diverses options et fonctionnalités comme la prise de rendez-vous en ligne ou la consultation de son dossier médical, permettant ainsi d'améliorer la collaboration entre les acteurs dans le secteur médical afin d'offrir un meilleur rendement et de meilleurs prestations. Pour ce faire, nous avons choisi de modéliser notre système avec le formalisme UML. Notre choix s'est porté sur ce dernier en raison de sa simplicité, sa performance et son adaptation en matière de conception. Afin de réaliser notre application mobile, nous avons utilisé le langage de programmation JAVA sous l'environnement de développement Android Studio, et IntelliJ IDEA, quant à l'implémentation de notre base de données nous l'avons hébergé sur un serveur privé en utilisant le langage SQL.

Mots clés : Suivi d'un cabinet médical, DMP, DMPI, Android, Système d'information médical

Abstract

This document presents the design and implementation of a mobile application for managing a medical practice of Dermatology. The Algerian health system is modernized day by day and is reflected in the need to computerize medical information. Indeed the growing complexity of western medicine grows naturally in the implementation of information systems being able to assist the practitioner in his daily tasks. To achieve this goal, we propose to design and build a mobile application on Android ensuring the management and tracking of patients while offering various options and features such as making appointments online or consulting his medical record, enabling improved collaboration between stakeholders in the health sector to provide better performance and benefits. To do this, we chose to model our system with the UML formalism. Our choice fell on it because of its simplicity, performance and adaptation in design. To achieve our mobile application, we used the Java programming language in Android Studio development environment, IntelliJ IDEA and, for the implementation of our database we have hosted on a private server using the SQL.

Keywords : Managing a medical office, PHR, CPR, Android, Medical information system