

N° Réf :.....

Centre Universitaire
Abdelhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

Spécialité: Sciences et Technologies de l'Information et de la
Communication (STIC)

*Conception et réalisation d'une application
mobile pour le suivie des diabétiques*

Préparé par :

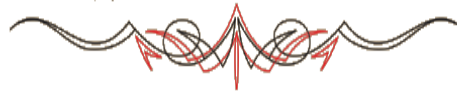
✚ Benabderrahmene Imane.
✚ Djeddar Somia.

Soutenue devant le jury :

Encadrer par :	Hedjaz Sabrina	M.A.A	C.U.Abd Elhafid Boussouf.
Président :	Aouag Mouna	M.C.B	C.U.Abd Elhafid Boussouf.
Examineur :	Boufaghes Hamida	M.A.A	C.U.Abd Elhafid Boussouf.

Année Universitaire : 2018/2019

REMERCIEMENTS



En préambule à ce mémoire nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apportées leurs aide et qui ont contribuées à l'élaboration ainsi qu'à la réussite de cette année universitaire.

Nous tenons à remercier sincèrement Mme Hedjaz Sabrina , qui en tant qu'encadreur, s'est toujours montrée à l'écoute et très disponible tout au long de la rédaction, sans oublier les membres du jury qui évaluera notre travail.

Anisi que les enseignants qui ont contribués à notre formation et appuyés notre cursus universitaire, et le personnel administratif du département d'informatique.

Enfin, nous adressons nos plus sincères remerciements à nos familles et ami(e)s, pour leurs soutien et encouragements tout au long de la réalisation de ce projet car ils nous ont toujours supportés moralement et financièrement pendant toutes nos années d'études.

Merci à tous et à toutes

dédicace

A MA TRÈS CHÈRE MÈRE : DALILA

Autant de phrases aussi expressives soient-elles ne sauraient montrer le degré d'amour et d'affection que j'éprouve pour toi. Tu m'as comblé avec ta tendresse et affection tout au long de mon parcours. Tu n'as cessé de me soutenir et de m'encourager durant toutes les années de mes études, tu as toujours été présente à mes côtés pour me consoler quand il fallait. En ce jour mémorable, pour moi ainsi que pour toi, reçoit ce travail en signe de ma vive reconnaissance et mon profond estime. Puisse le tout puissant te donner santé, bonheur et longue vie afin que je puisse te combler à mon tour.

A MON TRÈS CHER PÈRE : ABDESSELAM

Autant de phrases et d'expressions aussi éloquentes soit-elles ne sauraient exprimer ma gratitude et ma reconnaissance. Tu as su m'inculquer le sens de la responsabilité, de l'optimisme et de la confiance en soi face aux difficultés de la vie. Tes conseils ont toujours guidé mes pas vers la réussite. Ta patience sans fin, ta compréhension et ton encouragement sont pour moi le soutien indispensable que tu as toujours su m'apporter. Je te dois ce que je suis aujourd'hui et ce que je serai demain et je ferai toujours de mon mieux pour rester ta fierté et ne jamais te décevoir. que Dieu le tout puissant te préserve, t'accorde santé, bonheur, quiétude de l'esprit et te protège de tout mal.

A MON TRÈS CHER FRÈRE : HOUSSAM EDDINE

En souvenir d'une enfance dont nous avons partagé les meilleurs et les plus agréables moments. Pour toute la complicité et l'entente qui nous unissent, ce travail est un témoignage de mon attachement et de mon amour.

A MA TRÈS CHÈRE SOEUR : DOUAA

Pour toute l'ambiance dont tu m'as entouré, pour toute la spontanéité et ton élan chaleureux, Je te dédie ce travail . Puisse Dieu le tout puissant exhausser tous tes vœux.

A TOUTE MA FAMILLE

à tous ceux qui ont su m'aimer, me supporter je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite. .

A MES AMIS

qui m'ont énormément aidé et pour qui je témoigne mon affection et ma profonde reconnaissance.

Soumaya

dédicace

À MES CHÈRES PARENTS

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.

Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez.

Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.

À MES CHÈRES ET ADORABLE FRÈRES ET SŒURS

*Mon très cher frère Ayyoub et mes chères soeurs Yasmina, Housna ,
Rim et sara.*

*En témoignage de mon affection fraternelle, de ma profonde tendresse
et reconnaissance, je vous souhaite une vie pleine de bonheur et de
succès et que Dieu, le tout puissant, vous protège et vous garde.*

À MES CHÈRES PETITS NEVEUX ET NIECES

*Israa, Lokman, Sirine, Nada Malak, Abde Raouf, Mohammed Kacim,
Mohammed Mostafa Yanis et Iline.*

*Aucune dédicace ne saurait exprimer tout l'amour que j'ai pour vous,
Votre joie et votre gaieté me comblent de bonheur.*

*Puisse Dieu vous garder, éclairer votre route et vous aider à réaliser à
votre tour vos vœux les plus chers.*

À MA FAMILLE ET MES AMIS

*A toute ma famille, mes amis, à tous ceux qui ont su m'aimer, me
supporter je vous dédie ce travail avec tous mes vœux de bonheur, de
santé et de réussite*

Imene mimi

Résumé

Les dispositifs mobiles sont des outils actuellement utilisés dans plusieurs domaines pour faciliter les tâches quotidiennes, nous avons utilisé ces derniers afin d'automatiser le suivi des diabétiques.

Pour mettre en œuvre notre solution, nous avons utilisé un processus de développement appelé 2TUP, qui se base sur UML comme langage de modélisation conçu pour fournir une méthode normalisée pour la conception, et la construction des documents nécessaires au bon développement.

Cette conception est mise en œuvre sous l'environnement de développement Android Studio avec l'utilisation du framework laravel pour l'application web, java, XML et php comme langages de programmation, un système de gestion de base de données MySQL.

Mots clés : Glycémie, Diabète, Application Mobile, UML, 2TUP, MySQL, PHP, JAVA, Android, Laravel.

Abstract

Mobile devices are tools currently used in several areas to facilitate daily tasks, we have used them to automate the tracking of diabetics.

To apply our solution, we have used a development process called 2TUP, which is based on UML as a modeling language designed to provide a standardized method for design, and the construction of the necessary documents for proper development.

This design is executed under the Android Studio development environment with the use of the Laravel framework for web application, java, XML and php as programming languages, a MySQL database management system.

Keywords : Glycemie, Diabetes, Mobile Application, UML, 2TUP, MySQL, PHP, JAVA, Android, Laravel.

Liste des Acronymes

TIC	Technologies de l'information et de la communication.
IOS	IPhone Operating System.
UML	Unified Modeling Language.
OS	Operating System.
WIFI	Wireless Fidelity.
SDK	Software Developement Kit.
ADB	Android Debug Bridge.
IDE	Integrated Development Environment.
OMS	Organisation mondiale de la santé.
NTIC	Nouvelles Technologie d'Information et de Communication.
EPH	Établissement Public Hospitalier.
2TUP	2 tracks unified process .
SGBDR	Système de Gestion de Bases de Données Relationnelles.
MYSQL	My Structured Query Language.
XML	eXtensible Markup Language .
PHP	Hypertext Preprocessor .
HTML	HyperText Markup Language.
W3C	World Wide Web Consortium.
BDD	Base de données.
SQLITE	Structured Query Language.
SQL	Structured Query Language.
WAN	Wide Area Network.
SGBD	System de Gestion de Base de Données.

Table des matières

Table des matières	i
Liste des figures	v
Liste de table	x
Introduction Générale	2
1 Introduction aux applications mobiles(Android)	6
1.1 Systèmes d'exploitations mobiles	6
1.1.1 Définition	6
1.1.2 Liste des systèmes d'exploitations mobiles	7
1.1.2.1 IOS	7
1.1.2.2 Symbian	7
1.1.2.3 Windows Mobile	7
1.1.2.4 Blackberry	8
1.1.3 Android	8
1.1.4 Part de marché	9
1.2 les applications mobiles	10
1.2.1 Définition	10
1.2.2 Cycle de vie d'une activité d'une application mobile	10
1.3 Découverte de la plateforme Android	11
1.3.1 Présentation d'Android	11
1.3.2 Les versions d'Android	11
1.3.3 Répartition des versions Android	12

1.3.4	l'architecture d'android	12
1.3.5	Les avantages d'Android[22]	14
1.3.6	Les outils de développement Android	15
2	Etude préliminaire	16
2.1	La télémédecine	16
2.1.1	Définition	16
2.1.2	Les actes de la télémédecine	17
2.1.2.1	La téléconsultation	17
2.1.2.2	La télé-expertise	17
2.1.2.3	La télésurveillance	18
2.1.2.4	La téléassistance	18
2.1.3	Intérêt de la télémédecine	19
2.2	Le diabète	19
2.2.1	Définition	19
2.2.2	Les symptômes	20
2.2.3	Les types de diabète	20
2.2.4	Les complications du diabète	21
2.3	La télémédecine et le diabète	22
2.4	Mesure de la Glycémie	22
2.5	Présentation de l'entreprise du lieu de stage	23
2.5.1	Identification de l'établissement	23
2.5.2	Localisation de l'établissement	23
2.5.3	Fiche technique	24
2.5.4	Organigramme d'accueil	25
2.6	Contribution	25
2.7	Elaboration du cahier de charge	26
2.7.1	Grands choix techniques	26
2.7.2	Le processus 2TUP	26
2.8	Description de contexte du système	27
2.8.1	Identification des acteurs	27
2.8.2	Identification des messages	28
2.8.3	Modélisation de contexte	29

2.8.3.1	Diagramme de contexte dynamique	29
2.8.3.2	Description détaillée des messages	30
3	Capture des besoins fonctionnels et techniques	32
3.1	Capture des besoins fonctionnels	32
3.1.1	Identification des cas d'utilisations	33
3.1.1.1	Liste préliminaire des cas d'utilisations	33
3.1.1.2	Description préliminaire des différents cas d'utilisation	34
3.1.1.3	Le diagramme de cas d'utilisation pour les besoins fonctionnels	37
3.1.2	Description détaillée des différents cas d'utilisation	39
3.1.3	Diagrammes d'activité des différents cas d'utilisation	49
3.1.4	Identification des classes candidates	58
3.1.5	Diagrammes de classe participante des cas d'utilisations	61
3.2	Capture des besoins techniques	74
3.2.1	Définition	74
3.2.2	Spécification techniques du point de vue matériel	75
3.2.3	Spécification d'architecture	75
3.2.3.1	Spécification d'organisation du modèle déploiement . .	77
3.2.4	Elaboration du modèle de spécification logicielle	77
3.2.4.1	Modèle de spécification logicielle	78
3.2.4.2	Description textuelle des cas d'utilisation technique . .	78
3.2.5	Organisation en couche du modèle de spécification	79
4	L'analyse	82
4.1	Découpage en catégories	82
4.1.1	Répartition des classes candidates en catégories	83
4.1.2	Elaboration des diagrammes de classe préliminaire pour chaque catégorie	83
4.1.3	Dépendance entre les catégories	84
4.2	Développement du modèle statique	84
4.2.1	Affiner les classes	85
4.2.2	Affiner les associations	85

4.2.3	Ajouter les attributs	85
4.2.4	Ajouter les opérations	85
4.2.5	Diagramme de classe détaillé pour chaque catégorie	86
4.3	Développement du modèle dynamique	87
4.3.1	Diagramme de séquence détaillé	87
5	Conception	101
5.1	Conception préliminaire	101
5.1.1	Définition	101
5.1.2	Développement du modèle de déploiement	102
5.1.2.1	Définition du poste de travail	102
5.1.2.2	Le modèle de déploiement	102
5.1.2.3	Architectures adoptée	102
5.2	Conception détaillée	103
5.2.1	Définition	103
5.2.2	Dictionnaire de données	103
5.2.2.1	Concevoir les attributs et les classes	103
5.3	Diagramme de classe détaillé	105
5.4	Le modèle relationnel	106
5.5	Les règles de passage	106
5.6	Les règles de gestion	107
5.7	Les tables de la base de données	107
6	L'implémentation	109
6.1	Langages de développement	109
6.1.1	JAVA	109
6.1.2	XML	110
6.1.3	PHP	110
6.2	Les bases de données	110
6.2.1	Les bases de données MySQL	112
6.3	Outils de développement	112
6.3.1	Android Studio	112
6.3.2	Le framework Laravel	113

6.3.2.1	Le Framework	113
6.3.2.2	L'architecture MVC (Modèle-Vue-Contrôleur)	113
6.3.2.3	Les avantages du modèle MVC	113
6.3.2.4	Laravel	114
6.3.2.5	Les statistiques de part de marché pour les Frameworks PHP en 2019	115
6.3.3	WampServer	115
6.3.4	Pacestar UML Diagrammer	115
6.4	Présentation des interface de notre application	116
Conclusion générale et perspectives		122

Liste des figures

1.1	logo IOS	7
1.2	logo symbian	7
1.3	windows phone	8
1.4	logo blackberry	8
1.5	logo Android	9
1.6	Part de marché mondiale des systèmes d'exploitation mobiles	9
1.7	Part de marché mondiale des OS	9
1.8	cycle de vie d'une activité[3]	10
1.9	Les versions d'Android	12
1.10	La répartition des versions d'Android[21]	12
1.11	Architecture Android[6]	13
2.1	La téléconsultation [8]	17
2.2	La télésurveillance [9]	18
2.3	La télésurveillance [9]	18
2.4	La téléassistance [10]	19
2.5	Localisation de l'établissement[7]	24
2.6	organigramme de l'établissement	25
2.7	Processus de développement en Y[13]	27
2.8	Le diagramme de contexte dynamique du système	30
3.1	Diagramme de cas d'utilisation globale	38
3.2	Représentation d'un diagramme de séquence[11]	39
3.3	Cas d'utilisation « ajouter valeur glycémie »	40

3.4	Cas d'utilisation « annuler rendez-vous »	41
3.5	Cas d'utilisation « prendre rendez-vous »	42
3.6	Cas d'utilisation « modifier les dossiers medicaux »	43
3.7	Cas d'utilisation « créer profil assistant »	44
3.8	Cas d'utilisation « créer profil patient »	45
3.9	Cas d'utilisation « modifier valeur glycémie »	46
3.10	Cas d'utilisation « bloquer profil »	47
3.11	Cas d'utilisation « modifier profil utilisateur »	48
3.12	Cas d'utilisation « s'authentifier »	49
3.13	Cas d'utilisation « ajouter valeur glycémie »	50
3.14	Cas d'utilisation « créer profil médecin »	50
3.15	Cas d'utilisation « annuler rendez-vous »	51
3.16	Cas d'utilisation « consulter son profil »	51
3.17	Cas d'utilisation « prendre rendez-vous »	52
3.18	Cas d'utilisation « consulter les profils utilisateurs »	52
3.19	Cas d'utilisation « modifier les dossiers médicaux »	53
3.20	Cas d'utilisation « confirmer rendez-vous »	53
3.21	Cas d'utilisation « créer profil assistant »	54
3.22	Cas d'utilisation « saisir analyses 3 mois »	54
3.23	Cas d'utilisation « créer profil patient »	55
3.24	Cas d'utilisation « consulter planning des rendez-vous »	55
3.25	Cas d'utilisation « modifier valeur glycémie »	56
3.26	Cas d'utilisation « consulter les dossiers médicaux »	56
3.27	Cas d'utilisation « bloquer profil »	57
3.28	Cas d'utilisation « modifier son profil »	57
3.29	Cas d'utilisation « modifier les profils utilisateurs »	58
3.30	Cas d'utilisation « consulter son dossier médicale »	58
3.31	Les classes candidates et ses responsabilités	61
3.32	Cas d'utilisation « s'authentifier »	62
3.33	Cas d'utilisation « ajouter valeur glycémie »	62
3.34	Cas d'utilisation « créer profil médecin »	63
3.35	Cas d'utilisation « annuler rendez-vous »	63

3.36	Cas d'utilisation « consulter son profil »	64
3.37	Cas d'utilisation « prendre rendez-vous »	64
3.38	Cas d'utilisation « consulter profils(médecins) »	65
3.39	Cas d'utilisation « consulter profils(patients) »	65
3.40	Cas d'utilisation « consulter profils(assistants) »	66
3.41	Cas d'utilisation « modifier les dossiers médicaux »	66
3.42	Cas d'utilisation « confirmer rendez-vous »	67
3.43	Cas d'utilisation « créer profil assistant »	67
3.44	Cas d'utilisation « saisir analyses 3 mois »	68
3.45	Cas d'utilisation « créer profil patient »	68
3.46	Cas d'utilisation « consulter planning des rendez-vous »	69
3.47	Cas d'utilisation « modifier valeur glycémie »	69
3.48	Cas d'utilisation « consulter les dossiers médicaux »	70
3.49	Cas d'utilisation « bloquer profil (médecins)»	70
3.50	Cas d'utilisation « bloquer profil(patients) »	71
3.51	Cas d'utilisation « bloquer profil (assistants) »	71
3.52	Cas d'utilisation « modifier son profil »	72
3.53	Cas d'utilisation « modifier profils (médecins)»	72
3.54	Cas d'utilisation « modifier profils(patients) »	73
3.55	Cas d'utilisation « modifier profils(assistants) »	73
3.56	Cas d'utilisation « consulter son dossier médicale »	74
3.57	Configuration matérielle de système	75
3.58	Architecture 2 tiers de notre système[20]	76
3.59	Le modèle de déploiement du système	77
3.60	Modèle de spécification logicielle initial	78
3.61	Description textuelle des cas d'utilisation techniques	79
3.62	Organisation du modèle de spécification logicielle(application mobile) .	80
3.63	Organisation du modèle de spécification logicielle(application web) . . .	80
4.1	Découpage en catégories	83
4.2	Les diagrammes de classe préliminaire du catégorie(utilisateur)	83
4.3	Les diagrammes de classe préliminaire du catégorie(patient)	84
4.4	Modèle structurel d'analyse	84

4.5	Le diagrammes de classe détaillé de catégorie utilisateur	86
4.6	Le diagramme de classe détaillé de catégorie patient	86
4.7	Cas d'utilisation « s'authentifier »	87
4.8	Cas d'utilisation « ajouter valeur glycémie »	88
4.9	Cas d'utilisation « créer profile médecin »	88
4.10	Cas d'utilisation « annuler rendez-vous »	89
4.11	Cas d'utilisation « consulter son profile »	89
4.12	Cas d'utilisation « prendre rendez-vous »	90
4.13	Cas d'utilisation « consulter profiles (médecins) »	90
4.14	Cas d'utilisation « consulter profiles (patients) »	91
4.15	Cas d'utilisation « consulter profiles(assistants)»	91
4.16	Cas d'utilisation « modifier les dossiers medicaux »	92
4.17	Cas d'utilisation « confirmer rendez-vous »	92
4.18	Cas d'utilisation « créer profile assistant »	93
4.19	Cas d'utilisation « saisir analyses 3 mois »	93
4.20	Cas d'utilisation « créer profile patient »	94
4.21	Cas d'utilisation « consulter planning des rendez-vous »	94
4.22	Cas d'utilisation « modifier valeur glycémie »	95
4.23	Cas d'utilisation « consulter les dossiers médicaux »	95
4.24	Cas d'utilisation « bloquer profiles (médecins)»	96
4.25	Cas d'utilisation « bloquer profiles (patients) »	96
4.26	Cas d'utilisation « bloquer profile (assistants »	97
4.27	Cas d'utilisation « modifier son profile »	97
4.28	Cas d'utilisation « modifier profiles (médecins) »	98
4.29	Cas d'utilisation « modifier profiles (patients) »	98
4.30	Cas d'utilisation « modifier profiles (assistants) »	99
4.31	Cas d'utilisation « consulter son dossier médicale »	99
5.1	Le modèle de déploiement du système(application web)	102
5.2	diagramme de classes détaillé de notre système	105
6.1	Logo java [16]	110
6.2	Logo php [17]	110

6.3	Interface class JAVA	111
6.4	Interface fichier PHP	111
6.5	L'environnement Android Studio	112
6.6	L'architecture MVC [25]	113
6.7	Le Framework Laravel	114
6.8	Les statistiques de parts de marché pour les Frameworks PHP en 2019[26]	115
6.9	Logo wamp server [18]	115
6.10	Logo Pacestar UML Diagrammer [19]	116
6.11	Login	116
6.12	Espace administrateur	116
6.13	Interface ajouter médecin	117
6.14	Ajouter dossier médicale	117
6.15	Graphe des valeurs glycémies	117
6.16	Dossier médicale d'un patient	118
6.17	Planning des rendez-vous	118
6.18	Liste des assistants	118
6.19	Login	119
6.20	Accueil	119
6.21	Ajouter valeur	119
6.22	Profile	119
6.23	Historique	120
6.24	Prendre rendez-vous	120
6.25	pour sélectionner la date de rendez-vous	120
6.26	pour sélectionner l'heure de rendez-vous	120
6.27	Géolocalisation de l'hopitale	121

Liste des tableaux

2.1	Valeurs normales de la glycémie	23
2.2	Fiche technique.	24
2.3	Légende des messages de diagramme de contexte	31
3.1	Liste des acteurs et des messages par cas d'utilisation	34
3.2	Cas d'utilisation « s'authentifier »	40
3.3	Cas d'utilisation « créer profil médecin »	41
3.4	Cas d'utilisation « consulter son profil »	42
3.5	Cas d'utilisation « consulter les profils utilisateurs »	43
3.6	Cas d'utilisation « confirmer rendez-vous »	44
3.7	Cas d'utilisation « saisir analyses 3 mois »	45
3.8	Cas d'utilisation « consulter planning des rendez-vous »	46
3.9	Cas d'utilisation « consulter les dossiers médicaux »	47
3.10	Cas d'utilisation « modifier son profil »	48
3.11	Cas d'utilisation « consulter son dossier médicale »	49
5.1	Dictionnaire de données avec les classes et les attributs	104

Introduction générale

La télémédecine ne vient pas remplacer les pratiques médicales actuelles mais permet de mettre sur pied une solution aux défis auxquels est confrontée l'offre de soins aujourd'hui. C'est une discipline marquante du domaine médical qui exploite les nouvelles technologies de l'information (TIC), afin d'améliorer l'accessibilité aux soins de santé en ne déplaçant que les données plutôt que les patients, et ce, à l'aide de transfert de données à distance.

De nos jours, la télémédecine prend un grand envol avec l'évolution du téléphone portable qui a carrément révolutionné la médecine à distance. Porté par le succès des plateformes IOS et Android, le développement mobile connaît une très forte expansion. Des applications de tous types fleurissent, loisirs, media sociaux, boutiques en ligne, gestion, médecine, etc. Les smartphones et les tablettes ont envahi le quotidien des consommateurs, en témoigne le nombre de téléchargement d'application mobile en perpétuelle hausse.

Grace à son succès, le Smartphone est vite devenu indispensable dans une chaîne de télémédecine. Cet appareil miniature, puissant, pouvant se connecter presque à tous les réseaux, est désormais jugé comme étant un outil idyllique pour plusieurs applications destinées au suivi de certaines maladies chroniques telle que les maladies cardiaques et le Diabète.

Notre choix s'est porté sur la maladie du diabète, qui est devenu un problème de santé à l'échelle mondiale, il ne cesse de s'accroître. Certains le qualifient de « La

maladie du XXI siècle ».

Problématiques

Le diabète est une maladie chronique nécessitant un suivi permanent. La plupart du temps, ce sont les patients eux-mêmes qui s'administrent leur traitement. Ils doivent également noter quotidiennement leur niveau de glycémie, au rythme des repas et de l'activité physique, ainsi que leur prise de médicaments, afin que leur médecin traitant soit en mesure d'adapter le traitement. Parfois, Les patients et les médecins font faces à des difficultés :

- Côté patient :
 - Les patients peuvent oublier de mesurer quotidiennement leurs niveau de glycémie sur un carnet.
 - Les patients ont des difficultés à se déplacer.
 - Les médecins ne peuvent pas gérer les carnets de leurs patients.
 - Les patients trouvent des difficultés à prendre les rendez-vous chez le médecin.
- Côté médecin :
 - Les médecins ne peuvent pas consulter le grand nombre des patients.
- Côté assistant :
 - la difficulté de l'organisation du planning des rendez-vous.

L'objectif de notre travail consiste à concevoir et à développer une application mobile sous Android permettant à un utilisateur via son smartphone de suivre son diabète au quotidien en notant et en enregistrant toutes les valeurs prises qui pourront être consultée par son médecin traitant et établir un diagnostic à distance. Le présent mémoire est divisé en six chapitres :

On commence par **Le premier s'intitule « Introduction à Android »** : il a pour objectif de présenter les différentes notions et concepts liés au monde du mobile, en

l'occurrence les dispositifs matériels et les systèmes d'exploitation mobiles, nous allons évoquer aussi le système d'exploitation mobile « Android », ses caractéristiques et son architecture. Nous terminerons avec une présentation des outils de développement Android.

Dans le deuxième chapitre « Etude préliminaire » : nous allons faire le tour d'horizon d'une nouvelle technologie qui est la télémédecine, nous allons aussi aborder la maladie du diabète pour laquelle sera appliquée notre solution de télésurveillance en exploitant les Smartphones. De plus, il vise à effectuer un premier repérage des besoins fonctionnels et opérationnels, en donnant une version textuelle préliminaire du cahier de charge. Après avoir identifié les acteurs qui interagiront avec le système, nous développerons un premier modèle UML de niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du système.

Le troisième chapitre « Capture des besoins fonctionnels et technique » : ce chapitre comporte deux étapes : la capture des besoins fonctionnels et celle des besoins techniques. La phase de capture des besoins fonctionnels formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire, en donnant une description textuelle et une autre graphique pour chaque cas d'utilisation. La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative.

Le quatrième chapitre « Analyse » : ce chapitre comporte les étapes de découpage en catégories, de développement du modèle statique et développement du modèle dynamique. Le découpage en catégories consiste à découper le modèle UML en blocs logiques les plus indépendants possibles. Le développement du modèle statique va nous permettre d'illustrer les principales constructions du diagramme de classe UML durant l'étape d'analyse. Le développement du modèle dynamique va nous permettre d'illustrer comment décrire des scénarios mettant en jeu un ensemble d'objets échangeant des messages.

Le cinquième chapitre « Conception » : ce chapitre comporte la conception préliminaire et la conception détaillée. La conception préliminaire où s'effectue la fusion

des études fonctionnelles et techniques. La conception détaillée consiste à construire et à documenter précisément les classes, les interfaces, les tables et les méthodes qui constituent le codage de la solution.

Le sixième chapitre « Implémentation » : dans ce chapitre on donne une description de l'application, les technologies de programmation utilisées, ainsi que les interfaces graphiques de l'application.

La conclusion générale : résume les résultats de notre travail, et présente les perspectives que nous souhaitons réaliser dans le futur.

Chapitre 1

Introduction aux applications mobiles(Android)

Introduction

Dans le monde actuel, les smartphones et les tablettes font partis de notre vie quotidienne, en la rendant plus facile et commode, ils sont équipés d'applications qui sont conçues pour des plateformes mobiles et utilisés pour des services d'information, de medias sociaux, de jeux, etc.

Un smartphone est doté d'un système d'exploitation et cela varie selon sa marque, celui qui est le plus répandu sur le marché est « Android » (grâce à son open source). Dans ce chapitre nous nous intéressons au développement mobile sous Android , en détaillant son historique, ses caractéristiques ainsi que son architecture, ses avantages et enfin la concurrence sur le marché.[1]

1.1 Systèmes d'exploitations mobiles

1.1.1 Définition

Tout comme un ordinateur dispose d'un système d'exploitation, les téléphones mobiles se composent également d'une plate-forme qui contrôle toutes ses fonctionnalités. Ceci est connu comme un système d'exploitation mobile. Généralement connu sous le nom d'OS mobile, il s'agit d'un système d'exploitation qui exploite un appareil mobile

(smartphone, tablette, etc.). Il contrôle toutes les opérations de base du téléphone mobile comme option d'écran tactile, cellulaires, Bluetooth, Wifi, appareil photo, lecteur de musique et d'autres fonctionnalités.

Parmi les système d'exploitation les plus populaires : [4]

1.1.2 Liste des systèmes d'exploitations mobiles

1.1.2.1 IOS

IOS est le système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod touch, et iPad. Il est dérivé de Mac OSx dont il partage les fondations (le Keren hybride XNU basé sur le micronoyau Mach, les services Unix et Cocoa, etc.). IOS comporte quatre couches d'abstraction, similaires à celles de Mac OSx : une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa ».



Figure 1.1 – logo IOS

1.1.2.2 Symbian

Symbian OS a été développé par la société Symbian, il comprend trois types de couche de l'interface de l'utilisateur (UI layer ou middleware du SE Symbian), nous distinguons : La couche S60 développée par Nokia, UIQ par Sonny Ericsson et MOAP par NTT DoCoMo.



Figure 1.2 – logo symbian

1.1.2.3 Windows Mobile

Windows Phone, challenger du marché apparu en 2010, a su s'inspirer de ses concurrents et de son expérience passée dans le domaine mobile tout en innovant. Windows

propose une interface simple et épurée ainsi que de multiples possibilités de personnalisation, il permet d'arriver rapidement à l'exécution d'une tâche ce qui a particulièrement séduit les utilisateurs. Il est considéré comme l'OS mobile le plus adapté aux utilisateurs néophytes.



Figure 1.3 – windows phone

1.1.2.4 Blackberry

BlackBerry a été créé par Research In Motion (RIM). BlackBerry fournit du service électronique pour les entreprises en utilisant BlackBerry Enterprise Server. Le fournisseur offre aux abonnés des fonctions push de messagerie et d'accès mobile pratique avec du contenu Internet pour leurs terminaux. Il comporte aussi la technologie de la pièce jointe qui supporte divers types de pièces jointes telles que les fichiers d'extensions .zip, .html, .doc, .dot, .ppt, .PDF, etc. Pour toutes ces précédentes raisons que son service de messagerie électronique est meilleur que les autres plateformes.



Figure 1.4 – logo blackberry

1.1.3 Android

Android est un SE basé sur Linux conçu principalement pour équipements à écran tactile comme les Smartphones et les tablettes. Android est devenu la plateforme plus populaire pour mobile avec plus de 600000 applications sur le marché. Il est prévu que le store d'Android atteigne le chiffre d'un million d'applications bien avant l'Apple store. Comme c'est la plateforme sur laquelle on va travailler dans ce projet, nous allons revoir cette plateforme plus tard en détails.



Figure 1.5 – logo Android

1.1.4 Part de marché

Chaque système d'exploitation embarqué et mentionné ci-dessous occupe une part du marché dans le monde en 2019 suivant le nombre de téléphone mobile des utilisateurs. Nous remarquons qu'Android OS possède une part de marché importante. En effet, il est classé à la première place d'où l'intérêt de notre application sur cette plateforme mobile.[23]

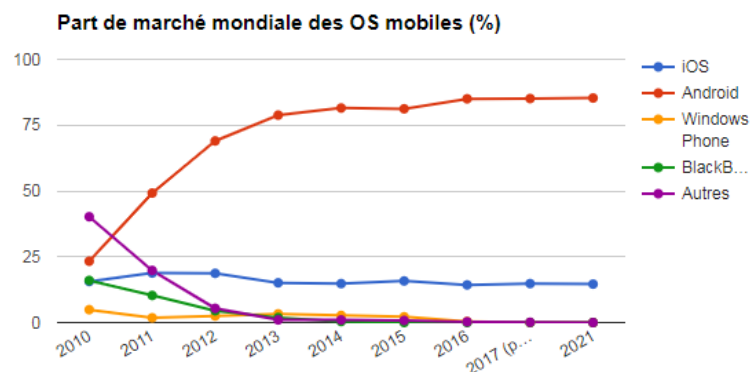


Figure 1.6 – Part de marché mondiale des systèmes d'exploitation mobiles

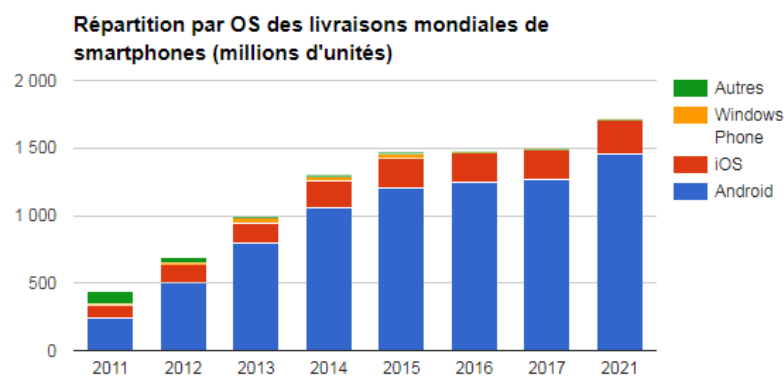


Figure 1.7 – Part de marché mondiale des OS

1.2 les applications mobiles

1.2.1 Définition

Une application mobile est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation d'un Smartphone ou d'une tablette. Les applications mobiles sont adaptées aux différents environnements techniques des Smartphones et à leurs contraintes et possibilités ergonomiques. Elles permettent généralement un accès plus confortable et plus efficace à des sites ou services accessibles par ailleurs en versions mobile ou web.[2]

1.2.2 Cycle de vie d'une activité d'une application mobile

Pour développer une application Android, il faut comprendre le cycle de vie d'une activité qui est illustré par la figure si dessous :[1]

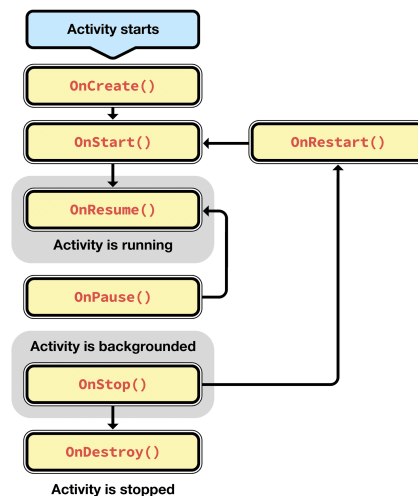


Figure 1.8 – cycle de vie d'une activité[3]

Les différentes méthodes qu'on peut trouver dans le cycle de vie d'une activité sont les suivantes :[1]

1. - **onCreate () /onDestroy ()** : permet de gérer les opérations à faire avant l'affichage de l'activité, et lorsqu'on détruit complètement l'activité de la mémoire. On met en général peu de code dans `onCreate ()` afin d'afficher l'activité le plus rapidement possible.

2. - **onStart ()** / **onStop ()** : ces méthodes sont appelées quand l'activité devient visible/invisible pour l'utilisateur.
3. - **onPause ()** / **onResume ()** : une activité peut rester visible mais être mise en pause par le fait qu'une autre activité soit en train de démarrer, par exemple B. **onPause()** ne doit pas être trop long, car B ne sera pas créé tant que **onPause()** n'a pas fini son exécution.
4. - **onRestart ()** : cette méthode supplémentaire est appelée quand on relance une activité qui est passée par **onStop ()**, puis **onStart ()** est aussi appelée. Cela permet de différencier le premier lancement d'un relancement.

1.3 Découverte de la plateforme Android

1.3.1 Présentation d'Android

Android est une plate-forme logicielle open source destinée aux appareils mobiles qui intègre différents composants : un système d'exploitation, Un middleware, une interface graphique et Un éventail d'applications. Android n'est donc pas lié à un appareil donné, un constructeur de téléphone mobile, ou un opérateur. Google a fédéré autour d'Android une trentaine de sociétés (dont Samsung, Motorola, Sony Ericsson ou LG) à l'intérieur de l'Open Handset Alliance pour diffuser en masse son système. Android laisse la possibilité aux développeurs de créer leurs propres applications à l'aide du langage de programmation Java, et Google fournit aux développeurs de nombreux outils pour écrire et mettre au point leurs programmes : émulateur pour les tests, SDK, et plugins pour le développement, ainsi qu'une documentation fournie. Android repose sur un noyau Linux qui gère les services système comme la sécurité, la gestion mémoire, la gestion des processus, la pile réseau, etc.[4]

1.3.2 Les versions d'Android

Google a adopté la tendance de nommer versions Android avec le nom d'un dessert ou un bonbon dans l'ordre alphabétique. Parmi les différentes versions d'Android, on trouve : Gingerbread 2.3, Honeycomb 3.0, IceCream Sandwich 4.0, Jelly Bean 4.1 et la dernière Lollipop 5.0, etc.[21]

Version	Nom de code	Distribution
2.3.3-2.3.7	Gingerbread	0.3%
4.0.3-4.0.4	Ice Cream Sandwich	0.3%
4.1.x, 4.2.x et 4.3	Jelly Bean	3.2%
4.4	KitKat	6.9%
5.0 et 5.1	Lollipop	14.5%
6.0	Marshmallow	16.9%
7.0 et 7.1	Nougat	19.2%
8.0 et 8.1	Oreo	28.3%
9	Pie	10.4%

Figure 1.9 – Les versions d’Android

1.3.3 Répartition des versions Android

Tous les mois, Google publie des statistiques tirées du Google Play dénombrant les versions d’Android installées sur les appareils qui se sont connectés.

Les dernières statistiques de la répartition des versions Android relevées par Google sont représentés par le graphique suivant :

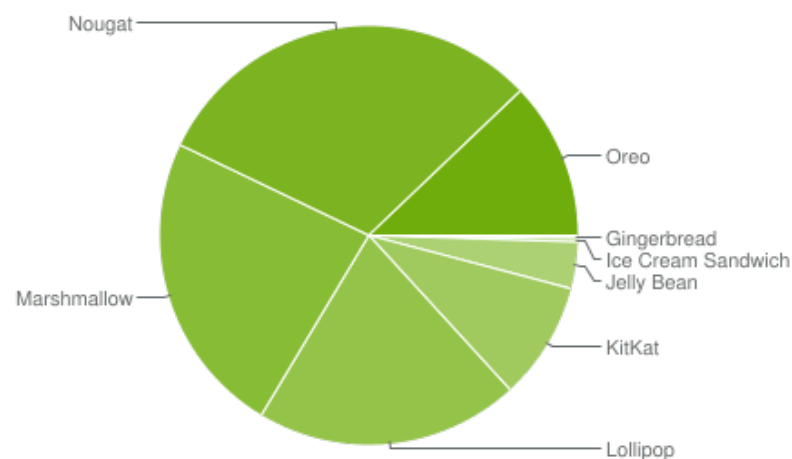


Figure 1.10 – La répartition des versions d’Android[21]

1.3.4 l’architecture d’android

Le schéma suivant illustre les principaux composants du système d’exploitation Android. Chaque section est décrite plus en détail ci-dessous :[5]

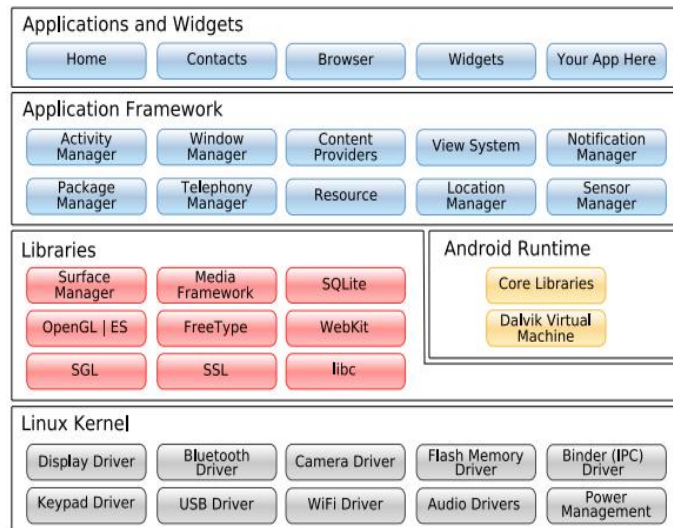


Figure 1.11 – Architecture Android[6]

1. Partie applications :

Toute l'application Android se trouve à la couche supérieure.

2. Partie applications framework :

La couche Application Framework fournit de nombreux services de haut niveau à des applications sous la forme de classes Java.

3. Partie libraries :

Android fournit un ensemble de librairies C/C++. Ces fonctionnalités sont rendues disponibles aux développeurs à travers le Framework d'application d'Android.

4. Partie Android runtime :

Le runtime Android fournit également un ensemble de librairies fournissant des fonctionnalités des librairies standards java.

5. Partie Linux Kernel :

Android s'appuie sur le noyau Linux 2.6 pour les services systèmes de base tels que la sécurité, la gestion de la mémoire et des processus, le réseau et la gestion des drivers.

1.3.5 Les avantages d'Android[22]

*** Open source :**

Le contrat de licence pour Android respecte les principes de l'opensource,Android utilise des bibliothèques opensource puissantes, comme par exemple SQLite ou hyper file SQL pour les bases de données et OpenGL pour la gestion d'images 2D et 3D (pour faire des jeux!).

*** Facile à développer :**

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale, on peut dire que vous pouvez envoyer un SMS en seulement deux lignes de code par exemple.

*** Facile à vendre :**

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

*** Flexible :**

Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. Les smartphones, les tablettes, la présence ou l'absence de clavier ou de trackball, différents processeurs, etc. On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android! Non seulement c'est une immense chance d'avoir autant d'opportunités, mais en plus Android est construit de manière à faciliter le développement et la distribution en fonction des composants en présence dans le terminal.

*** Complémentaire :**

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

combine l'appareil photo avec le GPS, on peut poster les coordonnées GPS des photos prises.

1.3.6 Les outils de développement Android

Les différents outils de développement Android sont :[1]

- **Le SDK Android :**

Le Kit de développement logiciel Android (Android SDK) contient les outils nécessaires pour créer, compiler et déployer les applications Android. La plupart de ces outils sont en ligne de commande.

- **Le débogueur « ADB » :**

Le SDK Android contient un débogueur appelé « Android debug bridge » ou aussi « ADB », qui permet de connecter un appareil Android virtuel ou réel, dans le but de gérer le périphérique ou de déboguer votre application.

- **L'IDE « Android Studio » :**

Google propose également cet IDE appelé pour la création d'applications Android quel que soit le terminal sous-jacent (smartphone, tablette, montre, TV, etc.) et qui est basé sur l'IDE IntelliJ.

Conclusion

Dans ce chapitre nous avons présenté brièvement l'informatique mobile, les applications mobiles et le cycle de vie d'une application android et le monde Android tout en décrivant le système Android et son architecture. Cela nous a aidés à bien comprendre le fonctionnement de ce système.

Chapitre 2

Etude préliminaire

Introduction

Dans ce chapitre, nous allons aborder la maladie du diabète, ses types et ses symptômes avec une présentation de lieu de stage. Nous essayerons aussi de définir la télé-médecine avec ses différents actes et applications, et son apport dans la médecine d'aujourd'hui. Ensuite, on déterminera le lien entre la télémédecine, le diabète en termes de « Télédiabète » et certaines méthodes de mesures de glycémie. Par la suite, nous allons présenter le processus 2TUP pour la modélisation unifiée UML. Finalement, nous allons rassembler les informations initiales sur le système. Dans cette phase, on va définir également le contour du système, les différents acteurs, ainsi que les messages d'interaction avec le système et on terminera par une conclusion.

2.1 La télémédecine

2.1.1 Définition

La télémédecine a été définie par l'OMS en 1997 comme « la partie de la médecine qui utilise la transmission par télécommunications d'informations médicales (images, comptes-rendus, enregistrements, etc.), en vue d'obtenir, à distance, un diagnostic, un avis spécialisé, une surveillance continue d'un malade, une décision thérapeutique ». Elle peut également être l'examen clinique, le suivi et la gestion des patients, ainsi que la formation du personnel médical, au moyen de systèmes permettant l'accès direct à l'avis d'un spécialiste, quel que soit le lieu où se trouve le patient. Elle représente

un enjeu considérable pour l'amélioration des conditions de soins et de vie de beaucoup de personnes. De nombreux auteurs définissent la télémédecine comme l'union des télécommunications et de la médecine. Elle représente l'utilisation des Nouvelles Technologies de l'Information et de la Communication (NTIC) dans le secteur médical. Elle médiatise l'acte médical en interposant un outil de communication entre les médecins ou entre un médecin et son patient.[7]

2.1.2 Les actes de la télémédecine

L'offre de soins médicaux est confrontée à de rudes défis. La télémédecine s'appuie sur de nombreux actes réalisés à distance. On mentionne les principaux qui couvrent le champ de cette vaste discipline à savoir :[7]

2.1.2.1 La téléconsultation

La téléconsultation, est l'un des piliers des actes de la télémédecine. Elle permet, à un patient, d'obtenir un diagnostic médical à distance. Le patient peut également être assisté d'un professionnel de santé.[7]



Figure 2.1 – La téléconsultation [8]

2.1.2.2 La télé-expertise

La télé-expertise est une pratique qui concerne uniquement les professionnels médicaux. Le professionnel médical chargé de surveiller l'évolution de la thérapie d'un patient peut ainsi demander un ou plusieurs conseils à d'autres confrères dans le but de prendre la décision la plus juste possible. Les télé-expertises sont caractérisées par différentes thématiques. Ce procédé simple et novateur permet de : rompre l'isolement

des professionnels de santé, éviter les consultations inutiles ou de préparer le passage vers une autre consultation spécialisée, avoir l'avis d'un professionnel compétent de proximité.[7]



Figure 2.2 – La télésurveillance [9]

2.1.2.3 La télésurveillance

La télésurveillance est un acte médical qui découle de la transmission et de l'interprétation, par un médecin, d'un indicateur clinique, radiologique ou biologique, recueilli par le patient lui-même ou par un professionnel de santé. L'interprétation peut, dans certains cas, conduire à la décision d'une intervention auprès du patient.[7]



Figure 2.3 – La télésurveillance [9]

2.1.2.4 La téléassistance

La téléassistance médicale permet à un professionnel médical d'assister à distance un autre professionnel de santé au cours de la réalisation d'un acte médical ou chirurgical. Le médecin peut également assister un autre professionnel de santé qui réalise un acte de soins ou d'imagerie, voire, dans le cadre de l'urgence, assister, à distance, un secouriste ou toute personne portant assistance à une personne en danger, en attendant l'arrivée d'un médecin.[7]



Figure 2.4 – La téléassistance [10]

2.1.3 Intérêt de la télé médecine

On peut à présent estimer que la télé médecine n'est pas une nouvelle discipline médicale, mais un nouvel aspect d'exercice de la médecine, s'adaptant à chacune des spécialités, dont le but est, lors d'une prise en charge médicale, de minimiser au maximum les problèmes de distance entre différents intervenants transférant les données médicales plutôt que faire déplacer les patients. On peut résumer quelques avantages de la télé médecine en citant ce qui suit :

- Développer les soins à domicile.
- Limiter les déplacements aux personnes âgés ou handicapés.
- Faciliter l'accès aux soins dans les zones d'accès difficile.
- Raccourcir les délais d'attente.
- Apporter un soutien psychologique aux malades, de sorte à ce qu'ils ne se sentent plus seuls face à leur maladie.
- Faciliter la concertation entre médecins.[7]

2.2 Le diabète

2.2.1 Définition

Le diabète est une maladie chronique jusque-ici incurable, causée par une carence ou un défaut d'utilisation de l'insuline, entraînant un excès de sucre dans le sang. Produite par le pancréas, plus précisément par les cellules Beta des îlots Langerhans, l'insuline est une hormone qui permet au glucose (sucre), contenu dans les aliments, d'être utilisé

par les cellules du corps humain. Les cellules disposent de toute cette énergie dont elles ont besoin pour fonctionner. Si l'insuline est insuffisante ou si elle ne remplit pas son rôle adéquatement, comme c'est le cas dans le diabète, le glucose ne peut pas servir de carburant aux cellules. Il s'accumule alors dans le sang et est ensuite déversé dans l'urine. À la longue, l'hyperglycémie provoquée par la présence excessive de glucose dans le sang entraîne certaines complications, notamment au niveau des yeux, des reins, des nerfs, du cœur et des vaisseaux sanguins. À ce jour, la cause réelle du diabète demeure inconnue. Nous savons toutefois, que certains facteurs peuvent influencer l'apparition du diabète : hérédité, obésité, grossesse, certains virus ou certains médicaments.[7]

2.2.2 Les symptômes

Les symptômes du diabète ne se présentent pas tous de la même manière ni avec la même intensité. Les principaux symptômes sont :[7]

- Fatigue.
- Somnolence.
- Augmentation du volume des urines.
- Soif intense et faim exagérée.
- Amaigrissement.
- Vision embrouillée.
- Cicatrisation lente.
- Infection des organes génitaux.
- Picotements aux doigts ou aux pieds.

2.2.3 Les types de diabète

Il existe deux principaux types de diabète : le type 1 et le type 2. Parfois, le diabète se développe aussi pendant la grossesse :[7]

- **Le diabète de type 1 :**

Le diabète de type 1, appelé autrefois diabète insulino-dépendant (DID), touche environ 6% des diabétiques, est habituellement découvert chez les personnes

jeunes : enfants, adolescents ou jeunes adultes. Ce diabète résulte de la disparition des cellules bêta du pancréas entraînant une carence totale en insuline. L'organisme ne reconnaît plus ces cellules bêta et les détruit (les cellules bêta sont détruites par des anticorps et des cellules de l'immunité, les lymphocytes, fabriquées par l'organisme) : on dit que le diabète de type 1 est une maladie auto-immune. Le glucose ne pouvant entrer dans les cellules retourne dans le sang. Le taux de glucose dans le sang s'élève alors.[7]

- **Le diabète de type 2 :**

Le diabète de type 2 se manifeste beaucoup plus tard dans la vie, généralement après l'âge de 40 ans. La très grande majorité des personnes atteintes de diabète présentent ce type de diabète (90 % des cas). Depuis quelques années, on remarque que ce diabète apparaît plus tôt. Une prédisposition génétique, l'obésité et le manque d'activité physique contribuent à l'apparition d'un diabète de type 2. Cependant, certaines études tendent à démontrer qu'une alimentation en trop grande quantité et riche en gras pourrait aussi être un facteur de risque. En apportant des corrections importantes à nos habitudes de vie, il est possible de retarder l'apparition de la maladie et d'en diminuer l'impact.[7]

- **Le diabète de grossesse (gestationnel) :**

Il s'agit d'une maladie que les femmes peuvent contracter aux premiers stades de la grossesse. Contrairement aux diabètes de type 1 et type 2, le diabète de grossesse disparaît après la naissance du bébé.[11]

2.2.4 Les complications du diabète

Les complications du diabète sont nombreuses et peuvent être sévères : infarctus, troubles de la vision, cécité, accident vasculaire, amputations, maladies rénales, etc. Ces complications aggravent le diabète et diminuent l'espérance de vie des personnes atteintes de cette maladie. La majorité des complications, liées au diabète, peuvent être évitées, diminuées ou retardées si le diabète est dépisté et traité précocement et correctement. Qu'il s'agisse du type 1, du type 2 ou du diabète de grossesse, une consultation chez le médecin s'impose. Généralement, la simple mesure de la glycémie à jeun, par prise de sang, suffit pour dépister un diabète.[11]

2.3 La télémedecine et le diabète

Le fonctionnement de la télémedecine est simple : les échanges entre le patient et son médecin ont lieu à travers divers supports technologiques, comme Internet, à l'aide d'e-mails ou de messages sur des forums dédiés, ou par le biais de vidéo-consultations. C'est l'occasion pour le professionnel de santé de s'entretenir avec son malade, pour lui délivrer différents conseils pratiques, d'ordre nutritionnel ou relatif à la pratique d'une activité physique, par exemple. Mais tout cela ne peut se faire, dans le cas du diabète, que si le patient transmet correctement le recueil de ses valeurs glycémiques. Il doit donc être impliqué, pour contrôler régulièrement son taux de sucre sanguin et fournir ces chiffres à son médecin. Un suivi quotidien qui n'est pas réalisable sans la télémedecine, c'est là sans doute l'un de ses points forts.[7]

2.4 Mesure de la Glycémie

Aspect médical : la glycémie est la concentration plasmatique du glucose. Au cours de la journée, sa valeur varie en fonction des apports et des besoins énergétiques de l'individu. La glycémie est ajustée par l'action d'hormones sécrétées par des cellules du pancréas. Ce système de régulation permet de maintenir le même taux alors même que les cellules des organes ont des besoins différents en fonction de leur activité. Pour les périodes de mesure de la glycémie, nous avons : [7]

- **Glycémie à jeun :** pour la glycémie à jeun, la prise de sang a lieu sans apport calorique pendant huit heures au moins.[7]
- **Glycémie postprandiale :** la glycémie est vérifiée 1 h 1/2 à 2 h après le début du repas.[7]
- **Analyse de la glycémie :** la mesure de la glycémie permet de savoir s'il y a une bonne régulation du taux de sucre dans le sang. Cet examen est prescrit lorsque l'hyperglycémie symptomatique est soupçonné entre autre, du diabète. Mais il est aussi prescrit pour détecter une hypoglycémie, c'est-à-dire un taux de sucre insuffisant dans le sang.[7]

- **Valeurs normales de la glycémie :**

Valeur normal ou cible chez les non diabétiques	
A jeun	Entre 70 et 99 mg/dl
Après les repas	Entre 70 et 140 mg/dl
Valeur normal ou cible chez les diabétiques	
A jeun	Entre 70 et 130 mg/dl
1h à 2h après le repas	Au-dessous de 180 mg/dl
Valeur normal ou cible chez la femme enceinte	
A jeun	Entre 60 et 95 mg/dl
Avant le repas	Entre 60 et 105 mg/dl
2h après le repas	Au-dessous de 120 mg/dl

Table 2.1 – Valeurs normales de la glycémie

2.5 Présentation de l'entreprise du lieu de stage

Dans cette partie, on commence par une brève présentation de l'établissement afin de tirer et d'analyser les objectifs du projet.

2.5.1 Identification de l'établissement

L'EPH Frères TOBAL de MILA est un établissement public à caractère administratif doté de la personnalité morale et de l'anatomie financière. Il est constitué d'une structure de diagnostic, de soins, d'hospitalisation et de réadaptation médicale couvrant la population de 08 communes soit 186317 habitants. L'établissement a été créé par décret exécutif N07/140 du 19 mai 2007.

2.5.2 Localisation de l'établissement

L'EPH Frères TOBAL se situe au niveau de Mila, chef lieu de wilaya.



Figure 2.5 – Localisation de l'établissement[7]

2.5.3 Fiche technique

Le tableau suivant représente quelques informations relatives à l'établissement dans laquelle nous avons effectué notre stage de projet de fin d'étude.

Dénomination	EPH des Frères TOBAL de MILA
Classification	Catégorie C.
Localisation	MILA, chef lieu de wilaya.
Adresse	Rue ALLIOUCHE NOUAR.MILA(4300).
Surface	1600 mètre carré.
Communes couvertes	08.
Population desservie	186317.
Médecine Interne	40 lits.
Maladies infectieuses	24 lits.
Hôpital de jour	08 lits.
Hémodialyse	16 lits.
TOTAL	88 lits.

Table 2.2 – Fiche technique.

2.5.4 Organigramme d'accueil

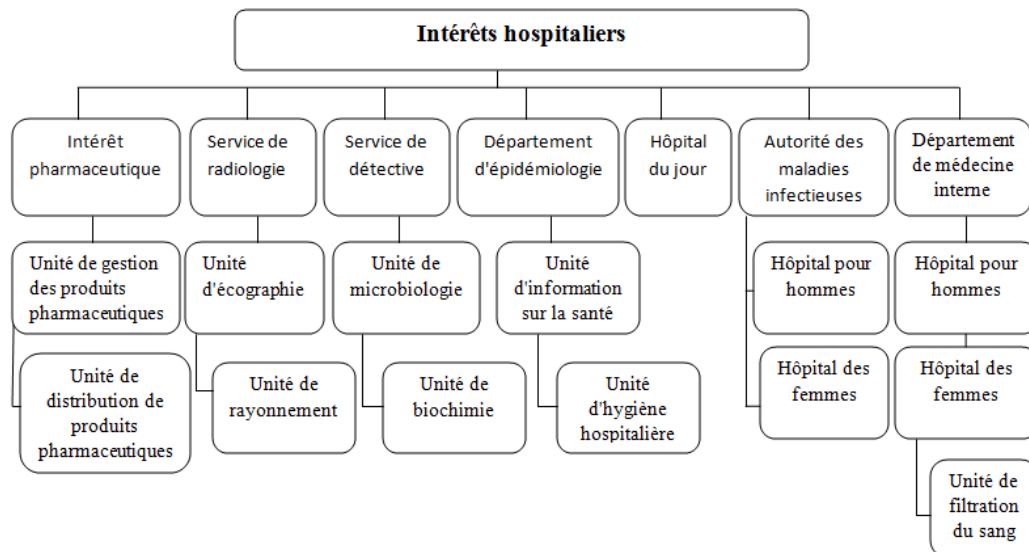


Figure 2.6 – organigramme de l'établissement

2.6 Contribution

Dans le cadre de cette étude, l'objectif est de remplacer le carnet de suivi papier du diabète par le développement d'une application mobile qui améliore le quotidien de toutes les personnes atteintes de diabète ainsi que permettre :

- Au patient de :
 - Introduire manuellement les valeurs (glucomètre doté de Bluetooth non disponible) et les enregistrer dans la base de données .
 - Lui rappeler quotidiennement de mesurer sa glycémie grâce au système de notification.
 - Consulter son historique .
 - Connaître l'état de sa glycémie.
 - Faciliter la prise des rendez-vous
- Au médecin/assistant de :
 - Consulter les graphes et établir un diagnostic facilement.
 - Améliorer la gestion des patients.

- Améliorer gestion des rendez-vous.
- Augmenter le nombre de consultations.

2.7 Elaboration du cahier de charge

2.7.1 Grands choix techniques

Pour modéliser ce projet d'une manière claire et précise la structure et le comportement de notre application mobile, nous adoptons l'approche itérative et incrémentale, fondée sur le processus en Y.

Nous avons choisi aussi un certain nombre de techniques-clés. Ces techniques-clés sont principalement :

- Le langage de modélisation UML 2.0.
- Le processus de développement en Y (2TUP).
- L'environnement de programmation android studio et laravel.
- L'adoption d'une architecture(client/serveur 2-tiers).
- Le serveur Apache.
- Le SGBDR MYSQL.
- Le langage de programmation JAVA+XML pour l'application mobile et PHP pour l'application web.

2.7.2 Le processus 2TUP

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens). Le processus s'articule ensuite autour de 3 phases essentielles :[13]

- Une branche technique.

- Une branche fonctionnelle.
- Une phase de réalisation.

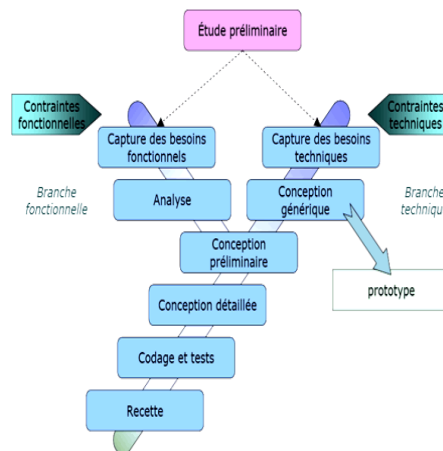


Figure 2.7 – Processus de développement en Y[13]

2.8 Description de contexte du système

Dans cette partie nous développons un premier modèle UML de niveau contexte, pour pouvoir établir les frontières fonctionnelles du système. La description du contexte du système contient 3 actions :[14]

- Identification des acteurs.
- Identification des messages entre le système et ses acteurs.
- La réalisation du diagramme du contexte.

2.8.1 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (Utilisateur, dispositifs matériels ou autre système) qui interagit directement avec le système étudié. En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. Les acteurs du système sont :[14]

- **L'Administrateur** :son rôle de l'administrateur est la gestion des utilisateurs qui peuvent accéder au système.
- **Le médecin** :désigne la personne habilitée à pratiquer la médecine et chargée d'examiner les patients et de diagnostiquer leurs états.

- **Le patient** : désigne la personne concerné par le diabète et qui se voit administrer un traitement.
- **L'assistant(e) médical(e)** : désigne la personne qui, dans un cabinet médical, effectue des tâches qui s'articulent autour de trois grands axes : l'accueil des patients et la gestion des rendez-vous, l'assistance des travaux de laboratoire.

2.8.2 Identification des messages

Un message représente la spécification d'une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur. Cette notion de message est également tout à fait applicable pour décrire les interactions de plus haut niveau entre les acteurs et le système. Pour chaque acteur, demandez-vous quels sont les messages qui déclenchent un comportement du système attendu par l'acteur dans le cadre de son activité.[14]

- **Le patient peut :**

- S'authentifier.
- Gérer la valeur glycémie manuellement.
- Consulter son dossier médicale.
- Consulter son profile.
- Modifier son profile.
- Prendre un rendez-vous.
- Annuler un rendez-vous.

- **Le médecin peut :**

- S'authentifier.
- Consulter les dossiers médicaux.
- Modifier les dossiers médicaux des malades.
- Consulter son profile.
- Modifier son profile.

- **L'assistant peut :**

- S'authentifier.

- Confirmer les rendez-vous.
- Consulter le planning des rendez-vous.
- Consulter son profile.
- Modifier son profile.
- Saisir la valeur glycémie 3 mois.
- **L'administrateur peut :**
 - S'authentifier.
 - Créer les comptes.
 - Consulter les profiles des utilisateurs.
 - Bloquer les profiles des utilisateurs.
 - Modifier les profiles des utilisateurs.

2.8.3 Modélisation de contexte

2.8.3.1 Diagramme de contexte dynamique

Les messages entre le système et les acteurs identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme, que l'on peut qualifier de diagramme de contexte dynamique en utilisant un diagramme de communication de la manière suivant :

- Le système étudié est représenté par un objet central.
- Cet objet central est entouré par d'autres objets symbolisant les différents acteurs.
- Des liens relient le système à chacun des acteurs.
- Sur chaque lien sont montrés les messages en entrées et en sorties du système.[11]

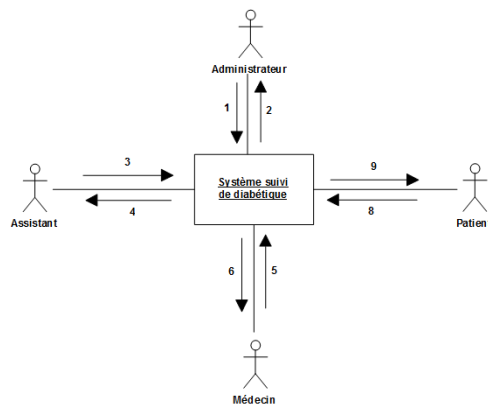


Figure 2.8 – Le diagramme de contexte dynamique du système

2.8.3.2 Description détaillée des messages

1 : Administrateur au Système

- Les informations concernant le compte d'authentification (login et mot de passe).
- Le blocage des compte d'utilisateurs.
- La modification des informations sur les comptes (mot de passe, login).

2 : Système au Administrateur

- Confirmation des informations d'authentification.
- Confirmation des mises à jour concernant les comptes.

3 : Assistant au Système

- Les informations concernant le compte d'authentification (login et mot de passe).
- L'analyse de la glycémie de 3 mois .
- La confirmation des rendez-vous.

4 : Système au Assistant

- Les informations du compte assistant.
- Liste des patients et des médecins.
- Planning des rendez-vous.
- La liste des rendez-vous.

5 : Médecin au Système
• Les informations concernant le compte d'authentification (login et mot de passe). • Ajouter et valider des remarques dans les dossiers médicaux des patients.
6 : Système au Médecin
• Les informations du compte médecin. • Liste des patients. • Le dossier médicale de patient.

5 : Patient au Système
• Les informations concernant le compte d'authentification (login et mot de passe). • Les valeurs de glycémie manuellement. • Le jour et l'heure de rendez-vous.
6 : Système au Patient
• Les informations du compte patient. • L'historique de malade. • Confirmation de rendez-vous.

Table 2.3 – Légende des messages de diagramme de contexte

Conclusion

Dans ce chapitre, nous avons essayé de faire une petite vision sur la nouvelle technologie la télémédecine. Puis nous avons abordé la maladie du diabète pour laquelle sera appliquée notre solution de télésurveillance en exploitant les Smartphones. Ensuite, nous avons a présenté l'établissement de notre stage. Nous avons également définit un modèle UML de niveau contexte et le diagramme de cas d'utilisation, pour pouvoir établir précisément les frontières fonctionnelles du système. Tout ça prépare le terrain pour entamer la conception de notre projet par l'étape de capture des besoins fonctionnels et de capture des besoins techniques dans le chapitre suivant.

Chapitre 3

Capture des besoins fonctionnels et techniques

Introduction

Après avoir tracé les grandes lignes de phase de spécification de besoins, mettons l'accent maintenant sur une phase fondamentale dans le cycle de vie d'un logiciel, la phase de conception. Cette phase a pour objectif de déduire la spécification des besoins fonctionnels et techniques du système. L'objectif de la capture des besoins consiste à déterminer ce que le système doit faire, c.à.d. le « quoi » a fourni aux développeurs une meilleure compréhension des fonctionnalités du système qu'ils doivent développer, elle comporte deux étapes : la capture des besoins fonctionnels et la capture des besoins techniques.

3.1 Capture des besoins fonctionnels

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. Elle formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire. Elle est complétée au niveau de la branche droite du Y par la capture des besoins techniques et prépare l'étape suivante de la branche gauche : l'analyse.[14]

3.1.1 Identification des cas d'utilisations

3.1.1.1 Liste préliminaire des cas d'utilisations

Considérons l'intention fonctionnelle de l'acteur par rapport au système dans le cadre de l'émission ou de la réception de chaque message. Chaque intention fonctionnelle est modélisée par un cas d'utilisation.[20]

Cas d'utilisation	Acteur principal, acteurs secondaires	Message(s) émis / reçus par les acteurs
S'authentifier	Administrateur, patient médecin, assistant	Emet : s'authentifier(). Reçoit : La décision.
Créer profil médecin	Administrateur	Emet : créer profil médecin (), enregistrer création () . Reçoit : La décision.
Créer profil assistant	Administrateur	Emet : créer profil assistant (),enregistrer création () . Reçoit : La décision.
Créer profil patient	Administrateur	Emet : créer profil patient (),enregistrer création () . Reçoit : La décision.
Consulter profil utilisateur	Administrateur	Emet : consulter profil utilisateur(). Reçoit : profil utilisateur.
Modifier profil utilisateur	Administrateur	Emet : modifier profil utilisateur (),enregistrer modification () . Reçoit : La décision.
Bloquer profil utilisateur	Administrateur	Emet : bloquer profil utilisateur (),enregistrer modification () . Reçoit : La décision.
Consulter les dossiers médicaux	Médecin	Emet : consulter dossiers médicaux(). Reçoit : dossier médicale de patient.
Modifier les dossiers médicaux	Médecin	Emet : modifier dossiers médicaux (),enregistrer modification () . Reçoit : La décision .

Consulter planning rendez-vous	Assistant	Emet : consulter planning rendez-vous(). Reçoit : planning des rendez-vous.
Confirmer rendez-vous	Assistant	Emet : confirmer rendez-vous (),enregistrer la confirmation (). Reçoit : La décision.
Saisir analyses 3 mois	Assistant	Emet : saisir glycémie 3mois (),enregistrer la saisie () . Reçoit : La décision.
Prendre rendez-vous	Patient	Emet : prendre rendez-vous (), enregistrer création () . Reçoit : La décision.
Annuler rendez-vous	Patient	Emet : annuler rendez-vous (),enregistrer l'annulation () . Reçoit :La décision.
Ajouter valeur glycémie	Patient	Emet : ajouter valeur glycémie (),enregistrer l'ajout () . Reçoit : La décision.
Modifier valeur glycémie	Patient	Emet : modifier valeur glycémie (),enregistrer modification () . Reçoit : La décision.
Consulter son dossier médicale	Patient	Emet : consulter son dossier(). Reçoit : dossier médical.
Consulter son profil	Administrateur, patient médecin, assistant	Emet : consulter son profil () . Reçoit : son profil.
Modifier son profil	Administrateur, patient médecin, assistant	Emet : modifier son profil (),enregistrer modification () . Reçoit : La décision.

Table 3.1 – Liste des acteurs et des messages par cas d'utilisation

3.1.1.2 Description préliminaire des différents cas d'utilisation

Voici une description préliminaire des cas d'utilisations énumérés précédemment :

- **S'authentifier**

— Intention :s'authentifier.

- Actions :fait l'authentification avant chaque accées à l'application.
- **Créer profil médecin**
 - Intention :créer profil pour utilisateur type médecin.
 - Actions :fait la création de compte (profil) médecin.
- **Créer profil assistant**
 - Intention :créer profil pour utilisateur type assistant.
 - Actions :fait la création de compte (profil) assistant.
- **Créer profil patient**
 - Intention :créer profil pour utilisateur type patient.
 - Actions :fait la création de compte (profil) patient.
- **Consulter profil utilisateur**
 - Intention :consulter profil utilisateur par administrateur.
 - Actions :fait la consultation du compte (profil) utilisateur.
- **Modifier profil utilisateur**
 - Intention :modifier profil utilisateur par administrateur.
 - Actions :fait la modification du compte (profil) utilisateur.
- **Bloquer profil utilisateur**
 - Intention :bloquer profil utilisateur par administrateur.
 - Actions :fait le blocage du compte (profil) utilisateur.
- **Consulter les dossiers médicaux**
 - Intention :consulter les dossiers médicaux des patient par médecin.
 - Actions :la consultation des dossiers médicale.
- **Modifier les dossiers médicaux**
 - Intention :modifier les dossiers médicaux des patient par médecin.
 - Actions :la modification des dossiers médicale.
- **Consulter planning rendez-vous**
 - Intention :consulter le planning des rendez-vous par l'assistant.
 - Actions :la consultation de planning des rendez-vous.

- **Confirmer rendez-vous**

- Intention :confirmer le rendez-vous de patient par l'assistant.
- Actions :la confirmation des rendez-vous.

- **Saisir analyses 3 mois**

- Intention :saisir la valeur d'analyses 3mois de patient par l'assistant.
- Actions :la saisie de glycémie 3mois.

- **Prendre rendez-vous**

- Intention :prendre un rendez-vous par le patient.
- Actions :le patient prend un rendez-vous.

- **Annuler rendez-vous**

- Intention :annuler un rendez-vous par le patient.
- Actions :le patient annule un rendez-vous.

- **Ajouter valeur glycémie**

- Intention :ajouter valeur glycémie par le patient.
- Actions :le patient ajoute une valeur glycémie.

- **Modifier valeur glycémie**

- Intention :modifier valeur glycémie par le patient.
- Actions :le patient modifie une valeur glycémie.

- **Consulter son dossier médicale**

- Intention : consulter le dossier médicale par le patient.
- Actions :le patient fait la consultation de son dossier médicale.

- **Consulter son profil**

- Intention : consulter le profil.
- Actions :l'utilisateur consulte les informations de son profil.

- **Modifier son profil**

- Intention : modifier le profil.
- Actions :l'utilisateur fait les modification sur certains informations de son profil.

3.1.1.3 Le diagramme de cas d'utilisation pour les besoins fonctionnels

Représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est normalement utilisé lors des étapes de capture des besoins fonctionnels et techniques.

1. Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.[14]
2. Un message représente la spécification d'une communication unidirectionnelle entre les objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur.[14]
3. Un cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/systèmes.[14]

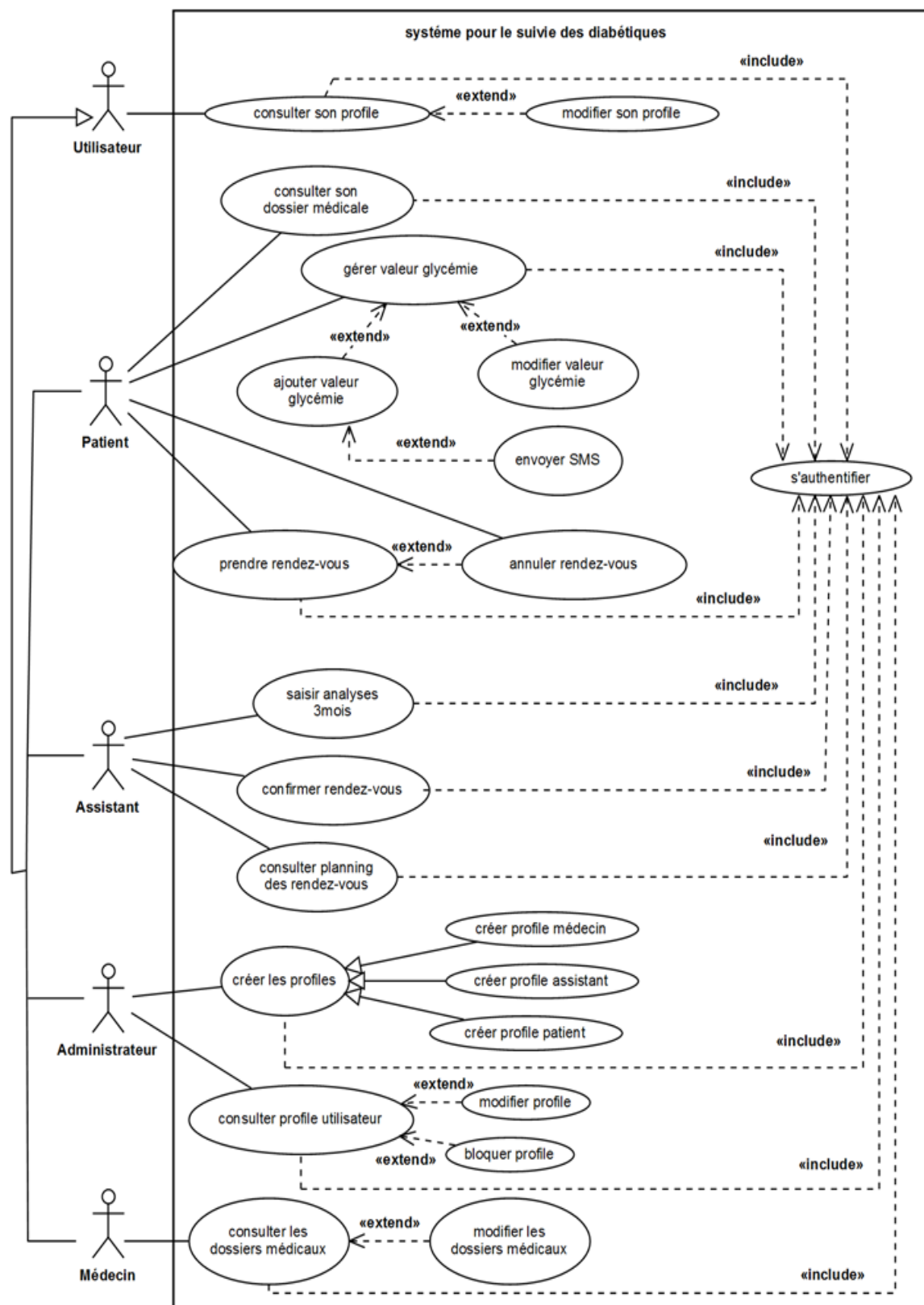


Figure 3.1 – Diagramme de cas d'utilisation globale

3.1.2 Description détaillée des différents cas d'utilisation

Nous allons maintenant détailler chaque cas d'utilisation qui doit faire l'objet d'une définition a priori qui décrit l'intention de l'acteur lorsqu'il utilise le système et les séquences d'actions principales qu'il est susceptible d'effectuer. Les descriptions vont être organisées de la façon suivante :

- **Une description textuelle détaillée** : des pré conditions au déclenchement du cas d'utilisation doivent être spécifiées, un scénario nominal décrivant celui-ci additionné à des scénarios alternatifs et d'exceptions.[11]
- **Une description graphique (les diagrammes)** : le diagramme de séquence permet de représenter les interactions entre différents objets, selon un point de vue temporel en se basant sur la chronologie des envois de messages. Le temps est représenté comme s'écoulant du haut vers le bas le long des « lignes de vie ». Des flèches représentant les messages qui transitent d'une entité à une autre, le message est synchrone. Si l'extrémité de la flèche est creuse, le message est asynchrone.[11]

La représentation :

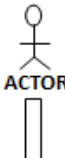
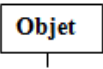
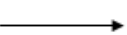
	Acteur	Les acteurs peuvent communiquer avec des objets, ainsi ils peuvent eux aussi être énumérés en colonne. Un acteur est modéliser en utilisant le symbole habituel :Stick man
	Objet	Les objets sont les entités appartenant au système (instance d'une classe) ou se trouvant à ses limites (acteurs).
	Ligne de vie	Elle est représentée par une ligne verticale en dessous des objets , représente la période de temps durant laquelle l'objet existe .
	Message récursif	L'envoi des messages récursifs se représente par un dédoublement de la bande d'activation.
	Message	Les objets communiquent en échangeant des messages représentés sous forme de flèches, ils sont étiquetés par le nom de l'opération ou du signal invoqué.
	Message de retour	Représenté par une flèche discontinue ,c'est la réponse au message envoyé.

Figure 3.2 – Représentation d'un diagramme de séquence[11]

1. Cas d'utilisation « s'authentifier »

Cas d'utilisation	S'authentifier
Les acteurs	Patient, médecin, administrateur, assistant.
Objectif	Vérifier l'autorisation d'accéder aux système.
Pré-condition	L'utilisateur possède un compte
Post-condition	L'utilisateur est identifié par le système et accède à l'application.
Scénario nominal	1-L'utilisateur demande l'accès à l'application. 2-Le système affiche le formulaire d'authentification. 3-L'utilisateur remplit le formulaire (nom d'utilisateur mot de passe). 4-Le système vérifie la validité du mot de passe et ouvre une session utilisateur.
Scénario alternatif	4.1- Le nom de l'utilisateur et/ou le mot de passe sont erronés, Le système affiche une notification d'erreur« Le nom ou mot de passe est incorrect ». -Le scénario nominal reprend au point 2.

Table 3.2 – Cas d'utilisation « s'authentifier »

2. Cas d'utilisation « ajouter valeur glycémie »

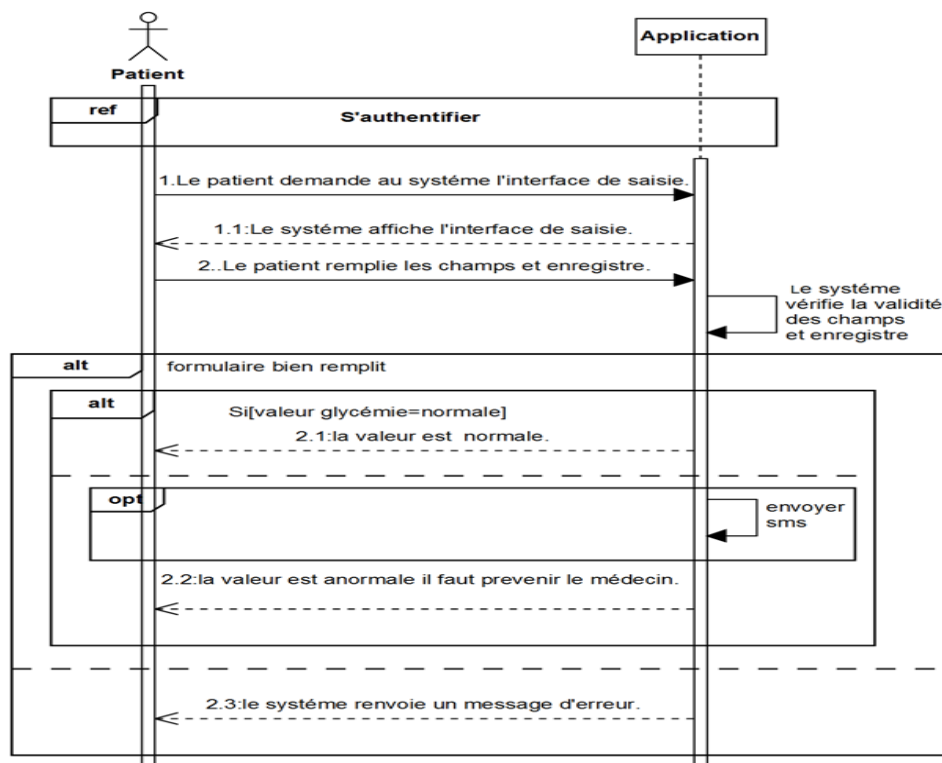


Figure 3.3 – Cas d'utilisation « ajouter valeur glycémie »

3. Cas d'utilisation « créer profil médecin »

Cas d'utilisation	Créer profil médecin
L'acteur principale	Administrateur.
Objectif	Créer un nouveau profil d'un médecin.
Pré-condition	Administrateur s'authentifie.
Post-condition	Profil médecin créé.
Scénario nominal	1-L'administrateur demande de créer profil médecin. 2-Le système affiche le formulaire de création profil médecin. 3-L'administrateur remplit les champs et enregistre. 4-Le système vérifie les informations, enregistre et affiche un message de confirmation «compte médecin créé».
Scénario alternatif	4.1-L'une des informations remplies est erronée ou un champ vide, le système affiche message d'erreur. -Le scénario nominal reprend au point 2.

Table 3.3 – Cas d'utilisation « créer profil médecin »

4. Cas d'utilisation « annuler rendez-vous »

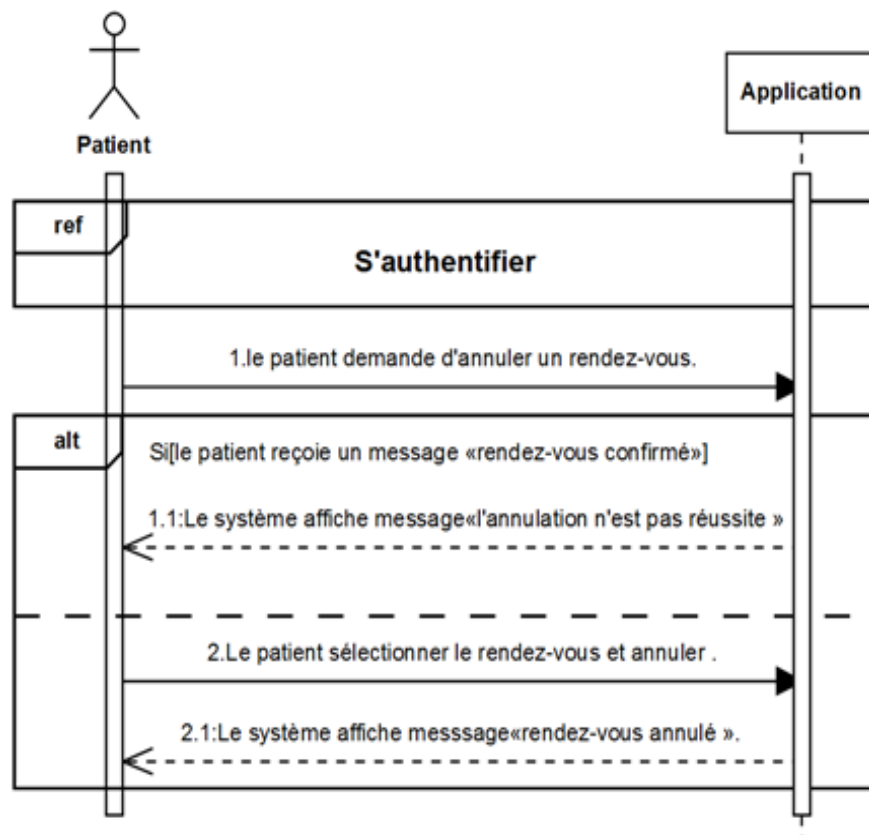


Figure 3.4 – Cas d'utilisation « annuler rendez-vous »

5. Cas d'utilisation « consulter son profil »

Cas d'utilisation	Consulter son profil
L'acteur principal	Utilisateur (patient, médecin, assistant).
Objectif	La consultation du compte utilisateur.
Pré-condition	Le compte existe dans la base de donnée, utilisateur est authentifié.
Post-condition	Profil utilisateur consulté.
Scénario nominal	1-L'utilisateur demande de consulter son profil. 2-Le système affiche les informations.

Table 3.4 – Cas d'utilisation « consulter son profil »

6. Cas d'utilisation « prendre rendez-vous »

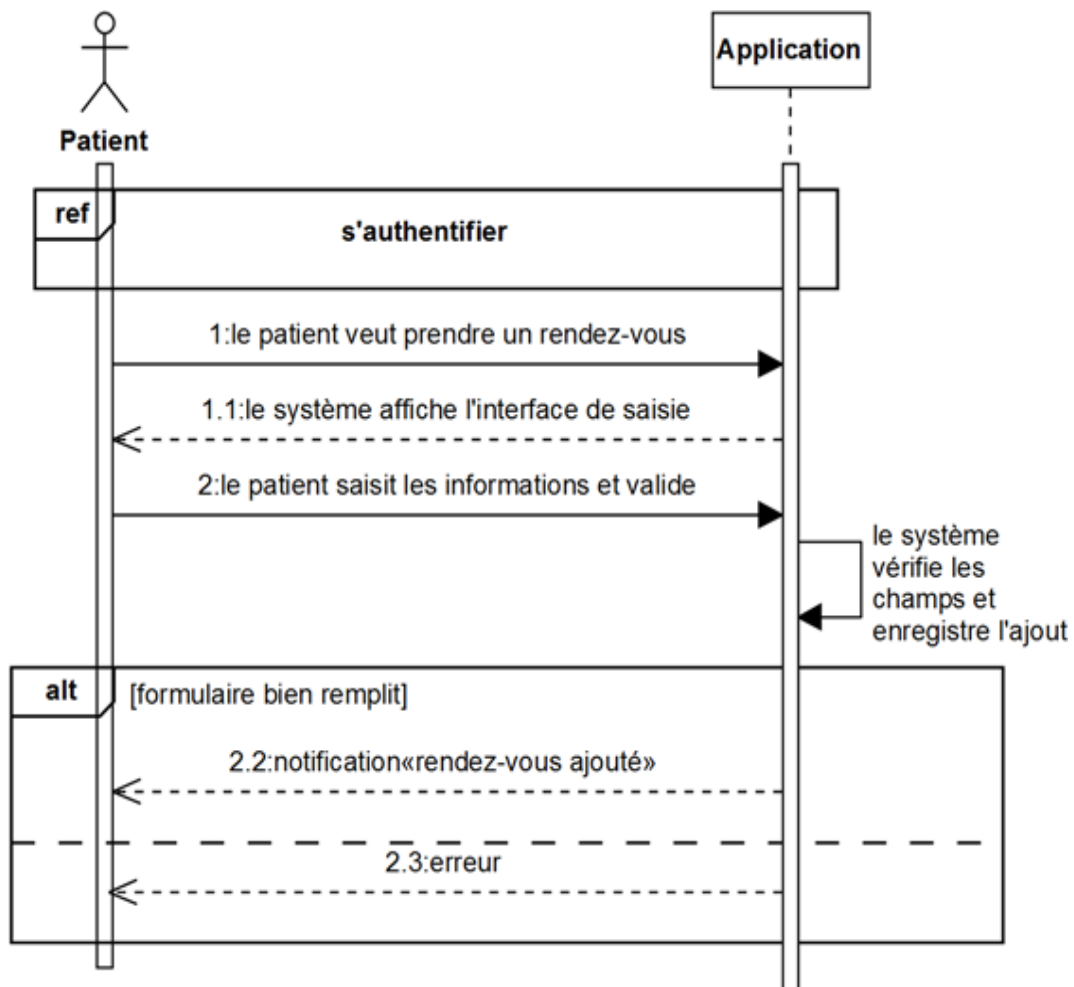


Figure 3.5 – Cas d'utilisation « prendre rendez-vous »

7. Cas d'utilisation « consulter les profils utilisateurs »

Cas d'utilisation	Consulter profil utilisateur
L'acteur principal	Administrateur.
Objectif	La consultation du compte d'un utilisateur.
Pré-condition	-Les utilisateurs ont des comptes -Administrateur est authentifié.
Post-condition	Profil utilisateur consulté.
Scénario nominal	1-Le système affiche la page d'accueil. 2-L'administrateur demande la liste des utilisateurs. 3-Le système affiche la liste des utilisateurs. 4-L'administrateur sélectionne l'utilisateur à consulter. 5-Le système affiche profil d'utilisateur.

Table 3.5 – Cas d'utilisation « consulter les profils utilisateurs »

8. Cas d'utilisation « modifier les dossiers médicaux »

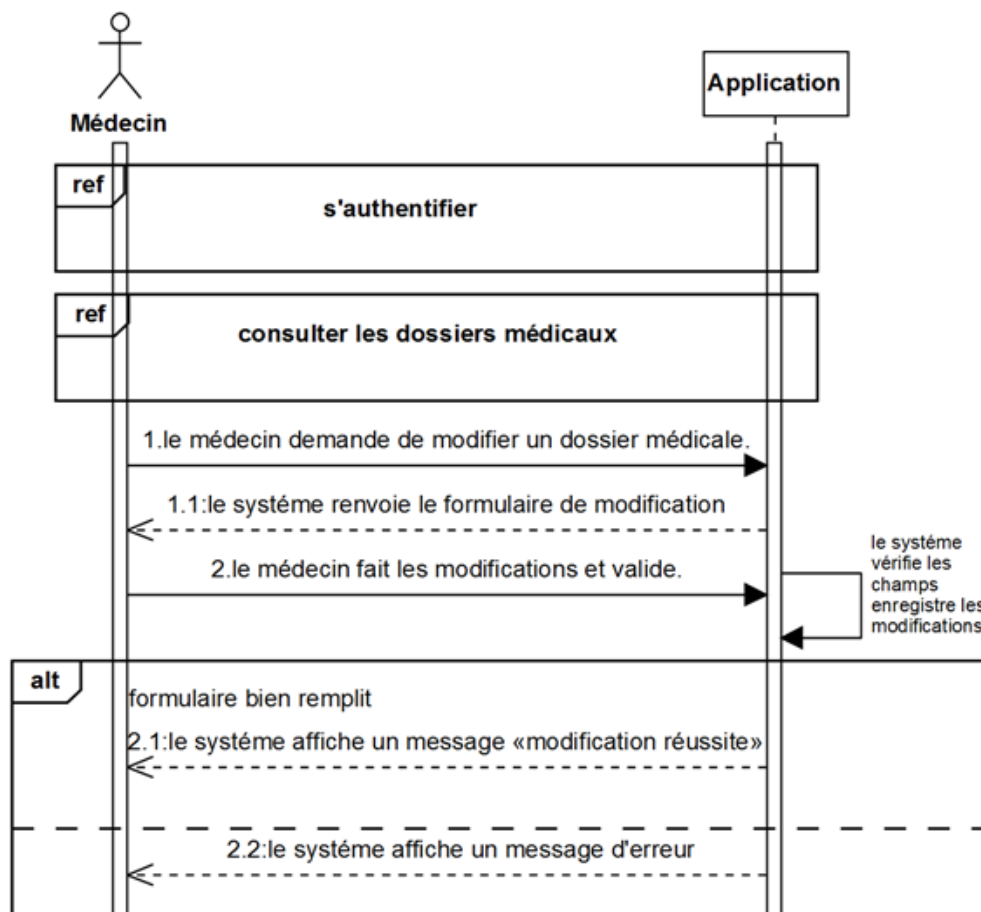


Figure 3.6 – Cas d'utilisation « modifier les dossiers médicaux »

9. Cas d'utilisation « confirmer rendez-vous »

Cas d'utilisation	Confirmer rendez-vous
L'acteur principal	Assistant.
Objectif	Confirmation rendez-vous d'un patient.
Pré-condition	Le rendez-vous existe dans la base, assistant s'authentifie.
Post-condition	Rendez-vous confirmé.
Scénario nominal	1-L'assistant demande au système la liste des rendez-vous. 2-Le système affiche la liste des rendez-vous. 3-L'assistant sélectionne un rendez-vous . 4-L'assistant confirme le rendez-vous. 5-Le système affiche notification de confirmation au patient .

Table 3.6 – Cas d'utilisation « confirmer rendez-vous »

10. Cas d'utilisation « créer profil assistant »

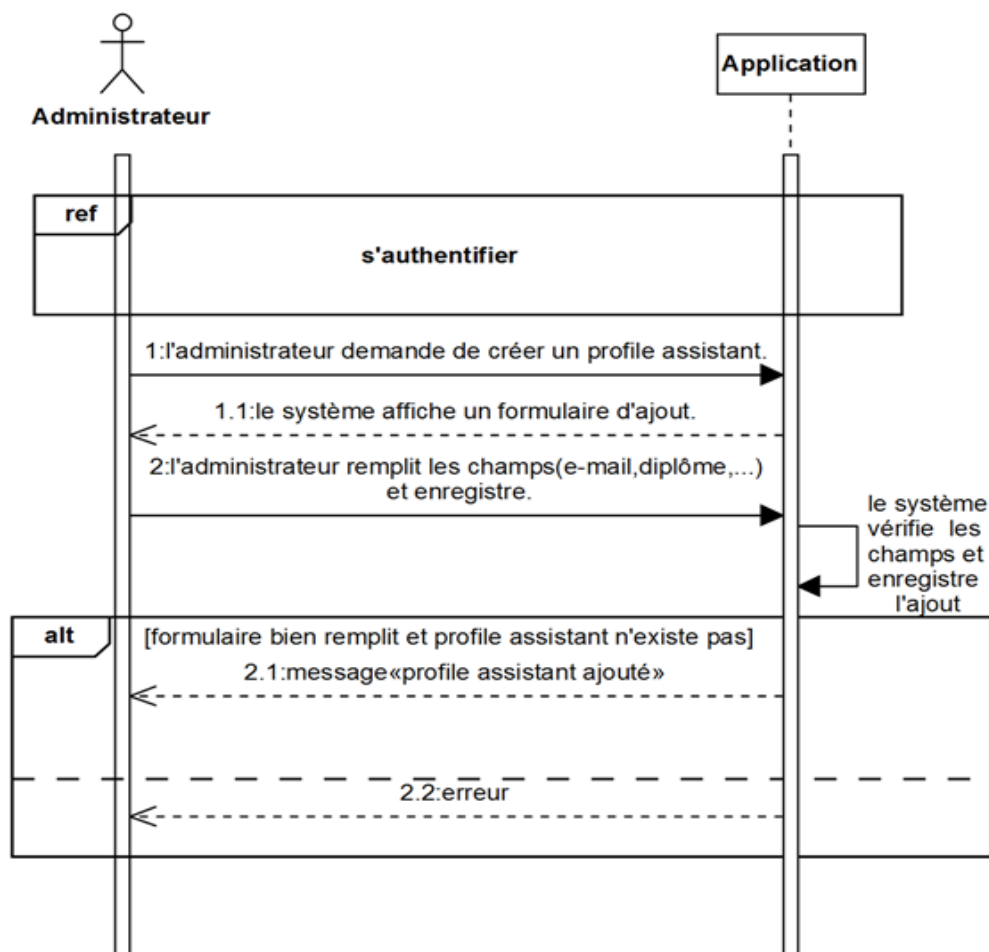


Figure 3.7 – Cas d'utilisation « créer profil assistant »

11. Cas d'utilisation « saisir analyses 3 mois »

Cas d'utilisation	Saisir analyses 3 mois
L'acteur principal	Assistant.
Objectif	Saisir les valeurs d'analyses 3 mois d'un patient.
Pré-condition	-Le patient existe dans la base de données. -Assistant s'authentifie.
Post-condition	Les valeurs d'analyses 3 mois ajoutée.
Scénario nominal	1-L'assistant demande au système la liste des patients. 2-Le système affiche la liste des patients. 3-L'assistant sélectionne un patient. 4-Le système affiche formulaire de saisie. 4-L'assistant saisit les valeurs d'analyses 3 mois. 5-Le système vérifie les informations, enregistre et affiche un message de confirmation.
Scénario alternatif	4.1-L'une des informations remplies est erronée ou un champ vide,le système affiche message d'erreur. -Le scénario nominal reprend au point 2.

Table 3.7 – Cas d'utilisation « saisir analyses 3 mois »

12. Cas d'utilisation « créer profil patient »

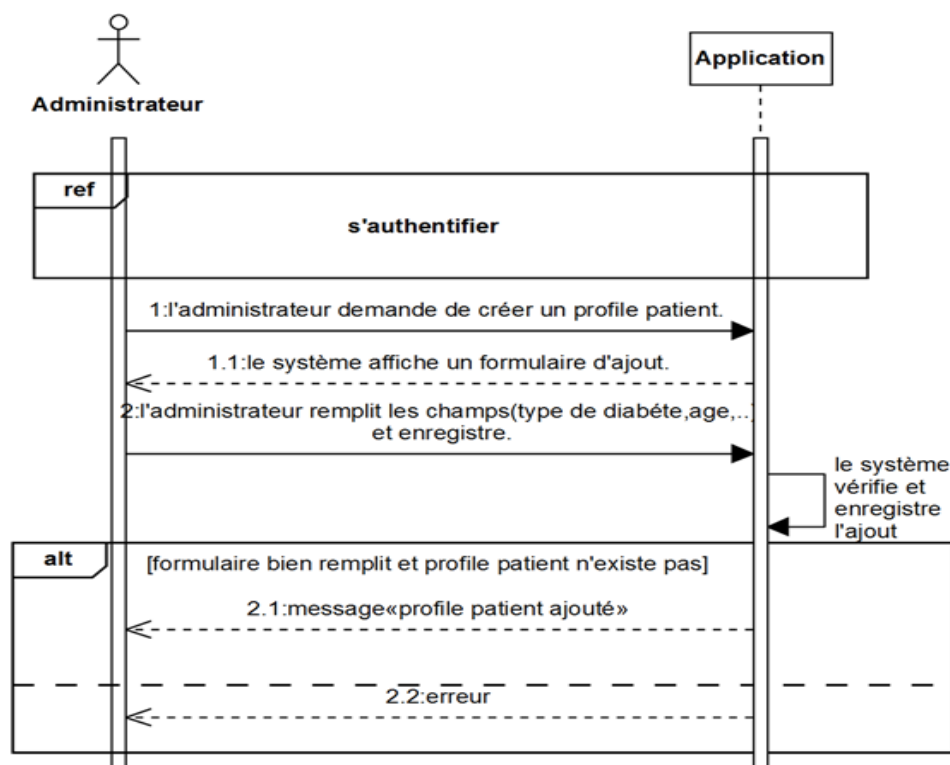


Figure 3.8 – Cas d'utilisation « créer profil patient »

13. Cas d'utilisation « consulter planning des rendez-vous »

Cas d'utilisation	Consulter planning des rendez-vous
L'acteur principal	Assistant.
Objectif	Consultation du planning des rendez-vous de chaque jour.
Pré-condition	L'assistant doit s'authentifier.
Post-condition	L'assistant consulte le planning.
Scénario nominal	1-Le système affiche l'accueil de l'assistant. 2-L'assistant demande le planning des rendez-vous. 3-Le système affiche le planning.

Table 3.8 – Cas d'utilisation « consulter planning des rendez-vous »

14. Cas d'utilisation « modifier valeur glycémie »

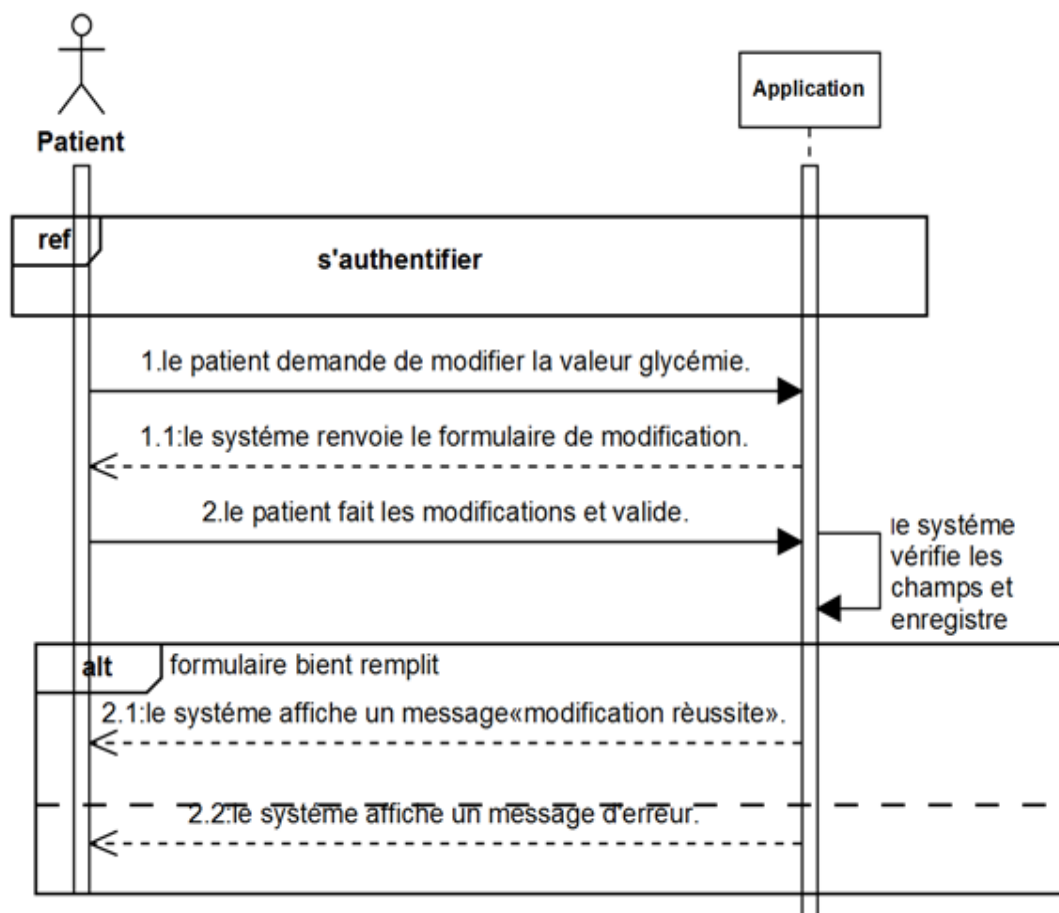


Figure 3.9 – Cas d'utilisation « modifier valeur glycémie »

15. Cas d'utilisation « consulter les dossiers médicaux »

Cas d'utilisation	Consulter les dossiers médicaux
L'acteur principal	Médecin.
Objectif	Consultation les dossiers médicaux des patients.
Pré-condition	Le médecin doit s'authentifier.
Post-condition	Le médecin consulte les dossiers médicaux de ses patients.
Scénario nominal	1-Le médecin demande au système la liste des patients. 2-Le système affiche la liste des patients. 3-Le médecin sélectionne le patient qui veut consulter son dossier médicale. 4-Le système affiche le dossier médicale (glycémie 3 mois, le graphe...etc).

Table 3.9 – Cas d'utilisation « consulter les dossiers médicaux »

16. Cas d'utilisation « bloquer profil »

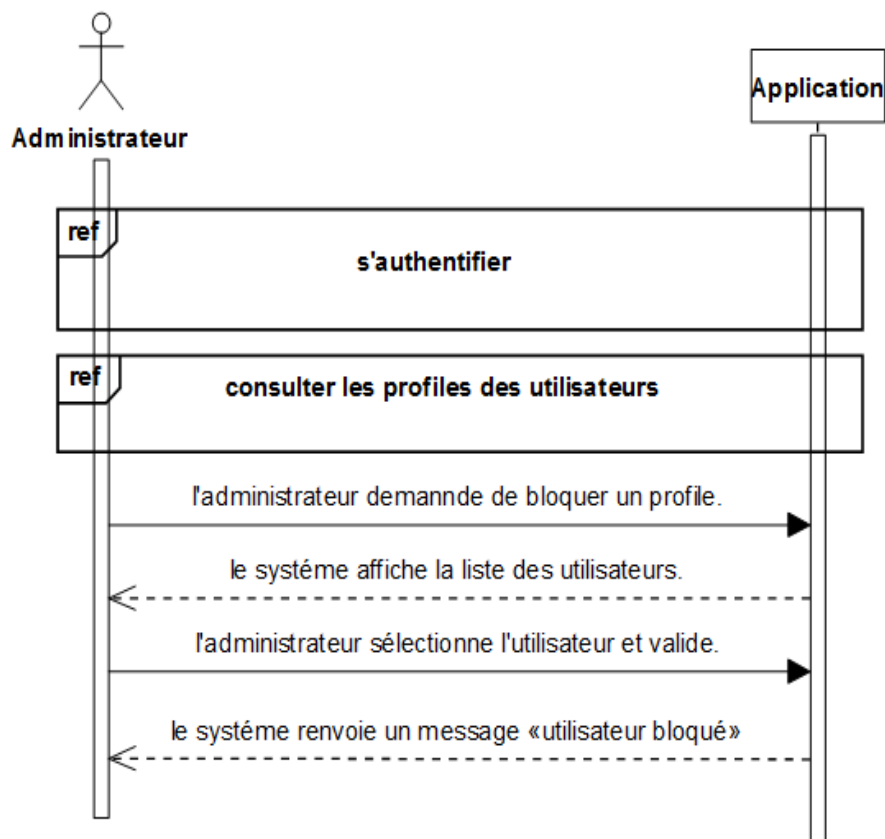


Figure 3.10 – Cas d'utilisation « bloquer profil »

17. Cas d'utilisation « modifier son profil »

Cas d'utilisation	Modifier son profil
L'acteur principal	Médecin, assistant, patient (utilisateur).
Objectif	Modification de son profil.
Pré-condition	L'utilisateur doit s'authentifier, l'utilisateur consulte le profil.
Post-condition	Profil utilisateur modifié.
Scénario nominal	1-L'utilisateur demande au système de modifier (nom utilisateur, mot de passe). 2-Le système renvoie le formulaire de modification. 3-L'utilisateur fait les modifications et enregistre. 4-Le système vérifie les informations, enregistre et affiche message de confirmation.
Scénario alternatif	4.1- L'une des informations remplies est erronée ou un champ vide, le système affiche message d'erreur. -Le scénario nominal reprend au point 2.

Table 3.10 – Cas d'utilisation « modifier son profil »

18. Cas d'utilisation « modifier profil utilisateur »

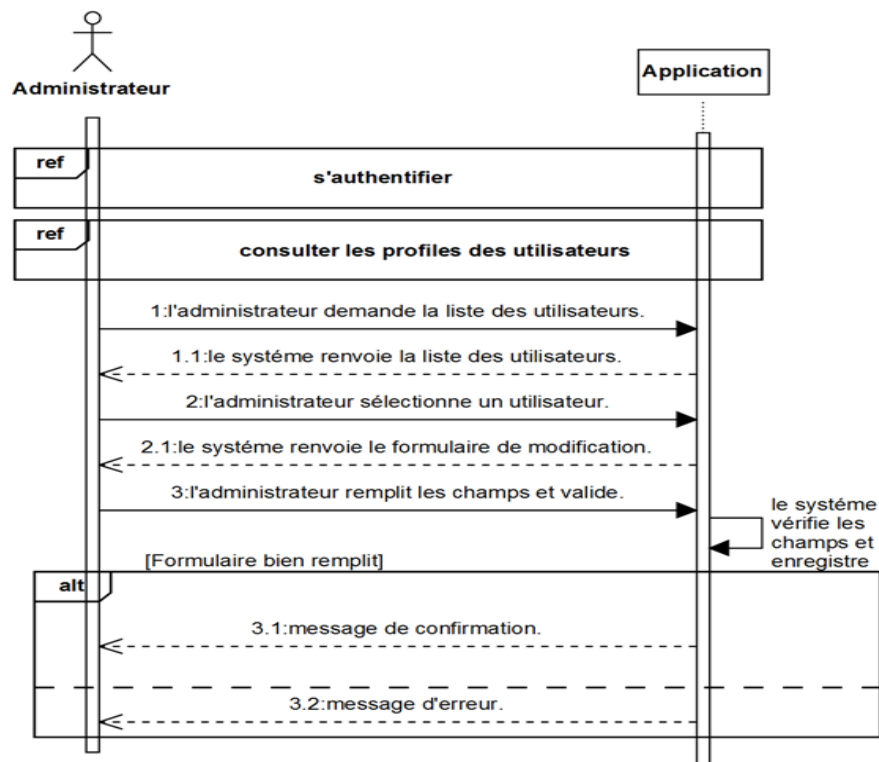


Figure 3.11 – Cas d'utilisation « modifier profil utilisateur »

19. Cas d'utilisation « consulter son dossier médicale »

Cas d'utilisation	Consulter son dossier médicale.
L'acteur principal	Patient.
Objectif	La consultation du dossier médicale
Pré-condition	Le compte patient existe dans la base de données, patient s'authentifie.
Post-condition	dossier médicale utilisateur consulté.
Scénario nominal	1-Le patient demande de consulter son dossier médicale. 2-Le système affiche les informations.

Table 3.11 – Cas d'utilisation « consulter son dossier médicale »

3.1.3 Diagrammes d'activité des différents cas d'utilisation

Pour documenter les cas d'utilisations, nous avons choisi le diagramme d'activité, car il permet de consolider les enchaînements de la fiche textuelle, comme nous l'avons représenté informellement. Ce diagramme est également très utile en cas d'actions parallèles. De plus, les utilisateurs le comprennent aisément, car il ressemble à un organigramme traditionnel. Il permet enfin d'identifier d'un seul coup d'œil la famille des scénarios d'un cas d'utilisation qui décrivent toutes les réactions du système. Il suffit en effet de dessiner les différents chemins du diagramme d'activité qui passent par toutes les transitions entre actions.[14]

1. Cas d'utilisation « s'authentifier »

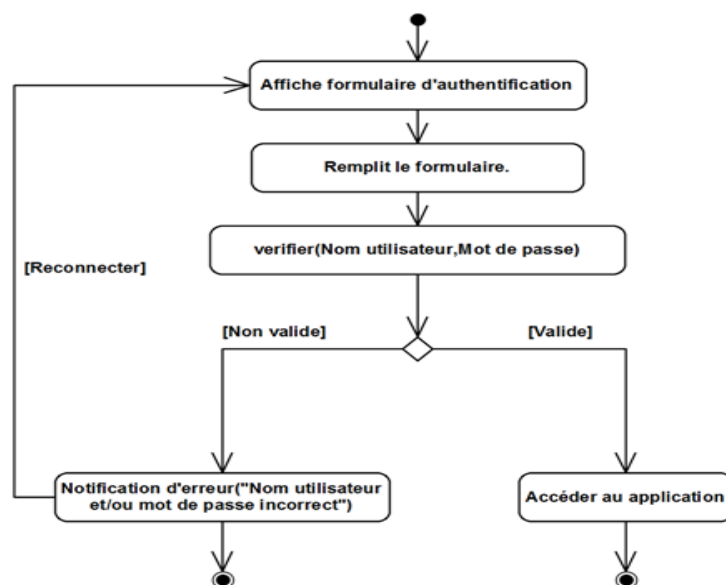


Figure 3.12 – Cas d'utilisation « s'authentifier »

2. Cas d'utilisation « ajouter valeur glycémie »

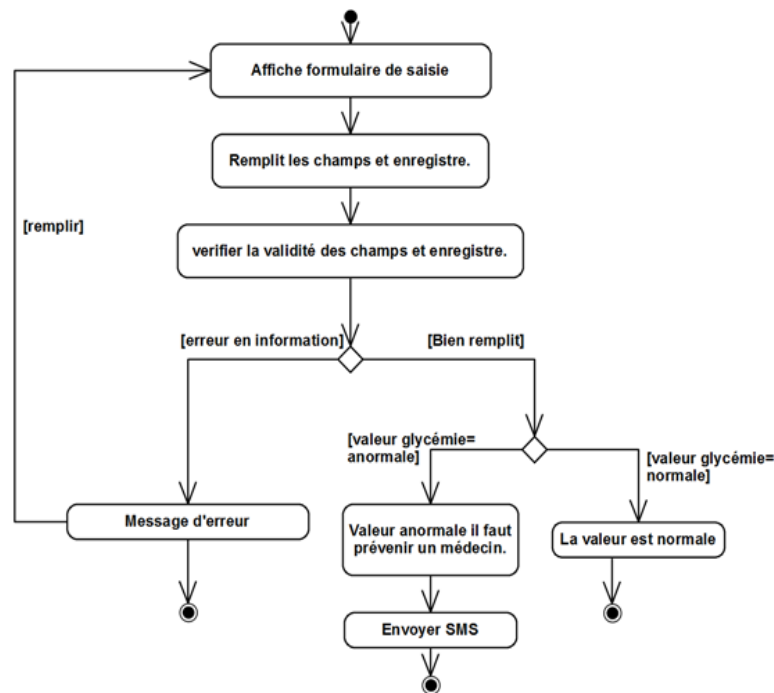


Figure 3.13 – Cas d'utilisation « ajouter valeur glycémie »

3. Cas d'utilisation « créer profil médecin »

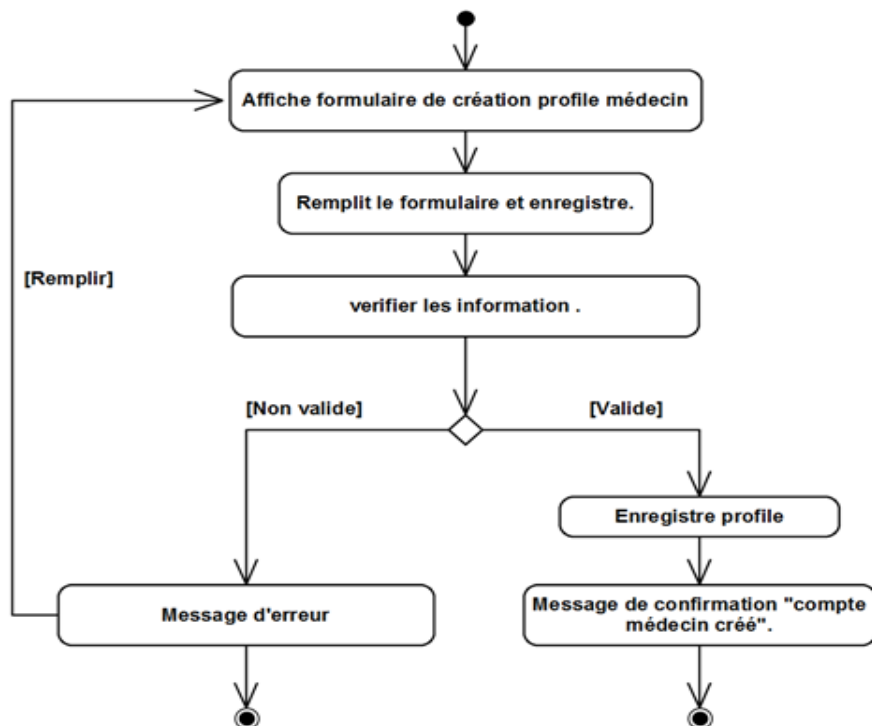


Figure 3.14 – Cas d'utilisation « créer profil médecin »

4. Cas d'utilisation « annuler rendez-vous »

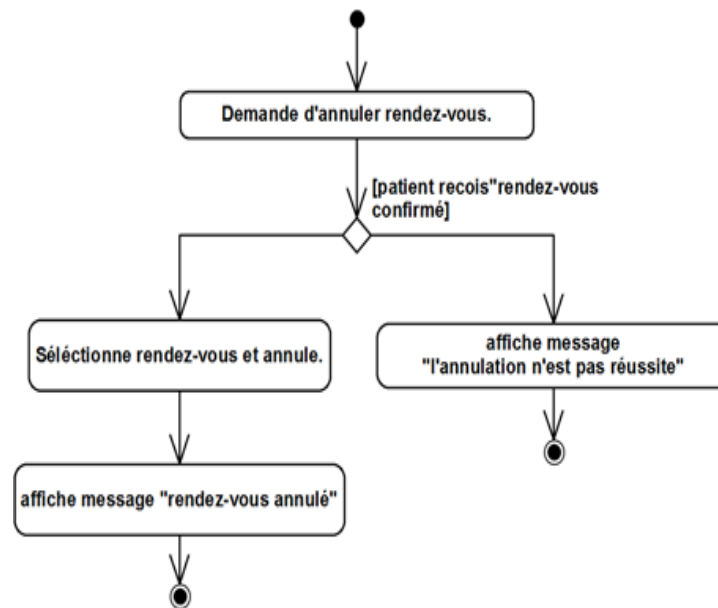


Figure 3.15 – Cas d'utilisation « annuler rendez-vous »

5. Cas d'utilisation « consulter son profil »

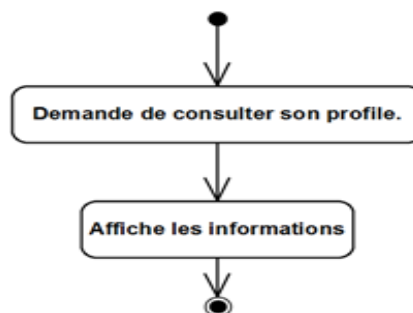


Figure 3.16 – Cas d'utilisation « consulter son profil »

6. Cas d'utilisation « prendre rendez-vous »

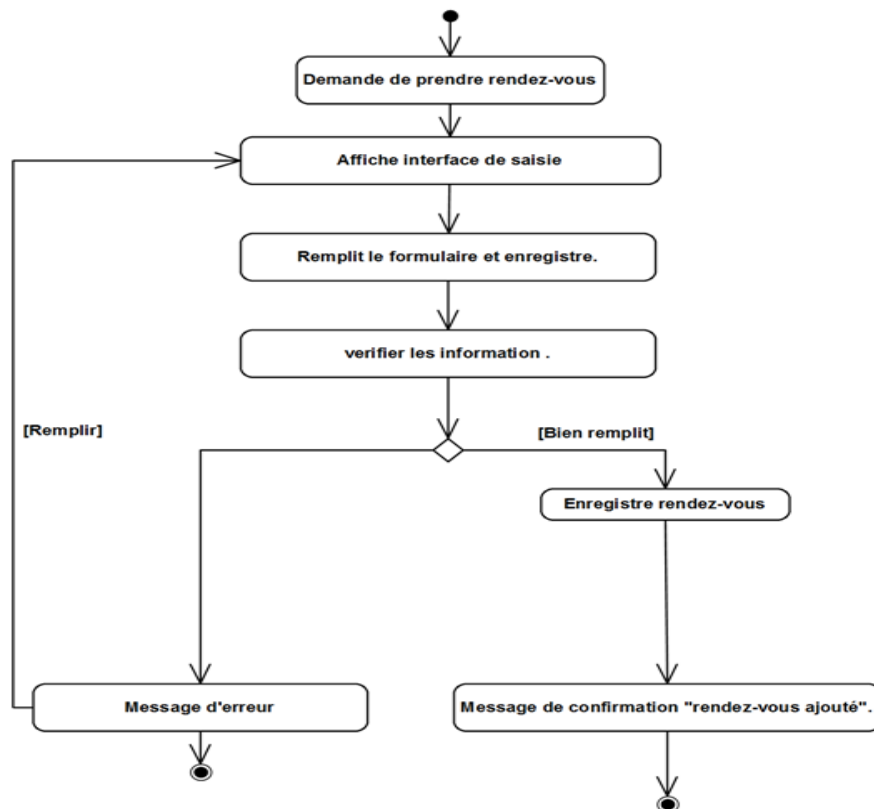


Figure 3.17 – Cas d'utilisation « prendre rendez-vous »

7. Cas d'utilisation « consulter les profils utilisateurs »

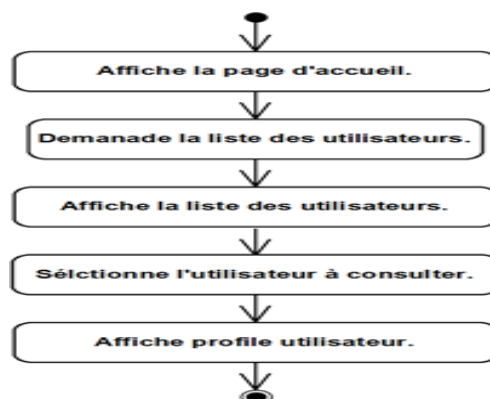


Figure 3.18 – Cas d'utilisation « consulter les profils utilisateurs »

8. Cas d'utilisation « modifier les dossiers médicaux »

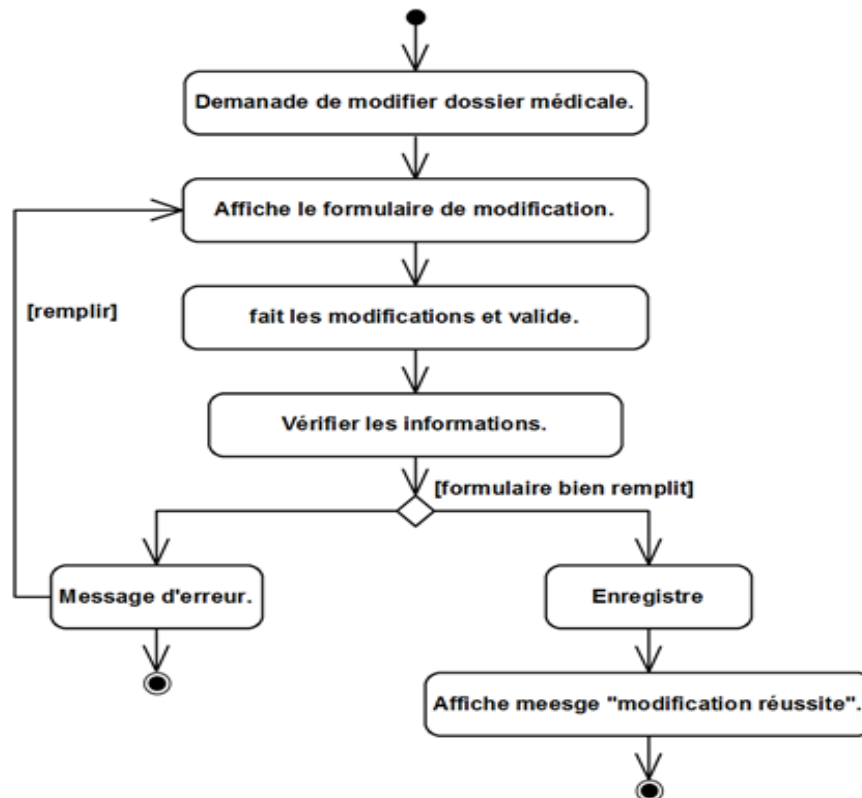


Figure 3.19 – Cas d'utilisation « modifier les dossiers médicaux »

9. Cas d'utilisation « confirmer rendez-vous »

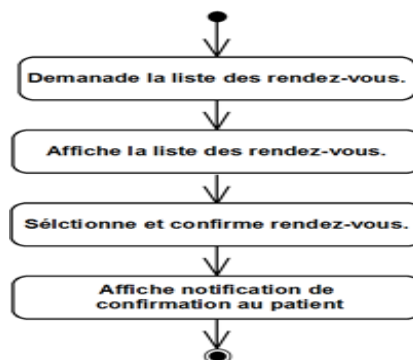


Figure 3.20 – Cas d'utilisation « confirmer rendez-vous »

10. Cas d'utilisation « créer profil assistant »

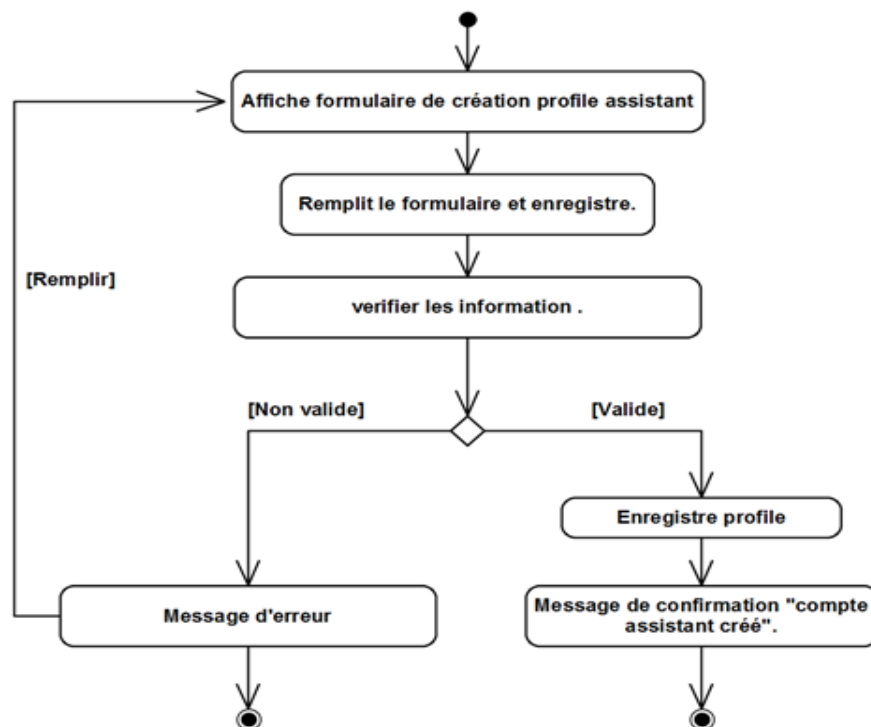


Figure 3.21 – Cas d'utilisation « créer profil assistant »

11. Cas d'utilisation « saisir analyses 3 mois »

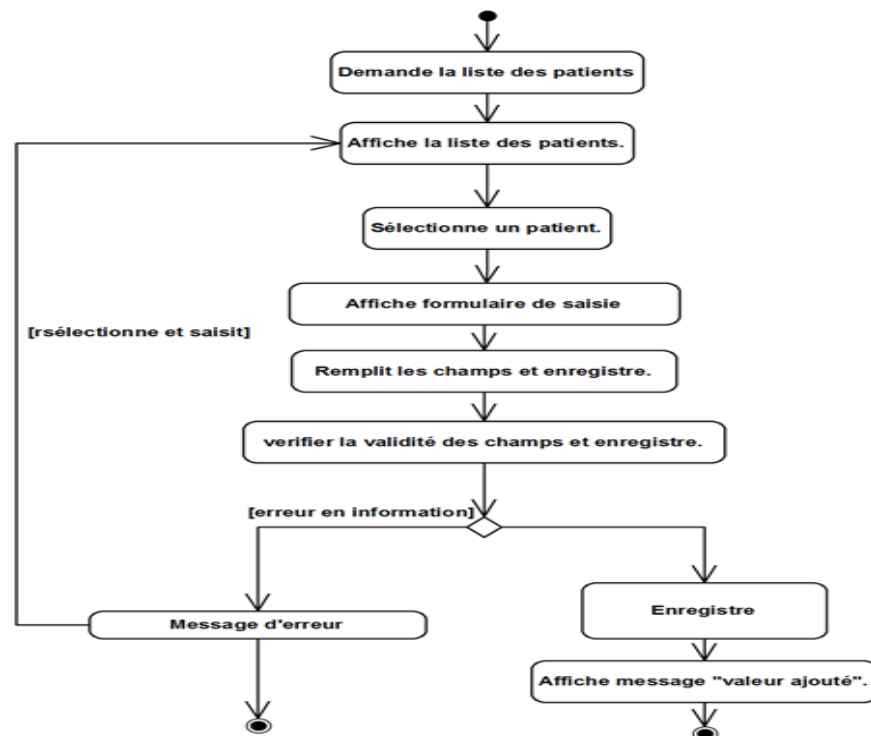


Figure 3.22 – Cas d'utilisation « saisir analyses 3 mois »

12. Cas d'utilisation « créer profil patient »

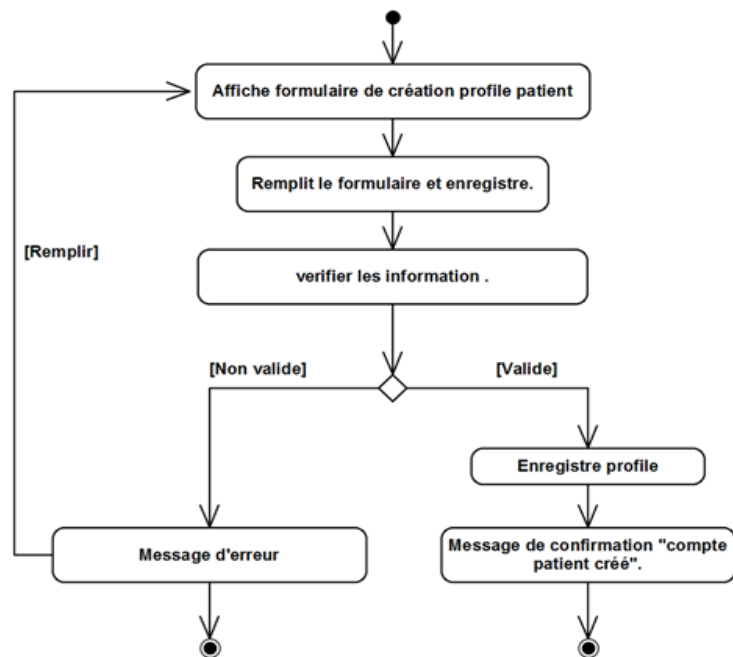


Figure 3.23 – Cas d'utilisation « créer profil patient »

13. Cas d'utilisation « consulter planning des rendez-vous »

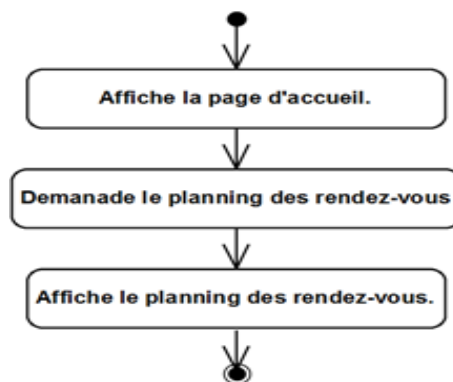


Figure 3.24 – Cas d'utilisation « consulter planning des rendez-vous »

14. Cas d'utilisation « modifier valeur glycémie »

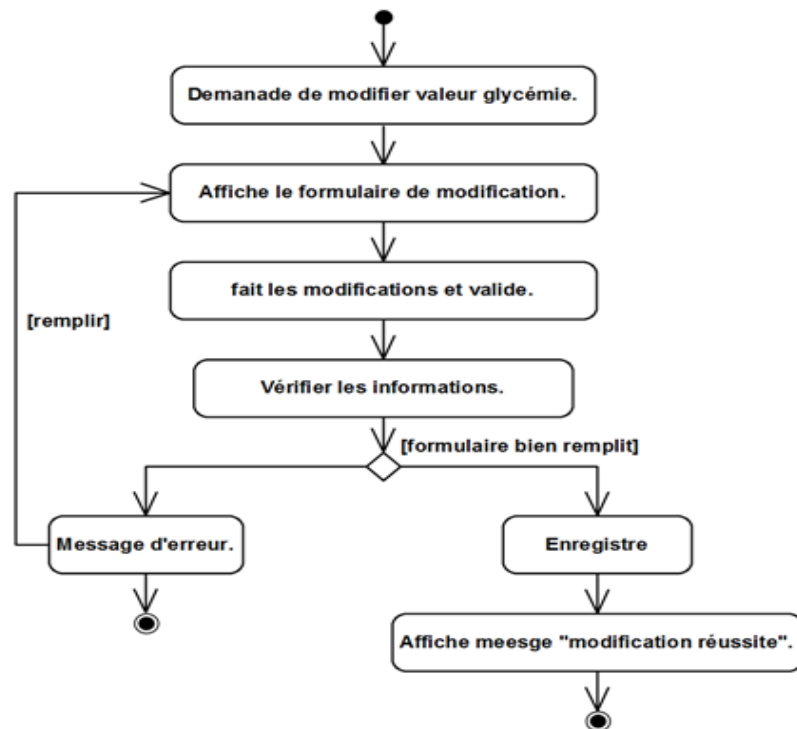


Figure 3.25 – Cas d'utilisation « modifier valeur glycémie »

15. Cas d'utilisation « consulter les dossiers médicaux »

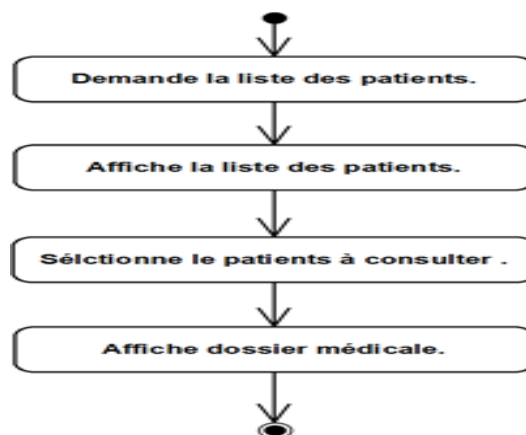


Figure 3.26 – Cas d'utilisation « consulter les dossiers médicaux »

16. Cas d'utilisation « bloquer profil »

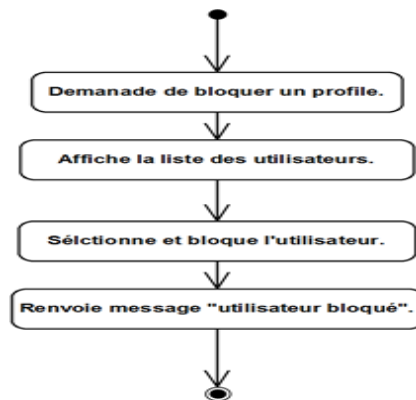


Figure 3.27 – Cas d'utilisation « bloquer profil »

17. Cas d'utilisation « modifier son profil »

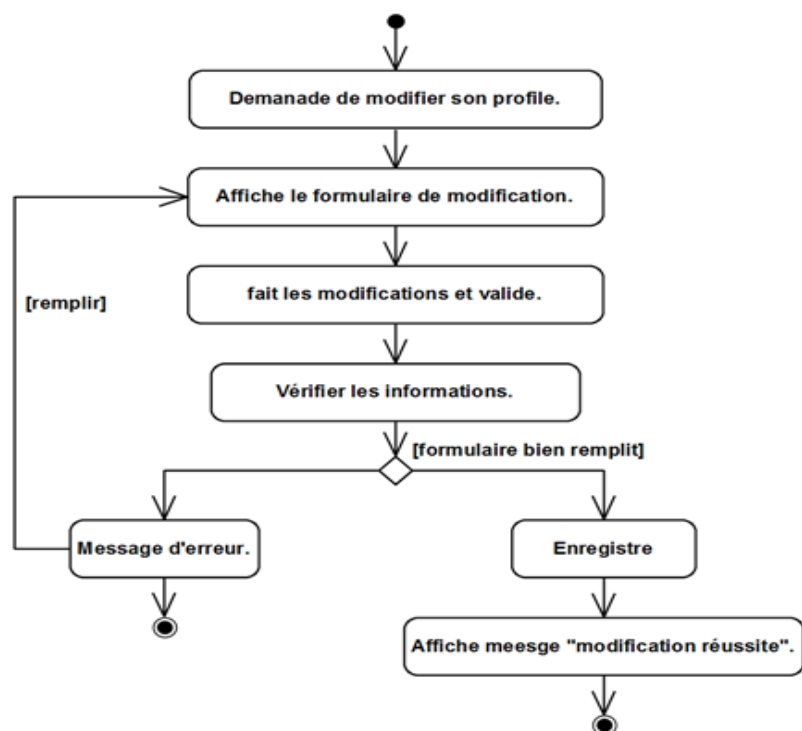


Figure 3.28 – Cas d'utilisation « modifier son profil »

18. Cas d'utilisation « modifier les profils utilisateurs »

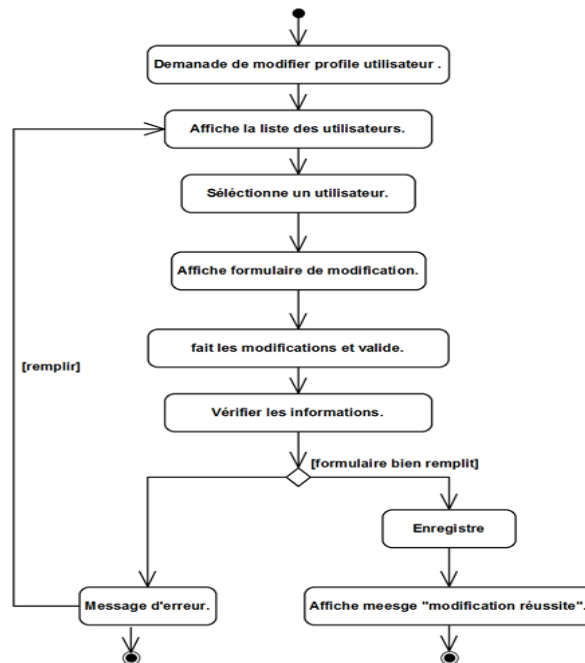


Figure 3.29 – Cas d'utilisation « modifier les profils utilisateurs »

19. Cas d'utilisation « consulter son dossier médicale »

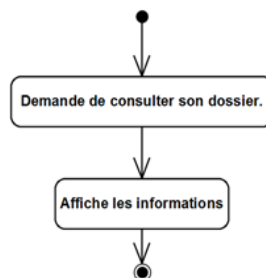


Figure 3.30 – Cas d'utilisation « consulter son dossier médicale »

3.1.4 Identification des classes candidates

Cette phase va préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d'analyse. La technique utilisée pour identifier les classes candidates est la suivante :

- Chercher les noms communs importants dans les descriptions textuelles des cas d'utilisation.

- Vérifier les propriétés « objet » de chaque concept (identité, propriétés, comportement), puis définir ses responsabilités.[11]

- **Définition :**

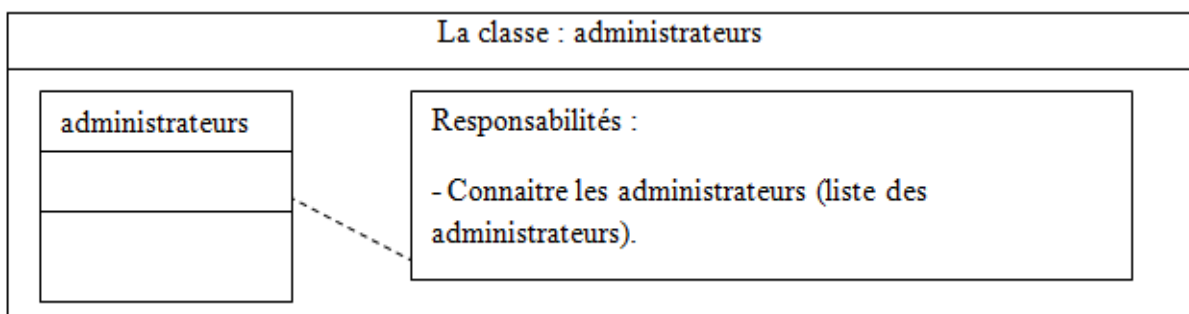
Une responsabilité est une sorte de contrat ou obligation pour une classe. Elle se place à un niveau d'abstraction plus élevé que les attributs ou les opérations. On formalise ensuite ces concepts métier sous forme de classes et d'associations rassemblées dans un diagramme statique pour chaque cas d'utilisation.[11]

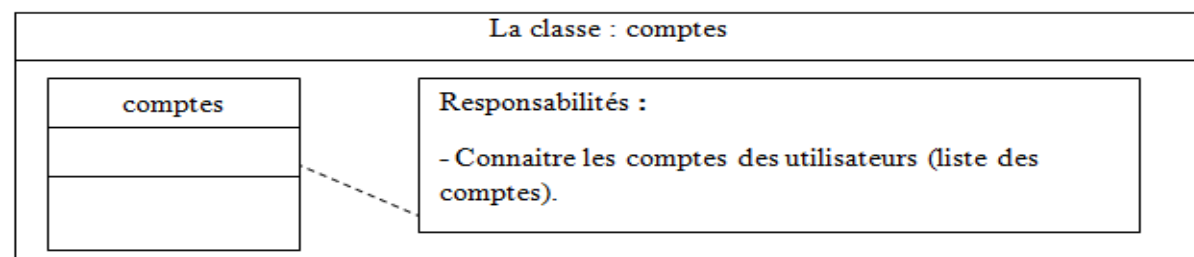
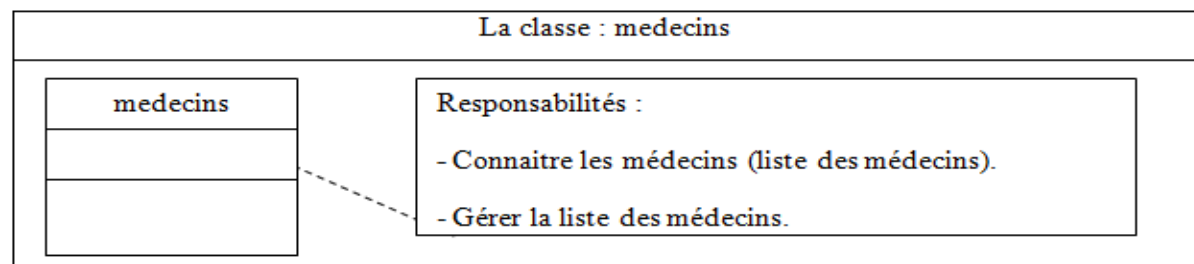
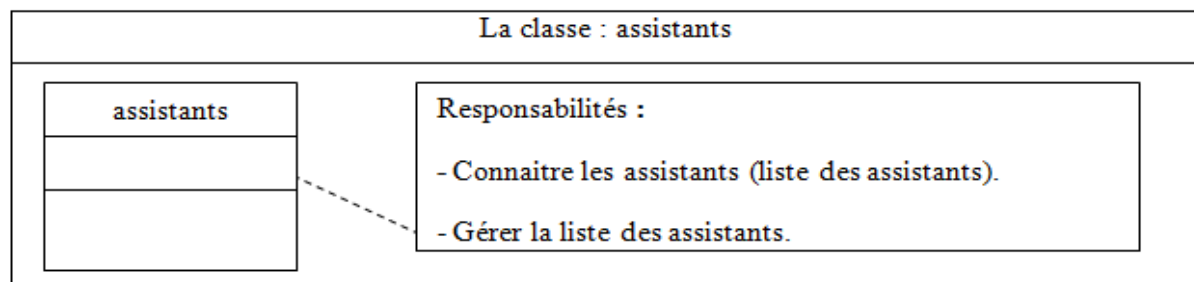
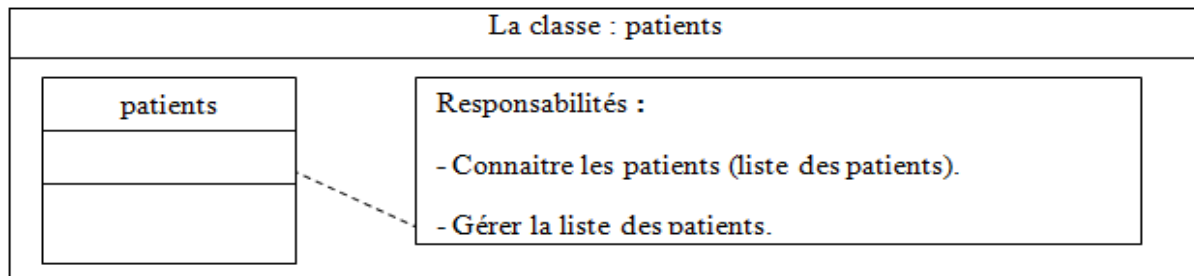
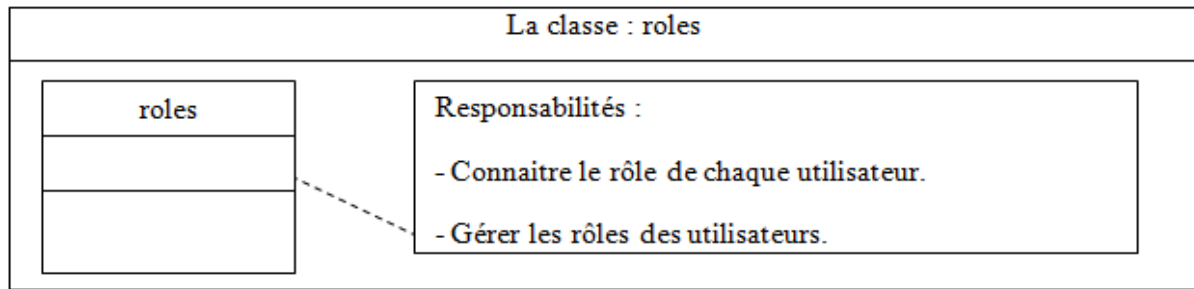
Ces diagrammes préliminaires sont appelés « diagrammes des classes participantes », ils n'ont pas pour objectif d'être complet. Ils servent uniquement à démarrer la découverte des classes du modèle d'analyse. [14]

- **La liste des classes candidates :**

- * administrateurs.
- * roles.
- * patients.
- * assistants.
- * medecins.
- * comptes.
- * glycemies.
- * dossier_medicales.
- * rendezvous.

- **Responsabilités des classes :**





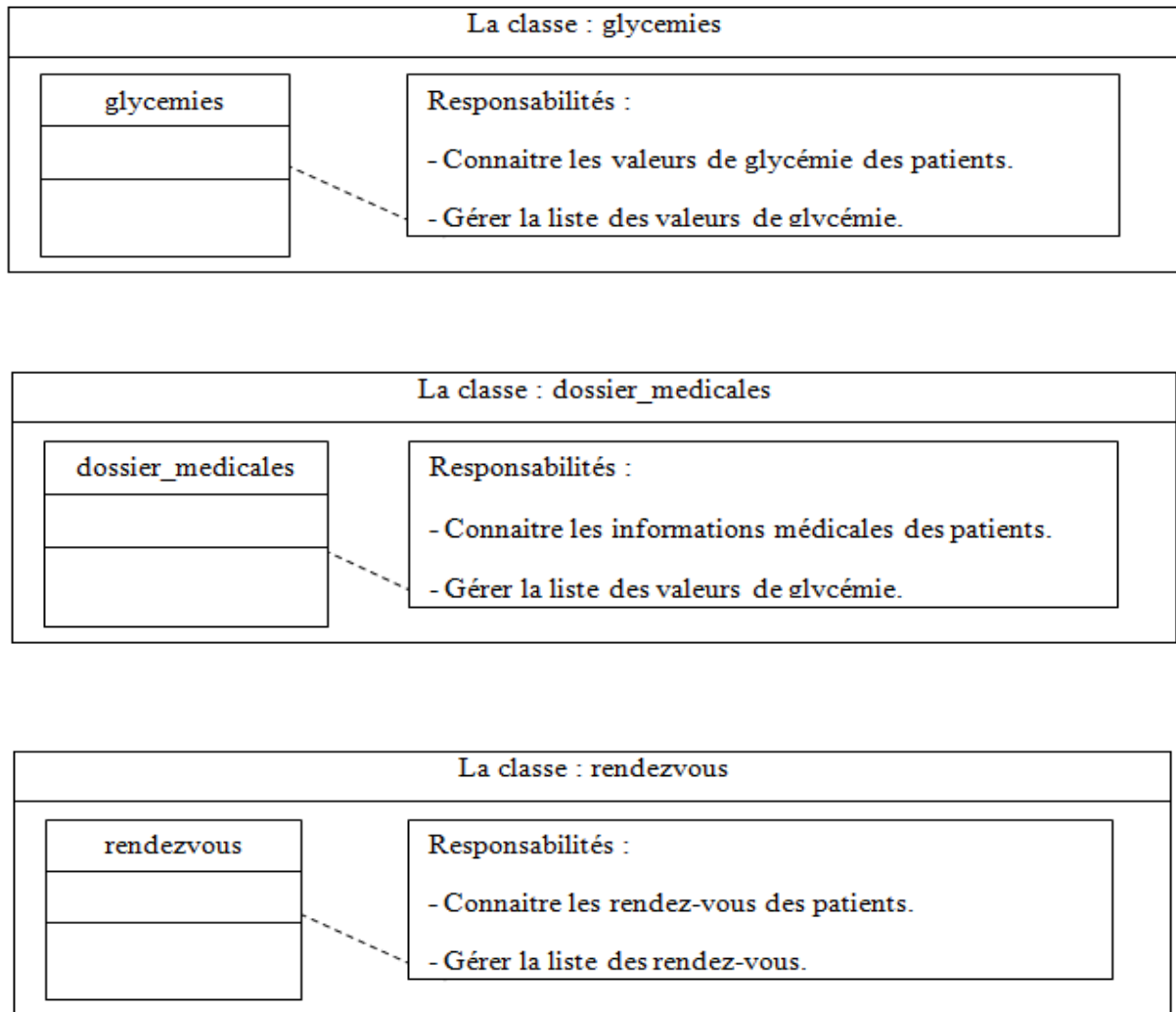


Figure 3.31 – Les classes candidates et ses responsabilités

3.1.5 Diagrammes de classe participante des cas d'utilisations

Le diagramme de classe participante est important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, les modèles de la couche métiers et l'interface avec l'utilisateur. Il semble particulièrement important pour guider la phase de production du livrable final. C'est cette importance qui nous a poussés à concevoir un tel diagramme dans le souci d'une phase de développement claire et efficace. On utilisera alors une implémentation de l'architecture 2-tiers, le pattern Modèle-Vue-Contrôleur (MVC).[14]

1. Cas d'utilisation « s'authentifier »

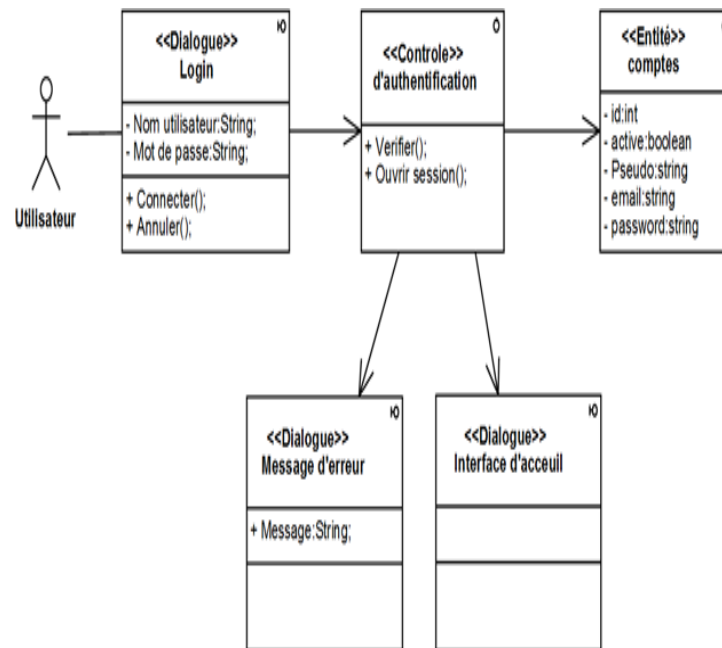


Figure 3.32 – Cas d'utilisation « s'authentifier »

2. Cas d'utilisation « ajouter valeur glycémie »

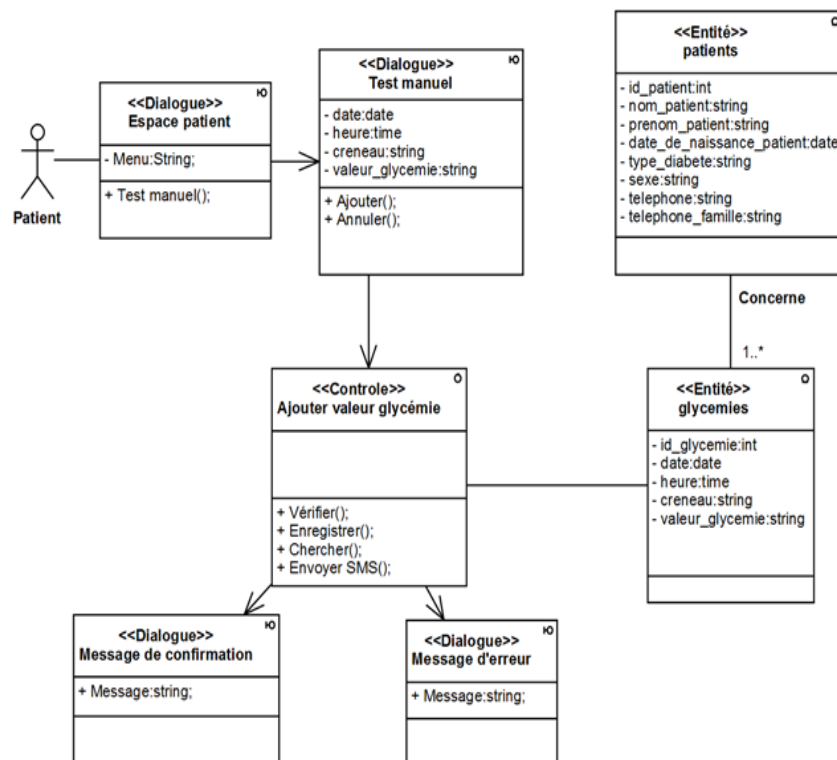


Figure 3.33 – Cas d'utilisation « ajouter valeur glycémie »

3. Cas d'utilisation « créer profil médecin »

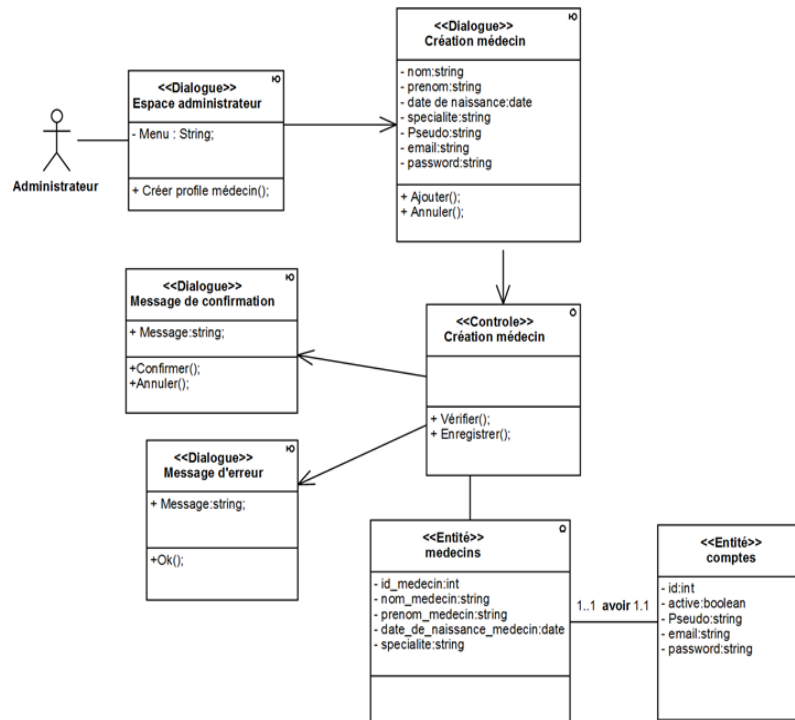


Figure 3.34 – Cas d'utilisation « créer profil médecin »

4. Cas d'utilisation « annuler rendez-vous »

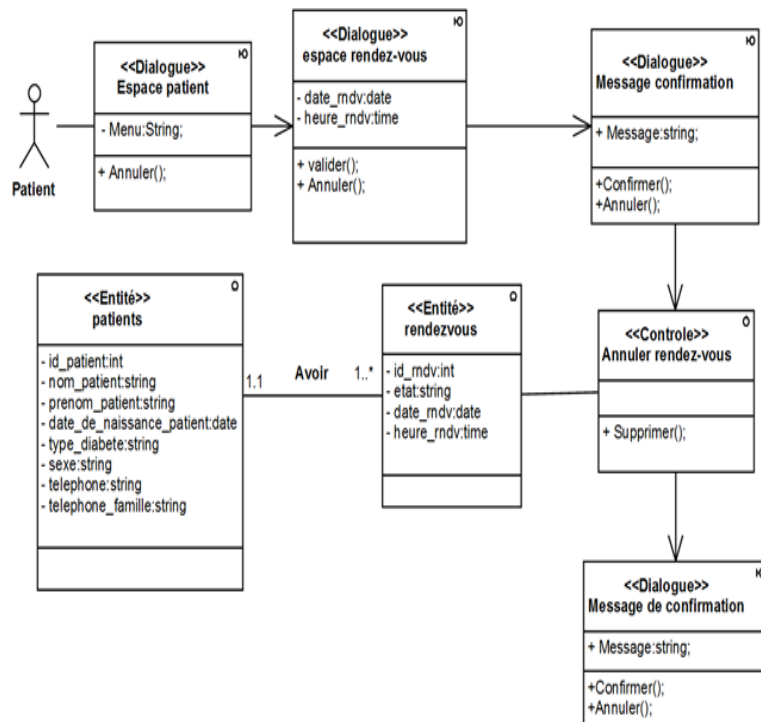


Figure 3.35 – Cas d'utilisation « annuler rendez-vous »

5. Cas d'utilisation « consulter son profil »

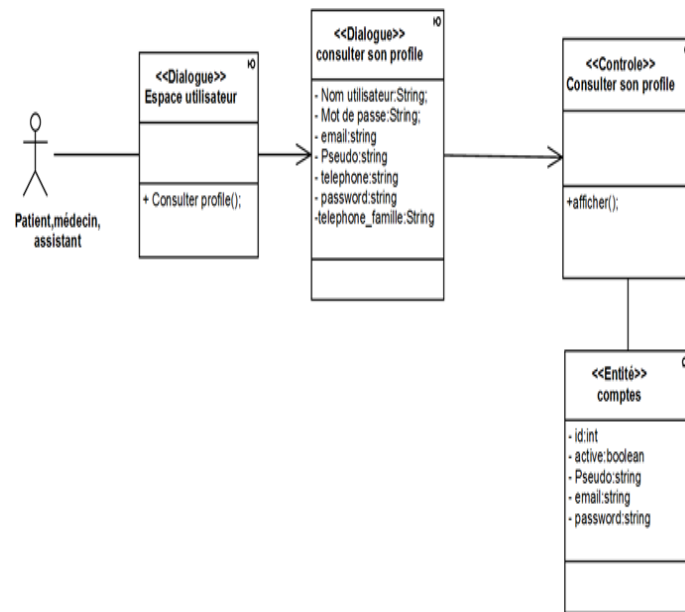


Figure 3.36 – Cas d'utilisation « consulter son profil »

6. Cas d'utilisation « prendre rendez-vous »

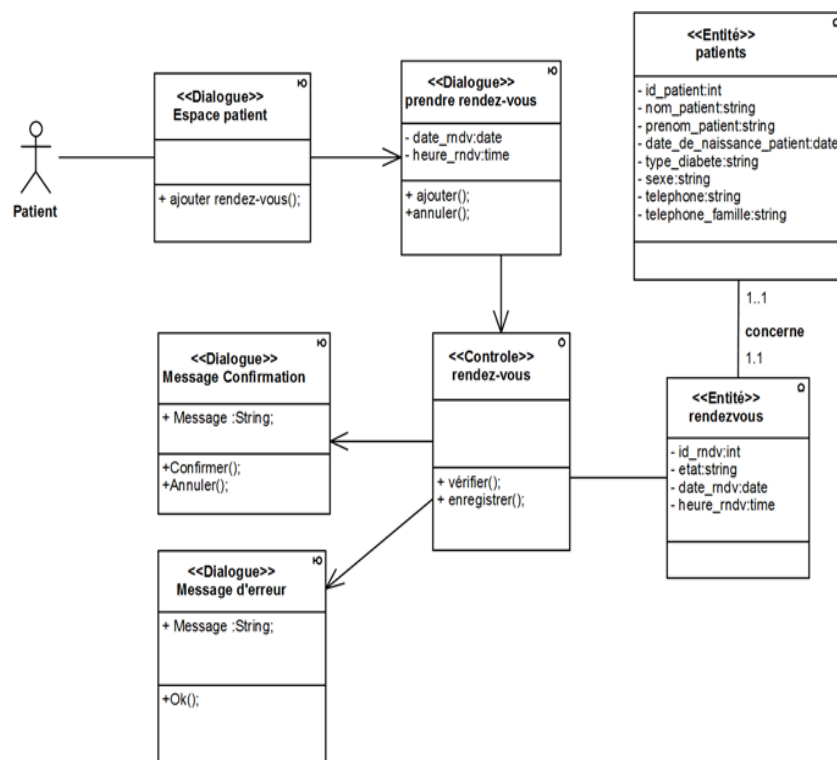


Figure 3.37 – Cas d'utilisation « prendre rendez-vous »

7. Cas d'utilisation « consulter les profils utilisateurs »

• Cas d'utilisation « consulter profils(médecins) »

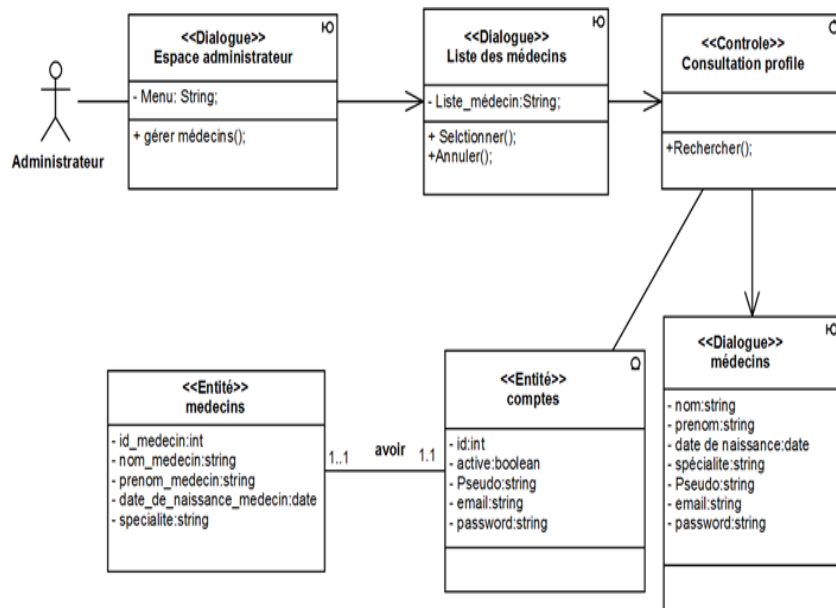


Figure 3.38 – Cas d'utilisation « consulter profils(médecins) »

• Cas d'utilisation « consulter profils(patients) »

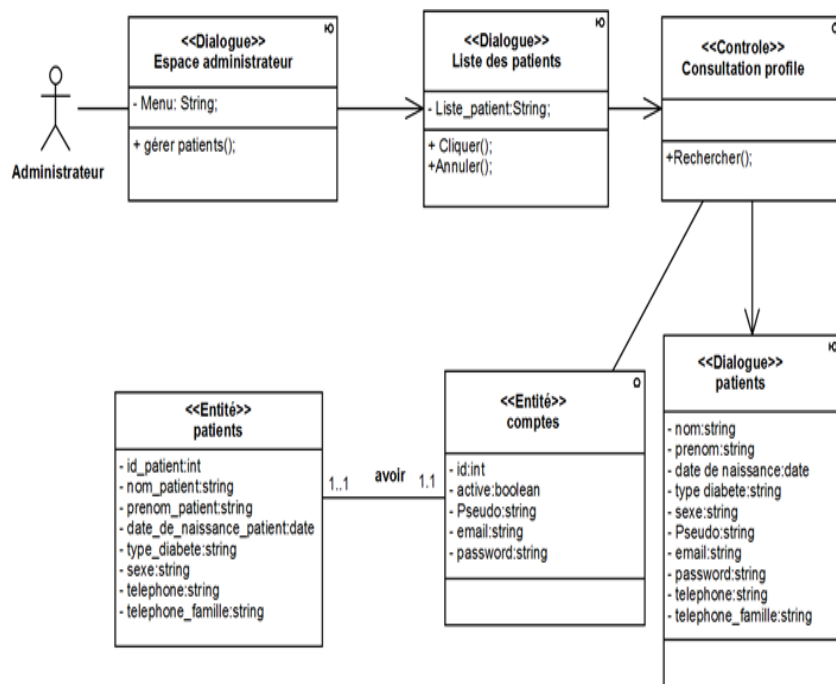


Figure 3.39 – Cas d'utilisation « consulter profils(patients) »

• Cas d'utilisation « consulter profils(assistants) »

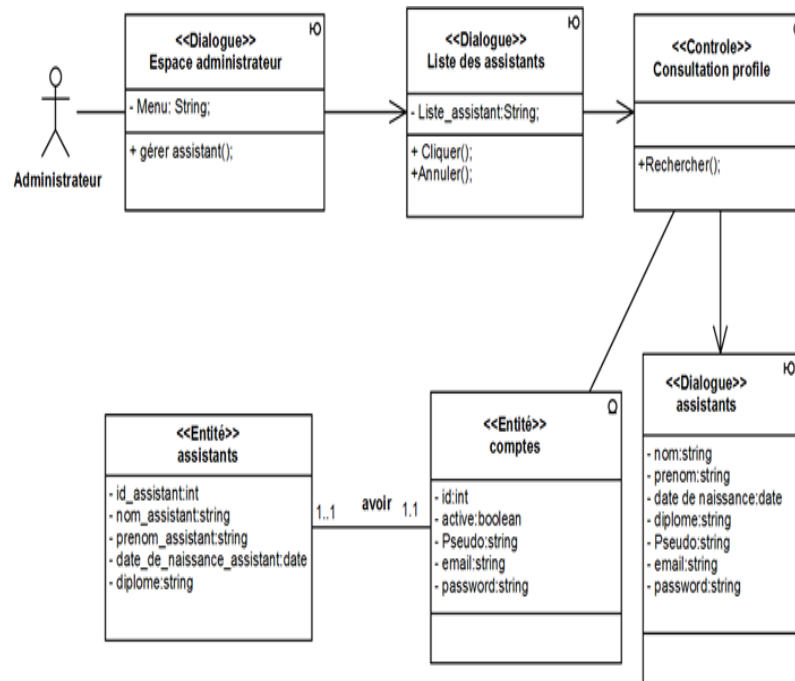


Figure 3.40 – Cas d'utilisation « consulter profils(assistants) »

8. Cas d'utilisation « modifier les dossiers médicaux »

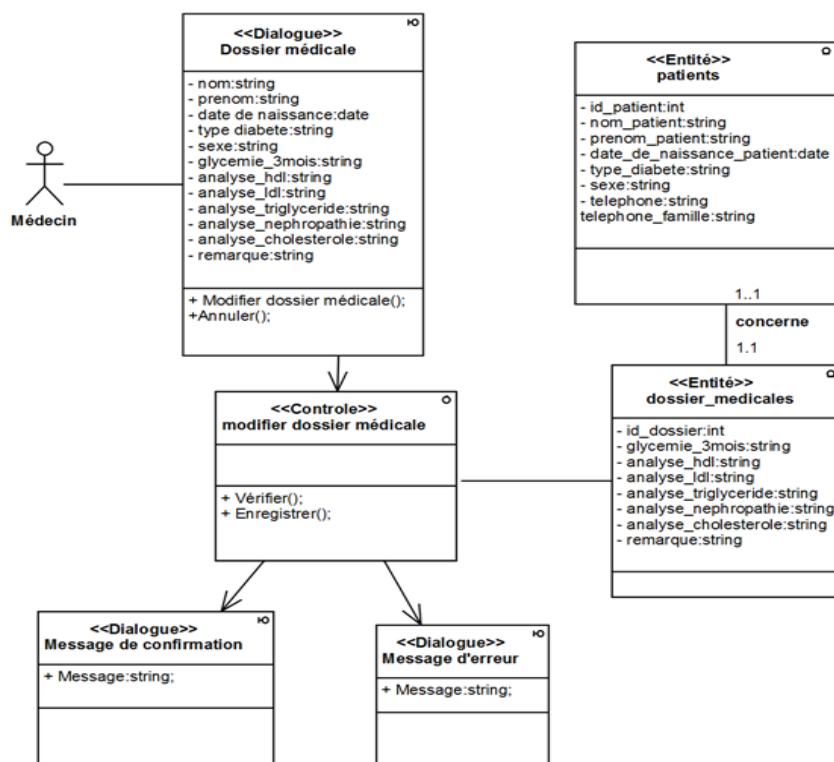


Figure 3.41 – Cas d'utilisation « modifier les dossiers médicaux »

9. Cas d'utilisation « confirmer rendez-vous »

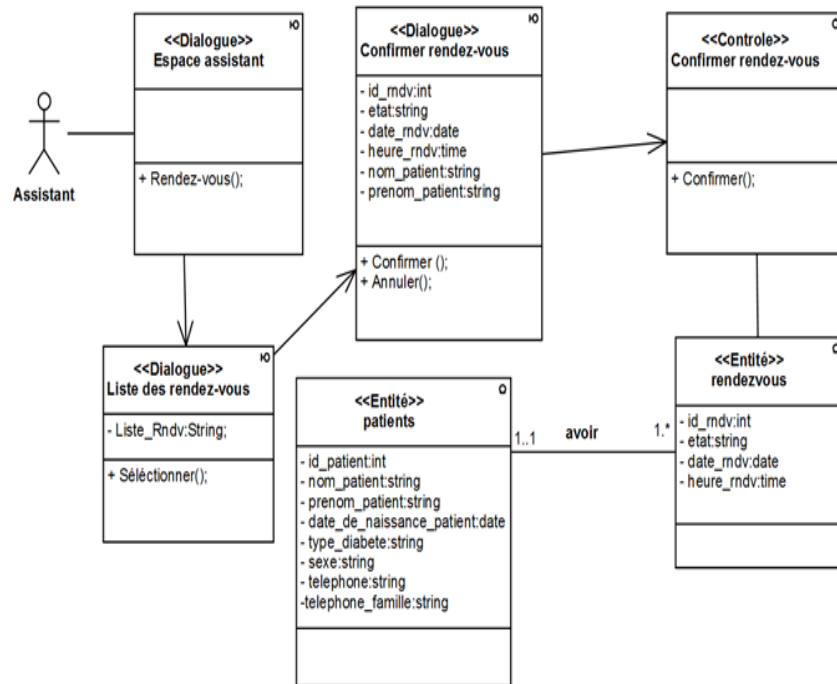


Figure 3.42 – Cas d'utilisation « confirmer rendez-vous »

10. Cas d'utilisation « créer profil assistant »

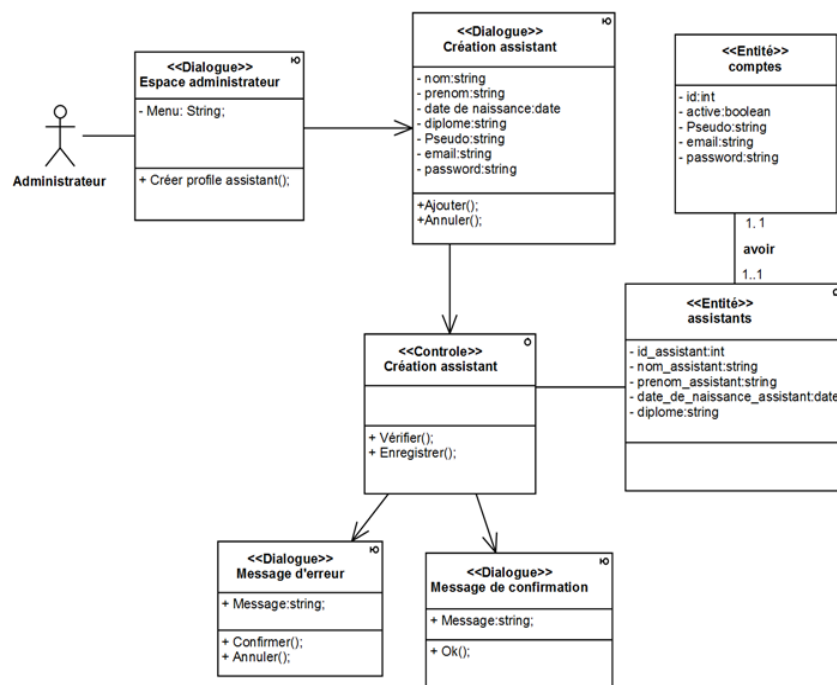


Figure 3.43 – Cas d'utilisation « créer profil assistant »

11. Cas d'utilisation « saisir analyses 3 mois »

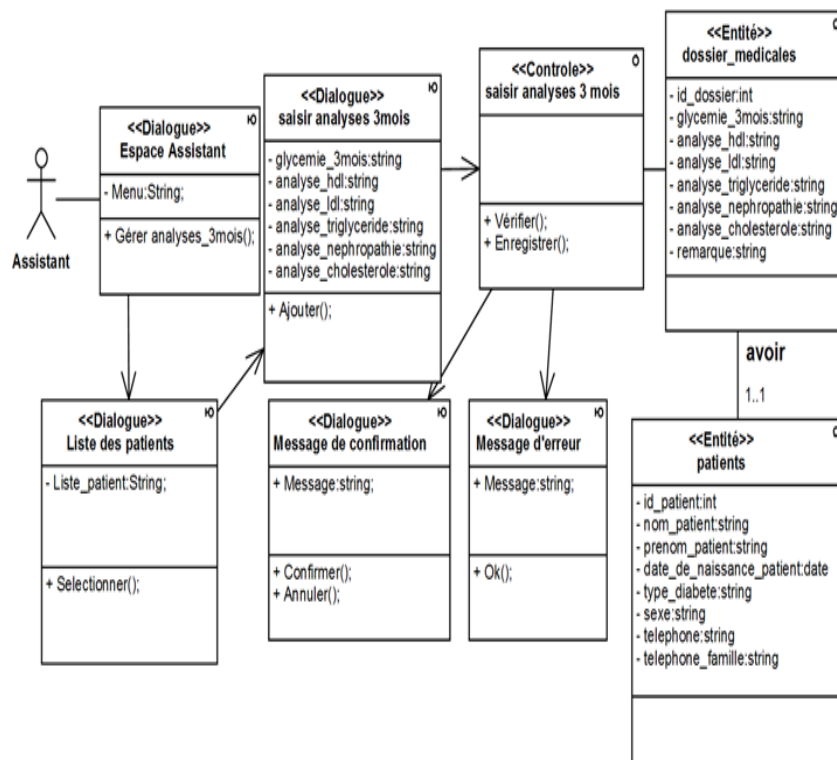


Figure 3.44 – Cas d'utilisation « saisir analyses 3 mois »

12. Cas d'utilisation « créer profil patient »

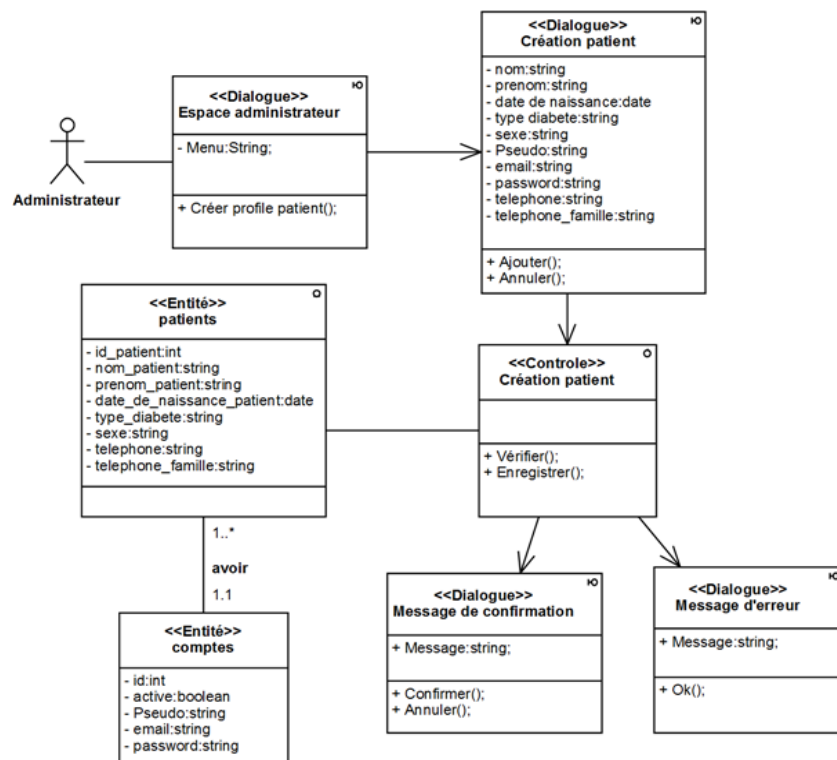


Figure 3.45 – Cas d'utilisation « créer profil patient »

13. Cas d'utilisation « consulter planning des rendez-vous »

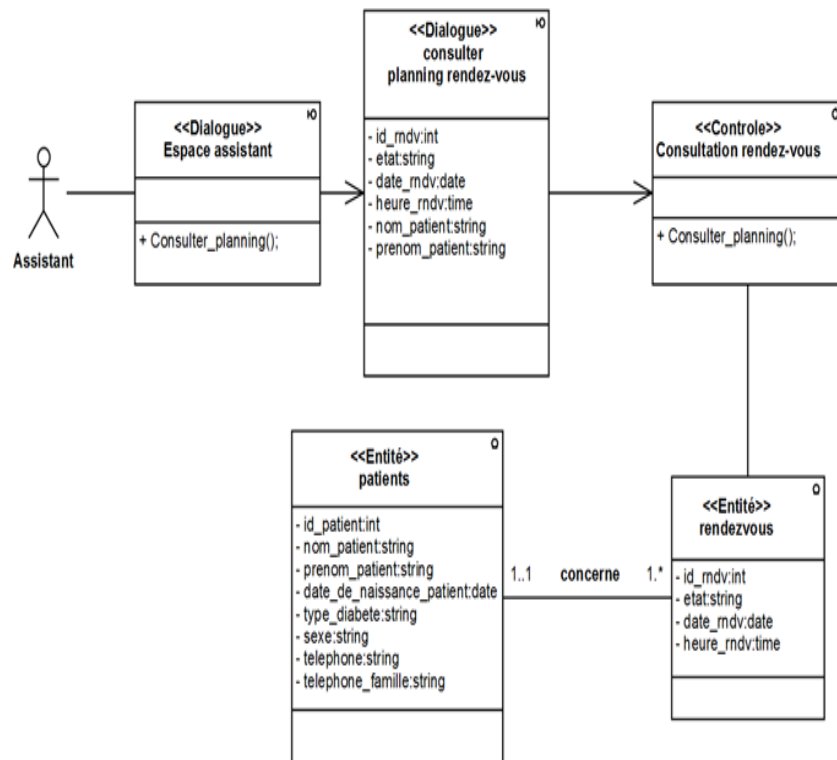


Figure 3.46 – Cas d'utilisation « consulter planning des rendez-vous »

14. Cas d'utilisation « modifier valeur glycémie »

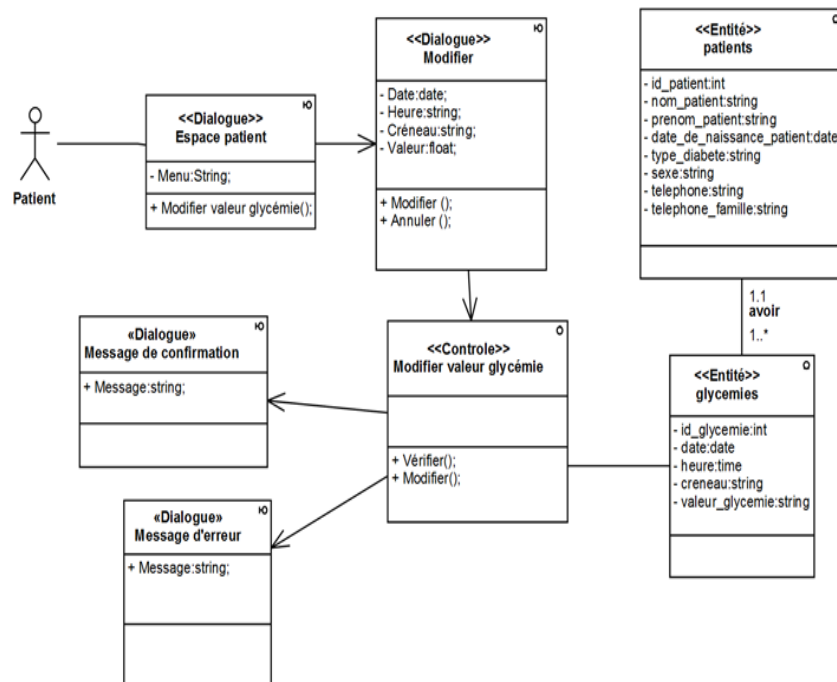


Figure 3.47 – Cas d'utilisation « modifier valeur glycémie »

15. Cas d'utilisation « consulter les dossiers médicaux »

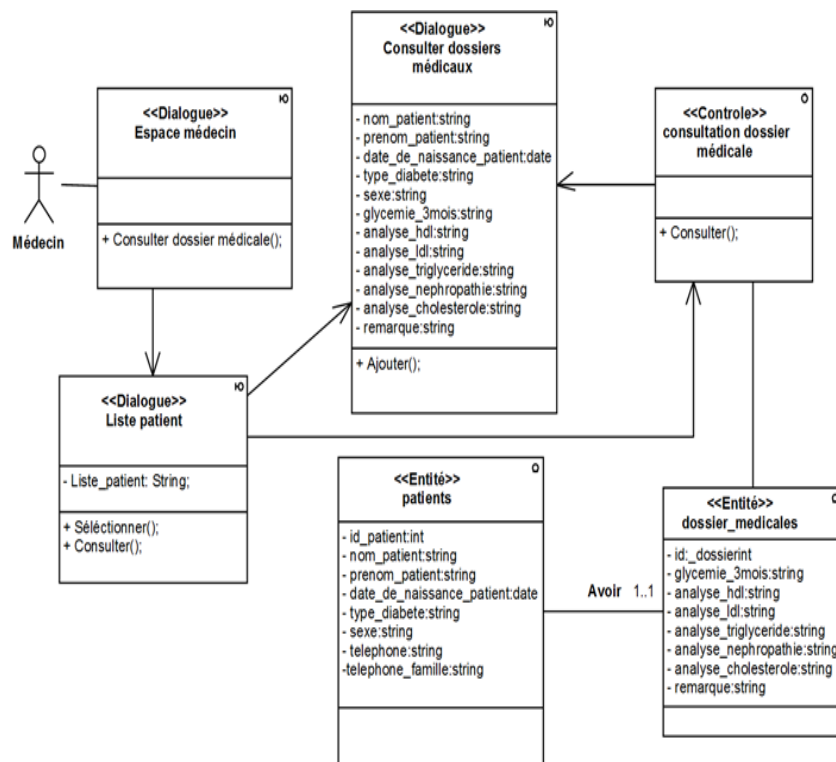


Figure 3.48 – Cas d'utilisation « consulter les dossiers médicaux »

16. Cas d'utilisation « bloquer profil »

• Cas d'utilisation « bloquer profil(médecins) »

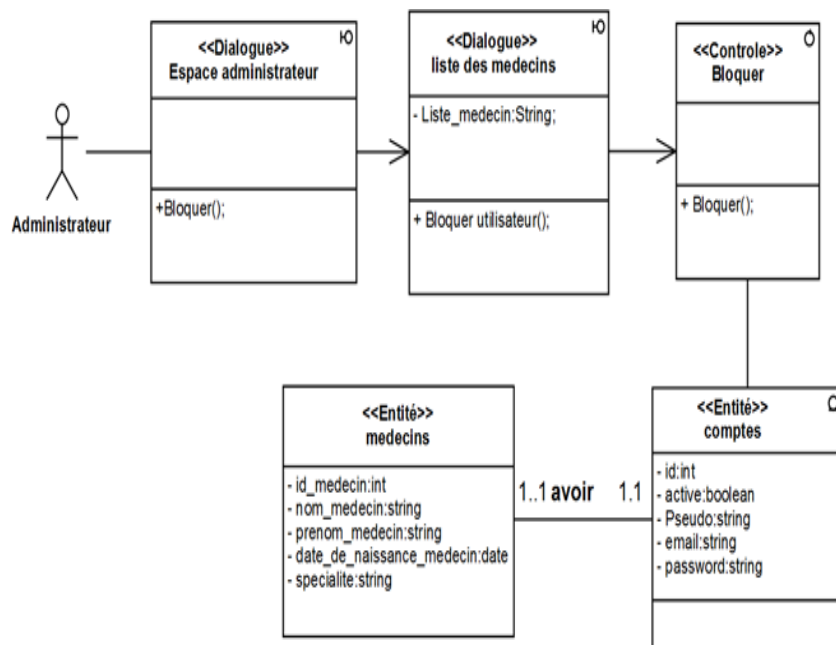


Figure 3.49 – Cas d'utilisation « bloquer profil (médecins)»

• Cas d'utilisation « bloquer profil(patients) »

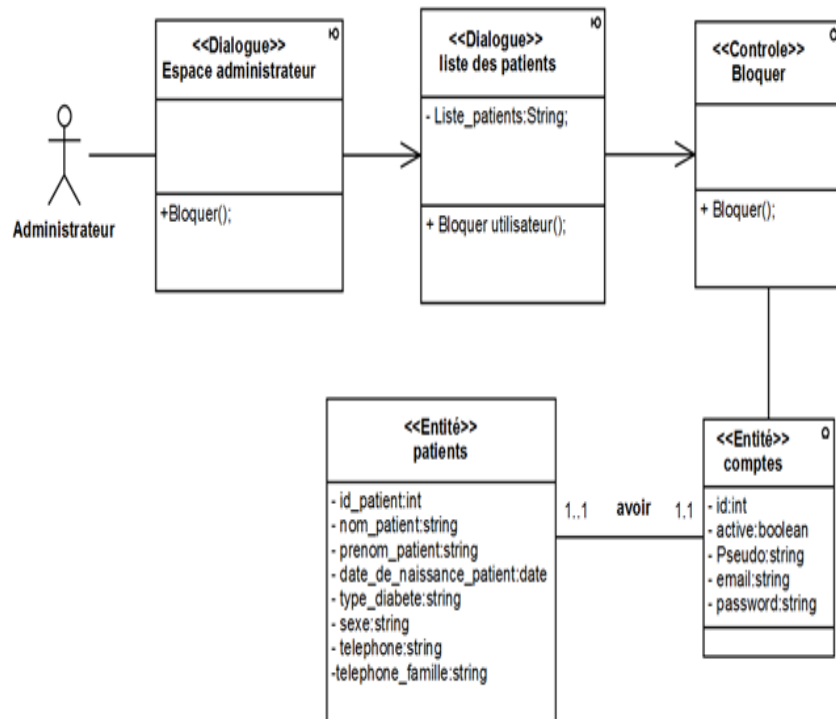


Figure 3.50 – Cas d'utilisation « bloquer profil(patients) »

• Cas d'utilisation « bloquer profil (assistants) »

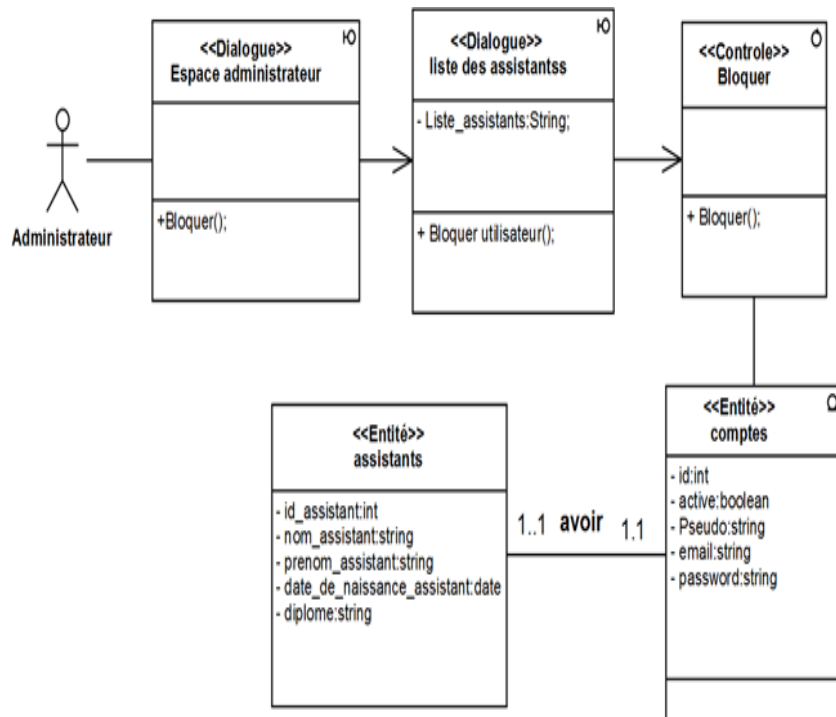


Figure 3.51 – Cas d'utilisation « bloquer profil (assistants) »

17. Cas d'utilisation « modifier son profil »

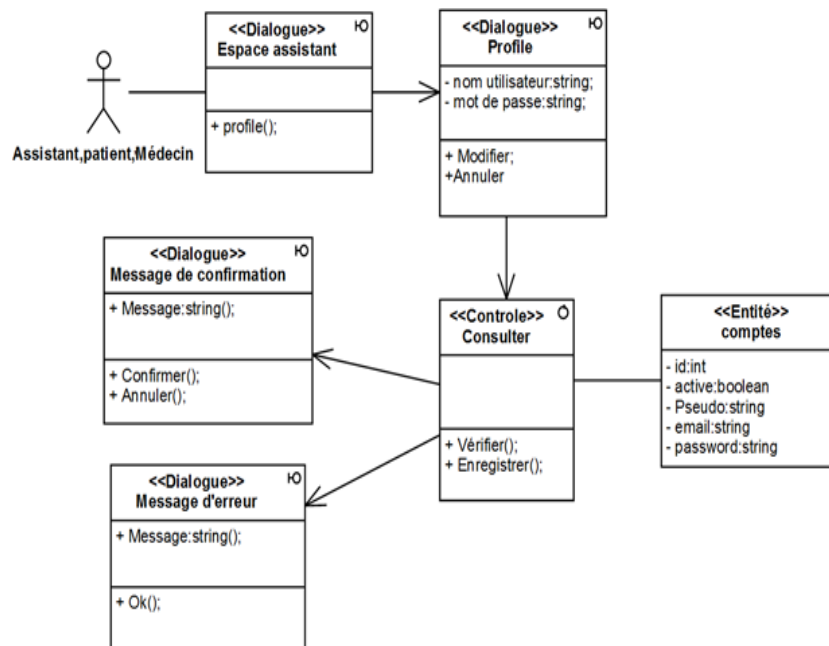


Figure 3.52 – Cas d'utilisation « modifier son profil »

18. Cas d'utilisation « modifier les profils utilisateurs »

• Cas d'utilisation « modifier profils (médecins) »

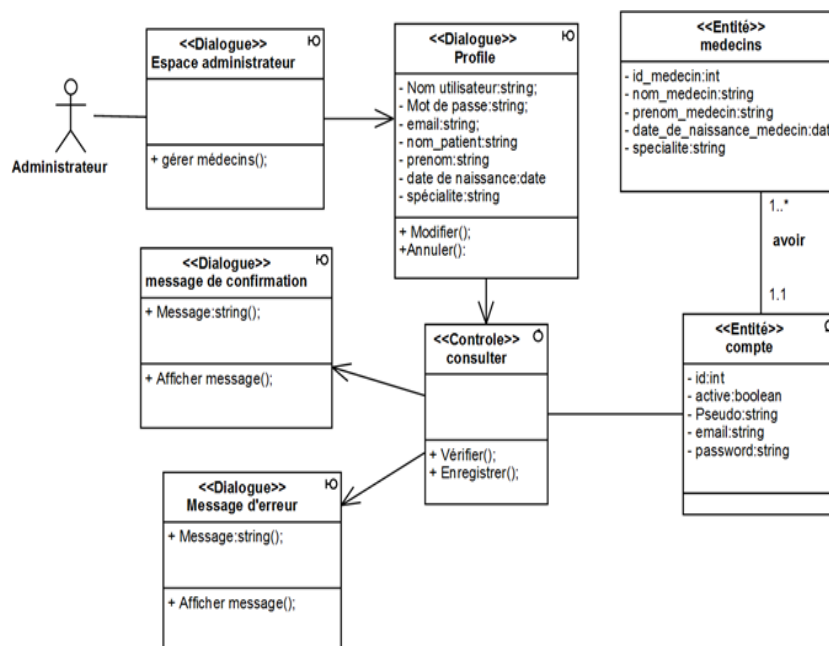


Figure 3.53 – Cas d'utilisation « modifier profils (médecins) »

• Cas d'utilisation « modifier profils(patients)»

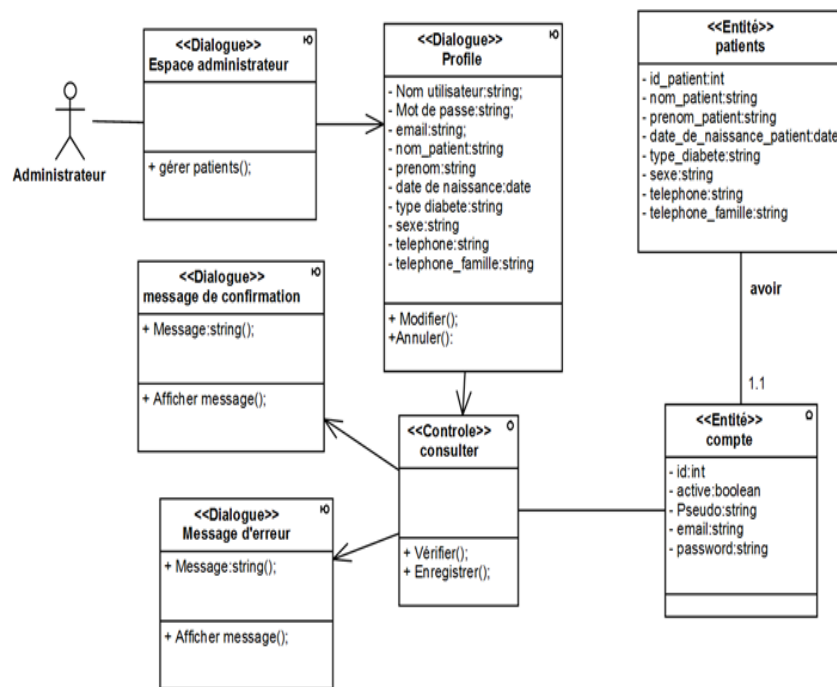


Figure 3.54 – Cas d'utilisation « modifier profils(patients) »

• Cas d'utilisation « modifier profils(assistants)»

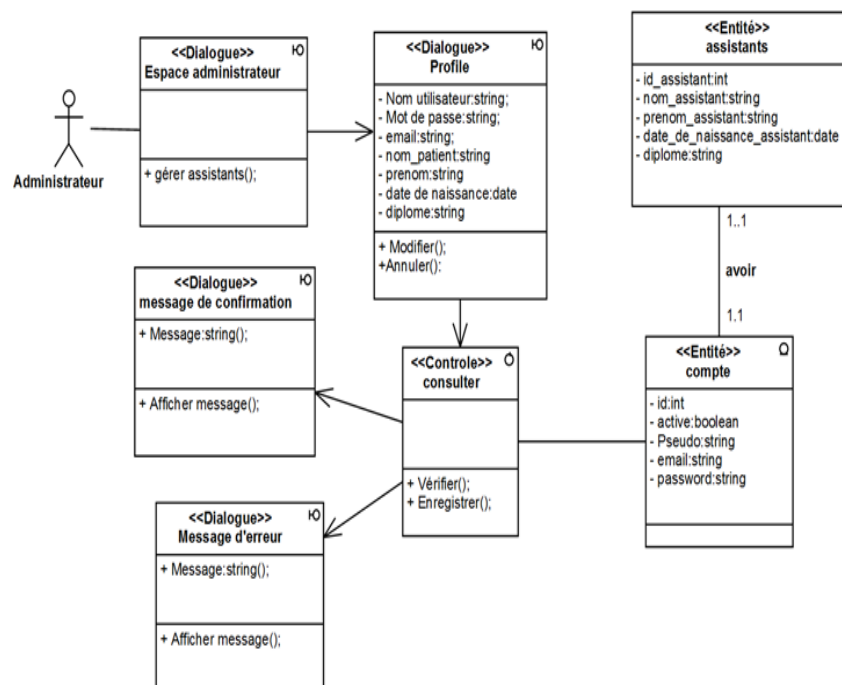


Figure 3.55 – Cas d'utilisation « modifier profils(assistants) »

19. Cas d'utilisation « consulter son dossier médicale »

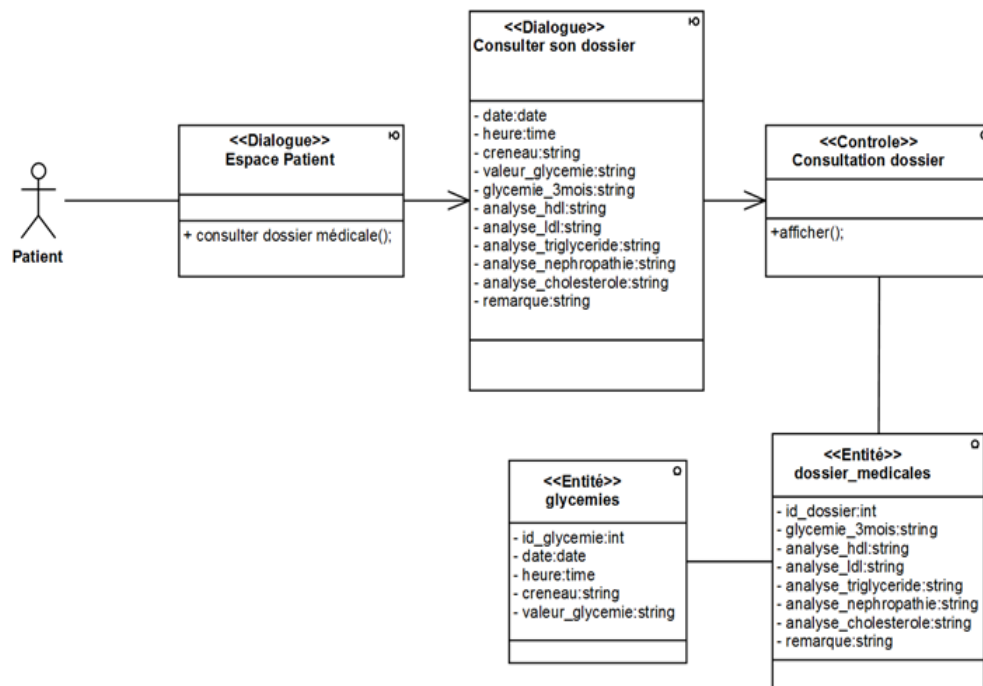


Figure 3.56 – Cas d'utilisation « consulter son dossier médicale »

3.2 Capture des besoins techniques

3.2.1 Définition

La capture des besoins techniques est la première étape de la branche droite du cycle en Y, c'est une étape de prise en compte des contraintes techniques et logicielles. Elle se fait selon deux points de vue :[14]

1. **Matériel** :sert à choisir une configuration matérielle adéquate selon une architecture matérielle.[14]
2. **Logiciel** :sert à exprimer les différentes parties du système par des composants d'exploitation et leurs organisations par des modèles de spécification logicielle.[14]

Pour la réalisation de cette étape nous allons suivre le processus suivant :[14]

- Spécification techniques du point de vue matériel.
- Spécification d'architecture.
- Élaboration du modèle de spécification logicielle.
- Organisation en couches du modèle de spécification.

3.2.2 Spécification techniques du point de vue matériel

Les prés requis techniques ont été exprimés dans l'étude préliminaire, lors de l'expression des besoins opérationnels et de celle des choix stratégiques de développement. Ces choix impliquent des contraintes relatives à la configuration du réseau matériel. Elles sont de natures géographiques, organisationnelles et techniques.[14]

Le matériel à mettre dans un système dépend de :

- **Style d'architecture en niveaux** :qui spécifie le nombre de niveaux géographiques et organisationnels où vont se situer les environnements d'exécution du système. L'architecture peut être à deux niveaux, à trois niveaux ou multi niveaux dans une organisation plus complexe.[14]
- **Contraintes techniques** :peuvent se manifester de différentes manières :[14]
 - La sécurité.
 - La disponibilité.
 - La performance.
- **La configuration matérielle** :

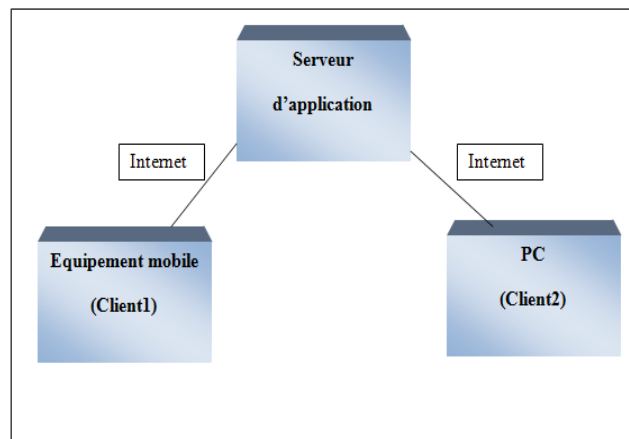


Figure 3.57 – Configuration matérielle de système

3.2.3 Spécification d'architecture

L'expression des prés requis techniques implique également le choix d'un style d'architecture client/serveur, ce choix conditionne la façon dont seront organisés et déployés les composants d'exploitation du système.[14]

- **Le style d'architecture en tiers** : tiers signifie « partie », spécifier l'organisation des composants d'exploitation mis en œuvre pour réaliser le système. Chaque partie indique une responsabilité technique à laquelle souscrivent les différents composants d'exploitation d'un système. [14]
- **Le composant d'exploitation** : un composant d'exploitation est une partie du système logiciel qui doit être connue, installée, déclarée et manipulée par les exploitants du système. Il assure des fonctions bien identifiées dans le système. [14]

L'application réalisée repose sur une architecture client/serveur de 2 tiers. L'architecture 2 tiers est composée de deux niveaux :

- **Le client (navigateur web) et le client (application mobile)** : le demandeur de ressources.
- **Le serveur d'applications et le serveur de base de données** : fournit directement la ressource demandé par le client. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service

Dans notre travail, notre choix porte sur « Mozilla Firefox, Google chrome » comme navigateur Web, un serveur d'applications « Apache » et le serveur de base de données « MySQL ».

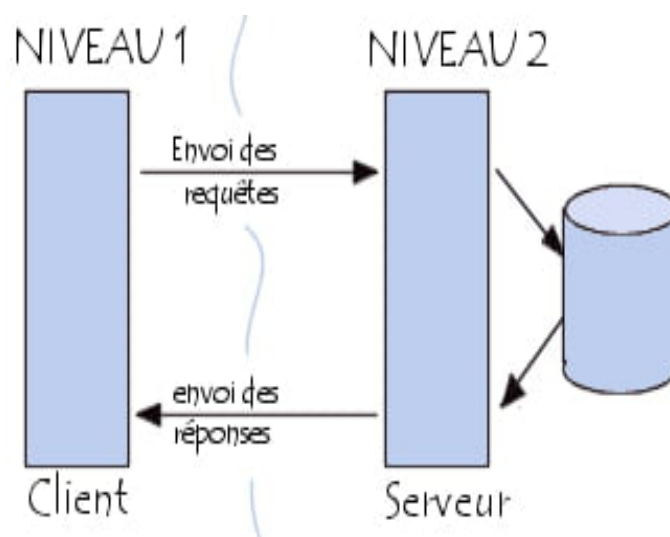


Figure 3.58 – Architecture 2 tiers de notre système[20]

3.2.3.1 Spécification d'organisation du modèle déploiement

La spécification d'architecture à composants métier 2tiers mène à répartir les composants d'exploitation selon les responsabilités suivantes :[13]

- **Le stockage des données** : correspond à un système de gestion de base de données relationnelle (SGBDR) qui est MySQL.
- **Gestion d'application** : correspond aux différents composants d'application et de service du serveur web, les applications sont développées en android studio pour l'application mobile et laravel pour l'application web.

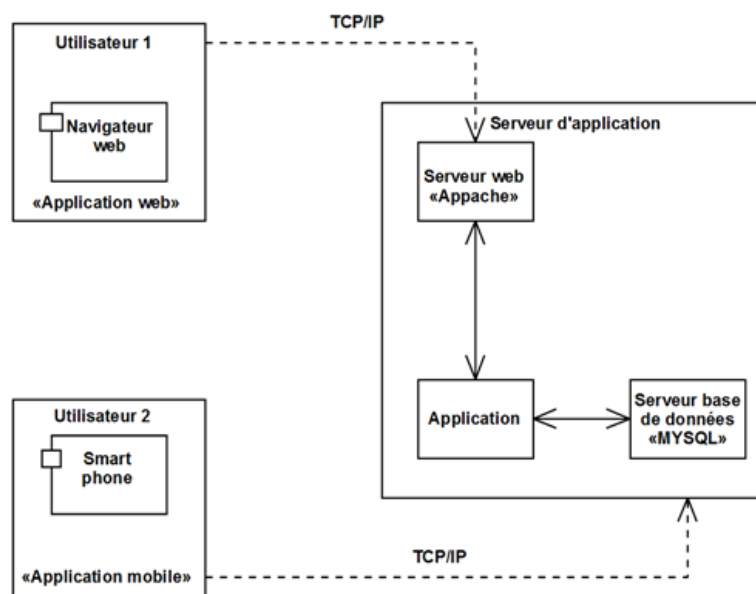


Figure 3.59 – Le modèle de déploiement du système

3.2.4 Elaboration du modèle de spécification logicielle

C'est à cette étape que l'on va se référer aux cas d'utilisation techniques. Un cas d'utilisation technique est un cas qui ne produit aucune valeur fonctionnelle (métier) mais fournit des services techniques à l'exploitant, c.à.d. la personne qui utilise le cas. Des exemples de services techniques sont la connexion au système et l'utilisation de l'aide.[14]

- **L'exploitant** : il s'agit d'un acteur au sens d'UML, qui bénéficie des services techniques du système. L'utilisateur classique d'une application est dans ce sens un exploitant car il bénéficie au moins du service de connexion à l'application.[14]

- **Le cas d'utilisation technique** :un cas d'utilisation technique est destiné à l'exploitant, c'est une séquence d'action produisant une valeur ajoutée opérationnelle ou purement techniques. [14]

Les exploitants du système sont :

- L'utilisateur :qui utilise une des applications du système. La majorité des acteurs de la branche fonctionnelle sont donc des utilisateurs dans le domaine technique (assistant, médecin, patient).
- L'ingénieur d'exploitation :qui est chargé de déployer et de dépanner le système (administrateur).

Les cas technique du système sont :

1. Gérer la sécurité.
2. Gérer l'intégrité.
3. Gérer les erreurs.

3.2.4.1 Modèle de spécification logicielle

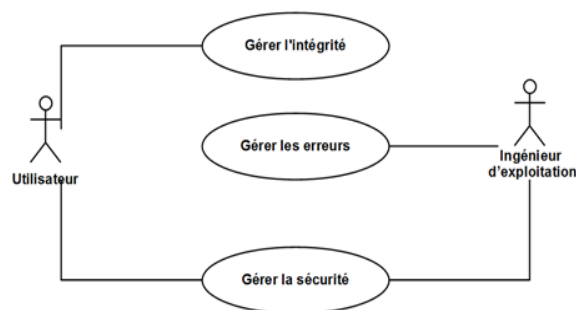


Figure 3.60 – Modèle de spécification logicielle initial

3.2.4.2 Description textuelle des cas d'utilisation techniques

Cas d'utilisation 1 : Gérer l'intégrité	
Intention	C'est le mécanisme qui empêche la mise à jour simultanée d'une même entité par plusieurs utilisateurs.
Action	Empêche la mise à jour simultanée.

Cas d'utilisation 2 : Gérer les erreurs	
Intention	Gérer les erreurs de frappe et de manipulation.
Action	Spécifie l'erreur, la traiter, et le déboguer.

Cas d'utilisation 3 : Gérer la sécurité	
Intention	L'utilisateur doit se connecter, s'authentifier et être reconnue par le système à travers le mot de passe.
Action	Permission de faire des opérations.

Figure 3.61 – Description textuelle des cas d'utilisation techniques

3.2.5 Organisation en couche du modèle de spécification

Le modèle de spécification logicielle est cependant trop sommaire pour permettre une spécification technique détaillée, Dans ce contexte, il est difficile de pouvoir préciser de manière détaillée les comportements techniques attendus, si l'on n'organise pas la spécification suivants les différentes responsabilités de traitement. [14]

- **Couche logicielle** : une couche logicielle représente un ensemble de spécification ou de réalisation qui respectivement expriment ou mettent en œuvre un ensemble de responsabilités techniques et homogènes pour un système logiciel. Les couches s'empilent en niveaux pour couvrir des transformations logicielles successives, de sorte que la couche d'un niveau ne puisse utiliser que les services des couches des niveaux inférieurs.[14]
 - * **Présentation** : restitue les données à l'utilisateur et transforme ses actions en événement de l'application.
 - * **Application** : représente les objets de contrôle et pilote les règles de l'application.
 - * **Métier** : représente les objets du métier et implémente leurs règles de gestion.

- * **Accès aux données** :restitue les représentations métier à partir du moyen de stockage.
- * **Stockage des données** :assure la persistance des données. [14]

Dans le modèle UML, les couches logicielles correspondent à des packages.

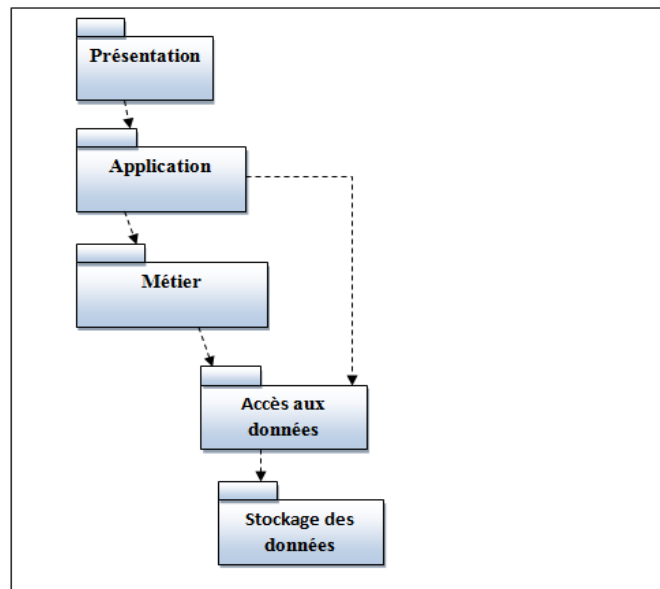


Figure 3.62 – Organisation du modèle de spécification logicielle(application mobile)

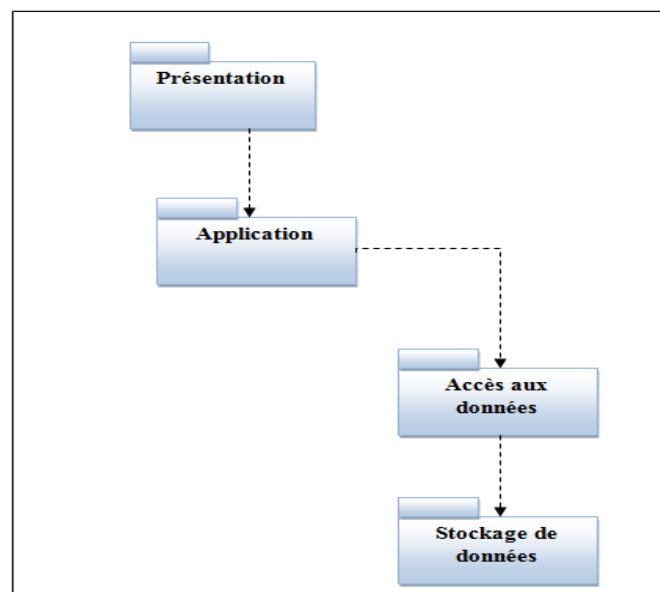


Figure 3.63 – Organisation du modèle de spécification logicielle(application web)

Conclusion

La capture des besoins fonctionnels joue un rôle essentiel dans le cadre d'une part, de compléter les recueils initiaux des besoins opérés pendant l'étude préliminaire, et de l'autre part, elle donne une première vue pour le prochain chapitre concernant l'analyse afin d'identifier les classes candidates du modèle statique qui présentent une des approches orientées objets. La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description de métier des utilisateurs, ni de la description applicative.

Chapitre 4

L'analyse

Introduction

L'étape d'analyse est entamée à la suite de l'étape de capture des besoins fonctionnels, Elle est située sur la branche gauche du processus 2TUP.

Dans ce chapitre là on va traiter la phase de l'analyse objet du système, en suivant les étapes suivantes :

- Le découpage en catégories.
- Le développement du modèle statique.
- Le développement du modèle dynamique.

4.1 Découpage en catégories

Le découpage en catégories constitue la première activité de l'étape d'analyse, Cette phase utilise la notion de package pour définir des catégories de classe d'analyse et découper le modèle UML en blocs logiques les plus indépendants possibles.

Une catégorie consiste en un regroupement logique de classe à forte cohérence interne et faible couplage externe.[14]

• Démarche d'élaboration du modèle structurel d'analyse :

La démarche mise en œuvre dans ce chapitre suit les étapes suivantes :

- Répartir les classes candidates en catégories.
- Elaborer les diagrammes de classe préliminaires par catégorie.
- Décider des dépendances entre catégories.

4.1.1 Répartition des classes candidates en catégories

Le découpage en catégories de notre projet a donné le résultat suivant :

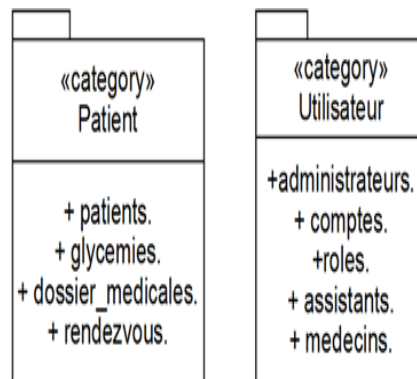


Figure 4.1 – Découpage en catégories

4.1.2 Elaboration des diagrammes de classe préliminaire pour chaque catégorie

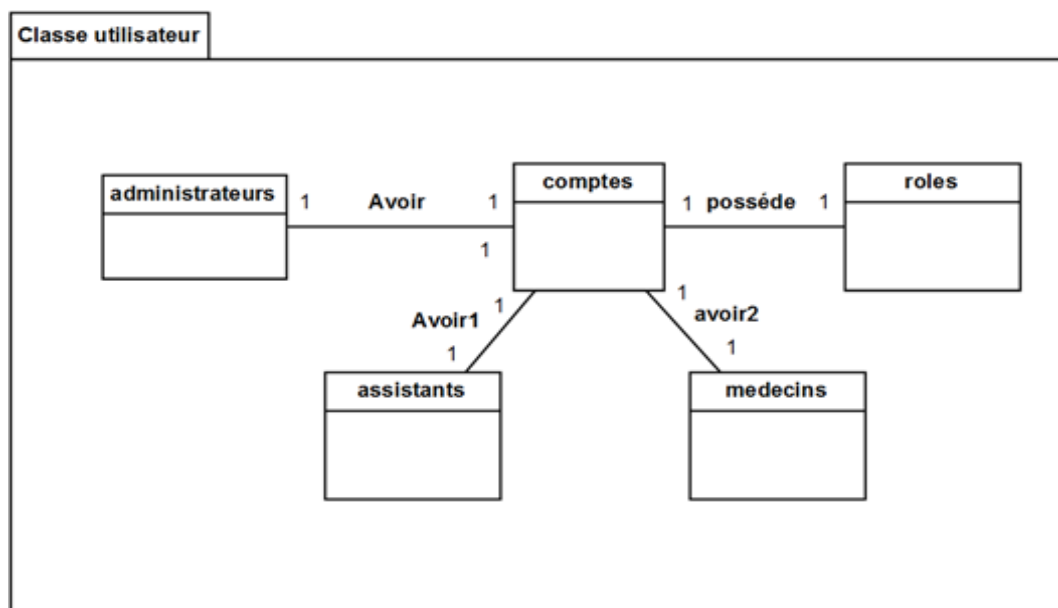


Figure 4.2 – Les diagrammes de classe préliminaire du catégorie(utilisateur)

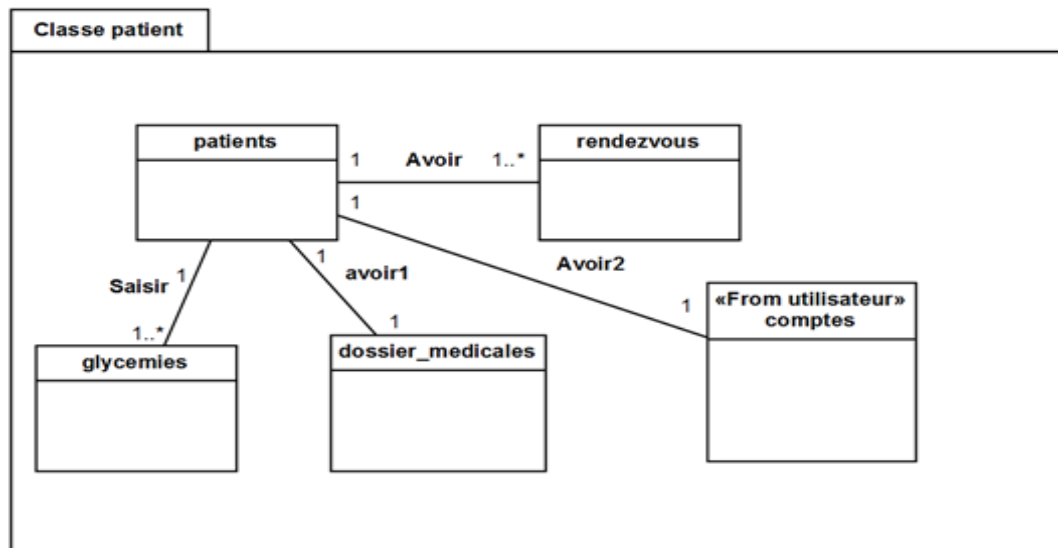


Figure 4.3 – Les diagrammes de classe préliminaire du catégorie(patient)

4.1.3 Dépendance entre les catégories

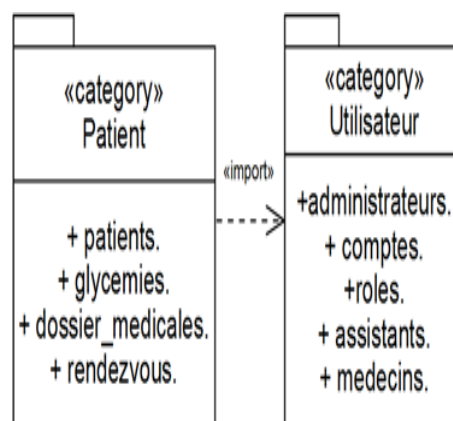


Figure 4.4 – Modèle structurel d'analyse

4.2 Développement du modèle statique

Le développement du modèle statique constitue la deuxième activité de l'étape d'analyse. Les diagrammes de classes établis sommairement dans les diagrammes de classes participantes, puis réorganisés lors du découpage en catégories, vont être détaillés, complétés et optimisés.[14]

4.2.1 Affiner les classes

Les classes identifiées lors de l'étude des cas d'utilisations sont simplement des classes candidates pour l'analyse objet, il convient désormais de les examiner de manière détaillé, d'en éliminer certaines, ou au contraire d'en ajouter d'autre.[14]

4.2.2 Affiner les associations

Les associations représentent des relations conceptuelles entre les classes. On peut également dire qu'elles impliquent des responsabilités en termes de navigation.

la navigation dans un modèle statique représente la capacité à obtenir des informations en parcourant les associations entre les classes. On peut donc considérer les associations comme porteuses d'une partie fondamentale de la structure statique des classes.[14]

Affiner les associations convient de les valider, les préciser, en éliminer et en ajouter. Il s'agit d'une activité, consiste également à utiliser deux notions complémentaires fournies par UML, l'agrégation et la composition, qui sera complétée grâce à l'identification des attributs.[14]

4.2.3 Ajouter les attributs

Un attribut est une propriété nommée d'une classe qui décrit un domaine de valeur possible partagé par tous les objets de la classe. Dans un modèle d'analyse, nous conservons uniquement comme attribut, les propriétés simples des classes que le système doit mémoriser et utiliser.[14]

4.2.4 Ajouter les opérations

Une opération représente un service, un traitement qui peut être demandé à n'importe quel objet de la classe. Une opération est l'abstraction de ce qu'on peut réaliser sur un objet, et elle est partagée par tous les objets de la classe.[14]

4.2.5 Diagramme de classe détaillé pour chaque catégorie

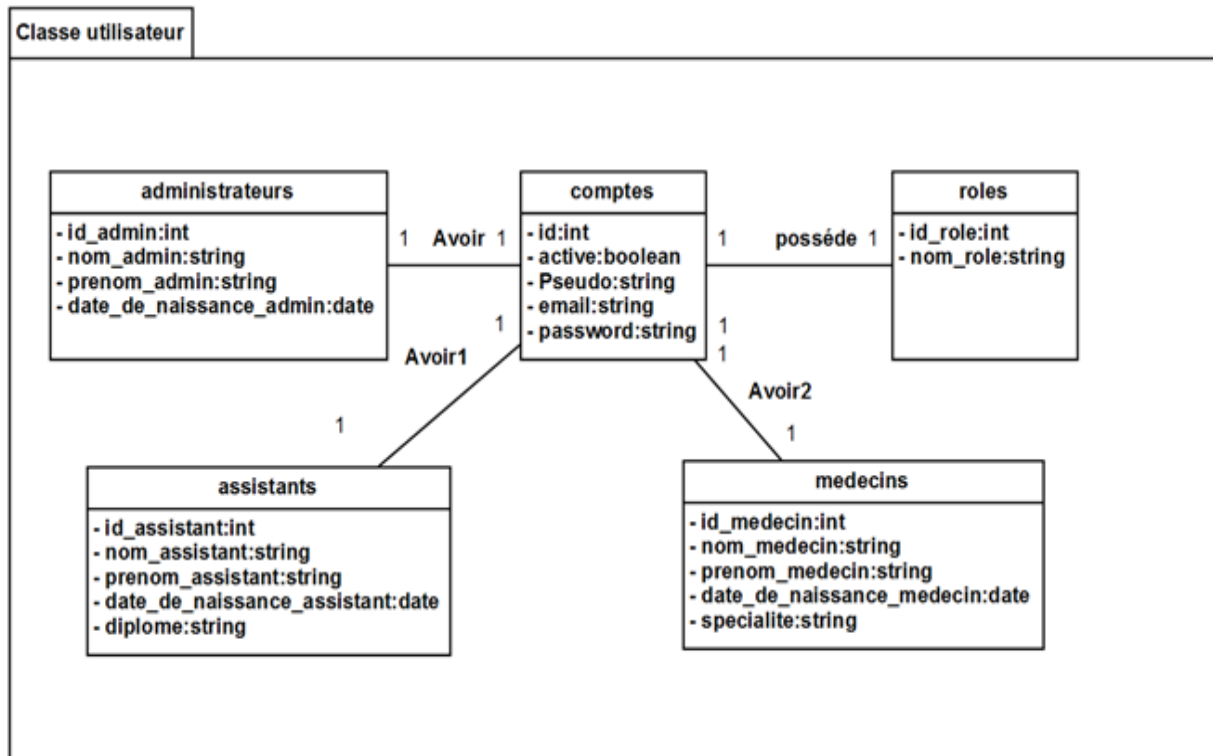


Figure 4.5 – Le diagrammes de classe détaillé de catégorie utilisateur

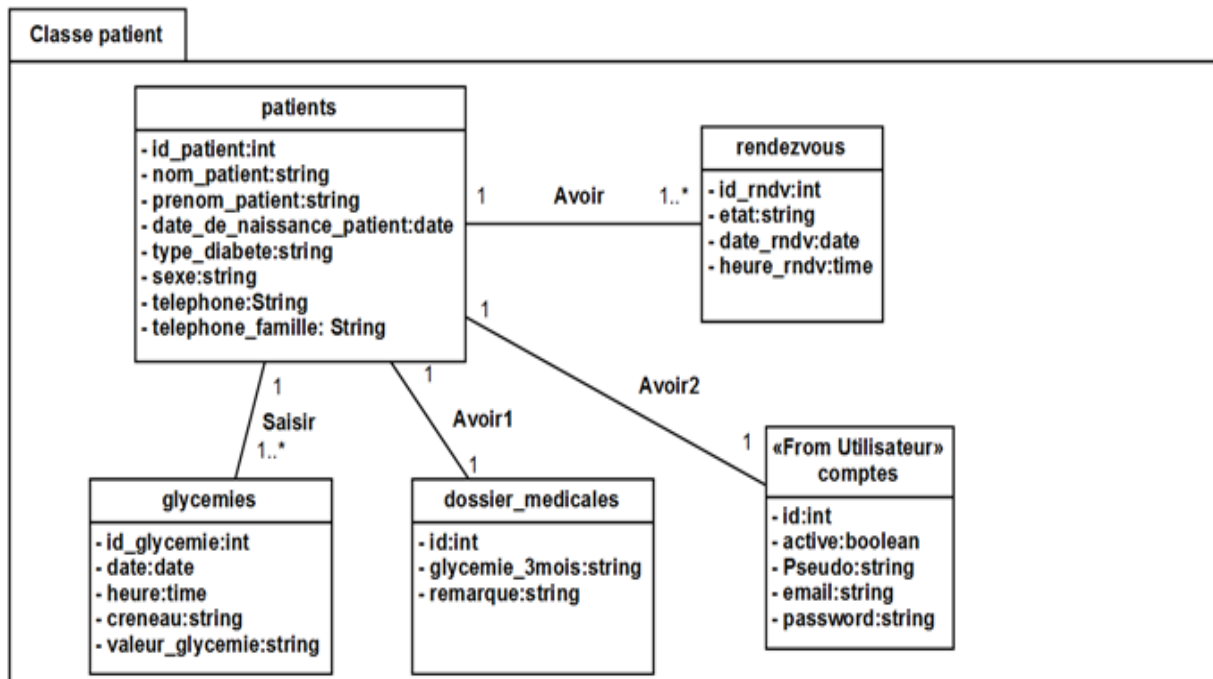


Figure 4.6 – Le diagramme de classe détaillé de catégorie patient

4.3 Développement du modèle dynamique

Le développement du modèle dynamique constitue la troisième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y. Il s'agit d'une activité itérative, fortement couplée avec l'activité de la modélisation statique.[14]

4.3.1 Diagramme de séquence détaillé

Les diagrammes de séquences permettent de représenter des collaborations entre objet selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. un diagramme de séquence qui est plus apte à représenter un scénario dans le contexte d'un cas d'utilisation. Pour élaborer le modèle dynamique de notre système, on va utiliser le diagramme séquence.[13]

1. Cas d'utilisation « s'authentifier »

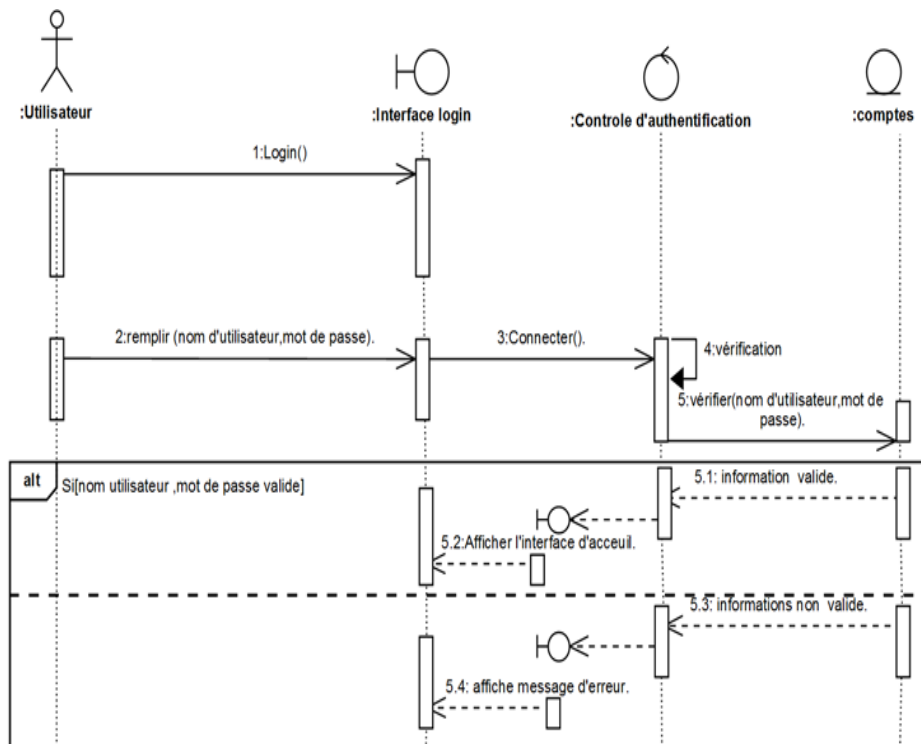


Figure 4.7 – Cas d'utilisation « s'authentifier »

2. Cas d'utilisation « ajouter valeur glycémie »

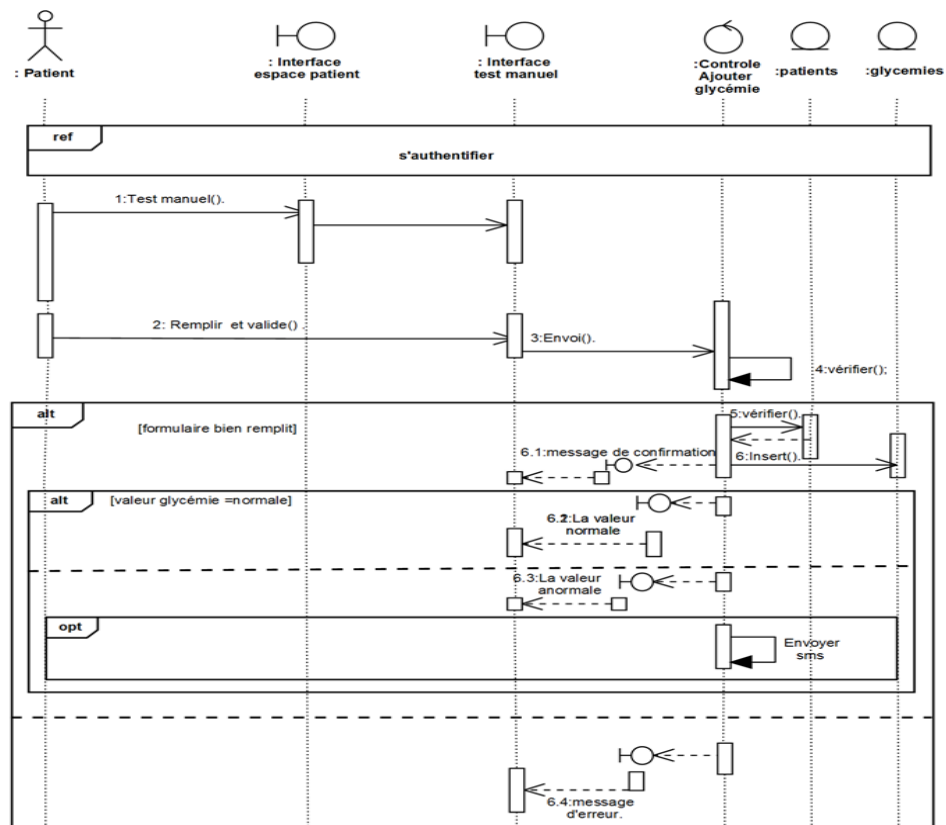


Figure 4.8 – Cas d'utilisation « ajouter valeur glycémie »

3. Cas d'utilisation « créer profile médecin »

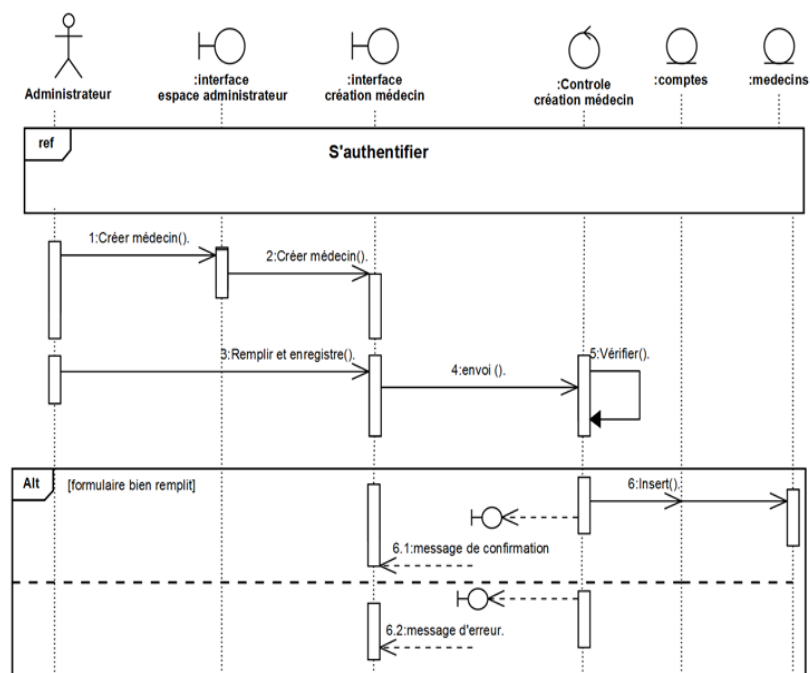


Figure 4.9 – Cas d'utilisation « créer profile médecin »

4. Cas d'utilisation « annuler rendez-vous »

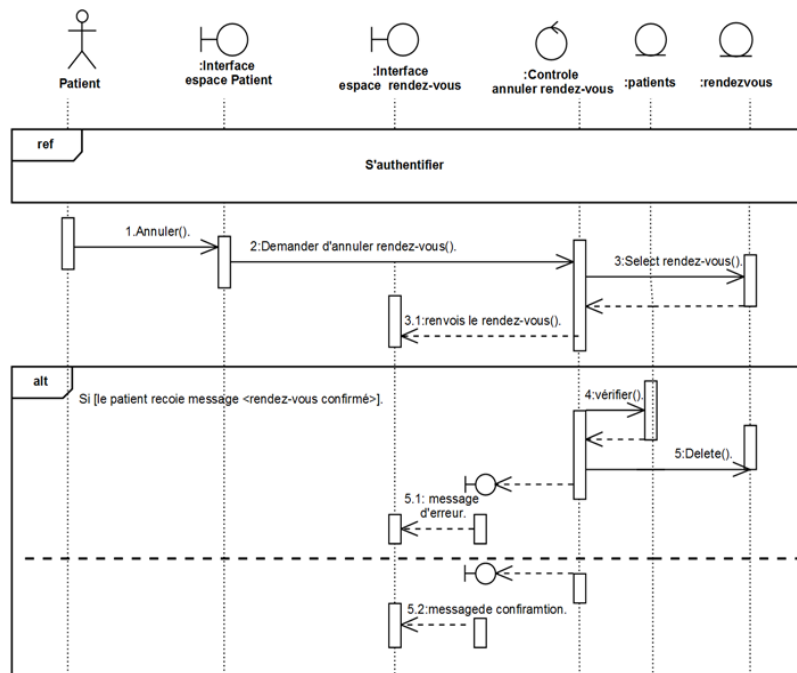


Figure 4.10 – Cas d'utilisation « annuler rendez-vous »

5. Cas d'utilisation « consulter son profile »

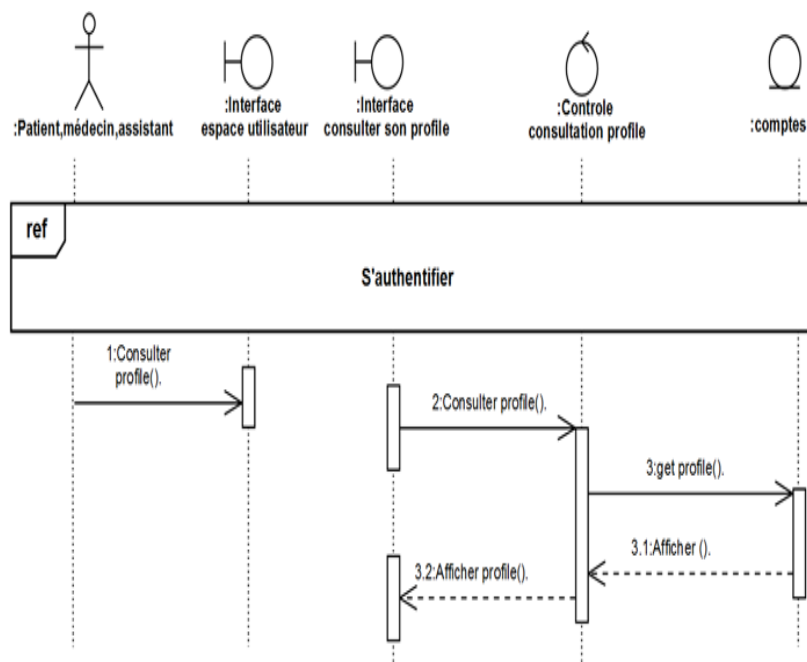


Figure 4.11 – Cas d'utilisation « consulter son profile »

6. Cas d'utilisation « prendre rendez-vous »

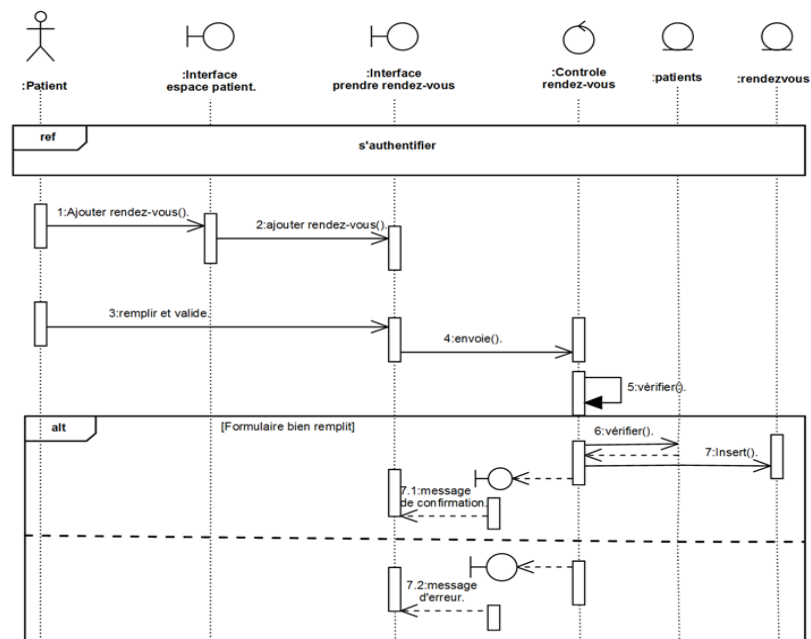


Figure 4.12 – Cas d'utilisation « prendre rendez-vous »

7. Cas d'utilisation « consulter profiles utilisateurs »

• Cas d'utilisation « consulter profiles (médecins) »

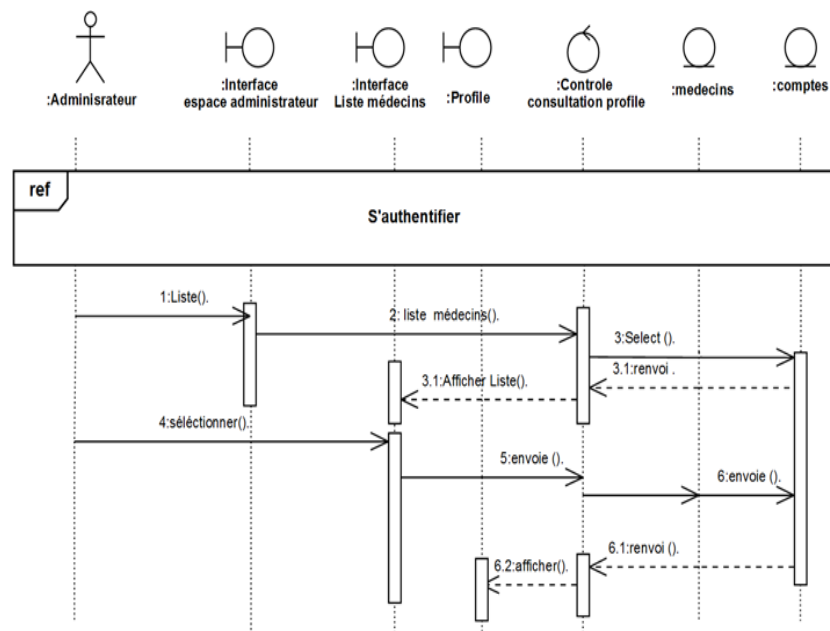


Figure 4.13 – Cas d'utilisation « consulter profiles (médecins) »

• Cas d'utilisation « consulter profiles (patients) »

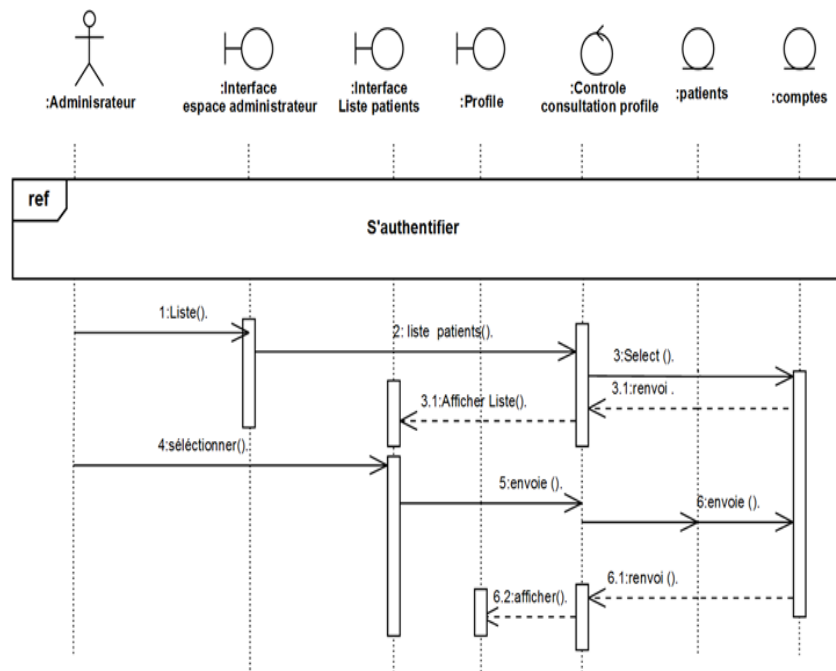


Figure 4.14 – Cas d'utilisation « consulter profiles (patients) »

• Cas d'utilisation « consulter profiles (assistants) »

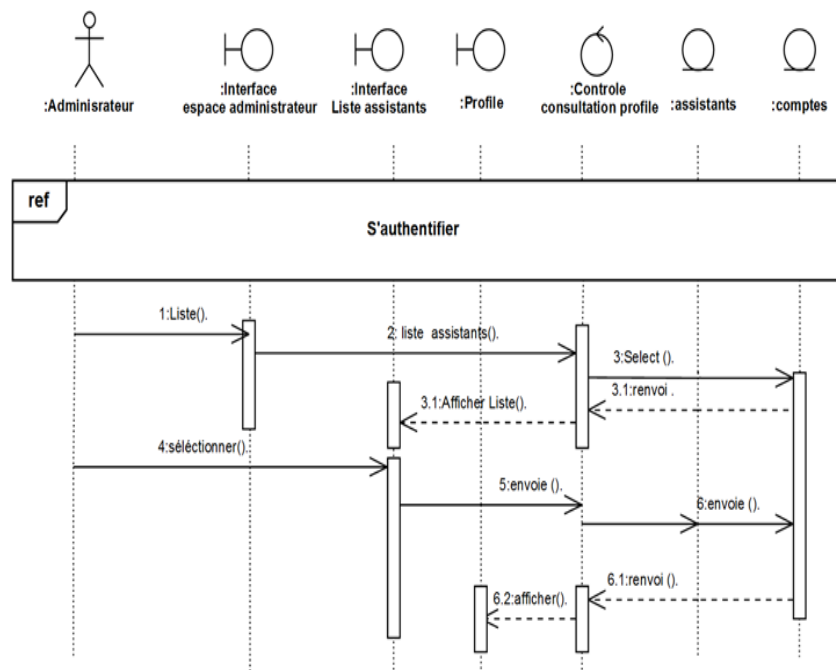


Figure 4.15 – Cas d'utilisation « consulter profiles(assistants) »

8. Cas d'utilisation « modifier les dossiers médicaux »

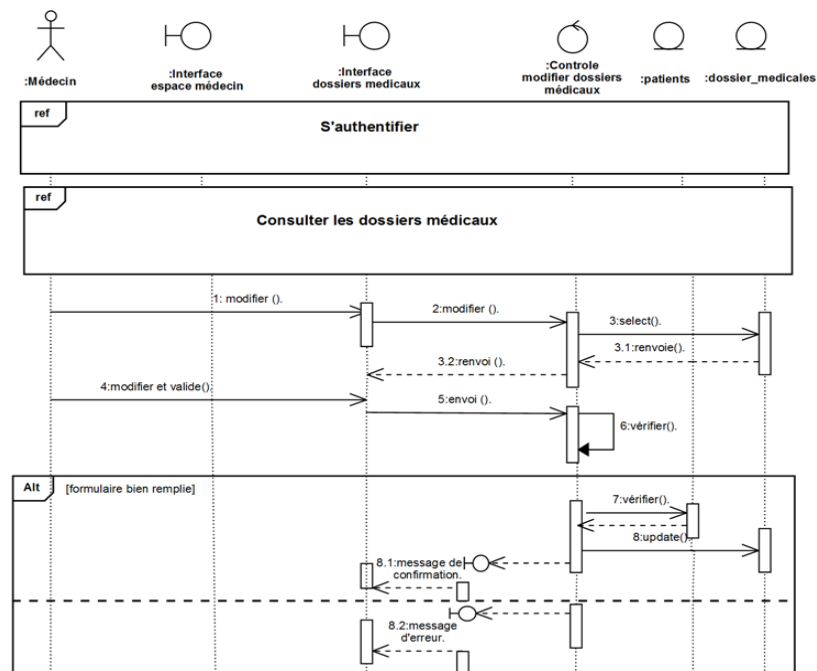


Figure 4.16 – Cas d'utilisation « modifier les dossiers médicaux »

9. Cas d'utilisation « confirmer rendez-vous »

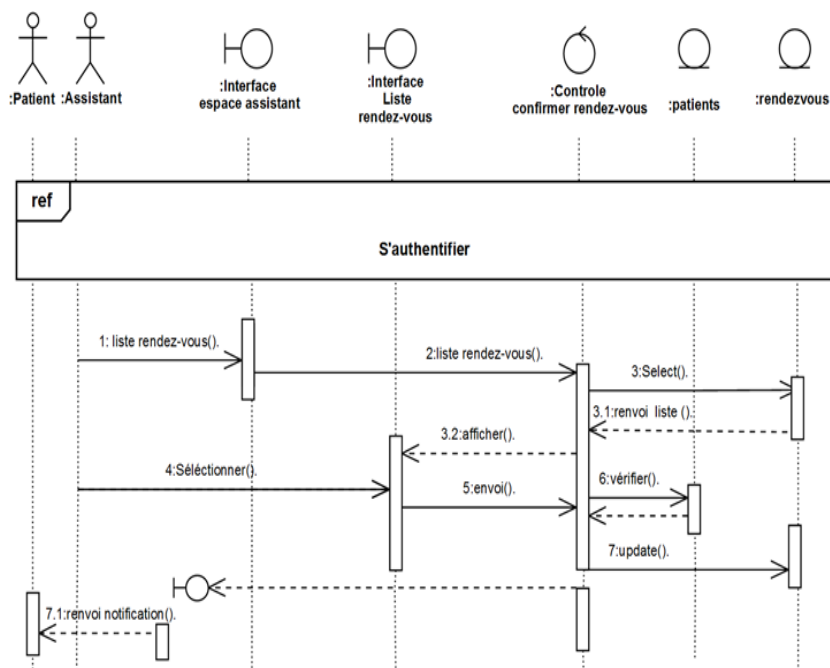


Figure 4.17 – Cas d'utilisation « confirmer rendez-vous »

10. Cas d'utilisation « créer profile assistant »

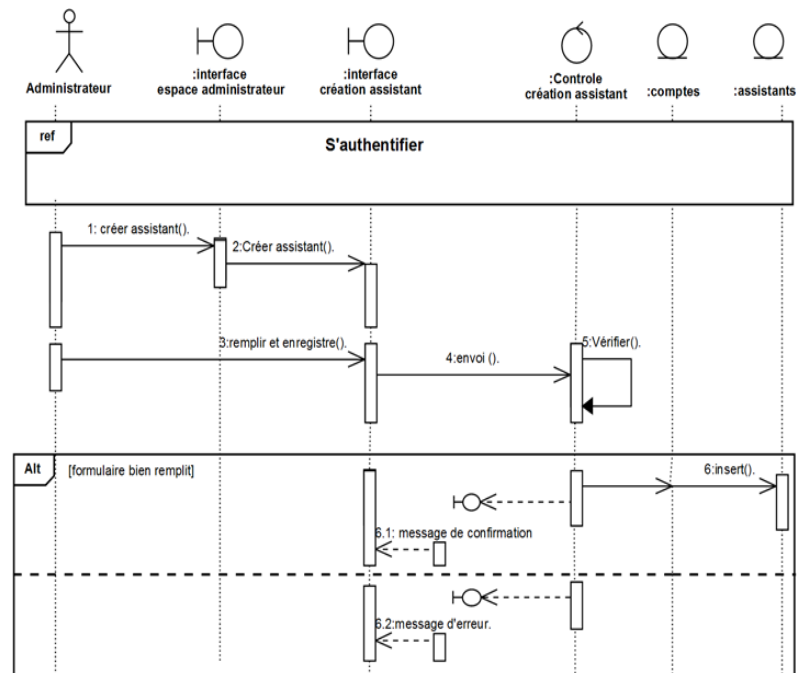


Figure 4.18 – Cas d'utilisation « créer profile assistant »

11. Cas d'utilisation « saisir analyses 3 mois »

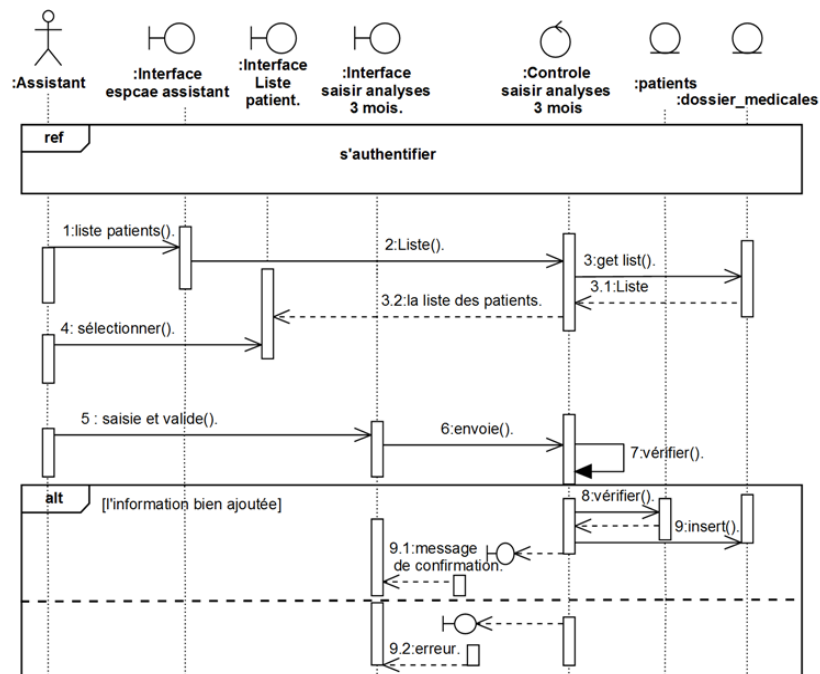


Figure 4.19 – Cas d'utilisation « saisir analyses 3 mois »

12. Cas d'utilisation « créer profile patient »

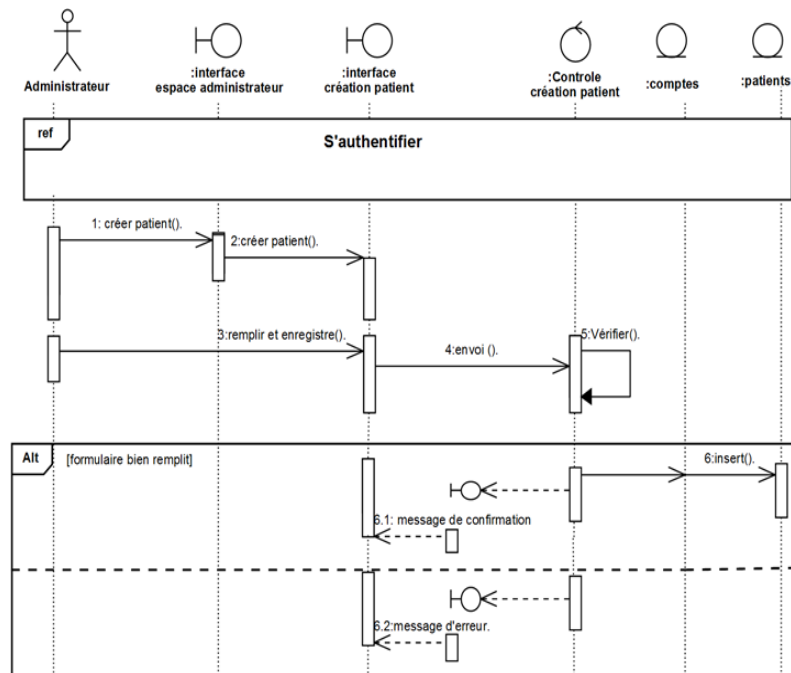


Figure 4.20 – Cas d'utilisation « créer profile patient »

13. Cas d'utilisation « consulter planning des rendez-vous »

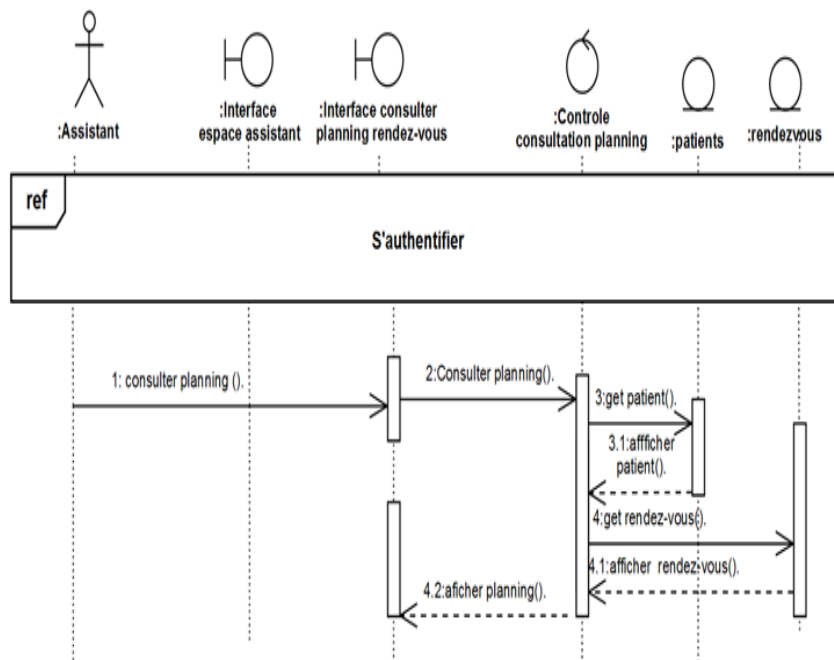


Figure 4.21 – Cas d'utilisation « consulter planning des rendez-vous »

14. Cas d'utilisation « modifier valeur glycémie »

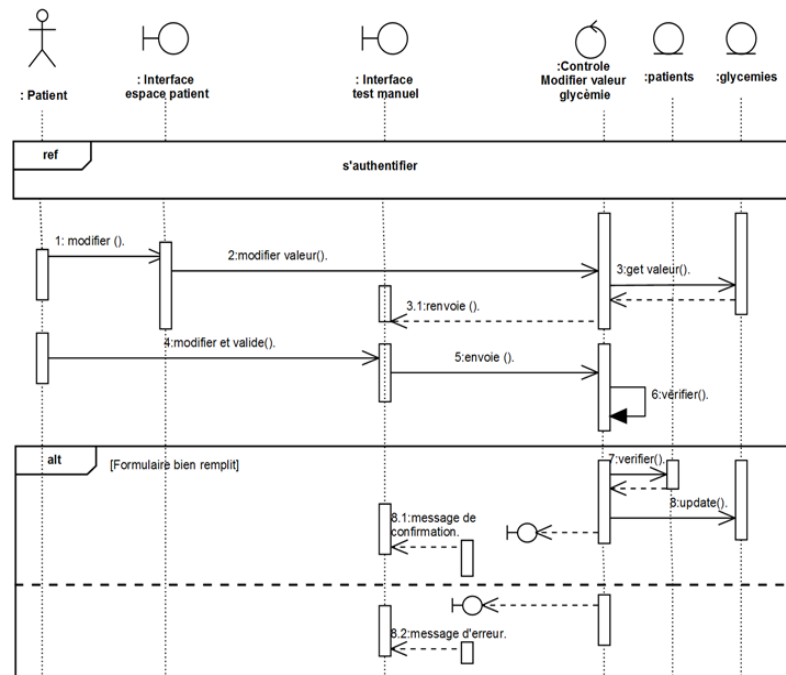


Figure 4.22 – Cas d'utilisation « modifier valeur glycémie »

15. Cas d'utilisation « consulter les dossiers médicaux »

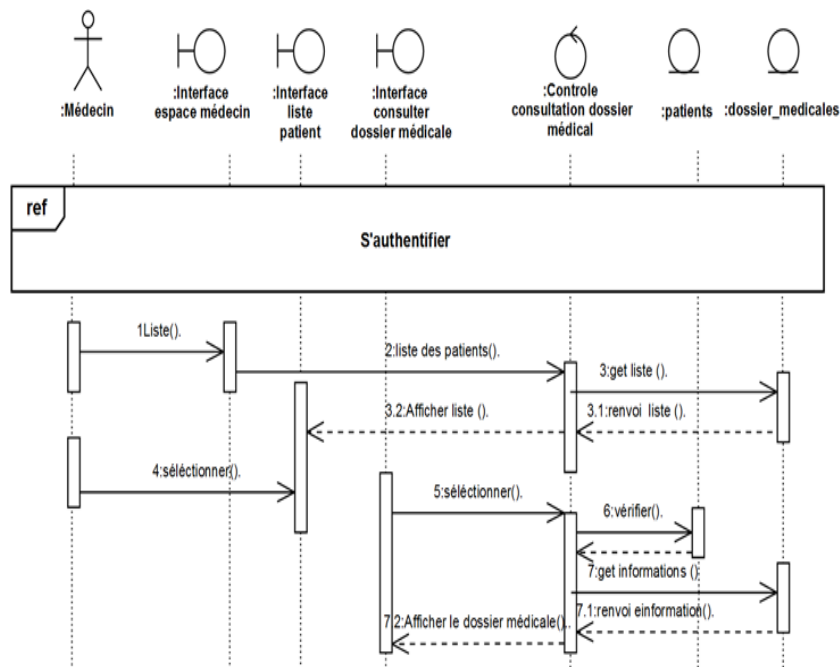


Figure 4.23 – Cas d'utilisation « consulter les dossiers médicaux »

16. Cas d'utilisation « bloquer profile »

• Cas d'utilisation « bloquer profiles(médecins) »

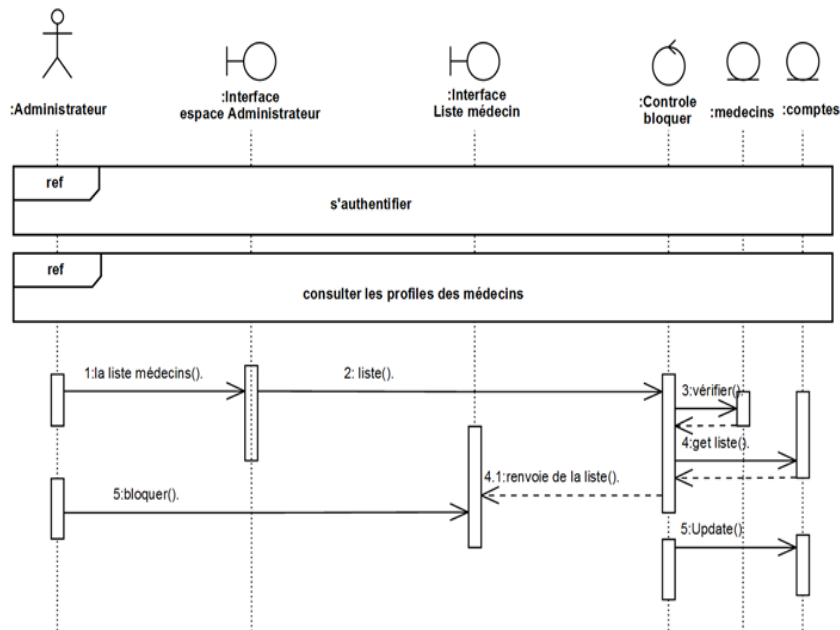


Figure 4.24 – Cas d'utilisation « bloquer profiles (médecins)»

• Cas d'utilisation « bloquer profiles(patients) »

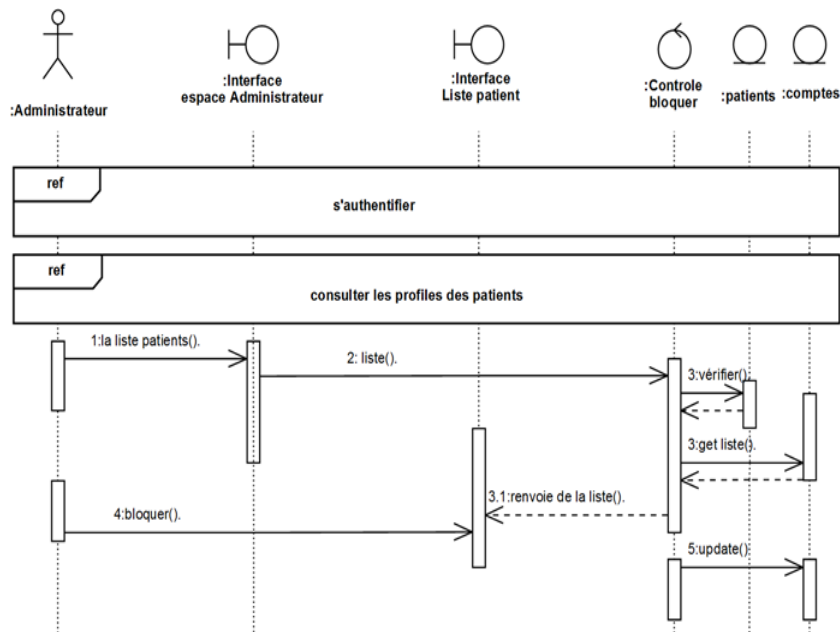


Figure 4.25 – Cas d'utilisation « bloquer profiles (patients) »

• Cas d'utilisation « bloquer profiles (assistants) »

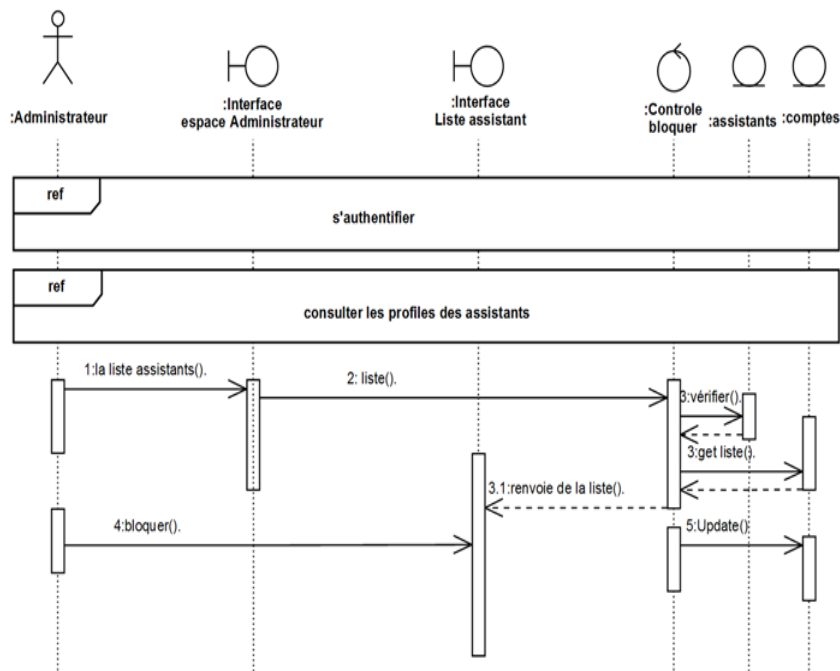


Figure 4.26 – Cas d'utilisation « bloquer profile (assistants) »

17. Cas d'utilisation « modifier son profile »

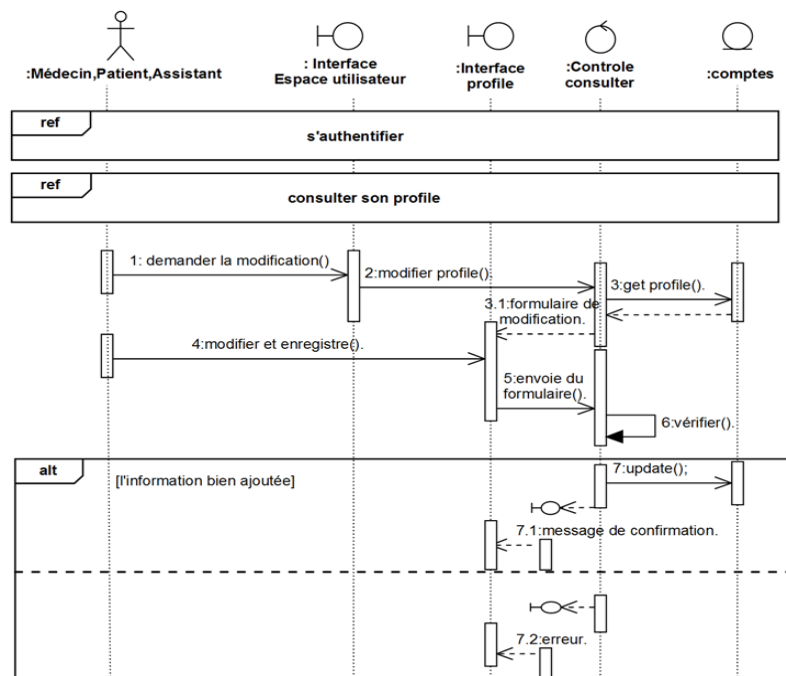


Figure 4.27 – Cas d'utilisation « modifier son profile »

18. Cas d'utilisation « modifier profiles »

• Cas d'utilisation « modifier profiles (médecins) »

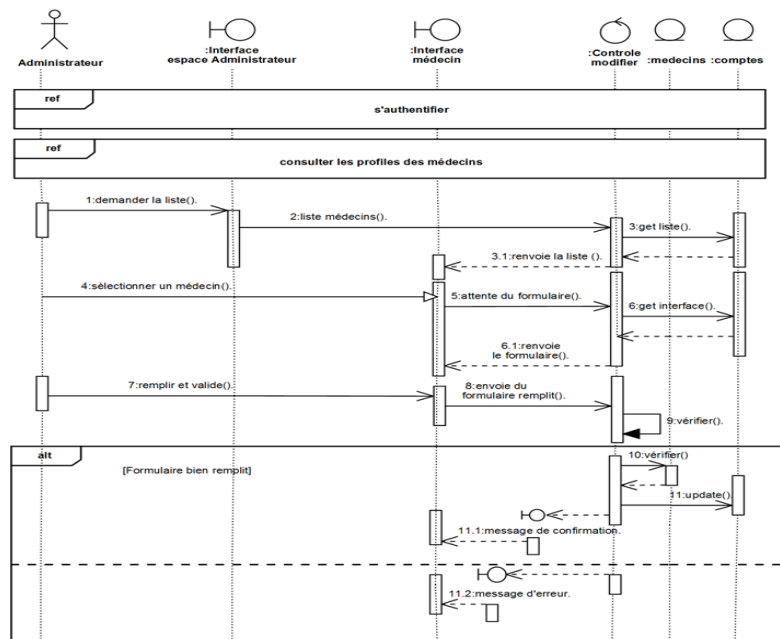


Figure 4.28 – Cas d'utilisation « modifier profiles (médecins) »

• Cas d'utilisation « modifier profiles (patients) »

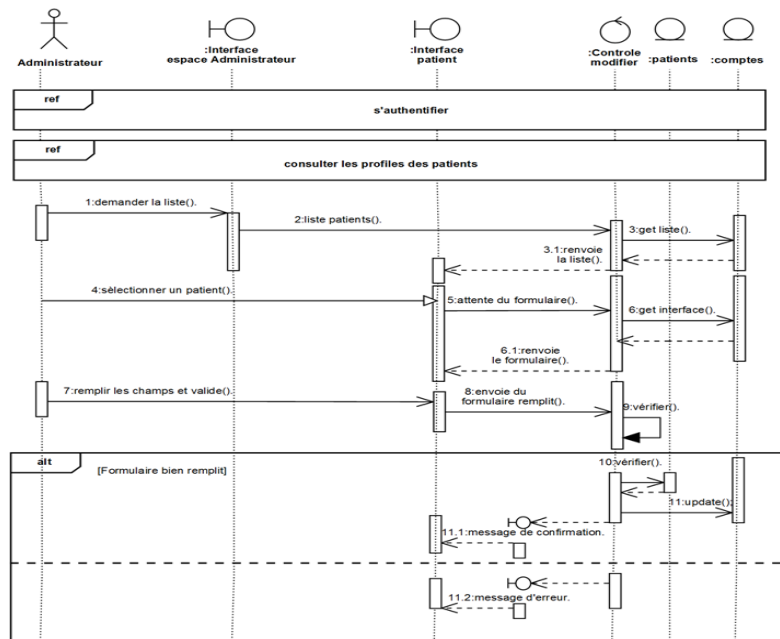


Figure 4.29 – Cas d'utilisation « modifier profiles (patients) »

• Cas d'utilisation « modifier profiles (assistants) »

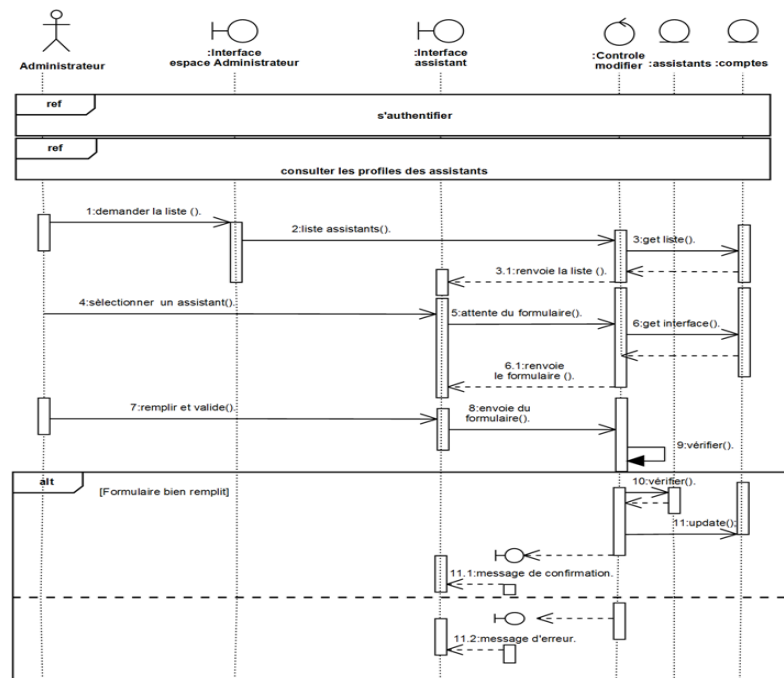


Figure 4.30 – Cas d'utilisation « modifier profiles (assistants) »

19. Cas d'utilisation « consulter son dossier médicale »

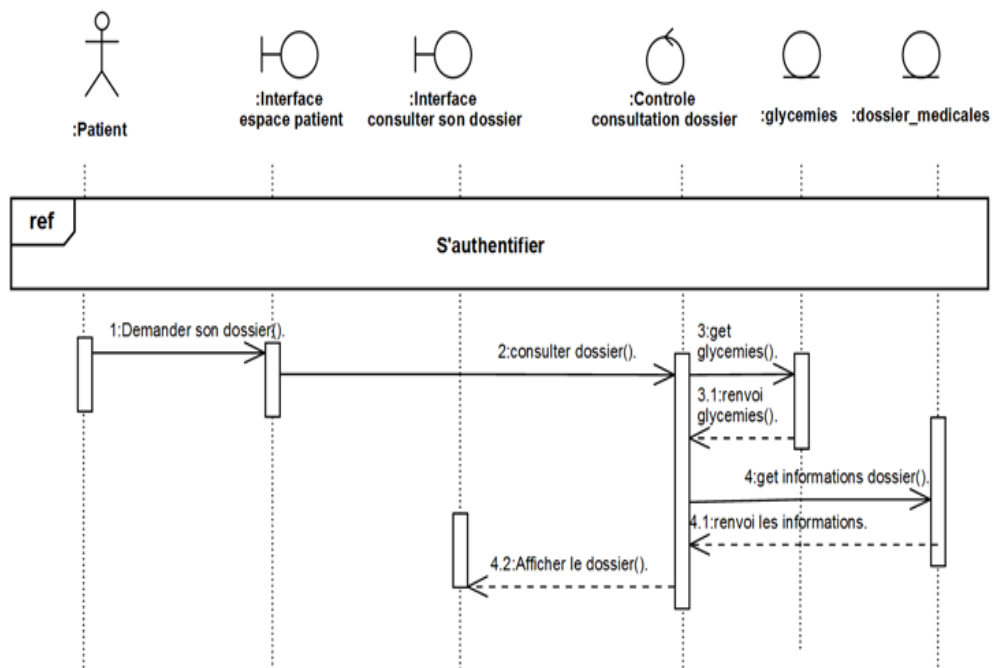


Figure 4.31 – Cas d'utilisation « consulter son dossier médicale »

Conclusion

Dans ce chapitre, nous avons présenté l'analyse objet du système, les classes issues des besoins fonctionnels sont regroupées en catégories pour organiser le modèle structurel d'analyse. Ce modèle nécessite un travail d'analyse détaillée de la structure des classes. Celui-ci est considéré comme une base pour le développement du modèle statique et dynamique. Nous allons décrire dans le chapitre suivant la conception de notre système.

Chapitre 5

Conception

Introduction

La phase de conception se déroule en trois étapes qui sont la conception générique qui est une étape de la branche droite, du processus en Y, et les deux étapes du milieu du processus qui sont la conception préliminaire, elle s'effectue la fusion des études fonctionnelles et techniques, et la conception détaillé qui vient juste après est une activité qui s'inscrit dans l'organisation définie par la conception préliminaire.[14]

5.1 Conception préliminaire

5.1.1 Définition

La conception préliminaire est certainement l'étape la plus délicate du processus 2TUP car elle représente le cœur.à cette étape on va effectue la fusion des études fonctionnelles et techniques, La conception préliminaire est avant tout affaire d'organisation ; elle s'appuie sur les points de vue de spécification fonctionnelle et structurelle de l'analyse, mais également sur les Frameworks de la conception technique.[14]

5.1.2 Développement du modèle de déploiement

5.1.2.1 Définition du poste de travail

Le poste de travail représente un ou plusieurs acteurs pouvant être localisés sur une machine d'un type particulier, et remplissant une fonction identifiée dans l'entre-

prise, le poste de travail ne représente pas forcément une machine physique, mais peut consister en plusieurs machines, à condition qu'elles donnent lieu au même type de déploiement.[14]

5.1.2.2 Le modèle de déploiement

Les modèles de déploiement et de configuration matérielle s'expriment tous deux à l'aide d'un digramme de déploiement. Le modèle de déploiement considère plutôt chaque nœud comme un poste de travail, il exprime la répartition physique des fonctions métier du système et permet de justifier la localisation des bases de données et des environnements de travail. Pour notre système on va donner le schéma suivant qui représente son digramme de déploiement.[14]

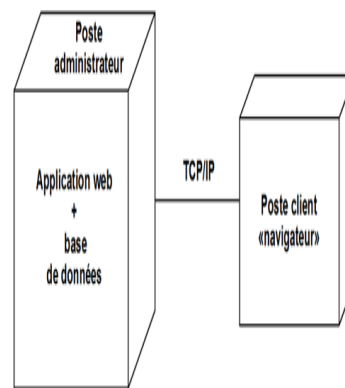


Figure 5.1 – Le modèle de déploiement du système(application web)

5.1.2.3 Architectures adoptée

Le choix de notre solution s'est porté sur une architecture 2 tiers (Client(navigateur, smart phone)/application/base de données), en se basant sur un réseau WAN. Ce choix repose sur les arguments suivants :

- Les ressources sont centralisées sur le serveur. Il est donc plus simple de gérer les ressources communes aux utilisateurs comme la base de données par exemple.
- Cette architecture est plus sécurisée étant donné que le client dispose de moins de point d'entrée pour accéder aux données.

5.2 Conception détaillée

5.2.1 Définition

L'étape de conception détaillée est la dernière étape de la conception, dans cette étape les concepteurs construisent les classes par la conception des attributs, des opérations et faire aussi la conception des associations, ensuite ils construisent les différentes tables de l'application.[14]

5.2.2 Dictionnaire de données

5.2.2.1 Concevoir les attributs et les classes

La conception des attributs consiste principalement à définir le type des attributs identifiés en analyse, pour chaque attribut on donne leur type, qui est en général un type de base. pour concevoir ou documenter une méthode, UML met à notre disposition toute la palette des diagramme du modèle dynamique, nous avons basé sur le diagramme de séquence réalisé dans l'étape d'analyse.[14]

Classe	Description	Code	Type
comptes	Numéro de compte	id	bigint(20)
	Numéro de role	role_id	bigint(20)
	Numéro de patient	patient_id	bigint(20)
	Numéro de assistant	assistant_id	bigint(20)
	Numéro de médecin	medecin_id	bigint(20)
	Numéro d'administrateur	administrateur_id	bigint(20)
	Etat de compte	active	enum('oui','no')
	Nom utilisateur	pseudo	varchar(191)
	Email	email	varchar(191)
	Mot de passe	password	varchar(191)
administrateurs	Numéro d'administrateur	id	bigint(20)
	Nom d'administrateur	nom	varchar(191)
	Prénom d'administrateur	prenom	varchar(191)
	Date de naissance	date _de_naissance	date

assistants	Numéro d'assistant	id	bigint(20)
	Nom d'assistant	nom	varchar(191)
	Prénom d'assistant	prenom	varchar(191)
	Date de naissance	Date _de_naissance	date
	Diplôme	diplome	varchar(191)
patients	Numéro de patient	id	bigint(20)
	Nom de patient	nom	varchar(191)
	Prénom de patient	prenom	varchar(191)
	Date de naissance	Date _de_naissance	date
	Numéro téléphone	telephone	varchar(20)
	Num téléphone famille	telephone_famille	varchar(20)
	Type de diabète	type_diabete	varchar(191)
	Sexe de patient	sexe	enum(male, femelle)
medecins	Numéro de médecin	id	bigint(20)
	Nom de médecin	nom	varchar(191)
	Prénom de médecin	prenom	varchar(191)
	Date de naissance	date _de_naissance	date
	Spécialité	specialite	varchar(191)
rendezvous	Numéro rendez-vous	id	bigint(20)
	Numéro de patient	patient_id	bigint(20)
	Etat rendez-vous	etat	enum(confirme, refuse, en attente)
	Date de rendez-vous	date_rndv	date
	Heure de rendez-vous	heure_rndv	time
glycemies	Numéro de glycémie	id	bigint(20)
	Numéro de patient	patient_id	bigint(20)
	Date de teste	date	date
	Heure de teste	heure	time
	créneau	creneau	varchar(191)
	Valeur de glycémie	valeur	float
dossier_medicales	Numéro de dossier	id	bigint(20)
	Numéro de patient	patient_id	bigint(20)
	analyse HDL	analyse_hdl	string(191)
	analyse LDL	analyse_ldl	string(191)
	analyse triglyceride	analyse_triglyceride	string(191)
	analyse nephropathie	analyse_nephropathie	string(191)
	analyse cholesterole	analyse_cholesterole	string(191)
	Glycémie de 3 mois	glycemie_3mois	float
	Remarque médecin	remarque	varchar(191)
roles	Numéro de role	id	bigint(20)
	Nom de role	nom	varchar(191)

Table 5.1 – Dictionnaire de données avec les classes et les attributs

5.3 Diagramme de classe détaillé

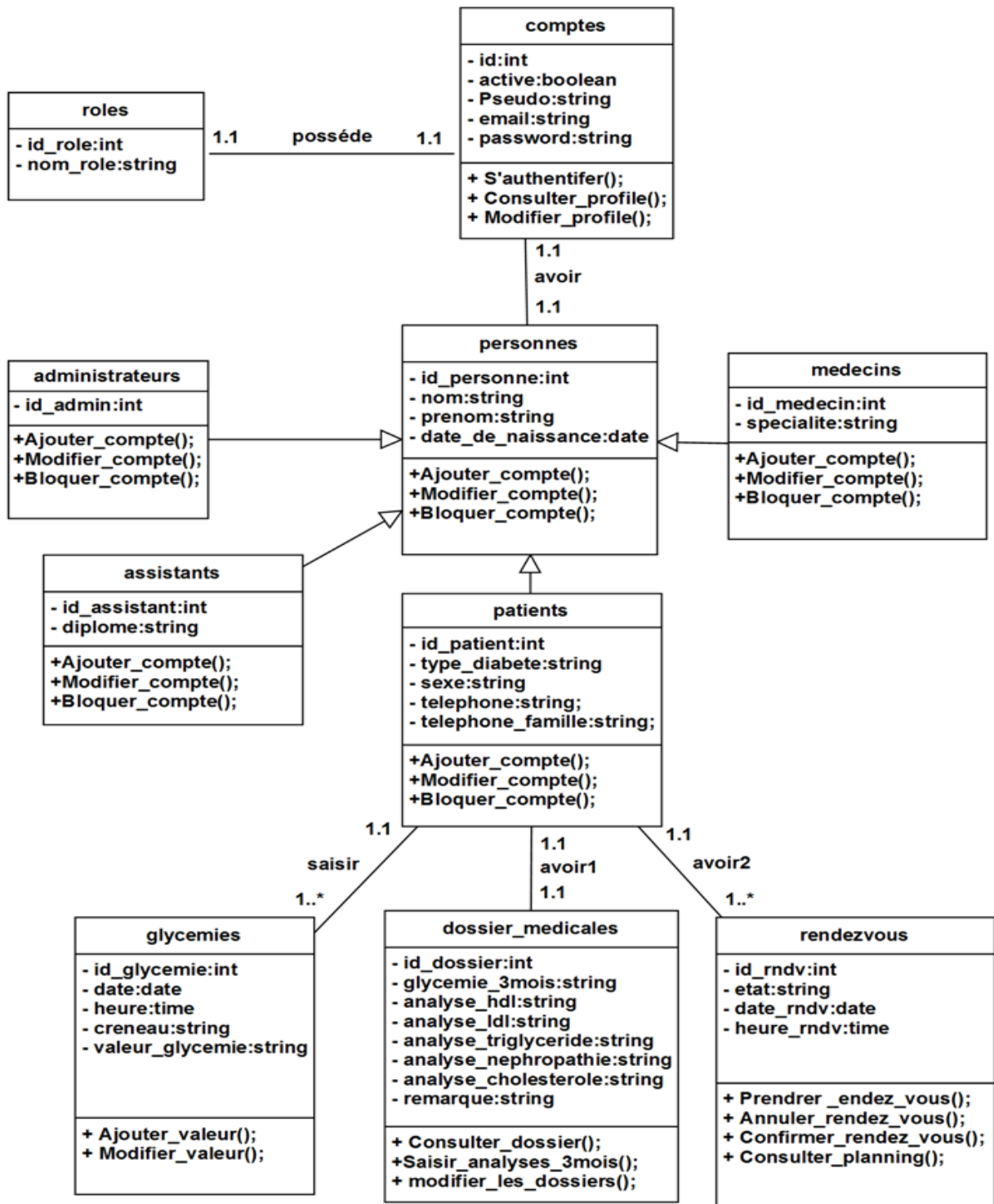


Figure 5.2 – diagramme de classes détaillé de notre système

5.4 Le modèle relationnel

A partir de la description conceptuelle que nous avons effectuée, on peut réaliser le modèle relationnel ; vu que le système d'information ne peut pas le manipulé directement ; et ça en utilisons des règles de passage de l'UML vers le relationnel.[14]

Quelques notions essentielles :[14]

- **Domaine** :c'est l'ensemble des valeurs d'un attribut.
- **Relation** :c'est un sous ensemble du produit cartésien d'une liste de domaine. C'est en fait un tableau à deux dimensions dont les colonnes correspondent aux Domaines et dont les lignes contiennent des tuples. On associe un nom à Chaque colonne.
- **Attribut** :c'est une colonne d'une relation, caractérisé par un nom.
- **Tuples** :c'est la liste des valeurs d'une ligne d'une relation.
- **Cardinalité** :elle permet de définir les conditions de participation d'une entité à une relation. Toutefois, une entité peut participer à plusieurs relations.
- **L'arité** :est le nombre d'attributs d'une relation.
- **Clé** :On distingue deux types de clés :
 1. **Clé primaire** :ensemble d'attributs dont les valeurs permettent de distinguer les nuplets les uns des autres (notion d'identifiant).
 2. **Clé étrangère** :attribut qui est clé primaire d'une autre entité.

NB :pour la notation, nous avons choisi de mettre en gras les clés primaires et de mettre # au début de chaque clé étrangère.

5.5 Les règles de passage

- **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 de clé.
- **Règle 2** :(**Association 1..***) il faut 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.

- **Règle 3 : (Association 1..1)** il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule.
- **Règle 4 : (cas héritage)** transformer chaque sous classe en une relation, la clé primaire de la super classe devient clé primaire de chaque sous classe.
- **Règle 5 : (cas de composition)** migration de clé primaire de la super classe composé clé étranger de la classe composant.

5.6 Les règles de gestion

1. Une personne avoir un et un seule compte (règle 4) .
 - (a) Un administrateur avoir un et un seule compte .
 - (b) Un patient avoir un et un seule compte.
 - (c) Un médecin avoir un et un seule compte .
 - (d) Un assistant avoir un et un seule compte.
2. Un patient avoir un et un seule rendez-vous.
3. Un patient possède un et un seule dossier médicale.
4. Un médecin consulte un ou plusieurs dossiers médicaux.
5. Un patient saisir un ou plusieurs valeurs glycémies.
6. Un assistant gère un ou plusieurs rendez-vous des patients.
7. Un compte possède un et un seule role.

5.7 Les tables de la base de données

En se basant sur les règles ci-dessus, nous avons converti les classes entités et leurs associations, à des tables dans la base donnée. Les tables générées sont :

* **comptes**(id_compte, #role_id, #patient_id, #assistant_id, #medecin_id, #administrateur_id, active, pseudo, email, password).

- * **roles**(id_role, nom_role).
- * **administrateurs**(id_admin, nom_admin, prenom_admin, date_de_naissance_admin).
- * **assistants**(id_assistant, nom_assistant, prenom_assistant, date_de_naissance_assistant, diplome).
- * **patients**(id_patient, nom_patient, prenom_patient, date_de_naissance_patient, type_diabete, sexe, telephone, telephone_famille).
- * **medecins**(id_medecin, nom_medecin, prenom_medecin, date_de_naissance_medecin, specialite).
- * **rendezvous**(id_rndv, #patient_id, etat, date_rndv, heure_rndv).
- * **glycemies**(id_glycemie, #patient_id, date, heure, creneau, valeur_glycemie).
- * **dossier_medicales**(id_dossier, #patient_id, glycémie_3mois, analyse_hdl , analyse_ldl, analyse_triglyceride, analyse_cholesterole, remarque,).

Conclusion

La phase de conception prend en compte les choix d'architecture technique retenue pour le développement et l'exploitation du système. Elle vise à trouver des solutions informatiques et techniques pour mettre en œuvre et construire le système analysé au cours des phases précédentes.

Il nous reste qu'une dernière étape qui est la phase de réalisation qu'on va l'aborder dans le chapitre suivant.

Chapitre 6

L'implémentation

Introduction

Dans ce chapitre, nous allons faire des choix sur le codage. Nous sommes arrivés pratiquement à la fin du processus de développement. Nous commençons par la présentation des langages de programmation, ainsi que les outils de développement utilisés dans notre application. Enfin, nous terminons par la présentation des interfaces de notre application.

6.1 Langages de développement

6.1.1 JAVA

Java est un langage de programmation orienté objet qui a été développé par Sun Microsystems dans les années 90. C'est l'un des langages de programmation les plus populaires, devenu un standard chez les programmes d'entreprise, dans les contenus et jeux Web ou encore les applications mobiles. C'est également le langage de programmation utilisé pour le système d'exploitation mobile Android. Java est fait pour travailler sur de nombreuses plateformes.[11]



Figure 6.1 – Logo java [16]

6.1.2 XML

XML (eXtensible Markup Language) est un langage informatique qui sert à enregistrer des données textuelles. Ce langage a été standardisé par le W3C en février 1998 et est maintenant très populaire. Ce langage, grosso-modo similaire à l'HTML par son système de balisage, permet de faciliter l'échange d'information sur l'internet. Contrairement à l'HTML qui présente un nombre finit de balise, XML donne la possibilité de créer de nouvelles balises à volonté.[15]

6.1.3 PHP

PHP (Hypertext Preprocessor) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement des applications web. Il peut être intégré facilement avec HTML.[15]



Figure 6.2 – Logo php [17]

6.2 Les bases de données

Une base de données est un dispositif permettant de stocker un ensemble d'informations de manière structurée. Il existe plusieurs solutions pour le stockage des données sur android :

- Mettre en place une BDD externe (MySQL, PostgreSQL, Oracle, etc.).

- Mettre en place une BDD interne : en ce qui concerne Android, il s'agit d'une BDD SQLite.
- Stocker les données dans un fichier.[11]

Le choix du type de base dépend évidemment des utilisations que l'on veut faire des données. Dans notre cas nous avons choisi les BDD externe(MySQL).

On a utiliser les fichiers PHP comme intermédiaire entre le MYSQL et l'Android studio.

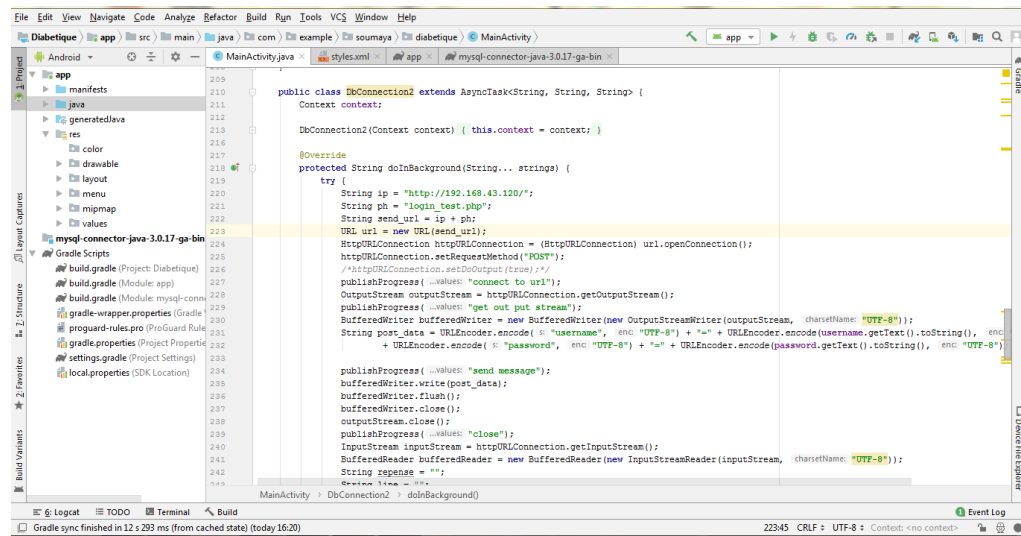


Figure 6.3 – Interface class JAVA

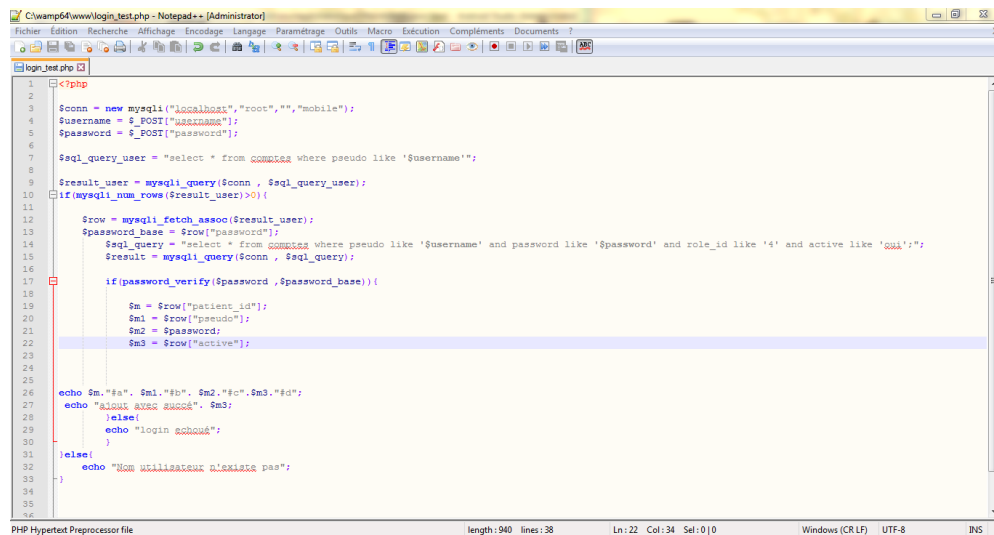


Figure 6.4 – Interface fichier PHP

6.2.1 Les bases de données MySQL

MySQL dérive directement de SQL (Structured Query Language) qui est un langage de requête vers les bases de données exploitant le modèle relationnel. Il en reprend la syntaxe mais n'en conserve pas toute la puissance puisque de nombreuses fonctionnalités de SQL n'apparaissent pas dans MySQL (sélections imbriquées, clés étrangères, etc).

Le serveur de base de données MySQL est très souvent utilisé avec le langage de création de pages web dynamiques : PHP.

L'avantage d'une BDD externe est l'espace de stockage des données qui peut être immense, contrairement à une BDD stockée sur le Smartphone espace de stockage limité. Cependant le temps de réponse de la BDD interne peut être largement inférieur à celui d'une BDD externe.[11]

6.3 Outils de développement

6.3.1 Android Studio

Google a créé un IDE complet pour la création d'application mobile Android nommé Android Studio, annoncé lors du Google I/O le 15 mai 2013. Il est open source et disponible gratuitement, permettant de réaliser des projets sur différents types de support, tablette ou smartphone. Principalement utilisé pour éditer des fichiers Java étant le langage d'une application android native ainsi que des fichiers de mise en page xml avec la possibilité de visualiser le rendu et les manipuler en utilisant une interface graphique.[24]

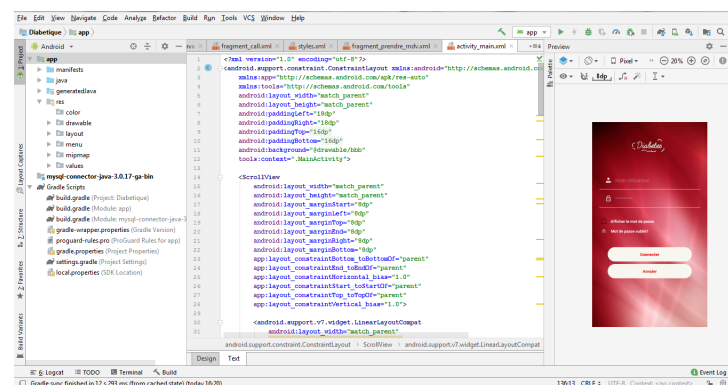


Figure 6.5 – L'environnement Android Studio

6.3.2 Le framework Laravel

6.3.2.1 Le Framework

Un framework c'est tout simplement un ensemble d'outil venant à faciliter le travail du développeur lui permettant d'obtenir des gains en productivité. Il constitue une base cohérente et regroupe en général les fondations d'un logiciel informatique ou d'une application web.

Dans le cas du php, les frameworks regroupent un ensemble de fonctions de base que l'on retrouve sur la plupart des applications php. C'est le cas par exemple du système d'authentification, de la gestion des erreurs ou encore la gestions des droits (administrateurs, rédacteurs, visiteurs, etc.).[25]

6.3.2.2 L'architecture MVC (Modèle-Vue-Contrôleur)

- Modèle : gestion des données de l'application.
- Vue : affichage et interaction avec l'utilisateur.
- Contrôleur : gestion de la logique du code (il prend les décisions).[25]

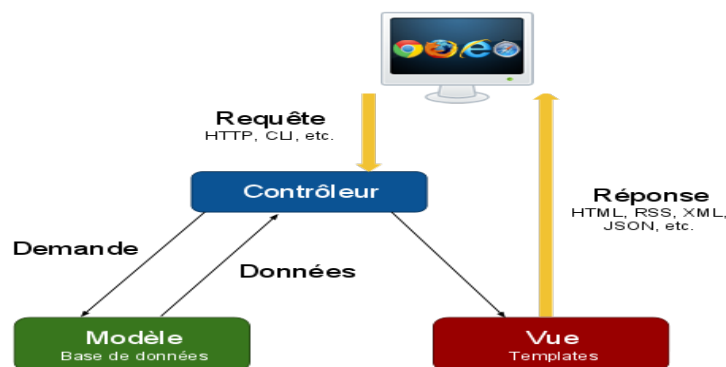


Figure 6.6 – L'architecture MVC [25]

6.3.2.3 Les avantages du modèle MVC

- La séparation des tâches, séparer la logique métier, l'interface utilisateur et la dynamique du système.
- Une modularité qui offre une grande souplesse pour organiser le développement du site entre différents développeurs.
- Une rapidité de développement.

- La réutilisabilité et gain en temps, cette option fournit aux développeurs la possibilité d'utiliser et de substituer les Contrôleurs et Vues pour le même Modèle afin de fournir une autre interface.[25]

6.3.2.4 Laravel

Laravel est un framework PHP open sources basé sur l'architecture MVC. Sortie en juin 2011, il est actuellement l'un des frameworks PHP les plus populaires, créé par Taylor Otwell, initie une nouvelle façon de concevoir un framework en utilisant ce qui existe de mieux pour chaque fonctionnalité. Par exemple, toute application web a besoin d'un système qui gère les requêtes HTTP. Plutôt que de réinventer quelque chose, le concepteur de Laravel a tout simplement utilisé celui de Symfony en l'étendant pour créer un système de routage efficace. De la même manière, l'envoi des courriels se fait avec la bibliothèque SwiftMailer. En quelque sorte, Otwell a fait son marché parmi toutes les bibliothèques disponibles. Nous verrons dans cet ouvrage comment cela est réalisé. Néanmoins, Laravel n'est pas seulement le regroupement de bibliothèques existantes ; c'est aussi un ensemble de nombreux composants originaux et surtout une orchestration de tout cela.[25]

Vous allez trouver dans Laravel :

- Un système d'authentification pour les connexions.
- Un système de validation.
- Un système de migration pour les bases de données.
- Un système d'envoi de courriels.
- Un système d'autorisations, et bien d'autres choses encore.[25]

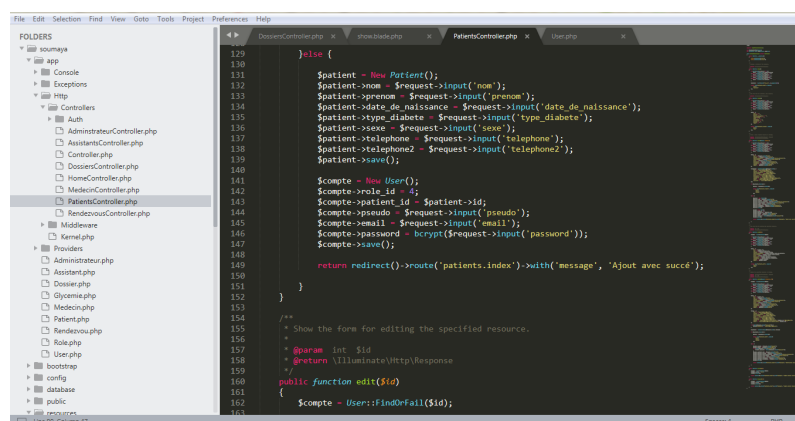


Figure 6.7 – Le Framework Laravel

6.3.2.5 Les statistiques de part de marché pour les Frameworks PHP en 2019

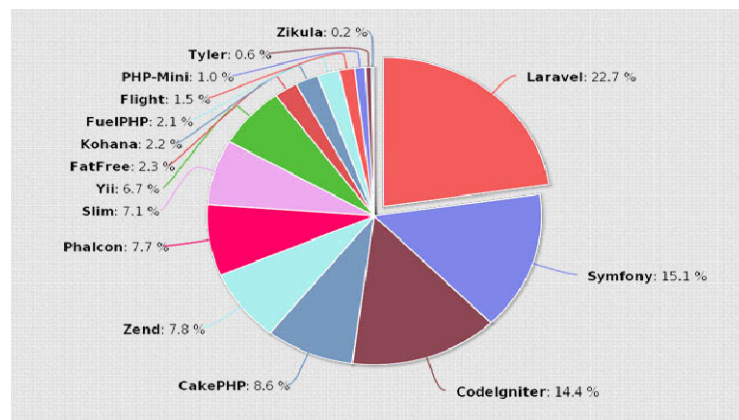


Figure 6.8 – Les statistiques de parts de marché pour les Frameworks PHP en 2019[26]

6.3.3 WampServer

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache, du langage de scripts PHP et d'une base de données MySQL. Il possède également PhpMyAdmin pour gérer plus facilement vos bases de données.[15]



Figure 6.9 – Logo wamp server [18]

6.3.4 Pacestar UML Diagrammer

Pacestar UML Diagrammer est une alternative à d'autres programmes qui aident les programmeurs, ingénieurs système ou d'autres professionnels pour créer des diagrammes UML. Il est livré avec un grand nombre de fonctionnalités, ainsi que des options de personnalisation et incluent tous les outils nécessaires pour aider les utilisateurs des schémas de conception à des fins diverses.[15]



Figure 6.10 – Logo Pacestar UML Diagrammer [19]

6.4 Présentation des interface de notre application

1. Coté web :



Figure 6.11 – Login



Figure 6.12 – Espace administrateur

DIABETES

AJOUTER MÉDECIN

Nom
Bensaid

Prénom
mouhammed

Date de naissance
02/02/1956

Spécialité
diabétologue

Nom utilisateur
mouhammed

Email
mouhammed.ben@gmail.com

Mot de passe

Ajouter Annuler

Figure 6.13 – Interface ajouter médecin

DIABETES

AJOUTER DOSSIER MÉDICALE

Patient
Benali Omar

Glycémie 3 mois
4.3%

Analyse Hdl
0.37 G/L

Analyse Ldl
1.0 G/L

Analyse Triglyceride
0.53 G/L

Analyse MicroAlbuminurie
29%

Analyse Cholesterol
1.30 G/L

Ajouter Annuler

Figure 6.14 – Ajouter dossier médicale

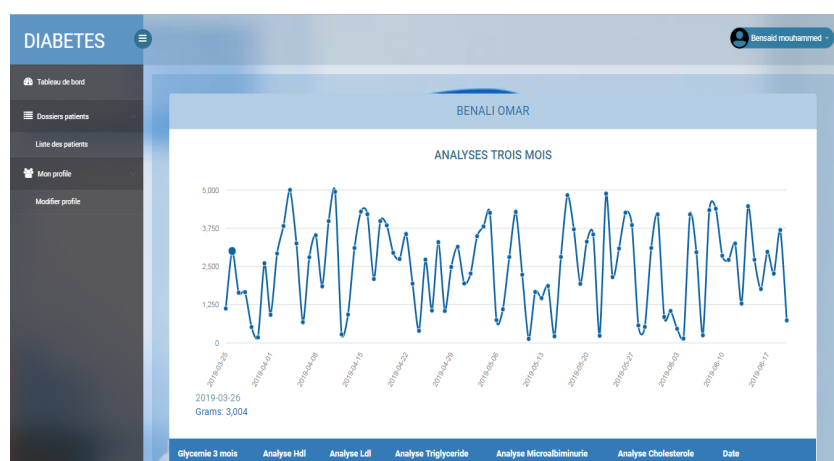


Figure 6.15 – Graphe des valeurs glycémies

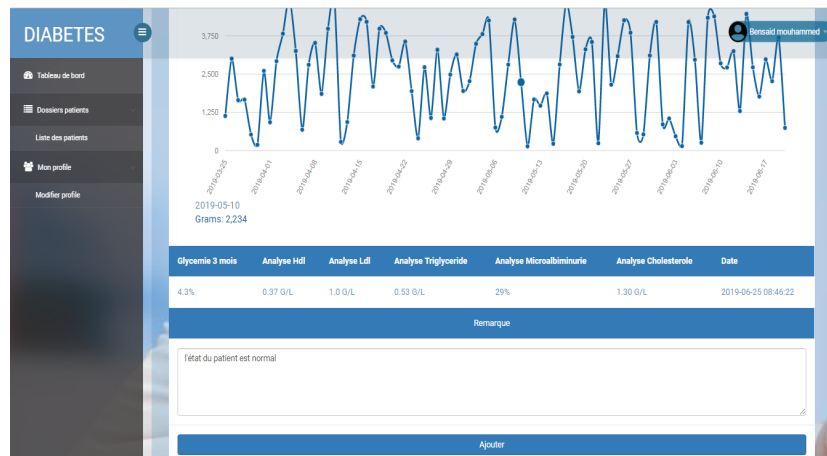


Figure 6.16 – Dossier médicale d'un patient



Figure 6.17 – Planning des rendez-vous

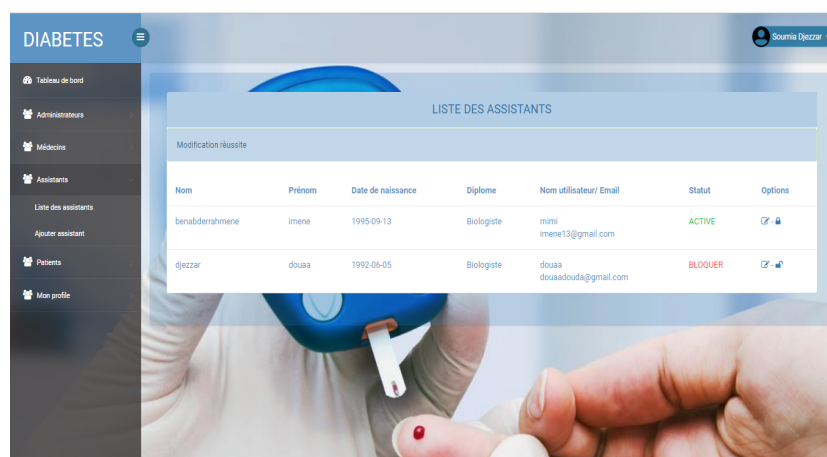


Figure 6.18 – Liste des assistants

2. Coté mobile :

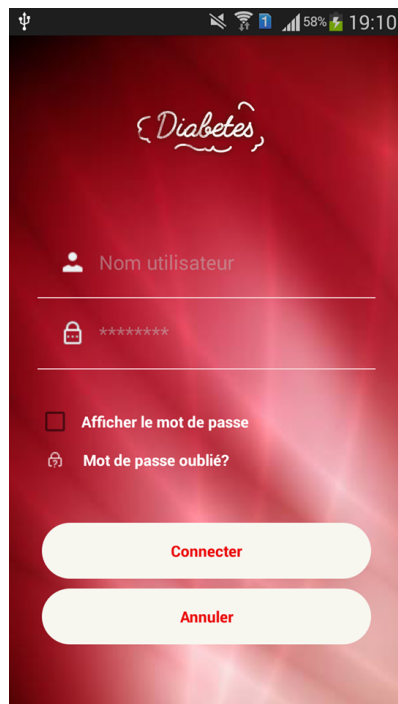


Figure 6.19 – Login

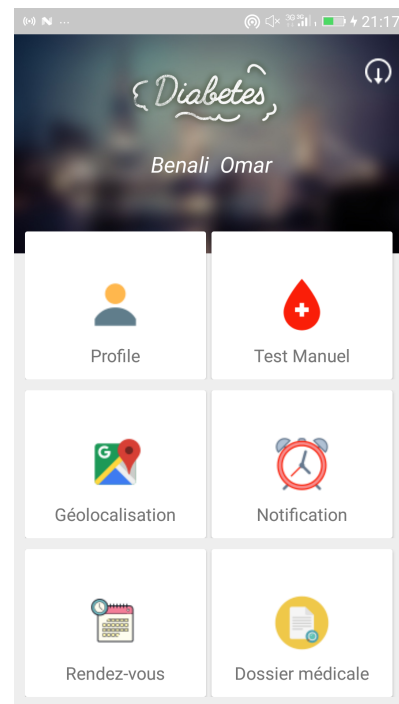


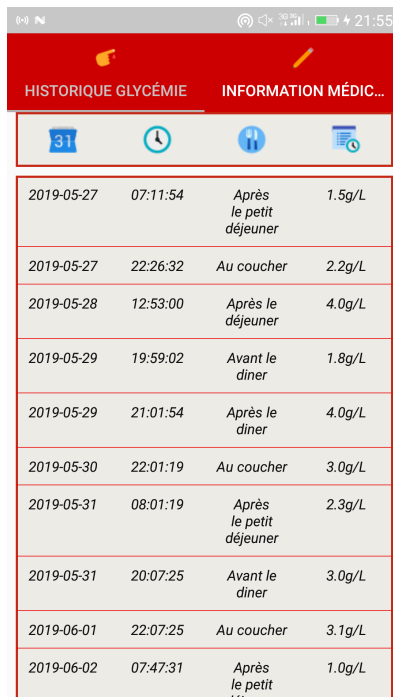
Figure 6.20 – Accueil



Figure 6.21 – Ajouter valeur

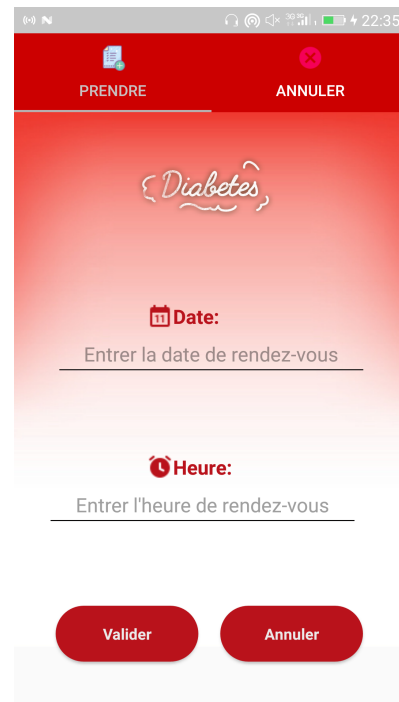


Figure 6.22 – Profile



Date	Heure	Repas	Valeur (g/L)
2019-05-27	07:11:54	Après le petit déjeuner	1.5g/L
2019-05-27	22:26:32	Au coucher	2.2g/L
2019-05-28	12:53:00	Après le déjeuner	4.0g/L
2019-05-29	19:59:02	Avant le dîner	1.8g/L
2019-05-29	21:01:54	Après le dîner	4.0g/L
2019-05-30	22:01:19	Au coucher	3.0g/L
2019-05-31	08:01:19	Après le petit déjeuner	2.3g/L
2019-05-31	20:07:25	Avant le dîner	3.0g/L
2019-06-01	22:07:25	Au coucher	3.1g/L
2019-06-02	07:47:31	Après le petit déjeuner	1.0g/L

Figure 6.23 – Historique



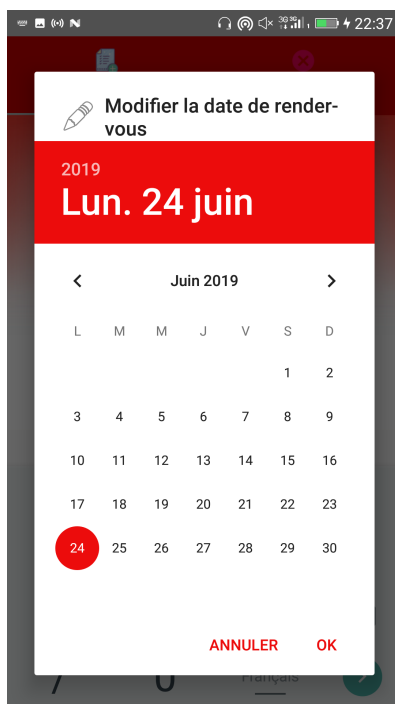
Diabetes

Date:
Entrez la date de rendez-vous

Heure:
Entrez l'heure de rendez-vous

Valider **Annuler**

Figure 6.24 – Prendre rendez-vous



Modifier la date de rendez-vous

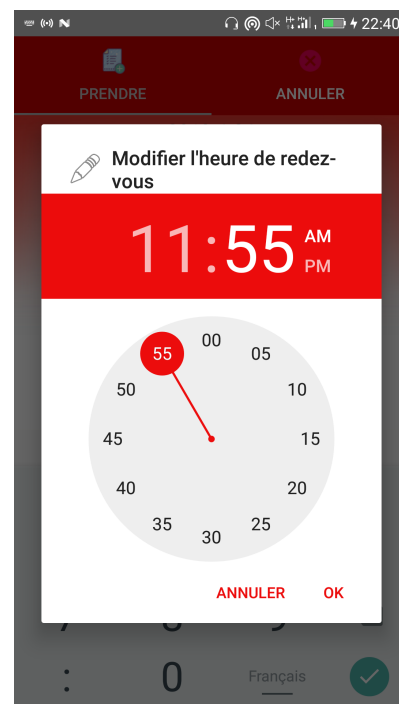
2019
Lun. 24 juin

< Juin 2019 >

L	M	M	J	V	S	D
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

ANNULER **OK**

Figure 6.25 – pour sélectionner la date de rendez-vous



Modifier l'heure de rendez-vous

11:55 AM PM

ANNULER **OK**

Figure 6.26 – pour sélectionner l'heure de rendez-vous

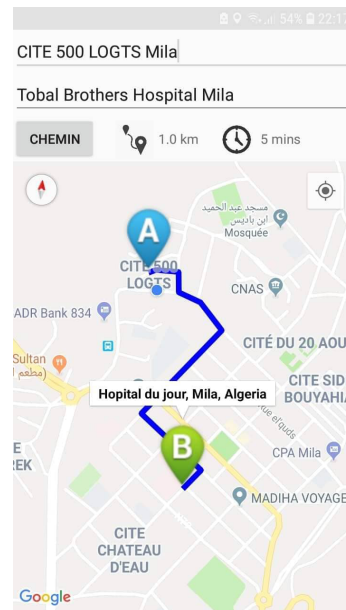


Figure 6.27 – Géolocalisation de l'hopitale

Conclusion

Dans ce dernier chapitre nous avons présenté tout ce qui est lié au développement de notre application. Nous avons parlé des différents outils et choix technologiques utilisés dans l'implémentation des différents composants. Le chapitre est clôturé par une description détaillée des interfaces graphiques de notre application. Il ne reste qu'à conclure ce projet dans la section suivante.

Conclusion générale

Dans ce projet, nous avons projeté la lumière sur l'une des plus remarquables disciplines médicales utilisant les nouvelles technologies de l'information, la télémédecine, son intérêt ainsi que son application au diabète.

En effet, le stage nous a permis de prendre certaines responsabilités, par la suite de consolider de plus en plus nos connaissances théoriques et pratiques. C'est là que réside la valeur d'un tel projet de fin d'études qui combine les exigences de la vie professionnelle aux côtés bénéfiques de l'enseignement pratique que nous avons eu à l'UVT.

Pour cela, nous avons en premier lieu présenté le monde mobile sous différents aspects : appareils mobiles, OS pour mobiles ainsi que les différentes stratégies du développement mobile. Comme plateforme, notre choix s'est porté sur Android, de ce fait, ses caractéristiques et son architecture ont été amplement détaillées.

Nous avons utilisé le langage UML pour la modélisation du système en suivant les étapes du processus (2 TUP). Au niveau du développement de l'application, nous avons utilisé Android Studio et Laravel, basée sur la souplesse de Java, XML (pour Android) et PHP (pour Laravel), qui sont des plates-formes très puissantes, elles intègrent différents outils qui facilitent le développement des applications mobiles et webs. Pour le déploiement nous avons utilisé le serveur web Apache, et le SGBD MySQL pour la base de données.

Malgré ce qui a été fait, ce travail n'est pas encore terminé il peut être enrichi par d'autres fonctionnalités telles que l'utilisation des capteurs, les agents mobiles, une boîte de messagerie, etc. Enfin, nous espérons que ce modeste mémoire soit un modèle pour les autres étudiants et que sa lecture a été agréable et claire.

Bibliographie

- [1] **BAGHLI Zineb, BENMAZOUZ Hakim.** Conception et réalisation d'une application mobile sous Android de recherche de corps médicaux. Mémoire de licence. Tlemcen : Université Abou Bakr Belkaid-Tlemcen, 2015, 36p.

- [2] **Altaama Hadjer.** Application Mobile Guide. [en ligne]. Système d'Information et de Connaissances (S.I.C). Tlemcen : Université Abou Bakr Belkaid-Tlemcen, 2016, 64p.

- [3] **Dunn, Olprod.** Xamarin [en ligne].(crèer le 28/02/2018)Disponible sur :« [https ://docs.microsoft.com/fr-fr/xamarin/android/app-fundamentals/activity-lifecycle/](https://docs.microsoft.com/fr-fr/xamarin/android/app-fundamentals/activity-lifecycle/) ». (consultée le 04/02/2019).

- [4] Les systèmes d'exploitation des smartphones. (n.d.). Retrieved mai 10, 2016, from Mon Petit Monde : www.monpetitmobile.com/choisirmobile/systemes-exploitation-smartphones.

- [5] DIABETE, S. (n.d.). Guide des applications diabètes. Retrieved mars 20, 2016, from dmdpost : [http ://gad.dmdpost.com/](http://gad.dmdpost.com/).

- [6] **Dheeraj Vuppuliri.** Android 'N' Many More [en ligne]. (créer le 23/11/2015) Disponible sur : « [http ://androidnmanymore.blogspot.com/2015/11/android-architecture.html](http://androidnmanymore.blogspot.com/2015/11/android-architecture.html) ». (consulté le 0/02/2019).

- [7] **BAKOUR Mehdi, BENNACER Nadjim.** Etude et réalisation d'un glucomètre non-invasif connecté et une application de Télédiabète sous Android. [en ligne]. Télémédecine. Tlemcen : Université Aboubakr Belkaïd– Tlemcen –, 2017, 75p Disponible sur :« [http ://dspace.univ-tlemcen.dz/bitstream/112/10621/1/Ms.EBM.BAKOUR-BENNACER.pdf](http://dspace.univ-tlemcen.dz/bitstream/112/10621/1/Ms.EBM.BAKOUR-BENNACER.pdf) ».
- [8] **Lisette Cazellet.** Santé connectée et la télémédecine [en ligne]. (mise à jours le 18 décembre 2018).Disponible sur :« [http ://www.telemedaction.org/437468922](http://www.telemedaction.org/437468922) ». (consulté le 8/2/2019).
- [9] **Elisa Frisullo.** le pancréas artificiel expérimenté sur des patients, un espoir pour les diabétiques [en ligne]. (mise à jours le 27/06/2017). Disponible sur : « [https ://www.20minutes.fr/lyon/2094099-20170627-lyon-pancreas-artificiel-experimente-patients-espoir-diabetiques](https://www.20minutes.fr/lyon/2094099-20170627-lyon-pancreas-artificiel-experimente-patients-espoir-diabetiques) ».(consulté le 08/2/2019).
- [10] Agences de services en particuliers. téléassistance securitas [en ligne]. (mise à jours le 16/06/2017). Disponible sur :« [http ://lafleche.generaledesservices.com/teleassistance-securitas-a1823.html](http://lafleche.generaledesservices.com/teleassistance-securitas-a1823.html) ».(consulté le 08/2/2019).
- [11] **BENMAHIDDINE Djamel, BOUADJENEK Yacine.** Conception et réalisation d'une application mobile sous Android pour Le suivi des diabétiques. [en ligne]. Administration et sécurité des réseaux. Bejaia : Abderrahmane Mira, Bejaia. 2016, 77p. Disponible sur :« [http ://www.univ-bejaia.dz/dspace/bitstream/handle/123456789/706/Conception/20et/20realisation/20d'une/20application/20mobile.pdf?sequence=1.isAllowed=y](http://www.univ-bejaia.dz/dspace/bitstream/handle/123456789/706/Conception/20et/20realisation/20d'une/20application/20mobile.pdf?sequence=1.isAllowed=y) ».
- [12] **SAICHE Cylia, OUYOUGOUTE Abdelatif.** Conception et réalisation d'une application web pour la gestion des étudiants d'une école privée. [en ligne]. Bejaia : Université A/Mira de Bejaia, 2015, 70p. Disponible sur :« [http ://www.univ-bejaia.dz/dspace/bitstream/handle/123456789/556/Conception20/et20/r/C3/](http://www.univ-bejaia.dz/dspace/bitstream/handle/123456789/556/Conception20/et20/r/C3/)

- A9alisation/20d/27application/20web/20pour/20la.pdf »(consulté le 15/02/2019).
- [13] **Frédéric MILLOGO**. mise en place d'une application webmapping de géo-localisation des point d'intérêts de la ville de Ouagadougou. [en ligne]. Système d'information géographique. université de Douala et de l'AUF, 2012, 70p. Disponible sur :« [https ://www.memoireonline.com/05/13/7195/m-Mise-en-place-dune-application-webmapping-de-geolocalisation-des-points-dintert-de-la-vill6.html](https://www.memoireonline.com/05/13/7195/m-Mise-en-place-dune-application-webmapping-de-geolocalisation-des-points-dintert-de-la-vill6.html) »(consulté le 15/02/2019).
- [14] **Pascal Roques, Franck Vallée**. UML2 en action. Eyrolles, 2007, 394p.
- [15] **MOUSSOUNI Zobir, RAMDANI Massinissa**. Conception et réalisation d'une application mobile pour le service de tourisme, cas d'étude "Wilaya de Bejaia". Génie Logiciel. éjaïa :Université Abderrahmane Mira de Béjaïa, 2016-2017, 102p.
- [16] **Oxiane**. Oxiane [en ligne]. (créer le 15 mai 2009) Diponible sur :« [https ://blog.oxiane.com/2009/04/06/faire-du-ria-avec-des-composants-ajax-et-des-services-rest/logo-java/](https://blog.oxiane.com/2009/04/06/faire-du-ria-avec-des-composants-ajax-et-des-services-rest/logo-java/) »(consulté le 12/04/2019).
- [17] **Suomitech**. Web para la empresa 17mediatraining[en ligne]. (créer le 17/10/2012)Disponible sur :« [https ://suomitech.com/web-para-la-empresa-17mediatraining](https://suomitech.com/web-para-la-empresa-17mediatraining) »(consulté le 12/04/2019).
- [18] **DOCPLAYER**. Pacestar UML Diagrammer[en ligne].(créer le 30/01/2019)Disponible sur :« [https ://docplayer.net/45706969-Pacestar-uml-diagrammer.html](https://docplayer.net/45706969-Pacestar-uml-diagrammer.html) »(consulté le 12/04/2019).
- [19] **eprojet.fr**. Configuration De Wamp[en ligne].(créer le 02/02/2019)Disponible sur :« [https ://www.eprojet.fr/cours/wamp/01-wamp-configuration-de-wamp](https://www.eprojet.fr/cours/wamp/01-wamp-configuration-de-wamp) »(consulté le 12/04/2019).

- [20] **Remita Rachad, Tebbani Aymen.** Développement d'un système d'information pour la gestion des ressources humaines : Automatisation de la gestion de la paie au sein de la CASNOS de Mila. Sciences et Technologies de l'Information et de la Communication(STIC). Mila :CUM, 2017-2018, 183p.
- [21] **Julien Lausson.** Entre Nougat, Oreo et Pie, quelles sont les versions d'Android les plus courantes?. RÉPARTITION DES VERSIONS D'ANDROID [en ligne]. 2019, n 1, p.20-22. (Consulté le 22-06-2019).
- [22] **Altaama Hadjer.** Application Mobile Guide. Système d'Information et de Connaissances (S.I.C). Tlemcen : Université Abou Bakr Belkaid– Tlemcen, 2015-2016, 64p.
- [23] **Christophe Auffray.** Chiffres clés : les OS pour smartphones. le marché des smarphones [en ligne]. 2018, n 2, p.30-31.Disponible sur : <https://www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.htm?fbclid=IwAR3F1CYi6kSc7fXk0QOCTPwpFeRknHpMKkTvdHJtWXPPh3LM2z6HH9q2txV8>. (Conculté le 22-06-2019).
- [24] **Maxime Nogues.** Présentation Android Studio. Présentation Android Studio[en ligne]. 2016. Diponible sur :<https://www.supinfo.com/articles/single/1641-presentation-android-studio>. (Consulté le 22-06-2019).
- [25] **Maurice Chavelli.** Découvrez le framework PHP LARAVEL [en ligne]. Paris : ÉDITIONS EYROLLES, 2016, 30p pdf. Disponible sur : <https://www.eyrolles.com/Chapitres/9782212143980/9782212143980.pdf>. (Consulté le 22-06-2019).
- [26] **ValueCoders.** 15 Top PHP Frameworks For Startups | 2019 [en ligne]. (Créer le 20-02-2007, modifier le 03-03-2019) Disponible sur : <https://www.valuecoders.com/blog/technology-and-apps/top-popular-php->

frameworks-web-dev/ (Consulté le 22/06/2019).