



République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Sousse
Institut Supérieur d'Informatique et des Techniques de Communication
Hammam Sousse



Rapport de Projet de Fin d'Études

Présenté en vue de l'obtention du
Diplôme de Mastère SSA-IOT

Réalisé par : Rim Jellibi

Développement d'une Solution « EHealth »

Encadrante technique : **Mme. Ines Ben Abdallah**

Encadrante pédagogique : **Mme. Sonia Mili**

Société d'accueil



Année Universitaire 2018 - 2019

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrante Technique : **Mme. Ines Ben Abdallah**

Signature et cachet

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrante Pédagogique : **Mme. Sonia Mili**

Signature

Dédicaces

Je dédie ce modeste travail :

*A la lumière de mes jours, mes parents : **Sadok et Cherifa***

Aucune dédicace ne saurait être assez éloquente pour exprimer mon amour éternel, mon respect et ma considération pour les sacrifices que vous avez consentis pour ma formation et mon bien être. Puisse dieu vous combler de santé et vous procurer une longue vie.

*A ma chère sœur : **Nada***

Tu m'as soutenu, réconforté et encouragé. Puissent nos liens se consolider et se pérenniser encore plus.

*A ceux qui ont toujours été mes sources d'inspiration : **Mes professeurs***

Je tiens à vous écrire un « Merci » sincère pour votre soutien, votre enseignement et vos conseils tout au long de mes cinq années au sein de l'ISITCOM.

A toutes les personnes qui me sont chères, et à toutes celles qui veulent partager mon bonheur

Rim jellibi

Remerciement

C'est avec un grand plaisir que je tiens tout d'abord à exprimer mes sincères et profonds remerciements envers toute personne qui a collaboré de près ou du loin à la réussite de ce modeste travail.

Dès lors, je profite d'exprimer mon immense gratitude et respectueuse connaissance à :

Mon encadrante académique Madame **Sonia Mili** pour son encadrement, son soutien, sa disponibilité et ses conseils qui m'ont guidé tout au long de mon stage ainsi que sa confiance tout au long de la réalisation de ce projet.

Monsieur **Anis Sahbani** de m'avoir accueilli dans son entreprise, de m'avoir donné l'occasion d'être associé à son travail et d'acquérir de nouvelles connaissances et compétences.

Mon encadrante de la société Madame **Ines Ben Abdallah** pour son intérêt à accepter de m'encadrer, son encouragement permanent ainsi que ses critiques constructives.

Les différents experts rencontrés, et notamment Monsieur **Khaled Salhi** et madame **Safa** pour leurs conseils si précieux et tout le temps qu'ils m'ont octroyé.

Mes chers professeurs et plus particulièrement les membres du jury pour avoir accepté de faire partie du jugement de mon travail.

Table des matières

Introduction générale	1
Chapitre 1: Cadre général du projet.....	3
Introduction	4
1.1 Présentation de l'entreprise	4
1.1.1 Organisation de l'entreprise	4
1.1.2 Produits et services	5
1.2 Présentation du projet.....	8
1.2.1 Problématique.....	8
1.2.2 Étude de l'existant.....	8
1.2.3 Solution	10
1.3 Méthodologie du travail	11
1.3.1 Méthode agile	11
1.3.2 Méthode Scrum	12
Conclusion.....	15
Chapitre 2 : Lancement du projet.....	16
Introduction	17
2.1 Analyse et Spécification des besoins	17
2.1.1 Identification des acteurs.....	17
2.1.2 Diagramme de contexte.....	17
2.1.3 Besoins fonctionnels	18
2.1.4 Besoins non fonctionnels	18
2.2 Architecture de la solution	19
2.2.1 Architecture physique.....	19
2.2.2 Architecture technologique	19

2.3 Product backlog	20
2.3.1 Technique de priorisation du PB	20
2.3.2 Transposition du PB en diagramme de cas d'utilisation général	22
2.4 Pratique de principes Scrum	23
2.4.1 Établissement de la « Definition of Done »	23
2.4.2 Planification des sprints	24
2.5 Environnement matériel :	25
2.5.1 kit médical MySignal :	25
2.5.2 Robot de téléprésence « COVEA »	31
2.6 Environnement logiciel	32
Conclusion	34
Chapitre 3. Sprint1 : Connexion, Gestion des médecins	35
Introduction	36
3.1 Ehealth Sprint 1 planning	36
3.1.1 Objectif du sprint 1	36
3.1.2 Sprint Backlog du Sprint 1	36
3.2 Conception	38
3.2.1 Diagramme de cas d'utilisation du premier sprint	38
3.2.2 Diagrammes de séquences	40
3.2.3 Diagramme de classe	42
3.3 Réalisation	43
3.4 Test et validation	45
Conclusion	46
Chapitre 4. Sprint2 :Gestion des patients , Gestion des diagnostics	47
Introduction	48
4.1 Ehealth Sprint 2 planning	48
4.1.1 Objectif du sprint 2	48

4.1.2 Sprint Backlog du Sprint 2	48
4.2 Conception	50
4.2.1 Diagramme de cas d'utilisation du deuxième sprint	50
4.2.2 Diagrammes de séquences	51
4.2.3 Diagramme de classe.....	51
4.3 Réalisation.....	52
4.4 Test et validation	54
Conclusion.....	56
Chapitre 5.Sprint3: Suivi états patients ,Consultation mesures médicales . 57	
Introduction	58
5.1 Ehealth Sprint 3 planning.....	58
5.1.1 Objectif du sprint 3.....	58
5.1.2 Sprint Backlog du Sprint 3	58
5.2 Conception	60
5.2.1 Diagramme de cas d'utilisation du troisième sprint.....	60
5.2.2 Diagrammes de séquences	61
5.2.3 Diagramme de classe.....	62
5.3 Réalisation.....	63
5.4 Test et validation	65
5.5 Synthèse de la Release.....	65
5.5.1 Diagramme de classe final	65
5.5.2 Diagramme de déploiement	67
Conclusion.....	67
Chapitre 6. Sprint 4 : Sécurité de la base de données 68	
Introduction	69
6.1 EHealth Sprint 4 planning.....	69
6.1.1 Objectif du sprint 4.....	69
6.1.2 Sprint Backlog du Sprint 3	69

6.2 Authentification	70
6.2.1 Créer un utilisateur	70
6.2.2 Activer l'autorisation.....	71
6.2.3 Démarrer MongoDB sans authentification	71
6.2.4 Se connecter en s'authentifiant	72
6.3 Backup du base et restaurastion de la base	74
6.3.1 Backup du base.....	74
6.3.2 Backup automatique du base.....	75
6.3.3 Nettoyage des anciennes sauvegardes du base.....	75
6.4 Restauration du base de données MongoDB	76
Conclusion.....	77
Conclusion générale	78
Webographie.....	79

Liste des tableaux

Tableau 1.1 Produits développés par Enova Robotics	6
Tableau 1.2 Etude d'existant.....	9
Tableau 1.3 Comparaison des méthodes de travail	11
Tableau 2.1 Product Backlog.....	21
Tableau 2.2 Planification des sprints	24
Tableau 3.1 Sprint Backlog	37
Tableau 3.2 Description textuelle du cas d'utilisation « Authentification »	38
Tableau 3.3 Description textuelle du cas d'utilisation « Ajout médecin ».....	39
Tableau 3.4 Description textuelle du cas d'utilisation « Suppression médecin »	40
Tableau 4.1 Sprint backlog du sprint 2.....	48
Tableau 4.2 Description textuelle du cas d'utilisation « Consulter diagnostic »	50
Tableau 5.1 Sprint backlog du sprint 3.....	57
Tableau 5.2 Description textuelle du cas d'utilisation « Suivre les états des patients »	59
Tableau 5.3 Description textuelle du cas d'utilisation « consulter les valeurs biomédicales des patients »	59
Tableau 6.1 Sprint Backlog du Sprint 4	68

Liste des figures

Figure 1.1 : Logo du Enova robotics	4
Figure 1.1 Organigramme du Enova robotics	4
Figure 1.2: Principe de Scrum.....	13
Figure 2.1: Diagramme de contexte	17
Figure 2.2: Architecture physique de notre solution	19
Figure 2.3: Architecture technologique de notre solution	20
Figure 2.4 : Signification des Priorités	21
Figure 2.5 : Product backlog	22
Figure 2.6 : Diagramme de cas d'utilisation global	23
Figure 2.7 : Definition of done	23
Figure 2.8 : Kit médical mysignal	25
Figure 2.9 : Capteur SPO2	25
Figure 2.10 : Capteur ECG	26
Figure 2.11 : Capteur Airflow	26
Figure 2.12 : Capteur Blood pressure	27
Figure 2.13 : Capteur Glucometer	27
Figure 2.14: Capteur Température	28
Figure 2.15 : Capteur EMG	28
Figure 2.16 : Capteur Spiromètre	28
Figure 2.17 : Capteur GSR	29
Figure 2.18 : Capteur Body position	29
Figure 2.19 : Capteur sonore	30
Figure 2.20 : Carte arduino	30
Figure 2.21 : Carte MySignals	31
Figure 2.22 : Fonctionnement du robot	32

Figure 3.1 : Diagramme de cas d'utilisation du premier sprint	38
Figure 3.2 : Diagramme de séquence du cas d'utilisation « Authentification »	42
Figure 3.3 : Diagramme de séquence du cas d'utilisation « Ajout de médecin »	41
Figure 3.4 : Diagramme de séquence du cas d'utilisation « suppression de médecin	42
Figure 3.5 : Diagramme de classe du premier sprint	42
Figure 3.6 : Interface du « Login ».....	43
Figure 3.7 : Interface du « liste des médecins »	44
Figure 3.8 : Interface du « Modifier médecin »	44
Figure 3.9 : Interface du « Ajout médecin »	45
Figure 3.10 : Test des services web	45
Figure 3.11 : Affichage de la base de données	46
Figure 4.1 : Diagramme de cas d'utilisation du deuxième sprint	50
Figure 4.2: Diagramme de séquence « Consulter diagnostic »	51
Figure 4.3 : Diagramme de classe du deuxième sprint	51
Figure 4.4 : Liste des patients	52
Figure 4.5 : Interface de gestion des patients	53
Figure 4.6 : Liste diagnostics	53
Figure 4.7 : Interface d'ajout de diagnostics	54
Figure 4.8 : Test des services web	54
Figure 4.9 : Contenu de la collection des patients dans la base de données	55
Figure 4.10 : Contenu de la collection des diagnostics dans la base de données	55
Figure 5.1 : Diagramme du cas d'utilisation du troisième sprint	60
Figure 5.2 : Diagramme de séquence du scénario « Suivre les états des patients ».....	61
Figure 5.3 : Diagramme de séquence du scénario « consulter les valeurs biomédicales des patients »	62
Figure 5.4 : Diagramme de classe du sprint 3	62
Figure 5.5 : Interface de suivi des états des patients	63
Figure 5.6 : Interface de récupération de la valeur récupéré par le capteur	63

Figure 5.7 : Interface de l'interface de l'archive	65
Figure 5.8 : Contenu de la collection « archive » dans la base de données	65
Figure 5.9 : Diagramme de classe finale	66
Figure 5.10 : Diagramme de déploiement	67
Figure 6.1 :Création utilisateur en mongodb	70
Figure 6.2 : Liste des utilisateurs pour notre base	70
Figure 6.3 : Activation de l'autorisation	71
Figure 6.4 : Redémarrage du service mongod	71
Figure 6.5 : Démarrer MongoDB sans authentification	71
Figure 6.6 : Afficher les bases MongoDB sans authentification à partir du Robo3T	72
Figure 6.7 : Lancer MongoDB après authentification.....	72
Figure 6.8 : Authentification en Robo3T	73
Figure 6.9 : Authentification en Robo3T pour accéder au base	73
Figure 6.10 : Backup de notre base	74
Figure 6.11 : Backup automatique de notre base	75
Figure 6.12 : Nettoyage des anciennes sauvegardes	75
Figure 6.13 : Nettoyage des anciennes sauvegardes automatiquement	76
Figure 6.14 : Dossier du base de données récupéré	76
Figure 6.15 : Restoration d'une base de données.....	77

Introduction générale

L'e-Health se développe de manière exponentielle et peut recouvrir des formes diverses et variées. A nos jours le numérique est au cœur de l'innovation dans la prise en charge et l'accompagnement des patients.

En quelques années, l'e-Health est devenue incontournable, tant pour les patients que pour les médecins qui l'identifient comme un moyen pour garantir un système de santé équitable et de qualité. Mettant en relation patients et médecins ou plusieurs professionnels de santé entre eux, la télémédecine s'appuie sur les nouvelles technologies pour établir un diagnostic, assurer un suivi de patient ou réaliser des actes médicaux.

Dans ce contexte s'inscrit mon projet de fin d'études effectué au sein de la société Enova ROBOTICS qui consiste à réaliser une application d'assistance médicale. Cette application assure l'acquisition des mesures provenant d'un kit médical « MySignal » monté sur le robot téléprésence « COVEA », ensuite les afficher en temps réel. Ces mesures doivent être sauvegardés pour assurer leur consultation ultérieurement. Selon les mesures obtenues, l'application classe les patients pour mieux alerter le médecin des cas critiques. La base de données contient des données à conserver au mieux et c'est pour cela que nous avons pensé à la sécuriser.

Notre rapport s'articule autour de six chapitres :

- Le premier chapitre « Présentation générale du projet » décrit le cadre du projet avec une présentation de la société d'accueil.
- Le deuxième chapitre « Lancement du projet » tente d'aborder la phase d'analyse en présentant l'architecture de la solution, les besoins fonctionnels et non fonctionnels afin d'aboutir à la description du product backlog et la planification des sprints nécessaires pour la mise en œuvre de l'application.
- Le troisième chapitre qui fait l'objet d'un premier sprint a pour objectif de développer l'interface de connexion et celle de gestion des médecins.
- Le quatrième chapitre présente le deuxième sprint qui vise la gestion des patients et la gestion des diagnostics.

- Le cinquième chapitre expose le troisième sprint qui a pour but la classification des états des patients et la consultation des mesures médicales récupérées.
- Le sixième chapitre nous mène vers la réalisation du quatrième sprint qui vise la sécurisation de la base de données.

Enfin, une conclusion générale clôturera ce rapport et lancera des perspectives qui serviront à concocter le product Backlog de la release suivante .

Cadre général du projet

Sommaire

1.	Présentation de l'entreprise	4
2.	Présentation du contexte du projet.....	Erreur ! Signet non défini.
3.	Présentation du projet	8
4.	Méthodologie du travail.....	11

Introduction

Dans ce premier chapitre, nous allons focaliser l'attention sur le cadre général de notre projet. Il s'agit en effet d'une présentation de la société d'accueil. Nous présenterons par la suite l'existant ainsi que les objectifs de notre travail. Enfin, nous terminerons par le choix de la méthode de gestion que nous allons l'utiliser dans notre projet.

1.1 Présentation de l'entreprise

1.1.1 La société Enova Robotics

« Enova Robotics » fondée en 2014 est la première société tunisienne en Afrique et au Moyen-Orient qui fabrique sa propre marque de robots et les exporte à partir de la Tunisie. Cette société a comme but de développer des solutions mobiles automatisées « Robots intelligents » construites sur mesure et de sa propre marque afin de répondre à des usages principalement dans les domaines de la sécurité, de la santé et de l'enseignement.



Figure 1.1 : Logo du Enova Robotics

1.1.2 Organisation de l'entreprise

La figure 1.2 présente l'organigramme de la société afin de présenter les liens hiérarchiques, organisationnels et fonctionnels existants au sein de la société.

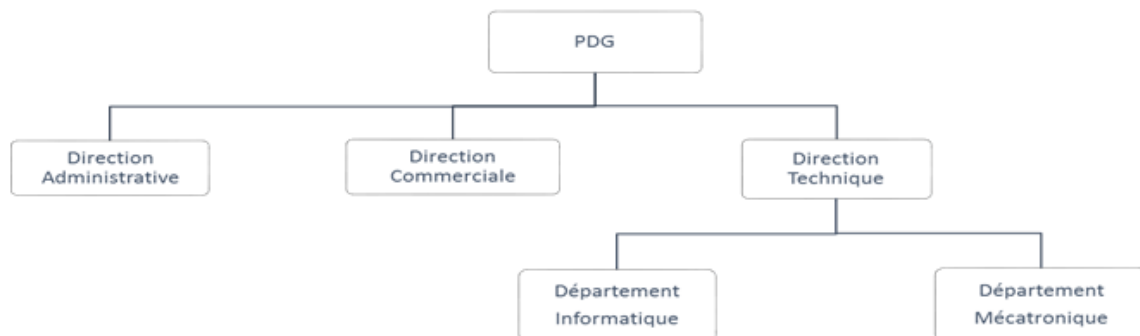


Figure 1.2 Organigramme du Enova Robotics

Au sein d'Enova ROBOTICS le travail est réparti en quatre départements à savoir le département administratif, le département informatique, le département mécatronique et le département commercial.



- **Le département administratif** : Ce département est constitué du directeur de l'entreprise et de l'assistance administrative. Ils prennent en charge l'organisation du travail, le quotidien des activités administratives de l'entreprise et la synchronisation de l'équipe.
- **Le département informatique** : Ce département est formé de six ingénieurs informatiques, deux ingénieurs réseaux, un docteur chercheur en intelligence artificielle et un doctorant. Ils prennent en charge la programmation du robot et de son interface d'utilisation.
- **Le département mécatronique** : L'équipe de ce département est formée de quatre ingénieurs en mécatroniques et un technicien mécanique qui se chargent de la définition des cahiers de charges, de la conception, et de la fabrication des produits de l'entreprise.
- **Le département commercial** : L'équipe de ce département est formée de trois personnes qui sont responsable des études du marché et des besoins clientèle. Ils se chargent d'orienter la production vers les besoins et désirs des consommateurs et s'occupent de la publicité des produits

1.1.2 Produits et services

Au cours de ces cinq années d'existence, Enova Robotics a développé un portefeuille complet de robots :

Tableau1.1: Produits développés par Enova Robotics

Robots	Domaine	Description
 <p>P-GUARD</p>	Sécurité	<p>P-Guard est un robot de sécurité robuste destiné aux applications multi-terrains. Pour remplir ses fonctions, le robot intègre un ensemble de caméras infrarouges agencées de manière à couvrir tout son environnement, un système d’acquisition audio omnidirectionnel, une caméra thermique et un système d’alarme sonore et lumineuse. Le robot intègre un GPS pour localiser dans son environnement et un système de télémétrie laser.</p>
 <p>Mini-Lab</p>	Education	<p>Mini-Lab est un robot mobile de taille moyenne qui offre le meilleur compromis entre robustesse et compétitivité économique</p>
 <p>E Touch-Bot</p>	Santé	<p>E Touch-Bot est un robot conçu pour assister les personnes âgées dans une maison de retraite. C’est la réponse à la pénurie prédéfinie d’aidants naturels pour servir notre population qui vieillit rapidement.</p>

 <p>CovéaLink</p>	<p>Domicile</p> <p>Santé</p>	<p>C'est un robot de téléprésence mobile doté d'une écran, une caméra et un micro dirigeable à distance grâce à une tablette. Il peut intervenir dans plusieurs domaines tels que la santé et l'enseignement.</p>
 <p>OGY</p>	<p>Domicile</p>	<p>C'est un robot de sécurité à domicile développé par Enova Robotics en partenariat avec d'autres entreprises tunisiennes : Chifco et OPCMA et sous la tutelle du cluster mécatronique</p>

1.2 Présentation du projet

1.2.1 Problématique

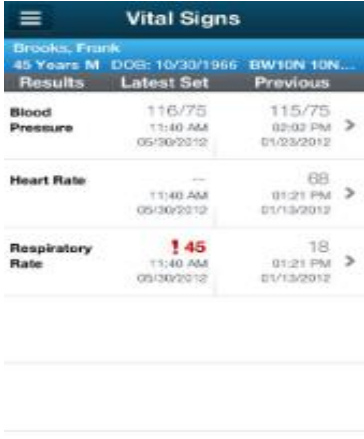

Personne ne peut nier qu'il existe plusieurs problèmes dans le domaine médical parmi ces problèmes on peut citer :

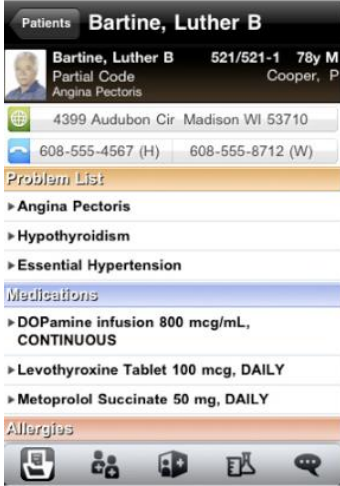
- Le suivi d'un grand nombre de patients est une tâche lourde ce qui rend difficile d'assurer un suivi en temps réel de tous les patients.
- Médecins, infirmiers et personnels disposent d'une masse volumineuse de données qu'ils doivent faire circuler entre eux de façon efficace et instantanée ce qui est difficile à garantir.
- Le rythme de travail des médecins est trop intense ce qui engendre un rendement faible et affecte la qualité de leurs services.

1.2.2 Étude de l'existant

Le staff médical a besoin d'outil d'aide pour alléger leurs tâches et rend leurs interventions rapide et efficace. Plusieurs solutions ont été développées pour pallier à ces problèmes. Ainsi une étude de l'existant s'avère nécessaire en vue de déterminer les points faibles et les points forts des solutions existantes en vue d'en prendre en considération lors de la conception et la réalisation. Dans cette section, nous allons présenter un tableau comparatif des différentes solutions existantes.

Tableau 1.2: Etude d'existant

Application	Description	Points fort	Points faibles
<p>Cerner Physician Express II</p> 	<p>Cette application permet au personnel de consulter les dossiers des patients par nom ou à partir d'une liste et ainsi accéder à des données pertinentes à propos de leurs allergies, diagnostics, médicaments, signes vitaux et plus encore.</p>	<ul style="list-style-type: none"> - Transmission rapide de l'information des patients 	<ul style="list-style-type: none"> - Pas de notification en cas de cas critique - Exige une licence compatible.
<p>Sunrise Mobile</p> 	<p>Cette application permet aux médecins d'accéder aux dossiers médicaux électroniques et elle leur offre un accès aux listes de patients et aux notes des médecins</p>	<ul style="list-style-type: none"> - Facile à utiliser. - Offre plusieurs fonctionnalités à la faveur des médecins : visualiser, analyser et agir. 	<ul style="list-style-type: none"> - Interface très chargée. - N'est pas optimisée avec tous les appareils.

<p>Epic Haiku</p> 	<p>Cette application fournit aux professionnels de la santé la liste de leurs patients à l'hôpital, les résumés de leur état de santé, les horaires cliniques ainsi qu'une section pour saisir des notes. Il permet au personnel hospitalier de voir les médicaments actuels d'un patient d'une manière claire.</p>	<ul style="list-style-type: none"> - Accès sécurisé. - Application multi-platform. 	<ul style="list-style-type: none"> - Impose des options facturées et un peu cher. -Utilisation complexe - Interface chargé
--	---	--	---

1.3.3 Solution

Ce projet consiste à développer une solution pour récupérer des données à partir d'un kit médical mysignal monté sur un robot de téléprésence « COVEA » fabriqué par la société "Enova Robotics". Les données collectées sont affichées via une application mobile cross platform dédiée au médecin. Le médecin peut gérer la liste des patients, consulter leurs états, gérer les diagnostics de chaque patient et récupérer les mesures des capteurs. Cette application va faciliter le transfert des mesures des patients vers leurs médecins et de la rendre flexible et surtout elle va alléger le rythme de travail des médecins. Cette application permet aux médecins de consulter la progression de l'état des patients peu importe où ils sont. Elle offre aux médecins un moyen facilement accessible afin de prendre les décisions adéquates quand ils sont loin de l'hôpital.

1.3 Méthodologie du travail

L'adoption d'une méthodologie de travail est nécessaire pour garantir un niveau de qualité acceptable et respecter les délais fixés.

1.3.1 Méthode agile

Agile représente un ensemble de méthodes et pratiques basées sur les valeurs et les principes du manifeste Agile, qui repose entre autre sur la collaboration, l'autonomie et des équipes pluri-disciplinaires. On a plusieurs méthodes agiles qu'on peut les utiliser, le tableau ci-dessous présente une étude comparative entre les méthodes agiles les plus connus et les plus utilisés afin d'effectuer le bon choix de la méthodologie à adopter.

Tableau 1.3: Comparaison des méthodes de travail

	Description	Points Forts	Points Faibles
XP	<ul style="list-style-type: none"> - Représente un ensemble de pratiques de développement. - Cible des projets de moins de 5 personnes. 	<ul style="list-style-type: none"> - Une planification ultra souple à court horizon. - Livraison rapide de prototype. 	<ul style="list-style-type: none"> - Méthode adaptée à des petites équipes pouvant aller de 10 à 16 développeurs maximum. - Il est impératif de disposer d'une équipe stable.
Kanban	<ul style="list-style-type: none"> - Elle est basée sur le principe de la limitation du nombre de travaux à faire. 	<ul style="list-style-type: none"> - Décentralisation de la gestion de la production. - Gestion visuelle de la production et meilleure implication des employés. 	<ul style="list-style-type: none"> - La gestion des deadlines sont moins visibles. - Risque de rupture de stocks.

Scrum	<ul style="list-style-type: none"> - Une équipe de 5 à 9 personnes - Les décisions sont prises collectivement 	<ul style="list-style-type: none"> - Travail en Equipe et responsabilité collective. - Méthode rigoureuse - Augmentation de productivité 	<ul style="list-style-type: none"> - Difficile à maîtriser. - Scrum est adaptée aux équipes de petite taille .
--------------	---	---	--

Suite à l'étude comparative nous avons constaté que les trois méthodologies étudiées permettent le travail itératif pour un meilleur suivi de l'avancement global du projet.

Dans notre projet nous avons choisi la méthodologie scrum car elle satisfait les conditions suivantes :

- Scrum convient aux équipes ayant un nombre de développeurs réduits : Ceci est le cas car le travail est individuel.
- La progression des tâches s'effectue pendant une durée de développement courte.

1.3.2 Méthode Scrum

1.3.2.1 Présentation de la méthode Scrum

La méthodologie Scrum a été conçue pour améliorer grandement la productivité dans les équipes auparavant paralysées par des méthodologies plus lourdes. Le principe de base de Scrum est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de durée fixe d'une à quatre semaines, appelées sprints. Chaque sprint possède un but à atteindre, défini par le directeur de produit, à partir duquel sont choisies les fonctionnalités à implémenter dans le sprint. Un sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel. Ce processus est illustré par la figure suivante

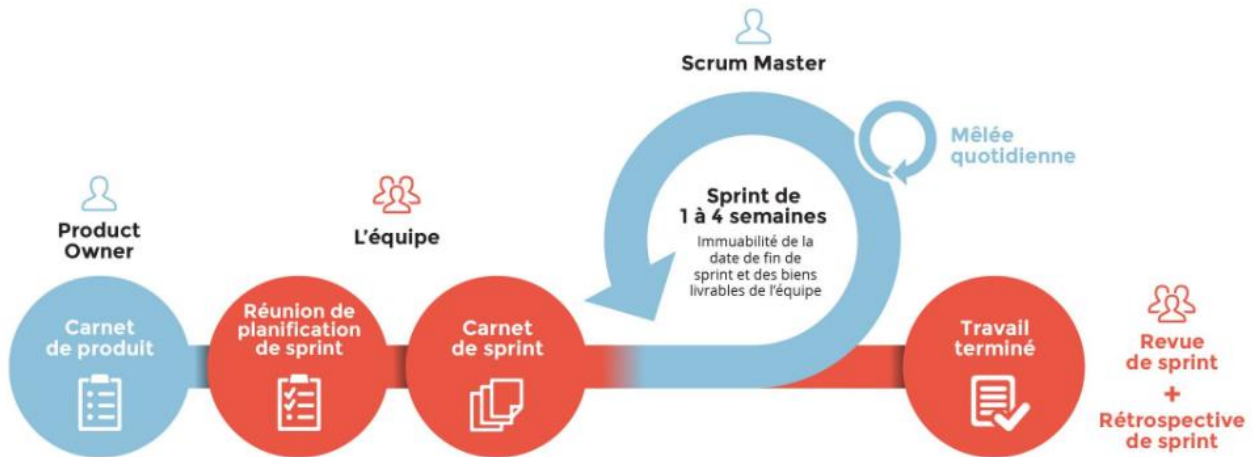


Figure 1.3: Principe de Scrum

La figure ci-dessus décrit le principe de Scrum qui se résume comme suit :

- Propriétaire de produit Le « Product Owner » crée une liste de tâches prioritaires appelée **Product Backlog**.
- Pendant le « Sprint Planning », l'équipe prend une partie des tâches du « Product Backlog » pour préparer le Sprint Backlog et discuter la manière de la réalisation de ces tâches.
- L'équipe a une certaine période de temps de deux à quatre semaines Sprint pour achever le travail, mais elle se réunit chaque jour pour évaluer le progrès « DailyScrum ».
- Tout au long de la période, le « Scrum Master » essaie de maintenir l'équipe concentrée sur son objectif.
- À la fin de chaque Sprint, le travail devrait être potentiellement livrable.
- Au départ du Sprint suivant, l'équipe prend une autre partie des tâches de Product Backlog et répète les étapes précédentes.

Au-delà du Sprint, le cycle se répète jusqu'à ce que la liste des tâches du Product Backlog soit achevée ou le budget du projet soit épuisé ou une date limite soit atteinte. Peu importe la cause de Clôture, « Scrum » assure que le travail le plus important a été achevé lors de la fin du projet.

1.3.2.2 L'Équipe Scrum

Scrum définit un modèle d'équipe optimisant la flexibilité, la créativité et la productivité. Les équipes Scrum sont auto-organisées et pluridisciplinaires. Ces équipes ont toutes les compétences nécessaires pour effectuer le travail.

➤ **Product owner :**

Ce responsable du produit représente les clients et/ou utilisateurs en transcrivant leurs besoins, définit et priorise les demandes produit. Le Product Owner est la seule personne responsable de gérer le carnet de produit (Product Backlog).

➤ **Scrum master :**

Membre de l'équipe de développement. Il doit faciliter l'application de Scrum en aidant ceux qui sont externes à l'équipe Scrum à comprendre lesquelles de leurs interactions avec l'équipe Scrum sont bénéfiques et lesquelles ne le sont pas.

➤ **L'équipe (le team)**

Elle se gère en toute autonomie et est chargée du développement du produit. Sa taille doit rester réduite (7 à 9 personnes). Auto-organisée, dépourvue de hiérarchie, sans spécialisation des rôles, elle est composée des professionnels qui livrent à chaque Sprint un incrément terminé et potentiellement livrable du produit.

On ne peut pas clôturer cette section sans définir les membres de notre équipe « Scrum » ce qui est évident et très intéressant pour notre travail qui clarifie le rôle de chaque membre de l'équipe.

Notre équipe Scrum :

- **Scrum Team :** Rim Jellibi
- **Scrum Master :** Ines ben Abdallah
- **Product Owner :** Anis sahbani

Conclusion

Dans ce chapitre, nous avons présenté la société. Puis nous avons effectué une étude des solutions existantes pour sensibiliser l'importance du projet. Ensuite nous avons dévoilé la problématique qui nous a conduit à proposer une solution. Enfin nous avons évoqué la méthodologie de travail à adopter « Scrum », que nous allons respecter durant les chapitres suivants. Une initialisation du projet fera l'objet du prochain chapitre.

Chapitre 2

Lancement du projet

Sommaire

1. Analyse et Spécification des besoins	17
2. Architecture de la solution	19
3. Product backlog.....	20
4. Pratique de principes Scrum.....	23
5. Environnement matériel	25
6. Environnement logiciel	32

Introduction

Ce chapitre présente une phase d'analyse du projet. Dans un premier lieu nous allons présenter l'architecture convenable de notre solution ainsi que les acteurs nécessaires pour notre application. Dans un deuxième lieu nous allons identifier les besoins fonctionnels et non fonctionnels et présenter le Backlog du produit et la planification des sprints. Enfin nous clôturons notre chapitre par la modélisation du diagramme du cas d'utilisation global du projet.

2.1 Analyse et Spécification des besoins

2.1.1 Identification des acteurs

- **Administrateur** : L'administrateur est le responsable de la partie backend. Il intervient dans la gestion de la liste des médecins qui peuvent accéder à l'application mobile.
- **Médecin** : C'est l'utilisateur de l'application mobile qui permet de gérer la liste des patients, gérer les diagnostics, consulter les statistiques des états des patients et les mesures médicales récupérées.

2.1.2 Diagramme de contexte

Le diagramme de contexte exprime l'environnement du système dans une situation donnée. Il permet donc de définir les frontières de l'étude et de répondre à la question : « quels sont les acteurs et les éléments environnants au système ? »

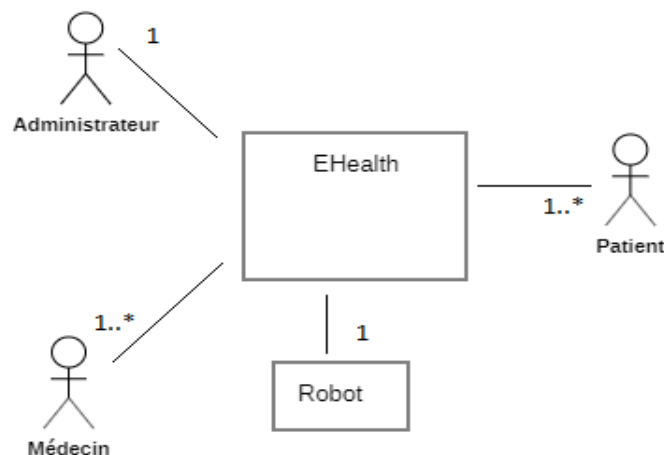


Figure 2.1 : Diagramme de contexte

2.1.3 Besoins fonctionnels

Les besoins fonctionnels constituent les besoins spécifiant le comportement des entrées/sorties relatives au processus. Nous regroupons ces fonctionnalités dans les points suivants:

- Permettre à l'administrateur de s'authentifier.
- Permettre à l'administrateur de gérer la liste des médecins.
- Permettre aux médecins de s'authentifier.
- Permettre aux médecins de gérer la liste des patients.
- Permettre aux médecins de suivre les états des patients.
- Permettre aux médecins de récupérer les valeurs biomédicales d'un patient.
- Permettre aux médecins de gérer les diagnostics des patients.

2.1.4 Besoins non fonctionnels

Les besoins non fonctionnels correspondent à la manipulation de l'application et précisent l'environnement de l'application :

- L'extensibilité : L'architecture de l'application permettra l'évolution au niveau de ses différents modules d'une manière flexible.
- La sécurité : Sécuriser l'accès aux informations en assurant une connexion sécurisée. Ainsi une phase d'authentification est obligatoire pour qu'un utilisateur puisse utiliser l'application.
- La convivialité : L'application doit être simple à utiliser et ergonomique.
- Performance : L'application doit fonctionner avec un temps de réponse minimal.

2.2 Architecture de la solution

2.2.1 Architecture physique

La structure de l'application est adaptable parfaitement à L'architecture trois tiers, ce modèle de l'architecture client-serveur est appelé aussi architecture à trois couches. Donc notre architecture va être divisé en trois parties :

- **Couche de présentation** : C'est l'interface utilisateur qui affiche les données et lui permet d'interagir avec elle.
- **Couche métier** : cette couche s'occupe du traitement d'un ensemble d'actions sur le serveur et l'envoi de la réponse au client.
- **Couche d'accès aux données** : permet de fournir au serveur d'application les données dont il a besoin et de conserver les données et assure leur persistance.

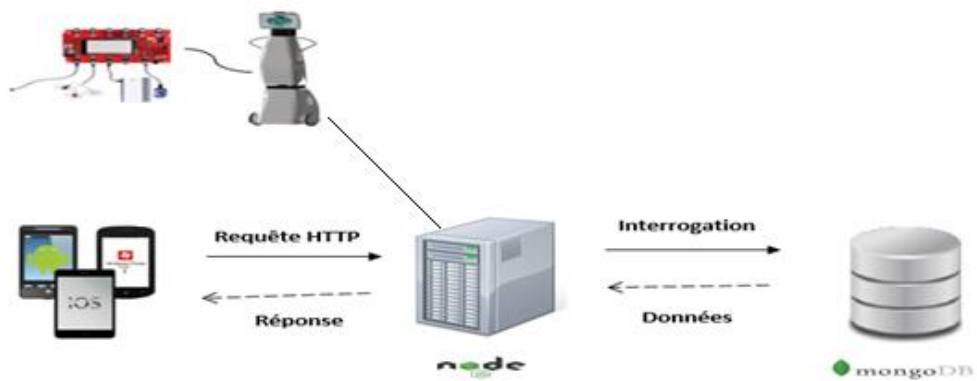


Figure 2.2: Architecture physique de notre solution

2.2.2 Architecture technologique

- **FrontEnd** : Se base sur PhoneGap/Cordova pour créer des applications mobiles multi-plateformes avec les technologies traditionnelles du web.
- **Serveur** : Le framework node js.
- **Base de données** : Toutes les données issues de différentes sources sont sauvegardées dans une base de données Noql orientée documents MongoDB.

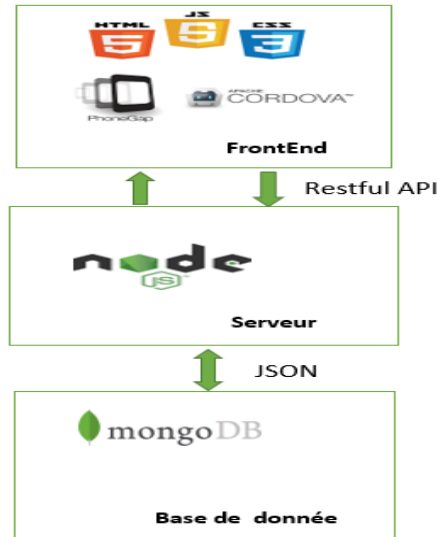


Figure 2.3: Architecture technologique de notre solution

2.3 Product backlog

Le Product Backlog est une liste ordonnée de toutes les fonctionnalités souhaitées par le client formulé sous la forme de « user story ». Dans cette partie nous décrivons le « product Backlog ».

2.3.1 Technique de priorisation du PB

La priorisation du Product Backlog est une technique qui permet d'estimer l'effort nécessaire à une équipe pour implémenter une fonctionnalité dans le but de simplifier la complexité des tâches.

Deux éléments doivent être pris en compte pour l'estimation :

- Le « Business Value » : La valeur ajoutée au client en lui ramenant un maximum de confort et convivialité. La priorisation des fonctionnalités qui permettent d'augmenter la valeur de notre solution.
- Le « Risk Minimizing » : Prioriser les fonctionnalités les plus difficiles et qui nécessitent un maximum d'effort de travail ou plus de recherche.

Pour estimer la taille d'une fonctionnalité nous allons choisir la méthode « MoSCoW »
Les lettres de l'acronyme MoSCoW signifient :

- **M** comme Must have, ce qui est indispensable et obligatoire, ce qui **doit** absolument **être fait**, sans quoi le projet est un échec.
- **S** comme Should have, ce sont les tâches du projet qui doivent être faites mais dans la mesure du possible, elles sont importantes mais pas vitales. S'il s'agit des fonctionnalités d'un logiciel, elles seront développées une fois que celles de la catégorie Must auront été livrées et s'il reste assez de temps disponible.
- **C** comme Could have, cela correspond à des tâches qui peuvent être réalisées dans la mesure où cela n'affecte pas les autres tâches importantes, ce sont des tâches de confort, on les fait que s'il reste assez de temps et que les tâches des 2 premières catégories ont été réalisées.
- **W** Would like but won't get this time, ce sont des tâches auxquelles on renonce car elles sont très secondaires, ce serait possible dans le futur avec un nouveau budget mais elles n'entrent pas dans le budget actuel, ce sont des tâches supplémentaires de « luxe ».
- Les « o » ont été ajoutés pour rendre l'acronyme lisible et mémorisable

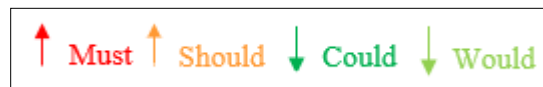


Figure 2.4 : Signification des Priorités

Projets / Ehealth / Tableau EH

Backlog

Share ...

Uniquement mes tickets Récemment mis à jour

VERSIONS EPICS	Backlog 10 tickets		Créer un sprint	
	En tant qu'administrateur, je veux gérer la liste des médecins	EH-1	↑	-
	En tant que médecin , je veux me connecter en toute sécurité	EH-2	↑	-
	En tant que médecin, je veux gérer la liste des patients	EH-3	↑	-
	En tant que médecin, je veux consulter les mesures biomédicales récupéré à partir du kit monté sur le robot	EH-4	↑	-
	En tant que médecin, je veux gérer la liste des diagnostics des patients	EH-5	↑	-
	En tant que médecin, je veux visualiser les patients classés en catégories selon leurs états	EH-6	↑	-
	En tant que médecin, je veux consulter la liste des patients correspondant à chaque état	EH-7	↑	-
	En tant que médecin, je veux chercher un patient précis dans la liste des patients	EH-8	↑	-
	En tant que médecin, je veux avoir une liste archive qui contient la liste des patients supprimés	EH-9	↑	-
	En tant qu'utilisateur, je veux utiliser mon application soit sur iphone ou sur un smartphone Android .	EH-10	↑	-
+ Créer un ticket				

Figure 2.5 : Product Backlog

2.3.2 Transposition du PB en diagramme de cas d'utilisation général

Un diagramme de cas d'utilisation est utilisé pour donner une vision globale du comportement fonctionnel d'un système. Le diagramme de cas d'utilisation global représenté par la figure 2.6 modélise les fonctionnalités présentées dans notre product backlog.

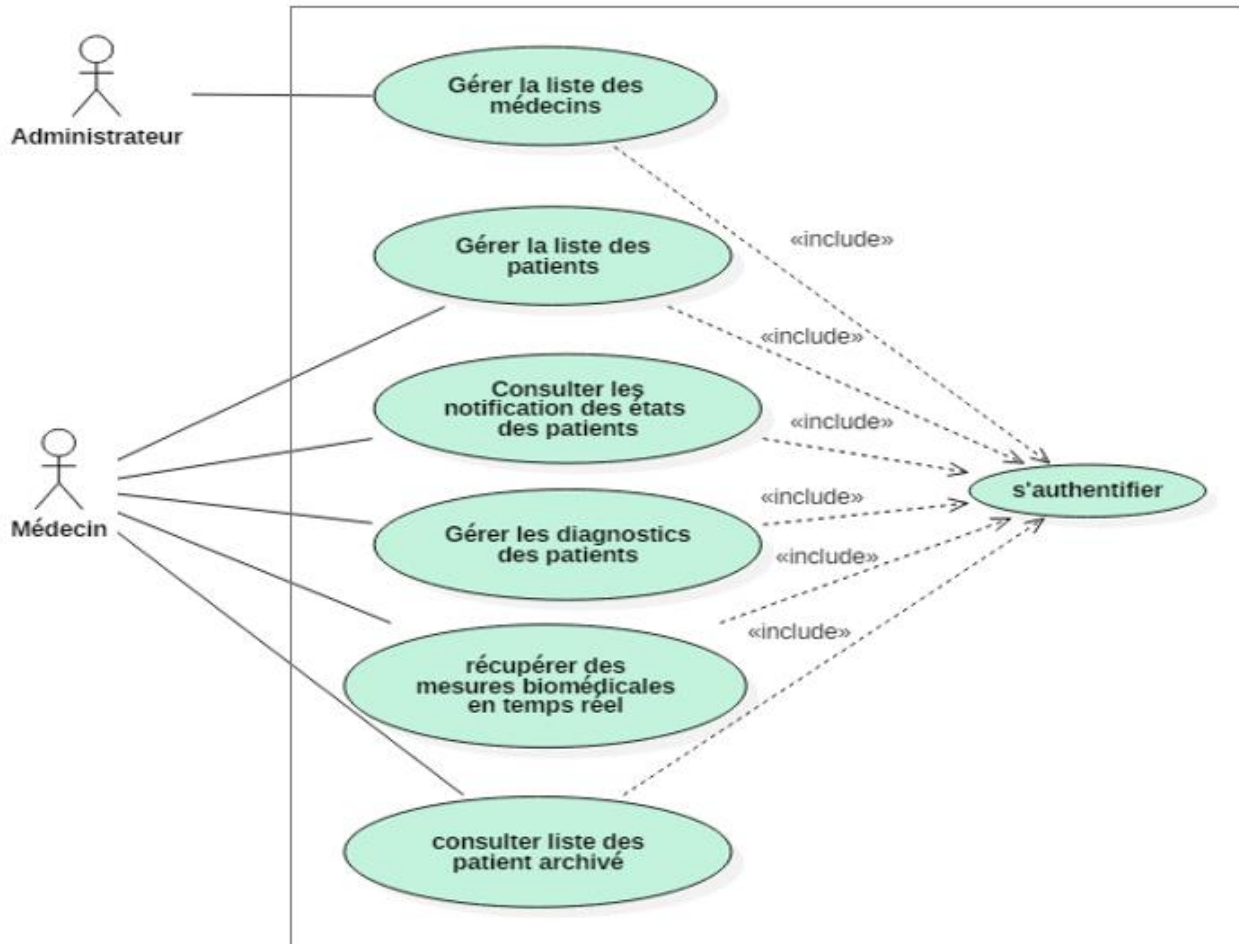


Figure 2.6 : Diagramme de cas d'utilisation global

2.4 Pratique de principes Scrum

2.4.1 Établissement de la « Definition of Done »

La définition de Fini (Definition of Done) Celle-ci est utilisée pour évaluer si le travail est terminé dans un incrément produit.

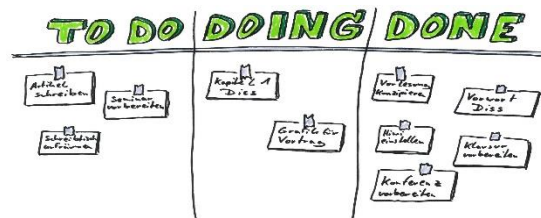


Figure 2.7 : Definition of done

2.4.2 Planification des sprints

En se basant sur notre product backlog et en décomposant notre produit en microservices, notre travail sera planifié sur 4 sprints que chacun dure un mois puisque notre période de stage est limité à quatre mois. Ces 4 sprints vont constituer la première release comme il est montré dans le table 2.2 :

Tableau 2.2 : Planification des sprints

Release	Sprint	Objectif du sprint	Période
Release1	Sprint 1	Connexion et gestion la liste des médecins	01/08 – 30/08
	Sprint 2	Gestion de la liste des patients et la liste des diagnostiques .	02/09 – 30/09
	Sprint 3	-Afficher statistiques sur les états des patients. -Acquisition de mesure biomédicale en temps réel.	1/10 – 31/10
	Sprint 4	Sécurité de la base de donnée	1/11 – 23/11

2.5 Environnement matériel :

2.5.1 kit médical MySignal :

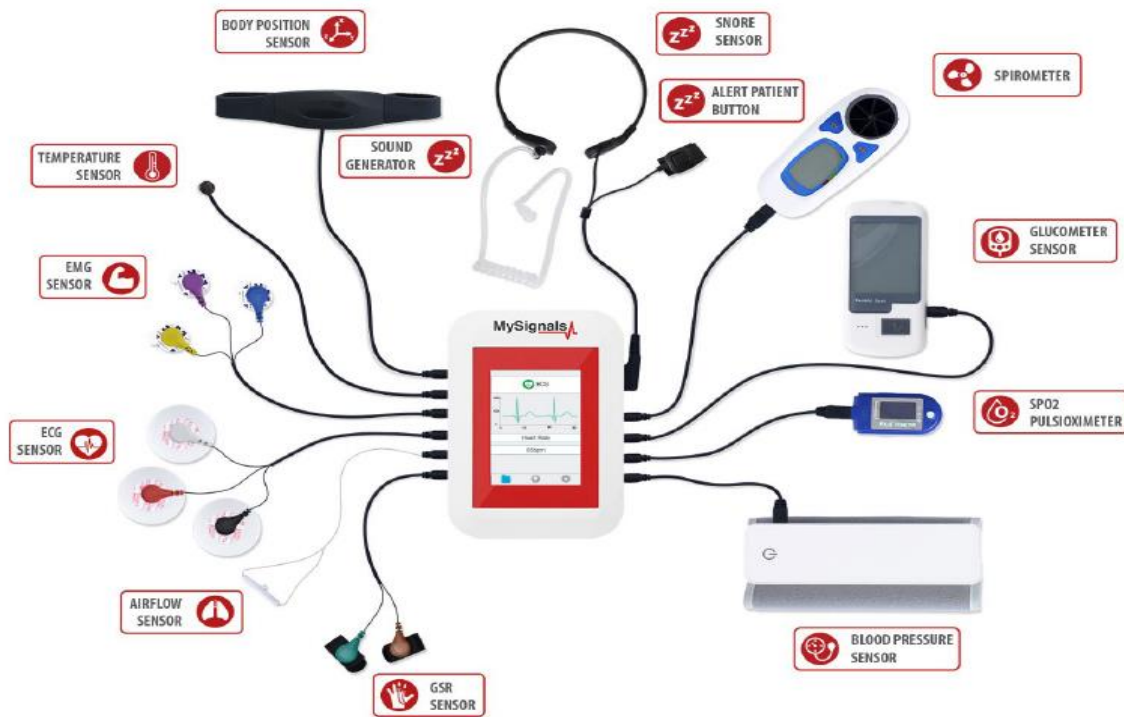


Figure 2.8 : Kit médical MySignal

▪ Pulse and Oxygen in Blood (SPO2)

L'oxymétrie permet de mesurer de façon simple, fiable et continue la saturation artérielle de l'hémoglobine.



Figure 2.9: Capteur SPO2

- **ECG**

L'électrocardiographie (ECG) est une représentation graphique de l'activité électrique du cœur.



Figure 2.10 : Capteur ECG

- **Airflow**

Le capteur de débit d'air nasal / buccal est un appareil utilisé pour mesurer le rythme respiratoire d'un patient

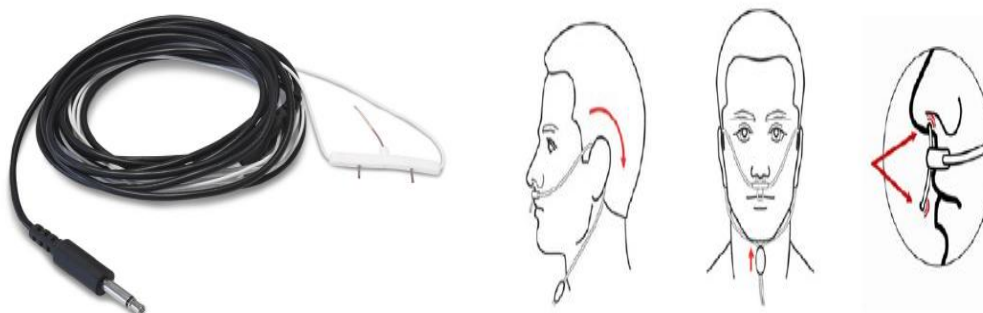


Figure 2.11 : Capteur Airflow

▪ Blood Pressure Monitor

La pression artérielle est la pression du sang dans les artères lorsque celui-ci est pompé dans le corps par le cœur.

La pression artérielle est enregistrée sous forme de deux chiffres :

- La pression systolique (lorsque le cœur bat)
- La pression diastolique (lorsque le cœur se détend entre les battements).



Figure 2.12 : Capteur Blood pressure

▪ Glucometer

Un glucomètre est un appareil permettant de mesurer le taux de glucose dans le sang (la glycémie), le plus souvent sur du sang capillaire.



Figure 2.13 : Capteur Glucometer

- **Body Temperature**

La température corporelle dépend de l'endroit dans le corps où la mesure est effectuée et de l'heure de la journée et le niveau d'activité de la personne.



Figure 2.14 : Capteur Température

- **EMG**

L'électromyogramme ou EMG, a pour objectif d'analyser l'influx nerveux dans les nerfs moteurs, les nerfs sensitifs et les muscles. Il permet d'évaluer le fonctionnement des nerfs et des muscles.

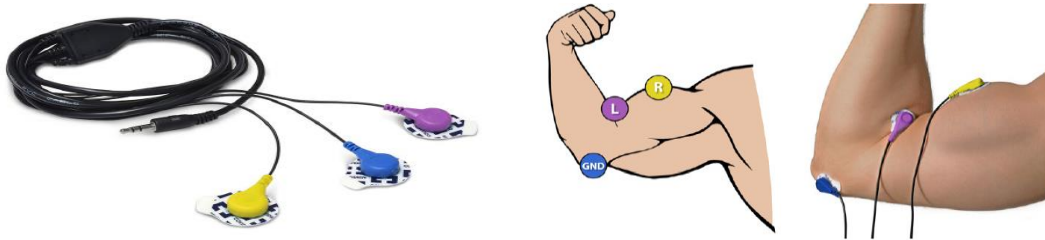


Figure 2.15 : Capteur EMG

- **Spiromètre**

Permet de mesurer la fréquence de la respiration



Figure 2.16 : Capteur Spiromètre

- **GSR**

La conductance est utilisée comme une indication de l'excitation psychologique ou physiologique. Le capteur de réponse cutanée galvanique (GSR) mesure la conductance électrique entre 2 points et est essentiellement un type d'ohmmètre.



Figure 2.17 : Capteur GSR

- **Body Position**

Le capteur de position du patient (accéléromètre) surveille cinq positions différentes du patient (debout /assis, couché sur le dos, couché sur le ventre, à gauche et à droite.).

Le capteur de position corporelle pourrait aider à détecter l'évanouissement ou la chute de personnes âgées ou présentant une handicapée.



Figure 2.18 : Capteur Body position

▪ Sonre

Ce capteur maintient un contact constant avec la peau du patient tout au long de ses mouvements ou de sa transpiration, de sorte qu'on reçoit systématiquement des signaux.



Figure 2.19 : Capteur sonore

Carte Arduino uno

La carte Arduino Uno est basée sur un ATmega328 cadencé à 16 MHz. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires.

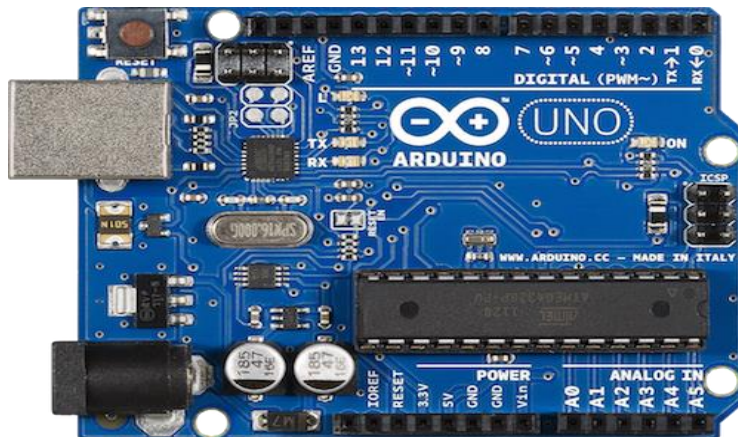


Figure 2.20 : Carte Arduino

Carte MySignal

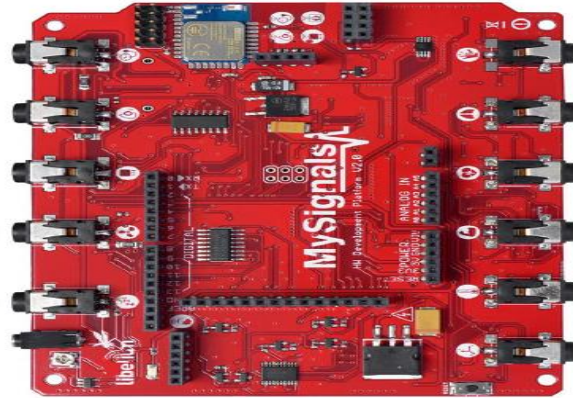


Figure 2.21 : Carte MySignals

2.5.2 Robot de téléprésence « COVEA »

Un robot de téléprésence mobile permet à un **utilisateur à distance** (parfois appelé **pilote**) d'interagir de manière réaliste avec d'autres personnes regroupées dans un lieu précis. À partir de son application de commande, un utilisateur distant peut à la fois communiquer de manière vidéo et contrôler les déplacements du robot dans l'environnement où ce dernier se trouve. Cette technologie offre à l'utilisateur une mobilité, une autonomie et une capacité d'interagir qui résultent en une expérience de téléprésence de grande qualité. Cela accentue l'impression de présence et favorise des échanges gratifiants entre les intervenants impliqués dans les multiples contextes pédagogiques envisageables.

- **Fonctionnement du robot**

À l'aide de l'application mobile téléchargée sur son tablette, l'utilisateur se connecte à distance au robot de téléprésence mobile qui lui, est connecté au réseau internet sans fil. Sa vidéo et sa voix sont alors diffusées par l'intermédiaire du robot. L'utilisateur peut également déplacer le robot de téléprésence mobile dans différentes directions de l'espace à l'aide des flèches de son clavier. Au besoin, un **accompagnateur** sur place assiste les opérations de l'utilisateur-pilote.



Figure 2.22 : Fonctionnement du robot

Le robot de téléprésence mobile dispose de deux caméras vidéo. La première caméra, située directement au-dessus de l'écran, permet à l'utilisateur de prendre connaissance du lieu où interagit le robot (champ de vision de 105 degrés).

La seconde caméra est située sous l'écran et pointe vers la base du robot. Elle permet de prendre connaissance des obstacles potentiels et d'évaluer les différents déplacements effectués. L'interface de pilotage permet également de visualiser la vidéo diffusée sur l'écran du robot à l'intention de vos interlocuteurs.

Pour différentes raisons (perte de contact dans certaines zones sans Wifi, obstacles au sol, mauvaise connaissance des nouveaux patients du mode d'utilisation de certains capteurs), le robot « COVEA » ne peut être utilisé en autonomie complète par un utilisateur-pilote. Un **accompagnateur** (infirmier par exemple) doit veiller au bon déroulement de l'utilisation du robot et dépanner son utilisateur s'il y a lieu. Exemples d'actions d'accompagnement : déplacer le robot dans les chambres d'un hôpital, le placer près d'un patient et effectuer une mesure de certaine attribut médicale.

2.6 Environnement logiciel

Html

L'HyperText Markup Language, est le langage de balisage conçu pour représenter les pages web. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages.

Jira Software

Jira Software est conçu de sorte que chaque membre d'une équipe de développement logiciel puisse planifier, suivre et livrer d'excellents logiciels.

Css

Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit en HTML ou en XML . CSS décrit la façon dont les éléments doivent être affichés

Javascript

JavaScript (JS) est un langage de programmation dynamique complet qui, appliqué à un document HTML, peut fournir une interactivité dynamique sur les sites Web.

PhoneGap/Cordova

PhoneGap/Cordova est un outil permettant de créer des applications mobiles multiplateformes en utilisant les standards du web.

Node.js

Node.js est une plate-forme de développement open source permettant d'exécuter du code JavaScript côté serveur. Node est utile pour développer des applications nécessitant une connexion persistante du navigateur au serveur et est souvent utilisé pour des applications en temps réel.

Python

Python est un langage de programmation populaire. Il est utilisé pour le développement Web (côté serveur), le développement de logiciels, les mathématiques et les scripts système.

MongoDB

MongoDB est un système de gestion de bases de données NoSQL, orienté document classé parmi les « leader » de systèmes de gestion de base de données. Les données sont stockées dans des collections, qui sont l'équivalent des tables dans une base relationnelle.

Visual Studio

Visual Studio Code est un éditeur de code open source et gratuit, qui peut supporter plus qu'une dizaine de langages.

Robot 3T

L'outil Robo 3T (anciennement Robomongo) est une interface utilisateur de bureau populaire pour les déploiement d'hébergement MongoDB qui permet d'interagir avec les données par des indicateurs visuels.

POSTMAN

Une extension sur google qui nous aide à tester les requêtes http rapidement et de consommer des web services afin de mieux comprendre la structure de données reçues.

Conclusion

Dans ce chapitre nous avons présenté le Product Backlog, le diagramme des cas d'utilisation global ainsi que l'environnement matériel et logiciel que nous préparons pour la mise en place de notre solution. Nous avons aussi présenté l'architecture physique et logique de notre application. Le chapitre suivant est consacré aux travaux effectués durant le premier sprint.

Chapitre 3

Sprint 1 : Connexion, Gestion des médecins

Sommaire

1. Ehealth Sprint 1 planning	36
2. Conception	38
3. Réalisation	43
4. Test et validation	45

Introduction

Après avoir analysé et spécifié les besoins globaux de notre client, nous commençons la réalisation de sprint 1. Nous avons organisé le travail en trois phases principales qui sont la conception, la réalisation et enfin les tests.

3.1 Ehealth Sprint 1 planning

3.1.1 Objectif du sprint 1

Ce Sprint a pour objectif le développement de l'interface de connexion à l'application c'est-à-dire seulement les utilisateurs autorisés peuvent connecter et accéder à l'application. La deuxième partie de ce sprint est consacré au développement de la partie backend de l'administrateur qui permet la gestion des médecins c'est-à-dire que l'administrateur peut ajouter et supprimer un médecin. Il peut aussi modifier les informations d'un médecin.

3.1.2 Sprint Backlog du Sprint 1

Comme je suis la « Dev Team » de ce projet, alors j'ai choisi des « user stories » à partir de notre Product Backlog pour construire le premier sprint Backlog. J'ai décomposé les « user stories » en « tasks » à accomplir avant la fin du sprint pour arriver à un incrément fini et livrable.

Chaque task est caractérisé par un degré de complexité pour l'effectuer. En scrum on a différentes techniques d'estimation de complexité tel que : « T-Shirts Sizes », « Dot voting », « Planning Pocker ».

J'ai choisi à utiliser la méthode « T-Shirts Sizes » Comme technique de priorisation.

Tableau 3.1 : Sprint Backlog

ID	User Story	ID Task	Task	Complexity
1	En tant qu'utilisateur (médecin ou administrateur) je veux connecter mon application afin d'accéder à mon espace en toute sécurité	1.1	Préparer le design de l'interface de connexion	S
		1.2	Implémentation du code nécessaire à la connexion	M
2	En tant qu'administrateur je veux gérer la liste des médecins	2.1	Préparer le design de l'espace administrateur.	S
		2.2	Création des interfaces graphiques.	S
		2.3	Préparation et test des requêtes permettant l'ajout et la récupération d'un médecin.	L
		2.4	Préparation et test des requêtes permettant la modification des données d'un médecin.	L
		2.5	Préparation et test des requêtes permettant la suppression d'un médecin.	L

3.2 Conception

Dans cette partie, nous présentons le diagramme de cas d'utilisation du premier sprint, les diagrammes de séquences et le diagramme de classe.

3.2.1 Diagramme de cas d'utilisation du premier sprint

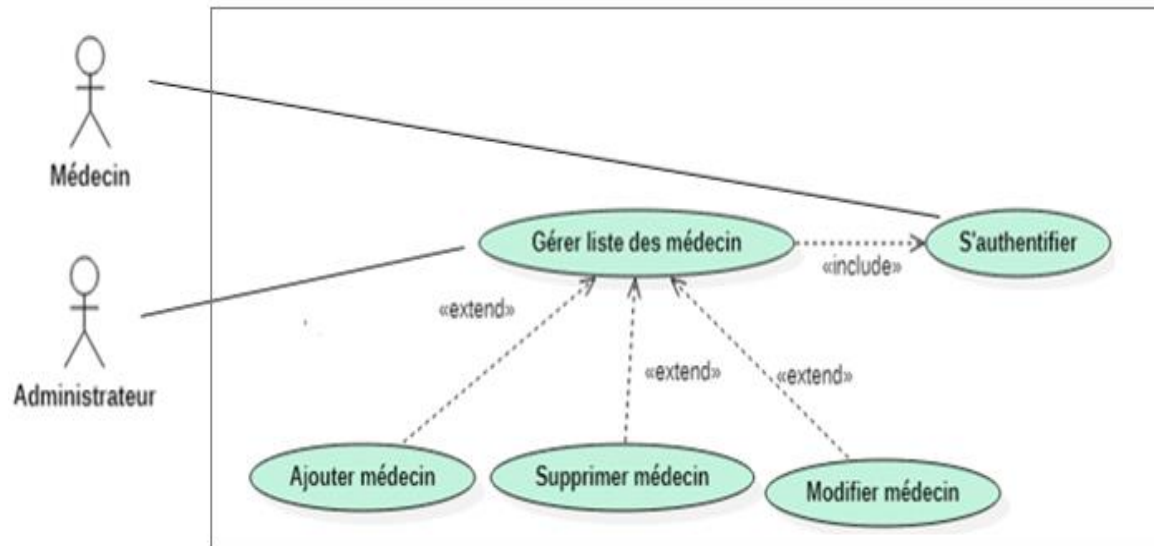


Figure 3.1 : Diagramme de cas d'utilisation du premier sprint

Tableau 3.2: Description textuelle du cas d'utilisation « Authentification »

Titre : S'authentifier
Acteur : Administrateur ou Médecin
Scénario nominal : <ol style="list-style-type: none"> 1. L'utilisateur demande l'interface d'authentification. 2. L'utilisateur saisit son login et le mot de passe puis clique sur le bouton « Login ». 3. Le système vérifie l'existence des données saisies dans la base de donnée. 4. Le système retourne la page d'accueil.

<p>Scénario alternatif :</p> <p>Les données saisis sont incorrecte L'enchaînement démarre du point 3 du scénario nominal :</p> <ul style="list-style-type: none"> — Le système retourne la page d'authentification, affiche un message d'erreur dans lequel il indique que l'un des champs est incorrecte ou vide. — La séquence nominale démarre du point 2.
<p>Post-condition : Utilisateur authentifié</p>

Tableau 3.3: Description textuelle du cas d'utilisation « Ajout médecin »

<p>Titre : Ajouter un médecin</p>
<p>Acteur : Administrateur</p>
<p>Pré-condition : L'administrateur du système doit s'authentifier avant d'effectuer cette tâche.</p>
<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1. L'administrateur demande l'interface d'ajout en cliquant sur le bouton « Add doctor ». 2. L'administrateur remplit le formulaire d'ajout et valide son choix par cliquer sur le bouton « Ajouter ». 3. Le système vérifie les données saisies. 4. Le système enregistre le médecin ajouté. 5. Le système retourne la nouvelle liste mise à jour.
<p>Scénario alternatif :</p> <p>L'un des champs est vide, l'enchaînement démarre du point 3 du scénario nominal :</p> <ul style="list-style-type: none"> — Le système affiche un message d'erreur et indique que l'un des champs est vide. — La séquence nominale démarre du point 2.
<p>Post-condition : Médecin ajouté</p>

Tableau 3.4: Description textuelle du cas d'utilisation « Suppression médecin »

Titre : Supprimer un médecin
Acteur : Administrateur
Pré-condition : L'administrateur du système doit s'authentifier avant d'effectuer cette tâche.
Scénario nominal : <ol style="list-style-type: none"> 1. L'administrateur accède à la liste de gestion des médecins. 2. Le système affiche la liste de tous les médecins. 3. L'administrateur choisit un médecin à supprimer. 4. Le système retourne la nouvelle liste des médecins.

3.2.2 Diagrammes de séquences

3.2.2.1 Scénario « Authentification »

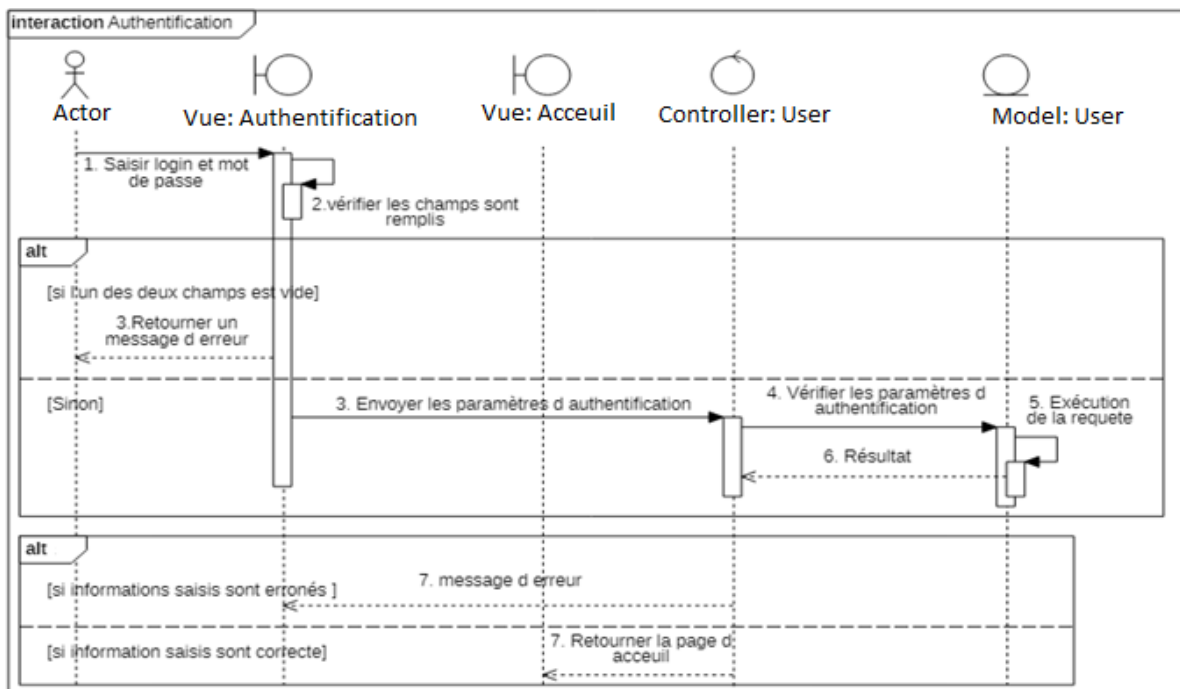


Figure 3.2 : Diagramme de séquence du cas d'utilisation « Authentification »

3.2.2.2 Scénario « Ajout du médecin »

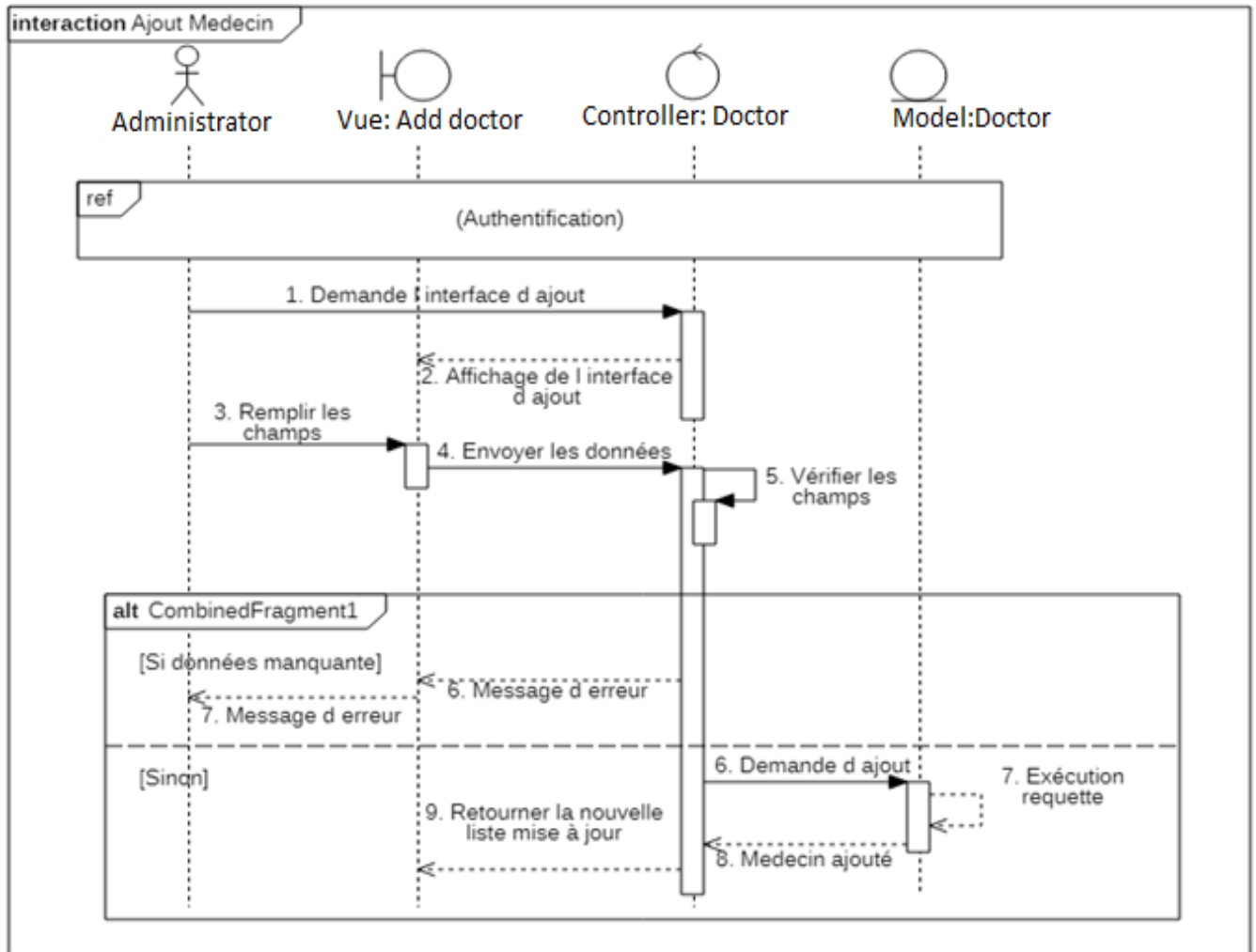


Figure 3.3 : Diagramme de séquence du cas d'utilisation « Ajout médecin »

3.2.2.3 Scénario « Suppression médecin »

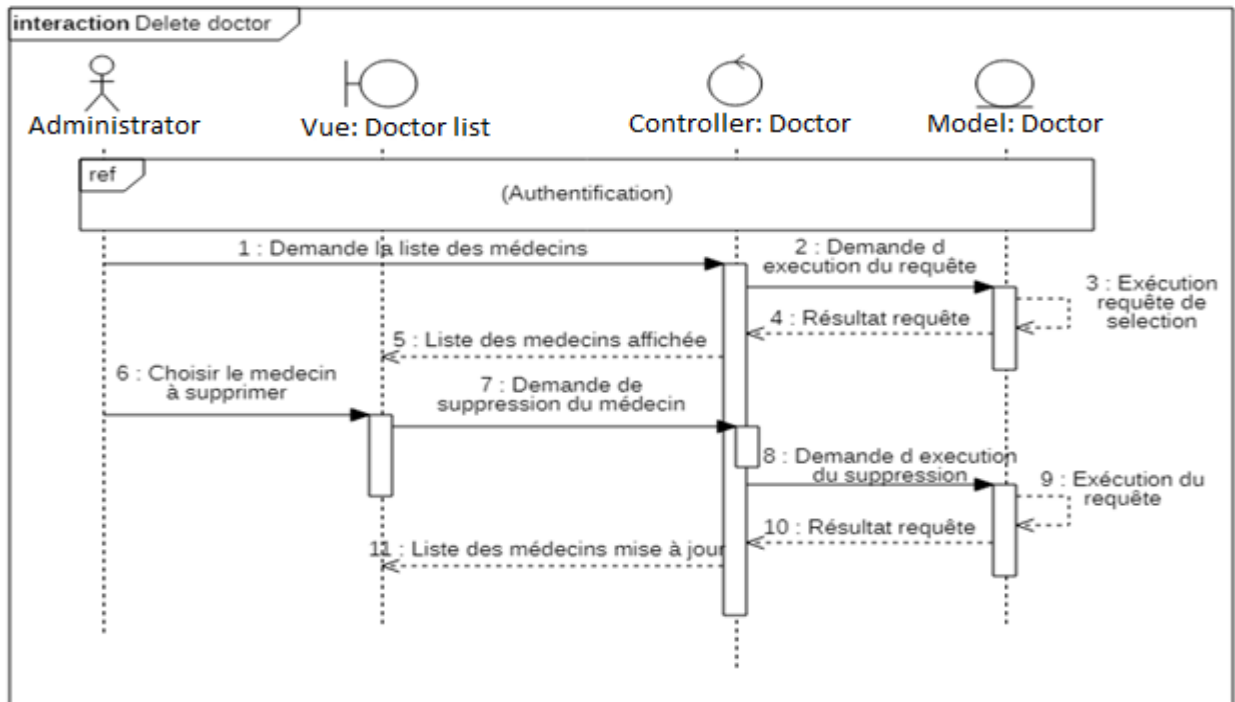


Figure 3.4 : Diagramme de séquence du cas d'utilisation « suppression médecin »

3.2.3 Diagramme de classe

Le diagramme de classe est une représentation statique des éléments (attributs et méthodes) qui est composé d'un système et de leurs relations. La figure 3.5 décrit le diagramme de classes relatif au développement du premier sprint.

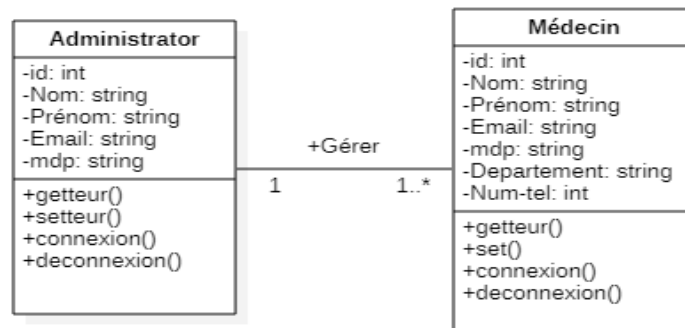


Figure 3.5 : Diagramme de classe du premier sprint

3.3 Réalisation

Cette partie est consacrée à la présentation du travail achevé à travers des captures d'écran des différentes interfaces développées durant le premier sprint :

- **User story 1 : Authentification**

Afin d'accéder à l'application, chaque utilisateur doit s'authentifier en fournissant son login et son mot de passe à travers l'interface de la figure 3.6

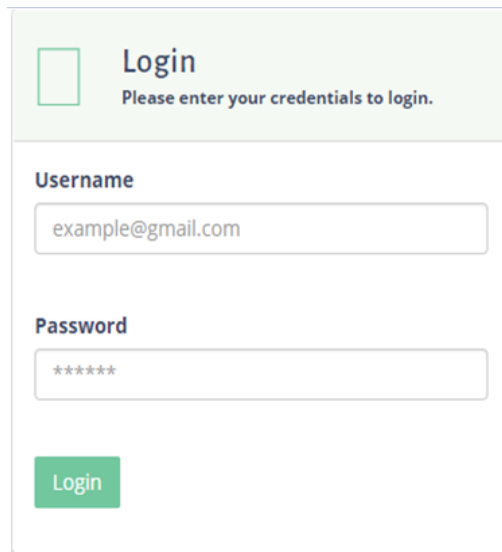


Figure 3.6 : Interface du « Login »

- **User story 2 : Gestion Médecin**

La figure 3.7 présente l'espace admin qui contient la liste des médecins à travers laquelle l'administrateur peut ajouter, supprimer ou modifier un médecin.

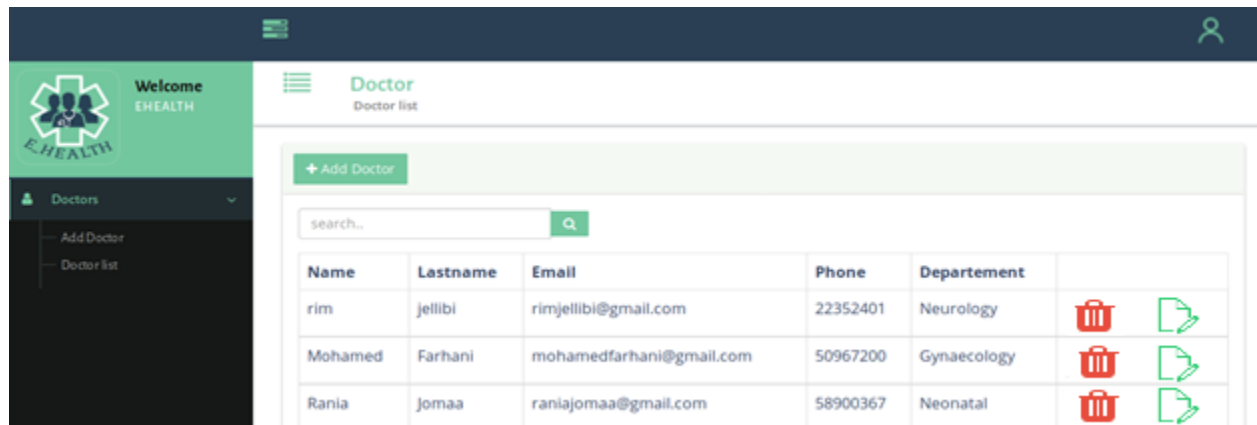


Figure 3.7 : Interface « Liste des médecins »

En consultant cette liste l'administrateur peut soit supprimer ou modifier un médecin

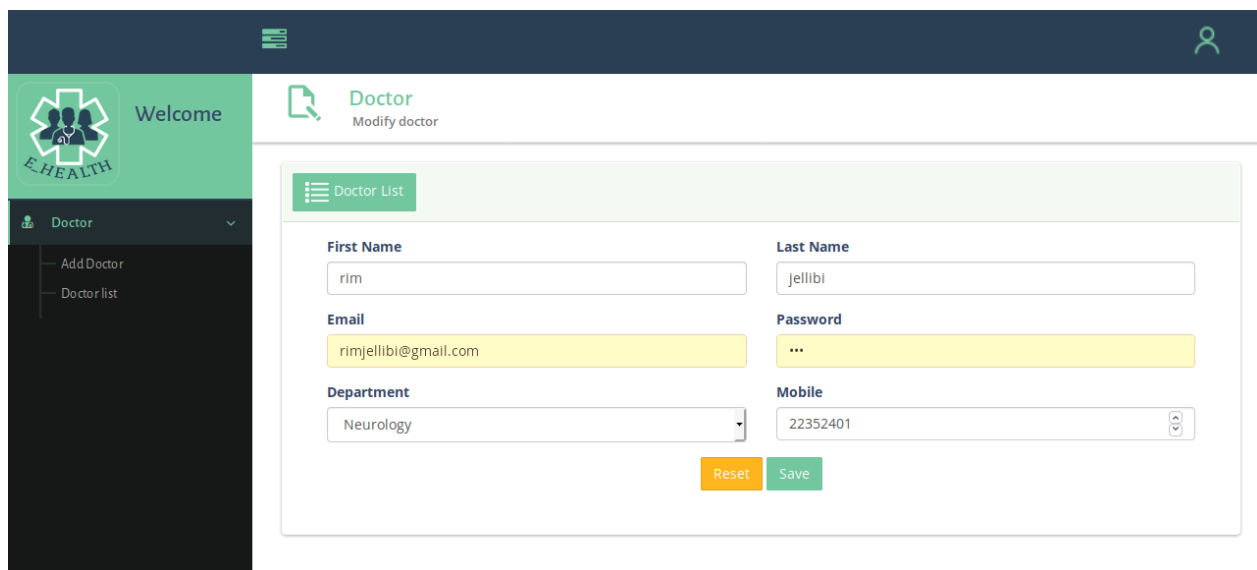


Figure 3.8 : Interface « Modifier médecin »

L'administrateur peut aussi ajouter un nouveau médecin, en cliquant sur le bouton « Add Doctor ». Un formulaire vide s'affiche dans lequel l'administrateur peut saisir les informations nécessaires. La figure 3.9 présente l'interface d'ajout d'un médecin.

Figure 3.9 : Interface « Ajout médecin »

3.4 Test et validation

L'extension Postman nous a donné la main pour tester les web services. J'ai utilisé POST comme type de service pour tester la méthode « Ajout médecin ».

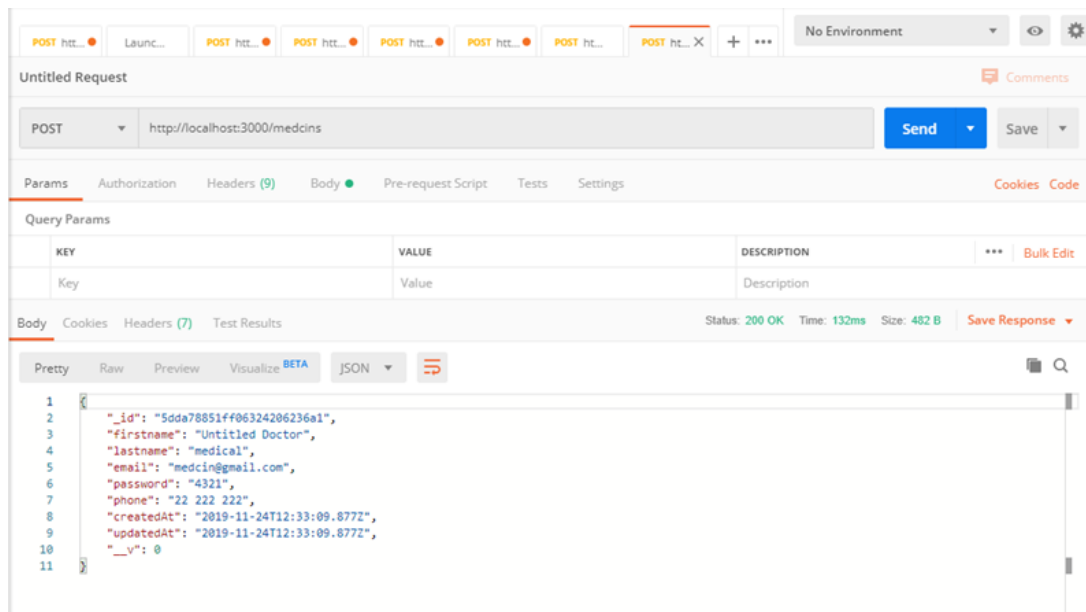


Figure 3.10 : Test des services web

Pour vérifier que les données sont bien insérées dans la base de données on utilise le Robo 3T.

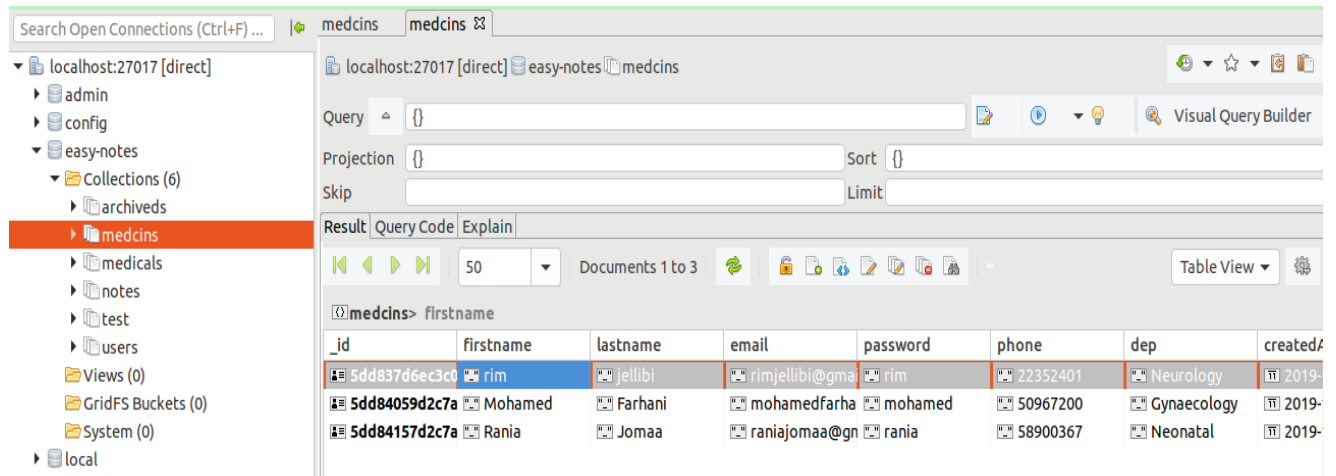


Figure 3.11 : Affichage du contenu de la collection « médecins » dans la base de données

Conclusion

Dans ce sprint nous avons commencé par un planning. Ensuite, nous avons détaillé la conception en présentant le diagramme du cas d'utilisation, les diagrammes de séquence et le diagramme de classe. Enfin, nous avons présenté les interfaces de l'application réalisées au cours de ce sprint.

A ce niveau la partie backend de la gestion des médecins et l'interface de connexion ont été développés.

Chapitre 4

Sprint 2 : Gestion des patients, Gestion des diagnostics

Sommaire

1. Ehealth Sprint 2 planning	48
2. Conception	50
3. Réalisation	52
4. Test et validation	54

Introduction

Ce chapitre représente le deuxième sprint qui focalise sur la gestion des patients et la gestion des diagnostics. Nous expliquons dans ce chapitre le backlog du sprint, la conception ainsi que la réalisation et nous terminons par les tests fonctionnels pour vérifier le bon fonctionnement de l'incrément qu'on va le livrer après la fin de ce sprint.

4.1 Ehealth Sprint 2 planning

4.1.1 Objectif du sprint 2

Ce Sprint a pour objectif, de développer la partie de gestion des patients c'est-à-dire que le médecin peut ajouter, supprimer un patient ou modifier ses informations. Il peut aussi consulter les valeurs récupérées par le kit médical. La deuxième partie de ce sprint est consacré à développer la partie de gestion des diagnostics pour l'application.

4.1.2 Sprint Backlog du Sprint 2

Tableau 4.1 : Sprint backlog du sprint 2

ID	User Story	ID Task	Task	Complexity
1	En tant que médecin je veux gérer la liste des patients	1.1	Préparer le design de la liste des patient et celui des interfaces de gestion des patients.	S
		1.2	Création des interfaces graphiques.	S
		1.3	Préparation et test des requêtes permettant l'ajout et la récupération d'un patient.	L

		1.4	Préparation et test des requêtes permettant la suppression d'un patient.	L
		1.5	Préparation des requêtes permettant la modification d'un patient.	L
2	En tant que médecin je veux gérer les diagnostics des patients	2.1	Préparer le design de l'interface de l'ajout d'un patient	S
		2.2	Préparation et test des requêtes permettant l'ajout et la récupération d'un diagnostics pour un patient bien précis.	L
		2.3	Préparation et test des requêtes permettant la suppression d'un diagnostic.	L
		2.4	Préparation et test des requêtes permettant la modification d'un diagnostic.	L

4.2 Conception

4.2.1 Diagramme de cas d'utilisation du deuxième sprint

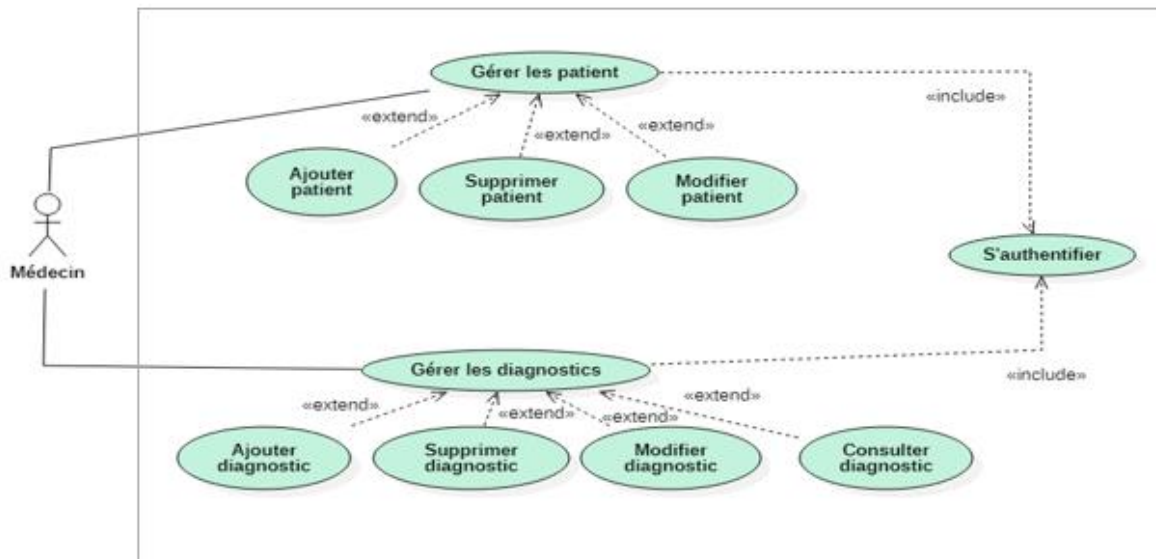


Figure 4.1 : Diagramme de cas d'utilisation du deuxième sprint

Tableau 4.2: Description textuelle du cas d'utilisation « Consulter diagnostic »

Titre : Consulter diagnostic
Acteur : Médecin
Pré-condition : Le médecin doit s'authentifier avant d'effectuer cette tâche.
Scénario nominal : <ol style="list-style-type: none"> 1. Le médecin demande la liste des diagnostics. 2. Le système affiche la liste de tous les diagnostics. 3. Le médecin cherche le patient voulu pour afficher son diagnostic. 4. Le système retourne la liste des diagnostics du patient choisi. 5. Le médecin choisi le diagnostic voulu 6. Le système retourne les détails du diagnostics choisi.
Post-condition : Détails diagnostic Affichés

4.2.2 Diagrammes de séquences

Scénario « consulter diagnostic »

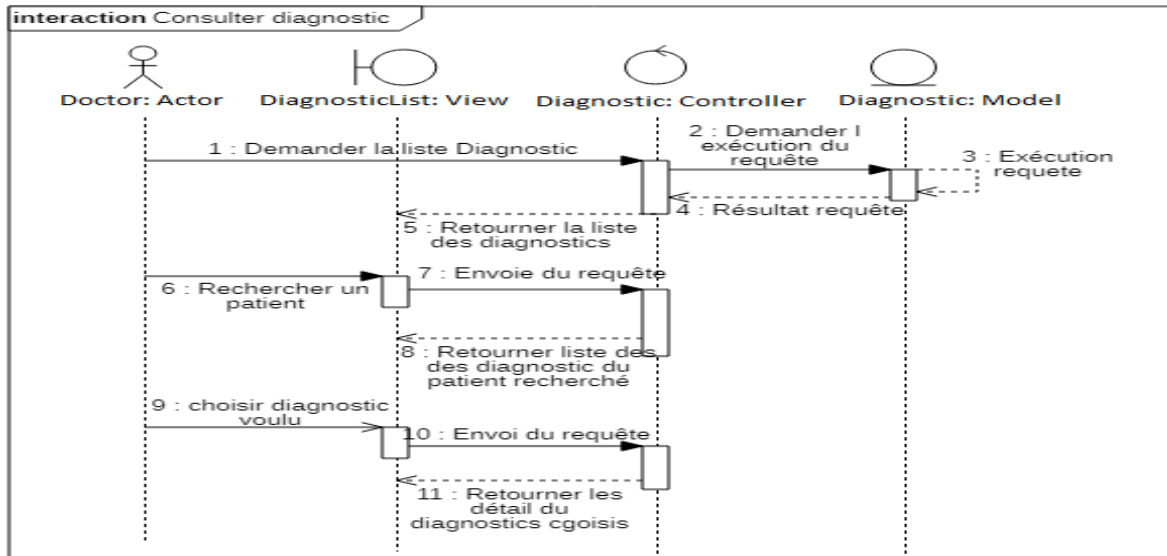


Figure 4.2: Diagramme de séquence « Consulter diagnostic »

4.2.3 Diagramme de classe

Le diagramme de classe présente la structure statique de ce sprint

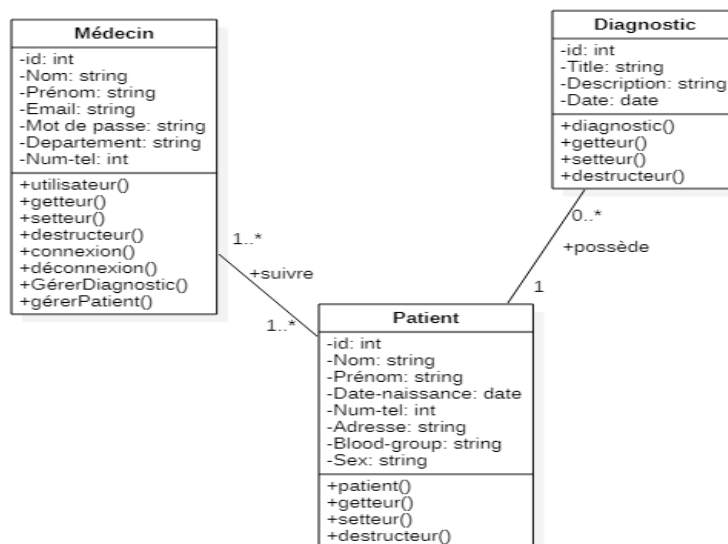


Figure 4.3 : Diagramme de classe du deuxième sprint

4.3 Réalisation

Dans cette partie nous allons présenter le travail réalisé à travers des captures d'écrans des différentes interfaces développées durant le Sprint 2 :

- **User story 1 : Gestion des patients**

Après l'accès à son espace, chaque médecin peut afficher la liste des patients dans laquelle il peut ajouter, supprimer ou modifier un patient. Le médecin a aussi la possibilité de chercher un patient dans la liste affichée.

La figure 4.4 présente l'interface de la liste des patients

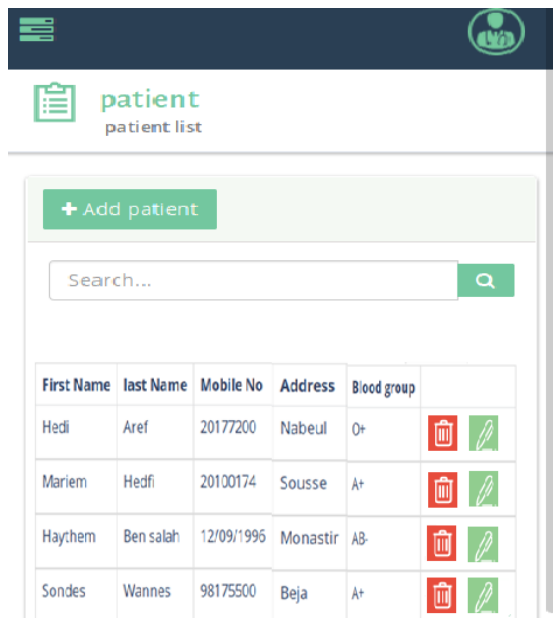


Figure 4.4 : Liste des patients

La figure 4.5 présente le formulaire dans lequel le médecin peut soit saisir les données du patient qu'il veut ajouter et le formulaire dans lequel le médecin peut modifier les données d'un patient.

The figure shows two side-by-side screenshots of a patient management interface. The left screen is titled "Patient add patient" and features a "patient List" button at the top. Below it, there are four input fields: "First Name" (placeholder: Enter First Name), "Last Name" (placeholder: Enter last Name), "Date of Birth" (placeholder: Date of Birth), and "Phone number" (placeholder: Phone number). The right screen is titled "Patient Modify patient" and also has a "patient List" button. It contains the same four input fields, but they are pre-filled with data: "First Name" is "Hedi", "Last Name" is "Aref", "Date of Birth" is "05/08/1998", and "Phone number" is "20177200".

Figure 4.5 : Interface de gestion des patients

▪ **User story 2 : Gestion diagnostics**

La figure 4.6 présente l'espace médecin qui contient la liste des diagnostics à travers laquelle le médecin peut ajouter un nouveau diagnostic pour un patient bien précis, modifier ou supprimer un de la liste.

The figure shows a screenshot of a "Note list" interface. At the top, there is a "Note list" title and a "Note list" subtitle. Below this, there is a green button labeled "+ Add Note". Underneath the button is a search bar with the placeholder text "search.." and a green search icon. Below the search bar is a table with three columns: "Title", "description", and "Date". The table contains two rows of data:

Title	description	Date
diabète type 1	Glycémie: 2.5 g/l	05/07
Hypertension	PS: 140 , PD: 80	03/11

At the bottom of the table, there are pagination controls showing "« 1 2 ... Next".

Figure 4.6 : Liste diagnostics

Le médecin peut ajouter un nouveau diagnostic pour ceci le médecin est invité à saisir : le titre, la description, la date et sélectionne le patient pour lequel il veut ajouter ce diagnostic

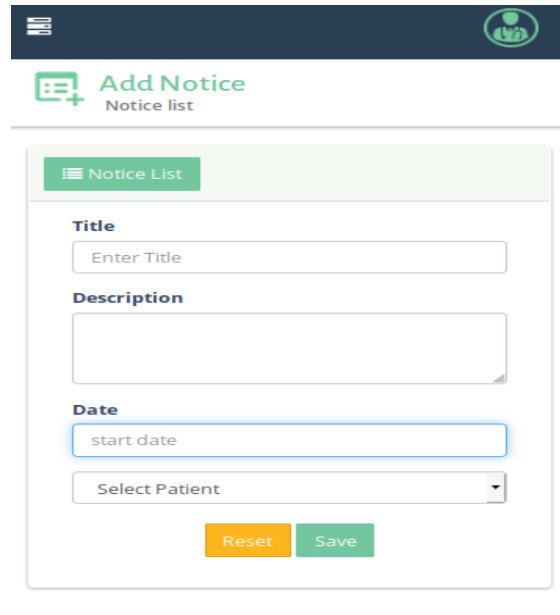
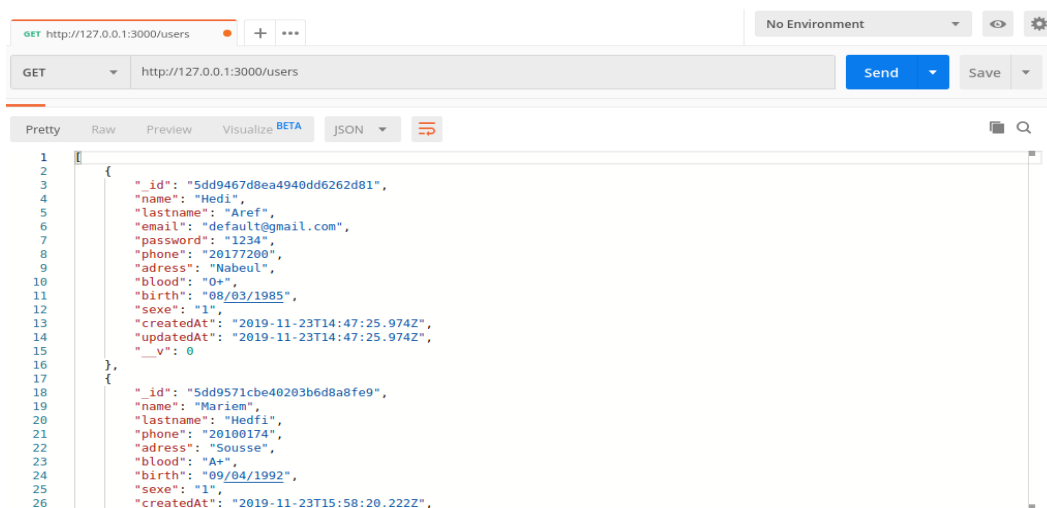


Figure 4.7 : Interface d'ajout de diagnostics

4.4 Test et validation

En utilisant Postman, j'ai testé le web service « Get Patient ». Le résultat est sous forme JSON qui contient le contenu de la liste des patients



```
1 {
2   {
3     "id": "5dd9467d8ea4940dd6262d81",
4     "name": "Hedi",
5     "lastname": "Aref",
6     "email": "default@gmail.com",
7     "password": "1234",
8     "phone": "20177200",
9     "adress": "Nabeul",
10    "blood": "O+",
11    "birth": "08/03/1985",
12    "sexe": "1",
13    "createdAt": "2019-11-23T14:47:25.974Z",
14    "updatedAt": "2019-11-23T14:47:25.974Z",
15    "_v": 0
16  },
17  {
18    "id": "5dd9571cbe40203b6d8a8fe9",
19    "name": "Mariem",
20    "lastname": "Hedfi",
21    "phone": "20100174",
22    "adress": "Sousse",
23    "blood": "A+",
24    "birth": "09/04/1992",
25    "sexe": "1",
26    "createdAt": "2019-11-23T15:58:20.222Z",
```

Figure 4.8 : Test des services web

On vérifie que les données affichés dans le résultat précédant sont les mêmes dans la collection des patient dans la base de données.

	name	lastname	phone	adress	blood
d8ea45	Hedi	Aref	20177200	Nabeul	O+
cbe402	Marlem	Hedfi	20100174	Sousse	A+
7be402	Haythem	Ben salah	12/09/1996	Monastir	AB-
db402	Sondes	Wannes	98175500	Beja	A+
3be402	Heni	Salami	20302320	Monastir	B+

Figure 4.9 : contenu de la collection des patients dans la base de données

De même pour la vérification des diagnostics

	title	content	date	patient
5dd90ece8ea45	diabète type 1	Glycémie: 2.5 g	05/07/2019	Nour
5dd911658ea45	Hypertension	PS: 140, PD: 80	03/11/2019	Maroua

Figure 4.10 : contenu de la collection des diagnostics dans la base de données

Conclusion

Durant ce sprint nous avons présenté l'objectif de ce sprint par la suite nous avons réalisé la conception en présentant le diagramme de cas d'utilisation, le diagramme de classe et les diagrammes de séquence. Enfin, nous avons présenté les interfaces réalisées.

Ce sprint a été exposé dans la réunion « Sprint Review » ce qui nous a permis de réaliser un contrôle de qualité et de préparer pour le sprint suivant qui va être présenté dans le chapitre suivant.

Chapitre 5

Sprint 3 : Suivre états patients, consultation mesures médicales

Sommaire

1. Ehealth Sprint 3 planning	58
2. Conception	60
3. Réalisation	63
4. Test et validation	65
5. Synthèse de la Release	65

Introduction

Durant ce chapitre, nous entamons la réalisation de notre application mobile. Nous traitons, dans un premier temps le backlog du sprint 3. Par la suite, nous abordons la conception et la présentation des interfaces réalisées. Enfin nous clôturons ce chapitre par le déploiement de notre application.

5.1 Ehealth Sprint 3 planning

5.1.1 Objectif du sprint 3

Ce Sprint a pour objectif, de développer la partie qui permet le suivi des états des patients: les patients sont regroupés en 3 catégories (Patient à état stable, Patient à état anormal, Patient à état critique). La deuxième partie de ce sprint se charge de développer l'interface qui permet l'affichage des valeurs récupérées des capteurs. la dernière partie est consacré à réaliser une archive pour stocker la liste des patients supprimés.

5.1.2 Sprint Backlog du Sprint 3

Tableau 5.1 : Sprint Backlog du Sprint 3

ID	User Story	ID Task	Task	Complexity
1	En tant que médecin je veux consulter les états des patients.	1.1	Préparer le design de l'interface de consultation des états des patients.	S
		1.2	Création des interfaces graphiques.	S
		1.3	Préparation et test des requêtes de classification des patients selon leurs états.	L

2	En tant que médecin je veux récupérer les mesures biomédicales pour chaque patient.	2.1	Préparer le design de l'interface qui affiche les mesures récupérés par les capteurs.	S
		2.2	Création des interfaces graphiques.	S
		2.3	Préparation et test des requêtes de récupération des mesures.	L
3	En tant que médecin je veux maintenir les patients supprimés dans une archive	3.1	Préparer le design de l'interface qui affiche la liste des patients supprimés.	S
		3.2	Création des interfaces graphiques.	S
		3.3	Préparation et test des requêtes de récupération des patients supprimés.	M

5.2 Conception

5.2.1 Diagramme de cas d'utilisation du troisième sprint

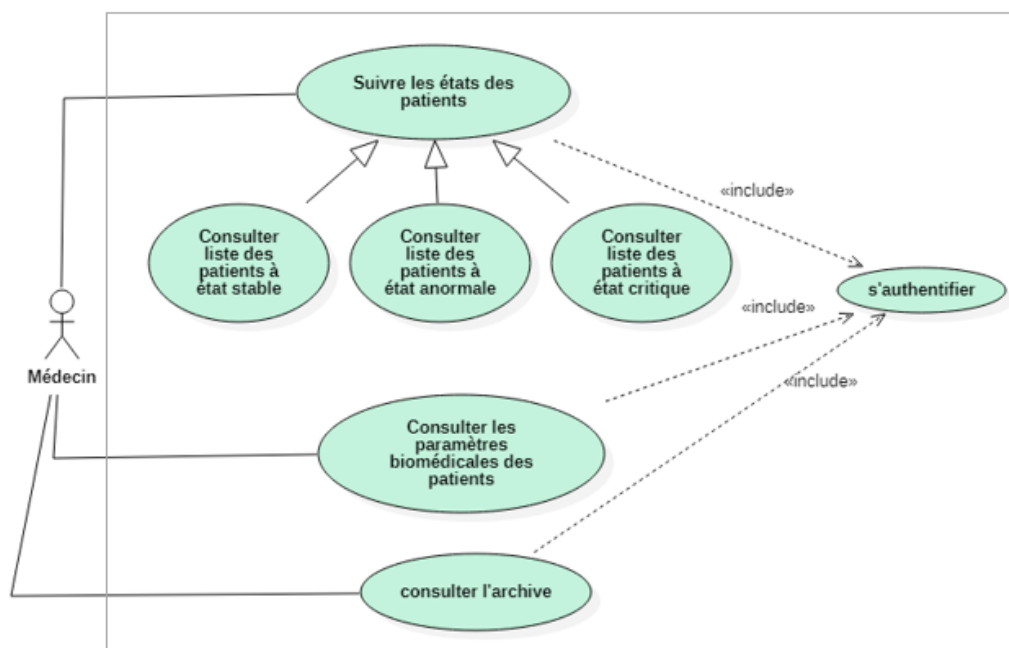


Figure 5.1 : Diagramme du cas d'utilisation du troisième sprint

Tableau 5.2: Description textuelle du cas d'utilisation « Suivre les états des patients »

Titre : Suivre les états des patients
Acteur : Médecin
Pré-condition : Le médecin du système doit s'authentifier avant d'effectuer cette tâche.
Scénario nominal : <ol style="list-style-type: none"> 1. Le médecin accède à l'interface des états des patients. 2. Le médecin choisit la catégorie d'état qu'il veut consulter (Stable, anormale ou critique). 3. Le système retourne la liste de tous les patients appartenant à l'état choisi .
Post-condition : Etat patient défini

Tableau 5.3: Description textuelle du cas d'utilisation « consulter les mesures biomédicales »

Titre : Afficher les mesures biomédicales d'un patient
Acteur : Médecin
Pré-condition : Le médecin du système doit s'authentifier avant d'effectuer cette tâche.
Scénario nominal : 1. Le médecin accède à la liste des patients. 2. Il choisit le patient auquel il veut consulter ses mesures. 3. Le système affiche les valeurs médicales du patient choisi
Post-condition : valeur médicale affichée

5.2.2 Diagrammes de séquences

5.2.2.1 Scénario 1 : « Suivre les états des patients »

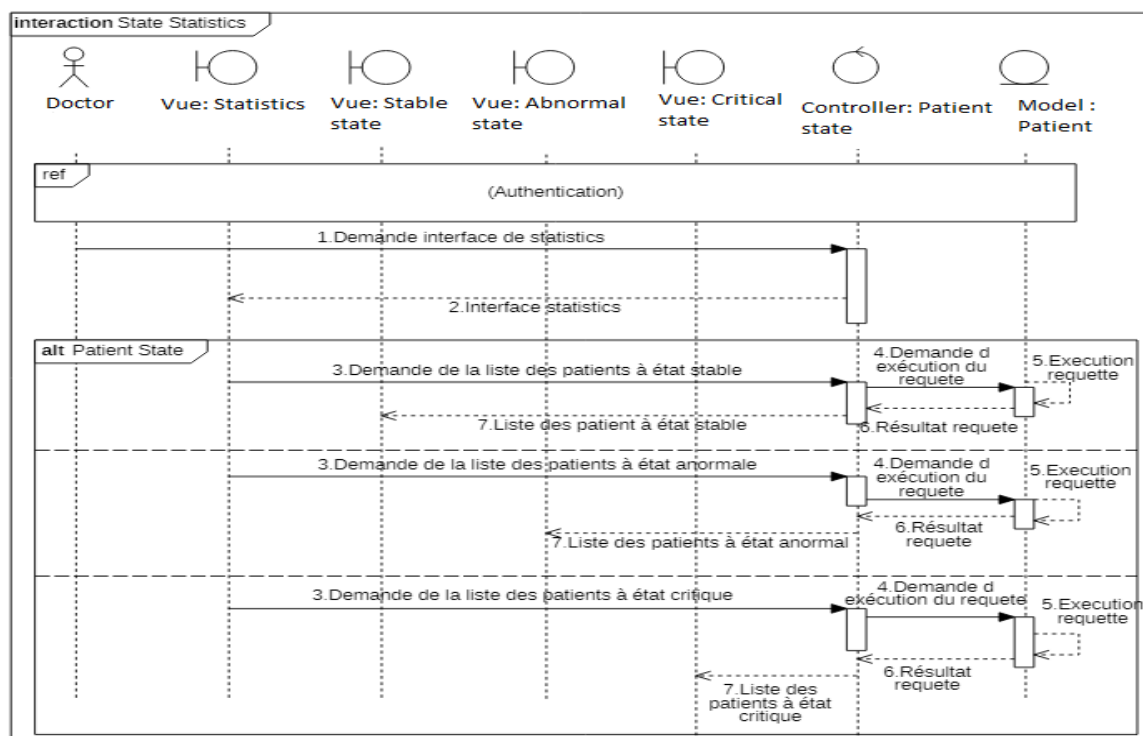


Figure 5.2 : Diagramme de séquence du scénario « Suivre les états des patients »

5.2.2.2 Scénario 2 « consulter les mesures biomédicales des patients »

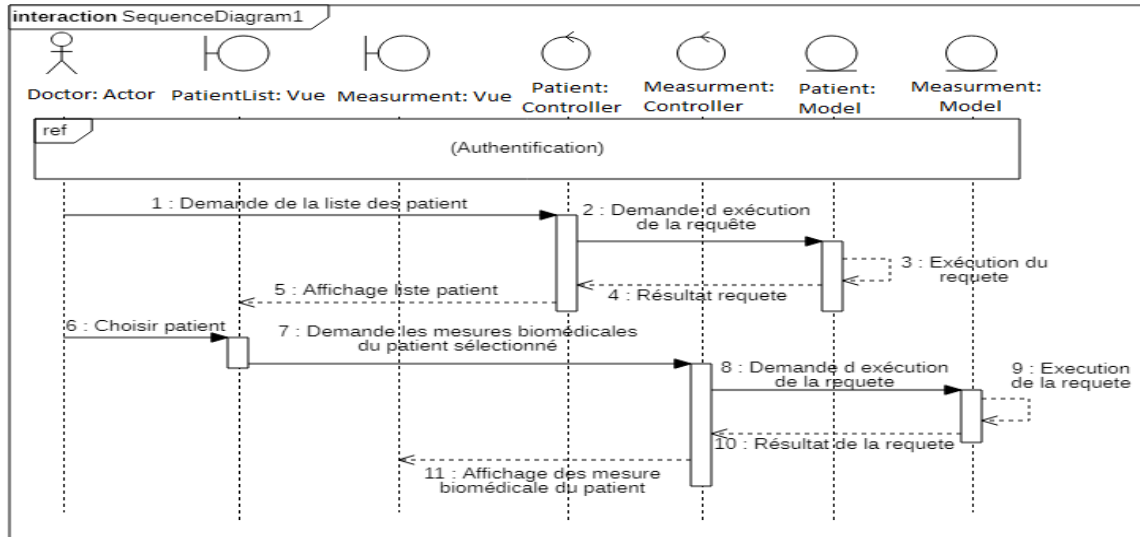


Figure 5.3 : Diagramme de séquence du scénario « consulter les valeurs biomédicales des patients »

5.2.3 Diagramme de classe

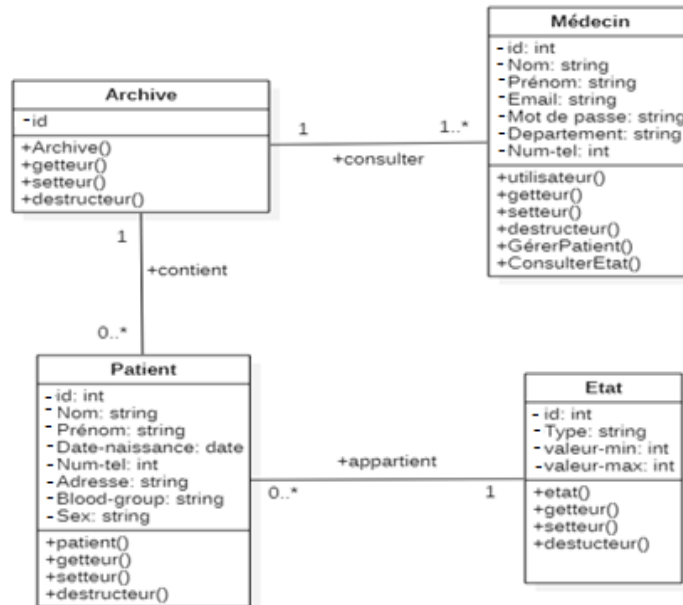


Figure 5.4 : Diagramme de classe du sprint 3

5.3 Réalisation

Dans cette section, nous allons présenter quelques interfaces développées durant le Sprint 3.

- **User story 1 :**

Après l'accès à son espace le médecin peut consulter les états des patients regroupés en trois sections :

- Patients à état stable
- Patients à état anormale
- Patients à état critique

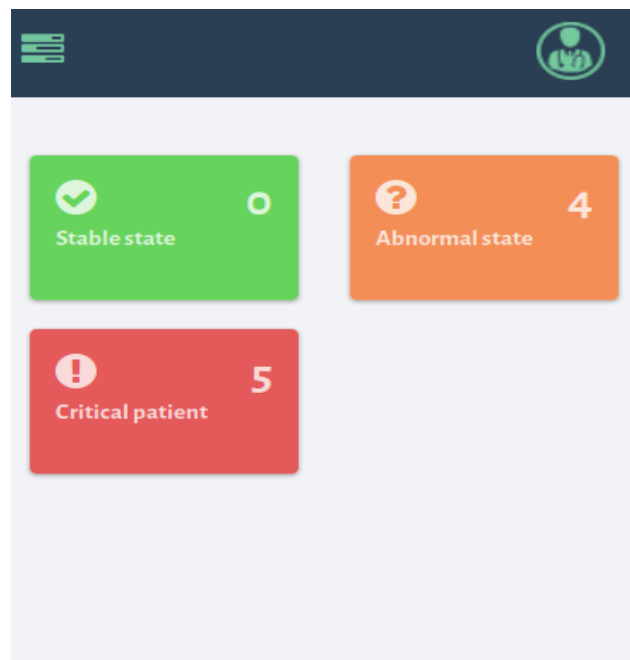


Figure 5.5 : Interface de suivi des états des patients

▪ User story 2 :

La figure 5.6 présente l'interface à travers laquelle le médecin peut consulter les valeurs biomédicales d'un patient.



Figure 5.6 : Interface de récupération de la valeur récupéré par le capteur

▪ User story 2 :

La figure 5.7 présente l'interface à travers laquelle le médecin peut consulter l'archive qui contient la liste des patients supprimés.

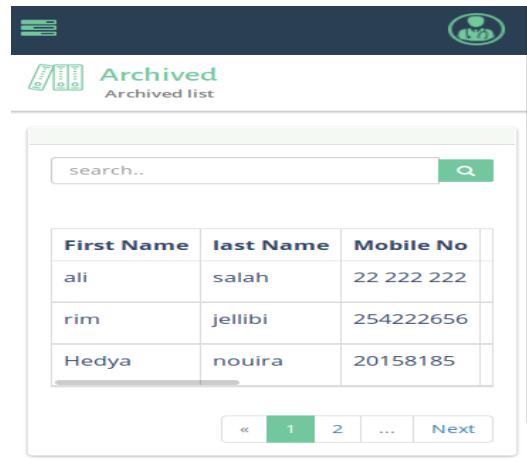


Figure 5.7 : Interface de l'interface de l'archive

5.4 Test et validation

On vérifie l'existence des mesures récupérées, dans la collection « Archive » de notre base

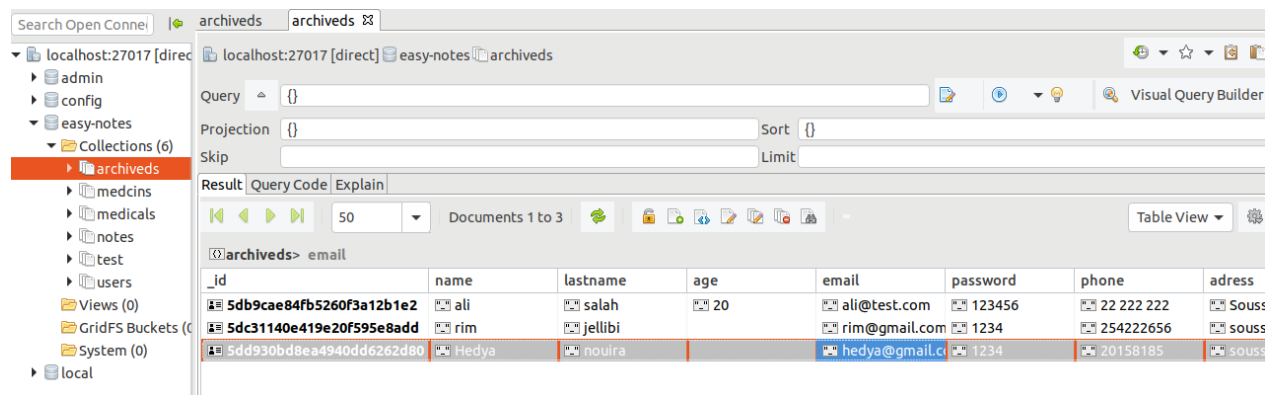


Figure 5.8 : contenu de la collection « Archive » dans la base de données

5.5 Synthèse de la Release

5.5.1 Diagramme de classe final

En avançant dans les sprints au cours de ce projet, nous sommes arrivés à un diagramme de classe final pour cette première release. Ce diagramme regroupe les classes précédemment faites.

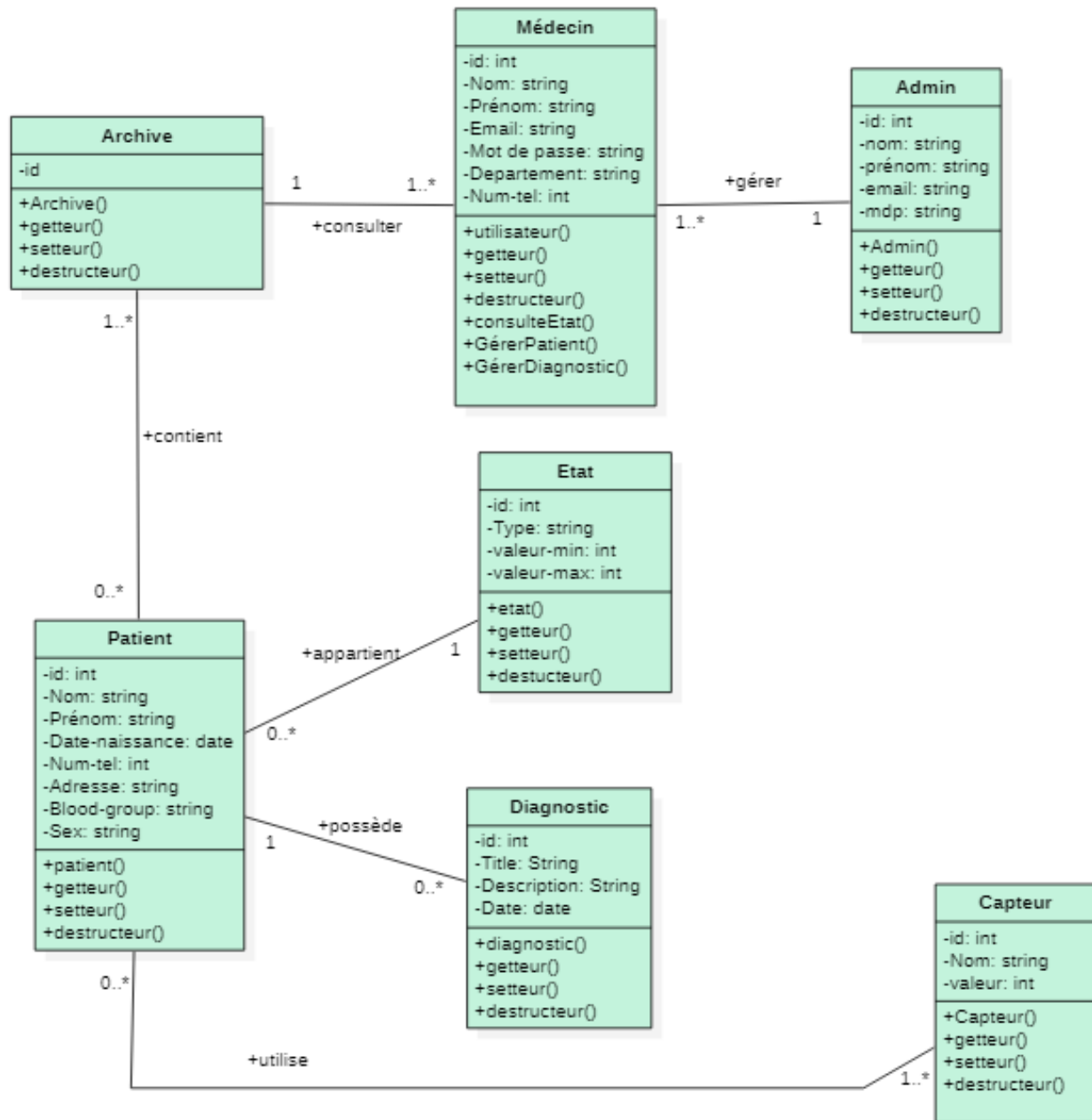


Figure 5.9: Diagramme de classe finale

5.5.2 Diagramme de déploiement

Un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. La figure 5.10 présente le diagramme de déploiement de la solution « EHealth » :

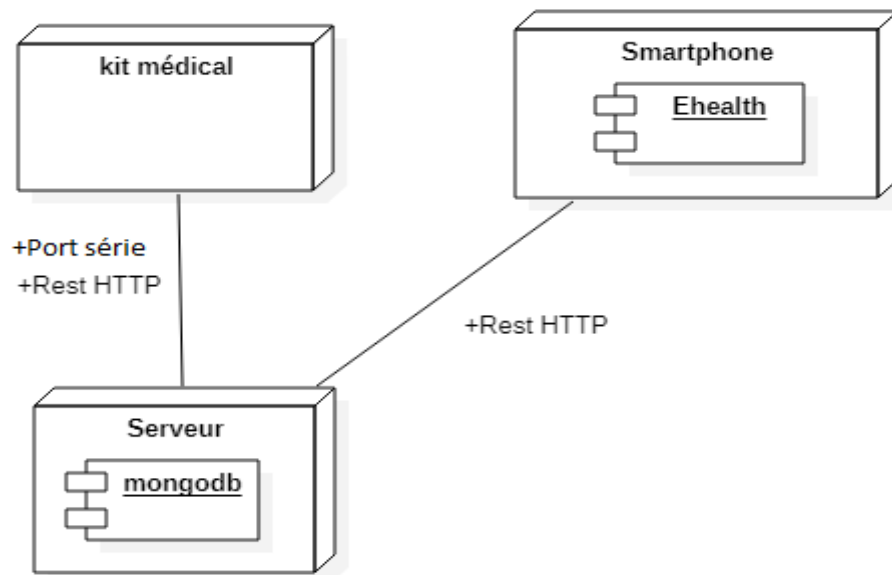


Figure 5.10: Diagramme de déploiement

Conclusion

Au cours de ce chapitre, nous avons traité le troisième sprint à travers les étapes de conception, réalisation et du test. Après avoir finalisé le troisième sprint, nous passons au sprint 4 dans lequel nous allons traiter la partie de sécurité de notre base de données mongodb.

Chapitre 6

Sprint 4 : Sécurité de la base de données

Sommaire

1. EHealth Sprint 4 planning	69
2. Authentification.....	70
3. Backup du base et restaurastion de la base	74
4. Restauration du base de données MongoDB.....	76

Introduction

Ce chapitre, est consacré pour réaliser la partie de la sécurité de la base de données de notre application. Nous traitons, dans un premier temps, le backlog du dernier sprint. Ensuite, nous présentons les méthodes que nous avons utilisé pour sécuriser notre base de donnée.

6.1 EHealth Sprint 4 planning

6.1.1 Objectif du sprint 4

Ce Sprint a pour objectif, de développer la partie qui permet la protection de la base de données. Pour la première partie de ce sprint nous allons commencer par la création d'un utilisateur et l'authentification à notre base. La deuxième partie se charge de réaliser la partie qui permet le backup de la base.

6.1.2 Sprint Backlog du Sprint 3

Tableau 6.1 : Sprint Backlog du Sprint 4

ID	User Story	ID Task	Task	Complexity
1	En tant que développeur je veux interdire les l'utilisateurs non authentifiés de consulter la base de données	1.1	Créer utilisateur et lui attribuer un rôle	S
		1.2	Activer l'autorisation	S
		1.3	Tester l'essai d'accès au base sans authentification et avec authentification	S

2	En tant que développeur je veux créer un backup de notre base de données .	2.1	Backup de la base	S
		2.2	Backup automatique de la base	S
		2.3	Nettoyage des anciennes sauvegardes de la base	S
		2.4	Restauration de la base de données	S

6.2 Authentification

Pour sécuriser l'accès à la base, il faut ajouter un ou plusieurs utilisateurs. Ce(s) dernier pourra ainsi se connecter à une base de données MongoDB avec les identifiants d'accès de l'utilisateur approprié.

6.2.1 Créer un utilisateur

Dans notre base nous avons créé l'utilisateur « rimjellibi » avec mot de passe « rimjellibi96 » et le rôle spécifié (dbOwner).

```
> db.createUser( { user: "rimjellibi",pwd: "rimjellibi96", roles: [ { role: "dbOwner", db: "easy-notes"} ] } )
```

Figure 6.1 :Création utilisateur en mongodb

On peut afficher les utilisateurs disponibles pour notre base de donnée

```
> show users
{
  "_id" : "easy-notes.rim",
  "userId" : UUID("4c724324-0d5d-44b8-b7b8-7ec5a05b549a"),
  "user" : "rim",
  "db" : "easy-notes",
  "roles" : [
    {
      "role" : "dbOwner",
      "db" : "easy-notes"
    }
  ]
}
{
  "_id" : "easy-notes.rimjellibi",
  "userId" : UUID("b0284843-344a-4ba0-91f9-aa53b8db72dd"),
  "user" : "rimjellibi",
  "db" : "easy-notes",
  "roles" : [
    {
      "role" : "dbOwner",
      "db" : "easy-notes"
    }
  ]
}
}
```

Figure 6.2 : liste des utilisateurs pour notre base

Il faut noter que la création des utilisateurs n'est pas suffisante pour contrôler l'accès à la base. Il faut activer l'autorisation relative aux utilisateurs créés.

6.2.2 Activer l'autorisation

Dans le fichier `/etc/mongod.conf`, on ajoute ces deux lignes :

```
security:
  authorization: "enabled"
```

Figure 6.3 : Activation de l'autorisation

Puis on redémarre le service mongod en tapant la commande située dans la figure 6.4 pour que le changement effectué dans le fichier de configuration soit pris en considération :

```
client@Lenovo-G580:~$ sudo service mongod restart
```

Figure 6.4 : Redémarrage du service mongod

6.2.3 Démarrer MongoDB sans authentification

Lorsque on essaie de démarrer MongoDB et afficher les bases disponibles et d'afficher les collections disponibles dans la base easy-notes cette erreur s'affiche « **there are no users authenticated** »

```
client@Lenovo-G580:~$ mongo --host 127.0.0.1:27017
MongoDB shell version v3.6.14
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongod
Implicit session: session { "id" : UUID("545c5467-6f1b-4ef4-b7ce-71c672be12b8") }
MongoDB server version: 3.6.14
> show dbs
2019-12-01T13:56:30.505+0100 E QUERY [thread1] Error: listDatabases failed: {
  "ok" : 0,
  "errmsg" : "there are no users authenticated",
  "code" : 13,
  "codeName" : "Unauthorized"
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:67:1
shellHelper.show@src/mongo/shell/utils.js:860:19
shellHelper@src/mongo/shell/utils.js:750:15
@(shellhelp2):1:1
> use easy-notes
switched to db easy-notes
> show collections
2019-12-01T13:57:31.708+0100 E QUERY [thread1] Error: listCollections failed: {
  "ok" : 0,
  "errmsg" : "there are no users authenticated",
  "code" : 13,
  "codeName" : "Unauthorized"
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype._getCollectionInfosCommand@src/mongo/shell/db.js:941:1
DB.prototype.getCollectionInfos@src/mongo/shell/db.js:953:19
DB.prototype.getCollectionNames@src/mongo/shell/db.js:964:16
shellHelper.show@src/mongo/shell/utils.js:853:9
shellHelper@src/mongo/shell/utils.js:750:15
@(shellhelp2):1:1
```

Figure 6.5 : Démarrer MongoDB sans authentification

Même à partir du Robo3T on ne peut pas accéder au bases de données disponible

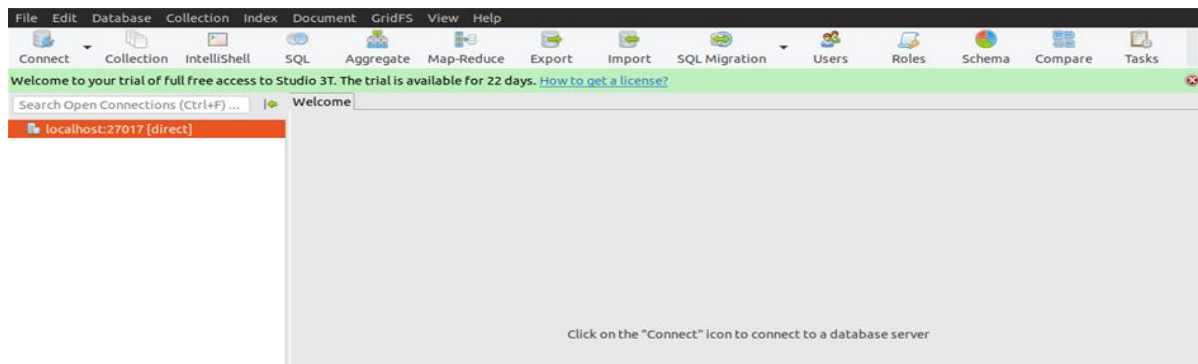


Figure 6.6 : Afficher les bases MongoDB sans authentification à partir du Robo3T

6.2.4 Se connecter en s'authentifiant

Lorsque on s'authentifie l'accès au collections devint possible

```

client@Lenovo-G580: ~
Fichier Édition Affichage Rechercher Terminal Aide
client@Lenovo-G580:~$ mongo --host 127.0.0.1:27017
MongoDB shell version v3.6.14
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("fc4c8791-4521-4678-8ecf-a6c135f43bc3")
}
MongoDB server version: 3.6.14
> use easy-notes
switched to db easy-notes
> db.auth("rimjellibi","rimjellibi96")
1
> show collections
archiveds
medcins
medicals
notes
test
users
>

```

Figure 6.7 : Lancer MongoDB après authentification

De meme pour accéder à notre base à partir du logiciel Robo3T nous devons s'authentifier d'abord pour avoir l'accès au base :

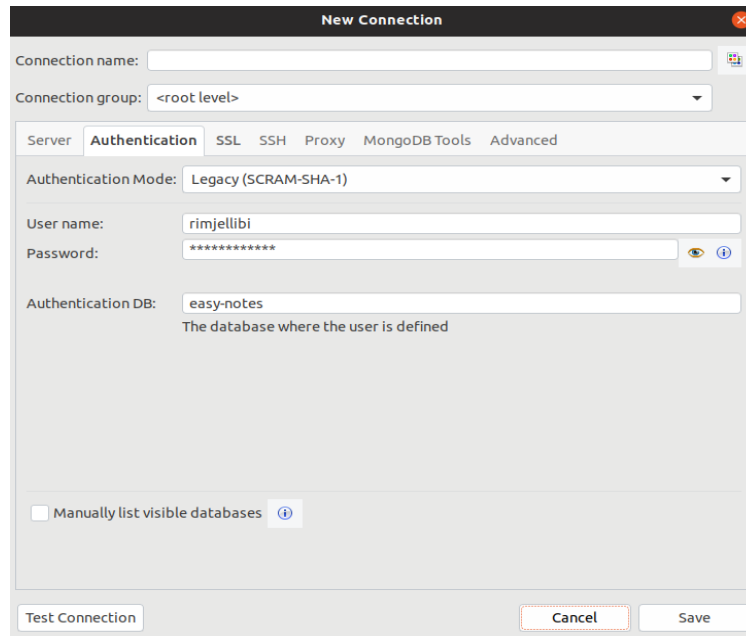


Figure 6.8 : authentification en Robo3T

Après l'authentification les collections sont visible dans Robo3T pour notre base de donnée. la figure 6.9 présente l'aspect du contenu de notre base de donnée dans Robo3T après authentification



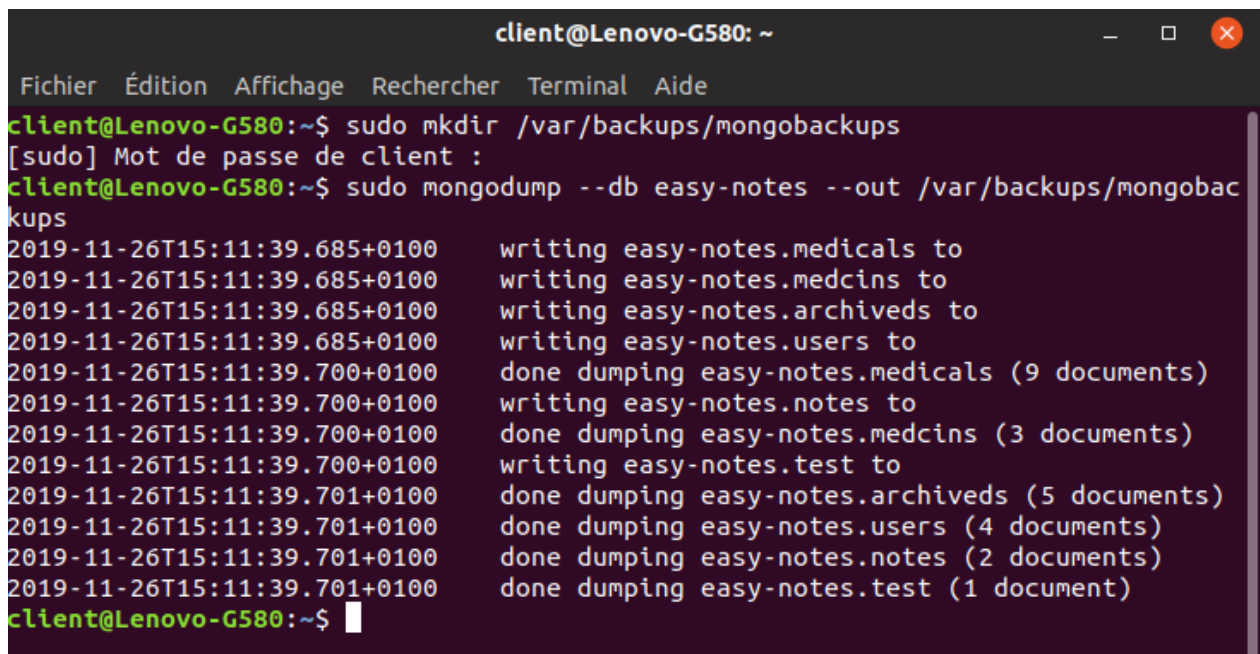
Figure 6.9 : contenu de la base après l'authentification

6.3 Backup et restauration de la base

6.3.1 Backup du base

mongodump est un outil livré avec mongoDB qui permet de sauvegarder en temps réel les données de mongoDB. Il peut être utilisé pour récupérer la totalité d'une base de données, une collection ou simplement le résultat d'une requête. mongodump peut créer une capture des données qui reflète un instant précis dans le temps pour ensuite pouvoir le rejouer durant un mongorestore

L'argument « --db » spécifie le nom de la base de données que on souhait sauvegarder. Le second argument « --out » sert à spécifier le répertoire dans lequel les données seront sauvegardées.



```

client@Lenovo-G580: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
client@Lenovo-G580:~$ sudo mkdir /var/backups/mongobackups
[sudo] Mot de passe de client :
client@Lenovo-G580:~$ sudo mongodump --db easy-notes --out /var/backups/mongobac
kups
2019-11-26T15:11:39.685+0100    writing easy-notes.medicals to
2019-11-26T15:11:39.685+0100    writing easy-notes.medcins to
2019-11-26T15:11:39.685+0100    writing easy-notes.archiveds to
2019-11-26T15:11:39.685+0100    writing easy-notes.users to
2019-11-26T15:11:39.700+0100    done dumping easy-notes.medicals (9 documents)
2019-11-26T15:11:39.700+0100    writing easy-notes.notes to
2019-11-26T15:11:39.700+0100    done dumping easy-notes.medcins (3 documents)
2019-11-26T15:11:39.700+0100    writing easy-notes.test to
2019-11-26T15:11:39.701+0100    done dumping easy-notes.archiveds (5 documents)
2019-11-26T15:11:39.701+0100    done dumping easy-notes.users (4 documents)
2019-11-26T15:11:39.701+0100    done dumping easy-notes.notes (2 documents)
2019-11-26T15:11:39.701+0100    done dumping easy-notes.test (1 document)
client@Lenovo-G580:~$

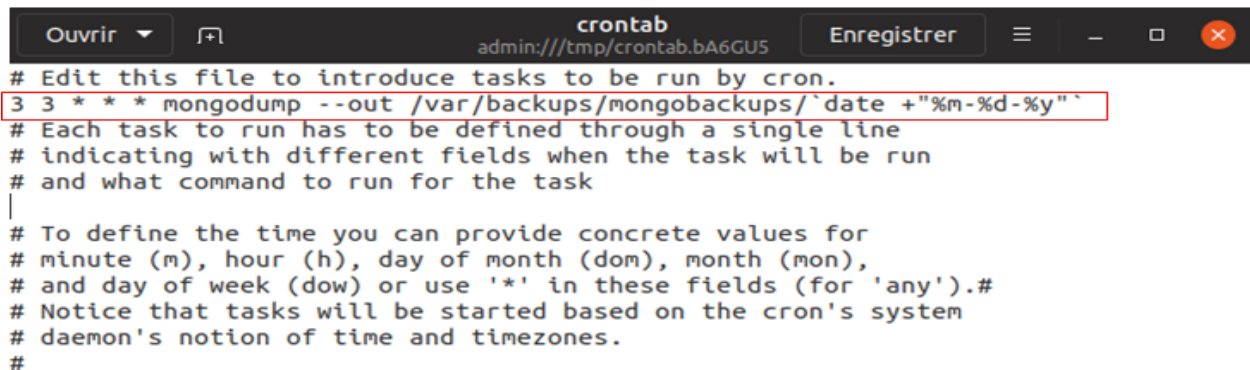
```

Figure 6.10 : Backup de notre base

6.3.2 Backup automatique du base

En règle générale, il est conseillé d'effectuer des sauvegardes régulières, par exemple, tous les jours et de préférence à un moment où le serveur est le moins chargé. Ainsi, nous définissons la commande mongodump en tant qu'une tâche cron afin qu'elle soit exécutée régulièrement, par exemple tous les jours à 03h03:

Dans l'invite crontab, on insère la commande mongodump présenté dans la figure 6.11:



```

Ouvrir  crontab  Enregistrer
admin:///tmp/crontab.bA6GU5
# Edit this file to introduce tasks to be run by cron.
3 3 * * * mongodump --out /var/backups/mongobackups/`date +"%m-%d-%y"`
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
|
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#

```

Figure 6.11 : Backup automatique de notre base

6.3.3 Nettoyage des anciennes sauvegardes du base

En fonction de la taille de notre base de données MongoDB, il est possible que nous manquions rapidement d'espace disque avec trop de sauvegardes. C'est pourquoi il est également recommandé de nettoyer les anciennes sauvegardes régulièrement ou de les compresser. Nous allons supprimer toutes les sauvegardes de plus de 7 jours.

```

client@Lenovo-G580:~$ find /var/backups/mongobackups/ -mtime +7 -exec rm -rf {} \;

```

Figure 6.12 : Nettoyage des anciennes sauvegardes

De la même manière que la commande précédente mongodump, celle-ci peut également être ajoutée en tant que tâche cron. Il doit être exécuté juste avant de lancer la prochaine sauvegarde. (Dans notre cas cette commande va être exécuté à 03h01).


```

client@Lenovo-G580: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
GNU nano 2.9.8 /tmp/crontab.kbrDAL/crontab

# Edit this file to introduce tasks to be run by cron.
3 3 * * * mongodump --out /var/backups/mongobackups/'date +%m-%d-%y'
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
3 1 * * * find /var/backups/mongobackups/ -mtime +7 -exec rm -rf {} \;
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#

```

Figure 6.13 : Nettoyage des anciennes sauvegardes automatiquement

La réalisation de toutes les tâches de cette étape garantira une solution de sauvegarde efficace pour les bases de données MongoDB.



Figure 6.14 : dossier du base de données récupéré

6.4 Restauration du base de données MongoDB

En restaurant notre base de données MongoDB à partir d'une sauvegarde précédente, nous obtenons la copie exacte de nos informations MongoDB à un moment donné, y compris tous les index et types de données. Pour restaurer MongoDB, nous utiliserons la commande mongorestore qui fonctionne avec la sauvegarde binaire produite par mongodump.

En tant qu'arguments, nous allons d'abord spécifier le nom de la base de données avec l'argument --db. Ensuite, avec --drop, nous nous assurerons que la base de données cible est d'abord supprimée afin que la sauvegarde soit restaurée dans une base de données vierge. En dernier argument, nous allons spécifier le répertoire de la dernière sauvegarde.

```

client@Lenovo-G580:~$ sudo mongorestore --db easy-notes --drop /var/backups/mongobackups/11-26-19/easy-notes/
2019-11-26T16:07:22.961+0100 the --db and --collection args should only be used when restoring from a BSON file. Other uses are deprecated
and will not exist in the future; use --nsInclude instead
2019-11-26T16:07:22.962+0100 building a list of collections to restore from /var/backups/mongobackups/11-26-19/easy-notes dir
2019-11-26T16:07:23.026+0100 reading metadata for easy-notes.archiveds from /var/backups/mongobackups/11-26-19/easy-notes/archiveds.metadata
a.json
2019-11-26T16:07:23.103+0100 reading metadata for easy-notes.medicals from /var/backups/mongobackups/11-26-19/easy-notes/medicals.metadata.
json
2019-11-26T16:07:23.148+0100 reading metadata for easy-notes.users from /var/backups/mongobackups/11-26-19/easy-notes/users.metadata.json
2019-11-26T16:07:23.214+0100 reading metadata for easy-notes.medicins from /var/backups/mongobackups/11-26-19/easy-notes/medicins.metadata.js
on
2019-11-26T16:07:23.414+0100 restoring easy-notes.archiveds from /var/backups/mongobackups/11-26-19/easy-notes/archiveds.bson
2019-11-26T16:07:23.683+0100 restoring easy-notes.medicals from /var/backups/mongobackups/11-26-19/easy-notes/medicals.bson
2019-11-26T16:07:23.907+0100 restoring easy-notes.users from /var/backups/mongobackups/11-26-19/easy-notes/users.bson
2019-11-26T16:07:24.113+0100 restoring easy-notes.medicins from /var/backups/mongobackups/11-26-19/easy-notes/medicins.bson
2019-11-26T16:07:24.113+0100 no indexes to restore
2019-11-26T16:07:24.114+0100 no indexes to restore
2019-11-26T16:07:24.114+0100 finished restoring easy-notes.archiveds (5 documents)
2019-11-26T16:07:24.114+0100 no indexes to restore
2019-11-26T16:07:24.114+0100 finished restoring easy-notes.medicals (9 documents)
2019-11-26T16:07:24.115+0100 finished restoring easy-notes.users (4 documents)
2019-11-26T16:07:24.168+0100 reading metadata for easy-notes.notes from /var/backups/mongobackups/11-26-19/easy-notes/notes.metadata.json
2019-11-26T16:07:24.224+0100 reading metadata for easy-notes.test from /var/backups/mongobackups/11-26-19/easy-notes/test.metadata.json
2019-11-26T16:07:24.224+0100 no indexes to restore
2019-11-26T16:07:24.224+0100 finished restoring easy-notes.medicins (3 documents)
2019-11-26T16:07:24.402+0100 restoring easy-notes.notes from /var/backups/mongobackups/11-26-19/easy-notes/notes.bson
2019-11-26T16:07:24.635+0100 no indexes to restore
2019-11-26T16:07:24.635+0100 finished restoring easy-notes.notes (2 documents)
2019-11-26T16:07:24.635+0100 restoring easy-notes.test from /var/backups/mongobackups/11-26-19/easy-notes/test.bson
2019-11-26T16:07:24.636+0100 no indexes to restore
2019-11-26T16:07:24.636+0100 finished restoring easy-notes.test (1 document)
2019-11-26T16:07:24.636+0100 done

```

Figure 6.15 : Restoration d'une base de donnée

Conclusion

Durant le dernier sprint, nous avons ajouter quelques aspect de sécurité à notre base de données en assurant un accès sécurisé par une authentification et en gardant un backup régulier de notre base pour la restaurer en cas de perte ou de problème. A ce stage là , notre application est prête à être livré.

Conclusion générale

Ce projet de fin d'études réalisé au sein de la société Enova ROBOTICS, a été pour moi une précieuse occasion pour développer mon savoir-faire et savoir être. En effet j'ai eu l'opportunité d'améliorer mes connaissances et d'enrichir mes capacités d'observation, d'analyse et de programmation. Ce stage alors m'a présenté une véritable occasion pour m'intégrer au milieu professionnel et mettre en œuvre mes acquis théoriques et pratiques.

Durant ce stage j'étais responsable tout d'abord de la mise en place de l'architecture de la solution, la conception, la réalisation, ainsi que les tests et le déploiement. En effet, ces modules consistent à la réalisation d'une application qui permet l'acquisition des mesures provenant d'un kit médical monté sur le robot téléprésence « COVEA », dans le but de surveiller les patients à distance. Selon les mesures obtenues, l'application classe les patients pour faciliter au médecin la suivie des états des patients.

Plusieurs fonctionnalités peuvent être groupés pour lancer le Product Backlog de la prochaine version, il contiendra notamment :

- L'ajout d'une interface d'inscription pour les médecins (ces inscriptions seront validé ou rejeté par l'administrateur).
- L'ajout d'un espace patient.
- L'ajout d'une notification temps réel si un patient passe de l'état stable à l'état anormal ou à l'état critique.

Webographie

[1] Enova Robotics. [Accès le 03 August 2019], adresse : <https://www.enovarobotics.eu/>

[2] Ken Schwaber et Jeff Sutherland. (July 2016). Scrum methodology. [Accès le 09 August 2019], adresse : <https://www.itpms.fr/wp-content/uploads/2018/01/2016-Scrum-Guide-FR.pdf>

[3] MySignal datasheet . [Accès le 11 August 2019], adresse : <https://www.cooking-hacks.com/mysignals-hw-ehealth-medical-biometric-iot-platform-arduino-tutorial/>

[4] Comparison between NoSQL databases. [Accès le 25 August 2019], adresse :

<https://www.journaldunet.com/web-tech/developpeur/1118927-comparatif-des-bases-de-donnees-nosql/>

[5] MongoDB database. [Accès le 13 August 2019], adresse : <https://buzut.net/commandes-de-base-de-mongodb/>

[6] Manuel Kiessling. Create Node.js application. [Accès le 11 September 2019], adresse : <https://nodejs.developpez.com/tutoriels/javascript/node-js-livre-debutant/>

[7] Olatunde Michael Garuba. (January 12 2017). Build Node.js RESTful APIs. [Accès le 20 September 2019], adresse : <https://www.codementor.io/olatundegaruba/nodejs-restful-apis-in-10-minutes-q0sgsfhbd>

[8] Diego Posa . (December 08, 2015). Convert Web App to a Mobile App with Apache Cordova. [Accès le 26 October 2019], adresse : <https://auth0.com/blog/convert-your-web-app-to-mobile/>

[9] Add user and authentication in MongoDB [Accès le 15 October 2019], adresse : <https://docs.mongodb.com/manual/tutorial/create-users/>

[10] Anatoliy Dimitrov. (April 15, 2016). Back Up and restore mongoDB database. [Accès le 28 november 2019], adresse : <https://www.digitalocean.com/community/tutorials/how-to-back-up-restore-and-migrate-a-mongodb-database-on-ubuntu-14-04>

Résumé

Le présent rapport synthétise le travail effectué dans le cadre du projet de fin d'études pour l'obtention du diplôme de master professionnelle de service et sécurité des réseaux et master IOT au sein de l'entreprise Enova Robotics. Le travail consiste à concevoir une application mobile « EHealth » pour gérer les informations des patients , des diagnostics et pour suivre les états des patients en récupérant des valeurs des attributs médicaux mesurées par kit médical « MySignal » monté sur un robot de téléprésence « COVEA »

Mots clés : Node js, python, Mongo DB, Arduino .

Abstract

This report summarizes the work carried out within the framework of the graduation project for obtaining the professional master's degree in service and network security and IOT master within Enova Robotics. The work consists of designing a mobile application to manage patient information, diagnoses and to follow the states of the patients by recovering values of the medical attributes measured by medical kit "MySignal" mounted on a telepresence robot "COVEA"

Keywords: Node js, python, Mongo DB, Arduino.