



RÉPUBLIQUE DU BÉNIN  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE  
RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88  
<http://www.ifri-uac.net> Courriel : contact@ifri.uac.bj



# MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Genie Logiciel

Présenté par :  
Ahouéfa Pascale Ninon KPOSSOU

**Développement d'une plateforme de  
gestion des rendez-vous et des dossiers  
médicaux dans les établissements de santé**

Sous la supervision :  
Dr Ing. Ratheil V. HOUNDJI  
Ing. Ronald HOUNKPONOU

Année Académique : 2016-2017

# Dédicaces

A

Mon père **Etienne KPOSSOU**

Ma mère **Honorine MENSAH**

Mes frères et sœurs

# Remerciements

Le présent mémoire que nous rédigeons à l'issue de nos trois années de formation à l'Institut de Formation et de Recherche en Informatique, n'aurait pas pu être réalisé sans conseils, concours et assistance des uns et des autres.

Qu'il nous soit permis d'adreser nos sincère remerciements à :

- DIEU le père tout puissant, pour nous avoir donner la santé, la force, l'inspiration et la perserverance tout au long de ce travail;
- **M. Ratheil HOUNDJI**, Docteur en Intelligeance artificielle et notre maitre de mémoire;
- **M. Ronald HOUKPONOU**, Ingerieur en Informatique et notre maitre de stage;
- **M. Gilles KOUNOU**, Ingénieur en Informatique et Directeur de OpenSi pour nous avoir accueillis au sein des a société;
- **Prof. C. Eugène EZIN**, Directeur de l'Institut de Formation et de Recherche en Informatique(IFRI);
- **M. Gaston EDAH**, Directeur Adjoint de l'IFRI, Chargé des affaires académiques, pour son sens de l'écoute et de la compréhension;
- **Ma famille**, pour leur soutien et l'amour qu'ils porte à mon égard;
- Tous les enseignants de notre institut pour avoir accepté partager une partie de leurs connaissances avec nous;
- Tous mes camarades de promotion et en particuliers ceux avec qui j'ai eu à effectuer mon stage de fin de cycle pour leur aide et leurs différentes suggestions apportées au cours du développement de notre projet;
- Tous ceux que j'ai avons oubliés de citer dans ce document et qui ont d'une manière ou d'une autre participé à la réussite de cette œuvre.

# Table des matières

<b>Dédicace</b>	<b>i</b>
<b>Remerciements</b>	<b>ii</b>
<b>Glossaire</b>	<b>2</b>
<b>Sigles et Abréviations</b>	<b>3</b>
<b>Résumé/Abstract</b>	<b>4</b>
<b>Introduction</b>	<b>5</b>
<b>1 Contexte d'étude et Revue de littérature</b>	<b>6</b>
1.1 Problématique . . . . .	6
1.2 Objectif Général . . . . .	7
1.3 Etat de l'art . . . . .	8
1.3.1 Présentation des solutions existantes . . . . .	8
1.3.2 Intérêt de la solution par rapport aux existantes . . . . .	12
<b>2 Analyse, Conception et Choix technique</b>	<b>14</b>
2.1 Analyse . . . . .	14
2.1.1 Analyse des besoins . . . . .	14
2.2 Conception . . . . .	17
2.2.1 Diagramme de cas d'utilisation . . . . .	17
2.2.2 Diagramme de classe . . . . .	23
2.2.3 Diagramme de séquences . . . . .	24
2.3 Choix technique . . . . .	26
2.3.1 Architecture du système . . . . .	26
2.3.2 Choix du langage de modélisation . . . . .	27
2.3.3 Choix du SGBD . . . . .	28
2.3.4 MongoDB . . . . .	28
2.3.5 Choix des outils de programmation . . . . .	29
2.3.6 Choix d'architectures logicielles . . . . .	31

2.3.7 Sécurité de l'application . . . . .	31
<b>3 Présentation du prototype de l'application et discussion</b>	<b>33</b>
3.1 Présentation des interfaces de l'application mobile de gestion de rendez-vous . .	33
3.2 Présentation des interfaces de l'application web de gestion des dossiers médicaux	36
3.3 Discussion . . . . .	37
<b>Conclusion</b>	<b>39</b>
<b>Bibliographie</b>	<b>40</b>

# Liste des Algorithmes

## Table des figures

1.1	Interface d'accueil de Doctolib . . . . .	9
1.2	Interface de Mondocteur . . . . .	9
1.3	Interfaces de Adaman7 . . . . .	10
1.4	Interfaces de DossierSanté . . . . .	10
1.5	Interface de Allo médecin . . . . .	11
1.6	Interfaces de KelDoc-Rdv . . . . .	11
1.7	Interfaces de AlloDoc . . . . .	12
1.8	Interface de Dis Moi Doc . . . . .	12
2.1	Diagramme de cas d'utilisation de MediCAL . . . . .	22
2.2	Diagramme de cas d'utilisation de MediCIS . . . . .	23
2.3	Diagramme de cas d'utilisation de l'API . . . . .	23
2.4	Diagramme de classe du système . . . . .	24
2.5	Diagramme de séquence du cas d'utilisation de demande d'un rendez-vous avec MediCal . . . . .	25
2.6	Diagramme de séquence de l'inscription avec MédiCal . . . . .	25
2.7	Diagramme de séquence du paiement avec MédiCal . . . . .	26
2.8	Diagramme de séquence d'ajout de patient avec MédICIS . . . . .	26
2.9	Architecture conceptuelle du système . . . . .	27
3.1	Interfaces de connexion et d'inscription . . . . .	33
3.2	Interfaces d'accueil du patient . . . . .	34
3.3	Interfaces de prise de rendez-vous . . . . .	34
3.4	Interfaces du médecin . . . . .	35
3.5	Interfaces d'accueil de MédICIS . . . . .	36
3.6	Interface d'ajout d'un nouveau patient . . . . .	36
3.7	Interface d'ajout d'un nouveau personnel . . . . .	37
3.8	Interfaces de consultation . . . . .	37

# Glossaire

- **API :** Application programming Interface  
Elle regroupe un ensemble de fonctions ou méthodes, leurs signatures et ordre d'usage pour obtenir un résultat.
- **HTTP :** HyperText Transfert Protocol  
C'est le protocole de transfert utilisé pour le World Wide Web. Il définit les messages que les clients peuvent envoyer au serveur, et ceux que le serveur peut transmettre en réponse.
- **Framework :**  
Ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme.
- **REST :** Representationnal State Transfert  
C'est une manière de construire les applications pour les systèmes distribués.
- **JSON :** JavaScript Object Notation  
C'est un format léger d'échange de données facile à lire ou écrire pour des humains
- **Android :**  
Il s'agit d'un système d'exploitation mobile basé sur le noyau mobile et développé actuellement par Google

# Sigles et Abréviations

- **API** : Application programming Interface
- **CSS** : Cascading Style Sheet
- **SGBD** : Système de Gestion de Base de Données
- **PHP** : Hypertext Preprocessor
- **HTTP** : HyperText Transfert Protocol
- **SQL** : Structured Query Language
- **UML** : Unified Modeling Language
- **SDK** : Software Development Kit
- **JSON** : JavaScript Object Notation
- **REST** : Representationnal State Transfert

# Résumé

Les établissements de santé sur le territoire béninois connaissent d'énormes difficultés dans la gestion efficace des prises de rendez-vous pour les consultations, des dossiers médicaux et informations des patients ainsi que les données relatives à leur personnel. La mise en place d'un système de gestion de rendez-vous et des dossiers médicaux dans les établissements de santé utilisant une application mobile implémentée avec le langage Java pour l'organisation des prises de rendez-vous et une application web implémentée avec Nodejs, Vuejs, Sailsjs pour la gestion des dossiers médicaux toutes deux communiquant avec une base de donnée MongoDB, sera profitable pour les populations. Cela leur permettra en effet de prendre rendez-vous en un clic dans un établissement de santé de leur choix à une heure de leur convenance au vue de la disponibilité du médecin ; permettra aux médecins de définir leurs plages horaires et consulter leur historique de rendez-vous ; également de mettre à jour le dossier médical de leur patient au cours des consultations et aux personnes en charge, selon le cas, d'enregistrer le personnel de l'établissement de santé et d'ajouter de nouveaux patients. L'objectif de ce système est en effet d'informatiser la gestion des prises de rendez-vous et des dossiers médicaux dans les établissements de santé.

**Mots clés :** *rendez-vous, établissement de santé, gestion, dossier médical.*

# Abstract

Healthcare in the Beninese territory is noticing a huge difficulty in the effective harvest management of appointments for consultations, information and medical records of patients as well as data on their staff. The implementation of a system of appointment management and medical records in hospitals using a mobile application is implemented with the language Java for the Organization of the catch of appointment and a web application practiced with Nodejs, Vuejs, Sailsjs for records management medical both contacting a MongoDB database, which will be beneficial for the populations. This will indeed allow them to make an appointment with a click in a facility of their choice at a time of their convenience in view of the availability of the doctor; allowing doctors to set their time slots and view their history of appointment; to also update medical records of their patients during consultations and people known as appropriate to record the health facility staff and add new patients. The goal of this system is to computerize any appointment and to manage all medical records in healthcare institutions.

**Key words:** *appointment, healthcare institution, management, medical records.*

# Introduction

Au cours de ces dernières années, les ordinateurs et les smartphones (téléphones intelligents) ont connu un véritable essor et sont de plus en plus dotés d'une puissance avec des fonctionnalités assez évoluées. Celles-ci favorisent la création et la distribution en toute simplicité des applications mobiles et web innovantes destinées à satisfaire les besoins des utilisateurs. L'informatique médicale est en plein essor et présente un avantage particulier pour l'innovation. Les hôpitaux et cliniques se digitalisent de plus en plus, confirmant ainsi la véritable révolution culturelle et technologique que vit depuis quelques années, la médecine.

Malheureusement, au Bénin, cette avancée notable introduite dans la médecine par la technologie peine à se faire sentir surtout à cause du faible taux de pénétration des TICs d'une part et d'autre part à cause de la législation assez rigide qui encadre la gestion des données personnelles. Face à cette situation, il urge de trouver une solution adéquate et adaptée permettant de répondre au mieux aux besoins du domaine de la santé au Bénin. C'est justement dans cet objectif qu'intervient la société d'expertise et de conseil sur les technologies de l'information et de la communication, OpenSI. Dans le présent travail, nous nous sommes intéressés d'une part à la problématique de la gestion des prises de rendez-vous et leur tenue dans les établissements de santé locaux puis d'autre part, à la gestion des dossiers médicaux des patients ainsi qu'à leur archivage dans lesdits établissements.

Ce projet "Développement d'une plateforme de gestion des rendez-vous et des dossiers médicaux dans les établissements de santé" qui fait l'objet dudit mémoire tente de trouver une solution à ce problème. Il se veut en effet, d'offrir aux populations béninoises et aux médecins une solution numérique pour informatiser la gestion des prises de rendez vous, des dossiers médicaux des patients et des données propres aux établissements de santé. Il s'agit d'un panel d'applications destiné aussi bien aux médecins qu'à leurs patients dans le but d'améliorer leurs interactions dans le processus de prise de rendez-vous et de gestion des dossiers médicaux.

Le présent document organisé en trois (03) chapitres présente les étapes de la réalisation du travail. Le premier chapitre définit le contexte d'étude et la revue de littérature en abordant la problématique, l'objectif et l'étude des solutions existantes. Le deuxième quant à lui porte sur les méthodologies de travail et de conception utilisées ainsi que sur les choix techniques réalisés pour mener à bien la tâche. Enfin, le troisième chapitre présente les résultats du travail effectué.

# Contexte d'étude et Revue de littérature

## 1.1 Problématique

Le constat est que le secteur médical béninois de nos jours ne dispose malheureusement pas d'un système de gestion de prises de rendez-vous des consultations efficace entraînant ainsi le non respect et suivi réel des horaires prises par chaque patient. L'impact de cette mauvaise gestion se traduit par le fait que le patient n'a aucune visibilité sur son heure de consultation , et ainsi ne peut organiser efficacement les autres engagements de sa journée : notion de perte de temps. En effet, dans les établissements de santé, il est possible de prendre rendez-vous pour une consultation aussi bien avec un médecin généraliste qu'avec un spécialiste. La prise de rendez-vous se fait actuellement soit en appelant soit en se déplaçant vers l'établissement. Les appels téléphoniques ne sont pas toujours concluants parce qu'il n'est pas aisé d'avoir le contact d'un centre de santé ou d'un médecin et les appels ne sont reçus qu'aux horaires d'ouverture du secrétariat de l'établissement de santé. De plus, il n'est pas rare de constater que le patient ou le médecin ne soit pas en mesure d'honorer un rendez-vous préalablement validé. Dans ces cas de figure, la seconde partie n'est généralement pas tenue informée à temps de l'annulation du rendez-vous. Il arrive aussi que l'une des deux personnes en particulier le patient, oublie sa date de consultation. Une seconde remarque, toujours dans le domaine sanitaire, est qu'il est difficile pour les agents sanitaires de réunir à temps les dossiers médicaux des patients surtout quand ceux-ci possèdent des dossiers sur plusieurs années dans ledit centre. Faute d'organisation, on note également un nombre important de cas de carnets médicaux et de dossiers médicaux égarés. Ces pertes d'archives entraînent une perte de continuité dans les dossiers médicaux des patients. De plus, chaque centre dispose de sa gestion de dossier. Ceci amène un éparpillement des informations des patients qui doivent aussi en cas de besoin se déplacer avec leur dossier. En effet, le patient qui se rend à l'établissement sans carnet de santé, se retrouve dans l'obligation d'en racheter un autre. Lorsque ce n'est pas sa première consultation dans cet

établissement, on lui demande la date ou la période de sa dernière consultation ; il ne s'en rappelle pas toujours ; il lui est alors créé un nouveau dossier médical. Ceci implique l'impossibilité d'un suivi efficace du patient sur une longue durée. Dans le cas où il se rappelle vaguement, les infirmiers cherchent dans les archives de dossiers médicaux le dossier médical du patient. Etant manuel, ce dossier s'il est retrouvé peut être dans un mauvais état, déchiré ou l'écriture effacée lorsque la dernière consultation remonte à longtemps. Le risque de perte des informations relatives aux antécédents du patient est ainsi élevé. Ces problèmes recensés dans la branche médicale reflètent d'une manière générale, un problème de gestion manuelle des archives au niveau des hôpitaux en ce qui concerne les dossiers de consultations. En effet, le processus de création, de classement, de conservation, et de communication des informations relatives à un patient, constitue une énorme perte de temps, de fiabilité et d'énormes erreurs techniques. De plus, ces dossiers créés sont laissés dans des casiers à la merci de toute source d'altération. À ce siècle marqué par la technologie où les pays recherchent une numérisation totale des activités dans tous les secteurs, une informatisation du système médical béninois s'avère profitable. La mise en place d'applications pouvant assurer la libre communication patient/centre sanitaire depuis la création d'un dossier médical en passant par sa conservation, jusqu'à son utilisation sur une période indéfinie, favorisera l'accès rapide aux archives ou dossiers médicaux. Nous proposons pour cela une application web MédiCis destinée aux professionnels de santé et une application mobile MédiCal pour la gestion des prises de rendez-vous destinée au public.

## 1.2 Objectif Général

---

Ce projet vise l'amélioration de la gestion des prises de rendez-vous dans les établissements de santé. Cela permettra la prise d'un rendez-vous, son report ou son annulation tant par le patient que par le médecin 24h/24 et 7jr/7. Il sera ainsi question de réduire le flux de coups de fil dans les établissements de santé ainsi que le nombre de rendez-vous non honorés ; situation qui pénalise aussi bien le patient que le médecin. Ce projet se veut également d'aider dans la gestion des dossiers médicaux. Il s'agira de permettre au patient à travers son application mobile de gestion de prise de rendez-vous de consulter son dossier médical. Il pourra ainsi accéder à son dossier médical à tout instant et autoriser ou non l'accès à son médecin et la modification de son dossier médical. Au médecin, il sera permis de gérer ses patients, son agenda, de retrouver le dossier médical d'un patient ; ce qui lui permet de suivre l'évolution de la santé du patient en ayant accès à ses antécédents et d'y faire des modifications au cours de sa consultation.

- Développer une application mobile pour la gestion des prises de rendez-vous qui répondra aux besoins suivants :
  - permettre la création et la gestion d'un compte utilisateur ;

- permettre la recherche des médecins ;
  - permettre la recherche des établissements de santé ;
  - permettre la prise d'un rendez-vous ;
  - permettre le paiement de consultation à l'avance ;
  - informer les patients sur les coûts des consultations dans les établissements ;
  - permettre le report ou l'annulation d'un rendez-vous ;
  - donner à l'utilisateur la possibilité d'avoir sur lui en tout temps son dossier médical électronique ;
  - grâce à son système intelligent de rappel, permettre de signaler au patient qu'il est proche de l'échéance d'un rendez-vous ou d'un contrôle, d'une vaccination ou d'une prise de ses médicaments pour un bon suivi
- Développer une application web pour la gestion des données des établissements de santé qui pour son compte permettra :
    - d'enregistrer le personnel
    - de gérer les profils et les droits d'accès à l'application
    - d'enregistrer les médecins intervenant dans l'établissement
    - de trouver un dossier médical
    - de rechercher un dossier médical
    - d'enregistrer les patients et de créer leur identifiant de santé unique
    - de rechercher un patient
    - de mettre à jour le dossier médical d'un patient au cours d'une consultation
    - de gérer les rendez-vous et les files d'attente
    - de gérer les statistiques de données en temps réels
    - de gérer la comptabilité

## 1.3 Etat de l'art

---

### 1.3.1 Présentation des solutions existantes

Il existe plusieurs solutions pour la gestion des prises de rendez-vous et des dossiers médicaux dans les établissements de santé. Il s'agit notamment de :

- **doctolib**

Doctolib<sup>1</sup> est une solution en France qui permet au patient la prise de rendez-vous en ligne chez un médecin ou un dentiste. Le patient a la possibilité de gérer son compte, d'annuler ou de modifier son rendez-vous. Au praticien, il permet de gagner du temps, de réduire le taux de rendez-vous non honorés, d'apporter un service nouveau à leur patient.

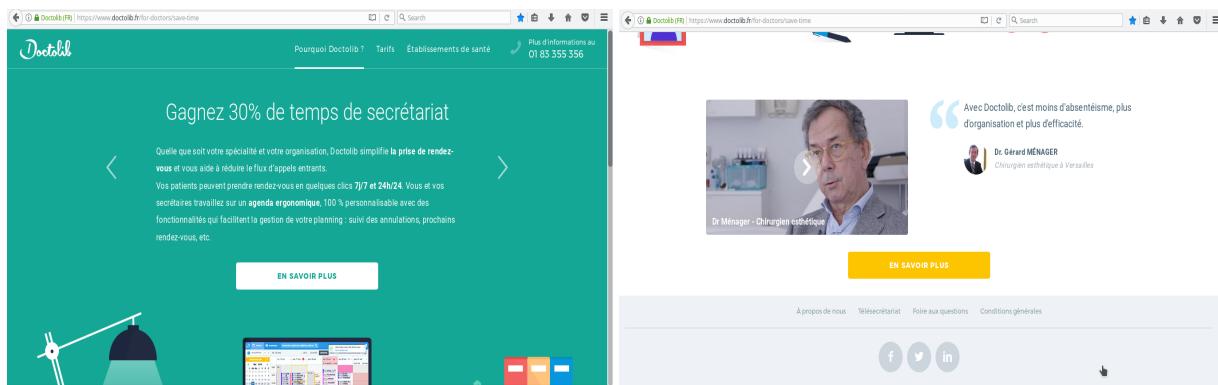


FIGURE 1.1 – Interface d'accueil de Doctolib

- **mondocteur**

Mondocteur<sup>2</sup> donne au praticien un site qu'il personnalise à son aise en gardant le contrôle sur son agenda. Ce site permet aux patients de prendre rendez-vous en ligne quelque soit l'heure. Il donne au médecin une vue sur son agenda, lui fournit une base patient personnalisable et lui permet de communiquer avec ses confrères.



FIGURE 1.2 – Interface de Mondocteur

- **adaman7**

<sup>1</sup>[www.doctolib.fr](http://www.doctolib.fr)

<sup>2</sup>[www.mondocteur.fr](http://www.mondocteur.fr)

Andaman<sup>7</sup> est une application mobile pour la gestion des dossiers médicaux des patients. Il permet au patient de garder son dossier médical sur lui partout où il est et de partager son dossier médical avec seulement les personnes qu'il a choisi (médecin comme proche).

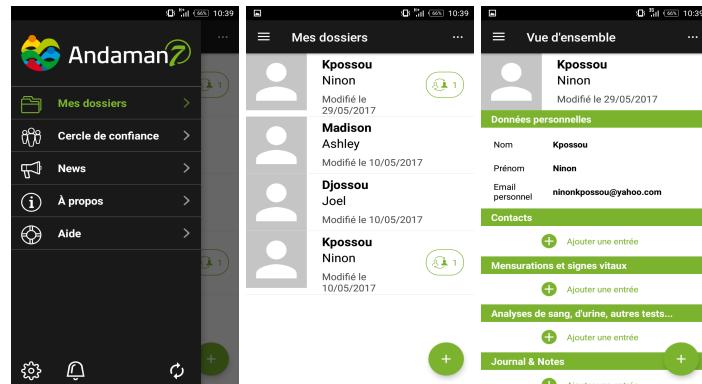


FIGURE 1.3 – Interfaces de Adaman7

- **DossierSanté**

DossierSanté<sup>4</sup> est une application mobile qui vous permet d'avoir à tout moment accès à votre dossier de santé. Avec DossierSanté l'utilisateur peut remplir lui même son dossier médical.

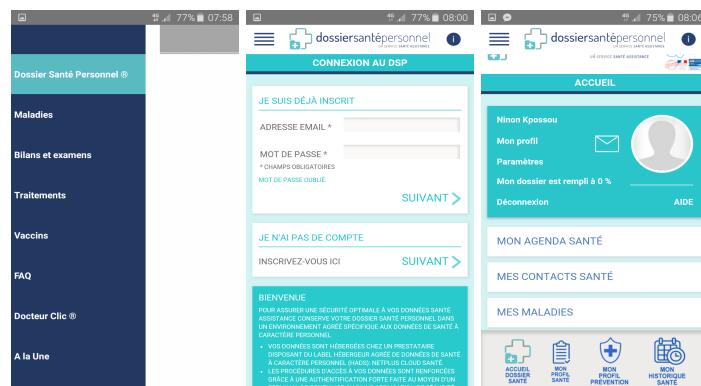


FIGURE 1.4 – Interfaces de DossierSanté

- **Allo médecin**

Allo médecin<sup>5</sup> est une application mobile pour la gestion des rendez-vous en ligne avec un spécialiste de santé en Belgique. Il permet au patient de gérer ses rendez-vous (annulation et modification).

<sup>3</sup>[www.adaman7.com](http://www.adaman7.com)

<sup>4</sup>[www.dossiersantepersonnel.com](http://www.dossiersantepersonnel.com)

<sup>5</sup>[www.allomedecin.be](http://www.allomedecin.be)



FIGURE 1.5 – Interface de Allo médecin

- **KelDoc-Rdv**

KelDoc-Rdv<sup>6</sup> est une application mobile qui permet de prendre rendez-vous avec un spécialiste de santé proche de chez eux moins cher et disponible.

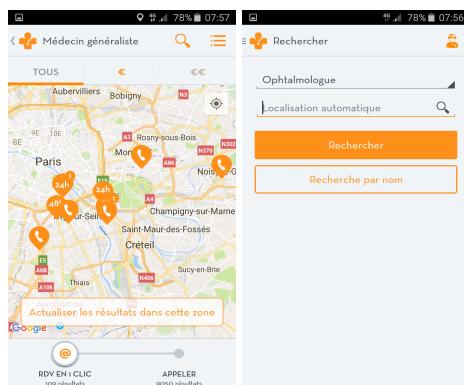


FIGURE 1.6 – Interfaces de KelDoc-Rdv

Toutes ces applications citées sont d'une grande aide dans le domaine médical mais ne prennent pas en compte le Bénin.

Par ailleurs il existe **AlloDoc** et **Dis Moi Doc** qui prennent le Bénin en compte. AlloDoc<sup>7</sup> est une application mobile qui met en contact patient et médecin. Avec AlloDoc les patients peuvent retrouver un médecin dans une ville ou un pays et les joindre par appels ou messages. Ils ont également des informations utiles sur la santé à travers des articles écrits par des médecins. Ils peuvent poser les inquiétudes et obtenir des réponses de médecins. Ils peuvent également se renseigner sur le prix des médicaments.

Avec Dis Moi Doc<sup>8</sup>, application mobile pour le Bénin et le Togo, les utilisateurs peuvent

<sup>6</sup>

<sup>7</sup>[www.ashabistore.com](http://www.ashabistore.com)

<sup>8</sup><https://dismoidoc.com>

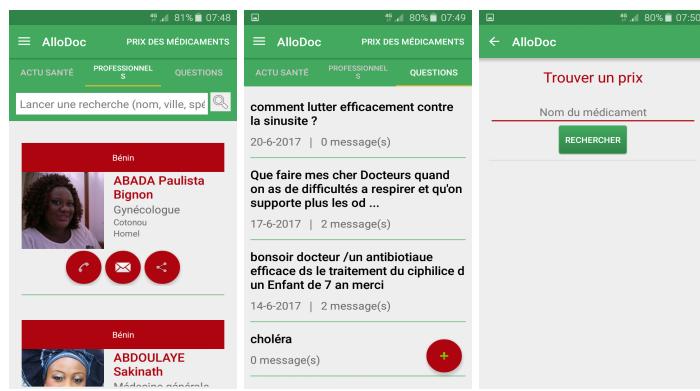


FIGURE 1.7 – Interfaces de AlloDoc

discuter avec des patients. Elle offre également aux utilisateurs un accès au carnet vaccinal et aux prix des médicaments sur le marché.



FIGURE 1.8 – Interface de Dis Moi Doc

### 1.3.2 Intérêt de la solution par rapport aux existantes

Des solutions étudiées, seules deux sont utiles sur le territoire béninois. Bien qu'elles offrent un espace de messagerie avec les médecins, elles ne facilitent ni la gestion des rendez-vous pour les patients, ni la gestion de l'agenda pour le médecin ou encore la gestion et l'archivage des dossiers médicaux des patients. Le temps que perdent les patients dans les hôpitaux pour rencontrer un médecin, la perte qui revient aux praticiens lorsque les patients n'arrivent pas à honorer leur rendez-vous, la perte courante des carnets de santé et des dossiers médicaux des patients ajoutés à l'importance capitale pour le médecin de la connaissance des antécédents médicaux d'un patient pour une nouvelle consultation justifient la nécessité de la mise en place d'un système numérique pour la gestion efficace des rendez-vous et des dossiers médicaux. En effet, avec ce système les patients pourront rapidement prendre rendez-vous dans un établissement de santé pour une consultation et gérer leur rendez-vous. Ils auront également en tout

temps accès à leurs dossiers médicaux. Pour toute consultation, les médecins pourront facilement retrouver le dossier médical de leur patient et le mettre à jour avec les nouvelles données de consultation. Ceci assure le suivi effectif à long terme du patient.

### **Conclusion partielle**

Au terme de ce chapitre, nous pouvons retenir que l'objectif que vise ce projet est d'offrir aux populations béninoises un outil efficace pour l'informatisation de la gestion des rendez-vous et des dossiers de santé. Nous retenons également qu'il existe de nombreuses solutions dans le domaine, pour la plupart n'intervenant pas au Bénin et pour le reste ne couvrant pas les aspects auxquels nous nous intéressons ici.

# Analyse, Conception et Choix technique

## 2.1 Analyse

Pour conduire ce projet, nous avons opté pour la Méthode **Scrum** qui est un schéma d'organisation de développement de produits complexes considérée comme une pratique **agile**; pratique qui implique au maximum le demandeur (client) et permet une grande réactivité à ses demandes. Scrum est défini comme un cadre de travail suivant un cycle de développement itératif, incrémental et adaptatif. S'appuyant sur le découpage d'un projet en boîtes de temps, nommées « sprints » pouvant durer entre quelques heures et un mois (avec une préférence pour deux semaines). Chaque sprint commence par une estimation suivie d'une planification opérationnelle et se termine par une démonstration de ce qui a été achevé.

### 2.1.1 Analyse des besoins

L'analyse consiste à l'aboutissement de l'élaboration d'une solution technique à partir de l'étude des besoins. C'est la première phase du cycle de développement d'un logiciel. Elle sert à identifier les acteurs du système et leur associer chacun l'ensemble des actions avec lesquelles il intervient dans l'objectif de donner un résultat optimal et satisfaisant au client.

#### 2.1.1.1 Les besoins fonctionnels

Les besoins fonctionnels répondent aux points précis du cahier de charges et sont donc requis par le client. Ils constituent le besoin primaire du client et définissent une fois résolu l'opérationnalité du système. Ainsi l'application de gestion de prise de rendez-vous dans les établissements de santé doit permettre :

- **au patient de :**
  - gérer les informations de son compte

- consulter l'annuaire des établissements de santé
- consulter l'annuaire des professionnels de santé
- prendre rendez-vous dans un établissement de santé
- prendre connaissances des tarifs
- payer une consultation
- reporter ou annuler un rendez-vous dans les délais autorisés
- consulter son historique de rendez-vous et de paiements
- consulter son carnet de santé électronique
- recevoir des notifications et rappels.

- **au médecin de :**

- gérer son compte utilisateur
- consulter l'annuaire des établissements de santé
- consulter son historique de rendez-vous
- reporter ou annuler un rendez-vous dans les délais autorisés
- définir sa disponibilité
- recevoir des notifications.

L'application web de gestion des dossiers de consultations dans les établissements de santé doit permettre quant à elle :

- **au médecin :**

- de créer un dossier médical
- de rechercher un dossier médical
- de consulter son historique de rendez-vous
- d'annuler ou de reporter un rendez-vous
- de consulter l'ordre du jour
- de recevoir des notifications.
- de mettre à jour le dossier médical avec le nouveau diagnostic et les informations utiles
- de gérer sa disponibilité pour les horaires de rendez-vous.
- d'informer des rendez-vous pris par appels pour une mise à jour des horaires disponibles.

- **au personnel médical selon le rôle :**

- de créer un dossier médical
- de rechercher un dossier médical
- d'établir la file d'attente
- de mettre à jour le dossier médical en renseignant les constantes
- de consulter les historiques de rendez-vous des médecins
- d'annuler ou de reporter un rendez-vous
- de recevoir des notifications.
- d'informer le système de la disponibilité des médecins.
- d'informer le système des rendez-vous pris par appels pour une mise à jour des horaires disponibles

- **au système :**

- de mettre à jour les horaires de rendez-vous disponibles
- d'envoyer les notifications
- d'interdire le report ou l'annulation d'un rendez-vous à moins de 24 heures du jour de consultation

- **à l'établissement de santé :**

- d'enregistrer le personnel avec les rôles

### 2.1.1.2 Les besoins non fonctionnels

Ces besoins sont soit des besoins optionnels soit des besoins/contraintes liés à l'implémentation. Ainsi, il faudra que **MédiCal** soit :

- sécurisé
- compatible à tout système android de version minimum 4.3
- doté d'une bonne expérience utilisateur
- performante

**MédiCIS** quand à elle, étant une application web , devra être :

- compatible à la plupart des navigateurs
- performante
- sécurisée

## 2.2 Conception

---

Cette phase permettra de définir de manière non ambiguë, en utilisant un langage de modélisation le fonctionnement futur du système, afin d'en faciliter laa réalisation. Pour la modélisatin de chacune des appplications, nous distinguerons :

- un diagramme de cas d'utilisation comme diagramme de comportement
- des diagrammes de séquences comme diagrammes d'interactions

Nous terminerons par un diagramme de classe comme diagramme statique

### 2.2.1 Diagramme de cas d'utilisation

Cette partie de notre modélisation consistera à présenter les différents acteurs du système ainsi que les différentes interactions qu'ils pourront effectuer avec celui-ci.

#### 2.2.1.1 Identification des acteurs

En UML, on entend par acteur toute idéalisation d'un rôle joué par une personne externe, un processus ou une entité qui interagit avec le système. Il existe deux types d'acteurs, à savoir les acteurs principaux, qui utilisent directement les fonctionnalités du système et les acteurs secondaires qui contribuent à la réalisation d'un cas d'utilisation. Dans le cadre de notre système nous avons retenu :

- Pour MédiCal les acteurs suivants :
  - le patient
  - le médecin
  - le système
- Pour MédICIS les acteurs suivants :
  - le médecin
  - l'infirmière à l'accueil
  - l'infirmière qui prend les constantes
  - le système

### 2.2.1.2 Description textuelle des cas d'utilisation

Nous décrirons deux (2) cas d'utilisation pour l'application mobile de gestion de prise de rendez-vous et deux cas d'utilisation pour l'application web de gestion des données des établissements de santé.

Pour **MédiCal** nous décrirons :

- prendre rendez-vous pour une consultation
- définir sa disponibilité

#### Cas d'utilisation 1 : Prendre rendez-vous pour une consultation

On distingue trois possibilités :

- L'utilisateur connaît son médecin
- L'utilisateur ne connaît pas de médecin en particulier et cherche ainsi un médecin
- L'utilisateur désire prendre le rendez-vous pour une date précise ; il ne veut donc pas juste se conformer à la disponibilité du médecin

**Acteur :** Patient

**Description :** Toute personne souhaitant prendre rendez-vous dans un établissement de santé avec un praticien de santé pour une consultation

**Pré-condition :** être connecté à internet

**Démarrage :** L'utilisateur lance l'application

**Possibilité 1 : L'utilisateur connaît son médecin**

**Scénario principal :**

1. L'utilisateur s'authentifie
2. L'utilisateur arrive sur l'interface principale
3. Il clique sur le bouton Crée un Rendez-vous
4. Il choisit l'option Je connais mon médecin ce qui lui présentera la liste de tous les spécialistes de santé avec une barre de recherche pour lui permettre de rechercher son médecin suivant son nom
5. Il sélectionne son médecin et obtient une liste lui présentant les établissements de santé pour la spécialité concernée et la disponibilité du médecin dans l'établissement. Il peut faire un filtre suivant la localisation ou le prix de la consultation
6. Il sélectionne l'établissement qui lui convient et vient sur une interface qui lui permet de choisir une date et une heure en fonction de la disponibilité du médecin pour la consultation

7. Une fois la date et l'heure choisies, l'utilisateur sélectionne son mode de paiement
  - a. S'il choisit de **payer à la consultation**, il peut valider son rendez-vous
  - b. S'il choisit de faire **payer par un référent**, on lui demande de renseigner les informations de ce dernier à qui le système envoie une notification et il peut valider son rendez-vous
  - c. S'il choisit de **payer maintenant**, il choisit la nature du paiement Mobile Money ou Carte Visa, il paie et peut valider son rendez-vous
8. Il appuie sur le bouton **Valider le rendez-vous** pour valider son rendez-vous et obtient une interface récapitulant les informations relatives aux rendez-vous
9. Il appuie sur le bouton **Terminer** pour revenir à l'accueil

**Scénario d'erreur :** authentification échouée.

**Possibilité 2 : L'utilisateur cherche un spécialiste**

**Scénario principal :**

1. L'utilisateur s'authentifie
2. L'utilisateur arrive sur l'interface principale
3. Il clique sur le bouton de création de Rendez-vous
4. Il choisit l'option **Je cherche un spécialiste** ce qui lui présentera la liste de toutes les spécialités avec une barre de recherche pour lui permettre de rechercher une spécialité en inscrivant son nom
5. Il sélectionne la spécialité et obtient une liste lui présentant les établissements de santé pour la spécialité concernée et la disponibilité des médecins dans l'établissement. Il peut faire un filtre suivant la localisation ou le prix de la consultation
6. Il sélectionne l'établissement qui lui convient et vient sur une interface qui lui permet de choisir une date et une heure en fonction de la disponibilité du médecin pour la consultation
7. Une fois la date et l'heure choisies, l'utilisateur sélectionne son mode de paiement
  - a. S'il choisit de **payer à la consultation**, il peut valider son rendez-vous
  - b. S'il choisit de faire **payer par un référent**, on lui demande de renseigner les informations de ce dernier à qui le système envoie une notification et il peut valider son rendez-vous

- c. S'il choisit de **payer maintenant**, il choisit la nature du paiement Mobile Money ou Carte Visa, il paie et peut valider son rendez-vous
- 8. Il appuie sur le bouton **Valider le rendez-vous** pour valider son rendez-vous et obtient une interface récapitulant les informations relatives aux rendez-vous
- 9. Il appuie sur le bouton **Terminer** pour revenir à l'accueil

**Scénario d'erreur :** authentification échouée.

#### Possibilité 3 : L'utilisateur choisit une date

##### Scénario principal :

1. L'utilisateur s'authentifie
2. L'utilisateur arrive sur l'interface principale
3. Il clique sur le bouton **Créer un Rendez-vous**
4. Il choisit l'option **Je choisis une date** ce qui lui présentera un calendrier
5. Il sélectionne la date qui lui convient et obtient la liste des spécialités
6. Il sélectionne une spécialité et obtient la liste des médecins disponibles pour la spécialité et la date sélectionnées et la liste des établissements de santé dans lesquels il ya des médecins disponibles pour la spécialité et la date sélectionnées
7. Lorsqu'il ne connaît pas de médecin en particulier, il choisit l'établissement de santé et arrive sur l'interface pour le choix de l'heure du rendez-vous. Dans le cas contraire, il sélectionne le médecin, obtient la liste des centres sanitaires dans lesquels il intervient, choisit ensuite le centre qui lui convient et arrive sur l'interface du choix de l'heure
8. Une fois l'heure choisie, l'utilisateur sélectionne son mode de paiement
  - a. S'il choisit de **payer à la consultation**, il peut valider son rendez-vous
  - b. S'il choisit de faire **payer par un référent**, on lui demande de renseigner les informations de ce dernier à qui le système envoie une notification et il peut valider son rendez-vous
  - c. S'il choisit de **payer maintenant**, il choisit la nature du paiement Mobile Money ou Carte Visa , il paie et peut valider son rendez-vous
9. Il appuie sur le bouton **Valider le rendez-vous** pour valider son rendez-vous et obtient une interface récapitulant les informations relatives aux rendez-vous
10. Il appuie sur le bouton **Terminer** pour revenir à l'accueil

**Scénario d'erreur :** authentification échouée.

**Cas d'utilisation 2 :** Définir sa disponibilité

**Acteur :** Médecin

**Description :** Le praticien de santé peut définir les heures auxquelles il est disponible

**Pré-condition :** être connecté à internet

**Démarrage :** L'utilisateur lance l'application

**Scénario principal :**

1. L'utilisateur s'authentifie
2. L'utilisateur arrive sur l'interface principale
3. Il clique sur le bouton **Ajouter un événement** lorsque c'est la première fois qu'il veut prendre rendez-vous à travers l'application ou sur le bouton comportant le "+" lorsque ce n'est pas sa première prise de rendez-vous et obtient la liste des établissements de santé dans lesquels il intervient
4. Il choisit l'établissement de santé dans lequel il aimerait modifier sa disponibilité et arrive sur une interface qui lui présente sa disponibilité actuelle
5. Il clique ensuite sur le bouton **Modifier**
6. Il définit les mois, les jours de la semaine et les heures de début et de fin
7. Il clique sur le bouton Enregistrer pour enregistrer ses changements
8. Une boîte de dialogue avec un message de confirmation s'affiche
9. Il clique sur **OUI** pour terminer et revenir à l'accueil

**Scénario d'erreur :** authentification échouée.

Pour MédICIS nous décrirons :

Ajouter un nouveau patient

**Acteur :** Infirmière à l'accueil ou médecin

**Description :** Tl'infirmière à l'accueil ou le médecin peut enregistrer un nouveau patient

**Pré-condition :** être connecté à internet

**Démarrage :** L'utilisateur lance l'application

**Scénario principal :**

1. L'utilisateur s'authentifie
2. L'utilisateur arrive sur l'interface principale
3. Il clique sur le bouton **Ajouter un nouveau patient**

4. Il remplit le formulaire avec les informations d'identification du patient et clique sur le bouton **Valider**
5. Il reçoit une notification de succès d'enregistrement
  - a. S'il choisit de **payer à la consultation** il peut valider son rendez-vous
  - b. S'il choisit de faire **payer par un référent** on lui demande de renseigner les informations de ce dernier à qui le système envoie une notification et il peut valider son rendez-vous
  - c. S'il choisit de **payer maintenant** il choisit la nature du paiement Mobile Money ou Carte Visa , il paie et peut valider son rendez-vous
8. Il appuie sur le bouton **Valider le rendez-vous** pour valider son rendez-vous et obtient une interface récapitulant les informations relatives aux rendez-vous
9. Il appuie sur le bouton **Terminer** pour revenir à l'accueil

#### Scénario d'erreur :

1. authentification échouée.
2. enregistrement du patient échoué.Ceci peut survenir lorsque l'utilisateur ne renseigne pas tous les champs requis.

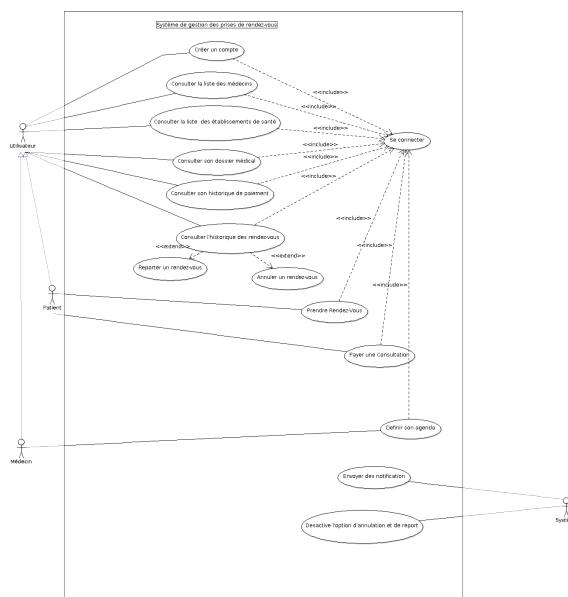


FIGURE 2.1 – Diagramme de cas d'utilisation de MediCAL

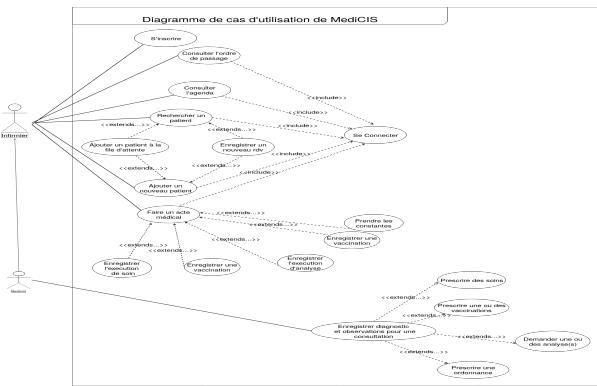


FIGURE 2.2 – Diagramme de cas d'utilisation de MediCIS

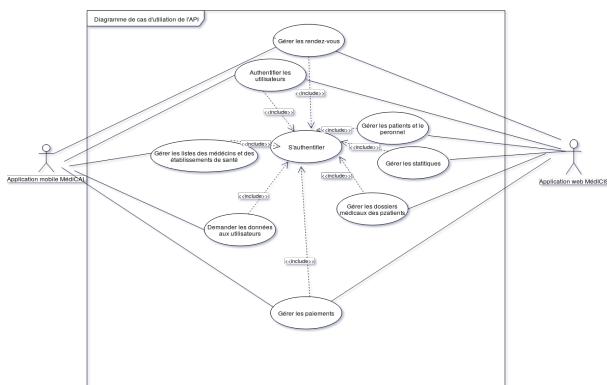


FIGURE 2.3 – Diagramme de cas d'utilisation de l'API

## 2.2.2 Diagramme de classe

Le diagramme de classe montre la structure interne d'un système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

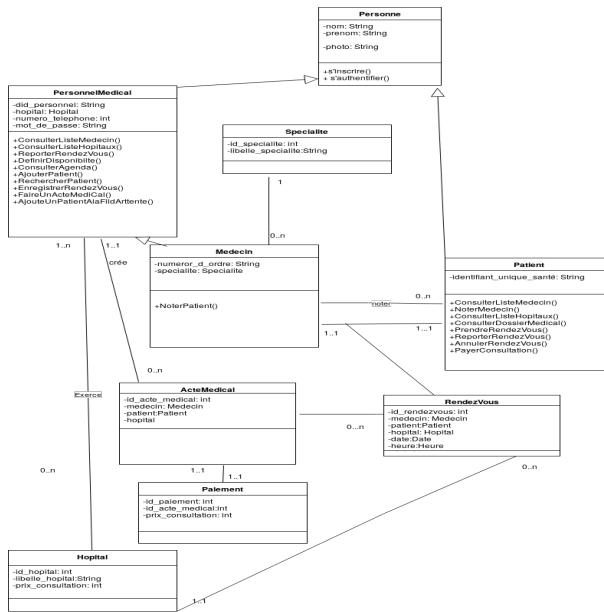


FIGURE 2.4 – Diagramme de classe du système

### 2.2.3 Diagramme de séquences

Le diagramme de séquence est une représentation intuitive lorsque l'on souhaite concrétiser des interactions entre deux entités (deux sous-systèmes ou deux classes d'un futur logiciel). Ils permettent à l'architecte ou le designer de créer au fur et à mesure sa solution. Cette représentation intuitive est également un excellent vecteur de communication dans une équipe d'ingénierie pour discuter cette solution. Ils permettent de décrire COMMENT les éléments du système interagissent entre eux et avec les acteurs :

- Les objets au cœur d'un système interagissent en s'échangeant des messages.
- Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

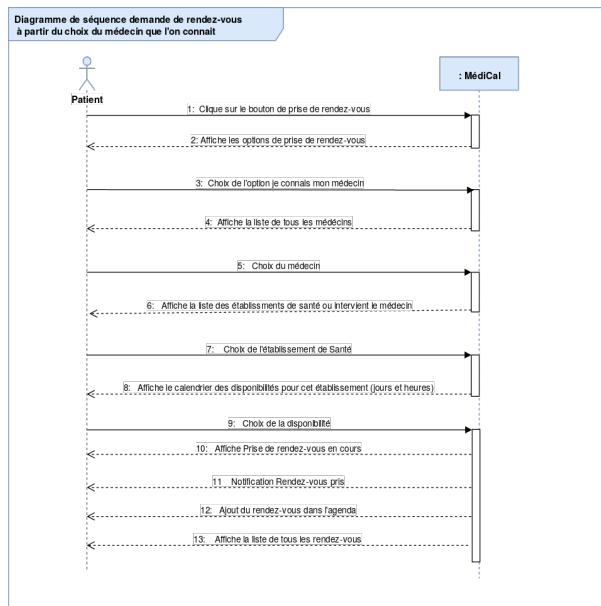


FIGURE 2.5 – Diagramme de séquence du cas d'utilisation de demande d'un rendez-vous avec MediCal

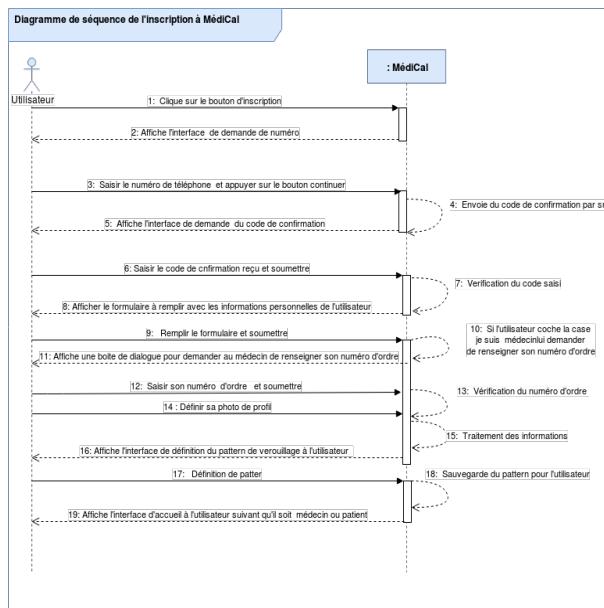


FIGURE 2.6 – Diagramme de séquence de l'inscription avec MédiCal

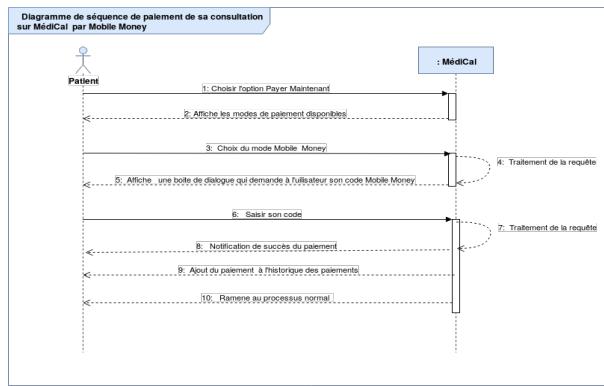


FIGURE 2.7 – Diagramme de séquence du paiement avec MédiCal

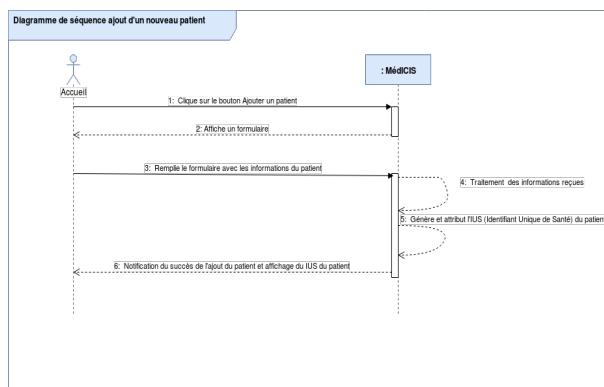


FIGURE 2.8 – Diagramme de séquence d'ajout de patient avec MédICIS

## 2.3 Choix technique

### 2.3.1 Architecture du système

Le projet comporte quatre applications à réaliser qui partagent un ensemble d'informations mais sont destinées à différents publics. Pour cela il faudrait déjà dans un premier temps que ces applications partagent une base de données commune, mais aussi un point d'entrée commun pour accéder à celle-ci. Ce besoin nous amène donc à concevoir une interface de programmation au dessus du système d'information. Une architecture client serveur n - tiers se décerne donc comme solution architecturale à notre cas. Au cœur de cette architecture un composant API ( Application Programming Interface soit Interface de Programmation Applicative) devra fournir sous forme de services les fonctionnalités qui ont trait à la logique du métier et aux systèmes d'informations des établissements de santé pour toutes nos applications clientes à condition d'être autorisées. Ainsi toutes les fonctionnalités métiers citées dans le projet doivent être développées au niveau de cette API. Les applications clientes tierces n'auront qu'à demander

des services.

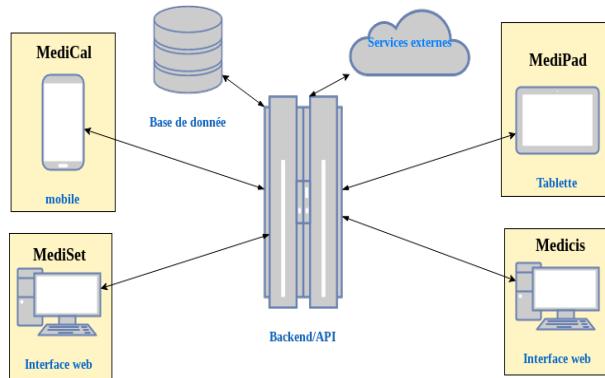


FIGURE 2.9 – Architecture conceptuelle du système

Dans notre travail nous nous sommes intéressés à l’application mobile **MédiCal** et à l’application web **Médicis**.

### 2.3.2 Choix du langage de modélisation

Un langage de modélisation est un langage artificiel qui peut être utilisé pour exprimer de l’information ou de la connaissance ou des systèmes dans une structure qui est définie par un ensemble cohérent de règles.

#### 2.3.2.1 MERISE

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d’arriver à concevoir un système d’information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques qui assure une longévité au modèle. En effet, l’agencement des données n’a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment. Merise a été très utilisée dans les années 1970 et 1980 pour l’informatisation massive des organisations. Cette méthode reste adaptée pour la gestion des projets internes aux organisations, se limitant à un domaine précis. Elle est en revanche moins adaptée aux projets transverses aux organisations, qui gèrent le plus souvent des informations à caractère sociétal (environnemental et social) avec des parties prenantes.

#### 2.3.2.2 UML

L’UML est un langage nous permettant de modéliser nos classes et leurs interactions. Autrement c’est un ensemble d’annotations graphiques s’appuyant sur des diagrammes et permettant de spécifier, visualiser et de documenter les systèmes logiciels orientés-objets. UML s’im-

pose aujourd’hui comme langage de modélisation objet standardisé pour la conception de logiciels. La version finalisée, UML2 largement enrichie et corrigée est celle que nous utiliserons.

**Choix :** Notre choix s'est porté sur le langage UML pour ses nombreux avantages. En effet, avec UML nous avons le concept des objets auxquels associés les principes d'héritages de composition et transition d'état.

### 2.3.3 Choix du SGBD

Un système de gestion de base de données (SGBD) est un logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations. Il permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données.

#### 2.3.3.1 PostgreSQL

PostgreSQL est un système de gestion de bases de données relationnel robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données, même dans des situations critiques. Il est de type SQL et extrêmement respectueux des standards, se conformant au plus près à la norme ANSI-SQL 2008.

### 2.3.4 MongoDB

MongoDB (de l’anglais humongous qui peut être traduit par « énorme ») est un système de gestion de base de données orientée documents, répartissable sur un nombre quelconque d’ordinateurs et ne nécessitant pas de schéma prédefini des données. Il est écrit en C++. MongoDB est développé depuis 2007 par MongoDB et fait partie de la mouvance NoSQL. MongoDB permet de manipuler des objets structurés au format BSON (JSON binaire), sans schéma préterminé. En d’autres termes, des clés peuvent être ajoutées à tout moment « à la volée », sans reconfiguration de la base.

**Choix :** Notre choix s'est porté sur MongoDB. La raison principale de notre orientation est que pour stocker les données de traitement de notre système nous avons utilisé une base de données NoSQL “Not only SQL” pour sa scalabilité horizontale et sa capacité à permettre la gestion de volumes de données importants. NoSQL permet également des traitements plus rapides pour certains types d’applications.

### 2.3.5 Choix des outils de programmation

Dans un premier temps nous parlerons des choix réalisés pour le développement de l'application mobile. Pour le développement de MédiCal, nous avons eu les choix suivants :

- Développer une application android native qui désigne le type d'applications conçues spécifiquement pour une plateforme (un type d'OS mobile) grâce au kit de développement appelé Software Development Kit qui lui est propre. Le SDK d'Apple se nomme iOS 5 SDK et celui de Google se nomme Android SDK. Une application mobile développée en utilisant ce langage spécifique sera proposée uniquement sur la boutique en ligne associée au SDK. On ne peut pas consulter ou installer une application iPhone sur un BlackBerry et vice versa (idem vis-à-vis d'Android).
- Développer une application web mobile (Web Apps) qui, en opposition à une application native, est une application mobile développée avec les outils de développement web mais en la rendant utilisable sur mobile . C'est un site web adapté au mobile. C'est une application qui, une fois développée, est accessible et exécutable sur tous les smartphones via leur navigateur web
- Développer une application hybride. Il s'agit ici du type d'applications qui combine les éléments d'une WebApps et les éléments d'une application native. Elles servent de passerelle entre les deux langages de développement. Cette solution nous permet d'utiliser un seul et même outil pour le développement et les langages issus du développement Web pour toutes les plateformes mobiles (iOS, Android et Windows Phone). Le développement est ainsi mutualisé.

**Choix :** De l'analyse de ces trois possibilités, nous avons choisi de faire une application mobile native parce que le cahier des charges demandait une application mobile sur android mais aussi parce que ce modèle offre plus de richesses en termes de fonctionnalités, de qualité, de performance, d'ergonomie et permet des applications accessibles hors connexion, facile à distribuer notamment via le Play Store. Pour développer notre application mobile native sous android, nous avons utilisé conformément aux spécifications de Google (supporteur officiel d'android) le langage de programmation Java notamment sa version 8 ainsi que le SDK (Software Development Kit ) fournis dans l'environnement Intégré Android Studio.

Dans un second temps, nous parlerons des choix réalisés pour le développement de l'application web. Pour le développement de MédICIS :

- **Node.js**

Node.js est un environnement d'exécution de JavaScript côté serveur. Il est bâti sur le moteur V8 JavaScript de Chrome V8. Il est plus scalable, léger, efficient et performant que le langage Php. Il est également rapide de programmer en Node.Js.

- **Sails.js**

Sails.js est un framework web qui facilite la création d'applications Node.js personnalisées. Il est conçu pour ressembler à l'architecture MVC, mais avec un support pour le développement de l'API et des applications Web plus modernes et orientées vers les données. C'est particulièrement bon pour créer des fonctionnalités en temps réel.

- **Bulma**

Bulma est un framework CSS qui se démarque des autres parce qu'il est basé sur les flexbox pour faciliter la disposition des composants. Aussi, les noms des classes sont plus intuitifs. Il n'y a pas de JavaScript. En effet les développeurs veulent généralement utiliser leur propre code JS (et généralement en possèdent déjà un). Il est plus facile à manier comparé aux autres framework comme Bootstrap et Materialize.

- **Vue.js**

Vue.js est une librairie Javascript qui permet de créer des interfaces web interactives. C'est aussi un framework très rapide dont la vitesse d'exécution est facilitée par sa façon de rendre les vues de façon asynchrone, afin d'éviter des rendus inutiles au milieu de longues opérations. La bibliothèque est principalement concentrée sur la couche de vue et est très facile à intégrer avec d'autres bibliothèques ou des projets existants.

Nous terminerons cette partie par la présentation de **Git** un outil qui a été au cœur de notre travail de développement. Pour une bonne continuité dans l'implémentation et pour prévenir le risque que survienne un problème dans l'implémentation ou une perte des données du disque dur, nous avons besoin d'un gestionnaire de version. Notre choix s'est donc porté sur Git qui est un logiciel de gestion de version décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux et distribué selon les termes de la licence publique générale GNU version 2. En 2017, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes (d'après Github). Comme Dépôt en ligne, nous avons utilisé la version communautaire de Gitlab avec des dépôts privés.

### 2.3.5.1 Services Cloud

Pour la réalisation du projet, nous avons eu à utiliser en plus des services que nous avons conçus, deux services extérieurs. Il s'agit de Twilio et de Firebase.

Twilio propose une infrastructure et un ensemble d'APIs qui permettent aux développeurs d'applications et services d'utiliser facilement et programmatiquement les services de Téléphonie. Nous avons eu donc à utiliser Twilio sms pour envoyer des messages aux utilisateurs.

FireBase est un ensemble de services d'hébergement pour n'importe quel type d'application mobile, web ou desktop. Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification, de l'intégration sociale (Google, Facebook, Twitter, Github) et

de pousser des notifications. Pour l'utiliser, les applications intègrent une bibliothèque (SDK) qui permet les diverses interactions possibles. Firebase fait partie d'une offre cloud récente appelée BaaS (Backend as a Service). Pour notre travail, nous avons utilisé 3 modules de firebase :

le module Crash Report nous envoie un email à chaque fois qu'un bug ou un dysfonctionnement survient dans l'usage de l'application mobile sur le téléphone des utilisateurs finaux. le module analytics quant à lui nous permet de monitorer sur un dashboard l'ensemble des interactions des utilisateurs avec l'application mobile. Cela nous permet de pouvoir améliorer l'application parce que l'on connaît exactement le temps que l'utilisateur passe à un endroit donné de l'application mais aussi les fonctionnalités qu'ils utilisent fréquemment et ceux qu'ils n'utilisent pas. FireBase Auth qui est un module nous permettant de sécuriser notre backend firebase en limitant l'accès qu'à notre application uniquement.

### 2.3.6 Choix d'architectures logicielles

Tout au long de notre projet, nous avons utilisé des standards d'architecture logicielle. Il s'agit de :

#### 2.3.6.1 Architecture RESTFUL

RESTful est une architecture créée par Roy Fielding basé sur le protocole HTTP permettant de construire des applications (Web, Intranet, Web Service). Il s'agit d'un ensemble de conventions et de bonnes pratiques à respecter. Elle utilise les méthodes GET / POST /PUT / DELETE et les URIs comme représentant des ressources. Dans cette architecture REST, nous avons choisi le JSON comme format d'achat de données pour sa syntaxe souple qui favorise la rapidité des échanges.

#### 2.3.6.2 Patron de conception MVP

Le Modèle-Vue-Présentation (en abrégé MVP, de l'anglais model-view-presenter) est un patron d'architecture, considéré comme un dérivé du patron d'architecture modèle-vue-contrôleur. Il garde les mêmes principes que MVC sauf qu'il élimine l'interaction entre la vue et le modèle qui sera effectuée par le biais de la présentation, qui organise les données à afficher dans la vue. Nous l'avons utilisé pour développer l'application mobile android afin de rendre le code plus maintenable.

### 2.3.7 Sécurité de l'application

Le système que nous avons développé traite des informations sensibles des établissements de santé, de leurs personnels, de leurs patients. A partir du moment où nous faisons transiter ces informations entre différents réseaux interconnectant différentes applications, nous devons prendre des mesures sécuritaires pour réduire autant que possible les failles de sécurité. Nous

classons ci-dessous les mesures que nous avons adoptées en fonction des principes de sécurité informatique :

- **La confidentialité** La confidentialité est le fait de s'assurer qu'une information est accessible uniquement par les entités qui ont le droit d'accéder à celle-ci. Nous avons en effet un système de gestion de rôle et de droit intégré qui nous permet de respecter la confidentialité des données dans chaque sous-système. En effet, chaque utilisateur appartient à un groupe d'utilisateurs donné et pour chaque groupe seulement un ensemble d'actions bien définies est possible.
- **L'Intégrité** L'intégrité s'assure que la donnée reste toujours intègre c'est-à-dire qu'elle n'a pas été modifiée par un tiers non autorisé. Ce principe de sécurité est pris en compte par la mise en place d'un certificat ssl sur le serveur d'applications.
- **La disponibilité** La disponibilité est le fait de s'assurer que l'information soit toujours disponible peu importe le moment choisi. Elle a été assurée en partie par l'infrastructure mise en place comme des serveurs de redondance mais aussi par la virtualisation des ressources et des applications qui ont été déployées.
- **L'Authenticité** Au sein d'un système d'information, il est important de vérifier l'authenticité de chaque ressource. Cela est possible grâce au mécanisme d'authentification qui permet de prouver l'identité d'une personne via le processus d'identification. A notre niveau, nous avons mis cela en place par une forte authentification à deux niveaux. Dans un premier temps, avant de faire appel à une quelconque URI, il faudrait être authentifié avec un couple d'identifiants comme service tiers de confiance. Donc nos applications clientes s'authentifient avant de demander un quelconque service au Backend. Après être autorisées, les applications clientes ont le devoir d'authentifier leurs utilisateurs avant de pouvoir faire une opération (lecture ou écriture) mettant en relation un utilisateur du système.
- **Imputabilité** Il s'agit des techniques mises en œuvre pour assurer la traçabilité des actions d'un individu sur un système. Nous avons mis en place cette traçabilité en mettant en place une journalisation de toutes les actions menées sur les objets du système.

**Conclusion partielle** Dans ce chapitre, nous avons précisé la méthodologie de gestion de projet que nous avons utilisée. Nous avons fait une analyse et réalisé les diagrammes de cas d'utilisation et de séquences. Ceci nous a permis de définir l'architecture de notre système et de choisir convenablement les outils techniques.

# Chapitre 3

## Présentation du prototype de l'application et discussion

### 3.1 Présentation des interfaces de l'application mobile de gestion de rendez-vous

MédiCal se présente en trois parties. Nous distinguons le panel d'authentification qui permet à l'utilisateur de s'inscrire lors de la première utilisation et de se connecter afin d'accéder aux autres modules de l'application. Les interfaces ci après la présentent :

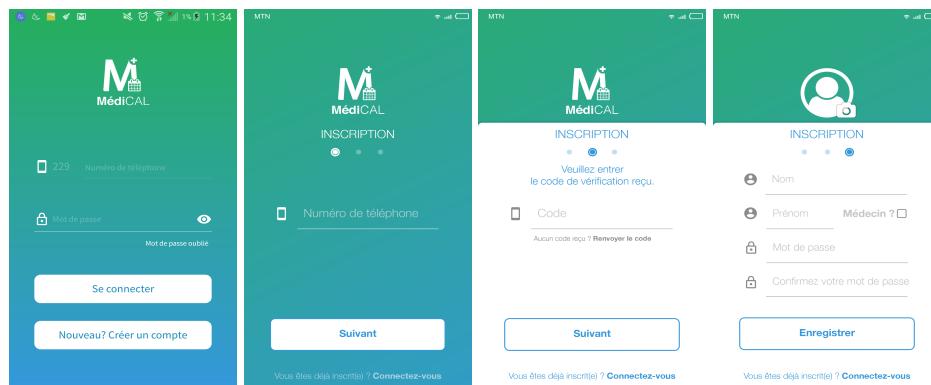


FIGURE 3.1 – Interfaces de connexion et d'inscription

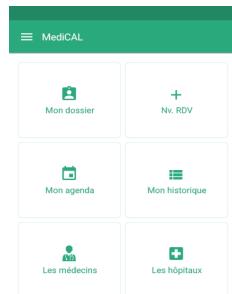


FIGURE 3.2 – Interfaces d'accueil du patient

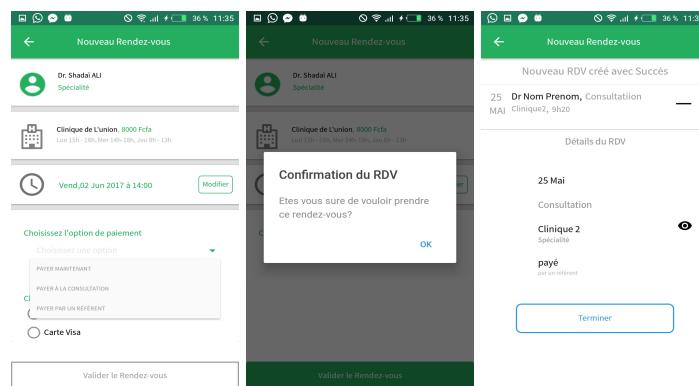
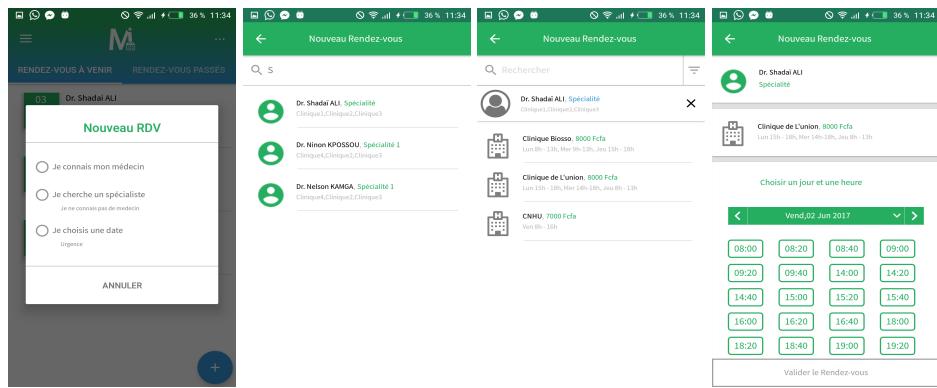


FIGURE 3.3 – Interfaces de prise de rendez-vous

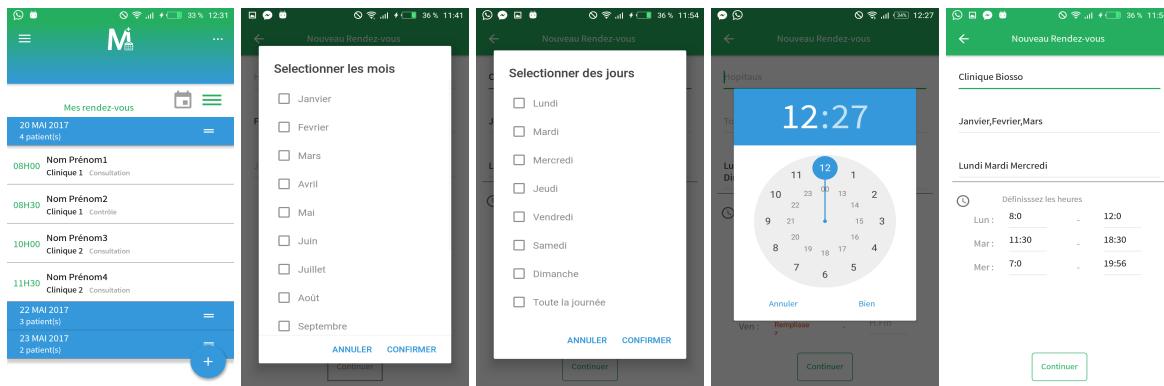


FIGURE 3.4 – Interfaces du médecin

## 3.2 Présentation des interfaces de l'application web de gestion des dossiers médicaux

---

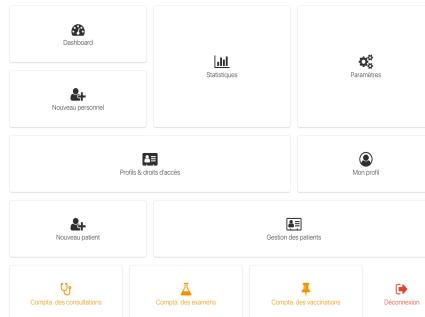


FIGURE 3.5 – Interfaces d'accueil de MéDICIS

The screenshot shows the 'Ajouter un patient' (Add patient) page. The left sidebar has a green header 'MÉDICIS' and a 'Ajouter' button under the 'PATIENTS' section. The main form is titled 'Ajouter un patient' and contains two sections:

- Informations personnelles:** Fields for Civilité (Civil status), Situation maritale (Marital status), Nom (Name), Prénom (First name), Email (Email), and Téléphone (Phone number).
- Adresse:** Fields for B.P (Building number), Rue (Street), Carré / Parcelle, Maison (Plot / House), Quartier (Neighborhood), and Ville (City).

FIGURE 3.6 – Interface d'ajout d'un nouveau patient

FIGURE 3.7 – Interface d'ajout d'un nouveau personnel

FIGURE 3.8 – Interfaces de consultation

### 3.3 Discussion

La solution développée connaît certaines insuffisances au nombre desquelles nous avons :

- la nécessité pour les utilisateurs d'être connectés à internet
- la non disponibilité de l'application mobile sur toutes les plateformes

#### Conclusion partielle

Cette étape nous a permis de présenter le prototype développé. Nous avons également sou-

levé dans ce chapitre les insuffisances de notre prototype.

# Conclusion

Le présent rapport est le résultat de notre stage effectué dans le cadre de la réalisation de notre projet de fin d'études de Licence en Génie Logiciel au sein de l'entreprise OpenSI. Au cours de ce stage de 3 mois, nous avons pu mettre en pratique nos connaissances théoriques acquises durant notre formation.

Notre projet de fin d'année avait donc pour but de mettre en place une application mobile de gestion des prises de rendez-vous dans les établissements de santé et une application web pour la gestion des dossiers médicaux de santé.

Nous avons, tout d'abord, entamé notre étude par l'analyse qui est une étape cruciale et nécessaire pour mieux assimiler le système déjà existant, puis par la définition des principaux intervenants et l'identification des besoins. Elle nous a permis la conception d'une architecture de base stable. Enfin, nous avons procédé au choix justifié des technologies avant de passer à l'implémentation, qui nous a permis de développer ces applications en tenant compte de l'architecture matérielle, de l'environnement logiciel et des contraintes techniques de l'entreprise.

De ce qui précède, il est difficile de prétendre avoir eu une solution idéale. Toutefois, nous espérons avoir répondu tant soi peu à notre problématique. Cependant, nous sommes conscientes des améliorations à faire. Nous avons prévu de travailler sur des algorithmes de compression de données afin d'améliorer les temps d'échanges sur des connexions à faibles débits. Aussi, pour la messagerie instantanée nous prévoyons de réaliser des chatbots (agents de conversation virtuel) qui auront l'intelligence nécessaire pour répondre automatiquement aux patients afin de faire un pré-diagnostic et orienter celui-ci dans le choix d'une consultation à base de prédiction effectué sur l'ensemble des données médicales que nous aurions collectées.

# Bibliographie

- [1] Fien VAN DER HEYDE-Laurent DEBRAUWER *initiation, exemples et exercices corrigés(4e édition)*
- [2] Jean Luc Baptiste *Modélisation des données et traitements, manipulation avec le langage SQL(2e édition)*, O'Reilly Media, 462 p,
- [3] Kyle Banker *MongoDB in Action*, Manning Publications Co.Greenwich, CT, USA ©2011  
Manning Publications Co. Greenwich, CT, USA ©2011
- [4] Danny B. Lange, Oshima Mitsuru *Programming and Deploying Java Mobile Agents Aglets*, 1st edition Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©1998 ISBN :0201325829
- [5] Rick Rogers, John Lombardo, Zigurd Mednieks, Blake Meike *Android Application Development : Programming with the Google SDK*, 1st edition O'Reilly Media, Inc. ©2009 ISBN :0596521472 9780596521479
- [6] Ronan CHARDONNEAU *Concevez des applications Android pour mobile*,
- [7] Julien FONTANET, Olivier LAMBERT *Exploitez la puissance de JavaScript côté serveur*, Date de parution : mars 2015 ISBN : 978-2-7460-8978-5
- [8] Pedro Teixeira *Building Javascript Based Scalable Software*,

# Webographie

- [1] <https://www.mongodb.com/what-is-mongodb/>, consulté le 21-03-2017.
- [2] <https://fr.wikipedia.org/wiki/NoSQL>, consulté le 21-03-2017.
- [3] <http://www.abbeal.com/2016/09/13/vue-js/>, consulté le 03-04-2017.
- [4] <http://www.abbeal.com/2016/09/13/vue-js/>, consulté le 03-04-2017.
- [5] <https://www.drupal.org/project/bulma>, consulté le 21-03-2017.
- [6] <https://github.com/jgthms/bulma>, consulté le 21-03-2017.
- [8] [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer), consulté le 21-03-2017.
- [9] <http://developer.android.com>, consulté le 20-03-2017.
- [10] <http://cian.developpez.com/uml2/>, consulté le 16-03-2017.
- [11] <https://github.com/jgthms/bulma>, consulté le 20-05-2017.

