

nytnyt



UNIVERSITÉ DE NOUAKCHOTT AL-AASRIYA  
INSTITUT SUPÉRIEUR DE COMPATIBILITÉ ET  
D'ADMINISTRATION DES ENTREPRISES

MÉMOIRE DE FIN D'ÉTUDES

Présenté par  
Mohamed Saleck Lebchir  
Pour obtenir le diplôme de  
Master Professionnel de l'ISCAE

Spécialité :  
Informatique Appliquée à la Gestion

---

**Conception et simulation d'un système  
d'échange de clés S-SEJAD**

---

**Président :** Hamidou DATHE, Professeur, Université Cheikh Anta Diop

**Rapporteurs :** .... ...., .....

.... ...., .....

**Examinateurs :** .... ...., .....

.... ...., .....

**Co-endadrant :** Khadidiatou WANE KEITA, Maître de Conférences , Université Cheikh Anta Diop

**Directeur de thèse :** Khaly TALL, Professeur, Université Cheikh Anta Diop

# Table des matières

---

<b>1</b>	<b>Introduction générale</b>	<b>VII</b>
1.1	Motivations . . . . .	VII
1.2	Problématiques . . . . .	VII
1.3	Objectifs . . . . .	VII
<b>2</b>	<b>Présentation de la société</b>	<b>1</b>
2.1	Introduction . . . . .	1
2.2	Missions . . . . .	1
2.3	Organigramme . . . . .	2
<b>3</b>	<b>Environnement de travail</b>	<b>3</b>
3.1	Gestion du projet . . . . .	3
3.1.1	Méthodologie de travail . . . . .	3
3.1.2	Logiciel de gestion du projet : Trello . . . . .	7
3.2	Conception . . . . .	7
3.2.1	Langage de modélisation : UML . . . . .	7
3.2.2	Logiciel de modélisation . . . . .	8
3.3	Implémentation . . . . .	8
3.3.1	Front-end . . . . .	9
3.3.2	Back-End . . . . .	10
3.3.3	Motifs d'architecture logicielle . . . . .	10
3.4	Sécurité . . . . .	12
3.4.1	Hachage des mots de passe . . . . .	13
3.5	Autres . . . . .	13
3.5.1	Environnement de rédaction : MikTex & TexStudio . . . . .	13
3.5.2	Logiciel de sauvegarde et de partage : Dropbox . . . . .	14
<b>4</b>	<b>Analyse fonctionnelle et conceptuelle</b>	<b>15</b>
4.1	Analyse fonctionnelle . . . . .	15
4.1.1	Diagramme de cas d'utilisation . . . . .	15
4.1.2	Diagrammes d'activité . . . . .	17
4.1.3	Diagrammes de séquence . . . . .	19
4.2	Modélisation de la base de données . . . . .	20
4.2.1	Diagramme de classes . . . . .	20
<b>5</b>	<b>Implémentation</b>	<b>22</b>
5.1	Architecture de l'application . . . . .	22
5.2	Services web . . . . .	23

5.3	Vue sur l'application . . . . .	23
5.3.1	Options d'internaute . . . . .	23
5.3.2	Espace Patient . . . . .	24
5.3.3	Espace Consultant Médical . . . . .	25
<b>6</b>	<b>Conclusion et perspectives</b>	<b>30</b>
<b>7</b>	<b>Annexe 1 : Génération de l'App Bundle et l'APK en mode release</b>	<b>32</b>
<b>8</b>	<b>Annexe 2 : Déploiement des Web API et de Base de données sur Azure</b>	<b>34</b>
<b>9</b>	<b>Annexe 3 : Publication d'application sur Google Play Store</b>	<b>36</b>
	<b>Bibliographie</b>	<b>39</b>

# Table des figures

---

2.1	SMPNT . . . . .	1
2.2	Organigramme de SMPNT . . . . .	2
3.1	Articulation générale de la méthode Scrum . . . . .	5
3.2	Rôle des trois couches logicielles du motif MVC . . . . .	12
4.1	Diagramme de cas d'utilisation d'utilisateur . . . . .	16
4.2	Diagramme de cas d'utilisation du patient . . . . .	17
4.3	Diagramme de cas d'utilisation du consultant médical . . . . .	17
4.4	Diagramme d'activité : Création de compte . . . . .	18
4.5	Diagramme d'activité : Demande de rendez-vous . . . . .	19
4.6	Diagramme d'activité : Ajournement de rendez-vous . . . . .	19
4.7	Diagramme de séquence : Création de compte . . . . .	20
4.8	Diagramme de séquence : Demande de rendez-vous . . . . .	21
4.9	Diagramme de classes. . . . .	21
5.1	Architecture logicielle de l'application. . . . .	22
5.2	Interfaces d'accueil pour l'internaute. . . . .	24
5.3	Interfaces d'accomplissement des informations de patient. . . . .	25
5.4	Interfaces des options de navigation et paramétrage. . . . .	26
5.5	Interfaces de demande de rendez-vous. . . . .	27
5.6	Interfaces d'accomplissement des informations de consultant médical. . . . .	28
5.7	Interfaces des options de navigation et de paramétrage. . . . .	28
5.8	Interfaces d'approbation et refus d'une demande de rendez-vous. . . . .	29
5.9	Interfaces de mise à jour d'emploi du temps. . . . .	29
7.1	Choix de génération et génération d'un nouveau Key Store. . . . .	32
7.2	Aperçu sur la configuration de l'APK ou de l'App Bundle. . . . .	33
7.3	Options de génération de l'App Bundle ou de l'APK en mode release. . . . .	33
8.1	Création d'une base de données sur Azure. . . . .	34
8.2	Génération du chaîne de connexion. . . . .	35
8.3	Creation de l'application web. . . . .	35
9.1	Publication sur Google Play Store. . . . .	36
9.2	Publication sur Google Play Store 2. . . . .	37
9.3	Publication sur Google Play Store 3. . . . .	38

# Liste des tableaux

---

3.1 Rôles des diagrammes UML utilisés.	8
3.2 Contexte d'utilisation des différents langages utilisés.	9
3.3 Rôle des trois couches logicielles du motif MVC.	11
4.1 Modules de l'application.	16
5.1 Rôles des différents services web de l'application.	23
6.1 Bilan rétrospectif sur les fonctionnalités implémentées.	30

# Introduction générale

---

La mise en relation entre les patients et les consultants médicaux consiste à collecter des informations sur eux, plus précisément leurs horaires disponibles pour faire des consultations aux maisons des patients. Comme ça nous offrons aux patients la possibilité de savoir les consultants médicaux disponibles à chaque instant.

Le fait de lier le patient et le consultant médical, a un double avantage, d'une part, pour le patient, car il va il peut savoir facilement des consultants médicaux disponibles, et d'autre part, pour le consultant, car il va obtenir un travail supplémentaire. Comme l'intitule du mémoire l'indique, notre Objectif est justement de lier les deux pôles de la problématique (Patient et Consultant médical).

## 1.1 Motivations

La demande de services médicaux devient de plus en plus importante en Mauritanie en tenant compte ceux qui nécessitent des visites à domicile. Les rendez-vous sont toujours organisés par des appels téléphoniques ou via les applications de messagerie, ce qui les rendent plus lents en terme d'organisation dû à l'existence d'un nombre peu important des fournisseurs connus de ces services ce qui n'est pas expliqué par la rareté de ces fournisseurs de façon général.

## 1.2 Problématiques

En réalité, la réalisation d'une application, fournissant les services sanitaires nécessaires, nécessite de faire face à des problématiques diverses et complexes. Ainsi, la société a décidé de se contenter, dans un premier temps, de la thématique de gestion des rendez-vous médicaux a domicile. Ce sujet soulève de nombreuses questions aux implications différentes. Tout d'abord que signifie un « Consultant médical » ?, le patient pourrait-il rechercher une consultant médical disponible ? Quelle procédure le patient doit-il suivre pour demander une consultation ? Comment pourrait-il payer les frais de consultation ? Comment pourrait-il reporter une consultation ?

## 1.3 Objectifs

La mise en relation entre les fournisseurs de services et les consommateurs (clients de façon générale) est un des services offerts par SMPNT. La société veut offrir un meilleur service aux clients à l'aide d'une application mobile qui gère la relation entre les patients et les consultants médicaux. Autrement dit, l'application permet d'organiser des rendez-vous à domicile.

# Présentation de la société

---

Il s'agit là de présenter de manière globale la structure de la Société Mauritanienne pour les Nouvelles Technologies(SMPNT) en premier, pour ensuite dérouler ses missions et domaines d'activités, et enfin exposer son organisation interne.

## 2.1 Introduction



FIGURE 2.1 – SMPNT

Société Mauritanienne Pour les Nouvelles Technologies (SMPNT) est une société Mauritanienne spécialisée dans les domaines des ingénieries informatiques, fondée au début de l'été 2018. SMPNT propose un ensemble de services autour des technologies orientées objets, des architectures orientées services (SOA) ou de l'intégration d'applications d'entreprise (EAI). SMPNT met à la disposition de ses clients des solutions éprouvées pour la réalisation d'applications. Elle intervient également d'un point de vue organisationnel, méthodologique et technique depuis la phase de conseil jusqu'à la mise en œuvre de la solution tout en assurant l'accompagnement des équipes clientes dans leur montée en compétences.

## 2.2 Missions

SMPNT offre une large palette de prestations organisées autour des activités suivantes :

1. Conception et Développement complet des solutions SI personnalisées
2. (Client/serveur, Intranet, Web)
3. Vente matériel informatique
4. Maintenance

5. Assistance technique sur l'ensemble des produits EBP Gestion
6. Audit orienté système d'information et de gestion
7. Intégration et implémentation d'applications de gestion pour l'entreprise
8. Développement et mise en place de solution de gestion des documents électroniques GED.

### 2.3 Organigramme

La structure organisationnelle de SMPNT comprend :

1. Une direction générale qui coiffe toutes les unités de la société et qui définit la vision stratégique à adopter en vue de pleinement remplir leurs missions.
2. Un assistanat de direction qui se charge d'assister et d'aider la direction générale dans sa politique de bonne gestion de la société
3. Une comptabilité
4. Une équipe commerciale
5. Une équipe technique et fonctionnelle chargée de la mise en œuvre des missions de SMPNT et constituée
6. D'une équipe fonctionnelle dont le rôle est de fournir à l'équipe technique toute la logique métier pour parvenir à la réalisation des produits
7. D'un team leader dont la mission est de proposer et concevoir des architectures de projet en collaboration avec l'équipe fonctionnelle et l'équipe de développeurs
8. Des développeurs qui se chargent de la réalisation des solutions logicielles

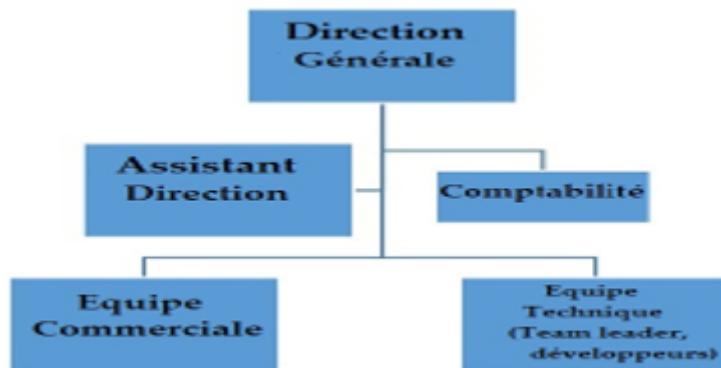


FIGURE 2.2 – Organigramme de SMPNT

# Environnement de travail

---

L'objectif de ce chapitre consiste à expliciter tous les éléments relatifs à l'environnement logiciel sur lequel j'ai travaillé pour réaliser cette application. Il s'agit donc des logiciels, langages, frameworks, et motifs d'architecture auxquels j'ai fait recours tout au long du processus de mise en œuvre de ma solution.

## 3.1 Gestion du projet

Dans cette section, je présente la méthodologie de travail et le logiciel de gestion de projet que j'ai utilisé pour déclencher, organiser et gérer le travail sur ma solution.

### 3.1.1 Méthodologie de travail

C'est la manière de mener un processus de développement. Il s'agit d'une démarche, un ensemble d'étapes ou procédures à mettre en œuvre dans une logique méthodologique, accompagnés d'outils et de techniques. L'utilisation d'une méthode est incontournable dans l'entreprise de tout projet, particulièrement dans la réalisation de projets informatiques. Dans ce cas-ci, les méthodes utilisées sont des méthodes d'analyse et de conception qui ont pour but la formalisation des étapes préliminaires au développement de systèmes logiciels, en d'autres termes : analyse, modélisation et conception. L'urgence de l'utilisation de ces méthodes trouve son explication dans un certain nombre de facteurs :

1. De nombreux échecs de projets informatiques dans le passé dus à un manque d'organisation, ou une non satisfaction des besoins ;
2. La révolution de l'industrie logicielle engendrée par les échecs informatiques et qui introduit de nouveaux facteurs de validation de la qualité logicielle : le génie logiciel ;
3. Les nombreuses exigences liées au coût, aux délais et à la complexité des projets informatiques.

L'utilisation de méthodes de développement de logiciels permet ainsi l'élaboration de systèmes informatiques de manière fiable et viable tout en répondant à l'ensemble des exigences du client et du génie logiciel.

Il existe plusieurs méthodes de développement informatique. L'on distingue deux approches : l'approche traditionnelle et l'approche agile. Les deux approches se distinguent essentiellement dans la manière de décomposer le projet. Les méthodes cartésiennes ou fonctionnelles ou encore traditionnelles se sont imposées les premières.

### 3.1.1.1 L'approche traditionnelle

Cette approche s'inspire directement de l'architecture des ordinateurs. Les méthodes traditionnelles prônent une démarche strictement planifiée avec une séquence d'activités bien définie. La succession des activités et le planning doivent être clairement respectés et aucun changement n'est permis. Il est attendu du client une spécification des besoins globale, détaillée, claire, précise et validée en entrée. Ainsi, tout doit être prévisible, du début du projet à la livraison du produit, d'où l'appellation de méthodes prédictives.

Selon le planning adopté, les méthodes cartésiennes proposent plusieurs modèles d'exécution des activités du projet :

1. Le **modèle en cascade** : dans ce modèle, le processus de développement est découpé séquentiellement et de façon linéaire selon les activités intrinsèques du cycle de vie du développement logiciel : l'analyse, la conception, le codage et les tests. Le plan de déroulement des phases (planification prédictive) est élaboré en tout début de processus. Le passage à une phase donnée n'est fait que si le résultat de la phase précédente a été validé et jugé satisfaisant par le client et les utilisateurs.
2. Le **modèle en V** : Le cycle en V est à la base de tout développement logiciel, il en représente les activités intrinsèques. Il tient d'avantage compte de la réalité que le modèle en cascade, le processus de développement n'est pas réduit à un enchainement de tâches séquentielles. Le modèle en V permet d'anticiper sur les phases ultérieures de développement du produit en particulier les plans de test de qualifications et de performance.

Parmi les méthodes traditionnelles, nous pouvons citer : SADT, CORIG, ...

### 3.1.1.2 L'approche agile

Cette approche est définie par les concepts suivants : la simplicité, la légèreté, la souplesse, un lien fort avec le client. C'est dans cette optique que certains apparentent le développement agile aux notions de flexibilité, de rétroaction et d'adaptation au changement rapide et continu.

Une méthode agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients et en anticipant sur les risques. Il y'a continuellement des aller et retour avec le client. L'application logicielle est livrée par version incrémentale. Les versions successives sont aussi fiables que le livrable final en termes de tests et de validation. En quelque sorte le processus est déroulé comme un enchaînement de « mini-cascades ». A chaque nouvelle itération, l'ensemble de l'architecture et de la conception logicielle est reconstruit, le code est retravaillé.

Les méthodes agiles aspirent donc à améliorer la réactivité et l'adaptabilité des sociétés de logiciels et constituent un moyen de survie dans un environnement instable en s'accompagnant des valeurs suivantes :

1. Les individus et les interactions plutôt que les processus et les outils ;
2. L'application fonctionnelle plutôt que la documentation compréhensive ;
3. La collaboration avec le client plutôt que la négociation des contrats ;
4. La réponse au changement plutôt que le suivi d'un plan.

L'agilité comprend plusieurs courants de pensée qui ont conduits à des méthodes différentes, reposant sur les mêmes concepts mais présentant des singularités. Les méthodes Scrum, Kanban, et XP (eXtreme Programming) sont des exemples de ces méthodes.

### La méthode SCRUM

**Scrum** est une méthode agile de gestion de projet qui permet de produire la plus grande valeur métier dans la durée la plus courte. Elle a pour objectif d'améliorer la cohésion de l'équipe et la rapidité du processus de développement. Le nom Scrum renvoie à une pratique généralement connue au rugby signifiant la « mêlée ».

Cette méthode qualifie un ensemble de rôles, d'instruments de gestion et de pratiques managériales favorisant un environnement basé sur la transparence, l'inspection, le suivi et l'adaptation. Le cycle de vie d'un projet Scrum peut être découpé en trois parties :

1. Phase d'**initiation ou démarrage** : il s'agit d'une phase linéaire où l'on définit le périmètre fonctionnel du système et la liste des fonctionnalités (**Backlog**) agencées par ordre de priorité, d'effort, de complexité et de risque. C'est aussi à ce niveau que l'architecture est définie.
2. Phase de **développement** est un processus empirique : le projet est découpé en cycles itératifs d'une durée de deux semaines ou **sprints**. Chaque sprint regroupe une ou plusieurs fonctionnalités du Backlog. Tout au long de cette phase, le travail réalisé est mesuré et contrôlé et une amélioration constante du prototype est faite.
3. Phase de **Clôtures** est une phase linéaire de gestion de la livraison du produit final.

La figure 3.1 montre l'articulation générale de Scrum.

Les responsabilités managériales sont réparties sur trois rôles fondamentaux :

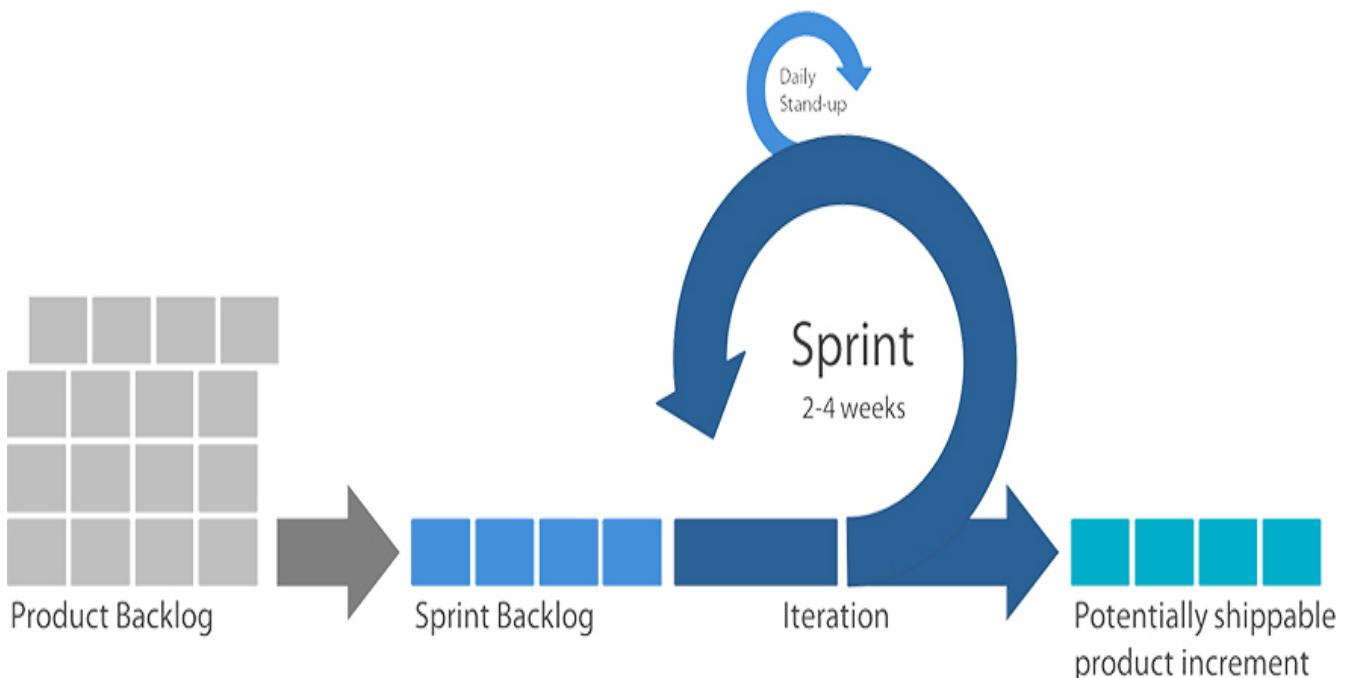


FIGURE 3.1 – Articulation générale de la méthode Scrum

1. Scrum Master
2. Product Owner
3. Équipe Scrum

Les artéfacts et pratiques de Scrum

1. **Product Backlog** : état courant des tâches à accomplir ;
2. **Sprint** : itération de deux semaines ;
3. **Effort-Estimation** : permanente sur les entrées du Backlog ;
4. **Sprint Backlog** : Product Backlog limité au sprint en cours ;
5. **Daily Scrum Meeting** : ce qui a été fait, ce qui reste à faire ;
6. **Sprint Review Meeting** : Présentation des résultats du Sprint. [3.1](#)

## SCRUM contre KANBAN

Scrum est plus prescriptif que Kanban, qui évite de définir les rôles et les équipes et qui n'a pas de structure formelle de réunions. Kanban ne prescrit pas non plus d'itérations – bien qu'elles puissent être incorporées si vous le souhaitez.

Les techniques de visualisation des processus de Kanban le rendent idéal pour les équipes colocalisées qui travaillent sur un backlog d'éléments sujets à des changements fréquents (par exemple, Kanban est souvent utilisé par les équipes de support).

Le tableau Kanban est cependant souvent adopté par les équipes Scrum sous la forme d'un tableau de tâches et est utilisé pour suivre la progression tout au long d'un sprint.

La limite de la règle Work In Progress dans Kanban la rend également adaptée aux équipes ayant des ressources limitées ou lorsque l'entrée de chaque membre est requise sur chaque élément. Cela pourrait s'appliquer, par exemple, à une équipe de communication au sein d'une grande organisation.

Alors que Scrum limite la quantité de travail dans chaque sprint, la charge de travail est déterminée par l'estimation relative de la taille de chaque histoire (en points) et est approuvée par l'équipe Scrum à chaque session de planification.

Alors qu'une équipe Kanban suit le « temps de cycle » et optimise les délais d'exécution aussi courts et prévisibles que possible, une équipe Scrum vise à améliorer son rendement sur les sprints successifs et à améliorer la « vélocité » de l'équipe (le nombre de points d'estimation relatifs complétés dans un sprint). Cela rend sans doute Scrum plus adapté à la mise à l'échelle – il semble certainement plus familier et prévisible, ce qui peut être rassurant pour les grandes organisations.

## SCRUM contre XP

Dans Scrum, les équipes et les réunions sont assez gravées dans le marbre<sup>1</sup> alors que la question de

---

1. Dans l'antiquité, les engagements pour la constructions de bâtiments importants étaient gravés sur des plaques de marbre (Athènes : arsenal du Pirée, Delphes). Les travaux s'étendant sur de nombreuses années, on ne pouvait faire confiance aux tablettes de cire ou aux papyrus. Sur ces plaques, on définissait par exemple la grandeur du bâtiment ou le montant des amendes pour les retards. Ce qui n'était pas « gravé dans le marbre » n'était donc pas contractuel. Voir le lien <https://fr.wiktionary.org/wiki/graver-dans-le-marbre>

savoir comment le travail est réellement fait est laissée aux équipes pour décider elles-mêmes. XP, d'autre part, est livré avec un ensemble de pratiques de base qui pourraient sembler accablantes pour le débutant Agile.

On pourrait dire que Scrum est une méthodologie, qui est plus concernée par la productivité tandis que XP est plus préoccupé par l'ingénierie.

La valeur que les pratiques XP peuvent ajouter est incontestable et de nombreuses organisations qui utilisent Scrum adoptent la programmation par paires, le développement piloté par les tests et le refactoring comme des pratiques qui améliorent la qualité, accélèrent le processus de publication et / ou réduisent le besoin de revoir le travail en raison de la dette technique.

Outre les itérations plus courtes, d'autres éléments importants qui différencient XP de Scrum sont les suivants :

1. Les équipes XP travaillent sur des éléments dans un ordre de priorité strict alors qu'une équipe Scrum ne s'attaque pas nécessairement à chaque élément dans l'ordre de priorité une fois dans le sprint ;
2. Les équipes XP peuvent intégrer de nouveaux éléments de travail dans une itération et changer d'éléments de taille équivalente (tant qu'ils n'ont pas été démarrés) si le client décide d'une nouvelle priorité.

En termes de similitudes, le rôle du client dans XP est très similaire à celui du Product Owner dans Scrum – en ce sens qu'ils aident à écrire des user stories, à les hiérarchiser et sont toujours disponibles pour les développeurs – bien que moins bien définis.

Scrum et XP imposent tous deux une réunion debout quotidienne. Bien que les deux soulignent l'importance de la co-localisation, seul XP le rend décisif. Voir le site <https://manifesto.co.uk/kanban-vs-scrum-vs-xp-an-agile-comparison/>.

### 3.1.2 Logiciel de gestion du projet : Trello

Trello est un outil de gestion de projet en ligne, lancé en septembre 2011 et inspiré par la méthode Kanban. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement. Pour en savoir plus, veillez visiter le lien <https://fr.wikipedia.org/wiki/Trello>.

## 3.2 Conception

Dans cette section, je présente le langage et le logiciel de modélisation que j'ai utilisé pour concevoir notre solution.

### 3.2.1 Langage de modélisation : UML

[Unified Modeling Language. \(UML\)](#) est un langage de modélisation orientée objet permettant aux développeurs de modéliser un système d'information en considérant plusieurs vues chacune reflétant un aspect comportemental<sup>2</sup> ou structurel<sup>3</sup> du système.

---

2. Diagramme de cas d'utilisation, diagramme d'activité, diagramme de séquence, diagramme d'interaction, etc.

3. Diagramme de classe, diagramme de composants, diagramme de déploiement, diagramme de structure composite, etc.

En effet, nous avons opté pour UML au détriment de la Méthode d'Étude et de Réalisation Informatique par les Sous-Ensembles ou pour les Systèmes d'Entreprise. (MERISE) car nous avons besoin d'une approche de conception prenant en considération l'aspect orienté objet pour :

1. Pouvoir mettre le focus sur le rôle temporel des instances d'objets lors de déclenchement des actions (à travers le diagramme de séquence) ;
2. Faciliter par la suite la génération des classes Data Access Object. (DAO) à partir du diagramme de classes.

Certes, UML est très riche en matière de modélisation et propose au total 13 diagrammes chacun fournissant une vision particulière du système à concevoir. Dans notre contexte, je me suis limités à 4 diagrammes explicités sur le tableau 3.1.

### 3.2.2 Logiciel de modélisation

Modelio est un logiciel open source et multiplateforme permettant, entre autres, la modélisation UML et Business Process Model and Notation (BPMN). Pour en savoir plus, veuillez visiter le lien <https://www.modelio.org/about-modelio/features.html>.

Sans doute, les logiciels de modélisation UML sont nombreux, à savoir, Visual Paradigm, Eclipse Papyrus, StarUML, PowerDesigner, Umbrello, etc. Vu que les diagrammes UML que nous voulons réaliser sont disponibles dans tous ces logiciels, il n'y avait pas en effet un choix à argumenter car tous les choix étaient satisfaisants. Mais, de façon subjective, nous pouvons préciser que l'avantage de Modelio dans notre contexte est le fait que je m'y suis déjà habitués. Les diagrammes que nous avons réalisés avec Modelio sont ceux mentionnés ci-après.

TABLEAU 3.1 – Rôles des diagrammes UML utilisés.

Diagramme	Rôle
Diagramme de cas d'utilisation	Présenter les acteurs du système, ses fonctionnalités, les relations entre les acteurs et entre les fonctionnalités.
Diagramme d'activité	Déterminer l'enchaînement des différentes étapes qui composent une fonctionnalité du système.
Diagramme de séquence	Fournir une vue détaillée du diagramme d'activité en mettant le focus sur l'ordre chronologique et sur les objets créés et les méthodes appelées.
Diagramme de classe	Représenter la structure interne du système sous forme de classes et d'interfaces et préciser les différentes relations entre elles.

## 3.3 Implémentation

Dans cette section, je présente les langages, les logiciels, les frameworks et les motifs d'architecture que j'ai utilisé.

### 3.3.1 Front-end

#### 3.3.1.1 Editeur de texte : VS Code

VS Code (Visual Studio Code) est un éditeur de code source réalisé par Microsoft pour Windows, Linux et macOS. [9] Les fonctionnalités incluent la prise en charge du débogage, la coloration syntaxique, la saisie semi-automatique intelligente du code, les extraits de code, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. Pour en savoir plus, veillez visiter le lien <https://en.wikipedia.org/wiki/VisualStudioCode>.

#### 3.3.1.2 Languages

TABLEAU 3.2 – Contexte d'utilisation des différents langages utilisés.

Langage	Contexte d'utilisation
HTML	Création des pages web
SCSS : Version avancée du CSS	Modifier le style des pages
TypeScript	Utiliser pour tout traitement et exécution côté client

#### 3.3.1.3 Framework : IONIC

Ionic est un SDK open source complet pour le développement d'applications mobiles hybrides. La dernière version a été republiée en tant qu'ensemble de composants Web, permettant à l'utilisateur de choisir n'importe quel framework d'interface utilisateur<sup>4</sup>, tel que Angular, React ou Vue.js. Les applications mobiles peuvent être construites avec ces technologies Web, puis distribuées via des magasins d'applications natifs à installer sur les appareils en utilisant Cordova ou Capacitor<sup>5</sup>. Pour en savoir plus, veillez visiter le lien <https://en.wikipedia.org/wiki/Ionic>.

#### 3.3.1.4 Framework Web : Angular

Angular est un framework<sup>6</sup> côté client, open source, basé sur TypeScript. Il permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Pour en savoir plus, veillez visiter le lien <https://fr.wikipedia.org/wiki/Angular>.

#### 3.3.1.5 IDE : Android Studio

Android Studio est l'[Integrated Development Environment \(Environnement de Développement Intégré\)](#). (IDE) officiel pour le système d'exploitation Android de Google, construit sur le logiciel IntelliJ IDEA de JetBrains conçu spécifiquement pour le développement Android. Pour en savoir plus veillez

4. Framework Web

5. Ce sont des runtimes natifs multiplateformes qui facilitent la création d'applications Web modernes qui s'exécutent en mode natif sur iOS, Android et le Web.

6. En programmation informatique, un framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).

visiter le lien

<https://en.wikipedia.org/wiki/AndroidStudio>.

Essentiellement, je l'ai utilisé pour lancer l'application sur android.

### 3.3.2 Back-End

#### 3.3.2.1 IDE : Visual Studio

Microsoft Visual Studio est un [IDE](#) de Microsoft. Il est utilisé pour développer des programmes informatiques, ainsi que des sites Web, des applications Web, des services Web et des applications mobiles. Pour en savoir plus, veillez visiter le lien <https://en.wikipedia.org/wiki/MicrosoftVisualStudio>.

#### 3.3.2.2 Language : C#

C# (C sharp en anglais britannique) est un langage de programmation orientée objet destiné à développer sur la plateforme Microsoft .NET. Il est dérivé du C++ et très proche du Java dont il reprend la syntaxe générale ainsi que les concepts, y ajoutant des notions telles que la surcharge des opérateurs, les indexeurs et les délégues. Il est utilisé notamment pour développer des applications web sur la plateforme ASP.NET. Pour en savoir plus, veillez visiter le lien <https://fr.wikipedia.org/wiki/Csharp>. C# est utilisé dans cette application pour faire tout traitements back-end.

#### 3.3.2.3 Framework : .NET Framework

Le .NET Framework (prononcé "dot net") est un framework logiciel développé par Microsoft qui fonctionne principalement sur Microsoft Windows. Pour en savoir plus, visiter le lien <https://en.wikipedia.org/wiki/.NETFramework>.

#### 3.3.2.4 Serveur Web : IIS

Internet Information Services (IIS, anciennement Internet Information Server) est un logiciel de serveur Web extensible créé par Microsoft pour une utilisation avec la famille Windows NT. IIS prend en charge [HyperText Transfer Protocol](#). ([HTTP](#)), [HTTP/2](#), [HyperText Transfer Protocol Secure](#). ([HTTPS](#)), [FTP](#), [FTPS](#), [SMTP](#) et [NNTP](#).

Pour en savoir plus <https://en.wikipedia.org/wiki/InternetInformationServices>.

#### 3.3.2.5 Système de Gestion de Bases de Données. (SGBD) : SQL Server

Microsoft SQL Server est un [Système de Gestion de Bases de Données Relationnelle](#). (SGBDR) développé par Microsoft. En tant que serveur de base de données, il s'agit d'un produit logiciel dont la fonction principale est de stocker et de récupérer des données comme demandé par d'autres applications logicielles, qui peuvent s'exécuter sur le même ordinateur ou sur un autre ordinateur sur un réseau (y compris Internet). Pour en savoir plus <https://en.wikipedia.org/wiki/MicrosoftSQLServer>.

### 3.3.3 Motifs d'architecture logicielle

Dans cette section, je définis tout d'abord ce qu'est une architecture d'un système d'information. Puis, je clarifie la différence entre l'architecture matérielle et logicielle. Ensuite, j'explique pourquoi j'ai

besoin des motifs d'architecture logicielle et quels sont les différents motifs existants avant d'expliciter les motifs que j'ai implémentés.

En effet, réaliser l'architecture d'un système d'information consiste à décrire, généralement sous forme de figure, l'ensemble des différents composants qui constituent le système et les relations qui les relient les uns aux autres. Nous distinguons<sup>7</sup> deux types d'architecture : matérielle et logicielle. La première s'intéresse à l'aspect hardware<sup>8</sup> du système d'information. La seconde concerne plutôt l'aspect software, c'est-à-dire les couches logicielles qui composent l'application.

L'architecture logicielle peut être implantée de plusieurs façons, c'est là où intervient le motif d'architecture : il définit tout d'abord les différentes couches logicielles ainsi que la manière dont elles doivent communiquer entre elles. Les motifs d'architecture sont nombreux, sans prétendre être exhaustif, citons : **Model-View-Controller. (MVC)**, MVP, MVVM, MVI, MVT, etc. J'ai fait recours à deux motifs d'architecture logicielle, à savoir : **Model–View–Presenter (MVP)**, Data Access Object (**DAO**), **R**Epresentational **S**tate **T**ransfer (**REST**). Je présente dans la suite le principe de fonctionnement de chacun. Mais, avant de rentrer dans les détails, je réponds d'abord à la question : pourquoi ai-je implanté des motifs d'architecture logicielle ?

En réalité, n'étant pas convaincus de la « programmation spaghetti » et dans l'espoir de pouvoir réaliser un « clean code », nous avons jugé nécessaire l'implémentation d'un motif d'architecture logicielle non seulement pour assurer la séparation des préoccupations (separation of concerns) mais aussi pour garantir une cohésion élevée, un couplage faible et une abstraction du code.

### 3.3.3.1 MVC

L'objectif du motif **MVC** (Model-View-Controller ou Modèle-Vue-Contrôleur) est un modèle dans la conception de logiciels. Il met l'accent sur la séparation entre la logique métier et l'affichage du logiciel. Cette « séparation des préoccupations » permet une meilleure répartition du travail et une maintenance améliorée. Le tableau ci-dessous en présente une explication.

TABLEAU 3.3 – Rôle des trois couches logicielles du motif MVC.

Couche logicielle	Rôle
Modèle	Gère les données et la logique métier
Vue	Gère la disposition et l'affichage
contrôleur	Chemine les commandes des parties "model" et "view"

La figure ci-dessous explique les moments d'intervention de chaque couche du motif **MVC**.

### 3.3.3.2 REST

REST (Representational State Transfer) est une architecture de messagerie utilisée par de nombreuses **Application Programming Interface. (API)** de service Web. REST est une architecture flexible,

7. également l'architecture métier et l'architecture fonctionnelle mais ces types d'architecture ne rentrent pas dans le cadre du contexte de cette section.

8. Routeurs, commutateurs, serveurs, imprimantes, ordinateurs, câblage, caméras, pointeuses biométriques, etc

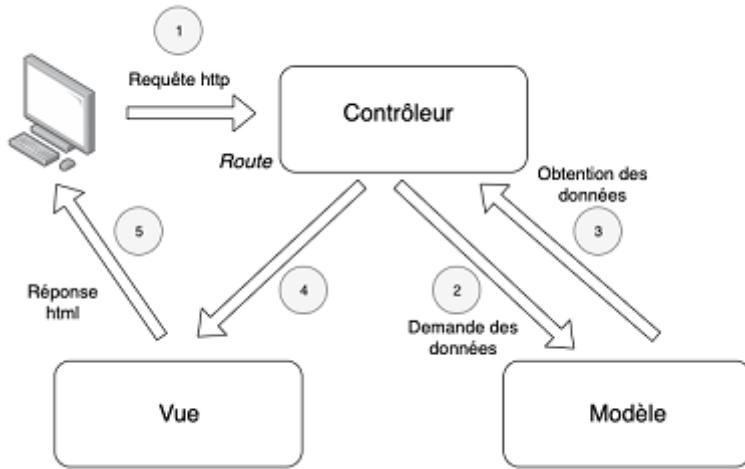


FIGURE 3.2 – Rôle des trois couches logicielles du motif MVC

avec des instructions et la prise en charge de plusieurs normes, telles que [HTTPS](#), XML et . Ce dernier signifie JavaScript Object Notation, un format de données léger et minimal. [JavaScript Object Notation. \(JSON\)](#) est celui utilisé dans la solution pour stocker et transporter des données.

### 3.3.3.3 DAO

Grâce aux objets métier, le motif [DAO](#) permet de fournir une abstraction des détails du mode de stockage. Ainsi, l'objectif du DAO peut être résumé en deux points :

1. Sauvegarder les informations provenant de la base de données dans des classes au lieu de laisser dispersés les attributs faisant partie à la même entité ;
2. Déléguer les accès aux données persistantes à une couche à part au lieu de gérer tout dans le modèle. De ce fait, le reste du modèle ne serait modifié que si les règles de gestion métier changent.

**Entity Framework** est utilisé pour faire ce motif. C'est un outil permettant de créer une couche d'accès aux données (DAL pour Data Access Layer) liées à une base de données relationnelle. Il propose la création d'un schéma conceptuel composé d'entités qui permettent la manipulation d'une source de données, sans écrire une seule ligne de SQL, grâce à LINQ To Entities. Comparé à d'autres solutions de mapping objet-relationnel (**ORM**), Entity Framework assure l'indépendance du schéma conceptuel (entités ou objets) du schéma logique de la base de données, c'est-à-dire des tables. Ainsi, le code produit et le modèle conceptuel ne sont pas couplés à une base de données spécifique. Pour en savoir plus veillez visiter le lien <https://pmusso.developpez.com/tutoriels/dotnet/entity-framework/introduction/>

## 3.4 Sécurité

Dans cette section, je présente les algorithmes et techniques de chiffrement que nous avons utilisés pour sécuriser davantage l'application.

### 3.4.1 Hachage des mots de passe

Il s'avère unanime que les mots de passe représentent l'élément clé permettant aux systèmes d'information d'authentifier les utilisateurs. De ce fait, la protection des mots de passe constitue une tâche primordiale dans les stratégies de sécurité de toute application informatique.

L'une des règles de base de la protection des mots de passe est qu'ils ne doivent pas être sauvegardés en clair dans la base de données. Le moyen le plus fréquent pour protéger les mots de passe consiste à sauvegarder plutôt les hash des mots de passe dans la base de données. Puis, si l'utilisateur saisit son mot de passe, l'application calcule le hash et vérifie s'il existe ou non dans la base de données. Si oui, c'est-à-dire qu'il s'agit du bon mot de passe. Sinon, cela veut dire que le mot de passe fourni est erroné.

Le calcul des hash des mots de passe est réalisé par une fonction de hachage qui désigne une fonction à sens unique qui, à partir d'une donnée fournie en entrée, calcule une empreinte numérique, appelée le hash, servant à identifier de façon unique la donnée en entrée. Bien évidemment, les fonctions de hachage sont nombreuses, citons à titre d'exemple :

1. MD4 (R. L. Rivest, 1990) ;
2. MD5 (R. Rivest et Dusse, 1992) ;
3. Tiger (Anderson et Biham, 1996) ;
4. RIPEMD (Dobbertin, Bosselaers et Preneel, 1996) ;
5. Whirlpool (Barreto et Rijmen, 2000) ;
6. MD6 (R. L. Rivest et al., 2008) ;
7. BLAKE (Aumasson et al., 2008) ;
8. la famille SHA (Michail et al., 2005 ; Glabb et al., 2007 ; Dworkin, 2015) ;
9. GOST (Dolmatov et Degtyarev, 2013).

Certaines de ces fonctions de hachage sont actuellement vulnérables car elles ont été déjà cassées et ne sont plus résistantes aux collisions<sup>9</sup> (Chabaud et Joux, 1998 ; Wang et Yu, 2005 ; Wang, Yin et Yu, 2005 ; Mendel et al., 2008). Nous avons choisi **SHA-256** comme elle n'a, jusqu'à présent, pas été compromise par une attaque significative.

## 3.5 Autres

Dans cette section, je présente tout d'abord l'environnement de rédaction puis le logiciel de sauvegarde et de partage des fichiers.

### 3.5.1 Environnement de rédaction : MikTex & TexStudio

Dans cette section, j'explique tout d'abord ce que signifie Tex puis LaTeX. Ensuite, je présente la distribution Tex et l'éditeur LaTeX que j'ai utilisé. Enfin, j'argumente pourquoi j'ai choisi LaTeX. En effet, TeX est un langage et un système de composition de documents. Il fournit de nombreuses

---

9. Dans notre contexte, une collision désigne deux mots de passe différent ayant le même hash.

commandes permettant de spécifier le format du document avec beaucoup de détails<sup>10</sup> et dispose d’algorithmes spécialisés pour le calcul de flux optimal de texte dans les documents<sup>11</sup>. TeX met donc à disposition des algorithmes et des commandes puissants pour spécifier les moindres détails lors de l’élaboration d’un document. Mais, vu que l’utilisation directe du TeX est assez ardue, il a été étendu en LaTeX qui constitue un jeu de macro-commandes basées sur TeX.

L’idée sur laquelle se base LaTeX est de cacher la complexité de la mise en page en séparant le format (mise en page) du contenu du document. Ainsi, il suffit de donner une structure au contenu du document sans besoin de rentrer dans les détails. Par exemple, pour créer une nouvelle section, il n’est demandé que d’écrire `section{...}` au lieu de sélectionner une police, définir son style, insérer les espaces appropriés avant et après l’en-tête de section, etc. Tous ces détails seront gérés automatiquement par TeX.

Pour pouvoir travailler sur LaTeX, il faut tout d’abord choisir une distribution TeX puis un éditeur LaTeX. Bien évidemment, les distributions TeX sont nombreuses ainsi que les éditeurs LaTeX. La distribution Tex que j’ai utilisé est **MikTex** et j’ai utilisé **TeXStudio** comme éditeur LaTeX.

J’ai privilégié LaTeX au détriment de Microsoft Office non seulement parce que LaTeX est gratuit et multiplateforme tandis que Microsoft Office est payant et n’est natif que sur Windows mais aussi pour les raisons suivantes :

1. La gestion automatique de la mise en page en LaTeX du document via les templates ;
2. La gestion automatique des références bibliographiques (via BibTeX par exemple) ;
3. La génération automatique et en une seule commande de tables de matières, liste de figures et liste de tableaux ;
4. La génération automatique de la liste des acronymes cliquables tout en précisant les pages où figurent chaque acronyme ;
5. La taille des document produits par LaTeX fait généralement la moitié de leurs équivalents sur Microsoft Word ;

### 3.5.2 Logiciel de sauvegarde et de partage : Dropbox

Pour pouvoir travailler sur le projet à distance, j’avais besoin d’un moyen nous permettant non seulement de sauvegarder les documents issus du projet mais aussi de pourvoir les partager avec notre encadrant et les membres du groupe. Pour ce faire, j’avais le choix entre utiliser un service de messagerie électronique<sup>12</sup> ou faire recours à un service d’hébergement de fichiers. J’ai opté pour ce dernier car il m’a permis de pouvoir travailler sur le même espace de travail avec une synchronisation de modifications en temps réel.

Bien évidemment, les services d’hébergement de fichiers sont nombreux : Google Drive, Microsoft OneDrive, Dropbox, pCloud, etc. J’ai choisi Dropbox comme je m’y suis déjà habitué.

---

10. Styles de police, espacement, crénage, ligatures, etc.

11. Par exemple, où s’achèvent les lignes, les pages, etc.

12. Il s’agit de partager les documents par e-mail.

# Analyse fonctionnelle et conceptuelle

---

Dans ce chapitre, je passe à la phase de conception de l'application dans lequel j'explique en détails les différents diagrammes UML relatifs à l'analyse fonctionnelle.

## 4.1 Analyse fonctionnelle

L'analyse fonctionnelle peut être expliquée comme une traduction du cahier de charges en une langue de conception (UML dans notre cas) permettant de caractériser les fonctionnalités d'un logiciel de façon plus compréhensible, répartissable, et réalisable. Pour ce faire, je me suis posés, essentiellement, trois questions, à savoir :

1. **Qui peut faire quoi ?** Pour répondre à cette question, je dois décrire non seulement l'ensemble des acteurs (**qui**) et l'ensemble des fonctionnalités du système (**quoi**) mais aussi les priviléges de chaque acteur (**peut faire**), les relations entre acteurs et entre fonctionnalités ;
2. **Comment ?** La question étant « Comment un acteur procède à une fonctionnalité ? », il s'agit donc de détailler comment se déroule le processus d'interaction entre l'acteur et le système lorsque ce premier sollicite une fonctionnalité de ce dernier ;
3. **Quand ?** Cette question reprend la question précédente mais s'intéresse plutôt à l'aspect temps : à quel moment se déroule chaque étape du processus d'interaction ? Comment les étapes se succèdent dans l'ordre chronologique ?

Le diagramme de cas d'utilisation permet de répondre à la première question, le diagramme d'activité répond à la seconde question et le diagramme de séquence à la troisième.

### 4.1.1 Diagramme de cas d'utilisation

Dans cette section, j'identifie le système, les acteurs et les cas d'utilisations.

#### 4.1.1.1 Système

Le système représente une application mobile, nommée **Ri3aya** permettant aux patients de réserver des rendez-vous à domicile avec des consultants médicaux (médecins, infirmiers, pharmaciens, etc). L'application **Ri3aya** est composée de deux espaces, à savoir, l'espace Consultant Médical et l'espace Patient. Les espaces que comprend cette application sont décrits à travers le tableau 4.1 :

#### 4.1.1.2 Acteurs du système

Les fonctionnalités du système peuvent être sollicitées par deux types d'acteurs, à savoir :

1. **Patient** : c'est toute personne demandant un rendez-vous d'un consultant médical ;

TABLEAU 4.1 – Modules de l'application.

Espace	Fonctionnalités
Patient	<ul style="list-style-type: none"> <li>- Créer un compte en tant que Patient</li> <li>- Réinitialiser le mot de passe</li> <li>- Modifier les informations du profil</li> <li>- Rechercher une consultation</li> <li>- Demander un rendez-vous</li> <li>- Payer les frais d'un rendez-vous</li> <li>- Retirer une demande avant être accepter</li> <li>- Visualiser la liste de ses rendez-vous</li> <li>- Visualiser les détails d'un rendez-vous</li> <li>- Visualiser le bilan statistique</li> <li>- Visualiser les notifications reçues</li> </ul>
Consultant Médical	<ul style="list-style-type: none"> <li>- Créer un compte en tant que Consultant Médical</li> <li>- Réinitialiser le mot de passe</li> <li>- Rectifier les informations du profil</li> <li>- Visualiser la liste des rendez-vous</li> <li>- Visualiser les informations d'un patient</li> <li>- Accepter un rendez-vous</li> <li>- Refuser un rendez-vous</li> <li>- Planifier, initialement, les horaires de consultation</li> <li>- Mettre à jour les horaires de consultations</li> <li>- Reporter un rendez-vous pour un patient</li> <li>- Visualiser le bilan statistique</li> <li>- Visualiser les notifications reçues</li> </ul>

2. **Consultant Médical :** c'est l'acteur que les patients demandent des rendez-vous. J'ai lui désigné le terme « Consultant Médical » puisqu'il englobe le médecin, l'infirmier, le pharmacien et tout personne dans le domaine médical pouvant être sollicité.

#### 4.1.1.3 Cas d'utilisation

Les figures 4.1, 4.2 et 4.3 présentent les diagrammes de cas d'utilisation des acteurs du système.

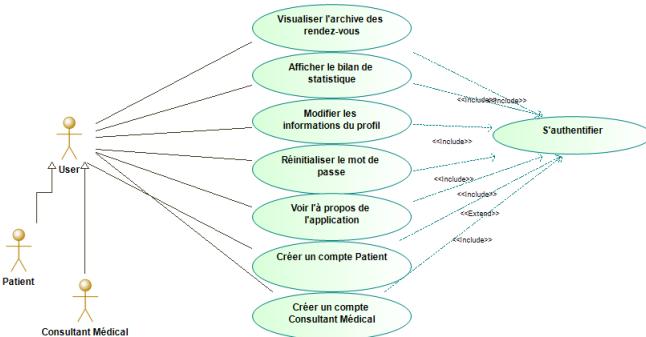


FIGURE 4.1 – Diagramme de cas d'utilisation d'utilisateur

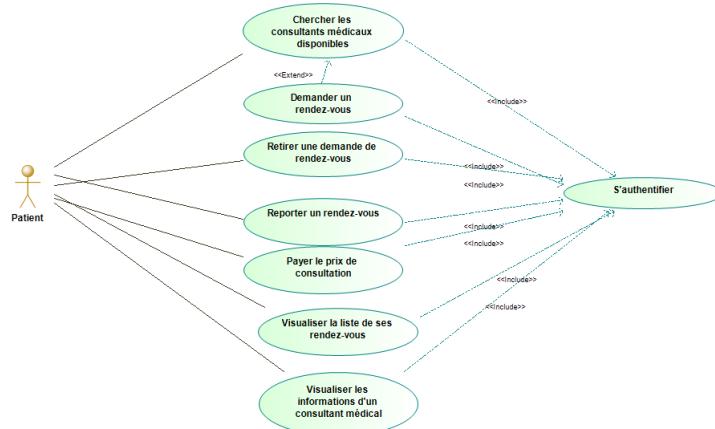


FIGURE 4.2 – Diagramme de cas d'utilisation du patient

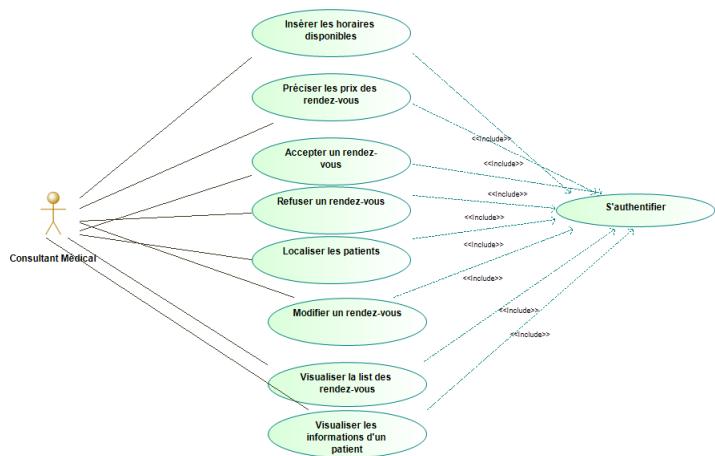


FIGURE 4.3 – Diagramme de cas d'utilisation du consultant médical

#### 4.1.2 Diagrammes d'activité

Cette section a pour objectif de mettre en surbrillance le processus quelques fonctionnalités de l'application pour voir les détails. J'ai choisi trois cas d'utilisation, à savoir, la création d'un compte, la demande d'un rendez-vous et l'ajournement d'un rendez-vous.

##### 4.1.2.1 Crédation de compte

Le processus de création de compte commence tout d'abord par la saisie d'informations requises. Ensuite, l'utilisateur soumet une demande de création du compte. Pour des raisons de sécurité, afin de valider la création de compte, nous envoyons au numéro de téléphone de l'utilisateur un SMS comprenant un code **One-Time Password (OTP)** qui serait expiré dans 2 minutes. Cela nous permettrait de vérifier qu'il est bien le sien. L'envoi des SMS est effectué via l'Application Programming Interface (**API**) Firebase Authentication. L'application bloquerait l'utilisateur pour une durée de quatre heures après avoir reçu 5 codes **OTP** sans utiliser aucun. Si l'utilisateur reçoit le SMS, Firebase à travers Google Play Services essaierait de récupérer automatiquement le code **OTP** du SMS ; si l'opération réussit, un courrier de bienvenue serait envoyé à l'utilisateur et la création du compte s'achève avec succès. Toutefois, si Firebase n'arrive pas à récupérer le SMS reçu, l'application demande à l'utilisateur

de saisir manuellement le code **OPT**. Voir la figure 4.4.

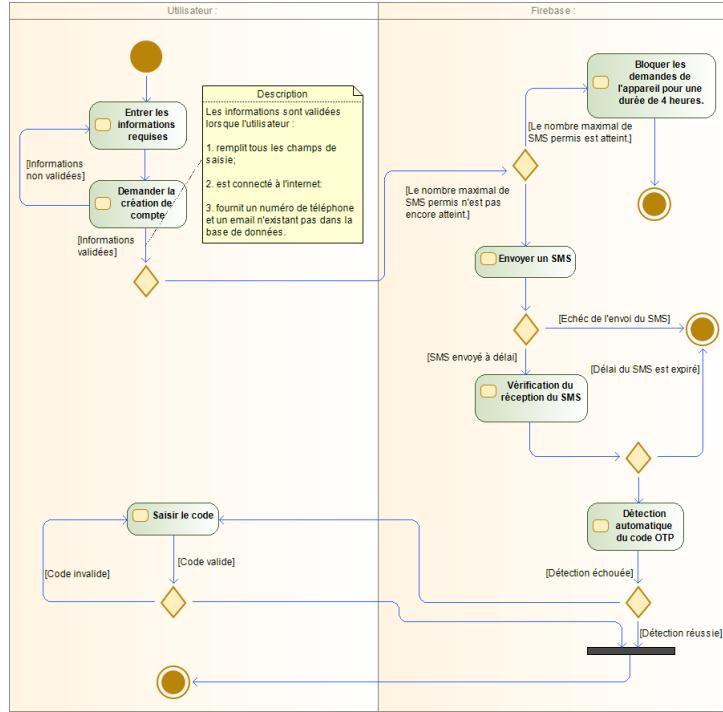


FIGURE 4.4 – Diagramme d'activité : Crédation de compte

#### 4.1.2.2 Demande de rendez-vous

Pour une demande de rendez-vous, il faut d'abord rechercher une consultation par consultant médical ou par spécialité. Une fois le patient choisit le profil du consultant médical qui lui satisfait, il peut directement lancer la demande de rendez-vous. Une notification sera envoyée au consultant médical en question, ensuite, le patient aussi recevra une notification de la décision du consultant médical concernant le rendez-vous. S'il s'agit d'une approbation, le patient doit procéder au paiement des frais de consultation qui doit être effectué en Bankily. Par contre si il s'agit un rejet, le processus terminera par la notification. Voir la figure 4.5.

#### 4.1.2.3 Ajournement de rendez-vous

Le patient peut lancer une demande de report de rendez-vous, dans ce cas une notification sera envoyée au consultant médical en question. Ce dernier devra proposer au patient la date et l'horaire du nouveau rendez-vous. Si le patient confirme la proposition, une notification à ce propos sera envoyée au consultant médical concerné. En revanche, si les nouvelles date et horaire ne satisfait pas au patient, il pourrait les rejeter, avec la possibilité d'indiquer sous forme de commentaire les dates et les horaires de préférence, et demander au consultant médical de proposer une nouvelle date et horaire. Voir la figure 4.6.

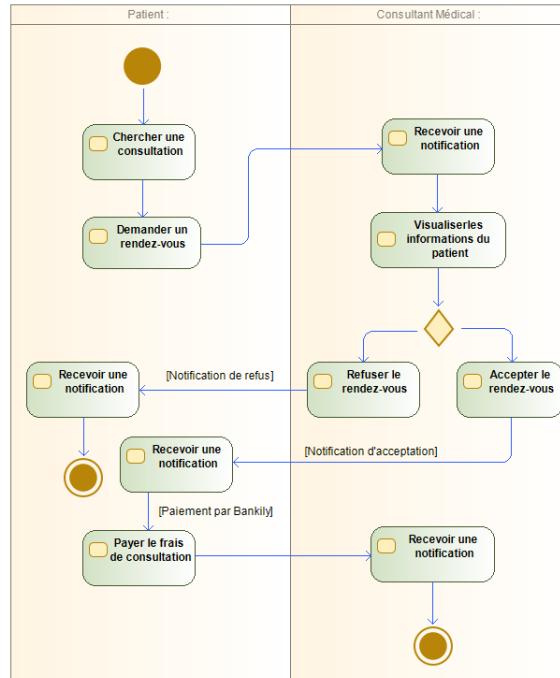


FIGURE 4.5 – Diagramme d’activité : Demande de rendez-vous

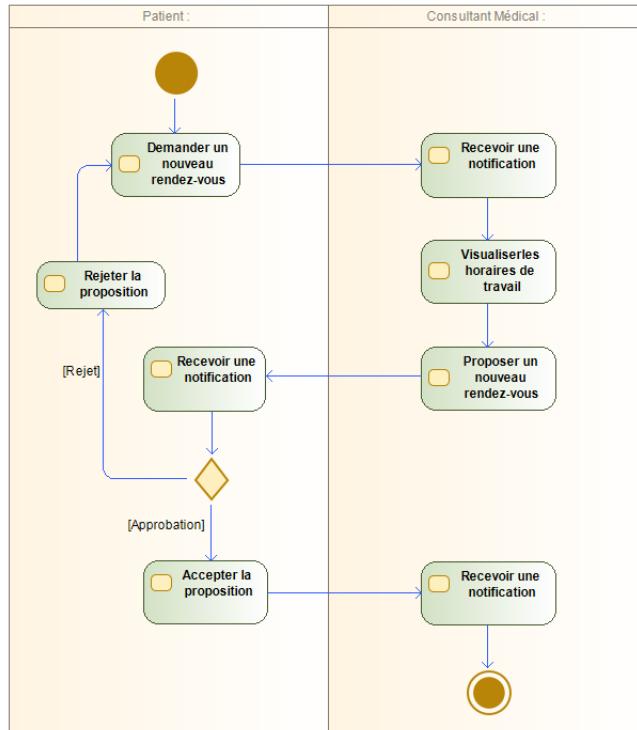


FIGURE 4.6 – Diagramme d’activité : Ajournement de rendez-vous

#### 4.1.3 Diagrammes de séquence

L’objectif de cette section est de mettre l’accent sur l’aspect temps du déroulement de processus des fonctionnalités pour lesquelles nous avons fait les diagrammes d’activité. Voir les figures 4.7 et 4.8.

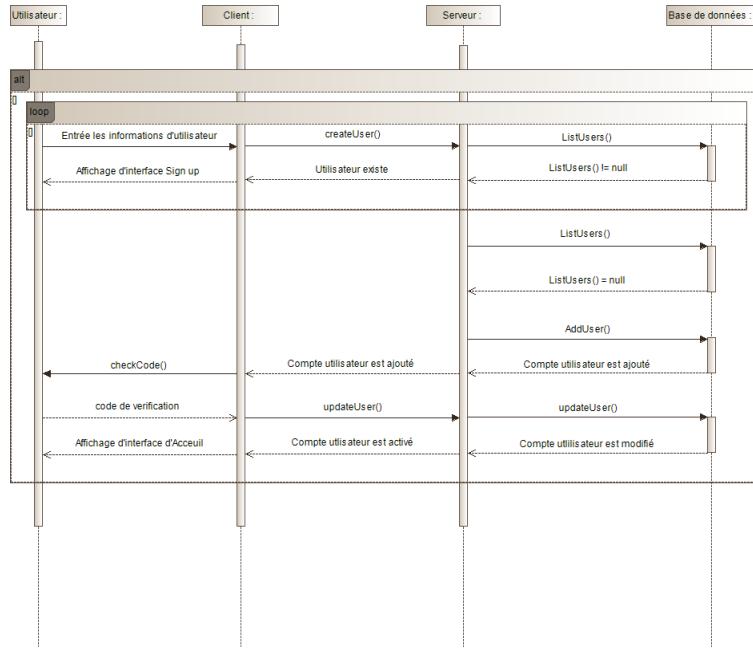


FIGURE 4.7 – Diagramme de séquence : Création de compte

## 4.2 Modélisation de la base de données

### 4.2.1 Diagramme de classes

Suite à l'analyse fonctionnelle de l'application, je m'oriente désormais la modélisation de la base de données. C'est dans ce cadre que j'ai réalisé le diagramme de classes qui permet d'identifier les différentes entités<sup>1</sup> du système et les relations entre elles.

Les points suivants résument les principales contraintes sur lesquelles je me suis basé pour réaliser le diagramme de classes :

1. L'entité **User** désigne la classe qui englobe tout type d'utilisateur : Patient et Consultant Médical ;
2. L'entité **Consultation** C'est la classe par laquelle l'utilisateur initialise ses horaires disponibles pour les rendez-vous ;
3. La fonction du consultant médical représente une entité nommée **Function** ;
4. Le lieu de travail du consultant médical représente aussi une entité sous le nom **Workspace** ;

La figure 4.9 représente le diagramme de classes.

1. il s'agit des classes qui seront converties en tables.

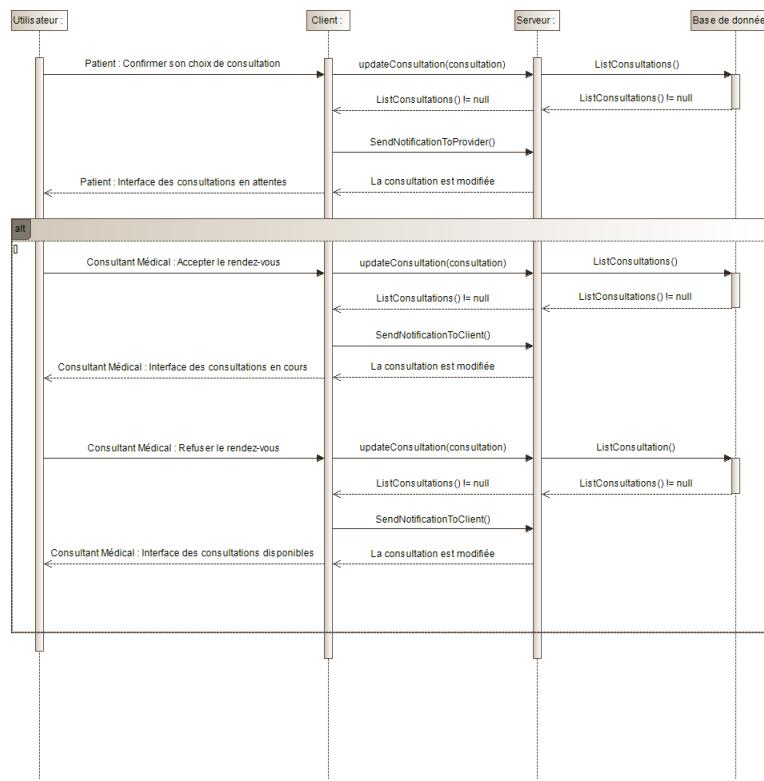


FIGURE 4.8 – Diagramme de séquence : Demande de rendez-vous

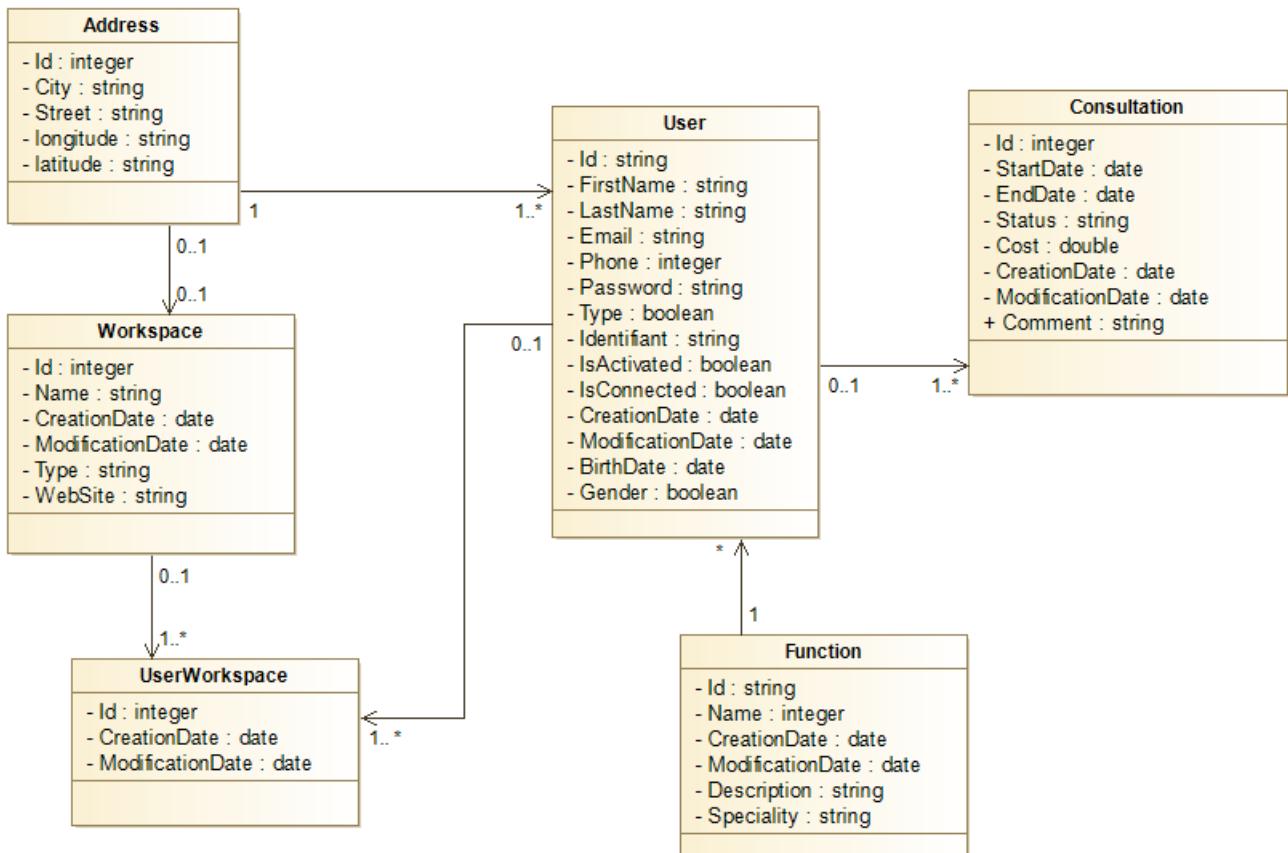


FIGURE 4.9 – Diagramme de classes.

# Implémentation

Dans ce chapitre, j'essaie de faire une évocation globale des détails techniques relatifs au développement de l'application **Ri3aya**. Pour ce faire, je présente, tout d'abord, l'architecture logicielle de l'application. Ensuite, j'illustre les services web que j'ai réalisé pour pouvoir interagir entre le **SGBD** et l'application. En outre, je décris les différents motifs d'architecture que j'ai implémenté. Après cela, j'explose comment j'ai procédé à l'obfuscation du code source pour le protéger contre les attaques de rétro-ingénierie. Enfin, à travers des captures d'écran, je réalise une démonstration des principales fonctionnalités de l'application.

## 5.1 Architecture de l'application

L'architecture de notre application peut être divisée en deux blocs, le premier concerne le côté front-end (IONIC), le second est relatif au côté back-end (services web et **SGBD**). La communication entre ces deux blocs est garantie à travers l'architecture Service-Oriented Architecture ([Service Oriented Architecture \(SOA\)](#)). La figure ci-dessous en présente une illustration.

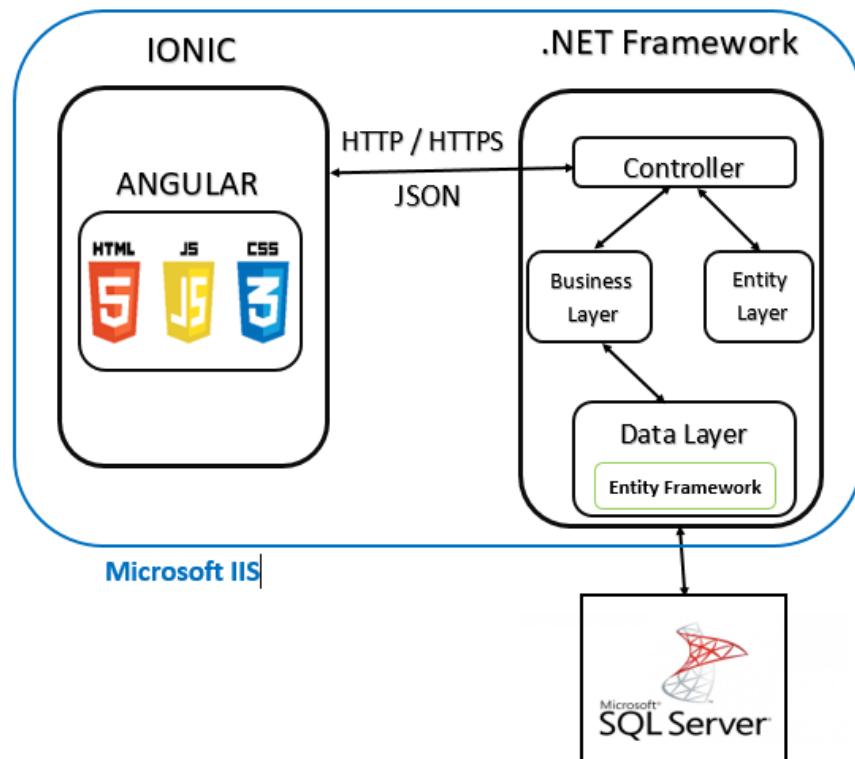


FIGURE 5.1 – Architecture logicielle de l'application.

Le bloc à gauche constitue le côté front-end de l'application. J'ai utilisé **IONIC** (voir la section 3.3.1.3) avec **ANGULAR** (voir la section 3.3.1.4), ce dernier repose essentiellement sur une approche **MVC** que nous avons déjà expliqué dans la section 3.3.3.1.

Le bloc à droite représente le côté back-end de l'application. Autrement dit, il s'agit du serveur de l'application, là où je récupère les requêtes **HTTP** provenant du front-end, je les traite et je les fait retourner au front-end.

Toutes les requêtes de base de données sont gérées par un RESTful **API** regroupant un ensemble de services web réalisés via l'**Object-Relational Mapping** (en français **Mapping Objet-Relationnel**) (**ORM**) Entity Framework dans le **Data Layer** qui représente le couche **DAO** (voir la section 3.3.3.3). Ainsi, lorsque l'application aurait besoin d'exécuter une requête, elle communique avec le Contrôleur correspondant, à travers un tunnel **HTTP**, qui lui retourne le résultat sous format JavaScript Object Notation (**JSON**) comme la figure ci-dessus montre.

## 5.2 Services web

Je présente dans cette section les essentiels scripts C# dans le **Data Layer** qui constituent les services web du back-end.

Le tableau ci-dessous en présente la liste.

TABLEAU 5.1 – Rôles des différents services web de l'application.

Service web	Rôle
AuthRepository.cs	Authentification d'utilisateurs
AdressProvider.cs	Insertion, récupération, modification et suppression des données des adresses
ConsultationProvider.cs	Insertion, récupération, modification et suppression des données des consultations
FunctionProvider.cs	Insertion, récupération, modification et suppression des données des fonctions des consultant médical
UserProvider.cs	Insertion, récupération, modification et suppression des données des utilisateurs
WorkspaceProvider.cs	Insertion, récupération, modification et suppression des données des lieux de travaux des consultants médical
UserWorkspaceProvider.cs	Insertion, récupération, modification et suppression des données de la table associative entre "User" et "Workspace"

## 5.3 Vue sur l'application

Dans cette section, je présente les principales fonctionnalités dans chaque espace de l'application.

### 5.3.1 Options d'internaute

Un internaute désigne tout individu qui navigue dans l'application sans qu'il soit connecté à un compte.

La première figure 5.2 présente l'écran d'accueil de l'application. À ce stade, l'internaute peut se connecter à son compte **Ri3aya** ou créer un compte s'il n'en a pas. Lorsque l'internaute clique sur **Se**

**connecter**, s'apparaît l'écran indiqué à la deuxième figure pour fournir les informations nécessaires relatives au compte en question avec l'option **Mot de passe oublié**. Dans le cas où l'internaute n'a pas de compte, il doit cliquer sur **S'inscrire** de la première figure pour accéder aux écrans de création de compte indiqués aux deux dernières figures dans lesquels il doit fournir aussi des informations relatives au compte.

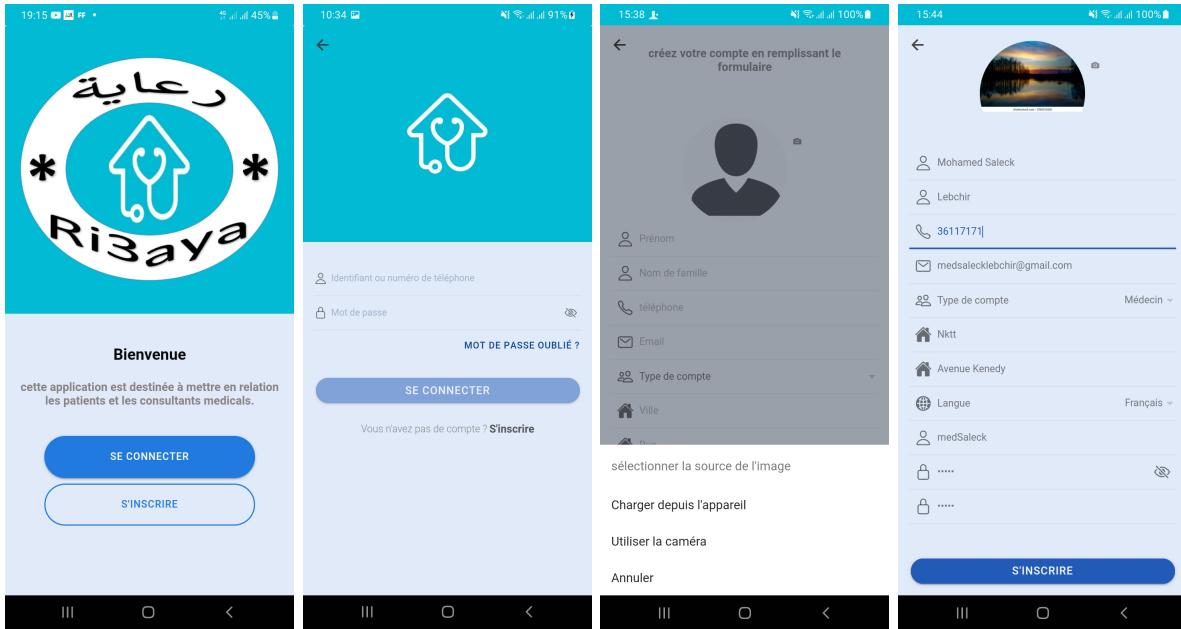


FIGURE 5.2 – Interfaces d'accueil pour l'internaute.

En cas de création de compte, l'internaute doit fournir l'ensemble des informations requises, sinon, un message d'erreur lui sera affiché en dessous des champs vides ou dans lesquels les informations sont invalides syntaxiquement. Après avoir fourni ces informations, un code **OPT** lui sera envoyé dans un SMS dont la détection est automatiquement faite via Firebase<sup>1</sup>. Le code **OPT** sert d'authentification d'internaute et de vérification de la validité du numéro de téléphone. Une fois le code **OPT** validé, les informations du compte seront insérée dans la base de données.

### 5.3.2 Espace Patient

Lorsque l'internaute qui a sélectionnée "Patient" comme type de compte, dépasse la vérification du code **OPT**, il passe à l'écran dans lequel il doit accomplir des informations relatives au type de compte choisi. Voir la deuxième figure de 5.3.

#### 5.3.2.1 Options de navigation et paramétrage

Après que le patient accomplis la création de son compte, il passe à la page d'accueil, dans laquelle il peut visualiser les consultants médicaux disponibles, ainsi que ses rendez-vous en attente (en attendant les décisions des consultants médicaux concernes), celles en cours et terminées. Il peut aussi changer les

1. Si la carte SIM est dans l'appareil de l'internaute, la détection du code **OPT** sera en arrière-plan sans passer par l'écran indiqué à la première figure de 5.3 mais plutôt il passe directement à la page suivante. Voir la figure

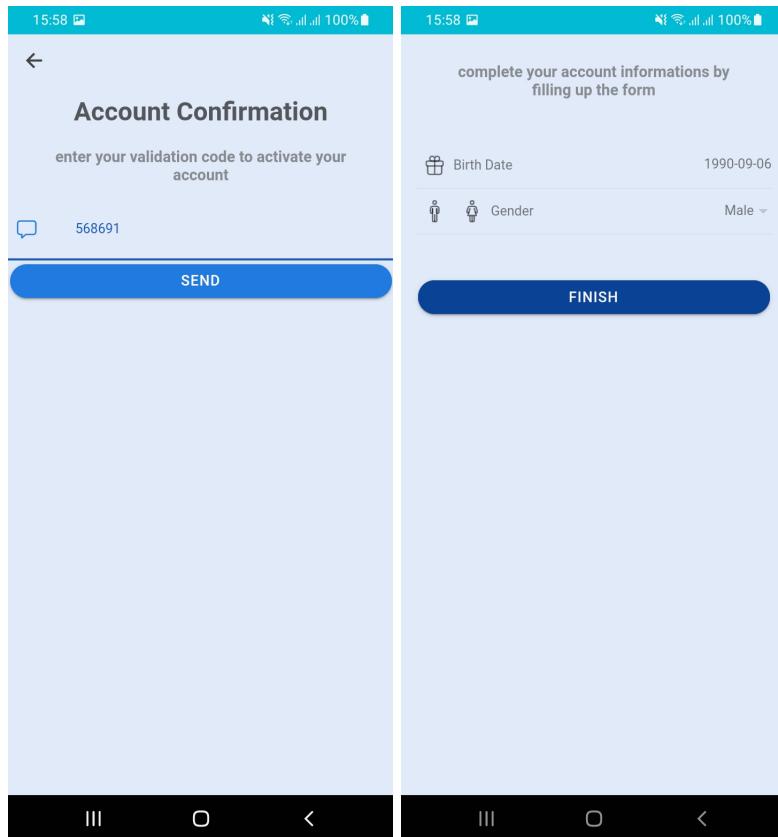


FIGURE 5.3 – Interfaces d’accomplissement des informations de patient.

paramétrés de son compte, visualiser l'historique de notifications, voir les informations du profil et les changer. Voir les figures 5.4

### 5.3.2.2 Demande de rendez-vous

Après avoir naviguer dans les consultations disponibles, le patient peut faire une demande de rendez-vous, et bien évident la demande peut être acceptée ou refusée comme déjà expliquer précédemment dans le diagramme d'activité 4.5. Voir les figures 5.5

### 5.3.3 Espace Consultant Médical

De même, si l'internaute qui a sélectionnée "Consultant Médical" comme type de compte, dépasse la vérification du code OPT, il passe à l'écran dans lequel il doit accomplir des informations relatives au type de compte choisi. Elles comportent : la fonction, le lieu de travail, les horaires avec les dates, ainsi qu'une pièce de jointure confirmant l'identité professionnelle . Voir les figures 5.6.

#### 5.3.3.1 Options de navigation et paramétrage

Après que le consultant médical accomplis la création de son compte, il passe à la page d'accueil, dans laquelle lui aussi peut visualiser les consultations en cours, celles en attentes, mais avec l'option de localiser ces patients ainsi que les rendez-vous termimes. Il peut aussi changer les paramétrés de son compte, visualiser l'historique de notifications, voir les informations du profil et les changer exactement comme le patient. Voir les figures 5.7.

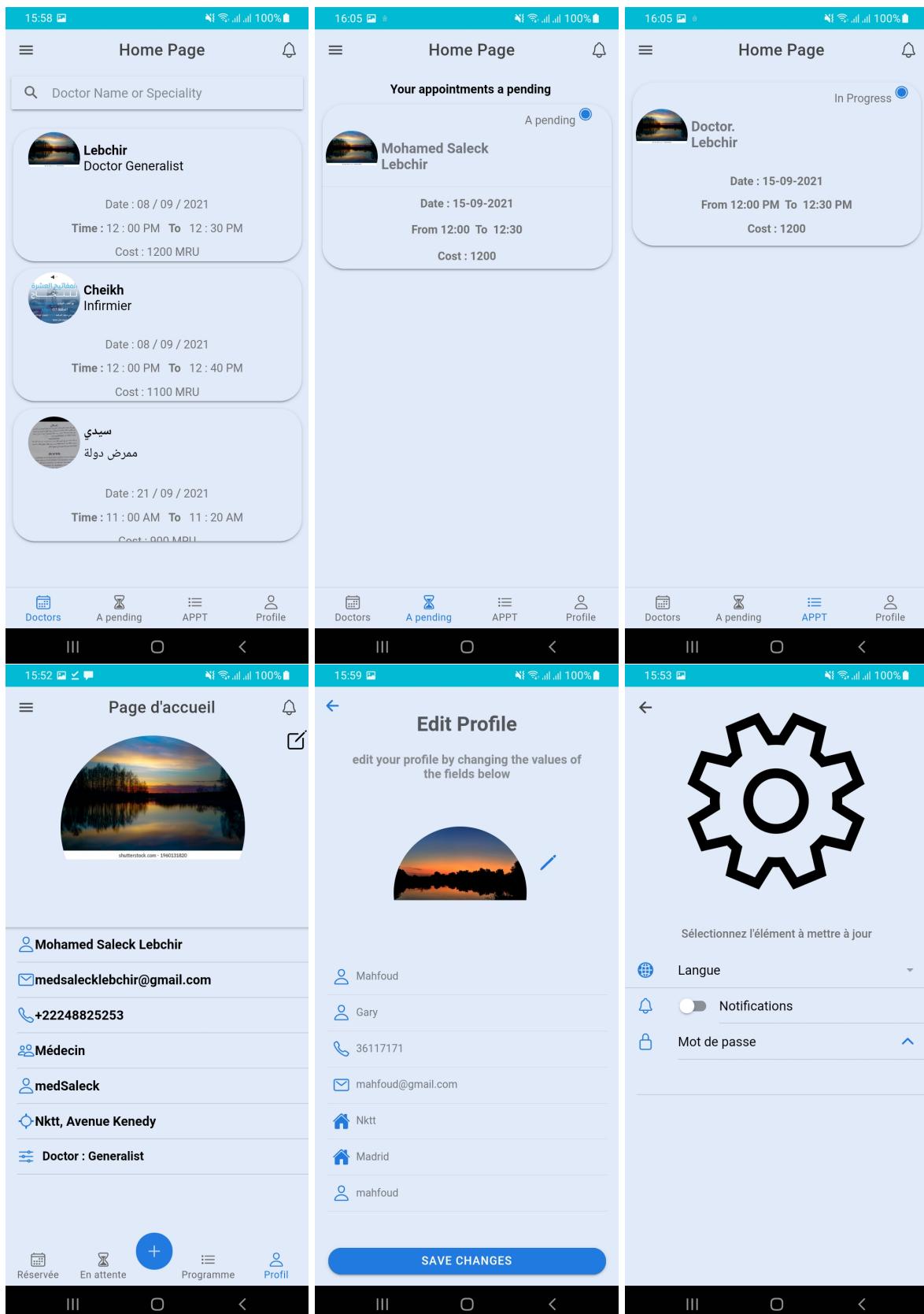


FIGURE 5.4 – Interfaces des options de navigation et paramétrage.

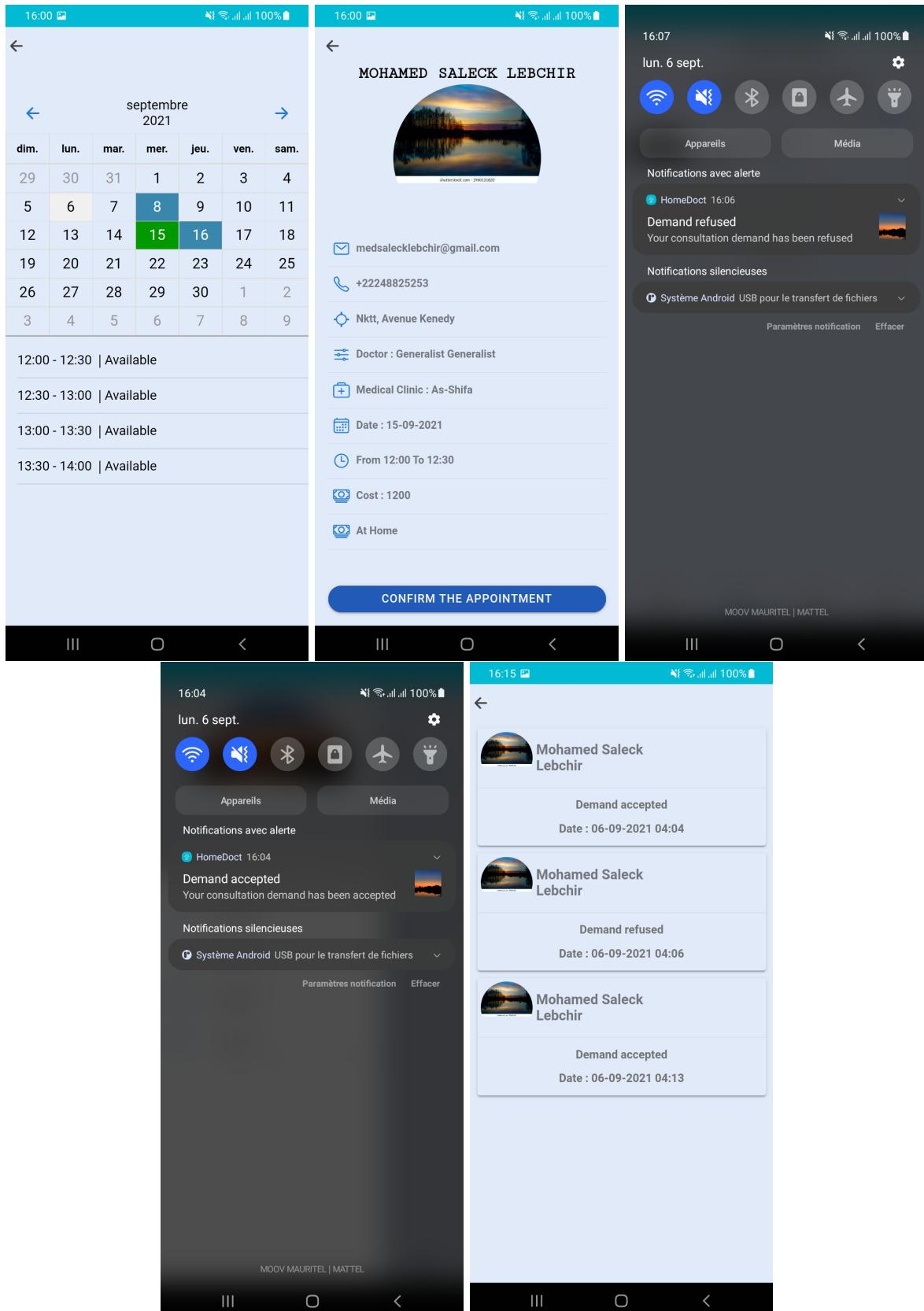


FIGURE 5.5 – Interfaces de demande de rendez-vous.

### 5.3.3.2 Approbation ou Refus d'une demande

Le consultant médical est notifié de toute demande de rendez-vous, ensuite, il peut prendre sa décision concernant la consultation en question. Voir les deux premières figures de 5.8. En cas où le

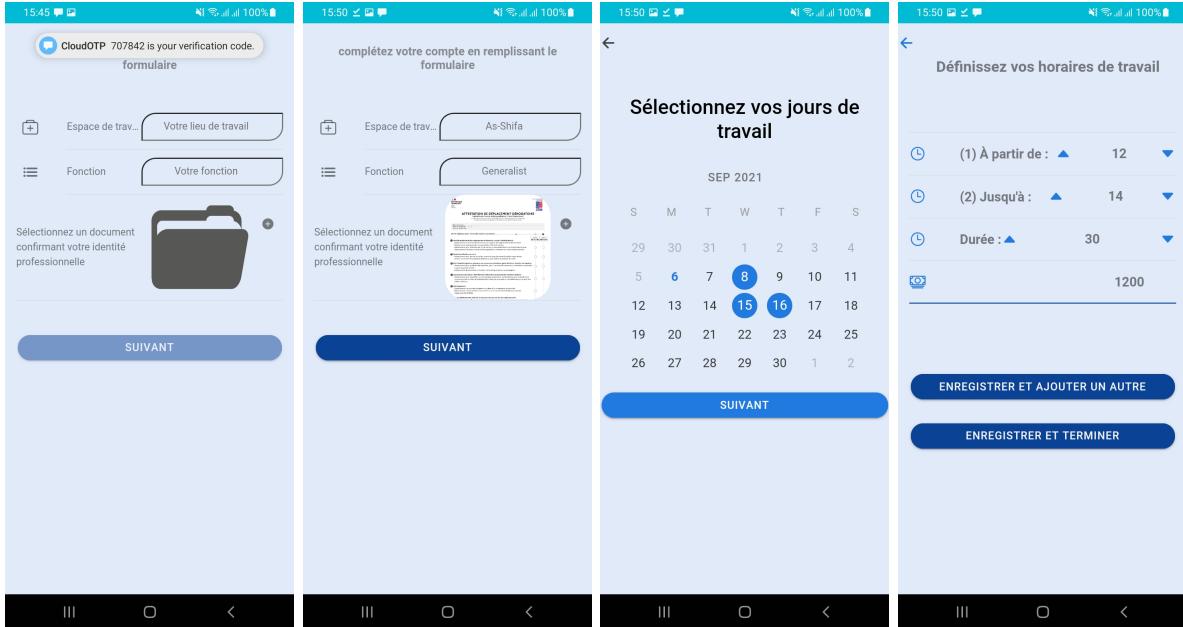


FIGURE 5.6 – Interfaces d’accomplissement des informations de consultant médical.

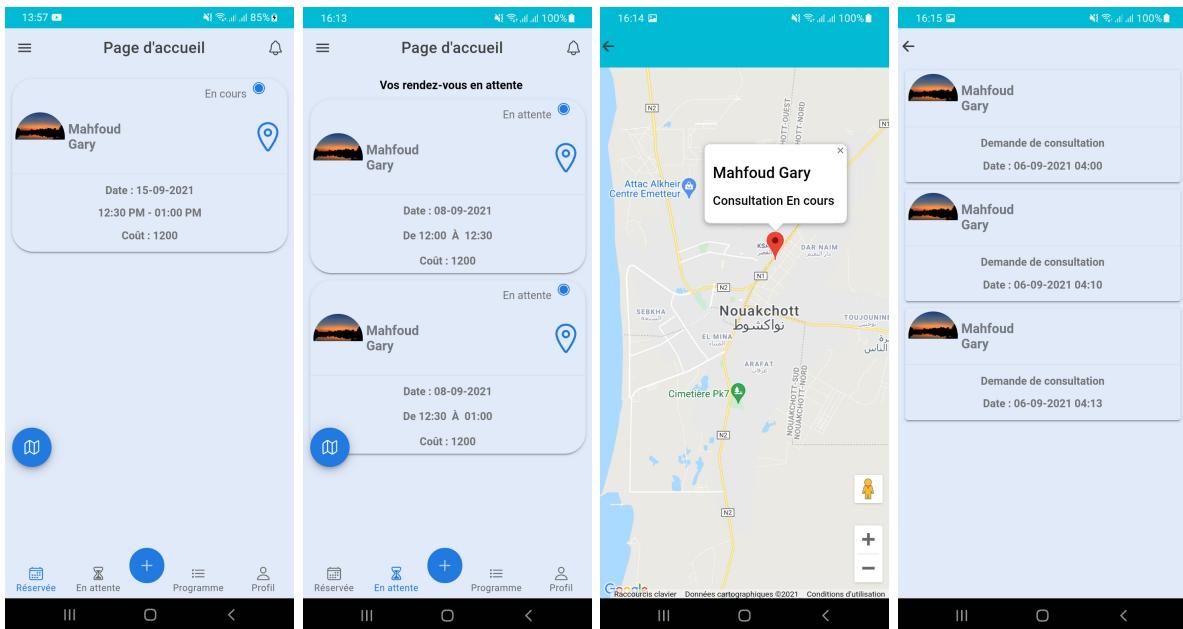


FIGURE 5.7 – Interfaces des options de navigation et de paramétrage.

consultant médical accepte la demande de rendez-vous, un événement se créera dans le calendrier de son téléphone. Voir la dernière figure de 5.8

### 5.3.3.3 Mise à jour d’emploi du temps

En cliquant sur l’icône "+" dans la page d’accueil du consultant médical, il lui apparait la calendrier du mois actuel pour sélectionner ses jours de travail, ensuite l’écran dans lequel il doit définir les horaires de travail. Cependant, il a la possibilité de voir les horaires sélectionnés au cas où il souhaite en supprimer ou en modifier un.

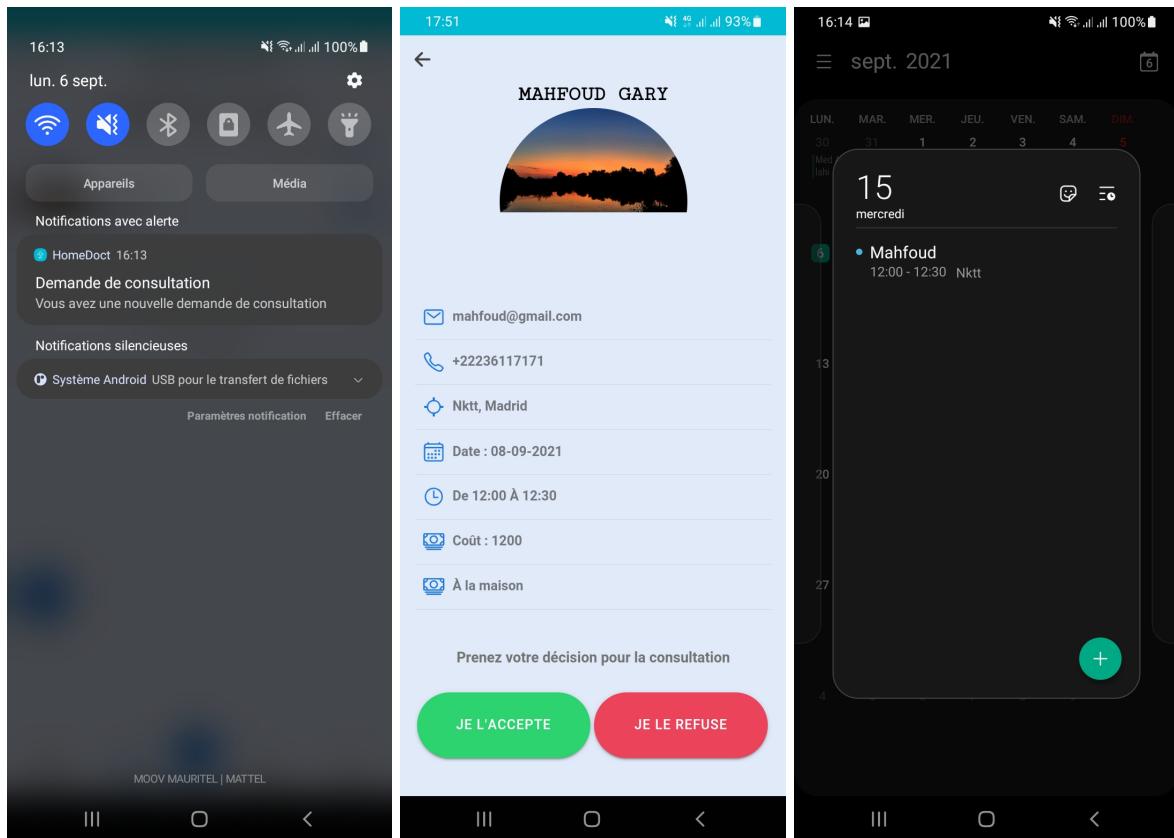


FIGURE 5.8 – Interfaces d'approbation et refus d'une demande de rendez-vous.

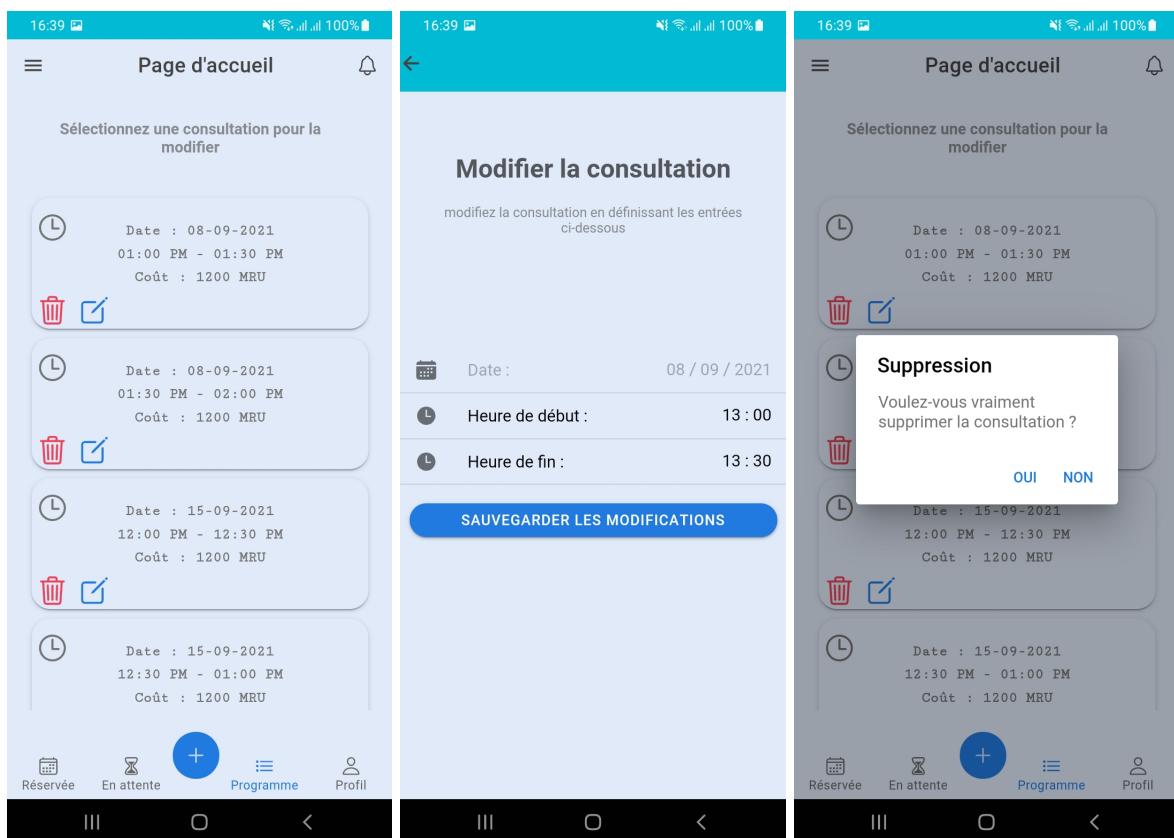


FIGURE 5.9 – Interfaces de mise à jour d'emploi du temps.

# Conclusion et perspectives

---

Pour conclure, l'objectif de mon projet de fin d'études consiste à concevoir et développer une application mobile permettant aux patients de solliciter des rendez-vous à domicile avec consultants médicaux. Le tableau ci-dessous présente un bilan rétrospectif de ce que j'ai achevé des fonctionnalités définies dans la section 4.1.1.1. D'après le tableau ci-dessus, j'ai implémenté 21 fonctionnalités sur 23,

TABLEAU 6.1 – Bilan rétrospectif sur les fonctionnalités implémentées.

Espace	Fonctionnalités	Achevée ?
Patient	- Créer un compte en tant que Patient	✓
	- Réinitialiser le mot de passe	✓
	- Modifier les informations du profil	✓
	- Rechercher une consultation	✓
	- Demander un rendez-vous	✓
	- Payer les frais d'un rendez-vous	✗
	- Retirer une demande avant d'être accepter	✗
	- Visualiser la liste de ses rendez-vous	✓
	- Visualiser les détails d'un rendez-vous	✓
	- Visualiser le bilan statistique	✓
Consultant Médical	- Visualiser les notifications reçues	✓
	- Créer un compte en tant que Consultant Médical	✓
	- Réinitialiser le mot de passe	✓
	- Rectifier les informations du profil	✓
	- Visualiser la liste des rendez-vous	✓
	- Visualiser les informations d'un patient	✓
	- Accepter un rendez-vous	✓
	- Refuser un rendez-vous	✓
	- Planifier, initialement, les horaires de consultation	✓
	- Mettre à jour les horaires de consultations	✓
	- Reporter un rendez-vous pour un patient	✗
	- Visualiser le bilan statistique	✓
	- Visualiser les notifications reçues	✓

soit un pourcentage de 91%, ce qui est tout à fait très significatif.

En effet, ce stage et par conséquence, ce projet m'a été d'un apport primordial. Il m'a permis, entre autres, de découvrir beaucoup de concepts et d'outils que je n'ai jamais appris tout au long de mon cursus universitaire, notamment :

1. Développement hybride : IONIC ;
2. Développement des web services C# ;
3. Motifs d'architecture logicielle ([MVC](#), [DAO](#) et [SOA](#)) ;
4. Fonctions de hachage (SHA-256) ;

5. Déploiement sur Azure ;
6. Test des services web (via Swagger<sup>1</sup>) ;
7. La publication d'application Android sur Google Play Store ;
8. La sauvegarde et le partage de documents (Dropbox) ;
9. L'environnement de rédaction LaTeX ;

Enfin, j'aimerai mentionner que l'objectif de **Ri3aya**, à long terme, est de développer toute une plateforme disponible sous forme d'application web, Android, iOS et pourquoi pas de bureau ?

En effet, je crois, sincèrement, à ce projet que je trouve prometteur et je n'épargnerai aucun effort à son développement avec le temps.

---

1. Swagger est un framework logiciel open source qui aide les développeurs à concevoir, construire, documenter et utiliser l'API Web RESTful. Pour en savoir plus, pour en savoir plus, veuillez visiter les liens <https://docs.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger?view=aspnetcore-5.0> et <https://www.nuget.org/packages/Swagger.Net.UI>

# Annexe 1 : Génération de l'App Bundle et l'APK en mode release

---

Pour pouvoir générer l'**Android Package Kit (APK)** ou l'App Bundle en mode release, cliquez sur **Build** du menu horizontal d'Android Studio puis sur **Generate Signed Bundle/APK**. Une fenêtre s'ouvrira et vous serez demandé de choisir entre la génération d'App Bundle ou d'APK. Sélectionnez votre choix puis cliquez sur **Next**. Voir la première figure de 7.1. Dans la nouvelle fenêtre ouverte, cliquez sur **Create New** puis renseignez les informations demandées comme illustre la deuxième figure de 7.1.

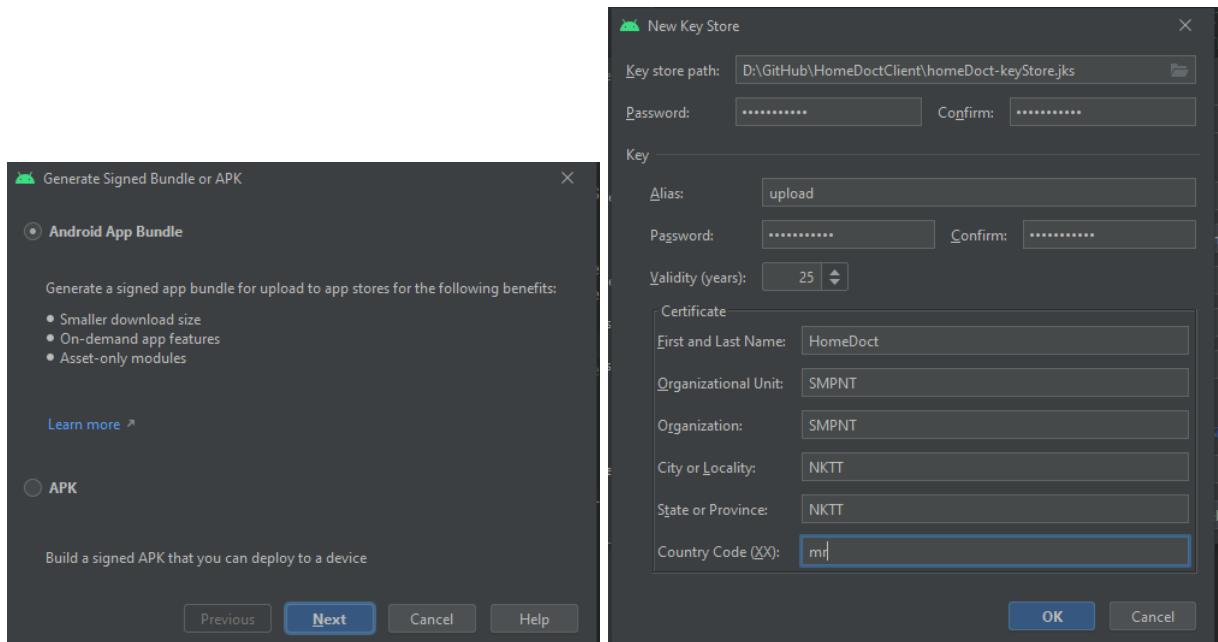


FIGURE 7.1 – Choix de génération et génération d'un nouveau Key Store.

Une fois cliqué sur **Ok**, vous aurez l'interface 7.2 qui présente un aperçu sur l'ensemble des informations précédemment saisies.

Lorsque vous cliquez sur **Next**, vous aurez l'interface ci-dessous. Sélectionnez le mode release, puis cliquez sur **Finish**. La figure 7.3 en présente une illustration.

Le processus de génération de l'App Bundle ou de l'APK peut durer quelques minutes suivant la performance de votre ordinateur. Une fois achevé, vous serez notifiés à ce propos. L'App Bundle ou l'APK généré en mode release se trouve dans /android/app/release du dossier du projet.

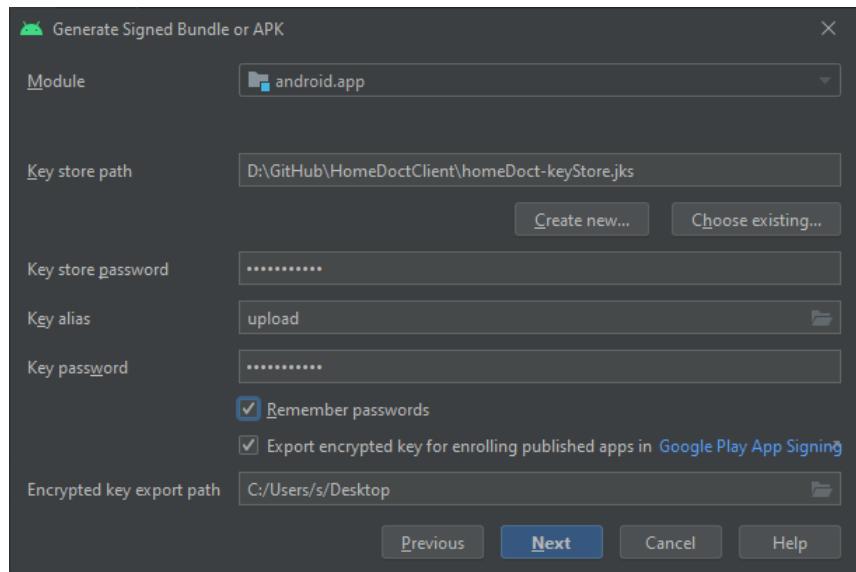


FIGURE 7.2 – Aperçu sur la configuration de l’APK ou de l’App Bundle.

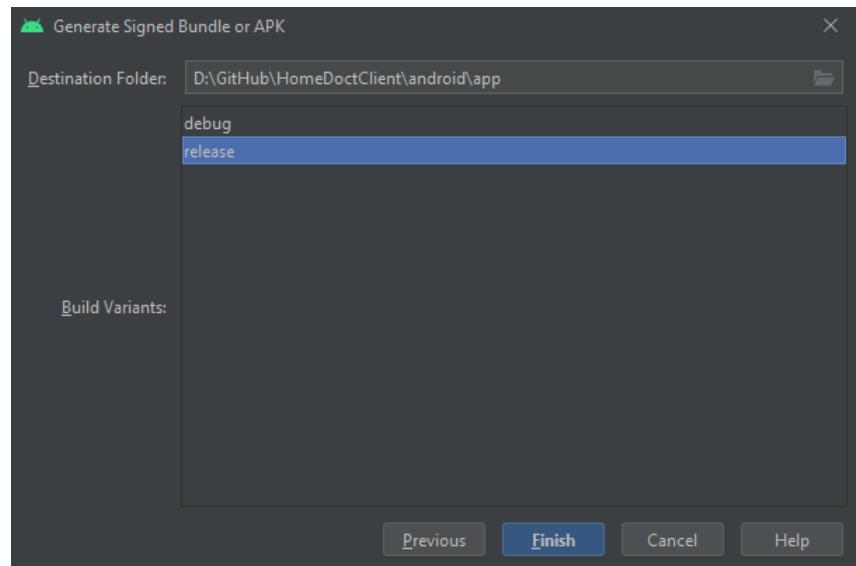


FIGURE 7.3 – Options de génération de l’App Bundle ou de l’APK en mode release.

# Annexe 2 : Déploiement des Web API et de Base de données sur Azure

Après avoir créer un compte Azure sur <https://portal.azure.com/>, commencez par créer une nouvelle base de données en cliquant sur **Base de données** dans le menu à gauche puis sur **Créer**, ce qui nécessite de sélectionnez un **Groupe de ressources**<sup>1</sup> déjà existant ou en créer un nouveau. La création de la base de données nécessite également la création d'un nouveau serveur de base de données à partir duquel vous pouvez accéder à celle-ci. Enfin, réglez les paramètres de pare-feu et votre base de donnez sera prête. Les figures 8.1 expliquent le processus.

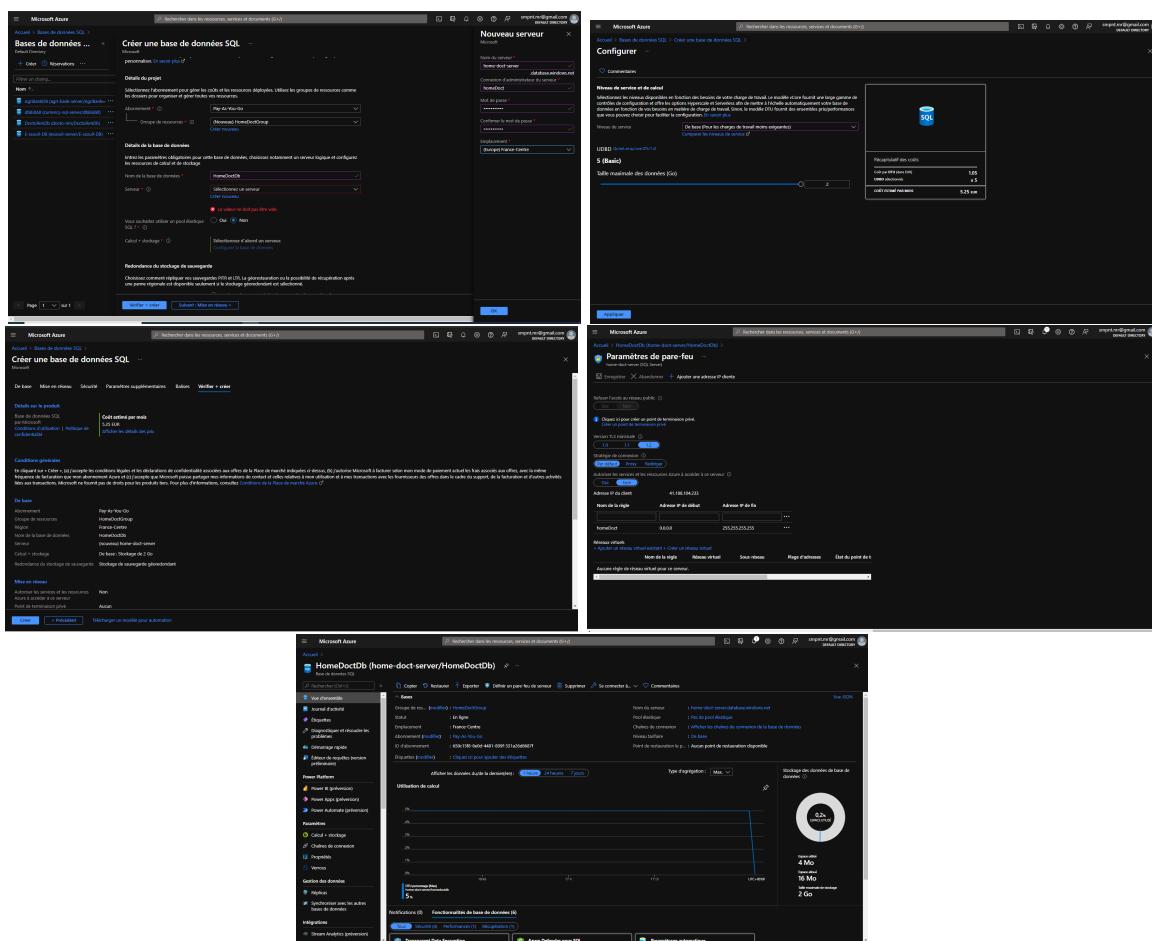


FIGURE 8.1 – Création d'une base de données sur Azure.

Juste après la creation de votre base de données, Azure vous générera le chaîne de connexion (Connec-

1. Un groupe de ressources est un conteneur regroupant les applications web et les base de données

tion String) pour que votre projet ASP .NET<sup>2</sup> pouvoir connecter à celle-ci. Voir la figure 8.2.

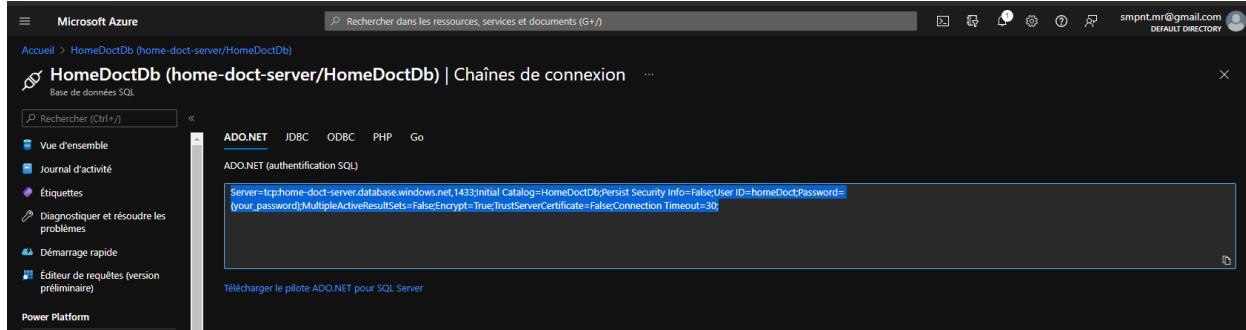


FIGURE 8.2 – Génération du chaîne de connexion.

Pour déployer votre web api sur Azure, vous devez créer une nouvelle application web (Web App) en cliquant sur **App Services** dans le menu à gauche puis sur **Créer**. En suite sélectionnez le même groupe de ressources que celle créé lors de la création de votre base de données puis renseignez les informations demandées, une fois terminé cliquez sur **Vérifier + créer**. Enfin, sélectionnez le niveau de tarif, puis cliquez sur **Appliquer**. Voir les figures 8.3.

The figure consists of two screenshots of the Microsoft Azure portal.

The top screenshot shows the 'Create Web App' wizard. In the 'Abonnement' dropdown, 'Pay-As-You-Go' is selected with 'HomeDocGroup' as the group. The 'Nom' field contains 'home-doct-web-api'. Under 'Plan App Service', 'Premium V2 P1v2' is selected with a note about 210 ACU. The 'Référence et taille' section shows '210 ACU au total, 3.5 Go de mémoire'. The 'Suivant : Déploiement (préversion) >' button is visible at the bottom.

The bottom screenshot shows the 'home-doct-web-api' application service in the Azure portal. The left sidebar includes sections like 'Vue d'ensemble', 'Contrôle d'accès (IAM)', 'Déploiement', 'Paramètres', and 'Réseau'. The main content area shows deployment logs, application insights, and monitoring charts for 'Http 5xx', 'Données entrantes', and 'Données sortantes'. The URL listed is 'https://home-doct-web-apizurewebsites.net'.

FIGURE 8.3 – Creation de l'application web.

2. Le projet ASP .NET représente les web api

# Annexe 3 : Publication d'application sur Google Play Store

---

Après avoir créer un compte Google Play sur <https://play.google.com/console/>, cliquez sur **Create app** à droite de l'écran pour créer une nouvelle application puis renseignez les informations demandées et cliquez sur **Create app** en bas de l'écran. Ensuite, cliquez sur **Main store listing** dans le menu à gauche, puis renseignez les informations demandées et cliquez sur **Save**. Ensuite, cliquez sur **Production** dans le menu à gauche aussi, puis charger l'**Android App Bundle (AAB)** de votre application<sup>1</sup>, puis renseignez les informations demandées. Une fois terminé, cliquez sur **Start rollout to Production**. Les figures 9.1, 9.2 et 9.3 expliquent toutes les étapes à suivre.

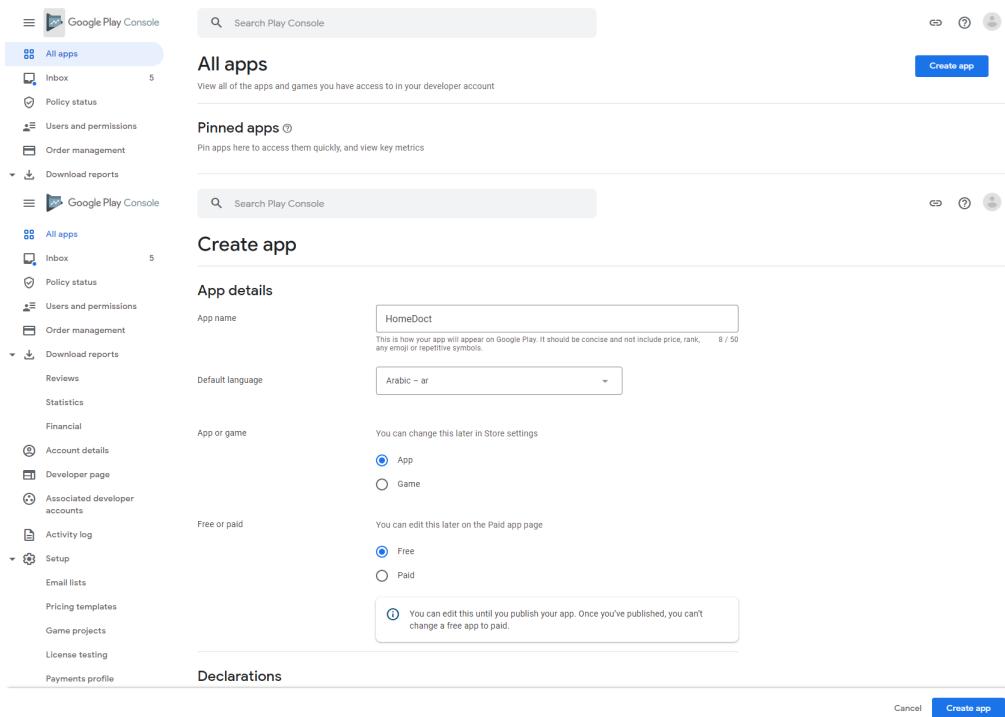


FIGURE 9.1 – Publication sur Google Play Store.

1. Google ne n'accepte plus du format APK sur le Play Store. Pour en savoir plus, veillez visiter le lien <https://www.phonandroid.com/tout-savoir-sur-format-aab-android-app-bundles-google-play-store.html>

The figure consists of three vertically stacked screenshots of the Google Play Console, illustrating the steps to create a production release:

- Screenshot 1: Privacy Policy**  
Shows the 'Privacy Policy' section where a URL (<https://homedoc.flycricket.io/privacy.html>) is entered into a text input field. The sidebar shows navigation options like 'Inbox', 'Statistics', 'Publishing overview', and 'Production'.
- Screenshot 2: Content ratings**  
Shows the 'Content ratings' section. It displays a summary table for 'Brazil' with a rating of 'All ages'. Below it, there's a 'Your ratings' section and a 'Category' dropdown set to 'Utility, Productivity, Communication, or Other'. A message at the bottom says 'Your changes have been saved'.
- Screenshot 3: Create production release**  
Shows the 'Create production release' screen. It indicates '2 Warnings' and has tabs for 'Prepare' and 'Review and release'. The 'Review and release' tab is selected. A table for 'New app bundles' is shown, with one entry: 'App bundle' (Version 1 (1.0), API levels 21+, Target SDK 30, Screen layouts 4, ABIs 5, Required features 5). Below it is a 'Release notes' section with a note about default language and required features. At the bottom, there are buttons for 'Edit release' and 'Start rollout to Production'.

FIGURE 9.2 – Publication sur Google Play Store 2.

The figure consists of three vertically stacked screenshots of the Google Play Console.

- Privacy Policy:** Shows the 'Privacy Policy' section where a URL (<https://homedoc.flycricket.io/privacy.html>) is entered into a text input field. A note states: "Add a privacy policy to your store listing to help provide transparency about how you treat sensitive user and device data. [Learn more](#). You must add a privacy policy if your target audience includes children under 13. Check the [User Data policy](#) to avoid common violations." Buttons for 'Discard changes' and 'Save' are at the bottom.
- Content ratings:** Shows the 'Content ratings' section. It displays a summary table with categories like 'Category' (Utility, Productivity, Communication, or Other) and 'Rating' (All ages). Below this, there's a 'Your ratings' section for Brazil, showing a green rating icon and the text 'Rating authority: Classificação Indicativa (ClassInd)'. A note says: 'Ratings shown below may differ from the ratings shown to users on Google Play.' A summary table shows 'Content descriptors' and a note: 'Google and IARC will share your contact information, questionnaire responses, ratings, developer support requests, and app details with rating authorities. [Learn more](#)'.
- Create production release:** Shows the 'Create production release' section. It indicates '2 Warnings' and has a 'Show more' button. Below this is a table for 'New app bundles' with columns: File type, Version, API levels, Target SDK, Screen layouts, ABIs, Required features. An 'App bundle' row is listed with version 1 (1.0), API level 21+, Target SDK 30, 4 screen layouts, 5 ABIs, and 5 required features. A note says: 'Default language - French. Installer l'application. Release notes provided for 2 of 2 languages. [Show all](#)'.

FIGURE 9.3 – Publication sur Google Play Store 3.

# Bibliographie

---

1. Véronique Messager Rota, Gestion de projet vers les méthodes agiles.
2. Rivest, Ronald L (1990), « The MD4 message digest algorithm », in : Conference on the Theory and Application of Cryptography, Springer, p. 303–311.
3. Rivest, Ronald et S Dusse (1992), The MD5 message-digest algorithm.
4. Anderson, Ross et Eli Biham (1996), « Tiger : A fast new hash function », in : International Workshop on Fast Software Encryption, Springer, p. 89–97.
5. Dobbertin, Hans, Antoon Bosselaers et Bart Preneel (1996), « RIPEMD-160 : A strengthened version of RIPEMD », in : International Workshop on Fast Software Encryption, Springer, p. 71– 82.
6. Barreto, PSLM, Vincent Rijmen et al. (2000), « The Whirlpool hashing function », in : First open NESSIE Workshop, Leuven, Belgium, t. 13, p. 14.
7. Rivest, Ronald L et al. (2008), « The MD6 hash function—a proposal to NIST for SHA-3 », in : Submission to NIST 2.3, p. 1–234.
8. Aumasson, Jean-Philippe et al. (2008), « Sha-3 proposal blake », in : Submission to NIST 92.
9. Michail, Haralambos et al. (2005), « A low-power and high-throughput implementation of the SHA-1 hash function », in : 2005 IEEE International Symposium on Circuits and Systems, IEEE, p. 4086– 4089.
10. Glabb, Ryan et al. (2007), « Multi-mode operator for SHA-2 hash functions », in : journal of systems architecture 53.2-3, p. 127–138.
11. Dworkin, Morris J (2015), SHA-3 standard : Permutation-based hash and extendable-output functions, rapp. tech.
12. Dolmatov, Vasily et Alexey Degtyarev (2013), « GOST R 34.11-2012 : hash function », in : Independent Submission, Ed. Request for Comments : Updates 5831, p. 2070–1721.