

République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique  
Dans le cadre du décret 1275 diplôme Master-Label

*Option : Génie Logiciel (G.L)*

*Thème*

**Conception et réalisation d'une application mobile médicale  
Avec flutter**

**Réalisé par :**

- BENHAMMOU Imene.

**Présenté le 02 juillet 2023 devant le jury composé de :**

Mr. BENZIAN Yaghomrasan.

*Président*

Mme. BENOSMAN Amina.

*Examinatrice*

Mr. BENDAHMANE Mohammed Fawzi.

*Expert*

Mme. REMACI Zeyneb Yasmina.

*Encadrante*

## Remerciement

Au seuil de la conclusion de mon parcours d'études, je souhaite exprimer mon sincère remerciement à Dieu le Tout-Puissant. Je le remercie d'avoir illuminé mon chemin et de m'avoir donné la force et la foi nécessaires pour surmonter chaque épreuve. Je suis également reconnaissante envers lui pour toutes les bénédictions, l'inspiration et la guidance qu'il m'a accordée tout au long de mon chemin académique.

À mon encadrante, Mme Remaci yasmina zyneb, je tiens à exprimer mon profond respect pour vos compétences et votre encadrement tout au long de ce projet. Ce projet n'aurait pu être mené à bien sans votre suivi attentif, vos conseils éclairés et votre aide précieuse. Je suis reconnaissante de votre accueil chaleureux et de vos précieux conseils. Je tiens à vous remercier sincèrement pour votre soutien constant et pour avoir été une source d'inspiration inépuisable. Votre expertise et votre dévouement ont été d'une valeur inestimable pour moi, et je suis honorée d'avoir eu la chance de travailler sous votre direction.

À tous les membres du jury, je souhaite exprimer ma profonde gratitude pour avoir accepté de juger ce travail. Votre engagement et votre expertise sont hautement appréciés. Je suis consciente du grand honneur que vous me faite en évaluant mon travail, et je tiens à vous assurer de mon plus grand respect et de mes remerciements sincères.

Je voudrais également exprimer ma reconnaissance envers tous les enseignants du département d'informatique pour leurs connaissances scientifiques et leurs compétences pédagogiques. Leur enseignement a été essentiel à mon développement académique, et je suis reconnaissante de leur dévouement à transmettre leur savoir.

Enfin, je tiens à remercier chaleureusement tous mes proches et toutes les personnes qui, de près ou de loin, m'ont soutenue dans la réalisation de ce travail. Leurs encouragements et leur soutien moral ont été d'une importance capitale, et je leur suis profondément reconnaissante.

## Dédicace

À mon père et à ma mère je souhaite exprimer toute mon affection et ma profonde gratitude pour leur soutien moral et financier, ainsi que pour leurs encouragements tout au long de mes études. Leurs sacrifices, leur amour, leur tendresse, leur soutien inconditionnel et leurs prières ont été d'une valeur inestimable pour moi.

À mes sœurs Hadjer, Sabrina et Hidayet.

À mon petit neveu Khalil.

Dans ce cadre je n'oublie pas les soutiens de ma chère tante Fouzia et mes chers oncles Abdelaziz et Smail, mes grands-parents que Dieu accorde la paix à leurs âmes au paradis.

À ma chère amie Sarra, tu as apporté une couleur spéciale à ma vie, notre amitié est un trésor que je chérirai toujours.

À tous ceux dont je n'ai pas pu citer leur nom sachez que vous êtes toujours présent dans mon cœur.

À tous ceux qui me sont chers, à tous ceux qui m'aiment.

# Résumé

Ce mémoire se concentre sur la conception et la réalisation d'une application mobile médicale en utilisant le Framework Flutter. L'objectif principal est de développer une application mobile sécurisée, simple et conviviale pour tous les groupes d'âge. Cette application permet aux utilisateurs de demander le déplacement d'un personnel médical à domicile et de garder leurs documents médicaux. Pour atteindre cet objectif, nous avons utilisé le langage de programmation Dart dans l'environnement de développement Android Studio. De plus, nous avons opté pour l'utilisation de la plateforme Firebase de Google pour stocker et organiser nos données dans un format facilement compréhensible.

En fournissant cette application, nous visons à simplifier le processus de demande de personnel médical et à offrir aux utilisateurs un moyen pratique pour gérer leurs documents médicaux.

**Mots clé :** Application mobile, Framework, Flutter, Dart, Android Studio, Firebase.

## Abstract

This thesis focuses on the design and implementation of a medical mobile application using the Flutter Framework. The main objective is to develop a secure, simple and user-friendly mobile application for all age groups. This application allows users to request the movement of medical personnel to their homes and keep their medical documents. To achieve this goal, we used the Dart programming language in the Android Studio development environment. In addition, we opted to use Google's Firebase platform to store and organize our data in an easily understandable format.

By providing this application, we aim to simplify the medical personnel application process and provide users with a convenient way to manage their medical documents.

**Keywords :** Mobile application, Flutter, Framework, Dart, Android Studio, Firebase.

## ملخص

تركز هذه الأطروحة على تصميم وتنفيذ تطبيق الهاتف المحمول الطبي باستخدام إطار Flutter. الهدف الرئيسي هو تطوير تطبيق هاتف محمول آمن وبسيط وسهل الاستخدام لجميع الفئات العمرية، يسمح هذا التطبيق للمستخدمين بطلب نقل الموظفين الطبيين إلى منازلهم والاحتفاظ بوثائقهم الطبية. لتحقيق هذا الهدف، استخدمنا لغة برمجة Dart في بيئة التطوير Android Studio. بالإضافة إلى ذلك، اخترنا استخدام منصة Firebase من Google لتخزين بياناتنا وتنظيمها بتنسيق يسهل فهمه.

من خلال تقديم هذا التطبيق، نهدف إلى تبسيط عملية تقديم طلبات الموظفين الطبيين وتزويد المستخدمين بطريقة مريحة لإدارة مستنداتهم الطبية.

**الكلمات الرئيسية:** تطبيق الهاتف المحمول، Flutter، Dart، Android Studio، Firebase

# Table des matières

Introduction générale .....	1
Chapitre I Les applications mobiles.....	3
1.1 Introduction .....	4
1.2 Quelque statistique sur le marché des applications mobiles .....	4
1.2.1 Les usages des applications mobiles .....	4
1.2.2 Les revenus des applications mobiles .....	5
1.2.3 Les applications e-santé .....	5
1.3 Application mobile .....	6
1.3.1 Définition.....	6
1.4 System d'exploitation mobile .....	6
1.4.1 Définition.....	6
1.4.2 Android.....	7
1.4.3 iOS .....	8
1.5 Types des applications mobiles.....	8
1.5.1 Applications natives .....	8
1.5.2 Applications web .....	9
1.5.3 Applications hybrides.....	10
1.6 Flutter.....	19
1.6.1 Widget.....	20
1.6.2 Dart.....	23
1.6.3 Firebase .....	24
1.6.4 Les avantages de Flutter .....	27
1.7 Conclusion.....	27
Chapitre II Conception.....	28
2.1 Introduction .....	29
2.2 Présentation de l'application .....	29
2.3 Planification du projet.....	30
2.4 Etude de l'existant.....	32
2.5 Identification des acteurs.....	34
2.6 Planification des besoins.....	35
2.6.1 Besoin fonctionnel .....	35
2.6.2 Besoin non-fonctionnel.....	35

2.7	UML.....	36
2.7.1	Définition.....	36
2.8	Les diagrammes du système .....	36
2.8.1	Diagramme de cas d'utilisation.....	36
2.8.2	Description textuelle des cas d'utilisation .....	37
2.8.3	Diagramme de séquence .....	40
2.8.4	Diagramme d'activité.....	45
2.9	Conclusion.....	48
	Chapitre III Réalisation .....	49
3.1	Introduction .....	50
3.2	Environnement du travail .....	50
3.2.1	Environnement logiciel .....	50
3.2.2	Environnement matériel .....	56
3.3	Présentation des interfaces .....	56
3.3.1	Interfaces Hommes-Machines .....	56
3.3.2	Interfaces de la plateforme.....	71
3.4	Conclusion.....	73
	Chapitre IV BMC .....	74
4.1	Value propositions (Proposition de valeur) .....	76
4.2	Customer segments (Segments clients).....	76
4.3	Customer relationships (Relation client).....	77
4.4	Channels (Canaux de distribution).....	77
4.5	Key Partners (Partenaires clés) .....	78
4.6	Key activities (Activités clés) .....	79
4.7	Cost structure (Structure de coûts).....	79
4.8	Revenue streams (Sources de revenus).....	80
4.9	Key ressources (Ressources clés) .....	81
	Conclusion générale.....	84
	Références.....	85

# Introduction générale

La prolifération des technologies mobiles a pris une grande ampleur dans de nombreux domaines, y compris le secteur médical. Aujourd'hui, les applications mobiles médicales jouent un rôle de plus en plus important dans la prestation des soins de santé, en offrant aux patients un accès plus rapide et plus pratique aux services médicaux.

Cependant, en Algérie, malgré qu'il existe des initiatives qui visent à créer des applications axées sur la santé, le nombre et les fonctionnalités des applications mobiles médicales restent encore limitées.

De ce fait, en accroissant la sensibilisation des professionnels de la santé et du grand public aux avantages des applications mobiles médicale, le pays peut tirer parti de solutions pratique qui améliorent l'accès aux soins et la gestion individuelle de la santé. Ainsi, il devient essentiel de promouvoir la compréhension des avantages potentiels, de sensibiliser et de fournir une formation pour encourager l'adoption de ces applications.

Dans le cadre de ce projet de fin d'étude, notre objectif consiste à améliorer le système informatique de santé en Algérie, en développant une application mobile médicale novatrice pour assurer une meilleure prise en charge des patients à leurs domiciles. Cette application innovante permet de réduire les contraintes liées aux déplacements et d'offrir une grande commodité aux patients, en particulier ceux qui sont âgés, handicapés ou à mobilité réduite.

Nous souhaitons combler un vide dans le système de santé actuel en offrant une solution pratique et technologiquement avancée pour offrir un soutien médical efficace à domicile, en utilisant les avantages offerts par les technologies mobiles.

Dans un premier temps, nous avons jugé important de présenter en détail le processus de conception et de développement de notre application mobile, en définissant les différentes étapes, allant de l'analyse des besoins et des exigences, à la conception des interfaces utilisateur conviviales, en passant par la mise en œuvre des fonctionnalités clés telles que la gestion des rendez-vous, la localisation des patients.

Notre travail est organisé en quatre chapitres, comme suit :

- Le premier chapitre vise à présenter une étude sur les applications mobiles. Dans ce cadre, nous définissons les différents types des applications mobiles et leurs systèmes d'exploitation. Par la suite, nous examinons l'environnement de développement Flutter, le langage de programmation Dart et la plateforme Firebase.
- Le deuxième chapitre présente notre application mobile. Le chapitre dans une première partie fait le point sur l'analyse des besoins et la conception de l'application.
- Le troisième chapitre s'attarde à décrire la contribution que nous apportons pour offrir des solutions pratiques et efficaces aux patients qui ont une mobilité réduite.
- Le quatrième chapitre présente en détail notre Business Model Canvas (BMC) pour notre application médicale, nous passons en revue les neuf éléments clés du BMC. En décrivant chaque élément du BMC, nous mettons en évidence notre stratégie globale pour générer des revenus, atteindre notre public cible, établir des relations solides avec nos clients et optimiser nos ressources.
- Nous terminerons ce rapport par une conclusion générale, et quelques perspectives.



# Chapitre I Les applications mobiles

## 1.1 Introduction

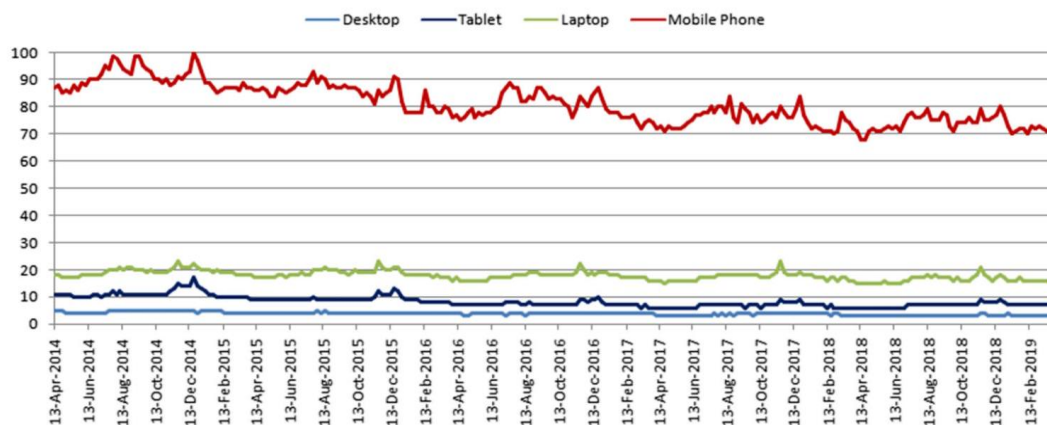
Dans un monde moderne où le smartphone et les applications mobiles font désormais partie de notre culture quotidienne, il est devenu motivant pour les développeurs de s'impliquer dans le domaine du développement mobile.

Dans ce chapitre, nous présenterons une analyse du marché des applications mobiles, en mettant particulièrement l'accent sur les systèmes d'exploitation. Nous explorerons également les différents types d'applications mobiles, ainsi que les technologies les plus utilisées pour le développement. Enfin, nous consacrerons une partie importante de ce chapitre à la présentation de la technologie Flutter.

## 1.2 Quelques statistiques sur le marché des applications mobiles

### 1.2.1 Les usages des applications mobiles

Selon les statistiques fournies par Google Trends data [1], l'intérêt des utilisateurs pour les smartphones est de plus en plus important par rapport à d'autres plateformes telles que les ordinateurs de bureau, les ordinateurs portables ou les tablettes, au cours des années 2014 à 2019.



**Figure I-1** Tendances d'intérêt des utilisateurs au fil du temps. [1]

Les études montrent que l'utilisateur moyen de smartphone passe environ 3 heures et 10 minutes sur son appareil chaque jour. De cela, 2 heures 51 minutes (environ 90 % du temps passé sur smartphone) sont consacrées aux applications. D'autres recherches montrent que l'utilisateur moyen vérifie son téléphone toutes les 12 minutes. En fait, selon Social Media Today<sup>1</sup>, 49 % des personnes ouvrent des applications 11 fois ou plus par

<sup>1</sup> Social Media Today : est une publication leader de l'industrie exploitée par Industry Dive, qui fournissons une analyse originale sur les derniers événements dans l'industrie des médias sociaux.

jour, et près de 2 % des utilisateurs ouvrent des applications 100 fois ou plus par jour. Ils rapportent également que le smartphone moyen a plus de 80 applications installées.

Cependant, l'utilisateur moyen n'utilise que 10 applications mobiles par jour et utilise environ 30 applications mobiles par mois. Cela signifie que plus de 60 % des applications mobiles sont téléchargées mais pas utilisées régulièrement. En fin de compte, ces chiffres montrent que la croissance des applications mobiles n'est pas seulement tirée par les téléchargements, elle est également tirée par l'utilisation et la rétention des consommateurs. Cela confirme que fournir une excellente expérience utilisateur et des fonctionnalités à valeur ajoutée est plus important que jamais si vous souhaitez que vos utilisateurs utilisent régulièrement votre application mobile. [2]

### 1.2.2 Les revenus des applications mobiles

Les statistiques indiquent que les revenus des applications mobiles augmentent de manière significative.

De 97,7 milliards de dollars en 2014, l'industrie n'a cessé de croître chaque année. En 2020, le marché des applications mobiles a clôturé à 581,9 milliards de dollars de revenus. Il devrait continuer de grimper, générant environ 693 milliards de dollars en 2021, 808,7 milliards de dollars en 2022 et 935 milliards de dollars d'ici 2023. [2]

### 1.2.3 Les applications e-santé

Actuellement, le marché des applications mobiles de santé connaît une croissance rapide, notamment après la crise sanitaire causée par la pandémie COVID-19.

Selon Health Market Outlook de Statista<sup>2</sup>, parmi sur les 2,8 millions d'applications disponibles sur Google Play et les 1,96 million d'applications disponibles sur l'App Store, environ 99 366 appartiennent aux catégories médicales, de santé et de fitness. Ces applications représentent environ 2% (47 890) de celles disponibles sur Google Play et environ 3% (51 476) de celles disponibles sur l'App Store.

Il est estimé que d'ici 2025, les applications mobiles médicales (mHealth apps) vont générer plus de 111.1 milliards de dollars de revenus [3]. De plus, le marché de l'e-santé pour les consommateurs (B2C) devrait connaître une augmentation de plus de 60 % entre 2021 et 2025, atteignant environ 78 milliards d'euros. [4]

---

<sup>2</sup> Health Market Outlook de Statista : est un portail en ligne allemand offrant des statistiques issues de données d'instituts, dans 170 secteurs différents.

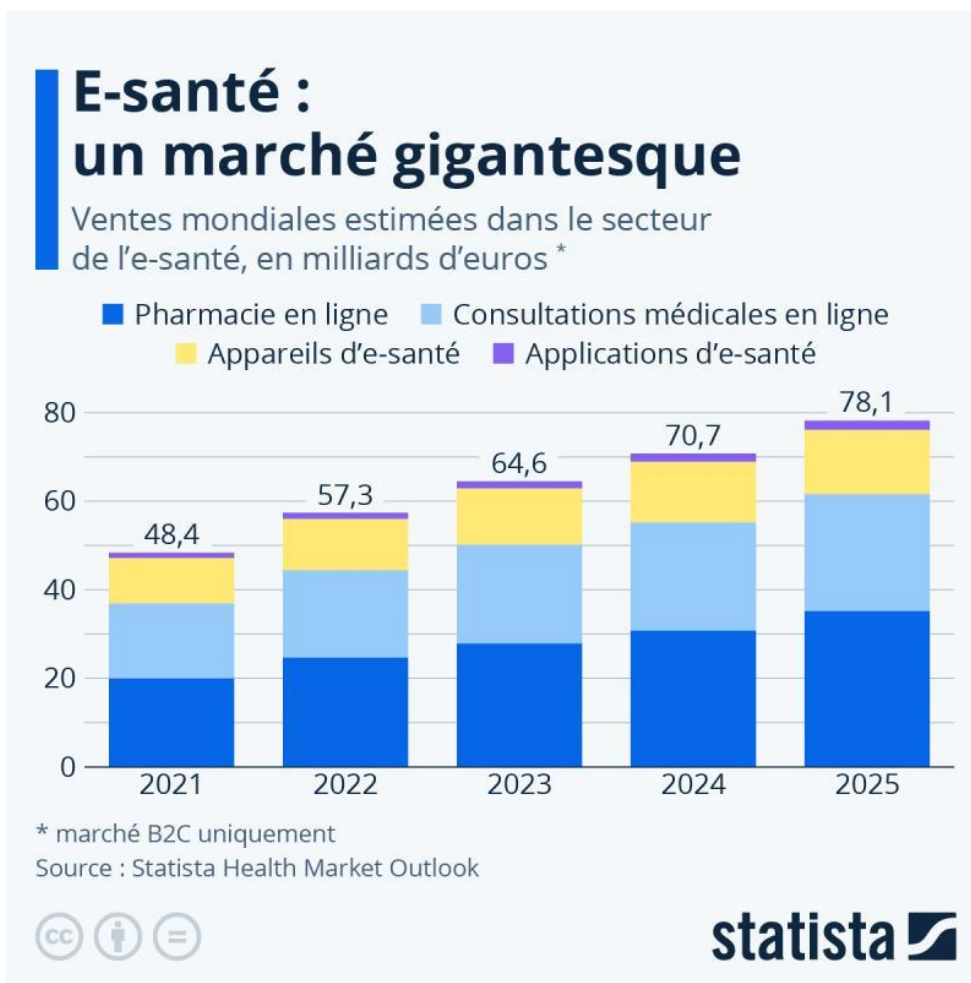


Figure I-2 E-santé. [4]

## 1.3 Application mobile

### 1.3.1 Définition

Une application mobile est un logiciel applicatif développé pour être installé sur un appareil mobile, généralement un téléphone mobile, un téléphone intelligent ou une tablette numérique. Les applications mobiles : sont des programmes relativement légers conçus pour des plates-formes mobiles et sont utilisées pour des services, de l'information, des médias sociaux, des jeux, etc. [5]

## 1.4 System d'exploitation mobile

### 1.4.1 Définition

Un système d'exploitation, abrégé OS pour Operating System en Anglais, est en réalité une multitude de programmes combinés ensemble qui permettent de faire l'interface entre le matériel informatique et les applications développées par les

développeurs. Chaque système d'exploitation dispose de fonctionnalités spécifiques et propres à ce dernier.

Concrètement, le système d'exploitation sert de base au développement et au fonctionnement des applications. Il sert d'interface entre le matériel informatique et les applications.

Chaque système d'exploitation dispose de ses propres langages de développement (C++, Objective C, Swift pour iOS, Java et C++ pour Android) et de ces propres documentations pour créer des applications compatibles. Ceci explique en partie pourquoi une application est parfois uniquement disponible sur iOS ou Android (le développement est totalement différent pour rendre une application compatible avec plusieurs systèmes d'exploitation). [6]

Il existe plusieurs systèmes d'exploitation mobile, notant : BlackBerry, Bada, iOS, Symbian Windows Phone, RIM , Ubuntu Touch, Firefox OS, Android, etc.

Cependant, le marché mondial des OS mobiles est aujourd'hui dominé par deux grandes entreprises : Google et Apple, qui développent respectivement les systèmes d'exploitation Android, iOS.

### 1.4.2 Android

Android est un système d'exploitation mobile développé par Google. Il est utilisé par plusieurs smartphones et tablettes. Les exemples incluent le Sony Xperia, le Samsung Galaxy et le Google Nexus One.

Le système d'exploitation (OS) Android est basé sur le noyau Linux. Contrairement à iOS d'Apple, Android est open source, ce qui signifie que les développeurs peuvent

modifier et personnaliser le système d'exploitation pour chaque téléphone. Par conséquent, différents téléphones basés sur Android ont souvent des interfaces utilisateur graphiques différentes, même s'ils utilisent le même système d'exploitation. [7]



**Figure I-3** Android. [31]

### 1.4.3 iOS

Lorsque l'on parle de iOS, on fait référence au système d'exploitation (Operating System) propre à la marque Apple et qui concerne donc toutes les gammes d'appareils de types iPhone, iPod, le iPad et les périphériques compatibles. Il est utilisé par les générations successives d'iPhone depuis la sortie du premier modèle en 2007 puis par l'iPad depuis 2010. Les puristes seront curieux de savoir que les applications qui sont développées pour le iOS sont codées en Swift ou bien en Objective-C.

A contrario de son principal rival, Android de Google, qui peut être repris et adapté par n'importe quel fabricant de terminaux mobiles, iOS est un système d'exploitation propriétaire sur lequel Apple exerce un strict contrôle. [8]



Figure I-4 iOS. [32]

## 1.5 Types des applications mobiles

Les applications mobiles sont regroupées en trois catégories : applications native, applications web et applications hybride.

### 1.5.1 Applications natives

Une application native est développée spécifiquement pour un système d'exploitation : iOS ou Android. Elle est installée directement sur le smartphone et peut fonctionner sans connexion Internet, selon son objectif et sa nature.

Pour l'installer, le mobinaute passe par l'intermédiaire d'un store tel que Google Play Store ou Apple Store. L'utilisateur n'a pas besoin d'un outil ou d'un navigateur pour la lancer. Les données qu'elles génèrent sont stockées dans la mémoire de l'application ou dans le cloud, selon la configuration de l'appareil mobile.

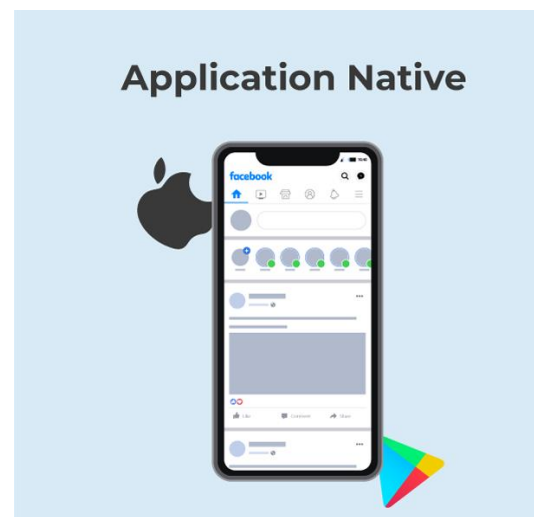


Figure I-5 Application native. [33]

Les applications natives tirent pleinement parti des fonctionnalités de l'appareil (sous conditions d'acceptation par l'utilisateur), comme : Le GPS, L'appareil photo, Le micro, La liste des contacts, Les fichiers multimédias. [9]

### 1.5.1.1 Les avantages des applications native :

- Le principal avantage de l'application native, c'est son temps de réaction. Du fait que l'appli soit installée directement dans la mémoire du mobile, elle est stable, responsive et exploite parfaitement et rapidement tous ses modules, parfois même sans connexion à internet.
- En optimisant le code, elle permet de créer une application performante et sur-mesure et de développer des fonctionnalités avancées adaptées aux besoins des utilisateurs. Elle convient donc parfaitement aux projets les plus complexes. [10]

### 1.5.1.2 Les inconvénients des applications native

- Les développeurs capables de créer une application en natif sont rares. Donc ils reviennent cher.
- Les codes étant radicalement différents d'un OS à un autre, il faut multiplier par deux les coûts de développement pour être présent sur Apple et sur Android. Les développeurs sont contraints de s'adapter au code imposé par Apple et Google qui évolue en permanence.
- Les applications natives dépendent des « stores » (boutiques de distribution des applications : App Store / Google Play...) ce qui complique les évolutions et les mises à jour. [10]

## 1.5.2 Applications web

Une application web est un programme de type client-serveur qui s'exécute sur le web et rend un service. [11]

Une application web est une application hébergée sur un serveur et accessible depuis un navigateur web. Contrairement à une application mobile, aucune installation n'est nécessaire ouvrant la porte à de nombreux avantages. [12]

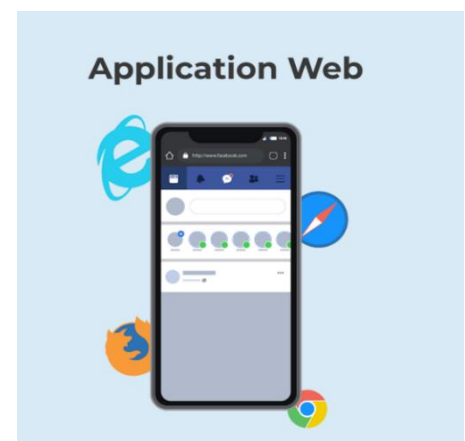


Figure I-6 Application web. [33]

### 1.5.2.1 Les avantages des applications web

- Aucun téléchargement ni aucune mise à jour ne sont nécessaires, ce qui évite de « polluer » la mémoire de l'appareil.
- Les applications web mobilisent peu de ressources financières puisque leur développement et leur déploiement sont rapides et faciles. Elles représentent donc une solution peu onéreuse pour atteindre les internautes mobiles.
- Elles sont également plus ouvertes aux évolutions et ne présentent aucun problème de compatibilité avec les systèmes d'exploitation du mobile. [10]

### 1.5.2.2 Les inconvénients des applications web

- Etant créée pour les navigateurs web mobile (Safari, Google Chrome etc.), elle requiert cependant une connexion à internet pour fonctionner.
- L'utilisation des fonctionnalités du téléphone reste limitée avec le web App, offrant ainsi une expérience utilisateur plus sommaire, moins intuitive et moins interactive.
- De même, il n'y a pas de possibilités de notifications push, d'alertes ou de pastilles sur l'appareil et la rapidité d'exécution est plus faible comme elle dépend du débit internet. Elle n'est pas non plus optimisée pour tous les formats d'écran. [10]

## 1.5.3 Applications hybrides

Situé entre l'application mobile et web, l'application hybride est accessible sur toutes les plateformes d'application. Il s'agit d'une application mobile qui fusionne entre les caractéristiques d'une application web (développement en langage HTML 5) et celles de l'application native citées ci-dessus. C'est une technologie multi-plateforme, ce qui est un gain de temps considérable si le projet n'est pas trop complexe et ne nécessite pas le développement de plugins spécifiques.

Ce genre d'application mobile minimise les charges et la durée de son développement même si cela sera au détriment du perfectionnement et de la qualité qui caractérise l'application native.



Figure I-7 Application hybride. [34]



Les applications cross-platforms sont réputées moins performantes et moins fluides que les applications natives. [13]

### 1.5.3.1 Les avantages des applications hybrides

Les applications hybrides offrent plusieurs avantages :

- Allier l'expérience utilisateur avec l'agilité du cycle de développement et des coûts maîtrisés.
- Contourner les limitations de l'App store Apple : pour déployer une application sur l'App store Apple, il faut soumettre l'application et attendre sa validation. Le temps de validation variera en fonction du moment de l'année, mais aussi de s'il s'agit d'une première version de l'application ou bien d'une mise à jour. Il est généralement compris entre 1 à 7 jours. Les applications hybrides présentent donc un grand avantage pour les développeurs souhaitant pouvoir mettre à jour leur application fréquemment puisqu'il n'est pas nécessaire de resoumettre la nouvelle version si les éléments modifiés ne touchent pas au code natif.
- Trouver des talents : la plupart des applications ont une version iOS et une version Android. Elles sont donc développées en utilisant le langage de programmation correspondant : Objective-C ou Swift pour iOS, Java pour Android. Les applications hybrides permettent quant à elles l'utilisation de langages utilisés de façon fréquente par les développeurs web (HTML, JavaScript et CSS) qui peuvent donc utiliser leurs connaissances. Il est ainsi plus facile de trouver des talents pour développer une application hybride.
- Réutiliser le code de la partie web app : le code est écrit une fois et déployé ensuite sur toutes les plateformes mobiles.
- Réduire les temps de développement et les coûts : le code étant écrit une fois, cela réduit considérablement les temps de développement et les coûts par rapport aux applications natives nécessitant d'une part un développement pour iOS et d'autre part un développement pour Android. [14]

### 1.5.3.2 Les inconvénients des applications hybride

- Une interface utilisateur limitée : les applications hybrides on en effet un design qui n'a pas le ressenti d'un design natif. L'interface utilisateur n'est donc pas aussi fluide. Les possibilités (3D par exemple) sont aussi limitées du fait de l'utilisation du WebView qui ne permet pas d'exploiter tout le potentiel du device.

- La difficulté ou l'impossibilité d'exploiter toutes les capacités des plateformes : chaque plateforme, iOS ou Android, a des capacités uniques qu'un développeur peut souhaiter exploiter. Si tel est le cas, c'est en utilisant une combinaison de plugins et de code spécifique à la plateforme que cela pourra être réalisé pour une application hybride. Cela complexifie d'autant le projet de développement de l'application.
- Des performances et des transitions entre les pages plus lentes
- Une dépendance vis à vis de la vitesse du navigateur.

Il existe plusieurs Framework pour le développement des applications mobiles hybrides.

Les principaux Framework sont Ionic, Flutter, React Native et Xamarin. Dans la suite, nous allons présenter les détails de chaque plateforme.

### 1.5.3.3 React Native

Framework créé par Facebook, React Native utilise comme langages de programmation Javascript et (couplé à l'implémentation mobile du Framework ReactJS). Il se distingue des applications hybrides par le fait qu'il emploie les composants natifs du téléphone.

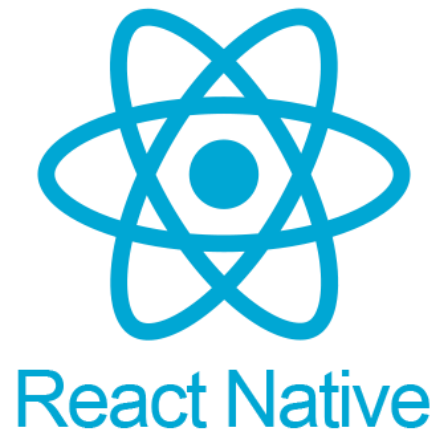
Le Javascript est langage souple et rapidement assimilable. Il permet une grande diversité dans son écriture et dans sa syntaxe. L'orientation composants du Framework permet également de réutiliser différentes parties du code sans avoir à les réécrire.

L'entreprise ne développera qu'une seule application pour Android et iOS. L'équipe de développement de l'entreprise pourra mener à succès un projet alliant gain de temps, gain budgétaire et performance.

Autre facteur qui favorise le développement : l'existence de nombreux outils préconçus et utilisables directement pour le développement. Vous n'avez pas à les créer à partir de rien.

Également React-Native est le Framework de développement mobile possédant la plus grande communauté. Ce point n'est vraiment pas à négliger car une communauté fournie permettra de résoudre plus facilement vos éventuellement blocages techniques.

[15]



**Figure I-8** Logo React Native. [33]

#### 1.5.3.4 Flutter

Google a développé ce cadre d'interface pour offrir à ses utilisateurs une belle et immersive interface. Flutter est entièrement compilé en code ARM natif, par opposition aux applications mobiles qui emploient des vues Web. Une application construite avec Flutter offre une sensation native. Skia est un moteur graphique open Source populaire en 2D, propre à Flutter. Le développement de ce Framework se fait sur Dart open Source. Une application construite sur ce cadre se caractérise par sa rapidité, environ 60ips. D'autres développeurs atteignent même les 120ips.

Cependant, pour développer un tel cadre, les développeurs devront s'approprier un nouveau langage, inconnu auparavant. Si vous voulez développer un jeu ou une application qui requiert plusieurs fonctions propres à l'appareil, Flutter n'est pas à priori le plus adapté pour vous. [15]



**Figure I-9** logo flutter. [34]

#### 1.5.3.5 Ionic

Le Framework multiplateforme Ionic est élaboré avec Apache Cordova et Angular. Avec ce cadre, les applications mobiles développées sont utilisables sur plus

d'une plateforme mobile. Elles fonctionnent sur Android et IOS. La conception de ces applications est classique. Elle se fait à l'aide de Html, d'Angular ou de CSS.

En tant que cadre multiplateforme, Ionic permet aux entreprises de développer leur application mobile avec un minimum de dépenses. L'entreprise n'a pas besoin de faire appel à plusieurs ingénieurs logiciels pour la conception d'une application mobile.

En plus de faire économiser de l'argent, le Framework Ionic permet de gagner en temps. Demandant moins de ressources, moins de jours homme, l'entreprise pourra mettre son application plus tôt à la disposition du public et être plus réactif dans le lancement de nouvelles fonctionnalités.

Le véritable inconvénient d'Ionic est sa performance comparée à une autre native. Si l'entreprise désire concevoir une application exigeant des traitements graphiques pointus, Ionic n'est certainement pas le choix idéal. Le souci majeur d'Ionic est sa performance. En effet comme cité précédemment, Ionic se sert d'une WebView afin d'afficher son contenu (qui est fabriqué grâce à des composants web) et cela bride grandement les performances. [15]



**Figure I-10** logo Ionic. [35]

### 1.5.3.6 Xamarin

Xamarin est une plateforme open source qui permet de générer des applications modernes et performantes pour iOS, Android et Windows avec .NET. Xamarin est une couche d'abstraction qui gère la communication entre le code partagé et le code de plateforme sous-jacent. Xamarin s'exécute dans un environnement managé qui offre des commodités telles que l'allocation de mémoire et le garbage collection.

Xamarin permet aux développeurs de partager en moyenne 90 % de leur application entre les plateformes. Ce modèle permet aux développeurs d'écrire toute leur logique métier dans un seul langage (ou de réutiliser du code d'application existant) tout

en obtenant des performances, une apparence et une convivialité natives sur chaque plateforme.

Les applications Xamarin peuvent être écrites sur PC ou Mac et être compilées dans des paquets d'application natifs, par exemple un fichier .apk sur Android ou un fichier .ipa sur iOS. [16]



Figure I-11 logo Xamarin. [36]

### 1.5.3.7 React Native vs Flutter vs Ionic vs Xamarin

Avant de déterminer les meilleurs outils pour le développement d'applications mobiles multiplateformes, comprenons les caractéristiques sur lesquelles nous allons comparer ces quatre outils : [17]

✓ **Les langages de programmation :**

Les langages de programmations utilisée par chaque Framework : [17]

1. React Native : React Native est un Framework sponsorisé par Facebook. Il utilise JavaScript, qui est l'un des langages les plus populaires et performants. De plus, il combine les avantages de ReactJS et JavaScript.

Il permet au développeur d'écrire des composants ou des modules en Objective-C, Java et Swift. Grâce à l'utilisation des bibliothèques et des modules natifs, il est possible de gérer des opérations complexes telles que le traitement vidéo et l'édition d'images. Il permet aussi de gérer des opérations non conventionnelles qui ne sont pas incluses dans l'API du Framework.

2. Ionic : Le Framework Ionic utilise JavaScript, CSS et HTML5 et nécessite un wrapper Cordova pour utiliser les contrôleurs de la plate-forme native. Le noyau Ionic est basé sur AngularJS, qui est un Framework JavaScript populaire et est écrit en SaaS.

Le langage principal d'Ionic est TypeScript, qui est un sous-ensemble syntaxique de JavaScript. Typescript aide à identifier et éliminer les erreurs, ce qui améliore la qualité du code.

Xamarin : Xamarin utilise la programmation C# et .NET pour développer des applications Mac, Android et iOS. Généralement, les développeurs ne peuvent pas utiliser directement les bibliothèques open source natives pour IOS et Android. Cependant, ils peuvent utiliser les bibliothèques .NET pour pallier cette limite et répondre à ces exigences.

3. Flutter : Flutter utilise Dart pour le développement d'applications IOS et Android et peut être utilisé pour créer des applications de haute qualité.

✓ **Performances :**

Les performances sont un facteur important à prendre en compte lors de la création d'applications mobiles. [17]

1. React Native : Les performances de son application sont assez similaires à celles d'autres applications natives car elles utilisent les éléments du code qui sont destinés à l'API React Native. En fait, React Native permet aux développeurs d'utiliser des modules natifs lorsqu'ils souhaitent scripter du code pour des opérations complexes.

Étant donné que les modules natifs sont scriptés dans des langues natives, cela le rend plus facile et plus accessible pour les développeurs. Ces modules natifs offrent des performances élevées. Cependant, l'inconvénient est qu'il ne peut pas être utilisé sur d'autres plates-formes. [17]

2. Xamarin : les performances de Xamarin sont assez similaires aux performances natives. Vous pouvez développer des applications mobiles de deux manières avec Xamarin, Xamarin.forms, Xamarin.android et Xamarin.IOS.

Les applications ont été développées avec Xamarin.android et xamarin.ios fonctionnent de la même manière que les applications natives, et la fonctionnalité multiplateforme des applications est basée sur la logique métier plutôt que sur les bases de code. De plus, vous obtenez des performances élevées en tant qu'applications natives.

Bien que Xamarin.forms se concentre sur le partage de code étendu, il est moins spécifique à la plate-forme, ce qui réduit les performances du code par rapport au natif pur et au natif Xamarin. [17]

3. Ionic : Ionic n'offre pas des performances similaires aux applications natives. Bien qu'il n'utilise pas de composants natifs, son apparence est assez similaire aux applications natives, qui sont réalisées via les technologies Web. La doublure argentée de l'utilisation d'Ionic est que les tests d'applications se déroulent rapidement, ce qui les rationalise et les distingue dans le processus de développement d'applications. [17]
4. Flutter : Indéniablement, Flutter surpasse toutes les autres applications en termes de performances. Il offre une grande vitesse car il utilise des fléchettes. [17]

✓ **Interface utilisateur :**

L'interface utilisateur doit être attrayante et simple pour attirer l'attention des utilisateurs. [17]

1. React Native : Ses modules interagissent avec les contrôleurs d'interface utilisateur natifs qui améliorent l'interface utilisateur, comme les applications natives. En outre, il utilise les bibliothèques ReactJS pour le développement de l'interface utilisateur.
2. Xamarin : lors du développement de l'interface utilisateur dans Xamarin, vous pouvez utiliser à la fois Xamarin.android et Xamarin.ios. Le développement de l'interface utilisateur à l'aide de Xamarin.android et iOS prend du temps, mais ressemblera aux applications natives.  
Alors que dans Xamarin.forms, cela prend moins de temps et le processus de développement devient plus facile. Cependant, le développement peut ne pas ressembler aux applications natives. Ce développement est généralement utilisé par les entreprises où l'interface utilisateur n'est pas prioritaire.
3. Ionic : Ionic développe une interface utilisateur en utilisant CSS et HTML et n'utilise pas ses éléments natifs. En fait, il utilise Cordova pour obtenir un look natif. À l'heure actuelle, Ionic a une nouvelle disposition d'interface utilisateur native et des éléments qui n'existaient pas auparavant. Ils ne sont disponibles que sur Android et IOS avec un SDK natif.
4. Flutter : Flutter est l'interface utilisateur parmi tous. En comparaison avec React Native, Flutter et React Native ont une grande efficacité et des performances élevées.

**✓ Réutilisabilité du code :**

La réutilisabilité du code vous donne une idée de la façon dont vous pouvez réutiliser le code existant pour les deux plates-formes. [17]

1. React Native : React Native utilise des composants natifs dans Swift, Objective-C et Java pour améliorer les performances de l'application. Généralement, quatre-vingt-dix pour cent de ces composants peuvent être réutilisés sur d'autres plates-formes, et le reste des composants natifs de la base de code doit être accepté. Les développeurs doivent modifier une base de code spécifique.
2. Xamarin : dans Xamarin, quatre-vingt-seize pour cent du code source avec Xamarin.forms peuvent être réutilisés. En fait, cela améliore et accélère le processus de développement.
3. Ionic : Généralement, Ionic fonctionne bien sur toutes les plates-formes de système d'exploitation. Cependant, certains composants doivent être modifiés en fonction des plates-formes, pour lesquelles il faut fournir un effort supplémentaire.
4. Flutter : dans React Native, il existe une fonction toujours prête à l'emploi qui améliore la vitesse de développement. Dans Flutter, vous devez ajouter des fichiers dédiés sur les plates-formes Android et iOS, qui dépendent des règles.



### 1.5.3.8 Résumé des caractéristiques de React Native, Xamarin, Ionic et Flutter

Feature	React Native	Xamarin	Ionic	Flutter
Platform Support	iOS, Android	iOS, Android	iOS, Android	iOS, Android, Web, Desktop
Language	JavaScript	C#	JavaScript	Dart
UI Components	Native	Native	Web	Custom
Performance	Good	Good	Average	Excellent
Development Speed	Fast	Fast	Fast	Very Fast
Community Support	Large	Large	Large	Growing
Testing	Good	Good	Good	Excellent
Code Sharing	High	High	Medium	High
IDE	Visual Studio	Visual Studio	Visual Studio Code, Visual Studio	Android Studio, IntelliJ IDEA
Cost	Free	Free + Paid	Free	Free

**Figure I-12** flutter vs React vs Ionic vs Xamarin. [17]

## 1.6 Flutter

Après avoir examiné les informations fournies sur les différents Frameworks de développement d'applications mobiles, nous avons décidé d'utiliser Flutter pour développer notre application.

Flutter est un kit de développement multiplateforme dévoilé par Google pour la première fois en 2015. Il utilise le langage de programmation DART qui est également une création de Google, inventé en 2011.

Flutter est utilisé pour concevoir des applications cross plateforme, c-à-d. qu'on crée une base de code commune à tous les Os (Apple et Android...). Ces applis s'adaptent aussi bien aux systèmes d'exploitation mobiles qu'à ceux intégrés au web, ordinateurs et

TV connectés et au web. Ainsi, ce kit de développement est bel et bien compatible avec : IOS, Android, Windows, Linux, Mac, Google Fuchsia...

Flutter vient pallier l'ensemble des limites qui pouvaient découler de la conception d'applications à l'aide d'un outil monoplateforme. La principale limite était d'ordre financier, car les développeurs web se retrouvaient dans l'obligation de mettre en place diverses équipes, pour concevoir plusieurs versions d'une même application. C'était autrefois la seule solution pour développer une application exploitable sur différentes plateformes. [18]

Les principales composantes de Flutter sont :

### 1.6.1 Widget

Les widgets Flutter sont les principaux composants d'une application. Ils donnent l'apparence en fonction de leur configuration et de leur état.

Dans Flutter, tout ce qui apparaît dans l'interface utilisateur s'appelle widget et ils héritent tous de la classe Widget. Lorsque vous créez une interface utilisateur dans Flutter, vous le faites en attachant chaque widget à l'écran de l'application afin qu'il s'adapte exactement à l'endroit où vous le souhaitez. [19]

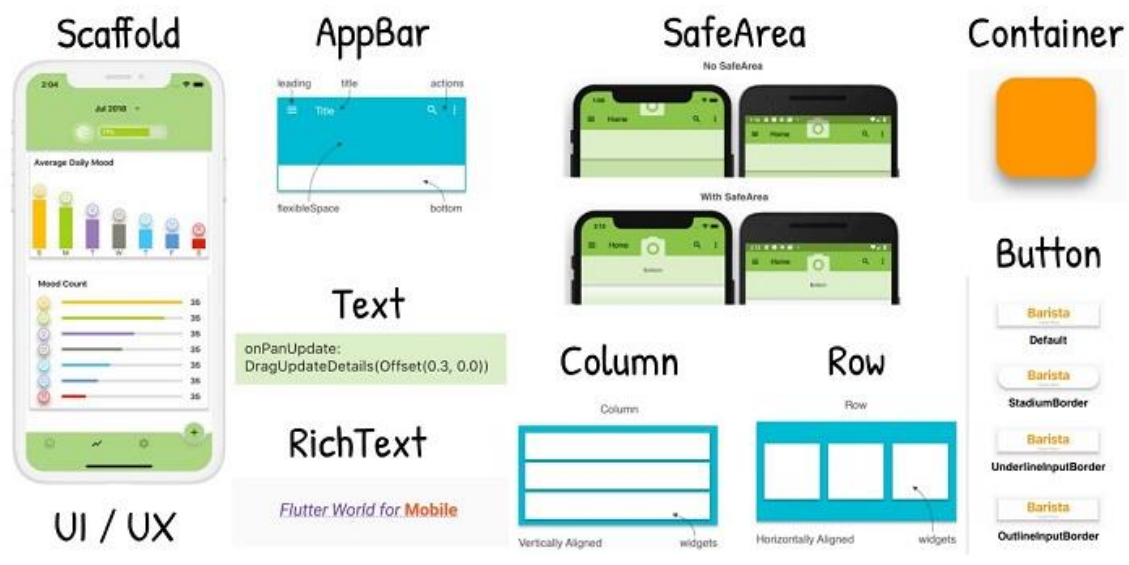


Figure I-13 les widgets. [20]

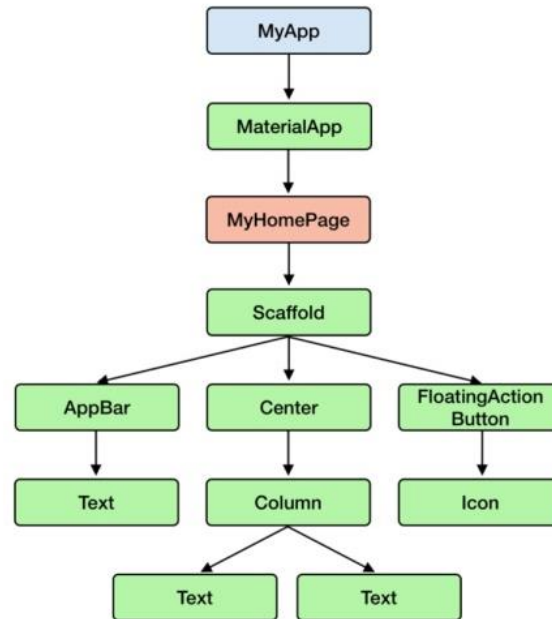
Google distingue 14 catégories de Widgets : [21]

- Accessibilité : permet d'ajouter de l'accessibilité à l'application
- Animation et mouvement : gère les animations et déplacements des widgets
- Assets, images et icônes : gère les assets tels que l'affichage des images et icônes.
- Async : permet de profiter de fonctionnalités asynchrones.

- Bases : ce sont les éléments de base de toute application Flutter
- Cupertino : ces widgets ont été spécifiquement conçus pour simuler les éléments d'affichage propre à iOS
- Input : prend en charge les entrées telles que saisie de texte et de formulaires.
- Modèles d'interaction : gère le tactile et la navigation.
- Mise en page : permet de disposer l'ensemble de widgets, en les alignant ou les ordonnant en grille, lignes, colonne ...
- Composants Materials : widgets reposant sur Material Design et respectant les lignes directrices de Google.
- Peinture et effets : prend en charge les caractéristiques du widget telles que la disposition, la taille et la position, mais aussi les effets visuels
- Défilement : gère le scroll (défilement) des composants enfants contenus dans le widget parent
- Style : gère le thème et l'adaptation des éléments en fonction de la taille de l'écran (responsive design)
- Texte : prend en charge l'affichage de texte.

### 1.6.1.1 L'arborescence des widgets

Les widgets Flutter sont organisés dans une arborescence de parents-enfants, pour former ce que vous voyez à l'écran. Cette arborescence est tous les widgets que vous utilisez pour construire l'application, c'est-à-dire que le code que vous écrivez créera cette structure. [19]



**Figure I-14** L'arborescence de widget. [37]

### 1.6.1.2 L'état des widgets Flutter

Chaque widget Flutter peut être sans état ou avec état. La principale différence est la possibilité de restituer les widgets au moment de l'exécution. Le widget sans état ne sera utilisé qu'une seule fois et est immuable. Par contre un widget avec état peut être utilisé plusieurs fois en fonction du changement de son état interne. [19]

- StatefulWidget :

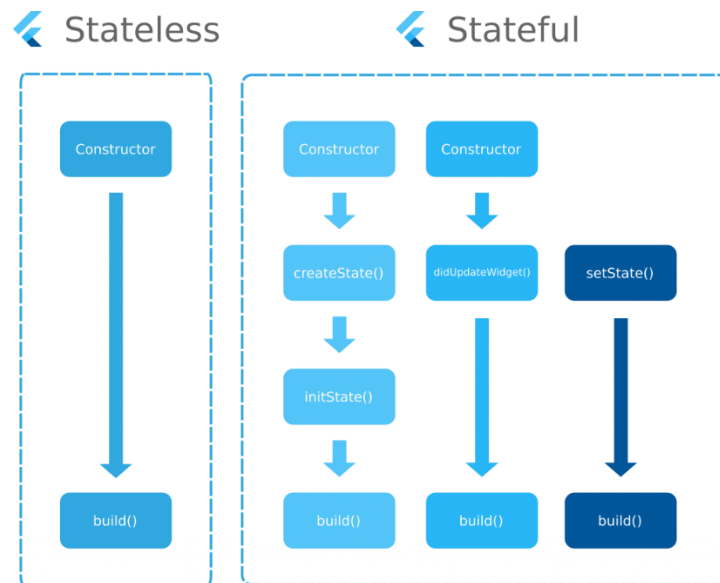
Un widget avec état est utilisé lorsqu'une partie doit changer de manière dynamique pendant l'exécution. En fait, il peut subir plusieurs modifications pendant l'exécution de l'application. Donc, il peut suivre les modifications et mettre à jour l'interface utilisateur.

Les widgets avec état sont utiles lorsque la partie de l'interface utilisateur que nous décrivons change de manière dynamique. Si nous créons un bouton qui se met à jour chaque fois qu'un utilisateur clique dessus, il s'agit d'un widget avec état. [19]

- StatelessWidget :

Un widget sans état ne peut pas changer pendant l'exécution d'une application Flutter. Cela signifie qu'il est impossible de le modifier pendant que l'application est en action. Pour cette raison, l'apparence et les propriétés restent inchangées pendant toute la durée de vie du widget.

De plus, il ne stocke pas les valeurs susceptibles de changer. Les widgets sans état peuvent être utiles lorsque la partie de l'interface utilisateur que nous décrivons ne dépend d'aucun autre widget. Par exemple, le texte, les icônes et les boutons d'icônes. [19]



**Figure I-15** Stateless vs Stateful. [38]

## 1.6.2 Dart

Dart est un langage de programmation, qui a été et est développé principalement par Google. Il est standardisé par Ecma (Ecma est une organisation internationale de normalisation des systèmes d'information et de communication ainsi que des appareils électroniques grand public). La programmation dans Dart doit être une alternative attrayante à JavaScript dans des navigateurs Web modernes. D'après les développeurs de Dart, les faiblesses de JavaScript ne peuvent plus être corrigées par son développement.

Le langage Dart est développé depuis 2010 et lancé la même année. Comme les navigateurs ne pouvaient et ne peuvent pas le manier en mode natif et que JavaScript peut être exécuté dans tous les navigateurs modernes, il existe le Compiler Dart2js, autrement dit « Dart vers JavaScript ». Le langage Dart ressemble aux langages de programmation orientés objet établis, dont Swift, C# ou Java font partie, qui sont soumis à des paradigmes de programmation déterminés. Les règles de combinaison de signes établis, c'est-à-dire la syntaxe, ressemblent au langage de programmation C. Cette « parenté » facilite grandement la prise en main de sorte que l'on peut y accéder sans avoir à surmonter trop d'obstacles. [22]



**Figure I-16** logo Dart. [39]

## 1.6.3 Firebase

### 1.6.3.1 Définition

Firebase est le nom d'une plateforme mobile de Google qui facilite la création de backend à la fois scalable et performant. En d'autres termes, il s'agit d'une plateforme qui permet de développer rapidement des applications pour mobile et pour le web.

L'objectif de la création de [firebase.google.com](https://firebase.google.com) en 2011 par James Tamplin et Andrew Lee est d'éviter aux professionnels et aux particuliers de s'engager dans un processus complexe de création et de maintenance d'une architecture serveur.

De plus, la plateforme peut être exploitée par plusieurs utilisateurs en même temps sans connaître un quelconque bug. La praticité est également au rendez-vous grâce à ses fonctionnalités intuitives. Depuis le rachat de la plateforme par Google en 2014, Firebase SDKs a connu de nombreuses améliorations et n'a de cesse de satisfaire ses utilisateurs. [23]



**Figure I-17** logo Firebase. [40]

### 1.6.3.2 Le secret de Firebase

Dans Firebase, vous trouverez des API intuitives regroupées dans un SDK unique. Ces API, en plus de vous faire gagner du temps, vous permettent de réduire le nombre d'intégrations que vous devez gérer par le biais de votre application.

Vous profiterez ainsi d'une offre sur mesure ainsi qu'une intégration étroite entre les différents produits que vous exploitez. Étant donné que Firebase utilise l'infrastructure de Google, la plateforme n'a aucun mal à s'adapter à l'évolution de votre application.

Ainsi, vous pourrez développer votre Firebase appli dans les meilleures des conditions, d'autant plus que la plateforme met à votre disposition une solution complète, évolutive et boostée par Google. [23]

### 1.6.3.3 Les services utiliser avec Firebase

Firebase met à votre disposition différents services pouvant être répartis en deux catégories :

- ▶ Les outils de développement et de test de son application.
- ▶ Les outils permettant d'augmenter et d'engager ses cibles.

Voici quelques outils que les développeurs exploitent le plus dans le cadre du développement d'amplifications ou encore du test de performance des amplifications : [23]

- Realtime Database : Est une base de données hébergée dans le cloud. Les données sont stockées au format JSON et synchronisées en temps réel avec chaque client connecté. [24]
- Firebase Authentification : Fournit des services backend et des SDK faciles à utiliser pour authentifier les utilisateurs auprès de votre application. [24]
- Firebase Cloud Messaging (FCM) : Est une solution de messagerie multiplateforme qui vous permet d'envoyer des messages de manière fiable et sans frais. [24]
- Cloud Firestore : Une base de données cloud NoSQL flexible et évolutive pour stocker et synchroniser les données. Il synchronise vos données entre les applications clientes grâce à des écouteurs en temps réel et offre une assistance hors ligne. [24]
- ML kit : Est un SDK mobile qui apporte l'expertise de Google en matière d'apprentissage automatique aux applications Android et iOS dans un package puissant mais facile à utiliser. [25]
- Cloud Storage : Est conçu pour vous aider à stocker et à diffuser rapidement et facilement du contenu généré par l'utilisateur, tel que des photos et des vidéos. [24]

- **Firebase Analytics** : Est une solution gratuite de mesure des applications qui fournit des informations sur l'utilisation des applications et l'engagement des utilisateurs. [24]
- **Prédictions** : Est basé sur une technologie d'apprentissage automatique. Il apprend vos données analytiques pour développer des segments d'utilisateurs dynamiques basés sur le comportement prédit de vos utilisateurs. [26]
- **Remonte config** : Est un service cloud qui vous permet de modifier le comportement et l'apparence de votre application sans obliger les utilisateurs à télécharger une mise à jour de l'application. [24]
- **Dynamics Links** : Sont des liens qui fonctionnent comme vous le souhaitez, sur plusieurs plates-formes, et que votre application soit déjà installée ou non. Si un utilisateur ouvre un lien dynamique sur iOS ou Android, il peut être directement redirigé vers le contenu lié dans votre application native. [24]
- **Crashlytics** : Aide à collecter des analyses et des détails sur les plantages et les erreurs qui se produisent dans votre application. [24]
- **Performance monitoring** : Est un service qui vous aide à mieux comprendre les caractéristiques de performance de vos applications iOS, Android et Web. [24]
- **Test Lab** : Est une infrastructure de test d'applications basée sur le cloud qui vous permet de tester votre application sur une gamme d'appareils. [25]

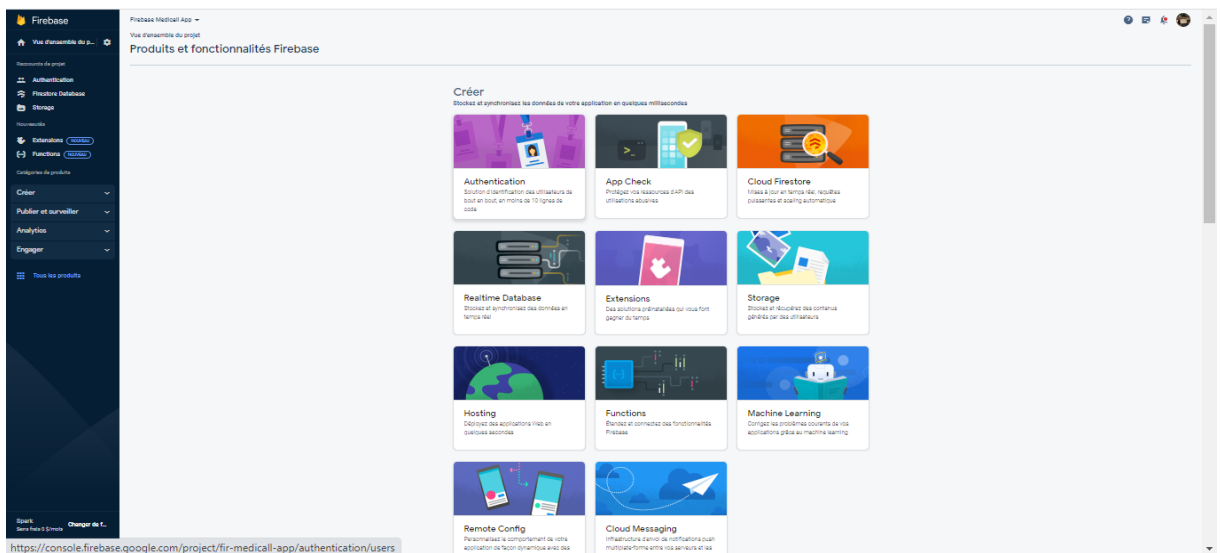


Figure I-18 Services de Firebase.



### 1.6.3.4 Les avantages d'utiliser Firebase

Firebase se démarque d'autres plateformes de développement d'application grâce notamment à ses nombreuses fonctionnalités. Voici quelques avantages à exploiter via l'exploitation de cette plateforme : [23]

- Développement rapide d'application.
- Plus besoin d'infrastructures complexes.
- Des décisions raisonnées.
- Exploiter une comptabilité multiplateforme.
- Une évolution constante et sûre.
- Un service d'assistance totalement gratuit.
- Les meilleurs produits sur une seule plateforme.
- Une plateforme soutenue par Google.

### 1.6.4 Les avantages de Flutter

- Utilisation d'une base de code unique : on code pour une plateforme, et ça fonctionne sur les autres (hors spécificité propre à la plateforme).
- Le côté réutilisable permet de réduire les coûts et temps de développement.
- Un seul langage unique, ici Dart, plutôt que de devoir coder en Objective-C / Swift et Java / Kotlin.
- Montée en compétence rapide des équipes (une seule techno à appréhender).
- Les performances avec Flutter sont similaires au natif.
- Le risque de breaking change à la sortie d'une nouvelle version du système d'exploitation est généralement limité (plutôt côté lib de fonction native, que côté widget puisque tout est dessiné par Flutter).
- Une communauté grandissante et toujours prête à aider. [21]

## 1.7 Conclusion

Il est essentiel de rester à jour avec les dernières avancées technologiques et de choisir les bons outils et Framework pour créer des expériences mobiles de qualité. Dans ce chapitre, nous avons introduit les concepts fondamentaux des applications mobiles, puis nous avons approfondi le sujet en présentant le Framework Flutter. Le chapitre suivant sera consacré à la présentation de notre projet et à l'étude préliminaire.

# Chapitre II Conception

## 2.1 Introduction

Les projets informatiques nécessitent une étude préalable constituée de plusieurs étapes. Cette étude commence par une analyse, suivie par la spécification des besoins, où il faut définir les acteurs et les exigences du système. Ensuite, il convient d'entamer l'étape de la conception afin de rendre le projet final conforme aux besoins du client. Cette étude mène à la réalisation et la mise en œuvre du projet.

À cet effet, nous présentons dans ce chapitre notre projet à réaliser. Ensuite, nous entamerons la conception de notre application en utilisant UML comme langage de modélisation. Ce langage nous permettra d'obtenir des représentations graphiques et de modéliser les applications à l'aide de diagrammes.

## 2.2 Présentation de l'application

Le besoin de la médecine à domicile est devenu de plus en plus important dans notre société. Plusieurs personnes, telles que les personnes âgées, les personnes atteintes de maladies chroniques ou à mobilité réduite, ont besoin de recevoir des soins médicaux ou des consultations sans être obligées de se déplacer.

Par conséquent, la médecine à domicile permet de réduire les déplacements du patient et de ses proches, de lui offrir des soins personnalisés et adaptés à ses besoins, tout en favorisant le maintien d'une certaine autonomie. Les consultations et les soins à domicile peuvent inclure plusieurs services, tels que les visites régulières de professionnels de santé, la surveillance de l'état de santé, les soins infirmiers, la prise de sang...

Dans le cadre de notre projet de fin d'étude, nous proposons de développer une application mobile qui vise à faciliter la consultation et les soins à domicile. Cette application permettra aux utilisateurs de trouver en quelques clics des professionnels de santé qualifiés (tels que des médecins, des infirmiers, des aides-soignants, des laboratoires d'analyse, et des rééducateur) qui peuvent se déplacer à domicile. Elle permettra également de réserver une ambulance équipée pour le transport des patients.

La recherche des professionnels de santé peut être effectuée selon le type et la spécialité.

L'application offre la possibilité de la planification des rendez-vous selon la date, l'heure et la localisation. En conclusion, notre application est une solution innovante et intéressante pour améliorer la qualité des soins à domicile.

## 2.3 Planification du projet

La planification de projet est la structuration logique des tâches et des fonctionnalités à réaliser. Pour présenter une vue globale du processus de réalisation de notre application, nous exposons le diagramme de Gantt ci-dessous.

**Diagramme de Gantt :**

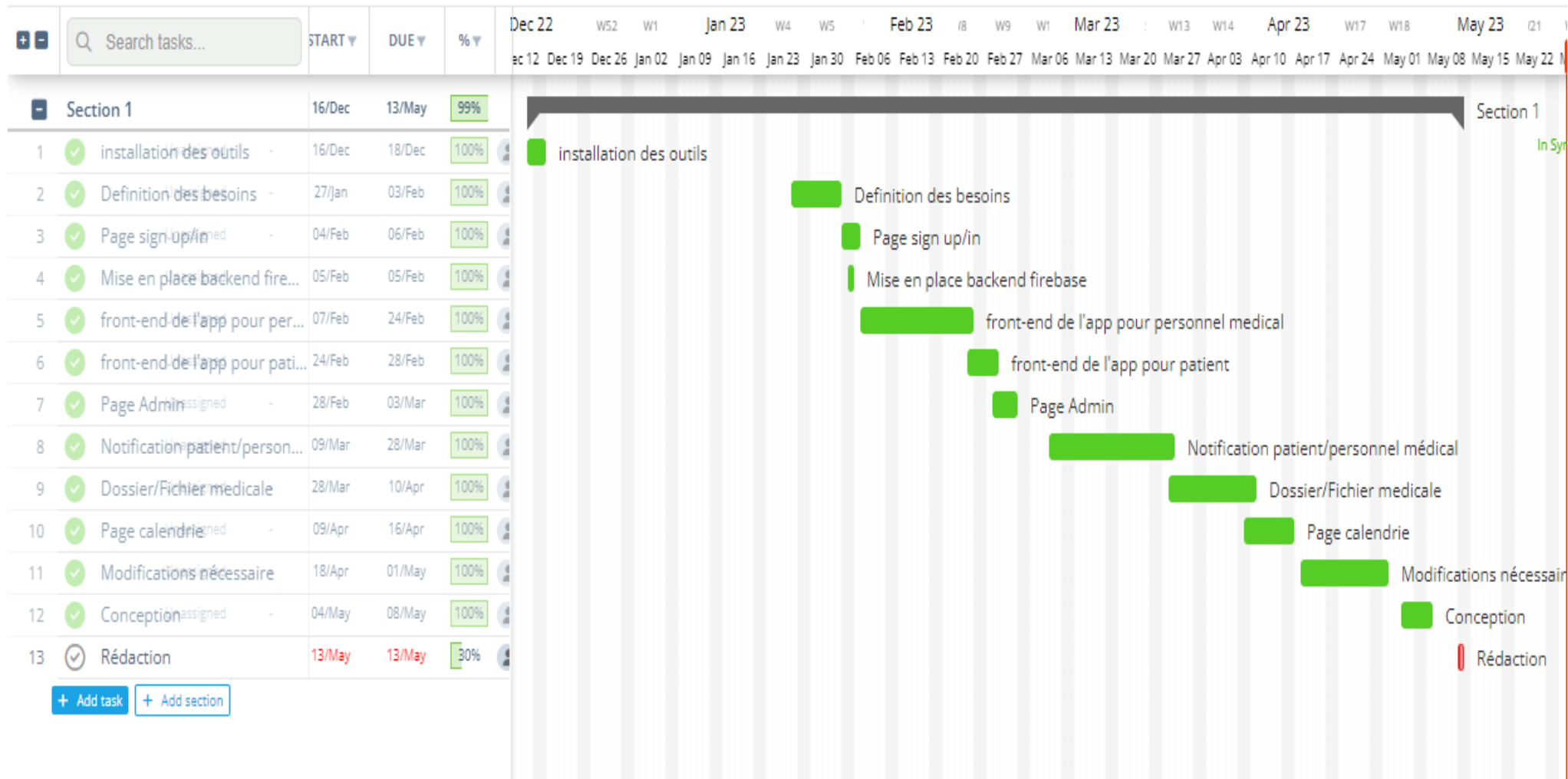


Figure II-1 Diagramme de Gantt.

## 2.4 Etude de l'existant

### - Modèles existants :

- ✓ **eSiha** : Une application de télémédecine mobile algérienne qui permet aux patients de trouver et géolocaliser les médecins, les cliniques et les centres médicaux. Elle permet aussi d'utiliser les services de télémédecine, de gérer les dossiers médicaux, et les initiatives de santé de toute la famille dans un seul compte.



**Figure II-2** eSiha. [42]

- ✓ **Beesiha** : C'est une application mobile médicale spécialement conçue pour l'Algérie. Elle offre aux utilisateurs la possibilité de programmer une consultation en ligne avec le docteur de son choix et retrouvez leur historique de rendez-vous, leurs résultats, leurs ordonnances.



**Figure II-3** Beesiha.[41]

- ✓ **eTabib** : Est une application de télémédecine conçue pour permettre aux utilisateurs de consulter un médecin en ligne via leurs appareils mobiles. L'utilisateur peut choisir le médecin qu'il souhaite consulter en fonction de son expertise, prendre rendez-vous et faire le paiement en ligne. Les médecins peuvent également se connecter à l'application pour gérer leurs rendez-vous et communiquer avec leurs patients en ligne. Cette application est disponible en Algérie, elle est en française.

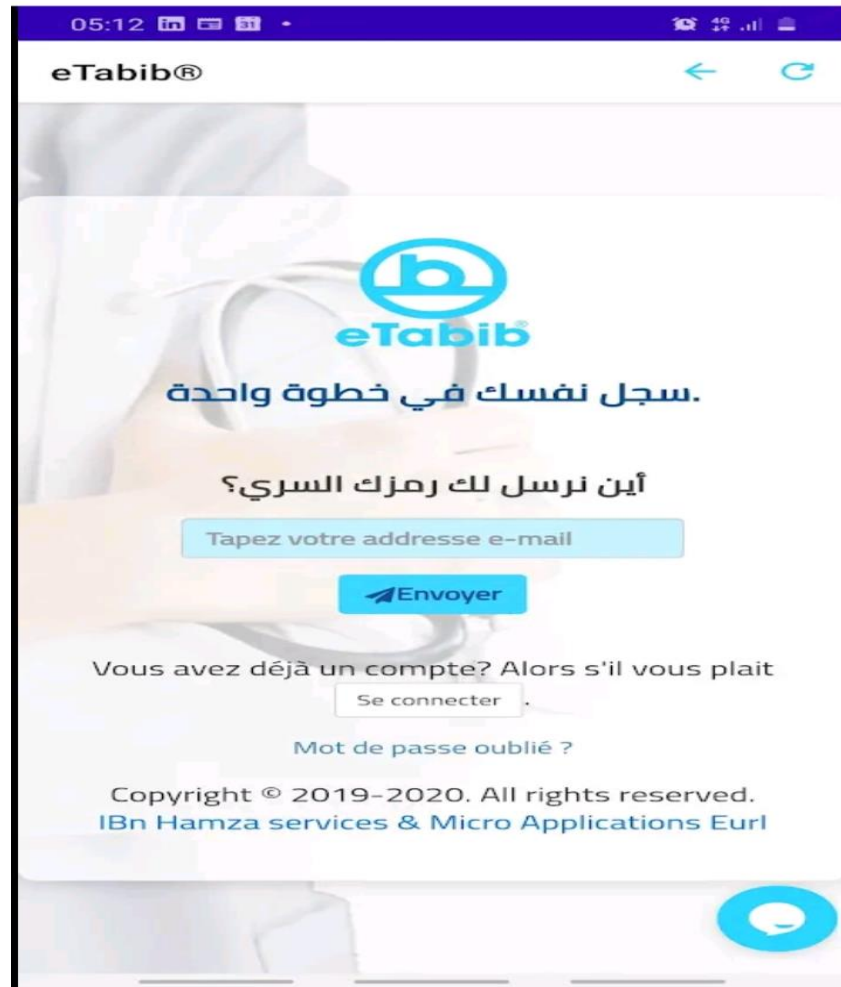


Figure II-4 eTabib. [43]

## 2.5 Identification des acteurs

L'ensemble des acteurs de notre système est présenté comme suit :

- **Patient** : Après avoir créé un compte et vérifié son identité, il pourra accéder aux fonctionnalités de recherche de professionnels de santé, de prendre un rendez-vous et voir ses dossiers médicaux.
- **Professionnels de santé** : Est l'acteur qui gère les rendez-vous avec les patients et crée les dossiers médicaux des patients et ajouter des notes et des informations supplémentaires, et pour pouvoir faire tout ça, il faut créer un compte et s'authentifier pour y accéder
- **Admin** : C'est le responsable de la gestion du personnel médical et des spécialités disponibles dans l'application. Il peut s'authentifier pour accéder à son l'interface. Il peut ajouter les spécialités, les comptes de personnels médicaux et les supprimer en fonction des besoins.



## 2.6 Planification des besoins

Il existe deux types de besoins : les besoins fonctionnels et les besoins non fonctionnels.

### 2.6.1 Besoin fonctionnel

Un besoin est un ensemble d'opérations logiques qu'un programme ou une partie de celui-ci doit effectuer pour répondre aux attentes des acteurs.

**Tableau II-1** Tableau de spécification des besoins.

Acteur	Besoin
Patient	B1 : Créer compte. B2 : S'authentifier. B3 : Rechercher un personnel médical. B4 : Prendre un rendez-vous. B5 : Consulter le dossier médical. B6 : Télécharger un fichier médical. B7 : Modifier profil. B8 : Supprimer compte.
Personnel médical	B1 : Créer compte. B2 : S'authentifier. B3 : Accepter un rendez-vous. B4 : Refuser un rendez-vous. B5 : Créer dossier médical. B6 : Exporter un fichier. B7 : Modifier profil. B8 : Supprimer compte.
Admin	B1 : S'authentifier. B2 : Accepter un personnel médical. B3 : Supprimer un personnel médical. B4 : Ajouter une spécialité.

### 2.6.2 Besoin non-fonctionnel

Outre les besoins fondamentaux, l'application doit répondre aux besoins suivants qui la caractérisent :

- Performance : Avant tout, l'application doit être performante. En d'autres termes, ses fonctionnalités doivent répondre de manière optimale à toutes les exigences des utilisateurs.
- Conviviale : L'application doit être facile à utiliser. En effet, l'interface d'utilisateur doit être conviviale (simple), ergonomique et adaptée à l'utilisateur.

- La rapidité de traitement : Etant donné le grand nombre de transactions quotidiennes, il est vraiment important que le temps d'exécution du traitement soit aussi proche que possible du temps réel.
- Sécurité : Cette caractéristique est essentielle car l'application gère des données sensibles et l'authentification est essentielle.
- Disponibilité : Il est indispensable que l'application soit toujours en service (24/24, 7/7).

## 2.7 UML

### 2.7.1 Définition

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie, etc. [27]

## 2.8 Les diagrammes du système

### 2.8.1 Diagramme de cas d'utilisation

Il représente une fonctionnalité spécifique dans un système et est créé pour illustrer comment les différentes fonctionnalités sont interconnectées et montrer leurs contrôleurs (ou acteurs) internes et externes. [27]

### 2.8.1.1 Diagramme de cas d'utilisation général

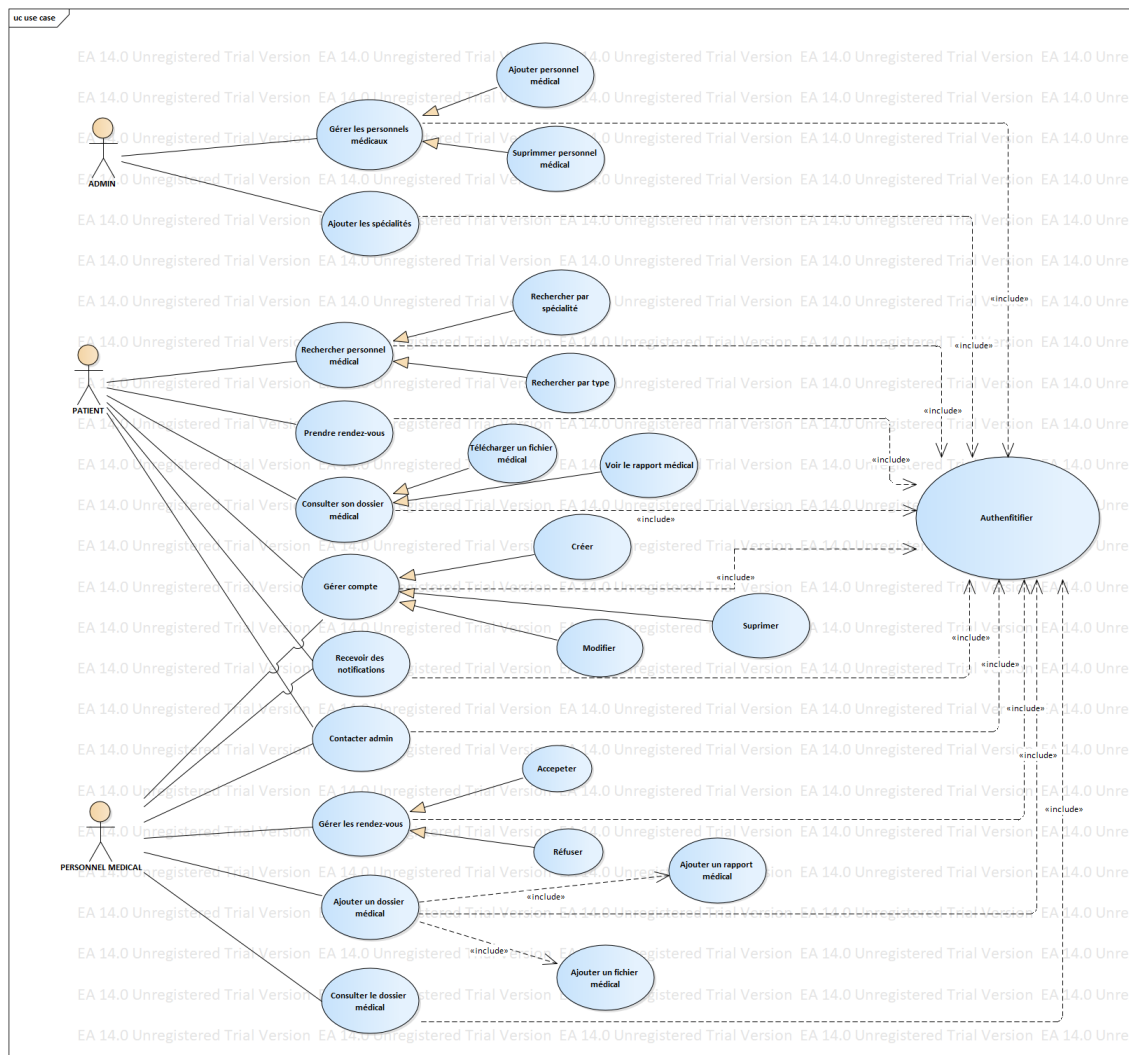


Figure II-5 Diagramme de cas d'utilisation.

## 2.8.2 Description textuelle des cas d'utilisation

### 2.8.2.1 Cas d'utilisation « Authentifier »

Ce tableau montre et affiche le cas d'utilisation de "Authentifier" et les différents acteurs peuvent accéder à ce service.

Tableau II-2 Authentifier.

**Identification :**

- **Nom du cas d'utilisation :** Authentifier.
- **But :** Faire accès à l'application.
- **Acteur :** Patient, Personne médicale, Admin.

**Séquencement :**

- **Précondition :** Créer un compte.
- **Enchainements nominaux :**
  - L'utilisateur saisie son email et son mot de passe.
- **Enchainements alternatifs :**
  - Email non valide.
  - Mot de passe non valide.
- **Post-conditions :**
  - L'utilisateur accède à l'application.

**2.8.2.2 Cas d'utilisation « Consulter dossier médical »**

Ce tableau montre et affiche le cas d'utilisation de " Consulter dossier médical " et les différents acteurs qui peuvent accéder à ce service.

**Tableau II-3** Consulter dossier médical.

**Identification :**

- **Nom du cas d'utilisation :** Consulter dossier médical.
- **But :** Afficher et vérifier l'état de santé.
- **Acteur :** Patient.

**Séquencement :**

- **Précondition :** S'authentifier.
- **Enchainements nominaux :**
  - L'utilisateur accéder à la page documents médical.
- **Enchainements alternatifs :**
  - Aucune.
- **Post-conditions :**
  - L'utilisateur accède tous les dossiers médicaux.
  - L'utilisateur consulte tous les dossiers médicaux.

- L'utilisateur importe les fichiers de consultation.

### 2.8.2.3 Cas d'utilisation « Ajouter personne médical »

Ce tableau montre et affiche le cas d'utilisation de " Consulter dossier médical " et les différents acteurs qui peuvent accéder à ce service.

**Tableau II-4** Ajouter personne médical.

<p><b><u>Identification :</u></b></p> <ul style="list-style-type: none"> <li>• <b>Nom du cas d'utilisation :</b> Ajouter personne médical.</li> <li>• <b>But :</b> Rapprocher un personnel médical au patient.</li> <li>• <b>Acteur :</b> Admin.</li> </ul>
<p><b><u>Séquencement :</u></b></p> <ul style="list-style-type: none"> <li>• <b>Précondition :</b> S'authentifier.</li> <li>• <b>Enchainements nominaux :</b> <ul style="list-style-type: none"> <li>- L'utilisateur accéder à la page des personnes médical.</li> <li>- L'utilisateur consulte les informations de ces personnes médicales.</li> <li>- L'utilisateur accepte la personne médicale.</li> </ul> </li> <li>• <b>Enchainements alternatifs :</b> <ul style="list-style-type: none"> <li>- La personne médicale existe déjà.</li> </ul> </li> <li>• <b>Post-conditions :</b> <ul style="list-style-type: none"> <li>- Mise à jour de la base de données.</li> </ul> </li> </ul>

### 2.8.2.4 Cas d'utilisation « Ajouter un dossier médical »

Ce tableau montre et affiche le cas d'utilisation de " Ajouter un dossier médical " et les différents acteurs qui peuvent accéder à ce service.

Tableau II-5 Ajouter un dossier médical.

<p><b><u>Identification :</u></b></p> <ul style="list-style-type: none"> <li>• <b>Nom du cas d'utilisation :</b> Ajouter dossier médical.</li> <li>• <b>But :</b> Avoir une bonne prise en charge médicale, fournir un bon suivi de santé et assurer une gestion médicale utile.</li> <li>• <b>Acteur :</b> Personne médical.</li> </ul>
<p><b><u>Séquencement :</u></b></p> <ul style="list-style-type: none"> <li>• <b>Précondition :</b> S'authentifier.</li> <li>• <b>Enchainements nominaux :</b> <ul style="list-style-type: none"> <li>- L'utilisateur accède à la page 'event'.</li> <li>- L'utilisateur accède à la page de consultation.</li> <li>- L'utilisateur saisie les informations et le document nécessaires.</li> <li>- L'utilisateur enregistre le document dans la base de données.</li> </ul> </li> <li>• <b>Enchainements alternatifs :</b> <ul style="list-style-type: none"> <li>- L'oublie de remplir un champ.</li> </ul> </li> <li>• <b>Post-conditions :</b> <ul style="list-style-type: none"> <li>- Mise à jour de la base de données.</li> </ul> </li> </ul>

### 2.8.3 Diagramme de séquence

Montre comment les objets interagissent les uns avec les autres et dans quel ordre. Ils représentent les interactions d'un scénario particulier. [27]

#### 2.8.3.1 Diagramme de séquence « Authentification »

L'authentification consiste à assurer la confidentialité des données préexistantes dans la base de données. Nous gérons, dans ce diagramme, la séquence d'authentification au système lors de la vérification de l'e-mail et du mot de passe. Deux scénarios peuvent se produire lors de l'authentification de l'utilisateur : si les données sont correctes, le système affiche un message de réussite et permet à l'utilisateur d'accéder à l'interface appropriée. Par contre, il affiche un message d'erreur si les données sont incorrectes.

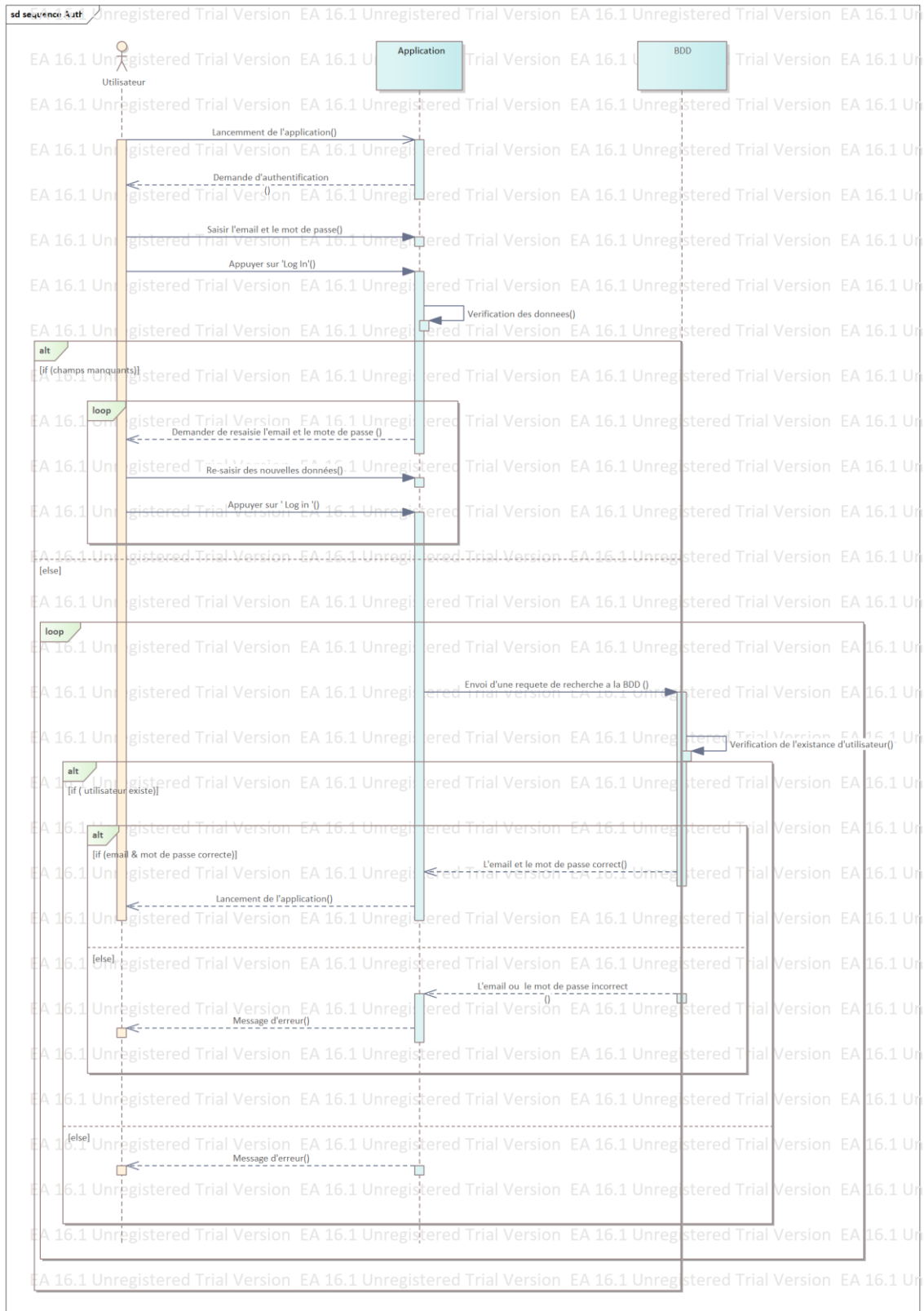


Figure II-6 Diagramme de séquence « Authentification ».

### 2.8.3.2 Diagramme de séquence « Modifier compte »

Dans ce diagramme on traite la séquence « Modifier compte ». L'utilisateur va pouvoir modifier ses informations par remplir un formulaire, puis il confirme la modification en saisissant son mot de passe actuel. En cas d'erreur dans le formulaire de modification ou le mot de passe, le système affiche un message au lieu d'erreur.



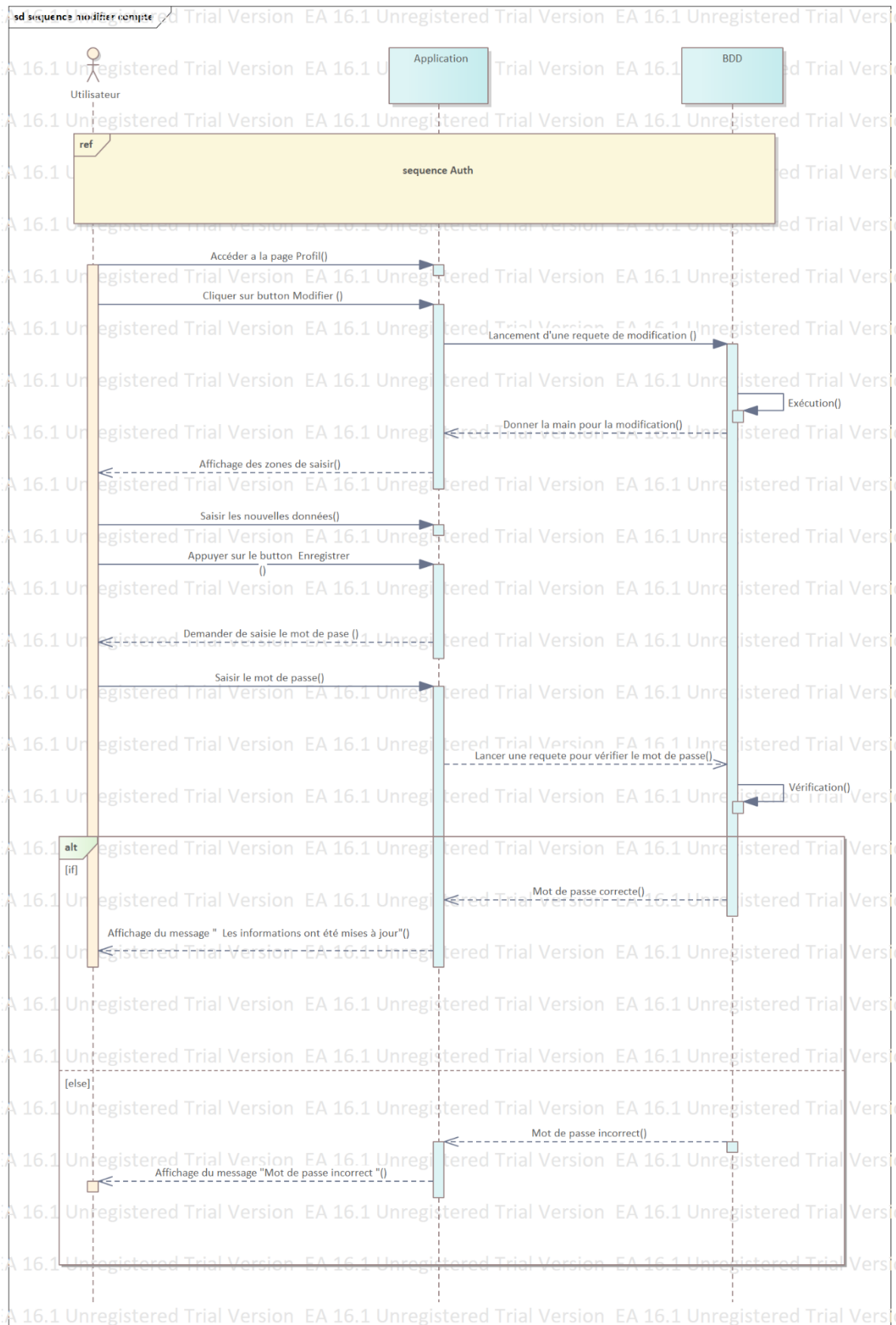


Figure II-7 Diagramme de séquence « Modifier compte ».

### 2.8.3.3 Diagramme de séquence « Rechercher personnel médical »

Dans ce diagramme, nous avons une séquence « Rechercher personne médical », où l'utilisateur peut rechercher des professionnels de santé en fonction de leur type (Médecin, Infirmier, etc.) ou de leur spécialité (Pédiatre, généraliste, etc.). Après avoir saisi les critères de recherche, une liste de résultats sera affichée, avec les informations pertinentes pour chaque professionnel. Si une erreur se produit lors de la saisie des critères de recherche, une page vide sera affichée, sinon les résultats s'afficheront avec succès.

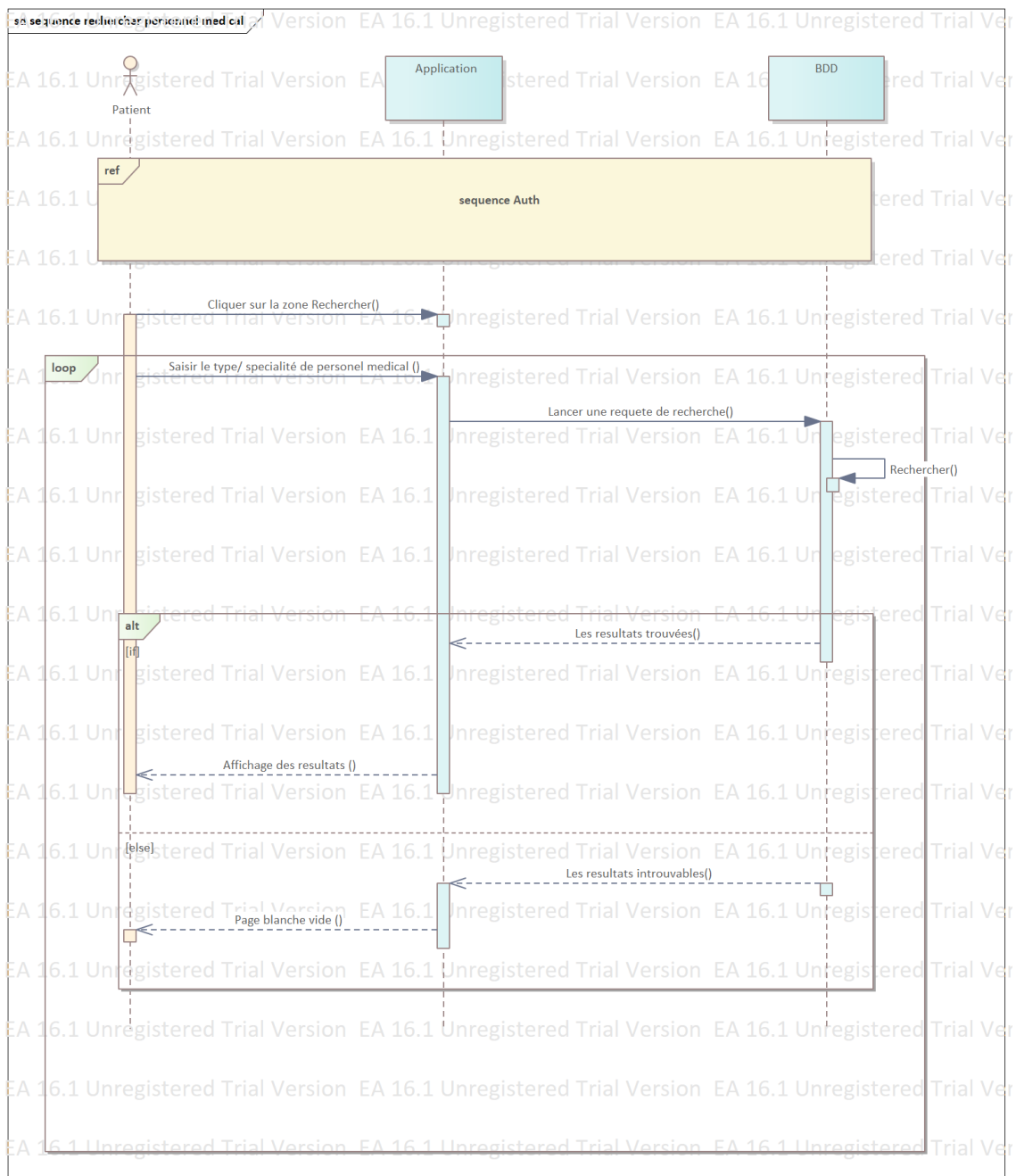


Figure II-8 Diagramme de séquence « Rechercher personnel médical ».

### 2.8.3.4 Diagramme de séquence « Prendre rendez-vous »

Dans ce diagramme, nous allons présenter une séquence « Prendre rendez-vous », où l'utilisateur peut prendre un rendez-vous via l'application en sélectionnant l'heure, la date et le lieu. Ensuite, il envoie le formulaire en cliquant sur « Enregistrer », et les données seront envoyées à la base de données.

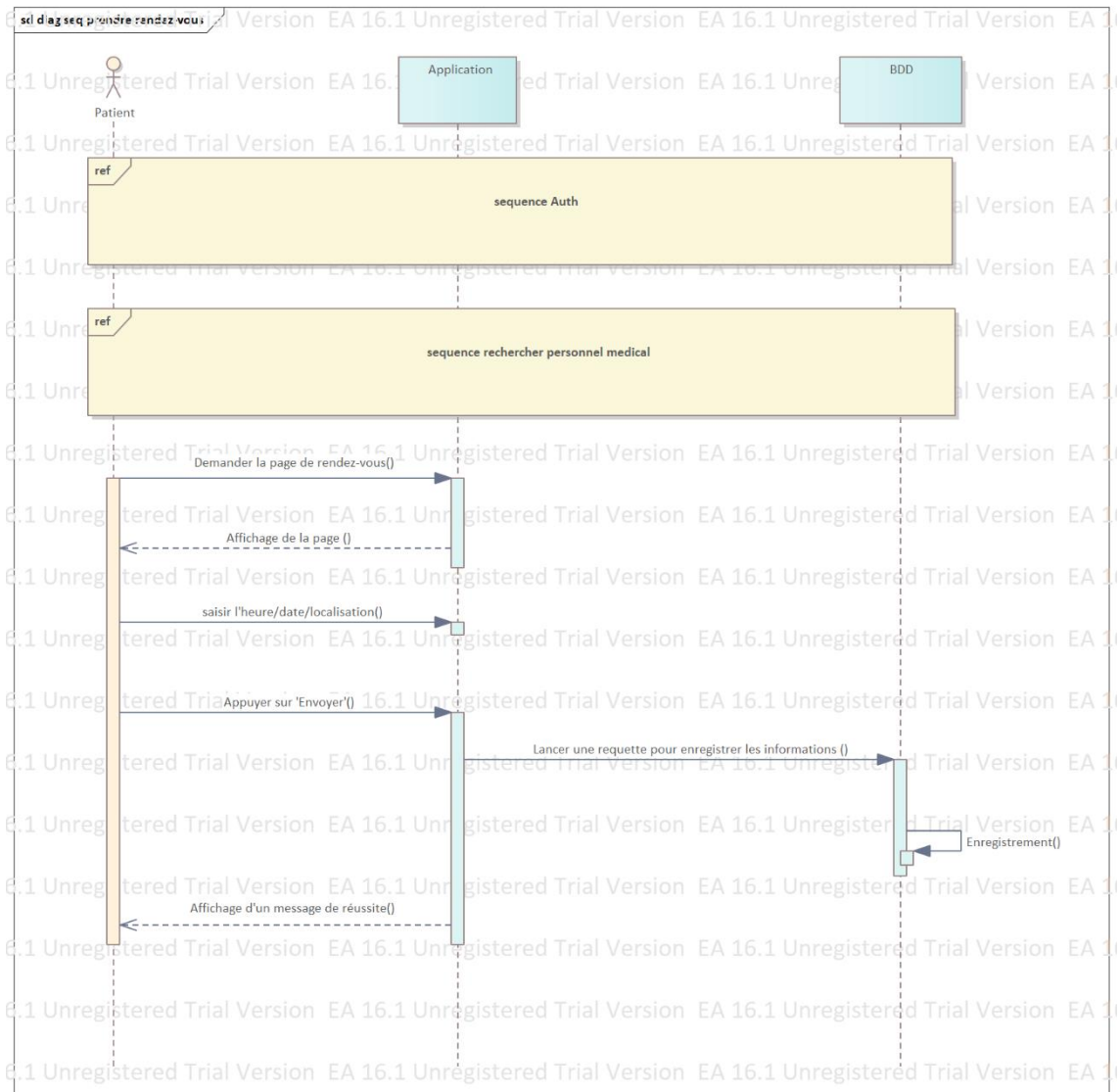


Figure II-9 Diagramme de séquence « Prendre rendez-vous ».

### 2.8.4 Diagramme d'activité

Flux de travail d'entreprise ou opérationnels représentés graphiquement pour montrer l'activité de chacune des composantes du système. [27]

### 2.8.4.1 Diagramme d'activité « Prendre un rendez-vous »

Après avoir recherché un professionnel de santé, le patient peut prendre un rendez-vous en remplissant un formulaire en indiquant la localisation, l'heure et la date de la consultation souhaitée. Une fois que le formulaire est rempli, le patient envoie sa demande de rendez-vous.

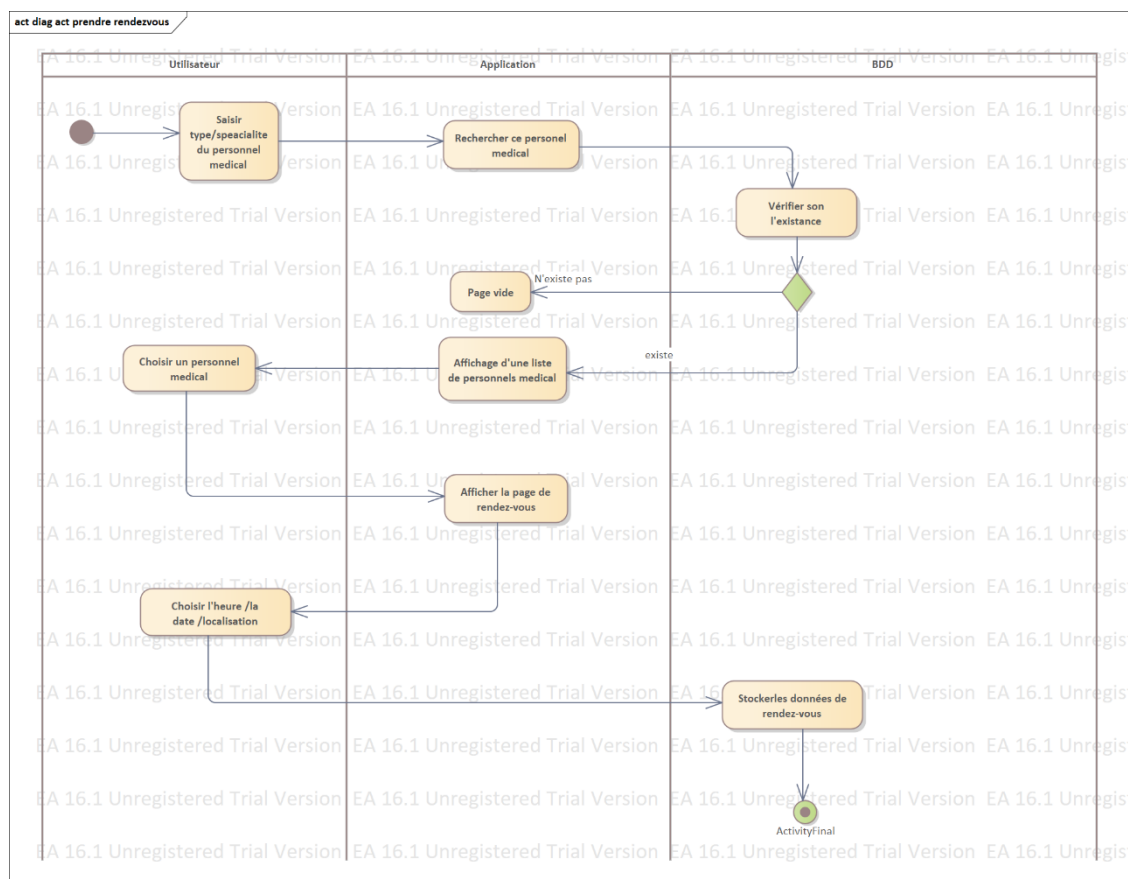
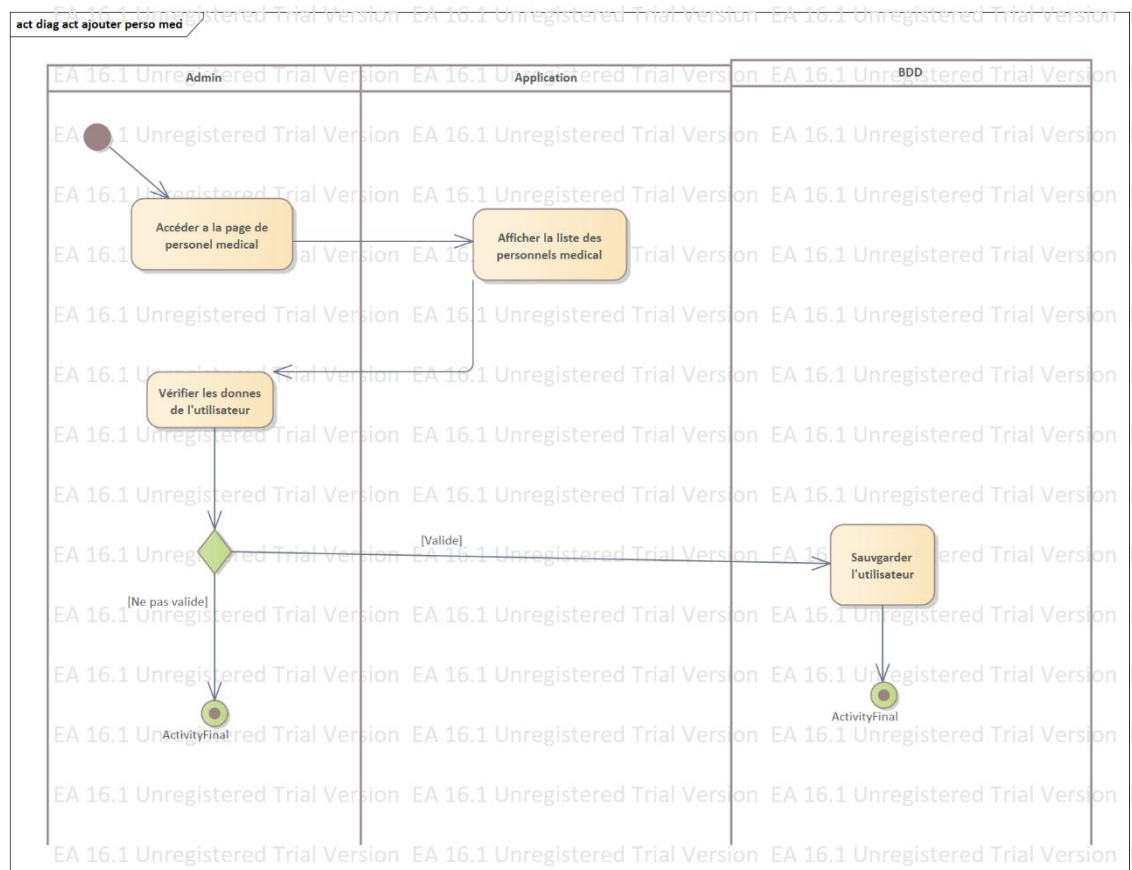


Figure II-10 Diagramme d'activité « Prendre un rendez-vous ».

### 2.8.4.2 Diagramme d'activité « Ajouter un personnel médical »

L'administrateur est chargé de la gestion du personnel médical. Il peut ajouter de nouveaux membres à l'équipe. Une fois que le personnel médical a créé un compte, l'administrateur vérifie les données, et il a le choix d'accepter ou refuser l'ajout.



**Figure II-11** Diagramme d'activité « Ajouter un personnel médical ».

### 2.8.4.3 Diagramme d'activité « Créer compte »

Un compte utilisateur est nécessaire pour accéder à l'application en tant que patient ou personnel médical. Pour créer un compte, les utilisateurs doivent remplir le formulaire avec les informations personnelles et de contact. Après avoir saisi toutes les informations, l'utilisateur peut enregistrer son compte. Si toutes les informations sont valides, le système enregistrera le compte utilisateur et lui permettra d'accéder à l'application. Si une erreur est détectée lors de la saisie, le système affichera un message d'erreur et invitera

l'utilisateur à corriger les informations incorrectes avant d'enregistrer un compte.

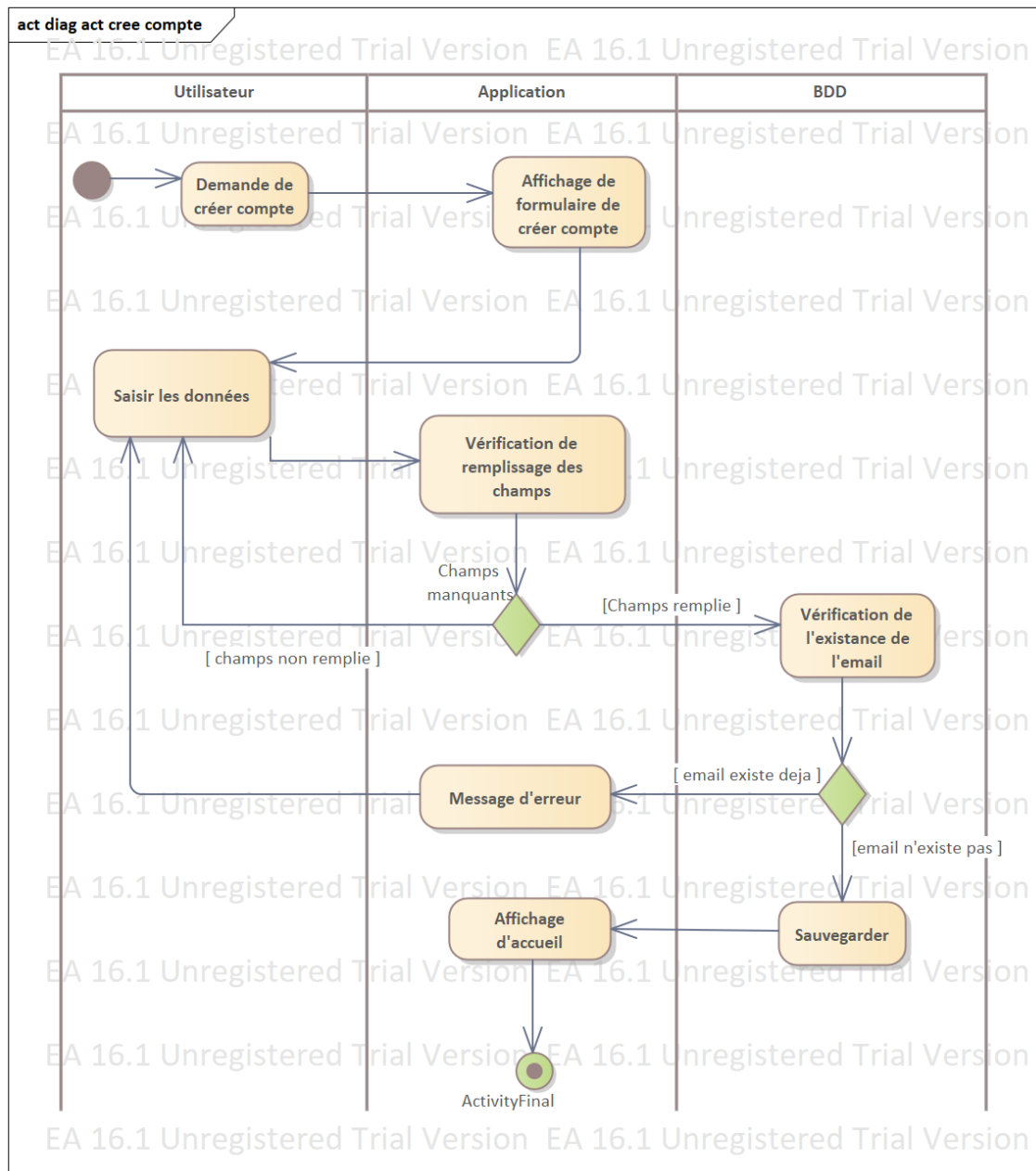


Figure II-12 Diagramme d'activité « Créer compte ».

## 2.9 Conclusion

Ce chapitre a donné une vision de notre application à réaliser. Nous avons commencé par une brève description de notre application. Ensuite nous avons développé une étude conceptuelle à l'aide de plusieurs diagrammes de UML. Dans le prochain chapitre, nous entamerons par la phase de mise en œuvre.

# Chapitre III Réalisation

## 3.1 Introduction

Une fois la conception est achevée, nous passons à l'étape de l'implémentation et de la réalisation de notre application. Dans ce chapitre, nous présentons l'application finale, en mettant en avant les fonctionnalités proposées, les interfaces et quelques scénarios d'utilisation.

## 3.2 Environnement du travail

Ci-dessous nous présentons l'ensemble des logiciels et matériels utilisés pour la réalisation de notre application.

### 3.2.1 Environnement logiciel

#### ❖ **Android studio :**

Android Studio est l'environnement de développement intégré (IDE) officiel des applications Android. Basé sur le puissant outil de développement et d'édition de code d'IntelliJ IDEA, Android Studio offre encore plus de fonctionnalités qui améliorent la productivité lors de la création d'applications Android. [28]



**Figure III-1** Logo de Android studio. [29]

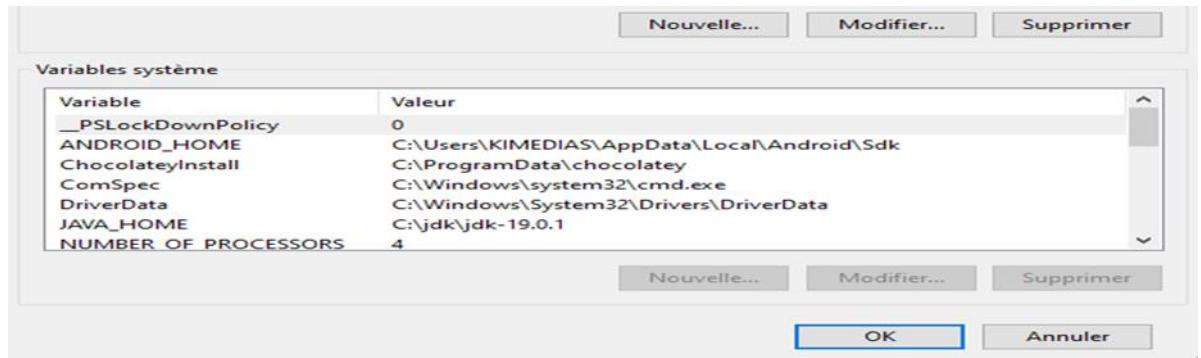
Afin de réaliser notre application, nous avons téléchargé et installé Android Studio à partir du site officiel d'Android Studio. Cette installation inclut également l'installation du JDK, qui est nécessaire pour exécuter les outils de développement. En outre, nous avons installé les packages suivants :

- **SDK Android :**

Le SDK, c'est-à-dire un Kit de développement logiciel, est un ensemble d'outils que met à disposition un éditeur afin de permettre de développer des



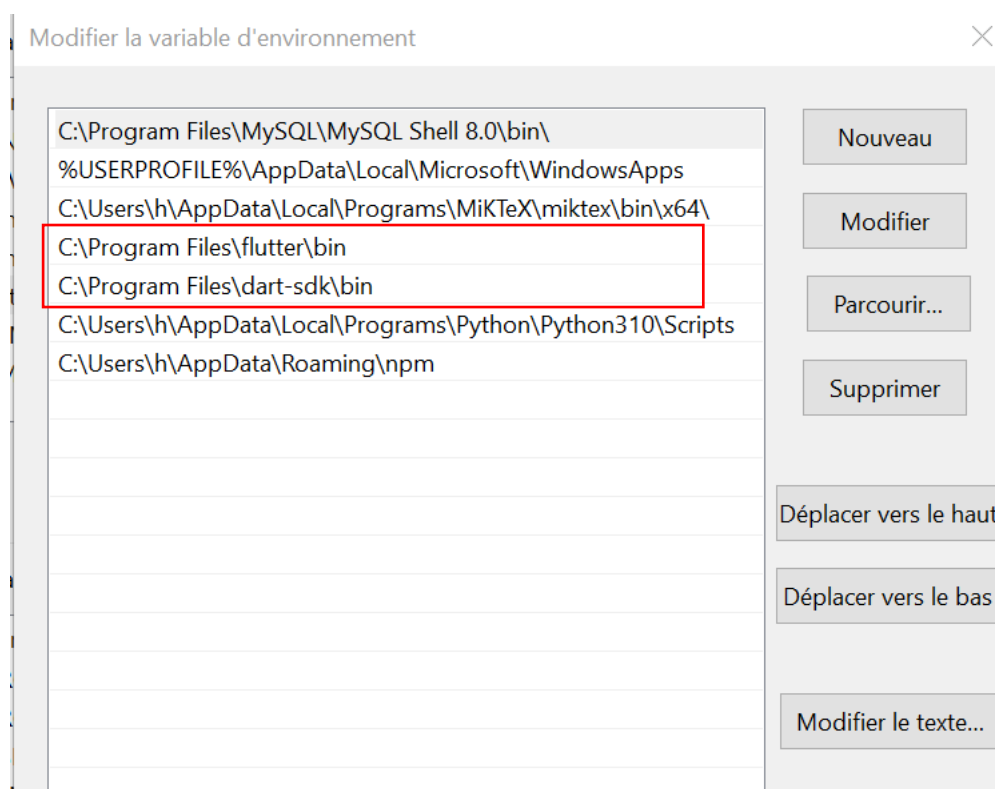
applications pour un environnement précis. Le SDK Android permet, donc, de développer des applications pour android et uniquement pour Android. [30]  
Tous d'abord nous avons installé le SDK et l'ajouter l'emplacement du SDK Android a la variable d'environnement ANDROID\_HOME et PATH.



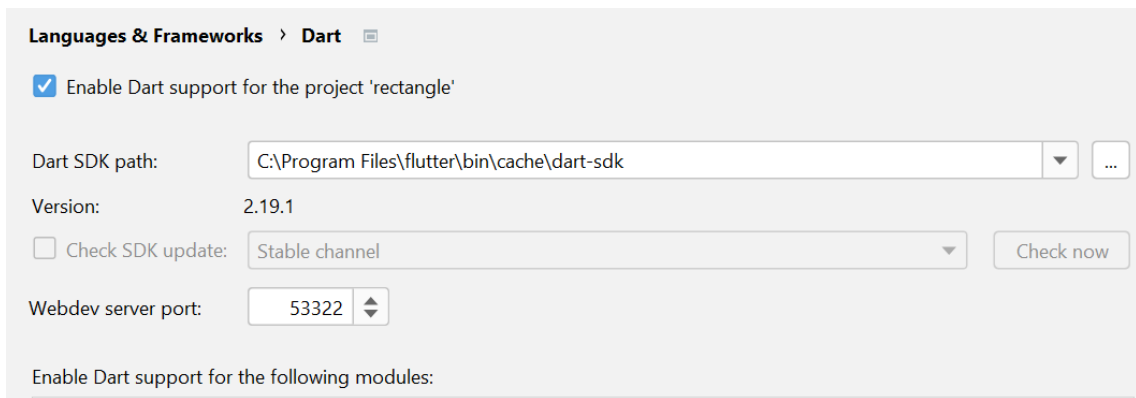
**Figure III-2** Emplacement du SDK.

- **SDK de Flutter et Dart :**

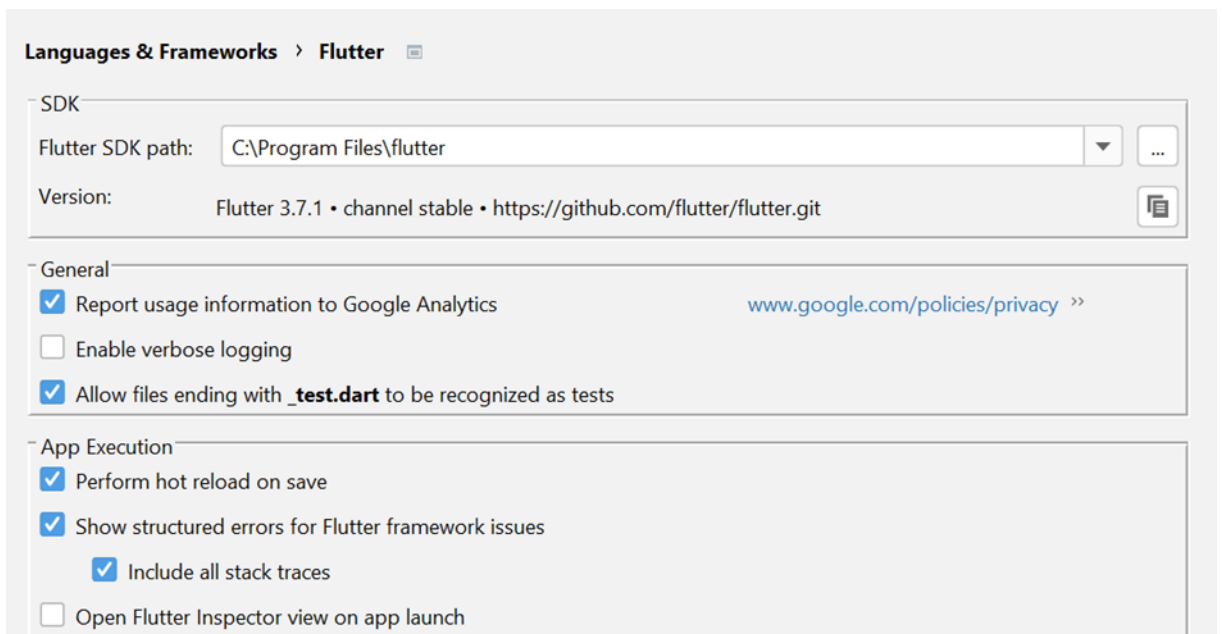
Nous avons installé le package de Flutter et Dart sur notre système et configuré les variables d'environnement pour y accéder depuis Android Studio. Ensuite, nous avons installé les extensions Flutter et Dart sur Android Studio, ce qui permet d'utiliser les commandes Flutter dans le terminal intégré d'Android Studio.



**Figure III-3** Emplacement de flutter&Dart.



**Figure III-4** Intégration SDK Dart sur android studio.



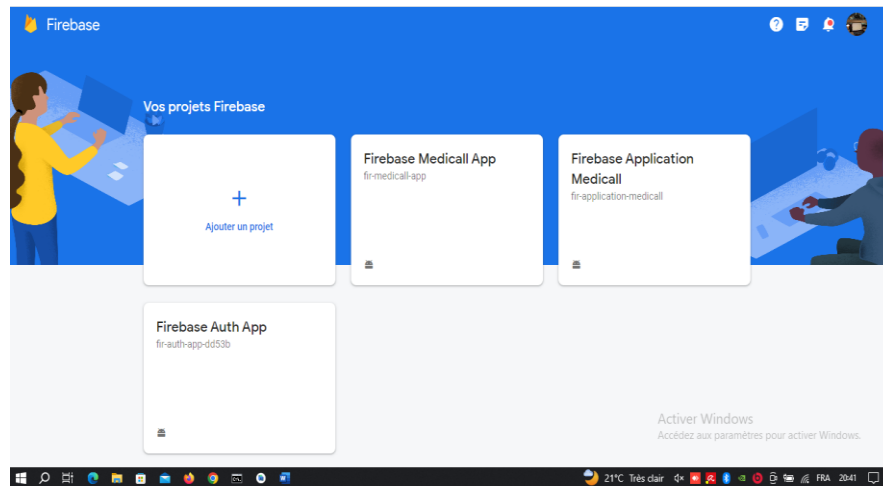
**Figure III-5** Intégration SDK flutter sur android studio.

#### ❖ **Firebase :**

Nous avons choisi d'utiliser Firebase en raison de sa convivialité pour notre projet, de sa capacité à gérer les mises à jour en temps réel et de sa compatibilité avec nos exigences de modélisation de données que nous avons.

#### **Etape 1 :**

Premièrement nous avons créé un projet Firebase sur le site officiel de Firebase

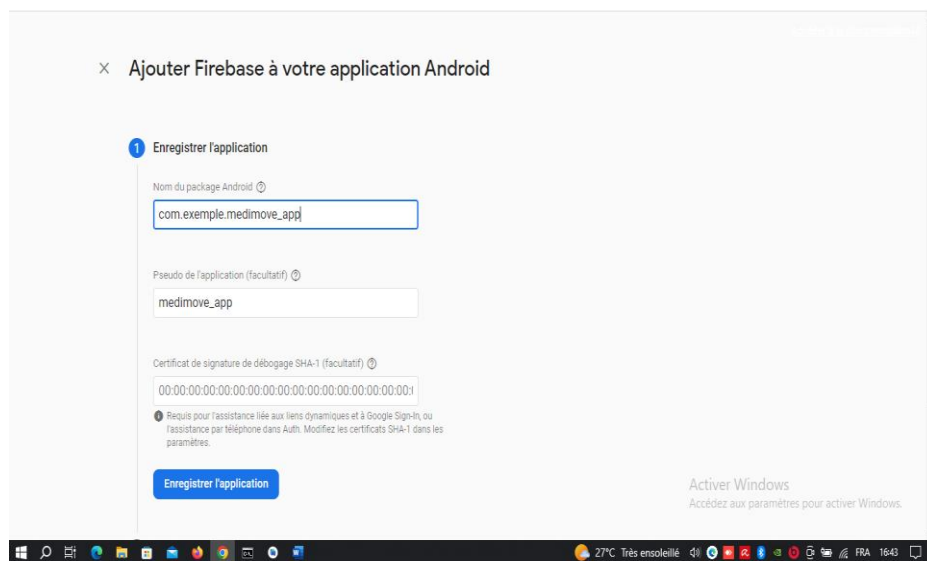


**Figure III-6** Ajouter projet dans Firebase.

En cliquant sur "Ajouter un projet" puis on donne un nom à notre projet, après nous suivons les étapes pour créer un nouveau projet Firebase.

### Etape 2 :

Une fois le projet est créé, nous devons enregistrer notre application sur la plateforme de Firebase en remplissant les champs requis (voire l'image ci-dessous).



**Figure III-7** Les donne de l'application sur Firebase.

### Etape 3 :

Ensuite, nous devons télécharger le fichier de configuration JSON qui contient les services de Google, et le placer dans le répertoire racine de notre projet Flutter (dans le répertoire android/app).

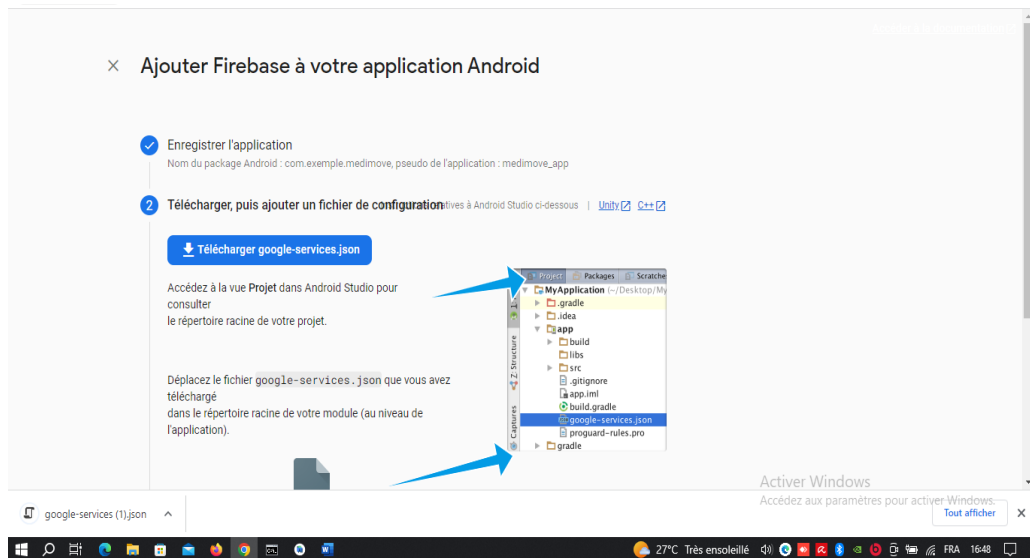


Figure III-8 Téléchargement de JSON.

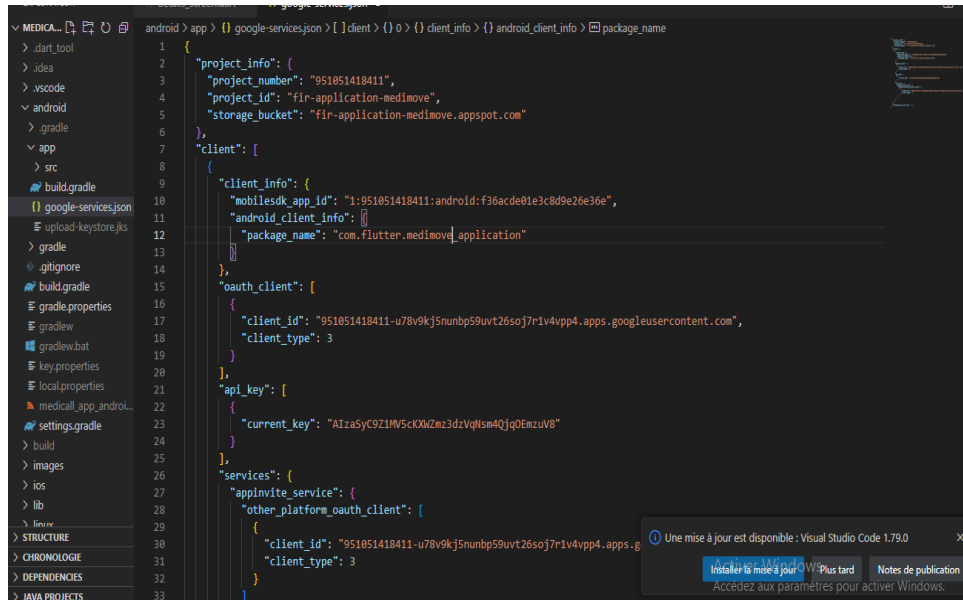


Figure III-9 Fichier JSON.

### Initialisation de Firebase :

Nous avons initialisé Firebase dans notre application Flutter en utilisant le package "firebase\_core". Cela se fait généralement au niveau de fichier main.dart, dans la méthode "main()". L'initialisation se fait en utilisant la méthode "Firebase.initializeApp()".

```

1 import ...
10
11 void main() async {
12   WidgetsFlutterBinding.ensureInitialized();
13   await Firebase.initializeApp();
14   runApp(const MyApp());
15 }
16
17 class MyApp extends StatelessWidget {
18   const MyApp({super.key});
19
20   // This widget is the root of your application.
21   @override
22   Widget build(BuildContext context) {
23     return MaterialApp(
24       debugShowCheckedModeBanner: false,
25       title: 'Flutter Demo',
26       theme: ThemeData(
27         primarySwatch: Colors.blue,
28       ),
29       //home: const NavBarScreen(),
30       // home: const Auth(),
31       home: SplashScreen(),
32       routes: {
33         // 'SignupPatientScreen': (context) => const SignupPatientS

```

Figure III-10 Initialisation Firebase.

### ❖ Package flutter et Dart :

Intégration des packages Flutter : nous avons ajouté les packages Flutter requis dans notre fichier pubspec.yaml. Ce fichier indique les dépendances de notre projet, et en exécutant la commande "flutter pub get", les packages sont téléchargés et ajoutés à notre projet.

```

! pubspec.yaml
35
36 # The following adds the Cupertino Icons font to your application.
37 # Use with the CupertinoIcons class for iOS style icons.
38 cupertino_icons: ^1.0.2
39 curved_labeled_navigation_bar: ^2.0.2
40 google_fonts:
41   firebase_core: ^2.6.1
42   firebase_auth: ^4.2.8
43   cloud_firestore: ^4.4.2
44   image_picker: ^0.8.6+2
45   image_cropper: ^3.0.1
46   firebase_storage: ^11.0.14
47   intl: ^0.17.0
48   location: ^4.4.0
49   firebase_service: ^0.0.1+1
50   geocoding: ^2.0.5
51   flutter_badged: ^2.0.3
52   firebase_messaging: ^14.2.6
53   flutter_local_notifications: ^13.0.0
54   notification_permissions: ^0.6.1
55   getwidget: ^3.0.1
56   syncfusion_flutter_calendar: ^21.1.38
57   file_picker: ^5.2.11
58   awesome_gallery_saver: ^0.0.6
59   permission_handler: ^10.2.0
60   path_provider: ^2.0.14
61   shared_preferences: ^2.1.0
62   flutter_file_downloader: ^1.1.2
63   http: ^0.13.6
64   rename: ^2.1.1
65 dev_dependencies:
66   flutter_test:
67     sdk: flutter

```

Figure III-11 Les packages.

### 3.2.2 Environnement matériel

Afin d'installer et d'exécuter Flutter, l'environnement de développement doit répondre à des exigences minimales.

Pour la réalisation de l'application, nous avons déployé :

- Un pc portable HP pour le développement ayant les caractéristiques suivantes :
  - Intel Core i5.
  - 6 Go de mémoire vive.
  - Windows 10 Professionnel - 64 bits.
- Un Smartphone OPPO A15 pour installer et exécuter l'application et réaliser les tests ayant les caractéristiques suivantes :
  - Version Android : 10.
  - RAM : 8 Go.
  - Mémoire interne : 128 Go.

### 3.3 Présentation des interfaces

Cette section met en avant les interfaces de l'application, conçues selon des principes ergonomiques visant à rendre son utilisation facile pour tous les utilisateurs.

#### 3.3.1 Interfaces Hommes-Machines

✓ **Logo de l'application :**

Le choix du nom "MediMove" pour notre application mobile médicale exprime l'idée de la demande des professionnels de santé à domicile pour fournir des consultations ou des soins médicaux.



Figure III-12 Logo.



Figure III-13 L'application.

✓ Interface de démarrage :



Figure III-14 Page de démarrage.

✓ **L'interface principale :**

Cette interface permet aux utilisateurs de saisir leurs emails et leurs mots de passe afin d'accéder à leurs interfaces.

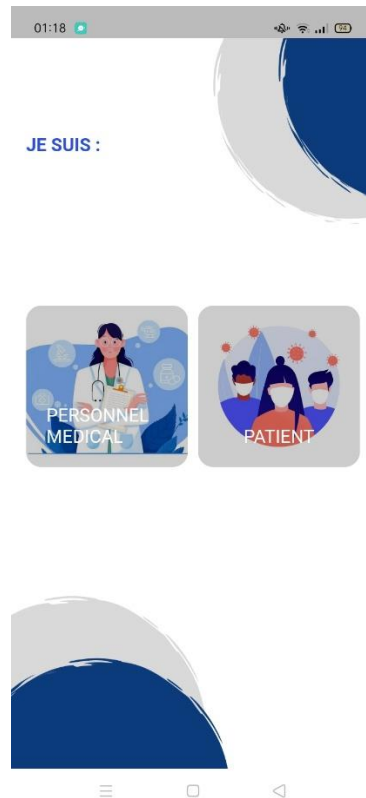


**Figure III-15** Page Connexion.

✓ **La page d'inscription :**

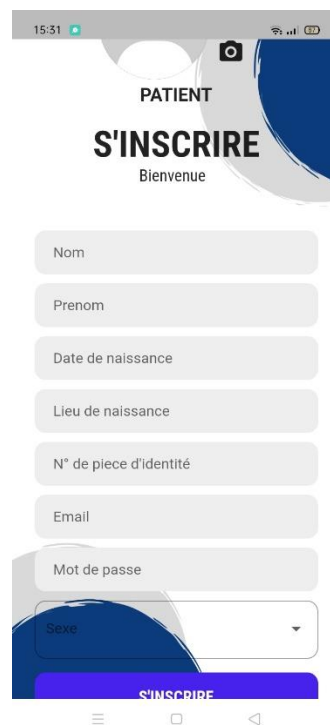
Cette interface permet au nouvel utilisateur de créer un nouveau compte, où il peut choisir son profil soit patient ou personnel médical.





**Figure III-16** Page de choisir le type d'utilisateur.

- Patient :



**Figure III-17** Page de créer compte -Patient-.

• Personnel médical :

The figure displays two screenshots of a mobile application interface for medical personnel registration. The left screenshot, taken at 15:30, shows the initial registration screen. It features a profile picture placeholder, the text "PERSONNEL MEDICAL", a large "S'INSCRIRE" button, and the word "Bienvenue". Below this, there are several input fields for registration details: Nom, Prenom, Date de naissance, Lieu de naissance, N° de piece d'identité, and Email. The right screenshot, taken at 01:18, shows the registration form with the following fields: Nom, Prenom, Date de naissance, Lieu de naissance, N° de piece d'identité, Email, Mot de passe, Sexe (dropdown menu), MEDECIN (dropdown menu), and Spécialité (dropdown menu). A blue "S'INSCRIRE" button is positioned at the bottom of the form.

**Figure III-18** Page de créer compte -personnel médicale-.

✓ **Interface d'accueil :**

Une fois que l'utilisateur est authentifié, il sera dirigé vers une fenêtre d'accueil pour accéder aux différents services d'application.

❖ **Patient :**

Cette page d'accueil contient un menu situé en bas, qui permet d'accéder à quatre pages différentes : Notifications, Accueil, Dossier et Profil.

- **Notification :** La page de notifications regroupe les réponses du personnel médical relatif à l'acceptation ou refus des rendez-vous.

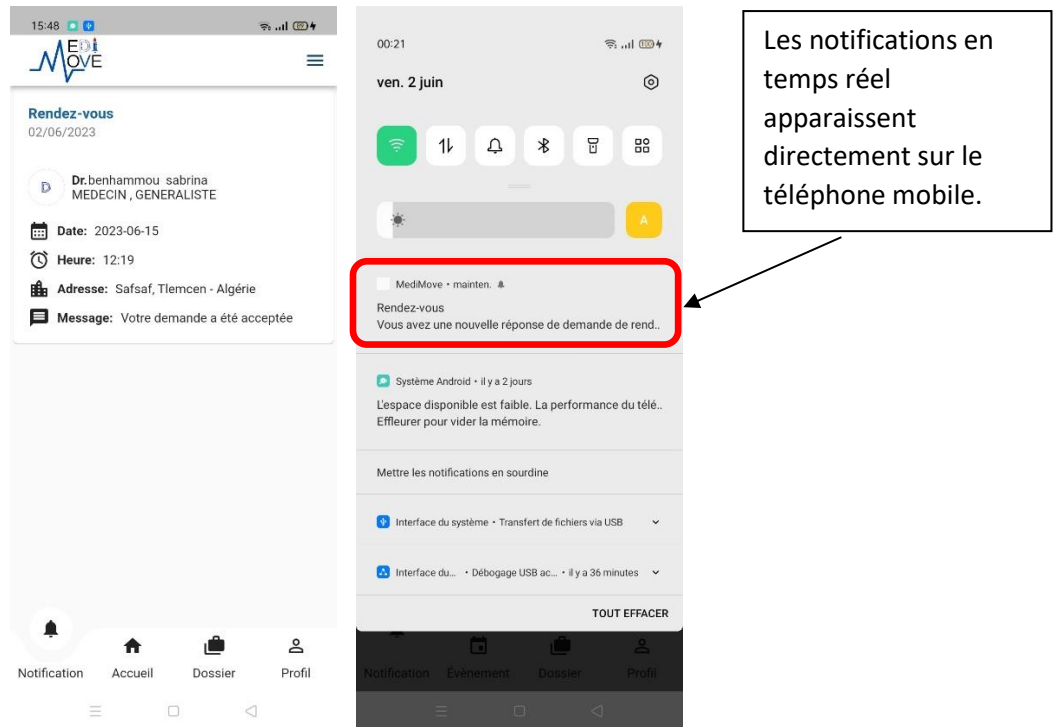


Figure III-19 Page de notification.

- Accueil : La page d'accueil est l'interface où les patients peuvent accéder aux services de l'application.



Figure III-20 Page d'accueil.

- **Dossiers** : La page "Dossiers" permet au patient d'accéder à ses dossiers de consultation, aux rapports médicaux et aux ordonnances générés par le personnel de santé.

Le patient a également la possibilité de télécharger ces documents à partir de bouton "Télécharger le rapport".

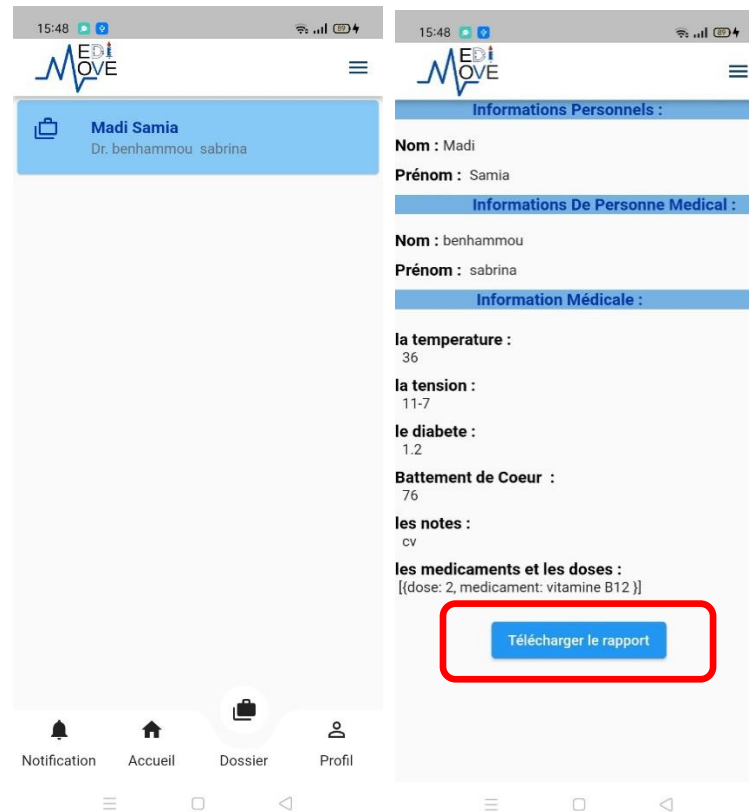


Figure III-21 Page de dossiers.

Il trouvera le fichier téléchargé dans le dossier de téléchargement de son appareil mobile.



**Figure III-22** L'endroit de téléchargement.

- Profil : Cette page représente la page de profil d'utilisateur avec un bouton permettant de l'éditer.

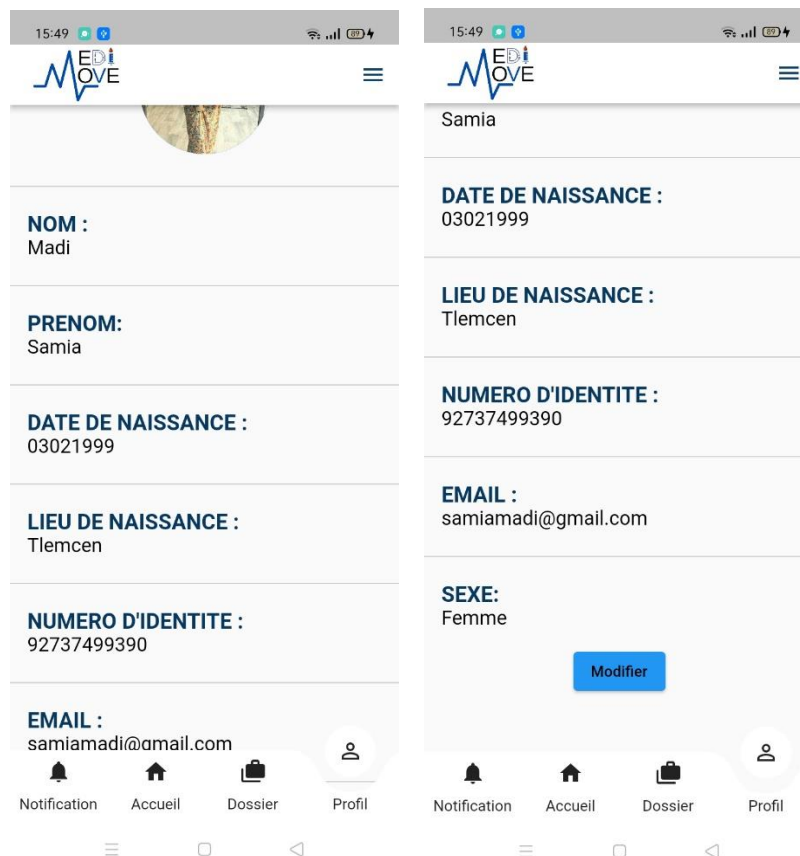


Figure III-23 Page de profil.

❖ Personnel médicale :

Dans la page d'accueil on trouve un menu en bas qui permet d'accéder à quatre différentes pages : Notifications, Évènement, Dossier et Profil.

- Notifications :

La page "Notifications" rassemble les rendez-vous pris par les patients, permettant au personnel médical d'accepter ou de refuser ces rendez-vous après avoir consulté les informations du patient.

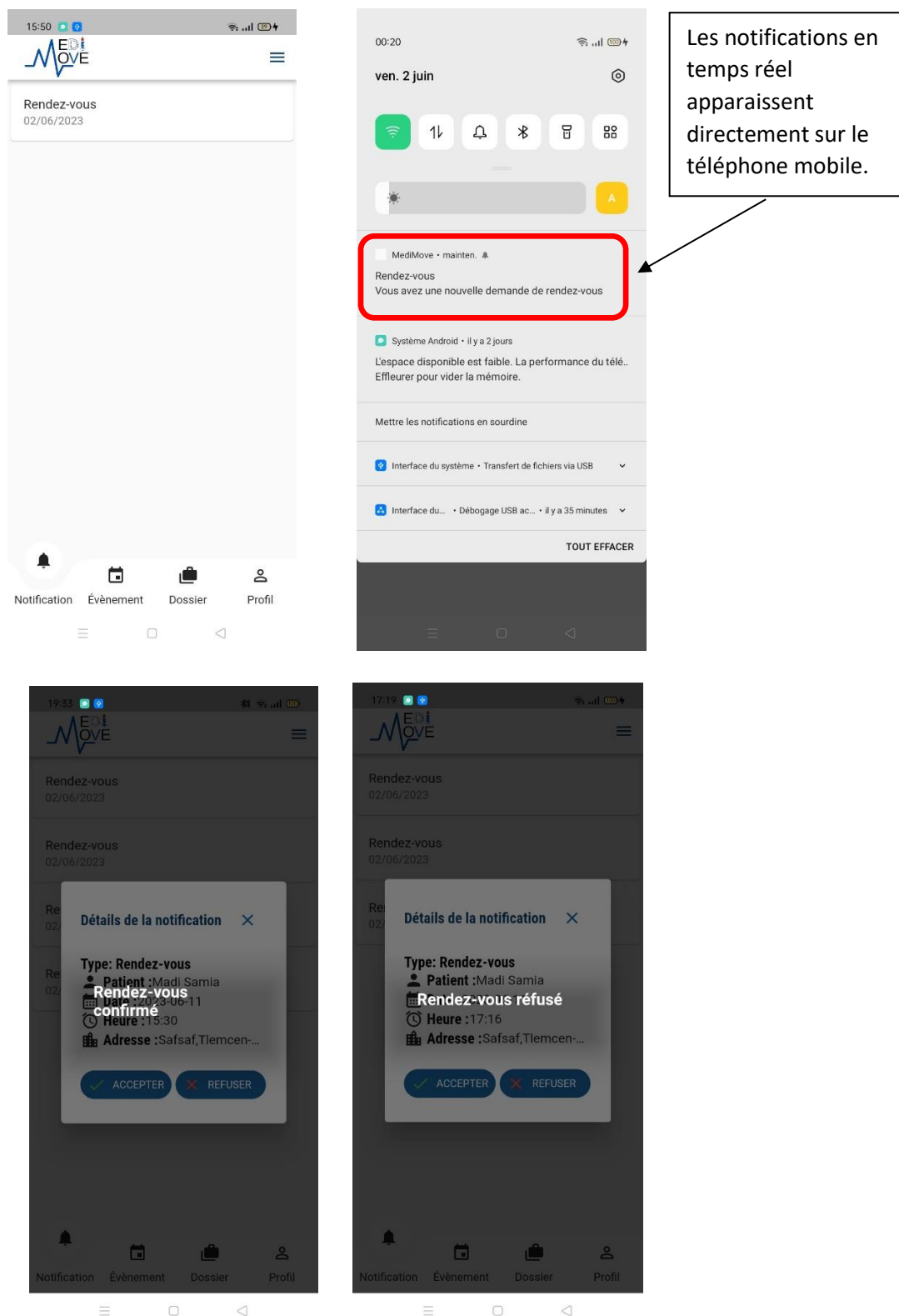


Figure III-24 Page de notification -personnel médicale-.

- Évènement :  
 La page "Événements" présente un calendrier qui répertorie les rendez-vous acceptés. Elle comprend également une section dédiée

aux notes, où les remarques peuvent être saisies pendant la consultation.



Figure III-25 Page d'événement.



Figure III-26 Page des notes de consultation.

- Dossier :

La page "Dossiers" affiche des fichiers contenant les remarques prises pendant la consultation. De plus, un bouton est disponible pour soumettre des rapports au format PDF à partir de cette page.



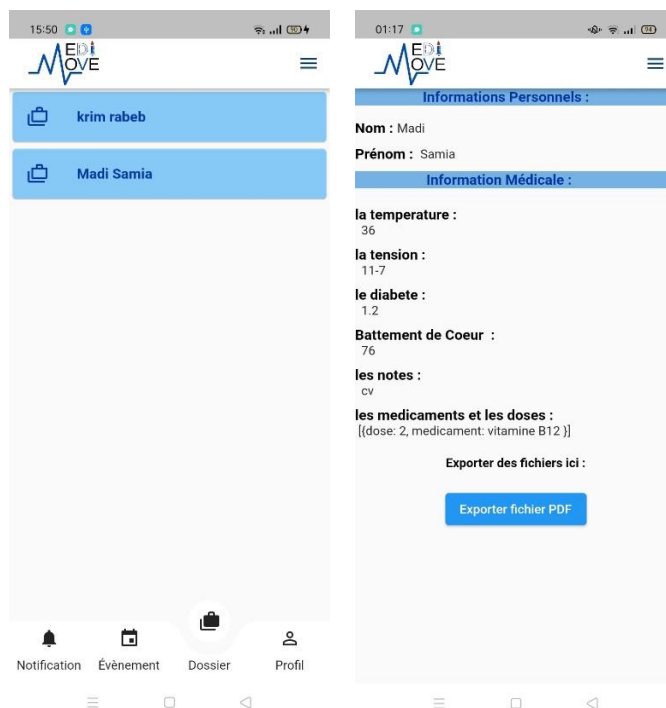


Figure III-27 Page de dossiers - personnel médicale -.

- Profil :

Cette page représente la page de profil avec un bouton pour l’éditer.

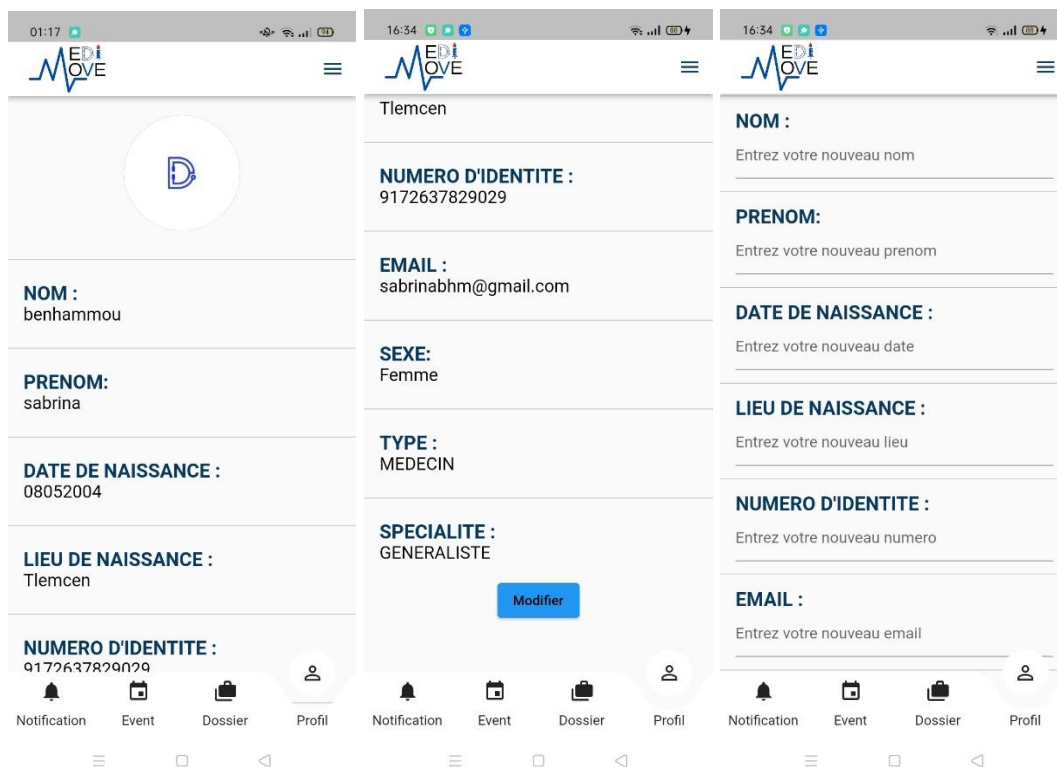
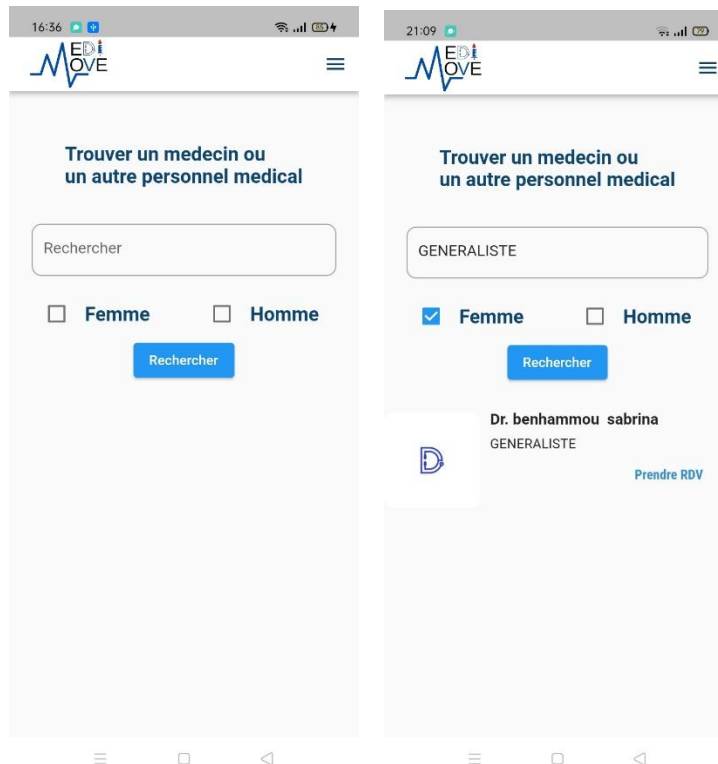


Figure III-28 Page de profil.

✓ **Interface Rechercher :**

Via cette interface, le patient peut effectuer une recherche pour trouver le personnel médical. Il peut choisir la spécialité (s'il veut choisir un médecin), il peut aussi faire un filtre par sexe (homme ou femme).



**Figure III-29** Page de rechercher.

✓ **Interface Prendre un rendez-vous :**

Pour prendre un rendez-vous, le patient doit juste choisir l'heure, la date et la location.

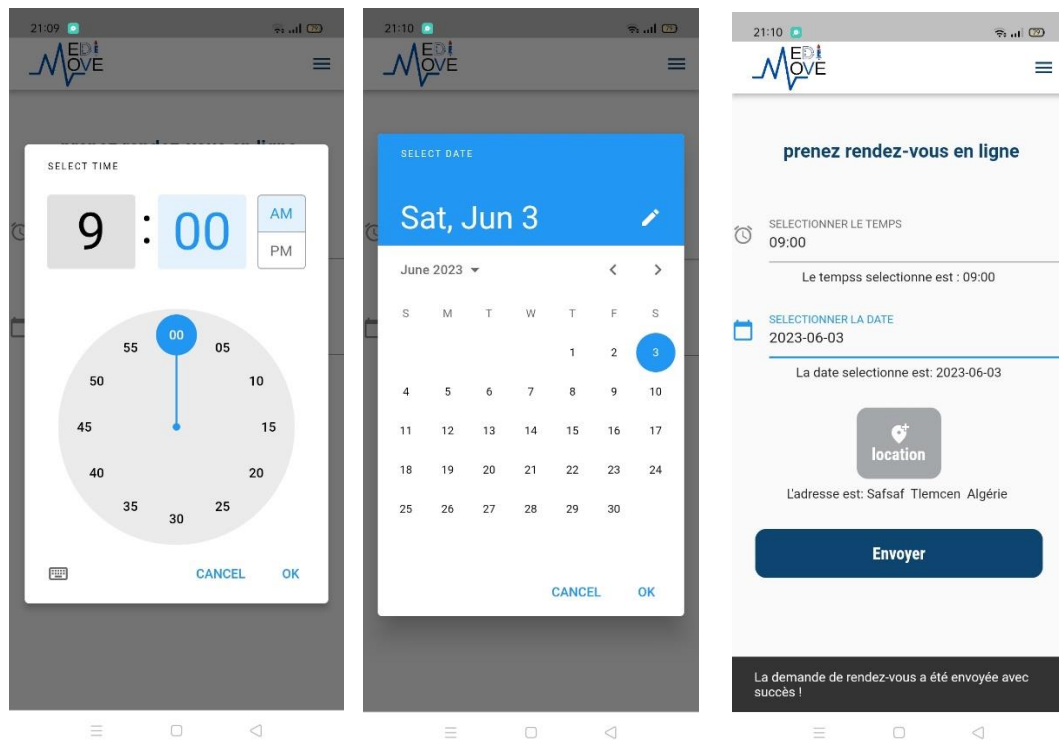


Figure III-30 Page de prendre un rendez-vous.

✓ **Interface menu :**

Pour afficher le menu de l'application, l'utilisateur peut cliquer sur l'icône de menu en haut à gauche ou faire glisser vers la gauche. Ce menu contient quatre pages auxquelles il peut accéder (Contactez-nous, A propos, Aide) et deux icônes : une pour se déconnecter et l'autre pour supprimer le compte définitivement.

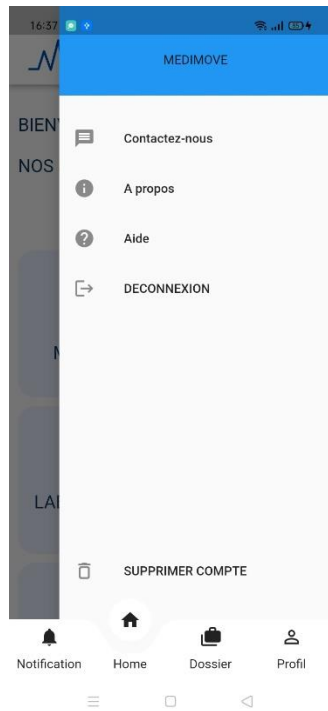


Figure III-31 Interface de menu.

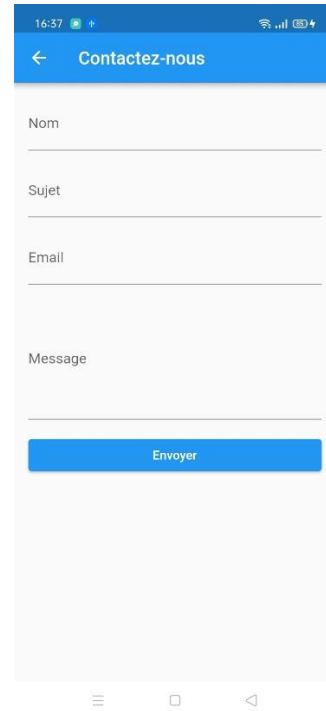


Figure III-32 Page de contactez-nous.



Figure III-33 Page de à-propos.

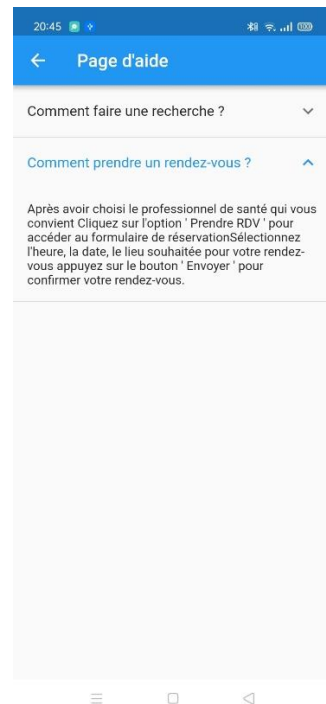


Figure III-34 Page d'aide.

### 3.3.2 Interfaces de la plateforme

#### ✓ Vue d'ensemble de projet :

**Firestore :** C'est un ensemble d'outils et de services hébergés dans le cloud qui facilite la création rapide et la gestion d'applications de haute qualité.

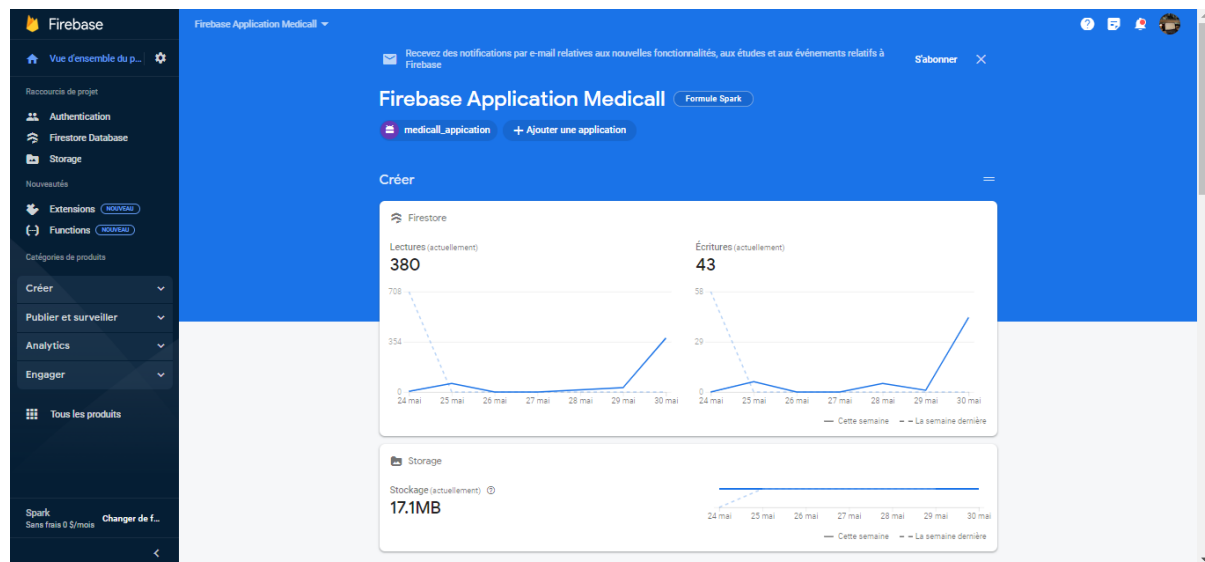


Figure III-35 Interface vue d'ensemble.

#### ✓ Les services utilisés :

##### **Authentification :**

C'est un service de gestion des identités et des accès qui permet aux utilisateurs de s'inscrire, de se connecter et de s'authentifier facilement sur les applications. Firebase Auth offre une prise en charge de l'authentification par e-mail et mot de passe.

Dans notre application, nous avons intégré Firebase Authentication, pour gérer l'authentification des utilisateurs en utilisant des mots de passe et des adresses e-mail.

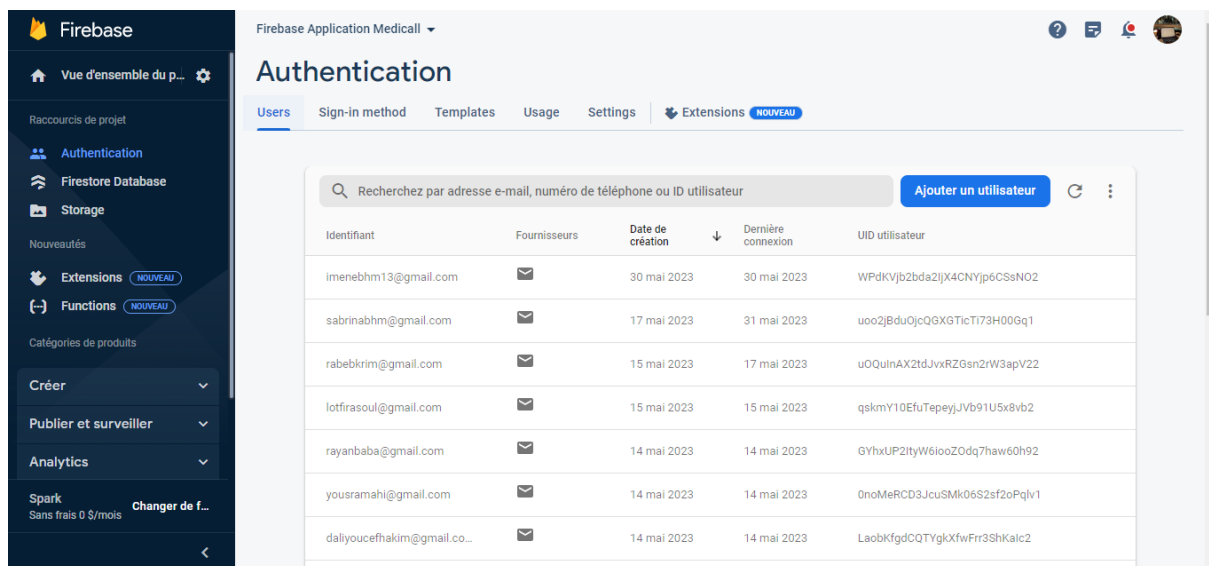


Figure III-36 Interface d'authentification.

**Firestore Database :**

Il s'agit d'une base de données NoSQL en temps réel et évolutive dans le cloud Firebase.

Nous avons utilisé Cloud Firestore pour stocker et synchroniser facilement les données entre les utilisateurs de l'application.

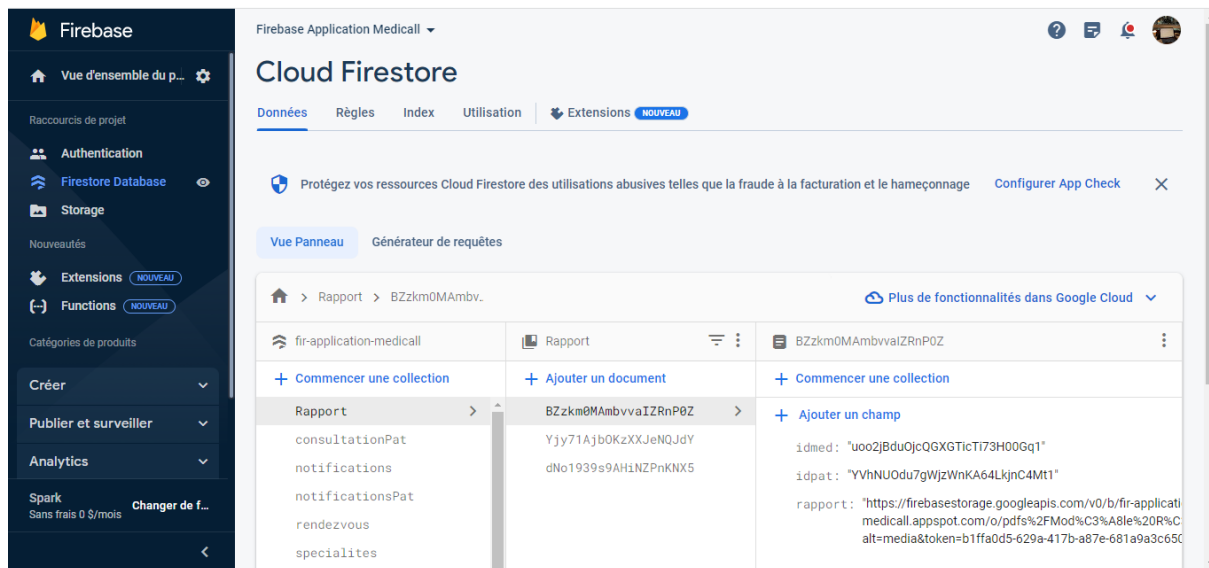


Figure III-37 Interface de Firestore Database.

**Storage :**

Il s'agit d'un service de stockage en nuage (cloud) qui permet de sauvegarder et de récupérer des fichiers, tels que des images, des vidéos et des documents, dans le cloud Firebase.

Nous avons utilisé Cloud Storage pour gérer le stockage des images de profil de nos utilisateurs et faciliter les téléversements et les téléchargements de fichiers pour le personnel médical et les patients de notre application.

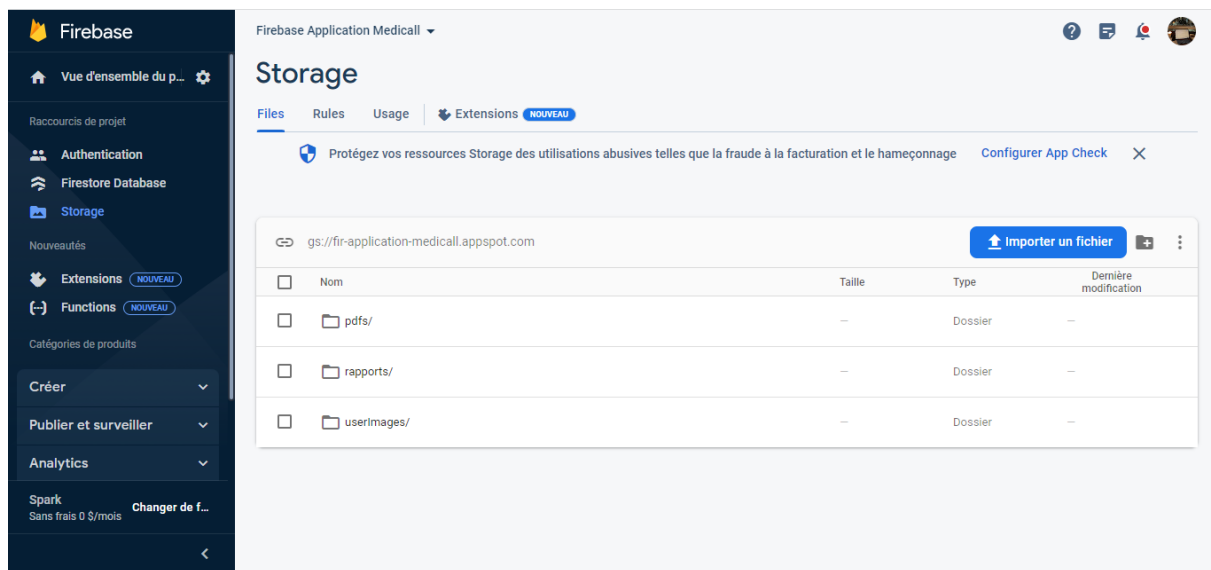


Figure III-38 Interface de cloud Storage.

### 3.4 Conclusion

Dans ce chapitre, nous avons présenté notre application finale, en essayant d'être fidèle à la majorité des besoins et de respecter la conception élaborée. Plus précisément, nous avons fourni une explication en détail des différentes interfaces de notre application MediMove.

# Chapitre IV BMC





الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي و البحث العلمي  
جامعة أوبكر بلكايد - تلمسان -



جامعة أبو بكر بلقايد  
UNIVERSITY OF TLEMEN

## Business Model Canevas

### نموذج العمل التجاري

تاريخ الإيداع: 14/06/2023

الطالب:

الاسم : ايمان

اللقب : بن حمو

رمز المشروع :

الموخر:

الاسم : زينب ياسمينه

اللقب : رماسي



اسم المشروع : Développement d'une application mobile médical

السنة الجامعية : 2022/2023

## 4.1 Value propositions (Proposition de valeur)

1. **Offrir un moyen pratique et facile de demander des soins médicaux à domicile :** Notre application conviviale et intuitive permet aux patients d'obtenir une assistance médicale à domicile en quelques clics. Plus besoin de voyager ou d'attendre dans les salles d'attente des hôpitaux ou des cliniques pendant des heures.
2. **Permettre aux patients de bénéficier de soins personnalisés :** Les options de soins à domicile permettent aux patients de bénéficier de soins plus individualisés. Les professionnels de la santé travaillent directement avec les patients pour établir une relation de confiance qui leur permet de mieux comprendre leurs besoins spécifiques.
3. **Offrir des alternatives aux visites à l'hôpital et à la clinique :** Les visites à l'hôpital et à la clinique peuvent être inconfortables et stressantes pour les patients. Grâce aux soins de santé à domicile, nous offrons une alternative plus confortable et accessible qui réduit les niveaux d'anxiété et améliore la santé des patients.
4. **Assurer la continuité des soins :** Les soins à domicile assurent la continuité des soins pour les patients qui nécessitent un suivi médical régulier. Un professionnel de la santé peut suivre de près la progression de l'état d'un patient et adapter le traitement en conséquence.
5. **Soulager la pression sur les établissements de santé :** les soins de santé à domicile peuvent aider à réduire la pression en offrant une alternative aux visites à l'hôpital ou à la clinique.

## 4.2 Customer segments (Segments clients)

1. **Personnes ayant besoin de soins médicaux à domicile :** Les soins à domicile profitent aux personnes ayant besoin de soins réguliers ou spéciaux. Cela inclut de patients atteints de maladies chroniques, de blessures postopératoires ou de conditions nécessitant une surveillance étroite.
2. **Personnes à mobilité réduite :** Les personnes ayant des difficultés à se déplacer en raison de problèmes de mobilité trouvent des solutions qui répondent à leurs besoins en soins à domicile. Les travailleurs de la santé se rendent directement

chez eux, évitant ainsi les obstacles liés aux transports ou à l'accessibilité des établissements médicaux.

3. **Personnes ne pouvant pas se déplacer en raison de leur état de santé :** Certaines personnes sont physiquement incapables de se déplacer en raison de leur état de santé fragile. Le service de réservation d'une ambulance pour les emmener à l'hôpital leur offre la possibilité de voyager confortablement.
4. **Personnes ayant besoin d'un suivi médical régulier :** Certains patients nécessitent un suivi médical régulier pour gérer leur condition de santé. A cet effet, toutes les informations cruciales de chaque patient (telles que les ordonnances, les rapports, les diagnostics, les résultats des analyses) sont enregistrées dans un dossier patient.

### 4.3 Customer relationships (Relation client)

1. Support client : fournissez un support client réactif et de haute qualité pour répondre aux questions, préoccupations et demandes d'assistance de nos utilisateurs. Ils peuvent utiliser une ligne téléphonique dédiée, une messagerie intégrée à l'application ou une assistance par e-mail.
2. Collecte de feedback : mettre en place des mécanismes pour recueillir les commentaires et suggestions des utilisateurs afin d'améliorer en permanence nos applications et services. Nous utilisons des sondages, des évaluations ou des commentaires sur les médias sociaux de notre application pour recueillir ces précieuses informations.
3. Communication avec les professionnels de la santé : nous établissons une communication fluide et efficace avec les professionnels de la santé inscrits dans l'application. Cela se fait par e-mail et lors de réunions régulières.
4. Communications marketing : Nous utilisons des canaux de communication marketing, tels que les médias sociaux et les campagnes de sensibilisation, pour informer et engager nos utilisateurs. Et on partage avec eux des informations utiles, des mises à jour sur les nouvelles fonctionnalités ou des conseils de santé connexes.

### 4.4 Channels (Canaux de distribution)

Notre application est un outil essentiel pour connecter les patients et les professionnels de la santé afin de fournir des soins pratiques et de haute qualité à

domicile. Nous fournissons des solutions innovantes qui répondent aux besoins changeants du secteur de la santé et facilitent les soins à domicile. Il offre une accessibilité optimale pour les utilisateurs d'appareils Android car il peut être téléchargé à partir des principaux magasins d'applications, en particulier le Play Store.

## 4.5 Key Partners (Partenaires clés)

1. **Médecins, infirmières et les aides-soignants prodiguant des soins à domicile** : ces partenaires clés sont des professionnels de la santé qui interviennent directement auprès des patients à leur domicile. Les médecins peuvent diagnostiquer, prescrire des traitements et superviser les soins, tandis que les infirmières et les aides-soignants fournissent des soins médicaux directs, tels que des injections et des bandages, et une aide à la vie quotidienne. Ils jouent un rôle clé dans la prestation de soins de qualité aux patients et dans le soutien de leur rétablissement dans un environnement confortable et familial.
2. **Laboratoires d'analyses pour effectuer des prélèvements sanguins à domicile** : Dans le cadre des soins médicaux à domicile, il est souvent nécessaire d'effectuer des tests sanguins pour diagnostiquer et surveiller l'état de santé d'un patient. Les laboratoires d'analyses sont des partenaires clés qui proposent des services de prélèvements sanguins à domicile. Ils envoient des professionnels formés et qualifiés pour prélever des échantillons de sang chez les patients et les transmettent ensuite au laboratoire pour les analyses. Cela le rend plus pratique pour les patients car ils n'ont pas à se rendre au laboratoire pour obtenir ces tests.
3. **Services de transport sanitaire et d'ambulance pour le déplacement des patients** : certains patients peuvent avoir besoin d'une assistance supplémentaire pour se déplacer de leur domicile vers des établissements médicaux ou pour des consultations médicales externes. Les services de transport sanitaire et d'ambulance sont des partenaires clés qui fournissent des moyens de transport spécialisés adaptés aux besoins médicaux des patients. Ils peuvent fournir des ambulances ou des véhicules équipés pour le transport des patients à mobilité réduite, des personnes nécessitant une surveillance médicale continue ou les personnes nécessitant des soins urgents. Ces partenaires garantissent un transport sûr et adéquat des patients.

## 4.6 Key activities (Activités clés)

1. **Développement et maintenance d'applications mobiles :** Cette activité majeure consiste en la conception, le développement et la maintenance de l'application. Cela comprend la programmation, la conception de l'interface utilisateur, les tests, l'optimisation des performances et les mises à jour régulières pour garantir le bon fonctionnement de l'application, pour assurer une expérience utilisateur fluide et facile à utiliser.
2. **Marketing et acquisition des patients et des professionnels de la santé :** Cette activité importante vise à promouvoir l'application auprès des patients et des professionnels de la santé et à attirer de nouveaux utilisateurs. Cela comprend le développement de stratégies de marketing et de communication, la création de campagnes publicitaires ciblées, la participation à des événements du secteur de la santé, la collaboration avec des partenaires et la sensibilisation sur plusieurs canaux. L'objectif est de faire connaître l'application et d'attirer davantage d'utilisateurs et de professionnels de la santé intéressés par les soins à domicile.
3. **Gestion des abonnements :** Cette activité clé implique la mise en place d'un système de gestion des abonnements pour les professionnels de la santé. Elle comprend la création de plans d'abonnement et la gestion des comptes des utilisateurs. Il est important d'avoir une infrastructure solide pour gérer les transactions financières de manière sécurisée, ainsi qu'un système de support client pour résoudre les problèmes liés aux paiements de l'abonnements. La gestion efficace des abonnements garantit la rentabilité de l'entreprise et facilite la relation entre l'application et les professionnels de la santé.

## 4.7 Cost structure (Structure de coûts)

1. **Marketing et publicité pour attirer les utilisateurs :** Cette catégorie comprend les dépenses liées à la promotion et à la publicité de l'application afin d'attirer de nouveaux utilisateurs. Cela peut inclure les coûts de publicité en ligne, tels que les annonces payantes sur les moteurs de recherche ou les médias sociaux, ainsi que les campagnes de marketing traditionnelles, comme les spots publicitaires à la télévision, les affiches. Pour notre application, nous avons estimé un coût annuel

de 600 000 DA pour le marketing et la publicité, qui sera alloué à ces différentes activités promotionnelles.

2. **Développement et maintenance de l'application :** Cette catégorie de coûts comprend les coûts associés à la conception, au développement et à la maintenance de l'application. Cela peut inclure les salaires des développeurs, les coûts liés à l'infrastructure informatique (frais d'hébergement), les achats de logiciels ou de licences, les mises à jour d'application et la correction de bogues. Nous avons estimé un coût mensuel de 50 000 DA pour un développeur et 40 000 DA pour un mainteneur, couvrant ainsi les dépenses salariales liées à ces rôles clés dans le processus de développement et de maintenance de l'application.

## 4.8 Revenue streams (Sources de revenus)

Notre application médicale mobile offre aux professionnels de la santé une opportunité unique de rejoindre notre plateforme avec un abonnement mensuel de 3 000 DA ou un abonnement annuel de 36 000 DA. En payant ces frais récurrents, les praticiens bénéficieront d'une visibilité accrue auprès des patients ainsi que du privilège d'accéder aux nombreuses opportunités de soins à domicile disponibles via notre application. Grâce à notre vaste réseau de patients et à notre interface conviviale, les professionnels de la santé peuvent facilement étendre leur pratique et fournir des soins de qualité à un large éventail de patients dans le confort de leur foyer.

Notre application mobile offre d'excellentes opportunités de partenariat pour les sociétés pharmaceutiques, les cliniques spécialisées et les services de santé connexes. En proposant des espaces publicitaires in-app, nous offrons à nos partenaires la possibilité d'atteindre directement notre base d'utilisateurs cibles. Ces partenaires sont facturés en fonction du nombre de vues, de clics ou de conversions générés par leurs annonces, garantissant un retour sur investissement mesurable. Nous travaillons en étroite collaboration avec nos partenaires pour offrir des expériences publicitaires pertinentes et discrètes tout en générant un flux de revenus stable et durable pour notre entreprise.

## 4.9 Key ressources (Ressources clés)

Avant tout, l'une des ressources les plus importantes de notre application mobile médicale est notre personnel médical qualifié. Nous nous appuyons sur une équipe compétente et expérimentée de médecins, infirmiers, d'aide-soignant et rééducateur. Leur expertise et leur dévouement assurent la prestation de soins à domicile de qualité à leurs usagers. Outre les équipes médicales, l'application mobile est également l'une des ressources importantes. Elle dispose d'une interface conviviale et intuitive qui permet aux utilisateurs d'accéder aux soins médicaux à domicile en quelques clics. L'application facilite également la gestion des rendez-vous, l'accès aux résultats des tests...etc.

Nous avons besoins également des partenariats avec des laboratoires d'analyse. Ces collaborations stratégiques nous permettent de fournir aux utilisateurs des services de prélèvement sanguin pour faire les tests à domicile, améliorant ainsi la commodité et l'efficacité des soins de santé. Grâce à ces partenariats, nous pouvons garantir des délais de réponse rapides et des résultats fiables.

Enfin, nous avons aussi besoin du partenariat avec une société de transport sanitaire terrestre. Ces partenaires jouent un rôle essentiel dans la logistique des soins à domicile, garantissant que les patients sont transportés en toute sécurité et rapidement vers les hôpitaux ou les cliniques. Leur service professionnel assure une coordination efficace et une réponse rapide aux demandes de transfert de patients.

En plus des ressources clés mentionnées précédemment, notre équipe de développement qui sont équipés des PC performants joue un rôle essentiel dans la mise à jour et la maintenance de notre application mobile. Ces spécialistes du développement des applications mobile travaillent en étroite collaboration pour s'assurer que notre application reste efficace, sécurisée et répond aux besoins changeants de nos utilisateurs.

Nom de l'entreprise :  
MediMove

Date:  
18/05/2023

Business Model Canvas

Tableau IV-1 Tableau de BMC.

<b>Partenaires clés</b> <ul style="list-style-type: none"> <li>• Médecin.</li> <li>• Infirmier.</li> <li>• Aide-soignant.</li> <li>• Rééducateur.</li> <li>• Laboratoire d'analyse.</li> <li>• Service de transport sanitaire et d'ambulance.</li> </ul>	<b>Activités clés</b> <ul style="list-style-type: none"> <li>• Développement et maintenance de l'application.</li> <li>• Marketing et acquisition des patients et des professionnels de la santé.</li> <li>• Admin pour gestion des abonnements.</li> </ul>	<b>Offre (proposition de valeur)</b> <ul style="list-style-type: none"> <li>• Offrir un moyen facile et pratique.</li> <li>• Demander les soins médicaux à domicile.</li> <li>• Bénéficiaire de soins personnalisés.</li> <li>• Offrir des alternatives aux visites à l'hôpital et à la clinique.</li> <li>• Soulager la pression sur les établissements de santé.</li> </ul>	<b>Relation client</b> <ul style="list-style-type: none"> <li>• Support client</li> <li>• Communication avec les professionnels de la santé</li> <li>• Communication marketing</li> <li>• Collecte les feedbacks à partir des réseaux sociaux.</li> </ul>	<b>Segments de clientèle</b> <ul style="list-style-type: none"> <li>• Patient à mobilité réduite.</li> <li>• Patient âgé.</li> <li>• Personne qui ne peuvent pas se déplacer.</li> <li>• Le personnel médical qui propose le déplacement à domicile.</li> </ul>
	<b>Ressources clés</b> <ul style="list-style-type: none"> <li>• L'équipe des professionnels de la santé.</li> <li>• Les sociétés de transport sanitaire terrestre.</li> <li>• Les laboratoires d'analyses.</li> </ul>		<b>Canaux de distribution</b> <ul style="list-style-type: none"> <li>• Application mobile dans Play store (Android).</li> </ul>	



Structure des coûts	Sources de revenus
<ul style="list-style-type: none"><li>• Développement et maintenance de l'application (110000 DA/mois).</li><li>• Marketing et publicité pour l'application (600 000 DA/année)</li><li>• Les frais de déploiement de l'application en Play store (25\$).</li></ul>	<ul style="list-style-type: none"><li>• L'abonnements annuel des professionnels de santé qui cout 36000 DA.</li><li>• Espace publicitaire in-app.</li></ul>

# Conclusion générale

Dans notre projet de fin d'études, nous avons développé une application mobile médicale visant à renforcer le lien entre le personnel médical et les patients. Cette application représente une avancée significative dans le domaine médical en Algérie, car elle vise à améliorer l'accessibilité aux soins de santé, en particulier pour les personnes à mobilité réduite et les personnes âgées. Elle permet aux patients de bénéficier de consultations médicales dans le confort de leur domicile.

Notre projet innovant a abouti à la création d'une application conviviale qui facilite les visites médicales à domicile en fournissant des services de qualité et fiables. En combinant les avantages de la technologie mobile avec la médecine traditionnelle, nous avons créé de nouvelles opportunités pour répondre aux besoins des patients et améliorer l'expérience des services de santé.

La première étape de notre étude a consisté à se concentrer sur les applications mobiles, leurs types et leurs systèmes d'exploitation. Ensuite, nous avons procédé à l'analyse et à la collecte des besoins, suivies de la conception en utilisant le langage de modélisation UML. Enfin, nous avons présenté la solution proposée en utilisant le Framework Flutter et le langage de programmation Dart.

Ce travail nous a permis de mettre en œuvre une application mobile médicale souple, flexible et conviviale. En terminant ce projet, il est essentiel de noter que nous avons pu couvrir la majorité des fonctionnalités. Cependant, nous pouvons toujours améliorer notre travail en ouvrant la voie vers de nouvelles perspectives. Nous envisageons d'ajouter des fonctionnalités telles que la possibilité d'envoyer les documents par e-mail, intégrer un système de rappels de rendez-vous pour le personnel médical afin de réduire les oublis et les absences, et d'ajouter une fonctionnalité de messagerie sécurisée entre les patients et le personnel médical.

# Références

- [1] «Mobile Data Science and Intelligent Apps: Concepts, AI-Based Modeling and Research Directions,» [En ligne]. Available: <https://trends.google.com/trends/>, 2019. [Accès le 31 05 2023].
- [2] «Marché des applications mobiles: +83% de téléchargements en 5 ans,» [En ligne]. Available: <https://fr.goodbarber.com/creer-app/94-marche-des-applications-mobiles-83-de-telechargements-en-5-ans/>. [Accès le 23 05 2023].
- [3] «Global mHealth Apps Market Will Reach USD 111.1 Billion By 2025: Zion Market Research,» [En ligne]. Available: <https://www.globenewswire.com/news-release/2019/01/24/1704860/0/en/Global-mHealth-Apps-MarketWill-Reach-USD-111-1-Billion-By-2025-Zion-Market-Research.html>. [Accès le 29 05 2023].
- [4] Claire Villiers, «E-santé : un marché gigantesque,» [En ligne]. Available: <https://fr.statista.com/infographie/28123/ventes-dans-le-secteur-e-sante-numerique/#:~:text=Les%20applications%20num%C3%A9riques%20de%20sant%C3%A9,Health%20Market%20Outlook%20de%20Statista..> [Accès le 23 05 2023].
- [5] F. Poirier et M. Lehoux, «besoins de compétences dans le développement d applications mobiles,» 2013. [En ligne]. Available: <https://docplayer.fr/1107511-Besoins-de-competences-dans-le-developpement-d-applications-mobiles.html>. [Accès le 08 06 2023].
- [6] B. Beinard, «Les parts de marché des systèmes d'exploitation mobiles iOS, Android et Windows en 2023,» [En ligne]. Available: <https://www.leptidigital.fr/technologie/parts-de-marche-systemes-exploitation-mobiles-ios-android-windows-12957/>. [Accès le 20 05 2023].
- [7] P. Christensson, «Android Definition,» [En ligne]. Available: <https://techterms.com/definition/android>. [Accès le 20 05 2023].
- [8] L. R. D. FUTURA, «iOS : qu'est-ce que c'est ?,» [En ligne]. Available: <https://www.futura-sciences.com/tech/definitions/multimedia-ios-15211/>. [Accès le 20 05 2023].
- [9] [En ligne]. Available: <https://www.codeur.com/blog/developpement-application-mobile-native/>.
- [10] L. AZOULAY, «« Quelle technologie choisir pour créer une application mobile ? »,» [En ligne]. Available: <https://www.adimeo.com/blog/quelle-technologie-pour-une-application-mobile>. [Accès le 19 05 2023].
- [11] A. Cazes et D. Joëlle , «Développer une application web,» [En ligne]. Available: <https://www.dunod.com/sites/default/files/atoms/files/9782100743759/Feuilletage.pdf>.
- [12] N. Mérouze, «Qu'est-ce qu'une application web ? Définition [2019],» [En ligne]. Available: <https://www.synbioz.com/blog/definition-application-web>. [Accès le 03 06 2023].
- [13] «Qu'est ce qu'une application mobile ?,» [En ligne]. Available: <https://www.numidev.fr/une-application-mobile-cest-quoi/>. [Accès le 19 05 2023].
- [14] «APPLICATIONS HYBRIDES : AVANTAGES, LIMITATIONS & CONSÉQUENCES DE TEST,» [En ligne]. Available: <https://www2.stardust-testing.com/blog-fr/applications-hybrides-avantages-limitations-consequences-phases-de-test>. [Accès le 19 05 2023].

- [15] M. LB, «Quel Framework pour développer une application mobile en 2020 ?», [En ligne]. Available: <https://www.maximelb.com/quel-framework-pour-developper-une-application-mobile-en-2020/#:~:text=Framework%20cr%C3%A9%C3%A9%20par%20Facebook%2C%20React,lan>.
- [16] «Qu'est-ce que Xamarin ?», [En ligne]. Available: <https://learn.microsoft.com/fr-fr/xamarin/get-started/what-is-xamarin>. [Accès le 22 05 2023].
- [17] «React Native vs Xamarin vs Ionic vs Flutter : quel est le meilleur en 2023 ?», [En ligne]. Available: <https://graffersid.com/react-native-vs-xamarin-vs-ionic-vs-flutter/#>. [Accès le 22 05 2023].
- [18] «Flutter : Présentation du framework cross-platform de Google», [En ligne]. Available: <https://citronnoir.com/creer-une-application-mobile/flutter-loutil-de-developpement-dapplication-moderne-cree-par-google/>. [Accès le 22 05 2023].
- [19] «Le concept des widgets Flutter et votre première application», [En ligne]. Available: <https://pythonforge.com/widgets-flutter/>. [Accès le 23 05 2023].
- [20] M. Napoli, «Beginning Flutter — Intermediate — Using Common Widgets», [En ligne]. Available: <https://medium.com/@JediPixels/beginning-flutter-intermediate-using-common-widgets-b4a01fe9c8b3>. [Accès le 27 05 2023].
- [21] P. Moati, «Flutter pour le développement de vos applications mobiles cross-platform», [En ligne]. Available: <https://www.kaliop.com/fr/flutter-pour-le-developpement-de-vos-applications-mobiles-cross-platform/>. [Accès le 23 05 2023].
- [22] «Dart : présentation du langage de programmation», [En ligne]. Available: <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/le-langage-de-programmation-dart/>. [Accès le 23 05 2023].
- [23] E. Alcouffe, «A quoi sert Firebase, la plateforme mobile de Google?», [En ligne]. Available: <https://junto.fr/blog/firebase/>. [Accès le 21 05 2023].
- [24] «Base de données en temps réel», [En ligne]. Available: <https://firebase.flutter.dev/docs/database/overview/>.
- [25] [En ligne]. Available: <https://firebase.google.com/docs/ml-kit?hl=fr>.
- [26] [En ligne]. Available: <https://blog.back4app.com/fr/quest-ce-que-firebase-predictions/#:~:text=Firebase%20Predictions%20est%20bas%C3%A9%20sur,les%20utilisateurs%20du%20service%20firebase..>
- [27] [En ligne]. Available: <https://www.lucidchart.com/pages/fr/langage-uml>.
- [28] «Découvrir Android Studio», [En ligne]. Available: <https://developer.android.com/studio/intro?hl=fr>. [Accès le 13 06 2023].
- [29] [En ligne]. Available: [https://ar.wikipedia.org/wiki/%D9%85%D9%84%D9%81:Android\\_Studio\\_Trademark.svg](https://ar.wikipedia.org/wiki/%D9%85%D9%84%D9%81:Android_Studio_Trademark.svg). [Accès le 22 06 2023].
- [30] [En ligne]. Available: <https://www.ummtto.dz/dspace/bitstream/handle/ummtto/12948/BenTouradSadia.pdf?sequence=1>.

- [31] [En ligne]. Available: [https://fr.m.wikipedia.org/wiki/Fichier:Android\\_logo\\_2019\\_\(stacked\).svg](https://fr.m.wikipedia.org/wiki/Fichier:Android_logo_2019_(stacked).svg).
- [32] CECILE, «Quels sont les avantages du système d'exploitation iOS ?,» [En ligne]. Available: <https://www.numeriguide.com/quels-sont-les-avantages-du-systeme-dexploitation-ios%E2%80%89>.
- [33] «Pourquoi les applications mobiles séduisent-elle ? Avantages et contraintes,» [En ligne]. Available: <https://bemobee.com/avantages-et-contraintes-des-applications-mobiles/>.
- [34] «APPLICATIONS HYBRIDES : AVANTAGES, LIMITATIONS & CONSÉQUENCES DE TEST,» [En ligne]. Available: <https://www2.stardust-testing.com/blog-fr/applications-hybrides-avantages-limitations-consequences-phases-de-test>.
- [35] «React Native pour le développement de vos applications mobiles,» [En ligne]. Available: <https://www.ideematic.com/dictionnaire-digital/react-native/>.
- [36] «React Native pour le développement de vos applications mobiles,» [En ligne]. Available: <https://www.ideematic.com/dictionnaire-digital/react-native/>.
- [37] «React Native pour le développement de vos applications mobiles,» [En ligne]. Available: <https://www.ideematic.com/dictionnaire-digital/react-native/>.
- [38] «Fichier:Google-flutter-logo.png,» [En ligne]. Available: <https://fr.wikipedia.org/wiki/Fichier:Google-flutter-logo.png>.
- [39] «Ionic Tutorial,» [En ligne]. Available: <https://www.javatpoint.com/ionic>.
- [40] «Xamarin Tutorial,» [En ligne]. Available: <https://www.javatpoint.com/xamarin>.
- [41] R. HAJI, «Qu'est ce qu'un widget dans Flutter ?,» [En ligne]. Available: <https://apcpedagogie.com/quest-ce-quun-widget-dans-flutter/>.
- [42] M. Gerken, «StatelessWidget vs. StatefulWidget,» [En ligne]. Available: <https://www.flutterclutter.dev/flutter/basics/statelesswidget-vs-statefulwidget/2020/1195/>.
- [43] «Fichier:Logo du langage de programmation Dart.svg,» [En ligne]. Available: [https://ar.wikipedia.org/wiki/%D9%85%D9%84%D9%81:Dart\\_programming\\_language\\_logo.svg](https://ar.wikipedia.org/wiki/%D9%85%D9%84%D9%81:Dart_programming_language_logo.svg).
- [44] «File:Firebase Logo.svg,» [En ligne]. Available: [https://commons.wikimedia.org/wiki/File:Firebase\\_Logo.svg](https://commons.wikimedia.org/wiki/File:Firebase_Logo.svg).
- [45] «Beesiha,» [En ligne]. Available: <https://play.google.com/store/apps/details?id=io.beesolution.beesiha&hl=fr&gl=FR>.
- [46] [En ligne]. Available: <https://apps.apple.com/tr/app/esiha/id1551554967?platform=iphone>.
- [47] [En ligne]. Available: <https://play.google.com/store/apps/details?id=com.ibnhamza.tele>.

# Liste des figures

<b>Figure I-1</b> Tendances d'intérêt des utilisateurs au fil du temps.....	4
<b>Figure I-2</b> E-santé. ....	6
<b>Figure I-3</b> Android. ....	7
<b>Figure I-4</b> iOS. ....	8
<b>Figure I-5</b> Application native.....	8
<b>Figure I-6</b> Application web.....	9
<b>Figure I-7</b> Application hybride. ....	10
<b>Figure I-8</b> Logo React Native. ....	13
<b>Figure I-9</b> logo flutter.....	13
<b>Figure I-10</b> logo Ionic. ....	14
<b>Figure I-11</b> logo Xamarin. ....	15
<b>Figure I-12</b> flutter vs React vs Ionic vs Xamarin.....	19
<b>Figure I-13</b> les widgets.....	20
<b>Figure I-14</b> L'arborescence de widget.....	22
<b>Figure I-15</b> Stateless vs Stateful. ....	23
<b>Figure I-16</b> logo Dart. ....	24
<b>Figure I-17</b> logo Firebase.....	24
<b>Figure I-18</b> Services de Firebase.....	26
<b>Figure II-1</b> Diagramme de Gantt. ....	31
<b>Figure II-2</b> eSiha. ....	32
<b>Figure II-3</b> Beesiha. ....	33
<b>Figure II-4</b> eTabib.....	34
<b>Figure II-5</b> Diagramme de cas d'utilisation. ....	37
<b>Figure II-6</b> Diagramme de séquence « Authentification ». ....	41
<b>Figure II-7</b> Diagramme de séquence « Modifier compte ». ....	43
<b>Figure II-8</b> Diagramme de séquence « Rechercher personnel médical ». ....	44
<b>Figure II-9</b> Diagramme de séquence « Prendre rendez-vous ». ....	45
<b>Figure II-10</b> Diagramme d'activité « Prendre un rendez-vous ». ....	46
<b>Figure II-11</b> Diagramme d'activité « Ajouter un personnel médical ». ....	47
<b>Figure II-12</b> Diagramme d'activité « Créer compte ». ....	48
<b>Figure III-1</b> Logo de Android studio. ....	50
<b>Figure III-2</b> Emplacement du SDK. ....	51
<b>Figure III-3</b> Emplacement de flutter&Dart.....	51
<b>Figure III-4</b> Intégration SDK Dart sur android studio. ....	52
<b>Figure III-5</b> Intégration SDK flutter sur android studio. ....	52
<b>Figure III-6</b> Ajouter projet dans Firebase. ....	53
<b>Figure III-7</b> Les données de l'application sur Firebase.....	53
<b>Figure III-8</b> Téléchargement de JSON. ....	54
<b>Figure III-9</b> Fichier JSON. ....	54
<b>Figure III-10</b> Initialisation Firebase.....	55
<b>Figure III-11</b> Les packages.....	55
<b>Figure III-12</b> Logo. <b>Figure III-13</b> L'application. ....	57
<b>Figure III-14</b> Page de démarrage.....	57
<b>Figure III-15</b> Page Connexion.....	58
<b>Figure III-16</b> Page de choisir le type d'utilisateur.....	59
<b>Figure III-17</b> Page de créer compte -Patient-. ....	59
<b>Figure III-18</b> Page de créer compte -personnel médicale-. ....	60
<b>Figure III-19</b> Page de notification. ....	61

<b>Figure III-20</b> Page d'accueil. ....	61
<b>Figure III-21</b> Page de dossiers. ....	62
<b>Figure III-22</b> L'endroit de téléchargement.....	63
<b>Figure III-23</b> Page de profil. ....	64
<b>Figure III-24</b> Page de notification -personnel médicale-.....	65
<b>Figure III-25</b> Page d'événement. <b>Figure III-26</b> Page des notes de consultation. ....	66
<b>Figure III-27</b> Page de dossiers - personnel médicale - . ....	67
<b>Figure III-28</b> Page de profil. ....	67
<b>Figure III-29</b> Page de rechercher. ....	68
<b>Figure III-30</b> Page de prendre un rendez-vous. ....	69
<b>Figure III-31</b> Interface de menu. <b>Figure III-32</b> Page de contactez-nous.....	70
<b>Figure III-33</b> Page de à-propos. <b>Figure III-34</b> Page d'aide.....	70
<b>Figure III-35</b> Interface vue d'ensemble.....	71
<b>Figure III-36</b> Interface d'authentification. ....	72
<b>Figure III-37</b> Interface de Firestore Database.....	72
<b>Figure III-38</b> Interface de cloud Storage. ....	73

# Liste des tableaux

<b>Tableau II-1</b> Tableau de spécification des besoins.....	35
<b>Tableau II-2</b> Authentifier.....	37
<b>Tableau II-3</b> Consulter dossier médical.....	38
<b>Tableau II-4</b> Ajouter personne médical.....	39
<b>Tableau II-5</b> Ajouter un dossier médical.....	40
<b>Tableau IV-1</b> Tableau de BMC.....	82