

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Mohamed el-Bachir el-Ibrahimi Bordj Bou Arréridj
Faculté des Mathématiques et L'informatiques



MEMOIRE

Présente en vue de l'obtention du diplôme

Master en informatique

Spécialité : Technologie de l'Information et de Communication

THEME :
Application mobile pour le suivi des patients

Présenté Par :

– TIBOURTINE Serine Selma

Devant le jury composé de :

Président : Hammoudi Sara – MCB Université de Bordj Bou Arréridj

Examineur : Sabri Lyazid - MCA Université de Bordj Bou Arréridj

Encadreur : Mr Moussaoui Boubakeur - MAA Université de Bordj Bou Arréridj

Année Universitaire 2020-2021

Remerciement

Je remercie Allah le tout puissant de m'avoir guidée vers le droit chemin, de m'avoir aidée tout au long de mes années d'étude.

Je m'adresse mon profond remerciement à Monsieur « MOUSSAOUI Boubaker » mon encadreur et les professeurs qui m'ont aidée et pour les encouragements.

Je tiens a remercié aussi, tout l'équipe de l'enseignement de Licence en Ingénierie des Systèmes de l'Information et de Logiciels.

Ainsi, je m'adresse mes remerciements le plus chaleureux à toutes les personnes qui ont aidée de près ou de loin par le fruit de leur connaissance pendant toute la durée de mon parcours éducatif.

Dédicace

Je dédie ce modeste travail à :

Vous ma mère et mon père. Aucun hommage ne pourrait exprimer l'amour, l'estime et le respect que j'ai toujours eu pour vous.

Merci à mes deux sœurs Norine et Alla.

Dans ce cadre je n'oublie pas le soutiens de mes chères tantes Chahra et Aicha, mes grands-parents que dieu pose leurs à mes aux paradis et mon entière famille.

Mes meilleures amies merci d'être à mes cotes de m'avoir encouragé et de croire en moi pendant toutes ces années de notre grande amitié.

Merci à Maroua une merveilleuse amie que j'ai rencontré qui ma soutenue, m'encourager et m'aidée avec tous ce qu'elle a.

Merci à mes professeurs qui m'ont aidée avec des merveilleuses idées pour réaliser ce projet.

Merci à tous ceux qui me sont chères, a tous ceux qui m'aiment.
Merci à tous ceux dont je n'ai pas pu citer leur nom sachez que vous êtes toujours présent dans mon cœur.

Merci à mon encadreur Mr MOUSSAOUI Boubaker et à tous les professeurs de la faculté ayant encadré de prêt ou de loin mon travail.

-Tibourtine Serine Selma-

Table des matières

<i>Remerciement</i>	2
<i>Dédicace</i>	3
<i>Introduction générale</i>	7
Chapitre I : Les applications mobiles	8
1.1.Introduction	8
1.2. Les applications mobiles	8
1.2.1.Définition	8
1.3. Le marché des Smartphones	8
1.4.Historique	9
1.5.Les types d'applications mobiles	11
1.5.1.Les applications web (connectées)	11
1.5.2. Les applications natives (non connectées).....	11
1.5.3. Les applications hybrides	12
1.6.Domaine d'application	13
1.7 Cycle de vie d'une application Flutter	14
1.8 Conclusion	15
Chapitre II : Conception	16
2.1.Introduction	16
2.2.Objectif attendu de l'application	16
2.3. Présentation de l'application	16
2.4 définition du dossier médical	17
2.5. Définition des besoins	17
2.6. Identification des acteurs	18
2.7.Conception	18
2.7.1.diagramme de cas d'utilisation	19
2.7.1.2.Inscription de l'utilisateur	21
2.7.1.3.Identification	21
2.7.1.4.Ajouter une maladie	21
2.7.2.Diagramme de Séquences	22
2.7.3 Diagrammes de Classe	24
2.8. Conclusion	26
Chapitre III : Réalisation	27
3.1.Introduction	27
3.2. Systèmes d'exploitation mobiles	27
3.2.1 iOS	27
3.2.2 Windows Phone	27
3.2.3 BlackBerry OS	28
3.2.4 Android	28
3.3.Environment de travail	28
3.3.1 Environnement Matériel	28
3.3.2 Environnement Logiciel	29

3.4.1 Préparation de l'environnement de travail	32
3.5 Écriture du code.....	32
3.6 Commentaire du code	34
3.7 Fonction de test	35
3.8 Préparation la base sur Firebase.....	35
3.9 Lier Firebase a Flutter	36
3.10 Teste de l'application.....	38
3.11 Conclusion	43
<i>Conclusion générale</i>	44
<i>Webographie</i>	45

Liste des figures

Figure 1 : l'utilité de Flutter	10
Figure 2:Les types d'applications mobile.....	13
Figure 3: StatefulWidget et WidgetsBindingObserver.....	15
Figure 4 :Diagramme de cas d'utilisation général	20
Figure 5:Diagramme de séquence « Inscription de l'Utilisateur»	22
Figure 6:Diagramme de séquence « Identification ».....	23
Figure 7:Diagramme de séquence «Ajouter maladies »	24
Figure 8:Diagrammes de Classe Générale	25
Figure 9: home-screen.dart	34
Figure 10: authentification avec Firebase	36
Figure 11:authentification.dart.....	37
Figure 12: Welcome Page	38
Figure 13: Login Page	39
Figure 14: Sign Up Page	40
Figure 15:On boarding Page	41
Figure 16:Home Page.....	42
Figure 17:Doctor Page	43

Liste des tableaux

Tableau 3-1: Caractéristique du matériel utilisé.....	28
---	----

Introduction générale

Actuellement, le monde connaît un avancé technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information, elle joue un rôle important dans le quotidien de l'être humain.

La démarche médicale est fondée sur l'observation du malade. La mémoire du médecin était autrefois suffisante pour enregistrer les données relatives aux patients et servir l'exercice médical.

Les données médicales étaient rassemblées sous forme d'articles médicaux, de registres à visée épidémiologique, nosologique et administrative, avec la multiplication des effets de l'environnement, de nos jours la bonne tenue d'un dossier exige des moyens informatiques L'automatisation du système d'information consiste à structurer et gérer un ensemble de données dont le but de les organiser et d'avoir des résultats rapides.

La crise sanitaire qu'on a vécu ces deux dernières années à cause de la pandémie COVID-19, nous a obligé de penser qu'il faut profiter de toutes les technologies et outils afin d'assurer des services à distance, surtout lorsque le problème est lié à la santé des gens.

Le suivi des patients à distance est devenu de plus en plus un défi pour protéger la vie des gens et de faciliter que possible la période du traitement d'un patient, justement notre travail s'articule dans ce cadre on a pensé à développer une application mobile qui fournit un suivi des dossiers médicaux d'un patient par son (ses) médecin(s).

On vous présente dans le premier chapitre le contexte de notre travail.

Et Le deuxième chapitre on termine par une conclusion générale porte sur l'étude conceptuelle de notre application suivant le processus UP d'UML (Unified Modeling Language).

Enfin, le troisième chapitre qui décrit le projet avec la présentation des choix techniques de réalisation. Ce chapitre est consacré à la présentation de l'environnement matériel et logiciel utilisé pour cette application.

Chapitre I : Les applications mobiles

1.1. Introduction

L'automatisation des systèmes nécessite l'introduction des applications informatiques afin de rendre le système plus performant et efficace, plusieurs types d'application sont disponibles (desktop, web, mobile) on s'intéresse ici aux applications mobiles, dans ce qui suit on présente un aperçu plus ou moins approfondi sur les applications mobiles ainsi qu'au Framework Flutter.

1.2. Les applications mobiles

1.2.1. Définition

Une application mobile, est un type de logiciel d'application conçu pour fonctionner sur un appareil mobile, tel qu'un smartphone ou une tablette informatique. Les applications mobiles servent souvent à fournir aux utilisateurs des services similaires à ceux auxquels ils accèdent sur un ordinateur ainsi ils sont généralement de petites unités logicielles individuelles aux fonctions limitées. Cette utilisation des logiciels d'application a été popularisée à l'origine par Apple et son App Store qui propose des milliers d'applications pour l'iPhone, l'iPad et l'iPod Touch. [1]

1.3. Le marché des Smartphones

Le premier trimestre 2021 marque une reprise du marché des smartphones avec une augmentation de 27% sur le total des ventes par rapport à l'année précédente. Après une année 2020 a marquée par la pandémie de coronavirus, l'année 2021 commence sous les meilleurs auspices.

Le coronavirus a bouleversé le monde de travail dès le début 2020 avec les premières difficultés au niveau des chaînes d'approvisionnement chinoises lorsque les premiers confinements ont commencé même si la crise sanitaire n'est pas finie les restrictions et confinements se sont globalement assouplis dans le monde.

Les marques de smartphones qui ont vendu les plus d'unités ce trimestre :

Samsung est encore numéro 1er ce trimestre avec 76,5 millions d'appareils vendus et 22% de parts de marché. La sortie récente des Galaxie S21 a favorisé le constructeur sud coréen.

Apple est 2eme avec 52,4 millions d'iPhone écoulés correspondant à 15% des parts de marché du secteur. La firme à la pomme profite du succès des iPhone 12, les premiers modèles compatibles 5G.

Xiaomi consolide sa troisième place avec 49 millions de smartphones vendus soit 14% de parts de marché.

Et les deux constructeurs chinois Oppo et Vivo ont marqué respectivement 37,6 et 36 millions d'appareils vendus. Huawei a réalisé 7ème avec 18,6 millions d'unités vendues. Les sanctions américaines semblent finalement avoir eu raison du géant chinois qui se retrouve relégué en queue de peloton. [2]

1.4. Historique

Les applications mobiles sont apparues dans les années 1990 elles sont liées aux développements d'Internet et des télécommunications des réseaux sans fils et des technologies agents et à l'apparition et la démocratisation des terminaux mobiles : smartphones, tablettes tactiles... Le nombre d'applications mobiles disponible augmente rapidement à partir de 2007 date du lancement de l'iPhone. Les deux principaux services de plateformes de téléchargement l'App Store et le Google Play sont lancées en 2008[3]

À propos de Flutter sa première version était mai 2017 mais à partir décembre 2018 il est en version stable. Il est développé en C, C++, Dart et Skia Graphics Engine.

Flutter se compose principalement de la plateforme Dart, du moteur Flutter, de la fondation library et des différents widgets.

Le langage DART est un langage de programmation web son but est de se substituer au langage JavaScript et de devenir la référence pour le développement multiplateforme.

Depuis mai 2019 jusqu'à maintenant est de développement pour IOS et Android, Flutter permet de créer des applications web et desktop.[4]

La figure 1 présente l'utilité de l'outil Flutter dans le domaine de développement des applications entre 2019 et 2020

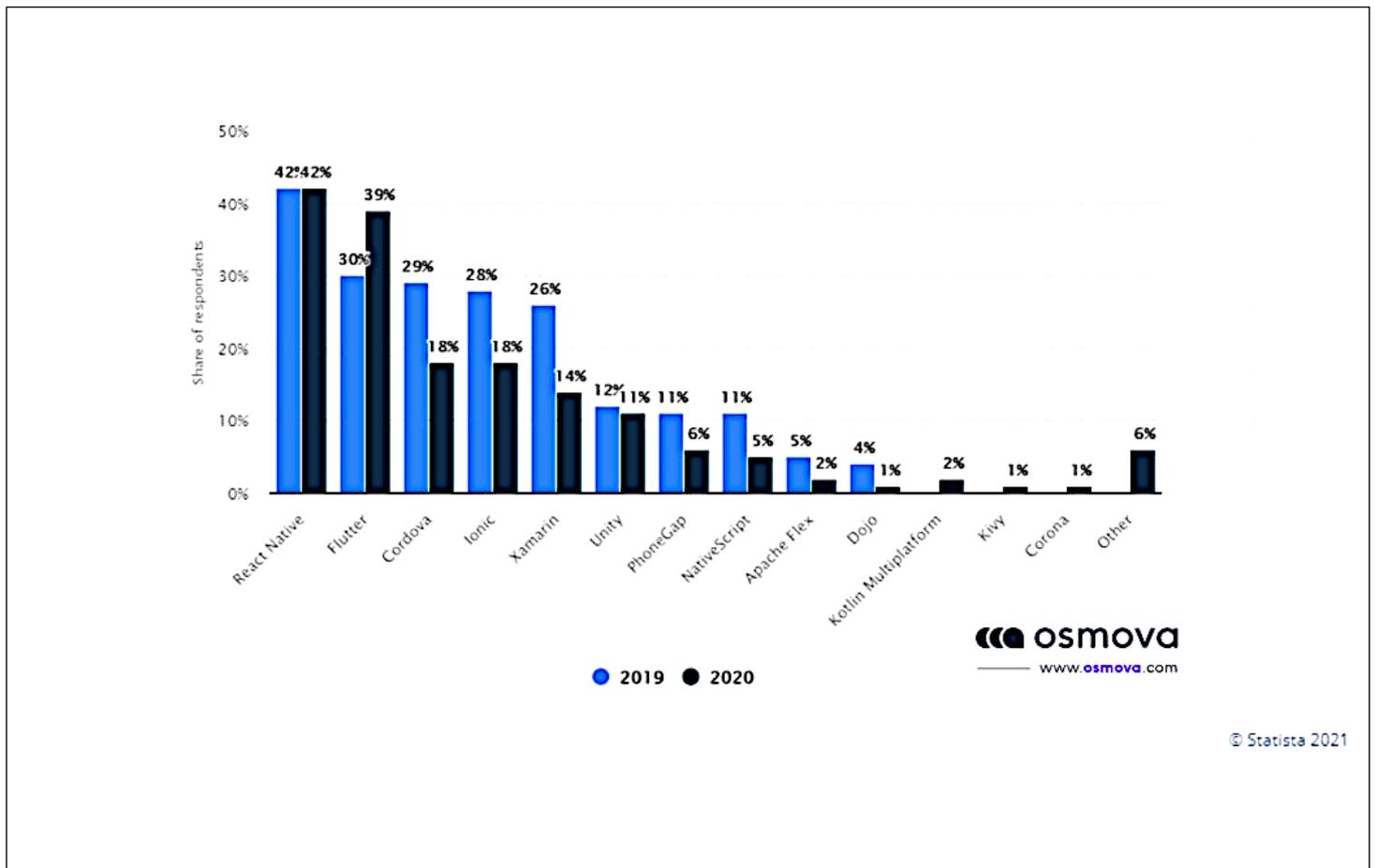


Figure 1 : l'utilité de Flutter

1.5. Les types d'applications mobiles

Selon leurs modes de fonctionnement les applications mobiles se répartissent en trois catégories : les applications web (connectées), les applications natives (non connectée) et les applications hybrides.

1.5.1. Les applications web (connectées)

L'application Web est un logiciel applicatif hébergé sur un serveur et accessible à travers un navigateur web. Cependant contrairement à un logiciel traditionnel les utilisateurs n'ont pas la nécessité d'installer le logiciel. La connexion avec l'application se fait via un navigateur internet comme si c'était un site web.

Les avancées technologiques ont permis l'apparition des progressive Web App (PWA). Une Progressive Web App est une Web App ayant intégré des fonctionnalités grâce à des « Service Worker » qui était jusqu'à présent réservé aux applications natives.

Les avantages d'une PWA sont divers. D'une part elle permet d'avoir des frais de développement moins élevés qu'une application native ou hybride car les langages de programmation sont plus courants (HTML, JAVA, PHP, REACT, Angular ...). Son développement repose sur la segmentation et l'association des briques d'une application au travers d'API cette conception permet de prendre le meilleur de chaque technologie utilisée dans l'architecture métier afin d'obtenir un parfait compromis entre une application Native/mobile et un site web.

D'autre part son accessibilité depuis un navigateur lui donne également l'avantage d'être cross-plateforme c'est-à-dire accessible depuis n'importe quel device : Desktop, tablette et mobile. Les ressentis avancés permettent aux PWA d'utiliser des fonctionnalités natives comme l'utilisation Offline, GPS à la demande, géolocalisation, notifications push et la gestion du cache, appareil photo, le micro, la boussole.

1.5.2. Les applications natives (non connectées)

Une application native est une application mobile qui fait l'objet de développements distincts pour chacun des systèmes d'exploitation utilisés (iOS, Android, etc.), et ainsi elle fonctionne de façon optimale avec tous les types d'appareils. Elle a un développement plus

complexe que les autres car les langages de programmation sont différents selon les systèmes d'exploitation. Par exemple, les applications pour l'Apple Store sont développées soit en Objective-C ou en Swift mais les applications pour Google Play sont développées en Java et Kotlin. Les applications Windows sont développées en C#.

La différence est significative car ces divers langages feront appel à ce que l'on appelle les couches de bases de l'appareil comme l'appareil photo ou encore l'accéléromètre.

1.5.3. Les applications hybrides

Entre l'application native et l'application web on trouve l'application « cross-platform » Hybride. Comparées aux applications natives les applications hybrides sont moins chères et plus rapides à développer.

Une application Hybride a une conception plus simple et rapide que les applications natives car seulement une version du code est déployée pour les différents systèmes d'exploitation (Apple, Android, Windows). Lors du développement de l'application un Software Développement Kit (SDK) est utilisé pour avoir accès à une librairie d'ensemble de briques préconstruites adaptées aux différents systèmes d'exploitation et permettant la transposition en code binaire (langage natif).

Les développements en « cross-platform » ont l'avantage de réduire la maintenance et de faciliter l'ajout de fonctionnalités car il n'y a qu'une seule version du code.

Dans les applications « cross-platform » on trouve les applications « hybrides » et les applications « natives générées ».

Une application « natives générées » est une application développée à partir d'un langage « cross-platform » type C#, Ruby, JavaScript, React Native les SDK les plus utilisés sont Xamarin et Titanium.

Une application « hybride » est une application dont le contenu web est alors « encapsulé » avec une interface mobile on y retrouve à la différence de la « native générée » du code HTML. Pour ce faire, des SDK telles qu'Apache Cordova (le plus courant), PhoneGap, Rho Mobile sont utilisés.^[5]

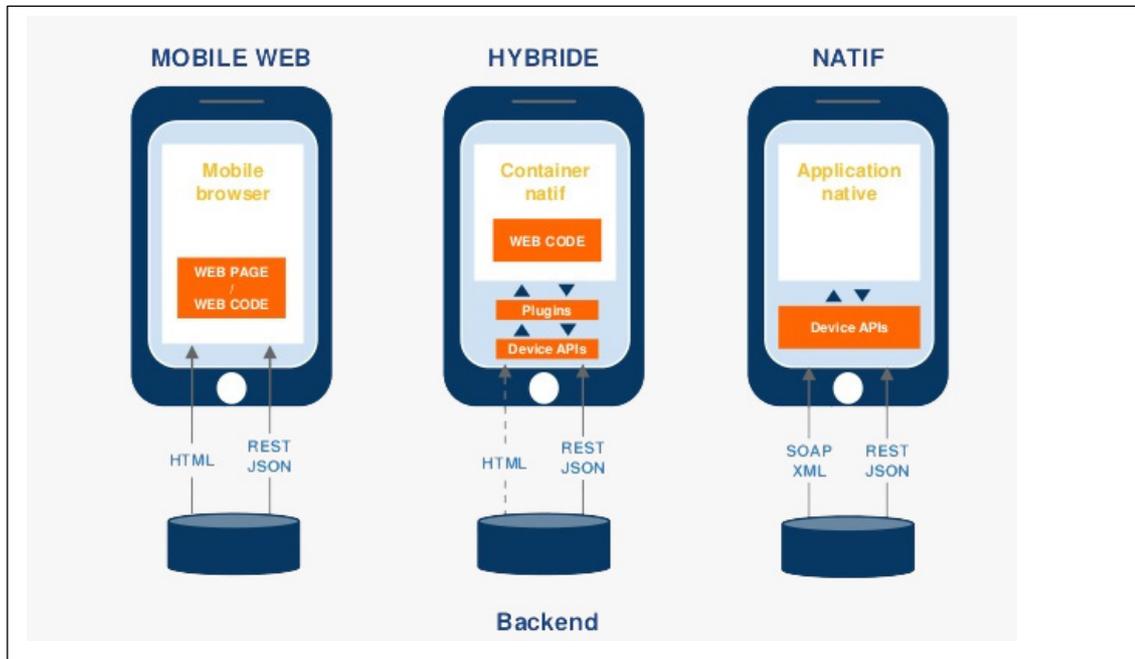


Figure 2: Les types d'applications mobile

1.6. Domaine d'application

Avec les possibilités matérielles incorporées aux terminaux (caméras, GPS, gyroscope, ...), les applications Smartphones, Tablettes et les télévisions peuvent intégrer des fonctionnalités spécifiques et dédiées pour les utilisateurs permettant ainsi d'enrichir des usages non couverts jusqu'à présent par les systèmes d'information. Le succès de Flutter est indéniable les fournisseurs d'où bien compris d'où l'apparition sur le marché utilisant ce system d'exploitation :

- Géolocalisation, Itinéraires
- Scan de Code barre, Flash, QR Code.
- Réalité augmentée.
- M-commerce, Paiement mobile.
- Push et notification.
- Gestion de documents, dématérialisation, Workflow.
- Analyse d'Audience.
- Gestion et Sécurisation de parc et de déploiement de terminaux mobiles. [6]

1.7 Cycle de vie d'une application Flutter

Comme pour tous les Framework, Flutter a également un cycle de vie associé à chaque application. Dans cet article, nous examinerons les différentes méthodes de cycle de vie des applications disponibles dans Flutter.

Voici les états du cycle de vie de l'application dans Flutter tels que définis par `AppLifecycleState` :

- **Détaché** : l'application est toujours hébergée sur un moteur de flutter mais est détachée de toutes les vues d'hôte. Cela peut se produire lorsque le moteur a démarré mais n'est attaché à aucune vue ou lorsque la vue est détruite en raison du `pop Navigator`.
- **Inactive** : l'application est dans un état inactif et ne reçoit aucune entrée utilisateur. L'application entre dans cet état lors d'un appel téléphonique sur les deux plates-formes. Les applications dans cet état doivent supposer qu'elles peuvent être mises en pause à tout moment.
- **En pause** : l'application n'est actuellement pas visible par l'utilisateur et s'exécute en arrière-plan. C'est à ce moment que vous appuyez sur le bouton Accueil.
- **Reprise** : l'application est visible et répond aux entrées de l'utilisateur. Dans cet état, l'application est au premier plan.

Mise en œuvre

Afin d'observer les différents états du cycle de vie à partir de la couche widgets, nous utilisons la classe `WidgetsBindingObserver` (Figure 3).

Créez un `StatefulWidget` et `WidgetsBindingObserver` comme :

```

1  class MyHomePage extends StatefulWidget {
2    MyHomePage({Key key, this.title}) : super(key: key);
3
4    final String title;
5
6    @override
7    _MyHomePageState createState() => _MyHomePageState();
8  }
9
10 class _MyHomePageState extends State<MyHomePage> with WidgetsB
11
12     // Code
13
14 }

```

Figure 3: StatefulWidget et WidgetsBindingObserver

1.8 Conclusion

Dans ce chapitre nous avons présenté ce que on appelle Flutter, le système dans sa jeunesse. De ce fait nous avons conclu que cette Plateforme est un système d'exploitation riche et complet. Il s'est développé tellement vite en quelques années qu'il est devenu l'OS le plus utilisé pour le smartphone dans le monde avec plus d'un milliard d'utilisateurs. Dans le prochain chapitre, Nous allons passer à la modélisation de notre application puis nous détaillerons les étapes de la conception afin de d'en assurer une utilisation pratique et surtout intuitive.

Chapitre II : Conception

2.1. Introduction

Afin d'aboutir à une meilleure organisation et une bonne maîtrise du travail et donc arriver à déployer de meilleures applications, il est nécessaire de suivre une démarche méthodologique rigoureuse. Pour cela le choix d'une méthode d'analyse et de conception est d'une très grande importance.

Le langage UML (Unified Modeling Language) est un langage de modélisation objet très utilisée et qui a prouvé son efficacité dans le domaine de modélisation des systèmes pour cela on a choisi ce langage pour notre étude.

2.2. Objectif attendu de l'application

Les applications mobiles sont devenues les canaux les plus stratégiques d'accès aux clients, on a pensé à développer une application qui touche à un domaine très sensible à la santé des gens.

L'objectif principal de l'application est d'offrir aux utilisateurs un suivi quotidien de leurs états de santé et ce n'importe où et à n'importe quelle heure. Disposant d'une liste de maladies avec les traitements correspondant afin que l'utilisateur puisse enregistrer son état et surveille ses progrès au fil de sa vie (garder trace de toutes les maladies atteintes et leurs médicaments). A l'image de la plupart des applications proposées désormais, l'application que nous cherchons à développer disposera d'un suivi médical de la part d'un médecin spécialiste pour prescrire une ordonnance et un suivi d'un régime alimentaire à partir d'état d'un patient si nécessaire.

2.3. Présentation de l'application

Le suivi d'un patient est une activité principale du médecin vers le patient ce dernier lui ramené la fiche médicale descriptive ainsi que son dossier médical, l'écoute attentive et patiente des propos du patient est un moment privilégié de la consultation.

Le médecin analyse le dossier (passeport médical) pour rédiger ensuite l'ordonnance ou décrit un régime approprié.

Dans le cas où le médecin n'est pas sûr de son diagnostic il peut demander au patient de prendre un rendez-vous et faire la consultation dans une manière traditionnelle.

2.4 définitions du dossier médical

C'est un outil indispensable pour la pratique médicale. Il est intrinsèquement lié au processus de prise en charge du patient. Il contient des données d'origine et de nature différentes (administratives, médicales, paramédicales, sociales, etc.) qui sont générées, inférées, recueillies et notées par autant de type d'acteurs impliqués dans la prise en charge. Ces données sont autant de faits utiles correspondant aux diverses décisions et actions que nécessite l'état du patient.

Le dossier médical est un document médico-légal justifiant la consultation et l'attitude thérapeutique qui en découle, il doit être soigneusement gardé par le médecin (l'obligation au secret médical).

Ce dossier doit être archivé et gardé aussi longtemps que possible car un acte médical peut être remis en cause.

En cas de visite de contrôle ou visite périodique d'un patient en maladie de longue durée, il sera programmé à un moment précis de la journée.

A chaque consultation un résumé de la nouvelle consultation et du traitement donné sera porté sur le dossier médical.

Le Dossier Médical Personnel Informatisé (DMPI) est tout simplement la version électronique du dossier papier du patient, pouvant être consulté par les professionnels de santé, afin d'effectuer des opérations bien définies nécessaires à la prise en charge du patient.

Dans le DMPI, le secret médical du patient est assuré par des moyens de sécurité électronique.

2.5. Définition des besoins

On va identifier les besoins de l'utilisation du patient de l'application médicale

- Un Patient peut s'inscrire.

- Un patient peut envoyer un dossier.
- Un patient peut connecter un médecin.
- Un patient peut prendre un rendez-vous.

2.6. Identification des acteurs

Patient : Personne qui subit un traitement diagnostique et / ou la thérapie. Dans le domaine de la médecine un patient est une personne recevant une attention médicale ou a qui est prodigué un soin et bénéficie d'examens médicaux de traitements prodigués par un médecin ou autres professionnels de la santé pour faire face à une maladie ou à des blessures

Médecin : Personne qui diagnostique et en traitant les maladies, un médecin est un professionnel de la santé titulaire d'un diplôme de docteur en médecine, il soigne les maladies, pathologies et blessures, il travaille généralement au sein d'une équipe de professionnels de la santé comme le psychologue, le pharmacien, l'infirmière ou le chirurgien-dentiste.

2.7. Conception

La conception est la dernière étape avant la réalisation technique de la plateforme de l'application elle consiste donc à réaliser le modèle qui va être implémenté l'UML (Unified Modeling Language) qui est un langage de modélisation visuelle il est destiné à l'architecture conception et la mise en œuvre de systèmes logiciels complexes par leur structure L'UML est un outil utilisé pour générer du code en plusieurs langages structurés des diagrammes et l'analyse et la conception orienté objet L'UML utilise de éléments et les associe de différentes manières pour former des diagrammes qui sont représenter sous deux types de vue

Diagrammes UML structurels :

- Diagramme de classes
- Diagramme de composants
- Diagramme de structure composite
- Diagramme de déploiement
- Diagramme d'objets
- Diagramme de paquetages

Diagrammes UML comportementaux :

- a) Diagrammes d'activités
- b) Diagramme de communication
- c) Diagramme global d'interaction
- d) Diagramme de séquence
- e) Diagramme états-transitions
- f) Diagramme de temps
- g) Diagramme de cas d'utilisation

Dans notre cas on va élaborer des diagrammes de séquence ainsi que les diagrammes de classes pour les cas d'utilisation étudié et un diagramme de classe générale.

2.7.1. Diagramme de cas d'utilisation

Un cas d'utilisation est une unité cohérente d'une fonctionnalité visible de l'extérieur il réalise un service de bout en bout avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système sans imposer le mode de réalisation de ce service elle permet aussi de représenter l'ensemble des besoins et des exigences que notre application doit respecter.

2.7.1.1. Diagramme de cas d'utilisation général

Un diagramme de cas d'utilisation général capture le comportement d'un système, d'un sous- système, tel qu'un utilisateur extérieur le voit les cas d'utilisation ayant un sens pour les acteurs, les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

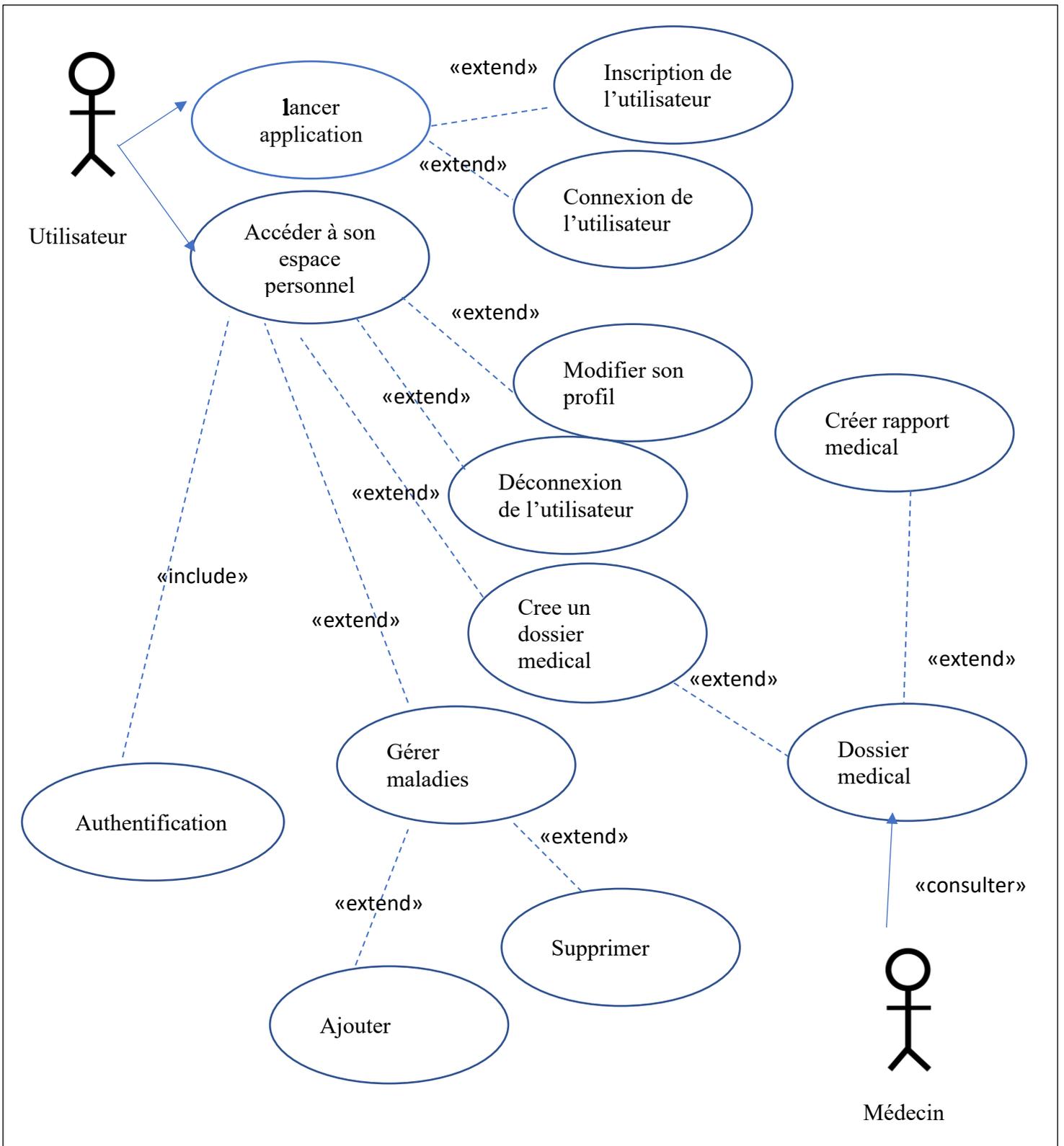


Figure 4 : Diagramme de cas d'utilisation général

2.7.1.2. Inscription de l'utilisateur :

Cas d'utilisation : inscription de l'utilisateur.

Acteurs : Utilisateur

Description :

- 1- L'utilisateur lance l'application en cliquant sur l'icône.
- 2- Le système affiche l'activité principale (l'accueil).
- 3- L'utilisateur clique sur le bouton « inscription » si n'a pas déjà un compte sinon.
- 4- Le système affiche l'activité contenant le formulaire d'inscription.
- 5- L'utilisateur remplit les champs du formulaire d'inscription et clique sur le bouton « valider ».
- 6- Le système lui affiche une autre activité lui proposant de s'authentifier pour accéder à son espace.

2.7.1.3. Identification :

Acteurs : Utilisateur

cas d'utilisation : identification de l'utilisateur. Description :

- 1- L'utilisateur atteint l'activité principale de l'application ensuite il clique sur le bouton « connexion ».
- 2- Le système répond en affichant le formulaire d'authentification.
- 3- L'utilisateur remplit le formulaire en saisissant son login et son mot de passe et

Clique sur le bouton « connexion ».

2.7.1.4. Ajouter une maladie :

Acteurs : Utilisateur

cas d'utilisation : Ajouter une maladie

Description :

- 1- Après authentification l'utilisateur atteint son espace personnel.
- 2- l'utilisateur clique sur le bouton « maladie ».
- 3- Le système lui affiche l'activité maladie qui contient le bouton « ajouter ».
- 4- l'utilisateur appuis sur le bouton « ajouter ».
- 5- Le système lui affiche l'activité correspondante, celle qui contient un formulaire d'ajout.
- 6- l'utilisateur remplit les champs et clique sur « valider ».

2.7.2. Diagramme de Séquences

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre.

2.7.2.1. Diagramme de séquence « Inscription de l'Utilisateur »

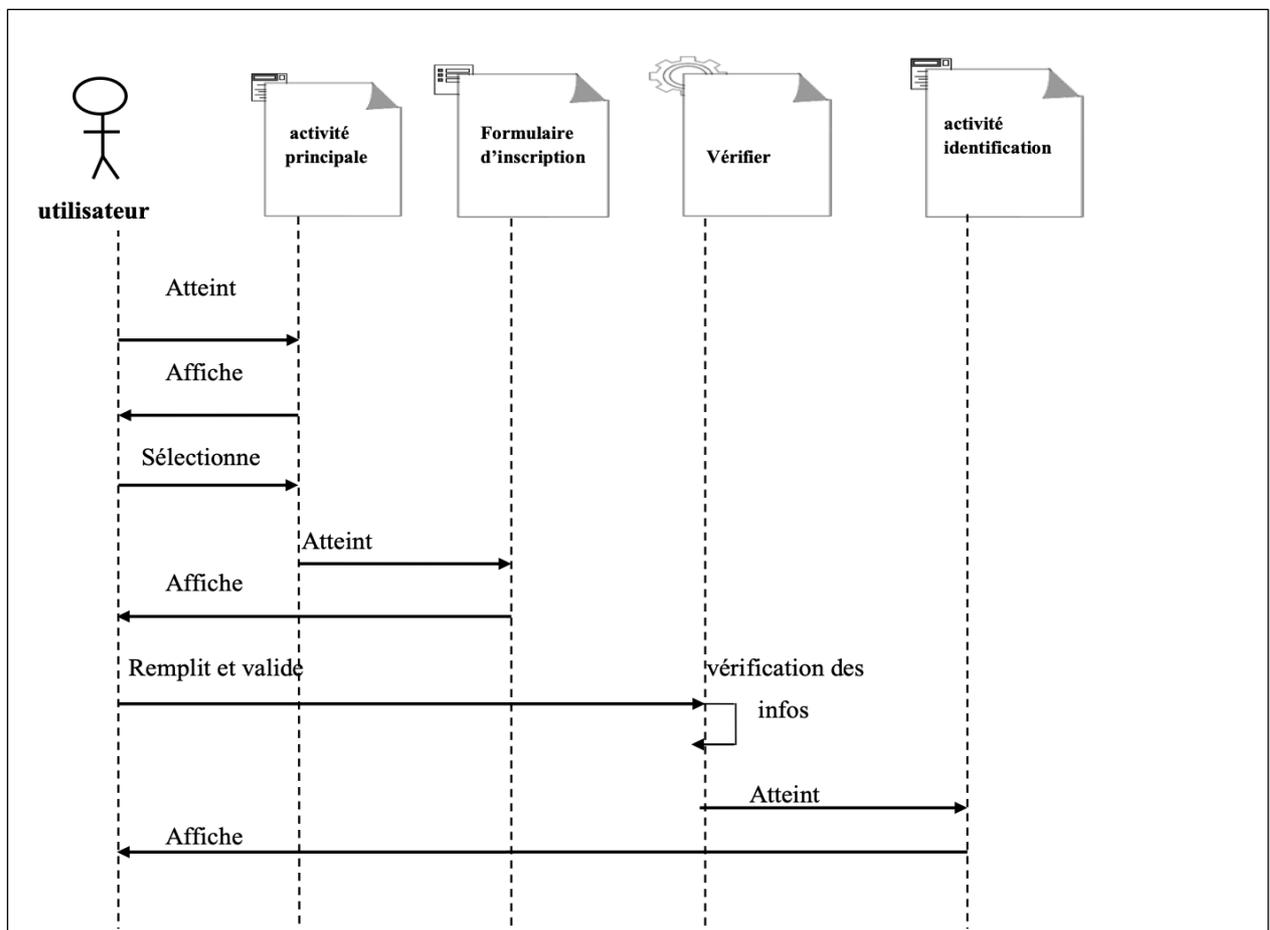


Figure 5:Diagramme de séquence « Inscription de l'Utilisateur »

2.7.2.2. Diagramme de séquence « Identification ».

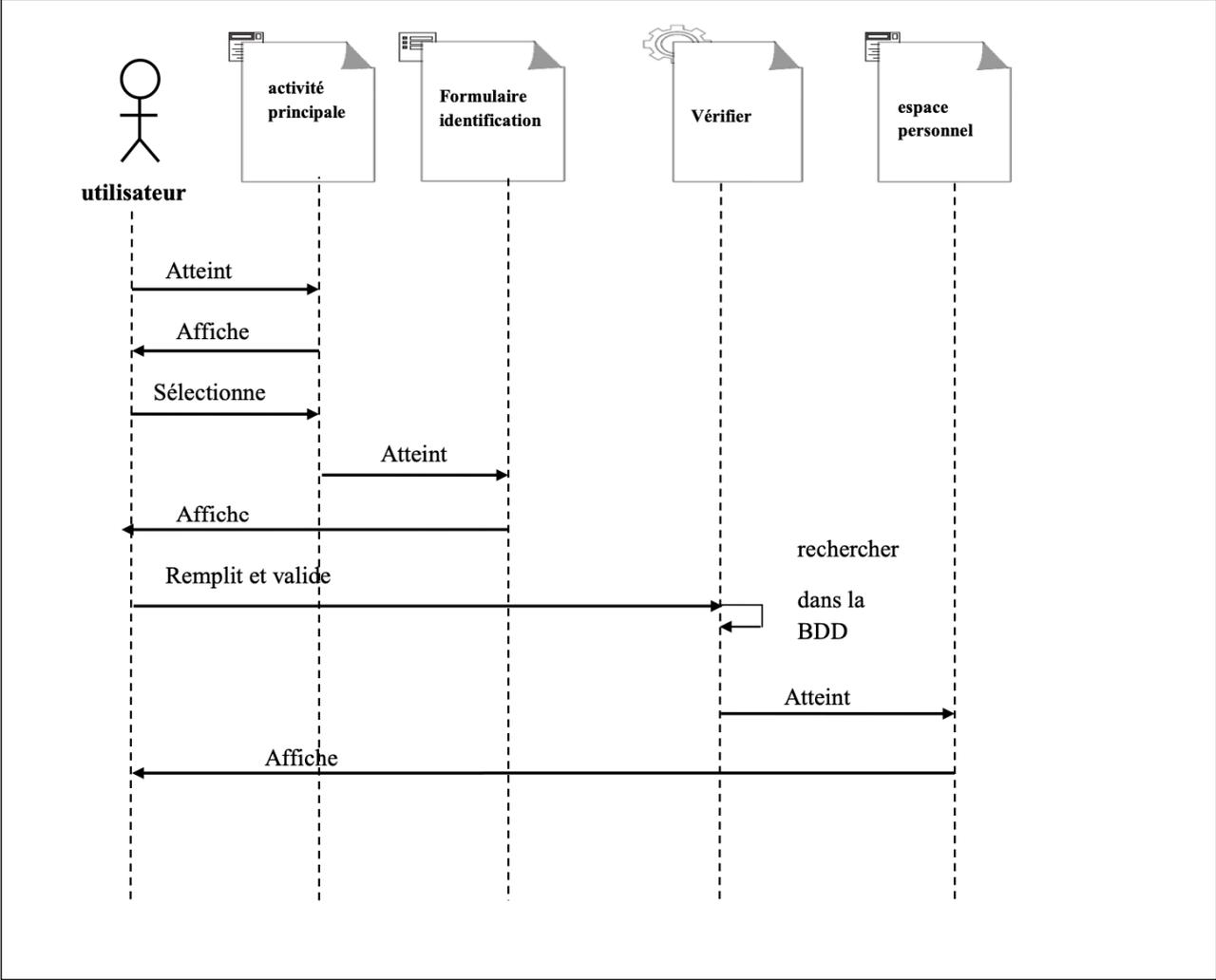


Figure 6:Diagramme de séquence « Identification »

2.7.2.3. Diagramme de séquence « Ajouter maladies ».

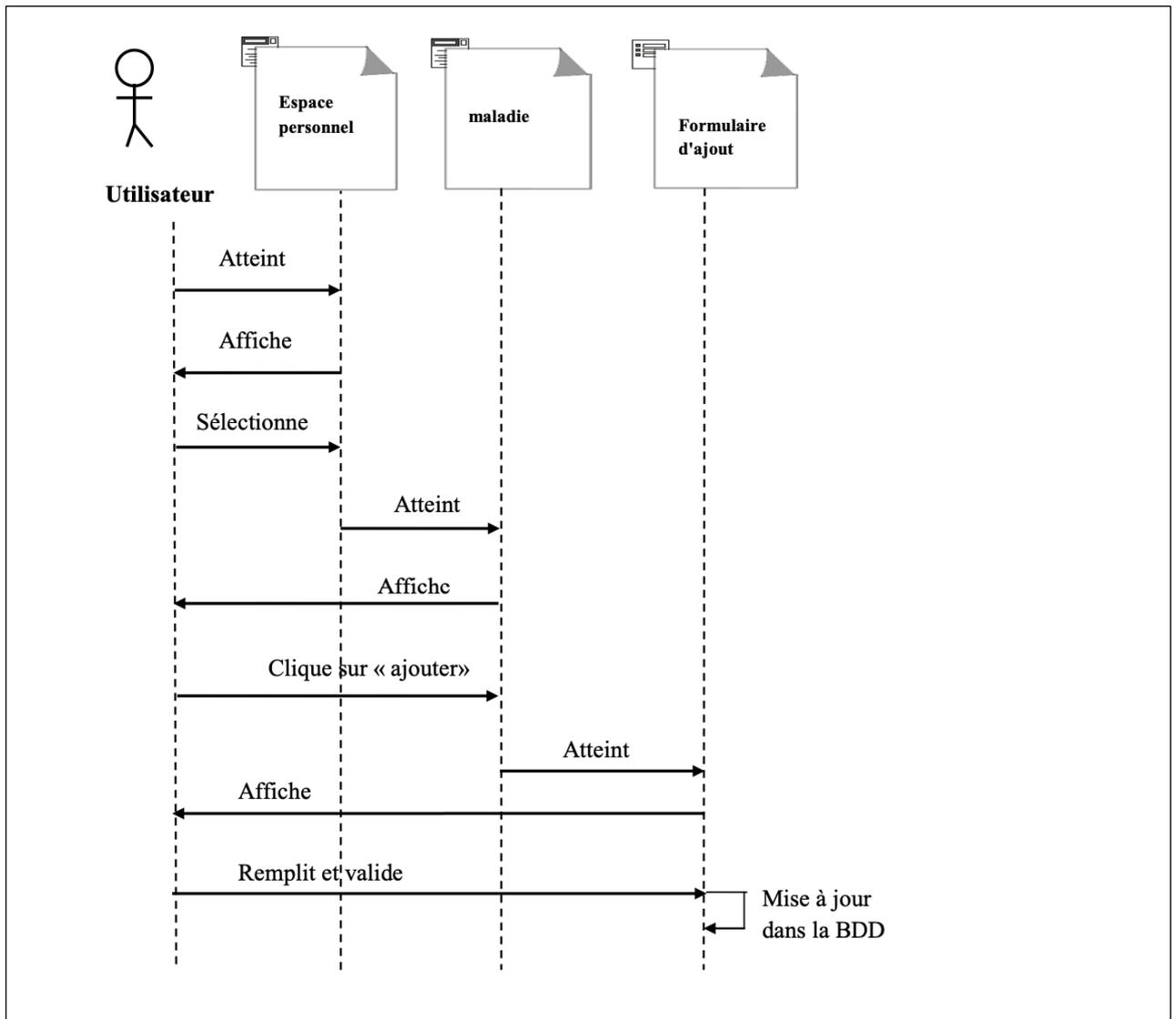


Figure 7:Diagramme de séquence « Ajouter maladies »

2.7.3 Diagrammes de Classe

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Il s'agit donc d'une vue purement statique car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Il permet aussi de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier

2.7.3.1. Diagrammes de Classe Générale

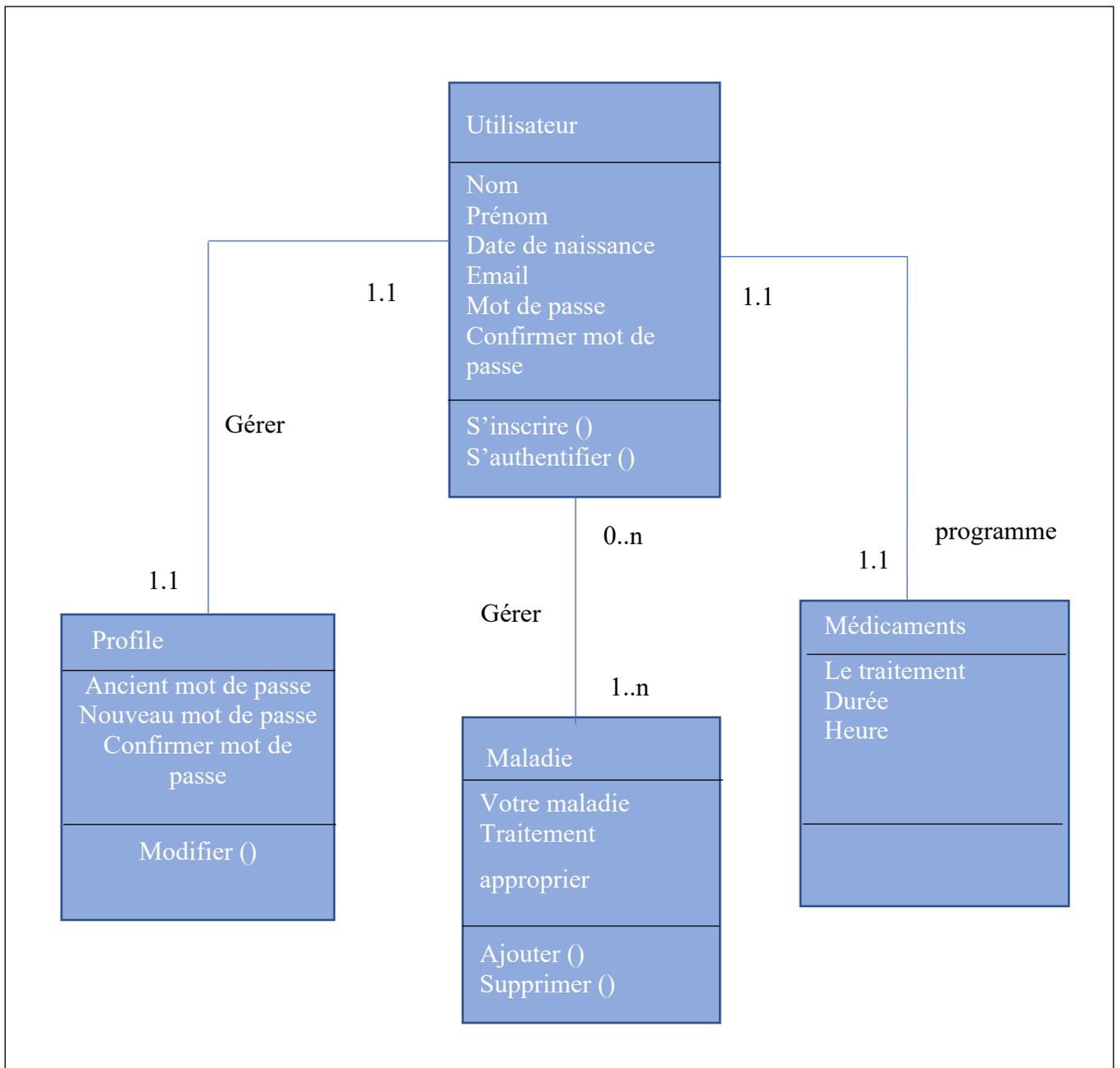


Figure 8:Diagrammes de Classe Générale

2.8. Conclusion

Dans ce chapitre, nous avons proposé une démarche de modélisation pour développer notre application. Nous avons d'abord commencé par la spécification du cas d'utilisation dans un premier temps suivi avec une étude de conception de notre application. La réalisation du projet sera détaillée dans le prochain chapitre.

Chapitre III : Réalisation

3.1. Introduction

Une fois le travail de modélisation est achevé, on arrive à l'étape du codage (Implémentation ou programmation) qui consiste en la traduction dans un langage de programmation des différentes fonctionnalités définies lors des phases d'analyse et de conception. Et afin d'avoir en résultat une application qui réalise correctement toutes les tâches attendues par ses utilisateurs il faut bien choisir les logiciels ainsi que les langages adaptés pour la mise en place de l'application.

3.2. Systèmes d'exploitation mobiles

Tout comme un ordinateur dispose d'un système d'exploitation, les téléphones mobiles se composent également d'une plateforme qui contrôle toutes leurs fonctionnalités. Ceci est connu comme un système d'exploitation mobile. Généralement connu sous le nom d'OS (Operating System) mobile, il s'agit d'un système d'exploitation qu'exploite un appareil mobile tel qu'un Smartphone, une tablette tactile, ... Il contrôle et coordonne toutes les opérations de base du téléphone mobile comme les options d'écran tactile, Bluetooth, Wifi, appareil photo, etc. et assure la liaison entre les ressources matérielles, l'utilisateur et les applications.

3.2.1 iOS

Anciennement connu sous le nom iPhone OS, iOS est le système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod et l'iPad. Il est dérivé de OS X dont il partage les fondations iOS est le deuxième système d'exploitation le plus répandu sur le marché.

3.2.2 Windows Phone

Développé par Microsoft pour les Smartphones et Pocket PC, Windows Phone succédé à Windows Mobile en proposant en plus des applications basiques comme la messagerie électronique, Internet, Chat et MultiMedia, des fonctionnalités média sociaux tels que Facebook et Twitter.

3.2.3 BlackBerry OS

Système d'exploitation fonctionnant sur les Smartphones BlackBerry. Il permet aux développeurs de mettre en place des applications en utilisant les APIs (Application Programming Interface) BlackBerry, toute application doit être signée numériquement par le compte RIM (Research In Motion) du développeur.

3.2.4 Android

Android est un système d'exploitation open source développé par Android une startup rachetée par Google en 2007 fondé sur un noyau Linux le système a d'abord été conçu pour les Smartphones et tablettes tactiles. Puis s'est diversifié dans les objets connectés et ordinateurs comme les télévisions, les voitures, les ordinateurs et les Smart Watch. Android a été conçu pour intégrer au mieux des applications existantes de Google telles que le service de messagerie électronique Gmail, celui de la cartographie Google Maps ou encore YouTube, Google Talk et Google Calendar ... En 2015 Android est le système d'exploitation le plus utilisé dans le monde avec plus de 80% des parts de marché dans les Smartphones.

3.3. Environnement de travail

3.3.1 Environnement Matériel

Nous avons utilisé un ordinateur portable pour le développement de notre application, le tableau 3-1, présente le détail des caractéristiques du matériel utilisé.

Caractéristiques	PC
Marque	MacBook Pro
Processeur	2,7 GHz Intel Core i7 quatre cœurs
RAM	16 GO
Disque dur	925 GO
Système d'exploitation	MacOS Big Sur version 11.1

Tableau 3-1: Caractéristique du matériel utilisé

3.3.2 Environnement Logiciel

3.3.2.1 iOS

iOS, anciennement iPhone OS le « i » de iOS étant pour iPhone d'où la minuscule, est le système d'exploitation mobile développé par Apple pour plusieurs de ses appareils. Il est dérivé de MacOS il partage les fondations iOS comporte quatre couches d'abstraction, similaires à celles de MacOS : une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa ».

3.3.2.2 Flutter

Flutter est un kit de développement de logiciel (SDK) d'interface utilisateur open-source créé par Google il est utilisé pour développer des applications multiplateformes pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code. Dont la première version a été publiée sous forme de projet open source à la fin de l'année 2018. Flutter met à disposition une grande variété de bibliothèques d'éléments d'IU standard pour Android et iOS. Et permet aussi de recharger la page automatiquement sans faire l'exécution du code ce rechargement simplifie le débogage.

Les applications développées avec Flutter prennent l'aspect d'applications typiques des systèmes correspondants et se comportent également de manière similaire sans que le programmeur c'est-à-dire vous n'ait besoin de prêter attention à ces caractéristiques.

Les principales composantes de Flutter sont :

- La plate-forme Dart les application flutter sont écrit en Dart et utilisent de nombreuses fonctionnalités plus avancées du langage les versions des applications Flutter sont compilées avec ahead-of-time (AOT) compilation à la fois sur Android et iOS rendant possible les hautes performances de Flutter sur les appareils mobile et fonctionne dans la machine virtuelle Dart et dispose d'un moteur d'exécution « Just-in-time »
- Le moteur de Flutter le moteur de Flutter écrit principalement en C++ fournit un support de rendu de bas niveau en utilisant la bibliothèque graphique Sika de Google, il s'interface avec des SDK spécifiques à la plate-forme fournis par Android et iOS

- La bibliothèque fondation la bibliothèque fondation écrite en Dart fournit des classes et des fonctions de base qui sont utilisées pour construire des applications comme API pour communiquer avec le moteur
- Les Widget spécifiques à la conception le Framework Flutter contient deux ensembles de widget qui sont conformes à des langages de conception spécifiques, les widgets Material Design implémente le langage de conception

3.3.2.3 Dart

C'est un langage de programmation orienté objet basé sur la classe récupérateur de mémoire optimisé pour les applications sur plusieurs plateformes il est développé par Google et se concentre sur la réalisation d'application mobile, web, bureau...

La programmation dans Dart doit être une alternative attrayante à JavaScript dans des navigateurs Web modernes à cause des faiblesses de JavaScript les développeurs ne peuvent pas être corrigées par son développement.

Le langage Dart est développé et lancé depuis 2010 la syntaxe de son programme ressemble au langage de programme C.

Google a créé de nombreux langages de programmations Dart depuis l'émergence de Flutter ce langage retrouve des couleurs et a même connu une version 2.0. Il est très performant pour gérer l'allocation et éliminer des objets en mémoire qui ne sont plus utilisés « garbage collector ». Il offre deux modes de fonctionnement AOT (Ahead Of Time) permet de générer une application native pour chaque plateforme et le code sera optimisée directement pour l'architecture qu'il fonctionnera, et JIT (Just In Time) offre la fonctionnalité de Hot Reload lors de développement. [7]

3.3.2.4 Android Studio

Android Studio est l'environnement de développement intégré (IDE) pour le développement des application mobiles il offre encore plus de fonctionnalités qui améliorent la productivité lors de la création des applications telles que

- Un système de construction flexible basé sur Gradle.

- un environnement unifié où vous pouvez développer pour tous les appareils Android.
- Modèles de code et intégration GitHub pour l'aide à créer des fonctionnalités d'application communes et à importer le code.
- Outils et cadres tests étendus.

3.3.2.4. X-Code

Est un environnement de développement pour MacOS et tvOS. L'API Cocoa permet de programmer avec les langages Objective-C, Ruby, Swift (langage de programmation d'Apple) fourni avec toute une suite logicielle pour développeurs et programmeurs il permet de créer des logiciels utilisant toutes les fonctionnalités de MacOS et d'UNIX il est obtenu gratuitement sur Mac App Store. Il était fourni en standard avec chaque Mac, sur les disques d'installation de Mac OS X 10.6, Mac OS X 10.5, 10.4 et 10.3.

3.3.2.6 Simulateur

Simulator vous permet de créer rapidement des prototypes et de tester des versions de votre application pendant le processus de développement est installé dans le cadre des outils Xcode, Simulator s'exécute sur votre Mac et se comporte comme une application Mac standard tout en simulant un environnement iPhone, iPad, Apple Watch ou Apple TV. Considérez le simulateur comme un outil de test préliminaire à utiliser avant de tester votre application sur un appareil réel.

Le simulateur vous permet de simuler des appareils iOS, watchOS et tvOS exécutant les systèmes d'exploitation actuels et certains anciens systèmes chaque combinaison d'un appareil simulé et d'une version logicielle est considérée comme son propre environnement de simulation, indépendant des autres avec ses propres paramètres et fichiers ces paramètres et fichiers existent sur chaque appareil que vous testez dans un environnement de simulation.

3.3.2.7 Firebase

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, JavaScript, PHP, C++...) il propose d'héberger en NoSQL et en temps réel des bases de données du contenu de l'authentification sociales des notifications ou

encore des services tel qu'un serveur de communication temps réel. Lancé en 2011 sous le nom d'Envolv par Andrew Lee et par James Templin le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google « Alphabet ».

Toute l'implémentation et la gestion serveur de Firebase est à la charge exclusive de la société Alphabet. Les applications qui utilisent Firebase intègrent une bibliothèque qui permet les diverses interactions possibles.

3.4. Démarche à suivre

Dans cette section on va essayer de parler un peu plus en détail sur les étapes à suivre afin de développer et mettre en œuvre une application mobile propre et stable.

3.4.1 Préparation de l'environnement de travail

- a) Installer Flutter adéquat au système à partir du site www.Flutter.Dev, dans notre cas on a choisi MacOS. Pour tester la bonne installation ou non du Flutter, il suffit d'exécuter la commande **Flutter doctor** dans le terminal.
- b) Installer XCode pour pouvoir utiliser les simulateurs propres aux appareils non disponibles à l'émulateur Android studio. Puisque notre smartphone est un appareil iPhone on doit installer cet outil. Après l'installation de XCode, un utilitaire d'amélioration et de mise à jour sera installer automatiquement celui de git.
- c) Copier Flutter SDK pour l'exécuter directement à partir de son emplacement d'installation, on peut bien sur l'ajouter au **path** pour plus tard le lancer à partir de n'importe quel emplacement.
- d) Installer Android Studio et Android SDK, cet utilitaire sert comme une plateforme du système d'exploitation mobile.
- e) Exécuter maintenant la commande « flutter config –android-studio-dir » pour bien s'assurer de la bonne configuration et le chainage entre les deux outils.
- f) Installer la machine virtuelle JAVA.

3.5 Écriture du code

On peut maintenant passer à la partie programmation et préparation des pages, ou des programmes spécifiques à chaque fonctionnalité. On s'intéresse ici à présenter une partie de

notre programme comme exemple (Figure 9) puisque c'est trop de copier toutes les parties du code. Le programme choisi est celui de la page d'accueil. Le choix de ce programme n'est pas au hasard mais il est fondé sur le fait que les lignes écrites sur cette classe contiennent les différentes surprises qu'on peut rencontrer dans la programmation des autres classes à savoir : (1) la déclaration des variables et leurs utilisations, (2) les tests et les codes de vérification, et (3) la connexion à une table de la base de données pour vérifier la vraisemblance avec le contenu de la base et ce qui a été saisi par l'utilisateur.

3.6 Commentaire du code

```
buildDoctorList() {  
  return Padding(  
    padding: EdgeInsets.symmetric(  
      horizontal: 30,  
    ), // EdgeInsets.symmetric  
    child: Column(  
      children: <Widget>[  
        DoctorCard(  
          'Dr. Khadidja',  
          'Heart Surgeon - Hospitals',  
          'assets/images/doctor1.png',  
          kBlueColor,  
        ), // DoctorCard  
        SizedBox(  
          height: 20,  
        ), // SizedBox  
        DoctorCard(  
          'Dr. Benyahia',  
          'Dental Surgeon - Hospitals',  
          'assets/images/doctor2.png',  
          kYellowColor,  
        ), // DoctorCard  
        SizedBox(  
          height: 20,  
        ), // SizedBox  
        DoctorCard(  
          'Dr. Boudekak',  
          'Eye Specialist - Hospitals',  
          'assets/images/doctor3.png',  
          kOrangeColor,  
        ), // DoctorCard  
      ],  
    ),  
  );  
}
```

Figure 9: home-screen. Dart

1

Crée des inserts avec des décalages horizontaux symétriques pour justifier l'interface

2

Ajouter le texte Widget qui indique les informations du médecin (nom, Spécialité...)

3

L'extrait qui fait en sorte la taille exacte de box

4

Un commentaire

3.7 Fonction de test

C'est la fonction qui permet de faire exécuter l'application Flutter

```
void main() => runApp(MyApp());
```

L'interface BuildContext est utilisée pour décourager la manipulation directe des objets Élément.

```
@override  
Widget build(BuildContext context) {
```

3.8 Préparation la base sur Firebase

Firebase vous donne accès à un grand nombre de ressources gratuites de stockage de fichiers et permet à votre application de gagner en complexité et professionnalisme pour y avoir accès vous allez devoir lier votre application Flutter à Firebase.

- Configurer projet Firebase sur le site.
- Ajouter une clé SHA de l'application qui permet à Firebase de communiquer avec l'application.
- Télécharger le fichier google-services. Json.

- Paramétrer le projet Flutter.

La figure 10 nous affiche notre base de données de compte d'inscription à notre application avec Firebase.

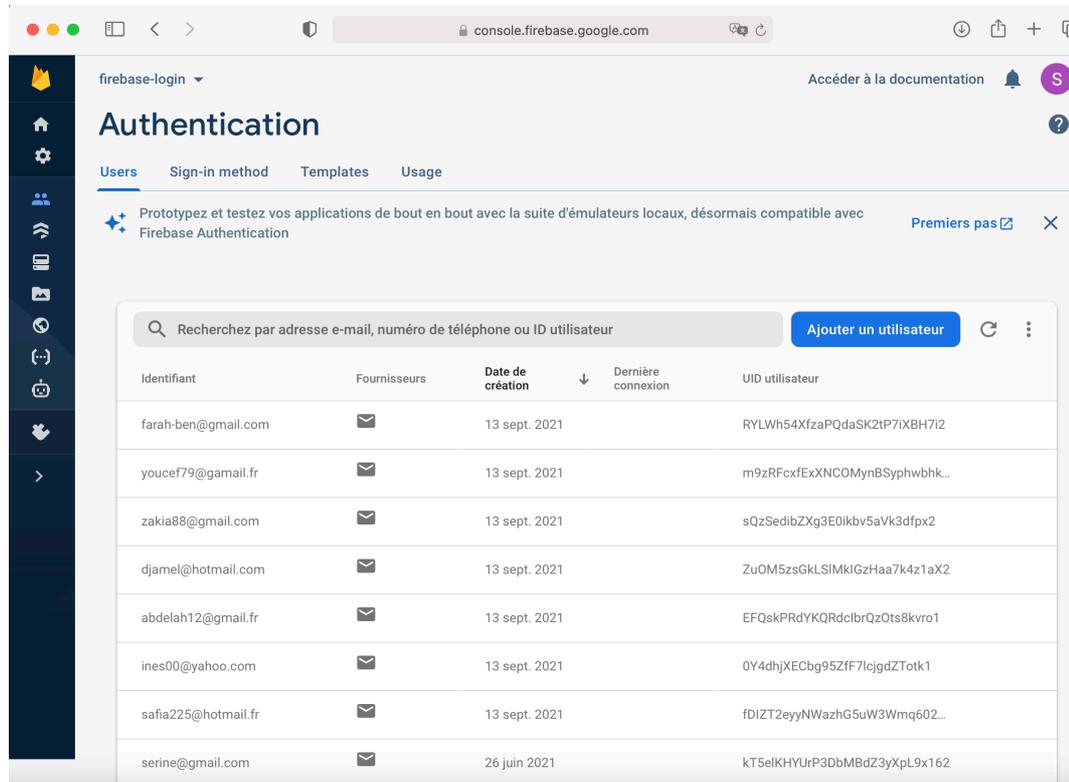


Figure 10: authentification avec Firebas

3.9 Lier Firebase a Flutter

Dans ce cas on va créer un Dart file « Authentification » qui contient l'URL de Firebase.

```
authentication.dart x pubspec.yaml x login.dart x signup.dart x main.dart x http_exception.de v
1 import 'dart:convert';
2 import 'dart:io';
3
4 import 'package:flutter/cupertino.dart';
5 import 'package:http/http.dart' as http;
6
7 class Authentication with ChangeNotifier {
8   Future<void> signUp(String email, String password) async {
9     const url =
10       'https://identitytoolkit.googleapis.com/v1/accounts:signUp?key=AIzaSyCRID
11
12     try {
13       final response = await http.post(url,
14         body: json.encode({
15           'email': email,
16           'password': password,
17           'returnSecureToken': true,
18         }));
19       final responseData = json.decode(response.body);
20       // print(responseData);
21       if (responseData['error'] != null) {
22         throw HttpException(responseData['error']['message']);
23       }
24     } catch (error) {
25       throw error;
26     }
27   }
28
29   Future<void> logIn(String email, String password) async {
30     const url =
31       'https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?ke
```

Figure 11:authentication.dart

3.10 Teste de l'application

Welcome page

On va commencer par la présentation la première interface de l'application qui affiche à l'utilisateur l'accès à notre application « welcome page ».

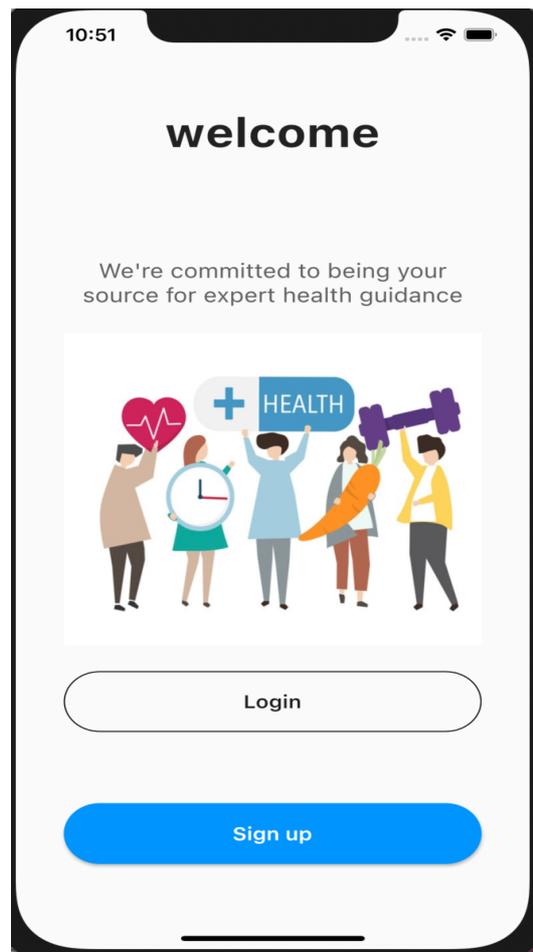
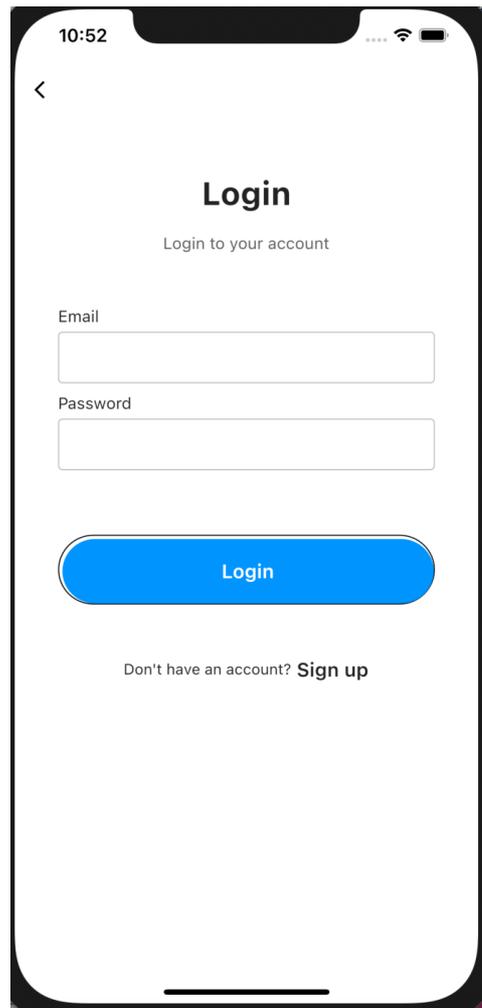


Figure 12: Welcome Page

Login page

Cette interface est faite pour les utilisateurs qui ont déjà admis à cette application il suffit juste entrer l'email et le mot de passe.

A mobile application login page. At the top, the status bar shows the time 10:52, signal strength, Wi-Fi, and battery icons. Below the status bar is a back arrow icon. The main heading is "Login" in bold, followed by the subtitle "Login to your account". There are two input fields: "Email" and "Password". Below the input fields is a blue rounded button labeled "Login". At the bottom, there is a link: "Don't have an account? Sign up".

10:52

<

Login

Login to your account

Email

Password

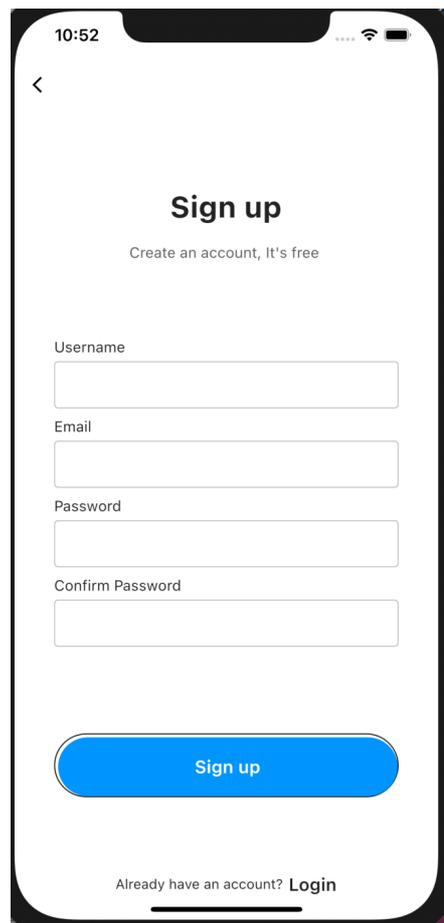
Login

Don't have an account? [Sign up](#)

Figure 13: Login Page

Sign Up page

La figure ci-dessous autorise aux nouveaux utilisateurs de saisir leurs information (nom, email, mot de passe) et de confirmer leurs mot de passe pour assurer que les données sont bien assurées.

A mobile application sign-up page. At the top, the status bar shows the time 10:52, signal strength, Wi-Fi, and battery icons. Below the status bar is a back arrow. The main heading is "Sign up" in bold, followed by the subtext "Create an account, It's free". There are four input fields: "Username", "Email", "Password", and "Confirm Password". Below these fields is a blue rounded button labeled "Sign up". At the bottom, there is a link: "Already have an account? Login".

10:52

<

Sign up

Create an account, It's free

Username

Email

Password

Confirm Password

Sign up

Already have an account? [Login](#)

Figure 14: Sign Up Page

On boarding Page

Après la saisie d'information on passe à l'interface qui nous envois vers le contenu de l'application.

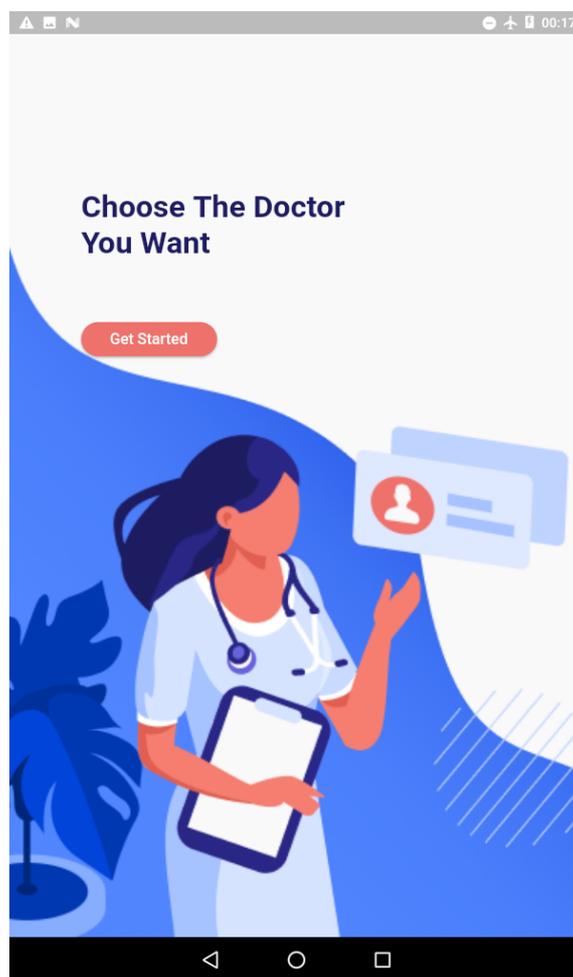


Figure 15:On boarding Page

Home Page

Cette figure affiche l'interface « Home Page » qui contient une barre de recherche des spécialités et leurs médecins.

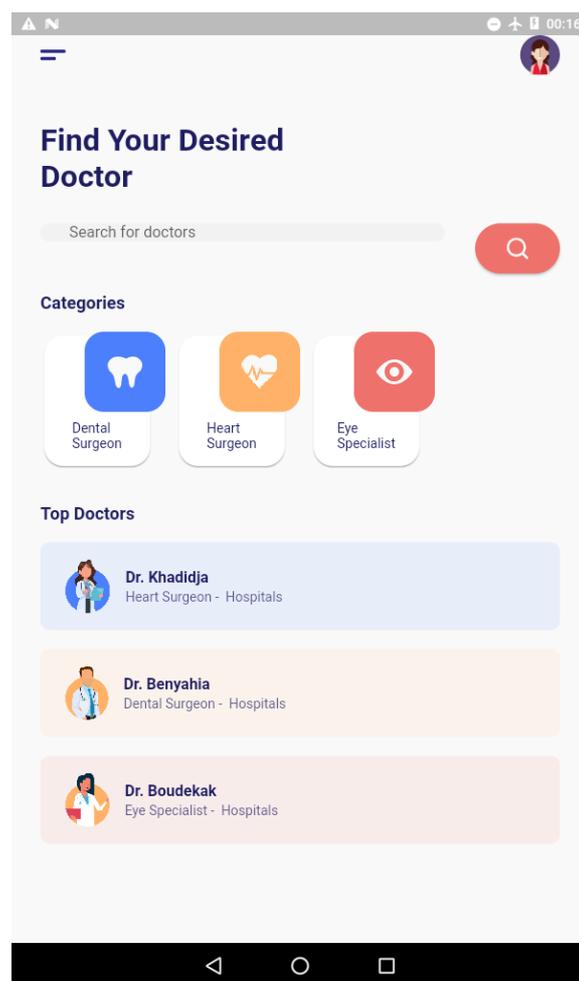


Figure 16:Home Page

Doctor Page

Pour bien choisir un bon médecin on clique sur le bouton de médecin souhaiter pour avoir ses informations et les moyens de le contacter et en plus son programme de travail qui facilite de prendre un rendez-vous.

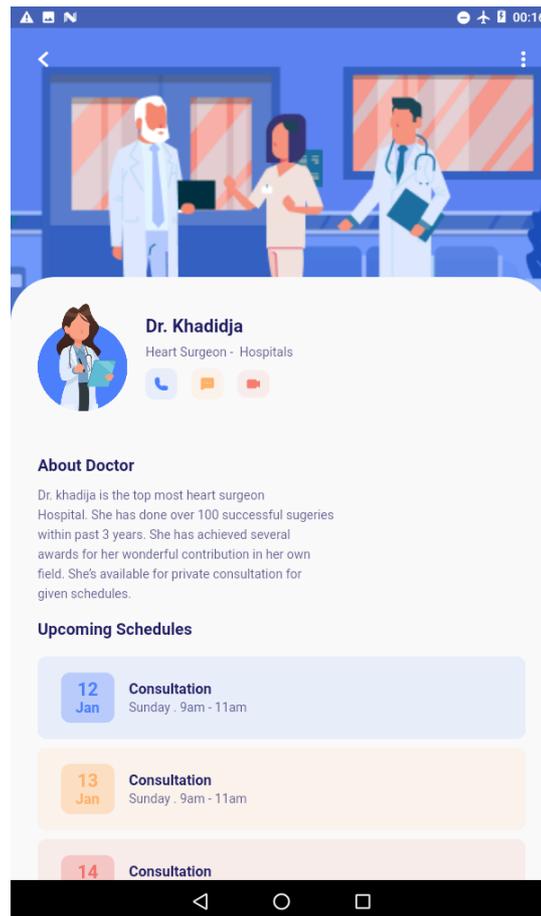


Figure 17:Doctor Page

3.11 Conclusion

Dans le dernier volet de ce mémoire nous avons présenté l'environnement de développement en plus des Framework les plus importants qui nous ont permis d'implémenter l'application. Par la suite, nous avons présenté les différentes interfaces de notre application.

Conclusion générale

Ces dernières années, la téléphonie mobile a été sans doute le secteur le plus dynamique, le plus rentable et le plus innovant de toute l'Industrie des Télécommunications. Notre Projet porte sur l'organisation et l'automatisation de la prise d'un suivi médical d'un patient afin d'augmenter la fiabilité, l'efficacité de l'effort humain et de faciliter les tâches pénibles au sein d'un organisme en le déroulant sur une plateforme d'une application mobile. Le but principal de ce projet se base sur l'optimisation du temps ainsi les patients n'ont plus besoin d'attendre l'ouverture de la clinique ou l'arrivée d'un médecin pour avoir une consultation. Ils seront libérés de leurs contraintes professionnelles et familiales, d'où une simplicité de la charge quotidienne

La réalisation de notre application est à présent achevée, elle comporte les fonctionnalités de

- Gestion et suivi du dossier médical
- Gestion du fichier patient
- la fiche complète d'un médecin

Nous avons donc eu l'opportunité d'approfondir nos connaissances que ce soit au plan scientifique ou personnel. Pour conclure, on a évalué les principaux avantages et les points forts du Framework Flutter pour améliorer la gestion d'un cabinet médical

Comme une autre expérience au niveau de l'application des concepts de langages, c'est normal de ne pas pouvoir éviter certains problèmes et difficultés au niveau de la modélisation conceptuelle et au niveau de l'implémentation et la programmation. Cependant, nous avons essayé de dégager les solutions les mieux adaptées à nos objectifs, nos contraintes et nos moyens disponibles ces solutions ne prétendent nullement être les meilleures car en informatique il n'y a pas de solution absolue. Il est à noter que cette application peut être améliorée, pour répondre aux besoins des autres spécialités plus appropriées, ainsi que le suivi de rapport d'activités des dossiers médicaux.

Le sujet sur lequel nous avons travaillé était bon et utile à la vie des gens nous étions très intéressés par la recherche et le développement de nos connaissances nous avons appris à exploiter nos acquis.

Webographie

- [1] <https://jobphoning.com/dictionnaire/application-mobile#>
- [2] <https://selectra.info/telecom/actualites/acteurs/marche-smartphone-t1-2021-quelles-sont-les-marques-les-plus-demandees>
- [3] https://fr.wikipedia.org/wiki/Application_mobile
- [4] <https://edutechwiki.unige.ch/fr/Flutter>
- [5] <https://mbamci.com/comprendre-le-developpement-dapplication/>
- [6] <http://dspace.univtlemcen.dz/bitstream/112/10950/1/Ms.Tel.Kehli%2BBennezair.PDF>
- [7] <https://www.apollo-formation.com/developpement/flutter-framework-applications-mobiles-multiplateformes.html>

- [8] <https://ichi.pro/fr/cycle-de-vie-des-applications-dans-flutter-251489267029758>
- [9] <https://fr.wikipedia.org/wiki/Firebase>
- [10] <https://fr.wikipedia.org/wiki/IOS>
- [11] <https://fr.wikipedia.org/wiki/Xcode>
- [12] <https://dspace.univ-guelma.dz/jspui/bitstream/123456789/4277/1/PFE2019%20Kouadria%20Lina%20%28Tchi%20drive%29.pdf>
- [13] <http://dspace.univ-msila.dz:8080/xmlui/bitstream/handle/123456789/21512/Charik%20Abdessamie%2CCharif%20Noureddine.pdf?sequence=1&isAllowed=y>
- [14] <http://dspace.univ-tlemcen.dz/bitstream/112/10906/1/Ms.EBM.Beneddra.pdf>
- [15] <https://www.hebergementwebs.com/nouvelles/les-avantages-et-les-inconvenients-du-developpement-d-applications-flutter>
- [16] <http://dspace.univ-msila.dz:8080/xmlui/bitstream/handle/123456789/1922/MANSOURI%20Bilal.pdf?sequence=1&isAllowed=y>
- [17] <https://dspace.univ-guelma.dz/jspui/bitstream/123456789/4277/1/PFE2019%20Kouadria%20Lina%20%28Tchi%20drive%29.pdf>
- [18] <http://www.univ-bejaia.dz/dspace/bitstream/handle/123456789/8639/Conception%20et%20réalisation%20d'un>

e%20application%20mobile%20sensible%20au%20contexte%20pour%20un%20musée.pdf?s
equence=1&isAllowed=y

[19] <https://www.memoireonline.com/03/12/5572/Gestion-d-un-cabinet-medical-mise-en-place-d-un-logiciel-pour-la-gestion-dun-cabinet-medical.html>

Résumé

Ce projet de fin d'étude présente la conception et la réalisation d'une application mobile pour le suivi des patients. Le système de santé Algérien se modernise de jour en jour se trouve dans la nécessité d'informatiser l'information médicale. Cette application destinée aux smartphones vise à satisfaire les besoins des patients. Afin d'atteindre cet objectif, il nous a été proposé de réaliser une application sous iOS en proposant diverses options et fonctionnalités par exemple on peut contacter un médecin, on a choisi de modéliser notre système avec le formalisme UML pour réaliser l'étude conceptuelle du système, on a utilisé le Framework Flutter sous l'environnement de développement Android Studio, et Firebase l'implémentation de la base de données.

Mots clés : Flutter, Android Studio, Xcode, iOS, Dart, Firebase, Dossier médical

Abstract

This end-of-study project presents the design and realization of a mobile application for the follow-up of patients. The Algerian health system is modernizing day by day is in the need to computerize medical information. This application intended for smartphones aims to satisfy the needs of patients. In order to achieve this objective, we were proposed to realize an application under iOS by proposing various options and functionalities for example we can contact a doctor, we chose to model our system with the UML formalism to realize the conceptual study of the system, we used the Flutter Framework under the development environment Android Studio, and Firebase the implementation of the database.

Key words : Flutter, Android Studio, XCode, iOS, Dart, Firebase, Medical file

المخلص

يقدم مشروع نهاية الدراسة هذا تصميم وتنفيذ تطبيق للهاتف المحمول لمتابعة المرضى. النظام الصحي الجزائري يتم تحديثه يوما بعد يوم وهو بحاجة إلى حوسبة المعلومات الطبية. يهدف هذا التطبيق المخصص للهواتف الذكية إلى تلبية احتياجات المرضى. من أجل تحقيق هذا الهدف، اقترحنا تحقيق تطبيق تحت iOS من خلال اقتراح خيارات ووظائف مختلفة على سبيل المثال يمكننا الاتصال بالطبيب، واخترنا نمذجة نظامنا مع شكليات UML لتحقيق الدراسة المفهومية للنظام، نحن استخدم Flutter Framework في ظل بيئة التطوير Android Studio، و Firebase لتنفيذ قاعدة البيانات.

الكلمات المفتاحية: Flutter، Android Studio، Xcode، iOS، Dart، Firebase، ملف طبي