



PFA n° : F1-082-155

Université Chouaib Doukkali  
Ecole Nationale des Sciences Appliquées d'El Jadida  
Département Télécommunications, Réseaux et Informatique



## PROJET DE FIN D'ANNEE

Filière : **2ITE**

Niveau : **2<sup>ème</sup> Année**

Sujet :

Application Mobile pour le Déploiement d'un Modèle Machine Learning pour le Dépistage Automatique du COVID19

Réalisé Par :

**Omar Mazer**  
**Hassane HASSAR**

Encadré par :

**Prof. Asmaa EL HANNANI**

---

Présenté le **31/05/2021** devant le jury composé de :

Prof. Asmaa EL HANNANI,  
Prof. Mohamed EL BOUJNOUNI,

Professeur à l'ENSAJ  
Professeur à l'ENSAJ

Année Universitaire : 2020/2021

# Résumé

Les applications concrètes du **Machine Learning** (ML) sont de plus en plus nombreuses dans notre quotidien. Néanmoins, beaucoup de modèles ML sont validés uniquement en expérimentation et ne passent jamais à l'industrialisation.

Ce projet intervient dans la phase précédemment citée, en mettant en place une **application mobile Android** pour déployer un modèle ML (déjà entraîné) de dépistage automatique du **COVID19** dans un scénario réel afin de mieux évaluer son apport.

Concrètement, notre application collecte des données saisies par l'utilisateur et en réponse elle lui donne la probabilité de son infection. Les données téléchargées sur l'application vont être utilisées par la suite pour augmenter la base de données d'apprentissage des algorithmes de dépistage automatique.

**Mots-clefs :** Application mobile Android, Machine Learning, Covid-19, industrialisation.

# Remerciements

C'est un plaisir et un moment très agréable de rendre hommage et de formuler des remerciements aux personnes qui, d'une manière ou d'une autre, ont apporté leur soutien et contribué à finaliser ce travail.

Nous tenons, tout d'abord, à exprimer nos sincères gratitude envers notre encadrant Pr. **Asmaa EL HANNANI**, pour les conseils qu'il n'a pas cessé de nous prodiguer, sa compréhension et la confiance qu'il a toujours témoignée à notre égard. De même, nous adressons nos plus vifs remerciements à monsieur Pr. **Mohamed EL BOUJNOUNI** pour nous avoir honoré en participant à la soutenance de ce projet dans la personne de membre du jury.

Nous tenons également à exprimer notre profonde gratitude et respectueuse reconnaissance pour toute personne pour l'aide et les conseils concernant les missions évoquées dans ce rapport. Bien évidemment, nous remercions nos amis et familles qui tout au long de nos études n'ont pas cessé de nous soutenir et nous accompagner dans tous ces moments.

Merci à tous !



# Table des matières

<b>1</b>	<b>Introduction .....</b>	<b>8</b>
<b>2</b>	<b>Généralités et étude de l'existant .....</b>	<b>10</b>
	Introduction .....	10
	2.1 Généralités.....	10
	2.1.1 Utilisation des modèles ML pour le dépistage automatique des maladies .....	10
	2.1.2 Analyse de l'image comme outil de diagnostic des maladies.....	11
	2.1.3 Les symptômes de Covid-19 .....	11
	2.1.4 Le diagnostic de Covid-19 .....	12
	2.2 Etude de l'existant.....	12
	Conclusion.....	14
<b>3</b>	<b>Etude Conceptuelle .....</b>	<b>15</b>
	Introduction .....	15
	3.1 Besoins fonctionnels .....	15
	3.2 Besoins non-fonctionnels .....	15
	3.3 Diagramme de cas d'utilisation .....	16
	3.4 Diagramme de classes .....	16
	3.5 Diagramme de séquences.....	17
	Conclusion.....	18
<b>4</b>	<b>Choix technologiques et Réalisation.....</b>	<b>19</b>
	Introduction .....	19
	4.1 Choix des technologies et environnement du travail .....	19
	4.1.1 Choix de technologie de programmation .....	19
	4.1.2 Choix de Base de données :.....	22
	4.2 Présentation de l'application .....	23
	4.2.1 Plan de navigation de l'application .....	23
	4.2.2 Interfaces de l'application : .....	24
	Conclusion.....	30

<b>5 Test et Evaluation .....</b>	<b>31</b>
Introduction .....	31
5.1 Test d'utilisabilité : .....	31
5.2 Test de compatibilité : .....	32
5.3 Test de la base de données : .....	33
5.4 Test de montée en charge : .....	34
Conclusion.....	34
<b>6 Conclusion et perspectives .....</b>	<b>35</b>
<b>Bibliographie .....</b>	<b>36</b>

# Liste des figures

<b>Figure 1 :</b> <i>COVID-19 Sounds App</i> .....	13
<b>Figure 2 :</b> <i>Projet OpenSigma MIT</i> .....	13
<b>Figure 3 :</b> <i>Projet Coswara</i> .....	14
<b>Figure 4:</b> <i>Diagramme de cas d'utilisation</i> .....	16
<b>Figure 5 :</b> <i>Diagramme de classes</i> .....	17
<b>Figure 6 :</b> <i>Diagramme de séquences</i> .....	17
<b>Figure 7 :</b> <i>Déployer le modèle sur un Serveur</i> .....	20
<b>Figure 8 :</b> <i>Déployer le modèle au niveau de l'application mobile</i> .....	21
<b>Figure 9 :</b> <i>Le graphe obtenu à la base du Tableau 2</i> .....	23
<b>Figure 10 :</b> <i>Plan de navigabilité de l'application</i> .....	24
<b>Figure 11 :</b> <i>Page de chargement</i> .....	24
<b>Figure 12 :</b> <i>Page de politique de confidentialité</i> .....	25
<b>Figure 13 :</b> <i>Page d'accueil</i> .....	25
<b>Figure 14 :</b> <i>Page A propos de</i> .....	26
<b>Figure 15 :</b> <i>Pages dédiés pour la collecte de données</i> .....	26
<b>Figure 16 :</b> <i>Pages de sponsors et partenaires</i> .....	27
<b>Figure 17 :</b> <i>Page de remerciement pour la participation dans l'enquête</i> .....	27
<b>Figure 18 :</b> <i>Page de prise de l'image</i> .....	28
<b>Figure 19 :</b> <i>Page de choix de méthode de prise de l'image</i> .....	28
<b>Figure 20 :</b> <i>Résultat du dépistage et la page de directives et conseils : cas non malade</i> .....	29
<b>Figure 21 :</b> <i>Résultat du dépistage et la page de directives et conseils : cas malade</i> .....	29
<b>Figure 22 :</b> <i>Test d'utilisabilité</i> .....	31
<b>Figure 23 :</b> <i>Test de compatibilité</i> .....	32
<b>Figure 24 :</b> <i>Notre base de données MongoDB</i> .....	33
<b>Figure 25 :</b> <i>Test de la base de données</i> .....	33
<b>Figure 26 :</b> <i>Test de montée en charge</i> .....	34

# Liste des tableaux

**Tableau 1:** *Comparaison des différentes approches de déploiement* ..... 20

**Tableau 2:** *Comparaison entre le temps d'insertion* ..... 22

# Liste des acronymes

**BASE** : Basically Available Soft-state services with Eventual-consistency

**BD** : Base de données

**IA** : Intelligence Artificielle

**ML**: Machine Learning

**PCR**: Polymerase Chain Reaction



# 1

## Introduction

La détection assistée par ordinateur (CADE), également appelée diagnostic assisté par ordinateur (CADx), est un système qui aide les médecins à interpréter les images médicales. ... La DAC est une technologie interdisciplinaire combinant des éléments d'intelligence artificielle et de vision par ordinateur avec le traitement des images radiologiques et pathologiques. En effet, les applications concrètes du Machine Learning (ML) sont de plus en plus nombreuses dans notre quotidien. Néanmoins, beaucoup de modèles ML sont validés uniquement en expérimentation et ne passent jamais à l'industrialisation.

L'objectif de ce projet est de développer une **application mobile Android** pour le dépistage automatique du COVID-19. L'approche envisagée est basée sur la classification automatique des images radiographiques du thorax en utilisant des techniques avancées de traitement d'image et d'apprentissage automatique (ML).

Le système sera conçu pour identifier les anomalies et estimer l'étendue de l'infection pulmonaire en temps réel et détecter automatiquement les patients à haut risque présentant une pneumonie ou d'autres symptômes pulmonaires du COVID-19. L'identification d'une signature visuelle associée au COVID-19 est intéressante, car le test serait très portable, simple à réaliser, peu coûteux, et pourrait permettre un dépistage primaire du COVID-19 dans des zones où les kits de test et/ou l'expertise médicale ne sont pas disponibles ou insuffisants (régions rurales et éloignées, foyers d'infection et groupes de population à risque). Les images et les métadonnées téléchargées vers le système de dépistage seront utilisées ultérieurement pour augmenter le jeu de données d'entraînement des modèles ML et améliorer progressivement la précision et la fiabilité des tests Covid-19. Les données seront stockées sur les serveurs de l'université et seront utilisées uniquement à des fins de recherche.

Ce rapport s'organise en six chapitres de la manière suivante :

- **Généralités et étude de l'existant** : dans cette partie nous allons découvrir les notions générales concernant les modèles ML pour le dépistage automatique du COVID19 décrites par la littérature
- **Etude conceptuelle**: dans ce chapitre, nous allons étudier nos besoins pour résoudre la problématique ainsi que, nous allons détailler les différents diagrammes réalisés.
- **Choix technologiques et réalisation** : ce chapitre consiste à choisir les technologies ainsi que l'environnement de travail, afin d'implémenter une application qui répond à nos besoins.

- **Interfaces de l'application :** dans ce chapitre nous présentons quelques interfaces pour faire expliquer les principales fonctionnalités des différents modules de notre projet.
- **Test et évaluation :** ce chapitre est une partie primordiale dans le cycle de vie génie logiciel afin de rectifier les lacunes et bugs commis dans la phase de réalisation.
- **Conclusion et perceptives :** dans cette partie, nous allons résumer l'étude et le travail réalisé dans ce projet, et dresser les perspectives et le travail restant afin de parfaire relativement notre projet.

# 2

## Généralités et étude de l'existant

### Introduction

Commençons d'abord par la citation de quelques notions générales concernant les modèles Machine Learning pour le dépistage automatique du COVID19 décrites par la littérature. Puis, comparons les solutions existantes.

### 2.1 Généralités

#### 2.1.1 Utilisation des modèles ML pour le dépistage automatique des maladies

L'intelligence artificielle (IA) est définie comme la technologie qui utilise les connaissances informatiques pour représenter un comportement intelligent avec une implication humaine nominale, et l'apprentissage machine (ML) est considéré comme un sous-ensemble des techniques d'IA.

En général, il est reconnu que ce type d'intelligence a commencé avec l'innovation de la robotique. Avec la croissance rapide des vitesses électroniques et de la programmation, les ordinateurs pourraient afficher un comportement intelligent similaire à celui des humains dans un avenir proche. Cela s'explique par les grandes avancées des idées contemporaines en matière de développement de l'IA.

L'intelligence artificielle peut être définie comme l'intelligence humaine exécutée par des machines. En informatique, elle est définie comme la capacité de la machine à émuler un comportement intelligent par elle-même, en n'utilisant rien d'autre que le ML.

Les applications de l'IA en médecine se développent rapidement. Ces dernières années, les projets d'IA couplés à la médecine ont attiré plus de spéculation de l'économie mondiale que les autres projets. En médecine, l'IA fait référence à l'utilisation de processus de diagnostic automatisés et au traitement des patients qui nécessitent des soins. L'utilisation accrue de l'IA dans la prescription permettra d'automatiser une grande partie du rôle, ouvrant ainsi la voie aux experts médicaux d'automatiser une grande partie du rôle, ce qui permettra aux experts médicaux de consacrer leur temps à d'autres obligations, celles qui ne peuvent pas être automatisées.

### 2.1.2 Analyse de l'image comme outil de diagnostic des maladies

Vu que notre application collecte des données saisies par l'utilisateur principalement des images thoraciques et en réponse elle lui donne la probabilité de son infection. Nous découvrons succinctement le principe étant donné que notre model ML est déjà entraîné nous l'utilisons dans ce cadre de projet que comme boîte noire.

La classification automatique d'images est une application de la reconnaissance de formes consistant à attribuer automatiquement une classe à une image à l'aide d'un système de classification. On retrouve ainsi la classification d'objets, de scènes, de textures, la reconnaissance de visages, d'empreintes digitales, de caractères parmi les applications courantes. Il existe deux principaux types d'apprentissage dépendant des informations disponibles sur les données à classer : l'apprentissage supervisé et l'apprentissage non-supervisé. Dans l'approche supervisée, chaque image est associée à une étiquette qui décrit sa classe d'appartenance. Dans l'approche non-supervisée (ou clustering), les données disponibles ne possèdent pas d'étiquettes. Alors au système d'extraire une règle d'appartenance de chaque image à un groupe donne.

Un système de classification automatique d'images est composé des étapes suivantes : l'étape de pré-traitement permettant de "nettoyer" les images ; la phase d'extraction de caractéristiques permettant de décrire l'information pertinente contenue dans l'image à l'aide d'opérateurs ou de descripteurs discriminants ; la phase d'apprentissage permettant de construire une frontière de décision pour identifier la classe d'une image présente à l'entrée du système. Ces trois phases sont essentielles dans la construction du système de classification.

L'objectif de la classification d'images est d'élaborer un système capable d'affecter une classe automatiquement à une image. Les applications de la classification automatique d'images sont nombreuses. On retrouve des applications dans le domaine médical comme la reconnaissance de cellules, de tumeurs dans les mammographies... etc.

### 2.1.3 Les symptômes de Covid-19

Le COVID-19 affecte différentes personnes de différentes manières. La plupart des personnes infectées développent une maladie légère à modérée et se rétablissent sans être hospitalisées.

➤ **Symptômes les plus courants :**

- Fièvre
- Toux sèche
- Fatigue

➤ **Symptômes moins fréquents :**

- Douleurs et courbatures
- Maux de gorge
- Diarrhée
- Conjonctivite

- Maux de tête
- Perte du goût ou de l'odorat
- Éruption cutanée ou décoloration des doigts ou des orteils.

➤ **Symptômes graves :**

- Difficulté à respirer ou essoufflement
- Douleur ou pression dans la poitrine
- Perte de la parole ou des mouvements

En moyenne, il faut **5 à 6** jours à partir du moment où une personne est infectée par le virus pour que les symptômes apparaissent, mais cela peut prendre jusqu'à **14** jours.

### 2.1.4 Le diagnostic de Covid-19

Les tests PCR, ou tests de dépistage virologique, sont utilisés pour le diagnostic de l'infection COVID-19. Ils identifient la présence de gènes propres au SARS-CoV-2. Ils sont spécifiques et fiables, si le prélèvement est bien réalisé. Les tests dits sérologiques sont, eux, destinés à identifier les personnes ayant développé une immunité contre le SARS-CoV-2.[1] Or, force de constater que, ces tests PCR sont d'une part très coûteux et d'autre part ils ne sont pas toujours disponibles. Par contre, l'identification d'une signature visuelle associée au COVID-19 est intéressante, car le test serait très portable, simple à réaliser, peu coûteux, et pourrait permettre un dépistage primaire du COVID-19 dans des zones où les kits de test et/ou l'expertise médicale ne sont pas disponibles ou insuffisants (régions rurales et éloignées, foyers d'infection et groupes de population à risque).

## 2.2 Etude de l'existant

Le domaine de recherche médicale, et plus précisément pour la Covid-19, a réussi, avec la contribution de plusieurs développeurs, à créer des solutions dédiées principalement pour la collecte des données qui seront exploiter par la suite au niveau d'amélioration du modèle de dépistage automatique et d'autres sont encore dans la phase d'industrialisation et le test vu que l'évolution de cette maladie jusqu'à maintenant, comme applications nous citons :

**COVID-19 Sounds App :** La **Figure 1** montre l'application sera utilisée pour collecter des données afin de développer des algorithmes d'apprentissage automatique qui pourraient détecter automatiquement si une personne souffre de COVID-19 en se basant sur le son de sa voix, sa respiration et sa toux, a été lancé par des chercheurs de l'Université de Cambridge.[2]

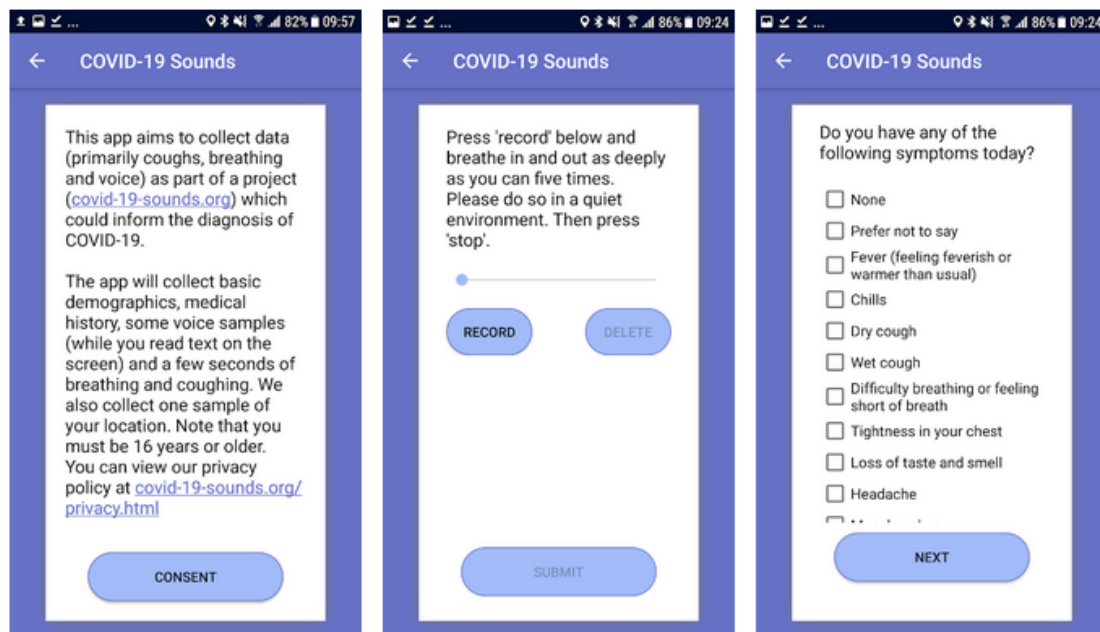


Figure 1 : COVID-19 Sounds App

### Projet OpenSigma MIT :

L'objectif dans cette étude de recherche menée par Brian Subirana, du département de génie mécanique du **Massachusetts Institute of Technology** (M.I.T.) est collecter des échantillons de toux de patients atteints de COVID-19 et de non-patients et tenter de distinguer les deux groupes en utilisant l'intelligence artificielle comme le montre la **Figure 2** ci-dessous. La participation contribuera au développement d'un outil de dépistage automatisé du COVID-19.[3]

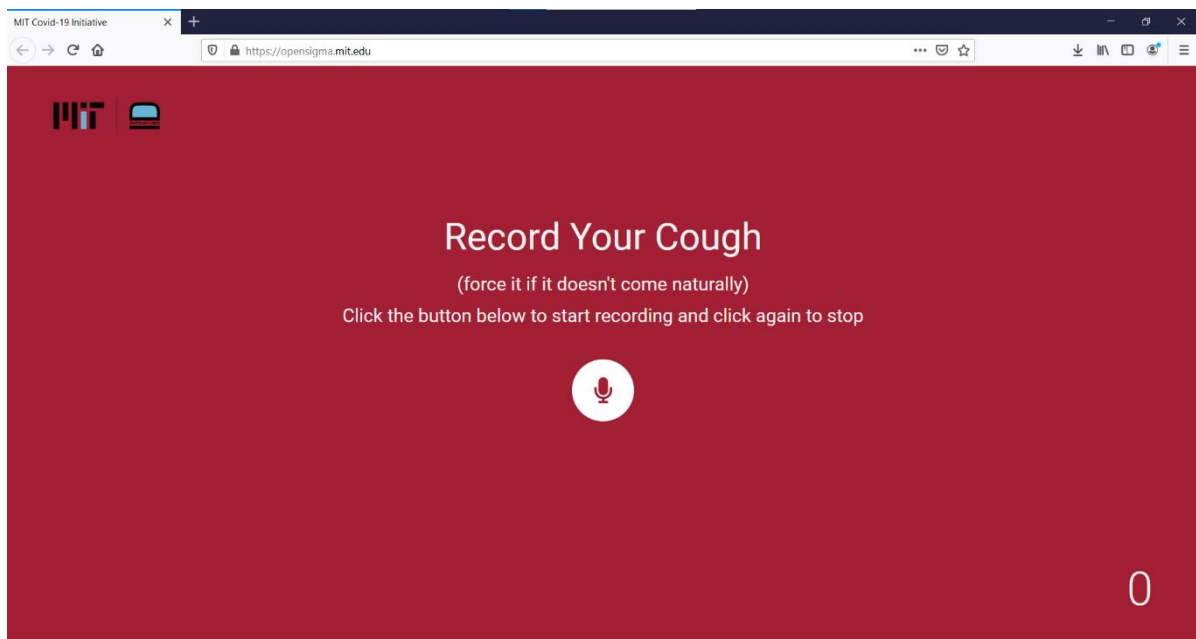


Figure 2 : Projet OpenSigma MIT

## Projet Coswara :

Le projet Coswara de l'**Indian Institute of Science (IISc)** Bangalore est une tentative de construire un outil de diagnostic du Covid-19 basé sur les sons respiratoires, la toux et la parole. Le projet est actuellement en phase de collecte de données la **Figure 3** ci-dessous montre la page de collecte. Les participants doivent fournir un enregistrement des sons respiratoires, des sons de la toux, de la phonation soutenue des voyelles et d'un exercice de comptage qui prend environ 5 à 7 minutes de leur temps. Aucune donnée personnelle identifiable n'est collectée auprès des participants.[4]

**Figure 3 :** *Projet Coswara*

## Conclusion

D'après l'étude de l'état de l'art précédente. En effet, la pandémie de COVID-19 étant un phénomène récent, les applications disponibles dans ce domaine sont encore limitées et des efforts supplémentaires seront déployés à l'avenir. Cependant, dans le même temps, nous pensons que, comme il s'agit d'une pandémie mondiale, la recherche doit examiner les applications qui ont été développées et faire une synthèse pour guider le développement des services futurs.

De notre part, notre application mobile Android proposée a pour objective purement le dépistage automatique du COVID19 à partir du déploiement d'un model ML déjà entraîne- cette dernière point le fait que le model ML déjà entraîne-nous donne l'avantage de se concentrer sur la partie déploiement et développement afin de proposer une solution qui respecte non seulement les besoins fonctionnels à savoir donne la probabilité de l'infection de l'utilisateur mais également révélé des caractéristiques de conception telles que la fiabilité, la performance, l'utilité, le soutien, la sécurité, la confidentialité, la flexibilité, la réactivité et la facilité d'utilisation.

# 3 Etude Conceptuelle

## Introduction

Au cours de ce chapitre on va dégager les différents besoins fonctionnels et non fonctionnels ainsi que les fonctionnalités attendues de notre projet ; puisque pour assurer la réussite de tout projet il est essentiel de se donner une vue claire sur les objectifs ainsi que les besoins à satisfaire à travers ce projet, en définissant les différents cas d'utilisation et quelques scénarios qui expliquent ces cas.

### 3.1 Besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée/ sortie du système qui doit permettre :

L'utilisateur doit :

- Répondre à un questionnaire
- Charger des images thoraciques
- Visualiser le résultat

### 3.2 Besoins non-fonctionnels

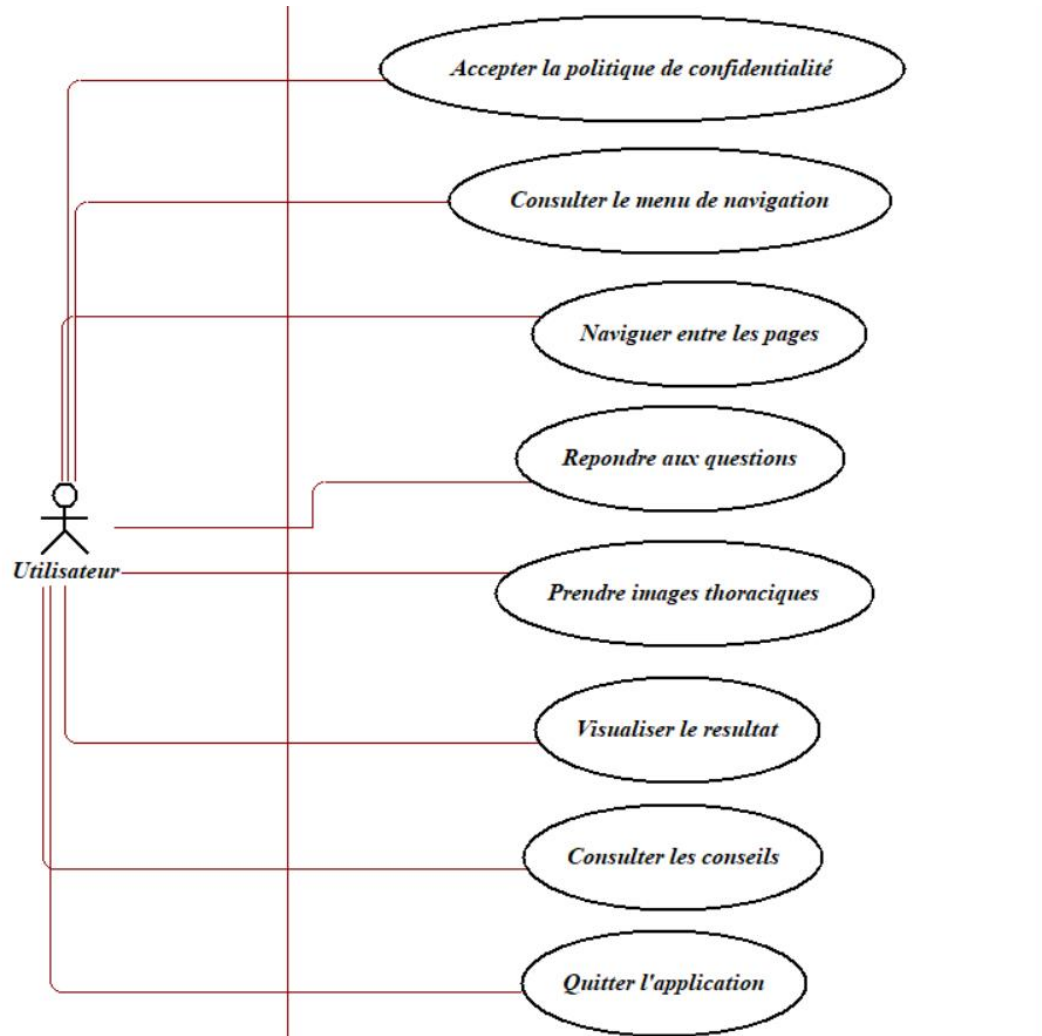
Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type SGBD, de système d'exploitation...).

- **Facilité d'utilisation** : prise en main et robustesse.
- **Performance** : temps de réponse, débit, fluidité...etc.
- **Fiabilité** : tolérance aux pannes.
- **Maintenabilité** : facilité à corriger et à faire évoluer le logiciel.
- **Portabilité** : adaptabilité à d'autres environnements matériels ou logiciels.
- **Sécurité** : intégrité des données et protection des accès. Par exemple, l'authentification est obligatoire pour tous les utilisateurs de l'application elle se fait par login et un mot de passe



### 3.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation nous permet d'avoir une idée générale sur le fonctionnement de notre application comme montré dans la **Figure 4**.



**Figure 4:**Diagramme de cas d'utilisation

### 3.4 Diagramme de classes

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 permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

La **Figure 5** représente le diagramme de classes des entités utilisées dans notre système :

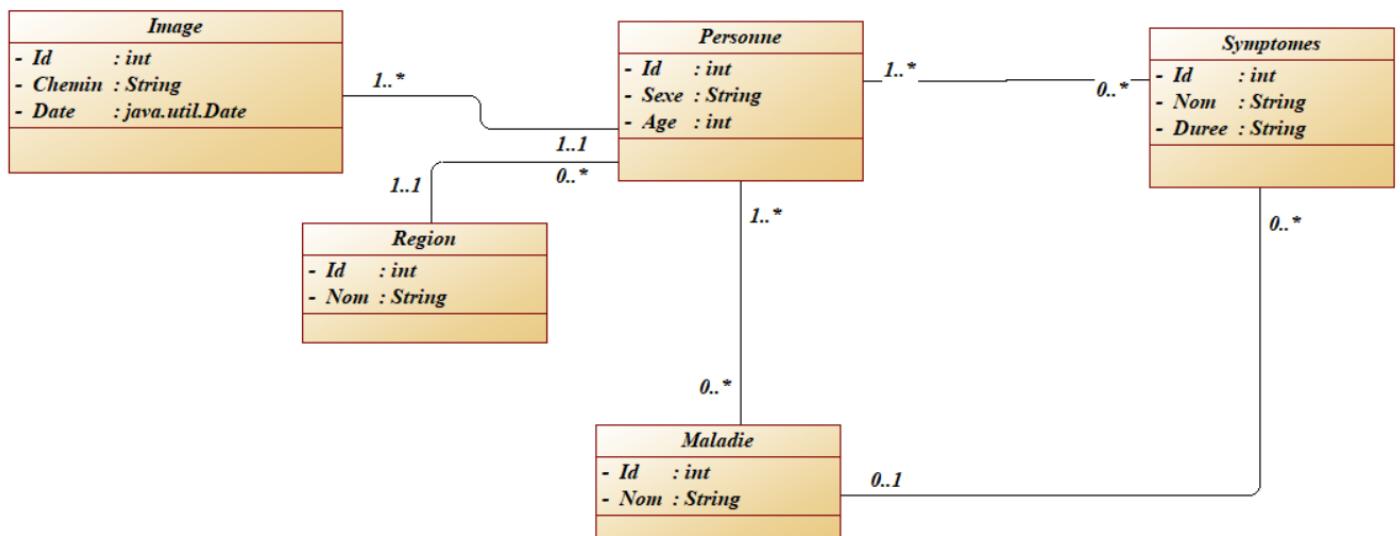


Figure 5 : Diagramme de classes

### 3.5 Diagramme de séquences

Le diagramme de séquence sert à représenter les interactions entre objets en précisant la chronologie des échanges de messages, il représente une instance d'un cas d'utilisation (les scénarios possibles d'un cas d'utilisation donné), il montre sous forme de scénarios, la chronologie des envois de messages issus d'un cas d'utilisation.

Le diagramme de la **Figure 6** illustre le scénario complet du démarrage de l'application.

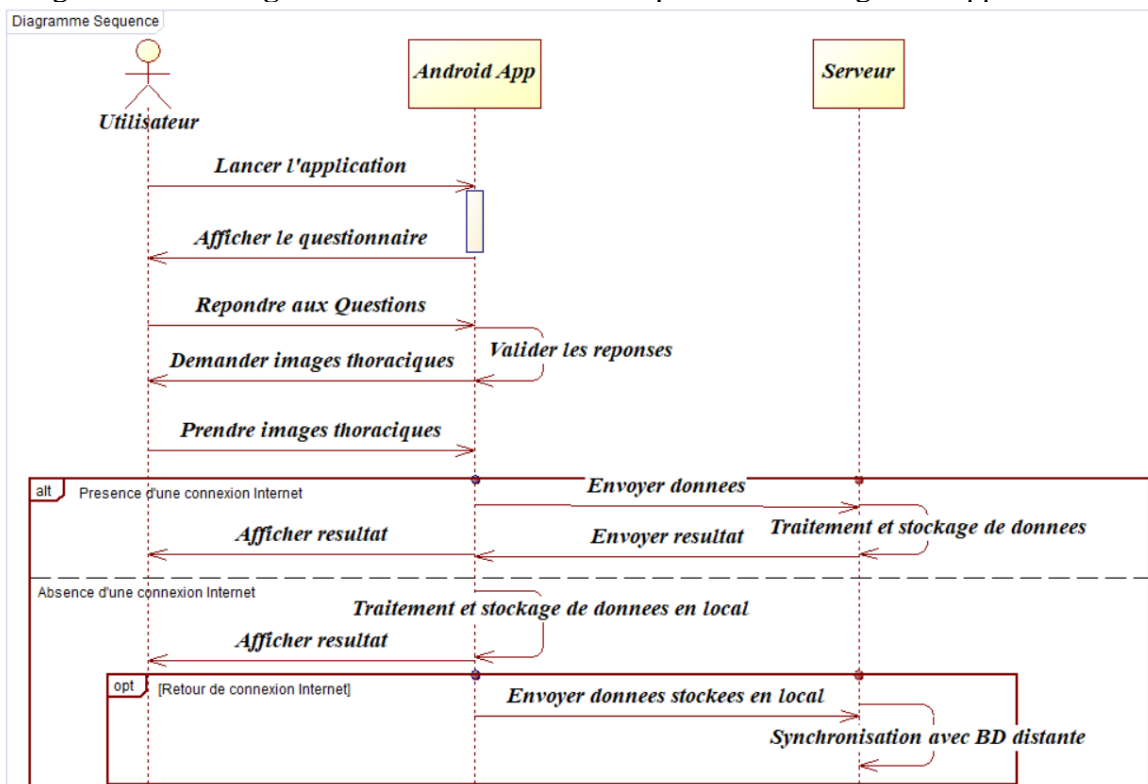


Figure 6 : Diagramme de séquences

## Conclusion

Dans ce chapitre, nous avons modélisé le fonctionnement de l'application afin d'avoir une vue globale et simplifiée du système. Nous avons aussi détaillé les différents modules de l'application ce qui nous a permis d'organiser le travail et d'avoir une idée claire sur le travail à réaliser. Ce travail est décrit plus précisément dans le chapitre qui suit.

# 4

## Choix technologiques et Réalisation

### Introduction

Après avoir spécifier les besoins et fixer les objectifs de notre projet ainsi qu'avoir passer par la phase de conception, nous arrivons enfin à la phase finale qui n'est que la phase de réalisation qui se présente comme sujet principal de ce chapitre.

### 4.1 Choix des technologies et environnement du travail

#### 4.1.1 Choix de technologie de programmation

Le déploiement d'un modèle Machine Learning peut se faire de deux sortes dans le cas d'un projet mobile (Android) comme c'est le cas pour notre projet, donc le **Tableau 1** présente un comparatif entre les deux approches de déploiement :

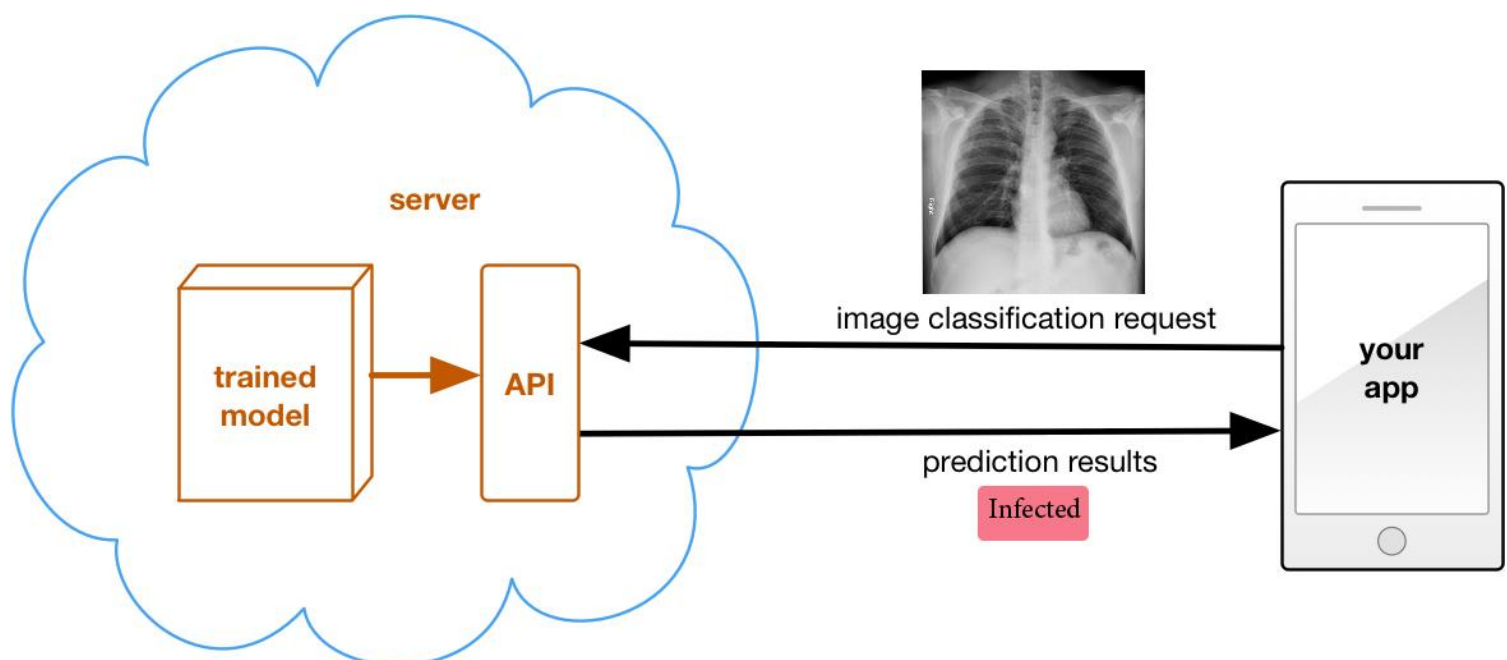
	Déployer le modèle sur un Serveur	Déployer le modèle au niveau de l'application mobile
Avantages	<ul style="list-style-type: none"><li>-Le modèle peut être modifié de manière transparente sans besoin d'envoyer des mises à jour de l'application mobile aux utilisateurs.</li><li>-Le serveur qui héberge le modèle possède une puissance de calcul bien plus importante que celle d'un smartphone.</li><li>-Lorsque toute la logique d'apprentissage automatique est sur le serveur, il est facile de porter l'application sur différentes plates-formes : iOS, Android, Web, etc..</li></ul>	<ul style="list-style-type: none"><li>-Les utilisateurs peuvent utiliser l'application même s'ils n'ont pas accès à Internet.</li><li>-Vitesse : cela peut être beaucoup plus rapide et plus fiable que de faire des requêtes réseau.</li></ul>

<b>Inconvénients</b>	-L'application ne fonctionnera pas sans la connectivité réseau. -Charges d'hébergement et de maintenance.	-Il est plus difficile de mettre à jour le modèle. Les utilisateurs doivent télécharger une mise à jour de l'application pour obtenir le modèle amélioré, ou l'application doit le télécharger automatiquement. -Le portage de l'application sur une plate-forme différente peut être difficile, car vous devez réécrire la partie inférence pour chaque plate-forme (et éventuellement pour chaque type d'appareil).
----------------------	--	--

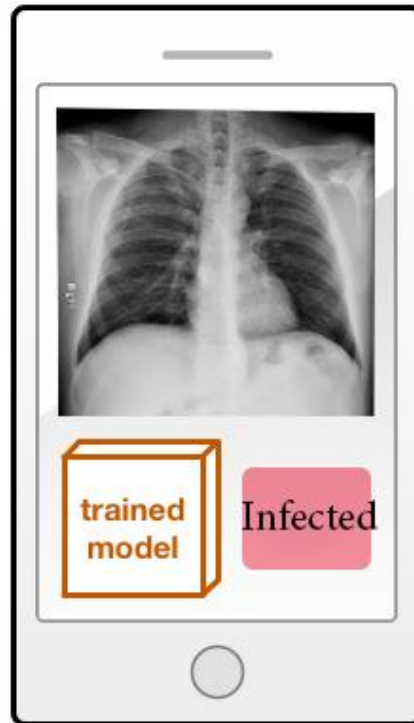
**Tableau 1:** *Comparaison des différentes approches de déploiement*

Après avoir analysé le tableau ainsi que rapprocher les deux approches comme illustré dans la **Figure 7** et la **Figure 8** aux besoins fonctionnels et non fonctionnels de l'application qui est objet de notre projet, nous avons vu la nécessité de combiner entre ces deux approches pour pouvoir assurer une bonne qualité et continuité des services proposés par notre application de telle sorte que dans le cas où :

- **L'utilisateur est connecté à internet :** l'application pourra effectuer une prédiction par le biais d'une communication Client/serveur qu'elle va établir avec le serveur hébergeant le modèle Machine Learning.
- **L'utilisateur n'est pas connecté à internet :** l'application pourra effectuer une prédiction localement avec une possibilité d'avoir des résultats moins précis qu'avec internet.



**Figure 7 :** *Déployer le modèle sur un Serveur*



**Figure 8 :** Déployer le modèle au niveau de l'application mobile

Puisque notre s'est orienté vers l'approche hybride (la combinaison des deux approches), on a choisi de l'implémenter à l'aides des technologies suivantes :

✓ **Android :**

Android est un système d'exploitation mobile basé sur une version modifiée du noyau Linux et d'autres logiciels open source, conçu principalement pour les appareils mobiles à écran tactile tels que les smartphones et les tablettes. Android est développé par un consortium de développeurs connu sous le nom d'Open Handset Alliance et sponsorisé commercialement par Google.[5]

✓ **Flask :**

Flask est définitivement un choix très populaire pour les Data Scientists ; Flask est l'outil qui peut être utilisé pour créer un serveur API. Il s'agit d'un micro-Framework, ce qui signifie que sa fonctionnalité de base reste simple, mais il existe de nombreuses extensions pour permettre aux développeurs d'ajouter d'autres fonctionnalités (telles que l'authentification et la prise en charge de la base de données).[6]

✓ **Tensorflow lite :**

TensorFlow Lite est un ensemble d'outils conçus pour aider les développeurs à exécuter des modèles TensorFlow sur des appareils mobiles, intégrés et IoT. Il permet une inférence de machine learning sur l'appareil avec une faible latence et une petite taille binaire.[7]

TensorFlow Lite comprend deux composants principaux :

- ***L'interpréteur TensorFlow Lite***, qui exécute des modèles spécialement optimisés sur de nombreux types de matériel, dont des téléphones mobiles, des appareils Linux intégrés et des microcontrôleurs.
- ***Le convertisseur TensorFlow Lite***, qui convertit les modèles TensorFlow en un format efficace utilisable par l'interpréteur. Il peut effectuer des optimisations afin d'améliorer les performances et la taille binaire.

#### 4.1.2 Choix de Base de données :

Une fois le choix des technologies de programmation est fixe, la question de la base de données à utiliser est très primordiale notamment dans notre projet dont la base de données constitue la matière première. Toujours dans l'optique d'une optimisation de l'outil, il faut choisir le système de gestion de bases de données le plus efficace possible. Son adéquation avec les besoins du programme impacte directement le temps de développement et la stabilité du système.

En effet, trois SGBD ont retenu notre attention : **Firestore**, **MySQL** et **MongoDB**. Pour la première, c.-à-d., **Firestore** c'était principalement l'inconvénient du coût. Puisqu'elle offre des services limités en quantité de données à stocker dans Firestore database et storage.

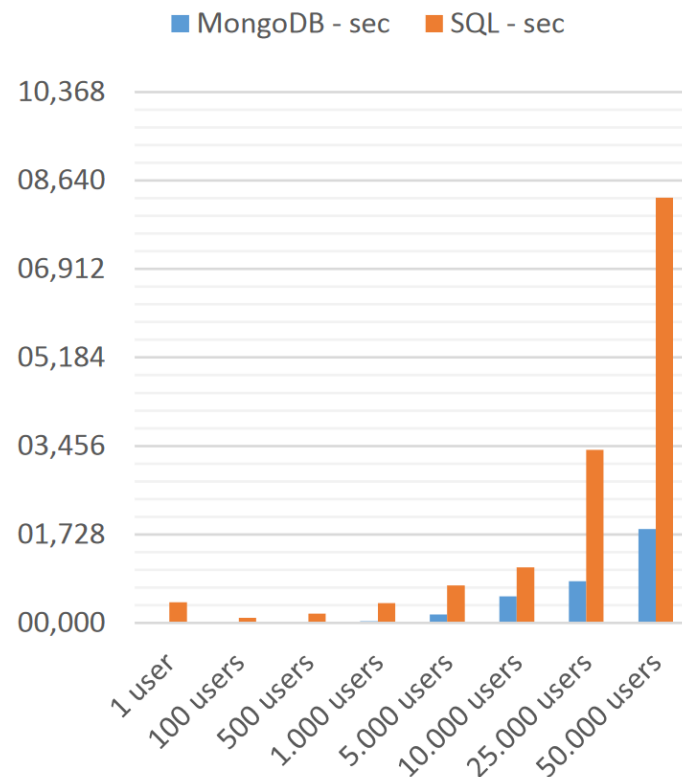
Alors, ils nous restent que le choix entre **MySQL** et **MongoDB**. Dans ce cas-là, les deux solutions sont gratuites, donc le coût ne pose aucun souci. Reste alors à trancher en se basant sur des critères objectives et quantifiables. Fixons comme critères :

##### 1. *Montée en charge*

**MongoDB** : L'un de ses principaux avantages par rapport à **MySQL** est sa capacité à traiter de grandes quantités de données non structurées. Il est magiquement **plus rapide** comme le montre les résultats présentés dans le **Tableau 2** et illustré dans la **Figure 9**. Les gens font l'expérience des performances réelles de MongoDB principalement parce qu'il permet aux utilisateurs d'effectuer des requêtes d'une manière différente qui est plus sensible à la charge de travail.[8]

	<b>MongoDB</b>	<b>SQL</b>
1 utilisateur	00:00:00:003	00:00:00:402
100 utilisateurs	00:00:00:005	00:00:00:096
500 utilisateurs	00:00:00:018	00:00:00:183
1.000 utilisateurs	00:00:00:033	00:00:00:387
5.000 utilisateurs	00:00:00:162	00:00:00:736
10.000 utilisateurs	00:00:00:521	00:00:01:085
25.000 utilisateurs	00:00:00:816	00:00:03:378
50.000 utilisateurs	00:00:01:835	00:00:08:306

**Tableau 2:** Comparaison entre le temps d'insertion



**Figure 9 :** Le graphe obtenu à la base du Tableau 2

## 2. Exigences en matière de données volumineuses

Notre base de données est censée d'agrandir dans le futur, vue que d'une part l'augmentation du nombre d'utilisateurs de notre application et d'autre part, le fait que les données exploitées par notre projet sont qualifiées de volumineuses telles que les images et le son.

Pour conclure cette partie, elle n'existe pas une solution idéale, mais suivant nos besoins en termes de performance, la scalabilité et nos exigences en matière de données volumineuses pour notre application. Par conséquent, nous avons opté pour **MongoDB** comme type de BD NoSQL qui respecte le principe **BASE**, à savoir : **simplement disponible, état souple, finalement consistant**.

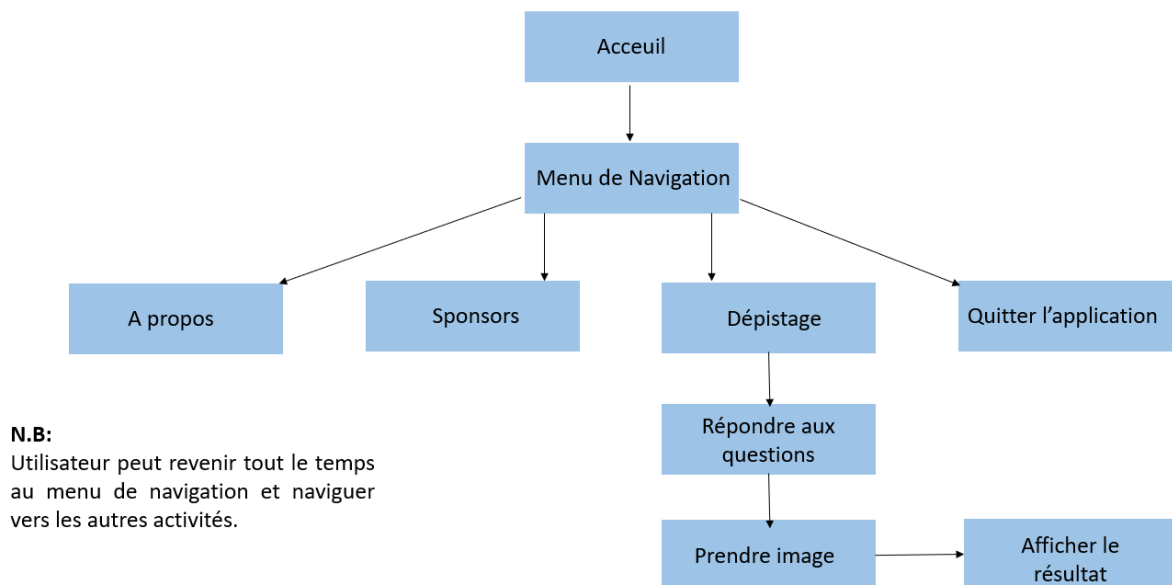
## 4.2 Présentation de l'application

Dans cette partie, nous allons présenter les différentes interfaces de notre application mobile ainsi que leurs utilités.

### 4.2.1 Plan de navigation de l'application

Le plan de l'application mobile offre une vue d'ensemble de l'application, il permet une navigation plus facile, il offre l'accès direct aux différentes activités de l'application comme le montre la **Figure 10**.

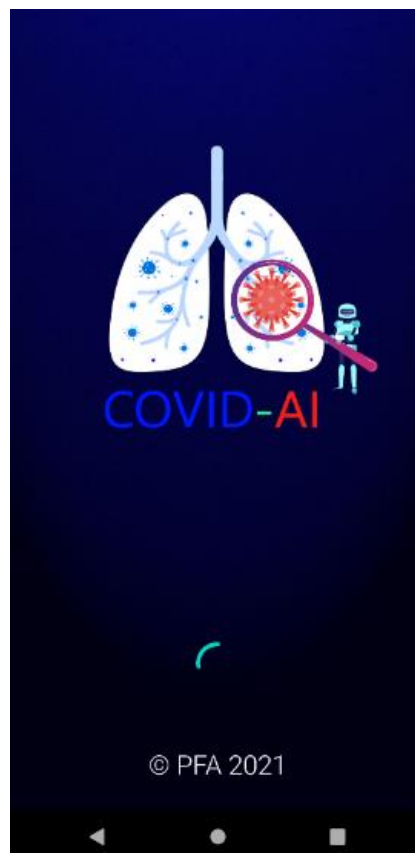




**Figure 10 :** *Plan de navigabilité de l'application*

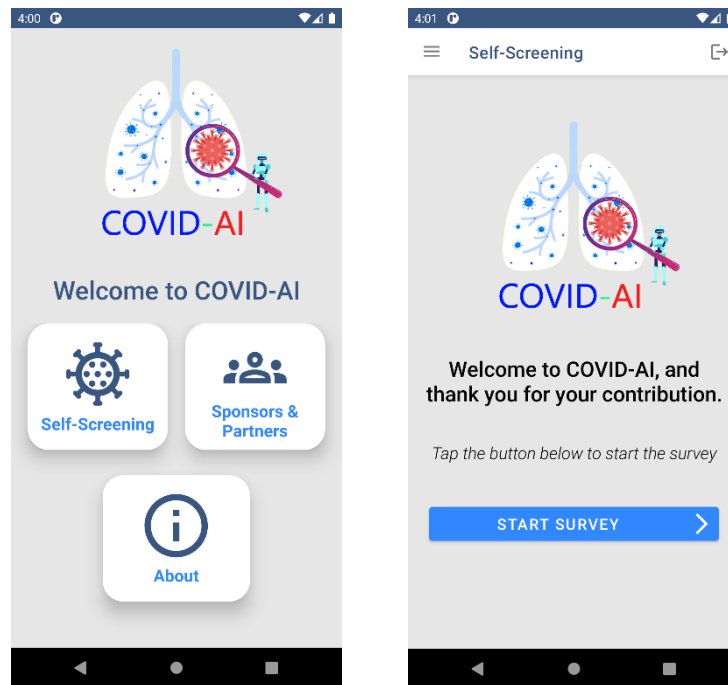
#### 4.2.2 Interfaces de l'application :

La **Figure 11** suivante montre la page de chargement.



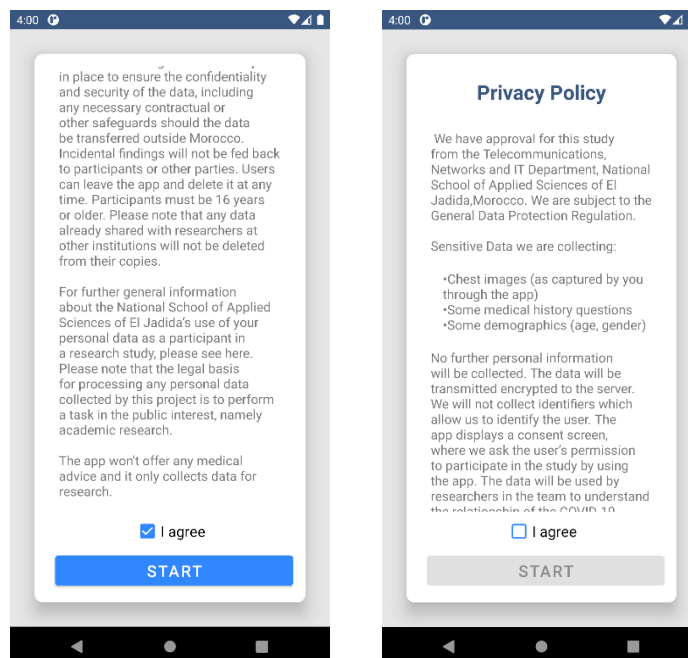
**Figure 11 :** *Page de chargement*

La **Figure 12** suivante montre la page d'acceptation de politique de confidentialité pour pouvoir utiliser notre application.



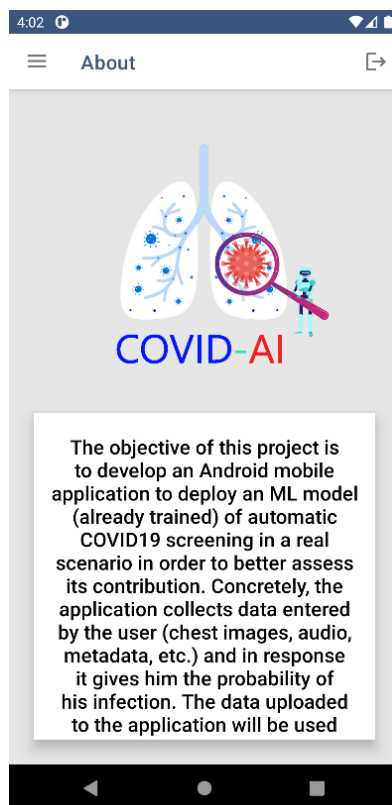
**Figure 12 :** *Page de politique de confidentialité*

La **Figure 13** suivante montre la page d'accueil qui contient le menu de navigation et la page de self-screening après l'utilisateur la choisit parmi les éléments du menu de navigation.



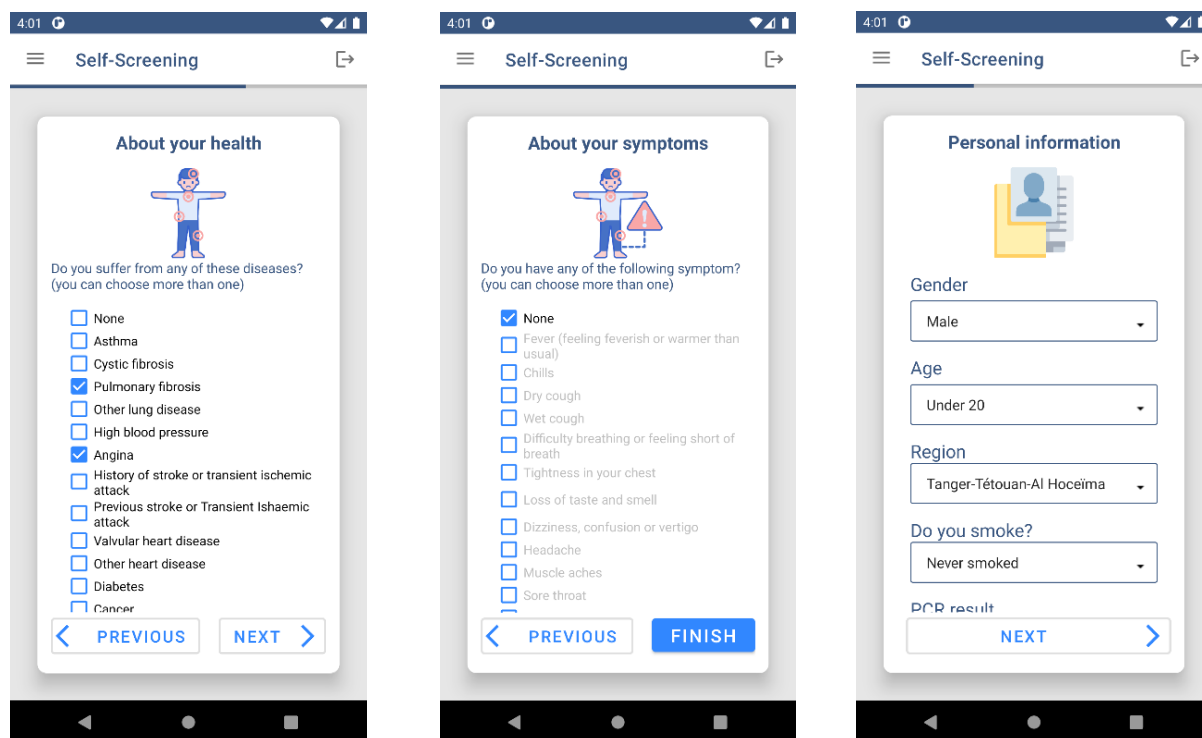
**Figure 13 :** *Page d'accueil*

La **Figure 14** suivante montre la page d'à propos de qui contient un aperçu général à propos de notre projet



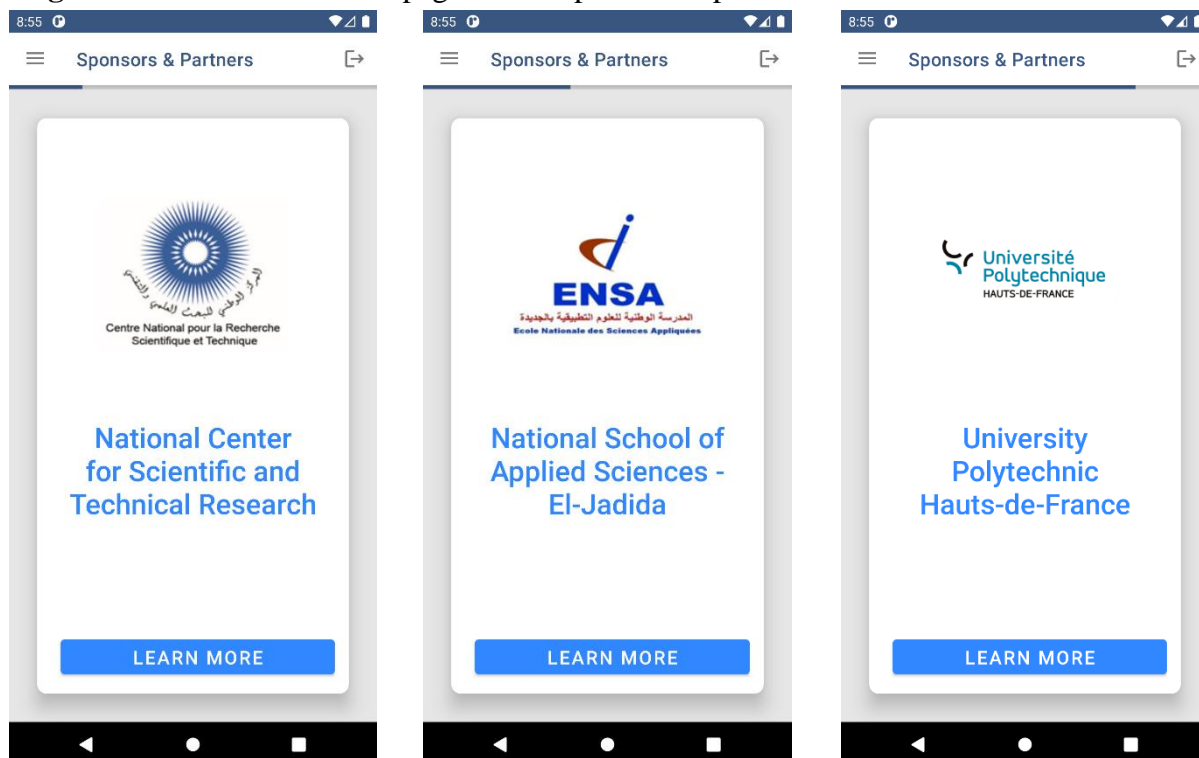
**Figure 14 :** Page A propos de

La **Figure 15** suivante montre les pages dédiés pour la collecte de données.



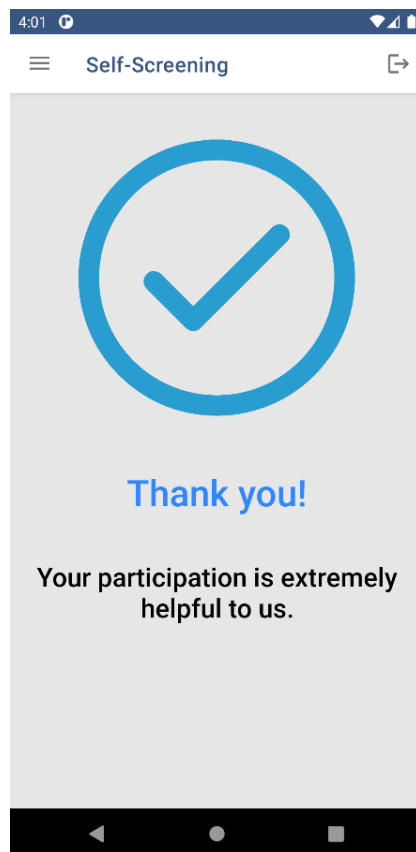
**Figure 15 :** Pages dédiés pour la collecte de données

La **Figure 16** suivante montre la page de nos sponsors et partenaires.



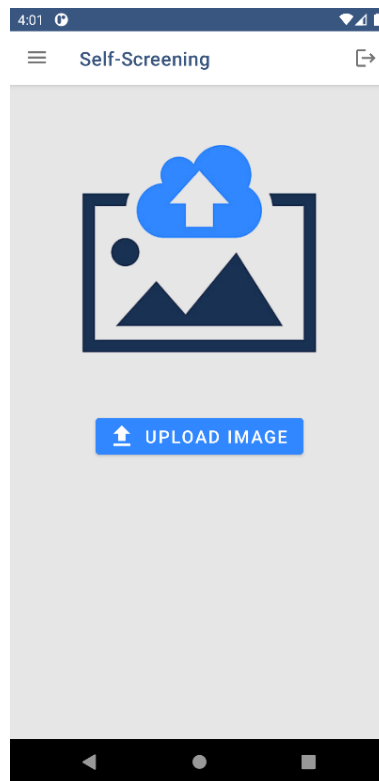
**Figure 16 :** *Pages de sponsors et partenaires*

La **Figure 17** suivante montre la page de remerciement pour la participation dans l'enquête.



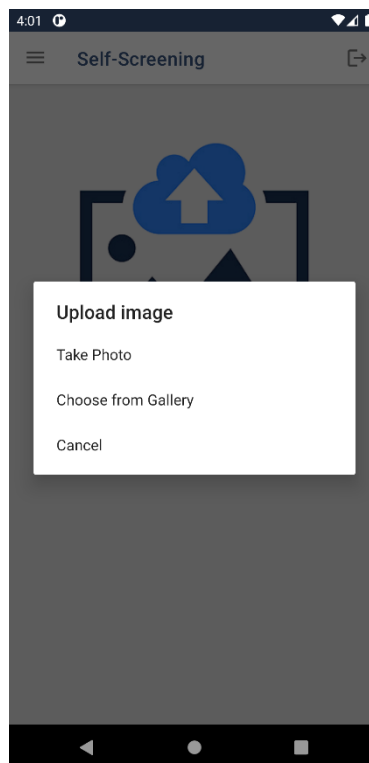
**Figure 17 :** *Page de remerciement pour la participation dans l'enquête*

La **Figure 18** suivante montre la page de prise de l'image. En effet, on peut charger l'image depuis la galerie ou prendre image à l'aide du camera.



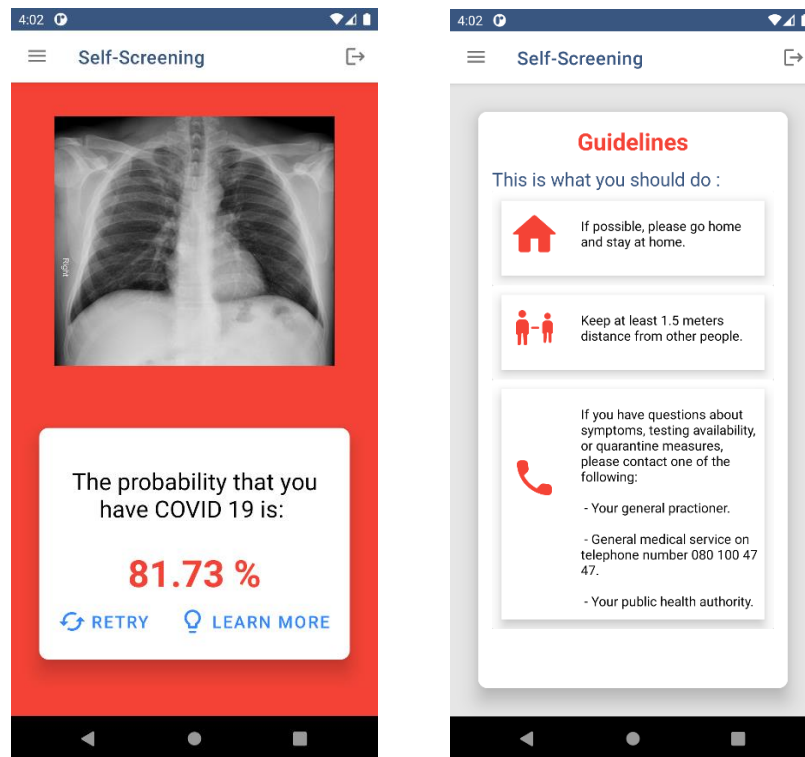
**Figure 18 :** *Page de prise de l'image*

La **Figure 19** suivante montre la page de choix de méthode de prise de l'image.



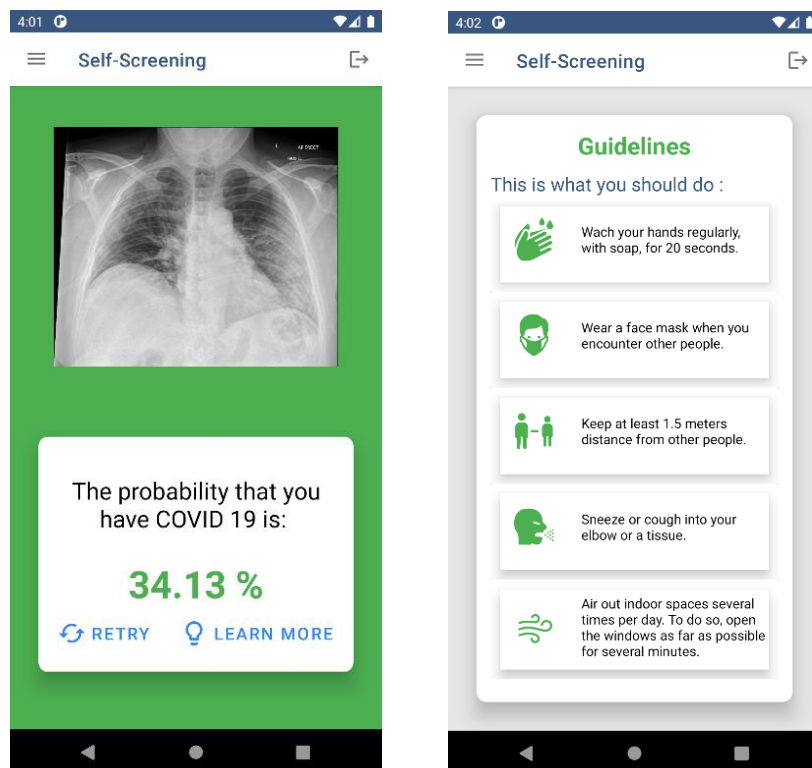
**Figure 19 :** *Page de choix de méthode de prise de l'image*

La **Figure 20** suivante montre la page de résultat du dépistage et la page de directives et conseils pour l'utilisateur dans le cas où il/elle n'est pas malade.



**Figure 20 :** *Résultat du dépistage et la page de directives et conseils : cas non malade*

La **Figure 21** suivante montre la page de résultat du dépistage et la page de directives et conseils pour l'utilisateur dans le cas où il/elle est malade.



**Figure 21 :** *Résultat du dépistage et la page de directives et conseils : cas malade*

## Conclusion

Dans ce chapitre, nous avons présenté les différents outils qu'on a exploité pour en fin de compte mener à bien notre projet, aussi nous avons eu l'occasion de présenter les diverses interfaces que comporte notre application mobile.

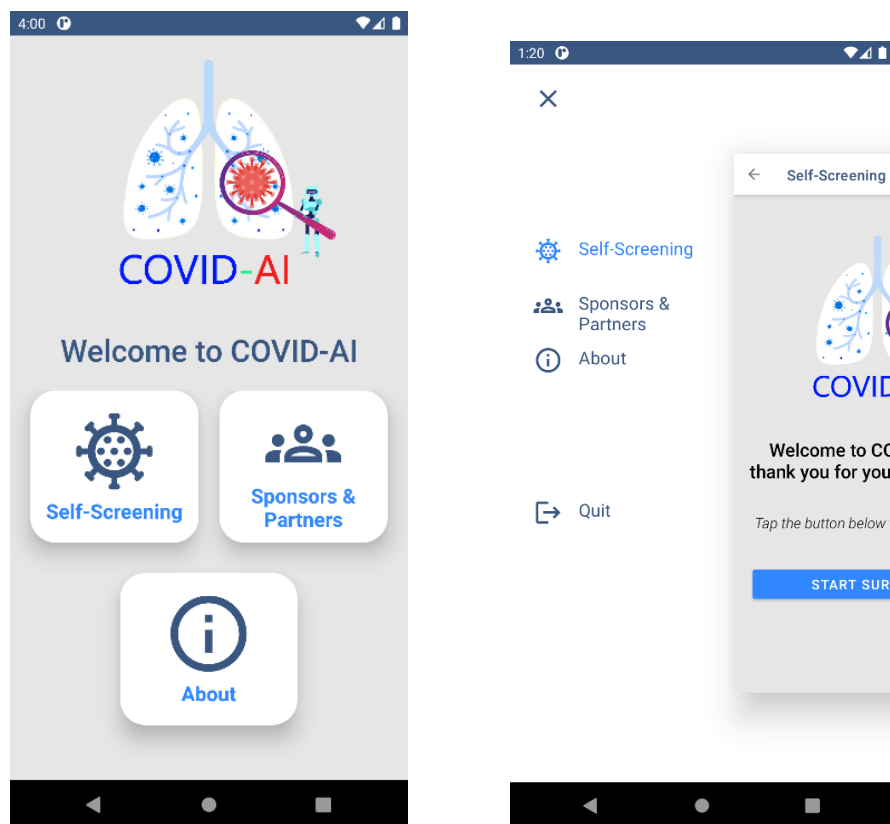
# 5 Test et Evaluation

## Introduction

Cette session est dédiée aux test et l'évaluation. Ce qui nous permettra de connaître les points forts et faibles de notre application, les exigences pour son utilisation et ses performances sur les différentes plateformes. D'une autre part, les résultats des enregistrements vont y être analysés en fonction de leur qualité, et de leur quantité. Les résultats obtenus après ces tests vont déterminer, les améliorations et changements qui doivent être effectués au niveau de l'application.

### 5.1 Test d'utilisabilité :

Le contenu des pages est-il placé de sorte que les utilisateurs puissent se diriger facilement vers les fonctions les plus importantes ? Nous avons essayé de créer des interfaces simple et compréhensible par n'importe quelle catégorie d'utilisateur, comme par exemple les interfaces de navigation présentées dans la **Figure 22**.

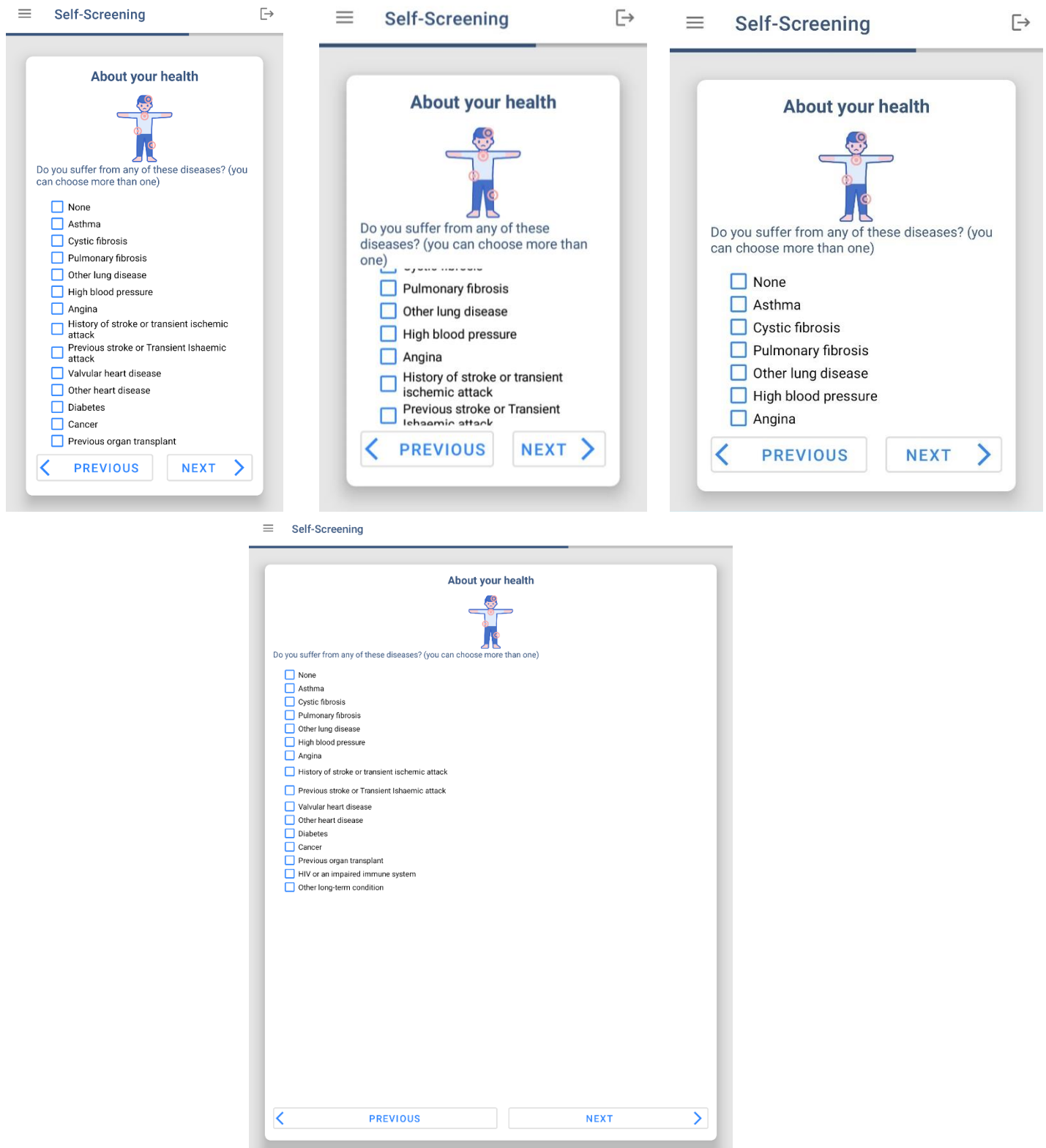


**Figure 22 : Test d'utilisabilité**



## 5.2 Test de compatibilité :

Notre application est compatible avec différentes versions Android ainsi pour l’affichage avec les différentes tailles d’écran et résolution comme le montre la **Figure 23**.



**Figure 23 :** *Test de compatibilité*

## 5.3 Test de la base de données :

Ce test consiste à vérifier les données enregistrées dans la base et finalement la conformité des données avec ceux affichés dans l'application comme montre dans la **Figure 24** et la **Figure 25**.

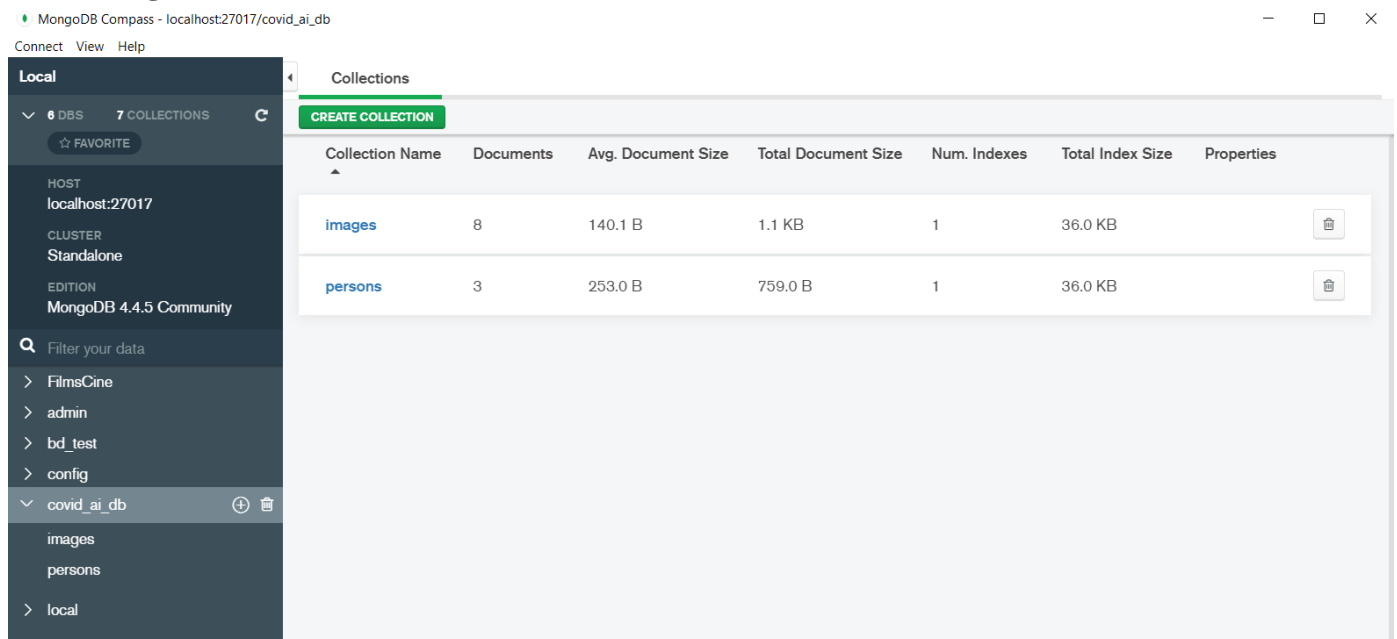


Figure 24 : Notre base de données MongoDB

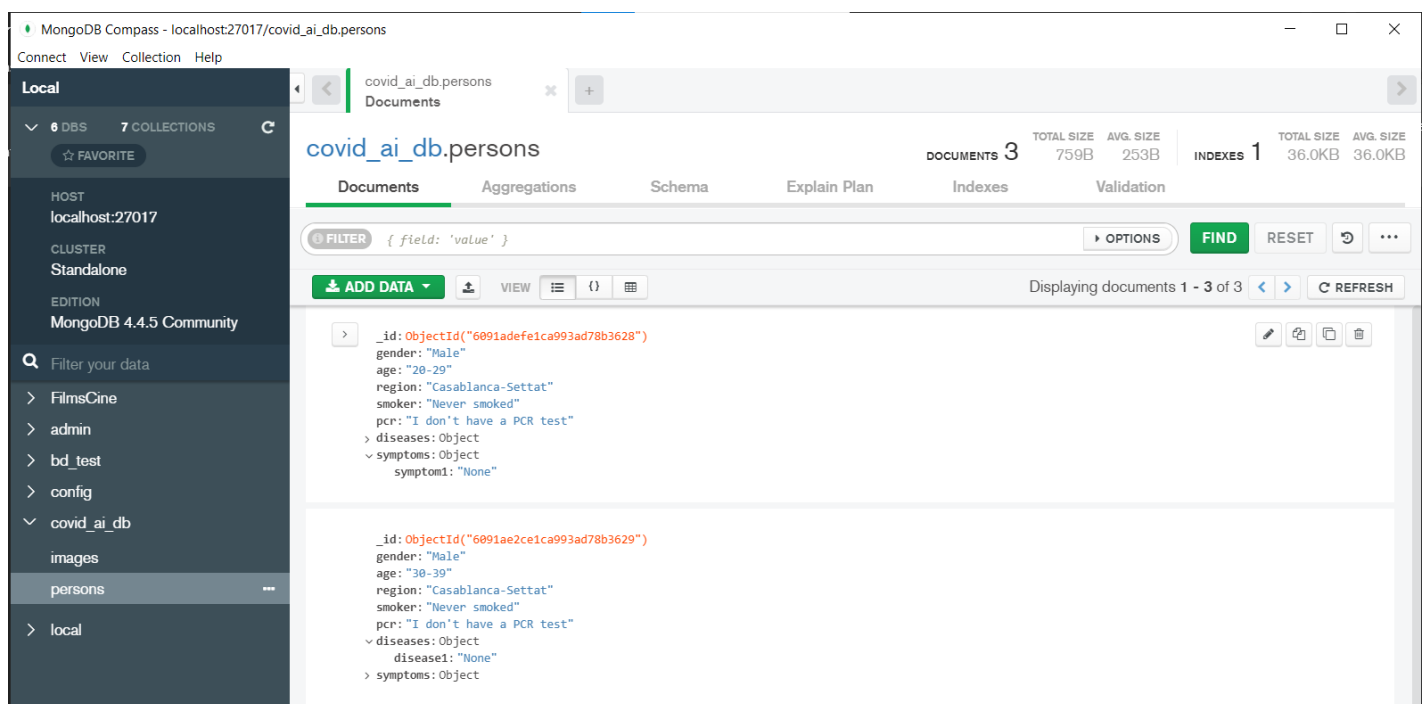


Figure 25 : Test de la base de données

## 5.4 Test de montée en charge :

La **Figure 26** montre le résultat du test d'insertion de 100000 requêtes dans un temps restreint et nous avons constaté que MongoDB a pu gérer plus de 3707 opérations simultanément durant 5 secondes.



**Figure 26 :** *Test de montée en charge*

## Conclusion

D'après les résultats des tests effectués dans le cadre de ce projet, et bien sûr après avoir remédié aux problèmes et bugs rencontrés, nous pouvons dire que l'application répond au cahier des charges, et respecte les règles de l'IHM (l'ergonomie, l'interaction), ainsi que la sécurité et la normalisation de la base de données sans oublier UX (expérience de l'utilisateur), même si nous avons plusieurs tâches à faire afin d'avoir une application idéale et parfaite, et cela sera l'objet du chapitre suivant.

# 6

## Conclusion et perspectives

Dans ce rapport, nous avons présenté les étapes de conception et de développement de notre application mobile Android pour déployer un modèle ML (déjà entraîné) de dépistage automatique du COVID19 dans un scénario réel afin de mieux évaluer son apport. C'est un exemple concret de l'industrialisation et l'expérimentation d'un modèle ML.

Notre travail s'est déroulé sur cinq phases. Nous avons commencé par une étude de l'existant, suivie de la proposition d'une solution adéquate.

Dans la deuxième phase, nous avons spécifié les besoins fonctionnels et non fonctionnels que doit respecter l'application suivis par un diagramme des cas d'utilisation mettant en jeu les acteurs qui interagissent avec le système.

La troisième phase concernait la conception et l'architecture de notre application en utilisant des diagrammes de classes et de séquences.

Dans la quatrième phase, nous avons évoqué les différentes technologies utilisées ainsi que l'implémentation de notre système en présentant les différentes interfaces.

Dans la dernière phase, nous avons bouclé les différents tests de génie logiciel pour notre application.

Ce projet se situe en effet, dans le cadre du projet de fin de la deuxième année du cycle ingénieur en informatique et technologies émergentes. Ce projet était une véritable expérience de travail en collaboration, qui nous a permis de bien gérer la répartition des tâches et de renforcer l'esprit de partage de connaissances ainsi que la synchronisation de notre travail et de mettre en pratique nos compétences en termes de développement mobile vu dans le premier semestre de cette année ainsi que, les applications de l'apprentissage automatique.

Cependant, il nous reste dans nos perspectives d'ajouter la fonctionnalité d'effectuer le dépistage en local en absence de connexion Internet et une fois la connexion retourne on synchronise les données dans le serveur. Ainsi, pourquoi pas notre application devienne multi-plateformes.

# Bibliographie

- [1] : Fiabilité du test PCR, [https://www.metrology-journal.org/articles/ijmqe/full\\_html/2020/01/ijmqe200027/ijmqe200027.html](https://www.metrology-journal.org/articles/ijmqe/full_html/2020/01/ijmqe200027/ijmqe200027.html), consulté le 17.04.2021.
- [2] : Cambridge Sound App, <https://www.covid-19-sounds.org/fr/> ,consulté le 20.04.2021.
- [3] : Projet MIT OpenSigma, <https://opensigma.mit.edu/> ,consulté le 20.04.2021.
- [4] : Projet Coswara, <https://record.coswara.iisc.ac.in/?locale=en-US>,consulté le 20.04.2021
- [5] : Android Studio, <https://developer.android.com/about>,consulté le 03.05.2021.
- [6] : Flask, <https://flask.palletsprojects.com/en/2.0.x/> ,consulté le 02.05.2021.
- [7] : TensorFlow Lite, <https://www.tensorflow.org/lite> ,consulté le 03.05.2021.
- [8] : Comparaison entre MySQL et MongoDB,  
[https://www.researchgate.net/publication/308672555\\_A\\_comparative\\_study\\_MongoDB\\_vs\\_MySQL](https://www.researchgate.net/publication/308672555_A_comparative_study_MongoDB_vs_MySQL) ,consulté le 17.05.2021.