République Algérienne Démocratique et Populaire

Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj

Faculté des Mathématiques et d'Informatique
Département d'informatique

MÉMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : **Technologie de l'Information et de la Communication**

THÈME

Système de gestion d'un laboratoire d'analyses médicales (mobile & web) basé sur une architecture Microservices

-SMARTLAB-

Réalisé par :

ATTIA Nour El Islam

SAOUDI Anis

***Soutenu publiquement le* : / 09/ 2023**

Devant le jury composé de:

Président

Examinateur

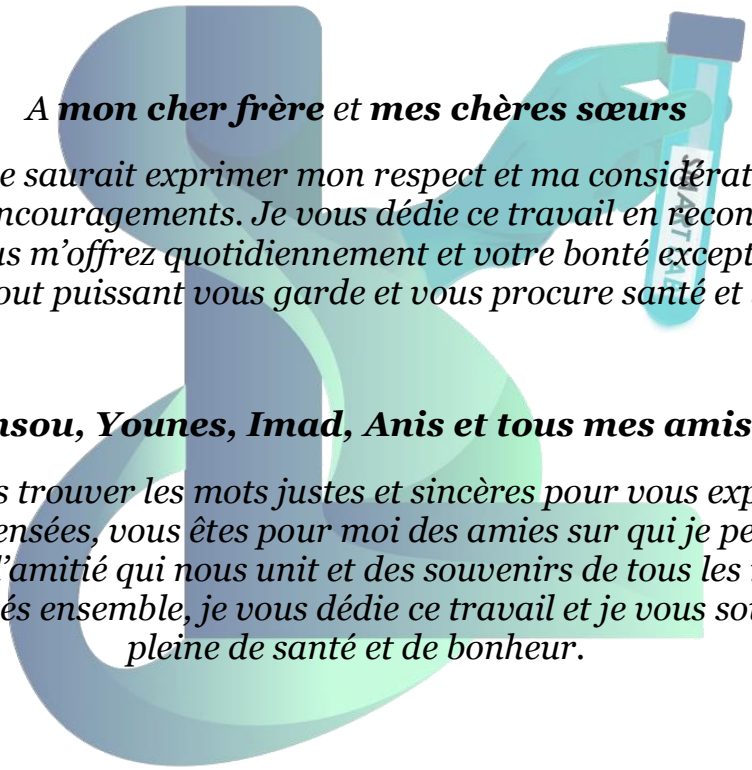
Encadreur **Dr. SAIDANI Kaouther** MCB à L'U. Mohamed El Bachir El Ibrahimi- BBA

Promotion : 2022/2023

Dédicaces

Je dédie cet humble travail

A ma source de bonheur ***mes chers parents***
Vous avez toujours été pour moi un exemple des parents respectueux et honnêtes. Grâce à vous, j'ai appris le sens du travail et de la responsabilité. Je voudrais vous remercier pour votre amour, votre générosité et votre compréhension… Votre soutien fut une lumière dans tout mon parcours. Aucune dédicace ne saurait exprimer l'amour, l'estime et le respect que j'ai toujours eu pour vous.
Ce modeste travail est le fruit de tous les sacrifices que vous avez déployé pour mon éducation et ma formation. Je vous aime mes chers parents et je vous souhaite une bonne santé et une vie longue, heureuse et pleine de joie.



A ***mon cher frère*** *et* ***mes chères sœurs***

Aucun langage ne saurait exprimer mon respect et ma considération pour votre soutien et vos encouragements. Je vous dédie ce travail en reconnaissance de l'amour que vous m'offrez quotidiennement et votre bonté exceptionnelle. Que ALLAH le tout puissant vous garde et vous procure santé et bonheur.

A mes amis ***Chamsou, Younes, Imad, Anis et tous mes amis de FG School***

Je ne peux pas trouver les mots justes et sincères pour vous exprimer mon affection et mes pensées, vous êtes pour moi des amies sur qui je peux compter. En témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passés ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur.

A TOUTE MA FAMILLE

Ceux qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail, vous m'avez chaleureusement supporté et encouragé tout au long de mon parcours.

ATTIA Nour El Islam

Dédicaces

Je dédie mon travail

A ma très chère mère

Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Votre affection me couvre, votre bienveillance me guide et votre présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.

A mon cher frère

Tu as toujours été à mes cotés pour me soutenir et m'encourager.

Que ce travail traduit ma gratitude et mon affection.

Puisse ALLAH vous donne santé, bonheur et une vie pleine de joie.

Je dédie ce travail également à toute personne qui m'a soutenu

Et motivé durant mon parcours universitaire, à tous mes amis qui ont cru en moi

Je vous remercie un par un

SAOUDI Anis

Remerciements

*Nous tenons tout d'abord à remercier **ALLAH** le tout puissant, qui nous a donné la force, le courage et la patience d'accomplir ce modeste travail.*

Nous tenons à exprimer nos sincères remerciements à tous les ***Professeurs*** *qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.*

*En second lieu, nous tenons à remercier plus particulièrement Mme **SAIDANI Kaouther** pour son orientation, sa confiance, sa patience et sa supervision tout au long de notre mémoire, pour les nombreux conseils qu'elle nous a prodigués, ainsi que pour le temps qu'elle a passé avec nous afin que ce mémoire soit une réussite.*

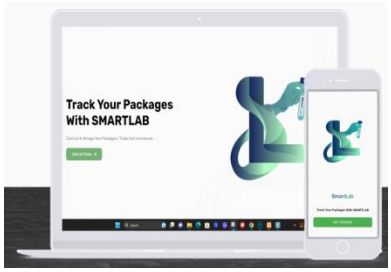
*Merci à l'**équipe pédagogique** de notre faculté des mathématiques et d'informatique pour avoir répondu à nos questions tout au long de ces cinq années.*

*Nos vifs remerciements vont également aux **Membres du Jury** pour l'intérêt qu'ils ont porté à notre projet en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.*

*Enfin, nous souhaitons remercier du fond du cœur nos **familles** et nos **amis**. Ce mémoire est le fruit de cinq années d'étude.*

*Nos plus grandes reconnaissances s'adressent à nos **parents** pour leur aide et leurs conseils, mais surtout pour leur soutien et leur confiance depuis toujours.*

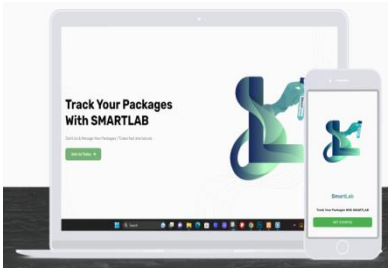
Résumé



Les ordinateurs et les applications mobiles sont deux éléments technologiques qui ont profondément transformé la façon dont nous vivons, travaillons et interagissons dans notre société contemporaine. Ils ont révolutionné la manière dont nous accédons à l'information, communiquons et accomplissons nos tâches quotidiennes. Ce projet consiste à développer une application mobile & web de gestion d'un laboratoire d'analyses médicales sous Android et iOS basé sur une architecture microservice, dédiée aux médecins/managers et employés du laboratoire médicale, en utilisant une architecture microservice. Cette application mobile & web nommée « SMARTLAB » conçu pour gérer les sous-traitances (Paquets/ Tubes) d'une manière simple, facile, efficace, sécurisé et aussi de garantir la bonne communication et interaction entre les laboratoires d'analyses médicales et entre les managers et les employés. Afin de réaliser notre projet nous avons utilisé plusieurs outils et lagunages de programmation, citons : UML, Spring Boot (JAVA), Flutter, Dart, Html, CSS, Java Script, Docker….

Mots clé: médecin, manager, laboratoire d'analyse médicale, architecture microservice, Android, iOS, UML, Flutter, JAVA, Dart, Html, CSS, Java Script, Docker.

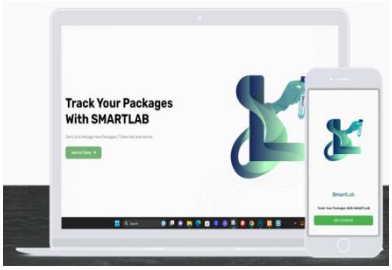
Abstract



Computers and mobile applications are two technological elements that have profoundly transformed the way we live, work and interact in our contemporary society. They have revolutionized the way we access information, communicate and perform our daily tasks. This project consists in developing a mobile & web application for the management of medical analysis laboratories under Android and iOS, dedicated to doctors/managers and employees of the medical laboratory, using a micro service architecture. This mobile & web application called "SMARTLAB" designed to manage subcontracts (Packages/Tubes) in a simple, easy, efficient, secure way and also to guarantee good communication and interaction between medical analysis laboratories and between managers and employees. In order to realize our project, we used several tools and programming language, including: UML, Spring Boot (JAVA), Flutter, Dart, Html, CSS, Java Script, Docker….

Keywords: doctor, manager, medical analysis laboratory, microservice architecture, Android, iOS, UML, Flutter, JAVA, Dart, Html, CSS, Java Script, Docker

الملخص



تعد أجهزة الكمبيوتر و تطبيقات الهاتف المحمول عنصرين تكنولوجيين أحدثا تغييرًا عميقًا في الطريقة التي نعيش ونعمل ونتفاعل بها في مجتمعنا المعاصر. لقد أحدثوا ثورة في طريقة وصولنا إلى المعلومات والتواصل وأداء مهامنا اليومية. يهدف مشروعنا إلى تطوير تطبيق ويب وتطبيق جوال لإدارة مختبرات التحاليل الطبية مخصص للأطباء/ المديرين والموظفين في المختبرالطبي، باستخدام بنية الخدمة الصغيرة. يتيح هذا التطبيق المسمَى "SMARTLAB" بإدارة الحزم / الأنابيب بطريقة بسيطة وسهلة وفعالة وآمنة وأيضًا لضمان التواصل والتفاعل الجيد بين مختبرات التحاليل الطبية وبين المديرين والموظفين. من أجل تنفيذ مشروعنا، استخدمنا العديد من أدوات ولغات البرمجة، بما في ذلك:UML ،Spring Boot (JAVA)، Flutter، Dart، Html، CSS، Java Script، Docker ....

الكلمات المفتاحية : طبيب، مسؤول، مخبر التحاليل الطبية، بنية خدمات مصغرة، UML، Spring Boot (JAVA)، Flutter، Dart، Html، CSS،Java Script، Docker ....

Table des matières

Chapitre 2 Etat de l'art

Chapitre 3 Architecture et Modélisation

Chapitre 4 Implémentation

Liste des abréviations

API Application Programing Interface

BFF Backend For Frontend

REST Representational State Transfer

HTTP Hypertext Transfer Protocol

NIST National Institute of Standards and Technology

SAAS Software as a Service

PAAS Platform as a Service

IAAS Infrastructure as a Service

VMM Virtual Machine Monitor

RAM Random Access Memory

PCR Polymerase Chain Reaction

LIMS Laboratory Information Management System

PDF Portable Document Format

UML Unified Modeling Language

OMG Object Management Group

CU Cas d'Utilisation

SDK Software Development Kit

JVM Java Virtual Machine

IDP Identity Provider

SAML Security Assertion Markup Language

SQL Structured Query Language

SGBD Système de Gestion de Bases de Données

NOSQL Not OnlyStructured Query Language

DB Data Base

JSON JavaScript Object Notation

IDE Integrated DevelopmentEnvironment

Liste des Figures

Liste des Tableaux

Chapitre 1

Introduction générale et problématique

1.1 Introduction

Ce chapitre présente le contexte général dans lequel s'inscrit notre travail 'Système de gestion d'un laboratoire d'analyses médicales (mobile&web) basé sur une architecture microservices'. Dans un premier temps nous allons présenter la notion de l'architecture microservices et le Cloud Computing. Par la suite nous allons aborder le rôle des applications mobiles & web pour gérer un laboratoire d'analyses médicales auquel nous nous intéressons. Par la suite, nous définirons la problématique et les objectifs de notre projet. Enfin nous décrirons notre contribution et notre plan du mémoire.

1.2 Contexte

Notre travail rentre dans le cadre d'un projet de développement d'une application mobile& web nommée 'SMARTLAB' qui a pour rôle d'assurer une gestion plus efficace au sein des laboratoires d'analyses médicales et de garantir une meilleurs collaboration entre les différents laboratoires voisins. L'objectif serait donc, de permettre aux employés et aux gérants des laboratoires de prendre en charge rapidement et efficacement les résultats des analyses médicales au niveau du laboratoire et de surveiller et suivre l'état des tubes de prélèvement destinés à la sous-traitance inter-laboratoires à tout moment, en tous lieux et avec facilité et confiance.

A cet effet, deux domaines seront considérés et étudiés:

- L'architecture microservices et le Cloud Computing.
- Le rôle des applications mobiles & web dans la gestion des laboratoires d'analyses médicales.

1.2.1 L'architecture microservices et le Cloud Computing

1.2.1.1 L'architecture monolithique

L'architecture monolithique est un style d'architecture logicielle traditionnel dans lequel tous les composants d'une application sont combinés en une seule unité étroitement couplée. Dans une architecture monolithique, l'application est construite comme un seul et grand exécutable, tous les composants étant étroitement intégrés et dépendants les uns des autres [1].

Dans ce type d'architecture étroitement intégrée, chaque composant et ceux qui lui sont associés doivent être présents pour permettre l'exécution ou la compilation du code [1].

1.2.1.1.1 Avantages et inconvénients de l'architecture monolithique

a) Les avantages

- *Déploiement plus facile* : les applications monolithiques nécessitent moins d'étapes de déploiement, car l'ensemble de la solution est regroupé dans une seule unité. Ainsi, les processus de déploiement ont tendance à être plus simples et plus rapides avec des applications monolithiques.
- *Facile à développer* : l'architecture monolithique est relativement simple à développer, car tous les composants de l'application sont combinés en une seule unité. Cela permet aux développeurs de comprendre plus facilement le code, ainsi que d'écrire, tester et déboguer l'application.
- *Performance améliorée* : les applications monolithiques peuvent offrir de meilleures performances en raison de l'absence de surcharge de communication interservices. De plus, aucune latence supplémentaire n'est introduite par plusieurs services communiquant sur le réseau, ce qui améliore davantage les performances.
- *Complexité réduite* : l'architecture monolithique peut contribuer à réduire la complexité, car il y a moins de composants à gérer et moins de dépendances à gérer. Cela peut faciliter la maintenance de l'application et garantir qu'elle continue de fonctionner correctement au fil du temps [2].

b) Les inconvénients

Lorsque l'application grandit en complexité et en taille, une architecture monolithique peut rencontrer certaines difficultés :

- *Difficulté de maintenance* : la maintenance d'une base de code monolithique devient plus difficile à mesure que l'application grandit en complexité et en taille. Cette difficulté est due au couplage étroit des composants, ce qui rend plus difficile pour les développeurs de modifier ou de déboguer l'application sans impacter les autres parties.

- *Scalabilité limitée* : la mise à l'échelle d'une application monolithique peut devenir difficile, car l'ensemble de l'application doit être mis à l'échelle ensemble plutôt que de ne mettre à l'échelle que les parties nécessaires. Ce manque de flexibilité augmente souvent les coûts et réduit l'efficacité lors de la manipulation de charges élevées.
- *Risque d'un point de défaillance unique* : dans une architecture monolithique, si un composant tombe en panne, l'ensemble de l'application peut devenir non fonctionnel. Ce risque pose des défis importants pour garantir une haute disponibilité et une tolérance aux pannes pour les applications critiques.
- *Flexibilité* : l'architecture monolithique peut être moins flexible que d'autres approches, car tous les composants de l'application sont combinés en une seule unité. Cela peut rendre plus difficile l'adaptation aux exigences changeantes ou l'ajout de nouvelles fonctionnalités [2].

1.2.1.2 L'architecture microservices

Il existe un certain nombre de définitions des microservices. La plus commune et généralisée reste celle de *Martin Fowler* qui dit: «Le style architectural des microservices est une approche permettant de développer une application unique sous la forme d'une suite logicielle intégrant plusieurs services. Ces services sont construits autour des capacités de l'entreprise et peuvent être déployés de façon indépendante» [3].

Concrètement, les microservices sont une méthode de développement de logiciel utilisée pour concevoir une application comme un ensemble de services modulaires. Chaque module répond à un objectif métier spécifique et communique avec les autres modules. Pour Fowler, les microservices doivent nécessiter le strict minimum en termes de gestion centralisée des services et peuvent être créés sous différents langages de programmations [3].

L'architecture microservices désigne un style d'architecture utilisé dans le développement d'applications. Elle permet de décomposer une application volumineuse en composants indépendants, chaque élément ayant ses propres responsabilités. Pour diffuser la requête d'un utilisateur unique, une application basée sur des microservices peut appeler plusieurs microservices internes pour

composer sa réponse [3]. Elle désigne aussi un style particulier de conception des applications logicielles. Contrairement aux styles architecturaux traditionnels qui visent à construire un logiciel comme une seule et unique unité, l'architecture microservices repose sur une approche modulaire. Ainsi, les applications sont conçues comme des suites comprenant plusieurs microservices qui peuvent être déployés et gérés indépendamment [3].

Une architecture de microservices se différencie d'une approche monolithique classique par le fait qu'elle décompose une application pour en isoler les fonctions clés. Chacune de ces fonctions est appelée « service » et ces services peuvent être développés et déployés indépendamment les uns des autres. Ainsi, chacun peut fonctionner (ou dysfonctionner) sans affecter les autres [4].

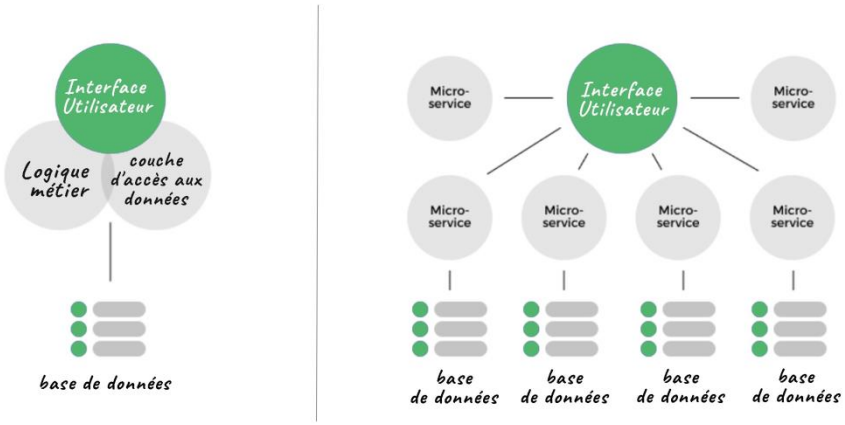


FIGURE 1-1- Différence entre l'architecture monolithique et l'architecture microservices [4]

1.2.1.2.1 Avantages de l'architecture microservices

L'architecture microservices a émergé pour répondre aux besoins croissants des entreprises en matière de systèmes logiciels de plus en plus complexes, évolutifs, résilients et flexibles [5]:

- *Scalabilité et évolutivité* : les microservices peuvent être redimensionnés indépendamment les uns les autres. Cela implique l'ajustement de la capacité de chaque service en fonction de l'évolution de la charge de travail ainsi que les ressources allouées au développement et à la maintenance qui

peuvent être ajustées de manière indépendante en fonction de l'évolution des besoins métiers.

- *Flexibilité technologique* : chaque microservice peut être développé, testé et déployé en utilisant des langages de programmation et des technologies différentes, ce qui permet de choisir les outils et les processus les mieux adaptés pour chaque service (langage de programmation et framework, modèle de base de données, techniques et approches DevOps, etc.).
- *Réduction des pannes* : les microservices étant indépendants, lorsqu'un élément tombe en panne ou rencontre un problème, l'ensemble de l'application ne cesse pas de fonctionner contrairement aux applications monolithiques. Il est également plus facile d'identifier et de résoudre une panne dans ce type d'écosystème.
- *Résilience* : les microservices doivent être conçus de manière à ce que les erreurs ou les problèmes de performance affectant l'un d'entre eux ne se propagent pas aux autres services, garantissant ainsi une meilleure résilience globale du système [5].

1.2.1.2.2 Choix de l'architecture microservices

L'architecture microservices a été inventée pour résoudre certaines difficultés causées par les gros projets car avec le temps les projets informatiques ont tendance à grossir et petit à petit on étend les fonctionnalités existantes ; avec de nombreux rajouts et peu de suppressions, on termine avec un mille feuilles fonctionnel compliqués et difficilement gérable. Quand la quantité de code augmente, sa maintenance devient de plus en plus complexe. Même avec une architecture logicielle solide, les interdépendances entre les différentes briques augmentent avec le temps [6].

1.2.1.2.3 Fonctionnement de l'architecture microservices

Les microservices permettent de diviser les grandes applications en éléments plus petits fonctionnant de manière indépendante. Chaque « élément » a ses responsabilités et peut les assumer indépendamment de ce que font les autres composants. Une application basée sur des microservices fait appel aux services collectifs de ces éléments pour répondre aux demandes des utilisateurs. Les services

d'une architecture de microservices « communiquent » entre eux à l'aide d'interfaces de programmation d'applications (API) légères qui se connectent à des interfaces détaillées. Ces services sont créés pour exécuter des fonctions commerciales spécifiques, telles que les transactions monétaires, la création de factures et le traitement des données. Chaque service effectue une seule opération. Comme ils fonctionnent de manière indépendante, les services peuvent être déployés, mis à jour et mis à l'échelle en fonction de la demande pour leurs fonctions spécifiques. La mise en œuvre d'une architecture de microservices aboutit à la création de systèmes métier flexibles et évolutifs [7].

1.2.1.2.4 La transition vers une architecture microservices

Nombreuses sont les raisons de préférer les microservices à une architecture monolithe pour une application. La première d'entre elles est la caractéristique fondatrice de cette architecture : l'indépendance. En effet, un microservice est un service autonome. Il tourne sur son propre processus et s'adresse à un domaine métier spécifique. Ce « couplage faible » permet notamment de limiter le risque de régression grâce à la présence de bases de code et de repository (stockage de toutes les modifications apportées) propres à chaque microservice, ce qui rend cette architecture simple à maintenir [8].

Les avantages et les enjeux de la transition vers une architecture microservices peuvent être cités comme suit :

- *Compétitivité* : l'architecture microservices autorise par ailleurs une forte scalabilité de l'infrastructure cloud. Avantage non négligeable, puisque cela permet d'augmenter les ressources pour faire face aux montées de charge ponctuelles. C'est exactement ce qu'il faut si l'application considérée est, par exemple, mise en lumière à une heure de grande écoute lors d'un reportage télévisé. Avec une architecture monolithe, un nombre de connexions élevé risquerait de saturer l'application sans possibilité de réagir. L'expérience client est mauvaise, l'image de l'entreprise ternie. Les microservices permettent, eux, une flexibilité et une évolutivité rapides [8].

- *Optimisation des coûts* : le passage à une architecture microservices permet de réaliser des économies de coût, liées à la consommation de ressources sur le cloud. En effet, avec cette architecture, un service peut être scalé indépendamment des autres, seules les ressources nécessaires à cette modification seront donc utilisées. À l'inverse, avec les applications monolithes, l'approvisionnement des ressources s'effectue sur tout le monolithe. Cela, même si un seul service exige d'être renforcé [8].
- *Résilience* : la résilience apportée par les microservices est un de leurs avantages majeurs. En effet, l'autonomie des services réduit le couplage entre les « services by design » (service par conception), ce qui permet leur continuité en mode dégradé. Il est pour autant nécessaire que les patterns mis en place l'autorisent (le « circuit breaker », par exemple, limite les risques de régressions liés au développement et à la configuration) [8].

1.2.1.2.5 Caractéristiques de l'architecture microservice

Voici quelques-unes des principales caractéristiques de cette architecture :

- *Modularité* : les applications sont décomposées en services autonomes qui peuvent être développés, déployés et mis à l'échelle indépendamment les uns des autres.
- *Indépendance* : chaque service peut être développé dans un langage de programmation différent, avec des bibliothèques et des outils différents, et peut être déployé sur des serveurs différents.
- *Scalabilité* : en utilisant des services indépendants, il est facile de mettre à l'échelle horizontalement chaque service en fonction de ses besoins spécifiques, sans affecter les autres services.
- *Facilité de déploiement* : les services peuvent être déployés indépendamment les uns des autres, ce qui permet des déploiements plus rapides et plus fiables.
- *Facilité de maintenance* : les services étant de petite taille, leur maintenance est plus facile et rapide, ce qui réduit les coûts de maintenance globaux de l'application [9].

1.2.1.2.6 Les composants de l'architecture microservices

L'architecture microservices est composée de plusieurs éléments clés qui permettent la création et le fonctionnement de services indépendants les uns des autres :

- *Services* : les services sont les blocs de construction fondamentaux de l'architecture microservices. Chaque service est conçu pour effectuer une tâche spécifique et est autonome. Ils peuvent communiquer avec d'autres services via des API.
- *API Gateway* : l'API Gateway est un point d'entrée unique qui permet d'exposer les services au monde extérieur. Il permet également de gérer les requêtes, l'authentification et l'autorisation des utilisateurs, ainsi que la mise en cache des données pour améliorer les performances.
- *Bases de données* : dans une architecture microservices, chaque service peut avoir sa propre base de données. Cela permet d'optimiser les performances et d'améliorer la scalabilité.
- *Service Registry* : le Service Registry est une base de données centrale qui enregistre les informations sur tous les services disponibles dans le système. Il permet de découvrir et de localiser les services.
- *Circuit Breaker* : le Circuit Breaker est un composant qui permet de gérer les erreurs et les pannes. Il détecte les erreurs et les pannes potentielles, puis active un mode de secours pour éviter une dégradation du système.
- *Outils de déploiement* : les outils de déploiement automatisent le processus de déploiement des services. Ils permettent de déployer les services de manière indépendante et de gérer les mises à jour et les révisions.
- *BFF (Backend for frontend)* : un microservice conçu pour gérer les requêtes et agréger les réponses des différents microservices pour les renvoyer au frontend.
- *Message Broker* : un Message Broker est un service qui facilite la communication entre les microservices [9].

1.2.1.2.7 Communication inter-microservices

Les microservices communiquent entre eux de deux manières : synchrones et asynchrones.

Dans une communication synchrone, l'appelant attend une réponse avant d'envoyer le message suivant, et il fonctionne comme un protocole REST en plus de HTTP. Dans cette approche, le client doit attendre de recevoir une réponse du serveur avant de pouvoir poursuivre sa tâche.

Dans une communication asynchrone, les messages sont envoyés sans attendre de réponse. Cela convient aux systèmes distribués et nécessite généralement un message broker pour gérer les messages. Dans cette approche, le client n'attend pas de réponse et peut poursuivre sa tâche.

Le type de communication qu'on choisit doit tenir compte de différents paramètres, tels que la façon dont on structure nos microservices, quelle infrastructure nous avons en place, la latence, l'échelle, les dépendances et le but de la communication. En général, la communication synchrone est mieux adaptée aux cas d'utilisation qui nécessitent des réponses immédiates, tandis que la communication asynchrone est mieux adaptée aux cas d'utilisation qui peuvent tolérer des retards et nécessitent une évolutivité et une tolérance aux pannes élevées [10].

1.2.1.2.8 Déploiement de microservices

Le déploiement de microservices s'effectue généralement à l'aide de technologies de conteneurisation telles que Docker ou Kubernetes. Chaque microservice est conditionné sous forme de conteneur et déployé séparément, ce qui facilite la gestion et la mise à l'échelle des composants individuels de l'application.

Ces conteneurs sont des paquets qui contiennent tout ce dont un programme a besoin pour s'exécuter. Une image de conteneur est une unité autonome qui peut s'exécuter sur n'importe quel serveur sans avoir à installer au préalable des dépendances [11].

1.2.1.3 Cloud Computing

Le Cloud Computing fait référence aux applications et aux services qui s'exécutent sur un réseau distribué à l'aide de ressources virtualisées et sont accessibles par des protocoles Internet et des normes de mise en réseau courants. Il se distingue par la notion que les ressources sont virtuelles et illimitées et que les

détails des systèmes physiques sur lesquels le logiciel s'exécute sont abstraits de l'utilisateur. De plus, ces applications et services peuvent être rapidement provisionnées et libérées avec un minimum d'effort de gestion ou d'interaction de fournisseur de services [12].

1.2.1.3.1 Caractéristiques

a) Caractéristiques essentielles

Selon le NIST, Cloud Computing doit posséder cinq caractéristiques essentielles :

- *Accès libre-service et à la demande* : accès libre à tout instant n'importe quelle ressource informatique et celle-ci lui est fournie automatiquement sans interaction humaine entre l'utilisateur et le fournisseur de services.
- *Large accès au réseau* : les ressources sont accessibles sur le réseau depuis des plateformes hétérogènes (tablettes, stations de travail, smartphones, etc.).
- *Mise en commun des ressources* : le fournisseur a rencontré les ressources nécessaires à la disposition de plusieurs clients. Cela est possible en utilisant des outils de virtualisation.
- *Élasticité rapide* : la capacité du Cloud d'adapter (augmenter ou réduire) l'allocation des ressources de façon dynamique, rapide, et efficace, selon les besoins de l'utilisateur.
- *Service mesuré* : la facturation est calculée en fonction de la durée et de la quantité de ressources utilisées. Ainsi, Le Cloud Computing permet au client de payer juste ce qui a été consommé selon le type de service et le temps d'utilisation du service [12].

b) Modèle de déploiement

- *Cloud privé* : c'est un cloud dédié appartenant à l'entreprise et géré par celle-ci. Les services de cloud privé peuvent être hébergés sur site ou par un fournisseur tiers et peuvent offrir des avantages similaires aux services de cloud public, avec l'avantage supplémentaire d'être isolés et dédiés à une organisation ou un utilisateur spécifique, ce qui permet de mieux

contrôler les ressources et d'accroître la sécurité sur les données et l'infrastructure.

- *Cloud public* : l'infrastructure cloud est prévue pour une utilisation ouverte par le grand public. Il peut être détenu, géré et exploité par une entreprise, un universitaire ou un organisme gouvernemental, ou une combinaison de ceux-ci. Il existe chez le fournisseur de cloud.
- *Cloud communautaire* : l'infrastructure cloud est mise à disposition pour une utilisation exclusive par une communauté spécifique de consommateurs issus d'organisations partageant des préoccupations communes (par exemple, mission, exigences de sécurité, politiques et considérations de conformité). Il peut être détenu, géré et exploité par une ou plusieurs organisations de la communauté, un tiers ou une combinaison de celles-ci, et il peut exister sur place ou à l'extérieur.
- *Cloud hybride* : combine l'utilisation de services de cloud public et privé, permettant aux utilisateurs de profiter des avantages des deux modèles. Cette approche permet aux organisations d'optimiser leurs ressources informatiques en utilisant des services de cloud public pour les charges de travail non sensibles, tout en conservant les données et applications critiques dans un environnement de cloud privé sécurisé [13].

c) Modèle de service

- *Software as a Service (SaaS)* : modèle dans lequel les applications logicielles sont hébergées par un fournisseur tiers et mises à la disposition des utilisateurs sur Internet, éliminant ainsi le besoin d'installation et de maintenance locales. Les utilisateurs paient généralement pour les applications SaaS sur la base d'un abonnement.
- *Plateforme as a service (PaaS)* : modèle dans lequel un fournisseur tiers propose une plate-forme et des outils pour le développement, les tests et le déploiement d'applications, éliminant ainsi le besoin pour les utilisateurs de créer et de maintenir leur propre infrastructure. Les utilisateurs paient généralement les services PaaS sur une base de paiement à l'utilisation.

- *Infrastructure as a Service (IaaS)* : modèle dans lequel un fournisseur tiers propose des ressources informatiques virtualisées, telles que des serveurs, du stockage et des réseaux, auxquelles les utilisateurs peuvent accéder et utiliser selon leurs besoins, sans avoir besoin d'acheter et d'entretenir leur propre infrastructure physique. Les utilisateurs paient généralement les services IaaS selon le principe du paiement à l'utilisation [13].

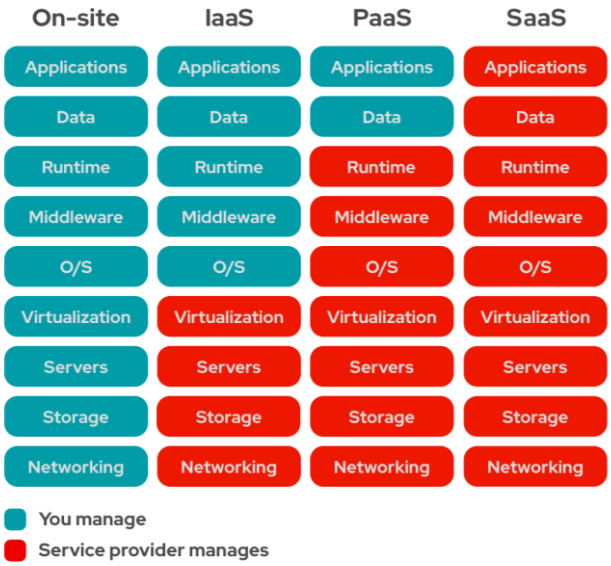


FIGURE 1-2-Les modèles de service du Cloud Computing [14]

1.2.1.3.2 Virtualisation

La virtualisation consiste à créer une représentation virtuelle, basée logicielle, d'un objet ou d'une ressource telle qu'un système d'exploitation, un serveur, un système de stockage ou un réseau. La virtualisation permet à plusieurs systèmes d'exploitation ou applications de s'exécuter sur une seule machine physique, ou à plusieurs ressources physiques d'apparaître comme une seule ressource virtuelle [12].

a) Hyperviseur

Hyperviseur permet à plusieurs systèmes d'exploitation de travailler sur une même machine physique en même temps. Le système d'exploitation installé sur la machine virtuelle s'appelle un SE invité. Une console de gestion d'hyperviseur, également appelée gestionnaire de machine virtuelle (VMM), est un logiciel

permettant de gérer facilement les machines virtuelles. Il existe deux principaux types d'hyperviseurs :

- *L'hyperviseur de type 1*: est également appelé hyperviseur natif ou bare-metal, installé directement sur le matériel, ce qui le divise en plusieurs machines virtuelles sur lesquelles nous pouvons installer des systèmes d'exploitation invités.
- *L'hyperviseur de type 2* : également appelé hyperviseur hébergé, installé dans un système d'exploitation hôte, avec l'avantage de ne pas avoir besoin d'une console de gestion d'hyperviseur. Les hyperviseurs de type 2 ne prennent pas en charge l'allocation dynamique / sur RAM [12].

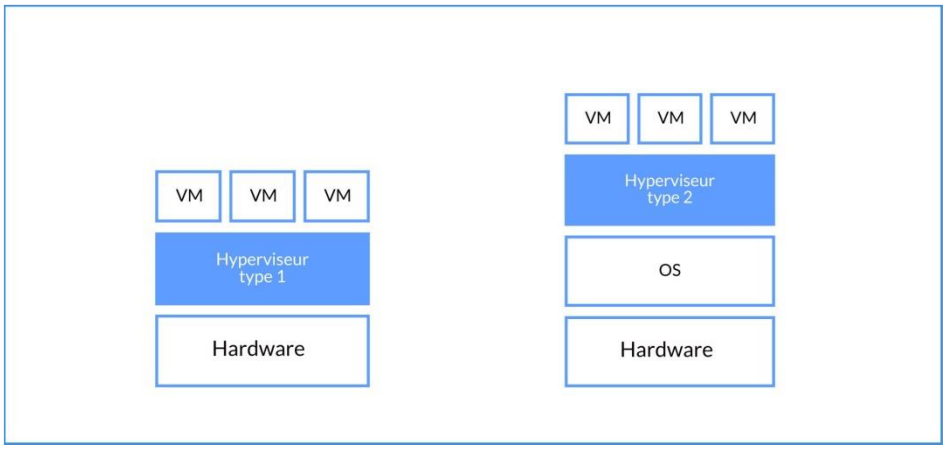


FIGURE 1-3-Les types d'Hyperviseurs [15]

b) Type de virtualisation

- *Virtualisation du réseau* : cela implique la création de réseaux virtuels pouvant être utilisés pour connecter différents réseaux physiques ou pour partitionner un seul réseau physique en plusieurs réseaux virtuels.
- *Virtualisation des postes de travail* : cela implique la création de postes de travail virtuels accessibles à distance par les utilisateurs finaux, permettant une plus grande flexibilité et mobilité.
- *Virtualisation des serveurs* : cela implique la création de plusieurs serveurs virtuels sur un seul serveur physique, permettant une utilisation plus efficace des ressources et une meilleure gestion des charges de travail [16].

1.2.1.3.3 Avantages du Cloud

- *Pas d'investissement initial et Souplesse* : plus grande flexibilité des outils informatiques (pas d'installation ni de mises à jour, pas de maintenance matérielle).
- *Réduction des coûts* : les utilisateurs ne payent que ce qu'ils consomment. Forte économie en coût et énergie notamment dans les cas de besoins non constants ou linéaires.
- *Scalabilité* : le cloud computing permet aux organisations d'augmenter ou de réduire rapidement leurs ressources informatiques selon leurs besoins.
- *Sécurité*: Diminution du risque de panne matérielle. Les données sont sécurisées.
- Mobilité : l'utilisateur peut à tout moment et à partir de n'importe quel appareil se connecter à ses applications.
- *Flexibilité* : le cloud computing offre une large gamme de services et de modèles de déploiement, permettant aux organisations de choisir les services et les configurations qui répondent le mieux à leurs besoins.
- *Gain de productivité et de temps* [12].

1.2.1.3.4 Conteneurisation

La conteneurisation consiste à rassembler le code du logiciel et tous ses composants (bibliothèques, Framework et autres dépendances) de manière à les isoler dans leur propre conteneur. Le logiciel ou l'application dans le conteneur peut ainsi être déplacé et exécuté de façon cohérente dans tous les environnements et sur toutes les infrastructures, indépendamment de leur système d'exploitation. Le conteneur fonctionne comme une sorte de bulle, ou comme un environnement de calcul qui enveloppe l'application et l'isole de son entourage [17].

a) Avantages de la conteneurisation sur les microservices

- *Portabilité* : un conteneur crée un package exécutable de logiciels qui est extrait du système d'exploitation hébergeur. Par conséquent, ce package est portable et capable de fonctionner de manière uniforme et cohérente sur n'importe quelle plateforme ou cloud.

- *Isolation* : chaque application conteneurisée est isolée et fonctionne indépendamment des autres. La défaillance d'un conteneur n'affecte pas le fonctionnement continu des autres conteneurs. Les équipes de développement peuvent identifier et corriger tout problème technique dans un conteneur sans aucun temps d'arrêt dans d'autres conteneurs.
- *Déploiement rapide* : les conteneurs peuvent être déployés rapidement et facilement, ce qui permet des cycles de publication plus rapides et une mise sur le marché plus rapide.
- *Evolutivité* : les conteneurs peuvent être facilement agrandis ou réduits pour répondre aux demandes de chaque microservice, permettant ainsi une plus grande flexibilité et efficacité [18].

1.2.2 Rôle des applications mobiles/web dans la gestion des laboratoires d'analyses médicales

1.2.2.1 Les laboratoires d'analyses médicales

Un laboratoire d'analyse médicale (ou laboratoire de biologie médicale) est une structure où des professionnels de la santé prélèvent et analysent différents fluides de l'organisme. Il peut s'agir de prélèvement de sang, de peaux, d'urines, de selles ou de muqueuses. Un laboratoire de biologie médicale peut être hospitalier ou privé. Il regroupe de nombreux professionnels de la santé tels que des infirmiers, des techniciens de laboratoire, des biologistes et plus rarement des médecins [19].

La première étape du processus du travail dans un laboratoire d'analyses médicales est la réception des échantillons de sang, d'urine collectés par les médecins ou les techniciens des prélèvements. Après la préparation des tubes en fonction du type et de test requis, ses échantillons sont analysées à l'aide des équipements spécialisés et spécifique. Les résultats d'analyses sont interprétés par des microbiologistes ou des technologues de laboratoire médical en fonction des tests réalisées. Les résultats d'analyse sont enregistrés dans le système de gestion de laboratoire et sont envoyés aux médecins ou aux cliniques, qui interprètent ces résultats. Finalement, les résultats d'analyse et les rapports sont archivés dans le système de gestion de laboratoire pour référence future [20].

Les Services Principaux D'un Laboratoire D'analyses Médicales :

- *La chimie* : il s'agit de l'étude scientifique de la matière et des composés des éléments relatifs à l'organisme humain.
- *L'hématologie* : il consiste également en l'étude des maladies du sang. Quant à la *coagulation*, c'est la science qui étudie l'activité de coagulation du sang.
- *La microbiologie* : c'est l'étude des micro-organismes tels que les virus, les protozoaires, les algues, les champignons et les bactéries.
- *La cytologie* : consiste à examiner les cellules au microscope dans le but de rechercher des signes de maladies telles que le cancer.
- *L'immunologie* : cette branche de la médecine étudie les produits immunitaires, comme son nom l'indique. On peut donc citer les anticorps qui se développent en réponse d'un corps étranger [20].

1.2.2.2 Rôle des applications mobiles/ web dédiées aux laboratoires d'analyse médicales

La gestion classique utilisée par certain laboratoires d'analyses médicales qui s'en passe des applications mobiles/web est désormais confronté à une mauvaise organisation dans les différentes taches exécutées au labo et un manque certain d'efficacité dans la gestion des données de prélèvements intra-laboratoire et extra-laboratoire.

Au niveau du laboratoire, lorsque la méthode de gestion classique est pratiquée celle-ci présente plusieurs anomalies dont on peut citer comme exemples :

- La lenteur dans la réalisation des différentes taches du processus des analyses.
- Le manque d'efficacité
- Une probabilité d'erreur assez élevée.
- Le manque d'organisation.
- La perte de certains dossiers médicaux.
- La non disponibilité des données n'importe où et n'importe quand.

De nos jours, les applications mobiles/web touchent tous les domaines de la vie courante : l'actualité, l'éducation, la cuisine, les finances, la santé ou encore les loisirs et les divertissements. C'est pourquoi on trouve de plus en plus d'applications

mobiles/web variées et utiles destinées aux patients et professionnels de santé et qui répondent à de nombreux besoins [21].

Le rôle de l'informatique dans la biologie est en constante évolution. Celle-ci facilite les processus et la centralisation des données et on ne peut pas imaginer aujourd'hui un laboratoire travaillant sans informatique [22]. Les smartphones ou les tablettes sont devenus de véritables objets du quotidien pour les gestionnaires/managers et les employés des laboratoires d'analyses médicales.

Les applications mobiles/web ont déjà assuré leur place dans la gestion des laboratoires d'analyses médicales à travers :

- La gestion efficacement de toutes les opérations liées à la réalisation du processus des analyses médicales.
- La bonne gestion des factures et des paiements en utilisant des systèmes des facturations électroniques.
- Le stockage les données sur des serveurs d'une manière plus sécurisée avec possibilité de récupération plus rapide et plus facile.
- La facilitation de la saisie de données et de la vérification des résultats.
- L'informatisation de la communication entre, d'un côté, le patient et le personnel médicale, et d'un autre côté, entre le gérant et ses employés. Dès lors, les patients peuvent, sans se déplacer, prendre un rendez-vous et consulter à distance les résultats des tests effectués. Par ailleurs, la communication entre le gérant et le personnel, est plus efficace et plus facile.
- La possibilité de contrôle et de gestion à distance.
- L'accès facile des utilisateurs à leurs informations médicales [22].

1.3 Problématique

Depuis plusieurs années l'Internet a révolutionné la façon d'accéder à l'information et de la partager. L'extraction d'information est plus facile maintenant que jamais auparavant. Depuis l'avènement des moteurs de recherche modernes, des réseaux sociaux et de l'accès omniprésent à des appareils comme les téléphones intelligents, les tablettes et les ordinateurs portables, les gens ont l'information au bout des doigts à tous moments et à toute heure de la journée. Le secteur de la santé s'est saisi de ces nouvelles technologies tant du côté des professionnels de santé, que

de celui des patients [23].

Cependant, malgré l'avancée remarquable de l'informatisation dans les laboratoires d'analyse médicale, un autre problème compromettant, lié aux services de sous-traitance, reste à résoudre. En effet, les laboratoires d'analyses médicales font très souvent appel à des laboratoires sous-traitants pour diverses raisons, telles que :

a) Le besoin de s'assurer de la validation de certains résultats douteux.
b) Au cas où certains appareils du labo tombent en panne.
c) L'externalisation de tests spécialisés.
d) La gestion de la surcharge de travail.

Dans ce cas, le problème réside dans le manque du suivi et l'absence d'une communication efficace et d'une coordination étroite entre le laboratoire d'analyses médicales et le sous-traitant, ce qui peut entrainer des retards dans la réception des résultats d'analyses, des pertes de tubes de sang ou même des erreurs de transmission des échantillons.

Dans la suite de ce travail on proposera le développement d'une application mobiles/web capable d'offrir une solution significative et rentable à ce type de problèmes de sous-traitance et de revaloriser, ainsi, l'image de marque des laboratoires d'analyses médicales.

1.4 Objectifs et contribution

Les laboratoires d'analyses médicales jouent un rôle essentiel dans notre environnement en contribuant à la santé publique, à la prévention des maladies et à la prise en charge des patients. Ils fournissent des informations cruciales sur les conditions de santé des individus, ce qui permet de diagnostiquer et de traiter les maladies de manière précoce.

Les applications mobiles/web de gestion des laboratoires d'analyses médicales sont devenues des éléments de communication interne essentielle, car elle donne aux employés la flexibilité de travail dont ils ont besoin. Elles rendre l'accès tellement facile à la communication interne, aux outils internes et aux informations de manière plus fluide. Ces derniers ont pour but d'augmenter la productivité des employés, de gérer de manière efficace les données et d'optimiser certains processus.

Dans le cadre de ce travail, notre objectif consiste à réaliser une application mobile & web basée sur une architecture microservices, nommée « SMARTLAB » pour :

- Faciliter la collaboration et la communication entre les laboratoires d'analyses médicales.
- Gérer de manière efficace les données et optimiser certains processus.
- Economiser du temps et de l'argent.
- Aider les employées d'accéder facilement aux informations de leurs patients.
- Garantir la sécurité des données.
- Augmenter la productivité et améliorer l'efficacité du travail des employées.
- Assurer le bon suivi des sous-traitances entre les laboratoires.
- Offrir une expérience utilisateur conviviale et intuitive.
- Apporter un aide numérique avec une interface plusclaire, plus largeet simple à utiliser pour aider les employées de gérer leurs taches de manière simple, pratique et fiable à tout moment en tous lieux avec facilité et confiance.

Ce qui différencie notre application « SMARTLAB » des autres applications existantes, c'est l'ajout d'une nouvelle idée qui est le suivie des sous-traitances (Paquets et tubes de sang), en utilisant l'architecture des microservices.

1.5 Plan du mémoire

Après une description globale du contexte, de la problématique et des objectifs de notre travail nous nous focaliserons **en deuxième partie** sur l'état de l'art des modèles existants des applications mobiles/ web de gestion des laboratoires d'analyses médicales et nous présenterons une description sommaire de notre modèle proposé.

En troisième partie nous présenterons la modélisation et la conception de notre modèle proposé qui consiste à améliorer les applications mobiles/ web de gestion des laboratoires d'analyses médicales. **La quatrième partie** sera consacrée à la description des outils et langages utilisés et à la présentation des résultats obtenus. Enfin nous terminerons par une conclusion générale et des perspectives.

Chapitre 2

Etat de l’art

2.1 Introduction

Il existe de nombreuses applications de gestion des laboratoires d'analyses médicalesqui permettent aujourd'hui aux managers d'augmenter la productivité des employés, d'assurer le plus haut degré de suivi des tâches, d'optimiser les processus et aussi de garantir la bonne gestion au milieu du laboratoire. Dans ce chapitre nous présenterons, dans un premier temps, une liste non exhaustive d'applications mobiles/webexistantes dont l'objectif est de répondre aux besoins des laboratoires en termesd'optimisation du processus de réalisation des différentes taches et d'amélioration de la communication interne et externe ; par la suite, nous donnerons une description sommaire de notre modèle proposé.

2.2 Modèles existants

2.2.1 Application 'LabCollector'

LabCollector, tel que présentée sur la figure 2-1, est une plateformede gestion de laboratoirecomplète, disponible à la fois sous forme d'application web, accessible via un navigateur, et sous forme d'application mobile pour un accès pratique à partir de smartphones et de tablettes.

Les fonctionnalités clés de cette plateforme sont :

- Création de protocoles de tests.
- Rapport final et générateur de factures.
- Assistance pour se connecter aux machines PCR.
- Gérer l'inventaire des réactifs et des fournitures avec des alertes & notifications [24].

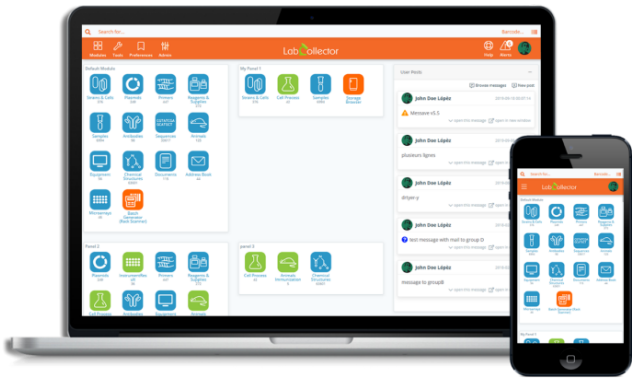


FIGURE 2-1-Application 'LabCollector' [25]

2.2.2 Application ' Lab Tests Online'

Lab Tests Online, illustré sur la figure 2-2, est une application web/mobilequi fournit aux utilisateurs des informations détaillées sur des centaines de tests de laboratoire ainsi que des descriptions de maladies qui sont réticulés par les tests utilisés pour les dépister, les diagnostiquer et les traiter [26].

Cette application offre un certain nombre d'articles de fonctionnalités qui expliquent des concepts tels que la fiabilité des tests et la difficulté de définir et d'utiliser des gammes de référence. Le site a également des informations originales qui résument brièvement les dernières avancées en sciences de laboratoire et les politiques et directives qui régissent l'utilisation des tests [26].

L'objectif principal est de permettre aux patients, aux professionnels de la santé et au grandpublic de mieux comprendre les tests de laboratoire, leur signification et leur utilisation dans le diagnostic et le suivi des conditions médicales [26].

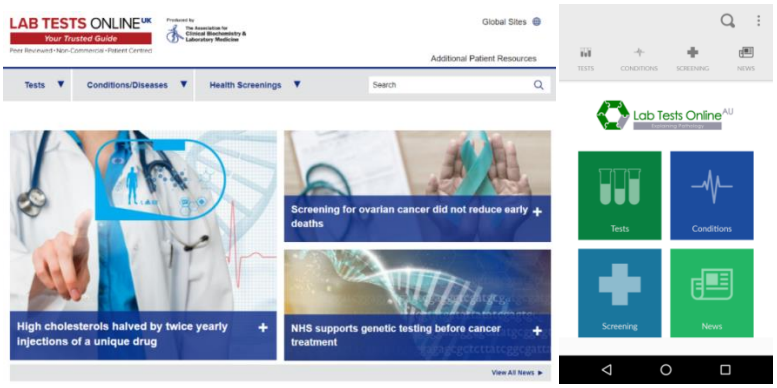


FIGURE 2-2-Application 'Lab Tests Online' [27]

2.2.3 Application ' LabWare'

Le site web de LabWare qui est représenté par la figure 2-3 est une plateforme en ligne qui fournit des informations sur les solutions de gestion de laboratoire, y compris les systèmes d'information de gestion de laboratoire (LIMS).

LabWare propose également une application mobile, qui permet d'accéder aux fonctionnalités du système LabWare depuis des appareils mobiles tels que les smartphones et les tablettes [28].

Cette application peut être utilisée pour exécuter des fonctions telles que :

- Enregistrement des échantillons.

- Gestion des patients dans un laboratoire clinique.
- Création et téléchargement de fichiers et de photos.
- Appareil photo, y compris la lecture de codes-barres et les QR codes [28].

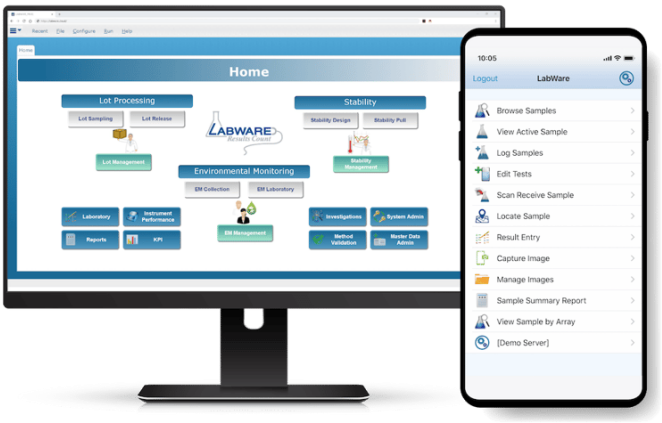


FIGURE 2-3-Application 'LabWare' [29]

2.2.4 Application ' Polytech'

Dans la figure 2-4 est exposé l'application Polytech LIS qui est un système de laboratoire médical basé sur le cloud qui aide les laboratoires médicaux à gérer les dossiers des patients et les flux de travaux du laboratoire [30].

Il aide au contrôle de la qualité en exécutant des contrôles automatiques et en affichant les résultats avec des erreurs signalées en fonction des valeurs saisies. En outre, il inclut une vérification automatique du delta par test sur des valeurs absolues ou en pourcentage [30].

Les principales caractéristiques comprennent :

- Une interface d'instrument.
- Une interface de laboratoire de référence.
- La gestion du contrôle de la qualité.
- Le stockage des données.
- Le support client.
- Le traitement des résultats.

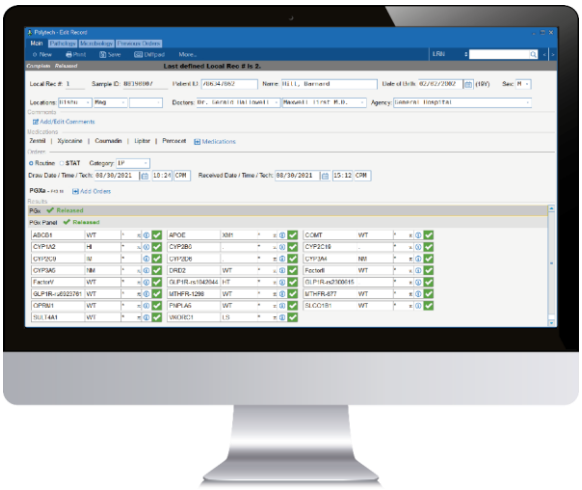


FIGURE 2-4-Application 'Polytech' [31]

Alors, il existe des milliers d'applications mobiles/ web (gratuites ou payantes) dont chacune a son principe et son type de fonctionnement. Chaque personne trouvera donc une application qui correspondra à ses propres besoins.

2.3 Description sommaire du modèle proposé

Après avoir étudié les modèles existants des applications mobiles& web de gestion des laboratoires d'analyses médicales, on a remarqué l'absence quasi-totale d'une prise en charge desdifficultés liées à la sous-traitance telle que détaillée dans le chapitre précédant. C'est sur cette base qu'on a décidé de contribuer au développement d'une application mobile/web qui peut solutionner cette problématique.Dans ce qui suit, nous présenterons une descriptionsommaire de notre modèle proposé.

Le modèle proposé,dont l'image symbolique est donné sur la figure 2-5,est un systèmede gestion d'un laboratoire d'analyses médicales mobile& web (disponible sur Android/ iOS), nommé 'SMARTLAB', qui est basé sur une architecture microservices. Ce dernier est dédié aux médecins/managers et aux employés du laboratoire médical.

SMARTLAB est un intermédiaire mettant en relationles différents laboratoires d'analyses médicales, c'est un véritable accompagnateur numérique, oùl'utilisateur peut l'utiliserpour gérer les sous-traitances (Paquets/ Tubes) facilement à tout moment et en tout lieu avecefficacité et confiance. De plus, il est conçu pour garantir une bonne sécurisation des informations et garantir une parfaite communication et interaction entre les différents laboratoires et entre les managers

et les employés.

En somme, ce qui différencie notre application « SMARTLAB » des autres applications existantes, c'est l'ajout d'une nouvelle idée qui est le suivie des sous-traitances (Paquets et tubes de sang), en utilisant l'architecture des microservices.

Parmi les fonctionnalités de notre application :

- L'ajout des informations des tubes de sang.
- Le suivi de l'état des paquets.
- Le choisir d'un livreur.
- L'envoie des résultats sous forme PDF.
- La recherche rapide des paquets et des tubes de sang.
- La consultation et la vérification des résultats à tout moment et en tous lieux.
- La sélection des laboratoires capable derecevoir les paquets à partir du type d'analyse.
- L'affichage de l'historique des transactions des tubes et des paquets.
- L'ajout des comptes des laboratoires d'analyses médicales.

SMARTLABa trois types d'utilisateurs:

- Le Manager/Employé du laboratoire d'analyse médicale.
- Le Livreur.
- L'administrateur.

1- Utilisateur : Manager/Employé du Laboratoire :

Après la création du compte par l'admin, l'utilisateur doit s'authentifier avec ce compte ; par la suite, il peut :

- Consulter la liste des paquets.
- Faire une recherche rapide sur la liste des paquets.
- Gérer le statut du paquet.
- Consulter la liste des tubes de sang de chaque paquet.
- Supprimer un paquet ou un tube dans un paquet.
- Imprimer le code bar d'un paquet.
- Créer une demande d'un nouveau tube de sang.
- Consulter la liste des livreurs de son propre laboratoire.
- Ajouter un livreur ou modifier les informations d'un livreur.

- Rechercher rapidement un livreur.
- Accéder aux informations des tubes de sang.
- Imprimer le code bar d'un tube de sang.
- Consulter la liste des paquets reçus.
- Consulter la liste des tubes de sang de chaque paquet.
- Télécharger les fichiers PDF des résultats.
- Envoyer les fichiers des résultats.
- Consulter la liste des paquets reçus et faits.
- Consulter la liste des résultats reçus des tubes de chaque paquet avec l'option d'imprimer et télécharger le résultat.

2- Utilisateur : Livreur

Après la création du compte par le manager du laboratoire, l'utilisateur doit s'authentifier avec ce compte, et ensuite il peut :

- Consulter la liste des paquets prêts à envoyer.
- Consulter la liste des tubes de chaque paquet.
- Changer le statut des paquets « Ready to Transport ».
- Accepter la livraison du paquet.
- Accéder à la localisation du laboratoire destinataire.

3- Utilisateur : Administrateur

L'utilisateur doit s'authentifier, et il peut alors:

- Gérer son compte.
- Créer les comptes des laboratoires d'analyses médicales.
- Accéder à la liste des laboratoires existants.
- Gérer la liste des analyses de chaque laboratoire.

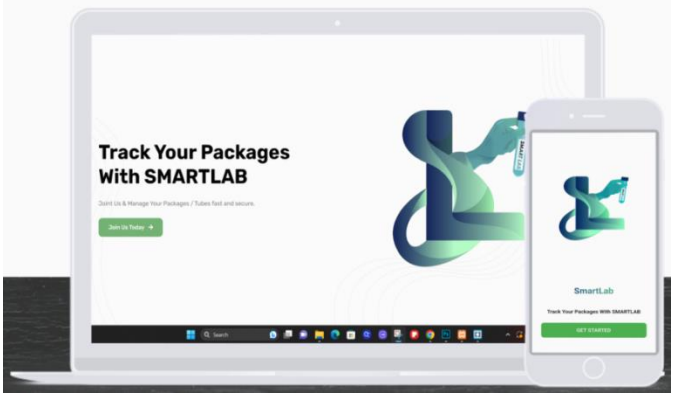


FIGURE 2-5-Application 'SMARTLAB'

2.4 Objectifs du modèle proposé

Les grands objectifs qui caractérisentle modèle proposé sont :

- Développer une application avec une interface plus large, plus claire et simple à utiliser.
- Avoir un accès rapide aux échantillons des paquets et aux résultats des analyses.
- Assurer le plus haut degré de suivi des sous-traitances (Paquets, Tubes de sang), et la bonne communication et interaction entre les laboratoires.
- Gérer de manière efficace et sécuriséles données.
- Economiser du temps et de l'argent.
- Travailler n'importe quand, n'importe où.
- Répondre aux besoins spécifiques des laboratoires médicaux.
- Améliorer la communication interne et externe.
- Augmenter la productivité et améliorer l'efficacité du travail des employées des laboratoires.
- La bonne gestion de la charge des paquets reçus (les cas d'une épidémie par exemple le cas du COVID-19).

2.5 Conclusion

Dans ce chapitre, nous avons présenté quelques modèlesexistants d'application de gestion des laboratoires d'analyses médicales, leurs fonctionnements et leurs objectifs. Après l'analyse de ces modèles, nous avons préparé une description sommaire pour notre propre modèle et nous avons identifié ces principaux objectifs.

Dans le chapitre suivant, nous allons entamer la modélisation et la conception de notre modèle.

Chapitre 3

Architecture et modélisation

3.1 Introduction

La réalisation d'un système nécessite la modélisation qui permet d'anticiper, de prévoir et d'étudier les informations relatives à ce système. Pour se faire, on a opté pour le langage UML qui permet de représenter des concepts graphiques et de modéliser les applications. Cette modélisation UML montre les différents acteurs du système ainsi que les rôles qu'ils peuvent tenir.

3.2 Méthodologie de conception

Dans ce qui suit nous allons présenter le langageUML.

3.2.1 Présentation du UML

UML est un langage de modélisation orientée objet développé en réponse à l'appel de la proposition lancée par l'OMG dans le but de définir une notation standard pour la modélisation des applications construites à l'aide d'objets et aussi pour la conception des logiciels. Aussi, UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter [32].

UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc [32].

3.3 Analyse et conception

3.3.1 Diagramme de cas d'utilisation

Le Diagramme de cas d'utilisation est utilisé pour la modélisation des besoins des utilisateurs. Les cas d'utilisations décrivent le comportement du système étudié du point de vue de l'utilisateur, et les possibilités d'interactions fonctionnelles entre le système et les acteurs ; ils permettent de définir les limites et les relations entre le système et son environnement [32].

3.3.1.1 Rôle du diagramme de cas d'utilisation

- Donne une vue du système dans son environnement extérieur.
- Définit la relation entre l'utilisateur et les éléments que le système met en œuvre.

3.3.1.2 Les composants d'un diagramme de cas d'utilisation

Les composants de base des diagrammes de cas d'utilisation sont l'acteur, le cas d'utilisation, et l'association [32]:

- **Acteur :** Un acteur est un utilisateur qui communique et interagit avec les cas d'utilisation du système. C'est une entité ayant un comportement comme une personne ou système.
- **Cas d'utilisation :** Un cas d'utilisation représente une fonctionnalité fournie par le système, typiquement décrite sous la forme Verbe+objet(par exemple immatriculer voiture, effacer utilisateur). Les cas d'utilisation sont représentés par une ellipse contenant leurs noms.
- **Association :** Les associations sont utilisées pour lier des acteurs avec des cas d'utilisation. Elles indiquent qu'un acteur participe au cas d'utilisation sous une forme quelconque. Les associations sont représentées par une ligne reliant l'acteur et le cas d'utilisation.

3.3.1.3 Diagramme de cas d'utilisation de notre application mobile/ web

Les diagrammes de cas d'utilisations relatifs aux applications mobile et web proposées sont respectivement montrés sur les FIGURES 3-1 et 3-2.

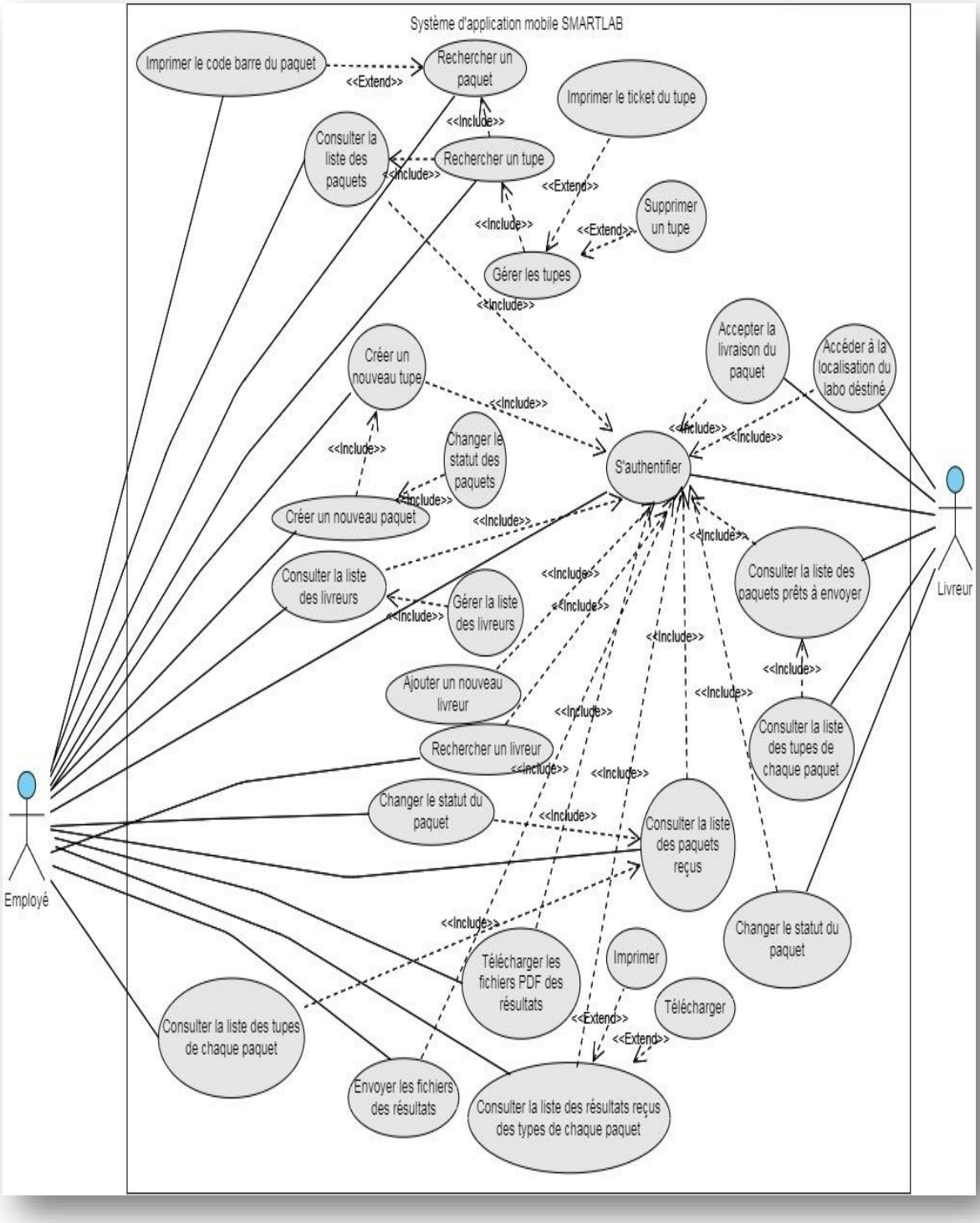


FIGURE 3-1- Diagramme de cas d'utilisation général –Application mobile-

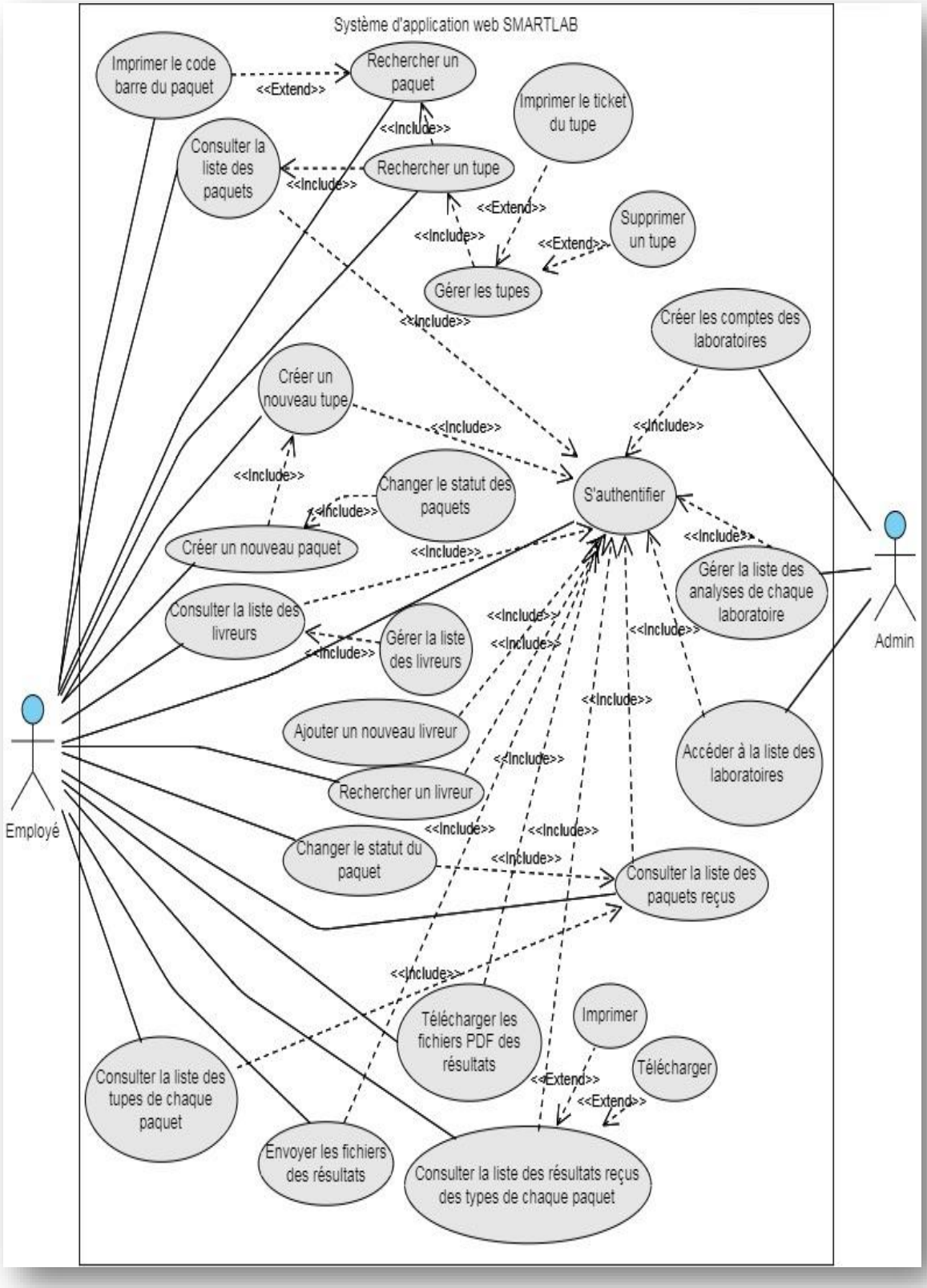


FIGURE 3-2-Diagramme de cas d'utilisation général -Application Web–

3.3.1.4 Description textuelle des cas d'utilisation

Un cas d'utilisation 'CU' permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié [33].

- **Pré-conditions :** définissent les conditions qui doivent être satisfaites pour que la CU puisse démarrer.
- **Post-conditions :** définissent ce qui doit être vrai lorsque le CU se termine avec succès, qu'il s'agisse d'un scénario nominal ou alternatif.

3.3.1.4.1 Cas d'utilisation ' S'authentifier'

Le TABLEAU 3-1illustre le cas d'utilisation de '*S'authentifier*' et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-1-S'authentifier

Identification • **Nom du cas d'utilisation :** s'authentifier. • **But :** avoir un accès à l'application. • **Acteur:** Employé, Livreur, Administrateur.
Séquencement • L'utilisateur lance l'application. **Pré-condition:** avoir un compte. **Enchainements nominaux:** • L'utilisateur accède à la zone d'authentification. • L'application affiche le formulaire d'authentification. • L'utilisateur saisit son identifiant (email) et son mot de passe pour se connecter. **Enchainements alternatifs:** • Identifiant non valide. • Mot de passe incorrect. • Compte inexistant. • L'oubli d'un champ. **Post-conditions:** • Utilisateur authentifié. • Accédez aux ressources de l'application.

3.3.1.4.2 Cas d'utilisation ' Créer un nouveau tube de sang'

Le TABLEAU 3-2 illustre le cas d'utilisation de '*Créer un nouveau tube de sang'* et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-2-Créer un nouveau tube de sang

Identification • **Nom du cas d'utilisation :** créer un nouveau tube de sang. • **But:** créer un tube de sang avec toutes les informations du patient. • **Acteur:** Employé.
Séquencement **Pré-condition:** authentification. **Enchainements nominaux :** • L'utilisateur accède à l'interface de création. • L'application demande à l'utilisateur de remplir un formulaire d'information. • L'utilisateur saisit les informations et valide sa demande. **Enchainements alternatifs:** • Oubli d'un champ obligatoire. • Tube existe déjà. **Post-conditions:** • Mettre à jour la base de données.

3.3.1.4.3 Cas d'utilisation ' Créer un nouveau paquet'

Le TABLEAU 3-3 illustre le cas d'utilisation de '*Créer un nouveau paquet*' et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-3-Créer un nouveau paquet

Identification • **Nom du cas d'utilisation :** créer un nouveau paquet. • **But:** créez un paquet contenant tous les tubes de sang qui seront envoyés au laboratoire destinataire. • **Acteur:** Employé.
Séquencement **Pré-condition :** authentification. **Enchainements nominaux :** • L'utilisateur accède à l'interface d'accueil. • L'utilisateur clique sur le bouton « Ajouter un nouveau tube de sang ». • L'application demande à l'utilisateur de remplir un formulaire d'information. • Lors de la saisie des informations, l'application propose à l'utilisateur de sélectionner un nouveau paquet. • L'utilisateur sélectionne le choix « Nouveau paquet ». • L'utilisateur saisit les autres informations et valide la demande. **Enchainements alternatifs :** • Données saisies invalides. **Post-conditions:** • Mettre à jour la base de données. • Créer un nouveau paquet.

3.3.1.4.4 Cas d'utilisation ' Consulter la liste des livreurs '

Le TABLEAU 3-4 illustre le cas d'utilisation de '*Consulter la liste des livreurs'* et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-4-Consulter la liste des livreurs

Identification • **Nom du cas d'utilisation :** consultez la liste des livreurs. • **But:** consultez toutes les informations des livreurs. • **Acteur:** Employé.
Séquencement **Précondition :** authentification. **Enchainements nominaux :** • L'utilisateur accède au tiroir situé à gauche de la page d'accueil. • L'utilisateur sélectionne le choix « Livreurs ». • L'application affiche la liste des livreurs du laboratoire. **Enchainements alternatifs :** • Aucun. **Post-conditions:** • L'utilisateur consulte toutes les informations des livreurs

3.3.1.4.5 Cas d'utilisation ' Changer le statut du paquet '

Le TABLEAU 3-5 illustre le cas d'utilisation de '*Changer le statut du paquet*' et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-5-Changer le statut du paquet

Identification • **Nom du cas d'utilisation :** changer le statut du paquet. • **But:** aide à suivre l'état du paquet. • **Acteur:** Employé, Livreur.
Séquencement **Précondition:** authentification, consulter la liste des paquets. **Enchainements nominaux :** • L'utilisateur accède à la liste des paquets. • L'utilisateur sélectionne un paquet. • L'application affiche les informations sur le paquet. • L'utilisateur clique sur le bouton déroulant « Modifier le statut ». **Enchainements alternatifs :** • Aucun. **Post-conditions:** • L'état du colis a été modifié. • Mettre à jour la base de données.

3.3.1.4.6 Cas d'utilisation ' Créer compte laboratoire '

Le TABLEAU 3-6 illustre le cas d'utilisation de '*Créer compte laboratoire'* et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-6-Créer compte laboratoire

Identification • **Nom du cas d'utilisation:** créer compte laboratoire. • **But:** ajouter ou supprimer compte Laboratoire. • **Acteur:** Administrateur.
Séquencement **Précondition :** authentification. **Enchainements nominaux :** • L'utilisateur « Admin » accède à l'interface « Gérer les comptes utilisateurs ». • L'utilisateur a la possibilité d'ajouter un nouveau compte en cliquant sur le bouton « Ajouter » et en saisissant les informations nécessaires. • Ou supprimez un compte déjà présent. **Enchainements alternatifs :** • Aucun compte à supprimer. • Le compte existe déjà. **Post-conditions:** • Mettre à jour la base de données.

3.3.2 Diagramme de séquence

Le diagramme de séquence, permet de représenter les interactions entre différents objets, selon un point de vue temporel en se basant sur la chronologie des envois de messages. Le temps est représenté comme s'écoulant du haut vers le bas le long des « lignes de vie ».Des flèches représentant les messages qui transitent d'une entité vers l'autre, le message est synchrone. Si l'extrémité de la flèche est creuse, le message est asynchrone [33].

Les différents composants d'un diagramme de séquences sont donnés sur la FIGURE 3-3.

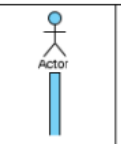
	Acteur	Les acteurs peuvent communiquer avec des objets, ainsi ils peuvent eux aussi être énumérés en colonne. Un acteur est modélisé en utilisant le symbole habituel: Stickman.
	Objet	Les objets sont des entités appartenant au système (instance d'une classe) ou se trouvant à ses limites (acteurs)
	Ligne de vie	Elle est représentée par une ligne verticale en dessous des objets, représente la période de temps durant laquelle l'objet "existe".
	Message récursif	L'envoi de messages récursifs se représente par un dédoublement de la bande d'activation
	Message	Les objets communiquent en échangeant des messages représentés sous forme de flèches, ils sont étiquetés par le nom de l'opération ou du signal invoqué.
	Message de retour	Représenté par une flèche discontinue, c'est la réponse au message envoyé.

FIGURE 3-3-Composants d'un diagramme de séquence [33]

3.3.2.1 Diagramme de séquence 'Authentification'

L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification du login et du mot de passe. Ces informations sont préétablies dans une base de données. Lors de l'authentification de l'utilisateur, deux cas peuvent se présenter : informations correctes ou incorrectes, ce qui explique l'utilisation de l'opérateur «alt». Si les informations fournies sont correctes, alors le système accorde l'accès à l'interface appropriée. En revanche, si l'utilisateur saisit des informations incorrectes, le système génère un message d'erreur et réaffiche la page d'authentification d'où l'utilisation de l'opérateur «loop». Le digramme de séquence d'authentification proposé est illustré sur la FIGURE 3-4.

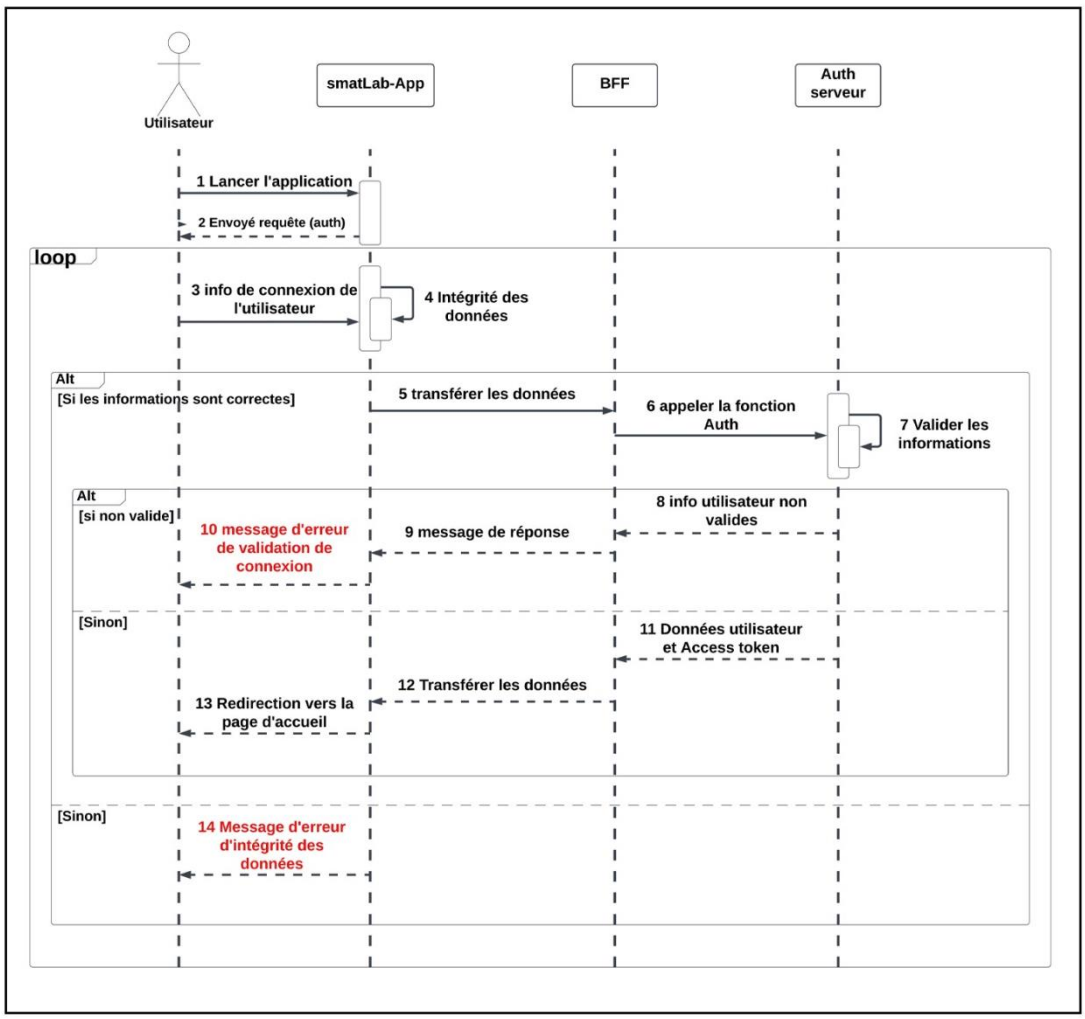


FIGURE 3-4-Diagramme de séquence d'authentification proposé.

3.3.2.2 Diagramme de séquence ‘ Créer un nouveau tube de sang ’

Le diagramme de séquence ‘Créer un nouveau tube de sang’ proposé est présenté sur la FIGURE 3-5.

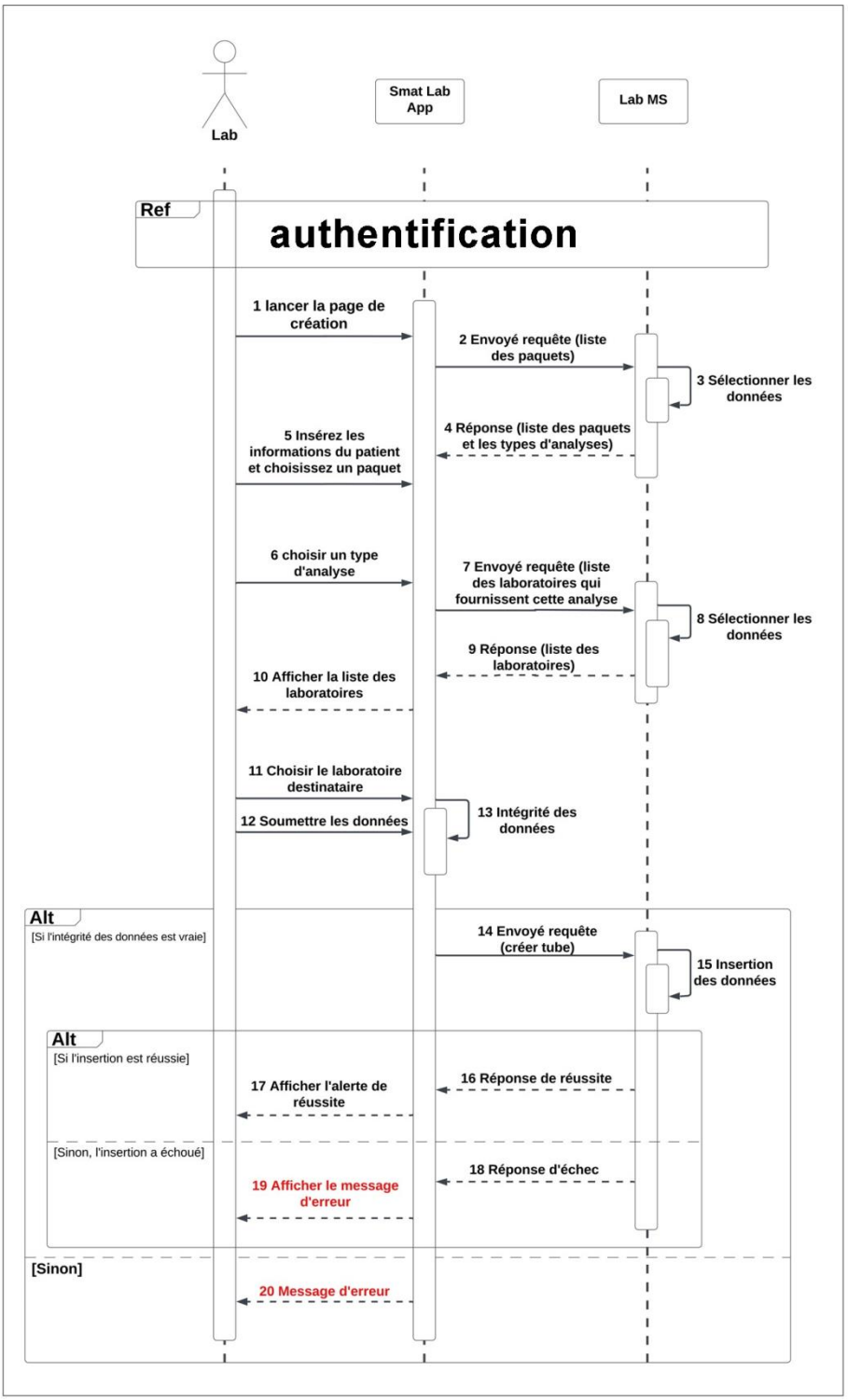


FIGURE 3-5-Diagramme de séquence ‘Créer un nouveau tube de sang’ proposé.

3.3.2.3 Diagramme de séquence ‘ Télécharger le résultat ’

Le Diagramme de séquence ‘ Télécharger le résultat ’proposé est présenté sur la FIGURE 3-5.

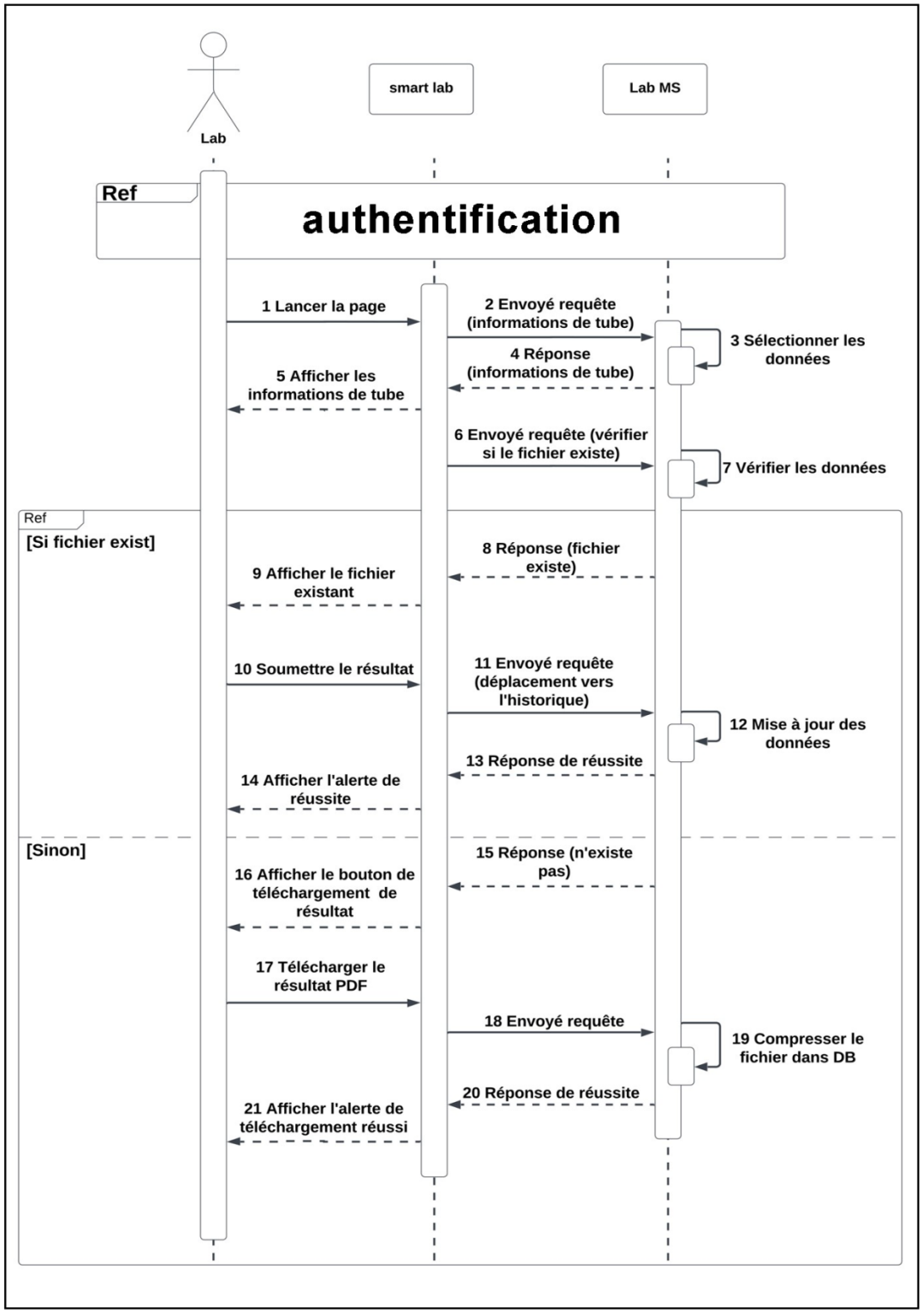


FIGURE 3-6- Diagramme de séquence ‘Télécharger le résultat’ proposé.

3.3.3 Diagramme de classe

Après l'étude détaillée des cas d'utilisation, nous avons déduit le diagramme de classe global du système tel que présenté sur la FIGURE 3-7. Ce diagramme est considéré comme la phase finale de la conception théorique de notre système et sera pris comme la référence à partir de laquelle va se dérouler le développement logiciel, et l'écriture du code source de notre application.

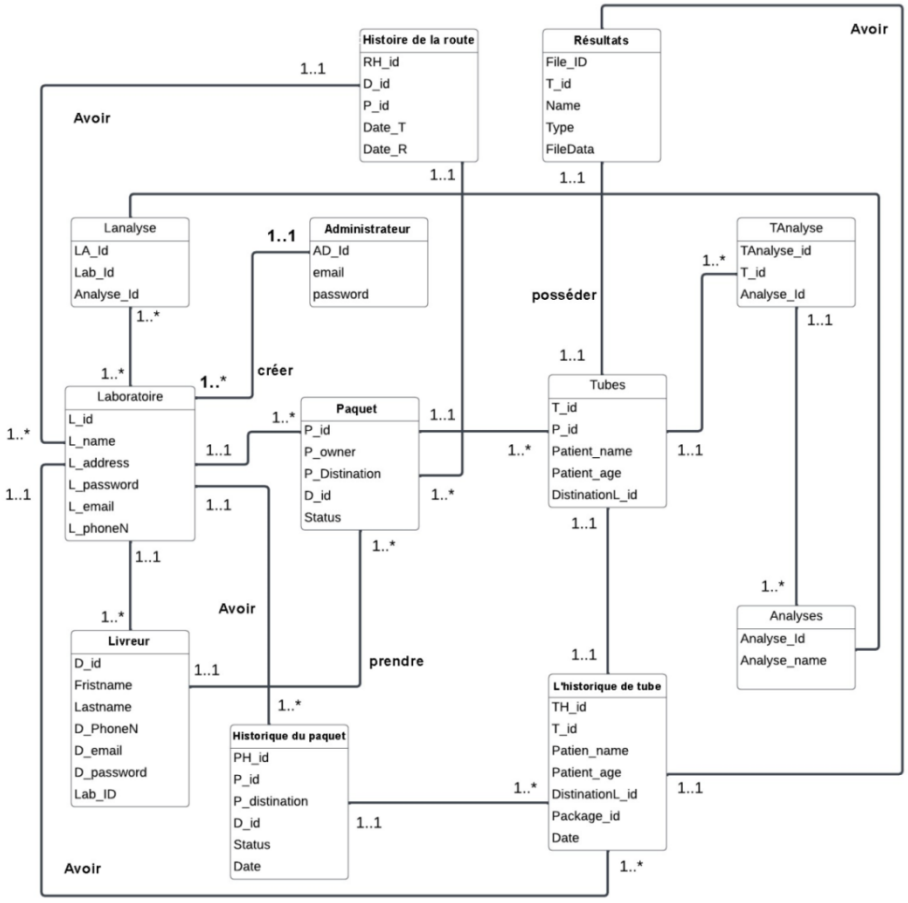


FIGURE 3-7- Diagramme de classe de l'application 'SMARTLAB'

3.4 Conclusion

Dans ce chapitre, nous avons traité la phase d'analyse et conception qui est faite grâce à des diagrammes UML : diagrammes de cas d'utilisation, de séquences, et de classes.

Dans le chapitre suivant nous allons entamer la phase d'implémentation de notre modèle proposé.

Chapitre 4

Implémentation

4.1 Introduction

Après avoir compléter le contenu du chapitre précédant 'Architecture et modélisation', nous aborderons dans ce qui suit la partie implémentation.Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation.

Dans ce chapitrenous présenterons, avant tout, la description des environnements matériels et logiciels qui nous ont permis de réaliser notre projet ainsi queles technologies et les langages de programmation que nous avons utilisée. Ensuite, nous expliquerons le fonctionnement de notre application mobile& web 'SMARTLAB' en présentant ses différentes interfaces qui permettent l'interaction entre l'utilisateur et le système.

4.2 Environnement du travail

4.2.1 Environnement matériel

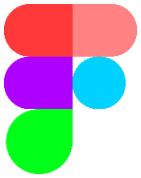
Pour la réalisation de notre projet, nous avons utilisé deux ordinateurs caractérisés par :

1) HP EliteBook 830 G5 caractérisé par :
   - Système d'exploitation : Windows 11 Professionnel.
   - Processeur : Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz 2.71 GHz.
   - Mémoire : 16,0 Go.
   - Disque Dur : 512 Go SSD.
2) Samsung Galaxy Book Pro caractérisé par :
   - Système d'exploitation : Windows 11 Professionnel.
   - Processeur : Intel Corei7 1165G7.
   - Mémoire : 16,0 Go.
   - Disque Dur : 512 Go SSD.

Pour les différentes étapes de test, d'installation et de déploiement de l'application nous avons eu besoin d'un émulateur mobile; nous avons utilisé un Samsung Galaxy M52 5G.

4.2.2 Environnement logiciel

4.2.2.1 Figma



Figma est un éditeur de graphiques vectoriels et un outil de prototypage. Il est principalement basé sur le web, avec des fonctionnalités hors ligne supplémentaires activées par des applications de bureau pour MacOs et Windows [34].

4.2.2.2 Adobe Photoshop

Adobe Photoshop est un logiciel de traitement et de retouche d'images et de photo produit par la société Adobe. Photoshop est devenu le standard en matière de gestion des images matricielles (ou images "bitmap", constituées d'un "tapis de points"). Un logiciel tel qu'Illustrator lui, gère l'image numérique sous la forme de vecteurs (on parle alors d'images vectorielles) [35].

4.2.2.3 Flutter

Flutter est un kit de développement logiciel (SDK) d'interface utilisateur open-source créé par Google. Il est utilisé pour développer des applications pour Android, iOS, Linux, Mac, Windows et le web à partir d'une seule base de code, il est basé sur le langage de programmation Dart développé par google [36].

4.2.2.4 Web

Dans le développement web on a utilisé plusieurs langages de programmation et technologies, comme : HTML, CSS, JavaScript, Bootstrap, jQuery.

4.2.2.5 Java

La technologie Java définit à la fois un langage de programmation orienté objet et une plateforme informatique. La technologie Java est indissociable du domaine de l'informatique et du Web. On la retrouve donc sur les ordinateurs, mais aussi sur les téléphones mobiles, les consoles de jeux, etc. L'avènement du smartphone et la puissance croissante des ordinateurs, ont entraîné un regain d'intérêt pour ce langage de programmation [37].

4.2.2.6 Spring Boot

Java Spring Boot (Spring Boot) est une infrastructure open source ; c'est un outil qui accélère et simplifie le développement d'applications Web et de microservices avec Spring Framework, qui permet de créer des applications autonomes de production qui fonctionnent sur la machine virtuelle Java (JVM) [38].

4.2.2.7 Keycloak

Keycloak est un produit logiciel open source qui permet l'authentification unique (IdP) avec Identity Management et Access Management pour les applications et services modernes. Ce logiciel est écrit en Java et prend en charge les protocoles de fédération d'identité par défaut SAML v2 et OpenIDConnect (OIDC) / OAuth2. Il est licencié par Apache et pris en charge par Red Hat[39].

4.2.2.8 SQL DB

SQL (StructuredQueryLanguage) est un langage informatique utilisé pour exploiter des bases de données. Il permet de définir, manipuler et contrôler la sécurité des données. En pratique, SQL est utilisé pour créer des tables, ajouter des enregistrements sous forme de lignes, interroger une base de données, la mettre à jour, ou gérer les droits des utilisateurs de cette base. Il est bien supporté par la grande majorité des systèmes de gestion de bases de données (SGBD) [40].

4.2.2.9 PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle orienté objet puissant et open source qui est capable de prendre en charge en toute sécurité les charges de travail de données les plus complexes [41].

4.2.2.10 No SQL DB

NoSQL est une approche de la conception de bases de données qui peut s'adapter à une grande variété de modèles de données, y compris

les formats avec des clés, des documents, des colonnes et des graphiques **(**Graph DB**)**. NoSQL, qui signifie « not only SQL », est une alternative à la base de données relationnelle traditionnelle dans lesquelles les données sont placées dans des tables et le schéma de données est soigneusement conçu avant la construction de la base de données [42].

4.2.2.11 MongoDB

MongoDB est une base de données orientée documents, classé comme programme de base de données NOSQL, il stocke les données dans des documents flexibles de type JSON avec des schémas facultatifs [43].

4.2.2.12 Docker

Docker est une plate-forme logicielle qui vous permet de concevoir, tester et déployer des applications rapidement. Docker intègre les logiciels dans des unités normalisées appelées conteneurs, qui rassemblent tous les éléments nécessaires à leur fonctionnement. Il est distribué en tant que projet open source à partir de mars 2013 [44].

4.2.2.13 Postman

Postman est une Plateforme permettant de créer et tester des API, créée en 2012 pour répondre à une problématique de test d'API partageable. Il simplifie chaque étape du cycle de vie de l'API et rationalise la collaboration afin que vous puissiez créer de meilleures API [45].

4.2.2.14 Editeur de code

On a utilisé VS Code, Brackets et Intellij comme éditeur de code dans notre projet.

4.2.3 Le développement de notre application

- Pour l'application mobile :
  - *Logo* :nous avons créé le logo de notre application "SMARTLAB" avec Photoshop.

- *User Interface (UI):* nous avons créé l'interface de notre application avec « Figma ».
- *Langage utilisé* : nous avons utilisé le framework « Flutter » qui est basé sur le langage « Dart ».
- *Environnement de travail* : nous avons utilisé « VS Code » comme IDE.

➢ Pour l'application web :

- *Front-end*: nous avons créé l'interface de notre application web avec « HTML, CSS, Java Script, jQuery, Bootstrap,Ajax ».
- *Environnement de travail* : nous avons utilisé « Brackets » comme IDE.

➢ Pour le Back-end nous avons utilisé :

- *Langage* : « JAVA ».
- *Framework* : « Spring ».
- *Environnement de travail* : «Intellij» comme IDE.

➢ Pour les bases de données nous avons utilisé :

- *Langage*:« SQL, NoSQL ».
- *SGBD* :« PostgreSQL, MongoDB ».
- *Logiciel* :« DBeaver ».
- *Platform* : « Docker ».

4.3 L'architecture globale de « SMARTLAB »

Notre architecture est principalement basée sur des microservices, ces derniers utilisent différents composants pour interagir les uns avec les autres comme illustré ci-dessous sur la FIGURE 4-1. Tous les microservices communiquent entre eux,lorsqu'une requête est effectuée depuis un point de terminaison (application mobile ou Web) vers le système, elle passe par les stations suivantes :

1. *Requête du point de terminaison* : la requête Http provient de l'application client, contenant les données et paramètres nécessaires.

2. *API Gateway (Spring cloud Gateway)* : La passerelle API est le point d'entrée pour les demandes entrantes. Il reçoit la demande du point de terminaison et agit comme médiateur entre le client et les microservices.

3. *Authentification et autorisation* : la passerelle API vérifie l'authenticité du (jeton d'accès) via Keyckloak (serveur d'authentification). Il garantit que le client est autorisé à accéder aux ressources demandées.

4. *Routage des requêtes* : en fonction du point de terminaison ou de l'URL demandé, la passerelle API achemine la requête vers le microservice approprié. Dans notre cas, le Bff microservice qui regroupent les données et assurent l'accès des différents microservices avec une seule requête.

5. *Service Communication* : après réception de la demande, le Bff microservice ouvre des canaux de communication avec les microservices dont il a besoin :

a. LabMs : (communication synchrone via OpenFeign).

b. DelMs : (communication synchrone via OpenFeign).

c. AdminMs : (communication synchrone via OpenFeign).

6. *Génération de réponse* : après avoir traité la demande, le microservice génère une réponse contenant le résultat de l'opération demandé. La réponse est ensuite renvoyée à la passerelle API et, enfin, au point de terminaison d'origine qui a initié la demande.

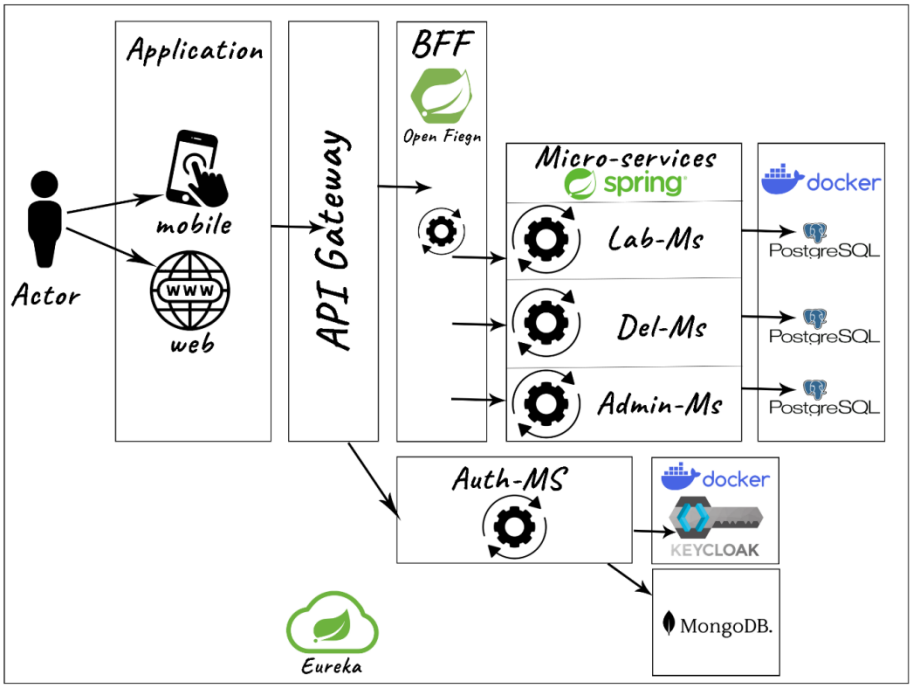


FIGURE 4-1-L'architecture globale de notre application 'SMARTLAB'

4.4 Présentation des interfaces de notre application

Les interfaces graphiques de l'application sont très importantes, car elles permettent de faciliter le dialogue entre l'homme et la machine ainsi que d'améliorer les performances de l'application.

Dans cette partie nous présentons les principales fonctionnalités de notre application par la description de quelques interfaces.

4.4.1 L'interface ' logo de l'application SMARTLAB'

La FIGURE 4-2 illustre l'interface du logo de l'application avec un bouton 'GET STARTED'. Lorsque l'utilisateur clique sur ce bouton, il sera redirigé directement vers la page suivante.

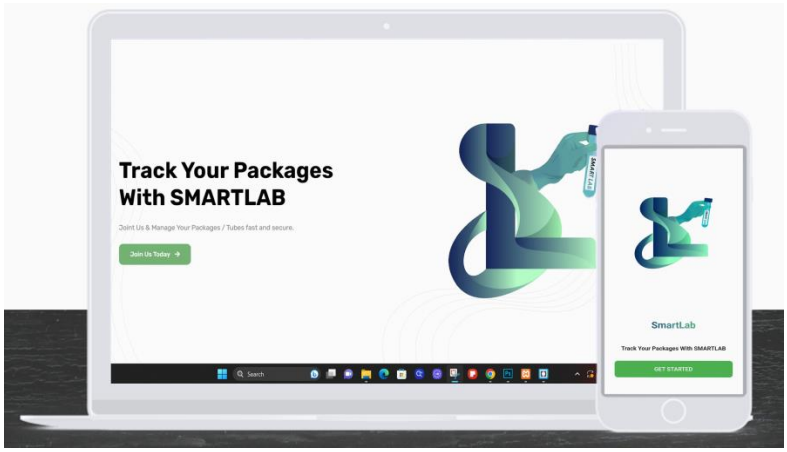


FIGURE 4-2-Interface logo de l'application '**SMARTLAB**'

4.4.2 Interface 'Choose your role'

La FIGURE 4-3 présente le premier lancement de l'application, où l'utilisateur doit choisir : 'laboratoire' ou 'livreur'.

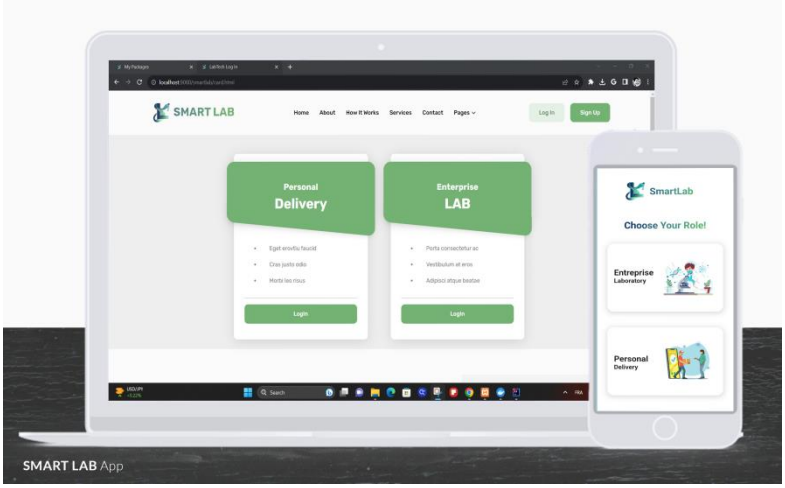


FIGURE 4-3-Interface 'Choose your role'

4.4.3 Interface 'Authentification- Log in Laboratoire'

L'utilisateur doit saisir correctement son email et son mot de passe via l'interface 'Authentification-Log in Laboratoire' présenté sur la FIGURE 4-4 pour accéder à l'application.

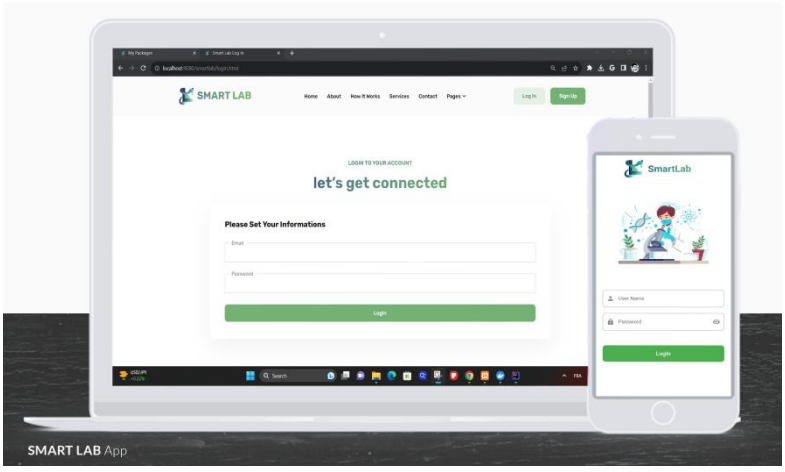


FIGURE 4-4-Interface 'Authentification- Log in Laboratoire'

4.4.4 Interface 'List of packages- Page d'accueil Laboratoire'

Une fois l'utilisateur authentifié, il accède à l'interface 'List of packages – Page d'accueil Laboratoire' décrite sur la FIGURE 4-5, là où il pourra visualiser tous ses paquets. La version web dispose d'une barre supérieure pour la navigation, tandis que la version mobile comporte une barre latérale de navigation et aussi des cartes à défilement horizontal, pour un accès facile aux différentes pages.

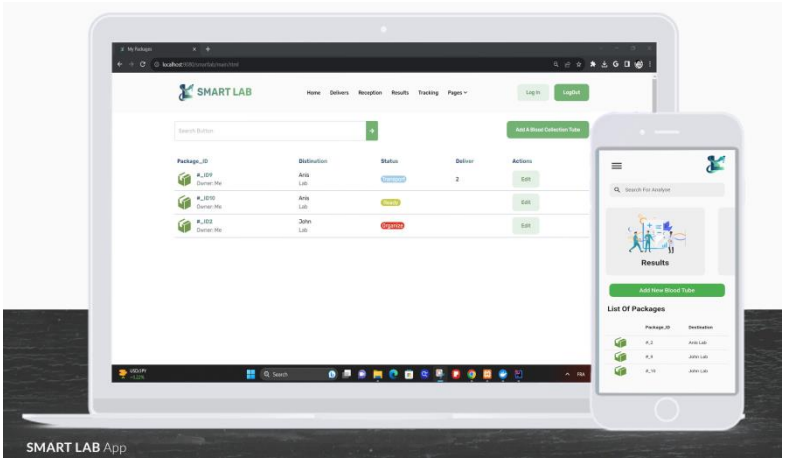


FIGURE 4-5-Interface 'List of packages- Page d'accueil Laboratoire'

4.4.5 Interface ' Création du nouveau tube d'analyse'

Dans cette interface, telle que illustrée sur la FIGURE 4-6, l'employé du laboratoire saisit toutes les informations du patient, avec le type d'analyse. L'utilisateur a la possibilité de choisir un paquet existant ou bien de créer un nouveau paquet et aussi de choisir le laboratoire destinataire qui offre ce type d'analyse.

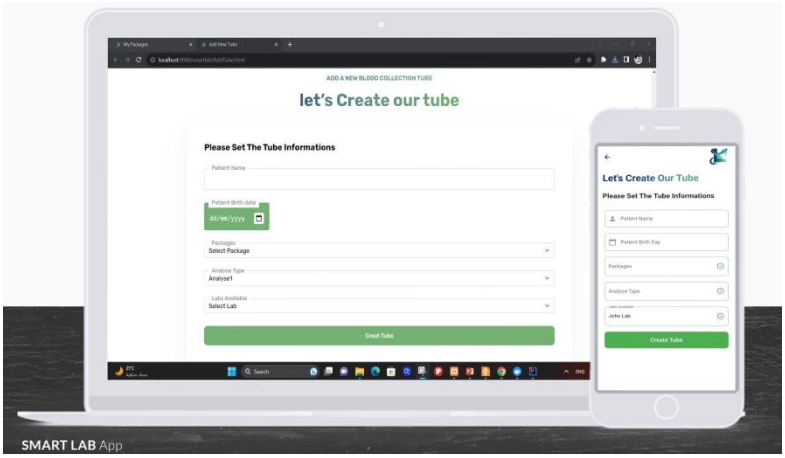


FIGURE 4-6-Interface 'Création du nouveau tube d'analyse'

4.4.6 L'interface 'Packages Information'

Lorsque l'utilisateur clique sur un paquet de la page d'accueil, il sera redirigé directement vers la page 'Packages Information' illustrée sur la FIGURE 4-7, ou il trouve toutes les informations concernant ce paquet.

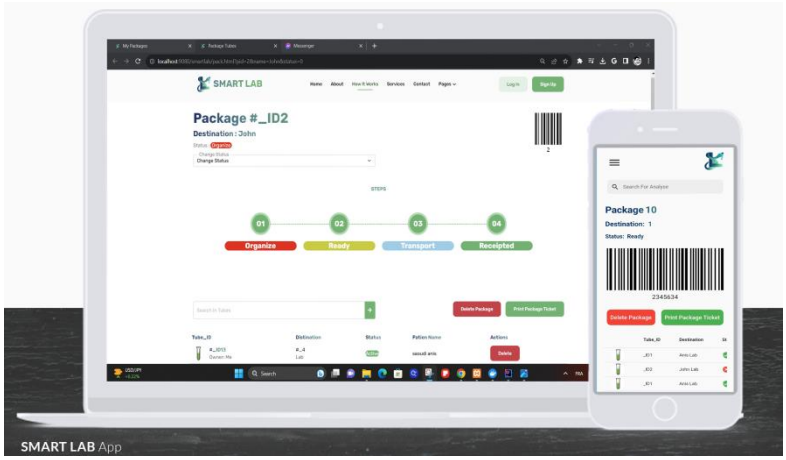


FIGURE 4-7-Interface 'Packages Information'

4.4.7 L'interface ' Tube information'

La FIGURE 4-8 illustre l'interface 'Tube Information' où l'utilisateur peut visualiser les informations du tube.

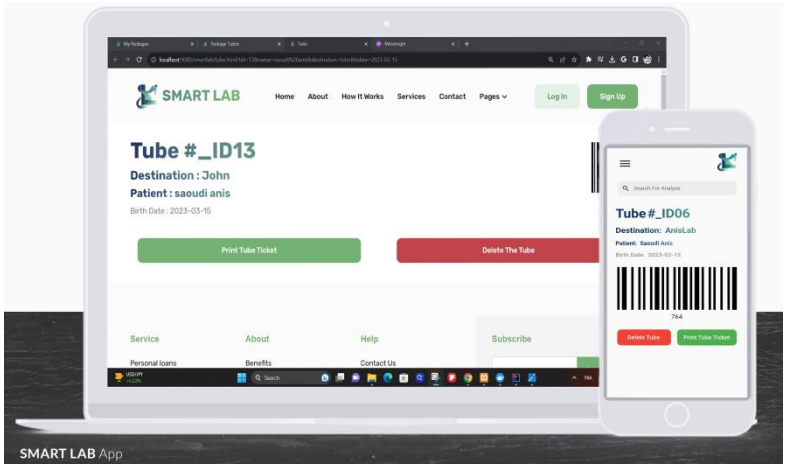


FIGURE 4-8-Interface 'Tube information'

4.4.8 L'interface 'Liste des livreurs'

Dans cette interface présenté sur la FIGURE 4-9 le laboratoire pourra visualiser la liste de tous ses livreurs.

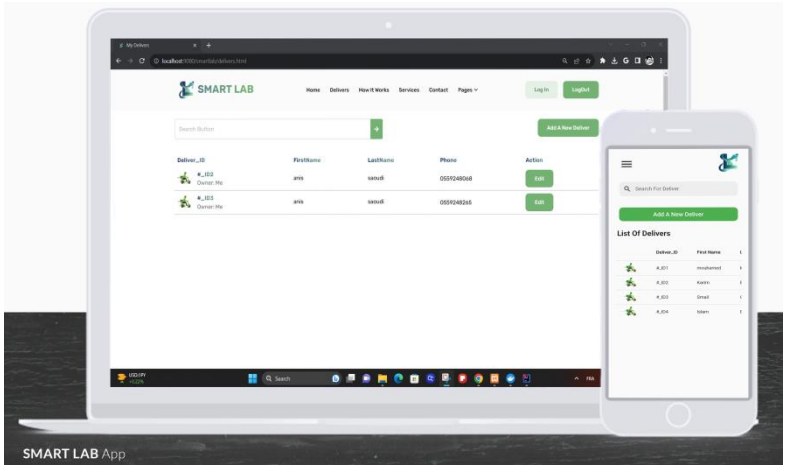


FIGURE 4-9-Interface 'Liste des livreurs'

4.4.9 L'interface 'Ajouter un compte livreur'

La FIGURE 4-10 illustre l'interface de création du compte livreur. Le gestionnaire de laboratoire remplit les champs des informations de son livreur puis il clique sur le bouton 'Createmy new deliverer'.

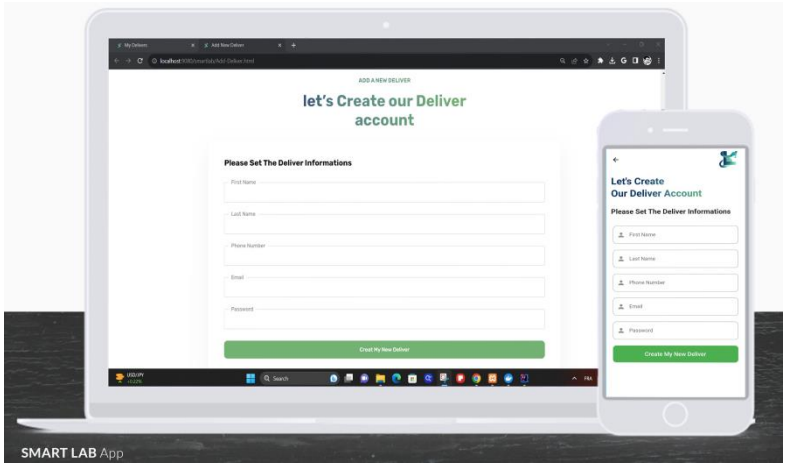


FIGURE 4-10-Interface 'Ajouter un compte livreur'

4.4.10 L'interface 'Mise à jour des informations du livreur'

La FIGURE 4-11 illustre l'interface de mise à jour des informations du livreur où le gestionnaire de laboratoire peut modifier les informations de son livreur.

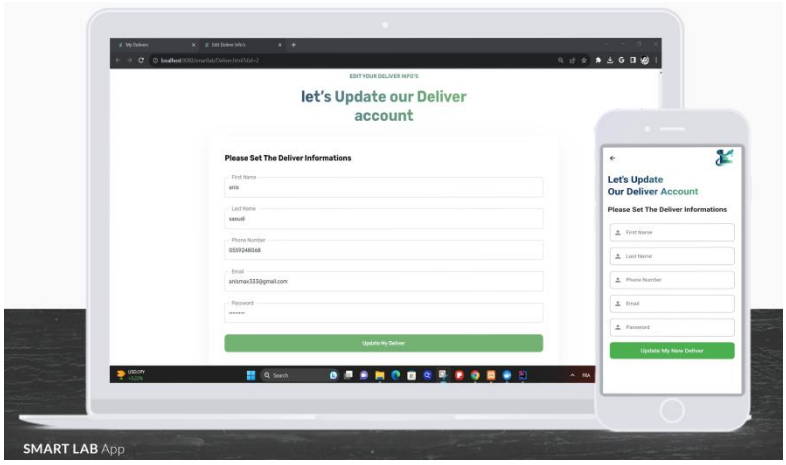


FIGURE 4-11-Interface 'Mise à jour des informations du livreur'

4.4.11 L'interface 'Réception'

La FIGURE 4-12 illustre la liste des paquets que devrait recevoir le laboratoire.

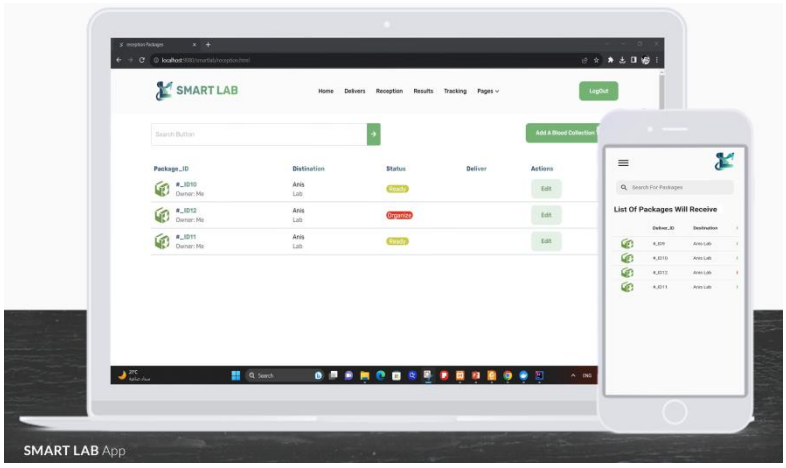


FIGURE 4-12-Interface 'Réception'

4.4.12 L'interface ' Télécharger résultat'

L'interface 'Télécharger résultat' montrée sur la FIGURE 4-13 permet à l'employé du laboratoire d'accéder aux informations de chaque tube du paquet et de télécharger et envoyer le résultat correspondant.

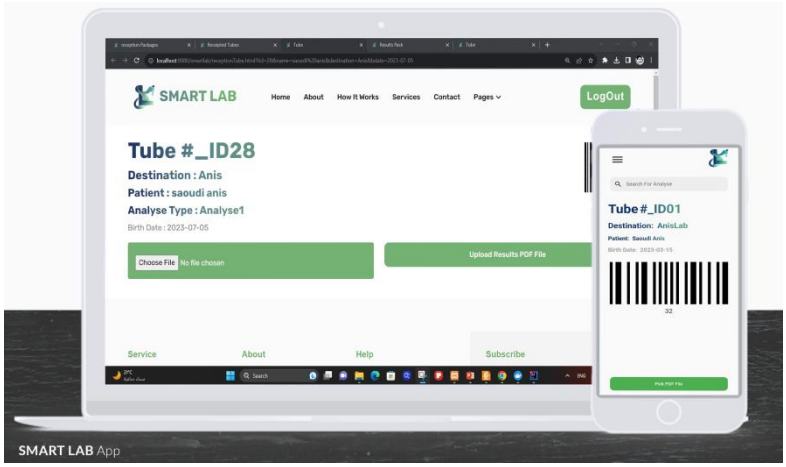


FIGURE 4-13-Interface ' Télécharger résultat'

4.4.13 L'interface ' Résultats'

La FIGURE 4-14 illustre l'interface 'Résultats' là où l'employé du laboratoire peut visualiser la liste de tous les résultats des paquets qu'il a déjà envoyés.

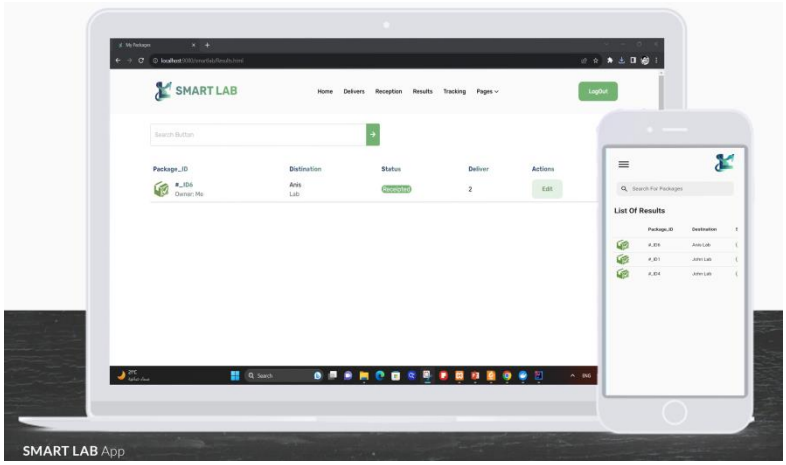


FIGURE 4-14-Interface ' Résultats'

4.4.14 L'interface ' Tracking'

La FIGURE 4-15 présente l'interface 'Tracking' contenant la liste des paquets avec toutes les informations nécessaires (date et heure exacte de sortie et de réception du paquet ainsi que le livreur qui a pris le paquet).

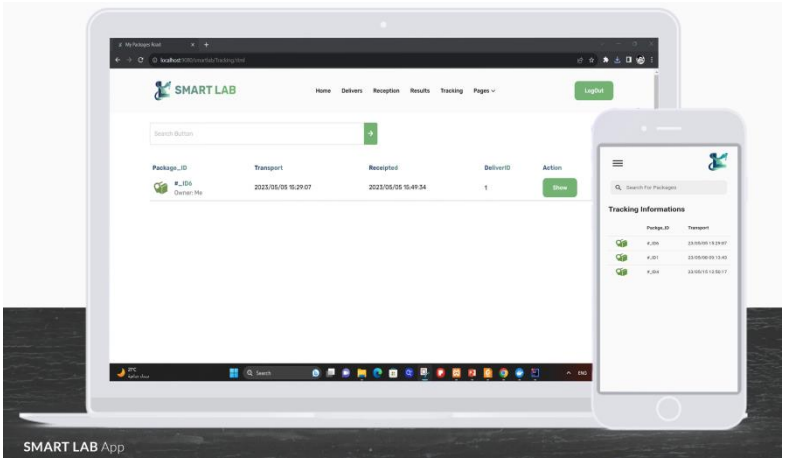


FIGURE 4-15-Interface 'Tracking'

4.4.15 L'interface ' Application mobile- Authentification- Log in Livreur'

La FIGURE 4-16 montre l'interface 'Application mobile- Authentification- Log in Livreur' là où l'utilisateur (Livreur) doit saisir correctement son nom d'utilisateur et son mot de passe pour accéder à l'application.

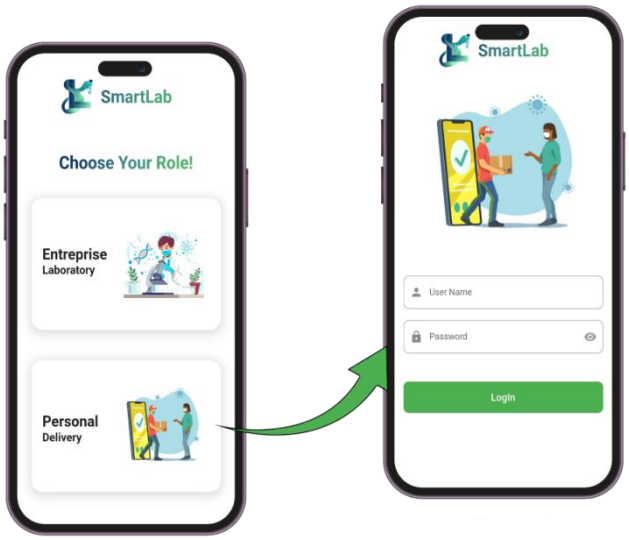


FIGURE 4-16-Interface 'Application mobile- Authentification – Log in Livreur'

4.4.16 L'interface ' Application mobile- Page d'accueil livreur – Informations des paquets'

Cette interface est présenté sur la FIGURE 4-17. Une fois l'utilisateur authentifié, il accède à l'interface 'Page d'accueil Livreur', là où il pourra visualiser tous les paquets prêts. Lorsque l'utilisateur clique sur un paquet de la page d'accueil,

il sera redirigé directement vers la page des informations du paquet, ou il trouve toutes les informations concernant ce paquet. Pour scanner le code bar du paquet l'utilisateur clique sur le bouton 'Scan Bar Code' et il sera redirigé vers la page du scanne.

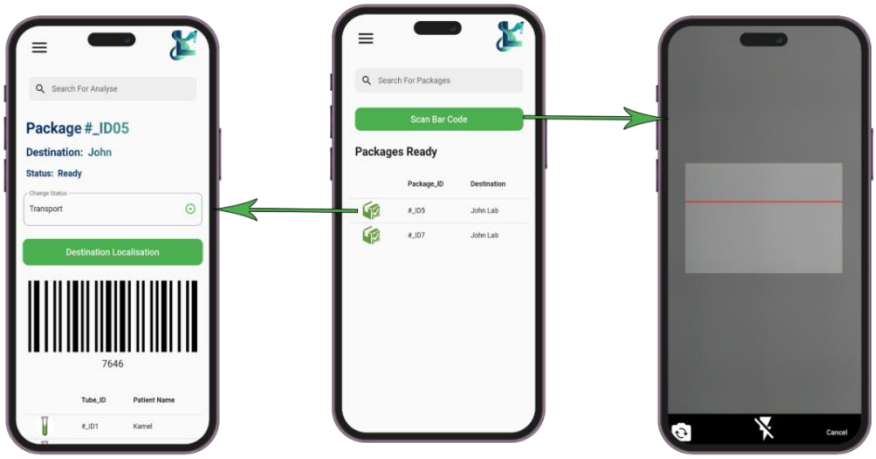


FIGURE 4-17-Interface ' Page d'accueil livreur – Informations des paquets'

4.4.17L'interface ' Application mobile – Localisation de la destination '

Cette interface est illustrée sur la FIGURE 4-18. Lorsque l'utilisateur clique sur un paquet de la page d'accueil, il sera redirigé directement vers la page des informations du paquet, où il trouve toutes les informations concernant ce paquet. Pour accéder à la localisation du laboratoire destinataire, l'utilisateur clique sur le bouton 'Destination Location' et il sera redirigé vers l'application Google Maps du téléphone avec les coordonnés du laboratoire destinataire.

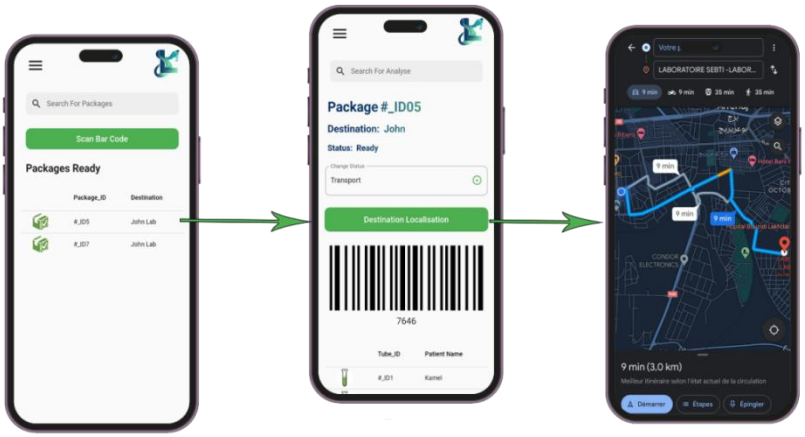


FIGURE 4-18-Interface ' Localisation de la destination'

4.4.18 L'interface ' Application web – Accueil administrateur/ Création du compte du laboratoire'

Cette interface présenté sur la FIGURE 4-19, permet à l'administrateur de visualiser la liste des laboratoires d'analyses médicales inscrits sur l'application 'SMARTLAB' ainsi que toutes leurs données (ID du laboratoire, nom du laboratoire, email, numéro de téléphone.). Lorsque l'administrateur clique sur le bouton 'Add New Laboratory' de la page d'accueil, il sera redirigé directement vers la page de création de compte de laboratoires. Cette interface permet à l'administrateur de créer les comptes des nouveaux laboratoires. Pour ajouter la liste des analyses fournis par le laboratoire inscrits dans l'application 'SMARTLAB' l'administrateur clique sur le bouton 'Edit' de la page d'accueil et il sera redirigé vers l'interface de sélection des analyses.

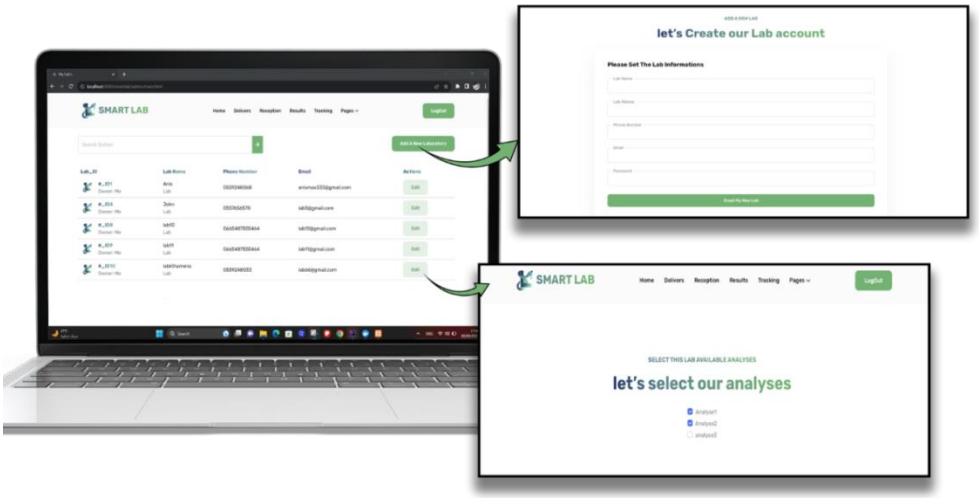


FIGURE 4-19-Interface ' Application web – Accueil administrateur/ Création du compte laboratoire'

4.5 Conclusion

La phase de réalisation est l'étape la plus importante dans le cycle de vie d'une application. Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement, les outils et les langages de développement associés à notre système avec une représentation de l'architecture globale de notre application mobile & web « SMARTLAB ». En effet, nous avons achevé l'implémentation tout en respectant la conception élaborée.

Conclusion générale et perspectives

Conclusion générale et perspectives

Avec le développement accéléré de l'informatique, les applications mobiles & web sont de plus en plus utilisées dans pratiquement tous les domaines de notre vie quotidienne et tous les secteurs socio-économiques notamment le secteur de santé auquel nous nous intéressons dans ce travail. La gestion des laboratoires d'analyses médicales a réalisé un avancement faramineux grâce aux apports de l'outil informatique. En outre, l'usage d'applications mobiles & web au sein de ces laboratoires a permis une meilleure prise en charge des malades, une gérance plus efficace et une qualité de service nettement améliorée.

L'état de l'art sur l'actualité de l'informatisation des laboratoires d'analyses médicales nous a permis de constater un manque d'efficacité dans la prise en charge des transactions inter-laboratoires concernant les sous-traitances. le problème réside dans le manque du suivi et l'absence d'une communication efficace et d'une coordination étroite entre le laboratoire d'analyses médicales et le sous-traitant, ce qui peut entrainer des retards dans la réception des résultats d'analyses, des pertes de tubes de sang ou même des erreurs de transmission des échantillons.

Vu l'impact de cette problématique sur la réussite du processus d'analyses médicale, nous avons contribué, dans ce travail, au développement d'une application mobile & web (disponible sous Android et iOS), nommée 'SMARTLAB', basée sur une architecture microservices, qui prend en charge une bonne partie des déficiences qui entrave la réalisation des sous-traitances. Celle-ci a pour rôle de gérer les sous-traitances (Paquets/ Tubes) d'une manière simple, rapide efficace et sécurisée. Elle est capable de garantir la bonne communication et interaction entre les laboratoires d'analyses médicales et entre les managers et les employés. De plus, elle peut être utilisée à tout moment, en tous lieux avec facilité et confiance.

Pour cela, nous avons en premier lieu présenté les deux domaines: l'architecture microservices/ Cloud computing et le rôle des applications mobiles & web pour la gestion des laboratoires d'analyses médicales. Ensuite, nous avons fait une description du cadre du projet tout en présentant la méthodologie de conception en l'occurrence UML comme langage de modélisation.

Nous avons établi par la suite, une étude préliminaire pour identifier les différents acteurs qui interagissent avec le système à réaliser, suivi de la spécification

des besoins fonctionnels à travers un diagramme de cas d'utilisation, de séquence et de classe.

Enfin, les outils et les langages de développement mobile que nous avons utilisés pour implémenter notre application ont été exposés.

Ce projet nous a été très bénéfique, car nous avons enrichi nos connaissances sur les deux plans : théorique et pratique. Il nous a aussi permis de découvrir et d'acquérir de nouvelles connaissances en matière de développement mobile & web, et aussi d'explorer de nouveaux concepts comme l'architecture microservices, cloud, conteneurisation….

Finalement on peut imaginer de nombreuses perspectives pour améliorer ce système, on peut citer par exemple :

1. L'ajout de la localisation GPS avec l'option de suivre le livreur en temps réel.
2. L'ajout d'une messagerie pour une meilleure communication entre les équipes.
3. L'ajout de l'option « ajouter plusieurs types d'analyses médicales sur le même tube d'analyse ».
4. L'ajout de la gestion financière entre les laboratoires en ajoutant l'option de la facturation, où chaque laboratoire peut voir ces propres factures.
5. L'introduction de l'intelligence artificielle lors de choisis des laboratoires en fonction des types d'analyse (le plus proche, rapidité des résultats…).

Tout ça va augmenter la fiabilité, et si on arrive à ce stade on peut envisager l'utilisation de ce nouveau moyen pour une excellente gestion des laboratoires d'analyses médicales

Références

Références

[1]https://www.lemagit.fr/definition/architecture-monolithique, consulté le 06/08/2023

[2]https://appmaster.io/fr/blog/architecture-monolithique-vs-microservices, consulté le 06/08/2023

[3]https://martinfowler.com/articles/microservices.html, consulté le 13/05/2023

[4]https://www.leanix.net/fr/wiki/vsm/architecture-microservices,consulté le 13/05/2023

[5]https://www.atlassian.com/fr/microservices/microservices-architecture/microservicesvs-monolith,consulté le 13/05/2023

[6] https://aymax.fr/pourquoi-une-architecture-de-microservices/, consulté le 14/05/2023

[7]https://www.spiceworks.com/tech/devops/articles/what-are-microservices/,consulté le 06/08/2023

[8]https://www.alter-solutions.fr/blog/adopter-larchitecture-microservices, consulté le 18/05/2023

[9] https://www.optisolbusiness.com/insight/8-core-components-of-microservice-architecture, consulté le 10/08/2023

[10] https://fr.linkedin.com/pulse/redis-kafka-ou-rabbitmq-quel-message-broker-pour-les-choisir, consulté le 10/08/2023

[11]https://semaphoreci.com/blog/deploy-microservices, consulté le 10/08/2023

[12] https://elearning.univ-bba.dz/enrol/index.php?id=3835 Cour Big Data Cloud Computing Madame Khelifi Hakima Université BBA, consulté le 02/06/2023

[13] https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf, consulté le 10/08/2023

[14] https://www.redhat.com/fr/topics/cloud-computing/iaas-vs-paas-vs-saas, consulté le 10/08/2023

[15] https://techtoday.lenovo.com/fr/fr/solutions/smb/hyperviseur#:~:text=Les%20diff%C3%A9rents%20types%20d'hyperviseurs,-Types%20d'hyperviseurs&text=Les%20hyperviseurs%20de%20type%201%20sont%20aussi%20appel%C3%A9s%20hyperviseurs%20natifs,un%20syst%C3%A8me%20d'exploitation%20h%C3%B4te, consulté le 11/08/2023

[16] https://www.techtarget.com/searchitoperations/definition/virtualization, consulté le 11/08/2023

[17]https://www.redhat.com/fr/topics/cloud-native-apps/what-is-containerization, consulté le 11/08/2023

[18]https://www.hebergeurcloud.com/conteneurs-avantages-inconvenients/, consulté le 11/08/2023

[19]https://sante-medecine.journaldesfemmes.fr/faq/20654-laboratoire-d-analyse-medicale-definition, consulté le 16/04/2023

[20]https://europaternite.fr/laboratoire-biomedical-les-differents-services-dun-labo/, consulté le 16/04/2023

[21]https://www.doctena.com/fr-be/blog/limpact-des-applications-mobiles-au-niveau-dusecteur-des-soins-de-sante, consulté le 16/04/2023

[22]https://www.gisnt.org/pdf/60082750e6cb9-0-Numerocomplet.pdf, consulté le 16/04/2023

[23]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4020637/,consulté le 16/04/2023

[24]https://labcollector.com/covid19-pack-fr/, consulté le 20/05/2023

[25]https://labcollector.com/labcollector-lims/features/, consulté le 20/05/2023

[26]https://apkgk.com/fr/com.livewirekiosk.lwk.labtestsonline, consulté le 20/05/2023

[27]https://www.rcpath.org/profession/publications/college-bulletin/july-2023/lab-tests-online-uk-a-vital-patient-resource.html, consulté le 20/05/2023

[28]https://www.labware.com/fr/,consulté le 23/05/2023

[29]https://www.labware.com/fr/lims, consulté le 23/05/2023

[30]https://www.softwareadvice.com/lim/polytech-lis-profile/, consulté le 10/06/2023

[31]https://www.getapp.fr/software/113395/polytech-lis, consulté le 10/06/2023

[32]Chantal morley, Jean hugues, Bernard le blanc. UML2, pour l'analyse d'un Système d'information 4e édition, 2009.

[33]Pascal Roques, Les cahiers du programmeur UML2 modélisé une application web, Eyrolles, 2007,4ème édition.

[34] https://fr.wikipedia.org/wiki/Figma, consulté le 18/08/2023

[35]http://www.mosaique-info.fr/glossaire-web-referencement-infographie-multimediainformatique/p-glossaire-informatique-et-multimedia/228-photoshop-definition.html, consulté le 18/08/2023

[36] https://fr.wikipedia.org/wiki/Flutter_(logiciel), consulté le 18/08/2023

[37] https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203555-java-definition/, consulté le 18/08/2023

[38] https://www.ibm.com/fr-fr/topics/java-spring-boot, consulté le 18/08/2023

[39] https://blog.desdelinux.net/fr/keycloak-una-solucion-de-gestion-de-acceso-e-identidad-de-codigo-abierto/, consulté le 18/08/2023

[40] https://sql.sh/, consulté le 18/08/2023

[41] https://www.oracle.com/fr/database/definition-postgresql/, consulté le 18/08/2023

[42] https://actualiteinformatique.fr/data/definition-nosql-not-only-sql-database, consulté le 18/08/2023

[43] https://www.mongodb.com/fr-fr/what-is-mongodb, consulté le 18/08/2023

[44] https://aws.amazon.com/fr/docker/, consulté le 18/08/2023

[45] https://www.postman.com/product/what-is-postman/, consulté le 18/08/2023