Un Peuple – Un But – Une Foi

REPUBLIQUE DU SENEGAL

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE ET DE L'INNOVATION

UNIVERSITE GASTON BERGER DE SAINT-LOUIS

U.F.R DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE

CENTRE DE FORMATION PROFESSIONNELLE PLURIDISCIPLINAIRE

RAPPORT DE STAGE DE FIN D'ETUDES POUR L'OBTENTION DU

MASTER PROFESSIONNEL RESEAUX ET TELECOMMUNICATIONS

Sujet : Conception et réalisation d'un système sécurisé de gestion des campagnes de vaccination

Présenté le 01/04/2021 par :

NEMA SIDI MED MAOULOUD

Sous la direction de :

Dr. Chérif DIALLO

Jury :

Président :	M. Cheikh Tidiane DIENG
Examinateur :	M. Abdoulaye DEME
Examinateur :	M. Mahamadou TRAORE
Examinateur :	M. Cheikh CISSE
Examinateur :	M. Abdou Karim MBODJI
Encadrant :	M. Chérif DIALLO

Année académique 2019/2020

Remerciements

En préambule à ce rapport je rende grâce à ALLAH le tout puissant et miséricordieux de m'avoir donné la foi, la volonté et la patience tout au long de mes études pour la réalisation de ce travail et de m'avoir mis en relation avec des personnes d'une qualité exceptionnelle.

Je tiens à exprimer mes sincères remerciements à :

Pr. Cherif Diallo enseignant et encadreur académique, pour l'orientation, la confiance, la patience, les conseils et remarques constructives, ainsi que ses qualités professionnelles, qui ont constitué un apport considérable, sans lequel ce travail n'aurait pas pu être mené au bon port.

Nous tenons aussi à remercier, **Mme Mariem Yahya** mon encadreuse au niveau du Ministère de la Santé pour son aimable accueil au sein de son entreprise pour élaborer ce stage, pour son suivi et ses conseils durant de ce stage.

Je tiens aussi à exprimer ma profonde reconnaissance à tous les professeurs de Master qui m'ont enseigné au cours des années passées, pour leurs compétences et la formation d'excellence qu'ils m'ont offert afin que je puisse entamer ma carrière future avec des bases solides.

Enfin, j'exprime mes remerciements et ma gratitude à tous les enseignants, à tout le personnel administratif et technique et à tous ceux qui de loin ou de près ont contribué à la réalisation de ce travail.

Dédicaces

Toutes les lettres ne sauraient trouver les mots qu'il faut, tous les mots ne sauraient exprimer, la gratitude, l'amour, le respect, la reconnaissance, je dédie ce travail à :

- Mes chers parents pour leurs sacrifices, leur patience illimitée, leurs encouragements continus, trésors de bonté, de générosité et de tendresse, que Dieu leur donne longue vie ;
- Tous mes enseignants à Université de Gaston Berger
- Mes frères et sœurs pour leurs encouragements ;
- Mes très chers amis pour les instants inoubliables ;
- Tous ceux qui m'ont enseigné et à toute personne qui a contribué à l'élaboration de ce travail.

Résumé

Durant une campagne de vaccination qui se passe sur le territoire Mauritanien le PEV (Programme Elargi de Vaccination) souffre de plusieurs difficultés telles que la mauvaise gestion, la perte de temps et parfois des données.

Le présent rapport est le fruit de mon travail dans le cadre de mes études de fin de formation Master à l'UGB (Université de Gaston Berger). Ce stage a été effectué au sein du Ministère de la Santé.

Actuellement, le constat et perceptuelle dans le domaine de la vaccination ou demeurent préoccupantes les attentes en vue d'une solution efficiente et efficace qui disponibilise des vaccins performantes, les exemples sautent à l'œil, il s'agit entre autres de la mauvaise organisation du travail, des retards accusés dans le traitements des données et de leur perte c'est ainsi qu'il est urgent d'introduire des outils informatiques capables de faciliter cette mission et de rendre la gestion des campagnes de vaccinations optimale et salutaire.

Le travail consiste à développer une application sécurisée avec deux parties : web et mobile pour la gestion des campagnes de vaccination qui va apporter donc une solution adéquate au PEV.

Abstract

During a vaccination campaign taking place on Mauritanian territory, the EPI (Extended Vaccination Program) suffers from several difficulties such as the loss of time sometimes of data and poor management.

This report is the result of my work as part of the end of master's training studies at UGB, this internship was carried out within the health ministry in Mauritania.

the job, therefore, consists of developing a secure web and mobile application for the management of vaccination campaigns which will therefore provide an adequate solution to the EPI.

Table des matières

Chapitre 1 : Contexte du projet 11
I. Organisme d'accueil 11
a. L'organigramme du Ministére de la Santé 11
1. Le Ministre de la Santé 11
2. Le Cabinet du Ministre 12
3. Inspection Général 12
4. Les Directions centrales 13
5. Les Directions Régionales 13
6. Les établissements publics 13
II. Présentation du sujet 14
a. Présentation du programme élargie de vaccination(PEV) 14
b. Recueil et analyse des données lors d'une campagne de vaccination 15
c. Problématique 16
d. Objectifs du projet 16
Conclusion 16
Chapitre 2 : Analyse et Conception 17
I. Spécification des besoins 17
a. Les besoins fonctionnels 17
b. Les besoins non fonctionnels 17
II. Méthodologie de travail 18
a. La méthodologie Agile 18
b. Srum 18
III. Identification des acteurs du système 19
IV. Langage de modélisation 20
a. Définition de UML 20
b. Diagrammes UML utilisés 20
V. Modélisation des besoins fonctionnels 21
a. Diagramme de cas d'utilisations 21
b. Diagrammes de séquence 27
c. Diagramme de classes 34
Conclusion 35

Chapitre 3 : Réalisation du système 36
I. Environment de travail 36
II. Technologies et Outils de développement 38
III. Architecture du Système 42
a. Architecture 3 tiers 42
b. Architecture du système coté client 43
c. Architecture du système coté server 44
d. Architecture d'android 45
e. Architecture de cette application 46
IV. Présentation des interfaces du système 47
a. Application web 47
b. Application mobile 51
c. La création de l'APK et le déploiement 52
c.1 La création de l'APK 52
c.2 Le déploiement 53
Conclusion 54
Chapitre 4 Evaluation du niveau de sécurité de l'application 55
I. Les attaques 55
a. Attaque passive 55
b. Attaque active 55
II. Présentations des outiles utilisés 55
III. Test d'évaluation 57
a. Présentations de l'environnement de travail 55
Conclusion 63
Conclusion générale 64
Webographie 65

Glossaire des acronymes

-PEV : Programme Elargi de Vaccination

-AVS : Activité de vaccination supplémentaire

-CSS : Cascading Style Sheet

-EDI(IDE) : Environnement de Développement Intégré

-HTML: HyperText Mark up Language

-HTTP: HyperText Transfert Protocol

-JWT: Json Web Token

-MySql: My Structured Query Language

-SGBD : Système de Gestion de Bases de Données

-SQL: Structured Query Language

-TS: Type Script

-UML: Unified Modeling Language

-WAMP: Windows Apache MySQL PHP

Liste des figures

Figure 1: l'organigramme du Ministére de la Santé 11
Figure 2: Circulation des données de vaccinations 15
Figure 3: Scrum 19
Figure 4: Diagramme de cas d'utilisation global 21
Figure 5: Diagramme de cas d'utilisation Gérer les utilisateurs 22
Figure 6: Diagramme de cas d'utilisation Gérer les démographies 22
Figure 7: Diagramme de cas d'utilisation Gérer les campagnes 23
Figure 8: Diagramme de cas d'utilisation Gérer la traçabilité 23
Figure 9: Diagramme de cas d'utilisation La collecte de données 24
Figure 10: Diagramme de séquence Authentification 27
Figure 11: Diagramme de séquence Gérer les utilisateurs 28
Figure 12: Diagramme de séquence Gérer les démographies 29
Figure 13: Diagramme de séquence Gérer les campagnes 30
Figure 14: Diagramme de séquence Consulter les statistiques 31
Figure 15: Diagramme de séquence Consulter la MAP 32
Figure 16: Diagramme de séquence Envoyer les données collectés 33
Figure 17: Diagramme de classe 35
Figure 18: Logo StartUml 36
Figure 19: Logo Intellij 36
Figure 20: Logo WampServer 37
Figure 21: Logo android studio 37
Figure 22: Logo Spring boot 38
Figure 23: Logo Spring Data JPA 38
Figure 24: Logo OpenLayer 39
Figure 25: Logo MySQL 39
Figure 26: Logo JWT 39
Figure 27: Logo Angular 40
Figure 28: Logo Retrofit 40
Figure 29: Logo Java 41
Figure 30: Logo Maven 41
Figure 31: Logo Bootstrap 41
Figure 32: Logo Chart.js 42
Figure 33: Logo Android 42
Figure 34: Logo SQLite 42
Figure 35: Architecture 3 tiers 43
Figure 36: Architecture d'angular 44
Figure 37: Architechture coté serveur 44
Figure 38: Architechture d'Android 45
Figure 39: l'Architechture de l'application 46
Figure 40: Interface d'Authentification 47
Figure 41: Interface de Gestion des utilisateurs 47
Figure 42: Interface de Gestion des démographies 48
Figure 43: Interface de Détail d'une démographie 48
Figure 44: Interface de Gestion des campagnes 49

Figure 45: Interface de détail d'une campagne 49
Figure 46: Interface des Statistiques 50
Figure 47: Interface de Tracabilite 50
Figure 48: Interface d'Authentification 51
Figure 49 et 50: Interfaces de Collecte de données 51
Figure 51: Interface des Vaccinations 52
Figure 52: Première étape de la création de l'APK 52
Figure 53: Dernière étape de la création de l'APK 53
Figure 54: Le déploiement sur un serveur Tomcat 53
Figure 55: Logo NMAP 55
Figure 56: Logo kali Lunix 56
Figure 57: Logo VMware 56
Figure 58: Logo Wireshark 56
Figure 59: Logo CentOS 56
Figure 60: La topologie implémentée 57
Figure 61: Une requête affiche les vaccins 58
Figure 62: Les adresses utilisés 58
Figure 63: L'activation de IP forward router 59
Figure 64: Scan NMAP 59
Figure 65: Arpsoof 60
Figure 66: Une requête affiche les wilayas 60
Figure 67: Une requête sur la port 8443 61
Figure 68: L'interception des données 61
Figure 69: Capture du paquet sur wireshark 61
Figure 70: Figure1 62
Figure 71: Figure2 62

Liste des Tableaux

Table 1: Diagrammes utilisés 20
Table 2: Description textuelle du cas d'utilisation <<Gérer les utilisateurs>> 24
Table 3: Description textuelle du cas d'utilisation <<Gérer les démographies>> 25
Table 4: Description textuelle du cas d'utilisation <<Gérer les campagnes>> 25
Table 5: Description textuelle du cas d'utilisation <<Consulter la carte géographique>> 26
Table 6: Description textuelle du cas d'utilisation <<Collecte des données>> 26
Table 7: Description textuelle de diagramme de classe 34

Introduction générale

L'informatique représente une révolution importante et innovante qui a marqué la vie de l'humanité ces dernières décennies. Elle offre à tout type d'environnement professionnel, tant de services, aussi bien pour l'administration ou les autorités gouvernementales que pour les entreprises et le personnel.

Le domaine de la santé fait partie des domaines où l'informatique pourra beaucoup aider et plus précisément dans le domaine de vaccination.

En effet avec les outils informatiques, la collecte et l'exploitation des données du domaine de la vaccination seront beaucoup plus faciles.

A ce jour, on constate encore dans le domaine de la vaccination, beaucoup de préoccupations non résolues telles que la mauvaise organisation du travail, les retards dans le traitement et les pertes de données d'où la nécessité d'introduire un moyen informatique performant permettant de faciliter les tâches et améliorer efficacement la gestion des campagnes de vaccination.

Vu cet état de fait, le présent stage de fin d'études vise à répondre aux difficultés posées et à améliorer les moyens de collecte et d'exploitation des données par un système de gestion et de suivi de campagnes de vaccination.

Le présent rapport comporte quatre parties :

- La première partie est consacrée à la présentation du contexte du projet, la problématique posée ainsi que la solution à adopter ;
- La deuxième partie est réservée à l'analyse des besoins et à la conception du projet ;
- La troisième partie concerne la réalisation, les outils et les technologies pour la solution adéquate ;
- La quatrième partie porte sur la sécurité et la protection de l'application contre tous genre d'attaques.

Chapitre 1 Contexte du projet

Dans ce chapitre, je vais tout d'abord présenter l'organisme d'accueil, ensuite je vais procéder à la présentation du cadre du projet pour permettre de mieux comprendre la problématique étudiée et présenter le principe de fonctionnement du système.

I. Organisme d'accueil

J'ai effectué mon stage au sein du Ministère de la Santé en Mauritanie qui s'occupe de préparer et mettre en œuvre la politique du gouvernement dans le domaine de la santé.

a. L'organigramme du Ministère de la Santé

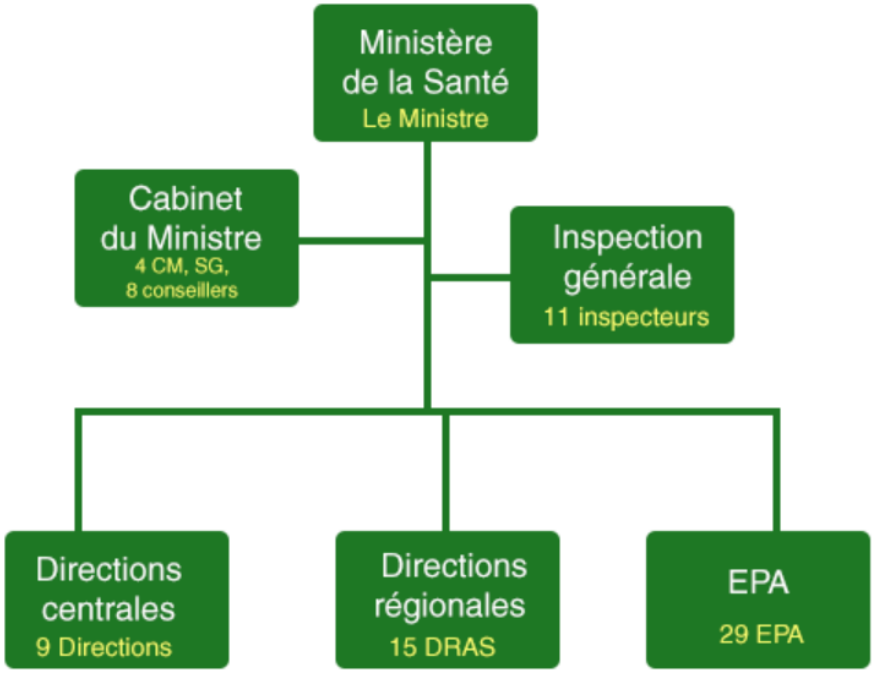


Figure 1 : L'organigramme du Ministère de la Santé

1. Le Ministre de la Santé

Le cadre organisationnel de l'administration du département de la santé est établi par un décret du Premier Ministre fixant les attributions du Ministre de la Santé et l'organisation de l'administration centrale du département. Le système de santé en Mauritanie est de type pyramidal, avec trois niveaux de prestations.

2. Le Cabinet du Ministre

Le Cabinet du ministre comprend deux chargés de mission, sept conseillers techniques, une Inspection interne et le Secrétariat particulier.

Le Secrétariat particulier du ministre gère les affaires réservées du ministre.

Le Secrétariat particulier est dirigé par un Secrétaire particulier nommé par arrêté du ministre, ayant rang et mêmes avantages des chefs de service centraux.

Un Attaché de Cabinet sera chargé d'accomplir les tâches qui lui seront confiées dans le cadre de la répartition du travail au sein du cabinet.

Le Secrétariat Général comprend :

- Le Secrétaire général ;
- Les services rattachés au Secrétaire Général.

Les Services rattachés au Secrétaire Général :

- Le Service de la Traduction ;
- Le Service de l'Informatique ;
- Le Service du Secrétariat central ;
- Le Service Accueil du Public

3. Inspection Général

Inspection Interne de la Santé est chargée, sous l'autorité du ministre, des missions de l'inspection interne.

- Dans ce cadre, elle a notamment pour attributions de :
- Vérifier l'efficacité de la gestion des activités de l'ensemble des services du Département et des Organismes sous tutelle et leur conformité aux lois et règlements en vigueur ainsi qu'à la politique et aux programmes d'actions prévus dans les différents secteurs relevant du Département.
- Évaluer les résultats effectivement acquis, analyser les écarts par rapport aux prévisions et suggérer les mesures de redressement nécessaires.
- Veiller au respect des textes législatifs et réglementaires régissant les activités médicales et pharmaceutiques. Elle rend compte au ministre des irrégularités constatées. L'Inspection Interne est dirigée par un Inspecteur général qui a rang de conseiller technique du Ministre. L'Inspecteur Général de la Santé est assisté par 9 inspecteurs qui ont rang de Directeurs de l'Administration centrale chargés respectivement des missions suivantes :

- L'Inspection médicale : 03 Inspecteurs
- L'Inspection de la Pharmacie et du Médicament : 03 Inspecteurs
- L'Inspection administrative et financière : 03 Inspecteurs

4. Les Directions centrales

Les Directions Centrales du Ministère sont :

- La Direction de la Programmation, de la Coopération et de l'Information Sanitaire ;
- La Direction de la Médecine Hospitalière ;
- La Direction de la Lutte contre les Maladies ;
- La Direction de la Santé de Base et de la Nutrition ;
- La Direction de la Pharmacie et des Laboratoires ;
- La Direction de l'Hygiène Publique ;
- La Direction des Ressources Humaines ;
- La Direction des Affaires Financières ;
- La Direction des Infrastructures, du Matériel et de la Maintenance.

5. Les Directions Régionales

Les Directions régionales de la Santé sont chargées sous l'autorité du Wali de la conception et de l'exécution de la politique sanitaire dans les wilayas.

Elles sont notamment chargées :

– de l'application de la politique nationale de santé,

– de la planification, de la coordination, de l'exécution, du suivi, et de l'évaluation de l'action sanitaire dans la wilaya,

– de la tenue et de la mise à jour des statistiques sanitaires de la wilaya ;

– des questions relatives à l'information sanitaire et à la surveillance épidémiologique ;

– de la supervision des formations médicales et pharmaceutiques publiques et privées implantées dans la wilaya ;

– de la gestion des moyens humains, matériels et financiers mis à leur disposition ;

– des questions relatives à l'éducation pour la santé, à l'hygiène, au contrôle de qualité, à la santé scolaire et universitaire ;

6. Les Etablissements Publics

EPA est l'organisme qui regroupe les Hôpitaux et établissements publics y compris les hôpitaux régionaux et les centres nationaux tel que :

- Centre National de transfusion sanguine
- Centrale d'Achats de Médicaments Essentiels et Consommables Médicaux
- Centre hospitalier mère et enfant
- Centre national de cardiologie
- Centre National d'Orthopédie et de Réadaptation Fonctionnelle
- Centre Hospitalier CHEIKH ZAYED
- Centre Hospitalier National

Ainsi que les établissements tel que :

- L'Etablissement National de Contrôle la qualité des médicaments
- La Caisse National d'Assurance Maladie

II. Présentation du sujet

a. Présentation du programme élargie de vaccination (PEV)

L'Organisation Mondiale de la Santé (OMS) a lancé en 1974 le Programme Elargi de Vaccination (PEV) dans le but de rendre la vaccination accessible à tous les enfants dans le monde.

Dans ce cadre en 1984 l'OMS a établi un calendrier standard pour les premiers vaccins du PEV : le DTCoq (diphtérie, tétanos, coqueluche), le BCG (bacille de Calmette et Guérin), le vaccin oral contre la poliomyélite, et la rougeole.

Les objectifs actuels de ce programme sont d'assurer l'immunisation totale des enfants de moins d'un an, diminuer de 50% le nombre de décès liés à la rougeole, et étendre tous les vaccins et les interventions de santé préventive pour les enfants dans toutes les régions du monde.

b. Recueil et analyse des données lors d'une campagne de vaccination

Durant l'Activité de Vaccination Supplémentaire **(AVS)**, les données administratives sont importantes comme source d'informations pour l'évaluation du nombre d'enfants vaccinés, l'utilisation du vaccin et d'autres données essentielles associées à d'autres données provenant de la supervision et du suivi, ces données doivent être analysées pour identifier à quel moment des mesures correctives seront prises au cours de l'AVS.

En conséquence, il convient d'établir un plan de transmission et d'analyse rapides des données relatives aux doses administrées durant l'AVS.

- Cartes de vaccination/fiches conservées à domicile : garder une trace

Il est déterminant d'enregistrer les vaccinations effectuées pendant l'AVS sur une carte de vaccination/fiche gardée à domicile si l'on veut vérifier l'exactitude de la couverture vaccinale pendant les enquêtes post AVS et déterminer le nombre total de doses de vaccin qu'un enfant a reçu.

- Feuilles de pointage

Sur les feuilles de pointage, les équipes de vaccination enregistrent chaque dose de vaccin administrée. Ces fiches doivent être simples à comprendre, faciles à utiliser, et ne nécessitant pas une formation importante. Afin de faciliter le comptage journalier des doses administrées, les cases seront de préférence groupées par cinq, et les pointages ventilés par tranche d'âge.

Le responsable du poste et le superviseur de l'équipe de vaccination devront à la fin de chaque journée examiner les feuilles de pointage et approuver si elles ont bien été remplies.

- La date, le lieu, le numéro de l'équipe et les membres de l'équipe seront clairement indiqués.
- Les nombres de flacons de vaccin reçus, utilisés, endommagés et restitués doivent être cohérents.
- Les pointages seront correctement notés, sans saut de lignes.

Les feuilles de pointage seront synthétisées et les données, transmises au superviseur de l'équipe, notamment :

- Le nombre total des enfants vaccinés et l'état d'avancement par rapport aux objectifs ;
- Les informations relatives au comptage des flacons de vaccin (flacons reçus, entamés, jetés et restitués par l'équipe) ;

- Formulaires de compte-rendu

Pour toutes les interventions réalisées durant l'AVS, les superviseurs et les équipes de vaccination devront analyser les données et les communiquer quotidiennement au niveau immédiatement supérieur.

Il faut prendre en compte que les retards de transmission des feuilles de pointage de certaines équipes sont souvent dûs au manque de moyens de déplacement ou de communication.

Le superviseur est chargé d'obtenir toutes les données manquantes et de les transmettre au niveau immédiatement supérieur dès que possible.

Enfin, toutes les feuilles de pointage seront réexaminées afin de vérifier que les résultats sont bien pris en compte dans le rapport final.

- Circulation des données

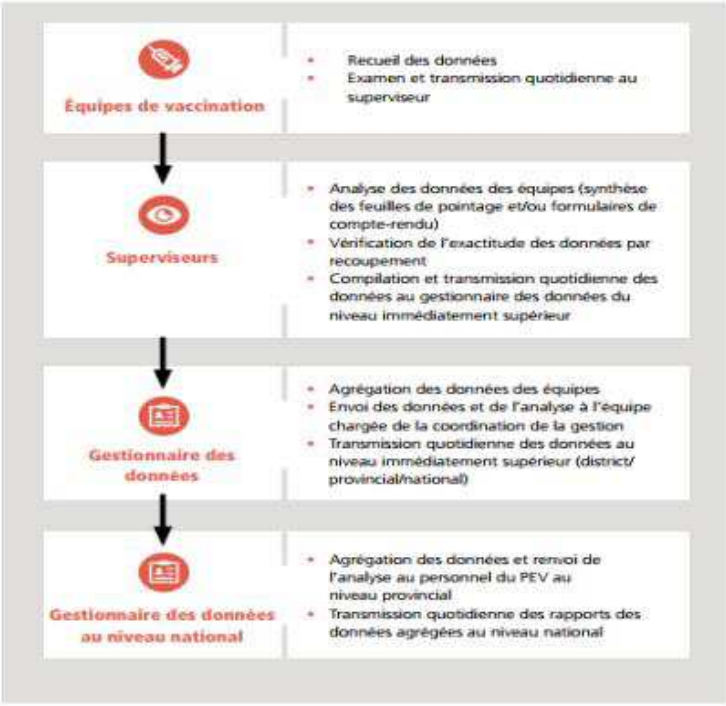


Figure 2 : Circulation des données de vaccinations

c. Problématique

Le remplissage des outils de collecte des données est effectué à tous les niveaux de la pyramide sanitaire mais sa qualité fait défaut surtout dans la partie réservée à la gestion des vaccins.

En effet d'après le PEV, les données collectées arrivent souvent en retard et parfois certaines données sont perdues en chemin sans parler du temps qu'elles mettent pour passer des équipes de vaccination au gestionnaire des données au niveau national.

Il existe des technologies informatiques capables de résoudre ces problèmes tel que DHS2 qui est un entrepôt de données conçu pour stocker et visualiser les données de santé essentielles, mais ces technologies nécessitent d'avoir une connexion internet ce qui peut générer des problèmes si le réseau est faible ou que les données à stocker sont très volumineuses pour la connexion.

En plus en Mauritanie l'accès au réseau d'internet est limité dans certains endroits. Je propose un système autonome, sécurisé indépendant de l'internet capable de faciliter la collecte, de visualiser les données en temps réel et minimum.

d. Objectifs du projet

L'objectif de ce projet est la conception et le développement d'une plateforme sécurisé de gestion et de suivi des campagnes de vaccination composée de deux parties :

-Une partie mobile nomade qui permet de collecter toutes les informations relatives à la vaccination (tel que : le nombre d'enfants vaccinés, leurs tranches d'âge, nom du vaccin etc…), les insère dans une base de données locale (Sqlite) et les envoie à la partie web.

- Une partie web qui récupère ces informations dans une base de données (Mysql) pour les afficher sur des graphes et sur une carte.

Conclusion

A travers ce premier chapitre j'ai présenté le projet dans son contexte général, le programme élargi de vaccination, les outils de recueil des données de vaccination, identifier la problématique et proposer la solution à adopter.

Dans le deuxième chapitre de ce rapport je vais traiter les thèmes relatifs à l'analyse et conception du projet, notamment la spécification des besoins, l'identification des acteurs, la méthodologie de développement, ainsi que le langage et la modélisation du système de gestion des campagnes de vaccination.

Chapitre 2 Analyse et conception

Dans ce chapitre je vais me focaliser sur l'analyse des besoins du système, l'identification des utilisateurs du système ainsi que la conception et les différents diagrammes que j'ai utilisé.

I. Spécification des besoins

Les besoins attendus de l'application

L'application envisagée doit satisfaire des besoins fonctionnels et des besoins non fonctionnels.

a. Les besoins fonctionnels

Les besoins fonctionnels représentent les actions que le système doit faire, donc il doit les satisfaire pour qu'il devient opérationnel.

Cette application doit répondre aux besoins fonctionnels suivants :

- Gestion des utilisateurs de l'application

- Gestion des campagnes de vaccination

- Gestion des démographies

- Consultation des statistiques

- Consultation de la traçabilité des vaccinateurs sur une carte géographique

- La géolocalisation

- La collecte des données de vaccination

b. Les besoins non fonctionnels

Il ne s'agit pas des exigences concernant spécifiquement le comportement du système mais plutôt la performance du système.

Les besoins non fonctionnels que l'application doive répondre sont :

- La clarté du code : il doit être clair pour des futures évolutions ou améliorations ;

- L'ergonomie : l'interface de notre application est conviviale et facile à utiliser ;

- La sécurité : l'application doit assurer la confidentialité des données ;

- Garantir la cohérence et l'intégrité des données au cours de chaque mise à jour et à chaque insertion.

II. Méthodologie de développement

a. La méthodologie agile

La méthode agile est une méthode très précise avec des règles, un cadre souple et adapté. Elle permet d'accompagner les mouvements d'un environnement changeant des projets de développement Web et informatique. Elle place les besoins du client au centre des priorités du projet.

Aujourd'hui, la méthode agile s'est considérablement améliorée et a beaucoup évolué. Elle est de plus en plus répandue car elle est applicable à de nombreux types de projets professionnels, tous secteurs confondus.

Les principales méthodes agiles :

• Scrum

• EXtreme Programming (XP)

• Rational Unified Process (RUP)

• Feature Driven Development (FDD)

• Rapid Application Development (RAD)

• Dynamic systems development method (DSDM)

b. SCRUM

J'ai choisi la méthode Scrum car elle prend en compte les grands principes de productivité, de simplicité, d'efficacité et de transparence. Scrum, permet de produire la plus grande valeur métier dans la durée la plus courte. Elle définit principalement trois rôles :

• Le Scrum Master : c'est le garant de la méthode son rôle est de faciliter le travail de son équipe, s'assurer que la méthode est bien respectée en impliquant chaque membre et en l'aidant dans la recherche et l'identification des solutions.

• L'équipe : est responsable de la réalisation du produit, comportant généralement un effectif réduit pour faciliter l'interactivité et le savoir-faire.

• Le Product Owner : c'est le représentant du client, il choisit la date et le contenu de la livraison, accepte ou pas cette dernière.

L'équipe va donc fonctionner sur la base du sprint. Un sprint est une période qui varie entre une et quatre semaines durant laquelle l'équipe délivre un incrément du produit. Un sprint fonctionne avec le déroulement suivant :

• Le sprint planning meeting : c'est une réunion Scrum qui débute le sprint, l'équipe sélectionne les fonctionnalités prioritaires qui devront être livrée au Product Owner au terme du sprint. Ces fonctionnalités constituent le carnet de sprint ou sprint backlog.

• La mêlée quotidienne : réunion de planification qui dure une quinzaine de minutes, permet à l'équipe de faire un point de coordination sur la réalisation et sur les difficultés rencontrées. La présence du Product Owner est facultative.

• Le sprint review meeting : une réunion qui permet à l'équipe de présenter au Product Owner l'incrément de produit qui a été réalisé pendant le sprint et c'est à lui de le valider ou non. [2]

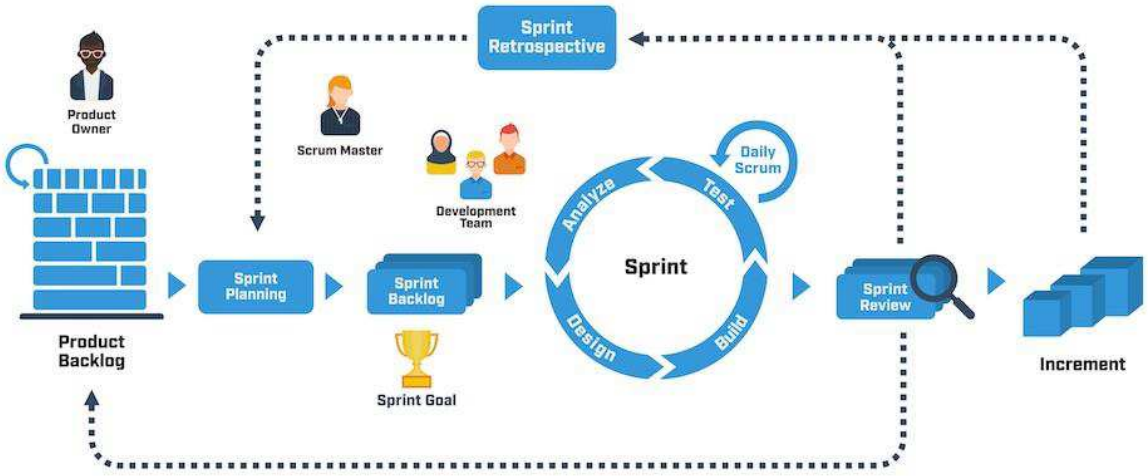


Figure 3 : Scrum

III. Identification des acteurs du système

Un acteur représente un rôle probablement joué par une personne qui interagit avec le système. Les acteurs se sont les utilisateurs du système et aussi les responsables de sa configuration et de sa maintenance. D'où, les acteurs potentiels qui peuvent interagir avec l'application

J'ai identifié 3 acteurs qui utiliseront ce système :

- L'administrateur : qui détient les droits et peut accéder à toutes les fonctionnalités de l'application.

- Le superviseur : qui peut consulter les statistiques et la carte géographique.

- L'agent vaccinateur : qui peut collecter les informations de vaccination.

IV. Langage de modélisation

a. Définition de UML

UML est un langage de modélisation objet né de la fusion de trois langages de modélisation objet Booch, OMT et OOSE qui ont révolutionné la modélisation. UML est dit universel car il est indépendant des langages de programmation, des domaines d'application et aussi du processus de développement adopté. Il permet de représenter un système sous forme de schémas. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de la solution. Sa véritable force repose sur un méta modèle qui normalise la sémantique des concepts qu'il véhicule. [3]

b. Diagrammes UML utilisés

Pour modéliser ce système j'ai utilisé les trois types de diagrammes suivants :

Diagramme de cas d'utilisation, Diagramme de séquence, Diagramme de classe.

Diagramme	Objectifs	Type
Diagramme de cas d'utilisation	1. Décrire la manière dont une organisation ou un système externe doit interagir avec le système. 2. Décrire ce que doit faire le système. 3. Mettre en évidence les services rendus par le système	Fonctionnel
Diagramme de séquence	4. Validation des cas d'utilisation pour comprendre la logique de l'application. 5. Complète le diagramme de cas d'utilisation en mettant en évidence les objets et leurs interactions d'un point de vue temporel.	Dynamique
Diagramme de classe	6. Point central de la modélisation du système pour décrire ce que le système doit faire (analyse) et avec quoi il va le faire (conception). 7. Représentation de la structure statique du système d'information. 8. Modélisation des classes et de leurs relations.	Statique

Table 1: Diagrammes utilisés

V. Modélisation des besoins fonctionnels

a. Diagramme de cas d'utilisations

Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation global représente les différentes fonctions de cette application autour desquelles, sont érigées les besoins et les exigences des différents acteurs qui interagiront au sein du système.

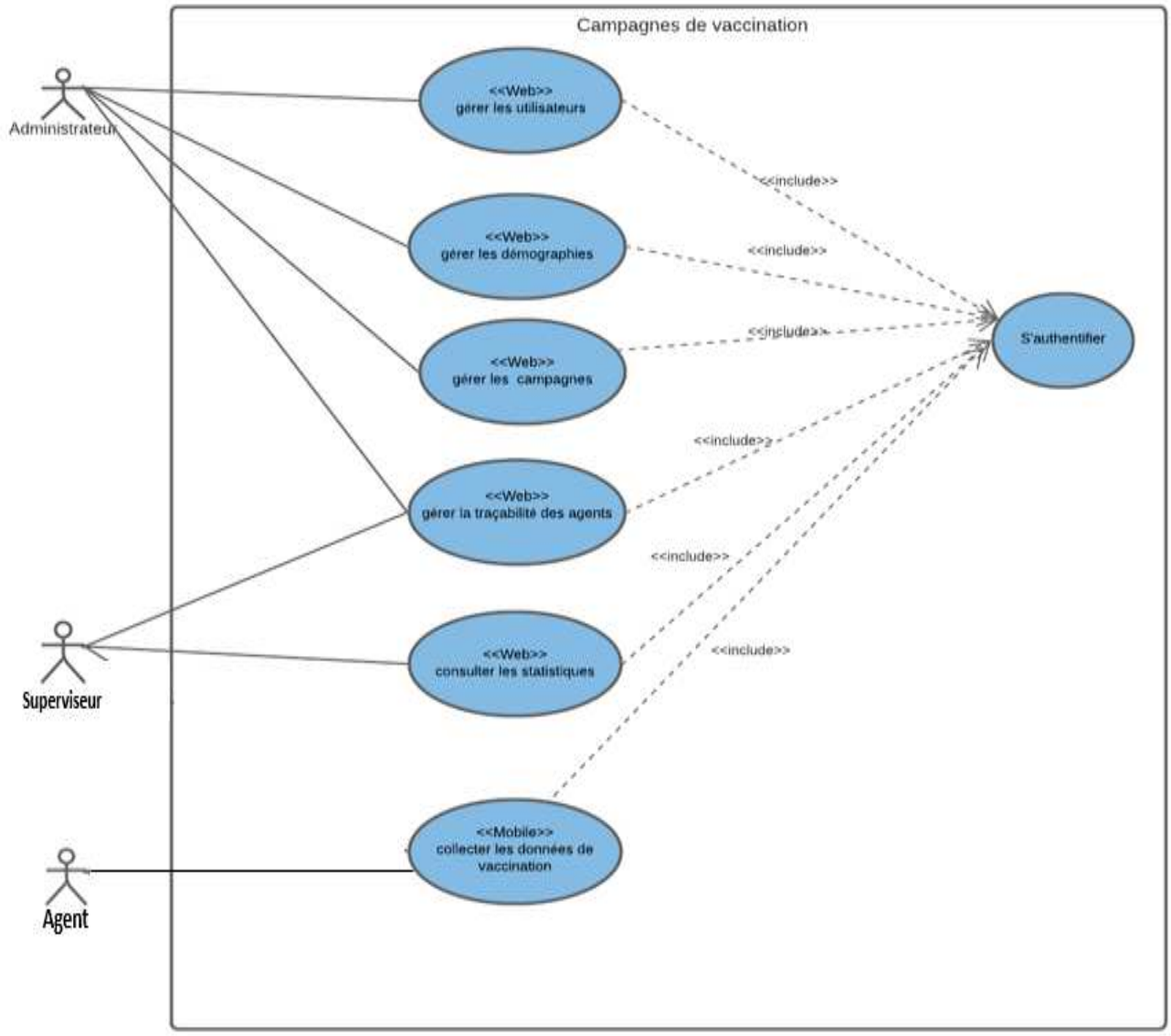


Figure 4 : Diagramme de cas d'utilisation global

Diagramme de cas d'utilisation détaillé

Ici je vais détailler les fonctions complexes de l'application.

❖ Gérer les utilisateurs

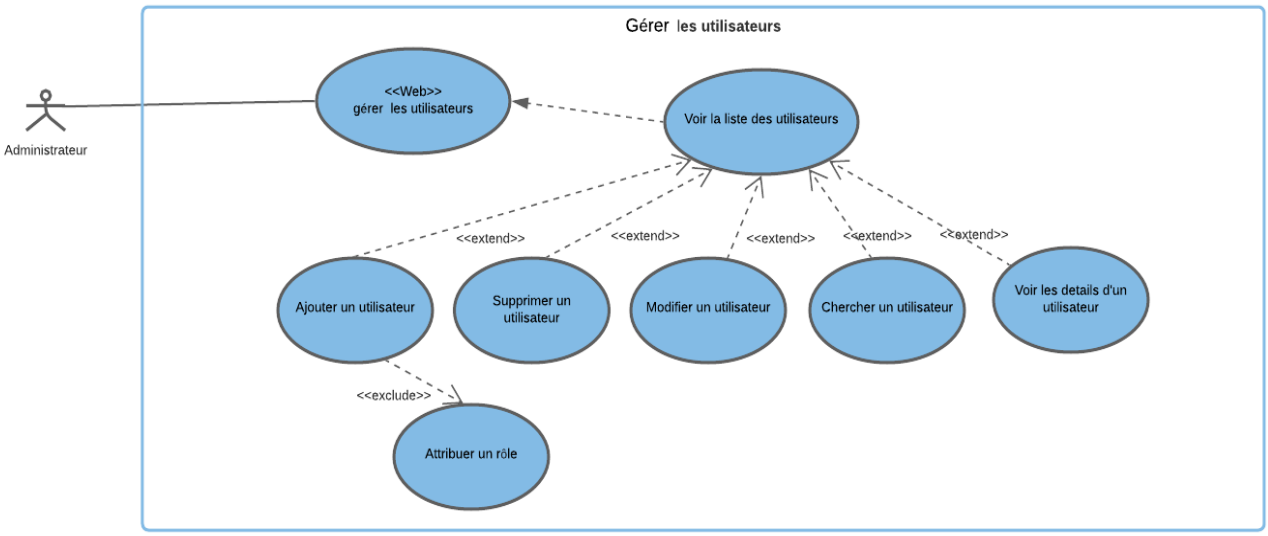


Figure 5: Diagramme de cas d'utilisation<< Gérer les utilisateurs>>

❖ Gérer des démographies

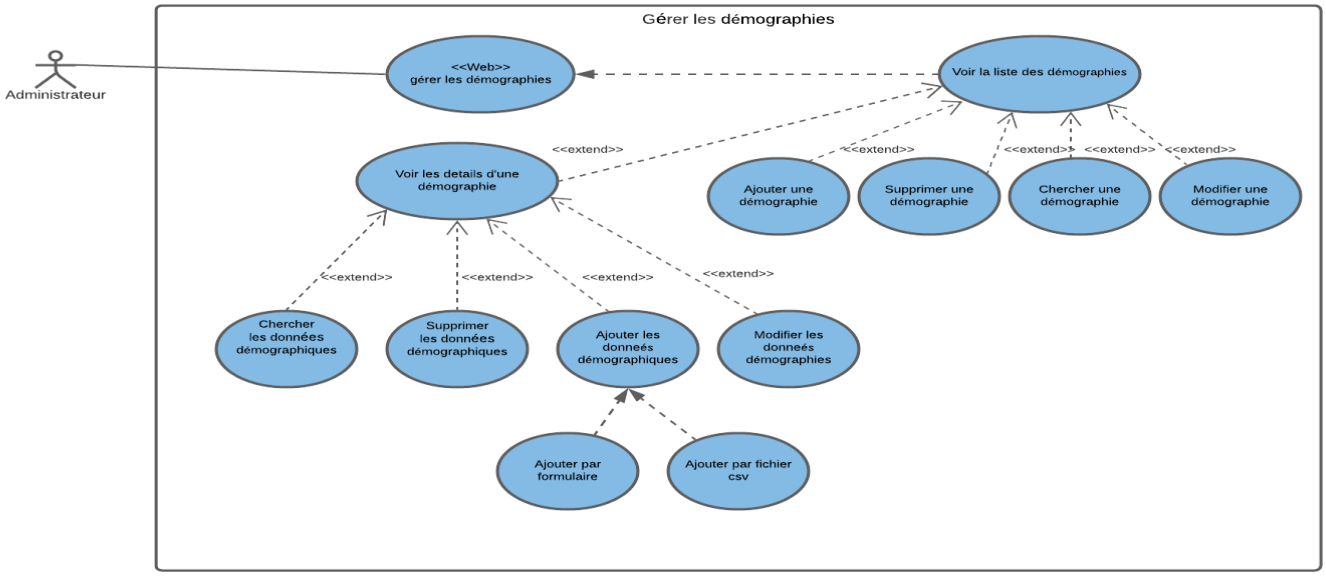


Figure 6 : Diagramme de cas d'utilisation<<Gérer les démographies>>

❖ Gérer les campagnes

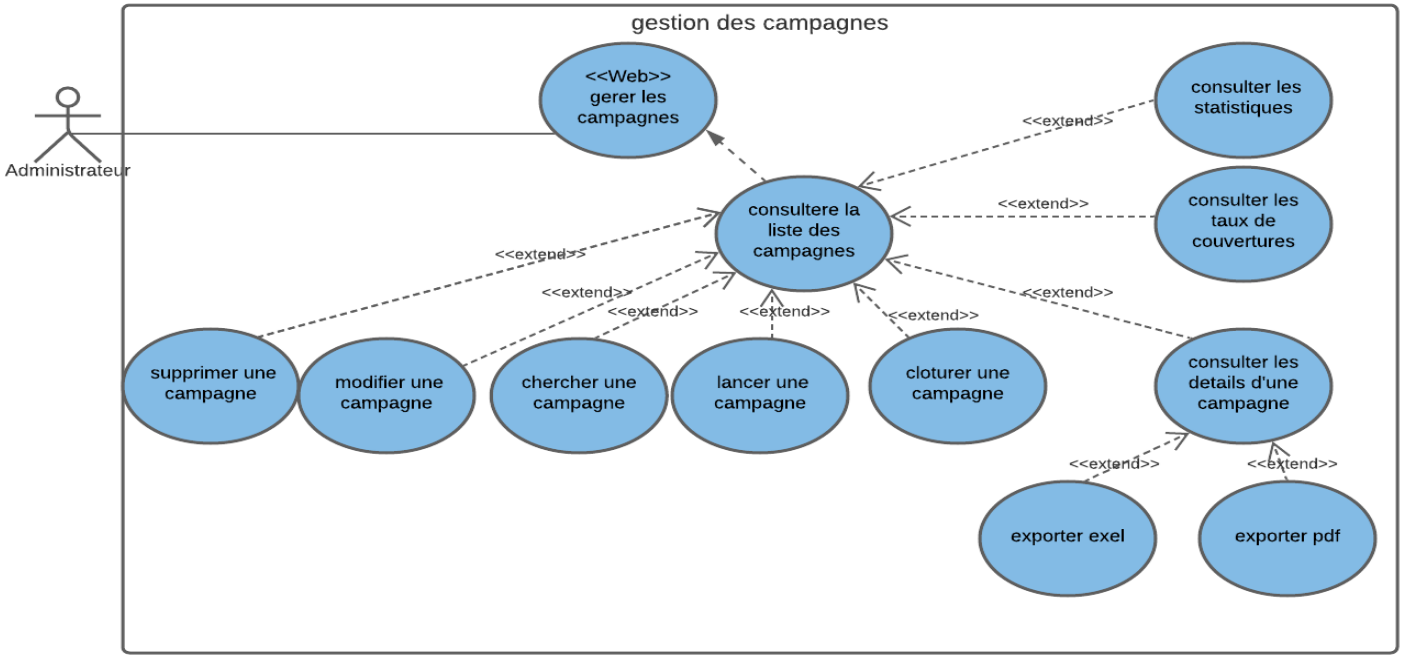


Figure 7 : Diagramme de cas d'utilisation<<Gérer les campagnes >>

❖ Gérer la traçabilité

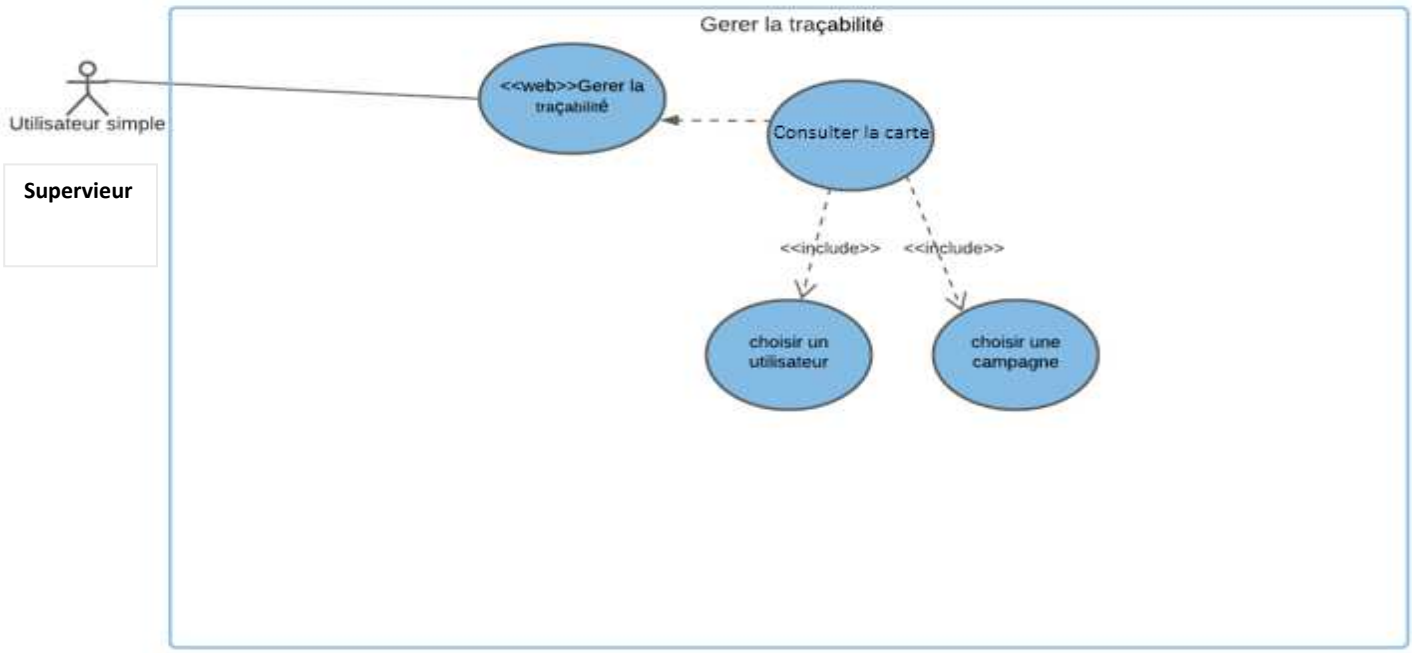


Figure 8 : Diagramme de cas d'utilisation<<Gérer la traçabilité>>

❖ La collecte des données

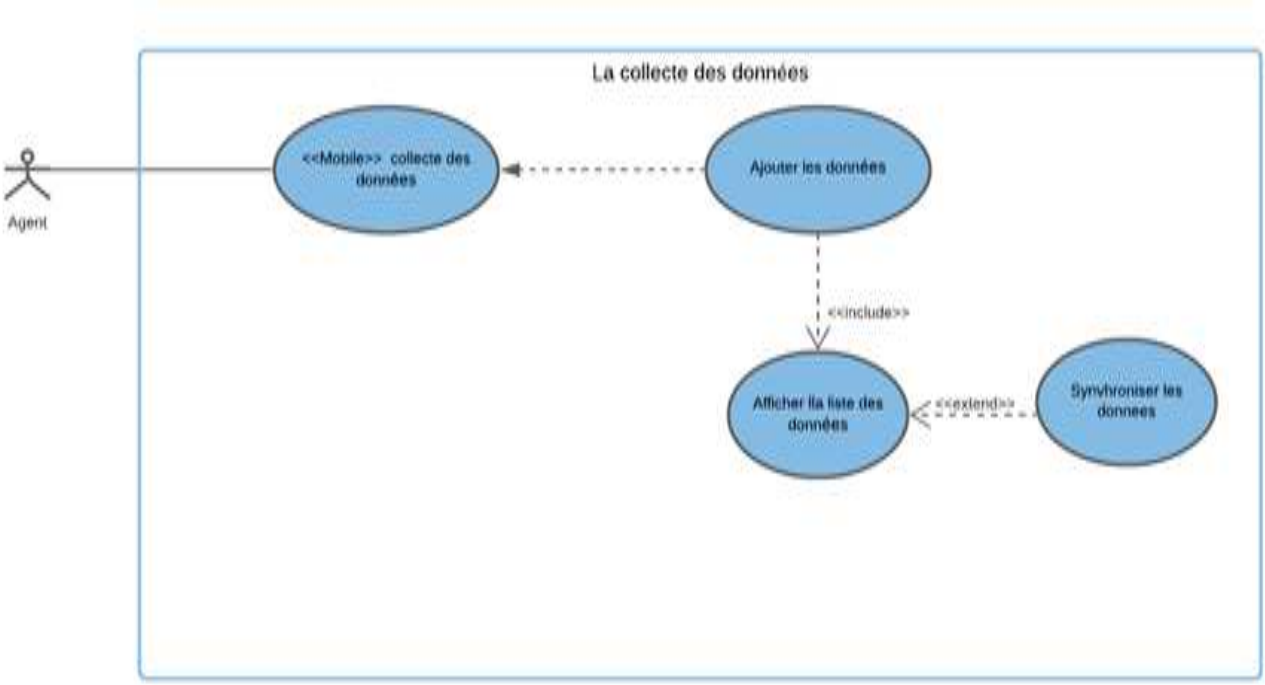


Figure 9 : Diagramme de cas d'utilisation <<La collecte des données>>

Description textuelle des cas d'utilisations

❖ Cas d'utilisation N° 1 : Gérer des utilisateurs

Acteur principal	Administrateur
Objectif	Ajouter un utilisateur.
Préconditions	Authentification.
Scénarios	1. L'administrateur demande l'accès à la gestion des utilisateurs. 2. Le système affiche la liste des utilisateurs. 3. L'administrateur clique sur le bouton ajouter. 4. Le système demande les informations du nouvel utilisateur. 5. L'administrateur fournit les informations 6. Le système vérifie les informations et envoie une réponse favorable tout en actualisant la liste des utilisateurs.
Alternative	En cas de réponse défavorable, le système affiche un message d'erreur en cas d'erreur de saisie ou bien d'un champ incomplet (retour à 4).

Table 2:Description textuelle du cas d'utilisation <<Gérer les utilisateurs>>

❖ Cas d'utilisation N°2 : Gérer les démographies

Acteur principal	Administrateur
Objectif	Ajouter une démographie et ses informations
Préconditions	Authentification
Scénarios	1. L'administrateur demande l'accès à la gestion des démographies. 2. Le système affiche la liste des démographies. 3. L'administrateur clique sur le bouton ajouter. 4. Le système demande les informations de la démographie 5. L'administrateur saisi les informations 6. Le système valide les informations et effectue une redirection vers les détails de la nouvelle démographie. 7. L'administrateur clique sur le bouton détails. 8. Le système propose l'ajout un à un et par csv. 9. L'utilisateur clique sur ajouter par csv et sélection un fichier 10. Le système vérifie la conformité des informations du fichier, envois une réponse favorable et met à jour la liste des démographies.
Alternative	En cas de réponse défavorable du serveur, le système affiche un message d'erreur en cas d'erreur de saisie ou bien d'un champ incomplet (retour à 4 ou 8).

Table 3: Description textuelle du cas d'utilisation <<Gérer les démographies>>

❖ Cas d'utilisation N°3 : Gérer des campagnes

Acteur principal	Administrateur
Objectif	Lancer une nouvelle campagne
Préconditions	Authentification.
Scénarios	1. L'administrateur demande l'accès à la gestion des campagnes. 2. Le système affiche la liste des campagnes. 3. L'administrateur clique sur le bouton ajouter. 4.Le système demande les informations de la campagne. 5. L'administrateur saisi les informations. 6. Le système valide les informations et effectue une redirection vers les détails de la nouvelle campagne. 7. L'administrateur clique sur le bouton exporter les détails. 8. Le système export les détails dans un fichier Excel ou PDF.
Alternative	En cas de réponse défavorable du serveur, le système affiche un message d'erreur en cas d'erreur de saisie ou bien d'un champ incomplet (retour à 4).

Table 4: Description textuelle du cas d'utilisation <<Gérer les campagnes>>

❖ Cas d'utilisation N°4 : Gérer la traçabilité.

Acteur principal	Administrateur/Superviseur
Objectif	Consulter la carte géographique.
Préconditions	Authentification.
Scénarios	1. L'administrateur demande l'accès à la carte. 2. Le système affiche la carte. 3. L'administrateur sélection un agent et une campagne. 4. Le système affiche les positions de cet agent durant cette campagne.

Table 5: Description textuelle du cas d'utilisation <<Gérer la traçabilité>>

❖ Cas d'utilisation N°5 : Envoyer les données collectées

Acteur principal	Agent
Objectif	Collecte et envoie des données
Préconditions	Authentification
Scénarios	1.L'agent saisi les données et clique sur envoyer. 2. Le système enregistre les données en local. 3. Le système affiche la liste des données. 4.L'agent clique sur le bouton synchroniser. 5. Le système envoie les données au serveur. 6.Le système supprime les données en local. 7.Le server insert les données dans base de données.

Table 6: Description textuelle du cas d'utilisation <<Envoyer les données collectés>>

b. Diagrammes de séquence

Diagramme de séquence N°1 : cas d'utilisation Authentification

Lorsque l'utilisateur (Administrateur, Utilisateur, Agent) souhaite accéder à ce système, il sera obligé de s'authentifier avant d'y accéder en saisissant son identifiant et mot de passe, après la saisie le système vérifie les informations envoyées, si les informations sont correctes l'utilisateur accédera au système si non un message d'erreur sera affiché et reconduira l'utilisateur à la page authentification.

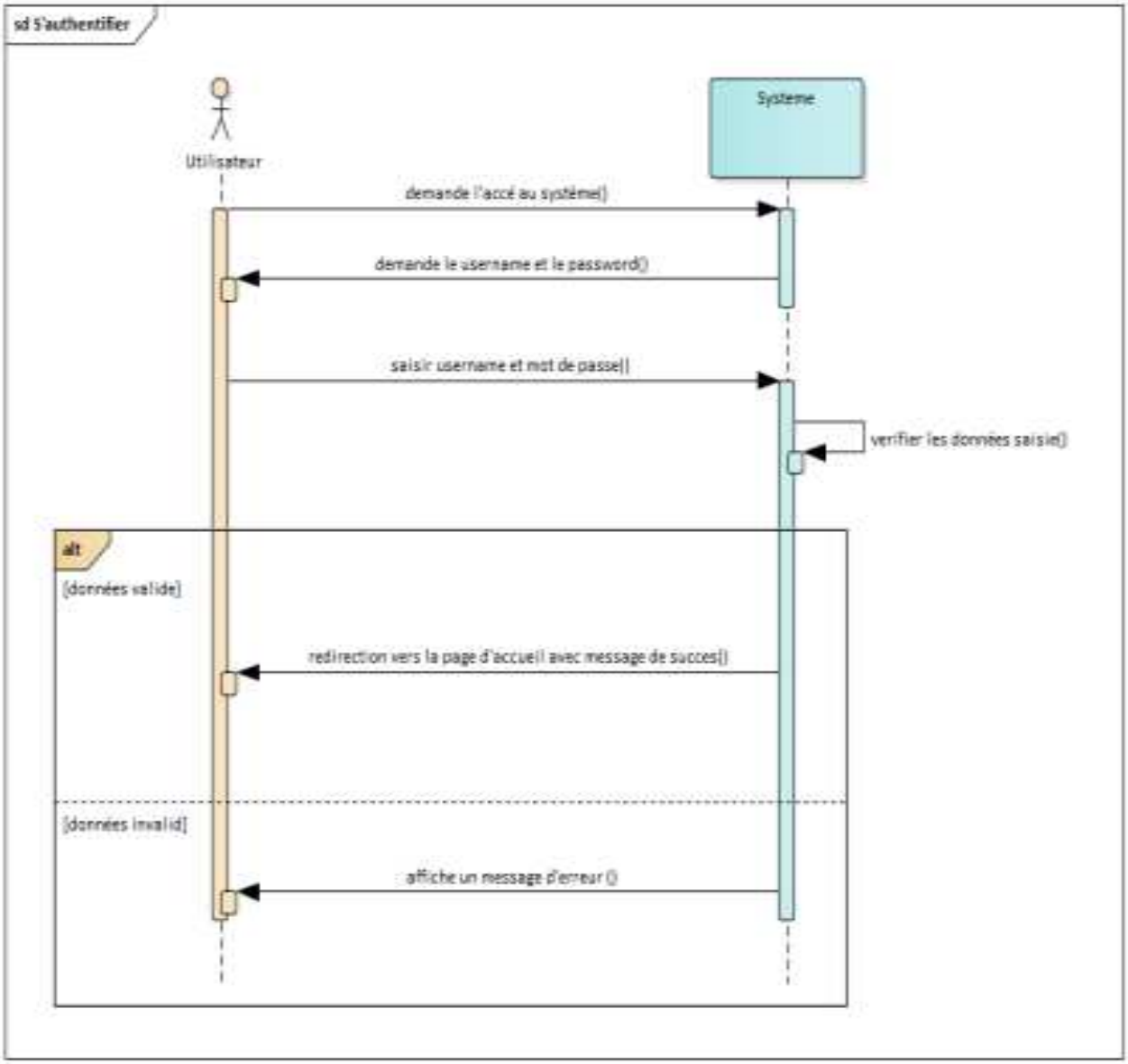


Figure 10 : Diagramme de séquence <<Authentification>>

Diagramme de séquence N°2 : cas d'utilisation Gérer les utilisateurs

Quand l'administrateur veut gérer les utilisateurs, il doit tout d'abord accéder à la gestion des utilisateurs puis il aura la possibilité d'ajouter, modifier, chercher ou supprimer un utilisateur. S'il choisit d'ajouter un utilisateur, il n'a qu'à cliquer sur le bouton ajouter et remplir les champs requis et cliquer sur enregistrer. Ensuite le système va traiter et enregistrer les informations fournies par l'administrateur. Un message d'erreur est envoyé en cas de champ incomplet sinon un message de succès.

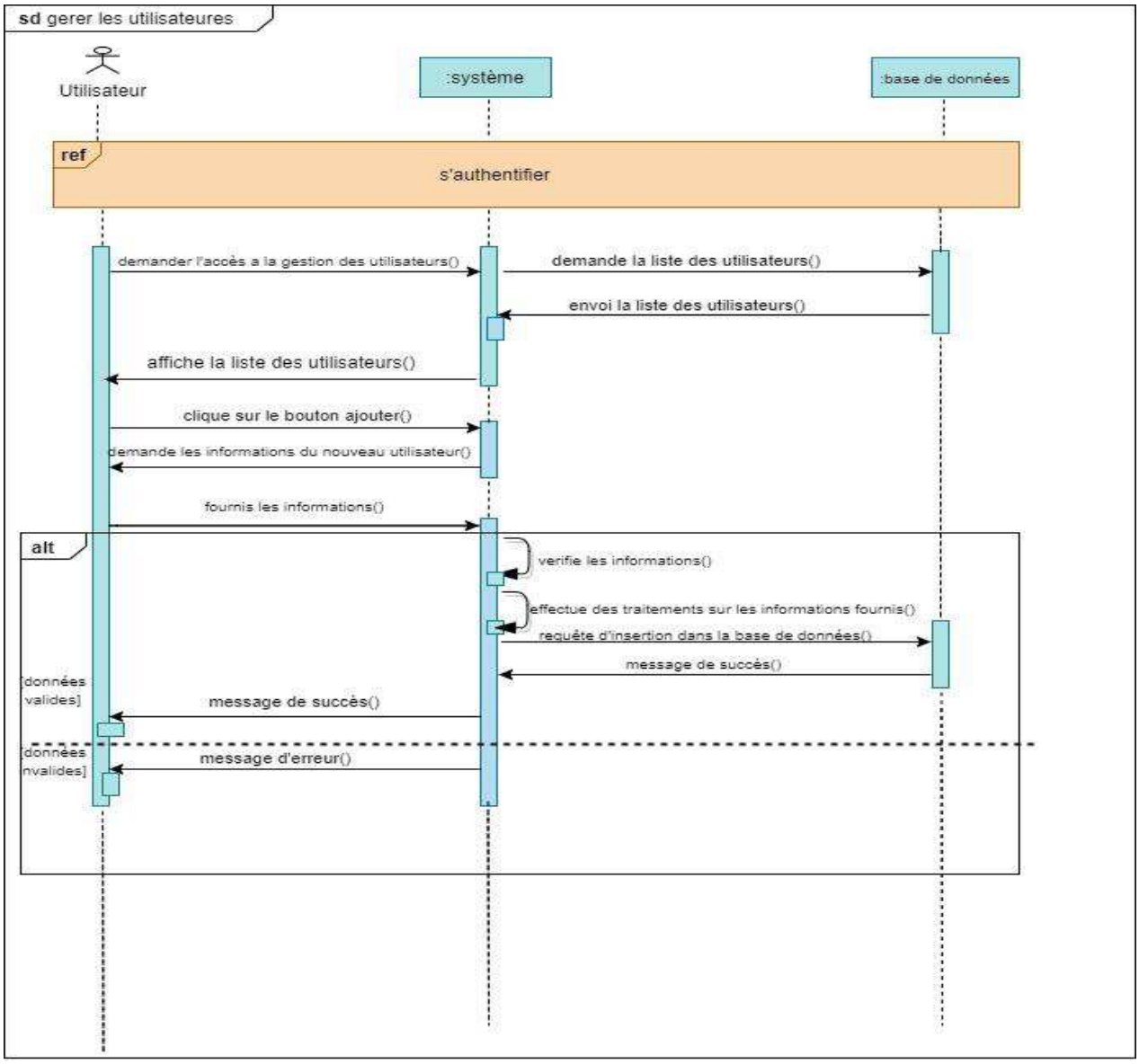


Figure 11 : Diagramme de séquence <<Gérer les utilisateurs>>

Diagramme de séquence N°3 : cas d'utilisation Gérer les démographies

Lorsque l'administrateur veut gérer les démographies, il doit tout d'abord accéder à la gestion des démographies puis il aura la possibilité d'ajouter, chercher, modifier ou supprimer une démographie.

S'il choisit d'ajouter une démographie, il doit cliquer sur le bouton ajouter et remplir les champs requis et cliquer sur enregistrer. Ensuite, le système va traiter et enregistrer les informations fournis par l'administrateur. Un message d'erreur est envoyé en cas de champ incomplet sinon un message de succès.

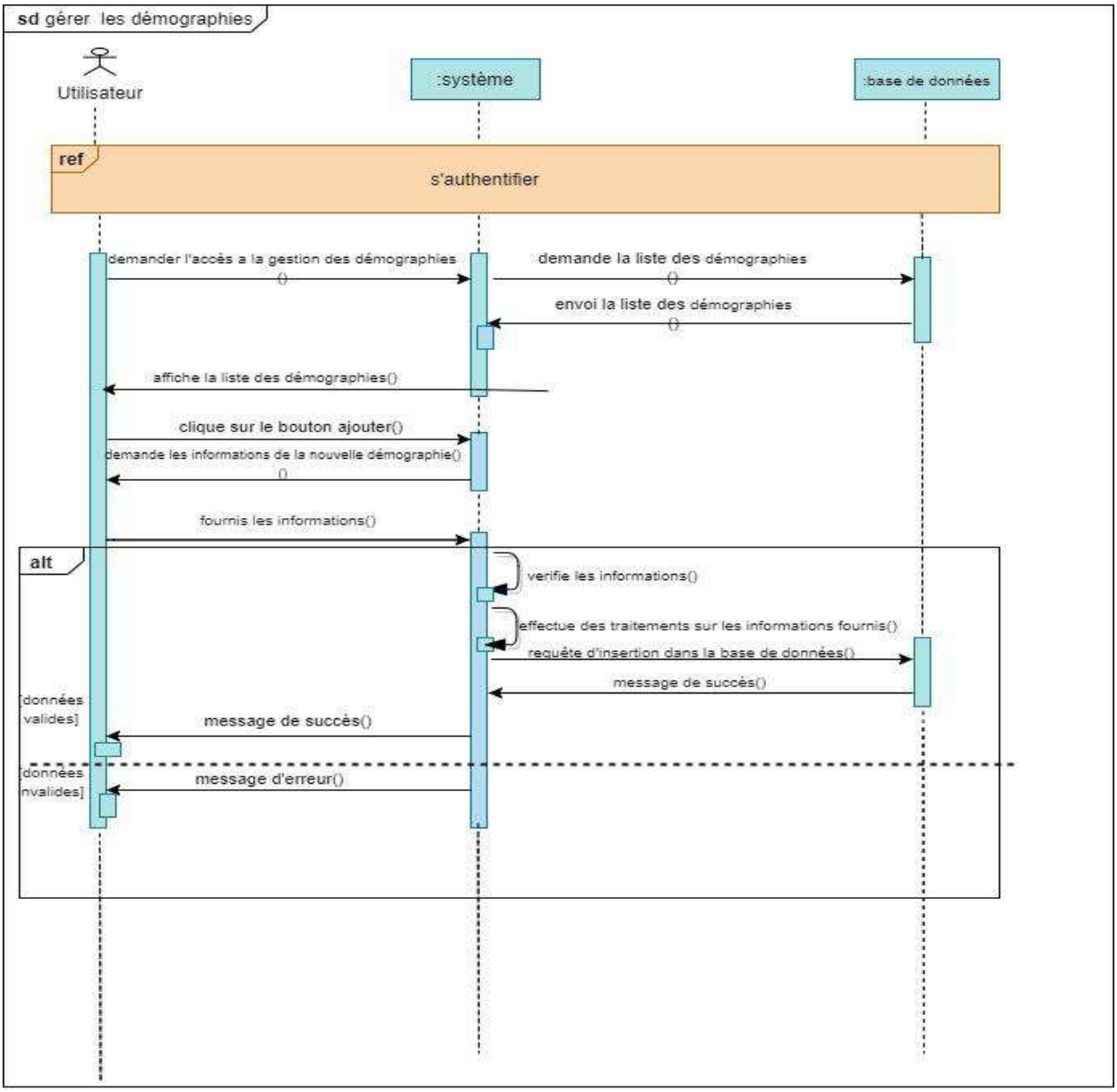


Figure 12 : Diagramme de séquence <<Gérer les démographies>>

Diagramme de séquence N°4 : cas d'utilisation Gérer les campagnes

Quand l'administrateur veut gérer les campagnes, il doit tout d'abord accéder à la gestion des campagnes puis il aura la possibilité d'ajouter, modifier, chercher ou supprimer une campagne.

S'il choisit d'ajouter une campagne, il faudrait qu'il cliquer sur le bouton ajouter et remplir les champs requis et cliquer sur enregistrer. Ensuite le système va traiter et enregistrer les informations fournis par l'administrateur puis va le rediriger vers les détails de cette campagne.

Un message d'erreur est envoyé en cas de champ incomplet sinon un message de succès.

Ensuite s'il le veut, l'administrateur pourra exporter les données de la campagne dans un fichier Excel ou PDF en appuyant sur le bouton exporter.

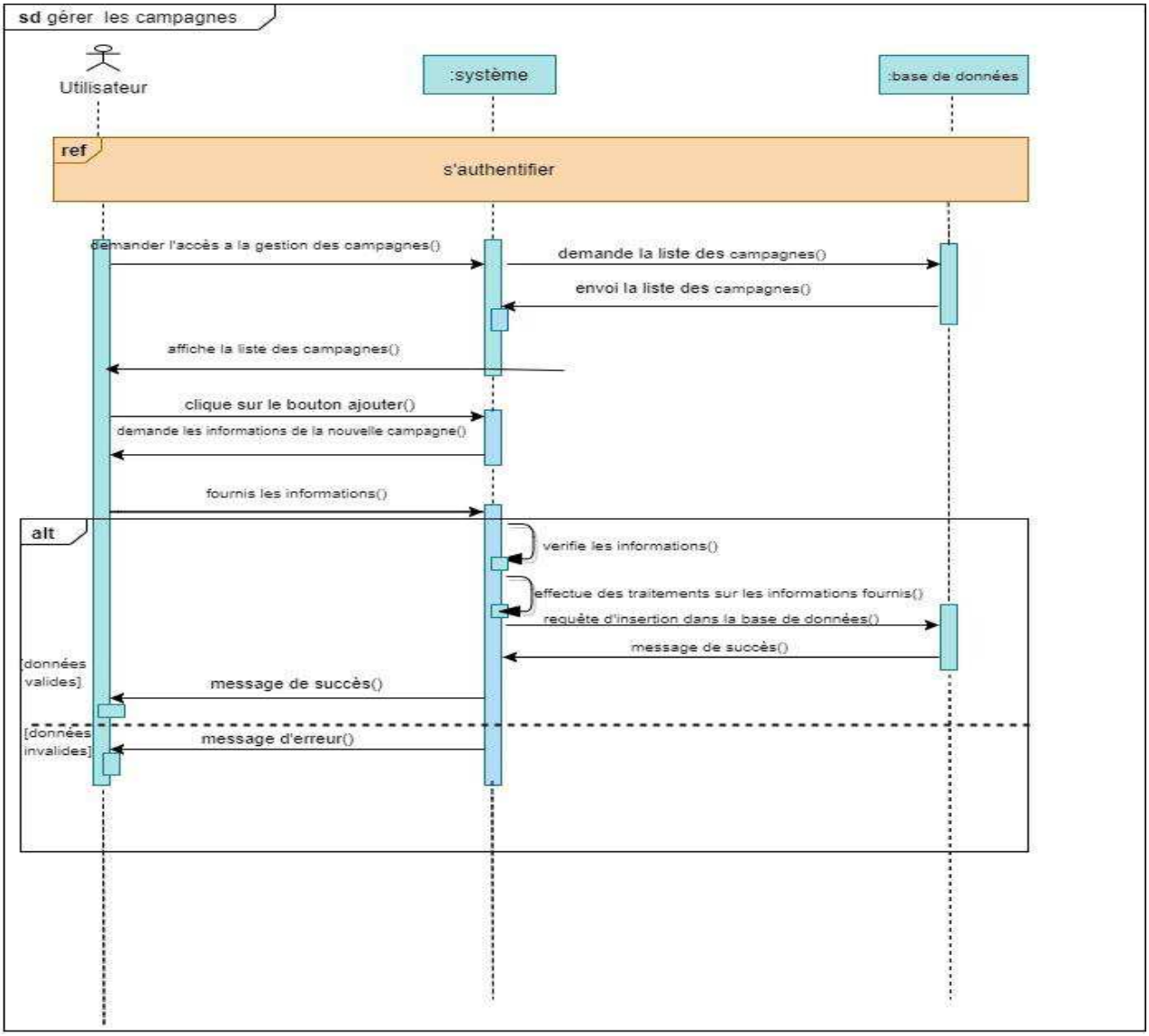


Figure 13 : Diagramme de séquence <<Gérer les campagnes>>

Diagramme de séquence N°6 : cas d'utilisation Consulter les statistiques

En cliquant sur consulter les statistiques, l'utilisateur peut visionner les statistiques d'une campagne. En effet si l'utilisateur clique sur l'icône des statistiques d'une campagne, le système va afficher les statistiques de cette campagne.

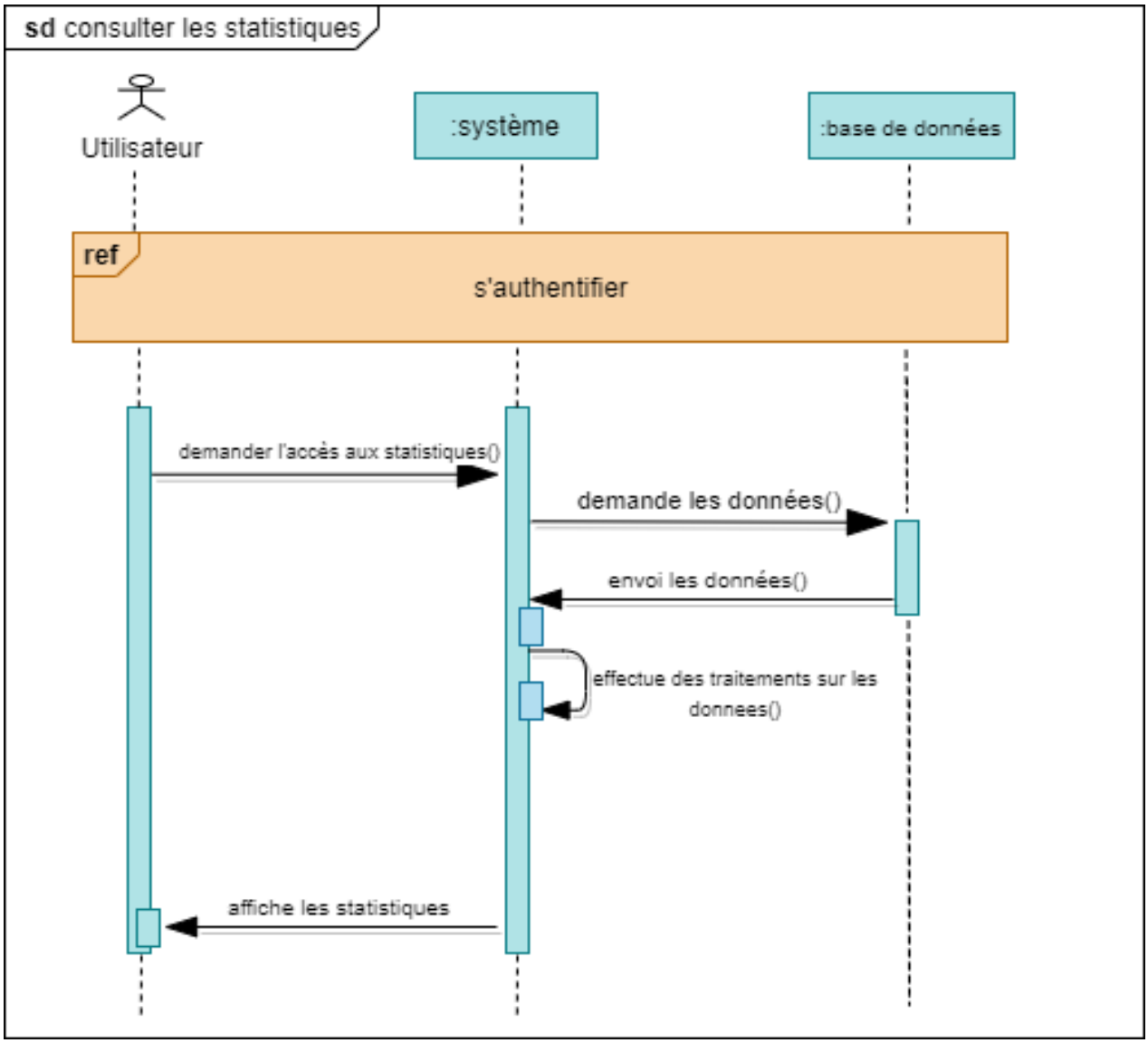


Figure 14 : Diagramme de séquence <<Consulter les statistiques>>

Diagramme de séquence N°7 : cas d'utilisation Consulter la Map

Pour afficher les positions d'un agent lors d'une campagne sur la Map, l'utilisateur doit cliquer sur consulter la Mapp, le système vas alors afficher une Mapp et proposer à l'utilisateur une liste d'agents et de campagnes. Ensuite l'utilisateur vas sélectionner un agent et une campagne puis le système va récupérer les données, les traiter et afficher les positions de l'agent durant cette campagne sur la Mapp.

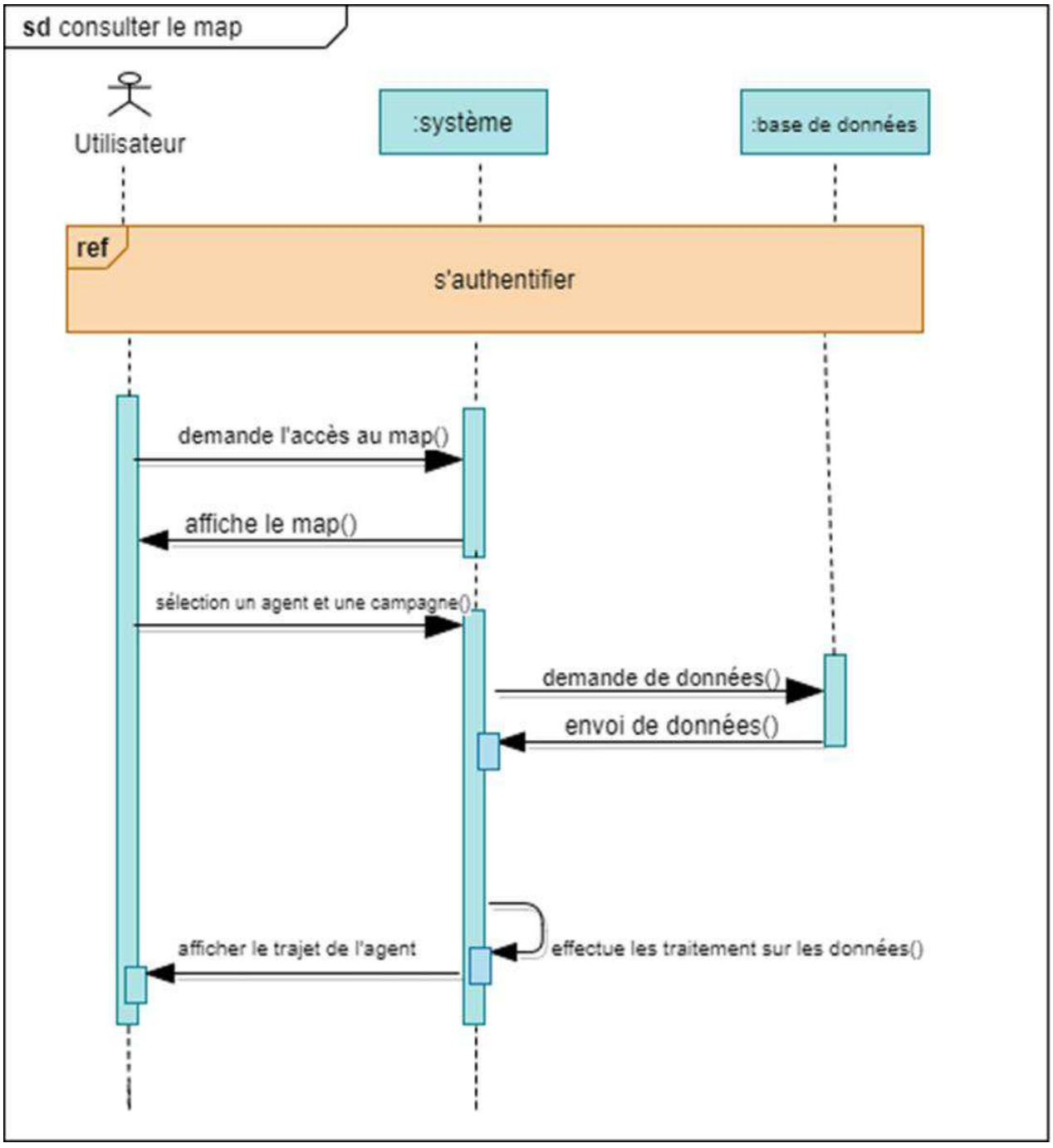


Figure 15 : Diagramme de séquence <<Consulter la carte>>

Diagramme de séquence N°8 : cas d'utilisation La collecte des données

Pour collecter et envoyer les données, l'agent doit fournir les informations en cliquant sur ajouter, le système va alors lui demander les informations, après avoir fournir tous les informations, l'agent doit cliquer sur enregistrer, le système va insérer ces informations dans la base de données locale.

La liste des vaccinations faites sera alors affichée, l'agent doit cliquer sur synchroniser et le système va insérer ces informations dans la base de données distante un message d'erreur est envoyé en cas de champ incomplet sinon un message de succès.

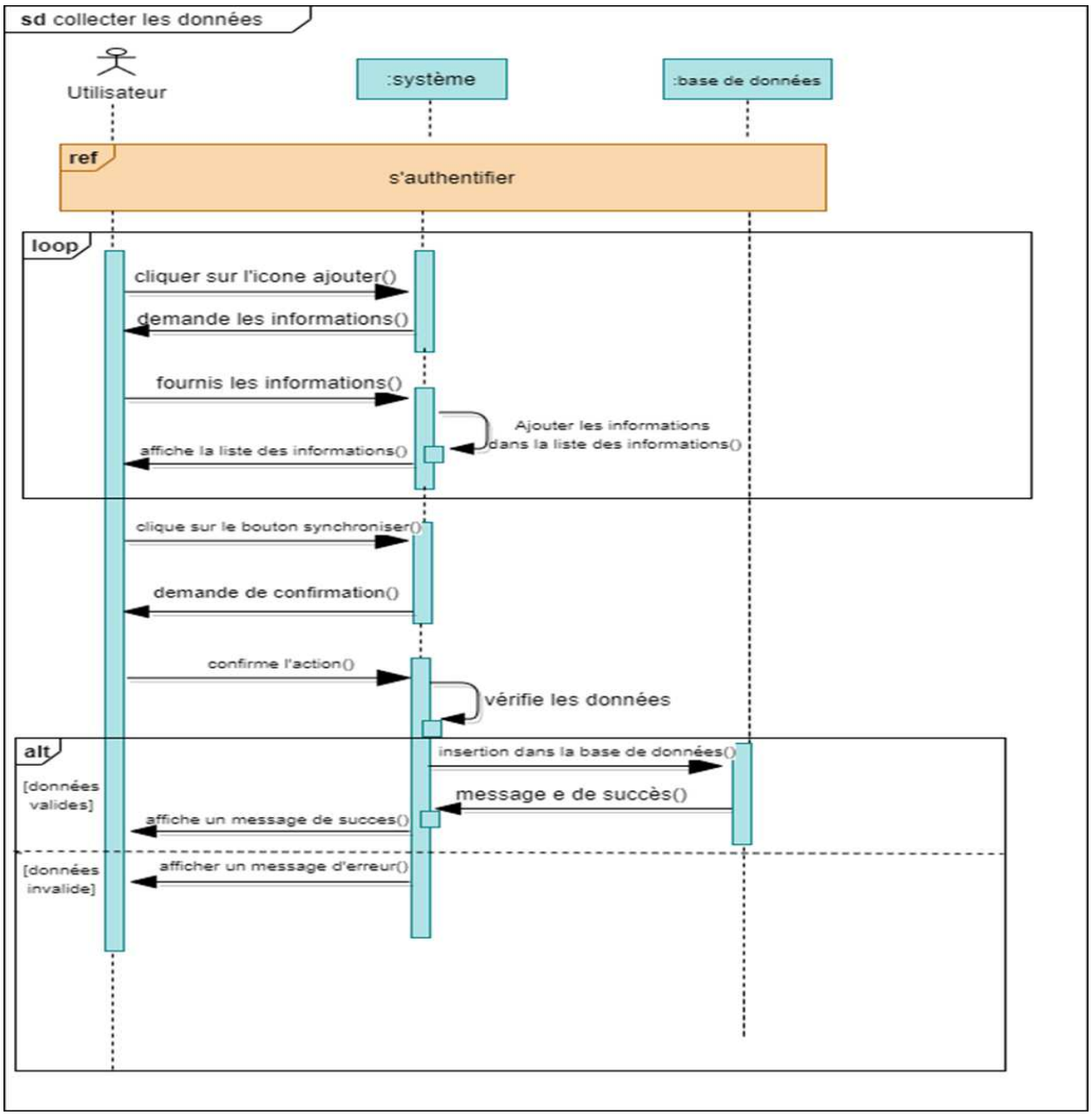


Figure 16: Diagramme de séquence <<Envoyer les données collectés>>

c. Diagramme de classe

Le diagramme de classe est sans doute le diagramme le plus important à représenter pour les méthodes d'analyse orientées objet. En effet, il permet de spécifier qui intervient à l'intérieur du système. Un diagramme de classes fait abstraction des aspects dynamiques et temporels du système, il va permettre de représenter une vue statique du système d'information. Il s'agit plutôt des relations entre les classes, des services rendus et utilisés par chacune d'elles et de l'articulation de l'ensemble.

Table	Description
AppUser	Représente l'utilisateur de l'application
AppRole	Représente le rôle attribué à l'utilisateur
Campagne	Représente la campagne à effectuer
Démographie	Représente les populations cibles
Vaccination	Représente les vaccinations à faire
Vaccin	Représente le vaccin utilisé
Moughataa	Représente la moughataa
Wilaya	Représente le wilaya

Table 9: Description textuelle du diagramme de classe

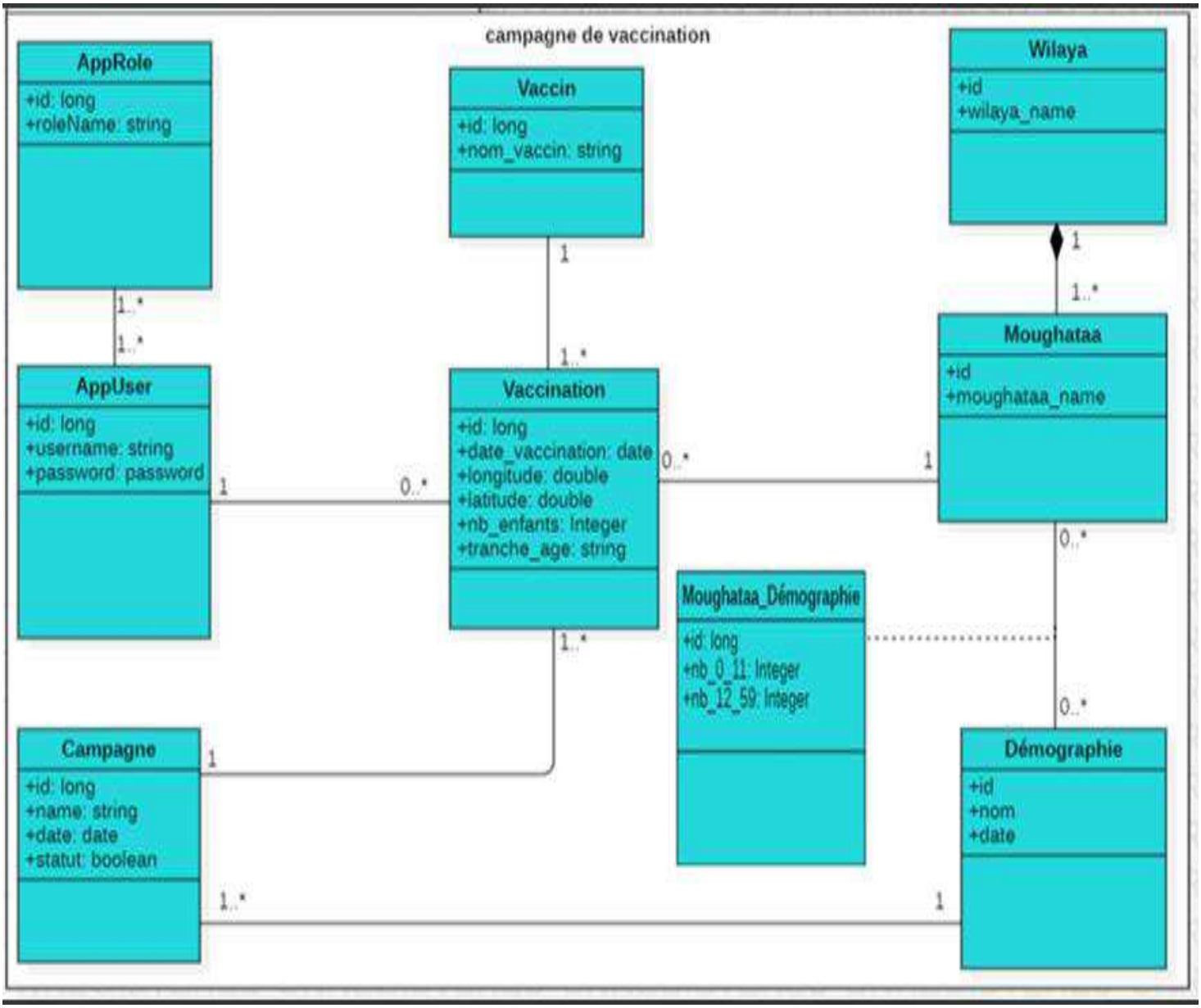


Figure 17 : Diagramme de classe.

Conclusion :

Au cours de ce chapitre j'ai présenté, les besoins attendus et les utilisateurs de l'application, ainsi que la méthode de travail Processus Unifiés et les diagrammes qui ont modélisé ce système tels que les diagrammes de cas d'utilisation, les diagrammes de séquences, le diagramme de classes ce qui va aboutir immédiatement à l'implémentation.

Dans le chapitre suivant nous allons présenter, l'environnement, la technologie et les outils de développement ainsi que l'architecture et les interfaces du système qui nous ont permis de réaliser l'implémentation.

Chapitre 3 Réalisation du système

Après avoir effectué l'étude et la conception de l'application, je passe à la phase d'implémentation. Ce chapitre présente le résultat du travail effectué durant ce projet.
Je vais présenter les outils de développement utilisés, et je clôture ce chapitre par quelques captures d'écrans démontrant les fonctionnalités de l'application.

I. Environment de travail

StarUML :



Figure 18 : Logo StartUml

StarUML est un outil de modélisation de logiciel open source qui prend en charge la plateforme UML pour la modélisation de systèmes et de logiciels, il permet de schématiser des diagrammes. [4]

IntelliJ IDEA :

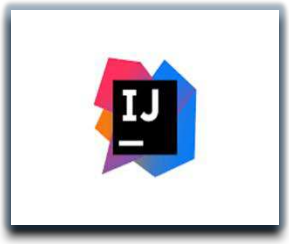


Figure 19 : Logo Intellij IDEA

IntelliJ IDEA également appelé IntelliJ, est un environnement de développement intègre initialement pour Java destinée au développement de logiciels informatiques. Il est disponible en deux versions, l'une communautaire, open source, et l'autre propriétaire, protégée par une licence commerciale. Tous deux disposent des plugins pour des langages de programmation comme JavaScript, Kotlin,TypeScript et Scala. [5]

Wamp Server :



Figure 20 : Logo WampServer

WAMP permet d'installer et de gérer Apache, PHP, MySQL, phpMyAdmin, et un gestionnaire de services. Wamp fourni avec phpMyAdmin, une interface permettant de manipuler très facilement les enregistrements de la base de données. Je l'ai utilisé pour mon SGBD MySQL. [6]

Android Studio :



Figure 21 : Logo Android studio

Android Studio est un environnement de développement pour développer des applications mobiles natives de type Android. Il utilise le moteur de production Gradle et il est basé sur IntelliJ IDEA. [7]

II. Technologie et Outils de développement

Spring boot:



Figure 22 : Logo Spring boot

Spring Boot est une déclination du framework spring avancé qui simplifie le démarrage et le développement de nouvelles applications Java EE. Les configurations sont atténuées avec Spring Boot, qui soutient des conteneurs embarqués. Cela permet à des applications web de s'exécuter indépendamment et sans déploiement sur le serveur web.

Cette plateforme offre entre autres, la possibilité de construire une application Java intégrant par défaut un écosystème des Framework Spring. De plus, il donne la possibilité à l'application une fois construite d'être déployée aisément et de fonctionner de façon autonome. [8]

Spring Data JPA:



Figure 23 : Logo Spring Data JPA

Spring Data JPA vise à améliorer de manière significative la mise en œuvre des couches d'accès aux données en réduisant la quantité de code à écrire. Les développeurs n'ont à écrire que leurs interfaces de référentiel, et les méthodes de recherche personnalisées, et l'implémentation sera fournis automatiquement par Spring. [9]

OpenLayers :



Figure 24 : Logo OpenLayer

OpenLayers Est un logiciel libre, publié sous licence BSD. Il constitue une bibliothèque de fonctions JavaScript assurant un noyau de fonctionnalités orienté vers la mise en place d'applications clientes Web cartographiques fluides, elle permet d'afficher des fonds cartographiques tuilés ainsi que des marqueurs. [10]

Base de données MySql :



Figure 25 : Logo MySQL

Pour le système de gestion de base de données (SGBD) j'ai utilisé MySQL. Il est distribué sous une double licence en fonction de l'utilisation. Il fait partie des logiciels de gestion de base de données les plus populaires, en concurrence avec oracle, Informix et Microsoft SQL Server il est également utilisé par le grand public (application web principalement) que des professionnels. [11]

JWT



Figure 26 : Logo JWT

Un JWT est un mécanisme permettant de vérifier le propriétaire de certaines données JSON. Il s'agit d'une chaîne codée, s´sécurisée pour les URL, pouvant contenir un nombre illimité de données, et signée de manière cryptographique. Lorsqu'un serveur reçoit un JWT, il peut garantir que les données qu'il contient peuvent être approuvées car elles sont signées par la source. Aucun intermédiaire ne peut modifier un JWT une fois qu'il est envoyé.

Le fonctionnent les jetons Web JSON se fait en authentification, lorsque l'utilisateur se connecte avec succès à l'aide de ses informations d'identification, un jeton Web JSON est renvoyé. Etant donné que les jetons sont des informations d'identification, il convient de prendre toutes les mesures nécessaires pour éviter les problèmes de sécurité. Chaque fois que l'utilisateur souhaite accéder à une route ou à une ressource protégée, l'agent d'utilisateur doit envoyer le JWT. [12]

Angular 9 :



Figure 27 : Logo Angular

Pour le front-end j'ai utilisé Angular. C'est un Framework JavaScript côté client.

- Il est considéré comme un langage côté client, ceux-ci permettent de gérer l'interface utilisateur de chaque page.

- Il combine l'utilisation de modèles déclaratifs, l'injection de dépendance - Il utilise le langage typeScript qui est un langage gratuit et open-source développé et maintenu par Microsoft et basé sur JavaScript, est l'un des Framework les mieux réputés et les plus utilisés par les développeurs de front-end. [13]

Retrofit :



Figure 28 : Logo Retrofit

Retrofit est une librairie permettant de réaliser des appels à des web services REST sur Android de la façon la plus simple qui soit, Elle a été développée par le groupe Square, il est open source.

Java :

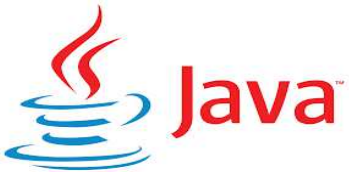


Figure 29 : Logo Java

Java est un langage de programmation orienté objet, est considérer aujourd'hui parmi les langages le plus utilisé.

Maven :



Figure 30 : Logo Maven

Maven est un outil permettant l'automatiser la gestion de projets java. Il offre entre autres les fonctionnalités suivantes.

-Compilation et déploiement des applications java (Jar, War).

- Gestion des dépendances requises par l'application.

- Exécution des tests unitaire. [14]

Bootstrap :



Figure 31 : Logo Bootstrap

Bootstrap est une collection d'outils importants pour la création du design d'applications web, de sites, c'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. [15]

Chart.js :



Figure 32 : Logo Chart.js

Chart.js est une bibliothèque JavaScript open-source gratuite pour la visualisation de données, qui prend en charge 8 types de graphiques : barre, ligne, zone, secteur (anneau), bulle, radar, polaire et scatter. [16]

Android :



Figure 33 : Logo Android

Android est un système d'exploitation mobile Open Source pour Smartphone, MP3, tablette, etc..... Conçu initialement par Android Inc, il a été racheté par Google. Il ouvre d'autre possibilité d'utilisation pour les Smartphones, tablettes.

SQLite :



Figure 34 : Logo SQLite

SQLite est un système de gestion de base de données, parmi les choix populaire en tant que logiciel de base de données intégré pour le stockage local / client dans des logiciels d'application tels que les navigateurs Web . C'est sans doute le moteur de base de données le plus largement déployé , car il est utilisé aujourd'hui par plusieurs navigateurs, systèmes d'exploitation et systèmes embarqués répandus (tels que les téléphones mobiles). [17]

III. Architecture du Système

a. Architecture 3 tiers

Pour le choix de l'architecture j'ai opté pour une architecture 3 tiers pour pouvoir séparer les couches stockage des données, logique applicative et présentation et ainsi pouvoir bénéficier d'une plus grande évolutivité du système et d'une grande flexibilité au changement.

Il s'agit d'un modèle logique d'architecture applicative qui sert à modéliser une application comme un suite de trois couches logicielles dont le rôle est bien défini :

- La présentation des données, correspondant à l'affichage, le dialogue avec l'utilisateur ;

- Le traitement métier des données, correspondant à la mise en place des règles de gestion et de la logique applicative ;

-L'accès aux données : correspondant aux données qui sont destinées à être conservées sur la durée, ou d'une manière définitive.

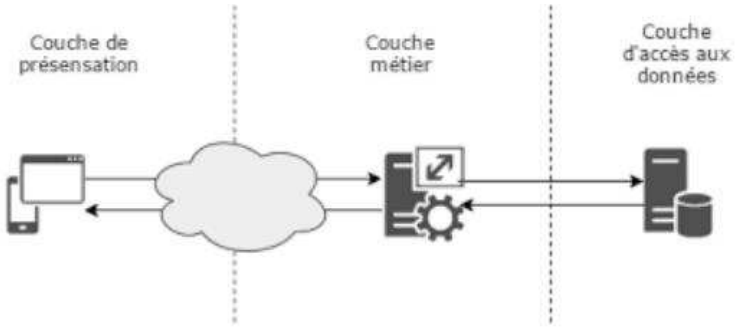


Figure 35: Architecture 3 tiers

b. Architecture du système coté client

La partie client de ce système est géré par angular dont l'architecture fait partie des architectures les plus optimisé.

Les modules de base d'une application angular sont NgModules, qui fournissent un contexte de compilation pour les composants. Les NgModules collectent le code associé dans des ensembles fonctionnels.

Les composants définissent des vues, qui sont des ensembles d'éléments d'écran parmi lesquels

Angular peut en choisir et modifier.

Les composants utilisent des services, qui fournissent des fonctionnalités spécifiques. Les fournisseurs de services peuvent être intégrés aux composants en tant que dépendances, ce qui rend notre code modulaire, réutilisable et efficace.

Les composants et les services sont simplement des classes, avec des décorateurs qui marquent leur type et fournissent des métadonnées qui indiquent à Angular comment les utiliser.

Les métadonnées d'une classe de composant l'associent à un modèle qui définit une vue. Un modèle combine du code HTML ordinaire avec des directives angulaires et un balisage de liaison qui permet à Angular de modifier le code HTML avant de le restituer pour affichage.

Les métadonnées d'une classe de service fournissent les informations qu'Angular a besoin pour les rendre disponibles aux composants a l'aide l'injection de dépendance (DI).

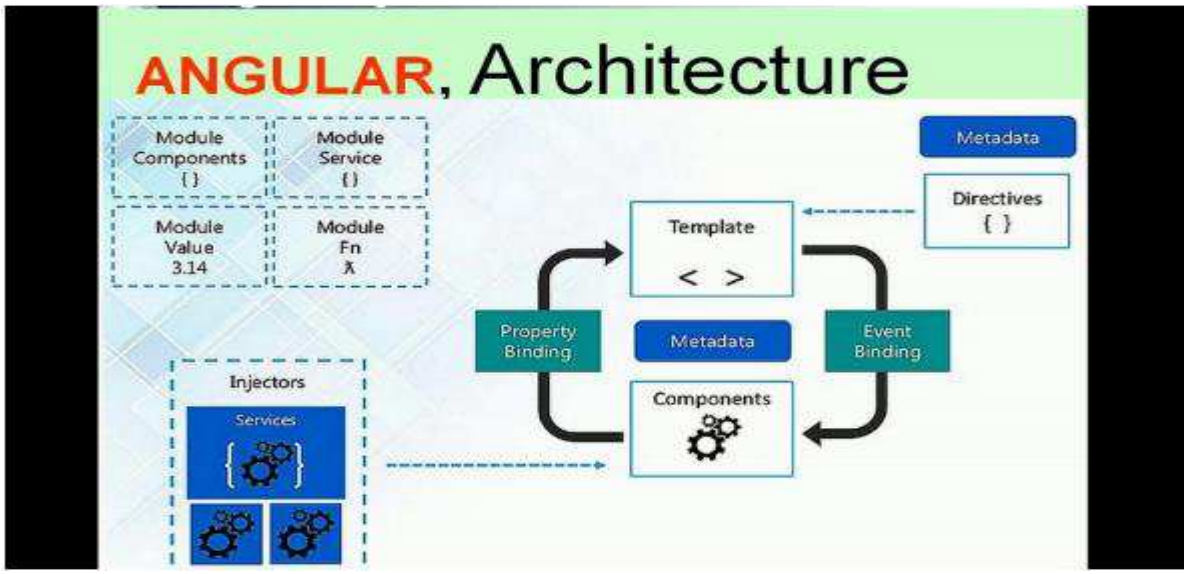


Figure 36: Architecture d'angular

c. Architecture du système coté serveur

La partie server du server est gérer par spring boot.

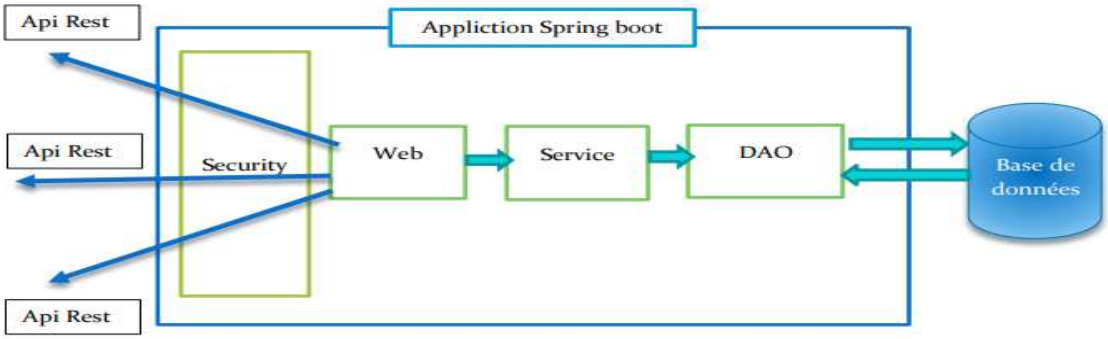


Figure 37 : Architecture coté serveur

La partie serveur de notre système est composé de quatre couche :

- La couche DAO : qui communique avec la base de données et permet l'accès aux données grâce à spring data.

- La couche Service : c'est la partie de l'application qui s'occupe des traitements métier des données, elle accède aux données en passant par la couche dao.
- La couche Web : utilise la couche service pour fournir des api rest utilisable par le côté client de notre système.
- La couche Security : permet de sécuriser l'application en filtrant les requête http entrant et en contrôlant les réponses http sortant. [12]

d. Architecture d'Android

La plate-forme Android est constituée de plusieurs couches :

- Un noyaux Linux ;
- Des bibliothèques graphiques, multimédia ;
- Une machine virtuelle JAVA adaptée Dalvik Virtual Machine ;
- Le Framework Android pour la gestion des fenêtres, de téléphone, de gestion de contenu et des fonctionnalités natives du dispositif mobile ;
- Des applications : un navigateur Web, une gestion des contacts, un calendrier, ... etc. [7]

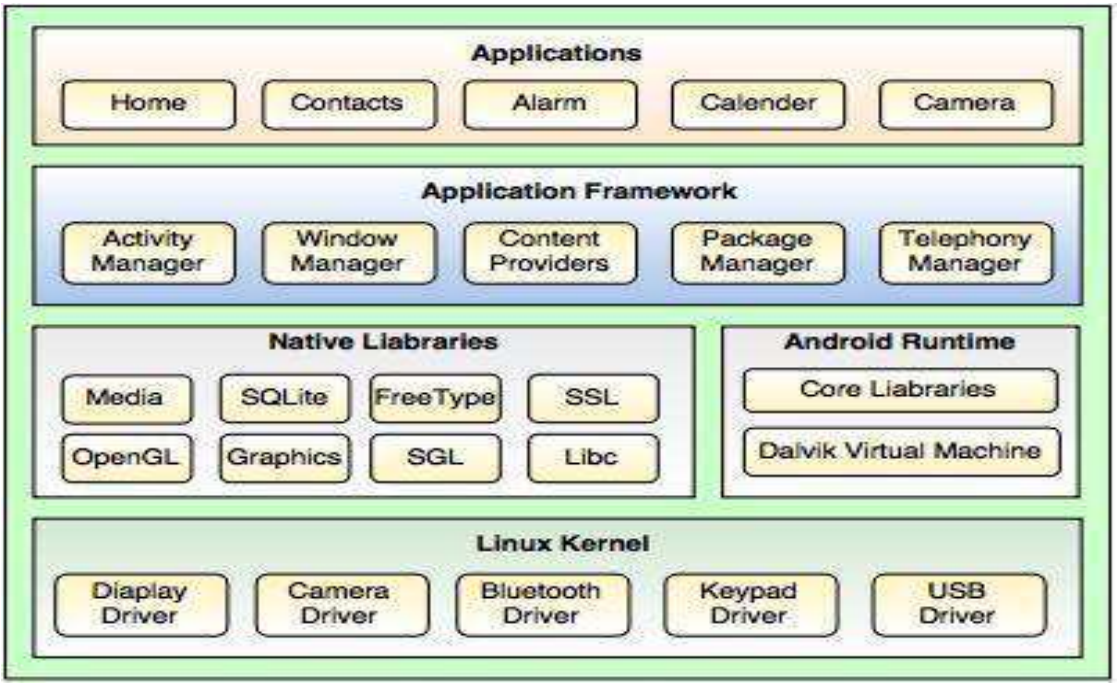


Figure 38 : Architecture d'Android

e. Architecture de cette application

L'application mobile envoi et récupère les données via des API REST en cas de disposition d'un réseau internet.

Les données seront échangées de l'application mobile via des requêtes http et transférées à l'application web qui a son tour va communiquer les données au client web à travers des requêtes http et la réponse est au format JSON.

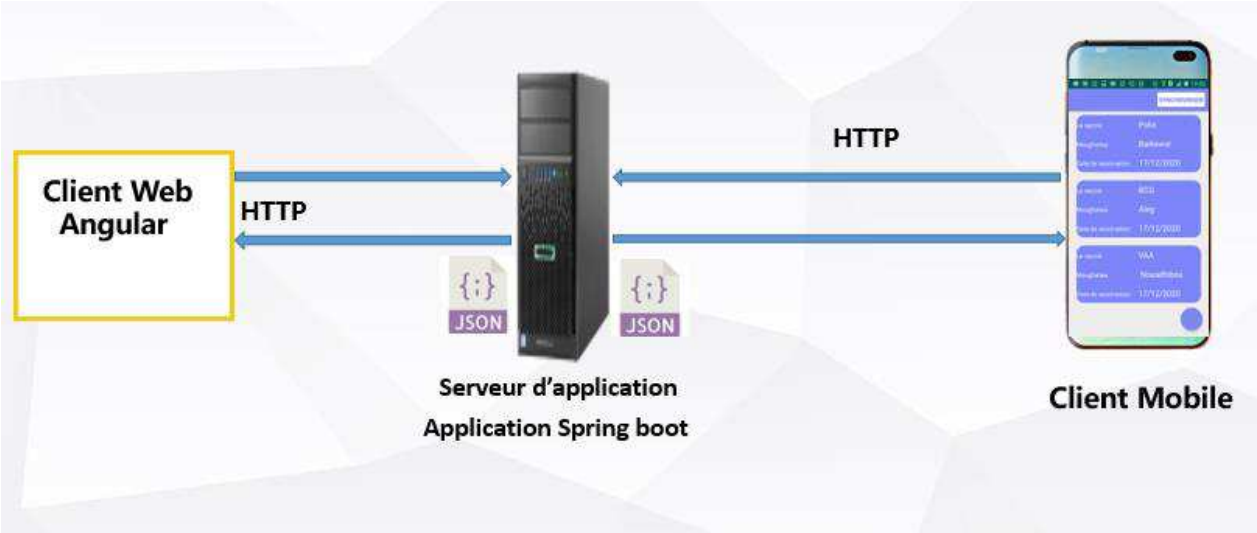


Figure 39 : l'Architecture de l'application

IV. Présentation des interfaces du système

a. Application web

- Interface d'Authentification

Les utilisateurs du système doivent impérativement se connecter via cette interface avant de pouvoir accéder au système.

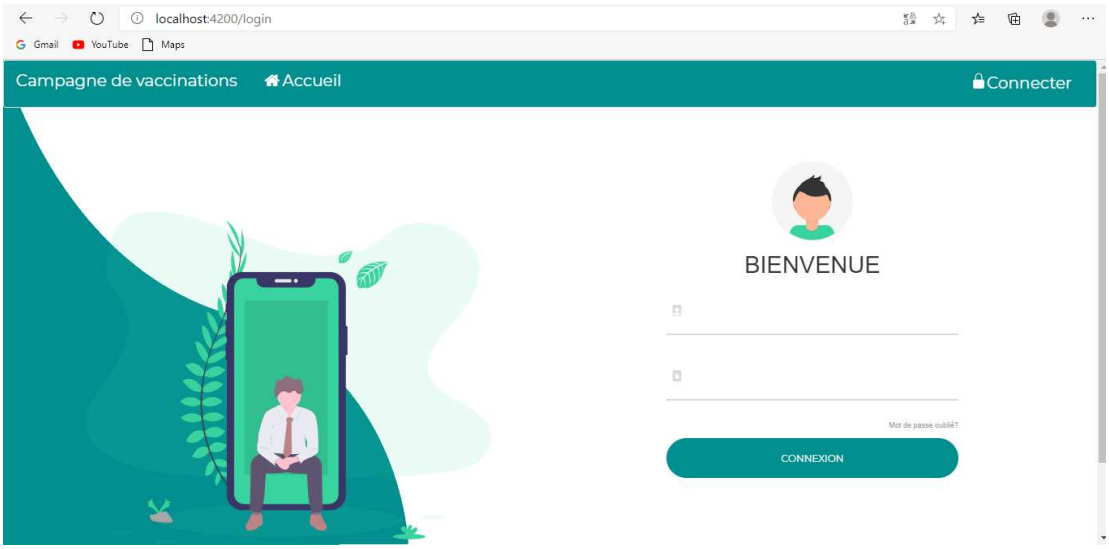


Figure 40 : Interface d'authentification

- Interface de Gestion des utilisateurs

L'interface gestion des utilisateurs permet d'ajouter, modifier, supprimer, chercher ou voir ses détails un utilisateur, et accorder de droit à un utilisateur au sein du système.

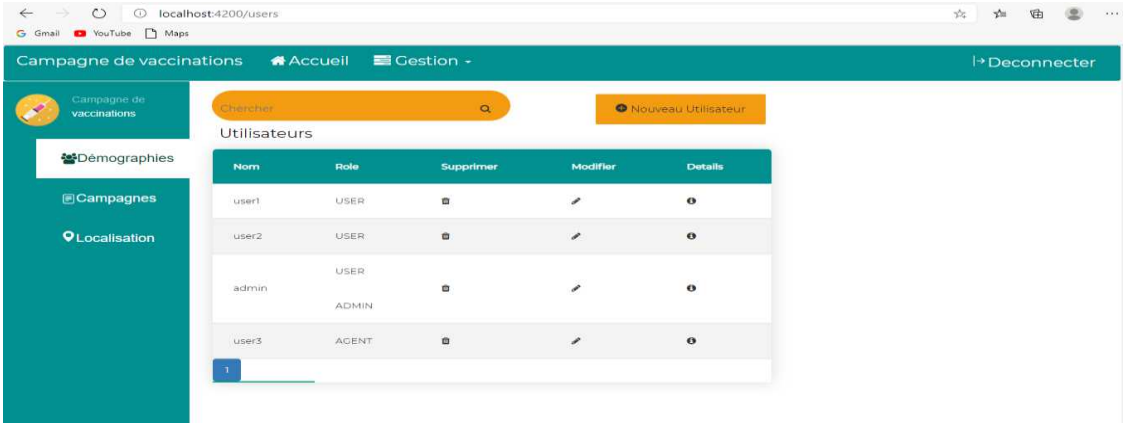


Figure 41 : Interface de Gestion des utilisateurs

- **Interface de Gestion des démographies**

Ici, l'administrateur peut effectuer les opérations de CRUD sur les démographies ainsi que consulter les détails de chaque démographie.

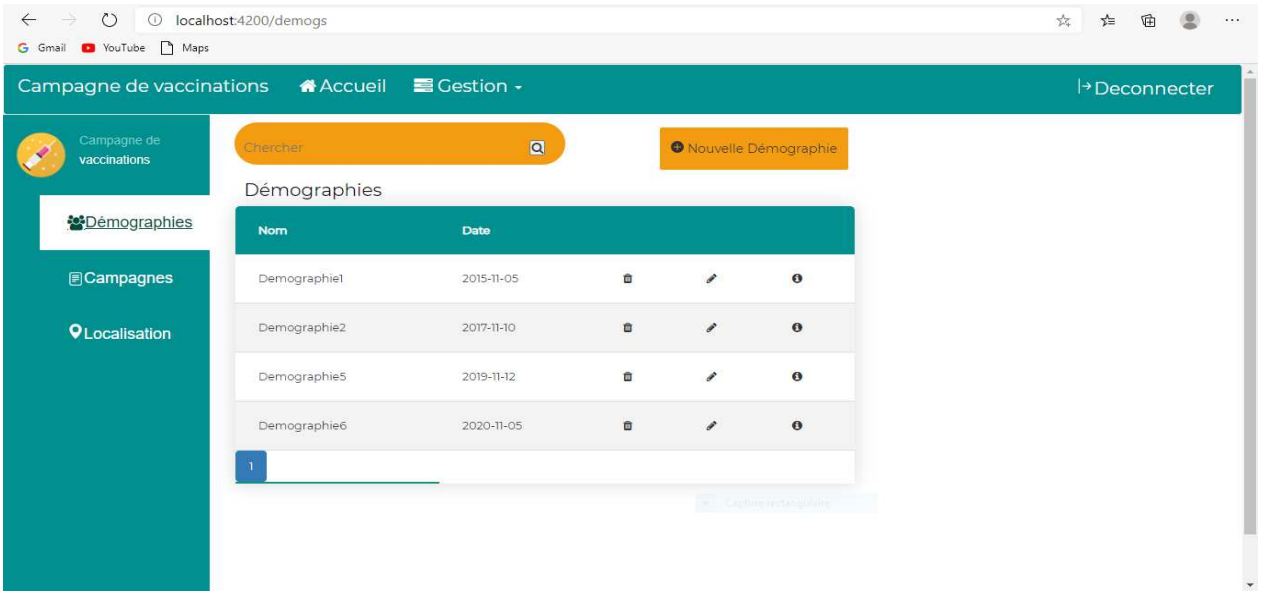


Figure 42 : Interface de Gestion des démographies

- **Interface de détail d'une démographie**

Dans cette interface, l'administrateur peut effectuer les opérations de CRUD sur les données démographiques qui sont les enquêtes faites.

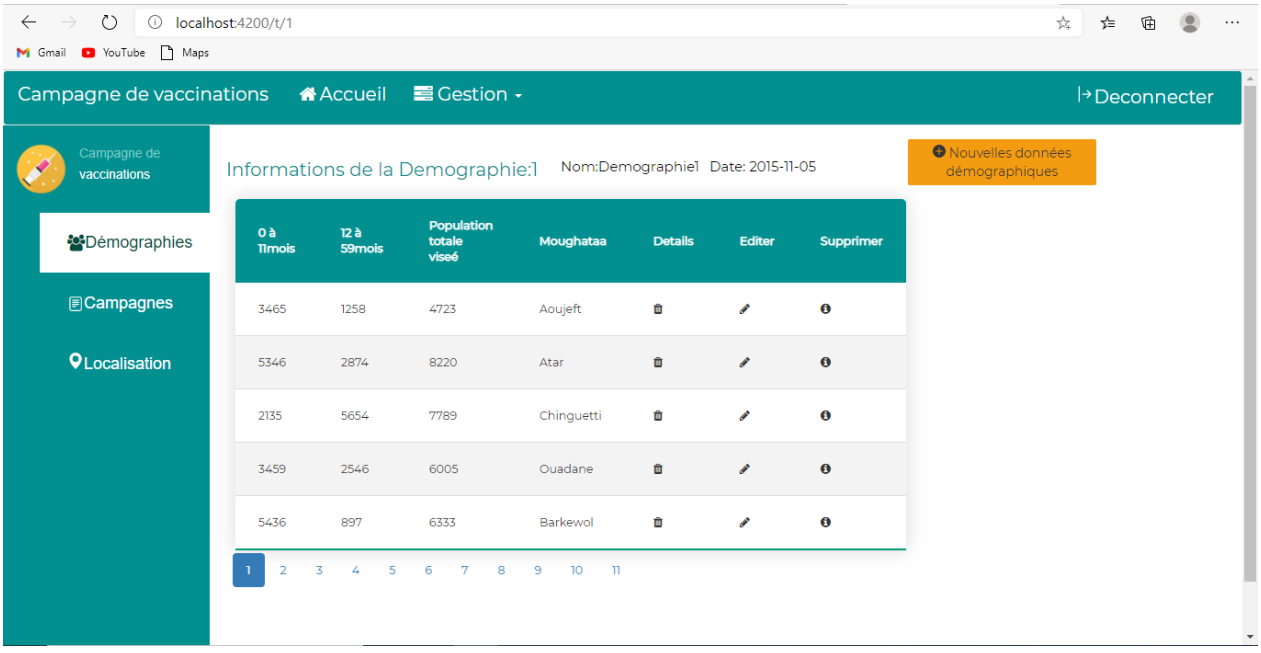


Figure 43 : Interface de détail d'une démographie

- **Interface de Gestion des campagnes**

L'interface gestion des campagnes permet d'ajouter, modifier, supprimer et clôturer une campagne et voir les statistiques et taux de couverture.

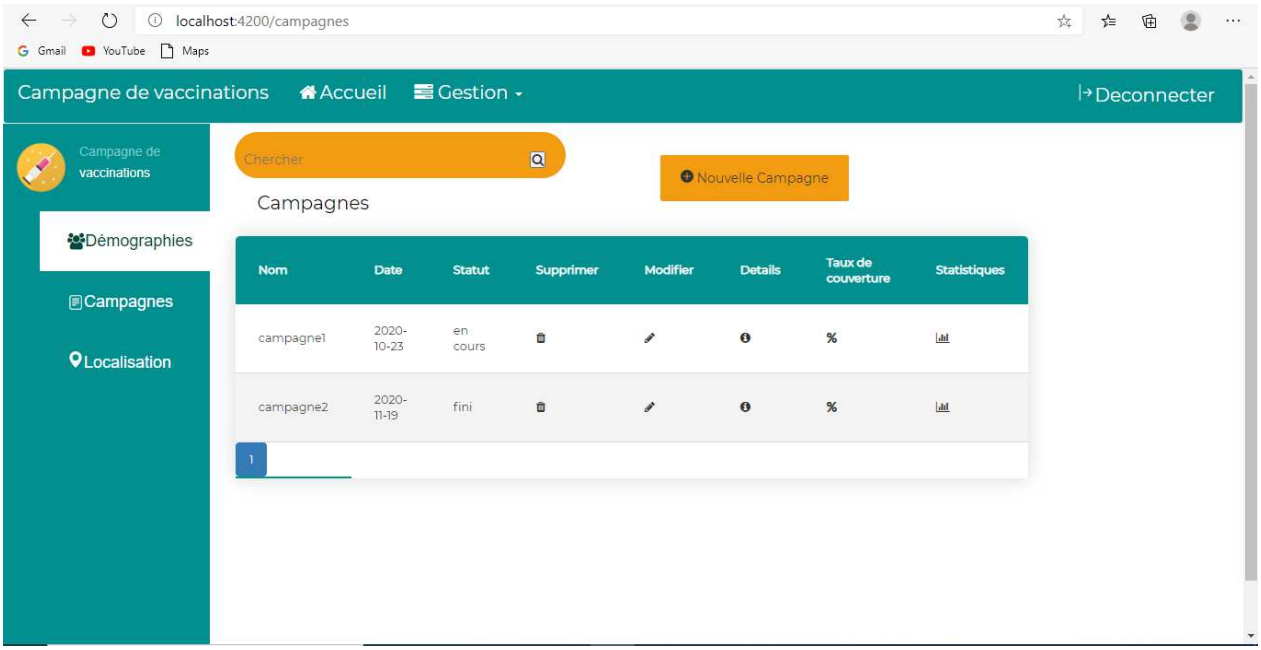


Figure 44 : Interface de Gestion des campagnes

- **Interface de détail d'une campagne**

Cette interface affiche les données agrégé d'une campagne de vaccination et de les exporter dans un fichier excel ou pdf si besoin.

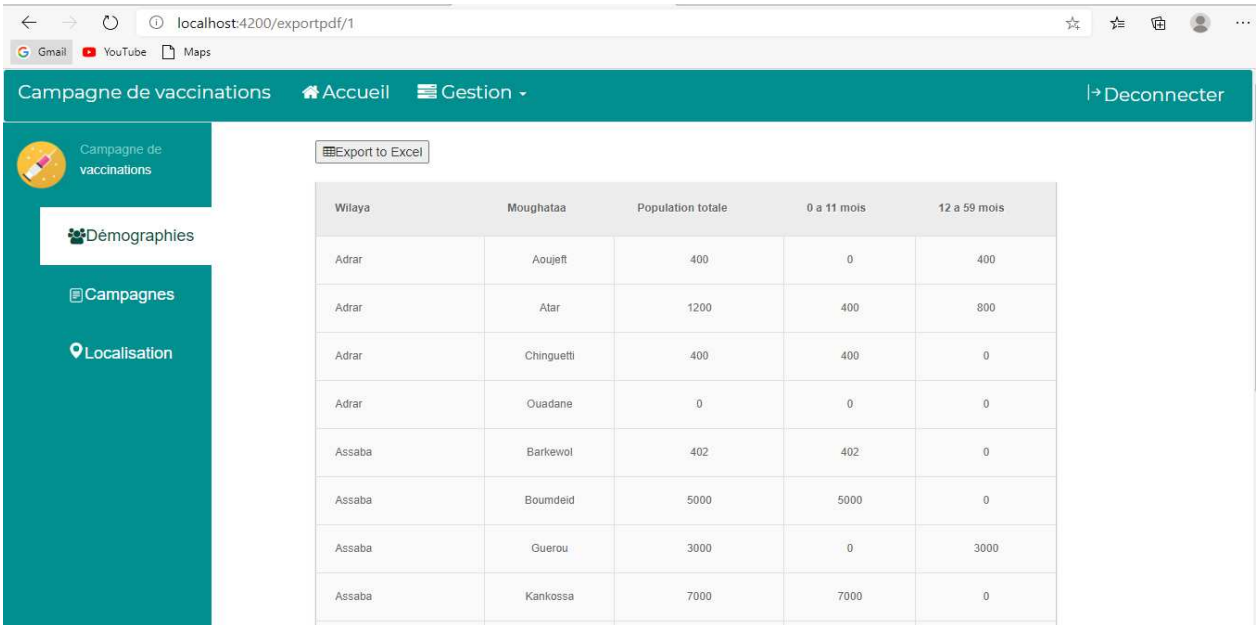


Figure 45 : Interface de détail d'une campagne

-Interface des statistiques

L'interface des statistiques permet de visualiser les données d'une campagne de vaccination sous forme de graphique.

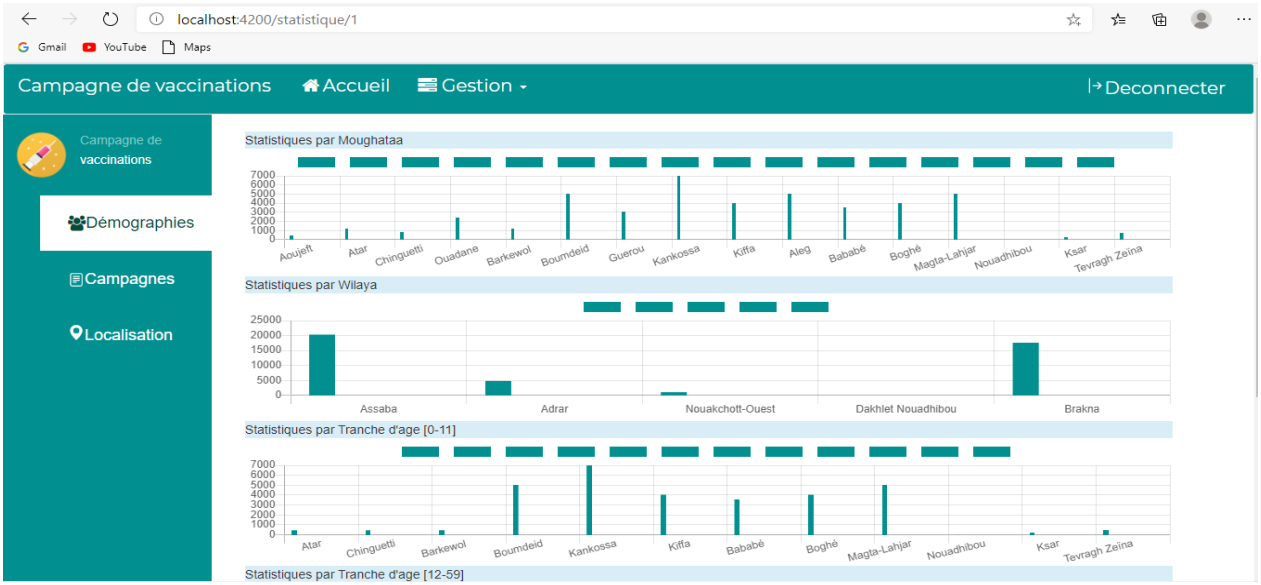


Figure 46 : Interface des statistiques

-Interface des traçabilités

L'interface des traçabilités permet de visualiser les positions des vaccinateurs sur une carte géographique.

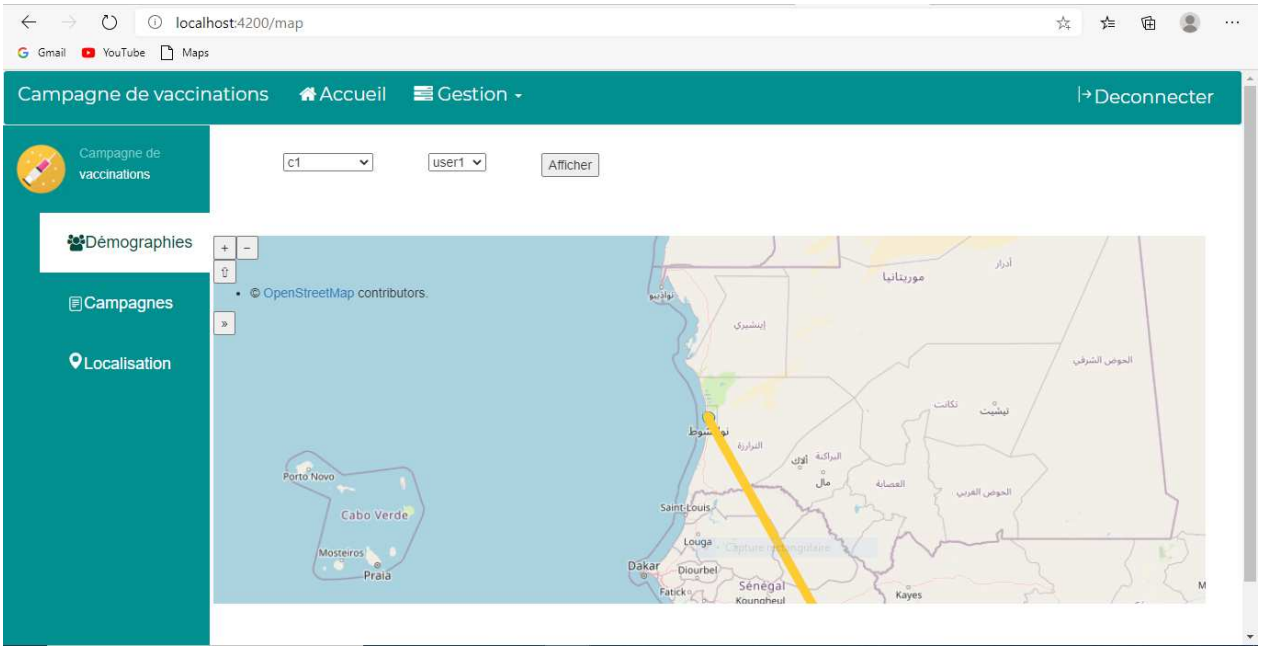


Figure 47: Interface de traçabilité

b. Application mobile

- **Interface d'Authentification**

Les agents vaccinateurs doivent s'authentifier via cette interface avant de pouvoir accéder à l'application mobile.

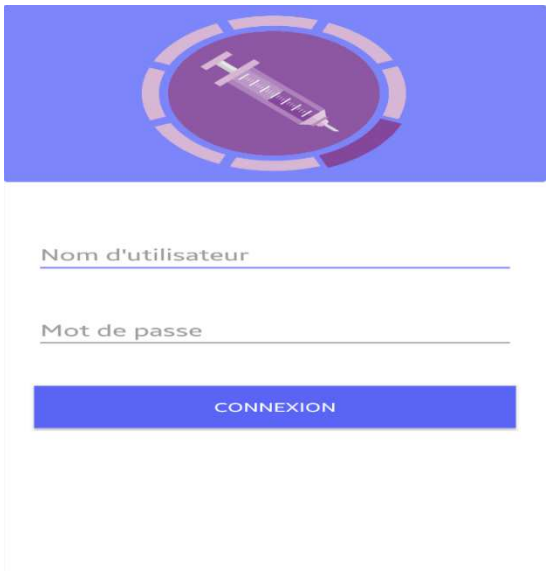


Figure 48 : Interface d'Authentification

- **Interface de vaccination**

Permet de collecter les informations de chaque enfant vacciné.

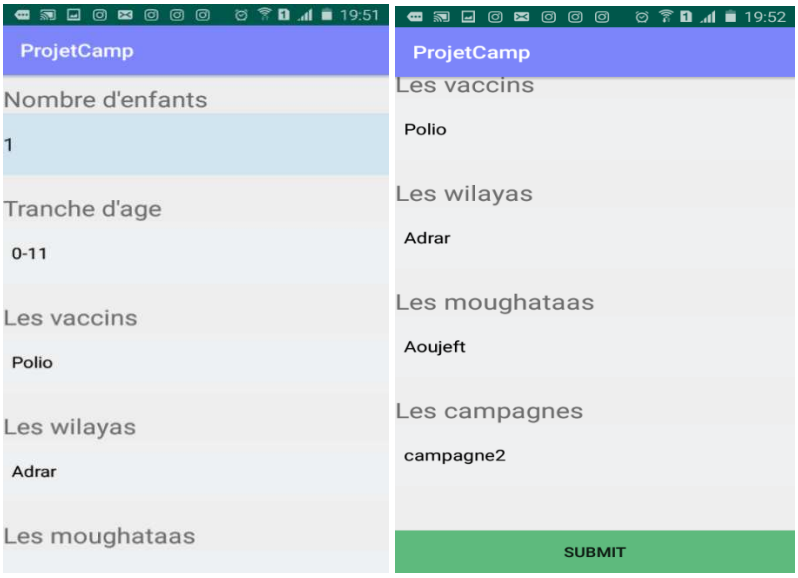


Figure 49 et 50: Interfaces de collecte de données

- **Interface des vaccinations**

Permet de lister les vaccinations enregistrées en local.



Figure 51 : Interface des vaccinations

C. La création de l'APK et le déploiement

J'ai réalisé un système opérationnel chez le ministère, j'ai déployé la partie web dans un serveur Tomcat pour débuter la phase de test au niveau du ministère et pour la partie mobile j'ai généré l'APK pour faciliter l'installation de l'application sur les téléphones des agents.

C.1 Génération de l'APK

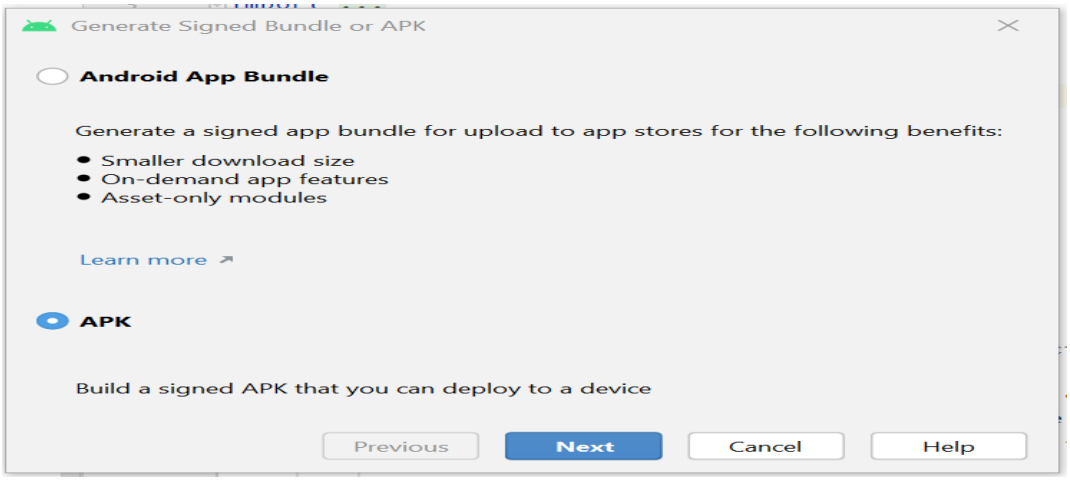


Figure 52 : Première étape de la création de l'APK

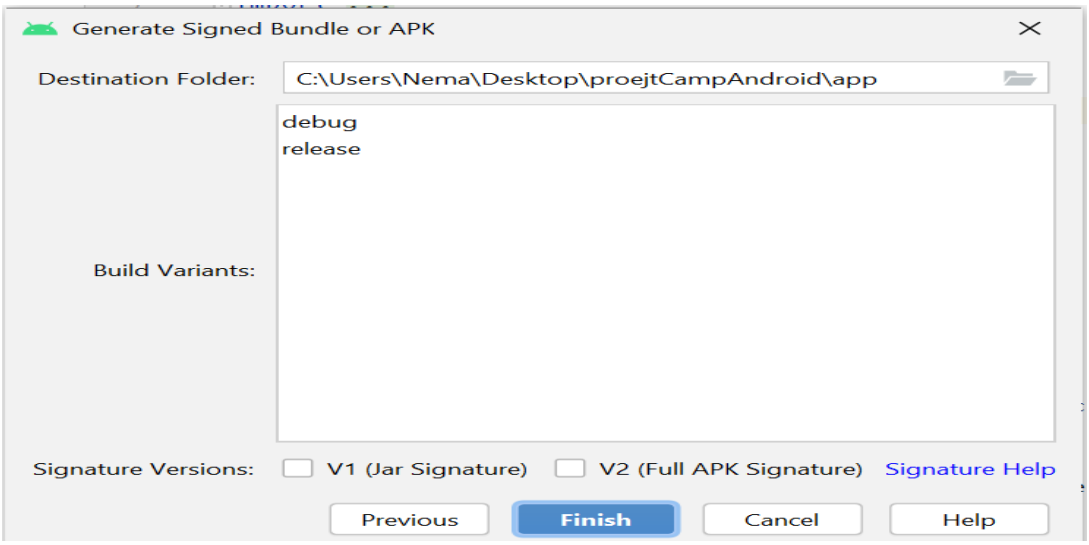


Figure 53 : Dernière étape de la création de l'APK

C.2 Déploiement

Pour le déploiement de la partie web j'ai débuter par exporter ma base de données et l'importer dans la machine du serveur ensuite la génération de la fichier war du back-end en lançant la commande MVN package dans l'invité de commande et dans l'emplacement du projet en spécifient son port et celui du front-end en lançant les commandes ng build et ng build -- prod qui doivent être ajoute dans webapps dans Tomcat.

Le port du back-end j'ai l'ajouté dans fichier front-end —» index —»href

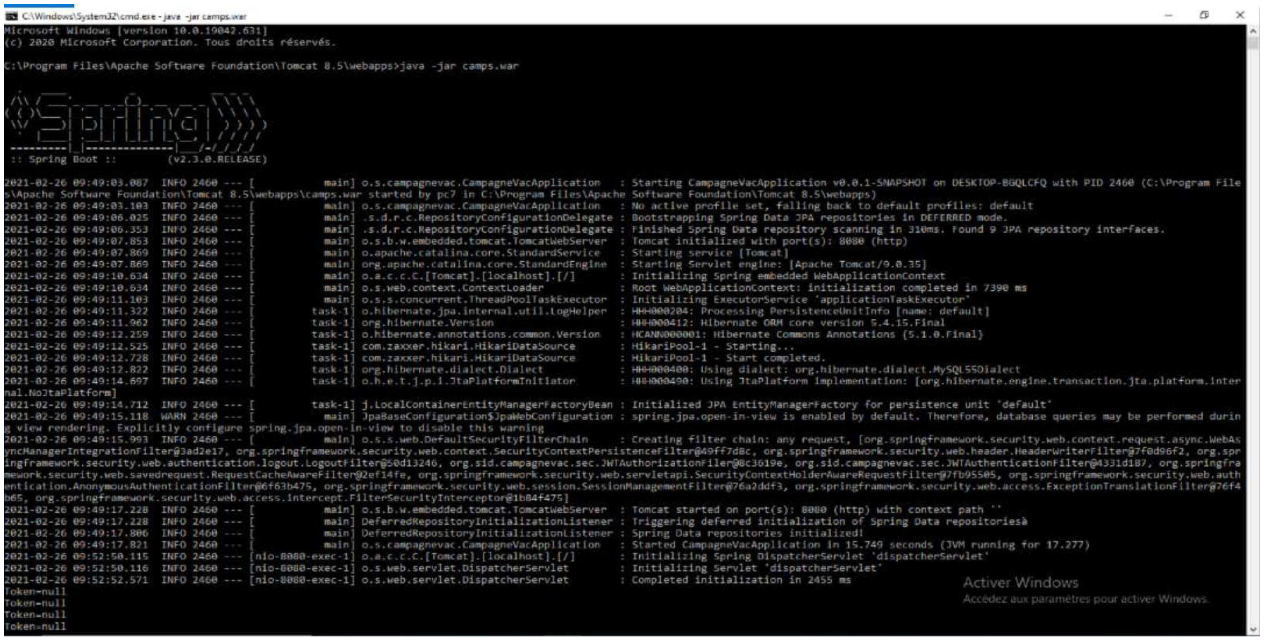


Figure 54 : Le déploiement sur un serveur Tomcat

Conclusion :

Dans ce chapitre j'ai traité les détails de la réalisation de cette application qui est la phase qui m'a pris le plus de temps.
J'ai montré les principales interfaces graphiques de l'application.

Chapitre 4 Evaluation du niveau de sécurité de l'application

Dans ce chapitre je vais concentrer sur l'aspect sécurité de ce système, car sécuriser toutes genre d'application est devenu un aspect primordial, encore le secteur sanitaire est l'un parmi les secteurs les plus sensible aux attaques et les intrusions sur le web.

Pour protéger ce système j'ai renforcé sa sécurité pour qu'il soit capable de lutter contre les pirates, pour évaluer le niveau de sécurité de ce système je vais se baser sur NMAP pour réaliser des scans (des services et des vulnérabilité).

I. Les attaques

a. Attaque passive

Une attaque passive consiste à écouter sans modifier les données ou le fonctionnement du réseau. Elle est généralement indétectable mais une prévention est possible, cette attaque est plus souvent faite par un logiciel de scanner comme : Nmap, Wireshark, Kismet, TcpDump, iStumbler...

b. Attaque active

Une attaque active modifie désactive ou détruire des ressources ou des données, le contenu des informations du système ou le comportement du système. Elles sont en général plus critiques que les passives. On a plusieurs types d'attaques actives comme : Spoofing, Dos, Man In The Middel…[19]

II. Présentation des outils utilisés

- **NMAP**



Figure 55 : Logo NMAP

Nmap est un outil de scan open source pour l'exploration de réseau et l'audit de sécurité aussi connu sous le nom d'analyseur de port, il est principalement utilisé pour voir l'état des ports (ouvert ou fermé). [20]

- **Kali linux**



Figure 56: Logo Kali Lunix

Kali Linux est une distribution de Linux qui est dédiée aux tests de pénétrations et les tests numériques, Kali contient plus de 600 programmes pré-installer dans son système comme : Nmap, Airecrack-ng, Jhon the Ripper.... [21]

- **VMware Workstation**



Figure 57 : Logo VMware

VMware Workstation est un hyperviseur hébergé qui permet d'exécuter plusieurs systèmes d'exploitation en tant que machines virtuelles sur une machine Linux ou Windows. Les professionnels de l'informatique, les développeurs et les entreprises qui construisent, testent ou démontent des logiciels pour n'importe quel appareil, plate-forme ou cloud s'appuient sur Workstation. [22]

- **Wireshark**



Figure 58: Logo Wireshark

Wireshark est un analyseur de paquets et de protocoles réseaux open source qui fonctionne sur la plus part des systèmes d'exploitation (Windows, linux, macOS….).

- **CentOS**



Figure 59 : Logo CentOS

CentOS à partir de Community Enterprise Operating System est une distribution Linux qui fournit une connexion, la plateforme de calcul soutenue par la communauté fonctionnellement compatible avec son amont source Red Hat Enterprise Linux (RHEL). En janvier 2014, CentOS a annoncé la jonction officielle avec Red Hat tout en restant indépendant de RHEL, sous un nouveau conseil d'administration CentOS. [23]

III. Tests d'évaluation

a. Présentation de l'environnement de travail

- Raspberry pi (raspbian)
- Machine virtuelle (Kali linux)
- Machine virtuelle (CentOs)
- VMware (Workstation Pro)
- Wireshark

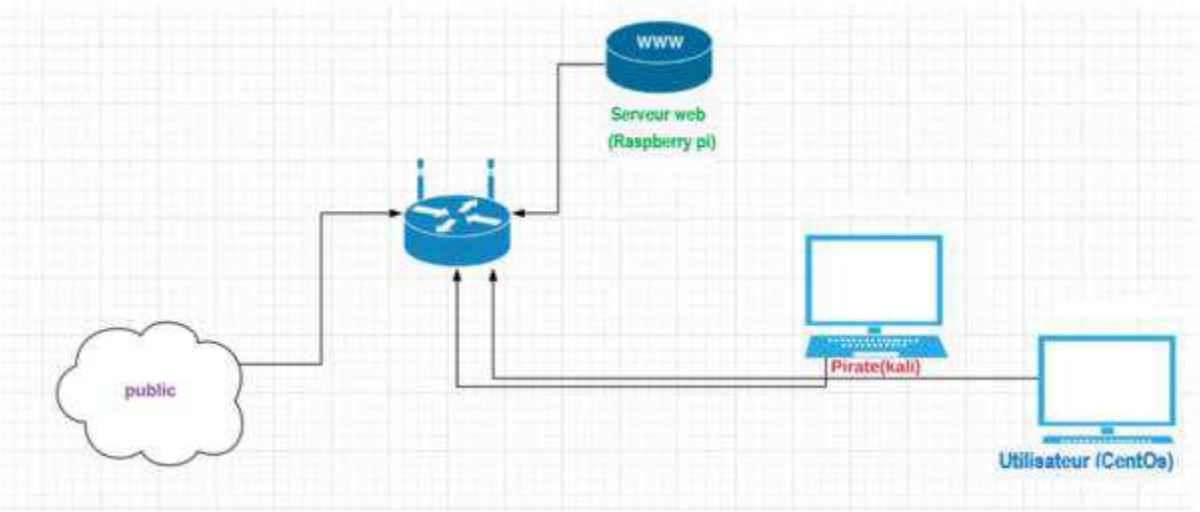


Figure 60 : La topologie implémentée

j'ai voulais sécurisé l'application contre les attaques qui sont dangereux et peuvent effecter negativement sur la confidentialité, et pour cela j'ai choisi man in the middle comme un exemple.

Man In The Middle

Dans le domaine de la cryptographie et de la sécurité informatique , un man in the middle est une cyberattaque où l'attaquant relaie secrètement et peut-être altère les communications entre deux parties qui croient qu'elles communiquent directement entre elles. Au niveau de cet écoute active l'attaquant établit des connexions indépendantes avec les victimes et relaie les messages entre elles pour leur faire croire qu'ils se parlent directement via une connexion privée, alors qu'en fait toute la conversation est contrôlée par l'attaquant.

Avant de commencer les Tests il faut savoir que le travail va être faire localement suit à une exigence de la Ministère ça veut dire que le serveur web ici est représenté tout simplement par un raspberry pi, donc pour commencer j'ai assuré que tous les équipements sont connectés à un même réseau. [24]

- Une requête envoyée par l'administrateur pour lister tous les Vaccins qui nous retourne des résultats justes pour vérifier que le serveur est bien accessible par le client.

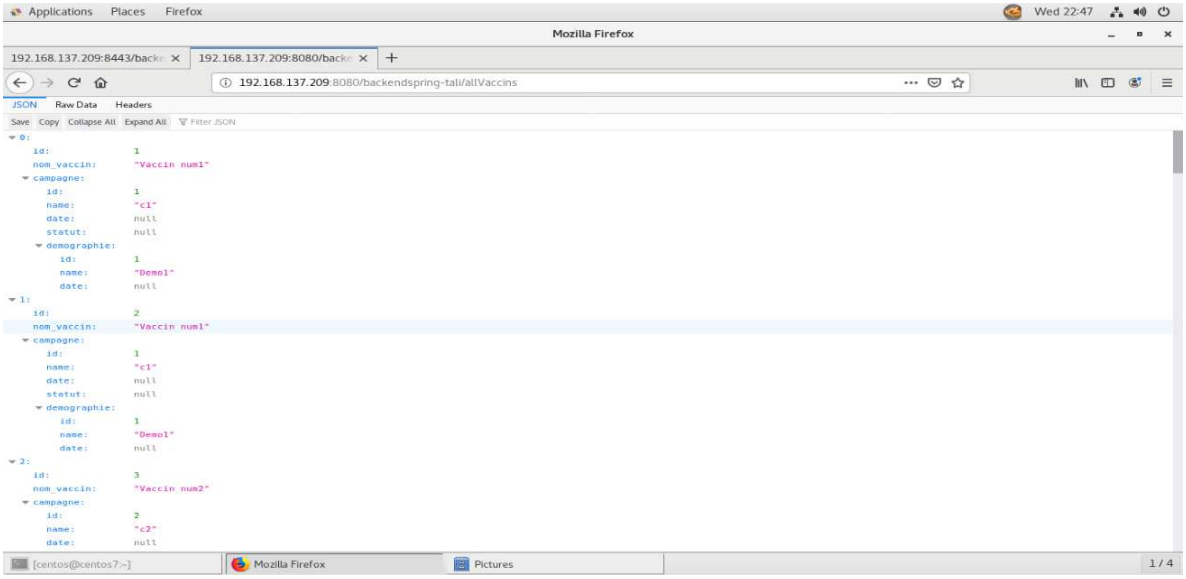


Figure 61 : Une requête affiche les vaccins

- Au niveau du machine Kali j'ai vérifié les adresses IP utilisés ainsi que la table de routage

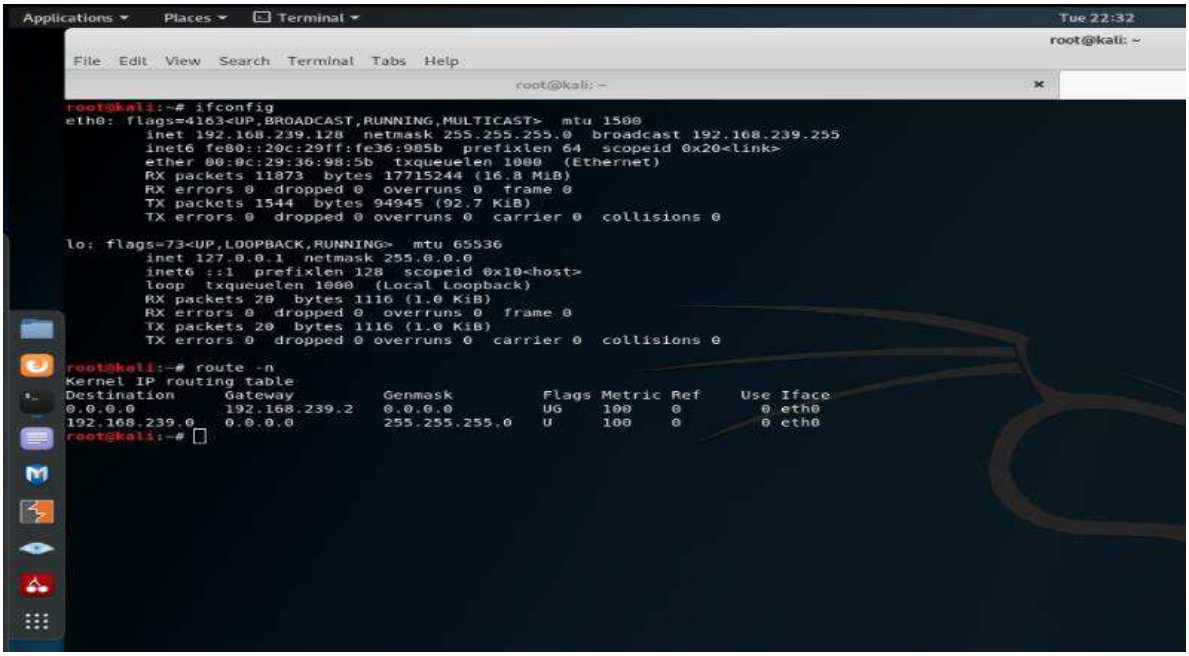


Figure 62 : Les adresses utilisés

❖ J’ai activé IP forward router sur Kali pour l’utiliser comme un routeur

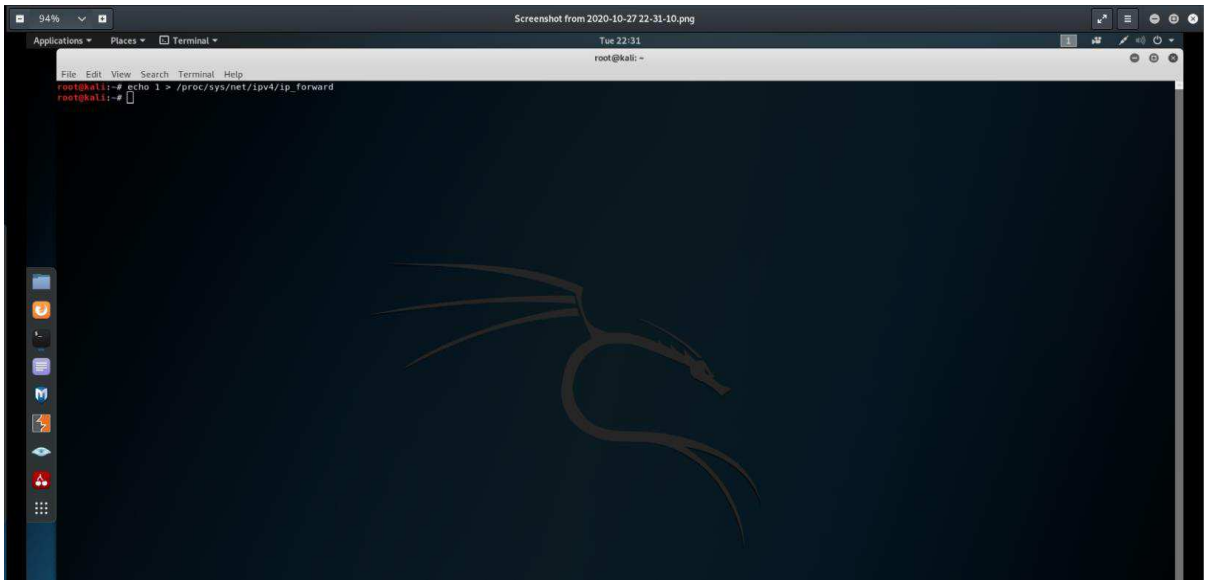


Figure 63 : L’activation de IP forward router

❖ Scan NMAP pour identifie l’IP la victime

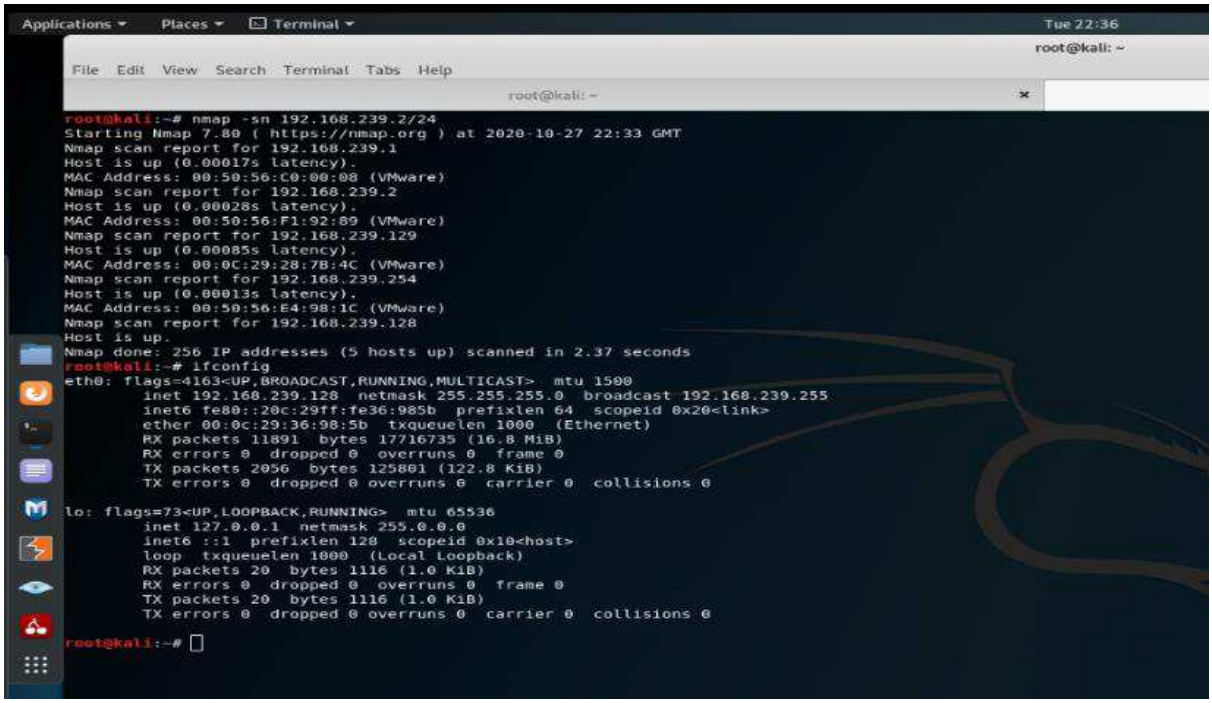


Figure 64 : Scan NMAP

- La technique arpspoof m’a permis d'associer l'adresse MAC de l'attaquant avec l' adresse IP du serveur web et de le considérer , comme la passerelle par défaut , ce qui provoque que tout le trafic destiné aux serveur vont être envoyer à l'attaquant à la place.

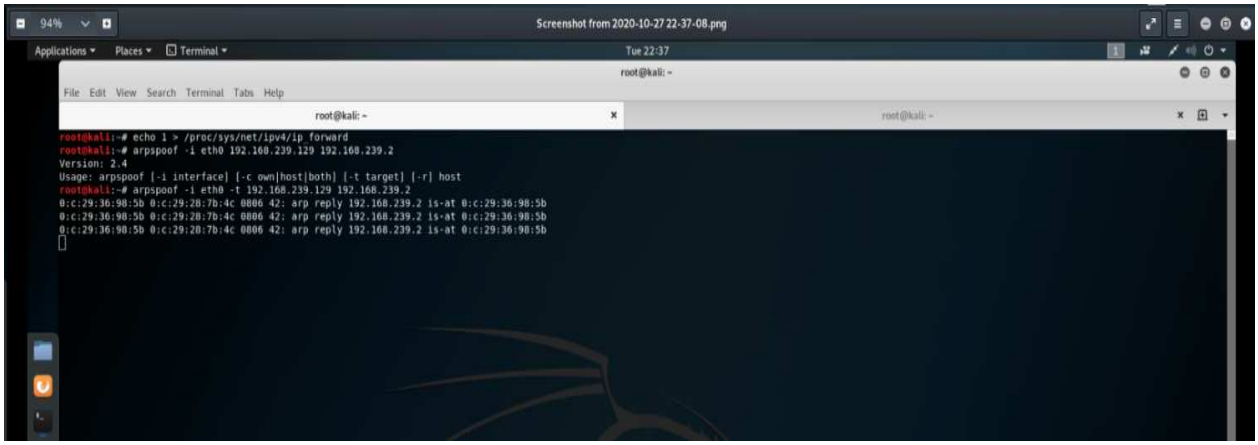


Figure 65 : Arpsoof

- Voici une requête envoyée par l’administrateur pour lister tous les Wilayas en utilisant HTTP sans sécurités sure le port 8080 d’où les données sont envoyées en clair dont ils peuvent être interceptées.

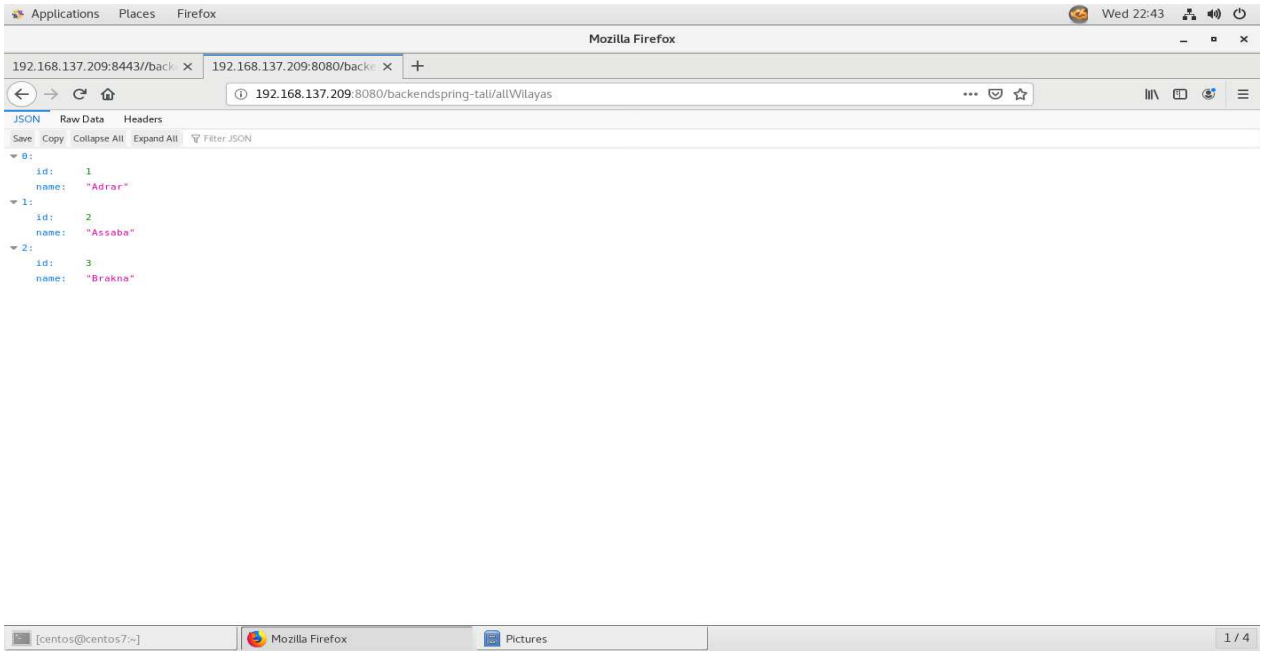


Figure 66 : Une requête affiche les wilayas

- Voici une requête envoyée par l'administrateur pour lister tous les Wilayas en utilisant HTTPS avec sécurité sur le port 8443 d'où les données ne peuvent pas être interceptées.

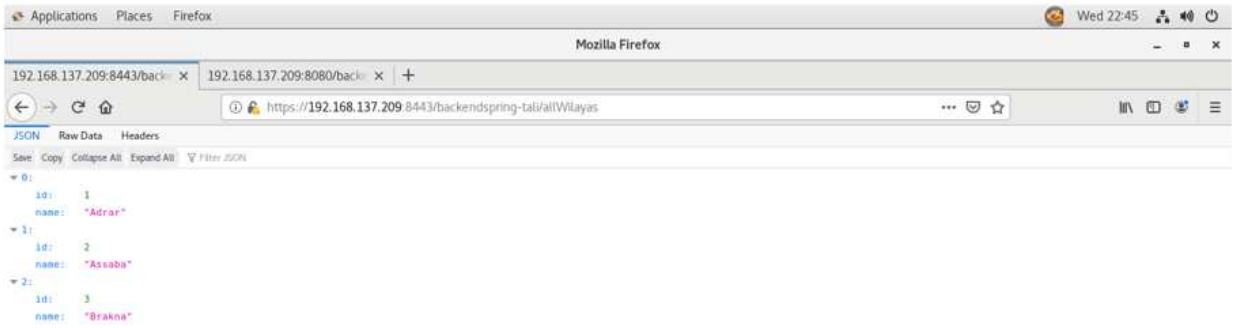


Figure 67 : Requête sur le port 8443

- Il est clair que l'attaquant a pu intercepter la requête HTTP sans sécurisée sur le port 8080 mais il n'a pas pu intercepte celle HTTPS sécurisée sur le port 8443.

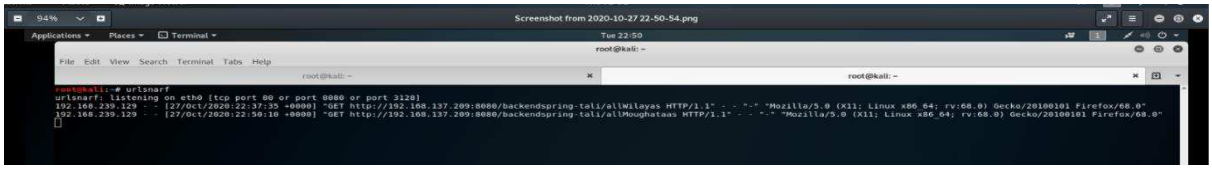


Figure 68 : L'interception des données

- Voici la capture de Wireshark

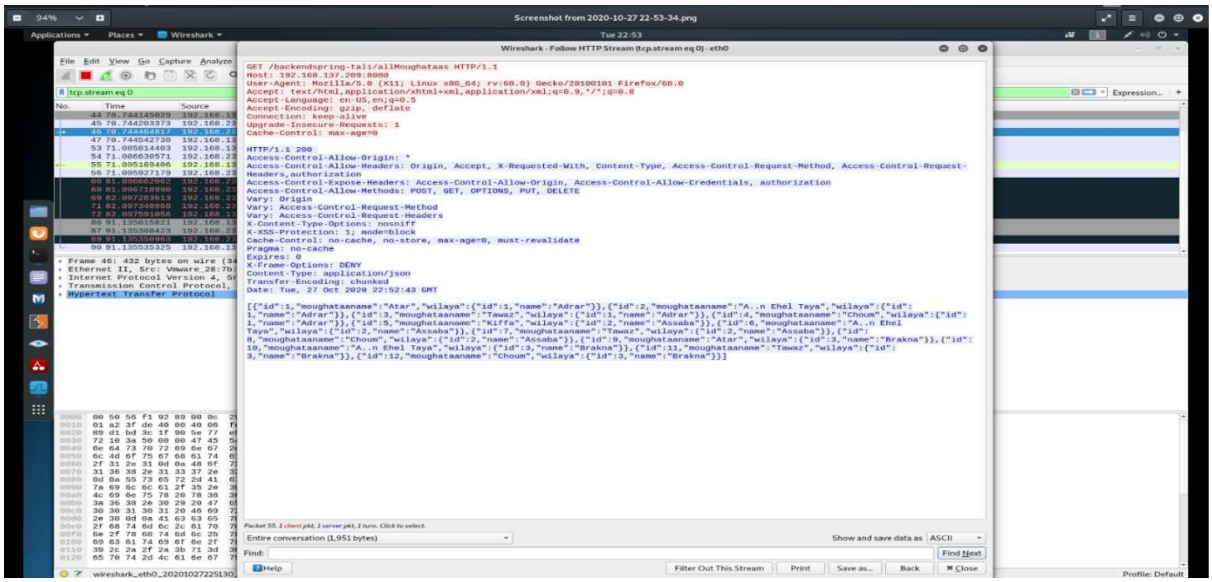


Figure 69 : Capture du paquet sur wireshark

❖ Pour utiliser HTTPS j'ai installé un certificat SSL est un type de certificat numérique qui fournit une authentification pour un site web et permet une connexion cryptée SSL, SSL garantit un chiffrement parfait car il est le protocole de sécurisé le plus répandu de nos jours, Les deux dernières Figures décrit les étapes de création du certificat SSL

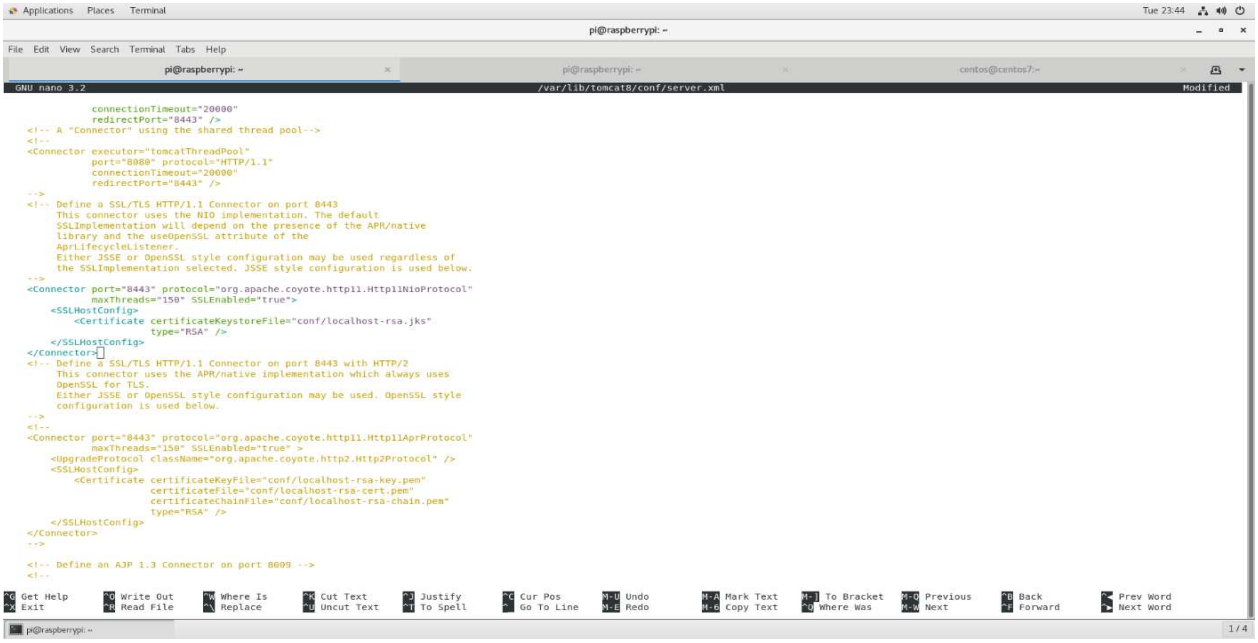


Figure 70 : Figure 1

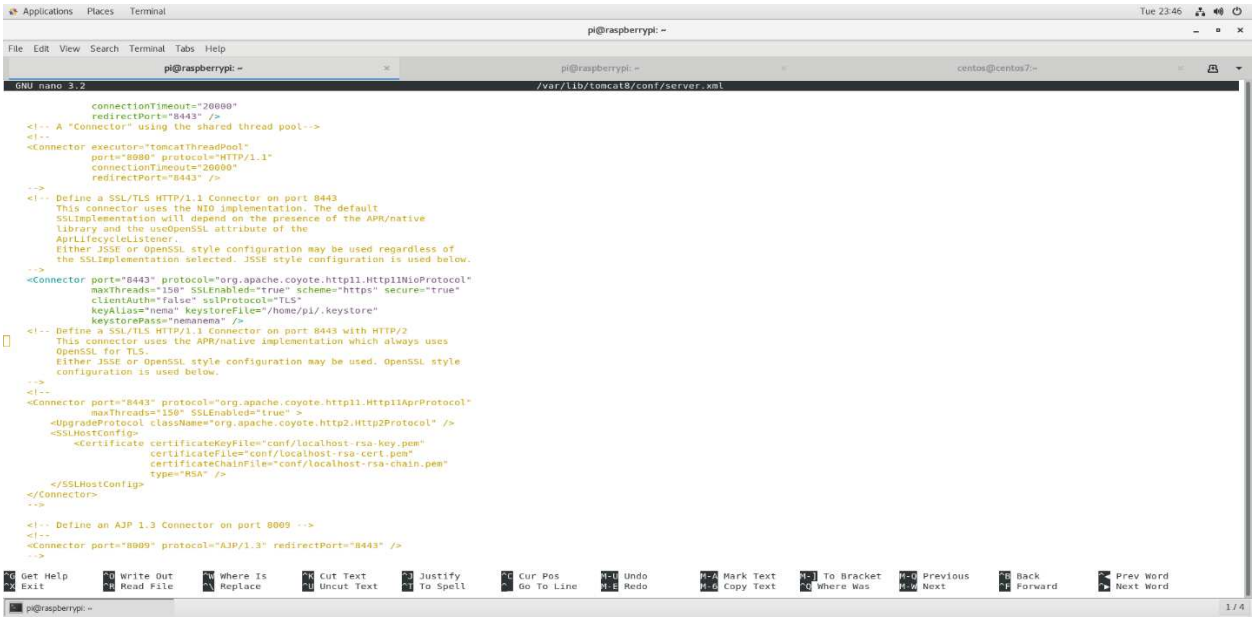


Figure 71 : Figure 2

Conclusion

Dans ce chapitre j'ai concentré sur l'aspect de confidentialisé pour éviter de changement des informations lors de l'envoi ou la réception des données, j'ai basé sur HTTPS qui optimise le référencement et la performance en garantissant les objectifs principaux de la sécurité.

Conclusion générale

Au cours de ce travail, j'ai présenté les différentes étapes ayant conduit à la mise en œuvre d'un système sécurisé de gestion des campagnes de vaccination.
J'ai commencé par recenser les difficultés que rencontre les équipes de vaccination et les responsables des campagnes de vaccination afin d'apporter la solution adéquate et spécifier ainsi les besoins.
Le langage de modélisation UML et la méthode Scrum ont constitués le support de l'analyse des besoins et la conception de notre application web via les différents diagrammes UML couvrant les aspects fonctionnels, dynamiques et statiques de tout le développement.

Pour enfin réaliser le système, j'ai utilisé le PhpMyAdmin MYSQL pour implémenter la base de données et les frameworks angular, spring boot et Android pour la partie mobile et d'autre part pour renforcer le taux de sécurité j'ai utilisé HTTPS afin de mieux protéger l'application.
Pour conclure, ce stage m'a permis à la fois de faire un grand pas vers le monde professionnel et à la fois de découvrir l'univers des start-up. Je vais ainsi pouvoir mettre à profit toutes les bonnes pratiques et connaissances acquises dans mes futurs emplois et projets.

Cependant des perspectives restent envisageables, telles que l'enrichissement du système en l'étendant pour qu'il puisse gérer tous les types de campagnes de vaccination.

Webographie

(1) https://fr.wikipedia.org/wiki/Programme_%C3%A9largi_de_vaccination
(2) https://en.wikipedia.org/wiki/Serum_(blood)
(3) https://en.uml.org/wiki/UMl)
(4) https://en.wikipedia.org/wiki/StarUML
(5) https://fr.wikipedia.org/wiki/IntelliJ_IDEA
(6) https://fr.wikipedia.org/wiki/WampServer
(7) https://fr.wikipedia.org/wiki/Android_Studio
(8) https://spring.io/
(9) https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#reference
(10) https://openlayers.org/
(11) https://fr.wikipedia.org/wiki/MySQL
(12) https://fr.wikipedia.org/wiki/JSON_Web_Token
(13) https://angular.io/guide/architecture
(14) https://maven.apache.org/
(15) https://www.w3schools.com/bootstrap4/
(16)https://ssl+wikipedia+https&oq=SSL+wikipedia7&sourceid=chrome&ie=U
(17) https://www.networh//kworld.com/article/32940/what
(18) https://fr.wikipedia+https&oq=SSL+wikipedia7&sour
(19) http://biblio.univ-antananarivo.mg/pdfs/rasamimisahobyhn_espa_ing_06.pdf
(20)https://www.networkworld.com/article/32940/what-is-nmap-why-you-need-this-network-mapper.html
(21) https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image
(22)https://www.vmware.com/products/workstation-pro.html
(23) https://en.wikipedia.org/wiki/CentOS
(24) https://en.wikipedia.org/wiki/Man-in-the-middle_attack
(25) https://www.ms.mr/