

République Algérienne Démocratique et Populaire

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

CENTRE UNIVERSITAIRE ABDELHAFID BOUSSOUF. MILA

INSTITUT SCIENCES ET TECHNOLOGIE

DÉPARTEMENT MATHÉMATIQUE ET INFORMATIQUE

FILIÈRE INFORMATIQUE

MÉMOIRE DE FIN D'ÉTUDES

EN VUE DE L'OBTENTION DU DIPLÔME DE MASTER EN INFORMATIQUE

OPTION STIC

Conception et développement d'une application android pour le suivi des objets mobiles

Réalisé par:

Hamimes OUSSAMA

Kedadra ABDELHAK

Encadé par:

Mr.Boukhechem NADHIR

Membres du jury

President: Mr.Guetich Mourad

Examineur: Mr.DIB Abderrahim

Année Universitaire:2018/2019

*Je dédie ce travail à:
Toute ma famille et tous mes amis.*

Oussama

*Je dédie ce travail à:
Ma famille: Ils vont trouver ici l'expression de mes
sentiments de respect et de reconnaissance pour leur
soutien, encouragements et grands sacrifices.*

*Mes frères et sœurs.
Tous mes amis et mes collègues.*

Abdelhak

Remerciement

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

En second lieu, nous tenons à remercier notre encadreur Mr: N.Boukhechem, son précieux conseil et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions, et à toute l'équipe pédagogique du centre universitaire Abdelhafid Boussouf, Mila.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Résumé

Le but de ce travail est de réaliser une application Android pour le suivi en temps réel des objets mobiles, cette application représente une solution utile qui peut être utilisée par les individus pour suivre ses enfants par exemple, ainsi que par les entreprises pour suivre ses ressources humaines et matérielles. En plus de ces fonctionnalités nécessaires, cette application intègre une fonction intelligente permet de faciliter la tâche de la surveillance, en effet, et après une tâche d'apprentissage, un surveillant intelligent peut remplacer la surveillance humaine.

Notre application se base sur l'architecture MVC (Model, Vue, Contrôleur). Pour la conception, nous avons utilisé le langage UML. Pour l'implémentation, on a choisi Android Studio et Laravel comme des environnements de développement, le système de gestion de bases de données MySQL, la bibliothèque PHP-ML pour l'implémentation des algorithmes de l'apprentissage automatique et l'API de Google Maps.

Mots clés: Géolocalisation, Google Maps, Machine Learning, UML, MVC, Android, Laravel.

Abstract

The aim of this work is to make an Android application for real-time tracking of mobile objects, this application represents a useful solution that can be used by individuals to track his children for example, as well as by companies to track his resources human and material. In addition to these necessary features, this application incorporates an intelligent function makes the task of monitoring easier, indeed, and after a learning task, a smart supervisor can replace human monitoring.

Our application is based on the MVC architecture (Model, View, Controller). For the design, we used the UML language. For the implementation, we chose Android Studio and Laravel as development environments, the MySQL database management system, the PHP-ML library for the implementation of the algorithms of the machine learning and the Google Maps API.

Keywords: Geolocation, Google Maps, Machine Learning, UML, MVC, Android, Laravel.

Table des matières

Remerciement	ii
Résumé	iii
Introduction générale	1
1 Généralités sur la géolocalisation et l'apprentissage automatique	3
1.1 Introduction	3
1.2 Définition de géolocalisation	3
1.3 Techniques de Géolocalisation	3
1.3.1 La géolocalisation par satellite	4
1.3.2 Géolocalisation par GSM	4
1.3.3 Géolocalisation par Wifi	4
1.3.4 Géolocalisation par adresse IP	4
1.3.5 Géolocalisation hybride	5
1.4 Plateformes logicielles de géolocalisation	5
1.4.1 Les composants de la plateforme	5
1.4.2 Le mode de fonctionnement de la plateforme	6
1.5 Applications pratiques de la géolocalisation	7
1.5.1 Applications professionnelles	7
1.5.2 Applications personnelles	8
1.6 L'apprentissage automatique	8
1.6.1 Domaines d'applications de l'apprentissage automatique	9
1.6.2 Données d'apprentissage	9
1.6.3 Types d'apprentissage	9
L'apprentissage supervisé	9
L'apprentissage non-supervisé	10
L'apprentissage par renforcement	10
1.6.4 La classification	10
1.6.5 Quelques algorithmes	10
1.7 Conclusion	11
2 Analyse	12
2.1 Introduction	12
2.2 La méthodologie	12
2.3 Spécification des besoins fonctionnels	12
2.3.1 Identification des acteurs	13
2.3.2 Analyse des besoins fonctionnels	13
2.4 Diagramme de cas d'utilisation global	15

2.4.1	Analyse détaillée de cas d'utilisation pour Le visiteur: . . .	16
2.4.2	Analyse détaillée de cas d'utilisation pour Client:	17
2.5	Spécification des besoins non fonctionnels	22
2.5.1	Les contraintes ergonomiques	22
2.5.2	Les contraintes techniques	22
2.5.3	Les contraintes du matériel	22
2.5.4	Les contraintes du déploiement	23
2.6	Conclusion	23
3	Conception	24
3.1	Introduction	24
3.1.1	Diagrammes de séquences	24
	Diagrammes de séquences Système	24
	Diagrammes de séquence détaillé	31
3.1.2	Diagramme de classes	38
3.1.3	Le modèle relationnel	39
3.1.4	Schémas de navigation	41
3.2	Conclusion	42
4	Réalisation	43
4.1	Introduction	43
4.2	Architecture MVC (Modèle, Vue et Contrôleur)	43
4.3	Environnement du développement de l'application	44
4.3.1	Environnement matériel	45
4.3.2	Environnement logiciel	46
4.3.3	Outils pour le développement web	47
4.3.4	GoogleMaps	48
4.3.5	PHP Machine learning	49
4.4	Implémentation et Test	50
4.4.1	Implémentation de la base de données	50
4.4.2	Les Scénarios d'exécution	50
	1.L'application mobile	50
	2.L'application Web	53
4.5	Conclusion	55
	Conclusion générale et Perspectives	56
A	Le GPS	57
A.1	définition	57
A.2	Composants du GPS	57
A.3	Les coordonnées géographiques	58
A.4	Avantages du GPS	59
A.5	Conclusion	60
B	ANDROID	61
B.1	Présentation du système d'exploitation ANDROID	61
B.2	Architecture d'ANDROID	62
B.2.1	Kernel Linux	62

B.2.2	Les Bibliothèques	63
B.2.3	Android Runtime	64
B.2.4	Couche Application Framework	64
B.2.5	Couche Applications	65
B.3	Les applications Android	65
B.3.1	Présentation	65
B.3.2	État d'une activité (application)	65
B.3.3	Cycle de vie d'une activité Android	66
B.4	Conclusion	67

Table des figures

1.1	Logo de l'application Geoloc Systems	7
1.2	Logo de l'application KidControl	8
2.1	Le diagramme de cas d'utilisation global	15
3.1	Diagramme de séquence système « S'enregistrer »	25
3.2	Diagramme de séquence système « S'authentifier »	26
3.3	Diagramme de séquence système « Afficher sa Position »	27
3.4	Diagramme de séquence système « Gérer son Compte »	27
3.5	Diagramme de séquence système « Envoyer un Message »	28
3.6	Diagramme de séquence système « Consulter un Message »	29
3.7	Diagramme de séquence système « Supprimer un Message »	30
3.8	Diagramme de séquence détaillé « S'enregistrer »	31
3.9	Diagramme de séquence détaillé « S'authentifier »	32
3.10	Diagramme de séquence détaillé « Créer un groupe »	32
3.11	Diagramme de séquence détaillé « Envoyer le Code »	33
3.12	Diagramme de séquence détaillé « Modifier un groupe »	33
3.13	Diagramme de séquence détaillé « Modifier un groupe »	34
3.14	Diagramme de séquence détaillé « Envoyer un Message »	34
3.15	Diagramme de séquence détaillé « Consulter un Message »	35
3.16	Diagramme de séquence détaillé « Créer une Zone »	35
3.17	Diagramme de séquence détaillé « Supprimer une Zone »	36
3.18	Diagramme de séquence détaillé « Gérer le Surveillant intelligent »	36
3.19	Diagramme de séquence détaillé « Suivre les Utilisateurs »	37
3.20	Diagramme de séquence détaillé « Envoyer une Alerte »	37
3.21	Diagramme de classes	38
3.22	Schéma de navigation pour l'application Mobile	41
3.23	Schéma de navigation pour l'application Web	41
4.1	Le schéma de l'architecture MVC	44
4.2	Android Studio	46
4.3	Java Development Kit	46
4.4	framework laravel.png	47
4.5	Le package WampServer	48
4.6	Les tables de la base de données.	50
4.7	Interface de la page principale.	51
4.8	S'enregistrer et S'authentifier.	51
4.9	Créer un groupe et la liste des utilisateurs.	52
4.10	Boite des Messages et Lire un message.	52
4.11	Interface « Afficher sa position »	53

4.12	Interface liste des groupes.	53
4.13	Interface liste des utilisateurs.	54
4.14	Interface afficher liste des messages.	54
4.15	Interface afficher « afficher sa position »	55
A.1	Composants du GPS	58
A.2	Les coordonnées géographiques	58
B.1	Évolution des versions d'ANDROID	61
B.2	L'architecture ANDROID	63
B.3	Cycle de vie d'une application ANDROID	66

Liste des tableaux

2.1	Analyse de cas d'utilisation « afficher sa position »	16
2.2	Analyse de cas d'utilisation « S'enregistrer »	17
2.3	Analyse de cas d'utilisation « S'authentifier »	17
2.4	Analyse de cas d'utilisation « Gérer les messages »	18
2.5	Analyse de cas d'utilisation « Gérer les groupes »	19
2.6	Analyse de cas d'utilisation « Suivre les Utilisateurs »	19
2.7	Analyse de cas d'utilisation « Gérer les Utilisateurs »	20
2.8	Analyse de cas d'utilisation « Gérer les alertes »	21
2.9	Analyse de cas d'utilisation « Gérer l'historique »	21
2.10	Analyse de cas d'utilisation « Gérer le Surveillant intelligent »	22
A.1	Présentation du GPS selon le milieu.	60
B.1	Détails des versions d'ANDROID.	62

Liste des Abréviations

GPS	G lobal P ositionning S ystem
GSM	G lobal S ystem for M obile C ommunications
GPRS	G eneral P acket R adio S ervice
WIFI	W ireless F idelity
IP	I nternet P rotocol
UML	U nified M odeling L anguage
MVC	M odel V ue C ontrôler
AOA	A ngle O f A rrival
TDOA	T ime D ifference O f A rrival
TOA	T ime O f A rrival
Cell ID	C ell I dentification
API	A pplication P rogramming I nterface
UMTS	U niversal M obile T elecommunications S ystem
RSS	R eceived S ignal S trength
SSID	S ervice S et I dentifier
MAC	M edia A ccess C ontrol
SMS	S hort M essage S ervice
VM	V irtual M achine
URL	U niform R esource L ocator
IANA	I nternet A ssigned N umbers A uthorit

Introduction générale

De nos jours, les Smartphones font partie intégrante de notre société. Ils sont connectés au monde entier grâce aux différentes technologies, l'une de ces technologies est la géolocalisation. En effet, connaître la position d'un objet mobile (personne ou machine) à l'instant T devient un véritable atout pour les individus et les entreprises. Cette traçabilité offre de réels avantages permettant dans le cas des entreprises de gérer d'une manière centralisée l'ensemble de ses ressources, ceci donne l'occasion par exemple d'améliorer la qualité de prestations, de réduire les coûts de déplacement, de garantir la sécurité des conducteurs et de prendre en main le contrôle de ses matériels. Dans le cas des individus, la géolocalisation offre la possibilité de localiser ses biens (véhicule..) et spécialement ses enfants pour éviter les risques de vols et de kidnapping.

Dans cette perspective, notre travail vise à concevoir et à réaliser une application qui permettra aux entreprises ainsi qu'aux individus de gérer et de suivre en temps réel les différents objets en mouvement sur une carte géographique. L'application offrira de nombreuses fonctionnalités notamment les notifications automatiques grâce à un agent intelligent.

Ce mémoire est organisé en quatre chapitres comme suit:

- Le premier chapitre donne des généralités sur les différentes technologies de géolocalisation comme les (GPS, GSM, WIFI, adresse IP), et présente le principe de fonctionnement des plateformes de géolocalisation. Ainsi que des notions sur le domaine de l'apprentissage automatique (Machine learning).
- Le deuxième chapitre est réservé pour la spécification des besoins qui nous permet de décrire sans ambiguïté l'application à développer à travers le diagramme de cas d'utilisation.
- Le troisième chapitre présente en détail la conception de l'application via le diagramme de classes et les diagrammes de séquences.
- Le quatrième chapitre est dédié à la réalisation, où nous présentons l'environnement de développement ainsi que quelques interfaces de notre application.

Nous terminons notre travail par une conclusion générale en évoquant les différentes perspectives que nous envisageons d'aborder dans le futur.

En plus de ces chapitres, nous avons aussi ajouté deux annexes pour bien éclaircir quelques notions:

- Annexe A: Présente la technologie GPS, ces composants, et nous avons expliqué les notions de coordonnées géographiques essentielles pour la réalisation de notre application.
- Annexe B: Présente la plateforme Android, son évolution, son architecture et le cycle de vie d'une application Android.

Chapitre 1

Généralités sur la géolocalisation et l'apprentissage automatique

1.1 Introduction

L'histoire humaine est pleine d'inventions qui avaient pour but de permettre aux populations mais surtout à leurs gouvernants, de décrire les territoires pour pouvoir mieux les contrôler et éventuellement en chasser les ennemis (La géographie ça sert d'abord à faire la guerre). De l'astrolabe antique, en passant par la boussole, le sextant, le gyroscope et le satellite jusqu'au GPS d'aujourd'hui, grâce à ce développement qui a donné naissance la géolocalisation. Dans ce chapitre on va décrire brièvement les différentes techniques utilisées ainsi que les plateformes logicielles de géolocalisation. On va présenter aussi quelques notions sur le domaine de l'apprentissage automatique qui est devenu une approche avantageuse dans le développement des applications.

1.2 Définition de géolocalisation

C'est une technologie permettant de déterminer la localisation d'un objet ou d'une personne avec une certaine précision. La technologie s'appuie généralement sur le système GPS ou sur les interfaces de communication d'un téléphone mobile. Les applications de la géolocalisation sont multiples: de l'assistance à la navigation, à la mise en relation des personnes, mais aussi à la gestion en temps réel des moyens en personnel et en véhicules des entreprises,...etc. [5]

1.3 Techniques de Géolocalisation

La géolocalisation existe dans des différentes technologies. Dans la partie suivante, on va visualiser les plus importantes techniques de la géolocalisation.

1.3.1 La géolocalisation par satellite

La géolocalisation par satellite permet de situer, sur la face terrestre, un terminal équipé d'une puce. Cette puce doit être compatible avec les signaux émis par une constellation de satellites prévue à cet effet. La position détectée et traduite en termes de latitude, longitude et parfois altitude peut alors être représentée physiquement sur une carte. L'utilisation la plus connue de cette technique est le système américain GPS (Global Positioning System)[Voir Annexe A]. On trouve aussi le système russe GLONASS, le système chinois BEIDOU et le système européen GALILEO dont le déploiement doit s'achever vers 2020.

1.3.2 Géolocalisation par GSM

Le GSM (Global System for Mobile communication) est un standard pour les communications de téléphonie mobile. Le GSM permet de transmettre numériquement non seulement la voix, mais aussi des données informatiques (connexion possible à internet, transmission de fax ou de messages électroniques). Les applications de localisation à l'aide du réseau GSM sont apparues au début des années 2000. Plusieurs techniques de localisation existent. On trouve des méthodes d'identification de la cellule dans laquelle se trouve le mobile (méthode de Cell ID ou Cell Identification), puis des méthodes plus complexes basées sur des mesures temporelles (estimation de la distance entre chacune des stations de base et le mobile) comme TOA (Time Of Arrival) ou TDOA (Time difference Of Arrival). Enfin, l'utilisation des directions d'arrivée avec la méthode AOA (Angle Of Arrival) conduit aussi à l'identification de la position occupée par le mobile. Chacune de ces méthodes est développée par la suite car elles sont aussi proposées pour les réseaux de nouvelle génération de type UMTS. [7]

1.3.3 Géolocalisation par Wifi

La géolocalisation Wi-Fi est utilisée pour déterminer la position d'un terminal possédant une balise active. Cette localisation repose sur le mécanisme classique de triangulation, la mesure de l'intensité du signal reçu (Received Signal Strength ou RSS) du point d'accès. [12]

1.3.4 Géolocalisation par adresse IP

Cette méthode permet de déterminer la position géographique d'un ordinateur ou de n'importe quel terminal connecté à internet en se basant sur son adresse IP. Les adresses IP sont gérées par l'IANA (Internet Assigned Numbers

Authorit), une organisation qui s'occupe de découper les blocs d'adresse IP disponibles et de les distribuer de façon très contrôlée aux pays qui en demandent [12].

Le principe est d'identifier à partir de l'adresse IP utilisée par un Internaute pour la comparer en temps réel à un référentiel d'adresses IP comprenant leur localisation. C'est le point d'accès technique de l'internaute qui est géolocalisé. Avec les meilleurs référentiels d'adresses IP, la précision de la géolocalisation IP permet une précision fiables à l'échelle de la ville.

1.3.5 Géolocalisation hybride

Aucune technique n'est absolue, chacune a son propre défaut, généralement accordée à leur dépendance à un certain réseau.

- La dépendance au réseau GPS: l'incapacité de l'utiliser en intérieur;
- La dépendance au réseau GSM: sa couverture géographique, l'accès au réseau GPRS pour exploiter l'information;
- La dépendance à la présence de bornes d'accès Wi-Fi: en zone rurale par exemple.

Dans certain système de localisation, différentes technologies ou métriques sont combinée pour améliorer la précision de la localisation et réduire la consommation de l'énergie.

La plus part des Smartphones actuels sont capables d'utiliser une méthode hybride de géolocalisation grâce à ses interfaces GSM/UMTS, Wi-Fi et à son récepteur GPS.

1.4 Plateformes logicielles de géolocalisation

1.4.1 Les composants de la plateforme

Les composants essentiels d'une plateforme de géolocalisation sont les suivants:

1. Terminal communicant: c'est le terminal qui reçoit les coordonnées géographiques (via GPS ou tout autre moyen) et qui les envoie via un réseau de télécommunications à la plateforme;
2. Système informatique: capable de recevoir, stocker et traiter les informations: il s'agit des serveurs informatiques qui hébergent l'infrastructure et qui reçoivent et traitent les données envoyées par les terminaux. Ce sont les mêmes serveurs qui mettront à disposition l'information aux utilisateurs (via une interface web par exemple);

3. Module cartographique: c'est le module intégré au système informatique qui va permettre d'afficher la position des terminaux sur un fond cartographique adapté. Ce module prend en charge les calculs de distances et d'itinéraires, détecte l'interaction avec les zones et permet d'avoir accès à des informations sur le terrain (sens interdits, restrictions pour les poids lourds, vitesses autorisées...).

1.4.2 Le mode de fonctionnement de la plateforme

La position géographique d'un terminal géolocalisé reste néanmoins une information brute qui peut être exploitée et couplée à d'autres données afin de créer une vaste quantité de services à forte valeur ajoutée.

Afin d'exploiter cette information, la donnée (position) générée par un terminal qui se trouve sur le terrain doit être transmise à une plateforme logicielle qui la traite, la présente graphiquement à l'utilisateur et l'associe à d'autres données afin d'enrichir les informations relatives à l'état du terminal ou de la flotte de terminaux.

Voici les étapes de la chaîne de traitement:

1. Le terminal détermine sa position géographique grâce à une des techniques de géolocalisation citées précédemment (de préférence GPS, GSM et/ou Wi-Fi);
2. Il envoie ces données vers une plateforme logicielle, soit par le réseau GSM/GPRS, soit par un réseau satellitaire;
3. La plateforme logicielle de géolocalisation traite la donnée et positionne le terminal géographiquement sur une carte moyennant la précision offerte par la technique de géolocalisation utilisée. De plus, en combinant plusieurs informations, notamment récupérées via un système de télérelève (trafic routier, autonomie du véhicule, points à visiter, etc.), des calculs d'itinéraires ou de tournées peuvent par exemple être générés;
4. La carte ainsi que tous les traitements effectués sont mis à disposition de l'utilisateur par le biais d'un portail web hébergé sur un serveur accessible depuis Internet ou par l'entremise d'une application métier installée sur le poste de travail.

1.5 Applications pratiques de la géolocalisation

Les applications de la géolocalisation sont en plein développement, tant sur le plan privé que sur le plan professionnel. De plus, couplées à des systèmes de télérelève intégrés et sur mesure, de vraies applications métier ont rapidement vu le jour.

1.5.1 Applications professionnelles

La géolocalisation dans le milieu professionnel est presque toujours synonyme de gain de productivité, d'économies de carburant, d'économies de communications et de sécurité accrue. De plus, ces solutions offrent aux responsables de l'exploitation du parc une vision globale et un meilleur temps de réactivité en cas d'incident. Cela permet à l'entreprise utilisant un système de géolocalisation d'améliorer son service client et de réduire ses coûts afin d'accroître sa compétitivité.

Quelques domaines dans lesquels la géolocalisation est communément utilisée sont listés ci-dessous:

- Transport de marchandises et logistique;
- Transport de passagers;
- Suivi et protection de personnes;
- Protection de marchandises, véhicules et antivol.

Exemple

Geoloc Systems est une solution pour géolocalisation des conducteurs, elle aide à connaître la position précise des véhicules de l'entreprise. Afin d'anticiper les coûts, l'entreprise possède une visibilité sur les horaires de ses employés mais également sur les frais de carburant.



FIGURE 1.1 – Logo de l'application Geoloc Systems.

1.5.2 Applications personnelles

Sur le plan personnelle, les applications de géolocalisation offrent des fonctionnalités tel que:

- Suivi et sécurité des personnes;
- Sécurité des animaux.

Exemple

KidControl Cette application gratuite est très complète: elle permet de géolocaliser le téléphone de vos proches, de consulter l'historique de leurs déplacements, de recevoir une alerte si votre enfant entre dans une zone que vous aurez définie (l'école, la maison, ou bien un endroit dangereux où il n'a pas le droit d'aller) ou quand la batterie du téléphone de votre enfant est faible, d'échanger des messages.



FIGURE 1.2 – Logo de l'application KidControl.

1.6 L'apprentissage automatique

L'apprentissage automatique fait référence au développement, l'analyse et l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer et de remplir des tâches associées à une intelligence artificielle grâce à un processus d'apprentissage. Cet apprentissage permet d'avoir un système qui s'optimise en fonction de l'environnement, les expériences et les résultats observés [1].

1.6.1 Domaines d'applications de l'apprentissage automatique

L'apprentissage automatique s'applique à un grand nombre d'activités humaines et convient en particulier au problème de la prise de décision automatisée. Il s'agira, par exemple:

- D'établir un diagnostic médical à partir de la description clinique d'un patient;
- De donner une réponse à la demande de prêt bancaire de la part d'un client sur la base de sa situation personnelle;
- De déclencher un processus d'alerte en fonction de signaux reçus par des capteurs;
- De la reconnaissance des formes;
- De la reconnaissance de la parole et du texte écrit;
- De contrôler un processus et de diagnostiquer des pannes.

1.6.2 Données d'apprentissage

Les données d'apprentissage sont, souvent, réparties en 3 catégories: [21]

L'ensemble d'apprentissage ou population d'entraînement: constitue l'ensemble des candidats ou exemples (images, attributs, DB, ...) utilisés pour générer le modèle d'apprentissage.

L'ensemble de Test: est constitué des candidats sur lesquels sera appliqué le modèle d'apprentissage (pour tester et corriger l'algorithme).

L'ensemble de validation: peut être utilisé lors de l'apprentissage (comme sous population de l'ensemble d'apprentissage) afin de valider (intégrer) le modèle et d'éviter le sur-apprentissage.

NB: Selon les domaines, les données d'apprentissage peuvent être de diverses formes: mots, phrases, variables ou attributs, des vecteurs de valeurs: définissant un ensemble de propriétés d'un objet,...etc

1.6.3 Types d'apprentissage

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient:[21]

L'apprentissage supervisé

Si les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classement; on parle alors d'apprentissage supervisé (ou d'analyse discriminante).

L'apprentissage non-supervisé

Quand le système ou l'opérateur ne dispose que d'exemples, mais non d'étiquettes, et que le nombre de classes et leur nature n'ont pas été prédéterminés, on parle d'apprentissage non supervisé (ou clustering). Aucun expert n'est disponible ni requis. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données.

L'apprentissage par renforcement

L'algorithme apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme.

1.6.4 La classification

Nous dénotons x l'espace d'entrée, c'est-à-dire l'ensemble contenant les descriptions du problème à résoudre, et y l'espace de sortie, c'est-à-dire l'ensemble contenant toutes les solutions possibles.

La classification consiste à catégoriser des exemples parmi un ensemble de classes possibles. Le cadre de classification le plus étudié est la classification binaire, où un algorithme doit apprendre à distinguer parmi deux différentes classes, couramment appelées les classes positive et négative. On a donc $y = \{-1, 1\}$. Certains types de problèmes n'ont qu'une seule classe, qui peut être considérée comme la classe « normale ». On parle alors de détection d'anomalie, où l'on tente de détecter les exemples qui sortent de l'ordinaire. La classification multi-classe consiste simplement à la situation où chaque exemple appartient à une catégorie parmi un ensemble de $k > 2$ catégories. Dans cette situation, nous avons $y = 1, 2, \dots, k$. Finalement, la classification multi-étiquette correspond à la situation où chaque exemple peut appartenir à zéro, une ou plusieurs catégories à la fois, parmi un ensemble de k catégories possibles. L'espace de sortie y est donc un vecteur de k éléments $\{0, 1\}^k$, où chaque position correspond à une catégorie, et où on observe un 1 si l'exemple appartient à cette catégorie et 0 autrement. [19]

1.6.5 Quelques algorithmes

k Nearest Neighbors

KNN (k plus proches voisins) est une approche de classification supervisée intuitive, elle a pour but de classer des points cibles (classe méconnue) en fonction de leurs distances par rapport à des points constituant un échantillon

d'apprentissage (c'est-à-dire dont la classe est connue a priori). Les k-PPV nécessitent seulement un entier k , une base d'apprentissage et une métrique pour la proximité.

Naïve Bayes

est un algorithme probabiliste d'apprentissage automatique basé sur le théorème de Bayes. Un classifieur bayésien naïf suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques.

Réseaux de Neurones

Le neurone artificiel, reprend le fonctionnement du neurone biologique. Il reçoit des entrées et fournit une sortie, grâce à différentes caractéristiques:

- Des poids accordés à chacune des entrées, permettant de modifier l'importance de certaines par rapport aux autres;
- Une fonction d'agrégation, qui permet de calculer une unique valeur à partir des entrées et des poids correspondants;
- Un seuil (ou biais), permettant d'indiquer quand le neurone doit agir;
- Une fonction d'activation, qui associe à chaque valeur agrégée une unique valeur de sortie dépendant du seuil.[13]

1.7 Conclusion

Dans ce chapitre nous avons présenté les différentes méthodes de la géolocalisation qui permettent d'obtenir la position géographique d'une personne ou d'une ressource, on a expliqué les composants de la plateforme logicielle et son mode de fonctionnement. on a présenté aussi quelques exemples d'applications basées sur la géolocalisation, ainsi que des brèves descriptions sur le domaine de l'apprentissage automatique, ce qui nous a donné une idée très claire des technologies et de l'équipement à utiliser pour réaliser notre travail.

Chapitre 2

Analyse

2.1 Introduction

La spécification des besoins représente la première phase du cycle de développement d'une application. En effet, elle doit décrire sans ambiguïté l'application à développer. Dans ce chapitre nous allons spécifier l'ensemble des besoins fonctionnels et non fonctionnels liés à notre application. Ensuite, nous allons la modéliser à l'aide des diagrammes de cas d'utilisation.

2.2 La méthodologie

La première étape de ce projet consiste à définir les besoins fonctionnels de l'application à développer. Pour ce cela, nous avons opté pour une méthode générique personnel (analyse, conception, réalisation) qui utilise un sous-ensemble du langage de modélisation UML, nous avons trouvé ce sous ensemble suffisant pour atteindre notre objectif; Pour l'analyse et la conception on a utilisé respectivement le diagramme de cas d'utilisation, le diagramme de classe et le diagramme de séquence système détaillé.

2.3 Spécification des besoins fonctionnels

La partie fonctionnelle du modèle UML d'une application permet de spécifier les fonctionnalités offertes par l'application sans pour autant spécifier la façon dont ces fonctionnalités sont réalisées par les objets de l'application.[4]

Dans cette partie nous identifions les acteurs de notre future application afin de pouvoir dégager ces besoins fonctionnels.

2.3.1 Identification des acteurs

L'application doit fournir un ensemble de fonctionnalités aux clients, qui permettront aux clients de localiser et de suivre des objets sur une carte géographique. Nous allons énumérer, dans ce qui suit, les acteurs susceptibles d'interagir avec le système.

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié[17]. Les acteurs de notre application sont:

1- Le Visiteur

Toute personne qui bénéficie des fonctionnalités minimales de l'application après l'avoir installée sur son terminale sans avoir besoin de s'enregistrer ou de s'authentifier.

2- Le Client

Toute personne inscrite dans l'application, et qui peut utiliser des fonctionnalités plus avancées de l'application.

3- L'administrateur

C'est le superviseur qui contrôle, et exploite les données et qui doit gérer le bon fonctionnement du système.

4- Le Surveillant intelligent

C'est un acteur artificiel intelligent qui sert au suivi automatique des objets mobiles.

2.3.2 Analyse des besoins fonctionnels

Dans la suite, nous désignons les différents services offerts par notre application aux différents acteurs:

1-Coté Visiteur

Une fonctionnalité minimale est offerte au visiteur, il a la possibilité de:

Afficher sa position: Localiser et suivre son terminal sur une carte géographique, et déterminer sa vitesse de mouvement.

S'enregistrer: S'inscrire pour devenir un client.

2- Coté Client

Les fonctionnalités offertes au client:

S'authentifier: Après l'inscription, Un compte est attribué à chaque client inscrit. Le système doit permettre au client qui est déjà inscrit de passer à l'étape de l'authentification.

Gérer son compte: Le client a la possibilité de modifier les données de son compte.

Gérer les groupes: Le client a la possibilité de créer des groupes de suivi et d'inviter d'autres utilisateurs à rejoindre ses groupes.

Rejoindre un groupe: Le client peut recevoir une demande de suivi envoyé par d'autre client pour devenir un objet suivi, il peut répondre par accepter ou refuser (il donne sa permission).

Suivre les utilisateurs: Le client peut localiser et suivre en temps réel les utilisateurs ajoutés à un groupe sur une carte Google Maps.

Gérer les messages: Le client peut envoyer et recevoir des messages depuis et vers les utilisateurs du même groupe, et aussi consulter ses messages ou bien les supprimer.

Gérer les utilisateurs: Le client a la possibilité de supprimer un utilisateur, et de créer des zones de surveillance.

Gérer les alertes: Le client peut envoyer ou de recevoir des alertes.

Gérer l'historique: Le client a la possibilité de consulter, supprimer et d'exporter l'historique des utilisateurs.

Créer les rapports: Le client a la possibilité de générer des rapports.

Gérer le surveillant intelligent: le client peut activer ou désactiver le surveillant intelligent.

3- Coté administrateur

Les fonctionnalité offerte à l'administrateur:

Contrôler les Utilisateurs: Pour des raison de sécurité et de bon fonctionnement, l'administrateur peut supprimer des utilisateurs ou des groupes.

4- Coté Surveillant intelligent

Envoyer des notifications automatiques: Le surveillant intelligent peut envoyer des notifications automatiques à l'utilisateur suivi et au propriétaire de groupe.

2.4 Diagramme de cas d'utilisation global

La figure suivante désigne le diagramme de cas d'utilisation global Administrateur-Client-Visiteur-Surveillant intelligent dans lequel nous allons mettre en évidence les fonctionnalités offerts par l'application.

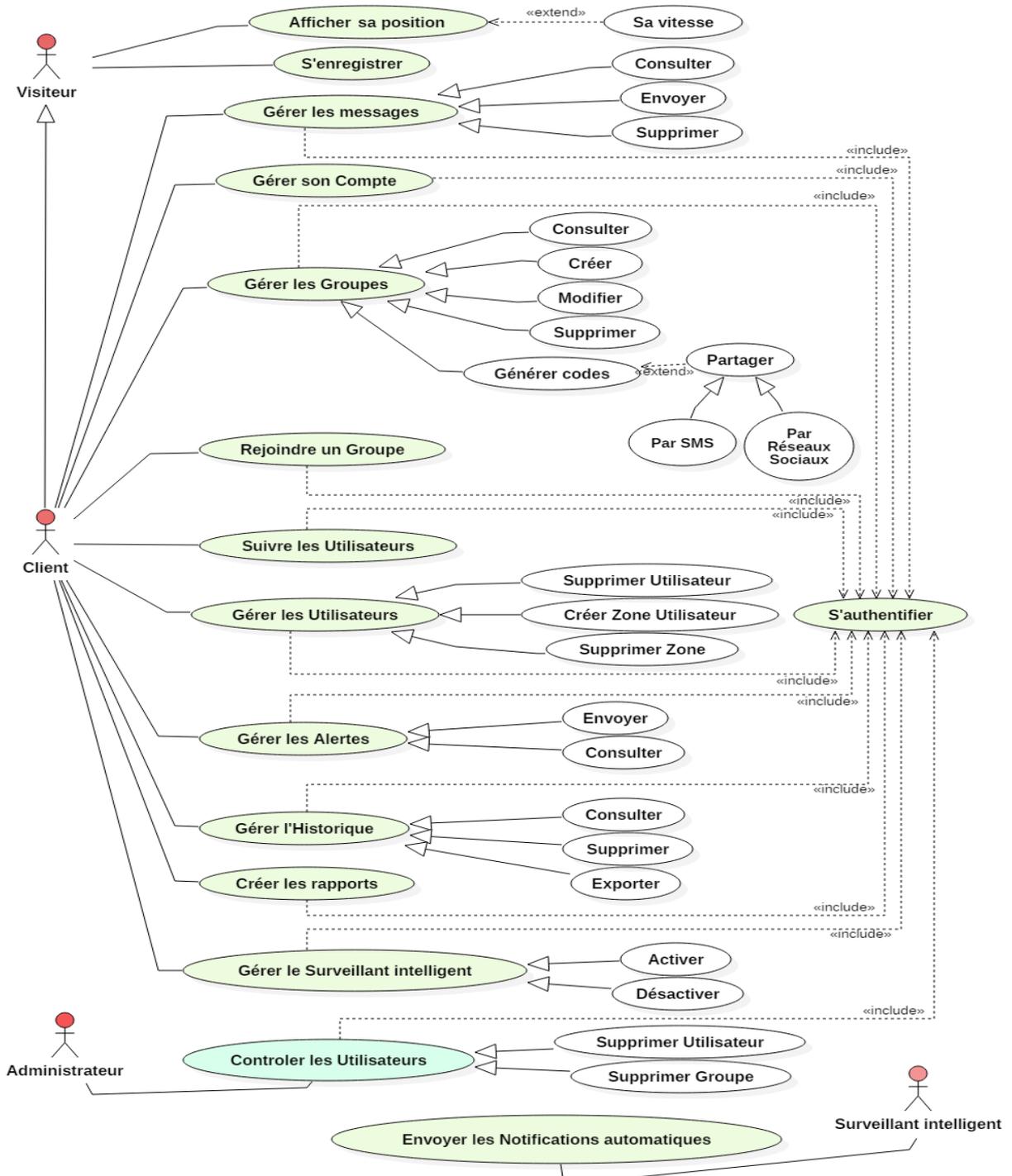


FIGURE 2.1 – Le diagramme de cas d'utilisation global.

2.4.1 Analyse détaillée de cas d'utilisation pour Le visiteur:

Nous allons décrire le diagramme de cas d'utilisation pour Le visiteur.

1. Cas d'utilisation (afficher sa position):

Sommaire:	
Titre:	afficher sa position
But:	trouver sa position sur une carte
Résumé:	Le visiteur demande de voir sa position géographique et consulter sa vitesse.
Acteur:	Le visiteur
Description des enchainements:	
Pré Conditions:	Post Conditions:
Le visiteur doit installer l'application	accéder à la carte et voir sa position
Scénario nominale:	
<ol style="list-style-type: none"> 1. démarrer l'application. 2. demander sa position. 3. accéder à la carte et voir sa position. 4. afficher la vitesse de mouvement sur la carte. 	

TABLE 2.1 – Analyse de cas d'utilisation « afficher sa position »

2. Cas d'utilisation (S'enregistrer):

Sommaire:	
Titre:	S'enregistrer
But:	Créer un compte
Résumé:	Le visiteur introduit ses données et valider
Acteur:	Le visiteur
Description des enchainements:	
Pré Conditions:	Post Conditions:
Le visiteur doit installer l'application	Le visiteur devient un client

Scénario nominale:
<ol style="list-style-type: none"> 1. démarrer l'application. 2. demander de s'enregistrer. 3. entrer les données requises et valider.
Enchaînement alternatif:
<ol style="list-style-type: none"> 1. Si les données ne sont pas valides ou incomplètes, le système affiche un message d'erreur.

TABLE 2.2 – Analyse de cas d'utilisation « S'enregistrer »

2.4.2 Analyse détaillée de cas d'utilisation pour Client:

Nous allons décrire le diagramme de cas d'utilisation pour le client.

1. Cas d'utilisation (S'authentifier):

Sommaire:	
Titre:	S'authentifier
But:	S'authentifier
Résumé:	Le client introduit son login et mot de passe
Acteur:	Le client
Description des enchaînements:	
Pré Conditions:	Post Conditions:
le client doit avoir un compte	Donne accès à la page principale
Scénario nominale:	
<ol style="list-style-type: none"> 1. démarrer l'application. 2. demande de login. 3. entrer son e_mail et mot de passe et valider. 	
Enchaînement alternatif:	
<ol style="list-style-type: none"> 1. Si les données ne sont pas valides ou incomplètes, le système affiche un message d'erreur. 	

TABLE 2.3 – Analyse de cas d'utilisation « S'authentifier »

2. Cas d'utilisation (Gérer les messages):

Sommaire:	
Titre:	Gérer les messages
But:	Gestion des messages
Résumé:	envoyer, consulter et supprimer des messages
Acteur:	Le client
Description des enchainements:	
Pré Conditions:	Post Conditions:
le client est authentifié	fonctionnalités de gestion de messages
Scénario nominale:	
<ol style="list-style-type: none"> 1. sélectionner le destinataire. 2. rédiger le message. 3. envoyer le message. 	
Enchainement alternatif:	
<ol style="list-style-type: none"> 1. le client peut supprimer les messages. 	

TABLE 2.4 – Analyse de cas d'utilisation « Gérer les messages »

3. Cas d'utilisation (Gérer les groupes):

Sommaire:	
Titre:	Gérer les groupes
But:	Gestion des groupes de suivis
Résumé:	Créer, modifier et supprimer des groupes de suivis, Générer et envoyer les Codes
Acteur:	Le client
Description des enchainements:	
Pré Conditions:	Post Conditions:
le client est authentifié	fonctionnalités de gestion des groupes

Scénario nominale:
<ol style="list-style-type: none"> 1. créer un groupe. 2. générer un code unique. 3. choisir la méthode d'envoi du code et Envoyer.
Enchaînement alternatif:
<ol style="list-style-type: none"> 1. Si le client récepteur de code ne confirme pas, il ne sera pas membre de groupe 2. le client peut modifier ou supprimer le groupe 3. le client peut régénérer et envoyer un autre code pour des raisons de sécurité

TABLE 2.5 – Analyse de cas d'utilisation « Gérer les groupes »

4. Cas d'utilisation (Suivre les Utilisateurs):

Sommaire:	
Titre:	Suivre les Utilisateurs
But:	Localiser et suivre des utilisateurs sur carte
Résumé:	le client localise et suivre en temps réel des utilisateurs sur carte
Acteur:	Client
Description des enchaînements:	
Pré Conditions:	Post Conditions:
le client est authentifié	Suivre les utilisateurs sur carte
Scénario nominale:	
<ol style="list-style-type: none"> 1. choisir un groupe de suivi. 2. choisir de localiser des utilisateurs. 3. suivre les utilisateurs sur la carte. 	

TABLE 2.6 – Analyse de cas d'utilisation « Suivre les Utilisateurs »

5. Cas d'utilisation (Gérer les Utilisateurs):

Sommaire:	
Titre:	Gérer les Utilisateurs
But:	Gérer les utilisateurs suivis
Résumé:	affecter un utilisateur à une zone, supprimer des utilisateurs
Acteur:	Le Client
Description des enchainements:	
Pré Conditions:	Post Conditions:
le client est authentifié	zone crée pour un utilisateur
Scénario nominale:	
<ol style="list-style-type: none"> 1. sélectionner un utilisateur. 2. créer une zone. 3. définir ses paramètres et valider. 	
Enchainement alternatif:	
<ol style="list-style-type: none"> 1. Le client peut supprimer une zone 2. Le client peut supprimer un utilisateur 	

TABLE 2.7 – Analyse de cas d'utilisation « Gérer les Utilisateurs »

6. Cas d'utilisation (Gérer les alertes):

Sommaire:	
Titre:	Gérer les alertes
But:	Gérer les alertes
Résumé:	consulter et envoyer les alertes
Acteur:	Le Client
Description des enchainements:	
Pré Conditions:	Post Conditions:
Le Client est authentifié	alertes envoyés

Scénario nominale:
<ol style="list-style-type: none"> 1. sélectionner un utilisateur. 2. envoyer une alerte.

TABLE 2.8 – Analyse de cas d’utilisation « Gérer les alertes »

7. Cas d’utilisation (Gérer l’historique):

Sommaire:	
Titre:	Gérer l’historique
But:	Gérer l’historique
Résumé:	consulter, exporter et supprimer l’historique
Acteur:	Le Client
Description des enchainements:	
Pré Conditions:	Post Conditions:
Le Client est authentifié	affichage de l’historique
Scénario nominale:	
<ol style="list-style-type: none"> 1. sélectionner un utilisateur. 2. consulter l’historique. 3. choisir les paramètres de l’historique. 4. exporter l’historique. 	

TABLE 2.9 – Analyse de cas d’utilisation « Gérer l’historique »

8. Cas d’utilisation (Gérer le Surveillant intelligent):

Sommaire:	
Titre:	Gérer le Surveillant intelligent
But:	Gérer le Surveillant intelligent
Résumé:	activer ou désactiver le Surveillant intelligent
Acteur:	Le Client
Description des enchainements:	
Pré Conditions:	Post Conditions:
Le Client est authentifié	Surveillant intelligent activé ou désactivé

Scénario nominale:
<ol style="list-style-type: none"> 1. sélectionner un groupe ou un utilisateur. 2. activer le Surveillant intelligent.
Enchaînement alternatif:
<ol style="list-style-type: none"> 1. désactiver le Surveillant intelligent.

TABLE 2.10 – Analyse de cas d’utilisation « Gérer le Surveillant intelligent »

2.5 Spécification des besoins non fonctionnels

Les besoins non fonctionnels spécifient les propriétés du système telles que les contraintes d’environnement et d’implémentation, la performance, la maintenance, l’extensibilité et la flexibilité.

2.5.1 Les contraintes ergonomiques

Les contraintes ergonomiques sont les contraintes liées à l’adaptation entre les fonctionnalités de l’application, leurs interfaces et leur utilisation.

Pour notre application, nous devons obéir aux contraintes ergonomiques suivantes:

- Permettre un accès rapide à l’information;
- Interface simple et compréhensible;
- L’application doit être développée avec un langage compréhensif par l’utilisateur, présenter les informations d’une façon simple et claire.

2.5.2 Les contraintes techniques

- Il faut que toute interface de l’application soit homogène, en effet, les différentes pages doivent suivre le même modèle de représentation (couleurs, images, textes,...etc);
- Le code doit être extensible et maintenable pour faciliter toute opération d’amélioration ou d’optimisation.

2.5.3 Les contraintes du matériel

L’application sera installée sur un terminal (téléphone ou tablette) mobile doté d’un système d’exploitation ANDROID.

2.5.4 Les contraintes du déploiement

L'application client peut être téléchargeable à partir de google play store (<https://play.google.com>), ou d'autre alternatif comme <https://fr.uptodown.com>.

2.6 Conclusion

La spécification des besoins nous a permis de définir les fonctionnalités de notre application, ce qui nous mène à entamer la phase de la conception (technique et graphique) pour assurer une bonne mise en œuvre d'un système fonctionnel répondant aux besoins cités.

Chapitre 3

Conception

3.1 Introduction

Ce chapitre sera consacré à conception de la future application avec la réalisation des diagrammes UML; le diagramme de classe pour définir la vue statique, et les diagrammes de séquence (système et détaillé) pour définir les interactions entre les différents composants de notre application.

3.1.1 Diagrammes de séquences

Dans cette partie, nous avons établi les diagrammes de séquence système et les diagrammes de séquence détaillé pour les cas d'utilisations précédents:

Diagrammes de séquences Système

Chaque cas d'utilisation est décrit textuellement de façon détaillée, mais donne également lieu à un diagramme de séquence simple représentant graphiquement la chronologie des interactions entre les acteurs et le système vu comme une boîte noire, dans le cadre du scénario nominal. Nous appellerons ce diagramme « diagramme de séquence système ».[17]

Nous allons présenter les diagrammes de séquence système des cas d'utilisation que nous avons déjà cité dans le chapitre précédent.

1. Diagramme de séquence système « S'enregistrer »

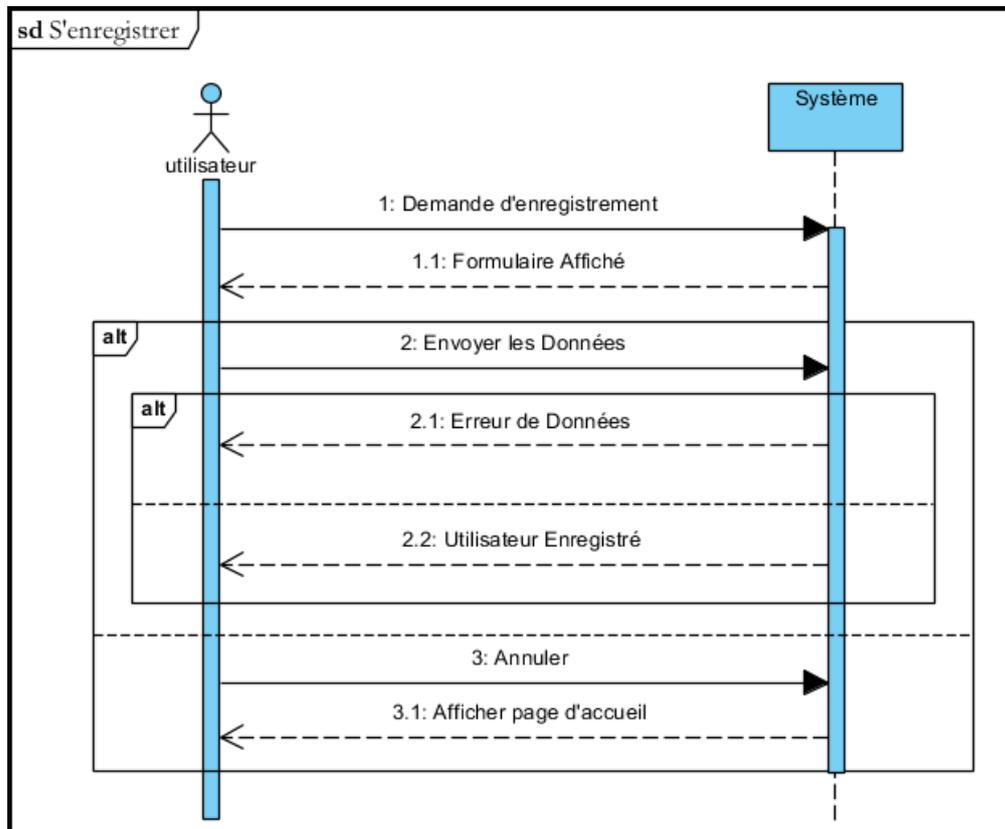


FIGURE 3.1 – Diagramme de séquence système « S'enregistrer »

L'utilisateur demande de S'enregistrer sur l'application, le système lui affiche un formulaire où il doit saisir ses données, il peut choisir de les envoyer, le système lui transmette une confirmation, si les données sont valides, sinon un message d'erreur est affiché. il peut également choisir d'annuler l'opération d'enregistrement, le system retourne à la page précédente.

2. Diagramme de séquence système « S'authentifier »

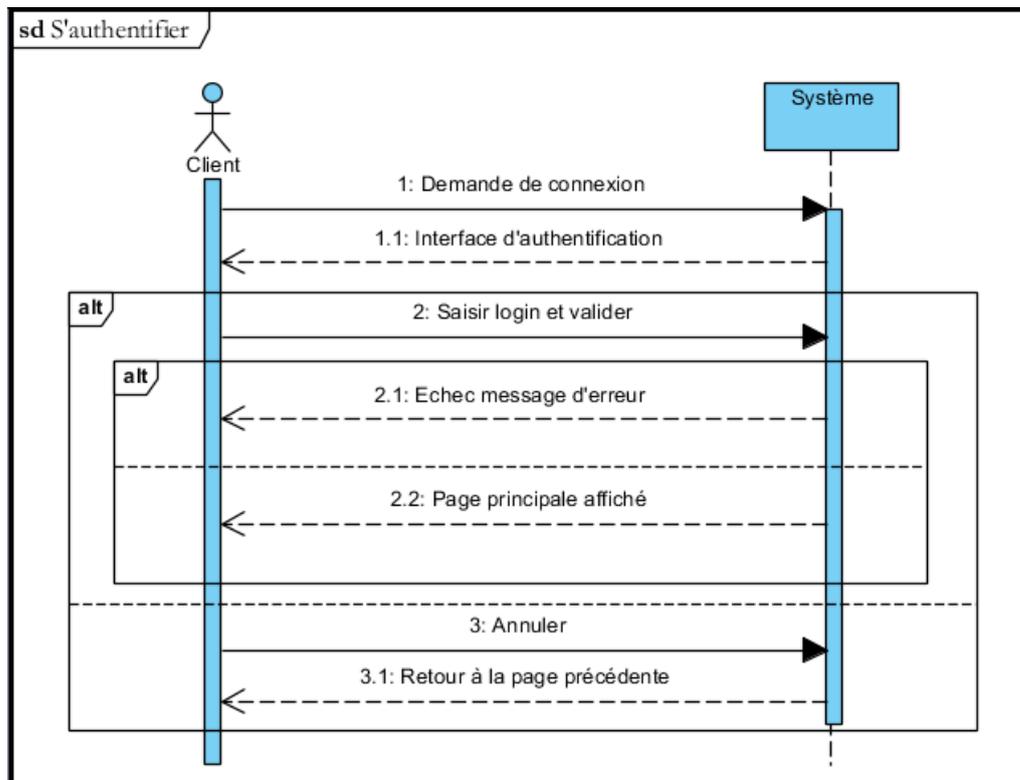


FIGURE 3.2 – Diagramme de séquence système « S'authentifier »

L'utilisateur demande de connecter à l'application, le système lui affiche une fenêtre où il doit s'authentifier en saisissant son identifiant et mot de passe, puis le système vérifie les informations introduites, si l'une des informations n'est pas valide le système affiche un message d'erreur, sinon il affiche la page principale de l'application.

3. Diagramme de séquence système « Afficher sa Position »

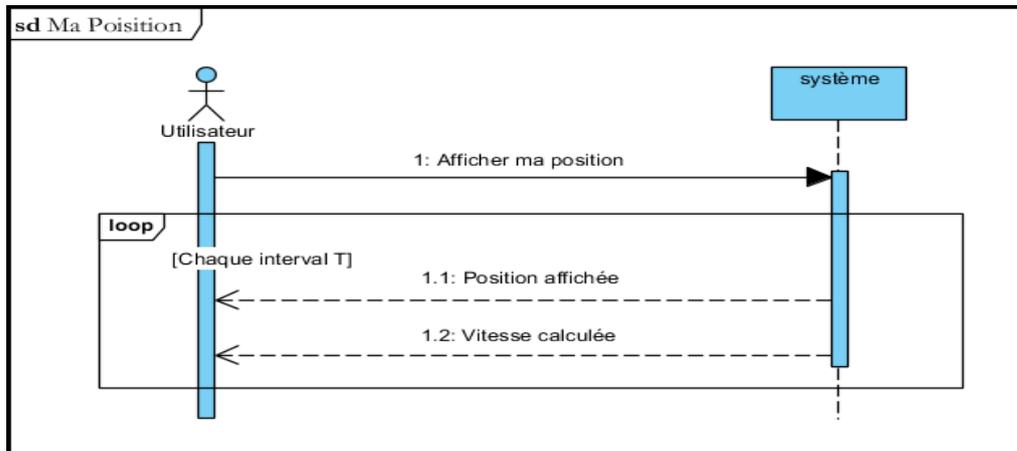


FIGURE 3.3 – Diagramme de séquence système « Afficher sa Position »

L'utilisateur demande au système d'afficher sa position, le système lui affiche une carte géographique sur laquelle il peut voir en temps réel sa position et sa vitesse.

4. Diagramme de séquence système « Gérer son Compte »

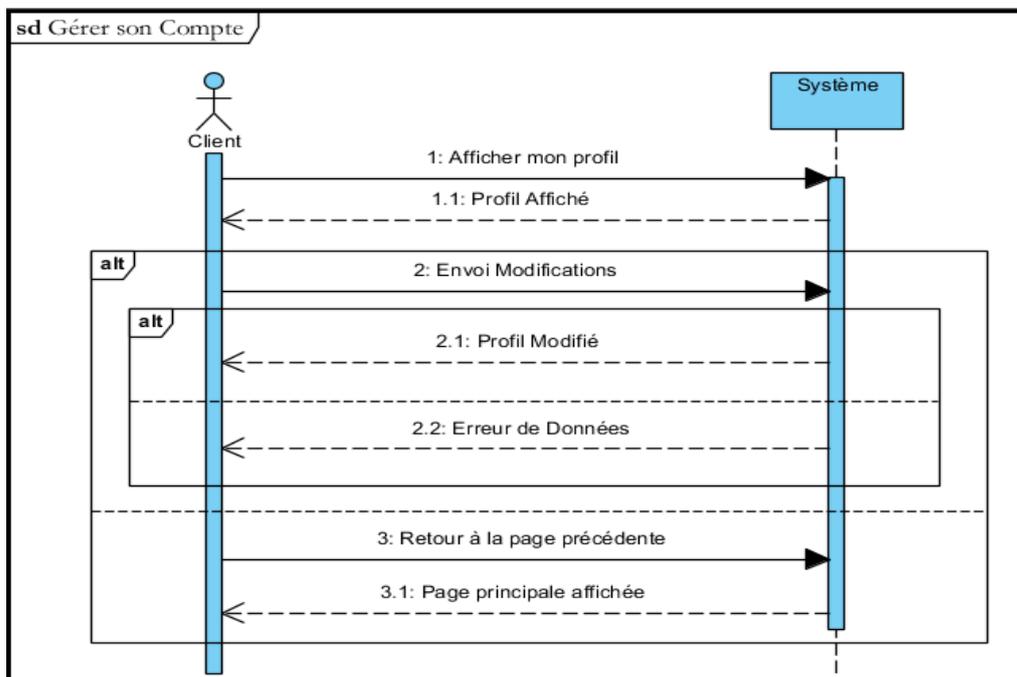


FIGURE 3.4 – Diagramme de séquence système « Gérer son Compte »

Pour que le client puisse afficher les données de son compte, il doit d'abord s'authentifier, ensuite il demande de voir son profil où il peut mettre à jour ses données, alors le système lui transmette une confirmation si les données sont valide, sinon un message d'erreur est affiché. Il peut également annuler les modifications, donc le système lui affiche la page précédente.

5. Diagramme de séquence système « Envoyer un Message »

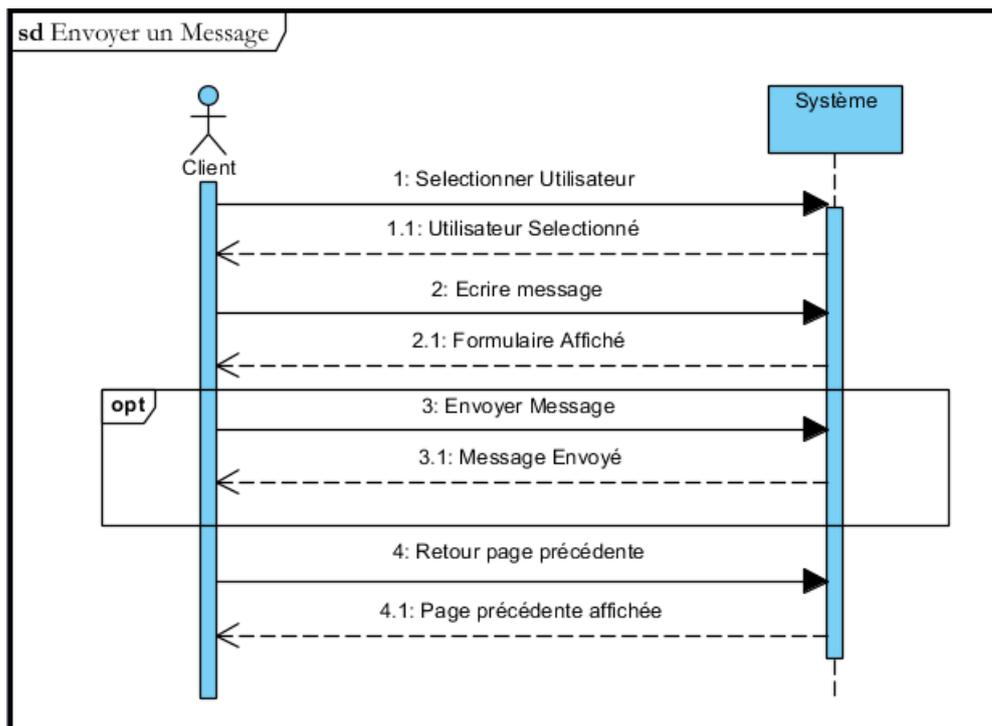


FIGURE 3.5 – Diagramme de séquence système « Envoyer un Message »

Pour que le client puisse envoyer un message, il doit d'abord s'authentifier, il rédige son message et l'envoi, le système lui affiche un message de confirmation. Un retour vers la page précédente lui permet de choisir d'autres fonctionnalités.

6. Diagramme de séquence système « Consulter un Message »

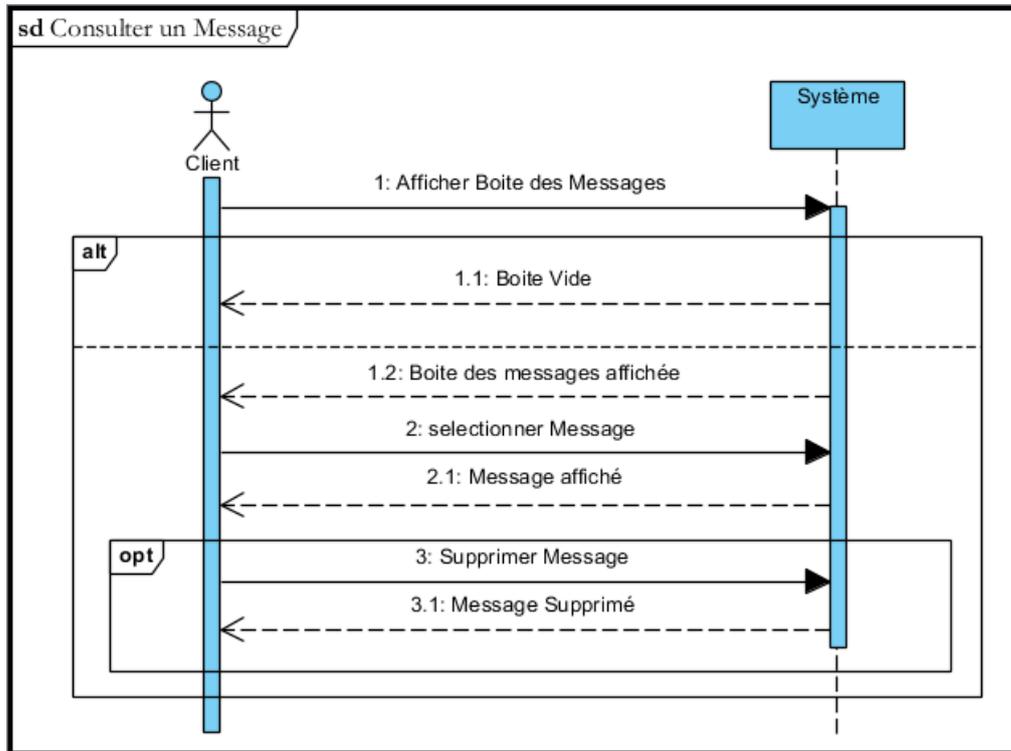


FIGURE 3.6 – Diagramme de séquence système « Consulter un Message »

Pour que le client puisse consulter le contenu d'un message reçu, il doit d'abord s'authentifier, il demande d'ouvrir la boîte des messages qui peut être vide, sinon une liste des messages est affichée. Il peut sélectionner un message pour l'ouvrir, il peut également choisir par la suite de le supprimer.

7. Diagramme de séquence système « Supprimer un Message »

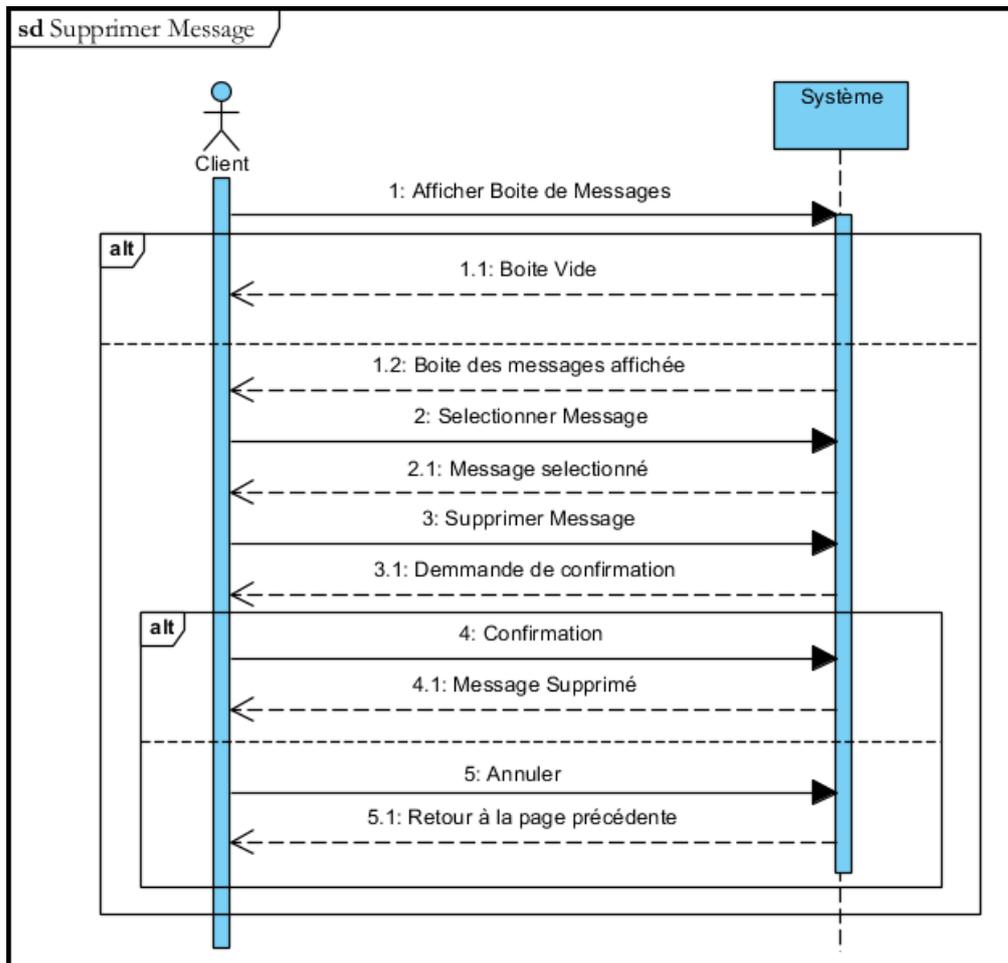


FIGURE 3.7 – Diagramme de séquence système « Supprimer un Message »

Pour que le client puisse supprimer un message reçu, il doit d'abord s'authentifier. Puis, il demande d'ouvrir la boîte des messages, et sélectionne un message pour le supprimer, le système demande une confirmation, si le client répond par oui, le système affiche un message de suppression, si le client annule l'opération, le système retourne à la page précédente.

Diagrammes de séquence détaillé

En remplaçant le système vu comme une boîte noire par un ensemble choisi d'objets de conception, nous décrivons l'attribution des responsabilités dynamiques, tout en conservant une traçabilité forte avec les cas d'utilisation.[17]

Nous allons présenter les diagrammes de séquences détaillés des cas d'utilisation que nous avons déjà cité dans le chapitre précédent.

1. Diagramme de séquence détaillé « S'enregistrer »

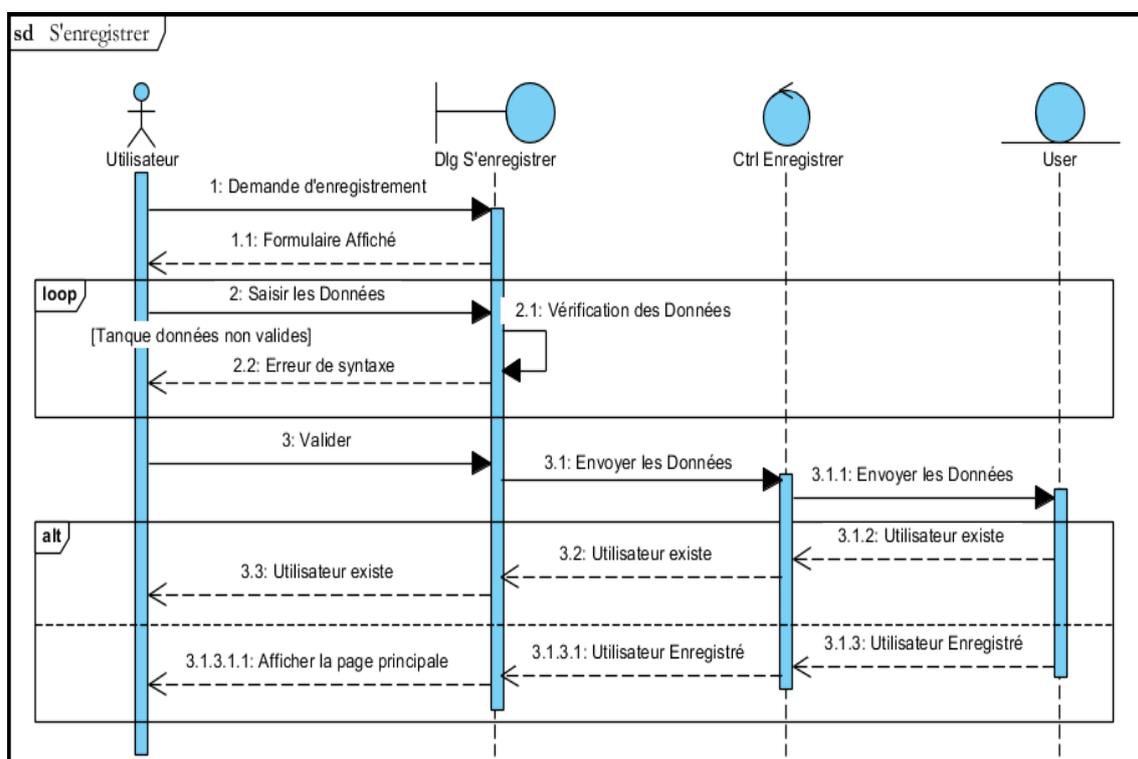


FIGURE 3.8 – Diagramme de séquence détaillé « S'enregistrer »

2. Diagramme de séquence détaillé « S'authentifier »

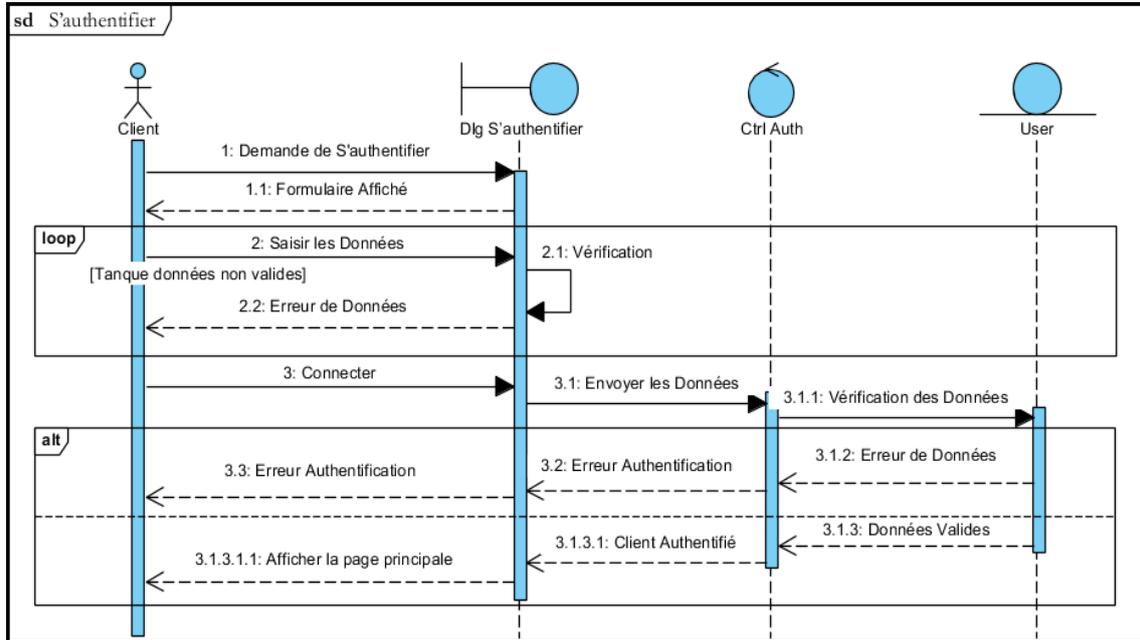


FIGURE 3.9 – Diagramme de séquence détaillé « S'authentifier »

3. Diagramme de séquence détaillé « Créer un groupe »

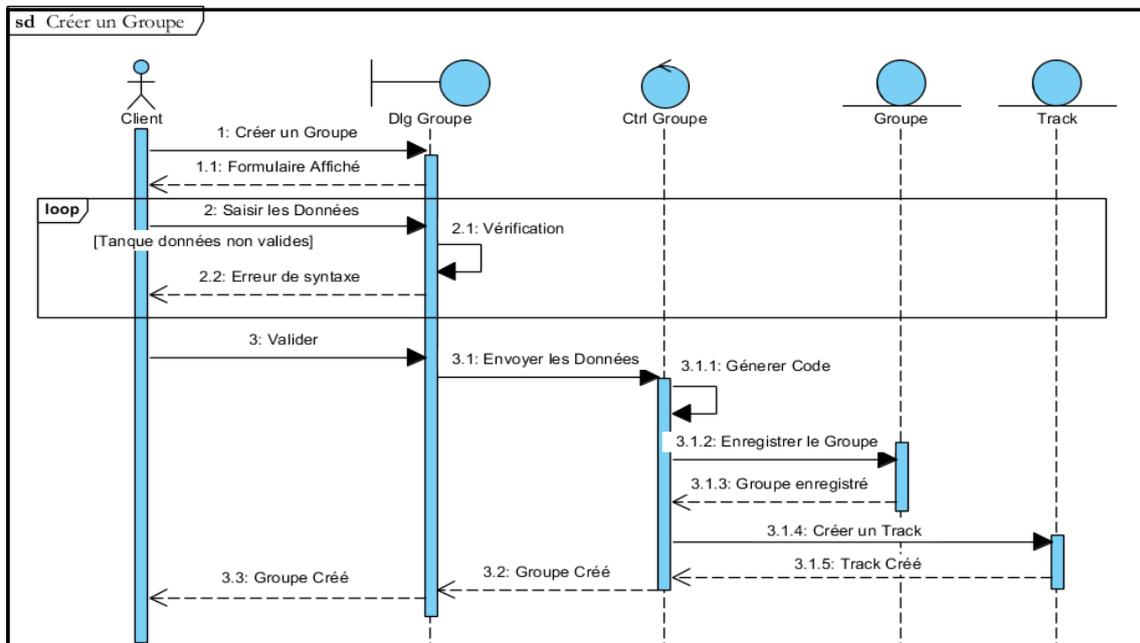


FIGURE 3.10 – Diagramme de séquence détaillé « Créer un groupe »

4. Diagramme de séquence détaillé « Envoyer le Code »

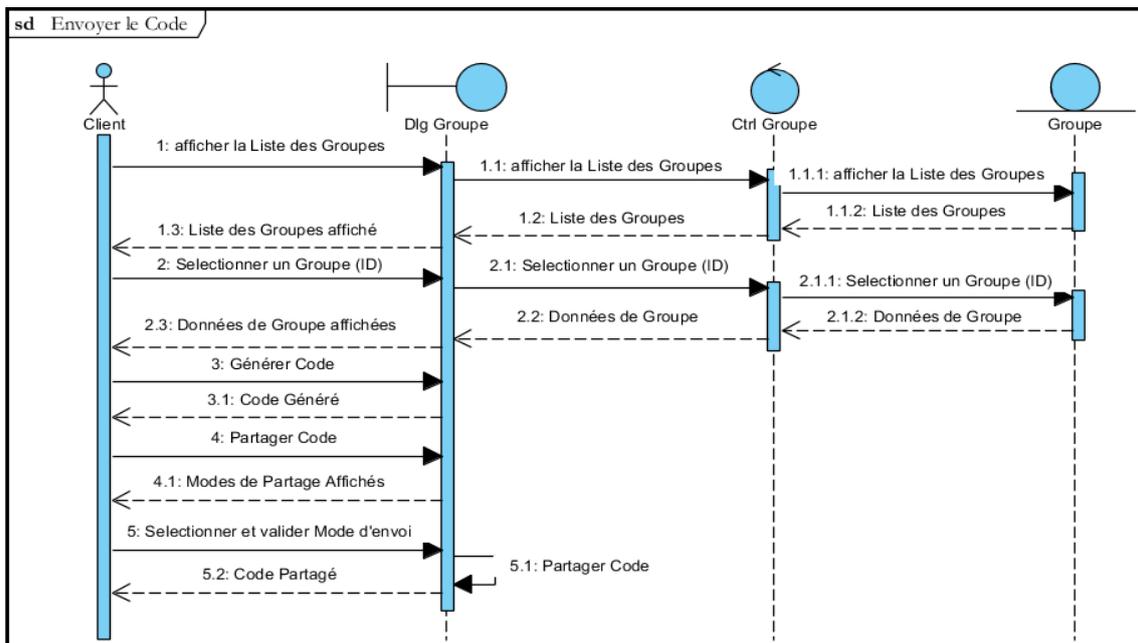


FIGURE 3.11 – Diagramme de séquence détaillé « Envoyer le Code »

5. Diagramme de séquence détaillé « Modifier un groupe »

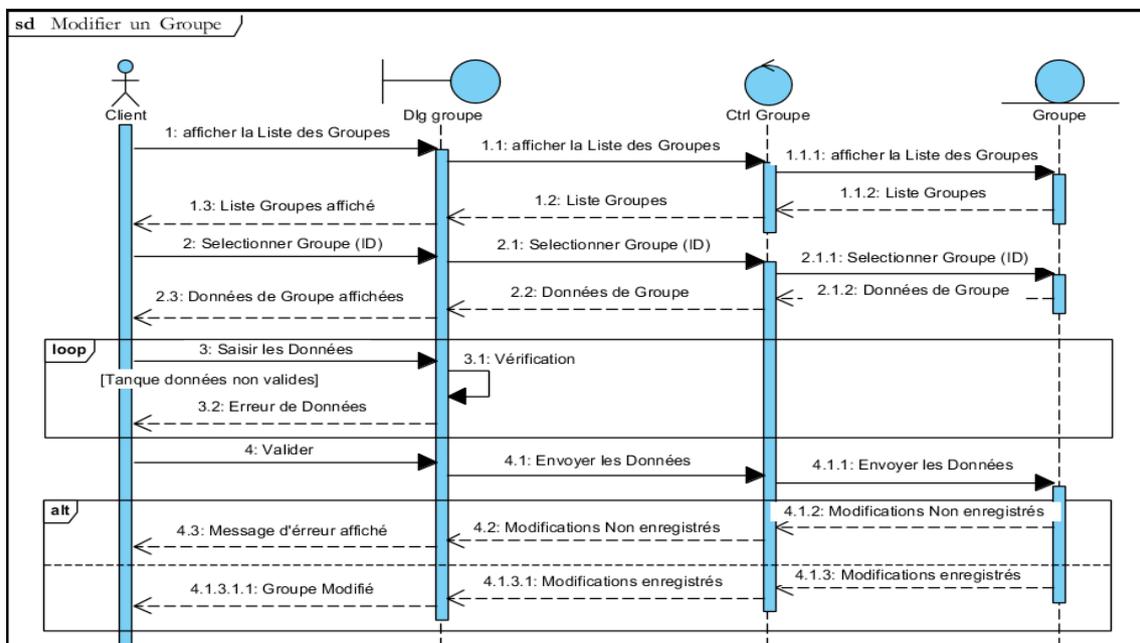


FIGURE 3.12 – Diagramme de séquence détaillé « Modifier un groupe »

5. Diagramme de séquence détaillé « Rejoindre un Groupe »

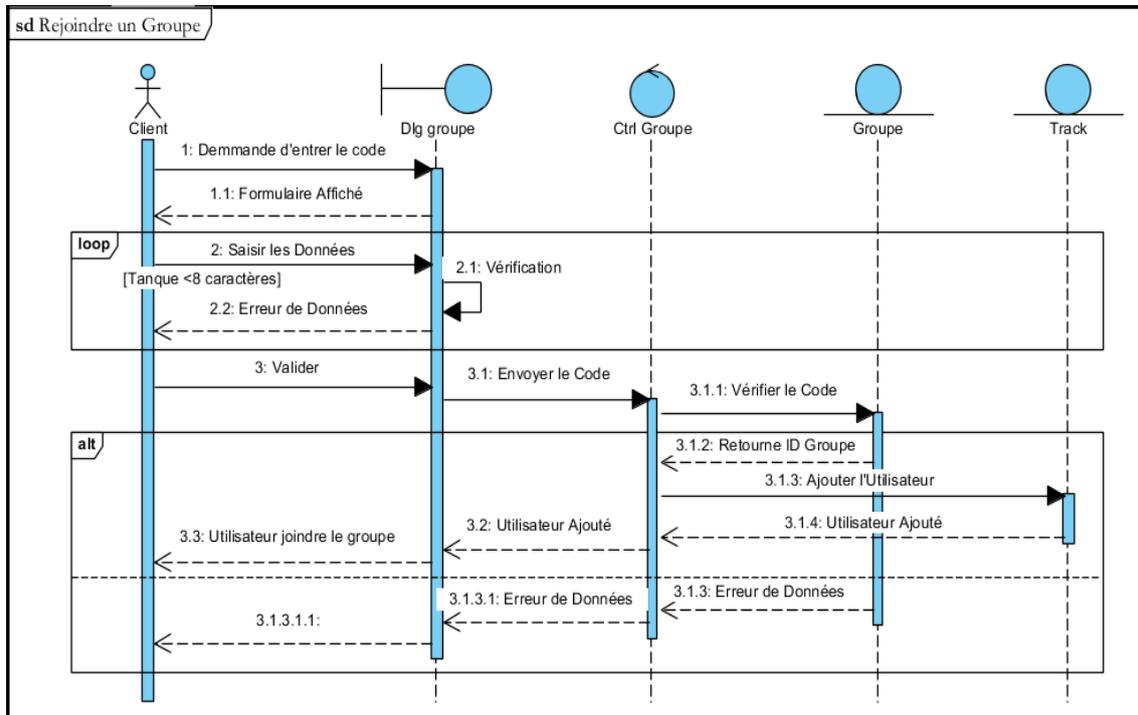


FIGURE 3.13 – Diagramme de séquence détaillé « Rejoindre un Groupe »

6. Diagramme de séquence détaillé « Envoyer un Message »

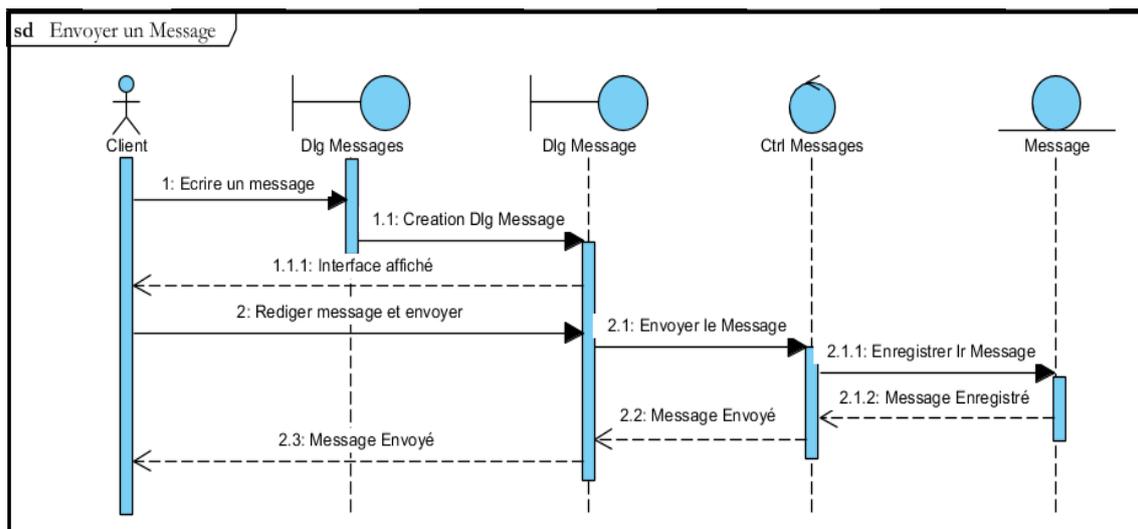


FIGURE 3.14 – Diagramme de séquence détaillé « Envoyer un Message »

7. Diagramme de séquence détaillé « Consulter un Message »

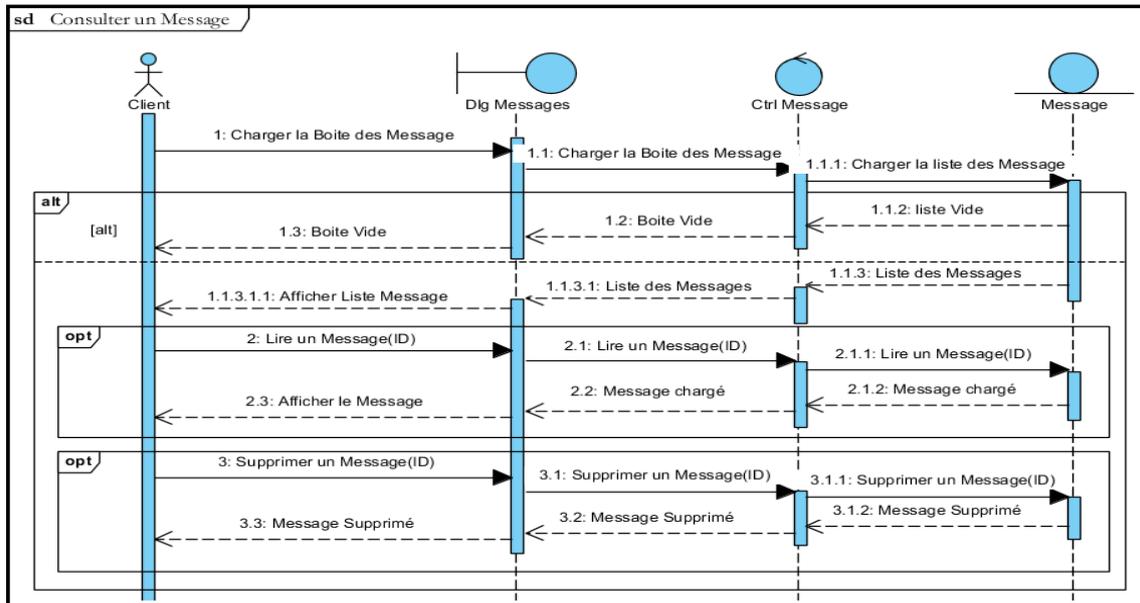


FIGURE 3.15 – Diagramme de séquence détaillé « Consulter un Message »

8. Diagramme de séquence détaillé « Créer une Zone »

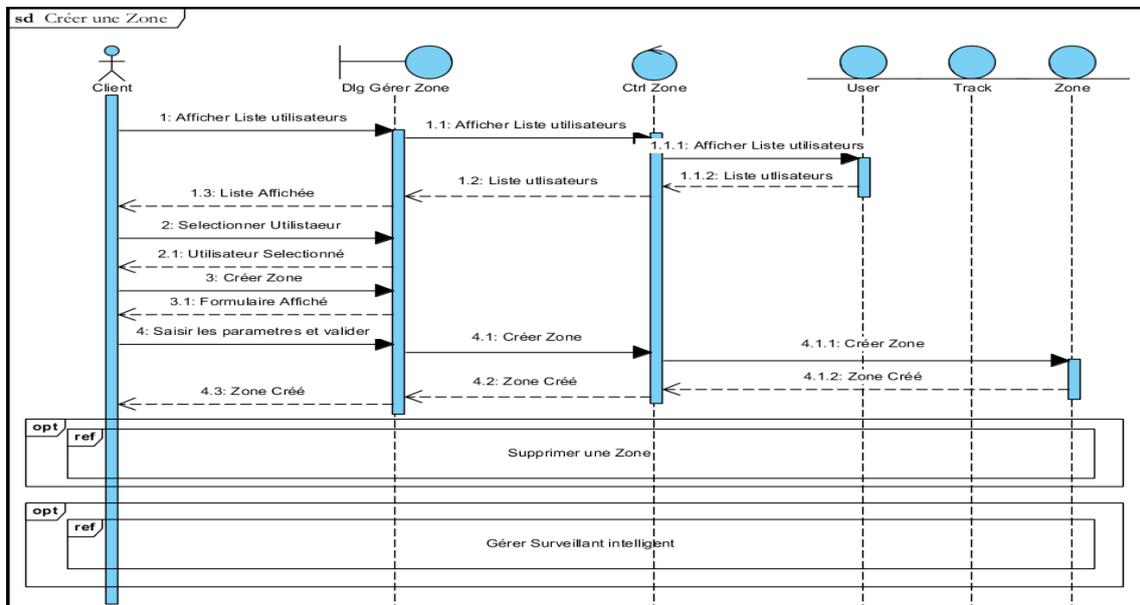


FIGURE 3.16 – Diagramme de séquence détaillé « Créer une Zone »

9. Diagramme de séquence détaillé « Supprimer une Zone »

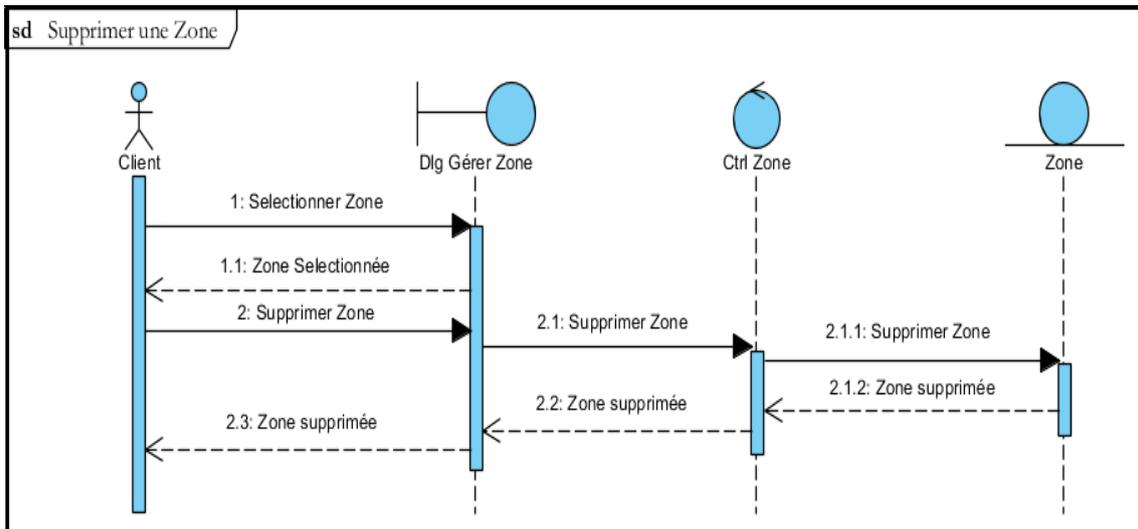


FIGURE 3.17 – Diagramme de séquence détaillé « Supprimer une Zone »

10. Diagramme de séquence détaillé « Gérer le Surveillant intelligent »

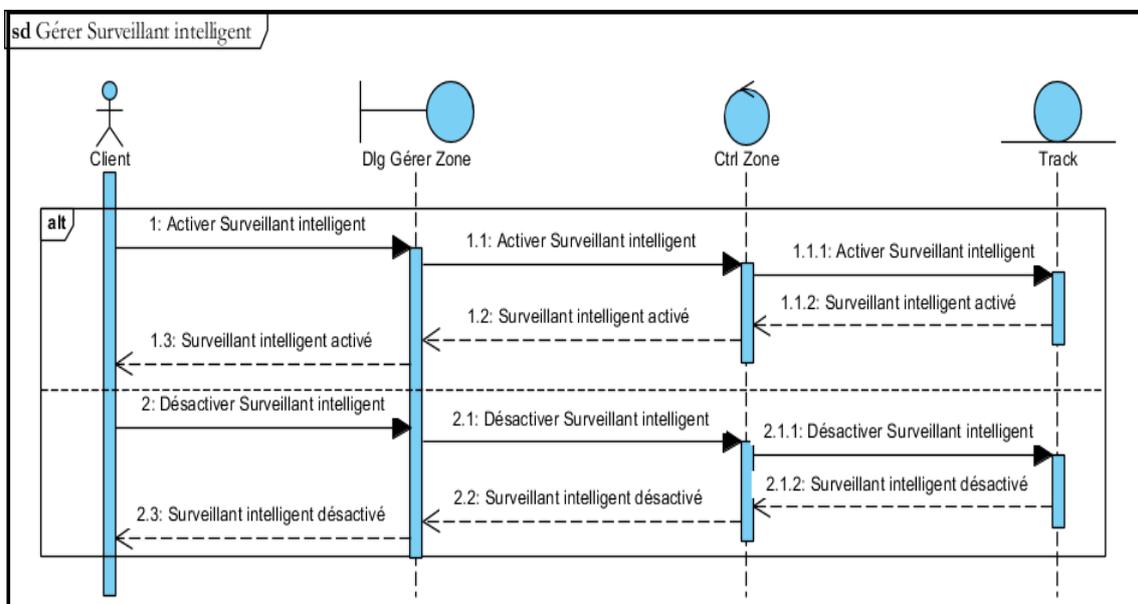


FIGURE 3.18 – Diagramme de séquence détaillé « Gérer le Surveillant intelligent »

11. Diagramme de séquence détaillé « Suivre les Utilisateurs »

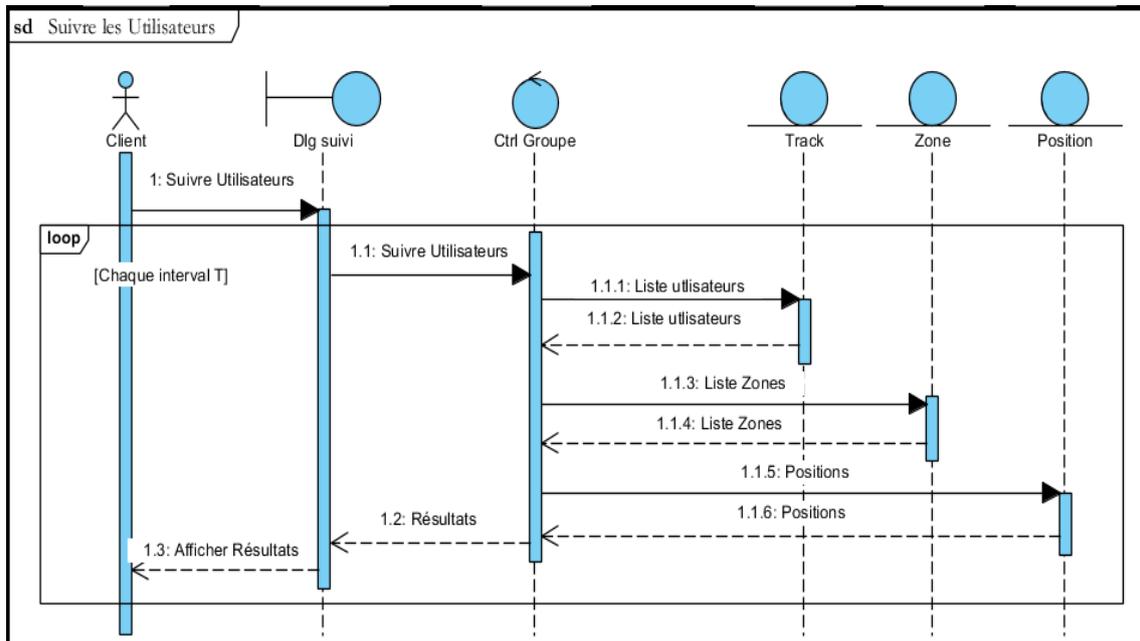


FIGURE 3.19 – Diagramme de séquence détaillé « Suivre les Utilisateurs »

12. Diagramme de séquence détaillé « Envoyer une Alerte »

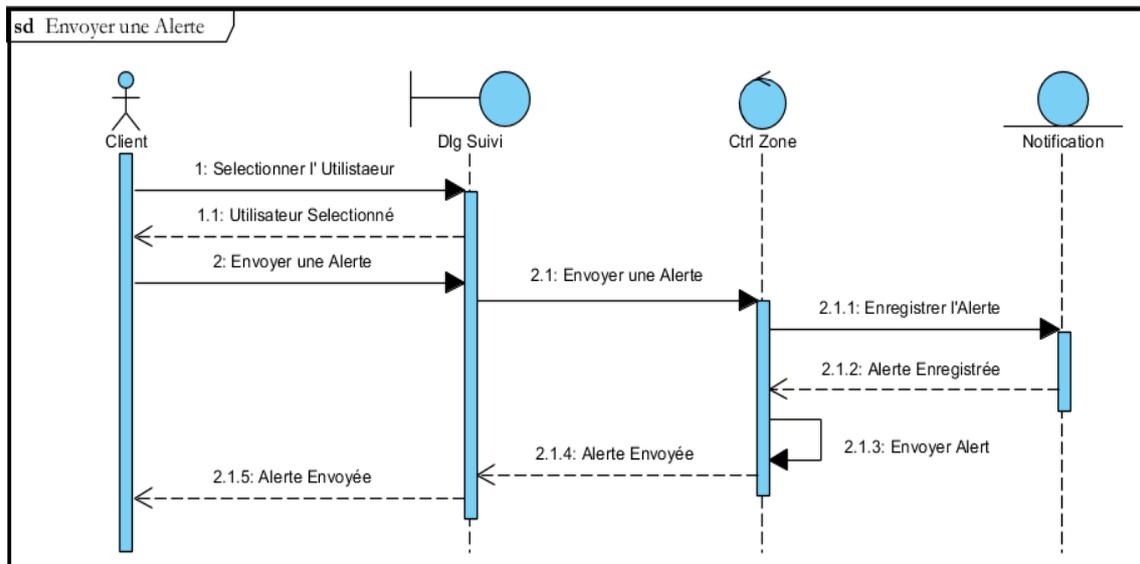


FIGURE 3.20 – Diagramme de séquence détaillé « Envoyer une Alerte »

3.1.2 Diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orienté objet. Il s'agit d'une vue statique du fait qu'on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment du langage de programmation. Le diagramme de classes retenu à la fin de la conception est le suivant:

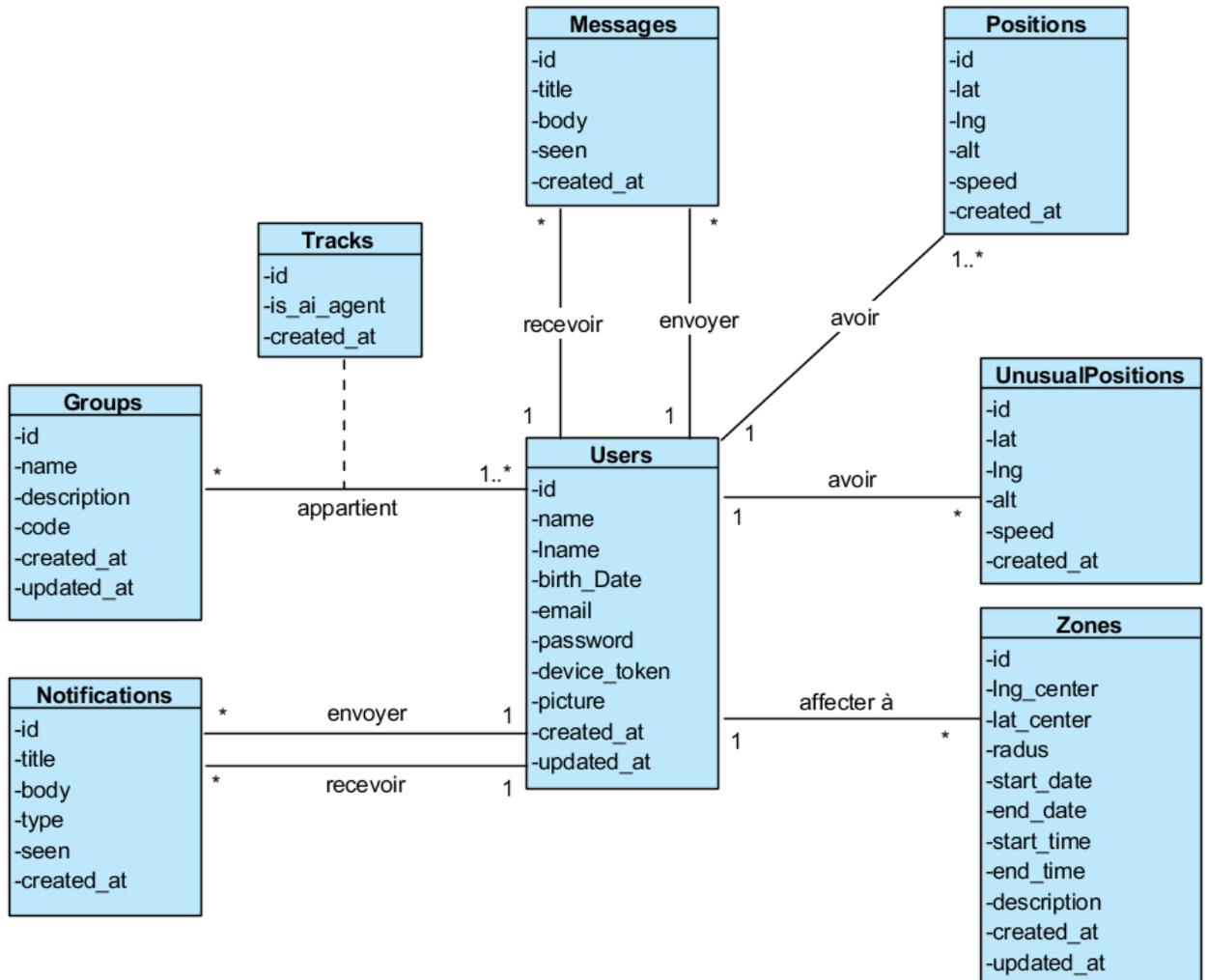


FIGURE 3.21 – Diagramme de classes.

3.1.3 Le modèle relationnel

est le modèle logique de donnée qui correspond à l'organisation des données dans les bases de données relationnelles. Un modèle relationnel est composé de relations, encore appelée table. Ces tables sont décrites par des attributs aux champs. Pour décrire une relation, on indique tout simplement son nom, suivi du nom de ses attributs entre parenthèses. L'identifiant d'une relation est composé d'un ou plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant une clé étrangère, qui correspond à la clé primaire de la relation référencée.[16]

Règles de passage au modèle relationnel

Cette section présente les règles permettant de décrire un schéma logique dans les modèles relationnel et objet-relationnel à partir d'un diagramme de classe UML.

Nous donnons ci-après quatre règles (de R1 à R4) pour traduire un schéma UML en un schéma relationnel équivalent. Il existe d'autres solutions de transformation, mais ces règles sont les plus simples et les plus opérationnelles.[20]

Transformation des classes (R1): Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant. Si aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire (les outils proposent l'ajout de tels attributs).[20]

Transformation des associations: Les règles de transformation que nous allons voir dépendent des cardinalités/multiplicités maximales des associations. Nous distinguons trois familles d'associations.

- **Association un à plusieurs (R2):** Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association. La clé de la relation père migre dans la relation fils.[20]
- **Association plusieurs à plusieurs (R3):** L'association (classe-association) devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectés à l'association. Chaque attribut devient clé étrangère si classe connectée dont il provient devient une relation en vertu de la règle R1. Les attributs de l'association (classe-association) doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.[20]
- **Association un à un (R4):** Il faut ajouter un attribut clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale

à zéro. Dans le cas de UML, il faut ajouter un attribut clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de l'entité (classe) connectée à l'association. Si les deux cardinalités (multiplicités) minimales sont à zéro, le choix est donné entre les deux relations dérivées de la règle R1. Si les deux cardinalités minimales sont à un, il est sans doute préférable de fusionner les deux entités (classes) en une seule.[20]

Schéma relationnel

En appliquant ces règles de transformation d'un diagramme de classe vers un modèle relationnel, nous avons aboutit au schéma relationnel suivant:

```
users (user_id,name,lname,birth_date,email,password,device_token,picture,  
        created_at,updated_at)  
messages (id,title,body,seen,created_at,id_sender,id_receiver)  
groups (groupe_id,name,description,code,created_at,updated_at)  
zones (id,lng_center,lat_center,radius,start_date,end_date,start_time,  
        end_time,description,created_at,updated_at,user_id)  
positions (id,lat,lng,alt,speed,created_at,user_id)  
unusualPositions (id,lat,lng,alt,speed,created_at,user_id)  
notifications (id,title,body,type,seen,created_at,id_sender,id_receiver)  
tracks (id,is_ai_agent,created_at,user_id,groupe_id)
```

3.1.4 Schémas de navigation

1. Application Mobile

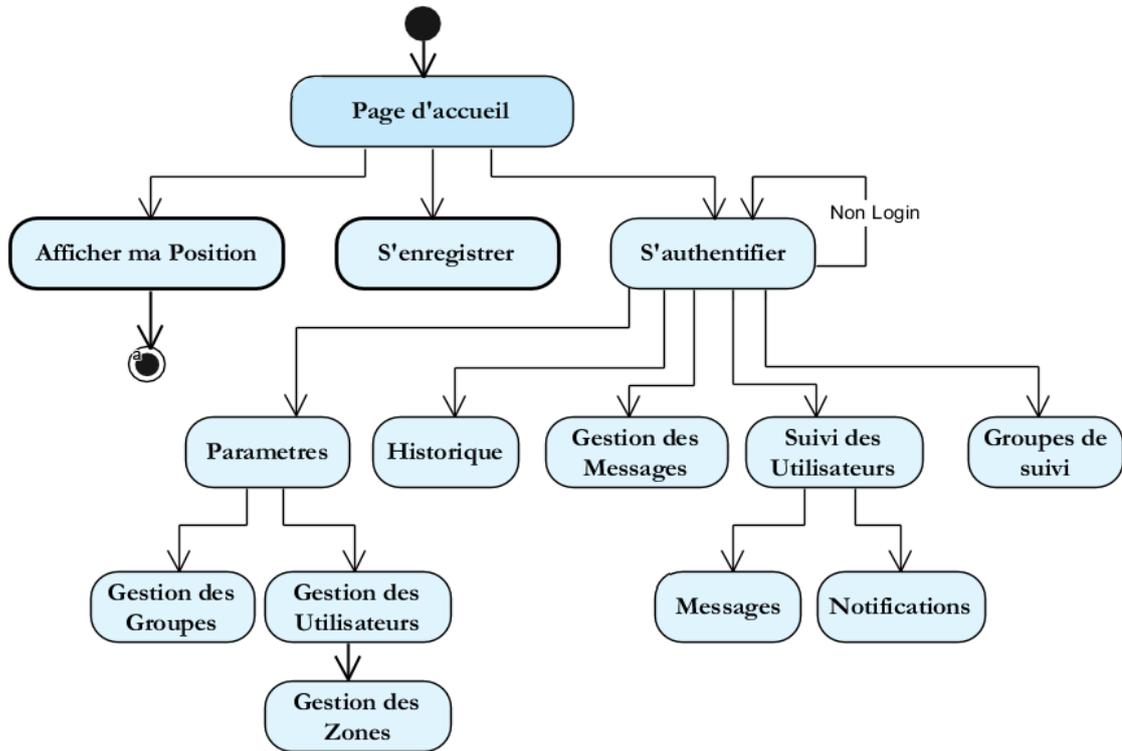


FIGURE 3.22 – Schéma de navigation pour l'application Mobile

2. Application Web

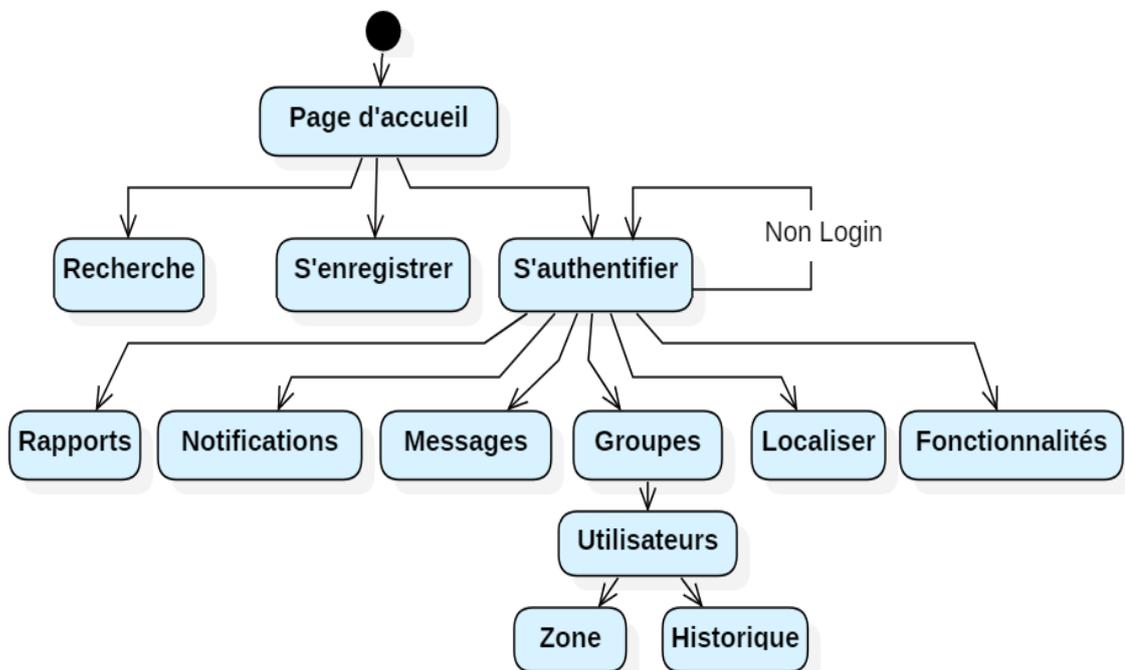


FIGURE 3.23 – Schéma de navigation pour l'application Web

3.2 Conclusion

La phase de conception a été importante pour pouvoir visualiser le fonctionnement de notre application d'une façon abstraite. Nous avons utilisé les diagrammes de séquence système et détaillé, ainsi que le diagramme de classes, et nous avons donné un schéma global de navigation. Dans le chapitre suivant, nous allons passer à la phase de réalisation.

Chapitre 4

Réalisation

4.1 Introduction

Ce chapitre est consacré à la présentation de l'architecture de notre application, ainsi que l'environnement matériel et logiciel utilisés pour le développement, nous expliquerons éventuellement nos choix techniques relatif aux langages de programmation et des outils utilisés. Nous donnons ensuite une description des résultats aboutis approuvés par quelques captures écrans.

4.2 Architecture MVC (Modèle, Vue et Contrôleur)

Nous avons opté pour l'architecture MVC qui est un concept très puissant qui intervient dans la réalisation des applications ou des sites web. Son principal intérêt est la séparation des données (**modèle**), de l'affichage (**vue**) et des actions (**contrôleur**); Il est ainsi facile de mettre à jour le design, sans toucher aux données ni à la façon dont elles sont organisées. Inversement, il est aussi facile de modifier la structure de stockage des données ou la manière dont elles sont gérées sans bouleverser l'affichage.

Le Modèle: représente le comportement de l'application: traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application et définit les méthodes d'accès.

La Vue: correspond à l'interface avec laquelle l'utilisateur interagit. Les résultats renvoyés par le modèle sont dénués de toute présentation mais sont présentés par les vues. Plusieurs vues peuvent afficher les informations d'un même modèle. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle, et de permettre à l'utilisateur d'interagir avec elles.

Le Contrôleur: prend en charge la gestion des évènements de synchronisation pour mettre à jour la vue ou le modèle.[23]

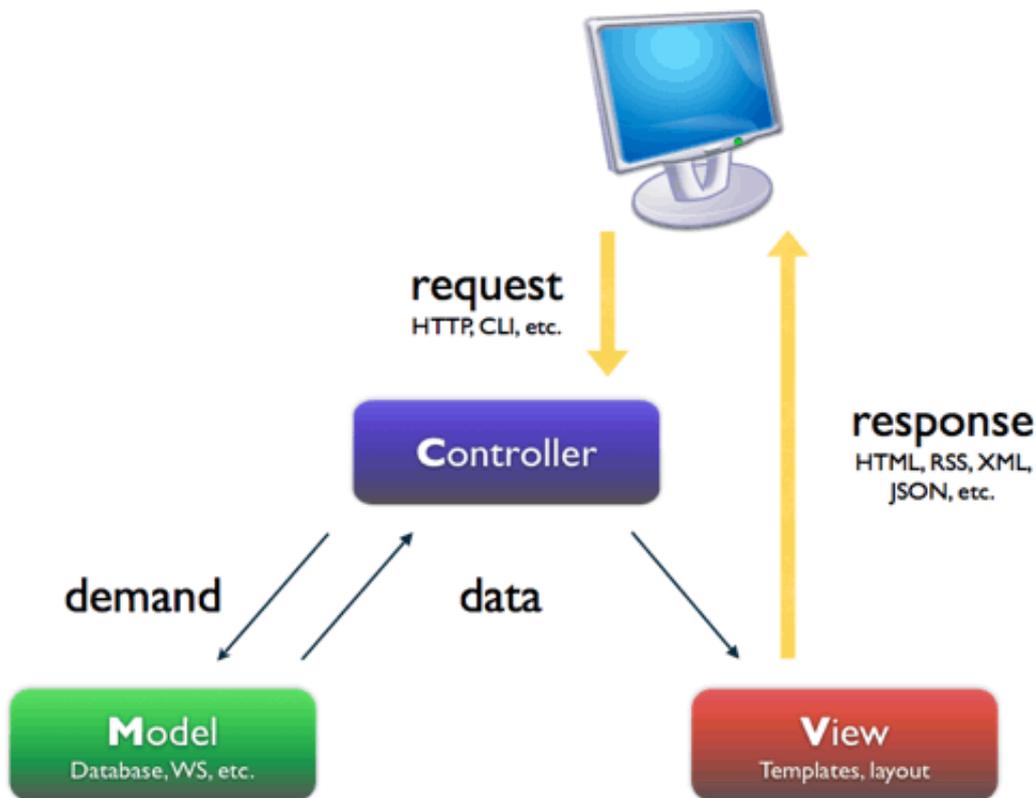


FIGURE 4.1 – Le schéma de l'architecture MVC.

4.3 Environnement du développement de l'application

Dans cette section, nous décrirons l'environnement du travail matériel et logiciel.

4.3.1 Environnement matériel

a. Développement

L'application a été développée sur un ordinateur possédant les caractéristiques suivantes:

- Processeur: intel(R) core(TM) i5 cpu
- Fréquence d'horloge: 2.53 ghz.
- RAM: 6 Go
- Ecran: 16"
- Disque dur: 500 Go
- Système d'exploitation: Windows 7 (64 bit)

b. Exécution

L'application mobile a été exécutée et testée sur:

Smartphone:

- Processeur: Kirin 655(octa-core 2.1 Ghz)
- RAM: 3 Go
- Ecran: 5.2"
- Disque dur: 16 Go
- Système d'exploitation: Android 7.0 Nougat

Tablette:

- Processeur: Mediatek MT8735(quad-core 1.3 Ghz)
- RAM: 1 Go
- Ecran: 7"
- Disque dur: 8 Go
- Système d'exploitation: Android 6.0 Marshmallow

L'application web a été exécutée et testée sur un ordinateur possédant les caractéristiques suivantes:

- Processeur: intel(R) core(TM) i5 cpu
- Fréquence d'horloge: 2.53 ghz.
- RAM: 6 Go
- Ecran: 16"
- Disque dur: 500 Go
- Système d'exploitation: Windows 7 (64 bit)
- Navigateur: Google chrome Version 74.0.3729.131 (64 bits)

4.3.2 Environnement logiciel

Pour le développement de l'application ANDROID nous avons utilisé les éléments suivants:

1. Android Studio



FIGURE 4.2 – Android Studio.

C'est l'IDE (environnement de développement intégré) officiel pour le développement d'applications Android par Google. Il est basé sur le logiciel IntelliJ IDEA de JetBrains et possède de nombreuses fonctionnalités étonnantes qui aident les développeurs à créer une application Android. Android Studio est disponible en téléchargement gratuit sur Windows, Mac OS X et Linux.

2. Java Development Kit

Le JDK est un environnement de développement permettant de créer des applications et des composants utilisant le langage de programmation Java.

Le JDK comprend des outils utiles pour développer et tester des programmes écrits dans le langage de programmation Java et s'exécutant sur la plate-forme Java.^[15]

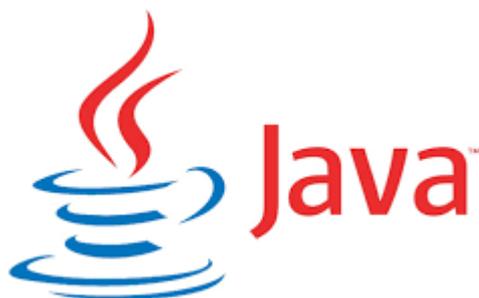


FIGURE 4.3 – Java Development Kit.

4.3.3 Outils pour le développement web

a. Laravel



FIGURE 4.4 – framework laravel.png.

C'est un framework PHP open-source, robuste et facile à comprendre. Il suit un modèle de conception modèle-vue-contrôleur.[22]

Laravel nous offre les avantages suivants:

Architecture MVC: Respecte le principe Model View Controller(MVC) et entièrement développé en programmation orienté objet.

- Le Model contient les données à afficher
- La vue (View) contient la présentation de l'interface graphique
- Enfin le Controller contient la logique concernant les actions effectués par l'utilisateur.

Eloquent: un moyen extrêmement facile de communiquer avec une base de données. Chaque table de base de données a un "Modèle" correspondant qui est utilisé pour interagir avec cette table.

Bootstrap: La création de la mise en page est une partie importante de tout projet, tandis qu'elle prend beaucoup de temps avec le CSS, pour cela Bootstrap est fait pour construire rapidement et facilement des sites web esthétiques et responsives.

Composer: est un outil écrit en PHP qui permet de gérer les dépendances d'un projet. Ce qu'il faut comprendre par dépendance, c'est toute librairie ou bibliothèque PHP qu'un projet en dépend.

b. WampServer

Le package WAMP est un environnement de développement sous Windows, très utile pour développer un site web dynamique. Cet environnement est composé d'Apache, MySQL et PHP(ces éléments composent d'ailleurs l'acronyme WAMP).[11]

Les éléments qui composent WAMP sont les suivants:

Apache: serveur HTTP très populaire

MySQL: système de gestion de base de données



FIGURE 4.5 – Le package WampServer

PHP: langage de développement populaire sur les sites web.

PHPMyAdmin: système de gestion pour MySQL. C'est une interface pour gérer plus facilement MySQL sans nécessairement avoir de nombreuses connaissances.

4.3.4 GoogleMaps

Définitions

Google Maps est un service gratuit de cartographie en ligne basé sur le web qui fournit des informations détaillées sur les régions géographiques et des sites dans le monde entier. En plus des cartes routières classiques, Google Maps propose des vues aériennes et par satellite de nombreux endroits.[18]

Api GoogleMaps

L'Api de Google permettant de géolocaliser des adresses sur une carte à l'aide de ses coordonnées (latitude et longitude); une petite bulle est affichée sur la carte pour indiquer la position du terminal.

Les étapes pour intégrer l'API Google Map à une application Android sont comme suit:

Compte Google developpeur: La première condition pour pouvoir accéder à l'API c'est d'avoir un compte Google.

SDK Google Play: Pour le développement d'applications utilisant ses API, Google recommande d'utiliser l'IDE Android Studio, dans lequel il est nécessaire d'installer et de configurer le SDK des services GooglePlay

Création d'un projet: Dé que tous les packages nécessaires ont été installés on peut créer une application dont une activité sera une GoogleMap Activity.

Obtenir une clé d'API Google Maps: l'application a besoin d'une clé API pour accéder aux serveurs Google Maps. Le type de clé dont on a besoin est une clé API avec restriction pour les applications Android. La clé est

gratuite. On peut l'utiliser avec n'importe quelle applications appelant le SDK Maps pour Android. Cette application prend en charge un nombre illimité d'utilisateurs.

4.3.5 PHP Machine learning

L'intelligence artificielle et l'apprentissage automatique sont un sujet d'actualité pour la plupart des développeurs, mais pas un sujet très commun pour les développeurs Web. PHP-ML est une bibliothèque d'apprentissage automatique disponible pour PHP qui fournit des algorithmes, une validation croisée, un réseau de neurones, un prétraitement, une extraction de fonctionnalités et bien plus dans une seule bibliothèque.

On a choisi cette bibliothèque pour sa simplicité et sa facilité d'intégration, elle fournit des interfaces simples pour la plupart des algorithmes les plus utilisés dans le domaine de l'apprentissage automatique. Le but d'intégrer l'intelligence artificielle dans notre projet et de fournir une surveillance automatique des objets suivis pour réduire l'intervention des agents humains en classifiant les emplacements des objets comme « permis » ou « non permis », pour notre projet on a choisi l'algorithme de classification « **KNearestNeighbors Classifier** » qui donne de bon résultats.

Exemples simple de classification on utilisant la bibliothèque:

Paramètres du constructeur:

- \$ k - nombre de voisins les plus proches à analyser (par défaut: 3)
 - \$ distanceMetric - Objet distance, euclidien par défaut
- ```
$classifier = new KNearestNeighbors($k=4);
$classifier = new KNearestNeighbors($k=3, new Minkowski($lambda=4));
```

**Entrainer:** Pour former un classificateur, il suffit de fournir des échantillons d'entraînement et des étiquettes (comme array).

Exemple:

- \$samples = [[1, 3], [1, 4], [2, 4], [3, 1], [4, 1], [4, 2]];
  - \$labels = ['a', 'a', 'a', 'b', 'b', 'b'];
- ```
$classifier = new KNearestNeighbors();
$classifier->train($samples,$labels);
```

Nous pouvons former le classificateur à l'aide de plusieurs ensembles de données. Les prédictions seront basées sur toutes les données d'apprentissage.

Prédire: Pour prédire nous utilisons la méthode « **predict** ». nous pouvons fournir un échantillon ou un tableau d'échantillons:

```
$classifier->predict([3, 2]);
// return 'b'
```

```
$classifier->predict([[3, 2], [1, 5]]);
// return ['b', 'a']
```

Dans notre projet, nous avons utilisé, les attributs liés à la position d'un objet suivi pour implémenter notre algorithme d'apprentissage automatique [id,lat,lng,speed,user_id]

4.4 Implémentation et Test

4.4.1 Implémentation de la base de données

Pour implémenter notre base des données, nous avons utilisé l'environnement de création de base de données « PHPMyAdmin » et le système de gestion de base de données « MySQL ». La figure ci-dessous présente les différentes tables de la base de données utilisée pour notre application.

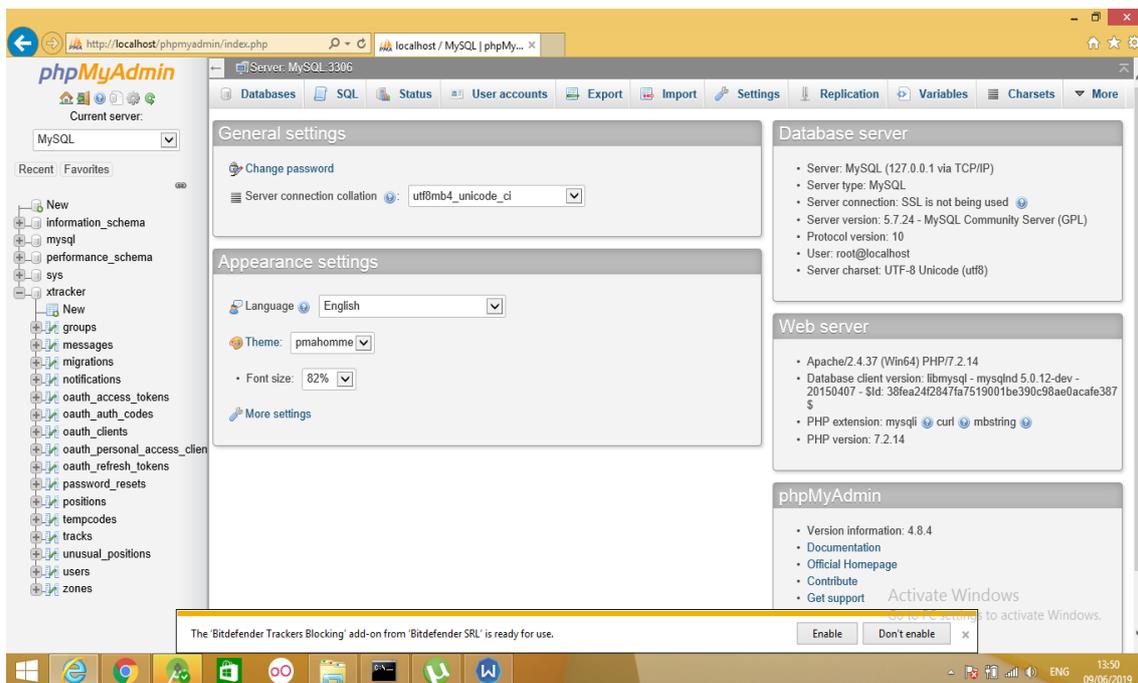


FIGURE 4.6 – Les tables de la base de données.

4.4.2 Les Scénarios d'exécution

1.L'application mobile

Nous présentons notre application mobile à travers des captures écrans selon chaque cas d'utilisation comme suit:

— La figure suivante illustre la page principale de l'application.

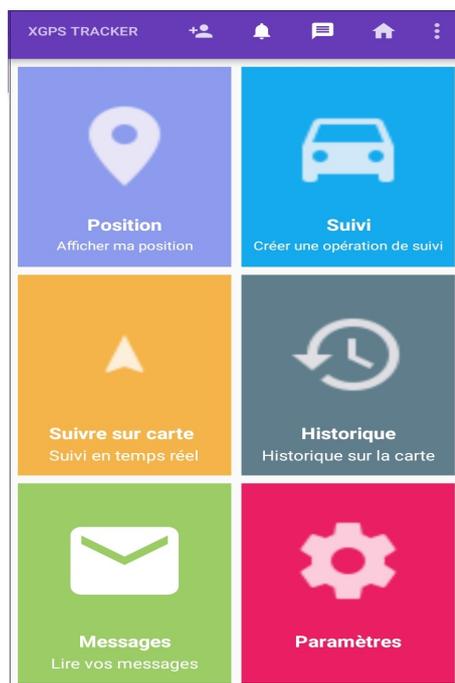


FIGURE 4.7 – Interface de la page principale.

— La figure à gauche illustre la fenêtre « S'enregistrer » et la figure à droite représente et la fenêtre « S'authentifier ».

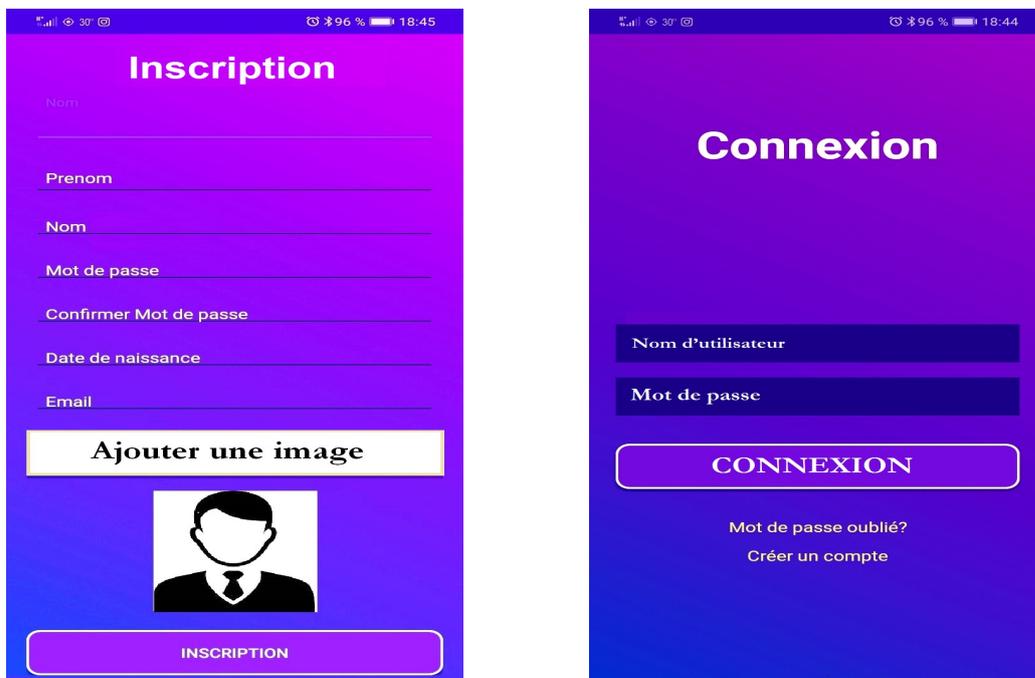


FIGURE 4.8 – S'enregistrer et S'authentifier.

- La figure à gauche illustre la page pour « Créer un Groupe » et la figure à droite représente la liste des utilisateurs.

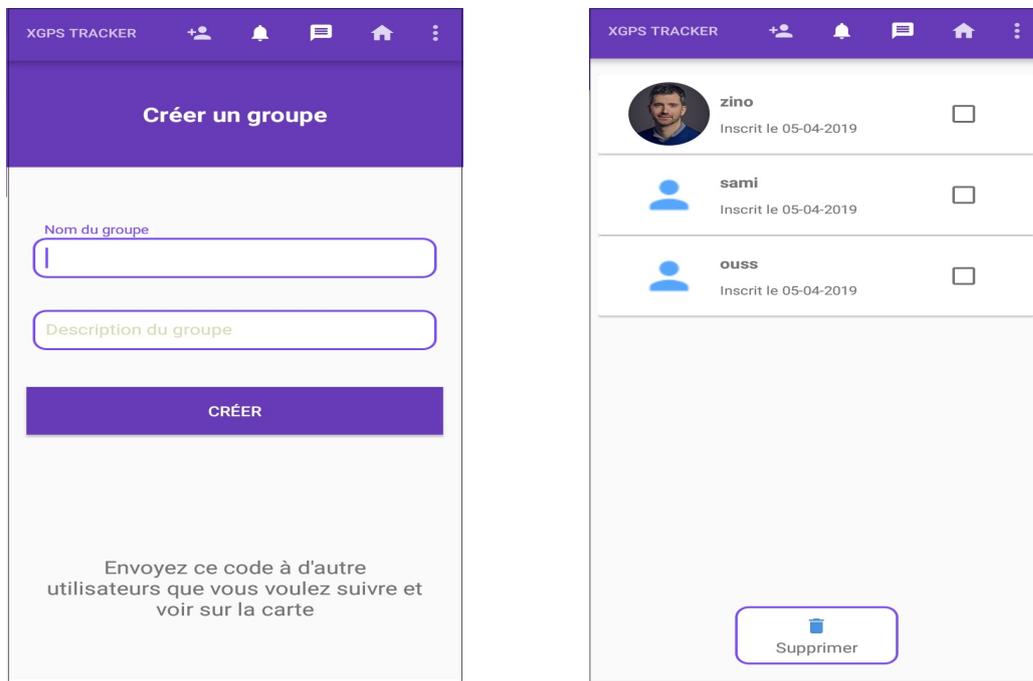


FIGURE 4.9 – Créer un groupe et la liste des utilisateurs.

- La figure à gauche illustre la boîte des messages et la figure à droite représente la lecture d’un message choisi.

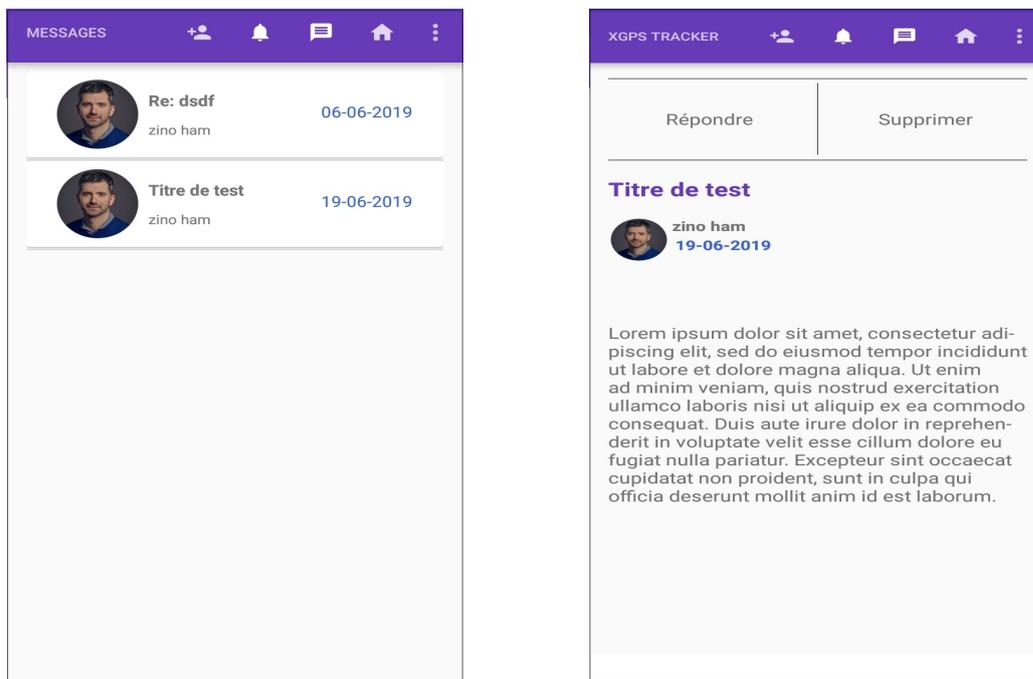


FIGURE 4.10 – Boîte des Messages et Lire un message.

— La figure illustre la page « Afficher sa position ».

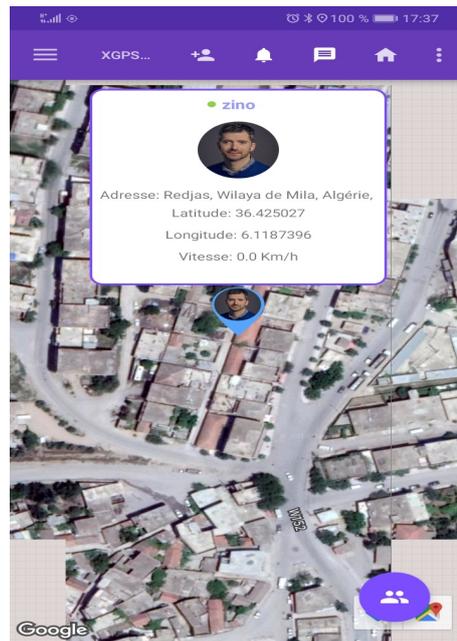


FIGURE 4.11 – Interface « Afficher sa position ».

2.L'application Web

Nous présentons notre application web à travers des captures écrans selon chaque cas d'utilisation comme suit:

— La figure suivante illustre la page qui affiche la liste des groupes de suivi.

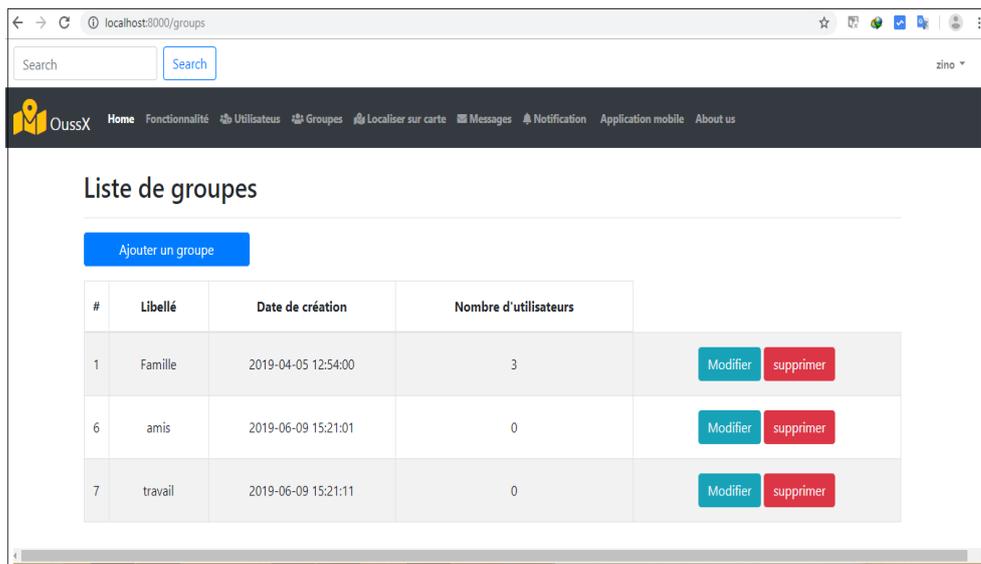


FIGURE 4.12 – Interface liste des groupes.

— La figure suivante illustre la page qui affiche la liste des utilisateurs par groupe.

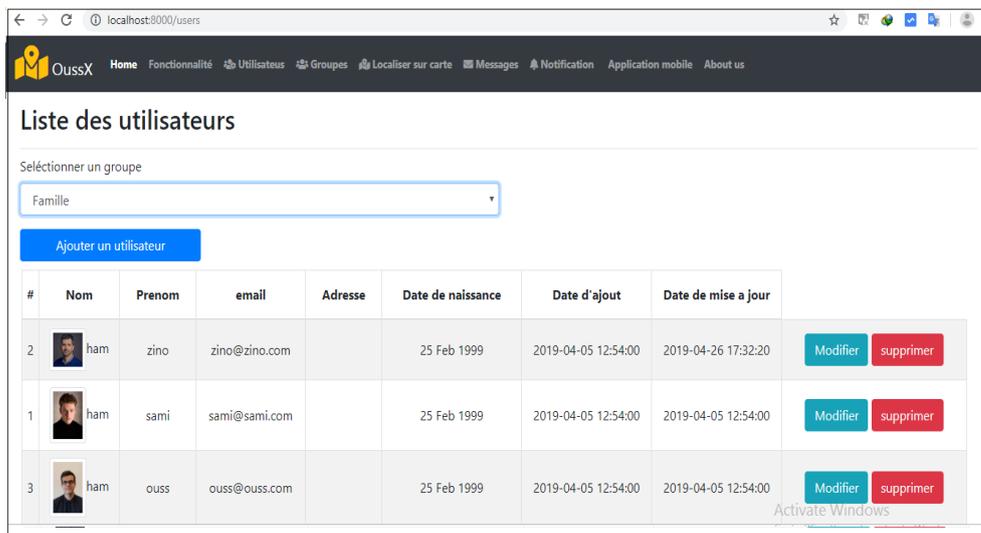


FIGURE 4.13 – Interface liste des utilisateurs.

— La figure suivante illustre la page qui affiche la liste des messages.

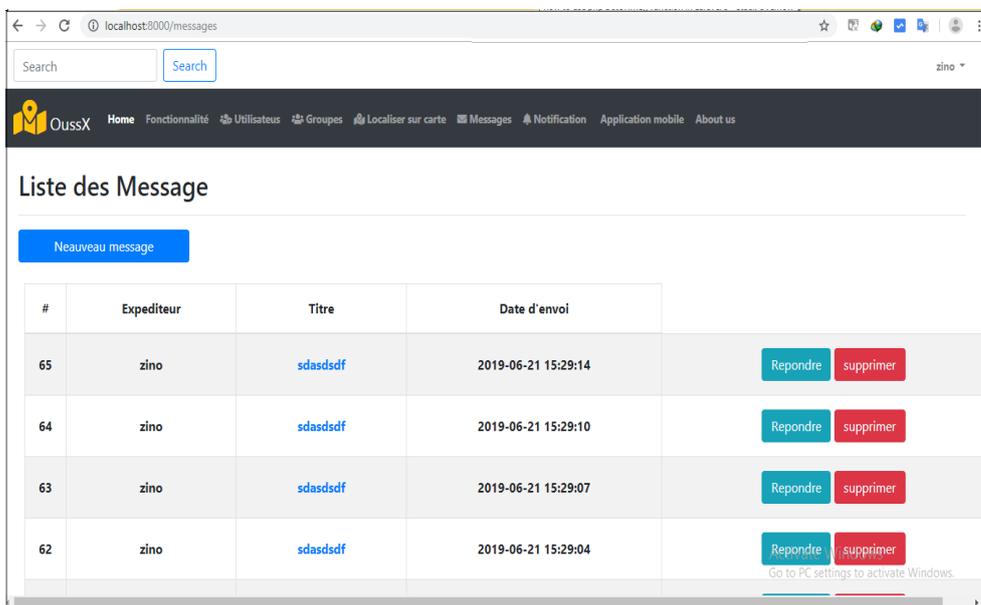


FIGURE 4.14 – Interface liste des messages.

— La figure suivante illustre la page « afficher sa position ».

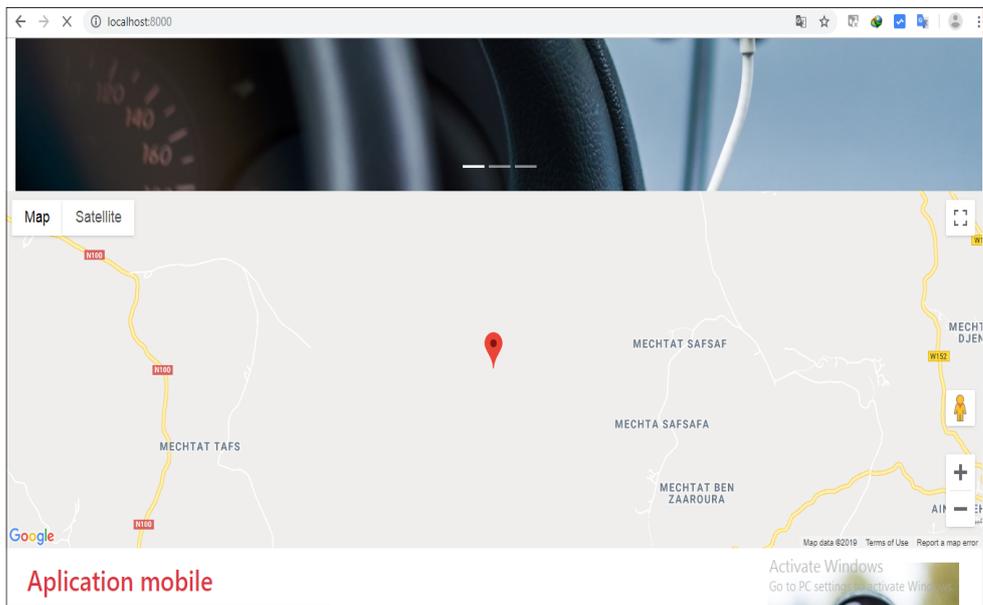


FIGURE 4.15 – Interface « afficher sa position ».

4.5 Conclusion

Au cours de ce chapitre, nous avons présenté les différentes phases de la réalisation. Nous avons commencé par recenser les différents technologies utilisés ainsi que l'environnement matériel et logiciel de notre travail, ensuite on a présenté l'implémentation de notre base de données, et nous avons terminé par la présentation de notre application (Mobile et Web) via quelques captures d'écrans.

Conclusion générale et Perspectives

Dans ce mémoire, nous avons réalisé une solution informatique permettant la géolocalisation et le suivi des objets mobiles en temps réel. Notre solution facilite grandement les opérations de contrôle et suivi dans les entreprises, tel que le suivi des véhicules et ressources humaines. Ceci augmente considérablement les performances de l'entreprise et permet une meilleure flexibilité. Cette solution peut aussi être exploitée par les individus pour, par exemple, suivre le déplacement de ses proches. En effet connaître l'emplacement en temps réel de ses enfants permet d'éviter les risques d'agressions et de kidnapping.

Notre solution est composée d'une partie web (pour PC), et une partie mobile (pour Smartphones et tablettes). En effet elle est très facile à utiliser et riche en fonctionnalités, telle que les notifications automatiques par zone, l'échange des messages, et surtout elle intègre un agent intelligent qui permet après une phase d'apprentissage de réduire considérablement l'intervention humaine.

Néanmoins, et comme tous les ouvrages humains, notre projet peut être amélioré par l'ajout d'autres fonctionnalités telles que:

1. L'utilisation des différents types de capteurs pour collecter plus de données sur l'objet suivi comme la température, le niveau de réservoir et l'état de l'objet...etc
2. Implémenter d'autres algorithmes d'apprentissage comme les réseaux de neurones pour améliorer les résultats.
3. Développer et exécuter l'application sur d'autres plateformes tel qu'iOS.
4. Supporter plus des équipements GPS autres que les smartphones et smart watch.

Annexe A

Le GPS

A.1 définition

Le système GPS (Global Positioning System) est mis en place à l'origine par le Département de la Défense des États-Unis. Il comporte une constellation de 24 satellites (plus 4 en réserve) et est opérationnel depuis 1995. La dégradation des signaux pour les usages non militaires a été levée le 1er mai 2000. Sa précision est dépendante du nombre de satellites en vue (3 au minimum) par le récepteur GPS, au moment de la géolocalisation. Elle est de 3 à 50 mètres de manière standard (10 mètres en moyenne) [2].

A.2 Composants du GPS

Le système GPS est composé de trois parties distinctes, appelées segments. Il se compose de trois segments qui s'appellent : le segment spatial, le segment de contrôle et le segment utilisateur. Ce sont les satellites en orbite autour de la Terre (segment spatial), les stations de contrôle au sol (segment contrôle), et enfin les récepteurs GPS des utilisateurs (segment utilisateur)[10].

Le segment spatial: Le segment spatial est constitué d'une constellation nominale de 24 satellites opérationnels qui transmettent des signaux unidirectionnels, lesquels donnent le positionnement et l'heure du réseau GPS.

Le segment de contrôle: Le segment de contrôle se compose de stations, réparties à travers le monde, qui sont chargées de piloter et de surveiller le système.

Le segment utilisateur: Le segment utilisateur comprend les récepteurs GPS, qui reçoivent les signaux des satellites et qui calculent, sur la base des informations transmises, la position tridimensionnelle et la référence temporelle

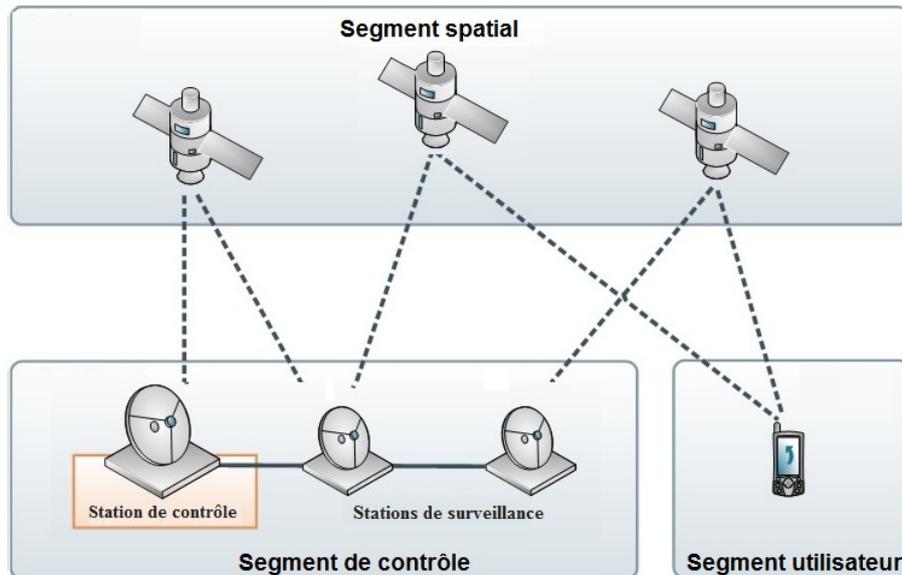


FIGURE A.1 – Composants du GPS.

de l'utilisateur. Parce qu'il est gratuit, ouvert et fiable, le Système de positionnement mondial (GPS, Global Positioning System) a permis aux utilisateurs du monde entier de mettre au point des centaines d'applications qui affectent pratiquement tous les aspects de la vie moderne.

A.3 Les coordonnées géographiques

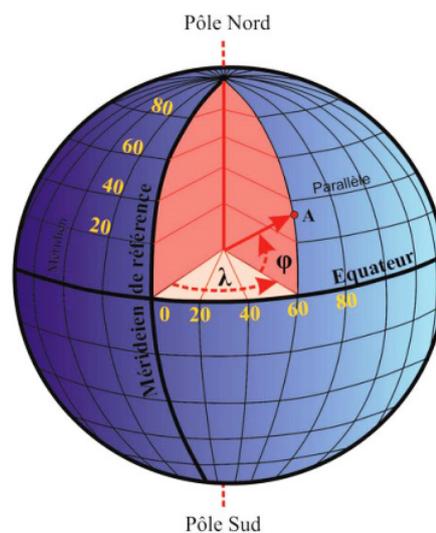


FIGURE A.2 – Les coordonnées géographiques.

Les coordonnées géographiques d'un lieu définissent sa position sur Terre. Ils sont décomposés en trois éléments : la latitude, la longitude et l'altitude.

Méridien : est une ligne imaginaire rejoignant les deux pôles terrestres. Pour pouvoir effectuer les calculs, les géographes ont désigné un méridien pour servir de point de départ : c'est le (méridien origine), ou (méridien de référence) (degré 0).

Latitude : La latitude d'un point donné est l'angle formé par l'arc du méridien entre l'équateur et le rayon terrestre passant par ce point. Elle est comptée de 0° à 90° depuis l'équateur vers le pôle pour chaque hémisphère Nord et Sud. Dans le schéma ci-dessus, la latitude du point A est φ .

Longitude : La longitude d'un point donné est l'angle formé entre le méridien de référence et le méridien passant par ce point. Elle est comptée de 0° à 180° depuis le méridien d'origine vers le méridien du lieu en précisant la direction Ouest ou Est. Dans le schéma ci-dessus, la longitude du point A est λ .

Altitude : L'altitude est la hauteur d'un point choisi par rapport à un niveau de référence (généralement le niveau de la mer, désigné comme « altitude 0 »).

A.4 Avantages du GPS

Parmi les avantages de GPS nous pouvons citer:

Position absolue : on obtient toujours une position absolue qui ne dépend pas des conditions initiales.

Conditions opérationnelles: le système marche jour et nuit et n'a pas besoin de conditions spéciales telles que support, orientation ou température.

Prix abordable : Des récepteurs simples coutent de moins en moins chers.

Précision à long terme : la précision ne se dégrade pas en fonction du temps, il n'y a pas de dérive. Le tableau suivant indique la précision du GPS selon le milieu.

technologie	rural	suburbain	urbain	intérieur
GPS	10m	20m	Entre 30 et 300 m	pas de couverture

TABLE A.1 – Présentation du GPS selon le milieu.

A.5 Conclusion

Dans cet annexe, nous avons donné une bref description du système GPS, et ces principaux composants, nous avons expliqué les trois notions des coordonnées géographiques ; « Latitude », « Longitude » et « Altitude » ayant une relation directe avec notre travail à réaliser.

Annexe B

ANDROID

B.1 Présentation du système d'exploitation AN-DROID

Android est le système d'exploitation mobile open-source de Google qui équipe la majorité des smartphones et tablettes du marché. Initialement, Android était une startup dirigée par Andy Rubin et qui a été rachetée par Google en 2005. Aujourd'hui, Android est le système d'exploitation mobile numéro un dans le monde.[8].

La figure au-dessous présente l'évolution des versions ANDROID :



FIGURE B.1 – Évolution des versions d'ANDROID.

La table ci-dessous donne plus de détails sur les différentes versions d'Android:

Nom	N version	Date	Niv Api
N/A	1.0	Septembre 2008	1
N/A	1.1	Février 2009	2
Cupcake	1.5	Avril 2009	3
Donut	1.6	Septembre 2009	4
Eclair	2.0-2.1	Octobre 2009	5-7
Froyo	2.2-2.2.3	Mai 2010	8
Gingerbread	2.3-2.3.7	Décembre 2010	9-10
Honeycomb	3.0-3.2.6	Février 2011	11-13
Ice Cream Sandwich	4.0-4.0.4	Octobre 2011	14-15
Jelly Bean	4.1-4.3.1	Juin 2012	16-18
Kitkat	4.4.x	Octobre 2013	19-20
Lollipop	5.0-5.1.1	Octobre 2014	21-22
Marshmallow	6.0-6.0.1	Mai 2015	23
Nougat	7.0-7.1.1	Septembre 2016	24-25
Oreo	8.0.0-8.1.0	Août 2017	26-27
Pie	9.0	Mars 2018	28

TABLE B.1 – Détails des versions d'ANDROID.

B.2 Architecture d'ANDROID

L'architecture de la plateforme Android se décline selon une démarche bottom up en quatre principaux niveaux que sont le noyau linux, les bibliothèques et la plateforme d'exécution, le module de développement d'applicatifs et enfin les différentes applications.[9]

B.2.1 Kernel Linux

Le noyau de Linux est la base de la plateforme Android. Par exemple, Android Runtime (ART) s'appuie sur le noyau Linux pour souligner des fonctionnalités telles que le threading et la gestion de la mémoire de bas niveau. L'utilisation d'un noyau Linux permet à Android de tirer parti des principales

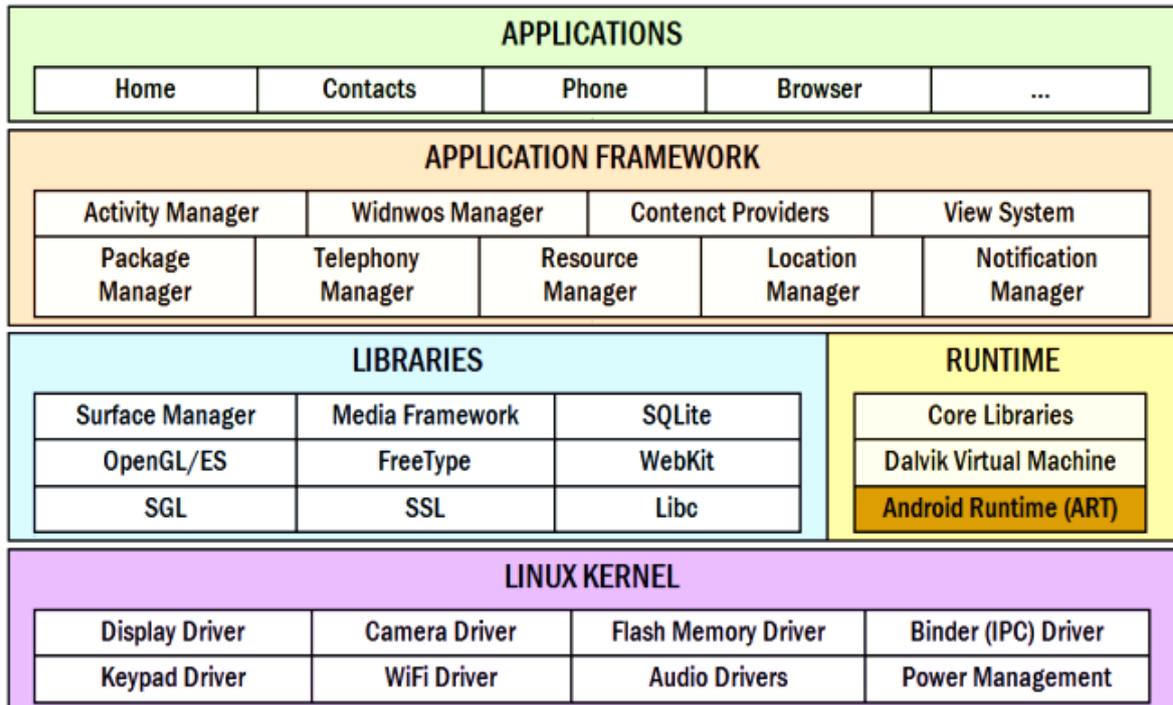


FIGURE B.2 – L'architecture ANDROID.

fonctionnalités de sécurité et aux fabricants de périphériques de développer des pilotes matériels pour un noyau bien connu.

La couche la plus basse de l'architecture repose sur un noyau linux, ce dernier a été choisit pour sa stabilité et sa performance, pour son modèle de sécurité, pour ses capacités d'abstraction avec le matériel et enfin pour son aspect open source et communautaire fort.

On cite dessous quelques exemples de programmes systèmes du noyau Linux qui servent à piloter les différents périphériques :

- Contrôleur de la caméra ou camera driver ;
- Contrôleur de la mémoire flash ou shared memory driver ;
- Contrôleur du clavier ou keypad driver ;
- Contrôleur du wifi ou wifi driver ;
- Contrôleur audio ou audio driver ;
- Contrôleur d'énergie ou power manager etc.

B.2.2 Les Bibliothèques

Écrites en C/C++, les librairies sont utilisées par les composants du système Android et sont utilisables par le développeur via le Framework Android, en voici quelques exemples :

- *Surface Manager* ou Gestionnaire de surface : gérant l'accès au sous-système d'affichage ;
- *Media Framework* ou Armature Environnement : permettant la gestion de fichier multimédia (Audio et vidéo) ;
- *SQLite* : le moteur de base de données relationnel ;
- *FreeType* : gérant les bitmap et le rendu des polices ;
- *WebKit* : permettant la navigation internet ;
- *SGL* : le moteur graphique 2D ;
- *SSL* : gérant le protocole SSL ;
- *Libc* : gérant la librairie C ;

B.2.3 Android Runtime

Il s'agit de la troisième section de l'architecture et disponible sur la deuxième couche en bas. Cette section fournit un composant clé appelé Dalvik Virtual Machine, qui est une sorte de machine virtuelle Java spécialement conçue et optimisée pour Android. La VM Dalvik utilise des fonctionnalités principales de Linux comme la gestion de la mémoire et le multi-threading, intrinsèque dans la langue Java. La VM Dalvik permet à toute application Android de fonctionner dans son propre processus, avec sa propre instance de la machine virtuelle Dalvik. Le runtime d'Android fournit également un ensemble de bibliothèques de base qui permettent aux développeurs d'applications Android d'écrire des applications Android en utilisant un langage de programmation Java standard.

B.2.4 Couche Application Framework

La couche Framework d'application fournit de nombreux services de niveau supérieur aux applications sous la forme de classes Java. Les développeurs d'applications sont autorisés à utiliser ces services dans leurs applications. Le cadre Android comprend les services clés suivants:

- Activity Manager: Contrôle tous les aspects du cycle de vie de l'application et de la pile d'activité.
- Fournisseurs de contenu: Permet aux applications de publier et de partager des données avec d'autres applications
- Resource Manager: Fournit l'accès à des ressources non intégrées au code telles que les chaînes, les paramètres de couleur et les mises en page de l'interface utilisateur.
- Notifications Manager: Permet aux applications d'afficher des alertes et des notifications à l'utilisateur.

- View System: Un jeu extensible de vues utilisé pour créer des interfaces utilisateur d'application.

B.2.5 Couche Applications

Android est fourni avec un ensemble de programmes de base (dites aussi applications natives) permettant d'accéder à des fonctionnalités telles que les e-mails, les SMS, le calendrier, les photos, les cartes, le web, etc. Ces applications sont développées à l'aide du langage de programmation Java. Pour l'utilisateur lambda (qui ne cherche pas à comprendre le fonctionnement du système), cette couche est la seule accessible et visible. [6]

B.3 Les applications Android

B.3.1 Présentation

Une application Android est une application mobile spécifiquement développée pour les smartphones et les tablettes utilisant le système Android. Elles sont de nature très variables tel que les applications jeux, mobile commerce, utilitaire, service d'information.

Sous Android, une application est composée d'une ou plusieurs activités. Une activité est la base d'un composant pour la création d'interfaces utilisateur. Afin de faciliter la cinématique de l'application, il est préconisé de n'avoir qu'une interface visuelle par activité. [14]

B.3.2 État d'une activité (application)

Une application Android peut se trouver dans des états différents qui sont : en cours d'exécution, en pause, stopper et tuer.

En cours d'exécution: l'activité se trouve au premier plan et reçoit les interactions utilisateurs. Si l'appareil a besoin de ressources, l'activité se trouvant en bas de la back stack.

Pause: l'activité est visible mais l'utilisateur ne peut pas interagir avec (cachée par une boîte dialogue par exemple). La seule différence avec l'état précédent est la non-réception des événements utilisateurs.

Stopper: l'activité n'est plus visible mais toujours en cours d'exécution. Toutes les informations relatives à son exécution sont conservées en mémoire. Quand une activité passe en état stopper, vous devez sauvegarder les données importantes et arrêter tous les traitements en cours d'exécution.

Tuer: l'activité est tuée, elle n'est plus en cours d'exécution et disparaît de la back stack. Toutes les données présentes en cache non sauvegardées sont perdues.

B.3.3 Cycle de vie d'une activité Android

Par défaut, chaque application Android s'exécute dans un processus séparé. Android gère les ressources disponibles dans un appareil et peut, si besoin, fermer des applications pour libérer des ressources (hors application en cours d'exécution). Le choix de l'application à fermer dépend fortement de l'état du processus dans lequel elle se trouve. Si Android doit choisir entre deux applications se trouvant dans le même état, il choisira celle qui se trouve dans cet état depuis plus longtemps. Le cycle de vie d'une activité est assez complexe et sa compréhension est indispensable dans le développement Android. Le schéma ci-dessous résume ce cycle de vie.

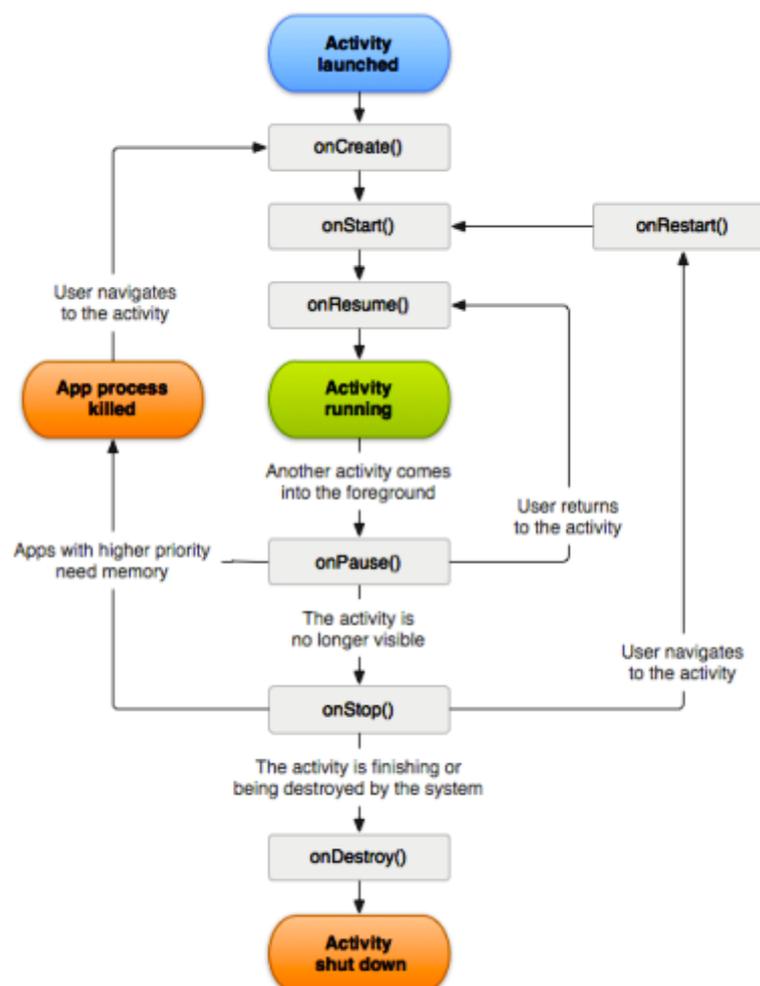


FIGURE B.3 – Cycle de vie d'une application ANDROID.

Lors du lancement d'une activité, la méthode « onCreate » est appelée. Dans cette méthode, vous devez initialiser votre vue et lier les données à une liste. Cette méthode prend en paramètre un Bundle (pile) contenant l'état précédent de l'activité. Cet appel est suivi par la méthode « onStart » afin de signifier le lancement effectif de l'application (elle est maintenant visible). Cette méthode peut aussi être appelée par la méthode « onRestart ». Puis la méthode « onResume » est appelée afin d'exécuter tous les traitements nécessaires au fonctionnement de l'activité (thread, processus, traitement), initialiser des variables et les listeners. Ces traitements devront être arrêtés lors de l'appel à la méthode « onPause » et relancés si besoin lors d'un futur appel à la méthode « onResume ». Après ces trois appels, l'activité est utilisable et peut recevoir les interactions utilisateurs. Si une autre activité passe au premier plan, l'activité en cours d'exécution passera en pause. Juste avant l'appel à la méthode « onPause », la méthode « onSaveInstanceState » est appelée afin de vous permettre de sauvegarder les informations importantes portées par l'activité. Ces informations pourront être appliquées aux futurs lancements de l'activité lors de l'appel à la méthode « onRestoreInstanceState » ou de l'appel à « onCreate ». La méthode « onPause » permet de stopper tous les traitements effectués (traitement non nécessaire si l'activité n'est pas visible) par l'activité (traitement, thread, processus). Si votre activité est de nouveau visible, cela correspondra à un appel à la méthode « onResume ». Le passage de l'activité à l'état « stopper » correspond à un appel à la méthode « onStop ». Dans cette méthode, il faut arrêter tous les traitements restants. Une fois stoppée, votre activité peut:

- Soit être relancée : cela s'effectue par un appel à la méthode « onRestart » suivi du cycle de vie normal de l'activité.
- Soit être tuée : cela s'effectue par un appel à la méthode « onDestroy », dans laquelle vous devez arrêter tous les traitements restants, fermer toutes les connexions à la base de données, tous les threads, tous les fichiers ouverts, etc. Vous pouvez provoquer l'appel à la méthode « onDestroy » en utilisant la méthode finish.[3]

B.4 Conclusion

Dans cet annexe, nous avons fait une étude du Systèmes d'Android tout en présentant l'évolution de ces versions à travers le temps depuis son apparition jusqu'à la version actuel, son architecture, à savoir les principaux composants du système, les applications mobiles sous android est leur cycle de vie. Ceci nous a permis de comprendre le fonctionnement de la plateforme de développement de notre projet.

Bibliographie

- [1] Y.KODRATOFF A.CORNUÉJOLS L.MICLET. *Apprentissage Artificiel, Concepts et algorithmes*. Paris : Éditions Eyrolles, 2002, p. 591.
- [2] ASSISTE. *Geolocalisation [en ligne]*. page consultée le 04/02/2019. URL : <https://assiste.com/Geolocalisation.html>.
- [3] NAZIM BENBOURAHLA. *Les fondamentaux du développement d'applications Java*. Paris : Editions ENI, 2017, p. 421.
- [4] XAVIER BLANC. *UML 2,pour les développeurs*. Paris : Éditions Eyrolles, 2006, p. 201.
- [5] CNIL. *Géolocalisation [en ligne]*. page consultée le 02/02/2019. URL : <https://www.cnil.fr/fr/definition/geolocalisation>.
- [6] ANDROID DEVELOPERS. *Platform Architecture [en ligne]*. page consultée le 28/04/2019. URL : <https://developer.android.com/guide/platform>.
- [7] FREDERIC EVENNOU. « Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor ». Thèse de doctorat. Grenoble : Université Joseph Fourier, 2007, p. 209.
- [8] FRANDROID. *Android [en ligne]*. page consultée le 30/03/2019. URL : <https://www.frandroid.com/android>.
- [9] RÉMI GAROT. *L'architecture de la plateforme [en ligne]*. page consultée le 30/03/2019. URL : <https://wapiti.telecom-lille.fr>.
- [10] GPS.GOV. *gps [en ligne]*. page consultée le 08/02/2019. URL : <https://www.gps.gov/systems/gps/french.php>.
- [11] INFOWEBMASTER. *Débuter avec WAMP [en ligne]*. page consultée le 15/03/2019. URL : <http://www.infowebmaster.fr/tutoriel/debuter-wamp>.
- [12] D. LARREY et L. RODIER. « Geolocalisation par WiFi ». Rapport de recherche. Nancy : Ecole des mines, 2006.
- [13] VIRGINIE MATHIVET. *L'Intelligence Artificiel, Concepts et implémentations en C*. France : Editions ENI, 2014, p. 512.

-
- [14] RETO MEIER. *Professional Android Application Development*. Indianapolis, Indiana : Wiley Publishing, 2009, p. 543.
- [15] ORACLE. *Java SE Development Kit 11 [en ligne]*. page consultée le 17/03/2019. URL : <https://www.oracle.com/technetwork/java>.
- [16] FRANCK VALLÉE PASCAL ROQUES. *UML 2 en action*. Paris : Éditions Eyrolles, 2007, p. 381.
- [17] PASCAL ROQUES. *UML 2, Modéliser une application web*. Paris : Éditions Eyrolles, 2008, p. 264.
- [18] MARGARET ROUSE. *Google Maps [en ligne]*. page consultée le 23/03/2019. URL : <https://whatis.techtarget.com/definition/Google-Maps>.
- [19] JEAN-FRANCIS ROY. « Apprentissage automatique avec garanties de généralisation à laide de méthodes denssemble maximisant le désaccord ». Thèse de doct. Université Laval, Canada, 2018.
- [20] CHRISTIAN SOUTOU. *UML 2 pour les bases de données avec 20 exercices corrigés*. Paris : Éditions Eyrolles, 2007, p. 309.
- [21] MOKHTAR TAFFAR. *Initiation à l'apprentissage automatique*. Université de Jijel, 2018, p. 82.
- [22] TUTORIALSPPOINT. *Tutoriel Laravel [en ligne]*. page consultée le 15/03/2019. URL : <https://www.tutorialspoint.com/laravel>.
- [23] GUILLAUME VOISIN. *Larchitecture MVC dans le développement dun site internet [en ligne]*. page consultée le 15/03/2019. URL : <http://www.guillaumevoisin.fr>.