

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Abderrahmane Mira de Béjaïa  
Faculté Des Sciences Exactes  
Département Informatique



Mémoire de fin de cycle  
En vue de l'obtention de diplôme de Master en informatique  
Spécialité : Administration et Sécurité des Réseaux

---

## Thème

---

*Conception et réalisation d'une application de géolocalisation*  
*Cas d'étude : Entreprise Bejaia Logistique*

---

*Realisé par :*

M<sup>lle</sup> ANKI Sylia  
M. AZI Nassim

*Devant le jury composé de :*

Présidente : M<sup>me</sup>. A. KOUICEM  
Encadreur : M. A. ACHROUFENE  
Examineur : M. A. AKILAL

ANNÉE UNIVERSITAIRE : 2015/2016

## Dédicaces

---

*Je dédie ce travail à mes très chers parents pour tous les sacrifices et les encouragements durant toute la période de mes études. Que Dieu les protègent et leur donnent tout le bonheur du monde qu'ils méritent.*

*À mon frère Billal et mes chères sœurs Dalya et Anies ainsi à mon cher hafid . Je leur souhaite un avenir radieux , plein de réussite .*

*À tous mes amis avec qui nous avons partagé les meilleurs moments de notre vie, ainsi leur aide et leur soutien moral durant l'élaboration de notre travail de fin d'études.*

*À mon binome Nassim et toute sa famille.*

*SYLIA*

## Dédicaces

---

*Je dédie ce travail à mes très chers parents et ma grande-mère et toute ma famille pour tout les sacrifices, et les encouragements durant toute la période de mes études, que Dieu les protègent et leur donnent tout le bonheur du monde qu'ils méritent ainsi à ma chère AIDA pour son soutien tout au long du projet .*

*À tous mes amis Hacene, Nabil et Lyes avec qui nous avons partagés les meilleurs moments de notre vie ainsi leur aide et leur soutien moral durant lélaboration du travail de fin détudes.*

*À mon binome Sylia et toute sa famille.*

*Nassim*

# Remerciements

---

*A l'issue de ce travail, nous tenons à remercier en premier lieu le bon Dieu qui nous a donné de l'aide pour concrétiser ce travail.*

*Un grand merci pour nos familles, surtout nos parents qui nous ont épaulés, soutenus et suivis tout au long de ce projet.*

*Nous tenons à exprimer notre vive gratitude à notre encadreur **M. ACHROUFENE Achour** de nous avoir suivis et orientés durant la réalisation de notre projet.*

*Nous tenons à remercier **M. OUZEGGUENE** pour son aide précieuse .*

*Nous remercions également les membres de jury qui ont accepté d'examiner et d'évaluer ce travail.*

*Nos sincères remerciements s'adressent aussi au personnel de **Bejaia Logistique** .*

*Enfin, nous tenons à présenter notre reconnaissance à tous ceux qui ont apporté une contribution de près ou de loin à la réalisation de ce mémoire.*

---

# TABLE DES MATIÈRES

<b>Table des matières</b>	<b>VI</b>
<b>Listes des figures</b>	<b>VIII</b>
<b>Listes des tableaux</b>	<b>IX</b>
<b>Listes des abréviations</b>	<b>X</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Généralités</b>	<b>3</b>
1.1 Technologies de géolocalisation . . . . .	3
1.1.1 Systèmes de géocalisation . . . . .	4
1.1.1.1 Système GPS . . . . .	4
1.1.1.2 Système Galileo . . . . .	4
1.1.1.3 Applications d'un GPS . . . . .	5
1.1.1.4 Composantes du GPS . . . . .	6
1.1.1.5 Principes de fonctionnement du GPS . . . . .	6

1.1.1.6	Avantages du GPS	7
1.1.1.7	Inconvénients du GPS	7
1.1.2	Localisation par GSM	8
1.1.2.1	Système global pour les communications mobiles	8
1.1.2.2	Identifiant de la cellule (Cell-ID)	8
1.1.3	Localisation par Wifi	9
1.1.4	Géolocalisation par adresse IP	9
1.2	Métriques de géolocalisation	9
1.2.1	Angle d'Arrivée (AOA)	9
1.2.2	Temps d'arrivée (TOA)	10
1.2.3	Différence de temps d'arrivée TDOA (Time Difference Of Arrival)	10
1.2.4	Puissance du Signal reçu	11
1.3	Méthodes de géolocalisation	12
1.3.1	Triangulation	12
1.3.2	Trilatération	13
1.4	Géolocalisation hybride	14
<b>2</b>	<b>Spécification des besoins</b>	<b>16</b>
2.1	UML	16
2.1.1	Définition	16
2.1.1.1	Diagrammes UML	17
2.1.1.2	Logiciels de modélisation UML	18
2.2	Méthodes agiles	19
2.2.1	Définition	19
2.2.2	Structures opérationnelles et pratiques du développement agile	19
2.2.3	Principes généraux de la modélisation agile	20
2.2.4	Processus de modélisation d'une application web	21

2.2.5	Présentation d'UP, RUP et XP . . . . .	22
2.2.5.1	UP - Unified Process . . . . .	22
2.2.5.2	RUP - Rational Unified Process . . . . .	23
2.2.5.3	XP - eXtreme Programming . . . . .	23
2.3	Présentation de l'entreprise . . . . .	24
2.3.1	Historique . . . . .	24
2.3.2	Missions et activités de l'entreprise . . . . .	25
2.3.3	Problématique . . . . .	25
2.3.4	Solutions proposées . . . . .	26
2.4	Spécification des besoins . . . . .	26
2.4.1	Exigences des utilisateurs . . . . .	26
2.4.2	Exigences des administrateurs . . . . .	27
2.5	Diagramme de cas d'utilisation . . . . .	27
2.5.1	Définitions . . . . .	27
2.5.2	Diagramme de cas d'utilisation côté d'administrateur . . . . .	28
2.5.3	Diagramme de cas d'utilisation côté utilisateur . . . . .	30
2.5.4	Maquette . . . . .	31
<b>3</b>	<b>Analyse et Conception</b>	<b>33</b>
3.1	Diagramme de séquence . . . . .	33
3.1.1	Définitions . . . . .	33
3.1.2	Réalisation des diagrammes de séquences . . . . .	34
3.1.2.1	Diagramme de séquence pour l'authentification . . . . .	35
3.1.2.2	Diagramme de séquence pour la géolocalisation en temps réel . . . . .	36
3.1.2.3	Diagramme de séquence pour contacter l'utilisateur . . . . .	36
3.1.2.4	Diagramme de séquence pour consulter l'historique de localisation . . . . .	37
3.1.2.5	Diagramme de séquence pour les recherches sur la carte . . . . .	38

3.1.2.6	Diagramme de séquence pour la gestion des zones d’alertes . . . . .	39
3.2	Diagrammes de communications . . . . .	40
3.2.1	Définitions . . . . .	40
3.2.2	Réalisation des diagrammes de communication . . . . .	41
3.2.2.1	Diagramme d’authentification . . . . .	41
3.2.2.2	Diagramme de communication pour géolocalisation en temps réel . . . . .	42
3.2.2.3	Diagramme de communication pour effectuer des recherches sur la carte . . . . .	42
3.2.2.4	Diagramme communication pour consulter l’historique de localisa- tion . . . . .	43
3.2.2.5	Diagramme de communication pour la gestion des zones alertes . . . . .	44
3.3	Diagramme de navigation . . . . .	47
3.4	Diagrammes de classes . . . . .	48
3.4.1	Définitions . . . . .	48
3.4.2	Dictionnaire de données . . . . .	49
3.4.3	Réalisation du diagramme de classes . . . . .	50
3.5	Modèle relationnel de données . . . . .	50
3.5.1	Définition . . . . .	50
3.5.2	Les règles de passage au relationnel . . . . .	51
3.5.3	Application . . . . .	51
<b>4</b>	<b>Réalisation</b> . . . . .	<b>53</b>
4.1	Langages de programmation . . . . .	53
4.1.1	Langage HTML5 . . . . .	53
4.1.2	Langage CSS . . . . .	54
4.1.3	Langage java . . . . .	54
4.1.4	Langage java script . . . . .	54



4.1.5	Langage PHP	54
4.1.6	Langage SQL	54
4.1.7	Langage XML	55
4.2	Environnements de développement	55
4.2.1	Wampserver	55
4.2.1.1	Apache	55
4.2.1.2	Phpmyadmin	55
4.2.1.3	MySQL	55
4.2.2	Android Studio	55
4.2.3	Éclipse	56
4.3	Schéma de fonctionnement de l'application	57
4.4	Architecture de développement	57
4.4.1	MVC	58
4.4.1.1	JSP ( Java Server Pages)	59
4.4.1.2	Les servlets	59
4.4.1.3	JavaBean	59
4.4.1.4	Service web	60
4.4.2	Communication avec la base de données	60
4.4.2.1	JDBC	60
4.4.2.2	Hibernate	60
4.4.3	GoogleMaps	61
4.4.3.1	Définitions	61
4.4.3.2	Api GoogleMaps	61
4.5	Présentation de quelques interfaces	62
4.5.1	Interface de connexion	62
4.5.2	Interface menu d'administrateur	63

4.5.3	Interface d'ajout d'un camion . . . . .	64
4.5.4	La liste des utilisateurs . . . . .	65
4.5.5	Interface suivi en temps réel . . . . .	66
4.5.6	Interface historique d'un camion . . . . .	67
4.5.7	Interface application d'android . . . . .	68
	<b>Conclusion générale</b>	<b>69</b>
	<b>Bibliographie</b>	<b>70</b>
	<b>Annexe</b>	<b>75</b>

---

# TABLE DES FIGURES

1.1	Schéma de principes de géolocalisation par GPS [w17]. . . . .	4
1.2	Identifiant de cellule Cell-ID . . . . .	8
1.3	Localisation par AOA . . . . .	9
1.4	Localisation par TOA . . . . .	10
1.5	TDOA . . . . .	11
1.6	La puissance du Signal . . . . .	12
1.7	Triangulation avec trois satellites. . . . .	13
1.8	Méthode de trilatération . . . . .	13
1.9	Système hybride de localisation [w19]. . . . .	14
1.10	GPS assisté (A-GPS). . . . .	15
2.1	Les 14 diagrammes d'UML . . . . .	18
2.2	Schéma complet du processus de modélisation d'une application web . . . . .	22
2.3	La méthode de développement UP [w7] . . . . .	23
2.4	Diagramme de cas d'utilisation d' administrateur . . . . .	28
2.5	Diagramme de cas d'utilisation pour le gestionnaire . . . . .	31

2.6	Maquette de l'application de géolocalisation . . . . .	32
3.1	Diagramme de séquence pour l'authentification . . . . .	35
3.2	Diagramme de séquence pour la géolocalisation en temps réel . . . . .	36
3.3	Diagramme de séquence pour contacter l'utilisateur . . . . .	37
3.4	Diagramme de séquence pour consulter l'historique de localisation . . . . .	38
3.5	Diagramme de séquence pour consulter les recherches sur la carte . . . . .	39
3.6	Diagramme de séquence pour la gestion des alertes . . . . .	40
3.7	Diagramme de communication d'authentification . . . . .	41
3.8	Diagramme de communication pour géolocalisation en temps réel . . . . .	42
3.9	Diagramme de communication pour effectuer des recherches sur la carte . . . . .	43
3.10	Diagramme de communication pour consulter l'historique de localisation . . . . .	44
3.11	Diagramme de communication pour la gestion des zones d'alertes . . . . .	46
3.12	Diagramme de navigation . . . . .	47
3.13	Diagramme de classes . . . . .	50
4.1	Fonctionnement général de l'application . . . . .	57
4.2	Architecture trois-tiers . . . . .	58
4.3	Interface de connexion . . . . .	62
4.4	Interface administrateur . . . . .	63
4.5	Interface d'ajout d'un camion . . . . .	64
4.6	Interface liste des utilisateurs . . . . .	65
4.7	Interface suivi en réel des camions . . . . .	66
4.8	Interface historique d'un camion . . . . .	67
4.9	Interface application android . . . . .	68

---

# LISTE DES TABLEAUX

1.1	Présentation du GPS selon le milieu . . . . .	7
2.2	Identification des cas d'utilisation . . . . .	30
3.2	Dictionnaire de données . . . . .	49

---

# LISTES DES ABRÉVIATIONS

**AOA** Angle of Arrival.

**BDD** Base De Donnée.

**BTS** Base Transciever Station .

**CID** Cell ID.

**CSS** Cascading Style Sheets.

**GSM** Global System for Mobile Communications.

**GPRS** General Packet Radio Service .

**GPS** General Packet Service .

**HTML** HyperText Markup Language.

**HTTP** Hypertext Transfer Protocol.

**IANA** Internet Assigned Numbers Authorit.

**IDE** International Development Enterprises.

**IP** Internet Protocol.

**JDBC** Java DataBase Connectivity.

**JEE** Java Entreprise Edition.

**JSP** Java Server Pages.

**MSC** Commutateur de Services Mobiles.

**PHP** Hypertext Preprocessor.

**RSS** Received SSignal Strength.

**RUP** Rational Unfied Process.

**SQL** Structured Query Language.  
**TDOA** Time Diference Of Arrival.  
**TOA** Time Of Arrival.  
**UML** Unified Modeling Language.  
**UP** Unfied Process.  
**XML** eXtensible Markup Language.

---

# INTRODUCTION GÉNÉRALE

---

Au cours de ces dernières années, les nouvelles technologies de l'information et de la communication ont connu un bouleversement marqué par l'apparition de l'internet et sa croissance exponentielle. Ces années ont aussi été marquées par l'entrée en scène du système de géolocalisation. Qu'est l'une des applications qui ont popularisé le monde et qui ont fait exploser le nombre de ses utilisateurs. Par ailleurs, connaître la position d'un mobile à l'instant T devient un véritable atout pour l'entreprise Bejaia Logistique. Cette traçabilité offre de réels avantages permettant à l'entreprise de gérer d'une manière centralisée l'ensemble des camions. Cette dernière est d'un intérêt capital puisqu'elle offre aux gestionnaires des camions, souvent confrontés à des problèmes d'ordre technique dans la localisation en temps réel de leurs engins, l'occasion 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 son matériels. Ainsi la position du camion est suivie depuis le siège de la société et l'on dispose alors de différentes informations : le chemin emprunté, la présence en dépôt, la position et la distance déjà parcourue.

Dans cette perspective , notre travail au sein de Bejaia Logistique vise à concevoir et à réaliser une application qui permettra aux gestionnaires de gérer et de suivre en temps réel les camions sur une carte géographique.

Pour bien pouvoir concrétiser notre objectif, nous avons décidé d'organiser notre travail de recherche en le répartissant en quatre chapitres :

- ✓ Le premier chapitre définit les généralités sur le système de géolocalisation comme les technologies (GPS, GSM, WIFI, adresse IP), les métriques de géolocalisation (AOA, TOA, TDOA et puissance du signal reçu) et les méthodes de géolocalisation (Triangulation, Trilatération).
- ✓ Le deuxième chapitre est consacré à la spécification des besoins pour l'application de géolocalisation (UML et méthode agile, diagramme de cas d'utilisation).
- ✓ Le troisième chapitre est dédié à l'analyse et la conception de notre application (diagramme de séquence, diagramme de communication, diagramme de navigation, diagramme de classe et enfin le modèle relationnel de données).



- ✓ Le quatrième chapitre, quant à lui, est consacré à la réalisation et l'implémentation de l'application de géolocalisation on utilisant les langages de programmation (Java, XML,HTML...), environnements développement(Wampserver, Android studio,Eclipse), architecture de développement (MVC) et communication avec la base de données (JDBC, hibernate).

Nous terminons ce rapport par une conclusion générale permettant de résumer le travail accompli et de donner quelques perspectives.

---

---

# CHAPITRE 1

---

## GÉNÉRALITÉS

### Introduction

L'homme cherche depuis son existence à savoir sa position pour pouvoir se déplacer sur la terre ou dans la mer. Alors au début il s'oriente grâce aux étoiles. Les explorateurs ont utilisé les étoiles, le soleil ... comme une boussole depuis des siècles pour tracer leur chemin au cours de leurs voyages.

Depuis quelques années ce concept est plus développé grâce à l'apparition de nouvelles technologies de géolocalisation, parmi celles-ci, on cite le GSM, wifi et GPS. On donnera plus de détails sur le GPS [w18]. Le système mondial de positionnement ou Géopositionnement par satellite est un système de géolocalisation fonctionnant au niveau mondial et reposant sur l'exploitation de signaux radio émis par des satellites dédiés.

Un système de positionnement est composé des éléments suivants : Technologies, métriques et méthode que nous allons présenter dans ce qui suit.

### 1.1 Technologies de géolocalisation

La géolocalisation utilise différentes technologies (GPS, GSM, WIFI, IP). Qui seront présentés dans cette section.

### 1.1.1 Systèmes de géolocalisation

#### 1.1.1.1 Système GPS

Le système GPS est mis en place par le département de la Défense des États-Unis à des fins militaires. Il est très rapidement apparu que des signaux transmis par les satellites pouvaient être librement reçus et exploités, et qu'ainsi un récepteur pouvait connaître sa position sur la surface de la terre, avec une précision sans précédent, dès l'instant qu'il était équipé des circuits électroniques et du logiciel nécessaires au traitement des informations reçues [w18]. Une personne munie de ce récepteur peut ainsi se localiser et s'orienter sur la terre, dans la mer, dans l'air ou dans l'espace au voisinage proche de la terre .

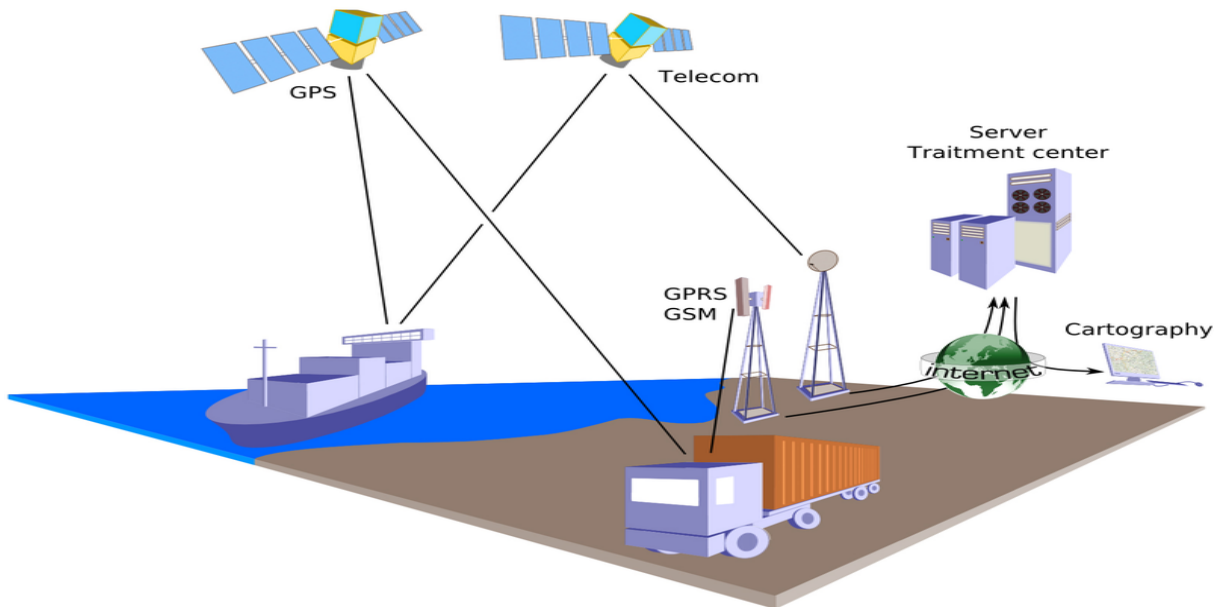


FIGURE 1.1 – Schéma de principes de géolocalisation par GPS [w17].

#### 1.1.1.2 Système Galileo

Galileo [w1] est un système de positionnement par satellites développé par l'union européenne dont le déploiement doit s'achever vers 2020. Il permettra à un utilisateur muni d'un terminal de réception d'obtenir sa position sur une carte géographique. Galileo couvrira toute la surface de la terre, chaque point donné étant balayé à tout moment par six à huit satellites. Cela permettra de disposer de données de positionnement précises

par exemple dans les villes à gratte-ciel dont les immeubles de grande hauteur peuvent gêner la réception des signaux.

### 1.1.1.3 Applications d'un GPS

Les fonctionnalités d'un traceur GPS sont nombreuses et peuvent être utilisées dans divers cas de figure, de manière professionnelle ou personnel.

#### ► Application personnelle

Les applications à usage personnel du traceur GPS sont aussi nombreuses, il s'applique par exemple dans le domaine automobile pour l'orientation des chauffeurs dans les différents axes routiers, la localisation des personnes malades, seules, ou vulnérables, localisation d'animaux.

Un exemple sur la géolocalisation pour personnes âgées est décrit dans [w2]. En fait le vieillissement démographique et le pourcentage de personnes âgées ne cesse de croître, avec son lot de maladies chroniques. Les personnes atteintes de la maladie d'Alzheimer par exemple devaient être constamment assistées et surveillées. Le traceur GPS s'utilise afin d'envoyer des informations sur la géolocalisation d'une personne. De manière automatique et instantanée, l'information arrive sur un site internet qui montre la position exacte sur une carte géographique. Des zones personnalisées peuvent être définies. Une alarme prévient si la personne quitte ou entre dans une zone.

#### ► Application professionnelle

La géolocalisation dans le milieu professionnel est presque un synonyme de gain de productivité, économies de carburant, économies de communications et sécurité [w3]. Ceci permet à une entreprise de gagner en temps, de réduire les coûts afin d'accroître sa compétitivité. Les applications de géolocalisation sont généralement réservées aux professionnels pour suivre une flotte de camions, véhicules de transports de passagers (y compris les taxis). D'améliorer la gestion d'une flotte de véhicules, notamment dans les transports. En outre ces solutions offrent à l'employeur une vision bien précise de sa flotte et une meilleure réactivité en cas d'incident.

#### 1.1.1.4 Composantes du GPS

Le système GPS comporte trois segments [2] :

※ **Segment spatial**

Il inclut tous les éléments orbitaux du dispositif de positionnement : les plates formes, le signal GPS et le message de navigation.

※ **Segment contrôle**

Une station principale (Colorado Springs) reçoit les données des autres stations, calculent les éphémérides des satellites et surveillent le fonctionnement du système dans le but de l'enregistrement et d'observation des signaux, le calcul des paramètres de synchronisations et d'horloge, la collecte d'informations météorologiques, l'envoi aux satellites des informations nécessaires à la composition du message de navigation .

※ **Segment utilisateur**

Le segment utilisateur est constitué de récepteurs qui ont été conçus afin de décoder le signal transmis par les satellites pour déterminer la position, la vitesse et le temps de l'utilisateur .

#### 1.1.1.5 Principes de fonctionnement du GPS

Les quatre étapes suivantes expliquent le fonctionnement d'un GPS [w4] :

1. Les satellites envoient des signaux à des intervalles de temps réguliers. Chacun de ces signaux contient les informations suivantes : un code permettant d'identifier le satellite, l'instant "t" auquel le signal est transmis, sa hauteur et sa position par rapport aux coordonnées géographiques terrestres ;
2. Le récepteur GPS reçoit ces ondes radios et utilise les informations contenues dans les signaux ;
3. En mesurant la distance qui le sépare du satellite, le récepteur peut calculer sa position : latitude, longitude, altitude. Avec les signaux de trois satellites, il calcule la latitude et la longitude. Avec les signaux de quatre, on obtient l'altitude. Il est à signaler qu'à tout moment, chaque point de la Terre est couvert par quatre satellites et que plus le récepteur reçoit des signaux, plus sa précision de calcul est importante ;

4. Le satellite renvoie sa position théorique à la station de contrôle, qui calcule alors l'erreur de position commise par ce dernier afin de lui renvoyer la valeur de cette erreur ;

#### 1.1.1.6 Avantages du GPS

Parmi les avantages de GPS nous pouvons citer [3] :

- Position absolue : on obtient toujours une position absolue qui ne dépend pas des conditions initiales.
- 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 300m	pas de couverture

TABLE 1.1 – Présentation du GPS selon le milieu

- 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 coûtent de moins en moins chers. Le prix est abordable pour des véhicules grand public. Le dispositif de géolocalisation se situe entre 12 500.00 DA et 45 000.00 DA en Algérie.

#### 1.1.1.7 Inconvénients du GPS

Le GPS présente également quelques inconvénients [3] :

- Le GPS est un système américain sur lequel les utilisateurs n'ont aucun contrôle ni aucune garantie légale de bon fonctionnement.
- Géométrie des satellites : ceci se rapporte à la position relative des satellites selon l'instant d'observation. La géométrie satellitaire idéale existe quand les satellites sont situés à des angles relativement larges. On dispose d'une géométrie faible quand les satellites sont alignés ou en groupe serré.
- les trajets multiples : ceci se produit quand le signal de GPS est reflété par des objets avant qu'il atteigne le récepteur. Ceci augmente le temps de parcours du signal, causant par conséquent une surévaluation du temps de vol et générant par conséquent des erreurs de localisation.

- Faible disponibilité : dépend du nombre de satellites visibles par le récepteur. Dans les applications de le domaine automobile, les conditions de visibilité sont souvent dégradées (les bâtiments, les tunnels, l'interférence électronique). Ceci peut bloquer la réception du signal, entraînant des erreurs de localisation voir causant un arrêt temporaire du système.

## 1.1.2 Localisation par GSM

### 1.1.2.1 Système global pour les communications mobiles

Le système global pour les communications mobiles en anglais Global System Mobiles (GSM) est un moyen de communication sans fil qui fonctionne par transmission d'ondes entre une base relais et le téléphone portable de l'utilisateur, couvrant une zone de plusieurs kilomètres [4].

#### 1.1.2.2 Identifiant de la cellule (Cell-ID)

Cette méthode est basée sur le réseau GSM et plus particulièrement sur l'indentification de la cellule. Elle utilise l'identifiant de la BTS (Base Transciever Station) qui prend en charge les communications radio des téléphones mobiles dans lequel la station mobile est actuellement enregistrée. La topologie du réseau GSM est ensuite utilisée afin d'estimer la position du mobile. La précision de cette méthode dépend principalement de la taille de la cellule ainsi que sur l'environnement avoisinant (c'est-à-dire rural ou urbain)[5].

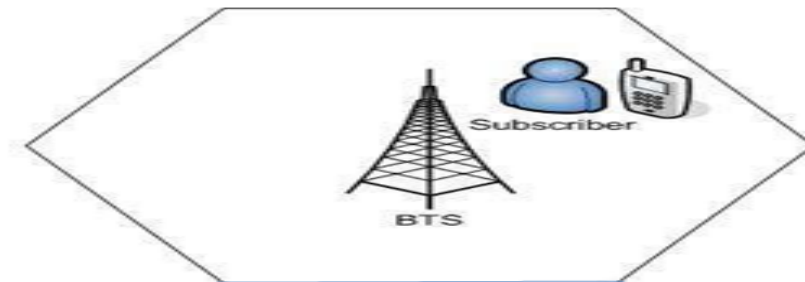


FIGURE 1.2 – Identifiant de cellule Cell-ID

### 1.1.3 Localisation par Wifi

La géolocalisation Wi-Fi [6] est utilisé 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) et sur la méthode d'empreinte digitale. Cette dernière est constituée du RSS, du SSID (Service Set Identifie) du point d'accès, et de l'adresse MAC du routeur.

### 1.1.4 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 adresse 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 [6]. Il est possible de savoir dans quel pays se trouve un terminal connecté à internet grâce à son adresse IP. Elle a un inconvénient majeur, la précision qu'est trop faible même parfois elle donne une localisation erronée.

## 1.2 Métriques de géolocalisation

### 1.2.1 Angle d'Arrivée (AOA)

En mesurant l'angle d'arrivé des signaux transmis par au moins deux BTS, il est possible de déterminer la position de la station mobile par triangulation. La contrainte majeure de cette métrique est que celle-ci requière un environnement libre d'obstacle, cette méthode est donc peu efficace en zone urbaine [7].



FIGURE 1.3 – Localisation par AOA



### 1.2.2 Temps d'arrivée (TOA)

TOA calcule le temps de transmission entre la station mobile et la BTS et vice versa. Considérant le fait que le temps de propagation d'une onde radio est connu, il est alors possible d'estimer la distance qui sépare la station mobile du BTS. Cette méthode permet de localiser l'utilisateur dans un cercle qui a pour rayon la distance qui sépare la BTS de la MSC (le commutateur de services mobiles) et qui a pour centre le BTS [8].

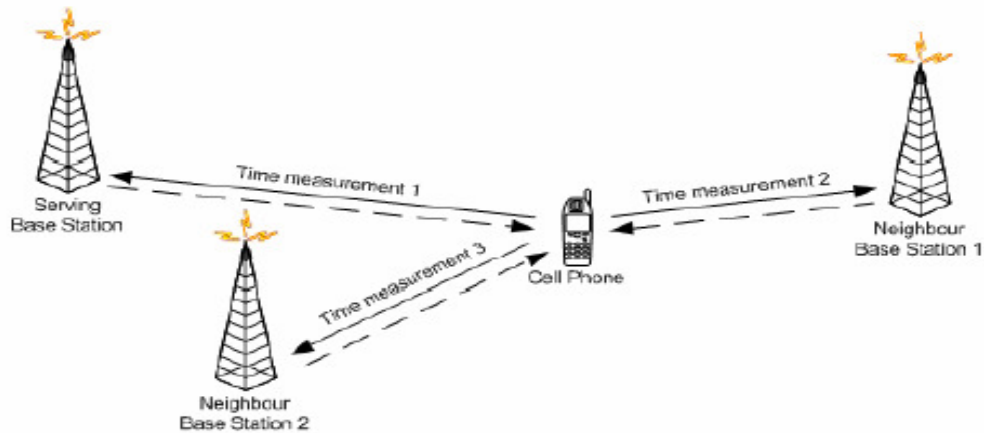


FIGURE 1.4 – Localisation par TOA

### 1.2.3 Différence de temps d'arrivée TDOA (Time Difference Of Arrival)

TDOA mesure la différence de temps de transmission entre la station mobile et trois BTS simultanément, la station mobile se situe à l'intersection des trois cercles.

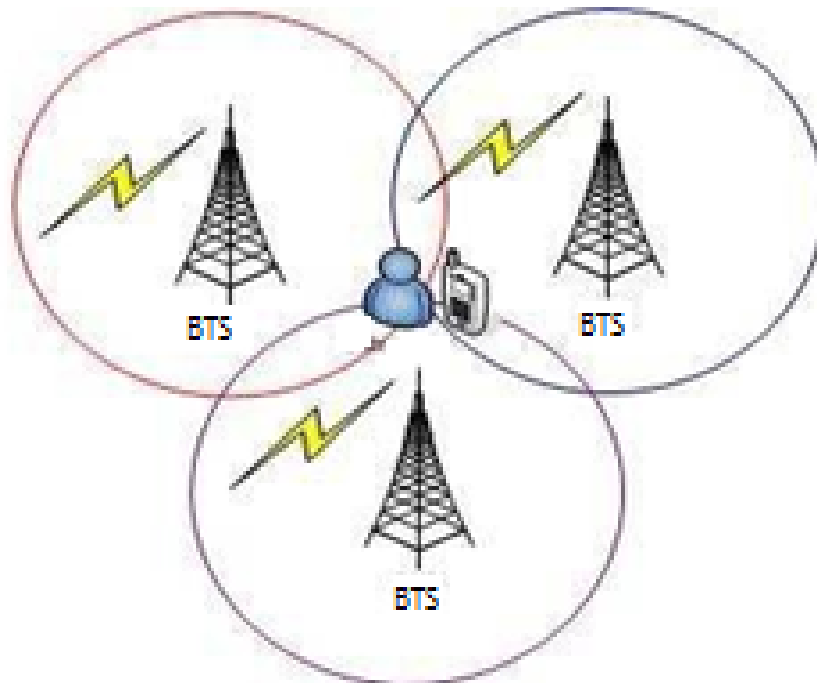


FIGURE 1.5 – TDOA

#### 1.2.4 Puissance du Signal reçu

Cette métrique utilise la force du signal pour déterminer la position de la station mobile. Si la puissance de transmission de la BTS et que la puissance de réception de la station mobile sont connus, il est possible d'estimer la distance qui sépare ces deux équipements. Encore une fois la station mobile se situera sur un cercle qui aura pour rayon la distance entre la BTS et la station mobile et pour centre la BTS. Si la puissance du signal est estimée entre trois BTS, la station mobile se situera à l'intersection des trois cercles [9].

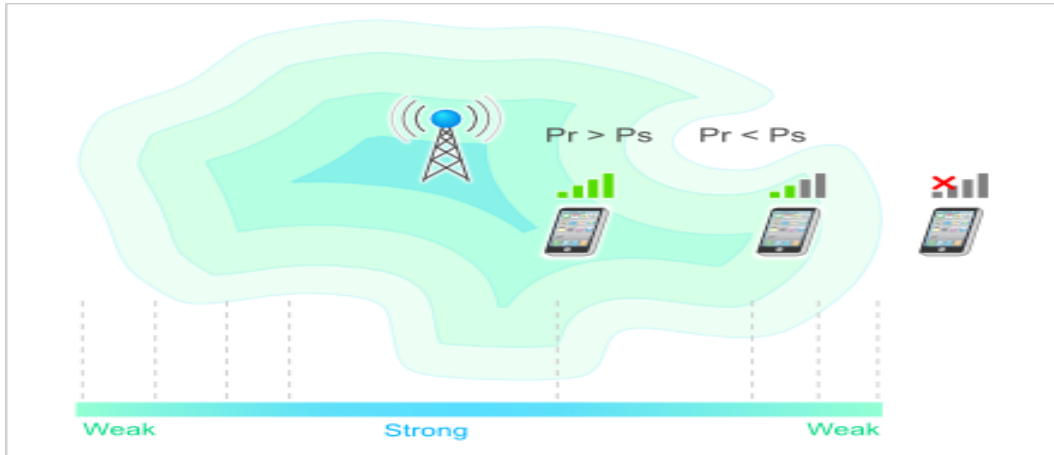


FIGURE 1.6 – La puissance du Signal

### 1.3 Méthodes de géolocalisation

Le troisième élément d'un système de positionnement est les méthodes de localisation. Nous présenterons deux méthodes les plus connues et les plus utilisées qui sont la triangulation et la trilatération.

#### 1.3.1 Triangulation

Nous expliquerons le fonctionnement de la méthode de triangulation par exemple de GPS. Le dernier calcule la position comme suit [1] :

- le satellite émet une onde électromagnétique de vitesse connue (c'est la vitesse de la lumière).
- le récepteur calcule le temps mis par cette onde pour l'atteindre.
- le récepteur sait alors qu'il se trouve sur une sphère centrée sur le satellite.
- Le récepteur intercepte les signaux et calcule la position à partir des données que fournissent les satellites.
- En recoupant les informations de 2 satellites, le lieu géométrique du récepteur devient un cercle.
- Avec 3 satellites, l'intersection se réduit à un (ou 2) points.

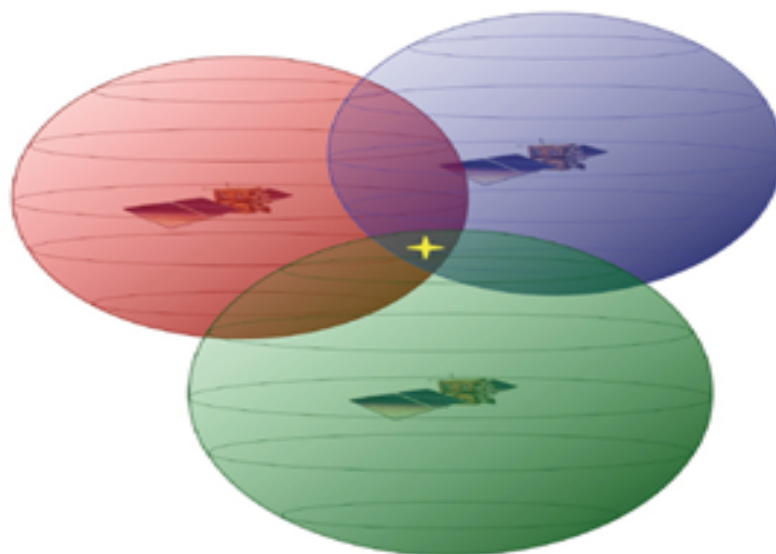


FIGURE 1.7 – Triangulation avec trois satellites.

### 1.3.2 Trilatération

La trilatération est une méthode qui permet de positionner un nœud à l'aide de trois nœuds de référence [10] .

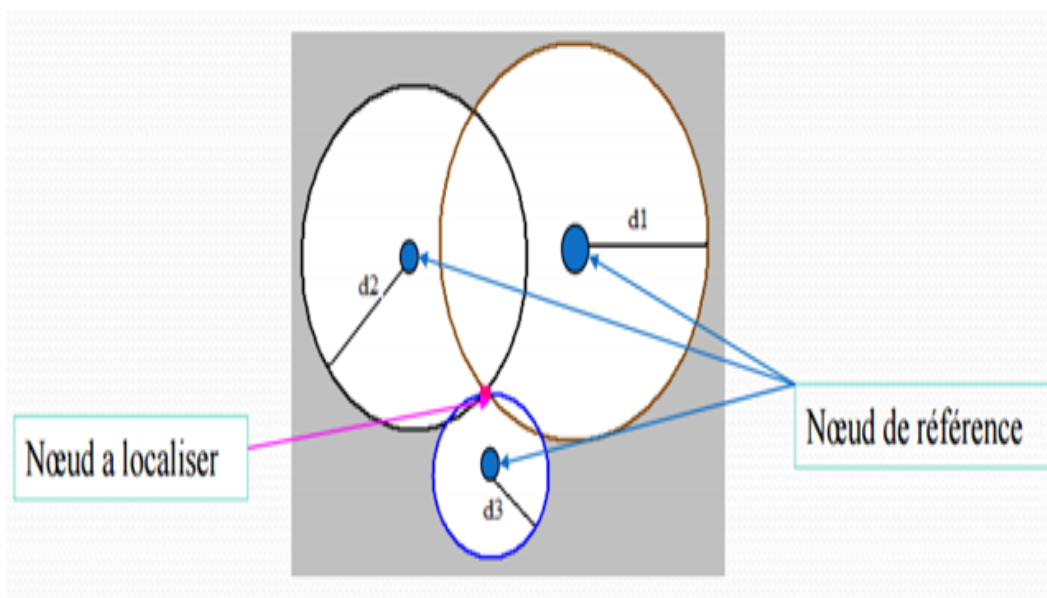


FIGURE 1.8 – Méthode de trilatération .

Cet méthode consiste donc à mesurer les distances qui séparent le nœud à positionner

aux nœuds de référence. Ensuite, on trace un cercle autour de chaque nœud de référence, le nœud à positionner se situe à l'intersection de ces trois cercles. Cela nécessite trois nœuds pour localiser un capteur.

## 1.4 Géolocalisation hybride

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 solution hybride de localisation offre une plus grande précision et une meilleure cohérence pour les services grand public (recherches locales ou publicité localisée).

La solution hybride de localisation peut intégrer une combinaison de technologies de localisation telle l'identification cellulaire (Cell ID, CID), l'angle d'arrivée (Angle of Arrival, AOA), la différence de temps d'arrivée, et le GPS assisté (Assisted GPS, A-GPS), afin de garantir une meilleure précision de localisation d'un terminal dans n'importe quel environnement.

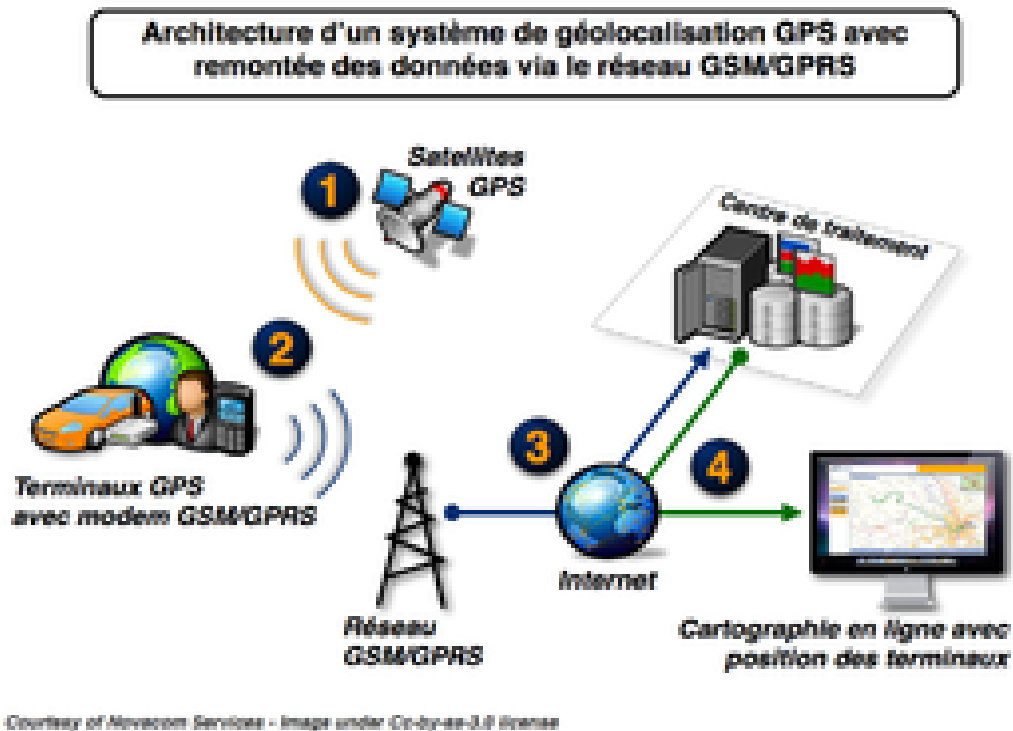


FIGURE 1.9 – Système hybride de localisation [w19].

Parmi les systèmes hybrides on trouve le GPS assisté (A-GPS) qui fonctionne de la manière suivante : la station mobile mesure le temps d'arrivée des signaux envoyés par trois Satellites ou plus, les données reçues par le récepteur sont ensuite envoyées via connexion radio aux BTS, ceci tend à résoudre les problèmes liés à la consommation d'énergie, puisque le traitement du signal ne se fait pas directement par la station mobile.

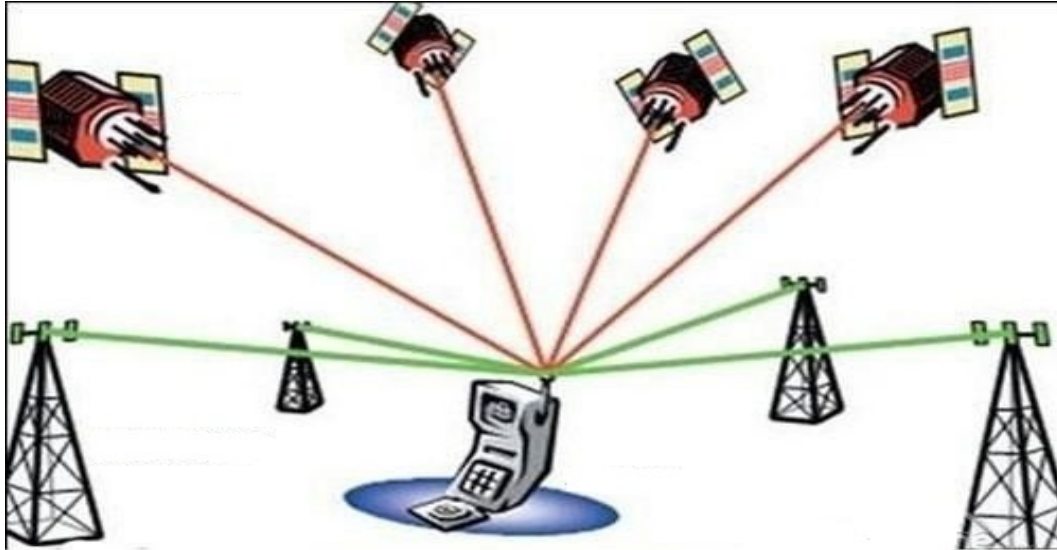


FIGURE 1.10 – GPS assisté (A-GPS).

### Conclusion

Dans ce chapitre nous avons présenté les méthodes de géolocalisation qui permettent d'obtenir et éventuellement transmettre la position géographique d'une personne ou d'une ressource. La combinaison de différentes techniques est capable de géolocaliser le terminal dans n'importe quelle position. Dans le prochain chapitre, nous allons présenter un autre élément que nous utiliserons pour la modélisation de notre application web, il s'agit d'UML (Unified Modeling Language), et nous expliquerons le choix des méthodes agiles et les spécification des besoins.

---

---

# CHAPITRE 2

---

## SPÉCIFICATION DES BESOINS

### Introduction

Le présent chapitre sera consacré à la spécification des besoins du projet et à sa modélisation. Pour cela, nous allons commencer par présenter le langage de modélisation UML que nous avons choisi pour ses divers avantages (modularité, abstraction, structuration cohérente des fonctionnalités et des données, diversité des diagrammes, ). Nous définirons par la suite les méthodes de développement agiles et nous justifierons le choix de cette méthode. Ensuite nous allons exposer le contexte du projet, les exigences du client et réaliser les diagrammes de cas d'utilisations. Enfin on appliquera ces outils pour notre projet de géolocalisation des camions de Bejaia Logistique qui consiste à suivre les mouvements en temps réel des camions.

### 2.1 UML

#### 2.1.1 Définition

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue [w5].

### 2.1.1.1 Diagrammes UML

La version UML 2.4 s’articule autour de quatorze types de diagrammes, chacun d’eux étant dédié à la représentation des concepts particuliers d’un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes [11] .

- **Diagrammes structurels ou statiques**

- Diagramme de classes : il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc.
- Diagramme d’objets : il montre les instances des éléments structurels et leurs liens à l’exécution.
- Diagramme de packages : il montre l’organisation logique du modèle et les relations entre packages.
- Diagramme de structure composite : il montre l’organisation interne d’un élément statique complexe.
- Diagramme de composants : il montre des structures complexes, avec leurs interfaces fournies et requises.
- Diagramme de déploiement : il montre le déploiement physique des ” artefacts ” sur les ressources matérielles.
- Diagramme de profile : il permet de spécialiser, de personnaliser pour un domaine particulier un méta-modèle de référence d’UML.

- **Diagrammes comportementaux**

- Diagramme de cas d’utilisation : il montre les interactions fonctionnelles entre les acteurs et le système à l’étude.
- Diagramme d’activité : il montre l’enchaînement des actions et décisions au sein d’une activité.
- Diagramme d’états : il montre les différents états et transitions possibles des objets d’une classe.

- **Diagrammes d’interaction ou dynamiques**

- Diagramme de vue d’ensemble des interactions : il fusionne les diagrammes d’activité et de séquence pour combiner des fragments d’interaction avec des décisions et des flots.



- Diagramme de séquence : il montre la séquence verticale des messages passés entre objets au sein d’une interaction.
- Diagramme de communication : il montre la communication entre objets dans le plan au sein d’une interaction.
- Diagramme de temps : il fusionne les diagrammes d’états et de séquence pour montrer l’évolution de l’état d’un objet au cours du temps.

La figure 2.1 présentée ci-dessous présente les 14 diagrammes d’UML.

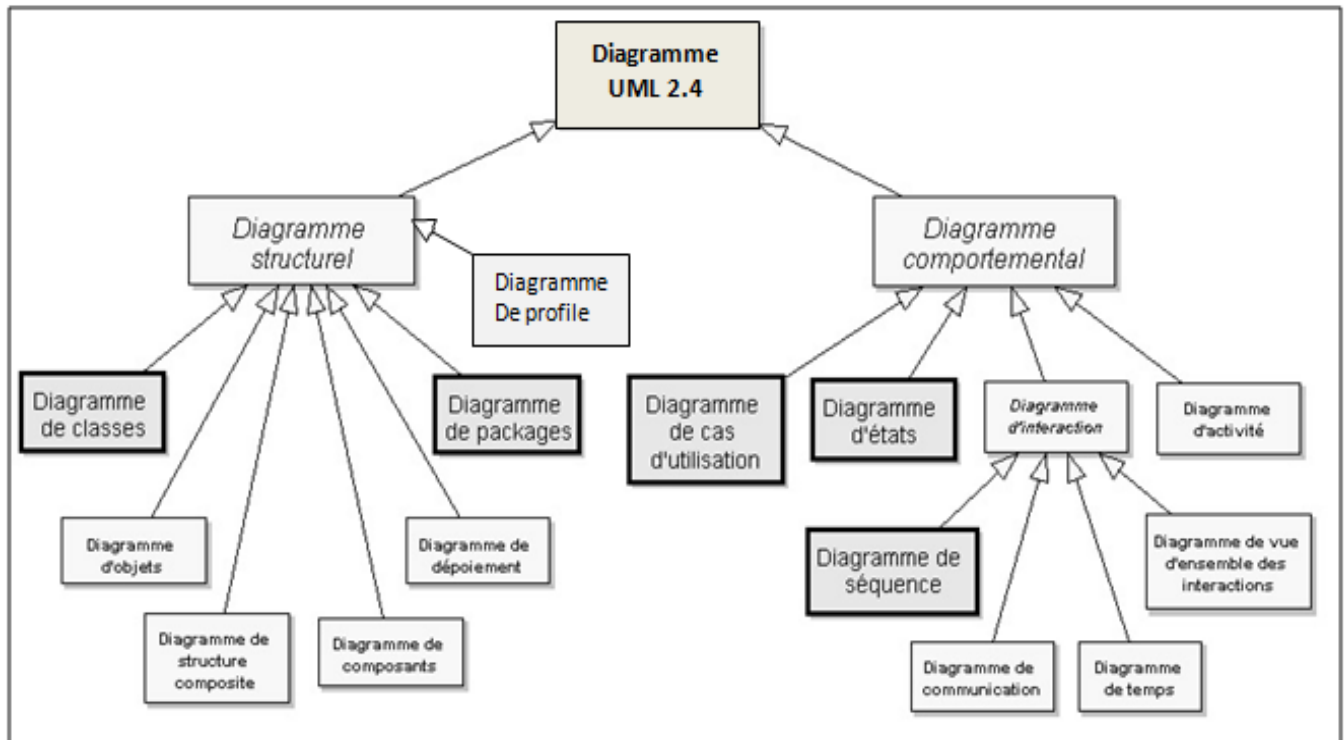


FIGURE 2.1 – Les 14 diagrammes d’UML

### 2.1.1.2 Logiciels de modélisation UML

Il existe de nombreux outils logiciels de modélisation UML. Comme le logiciel de dessin Dia [w16] qui permet la création de diagrammes UML. Le dessin s’effectue en choisissant des formes (flèches, acteurs, classes, notes, états ) et en utilisant des bibliothèques de symboles pour aboutir à un schéma correspondant. Les créations peuvent être exportées dans différents formats (PDF, JPG, PNG ...). Dans ce qui suit, on utilisera la méthode agile pour permettre de répondre mieux au besoin réel du client ainsi, permettre

au développeur une grande visibilité sur l'avancement du projet en s'appuyant sur la modalisation UML de notre future application.

## 2.2 Méthodes agiles

### 2.2.1 Définition

Les méthodes de développement dites méthodes agiles (en anglais Agile Modeling, noté AG) visent à réduire le cycle de vie du logiciel (donc accélérer son développement) en développant une version minimale, puis en intégrant les fonctionnalités par un processus itératif basé sur une écoute client et des tests tout au long du cycle de développement. Le terme agile fait référence à la capacité d'adaptation aux changements de contexte et aux modifications de spécifications intervenant pendant le processus de développement. En 2001, 17 personnes mirent ainsi au point le manifeste agile dont la traduction est la suivante [12] :

- individus et interactions plutôt que processus et outils .
- développement logiciel plutôt que documentation exhaustive .
- collaboration avec le client plutôt que négociation contractuelle.
- ouverture au changement plutôt que suivi d'un plan rigide.

Grâce aux méthodes agiles, le client est pilote à part entière de son projet et obtient très vite une première mise en production de son logiciel. Ainsi, il est possible d'associer les utilisateurs dès le début du projet .

### 2.2.2 Structures opérationnelles et pratiques du développement agile

Toutes les méthodes agiles utilisent un mode opératoire similaire, voire identique [13] :

- Un responsable fonctionnel définit et ordonne la production des composants de l'application.
- Le projet est structuré en incréments de 1 à 6 semaines suivant les nécessités (taille, réactivité, visibilité...).
- Une réunion initiale organise chaque incrément en définissant les tâches à réaliser.
- L'équipe pilote la qualité et la performance du projet de manière consensuelle.

- Chaque jour, une courte réunion d'avancement donne à l'équipe une vision globale du projet, met en évidence les éventuels problèmes et permet de factoriser les solutions.
- Un 'reporting' mural (Kanban, graphes de progression, etc.) est mis à jour en temps réel par les membres de l'équipe.
- Un incrément achevé contient une livraison complète, développée, approuvée et testée.
- Une réunion finale présente l'application et est suivie d'une rétrospective technique du processus de développement.
- Le responsable fonctionnel valide le travail de l'équipe et ajuste les besoins entre chaque incrément .

### 2.2.3 Principes généraux de la modélisation agile

Il y a 12 principes généraux communs à toutes les méthodes agiles [w6] :

1. La plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à forte valeur ajoutée ;
2. Le changement est accepté, même tardivement dans le développement, car les processus agiles exploitent le changement comme avantage concurrentiel pour le client ;
3. La livraison s'applique à une application fonctionnelle, toutes les deux semaines à deux mois, avec une préférence pour la période la plus courte ;
4. Le métier et les développeurs doivent collaborer régulièrement et de préférence quotidiennement au projet ;
5. Le projet doit impliquer des personnes motivées. Donnez-leur l'environnement et le soutien dont elles ont besoin et faites leur confiance quant au respect des objectifs ;
6. La méthode la plus efficace de transmettre l'information est une conversation en face à face ;
7. L'unité de mesure de la progression du projet est un logiciel fonctionnel (ce qui exclut de comptabiliser les fonctions non formellement achevées) ;
8. Les processus agiles promeuvent un rythme de développement soutenable (afin d'éviter la non qualité découlant de la fatigue) ;
9. Les processus agiles recommandent une attention continue à l'excellence technique et à la qualité de la conception ;

10. La simplicité et l'art de minimiser les tâches parasites, sont appliqués comme principes essentiels ;
11. Les équipes s'auto-organisent afin de faire émerger les meilleures architectures, spécifications et conceptions ;
12. À intervalle régulier, l'équipe réfléchit aux moyens de devenir plus efficace, puis accorde et ajuste son processus de travail en conséquence ;

Une méthode qualifiée d'agile doit donc se composer d'un ensemble de pratiques instrumentant le cadre décrit par les 12 principes généraux agiles et en conséquence s'inscrire dans le respect des 4 valeurs fondamentales ayant inspiré le Manifeste agile.

### 2.2.4 Processus de modélisation d'une application web

Dans ce qui suit on n'utilisera pas les quatorze types de diagrammes proposées par UML, en utilisant particulièrement les diagrammes de cas d'utilisation, les diagrammes de séquence détaillé, diagramme de communication, diagramme de classe ainsi le diagramme de navigation pour la modélisation UML de notre application. Et c'est ce qu'assure la modélisation agile.

Pascal Roques [14] propose un schéma complet montrant comment, en partant des besoins des utilisateurs formalisés par des cas d'utilisation et une maquette, et avec l'apport des diagrammes de classes participantes, on peut aboutir à des diagrammes de conception à partir desquels on dérivera du code assez directement comme le montre la figure 2.2.

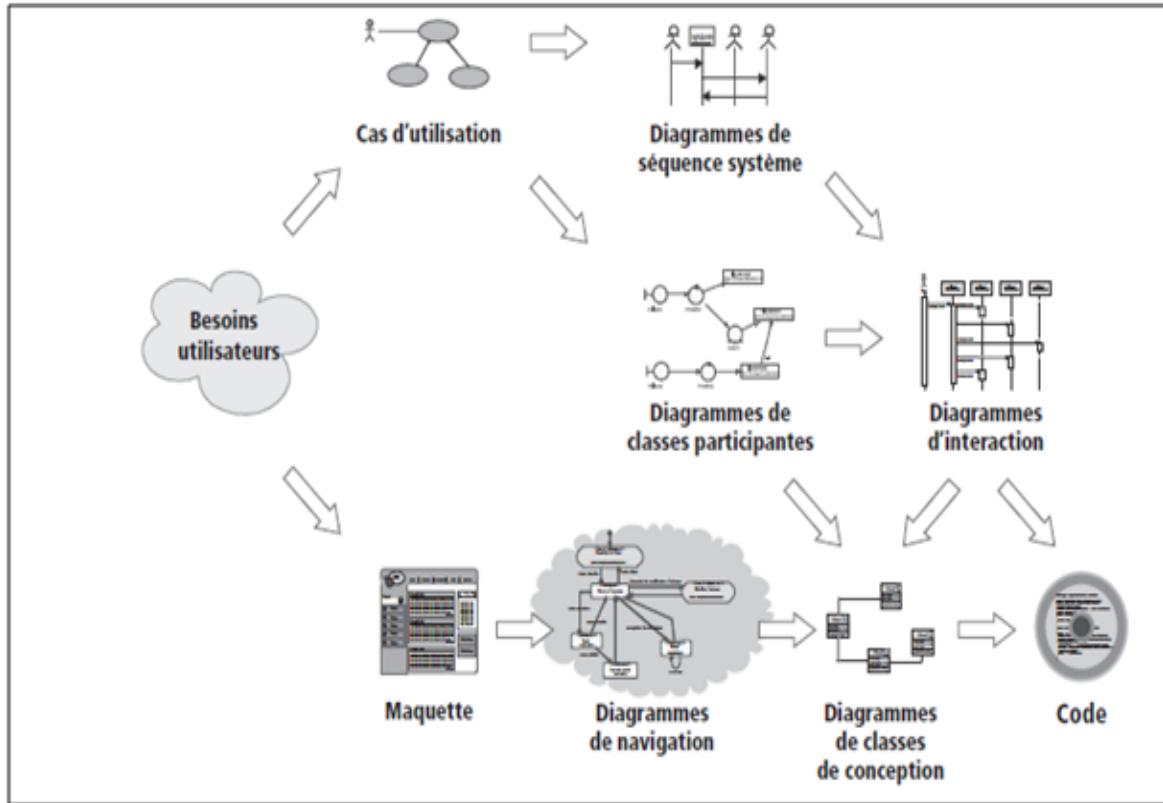


FIGURE 2.2 – Schéma complet du processus de modélisation d'une application web

### 2.2.5 Présentation d'UP, RUP et XP

La démarche agile que nous utiliserons, s'appuie essentiellement sur les méthodes de développement UP, RUP et XP :

#### 2.2.5.1 UP - Unified Process

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. UP gère le processus de développement par deux axes [w7].

**L'axe vertical** représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature.

**L'axe horizontal** représente le temps et montre le déroulement du cycle de vie du processus.

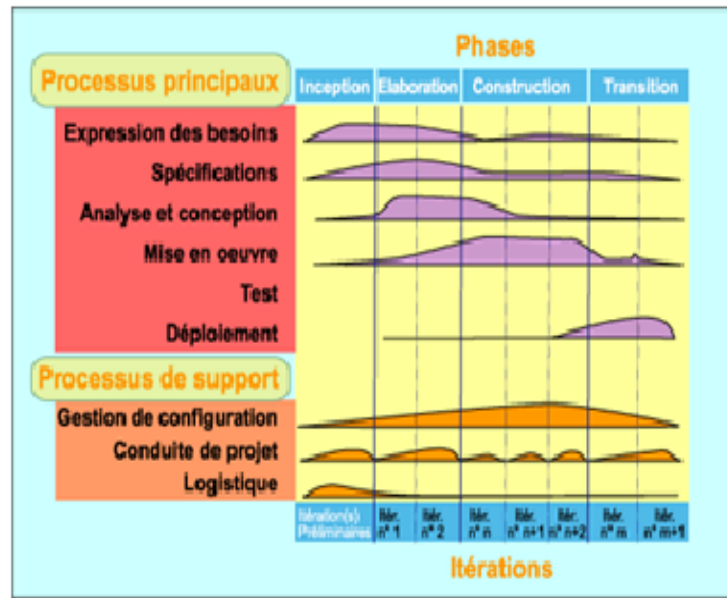


FIGURE 2.3 – La méthode de développement UP [w7]

### 2.2.5.2 RUP - Rational Unified Process

Le Rational Unified Process (RUP) est une méthode de développement commercialisé par Rational Software Corporation qui fournit les détails nécessaires à l'exécution des projets en utilisant l'UP.

### 2.2.5.3 XP - eXtreme Programming

La méthode XP est ensemble de bonnes pratiques de développement qui vise à remettre le développeur au centre du processus de développement. Ses prises de positions originales (et extrêmes) lui ont valu un succès d'estime depuis 4 ou 5 ans, surtout aux Etats-Unis. Voici les principaux éléments du fonctionnement de XP [14] :

- Gestion des livraisons : L'équipe fournit des livraisons fréquentes au client. Le contenu de ces livraisons est décidé par le client lui-même, à partir des estimations fournies par les développeurs.
- Gestion des itérations : Les livraisons sont réalisées en une suite d'itérations de 2 semaines environ, au sein desquelles le projet est géré à un niveau de détail plus fin.
- Suivi du projet : L'avancement du projet est mesuré de manière concrète par une batterie de tests de recette automatiques. Le rythme de progression est réévalué à chaque itération, et le plan de développement lui-même est revu fréquemment pour tirer parti de l'expérience acquise au cours du projet.
- Qualité du design et du code : Des pratiques strictes permettent de garder une vitesse de développement élevée tout au long du projet, tout en gardant une ouverture maximale au changement. La conception reste toujours le plus simple possible, le code est nettoyé en permanence et des tests unitaires de non-régression sont écrits pour chaque classe.
- Travail en équipe : L'équipe travaille réellement en équipe. Le code est partagé par tous, les développeurs travaillent systématiquement en binômes, et l'intégration est quasiment continue.

### 2.3 Présentation de l'entreprise

#### 2.3.1 Historique

BEJAIA LOGISTIQUE est située au pied de la montagne IFRI dans la commune d'Ouzellaguen. Elle est implantée à la zone industrielle Ahrik Ighzar Amokrane, dans la wilaya de Bejaia au nord-est de l'Algérie. Fondée en 2008, la Sarl Bejaia Logistique (BL) est la référence algérienne dans le domaine du transport routier. Elle bénéficie d'une image de qualité et d'une notoriété nationale. Ces activités sont étendues dans le transport public de marchandises, location d'engins et matériels de bâtiment, travaux publics et manutention, location de véhicules avec ou sans chauffeur et dans le transport des produits pétroliers.

### 2.3.2 Missions et activités de l'entreprise

- ▶ Les activités de la Sarl BL sont : transport public de marchandise, location d'engins, de véhicules avec ou sans chauffeur, du matériel pour bâtiments et travaux publics.
- ▶ L'entreprise a des missions très diverses liées aux activités qu'elle pratique tout les jours comme :
  - Assurer la sécurité des personnes et des biens en rapport avec les activités de la distribution.
  - Valider les programmes d'investissement.
  - Assurer la gestion (conduite, exploitation et maintenance) et le développement de ses réseaux d'activités.
  - Assurer la représentation de BL face à ses concurrents au niveau national.
  - Elargir son réseau d'activité en dehors du pays à des zones pas encore exploitées.

Nous allons maintenant entamer par une présentation du fonctionnement d'application de géolocalisation ensuite on passera à la spécification des besoins qui nous permettra de réaliser les diagrammes de cas d'utilisation.

### 2.3.3 Problématique

Comme nous l'avons précédemment, signalé Bejaia Logistique est une entreprise d'une notoriété nationale qu'exerce dans le domaine du transport routier. Notre choix de cette entreprise s'explique par le fait que lors des réunions qui ont lieu avec les dirigeants de BL, nous avons pu cerner un ensemble d'exigences pour les différents types d'utilisateurs. Ainsi que nous avons pu constater, durant ces réunions, que la gestion et le suivi en temps réel de l'ensemble des camions présente un certain nombre de difficultés qui se rapporte principalement à la surveillance de ces engins. Pour ce faire, nous avons pensé à développer un système de géolocalisation qui permet entre autres :

- Le suivi et la gestion en temps réel d'ensemble des camions en service.
- La localisation et la visualisation de ces derniers sur une carte géographique numérique.
- La conservation des traces de parcours et la possibilité d'alerter automatiquement les conducteurs au besoin.



### 2.3.4 Solutions proposées

On retrouve dans l'application de géolocalisation la visualisation des données et aussi un ensemble de paramètres qui permettent d'effectuer des recherches et un suivi des terminaux mobiles. Ces informations peuvent être projetées sur fond cartographique grâce aux données de géolocalisation. Voici une liste de fonctions typiquement offertes par les plateformes de géolocalisation professionnelles :

- Visualisation de la position de l'ensemble du parc géolocalisé suivi en temps réel de terminaux.
- Affichage d'un historique de déplacements.
- Création de points d'intérêt pour le traçage de chemins .
- Création de zones géographiques (pour le geofencing) et de routes (corridoring).
- Configuration d'alertes automatiques par courriel ou SMS sur des événements.
- Paramétrage d'événements (entrée/sortie de zone, dépassement de vitesse, temps d'arrêt...).

## 2.4 Spécification des besoins

Lors des différentes réunions qui ont lieu avec les dirigeants de BL, Nous avons pu établir un ensemble d'exigence pour les différents types d'utilisateurs

### 2.4.1 Exigences des utilisateurs

- Le Gestionnaire accède à son compte en communiquant son identifiant et son mot de passe.
- Le Gestionnaire peut voir la carte géographique et il peut faire des recherches des lieux.
- Le Gestionnaire peut visualiser la position de l'ensemble des terminaux dont il dispose de droit.
- Le Gestionnaire peut contacter l'administrateur via la messagerie.
- Le Gestionnaire peut voir l'historique des déplacements de l'ensemble des terminaux dont il dispose de droit.

### 2.4.2 Exigences des administrateurs

- L'informaticien accède à son espace d'administration en ligne en communiquant son identifiant et son mot de passe.
- L'informaticien gère une base de données de tous les clients qu'il remplira et actualisera.
- L'informaticien gère les zones d'alerte selon la demande des clients.
- L'informaticien peut ajouter, modifier et supprimer des terminaux pour chaque client.
- L'Informaticien peut ajouter des options pour améliorer le service selon le besoin du client.

## 2.5 Diagramme de cas d'utilisation

### 2.5.1 Définitions

**Diagrammes de cas d'utilisation [w8]** : il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.

**Acteur** : rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur participe à au moins un cas d'utilisation.

**Cas d'utilisation (use case)** : ensemble de séquences d'actions réalisées par le système produisant un résultat observable intéressant pour un acteur particulier. Collection de scénarios reliés par un objectif utilisateur commun.

**Association** : utilisée dans ce type de diagramme pour relier les acteurs et les cas d'utilisation par une relation qui signifie simplement " participe à ".

**Inclusion** : le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire, à un endroit spécifié dans ses enchaînements.

**Extension** : le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle, à un endroit spécifié indirectement dans celui qui procède à l'extension.

Après la concertation avec les responsables de l'entreprise nous avons recensé deux types d'acteurs qui sont :

L'administrateur : c'est le responsable informatique qui gérera l'application. L'utilisa-

teur : c'est le gestionnaire qui a un compte utilisateur et qui peut se connecter à son espace personnel pour la gestion de l'application.

### 2.5.2 Diagramme de cas d'utilisation côté d'administrateur

En partant des besoins de l'administrateur, on peut détecter les cas d'utilisation correspondant à l'acteur " administrateur " et qui sont représentés dans le diagramme de la figure 2.4 :

- L'administrateur doit s'identifier afin d'accéder à l'espace administrateur
- Les cas d'utilisation correspondant à l'acteur " administrateur " dans la figure 2.4 :

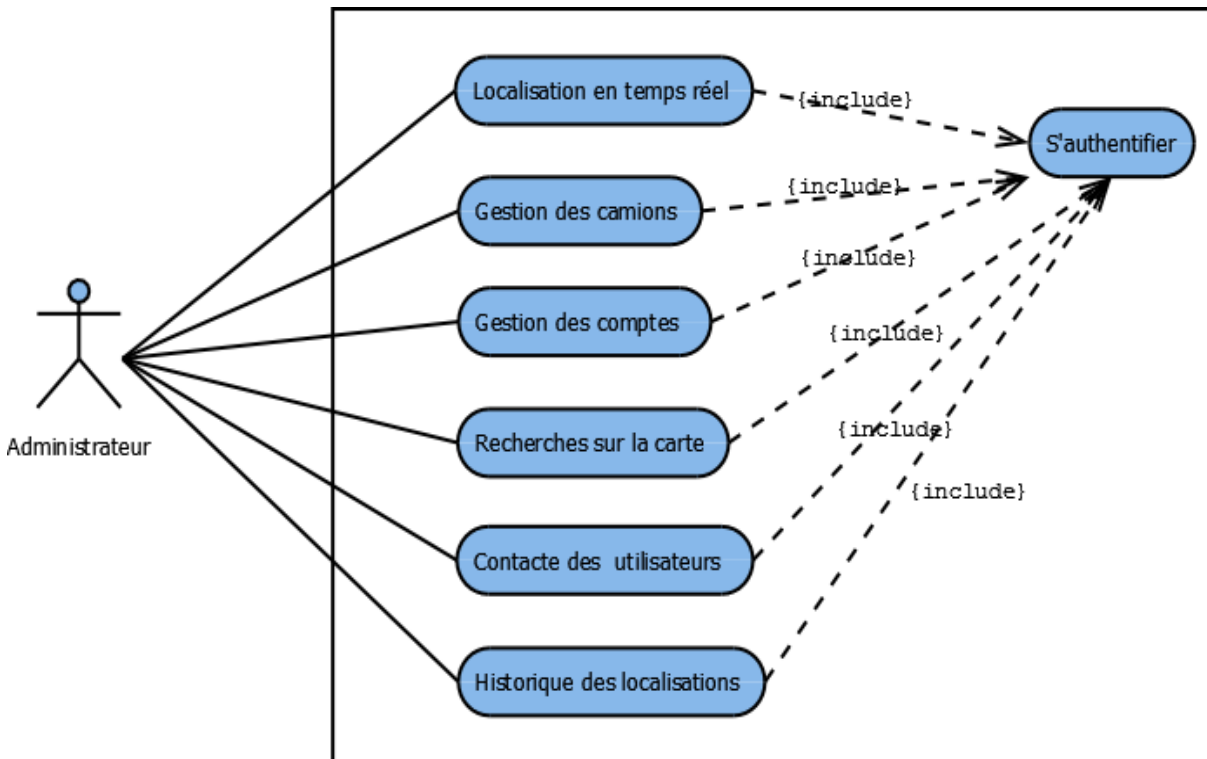


FIGURE 2.4 – Diagramme de cas d'utilisation d' administrateur

Le tableau 2.2 résume le diagramme de cas d'utilisation pour l'administrateur et l'utilisateur.

<b>cas d'utilisation</b>	<b>Acteurs</b>	<b>Messages émis/reçus</b>
Localisation en temps réel	Administrateur/ Utilisateur	Emis : Sélection d'un terminal Reçu : Une carte géographique avec un marker contenant les informations du camion
Gestions des camions	Administrateur	Emis : Ajouter/modifier/supprimer un camion Reçus : Liste des camions
Gestions des comptes	Administrateur	Emis : Ajoute/supprimer un compte Reçus : Liste des comptes
Recherche sur la carte	Administrateur/ Utilisateur	Emis : Saisir le lieu Reçus : Une carte géographique avec un marker contenant les informations du l'endroit
Contacteur l'utilisateur/administrateur	Administrateur	Emis : Envoi d'un message . Reçus : Accusé de réception

TABLE 2.2 – Identification des cas d'utilisation

Historique de localisation	Administrateur/ Utilisateur	Emis : Sélectionner d'un camion Reçus :  <ul style="list-style-type: none"><li>• Une carte géographique avec un marker contenant les informations de l'endroit</li><li>• Affichage des données sous forme d'un tableau</li></ul>
Voir la présentation	Utilisateur	Reçus : Présentation de l'entreprise

### 2.5.3 Diagramme de cas d'utilisation côté utilisateur

On peut constituer le diagramme de cas d'utilisation pour le gestionnaire qui est décrit comme suit :

- Le gestionnaire peut consulter la page d'accueil, consulter la présentation et contacter l'administrateur en lui envoyant un message, mais s'il veut consulter les autres sections une page apparaît pour lui demander de s'identifier, c'est-à-dire seules les personnes ayant un compte peuvent y accéder.
- Une fois qu'il s'est authentifié, le gestionnaire peut accéder aux autres.

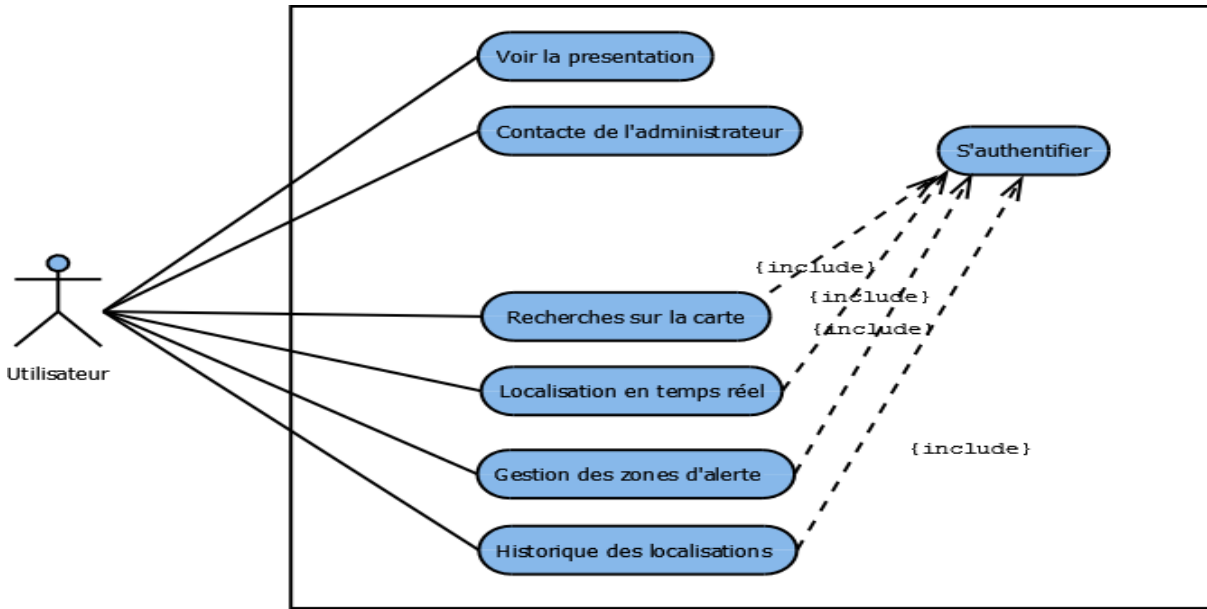


FIGURE 2.5 – Diagramme de cas d'utilisation pour le gestionnaire

#### 2.5.4 Maquette

Suivant les diagrammes de cas d'utilisation déjà réalisés et des besoins utilisateur et Administrateur, on peut déjà avoir une idée de ce que devrait être la maquette de notre application. La figure 2.6 illustre une première impression de ce que sera l'interface après la réalisation de l'application de géolocalisation.

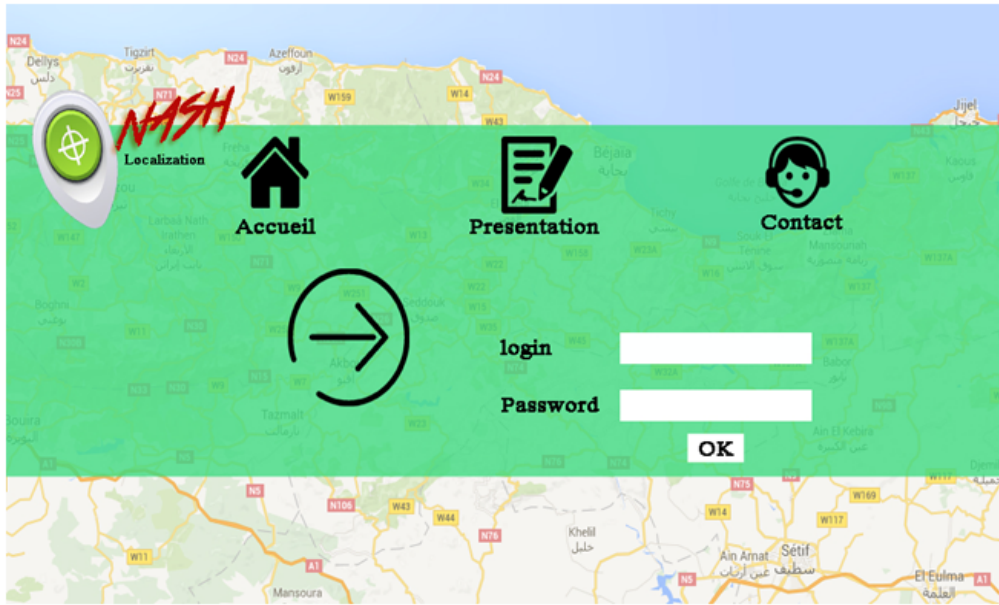


FIGURE 2.6 – Maquette de l'application de géolocalisation

## Conclusion

Dans ce chapitre nous avons présenté d'une manière générale notre cadre d'étude de l'entreprise Bejaia Logistique, qui subit des difficultés de gestions et de suivi des camions, leur mouvement et l'historique de déplacement de chacun des camions.

Nous avons également présenté UML d'une façon générale ainsi que la démarche agile et aussi la réalisation des diagrammes de cas utilisations.

Nous avons défini les besoins et les priorités pour faciliter la tâche au gestionnaire de BL. Cette étape nous servira de point de base dans l'élaboration de la prochaine étape, qu'est la spécification des besoins pour l'analyse et la conception de notre application de géolocalisation.

---

---

# CHAPITRE 3

---

## ANALYSE ET CONCEPTION

### Introduction

Ce chapitre sera consacré à l'analyse et la conception de la future application avec la réalisation des diagrammes UML de séquence, diagramme de communication et diagramme de navigation pour dégager le diagramme de classes afin de passer au modèle relationnel.

### 3.1 Diagramme de séquence

#### 3.1.1 Définitions

Les notations suivantes sont utilisées lors de la réalisation des diagrammes de séquence [15].

- **Diagramme de séquence**

Il montre la séquence verticale des messages passés entre objets au sein d'une interaction.

- **Rôle**

Montre la séquence verticale des messages passés entre objets au sein d'une interaction.



- **Ligne**

de vie représentation de l'existence d'un élément participant dans un diagramme de séquence. Cela peut être un acteur ou le système en modélisation d'exigences, des objets logiciels en conception préliminaire ou conception détaillée.

- **Message**

Élément de communication unidirectionnel entre objets qui déclenche une activité dans l'objet destinataire. La réception d'un message provoque un événement dans l'objet récepteur. La flèche pointillée représente un retour au sens UML. Cela signifie que le message en question est le résultat direct du message précédent.

- **Spécification d'activation**

Bande bleu qui représente une période d'activité sur une ligne de vie.

- **Message synchrone**

Envoi de message pour lequel l'émetteur se bloque en attente du retour et qui est représenté par une flèche.

- **Occurrence d'interaction**

Une interaction peut faire référence explicitement à une autre interaction grâce à un cadre avec le mot-clé **ref** et indiquant le nom de l'autre interaction.

- **Les cadres d'interaction** Chaque cadre possède un opérateur et peut être divisé en fragments. Les principaux opérateurs sont :

- **loop** : boucle. Le fragment peut s'exécuter plusieurs fois, et la condition de garde explicite l'itération.
- **opt** : optionnel. Le fragment ne s'exécute que si la condition fournie est vraie.
- **alt** : fragments alternatifs. Seul le fragment possédant la condition vraie s'exécutera.

### 3.1.2 Réalisation des diagrammes de séquences

Nous allons maintenant présenter quelques diagrammes de séquence qui sont les plus utiles pour la suite de notre travail.

### 3.1.2.1 Diagramme de séquence pour l'authentification

L'authentification se fait de la manière suivante : L'utilisateur demande l'accès au système (demande la page authentification). Le système affiche le formulaire, l'utilisateur le remplit par le pseudo et le mot de passe une fois validés, le système vérifie l'existence de l'utilisateur et trois cas sont possibles :

- Les données sont correctes donc, la page correspondante s'affiche pour l'utilisateur.
- Champs obligatoires non valides et/ou vides, le système affiche un message d'erreur champs obligatoire .
- Les données sont incorrectes donc, un message d'erreur apparait et réaffichage du formulaire d'authentification.

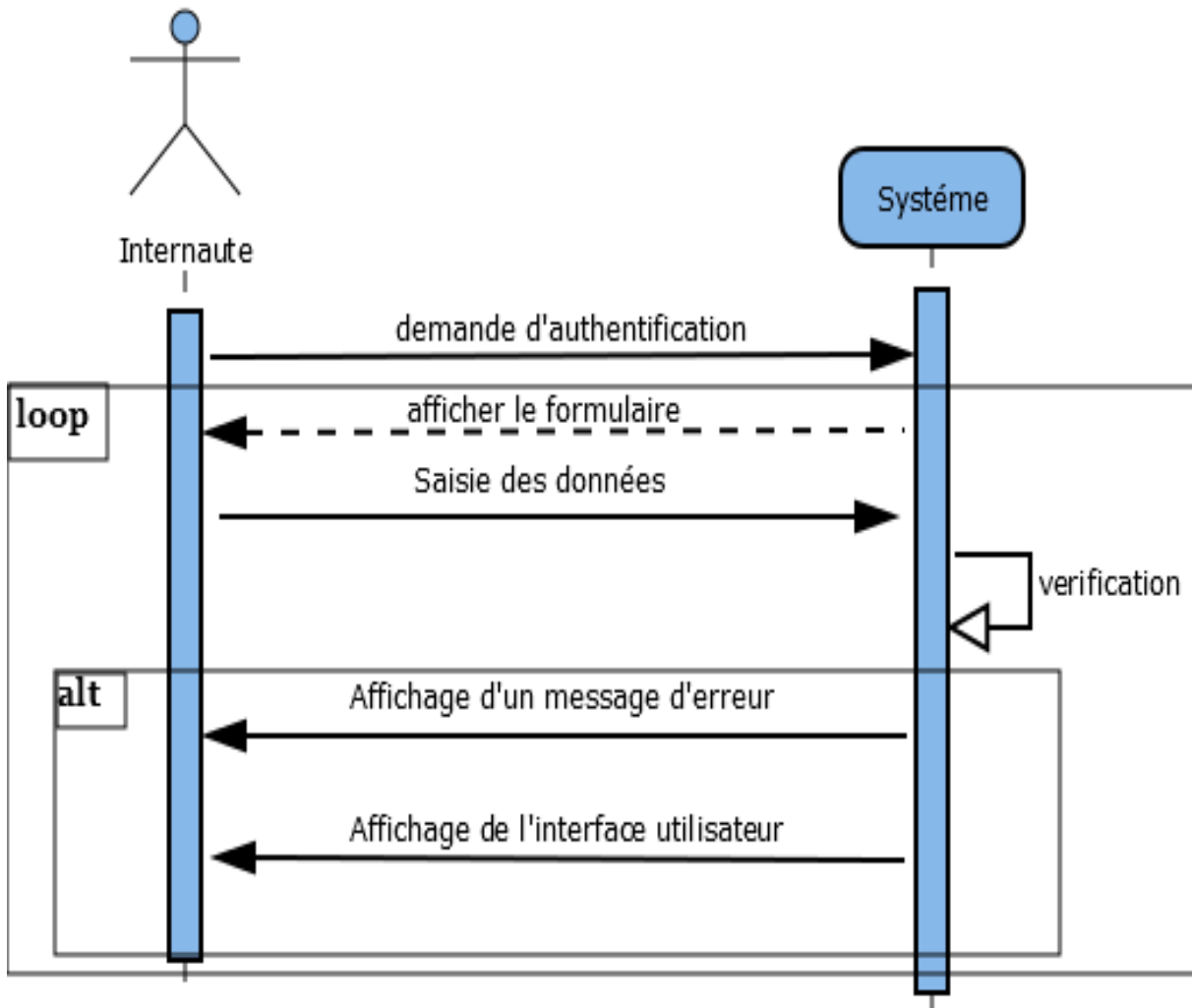


FIGURE 3.1 – Diagramme de séquence pour l'authentification

### 3.1.2.2 Diagramme de séquence pour la géolocalisation en temps réel

Pour la géolocalisation en temps réel, une fois que l'utilisateur est authentifié, il demande au système d'afficher la liste des camions, le système répond en chargeant les données et en envoyant une page d'affichage de la ressource demandée .

L'utilisateur sélectionne un terminal puis le système affiche une carte avec un marker représentant le camion sélectionné.

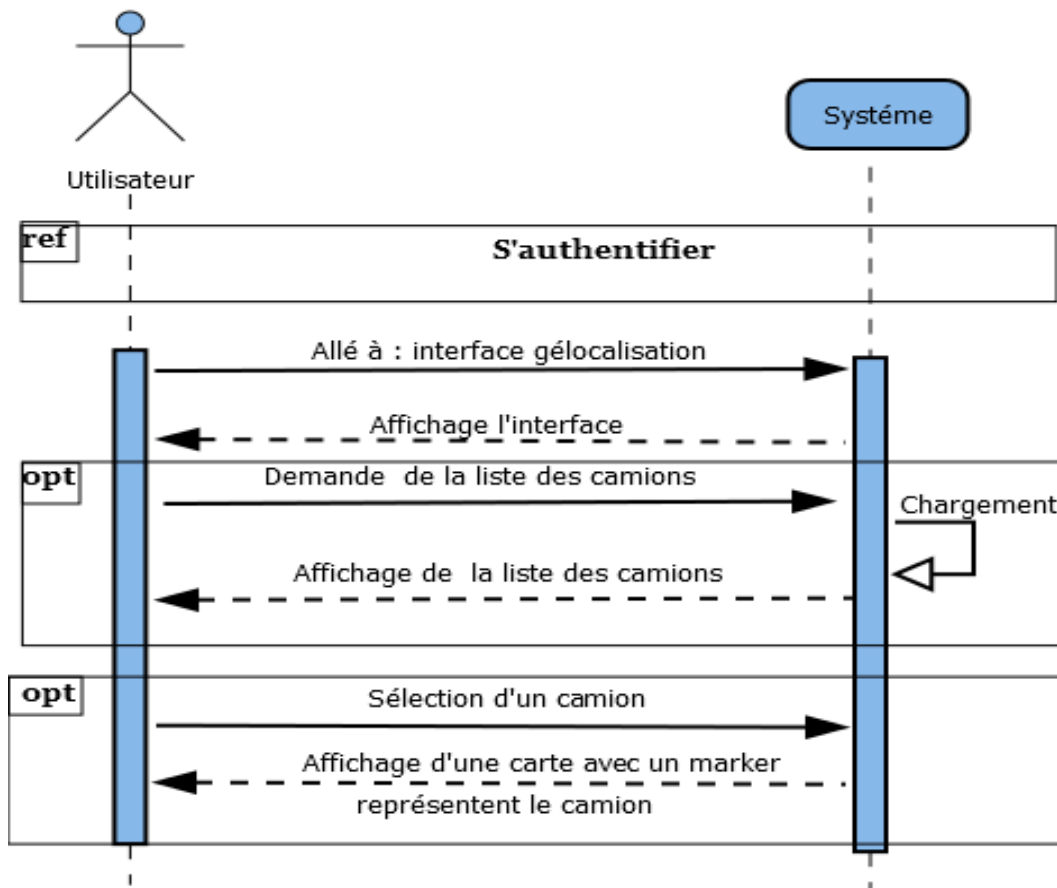


FIGURE 3.2 – Diagramme de séquence pour la géolocalisation en temps réel

### 3.1.2.3 Diagramme de séquence pour contacter l'utilisateur

Pour contacter l'utilisateur l'administrateur devra s'authentifier. Une fois l'authentification a eu lieu, un formulaire s'affichera pour saisir le message et l'envoi au système. Lorsque le message est enregistré, un accusé de réception sera envoyé à l'utilisateur.

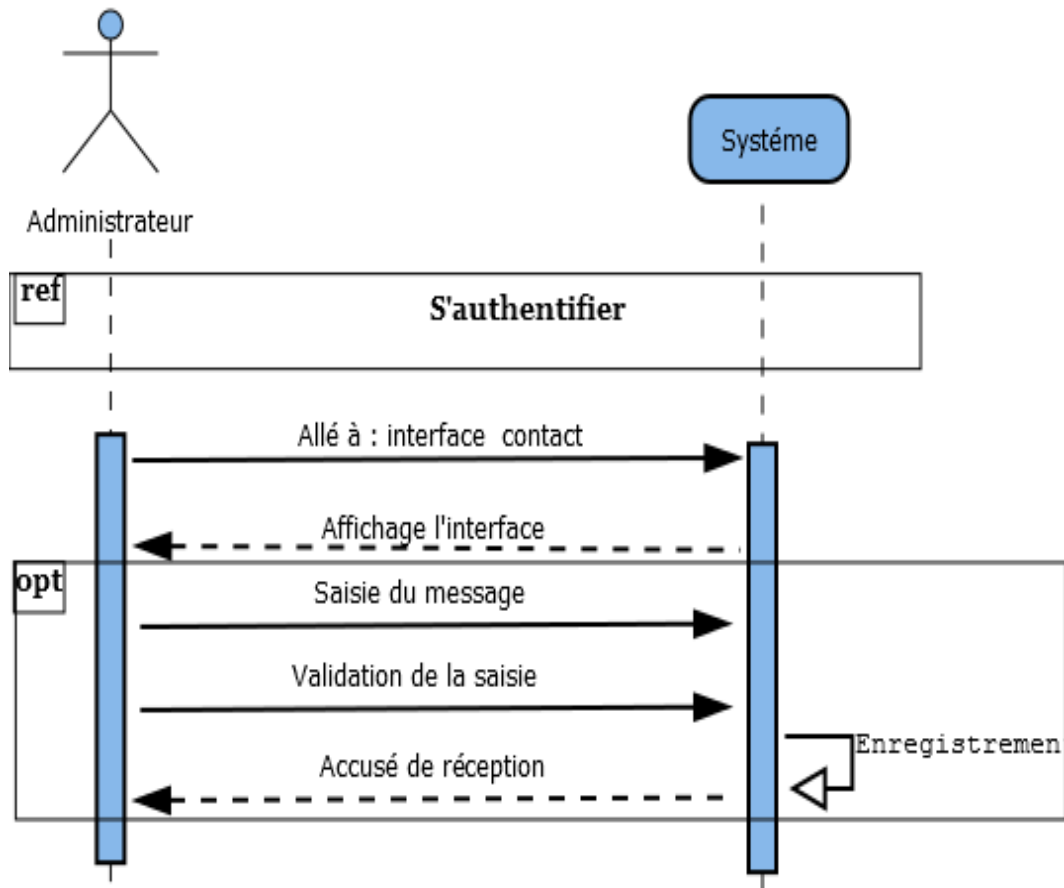


FIGURE 3.3 – Diagramme de séquence pour contacter l'utilisateur

#### 3.1.2.4 Diagramme de séquence pour consulter l'historique de localisation

Une fois que l'utilisateur est authentifié. Il se présente devant l'interface de consultation de l'historique, puis il demande la liste des camions, le système répond avec une liste de camions. L'utilisateur sélectionne un camion et fixe un intervalle temporel pour la consultation. Deux options sont offertes pour affichage :

- Affichage de l'historique de la localisation du camion sous forme d'un tableau.
- Ou, Affichage de l'historique de la localisation du camion sur une carte géographique.

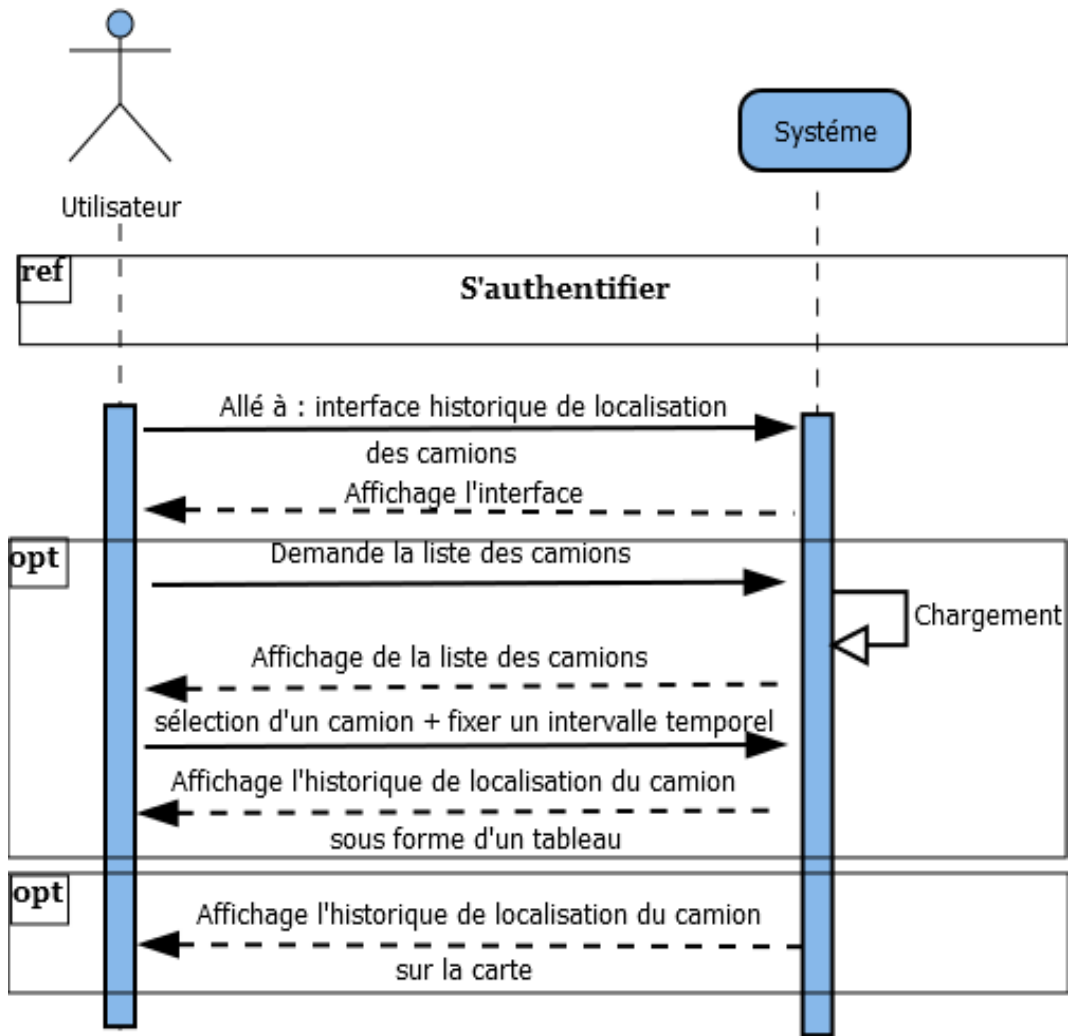


FIGURE 3.4 – Diagramme de séquence pour consulter l’historique de localisation

### 3.1.2.5 Diagramme de séquence pour les recherches sur la carte

L'utilisateur visualise la carte via son espace utilisateur. Il saisit le lieu puis au système demande de le localiser. Le système répond par l'affichage de la position de l'endroit sur la carte.

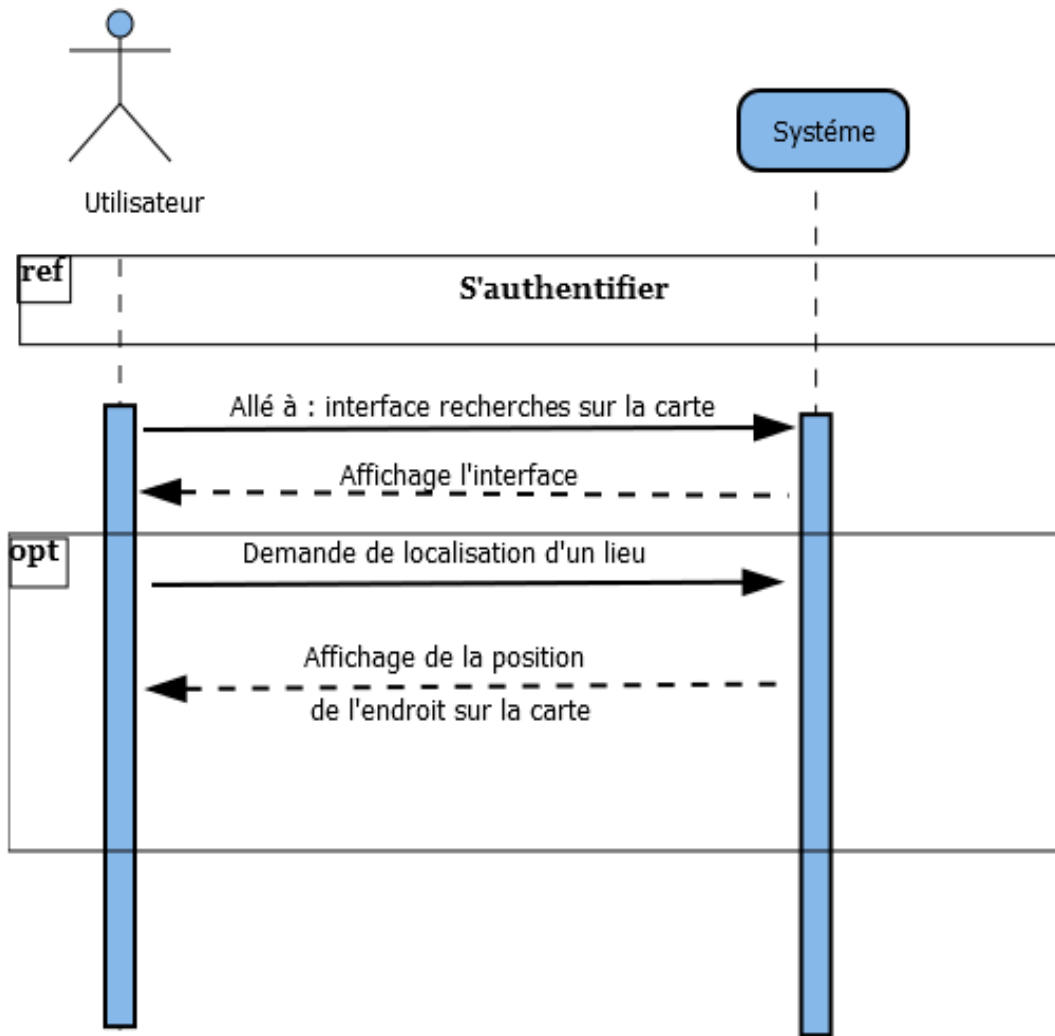


FIGURE 3.5 – Diagramme de séquence pour consulter les recherches sur la carte

### 3.1.2.6 Diagramme de séquence pour la gestion des zones d’alertes

L'utilisateur gère les zones d'alertes via son espace d'utilisation (après l'authentification). Il a trois options à effectuer :

- **l'ajout d'une zone d'alerte** : il demande à ajouter une zone d'alerte, le système lui envoie un formulaire pour saisir la zone d'alerte.
- **la modification d'une zone d'alerte** : la zone s'affiche et l'utilisateur peut modifier la zone.
- **la suppression** : la suppression se fait de manière simple, l'utilisateur supprime

directement la zone sélectionnée.

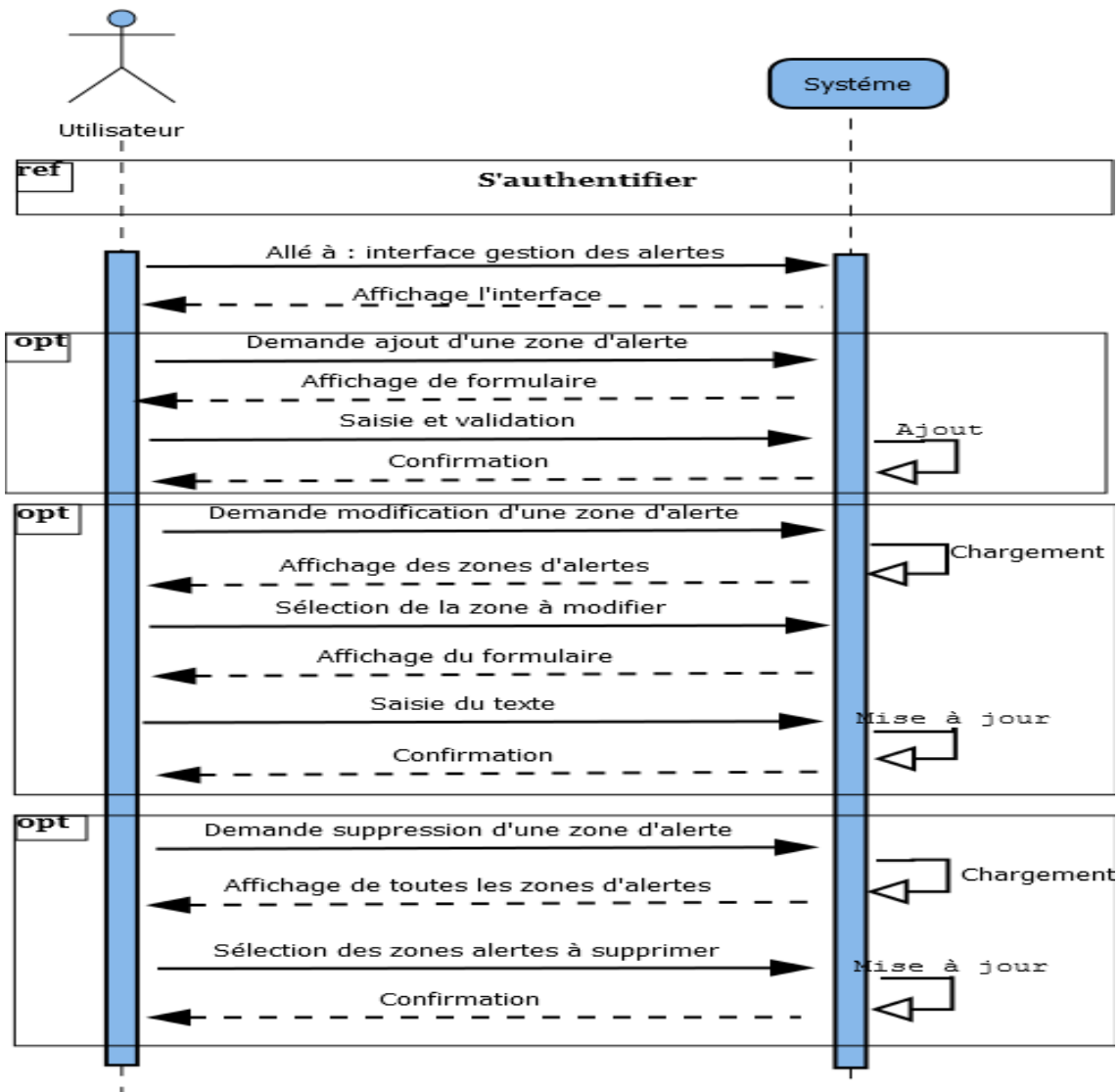


FIGURE 3.6 – Diagramme de séquence pour la gestion des alertes

## 3.2 Diagrammes de communications

### 3.2.1 Définitions

Pour la représentation des diagrammes d'interaction nous commençons par définir quelques notions [13].

#### Dialogue

Dialogue permet l'interaction entre les objets du système. Il s'agit typiquement des écrans proposés à l'utilisateur : les formulaires de saisie, les résultats de recherche, etc.

**Contrôle**

Il fait la transition entre les dialogues et les concepts du domaine, en permettant aux écrans de manipuler des informations détenues par des objets.

**Entité**

Elle permettra à des données et des relations d'être stockées dans des fichiers ou des bases de données.

**3.2.2 Réalisation des diagrammes de communication**

**3.2.2.1 Diagramme d'authentification**

Ce diagramme décrit les messages échangés entre les différents objets pour montrer le fonctionnement de l'opération d'authentification : le gestionnaire saisit le pseudo et le mot de passe puis le contrôleur authentification vérifie sa validité. L'application affiche soit le résultat soit un message d'erreur ou crée une interface valide.

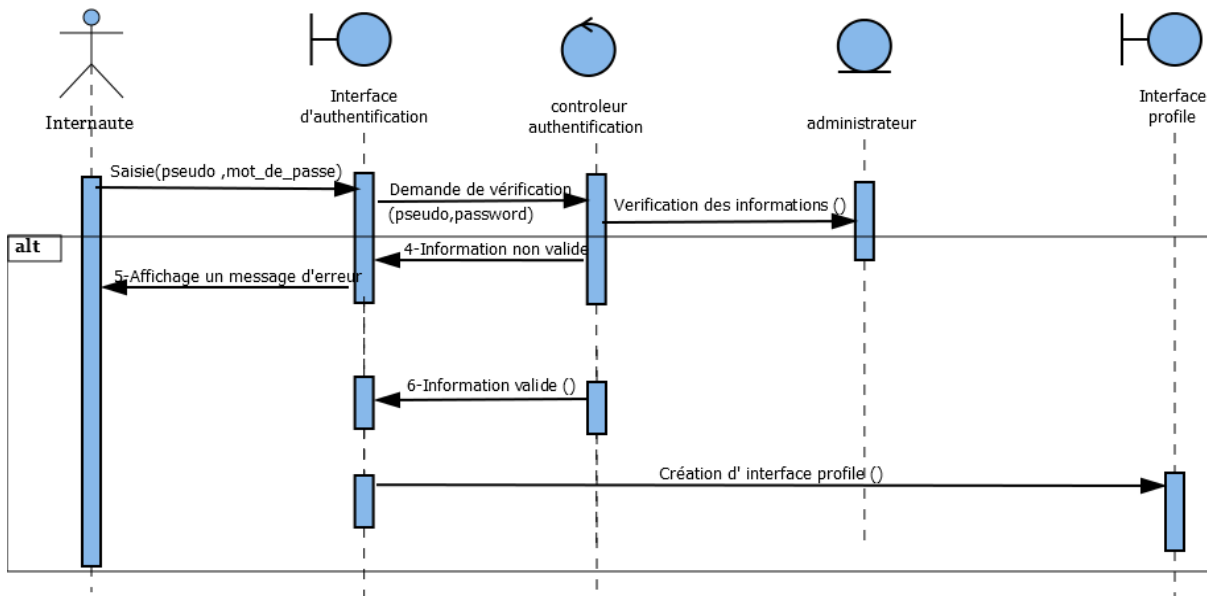


FIGURE 3.7 – Diagramme de communication d'authentification



### 3.2.2.2 Diagramme de communication pour géolocalisation en temps réel

Le premier dialogue est la sélection d'un camion à localiser par l'utilisateur, le contrôleur demande ces informations auprès de l'entité géolocalisation.

L'interface charge la carte GoogleMaps avec un marker représentant les informations du camion.

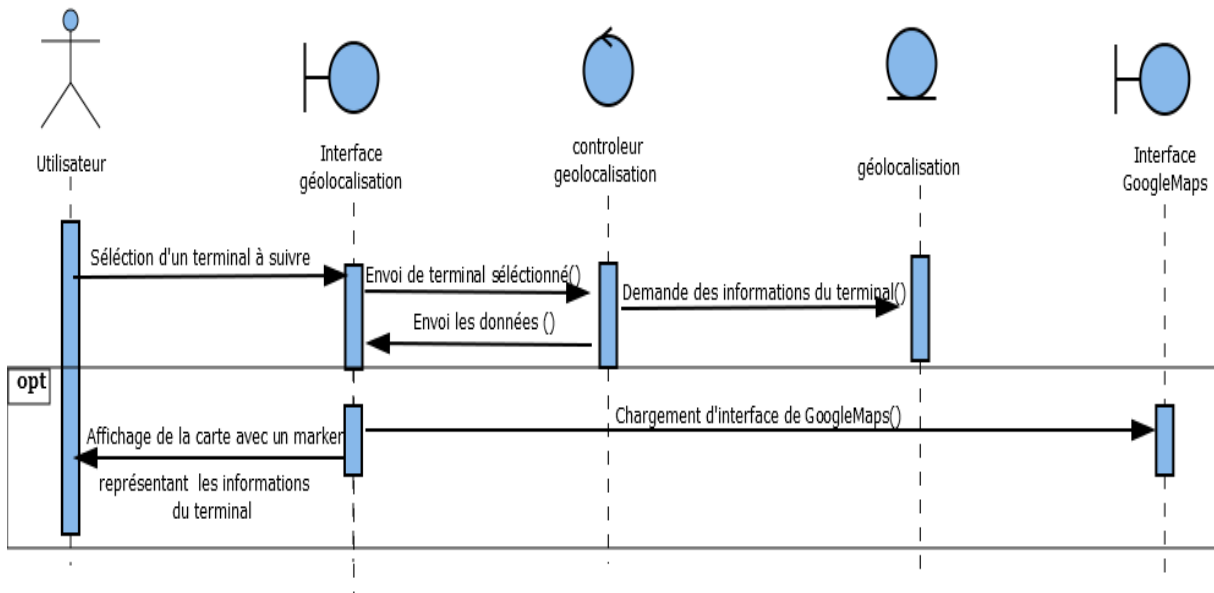


FIGURE 3.8 – Diagramme de communication pour géolocalisation en temps réel

### 3.2.2.3 Diagramme de communication pour effectuer des recherches sur la carte

Pour effectuer une recherche sur la carte il faut activer l'interface recherche, une zone de texte apparait pour la saisie (nom) de l'endroit. Une fois que la saisie est validée, le résultat sera affiché sur une carte géographique (GoogleMaps).

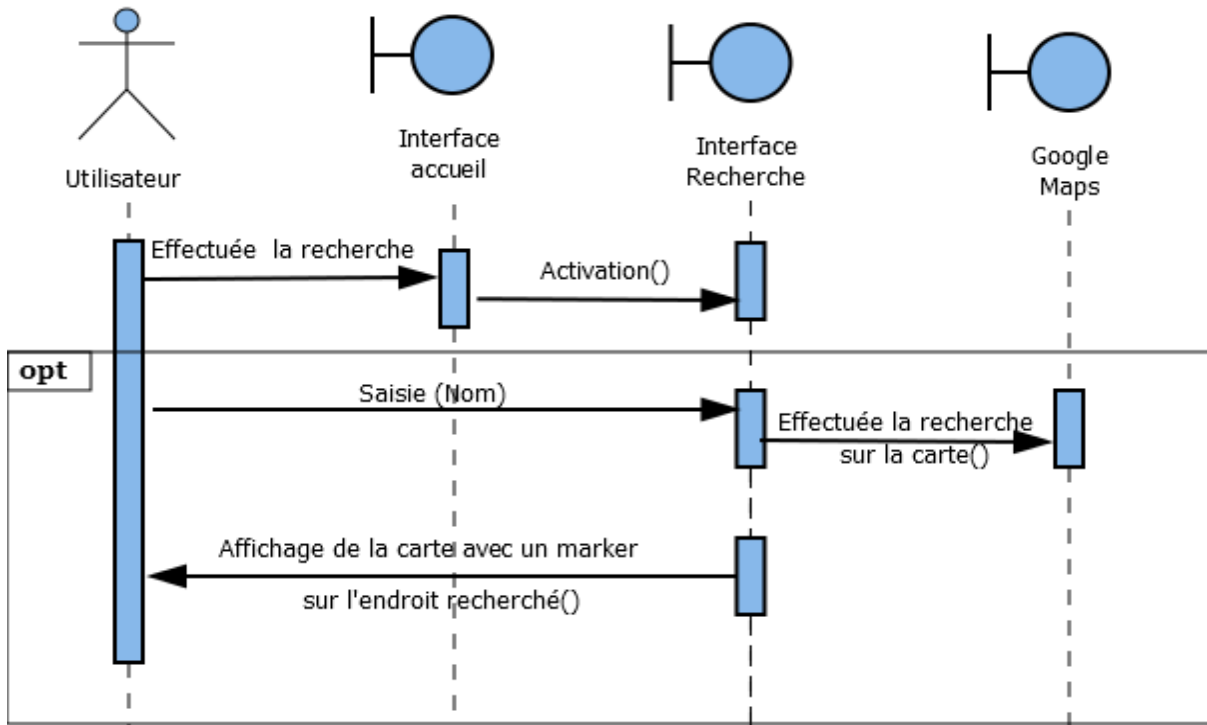


FIGURE 3.9 – Diagramme de communication pour effectuer des recherches sur la carte

### 3.2.2.4 Diagramme communication pour consulter l'historique de localisation

Le gestionnaire demande l'ouverture d'une interface d'historique, après l'activation de cette interface le gestionnaire sélectionnera un camion, deux cas sont possibles pour l'affichage :

- L'affichage sur un tableau qui contiendra toute les informations sur le camion sélectionné.
- Ou, l'affichage du l'historique sera sur une carte GoogleMaps.

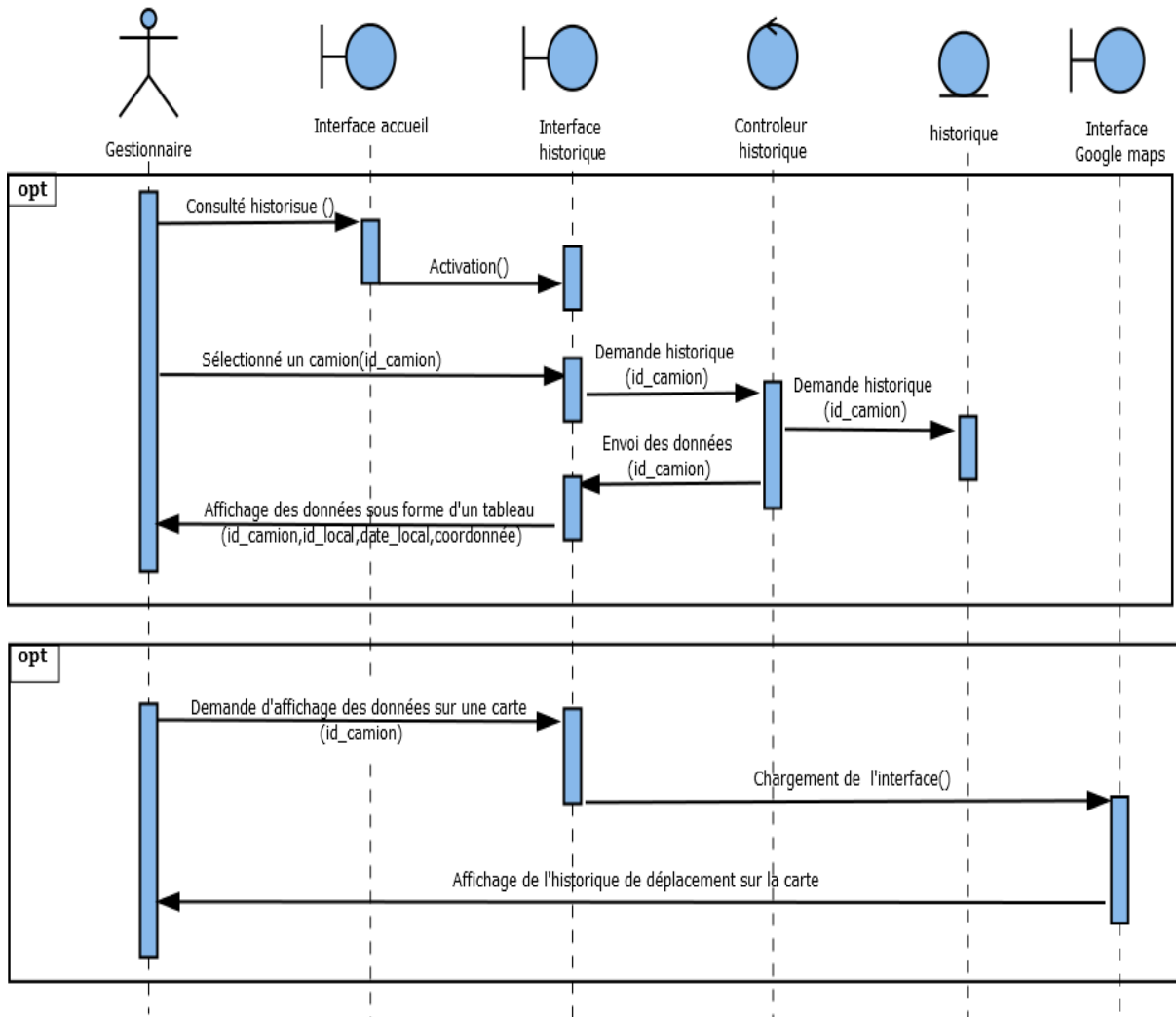


FIGURE 3.10 – Diagramme de communication pour consulter l’historique de localisation

### 3.2.2.5 Diagramme de communication pour la gestion des zones alertes

Les trois scénarios possibles de diagramme d’interaction pour la gestion des zones d’alertes sont :

#### Ajout d’une zone d’alerte

Cette opération commence lorsque le gestionnaire demande au système de créer une zone d’alerte. Il saisit les informations (nom\_zone, coordonnées, rayon) attachées à un zone d’alerte.

Les informations sont initialisées par l’interface et seront envoyées au contrôleur, qui envoie à son tour des informations à l’entité zone alerte.

### **Modification d'une zone d'alerte**

Le gestionnaire débute par une demande de modification au système (interface d'accueil) puis il choisit une zone d'alerte, l'interface envoie le `id_zone` au contrôleur pour récupérer les données dans l'entité concernée. Le contrôleur répond, l'interface à son tour affiche les informations demandées au gestionnaire.

L'utilisateur met à jour les informations correspondantes de la zone sélectionnée (`nom_zone`, `coordonnées`, `rayon`), les modifications seront enregistrées dans l'entité zone alerte.

### **Suppression d'une zone alerte**

Le gestionnaire active l'interface zone alerte par une demande de suppression. Lorsque cette demande est effectuée, il choisit une zone à supprimer, le contrôleur envoie une requête de suppression qui est identifiée par `id_zone` à la zone d'alerte pour exécuter la requête de suppression.

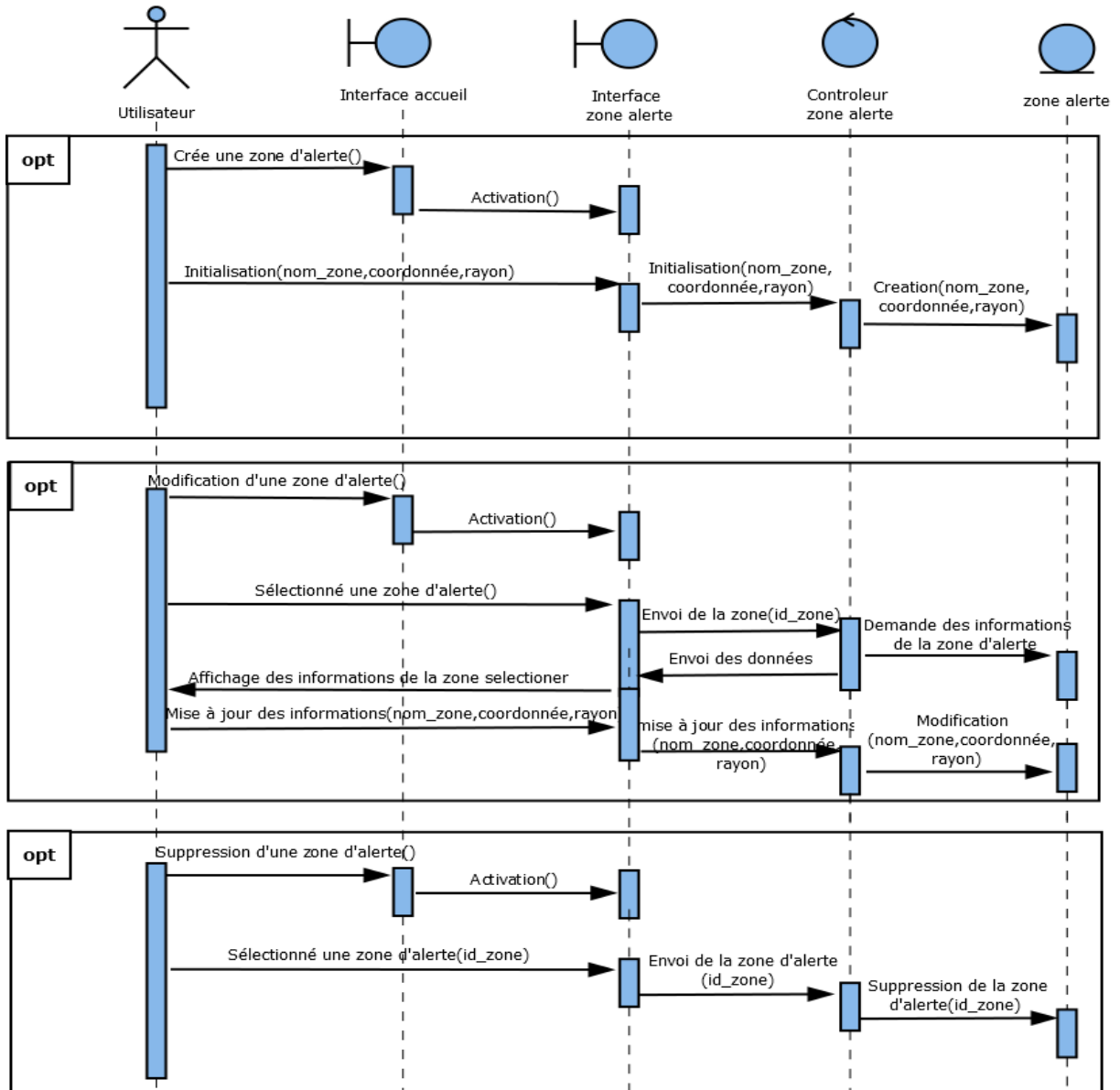


FIGURE 3.11 – Diagramme de communication pour la gestion des zones d’alertes

A présent on va présenter le diagramme de navigation qui illustre les différents liens qu’ont les pages de l’application entre elles.

### 3.3 Diagramme de navigation

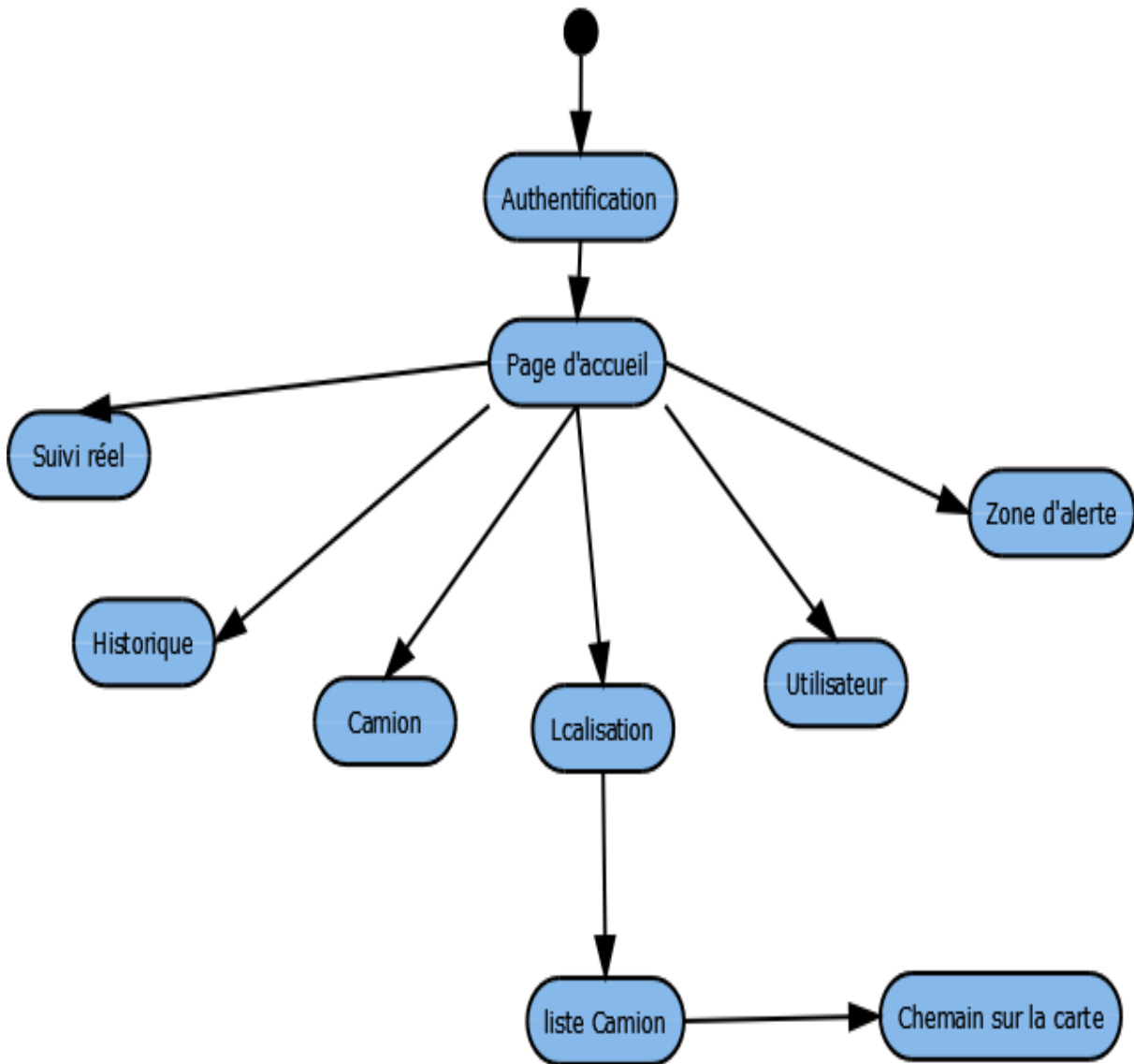


FIGURE 3.12 – Diagramme de navigation

D'autres possibilités de navigation existent comme, l'utilisateur peut aller de n'importe quelle rubrique à partir de la page d'accueil. nous verrons maintenant quelques concepts fondamentaux pour la réalisation du diagramme de classe.

## 3.4 Diagrammes de classes

### 3.4.1 Définitions

**Diagramme de classe :** Diagramme structurel qui décrit un ensemble de classe, d'interfaces et leurs relations [16].

**Classe :** description abstraite d'un ensemble d'objets qui partagent les mêmes propriétés (attributs et associations) et comportements (opérations et états) . **Attribut :** donnée déclarée au niveau d'une classe, éventuellement typée, à laquelle chacun des objets de cette classe donne une valeur. Un attribut peut posséder une multiplicité et une valeur initiale. Un attribut dérivé (/) est un attribut dont la valeur peut être déduite d'autres informations disponibles dans le modèle .

**Opération :** élément de comportement des objets, défini de manière globale dans leur classe. Une opération peut déclarer des paramètres (eux-mêmes typés) ainsi qu'un type de retour .

**Association :** relation sémantique durable entre deux classes, qui décrit un ensemble de liens entre instances. Une association est bidirectionnelle par défaut, sauf si l'on restreint sa navigabilité en ajoutant une flèche . Rôle : nom donné à une extrémité d'une association ; par extension, manière dont les instances d'une classe voient les instances d'une autre classe au travers d'une association.

**Multiplicité :** le nombre d'objets (min..max) qui peuvent participer à une relation avec un autre objet dans le cadre d'une association. Multiplicités fréquentes :

- 0..1 = optionnel (mais pas multiple)
- 1 = exactement 1
- 0..\* = \* = quelconque
- 1..\* = au moins 1

Catégorisation des classes d'analyse en trois catégories qui a été proposée par I. Jacobson et popularisée ensuite par le RUP (Rational Unified Process)

- Celles qui permettent les interactions entre le site web et ses utilisateurs sont qualifiées de " dialogues ". Il s'agit typiquement des écrans proposés à l'utilisateur.

TABLE 3.2 – Dictionnaire de données

- Les classes qui contiennent la cinématique de l’application sont appelées ” contrôles ”. Elles font la transition entre les dialogues et les concepts du domaine, et contiennent les règles applicatives.
- Celles qui représentent les concepts métier sont qualifiées d’ ” entités ”. Elles sont très souvent persistantes.

### 3.4.2 Dictionnaire de données

Attributs	Désignation	Type	Taille
Id	Numéro de camion	int	11
Mat_cam	Matricule de camion	varchar	25
Type_cam	Type de camion	varchar	25
Num_ter	Numéro de série de terminal	varchar	25
Id_conducteur	Identifiant de conducteur	int	11
Nom_c	Nom de conducteur	varchar	25
Prenom_c	Prenom de conducteur	varchar	25
Date_c		date	
Id_l	identifiant de localisation	int	11
Nom_l	Nom de localisation	varchar	25
Date_l	Date de localisation		
Longitude	la longitude d’une localisation	double	
Latitude	la latitude d’une localisation	double	
Id_zone	Identifiant de zone d’alerte	int	11
coordonnées	Coordonnées d’une zone d’alerte	varchar	25
Rayon	rayon d’une zone d’alerte	double	25
date_debut	debut d’une mission	date	
date_fin	fin d’une mission	date	



### 3.4.3 Réalisation du diagramme de classes

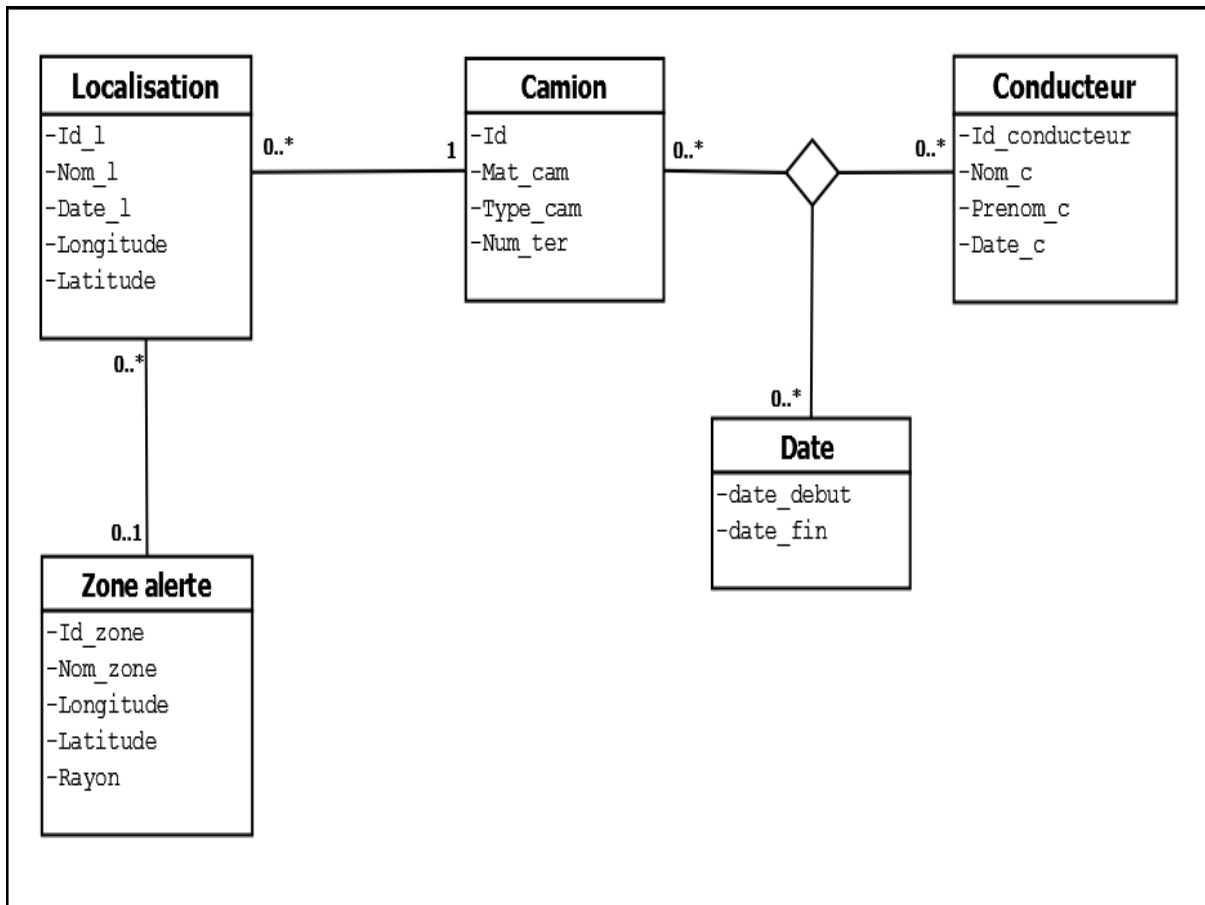


FIGURE 3.13 – Diagramme de classes

## 3.5 Modèle relationnel de données

### 3.5.1 Définition

Quelques définitions sur le modèle relationnel de données [17] :

- **Domaine** : c'est l'ensemble des valeurs d'un attribut .
- **Relation** : c'est un sous ensemble du produit cartésien d'une liste de domaines. C'est en fait un tableau à deux dimensions dont les colonnes correspondent aux Domaines et dont les lignes contiennent des tuples. On associe un nom à Chaque colonne .
- **Attribut** : c'est une colonne d'une relation, caractérisé par un nom .
- **Tuple** : c'est la liste des valeurs d'une ligne d'une relation .

- **Cardinalité** : elle permet de définir les conditions de participation d'une entité à une relation. Toutefois, une entité peut participer à plusieurs relations .
- **L'arité** : est le nombre d'attributs d'une relation .
- **Clé** : On distingue deux types de clés :
  - **Clé primaire** : ensemble d'attributs dont les valeurs permettent de distinguer les nuplets les uns des autres (notion d'identifiant).
  - **Clé étrangère** : Attribut qui est clé primaire d'une autre entité.

### 3.5.2 Les règles de passage au relationnel

A partir du diagramme de classes que nous avons effectué, on peut réaliser le modèle relationnel suivant les règles de passage suivantes [15] :

1. Transformation des classes : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.
2. Transformation des associations : Nous distinguons trois familles d'associations.
  - Association 1..\* : 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 .
  - Association \*.\* et n-aire et classes- association : la classe-association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association .
  - Association 1..1 : il faut ajouter un attribut de type 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 la classe connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule [4].

### 3.5.3 Application

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

**Zone.alerte** ( Id\_zone, Nom\_zone,longitude,latitude,Rayon)

**Camion** ( *Id\_camion*, *Mat\_Camion*, *Type\_Cam*, *Num\_ter*, *#id\_Condicteur*)

**Localisation**(*Id\_L*, *Nom\_l*, *date\_l*, *longitude*, *latitude*, *#Id*)

**Conducteur**(*Id\_conducteur*, *#Nom\_c*, *#Prenom\_c*, *Date\_c*)

**Date**(*date\_debut*, *date\_fin*)

**DCC**(*#Id\_camion*, *#Id\_conducteur*, *#date\_debut*, *#date\_fin*)

## Conclusion

Tout au long de ce chapitre, nous avons détaillé la conception de notre application à travers les diagrammes de séquence, les diagrammes de communication, le diagramme de classes et aussi les règles de passage vers le modèle relationnel est associés afin que la phase réalisation et la mise en place de l'application soient plus aisées.

---

---

# CHAPITRE 4

---

## RÉALISATION

### Introduction

Dans ce quatrième et dernier chapitre, on s'intéressera à la réalisation de notre application de la géolocalisation des camions de Bejaia Logistique. Nous allons tout d'abord présenter les différents outils et langages utilisés pour la réalisation. Ensuite, nous présenterons les différentes fonctionnalités offertes par le système et cela sera illustré par la présentation des différentes interfaces de l'application.

### 4.1 Langages de programmation

#### 4.1.1 Langage HTML5

HTML (HyperText Markup Language)[w9] a fait son apparition dès 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu des pages web. C'est donc en HTML que vous écrirez ce que vous souhaitez que la page affiche du texte, des liens, des images. Nous l'utiliserons pour l'élaboration des pages statiques qui constituent l'interface utilisateur.

### 4.1.2 Langage CSS

CSS (Cascading Style Sheets, aussi appelées Feuilles de style)[18] joue un rôle important dans la gestion et l'apparence de la page web (agencement, positionnement, décoration, couleur et taille du texte). Nous utiliserons ce langage pour rendre les pages programmées en HTML plus agréables et leurs donner un style spécifique.

### 4.1.3 Langage java

Java [19] est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitations (Windows, Linux, Macintosh). Nous utiliserons ce langage car il offre un ensemble de bibliothèques qui facilitent le développement web et dispose d'une documentation bien fournie.

### 4.1.4 Langage java script

JavaScript [20] est un langage de programmation de scripts principalement employé dans les pages web. Par exemple, JavaScript permet de tester que les champs obligatoires sont bien remplis sans avoir besoin d'envoyer le formulaire au serveur pour qu'il le vérifie. L'avantage dans ce cas là est de limiter le nombre de requêtes envoyées au serveur .

### 4.1.5 Langage PHP

PHP (Hypertext Preprocessorest)[21] est un langage de script du côté serveur . Sa syntaxe est largement inspirée du langage C et de Java , avec des améliorations spécifiques. Le but du langage est d'écrire rapidement des pages HTML dynamiques. Nous utiliserons PHP, pour élaborer les pages dynamiques de notre application.

### 4.1.6 Langage SQL

SQL (Structured Query Language c'est-à-dire Langage d'interrogation Structuré) est un langage standard des systèmes de gestion de bases de données relationnelles. Il a été conçu par IBM dans les années 70. Le langage SQL nous permettra d'élaborer les requêtes d'interrogation de la base de données.

### 4.1.7 Langage XML

XML ( eXtensible Markup Language )[\[w10\]](#) est un langage HTML amélioré permettant de définir de nouvelles balises (markup) et la mise en forme des documents .Le XML se veut également compatible avec le web afin que les échanges de données puissent se faire facilement à travers le réseau Internet.

## 4.2 Environnements de développement

### 4.2.1 Wampserver

Wampserver est un serveur d'évaluation qui regroupe tous les éléments indispensables à l'évaluation d'un site web dynamique (serveur web APACHE, serveur de bases de données MySQL et PhpMyAdmin) :

#### 4.2.1.1 Apache

Apache est un serveur web gratuit extensible, modulaire, configurable, robuste et performant pour des besoins matériels modeste et très portable. Apache Tomcat [\[w11\]](#) est une implémentation open source d'un conteneur web qui permet d'exécuter des applications web reposant sur les technologies servlets et JSP.

#### 4.2.1.2 Phpmyadmin

Phpmyadmin est un utilitaire qui consiste en un ensemble de scripts PHP rendant plus conviviale l'administration de bases de données MySQL en passant par un navigateur web.

#### 4.2.1.3 MySQL

MySQL est un serveur de bases de données multiutilisateurs et multitraitements, cela permet d'établir des connexions très rapides et d'utiliser la même mémoire cache pour plusieurs requêtes. Nous utilisons MySQL pour la gestion des bases de données de notre application.

### 4.2.2 Android Studio

Android [\[w12\]](#) est un système d'exploitation mobile composé d'un ensemble de logiciels destinés à fournir une application open source pour smartphones, tablettes. Nous développons une applica-

tion sous android dans l'objectif de récupérer les coordonnées géographique (longitude, latitude) à partir d'un terminal (smartphone) vers notre application. La fonction qui permet de récupérer les coordonnées à partir d'un terminal est :`LocationManager.requestLocationUpdates(String provider, long minTime, float minDistance, LocationListener listener)`

- provider le fournisseur de position (nous avons utilisé `GPS_PROVIDER`).
- minTime la période entre deux mises à jour en millisecondes. Nous avons mis la valeur 60000 (60 seconde).
- minDistance la période entre deux mises à jour en mètres. Nous l'avons mise à 0 pour qu'il ne soit pas pris en compte.
- listener est l'écouteur qui sera lancé dès que le fournisseur sera activé.

### 4.2.3 Éclipse

Eclipse IDE (Integrated Development Environment) est un environnement de développement intégré, il est à la fois simple et puissant. Grâce à sa force d'accepter des plug-ins, cela lui permet de s'adapter à plusieurs langages (c++, PHP, java). Pour réaliser notre application nous avons utilisé le JEE (Java Enterprise Edition) sous la plate-forme Ganymède version 3.4.1. L'objectif majeur de JEE est de faciliter le développement d'application web robustes et distribuées, déployées et exécutées sur un serveur d'application.

### 4.3 Schéma de fonctionnement de l'application



FIGURE 4.1 – Fonctionnement général de l'application

1. Les données sont récupérées à partir d'un smartphone et ceci à travers une application android.
2. Les données seront envoyées à l'application JEE grâce au service web.
3. Les données seront enregistrées dans la base de données.
4. Pour consulter la localisation d'un camion, les données seront récupérées à partir de la base de données.
5. Les données seront affichées sur une carte géographique (Googlemaps).

### 4.4 Architecture de développement

Parmi les architectures de développements qu'existe, on trouve le modèle MVC qu'est une application de trois-tiers entre l'utilisateur, le système ainsi qu'un niveau intermédiaire. Un utilisateur de



l'application, qui est le gestionnaire, tente d'accéder à une donnée en utilisant son navigateur web, qui contacte un serveur d'applications qui se charge de récupérer et d'orienter sa requête. Ensuite pour générer la réponse, le serveur d'applications contacte le serveur de BDD, qui lui effectue le traitement sur les données stockées et renvoie le résultat.

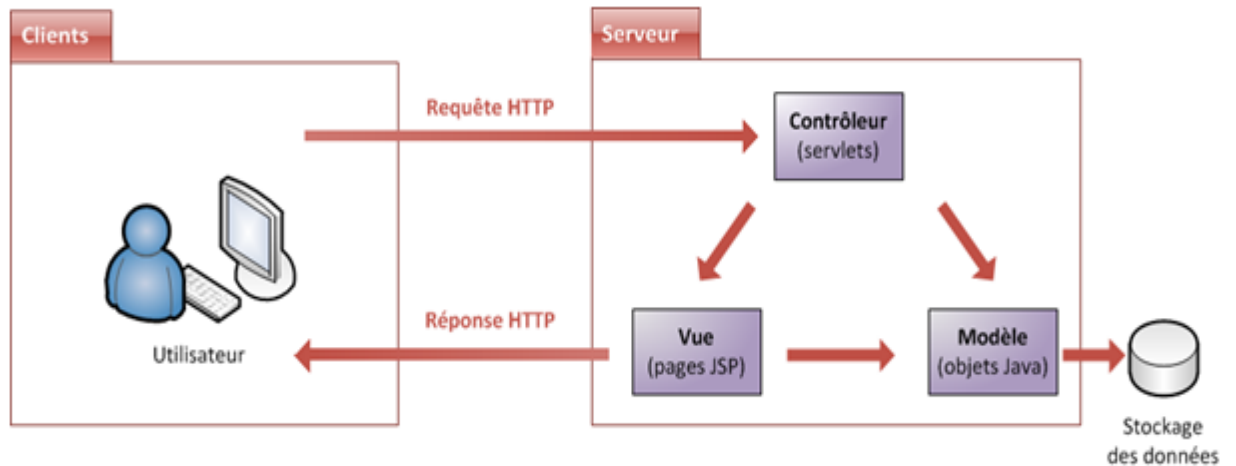


FIGURE 4.2 – Architecture trois-tiers

#### 4.4.1 MVC

MVC est un modèle permettant de découper une application web en trois couches pour en faciliter le développement [w13] :

- **Le modèle** représente l'algorithmique de l'application : traitements des données, interactions avec la base de données. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données.
- **La vue** : Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (clic de souris, sélection d'un bouton radio, couchage d'une case, entrée de texte, de mouvements, de voix,

etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

- **Le contrôleur** : prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

### 4.4.1.1 JSP ( Java Server Pages)

Un standard employé dans le développement d'applications Web, il est basé sur un langage de scripts (le code java) qui s'exécute du côté serveur et non pas du côté client, il tolère l'hétérogénéité du contenu de ses pages, des balises standards, générer de façon dynamique du code HTML, XML, etc. Cela permet l'ajout d'actions dynamiques dans des pages statiques.

### 4.4.1.2 Les servlets

Une servlet, est une technologie java coté serveur, elle effectue des traitements sur des requêtes d'un client et génère une réponse. Une servlet est définie dans un projet comme étant une classe java respectant certaines spécifications JEE, elle hérite de certaines classes et implémente certaines interfaces. Elles sont généralement employées pour traiter des requêtes de type HTTP et retourner dynamiquement des pages HTML. Une servlet joue le rôle d'un contrôleur dans une application web.

### 4.4.1.3 JavaBean

C'est un objet java auquel on applique certaines contraintes et doit être paramétrable, sérialisable, réutilisable. Un bean doit :

- être défini comme classe publique.
- contenir un constructeur par défaut.
- avoir des propriétés (champs privés) accessibles via des méthodes et ne pas avoir de champs publics.

### 4.4.1.4 Service web

Les services web (en anglais web services) représentent un mécanisme de communication entre applications distantes à travers le réseau internet indépendant de tout langage de programmation et de toute plate-forme d'exécution [22].

## 4.4.2 Communication avec la base de données

### 4.4.2.1 JDBC

La technologie JDBC (Java DataBase Connectivity) est une API fournie avec Java permettant de se connecter à des bases de données, c'est-à-dire que JDBC constitue un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données (SGBD) [13]. L'URL de connexion indispensable à Java pour se connecter à BDD est `jdbc:mysql://localhost:3306/application-gsp` qui est configuré dans le fichier `hibernate.cfg.xml`.

### 4.4.2.2 Hibernate

Hibernate est un framework permettant de gérer la persistance d'objets dans une BDD, un objet persistant étant un objet destiné à être conservée pour une durée plus longue que celle de l'exécution du programme [23]. Les étapes de configuration de hibernate sont :

1. Créer la BDD et le compte d'utilisateur pouvant y accéder.
2. Configurer Hibernate (`hibernate.cfg.xml`) avec les informations concernant la connexion à la base.
3. Créer la classe d'objets `JavaBean` qui encapsulera les données de chaque occurrence (ligne) de la table.
4. Ecrire les Fichiers de mapping pour assurer la correspondance entre les tables et les classes `JavaBeans`.

Hibernate effectue ce que l'on appelle le mapping relationnel - objet, il permet donc de sauvegarder une instance persistante d'une classe dans une ligne d'une table dans la BDD de façon transparente, hibernate offre aussi des avantages quand à la récupération des données, il est facile d'emploi et efficace. L'implémentation de classes métiers (`GenericDAO.java`) contenant les méthodes qui permettent l'interaction avec la BDD est rendue facile et pratique.

### 4.4.3 GoogleMaps

#### 4.4.3.1 Définitions

Google Maps[w15] est un service gratuit de cartographie en ligne. GoogleMaps est un service 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 .

#### 4.4.3.2 Api GoogleMaps

Api de Google permettant de géolocaliser des adresses sur une carte à l'aide de ses coordonnées (latitude et longitude ). Api permet de localiser tout type de données sur une carte (routière, satellite, mixte) à partir de sa localisation . Cet api s'avère très utile pour proposer aux internautes une vision globale et géographique de données. L'élément important que l'on retrouve dans le code est l'importation de la librairie GoogleMaps grâce aux balises. `<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false"> </script>`

GoogleMaps est basé sur JavaScripts . Lorsque l'utilisateur cherche un camion, une petite bulle est affichée sur la carte pour indiquer la position du terminal.

## 4.5 Présentation de quelques interfaces

### 4.5.1 Interface de connexion

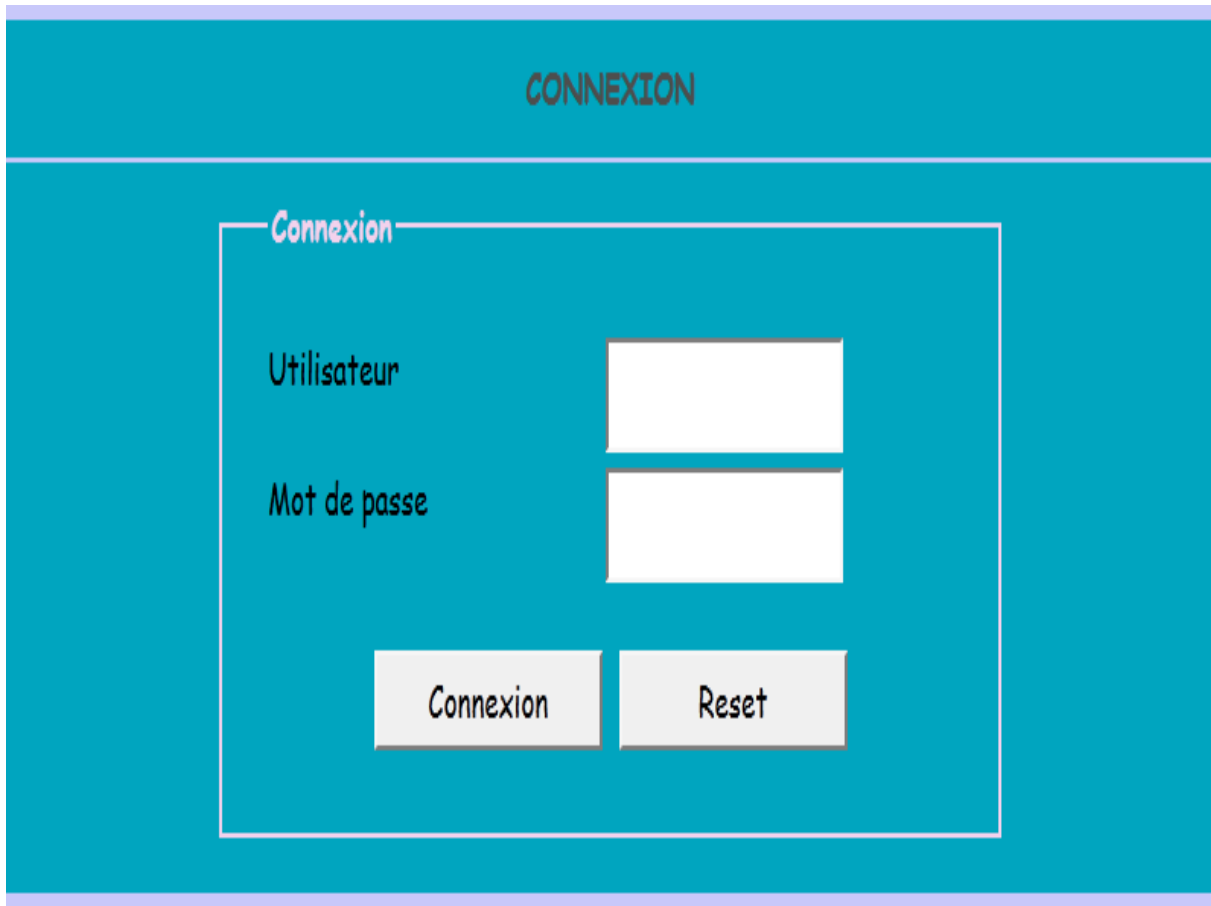


FIGURE 4.3 – Interface de connexion

### 4.5.2 Interface menu d'administrateur



FIGURE 4.4 – Interface administrateur

### 4.5.3 Interface d'ajout d'un camion

The image shows a web interface for adding a truck. The interface has a teal background with a light purple header and footer. In the top right corner, there is a button labeled "Deconnecter". The main content area is a white box titled "Ajouter Camion" with a thin border. Inside this box, there are three input fields: "Matricul camion", "Type\_cam", and "Num\_ter". Below the input fields, there are three buttons: "Valider", "Reset", and "Retour".

FIGURE 4.5 – Interface d'ajout d'un camion

## 4.5.4 La liste des utilisateurs

ID	UTILISATEUR	MOT DE PASSE	ROLE	SUPPRIMER
1	sylia	nassim	admin	<input type="radio"/> Supprimer
3	d	d	utl	<input type="radio"/> Supprimer
4	eeeeeeee	sssssss	utl	<input type="radio"/> Supprimer
7	a	a	admin	<input type="radio"/> Supprimer
8	v	v	admin	<input type="radio"/> Supprimer

FIGURE 4.6 – Interface liste des utilisateurs



4.5.5 Interface suivi en temps réel



FIGURE 4.7 – Interface suivi en réel des camions

### 4.5.6 Interface historique d'un camion



FIGURE 4.8 – Interface historique d'un camion

## 4.5.7 Interface application d'android

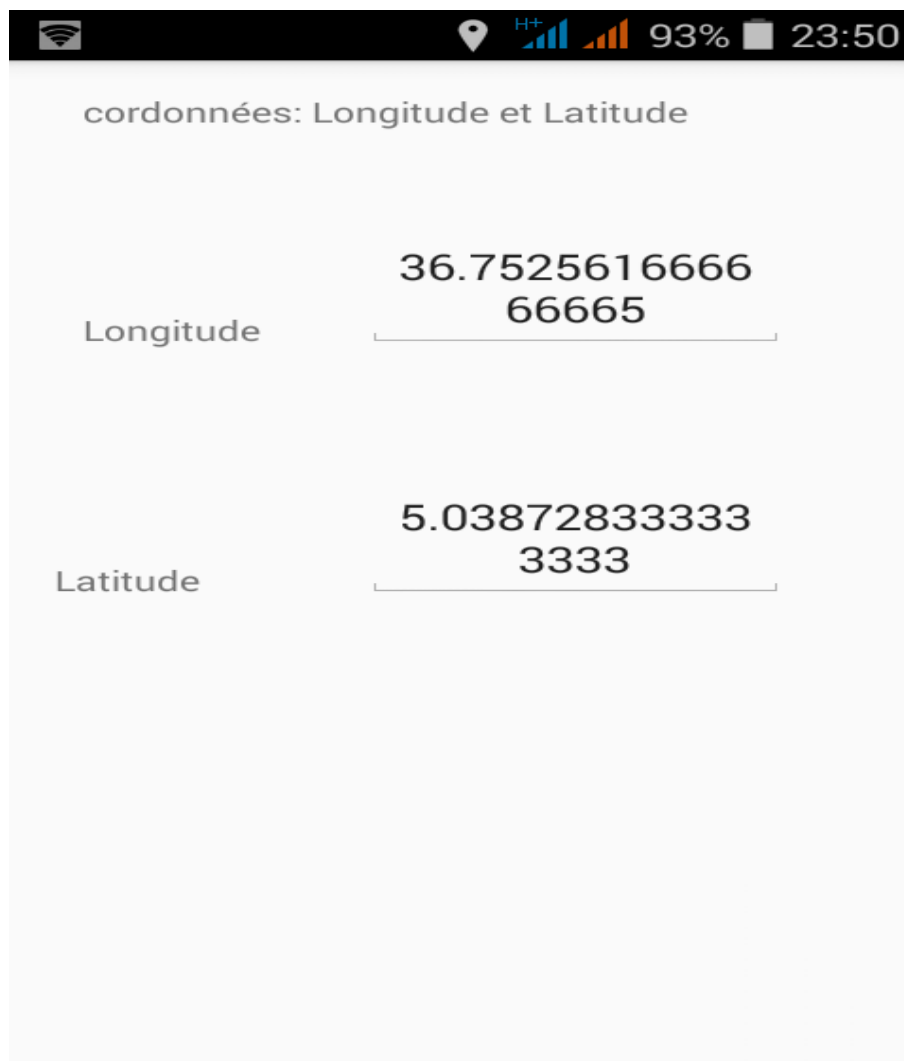


FIGURE 4.9 – Interface application android

## Conclusion

Dans ce chapitre, nous avons présenté notre application web en mettant en évidence l'environnement de développement et le modèle MVC comme architecture de développement. Ainsi, nous avons utilisé la technologie hibernante et JDBC pour communiquer avec la base de données. Enfin nous concluons notre chapitre par les présentations de quelques interfaces de notre application de géolocalisation des camions.

---

## CONCLUSION GÉNÉRALE

---

Au terme de notre travail, nous espérons avoir atteint l'objectif que nous nous sommes fixé au début, à savoir la réalisation d'une application web permettant principalement la localisation des camions sur une carte géographique. Nous espérons également que cette application sera bénéfique et rentable à l'entreprise Bejaia Logistique et ce dans la mesure où elle facilitera les tâches de ses utilisateurs en ce qui concerne la bonne gestion et la localisation des ses camions.

Pour concrétiser ce travail, nous avons étudié le fonctionnement théorique des systèmes de localisation et nous avons montré l'utilité et l'efficacité des systèmes GPS. Ce qui permettra dans cette perspective la généralisation de cette technologie en Algérie.

Nous avons présenté aussi les fonctionnalités de l'application modélisées avec le langage UML 2.4 suivant la démarche agile que nous avons appris à appliquer durant ce travail. Pour la réalisation nous avons choisi de suivre le modèle MVC qui est un modèle destiné à répondre aux besoins des applications lors du développement. Nous avons choisi Wampserver, Android studio pour récupérer les données à partir d'un Smartphone et Eclipse pour le développement de notre application.

Ce travail nous a amené à confirmer qu'avec de la détermination il est possible d'explorer des domaines totalement nouveaux . Un travail qui nous a permis également d'apprendre et de nous initier aux services de géolocalisation qui feront bientôt partie de la vie quotidienne de chacun de nous.

---

# BIBLIOGRAPHIE

- [1] T. DUDOKDEWIT, Guide pour l'utilisation GPS, 2010.
- [2] D. WELLS, Guide to GPS Positioning, 1987.
- [3] Q. PHAM, La localisation géographique des robots mobiles : Méthodes et Outils correspondants, 2007.
- [4] C. GUEDAT et O. BRETTE, Système de géolocalisation, INSA Lyon, Janvier 2010.
- [5] R. AHRIZ, Application des techniques d'apprentissage à la géolocalisation par radio fingerprint, Université pierre et marie curie, thèse Doctorat, 2010.
- [6] D. LARREY et L. RODIER, Géolocalisation par WiFi, Ecole des mines de nancy, Rapport de recherche, Février 2006.
- [7] C. SAAD, Quelques contributions dans les réseaux de capteurs sans fil : Localisation et Routage, Université d'Avignon et des Pays de Vaucluse. These Doctorat, 2008.
- [8] L. ADNANE, Géolocalisation dans les réseaux mobiles (GSM et CDMA à MOBI-LINFO), Master Informatique, (2008-2009).
- [9] H. SEGHIER, Localisation dun mobile dans un réseau UMTS, Option Technique de Communication Modernes, Diplome de Magister, (2012-2013).
- [10] H. BENSLIMANE, Localisation dans les réseau de capteurs, Université d'Avignon et des Pays de Vaucluse, Master RTM, 2012.
- [11] L. MAIGNON, UML-Unified Modeling Language (département statique), Université Claude Bernard Lyon 1, (2012-2013).

- [12] J. PILLOU, Méthode agiles (RAD, XP), Septembre 2015.
- [13] La communauté Agile de Valtech, Transition vers l'agilité à l'échelle d'une organisation deuxième édition, 2012.
- [14] P. ROQUES, UML 2.0 Modélisation une application web, les cahiers du programmeur, 4ème édition-Eyrolles, Octobre 2008.
- [15] P. ROQUES, UML 2 en action, De l'analyse des besoins à la conception, 4ème édition-Eyrolles, Mars 2007.
- [16] P. ALBERT, N. KETTANI, D. MIGNET, De Merise à UML, Edition-Eyrolles France édition, Octobre 2001.
- [17] G. JEROMR, SQL Server, édition Eni, Octobre 2015.
- [18] B. FRAIN, Responsive Web Design avec HTML 5 et CSS 3,1re édition-Pearson, février 2013.
- [19] C. DELANNOY, Programmer en Java, édition-Eyrolles, septembre 2004.
- [20] C. VIGOUROUX, Apprendre à développer avec JavaScript, Editions-ENI, Avril 2014.
- [21] E. DASPET et P. GEYER, PHP 5 avancé, 6ème édition-Eyrolles, Mai 2012.
- [22] H. KADIMA et V. MONFORT, Les web services, DUNOD 1 édition, 2004.
- [23] T. HAMON, Programmation Web Avancée Hibernate, Université Paris 13, (2012-2013).

## Webographie

- [w1] Autorité de surveillance du GNSS européen, Les programmes européens de navigation par satellite, [http://www.gsa.europa.eu/sites/default/files/eu\\_gnss\\_prog\\_leaflet4\\_fr.pdf](http://www.gsa.europa.eu/sites/default/files/eu_gnss_prog_leaflet4_fr.pdf), consulté le 25/04/2016 à 16 :34 .
- [w2] La mutuelle générale, Alzheimer que penser des dispositifs de géolocalisation <https://www.lamutuellegenerale.fr/le-mag-sante/senior/>

- [alzheimer-que-penser-des-dispositifs-de-geolocalisation.html](#), consulté le 15/04/2016 à 14 :30 .
- [w3] Danvincréation,Securite [http://www.mrsecurite.com/?page=arbo\\_spe&section=15&phpsessid=b25bef6d0181739aea4d6527268158d2](http://www.mrsecurite.com/?page=arbo_spe&section=15&phpsessid=b25bef6d0181739aea4d6527268158d2), consulté le 22/04/2016 à 16 :06.
- [w4] Drivstersarl, Automobile-sportive, <http://www.automobile-sportive.com/technique/gps.php>, consulté le 21/04/2016 à 17 :09 .
- [w5] Adproxima, Lexique de l'informatique, <http://www.additeam.com/ssii/uml/>, consulté le 18/04/2016 à 10 :27.
- [w6] Blog gestion de projet, Les méthodes agiles, <http://www.blog-gestion-de-projet.com/gerez-vos-projets-informatiques-avec-les-methodes-agiles/>, consulté le 25/04/2016 à 22 :17.
- [w7] Microsoft translator, Developpez.com, <http://sabricole.developpez.com/uml/tutoriel/unifiedprocess/>, consulté le 25/04/2016 à 22 :49.
- [w8] Microsoft translator, Developpez.com, <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation>, consulté le 18/05/2016 à 15 :27 .
- [w9] Sharpened Productions, Techterms, <http://techterms.com/definition/html>, consulté le 20/05/2016 à 13 :08.
- [w10] Jean-François PILLOU, commentcamarche, <http://www.commentcamarche.net/contents/1332-xml-introduction-a-xml>, consulté le 20/05/2016 à 15 :33 .
- [w11] Jean-Michel DOUDOUX, Développons en Java, <http://www.jmdoudoux.fr/java/dej/chap-tomcat.html>, consulté le 17/06/2016 à 13 :35.
- [w12] Developer, Androide, <https://developer.android.com/guide/index.html>, consulté le 21/05/2016 à 17 :03 .
- [w13] Cake Software Foundation, Comprendre le système M-V-C, <http://book.cakephp.org/2.0/fr/cakephp-overview/understanding-model-view-controller.html>, consulté le 22/05/2016 à 15 :21.
- [w14] Commentcamarche, Introduction JDBC, <http://www.commentcamarche.net/contents/594-introduction-a-jdbc>, consulté le le 22/05/2016 à 13 :33.

- [w15] Techtarget, GoogleMaps, <http://whatis.techtarget.com/definition/Google-Maps>, consulté le 22/05/2016 à 14 :14.
- [w16] Dia Diagram Editor, Product, <http://www.dia-installer.de>, consulté le 21/06/2016 à 13 :52.
- [w17] Éric Chassaing, Schéma de principe de la géolocalisation par GPS, <https://fr.wikipedia.org/wiki/GC3A9olocalisation#/media/File:Geolocation.png>, consulté le 21/06/2016 à 15 :00.
- [w18] Wikipedia, géolocalisation, <https://fr.wikipedia.org/wiki/GC3A9olocalisation>, consulté le 27/06/2016 à 14 :52.
- [w19] <http://www.centre-universitaire-cambrai.fr/ecole-de-printemps-systemes-de-localisation-et-continuite-de-service>.



## Résumé

Ce mémoire, réalisé dans le cadre de notre projet de fin cycle, avait pour principal objectif d'élaborer pour l'entreprise Béjaia Logistique, une application de géolocalisation de camions. Notre travail a consisté à la conception et à la réalisation d'une application qui permettra le suivi en temps réel des camions sur une carte géographique. La réalisation de l'application a mis en avant un ensemble d'exigences fonctionnelles, organisationnelles et techniques pour répondre aux besoins de Bejaia logistique. Ainsi, notre démarche a été conduite par le processus UP, en utilisant le langage UML pour la modélisation du système et le modèle MVC pour le développement de l'application.

## Mots clés

Géolocalisation, GPS, Android, UML, Méthodes Agiles, MVC, MYSQL, Google Maps, JEE, Wampserver.

## Abstract

This project, done as a part of our final of our project of the end of studies, the main objective was to develop the Bejaia logistics society, an application of a geolocation of its trucks. Our work consists on the conception and the realization of a platform that will enable the real-time tracking of trucks on a geographical map. The realization of this platform has put forward a set of functional, organizational and technical requirements in order to achieve a system that meets the needs of Bejaia logistics. Thus, our approach was made by the process UP, using the UML language for the modeling of the system and the use of MVC pattern for the development of the application .

# Annexe

## Procès verbal

En ce jour du 27 juin 2016, nous dressons un bilan des différentes réunions tenu au sein de l'entreprise Béjaïa Logistique pour recueillir les besoins des différents services de l'entreprise.

Après plusieurs réunions avec les responsables de l'entreprise Béjaïa Logistique nous avons pu recenser les difficultés que rencontre cette entreprise dans la gestion de ces camions et la manière dont fonction les différents services.

Nous avons eu plusieurs rencontres avec le directeur et aussi le responsable du service programmation et du service personnel ou nous nous sommes entretenu avec chaque un d'eux et nous avons recueilli les différents avis sur le fonctionnement et la gestion des camions.

A la fin de nos réunions nous avons recensé les besoins suivant :

1. Connaitre la position des camions en temps réel.
2. Une gestion des camions centralisés.
3. Garder l'historique de déplacement des camions.
4. Etablir des zones par ou les conducteur ne devons pas passer.
5. Prévenir les conducteurs en cas de besoin.
6. Etablir des missions à l'avance pour les conducteurs.