NIVERSITÉ ASSANE SECK DE ZIGUINCHOR

UFR SCIENCES ET TECHNOLOGIES

DÉPARTEMENT : INFORMATIQUE

MASTER : INFORMATIQUE

SPÉCIALITÉ : GÉNIE LOGICIEL

SUJET :

Étude et mise en place d'un système de suivi de flotte de véhicules

Présente par:

M. Mamadou BADJI

Sous la direction de :

Dr. Khadim DRAME

Sous la supervision de :

Pr. Salomon SAMBOU

Membre du jury :

Dr. Ibrahima DIOP (Rapporteur)

M. Assane SECK (Maitre de stage)

Pr. Salomon SAMBOU (président)

Dr. Youssou DIENG (Examinateur)

Année universitaire : 2016/2017

Dédicaces

Toutes les lettres ne sauraient trouver les mots qu'il faut… Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance…

Aussi, c'est tout simplement que :

Je dédie ce travail à :

À mes chers parents,

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consentis pour mon instruction et mon bien-être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez. Puisse Dieu, le très haut, vous accorde santé, bonheur et longévité et faire en sorte que jamais je ne vous déçoive.

À mes chers et adorables frères et sœurs,

Que j'aime profondément. En témoignage de mon affection fraternelle, de ma profonde tendresse et reconnaissance, je vous souhaite une vie pleine de bonheur et de succès et que Dieu, le Tout-Puissant, vous protège et vous garde.

À mes chers oncles, tantes, leurs époux et épouses, à mes chers cousins et cousines,

Veuillez trouver dans ce travail l'expression de mon respect le plus profond et mon affection la plus sincère.

À toutes les personnes qui ont participé à l'élaboration de ce travail, à tous ceux que j'ai omis de citer.

Remerciements

Louange à Dieu le clément, le miséricordieux, qui nous a donné le courage et la patience de mener ce travail.

Au terme de mon travail de fin d'études de Master 2 en Génie logiciel, j'ai le plaisir d'adresser mes sentiments de reconnaissance les plus profonds à Monsieur Khadim DRAME pour son soutien, pour aussi sa disponibilité et ses nombreux conseils tout au long de la rédaction de mon mémoire qui, malgré ses multiples occupations, a bien accepté de diriger et encadrer le présent mémoire.

Je remercie grandement Monsieur Assane SECK et Monsieur Ousseynou SECK qui m'ont inculpé de vastes connaissances érudites.

Nos vifs remerciements vont également aux membres du jury (Pr Salomon SAMBOU, Dr Youssou DIENG, Dr Ibrahima DIOP) pour l'intérêt qu'ils ont porté à notre travail. Nous leur remercions également pour leurs pertinentes remarques et suggestions.

Je tiens à remercier Monsieur Ablaye DIEME, Monsieur Malang BADJI pour leur soutien moral, matériel et financier, mais aussi leur disponibilité et leurs nombreux conseils tout au long de mon cursus universitaire.

Je ne pourrais en aucun cas oublier de remercier les enseignants du département d'informatique de l'UASZ et plus particulièrement M. Youssou DIENG et M. Youssou FAYE pour le savoir qu'ils m'ont inculpé, leurs conseils de qualité et orientations.

Que tous ceux qui se sont montrés disponibles pour me fournir des ressources tant orales qu'écrites trouvent ici l'expression de mes vifs remerciements.

Enfin, ma gratitude va à ma famille qui me soutient chaque jour dans ses prières.

Résumé

La géolocalisation est une technique qui sert à déterminer la position géographique précise d'un objet ou d'un individu dans un environnement bien déterminé.

Les systèmes de première génération permettaient simplement de localiser des véhicules. Or, on peut aujourd'hui suivre en temps réel le trajet complet d'un individu ou objet grâce à une balise GPS (Global Positioning System). Notre travail s'inscrit dans ce cadre et porte sur l'étude et la mise en place d'un système de suivi de flotte de véhicules. Après un état de l'art sur la géolocalisation, nous avons conçu un système, qui, intégré avec un autre module développé indépendamment, permettra à une entreprise ou un individu de suivre son ou ses véhicule(s) en temps réel à l'aide d'un smartphone ou d'une tablette. Un prototype du système est ensuite implémenté et testé au sein de l'entreprise Silimax.

Table des matières

Liste des figures

Liste des tableaux

INTRODUCTION

Connaitre son espace de vie dans les détails s'avère essentiel. Parler de cet aspect fait appel à l'espace géographique rendu accessible grâce aux systèmes d'information géographiques (SIG). Les SIG et leurs applications sont de plus en plus incontournables dans plusieurs domaines tels que la santé, l'environnement, l'urbanisme, le transport, les télécommunications, etc.

La localisation par Global Positioning System (GPS) s'adresse à tout particulier, entreprise, etc. désirant optimiser ses conditions de travail (contraintes de rentabilité et de productivité). De ce fait, pour répondre aux exigences des clients, la géolocalisation devient un réel outil de productivité tout en optimisant sa gestion de flotte.

L'intérêt d'une telle plateforme réside en sa capacité de suivre en temps réel les véhicules, sur une carte géographique. Elle conserve aussi l'historique pour tracer les chemins des voyages à la demande et permet de générer les rapports en cas de besoin. Ceci assurera à l'entreprise en question l'arrivée de leurs biens dans les conditions convenables et lui permettra en même temps de prendre des décisions appropriées au moment du changement d'itinéraire, d'affectation des chauffeurs, etc. Les informations (GPS) d'un véhicule (latitude, longitude, vitesse, destination, date/heure) sont envoyées par un équipement vers un serveur qui les stocke dans une base de données (BD).

Notre stage portant sur le sujet **étude et mise en place d'un système de suivi de flotte de véhicule** s'inscrit dans ce cadre. L'objectif est de concevoir et de mettre en œuvre une application pour le suivi de flotte de véhicules.

Ce rapport s'articule autour de cinq chapitres. Nous commençons, dans le premier chapitre, par présenter notre structure d'accueil et faire un état de l'art. Puis, le deuxième chapitre présentera notre méthodologie. Dans le troisième chapitre ; nous présenterons la spécification et l'analyse des besoins. Tout au long du quatrième chapitre, nous détaillerons la conception de notre application. Enfin, le cinquième chapitre décrit l'environnement de développement et présente l'application réalisée. Ce rapport sera clôturé par une conclusion qui récapitule les apports de notre projet et les perspectives de ce travail.

CHAPITRE 1 : ÉTUDE PRÉALABLE ET ÉTAT DE L'ART

Dans ce chapitre, nous commencerons par une présentation générale de notre structure d'accueil pour la réalisation de ce stage ; ensuite nous allons présenter le cadre et poser la problématique de notre travail. Nous allons ensuite terminer par une étude sur la géolocalisation et sur les solutions existantes.

1.1. Présentation de la structure d'accueil

Constituée en 2014 et basée à Dakar Sénégal, Silimax (Solution informatique à liaison maximale) est une société innovante dans le domaine des nouvelles technologies. Ayant une stratégie d'innovation et de développement, Silimax s'est intéressée sur des projets d'entreprises très en avance dans la réalité environnementale. Silimax est d'une part en partenariat avec les entreprises pour la réalisation et conduite de projet et d'autre part avec les universités pour l'appui à la formation professionnelle par le biais des ateliers et séminaires.

1.1.1. Missions

Spécialisée dans le domaine des technologies Web, mobiles, SMS, e-learning, solutions de paiements, et des systèmes d'informations géographiques, Silimax s'intéresse à la résolution des problèmes des entreprises. Elle s'attache aux tâches suivantes:

- ✓ fournir des services à valeur ajoutée pour le grand public ;
- ✓ rendre accessible des services de paiement partout et à moindre coût ;
- ✓ fédérer les énergies locales puis régionales pour la création d'un marché commun africain ;
- ✓ contribuer à pérenniser les initiatives privées pour l'entrée effective de l'Afrique dans l'économie mondiale ;
- ✓ positionner nos marchés dans la nouvelle économie qui est l'économie de l'Information ;
- ✓ promouvoir l'accès aux services à fortes valeurs ajoutées ;

1.1.2. Solutions proposées au marché

Silimax intervient dans des domaines variés tels que :

- ✓ le développement d'applications Web ;
- ✓ le développement d'applications mobile ;
- ✓ le développement de solutions SMS ;
- ✓ le développement de systèmes de paiement ;
- ✓ etc.

1.1.3. Organigramme

La société Silimax est constituée d'une direction générale, d'un département développement et d'un département réseau et système.

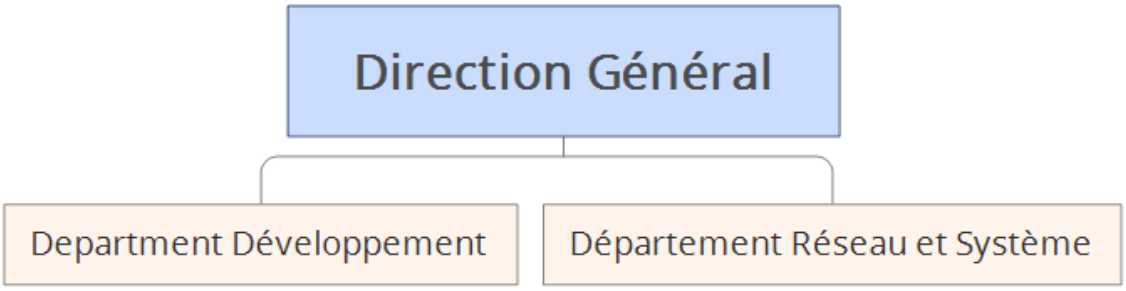

Figure 1 : Organigramme de Silimax

1.2. Problématique

L'industrie, le commerce, l'agriculture, le tourisme et plusieurs autres secteurs se fondent sur des flottes de véhicules pour importer les matières primaires, livrer des marchandises, transporter les clients, etc. Ces flottes sont généralement massives et leur gestion est très délicate.

Minimiser les coûts de transports, livrer dans les meilleurs délais, réagir le plus tôt possible, optimiser les trajets des véhicules, contrôler les déplacements des véhicules, se protéger contre le vol, améliorer la qualité des services proposés aux clients, accroitre la rentabilité reste les principaux soucis des gestionnaires des entreprises et des transporteurs.

Les systèmes de gestion de flottes à base de la technique GPS offrent des solutions à ces problèmes. Le GPS joue un rôle important dans le suivi des véhicules en temps réel.

Notre travail s'inscrite dans ce cadre a pour objectif principal la mise en place d'une plateforme de gestion d'une flotte des véhicules en s'appuyant sur les nouveaux techniques de positionnement par GPS et de communication General Packet Radio Service (GPRS). L'objectif est de proposer une solution qui va permettre aux entreprises de gérer le suivi en temps réel ou diffère de leurs véhicules.

1.3. Généralités sur la géolocalisation

1.3.1. Le principe de géolocalisation

La géolocalisation ou géoréférencement est un procédé permettant de positionner un objet ou une personne sur un plan ou une carte à l'aide de ses coordonnées géographiques. Cette opération est réalisée à l'aide d'un terminal capable d'être localisé (grâce à un récepteur GPS par exemple) et de publier (en temps réel ou de façon différée) ses coordonnées géographiques (latitude/Longitude). Les postillons enregistrés peuvent être stockés au sein du terminal et être extraits postérieurement, ou être transmis en temps réel vers une plateforme logicielle de géolocalisation. Ceci permet de visualiser la position du terminal au sein d'une carte à travers une plateforme de géolocalisation le plus souvent accessible depuis internet.

Le réseau satellitaire de positionnement le plus connu est le GPS .Dans le cas du GPS, pour que le repérage spatial fonctionne, un immense réseau constitué de 27 satellites différents est utilisé. Ces satellites servent de repères aux navigateurs GPS dans leur processus de calcul de position. Ce système de satellites est conçu de façon à ce qu'il y en ait toujours au moins quartes visibles par le navigateur GPS, sans quoi la position ne peut pas être déterminée. Pour qu'un terminal soit capable d'être localisé grâce au réseau GPS , il doit absolument être équipé d'une puce électronique GPS.

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

- un terminal communicant : c'est le terminal qui reçoit les coordonnées géographiques (via GPS ou tout autre moyen) et qui les envoie via un réseau de télécommunications à la plateforme ;

- un système informatique capable de recevoir, stocker et traiter les informations : il s'agit des serveurs informatiques qui hébergent l'infrastructure et qui reçoivent et traitent les données envoyées par les terminaux ;
- un module cartographique : c'est le module intégré au système informatique qui va permettre d'afficher la position des terminaux sur un fond cartographique adapté. Ce module prend en charge les calculs de distance, d'itinéraires, détecte l'interaction avec les zones et permet d'avoir accès à des informations sur le terrain.

En effet, les données générées par le terminal qui se trouve sur le terrain doit être transmise à une plateforme qui va les traiter, les présenter graphiquement à l'utilisateur et les associer à d'autres données afin d'enrichir les informations relatives à l'état du terminal ou la flotte le figure 2 ci-dessous illustre le principe de la géolocalisation.

Figure 2 : Principe de la géolocalisation d'un véhicule équipé d'un terminal embarqué

1.3.2. Les outils et les techniques de géolocalisation

✓ **Boitier de géolocalisation**

Un boitier GPS est un dispositif permettant de positionner un objet ou un véhicule sur une carte en temps réel ou selon une périodicité prédéfinie. Les fonctions des boitiers modernes ne se limitent pas à ce point et incluent en outre des fonctions d'alerte et d'historique de trajets. Il existe deux sortes de boitiers GPS :

- les boitiers GPS-GSM utilisent une simple carte SIM GSM prépayé pour l'envoi des SMS d'alerte et de localisation. Ceci à l'avantage de ne pas nécessiter un abonnement téléphonique chez un opérateur ;
- les boitiers GPS-GPRS permettent de visualiser en temps réel la position GPS d'un objet sur des cartes routières informatisées. Ils nécessitent une connexion internet pour pouvoir afficher les points de localisation en temps réel sur un écran d'ordinateur. Dans le cas d'une entreprise de transport, les boitiers GPS-GPRS seront par exemple utilisés pour gérer, contrôler et optimiser les trajets de ses employés.

✓ **Système GPS**

Le système de GPS a été conçu pour permettre d'obtenir, partout dans le monde et rapidement, des données de navigation tridimensionnelles, avec une précision de l'ordre de centaine de mètres. Il se base sur une constellation de satellites, qui émettent en permanence un signal daté, et un réseau de stations au sol qui surveillent et gèrent les satellites. Les récepteurs sont passifs et le nombre d'utilisateurs est donc illimité. La localisation est possible dès lors que quatre satellites sont visibles : il y a en effet quatre inconnues à déterminer, les trois coordonnées spatiales, ainsi que le temps, puisque le récepteur au sol n'est pas synchronisé avec les satellites. Par coordonnées géographiques (ou encore « repères géographiques ») d'un lieu, on entend la latitude, la longitude et le niveau de la mer. Pour se repérer à la surface de la planète, on peut utiliser un autre système appelé « repères cartographiques ».

Pour se localiser sur la terre, il est nécessaire d'utiliser un système géodésique duquel découlent les coordonnées géographiques.

- **Latitude**

La latitude est une coordonnée géographique représentée par une valeur angulaire, expression de la position d'un point sur la terre (ou sur une autre planète), au nord ou au sud de l' équateur qui est le plan de référence. Lorsque reliés entre eux, tous les endroits de la Terre ayant une même latitude forment un cercle, dont le plan est parallèle à celui de l'équateur, d'où l'autre terme « parallèle » utilisé pour désigner latitude.

- **Longitude**

Les méridiens passent tous par les pôles. La longitude est une coordonnée géographique représentée par une valeur angulaire, expression du positionnement est-ouest d'un point sur la terre (ou sur une autre planète). La longitude de référence sur terre est le méridien de Greenwich. Tous les points de même longitude appartiennent à une ligne épousant la courbure terrestre, coupant l'équateur à angle droit et reliant le pôle Nord au pôle Sud. Cette ligne est appelée « méridien ». À la différence de la latitude (position nord-sud) qui bénéficie de l'équateur et des pôles comme références, aucune référence naturelle n'existe pour la longitude.

La longitude, généralement notée λ, est donc une mesure angulaire sur 360° par rapport à un méridien de référence, avec une étendue de -180° (180°) Ouest à +180° (-180°) est. Le méridien 0° est le méridien de Greenwich.

- ✓ **Service GPRS**

General Packet Radio Service (GPRS) est employé comme un service de transmission de données, il s'agit d'une mise à niveau de n'importe quel réseau GSM. Il permet aux réseaux GSM d'être vraiment compatibles avec l'internet. GPRS emploie une technique de transfert en mode paquet pour transférer le trafic de données de façon efficace. Il permet des taux de transmission de 9.6 kb/s à plus de 150 kb/s par utilisateur. Le GPRS permet de fournir une connectivité IP constamment disponible à une station mobile (MS), mais les ressources radio sont allouées uniquement quand des données doivent être transférées, ce qui permet une économie de la ressource radio. Les utilisateurs ont donc un accès bon marché, et les opérateurs économisent la ressource radio. De plus, aucun délai de numérotation n'est nécessaire.

- ✓ **Fournisseurs de carte de géolocalisation**

Les coordonnées géographiques (longitude/latitude) permettent de positionner chaque adresse sur une carte numérique via un système d'informations géographiques (SIG). Deux principaux collecteurs/fournisseurs de cartographies numériques maintiennent un référentiel mondial d'adresses géolocalisées :

- TéléAtlas : Collecteur et fournisseur de cartographies routières mondial ;

- NAVTEQ : Fournisseur mondial de cartes, de données routières et de géolocalisation alimentant des systèmes de navigation, des services de géolocalisation et de publicité mobile dans le monde entier.

De multiples applications grand public utilisent ces deux supports pour géocoder des informations. Il y a plusieurs types de cartes de différentes interfaces sur le web. Parmi ces cartes nous citons : Google Map, Bing Mapz, etc.

- **Google Maps**

Google Maps est un service gratuit de cartes géographiques et de plans en ligne. Le service a été créé par Google en 2005. Très novateur dès sa création et ayant toujours une bonne longueur d'avance sur ses concurrents, Google Maps a révolutionné les services de cartes en ligne en proposant une vue satellitaire très détaillée de la surface de la Terre, les zones très densément peuplées bénéficiant d'une précision étonnante. Les fonctions classiques de calcul d'itinéraires routiers sont naturellement offertes, mais le site propose en outre de nombreuses autres fonctions intéressantes et souvent uniques. Google Maps est devenu pour beaucoup d'internautes un service indispensable pour le plaisir de la découverte comme pour des raisons pratiques.

- **Bing Map**

Bing Maps est le service de cartes et d'itinéraires proposé par Microsoft et son moteur de recherche bing. Le site propose toutes les fonctions qu'on peut attendre de ce genre de services : plans classiques, itinéraires routiers et vues satellitaires qui cependant n'atteignent pas le niveau de détails de celles proposées par Google Maps. Une fonction intéressante et innovante de Bing Maps est de proposer en plus des classiques vus satellitaires à la verticale, des photos aériennes en biais laissant apparaître davantage le détail des rues et des immeubles.

Les opportunités créées par des systèmes de gestion de flotte sont immenses et elles sont susceptibles d'avoir un impact de grande envergure. La solution de gestion de flotte basée sur GPS-GSM est une solution viable en raison de sa précision et de son aspect universel puisqu'elle permet la localisation dans n'importe quelle zone géographique en temps réel.

Dans ce chapitre, nous avons identifié les divers problèmes à résoudre et l'ensemble des besoins à satisfaire pour développer une application de gestion de flotte de véhicules, dans le chapitre suivant, nous allons présenter notre méthodologie de travail.

CHAPITRE 2 : MÉTHODOLOGIE

Ce chapitre présente notre méthodologie de travail. Dans un premier temps nous allons montrer pourquoi il faut utiliser une méthodologie, ensuite faire une classification des méthodes existante et enfin choisir et justifier le choix de notre méthodologie.

2.1 Pourquoi utiliser une méthodologie ?

L'utilisation d'une méthode dans la conduite d'un projet est fondamentale, car cette dernière garantit la réussite du projet. En effet, l'objectif principal d'un projet (de développement) informatique est de fournir un logiciel de qualité, dans les délais et à moindre coût tout en respectant les besoins du client. Par conséquent, la mise en aval du métier, du client et de la maitrise d'œuvre s'avère cruciale pour optimiser les ressources et la technologie afin de garantir le délai et le budget. Les méthodes répondent à ces exigences et permettent la construction d'application fonctionnellement et techniquement conforme aux attentes des différents participants du projet.

Donc, la réussite d'un projet informatique dépend essentiellement de deux facteurs notamment : l'implication des utilisateurs et une méthode garantissant le succès du projet et la qualité de l'application.

2.2 Classification des méthodologies

Nous disposons de nos jours d'un ensemble de méthodologies, chacune respectant un certain nombre de principes pour aboutir à la réalisation du logiciel. Nous allons présenter quelques méthodes existantes et, après une étude comparative, faire des choix en fonctions des avantages offerts et conformément aux attentes de notre projet.

1.2.1 Les méthodes classiques

Au début du génie logiciel, les projets étaient gérés avec la méthode dite « classique » qui se caractérise par les phases suivantes : recueillir les besoins des utilisateurs, définir le produit, le développer et le tester avant de le livrer. On parle alors ici d'une approche prédictive « cycle

en cascade ». Comme son nom l'indique, il s'agit ici de prévoir des phases séquentielles où il faut valider l'étape précédente pour passer à la suivante. Le chef de projet doit alors s'engager sur un planning précis de réalisation du projet en prévoyant des jalons de début et fin de phases ainsi que les tâches à effectuer. Il faut tout faire bien du premier coup, car cette approche ne peut pas permettre de retours en arrière. Une décision ou un problème rencontré dans une phase peuvent remettre en cause partiellement ou totalement les phases précédemment validées. Dans un cycle « en cascade », les risques sont détectés tardivement puisqu'il faut attendre la fin du développement pour effectuer la phase de test. Plus le projet avance, plus l'impact des risques augmente : il sera plus difficile et coûteux de revenir en arrière lorsqu'on découvre une anomalie tardivement [W2].

Par conséquent, comment peut-on augmenter la satisfaction du client en facilitant la gestion de projet et en améliorant la qualité de développement ? Pour ce faire, nous allons voir d'autres méthodes modernes plus adéquates.

1.2.2 Les méthodes unifiées

Un processus unifié est un processus construit sur Unified Modeling Language (UML). Plus exactement, ce sont les meilleures pratiques du développement objet suivies pour la réalisation d'un système. Un processus unifié se distingue par les caractéristiques suivantes :

- itératif : le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et incrémentaux par des itérations multiples ;
- piloté par les risques : les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité ;
- centré sur l'architecture : le choix de l'architecture logicielle est effectué lors des premières phases de développement du logiciel. La conception des composants du système est basée sur ce choix ;
- conduit par les cas d'utilisation : le processus est orienté par les besoins utilisateurs présentés par des cas d'utilisation.

Ils existent de nos jours plusieurs processus unifiés en vogue comme 2TUP et RUP :

- **Two Track Unified Process (2TUP):**

2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels [B1]. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent

les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens). Le processus s'articule ensuite autour de trois phases essentielles :

- une branche technique
- une branche fonctionnelle, et
- une phase de réalisation.

- **Rational Unified Process (RUP)**

RUP est l'une des plus célèbres implémentations des processus unifiés permettant de donner un cadre au développement logiciel [B1], et répondant aux exigences fondamentales préconisées par les créateurs d'UML :

- une méthode de développement doit être guidée par les besoins des utilisateurs ;
- elle doit être centrée sur l'architecture logicielle ;
- elle doit être itérative et incrémentale.

1.2.3 Les méthodes agiles

Une méthode agile est une méthode de développement informatique permettant de concevoir des logiciels en impliquant au maximum le demandeur (client).

Les méthodes agiles reposent sur une structure commune (itérative, incrémentale et adaptative). Elles utilisent un principe de développement itératif qui consiste à découper le projet en plusieurs étapes qu'on appelle « itérations ».

Ces itérations sont en fait des mini-projets définis avec le client en détaillant les différentes fonctionnalités qui seront développés en fonction de leur priorité. Le chef de projet établi alors une macro planning correspondant aux tâches nécessaires pour le développement de ces fonctionnalités.

Parmi les méthodes agiles, on peut citer :

- **Extrem programming(Xp)**

Plus particulièrement orientée sur l'aspect réalisation d'une application. XP est adapté aux équipes réduites avec des besoins changeants [B1].

L'Extreme Programming repose sur des cycles rapides de développement (des itérations de quelques semaines) dont les étapes sont les suivantes :

- une phase d'exploration détermine les scénarios client qui seront fournis pendant cette itération ;
- l'équipe transforme les scénarios en tâches à réaliser et en tests fonctionnels,
- chaque développeur s'attribue des tâches et les réalise avec un binôme,
- lorsque tous les tests fonctionnels passent, le produit est livré.

Le cycle se répète tant que le client peut fournir des scénarios à livrer. Généralement le cycle de la première livraison se caractérise par sa durée et le volume important de fonctionnalités embarquées. Après la première mise en production, les itérations peuvent devenir plus courtes (une semaine par exemple).

- **Scrum :**

Le principe de base de Scrum est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de durée fixe d'une à quatre semaines, appelées sprints [W5].

Chaque sprint possède un but à atteindre à partir duquel sont choisies les fonctionnalités à implémenter dans ce sprint. Un sprint aboutit toujours à la livraison d'un produit partiel fonctionnel.

L'utilisation de la méthodologie Scrum offre la possibilité de développer uniquement les fonctionnalités qui apportent une valeur ajoutée au produit, en toute transparence, avec le souci de la qualité et du respect des délais, et avec un retour rapide de la part du client.

1.2.4 Comparaison entre les méthodes

Il est très important de bien choisir la méthodologie qui répond parfaitement à ces besoins, s'adapte au mieux aux exigences du client et qui fournit le produit dans les plus brefs délais. Pour cela, il faut une étude comparative entre les méthodologies afin de faire un bon choix.

Le tableau ci-dessous décrit les méthodologies de développement ainsi que les points forts et les points faibles de chacune d'entre elles.

	Description	Points forts	Points faibles
Cascade	Dans ce modèle, le principe	• Facile à comprendre et à utiliser ;	• Tous les besoins doivent être

	est très simple : chaque phase se termine à une date précise par la production de certains documents ou logiciels. Ce modèle, développé dans les années 1970 par W. ROYCE a servi pendant des années de modèle de référence.	• adapté pour une équipe inexpérimentée ; • les limites de chaque étape sont visibles; • facilite un management du projet ; • la définition des besoins est non-évolutive • la qualité prime sur le coût.	bien spécifiés au départ ; • donne une fausse impression de l'avancée des travaux ; • pas d'interaction entre les phases de développement ; • l'intégration n'a lieu qu'à la fin du cycle ; • le client peut se retrouver non satisfait ; • pas de retour en arrière d'une phase à l'autre.
RUP : Rational Unified Process	• le RUP est à la fois une méthodologie et un outil prêt à l'emploi (documents types partagés dans un référentiel Web) ; • adapté pour des projets qui comportent plus de 10 personnes.	• Itératif • spécifie • le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes…	• coûteux à personnaliser ; • très axé processus, au détriment du développement : peu de place pour le code et la technologie
2TUP	• s'articule autour de • l'architecture • propose un cycle de développement en Y • cible des projets de toutes tailles	• itératif • fait une large place à la technologie et la gestion du risque • définit les profils des intervenants, les livrables, les plannings, les prototypes.	Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement…
XP	Méthode agile • ensemble de « Best practices » de	• itératives • simples à mettre en œuvre	une approche dénuée de modélisation

	développement (idéal pour le travail en groupe) • cible des projets de moins prototypes, règles de dix personnes	• fait une large place aux aspects techniques : prototypes, règles de développement, tests…	• ne couvre pas les phases en amont et en aval au développement : capture des besoins.
Scrum	La méthode s'appuie sur le découpage d'un projet en sprint », ainsi que l'auto-organisation de l'équipe de développement. Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé, et contribue à augmenter la valeur d'affaires du produit	offre la possibilité de développer uniquement les fonctionnalités qui apportent une valeur ajoutée au produit, en toute transparence, avec le souci de la qualité et du respect des délais, et avec un retour rapide de la part du client.	La méthode Scrum ne couvre aucune technique d'ingénierie du logiciel. Pour l'utiliser, il est nécessaire de la compléter avec des pratiques de qualité du logiciel. Par exemple, on pourra utiliser des pratiques issues de l'extrême Programming ou l'un des processus unifiés.

2.3 Choix de méthodologie : Scrum

Après l'étude comparative des différentes méthodologies, nous avons choisi Scrum pour les raisons suivantes que nous détaillerons ci-dessous :

En ce qui concerne la méthode Scrum:

- Elle convient aux équipes ayant un nombre de développeurs réduits. Ceci est le cas de notre projet ;
- le client est impliqué dans le développement de l'application : la consultation du client est nécessaire dès l'achèvement d'une tâche ;
- la progression des tâches s'effectue pendant une durée de développement courte ;
- elle est conforme aux réalités que nous avons trouvées au sein du groupe Silimax. Les projets sont répartis en équipe de 3 à 4 développeurs maximum dont un chef appelé scrum master qui coordonne le projet et veille au respect de l'application du processus Scrum tout au long du celui-ci. À la fin de chaque sprint, le produit est déployé dans un

serveur de test. Scrum n'étant pas complet, car elle se contente de préciser la manière par lequel le projet sera déroulé. Elle se veut donc être plus tôt un outil de gestion de projet. Pour ce fait donc nous lui avons couplé la méthode 2TUP qui elle se veut plus tôt comme un processus de développement logiciel.

1.3.1 Scrum

Scrum est une méthode agile pour la gestion de projets. Elle a été conçue pour améliorer la productivité en évitant de paralyser les équipes par l'emploi de méthodologies trop lourdes. Ken Schwaber et Jeff Sutherland ont mis au point les grands principes de Scrum au début des années 1990 [W5].

Le terme Scrum est emprunté au rugby et signifie mêler. L'utilisation de ce terme est une analogie au rugby du fait que le processus s'articule autour d'une équipe soudée, qui cherche à atteindre un but.

Dans cette méthode, on focalise l'équipe de façon itérative sur un ensemble de fonctionnalités. L'équipe doit réaliser ces fonctionnalités durant des itérations de l'ordre de 30 jours appelées Sprints. Un but à atteindre est défini dans chaque Sprint. On détermine un ensemble de fonctionnalités à implémenter. Le sprint aboutit à la livraison d'un produit proposant les nouvelles fonctionnalités.

Un ScrumMaster a pour rôle de réduire les perturbations extérieures et de résoudre les problématiques non techniques de l'équipe. Le client participe de façon active. Il priorise la réalisation des fonctionnalités du logiciel. Il a la possibilité de changer la liste des fonctionnalités désirées à condition qu'elles ne soient pas en cours de réalisation. La figure 3 présente de façon globale l'application du processus Scrum [W6].

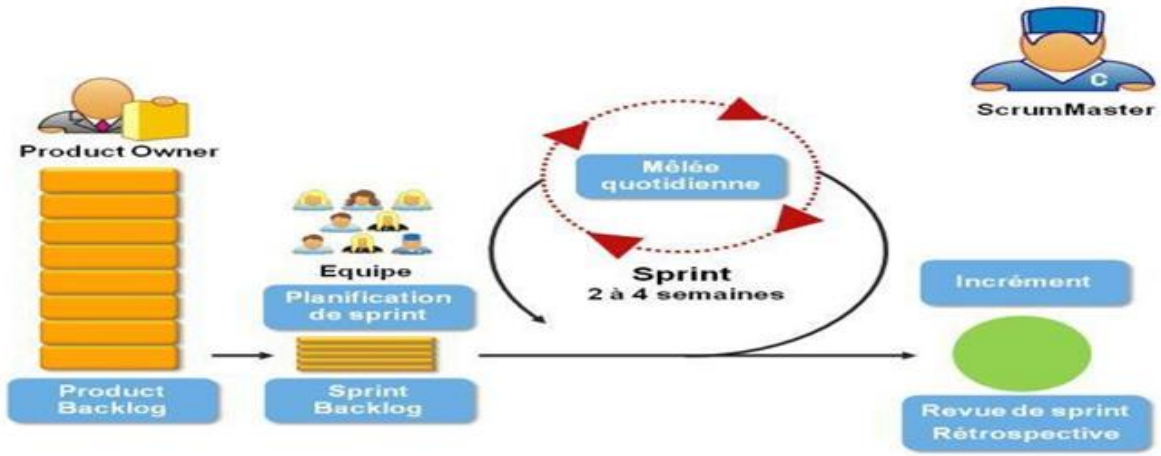

Figure 3: Processus Scrum

1.3.1.1 Les acteurs

On distingue plusieurs acteurs, dont :

- le directeur du produit (product owner) qui est le représentant des clients et des utilisateurs, et qui fait également parti de l'équipe ;
- le ScrumMaster qui veille à l'application de la méthodologie Scrum au sein de l'équipe.
- l'équipe qui contribue à la réalisation des fonctionnalités du projet (planification, développement, test et documentation). À noter que les membres de l'équipe travaillent tous ensemble : chaque membre peut faire ainsi des propositions, exprimer des idées.

1.3.1.2 Le processus

Tous les critères ou exigences du produit sont regroupés dans des journaux ou backlogs dont on distingue 2 types :

- le backlog de produit ou Product backlog : qui regroupe la liste des fonctionnalités du produit ;
- le backlog de sprint ou sprint backlog qui ; en fonction des fonctionnalités du produit, regroupe la liste des tâches qui devront être réalisées à l'itération en cours. Chaque tâche fait l'objet d'une estimation préalable de charge par l'ensemble des membres de l'équipe afin d'estimer au mieux les tâches que peut réaliser un sprint.

1.3.1.3 Planification :

- le sprint : dès le début d'un projet, la première planification permet de définir le périmètre de chaque itération appelé sprint [B1]. Chaque sprint dure quelques semaines et regroupe une liste de tâches (définies dans le backlog) ;
- la mêlée quotidienne : qui consiste chaque jour avec les membres de l'équipe ainsi que le directeur de produit de se tenir au courant de l'avancement du projet, notamment en :

  - faisant le point sur le travail effectué la veille par chacun ;
  - définissant les tâches qui sont réalisées durant la journée ;
  - résolvant les éventuels problèmes qui avait ou qui pourrait être rencontré par chacun ;

- ❖ le développement suit un processus itératif et incrémental : de nouvelles fonctionnalités sont rajoutées au produit.

1.3.1.4 La revue de sprint

La fin d'un sprint aboutit à la réalisation d'un produit avec des fonctionnalités partielles la documentation associée. Dans la plupart des cas, cela conduit à une revue de sprint consistant à faire une démonstration de la réalisation du sprint devant le client afin de valider le travail réalisé et d'avoir un retour pour éventuellement ajuster le backlog de produit [B1].

De façon globale, les valeurs mises en avant par scrum sont les suivantes :

- ➢ visibilité : Avoir une vision réelle sur le résultat ;
- ➢ inspection : Vérifier l'écart par rapport à l'objectif initial ;
- ➢ adaptation : S'adapter en fonction des écarts constatés afin de les ajuster.

1.3.2 Formalisme de modélisation (Langage UML)

UML se définit comme un langage de modélisation graphique et textuelle destiné à comprendre et décrire des besoins, les spécifier, concevoir des solutions et communiquer des points de vue [W3].

UML unifie à la fois les notations et les concepts objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage, c'est pour ça qu'UML est présenté parfois comme une méthode alors qu'il ne l'est pas.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la définition des besoins jusqu'au codage.

D'ailleurs, UML dans sa version 2 s'articule autour de treize (13) types de diagrammes qui peuvent être utilisés dans la conception d'un système. Elles sont regroupées dans deux grands ensembles que sont :

- ➢ les diagrammes structurels ou statiques : elles sont au nombre de six (6) et ont pour vocation de représenter l'aspect statique d'un système ;
- ➢ les diagrammes de comportement : elles sont au nombre de sept (7), ces diagrammes ;
- ➢ représentent la partie dynamique d'un système réagissant aux évènements et permettant de produire les résultats attendus par les utilisateurs.

La figure 6 illustre l'ensemble des treize (13) diagrammes d'UML [B3];

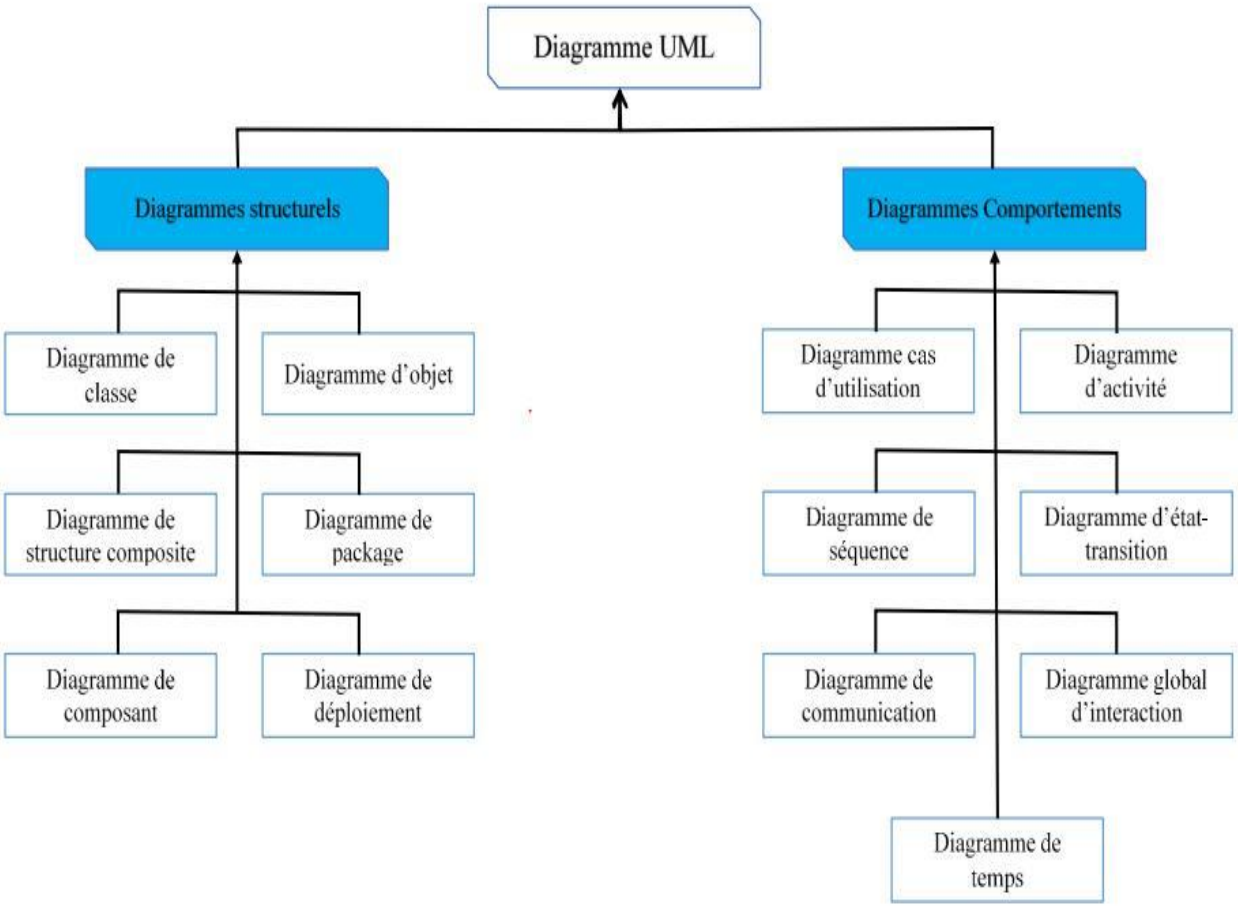

Figure 4: Diagramme UML [B3]

Dans ce chapitre nous décrite les différentes méthodologies de développement et choisit méthode qui s'adapte mieux à notre application.

Dans le chapitre suivant, nous allons parler les spécifications et analyses des besoins.

CHAPITRE 3 : SPÉCIFICATION ET ANALYSE DES BESOINS

Ce chapitre nous permet de mieux comprendre notre système de gestion de flotte de véhicules. Dans un premier temps, nous allons faire une description générale du fonctionnement de notre système, ensuite nous présenterons les diagrammes de cas d'utilisation.

3.1. Spécification des besoins

Après avoir identifié les cas d'utilisation et les acteurs de notre application, nous allons les représenter graphiquement via des diagrammes de cas d'utilisation.

Pour la modélisation des besoins, nous avons choisi le formalisme Unified Modeling Language (UML).

3.2.Diagramme de cas d'utilisation

3.2.1. Identifications des acteurs et leurs rôles

Les acteurs de notre application se répartissent dans les catégories suivantes :

Utilisateurs	Cas d'utilisation
Administrateur du système	**Gérer les comptes utilisateurs :** ajouter, modifier ou supprimer un compte. **Gérer les sociétés :** ajouter, modifier ou supprimer une société cliente.
Administrateur local	**Gestion des comptes utilisateurs locaux :** ajouter, modifier ou supprimer un compte. **Gestion des véhicules :** ajouter, modifier ou supprimer un véhicule. **Gestion des chauffeurs :** ajouter, modifier ou supprimer un chauffeur. **Gestion des points d'intérêt :** ajouter, modifier ou supprimer un point d'intérêt. **Gestion des lignes :** ajouter, modifier ou supprimer une ligne pour les véhicules de transport en commun. **Génération de rapports** : générer les rapports d'activité.
Gestionnaire	**Suivie des véhicules sur la carte** : localiser la position d'un véhicule en temps réel, consulter la trajectoire d'un véhicule. **Consultation de l'historique des trajets** : afficher la liste des trajets parcourus pour chaque véhicule, simuler les trajets parcourus sur la carte.
Receveur	**Gestion des ventes de tickets :** effectuer la vente de ticket pour les sociétés de transport en commun.

Tableau 1: Identification des acteurs et des cas d'utilisation

3.2.2. Les diagrammes de cas d'utilisation

3.2.2.1. Cas d'utilisation général

Maintenant que nous avons identifié les cas d'utilisation et leurs acteurs, nous allons les représenter graphiquement sur un diagramme de cas d'utilisation général (figure 7).

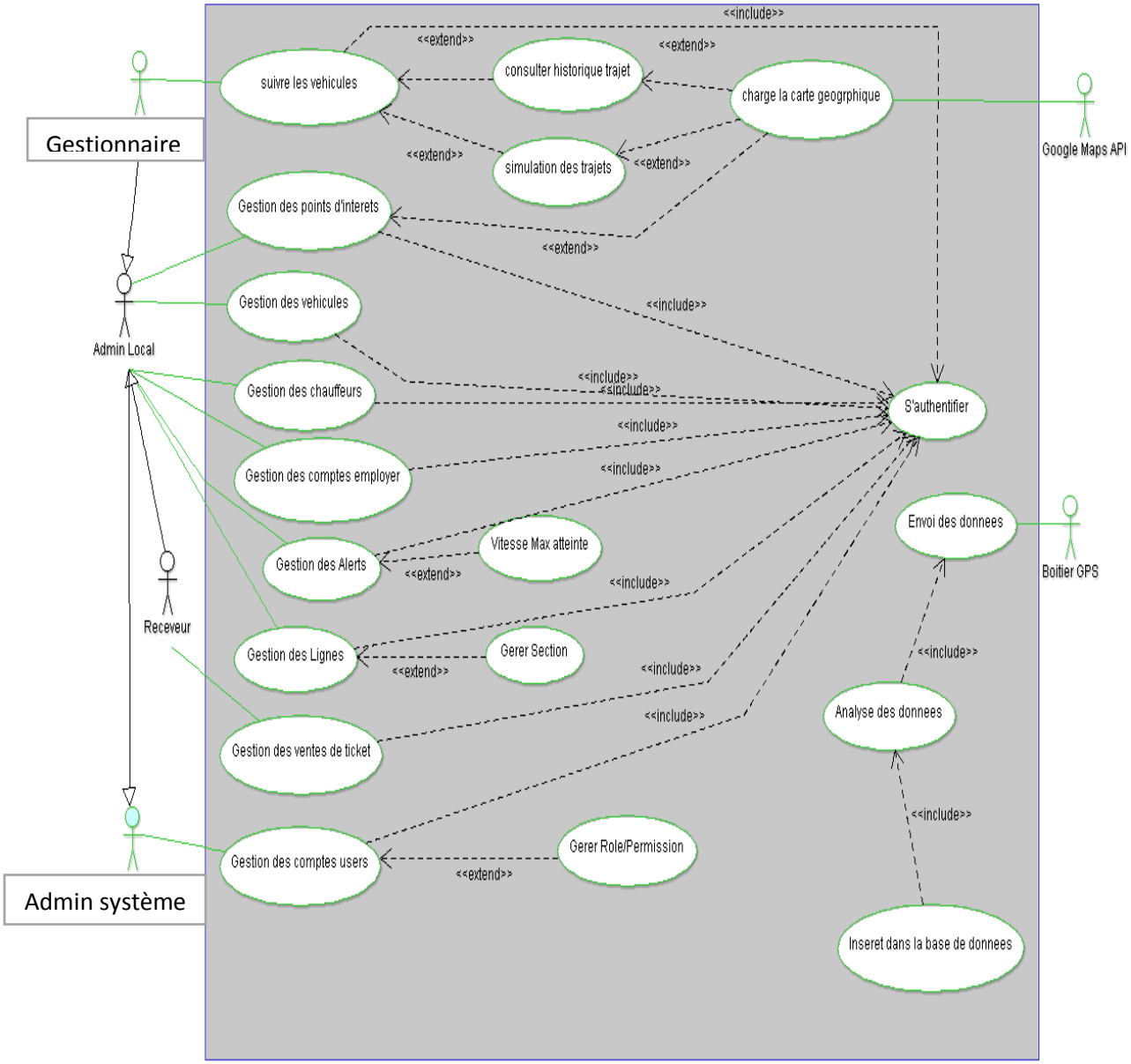

Figure 5: Cas d'utilisation général

3.2.2.2. Cas d'utilisation gestion des comptes

Ce cas d'utilisation permet la gestion des comptes utilisateurs et les actions possibles.

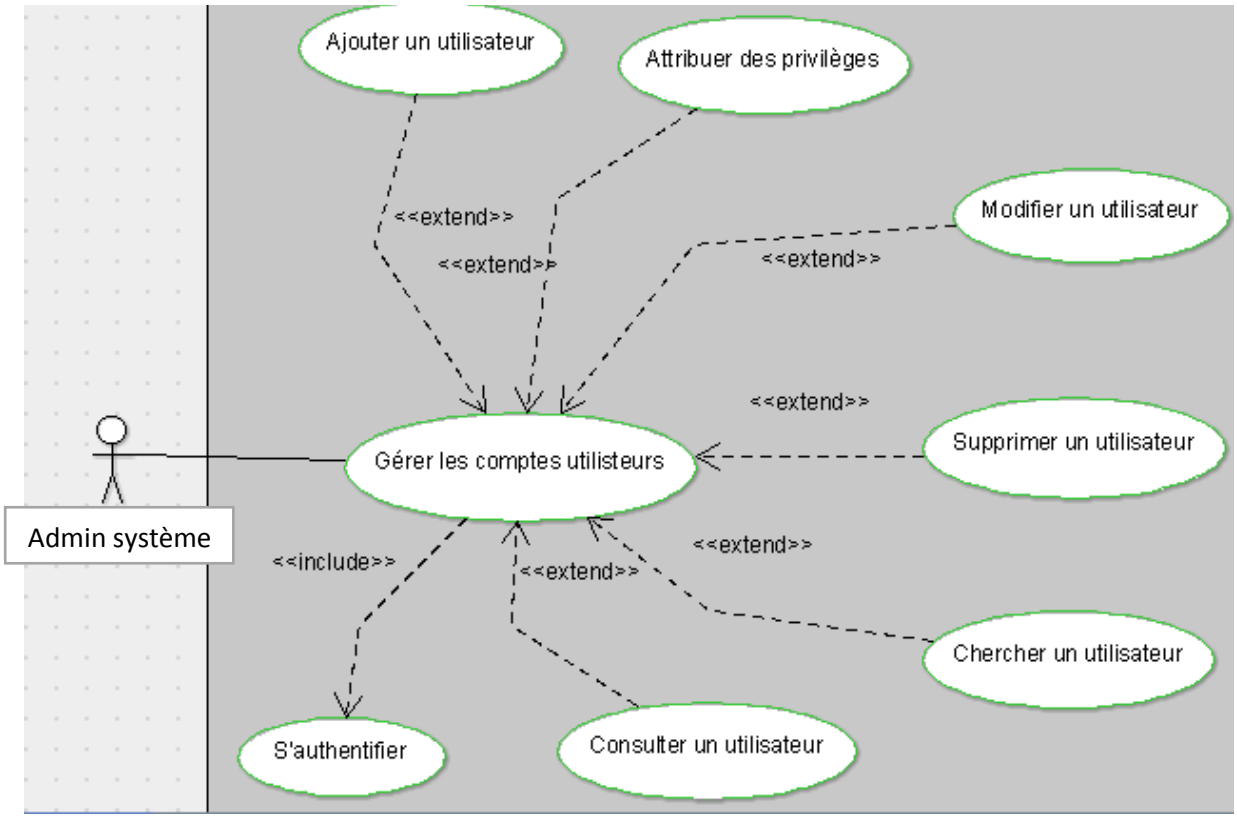

Figure 6: Cas d'utilisation gestion des comptes

3.2.2.3. Cas d'utilisation « visualiser et suivre les véhicules »

Ce cas d'utilisation permet de visualiser et de suivre les véhicules.

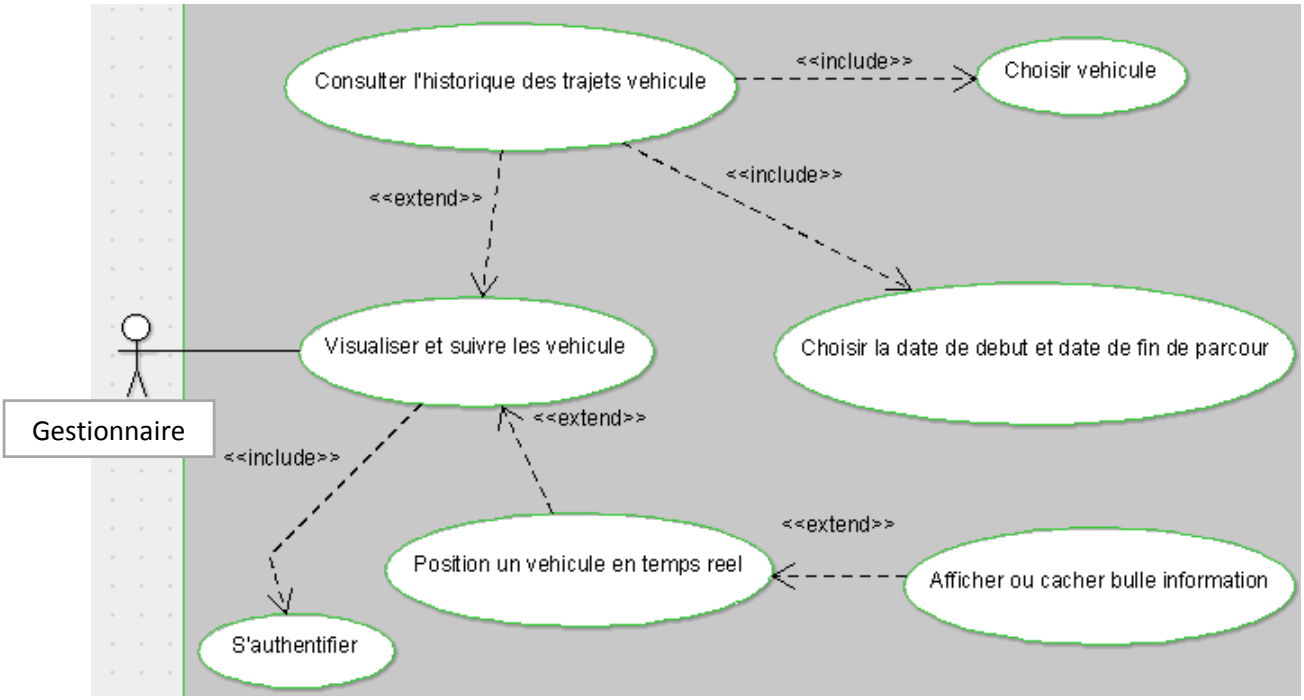

Figure 7: Cas d'utilisation « visualiser et suivre un véhicule »

1.3.2.1 Cas d'utilisation « Suivie des véhicules sur la carte »

Ce module permet de « Suivie des véhicules sur la carte »

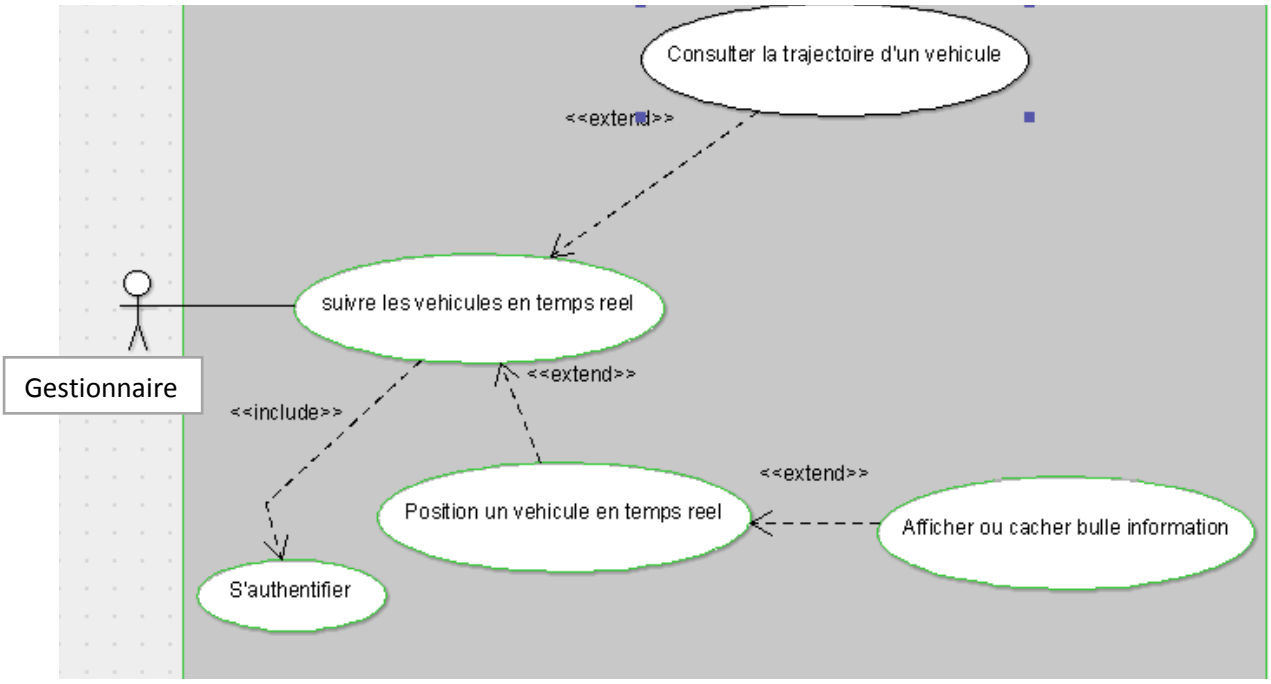

Figure 8: Diagramme de cas d'utilisation « Suivie des véhicules sur la carte »

3.2.2.4. Cas d'utilisation « Vente des tickets »

Ce module permet la Gestion de la vente des tickets

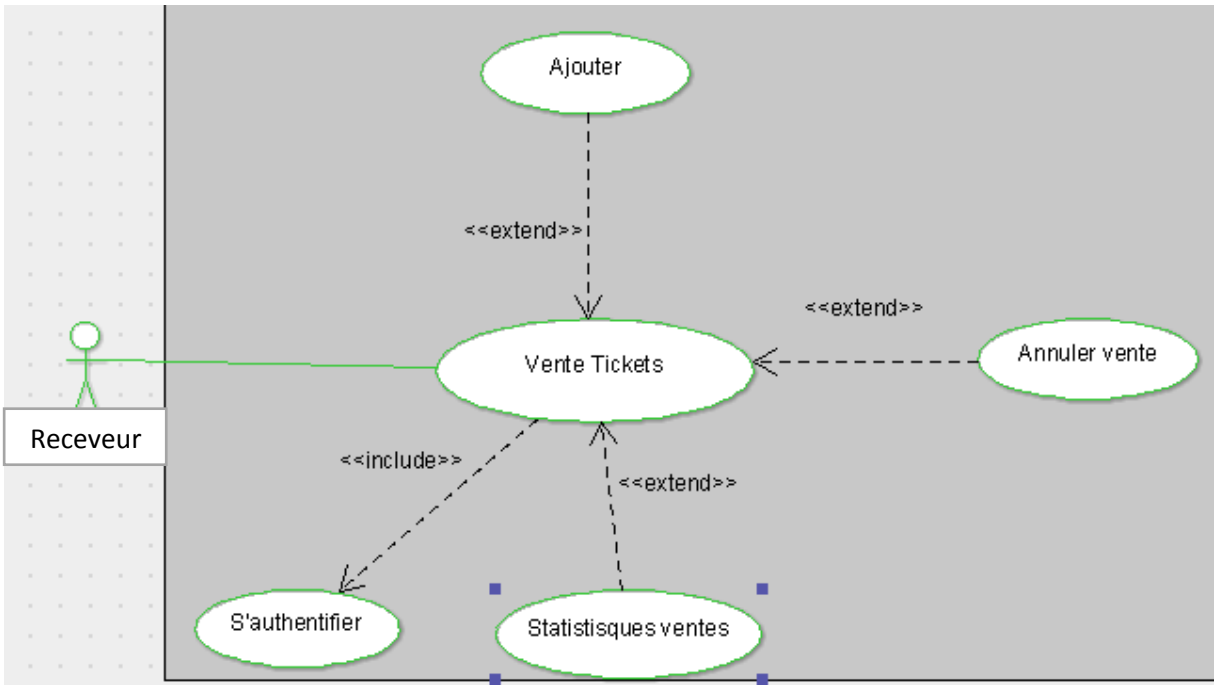

Figure 9: Diagramme de cas d'utilisation Vente des tickets

3.2.2.5. Description de quelques cas d'utilisation

3.2.2.5.1. Cas d'utilisation «s'authentifier »

Ce cas d'utilisation permet à un utilisateur de se connecter au système afin d'accéder à des fonctionnalités selon son profil.

Description du cas d'utilisation «s'authentifier »	
Titre	S'authentifier
Résumé	Permets à un utilisateur d'être identifié par le système.
Acteur (s)	Administrateur du système, administrateur local, gestionnaire, receveur
Pré condition	Il faut que chaque utilisateur ait un compte.
Scénario nominal	➢ L'utilisateur saisit son login et son mot de passe. ➢ Le système vérifie les informations. ➢ Il récupère le profil de l'utilisateur.
Post condition	Accéder à la page des fonctionnalités selon le profil de l'utilisateur
Exception	Si l'utilisateur saisit un login ou un mot de passe incorrect.

Tableau 2: Cas d'utilisation « s'authentifier »

3.2.2.5.2. Cas d'utilisation «ajouter un utilisateur »

Ce cas d'utilisation permet à l'administrateur d'ajouter des utilisateurs du système.

Description du cas d'utilisation «ajouter utilisateur »	
Titre	Ajouter un utilisateur
Résumé	Permet d'ajouter un ou de nouveau(s) utilisateur(s) au système.
Acteur (s)	Administrateur du système, administrateur local
Pré condition	Authentification
Scénario nominal	➢ L'administrateur choisit d'ajouter un nouvel utilisateur. ➢ Il remplit le formulaire et le valide. ➢ Le système procède à une vérification des données. ➢ Le système enregistre ensuite les données de l'utilisateur.
Post condition	Message de confirmation de l'enregistrement effectué
Exception	Si les données saisies sont invalides.

Tableau 3: Cas d'utilisation « ajouter un utilisateur »

3.2.2.5.3. Cas d'utilisation «ajouter un véhicule »

Ce cas d'utilisation permet à l'administrateur d'ajouter des véhicules dans le système.

Description du cas d'utilisation «ajouter un véhicule »	
Titre	Ajouter un véhicule
Résumé	Permets d'ajouter des véhicules
Acteur (s)	Administrateur local
Pré condition	Authentification
Scénario nominal	➢ L'administrateur choisit d'ajouter un nouveau véhicule dans la base de données. ➢ Il remplit le formulaire et le valide. ➢ Le système procède à une vérification des données. ➢ Il enregistre ensuite les données sur le véhicule.
Post condition	Message de confirmation de l'enregistrement effectué
Exception	Si les données saisies sont invalides.

Tableau 4: Cas d'utilisation « ajouter un véhicule »

3.2.2.5.4. Cas d'utilisation « localiser un véhicule »

Ce cas d'utilisation permet au gestionnaire de localiser un véhicule.

Description du cas d'utilisation «localiser un véhicule »	
Titre	Localiser un véhicule
Résumé	Permets de visualiser un véhicule sur une carte
Acteur (s)	Gestionnaire
Pré condition	Authentification
Scénario nominal	➢ Le gestionnaire choisit le ou les véhicules qu'il veut localiser et valide. ➢ Il visualise sur la carte la position en temps réel du ou des véhicules sélectionné(s).
Post condition	Vérification de la connexion
Exception	Si problème de connexion

Tableau 5: Cas d'utilisation « localiser un véhicule »

3.2.2.5.5. Cas d'utilisation « simuler un trajet »

Ce cas d'utilisation permet au gestionnaire de simuler le trajet d'un véhicule grâce à l'historique.

Description du cas d'utilisation «simuler un trajet »	
Titre	Simuler un trajet
Résumé	Permets de simuler le trajet parcouru par un véhicule sur une carte
Acteur (s)	Gestionnaire
Pré condition	Authentification
Scénario nominal	➢ Le gestionnaire choisit le véhicule, la date de début et la date de fin du trajet, puis il valide. ➢ Il visualise sur la carte le trajet parcouru entre ces dates.
Post condition	Serveur introuvable
Exception	Si problème de connexion

Tableau 6: Cas d'utilisation « simuler un trajet »

3.3. Analyse des besoins fonctionnels

Le diagramme de séquence permet de représenter les interactions entre objets en indiquant la chronologie des échanges. Ces interactions sont représentées par des messages qui sont envoyés de l'acteur vers le système et vice-versa. Dans ce qui suit, nous allons identifier les acteurs et leurs fonctionnalités et présenter les diagrammes de séquence pour quelque cas d'utilisation de notre système.

3.3.1. Identification des acteurs et leurs fonctionnalités

L'étude préliminaire liste les besoins de la structure. En réponse aux besoins ou aux problèmes posés par Silimax, un ensemble de fonctionnalités est identifié (tableau 7).

Fonctionnalités du système	**Acteurs**
S'authentifier	Admin système, admin local, gestionnaire, receveur
Créer un compte	Admin système
Créer une société	Admin système

Créer un compte employé	admin local
Consulter un compte	Admin système
Consulter un compte employé	admin local
Consulter une société	Admin système
Supprimer un compte	Admin système
Supprimer une société	Admin système
Supprimer un compte employé	Admin local
Modifier un compte	Admin système
Modifier une société	Admin système
Modifier un compte employé	admin local
Suivre un véhicule	Gestionnaire
Faire l'historique des trajets	Gestionnaire
Vendre les tickets	Receveur
Consulter les ventes	Admin local, Receveur
Annuler une vente	Receveur
Ajouter une ligne	Admin local
Consulter les lignes	Admin local
Supprimer une ligne	Admin local
Modifier une ligne	Admin local
Ajouter une section	Admin local
Consulter les sections	Admin local
Supprimer une section	Admin local
Modifier une section	Admin local
Affecter un chauffeur et un receveur a un véhicule et une ligne	Admin local
Ajouter un point d'intérêt	Admin local
Consulter les points d'intérêts	Admin local
Supprimer un point d'intérêt	Admin local
Modifier un point d'intérêt	Admin local
Ajouter un chauffeur	Admin local
Consulter les chauffeurs	Admin local
Supprimer un chauffeur	Admin local
Modifier un chauffeur	Admin local

Ajouter un véhicule	Admin local
Consulter les véhicules	Admin local
Supprimer un véhicule	Admin local
Modifier un véhicule	Admin local

Tableau 7: Identification des acteurs et leurs fonctionnalités

3.3.2. Analyse de « s'authentifier »

Le diagramme de séquence ci-dessous illustre le scénario nominal du cas d'utilisation « s'authentifier ». Un acteur se connecte au système et donne son nom d'utilisateur (login) et son mot de passe. Le système vérifie l'identité de l'acteur et autorise ou refuse la connexion.

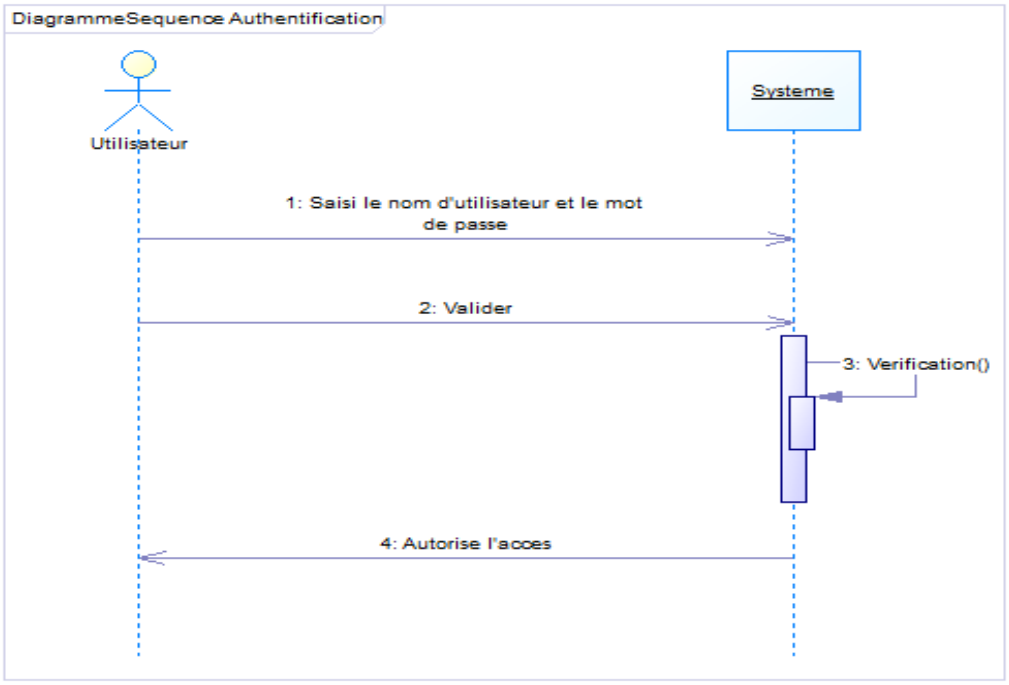

Figure 10: Diagramme de séquence « s'authentifier »

3.3.3. Analyse de la gestion des comptes utilisateurs

L'administrateur demande la liste des comptes utilisateurs. Le système affiche la liste avec des informations concernant le nom et le prénom, le nom utilisateur, le mot de passe, le type de compte et d'autres informations personnelles.

L'administrateur édit la liste en ajoutant un compte utilisateur. Ensuite il valide les modifications qui seront enregistrées par le système.

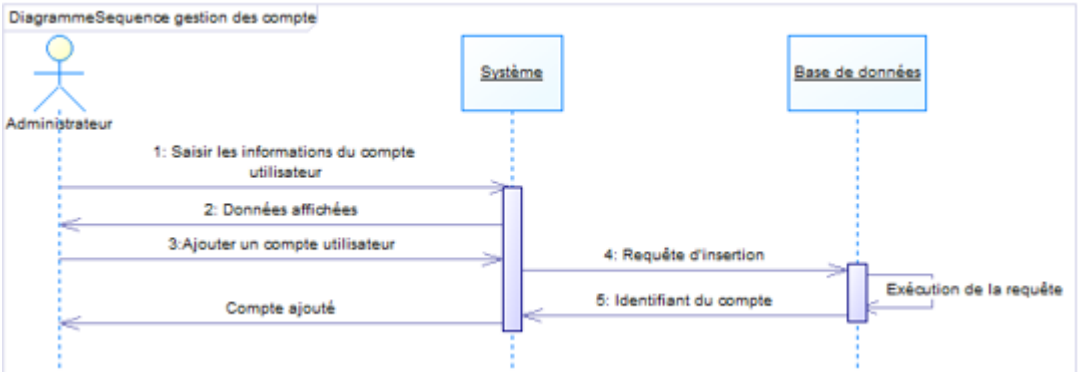

Figure 11: Diagramme de séquence gestion des comptes

3.3.4. Analyse de la Vente de Ticket

Pour vendre un ticket, l'utilisateur connecté choisit une section et le système enregistre les informations de la section.

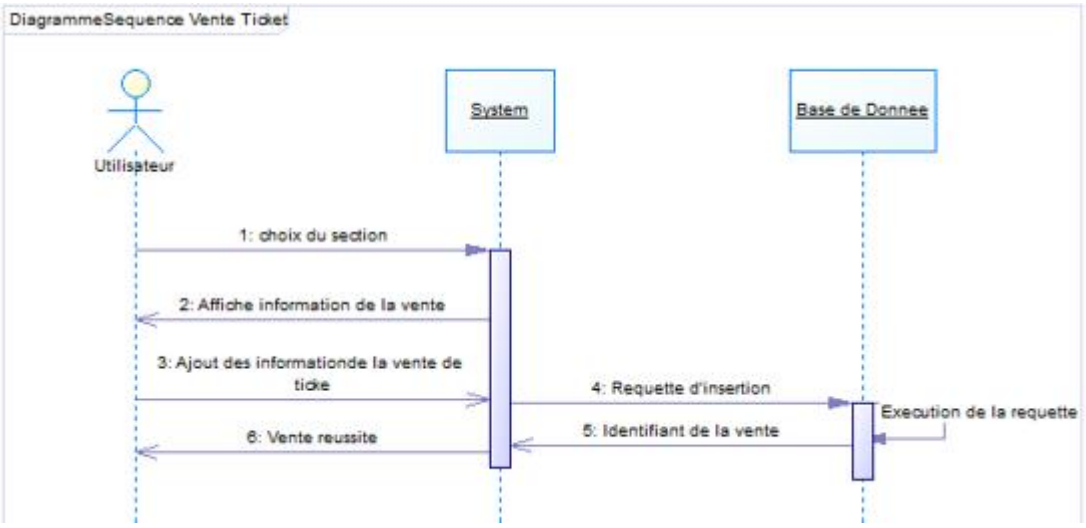

Figure 12: Diagramme de séquence « Vente de Ticket »

3.3.5. Analyse des coordonnées GPS

Les informations envoyées par une balise sont enregistrées automatiquement en base par le système.

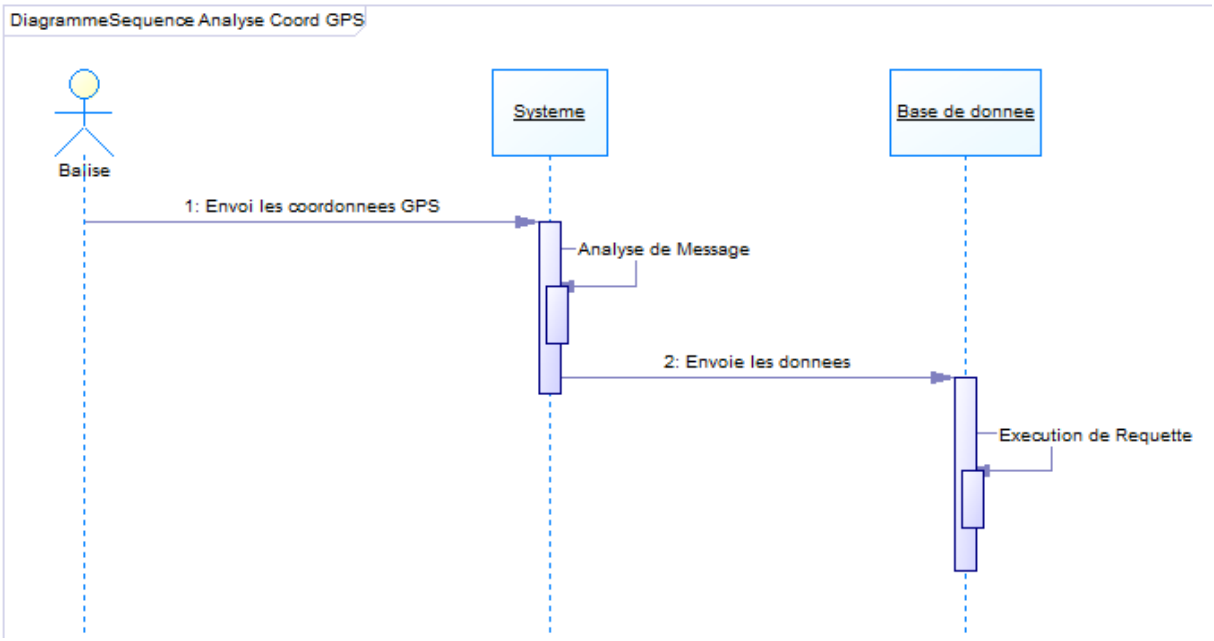

Figure 13: Diagramme de séquence « Analyse des coordonnées GPS »

3.4.Diagramme d'activité

Le diagramme d'activité est un moyen graphique pour donner une vision de l'ensemble des actions. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. La figure 14 suivante représente celui de la gestion des sociétés.

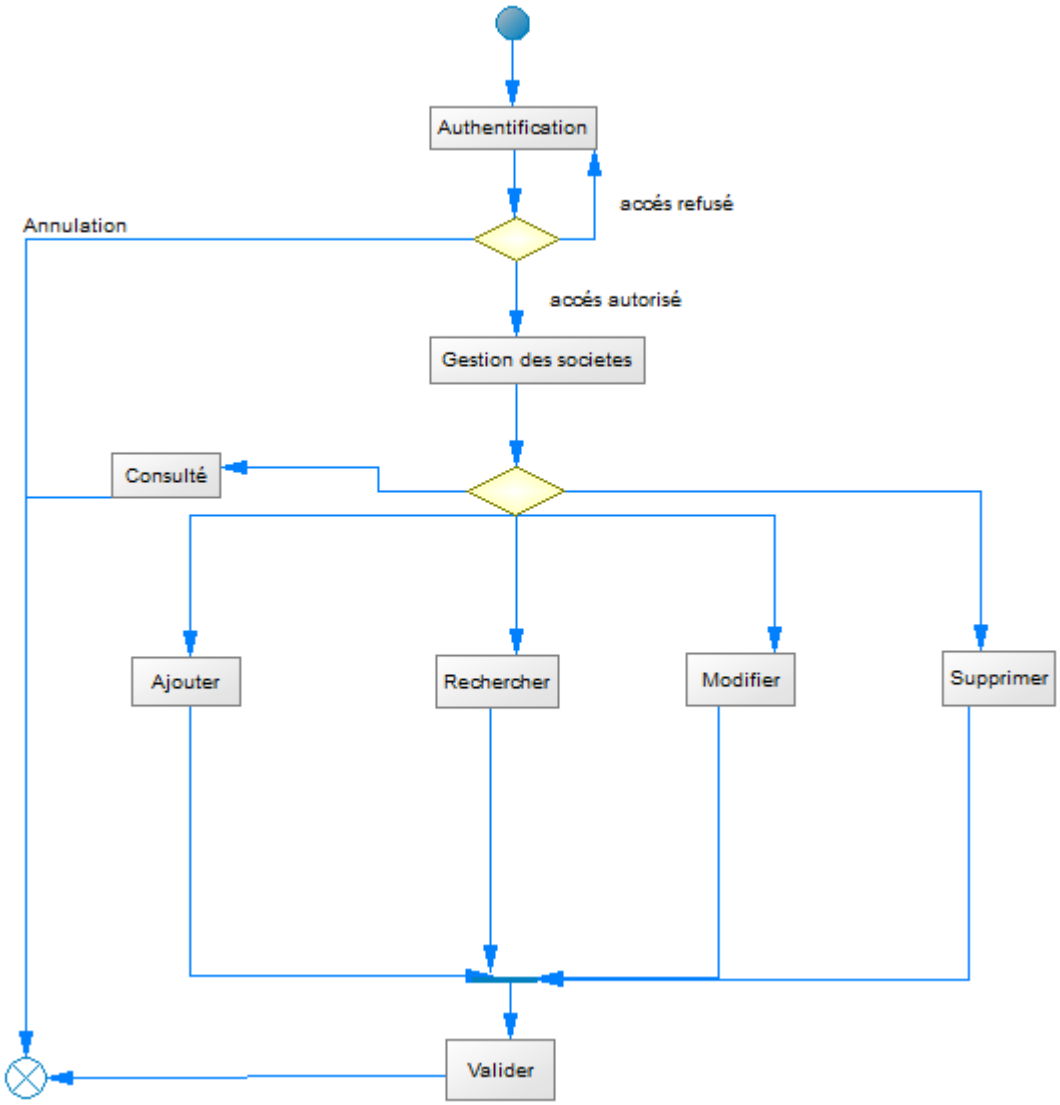

Figure 14: Diagramme d'activité gestion des sociétés

La spécification et l'analyse des besoins nous ont permis de mieux cerner notre sujet, d'identifier les différents acteurs et les fonctionnalités que doit offrir notre application. Des exemples de fonctionnalités sont ensuite décrits.

Dans le chapitre suivant, nous allons présenter la phase de conception de notre application.

CHAPITRE 4 : CONCEPTION

Dans ce chapitre, nous allons montrer les interactions entre notre système et ses utilisateurs en présentant le modèle conceptuel de donnée et ensuite les diagrammes de séquence du système.

4.1. Conception Générale de la solution

Cette section présente l'architecture technique et applicative à travers les différentes entités du système et leurs interactions. À cet effet, nous fournissons les diagrammes de classes et de conception.

4.1.1. Conception architecturale

L'architecture décrit d'une manière symbolique et schématique les différents éléments d'un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions. Contrairement aux spécifications produites par l'analyse fonctionnelle, le modèle d'architecture, produit lors de la phase de conception, ne décrit pas ce que doit réaliser un système informatique, mais plutôt comment il doit être conçu de manière à répondre aux spécifications [W6].

Nous allons à cet effet voir dans les prochains sous point les systèmes architecturaux de notre système.

La figure 16 ci-dessous nous montre l'architecture générale de notre système.

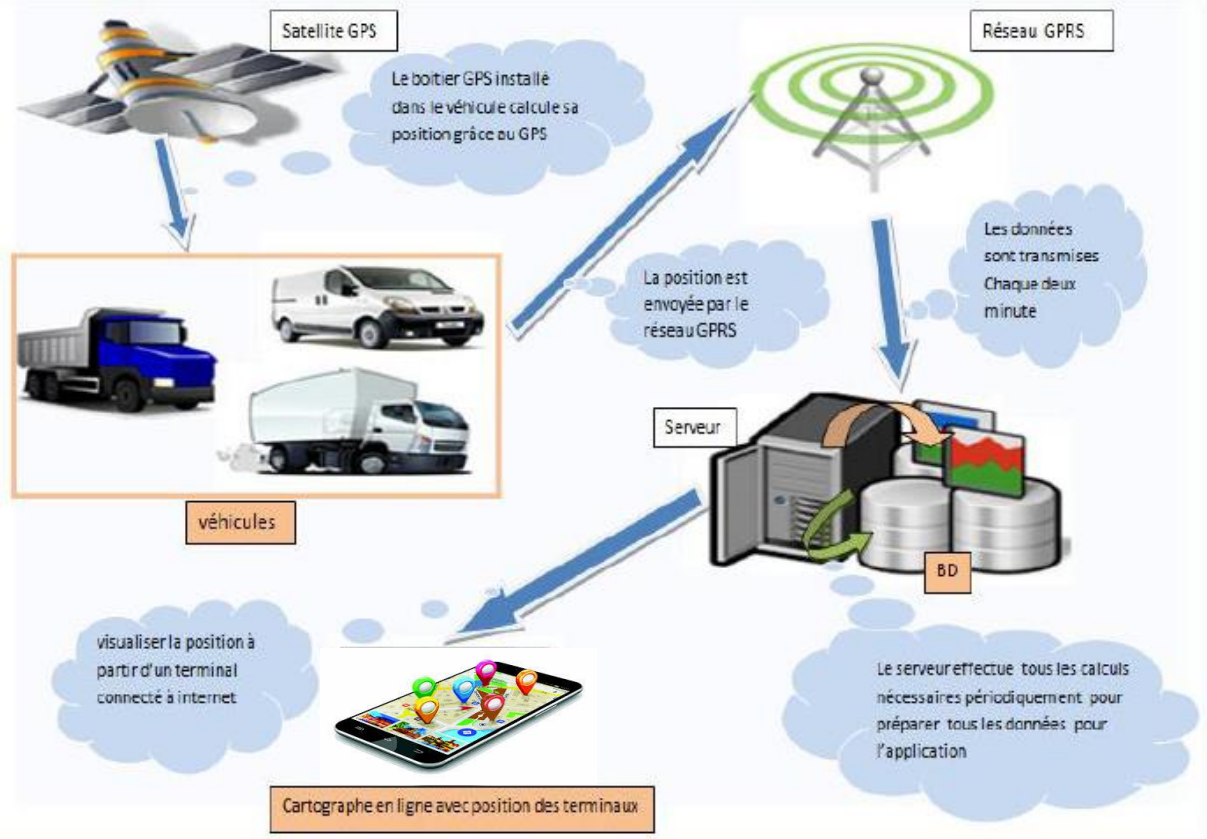

Figure 15: Description générale du système

4.1.2. Architecture de l'application web pour la suivi des voitures

C'est une architecture logicielle inhérente aux différents aspects fonctionnels d'une application d'entreprise, à savoir : l'aspect Présentation, l'aspect métier et l'aspect Données. Le but de la conception en général, est d'attribuer des responsabilités. Définir des couches est un moyen de séparer les responsabilités et ainsi de minimiser l'impact du changement. La figure suivante représente l'architecture applicative de notre système en intégrant le modèle MVC ainsi que l'ORM entre le modèle et le serveur de base de données.

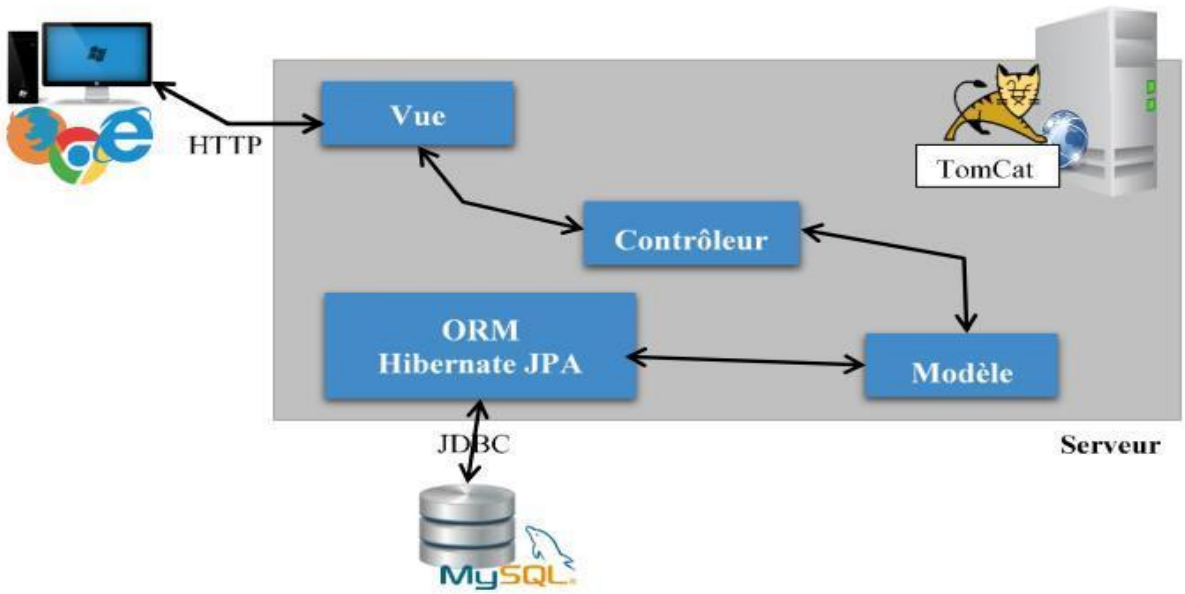

Figure 16: Architecture applicative

Commentaires :

- Le modèle (modèle de données) : il représente le cœur de l'application : traitement de données, interaction avec une base de base de données, etc. Il décrit les données manipulées par l'application. Le modèle regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour la mise à jour de ces données (insertion, suppression, etc.). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par ce modèle ne s'occupent pas de la présentation.
- La vue : présentation ou interface utilisateur représente l'interface par laquelle interagit l'utilisateur de la plateforme. Ces tâches sont de présenter les résultats renvoyés par le modèle ainsi que toutes les actions de l'utilisateur. Ces différentes actions sont envoyées 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 utilisateurs.
- Le contrôleur (logique de contrôle, gestion des évènements, synchronisation) : il joue le rôle de chef d'orchestre. C'est lui qui prend en charge la gestion des évènements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Si les actions

de l'utilisateur nécessitent le chargement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jour.

4.1.3. Diagramme de déploiement

Le diagramme de déploiement nous permet d'établir le lien avec l'architecture physique (figure 18 ci-dessous).

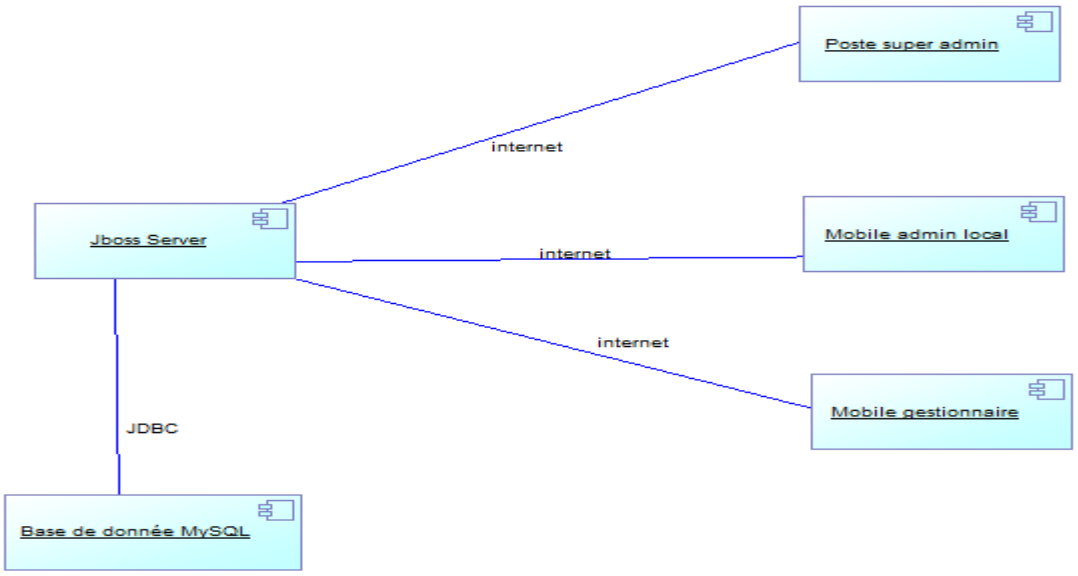

Figure 17: Diagramme de déploiement

4.1.4. Diagramme de package

Un diagramme de packages est un diagramme UML qui fournit une représentation graphique de haut niveau de l'organisation de notre application, et nous aide à identifier les liens de généralisation et de dépendance entre les packages (figure 19).

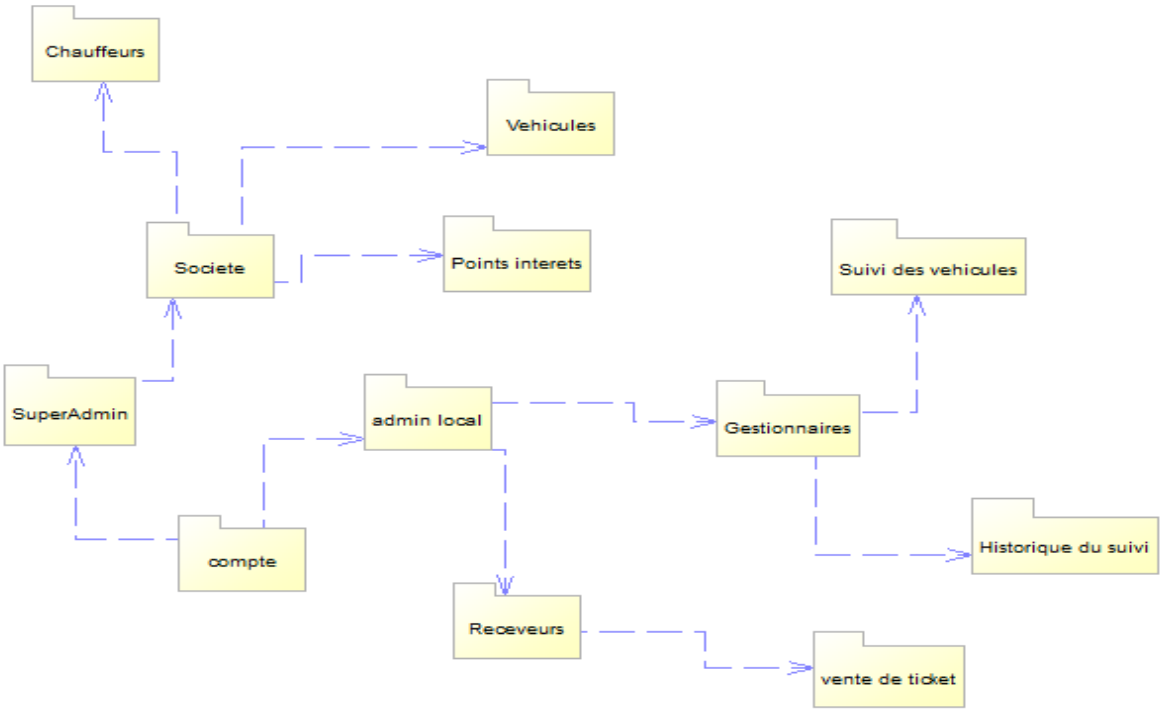

Figure 18: Diagramme de package

4.2. Conception détaillée

Cette section présente l'architecture technique et applicative à travers les différentes entités du système et leurs interactions. A cet effet, nous fournissons les diagrammes de classes de conception et le dictionnaire de données.

4.2.1. Diagramme de classe

Dans le chapitre précèdent nous avons vu que les diagrammes des cas d'utilisation permettent de modéliser à quoi sert le système. En fait, ils aident à répondre à la question « Quoi ? ». Un système peut être considéré comme un ensemble d'objets interagissant entre eux et avec les acteurs pour réaliser les cas d'utilisations.

Le modèle conceptuel de donnée nous permet de comprendre le lien entre les objets de notre système.

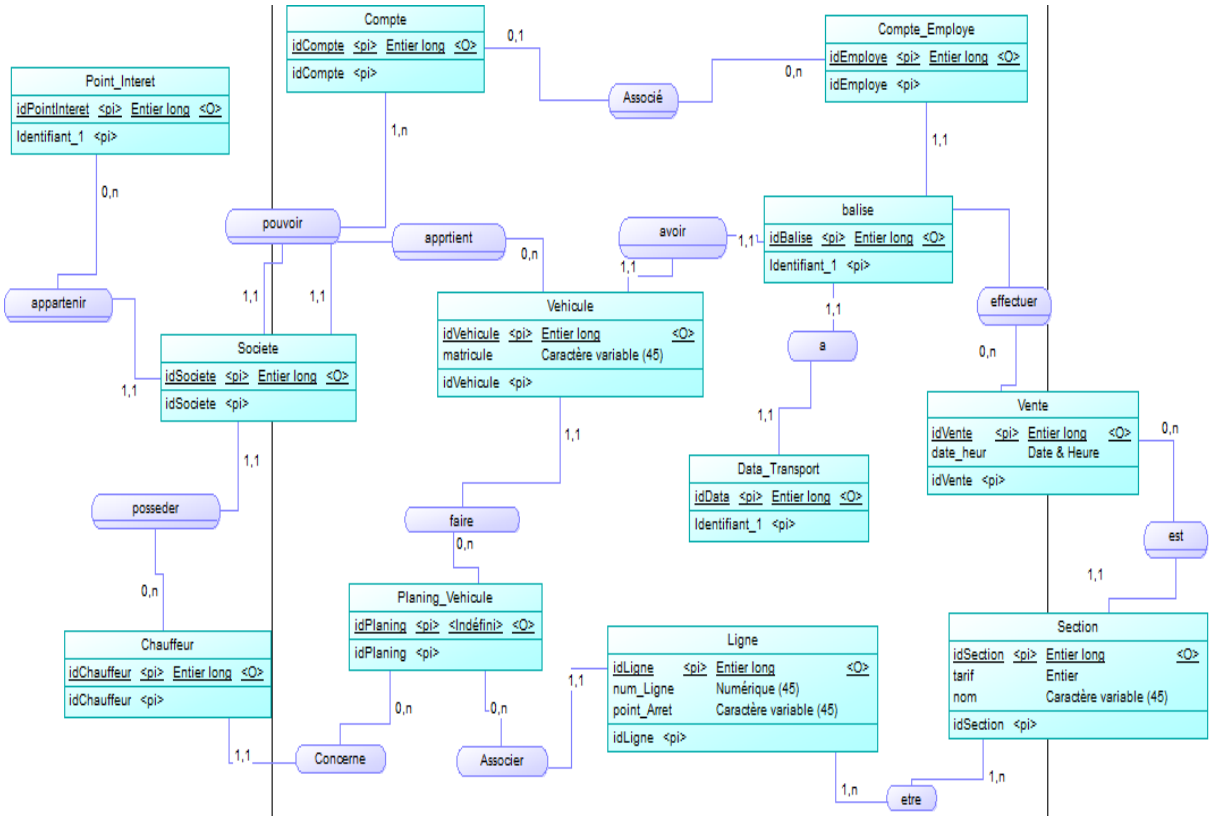

Figure 19: Diagramme de classe

4.2.2. Dictionnaire de données

Nom des Classe	Nom symbolique	Signification	Type de retour
Compte	id	Identifiant du compte	Long
	dateCreation	Date de création du compte	Date
	adresse_compte	Adresse de l'utilisateur	String
	cni_compte	Numéro carte national d'identité de l'utilisateur	String
	email_compte	Email de l'utilisateur	String
	login_compte	Login de l'utilisateur	String

	nom_compte	Nom de l'utilisateur	String
	password_compte	Mot de passe de l'utilisateur	String
	prenom_compte	Prénom de l'utilisateur	String
	Tel_compte	Numéro de téléphone de l'utilisateur	Int
	type_compte	Type de compte (Transport ou particulier)	String
	cle_societe	Identifiant de la class société	Long
Compte_Employe	id	identifiant	Long
	dateCreation	Date de création du compte	Date
	adresse_employe	Adresse de l'utilisateur	String
	cni	Numéro carte national d'identité de l'utilisateur	String
	dateEmbauche	Date d'embauche de l'utilisateur	Date
	email_employe	Email de l'utilisateur	String
	login_employe	Login de l'utilisateur	String
	nom_emplye	Nom de l'utilisateur	String
	password_employe	Mot de passe de l'utilisateur	String
	prenom_employe	Prénom de l'utilisateur	String
	telephone_employe	Numéro de téléphone de l'utilisateur	Int
	type_compte_employe	Type de compte (Gestionnaire ou receveur)	String
	cle_compte	Identifiant de la class compte	Long
chauffeur	id	Identifiant du chauffeur	Long
	dateCreation	Date de création du chauffeur	Date
	adresse_chauf	Adresse du chauffeur	String
	cni	Numéro carte national d'identité du chauffeur	String
	date_Embauche	Date d'embauche du chauffeur	Date

	nom_chauf	Nom du chauffeur	String
	prenom_chauf	Prénom du chauffeur	String
	telephone_chauf	Numéro de téléphone du chauffeur	String
	cle_societe	Identifiant de class société	Long
Véhicule	id	Identifiant du véhicule	Long
	dateCreation	Date de création du véhicule	Date
	carburant_seuil	Niveau maximum de consommation de carburant	Double
	etat_vehicule	Etat du véhicule lors de l'achat	String
	modele_vehicule	Modèle du véhicule	String
	numero_Matricule	Numéro de matricule	String
	puissance_vehicule	Puissance du véhicule	Double
	type_vehicule	Type d'utilisation	String
	vitesse_seuil	Vitesse max autorise	Int
	cle_societe	Identifiant de la class société	String
Ligne	id	Identifiant de la ligne	Long
	dateCreation	Date de création de la ligne	Date
	arrive	Fin de la ligne	String
	depart	Départ de la ligne	String
	logueur	Longueur de la ligne	Double
	numero_ligne	Numero de la ligne	String
	cle_societe	Identifiant de la classe société	Long
vente	id	Identifiant de la vente	Long
	dateCreation	Date de création de la vente	Date
	date_vente	Date de création de la vente	
	section	La section concernant la vente	
	statusVente	Statuts de la vente (vendu ou annuler)	Boolean
	cle_compteEmployer	Identifiant de la class compte_Employe	Long
Data_Transport	id	Identifiant de la balise	Long
	dateCreation	Date d'envoi des données	Date

	latitude	Latitude	Double
	longitude	longitude	Double
	numero	Numéro de la balise	int
	cle_equipement	Identifiant de la class balise	Long
Section	id	Identifiant de la section	Long
	dateCreation	Date de création de la section	Date
	latitude	latitude	Double
	longitude	longitude	Double
	Nom_section	Nom de la section	String
Planning_vehicule	Id	Identifiant du planning	Long
	dateCreation	Date de création du planning	Date
	Date_debut	Début d'affectation	Date
	Date_fin	Fin d'affectation	Date
	cle_chauffeur	Identifiant de la class chauffeur	Long
	cle_vehicule	Identifiant de la class véhicule	Long
	cle_receveur	Identifiant de la class compte_employe	Long
	cle_ligne	Identifiant de la class ligne	Long
Balise	id	Identifiant	Long
	dateCreation	Date de la création	Date
	numero	Numéro de la balise	Int
	cle_vehicule	Identifiant de la class véhicule	Long
Point_Interet	id	Identifiant du point d'intérêt	Long
	dateCreation	Date de création du point d'intérêt	Date
	Description	Description du point d'intérêt	String
	latitude	latitude	Double
	logitude	longitude	Double
	cle_societe	Identifiant de la class société	Long
Societe	Id	Identifiant de la société	Long
	dateCreation	Date de création de la société	Date
	adresse_societe	Adresse de localisation de la société	String

	code_societe	Code d'identification de la société	String
	email_societe	Adresse email de la société	String
	fax_societe	Fax de la société	Int
	garant_societe	Personne qui représente la société	String
	nom_societe	Nom de la société	String
	siteWeb_societe	Site web de la société	String
	telephone_societe	Numéro de téléphone de la société	Int
	type_societe	Type de société (transport ou particulier)	String

Tableau 8: Dictionnaire des données

Ce chapitre nous a permis de bien comprendre l'interaction du système et les actions que l'utilisateur peut faire sur le système.

Dans le chapitre suivant, nous allons décrire la phase d'implémentation et présenter l'application.

CHAPITRE 5 : IMPLÉMENTATION ET PRÉSENTATION DE L'APPLICATION

Dans ce chapitre nous allons présenter notre solution en montrant dans un premier temps la conception de la solution ensuite l'environnement de développement et présenté quelque vue de notre application.

5.1. Architecture générale d'implémentation

Cette architecture fait abstraction de toute technologie. Elle représente toute les couches qui interviennent dans la réalisation de notre système. Le système peut être reproduit dans n'importe quelle autre technologie en s'appuyant sur cette architecture générique. La figure 20 représente l'architecture générique de notre système.

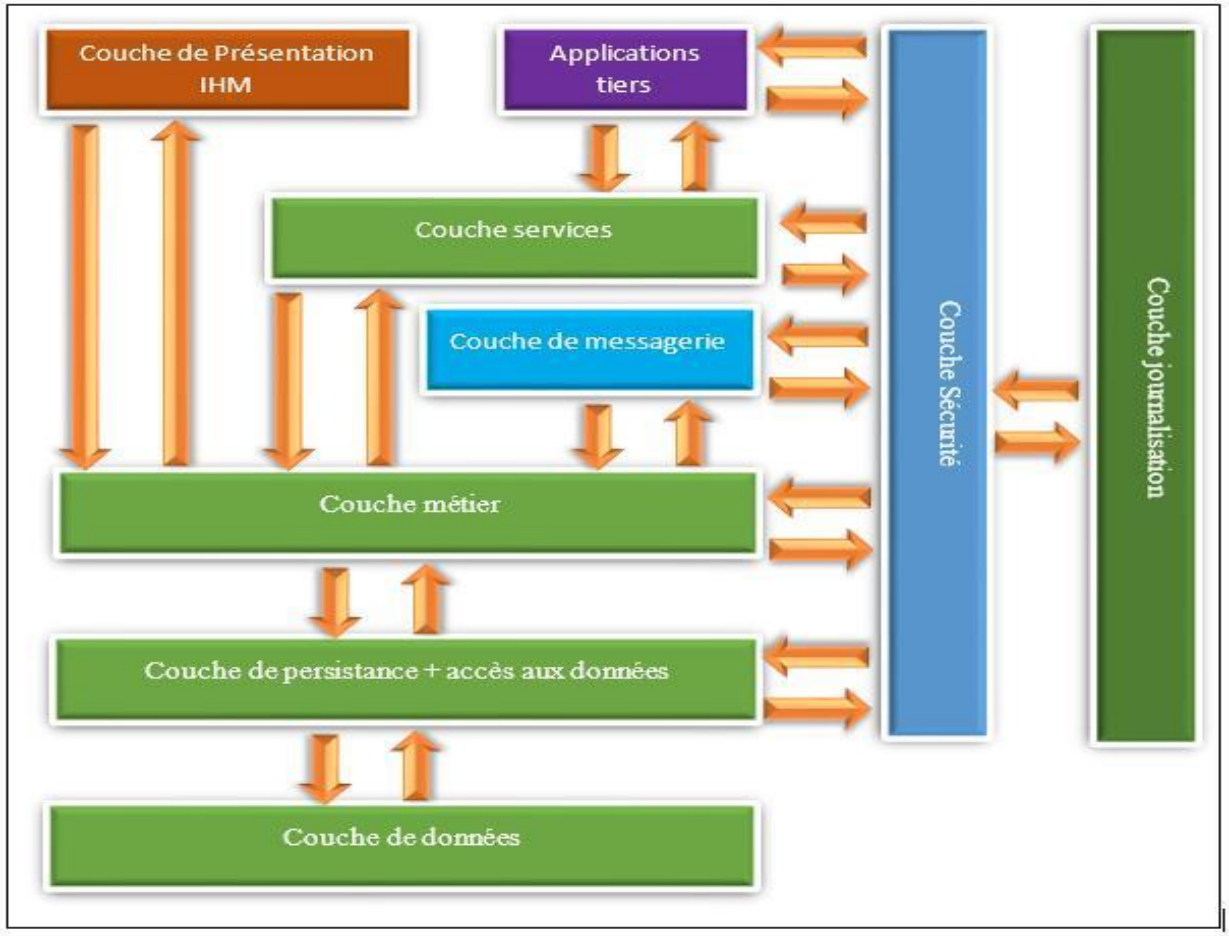

Figure 20: Architecture générique

Commentaires :

- La couche de présentation : c'est la couche par laquelle l'utilisateur accède au système à travers des interfaces web ou mobile.
- Applications tierces : cette couche représente les applications tierces avec lesquelles notre système va s'interfacer.
- La couche servie : cette couche nous permet de mettre en place tous les services de communications entre notre système et les applications tierces.
- La couche de messagerie : Nous utilisons cette couche essentiellement pour les envois de mails. Elle communique avec la couche métier du système ou les différentes méthodes d'envoi seront implémentées.
- La couche métier : C'est dans cette couche que sera implémenté la logique métier de l'application. Elle communique avec la couche accès aux données ou elle puisse ses ressources et avec la couche service aussi en lui offrant les éléments dont elle a besoin pour ses différentes opérations.
- La couche persistance + accès aux données : C'est à ce niveau que seras mis en place le modèle qui sera persister dans la base de données et aussi la définition du protocole d'accès aux données. Elle communique avec la couche de données et la couche métier.
- La couche de données : C'est la couche inférieure du système. C'est à ce niveau que seront sauvegardées les données qui seront utilisées par notre système.
- La couche de sécurité : c'est à ce niveau que sera mise en œuvre toute la politique de sécurité concernant le système. Cette politique concerne toutes les différentes couches du système.
- La couche de journalisation : ici, nous mettrons en œuvre le principe de non-répudiation. Elle nous permettra de savoir qui a fait quoi dans le système et quand. Cette couche communique avec la couche de sécurité et lui offre de ce fait des éléments supplémentaires pour sa mise en œuvre.

5.2. Environnement de développement

5.2.1. Niveau des ordinateurs

Afin de réaliser ce travail, on a eu recours à ces configurations :

- Configuration de l'ordinateur
  - Processeur : Intel core i3.
  - Fréquence d'horloge: 2.30 GHZ.

- Mémoire vive: 8 GO.
- Disque dur: 500 GO.
- Système d'exploitation : Windows 7 professionnel.

5.2.2. Niveau logiciel

- Power AMC

PowerAMC est une solution de modélisation et de gestion de métadonnées à la pointe de l'innovation, destinée aux architectures de données, aux architectures d'informations et aux architectures d'entreprise. PowerAMC est l'édition.

Française de PowerDesigner. La combinaison des techniques de modélisation et de la gestion des données confère à PowerAMC des fonctions uniques lui permettant de prendre en charge tous les environnements architecturaux. Le référentiel de métadonnées de PowerAMC permet également à toutes les parties prenantes de l'entreprise de collaborer et de communiquer efficacement. Ces dernières peuvent ainsi réagir plus rapidement face aux changements et garantir une meilleure capacité d'adaptation de l'entreprise.

- Éclipse Mars

Éclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement Mettant en œuvre n'importe quel langage de programmation. Éclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions. La spécificité d'Éclipse IDE vient du fait de son architecture totalement développée autour de la notion de plug-in (en conformité avec la norme OSGi «Open Services Gateway») , il est capable d'intégrer des Modules(Plugins) de base permettant de gérer des ensembles de ressources et faciliter le travail du programmeur.

- **JDK 1.7**

Le Java Développent Kit (JDK) désigne un ensemble De bibliothèques logicielles de base du langage de programmation Java, ainsi que l'environnement

dans lequel le code Java est compilé pour être transformé en bytecode afin que la machine virtuelle Java (JVM) puisse l'interpréter. Il existe en réalité plusieurs JDK, selon la plate-forme Java considérée (et bien évidemment la version de Java ciblée).

- **WampServer version 3.0.6**

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur apache et PHPMyAdmin pour gérer plus facilement notre base de données.

- **MySQL**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public que par des professionnels. MySQL est un serveur de base de données SQL très rapide, multithread, multi-utilisateur et robuste. Il est destiné aux missions stratégiques, aux systèmes de production à forte charge, et à l'intégration dans des logiciels déployés à grande échelle. MySQL est une marque déposée de MySQL AB [W4].

Les principaux concurrents de MySQL sont : PostgreSQL, Microsoft SQL Server, et Oracle. Ainsi, le choix de ce serveur de bases de données a été particulièrement déterminé par un certain nombre d'avantages qu'il offre aux développeurs. En effet, par rapport aux autres SGBD cités, MySQL est un logiciel intégrant un haut degré de portabilité, de sécurité et constitue un système de sauvegarde assez évolué avec utilisation optimale de ressources.

5.2.3. La plateforme JEE

L'élaboration de notre application s'appuie sur la plateforme JEE (Java Enterprise Edition) qui est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multiniveaux, basées sur des composants. On parle généralement de «plateforme JEE» pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution. J2EE comprend notamment :

- Les spécifications du serveur d'application, c'est-à-dire de l'environnement d'exécution : JEE définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Ces recommandations permettent ainsi à des entreprises tierces de développer des

serveurs d'application conformes aux spécifications ainsi définies, sans avoir à redévelopper les principaux services.

- Des services, au travers d'API, c'est-à-dire des extensions Java indépendantes permettant d'offrir en standard un certain nombre de fonctionnalités. Sun fournit une implémentation minimale de ces API appelées JEE SDK (J2EE Software développent Kit).
- Dans la mesure où J2EE s'appuie entièrement sur le langage Java, il bénéficie des

  avantages et inconvénients de ce langage, en particulier une bonne portabilité et une maintenabilité du code.

  Ce choix est justifié par plusieurs facteurs à savoir :

✓ La maturité et la richesse de cette technologie ;
✓ La possibilité de la réutilisation des différents composants qui en font partie ;
✓ La séparation forte qu'offrent la plupart des Frameworks relevant de cette architecture ;
✓ JEE dispose d'une documentation très riche, et l'ensemble des projets en JEE sont
✓ publiés dans le web, ce qui est très important

5.2.4. Framework Hibernate

Hibernate est un Framework qui permet de résoudre un très grand problème qui rencontre surtout les gens qui font la conception de leurs projets en UML et veulent implémenter leurs codes en java. On sait très bien que Java est un langage orienté objet qui permet facilement de codifier les classes UML. Or, cette conception suppose qu'on travaille avec une base de données orientée objet afin de pouvoir garder des concepts tels l'héritage, l'agrégation, les associations…etc. à moins si on voulait trouver une issue pour s'en échapper, et d'ailleurs c'est ça qu'on faisait toujours. On sait bien aussi que la plupart des SGBD sont relationnels. Donc la plus grande question qui se pose ici est la suivante : pourquoi je me suis cassé la tête pour faire la conception en UML ; alors que finalement je reviendrai au relationnel.

Hibernate est là pour résoudre ce problème. En effet, il permet aux développeurs d'interagir

avec la base de données relationnelle comme s'elle s'agissait d'une base de données orientée objet, et sans mettre en péril le diagramme de conception UML. En effet, Hibernate permet de créer une couche de persistance de données, c'est-à-dire une interface qui

donne aux développeurs des fonctionnalités pour l'interaction avec la base de données en 'mode' orienté objet. Les utilisateurs peuvent finalement voir le fruit de leurs conceptions et développer une vraie application orientée objet. Le principe de fonctionnement de Hibernate est simple ; pour chaque table on associe ce qu'on appelle un fichier de mapping (format XML) ; dans ce fichier on définit les champs de la table plus les règles de connexions avec les autres tables (héritages, associations…etc.) ; et pour chaque fichier de mapping on associe une classe Java que l'on peut générer automatiquement à partir du fichier de mapping. Le développeur n'a plus qu'à interagir avec la table dans la base de données, mais plutôt avec la classe Java. Pour cela Hibernate a mis en place un nouveau langage qui s'appelle HQL (Hibernate Query Language) qui est plus facile à utiliser et définit plus de fonctions pour interagir la base de données. La figure 20 décrit l'architecture du Framework Hibernate :

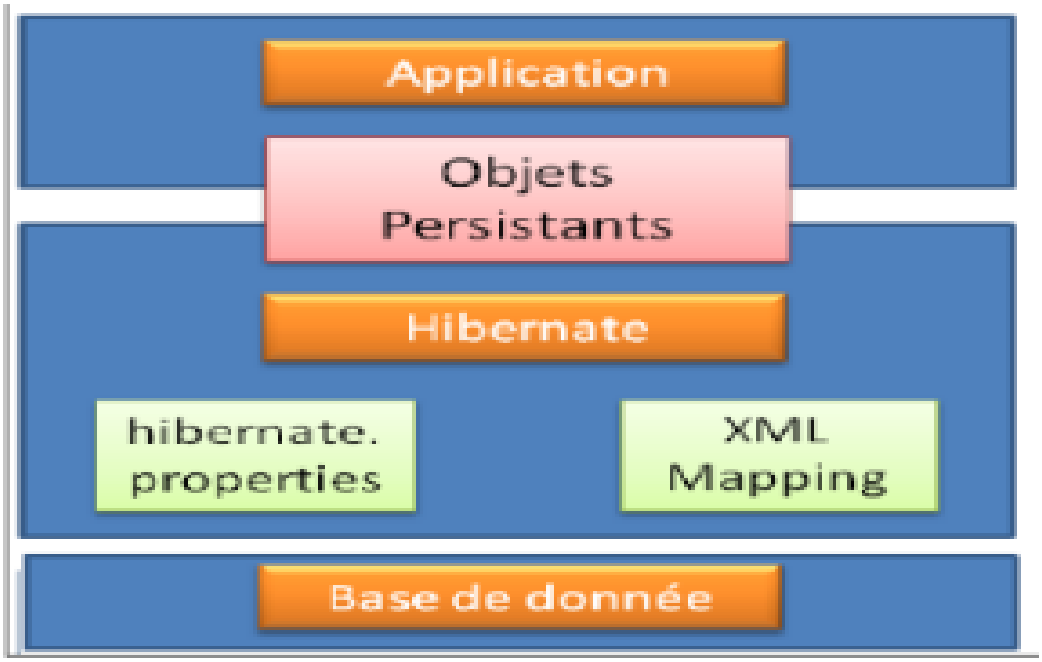

Figure 21: Architecture du framework Hibernate

5.3. Présentation de l'application

Pour une meilleure utilisation de notre application, nous l'avons scindé en deux parties :

- **la partie web** : elle constitue l'objet de mon mémoire ; j'ai développé tout ce qui concerne l'administration de l'application c'est-à-dire la gestion des utilisateurs, et la gestion des sociétés et des web services pour la gestion de l'application Android, permettant l'administration des utilisateurs, le suivi des

véhicules en temps réel, la gestion de la vente de tickets (selon le type de la société (transport ou particulier)) ;

- **la partie Android** : elle est réalisée par une autre étudiante de la même promotion en l'occurrence Salimata Diamanka ; elle permet de suivre les véhicules en temps réel sur une carte Google maps, d'effectuer la vente de tickets, etc.

5.4. Travail réaliser

5.4.1. Interface de connexion

L'interface de connexion est la page d'accueil de notre application. L'utilisateur doit introduire son identifiant et son mot de passe dans les champs correspondants pour pouvoir accéder aux différentes fonctionnalités de l'application.

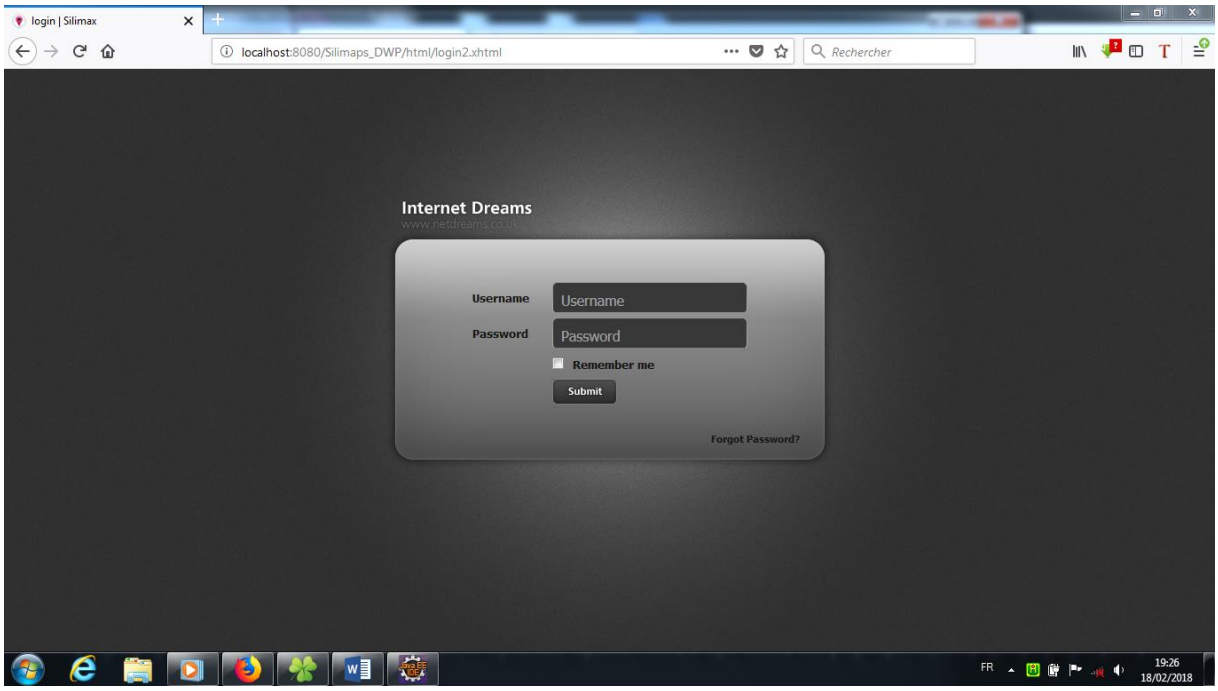

Figure 22: Interface de connexion

Une fois que le client a cliqué sur le bouton « Submit », le système vérifie les données entrées. En cas d'échec, il réaffiche la page d'authentification avec un message d'erreur. Sinon (le login et/ou mot de passe sont valides), le système lui renvoie sur son profil.

En cas d'oubli du mot de passe, le système lui envoie le formulaire de réinitialisation de mot de passe dans lequel il saisit son email. Il pourra ainsi réinitialiser son mot de passe via un email automatique généré par notre système.

5.4.2. Liste des sociétés

Cette interface présente la liste des sociétés avec toutes les informations les concertant et offre la possibilité de faire certaines actions comme la modification ou la suppression d'une société.

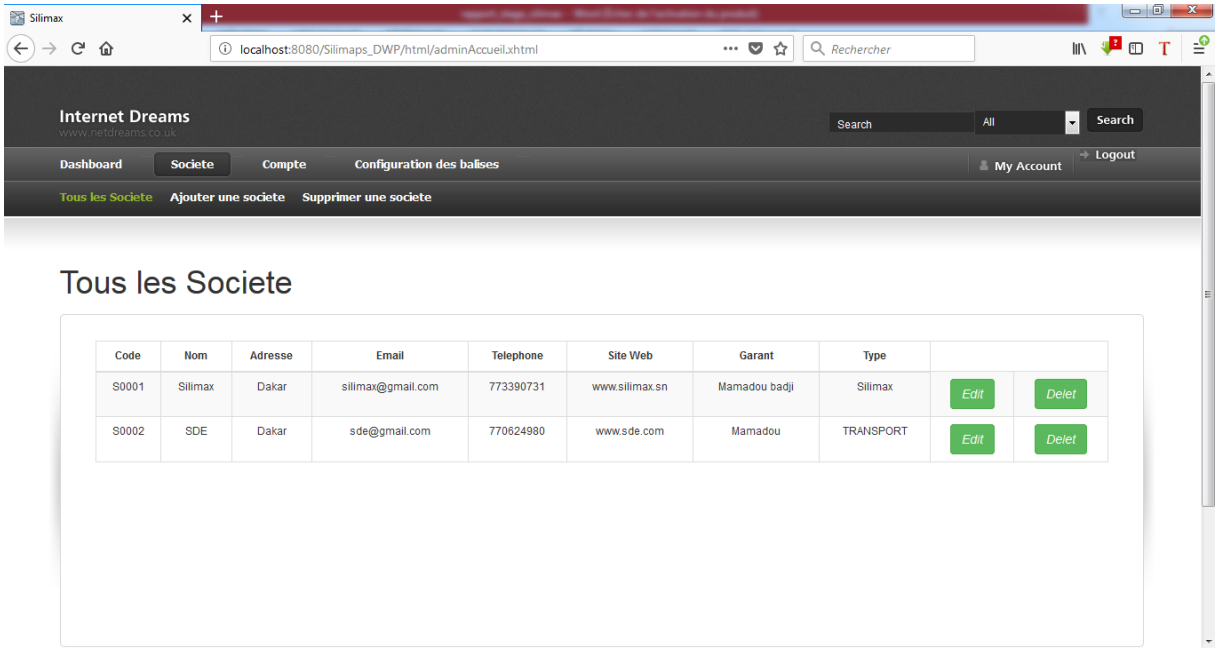

Figure 23: Liste des Sociétés

5.4.3. Formulaire d'ajout de Société

Cette interface nous permet de saisir les informations concernant une nouvelle société et de l'enregistrer dans la base de données.

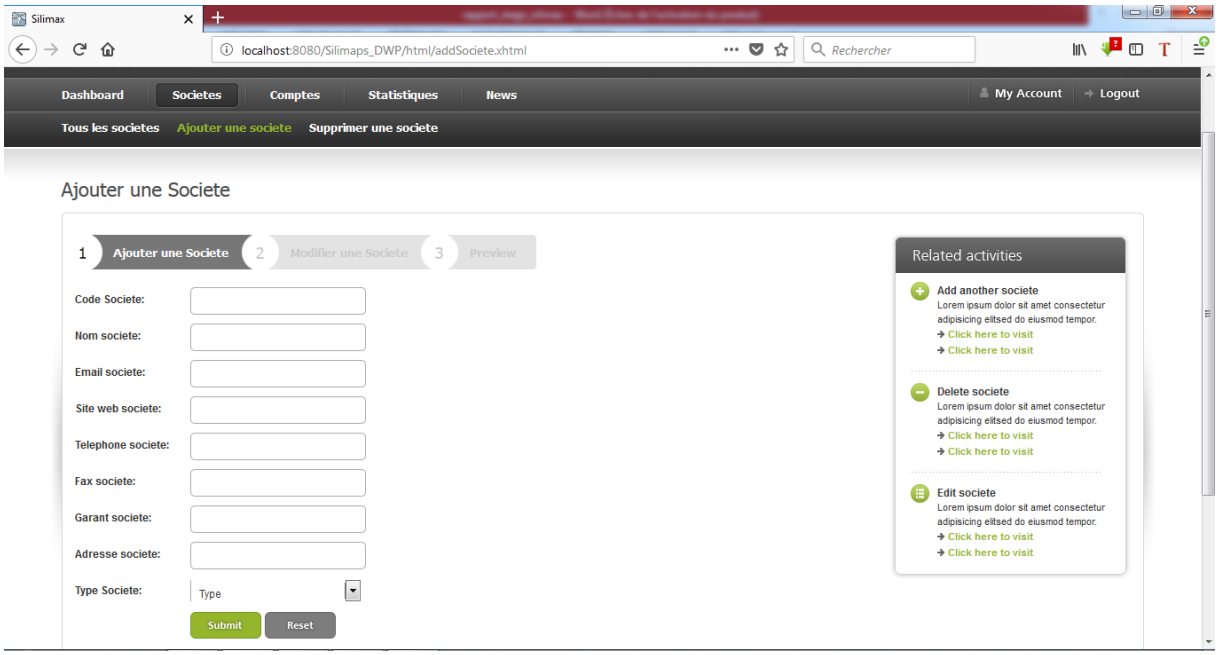

Figure 24: Formulaire d'ajout de Société

5.5. Web service la liste des comptes employés

Grâce à ce lien nous pouvons afficher la liste des comptes employés (http://localhost:8080/Silimaps_DWP/rest/utilisateur/checkUser/keba/keba) la figure 34 ci-dessous nous montre le résultat sur un navigateur Mozilla Firefox.

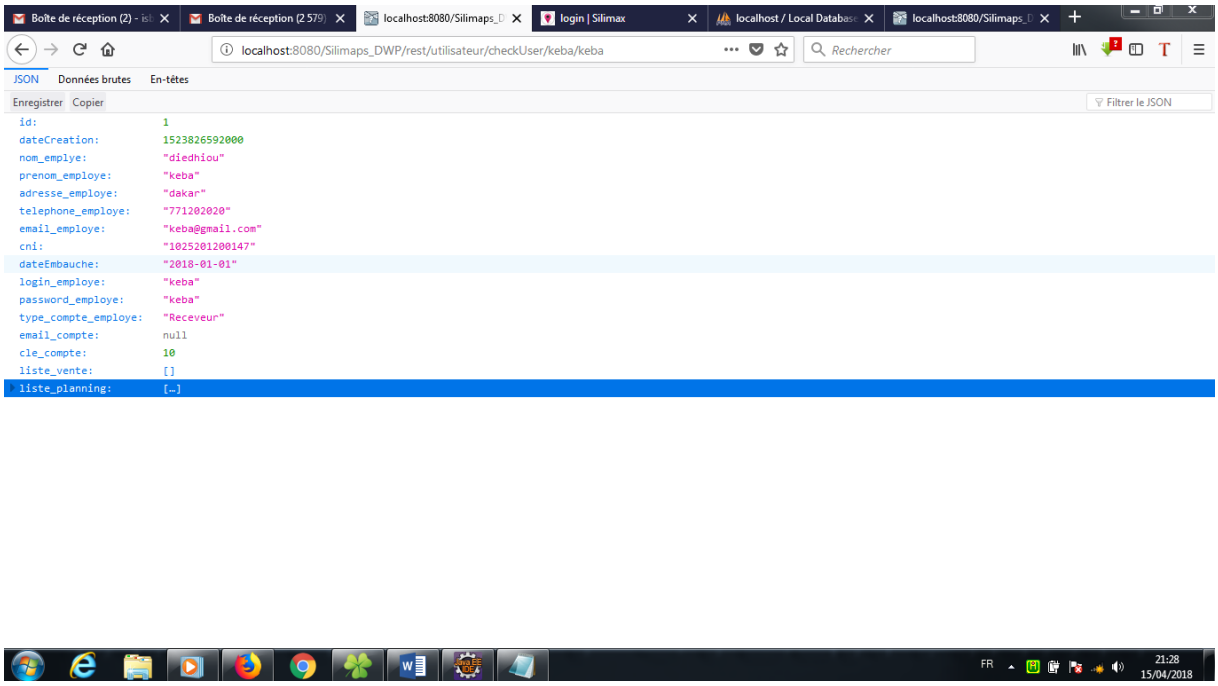

Figure 25:liste des comptes employés

5.6. Coordonnée d'un véhicule

Grâce à ce lien nous pouvons afficher l'ensemble des points occupé par le véhicule (http://localhost:8080/Silimaps_DWP/rest/balise/checkVehicule/ZG1771AB) la figure 35 ci-dessous nous montre le résultat sur un navigateur Mozilla Firefox.

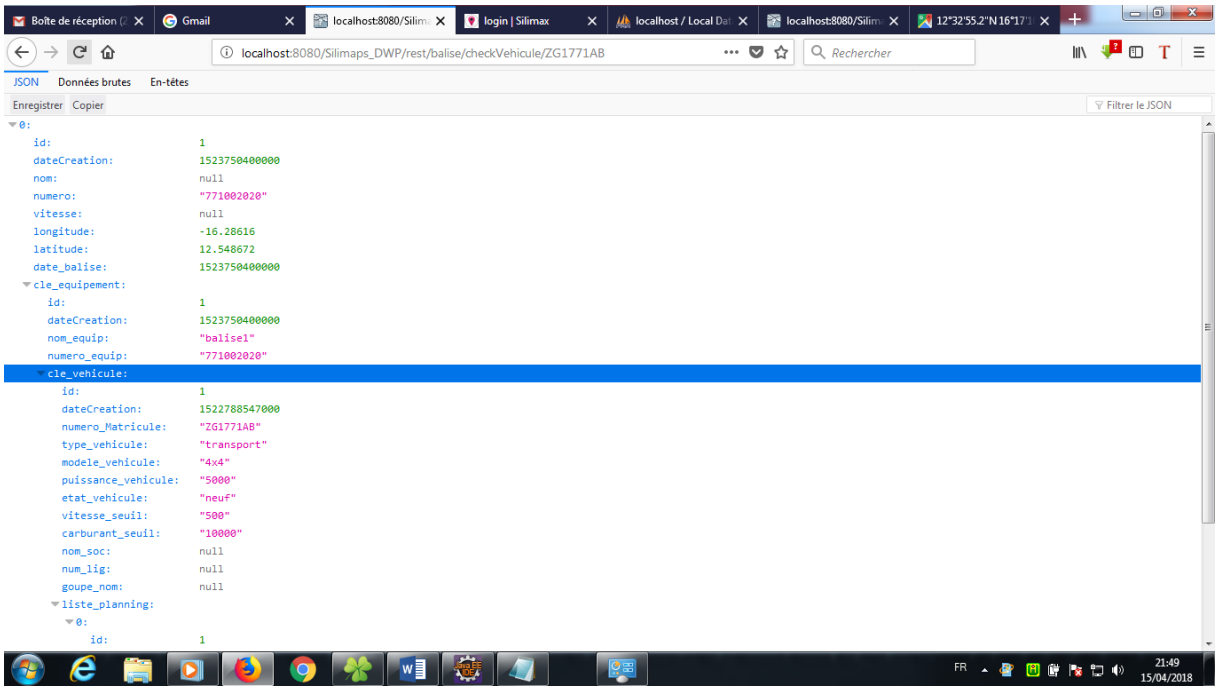

Figure 26: Coordonnée d'un véhicule

Dans ce chapitre, nous avons présenté notre environnement de développement, en spécifiant les différents outils de développement que nous avons utilisé, puis la structuration de l'application et enfin le travail que nous avons réalisé.

CONCLUSION

La réalisation de ce travail nous a été bénéfique à plus d'un titre. Ce fut pour nous une expérience des plus enrichissantes. L'objectif principal était de concevoir et de mettre en place un **système de suivi de flotte de véhicules** par Silimax.

Après le passage par les différentes phases de réalisation du projet, nous avons abouti à un système fonctionnel qui répond globalement aux critères imposé par le client.

Dans un premier temps, nous avons fait la présentation de la structure d'accueil, la présentation du sujet, de la problématique, et spécifié les objectifs à atteindre. En s'appuyant sur Scrum, nous avons ensuite identifié les différentes fonctionnalités que doit offrir notre application. Enfin, nous avons réalisé la conception et l'implémentation de notre système. Le prototype actuel permet de récupérer les coordonnées géographiques d'un véhicule et de le suivre sur une carte.

Toutefois, l'application que nous avons développée doit être intégrée avec un autre module chargé de visualiser des véhicules sur une carte, et de suivre leurs historiques. Ce module doit également permettre la vente de ticket, l'affectation de chauffeur et de receveur à un véhicule et gérer les comptes des employés.

En guise de perspectives, nous envisageons de développer des fonctionnalités permettant:

- la gestion des kilométrages : connaitre le nombre de kilomètre sur chaque tronçon ;
- la gestion de la consommation de carburant : connaitre la consommation de carburant sur chaque tronçon ;
- la recherche d'informations sur les lignes de transport urbain par les clients ;
- de connaitre des itinéraires, et de donner des informations précises sur les sections (délimitation de tarif).

BIBLIOGRAPHIE

[B1] Sujet : Conception et développement d'un bureau Virtuel « WebTop »

- Localisation = Université virtuelle de Tunis
- Auteur : Marwa KESRAOUI
- Type de Document : Mémoire

[B2] UML 2 par la pratique

- Auteurs : Pascal ROQUES
- Edition : Eyrolles

[B3] Sujet : Conception et la mise en place d'un système de gestion de pharmacie et d'IPM (Institution de Prévoyance Maladie) par l'entreprise IDYAL

- Localisation = ESP
- Auteur : M. Kailou Assoumane Abdoul-Nasser
- Type de Document : Mémoire
- Date de consultation: 01/2018

WEBOGRAPHIE

[W1] Two tracks umifed process: https://fr.scribd.com/doc/49697489/Processus-de-Developpement-Y-Processus-2TUP

- Date de consultation: 06/2017

[W2] les méthodes classiques : http://www.access-dev.com/access-dev/la-gestion-de-projet-methodes-classiques-vs-methodes-agiles/

- Date de consultation: 06/2017

[W3] UML https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/uml-c-est-quoi

- Date de consultation: 06/2017

[W4] MySQL: https://fr.wikipedia.org/wiki/MySQL

- Date de consultation: 06/2017

[W5] Fondateur Scrum : http://www.scrumguides.org/

➢ Date de consultation: 06/2017

[W6] Scrum : http://www.ingenosya.com/ingenosya/agilite