

RÉPUBLIQUE DU SÉNÉGAL  
Un Peuple – Un But – Une Foi



*MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR, DE LA RECHERCHE ET DE L'INNOVATION*



*UNIVERSITE AMADOU HAMPATE BA DE DAKAR*

\*\*\*\*\*

*FACULTE DES SCIENCES ET TECHNIQUES*

\*\*\*\*\*

*DEPARTEMENT DES SCIENCES ET TECHNOLOGIE DE L'INFORMATION DE LA  
COMMUNICATION (STIC)*

## **MÉMOIRE DE LICENCE PROFESSIONNELLE STIC**

**Option: Informatique**

**SUJET : CONCEPTION ET RÉALISATION D'UNE  
PLATEFORME ENTRE LES PROFESSIONNELS ET LES  
USAGERS**

Présenté et soutenu publiquement par :

**M. Cheikh Marr**

**Encadreur**

**Grade**

**M. DIOP**

**Ingénieur chef du système d'information ville de Dakar**

*Année Académique 2021/2022*

## **A LA MEMOIRE DE**

Tous ceux qui ont contribué à notre éducation, à notre formation et à notre réussite et qui malheureusement ne sont plus :

ADJA FATOU NDIAYE

Paix à leur âme et que Dieu les accueille au paradis.     Amine.

## **DEDICACE**

A mes chers parents en guise de reconnaissance et de gratitude pour leur amour, leur patience, leur soutien, leur encouragement et leur sacrifice.

Aucun mot, aucune dédicace ne pourrait exprimer notre respect, notre considération ainsi que l'expression de notre profond amour.

Puisse Dieu vous accorder santé, bonheur et longue vie afin que nous puissions un jour combler de joie vos vieux jours.

## **REMERCIEMENTS**

Après avoir rendu grâce à Dieu, prier au nom de son Prophète Mohamed paix et salut sur Lui, nous tenons à remercier tous ceux qui de près ou de loin ont participé à l'élaboration de ce mémoire. Nos sincères remerciements à :

- A mes chers parents qui m'ont soutenus et encouragés durant toute ma vie et durant tout mon cursus d'étude ;
- L'ensemble du corps professoral de l'Université **Amadou Hampaté Bâ** de **Dakar**, qui n'ont ménagé aucun effort pour la réussite de nos études et ont inculqué en nous des valeurs qui, certainement, faciliteront notre intégration dans la vie active ;
- Tous les membres du jury pour leur présence, leur lecture attentive de ce mémoire ainsi que pour les remarques qu'ils nous adresseront lors de cette soutenance afin d'améliorer le travail rendu ;
- Nous exprimons nos profonds remerciements à notre encadreur M. Lamine DIOP, qui a renforcé en nous le sens de la rigueur et le goût du travail bien fait. Ces remerciements ne sauraient se terminer sans une pensée à l'ensemble du corps professoral de l'université Amadou Hampaté Bâ de Dakar, particulièrement à M. Moustapha DER, M. Kara SAMB, M. Aliou BOLY, M. Baye NIASS et M. Lamine DIOP qui ont su développer en nous la détermination, la rigueur et le sens du professionnalisme, des valeurs qui, certainement, nous serons d'un grand aide dans notre période active ;
- Je souhaite remercier également mes amis particulièrement à Fatoumata Bintou COLY, Ekia Nkonga Josué Friedrich, Djibril MBAO, Mouhadoumadou Moustapha Sy DIEME, El Hadji Badou NGOM, Adama DANFA, Ndack SECK, Ndeye Coura GUISSSE et mes amis de classes et aussi ceux de STIC1 et STIC2 et les différentes personnes que j'ai pu croiser durant ces trois dernières années et qui ont su par leurs questions et leurs conseils me permettre de toujours pousser un peu plus notre raisonnement ;
- Toute la famille FAYE et MARR pour leur soutien inconditionnel Nous souhaitons remercier également les différentes personnes que nous avons pu croiser durant ces deux dernières années et qui ont su par leurs questions et leurs conseils nous permettre de toujours pousser un peu plus notre raisonnement ;

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Que tous ceux dont nous n'avons pas pu citer les noms trouvent ici l'expression de notre sincère reconnaissance à leurs égards.

### AVANT PROPOS

L'Université Amadou Hampaté Bâ de Dakar (UAHB) est un modèle d'université au Sénégal.

Autorisée par l'Etat du Sénégal (0040/AG/ME/DES).

Forme des professionnels dans plusieurs domaines, notamment l'informatique, pour des niveaux différents.

Ces formations sont sanctionnées par des diplômes, à savoir : Licence Professionnelle et Master.

L'obtention du diplôme de licence 3 en Informatique à UAHB est suivie d'un mémoire de fin de cycle que tout étudiant (en cycle licence) devra préparer et soutenir pour espérer bénéficier de ce noble titre. C'est dans ce cadre, que nous avons fourni ce document, comme mémoire, portant sur : ***Conception et réalisation d'une plateforme de mise en relation entre les professionnels et les usagers.***

La géolocalisation est un procédé permettant de positionner un objet, un véhicule ou une personne sur un plan ou une carte à l'aide de coordonnées géographiques. Il est difficile pour les personnes voyageant dans des régions inconnues de se repérer pour trouver des services adéquates pour la gestion ou réparation de leur véhicule. Leur préoccupation est de trouver des professionnels certifiés dans un rayon proche de lui et en un temps record. En outre, bon nombre de ces fournisseurs de services ont du mal à trouver des clients dans leurs communautés. Cela a ralenti le développement du secteur des transports, ralentissant ainsi le développement de l'économie.

Cher(s) membre(s) du jury, ce document n'est guère exempt de reproches et d'imperfections, peut-être : car étant l'œuvre d'un être humain d'abord, et de celui d'un chercheur scientifique en manque d'expérience. C'est pourquoi, nous implorons votre indulgence quant à l'évaluation de notre travail.

## **SOMMAIRE**

A LA MEMOIRE DE .....	i
DEDICACE .....	ii
REMERCIEMENTS .....	iii
AVANT PROPOS.....	iv
SOMMAIRE .....	v
GLOSSAIRE.....	vii
LISTE DES FIGURES.....	viii
LISTE DES TABLEAUX.....	x
RESUME .....	xi
ABSTRACT.....	xi
Chapitre 1 : INTRODUCTION GENERALE.....	1
1.1. Contexte.....	2
1.2. Problématique.....	3
1.3. Objectifs .....	4
1.4. Intérêt du sujet .....	4
Chapitre 2 : TRAVAUX EXISTANTS.....	6
2.1. Présentation de la solution Vroomly .....	6
2.2. Présentation de la solution Proximeca.....	7
2.3. Présentation de la solution IDGARAGES.....	8
2.4. Présentation de la solution Wutilma.....	9
2.5. Critique de l'existant .....	10
2.6. Proposition de notre solution.....	10
Chapitre 3 : ANALYSE ET CONCEPTION .....	11
3.1. Présentation et choix de la méthode d'analyse.....	11
3.2. Diagramme de contexte.....	14
3.3. Diagramme de cas d'utilisation.....	15
3.4. Diagramme de séquence.....	24
3.5. Diagramme de classe.....	32

## **Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers**

3.6. Diagramme d'activité.....	36
3.7. Diagramme de déploiement.....	38
3.8. Méthodologie.....	40
Chapitre 4 : REALISATION .....	45
4.1. Architecture et Environnement.....	45
4.2. SGBD .....	52
4.3. Outils .....	54
4.5. Sécurité.....	57
4.6. Présentation Solution.....	59
CONCLUSION ET PERSPECTIVES .....	65
TABLE DES MATIERES .....	67
BIBLIOGRAPHIE .....	70
WEBOGRAPHIE.....	71

## **GLOSSAIRE**

<b>ANSD</b>	Agence Nationale de Statistique et de la Démographie
<b>ERP</b>	Enterprise Ressource Planning
<b>GPS</b>	Global Positioning System
<b>IHM</b>	Interface Homme-Machine
<b>UML</b>	Unified Modeling Language
<b>MCC</b>	Modèle Conceptuel de Communication
<b>MCD</b>	Modèle Conceptuel de Données
<b>MCT</b>	Modèle Conceptuel des Traitements
<b>MOT</b>	Modèle Organisationnel des Traitements
<b>OMG</b>	Object Management Group
<b>OOSE</b>	Object Oriented Software Engineering
<b>OSM</b>	OpenStreetMap
<b>POO</b>	Programmation Orientée Objet



## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

<b>SGBD</b>	Système de Gestion de Base de Données
<b>SGBDR</b>	Système de Gestion de Base de Données Relationnel
<b>SGBDRO</b>	Système de Gestion de Base de Données Relationnel Objet

## LISTE DES FIGURES

Figure 1: Logo de Vroomly .....	6
Figure 2 : Logo de Proximeca .....	7
Figure 3: Logo de IDGARAGES .....	8
Figure 4: Diagramme de contexte du système .....	15
Figure 5 : Exemple de base d'un cas d'utilisation .....	16
Figure 6 : Diagramme de cas d'utilisation « gestion des utilisateurs » .....	18
Figure 7 : Diagramme de cas d'utilisation gestion des demandes de prestation .....	21
Figure 8 : Représentation du symbole d'objet .....	26
Figure 9 : Représentation d'une boîte d'activation .....	26
Figure 10 : Représentation d'un symbole d'acteur .....	26
Figure 11 : Représentation de symbole paquetage .....	27
Figure 12 : Représentation d'une ligne de vie .....	27
Figure 13 : Représentation de symbole de boucle optionnelle .....	27
Figure 14 : Représentation de symbole d'alternative .....	28
Figure 15 : Exemple de base d'un diagramme de séquence .....	28
Figure 16 : Diagramme de séquence authentification .....	29
Figure 17 : Diagramme de séquence inscription .....	30
Figure 18: Diagramme de séquence demande de prestation .....	31
Figure 19: Diagramme de séquence traiter demande usager .....	32
Figure 20 : Exemple de base diagramme de classe .....	34
Figure 21: Diagramme des classes du système .....	35
Figure 22 : Représentation des symboles d'un diagramme d'activité .....	36
Figure 23 : Diagramme d'activité demande d'enrôlement .....	37
Figure 24 : Diagramme d'activité géolocalisation professionnel véhicule .....	38
Figure 25: Diagramme de déploiement du système .....	39
Figure 26 : Illustration des rôles dans la méthodologie Scrum .....	42
Figure 27 : Graphique d'avancement de la méthodologie Scrum .....	44
Figure 28 : Architecture 2 tiers .....	46
Figure 29 : Architecture 3 tiers .....	47

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Figure 30 : Architecture n-tiers .....	48
Figure 32 : SQL Server .....	52
Figure 33 : Oracle Database .....	52
Figure 34 : MySQL .....	53
Figure 35 : PostgreSQL .....	53
Figure 36: Logo astah uml.....	57
Figure 38: Capture d'écran page inscription & recherche d'un service .....	59
Figure 39: Capture d'écran page recherche garages .....	59
Figure 40: capture d'écran page des demandes .....	60
Figure 41: Capture d'écran page accueil.....	61
Figure 43: capture d'écran page de recherche de garages.....	62
Figure 45: Capture d'écran page de connexion .....	62
Figure 46: Capture d'écran page mot de passe oublié .....	63
Figure 48: Capture géolocalisation.....	61
Figure 49: Capture d'écran liste des garages .....	64

## **LISTE DES TABLEAUX**

Tableau 1 : Tableau comparatif Merise et UML .....	14
Tableau 2 : Tableau descriptif du cas d'utilisation Gestion des utilisateurs .....	20
Tableau 3 : Tableau descriptif du cas d'utilisation Gestion des prestations .....	24
Tableau 4 : Symbole, libellé et description des messages d'un diagramme de séquence .....	26
Tableau 5: Fiche descriptive des classes .....	35

## **RESUME**

L'augmentation de la population et des besoins de transports ont engendré la hausse des demandes liés à l'automobile. De ce fait, certains peinent à accéder à ces services tandis que d'autres veulent rendre le service plus accessible et autonome. Des personnes en voyage rencontrent d'énormes difficultés pour trouver un mécanicien qualifié pour les aider.

Le but de notre étude est de créer une plateforme web et mobile permettant à toute personne désirant des services liés à l'automobile d'en disposer dans un rayon proche de sa position. Cette plateforme facilitera la recherche du professionnel du véhicule et l'assistance des usagers. Aussi, elle permettra de regrouper tous les professionnels du véhicule et de les aider à trouver des clients même dans les zones les plus reculées. Ainsi, leurs données pourraient être mieux gérées dans le futur pour l'amélioration de ce secteur.

Le client, par rapport à sa position pourra rechercher les garages mécaniques qui sont plus proches de lui. Pour un service bien spécifique, il fera une demande adressée à un garage spécifique. De son côté, le professionnel pourra gérer l'ensemble de ses garages et ses employés mais aussi faire une offre par rapport à une demande d'un client pour ensuite passer à la réalisation du service.

### **Mots clés :**

Garage mécanique, géolocalisation, automobile, voyage, autonomisation, service, offre.

## **ABSTRACT**

Due to the growing number of vehicles caused by the rapidly growing population, demand for automotive-related services continues to grow. As a result, some are struggling to access these services while others want to make the service more accessible and self-sufficient. Some drivers, while on the road, find it very difficult to meet an engineer who can instantly help them.

Thus, the aim of our study is to create a web and mobile platform allowing anyone seeking those services, to have them at a radius not far from their position. The platform will ease the search for an engineer and the assistance of the users. Also, it will gather all the professionals

## **Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers**

of the domain and help them find customers even in the most remote areas. Subsequently, their data could be better managed in the future for the improvement of this sector.

The customer, in relation to his position will be able to search for mechanical garages that are closer to him. For a specific service, he will make a request. For his part, the professional will be able to manage all his garages and his employees but also issue an offer in relation to a request from a customer and then move to the realization of the service.

### **Key words:**

Mechanical garage, geolocation, car, travel, empowerment, service, offer.

## **Chapitre 1 : INTRODUCTION GENERALE**

De nombreuses activités de la vie courante ont été simplifiées grâce à l'informatique. Actuellement, les individus peuvent facilement traiter des informations en se servant des logiciels et des réseaux informatiques. Compte tenu de son évolution, ce système regroupe la majorité des grandes entreprises quel que soit le secteur d'activité. Dans ce cadre, l'informatisation des processus est un facteur déterminant qui apporte sa contribution pour l'amélioration de la vie.

Une personne en voyage dans une région qui lui est inconnue rencontre d'énormes difficultés pour trouver des services liés à l'automobile (réparation, lavage, vulcanisation etc.). Son principal souci est de retrouver dans un rayon non éloigné et en un temps record. De plus, bon nombre des offreurs de service peinent à trouver des clients dans leurs localités. Ceci freine le développement du secteur de transport, donc de l'économie.

Conscients de la nécessité de l'utilisation des nouvelles technologies de l'information et de la communication dans le domaine de l'automobile, nous avons jugé nécessaire de cartographier et d'enrôler les professionnels du véhicule afin d'aider les particuliers à les localiser facilement et d'avoir une meilleure offre qualité/prix pour un service demandé.

L'objectif général de notre étude est de concevoir et d'implémenter une plateforme avec accès mobile et web pour relier spécialistes de véhicule et demandeurs de service à tout moment. Les objectifs spécifiques sont : **permettre à un particulier de localiser un professionnel du véhicule et vice versa ; aider ces professionnels à se formaliser (registre de commerce, ninéa) ; trouver pour ces particuliers la meilleure offre pour un service spécifique ; enrôler l'ensemble des professionnels de véhicule.**

Notre motivation est d'une part d'améliorer la qualité de service de ce secteur. D'autre part de pouvoir développer et implémenter une plateforme de géolocalisation permettant de relier les différentes entités. L'utilisation de l'architecture orientée services reste aussi une grande source de motivation pour nous.

## **Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers**

Ainsi, pour la mise en œuvre de cette solution les hypothèses suivantes sont à considérer : création d'une plateforme web pour l'enrôlement des professionnels et la gestion des prestations, ainsi que le développement d'une application mobile pour leur géolocalisation.

Dans la réalisation de ce document nous allons utiliser les techniques d'investigation suivantes : la recherche documentaire, l'observation et les questionnaires.

Ainsi, dans l'optique d'atteindre les objectifs que nous nous sommes fixés à travers ce présent mémoire : nous aborderons notre étude d'abord en commençant par les travaux existants, ensuite nous parlerons de la conception de la solution proposée puis terminer par son implémentation.

### **1.1. Contexte**

Au Sénégal, le parc automobile est constitué de l'ensemble des moyens de déplacement motorisé. Selon l'ANSD, dans son rapport nommé Situation économique et social du Sénégal 2017-2018, il est évalué à 820 289 contre 766 737, soit une hausse de 7%. Toutefois, il faut noter l'accroissement de 51,2% en 2017. Avec l'extension du parc automobile, l'allongement de la durée de vie des véhicules etc. le volume d'activité du secteur de la réparation et de l'entretien ne cesse d'accroître (i). Cette expansion se constate dans toutes les gammes de véhicules et elle a aussi diversifié les marques de voitures partout dans le monde.

Tout ceci augmente le nombre de poubelles ambulant (des vieux véhicules) dans nos rues et fait vieillir, par la même occasion, la moyenne d'âge du parc automobile. Ce qui contribue à la pollution (augmente le risque de maladies), les accidents, les embouteillages et les risques de pannes dans le pays.

Tandis que le parc automobile s'enrichit de jour en jour de nouveaux véhicules électromécaniques, et compte tenu de la vitesse folle du développement de la technologie automobile dans le monde, les mécaniciens travaillant au Sénégal sont débordés et impuissants face à des pannes mécaniques qui sont actuellement indissociables de l'électronique et même de l'informatique. Les jeunes mécaniciens en panne de compétences lancent un cri d'alarme, sollicitant le gouvernement pour une formation de mise à niveau en entretien, diagnostic et réparation automobile afin qu'ils soient en mesure de prendre en charge tous les problèmes liés à l'entretien des voitures.

## **Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers**

L'avancée technologique pointue traduit une certaine aisance sociale chez les propriétaires qui acquièrent des voitures venues d'Amérique, d'Europe ou d'Asie. Toutefois, elle accentue le manque de compétences de ces mécaniciens en particulier, des professionnels de véhicules en général.

La presque totalité des véhicules circulant au Sénégal fonctionnent au diesel et à l'essence causant ainsi des émissions de gaz à effet de serre et participant à la dégradation de la couche d'ozone et au réchauffement climatique. Le mauvais état des véhicules et leur extrême vieillesse accentue ces émissions.

Avec la recrudescence des accidents de la route, qui coûte au Sénégal près de 2% de son Produit Intérieur Brut par an, la problématique de la sécurité routière est devenue un enjeu national d'une extrême priorité. La production des titres de transport, telle qu'elle est prise en charge aujourd'hui, pose d'importants problèmes et ne répond pas à l'enjeu de réduction de l'insécurité routière. Parmi ces problèmes peuvent être mentionnés les fraudes liées à la non sécurisation des supports utilisés pour la production des titres de transport et l'absence d'interconnexion des différents acteurs entraînant d'importantes pertes de recettes pour l'Etat, l'absence d'un dispositif de centralisation et de vérification de l'intégrité des données et l'impossibilité de produire des statistiques fiables et à temps réel.

De plus, pour les personnes voyageant dans des régions inconnues, il est difficile de trouver des services liés à l'automobile. Ainsi la recherche d'une personne qualifiée pour gérer certains problèmes automobiles devient extrêmement difficile.

### **1.2. Problématique**

Compte tenu de la situation citée ci-haut, nous sommes parvenus à nous poser les questions suivantes :

- Comment aider un particulier à localiser un professionnel de véhicule ?
- Comment aider un professionnel des véhicules à retrouver des clients ?
- Aider ces professionnels à se formaliser (registre de commerce, ninéa) ?
- Trouver pour ces particuliers la meilleure offre c'est-à-dire le rapport qualité/prix ?



## **Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers**

- Comment enrôler l'ensemble des professionnels de véhicule pour avoir des statistiques pour l'aide à la décision ?
- Donner la possibilité aux particuliers de pouvoir marchander sur le prix d'une offre.
- Permettre le paiement en ligne pour les clients.

### **1.3. Objectifs**

Conformément à la méthodologie ; les objectifs seront de deux natures à savoir l'objectif général et les objectifs spécifiques.

#### **1.3.1. Objectif général**

De manière générale, nous espérons développer et mettre en œuvre une plate-forme avec accès mobile et Web pour connecter les experts en véhicules et les demandeurs de services.

#### **1.3.2. Objectifs spécifiques**

De façon spécifique, nos objectifs sont :

- Permettre à un particulier de localiser un professionnel du véhicule n'importe où au Sénégal ;
- Permettre aux professionnels de retrouver des clients et leurs donner une visibilité ;
- Aider les professionnels de véhicule à se formaliser (registre de commerce, ninéa) ;
- Simplifier la collecte des pièces justificatives pour renforcer la qualité des services ;
- Trouver pour ces particuliers la meilleure offre pour un service spécifique ;
- Enrôler l'ensemble des professionnels de véhicule.
- Permettre aux usagers de choisir la meilleure offre qualité/prix
- Intégrer le paiement en ligne pour les clients.

### **1.4. Intérêt du sujet**

## **Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers**

Notre choix pour ce sujet a été guidé par la résolution d'un besoin pressant. Mais aussi, de mettre en place une plateforme pouvant faciliter la géolocalisation afin de mettre en relation les usagers et les professionnels de véhicule ainsi que la gestion des demandes de prestation. De plus, avec l'utilisation de l'architecture orientée service, ce travail nous permettra de concilier la théorie et la pratique acquises au cours de notre formation académique.

## **Chapitre 2 : TRAVAUX EXISTANTS**

Cette partie est indispensable pour une meilleure compréhension du système actuel. Elle nous permet de clarifier des champs d'investigation. Il existe des solutions représentant des hypothèses, nous allons faire l'étude de ces derniers pour orienter notre choix sur un achat ou une implémentation selon ces spécifications. Ainsi, nous allons étudier les fonctionnalités de chaque application en mettant en exergue leurs avantages et inconvénients.

### **2.1. Présentation de la solution Vroomly**

Vroomly est une plateforme web permettant d'aider les usagers à comparer les meilleurs centres auto selon le prix, la distance et les avis des autres clients. Ainsi, les clients sont assurés de payer le meilleur prix pour toutes les interventions à réaliser sur leur véhicule.



Figure 1:Logo de Vroomly

#### **2.1.1. Fonctionnalités**

Vroomly dispose des fonctionnalités suivantes :

- Visibilité en ligne ;
- Devis instantané ;
- Réservation en ligne

#### **2.1.2. Avantages**

Il dispose de plusieurs avantages :

- Trouver rapidement les prestations d'un garage auto en étant rassuré sur le prix avant même de se déplacer ;
- Des avis clients sont disponibles pour chacun des Garages ;
- Possibilité de sélectionner l'horaire qui vous convient et prendre rendez-vous directement en ligne.

### **2.1.3. Inconvénients**

- Les prix sont fixes et ne varient pas ;
- Solution non disponible au Sénégal

## **2.2. Présentation de la solution Proximeca**

Proximeca est une plateforme web et mobile qui regroupe des réparateurs et mécaniciens automobiles toutes marques. Proximeca compte plus d'une centaine de garages à proximité de chez vous, en zone périurbaine pour la plupart. Les réparateurs sont des généralistes, capables d'entretenir les véhicules, quelle que soit sa marque et son modèle, mais ils n'en restent pas moins de véritables techniciens qualifiés et compétents pour entretenir votre véhicule pendant et après la période de garantie constructeur.



Figure 2 : Logo de Proximeca

### **2.2.1. Fonctionnalités**

Avec l'application **My Proximeca**, vous aurez accès à de nombreux services et vous pourrez :

- Localiser le garage le plus proche de vous ;
- Rechercher les garages Proximeca sur une carte interactive ;
- Ajouter un garage favori pour accéder à ses informations et coordonnées pour contacter votre réparateur facilement ;
- Obtenir votre plan d'entretien et être informé en temps réel des prochaines interventions à effectuer sur votre véhicule ;
- Sauvegarder toutes vos données de votre véhicule ;
- Bénéficier des offres promotionnelles en cours.

### **2.2.2. Avantages**

Avec Proximeca, vous pouvez avoir des avantages suivants :

- Rechercher les garages Proximeca sur une carte interactive ;
- Trouver le garage le plus proche ;
- Ajouter un garage en favori ;

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

- Bénéficier d'offres promotionnelles ;
- Sauvegarder toutes ses données véhicules ;
- Obtenir le plan d'entretien et être informé en temps réel des prochaines interventions à effectuer ;
- Estimer le budget de réparation et entretien.

### 2.2.3. Inconvénients

- Solution non disponible au Sénégal ;
- La plupart des utilisateurs ne sont pas conscients que de telles informations sont partagées, en particulier auprès d'un public aussi large, voir du monde entier ;
- Manque de control sur les informations des utilisateurs, une personne malveillante peut tout savoir de vous : votre adresse, votre lieu de travail, vos horaires...

## 2.3. Présentation de la solution IDGARAGES

IDGARAGES est une plateforme de comparateur de garages en ligne depuis 2012, il propose des milliers de garagistes selon différents critères : le meilleur prix, les meilleurs avis client, la proximité pour un entretien à petit prix mais toujours de qualité.



Figure 3: Logo de IDGARAGES

### 2.3.1. Fonctionnalités

En utilisant l'application iDGARAGES, vous aurez accès à de nombreux services tel que :

- Bénéficier d'un service de réservation d'entretien et de réparation au meilleur tarif ;
- Trouver le bon garage à côté de chez vous, proche de votre lieu de travail ou idéalement situé sur votre lieu de vacances ;
- Réaliser un devis pas cher en temps réel pour prendre un rdv pour vos interventions et la réparation des pannes de votre voiture

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

### 2.3.2. Avantages

- Devis en temps réel et rendez-vous immédiats ;
- Plus de 35 types d'interventions (révision et vidange, courroie de distribution, amortisseurs, échappement, batterie, freins, parallélisme...)
- Retrouver l'ensemble des interventions disponibles sur le site avec des remises pouvant atteindre jusqu'à -40% pour un budget auto maîtrisé ;
- Créer un lien entre garagistes et automobilistes

### 2.3.3. Inconvénients

- Solution non disponible au Sénégal ;
- Manque de professionnalisme et de fiabilité de certains professionnels référencés sur le site.

## 2.4. Présentation de la solution Wutilma

Wutilma offre à ses clients un accès facile à plusieurs services à travers une plateforme en ligne automatisée, intuitive et personnalisée avec la possibilité de donner leur avis. Il met à leur disposition des professionnels qualifiés et vérifiés



### 2.4.1. Fonctionnalités

Avec Wutilma, vous aurez accès à de nombreux services et vous pourrez :

- Bénéficier d'un professionnel fiable, compatissant et abordable sur votre calendrier, selon vos besoins ;
- De services fournis en fonction du lieu où vous vous trouvez.

### 2.4.2. Avantages

- Regrouper l'ensemble des services qui vous intéressent sur un forfait personnalisé ;

## **Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers**

- Les tâches réservées et payées sont effectuées par des professionnels expérimentés et approuvés qui sont très appréciés par des clients.

### **2.4.3. Inconvénients**

- La plateforme n'est pas spécifique aux services liés à l'automobile ;
- L'utilisateur n'a pas la possibilité de choisir le garage qu'il souhaite.

## **2.5. Critique de l'existant**

Après analyse, nous avons constaté que les plateformes tels que Vroomly et Idgarage sont indisponibles au Sénégal. Nous notons aussi un coût élevé. De plus, les services ne sont pas spécifiques à l'automobile sur certaines plateformes.

## **2.6. Proposition de notre solution**

Notre solution proposée est la mise en place d'une plateforme web et mobile pouvant aider les conducteurs dans leur recherche de services mécaniques à proximité. Nous proposons les fonctionnalités suivantes :

- Rechercher en fonction de sa position un garage mécanique pour satisfaire son besoin ;
- Faire une demande pour un service spécifique ;
- Enrôler les garages sur la plateforme ;
- Gérer ses garages ainsi que ses employés ;
- Proposer une offre par rapport à une demande ;
- Accomplir un service demandé ;
- Avoir un avis sur les services proposés par le garage à l'aide des commentaires des autres utilisateurs de la plateforme ;
- Effectuer un paiement.

## **Chapitre 3 : ANALYSE ET CONCEPTION**

### **3.1. Présentation et choix de la méthode d'analyse**

#### **3.1.1. Présentation**

La conception est une étape fondamentale dans le cycle de vie d'une application informatique. En effet, c'est d'elle que dépendent la qualité et la cohérence du produit réalisé au développement. Des méthodes de génie logiciel ont alors été développées afin de guider le concepteur dans sa tâche. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse.

##### **3.1.1.1. Merise**

La méthode Merise d'analyse et de conception propose une démarche articulée simultanément selon 3 axes pour hiérarchiser les préoccupations et les questions auxquelles répondre lors de la conduite d'un projet :

- Cycle de vie : phases de conception, de réalisation, de maintenance puis nouveau cycle de projet.
- Cycle de décision : des grands choix (GO-NO GO : Étude préalable), la définition du projet (étude détaillée) jusqu'aux petites décisions des détails de la réalisation et de la mise en œuvre du système d'information. Chaque étape est documentée et marquée par une prise de décision.
- Cycle d'abstraction : niveaux conceptuels, d'organisation, logique et physique/opérationnel (du plus abstrait au plus concret) L'objectif du cycle d'abstraction est de prendre d'abord les grandes décisions métier, pour les principales activités (Conceptuel) sans rentrer dans le détail de questions d'ordre de l'organisation ou technique.

La méthode Merise, très analytique (attention méthode systémique), distingue nettement les données et les traitements, même si les interactions entre les deux sont profondes et s'enrichissent mutuellement (validation des données par les traitements et réciproquement). Certains auteurs (Merise/méga, puis Merise/2) ont également apporté la notion



## **Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers**

complémentaire de communications, vues au sens des messages échangés. Aujourd'hui, avec les SGBD-R, l'objet, les notions de données et de traitements sont de plus en plus imbriquées.

### **3.1.1.2. UML**

UML (en anglais Unified Modeling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

UML est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jakobson, UML est à présent un standard défini par l'Objet Management Group (OMG). La dernière version diffusée par l'OMG est UML 2.3 depuis mai 2010.

UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet :

Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.

L'aspect formel de sa notation, limite les ambiguïtés et les incompréhensions.

Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel.

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

### 3.1.2. Etude comparative des méthodes Merise et UML

	Merise	UML
Niveau d'abstraction	Le cycle d'abstraction permet de sérier les niveaux de préoccupations lors de la description ou de l'analyse du système. Les trois niveaux retenus correspondent à des degrés de stabilité et d'invariance de moins en moins élevés (le niveau conceptuel, le niveau logique et le niveau physique).	L'approche UML propose différentes notions (cas d'utilisation, paquetage, classe, composant, nœud) et différents diagrammes pour modéliser le système aux différents niveaux d'abstraction.
Approche fonctionnelle	Merise propose une approche descendante où le système réel est décomposé en activités, elles-mêmes déclinées en fonctions. Les fonctions sont composées de règles de gestion, elles-mêmes regroupées en opérations. Ces règles de gestion au niveau conceptuel génèrent des modules décomposés en modules plus simples et ainsi de suite jusqu'à obtenir des modules élémentaires... Les limites d'une telle approche résident dans le fait que les modules sont difficilement extensibles et exploitables pour de nouveaux systèmes.	Les fonctions cèdent la place aux cas d'utilisation qui permettent de situer les besoins de l'utilisateur dans le contexte réel. A chaque scénario correspond des diagrammes d'interaction entre les objets du système et non pas un diagramme de fonction...
Dualité de données	Merise propose de considérer le système réel selon deux points de vue : un point de vue statique (les	L'approche objet associe les informations et les traitements. De cette façon, elle assure un certain

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

	données), un point de vue dynamique (les traitements). Il s'agit d'avoir une vision duale du système réel pour bénéficier de l'impression de relief qui en résulte, et donc consolider et valider le système final.	niveau de cohérence.
--	---	----------------------

Tableau 1 : Tableau comparatif Merise et UML

### 3.1.3. Choix de la méthode

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La plateforme gère des données plus complexes d'où le choix de la méthode d'UML qui est la mieux recommandée, vu qu'elle rend la modélisation plus simple à implémenter.

### 3.2. Diagramme de contexte

Le diagramme de contexte exprime l'environnement du système dans une situation donnée. Le diagramme de contexte permet donc de définir les frontières de l'étude et de préciser la phase du cycle de vie dans laquelle on situe l'étude.

Il recense les acteurs qui interagissent avec le système. Nous montrons ici un diagramme de contexte de notre plateforme.

Un acteur est une entité externe qui peut être une personne humaine ou robot interagissant avec le système. Une même personne peut être plusieurs acteurs c'est-à-dire affecter plusieurs rôles à une personne.

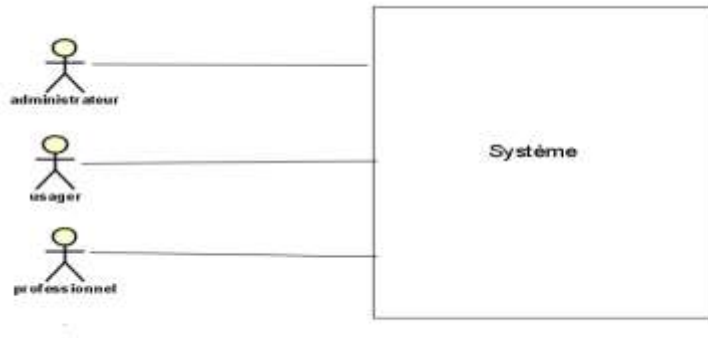


Figure 4: Diagramme de contexte du système

### **3.3. Diagramme de cas d'utilisation**

Les diagrammes des cas d'utilisations sont des vues qui décrivent les interactions entre les acteurs externes et le système lui-même. Il représente aussi un ensemble de séquences d'actions réalisées par le système produisant un résultat observable intéressant pour un acteur particulier.

Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.

#### **❖ Système :**

Un système représente une application dans le modèle UML. Il est identifié par un nom et regroupe un ensemble de cas d'utilisation qui correspond aux fonctionnalités offertes par l'application à son environnement.

#### **❖ Acteur :**

Un acteur représente le rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur peut consulter ou modifier directement l'état du système en émettant ou en recevant des messages successibles d'être porteur de donnée.

#### **❖ Association**

Une association est utilisée dans ce type de diagramme pour relier les acteurs et les cas d'utilisation par une relation qui signifie simplement « participe à ».

#### **❖ Inclusion**

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Une relation « include » est utilisée pour indiquer que le cas d'utilisation source (départ de la flèche) contient TOUJOURS le cas d'utilisation inclus. Pour représenter cette relation, on dessine donc une flèche en pointillé partant du cas d'utilisation principal vers le cas d'utilisation interne. Puis, on indique le stéréotype « include » sur la flèche

### ❖ Extension

Une relation « extend » est utilisée pour indiquer que le cas d'utilisation source (à l'origine de la flèche) n'est pas toujours nécessaire au cas d'utilisation principale, mais qu'il peut l'être dans certaines situations. L'ajout de cette relation se fait en dessinant une flèche en pointillé partant du cas d'utilisation interne vers le cas d'utilisation principal. Puis, on indique le stéréotype « extend » sur la flèche.

### ❖ Généralisation

Les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des relations spécifiques supplémentaires avec d'autres acteurs ou cas d'utilisation.

#### 3.3.1. Formalisme

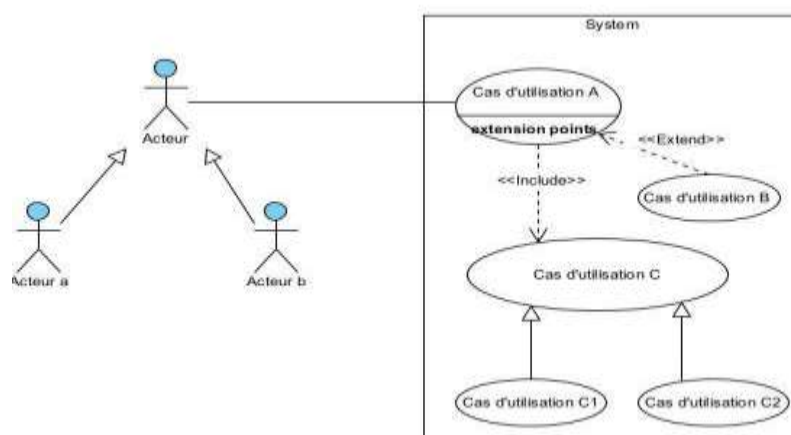


Figure 5 : Exemple de base d'un cas d'utilisation

#### 3.3.2. Description détaillée des cas d'utilisations

Cette partie est composée de deux éléments :

- Sommaire d'identification (titre, but, résumé, acteur).

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

- Description de l'enchaînement (précondition, post-condition, scenario nominale, scenario alternative).

**Titre :** Cas d'utilisation concerné.

**But :** l'objectif de ce cas d'utilisation dans le système.

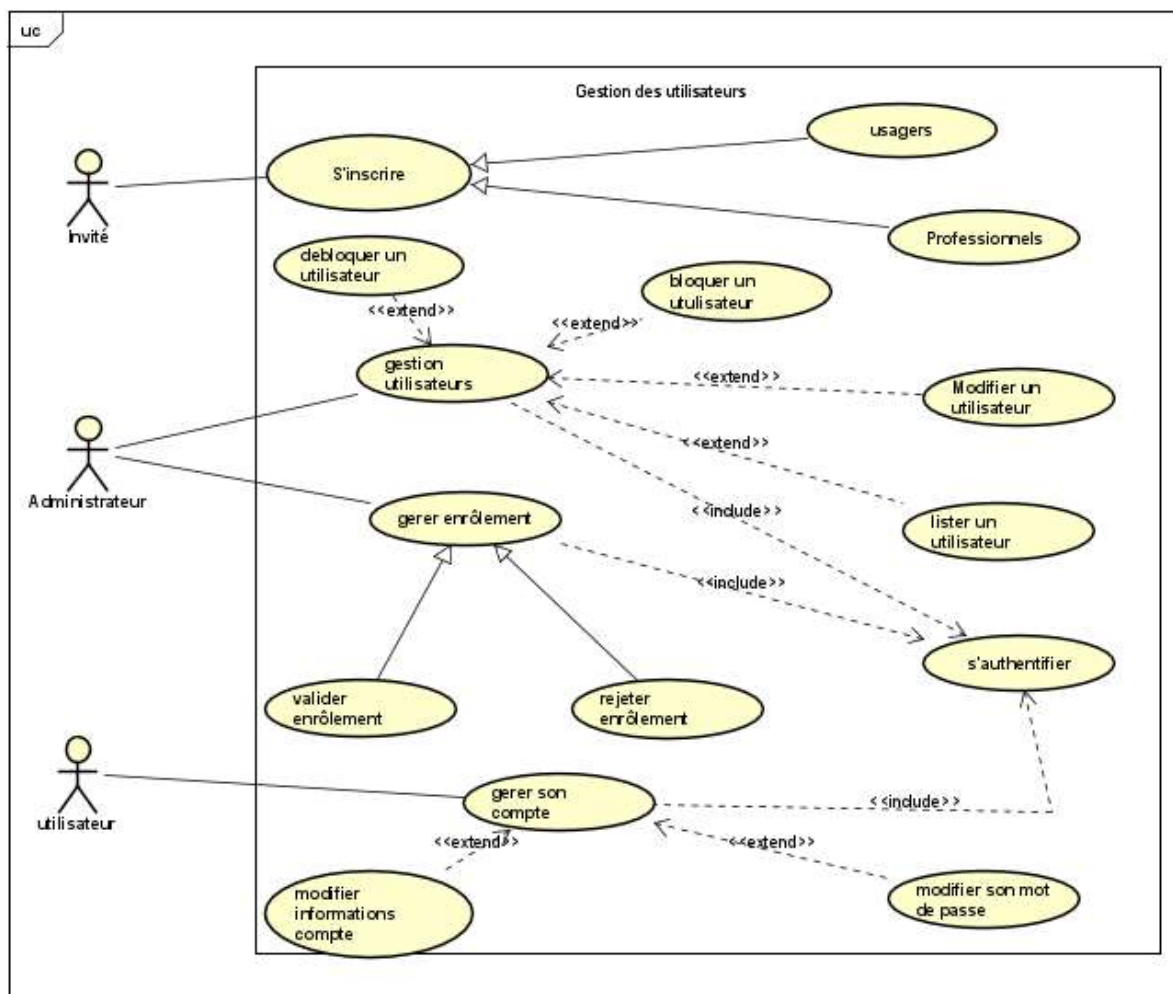
**Résumé :** c'est le résumé du contenu textuel

**Précondition :** Ce sont les conditions nécessaires pour déclencher les enchainements.

**Post-condition :** représente l'événement futur.

**Scenario nominale :** représente les événements produits par l'acteur et le système de la façon sans échec (sans erreur).

### 3.3.3. Diagramme de cas d'utilisation « Gestion des utilisateurs »



## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Figure 6 : Diagramme de cas d'utilisation « gestion des utilisateurs »

<b>SOMMAIRE D'IDENTIFICATION</b>	
<b>Titre :</b>	Gestion des utilisateurs
<b>But :</b>	<ul style="list-style-type: none"> <li>➤ Permettre à un invité de s'inscrire en tant que professionnel de véhicule ou un usager</li> <li>➤ Faciliter la gestion des utilisateurs (bloquer, débloquer...)</li> <li>➤ Gérer les enrôlements</li> <li>➤ Gérer les comptes</li> </ul>
<b>Résumé :</b>	Faciliter la gestion des inscriptions sur la plateforme, de même que la gestion des comptes des différents utilisateurs mais aussi permettre à l'administrateur du système d'enrôler les professionnels de véhicule
<b>Acteur :</b>	Utilisateur, administrateur, invité
<b>DESCRIPTION DES ENCHAINEMENTS</b>	
<i>Pré conditions</i>	<i>Post conditions</i>
Inscription	<ul style="list-style-type: none"> <li>➤ Création des usagers et des professionnels ;</li> <li>➤ Enrôler un professionnel de véhicule ;</li> <li>➤ Affichage des utilisateurs</li> </ul>
<b>Scénario nominal</b>	
<p><b>1. L'invité souhaite s'inscrire sur la plateforme</b></p> <p>1.1. Le système affiche l'interface concernant</p> <p>1.2. L'invité choisit le type de profil qu'il veut créer (usager ou professionnel)</p> <p>1.3. Le système affiche les champs adéquats au profil sélectionné</p>	

1.4. L'invité saisit les informations nécessaires et valide

1.5. Le système crée l'utilisateur

## **2. L'admin souhaite gérer ses utilisateurs**

2.1. L'admin clique sur le bouton gérer mes utilisateurs

2.2. Le système affiche l'interface concernant

2.3. L'admin choisit l'action à faire (modifier un utilisateur, bloquer, débloquer...)

2.4. Le système affiche la liste des utilisateurs disponibles

## **3. L'admin souhaite gérer les enrôlements**

3.1. L'admin demande de voir la liste des enrôlements

3.2. Le système affiche l'interface concernant

3.3. L'admin demande à traiter la demande d'enrôlement

3.4. Le système affiche un formulaire pré-rempli avec les infos de la demande

3.5. L'admin vérifie la conformité des données, étudie la faisabilité de la demande, renseigne les informations restantes et clique sur le bouton Valider

3.6. Le système valide la demande

## **4. Les utilisateurs (admin, usager, professionnel) souhaitent gérer leur compte**

4.1. L'utilisateur demande de voir les informations de son compte

4.2. Le système affiche l'interface concernant

4.3. L'utilisateur choisit l'action à faire (changement de mot de passe, modifier ces informations.)

4.4. Le système affiche le formulaire correspondant à la demande

4.5. L'utilisateur saisit les informations nécessaires et valide

4.6. Le système effectue la modification demandée



***Enchaînement alternatif***

**1.4. L'utilisateur existe déjà**

- a. Le système affiche un message d'erreur
- b. Le scénario reprend au point 1.3

**3.5.1. Demande non faisable**

- a. L'admin renseigne le motif du rejet et clique sur le bouton rejeter
- b. Le système rejette la demande

**3.5.2. Données non conformes**

- a. L'admin laisse un commentaire et clique sur le bouton rejeter
- b. Le système annule la demande

**4.5.1. Les données saisies sont incorrectes**

- a. Le système affiche un message d'erreur
- b. Le scénario reprend au point 4.4

Tableau 2 : Tableau descriptif du cas d'utilisation Gestion des utilisateurs

**3.3.4. Diagramme de cas d'utilisation « Gestion demande de prestation »**

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

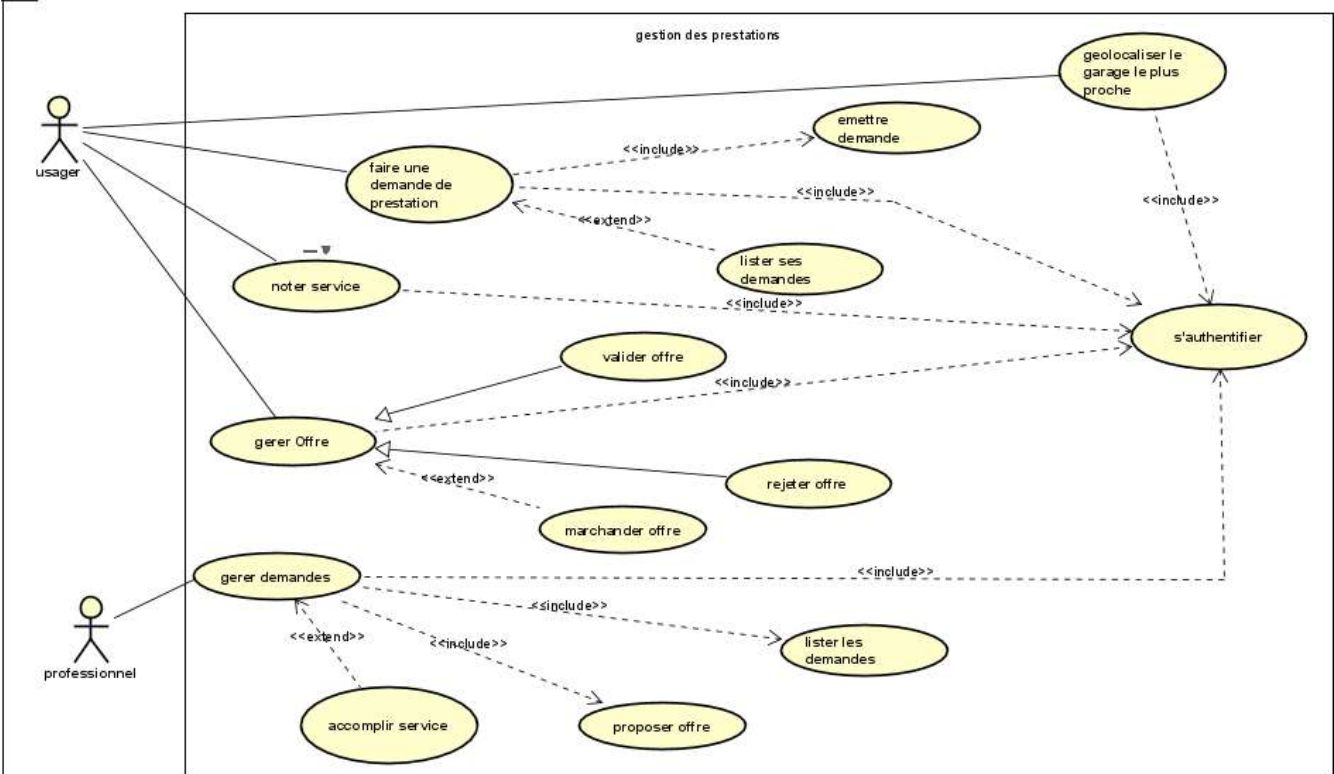


Figure 7 : Diagramme de cas d'utilisation gestion des demandes de prestation

SOMMAIRE D'IDENTIFICATION	
<b>Titre :</b>	Gestion des demandes de prestation.
<b>But :</b>	<ul style="list-style-type: none"> <li>➤ Permettre à un usager de localiser le garage plus proche, de faire une demande de prestation, lister ses demandes, noter les professionnels et de gérer ses offres.</li> <li>➤ Permettre à un professionnel de proposer des offres, lister ses demandes et d'accomplir des services demandés par les usagers.</li> </ul>
<b>Résumé :</b>	<p>L'utilisateur ayant besoin de services liés à l'automobile pourra localiser un professionnel dans un rayon non loin de lui et faire des demandes en fonction des offres de ces derniers.</p> <p>De même que pour le professionnel qui pourront gérer les demandes des usagers, accomplir les services demandés.</p>

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

<b>Acteur :</b>	Usager, professionnel de véhicule
<b>DESCRIPTION DES ENCHAINEMENTS</b>	
<b><i>Pré conditions</i></b>	<b><i>Post conditions</i></b>
S'authentifier	<ul style="list-style-type: none"> <li>➤ Localiser le garage le plus proche</li> <li>➤ Choisir la meilleure offre</li> <li>➤ Accomplir le service demandé</li> </ul>
<b><i>Scénario nominal</i></b>	
<p><b>1. L'utilisateur souhaite localiser le professionnel de véhicule le plus proche</b></p> <p>1.1. L'utilisateur demande à localiser des professionnels proches de lui</p> <p>1.2. Le système affiche le plan en fonction de la position de l'utilisateur</p> <p>1.3. L'utilisateur répertorie les garages les plus proches, fait une recherche au besoin sur un garage</p> <p>1.4. Le système affiche les informations (positions, services disponibles...) du garage</p> <p><b>2. L'utilisateur souhaite faire une demande de prestation</b></p> <p>2.1. L'utilisateur choisit un garage sur la liste des garages disponibles</p> <p>2.2. Le système affiche les services disponibles pour ce garage</p> <p>2.3. L'utilisateur choisit le service qu'il souhaite</p> <p>2.4. Le système affiche le formulaire de demande pour ce service</p> <p>2.5. L'utilisateur remplit les informations nécessaires et soumet sa demande</p> <p>2.6. Le système affiche les informations de la demande et le statut</p> <p><b>3. Le professionnel de véhicule liste ses demandes reçues et fait une offre</b></p>	

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

- 3.1. Le professionnel demande la liste des demandes des usagers
- 3.2. Le système affiche la liste des demandes reçues
- 3.3. Le professionnel choisit une demande, l'étudie et fait une proposition sur le service demandé
- 3.4. Le système enregistre la demande et notifie le client sur l'état de sa demande

### **4. L'utilisateur souhaite voir la liste de ses demandes**

- 4.1. L'utilisateur demande la liste de ces demandes
- 4.2. Le système affiche la liste des demandes
- 4.3. L'utilisateur choisit une demande, vérifie l'offre du professionnel et clique sur le bouton valider
- 4.4. Le système valide l'offre

### **5. L'utilisateur souhaite noter le professionnel**

- 5.1. L'utilisateur demande de voir la liste des professionnels
- 5.2. Le système affiche la liste des professionnels enrôlés
- 5.3. L'utilisateur choisit le professionnel et ajoute une note en fonction des différents services demandés
- 5.4. Le système rajoute la note sur profil du professionnel

### ***Enchaînement alternatif***

**2.2.5. Les données saisies sont incorrectes**

- a. Le système affiche un message d'erreur
- b. Le scenario reprend au point 2.1.4

**4.4.3 Offre non faisable**

- c. L'utilisateur renseigne le motif du rejet(commentaire) et clique sur le bouton rejeter
- d. Le système rejette l'offre

Tableau 3 : Tableau descriptif du cas d'utilisation Gestion des prestations

**3.4. Diagramme de séquence**

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent par le biais de messages

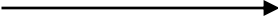



Les objets au cœur d'un système interagissent à travers des messages typés :

- **Message simple** : le message n'a pas de spécificité particulière d'envoi et de réception.
- **Message avec durée de vie** : l'expéditeur attend une réponse du récepteur pendant un certain temps et reprend ses activités si aucune réponse n'a lieu dans un délai prévu.

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

- **Message synchrone** : l'expéditeur est bloqué jusqu'au signal de prise en compte par le destinataire.
- **Message asynchrone** : le message est envoyé, l'expéditeur continue son activité que le message soit parvenu ou pris en compte ou non.
- **Message dérobant** : le message est mis en attente dans une liste d'attente de traitement chez le récepteur.

Les acteurs interagissent avec le système au moyen d'IHM c'est-à-dire d'une interface utilisateur permettant de connecter une personne à une machine, ou système ou appareil.

Symbole et description des messages		
Symbole	Nom	Description
	<b>Symbole de messages synchrones</b>	Représenté par une ligne pleine terminée par une pointe de flèche pleine. On utilise ce symbole lorsqu'un expéditeur doit attendre une réponse à un message avant de continuer. Le diagramme doit montrer à la fois l'appel et la réponse
	<b>Symbole de messages asynchrones</b>	Représenté par une ligne pleine terminée par une pointe de flèche. Les messages asynchrones ne nécessitent pas de réponse avant que l'expéditeur ne continue. Seul l'appel doit être inclus dans le diagramme.
	<b>Symbole de messages de retour asynchrones</b>	
	<b>Symbole de messages de création asynchrones</b>	Ces messages créent des nouveaux objets.


	<b>Symbole de messages de suppression</b>	Ces messages détruisent un objet
---	---	----------------------------------

Tableau 4 : Symbole, libellé et description des messages d'un diagramme de séquence

### 3.4.1. Les composants et symboles élémentaires d'un diagramme de séquence

Pour comprendre ce qu'est un diagramme de séquence, nous allons définir et détailler ses symboles et ses composants. Les diagrammes de séquence sont composés des icônes et des éléments suivants :

#### ➤ Symbole d'objet



Figure 8 : Représentation du symbole d'objet

Il représente une classe ou un objet en langage UML. Le symbole objet montre comment un objet va se comporter dans le contexte du système. Les attributs de classe ne doivent pas être énumérés dans cette forme.

#### ➤ Boite d'activation



Figure 9 : Représentation d'une boîte d'activation

Elle représente le temps nécessaire pour qu'un objet accomplisse une tâche. Plus la tâche nécessite de temps, plus la boîte d'activation est longue.

#### ➤ Symbole d'acteur

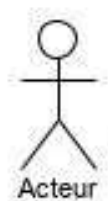


Figure 10 : Représentation d'un symbole d'acteur

Il montre les entités qui interagissent avec le système ou qui sont extérieures à lui.

### ➤ Symbole de paquetage



Figure 11 : Représentation de symbole paquetage

Utilisé dans la notation UML 2.0 pour accueillir les éléments interactifs du diagramme. Également connu sous le nom de « cadre », cette forme rectangulaire est représentée par un petit rectangle intérieur qui contient l'intitulé du diagramme.

### ➤ Symbole de ligne de vie



Figure 12 : Représentation d'une ligne de vie

Représente le passage du temps qui se prolonge vers le bas. Cette ligne verticale en pointillés montre les événements séquentiels affectant un objet au cours du processus schématisé. Les lignes de vie peuvent commencer par une forme rectangulaire avec un intitulé ou par un symbole d'acteur.

### ➤ Symbole de boucle optionnelle

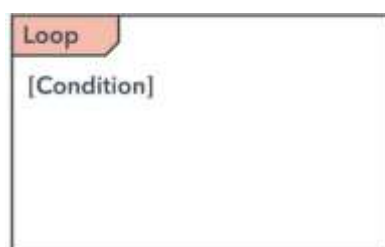


Figure 13 : Représentation de symbole de boucle optionnelle

On utilise ce symbole pour modéliser des scénarios ou une situation qui ne se produira qu'à certaines conditions.



## ➤ Symbole d'alternatives

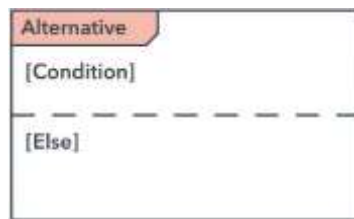


Figure 14 : Représentation de symbole d'alternative

Cette représentation symbolise des choix (qui en général s'excluent mutuellement) entre deux séquences de messages ou plus. Pour représenter les alternatives, nous utilisons la forme rectangulaire comportant un intitulé et une ligne en pointillés à l'intérieur.

### 3.4.2. Formalisme

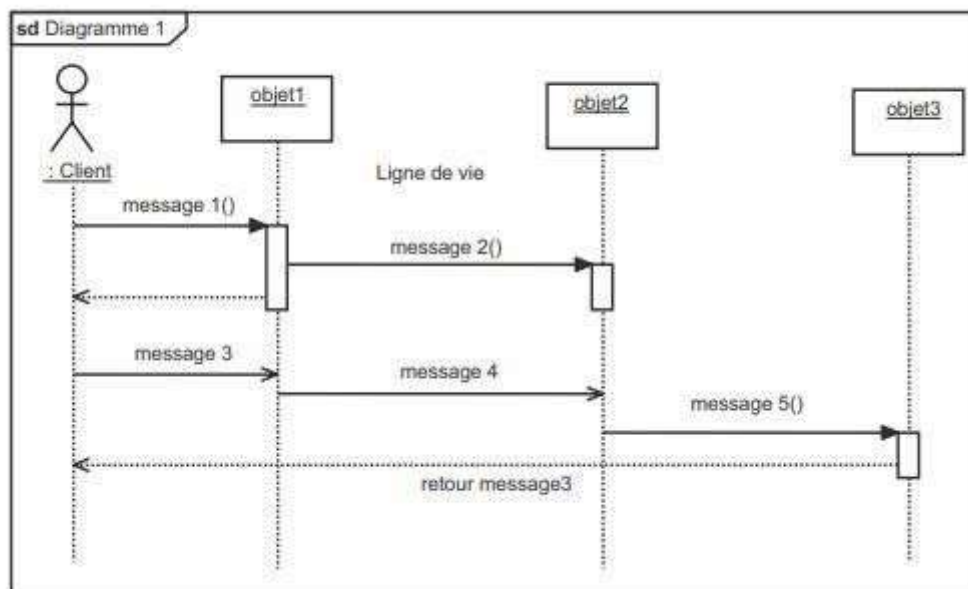


Figure 15 : Exemple de base d'un diagramme de séquence

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

### 3.4.3. Diagramme de séquence authentification

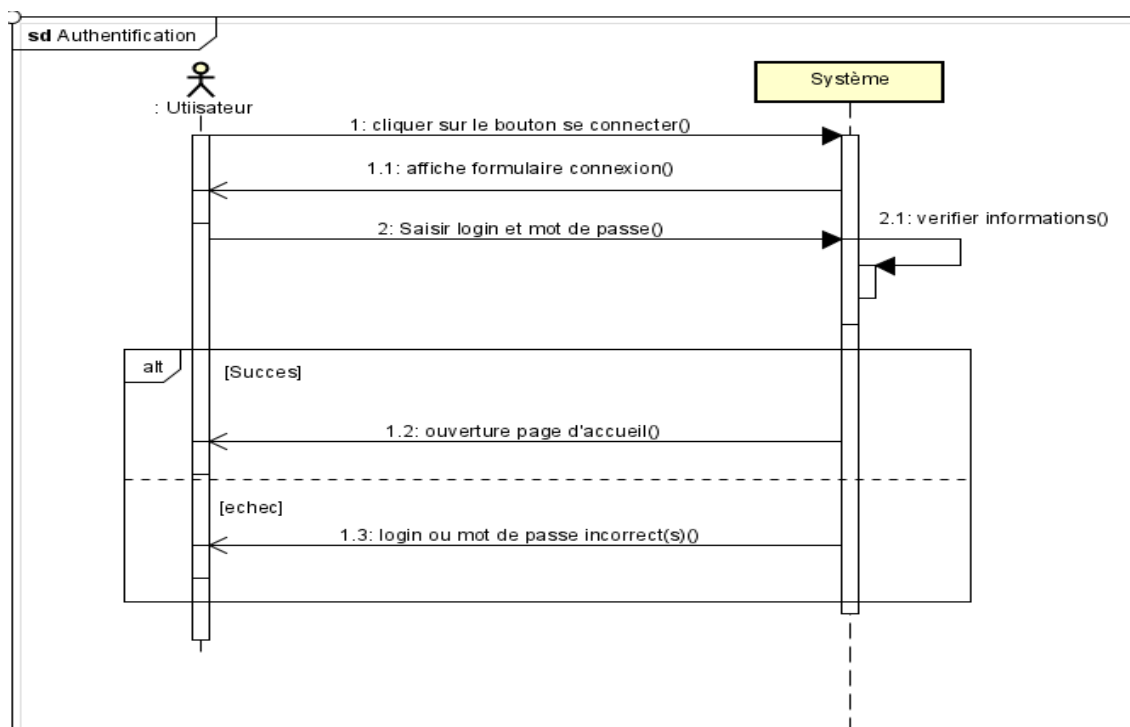


Figure 16 : Diagramme de séquence authentification

### 3.4.4. Diagramme de séquence inscription

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

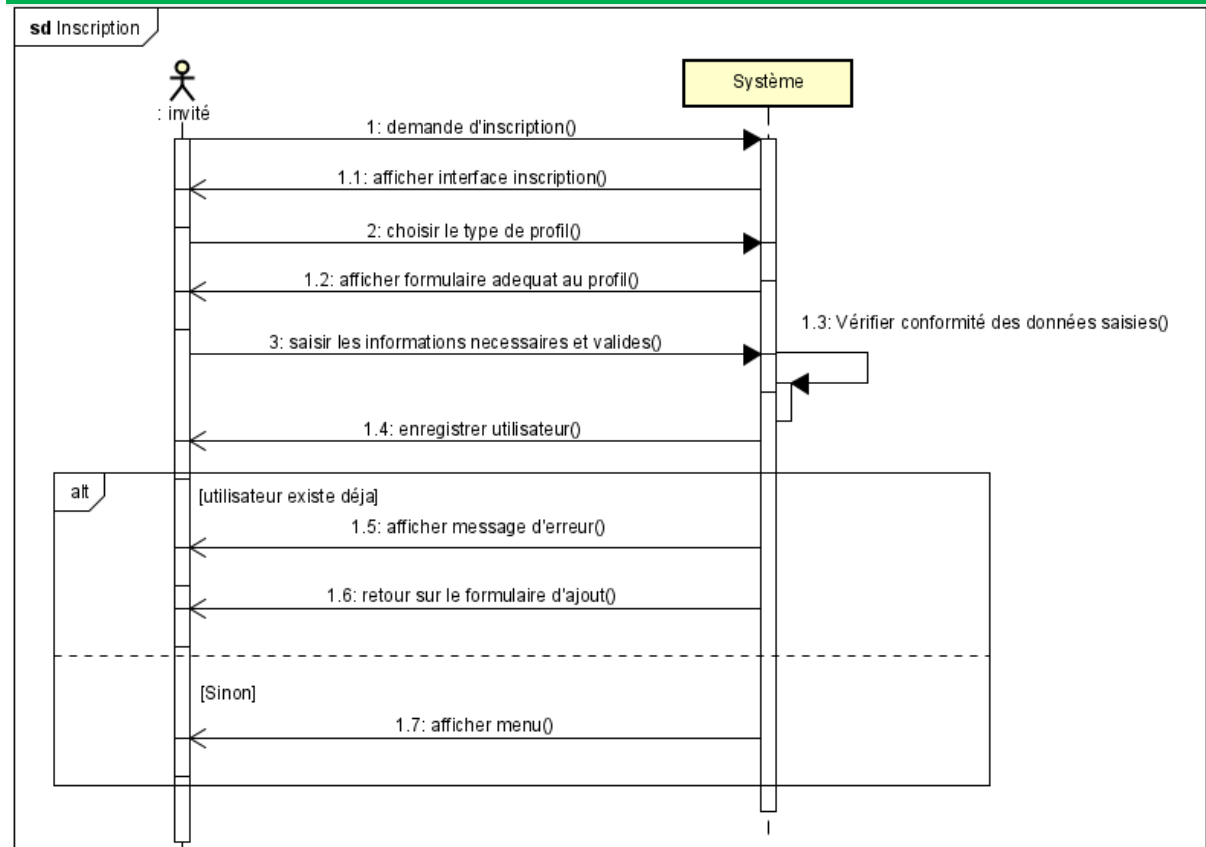


Figure 17 : Diagramme de séquence inscription

### 3.4.5. Diagramme de séquence demande de prestation

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

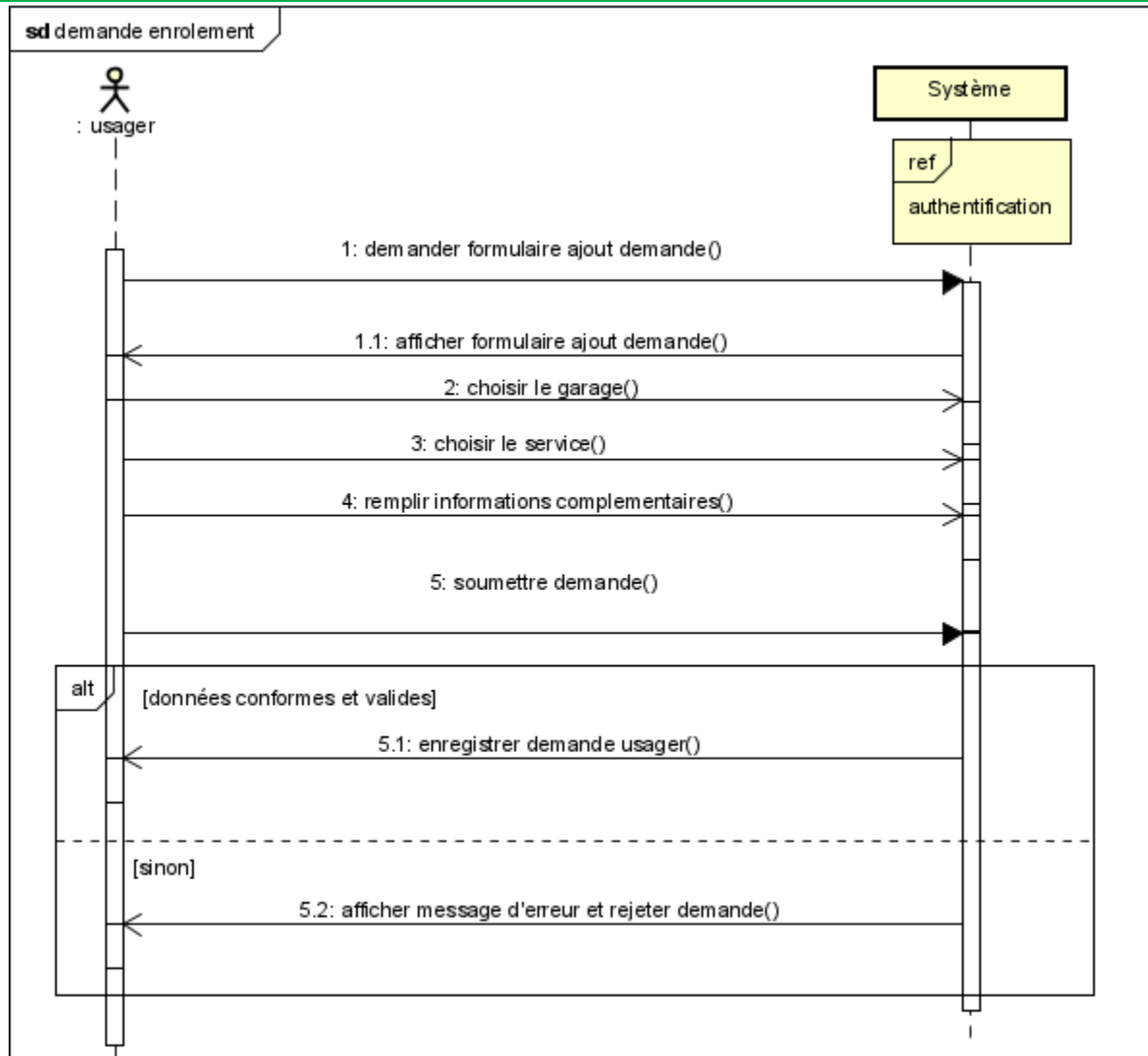


Figure 18: Diagramme de séquence demande de prestation

### 3.4.6. Diagramme de séquence traiter demande usager

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

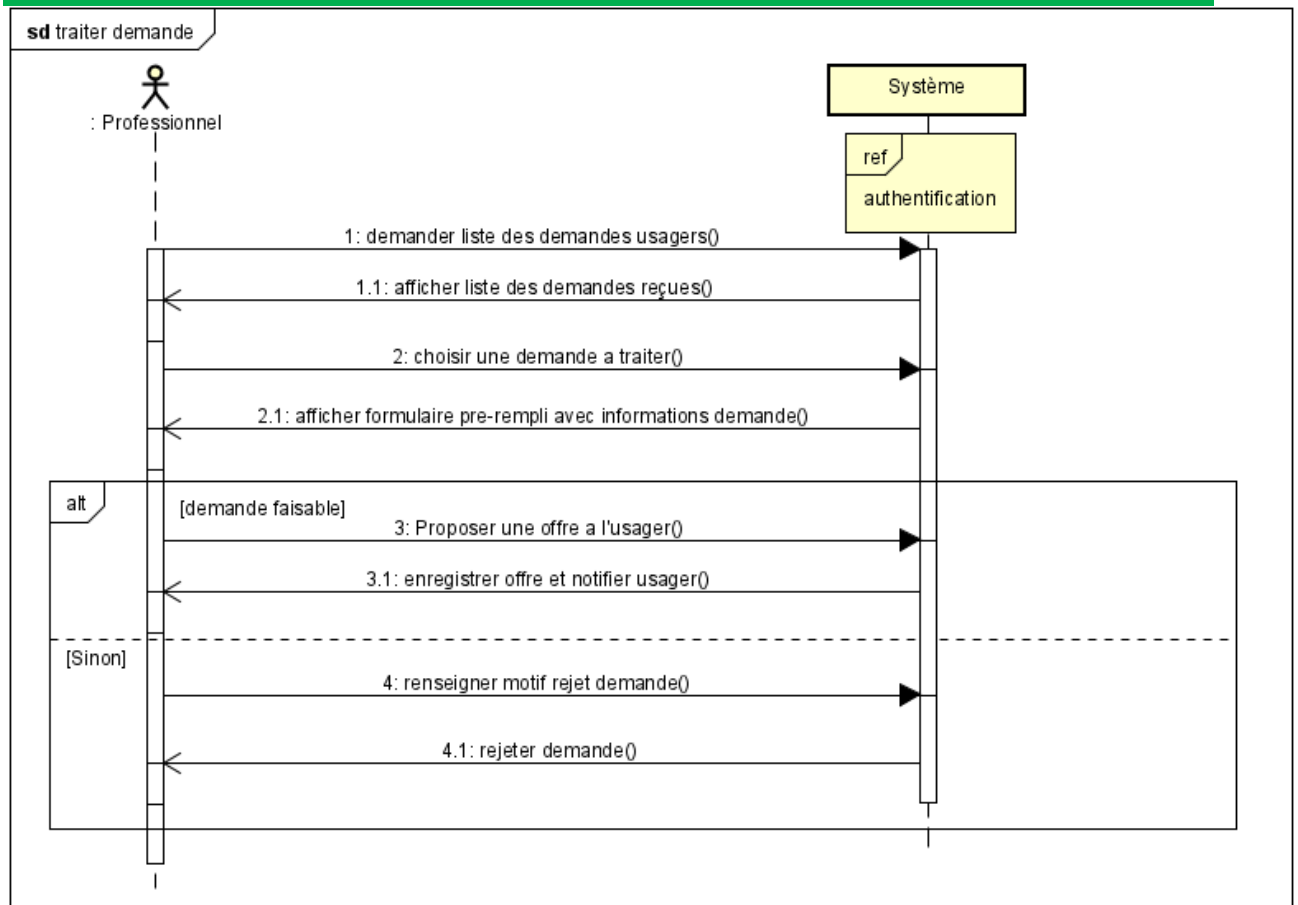


Figure 19: Diagramme de séquence traiter demande usager

### 3.5. Diagramme de classe

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Les cas d'utilisation ne réalisent donc pas une partition des classes du diagramme de classes. Un diagramme de classes n'est donc pas adapté (sauf cas particulier) pour détailler, décomposer, ou illustrer la réalisation d'un cas d'utilisation particulier.

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Chaque langage de Programmation orienté objet donne un moyen spécifique d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation. [6]

### ➤ Association

Une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc qu'il peut y avoir des liens entre des instances des classes associées.

### ➤ Multiplicité ou cardinalité

La multiplicité associée à une terminaison d'association, d'agrégation ou de composition déclare le nombre d'objets susceptibles d'occuper la position définie par la terminaison d'association.

Voici quelques exemples de multiplicité :

- Exactement un : 1 ou 1..1 ;
- Plusieurs : \* ou 0..\* ;
- Au moins un : 1..\* ;

Dans une association binaire, la multiplicité sur la terminaison cible contraint le nombre d'objets de la classe cible pouvant être associés à un seul objet donné de la classe source (la classe de l'autre terminaison de l'association).

Dans une association n-aire, la multiplicité apparaissant sur le lien de chaque classe s'applique sur une instance de chacune des classes, à l'exclusion de la classe-association et de la classe considérée. Par exemple, si on prend une association ternaire entre les classes (A, B, C), la

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

multiplicité de la terminaison C indique le nombre d'objets C qui peuvent apparaître dans l'association avec une paire particulière d'objets A et B.

### 3.5.1. Formalisme

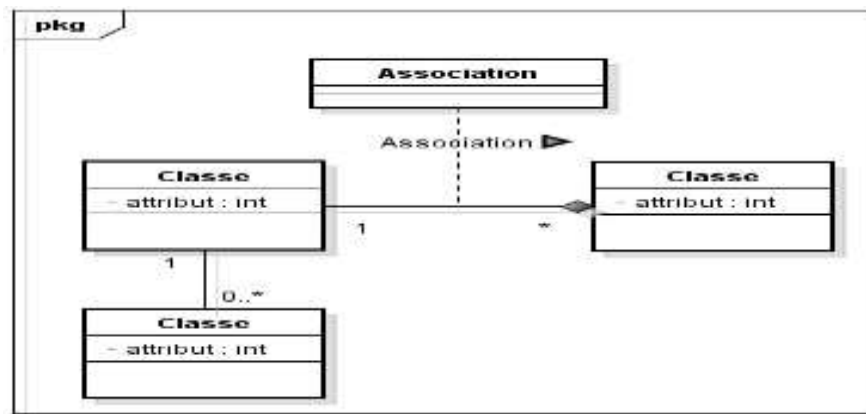


Figure 20 : Exemple de base diagramme de classe

### 3.5.2. Fiche descriptive des classes

Classes	Descriptions
Administrateur	Représente l'administrateur du système
Professionnel	Présente l'ensemble des professionnels de véhicule inscrit sur la plateforme
Usager	Répertorie l'ensemble des usagers
Demande	Représente les demandes faites par les usagers
Type de demande	Représente les types de demandes ou services disponibles pour un garage
Garage	Récense l'ensemble des garages
Enrôlement	Représente l'ensemble des enrôlements

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Commentaire	Récense l'ensemble des commentaires
Note	Représente les notes obtenues par le professionnel et données par l'utilisateur
Offre	Recense l'ensemble des offres faites par les usagers

Tableau 5: Fiche descriptive des classes

### 3.5.3. Diagramme de classe du système

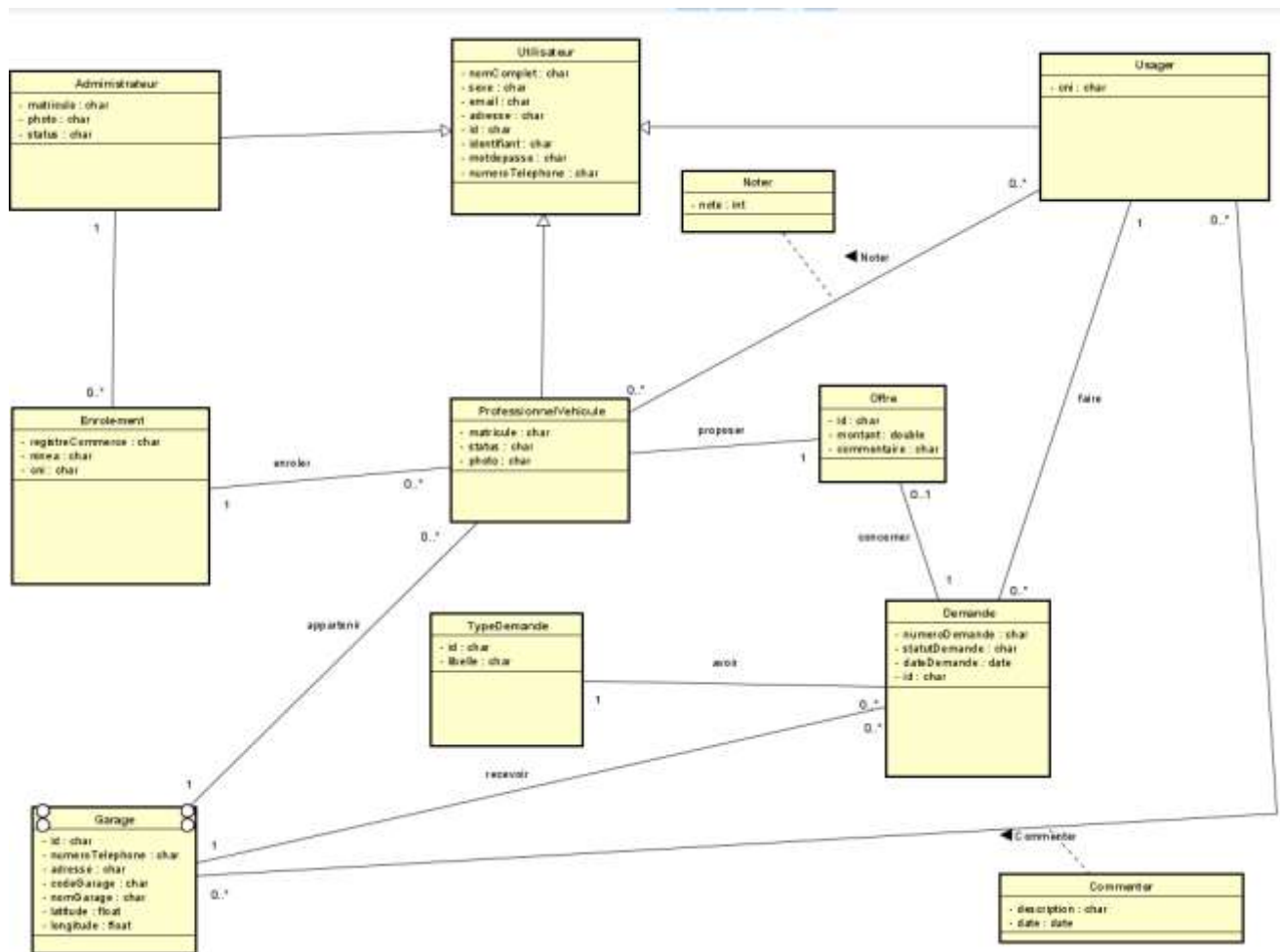


Figure 21: Diagramme des classes du système



### 3.6. Diagramme d'activité

Le diagramme d'activité fait partie des diagrammes comportementaux. Il est utilisé pour modéliser les aspects dynamiques d'un système. Il s'agit de représenter les opérations d'un processus et leurs conséquences sur les objets (logiciels ou matériels). La modélisation est utilisée pour décrire le déroulement d'un cas d'utilisation ou d'une méthode.

Les diagrammes d'activité affichent le flux de travail d'un point de départ à un point d'arrivée en détaillant les nombreux chemins de décision existant dans la progression des événements contenus dans l'activité.

#### 3.6.1. Les actions

Une action est le plus petit traitement qui puisse être exprimé en **UML**. Une action a une incidence sur l'état du système ou en extrait une information. Les actions sont des étapes discrètes à partir desquelles se construisent les comportements. La notion d'action est à rapprocher de la notion d'instruction élémentaire d'un langage de programmation (comme **C++** ou **Java**). Une action peut être, par exemple :

- Une affectation de valeur à des attributs ;
- La création d'un nouvel objet ou lien ;
- Un calcul arithmétique simple ;
- L'émission d'un signal ;
- La réception d'un signal.

#### 3.6.2. Les composants et symboles d'un diagramme d'activité

Le diagramme est relié par un point de départ et un point d'arrivée.



Figure 22 : Représentation des symboles d'un diagramme d'activité

On retrouve sur la figure ci-dessus de la gauche vers la droite : le point de départ, le point d'arrêt, nœud final de flot, un nœud de décision ou de fusion, un nœud de bifurcation, le nœud représentant une action, qui est une variété de nœud exécutable, un nœud objet.

#### 3.6.3. Diagramme d'activité demande d'enrôlement

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

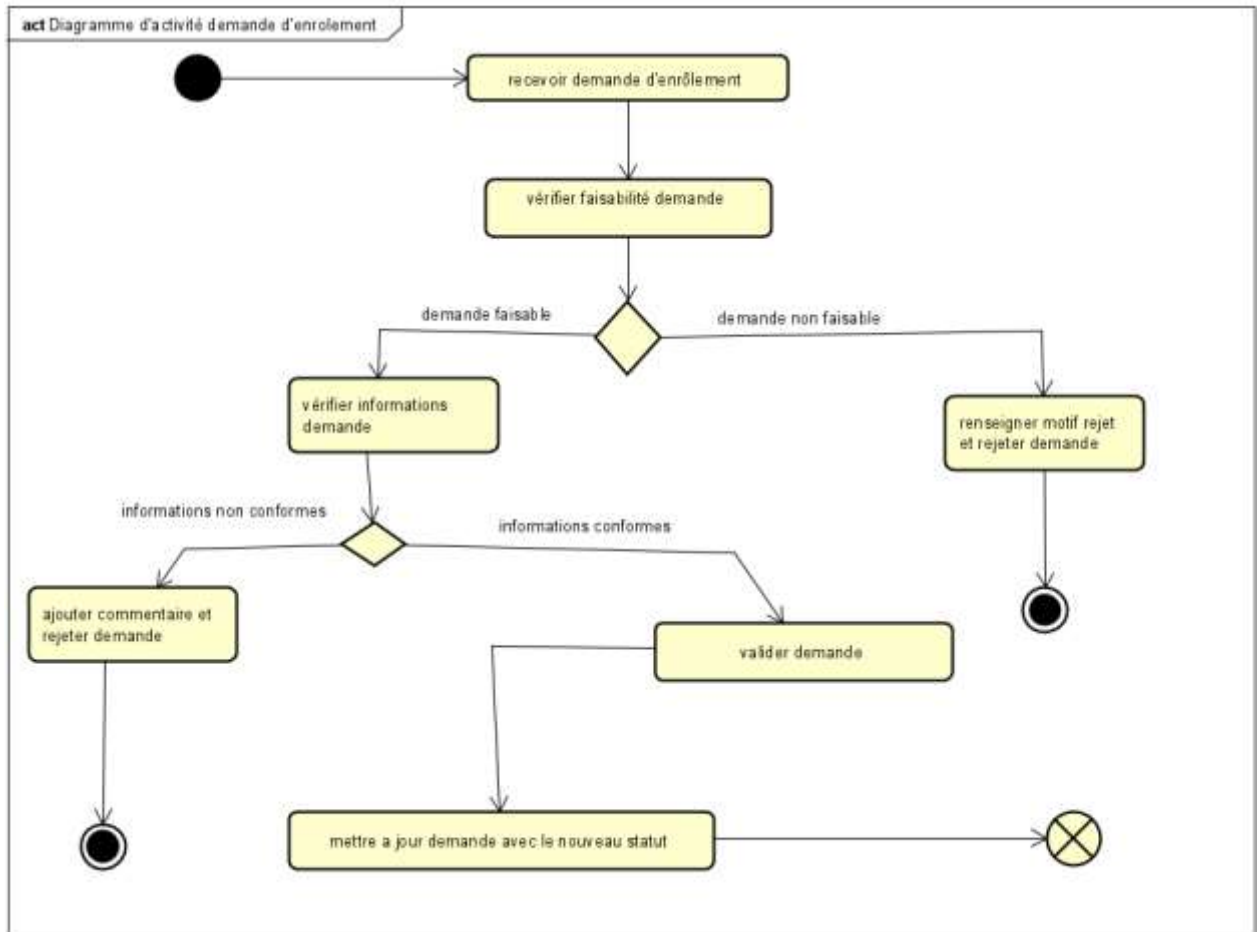


Figure 23 : Diagramme d'activité demande d' enrôlement

### 3.6.4. Diagramme d'activité géolocalisation professionnel véhicule

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

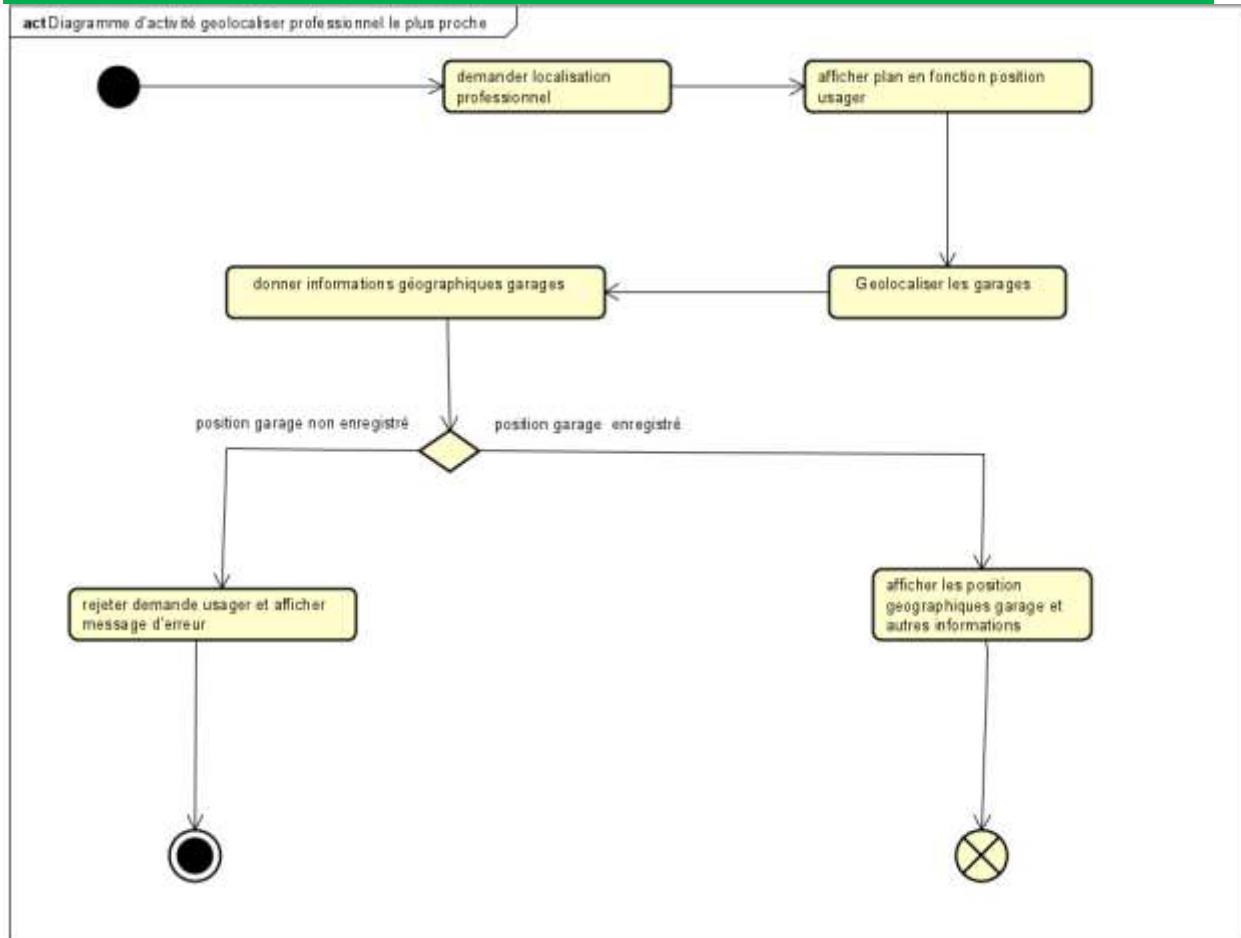


Figure 24 : Diagramme d'activité géolocalisation professionnel véhicule

### 3.7. Diagramme de déploiement

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype. Le diagramme de déploiement est utile pour :

- Montrer les éléments logiciels qui sont déployés par quels éléments matériels ;
- Illustrer le traitement d'exécution du point de vue matériel ;
- Visualiser la topologie du système matériel

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

### 3.7.1. Les éléments d'un diagramme de déploiement

Un diagramme de déploiement est composé de plusieurs formes :

- **Artefact** : Symbolisé par un rectangle avec le nom et le mot « artefact » entourés de flèches doubles.
- **Nœud** : Élément matériel ou logiciel représenté par une boîte en relief.
- **Association** : Ligne indiquant une communication entre deux nœuds.
- **Composant** : Rectangle avec deux onglets indiquant un élément logiciel.
- **Dépendance** : ligne en pointillés terminée par une flèche, qui indique qu'un nœud ou composant est dépendant d'un autre.
- **Interface** : Cercle qui indique une relation contractuelle. Les objets qui réalisent l'interface doivent remplir une sorte d'obligation.
- **Nœud conteneur** : Nœud qui en contient un autre.
- **Séréotype** : Dispositif contenu dans le nœud, présenté dans la partie supérieure du nœud et dont le nom est entouré de flèches doubles.

### 3.7.2. Diagramme de déploiement du système

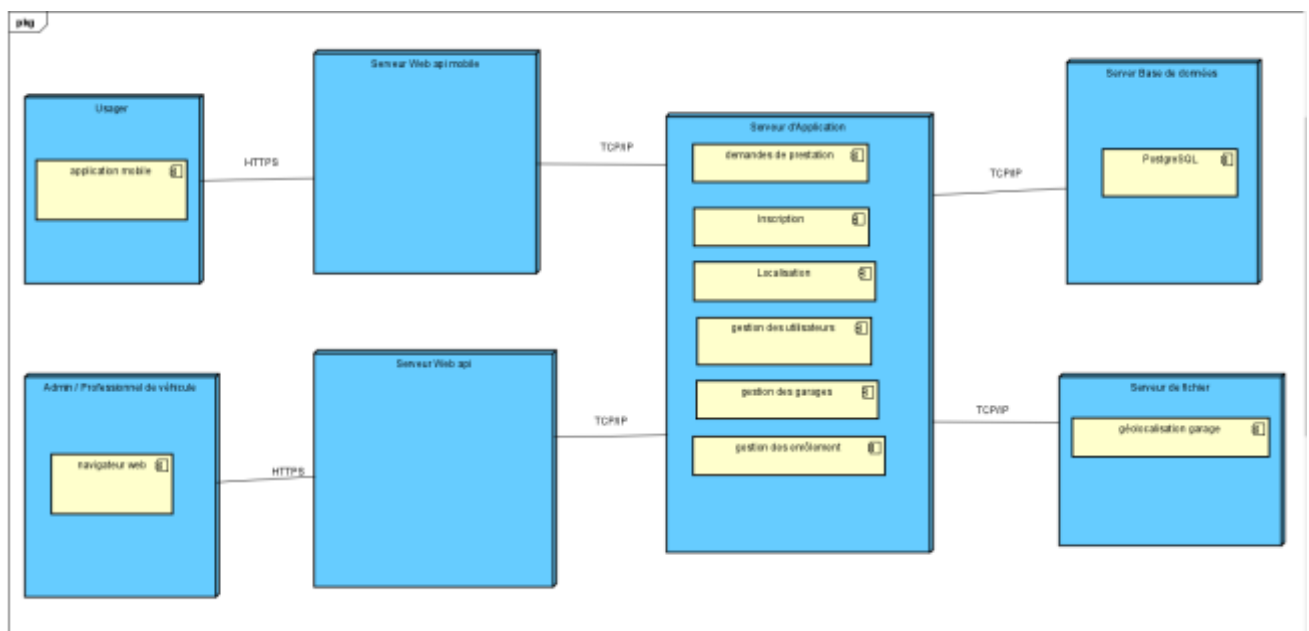


Figure 25: Diagramme de déploiement du système

### **3.8. Méthodologie**

La méthode d'analyse et de conception est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client.

La méthodologie Agile s'oppose généralement à la méthodologie traditionnelle waterfall. Elle se veut plus souple et adaptée, et place les besoins du client au centre des priorités du projet.

A l'origine, cette approche a été créée pour les projets de développement web et informatique. Aujourd'hui, la méthode Agile est de plus en plus répandue car elle est adaptable à de nombreux types de projets, tous secteurs confondus.

Par conséquent notre choix s'est porté sur la méthodologie Scrum pour différentes raisons.

#### **3.8.1. Définition**

**Scrum** est la méthodologie la plus utilisée parmi les méthodes Agiles existantes. Le terme Scrum (qui signifie mêlée) apparaît pour la première fois en 1986 dans une publication de Hirotaka Takeuchi et Ikujiro Nonaka qui décrit une nouvelle approche plus rapide et flexible pour le développement de nouveaux produits.

#### **3.8.2. Principe**

Evidemment, l'approche Scrum suit les principes de la méthodologie Agile, c'est-à-dire l'implication et la participation active du client tout au long du projet.

Considéré comme un cadre (framework en anglais) de gestion de projet, Scrum se compose de plusieurs éléments fondamentaux :

- Des **rôles**,
- Des **événements**,
- Des **artefacts**,
- Des **règles**.

Il s'agit d'une approche empirique (c'est-à-dire qui se base sur l'expérience), dynamique et participative de la conduite du projet. Au rugby, la mêlée est une phase indispensable car elle permet au jeu de repartir sur d'autres bases. Même chose pour Scrum : l'équipe se réunit quotidiennement lors d'une réunion de synchronisation, appelée mêlée quotidienne, afin de suivre l'avancement du projet.

### **3.8.3. Répartition des rôles**

L'équipe Scrum est auto-organisée et pluridisciplinaire, c'est-à-dire qu'elle choisit la meilleure façon d'accomplir son travail et qu'elle possède toutes les compétences nécessaires à l'accomplissement du projet. La flexibilité, la créativité et la productivité de l'équipe sont ainsi optimisées.

L'équipe Scrum se compose de :

- **Un Scrum Master**
- **Un Product Owner** (ou propriétaire du produit en français)
- **Une équipe de développement**

**Le Scrum Master** est responsable de la compréhension, de l'adhésion et de la mise en œuvre de la méthode Scrum qu'il maîtrise parfaitement. Il veille à ce que les principes et les valeurs de la méthodologie sont respectés. C'est un facilitateur qui aide à améliorer la communication au sein de l'équipe et cherche à maximiser la productivité et le savoir-faire de celle-ci. Il est considéré comme le coach de l'équipe de développement.

**Le Product Owner** porte la vision du produit à réaliser. Il travaille en interaction avec l'équipe de développement qui doit suivre ses instructions. C'est lui qui établit la priorité des fonctionnalités à développer ou à corriger, et qui valide les fonctionnalités terminées. Il est responsable de la gestion du product backlog (ou carnet de produit en français).

**L'équipe de développement** est chargée de transformer les besoins définis par le Product Owner en fonctionnalités utilisables. Elle est pluridisciplinaire et possède toutes les compétences nécessaires pour réaliser le projet, sans faire appel à des prestations externes. Parmi ses membres, on trouve un architecte, un développeur, un testeur, etc. La taille idéale de l'équipe de développement est de 3 à 9 personnes. Il n'y a pas de notion de hiérarchie, toutes les décisions sont prises ensemble.



Figure 26 : Illustration des rôles dans la méthodologie Scrum

#### **3.8.4. Les différents événements**

La vie d'un projet Scrum est rythmée par un ensemble de réunions définies avec précision et limitées dans le temps.

##### **a. Le Sprint**

Un Sprint est une itération. Il s'agit d'une période de 2 à 4 semaines maximum pendant laquelle une version terminée et utilisable du produit est réalisée. Un nouveau sprint commence dès la fin du précédent. Chaque sprint a un objectif et une liste de fonctionnalités à réaliser.

##### **b. Planification d'un Sprint**

Les tâches à accomplir pendant le Sprint sont déterminées par l'ensemble de l'équipe Scrum lors de la réunion de planification de Sprint. La durée de cette réunion est limitée à 8 heures pour les Sprints d'un mois. Cette réunion permet à l'équipe d'établir les éléments qu'elle traitera au cours de ce Sprint et comment elle procédera.

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

### c. Revue du Sprint

Il s'agit du bilan du Sprint réalisé. Une fois le Sprint terminé, l'équipe Scrum et les parties prenantes se réunissent pour valider ce qui a été accompli pendant le Sprint. Cette réunion dure 4 heures maximum.

### d. Rétrospective du Sprint

Cette réunion est interne à l'équipe Scrum et dure 3 heures pour un Sprint d'un mois. Le but est l'adaptation aux changements qui peuvent survenir et l'amélioration continue du processus de réalisation. L'équipe passe en revue le Sprint terminé afin de déterminer ce qui a bien fonctionné et ce qu'il faut améliorer.

### e. Mêlée quotidienne

Cette réunion quotidienne de 15 minutes est très importante. Elle se fait debout (d'où son nom anglais de “stand-up meeting”) afin d'éviter de s'éterniser. Le but est de faire un point sur la progression journalière du Sprint. Elle permet à l'équipe de synchroniser ses activités et de faire un plan pour les prochaines 24 heures.

La mêlée a lieu à la même heure et au même endroit chaque jour. Chaque membre de l'équipe de développement doit répondre à ces trois questions :

- Qu'est-ce qu'ils ont réalisé la veille ?
- Qu'est-ce qu'ils vont accomplir aujourd'hui ?
- Quels sont les obstacles qui les retardent ?

### 3.8.5. Les artefacts

#### a. Le product backlog (ou carnet du produit)

Il s'agit d'une liste hiérarchisée des exigences initiales du client concernant le produit à réaliser. Ce document évolue sans cesse durant le projet, en fonction des besoins du client. Le product owner est responsable du product backlog.

#### b. Le Sprint backlog (ou carnet de Sprint)

C'est le plan détaillé de la réalisation de l'objectif du Sprint, défini lors de la réunion de planification du Sprint. Le Sprint backlog est mis à jour régulièrement par l'équipe afin d'avoir une vision précise de la progression du Sprint.



## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

### c. L'incrément

Il s'agit de l'ensemble des éléments terminés du product backlog pour le Sprint en cours, ainsi que ceux des Sprints précédents. L'incrément doit fonctionner et être utilisable.

### d. Le Burndown Chart (ou graphique d'avancement)

Ce graphique simple indique l'état d'avancement dans la réalisation des tâches du Sprint backlog. Il s'agit du tracé de la charge de travail restante (exprimée généralement en heures) en fonction du temps (en jours). Le Burndown Chart est actualisé tous les jours par le Scrum Master après la mêlée quotidienne.

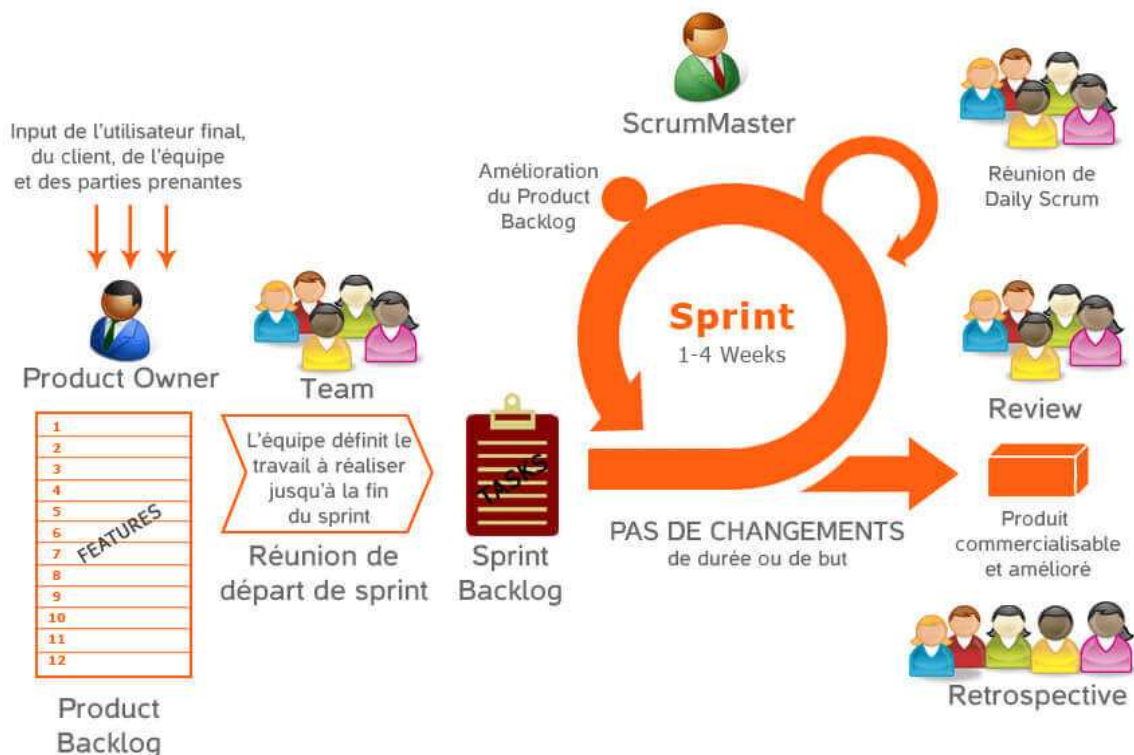


Figure 27 : Graphique d'avancement de la méthodologie Scrum

## **Chapitre 4 : REALISATION**

Le moment est venu de divulguer l'aspect pratique de notre projet. Dans cette partie, nous mettrons l'accent sur l'aspect visuel et en profiterons aussi pour parler des différents outils technologiques utilisés pour la réalisation de ce projet sans oublier les différentes interfaces utilisateur qu'il comporte. Notre plateforme est constituée d'une application web et d'une application mobile avec l'exposition de web services.

### **4.1. Architecture et Environnement**

#### **4.1.1. Architecture**

En informatique, une architecture applicative désigne la structure générale inhérente à un système informatique, l'organisation des différents éléments du système (logiciels et/ou matériels et/ou humains et/ou informations) et des relations entre les éléments. Cette structure fait suite à un ensemble de décisions stratégiques prises durant la conception de tout ou partie du système informatique, par l'exercice d'une discipline technique et industrielle du secteur de l'informatique dénommée elle aussi architecture, et dont le responsable est l'architecte informatique.

##### **4.1.1.1. Architecture en couche**

La couche logique n'est qu'un moyen d'organiser le code. Les couches typiques incluent la présentation, l'activité et les données, comme dans le modèle traditionnel à 3 couches. Mais quand on parle de couches, on ne parle que de l'organisation logique du code. Cela ne signifie en aucun cas que ces couches peuvent s'exécuter sur différents ordinateurs, ou dans différents processus sur un même ordinateur, ou même dans un seul processus sur un seul ordinateur. Tout ce que nous avons fait, c'est discuter d'une manière d'organiser le code en un ensemble de couches définies par des fonctions spécifiques.

##### **4.1.1.2. Architecture en tiers**

###### **4.1.1.2.1. Architecture 2 tiers**

Dans une architecture deux tiers, encore appelée client-serveur de première génération ou client-serveur de données, le poste client se contente de déléguer la gestion des données à un

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

service spécialisé. Le cas typique de cette architecture est une application de gestion fonctionnant sous Windows ou Linux et exploitant un SGBD centralisé. Ce type d'application permet de tirer parti de la puissance des ordinateurs déployés en réseau pour fournir à l'utilisateur une interface riche, tout en garantissant la cohérence des données, qui restent gérées de façon centralisée. La gestion des données est prise en charge par un SGBD centralisé, s'exécutant le plus souvent sur un serveur dédié. Ce dernier est interrogé en utilisant un langage de requête qui, le plus souvent, est SQL. Le dialogue entre client et serveur se résume donc à l'envoi de requêtes et au retour des données correspondant aux requêtes.

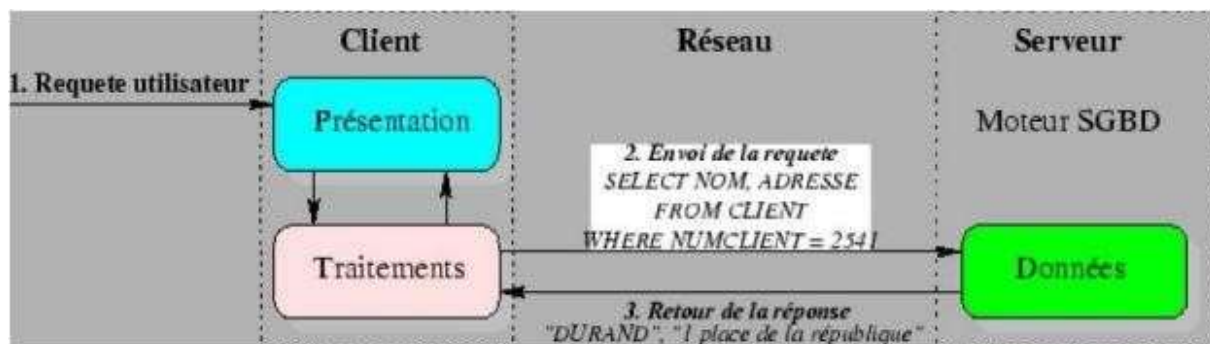


Figure 28 : Architecture 2 tiers

Les limites de l'architecture deux tiers proviennent en grande partie de la nature du client utilisé :

- Le frontal est complexe et non standard (même s'il s'agit presque toujours d'un PC sous Windows) ;
- Le middleware entre client et serveur n'est pas standard (dépend de la plate-forme, du SGBD...).

La solution résiderait donc dans l'utilisation d'un poste client simple communiquant avec le serveur par le biais d'un protocole standard.

### 4.1.1.2.2. Architecture 3 tiers

L'architecture trois tiers applique les principes suivants :

- Les données sont toujours gérées de façon centralisée ;
- La présentation est toujours prise en charge par le poste client ;

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

- La logique applicative est prise en charge par un serveur intermédiaire.

Cette architecture trois tiers, également appelée client-serveur de deuxième génération ou client-serveur distribué sépare l'application en 3 niveaux de services distincts, conformes au principe précédent :

- Premier niveau : l'affichage et les traitements locaux (contrôles de saisie, mise en forme de données...) sont pris en charge par le poste client ;
- Deuxième niveau : les traitements applicatifs globaux sont pris en charge par le service applicatif ;
- Troisième niveau : les services de base de données sont pris en charge par un SGBD.

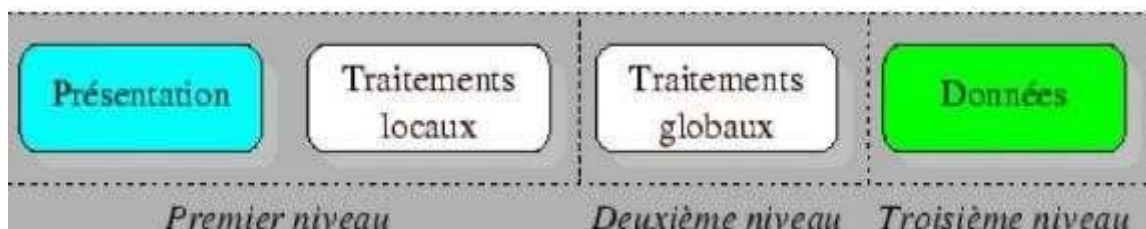


Figure 29 : Architecture 3 tiers

### 4.1.1.2.3. Architecture n-tiers

L'architecture n-tiers a été pensée pour pallier aux limites des architectures trois tiers et concevoir des applications puissantes et simples à maintenir. Ce type d'architecture permet de distribuer plus librement la logique applicative, ce qui facilite la répartition de la charge entre tous les niveaux.

Cette évolution des architectures trois tiers met en œuvre une approche objet pour offrir une plus grande souplesse d'implémentation et faciliter la réutilisation des développements.

Cette architecture est basée sur l'utilisation de composants "métier", spécialisés et indépendants, introduits par les concepts orientés objets (langages de programmation et middleware). Elle permet de tirer pleinement partie de la notion de composants métiers réutilisables.

Ces composants rendent un service si possible générique et clairement identifié. Ils sont capables de communiquer entre eux et peuvent donc coopérer en étant implantés sur des machines distinctes.

## Architecture n-tiers

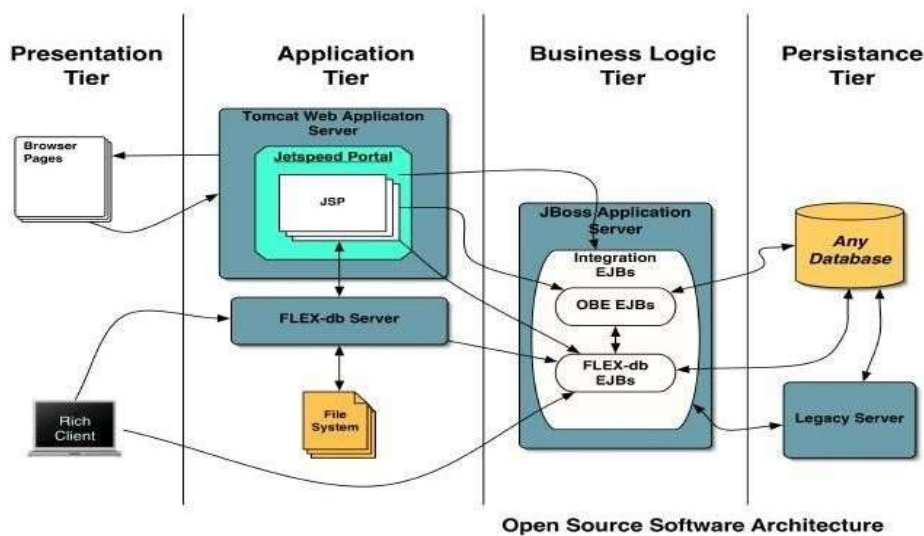


Figure 30 : Architecture n-tiers

### 4.1.1.3. Choix de l'architecture

Après étude nous avons opté pour une architecture n-tiers conformément aux besoins de notre projet suivant ces avantages ci-dessous :

- Réutilisation de services existants ;
- Découplage des aspects métiers et technique et des services entre eux : meilleure modularité ;
- Facilite évolution : nouvelle version de service
- Facilite le passage à l'échelle : évolution de certains services
- On recherche en général un couplage faible entre les services
- Permet de faire évoluer les services un par un sans modification du reste de l'application.

### 4.1.2. Environnements utilisés

#### 4.1.2.1. Framework

En programmation informatique, un Framework (appelé aussi cadre applicatif, cadre d'applications ou encore infrastructure de développement) désigne un ensemble cohérent de

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture). Les Framework sont conçus et utilisés pour modéliser l'architecture des logiciels applicatifs, des applications web, des middlewares et des composants logiciels. Les frameworks sont acquis par les informaticiens, puis incorporés dans des logiciels applicatifs mis sur le marché, ils sont par conséquent rarement achetés et installés séparément par un utilisateur final.

### 4.1.2.1.1. Définition de Laravel

Laravel a été créé par Taylor Otwell en juin 2011. Le référentiel Laravel/laravel présent sur le site GitHub contient le code source des premières Application Web de Gestion Scolaire : Cas Groupe Scolaire Ecole Plus - 77 - versions de Laravel. A partir de la cinquième version, le Framework est développé au sein du référentiel Laravel/Framework. En peu de temps, une communauté d'utilisateurs du Framework s'est constituée. Laravel, initie une nouvelle façon de concevoir un Framework en utilisant ce qui existe de mieux pour chaque fonctionnalité. Par exemple toute application web a besoin d'un système qui gère les requêtes HTTP. Plutôt que de réinventer quelque chose, le concepteur de Laravel a tout simplement utilisé celui de Symfony en l'étendant pour créer un système de routage efficace. En quelque sorte Otwell a fait son marché parmi toutes les bibliothèques disponibles.

### 4.1.2.1.2. Argument du choix de Laravel

Notre choix est porté sur Laravel car ce n'est pas seulement le regroupement de bibliothèques existantes, c'est aussi de nombreux composants originaux et surtout une orchestration de tout ça. Laravel est constitué de :

- Un système de routage perfectionné (RESTFul et ressources)
- Un créateur de requêtes SQL (langage de requête structurée) et ORM (objectrelational mapping) performants
  - Un moteur de Template efficace
  - Un système d'authentification pour les connexions
  - Un système de validation, de pagination, de migration pour bases de données, d'envoi d'emails
  - Un système d'autorisations
  - Une gestion des sessions ...

Durant cette section, nous présentons un scénario d'utilisation de l'application par quelques interfaces

#### **4.1.2.2. Le Langage de programmation**

##### **4.1.2.2.1. HTML**

Le HTML et sa variante plus stricte XHTML sont des langages de balisage des pages Web. Il n'y a pas si longtemps, le HTML servait à définir aussi bien la structure des pages que leur présentation visuelle. Aujourd'hui, ces deux aspects doivent être bien distincts et le X/HTML est destiné uniquement à représenter la structure d'une page : titres, sous-titres, paragraphes, images, formulaires de saisie, liens hypertextes, etc. C'est la base d'une page Web, parfois la seule considérée et utilisée par le logiciel qui visite cette page, comme les moteurs de recherche ou les navigateurs textuels. On qualifie de « statiques » les pages dont le code X/HTML n'est modifié ni par JavaScript, ni par PHP avant ou après l'affichage dans le navigateur.



##### **4.1.2.2.2. CSS**

Le code CSS (Cascading Style Sheets, ou feuilles de styles en cascade) permet de modifier la présentation des éléments X/HTML : couleur, taille, police de caractères, mais aussi position sur la page, largeur, hauteur, empilement, bref tout ce qui touche à la mise en page d'un document X/HTML. Ainsi, un même document X/HTML pourra changer d'apparence sans changer de structure, grâce uniquement à la modification des règles CSS qui lui sont appliquées. La séparation de la structure et de la présentation facilite ainsi la construction, mais aussi la maintenance et l'évolution des pages Web.



##### **4.1.2.2.3. Bootstrap**

Bootstrap a été développé en 2011 par l'équipe du réseau social Twitter. Bootstrap est un Framework frontend (HTML5, CSS et JavaScript) spécialement conçu pour le développement d'application web "responsive", c'est-à-dire qui s'adaptent automatiquement à différents dispositifs et tailles d'écran (tablettes, smartphones, desktop...etc.). Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. Il peut être utilisé pour créer par exemple, des pages de site de présentation, pour une interface graphique d'une application web ou être intégré à un thème d'un CMS, Bootstrap est de plus en plus utilisé, il est devenu « le plus populaire des



## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Framework FrontOffice pour développer des projets responsive et mobile-first sur le web. » (montuy, 2017)



### 4.1.2.2.4. PHP

La forme complète de PHP est le préprocesseur hypertexte. C'est un langage de script côté serveur qui vous aide à créer des sites Web dynamiques. Ce langage est principalement utilisé pour construire des applications logicielles basées sur le Web. C'est un logiciel open source et fonctionne bien avec MYSQL. Ce qui se passe réellement, c'est que le code PHP sera exécuté sur le serveur et que le code HTML sera affiché du côté du navigateur. (.xp-internet, n.d.)



Figure 4-5 : PHP logo

### 4.1.2.2.5. JQuery

JQuery est une bibliothèque JavaScript concise et rapide qui peut être utilisée pour simplifier la gestion des événements, le déplacement de documents HTML, les interactions Ajax et l'animation pour le développement rapide de sites Web. JQuery simplifie les scripts côté client du HTML, simplifiant ainsi le développement des applications Web 2.0. JQuery est une bibliothèque libre, opensource et à double licence sous licence publique générale GNU. Il est considéré comme l'une des Application Web de Gestion Scolaire : Cas Groupe Scolaire Ecole Plus - 82 - bibliothèques JavaScript (JS) préférées disponibles aujourd'hui. À partir de 2012, il est utilisé par plus de la moitié des meilleurs sites Web. (définition, 2017).



### 4.1.2.2.6. Ajax

Ajax est un acronyme pour Asynchronous JavaScript and XML (« XML et Javascript asynchrones ») et désigne une solution informatique libre pour le développement de pages dynamiques et d'applications Web. Le principal attrait d'Ajax réside dans le rendu dynamique des pages et une interactivité qui n'est pas sans rappeler le mode d'interaction du client lourd. L'avènement des interfaces riches (Rich Internet Application) permet au Web de fournir des sites associant une interactivité plus riche, jusque-là réservée au client lourd, tout en gardant une facilité de consultation qui leur sont propres. Pour expliquer le concept d'AJAX en



## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

quelques mots, il s'agit de permettre l'échange d'informations entre le navigateur web (sur le poste de l'utilisateur) et le serveur web (chez l'hébergeur) sans recharger l'ensemble de la page web utilisée. Ces échanges s'effectuent de façon asynchrone (le mode synchrone est aussi possible) à l'aide de fonctions développées en langage Javascript.



### 4.2. SGBD

#### 4.2.1. Présentation

##### ➤ SQL Server



Figure 31 : SQL Server

C'est le SGBD de Microsoft. Il a une administration et une fonction d'audit aisée. SQL Server a un niveau de SQL très près de la norme SQL et implémente presque toutes les possibilités de SQL. Malgré tous ses avantages, SQL Server est mono-plateforme (MS WINDOWS). Les distributions sont fortement liées au système d'exploitation.

##### ➤ Oracle Database



Figure 32 : Oracle Database

Utilisé par les très grosses entreprises, il est sans aucun doute un des SGBD les plus puissants et complets du monde informatique. Il a une richesse fonctionnelle et une fonction d'audit évolué. Malgré tout, le prix est élevé tant au point de vue des licences que des composants matériels (RAM, CPU) à fournir pour de bonnes performances. On a une administration complexe, liée à la richesse fonctionnelle. Il a cette faculté de gestion d'un gros volume de données et est multi-utilisateurs et multiplateformes.

##### ➤ MySQL



Figure 33 : MySQL

Probablement le SGBD le plus connu, MySQL est un serveur de base de données multi-utilisateurs fonctionnant sous Linux et Windows. Il est simple à mettre à œuvre et offre des performances du point de vue des temps de réponse et de stockage de données volumineuses. C'est une solution très courante en hébergement public.

Ses principaux atouts sont :

- ✓ La rapidité et la robustesse
- ✓ La facilité d'utilisation et la gratuité

#### ➤ PostgreSQL



Figure 34 : PostgreSQL

PostgreSQL est un système de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD.\

Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres comme MySQL ou propriétaires comme Oracle. Ce SGBDRO utilise des types de données modernes enrichis suivant les terminologies utilisées dans le vocabulaire informatique usuel. Ceci signifie que PostgreSQL peut stocker plus de types de données que les types simples traditionnels entiers, caractères, etc. L'utilisateur peut créer des types, des fonctions, utiliser l'héritage de type, etc.

#### **4.2.2. Choix du SGBD**

Après une étude comparative notre choix s'est porté sur PostgreSQL pour diverses raisons :

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

- **Architecture** : PostgreSQL utilise système de gestion de base est relationnel & objet à la différence de MySQL qui est rationnel. Dans un système de gestion de base de données relationnel-objet (SGBDRO) l'information est représentée sous forme d'objets comme dans la programmation orientée objet. Ces objets sont accessibles aux langages orientés-objets comme s'il s'agissait d'objets de ces langages (comme Java, C#, .NET, etc.). Cela confère à PostgreSQL des fonctionnalités qui facilitent la tâche des développeurs dans son exploitation.
- **Intégrité des données** : Il s'agit d'assurer à tout instant la cohérence, la fiabilité, et la pertinence des données que la base contient. Pour une application qui contient un nombre conséquent d'enregistrements, le choix de PostgreSQL est plus judicieux. Elle se concentre traditionnellement sur la robustesse et la fiabilité des données, c'est pourquoi les développeurs d'application préféreront ses fonctionnalités. Pour ne citer que cet exemple, elle dispose d'un planificateur de requêtes sophistiqué, capable de joindre (de « jointure ») efficacement un grand nombre de tables. Elle possède aussi des formats de tables qui peuvent être intéressants (Json entre autres) qui lui permettent aussi de bénéficier des avantages du NoSQL par exemple.
- **Maintenance de la base** : PostgreSQL est optimisé pour la production, en effet elle vient directement concurrencer Oracle sur ce point. Par exemple pour les backups : il n'est pas nécessaire d'arrêter la production comme sur MySQL, le système de backup et de réplication de PostgreSQL est plus adapté et récupère les données et les configurations tout en effectuant la mise à jour de façon autonome.

### 4.3. Outils

#### 4.3.1. Google Map

Google Maps est un service gratuit de carte géographique et de plan en ligne. Le service a été créé par Google. Il s'agit d'une forme de géo-portail lancé en 2004 aux États-Unis et au Canada et en 2005 en Grande Bretagne (sous le nom de Google Local).

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Ce service permet, à partir de l'échelle d'un pays, de pouvoir zoomer jusqu'à l'échelle d'une rue. Deux types de plan sont disponibles : un plan classique, avec nom des rues, quartier, villes et un plan en image satellite, qui couvre aujourd'hui le monde entier.

Google Maps dispose des fonctionnalités suivantes :

- Suivre votre chemin en réalité augmentée avec Live View (Live View est une fonction de réalité augmentée qui superpose les indications d'itinéraire à l'environnement réel) ;
- Enregistrer une carte hors ligne ;
- Planifier votre trajet même avec des étapes ;
- Trouver un parking plus facilement ;
- Vérifier un lieu avant de partir ;
- Télécharger des cartes pour les consulter hors-ligne ;
- Partager votre localisation avec vos amis.

### 4.3.2. Maven

Maven est un outil de construction de projets (build) open source développée par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine. Il permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java. Il permet notamment :

- D'automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet ;
- De gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet ;
- De générer des documentations concernant le projet ;

Pour gérer les dépendances du projet vis-à-vis de bibliothèques, Maven utilise un ou plusieurs dépôts qui peuvent être locaux ou distants. Maven est extensible grâce à un mécanisme de plugins qui permettent d'ajouter des fonctionnalités.

### 4.3.3. Apache Tomcat

Apache Tomcat est une implémentation open source d'un conteneur web qui permet donc d'exécuter des applications web reposant sur les technologies servlets et JSP. Tomcat est diffusé en open source sous une licence Apache. En tant qu'implémentation de référence de plusieurs versions des spécifications servlets/JSP, facile à mettre en œuvre et riche en

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

fonctionnalités, Tomcat est quasi incontournable dans les environnements de développements. Les qualités de ses dernières versions lui permettent d'être fréquemment utilisé dans des environnements de production. Tomcat est composé de plusieurs éléments :

- Coyote est le connecteur pour les protocoles de communications.
- Catalina est le conteneur de servlets
- Jasper est le moteur de JSP

### 4.3.4. IntelliJ IDEA

IntelliJ IDEA également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré (en anglais Integrated Development Environment - IDE) destiné au développement de logiciels informatiques reposant sur la technologie Java. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala.

De plus, IntelliJ IDEA Ultimate vous aide à développer des applications web full-stack grâce à ses puissants outils intégrés, à la prise en charge de JavaScript et de ses technologies connexes et à la prise en charge avancée de frameworks populaires tels que Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus et Helidon. IntelliJ IDEA peut être complété par des plugins gratuits développés par JetBrains afin de pouvoir travailler avec d'autres langages de programmation, parmi lesquels Go, Python, SQL, Ruby et PHP.

### 4.3.5. Android Studio

Android Studio est un environnement de développement pour réaliser des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, MacOS, Chrome OS et Linux. Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur

### 4.3.6. Astah UML

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Anciennement appelé Jude, Astah est un outil de modélisation UML créé par la compagnie japonaise Change Vision. Il fonctionne avec l'environnement d'exécution Java. Le nom vient de l'acronyme Java and UML developers environment. Astah est un logiciel propriétaire distribué gratuitement en version community. L'achat d'une licence "professional" permet de bénéficier d'un support client. Une fonctionnalité notable du logiciel est l'exportation en Java du modèle UML créé.



Figure 35: Logo astah uml

### 4.5. Sécurité

#### 4.5.1. Définition et Rôle

La sécurité est une phase très importante pour la création d'une application. En effet, la qualité d'une application dépend de sa sécurité les moyens à utiliser pour contrer certains types d'attaques. De plus, grâce à celle-ci l'application a plus de chance de rester longtemps sur le marché donc la confiance s'installe envers votre produit.

Les quatre critères de la sécurité sont :

##### ➤ Confidentialité

La confidentialité est cette caractéristique d'une information d'être accessible qu'à ceux qui y sont autorisés ;

##### ➤ Intégrité

L'intégrité est la caractéristique d'une information d'être modifiée que par des personnes autorisées et selon un procédé défini ;

##### ➤ Disponibilité

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

La disponibilité est la caractéristique d'une information d'être accessible et utilisable par son destinataire autorisé à l'endroit et à l'heure prévue ;

### ➤ **Traçabilité**

La traçabilité est la caractéristique qui conserve les traces de l'état et des mouvements de l'information. Sans elle, on n'a aucune chance d'avoir l'assurance que les trois autres critères sont respectés

### **4.5.2. Sécurité au niveau applicatif**

Il existe de nombreuses failles récurrentes dans l'application. Ces types d'attaques incluent une mauvaise gestion de l'authentification et des autorisations associées.

### ➤ **Authentification**

L'authentification pour un système informatique est un processus permettant de s'assurer de la légitimité de la demande d'accès faite par une entité afin d'autoriser l'accès de cette entité à des ressources du système conformément au paramétrage du contrôle d'accès.

L'authentification est une phase importante dans notre application. Pour cela nous allons utiliser le JWT Authentication pour gérer l'accès à notre plateforme.

JWT est un jeton permettant d'échanger des informations de manière sécurisée. Ce jeton est composé de trois parties, dont la dernière, la signature, permet d'en vérifier la légitimité.

### ➤ **Protection des routes**

Les routes nous permettent d'accéder à des fonctions précises de notre application. Donc il est nécessaire de bien protéger ces dernières pour que celle-ci ne soit pas accessible par des personnes tierces.

Pour ce faire nous allons utiliser les Auth Guards de Angular. Ces dernières nous permettent de faire de tel sorte que seules les personnes ayant les droits requis seront autorisées à accéder aux ressources de l'application.

### ➤ **Sécurité niveau base de données**

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

La base donnée est un élément important de tout système informatique. La plupart des données utilisateurs et consort y sont stockés. D'où la nécessité de bien veiller à la sécurité de cette dernière.

Pour se faire certaines données sensibles telles que les mots de passes seront chiffrées avant d'être sauvegardées. Les accès à notre serveur (username, password) seront changés régulièrement et ne seront connus que par des entités bien définies.

### 4.6. Présentation Solution

#### 4.6.1. Application web

Figure 36: Capture d'écran page inscription & recherche d'un service

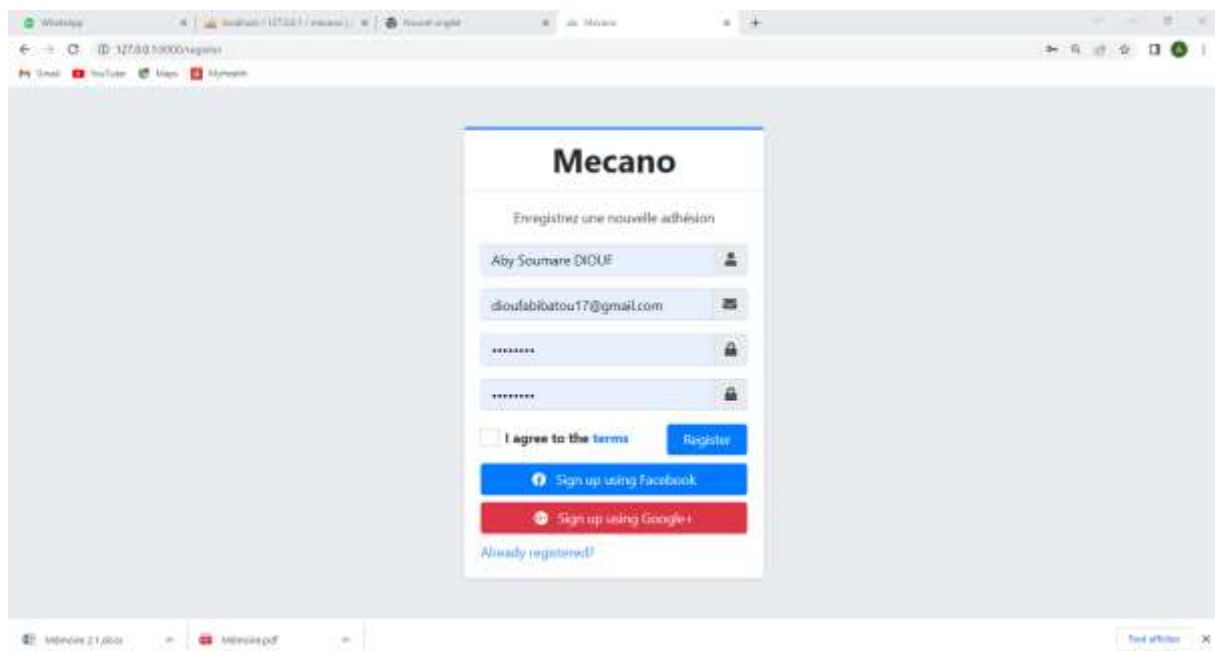


Figure 37: Capture d'écran page recherche garages



## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

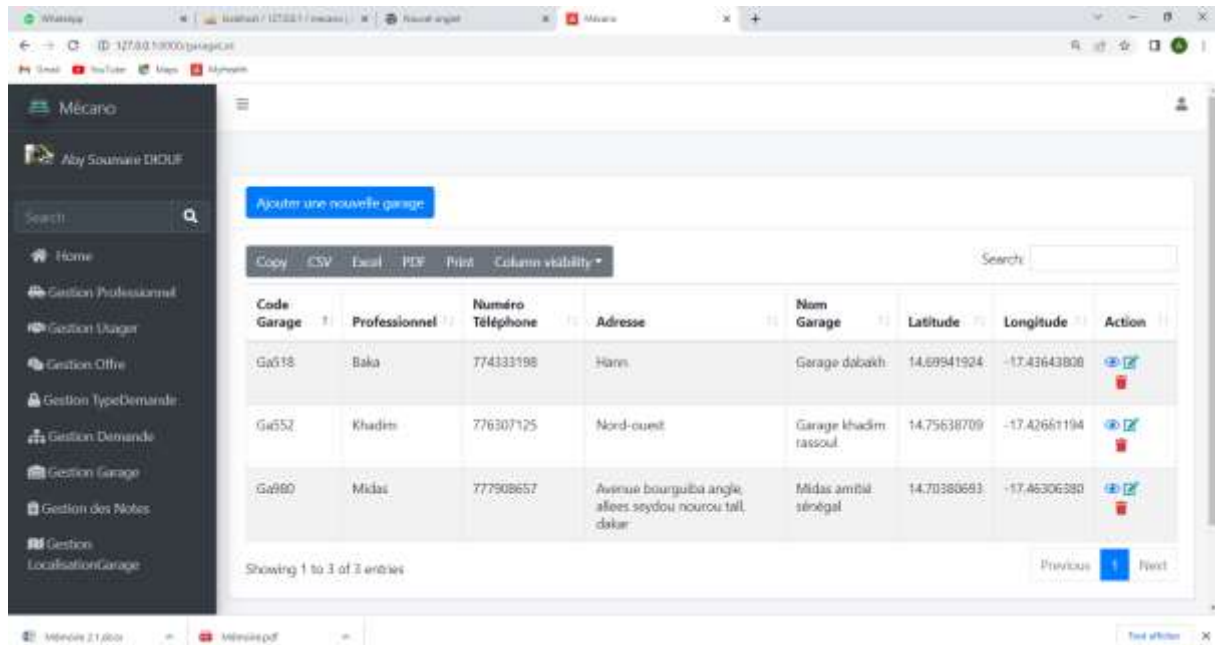
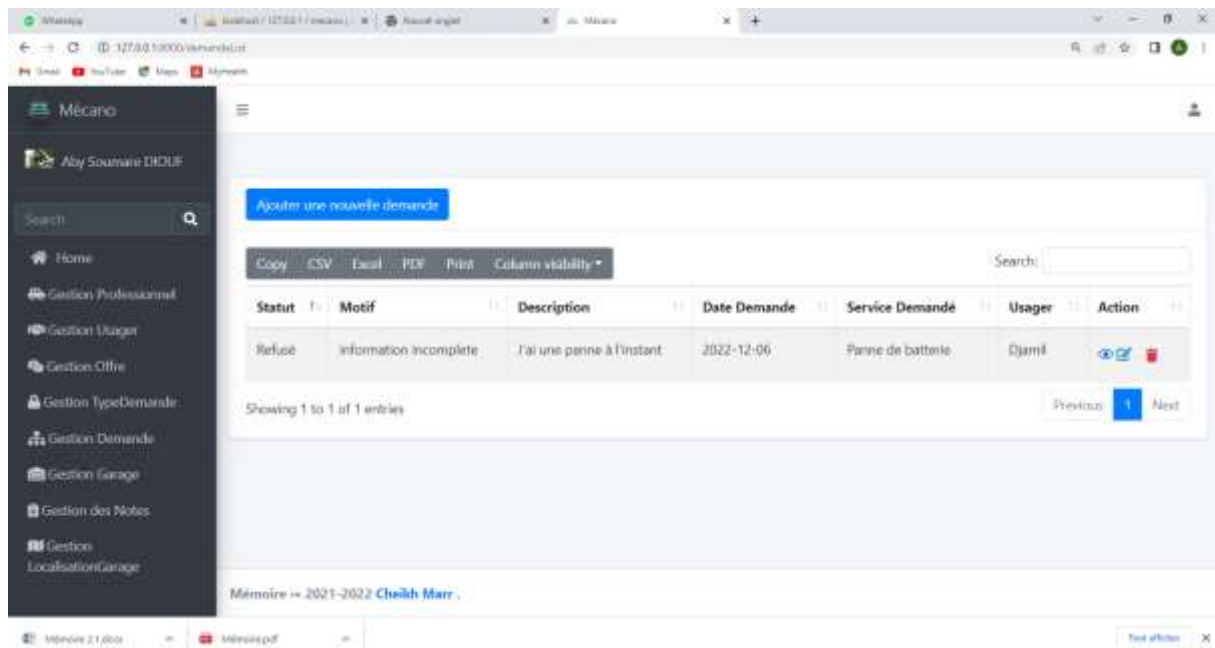


Figure 38: capture d'écran page des demandes



## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

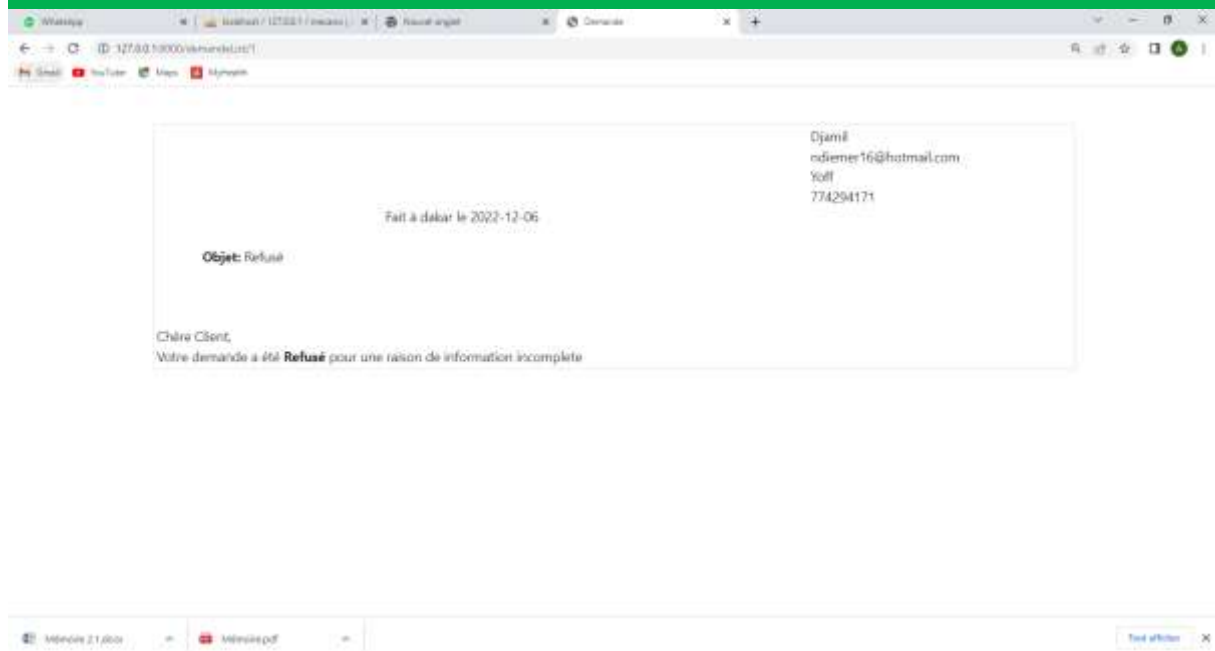


Figure 39: Capture d'écran page accueil

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

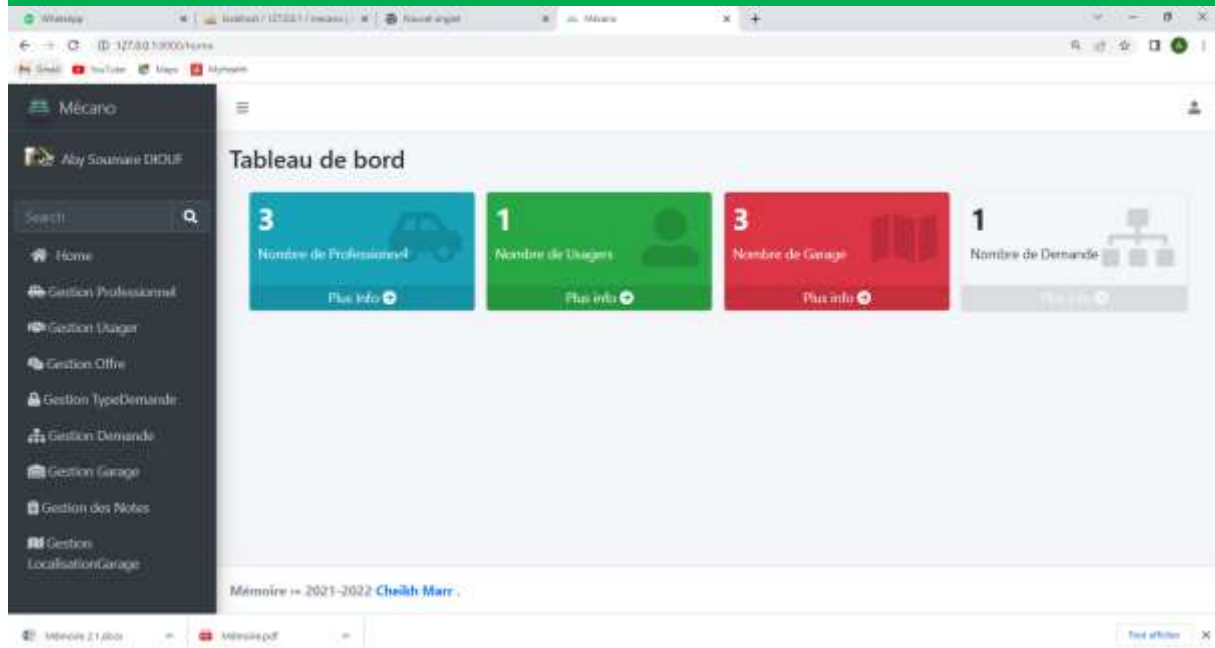


Figure 40: capture d'écran page de localisation de garages

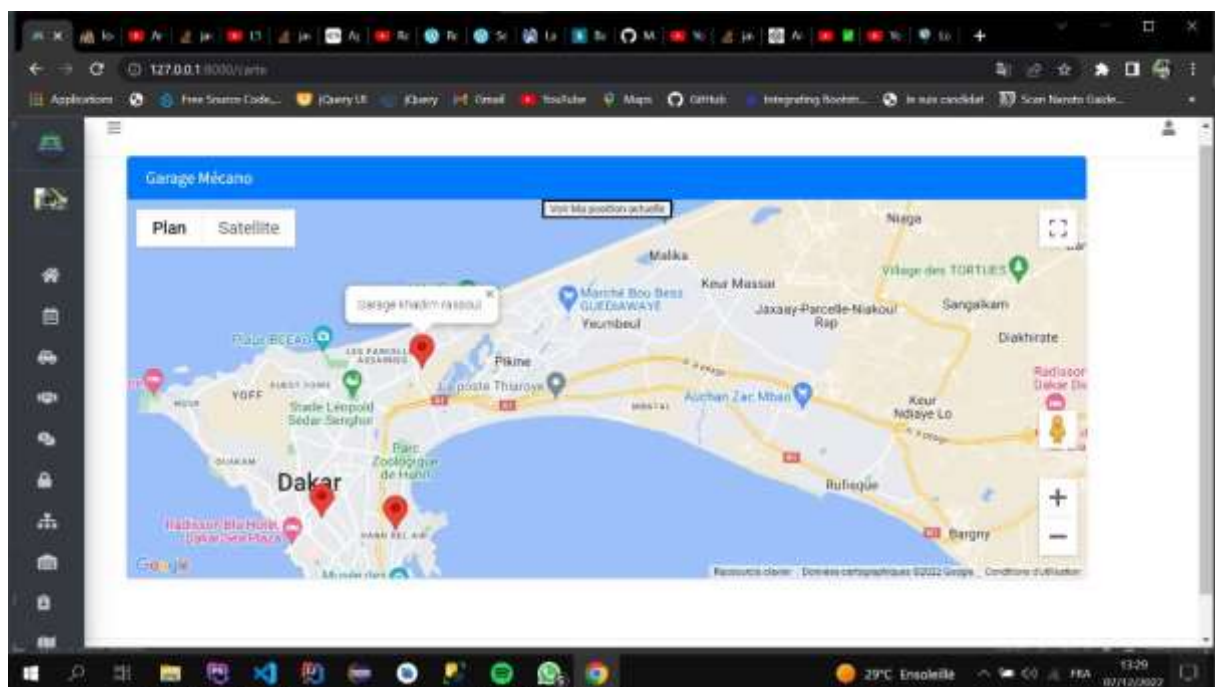


Figure 41: Capture d'écran page de connexion

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

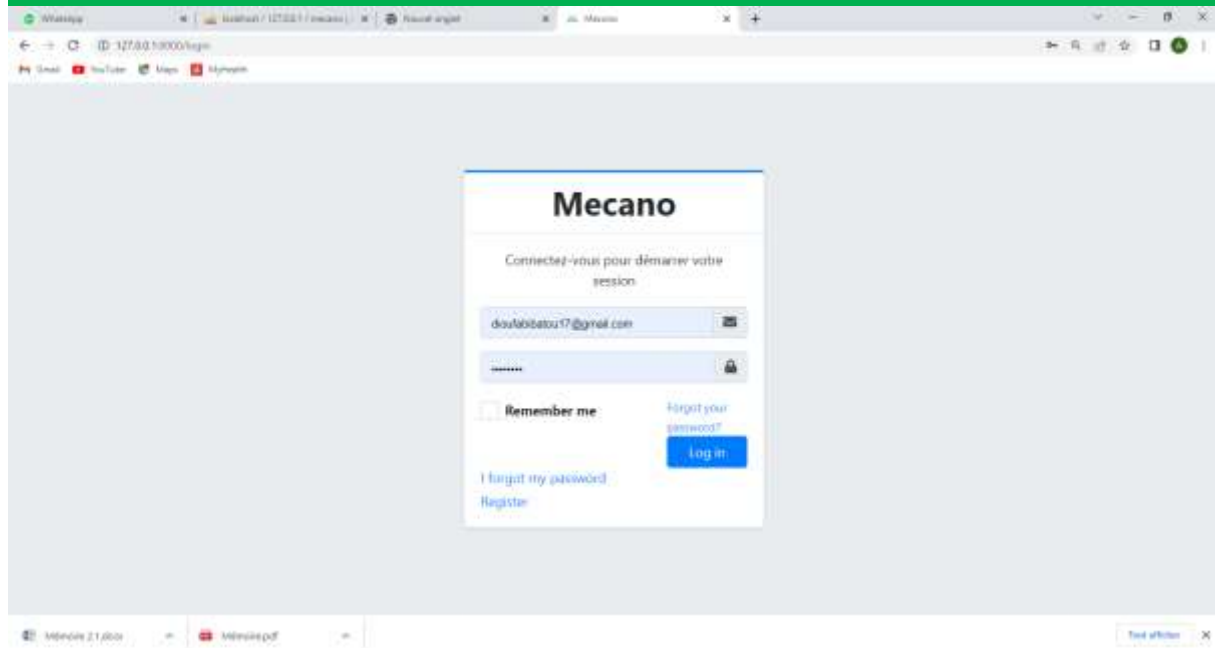
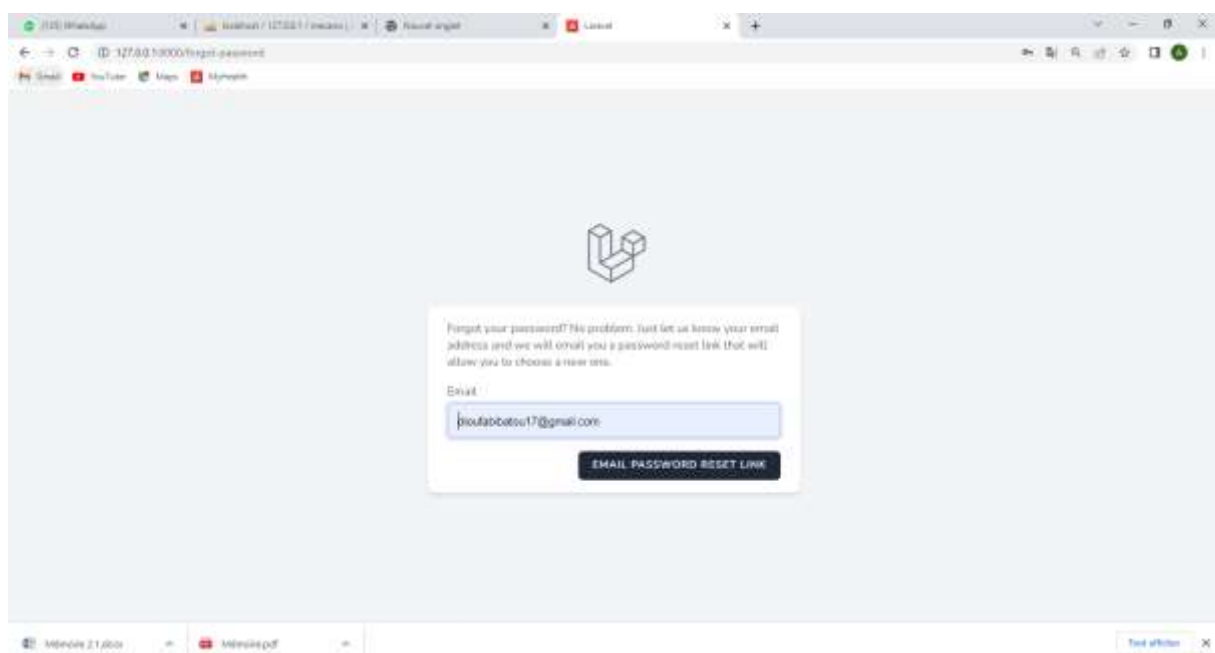
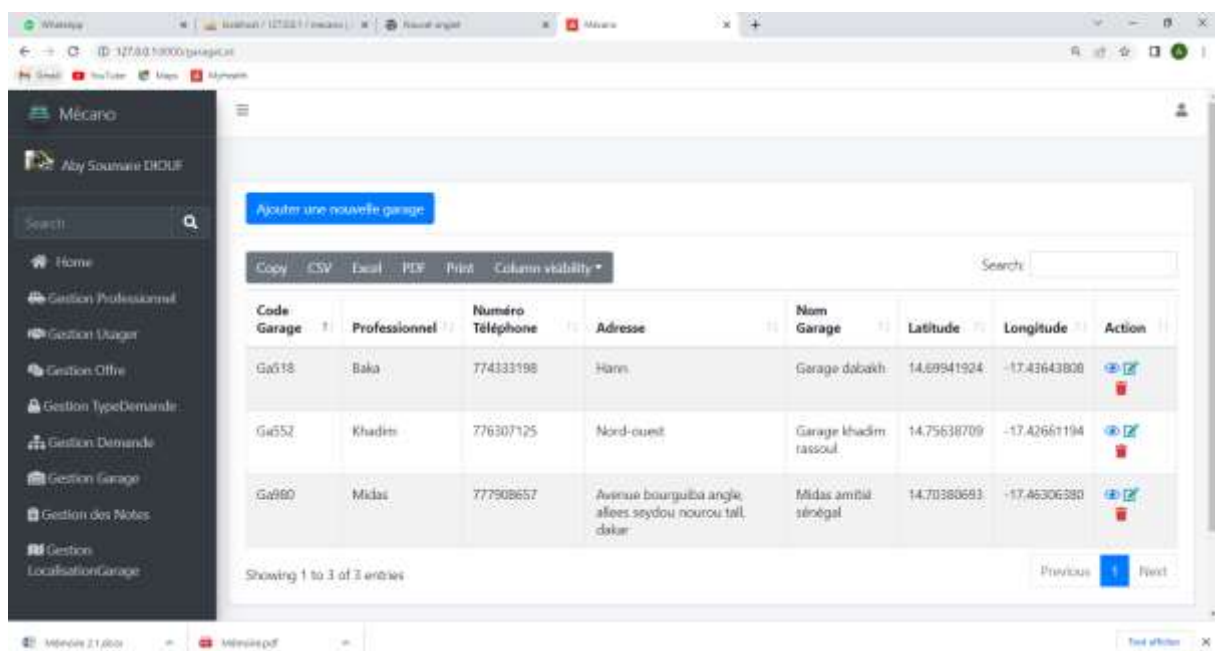


Figure 42: Capture d'écran page mot de passe oublié



## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

Figure 43: Capture d'écran liste des garages



Ajouter une nouvelle garage

Copy CSV Excel PDF Print Column visibility

Code Garage	Professionnel	Numéro Téléphone	Adresse	Nom Garage	Latitude	Longitude	Action
Ga018	Baka	774333198	Hann	Garage d'abakh	14.69941924	-17.43643808	<a href="#">+</a> <a href="#">-</a> <a href="#">x</a>
Ga552	Khadim	776307125	Nord-ouest	Garage khadim rassoul	14.75638709	-17.42661194	<a href="#">+</a> <a href="#">-</a> <a href="#">x</a>
Ga960	Midas	777508657	Avenue bourguiba angle allées seydou nourou tall dakar	Midas ambid sénégal	14.70380693	-17.46306380	<a href="#">+</a> <a href="#">-</a> <a href="#">x</a>

Showing 1 to 3 of 3 entries

Previous Next

## **CONCLUSION ET PERSPECTIVES**

L'objectif de notre étude était de mettre en place une application informatique qui servirait à cartographier et enrôler les professionnels du véhicule. Cette étude a pour finalité d'aider les particuliers à localiser facilement ces derniers et à avoir une meilleure offre qualité/prix pour le service demandé. A travers cette application, il fallait aussi répondre à la problématique de la gestion d'enrôlement de l'ensemble des professionnels du véhicule pour avoir des statistiques pour l'aide à la décision.

A ce jour, plusieurs fonctionnalités ont été implémenté tels que la recherche de garages de proximités, la gestion des garages etc... L'utilisateur peut consulter la liste des garages, en choisir un et émettre sa demande sur un service souhaité. Ainsi, le professionnel pourra consulter ses demandes et proposer une offre pour chacune d'elles. Après exécution du service, il sera noté.

Ce mémoire a fait l'objet d'un véritable projet car nous ayant permis de procéder à l'étude des méthodes de modélisation telle que la méthode UML que nous avons utilisé pour modéliser le système afin de déterminer les différentes vues ainsi que les principales fonctionnalités que notre application doit implémenter. Nous avons aussi choisi les outils nécessaires pour la mise en place de notre application de même que les mesures de sécurité à prendre pour la protéger contre toutes tentatives malintentionnées.

Bien qu'ayant rencontré de nombreuses difficultés, ce projet nous a permis de nous surpasser et aussi d'apprendre à retrouver des solutions ici et là sur Internet, des notions que nous n'avons pas reçues forcément en classe, favorisant ainsi l'accroissement de nos connaissances et attisant notre désir d'apprentissage et de perfectionnement.

## **Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers**

Au terme de cette étude, nous pouvons affirmer que la gestion numérique est une solution optimale pour la gestion des services offerts par les garages mécaniques. En guise de perspectives nous envisageons de :

- Permettre à un particulier d'avoir des garages favoris
- Intégrer les moyens de paiements (orange Money, Mastercard, Visa...)
- Répandre la solution au niveau continental

## **TABLE DES MATIERES**

Table des matières

A LA MEMOIRE DE .....	i
DEDICACE .....	ii
REMERCIEMENTS .....	iii
AVANT PROPOS.....	iv
SOMMAIRE .....	v
GLOSSAIRE.....	vii
LISTE DES FIGURES.....	viii
LISTE DES TABLEAUX.....	x
RESUME .....	xi
ABSTRACT.....	xi
Chapitre 1 : INTRODUCTION GENERALE.....	1
1.1. Contexte.....	2
1.2. Problématique.....	3
1.3. Objectifs .....	4
1.3.1. Objectif général .....	4
1.3.2. Objectifs spécifiques .....	4
1.4. Intérêt du sujet.....	4
Chapitre 2 : TRAVAUX EXISTANTS.....	6
2.1. Présentation de la solution Vroomly .....	6
2.1.1. Fonctionnalités .....	6
2.1.2. Avantages .....	6
2.1.3. Inconvénients.....	7
2.2. Présentation de la solution Proximeca.....	7
2.2.1. Fonctionnalités .....	7
2.2.2. Avantages .....	7
2.2.3. Inconvénients.....	8
2.3. Présentation de la solution IDGARAGES.....	8
2.3.1. Fonctionnalités .....	8
2.3.2. Avantages .....	9



## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

2.3.3.	Inconvénients.....	9
2.4.	Présentation de la solution Wutilma.....	9
2.4.3.	Inconvénients.....	10
2.5.	Critique de l'existant .....	10
2.6.	Proposition de notre solution.....	10
Chapitre 3 : ANALYSE ET CONCEPTION .....		11
3.1.	Présentation et choix de la méthode d'analyse.....	11
3.1.1.	Présentation .....	11
3.1.1.1.	Merise.....	11
3.1.1.2.	UML .....	12
3.1.2.	Etude comparative des méthodes Merise et UML.....	13
3.1.3.	Choix de la méthode.....	14
3.2.	Diagramme de contexte.....	14
3.3.	Diagramme de cas d'utilisation.....	15
3.3.1.	Formalisme .....	16
3.3.2.	Description détaillée des cas d'utilisations.....	16
3.3.3.	Diagramme de cas d'utilisation « Gestion des utilisateurs ».....	17
3.3.4.	Diagramme de cas d'utilisation « Gestion demande de prestation ».....	20
3.4.	Diagramme de séquence.....	24
3.4.1.	Les composants et symboles élémentaires d'un diagramme de séquence.....	26
3.4.2.	Formalisme .....	28
3.4.3.	Diagramme de séquence authentification.....	29
3.4.4.	Diagramme de séquence inscription.....	29
3.4.5.	Diagramme de séquence demande de prestation .....	30
3.4.6.	Diagramme de séquence traiter demande usager .....	31
3.5.	Diagramme de classe.....	32
3.5.1.	Formalisme .....	34
3.5.2.	Fiche descriptive des classes .....	34
3.5.3.	Diagramme de classe du système .....	35
3.6.	Diagramme d'activité.....	36
3.6.1.	Les actions .....	36
3.6.2.	Les composants et symboles d'un diagramme d'activité .....	36
3.6.3.	Diagramme d'activité demande d'enrôlement.....	36
3.6.4.	Diagramme d'activité géolocalisation professionnel véhicule.....	37

## Conception et réalisation d'une plateforme de mise en relation entre les professionnels du véhicule et les usagers

3.7.	Diagramme de déploiement.....	38
3.7.1.	Les éléments d'un diagramme de déploiement .....	39
3.7.2.	Diagramme de déploiement du système .....	39
3.8.	Méthodologie.....	40
3.8.1.	Définition.....	40
3.8.2.	Principe.....	40
3.8.3.	Répartition des rôles .....	41
3.8.4.	Les différents événements .....	42
3.8.5.	Les artefacts.....	43
Chapitre 4 : REALISATION .....		45
4.1.	Architecture et Environnement.....	45
4.1.1.	Architecture .....	45
4.1.2.	Environnements utilisés.....	48
4.2.	SGBD .....	52
4.2.1.	Présentation .....	52
4.2.2.	Choix du SGBD.....	53
4.3.	Outils .....	54
4.3.1.	Google Map.....	54
4.3.2.	Maven.....	55
4.3.3.	Apache Tomcat.....	55
4.3.4.	IntelliJ IDEA .....	56
4.3.5.	Android Studio .....	56
4.3.6.	Astah UML.....	56
4.5.	Sécurité.....	57
4.5.1.	Définition et Rôle .....	57
4.5.2.	Sécurité au niveau applicatif .....	58
4.6.	Présentation Solution.....	59
4.6.1.	Application web.....	59
CONCLUSION ET PERSPECTIVES .....		65
TABLE DES MATIERES .....		67
BIBLIOGRAPHIE .....		70
WEBOGRAPHIE.....		71

## **BIBLIOGRAPHIE**

### **Ouvrages**

Muriel GLATIN & Maïke STRUDTHOFF, La géolocalisation - Nouvelle arme des marketeurs, Paris, 2019, 178 pages.

Joseph Gabay et David Gabay, UML 2 Analyse et Conception, Algeria-Educ.com, Dunod, Paris, 2008, 242 pages

### **Articles**

Iris Borel, Informatique et Numérique, La géolocalisation : une technologie qui change le monde

## **WEBOGRAPHIE**

<http://www.memoireonline.com>: 10/04/2022, 12h07

[Géolocalisation des salariés : que dit la loi ? - Cadremploi](#) : 10/03/2022, 12h07

<https://fr.wikipedia.org>: 12/04/2022, 00h55

[article-geolocalisation.pdf \(open.global\)](#) : 15/05/2024, 22h35

[http://www.ansd.sn/ressources/publications/13-SES-2017-2018\\_Transport.pdf](http://www.ansd.sn/ressources/publications/13-SES-2017-2018_Transport.pdf) : 29/03/2022, 19h16

<https://www.sage.com/fr-fr/sage-business-cloud/compta-facturation/fonctionnalites/> : 26/05/2022 19h23

<https://www.proximeca.fr/> : 26/05/2022 21:57

<https://www.cours-gratuit.com/cours-java> : 04/06/2022 21:38

<https://www.jetbrains.com/fr-fr/idea/features/> : 09/06/2022 23:11

<https://fr.igraal.com/avis/idgarages> : 15/03/2022 14h16

<https://www.vroomly.com/> : 18/03/2022 13h00

<https://www.wutilma.com/#/website/index> : 18/03/2022 17h00

<https://startup.orange.com/fr/candidature/wutilma-technologies/> : 18/03/2022 16h00

<https://laravel.com/docs/8.x> : 18/02/2022 16h00

<https://developer.mozilla.org/fr/docs/Web/Guide/AJAX> : 18/11/2022 19h00

<https://api.jquery.com/> : 15/11/2022 16h00

---

i <https://fayauto.sn/situation-de-lindustrie-automobile-du-senegal-en-2020/>

