

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**



**Projet de Fin d'Etudes  
Licence Sciences et Techniques Génie Informatique**

---

**Développement d'une application mutualisée de Gestion  
de Parc Automobile pour les Administration Publiques**

---



**Lieu de stage :** *Ministère de la réforme de l'administration et de la  
fonction publique*

**Réalisé par :**

El Gaddar Chaimae

El Hamss Romaissa

**Encadré par :**

Pr. CHAKER Ilham

Mr. SANDADI El Mehdi

Mr. BOUKHANI Mohammed

**Soutenu le 08/06/2017 devant le jury composé de :**

Pr. K. ZENKOUAR

Pr. A. BOUSHABA

Pr. I. CHAKER

Année Universitaire 2016-2017



## Remerciement

*Après ALLAH, nous tenons à remercier :*

*Toutes les personnes qui ont participé de différentes façons à la réussite de notre stage et qui nous a aidé lors de la rédaction de ce rapport.*

*En premier lieu nos parents qui nous ont soutenus tout au long de notre vie. Nous espérons avant tout que nous serons toujours votre fierté. Que DIEU vous bénisse.*

*Ainsi, nous adressons nos remerciements à notre encadrante académique, Madame Ilham Chaker qui nous a beaucoup aidé dans nos recherches de stage. Son écoute, ses conseils, et son énorme gentillesse nous a permis de cibler nos buts, et de trouver ce stage qui était en totale adéquation avec nos attentes.*

*Nous tenons à remercier vivement nos encadrants professionnels, Mr BOUKHANI (responsable du service informatique au sein du ministère de la fonction public), et Mr SENDADI (ingénieur développement informatique) pour leur accueil, le temps passé ensemble et le partage de leur expertise au quotidien. Grâce aussi à leur confiance nous avons pu nous accomplir totalement dans nos missions. Il fut d'une aide précieuse dans les moments les plus délicats (merci à monsieur SINDADI).*

*Nous saisissons aussi l'occasion pour remercier les membres du jury : Pr K. ZENKOUAR et Pr A. BOUSHABA, qui nous ont fait l'honneur d'accepter de juger notre travail.*

*Nos remerciements les plus respectueux vont à tout le cadre professoral de la FST, pour la formation qu'il nous a prodiguée. Que tous ceux qui nous ont aidés, de près ou de loin, trouvent ici l'expression de nos meilleurs sentiments.*



## Table des matières

Liste des figures.....	4
Liste des tableaux.....	5
Liste des abréviations.....	6
Introduction générale.....	7
Chapitre 1 : Contexte général du projet.....	8
Introduction.....	9
I. Organisme d'accueil.....	9
1. Ministère : .....	9
2. Direction des systèmes d'information : .....	11
II. Présentation du projet.....	12
A. Problématique.....	12
1. Mise en position.....	12
2. Description du métier.....	13
B. Critique de l'existant.....	17
C. Solution proposée.....	18
D. Cahier des charges.....	18
Conclusion.....	20
Chapitre 2 : Méthodologie du développement et étude fonctionnelle du projet.....	21
Introduction.....	22
I. Pilotage du projet.....	22
A. Méthode SCRUM.....	22
B. Déroulement du projet.....	23
II. Analyse fonctionnelle : .....	24
A. Choix d'UML.....	24
B. Analyse et conception : .....	25
1. Description des acteurs : .....	25
2. Diagramme de cas d'utilisation : .....	26
3. Fiches techniques : .....	27
4. Diagramme de classes : .....	37
Conclusion.....	40
Chapitre 3 : Mise en place de l'application.....	41



Introduction.....	42
I. Environnement de réalisation .....	42
1. Java EE .....	42
2. Apache TomCat .....	43
3. Eclipse.....	43
4. Entreprise architecte .....	43
II. Langages et technologies utilisées .....	43
A. Architecture Java EE .....	44
B. Technologies utilisées.....	45
1. La couche présentation .....	45
2. La couche traitement .....	46
3. La couche données.....	46
III. Présentation de l'application .....	47
1. L'interface « Authentification ».....	47
2. « Demande Carburant » .....	48
3. « Demande de réparation » .....	51
4. « Paramétrage » .....	54
Conclusion générale .....	58
Bibliographie.....	59
Webographie.....	59
Annexes .....	60



## Liste des figures

Figure 1 organigramme ministère	10
Figure 2 organigramme DSI	11
Figure 3 : affectation	14
Figure 4:processus demande de carburant	15
Figure 5 : processus de consommation des VR	16
Figure 6 : les ordres de missions	17
Figure 7 : use case personne	26
Figure 8 : use case : employé	26
Figure 9 : use case superviseur	27
Figure 10:le processus d'affectation	28
Figure 11 : diagramme d'activité de : demande réparation	30
Figure 12 : diagramme d'activité de : demande carburant	32
Figure 13:diagramme d'activité de : paramétrage d'utilisateur	34
Figure 14 : diagramme d'activité de : valider bon de carburant/demande de réparation	36
Figure 15 : diagramme de la classe carnet	37
Figure 16 : diagramme de la classe vignette	38
Figure 17 : diagramme de la classe chauffeur	38
Figure 18 : diagramme de la classe demande carburant	39
Figure 19 : diagramme de la classe réparation	39
Figure 20 : diagramme de la classe demande réparation	40
Figure 21: Architecture JAVA2EE	44
Figure 22 : Interface authentification	47
Figure 23 : entête de l'application pour le profil employé	48
Figure 24 IHM demande carburant liste	48
Figure 25 : IHM ajouter/modifier demande carburant	49
Figure 26 : IHM attribuer les vignettes	50
Figure 27 : IHM règlement de facture	51
Figure 28: IHM demandes de réparation liste	52
Figure 29: : IHM ajouter/modifier demande de réparation	53
Figure 30:IHM ajouter/modifier bon de commande	54
Figure 31 : IHM liste des véhicules	55
Figure 32: IHM ajouter un véhicule	55
Figure 33: liste/ajouter une affectation entité	56
Figure 34 : Liste des chauffeurs	56
Figure 35 : pop-up Ajouter chauffeur	57
Figure 36 : pop-up affectation chauffeur	57
Figure 37: message informatif de suppression	57
Figure 38 : véhicule de fonction	63
Figure 39 : véhicule de mission	63
Figure 40 : véhicule utilitaire	63
Figure 41: la classe chauffeur	71
Figure 42: le champ véhicule	72



## Liste des tableaux

<i>Tableau 1 : les acteurs intervenant dans le projet</i>	23
<i>Tableau 2 : rédiger une demande de réparation</i>	29
<i>Tableau 3 : rédiger un bon de carburant</i>	31
<i>Tableau 4 : paramétrage des utilisateurs</i>	33
<i>Tableau 5 : valider bon</i>	35
<i>Tableau 6 : environnement de réalisation</i>	42
<i>Tableau 7 technologies utilisées dans la couche présentation</i>	45
<i>Tableau 8 technologies utilisées dans la couche traitement</i>	46
<i>Tableau 9 technologies utilisées dans la couche données</i>	47



## Liste des abréviations

<b>Abréviation</b>	<b>Désignation</b>
SI	Système d'Informations
DSI	Direction des Systèmes d'Informations
SNTL	Société National du Transport et du Logistique
VTT	Vignette de Transport Terrestre
VTA	Vignette de transport Aérien
IDE	Integrated Development Environment
API	Application Programming Interface
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
IHM	Interface Homme Machine
JEE	Java Entreprise Edition
JSF	Java Server Faces
SGBD	Système de Gestion des Bases de Données
UML	Unified Modeling Language



## Introduction générale

Dans le cadre de notre formation universitaire à la faculté des sciences et techniques de Fès en licence génie informatique. Nous sommes amenées à réaliser un projet de fin d'études en deux mois afin de concrétiser et raffiner nos connaissances acquises à travers notre parcours universitaire et les mettre en œuvre dans un milieu professionnel.

Note stage s'est déroulé au sein de la direction des systèmes d'informations du ministère de la réforme de l'administration et de la fonction publique du 03 avril au 02 juin 2017. Durant cette période, il nous a été confié de réaliser une application de gestion de parc automobile des administrations publiques destinée à la mutualisation en utilisant java EE comme langage de développement et Microsoft SQL server comme système de gestion de base de données.

Pour bien mener le projet, l'équipe de développement choisit de suivre une des méthodes agiles, à savoir SCRUM, qui a fait ses preuves dans le domaine des projets informatiques et qui est plus adaptée aux équipes réduites avec des besoins changeants.

Ce rapport repose sur trois principaux chapitres :

Le premier chapitre présente le cadre général du projet en décrivant l'organisme d'accueil, la problématique du sujet, et le pilotage du projet.

Le deuxième chapitre se concentre sur l'analyse fonctionnelle du projet, la conception et les différents diagrammes d'UML utilisés.

Le troisième chapitre décrit l'environnement technique du projet, les outils et les langages de programmation et de modélisation utilisés dans le cadre de ce projet.

Et nous terminerons ce rapport par une conclusion générale et les perspectives du travail.



---

# Chapitre 1 : **Contexte général du projet**



## Introduction

Ce chapitre décrit l'environnement dans lequel le projet a été initié et ce afin d'avoir une meilleure compréhension de ce dernier. Pour se faire, nous allons présenter dans un premier temps l'organisme d'accueil, son organigramme, ses objectifs, ses missions, et sa vision. Ensuite nous décrivons le projet, sa nécessité et les résultats attendus de ce travail en faisant un critique de l'existant.

### I. Organisme d'accueil

#### 1. Ministère :

##### a) *Missions principales*

- Participer au développement de l'administration numérique.
- Impulser, proposer et mener, en concertation avec les départements ministériels, toute action de simplification des procédures administratives ;
- Veiller à la rationalisation des structures administratives et proposer des mesures en vue d'en améliorer le fonctionnement ;
- Contribuer à l'amélioration de la qualité des services publics rendus aux usagers ;
- Promouvoir la transparence dans la gestion publique ;
- Constituer et gérer les bases de données relatives aux personnels de l'Etat ;
- Réaliser toutes études, enquêtes et statistiques relatives à la fonction publique ;
- ...



b) Organigramme

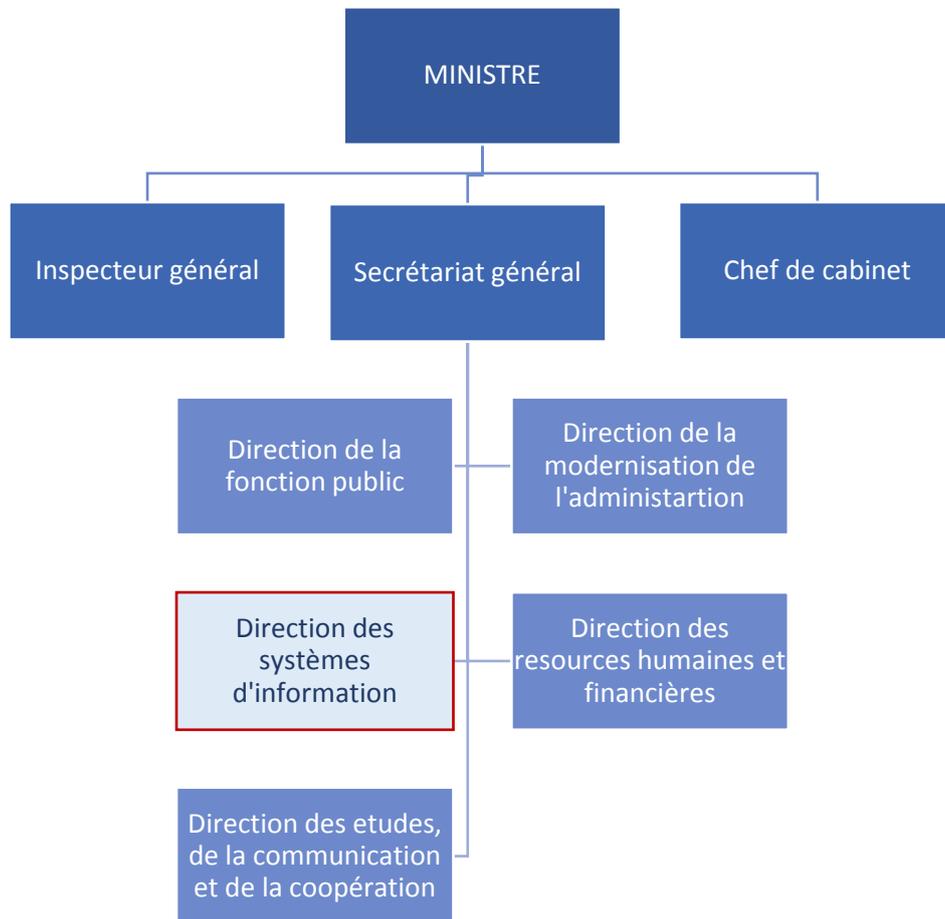


Figure 1 organigramme ministère



## 2. Direction des systèmes d'information :

### a) Organigramme du DSI

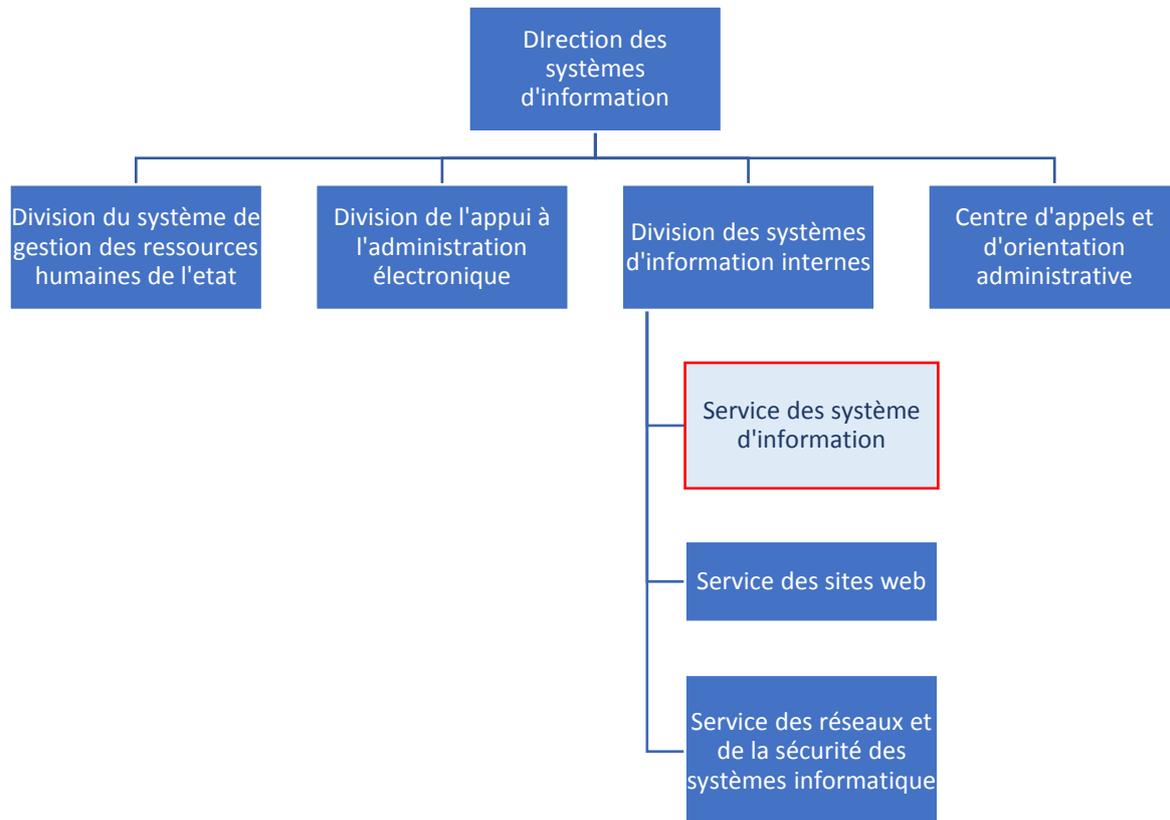


Figure 2 organigramme DSI

### b) Service des systèmes d'informations

#### 1. Mission principale

Développer le système d'information du ministère : Concevoir, développer et mettre en œuvre les systèmes d'information du Ministère tant au plan de la gestion administrative qu'au plan du cœur de métier.

#### 2. Principales activités et tâches /Fonctions

##### *Développement et maintenance des SI métier et gestion*

- Recensement des besoins
- Développement ou acquisition des applications
- Maintenance des applications
- Élaboration des termes de référence



- Suivi de projets

#### *Exploitation des SI*

- Assistance et formations des utilisateurs
- Suivi des applications
- Suivi des systèmes
- Maintenance et mise à jour des données

#### *Base de données*

- Backup et sauvegarde des bases de données
- Administration des Bases de données
- Migrations des Bases de données

#### *Mutualisation des applications de Gestion*

- Echanger des produits ou/et des expériences en matière de systèmes d'information en les Administrations.
- Mettre en commun des compétences humaines pour la réalisation de produits au profit de l'Administration ;
- Consolider les efforts de l'Administration pour la réalisation de solutions en commun dans le domaine des systèmes d'information

## II. Présentation du projet

### A. Problématique

#### 1. Mise en position

Le problème de gestion de parc auto est commun entre la majorité des organismes, et puisque cette application sera mutualisée, nous avons choisi de concaténer les besoins pour mieux les gérer en prenant comme exemple, le parc auto du ministère de la fonction public et de la modernisation de l'administration.

Ce ministère dispose d'une application de gestion du parc automobile, cette dernière a été mise en place pour la première fois pour gérer quelques simples



opérations selon certains besoins, par exemple gérer les vignettes<sup>1</sup> consommées, affecter un véhicule à une entité<sup>2</sup>, gérer des ordres de missions<sup>3</sup>...

L'application n'avait pas de conception claire suivie dès le début, donc à chaque fois qu'un besoin apparaissait, ils l'intégraient dans l'application sans prendre en considération les problèmes qui peuvent être générés suite aux fautes de conception ; comme la redondance ou même la génération des bugs.

L'application est destinée à gérer les différents composants du parc automobile tout en assurant les règles et la logique d'organisme.

## 2. Description du métier

### a) Réception des vignettes

Au début de chaque année l'organisme intérieure indique le montant du carburant et de réparation prévue, puis rédige une demande, cette dernière est envoyée à la SNTL<sup>4</sup> (avec les vignettes restantes de l'année dernière au cas du carburant), l'organisme attend donc une réponse de la part de la SNTL.

Lorsqu'elle reçoit une demande, la SNTL livre 1/3 de la somme demandée à l'organisme.

### b) Gestion des affectations

Le ministère est composé de plusieurs entités (secrétariat générale<sup>5</sup>, CABINET<sup>6</sup>...), tous les véhicules s'affectent à l'une de ces entités. À un moment donné un véhicule (voiture ou motorcycle) ne peut être affecté qu'à un seul chauffeur alors qu'un chauffeur peut disposer de plusieurs véhicules sous sa responsabilité.

L'ensemble des véhicules est divisé en trois catégories<sup>7</sup> :

---

<sup>1</sup> Vignette : des timbres remplacent l'argent utilisées pour le paiement des différentes dépenses. Voir exemple en annexe. (N° 1-1).

<sup>2</sup> Entité : une partition du ministère.

<sup>3</sup> Voir annexe N° 1

<sup>4</sup> SNTL : Société National du Transport et du Logistique.

<sup>5</sup> Secrétariat général : est le nom donné au leader de divers syndicats, partis, associations, ministères ou autres organismes.

<sup>6</sup> Cabinet : Cabinet du ministre comprend le ministre et ses plus proches collaborateurs et conseillers. Il est considéré comme une des institutions les plus puissantes du ministère.

<sup>7</sup> Voir annexe N° 2



1. **Véhicules de fonction** : affectés au ministre, secrétaire général et au président du cabinet...
2. **Véhicules de mission** : affectés pour accomplir une mission limitée dans le temps et le trajet
3. **Véhicules utilitaire** : destinés au transport des personnels et aux biens de l'administration

Pour des raisons de flexibilité, il est toujours possible de changer l'affectation d'un véhicule ou du chauffeur, avant cela, l'employé doit renseigner une décharge<sup>8</sup>. Le nouveau système doit générer automatiquement un document de décharge et donner la possibilité de l'imprimer à chaque moment.

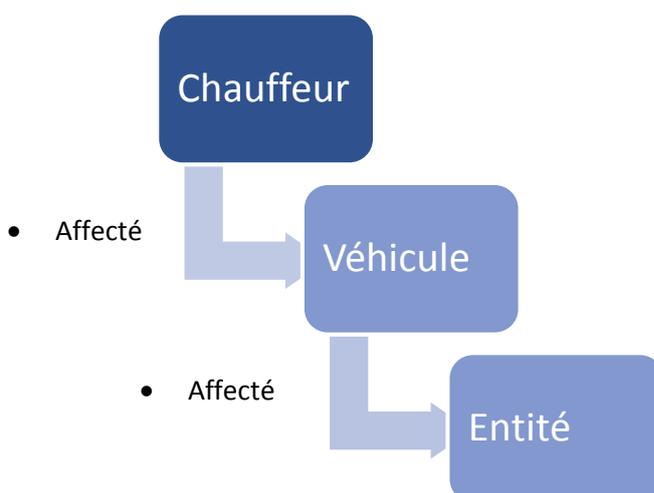


Figure 3 : affectation

### c) Gestion des vignettes de carburant

Le chauffeur se présente à l'entité pour obtenir ses vignettes de carburant. Avant qu'il soit servi, il doit signaler dans un bon de commande<sup>9</sup> signé : le kilométrage et le montant voulu avec une justification au cas où il dépasse le seuil quotidien.

<sup>8</sup> Voir annexe N° 3

<sup>9</sup> Voir annexe N° 4



L'entité livre les bons à l'administrateur pour qu'il les valide (bon en attente de validation), la validation se fait en vérifiant la souche<sup>10</sup>.

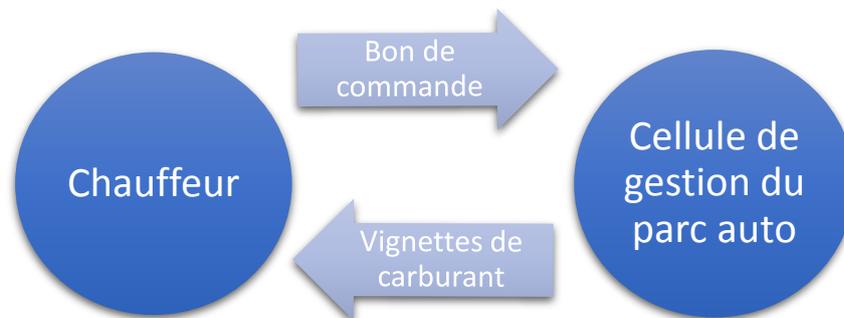


Figure 4: processus demande de carburant

#### d) Gestion des vignettes de réparation

Lors d'une panne le chauffeur doit consulter trois réparateurs différents pour estimer le montant total de réparation, afin de choisir un réparateur. Chacun d'eux rédige un Devi, l'employé choisit le réparateur le moins disant et lui envoie une demande de réparation (DR)<sup>11</sup>, une fois la réparation est complète le chauffeur doit signer un bon de livraison (BL)<sup>12</sup> rédigé par le réparateur avant de récupérer la facture et le véhicule.

La facture, DR, et BL sont envoyés à l'employé pour qu'il enregistre ces données dans le système et donne les vignettes au réparateur.

Au cas de grave sinistre (ou si le véhicule dépasse 10 ans de service) le véhicule sera réformé (envoyé à la SNTL).

<sup>10</sup> Souche : support de paiement sur lequel on colle les vignettes pour assurer la traçabilité des dépenses.

<sup>11</sup> Voir annexe N°5

<sup>12</sup> Voir annexe N°6

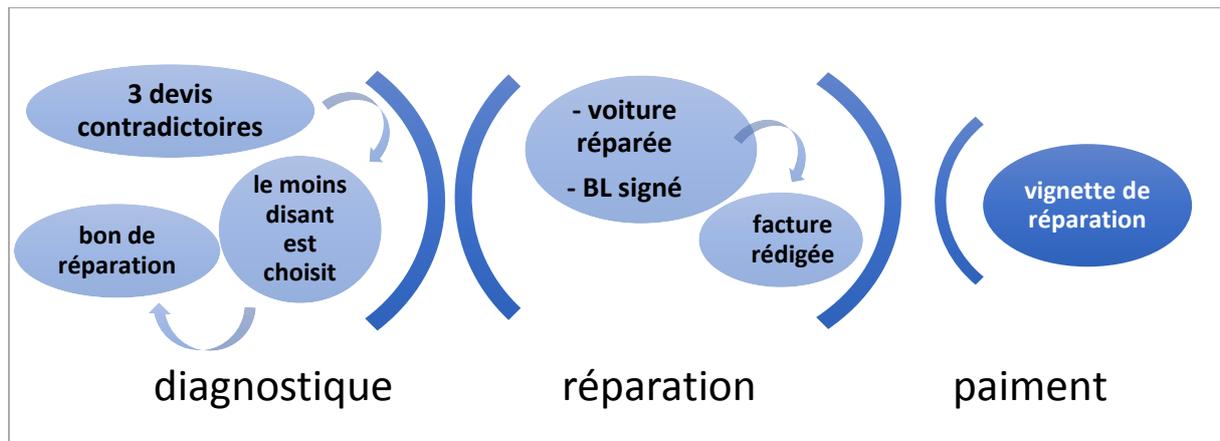


Figure 5 : processus de consommation des VR

e) *Gestion des vignettes de transport terrestre*

Ce type de vignettes est donné lors des missions ou des activités externes pour couvrir les dépenses quel que soit la situation<sup>13</sup>, les VTTs peuvent s'échanger en billet de train, CTM, Supratour, des cartes d'autoroutes, ou des dispositifs jawaz...

f) *Gestion des ordres de mission*

Quand il s'agit d'une mission, l'employé renseigne un ordre de mission dans lequel il indique le(s) véhicule(s) à utiliser, le(s) chauffeur(s) concerné(s), le trajet, et le montant de carburant à consommer, cet ordre est transféré au superviseur pour qu'il soit validé puis le chauffeur(s) pourra prendre les vignettes de carburant nécessaire (processus de demande de carburant).

Après l'accomplissement de la mission, le chauffeur remet les vignettes restantes, l'ordre de mission original sera modifié pour assurer la cohérence du système.

<sup>13</sup> Voir annexe N°7

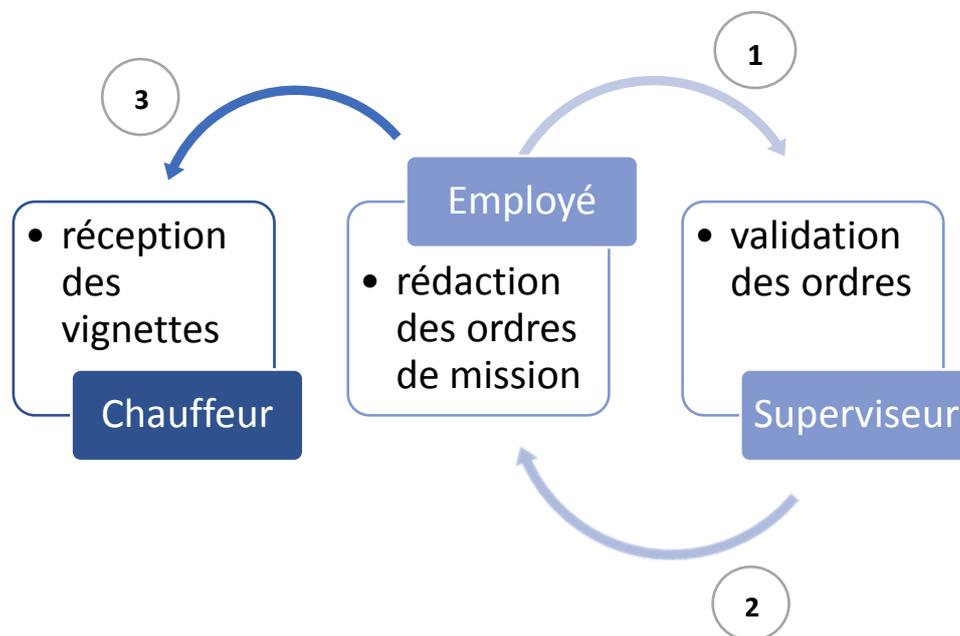


Figure 6 : les ordres de missions

## B. Critique de l'existant

Après l'utilisation de l'application existante de « gestion du parc auto » nous avons détecté les problèmes suivants : vue que l'application a été créée de manière progressive répondant instantanément aux besoins qui apparaissent, elle manquait d'analyse de domaine complète et donc il y avait des informations redondantes mal hiérarchisées, ce qui rendait la recherche d'une simple information une tâche difficile. De plus cette application manque de plusieurs fonctionnalités ergonomiques nécessaires pour la cellule de gestion de parc auto.

### Manque d'assistance :

Absence des info-bulles, pas de guide d'utilisation, manque de présentation de l'application et ses fonctionnalités, utilisations des titres insignifiants ...

Nous citons comme exemple, la page d'accueil qui apparaît après l'authentification, elle contient un tableau sous le titre de : « situation générale des vignettes ». Ceci donne l'impression que l'application ne fait que la gestion des vignettes et pas d'autre chose, ce qui n'est pas le cas.



## Les informations inutiles :

D'autre part l'application est surchargé des listages inutiles par exemple : lister les marques, les entités, les classes, ... sans les mettre dans un contexte qui pourra être utile.

### C. Solution proposée

Pour combler ces problèmes, nous avons proposé le développement d'une application pour la gestion du parc auto en se basant sur : Java EE et Microsoft SQL server comme système de gestion de base de données et en utilisant SCRUM comme méthodologie de pilotage du projet.

L'application aura comme but :

- ✓ Construire une architecture du logiciel de manière rationnelle, solide qui minimise le coût de la maintenance.
- ✓ Gérer toutes les données du métier, et assurer par la suite le suivi et la traçabilité des données tout en respectant les rôles des acteurs par responsabilités.
- ✓ Adapter le produit aux besoins le maximum possible.
- ✓ Implémenter toutes les fonctionnalités du métier : l'attribution des vignettes, les règlements des demandes par les souches.
- ✓ Travailler sur l'ergonomie de l'application

### D. Cahier des charges

Après avoir eu des discussions avec le responsable du parc auto et quelques fonctionnaires de cette cellule, et après une étude détaillée du système, nous avons pu élaborer le cahier de charge suivant. Les exigences sont structurées par thèmes :

#### **Gestion des utilisateurs :**

- ✓ Le système doit permettre à l'admin de créer un compte pour les autres employés.
- ✓ Le système doit permettre à l'admin de supprimer le compte d'un employé ou de régénérer un mot de passe en cas de son oubli.



- ✓ Le système doit permettre à tous les utilisateurs de gérer leurs profils : modification des informations, modification du login/mot de passe.
- ✓ Le système doit être sécurisé : chaque utilisateur est authentifié par un login et un mot de passe.

#### **Gestion des vignettes :**

- ✓ Le système doit permettre à l'employé de renseigner les informations sur une convention avec ses éléments. Le système doit donner ainsi la possibilité de consulter les informations des conventions des années précédentes. Avec la possibilité d'effectuer des recherches selon plusieurs critères.
- ✓ Le système doit permettre à l'administrateur/employé de renseigner les informations des carnets de vignettes avec leurs éléments. Le système doit donner ainsi la possibilité de lister tous les carnets avec la possibilité d'effectuer des recherches selon plusieurs critères.
- ✓ De la même façon, le système doit permettre à l'administrateur/employé d'initialiser les informations des vignettes. Il faut noter que la liste des vignettes est renouvelée chaque année.

#### **Gestion des dépenses :**

- ✓ Le système doit permettre à l'employé d'initier, d'enregistrer et de modifier les informations sur les demandes de carburant ou réparation avec la possibilité d'effectuer des recherches selon plusieurs critères (véhicule, chauffeur, etc.)
- ✓ Le système doit donner la possibilité à l'employé de supprimer une demande afin qu'ils n'apparaissent plus lors de la consultation.
- ✓ Le système doit permettre de lister toutes les demandes avec la possibilité d'effectuer des recherches selon plusieurs critères, aussi d'afficher les détails des éléments d'une demande spécifique.

#### **Gestion de parc :**

- ✓ Le système doit permettre à l'employé d'ajouter des nouveaux chauffeurs à la base de données.
- ✓ Le système doit permettre à l'employé d'ajouter des nouveaux véhicules à la base de données.



- ✓ Le système doit permettre à l'employé de supprimer/modifier les informations des véhicules et chauffeurs.
- ✓ Le système doit permettre l'affichage de tous les véhicules actifs ou réformés.
- ✓ Le système doit permettre l'affichage de tous les chauffeurs actifs ou en congé.

#### **Gestion des affectations :**

- ✓ Le système doit permettre à l'employé d'affecter un véhicule à un chauffeur.
- ✓ Le système doit permettre à l'employé d'affecter un véhicule à une entité.
- ✓ Le système doit permettre à l'employé de modifier les informations d'une affectation avec la possibilité d'effectuer des recherches selon plusieurs critères (véhicule, chauffeur, dates, etc.)
- ✓ Le système doit permettre de lister toutes les affectations avec la possibilité d'afficher les détails des éléments d'une affectation spécifique.

## Conclusion

Ce chapitre commence par une présentation de l'organisme d'accueil, ensuite il aborde le contexte et les objectifs du projet vis-à-vis l'application existante. Le chapitre suivant est consacré à l'étude fonctionnelle du projet, le processus de développement SCRUM adopté pour le développement de la solution, et le planning de travail du projet.



---

## Chapitre 2 : **Méthodologie du développement et étude fonctionnelle du projet**



## Introduction

Le présent chapitre a comme objectif tout d'abord de présenter la façon dont nous avons piloté le projet, le processus de développement adopté et l'analyse des besoins fonctionnels de chaque module en identifiant l'ensemble des acteurs du système. Des diagrammes de cas d'utilisation sont présentés donnant ainsi une vision sur le fonctionnement du système vis-à-vis de l'utilisateur.

## I. Pilotage du projet

Il existe différents types de cycles de développement entrant dans la réalisation d'un logiciel. Ces cycles prennent en compte toutes les étapes de la conception et suivent un processus de développement bien défini, ce processus a pour objectifs de produire des solutions informatiques de qualité répondantes aux besoins des utilisateurs en des temps et des coûts prévisibles. De ce fait, l'adéquation du projet au processus de développement peut largement affecter le sort d'un projet informatique.

Dans le souci de répondre à ces objectifs, et afin de privilégier l'interaction humaine en visant la satisfaction réelle des utilisateurs finaux, l'équipe du projet a choisi d'adopter un cycle de développement itératif incrémental, c'est la méthode agile SCRUM, s'adaptant le mieux possible à ce projet.

### A. Méthode SCRUM

SCRUM <sup>[1w]</sup> est l'une des méthodes agiles<sup>14</sup> les plus utilisées, elle est une méthode itérative incrémentale. La notion d'incrément facilite sensiblement le contrôle de pilotage du projet alors que la notion d'itération (Méthode itérative) assure l'adaptabilité et la qualité fonctionnelle ou technique. Ces deux points représentent les principes fondamentaux d'une Méthode agile appliquée à la complexité de l'Ingénierie du logiciel et qui s'adaptent parfaitement aux besoins.

Le principe de base de SCRUM est de focaliser l'équipe sur une partie limitée et maîtrisable des fonctionnalités à réaliser. Ces incréments se réalisent successivement lors de périodes de durée s'étendant entre une et quatre semaines, appelées sprints. Chaque sprint possède, préalablement à son exécution, un but à atteindre, défini par le directeur de produit « Product

---

<sup>14</sup> Agile : <sup>[2w]</sup> Le terme "agile" définit une approche de gestion de projet qui prend le contre-pied des approches traditionnelles prédictives et séquentielles



Owner », à partir duquel sont choisies les fonctionnalités à implémenter dans cet incrément. Un sprint aboutit toujours à la livraison d'un produit partiel fonctionnel. Pendant ce temps, le chef de l'équipe de développement « Scrum Master » a la charge de minimiser les perturbations extérieures et de résoudre les problèmes non techniques de l'équipe.

## B. Déroulement du projet

### a) L'équipe du projet

Selon les principes de SCRUM, les rôles des acteurs intervenant se limite dans :

- **Le Product Owner** (directeur de produit) : il représente les utilisateurs, définit et priorise les demandes du produit.
- **Le Scrum master**, qui n'est pas le chef de projet mais qui a un rôle de coach : sa mission est de tout mettre en œuvre pour que l'équipe travaille dans de bonnes conditions et se concentre sur l'objectif du projet.
- **L'équipe SCRUM** : avec moins de 5 personnes regroupant tous les rôles traditionnels (architecte, développeur, testeur, administrateur). Cette équipe développe le produit et se gère en toute autonomie.

Rôle	Acteurs intervenant
<b>Product Owner</b>	Mr M. BOUKHANI chef de service des systèmes d'informations
<b>Scrum master</b>	Mr E. SANDADI ingénieur de développement
<b>L'équipe SCRUM</b>	Mlle R. EL HAMSS Mlle C. EL GADDAR

Tableau 1 : les acteurs intervenant dans le projet

### b) Le Backlog de produit

Le Product Backlog ou le Backlog de produit est le cœur de SCRUM, il est géré par le Product Owner, et partagé avec l'ensemble du groupe. Les besoins sont priorisés et l'accent est mis sur les besoins les plus prioritaires et devant être adressés à court terme. Ils sont associés à des critères qui permettent de définir, sans ambiguïté, les critères qui permettront de considérer que le besoin est couvert. Le Backlog de produit est élaboré avant le lancement des sprints, dans la phase de préparation (ou sprint 0). Il est utilisé pour planifier la release, puis à



chaque sprint, lors de la réunion de planification du sprint pour décider du sous-ensemble qui sera réalisé.

### c) *Les réunions des Sprints*

La planification du projet est une phase importante d'avant-projet. Elle consiste à prévoir le déroulement du projet tout au long des phases constituant le cycle de développement.

La réunion de Sprint se déroule avant chaque nouveau sprint, elle a pour but la planification du sprint à venir, elle est critique pour le bon déroulement de tout le sprint. A l'issue de cette réunion, Toute l'équipe participe à cette réunion pour déterminer :

- le but et l'objectif du sprint.
- une durée du sprint et une date de démonstration.
- les histoires à inclure dans le Sprint qui vont construire le Backlog du Sprint.
- l'estimation des tâches, chaque membre de l'équipe est impliqué dans cette activité, et pour éviter que les membres ne soient pas influencés par une première estimation.
- Le lieu et l'heure de la mêlée quotidienne, elle a comme but le suivi de l'avancement du projet, lors de cette réunion qui dure 15 min maximum chaque membre doit répondre à trois questions :
  - « Qu'est-ce que j'ai fait hier ? »
  - « Qu'est-ce que je compte faire aujourd'hui ? »
  - « Quels sont les problèmes que je rencontre ? »

## II. Analyse fonctionnelle :

### A. Choix d'UML

UML<sup>[1B]</sup> est une méthode de modélisation orientée objet développée en réponse à l'appel à propositions lancé par l'OMG (Object Management Group) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets.

UML est utilisée pour spécifier un logiciel et/ou pour concevoir un logiciel. Dans la spécification, le modèle décrit les classes et les cas d'utilisation vus de



l'utilisateur final du logiciel. Le modèle produit par une conception orientée objet est en général une extension du modèle issu de la spécification. Il enrichit ce dernier de classes, dites techniques, qui n'intéressent pas l'utilisateur final du logiciel mais seulement ses concepteurs. Il comprend les modèles des classes, des états et d'interaction. UML est également utilisée dans les phases terminales du développement avec les modèles de réalisation et de déploiement.

## B. Analyse et conception :

### 1. Description des acteurs :

Quatre utilisateurs de l'application sont identifiés, en effet l'exploitation des données varie d'un utilisateur à un autre, les responsabilités des acteurs diffèrent selon leurs fonctions et leurs rôles.

Les acteurs de l'application sont les suivants :

#### a) *Employé 1 :*

L'employé 1 chargé du renseignement d'une affectation, une décharge ou une réforme, il peut aussi remplir et imprimer des bons de carburant ou de réparation ou un ordre de mission.

#### b) *Employé 2 :*

L'employé 2 est responsable des vignettes de transports VTT (terrestre) et VTA (aérien).

#### c) *Superviseur :*

Le superviseur est chargé de la création des comptes utilisateurs, la validation des bons de commande et réparation, il peut s'occuper aussi de l'ajout, modification, suppression d'un véhicule ou d'un chauffeur.

#### d) *Directeur :*

Il peut seulement consulter les différentes listes



## 2. Diagramme de cas d'utilisation :

Après avoir identifié les acteurs nous avons établi les diagrammes de cas d'utilisation suivants :

En UML, on établit des Diagrammes de Cas d'Utilisation pour répondre à la question suivante « A quoi va servir le logiciel ? ».

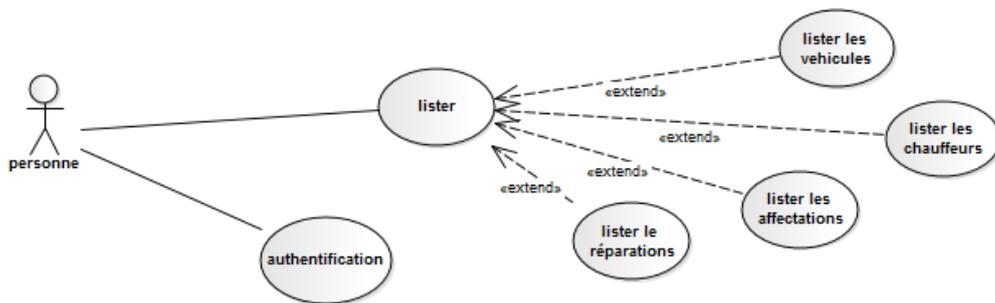


Figure 7 : use case personne

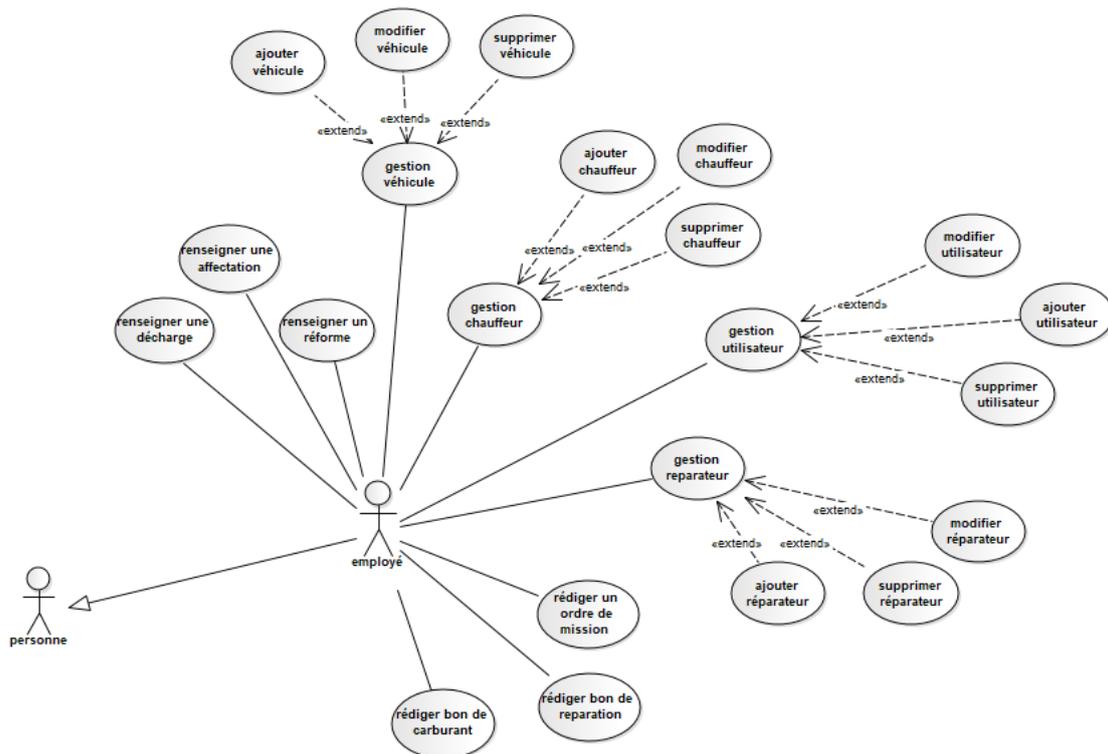


Figure 8 : use case : employé

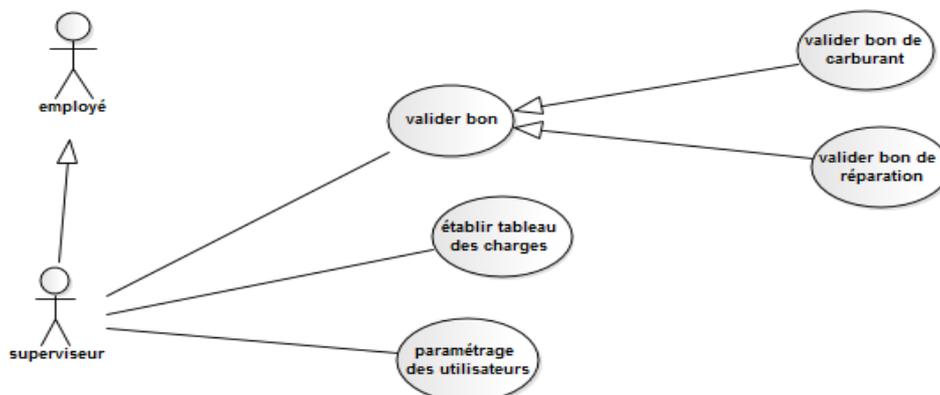


Figure 9 : use case superviseur

### 3. Fiches techniques :

#### a) Employé 1 :

Renseigner une affectation des chauffeurs	
<b>Acteurs</b>	Superviseur-employé
<b>Description</b>	Cette fonctionnalité permet de renseigner et organiser toutes les affectations (chauffeur-véhicule-entité)
<b>L'évènement déclencheur</b>	La prise de décision
<b>Préconditions</b>	<b>Postconditions</b>
La nécessité d'utiliser un véhicule ou un chauffeur	
<b>Scénario nominal</b>	
L'employé choisit un chauffeur parmi les chauffeurs disponibles, Le système charge la liste des véhicules actifs, L'employé remplit les champs nécessaires, L'employé clique sur enregistrer, Le système enregistre l'affectation,	
<b>Scénario alternatif 1</b>	
L'employé choisit un chauffeur parmi les chauffeurs disponibles, Le système charge la liste des véhicules actifs, L'employé clique sur enregistrer, Le système détecte le non remplissage du champ véhicule, Le système affiche une erreur d'enregistrement,	
<b>Scénario alternatif 2</b>	
L'employé choisit un chauffeur de la liste des chauffeurs disponibles, Le système charge la liste des véhicules actifs, L'employé clique sur enregistrer, Le système détecte que le type d'affectation n'est pas choisi, Le système affiche une erreur d'enregistrement,	

Tableau 4 : renseigner une affectation

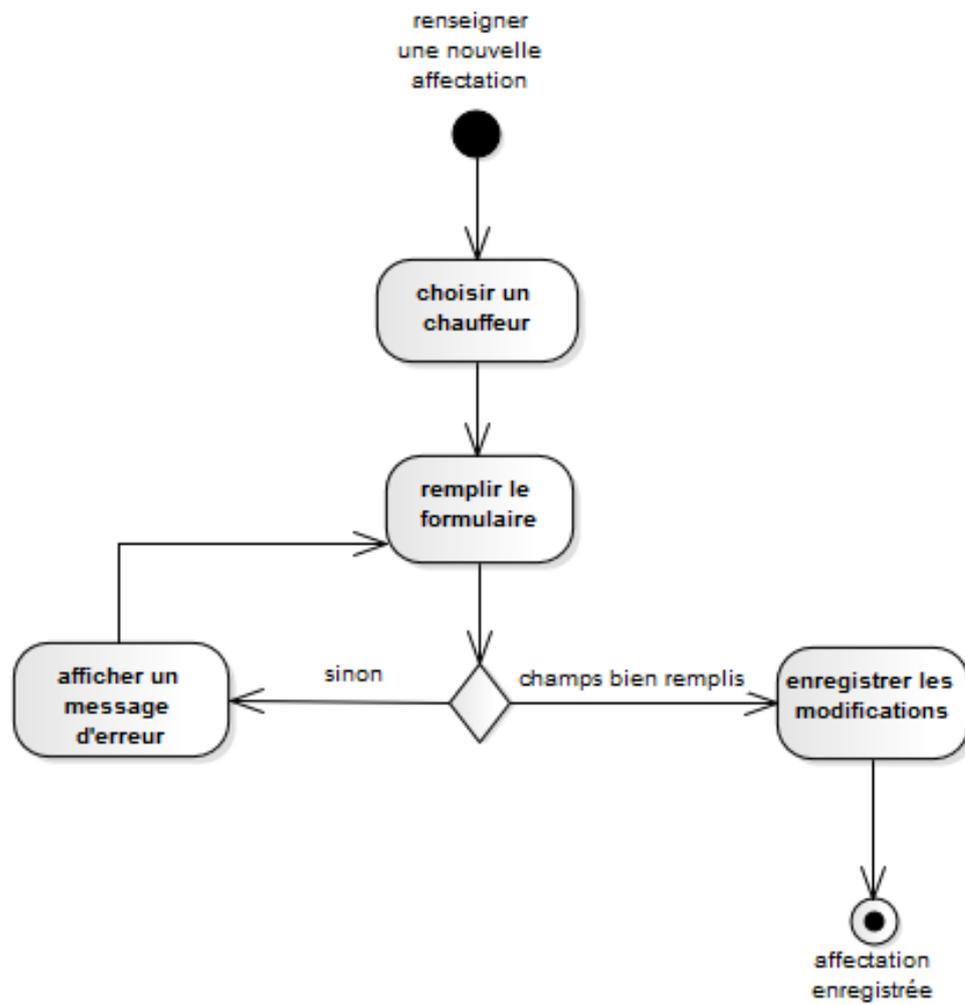


Figure 10:le processus d'affectation



Rédiger demande de réparation	
<b>Acteurs</b>	Employé - chauffeur - réparateur
<b>Description</b>	Sous le besoin de réparer un véhicule, l'employé rédige une demande de réparation et donne au chauffeur les vignettes correspondantes.
<b>L'évènement déclencheur</b>	Le véhicule tombe en panne. Moment de l'entretien quotidien.
<b>Préconditions</b>	<b>Postconditions</b>
Le chauffeur apporte trois devis contradictoires.	Véhicule réparé.
Scénario nominal	
<p>Le chauffeur déclare la réparation d'un véhicule</p> <p>L'employé Indique le véhicule à réparer en renseignant la demande de réparation</p> <p>Le système charges les informations correspondantes à ce véhicules (kilométrage, affectation, et la liste des chauffeurs responsables du véhicule)</p> <p>L'employé choisie le chauffeur, puis renseignes les autres informations</p> <p>L'employé choisie le réparateur le moins disant et remplit les champs nécessaires,</p> <p>Les informations du bon de réparation s'affiche ensuite sur l'écran</p> <p>Après l'enregistrement du bon de réparation, les informations du réception et règlement s'affiche ensuite sur l'écran</p> <p>L'employé enregistre les informations et ajoutes les pièces jointes puis valide</p> <p>Il donne ensuite les vignettes au chauffeur.</p>	
Scénario alternatif 1 :	
<p>Le chauffeur déclare la réparation d'un véhicule</p> <p>L'employé Indique le véhicule à réparer en renseignant la demande de réparation</p> <p>Le système charges les informations correspondantes à ce véhicules (kilométrage, affectation, et la liste des chauffeurs responsables du véhicule)</p> <p>Le système n'affiche pas des chauffeurs (affichage du « parc auto »)</p> <p>L'employé renseigne « parc auto » dans la case chauffeur, puis renseignes les autres informations</p> <p>L'employé choisie le réparateur le moins disant et remplit les champs nécessaires,</p> <p>Les informations du bon de réparation s'affiche ensuite sur l'écran</p> <p>Après l'enregistrement du bon de réparation, les informations du réception et règlement s'affiche ensuite sur l'écran</p> <p>L'employé enregistre les informations et ajoutes les pièces jointes puis valide</p> <p>Il donne ensuite les vignettes au chauffeur.</p>	

Tableau 2 : rédiger une demande de réparation

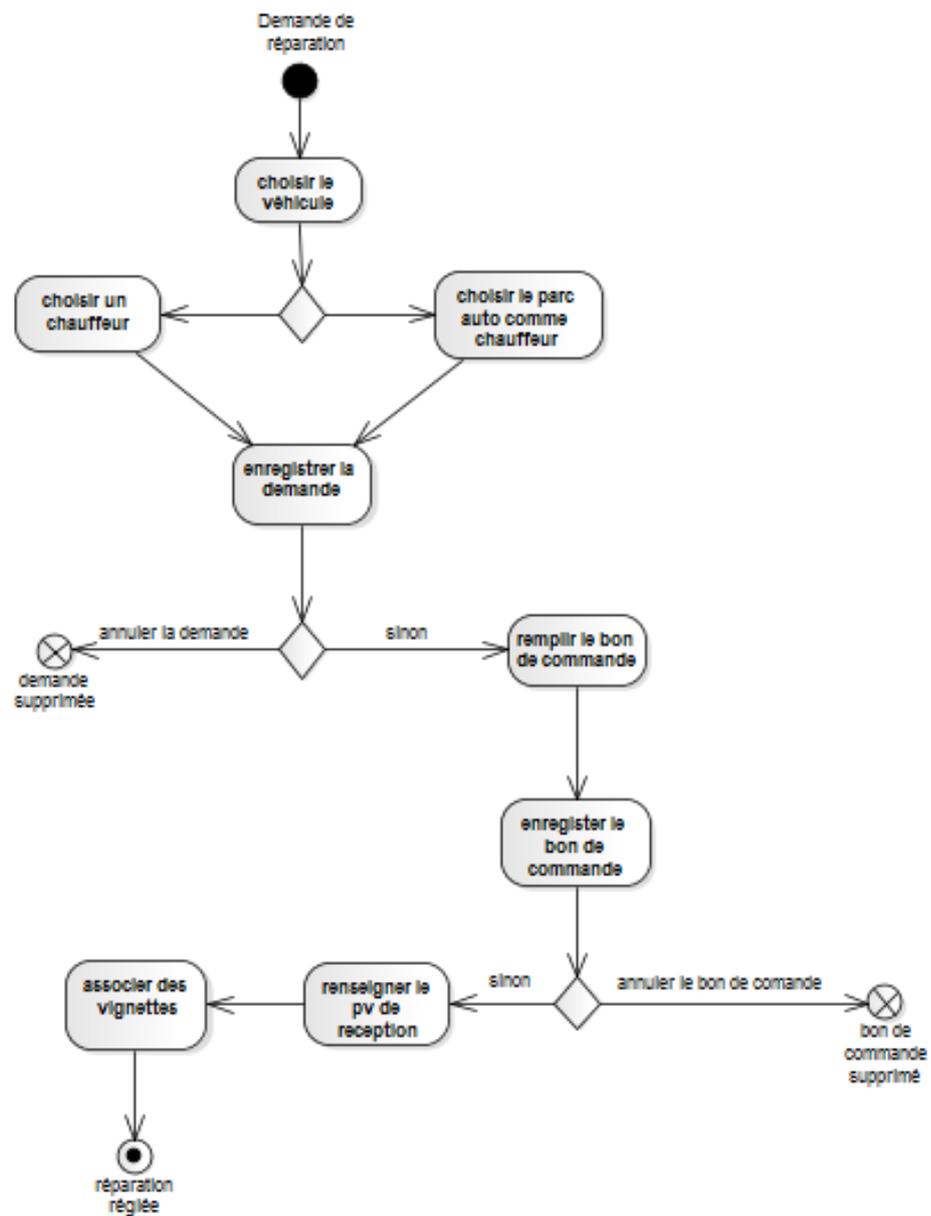


Figure 11 : diagramme d'activité de : demande réparation



Rédiger bon de carburant	
<b>Acteurs</b>	Employé – chauffeur
<b>Description</b>	Le chauffeur demande le carburant et l'employé lui donne les vignettes avec le montant voulu, en remplissant un bon de commande.
<b>L'évènement déclencheur</b>	Le chauffeur se présente à l'entité pour obtenir ses vignettes de carburant.
<b>Préconditions</b>	
Le chauffeur remplit une demande comportant le montant voulu, et le kilométrage vécu par la voiture.	
<b>Postconditions</b>	
<b>Scénario nominal</b>	
<p>Le chauffeur remplit un bon de commande signé, où il indique le montant voulu et le kilométrage.</p> <p>L'employé ouvre l'interface correspondante,</p> <p>Indique que le chauffeur X veut du carburant pour le véhicule Y qui a vécu A km</p> <p>Le système calcul la consommation du véhicule</p> <p>Le taux de consommation s'affiche sur l'écran et un bouton « imprimer »</p> <p>L'employé imprime le bon de carburant</p> <p>Livre les bons à l'administrateur pour qu'il les valide (bon en attente de validation)</p>	
<b>Scénario alternatif 1 :</b>	
<p>Le chauffeur remplit un bon de commande signé, où il indique le montant voulu et le kilométrage.</p> <p>L'employé ouvre l'interface correspondante,</p> <p>Indique que le chauffeur X veut du carburant pour le véhicule Y qui a vécu A km.</p> <p>Le système calcul la consommation du véhicule,</p> <p>Affiche le taux qui a dépassé le seuil normale (affiché en rouge) + un bouton « justifier »</p> <p>Le chauffeur donne une justification à l'employé (ordre de mission).</p> <p>L'employé remplit la case de justification et clic sur « justifier ».</p> <p>Le système affiche le taux de consommation + la justification + un bouton « imprimer ».</p> <p>L'employé imprime le bon de carburant,</p> <p>Livre les bons à l'administrateur pour qu'il les valide (bon en attente de validation)</p>	

Tableau 3 : rédiger un bon de carburant

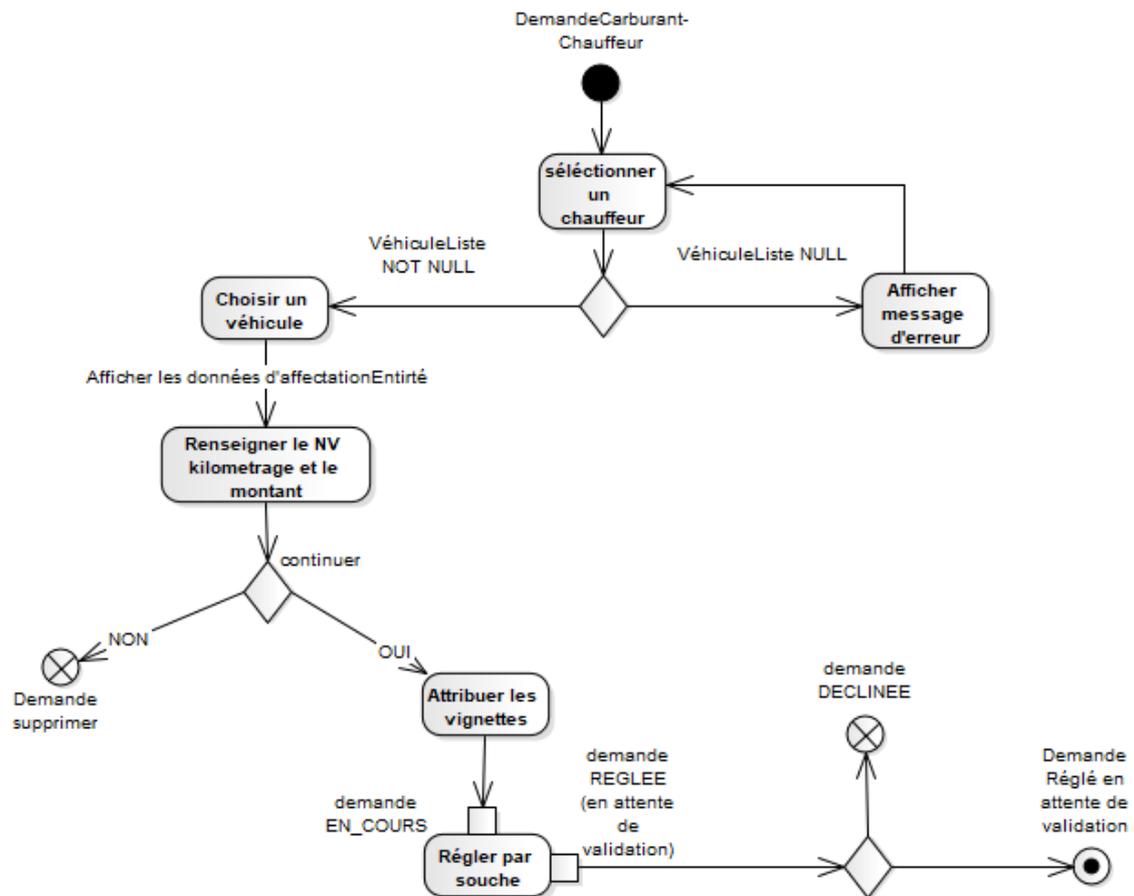


Figure 12 : diagramme d'activité de : demande carburant



b) *Superviseur :*

Paramétrages des utilisateurs	
Acteurs	Superviseur
Description	Le superviseur crée un compte utilisateur de l'application.
L'évènement déclencheur	La prise de décision.
Préconditions	
L'utilisateur est identifié, authentifié et habilité à paramétrer un utilisateur.	
Postconditions	
Scénario nominal :	
<b>Créer un utilisateur :</b> Saisir les informations de l'utilisateur et lui affecter des qualifications. Le système contrôle l'unicité du login. Le système enregistre l'utilisateur.	
<b>Modifier un utilisateur :</b> Sélectionner l'utilisateur à modifier dans la liste des utilisateurs. Modifier les informations de l'utilisateur. Le système contrôle l'unicité du login. Le système enregistre les modifications.	
Scénario alternatif 1 :	
<b>Créer un utilisateur :</b> Saisir les informations de l'utilisateur et lui affecter des qualifications. Le système contrôle l'unicité du login. Le système détecte une erreur et affiche le message suivant : « login déjà utilisé, veuillez saisir un autre ».	
<b>Modifier un utilisateur :</b> Sélectionner l'utilisateur à modifier dans la liste des utilisateurs. Modifier les informations de l'utilisateur. Le système détecte une erreur et affiche le message suivant : « veuillez remplir tous les champs ».	

Tableau 4 : paramétrage des utilisateurs

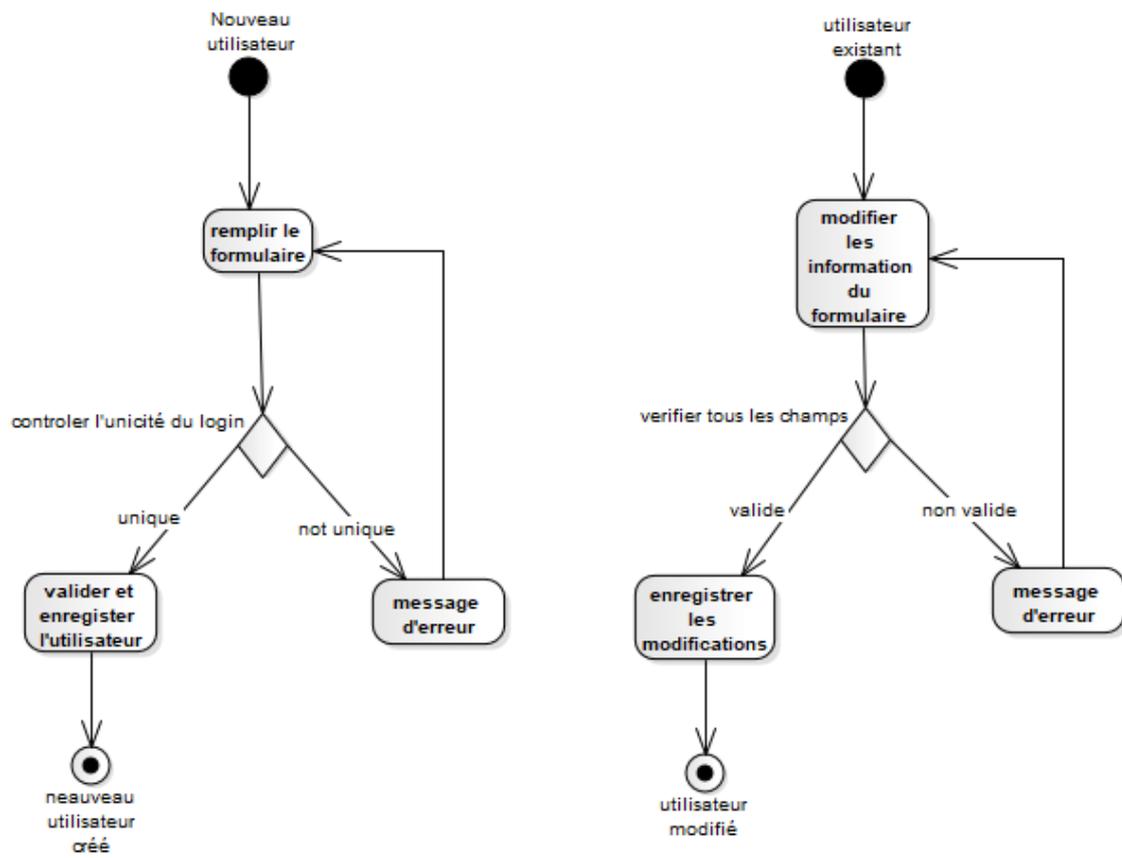


Figure 13:diagramme d'activité de : paramétrage d'utilisateur



Valider bon	
<b>Acteurs</b>	Superviseur
<b>Description</b>	Après avoir renseigné les bons de commande et de réparations, l'employé doit les envoyer au superviseur pour qu'il les valide.
<b>L'évènement déclencheur</b>	Notification du système déclarant la réalisation du bon.
<b>Préconditions</b>	<b>Postconditions</b>
Bon réalisé	
Scénario nominal	
<b>Valider bon de carburant :</b> Le superviseur reçoit un bon de commande Il vérifie la consommation du véhicule Il valide le bon Le système enregistre la validation	
<b>Valider bon de réparation :</b> Le superviseur reçoit un bon de réparation Il vérifie le type de panne et la date de la dernière consultation concernant la même panne Il valide le bon Le système enregistre la validation	
Scénario alternatif 1 :	
<b>Valider bon de carburant :</b> Le superviseur reçoit un bon de commande Il vérifie la consommation du véhicule Il détecte une incohérence entre la consommation du véhicule et le montant de carburant pris Il justifie cette incohérence Il valide le bon Le système enregistre la validation	
<b>Valider bon de réparation :</b> Le superviseur reçoit un bon de réparation Il vérifie le type de panne et la date de la dernière consultation concernant la même panne Il détecte une incohérence entre les données Il ne valide pas le bon	

Tableau 5 : valider bon

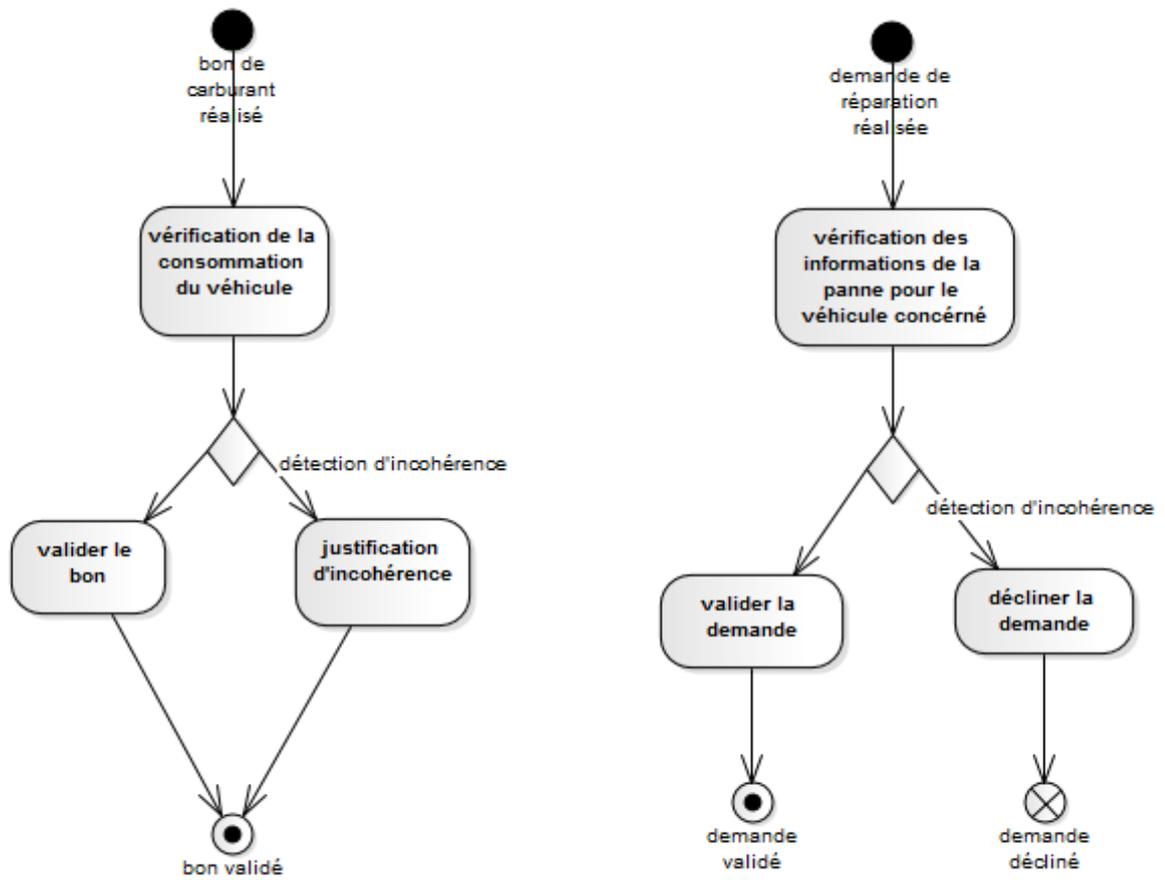


Figure 14 : diagramme d'activité de : valider bon de carburant/demande de réparation



#### 4. Diagramme de classes :

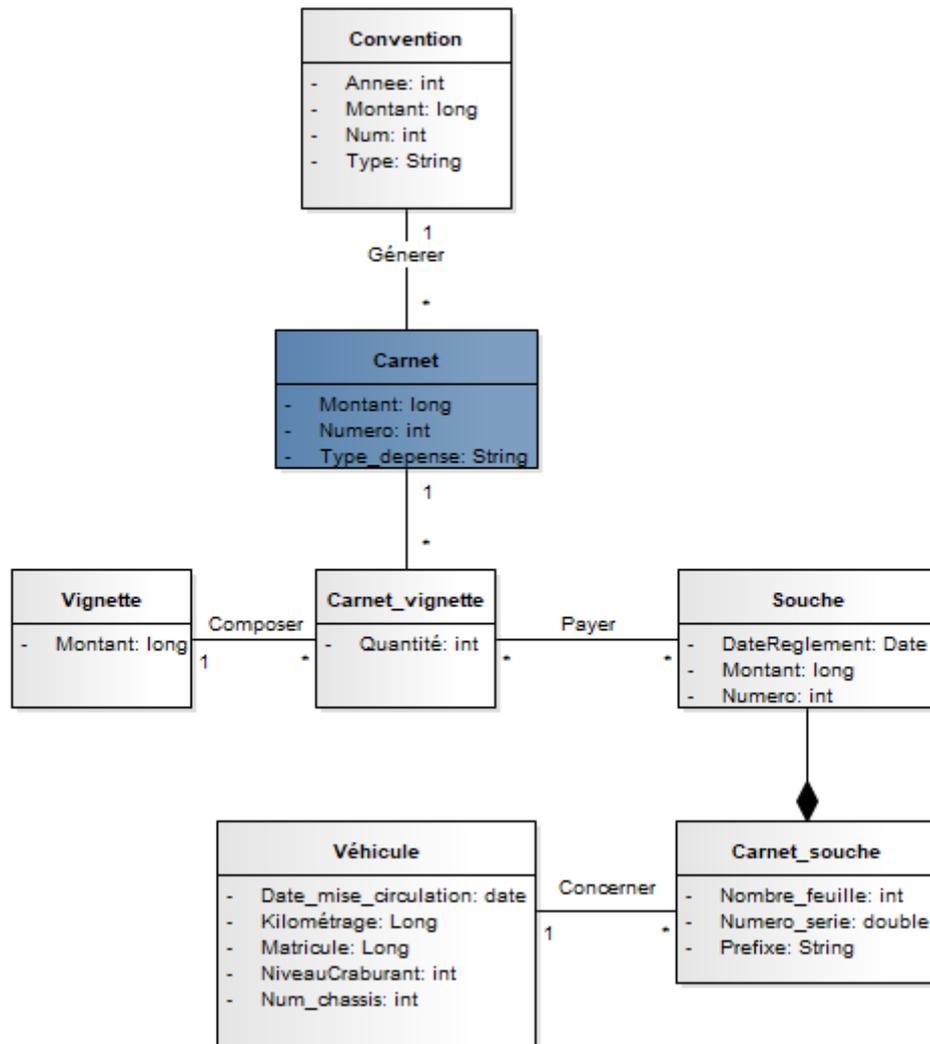


Figure 15 : diagramme de la classe carnet

La validation de la **convention** génère un ou plusieurs **carnets**, chacun d'eux est composé des **vignettes** (*carnet-vignette* est une classe association) ; qu'on les colle par la suite à des **souches** provenant du **carnet-souche** qui est propre au **véhicule**.

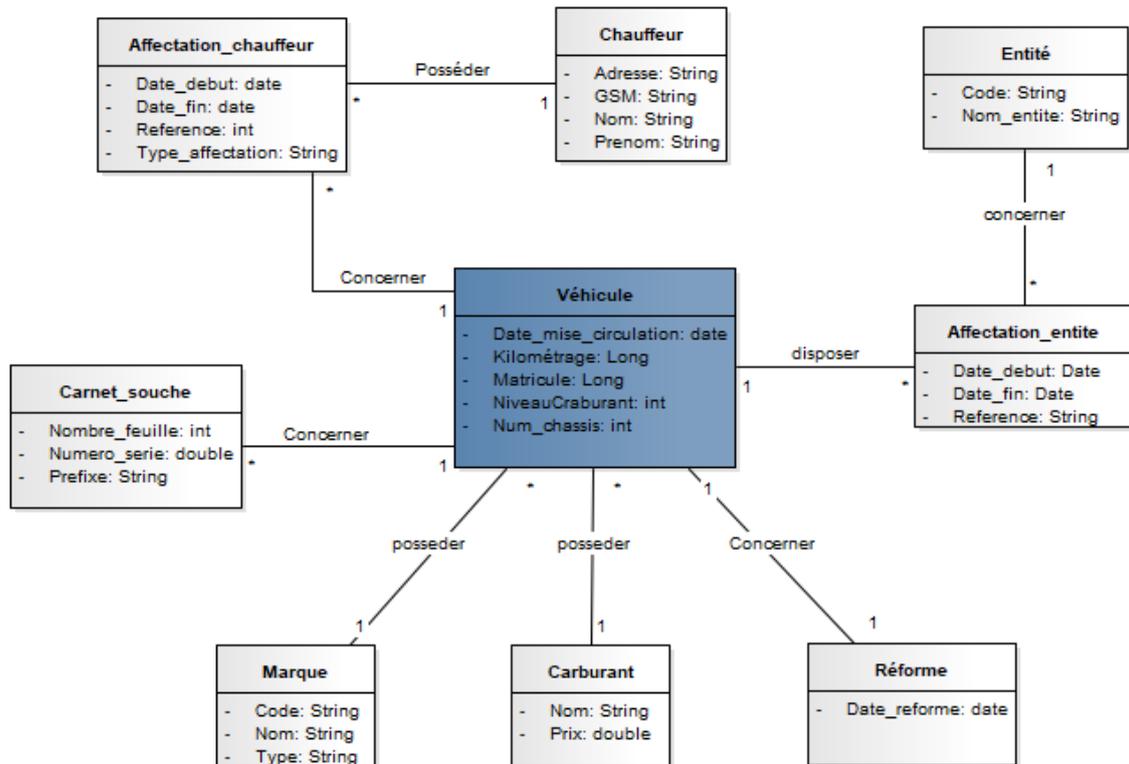


Figure 16 : diagramme de la classe vignette

Un **véhicule** peut être affecté à plusieurs **chauffeurs** à la fois (**affectation\_chauffeur**) mais a une seule entité (**affectation\_entité**), dans des cas exceptionnels (un grave accident ...) le véhicule doit être **réformé** (envoyé à la SNTL).

Le prix du **carburant** varie en fonction du type (essence ou gasoil) et de temps.

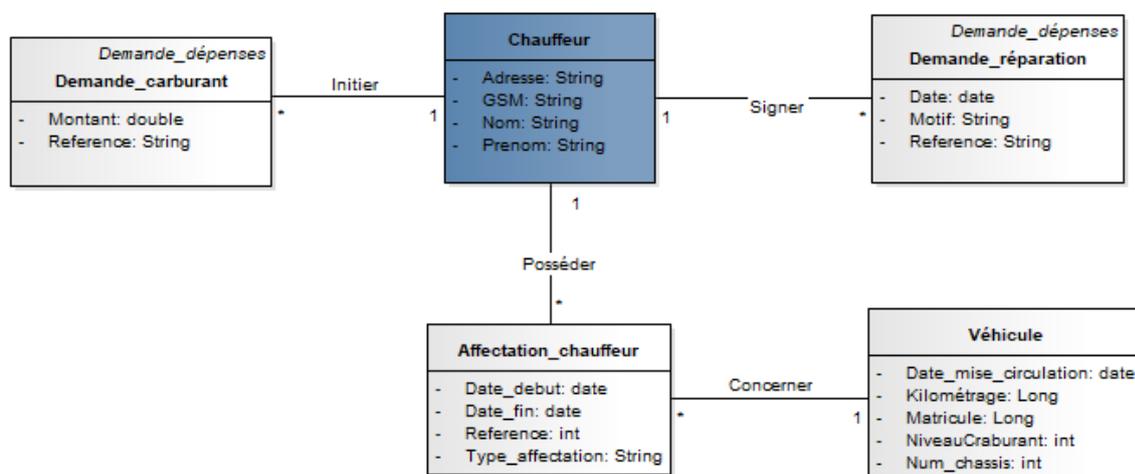


Figure 17 : diagramme de la classe chauffeur

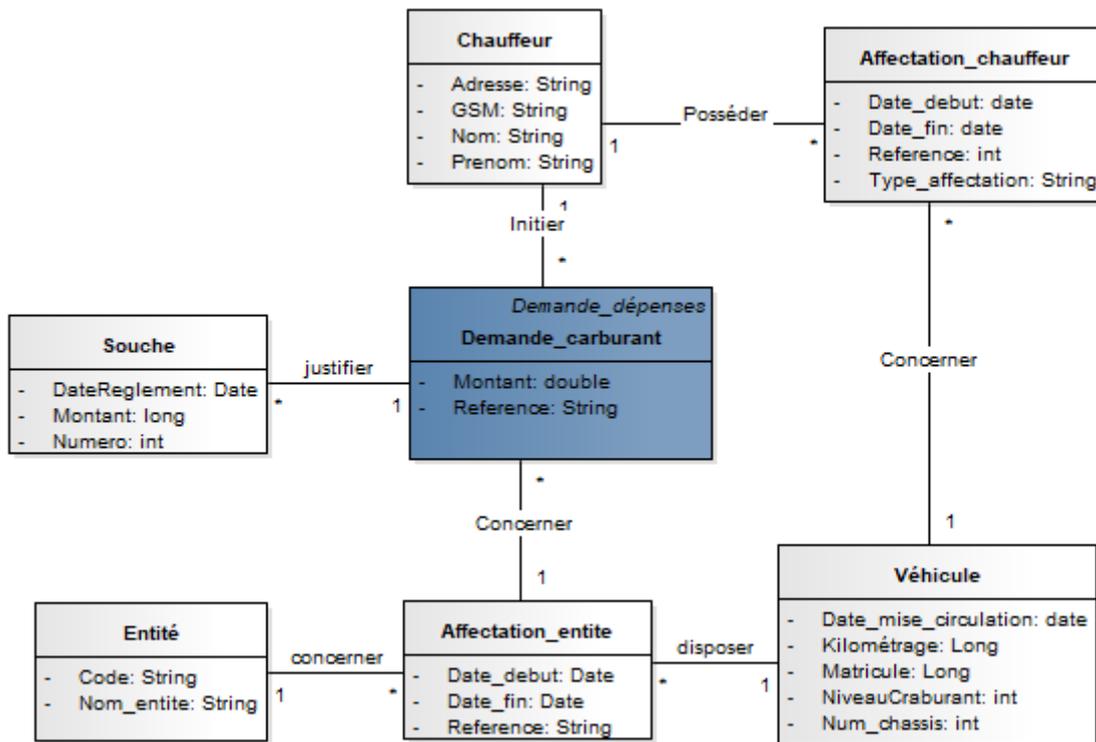


Figure 18 : diagramme de la classe demande carburant

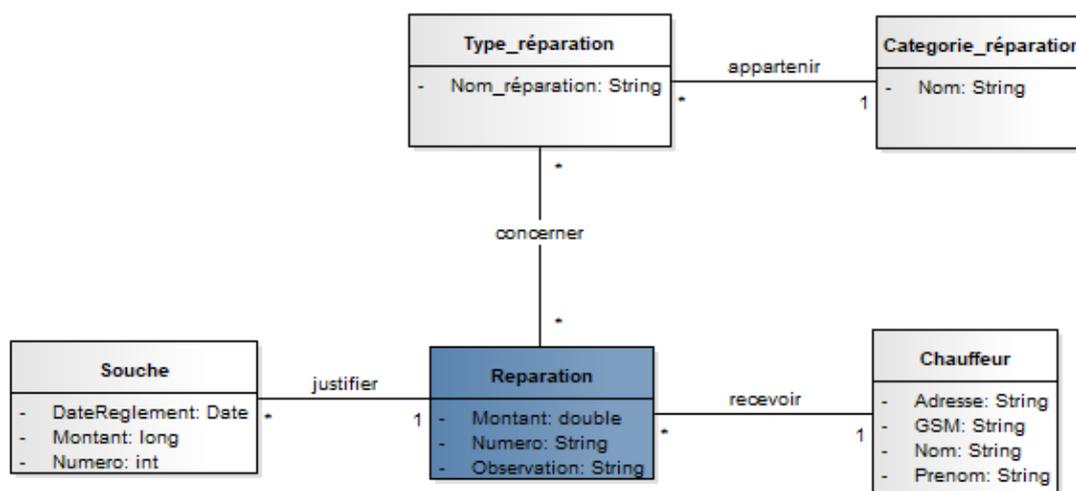


Figure 19 : diagramme de la classe réparation

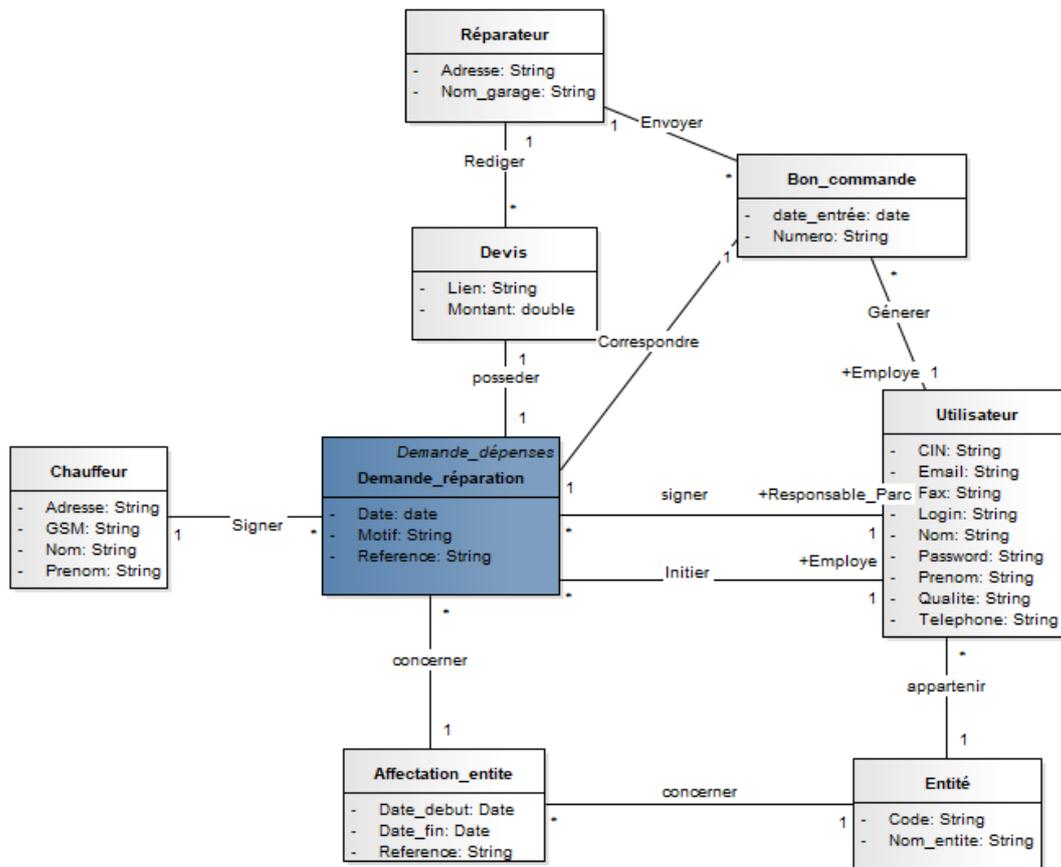


Figure 20 : diagramme de la classe demande réparation

## Conclusion

Le chapitre découlé collecte l'ensemble des besoins auprès de tous les utilisateurs du système, afin de dégager toutes les fonctionnalités explicites et implicites, pour veiller à ce que le système les prenne en charge. Les plateformes logiciel et matériel utilisées dans la réalisation seront présentées dans le chapitre qui suit.



---

## Chapitre 3 : Mise en place de l'application



## Introduction

Dans ce chapitre, nous aborderons la partie réalisation du projet. Plus précisément nous parleront des outils et des langages de développement, puis nous allons entamer les étapes de la réalisation et quelques captures d'écran pour expliquer le fonctionnement du système.

## I. Environnement de réalisation

Lors de développement de notre application, nous avons exploité plusieurs logiciels, dans cette partie allons présenter l'environnement de la réalisation intervenant dans le projet.

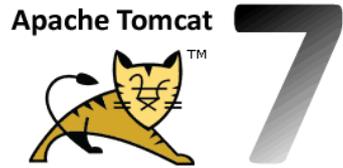
Outils	Version	Logo
Langage de programmation : Java EE		
Serveur : Apache TomCat	TomCat V7.0	
EDI : Eclipse	Mars Release (4.5.0)	
Logiciel de modélisation : Enterprise Architect	EA 12.0.1215	

Tableau 6 : environnement de réalisation

### 1. Java EE

La plateforme Java EE <sup>[2B]</sup> s'appuie entièrement sur le langage Java. Java EE est une norme, qui permet à des développeurs, entreprises et SSII de développer leur propre application qui implémente en totalité ou partiellement les spécifications de SUN. En simplifiant, il est possible de représenter Java EE comme un ensemble de spécifications d'API, une architecture, une méthode de packaging et de déploiement d'applications et la gestion d'applications déployées sur un serveur compatible Java.



## 2. Apache TomCat

Apache TomCat<sup>[12w]</sup> est un serveur HTTP à part entière a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant. Elle permet de générer une réponse HTML à une requête après avoir effectué un certain nombre d'opérations (connexion à une base de données, à un annuaire LDAP...). Pour le client (un navigateur web en général), il n'y a pas de différence avec une page web statique : il reçoit toujours du HTML, seul langage qu'il comprend. Seule la manière dont la réponse est formée côté serveur change.

## 3. Eclipse

Eclipse Mars Release (4.5.0) est un environnement de développement intégré aidant à écrire, compiler, déboguer et déployer des programmes, s'appuyant principalement sur Java. Il y a également un grand nombre de modules pour étendre l' IDE eclipse. Il fait partie des produits du projet Eclipse<sup>15</sup>. Les IDEs eclipse sont des produits gratuits, sans aucune restriction quant à leur usage.

## 4. Entreprise architecte

Enterprise Architect <sup>[14w]</sup> permet le développement d'applications selon le schéma d'architecture orienté modèle ainsi que le schéma d'Architecture orientée services. Enterprise Architect couvre tous les aspects du cycle de développement d'applications depuis la gestion des exigences, en passant par les phases de conception, la construction, tests et maintenance. Ces aspects sont appuyés par des fonctions de support tels que la traçabilité, la gestion de projet, ou encore le contrôle de version.

# II.Langages et technologies utilisées

Nous allons décrire brièvement dans cette partie, les technologies utilisés selon les normes du model en couches d'une application Java EE.

---

<sup>15</sup> Eclipse <sup>[13w]</sup> est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels de la fondation Eclipse, visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java



## A. Architecture Java EE

L'architecture trois tiers, aussi appelée architecture à trois niveaux ou architecture à trois couches, est l'application du modèle plus général qu'est le multi-tiers. L'architecture logique du système est divisée en trois niveaux ou couches :

- La couche présentation : la présentation des données, correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur, plus précisément c'est là où on développe les IHMs et leurs contrôleurs.
- La couche traitement : le traitement métier des données, correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- La couche donnée : c'est là où il y a la base de données relationnel avec une couche intermédiaire qui gère l'accès aux données persistantes et assure le mapping objet-relationnel.



Figure 21: Architecture JAVA2EE



## B. Technologies utilisées

### 1. La couche présentation

Description	Logo
<p>JSF <sup>[3w]</sup> (Java Server Faces) est une technologie dont le but est de proposer un framework qui facilite et standardise le développement d'applications web avec Java. Son développement a tenu compte des différentes expériences acquises lors de l'utilisation des technologies standard pour le développement d'applications web (servlet, JSP, JSTL) et de différents frameworks (Struts, ...).</p>	
<p>HTML <sup>[4w]</sup> L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. Langage de balisage hyper texte Permet de structurer et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables.</p>	
<p>Le terme CSS <sup>[5w]</sup> est l'acronyme anglais de « Cascading Style Sheets » qui peut se traduire par « feuilles de style en cascade ». Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.</p>	
<p>JavaScript <sup>[6w]</sup>, souvent abrégé en JS, est le langage de script développé par Netscape utilisé dans des millions de pages web et d'applications serveur dans le monde entier. Le JavaScript de Netscape est une extension du langage de script standard ECMA-262 Edition 3 (ECMAScript), ne différant que légèrement des standards publiés. JavaScript est un langage léger, interprété, orienté objet (les fonctions étant des objets à part entière).</p>	

Tableau 7 technologies utilisées dans la couche présentation



## 2. La couche traitement

Description	Logo
<p>SPRING <sup>[9w]</sup> est un conteneur dit « léger », c'est-à-dire une infrastructure similaire à un serveur d'application J2EE. Il prend donc en charge la création d'objets et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre ces objets. Le gros avantage par rapport aux serveurs d'application est qu'avec SPRING, vos classes n'ont pas besoin d'implémenter une quelconque interface pour être prises en charge par le framework (au contraire des serveurs d'applications J2EE et des EJBs). C'est en ce sens que SPRING est qualifié de conteneur « léger ».</p>	
<p>Maven <sup>[10w]</sup> est un outil de construction de projets (build) open source développé 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.</p>	

Tableau 8 technologies utilisées dans la couche traitement

## 3. La couche données

Description	Logo
<p>La Java Persistence API (abrégée en JPA), est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java. La Java Persistence API est à l'origine issue du travail du groupe d'experts JSR 220.</p>	
<p>SQL Server Management Studio <sup>[8w]</sup> (SSMS) est un environnement intégré d'accès, de configuration, de gestion, d'administration et de développement de tous les composants de SQL Server. SSMS associe un groupe d'outils graphiques à des éditeurs de script performants pour permettre aux développeurs et aux administrateurs de tous niveaux d'avoir accès à SQL Server. Cette version présente une meilleure compatibilité avec les versions précédentes de SQL Server, un programme d'installation web autonome, ainsi que des notifications toast dans SSMS quand de nouvelles versions deviennent disponibles.</p>	



Microsoft SQL<sup>[11w]</sup> Server est un système de gestion de base de données (abrégé en SGBD) incorporant entre autres un SGBDR (SGBD relationnel) utilisant un langage informatique normalisé servant à exploiter des bases de données relationnelles, développé et commercialisé par la société Microsoft.



Tableau 9 technologies utilisées dans la couche données

### III. Présentation de l'application

Après une analyse approfondie, nous avons décidé de découper l'application en modules : budget, dépense, paramétrage et statistique, dont nous avons réalisé le module « dépense » et commencé la réalisation du module paramétrage.

#### 1. L'interface « Authentification »

Il très important de noter que chaque opération ne peut être réalisée qu'après authentification. Seul « l'Administrateur » qui se charge de définir les login et mot de passe de chaque employé. Au lancement de l'application, la fenêtre d'authentification s'affiche automatiquement. Elle permet à l'utilisateur de se connecter. S'il y a des problèmes lors de la connexion, un message d'erreur sera affiché.

The image shows a simple authentication form. On the left, the labels 'Login' and 'Mot de passe' are written in a teal color. To the right of each label is a rectangular input field with a yellow background and a thin black border. Below these two fields is a single blue button with white text that reads 'Se connecter'.

Figure 22 : Interface authentification



Après l'authentification, on trouve ce menu comme entête de l'application. Il permet à l'utilisateur d'accéder facilement à toutes les fonctionnalités de l'application.



Figure 23 : entête de l'application pour le profil employé

## 2. « Demande Carburant »

L'application permet à l'utilisateur de lister les demandes de carburant déjà enregistrées et rechercher en appliquant tous les filtres possibles.

Référence	Date Demande	Chauffeur	Véhicule	Entité	Kilométrage	Montant	Etat des demandes	Actions
111	22/05/2017		M-9876-2	DSI	110000.0	600.0	En cours	[Edit] [Delete]
111	22/05/2017		A-1205-25	Service des systèmes d'Information	12000.0	600.0	En cours	[Edit] [Delete]
111	23/05/2017	ahmed ahmed	A-1205-25	Service des systèmes d'Information	20000.0	600.0	En cours	[Edit] [Delete]

Figure 24 IHM demande carburant liste



Pour ajouter ou modifier une demande l'employé fait appel à cette interface :

**DÉTAIL DE LA DEMANDE**

1 2 3  
Demande Vignettes Règlement

Enregistrer Accès rapide

Référence  Etat des demandes

Date Demande \*

Chauffeur \*

Véhicule

Véhicule \*

Dernier kilométrage

Affectation

Kilométrage \*

Montant \*

Suivant

Figure 25 : IHM ajouter/modifier demande carburant



L'étape suivante est d'attribuer les vignettes au demandes en utilisant cette interface qui est commune entre le paiement des dépenses (réparation/carburant) :

**DÉTAIL DE LA DEMANDE**

1      2      3  
Demande      Vignettes      Règlement

Attribuer vignettes      Accès rapide

Montant de la demande      600,00 DH

Le montant en vignette      600,00 DH

Carnet de vignette n° : 1      Carnet de vignette n° : 2      Carnet de vignette n° : 3

Montant global du carnet      15000.0

200 DH	100 DH
Quantité disponible      47	Quantité disponible      20
Quantité à saisir      3	Quantité à saisir      0

50 DH	20 DH
Quantité disponible      30	Quantité disponible      50
Quantité à saisir      0	Quantité à saisir      0

10 DH
Quantité disponible      50
Quantité à saisir      0

Suivant

Figure 26 : IHM attribuer les vignettes



Ensuite il faut garder la traçabilité des dépenses par l'enregistrement des souches utilisées pour régler la dépense, donc on utilise cette interface :

**DÉTAIL DE LA DEMANDE**

1 Demande      2 Vignettes      3 **Règlement**

Accès rapide

**Souche**

Carnet de paiement \*       N° du bon \*

Nombre de bons       Date de règlement \*

Réparateur

Générer

Numero	Date de règlement	Réparateur	Montant	Pièce jointe	Actions
CZ000003	06/06/2017		380.0		
CZ000002	06/06/2017		20.0		
CZ000001	06/06/2017		200.0	Aucune pièce n'est jointe	

[ 3 Nbr lignes ]    << << 1 >> >> 10

Figure 27 : IHM règlement de facture

### 3. « Demande de réparation »

L'application permet aussi à l'utilisateur de lister les demandes de réparation déjà enregistrées et rechercher en appliquant tous les filtres possibles.



### DEMANDES DE REPARATION

+ NouveauExporterAccès rapide

**Filtre par défaut**

Référence	<input type="text"/>	Chauffeur	<input type="text" value="Tapez nom ou prénom"/>
Véhicule	<input type="text" value="Tapez le Matricule ou la Marque"/>	Entité	<input type="text" value="Tapez le code ou le nom de l'entité"/>
Panne	<input type="text" value="Tapez le nom de la panne"/>		

Date Demande

Date début	<input type="text"/>
Date fin	<input type="text"/>

[Filtrer](#)

Référence	Chauffeur	Entité	Véhicule	Motif	Actions
147	ahmed ahmed	DSI	A-1205-25	iaaaaa	<a href="#">Edit</a> <a href="#">Print</a> <a href="#">Delete</a>

[ 1 Nbr lignes ]    << << 1 >> >>

Figure 28: IHM demandes de réparation liste

Pour ajouter ou modifier une demande l'employé fait appel à cette interface :



**DEMANDE DE RÉPARATION**

1      2      3

Demande de réparation      Bon de commande      Réception et règlement

EnregistrerAccès rapide

Référence       Date Demande \*

Véhicule \*

Véhicule

Chauffeur \*

Le dernier kilométrage

Affectation

Remarques

Motif

→ Suivant

Figure 29: : IHM ajouter/modifier demande de réparation

Lorsque la demande de réparation est enregistrée, l'employé peut associer un bon de commande en appelant l'interface suivante :



**DEMANDE DE RÉPARATION**

1      2      3

Demande de réparation      Bon de commande      Reception et règlement

Enregistrer Accès rapide

---

**Demande de réparation**

Référence : 147      Date Demande : 2017-06-06 00:00:00.0

Chauffeur : ahmed ahmed      Véhicule : A-1205-25

Affectation : DSI

---

**Bon de commande**

Numero       Date d'entrée \*

Réparateur \*       Montant \*

← Précédent      Suivant →

Figure 30: IHM ajouter/modifier bon de commande

#### 4. « Paramétrage »

Parmi les listes de paramétrage réalisées, il y a une liste très importante qui représente la brique capitale du métier du parc auto. C'est le véhicule.

D'après la liste des véhicules on peut affecter un véhicule à une entité et consulter l'historique des affectations



### LISTE DES VÉHICULES

[+ Nouveau](#) [Exporter](#) [Accès rapide](#)

**Filtre par défaut**

Numéro de châssis  Matricule

Marque  Niveau de carburant

Kilométrage

de  à

Date de mise en circulation

Date début  Date fin

[Filtrer](#)

Matricule	Numéro de châssis	Kilométrage	Niveau de carburant	Marque	type de carburant	Affectation	Actions
A-1205-25	12121212	600.0	4	Renault null	Gasoil	DSI	<a href="#">✎</a> <a href="#">✎</a> <a href="#">✎</a>
M-9876-2	11111	500.0	4	Audi null	Essence	Service des systèmes d'information	<a href="#">✎</a> <a href="#">✎</a> <a href="#">✎</a>
M-1111-5	222222	500.0	2	Renault null	Gasoil	DSI	<a href="#">✎</a> <a href="#">✎</a> <a href="#">✎</a>

[ 3 Nbr lignes ] 1 10

Figure 31 : IHM liste des véhicules

### Ajouter véhicule

Matricule \*

Numéro de châssis \*

Date de mise en circulation \*

Niveau de carburant \*

Kilométrage \*

type de carburant  Gasoil  Essence

Marque \*

[Enregistrer](#) [Retour](#)

Figure 32: IHM ajouter un véhicule



détail d'affectations

Entité  Date affectation

Historique des affectations		
Entité	Date affectation	Date fin d'affectation
DSI	01/06/2017	
SG	01/06/2017	01/06/2017
Service des systèmes d'information	01/01/2017	01/06/2017
SG	01/01/2016	01/12/2016

[ 1 ]

Enregistrer Retour

Figure 33: liste/ajouter une affectation entité

LISTE DES CHAUFFEURS

+ Nouveau Exporter Accès rapide

Filtre par défaut

Nom  Prénom

GSM  Adresse

Filtrer

Nom	Prénom	GSM	Adresse	Actions
ahmed	ahmed	0		<input type="text"/> <input type="text"/> <input type="text"/>
ibrahim	ibrahim	0		<input type="text"/> <input type="text"/> <input type="text"/>
aaaaaaa	bbbbbbbbbb	612121212	aaaaazzzz	<input type="text"/> <input type="text"/> <input type="text"/>

[ 3 Nbr lignes ] 1 10

Figure 34 : Liste des chauffeurs



Filtre par défaut

Nom

GSM

Nom

ahmed

ibrahim

aaaaaaa

ibrahim

bbbbbbbbbb

0

612121212

aaaaazzzzz

Actions

Enregistrer

Retour

### Ajouter un chauffeur ✕

Nom \*

Prénom \*

GSM \*

Adresse \*

Véhicule	Type d'affectation	Date affectation	Date fin d'affectation
A-1205-25		02/06/2017	
A-1205-25		02/06/2017	
A-1205-25	Utilitaire	05/06/2017	

Figure 35 : pop-up Ajouter chauffeur

LISTE DES CHAUFFEURS

### Véhicules affectés au chauffeur ✕

Véhicule

Type d'affectation

Date affectation

Date fin

Historique des affectations

Véhicule	Type d'affectation	Date affectation	Date fin d'affectation
A-1205-25		02/06/2017	
A-1205-25		02/06/2017	
A-1205-25	Utilitaire	05/06/2017	

Enregistrer

Retour

Figure 36 : pop-up affectation chauffeur

### Message de confirmation ✕

⚠️ Voulez-vous vraiment supprimer cet élément?

Oui

Non

Figure 37: message informatif de suppression



## Conclusion générale

Ce stage a été l'occasion de mettre à profit nos connaissances acquises à la Faculté des Sciences et Techniques Fès, notamment en termes de programmation, conception et de modélisation pour pouvoir concrétiser notre bagage théorique. Il nous a permis également d'utiliser, comprendre et se familiariser avec de nouveaux outils de développement.

Au cours de la réalisation de notre projet, nous avons été attachées par quelques limites notamment, la confrontation des nouvelles connaissances qui nous obligeait à s'autoformer dans un temps qui faisait lui-même un obstacle devant l'accomplissement du projet dans le délai prévu du stage. Ainsi nous avons appris des différentes technologies de Java EE à savoir : les Framework prime faces, spring, ...

Ce projet nous a donné l'opportunité de s'intégrer réellement dans la vie professionnelle et commencer notre carrière d'une expérience significative qui nous aidera certainement à continuer dans la bonne voie, comment réussir de bonnes relations, assurer le travail du groupe, résoudre les problèmes qui se présentent au cours du projet et surtout comment profiter le maximum possible des indications de nos encadrants académique, et professionnel.

En définitive, ce stage a été une expérience déterminante dont nous allons évidemment profiter au niveau professionnel et personnel.

Comme perspectives de ce travail, nous comptons continuer la réalisation des modules restants juste après le stage et cela dans le but d'améliorer nos connaissances en termes de moyens techniques et personnels.



## Bibliographie

**[1B] : définition UML** : Unified Modeling Language (UML). Article de Bruno BOUZY

**[2B] : Java EE-Guide de développement d'applications web en Java.**  
**Jérôme LAFOSSE**

## Webographie

**[1w] : Définition Scrum :**

[http://fr.wikipedia.org/wiki/Scrum\\_\(m%C3%A9thode\)](http://fr.wikipedia.org/wiki/Scrum_(m%C3%A9thode))

**[2w] : définition agile :** [www.agiliste.fr/introduction-methodes-agiles/](http://www.agiliste.fr/introduction-methodes-agiles/)

**[3w] : définition JSF:** <http://www.jmdoudoux.fr/java/dej/chap-jsf.htm>

**[4w] : définition HTML:** <http://slideplayer.fr/slide/4345759>

**[5w] : définition CSS:** <http://glossaire.infowebmaster.fr/css>

**[6w] : définition JavaScript :**

[https://developer.mozilla.org/fr/docs/Web/JavaScript/A\\_propos](https://developer.mozilla.org/fr/docs/Web/JavaScript/A_propos)

**[7w] : définition JQuery:** <http://distilleri.es/javascript/jquery>

**[8w] : définition SQL management studio :**

<https://msdn.microsoft.com/fr-fr/library/mt238290.aspx>

**[9w] : définition SPRING :** Erik Gollot - introduction du document « Introduction au framework »

**[10w] : définition MAVEN :** <https://www.jmdoudoux.fr/java/dej/chap-maven.htm>

**[11w] : définition Microsoft SQL server :**

[https://fr.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://fr.wikipedia.org/wiki/Microsoft_SQL_Server)

**[12w] : définition ApacheTomcat:**

[https://fr.wikipedia.org/wiki/Apache\\_Tomcat](https://fr.wikipedia.org/wiki/Apache_Tomcat)

**[13w] : définition Eclipse :** [https://fr.wikipedia.org/wiki/Eclipse\\_\(projet\)](https://fr.wikipedia.org/wiki/Eclipse_(projet))

**[14w] : définition enterprise-architect :** [http://republic.pink/enterprise-architect\\_1439556.html](http://republic.pink/enterprise-architect_1439556.html)



## Annexes

Annexes N°1 : Ordre de mission.

Annexes N°2 : Extrait du circulaire N°98/4

Annexes N°3 : Décharge / Affectation.

Annexes N°4 : Demande de carburant.

Annexes N°5 : Demande de réparation / bon de commande.

Annexes N°6 : PV de réception.

Annexes N°7 : Demande de VTT.

Annexes N°8 : Extrait du code de l'application.



---

## Annexes N°1-1 : *Vignette exemple*

---





## Annexes N°1 : *Ordre de mission*

Royaume du Maroc  
Le Chef du Gouvernement



المملكة المغربية  
رئيس الحكومة

الرباط في : 11/05/2017

رقم :

### أمر بمهمة

\*\*\*\*\*

إن الوزير المنتدب لدى رئيس الحكومة المكلف بالوظيفة العمومية وتحديث الإدارة يأمر :

- السيد :
- مهمته :
- التعيين :

باستعمال السيارة التالية :

- نوع : Honda Accord
- رقم التسجيل : M 187312
- العداد الكيلوميتري :

من أجل التنقل عبر الخط الرابط بين الرباط و الدار البيضاء المطار  
و سيرافقه السادة

وتتمد مدة صلاحية هذا الأمر من : 09/05/2017 إلى 09/05/2017

إمضاء رئيس الإدارة

## Annexes N°2 : Extrait du circulaire N°98/4

**أ - السيارات الوظيفية :**

تعتبر سيارة وظيفية كل سيارة السياقة الداخلية مخصصة ، بصفة فردية ، لأعضاء الحكومة ومن في حكمهم ، والسفراء والكتاب العامين لوزارات ومن في حكمهم ، ولرؤساء دواوين الوزراء .

ويحدد عدد السيارات الوظيفية كما يلي :

3	الوزير الأول
2	الوزير العضو في الحكومة
1	كاتب الدولة ونائب كاتب الدولة
1	السفير
1	كاتب عام وزارة
1	رئيس الديوان

Figure 38 : véhicule de fonction

**ب - سيارات المأمورية :**

تعتبر سيارة مأمورية كل سيارة غير مخصصة بصفة فردية ومرصودة لنقل موظفي وأعيان الدولة المكلفين ، في إطار مزاولتهم لعملهم ، بمأموريات محددة في الزمان والمكان ، وكذا لنقل كل شخص مرخص له بذلك من طرف رئيس الإدارة المعنية .

وتعد كذلك بمثابة سيارة مأمورية السيارات المخصصة لنقل الشخصيات المدعوة أو الخبراء القائمين بمهمة لدى الإدارات العمومية .

Figure 39 : véhicule de mission

**ج - العربات النفعية :**

تعتبر عربة نفعية :

- كل عربة نفعية غير مخصصة بصفة فردية ومرصودة إما لنقل الجماعي لموظفي وأعيان الدولة أو لنقل معدات وممتلكات الإدارة .

- العربات الخاصة المخصصة لانشائها بقرار وزير المالية والاستثمارات الخارجية رقم 96-1633 بتاريخ 29 ربيع الأول 1417 ( 15 غشت 1996 ) بتحديد قائمة العربات الخاصة المعفاة من الضريبة الإضافية على تسجيل العربات ذات المحرك ، باستثناء أليات الأوراش .

- كل عربة ورش أو عربة للاستعمال في كل الطرق .

Figure 40 : véhicule utilitaire



## Annexes N°3 : PV de décharge/affectation

Royaume du Maroc  
Le Chef du Gouvernement  
MINISTÈRE DE LA FONCTION PUBLIQUE  
ET DE LA MODERNISATION  
DE L'ADMINISTRATION  
ⵜⴰⴳⴷⴰⵏⵜ ⵜⴰⵎⴳⴷⴰⵢⵜ ⵜⴰⵏⵔⴰⵏⵜ  
ⵏ ⵜⴰⵎⴳⴷⴰⵢⵜ ⵜⴰⵏⵔⴰⵏⵜ  
Direction des Ressources  
Humaines et Financières



المملكة المغربية  
رئيس الحكومة  
وزارة الوظيفة العمومية  
وتحديث الإدارة  
ⵜⴰⴳⴷⴰⵏⵜ ⵜⴰⵎⴳⴷⴰⵢⵜ ⵜⴰⵏⵔⴰⵏⵜ  
ⵏ ⵜⴰⵎⴳⴷⴰⵢⵜ ⵜⴰⵏⵔⴰⵏⵜ  
مديرية الموارد  
البشرية والمالية

### PV DE DECHARGE OU AFFECTATION

\*\*\*\*\*

Décharge

Affectation

Chauffeur :

Affectation : D R H F

Matricule : M

Marque : Renault Master

Kilométrage : 67

Date : 16/02/2016

Niveau du carburant : 0

Etat du véhicule : Véhicule neuf  
3 extincteurs  
Attestation d'assurance  
Roue de secours avec accessoires.  
Carnet de bons : DE 318601 à 318650.

Fait à Rabat, le 11/05/2017

Signature



## Annexes N°4 : demande de carburant

Royaume du Maroc  
Le Chef du Gouvernement  
MINISTÈRE DE LA FONCTION PUBLIQUE  
ET DE LA MODERNISATION  
DE L'ADMINISTRATION  
ⵜⴰⴳⴷⴰⵢⵜ ⵜⴰⵎⴳⴷⴰⵢⵜ ⵜⴰⵏⴳⴷⴰⵢⵜ  
ⵏ ⵜⴰⵎⴳⴷⴰⵢⵜ ⵜⴰⵏⴳⴷⴰⵢⵜ  
Direction des Ressources  
Humaines et Financières



المملكة المغربية  
رئيس الحكومة  
وزارة الوظيفة العمومية  
وتحديث الإدارة  
ⵜⴰⴳⴷⴰⵢⵜ ⵜⴰⵎⴳⴷⴰⵢⵜ ⵜⴰⵏⴳⴷⴰⵢⵜ  
ⵏ ⵜⴰⵎⴳⴷⴰⵢⵜ ⵜⴰⵏⴳⴷⴰⵢⵜ  
مديرية الموارد  
البشرية والمالية

### طلب سمات الوقود لشهر ماي 2017

الرباط في : 11/05/2017

رقم:

المصلحة المستفيدة	نوع السيارة	رقم السيارة	المبلغ	العداد الكيلوميتري	المجموع من أول الشهر
<input type="text"/>	<input type="text"/>	<input type="text"/>	250.00 (درهم)	288,136	500.00 (درهم)

العداد السابق	التاريخ	العداد الحالي	الفرق	المبلغ	نسبة الاستهلاك (l/100km)
287,734	02/05/2017	288,136	402	250.00	7,07

أنا الموقع أسفله السيد :

أصرح أنني توصلت بمبلغ مائتين وخمسين درهم الخاص ب : الوقود

ملاحظات	مصلحة الشؤون العامة	مكتب حظيرة السيارات	الرئيس المباشر	للاستخلاص السائق	ملاحظات
		<input type="text"/>			



## Annexes N°5 : *demande de réparation*

Direction des Ressources et Systèmes d'Information

### DEMANDE DE REPARATION DE VEHICULE DE SERVICE

(A formuler et à signer par le responsable de l'entité concernée et à adresser au chef de service matériel)

REF. DEMANDE : **DR.2017.050**

DATE DEMANDE : **22/03/2017**

MARQUE DE VEHICULE : **IVECO**

IMMATRICULATION : ..... KILOMETRAGE : **133162**

AFFECTATION : **Direction de la Formation Professionnelle et de la Formation Continue des Artisans**

CHAUFFEUR : .....

NATURE DE RAPARATION : **révision**

Signature de l'utilisateur du véhicule et du vélomoteur

Fait à Rabat Le :

Signature du responsable du Parc Auto

Fait à Rabat Le :



Direction des Ressources et Systèmes d'Information

---

**BON DE COMMANDE DE REPARATION**

REFERENCE : BR.2017.008

GARAGE : **Atlas véhicules industriels** .....

MARQUE DE VEHICULE : **IVECO** .....

IMMATRICULATION : .....

AFFECTATION : **Parc Auto - Directions centrales** .....

OBJET DE REPARATION : .....

.....

.....

Fait à Rabat Le : 20/03/2017



## Annexes N°6 : Réception réparation

Direction des Ressources et Systèmes d'Information

### PV DE RECEPTION

Fait à Rabat Le : 22/03/2017

REFERENCE : **PR.2017.039**

IMMATRICULATION :

AFFECTATION : **Parc Auto - Directions centrales**

FOURNISSEUR : **Atlas véhicules industriels**

OBSERVATIONS :

#### PRESTATIONS :

<b>PIECE :</b>	<b>QUANTITE :</b>
Lampe projecteur	1
Feu de signale	1
Lave glace	1



Royaume du Maroc  
Le Chef du Gouvernement  
MINISTÈRE DE LA FONCTION PUBLIQUE  
ET DE LA MODERNISATION  
DE L'ADMINISTRATION  
ⵜⴰⴳⴷⴰⵢⵜ ⵜⴰⴳⴷⴰⵢⵜ ⵜⴰⴷⵓⵏⵏⵉⵢⵜ  
ⵏ ⵜⴰⴳⴷⴰⵢⵜ ⵜⴰⴷⵓⵏⵏⵉⵢⵜ  
Direction des Ressources  
Humaines et Financières



المملكة المغربية  
رئيس الحكومة  
وزارة الوظيفة العمومية  
وتحديث الإدارة  
ⵜⴰⴳⴷⴰⵢⵜ ⵜⴰⴳⴷⴰⵢⵜ ⵜⴰⴷⵓⵏⵏⵉⵢⵜ  
ⵏ ⵜⴰⴳⴷⴰⵢⵜ ⵜⴰⴷⵓⵏⵏⵉⵢⵜ  
مديرية الموارد  
البشرية والمالية

## Demande de réparation No: 69/2017

### I. Renseignement du conducteur

Matricule :	M 167699	Marque :	Dacia Logan
Direction :	Secrétariat Général		
Index kilométrage :	136,610		
Description détaillée des anomalies constatées :			
<input checked="" type="checkbox"/> Plaquettes de frein - - -			
Nom et signature du conducteur :		Date : 11/05/2017	
		Montant : .....	

### II. Rapport d'expertise du véhicule

Réparateur :	Garage BENBERQA	
Date d'entrée au garage :	11/05/2017	Date de sortie :
Description (responsable du parc + garagiste)		
<div style="border-bottom: 1px dotted black; width: 100%;"></div> <div style="border-bottom: 1px dotted black; width: 100%;"></div>		
Responsable	Responsable Parc auto	Pour l'Administration
d'entité		

### III. Réception de la réparation après vérification

Date de réception :		Retard :
Garagiste :	Responsable Parc auto :	Chauffeur :
	Réglé par bon n° : .....	Date de règlement : .....



## Annexes N°7 : Demande des VTT

Royaume du Maroc  
Le Chef du Gouvernement  
MINISTÈRE DE LA FONCTION PUBLIQUE  
ET DE LA MODERNISATION  
DE L'ADMINISTRATION  
ⵜⴰⴳⴷⴰⵢⵜ ⵏ ⵏⴰⵎⴻ ⵏ ⵏⴰⵙⴳⴷⴰⵢⵜ  
ⵏ ⵏⴰⵙⴳⴷⴰⵢⵜ ⵏ ⵏⴰⵙⴳⴷⴰⵢⵜ  
Direction des Ressources  
Humaines et Financières



المملكة المغربية  
رئيس الحكومة  
وزارة الوظيفة العمومية  
وتحديث الإدارة  
ⵜⴰⴳⴷⴰⵢⵜ ⵏ ⵏⴰⵎⴻ ⵏ ⵏⴰⵙⴳⴷⴰⵢⵜ  
ⵏ ⵏⴰⵙⴳⴷⴰⵢⵜ ⵏ ⵏⴰⵙⴳⴷⴰⵢⵜ  
مديرية الموارد  
البشرية والمالية

### الاستفادة من بطاقة الطريق السيارة

رقم:

الرباط في : 11/05/2017

المصلحة المستفيدة	نوع السيارة	رقم السيارة	العداد الكيلوميتري	الرصيد الحالي
CABINET	Honda Accord	<input type="text"/>	130539	660.00 (درهم)

أنا الموقع أسفله السيد

أصرح أنني توصلت ببطاقة الطريق السيارة رقم

الإدارة	مكتب حظيرة السيارات	للاستخلاص السائق	ملاحظات



## Annexes N°8 : *Extrait du code de l'application*

```
import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Entity;

@Entity
public class Chauffeur extends BaseEntity implements Serializable {
    private static final long serialVersionUID = 1L;
    // ++++++
    // Attributs //
    // ++++++

    @Column(length = 20)
    private String nom;

    @Column(length = 20)
    private String prenom;

    @Column
    private String GSM;

    @Column(length = 100)
    private String adresse;
    // ++++++
    // Constructeur //
    // ++++++

    public Chauffeur() {
    }
    // ++++++
    // Display Text //
    // ++++++

    @Override
    public String getDisplayText() {
        return nom + " " + prenom;
    }
}
```

Figure 41: la classe chauffeur



```
<h:outputText value="#{lbl['attr.vehicule']}" />
<h:panelGrid>
  <p:autoComplete id="vehiculeFilterId"
    value="#{demandeReparationListController.vehiculeFilter}"
    completeMethod="#{demandeReparationListController.doCompleteVehicule}"
    var="_vehicule" converter="#{entityConverter}"
    forceSelection="true" itemLabel="#{[_vehicule.displayText]}"
    itemValue="#{[_vehicule]}" size="40" scrollHeight="200"
    label="#{lbl['attr.vehicule']}">
    <p:column>
      <f:facet name="header">
        <h:outputText value="#{lbl['attr.matricule']}" />
      </f:facet>
      <h:outputText value="#{[_vehicule.matricule]}" />
    </p:column>

    <p:column>
      <f:facet name="header">
        <h:outputText value="#{lbl['attr.marque']}" />
      </f:facet>
      <h:outputText value="#{[_vehicule.marque]}" />
    </p:column>

  </p:autoComplete>
  <p:watermark for="vehiculeFilterId"
    value="#{lbl['watermark.vehicule']}" />
</h:panelGrid>
```

Figure 42: le champ véhicule

```
public void doRemovebonCommande(BonCommande bonCommande) {
  try {
    bonCommandeService.deleteBonCommande(bonCommande);
    bonCommandeList.remove(bonCommande);
    bonCommande = new BonCommande();
    showMessage("info.delete.ok", FacesMessage.SEVERITY_INFO);
  } catch (StagiairesException e) {
    e.printStackTrace();
    showMessage("error.admin", FacesMessage.SEVERITY_ERROR);
  }
}
```

Figure 43: demande carburant controller (la fonction de suppression)



```
-
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7 import org.springframework.transaction.annotation.Transactional;
8
9 import ma.gov.mmisp.parcAuto.dao.GenericRepository;
10 import ma.gov.mmisp.parcAuto.dao.repositories.ReparateurRepository;
11 import ma.gov.mmisp.parcAuto.model.Reparateur;
12 import ma.gov.mmisp.parcAuto.service.GenericServiceImpl;
13 import ma.gov.mmisp.parcAuto.utility.search.Filter;
14 import ma.gov.mmisp.parcAuto.utility.search.Search;
15
16 @Service
17 @Transactional
18 public class ReparateurServiceImpl extends GenericServiceImpl<Reparateur, Long> implements IReparateurService {
19     private static final long serialVersionUID = -2006163890024900986L;
20
21     @Autowired
22     private ReparateurRepository reparateurRepository;
23
24     @Override
25     protected GenericRepository<Reparateur, Long> getRepository() {
26         return reparateurRepository;
27     }
28
29     @SuppressWarnings("unchecked")
30     @Override
31     public List<Reparateur> getReparateurContainsQuery(String query) {
32         Search search = new Search(Reparateur.class);
33         search.addFilterOr(new Filter("nom", "%" + query + "%", Filter.OP_ILIKE),
34             new Filter("adresse", "%" + query + "%", Filter.OP_ILIKE));
35         return (List<Reparateur>) searchBean.search(search);
36     }
37
38 }
39
```

Figure 44: service réparateur